

三角形メッシュ生成プログラム grid3v4 について

2026.01

1. はじめに

本プログラム grid3v4.exe は、対象とする領域を三角形要素（メッシュ）で覆うプログラムである。本プログラムのメッシュ生成においては、解析対象の形状をいくつかの小領域に分割し、その小領域を多数の四辺形に分割し、そこから FEM 解析に必要な三角形メッシュデータを自動生成する手法に基づいている（このように、いくつかの小領域からメッシュを生成する手順は、通常、**ブロック分割法**(block-based mesh generation method)と呼ばれている)。

さらに、メッシュ生成後に、画面上で変位拘束条件および荷重条件をマウスで設定でき、ほぼ自動的に FEM プログラムに必要なデータを生成、ファイル出力できるようになっている。三角形要素に基づく FEM プログラム plane2ds.exe を別途提供しているので、この FEM プログラムに本プログラムによる出力ファイルを読み込ませれば、メッシュ生成と FEM 解析とがシームレスにつながる。したがって、これら 2 つのプログラムを用いれば、一般の汎用 FEM ソフトと同様な操作手順を体験することができる。

メッシュ分割のプログラムの基本は、ブロック分割法を詳述した文献(1)の手法を骨子としている。すなわち、その著書に示されている FORTRAN プログラムを参考に、VB2026 の GUI 機能を活かして作成している。

本プログラムが有するメッシュ分割機能は、

- (1) 生成可能な要素数、節点数を 30,000 節点まで。
- (2) 材料数を 10 個まで設定可能。
- (3) マウスの左クリック+ドラッグによって、図形の移動が行える。また、センターボタンの動きに応じて図形の拡大縮小が行える。これらの機能は、Google Maps と同様な操作法に準じていて、メッシュの細かい箇所での変位条件や荷重条件を容易に設定できる。
- (4) メニューに従って、節点の拘束条件 (x, y 方向変位拘束など) や荷重条件をマウスを用いて簡単に設定できる。
- (5) 本プログラムによってメッシュ生成したデータや拘束、荷重データをファイルに出力できる。さらに、この出力ファイルをただちに FEM 解析プログラム (plane2ds.exe) の入力ファイルとして利用可能である。

などである。

さらに、文献(2)の Collins の論文のアルゴリズムを参考にして、節点番号のリナンバリングを行い、生成されたメッシュのバンド幅を最小化している。これにより、剛性マトリックスのゼロ成分を削減し、有限要素解析の計算効率が向上する。

2. 対象とする例題

図 1(a)に示す円孔を有する帯板を例にとりメッシュ生成法の説明を行う。帯板の寸法は横 120mm, 縦 80mm, 板厚は $h=4\text{mm}$ であり、この帯板の中心に直径 40mm の穴が空いている。また、帯板の左右

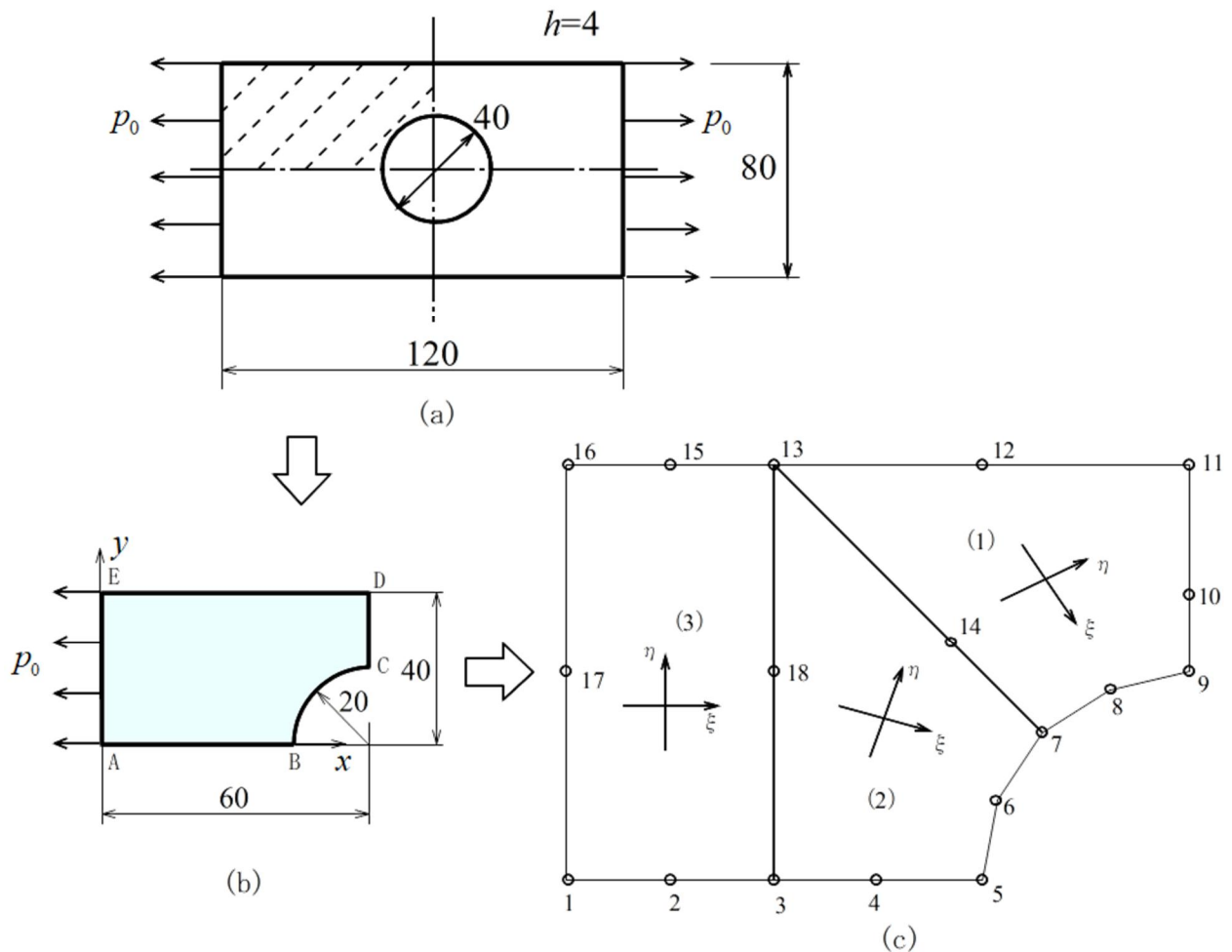


図 1 円孔を有する帯板の引張り

から 9800N の引張り力が作用するものとする。

本プログラムの入力データとなる**基本データ** (basic data) は、

- (1) あらかじめ解析対象をブロック分割し、それに基づいて必要な情報を数値化する。
- (2) 本プログラムの DataGrid (エクセルのようなデータ枠) を利用して (1) で準備したデータを入力する。
- (3) (2) の DataGrid を利用する代わりにエディタ (たとえば、Windows11 付属のノートパッドなど) で基本データを作成し、csv 形式でセーブしておいてもよい。ただし、データミスの少ない (2) の DataGrid の利用を推奨する。

により作成する。

なお、図 1(a)の問題は、形状や荷重の対称性を考慮して、図 1(b)のように左上半分の 1 / 4 領域を考えればよい。

3. 準備すべきデータ

3. 1 領域の分割

まず，図 1(b)を，図 1(c)のように 3 個の四辺形に分割し，各々の四辺形の 4 つの頂点のほかに，各辺の中心付近（必ずしも中心である必要はないが）にも節点を設ける．これらの節点については，1 から番号付けをしてそれらの (x,y) 座標をあらかじめ求めておく．番号の付け方は任意でよいが，規則性を与えると処理しやすい．ここでは，図 1(b)の点 A を原点にすると，18 個の各節点の番号と座標は

1, 0.0, 0.0	10, 60.0, 27.5
2, 10.0, 0.0	11, 60.0, 40.0
3, 20.0, 0.0	12, 40.0, 40.0
4, 30.0, 0.0	13, 20.0, 40.0
5, 40.0, 0.0	14, 37.0, 23.0
6, 41.52, 7.65	15, 10.0, 40.0
7, 45.85, 14.14	16, 0.0, 40.0
8, 52.35, 18.48	17, 0.0, 20.0
9, 60.0, 20.0	18, 20.0, 20.0

となる．

図 1 (c) に示した節点は，1 から始まる任意の番号を有していて，節点の総数は 18 個である．このとき，(1)，(2)，(3) と表示された四辺形から構成される各小領域は，隅部に 4 個，各辺の中間に 4 個の合計 8 個の節点を有している．なお，各小領域に示された座標 (ξ, η) は，各領域での正規化された局所座標である．

3. 2 各領域を碁盤の目のように分割

はじめに，領域 (1) を取り上げて説明する．

図 2 に示すように，領域 (1) は 13, 14, 7, 8, 9, 10, 11, 12 と左回りの節点番号により囲まれている．また，はじめの 3 個の節点 13, 14, 7 を含む辺を辺番号 1 と数え，左回りに辺 2, 辺 3, 辺 4 と数える．

領域分割の際には，これらの辺がどの領域と接続しているのかも前もって知っておく必要がある．たとえば，接続する領域がない場合は 0 として，領域 (1) の 4 つの辺の接続状況は
領域 (1) 辺 1→2, 辺 2→0,

辺 3→0, 辺 4→0

と表す．

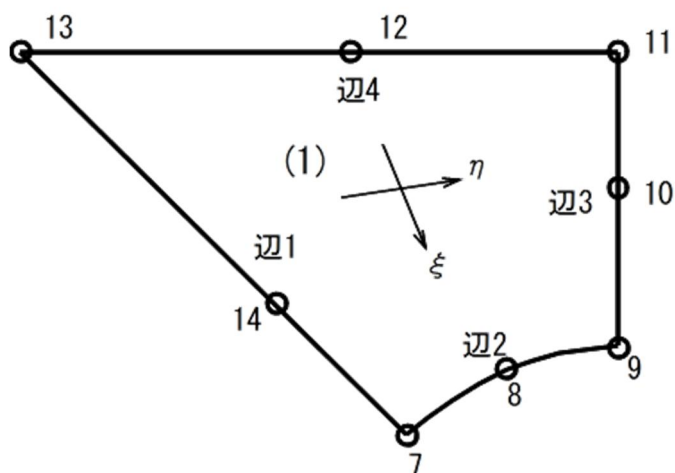


図 2 領域 (1) の辺番号，節点番号

ほかの領域についても

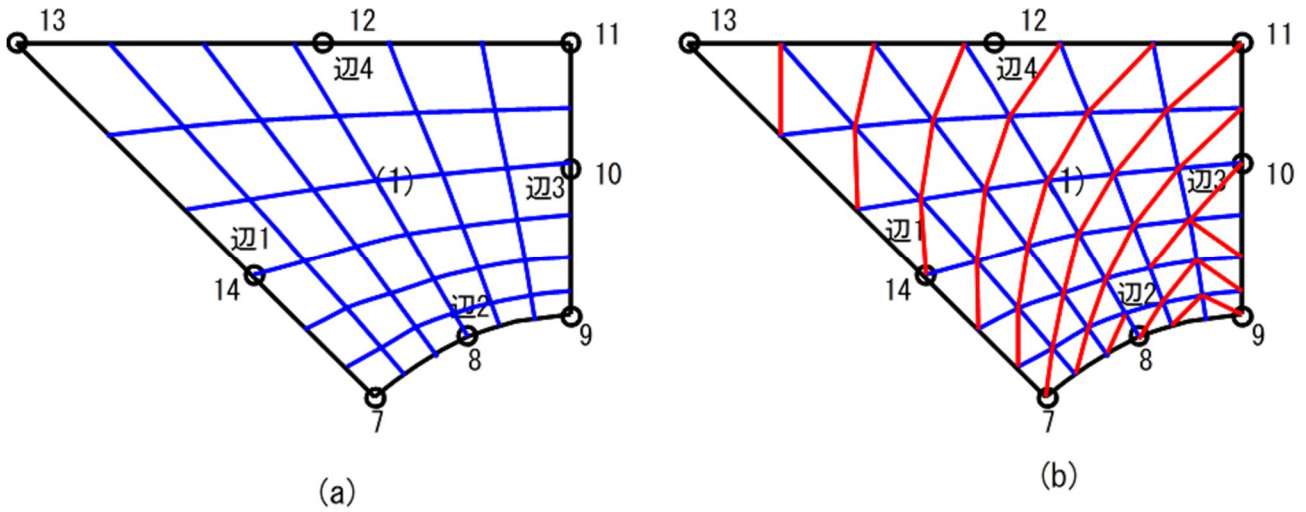


図 3 領域 (1) の格子線による細分割と三角形要素分割

領域 (2) 辺 1→0, 辺 2→0, 辺 3→1, 辺 4→3

領域 (3) 辺 1→0, 辺 2→2, 辺 3→0, 辺 4→0

となることがわかる。メッシュ分割プログラム内部では、この連結データは、次の領域に移って要素分割した際に、節点番号が 2 重にふられることのないように利用されている。

次いで、領域 (1) を格子線によって基盤の目のように細分割することを考える。図 3(a)は、領域 (1) を行方向に 7 本の分割線、列方向に 7 本の分割線で分割した例を示す。このとき、辺 1 と辺 3 に沿った方向を行(row)方向、辺 2 と辺 4 に沿った方向を列(column)方向とよぶことにする。

行分割線の数と列分割線の数によって要素分割の粗密を調整することができる。なお、図 3(a)の基盤

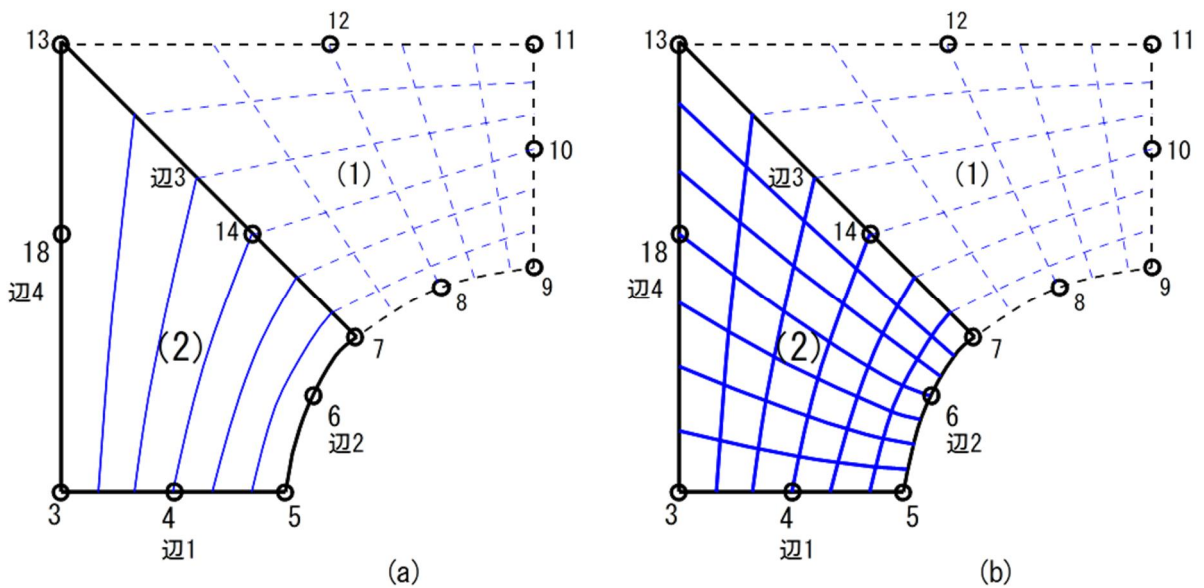


図 4 領域 (2) の格子線による細分割

の目の分割線は、座標変数 x の 2 次曲線によって表されている。この細分割によって、領域 (1) は 49 個の節点と 36 個の 4 角形要素に分割される。

次に、図 3(a) の各 4 角形要素に対角線を書き入れ、それぞれの 4 角形を 2 個の三角形に分割することを考える。ただし、この対角線を導入する際には、細長い三角形とならないように、2 個の対角線のうちで、その長さが短い方を採用する。これにより、図 3(a) の基盤の目状の四角形は、図 3(b) の三角形に分割される。なお、基盤の目状の分割に応じて各節点や各要素には番号付けがなされる。

次に、3, 4, 5, 6, 7, 14, 13, 18 と囲まれている、領域 (2) の細分割の様子を図 4(a),(b) に示す。

図 4(a) は、節点 3, 4, 5 を辺 1 と定め、領域 (2) の辺 2 と辺 4 とで挟まれた列方向を 7 分割した様子を示す。この場合の分割線数 7 は、領域 (1) の行分割線数 7 と同数である必要がある（そうでないと基盤の目の数に食い違いが生じて要素分割が正しく行われぬ）。一方、図 4(b) は、節点 3, 4, 5 の辺 1 と節点 7, 14, 13 の辺 3 とで挟まれた行方向を 8 分割した様子を示す。この行方向の分割数には特に制約はないが、次に説明する領域 (3) の分割数の設定に影響を与える。以上により、領域 (2) は 56 個の節点（ただし 7 個の節点は領域 (1) と重複している）と 42 個の 4 角形要素に分割される。

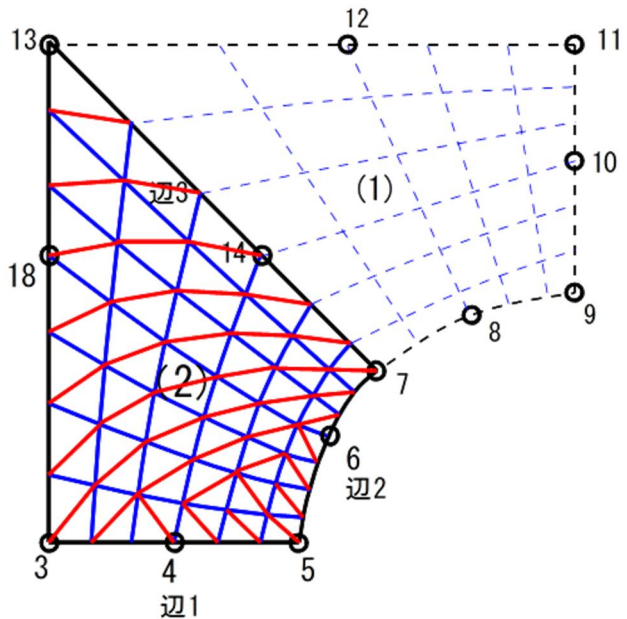


図 5 領域 (2) の三角形要素細分割

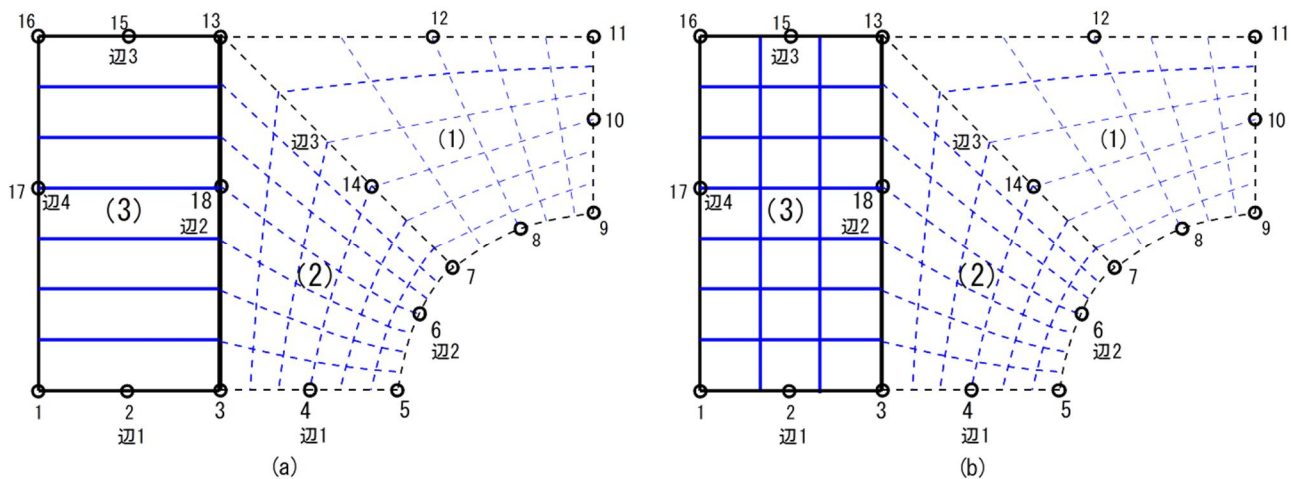


図 6 領域 (3) の格子線による細分割

図 5 は、領域 (2) をさらに、三角形要素に細分割した結果を示し、領域 (2) が 84 個の三角形要素に分割されていることがわかる。

図 6(a) は、1, 2, 3, 18, 13, 15, 17 の節点に囲まれた領域 (3) の行方向の分割である。節点 1, 2, 3 を含む辺を辺 1 と定め、領域 (3) の行方向を 8 本の分割線で分割した様子である。この場合の分割

線数 8 は、先の領域 (2) での説明と同様、領域 (2) の行分割線数 8 と同数である必要がある。また、図 6(b)は、節点 3, 18, 13 の辺 2 と節点 16, 17, 1 の辺 4 との間の列方向を 4 分割した様子を示す。この列方向の分割数に対する制約はないが、なるべく正方形に近い 4 角形を生成するように決めるとよい。領域 (3) の三角形分割は先に述べた手順に従えばよい。

以上の手順により、3 個の小領域を基盤の目に分割した後に三角形要素に分割し、その結果、領域全体を小さな三角形要素で覆うことになる。これが領域分割法の基本的な考え方である。したがって、正しい分割を得るには、接続する領域の行分割線数あるいは列分割線数が合致するように注意することが重要である。

領域ごとに任意の材料を与えることができ、ここでは、領域 (1), (2), (3) については軟鋼を仮定して、縦弾性係数 $E=206\text{GPa}(=206\times 10^3\text{N/mm}^2)$ 、ポアソン比 $\nu=0.3$ とし、これを、材料番号 1 と定義する。また、本問題を平面応力状態（薄い平板の面内変形）と考え、板厚は $h=4\text{mm}$ である。

3.3 各領域のデータ

以上により、ブロック分割法の考え方を示したが、それに必要な領域データを以下に示す。まず、各領域が隣の領域と接しているかを示す、連結データとして

領域 (1)・・・2, 0, 0, 0 (辺番号 4 が領域 (2) と接している。それ以外の辺は接していない)

領域 (2)・・・0, 0, 1, 3 (辺番号 1, 2 はどの領域とも接してなく、辺番号 3 が領域 (1) と、
辺番号 4 が領域 (3) と接している)

領域 (3)・・・0, 2, 0, 0 (辺番号 2 が領域 (2) と接している。それ以外の辺は接していない)
が必要である。

次に、領域番号、各領域の材料番号 (1 から始まる)、各領域の行分割線数、列分割線数および領域を反時計回りに囲む 8 個の節点番号 (ただし、辺番号 1 から節点番号を数え上げる) から成る領域データが必要である。これを、本例題の各領域について改めて書き下ろすと

領域 (1)・・・1, 1, 7, 7, 13, 14, 7, 8, 9, 10, 11, 12

領域 (2)・・・2, 1, 8, 7, 3, 4, 5, 6, 7, 14, 13, 18

領域 (3)・・・3, 1, 8, 4, 1, 2, 3, 18, 13, 15, 16, 17

となる。

さらに、材料番号に応じて、縦弾性係数、ポアソン比、板厚 (平面ひずみの場合には 0.0 とする) を与える必要がある。本問題では、材料数が 1 個なので、材料番号を 1 として

1, 206e3, 0.3, 4.0

となる。

なお、単位については、各単位が互いに整合するように与える必要がある。一般的には、縦弾性係数 E に対しては、 GPa (10^9N/m^2) の単位が用いられるが、本例のように、長さの単位に $[\text{mm}]$ などを用いている場合には、それに応じて E を与える必要がある。たとえば、通常の軟鋼では、 $E=206\text{GPa}=206\times 10^9\text{N/m}^2$ であるから、これを単位変換して $E=206\text{GPa}=206\times 10^9\text{N}/((1000\text{mm})^2)=206\times 10^3\text{N/mm}^2$ として入力する必要がある。また、このデータに応じて FEM 解析によって計算された応力は、 N/mm^2 の単位を有することにも注意が必要である。

3.4 要素分割のための基本データ

以上のアルゴリズムに沿って領域を要素分割する場合に、ユーザーが用意すべきデータを以下に示す。

【用意すべきデータ】

(1) 基本データ（1行）

領域の総数，境界点の総数，材料数（最大 50 個まで）

(2) 境界点データ（境界点の数だけ用意する）

境界点の番号， x 座標， y 座標

(3) 連結データ（領域の数だけ用意する）

領域番号，連結データ（4 個）

(4) 領域データ（領域の数だけ用意する）

領域番号，材料番号（1 から），領域の節点群の行分割線数，領域の節点群の列分割線数，四角形を定義する全体節点番号（8 個）（辺 1 より始めて反時計回り）

(5) 材料データ（材料の数だけ用意する）

材料番号（1 から），縦弾性係数，ポアソン比，板厚（平面ひずみの場合には 0.0 とする）

本問題では，(1) から (5) までについては

(1)

3, 18, 1

(2)

1, 0.0, 0.0	10, 60.0, 27.5
2, 10.0, 0.0	11, 60.0, 40.0
3, 20.0, 0.0	12, 40.0, 40.0
4, 30.0, 0.0	13, 20.0, 40.0
5, 40.0, 0.0	14, 37.0, 23.0
6, 41.52, 7.65	15, 10.0, 40.0
7, 45.85, 14.14	16, 0.0, 40.0
8, 52.35, 18.48	17, 0.0, 20.0
9, 60.0, 20.0	18, 20.0, 20.0

(3)

1, 2, 0, 0, 0
2, 0, 0, 1, 3
3, 0, 2, 0, 0

(4)

1, 1, 7, 7, 13, 14, 7, 8, 9, 10, 11, 12
2, 1, 8, 7, 3, 4, 5, 6, 7, 14, 13, 18
3, 1, 8, 4, 1, 2, 3, 18, 13, 15, 16, 17

(5)

1, 206e3, 0.3, 4.0

となる。

4. DataGrid による入力とメッシュ生成

以上の説明により、データを入力する準備ができた。そこで、以下に、プログラムを起動し、基本データを DataGrid に入力する手順とメッシュ生成手順を示す。

4.1 起動画面と各メニュー

図 7 は本プログラム grid3v4.exe をダブルクリックして起動した直後の画面であり、メニューバー、タブ状の DataGrid および描画領域から構成される。



図 7 本プログラムの起動画面

エクセルのセルに似た DataGrid には、メッシュ生成のための各種データを入力し、描画領域には、対象とする図形やメッシュされた様子が描画される。さらに、プルダウンメニューには

- * メッシュ生成 => メッシュ分割, メッシュ分割 (節点, 要素番号表示)
- * 変位拘束 => x 方向変位拘束, y 方向変位拘束, xy 方向変位同時拘束, x 方向変位拘束
- * 荷重 => x 方向荷重, y 方向荷重, xy 方向荷重, 荷重全解除
- * 表示 => 節点番号・要素番号, 境界条件
- * FEM ファイル => 保存

などがあり、これらの使い方は、必要に応じて後に紹介する。

4.2 簡単な例題（長方形領域）

前章では、円孔を有する帯板のデータ例を示したが、ここでは、はじめに、より簡単な例として長方形領域のメッシュ分割から始める。

図 8 のように、厚さを 1mm とし、横 100mm、縦 50mm の長方形領域を、縦 11 本、横 6 本の格子線から成るメッシュに分割することを考える。

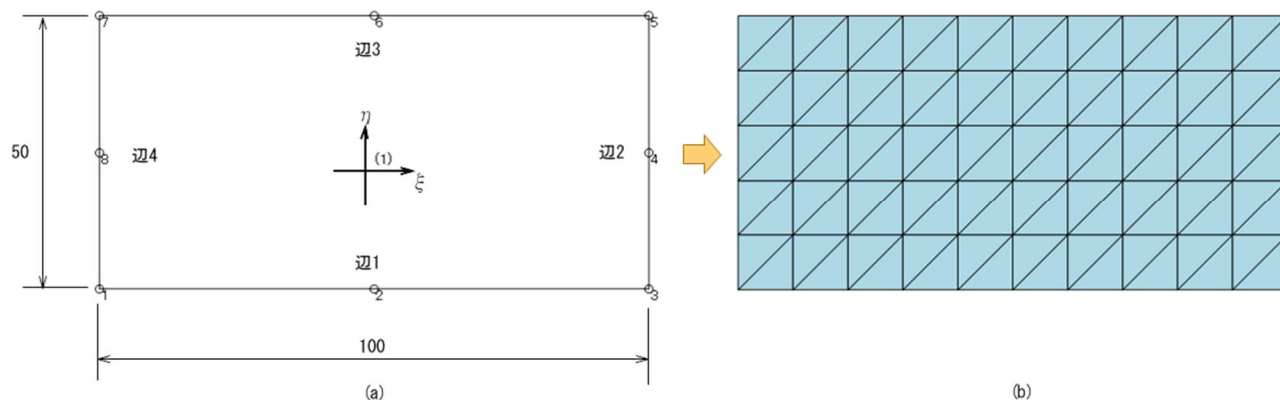


図 8 長方形領域のメッシュ分割

この場合の材料数を 1 個とすると、領域数は 1、節点数は 8、材料数は 1 だから、基本データは
1, 8, 1

である。

長方形領域の左下隅を原点にとり、8 個の節点を図 8(a) のように定めると、それぞれの節点の節点番号と (x, y) 座標は

1, 0.0, 0.0	5, 100.0, 50.0
2, 50.0, 0.0	6, 50.0, 50.0
3, 100.0, 0.0	7, 0.0, 50.0
4, 100.0, 25.0	8, 0.0, 25.0

となる。

領域番号 1 に隣接する領域は存在しないから、辺 1 から辺 4 までの領域連結データは

1, 0, 0, 0, 0

となる。

領域データについては、領域番号が 1、材料番号が 1、列分割数が 11、行分割数 6 なので、これに、辺 1 から反時計回りに数えて 8 個の節点番号を加えると

1, 1, 11, 6, 1, 2, 3, 4, 5, 6, 7, 8

となる。

材料データは、軟鋼の薄板（厚さ 1mm）を仮定して、 $E=206 \times 10^3 \text{N/mm}^2$, $\nu=0.3$, $h=1\text{mm}$ として

1, 206e3, 0.3, 1.0

とする。

以上のデータを，それぞれの DataGrid タブに入力すると，以下の図 9 のようになる。

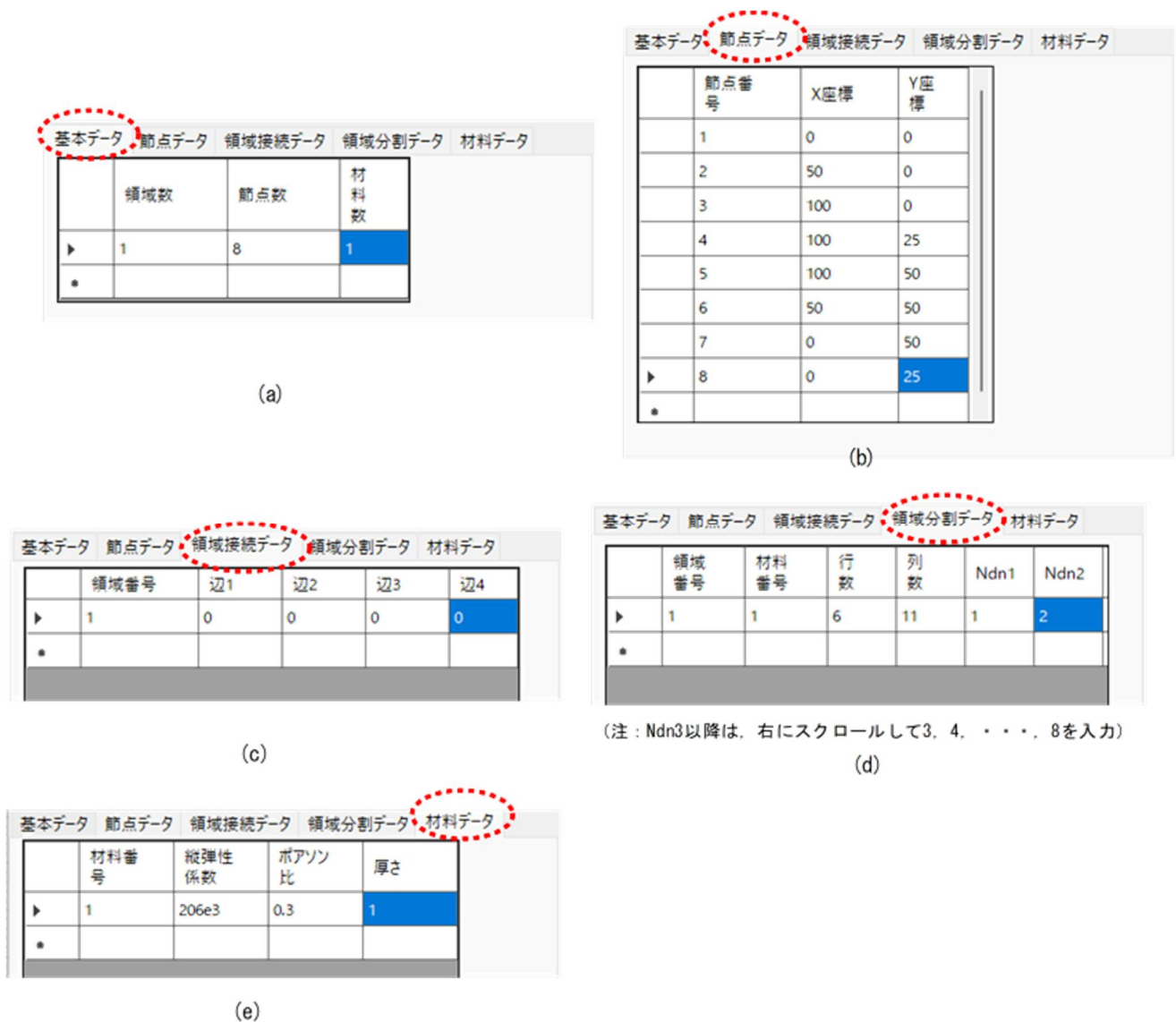


図 9 長方形領域の分割のためのデータ

入力データをファイルに保存するために，以上で作成したデータをプルダウンメニュー「ファイル」→「出力」により，csv形式でファイルに保存できる．ここでは，beam6-11.csv という名をつけて保存しておく．そのファイルの実際の内容は

Two Dimensional Elastic Problem	6, 50, 50
1, 8, 1	7, 0, 50
1, 0, 0	8, 0, 25
2, 50, 0	1, 0, 0, 0, 0
3, 100, 0	1, 1, 6, 11, 1, 2, 3, 4, 5, 6, 7, 8
4, 100, 25	1, 206e3, 0.3, 1
5, 100, 50	

である。ここで、1 行目の「Two Dimensional Elastic Problem」はデータファイルのコメントとしてつけ加えたものである。また、このファイルは、メニューバーの「ファイル」→「開く」で読み込むことができ、その内容を DataGrid に取り込むことができる。

以上でデータ入力を終えたあとに、メニューバーの「メッシュ」→メッシュ分割を選択すると図 10 のような領域の図形の確認画面が現れる。ここで、「はい (Y)」を選択すると図 11 のようなメッシュされた図が表示される。

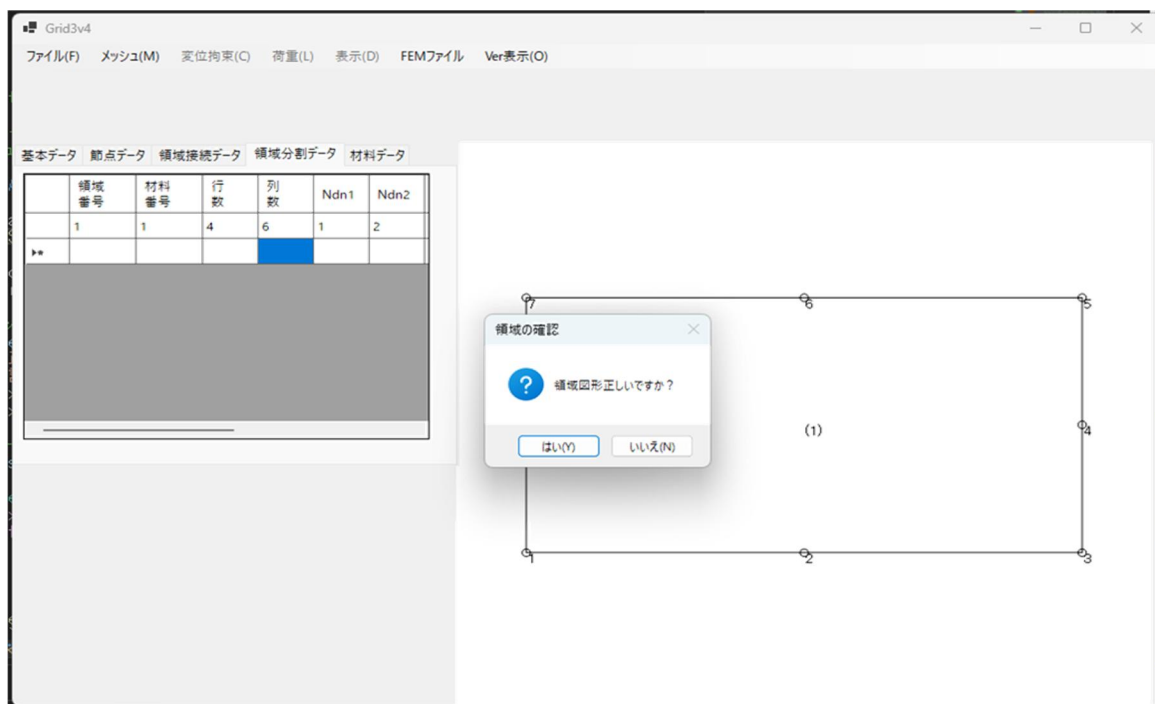


図 10 領域図形の確認

さらに、「領域分割データ」の行分割数、列分割数を書き換えれば、メッシュの細かさを容易に調整できる。図 12 は、行分割数を 21、列分割数を 41 と書き換え、「メッシュ」→「メッシュ分割」を実行した結果である。

このように、自在にメッシュの細かさを調整できるのが本プログラムの特長の一つである。

拘束条件、荷重条件および FEM ファイルの出力については、次の例題で詳しく述べる。

4.3 データファイルのテキストエディターによる作成

前節では、DataGrid に入力したデータについては、ファイルとしてセーブできることを示した。そのかわりに、DataGrid を経由せずに、テキストファイルを作成するエディタ、あるいは Microsoft Word などを用いてファイルを作成してもよい。データ作成の場合には、入力文字はすべて半角とし、データの区切りにコンマを用いる。なお、1 行目には「Two Dimensional Elastic Problem」という問題のコメントを挿入する。また、データを見やすくするために、適当に半角を入れてもよい。

Microsoft Word を用いてデータを作成する場合には、データを作成後（すべて半角文字で作成すること）に、1) 「ファイル」→「名前を付けて保存」、2) 保存先を選ぶ、3) 「ファイル名」欄に、拡張子を含

めた名前を入力（例：enkou.csv），4）「ファイルの種類」欄は「すべてのファイル（.）」を選択，5）「保存」をクリック，とすればよい。

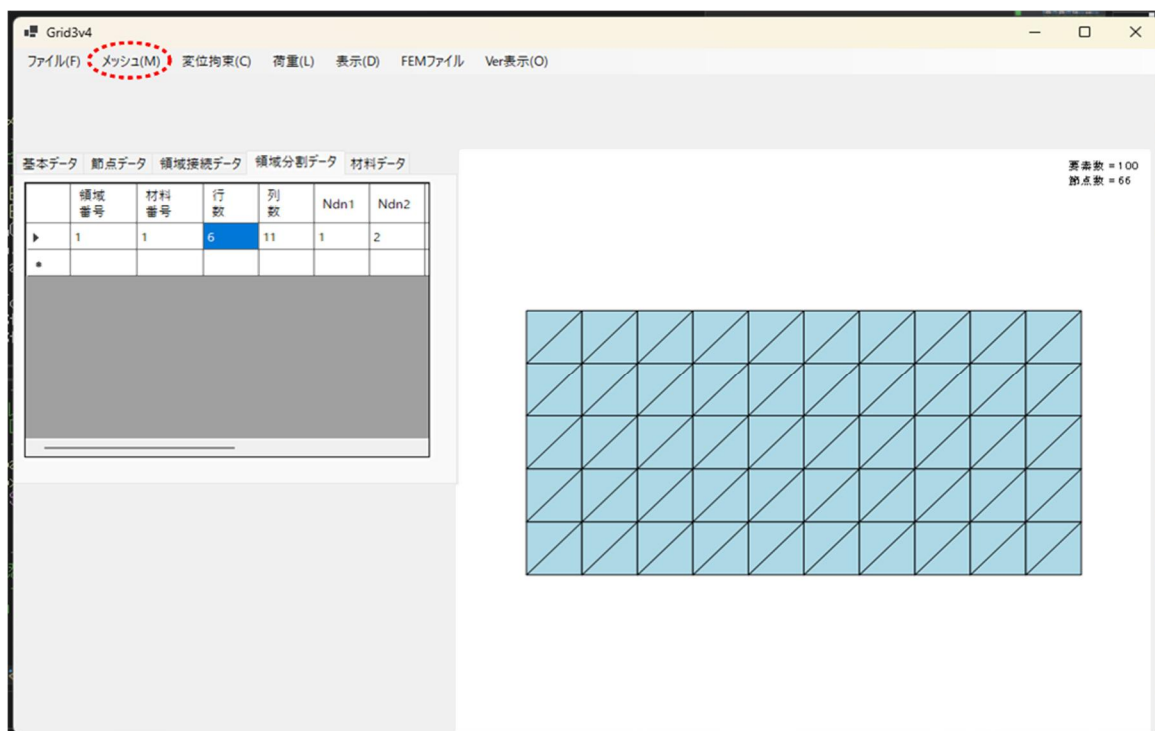


図 11 長方形領域のメッシュ分割（6×11）

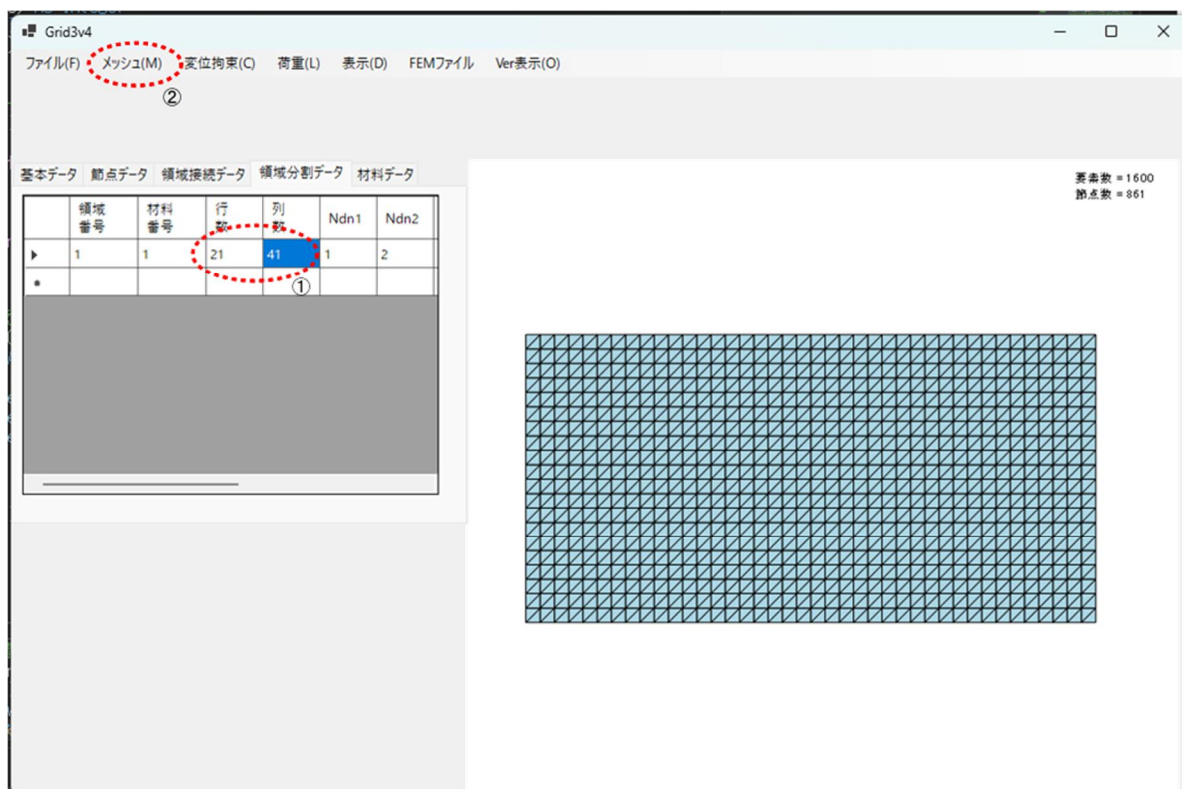


図 12 長方形領域のメッシュ細分割（21×41）

4.4 円孔を有する帯板

次に、図1の円孔を有する帯板のメッシュ生成を考える。図1(c)の領域に対して、3.4節に示したデータを DataGrid に入力した結果を図13に示す。あるいは、メニューバーの「ファイル」→「読み込み」から、添付したサンプルデータ enkou.csv を読ませてもよく、図13と同様な DataGrid の結果を得る。その後、メニューバーの「メッシュ」→「メッシュ分割」を選ぶと、図14のような三角形要素分割結果を得る。

行および列分割数については、領域1は 7×7 、領域2は 8×7 、そして領域3は 8×4 と与えており、総要素数198、総節点数122となっている。

さらに、この領域をより細かく分割しよう。「領域分割データ」の DataGrid タブを選び、行および列

基本データ	節点データ	領域接続データ	領域分割データ	材料データ
	領域数	節点数	材料数	
▶	3	18	1	
•				

(a) 基本データ

基本データ	節点データ	領域接続データ	領域分割データ	材料データ
	節点番号	X座標	Y座標	
▶	1	0	0	
	2	10	0	
	3	20	0	
	4	30	0	
	5	40	0	
	6	41.52	7.65	
	7	45.85	14.14	
	8	52.35	18.48	
	9	60	20	

(b) 節点データ（節点10以降は省略）

基本データ						節点データ	領域接続データ	領域分割データ	材料データ
	領域番号	辺1	辺2	辺3	辺4				
▶	1	2	0	0	0				
	2	0	0	1	3				
	3	0	2	0	0				
*									

(c) 領域接続データ

基本データ	節点データ	領域接続データ	領域分割データ	材料データ
-------	-------	---------	---------	-------

	領域番号	材料番号	行数	列数	Ndn1	Ndn2
▶	1	1	7	7	13	14
	2	1	8	7	3	4
	3	1	8	4	1	2
•						

(d) 領域分割データ（第3節点以降は省略）

基本データ	節点データ	領域接続データ	領域分割データ	材料データ
	材料番号	縦弾性係数	ポアソン比	厚さ
▶	1	206e3	0.3	4
•				

(e) 材料データ

図13 円孔を有する帯板のデータ

分割数のセルを、領域 1 は 21×20 、領域 2 は 19×20 、そして領域 3 は 19×11 と変更し、メニューバ

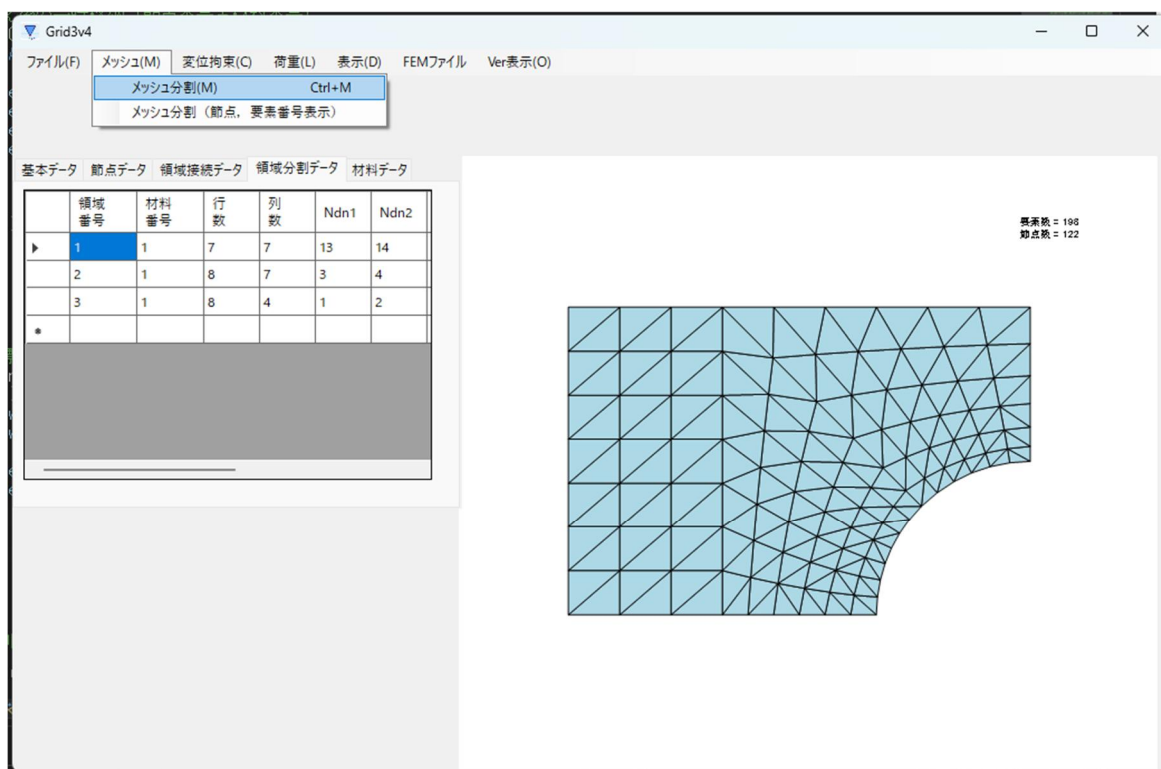


図 14 領域の三角形要素分割結果（122 節点，198 要素）

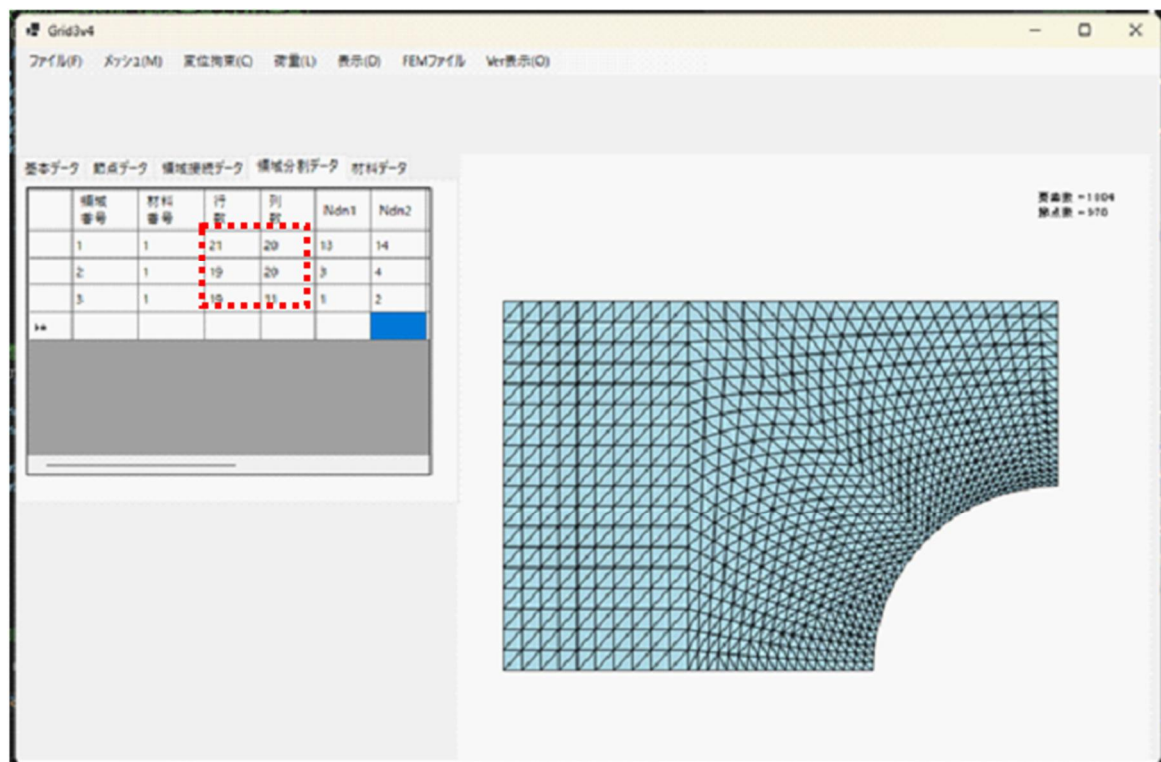


図 15 領域の三角形要素細分割結果（1804 要素，970 節点）

一の「メッシュ」→「メッシュ分割」を選ぶと、図 15 のような要素分割図を得る。

これにより、総節点数を 122 から 970 に一気に増やし、より細かいメッシュを生成することができる。
この際に、図 16 の色分けに示すように

- 1) 領域 1 の列分割数＝領域 2 の列分割数（＝20）
- 2) 領域 2 の行分割数＝領域 3 の行分割数（＝19）

の関係を満たさないと、正しい要素分割が得られないことを 3.2 節で注意した。このため、本プログラムでは、自動的にこの関係が満たされるようにプログラミングして、ユーザーの便宜を図っている。

例えば、領域 1 の列分割数(本例では 20)の数値を入力したときに、同時に領域 2 の列分割数のセルが同じ数値(本例では 20)となる (fill-in, 自動補完) になっている。あるいは、先に領域 2 の列分割数 (20) を入力したときには、自動的に領域 1 の列分割数に 20 が fill-in される。領域 2 の行分割数と領域 3 の行分割数についても同様な fill-in が行われる。この結果、メッシュ細分割作業がより効率化される。

基本データ 節点データ 領域接続データ 領域分割データ 材料データ						
	領域番号	材料番号	行数	列数	Ndn1	Ndn2
	1	1	21	20	13	14
	2	1	19	20	3	4
	3	1	19	11	1	2
▶▶						

図 16 各領域の行分割数、列分割数の整合性

4.5 リナンバリングについて

図 14 に示したように、これまで示したブロック分割法では総節点数が 122、総要素数 198 となること
がわかる。これらの節点には、当初は、図 17 に示すように網の目の規則性に応じた番号付けがなされて
いる。

すなわち、領域 (1) では、格子点に 1 から 49 まで、領域 (2) では、格子点に 50 から 98 まで、領域
(3) では、格子点に 99 から 122 までに規則的に節点番号が振られる。しかし、この番号づけでは、要
素を囲む 3 節点の番号差を小さくするという配慮が行われていない。たとえば、領域 (3) の右上の要素
(要素番号 162) では、104, 43, 101 の節点番号となっていて番号差が 61 となっており、このままの節
点番号では、FEM の定式化後に得られる連立一次方程式を解くには、計算アルゴリズムの観点から適切
とはいえない。

そこで、本プログラムでは Collins が提案したバンド幅最小化アルゴリズム⁽²⁾によって節点番号のつけ
替えを行うようにしている。その実行結果を図 18 に示す。これにより、例えば要素番号 162 の要素では、
節点番号が 113, 114, 117 につけ変わり、節点番号差が 4 までに縮小されていることがわかる。なお、

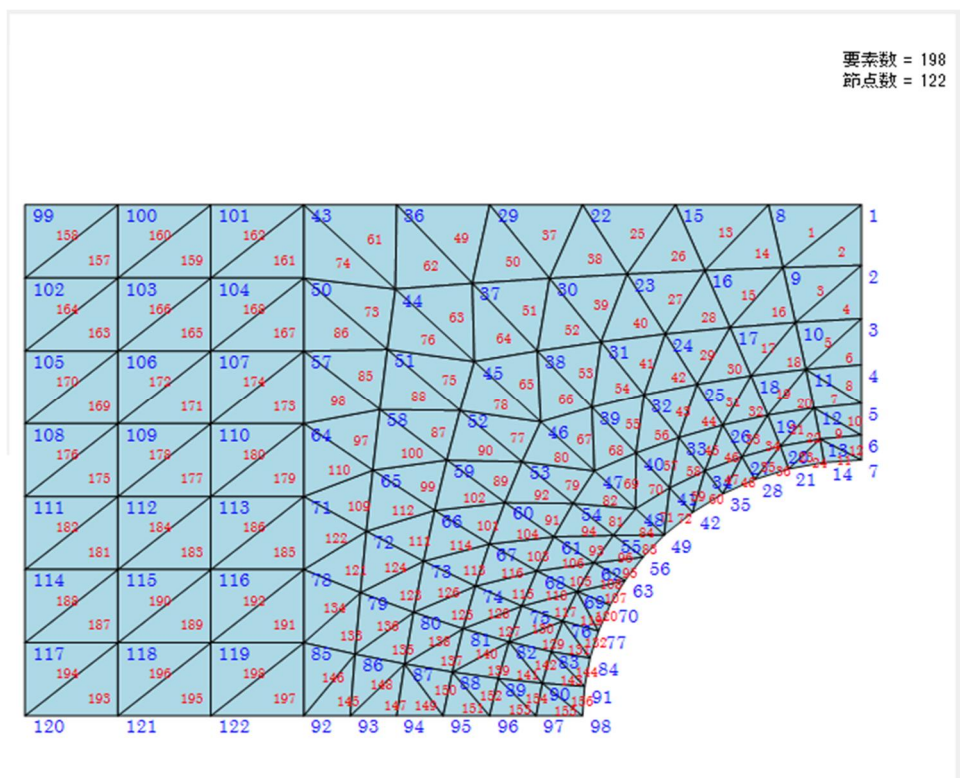


図 17 ブロック分割法によるナンバリング

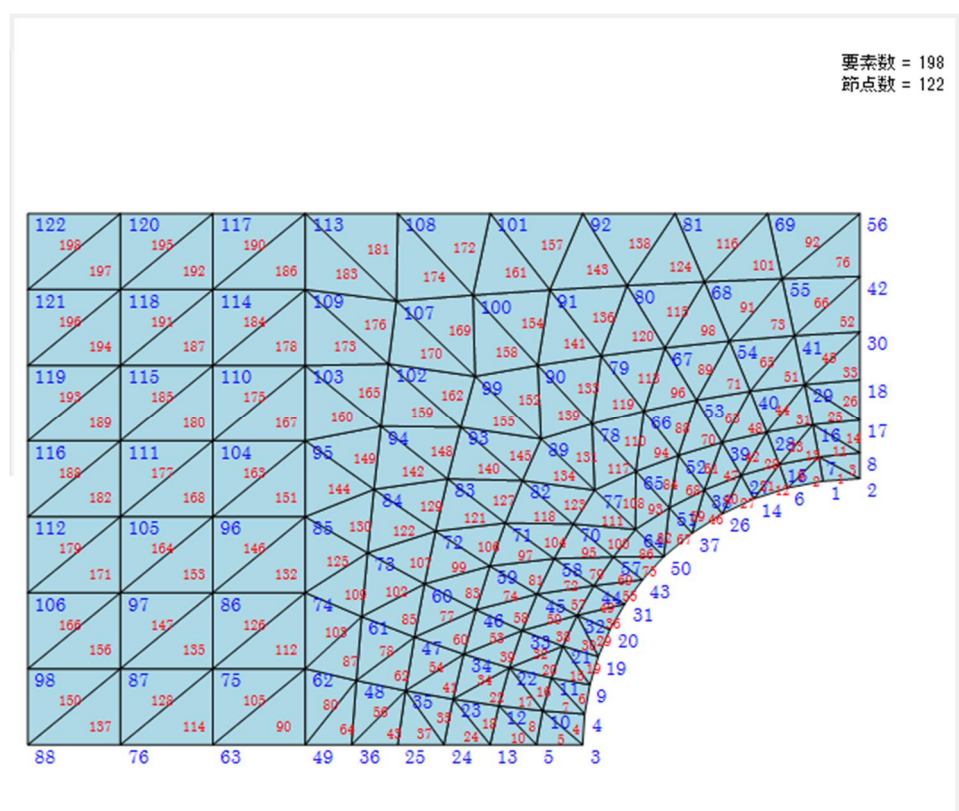


図 18 Collins のアルゴリズムによる節点番号のリナンバリン

この節点番号最適化処理は、ユーザーには見えない形で処理されているのでユーザーは特別に何かをす

る必要はない。

ここで、Collins のアルゴリズムとは、簡単に言えば、「節点番号が i 番目まで決定した場合、その次の節点番号 $i+1$ は、番号づけのされていない節点から、 i 番目の節点から最も近い距離（物理的な距離ではなく、グラフ理論に基づく距離、すなわち節点間の連結関係の階層数を指す）にある節点を選ぶ」という考え方である。この考えに従えば、各々の要素を囲む 3 節点の番号差が小さくなることを見込まれる。これは、FEM の全体剛性方程式の係数マトリックスの非対角成分が対角成分の近くに寄り集まってくることを意味し、結局は、連立一次方程式を効率的に解くことにつながる。

参考までに、図 1(c)の領域に対して、ファイルセーブしたときのデータ enkou.csv の実際の内容を以下に示す。

【enkou.csv の内容】

Two Dimensional Elastic Problem	13, 20.0, 40.0
3, 18, 1	14, 37.0, 23.0
1, 0.0, 0.0	15, 10.0, 40.0
2, 10.0, 0.0	16, 0.0, 40.0
3, 20.0, 0.0	17, 0.0, 20.0
4, 30.0, 0.0	18, 20.0, 20.0
5, 40.0, 0.0	1, 2, 0, 0, 0
6, 41.52, 7.65	2, 0, 0, 1, 3
7, 45.85, 14.14	3, 0, 2, 0, 0
8, 52.35, 18.48	1,1,7,7,13,14,7,8,9,10,11,12
9, 60.0, 20.0	2,1,8,7, 3, 4,5, 6, 7,14,13,18
10, 60.0, 27.5	3,1,8,4, 1, 2,3,18,13,15,16,17
11, 60.0, 40.0	1, 206e3, 0.3, 4.0
12, 40.0, 40.0	

4.6 拘束条件および荷重条件の設定

次に、円孔を有する帯板が左右に引張り力を受ける場合について、その拘束条件の設定例を以下に示す。

帯板の中央は左右に移動しないから、図 19 の辺 CD の x 方向の変位はゼロであり、これを拘束条件と考える。また、図 19 の辺 AB の y 方向変位もゼロとなるから、これも拘束条件と考える。

そこで、メニューバーの「変位拘束」→「 x 方向変位拘束」とたどり、「OK」をクリックして、右ボタンで x 方向変位を拘束したい節点をラバーバンドで囲むと図 14 のような画

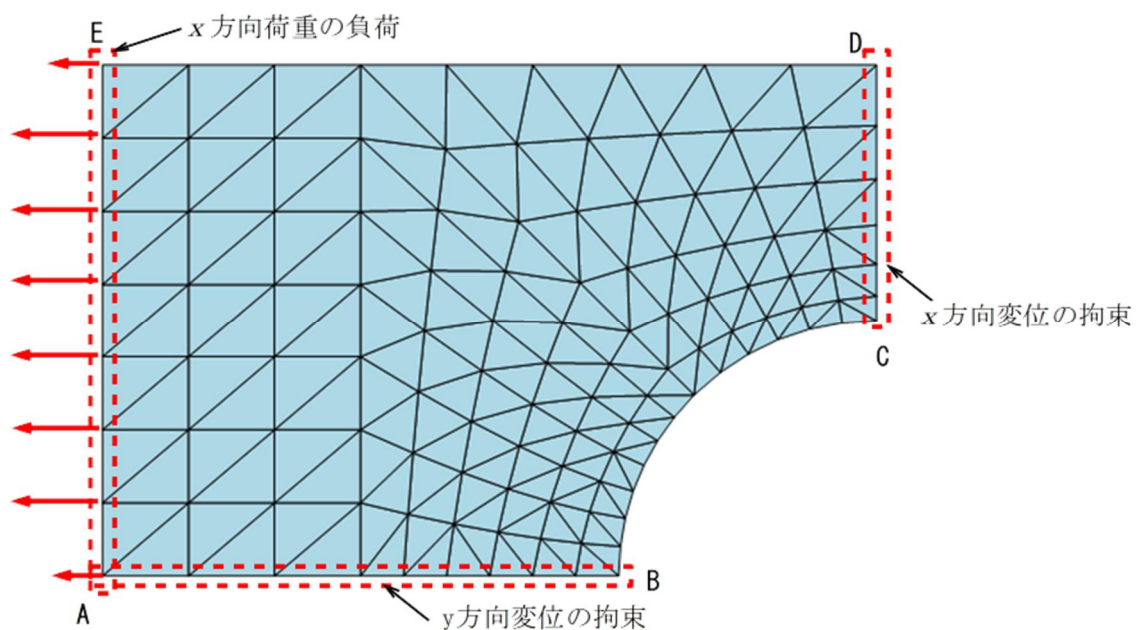


図 19 円孔問題の境界条件と荷重条件

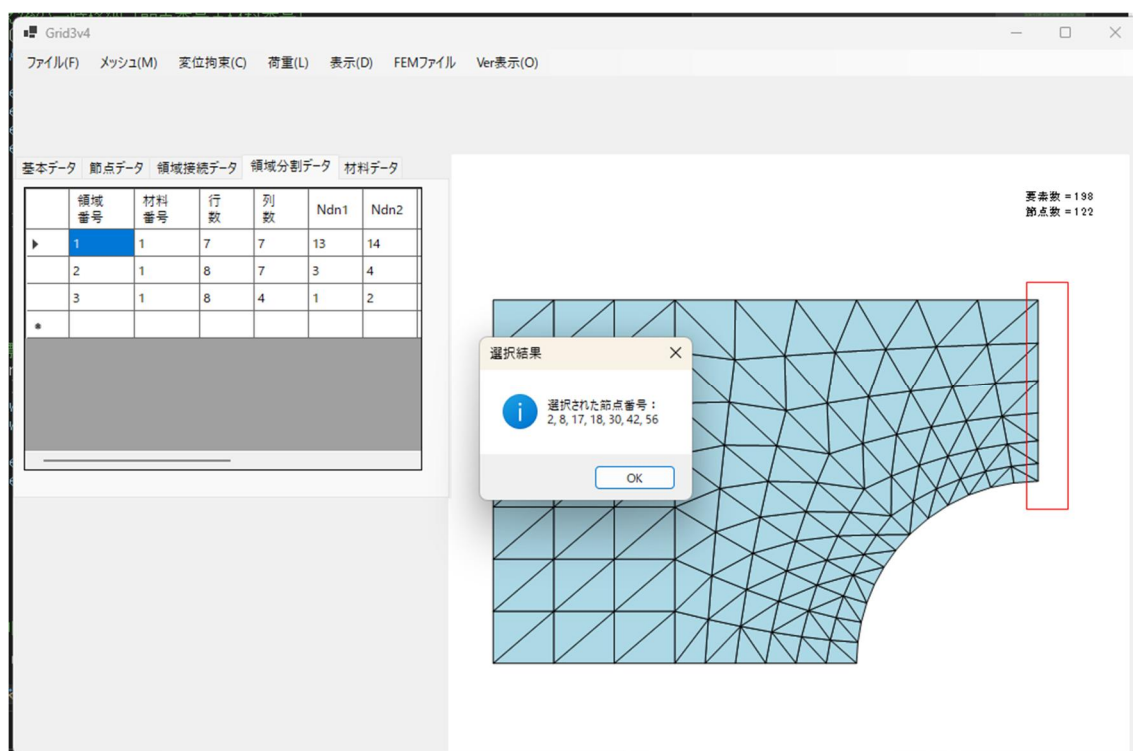


図 20 円孔問題の x 方向変位拘束

面になる。これにより、7 個の節点を選ばれ、x 方向の拘束変位量（ゼロがデフォルト）を 0 のまま「OK」をクリックすればよい。同様に、辺 AB についても y 方向変位拘束量をゼロと指定する操作を行う。

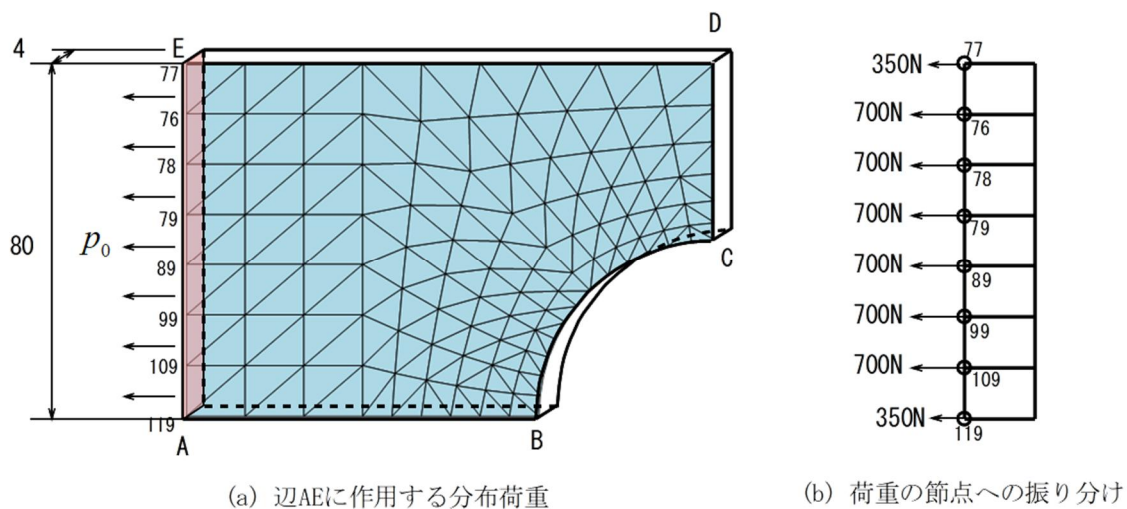


図 21 荷重の節点への振り分け

荷重を指定するには、以下のように考えればよい。図 1(a)の帯板を左右から 9800 N で引張る場合を想定する。すると、図 21(a)の左側の垂直な辺 AE には 4900 N の力が作用する。これを垂直辺の面積（ A とする）あたりの力 p_0 で考えると

$$p_0 = \frac{P}{A} = \frac{4900}{40 \times 4} = 30.625 \approx 30.6 \text{ N/mm}^2$$

(1)

が左辺に作用することになる。

また、辺 AE には同じ大きさの 7 個の要素が縦に並んでいるので、一つの要素には、左方に

$$F = p_0 \frac{A}{7} = \frac{P}{7} = \frac{4900}{7} = 700 \text{ N} \quad (2)$$

の力が作用する。荷重が作用する要素の荷重面には 2 つの節点が配置されているので、最終的には一つの節点あたりに 350 N の節点力が作用することになる。これらの力をそれぞれの節点に加えると、結局、全荷重 9800N は図 21(b)のような節点力として分配される。

以上の節点荷重値をもとに、メニューバーから、「荷重」→「x 方向荷重」とたどり、図 21(b)の 700 N が作用する節点をラバーバンドで囲んで、x 方向荷重値として「-700」を入力すればよい（負号を忘れないこと）。同様に、350 N の作用する節点を選んで x 方向荷重値として「-350」を入力する（この操作は、上端の節点 122 と下端の節点 88 に対して 2 回繰り返す）

変位拘束条件や荷重条件を表示するには、メニューバーの「表示」→「境界条件」とす

ればよく、図 21 のような拘束記号、荷重ベクトルが表示される。

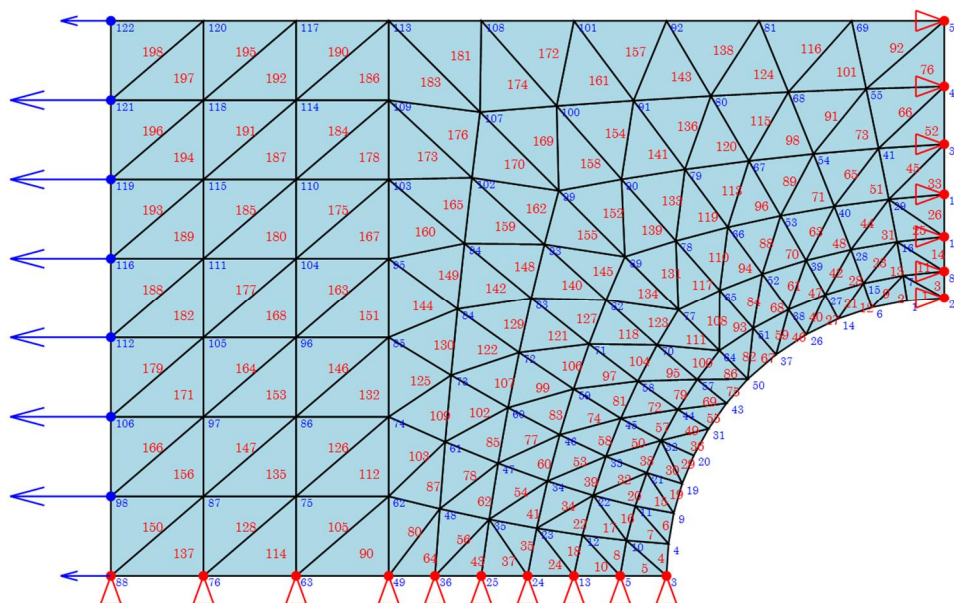


図 21 拘束条件、荷重条件の表示

4.7 ファイル出力

以上により、要素分割や拘束節点および荷重節点の指定を終えた。次は、メニューバーの「FEM ファイル」→「保存」とたどって、要素分割から荷重設定までのこれまでの結果を FEM 解析が必要とする形式でファイルとして出力する。ファイル名指定のダイアログボックスからファイル名（たとえば enkou122.dat）を入力してリターンキーを押すと

基本データ（ただし、総節点数、総要素数、材料数、拘束節点数、荷重節点数）

節点データ（節点番号, x , y 座標）

材料データ（ヤング率, ポアソン比, 板厚（平面応力の場合には実際の厚さ, 平面ひずみの場合には 0 が与えられる）

要素データ（要素番号, 左回りに要素を囲む節点番号, 材料番号）

拘束データ（拘束節点番号, x 拘束(0 or 1), x 拘束量, y 拘束(0 or 1), y 拘束量）

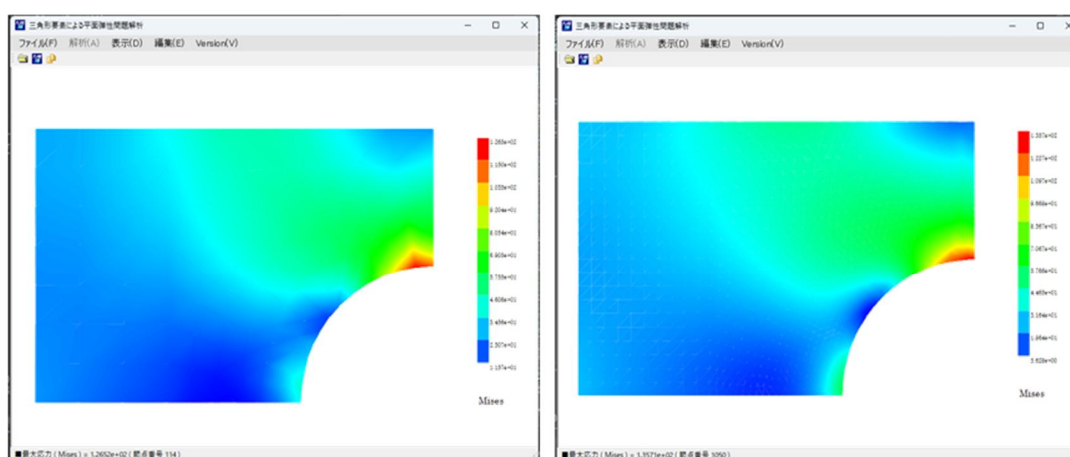
荷重データ（荷重節点番号, x 方向荷重, y 方向荷重）

が指定したファイル名で出力される。このファイルは、ただちに、三角形要素を用いた平面弾性解析プログラム plane2ds.exe から読み込むことができ、FEM 解析を実行できる。以下は、enkou122.dat の内容（一部省略）である。

122	,	1	,	198	,	17	,	8	1	,	3.3333e+01	,	0e+00
-----	---	---	---	-----	---	----	---	---	---	---	------------	---	-------

2 , 3e+01 , 0e+00	12 , 0 , 0 , 1 , 0
3 , 3.0656e+01 , 3.4278e+00	20 , 0 , 0 , 1 , 0
.....(途中省略).....	29 , 0 , 0 , 1 , 0
120 , 6e+01 , 2.4444e+01	39 , 0 , 0 , 1 , 0
121 , 6e+01 , 2.1944e+01	49 , 0 , 0 , 1 , 0
122 , 6e+01 , 2e+01	116 , 1 , 0 , 0 , 0
1 , 2.06E+10 , 0.3 , 4	117 , 1 , 0 , 0 , 0
1 , 108 , 109 , 116 , 1	118 , 1 , 0 , 0 , 0
2 , 109 , 117 , 116 , 1	119 , 1 , 0 , 0 , 0
3 , 109 , 110 , 117 , 1	120 , 1 , 0 , 0 , 0
.....(途中省略).....	121 , 1 , 0 , 0 , 0
	122 , 1 , 0 , 0 , 0
	53 , -700 , 0
	63 , -700 , 0
196 , 39 , 31 , 42 , 1	74 , -700 , 0
197 , 29 , 20 , 21 , 1	86 , -700 , 0
198 , 29 , 21 , 31 , 1	98 , -700 , 0
1 , 0 , 0 , 1 , 0	107 , -700 , 0
2 , 0 , 0 , 1 , 0	115 , -350 , 0
5 , 0 , 0 , 1 , 0	49 , -350 , 0
6 , 0 , 0 , 1 , 0	
11 , 0 , 0 , 1 , 0	2D FEM analysis by plane2ds

図 22(a)は、データ enkou122.dat を FEM 実行プログラム plane2ds.exe が読み込み、Mises



(a) 122節点の分割の場合のMises応力

(b) 970節点の分割の場合のMises応力

図 22 本プログラムの出力データによる FEM 解析結果 (Mises 応力のコンター図)

応力のコンター図の出力を行った例である．円孔周辺の応力集中の様子が正しく表現されていることがわかる．なお，ここに取り上げた例の節点数（122 節点）は，説明の便宜上，やや少ない分割数であったが，解析精度を高めたい場合にはより細かな要素分割が好ましい．図 22(b)は，図 15 のように 970 節点まで細かく分割した場合の Mises 応力のコンター図である．図 22(a)よりも滑らかな Mises 応力分布が得られていることがわかる．

5. その他の分割例

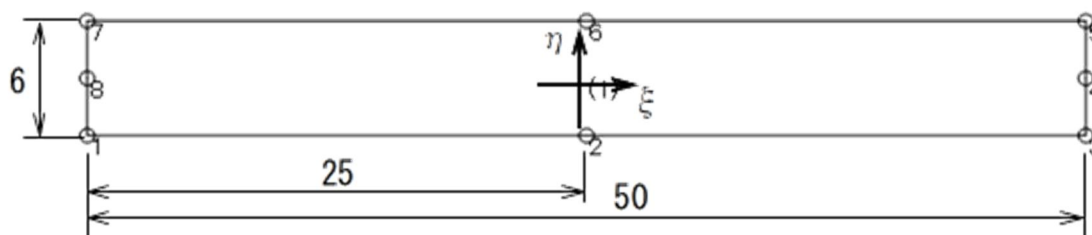
ほかの形状に対して，データおよび要素分割した 2, 3 の例を以下に示す．

【例題 1】（ファイル名：beam0.csv）

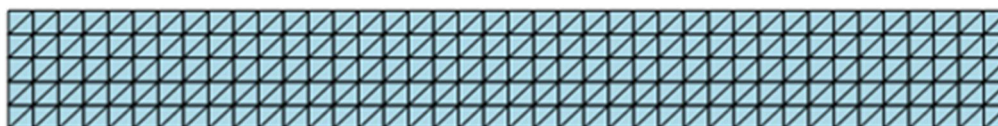
図 23(a)のような，はりのような横長の長方形領域の分割を考える．各辺の中間節点 2, 4, 6, 8 は，それぞれの辺の中央にあるものとする．これに対し，以下のような基本データを準備する．

Beam Subjected to Shear Force	6, 25.0, 6.0
1, 8, 1	7, 0.0, 6.0
1, 0.0, 0.0	8, 0.0, 3.0
2, 25.0, 0.0	1, 0, 0, 0, 0
3, 50.0, 0.0	1, 1, 6, 41, 1, 2, 3, 4, 5, 6, 7, 8
4, 50.0, 3.0	1, 206e3, 0.3, 1.0
5, 50.0, 6.0	

ここで，領域を 6 本の水平線および 41 本の垂直線で格子状に分けて三角形要素を作成する



(a) 細長い長方形領域



(b) 3角形要素分割結果

図 23 細長い長方形領域の三角形要素分割

ものとする．この結果，図 23(b)のような，総節点数 248 個で同じ大きさの三角形要素から成る分割図を得る．

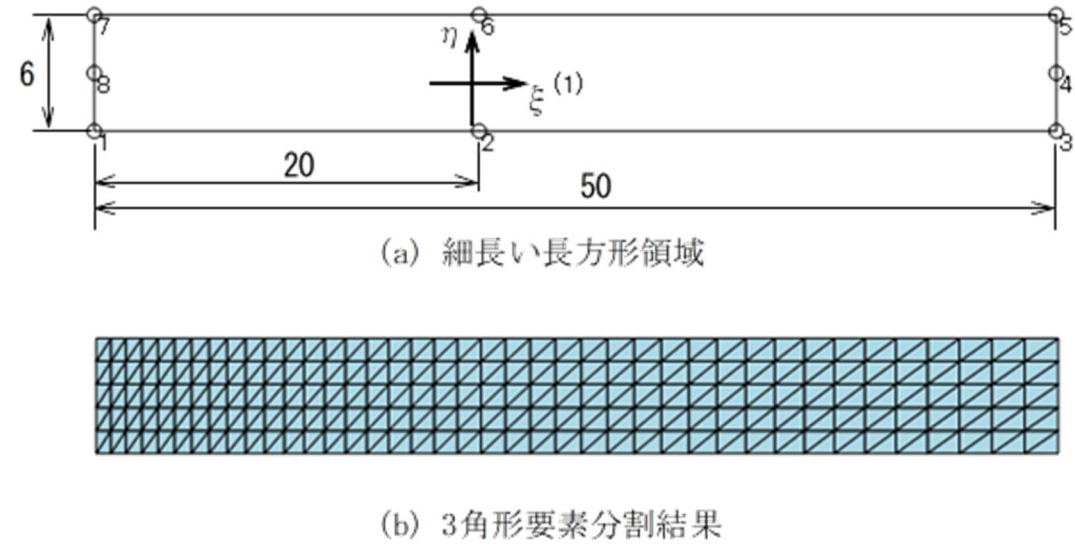


図 24 長方形領域の三角形要素分割（中間節点 2, 6 を左方に移動した場合）

次に，図 23(a)の中央節点 2, 6 の x 座標を 25.0 から 20.0 に左方に移動させた図 24 (a) の分割を考える（ファイル名：beam1.csv）．この場合の三角形要素分割では，図 24(b)に示すように，中間節点を移動させた側に密に要素が生成される．このように，縦横の格子線の分割線数のほかに，中間節点を移動させることによって要素分割の粗密さを調整することができる．

【例題 2】 L 字形領域（ファイル名：LshapeBeam.csv）

図 25 のような寸法を与えた L 字形領域の分割を考える．領域は 3 個に分けられるが，それぞれの弾性係数が異なるものとする．領域（1）では， $E_1=206 \times 10^3 \text{ N/mm}^2$ ，領域（2）では， $E_2=100 \times 10^3 \text{ N/mm}^2$ ，領域（3）では， $E_3=70 \times 10^3 \text{ N/mm}^2$ ，とする．

基本データ（3 個の材料を想定）

L shaped region	5, 10.0, 0.0
3, 18, 3	6, 0.0, 2.5
1, 0.0, 0.0	7, 5.0, 2.5
2, 2.5, 0.0	8, 10.0, 2.5
3, 5.0, 0.0	9, 0.0, 5.0
4, 7.5, 0.0	10, 2.5, 5.0

11, 5.0, 5.0	2, 0, 0, 0, 1
12, 7.5, 5.0	3, 1, 0, 0, 0
13, 10.0, 5.0	1, 1, 10, 10, 1, 2, 3, 7, 11, 10, 9, 6
14, 0.0, 7.5	2, 2, 10, 10, 3, 4, 5, 8, 13, 12, 11, 7
15, 5.0, 7.5	3, 3, 10, 10, 9, 10, 11, 15, 18, 17, 16, 14
16, 0.0, 10.0	1, 206e3, 0.3, 1.0
17, 2.5, 10.0	2, 100e3, 0.3, 1.0
18, 5.0, 10.0	3, 70e3, 0.3, 1.0
1, 0, 2, 3, 0	

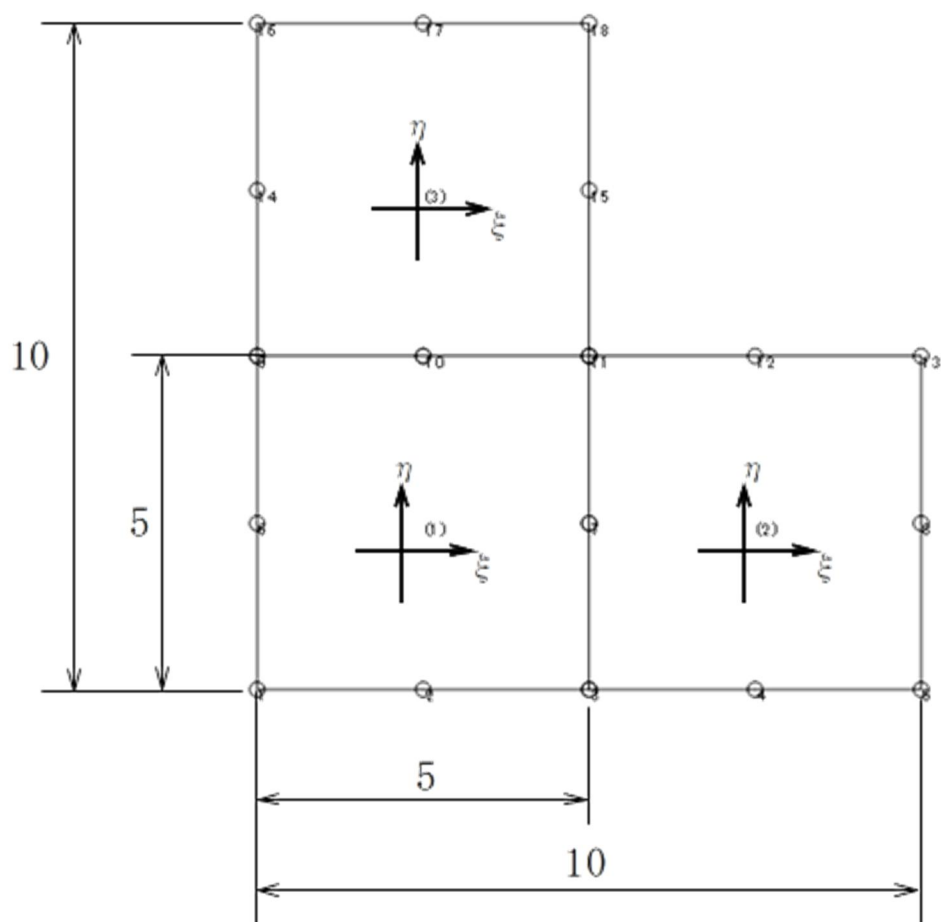


图 25 L 字形領域

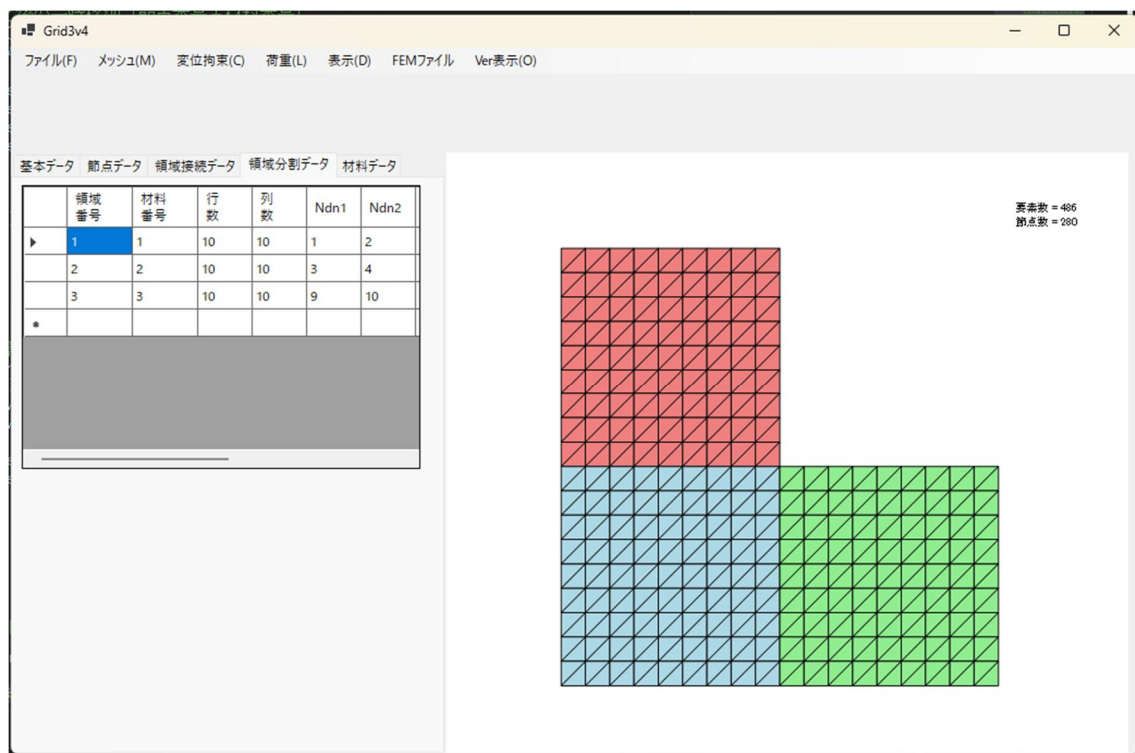


図 26 L 字形領域の三角形要素分割結果 (280 節点)

このデータに従って分割した結果を図 26 に示す。

【例題 3】 穴の空いた部分円輪（ファイル名：frange2.csv）

図 27 のように、 $1/4$ の部分円輪（内径 50mm，外径 100mm）に，直径 $d=20\text{mm}$ の穴が空いた領域の要素分割を考える．穴の位置は， x 軸から 45° の方向で，円輪幅の中央に中心があるものとする．この場合，図のような 6 個の領域に分けた．なお，ここでは基本データの内容については省略し，興味のある方は frange2.csv の DataGrid を参照すること．

図 28 は，穴の空いた部分円輪を三角形要素に分割した結果である．本例の場合の総節点

数は、540 個である。

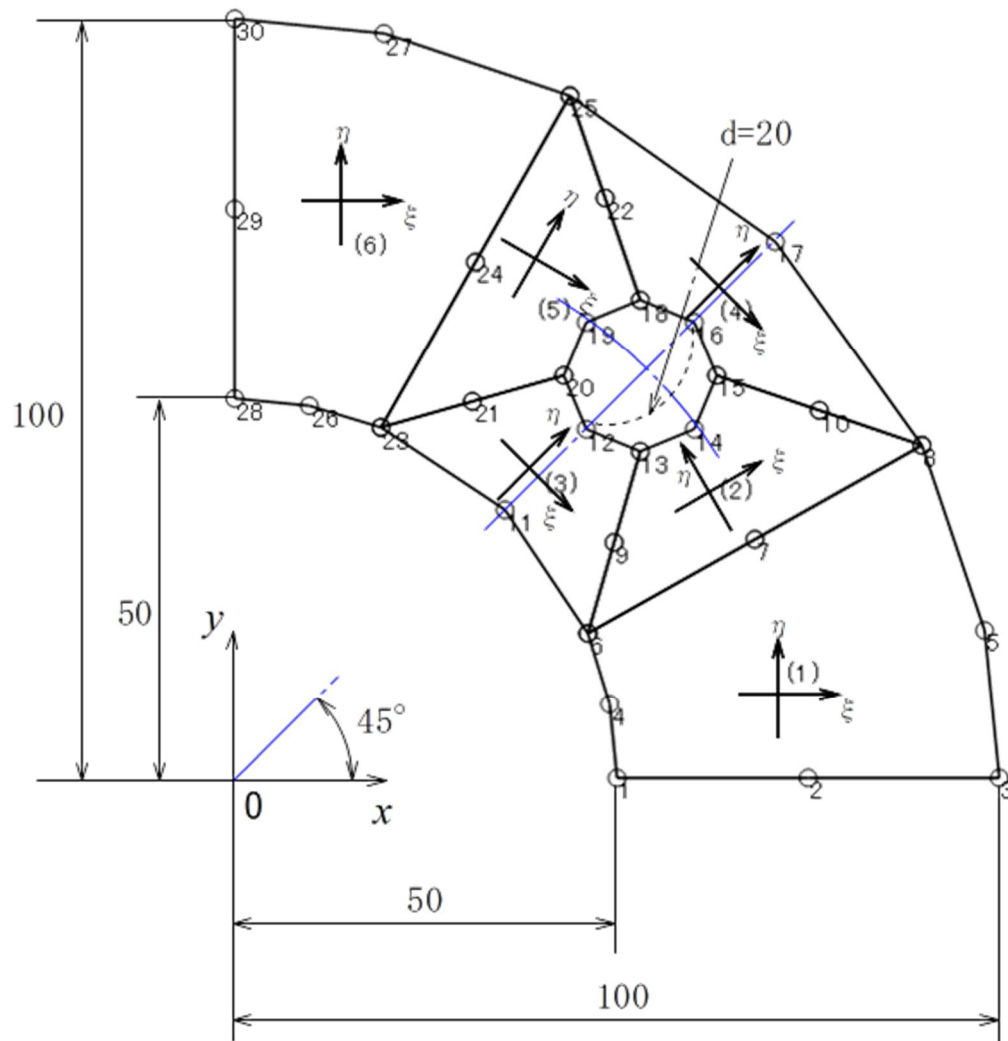


図 27 穴の空いた部分円輪領域

6. 利用上の注意、著作権ほか

かつて有限要素法 (FEM) は、ごく一部のエンジニアが利用する特殊なソフトウェアであった。しかし、現在ではほとんどのエンジニアに広く認知されるソフトウェアとなっている。車の性能や使い勝手が向上すればするほど、車のボンネットを開ける者が少なくなってきたと言われている。これと同様に、有限要素法もブラックボックス化が進み、その動作原理にまで思いを巡らす者は少なくなっているように思われる。

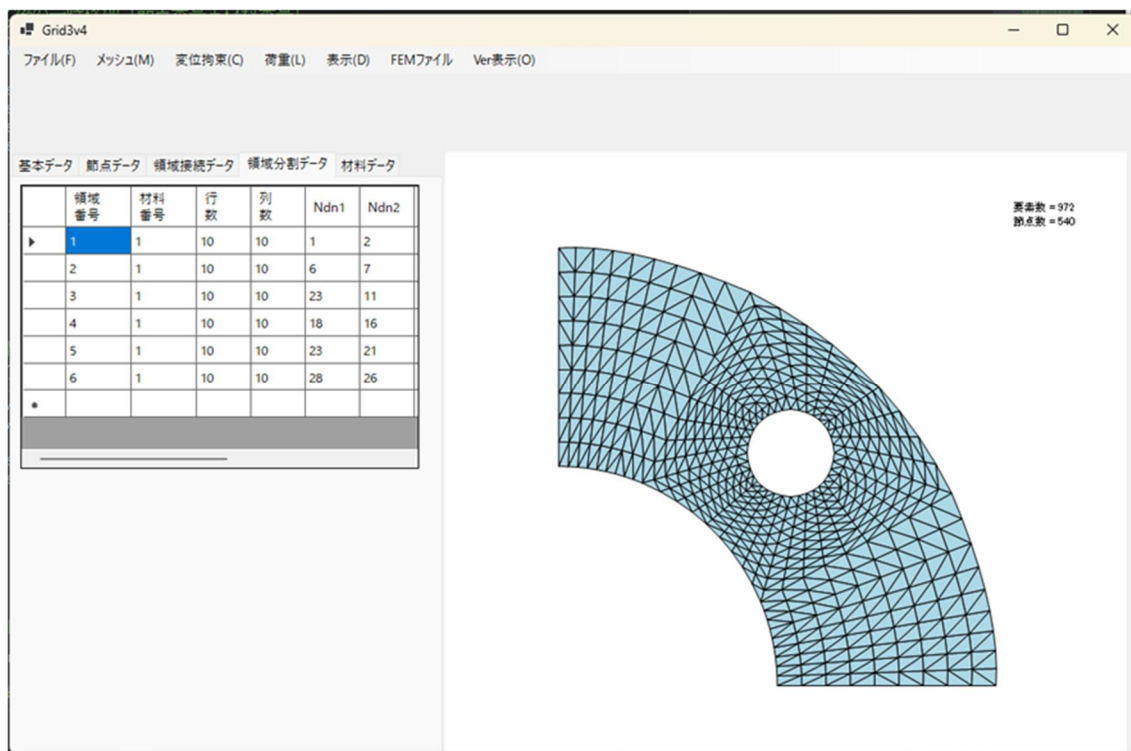


図 28 穴の空いた部分円輪の三角形要素分割結果（540 節点）

しかしながら、有限要素法がどのようなプロセスを経て動作しているのかを理解していれば、それを的確に、かつ正しく使いこなすことが可能となる。

近年の汎用 FEM ソフトは非常に高機能であるが、利用者にとって必要なのはその一部に限られる場合が多い。簡単な問題を汎用ソフトで解くのは、たとえば言えば、近所のコンビニエンスストアへ買い物に行くのに大型バスを使うようなものである。近所への買い物であれば、自転車で十分である。

このような問題意識のもと、小回りが利き、プログラムの内部構造が見通しやすい FEM ソフトが必要であると考え、本ソフトや FEM プログラムの開発に至っている。

本プログラムは、基本的にメッシュ作成に関する入門用として開発されたものであり、一般的な実務への利用を意図していない。このため、演習以外の目的で使用された場合の結果については、一切の保証を行わない。

なお、本プログラムは自由に利用可能であるが、著作権は著者に帰属しており、無断での複製や第三者への配布はご遠慮いただきたい。また、本プログラムには未だバグや操作上の不具合が潜在している可能性がある。万一不具合を発見された場合には、以下の連絡先までご一報いただけると幸いである（連絡先：tadashihoribe@gmail.com）。

参考文献

- (1) L. J. Segerlind 著, 川井監訳, 「応用有限要素解析」, 丸善 (1982).
- (2) R. J. Collins, Bandwidth Reduction by Automatic Renumbering, Int. J. for Num. Methods in Engng., Vol.6, 345-356(1973).
- (3) 戸川, 有限要素法概論, 培風館 (1980).
- (4) 堀辺, Visual Basic でわかるやさしい有限要素法, 森北出版 (2008).
- (5) 堀辺, 例題で学ぶ材料力学, 森北出版 (2022).
- (6) 中原ほか 5 名, 弾性学ハンドブック, 朝倉書店 (2002).
- (7) 朝井, 3 ステップでしっかり学ぶ Visual Basic 入門, 技術評論社 (2023).