

TR - 0 - 0028

39

有限要素法を用いた2次元  
静電場解析プログラムマニュアル

角田紀久夫

1990. 6. 8.

ATR光電波通信研究所

## 概要 (Abstract)

電磁界の数値解析手法はマイクロ波回路の設計において有力な道具となっている。これらソフトウェアの充実を計るため各種解析プログラムの自作及び共同開発を進めている。この度、有限要素法による2次元静電場解析プログラムを作成し、そのマニュアルを整備した。

この解析プログラムでは、2次元配置した導体間の静電容量値が求められる。この静電容量値は伝送線路の特性近似計算である準TEMモード近似法に使用できる。

この報告書は、プログラムマニュアルとして位置づけ、プログラムで使用している基本的な数式、プログラムの全体構成及び使用方法、プログラムリストの内容、使用例について説明している。

## 発行時 配付先 (Initial Distribution Specifications)

--

## 備考 (Notes)

--

## まえがき

マイクロ波回路の設計において、伝送線路の設計はひとつの重要な課題である。現在、ATR 光電波通信研究所ではこれら設計のためのソフトウェアの充実を計るため各種解析プログラムの整備を進めている。

この有限要素法 (FEM) による2次元静電場解析プログラムは、多層TEM線路解析用に作成・整備したものであり、本レポートはその使用マニュアルとしてまとめた。

マイクロ波の伝送線路のなかで同軸線路やマイクロストリップ線路などTEMモードまたはTEMに準ずるモードを伝送するいわゆるTEM伝送線路においてはその特性インピーダンスおよび実効誘電率の計算手法として準TEMモード近似法が知られている。この手法は、導波導体間の単位長さにおける静電容量値を用いて線路の電気的特性を求める方法であり、分散が無視できる領域においては簡易で有力な解析として現在でも使用されている。この導波導体間の静電容量値を求めるには2次元ポテンシャル問題を解けばよく、従来から各種解析手法が用いられている。本プログラムでは有限要素法 (FEM) を採用している。

本マニュアルは、以下の構成で記述している。1章は解析法についての簡単な説明、2章はこのプログラムの使用方法を記述したプログラムの使用マニュアルである。また3章はプログラムリスト、4章には実施例を示した。

プログラムコードは、計算時間・記憶容量などの効率化やプログラムの操作性がまだ不十分ではあるが、本説明書を参照のうえユーザーサイドで改訂してほしい。また誤り等発見した場合はこのマニュアルも含めて修正してほしい。

# 1 解析法についての簡単な説明

以下では、本解析手法について簡単な説明を行う。

## 1.1. 有限要素法 (FEM) による静電場解析<sup>1)</sup>

領域 $\Omega$ において定義された2次元ポアソン方程式(1)を近似的に解く方法について述べる。

$$\left( \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} \right) = -\frac{\rho}{\epsilon} \quad (1)$$

いま、近似解を $\psi$ として方程式(1)に代入すると残差 $R(\psi)$ が生じる。近似解(試験関数) $\psi$ が $\phi$ に充分近い値であるならばこの残差 $R(\psi)$ は領域 $\Omega$ において平均的に0となる。このとき重み付け関数 $\psi_0$ を用いて式(2)と表せる。

$$\int_{\Omega} \psi_0 \left( \frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} - \frac{\rho}{\epsilon} \right) d\Omega = 0 \quad (2)$$

(2)式に部分積分を用いて整理すると(3)式となる。

$$\iint_{\Omega} \left( \frac{\partial \psi}{\partial x} \frac{\partial \psi_0}{\partial x} + \frac{\partial \psi}{\partial y} \frac{\partial \psi_0}{\partial y} - \frac{\rho}{\epsilon} \psi_0 \right) dx dy - \int_{\Gamma} \left( \frac{\partial \psi}{\partial n} \psi_0 \right) dl = 0 \quad (3)$$

即ち、積分は領域内部 $\Omega$ とその境界線上 $\Gamma$ の積分項に分けられる。ここで(3)式の第2項は境界 $\Gamma$ からのフラックスを表す。領域 $\Omega$ の内部を複数の要素(有限要素)に分割すると全体の残差は要素ごとの残差の和と考えられる。さらに離散化のためにガラーキン法を適用する。すなわち、重み付け関数を試験関数と同じとし、各要素ごとに式(4)とする。

$$\psi = \psi_0 = \sum_i N_i(x,y) \psi_i \quad (4)$$

ここで $N_i(x,y)$ は要素内部の補間関数(形状関数)、 $\psi_i$ は補間点の値を示す。ここで $S_{ij}, F_i, Q_i$ を各々式(5)-(7)とすると、各要素 $e$ において(3)式は(8)式となる。

$$S_{ij} = \iint_e \left( \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \right) dx dy \quad (5)$$

$$F_i = \iint_e \frac{\rho}{\epsilon} N_i dx dy \quad (6)$$

$$Q_i = \int_{\Gamma} \frac{\partial \psi_i}{\partial n} N_i dl \quad (7)$$

$$\sum_i \sum_j \psi_i S_{ij} \psi_j - \sum_i F_i \psi_i - \sum_i Q_i \psi_i = 0 \quad (8)$$

ここで $N_i$ は補間関数として与えた既知関数であって式(5)-(7)は計算することができ、境界条件を与えることによって結局式(8)は $\psi_i$ を未知数とした連立一次方程式となる。領域 $\Omega$ 全体においては、各要素における式(8)の和として次式となる。

$$\sum_e \left( \sum_i \sum_j \psi_i S_{ij} \psi_j - \sum_i F_i \psi_i - \sum_i Q_i \psi_i \right) = 0 \quad (9)$$

従って、最終的には(10)式の連立一次方程式を解く問題に帰着する。

$$|S_{ij}| |\psi_i| = |F_i - Q_i| \quad (10)$$

ここで境界条件として金属導体部分のポテンシャル値をアース(電圧0ボルト)とホット(電圧1ボルト)として計算すると、式(10)を解いてポテンシャル値  $\psi_i$  を得たのち、式(7)を用いてホット(もしくはアース)導体上の総電荷量を計算すれば静電容量値が求められる。

### 1.2 形状関数<sup>[2]</sup>

本プログラムでは計算精度を考慮し2次三角要素を用いて未知関数の展開を行っている。図1-1に三角形要素を示す。

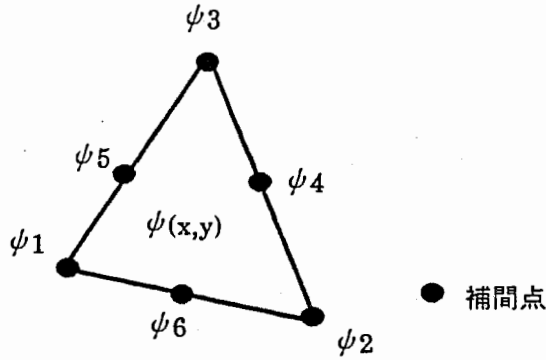


図 1-1 三角形要素

各頂点と各辺の midpoint に補間点をもつ。要素内の補間関数(形状関数)は三角形の頂点の座標を  $(X_1, Y_1), (X_2, Y_2), (X_3, Y_3)$  とするとき、次の様に定義されている。まず、任意の点  $(X, Y)$  において  $L_1, L_2, L_3$  (面積座標値) を次式とする。

$$L_1 = \frac{(X_2 \cdot Y_3 - X_3 \cdot Y_2) + (Y_2 - Y_3) \cdot X + (X_3 - X_2) \cdot Y}{X_1 \cdot Y_2 + X_2 \cdot Y_3 + X_3 \cdot Y_1 - X_2 \cdot Y_1 - X_3 \cdot Y_2 - X_1 \cdot Y_3}$$

$$L_2 = \frac{(X_3 \cdot Y_1 - X_1 \cdot Y_3) + (Y_3 - Y_1) \cdot X + (X_1 - X_3) \cdot Y}{X_1 \cdot Y_2 + X_2 \cdot Y_3 + X_3 \cdot Y_1 - X_2 \cdot Y_1 - X_3 \cdot Y_2 - X_1 \cdot Y_3}$$

$$L_3 = \frac{(X_1 \cdot Y_2 - X_2 \cdot Y_1) + (Y_1 - Y_2) \cdot X + (X_2 - X_1) \cdot Y}{X_1 \cdot Y_2 + X_2 \cdot Y_3 + X_3 \cdot Y_1 - X_2 \cdot Y_1 - X_3 \cdot Y_2 - X_1 \cdot Y_3} \quad (11)$$

これを用いて、形状関数は次式で表される。

$$N_1 = L_1^2 \quad N_2 = L_2^2 \quad N_3 = L_3^2$$

$$N_4 = L_2 L_3 \quad N_5 = L_3 L_1 \quad N_6 = L_1 L_2 \quad (12)$$

### 1.3 準TEMモード近似によるTEM伝送線路の特性解析<sup>[3]</sup>

同一媒質内に進行方向に一様な2導体配置されている、たとえば平行フィード線のような、TEM伝送線路の特性インピーダンス  $Z_0$  および位相定数  $\beta_0$  は線路の単位長さあたりの静電容量  $C_0$  とその媒質内部の光速  $vc$  周波数  $\omega$  を用いると次式となることが知られている。

$$Z_0 = \frac{1}{C_0 \cdot vc} \quad \beta_0 = \frac{\omega}{vc} \quad (13)$$

従って、線路の2導体 (ホット・アース) 間に生じる単位長さあたりの静電容量を計算できれば TEM伝送線路の特性を求められる。

マイクロストリップ線路のように伝送路を構成する空間に複数の媒質が存在する場合、その伝搬モードは厳密にはTEMモードからずれるが、その変動量が小さければTEMモードに近い取扱いができる。これを準TEMモード近似法という。すなわち、線路の単位長さあたりの静電容量Cを用いて伝送線路の特性を求めるものであって、特性インピーダンスZや位相定数 $\beta$ は式(14)で求められる。

$$Z = \sqrt{\frac{C_0}{C}} Z_0 \qquad \beta = \sqrt{\frac{C}{C_0}} \beta_0 \qquad (14)$$

すなわち誘電体等の媒質が存在するときと存在しないときの容量値を求めればよい。

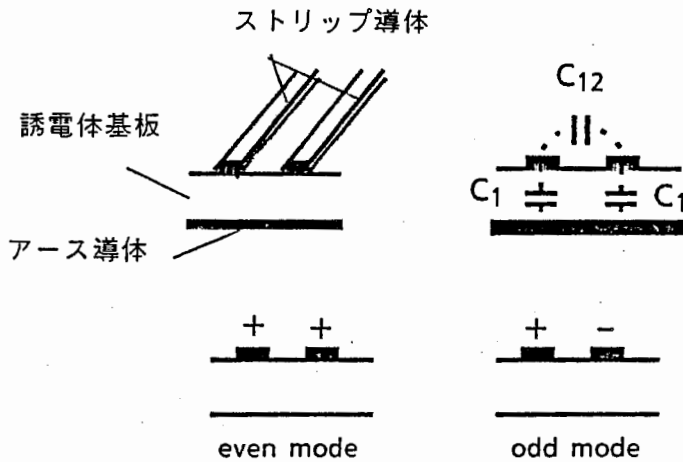


図 1-2 結合マイクロストリップ線路

TEM結合線路の場合でも結合モードを考慮することにより各モードにおける伝送線路特性が計算できる。今、図1-2に結合マイクロストリップ線路を示す。このような結合線路の2つの直交モード(even modeとodd mode)における特性は、各導体間容量 $C_1, C_{12}$ によって、even modeでは  $C = C_1$  , odd modeでは  $C = C_1 + 2 C_{12}$  を用いて(14)式により計算できる。すなわち、even mode ではストリップ導体をホットとし、アース導体をアースとしたときの静電容量値を用いる。またodd modeでは、一方のストリップ導体をホット、他方のストリップ導体を-1ボルト、アース導体をアースとしたときの静電容量値を用いればよい。

#### 1.4. その他

有限要素法においては要素の内部は形状関数によって補間・近似しているため、複雑な電界分布をしている場所は、細かく要素分割し、その電界分布の変化を表現できるようにしなければならない。そのために、解析領域における要素分割数と計算結果の比較を行うことが必要である。このとき一般に、要素を増すに従い計算値はある値に漸近する。この結果を見ながら要素分割数を決定しなければならない。

本プログラムでは、ホット導体上の電荷量を用いて直接的に容量値を計算しており、容量マトリクス作成機能は導入していない。従って複数の導体間容量を計算する場合は、各導体のポ

テンシャル設定値を適切に選び、考察するモードにおける線路特性計算に用いる静電容量値を求めなければならない。

また、浮遊導体の処理機能は導入していない。浮遊導体を処理するためには、式(10)において、同一浮遊導体上における未知数 $\psi_i$ をそれぞれ等しく置けば良いが、そのためにはプログラムコードの修正が必要である。

本プログラムは開放境界の表現は導入していない。開放領域を必要とする部分には大きな解析空間を設け境界壁を遠方に設置して計算する必要がある。境界壁の位置と計算値の関係を確認しながら境界壁の位置を決定しなければならない。

以上、ポテンシャル計算で用いる計算法と線路特性計算法および使用における注意点について記した。特に要素の分割数およびその配置は、この手法を用いるうえで、計算精度に大きな影響を与えることを注意しなければならない。

#### 参考文献

- [1]加川他、電気電子有限要素法、科学技術出版社 1986年
- [2]鷺津他、有限要素法ハンドブック、培風館 1985年
- [3]小西、電磁波問題へのアタックの仕方、電子情報通信学会 1983年

## 2. プログラム使用マニュアル

この章では、プログラムシステムの構成と使用方法について記す。プログラムソースファイルについての説明は3章、使用例は4章に示しているので参照してほしい。なお、プログラムソースファイルは、VAX8650の[MUSEN2.CAP.SOURCE]に収納している。

### 2.1 プログラムシステム構成

図2-1にシステム構成のブロック図を示す。3つのプログラムと4つのデータファイルからなる。

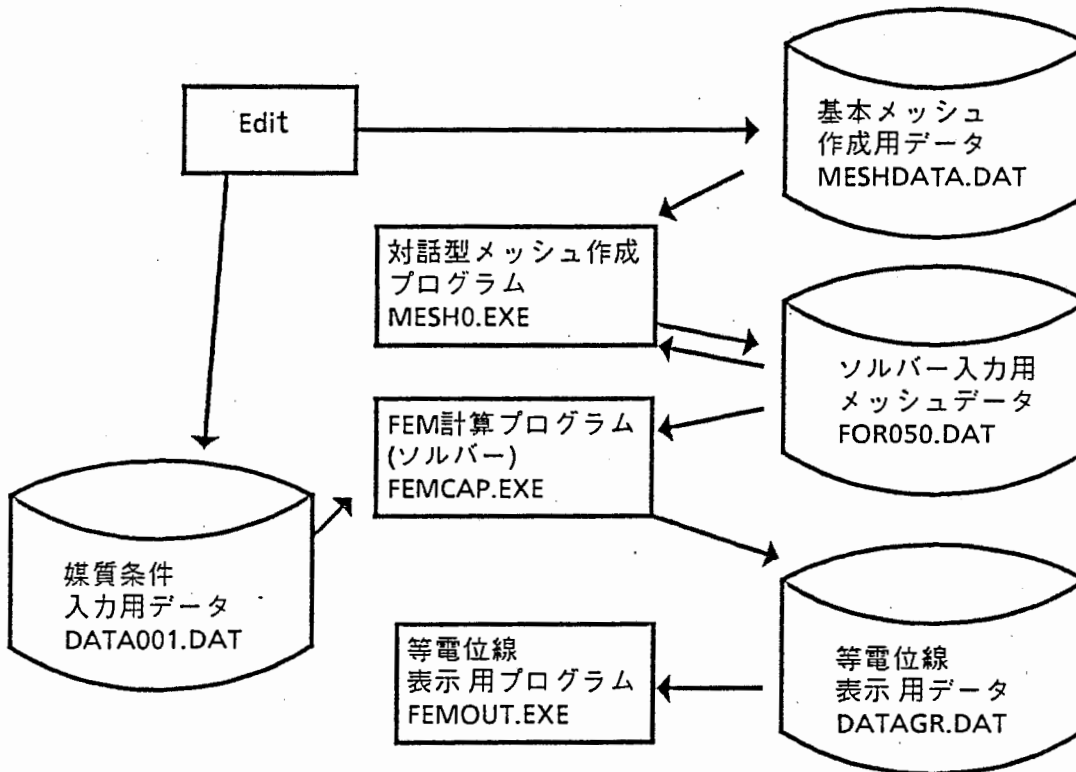


図 2-1 有限要素法による静電場解析  
プログラムシステム構成図

#### 使用プログラム

- (1) 対話型メッシュ作成プログラム  
MESH0.EXE
- (2) FEM計算プログラム  
FEMCAP.EXE
- (3) 等電位線表示用プログラム  
FEMOUT.EXE

入力用メッシュデータを作成するプリプロセッサ  
対話型操作によるデータの編集  
ソルバープログラム

計算結果を用いて等電位線を表示

#### 使用データファイル

- (1) 基本メッシュ作成用データ
- (2) ソルバー入力用メッシュデータ
- (3) 媒質条件入力データ
- (4) 等電位線表示用データ

MESHDATA.DAT  
FOR050.DAT  
DATA001.DAT  
DATAGR.DAT



プログラム(1)・(3)はグラフィック出力としてGKシステムを用いている。現在このサブルーチンパッケージはVAX8650にインストールされている。これらのプログラム起動にあたっては、GKシステムの起動とドライバーの設定を前もって行わなければならない。またデータファイルはASCII形式で記述されており、その書式については3章に記述する。

作業の流れは、まず基本メッシュ作成用データ(MESHDATA.DAT)をEditを用いて作成し、ついで対話型メッシュ作成プログラム(MESHO.EXE)を起動してソルバープログラム入力用データ(FOR050.DAT)を作成する。また、媒質の誘電率はEditを用いて媒質条件入力データ(DATA001.DAT)に記述する。次にソルバープログラム(FEMCAP.EXE)を実行するとアース・ホット導体間の静電容量値が計算され、さらに等電位線表示用データ(DATAGR.DAT)が出力される。等電位線表示用プログラム(FEMOUT.EXE)を起動すれば等電位線がグラフィック出力として得られる。

以下、基本メッシュ作成用データの作成手順及び各プログラムの操作手順について記す。

## 2.2 基本メッシュ作成用データの作成手順

このプログラムでは有限要素として三角要素を用いているが、すべての要素に関する情報を入力するのは多くの労力を要するしまた誤りも生じやすい。そのため、このシステムでは解析空間を大まかな記述方式で表現し(簡易表現データ)それに基づき自動的に三角要素を作成する方法を用いている。この簡易表現データはEditを用いてデータファイル(MESHDATA.DAT)に記述する。以下ではこの簡易表現データの作成手順について説明する。

### 作業手順

#### (1) 解析する領域の断面形状・寸法、境界条件を明確にする。

境界条件としてはホット導体、アース導体、磁気壁のほか、ユーザーが任意にポテンシャル値またはフラックス値を指定した境界壁を設定することができる。計算においてはアース導体の電位は0、ホット導体の電位は1に設定され、この時のホット導体における電荷量から静電容量値を求めている。

#### (2) 導波路の断面を、次の条件を満足するように、任意の四角形要素に分割する。

- \* 1つの四角形要素の内部は同一媒質であること。
- \* 四角形要素の各辺は各々単一の境界条件であること。(1つの辺上で2つ以上の境界条件を含まないこと)
- \* 隣り合う四角形要素同士の共通辺はその両端の点(ノード)が一致していること。(1つの辺はその辺をはさむ四角形要素の各々の辺であること)

#### (3) ノードに1番から順番に通し番号を付ける。

#### (4) 座標値原点を任意の位置に設定し、各ノードの座標値を計算する。このプログラムではmmが基準単位になっており、単位系の変換は拡張係数を用いて(入力数値×拡張係数)により可能である。たとえば、 $\mu\text{m}$ 単位を使用する場合は拡張係数を0.001、cm単位を用いるときは拡張係数を10とする。

#### (5) 四角形要素に1番から順番に通し番号を付ける。

#### (6) 辺上の分割点数を決める。このときその辺を含む四角形要素の対向する辺は同じ分割点数とする。(この分割点数に基づき、プログラムは四角形要素内部に三角要素を作成する。)

#### (7) Editを用いて上記手順により設定した入力数値を以下の書式でファイルに書き込む。

なお、入力数値間は1つ以上のブランクをおくこと。

- [1] 拡張係数を記入。
- [2] ノードの総数を記入。

- [3] ノード番号順に各ノードのX座標値、Y座標値を同一行に記入。(1行に1つのノードのデータを記入する。)
- [4] 四角形要素の総数を記入。
- [5] 四角形要素番号順に下記3行のデータを記述する。
- \* 要素番号、第1辺の分割数、第2辺の分割数を同一行に記入。
  - \* 四角形要素を作る4つのノードの番号を反時計回り方向に順に記入。
  - \* 四角形要素内部の媒質番号、要素の4つの辺の境界条件コードを記入。
- (注:ここで辺の番号は、四角形要素のノード番号記入において第1番目と第2番目のノードにより決まる辺を第1辺とし、反時計回り方向に順に数えるものとする。境界条件コードは、ホット導体を1、アース導体を-1、隣接要素と共有する辺は0とする。なお磁気壁は計算領域最外部に設定でき、境界条件コードは0とする。ユーザーが任意にポテンシャル値またはフラックス値を指定する境界は境界条件コードを2以降順次用いることとし、その設定数およびポテンシャル値またはフラックス値は次行[6],[7]に記入する。なお、媒質の誘電率、および電荷量は別途、媒質条件入力データ(DATA001.DAT)にて定義する。)
- [6] ユーザー設定境界条件の最大値を記入。
- (注:ユーザーが任意にポテンシャル値またはフラックス値を指定する境界を用いない場合は1を記入する。境界条件コード2以降を用いている場合はその最大値を記入する。)
- [7] ユーザー設定境界条件のポテンシャル値、フラックス値を記入。
- (注:境界条件コード2番から順番に、同一行にポテンシャル値、フラックス値を記入する。このときポテンシャル値を指定するときはフラックス値を0に、フラックス値を指定するときはポテンシャル値を0にする。)

以上の作業例は4章に示してある。

### 2.3 対話型メッシュ作成プログラム(MESH0.EXE)の操作手順

対話型メッシュ作成プログラム(MESH0.EXE)は、入力用メッシュデータを作成するプリプロセッサであり、簡易表現データに対するメッシュジェネレート機能と、メッシュのグラフィック表示および編集機能を持つ。なおバックアップ用に随時バックアップデータファイル(MESH.D)を出力しており、編集途中でトラブルがおきた場合はMESHDAT.DATにRENAMEすれば継続できる。メッシュデータに対しては以下の編集機能がある。

- (1) 三角要素の向きの変更
- (2) 要素の消去
- (3) 要素の追加
- (4) ノードの座標値の変更
- (5) 境界条件の書き換え、
- (6) 比誘電率の変更
- (7) 数値データの書き換え

グラフィック表示関係としては以下の機能がある。

- (8) 拡大・縮小表示
- (9) 境界条件を含むグラフィックカラー表示
- (10) 番号表示サイズの変更

#### 操作手順

- (1) GKシステムのLOGINとドライバーの設定をおこなう。
- (2) プログラムMESH0.EXEを起動する。

(3) 画面に下記表示が出る。

```
*** Welcome to Mesh Generation System !! ***
      Sselect Data File !! Input file   FOR050.DAT=1
                                          MESHDATA.DAT=2
      Input No. ??
```

簡易表現データファイルを用いる場合は[1]、既存のデータファイルを用いる場合は[2]を入力する。

(4) 続いて、画面に下記表示が出る。

```
Character size input !! ex. terminal T4014=0.05
```

ノード番号表示の文字の大きさを入力する。数値が大きいほど文字は大きくなる。

(5) 上記操作を行うと、ターミナルに入力メッシュが表示され、ノードの総数[NODE]と要素の総数[LE]が表示される。プログラムはここで一旦ポーズがかかり、リターンキーを叩くと次に進む。(注: ターミナルによっては空白キーの場合があり、このときリターンキーを叩くとエラーが発生することがある。)

(6) 画面に下記表示が出る。

```
Sselect No. !!
Change Diagonal  =1
Move Point to delet=2
Add element      =3
Change Coordinate =4
Change Boundary  =5
Change Element Er=6
Change parameter =7
Zoom             =8
Draw all data    =9
---- Change Character size =10
END =11
Quit =12
```

作業の番号を選択する。ENDは編集データをデータファイルに書き込みプログラムを終了する。QUITは編集データをデータファイルに書き込まないでプログラムを終了する。Change Character sizeを選択すると(4)にもどる。他の作業内容は次に示す。各作業を終了すると(5)にもどる。

[作業1] 三角要素の向きの変更

指定した辺を対角線とする四角形に注目し、その対角線の向きを変更する。

```
Change Form ??
Input Node No. !! 0=Return
```

上記表示がでるので辺の両端のノード番号を入力する。対角線の向きを変更したデータを表示し(5)に戻る。なお0を入力するとデータの変更なしに画面更新して(5)に戻る。例を図2-2に示す。上側を初期の状態とする。ここで辺の両端のノード番号として5、9を入力すると下側の状態となる。

[作業2] 要素の消去

最初に指定したノードを隣接する2番目に指定したノードに重ねて三角要素を消去する。

```
Move Node A to B
No. A & B Input !!
0 = Return
```

上記表示がでるので第1のノードの番号と第2のノードの番号を入力する。要素を消去したデータを表示し(5)に戻る。なお0を入力するとデータの変更なしに画面を更新して(5)に戻る。例を図2-3に示す。上側を初期の状態とする。ここでノード番号として17、16を入力すると図のように点17が点16に移動して要素の消去が行われる。

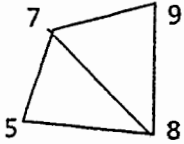
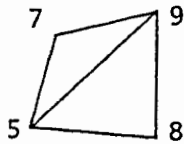


図2-2 三角要素の向きの変更

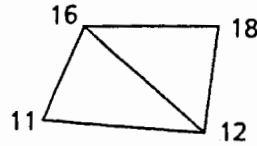
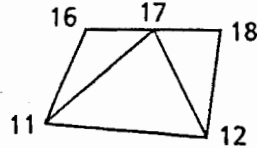


図2-3 要素の消去

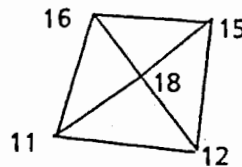
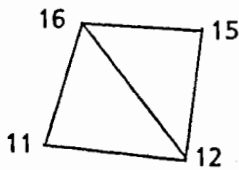


図2-4 要素の追加

**[作業3] 要素の追加**

指定した辺の midpoint に新たにノードを作成し、このノードを用いて要素の追加を行う。

Divided Line  
Input Node No. A & B !!

上記表示がでるので辺の両端のノード番号を入力する。要素を追加したデータを表示して(5)に戻る。なお0を入力するとデータの変更なしに画面を更新して(5)に戻る。図2-4に例を示す。左側を初期の状態とし、辺の両端のノード番号として16、12を入力すると midpoint に新たなノード18を作成して要素の追加を行い、右側の状態となる。

**[作業4] ノードの座標値の変更**

指定したノードの座標値を書き換える。この作業を選択すると次の表示がでる。

Input Node No. !!

ここでノード番号を入力するとその点の座標値が表示される。

No. : 10  
X : 10.0  
Y : 5.0

Input New Coordinate

X,Yの順に座標値を入力すると新しい座標値のデータを表示して(5)に戻る。

**[作業5] 境界条件の書き換え**

指定した辺の境界条件コードを書き換える。この作業を選択すると次の表示がでる。

Boundary Change  
Input Node No. A & B !! 0=Return

辺の両端のノード番号を入力するとその辺の境界条件コードが表示される。

Boundary = 1  
ex. 1 = Conductor(-1)  
-1 = Conductor(-1)  
0 = No setting  
Input New Condition !!

新しい境界条件コードを入力する。この作業はノード番号に0を入力するまで繰り返す。

[作業6] 媒質番号の変更

指定した要素の媒質番号を書き換える。この作業を選択すると次の表示ができる。

Input 3- Node No. !! End=0

媒質番号を変更する要素を構成する3つのノードの番号を入力すると要素の番号とその要素の媒質番号の表示ができる。

Ele = 10 Er = 2  
Input new Er !!

新しい媒質番号の値をいれる。この作業はノード番号に0を入力するまで繰り返す。

[作業7] 数値データの書き換え

グラフィック表示に関係しない数値データ(拡縮係数)を質問してくるので書き換える。さらに、ユーザー設定の境界条件値の設定値に対しても質問してくるので必要あれば書き換える。

[作業8] 拡大・縮小表示

画面に表示する時の縮尺を変更する。この作業を選択すると次の表示ができる。

Zoom Point Input !!  
if 0 = Normal Size

拡大・縮小表示の中心となるノード番号を入れる。(0を入力すると初期設定状態になる)

Magnification ??

上記表示ができるので縮尺を入力する。新しい縮尺でデータを表示し、(5)に戻る。

[作業9] 境界条件を含むグラフィック表示

境界条件・媒質条件をカラー表示する。この作業を選択すると次の表示ができる。

Write Node No. ??  
Yes=0 No=1

[1]を入力するとノード番号は表示されない。

Material draw ?? yes=0 No=1

[1]を入力すると誘電体媒質の区分は表示されない。データを表示し、(5)に戻る。

図2-5にこれらの作業のフローを示す。

## 2.4 媒質条件の設定手順

データファイル(DATA001.DAT)にEDITを用いて、一行に誘電率と電荷を記入する。これを使用媒質数繰り返す。(記述例は4章参照)

## 2.5 FEM計算プログラム(FEMCAP.EXE)の操作手順

特別な操作は必要なく起動するだけでよい。計算結果は次のように画面に表示される。

NODE = 117 ELEMENT = 48BAND = 24  
CAPACITY = 0.0393310227(pF/mm)

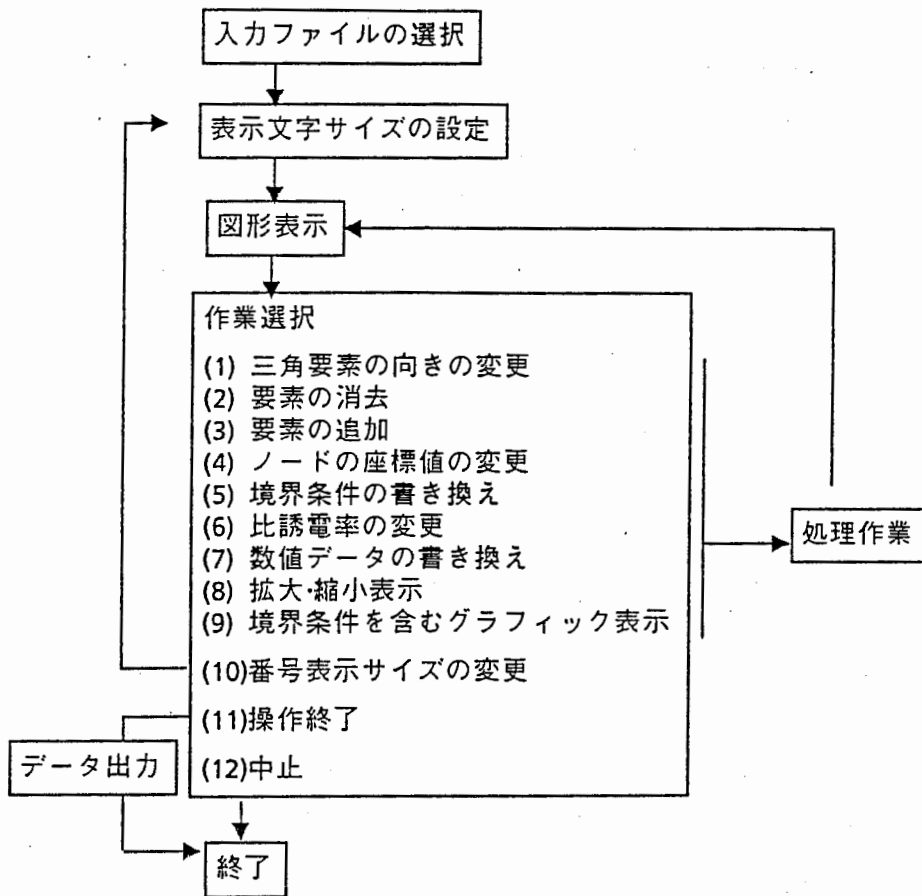


図2-5 作業フロー

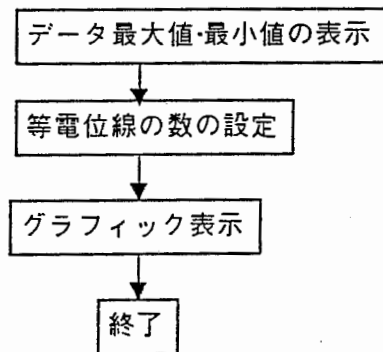


図2-6 作業フロー

ここで、NODEは計算に用いたノード数、ELEMENTは計算に用いた要素数であって、容量値は、単位長さあたりの値が表示される。

## 2.6 等電位線表示用プログラム(FEMOUT.EXE)の操作手順

等電位線表示用プログラム(FEMOUT.EXE)は、ソルバーで計算した電界値を用い、解析モデルにおける等電位線を表示する。指定する分割数ステップ毎に電界値をグラフィック表示する。

操作手順

- (1) GKシステムのLOGINとドライバーの設定をおこなう。
- (2) プログラムFEMOUT.EXEを起動する。
- (3) 画面に下記表示のように計算されたポテンシャル値の最大値・最小値が表示される。

```
Potential min = 0.0    max = 1.0  
How many lines do you want ??
```

ここで、数値を入力するとポテンシャルの最大値・最小値のあいだを、その数で等分した等ポテンシャル線を表示する。

- (8) 上記操作を行うと、ターミナル画面に等ポテンシャル線が表示される。プログラムはここで一旦ポーズがかかり、リターンキーを叩くと終了する。(注: ターミナルによっては空白キーの場合があり、このときリターンキーを叩くとエラーが発生することがある。)

### 3. プログラムリスト

以下では、プログラムで使用しているデータファイルの書式、およびプログラムのフローとリストを示す。

#### 3.1 データファイルの書式

以下に、使用ファイルの書式を列記する。

基本メッシュ作成用データ

MESHDATA.DAT

[拡縮係数]

[ノードの総数]

[ノードのX座標値] [ノードのY座標値]  ノードの数繰り返す

[四角形要素の総数]

[要素番号] [第1辺の分割数] [第2辺の分割数]  
[四角形要素を構成する4つのノードの番号]  要素の数繰り返す  
[媒質番号] [4つの辺の境界条件コード]

特殊境界条件設定数 + 1

[ポテンシャル値] [フラックス値]  設定数繰り返す

ソルバー入力用メッシュデータ

FOR050.DAT

[ノード総数] [要素総数] [境界条件設定数] [ユーザー境界条件設定数] [拡縮係数]

[ノードのX座標値] [ノードのY座標値]  ノードの数繰り返す

[三角形要素を構成する3つのノードの番号]  要素の数繰り返す

[要素番号] [ポテンシャル値] [フラックス値]  境界条件設定数繰り返す

[三角形要素を構成する辺の境界条件番号]  要素の数繰り返す

[ポテンシャル値] [フラックス値]  特殊境界条件設定数 + 1 繰り返す

媒質条件入力データ

DATA001.DAT

[誘電率] [媒質内電荷]  媒質の種類の数繰り返す

等電位線表示用データ

DATAGR.DAT

[ノード総数] [要素総数] [境界条件設定数] [ユーザー境界条件設定数] [拡縮係数]

[ノードのX座標値] [ノードのY座標値]  ノードの数繰り返す

[三角形要素を構成する6つのノードの番号] [媒質番号]  要素の数繰り返す

[ポテンシャル値]  ノードの数繰り返す



### 3.2 対話型メッシュ作成プログラム[MESH0.FOR]

プログラムフローを図3-1に示す。プログラムにおける配列のサイズはPARAMETA文により宣言している。サイズが不足する場合はこの部分を修正すること。

プログラムは複数のサブルーチン群により構成されている。

変数リストはプログラムコード内部に記載している。

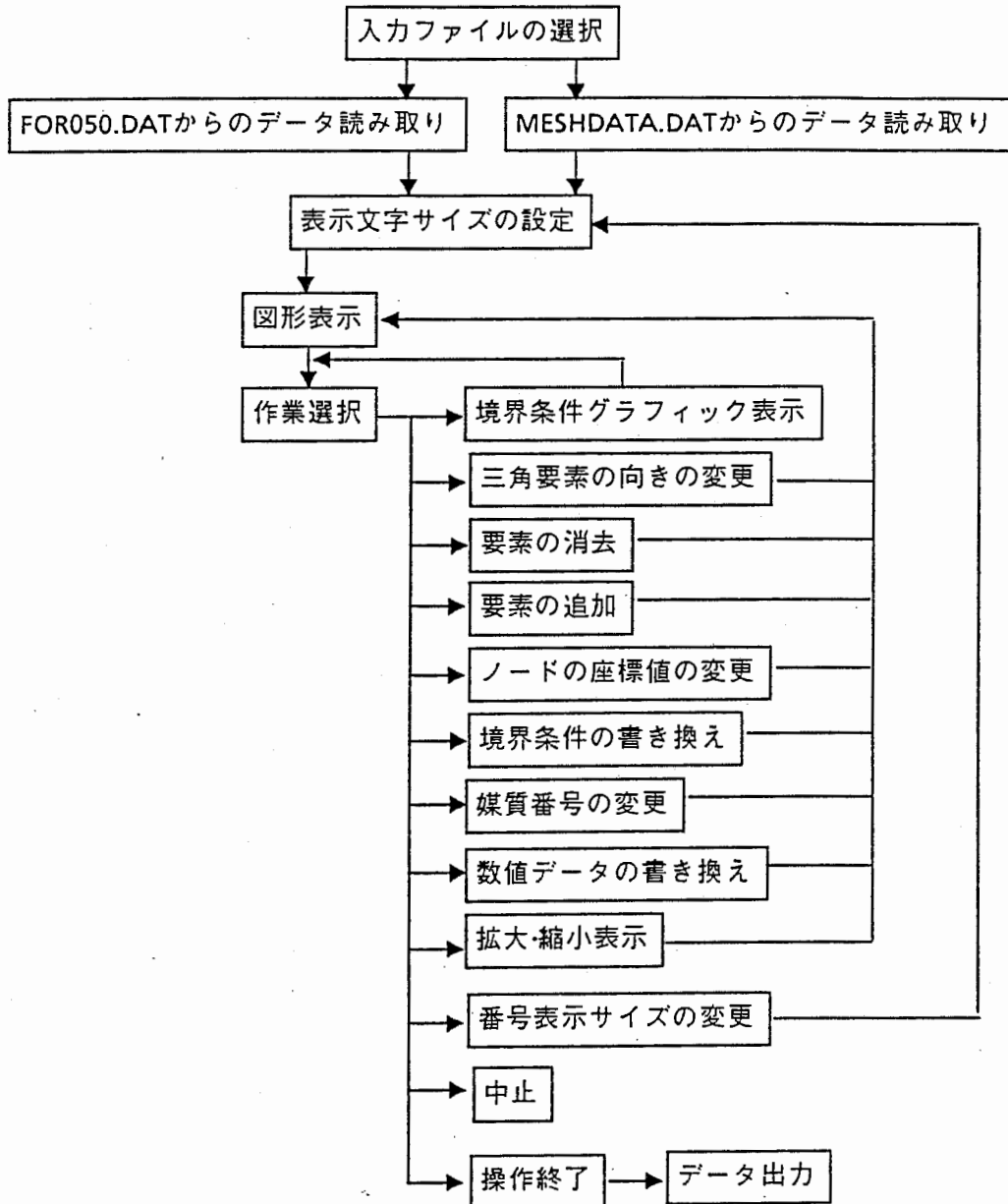


図3-1 プログラムフロー

```

C .....
C                                     作成日 1987.3.角田紀久夫
C                                     最新改訂日 1990.4.角田紀久夫
C .....
C   このプログラムは,2次元静電場 FEM解析に用いる入力データを
C   作成する。
C .....
C   PROGRAM MAIN
C .....
C   LB=160           出力できる分割後の三角要素の最大数
C   LA=120           出力できる分割後のノードの最大数
C .....
C   入力データ
C   SK   拡大縮小係数
C   NP   ノード番号総数           NE   要素番号総数
C   NB   境界条件設定ノード数     MB   バンド幅
C   XP(LA),YP(LA)   ノードのX座標Y座標
C   MESH(LB,7)     配列は(*,1)(* ,2)(* ,3)に要素を構成するノード番号
C                   (*,4)に媒質を区分する番号 (*,5)(* ,6)(* ,7)に境界条件を登録
C   AKYOU(2,*)     境界条件 :(1,*)ポテンシャル (2,*)フラックス
C
C   編集時に使用する変数
C   IPO1,IPO2      辺指定時の入力ノード番号
C   IENO(2),LENO(2) 変更対象要素番号収納
C   INO1(7),INO2(7) 変更対象要素のデータ格納
C   AXP,AYP        辺の分割時の中点座標
C   IEADD,IP12,IPAD 番号調整時のカウンター
C   POIN2          変数入力用メモリー
C   NNE,NONE      要素番号再構成用カウンター
C
C   グラフィック関係
C   IWKID          ワークのレイヤー番号
C   GMINX,GMAXX,GMINY,GMAXY   ウィンドウエリアサイズ
C   CIH            表示文字のサイズ
C   ICL            線の色設定
C   ICLTX         文字の色設定
C   ICR            塗り潰し部分の色設定
C   XXX(4),YYY(4) グラフィック表示用XY座標メモリー
C   IXY(4)         作図時の点のノード番号格納
C   AXX,AYY        表示範囲中心点座標
C   DXX            表示範囲での中心点から端までの距離
C   AXXO,AYYO      表示範囲中心点初期設定座標
C   DXXO,DYYO      表示範囲での中心点から端までの距離の初期設定値
C   XMAXM,XMINM,YMAXM,YMINM  入力座標の最大値最小値
C
C   制御用
C   JKAI          データセイブ用回数カウンター
C   JOK,JJOK      選択用変数(YES,NO判断,作業の選択)
C   I,J,JJ,JJJ,K  制御カウンター
C   IDATA0        入出力データのIO番号
C   IERR          エラーチェック
C
C   ***** メモリーサイズの設定 *****
C   PARAMETER (LB=500,LA=500)
C   ***** 配列宣言 *****
C   DIMENSION XP(LA),YP(LA),MESH(LB,7),
C   #           IENO(2),LENO(2),INO1(7),INO2(7),XXX(4),YYY(4),IXY(4),
C   #           AKYOU(2,6),IB(LA),BB(LA,2)

```

```

C. ***** サイズの表示
      WRITE(6,7)LB,LA
      7 FORMAT(' *** Welcome to Mesh Generation System !!)
C ***** 初期値の設定 *****
      JKAI=0
      NP=0
      NE=0
      IERR=0
      IWKID=1
      ICLTX=7
C ***** データブロックの選択 および データの読み込み
      WRITE(6,1)
      1 FORMAT(' Sselect Data File !! Input file FOR050.DAT=1',/,
# 32X,' MESHDATA.DAT=2',/,15X,' Input No. ?? ',)
      READ(5,*)JOK
C
      IF(JOK.EQ.1) THEN
        CALL MESHE(LA, LB, NP, NE, NB, MB, SK, XP, YP, MESHD, AKYOU, IERR)
      ELSE
        CALL MESHF(LB, LA, NP, NE, NB, MB, SK, XP, YP, MESHD, AKYOU, IERR)
      ENDIF
      IF(IERR.NE.0)GOTO 990
C ***** 表示範囲の設定
      DO I=1, NP
        XMAXM=MAX(XMAXM, XP(I))
        YMAXM=MAX(YMAXM, YP(I))
        XMINM=MIN(XMINM, XP(I))
        YMINM=MIN(YMINM, YP(I))
      ENDDO
      AXX0=(XMAXM+XMINM)/2.0
      AYY0=(YMAXM+YMINM)/2.0
      DXX0=(XMAXM-XMINM)/2.0
      DYY0=(YMAXM-YMINM)/2.0
      DXX0= MAX(DXX0, DYY0)*1.1
C ***** グラフィック用初期データ準備
      30 WRITE(6,3000)
      3000 FORMAT(' Character size input !! ex. tarminal T4014 =0.05')
      READ(5,*)CIHO
      300 CONTINUE
      CIH=CIHO
      AXX=AXX0
      AYY=AYY0
      DXX=DXX0
C
C ***** グラフィックルーチン (作図)
      301 CONTINUE
      GMAXX=AXX+DXX
      GMAXY=AYY+DXX
      GMINX=AXX-DXX
      GMINY=AYY-DXX
      WLINE=1.0
      CALL SETGR(GMINX, GMAXX, GMINY, GMAXY)
      CALL GSPLCI(1) !線の色選択
      CALL LINEGR(LA, LB, NE, WLINE, XP, YP, MESHD) !要素境界線記述
      CALL NUMBGR(LA, NP, CIH, ICLTX, XP, YP) !ノード番号記述
      CALL GPAUSE(IWKID)
C
C[400]***** 作業の選択

```

```

400 CONTINUE
C      -----バックアップ
      JKAI=JKAI+1
      IF(JKAI.EQ.5)THEN
      IO=40
      CALL MESHO(LA, LB, IO, NP, NE, NB, MB, SK, XP, YP, MESHD, AKYOU, IERR,
#      IAT, IB, BB)
      JKAI=0
      ENDIF
C
      WRITE(6,3104)NP,NE
3104  FORMAT(' NODE',/,2X,I3,/, ' ELE',/,2X,I3)
      WRITE(6,4001)
4001  FORMAT(/, ' Sselect No. !! ',/,
# ' Change Diagonal =1',/, ' Move Point to delete=2',/,
# ' Add Element =3',/, ' Change Coordinate =4',/,
# ' Change Boundary =5',/, ' Change Material NO. =6 ',/,
# ' Change parameter =7',/, ' Zoom =8',/,
# ' Draw all data =9',/, ' -- Change Character size =10',/,
# ' END=11 ',/, ' QUIT=12')
      READ(5,*)JOK
      IF(JOK.EQ.10) CALL OFFGR
      GOTO (450,500,700,600,650,890,860,550,750,30,800,850) JOK
C
C[450]***** 対角線の向きの変更
450 CONTINUE
      CALL DIAGGR(LA, LB, NE, IERR, MESHD)
      IF(IERR.EQ.400) GOTO 400
      IF(IERR.EQ.-1) GOTO 950
      CALL OFFGR
      GOTO 301
C
C[500]***** ノードの重ね合わせ
500 CONTINUE
      CALL DELGR(LA, LB, NP, NE, IERR, MESHD, XP, YP)
      IF(IERR.EQ.-1) GOTO 950
      CALL OFFGR
      GOTO 301
C
C[550]***** 図面拡大表示
550 CONTINUE
      WRITE(6,5501)
5501  FORMAT(' Zoom Point Input !!',/, ' if 0 = Normal Size ')
      READ(5,*)IPO1
      IF(IPO1.EQ.0)GOTO 551
      WRITE(6,5502)
5502  FORMAT(' Magnification ??')
      READ(5,*)POIN2
      AXX=XP(IPO1)
      AYY=YP(IPO1)
      DXX=DXX/POIN2
      CIH=CIH/POIN2
551 CONTINUE
      CALL OFFGR
      IF(IPO1.EQ.0)GOTO 300
      GOTO 301
C
C[600]***** 座標書き換え

```

```

600 WRITE(6,*) ' Input Node No. !! '
    READ(5,*)IPO1
    IF(IPO1.EQ.0)GOTO 601
    WRITE(6,6002)IPO1,XP(IPO1),YP(IPO1)
6002 FORMAT(' NO.:',I3,/, ' X :',F7.3,/, ' Y :',F7.3,/,
# ' Input New Cordination ',/, ' X - Y Cordination ?? ')
    READ(5,*)XP(IPO1),YP(IPO1)
601 CONTINUE
    CALL OFFGR
    GOTO 301

C
C[650]*****境界条件書き換え
650 CONTINUE
    CALL BOUNGR(LA,LB,NE,IERR,MESHD)
    CALL OFFGR
    GOTO 301

C
C[700]*****要素分割の追加
700 CONTINUE
    CALL ADDGR(LA,LB,NP,NE,IERR,MESHD,XP,YP)
    CALL OFFGR
    GOTO 301

C
C[750]*****グラフフルアウト
750 CONTINUE
    WRITE(6,7550)
7550 FORMAT(' Write Node number ??',/, ' yes=0, NO = 1 ')
    READ(5,*)JOK
    WRITE(6,7551)
7551 FORMAT(' Material draw ?? yes=0 NO=1')
    READ(5,*)JJOK

C
    CALL OFFGR
    CALL SETGR(GMINX,GMAXX,GMINY,GMAXY)
    WLINE=2.0
    CALL GSPLCI(1)
    CALL LINEGR(LA,LB,NE,WLINE,XP,YP,MESHD)
    IF(JOK.NE.1) CALL NUMBGR(LA,NP,CIH,ICLTX,XP,YP)
    IF(JJOK.NE.1) CALL ELEMGR(LA,LB,NE,MESHD,XP,YP)
    CALL ELEBGR(LA,LB,NE,MESHD,XP,YP)
    WRITE(6,3104)NP,NE
    CALL GSLWSC(1.0)
    CALL GSPLCI(1)
    CALL GPAUSE(IWKID)
    GOTO 400
!線幅選択
!線の色選択
!境界線の表示
!ノード番号表示
!要素の塗り分け
!境界条件

C
C *****[850] QUIT
850 CALL OFFGR
    STOP

C
C *****[860] other data
860 CONTINUE
    CALL TABLGR(SK,MB,AKYOU)
    GOTO 400

C
C[890] *****誘電率を変更する。
890 CALL MATNO(LA,LB,NP,NE,IERR,MESHD)
    IF(IERR.EQ.400) GOTO 400

```

```

        IF(IERR.EQ.-1) GOTO 950
        CALL OFFGR
        GOTO 301
C
C[800] *****計算されたデータをディスクに書き込む
      800 CONTINUE
        IO=50
        CALL MESHO(LA, LB, IO, NP, NE, NB, MB, SK, XP, YP, MESHD, AKYOU, IERR,
          # IAT, IB, BB)
C
      5654 CONTINUE
        STOP
C *****
C *** Error section
C *****
C ***** グラフィック時エラー
C
      950 WRITE(6,9500)
      9500 FORMAT(' INPUT MISS TRY AGEIN !! ')
        GOTO 400
C ***** データ meshdata.dat読み込み時エラー
      990 CONTINUE
        STOP
        END
C
C=====
C データ入出力 関係サブルーチン
C=====
      SUBROUTINE MESHE(LA, LB, NP, NE, NB, MB, SK, XP, YP, MESHD, AKYOU, IERR)
C ディスクからデータを読み取る
C ***** 配列宣言 *****
      DIMENSION XP(LA), YP(LA), MESHD(LB, 7), AKYOU(2, 6)
C
      IDATA0=11
      OPEN(IDATA0, FILE='FOR050.DAT', STATUS='OLD')
      READ(IDATA0, *) NP, NE, NB, MB, SK
      READ(IDATA0, *) (XP(J), J=1, NP)
      READ(IDATA0, *) (YP(J), J=1, NP)
      DO I=1, NE
        READ(IDATA0, *) (MESHD(I, J), J=1, 4)
      ENDDO
      DO I=1, NB
        READ(IDATA0, *) AA, BB, CC
      ENDDO
      READ(IDATA0, *) (MESHD(J, 5), MESHD(J, 6), MESHD(J, 7), J=1, NE)
      DO I=1, MB
        READ(IDATA0, *) (AKYOU(J, I), J=1, 2)
      ENDDO
      CLOSE(IDATA0)
      RETURN
      END
C=====
      SUBROUTINE MESHF(LB, LA, NP, NE, NB, MB, SK, XP, YP, MESHD, AKYOU, IERR)
C メッシュ簡易作成プログラム
C
C [変数リスト]
C INOMA=50 入力するノードの最大数
C IBLMA=60 入力する四角形ブロックの最大数

```

```

C      IBUN=10          入力要素の各辺の最大分割数
C      LB=(メインから) 出力できる分割後の三角要素の最大数
C      LA=(メインから) 出力できる分割後のノードの最大数
C      *****
C      入力データ
C      SK              拡大縮小係数
C      NNODE          入力ノード数
C      XX(INOMA)      入力ノードデータのX座標
C      YY(INOMA)      入力ノードデータのY座標
C      IBOUN1,IBOUN2,IBOUN3,IBOUN4  入力要素の辺の境界条件
C      NBLOK          入力要素数
C      NX,NY          入力要素の辺の分割数
C      AX(4),AY(4)    入力要素の4つのノードのX座標Y座標
C      X(IBUN,IBUN)   入力要素内での分割後のX座標
C      Y(IBUN,IBUN)   入力要素内での分割後のY座標
C      XP(IPONA)      出力ノードデータのX座標
C      YP(LA)         出力ノードデータのY座標
C      NOIN(INOMA)    入力ノードにおけるノード番号
C      ID(IBUN,IBUN)  入力要素内での分割後ノードの位置識別用メモリー
C      YED            入力要素の誘電率
C      NP             分割後ノード番号
C      NE             分割後要素番号
C      NUMBL         入力データの読み込み要素番号
C      XX1,YY1,XX2,YY2  計算時の内部メモリー
C      LINE(4,2)     共通辺の識別用メモリー
C      LNODE(IBLMA,4,IBUN)  辺上のノード番号登録用メモリー
C      MESHD(LB,7)   分割後の要素のノード誘電率境界条件の登録
C      NODE(5,IBLMA)  入力要素のノード番号格納
C      I,J,JK,K,N    制御用カウンター
C      IDATA0        入出力データのIO番号
C      IERR          エラーチェック
C
C      ***** 最大値の設定 *****
C      PARAMETER (INOMA=50,IBLMA=60,IBUN=10)
C      *****
C      DIMENSION X(IBUN,IBUN),Y(IBUN,IBUN),ID(IBUN,IBUN),
C      #AX(4),AY(4),XX(INOMA),YY(INOMA),XP(LA), YP(LA),
C      #LINE(4,2),LNODE(IBLMA,4,IBUN),AKYOU(2,6),
C      #NOIN(INOMA),MESHD(LB,7), NODE(5,IBLMA)
C      ***** 初期値の設定 *****
C      NP=0
C      NE=0
C      IERR=0
C[100]***** ブロックデータの読み込み
C      IDATA0=11
C      OPEN(IDATA0,FILE='MESHDATA.DAT',STATUS='OLD')
C      ***** ブロックの節点座標を読み込む
C      READ(IDATA0,*)SK
C      READ(IDATA0,*)NNODE          ! NNODE --- ブロックの節点数
C      IF(NNODE.LE.0)GOTO 900
C      IF(NNODE.GT.INOMA)GOTO 901
C      DO I=1,NNODE
C      NOIN(I)=0
C      ENDDO
C      DO I=1,NNODE
C      READ(IDATA0,*)XX(I),YY(I)    ! XX(I),YY(I) --- 節点座標
C      ENDDO
C      ***** ブロックの構成を読み込む

```

```

READ(IDATA0,*)NBLOK
DO 199 JJ=1,NBLOK
  READ(IDATA0,*,END=900)NUMBL,NX,NY
    IF(NUMBL.NE.JJ)GOTO 900
    IF(NUMBL.GT.NBLOK)GOTO 900
  READ(IDATA0,*)(NODE(J,JJ),J=1,4)
  NODE(5,JJ)=NODE(1,JJ)
C ----- ブロックの頂点座標を設定する
  DO J=1,4
    AX(J)=XX(NODE(J,JJ)) ! NODE(J,I) I 番目のブロックでJ番目の辺
    AY(J)=YY(NODE(J,JJ)) ! のノード番号
  ENDDO
C ----- ブロックの誘電率を読み込む
C ***** 境界条件を読み込みデータを作成する
  READ(IDATA0,*)YED,IBOUN1,IBOUN2,IBOUN3,IBOUN4
C ----- ブロック内分割後節点の座標を計算
  XX1=(AX(2)-AX(1))/FLOAT(NX)
  YY1=(AY(2)-AY(1))/FLOAT(NX)
  XX2=(AX(3)-AX(4))/FLOAT(NX)
  YY2=(AY(3)-AY(4))/FLOAT(NX)
  X(1,1)=AX(1)
  Y(1,1)=AY(1)
  X(1,NY+1)=AX(4)
  Y(1,NY+1)=AY(4)
  DO I=1,NX ! 第1辺と第3辺の分割後における
    X(I+1,1)=X(I,1)+XX1 ! 節点座標を求める
    X(I+1,NY+1)=X(I,NY+1)+XX2
    Y(I+1,1)=Y(I,1)+YY1
    Y(I+1,NY+1)=Y(I,NY+1)+YY2
  ENDDO
  DO 101 I=1,NX+1 ! 全節点における座標を計算する
    XX1=(X(I,NY+1)-X(I,1))/FLOAT(NY) ! 結果は配列 X(I,J)と Y(I,J)
    YY1=(Y(I,NY+1)-Y(I,1))/FLOAT(NY)
    DO J=2,NY
      X(I,J)=X(I,J-1)+XX1
      Y(I,J)=Y(I,J-1)+YY1
    ENDDO
101 CONTINUE
C ----- すでに入力されたブロックと重なる辺を判定する
  DO 102 J=1,2
    DO 102 I=1,4
102 LINE(I,J)=0
  DO 103 J=1,4
    DO 103 I=1,JJ-1
      DO 103 K=4,1,-1
        IF((NODE(J,JJ).NE.NODE(K+1,I)).OR.
          * (NODE(J+1,JJ).NE.NODE(K,I)))GOTO 103
        LINE(J,1)=I
        LINE(J,2)=K ! J番目の辺がIブロックのK辺と重なる
103 CONTINUE
C ----- 重なる辺を考慮して各ノードにナンバ付けをし、座標を取る。
  DO J=1,NY+1
    DO I=1,NX+1
      ID(I,J)=0
    ENDDO
  ENDDO
  IF(LINE(1,1).NE.0)THEN
    DO I=1,NX+1

```



```

        ID(I,1)=LNODE(LINE(1,1),LINE(1,2),NX+2-I)
    ENDDO
ENDIF
IF(LINE(2,1).NE.0)THEN
    DO J=1,NY+1
        ID(NX+1,J)=LNODE(LINE(2,1),LINE(2,2),NY+2-J)
    ENDDO
ENDIF
IF(LINE(3,1).NE.0)THEN
    DO I=NX+1,1,-1
        ID(I,NY+1)=LNODE(LINE(3,1),LINE(3,2),I)
    ENDDO
ENDIF
IF(LINE(4,1).NE.0)THEN
    DO J=NY+1,1,-1
        ID(1,J)=LNODE(LINE(4,1),LINE(4,2),J)
    ENDDO
ENDIF
IF(NOIN(NODE(1,JJ)).NE.0)ID(1,1)=NOIN(NODE(1,JJ))
IF(NOIN(NODE(2,JJ)).NE.0)ID(NX+1,1)=NOIN(NODE(2,JJ))
IF(NOIN(NODE(3,JJ)).NE.0)ID(NX+1,NY+1)=NOIN(NODE(3,JJ))
IF(NOIN(NODE(4,JJ)).NE.0)ID(1,NY+1)=NOIN(NODE(4,JJ))
DO 120 J=1,NY+1
DO 120 I=1,NX+1
    IF(ID(I,J).NE.0)GOTO 120
    NP=NP+1
    IF(NP.GT.LA)GOTO 903
    ID(I,J)=NP
    XP(NP)=X(I,J)
    YP(NP)=Y(I,J)
120 CONTINUE
C ----- ブロックの各辺上にある節点番号を登録する
DO I=1,NX+1
    LNODE(JJ,1,I)=ID(I,1)
    LNODE(JJ,3,I)=ID(NX+2-I,NY+1)
ENDDO
DO I=1,NY+1
    LNODE(JJ,2,I)=ID(NX+1,I)
    LNODE(JJ,4,I)=ID(1,NY+2-I)
ENDDO
NOIN(NODE(1,JJ))=ID(1,1)
NOIN(NODE(2,JJ))=ID(NX+1,1)
NOIN(NODE(3,JJ))=ID(NX+1,NY+1)
NOIN(NODE(4,JJ))=ID(1,NY+1)
C[130]----- 三角要素の頂点番号列を作成する
DO 130 J=1,NY
DO 130 I=1,NX
    NE=NE+1
    MESHD(NE,1)=ID(I,J)
    MESHD(NE,2)=ID(I+1,J)
    MESHD(NE,3)=ID(I,J+1)
    MESHD(NE,4)=YED
    MESHD(NE,5)=0.0
    IF(J.EQ.1)MESHD(NE,5)=IBOUN1
    MESHD(NE,6)=0.0
    MESHD(NE,7)=0.0
    IF(I.EQ.1)MESHD(NE,7)=IBOUN4
    NE=NE+1
! MESHD(1)~MESHD(3) ノード番号
! MESHD(4) 媒質番号

```

```

MESH(D,1)=ID(I+1,J+1)
MESH(D,2)=ID(I,J+1)
MESH(D,3)=ID(I+1,J)
MESH(D,4)=YED
MESH(D,5)=0.0
IF(J.EQ.NY)MESH(D,5)=IBOUN3
MESH(D,6)=0.0
MESH(D,7)=0.0
IF(I.EQ.NX)MESH(D,7)=IBOUN2
1300 FORMAT(2X,3(I4,2X))
130 CONTINUE
199 CONTINUE
C
C ***** 境界条件の設定値読み込み
      READ(IDATA0,*)MB
      AKYOU(1,1)=1.0
      AKYOU(2,1)=0.0
      IF(MB.GE.2)THEN
      DO I=2,MB
        READ(IDATA0,*)AKYOU(1,I),AKYOU(2,I)
      ENDDO
      ENDIF
      CLOSE(IDATA0)
      RETURN
C ***** データ meshdata.dat 読み込み時エラー
900 WRITE(6,9001)
9001 FORMAT(' Error !! Data is nothing or incorrect',
* ' !! confirm [MESHDATA.DAT] !!')
      CLOSE(IDATA0)
      RETURN
901 WRITE(6,9002)INOMA,N
9002 FORMAT(' Error !! Main Node Over !! max=[' ,I3,
#'] you input [' ,I4,'] !! ')
      GOTO 990
902 WRITE(6,9003)IBLMA,NUMBL
9003 FORMAT(' Error !! Main Block Over !! max=[' ,I3,
#'] you input [' ,I4,'] !! ')
      GOTO 990
903 WRITE(6,9004)LA
9004 FORMAT(' Error !! Divide node Over !! max=[' ,I3,'] !! ')
      GOTO 990
904 WRITE(6,9005)LB
9005 FORMAT(' Error !! Divide element Over !! max=[' ,I3,'] !! ')
990 CLOSE(IDATA0)
      RETURN
      END
C=====
      SUBROUTINE MESH(D,LA,LB,IO,NP,NE,NB,MB,SK,XP,YP,MESH(D),AKYOU,IERR,
# IAT,IB,BB)
C
C      データをディスクに出力する。
C
C ***** 配列宣言 *****
      DIMENSION XP(LA),YP(LA),MESH(D,7),AKYOU(2,6),IAT(LA),
# IB(LA),BB(LA,2),IAA(4)
C      境界条件ノードの検出及びその値の設定
      DO I=1,NP
        IB(I)=0.0

```

```

      IAT(I)=0.0
      ENDDO
DO I=1,NE
DO J=5,7
  IF(MESHD(I,J).NE.0) THEN
    IAA(1)=MESHD(I,1)
    IAA(2)=MESHD(I,2)
    IAA(3)=MESHD(I,3)
    IAA(4)=IAA(1)
    IAT(IAA(J-4))=MESHD(I,J)
    IAT(IAA(J-3))=MESHD(I,J)
  ENDIF
ENDDO
ENDDO
NB=0
DO I=1,NP
  IF( IAT(I) .EQ. 0 ) GOTO 10
  NB=NB+1
  IB(NB)=I
  IF(IAT(I).EQ. -1) THEN
    BB(NB,1)= 0.0
    BB(NB,2)= 0.0
  ELSE
    BB(NB,1)= AKYOU(1,IAT(I))
    BB(NB,2)= AKYOU(2,IAT(I))
  ENDIF
10  CONTINUE
ENDDO
C 出力先ファイル名の設定
  IF( IO.EQ.50) THEN
    OPEN(IO,FILE='FOR050.DAT',STATUS='UNKNOWN')
  ELSE
    OPEN(IO,FILE='MESHD.DAT;1',STATUS='UNKNOWN')
  ENDIF
C データの書き込み
WRITE(IO,*)NP,NE,NB,MB,SK
WRITE(IO,*)(XP(J),J=1,NP)
WRITE(IO,*)(YP(J),J=1,NP)
DO I=1,NE
  WRITE(IO,*)(MESHD(I,J),J=1,4)
ENDDO
DO I=1,NB
  WRITE(IO,*)IB(I),BB(I,1),BB(I,2)
ENDDO
WRITE(IO,*)(MESHD(J,5),MESHD(J,6),MESHD(J,7),J=1,NE)
DO I=1,MB
  WRITE(IO,*)(AKYOU(J,I),J=1,2)
ENDDO
C
CLOSE(IO)
RETURN
END
C=====
C グラフィック 関係サブルーチン
C=====
SUBROUTINE SETGR(GMINX,GMAXX,GMINY,GMAXY)
C GKシステムを起動,グラフィック画面およびパラメータの初期化
C ワールド座標系として(GMINX,GMINY),(GMAXX,GMAXY)を設定

```

C

```
IERR=6
ICONID=0
IWKID=1
ITRAN=1
CALL GOPKS(IERR)
CALL GQEWK(1,IERRIN,NUMBER,ITYPE)
CALL GOPWK(IWKID,ICONID,ITYPE)
CALL GACWK(IWKID)
CALL GQMDS(ITYPE,IERRIN,DCUNIT,RX,RY,LX,LY)
CALL GSWN(ITRAN,GMINX,GMAXX,GMINY,GMAXY)
CALL GSVF(ITRAN,0.0,1.0,0.0,1.0)
CALL GSELNT(ITRAN)
CALL GSWKWN(IWKID,0.0,1.0,0.0,1.0)
RX2=RX*0.95
RX1=RX*0.95-MIN(RX,RY)*0.9
RY1=0.0
RY2=MIN(RX,RY)*0.9
CALL GSWKVP(IWKID,RX1,RX2,RY1,RY2)
CALL GUWK(IWKID,1)
RETURN
END
```

C=====

SUBROUTINE OFFGR

C グラフィック画面の消去およびGKシステム終了

C

```
IWKID=1
CALL GDAWK(IWKID)
CALL GCLWK(IWKID)
CALL GCLKS
RETURN
END
```

C=====

SUBROUTINE LINEGR(LA,LB,NE,WLINE,XP,YP,MESHD)

C 三角形メッシュを作図する。

C

```
DIMENSION XP(LA),YP(LA),MESHD(LB,7),XXX(4),YYY(4)
```

C

```
CALL GSLWSC(WLINE)
DO I=1,NE
  DO J=1,3
    XXX(J)=XP(MESHD(I,J))
    YYY(J)=YP(MESHD(I,J))
  ENDDO
  XXX(4)=XXX(1)
  YYY(4)=YYY(1)
CALL GPL(4,XXX,YYY)
ENDDO
RETURN
END
```

C=====

SUBROUTINE NUMBGR(LA,NP,CIH,ICLTX,XP,YP)

C ノード番号を表示する

C

```
DIMENSION XP(LA),YP(LA)
CHARACTER*3 CHARA3
CHARACTER*2 CHARA2
CHARACTER*1 CHARA1
```

```

101 FORMAT(I1)
102 FORMAT(I2)
103 FORMAT(I3)
C
  CALL GSTXCI(ICLTX)
  CALL GSCHH(CIH)
  DO 10 I=1,9
    IF( I .GT. NP ) GOTO 20
    WRITE(CHARA1,101)I
10   CALL GTX(XP(I),YP(I),CHARA1)
  DO 11 I=10,99
    IF( I .GT. NP ) GOTO 20
    WRITE(CHARA2,102)I
11   CALL GTX(XP(I),YP(I),CHARA2)
  DO 12 I=100,999
    IF( I .GT. NP ) GOTO 20
    WRITE(CHARA3,103)I
12   CALL GTX(XP(I),YP(I),CHARA3)
C
20 CONTINUE
  RETURN
  END
C=====
SUBROUTINE ELEMGR(LA, LB, NE, MESHD, XP, YP)
C   要素の媒質種類ごとに色分けする。
C
  DIMENSION XP(LA), YP(LA), MESHD(LB, 7), XXX(4), YYY(4)
C
  CALL GSLWSC(1.0)
  CALL GSFAS(3)
  CALL GSFASI(24)
  DO 10 I=1, NE
    ICR=MESHD(I, 4)+1
    CALL GSFACI(ICR)
    DO 11 J=1, 3
      XXX(J)=XP(MESHD(I, J))
11   YYY(J)=YP(MESHD(I, J))
      XXX(4)=XXX(1)
      YYY(4)=YYY(1)
    CALL GFA(4, XXX, YYY)
10 CONTINUE
  RETURN
  END
C=====
SUBROUTINE ELEBGR(LA, LB, NE, MESHD, XP, YP)
C   境界条件ごとに色を塗り分ける
C
  DIMENSION XP(LA), YP(LA), MESHD(LB, 7), XXX(2), YYY(2), IXY(4)
C
  CALL GSLWSC(3.0)
  DO I=1, NE
    DO J=1, 3
      IF(MESHD(I, J+4).NE.0) THEN
        ICL=MESHD(I, J+4)
        IF(ICL.EQ.-1)ICL=7
        IF(ICL.EQ. 1)ICL=4
        CALL GSPLCI(ICL)
        IXY(1)=MESHD(I, J)

```

```

        IXY(2)=MESHD(I,J+1)
        IF(J.EQ.3)IXY(2)=MESHD(I,1)
        DO K=1,2
            XXX(K)=XP(IXY(K))
            YYY(K)=YP(IXY(K))
        ENDDO
        CALL GPL(2,XXX,YYY)
    ENDF
ENDDO
ENDDO
RETURN
END
C=====
C 要素データ操作関係サブルーチン
C=====
        SUBROUTINE DIAGGR(LA,LB,NE,IERR,MESHD)
C      対角線の向きを変更する。
C
        DIMENSION MESHD(LB,7),IENO(2),ILNO(2),INO1(7),INO2(7)
C
        IERR=0
        WRITE(6,101)
101  FORMAT('  Change Form ?? ',/, ' Input Node No. !!  0=Return')
        READ(5,*)IPO1
        IF(IPO1.EQ.0)GOTO 400
        READ(5,*)IPO2
        JJ=0
        DO 10 I=1,NE
            IL=0
            IF((MESHD(I,1).EQ.IPO1).AND.(MESHD(I,2).EQ.IPO2))  IL=1
            IF((MESHD(I,2).EQ.IPO1).AND.(MESHD(I,1).EQ.IPO2))  IL=1
            IF((MESHD(I,2).EQ.IPO1).AND.(MESHD(I,3).EQ.IPO2))  IL=2
            IF((MESHD(I,3).EQ.IPO1).AND.(MESHD(I,2).EQ.IPO2))  IL=2
            IF((MESHD(I,3).EQ.IPO1).AND.(MESHD(I,1).EQ.IPO2))  IL=3
            IF((MESHD(I,1).EQ.IPO1).AND.(MESHD(I,3).EQ.IPO2))  IL=3
            IF(IL.NE.0) THEN
                JJ=JJ+1
                IENO(JJ)=I
                ILNO(JJ)=IL
                IF(JJ.EQ.2)GOTO 11
            ENDF
10  CONTINUE
        GOTO 950
C
11  IF(MESHD(IENO(1),4).NE.MESHD(IENO(2),4)) GOTO 960
    DO 20 I=1,2
        IF(ILNO(I).EQ.1) THEN
            INO1(1)=MESHD(IENO(I),3)
            MESHD(IENO(I),3)=MESHD(IENO(I),2)
            MESHD(IENO(I),2)=MESHD(IENO(I),1)
            MESHD(IENO(I),1)=INO1(1)
            INO1(1)=MESHD(IENO(I),7)
            MESHD(IENO(I),7)=MESHD(IENO(I),6)
            MESHD(IENO(I),6)=MESHD(IENO(I),5)
            MESHD(IENO(I),5)=INO1(1)
        ENDF
        IF(ILNO(I).EQ.3) THEN
            INO1(1)=MESHD(IENO(I),1)

```

```

        MESHD(IENO(I),1)=MESHD(IENO(I),2)
        MESHD(IENO(I),2)=MESHD(IENO(I),3)
        MESHD(IENO(I),3)=INO1(1)
        INO1(1)=MESHD(IENO(I),5)
        MESHD(IENO(I),5)=MESHD(IENO(I),6)
        MESHD(IENO(I),6)=MESHD(IENO(I),7)
        MESHD(IENO(I),7)=INO1(1)
    ENDF
20 CONTINUE
    DO 30 I=1,7
        INO1(I)=MESHD(IENO(1),I)
        INO2(I)=MESHD(IENO(2),I)
30 CONTINUE
C
    MESHD(IENO(1),1)=INO1(2)
    MESHD(IENO(1),2)=INO2(1)
    MESHD(IENO(1),3)=INO1(1)
    MESHD(IENO(1),4)=INO1(4)
    MESHD(IENO(1),5)=INO2(7)
    MESHD(IENO(1),6)=INO1(6)
    MESHD(IENO(1),7)=INO1(5)
C
    MESHD(IENO(2),1)=INO2(2)
    MESHD(IENO(2),2)=INO1(1)
    MESHD(IENO(2),3)=INO2(1)
    MESHD(IENO(2),4)=INO2(4)
    MESHD(IENO(2),5)=INO1(7)
    MESHD(IENO(2),6)=INO2(6)
    MESHD(IENO(2),7)=INO2(5)
C
    RETURN
C
400 IERR=400
    RETURN
950 IERR=-1
    RETURN
960 IERR=-1
    WRITE(6,*) ' Material is not same !!'
    RETURN
    END
C=====
    SUBROUTINE DELGR(LA, LB, NP, NE, IERR, MESHD, XP, YP)
C     指定したノードを隣接ノードに重ねる形で要素を消去する
C
    DIMENSION XP(LA), YP(LA), MESHD(LB,7), IENO(2), LENO(2),
#         INO1(7), INO2(7)
C
    IERR=0
    WRITE(6,5001)
5001 FORMAT(' Move Node A to B',/, ' No.A & B Input !! ',/,
# ' 0=Return')
    READ(5,*)IPO1
    IF(IPO1.EQ.0)GOTO 501
    READ(5,*)IPO2
    IENO(1)=0
    IENO(2)=0
    JJ=0
    DO 501 I=1,NE

```

```

IF((MESH(I,1).EQ.IPO1).AND.(MESH(I,2).EQ.IPO2))GOTO 502
IF((MESH(I,2).EQ.IPO1).AND.(MESH(I,3).EQ.IPO2))GOTO 502
IF((MESH(I,3).EQ.IPO1).AND.(MESH(I,1).EQ.IPO2))GOTO 502
IF((MESH(I,3).EQ.IPO1).AND.(MESH(I,2).EQ.IPO2))GOTO 502
IF((MESH(I,2).EQ.IPO1).AND.(MESH(I,1).EQ.IPO2))GOTO 502
IF((MESH(I,1).EQ.IPO1).AND.(MESH(I,3).EQ.IPO2))GOTO 502
IF((I.EQ.NE).AND.(JJ.NE.0))GOTO 503
GOTO 501
502 CONTINUE
JJ=JJ+1
IENO(JJ)=I
IF(JJ.EQ.2)GOTO 503
501 CONTINUE
GOTO 950
503 CONTINUE
DO 504 I=1,NE
DO 504 J=1,3
504 IF(MESH(I,J).EQ.IPO1)MESH(I,J)=IPO2
NNE=0
DO 507 I=1,NE
IF( (I.EQ.IENO(1)).OR.(I.EQ.IENO(2)) )GOTO 507
NNE=NNE+1
DO 508 K=1,7
508 MESH(NNE,K)=MESH(I,K)
507 CONTINUE
NE=NNE
DO 510 I=1,NE
DO 510 J=1,3
510 IF(MESH(I,J).GT.IPO1)MESH(I,J)=MESH(I,J)-1
DO 509 I=IPO1+1,NP
XP(I-1)=XP(I)
509 YP(I-1)=YP(I)
NP=NP-1
RETURN
C
950 IERR=-1
RETURN
END
C=====
SUBROUTINE ADDGR(LA,LB,NP,NE,IERR,MESH,XP,YP)
C 指定した辺を2分割して新たな要素を追加する
C
DIMENSION XP(LA),YP(LA),MESH(LB,7),
# IENO(2),LENO(2),INO1(7),INO2(7)
C
IERR=0
WRITE(6,7024)
7024 FORMAT(' Divided Line ',/, ' Input Node No. A & B !! ',/,
# ' 0=Return')
READ(5,*)IPO1
IF(IPO1.EQ.0)GOTO 400
READ(5,*)IPO2
IP12=MIN(IPO1,IPO2)
AXP=(XP(IPO1)+XP(IPO2))/2.0
AYP=(YP(IPO1)+YP(IPO2))/2.0
JJ=0
DO 7025 I=1,NE
JJJ=0

```



```

IF((MESH(I,1).EQ.IP01).AND.(MESH(I,2).EQ.IP02)) JJJ=1
IF((MESH(I,2).EQ.IP01).AND.(MESH(I,3).EQ.IP02)) JJJ=2
IF((MESH(I,3).EQ.IP01).AND.(MESH(I,1).EQ.IP02)) JJJ=3
IF((MESH(I,3).EQ.IP01).AND.(MESH(I,2).EQ.IP02)) JJJ=2
IF((MESH(I,2).EQ.IP01).AND.(MESH(I,1).EQ.IP02)) JJJ=1
IF((MESH(I,1).EQ.IP01).AND.(MESH(I,3).EQ.IP02)) JJJ=3
IF(JJJ.NE.0)GOTO 7026
IF((I.EQ.NE).AND.(JJ.NE.0))GOTO 7027
GOTO 7025
7026 JJ=JJ+1
IENO(JJ)=I
LENO(JJ)=JJJ
IF(JJ.EQ.2)GOTO 7027
7025 CONTINUE
IF((I.EQ.NE).AND.(JJ.NE.0))GOTO 7027
GOTO 950
C ----- 新しい番号付け
7027 CONTINUE
IPAD=IP12+1
DO I=NP,IPAD,-1
XP(I+1)=XP(I)
YP(I+1)=YP(I)
ENDDO
XP(IPAD)=AXP
YP(IPAD)=AYP
NP=NP+1
C ----- 番号変換
DO 7038 I=1,NE
DO 7138 J=1,3
IF(MESH(I,J).LT.IPAD)GOTO 7138
MESH(I,J)=MESH(I,J)+1
7138 CONTINUE
7038 CONTINUE
C ----- 新しい要素の追加
DO 7061 K=1,JJ
IF(K.EQ.2)IENO(K)=IENO(K)+1
IEADD=IENO(K)+1
DO 7039 I=NE,IEADD,-1
DO 7039 J=1,7
7039 MESH(I+1,J)=MESH(I,J)
NE=NE+1
MESH(IEADD,4)=MESH(IENO(K),4)
GOTO (7051,7052,7053) LENO(K)
7051 MESH(IEADD,1)=IPAD
MESH(IEADD,2)=MESH(IENO(K),2)
MESH(IEADD,3)=MESH(IENO(K),3)
MESH(IEADD,5)=MESH(IENO(K),5)
MESH(IEADD,6)=MESH(IENO(K),6)
MESH(IEADD,7)=0
MESH(IENO(K),1)=MESH(IENO(K),1)
MESH(IENO(K),2)=IPAD
MESH(IENO(K),3)=MESH(IENO(K),3)
MESH(IENO(K),5)=MESH(IENO(K),5)
MESH(IENO(K),6)=0
MESH(IENO(K),7)=MESH(IENO(K),7)
GOTO 7060
7052 MESH(IEADD,1)=IPAD
MESH(IEADD,2)=MESH(IENO(K),3)

```

```

MESH(IEADD,3)=MESH(IENO(K),1)
MESH(IEADD,5)=MESH(IENO(K),6)
MESH(IEADD,6)=MESH(IENO(K),7)
MESH(IEADD,7)=0
MESH(IENO(K),1)=MESH(IENO(K),1)
MESH(IENO(K),2)=MESH(IENO(K),2)
MESH(IENO(K),3)=IPAD
MESH(IENO(K),5)=MESH(IENO(K),5)
MESH(IENO(K),6)=MESH(IENO(K),6)
MESH(IENO(K),7)=0
GOTO 7060
7053 MESH(IEADD,1)=IPAD
MESH(IEADD,2)=MESH(IENO(K),2)
MESH(IEADD,3)=MESH(IENO(K),3)
MESH(IEADD,5)=0
MESH(IEADD,6)=MESH(IENO(K),6)
MESH(IEADD,7)=MESH(IENO(K),7)
MESH(IENO(K),1)=MESH(IENO(K),1)
MESH(IENO(K),2)=MESH(IENO(K),2)
MESH(IENO(K),3)=IPAD
MESH(IENO(K),5)=MESH(IENO(K),5)
MESH(IENO(K),6)=0
MESH(IENO(K),7)=MESH(IENO(K),7)
7060 CONTINUE
7061 CONTINUE
7040 CONTINUE
RETURN
C
400 IERR=400
RETURN
950 IERR=-1
RETURN
END
C-----
SUBROUTINE BOUNGR(LA, LB, NE, IERR, MESH)
C 境界条件値の書き換え
C
DIMENSION MESH(LB, 7)
C
IERR=0
650 WRITE(6, 6501)
6501 FORMAT(' Boundary Change ', /,
#' Input Node No. A & B !! 0=Return ')
READ(5, *) IPO1
IF(IPO1.EQ.0) GOTO 699
READ(5, *) IPO2
JJ=0
DO 6025 I=1, NE
IF((MESH(I, 1).EQ.IPO1).AND.(MESH(I, 2).EQ.IPO2)) GOTO 6026
IF((MESH(I, 2).EQ.IPO1).AND.(MESH(I, 3).EQ.IPO2)) GOTO 6027
IF((MESH(I, 3).EQ.IPO1).AND.(MESH(I, 1).EQ.IPO2)) GOTO 6028
IF((MESH(I, 3).EQ.IPO1).AND.(MESH(I, 2).EQ.IPO2)) GOTO 6027
IF((MESH(I, 2).EQ.IPO1).AND.(MESH(I, 1).EQ.IPO2)) GOTO 6026
IF((MESH(I, 1).EQ.IPO1).AND.(MESH(I, 3).EQ.IPO2)) GOTO 6028
6025 CONTINUE
GOTO 950
6026 IPO2=5
GOTO 6029

```

```

6027 IPO2=6
      GOTO 6029
6028 IPO2=7
6029 WRITE(6,6030)MESH(I,IPO2)
6030 FORMAT(' BOUNDARY= ',I3,/, ' ex. 1=Hot conductor( 1)',/,
# 7X,'-1=Earth conductor(-1)',/,8X,'0=No setting',/,
# 8X,'2=< Other setting No.',/,
# ' Input New Condition !!')
      READ(5,*)MESH(I,IPO2)
      GOTO 650
699 CONTINUE
      RETURN
C
950 IERR=-1
      RETURN
      END
C=====
      SUBROUTINE MATNO(LA,LB,NP,NE,IERR,MESH)
C      指定した要素の媒質番号を変更する
C
      DIMENSION MESH(LB,7),IPO(5)
C
10  WRITE(6,*)' Input 3-node No. END=0'
      READ(5,*)IPO(1),IPO(2),IPO(3)
      IF(IPO(1).EQ.0)GOTO 400
      IPO(4)=IPO(1)
      IPO(5)=IPO(2)
      DO I=1,NE
        DO J=1,3
          IF((MESH(I,1).EQ.IPO(J)).AND.(MESH(I,2).EQ.IPO(J+1)).AND.
# (MESH(I,3).EQ.IPO(J+2)) ) GOTO 20
        ENDDO
      ENDDO
      GOTO 950
20  WRITE(6,100)I,MESH(I,4)
100 FORMAT(' ELE=',I3,' Material No.',I3,/, ' Input new No. ')
      READ(5,*)MESH(I,4)
      GOTO 10
C
400 IERR=400
      RETURN
950 IERR=-1
      RETURN
      END
C=====
      SUBROUTINE TABLGR(SK,MB,AKYOU)
C      計算条件の設定を行う。
C
      DIMENSION AKYOU(2,6)
C 寸法の単位系
      WRITE(6,102)SK
102 FORMAT('Magnification=',F9.6)
      WRITE(6,*) ' Change Magnification ?? yes=1'
      READ(5,*)JOK
      IF(JOK.EQ.1)THEN
        WRITE(6,*) ' Input Magnification'
        READ(5,*)SK
      ENDIF

```

C 境界条件の設定値の変更及び記述

```
WRITE(6,*)' Boundary set=',MB
50 WRITE(6,*)' No. Volt Charge '
DO I=1,MB
  WRITE(6,*) I,AKYOU(1,I),AKYOU(2,I)
ENDDO
WRITE(6,*) ' Change No. input ! 0=end'
WRITE(6,*) ' MB <= 0 '
READ(5,*)JOK
IF(JOK.EQ.0) GOTO 900
IF(JOK.GT.MB) MB=JOK
WRITE(6,*) ' Input Volt & Charge'
READ(5,*)AKYOU(1,JOK),AKYOU(2,JOK)
GOTO 50
900 CONTINUE
RETURN
END
```

### 3.3. FEM計算プログラム[FEMCAP.FOR]

本プログラムでは、メッシュデータを用いて2次三角要素のノードを算出したのち、マトリクスを作成して連立一次方程式を解き、ポテンシャル値および静電容量値を求めている。

プログラムは2つのサブルーチンにより構成されている。

SUBROUTINE SECOND では、2次ノードを作成したのち、バンド幅縮小のためリナンバリング機能を持っている。

SUBROUTINE SOLVE は本プログラムの中心であって、内部ではバンド行列を用いている。また連立一次方程式は掃き出し法を用いて解いている。本システムでは使用していないが各ノードにおける電界値の算出も行っている。

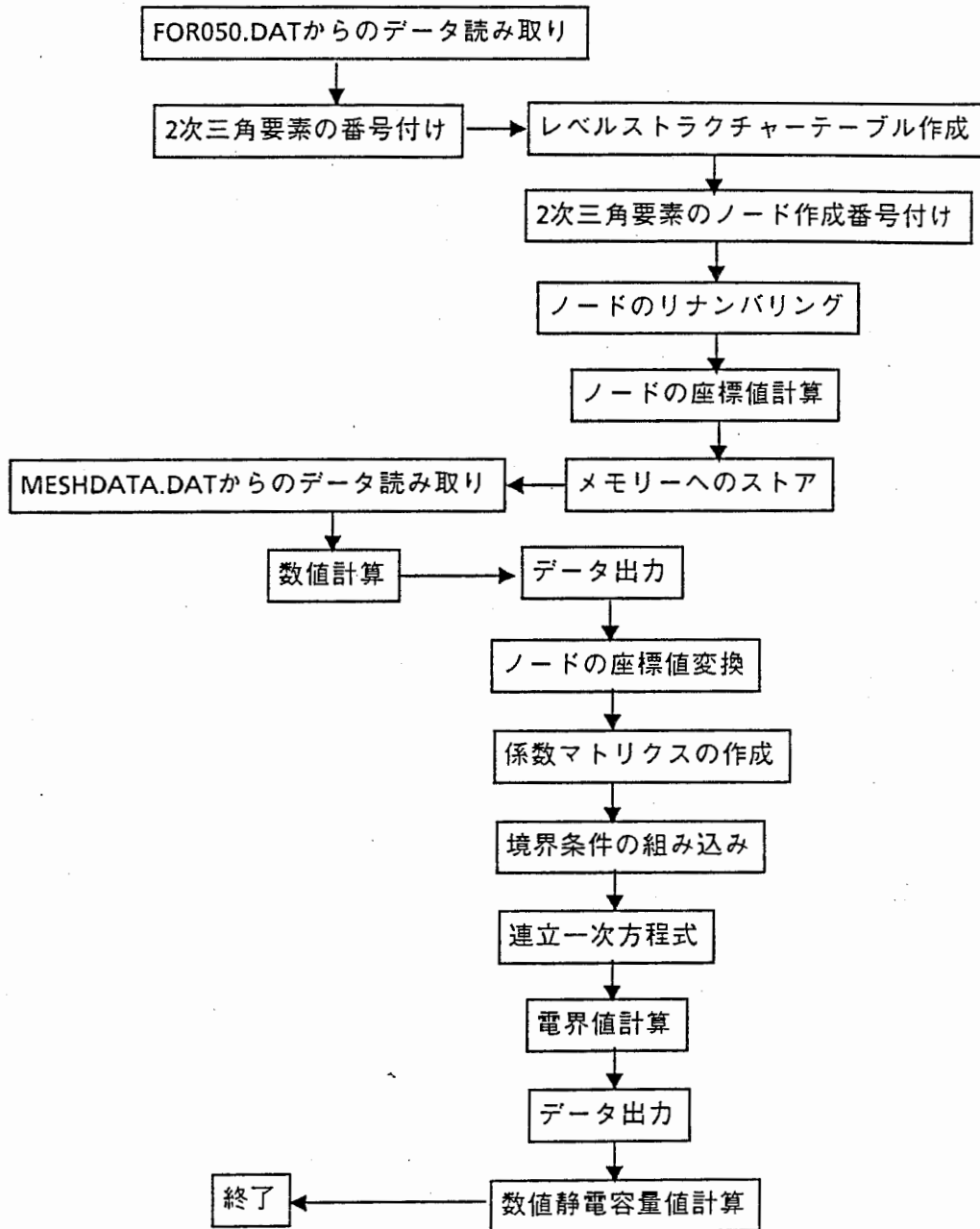


図3-2 プログラムフロー

```

C *****
C                                     作成日 1987.3.角田紀久夫
C                                     最新改訂日 1990.4.角田紀久夫
C *****
C   このプログラムは,2次元静電場問題を FEM を用いて解いている。
C   基本式は ポアソン方程式である。
C
C   入力データに対し,2次三角要素を作成し,指定導体間の静電容量値を
C   算出する。
C *****
C   PROGRAM MAIN
C *****
C   INELE  : 入力最大要素数
C   INNOD  : 入力最大ノード数
C   IMELE  : 2次要素最大要素数
C   IMNOD  : 2次要素最大ノード数
C   INBOU  : 境界条件最大設定数
C   INBAN  : 最大バンド幅設定数
C   IBAI   : 最大設定媒質数
C
C   PARAMETER ( INELE=2000, INNOD=2000, IMELE=4000, IBAI=8,
C   #           IMNOD=4000, INBOU=2000, INBAN=1000 )
C *****
C   NP : ノード数           NE : 要素数
C   NB : 境界条件設定ノード数   MB : バンド幅
C   SK : 拡大縮小係数
C   XX(*),YY(*) : ノードの X,Y座標値
C   IB(*) : 境界条件の設定されたノードの番号を格納
C   BB(*,2) : 設定された境界条件   BB(*,1)=電位 BB(*,2)= $\partial$ 電位/ $\partial n$ 
C   IE(*,7) : 各要素におけるノード番号   IE(*,7) は媒質番号
C   EPSI(2,*) : 各要素内部における電荷量
C
C *****
C   DIMENSION X(INNOD), Y(INNOD), IE(INELE,7),
C   #         IB(INBOU), BB(INBOU,2), EPSI(2,IBAI),
C   #         XX(IMNOD), YY(IMNOD)
C   DIMENSION IM(INNOD+1),JM(INNOD),IG(IMNOD*2),JG(IMNOD*2),
C   #         IW(INNOD),JW(INNOD),KW(INNOD),LW(INNOD)
C   DIMENSION G(IMNOD,INBAN),GV(IMNOD),GCAP(IMNOD,INBAN)
C *****
C   データの読み込み
C *****
C   IDATA0=60
C   OPEN( IDATA0, FILE='FOR050.DAT', STATUS='OLD',ERR=900 )
C   READ( IDATA0, *, END=910, ERR=990 ) NP,NE,NB,MB,SK
C   IF((NP*NE*NB*SK).LE.0) GOTO 950
C   IF((NP.GT.INNOD).or.(NE.GT.INELE)) GOTO 960
C   READ(IDATA0,*)(X(J),J=1,NP)
C   READ(IDATA0,*)(Y(J),J=1,NP)
C   DO 101 I=1,NE
101   READ(IDATA0,*,END=910)IE(I,1),IE(I,2),IE(I,3),IE(I,7)
C   DO 102 I=1,NB
102   READ(IDATA0,*,END=910)IB(I),BB(I,1),BB(I,2)
C   CLOSE(IDATA0)
C *****
C   2次要素番号付け
C *****
C   CALL SECOND(INELE,INNOD,IMELE,IMNOD,INBOU,INBAN,

```

```

#           NP,NE,NB,MB,SK,X,Y,XX,YY,IB,BB,IE,
#           IM,JM,IG,JG,IW,JW,KW,LW)
          IF((NP.GT.IMNOD).OR.(NE.GT.IMELE)) GOTO 970
          IF(MB.GT.INBAN) GOTO 980
C *****
C 媒質設定条件の読み込み
C *****
          OPEN(IDATA0,FILE='DATA001.DAT',STATUS='OLD',ERR=920)
          DO 300 I=1,IBAI
              READ(IDATA0,*,END=301)(EPSI(J,I),J=1,2)
              IF(EPSI(1,I).LE.0)GOTO 301
          300  CONTINUE
          301 IF(I.LE.0) GOTO 930
              CLOSE(IDATA1)
C *****
C 連立一次方程式の計算及び容量値の計算
C *****
          CALL SOLVE(INELE,INNOD,IMELE,IMNOD,INBOU,INBAN,IBAI,EPSI,
#           NP,NE,NB,MB,SK,XX,YY,IB,BB,IE,G,GV,GCAP)
C
          STOP
C *****
C エラーチェック
C *****
          900 WRITE(6,*) ' ERROR !! Data file is not found or open !!'
              STOP
          910 WRITE(6,*) ' ERROR !! End of file is detected during read !!'
              STOP
          920 WRITE(6,*) ' ERROR !! Mateiral data file is not found or open !!'
              STOP
          930 WRITE(6,*) ' ERROR !! Mateiral datas are not writen !!'
              STOP
          950 WRITE(6,*) ' Some datas are zero !! Check data file !!'
              CLOSE(IDATA0)
              STOP
          960 WRITE(6,*) ' Memory over !! Max node = ',INNOD,
#           'Max element=',INELE
              WRITE(6,*) ' Your setting          node = ',NP,
#           '          element=',NE
              CLOSE(IDATA0)
              STOP
          970 WRITE(6,*) ' Memory over !! Max node = ',IMNOD,
#           'Max element=',IMELE
              WRITE(6,*) ' Second order setting node = ',NP,
#           '          element=',NE
              STOP
          980 WRITE(6,*) ' Band over !! Max BW = ',INBAN
              WRITE(6,*) '          setting BW = ',NP
              STOP
          990 WRITE(6,*) ' ERROR !! Data format is not matched !!'
              CLOSE(IDATA0)
              STOP
C
          END
C
C *****
          SUBROUTINE SECOND(INELE,INNOD,IMELE,IMNOD,INBOU,INBAN,
# NP,NE,NB,MB,SK,X,Y,XX,YY,IB,BB,IE,IM,JM,IG,JG,IW,JW,KW,LW)

```

```

C *****
C * このルーチンでは、三角形メッシュ入力データを用いて2次三角要素を
C * 作成する。この時、各ノードは、バンド行列として格納されることを
C * 想定し、リナンバリングを行い、バンド幅を縮小している。
C * プログラムは、まず、1次三角要素を作成した後、2次要素へ変換する。
C *****
      DIMENSION IM(INNOD+1),JM(INNOD),IG(IMNOD*2),JG(IMNOD*2),
#           IW(INNOD),JW(INNOD),KW(INNOD),LW(INNOD),
#           X(INNOD), Y(INNOD), IE(INELE,7),
#           IB(INBOU), BB(INBOU,2), XX(IMNOD), YY(IMNOD),
#           KK(5)
C *****
C level structure tableの作成
C *****
      M=1
      N=0
      IM(1)=0
      DO 200 I=1,NP
        DO 201 J=1,NP
          201      IW(J)=0
          DO 202 J=1,NE
            KK(1)=IE(J,1)
            KK(2)=IE(J,2)
            KK(3)=IE(J,3)
            KK(4)=KK(1)
            KK(5)=KK(2)
            DO 203 K=1,3
              IF(I.EQ.KK(K)) THEN
                IF(IW(KK(K+1)).EQ.0) THEN
                  N=N+1
                  IG(N)=KK(K+1)
                  IW(KK(K+1))=1
                ENDIF
                IF(IW(KK(K+2)).EQ.0) THEN
                  N=N+1
                  IG(N)=KK(K+2)
                  IW(KK(K+2))=1
                ENDIF
              ENDIF
            203      CONTINUE
          202      CONTINUE
          IM(I+1)=N
        200      CONTINUE
C *****
C ノードへの Numbering
C *****
      DO 300 I=1,NP
        IW(I)=0
        JM(I)=0
      300      JW(I)=IM(I+1)-IM(I)
      DO 301 I=1,IM(NP+1)
      301      JG(I)=0
        IW(M)=1
        JM(M)=1
        N=1
        M=1
        NN=1
      DO 302 I=1,NP

```



```

DO 303 J=1,NP
  IF(I.NE.IW(J))GOTO 304
  IF(JW(J).LT.1)GOTO 305
  I1=IM(J)
  I2=IM(J+1)-I1
  DO 306 K=1,I2
    I3=IG(I1+K)
    KW(K)=I3
    JW(I3)=JW(I3)-1
306   LW(K)=I1+K
  DO 307 K=1,I2
    DO 308 L=K,I2
      IF(JW(KW(L)).LT.JW(KW(K))) THEN
        III=KW(L)
        KW(L)=KW(K)
        KW(K)=III
        III=LW(L)
        LW(L)=LW(K)
        LW(K)=III
      ENDIF
308     CONTINUE
307   CONTINUE
  DO 310 K=1,I2
    I3=KW(K)
    I4=LW(K)
    IF(IW(I3).EQ.0) THEN
      N=N+1
      IW(I3)=N
      JM(I3)=M+2
      JG(I4)=M+1
      M=M+2
    ELSE
      IF(JG(I4).NE.0)GOTO 312
      M=M+1
      JG(I4)=M
    ENDIF
  DO 314 L=IM(I3)+1,IM(I3+1)
    IF(IG(L).EQ.J) THEN
      JG(L)=JG(I4)
      GOTO 314
    ENDIF
    IF((JG(L).NE.0).OR.(IW(IG(L)).EQ.0)) GOTO 314
    M=M+1
    JG(L)=M
    I5=IG(L)
    DO 316 LL=IM(I5)+1,IM(I5+1)
      IF(IG(LL).EQ.IG(I4)) THEN
        JG(LL)=M
        GOTO 314
      ENDIF
316     CONTINUE
314     CONTINUE
312     CONTINUE
310     CONTINUE
      GOTO 302
C
304     CONTINUE
303     CONTINUE

```

```

305     CONTINUE
302 CONTINUE
C *****
C   2次三角要素の作成と renumbering
C *****
      DO 400 I=1,NE
          I1=IE(I,1)
          I2=IE(I,2)
          DO 401 J=IM(I1)+1,IM(I1+1)
              IF(I2.EQ.IG(J))GOTO 402
401     CONTINUE
402     I3=JG(J)
          IE(I,6)=I3
          I1=I2
          I2=IE(I,3)
          DO 403 J=IM(I1)+1,IM(I1+1)
              IF(I2.EQ.IG(J))GOTO 404
403     CONTINUE
404     I3=JG(J)
          IE(I,4)=I3
          I1=I2
          I2=IE(I,1)
          DO 405 J=IM(I1)+1,IM(I1+1)
              IF(I2.EQ.IG(J))GOTO 406
405     CONTINUE
406     I3=JG(J)
          IE(I,5)=I3
          MX=I3
          MN=I3
          IF(MX.LT.IE(I,6))MX=IE(I,6)
          IF(MX.LT.IE(I,4))MX=IE(I,4)
          IF(MN.GT.IE(I,6))MN=IE(I,6)
          IF(MN.GT.IE(I,4))MN=IE(I,4)
          DO 407 J=1,3
              IF(MX.LT.JM(IE(I,J)))MX=JM(IE(I,J))
              IF(MN.GT.JM(IE(I,J))) MN=JM(IE(I,J))
407     CONTINUE
          IF(NN.LT.(MX-MN))NN=MX-MN
400 CONTINUE
C *****
C   ノードの座標値の計算
C *****
      DO 500 I=1,NP
          XX(JM(I))=X(I)
          YY(JM(I))=Y(I)
          DO 501 J=IM(I)+1,IM(I+1)
              XX(JG(J))=(X(I)+X(IG(J)))/2.0
501     YY(JG(J))=(Y(I)+Y(IG(J)))/2.0
500 CONTINUE
C *****
C   パラメータ値の算出及び出力
C *****
      N=NB
      DO 602 I=1,N-1
          DO 603 J=I+1,N
              DO 604 K=IM(IB(J))+1,IM(IB(J)+1)
                  IF(IB(I).EQ.IG(K))GOTO 605
604     CONTINUE

```

```

        GOTO 603
605  IF((BB(I,1).NE.BB(J,1)).OR.(BB(I,2).NE.BB(J,2))) GOTO 603
        NB=NB+1
        IB(NB)=JG(K)
        BB(NB,1)=BB(I,1)
        BB(NB,2)=BB(I,2)
603  CONTINUE
602  CONTINUE
        DO 606 I=1,N
606  IB(I)=JM(IB(I))
        NP=M
        MB=NN
        DO 410 I=1,NE
            IE(I,1)=JM(IE(I,1))
            IE(I,2)=JM(IE(I,2))
410  IE(I,3)=JM(IE(I,3))
        WRITE(6,6004)M,NE,NN
6004  FORMAT('  NODE=',I5,'  ELEMENT=',I5,'  BAND='I5)
C
        RETURN
        END
C
C *****
C  SUBROUTINE SOLVE(INELE,INNOD,IMELE,IMNOD,INBOU,INBAN,
#  IBAI,EPSI,NP,NE,NB,MB,SK,XX,YY,IB,BB,IE,G,GV,GCAP)
C *****
C  DIMENSION XX(IMNOD),YY(IMNOD),G(IMNOD,INBAN),GV(IMNOD),
#  II(6),PG(4),WG(4),GG(3,3),GH(3,3),GCAP(IMNOD,INBAN),
#  IB(INBOU),BB(INBOU,2),IE(INELE,7),
#  GE(6,6),EPSI(2,IBAI)
C *****
C  グラフィック用の座標データ出力
C *****
        IDATA0=60
        OPEN(IDATA0,FILE='DATAGR.DAT',STATUS='UNKNOWN')
        WRITE(IDATA0,*)NP,NE,NB,MB,SK
        DO 8200 I=1,NP
8200  WRITE(IDATA0,*)XX(I),YY(I)
        DO 8201 I=1,NE
8201  WRITE(IDATA0,*)(IE(I,J),J=1,7)
C *****
C  計算用座標値変換
C *****
        DO 8202 I=1,NP
            XX(I)=XX(I)*SK
8202  YY(I)=YY(I)*SK
C *****
        DO 100 I=1,NP
            DO 100 J=1,MB
                G(I,J)=0.0
100  GCAP(I,J)=0.0
C *****
C  係数マトリクスの作成
C *****
        DO 200 N=1,NE
            II(1)=IE(N,1)
            II(2)=IE(N,2)
            II(3)=IE(N,3)

```

```

      II(4)=IE(N,4)
      II(5)=IE(N,5)
      II(6)=IE(N,6)
      L  =IE(N,7)
      SD = EPSI(2,L)
      B1=YY(II(2))-YY(II(3))
      B2=YY(II(3))-YY(II(1))
      B3=YY(II(1))-YY(II(2))
      C1=XX(II(3))-XX(II(2))
      C2=XX(II(1))-XX(II(3))
      C3=XX(II(2))-XX(II(1))
      S=C3*B2-B3*C2
      S1=0.5*EPSI(1,L)/S
      GE(1,1)=S1*(B1**2+C1**2)
      GE(1,2)=-S1/3.0*(B1*B2+C1*C2)
      GE(1,3)=-S1/3.0*(B1*B3+C1*C3)
      GE(1,4)=0.0
      GE(1,5)=-4.0*GE(1,3)
      GE(1,6)=-4.0*GE(1,2)
      GE(2,2)=S1*(B2**2+C2**2)
      GE(2,3)=-S1/3.0*(B2*B3+C2*C3)
      GE(2,4)=-4.0*GE(2,3)
      GE(2,5)=0.0
      GE(2,6)=GE(1,6)
      GE(3,3)=S1*(B3**2+C3**2)
      GE(3,4)=GE(2,4)
      GE(3,5)=GE(1,5)
      GE(3,6)=0.0
      GE(4,4)=8.0/3.0*(GE(2,2)+GE(3,3))+2.0*GE(3,4)
      GE(4,5)=GE(3,4)+4.0/3.0*GE(3,3)+GE(3,5)+2.0*GE(1,6)
      GE(4,6)=GE(1,6)+4.0/3.0*GE(2,2)+GE(3,4)+2.0*GE(1,5)
      GE(5,5)=8.0/3.0*(GE(1,1)+GE(3,3))+2.0*GE(1,5)
      GE(5,6)=GE(1,5)+4.0/3.0*GE(1,1)+GE(1,6)+2.0*GE(2,4)
      GE(6,6)=8.0/3.0*(GE(1,1)+GE(2,2))+2.0*GE(1,6)
      DO 201 I=1,5
      DO 202 J=I+1,6
      GE(J,I)=GE(I,J)
202 CONTINUE
201 CONTINUE
      DO 203 I=1,6
      DO 204 J=1,6
      IF(II(I).GT.II(J)) GOTO 204
      I1=II(I)
      J1=II(J)-II(I)+1
      G(I1,J1)=G(I1,J1)+GE(I,J)
      GCAP(I1,J1)=GCAP(I1,J1)+GE(I,J)
204 CONTINUE
203 CONTINUE
      IF(SD.EQ.0) GOTO 205
      SD=SD*S/6.0
      GV(II(4))=GV(II(4))+SD
      GV(II(5))=GV(II(5))+SD
      GV(II(6))=GV(II(6))+SD
205 CONTINUE
200 CONTINUE
C *****
C      駆動項の設定
C *****

```

```

30 CONTINUE
  DO 31 I=1,NB
    K = IB(I)
    V= BB(I,1)
    B= BB(I,2)
    IF(B.EQ.0.0)GOTO 33
    GV(K)=GV(K)+B
    GOTO 32
33 DO 34 J=1,NP
  IF((J.LT.(K-MB)).OR.(J.GT.(K+MB)).OR.(J.EQ.K)) GOTO 35
  IF(J.LE.K)GOTO 36
  I1=K
  J1=J-K+1
  GOTO 37
36 I1=J
  J1=K-J+1
37 CONTINUE
  GV(J)=GV(J)-G(I1,J1)*V
  G(I1,J1)=0.0
35 CONTINUE
34 CONTINUE
  G(K,1)=1.0
  GV(K)=V
32 CONTINUE
31 CONTINUE
C *****
C   連立1次方程式
C *****
  DO 40 I=1,NP-1
    M1=I+MB
    IF(M1.LE.NP) GOTO 41
    M1=NP
41 CONTINUE
  DO 42 J=I+1,M1
    J1=J-I+1
    IF(G(I,J1).EQ.0) GOTO 43
    A1=G(I,J1)/G(I,1)
    GV(J)=GV(J)-A1*GV(I)
    DO 44 K=J,M1
      J2=K-J+1
      J3=K-I+1
      G(J,J2)=G(J,J2)-A1*G(I,J3)
44 CONTINUE
43 CONTINUE
42 CONTINUE
40 CONTINUE
  GV(NP)=GV(NP)/G(NP,1)
  DO 45 I=1,NP-1
    J1=NP-I
    M1=J1+MB
    IF(M1.GT.NP)M1=NP
    DO 46 J=J1+1,M1
      J2=J-J1+1
      GV(J1)=GV(J1)-G(J1,J2)*GV(J)
46 CONTINUE
  GV(J1)=GV(J1)/G(J1,1)
45 CONTINUE
C *****

```

C 電界計算

C \*\*\*\*\*

DO 50 I=1,NP

G(I,1)=0.0

G(I,2)=0.0

G(I,3)=0.0

50 CONTINUE

DO 51 N=1,NE

I1=IE(N,1)

I2=IE(N,2)

I3=IE(N,3)

I4=IE(N,4)

I5=IE(N,5)

I6=IE(N,6)

L =IE(N,7)

SD = EPSI(2,L)

B1=YY(I2)-YY(I3)

B2=YY(I3)-YY(I1)

B3=YY(I1)-YY(I2)

C1=XX(I3)-XX(I2)

C2=XX(I1)-XX(I3)

C3=XX(I2)-XX(I1)

S=C3\*B2-B3\*C2

G1=3.0\*B1\*GV(I1)-B2\*GV(I2)-B3\*GV(I3)+4.0\*B3\*GV(I5)+4.0\*B2\*GV(I6)

G2=-B1\*GV(I1)+3.0\*B2\*GV(I2)-B3\*GV(I3)+4.0\*B3\*GV(I4)+4.0\*B1\*GV(I6)

G3=-B1\*GV(I1)-B2\*GV(I2)+3.0\*B3\*GV(I3)+4.0\*B2\*GV(I4)+4.0\*B1\*GV(I5)

H1=3.0\*C1\*GV(I1)-C2\*GV(I2)-C3\*GV(I3)+4.0\*C3\*GV(I5)+4.0\*C2\*GV(I6)

H2=-C1\*GV(I1)+3.0\*C2\*GV(I2)-C3\*GV(I3)+4.0\*C3\*GV(I4)+4.0\*C1\*GV(I6)

H3=-C1\*GV(I1)-C2\*GV(I2)+3.0\*C3\*GV(I3)+4.0\*C2\*GV(I4)+4.0\*C1\*GV(I5)

G(I1,1)=G(I1,1)+1.0

G(I1,2)=G(I1,2)-G1/S

G(I1,3)=G(I1,3)-H1/S

G(I2,1)=G(I2,1)+1.0

G(I2,2)=G(I2,2)-G2/S

G(I2,3)=G(I2,3)-H2/S

G(I3,1)=G(I3,1)+1.0

G(I3,2)=G(I3,2)-G3/S

G(I3,3)=G(I3,3)-H3/S

G(I4,2)=G(I4,2)-0.5\*(G2+G3)/S

G(I4,3)=G(I4,3)-0.5\*(H2+H3)/S

G(I5,2)=G(I5,2)-0.5\*(G1+G3)/S

G(I5,3)=G(I5,3)-0.5\*(H1+H3)/S

G(I6,2)=G(I6,2)-0.5\*(G1+G2)/S

G(I6,3)=G(I6,3)-0.5\*(H1+H2)/S

G(I4,1)=G(I4,1)+1.0

G(I5,1)=G(I5,1)+1.0

G(I6,1)=G(I6,1)+1.0

51 CONTINUE

DO 52 I=1,NP

G(I,2)=G(I,2)/G(I,1)

G(I,3)=G(I,3)/G(I,1)

52 CONTINUE

C \*\*\*\*\*

C データ出力

C \*\*\*\*\*

DO 61 I=1,NP

61 WRITE(IDATA0,601)GV(I)

601 FORMAT(2X,F15.9)

```

      CLOSE(IDATA0)
C *****
C  静電容量値の計算
C *****
      DO 301 L=1,NB
        K = IB(L)
        V= BB(L,1)
        B= BB(L,2)
        IF((B.NE.0.0).OR.(V.NE.1.0))GOTO 302
        K1=K-MB
        K2=K+MB
        IF(K1.LT.1)K1=1
        IF(K2.GT.NP)K2=NP
        DO 303 N=K1,K2
          IF(N.LE.K)GOTO 304
          I=K
          J=N-K+1
          GOTO 305
304 CONTINUE
          I=N
          J=K-N+1
305 CONTINUE
          Q=Q+GCAP(I,J)*GV(N)
303 CONTINUE
302 CONTINUE
301 CONTINUE
        CAP=8.854*Q*0.001
        WRITE(6,4000)CAP
4000 FORMAT('  CAPACITY=',F20.10,'(pF/mm)')
        WRITE(10,4001)CAP
4001 FORMAT(2X,F20.10)
C
      RETURN
C
      END

```

### 3.4. 等電位線表示用プログラム [FEMOUT.FOR]

本プログラムでは、ソルバーで計算されたポテンシャル値を用いて等ポテンシャル線を記述する。

計算は、三角要素の辺上における指定ポテンシャル点を算出した後、三角要素内部は直線で結んでいる。内部の補間関数を用いていない。そのため粗くメッシュが設定されている場所では、折れ曲がった線となっている。

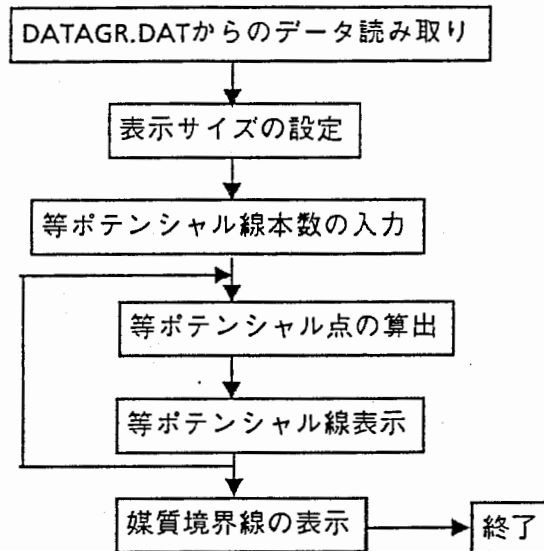


図3-3 プログラムフロー

```
C PROGRAM MAIN
C ***** Potential Line drawing program
C
C DIMENSION XXX(20),YYY(20),XX(3000),YY(3000),IP(3000,7),ZI(3000)
C #,AAA(6),NN(20,2),IIP(3000,8)
C *****Data read from disc
C IDATA1=60
C OPEN(IDATA1,FILE='DATAGR.DAT',STATUS='OLD')
C READ(IDATA1,*)NP,NE,NB,MB,SK
C DO 100 I=1,NP
C READ(IDATA1,*) XX(I),YY(I)
C XMAXM=MAX(XMAXM,XX(I))
C YMAXM=MAX(YMAXM,YY(I))
C XMINM=MIN(XMINM,XX(I))
C YMINM=MIN(YMINM,YY(I))
100 CONTINUE
C AMAXM=MAX(XMAXM,YMAXM)
C AMINM=MIN(XMINM,YMINM)
C
C DO 101 I=1,NE
C READ(IDATA1,*)(IP(I,J),J=1,7)
C WRITE(6,*) I
101 CONTINUE
C ZMIN=1000.0
```



```

ZMAX=-1000.0
DO 102 I=1,NP
  READ(IDATA1,*)ZI(I)
  ZMIN=MIN(ZMIN,ZI(I))
  ZMAX=MAX(ZMAX,ZI(I))
102 CONTINUE
CLOSE(IDATA1)
C
C *****Devide line set
  WRITE(6,2000)ZMIN,ZMAX
2000 FORMAT(' Potential min=',F10.5,5X,' max=',F10.5)
  WRITE(6,2001)
2001 FORMAT(' How many lines do you want ?? ',,$)
  READ(5,*)MANY
  AA=MANY
  ZDEV=(ZMAX-ZMIN)/AA
C
C *****Graphic setting
  IERR=6
  ICONID=0
  IWKID=1
  ITRAN=1
  CALL GOPKS(IERR)
  CALL GQEWK(1,IERRIN,NUMBER,ITYPE)
  CALL GOPWK(IWKID,ICONID,ITYPE)
  CALL GACWK(IWKID)
  CALL GQMDS(ITYPE,IERRIN,DCUNIT,RX,RY,LX,LY)
  CALL GSWN(ITRAN,AMINM-0.5,AMAXM+0.5,AMINM-0.5,AMAXM+0.5)
  CALL GSVP(ITRAN,0.0,1.0,0.0,1.0)
  CALL GSELNT(ITRAN)
  CALL GSWKWN(IWKID,0.0,1.0,0.0,1.0)
  RX=MIN(RX,RY)
  RY=RX*0.1
  RX=RX*0.9
  CALL GSWKVP(IWKID,RY,RX,RY,RX)
  CALL GUWK(IWKID,1)
C
C *****Drawing
C =====line setting
  ZO=ZMIN-ZDEV
  JOK=0
200 CONTINUE
  CALL GSLWSC(1.0)
  ZO=ZO+ZDEV
  IF(ZO.GT.ZMAX)JOK=1
  IF(JOK.EQ.1)ZO=ZMAX
  IF(ZO.EQ.ZMIN)CALL GSLWSC(5.0)
  IF(ZO.EQ.ZMAX)CALL GSLWSC(5.0)
  DO 201 I=1,NE
  JJ=0
  IF((ZI(IP(I,1)).EQ.ZO).AND.(ZI(IP(I,2)).EQ.ZO))GOTO 202
  GOTO 203
202 XXX(1)=XX(IP(I,1))
  YYY(1)=YY(IP(I,1))
  XXX(2)=XX(IP(I,2))
  YYY(2)=YY(IP(I,2))
  CALL GPL(2,XXX,YYY)
  JJ=1

```

```

203 IF((ZI(IP(I,3)).EQ.ZO).AND.(ZI(IP(I,2)).EQ.ZO))GOTO 204
GOTO 205
204 XXX(1)=XX(IP(I,3))
YYY(1)=YY(IP(I,3))
XXX(2)=XX(IP(I,2))
YYY(2)=YY(IP(I,2))
CALL GPL(2,XXX,YYY)
JJ=1
205 IF((ZI(IP(I,1)).EQ.ZO).AND.(ZI(IP(I,3)).EQ.ZO))GOTO 206
GOTO 207
206 XXX(1)=XX(IP(I,1))
YYY(1)=YY(IP(I,1))
XXX(2)=XX(IP(I,3))
YYY(2)=YY(IP(I,3))
CALL GPL(2,XXX,YYY)
JJ=1
207 IF(JJ.EQ.1)GOTO 290
AAA(1)=(ZI(IP(I,1))-ZO)
AAA(2)=(ZI(IP(I,2))-ZO)
AAA(3)=(ZI(IP(I,3))-ZO)
AAA(4)=(ZI(IP(I,4))-ZO)
AAA(5)=(ZI(IP(I,5))-ZO)
AAA(6)=(ZI(IP(I,6))-ZO)
JJ=0
IF((AAA(1)*AAA(6)).GE.0.0)GOTO 215
JJ=JJ+1
NN(JJ,1)=IP(I,1)
NN(JJ,2)=IP(I,6)
215 IF((AAA(6)*AAA(2)).GE.0.0)GOTO 216
JJ=JJ+1
NN(JJ,1)=IP(I,6)
NN(JJ,2)=IP(I,2)
216 IF((AAA(2)*AAA(4)).GE.0.0)GOTO 217
JJ=JJ+1
NN(JJ,1)=IP(I,2)
NN(JJ,2)=IP(I,4)
217 IF((AAA(4)*AAA(3)).GE.0.0)GOTO 218
JJ=JJ+1
NN(JJ,1)=IP(I,4)
NN(JJ,2)=IP(I,3)
218 IF((AAA(3)*AAA(5)).GE.0.0)GOTO 219
JJ=JJ+1
NN(JJ,1)=IP(I,3)
NN(JJ,2)=IP(I,5)
219 IF((AAA(5)*AAA(1)).GE.0.0)GOTO 220
JJ=JJ+1
NN(JJ,1)=IP(I,5)
NN(JJ,2)=IP(I,1)
220 CONTINUE
IF(JJ.LT.2)GOTO 290
DO 221 II=1,JJ
BB2=ZI(NN(II,2))-ZI(NN(II,1))
IF(BB2.EQ.0.0)GOTO 221
BB1=(ZO-ZI(NN(II,1)))/BB2
XXX(II)=XX(NN(II,1))+BB1*(XX(NN(II,2))-XX(NN(II,1)))
YYY(II)=YY(NN(II,1))+BB1*(YY(NN(II,2))-YY(NN(II,1)))
221 CONTINUE
CALL GPL(JJ,XXX,YYY)

```

```
290 CONTINUE
201 CONTINUE
  IF(JOK.EQ.1)GOTO 500
  IF(ZO.LE.ZMAX)GOTO 200
```

C

```
500 CONTINUE
  CALL GSLWSC(1.0)
  CALL GSPLCI(4)
  DO 298 I=1,NP
  DO 300 J=1,8
300 IIP(I,J)=0
298 CONTINUE
  DO 301 I=1,NE
  DO 302 J=4,6
302 IIP(IP(I,J),IP(I,7))=IIP(IP(I,J),IP(I,7))+1
301 CONTINUE
  DO 400 K=1,8
  DO 401 I=1,NE
  IF(IP(I,7).NE.K)GOTO 401
  IF(IIP(IP(I,6),K).GE.2)GOTO 303
  XXX(1)=XX(IP(I,1))
  YYY(1)=YY(IP(I,1))
  XXX(2)=XX(IP(I,2))
  YYY(2)=YY(IP(I,2))
  CALL GPL(2,XXX,YYY)
303 IF(IIP(IP(I,5),K).GE.2)GOTO 304
  XXX(1)=XX(IP(I,1))
  YYY(1)=YY(IP(I,1))
  XXX(2)=XX(IP(I,3))
  YYY(2)=YY(IP(I,3))
  CALL GPL(2,XXX,YYY)
304 IF(IIP(IP(I,4),K).GE.2)GOTO 401
  XXX(1)=XX(IP(I,3))
  YYY(1)=YY(IP(I,3))
  XXX(2)=XX(IP(I,2))
  YYY(2)=YY(IP(I,2))
  CALL GPL(2,XXX,YYY)
401 CONTINUE
400 CONTINUE
  CALL GPAUSE(IWKID)
```

C

```
  CALL GDAWK(IWKID)
  CALL GCLWK(IWKID)
  CALL GCLKS
  STOP
  END
```

#### 4. 使用例

以下では、本プログラムを用いた計算実施例を示す。各ファイルの作成法についての参考としてほしい。

##### 4.1 マイクロストリップ線路の解析

図4-1にマイクロストリップ線路を示す。

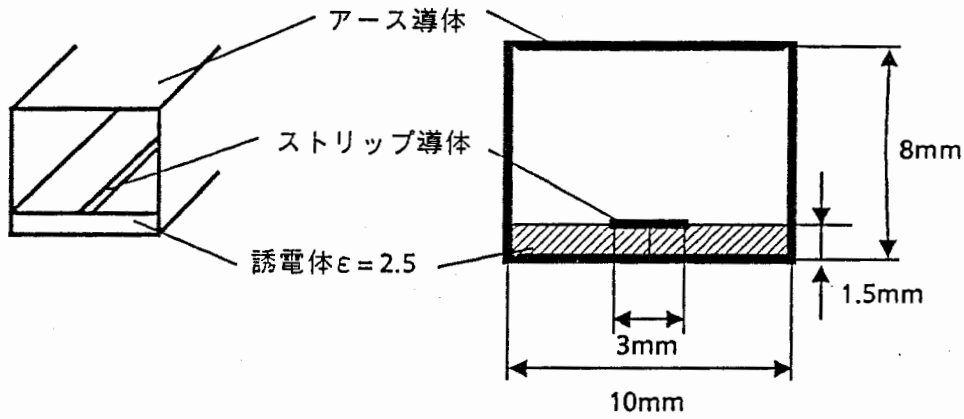
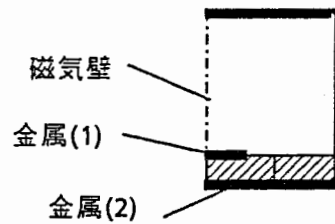


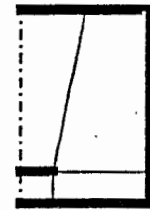
図4-1 解析モデル

まず、簡易データファイルの作成手順を示す。

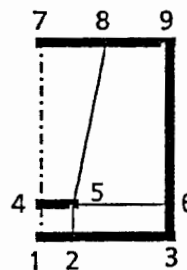
作業手順(1) 導波路の対称性を利用して1/2の領域をもちいて計算する。  
境界条件は右図となる。  
媒質は、空気を1、誘電体を2とする。



作業手順(2) 導波路の断面を四角形要素に分割する。



作業手順(3) ノードに番号付けする。





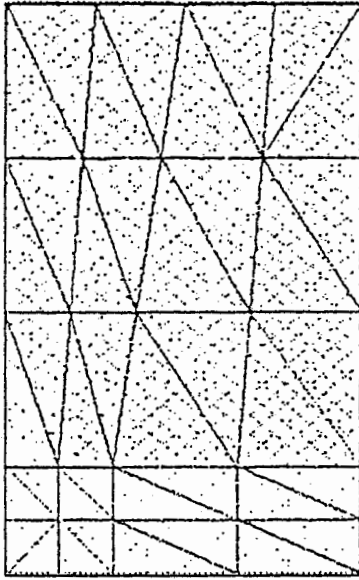


図4-2 メッシュ分割図

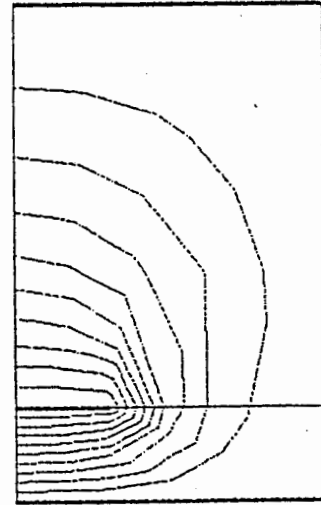


図4-3 等ポテンシャル線

以上により簡易データファイルが作成された。MESH0.EXE を起動すると、図4-2の様に分割メッシュが表示される。

計算を始めるまえに、媒質のデータをファイルDATA001.DATに記入する。本モデルでは、媒質の区分として空気を1、誘電体を2にしているので誘電率を次のように記入する。

```
1.0  0.0
2.5  0.0
```

線路の電気特性を計算するには、誘電体のない場合を計算しなければならない。その為には、上記データファイルDATA001.DATを次のように変更すればよい。

```
1.0  0.0
1.0  0.0
```

この時の静電容量計算値および線路の特性インピーダンス・実行誘電率は次のようになる。ここでは、1/2の領域で計算しているため、単位長さあたりの容量値は計算値の2倍となる。

静電容量値( $\epsilon_r=2.5$ )	静電容量値( $\epsilon_r=1.0$ )	特性インピーダンス	実行誘電率
0.0419117026pF/mm	0.0212966353pF/mm	55.8 $\Omega$	1.97

尚、マイクロストリップ線路のポテンシャル分布は図4-3となる。

## 4.2 結合マイクロストリップ線路の解析

図4-4に対称形な結合マイクロストリップ線路を示す。このような結合2線路の特性を計算する場合、一般にふたつの直行モード(even, odd)を考える。線路の対称形を利用し、半分の領域を計算対象とした場合、各モードは境界壁を各々電気壁・磁気壁に設定した場合に対応する。

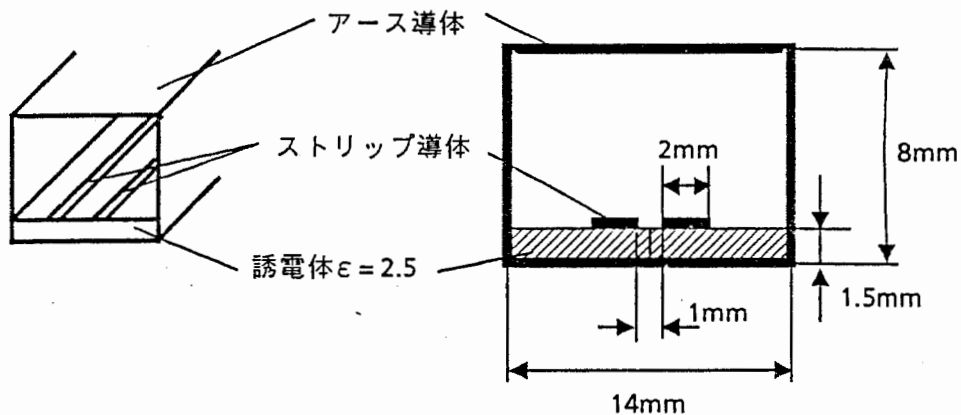
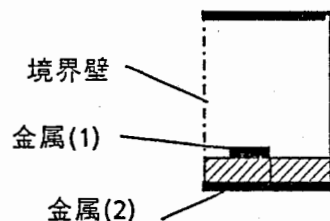


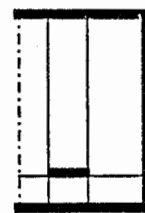
図4-4 解析モデル

以下に、簡易データファイルの作成手順を示す。

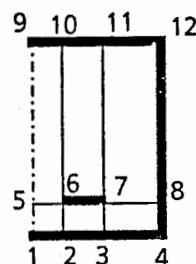
- 作業手順(1) 導波路の対称性を利用して1/2の計領域をもちいて計算する。この時、境界条件は2つのモードに対応させて磁気壁と電気壁の2条件について計算しなければならない。左図に計算領域を示す。媒質は、空気を1、誘電体を2とする。



- 作業手順(2) 導波路の断面を四角形要素に分割する。



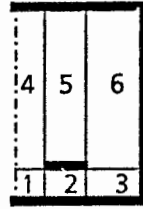
- 作業手順(3) ノードに番号付けする。



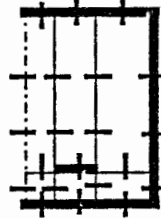
作業手順(4) ノード番号1を座標(0, 0)として各ノードの座標値を求める。

ノード番号	座標値	ノード番号	座標値
1	( 0, 0)	2	(0.5, 0)
3	(2.5, 0)	4	( 7, 0)
5	( 0,1.5)	6	(0.5, 1.5)
7	(2.5,1.5)	8	( 7,1.5)
9	( 0, 8)	10	(0.5, 8)
11	(2.5, 8)	12	( 7, 8)

作業手順(5) 要素に番号付けする。



作業手順(6) 要素の辺の分割数を決める。



作業手順(7) Editを用いて以下のようにデータを MESHDATA.DAT に記入する。

```

1
12
0.0 0.0
0.5 0.0
2.5 0.0
7.0 0.0
0.0 1.5
0.5 1.5
2.5 1.5
7.0 1.5
0.0 8.0
0.5 8.0
2.5 8.0
7.0 8.0
6
1 2 2
1 2 6 5
2 -1 0 0 -1
2 2 2
2 3 7 6
2 -1 0 1 0
3 2 2
3 4 8 7
2 -1 -1 0 0
4 2 3
5 6 10 9
1 0 0 -1 -1
5 2 3
6 7 11 10
1 1 0 -1 0
6 2 3
7 8 12 11
1 0 -1 -1 0
1

```

( 尚、アンダーラインのある境界条件値は、磁気壁の場合は0に置き換わる。)



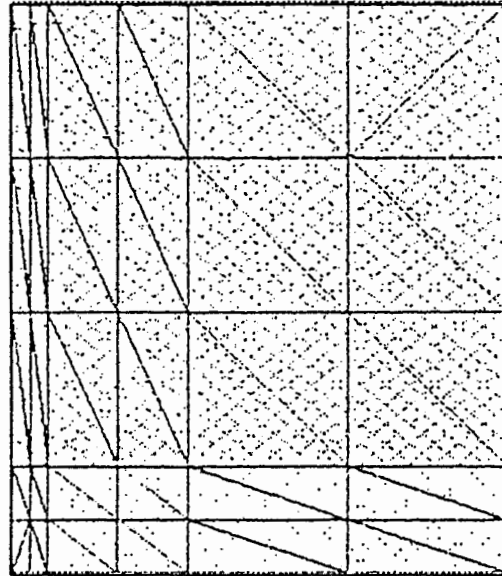
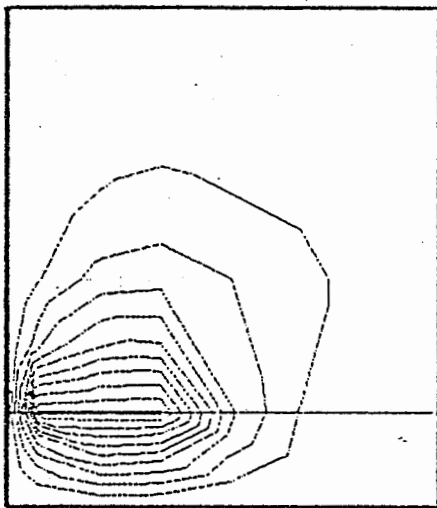
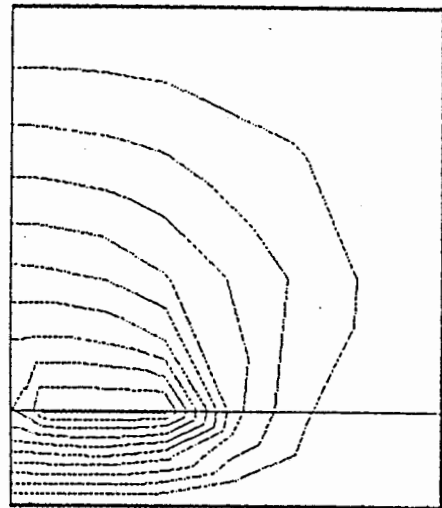


図4-5 メッシュ分割図



even モード



odd モード

図4-6 等ポテンシャル線

以上により簡易データファイルが作成された。MESH0.EXE を起動すると、図4-5の様に分割メッシュが表示される。媒質のデータファイルDATA001.DATは、マイクロストリップ線路の計算時と同様に設定する。

この結果、静電容量計算値および線路の特性インピーダンス・実行誘電率は次のようになる。

	静電容量値( $\epsilon_r=2.5$ )	静電容量値( $\epsilon_r=1.0$ )	特性インピーダンス	実行誘電率
even モード	0.05752pF/mm	0.02838pF/mm	82.5 $\Omega$	2.07
odd モード	0.08119pF/mm	0.04498pF/mm	55.2 $\Omega$	1.81

また、ポテンシャル分布は図4-6となる。

#### 4.3 その他

以上に示した計算においては、計算領域はすべて閉領域を対象としている。この時、容量値は外部境界壁の位置により影響を受ける。従って開放領域を対象とする場合は開放境界壁を充分遠方に設定しなければならない。この時、容量計算値は計算領域の拡張に伴い一定値に漸近していくため、これを目安とすればよい。

またポテンシャル分布の変動が大ききところでは、小さなメッシュを用いて分割しなければ充分近似できない。容量計算値はメッシュ分割数の増加に伴い一定値に漸近していくので、これを目安とすればよい。

なお、開放領域を計算するには無限要素法やハイブリッド法などが知られているが、本プログラムには導入していない。また、鋭角導体補正は行っていない。容量マトリクス作成機能は多導体系において有用であるが本プログラムには導入していない。

なお、本プログラム使用中に不具合を発見された方は、プログラムコードおよびマニュアルを修正して下さるようお願いいたします。

#### 謝辞

本レポートを作成するにあたり、全体の構成についてアドバイスを頂きました小川英一主幹研究員ならびに校正を補助して頂いた木村さんに感謝します。

## 改訂記録等

1990年5月25日初版作成

### \*\*\*\*\* プログラム使用者へのお願い

このマニュアルは、FUJI XEROX J Star で作成しており、文書ファイルはフロッピーに納めています。プログラムの修正・改良等変更事項があった場合は、改訂記録を追加するとともに本文の修正も行って下さい。