



Mesh 6.5

2次元有限要素法メッシュ作成ソフト

Field Precision LLC

PO Box 13595, Albuquerque, NM 87192 U.S.A.

E mail: techinfo@fieldp.com

Internet: <http://www.fieldp.com>

日本国内総代理店

株式会社アドバンスト・サイエンス・ラボラトリー

URL: <http://asl-i.com>

Eメール: info@asl-i.com

予備Eメール: info_asl@yahoo.co.jp

目 次

1	TC : TriComp プログラムコントローラ	4
1.1	はじめに	4
1.2	プログラムとデータのディレクトリ設定	5
1.3	ツール	5
1.4	TriComp の解析プログラム	6
2	メッシュ生成のコンセプト	7
2.1	基盤メッシュ	7
2.2	領域 (Region)	9
2.3	領域定義の順序	10
2.4	領域定義の例	11
2.5	要素サイズの選択	13
3	描画エディタによる Mesh スクリプトの作成	16
3.1	新たなメッシュの生成	16
3.2	解析空間の形状設定とバックアップファイル作成	18
3.3	領域の追加と名前の割り当て	21
3.4	直線、円弧、円の描画	22
3.5	編集コマンド	24
3.6	ビューとスナップ設定の変更	26
3.7	基盤メッシュ(foundation mesh)の改良	28
3.8	スクリプトの作成	29
4	描画エディタのメニューとコマンド	31
4.1	メインメニューから描画(drawing)エディタの起動	31
4.2	作業環境	32
4.3	「DRAWING」(描画)メニュー	36
4.4	「VIEW」(ビュー)メニュー	37
4.5	「INSERT」(挿入)メニュー	38
4.6	「EDIT」(編集)メニュー	39
4.3	「FOUNDATION」(基盤メッシュ)メニュー	42
4.3	「SETTINGS」(設定)メニュー	44
4.3	「INFORMATION」(情報)メニュー	45
4.3	「EXPORT PLOT」(プロット書き出し)メニュー	46

5	スクリプトからのメッシュの生成	48
5.1	メッシュの生成	48
5.2	「FILE」、「PROCESS」、「HELP」メニュー	49
5.3	「PLOT-REPAIR」メニュー — メッシュ表示画面	53
5.4	「PLOT-REPAIR」メニュー — 「REPAIRS」(修復)	55
5.5	Mesh の自律的実行	58
6	Mesh スクリプトのフォーマット	60
6.1	スクリプトファイルの構造	60
6.2	基盤メッシュの基本的定義	62
6.3	領域の定義	64
6.4	点	66
6.5	直線	67
6.6	円弧	68
6.7	充填 (Filled) 領域の境界	68
7	Mesh の高度な技法	69
7.1	可変サイズ要素分割	69
7.2	メッシュスムージング	70
7.3	基盤要素形状の設定	72
7.4	自動補正と境界適合パラメータ	75
7.5	メッシュ生成の問題点	76
7.6	Mesh 出力ファイルのフォーマット	77
8	画像からのメッシュ生成	79
8.1	はじめに	79
8.2	スクリプトの Image(イメージ)セクション	82
8.3	視覚画像(visual image)に対するコマンドと処理手順	84
8.4	データ画像からのメッシュ生成	88
8.5	Image コマンド	94

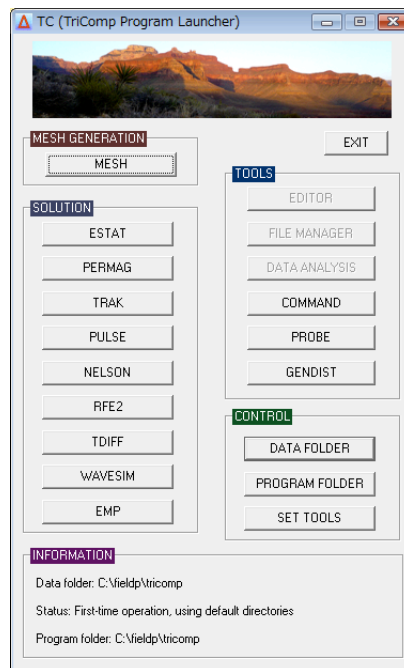


図 1 TC のウィンドウ

第 1 章 TC : TriComp プログラムコントローラ

1.1 はじめに

TriComp のプログラムを実行する際の最初のステップは、プログラムコントローラ TC(tc.exe) の設定です。TC によって、ユーザーは仕事を次の 2 つの点で整理できます。

- デスクトップ上に置くショートカットを 1 つだけにして、乱雑さを最小限に抑えます。
- すべてのプログラムを、指定した 1 つのディレクトリで実行するようにし、ツリーをたどってディレクトリ間を行き来する手間を省きます。

インストール時に、tc.exe へのショートカットを Windows のデスクトップ上や、スタートメニュー内に設定するか尋ねられますので、その指示に従ってショートカットを作成して下さい。また TC を起動したときに、不明のコンフィギュレーションファイルに関するメッセージが表示されますが、無視してください。このファイルは最初のセッションの終わりに作成されます。

TC のウィンドウは、「MESH GENERATION」、「SOLUTION」、「TOOLS」、「CONTROL」、「INFORMATION」という 5 つの機能グループに分けて表示されています。始めの 2 つのグ

ループは、メッシュを作成し、次に物理的解析を行い、結果を分析するという **TriComp** のシミュレーションの機能を示しています。

1.2 プログラムとデータのディレクトリ設定

図 1 は **TC** ウィンドウのスクリーンショットです。いくつかのコマンドボタンだけが利用可能になっていることに注意してください。**TC** は、設定されたプログラムディレクトリで実行可能なプログラムを検出すると、それに対応するボタンを利用可能にします。**Mesh** のボタンはいつでも使用可能になっていますが、解析セクションの機能 (SOLUTION) 項目は、購入した **TriComp** プログラムにより使用の可否が決まります。どのボタンも使用できない状態にあるときは、実行プログラムがインストールされていないディレクトリ内で、**TC** が実行プログラムを検出しようとしているときです。この場合、「PROGRAM FOLDER」ボタンをクリックしてください。ダイアログで実行可能なプログラムのあるディレクトリに移動し、OK ボタンをクリックしてください。「MESH」と「SOLUTION」セクションの実行可能なボタンが使用可能になります。

TriComp 解析関連のすべての入出力ファイルを、指定したデータディレクトリに集めておくのが便利です。「Data Folder」ボタンを用いて、現在の作業ディレクトリの位置を設定してください。この設定後に開かれるプログラムは、そのディレクトリで読み書きすることになります。(この設定変更は、それ以前に開かれていたプログラムには影響しないことを留意してください。) コントロールボックスの下側にある情報(INFORMATION)領域には、1) 現在のデータ (作業) ディレクトリ、2) 最後に行われた操作、および 3) プログラムディレクトリが表示されます。

1.3 ツール

TriComp およびその他のプログラムとともに、ユーザーが他のソフトウェアツールを用いることがあります。**TC** の設定を行うことにより、使用したいユーティリティプログラムを起動させることができます。例えばテキストエディタを定義するには、「CONTROL」セクションの「SET TOOLS」ボタンをクリックしてください。次に表示されるダイアログで、オプションの「Select editor」を選んでください。そして、ディレクトリを移動し、任意のエディタプログラムを選んでください。この操作を終えると、「TOOLS」セクションの「EDITOR」ボタンが使用可能になります。

良いテキストエディタを使うことは、技術的な作業にとっては重要です。ワードパッドのような簡易ツールでは使い勝手がよくありません。**TriComp** シリーズをインストールすると「fieldp」フォルダ内に強力なフリーウェアエディタプログラムである「Context」のインストーラーがコピーされます。このエディタには、**TriComp** シリーズの構文強調表示の機能を追加するとき用いる設定ファイル(highlighter)もあります。この機能は **TriComp** のスクリプトを読みやすくし、またキーワードのタイプミスが起こるのを防ぎます。「ConTextWithHighlighters.zip」のファイルを解凍し、「ConTexSetup.exe」を実行してインストールし、拡張子が「.chl」のファイルをディレクトリ\program files\context\highlighters にコピーしてください。エディタでスクリプトを作成した場合はファイル名の後に Mesh スクリプト用の拡張子「.MIN」またはユーザが購入した解析プログラムのスクリプト拡張子（たとえば、EStat の.EIN など）を付けてください。またそれらの拡張子を ConText に関連付けてください。なお、Field Precision 社のリソースサイト (<http://www.fieldp.com/fplinks.html>) から ConText をはじめ、他の便利なユーティリティプログラムもダウンロードできます。

「SET TOOLS」ボタンを用いて、ほかに 2 つのユーティリティを定義できます。「Select file manager」では、Windows Explorer のようなファイルマネージャプログラムを選択してください。上記リソースサイト上にフリーウェアプログラム「FreeCommander」へのリンクがあります。このプログラムは多くの有用な機能を多く持っており、ファイルを移動し、コピーするデュアル・スクリーンレイアウトを採用しています。スプレッドシート、プロットプログラム、または他の数学的解析ソフトウェアを起動するには「Select data analysis program」ボタンを用いてください。

コマンドプロンプトからまたは独自に作成したバッチファイルのもとで、プログラムを実行したいときには、「TOOLS」セクションの「COMMAND」ボタンをクリックして DOS のウィンドウを開いてください。「PROBE」ユーティリティは、初期値問題解析プログラム（Pulse、TDiff、EMP）により作成した標準の履歴ファイルのデータをプロットします。「GENDIST」は、Trak プログラムの粒子分布ファイルの生成、プロット、解析を行います。

1.4 TriComp の解析プログラム

TC の「SOLUTION」セクションのボタンとして示されているプログラムの機能を表 1 に

示します。

表 1 TriComp の解析プログラム

プログラム	機 能
EStat	誘電体および導体に関する静電場解析。異方性物質に対応。
PerMag	コイル、永久磁石および磁性体に関する静磁場解析。異方性および非線形物体に対応。
Trak	荷電粒子の輸送と粒子銃の設計用。 EStat および PerMag で得た電場および磁場の結果を入力。
Pulse	任意の波形の時間変化する磁場の計算。 非線形磁性体および渦電流に対応。
Nelson	渦電流のある非放射条件内での AC/RF 磁場解析。高周波磁場装置および誘導過熱に適用。
RFE2	RF 加熱および生物医学に応用する非放射条件内での RF 電場。
TDiff	温度依存性のある固体での静的および動的な熱輸送。RFE2、Nelson および WaveSim で得た電力分布をオプションとして入力可。
WaveSim	電磁場の周波数領域の解析。共振器、マイクロ波装置および散乱に適用。
EMP	パルス電源装置、デジタル回路および電磁界干渉等の時間領域電磁場解析。

第 2 章 メッシュ生成のコンセプト

2.1 基盤メッシュ

有限要素解析の最初のステップは、解析空間を小さな断片（要素: element）に分割することです。Mesh は、2 次元解析用として、 x - y または z - r 面内に三角形断面を持つ要素を生成します。メッシュの生成手順は、次の 2 つのステップからなります。

- 解析する断面を、適当なサイズの 3 角形の基本線(baseline)の集合で満たします。
- 要素のファセット (3 角形要素の辺) が物質境界に近く位置するように節点(node)をシフトします。

最初の処理を基盤メッシュ (foundation mesh) の生成、次の処理を要素の形状適合化 (conformalization) と言うことにします。

基盤メッシュは、解析領域を含む十分な大きさの四角い領域を満たす、三角形のほぼ均一な集合です。その一例を図 2 に示します。有効な解析領域が、四角い領域全体を占める場合も、占めない場合もあります。基盤メッシュは論理的に構造化されており、すべての節点にインデックスが付けられ、隣接する節点と要素を求めることができます。**Mesh** はユーザーが定義する各物体の境界ベクトルを解析して、境界近くにある節点を、与えられた点、直線または円弧の上に位置するように移動します。節点の移動は、移動後の要素の 3 点で形成される三角形が、すべて妥当な三角形になるような、十分小さなものである必要があります。基盤メッシュが物体の境界によく合致していない場合は、形状適合化に失敗する場合があります。したがって、すべての三角形の大きさを適切に選ぶこと、また三角形の大きさを位置によって変化させることも時には必要になります (variable resolution : サイズ可変要素分割)。

Mesh で作業しているときには、座標軸を x 軸 (水平方向) および y 軸 (垂直方向) としています。**Mesh** での直交座標系(planar)または円筒座標系(cylindrical)の表示はユーザーの便宜のためだけで、実際の操作には関係ありません。要素を z 方向に無限に延びる柱状の領域ととらえるか、 z 軸から離れるに従って体積が増えていく環状の領域ととらえるかは、各解析プログラムが解釈します。円筒解析用にメッシュを生成するときに覚えておくべきことが 2 つあります。

- 水平軸は z 軸であり、すべての物体は θ に方向に対称である必要があります。
- 垂直軸は r 軸です。垂直方向の座標の最小値は、 $r \geq 0.0$ という条件を満たさなければなりません。 z - r プロット図は円筒構造を、縦方向に切った断面ではないことを覚えておいてください。 r に負の値は定義できません。

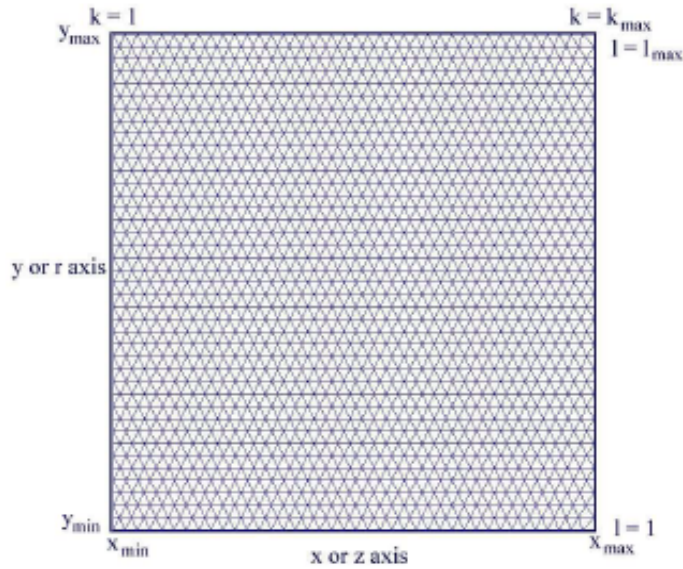


図2 基盤メッシュのパラメータ

基盤メッシュは、水平軸方向に $x_{\min}$ から $x_{\max}$ まで、垂直軸方向に $y_{\min}$ から $y_{\max}$ までの範囲の四角形です (図2)。節点(node)のインデックスは、 x 軸方向に $1 \leq K \leq K_{\max}$ 、 y 軸方向に $1 \leq L \leq L_{\max}$ の範囲にある数値です。境界条件を設定するために外部点を付け加えますので、基礎メッシュの点の総数は、 $I_{\max} = (K_{\max} + 2)(L_{\max} + 2) - 1$ となります。

2.2 領域 (region)

領域(region)とは、節点(node)と要素(element)が共通の物質特性を持つ、解析空間中の区域です。**Mesh** は、ある領域のすべての節点と要素に (1 から 127 までの) 整数の番号を割り当てます。この番号は、その領域がスクリプトに現れる順に指定されます。例えば **EStat** の解析では、電極、誘電体、空間電荷領域、固定ポテンシャル境界、対称境界が、それぞれ 1 つの領域を構成します。一方熱伝導解析では、一定温度の物体、熱源、または異なった熱伝導率、密度、熱容量の値を持つ各々の物質が各領域を構成します。**Mesh** で定義した時点では領域には特性が設定されておらず、同じメッシュを多様な物理解析に用いることができます。

領域には、Filled(充填)と Open(オープン)という 2 つのカテゴリがあります。オープン領域の境界は点(point)、直線(line)、円弧(arc)のいずれかの集まりからなります。この場合、**Mesh** はメッシュ作成処理時に、新たな領域番号を領域境界上の各節点(node)に割り当て、要素

(element)については新たな領域番号の再割り当てを行いません。たとえば、EStat では、オープン領域が対称境界(symmetry boundary)または固定ポテンシャル境界(fixed-potential boundary)を表すことはよくあります。連結されていない点の集合からなるオープン領域は、ワイヤグリッドをシミュレートするとき用いることができます。オープン領域を定義する直線と円弧は、終端点同士は接続されていても、されていなくても構いません。ただ一つの制約は、単一領域のベクトルは互いに交叉してはいけないことです。充填(Filled)領域は、電極または誘電体のような中身の詰まった物体を表します。ここで、直線または円弧のベクトルは、 x - y または z - r 空間で閉曲線を定義します。Mesh は境界を処理したあと、境界で囲まれたすべての要素と節点に（スクリプトに現れた順に）領域番号を割り当てます。

ある一つの点を形状適合化するために、Mesh は最も近くの、固定されていない節点を領域境界上の位置に移動し、その節点に領域番号を設定します。そして以降の操作によって動かないように、プログラムは位置決めされた節点を固定します。直線と円弧について、Mesh は始点と終点における節点をまず移動し、その後に固定します。そしてプログラムは、論理的に接続された点から点へと進む経路に沿って、ベクトルに最も近い節点を移動し、固定しながら、始点から終点まで進みます。この処理によって、辺が境界ベクトルに沿って整列した、一連の三角形要素を形成していきます。

2.3 領域定義の順序

Mesh スクリプトに領域が記述される順番は重要です。その理由は、後で定義された領域が、それ以前に共有空間中で節点または要素に割り当てられた領域番号を上書きするからです。たとえば、絶縁シースで覆われた円形導線近くの静電場を計算したいとします。そのための処理手順としては、始めに絶縁誘電体を表す円形の充填 (Filled) 領域を定義します。その次に、一定ポテンシャルを持つ導線の断面を表す、より小さな円形の領域を、前に定義した絶縁誘電体の Filled 領域の内側に定義します。小さな方の円形領域の内部にあるすべての要素は、第 2 の領域番号を持つことになります。また導線境界上の節点は、導線領域があとで付け加えられたため、固定ポテンシャル条件を持ちます。

上書き処理は、境界上の共有節点が正しい領域番号を持つために、特別な境界条件を持つ領域がスクリプトの最後の方に現れる必要があります。たとえば、電極と誘電体が共通表面を持つ解析をします。共通表面上の節点は一定ポテンシャル領域の番号を保持するように、電極領域は誘電体の後の方でスクリプトに現れるようにしなければなりません。仮に、解析結果が非物理的な電気力線を持つようであるなら、Mesh スクリプトにおける領域の順序付けが正しくないことが原因として考えられます。



図 3 領域の順序効果の例。第 1 象限のみを表示。

2.4 領域定義の例

図 3 で示す例により、**Mesh** における領域設定の仕方を説明します。この例は、接地した箱の内部にある対向電極による静電場の解析です。上側と下側の電極の電圧は、それぞれ+1500V および-1500V です。この問題では、2つの対称面があります。1つの対称面は上側と下側の電極に関する面 $y=0.0$ で、ポテンシャルは $\phi(x, y) = -\phi(x, -y)$ という条件を満たします。したがって、面 $y=0.0$ においては $\phi=0.0$ となります。もう 1 つの対称面は電極の中心線に関する面 $x=0.0$ です。この対称性により、ポテンシャルは $\phi(x, y) = \phi(-x, y)$ という関係が成り立ちます。したがって、面 $x=0.0$ における電場成分は $E_x=0.0$ 、または $\partial\phi(0, y)/\partial x=0.0$ となります。この 2つの対称面があることから、 $y=0.0$ の境界に沿って Dirichlet(ディリクレ)条件が、また $x=0.0$ の境界に沿って Neumann(ノイマン)条件が課されることになり、計算で決める必要があるのは第 1 象限内のポテンシャルだけになります。第 1 象限以外の位置におけるポテンシャルは対称性により決定できます。図 3 は、第 1 象限にある物体の幾何配置と指定された領域番号、および課される境界条件を示しています。

Mesh における規則は、始めに(誘電体などの)固定境界条件を持たない領域をすべて入力し、次に固定ポテンシャルを持つ物体と Dirichlet 条件を入力します。このような順序で入力することにより、重要な境界が間違って上書きされることがなくなります。この規則を具体的に示す

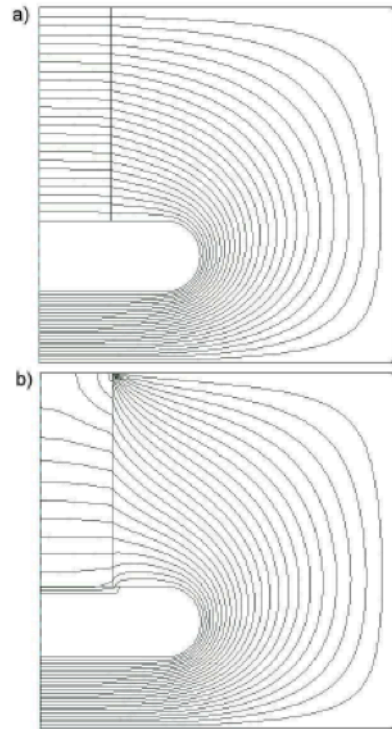


図4 図3の例に対する等ポテンシャル線。

a) 正しい領域定義の順序。b) 正しくない領域定義の順序。

と、まず始めに $\varepsilon_r=1.0$ の真空領域を表す要素を基盤メッシュ中に設定します。領域1は四角形の基礎メッシュ全体を占める充填 (Filled) 領域です。第2のステップは、誘電体の支持物を領域2として加えることです。領域2は領域1の内部にある閉領域です。**Mesh** はまだ固定されていない境界の頂点を移動し、節点と境界に囲まれた要素に領域番号2を付与します。領域3は電極です。電極をあとから加えることにより、電極が誘電体支持物と共有する境界上の節点は、固定ポテンシャル条件を満たすように変わります。左側の境界面 $x=0.0$ に沿った特定されていない点は、自動的に特別な Neumann 条件を満たします。最終ステップは、下端、右側、および上端の境界に沿って Dirichlet 条件 $\phi=0.0$ を設定するために、オープン (Open) 領域の領域4を定義することです。図4aは、解析結果として得られた等ポテンシャル線をプロットした図です。図4bは、誘電体支持物の領域が誤ってスクリプト最後に置かれたときの解析結果です。ポテンシャル線は、誘電体支持物が固定ポテンシャルの電極および境界と接触するところで、物理的に正しくない変化をしています。

2.5 要素サイズを選択

ほとんどの操作は **Mesh** が自動的に処理しますが、基盤メッシュの要素サイズについての決定は、ユーザーがしなければなりません。プログラムのデフォルトによる処理は推量に基づくもので、精度と実行時間のバランスを最適にするためには、ユーザーの判断が必要です。この選択は、経験を重ねていくうちに次第に容易になってきます。解析を始める上で、次のアドバイスが役立つでしょう。

- 最小必要要件として、要素は解析空間内で物体を解像できるように十分小さくなければなりません。要素が大きすぎると、メッシュ生成過程でエラーが生じます。
- 曲線はある程度滑らかであるかといった、メッシュの視覚的な見え方が通常良い目安となります。
- 初めてプログラムに接するユーザーに共通して見られる間違いは、要素の数を多くしすぎる（メッシュを細かくしすぎる）ことです。ユーザーの目標とするところは、実行時間を最も短くしながら、良い精度の解析を行うことであるはずですが、膨大なメッシュを生成して強引に解析していくと、メッシュ生成が遅くなり、解析が困難になり、的確さに欠ける解析になってしまいます。さらに言えば、要素のサイズを小さくしていくことは、必ずしも精度を向上させることになるわけではありません。要素があまりに小さいと、浮動小数点誤差が積もり、実際に精度を悪化させます。
- 重要な問題は、要素が必要なだけ十分小さくなっているかをどのようにして判断するかということです。ここで覚えておくべき重要な点は、数値解析コードは不可解なブラックボックスであると思うべきではないということです。計算結果の良否を確信するために、理論と実験の両方、またはどちらか一方と比較するためにベンチマーク計算を行ってもよいでしょう。
- 第 2 の問題は、正確なベンチマーク計算ができないような複雑な系について計算するときの精度のチェックをどのようにするかということです。標準的な数値計算の手法は、要素サイズを小さくしながら計算を繰り返し、結果を比較することです。結果の変わり方が比較的少なくなってきたときには、変わり方が小さくなり始めたときの要素サイズで十分であったということです。この過程で、解析空間における物体が鋭いエッジを持っている

か注意すべきです。そのような物体間の空間における静電場が正しくても、不連続な(鋭いエッジの)部分で計算される量は不正確です。要素サイズを小さくしていくことは、本来は無限を示す、単に数値的に異なった、無意味な別の値を得るだけです。

ベンチマークの計算例を示すために、図 5 に示す球面コンデンサについて考察することになります。このシステムは内径 R_i および外形 R_o の電極間の空隙からなります。理論的な電気容量は、

$$C = \frac{4\pi\epsilon_0}{1/R_i - 1/R_o}. \quad (1)$$

により与えられます。

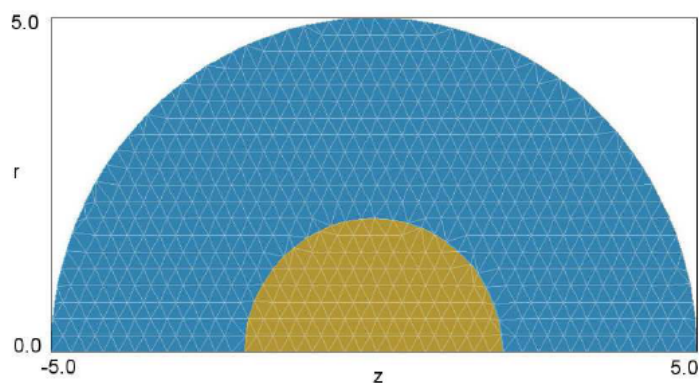


図 5 球面コンデンサのベンチマーク計算において要素サイズが 0.25cm の例

表 2 球面コンデンサのベンチマークシミュレーションにおける計算結果

要素サイズ (cm)	静電容量 体積分 (pF)	静電容量 表面積分 (pF)
0.25	3.718	3.259
0.10	3.710	3.524
0.05	3.710	3.616
0.02	3.708	3.670
0.01	3.708	3.690

$R_i=2.0\text{cm}$ 、 $R_o=5.0\text{cm}$ であるとき、式 1 は $C=3.709\text{pF}$ の値を与えます。図 5 の幾何形状を持つ球面コンデンサに対し、一様に整列した要素のサイズの値を変えて、静電計算を行いました。図 5 ではサイズが約 0.25cm の 3 角形からなる最も粗いメッシュを示しています。次の 2 通り

の方法で静電容量を算出しました。1) 静電場エネルギーを空気要素にわたり体積積分する方法、2) 誘導電荷を算出するために、電場の垂直成分を表面積分する方法。これらの結果を表 2 に示します。体積積分法は分解度が粗くても、相対誤差が 0.24%という精度の良い結果を与え、要素サイズが 0.10cm 以下では誤差がほとんど見られません。表面積分法では、要素サイズが小さくなるとともに、連続的に値が良くなっていきます。その相対誤差は 0.10cm で 5.0%、0.01cm で 0.48%です。計算時間は、0.25cm での 1.0s 未満で、0.01cm では 69.0s へと増大します。

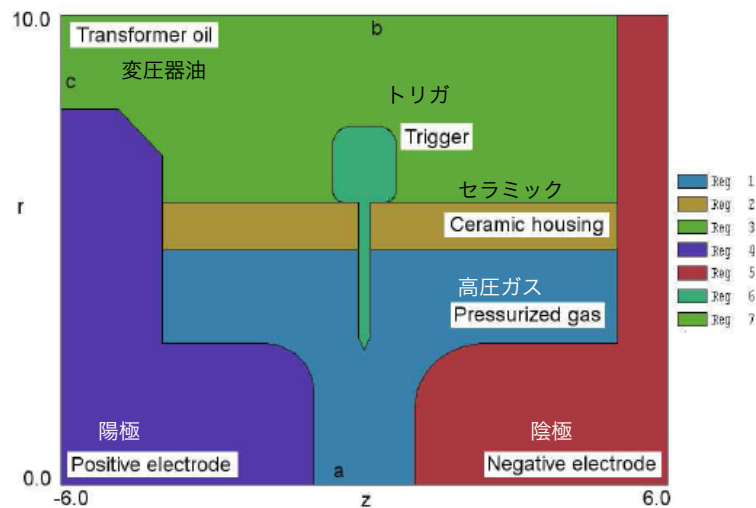


図 6 応用例：高電圧スパークギャップ・スイッチ

第 3 章 描画エディタによる Mesh スクリプトの作成

本章では、汎用 2 次元 CAD ユーティリティである、**Mesh** の描画(drawing)エディタの操作と機能について説明します。計算しようとするシステムの幾何学的構造を記述するスクリプトを作成するには、この描画エディタを使うのが最も早い方法です。ユーザーが描画エディタの機能について学べるように、本節での説明は図 6 に示した例に沿って行います。一歩ずつ手順を追って **Mesh** スクリプトを作成していきます。描画エディタの機能についての詳細は第 4 章で述べます。

3.1 新たなメッシュの生成

プログラムで作業をする前に、まずプランを練りましょう。幾何形状のスケッチを作成し、解析空間で必要な領域(region)を定め、それらの領域をメッシュに加えていく順序を決定しましょう。図 6 の例はパルスパワーシステムのスパークギャップ・スイッチです。この装置は、 z (水

平) 軸に関して円筒対称です。解析空間の下端は z 軸 ($r_{\min} = 0.0$) または軸外点 ($r_{\min} > 0.0$) に対応します。

静電解析には次の個別の物質領域が必要です。

- 領域 1: スパークギャップ容器内部の高圧ガス ($\epsilon_r \approx 1.0$)
- 領域 2: 高圧ガスを閉じ込めるセラミック ($\epsilon_r \approx 7.8$)
- 領域 3: スイッチ外側の変圧器油 ($\epsilon_r \approx 2.7$)
- 領域 4: 高電圧陽極
- 領域 5: 高電圧陰極
- 領域 6: トリガ電極

2.3 節の説明に従って、誘電領域はスクリプトのリストの始めに、固定ポテンシャル領域は後に現れなければなりません。上記の領域は、ゼロでない有限の体積があり、直線または円弧ベクトルの集合により囲まれ、閉じているという Filled(充填)属性を持っています。また、固定ポテンシャル電極領域以外の、解析空間の境界部分についても設定を行う必要があります (Dirichlet 条件)。対称性により、電極間軸上のギャップ (a で印を付けた線分) においては電場の半径方向成分がありません。境界のこの線分は条件

$$\frac{\partial \phi}{\partial r} = 0.0. \quad (2)$$

を満たします。

式(2)は特殊な Neumann 条件です。この条件は、有限要素解析の、特に設定されていない境界線分上では自動的に発生します。また、 $r = r_{\max}$ (線分 b) での境界もこの自然条件にしておきます。この場合、式(2)は正確には成り立ちませんが、ギャップ近傍を計算対象とするときには特殊な Neumann 境界と近似しても構いません。最後に、c で示した壁の線分は陽極と同じポテンシャルであるはずです。この固定ポテンシャル条件を設定するために、ここの線分領域を領域(region) 7 と設定しておきます。

プランの一部として、領域(region)の輪郭をどのように描いていくかを考える必要があります。各領域の輪郭をそのまま正確に描くこともできますが、必要以上の作業を要します。その代わり、次の手順に従って **Mesh** の上書き原則の利点を取り入れることにします。

1. 解析空間全体を満たす四角形の領域(region) 1 (ガス) を設定します。

2. 解析空間の横軸の全長 ($z = -6.0$ から $z = 6.0$) および $r = 5.0$ と $r = 6.0$ の間を範囲とする四角形の領域(region) 2 (セラミック) を加えます。
3. 空間 $-6.0 \leq z \leq 6.0$ および $6.0 \leq r \leq 10.0$ を範囲とする四角形領域 3 (変圧器油) を加えます。
4. 固定ポテンシャル電極 (領域 4~6) の輪郭および境界に沿った線 (領域 7) を描きます。

復習すると、次のような情報から始めます。 1) 計算したい装置の寸法入りのスケッチ、 2) 必要な領域(region)のリスト、 3) 領域を作成する手順のプラン。これでプログラムを用いて作業を始める準備が整いました。

3.2 解析空間の形状設定とバックアップファイル作成

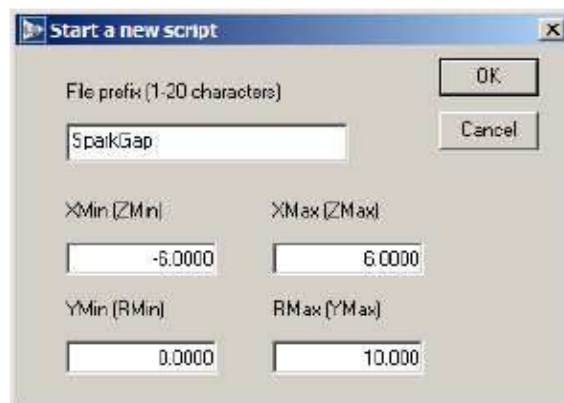


図7 グラフィックスモードでスクリプトを新たに作成するためのダイアログ

次節以降で述べる操作を行うために、TC プログラムランチャから **Mesh** を起動します。「FILE」メニューの「Create script → Create script, graphics」をクリックするか描画エディタで新たなスクリプト(インプットファイル)を作成するボタン(ペンと赤い三角が描かれたアイコン)をクリックしてください。プログラムは図7に示すダイアログを表示します。「File prefix」(拡張子を除いたファイル名)は、**Mesh** の入力スクリプト (SparkGap.MIN) や出力ファイル (SparkGap.MOU) に適用される名前(ここでは「SparkGap」)です。解析する四角形領域の境

界の位置を数値欄に入力して、領域を定義します。図 7 に示すような数値を範囲として入力し、OK ボタンをクリックしてください。図 8 に最初の画面表示を示します。

画面の配置をもう一度よく見てください。画面の上端のメインメニューとツールバーが描画コマンドのものに変わっていることを確認してください。右端にある情報ウィンドウは描画のステータスを示しています。初めは、描画ベクトルはありません。画面の主要部分を占める描画ウィンドウは、図 7 のダイアログで定義した解析空間の境界のみが示されています。**Mesh** はグリッド(格子線)間隔を自動的に決めています。このグリッド間隔が情報ウィンドウに(水平方向に 2.0、垂直方向に 1.0 と)表示されています。グリッド間隔はビュー(画像)をズーム(拡大/縮小表示)すると自動的に変わります。画面の下端にあるステータスバーには、スクリプトのファイル名およびベクトルを入力する現在の領域(region、最初は region 1)を表示しています。このステータスバーは、マウスのスナップモードとスナップ距離(0.5)も表示しています。スナップモードで入力すると、Filled(充填)領域の輪郭を描くベクトルの終端点を正確に連結できます。スナップモードについての詳細は 3.4 節で説明します。当面、デフォルトの設定で、マウス座標はグリッド上の点を示し、0.2 の離散間隔で変化するものと見なしてください。

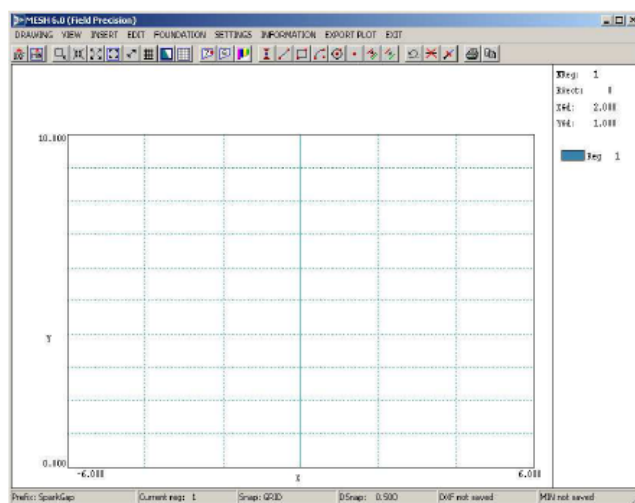


図 8 描画エディタの初期作業環境

描画の最初のステップは、解析空間である四角形領域 region 1 について、その輪郭の線ベクトルを定義することです。メニューから「INSERT」→「Add rectangle」を選ぶか、四角形のアイコンが描かれたボタンをクリックしてください。ステータスバーが座標入力モードに変

わかることに気付いてください。ここで、バーの最初のボックスは操作指示を簡単に表示します。第2と第3のボックスは x と y (または z と r) 座標を表示します。マウスカーソルを四角い解析空間のの左下に動かしてください。マウスの移動とともに、座標は 0.2 の刻みで変化します。座標値が解析領域の一頂点である $(-6.0, 0.0)$ に達したときに左ボタンをクリックしてください。次に、マウスを右上の一角 $(6.0, 10.0)$ まで移動し、左ボタンをクリックしてください。**Mesh** は region 1 の輪郭を示す青色の 4 本の線ベクトルを作成します。情報ウィンドウの中の region 1 に関する項目には、この領域が Filled(充填)属性を持つことを示すために文字 F が付けられます。region 1 は常にゼロでない有限の体積を持たなければならないということから、この設定はデフォルトになっています。(Filled 属性になっていない場合は、3.3 節で述べる「SETTINGS」→「Region properties」で region 1 の Filled チェックボックスをオンにします。) そのほかの領域は、初めは Open 属性を持っています。どの領域が Filled 属性を持つべきかを特定する必要があります (3.3 節参照)。

次のステップに進む前に、解析空間の範囲と region 1 のベクトルを保存することにします。「DRAWING」→「Export DXF file」をクリックするか、それに対応するボタン(DXF という文字と赤い三角のアイコン)をクリックしてください。デフォルトで付けられた(拡張子を除く)ファイル名「SPARKGAP」を採用するために OK ボタンをクリックします。プログラムは、これまで定義した情報を記録するファイル SPARKGAP.DXF を生成します。シンプルなフォーマットの DXF ファイル (Drawing Interchange File、Autodesk 社が Auto CAD 間のデータ互換のために作った中間ファイル) は、どの CAD プログラムでも認識できます (AutoCAD、TurboCAD など)。このファイルは、描画要素のほかに、(領域名など) **Mesh** に固有な情報をコメント行として取り入れます。この情報は CAD プログラムでは無視されます。保存(エクスポート)された DXF ファイルは 2 つの目的を果たします。

- ほかのプログラムへの情報の受け渡し
- 未完成描画のバックアップ

3.3 領域の追加と名前の割り当て

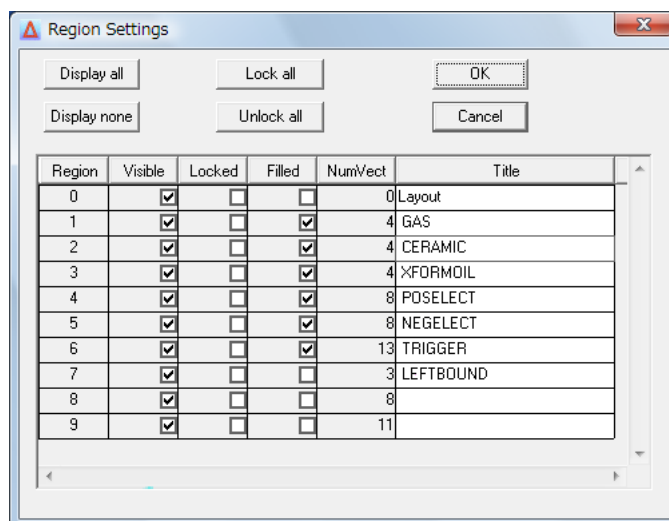


図 9 領域属性設定のダイアログ

次に、セラミック絶縁体の断面を表す四角領域の輪郭を描くことにします。この領域の一部は後で電極領域により上書きされることになります。セラミックの輪郭を描く 4 本のベクトルを region 2 として保存します。メニューの中の「INSERT」→「Start next region」をクリックするか、対応するボタンをクリックしてください。ステータスバーの中の現在の領域情報 (Current reg) が更新されることに留意してください。再び、四角形ツールを用いてください。今度は、四角の角の座標として指定するのは、(-6.0, 5.0) と (6.0, 6.0) です。指定を誤ったときには、メニュー項目の「EDIT」→「Undo last operation」またはそれに対応するボタンを用いてください。同様に、変圧器油を表す 3 番目の四角領域を付け加えてください。「INSERT」→「Start next region」をクリックするかボタンを用いて region 3 の定義を開始し、四角形ツールを用いて (-6.0, 6.0) と (6.0, 10.0) で指定される箱形を作成してください。

ここまでで、一般的な名前を持つ 3 つの領域(region)ができました。そのうち 1 つが Filled(充填)属性を付与されています。メニューコマンドの「SETTINGS」→「Region properties」をクリックして、図 9 に示すダイアログを表示させます。この表には、特別なレイアウト用領域である region 0 に対応する行のほかに、これまでに定義した 3 つの領域(region)に対応する 3 行の表示項目があります。(region 0 にはアライメント用またはテンプレートのベクトルを入力することができます。これらは **Mesh** スクリプトの一部として記録されることはありません。) ここで Filled 列に注目して、region 2 および region 3 に該当するボックスをオンにしてください。描画エディタでの作業を容易にするために、Title 列で分かり易い名前を付けることもできます。これらの名前は、スクリプトを作成するときに記録されます。図 9 は入力操作が完了し

たときのダイアログを示しています。OK ボタンをクリックして、このダイアログを終わらせ、入力した変化を反映させます。

Filled(充填)領域が正しく作られているか（すなわち、ベクトルが連結していて、閉領域を取り囲んでいるか）をチェックすることができます。メニューコマンドの「VIEW」→「Toggle FILL display」をクリックするか、それに対応するボタンを用いてください。画面プロットは、ベクトル表示から、Fille(充填)領域で占められた空間の表示に変わります（図 10）。ベクトルを入力し、編集するすべてのメニューコマンドが使用不可になっていることに留意してください。ベクトルプロットモードに戻るときには、再び「Toggle FILL display」コマンドをクリックしてください。「DRAWING」→「Export DXF file」コマンドを用いて、これまでの作業をバックアップしておくとい良いでしょう。

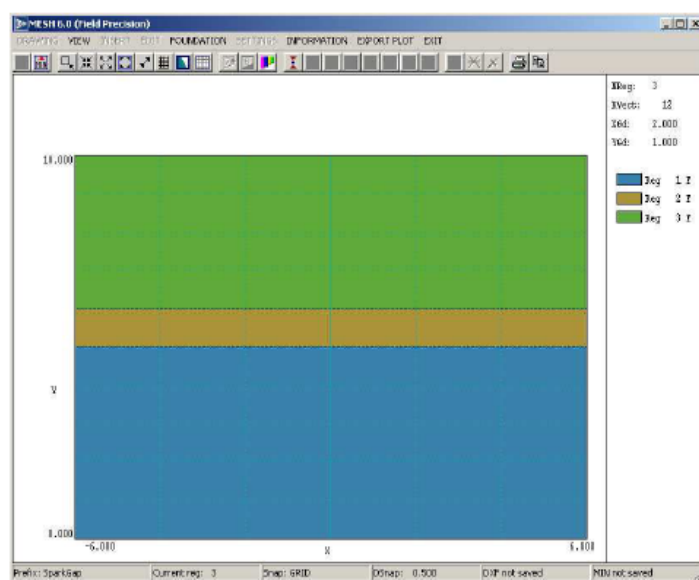


図 10 Fill（塗りつぶし）表示モードで図示された region 1、2、3

3.4 直線、円弧、円の描画

Region 4（陽極）の形状はより複雑です。この形状を描くことは、個々の直線および円弧のベクトル入力の方法、およびスナップモード操作の方法を学ぶ良い機会です。次のステップに進む前に、コマンド「INSERT」→「Start next region」を必ずクリックしてください。小さな誤りをしたときには、「EDIT」→「Undo last operation」コマンドを用いてください。大きな誤りをしたときには、描画した図を破棄して、最も新しく保存したバージョンに戻った方がよいでしょう。描画した現在の図を保存しないでメインメニューに戻るには、「EXIT」→

「Abandon」コマンドをクリックしてください。それから、「FILE」→「Create script」→「Create script, DXF import」を選び、SPARKGAP.DXFを選んでください。

「INSERT」→「Add line(s)」コマンドまたはボタンをクリックしてください。プログラムは座標入力モードになります。このとき、座標が画面の下端に表示され、数本の線ベクトルを一気に入力できます。電極の輪郭の一部を描くことにします。マウ斯卡ーソルを移動し、(-6.0, 8.0) と (-6.0, 0.0) で左ボタンをクリックして、最初の直線を引いてください。次に、マウ斯卡ーソルを移動し、(-6.0, 0.0) と (-1.0, 0.0) で左ボタンをクリックし、2 本目の直線を引いてください。最後に、(-1.0, 0.0) から (-1.0, 2.0) に至る 3 本目の直線を引いてください。マウスの右ボタンをクリックして、座標入力モードを終わらせてください。

ここで、電極のエッジに、半径を定めて円弧を描くことにします。コマンド「INSERT」→「Add arc(s)」→「Add arc: Start-end-center」をクリックしてください。**Mesh** は再び座標入力モードになります。円弧の始点(start)は (-1.0, 2.0)、終点(end)は (-2.0, 3.0) で、中心点(center)は (-2.0, 2.0) です。描いた円弧の曲線は図 11 に示すようになります（見やすくするために、region 1、2、3 は表示していません）。「INSERT」→「Add line(s)」コマンドを用いて (-2.0, 3.0) から (-4.0, 3.0) までと (-4.0, 3.0) から (-4.0, 7.0) までの直線を引いてください。

電極の輪郭を描く残りの 2 本の直線の終点は、正確にスナップグリッドの点上には乗りません。この 2 本の直線を引くために、キーボード入力と終点スナップ (endpoint snap) を利用することにします。コマンド「INSERT」→「Add line(s)」をクリックしてください。始点を決めるため、(-6.0, 8.0) でマウスの左ボタンをクリックします。終点を決めるにはマウスを使わず、F1 キーを押し、キーボード入力のダイアログを表示させてください。このダイアログで、座標値 $x = -4.866$ および $y = 8.000$ を入力してください。最後の直線については、グリッドスナップモードを用いて、(-4.0, 7.0) を始点として指定してください。終点を指定する前に、F2 キーを押して、図 12 の示すスナップモード・ダイアログを表示させてください。

「Endpoint」のチェックボタンにチェックを入れて、OK ボタンをクリックしてください。それから、前に引いた直線の終点 (-4.866, 8.000) 近くまでマウスポインタを移動し、左ボタンをクリックしてください。プログラムは最も近くにあるベクトルの終点を探し、正確に連結します。マウスの右ボタンをクリックして、座標入力モードを終わらせてください。コマンド「SETTINGS」→「Snap control」を選択し、スナップモードを「Grid」に戻してください。電極の輪郭描画を終わらせるために、領域属性(Region properties)ダイアログを呼び出してください。region 4 を Filled に設定し、「PosElect」という名前をつけてください。FILL display モードに切り替えると、プログラムは誘電体領域の一部を上書きして、閉曲線で描かれた電極形状をプロットします (図 6)。

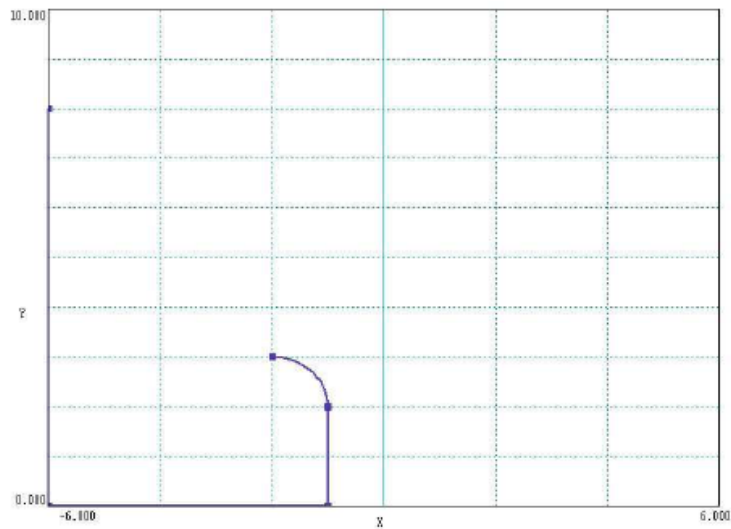


図 11 陽極の輪郭の一部。Region 1、2、3 の表示は省略。



図 12 スナップ制御ダイアログ

3.5 編集コマンド

本節では、右側の負電極の輪郭を描くのに役立つ、いくつかの編集方法を用います。
「INSERT」→「Start next region」をクリックするかボタンを用いて、新たな領域(region)の定義に取り掛かり、四角形ツールを選び、(1.0, 0.0) と (6.0, 3.0) を対角とする四角形を作ってください。また、(5.0, 3.0) と (6.0, 10.0) を対角とするもう 1 つの四角形を作ってください。この結果、全体として閉じた輪郭を形成することなく、2 つの隣接した四角形の組ができます。始めに、下側の四角形の右上側の角に丸みをつける (Fillet) ため、円弧を加えて形状を変

化させることにします。コマンド「EDIT」→「Fillet/chamfer width」を指定してください。表示されたダイアログで、Fillet(丸み)の半径として 1.375 という値を入力してください。OK ボタンをクリックしてダイアログを終わらせ、コマンド「EDIT」→「Fillet」を指定してください。

(1.0, 0.0) から (1.0, 3.0) までの垂直な線分の中央付近にマウスポインタを移動し、左ボタンをクリックして、この線分ベクトルをハイライトさせてください。次に、(1.0, 3.0) から (6.0, 3.0) までの水平線分の中央付近にマウスポインタを移動し、左ボタンをクリックして、この線分ベクトルをハイライトさせてください。丸みづけ(Fillet)操作は自動的にこの 2 つの線分ベクトルを短くするとともに、線分ベクトルに接するように指定半径の円弧を描き、2 つの線分ベクトルを結びつけます。丸みづけ(Fillet)操作は、線分ベクトルが互いに直角の位置になくとも、任意の交差する線ベクトルに対して適用できます。

陰極の輪郭線描画を完成するためには、2 本の線分を消去しなければなりません。上側四角形の下辺と、下側四角形の (5.0, 3.0) から (6.0, 3.0) までの線分を消去します。始めの線分ベクトルを消去するには、まず「EDIT」→「Select」→「Select vector(s)」コマンドを選んでください。マウスカーソルを消去したい線分の近くに移動し、左ボタンをクリックしてください。これで線分がハイライトされ、選択された状態になります。ESC キーを押して選択コマンドを終わらせ、次に EDIT」→「Delete selection」をクリックすることで線分が消去されます。この線分ベクトルが消去できても下側の四角形の上側線分が残っているため、プロット図が変わったようには見えません。2 つの消去したい線分が同じ位置を占めているのに、**Mesh** はなぜ短い線ベクトルを選んで消去したのか、疑問に思われるかもしれません。その答えは、上側の四角形が最後に入力したオブジェクトであるからです。近接しているため特定が困難なとき、**Mesh** は常に最後に入力した（描画要素）を選ぶことになっています。マウスの右ボタンをクリックして、Delete モードを終わらせ、ほかのメニューコマンドを再び使えるようにしてください。

陰極の輪郭線描画を完成させるには、下側オブジェクトである四角形の上辺に沿って線分ベクトルを短くする必要があります。そのために、コマンド「EDIT」→「Trim vector(s)」を選んでください。トリミングベクトルを指定するため、(5.0, 3.0) から (5.0, 10.0) までの上側の四角形の左辺近くにマウスカーソルを移動しクリックしてください。プログラムはトリミングベクトルをハイライト表示します。次に、削除しようとする、線ベクトルの (2.375, 3.0) から (6.0, 3.0) までの部分の近くにマウスポインタを移動してクリックしてください。ESC キーで Trim モードを抜けます。図 13 に、正負の 2 つの電極の完成した輪郭線を示します。region 5 のステータスを Filled に設定し、「NegElect」という名前にしてください。次の段階に進む前に、これまでの作業を必ず保存してください。

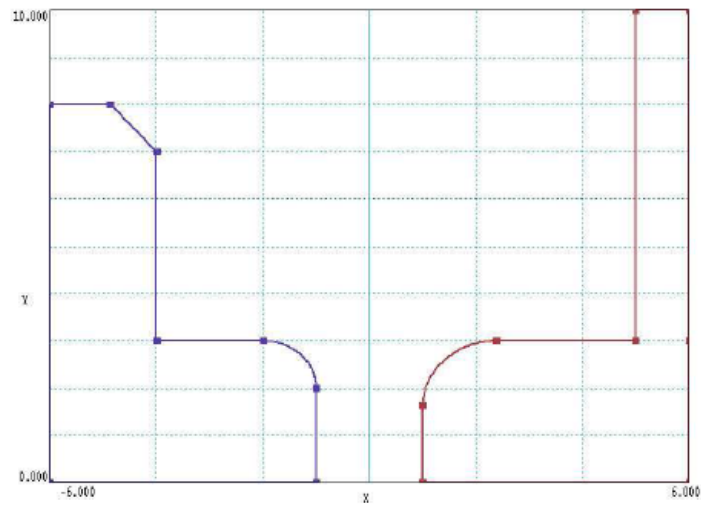


図 13 正負電極の輪郭線完成図

3.6 ビューとスナップ設定の変更

最後の Filled(充填)属性を持つ部品はトリガ電極です。今まで、全解析領域を視野としたビューを見てきました。このビューを狭めるようにスナップ設定を変更し、小さな部品の輪郭線を描きやすいようにしましょう。「VIEW」→「Zoom window」を選択するか、それに対応するボタンを用いてください。2つの対角 $(-3.0, 2.0)$ と $(3.0, 8.0)$ を入力して、ビューボックスを定義します。表示されたグリッド間隔が自動的に、 x 軸方向と y 軸方向に 1.0 に調整されていることに注意してください。次に、「SETTINGS」→「Snap control」を選んでください。モードが必ず「Grid」になるようにして、スナップ距離(Snap distance)を 0.125 に変更してください。また表示させた格子点がスナップ格子点と一致していると作業がしやすくなります。コマンド「SETTINGS」→「Grid control」を選んでください。「Automatic intervals」ボックスのチェックを外し、XGrid と YGrid のボックスに 0.125 を入力してください。「INSERT」→「Start next region」を選び、region 6 の輪郭線の描画に取ります。

図 14 に示す輪郭線を一連の線分を用いて描きましょう。曲線状のエッジを描くとき、コマンド「INSERT」→「Add arc(s)」→「Start-end-center」か、コマンド「EDIT」→「Fillet」のどちらかを使用できます。また誤りを訂正するいくつかの方法があります。

- 座標入力モードで作成した単一ベクトルまたは一連のベクトルを消去するときは、「EDIT」→「Undo last operation」を用いてください。
- 選択したいいくつかのベクトルを消去するときには、「EDIT」→「Select」→「Select vector(s)」で選択し、「EDIT」→「Delete selection」で削除します。この場合、region 2 付近のベクトルが影響を受けないように「Region properties」(領域属性)ダイアログを用いてロックすることが良い方策です。
- Region 6 のすべてのベクトルを消去したいときには、「EDIT」→「Select」→「Select region」で選択し、「EDIT」→「Delete selection」で削除します。

「SETTINGS」→「Region properties」の領域ダイアログを用いて、Region 6 の属性を Filled にし、Trigger という名前を割り当ててください。

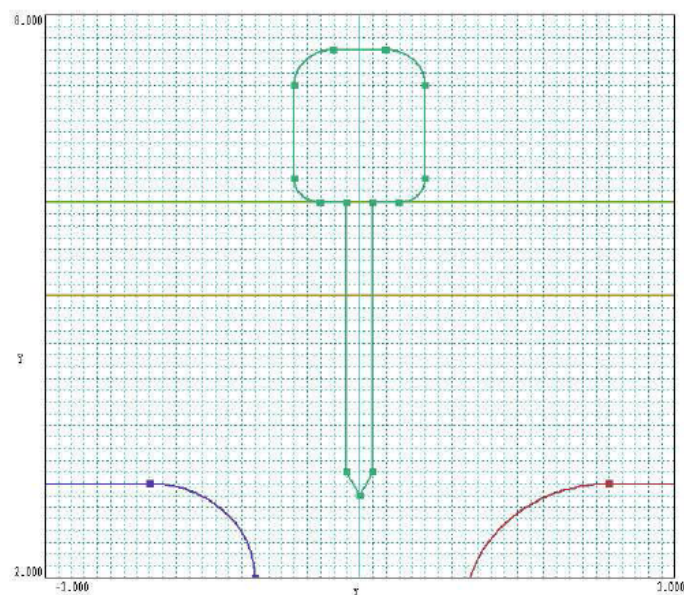


図 14 ビュー、スナップ、およびグリッドを最適に設定したトリガ電極の描画

最後の描画要素は、左側境界上にある $(-6.0, 8.0)$ から $(-6.0, 10.0)$ までの線領域 (region 7) です。この領域(region)の節点(node)には、陽極と同じ固定ポテンシャルを指定します。表示グリッドを自動設定の間隔に戻すには、「SETTINGS」→「Grid control」を用いてください。画面表示を解析の全領域に戻すには、「VIEW」→「Global View」というビューコマンドを用い

てください。「INSERT」→「Start next region」により region 7 の描画に取り掛かり、「INSERT」→「Add line(s)」コマンドを用いて線ベクトルを作成してください。「LeftBound」という名前を割り当て、Filled 属性のボックスにはチェックを入れないでください。完成した図を保存してください。

3.7 基盤メッシュ(foundation mesh)の改良

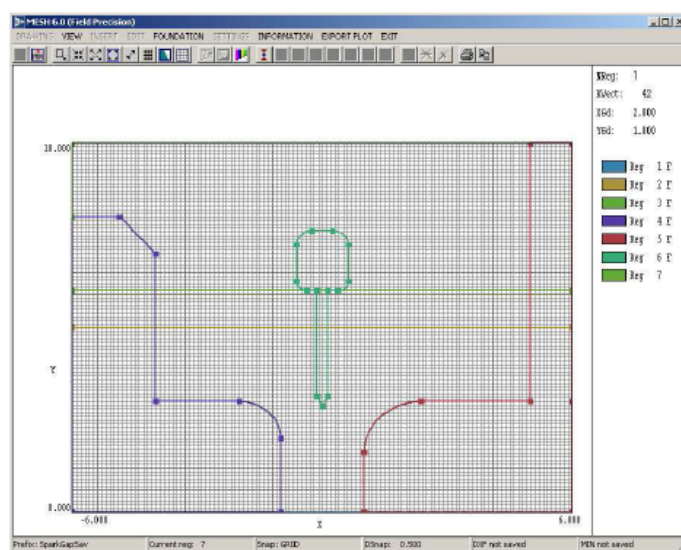


図 15 デフォルト分割した基礎メッシュの画面

これで輪郭ベクトルをすべて生成したので、計算領域を分割する要素(element)の大きさを決める必要があります。基盤メッシュ (foundation mesh) は領域境界に適合する、同じような 3 角形要素の基本線の集合からなります。描画(drawing)エディタには、基盤メッシュ要素の特性を設定するメニューがあります。「FOUNDATION」→「Foundation display」というメニューコマンドもしくは、図 15 の画面に切り替えるボタンを選んでください。灰色の線は 3 角形要素の底辺と高さの近似的な大きさを示しています。基盤メッシュ表示モードを選択すると、基盤メッシュコマンドが使えるようになり、ベクトルの生成・編集のコマンドが使えなくなります。

Mesh は、3 角形の底辺と高さが近似的に 0.10 であるデフォルトの分割を選択しています。この要素サイズは、ほとんどの描画形状に対して差し支えありませんが、トリガ電極の詳細部分を分割するには、もう少し小さな要素を使いたいところです。まず、水平軸方向の要素サイズを変更することから始めることにします。目標として、水平方向の要素サイズが異なる 3 つ

のゾーン（垂直方向に帯状の区域）を設定することにします。この3つのゾーンは、3 角形要素の底辺の大きさが一様に 0.10 である外側 2 つのゾーンと、トリガ電極付近の要素サイズが一様に 0.05 である内側ゾーンです。「FOUNDATION」→「X/Z axis」→「Divide zone X/Z」コマンドを選択してください。マウスポインタを、3 つに区分しようとしている水平単一ゾーンに移動させ、左ボタンをクリックしてください。単一ゾーン全体がハイライトされます。ここで、ポインタを $x = -2.5$ の位置に移動し、左ボタンをクリックしてください。Mesh は水平空間を $x > -2.5$ と $x < -2.5$ の 2 つのゾーンに区分します。再び、「X/Z axis」→「Divide zone X/Z」コマンドを選択してください。カーソルを右側のゾーンに移動し、左ボタンをクリックしてください。次にカーソルを $x = 2.5$ に移動し、左ボタンをクリックしてください。これで、3 角形要素の大きさが等しいゾーンが水平方向に 3 つできました。このゾーン区分操作を完成させるには、中央ゾーンの要素サイズを変更する必要があります。「XZ axis」→「Change element size XZ」コマンドを選択し、中央ゾーンにポインタを移動し、左ボタンをクリックしてください。ダイアログの要素サイズボックスに 0.05 を入力してください。同様な手順によって、空間を垂直方向に、要素サイズが 0.05 の $y < 6.5$ のゾーンと 0.10 の $y > 6.5$ という 2 つのゾーン（水平方向に帯状の区域）に区分してください。図 16 が最終結果です。

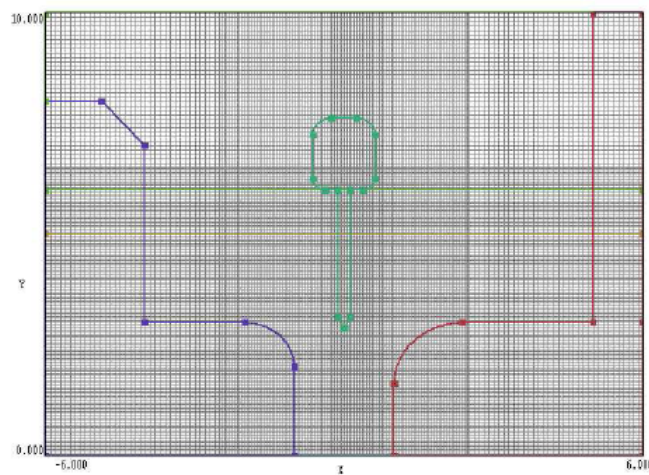


図 16 複数の要素サイズゾーンに区分された基盤メッシュ

3.8 スクリプトの作成

本章では数回にわたり、描画要素をファイル「SPARKGAP.DXF」にバックアップしてきました。これまで達成しようとしてきた目標を完了するには、今まで行った作業をメッシュスクリプト

リプト「SPARKGAP.MIN」に保存する必要があります。この2つのファイル形式の機能と違いを知っておくことは大切なことです。

- DXF ファイルは標準化されており、すべての CAD プログラムで認識できますが、少々難解なテキスト形式のファイルです。描画エディタにより作成された DXF ファイルには、**Mesh** のプログラムに特有の基盤メッシュについての情報が含まれません。
- MIN ファイルは、**Mesh** が認識できるテキストフォーマットで記述されています。このフォーマットは DXF ファイルよりも便利な構造になっており、MIN ファイルを編集して小さな変更を加えることが比較的容易になっています。MIN ファイルは、基盤メッシュの仕様を含め、メッシュ生成に必要なすべての情報を含んでいます。

基盤メッシュの詳細を設定したあと、それまでの作業の結果を消失しないようにするため、**Mesh** スクリプトを作成しておく必要があります。「DRAWING」→「Export MIN file」コマンドを用いて、描画ベクトルと基盤メッシュの現在の状態についての情報を記録しておくことができます。デフォルトのファイル名(拡張子を除く)を用いるか、異なるファイル名を用いて、任意の名前でファイルを保存してください。描画エディタを終了するとき、**Mesh** は情報を保存するかどうか問い合わせてきます。「EXIT」→「Save」を選択すると、プログラムはファイル「SPARKGAP.MIN」に情報を記録します。

メインメニューに戻ったら、「FILE」→「Edit script」→「Edit script, text」コマンドを選んで、出来上がった MIN ファイルの内容を確認してみてください。プログラムは「SPARKGAP.MIN」を内蔵テキストエディタにロードします。ファイルの構文規則に関しては、マニュアル第 6 章で説明します。詳細な知識がなくても、ファイルの主な特徴はわかります。始めの「GLOBAL」セクションには、3つの水平方向に一樣なゾーンおよび、2つの垂直方向に一樣なゾーンを含めた基盤メッシュ情報が入っています。その後、7つの REGION(領域)セクションがあります。各セクションには、輪郭を構成する直線と円弧の集合が記述されています。直線を定義するデータ入力形式は次の形式です。

L XStart YStart XEnd YEnd

また円弧を定義するデータ入力形式には 6 つの値があります。

A XStart YStart XEnd YEnd XCenter YCenter

第4章 描画エディタのメニューとコマンド

4.1 メインメニューから描画(drawing)エディタの起動

第3章の例を用いた説明において、**Mesh** の描画エディタの多くの特徴について述べました。本章では、すべての機能について述べます。まず始めに **Mesh** のメインメニュー(FILE メニュー)にあるコマンドを使って描画 (drawing) エディタを呼び出してから、スクリプトを新たに作成したり、すでに作成したスクリプトを編集したりすることにします。復習すると、描画(drawing)は、各領域の形状を定義する一セットのベクトル (点、直線、円弧) を含みます。描画は DXF フォーマットで保存でき、ほかの CAD ソフト用にエクスポート(書き出し)できます。スクリプト (script) には、**Mesh** により認識される簡単なフォーマットの描画ベクトルのほかに、基盤メッシュの特性などの詳細な情報、また画像を読み込む高度なコマンドも含まれています。描画とスクリプト間でフォーマット変換するには、いくつかのオプションがあります。

「FILE」→「Create script」→「Create script, Graphics」

描画を新たに始めるときに、このコマンドを用いてください。ダイアログにおいて、拡張しを除くファイル名 (file prefix) を (1~46 の英文字の長さで) 入力してください。このファイル名(ここでは"FPrefix"とします)は、描画ファイル (FPrefix.DXF) およびスクリプトファイル (FPrefix.MIN) に保存するときにデフォルトで適用されます。ダイアログで入力する空間的範囲は、四角い解析領域の大きさを定義します。解析空間はこの四角形を完全に占めることも、占めないこともあります。

「FILE」→「Edit script」→「Edit script, Graphics」

すでに作成してある **Mesh** スクリプトを変更するとき、または改良スクリプトを作成するとき、このコマンドが有効です。FILE/Load script コマンド (5.2 節参照) を用いて **Mesh** にスクリプトをロード (読み込み) すると、このコマンドが使用可能になります。**Mesh** は、(基盤メッシュの情報を含めた) スクリプト情報を翻訳し、描画エディタを開きます。

「FILE」→「Create script」→「Create script, DXF import」

このコマンドは次の2つの機能を果たします。1) DXF フォーマットで保存された **Mesh** の描画の再ロード、2) ほかの CAD プログラムからの描画をインポート。プログラムは情報をロード

して、描画エディタを起動します。

Mesh は、ほかのプログラムから DXF ファイルをインポートすると、点 (Point)、直線 (Line)、円弧 (Arc)、円 (Circle) およびポリライン (polyline: 線、円弧などのセグメントの連続により定まる 1 つ実体) の型を持つエンティティ (entity: 1 単位としてグラフィック操作が可能な表示要素の集まりである実体) を認識します。たとえばテキストなど、上述した以外のすべてのエンティティは無視します。プログラムは円を 4 つの円弧に分離し、ポリラインを個々の線ベクトルに分離します。プログラムはポリラインの直線部分のみを認識し、円弧、スプライン、またはそのほかの複雑な形状などを無視します。解析空間の四角形領域の大きさは、有効なエンティティの x と y (または z と r) の最大値および最小値により決まります。エンティティが領域として確実に指定されるようにするため、描画に「1」、「2」、「3」、...という名前を付与したレイヤー (Layer) を作成する必要があります。付与できるレイヤー名の最大数は「250」です。**Mesh** は、描画の Layer 1 のすべての有効なエンティティを Region 1 に、Layer 2 のすべての有効なエンティティを Region 2 に、以下同様に各 Layer に各 Region を指定していきます。認識されないエンティティ型を無視し、そのほかの Layer において有効なすべてのエンティティはレイアウト領域 (Region 0) に指定します。(注記: 過去において、AutoDesk およびそのほかの会社は通知することなく、DXF フォーマットを変えてきました。したがって、各 CAD プログラムの将来の新しいバージョンに対して、**Mesh** が翻訳できることを保証できません。2 次元のフリーウェアとして良い CAD パッケージである QCAD は **Mesh** と互換性があり、Field Precision 社のリソース・ダウンロードサイト <http://www.fieldp.com/fplinks.html> から入手できます。)

4.2 作業環境

本節では、描画エディタの作業環境の一般的な特徴について、図 17 を参考にしながら説明します。コマンドはすべて、画面上端のメニューからアクセスできます。ツールバーには、よく用いるコマンドの項目があります。これらのコマンド項目にマウスカーソルを置くと機能が表示されます。画面の主要部分は、描画エリアと情報エリアに分かれています。右側にある情報エリアに、領域の総数、ベクトルの総数、および x と y 方向の表示格子間隔が一覧表示されています。この情報エリアには、領域の色分け、領域の Filled 属性の状態も表示されます。

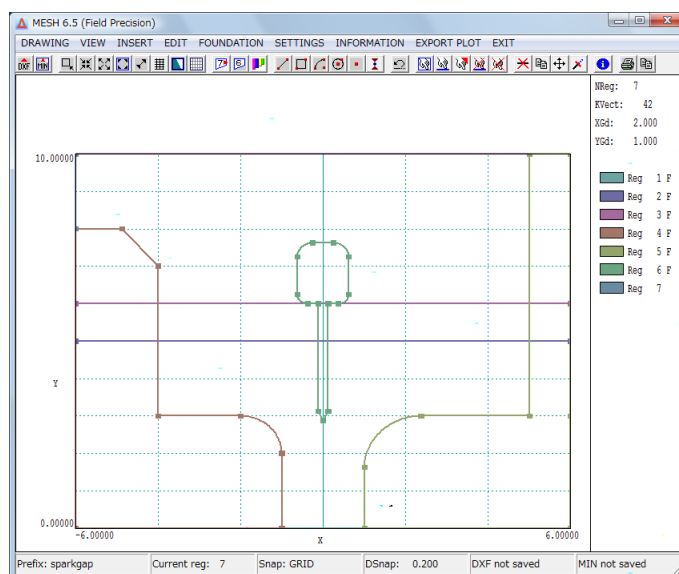


図 17 Mesh 描画エディタの作業環境

画面下端のステータスバーは、2つのモードで表示します。標準モードでは、保存した DXF ファイルおよび MIN ファイルの(拡張子を除く)ファイル名(Prefix)、ベクトル入力の際の現在領域番号 (Current region: 入力中のベクトルに付与される領域属性)、スナップ(吸着)モード (Snap)、スナップグリッドの間隔(DSnap)、および DXF ファイルおよび MIN をエクスポートするための保存状態がステータスバーに表示されます。プログラムがマウスの位置情報を必要とするとき、ステータスバーは座標モードで表示します。このとき、ステータスバーは操作のための簡単な指示、 x および y 座標、およびスナップモード(Snap mode)の状態を表示します。プログラムが座標モードであるときに「F1」キーを押すと、座標値をキーボードから直接入力できるダイアログを表示します。スナップモードに変えるときには「F2」キーを押してください。

2つのタイプのグリッド(格子)が描画エリアに適用されます。ビューグリッド (view grid) とスナップグリッド (snap grid) です。ビューグリッドは描画上に表示される一組の水平および垂直な線です。デフォルトのモードでは、プログラムは描画の形状およびズームしたビュー範囲に応じて x と y 方向のグリッド間隔を選びます。間隔の値は情報エリアに表示されます。任意の間隔を設定するときは、「SETTINGS」メニューの「Grid control」コマンドを用いてください。この場合、ビューグリッドはズーム比によって変わることはありません。スナップグリッドはマウスのグリッドスナップ (Grid snap) モードで適用されます。描画は x と y 方向で同じ間隔 (DSnap) の目に見えない線で分割されます。座標モードではマウスポインタに最も近いスナップグリッド上の座標が表示されます。ビューグリッドとスナップグリッドは、手入力パラメータを設定しない限り、必ずしも同じにはなりません。

座標入力を必要とするコマンドに適用されるスナップモードとして、No snap(吸着無し)、

Grid snap(グリッド吸着)、Endpoint snap(終点吸着)、Nearest snap(最近接点吸着)の3つがあります。ビューウィンドウの変更や、マウスポインタに最も近い物体の位置指定を必要とする操作(「SETTINGS」→「Region properties」、「EDIT」→「Select」、...)のときには、自動的にNo snap モードとなります。直線ベクトルや円弧ベクトルを定義する操作のときには、このモードは避けてください。このような操作では、終点が充填(Filled)領域の輪郭を描くように連結しなければならないからです。Grid snap モードでは、マウスによって返される座標はスナップ格子点に拘束されています。Endpoint snap モードでは、プログラムがマウスポインタに最も近いベクトルの終点の座標を返します。スナップグリッド位置で終端とならないベクトルに連結するには、このモードが便利です。Nearest snap では返される座標は、最も近接した直線または円弧ベクトルの、最も近い点の座標です。このモードはすでに存在する物体に正確に接合するベクトルを設定する際に便利です。3つのスナップモードでは、**Mesh** はマウスポインタの位置を黒い十字線で示し、吸着点(snap point)を赤い四角形で表示します。吸着点(赤い四角)はマウスの左ボタンをクリックした際に返される座標の位置を示しています。

描画の表示には vector と fill という2つのモードがあります。**Mesh** は、vector モードにおいては、領域境界を構成する直線、円弧および点を表示します。描画パーツの編集や挿入には、このモードで行います。fill モードには、充填(Filled)領域が囲むエリアを色付けして表示します。このモードでは、ベクトルを入力したり編集するコマンドの多くが使えません。fill モードには、以下のようないくつかの用途があります。

- プレゼンテーションや論文用のイラスト作成。
- 領域を生成する順番が正しく上書きしていることのチェック。たとえば、図 10 において、電極が気体、セラミック、油の領域と重なる空間をどのように上書きするか確認してください。
- 輪郭のベクトルが連結し、閉じた形状を形成していることの確認。

フィールド(Filled)領域を定義するとき、ベクトルの入力順序は任意です。**Mesh** は、ベクトルリストにおいて、適合する終点が連結するように、付け加えられるごとにベクトルをソートし、並べ替えます。実際、ある領域の一部を入力し、他のパーツの作業を行ってから、再び戻ってその領域を完成することができます。fill モードの制約の1つは、輪郭が完全に定義されるまで、領域が正しく表示されないことです。

(「INSERT」→「Add line(s)」、「EDIT」→「Delete selection」など)いくつかのベクトル操作では、コマンドを再入力することなく、複数個の入力ができるようになっています。これらのコマンドが入力されると、**Mesh** は座標入力モードに入り、メニューコマンドの多くが使

用できなくなります。ユーザーはいくつかのベクトルを定義する点を入力できます。別のコマンドを使いたいときには、マウスの右ボタンを押して、このモードを抜け出てください。一例として、DSnap=0.250 であり、次の 2 つの直線ベクトルを生成したいとします。

L 1.500 0.750 1.600 0.893

L 1.600 0.893 2.000 1.000

まずコマンド「INSERT」→「Add line(s)」を選んでください。次に、(1.500, 0.750) 近くの点にマウスを移動し、左ボタンを押してください。「F1」キーを押して、x=1.600 および y=0.893 の値をキー入力し、「OK」ボタンをクリックして、最初のベクトル生成を完了します。プログラムは二番目のベクトルの始点を入力できる状態になっています。「F2」キーを押して、「Endpoint」ラジオボタンを起動し、「OK」ボタンをクリックしてください。最初のベクトルの終点近くにマウスカーソルを移動し、左ボタンをクリックしてください。再び、「F2」キーを押して Grid snap モードに戻ってください。マウスカーソルを点 (2.000, 1.000) 近くに移動して、左ボタンをクリックしてください。最後に、右ボタンをクリックして、座標入力モードを抜け出てください。

描画エディタを終わらせる際には、次の 3 つのオプションがあります。

- **Save**：現在のファイル名(拡張子を除く)が「FPrefix」であるとき、「FPrefix.MIN」という名前で **Mesh** スクリプトを作成するか、または上書きします。
- **Save as**：スクリプトの名前を付けるために、新たなファイル名を入力します。このオプションは、名前を変更して新たなファイルを作成しても、もとの MIN ファイルを残しておきたいときに使えます。
- **Abandon**：スクリプトの新たな作成も、上書きもしないで描画エディタを抜け出ます。DXF ファイルを保存してあるなら、描画パーツを再生し、あとになってから **Mesh** スクリプトを作成できます。この場合、基盤メッシュの特別な設定はすべて失われることに留意してください。

任意の挿入操作または編集操作を行う前に、描画エディタは現在作成・編集したパーツ群を「TEMP.DXF」というファイルにコピーします。このファイルは、「Edit」→「Undo last operation」コマンドで用いられます。また「FILE」→「Create script」→「Create script, DXF import」コマンドを用いて、このファイルをロードし、失われた作業を回復できます。

4.3 「DRAWING」(描画)メニュー

「DRAWING」メニューのコマンドは、情報をエクスポート(書き出し)およびインポート(読み込み)します。重要なコマンドはツールバーのボタンまたは右クリックのメニューから呼び出すこともできます。

Export DXF file

ファイル「FPrefix(ファイル名).DXF」における描画パーツを記録します。ダイアログで、現在のファイルのファイル名(拡張子を除く)を用いるか、代わりのファイル名を入力してください。このファイルは、ほとんどの CAD プログラムで読み取れます。基盤メッシュに関する情報は含まれていないことに留意してください。

Inport DXF entities

DXF ファイルに含まれている(直線、ポリライン、円などの)有効なパーツを編集中の描画に付け加えるコマンドです。メインメニューのコマンド「FILE」→「Create script」→「DXF import」とは対照的に、このコマンドが解析空間の四角い範囲を変更することはありません。従って、すべてのオブジェクト(物体)がこの境界線の中に入っていることを確かめる必要があります。プログラムは自動的にクリッピング(範囲外の部分の切り落とし)を行うことをしません。「1」という名前のレイヤーのパーツが Region 1 に加えられ、「2」レイヤーのパーツが Region 2 に加えられるなど、同様に以下の各レイヤーのパーツが各領域に加えられます。その他のほかの任意のレイヤーのパーツは、レイアウト用領域である Region 0 に指定されます。編集コマンドを用いることにより、これらのパーツをほかの領域にコピーしたり移動したりすることができます。

Export MIN file

描画パーツおよび基盤メッシュに関する完全な情報を含む、**Mesh** スクリプト(.MIN ファイル)を書き込みます。描画エディタを抜け出るときにスクリプトを保存するオプションもあります。

Change drawing limits

解析領域の拡大または縮小を行います。縮小する場合、すべてのベクトルは縮小した場合の範囲に納まっている必要があります。**Mesh** はすべての基盤メッシュの設定を消去し、デフォルトの値を設定します。

Dimension list

設計の際に有用な、描画ベクトルの大きさのリストを作成します。このファイルは、「FPrefix(ファイル名).LST」という名前で保存されます。

4.4 「VIEW」(ビュー)メニュー

このメニューのコマンドは、画面表示とハードコピーのプロット範囲を変えます。

Zoom window

表示するボックスの対角の 2 点をマウスまたはキーボード入力で設定して、表示範囲をクローズアップ (拡大) します。このコマンドを実行すると、座標入力モードが有効になりスナップモードは一時的に No snap(スナップ無し)になります。

Zoom in

プロット図の中央点を中心として拡大表示し、表示範囲を狭めます。

Expand view

プロット図の中央点を中心として縮小表示し、表示範囲を拡大します。

Global view

表示範囲を、全解析領域を表示するように拡げます。

Pan

プロット図の中心を移動します。マウスまたはキーボード入力で、移動を表す変位ベクトルの 2 点を指定することによって設定します。

Toggle FILL display

ベクトルモードと Fill モードの間で表示モードを切り替えるコマンドです。Fill モードにおいては、Filled(充填)と指定した領域の閉じた多角形内部の画素を設定するアルゴリズムが用いられます。オープン (Open) 領域は輪郭としてのみ表示されます。Filled(充填)領域のベクトルが連結した閉じた形状でときには、Filled 領域はねじれた形に表示されます。Filled 領域が現れない場合は、Filled 領域がほかの領域により上書きされている可能性があります。この場合、「SETTINGS」→「Region order」コマンドをクリックして、領域リストの順序を変えてくだ

さい。

4.5 「INSERT」(挿入)メニュー

このメニューのコマンドは、描画ベクトルを作成し、それらを各領域にまとめるときに用います。次の2つのコマンドは、ベクトルを力する領域を選択します。

Start next region

描画パーツを新しい領域(region)に入力するときに用います。新たなスクリプトを作成したときは、プログラムは作業領域を Region 1 に、また既存のスクリプトを編集するときには最も大きな番号を持つ番号を作業領域(current region)に設定します。このコマンドは描画に新たな領域を加え、その領域を現在の作業領域に設定します。このコマンドの実行をし忘れてベクトルを加えたときには、あとからベクトルの領域属性を変更できます。

Set current region

すでに定義されている領域にベクトルを加える際に、その領域に移ります。指定した領域番号が定義されていないとき、**Mesh** は最大の領域番号の領域を作業領域(current region)として設定します。

直線、円弧および点などの描画パーツを加えるときには、次のコマンドを用います。

Add line(s)

作業領域(current region)に1つ以上の直線ベクトルを作るときに用います。このとき、プログラムは任意の数の直線の始点と終点を入力可能な、座標入力モードに入ります。このモードを抜け出るときには、マウスの右ボタンをクリックしてください。キーボードから座標を入力するには「F1」キーを、スナップモードを変えるには「F2」キーを押してください。

Add rectangle

4つの連結した直線ベクトルからなる、四角形を作ります。四角形を定義するためにボックスの対角を、マウスまたはキーボードにより設定してください。

Add arc(s) → Add arc: Start-end-center

マウスまたはキーボードをによって始点、終点、中心点を設定し、1つ以上の円弧を作成します。

これらの点の設定が矛盾しているとき、プログラムはエラーメッセージを出します。円弧は正方向および負方向のどちらの回転の向きにも定義できます。**Mesh** は Filled(充填)領域において、すべてのベクトルが連結した、一つの形状を構成するように円弧を配置します。

Add arc(s) → Add arc: Start-end-radius

マウスまたはキーボードを用いて始点と終点を設定し、ダイアログで半径の値を入力して円弧を入力します。次に、回転の向きを決めるため、円弧の中心の相対位置 (U: 上方/右側または L: 下方/左側) をダイアログに入力します。

Add arc(s) → Add arc: Start-center-angle

マウスまたはキーボードを用いて始点と円弧の中心点を設定し、円弧を入力します。角度を度の単位でダイアログに入力します。角度の値が正であるときは、反時計回りの方向の円弧になります。

Add circle → Circle: Center-point

円の中心点と円周上の 1 点を、マウスまたはキーボードを用いて設定し円を作成します。**Mesh** は円周角が 90 度の円弧ベクトルを 4 つ作ります。

Add circle → Circle: Center-radius

円の中心点をマウスまたはキーボードを用いて設定し円を作成します。半径の値をダイアログに入力してください。**Mesh** は円周角が 90 度の円弧ベクトルを 4 つ作ります。

Add point(s)

1 つ以上の点をマウスまたはキーボードを用いて設定します。点はオープン (Open) 領域にのみ現れます。多くの場合、整列した細いワイヤの集まりを表現するときに点が用いられます。

4.6 「EDIT」(編集)メニュー

「EDIT」メニューのコマンドは、描画ベクトルの寸法および特性を変更するときに用います。

Undo last operation

ファイル「TEMP.DXF」を再ロードして、直前の操作を取り消すコマンドです。座標入力モードで追加された複数のベクトルは取り除かれることに留意してください。

多くの編集操作は一つのベクトルもしくは一セットの複数ベクトルに対して行います。編集操作を行う前に、操作対象のセットを定義するために、複数の選択操作もしくは選択解除操作を行うことができます。ロックされた領域や不可視になっている領域のベクトルは選択できないことに注意して下さい。

Select → Select window

四角い領域内に終点があるすべてのベクトルを選択するコマンドです。マウスの左クリックで四角い領域の頂点の一つを選択し、さらにもう一つの頂点を再度左クリックすることにより選択します。選択されたベクトルはハイライトされます。

Select → Sselect vector(s)

マウスをベクトルに近づけて左クリックすることにより、一つ以上のベクトルを選択します。右クリックすることによって選択モードを抜けられます。

Select → Select region

ある一つの領域に含まれるベクトルのいずれかに、マウスを近づけて左クリックすることによって、その領域内のすべてのベクトルを選択します。

Deselect → Deselect vector(s)

マウスを近づけて左クリックすることにより、選択された状態の一セットのベクトルから一つ以上のベクトルを除外します。右クリックすることによって選択モードを抜けられます。

Deselect → Deselect all

選択された状態の一セットのベクトルすべてを選択解除します。

Delete Selection

選択した範囲のすべてのベクトルを消去します。消去したい各ベクトルにマウスを近づけ、左ボタンをクリックしてください。2つのベクトルが同じように共有境界上にあるとき、最も大きな領域番号を持つベクトルの方が取り除かれます。ある領域内にあるベクトルが消去されないようにするには、選択操作前に「SETTINGS」→「Region properties」コマンドにより表示されたダイアログの「Locked」チェックボックスにチェックを入れてください。すべての編集コマンド同様に、Mesh はベクトルを領域毎にまとめ、各領域のベクトルを終点が適合するように配置します。

Copy selection

選択したベクトルのセットを複製します。新しいベクトルのセットは同じ場所に同じ領域番号で作成されます。複製した新しいベクトルのセットは選択状態になっていますので、移動させたり他の領域に移したりすることができます。

Move selection

選択した(複数の)ベクトルを移動させます。移動する変位を定義するために、**Mesh** は始点と終点を指定するように指示します。他の座標入力モードの場合と同様、キーボードから座標を入力するには「F1」キーを、スナップモードを変えるには「F2」キーを押してください。

Rotate selection

選択した(複数の)ベクトルを回転させます。領域をマウスまたはキーボードにより選択してから、回転の原点を定義してください。そして回転角を（反時計回りの回転を正として）度の単位で入力してください。

「Move region」および「Rotate region」コマンドに関して、描画エディタはクリッピングを自動的に行わないことに注意して下さい。変位後のパーツのベクトルが全解析範囲である四角形領域の中に納まっていることを確かめる必要があります。スクリプト(.MIN ファイル)中に無効なベクトルがあると、**Mesh** はメインメニューの「PROCESS」コマンド実行中にエラーを通知します。

Transfer selection to current region

選択した全ベクトルの領域番号(region number)を現在の作業領域(current region)に変更します。例えば、ある領域のあるパーツを現在の領域にコピーしたいとします。まずコピーしたいパーツを「Select」コマンドや「Deselect」コマンドを使って選択し、その後「Copy selection」コマンドでそれらを複製し、このコマンドで現在の作業領域(current region)に移動します。

Split vector → Split vector(s): midpoint

ベクトルを中点で 2 分割します。直線または円弧ベクトルにマウスポインタを近づけ、左クリックします。そのベクトルは中点で 2 つのベクトルに分割されます。

Split vector → Split vector(s): closest

ベクトルをマウスポインタに最も近い点で 2 分割します。直線または円弧ベクトルにマウスポ

インタを近づけ、左クリックします。そのベクトルはマウスポインタに最も近い点で2つのベクトルに分割されます。

Trim vector(s)

切取り線 (trim line) の所で、直線または円弧ベクトルを終端させるベクトルです。レイアウト領域(region 0)に切取り線を加えるか、すでにある任意の直線ベクトルを切取り線として用いることができます。切取り線の近くにマウスカースルを移動し、左ボタンをクリックしてください。そして、切り取る 1 つ以上のベクトルを指定してください。除去すべき側のベクトルの終点の近くにマウスカースルを移動し、左ボタンをクリックしてください。

Fillet

接続された 2 つのベクトルの間の角 (edge) を、指定した半径円弧によって滑らかに結合するフィレットのコマンドです。まず後述の「Fillet/Chamfer width」コマンドを用いて、半径を指定してください。次に接続された 2 つのベクトルをマウスで指定してください。描かれる円弧は指定した半径を持ち、2 つの直線に接します。この条件を満たすことができないときには、プログラムはエラーを通知します。

Chamfer

接続された 2 つのベクトルの間の角 (edge) を、切断して指定した長さのベベル (斜角部) で結合する面取り (chamfer) のコマンドです。まず後述の「Fillet/Chamfer width」コマンドを用いて、ベベルの長さを指定してください。次に接続された 2 つのベクトルをマウスで指定してください。

Fillet/chamfer width

ダイアログで値を入力し、以降のフィレット (fillet) 半径および面取り (chamfer) 幅を設定します。

4.7 「FOUNDATION」 (基盤メッシュ) メニュー

このメニューのコマンドは、基盤メッシュを修正するときに用います。基盤メッシュ情報はメッシュスクリプト (.MIN ファイル) に含まれますが、描画ファイル (DXF) には保存されません。基盤メッシュが表示されていないと、多くのコマンドが使えません。

Foundation display

基盤メッシュの表示を切り換えるコマンドです。灰色の垂直線から 3 角形要素のおよその底辺の長さ、水平線からおよその高さがわかります。基盤メッシュが表示されているとき、基盤メッシュコマンドは使えますが、ベクトル編集のコマンドは使えません。

次のコマンドは、 x/z 軸(「X/Z axis」、水平方向)と y/r 軸(「Y/R axis」垂直方向)のどちらにも適用できます。

Divide zone (X/Z, Y/R)

新たにスクリプトを作成するとき、**Mesh** はデフォルトの基盤メッシュを作成します。この基盤メッシュは、 x および y の各軸に沿っておよそ 100~150 の要素からなる、ただ 1 つの要素(element)サイズの区域(zone)からなっています。細部の構造を再現するために、 x と y の各軸に沿って、要素の大きさが異なる区域に分割することが、実用上便利ながよくあります。このコマンドを用いて、すでに作成された区域を 2 つに分割できます。分割しようとする区域の中にマウスを移動し、左ボタンをクリックすると、選択した区域が強調表示されます。次に、分割する位置までマウスカーソルを移動し、再び左ボタンをクリックしてください。プロット図は太い線で 2 つの区域の境界が示されたプロット図に更新されます。

Shift zone boundary (X/Z, Y/R)

すでに設定されている、要素サイズの異なる区域(zone)間の境界の位置を調整します。調整しようとする境界の近くにマウスポインタを置いて、左ボタンをクリックしてください。次に、隣接する区域内の、新たな境界を設定したい位置にマウスを移動し、再び左ボタンをクリックしてください。

Change element size (X/Z, Y/R)

すでに作成した区域の要素(element)の大きさを変更します。要素の大きさを変更しようとする区域の中にマウスポインタを移動し、左ボタンをクリックしてください。このとき、その区域内にある要素の現在の底辺と高さがダイアログに表示されます。新たな値を入力し、「OK」ボタンをクリックしてください。

Undo last change (X/Z, Y/R)

直前に行った基盤メッシュ操作を取り消すコマンドです。

4.8 「SETTINGS」(設定)メニュー

Region properties

図 9 に示すようなダイアログを表示させ、領域特性(Region properties)を設定します。各領域について、チェックボックスを持つ Visible(可視)、Locked(ロック)および Filled(充填)という 3 つの欄があります。Visible 属性は、その領域のベクトルを描画エディタに表示させるか否かを決めます。Visible でない領域は、MIN ファイル、DXF ファイルには記録されています。また Locked 属性を設定することによって、ある領域のベクトルを変更させないようにしたり、任意のマウスによる選択操作の対象として選ばれることのないようにします。Filled 属性は、その領域が MIN ファイルで Filled という識別子(designator)が付けられることを指定します。Filled になっていると、領域番号(region number)が設定された要素を内部に持つ、閉じた境界で囲まれた領域であることになります。デフォルトでは、Region 1 は Filled になっていて、そのほかの領域(region)は始め Open になっています。最後の欄(Title)には分かり易い名前を入力します。このボックスの内側をクリックしてから、名称を入力してください。この名称には、スペースを含めた **Mesh** の標準区切り記号を使わないようにしてください。たとえば、「TRIGGER_PIN」は有効ですが、「TRIGGER PIN」は無効です。

Region order

2.3 節で述べたように、**Mesh** スクリプトの領域の順序(region order)は、領域間で共有された部分がどちらの領域によって上書きされるかを決定します。この順序はメッシュから導かれる計算結果に重大な影響を与えることがあります。このコマンドにより、図 18 のダイアログが表示され、スクリプトに記録される順序を修正できるようになります。順序を変えたい領域の右側にあるチェックボックスをチェックしてください。次に、「Move UP」または「Move DOWN」

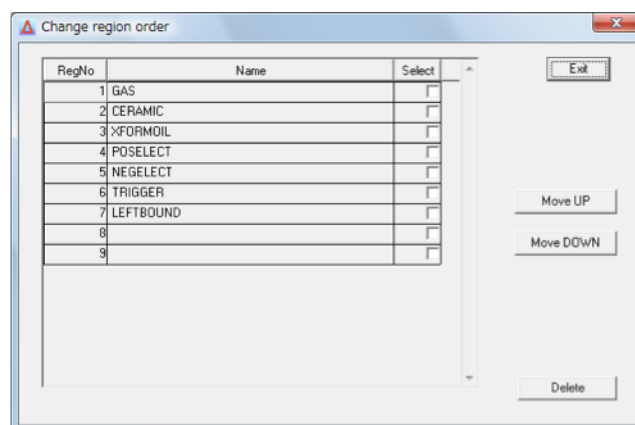


図 18 スクリプトの領域の順序を変更するダイアログ

ボタンを何回か押して、チェックした領域の他の領域に対する順位を変えてください。領域を削除するときは、その領域のチェックボックスをチェックし、「Delete」ボタンをクリックしてください。

Grid control

このコマンドをクリックすると、ビューグリッドの設定を変更するダイアログが表示されます（この設定はスナップグリッドには影響しません）。グリッド表示ボックスをチェックしたり、外したりして、グリッドの表示と非表示を切り換えてください。「Automatic intervals」のチェックボックスがチェックされていると、**Mesh** は解析領域の大きさとズームの倍率を基にして、グリッド線間の間隔を適切に選びます。このチェックを外すと、水平線と垂直線の間隔として任意の値が入力できるようになります。この場合、これらの間隔は表示ズーム倍率によって変わることはありません。XGrid=YGrid=DSnap となるように設定すると、グリッドスナップモードでパーツを定義できます。

Snap control

このコマンドにより、マウスのスナップ形式を選択する図 12 のようなダイアログが表示されます。「No snap」オプションでは、プログラムは現在のマウス位置に最も近い座標値を返します。「Grid」オプションでは、プログラムは距離「Snap distance」(DSnap)だけ離れた節点からなる正方形メッシュの最も近い位置を返します。このモードが有効なときには、DSnap の値を「Snap distance」ボックスに入力して設定できます。「Endpoint」モードでは、プログラムはカーソル位置に最も近いベクトルの終点の位置を返します。

TOGGLE MAGNETIC SNAP DISPLAY

スナップ点の表示／非表示を切り替えます。

TOGGLE NUMBER FORMAT

Mesh は、解析領域の寸法から座標表示に浮動小数点表記を使うか否かを決定します。数値の表示フォーマットを切り換えたいとき、このコマンドを用いてください。

4.9 「INFORMATION」(情報)メニュー

Vector information

描画ベクトルのタイプ、領域番号および終点の座標などの特性を表示させるコマンドです。特

性を表示したいベクトルの近くまでマウスカースルを移動し、左ボタンをクリックしてください。

Region information

領域のパラメータであるタイプ、ベクトルの数、Filled 属性のステータスを表示させるコマンドです。パラメータを表示したい領域の任意のベクトルの近くまでマウスポインタを移動し、左ボタンをクリックしてください。

4.10 「EXPORT PLOT」(プロット書き出し)メニュー

Mesh および TriComp 解析プログラムにより作成したすべての表示画面をプリンタやプロットファイルに書き出すとき、以下のコマンドを用います。

Default printer

現在作業中のプロット図 (current plot) のコピーを、Windows にデフォルト設定したプリンタ (通常使うプリンタ) に送るコマンドです。いくつかのプリンタをお持ちの場合、デフォルトのプリンタを変えるには、Mesh を実行する前に Windows のコントロールパネルなどから「通常使うプリンタ」を選択してください。

Graphics file → EPS, BMP, PNG

現在作業中のプロット図のグラフィックファイルを EPS(Encapsulated PostScript)ファイル、BMP(Windows Bitmap)ファイル、PNG(Portable Network Graphics)ファイルの各フォーマットで作成するときに用いるコマンドです。プログラムはファイル名(拡張子除く)の入力を求めます。グラフィックファイルはカレントディレクトリに、[ファイル名].EPS、[ファイル名].BMP、[ファイル名].PNG という形式の名前で作成されます。プロットファイルは、ユーザーが要求する解像度または鮮明さを満たさない場合があります。その場合、高解像度のディスプレイをお持ちのときは、PaintShop Pro のような市販のプログラムのスクリーンキャプチャ機能を用いるのが有効な方法の一つです。そうしたプログラムでは、イメージを編集し、テキスト文を書き込み、JPEG のような圧縮フォーマットにエクスポートすることができます。

Copy to clipboard

現在作業しているプロットを Windows のクリップボードにコピー (Windows Metafile フォー

マット) します。コピーしてから、ほかのアプリケーションに貼り付けることができます。

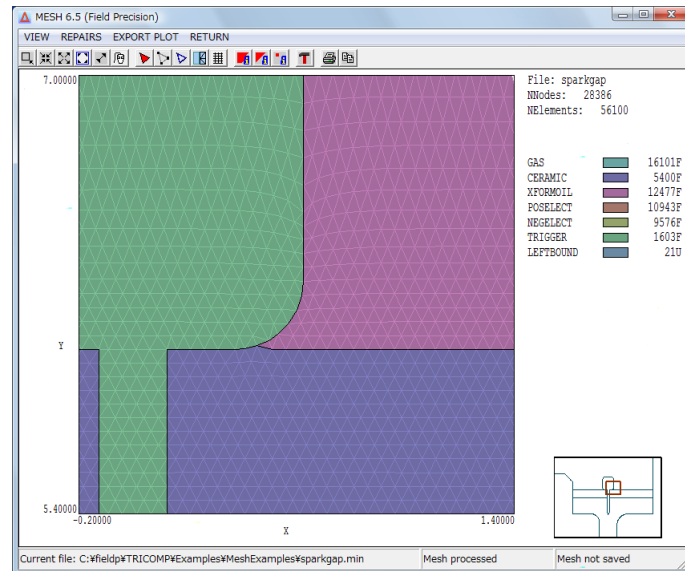


図 19 「PLOT-REPAIR」メニュー。SparkGap の例題の拡大表示

第5章 スクリプトからのメッシュの生成

5.1 メッシュの生成

これまでの章では、描画エディタによる **Mesh** スクリプトの作成方法について説明しました。スクリプトとは、解析空間内のオブジェクトの輪郭を定義するベクトルの集まりを含むテキストファイルです。本章では、スクリプトからメッシュを生成する方法について説明します。メッシュという用語は、有限要素法解析のために用いる節点(node)の座標と要素(element)単位のリストを意味します。節点の座標と要素単位は、FPrefix.MOU という形の名前を持つテキストファイルに記録されています。

第3章で描画エディタにより作成したスパークギャップ(SparkGap)の例題を用いて、スクリプトからメッシュを生成する手順を概説します。「EXIT」→「Save」コマンドによりプログラムを終了させず、描画エディタを起動しているなら、スクリプトはすでにロードされていますし、メッシュ生成(process)を行う準備もできています。もう1つの方法は、TC から **Mesh** を実行させ、メインメニューのコマンド「FILE」→「Load」→「Load script (MIN)」を選択してください。ダイアログが出たら、「SparkGap.MIN」を選びます。次に、「PROCESS」コマンドまたは対応するボタン(緑色の四角いアイコン)をクリックしてください。プログラムは自動的

にスクリプトの内容を解析し、メッシュを生成します。エラーが出なければ、「PLOT-REPAIR」のメニューおよびボタンが使用できる状態になります。エラーが発生したときには、例題ライブラリ(「Examples」フォルダ内の「MeshExamples」フォルダ)にある「SparkGap.MIN」のコピーをとりあえず用いてください。あとの章で、エラーを訂正する方法について説明します。

コマンド「FILE」→「View listing file (MLS)」を選択してください。**Mesh** は「SparkGap.MLS」を内部エディタにロードします。このファイルには、問題の診断に役立つ詳細な処理情報が含まれています。エディタを終了させ、「PLOT-REPAIR」コマンドまたは対応するボタンをクリックしてください。図 19 は、トリガ電極の一部を拡大した図とともに、**Mesh** のプロットメニューの作業環境を示しています。描画時に与えられた領域名は保存され、**Mesh** が作成した形状適合要素が、非常に正確に境界に沿って並んでいることが分かります。

5.2 「FILE」、「PROCESS」、「HELP」メニュー

Mesh を実行させる方法は、1) ウィンドウとして動作する対話的プログラムとして、2) バックグラウンドにで動作する自律的プログラムとして、という 2 通りの方法があります。始めは対話モードについて説明します。5.5 節で、バッチファイル制御のもとでデータを大量に処理するときに用いられる自律モードを説明します。

TC プログラムランチャーから **Mesh** を実行させた場合、プログラムは以前のセッションと同じディレクトリ位置で、また同じ画面サイズで起動します。画面は空白になっていて、いくつかのコマンドが使用不可になっています。「FILE」メニューの次のコマンドは初期状態から使用可能です。

Creat script → Creat script, Graphics

スクリプトを作成するために描画エディタを起動します。この処理については、3.1 節で述べました。

Creat script → Creat script, DXF import

描画エディタまたはほかの CAD プログラムを用いて作成した DXF ファイル読み込んで、描画パーツを定義し、また描画エディタを起動します。4.1 節において、使用可能なパーツと、それらを領域に割り当てる際の規則について説明しました。四角い解析領域の境界は、描画パーツの最大範囲により決まります。

Creat script → Creat script, text

メッシュスクリプト(.MIN ファイル)をテキストフォーマットで直接作成するときに用います。第 6 章において、スクリプトのフォーマットについて詳細に説明します。プログラムは、グラフィックスのオプションで用いたのと同じダイアログを表示します。ファイル名(拡張子不要)と解析領域の各座標両端の値を入力してください。内蔵テキストエディタは、各コマンドおよびオプションの一覧を含むテンプレートを表示します。テンプレートには、Global と Region という基本セクションがあり、使用できるコマンドがコメント行としてリストされています。

Load → Load script (MIN)

メッシュ生成処理やプロット表示のために、すでに作成された入力スクリプトをロードします。ダイアログは[ファイル名].MIN という形式の名前を持つファイルのリストを表示します。ダイアログでディレクトリを変えると、プログラムの作業ディレクトリも変わります。この場合、出力ファイル(.MOU ファイル)は、変えられた新たなディレクトリに書き込まれます。ステータスバーは、現在開いているスクリプトの名前(Current file)を表示します。

Load → Load mesh (MOU)

すでに作成(process)したメッシュファイルをロードしてプロット図を作成したり、小さな修正(repair)を行うときに用います (5.4 節参照)。処理されたメッシュにおけるオブジェクトの形状は固定されていて、変更できません。

Edit file

内部エディタを用いて、任意のテキストファイルの内容を見たり、修正したりするときに用います。ダイアログでディレクトリを変えても、プログラムの作業ディレクトリは変わりません。

次に示すコマンドは、スクリプトがロードされたときに使用可能になります。

Edit script → Edit script, graphics

MIN ファイルを描画図形に変換し、描画エディタを起動します。このエディタ内で図形を変更でき、また図形の変更を反映したスクリプトを前と同じ名前または別の名前で保存できます。

Edit script → Edit script, text

スクリプト(MIN ファイル)を内蔵テキストエディタにロードし、そこで直接変更したり追加したりすることができます。エディタを終了する前に、行った作業内容を同じ名前または別の名前で必ず保存してください。

Transform mesh

このコマンドを用いて、空間内でメッシュを移動するか、サイズを変える（倍率変更）ために、現在処理中 (current) のメッシュスクリプトを変更します。**Mesh** は図 20 に示すようなダイアログを表示します。メッシュを移動する値 x_{shift} および y_{shift} (または z_{shift} および r_{shift})、またはメッシュの倍率を変更する値 M (Magnification) を入力してください。次式のように座標が変わります。

$$x_{\text{new}} = Mx_{\text{old}} + x_{\text{shift}}, \quad (3)$$

$$y_{\text{new}} = My_{\text{old}} + y_{\text{shift}}. \quad (4)$$

ダイアログを終了するときに、**Mesh** は保存するファイルのファイル名(拡張子除く)の入力を促します。

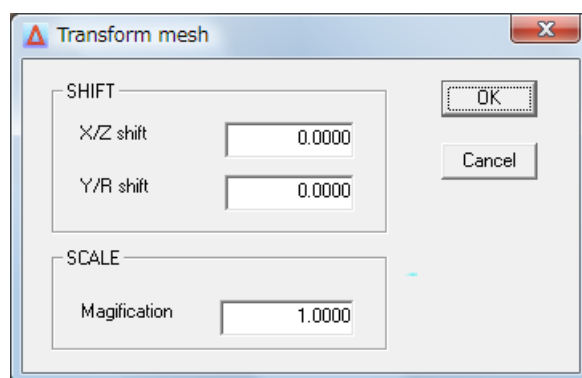


図 20 メッシュ変換ダイアログ

PROCESS

このコマンドは、メインメニューの見出し項目の 1 つになっています。メッシュ生成プロセスを開始します。**Mesh** は入力スクリプトを解析し、基盤メッシュを生成し、領域境界に適合させます。オペレーションに関する情報は、リストファイル ([ファイル名].MLS) に記録されます。メッシュ生成が成功したとき、ステータスバーは成功を通知します(「Mesh processed」)。

次の 2 つのコマンドは、メッシュスクリプトのメッシュ生成プロセス後に使用可能になります。

Save mesh (MOU)

7.6 節で述べるフォーマットでメッシュ生成処理が終わったメッシュを保存します。出力ファイ

ル名は、[ファイル名].MOU という形式です。このコマンドは、メッシュ生成に成功したときのみ使用可能になります。**Mesh** を終了させる場合や、新たなスクリプトをロードするとき、プログラムは現在のメッシュを保存するよう促します。

View listing file (MLS)

エディタを開いて、現在使用中のリストファイル(.MLS ファイル)を表示します。エラーの箇所があると、カーソルはファイル中のその箇所に配置されます。

最後に、「HELP」メニューには次の 2 つのコマンドがあります。

Mesh manual

本マニュアルの **Mesh** を、ユーザーのデフォルト PDF ブラウザに表示します。このとき、ファイル mesh.pdf が mesh.exe と同じディレクトリになければなりません。

Clean directory → Partial

Clean directory → Full

多数のデータファイルが長時間の作業中に蓄積することがあります。このコマンドは、作業ディレクトリから不要なファイルを消去します。オプションを「Partial」にすると、**Mesh** はすべてのプログラムから TEMP.DXF およびリストファイル (*.?LS) を消去します。オプションを「Full」にすると、**Mesh** および **TriComp** の解析プログラム（ソルバー）に関する、上記のファイルと出力ファイル (*.?OU) を消去します。このオペレーションは、解析を再構築するために必要な、入力ファイルを消去することはありません。

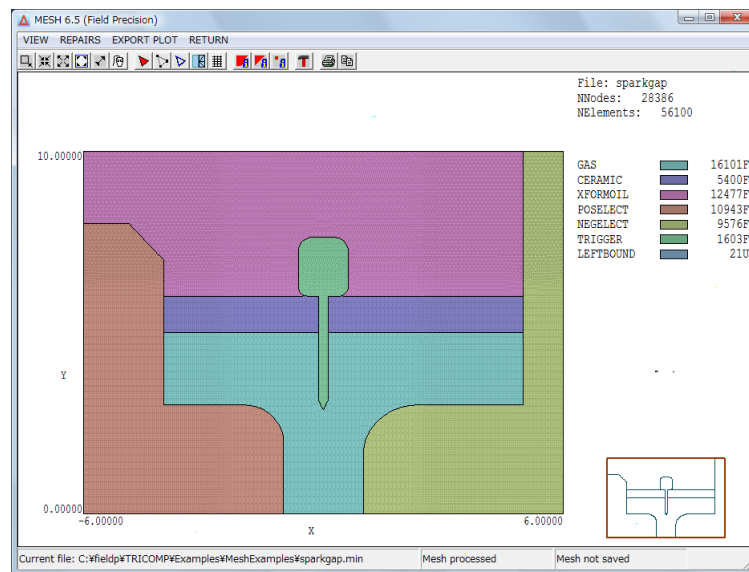


図 21 「PLOT-REPAIR」の作業環境

5.3 「PLOT-REPAIR」メニュー — メッシュ表示画面

メインメニューの「PLOT-REPAIR」コマンドは、メッシュの生成処理が終了したときに使用可能になります。このコマンドは、図 21 に示すようにメニュー、ツールバー、およびプロットの各エリアを表示します。プロットエリアは次の 3 つの部分からなります。

- **メインプロット。**この部分に、メッシュの拡大・縮小したプロット図が表示されます。
- **凡例。**画面右側の部分に、メッシュにおける節点(node)および要素(element)の数および領域(region)の色分けリストが表示されます。色分けリストの右側端の数値情報は、充填 (Filled) 領域 (F) の要素数および非充填 (Unfilled) 領域 (U) の要素数です。
- **配置。**右下の縮小イメージは、各領域の境界と現在の拡大・縮小プロット範囲を示すビューウィンドウです。

「VIEW」、「REPAIRS」、「EXPORT PLOT」という 3 つのメニューがあります。「EXPORT PLOT」メニューのコマンドは、描画エディタと同じ機能を持ちます (4.10 節参照)。同様に、表示エリアを制御する次のコマンドの機能については、4.4 節で説明しました。

Zoom window

Zoom in

Expand view

Global view

Pan

Grid control

Toggle number format

ビューウィンドウの設定にはマウスを用いることができ、またマウスは「REPAIRS」メニューのコマンドで重要な役割を果たします。描画エディタとは異なり、マウスは常に No snap(スナップ無し)または Grid snap(グリッドスナップ)モードで機能します。マウスの動作は次のコマンドにより制御できます。

Mouse control → On/Off

Mouse control → Toggle snap mode

Mouse control → Set DSnap

On/Off オプションは、座標入力をマウスで行うか、キーボードで行うかの指定です。Toggle snap mode は、No snap と Grid snap の間の切り替えです。描画エディタと同様、「DSnap」という量は、正方形スナップグリッドの間隔です。

Plot type → Plot type: Elements

Plot type → Plot type: Nodes

Plot type → Plot type: Facets

プロットタイプ「Elements」は、領域番号により色分けした要素による表示で、境界ファセット表示のオプションがあります。このプロットは番号が正しく充填 (Filled) 領域に割り当てられているかをチェックするときに役立ちます。「Nodes」プロットは、節点(node)を領域番号により色分けします。このプロットは領域番号が共有境界に沿って正しく割り当てられたかをチェックするときに役立ちます。最後の「Facets」プロットは、要素形状をチェックするのに役立ちます。

Toggle facet display

このコマンドは、Element タイププロットにおける、要素境界の表示のオンオフの切り替えをします。

5.4 「PLOT-REPAIR」メニュー — 「REPAIRS」(修復)

メッシュ作成処理中に発生するエラーが4種類あります。

1. スクリプトの構文エラー(Syntax error)。「XMesh」と「YMesh」コマンドにおける間隔がマッチしない場合や、充填 (Filled) 領域の境界が接続されていない部分がある場合などのエラーを含みます。
2. 直線または円弧を表すベクトルの始点と終点間の境界パス (経路) を見つけられない場合。
3. 反転要素がある場合。
4. 節点または領域に割り当てた領域番号の間違い。

始めの2種のエラーは重大であり、メッシュの生成処理を停止します。通常、1つ目のタイプのエラーは、「Edit script, text」コマンドを用いて直ちに訂正できます。エディタはエラーが発生した行にカーソルを表示して起動します。2番目のエラーは、基盤メッシュが、1つ以上の領域境界を構成する直線または円弧に対し、不適当に位置しているときに発生します。3番目と4番目のエラーは、入力スクリプト(.MIN ファイル)を変えるか、「REPAIRS」メニューのコマンドを用いて、作成したメッシュファイルにおける節点(node)または要素(element)を直接訂正することにより修正できます。しかしながら最初に述べた、入力スクリプトを編集して修正する方が望ましいことです。なぜなら、メッシュ生成を再度行った場合にも修正が適用されるからです。メッシュファイルの直接修正は、難しい状況にあるとき、慎重に用いるべきです。

次のコマンドは、処理したメッシュについての情報を表示します。

Information → Nodes

マウスポインタを情報を知りたい節点(node)の近くに移動し、左ボタンをクリックしてください。指定した節点が強調表示され、座標(x, y)、インデックス番号(k, l)、領域番号(Region number)、および領域名(Region name)の情報がダイアログに表示されます。「OK」ボタンをクリックして、操作を継続してください。

Information → Elements

マウスポインタを情報を知りたい要素(element)の近くに移動し、左ボタンをクリックしてください。指定した要素が強調表示され、重心座標(x, y)、要素面積(Area)、領域番号(Region number)、および領域名(Region name)の情報がダイアログに表示されます。

Information → Regions

マウスポインタを情報を知りたい領域の近くに移動し、左ボタンをクリックしてください。指定した領域が強調表示され、領域番号(Region number)、ステータス(Region status, Filled また

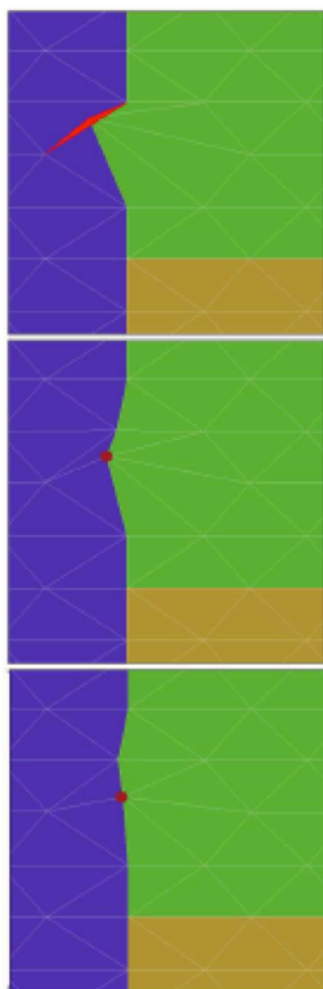


図 22 要素の修復。上図：反転要素を赤色で表示。中央図：「Relax bad nodes」コマンドを適用した後のメッシュの状態（処理した節点を明確にするため強調表示）。下図：「Move node」コマンドを用いた更なる修正

は Open)、領域面積(Area)、および領域名(Region name)の情報がダイアログに表示されます。

次のコマンドは、簡単には修正できないエラーがいくつかあるときに役立ちます。修正したあとにメッシュを必ずセーブし、今後の解析のためにファイル「[ファイル名].MOU」を保存してください。始めの3つのコマンドは、反転要素を修正するときに用います。

Relax bad nodes

反転要素があると、**Mesh** はエラーメッセージを表示し、Element プロットの Facet 表示モードで、その要素に伴う節点を赤色でマークします (図 22 上図)。Element モードにして、ビューを拡大して赤くマークされた部分を表示させてください。「Relax bad nodes」コマンドを1回以上クリックしてください。通常、この操作によって不良要素は修正されます (図 22、中央図)。次に、「Move node」コマンドを用いると、修復を完了します。

Relax nodes in box

場合によっては、メッシュの問題を解決するために、不良要素を囲むそのほかの節点を緩和 (relax) させる必要があります。このコマンドを選択したあと、マウスを用いて赤くマークされた部分を囲むボックスを2つの対角点を決めて作成します。プログラムは囲まれた節点を特定し、それらの節点の位置を緩和します。このコマンドを1回以上用いたあと、「Move node」コマンドに移り、修復を完了させてください。

Move nodes

このコマンドを用いて、メッシュの各節点を移動して境界を修正できます (図 22、下図)。マウスポインタを移動したい節点に近づけ、左ボタンをクリックしてください。望ましい新たな位置までマウスを移動し、左ボタンをクリックしてください。マウスの右ボタンをクリックすると、操作を中断します。

次のコマンドは、領域番号指定の誤りを訂正します。これらのコマンドを、「Change region number」コマンドとしてグループ化しています。

Single node

Node タイプのプロットを用いて、正しくない領域番号を持つ節点(node)を見出すことができます。このコマンドは、各節点の属性を変更します。マウスを用いて、正したい節点を指定して

ください。現在の領域番号を表示するダイアログが現れます。新しい領域番号を入力し、「OK」ボタンをクリックしてください。

Nodes in box

このコマンドは「Single node」コマンドと似ていますが、マウスを用いてボックスの対角点をクリックすることにより、メッシュの特定領域の外形を描いて指定するところに違いがあります。ボックス内部のすべての節点を変更されます。

Single element

このコマンドを用いて、単一要素の領域番号を変更できます。マウスポインタを要素の内部に置き、左ボタンをクリックしてください。

Elements in box

このコマンドは「Single node」コマンドと似ていますが、マウスを用いてボックスの対角点をクリックすることにより、メッシュの一部を指定するところに違いがあります。ボックス内部に重心を置くすべての要素が変更されます。

5.5 Mesh の自律的実行

バッチファイル制御のもとで、コマンドプロンプトから **Mesh** を実行できます。このモードでは、長時間にわたる一連の操作をコンピュータが実行するように設定できます。プログラムを呼び出して、拡張子を除くファイル名を入力すると、バックグラウンドモードで **Mesh** が作動します。たとえば、コマンドプロンプトから

```
\TRICOMP\MESH \DATA\IFACE <Enter>
```

と入力したとします。**Mesh** が起動し、現在のまたは指定したディレクトリにあるファイル IFACE.MIN を探します。ファイルを見つけると、プログラムはバックグラウンドで実行します。一連の幾何形状に関して、メッシュを生成し、静電解を求めるバッチファイルは、次のようになります。

```
REM Variation of focus electrode postion
REM Processing FELEC01
START \TRICOMP\MESH \GUNDESIGN\FELEC01\FELEC01
START \TRICOMP\ESTAT \GUNDESIGN\FELEC01\FELEC01
REM Processing FELEC02
START\TRICOMP\MESH \GUNDESIGN\FELEC02\FELEC02
START\TRICOMP\ESTAT \GUNDESIGN\FELEC02\FELEC02
REM Processing FELEC03
START \TRICOMP\MESH \GUNDESIGN\FELEC03\FELEC03
START \TRICOMP\ESTAT \GUNDESIGN\FELEC03\FELEC03
REM Job completed
```

第 6 章 Mesh スクリプトのフォーマット

6.1 スクリプトファイルの構造

Mesh スクリプト(.MIN ファイル)は、任意のエディタを用いて作成し、修正することができるテキストファイルです。描画エディタにより作成した図形から、このファイルを作成できます(第 3 章および第 4 章)。ファイルは「[ファイル名].MIN」という形式の名前を持ちます。ファイル名の文字列[ファイル名]は、1 から 32 文字までの長さの名前が可能です。

Mesh スクリプトの構造は、次のようになります。

```
Global
  (Global commands)
End

Region [RegName1]
  (Boundary vectors)
End

Region [RegName2]
  (Boundary vectors)
End

...

Region [RegNameN]
  (Boundary vectors)
End

EndFile
```

まず解析全体に影響を及ぼすコマンドで構成する「Global」セクションが一つあり、そして解析しようとする物理的対象物それぞれについての幾何情報を記述する「Region」セクションが最大 250 まであります。「Global」セクションはファイルの始めに置かなければなりません。

「Global」セクションの機能は、プログラムパラメータと基盤メッシュの特性を設定することです。「Global」セクションの入力の終わりは、「End」コマンドにより示します。「Global」セクションにおいて、コマンドの入力順序は問題ではありません。**Mesh** は「End」コマンドが現れるまでコマンドを読み続け、それから操作を実行します。「Region」コマンドは領域境界情報の始まりを示し、この情報は「End」コマンドまで続きます。最後に、すべての領域について入力を終えたことを「EndFile」で示します。

Mesh は自由構文解析ツールを用いてコマンドを読み込みます。スクリプトの 1 行は、コマンドといくつかの区切り記号により分けられた、1 つ以上のパラメータからなります。区切り記号として認識する記号は、次の記号です。

- スペース 「 」
- コンマ 「,」
- タブ
- コロン 「:」
- 左括弧 「(」
- 右括弧 「)」
- 等号 「=」

これらのどの記号も区切り記号として使うことができ、特色のあるスタイルで入力ファイルを作成できます。読みやすくするために、インデント(字下げ)することもできます。コマンドとキーワードは、大文字でも小文字でも入力できます。**Mesh** は、空白行とコメント(注釈)行を読み飛ばします。コメント行は、文字「*」(アスタリスク)から始まり、そのあとにコメントする文字が続きます。実数パラメータは、以下のどのフォーマットでも入力できます。

```
2.3456
2.63E12
-1.95E+0.2
5
```

最後の数「5」は、5.0 と解釈されます。「EndFile」コマンドのうしろに、テキスト文を必要なだけ記述できます。この場合は、アスタリスク(*)は必要ありません。完全なスクリプトの実例を表 3 に示します。

表 3: Mesh スクリプトの例

```
* File DTQUAD.MIN
GLOBAL
  XMesh
    0.000E+00  5.000E+00  9.999999E-02
  End
  YMesh
    0.000E+00  5.000E+00  9.999999E-02
  End
END
REGION FILL Vacuum
* Vacuum
  L 0.00E+00 0.00E+00 5.00E+00 0.00E+00
  L 5.00E+00 0.00E+00 5.00E+00 5.00E+00
  L 5.00E+00 5.00E+00 0.00E+00 5.00E+00
  L 0.00E+00 5.00E+00 0.00E+00 0.00E+00
END
REGION FILL Dielectric
* Dielectric
  L 0.00E+00 5.00E+00 1.00E+00 5.00E+00
  L 1.00E+00 5.00E+00 1.00E+00 2.00E+00
  L 1.00E+00 2.00E+00 0.00E+00 2.00E+00
  L 0.00E+00 2.00E+00 0.00E+00 5.00E+00
END
REGION FILL Electrode_Up
* Upper electrode
  L 0.00E+00 2.00E+00 1.75E+00 2.00E+00
  L 0.00E+00 1.00E+00 1.75E+00 1.00E+00
  A 1.75E+00 1.00E+00 2.25E+00 1.50E+00 1.75E+00 1.50E+00
  A 2.25E+00 1.50E+00 1.75E+00 2.00E+00 1.75E+00 1.50E+00
  L 0.00E+00 1.00E+00 0.00E+00 2.00E+00
END
REGION Gnd_Bound
* Grounded boundary
  L 0.00E+00 0.00E+00 5.00E+00 0.00E+00
  L 5.00E+00 0.00E+00 5.00E+00 5.00E+00
  L 5.00E+00 5.00E+00 0.00E+00 5.00E+00
END
ENDFILE
```

6.2 基盤メッシュの基本的定義

Global セクションにおける XMesh と YMesh コマンド(または ZMesh と RMesh コマンド)は、基盤メッシュの配置を指定します。これらのコマンドは、解析領域の範囲を設定し、各軸方向の3角形要素の大きさを設定します。XMesh と YMesh は直交座標に、また ZMesh と RMesh

は、円筒座標系に用います。XMesh および ZMesh コマンドは、水平軸方向の要素サイズ区域のリストが続くことを示します。End コマンドは、そのリストが終わることを示します。本節では、データ行が一行だけの簡単な形式について考えましょう。

XMesh

-6.00 10.00 0.25

End

データ行のパラメータは、直角座標系の問題では

Xmin Xmax Dx

であり、円筒座標系の問題では、ZMesh のデータラインは

Zmin Zmax Dz

という形になります。この例のデータでは、解析領域の x 軸方向における範囲が -6.00 から 10.00 までであり、3 角形要素の底辺のおよその大きさが 0.25 単位であることを指定しています。中程度の複雑さの幾何形状では、 x 軸方向に要素数が約 $100 \sim 200$ となるように Dx を選びます。 $x_{\max} > x_{\min}$ である限り、任意の範囲で、これらの値を指定することができます。

同様に、YMesh および RMesh では、縦軸方向のメッシュの特性を設定します。

YMesh

3.25 10.95 0.10

End

この場合、データ行のパラメータは

Ymin Ymax Dy (直交座標系の場合)

Rmin Rmax Dr (円筒座標系の場合)

となります。

円筒座標系の問題で $r_{\min} < 0$ になる値は、物理的には無意味な結果を与えます。このとき **Mesh** は r の値が負であるというエラーメッセージを出して停止します。

TriComp のプログラムでは、空間的な寸法については柔軟に対応できます。 $x_{\max}$ や $y_{\min}$ のような空間量を、都合の良い任意の単位で入力できます。(同一ファイル中では単位は同じである必要があります。) 解析プログラム(ソルバー)の実行の際に、**Mesh** により与えられた座標値を

SI 単位 (m) に変換する換算係数を入力できます。**Mesh** に入力する際は、数値が理解しやすく且つ精度が良くなるように、数字の大きさが大体 1 のオーダーになるように単位を選んでください。例として、センチメートル、ミクロン、マイルなどがあります。

註:**Mesh** の出力ファイル(.MOU)中のデータはインプットの際に XMesh-YMesh コマンドまたは ZMesh-RMesh コマンドのいずれを使った場合にも変わりません。これらの区別はユーザーが分かり易いようにするためだけに設けられています。これらの区別は **Mesh** のプロットメニューや描画エディタにおいて使われます。入力ファイル(スクリプト)中に ZMesh および RMesh コマンドがあった場合、縦方向の座標が 0.0 より小さい場合は、**Mesh** はエラーを出して止まります。

6.3 領域の定義

Mesh が「Global」コマンドを処理し、基盤メッシュを設定し終わると、幾何形状を定義する、一連の Region(領域)セクションの処理を行う準備が整います。領域定義の始まりを示す「Region」コマンドには、次の 2 つの形式があります。

Region

Region Fill

このあとに、24 文字以内の参照可能な領域名を付けて、スクリプトを分かりやすくすることもできます。

Region Fill UpperPlate

Mesh で使える (スペースを含む)区切り記号は、領域名には使用できないことに留意してください。したがって「Focus_electrode」は有効ですが、「Focus electrode」は無効です。

Mesh は、スクリプトに領域(region)が現れる順に番号を付けていきます。節点(node)およびそれらに囲まれた要素は、形状適合処理の後、領域番号が割り当てられます。プログラムは、領域番号(region number)と領域名(region name)のリストをコメント行の形式でリストファイル(.MLS)に作成します。ユーザーは、下記のようにこの情報をコピーして解析ファイルの入力スクリプトに貼り付け、領域番号とその物理的特性との関連情報として記録することができます。


```

Assignment of region numbers
Number of regions in the file: 6
Region  Region
number  name
=====
*      1  VACUUM
*      2  ANODE
*      3  CATH_SUPPORT
*      4  FOCUS_ELEC
*      5  CHAMBER_WALL
*      6  CATHODE_SURF

```

「Fill」(充填)というキーワードは、その領域が満たされていることを指定します。**Mesh** は境界ベクトルを処理し、並べ替え、境界ベクトルが閉曲線を構成するかチェックしたあと、閉曲線内部に含まれる節点と要素に、現在対象としている(current)領域番号を割り当てます。

完全な領域セクションの一例は次のような形になります。

```

Region Fill Upper_Pad
XShift: 2.5
YShift: 0.2
* Upper pad, 1.0 microamp source
  A -5.0 30.0 0.0 35.0 0.0 30.0
  A 0.0 35.0 5.0 30.0 0.0 30.0
  A 5.0 30.0 0.0 25.0 0.0 30.0
  A 0.0 25.0 -5.0 30.0 0.0 30.0
End

```

「Region」セクションの構成は、1) 「Region」コマンド、2) 任意の行数からなるコメント行、3) 「XShift」、「YShift」、「Rotate」コマンド、4) 境界を定義する 1 つ以上の円弧、直線、点ベクトル、および 5) 「End」ステートメント、からなります。以下の節で、点、線、および円弧の入力の仕方を説明します。

「XShift」と「YShift」コマンドの構文は、次のようになります。

```

XShift xs
YShift ys
ZShift zs
RShift rs

```

これらのコマンドは、その領域に属するすべての点、直線、円弧のデータを次式のように変換

します。

$$x(\text{program}) = x(\text{file}) + xs, \quad (5)$$

$$y(\text{program}) = y(\text{file}) + ys. \quad (6)$$

これらのコマンドは、幾何形状が繰り返し構造であるときに有効です。1つ以上の領域に関するベクトルデータ行を単にコピーして貼り付け、「XShift」および「YShift」コマンドを用いて配置させることができます。**Mesh**では、自動的にクリッピング(計算領域外部分の切り落とし)を行わないことに留意してください。移動した点とともに領域が解析空間の外に出ると、エラーメッセージを出します。

「Rotate」コマンドは次のような形式を持ちます。

```
Rotate Ang [xc yc]
```

```
Rotate Ang [zc rc]
```

「Ang」は度で表した回転角です。正の回転は、反時計回りの方向です。オプションのパラメータ x_c と y_c は回転の中心点を表します。デフォルト値は、 $x_c=0.0$ と $y_c=0.0$ です。「Rotate」(回転) コマンドは「Shift」(移動) コマンドと同じく、領域セクションの中のすべてのベクトルに作用します。プログラムは自動的にクリッピングを行いません。回転は移動の前に実行されることに留意してください。

6.4 点

点のデータ行は、単一の節点(node)を設定します。点のデータ行のフォーマットは、

```
P XPoint YPoint
```

```
P 0.1678 10.45
```

です。文字「P」はベクトルの型(Point)を示しています。2つの実数パラメータ（「XPoint」および「YPoint」）は、設定しようとする節点位置の座標値（ x, y ）または（ z, r ）です（図 23）。点の位置は解析範囲の四角い領域の中になければなりません。点座標の単位は、「XMesh」および「YMesh」のステートメントと同じ単位にする必要があります。ゼロ次元である点は本質的

に孤立しているので、充填 (Filled) 領域の境界の定義に用いてはいけません。一つの領域内に 2000 までの点を含めることができます。点はオープン (Open) 領域内の直線および円弧と接続することができます。静電解析では、点の配列は格子をシミュレートするときに便利です。

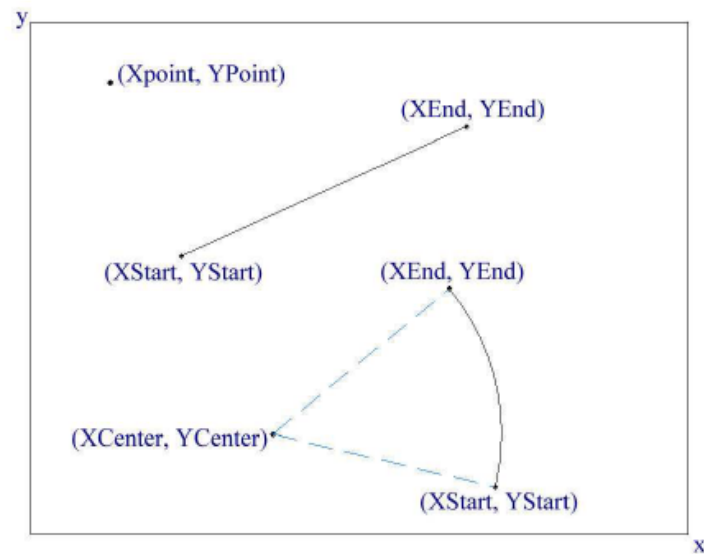


図 23 点、直線、円弧を定義するパラメータ

6.5 直線

次のコマンドは直線を定義しています。

```
L XStart YStart XEnd YEnd
L 0.571 0.986 1.756 -0.234
```

4 つの実数パラメータは、始点と終点の座標です (図 23)。直線は解析範囲内に納まっていないければなりません。解析プログラムが円筒座標系の問題を処理するとき、これらのパラメータは、次の量として解釈されます。

$$XStart \rightarrow ZStart, \quad (7)$$

$$XEnd \rightarrow ZEnd, \quad (8)$$

$$YStart \rightarrow RStart, \quad (9)$$

$$YEnd \rightarrow REnd. \quad (10)$$

6.6 円弧

次のステートメントにより円弧を設定します。

```
A XStart YStart XEnd YEnd XCenter YCenter  
A 0.500 1.000 1.000 0.500 0.500 0.500
```

6つの実数値は始点、終点、中心点の座標です。図 23 に各点の定義を示します。円弧の始点と終点が解析範囲の外側にあると **Mesh** はメッセージを表示しますが、円弧上のすべての点が妥当であるかについてのチェックは行いません。

ある始点-終点-中心の座標の組に対し、正の角と負の角という2つの円弧が可能です。**Mesh** は常に 180° 以下の広がり角の方

を選びます。 180° と 360° の角度範囲に広がる円弧を設定するには、2 つ以上の円弧を用いてください。

6.7 充填 (Filled) 領域の境界

オープン (Open) 領域のベクトルは連結されている必要がありません。 対照的に、充填 (Filled) 領域の境界ベクトルは閉じた連続の曲線を形成する必要があります。その終点は、小さな許容距離の範囲内で始点と一致する必要があります。ベクトルは任意の順序で入力できます。**Mesh** にはベクトルの始点が、以前に設定したベクトルの終点と一致するように、ベクトルを再配列する強力な機能があります。プログラムがキーワード「Fill」を検出すると、始点を選んで境界の周りを動き、終点が一致するベクトルを探し出し、必要なときには始点と終点を逆にします。修正されたベクトルの組は、リストファイル「[ファイル名].MLS」に記録されます。充填 (Filled) 領域のベクトルの終点が以前に設定したベクトルの始点に戻らないとき、**Mesh** はエラーメッセージを出します。

第 7 章 Mesh の高度な技法

7.1 可変サイズ要素分割

要素サイズが一様なメッシュは、ほとんどの計算に適しています。その一方で、要素サイズを変化させる (variable resolution) と解析精度が上がり、計算時間が短縮できることがあります。可変サイズ要素分割は、以下の 2 つの場合に特に有効です。

- 大きな解析空間の中にある小さな構造物近くの場合を正確に調べたい時
- 周囲に無限に広がる空間がある場合の境界条件を、粗い要素からなる広い領域を設定することによって再現したい時

多くの有限要素法プログラムとは異なり、**TriComp** は構造化メッシュ (structured mesh) を用います。構造化メッシュという用語は、要素全体が規則的な配列構造を持ち、インデックス操作により隣り合う要素と節点の識別が可能であることを意味します。構造化の利点は、膨大な検索処理を伴うプログラムの処理を高速化することです。欠点については、構造化の論理的結合が可変サイズ要素分割に制約を課すことがあることです。

X 軸と Y 軸 (または Z 軸と R 軸) の一方または両方に沿う要素サイズの変化は、「Global」セクションに記述する「XMesh」と「YMesh」(または「ZMesh」と「RMesh」) ステートメントに複数のデータ行を加えることにより設定することができます。次の例を考えましょう。

```
XMesh
  0.000  1.000  0.100
  1.000  2.550  0.200
  2.550  4.000  0.300
End
```

このステートメントは、 $x_{\min} = 0.00$ から $x_{\max} = 4.00$ までの要素サイズの情報です。さらに、 x 軸方向の 3 角形のサイズが、 0.00 と 1.00 の間では約 0.10 で、 1.00 と 2.55 の間では 0.20 に増加し、 2.55 と 4.00 では 0.30 になります。 x 軸方向の間隔の一連のデータ行は連続的した区域を構成し、また $x_{\min}$ から $x_{\max}$ への順でなければならないことに留意してください。YMesh、ZMesh、RMesh のデータ行も同じ形式です。

```
YMesh
  0.000  2.000  0.100
  2.000  4.000  0.250
End
```

上記の XMesh と YMesh の 2 つのコマンドの組によって作られた基盤メッシュを、図 24 に示します。

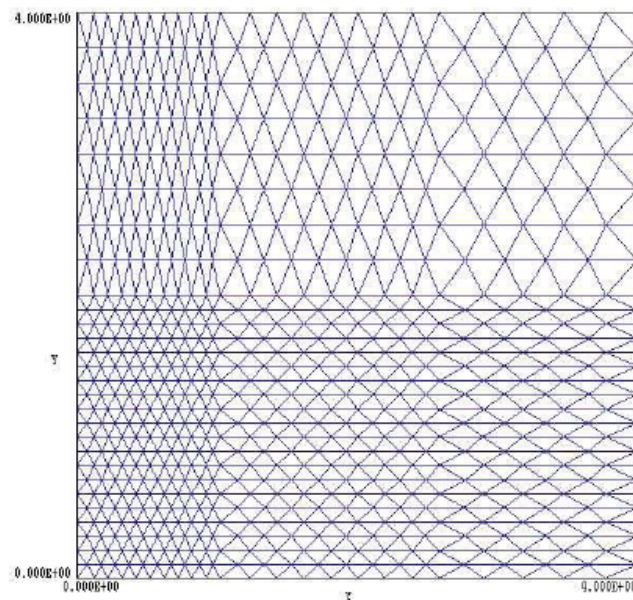


図 24 可変サイズ要素分割の例

7.2 メッシュスムージング

デフォルトの設定では、「XMesh」と「YMesh」コマンドにより各軸方向の 3 角形の要素サイズの変化は、図 24 に示すように不連続です。要素サイズの変化が連続的であると、物体形状への適合に信頼性が増し、より良い要素形状になります。「PreSmooth」コマンドにより、基本メッシュの要素サイズの変化を緩和することができます。このコマンドは「Global」セクションに次のような形式で記述されます。

PreSmooth 6

整数のパラメータは、スムージングサイクルの数です（デフォルト値は 0）。値が大きいほど、よりスムーズになります。図 24 の基本メッシュに対して 4 サイクルの事前スムージングを行った結果を図 25 に示します。

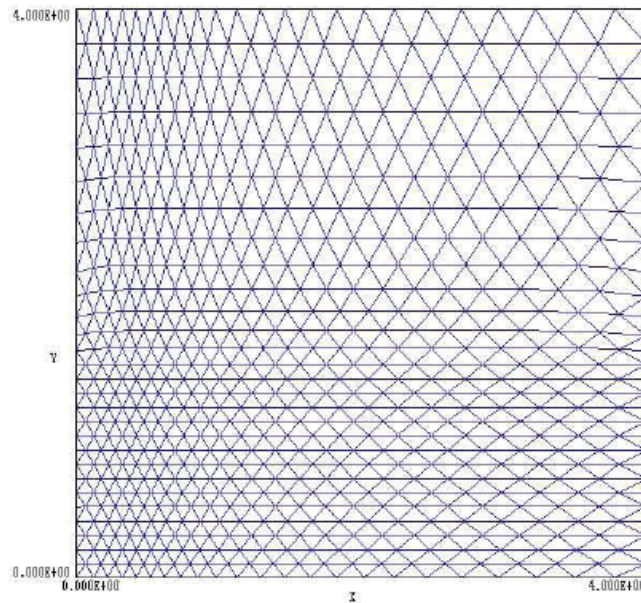


図 25 事前スムージングした可変サイズ要素分割

TriComp のソルバープログラムはすべての三角形要素が同じ形(正三角形)の場合に最も精度が良くなります。非常に小さな角度をもった鋭角三角形は数値計算時に丸め誤差が大きくなる場合があります。**Mesh** は、節点(node)を物質境界に移動した後、固定されていない節点の位置をずらす(緩和、relax)ことによって要素の形の改善を試みます。メッシュスムージング処理は次の Global コマンドによって制御されます。

SMOOTH NSmooth

整数パラメータは緩和(relax)サイクルの数です。XMesh と YMesh ステートメントにより定義された三角形要素の大きさを不連続のままにしておきたいときは、NSmooth = 0 としておいてください。デフォルト値は、NSmooth = 15 です。

スムージング有りと無しの際のメッシュを図 26 に示します。基盤メッシュ領域の端($x_{\min}$, $x_{\max}$, $y_{\min}$, $y_{\max}$)に沿って設定された節点は、この処理中によってずらされていることに注意して下さい。

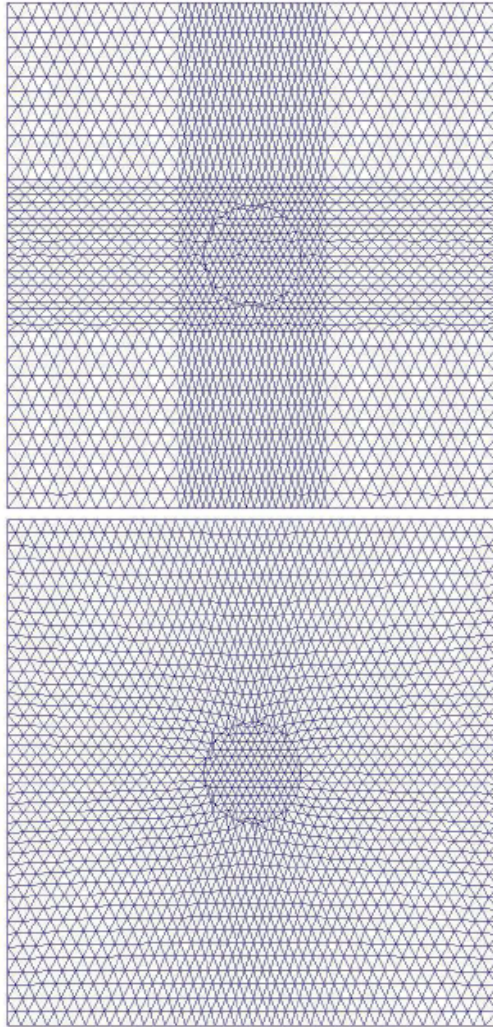


図 26 スムージング有り（上図）と無し（下図）のときの可変サイズ要素分割

7.3 基盤要素形状の設定

デフォルトでは、**Mesh** は始めに基盤メッシュを 2 等辺 3 角形で構成します。要素サイズが x 方向と y 方向で等しい場合には、要素はほぼ正 3 角形になります。正 3 角形は傾斜部または曲線部への境界適合処理の際に最良の結果をもたらします。特殊な状況においては、他の要素形状が有効になる場合もあります。Global セクションで用いられる「TriType」コマンドは三角形の形状を制御します。

TRITYPE ISO

基盤メッシュを 2 等辺 3 角形で構成します (図 27 上図)。これがデフォルトの形状です。

TRITYPE RIGHT

基盤メッシュを直角 3 角形で構成します (図 27、中央)。このオプションは、すべての境界が x 軸および y 軸に平行である系に対して「より良い」要素形状を与えます。ここで、「より良い」という言葉は、すべての要素がほとんど同じ形状になり、鋭い角を持つ 3 角形が少ないという意味です。傾斜または曲線の境界があるときには、このオプションを使わないでください。もし $NSmooth > 0$ である場合は、直角三角形は最終的に二等辺三角形に変換されてしまうことに注意して下さい。

TRITYPE GLASS [GlassAdjust]

不定形の要素による基盤メッシュを作ります (図 27、下図)。このオプションにより、**Mesh** は基盤メッシュを 2 等辺 3 角形で構成し、次にランダムに変位させてガラス状に分布させます。このオプションは、**Field Precision** 社の **KB2** 流体力学コードに適応するために組み込まれました。衝撃波のシミュレーションでは、規則的な要素境界が物質転位のようにふるまい、擬似的な要素変位を生ずる場合があります。オプションのパラメータ「GlassAdjust」は、無秩序さの度合いです。一般に、その値は 0.0 から 0.5 の間です。図 27 の例においては、GlassAdjust = 0.25 です。デフォルト値は 0.2 です。このモードは、電磁場や熱伝導のシミュレーションに対して得に利点はないことに留意してください。

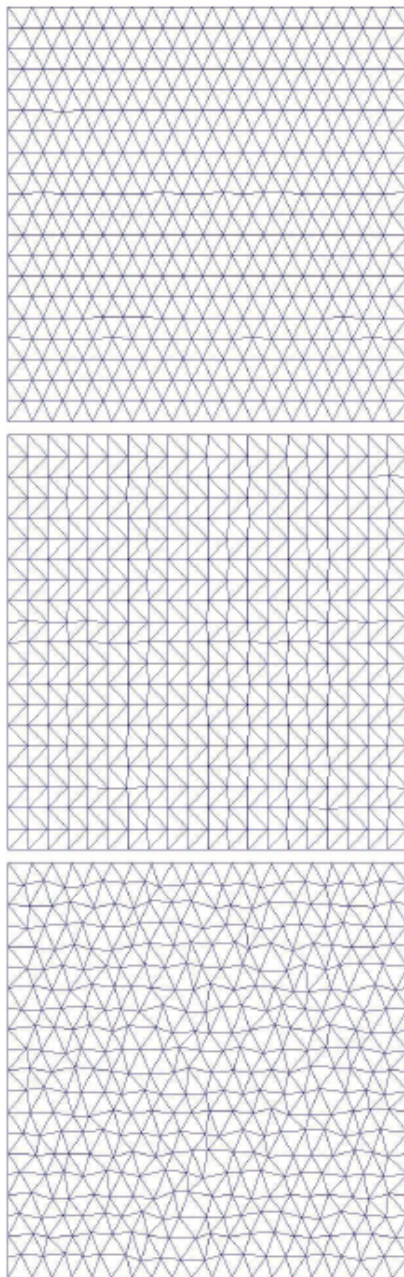


図 27 要素形状のオプション。上図：Iso、中央：Right、下図：Glass

7.4 自動補正と境界適合パラメータ

Mesh には、要素の形状適合処理により発生したエラーを補正する強力な機能があります。プログラムは、適合処理したあとに反転要素(inverted element)を検出し、その要素に接続する節点の位置を自動的に移動します。多くの場合、自動補正処理により反転要素の問題を除去できます。この処理ですべての要素を修正できない場合がまれにありますが、その場合は基盤メッシュの特性を変えたり、5.4 節で述べた修復ツールを使用して対応します。デフォルトでは、自動補正処理はオンになっています。この機能を無効にし、手動で補正したいときには、次のコマンドを使ってください。

AUTOCORRECT OFF

境界ベクトルを形状適合させる処理によりエラーが生じたときに、2 つの特別な Global コマンドが役立ちます。**Mesh** は適合処理中に節点を移動します。「Relax」コマンドのパラメータ(*Relax*)は、移動した節点に隣接する 6 つの節点の位置の移動を制御します。対象とする節点をベクトル **d** だけ変位させたとき、その隣接節点は $Relax \times \mathbf{d}$ だけ変位します。通常、この処理は良く機能し、*Relax* の値をデフォルトの 0.2 から変える必要はありません。その一方で、隣接する節点を変位させると、狭い曲がり角の場合、または同じ境界を数回たどる場合には問題が生じることがあります。*Relax* の値は、コマンドを用いて

RELAX 0.25

などとして調整できます。パラメータは、1.00 より小さい正の数でなければなりません。値を 0.00 にすると、調整機能はオフになります。

「Tolerance」コマンドは、2 つの座標位置を同じ点と認識する最大距離を設定します。すなわち、2 点の間隔が「Tolerance」のパラメータの値より小さいときに、この 2 点は同じ点と認識されます。不正確な浮動小数点入力を処理するとき、こうした同一性の近似的な判定基準が必要になります。

TOLERANCE 1.0E-5

2 つの実数座標値が同一の点であるかどうかを決める基準を設定します。デフォルトでは、**Mesh** は解析領域全体の寸法に基づいて許容値を設定します。

7.5 メッシュ生成の問題点

Mesh は 10 年以上も信頼性の向上がはかられ続けてきました。それでも、ユーザーのいかなる入力にも対応し、常に完璧に処理するメッシュ・ジェネレータを構築することは不可能なことです。基盤メッシュの 3 角形が 1 つ以上の領域の幾何形状に対して不適当である可能性は常にあります。基盤メッシュの仕様について多少の注意を払うことにより、すべての妥当な物理的幾何形状をモデル化することができます。本節では、問題の発生を回避し、診断する方法について概説します。

生成した物質境界が、計算したいシステムにまったく合致しないように見えるときには、スクリプトの入力ミスが原因であることがまず考えられます。次に、**Mesh** が境界を正しく認識し、すべてのベクトルが処理されているようでも、「Bad Triangle」というメッセージが表示されることがあります。これは、プログラムが節点を境界に適合させようとして大きく変位させた結果、いくつかの要素が反転してしまったことを意味します。まれに、自動補正処理では著しい歪みに対応できないことがあります。このような場合、計算対象のシステムの形状にもっと密接に適合するように基盤メッシュを変更すべきでしょう。反転要素の数がわずかであるときには、5.4 節で述べた **Mesh** のインタラクティブな修正機能を用いて修復できます。

不適切な要素が発生する主な原因に、次の 3 通りがあります。

- 基盤メッシュにおける要素の局所的なサイズが大きすぎて、特定のオブジェクトの幾何形状に適合できない。
- オブジェクトに鋭く尖ったエッジがある。
- 基礎メッシュの要素形状が、領域の輪郭に対して適切ではない。

最初の問題は、XMesh および YMesh コマンドを用いて要素分割数を変え、全体的または局所的に 3 角形のサイズを小さくすることにより解決できます。2 番目の問題は、最初の問題の特殊な場合です。これは、**Mesh** が鋭く尖った角の辺に沿って経路をたどるときに正しい経路を見失い、他の辺に沿って戻る、有効な節点を見出せなくなる場合です。3 番目の問題に関しては、可能な限り基盤メッシュ要素が正三角形になるような設定を試みてください。

論理経路に関するエラーが発生する可能性のある原因はいくつかあります。最も起こり得る原因はスクリプトの構文エラーで、ベクトル定義が矛盾している場合です。第 2 の可能性としては、直線または円弧のベクトルが、基盤要素を用いて分解するには小さすぎることです。こ

の場合には、小さな細部形状が物理的に意味のあることかどうか判断してください。意味がないときには、細部形状を除去してください。対象である細部形状物体が計算上重要であるときには、要素サイズを小さくするか、XMesh および YMesh コマンドにおいて可変サイズ要素分割を行ってください。第 3 の可能性としては、基盤メッシュにおける要素形状がベクトルに対して不適切であることです。極端に細長い三角形要素に円弧を適合させようとする、エラーを生じることがあります。この場合も、XMesh および YMesh コマンドにおいて要素サイズを変更して解析すべきです。最後に、オープン (Open) 領域の 2 つのベクトルが互いに交差する場合や、ベクトルの終点以外の位置で交わるとき、論理経路エラーが発生することを述べておきます。

もう少し高いレベルの問題は、メッシュが見かけ上うまく生成できていても、場の物理解が正しくないような場合です。非物理的な解が生じる最も一般的な原因は、**Mesh** スクリプトにおける領域の順序が正しくないことです。計算結果としてよくある現象としては、境界での奇妙な力線の振る舞いです。またまれな原因として、複雑な閉領域の内部において要素と節点が正しく認識されていないことがあります。この問題を診断する方法は、「Element」および「Node」タイププロット (5.3 節) を用いて、不適切な部分を見つけることです。最後になりますが、見かけ上問題ないメッシュにおいて、まれに場の局所的なひずみが見られることがあります。この問題は、ある部分における要素の形が (角度が非常に小さいなど) 適切でないときに発生します。このようなときには、「Facet」タイプのプロットを用いて問題領域を拡大表示し、問題部分におけるメッシュをチェックしてください。多くの場合、XMesh と YMesh コマンドを用いて局所的な要素サイズを少し変化させて、三角形を調整できます。また、「Repair」メニューのコマンドを用いて、節点を選んで移動し、また緩和させることもできます。

7.6 Mesh 出力ファイルのフォーマット

Meshの出力ファイル ([ファイル名].MOU) はテキストフォーマットです。この形式では、エディタ、スプレッドシートプログラム、および他の解析プログラムを用いて情報にアクセスすることが容易です。このファイルの書式は簡単なものです。表4は、このファイルの典型的な先頭部分を示します。

2行から7行までは、基盤メッシュの空間的範囲およびインデックスについての情報を与えています。FORTRANフォーマットの、[IP, E13.6] (7~19桁) および[I4] (7~11桁) が使われています。主なメッシュのデータは12~16行およびそれに続く行に記述されています。節点ご

とに1行のデータがあります。行の最初の2つの量は、節点の(k , l)インデックス（水平方向および垂直方向）です。3番目の量「RegNo」は、節点が属する領域(region)の番号です。解析空間の外側にある節点は、RegNo = 0です。構造化メッシュにおいては、各節点にはそれを囲む6つの要素があります。そして平均すると1節点あたり2つの要素があります。したがって、各節点の行は2つの要素の領域番号を含みます。便宜上、ある節点に伴う要素は、(1)右上側 (RegUp) と(2) 右下側 (RegDn) としています。解析領域外の要素は、RegUp = 0またはRegDn = 0です。最後の2つの数値パラメータは、節点の(x , y)または (z , r) 座標です。データ行のフォーマットは、[5I4, 1P, 2E14.6]です。

表4 Mesh 出力ファイル (MOU) のフォーマット

--- Run parameters ---							(1)
XMin:	0.000000E+00						(2)
XMax:	5.000000E+00						(3)
KMax:	51						(4)
YMin:	0.000000E+00						(5)
YMax:	5.000000E+00						(6)
LMax:	51						(7)
							(8)
--- Vertices ---							(9)
k	l	RgNo	RgUp	RgDn	x	y	(10)
=====							(11)
1	1	4	1	0	0.000000E+00	0.000000E+00	(12)
2	1	4	1	0	1.030236E-01	0.000000E+00	(13)
3	1	4	1	0	2.058450E-01	0.000000E+00	(14)
4	1	4	1	0	3.083962E-01	0.000000E+00	(15)
5	1	4	1	0	4.106382E-01	0.000000E+00	(16)

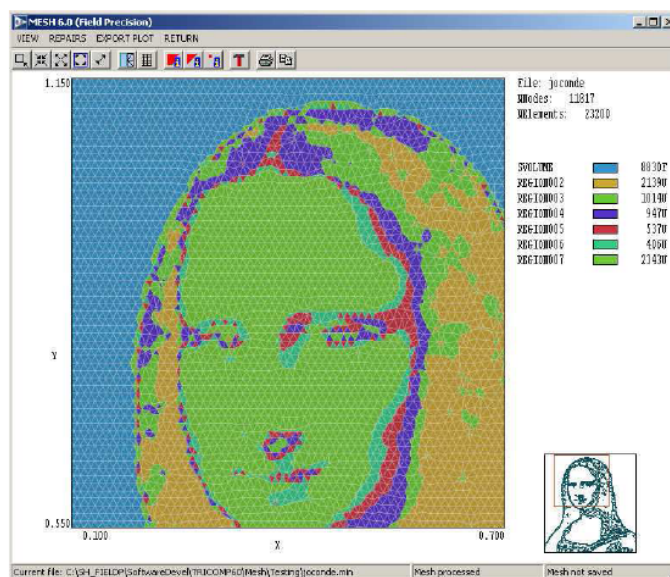


図28 Meshのスクリーンショット — イメージからのメッシュ生成例

第8章 画像からのメッシュ生成

8.1 はじめに

これまでの章で述べたメッシュ生成法は、比較的簡単な形状と明確な寸法を持つ、ほどほどの数の物体を対象としていました。このメッシュ生成法は機械的システムのシミュレーションには適切ですが、あいまいな境界を持つシステムには有効ではありません。例として、人の脚における骨の再生に、小さな電場が及ぼす効果を研究したいとします。メッシュを構成するには、脚断面の磁気共鳴画像（MRI）から電気伝導度の異なる領域（骨、筋肉組織および皮膚など）を識別することから始めるのがよいでしょう。通常の方法を用いようとする場合、領域境界の輪郭を描き、大きさを測定し、それらの結果を一連の直線と円弧のベクトルに変換することになります。生物システムには通常不規則で複雑な境界が含まれますので、処理にはかなりの作業を要します。さらに、人によってかなりのばらつきがあるので、正確に大きさを決めることに多大な苦心が払われることになります。

Mesh は画像データから形状適合メッシュを自動的に生成することができます(図 28)。この機能により、以下のような様々な種類の 2 次元画像データを 3 角形メッシュに直接変換することができます。

- 医用画像 (MRI、X 線など)
- 装置のデジタル写真
- 解剖図、設計図、地図のスキャン画像
- 温度と密度などの量の空間における連続的な分布のデータファイル

図 29 に示すのは、南太平洋のボラボラ島を標高により領域に分けたメッシュの例です。元のビットマップ・イメージを右下に示します。40,401 個の節点から成るメッシュの、要素境界を表示した詳細図を、左下の挿入図に示します。**Mesh** が画像変換に要した時間は、2 秒にもなりません。

本節の残りの部分では、**Mesh** により認識される視覚画像 (visual image) とデータ画像 (data image) という 2 種類のイメージの特徴について述べます。8.2 節では、画像を含める **Mesh** スクリプトの構造を、また 8.3 節ではビジュアルイメージを生成するコマンドと処理手順について述べます。8.4 節では、データイメージの技法を、図に即して例示します。最後に、8.5 節ではイメージ処理のコマンドと規則の概要を述べます。

Mesh における画像とは、2 次元の矩形メッシュの各節点における量の記録です。そのようなメッシュにおいて、図 30 に示すように I を水平方向、 J を垂直方向における節点の位置とすると、節点の位置をインデックス (I, J) により指定します。**Mesh** では、2 種類のイメージを扱います。

- 視覚画像 (スキャナおよびデジタルカメラにより生成した画像) は、ビットマップ・フォーマット (BMP、PCX、PNG) に変換できます。この種のイメージのメッシュは、正方形メッシュ ($\Delta x = \Delta y$) の節点で定義された値からなります。記録されている量は、節点における画素の RGB (赤-緑-青) の値です。
- データ画像は、(Δx と Δy とが必ずしも等しくない) 長方形メッシュの節点で定義された実数値からなるテキストファイルです。

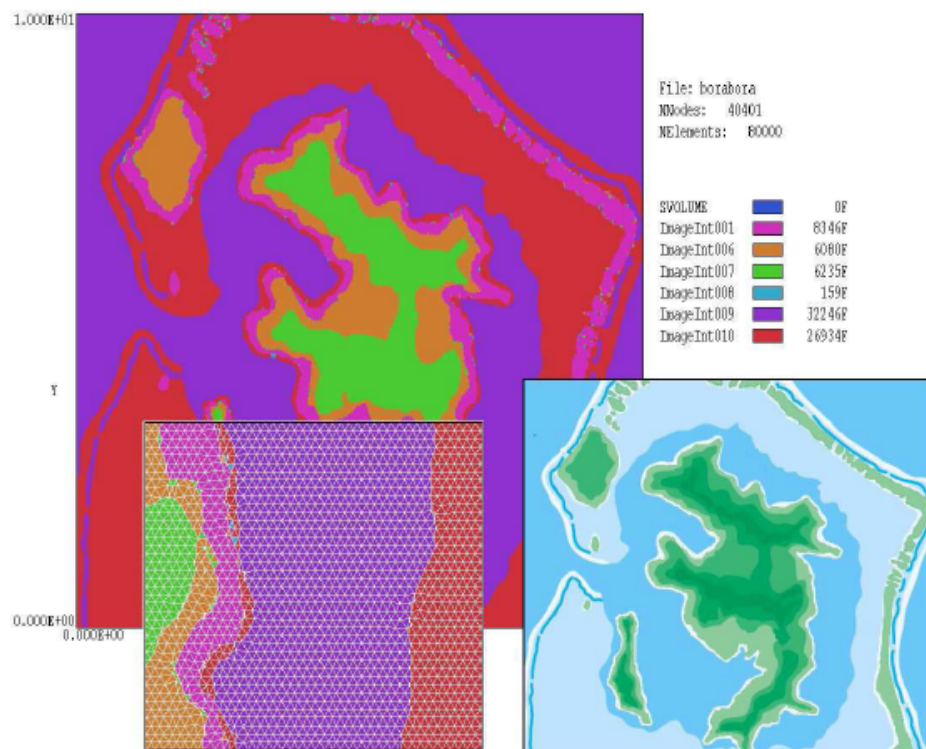


図29 ビットマップ・イメージからの形状適合3角メッシュの生成。右下：入力画像。左下：要素境界を表示したメッシュの詳細図。

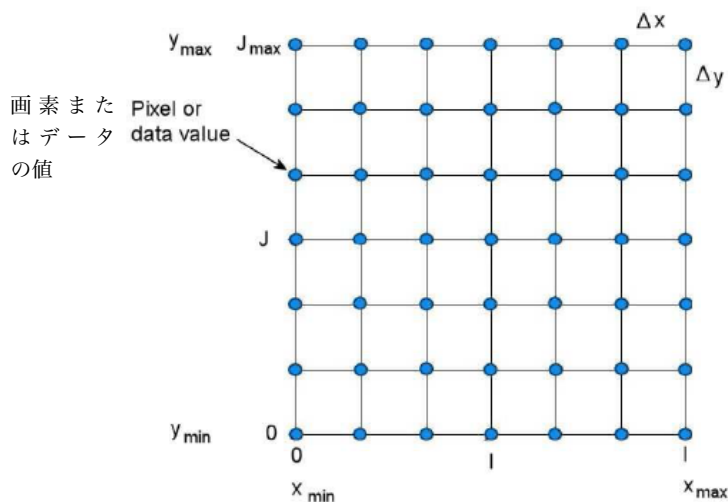


図 30 イメージの量の定義。量の値が 2 次元の正方形または長方形メッシュの格子点に与えられる。

8.2 スクリプトのImage(イメージ)セクション

復習すると、標準の **Mesh** スクリプトは、GLOBAL セクションとそれに続く 1 つ以上の REGION セクションから成ります。GLOBAL セクションは、基盤メッシュの特性（解析空間の形状およびその内部を占める 3 角形要素のおおよそのサイズ）を定義します。REGION セクションは、解析空間内にあるオブジェクトの輪郭を形成するベクトルの記述からなります。領域を処理するにあたり、プログラムは節点を材質境界上に位置するように移動し、境界の内部に含まれる節点と要素に現在処理中の領域(current region)の番号を付与します。

Mesh スクリプトは、次のような形式の 1 つ以上の「IMAGE (イメージ)」セクションを含むこともあります。

```
IMAGE
  (Image commands)
END
```

この構文の機能は、2次元情報を含むファイルをロードし、このファイルにおける値に基づいていくつかの領域を作成します。Imageセクションは、Globalセクションと、少なくとも1つの

Regionセクションのあとに続きます。多くの場合、スクリプトは次のような構成となります。

```
GLOBAL
(Global commands)
END

REGION FILL SolutionArea
(Region commands)
END

IMAGE
(Image commands)
END

REGION [FILL] Object01
(Region commands)
END

REGION [FILL] Object02
(Region commands)
END
...
ENDFILE
```

最初の Region セクションは、解析領域(region)の境界を定義します。この解析領域は、Global セクションで定義した長方形の全解析領域を満たすことも、満たさないこともあります。画像から生成した新たな領域は、解析空間内部に収まるように切り取られます。Image セクションに続く Region セクションは、Image セクションで定義されたイメージを部分的に上書きすることがあります。たとえば Image セクションで、肝臓断面内の電気伝導度の異なる領域を定義したとすると、それ以降の Region セクションで、正確な寸法を持つ電極や絶縁体を挿入することができます。

Region と Image セクションが **Mesh** スクリプトに現れる順序は重要です。次のような規則が適用されます。

- 処理中 (current) の Region または Image は、共有空間にある節点や要素の領域属性を上書きします。
- 最初の Region の機能は解析空間全体を定義することです。**Mesh** は指定された境界上の節点を移動して固定し、キーワード「fill」に応じて、定義された解析空間に含まれるすべての節点と要素に領域番号 RegNo = 1 を設定します。

- Image セクションは、1 つ以上の新たな領域に対して要素（および関連する節点）を再指定します。すでに有効な領域番号(RegNo > 0)を持っている要素にのみ、再指定がなされます。このようにして、解析空間領域は画像にクリッピング(切り取り)情報を与えます。
- Imageセクションに続くRegionセクションは、共有する空間の領域番号を変えることがあり、また指定領域の境界ベクトルに適合するように節点を移動することがあります。

現時点では、規則が多少わかりにくいかもしれませんが、実際には、この手順はとても簡単に行えます。次節でイメージ処理の操作例を紹介致します。

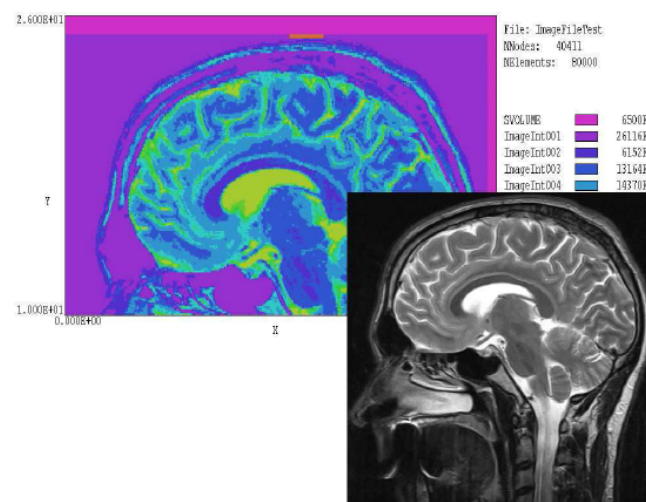


図31 「BRAINMRI」の例題。ビットマップ・イメージを形状適合3角形メッシュに変換。

8.3 視覚画像(visual image)に対するコマンドと処理手順

本節では、視覚画像からメッシュを生成する手順を、図 31 のグレースケールの MRI ビットマップ・イメージを形状適合メッシュに変換する例を使って説明します。この例題では、**Mesh** のいくつかの高度な機能が紹介されます。また視覚画像の取り込みによって、どのようなことが可能になるかということも説明します。まず説明を始める前に、限界と制約について理解することが大切です。

- 電場または熱を解析するためには、画像が **TriComp** シリーズのコードで処理される、2次元座標の対称性と矛盾がないようにしないといけません。図 31 の画像は、明らかに平面(xy 直交座標系)または円筒対称性(rz 円筒座標系)がありません。この例は、メッシュ技法のデモとしてだけ考えるべきです。
- 視覚画像を処理するときに、**Mesh** は「Lightness(明度)」または「Hue(色相)」の値に基づいて解析領域を分割することができます。これらの値は、次の段階の解析プログラムで領域に指定される物理的性質と、なんらかの相関関係を持たなければなりません。場合によっては画像が直接的に有用なことがあります。たとえば、特有の色を持つ、複雑なポールピース(磁極片)の断面を撮影し、この写真を磁場解析において、対応する領域の作成に使うことができることでしょう。図 31 の画像は、直接的に有用でないかもしれません。Lightness の値は核種の濃度に依存していて、熱伝導率および導電率のような量とはほとんど関係ないかもしれません。
- 「PaintShop Pro」、「PhotoShop」、「GIMP」のようなプログラムにより前処理すると、画像の有用性を非常に高めることができます。たとえば、「投げ縄 (lasso)」ツールを用いて、図 31 の重要な脳の構造部分を選択し、それらを **Mesh** により容易に識別可能な、特徴的な色に変換することができるでしょう。
- 「Image」セクションは **Mesh** の描画エディタ(drawing editor)には転送されません。描画エディタを用いて、通常の領域を作成し、画像をスクリプトに直接加えてください。

このような留意事項を念頭におき、図 31 の例の説明に進みます。次の入力ファイルが **Mesh** の例題ライブラリ(「Examples\MeshExamples」フォルダ)に入っています。

- **BRAINMRI.BMP**

Windows/OS2 ビットマップ・フォーマットによるビットマップ・イメージ (図 31 の右下の図)

- **BRAINMRI.MIN**

Mesh の入力スクリプト

このスクリプトを表 5 に示します。

表5 ファイル BRAINMRI.MIN の内容

```

GLOBAL
  XMesh
    0.00 25.0 0.10
  End
  YMesh
    10.00 26.0 0.10
  End
END
REGION FILL SVolume
  L 0.0 10.0 25.0 10.0
  L 25.0 10.0 25.0 26.0
  L 25.0 26.0 00.0 26.0
  L 0.0 26.0 00.0 10.0
END
IMAGE
  ImageFile Brain_MRI.bmp 0.00 0.00 24.5 25.0
  Intervals Lightness
    0.0 40.0
    40.0 50.0
    50.0 60.0
    60.0 70.0
    70.0 80.0
    80.0 90.0
    90.0 100.0
  End
  Correct 5
END
REGION FILL Electrode01
  L 24.0 13.0 24.2 13.0
  L 24.2 13.0 24.2 15.0
  L 24.2 15.0 24.0 15.0
  L 24.0 15.0 24.0 13.0
END
REGION FILL Electrode02
  L 13.0 24.8 15.0 24.8
  L 15.0 24.8 15.0 25.0
  L 15.0 25.0 13.0 25.0
  L 13.0 25.0 13.0 24.8
END
ENDFILE

```

「GLOBAL」セクションは、 x 方向に 25.0cm、 y 方向に 26.0cm の大きさの基礎メッシュを定義します。最初の Region セクションは全計算領域の輪郭を描き、直線状の領域境界に沿って要素節点を並べ、含まれる全要素と節点に $RegNo = 1$ を設定します。それに続く「Image」セクションには、次の 4 つのコマンドがあります。

IMAGEFILE BRAINMRI.BMP 0.0 0.0 24.5 25.0

このコマンドは画像ファイルをロードします。オプションの実数パラメータは、(0, 0)および(24.5, 25.0)を対角とする解析空間の長方形部分の中に画像をマッピングすることを指定しています。パラメータは物理的大きさを指定し、画像を移動または拡大・縮小することに用いることができます。省略した場合、プログラムは基礎メッシュの長方形に画像を適合させます。

INTERVALS LIGHTNESS

**0.0 20.0
20.0 30.0
30.0 40.0
40.0 50.0
50.0 60.0
60.0 70.0
70.0 80.0
80.0 90.0
90.0 100.0
End**

この構文は、各要素が Lightness の値に応じて領域に割り当てられることを指定します。**Mesh** は RGB の画素値を HLS (色相-明度-彩度) に変換します。色相値は 0.0° ~ 360.0° 、明度値は 0.0~100.0 です。9 区間のデータに基づき、9 つの新たな領域 (*RegNo* = 2~10) が作成されます。例えば 0.0~40.0 の範囲にある明度値を持つ要素は、*RegNo* = 2 に割り当てられます。

CORRECT NCorrect

CORRECT = 5

要素を割り当てたあと、**Mesh** にスムージングを5サイクル行うように指示するコマンドです。スムージング処理は、画像領域間の境界上にあるジグザグのエッジを削減します (8.4節参照)。

残りの 2 つの Region セクションで、電極部を頭部に取り付けます。「Electrode02」と指定された領域は、図 31 のメッシュの最上端近くにあるオレンジ色の四角形パーツとして見えます。

リストファイル「BRAINMRI.MLS」は画像変換処理についての有用な情報を含んでいます。たとえば、「ImageFile」コマンドがスクリプトに現れると、**Mesh** は画像情報の表を作成します。

```

Image file size    NX:  480 NY:  489
Image number of colors:    256
Image function limits  Min:    0.000E+00  Max:    1.000E+02
Image analyzed by LIGHTNESS
  LightMin    LightMax    Pixels
=====
    0.0000    2.0000      193
    2.0000    4.0000     11698
    4.0000    6.0000     36758
    6.0000    8.0000     16646
    ....
   94.0000   96.0000      431
   96.0000   98.0000      525
   98.0000  100.0000     2764

```

このデータは、領域分割のときの、適切な区間の決定の際に役立ちます。「Hue」または「Lightness」の値に基づいて要素を区間に割り当てたのち、**Mesh** は次の表を作成します。

Distribution of elements in intervals				
Interval	FMin	FMax	FAverage	NElem
=====				
1	0.000E+00	2.000E+01	7.404E+00	26275
2	2.000E+01	3.000E+01	2.600E+01	6153
3	3.000E+01	4.000E+01	3.622E+01	13164
4	4.000E+01	5.000E+01	4.524E+01	14370
5	5.000E+01	6.000E+01	5.473E+01	5835
6	6.000E+01	7.000E+01	6.525E+01	2734
7	7.000E+01	8.000E+01	7.538E+01	1742
8	8.000E+01	9.000E+01	8.507E+01	1113
9	9.000E+01	1.000E+02	9.797E+01	2114

この表は区間の境界値と、その区間にあるメッシュ要素の数を示しています。4 番目の列（「FAverage」）は、その区間に割り当てられた要素にわたる、Hue または Lightness の面積加重平均値です。この量は、次節で述べるデータ画像を取り扱うときに役立ちます。最後の表は、メッシュにおける新たな領域と区間の関連を示します。要素が割り当てられていない領域がある場合、プログラムはエラーメッセージを記述します。

8.4 データ画像からのメッシュ生成

データ画像とは、長方形メッシュの格子点において定義された、一組の実数値 $F(I, J)$ を記録したファイルです（図 30）。インデックス(指数)の値が位置を表しています。:

$$X_I = x_{min} + I \Delta x, \quad Y_J = y_{min} + J \Delta y, \quad (11)$$

ここで、

$$\Delta x = \frac{x_{max} - x_{min}}{I_{max}}, \quad \Delta y = \frac{y_{max} - y_{min}}{J_{max}}. \quad (12)$$

ファイルの形式は次のようになります。

```
Line 1: IMax JMax
Line 2: XMin YMin XMax YMax
Lines 3 through 2 + (IMax+1)(JMax+1)
  F(0,0)
  F(1,0)
  ...
  F(IMax,0)
  F(0,1)
  ...
  F(I,J)
  ...
  F(IMax,JMax)
```

インデックスの最大値 I_{max} と J_{max} は整数です。他のすべての量は有効な任意のフォーマットで表した実数です。ファイルには、空白行とアスタリスク「*」で始まるコメント行があります。インデックスの値の順序は、 J はループの外側、 I は内側であることに留意してください。

Mesh の例題ディレクトリには、データファイル作成の仕方を例示する Perl スクリプト「testimage.pl」があります。このスクリプトを修正して、任意の関数 $F(x, y)$ に関するファイルを作ることができます。提供されたスクリプトには、次の関数の $-5.0 \leq x \leq 5.0$ および $-4.0 \leq y \leq 4.0$ の範囲における値が記録されています。

$$F(x, y) = \sin\left(\frac{\pi x}{5.0}\right) \cos\left(\frac{\pi y}{8.0}\right), \quad (13)$$

次の例に、結果として得られるファイル「cosine.dat」が用いられます。

データ画像をメッシュに取り込む方法を、スクリプト「COSINEDEMO.MIN」(表 6) で例示します。この例において、最初の領域は、全解析領域の四角形のうち角を斜めに切り落とした残りの部分です。結果として得られたメッシュを図 32 に示します。この画像は最初の領域内に適合するようにクリップ (切り取り) されていることに留意してください。

```

GLOBAL
  XMesh
    -5.0 5.0 0.15
  End
  YMesh
    -4.0 4.0 0.15
  End
  Smooth 5
END
REGION FILL SVolume
  L -3.5 -4.0 3.5 -4.0
  L 3.5 -4.0 5.0 -2.5
  L 5.0 -2.5 5.0 2.5
  L 5.0 2.5 3.5 4.0
  L 3.5 4.0 -3.5 4.0
  L -3.5 4.0 -5.0 2.5
  L -5.0 2.5 -5.0 -2.5
  L -5.0 -2.5 -3.5 -4.0
END
IMAGE
  DataFile COSINE.DAT Smooth
  Intervals
    -1.0 -0.8
    -0.8 -0.6
    ...
    0.6 0.8
    0.8 1.0
  End
  Correct 9
END
REGION FILL Inclusion
  L -2.5 -0.5 2.5 -0.5
  L 2.5 -0.5 2.5 0.5
  L 2.5 0.5 -2.5 0.5
  L -2.5 0.5 -2.5 -0.5
END
ENDFILE

```

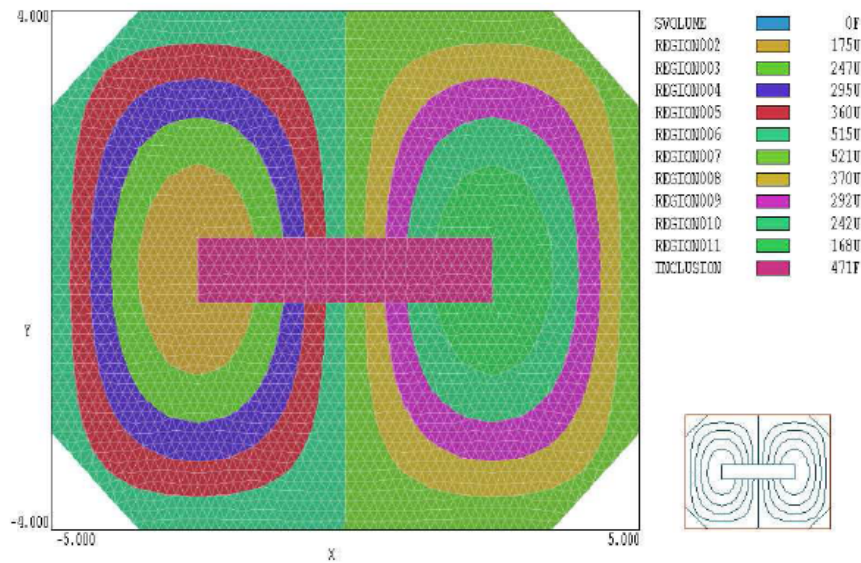


図 32 スクリプト「COSINEDEMO.MIN」およびデータ画像ファイル「COSINE.DAT」を用い、「DataFile」コマンドに「Fit」オプションを設定し、 $Correct = 9$ として生成されたメッシュ。

イメージセクションには 3 つのコマンドがあります。「Intervals」と「Correct」コマンドは 8.3 節で述べたように機能します。新たな次のコマンドが現れます。

DATAFILE COSINE.DAT FIT

ファイルから画像をロードし、データ値と、Intervals コマンドにより定義された区間にしたがって、要素を割り当てます。

このコマンドにおけるキーワード「Fit」は、領域境界上の節点を、曲線 $F(x, y) = F_{\max}$ および $F(x, y) = F_{\min}$ に密接に対応して移動するように **Mesh** に指示します。ここで、 $F_{\max}$ と $F_{\min}$ は Intervals コマンドで定義された領域の限界値です。**Mesh** は次の手順で表面への適合処理を行います。

- 画像の各領域境界を構成する要素ファセットを集めます。
- ファセットが $F_{\max}$ と $F_{\min}$ のどちらに近いかにより、ファセットを 2 つの組に分けます。
- $F_{\max}$ に近いファセットに接続された各節点の位置 X_0 における、イメージ関数の値 F_0 とそのグラディエント(勾配)の値 ∇F_0 を見出します。
- 次式により、変位ベクトルを計算します。

$$\delta \mathbf{x}_0 = \frac{\nabla F_0}{|\nabla F_0|^2} (F_{max} - F_0).$$

- 節点の位置を、 $X_1 = X_0 + \alpha \delta x_0$ ($\alpha < 1.0$) にしたがって補正します。
- F_{min} に近い節点についても同じ操作を行います。
- この手順を *NCorrect* 回繰り返して、節点を理想的表面に向けて徐々に移動します。

この処理は、関数のグラディエントが定義されるときだけ有効なことを認識するのが大切です。ノイズの多いデータや不連続なデータの場合、境界が予測している形と異なったり、要素がひずんだりします。このような場合、DataFile コマンドにおいて、キーワード「Smooth」を用いてください。このオプションは、視覚画像に用いられるスムージングルーティンを呼び出します。

「COSINEDEMO」の例題におけるスムージングと適合処理の結果を図 33 に示します。コマンド

CORRECT 0

を用いて、スムージングと適合処理を行わなかったときのメッシュの状態を図 33a に示します。重心において評価した関数の値が、領域の区間内にあることをチェックして、要素に領域を割り当てます。この処理は境界上のジグザグしたエッジを修正しません。図 33b は、エッジをやりくり仕上げるようにスムージングする処理を 9 サイクル行って修正したメッシュを示しています。境界の連続性は良くなっていますが、境界の位置は必ずしも正確ではありません。最後の図 33c は、適合処理(fitting)を 9 サイクル行った結果のメッシュの状態です。このオプションでは、イメージ関数の等高線に近くなるように正確に境界を位置させます。この処理はオプションとして推奨できますが、イメージ関数およびそのグラディエントが、位置に関して連続関数であるときにしか適用できません。

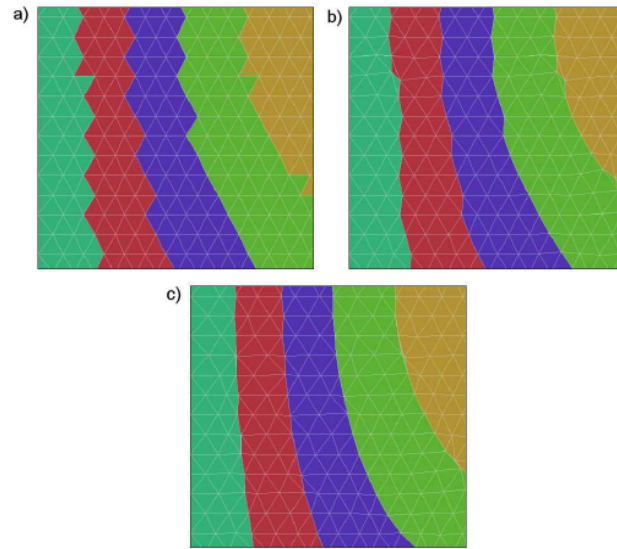


図 33 図 32 の拡大図。 a) $NCorrect = 0$ と設定して、適合処理とスムージングを抑制した場合。 b) $NCorrect = 9$ としてスムージングした結果。 c) $NCorrect = 9$ で適合処理した結果

8.5 Image コマンド

スクリプトには、一つ以上の Image セクションが Region セクションと混在できます。また、最初の Image セクションの現れる前に、少なくとも一つの Region セクションがなければなりません。最初の Region セクションでは、画像をクリップする、解析領域の境界を定義します。Region および Image セクションで定義された領域には、上書き規則が適用されます。Image セクション内に、次のコマンドが任意の順序で現れます。

IMAGEFILE FileName [XIMin YIMin XIMax YIMax]

IMAGEFILE = tulsa.bmp (0.0, 0.0, 5.0, 5.0)

視覚画像ファイルをビットマップフォーマット (BMP、PCX、PNG) でロードします。画像ファイルは TC.exe で指定した作業ディレクトリに存在する必要があります。オプションの実数パラメータ X_{imin} 、 Y_{imin} 、 X_{imax} 、 Y_{imax} が検出されると、画像は (X_{imin}, Y_{imin}) および (X_{imax}, Y_{imax}) を対角とする長方形領域にマッピングされます。この長方形は解析空間内に完全に納まることも、納まらないこともあります。パラメータが省略されると、画像は解析空間の長方形 $(X_{min}, Y_{min}, X_{max}, Y_{max})$ の境界内を満たすようにマッピングされます。

DATAFILE FileName [Fit, Smooth]

DATAFILE = VARDIELECTRIC.DAT (Fit)

データ画像ファイルを 8.4 節において述べた、テキストフォーマットでロードします。この画像ファイルは TC.exe で指定した作業ディレクトリに存在する必要があります。オプションのキーワード「Fit」は、画像領域の境界上の節点を、イメージ関数の該当する等高線上に乗るよう数学的計算により適合させて移動することを指示します。このオプションは、イメージ関数およびそのグラディエント (勾配) が連続的であるときにのみ使用できます。そうでない場合は、「Smooth」オプション (デフォルト設定) を用いてください。

INTERVALS [Abs, Rel] [Hue, Lightness]

Flow(1) FHigh(1)

...

Flow(NInt) FHigh(NInt)

END

この構造化されたコマンドは、画像要素を割り当てる区間を定義します。このコマンドは、メッシュに $NInt$ 個の領域を設定します。領域には連続的に番号が付けられ、*Region003*、...など

の形式でデフォルトの名前が付与されます。メッシュにおける領域の合計数が 250 を超えると、プログラムはエラーメッセージを出します。各データ行には、領域の上限値と下限値を（実数値で）記述します。区間は重なり合ってはなりませんが、連続的または一様である必要はありません。オプションの「Abs」（デフォルト設定値）は、イメージ関数の値を $F_{\text{low}}(i)$ および $F_{\text{high}}(i)$ と直接比較して、要素が領域内にあることを確かめるように指示しています。オプションの「Rel」は、相対比較を行うことを指示しています。この場合、 $F_{\text{low}}(i)$ および $F_{\text{high}}(i)$ は、0.0 と 1.0 の間の値でなければなりません。比較する量は、 F_{min} および F_{max} をイメージ関数の計算限界値とするとき、 $F_{\text{low}}(i) \times F_{\text{min}}$ および $F_{\text{high}}(i) \times F_{\text{max}}$ です。RGB 画像における明度の値は、0.0（黒）から 100.0（白）までの範囲で、色相は 0.0° から 360.0° まで変化します。ここで、0.0° は赤、120.0° は緑、180° はシアン、240° は青に対応します。

CORRECT NCorrect

CORRECT = 9

基盤メッシュの 3 角形要素を領域に割り当てるとき、境界のジグザグが残ります。適合処理またはスムージング（図 33）は、「NCorrect」で指定したサイクル数の間、領域境界に沿って位置を平均化します。NCorrect の値を 0 にした場合は、境界を修正しません。大きな値にすると、スムージングまたは適合処理が強くなります。「Fit」オプションは、データ画像にのみ適用し、また連続関数にのみ適用できます。スムージングは、領域が不連続な単一要素からなる、斑点状の画像に適用すべきではありません。また境界を過剰に修正すると、メッシュにひずみを生じさせることになりかねないので注意してください。（デフォルト値は $NCorrect = 0$ です。）