

GB001: ラプラス方程式、ポアソン方程式

本 whitepaper は Gunnar Backstrom 氏の承諾のもと、書籍“*Simple Fields of Physics by Finite Element Analysis*” に記されている多数の FlexPDE 適用事例の中からその一部を紹介するものです。

本 whitepaper では厳密解のわかっているラプラス方程式、ポアソン方程式に FlexPDE を適用し、有限要素法による数値解の演算精度を検証します。

1 ラプラス方程式 - Case1

ここでは 2 次元の正方形領域において変数 $U(x, y)$ に関するラプラス方程式

$$\Delta U \equiv \text{div}(\text{grad}(U)) = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = 0 \quad (1)$$

について考えることにします。実際の物理的な問題に FlexPDE を適用する前に、その計算精度を確認する意味で解析解との対比をまず行ってみましょう。良く知られているように正則な複素関数の実数部、あるいは虚数部はラプラス方程式を満足します。そこでここでは

$$U_{ex} = \text{Re}(z^2) = x^2 - y^2 \quad (2)$$

という関数を想定し、境界上でこの関数値を指定することになります(Dirichlet 型境界条件)。もちろん FlexPDE はこれが厳密解であることなど認識できるわけではないので、ひたすら有限要素法に基づき数値解を計算します。

1.1 Problem descriptor [laplace01a.pde]

まずタイトルを設定します。

```
TITLE
'Laplace Equation'      { laplace01a.pde }
```

次に演算精度に関するセクタをセットします。デフォルトは 0.002 なのですが、問題の素性が良いため、ここでは精度を $1e-5 (= 10^{-5})$ に引き上げることにします。

```
SELECT
  Errlim = 1e-5          { Requested relative accuracy }
```

従属変数を定義します。

```
VARIABLES
  U          { Unknown }
```

偏微分方程式の定義に先立ち、パラメータ類を定義します。これらは境界の定義、及び境界条件の設定に際して使用されます。

```
DEFINITIONS
  Lx = 1  Ly = 1
  U_ex = x^2 - y^2          { Exact solution }
```

ラプラス方程式を定義します。

```
EQUATIONS
  div(grad(U)) = 0          { Laplace PDE }
```

次に境界の形状と境界条件を設定します。一辺が2の正方形の境界上で関数 U の値が数式 (2) に従って与えられます。境界条件は各辺で共通なので指定は一度だけで済みます。

```
BOUNDARIES
  Region 1          { Set exact values on boundaries }
  Start(-Lx, -Ly)
  Value(U) = U_ex  Line to (Lx, -Ly)
                   to (Lx, Ly)
                   to (-Lx, Ly)
                   to Close
```

最後に出力すべき情報を指定します。

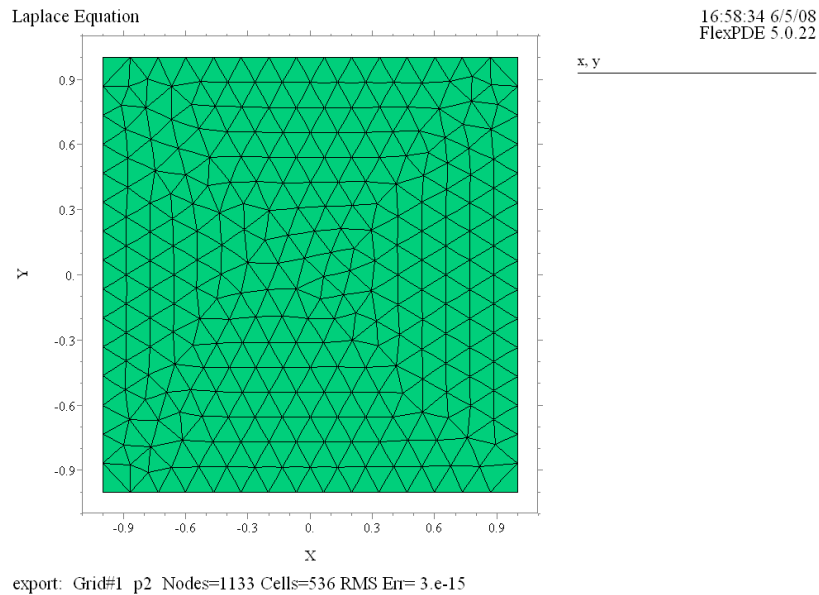
```
PLOTS
  Grid(x, y)
  Contour(U)
  Surface(U)
  Contour(U_ex)
  Contour(U - U_ex)
```

```
END
```

1.2 実行結果

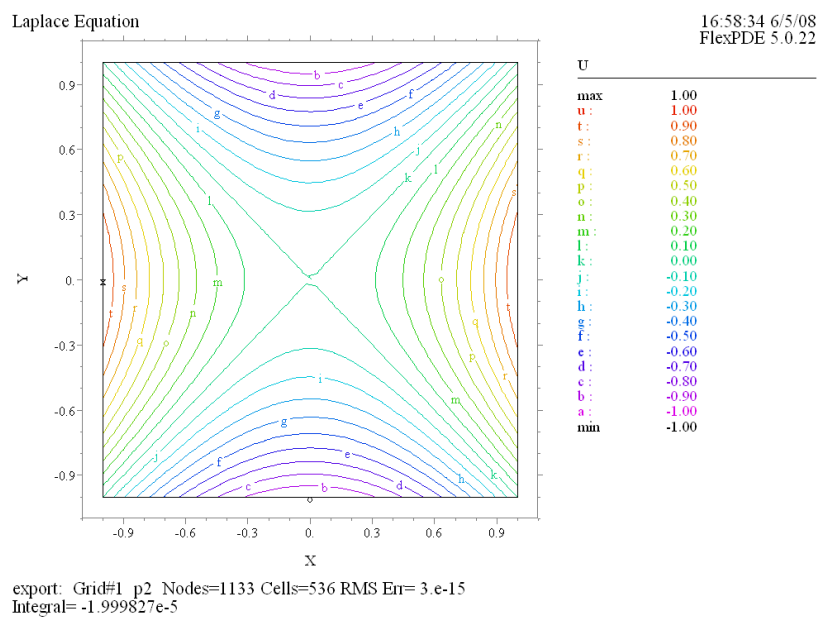
(1) Grid(x, y)

問題が単純なため、計算に使用されたメッシュも没個性的なものとなっています。なお誤差の指標 (RMS: 平均 2 乗偏差 (Root Mean Square)) が極めて小さな値となっている点に注意ください。



(2) Contour(U)

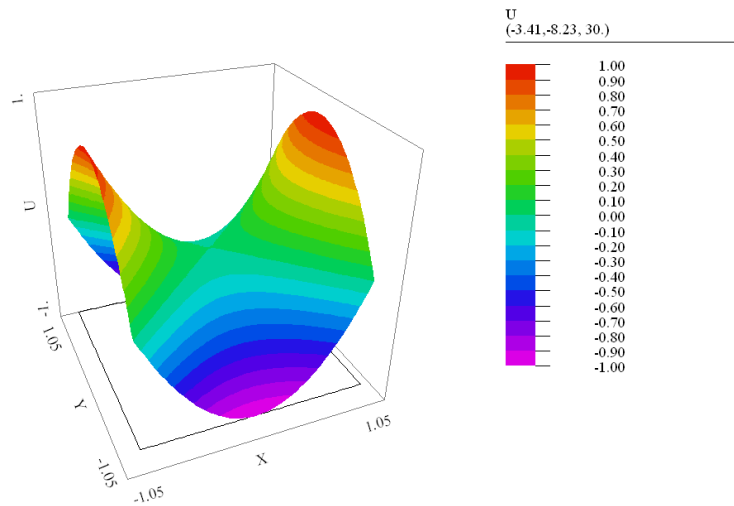
解析対象領域 (ドメイン) 上での関数 $U(x, y)$ の等高線図は次のようになります。



(3) Surface(U)

関数 $U(x, y)$ の曲面の形状をプロットしたものです。

Laplace Equation

16:58:34 6/5/08
FlexPDE 5.0.22

export: Grid#1 p2 Nodes=1133 Cells=536 RMS Err= 3.e-15
Integral=-1.999827e-5

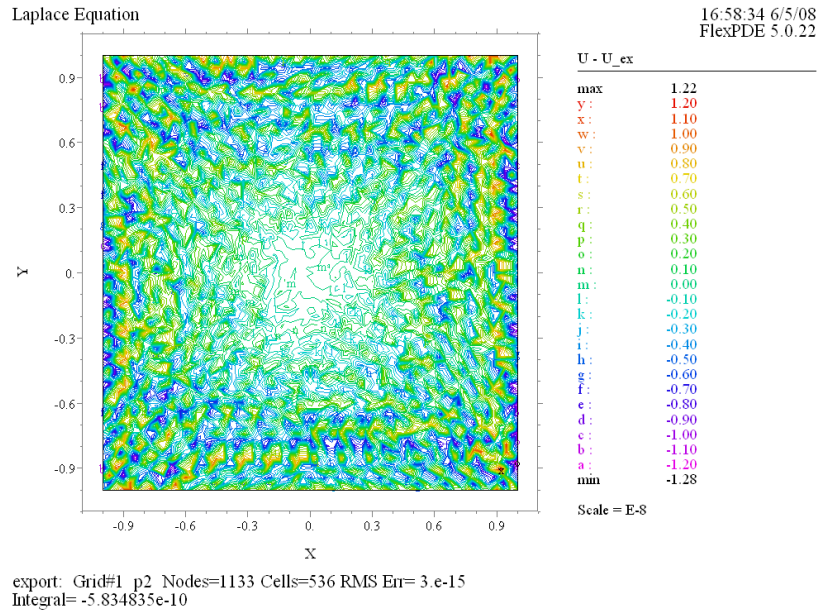
(4) Contour(U_ex)

厳密解の等高線図ですが、見た目は Contour(U) と全く変わらないため省略します。

(5) Contour(U - U_ex)

次ページの図は FlexPDE の計算結果と理論曲面との差をプロットしたものです。スケールが $E-8(= 10^{-8})$ という表示になっている点にご注意ください。

ここでは正方形領域を対象に計算結果の検証を行ったわけですが、境界の形状が円形であっても三角形であっても同様の操作で行えます。なお境界上で値が指定されたとき、その条件を満たすラプラス方程式の解は一つに限られる点にご注意ください。



2 ラプラス方程式 - Case2

Case1 の例ではすべて Dirichlet 型の境界条件を設定しましたが、ここでは左右の境界上で法線微分の値を設定してみることにします (Neumann 型境界条件)。すなわち

$$\text{grad}(U) \cdot \mathbf{n} = \frac{\partial U}{\partial n} \quad (3)$$

の関数式を指定しようというわけです。ここで \mathbf{n} は外向きの法線である点に注意してください。境界 $x = 1$ 上では $\frac{\partial U}{\partial x} = 2x$ で良いわけですが、境界 $x = -1$ 上では $\frac{\partial U}{\partial(-x)} = -2x$ と指定することになります。なお、このタイプの境界条件を FlexPDE で指定するときには Natural 文を使用します。

2.1 Problem descriptor [laplace01b.pde]

Case1 で作成したスクリプト laplace01a.pde に対し修正を施した部分のみを記すことにします。

```
TITLE
  'Laplace Equation'      { laplace01b.pde }
  ...
```

BOUNDARIES

Region 1

```

Start(-Lx, -Ly)
Value(U) = U_ex   Line to (Lx, -Ly)
Natural(U) = 2*x   Line to (Lx, Ly)
Value(U) = U_ex   Line to (-Lx, Ly)
Natural(U) = -2*x  Line to Close

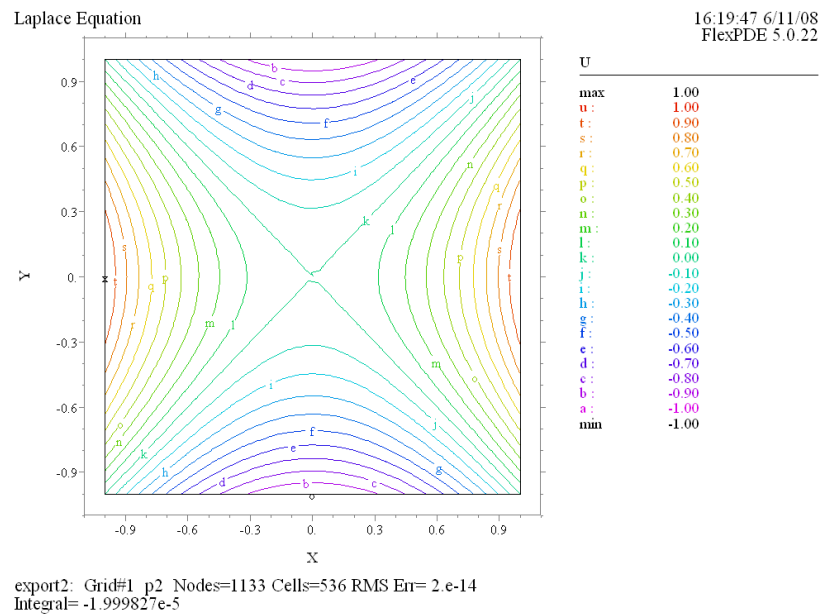
```

...

2.2 実行結果

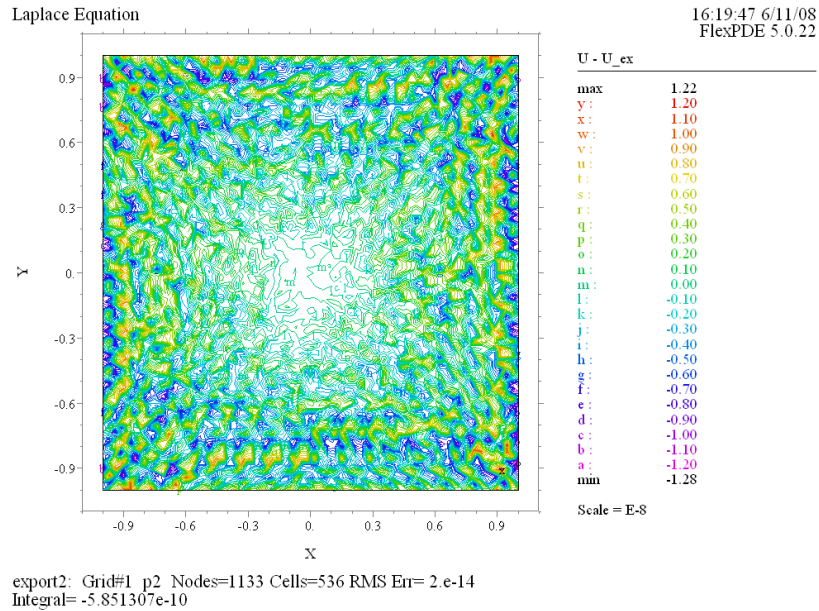
結果は laplace01a.pde の場合とほとんど差はありません。 $U(x, y)$ と誤差の等高線図のみを示しておくことにします。

(1) Contour(U)



(2) Contour(U - U_ex)

今回もスケールが E-8 という表示になっている点にご注意ください。



3 ラプラス方程式 - Case3

今回は極端な例として4辺すべてにおける境界条件を法線微分の形で与えてみましょう。

3.1 Problem descriptor [laplace01c.pde]

Case2 で作成したスクリプト laplace01b.pde に対し修正を施した部分のみを記すことにします。正方形下辺における法線微分は下向きになるので、 $-2y$ ではなく $2y$ となる点に注意してください。

```

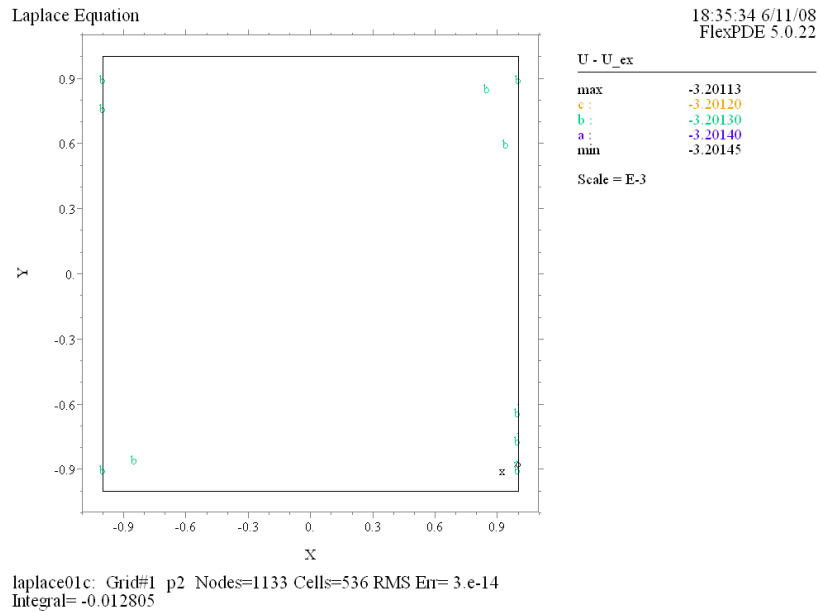
TITLE
  'Laplace Equation'      { laplace01c.pde }
  ...
BOUNDARIES
  Region 1
    Start(-Lx, -Ly)
    Natural(U) = 2*y      Line to (Lx, -Ly)
    Natural(U) = 2*x      Line to (Lx, Ly)
    Natural(U) = -2*y     Line to (-Lx, Ly)
    Natural(U) = -2*x     Line to Close
  ...

```

3.2 実行結果

ここでは誤差の等高線図のみを記しておきますが次の通りです。

(1) Contour (U - U_ex)



誤差のレベルが $E-3$ という比較的大きな値、しかもほぼ一定値となっています。良く考えてみればわかることですが、微係数だけでは関数を一意的に決定することができません。しかし 1 点でも良いから関数値を指定できれば、微係数だけの境界条件でも解に到達することができます。

3.3 Problem descriptor [laplace01d.pde]

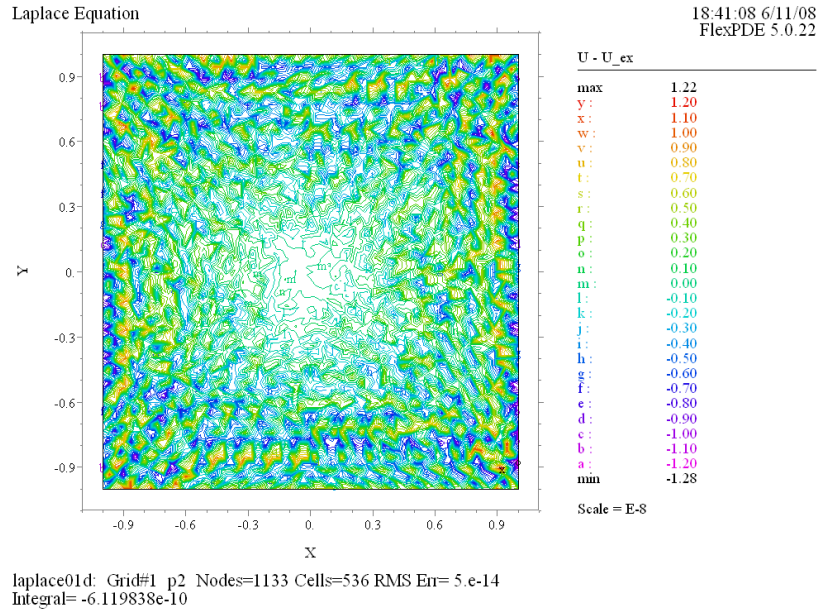
そこで境界定義の始点において関数値の指定を行っていきましょう。そのためには Point Value 文を使用します。スクリプトの変更部分は次の 1 文のみです。この Point Value 文は座標点の指定の直後に置く必要があります。関数値は該当の点にのみ適用され、他の境界条件には影響を与えません。

```
BOUNDARIES
...
    Start(-Lx, -Ly) Point Value(U) = U_ex
...
```


3.4 実行結果

やはり誤差の等高線図のみを記しておきますが、Case1 や Case2 と同等の精度となっています。

(1) Contour (U - U_{ex})



4 ラプラス方程式 - Case4

今度は解析対象のドメインを四分円としてラプラス方程式を解いてみます。検証用の厳密解としては

$$U_{ex} = \text{Im}(z^2) = 2xy \quad (4)$$

を想定し、境界条件はすべて Dirichlet 型で設定することにします。

4.1 Problem descriptor [laplace01e.pde]

Case1 で作成したスクリプト laplace01a.pde に対し修正を施したものを示しておきます。円弧の境界を設定するには Arc 文を使用します。

```
TITLE
  'Laplace Equation'      { laplace01e.pde }

SELECT
  Errlim = 1e-5          { Requested relative accuracy }
```

```
VARIABLES
  U                      { Unknown }

DEFINITIONS
  r0 = 1
  U_ex = 2*x*y           { Exact solution }

EQUATIONS
  div(grad(U)) = 0      { Laplace PDE }

BOUNDARIES
  Region 1               { Set exact values on boundaries }
  Start(0, 0)
  Value(U) = U_ex Line to (r0, 0)
                      Arc(Center = 0,0) to (0, r0)
                      Line to Close

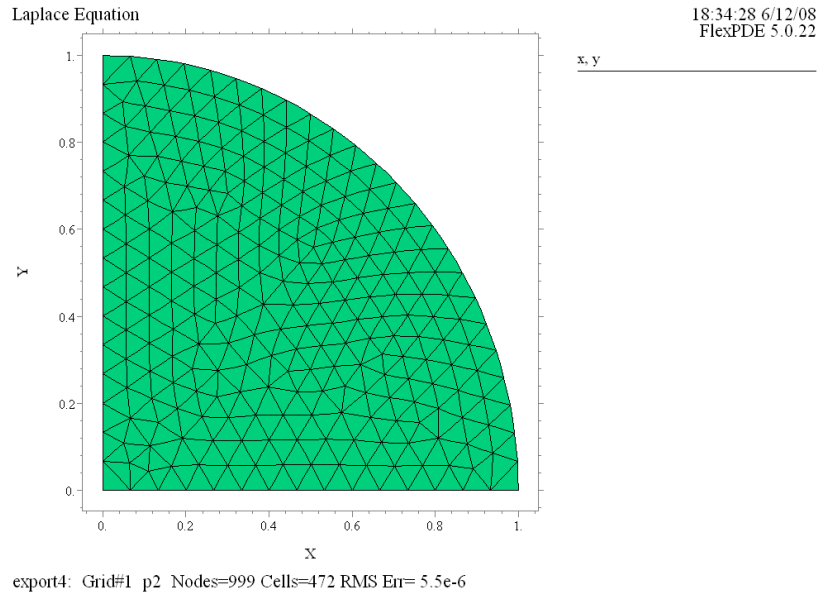
PLOTS
  Grid(x, y)
  Contour(U)
  Surface(U)
  Contour(U_ex)
  Contour(U - U_ex)

END
```

4.2 実行結果

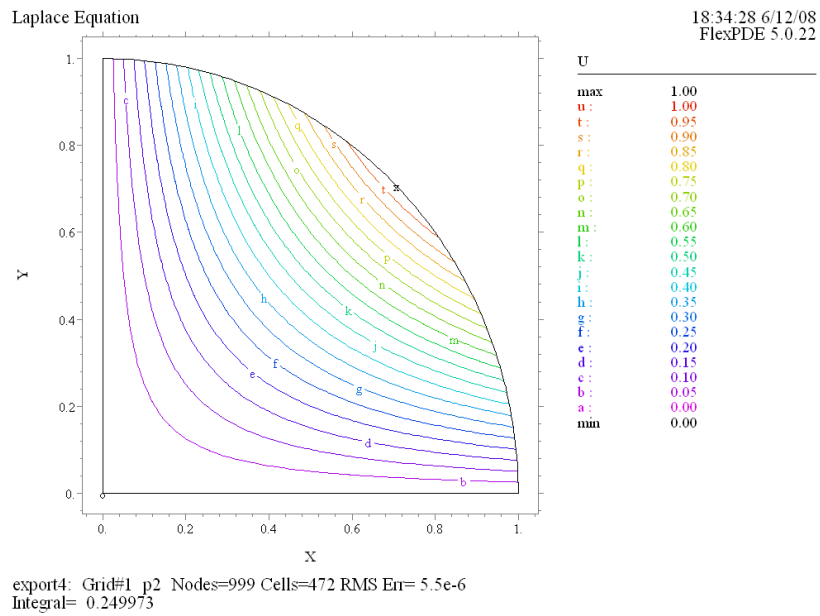
(1) Grid(x, y)

問題が単純なため、自動生成されたメッシュ構成も特に複雑なものとはなっていません。



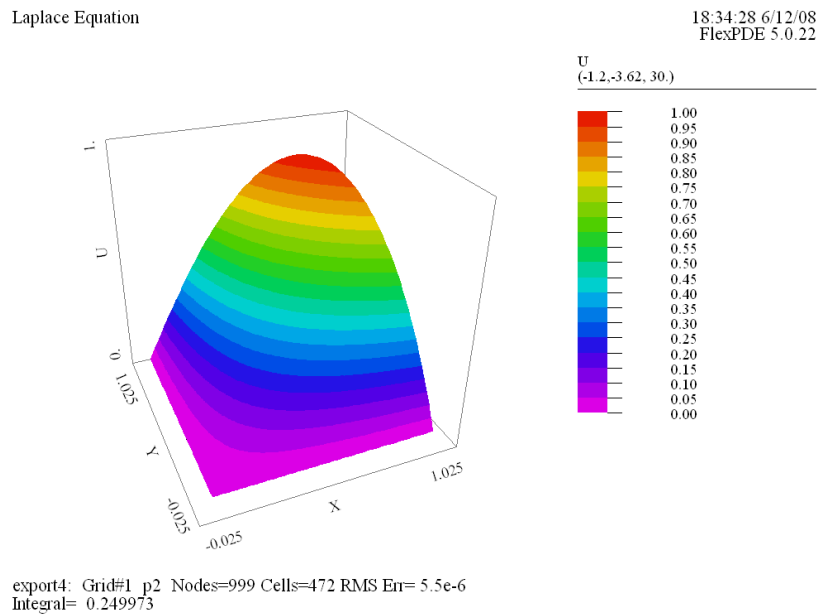
(2) Contour (U)

ドメイン上での関数 $U(x, y)$ の等高線図は次のようになります。



(3) Surface(U)

関数 $U(x, y)$ の曲面の形状をプロットしたものを示しておきます。

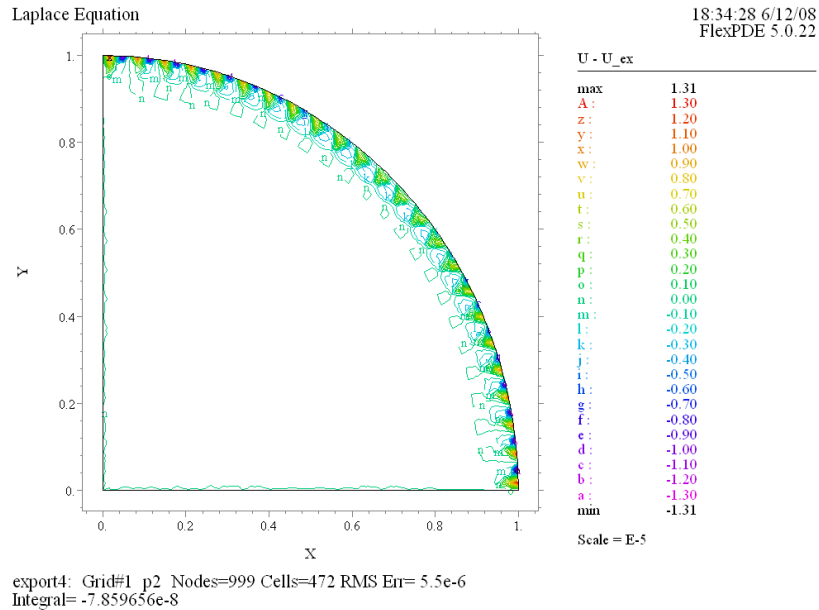


(4) Contour(U_ex)

厳密解の等高線図ですが、見た目は Contour(U) と全く変わらないため省略します。

(5) Contour(U - U_ex)

次ページの図は FlexPDE の計算結果と理論曲面との差をプロットしたものです。スケールは $E-5(= 10^{-5})$ ということ、矩形領域の場合に比べると悪くなっていますが、SELECT セクションで設定した誤差許容値のレベルには到達しています。境界に曲線を含むセル上で誤差が大きくなっています。



5 ポアソン方程式

ポアソン方程式はラプラス方程式に似てはいますが、右辺が 0 ではなく関数となる点が異なります。

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = f(x, y) \quad (5)$$

ポアソン方程式の厳密解を求めることは一般的に困難ですが、解を想定した上でそれに対応した PDE を導くことは簡単に行えます。例えば

$$U_{ex} = 2xy + (x^3 + 2y^2) \quad (6)$$

という解を想定した場合、第 1 項の $2xy$ は調和関数 (ラプラス方程式を満たす関数) ですから、ラプラシアンは 0 となります。従って数式 (5) の $f(x, y)$ に寄与するのは第 2 項の $(x^3 + 2y^2)$ の部分のみということになります。従って解 (6) が満足するポアソン方程式は次のようになります。

$$\Delta U \equiv \text{div}(\text{grad}(U)) = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = 6x + 4 \quad (7)$$

この PDE を FlexPDE で解いた結果がどの程度厳密解 (6) に一致するかを検証してみます。解析対象のドメインは正方形領域とし、境界上では関数 (6) の値を Value 文で与えるものとします。

5.1 Problem descriptor [laplace01f.pde]

```
TITLE
  'Poisson Equation'      { laplace01f.pde }

SELECT
  Errlim = 1e-3           { Requested relative accuracy }

VARIABLES
  U                        { Unknown }

DEFINITIONS
  Lx = 1  Ly = 1
  U_ex = 2*x*y + (x^3 + 2*y^2) { Exact solution }

EQUATIONS
  div(grad(U)) = 6*x + 4    { Poisson PDE }

BOUNDARIES
  Region 1                 { Set exact values on boundaries }
  Start(-Lx, -Ly)
  Value(U) = U_ex Line to (Lx, -Ly)
                        to (Lx, Ly)
                        to (-Lx, Ly)
                        to Close

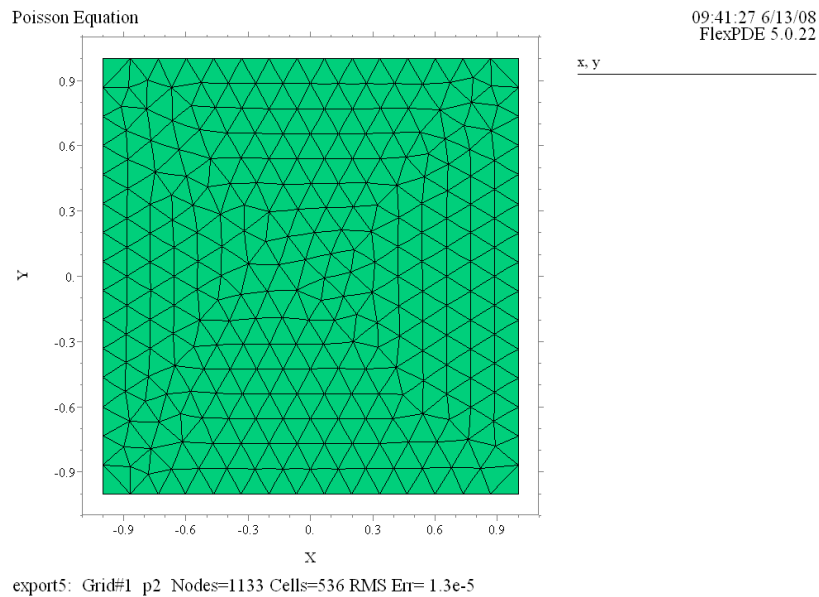
PLOTS
  Grid(x, y)
  Contour(U)
  Surface(U)
  Contour(U_ex)
  Contour(U - U_ex)

END
```

5.2 実行結果

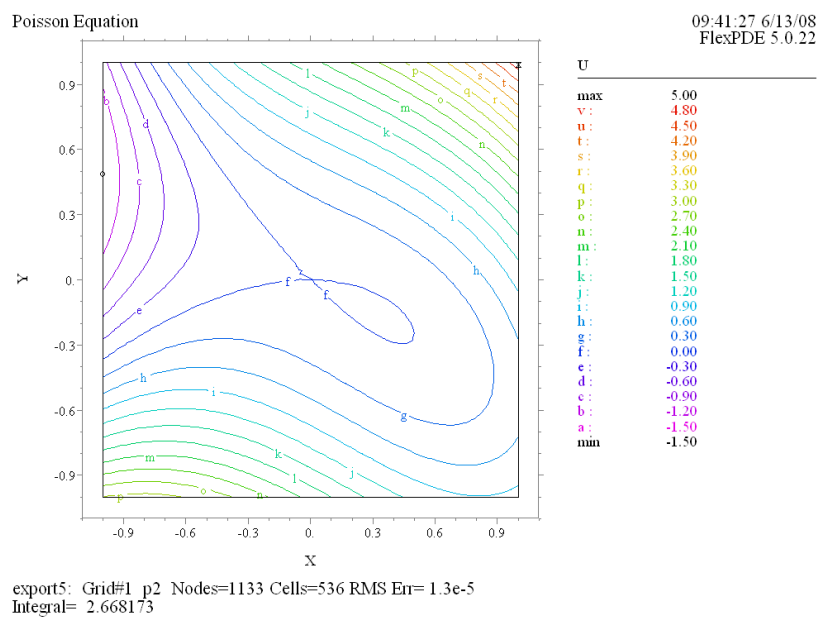
(1) Grid(x, y)

設定した演算精度が $1e-3$ とこれまでのものに比べると低いため、自動生成されたメッシュ構成も特に複雑なものとはなっていません。



(2) Contour(U)

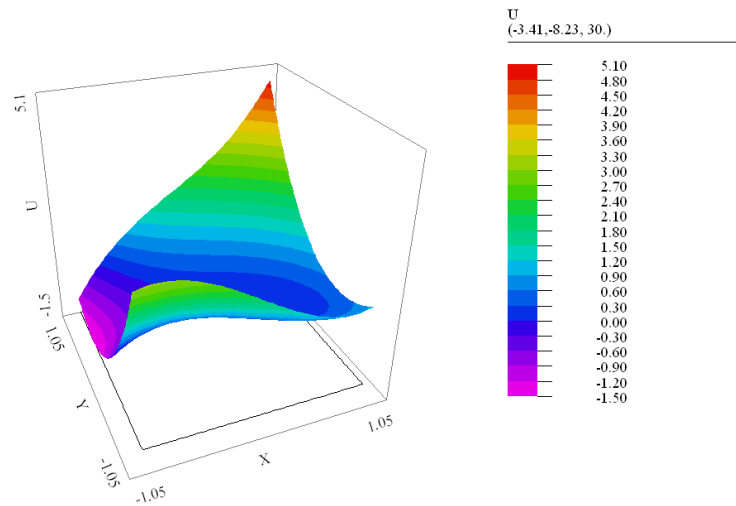
ドメイン上での関数 $U(x, y)$ の等高線図は次のようになります。



(3) Surface(U)

関数 $U(x, y)$ の曲面の形状をプロットしたものを示しておきます。

Poisson Equation

09:41:27 6/13/08
FlexPDE 5.0.22

export5: Grid#1 p2 Nodes=1133 Cells=536 RMS Er= 1.3e-5
Integral= 2.668173

(4) Contour(U_ex)

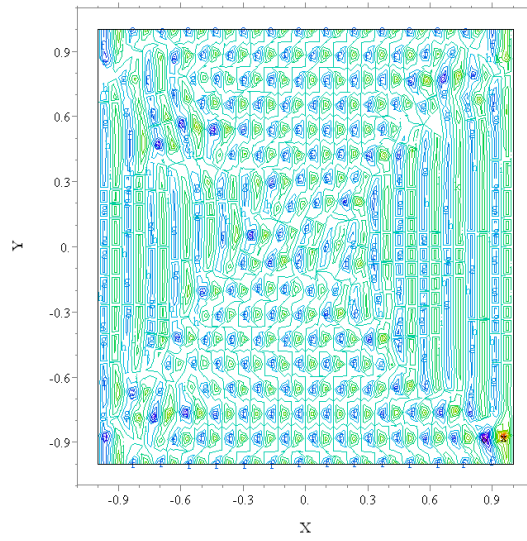
厳密解の等高線図ですが、見た目は Contour(U) と全く変わらないため省略します。

(5) Contour(U - U_ex)

次ページの図は FlexPDE の計算結果と理論曲面との差をプロットしたものです。スケールは $E-4(= 10^{-4})$ ということ、SELECT セクションで設定した誤差許容値の範囲におさまっていることがわかります。

Poisson Equation

09:41:27 6/13/08
FlexPDE 5.0.22



U - U_ex

max	3.09
s :	3.00
r :	2.70
q :	2.40
p :	2.10
o :	1.80
n :	1.50
m :	1.20
l :	0.90
k :	0.60
j :	0.30
i :	0.00
h :	-0.30
g :	-0.60
f :	-0.90
e :	-1.20
d :	-1.50
c :	-1.80
b :	-2.10
a :	-2.40
min	-2.40

Scale = E-4

export5: Grid#1 p2 Nodes=1133 Cells=536 RMS Er= 1.3e-5
Integral= 7.375722e-7

