FlexPDE　version 5

ユーザガイド

開発元：PDE Solutions Inc.
販売代理店：（株）ライトストーン

本書は *Scientific WorkPlace* を用いて作成されています。

目次

第 1 章　はじめに 1

第 2 章　概要 3

2.1 FlexPDE とは 3

2.2 FlexPDE の特長 4

2.3 FlexPDE の構成 5

2.4 対象ユーザ 6

2.5 スクリプトの概要 6

2.6 境界条件とは 8

第 3 章　基本的な用法 9

3.1 問題の設定方法 9

3.2 問題設定のガイドライン 10

3.3 記法 11

3.4 変数と方程式 11

3.5 ドメインのマッピング 12

3.6 例題 14

3.7 メッシュの生成 15

3.8 材質パラメータの設定 17

3.9 境界条件の設定 18

3.10 グラフィックス出力の指定 19

3.11 実行結果 20

第 4 章　主要制御項目 25

4.1 計算精度 25

4.2 積分計算 26

4.3 数値結果の出力 27

4.4 サマリプロット 28

4.5 ステージング 29
4.6 円柱座標系 31
4.7 時間依存性 34
4.8 固有値とモード解析 38

第 5 章 より高度な問題への適用 43
5.1 非線形係数と方程式 43
5.2 自然境界条件 46
5.3 不連続な変数 50

第 6 章 1 次元の問題への適用 57

第 7 章 3 次元の問題への適用 59
7.1 押出しの概念 59
7.2 押出しの設定 60
7.3 レイヤリング 62
7.4 材質特性の設定 63
7.5 空の区画 65
7.6 限定リージョン 66
7.7 切断面上のプロット 67
7.8 3D キャニスター用スクリプト 68
7.9 境界条件の設定（3 次元） 71
7.10 レイヤ境界の変形 75
7.11 3 次元における積分 78
7.12 より高度なプロット制御 82

第 8 章 メッシュ移動 85
8.1 メッシュバランシング 86
8.2 脈動する球体 87

第 9 章 メッシュ密度の制御 89

第 10 章 データのエクスポート 93

第 11 章 非線形問題への対応 97

第 12 章 技術サポート 101

索引 102

第 1 章

はじめに

本ドキュメントは偏微分方程式の問題を解くに当たって FlexPDE をどう使ったら良いかを順を追って紹介するものです。

最初に FlexPDE の基本的な特徴の議論から始めます。次に簡単なモデルの構築を行い、それに対し徐々に機能を追加して行きます。

このプロセスによって FlexPDE の最も良く使われる機能は一通りカバーされるので、その習得がより容易になると同時に、ユーザ固有の問題への適用がスムーズに行えるようになります。

なお、個々のコマンドに関する詳細な記述についてはコマンドリファレンスマニュアルをご参照ください。

第 2 章

概要

2.1 FlexPDE とは

FlexPDE はスクリプトベースの**有限要素法モデルビルダ**であると同時に数値演算に基づく PDE **ソルバ**でもあります。すなわちユーザによって記述されたスクリプトを入力として、FlexPDE はまず偏微分方程式の記述を有限要素法モデルに変換、次にそれを解き、その結果をグラフィックスや表形式のデータとして表示します。

FlexPDE は**問題解決用の統合環境**でもあります。それは偏微分方程式を解くために必要となる一連の機能、すなわち

- スクリプトの作成/編集を支援するエディタ
- 有限要素法のメッシュを構築するためのジェネレータ
- 有限要素法に基づくソルバ
- 結果を視覚化するためのグラフィックスシステム

をすべて含んでいます。スクリプトを作成、演算を実行し結果を確認、さらにスクリプトを編集し計算をやり直す — このような一連の操作を FlexPDE 環境の中ですべて行うことができます。

FlexPDE にはあらかじめ規定された問題領域があるわけではなく、また解法対象の方程式一覧が用意されているわけでもありません。どのような偏微分方程式を解くかはユーザが判断することです。

FlexPDE のスクリプト言語は**自然言語**と呼べるものです。通常の数学的記法を用いて偏微分方程式や問題領域の規定を行うことができます。

例えばスクリプト中の `EQUATIONS` セクションでラプラス方程式を記述する場合には次のように入力します。

```
Div(grad(u)) = 0
```

同様にスクリプトの BOUNDARIES セクションで 2 次元の矩形領域を規定する場合にはその周囲をなぞる形で次のように指定します。

```
Start(x1,y1) line to (x2,y1) to (x2,y2) to (x1,y2) to close
```

このようなスクリプト形式を用いることのメリットには次のようなものがあります。

- スクリプトによって方程式の系や問題領域が厳密に規定されます。問題固定型のシステムに良く見られるようなあいまいさは全くありません。
- 新たな変数や方程式、条件等を自由に追加することができます。従って特定の損失条件や物理的効果を表現できないといった問題は生じません。
- 多様な問題が FlexPDE という一つのプログラムによって解決できます。問題ごとに別個のプログラムについて学習し直すといった煩わしさはありません。

スクリプトモデルには次のような副次的効果もあります。

- ユーザは対象とする問題を数学的な形式で記述できなくてはなりません。

  教育環境においてこれは良いことと言えます。学生はまさにそれができるようにならなくてはならないからです。
  産業界においては特定の精通したユーザがスクリプトを用意し、それを一般ユーザが利用する、あるいは修正して利用するといった形態が考えられます。またアプリケーションスクリプトのライブラリは貴重な資産となるでしょう。

2.2　FlexPDE の特長

- FlexPDE は 1/2/3 次元直交座標系、あるいは軸対称 2 次元座標系における **1 階**、または **2 階**の偏微分方程式の系を解くことができます。（他の座標系については適切な形式の PDE を含めることによってサポートできます。）
- 求解対象としては**定常状態**を表す系、**時間依存型**の系、どちらも扱えます。また**固有値問題**を解くこともできます。なお、定常状態型の方程式と時間依存型の方程式が混在していても構いません。
- 連立させる方程式の数に制限はありません。ただし FlexPDE を動作させるコンピュータの性能によって制約が生ずることはあります。

- 方程式は**線形**であっても**非線形**であっても構いません。（非線形の問題に対しては**修正ニュートン-ラフソン反復法**（modified Newton-Raphson iteration）が自動的に適用されます。
- **素材特性**の異なる任意数の**リージョン**を定義することができます。
- モデル変数は素材の境界をまたいで連続であることが仮定されます。導関数の不連続性は偏微分方程式系に対する宣言文から導かれます。（`CONTACT` 境界条件は不連続な変数を扱うことができます。）
- FlexPDE の操作は非常に簡便であるため教育現場での使用に適しています。しかしそれはおもちゃではありません。機能を最大限活用することによって極めて難解な問題に対しても適用が可能になります。

2.3　FlexPDE の構成

FlexPDE は多数のモジュールが組み合わされた統合型の PDE ソルバです。

- 構文表示機能を備えた**スクリプトエディタ**は豊富なテキスト編集機能を提供すると同時に、グラフィカルなドメインプレビュー機能も提供します。
- 記号論理に基づく**数式アナライザ**は既定のパラメータや関係式を展開する機能を持つと共に空間微分を実行、さらに部分積分を用いて 2 階の項をなくし記号形式のガラーキン（Galerkin）方程式を生成します。次にこのモジュールはそれらの方程式をシンボリックな形で微分しヤコビ行列（Jacobian coupling matrix）を構成します。
- **メッシュ生成モジュール**は 2 次元、あるいは 3 次元の問題領域上に三角形、あるいは四面体の有限要素メッシュを構成します。2 次元の場合には任意のドメインが非構造の（unstructured）三角形メッシュによって埋め尽くされます。一方、3 次元の問題においては任意の 2 次元ドメインが第 3 の次元に伸張された上で分割曲面によって切り取られます。結果として生成される 3 次元の図形は非構造の四面体メッシュによって埋め尽くされます。
- **有限要素法数値演算モジュール**は問題が定常状態型か時間依存型か固有値問題型かによって、さらには線形か非線形かによって、適切な解法を選択します。有限要素法の基底関数としては 2 次式、あるいは 3 次式のいずれかが使用されます。
- **動的**（adaptive）**メッシュ調整機能**はメッシュが適切であるかどうかを監視し、誤差が大きくなった場合にはメッシュサイズの調整を行います。このプロセスは誤差がユーザ指定の許容値の範囲に収まるまで繰り返されます。
- **動的タイムステップ調整機能**は時間軸方向における解の曲率を計測し、積分計算の精度を維持できるようにタイムステップの調整を行います。

- **グラフィックス出力モジュール**は任意の代数関数を扱うことができ、等高線図、曲面図、ベクトルプロット、高度プロットの作成を行います。
- **データエクスポートモジュール**は単純な表データの他に、有限要素法メッシュデータや CDF, TecPlot 互換ファイル等、多様な形式でのレポート出力機能を提供します。

2.4　対象ユーザ

物理や工学の問題の大半は程度の差はあれ偏微分方程式によって記述されます。このことは実質どの分野の工学/科学上の問題に対しても FlexPDE（スクリプトベースの PDE ソルバ）が適用可能であることを意味しています。

- いろいろな分野の**研究者**が実験や機材のモデル化に FlexPDE を使用することができ、様々な効果の重要性予測や検証を行うことができます。パラメータ値を変化させたりその依存性を評価することにおいて何ら制約が加わることはありません。
- **エンジニア**は設計の最適化、フィージビリティスタディ、概念設計の過程に FlexPDE を適用することができます。設計のすべての側面を一つのソフトウェアでモデル化することができます。目的ごとに異なるツールについて学習する必要はありません。
- **アプリケーション開発者**は偏微分方程式に対する有限要素モデリングを必要とする特殊用途のアプリケーションにおいて、その中核として FlexPDE を使用することができます。
- 物理や工学の**教育現場**において FlexPDE を使用することができます。FlexPDE のみで問題のすべての側面について検証を行うことができます。
- 学生は実際の方程式を目にすることができるので、条件や領域に変更を加えながらその影響を対話的に調べるといった操作が可能になります。

2.5　スクリプトの概要

問題を記述するスクリプトはテキストファイルの形式で格納されるため、通常のテキストエディタを使えばその内容を確認することができます。スクリプトファイルは複数のセクションから構成されていますが、それらはヘッダによって識別されます。基本的なセクションには次のようなものがあります。

TITLE	出力に対するラベル。
SELECT	FlexPDE のデフォルト動作に関するコントロール。
VARIABLES	従属変数の規定。
DEFINITIONS	パラメータ、関係、関数の定義。
EQUATIONS	変数と偏微分方程式との対応付け。
BOUNDARIES	ドメインの幾何形状の定義。線分と円弧で接続可。
MONITORS, PLOTS	グラフィックス出力の規定。`CONTOUR`, `SURFACE`, `ELEVATION`, `VECTOR` の任意の組合せが可。
END	スクリプトの終端。

Note: 他にもいくつかオプショナルなセクションが用意されています。それらのいくつかについては後述します。セクションの記述様式詳細についてはコマンドリファレンスマニュアルをご参照ください。

コメントはスクリプト中の任意の位置に挿入できます。

{ }	中括弧で囲まれた文字列はコメントとみなされます。
!	感嘆符から始まる文字列はその行末までコメントとみなされます。

正方形領域上での簡単な拡散方程式は次のように記述できます。

```
TITLE 'Simple diffusion equation'
{ this problem lacks sources and boundary conditions }
VARIABLES
  u
DEFINITIONS
  k=3 { conductivity }
EQUATIONS
  div(k*grad(u)) = 0
BOUNDARIES
  REGION 1
    START(0,0)
    LINE TO (1,0)
         TO (1,1)
         TO (0,1)
         TO CLOSE
```

```
PLOTS
  CONTOUR(u)
  VECTOR(k*grad(u))
END
```

スクリプトの記述例については後続ページ中でも紹介して行きます。

2.6　境界条件とは

境界条件を適切に設定することは偏微分方程式を解く上で極めて重要なことです。

FlexPDE のスクリプトにおいては、境界定義の一環として境界条件が規定されます。

境界条件には VALUE と NATURAL という 2 つの基本的種別があります。

VALUE **型（Dirichlet）境界条件**はドメイン境界上で変数がどのような値を取らなくてはならないかを指定します。

これに対し NATURAL **型（自然）境界条件**はドメイン境界上での流束（flux）を規定します。（自然境界条件の厳密な意味は対象となる偏微分方程式の種類によって異なります。詳細はセクション 5.2“自然境界条件”を参照ください。）

一例として上記の拡散型問題において上端、下端に対しては固定値の、右端、左端に対しては流束 0 の境界条件を設定してみましょう。

```
...
BOUNDARIES
  REGION 1
    START(0,0)
    VALUE(u) = 0      LINE TO (1,0)    { fixed value on bottom }
    NATURAL(u) = 0    LINE TO (1,1)    { insulated right side }
    VALUE(u) = 1      LINE TO (0,1)    { fixed value on top }
    NATURAL(u) = 0    LINE TO CLOSE    { insulated left side }
...
```

VALUE と NATURAL という宣言文によって規定された条件は別の指定がない限り後続の境界セグメントにも適用されることになる点に注意してください。

第 3 章

基本的な用法

3.1 問題の設定方法

FlexPDE は問題の特質を記述したテキストファイル形式のスクリプトを読み込みます。簡単な問題であればスクリプトは非常に短いもので済みます。しかし複雑な問題になると FlexPDE の機能に精通した上でスクリプトを記述しなくてはならなくなります。

以下の議論においては FlexPDE のより簡潔な機能からスタートし、その後より複雑な機能を徐々に追加して行くことにします。

FlexPDE にはスクリプトエディタが組み込まれているので、それを用いてスクリプトの作成が行えます。作成されたスクリプトを編集し実行、さらに編集を加え実行という一連の操作を満足の行く結果が得られるまで繰り返すことになります。作成されたスクリプトは保存しておけるので後日再利用したり、あるいは別のスクリプト開発のベースとして使用することができます。

与えられた問題を定式化する上で最も簡便な方法は類似の問題に対する既存のスクリプトをコピーして利用する方法です。

既存のスクリプトをベースにするにせよ、あるいは全く新規に作成するにせよ、次の 5 つの項目については設定を行う必要があります。

- 変数と方程式の定義
- ドメインの定義
- 材質パラメータの定義
- 境界条件の定義
- グラフィックス出力項目の指定

これらのステップについては後続のセクションで説明します。具体例としては簡単な 2 次元の熱流体問題を用いることにし、FlexPDE の最も基本的な要素に対するスクリプトの設定から話を始め

ることにします。後半のセクションでは徐々にスクリプトを拡充し、FlexPDE のより高度な機能について見てゆくことにします。3 次元の問題については 2 次元での考え方が基盤となるため、章を改めて説明を行います。

Note: 後続のセクションにおいては選択可能なオプションのすべてについて言及するのは避け、最も良く使用される共通的な項目についてフォーカスすることにします。多くのオプションについては触れることになりますが、全般的な詳細仕様についてはコマンドリファレンスマニュアルを参照してください。

3.2 問題設定のガイドライン

FlexPDE で問題を定式化する際に念頭に置いておくべきガイドラインがいくつかあります。

- **物理系に関する基本的なステートメントから開始する。**
  基本的な保存則の記述は通常、教科書等に良く見られるような擬解析的な（pseudo-analytic）簡略化よりも有効に機能します。

- **なるべく解のわかっている単純なモデルから始める。**
  これによって問題の定式化が正しく行われたかどうかをチェックできるばかりでなく、FlexPDE に対する信頼感を醸成することにもつながります。（解析解がわかっている方程式を解かせるにしても、適切な境界条件を設定することを忘れてはなりません。）

- **材質パラメータについてはまず簡単な値を試す。**
  最初の段階では非線形項や材質の特性を厳密に規定しない方が良いでしょう。より単純なケースで動作を確認した後、複雑な要素については後から追加するようにしてください。

- **ドメイン全体の規定を先行させる。**
  ドメイン全体としての境界の記述と境界条件の設定を先に行ってください。次に材質の異なるリージョンを上に重ねて行きます。上に重ねられたリージョンによって下にあったものは置き換えられてしまうので、複雑な境界の定義を多数繰り返す必要はありません。

- `MONITORS` **機能を使って途中経過の情報を確認する。**
  最終結果の出力だけではおかしな結果が得られたとしてもその原因の解明は困難です。`MONITORS` セクションを活用してなるべく多くのフィードバックを得るようにしてください。

- **スクリプトには注釈を多く入れる。**
  スクリプトを後で見たとき、なぜそのようなコードを設定したのか思い出せなくなることが良くあります。参考文献に関する情報や導出過程に関する注釈なども役に立ちます。

- **保存操作を頻繁に行う。**
  FlexPDE は ”Domain Review” や ”Run Script” コマンドがクリックされた場合には常にスクリプトをディスク上に書き出します。しかし多くの入力を伴う場合などは不測の事態が起きても困らないよう、”Save”, あるいは ”Save as” コマンドを使用してスクリプトの保存を頻繁に行うようにしてください。

3.3 記法

FlexPDE の記法の多くはプログラミング言語の場合と同様、簡潔なテキストストリングです。

- $\frac{du}{dx}$ といった微分表記には `dx(u)` という記法が用いられます。アクティブな座標変数は自動的に認識されます。2 階微分は `dxx(u)` のように記述します。また `Div`, `Grad`, `Curl` といった微分演算子も使用できます。
- 名称に関しては大文字/小文字の区別はありません。例えば`"F"`と`"f"`は同一とみなされます。
- コメントは自由に挿入できます。{ } で囲まれた文字列、あるいは`!`に続く文字列はコメントとして解釈されます。

Note: FlexPDE の記法詳細についてはコマンドリファレンスマニュアルを参照してください。

3.4 変数と方程式

FlexPDE は次の 2 つの点について認識する必要があります。

- 分析対象の変数名は何か
- それらの変数を規定する偏微分方程式はどれか

スクリプト中の `VARIABLES` セクションと `EQUATIONS` セクションがこの情報を提供することになります。これら 2 つのセクションは密接に絡んでいます。それぞれの変数に対し通常は一つの方程式を対応させることになるからです。

簡単な問題の場合には電位や温度といった単一変数のみが求解の対象となることが良くあります。このような場合には変数と方程式を次のように指定するだけで済みます。

```
VARIABLES
  Phi
EQUATIONS
  Div(grad(Phi)) = 0
```

より複雑な問題においては複数の変数と複数の方程式の設定が必要になることがあります。この場合、変数と方程式の明確な対応付けが必要となります。具体的には個々の方程式の規定に際し、対応付けられる変数名をラベル（名称 + コロン）の形で指定します。

```
VARIABLES
  A,B
EQUATIONS
  A:  Div(grad(A)) = 0
  B:  Div(grad(B)) = 0
```

これらのラベルは境界条件を方程式と対応付ける際にも使用されます。

3.5 ドメインのマッピング

2 次元ドメインの設定

2 次元のドメイン（問題領域）は BOUNDARIES セクション中で規定されますが、それは独自の材質特性を有する一つ、または複数の REGION から構成されます。REGION には複数の閉ループや島が含まれていて構いませんが、それらは皆同一の材質特性を持っていることが仮定されます。

- REGION の指定は REGION <number>（または REGION "name"）というステートメントで始まり、それに続けてループの記述が行われます。
- スクリプトの後の方に出てくる REGION は前の方で設定された REGION の全体、もしくは一部を覆う形となります。
- 先頭の REGION はドメイン全体をカバーするようにしてください。常にそうあらねばならないというわけではありませんが、こうすることで境界条件の設定が容易になります。

リージョンの形状を規定は各頂点間を LINE, SPLINE, ARC セグメントのいずれかで結合することによって行います。先頭には START 文が置かれますが、それに続くセグメントは一つ前のセグメントの終端からスタートするものと仮定されます。セグメントを全体の始点に結んでループを閉じさせるには CLOSE という語を使用します。

- 矩形のリージョンを設定するには 4 つの LINE セグメントを使用します。

```
START(x1,y1)
LINE TO (x2,y1)
     TO (x2,y2)
     TO (x1,y2)
     TO CLOSE
```

  座標点の値を変えれば任意の四辺形の領域が設定できます。また点の数を増やすことによって多角形の領域が設定できます。

- 円弧を設定するにはいくつかの方法がありますが、中心の座標と角度を指定するのが最も簡便でしょう。

```
START(r,0)
ARC(CENTER=0,0) ANGLE=360
```

- 円弧は中心点と終点を指定する形でも設定できます。

```
START(r,0)
ARC(CENTER=0,0) TO (0,r)    { a 90 degree arc }
```

  中心点と終点間の距離が中心点と始点間の距離と異なる場合には楕円弧が設定されます。ただし楕円の長軸、短軸の方向は座標軸に平行であると仮定されます。傾いた楕円は設定できません。

- ループに対して名称を付けておけば別の場所から参照することができます。

```
START "Name" (...)
```

BOUNDARIES セクションの基本形は次のようになります。

```
BOUNDARIES
  REGION 1
    < ドメインを規定するループ >
  REGION 2
    < 第 2 の材質からなるリージョンを規定するループ >
  ...
```

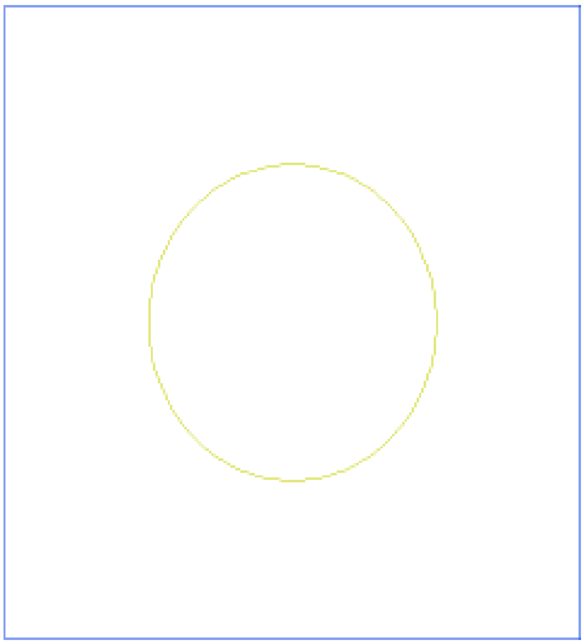
"domain" メニューボタンを使えばドメインの設定が途中であっても定義された部分について表示が行えるので、状態を確認しながらドメインの設定を進めることができます。

3.6 例題

ここでは 2 つの平板の間に円形の領域がはさまれているケースについて考えることにします。その際、平板については正方形領域の上端、下端に対応させ、その中間に円形領域を配置します。
BOUNDARIES セクションの設定は次のようになります。

```
BOUNDARIES
  REGION 1 'box'    { the bounding box }
    START(-1,-1)
    LINE TO(1,-1)
         TO (1,1)
         TO (-1,1)
         TO CLOSE
  REGION 2 'blob'    { the embedded circular 'blob' }
    START 'ring' (1/2,0)
    ARC(CENTER=0,0) ANGLE=360 TO CLOSE
```

"domain" ボタンを押すと次のような図が表示されます。

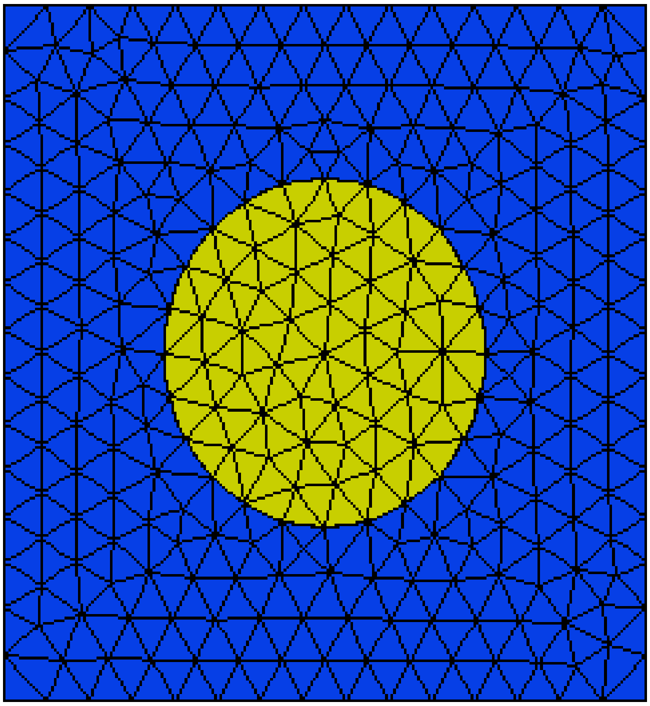


Note:　ドメイン境界の設定に関する詳細仕様についてはコマンドリファレンスマニュアルを参照してください。

3.7　メッシュの生成

Controls メニュー（または  ボタン）から ”Run Script” コマンドを選択すると FlexPDE は実行を開始し、まずドメインにフィットする有限要素メッシュを自動的に生成します。この自動生成メッシュの場合、セルの大きさはドメイン境界上の明示された点間の距離、円弧の曲率、及びユーザによる密度制御によって決定されます。

今回の例では自動生成されるメッシュのパターンは次のようになります。



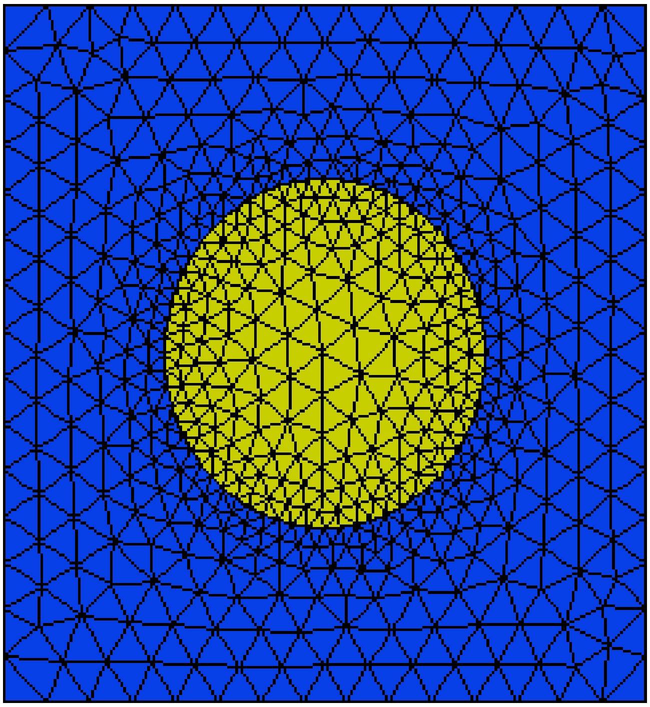
Region2 の円形の境界がセルの辺にマップされている点に注意してください。

メッシュの自動生成プロセスは何通りかの方法によって制御できます。詳細は第 9 章“メッシュ密度の制御”を参照ください。

一例として MESH_SPACING という修飾子（modifiers）を用いることによって、Region2 の円形境界上のメッシュ密度を増加させてみましょう。

```
REGION 2 'blob'    { the embedded 'blob' }
  START(1/2,0)
  MESH_SPACING = 0.05
  ARC(CENTER=0,0) ANGLE=360
```

結果は次のように変化します。



通常はこのメッシュ生成のプロセスに介入する必要はありません。後述するように FlexPDE には強い曲率を検知した場合にメッシュの密度を自動調節する機能が備わっているからです。

3.8　材質パラメータの設定

現実的な問題の取扱いが難しい理由の多くは、偏微分方程式中に含まれる係数が材質ごとに異なる値を持つという事実に起因します。

FlexPDE ではこの点に対応するために 2 つの機能が用意されています。第 1 のスキームとして、材質パラメータには `DEFINITIONS` セクションで名称とデフォルト値が与えられます。第 2 のスキームは材質パラメータに対して `REGION` ごとに固有の値が設定できるという点です。

これまでの議論においては対象としている問題が熱流体の問題なのか静電気の問題なのか、はたまた全く別の問題なのかによってほとんど違いは生じませんでした。しかしこれからの議論を具体的なものとするために、ここでは熱流の問題にフォーカスすることにします。すなわち熱源の間に熱伝導体（conductor）が置かれ、かつその中に非伝導体（insulator）が埋め込まれているモデルを想定します。円形の insulator に対しては熱伝導率 0.001 を、一方、周囲の conductor に対しては熱伝導率 1 を仮定することにしましょう。

まず `DEFINITIONS` セクションにおいてパラメータ名`"k"`とデフォルト値 1 を設定します。

```
DEFINITIONS
  k = 1
```

それぞれのリージョン中でその値が再設定されない限り、このデフォルト値が`"k"`の値として使用されます。

このパラメータを偏微分方程式の中に導入します。

```
EQUATIONS
  Div(-k*grad(phi)) = 0
```

次に Region2（insulator）中でのパラメータ値を設定します。

```
  ...
  REGION 2 'blob'    { the embedded blob }
    k = 0.001
    START(1/2,0)
    ARC(CENTER=0,0) ANGLE=360
```

Region1（conductor）において `k = 1` という設定を明示的に入れておいても構いません。あいまいさを減らす上では有効です。

3.9 境界条件の設定

境界条件はドメイン境界の定義の過程で修飾子として設定します。境界条件には VALUE と NATURAL という 2 つのタイプがあります。

VALUE 型（Dirichlet）境界条件はドメインの境界上で変数が取るべき値を規定します。値としては任意の数式表現が使用できます。他の変数に依存した非線形な数式であっても構いません。

NATURAL 型（自然）境界条件はドメイン境界上での流束（flux）を規定します。定義式としては任意の数式表現が使用できます。他の変数に依存した非線形な数式であっても構いません。ラプラス方程式の場合、NATURAL 型境界条件は Neumann（法線微分）境界条件と等価なものとなります。

Note: NATURAL 型境界条件の厳密な定義は PDE によって異なります。詳細についてはセクション 5.2“自然境界条件”を参照してください。

それぞれの境界条件ステートメントは引数として変数名を取ります。この名称が境界条件を特定の偏微分方程式に対応付けます。例えば VALUE(u)=0 という修飾子が適用される方程式は変数 u を規定すると定義された PDE です。NATURAL(u)=0 という指定の場合の意味は u を規定する方程式のタイプに依存します。

我々の具体例においては、上端、下端に対して温度一定（上端では温度 1、下端では温度 0）という境界条件を、右端、左端に対しては熱流 0（熱的絶縁）という境界条件を設定してみましょう。スクリプト上の指定は次のようになります。

```
...
REGION 1 'box'    { the bounding box }
  START(-1,-1)
  { Phi=0 on base line:  }
  VALUE(Phi)=0 LINE TO(1,-1)
  { normal derivative =0 on right side:  }
  NATURAL(Phi)=0 LINE TO (1,1)
  { Phi = 1 on top:  }
  VALUE(Phi)=1 LINE TO (-1,1)
  { normal derivative =0 on left side:  }
  NATURAL(Phi)=0 LINE TO CLOSE
```

VALUE や NATURAL のステートメントで設定された条件は、次に別の条件設定がされない限り、後続の境界セグメントにも適用されるので注意してください。また異なる境界条件を設定した場合には、LINE とか ARC といったセグメントの形状についても再度指定し直す必要があります。

Note: 境界条件の指定様式としては他の形式も許容されます。詳細についてはコマンドリファレンスマニュアルを参照してください。

3.10　グラフィックス出力の指定

グラフィックス出力の制御は MONITORS セクションと PLOTS セクションで行います。

MONITORS は計算の途中経過を表示するのに使用します。一方、PLOTS は最終的な出力タイプを規定するもので、この場合の出力結果はディスク上に保存されます。

FlexPDE は種々の出力タイプをサポートしていますが、主なものを列記すると次のようになります。

CONTOUR	等高線プロット。色塗りも可。
SURFACE	3 次元曲面のプロット。
VECTOR	ベクトル場のプロット。
ELEVATION	指定されたパス上での値の変化。
SUMMARY	テキストのみによるレポート。

プロットの数に制限はありません。またプロットする値としては変数（従属変数）、座標変数、パラメータを正しく組み合わせた任意の数式を指定できます。

ここでは

- 等温線
- 熱流束のベクトル場 k*grad(Phi) のプロット
- 中央線に沿った温度分布
- 円周上での法線方向熱流束

をプロットしてみましょう。

```
PLOTS
  CONTOUR(Phi)
  VECTOR(-k*grad(Phi))
  ELEVATION(Phi) FROM (0,-1) to (0,1)
  ELEVATION(Normal(-k*grad(Phi))) ON 'ring'
```

PLOTS セクションで指定された出力は、FlexPDE が最終的な解（メッシュサイズの動的調整を経てすべてのセルにおける誤差が許容範囲内となった解）に到達した時点で生成されます。これに対し、指定様式は PLOTS と変わりませんが MONITORS とラベリングされたセクションにおいて出力が要求された場合には、計算の途中経過がより高頻度な形で表示されます。

PLOTS の出力はスクリプトファイル（ファイル拡張子.PDE）と同名で拡張子が.PG5 というファイル中に記録されます。これらのプロットは FlexPDE の VIEW メニューを使って表示させることができます。

MONITORS の場合にはファイル出力は行われません。しかし MONITORS の機能はスクリプトの誤りを正し、所定の解が得られるようにする上で大変役に立つので、積極的に活用してください。

Note: FlexPDE には他にも GRID とか HISTORY といったプロットコマンドが用意されています。詳細はコマンドリファレンスマニュアルを参照してください。

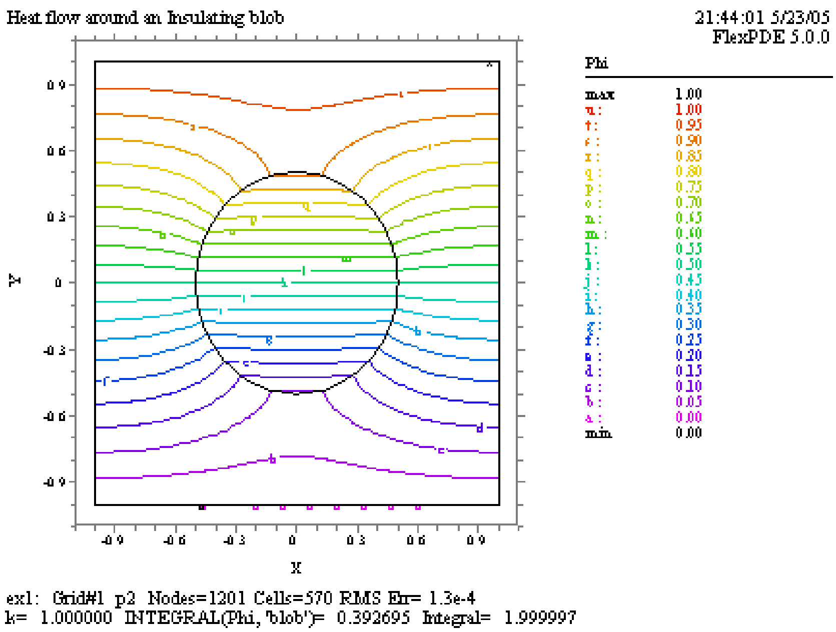
3.11 実行結果

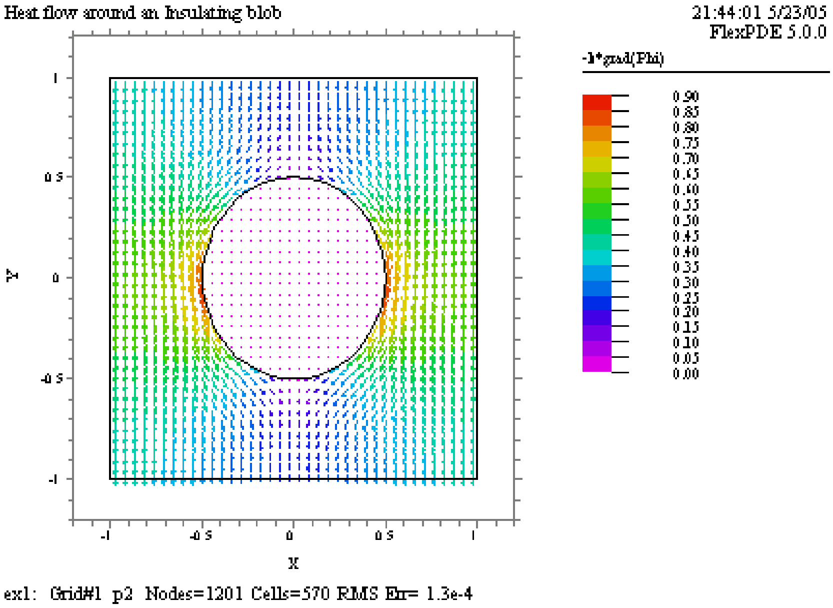
ここまでスクリプトを段階的に準備してきましたが、これらをまとめると次のようになります。

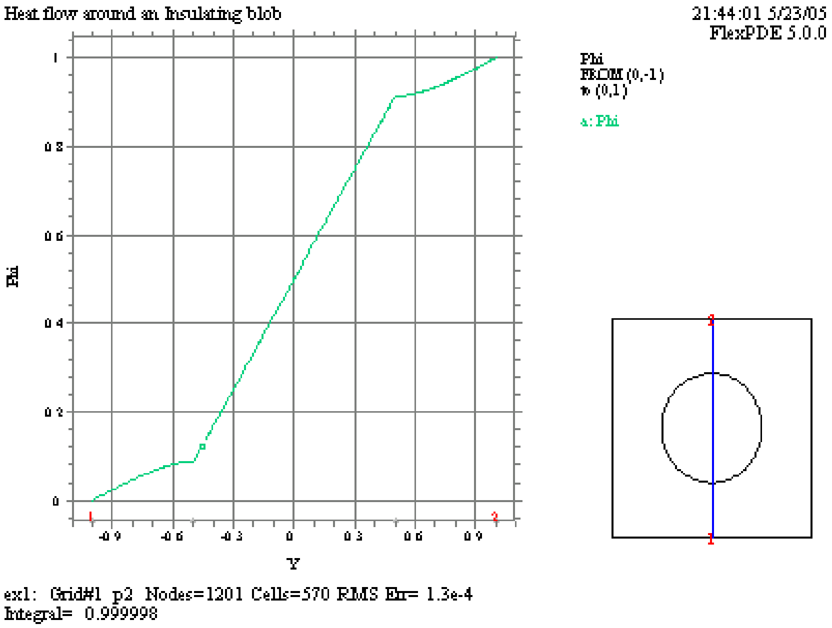
```
TITLE 'Heat flow around an Insulating blob'
VARIABLES
  Phi       { the temperature }
DEFINITIONS
  K = 1     { default conductivity }
  R = 0.5   { blob radius }
EQUATIONS
  Div(-k*grad(phi)) = 0
```

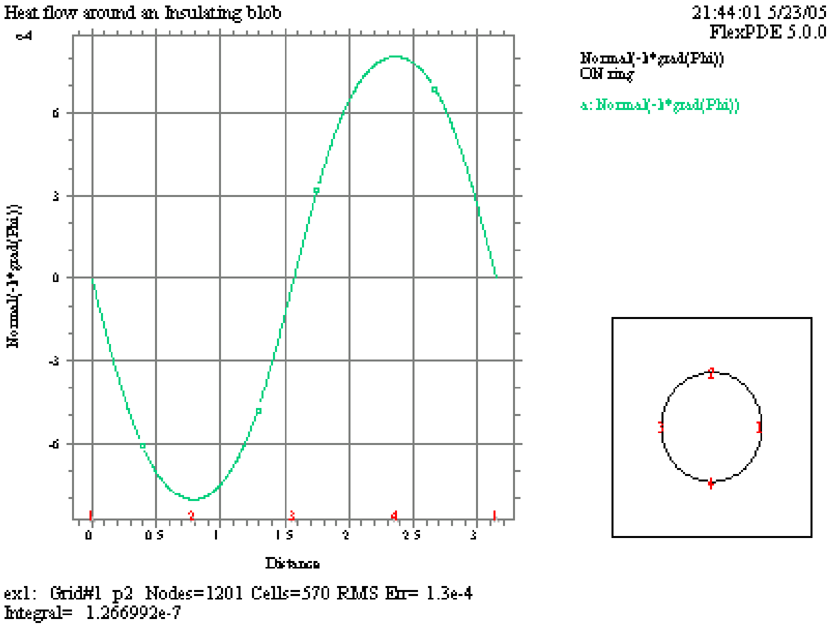
```
BOUNDARIES
  REGION 1     'box'
    START(-1,-1)
    VALUE(Phi)=0       LINE TO (1,-1)
    NATURAL(Phi)=0     LINE TO (1,1)
    VALUE(Phi)=1       LINE TO (-1,1)
    NATURAL(Phi)=0     LINE TO CLOSE
  REGION 2     'blob'     { the embedded blob }
    k = 0.001
    START 'ring' (R,0)
    ARC(CENTER=0,0) ANGLE=360 TO CLOSE
PLOTS
  CONTOUR(Phi)
  VECTOR(-k*grad(Phi))
  ELEVATION(Phi) FROM (0,-1) to (0,1)
  ELEVATION(Normal(-k*grad(Phi))) ON 'ring'
END
```

この 25 行のスクリプトが問題を完全な形で記述しています。これを FlexPDE に実行させると次のようなグラフィックス出力を得ることができます。




Heat flow around an Insulating blob
21:44:01 5/23/05
FlexPDE 5.0.0
-1*grad(Phi)
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
Y
X
ex1: Grid#1 p2 Nodes=1201 Cells=570 RMS Err= 1.3e-4


Heat flow around an Insulating blob
21:44:01 5/23/05
FlexPDE 5.0.0
Phi
FROM (0,-1)
to (0,1)
a: Phi
Phi
Y
ex1: Grid#1 p2 Nodes=1201 Cells=570 RMS Err= 1.3e-4
Integral= 0.999998


Heat flow around an Insulating blob
21:44:01 5/23/05
FlexPDE 5.0.0
Normal(-1*grad(Phi))
ON ring
a: Normal(-1*grad(Phi))
Normal(-1*grad(Phi))
6
3
0
-3
-6
0
0.5
1
1.5
2
2.5
3
Distance
ex1: Grid#1 p2 Nodes=1201 Cells=570 RMS Err= 1.3e-4
Integral= 1.266992e-7

第 4 章

主要制御項目

4.1 計算精度

FlexPDE はメッシュセル上で計算された積分値に対し一貫性チェックを行います。その結果から求解対象の変数値に対する相対誤差を推定し、それを**誤差許容値**と比較します。許容値を超えたセルがあった場合、そのセルは分割され、再度計算が実行されます。

この誤差許容値は `ERRLIM` と呼ばれ、スクリプトの `SELECT` セクションにおいてその値の指定が行えます。

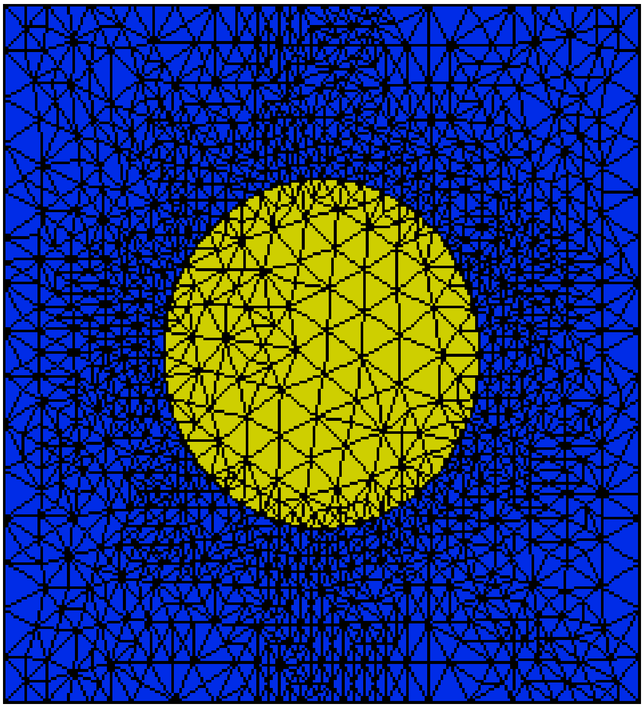
`ERRLIM` のデフォルト値は 0.001 です。従って FlexPDE はすべてのメッシュセルにおいて、変数値の推定誤差（変数の値域に対する相対値）が 0.1% より小さくなるまでメッシュの細分化を行うことになります。

Note: このことは得られた解の誤差がドメイン全体を通して 0.1% 以下であることを保証するものではありません。個々のセル上での誤差は予測し得ないような形で相殺したり累積したりすることがあります。

我々の例題においてステートメント

```
SELECT ERRLIM=1e-5
```

を追加してみましょう。これは FlexPDE に対し、推定誤差が 0.001% を超えるセルについては分割するように指示します。その結果、FlexPDE はメッシュ分割を 2 回行い、次のようなメッシュパターンが生成されることになります。



ただしこの事例について言うなら、得られるグラフィックスはデフォルト設定の場合と比べて大差はありません。

Note: 時間依存型の問題の場合、空間誤差と時間誤差の双方が ERRLIM によって設定されることになりますが、XERRLIM と TERRLIM を使って個別に設定することも可能です。詳細はコマンドリファレンスマニュアルを参照してください。

4.2 積分計算

多くの場合、偏微分方程式の求解において関心の対象となるのはある関数を積分したものです。FlexPDE には高度な積分計算機能が備わっており、**体積積分**の他、境界面上での**面積分**や境界線上での**線積分**を実行することができます。2 次元の場合の形式は次のようになります。

- `Result = LINE_INTEGRAL(<expression>, <boundary name>)`

  指定された境界上で数式 `<expression>` の線積分を実行します。

  Note: BINTEGRAL は LINE_INTEGRAL の別名です。

- Result = VOL_INTEGRAL(<expression>, <region name>)

  指定された領域上で数式 <expression> の面積分を実行します。

  Note: <region name> の指定が省略された場合にはドメイン全体が積分の対象となります。

  Note: INTEGRAL は VOL_INTEGRAL の別名です。

  Note: 2D 直交座標系の場合、AREA_INTEGRAL と VOL_INTEGRAL とは等価です。（ドメインは Z 方向に厚み 1 を持つと仮定されるため。）

次は'ring' 上での全流束を線積分で、'blob' 上の全熱エネルギーを面積分で計算する際の指定方法を示しています。

```
DEFINITIONS
{ the total flux across 'ring':
  (recall that 'ring' is the name of the boundary of 'blob')}
Tflux = LINE_INTEGRAL(NORMAL(-k*grad(Phi)), 'ring')
{ the total heat energy in 'blob': }
Tenergy = VOL_INTEGRAL(Phi, 'blob')
```

内部境界の場合、境界の両側で積分値が異なることがあります。積分を実行する領域を明示することでどちらの側で計算を行うかが区別されます。次の指定の場合には'ring' の外側で線積分が計算されることになります。

```
{ the total flux across 'ring': }
Tflux = LINE_INTEGRAL(NORMAL(-k*grad(Phi)), 'ring', 'box')
{ evaluated on the 'box' side of the boundary }
```

Note: 3 次元の場合の積分計算については後述します。詳細仕様についてはコマンドリファレンスマニュアルを参照してください。

4.3 数値結果の出力

FlexPDE からのグラフィック出力を評価したり分類したりする上で数値データが必要となることがしばしばあります。どのプロットコマンドに対しても REPORT ステートメントを付加することができます。

```
REPORT <value> AS "title"
```

単に

```
REPORT <value>
```

という形でも構いません。

指定できる REPORT の数に特に制限はありませんが、これらの数値データはプロットの下部に一つの行として出力されるため、数が多くなると紙面の幅をはみだしてしまう可能性があります。

一例として CONTOUR コマンドに対して REPORT ステートメントを追加してみましょう。

```
CONTOUR(Phi) REPORT(k) REPORT(INTEGRAL(Phi, 'blob'))
```

計算を実行するとプロットの下部に次のような出力を得ることができます。

```
ex1:  Grid#1  p2  Nodes=1121 Cells=530 RMS Err= 5.e-5
k=  1.000000  INTEGRAL(Phi, 'blob')=  0.392695  Integral=  1.999999
```

4.4 サマリプロット

数値情報の出力用に SUMMARY という特別なプロットコマンドが用意されています。このコマンドからはグラフィックスは出力されず、多数の REPORT を配置するための専用のページが生成されます。

引数としてテキストストリングが与えられた場合には、それはそのままヘッダとして出力されます。

例えば次のような指定を行ってみましょう。

```
SUMMARY
  REPORT(k)
  REPORT(INTEGRAL(Phi,'blob')) as "Heat energy in blob"
  REPORT('no more to say')
```

この場合、次のようなサマリページを得ることができます。

```
SUMMARY

k=  1.000000
Heat energy in blob=  0.392695
no more to say
```

4.5 ステージング

FlexPDE はパラメータ値の分析を支援するための**ステージング**と呼ばれる機能をサポートしています。この機能を使うと一つ、あるいは 1 群のパラメータに関し、ある範囲内の種々の値に対し問題を繰り返し実行させることができます。

このようなステージングされた形で FlexPDE を実行するにはパラメータ値を STAGED という形で宣言します。

```
DEFINITIONS
  Name = STAGED(<value1>,<value2>,...)
```

この場合、STAGED パラメータリスト中の値を順に一つずつ取り出し、それをパラメータ"name"にセットする形で繰り返し実行が行われます。

STAGED パラメータがドメインの形状に影響を与えない場合には、それぞれのランでは直前のランにおける実行結果とメッシュを初期条件として利用する形で実行が行われます。

Note: この手法は非線形性の強い問題にアプローチする際にも利用できます。すなわち、まず線形系からスタートし、徐々に非線形要素の重みを増加させて行けば良いわけです。

STAGED パラメータがドメインの形状定義の中で使用されている場合には、それぞれのランはドメイン定義から始まるため、それ以前のランからの解の継承は行われません。

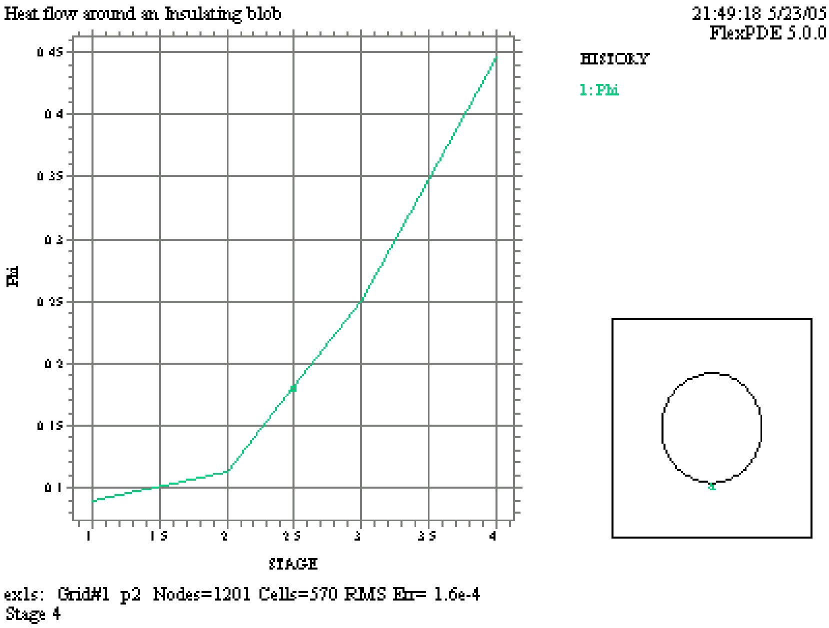
後述する時間依存型の問題の場合、HISTORY プロットの機能を使うと任意の量がステージ間でどう変化したかを表示させることができます。ステージングランの場合には HISTORY プロットにステージ番号が付加されます。

先の例で insulator の熱伝導率を変化させたとき、insulator 下端の温度がどう変化するかをヒストリとしてプロットさせてみましょう。なお、STAGED の設定はある名称が最初に定義されたときでないと行えない点に注意してください。リージョン内でパラメータ値が再設定された段階では STAGED の指定は行えません。

```
DEFINITIONS
  Kins = STAGED(0.01, 0.1, 1, 10)
  { Notice that the STAGED specification must appear at the initial
  declaration of a name. It cannot be used in a regional redefinition. }
...
```

```
   REGION 2 'blob'     { the embedded blob }
     K = Kins
     START(R,0) ARC(CENTER=0,0) ANGLE=360
 ...
 HISTORY(Phi) AT (0,-R)
```

スクリプトに対しこの変更を加えて FlexPDE を実行すると次のような HISTORY プロットが出力されます。



ステージングランの場合、PLOTS、及び MONITORS で指定された項目はそれぞれのステージごとに出力されます。

その他の制御項目

- 実行すべきステージの数をグローバルセレクタ STAGES によって制御することもできます。このセレクタの指定は DEFINITIONS セクションにおける STAGED リストの設定に優先します（リスト長が STAGES の値よりも短かった場合にはエラーが報告されます）。これは同時に STAGE というグローバル名称を定義することになるので、数式中で使用することもできます。詳細はコマンドリファレンスマニュアルを参照してください。

- デフォルトの設定では一つのステージが終わるとすぐ次のステージが開始されます。しかしプロットをチェックするために次のステージの実行を一旦中断させることもできます。
  `SELECT` セクション中に `AUTOSTAGE=OFF` というコマンドを設定してください。

4.6 円柱座標系

2 次元直交座標系に加え、FlexPDE は**軸対称**な場合に**円柱座標系** – (r, z) または (z, r) – を用いた問題も扱うことができます。

円柱座標系を使用は `COORDINATES` セクションで宣言します。`XCYLINDER` と `YCYLINDER` という 2 種類の様式が選択できます。両者は表示に際しての向きが異なるだけです。

XCYLINDER	円柱の回転軸である `Z` 座標をプロットの横座標（`"x"`軸）方向に、動径を縦軸（`"y"`軸）方向に配置するレイアウト。
YCYLINDER	円柱の回転軸である `Z` 座標をプロットの縦座標（`"y"`軸）方向に、動径を横軸（`"x"`軸）方向に配置するレイアウト。

いずれの場合でも座標軸の名前を（横軸、縦軸）の形で指定することができます。デフォルトの設定は次の通りです。

`XCYLINDER`	`XCYLINDER('Z','R')`
`YCYLINDER`	`YCYLINDER('R','Z')`

4.6.1 円柱座標系における積分

`VOL_INTEGRAL`（別名 `INTEGRAL`）は積分に際し $2\pi r$ という重みをかけます。これは求解対象が軸対称な回転体であることを反映したものです。

横断面上の積分は `AREA_INTEGRAL` によって実行できます。この場合、$2\pi r$ という重みはかかりません。

境界上の面積分は `SURF_INTEGRAL` によって計算できますが、この場合には面素に対する重み $2\pi r$ が適用されます。

4.6.2　円柱座標を用いた例題

先に用いた 2 次元直交座標系での問題を円柱座標系での問題に変換します。正方形領域（'box'）と円形領域（'blob'）を左端の境界を軸にして回転させると 2 つの円盤にはさまれたトーラスが形成されます。ただし次のような変更を加える必要があります。

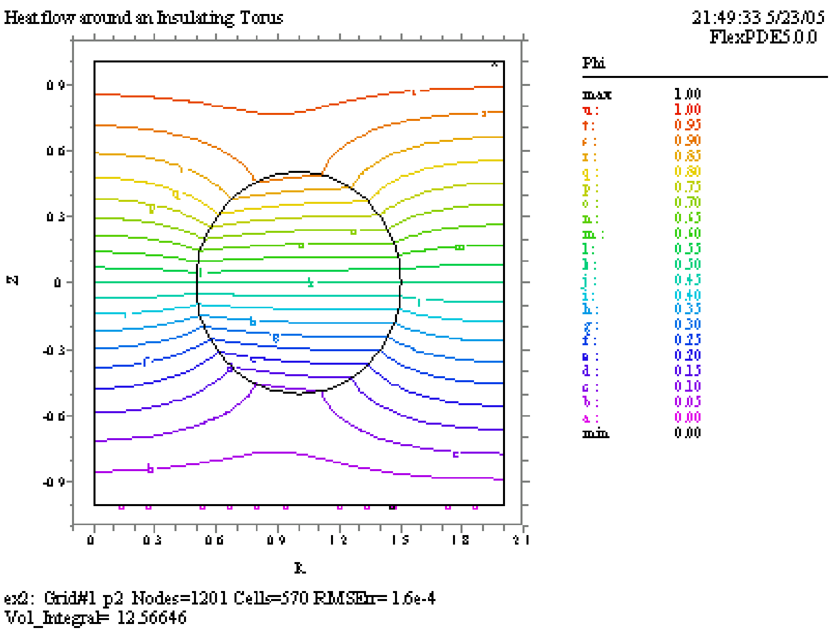
- 座標系をずらし、左端の境界が R=0 となるようにする。
- 回転軸を縦軸方向に取っているため YCYLINDER 座標系を使用する。
- 座標軸の名称として'R' が使用されることになるので、blob の半径としては別の名前を用いる。

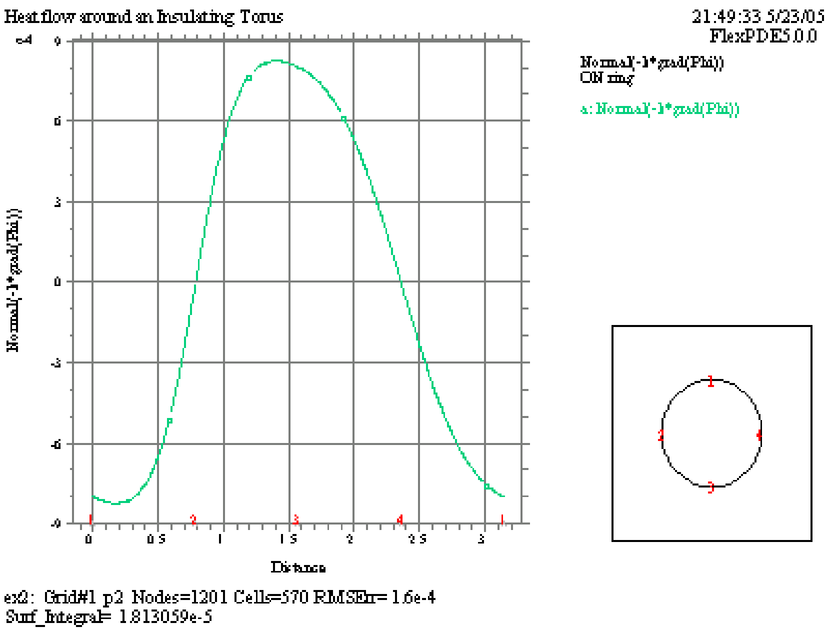
このトーラスに対する問題を記述するスクリプトは次のようになります。

```
TITLE 'Heat flow around an Insulating Torus'
COORDINATES
  YCYLINDER
VARIABLES
  Phi          { the temperature }
DEFINITIONS
  K = 1        { default conductivity }
  Rad = 0.5    { blob radius (renamed)}
EQUATIONS
  Div(-k*grad(phi)) = 0
BOUNDARIES
  REGION 1    'box'
    START(0,-1)
    VALUE(Phi)=0      LINE TO (2,-1)
    NATURAL(Phi)=0    LINE TO (2,1)
    VALUE(Phi)=1      LINE TO (0,1)
    NATURAL(Phi)=0    LINE TO CLOSE
  REGION 2    'blob'    { the embedded blob }
    k = 0.001
    START 'ring' (1,Rad)
    ARC(CENTER=1,0) ANGLE=360 TO CLOSE
```

```
PLOTS
  CONTOUR(Phi)
  VECTOR(-k*grad(Phi))
  ELEVATION(Phi) FROM (1,-1) to (1,1)
  ELEVATION(Normal(-k*grad(Phi))) ON 'ring'
END
```

ここでは出力結果のうち、等温線図と境界上での法線方向熱流束の図を示しておきます。





4.7　時間依存性

別に定めない限り FlexPDE は名称"T"（または"t"）を時間を表す変数と認識します。その時間への参照が定義項や方程式中に現れた場合、FlexPDE は初期値問題に適した解法を起動します。

FlexPDE は系の進展を追跡するために使用するタイムステップの制御にヒューリスティックスを用います。最初その値は変数の時間微分値に基づいて決められますが、その後変数の振舞いが 2 次関数的になるように調整されます。具体的に言うと、変数を時間についてテイラー級数に展開したとき、その 3 次の項がグローバルセレクタ ERRLIM の値よりも小さくなるようタイムステップを短くしたり伸ばしたりします。

時間依存型の問題の場合にはいくつか新たな項目を指定する必要があります。

- 各変数の適正値を表す THRESHOLD（初期値から明らかでない場合）
- 時間依存型の偏微分方程式
- 時間帯
- プロットを作成する時間値
- HISTORY プロット

Note: FlexPDE は時間に関する 1 階微分のみを扱うことができます。2 階微分が含まれる場合には中間変数を用いることにより、方程式を 2 つに分割してください。

時間帯については新たなセクション中で指定します。

```
TIME <start> TO <finish>
```

プロットを作成する時間値は該当するプロットコマンドに先行する形で指定します。指定様式としては特定の時間値をリストにして指定するか、あるいは時間間隔と終了時間を組み合わせて指定します。

```
FOR T = <t1> <t2> BY <step> TO <t3> ...
```

ここでは時間依存型の問題例を示すために、これまで扱ってきた熱流体の方程式に時間依存項（右辺）を追加することにします。

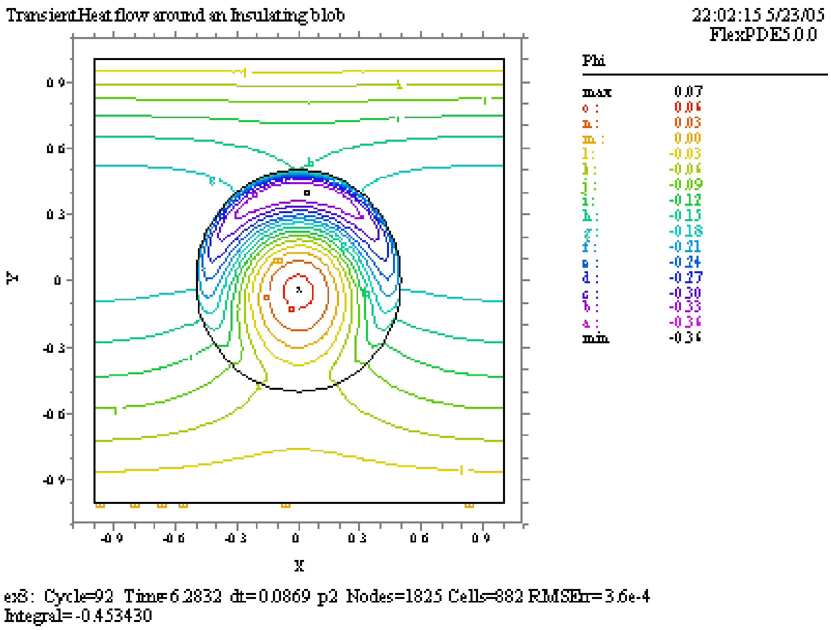
```
Div(k*grad(Phi)) = c*dt(Phi)
```

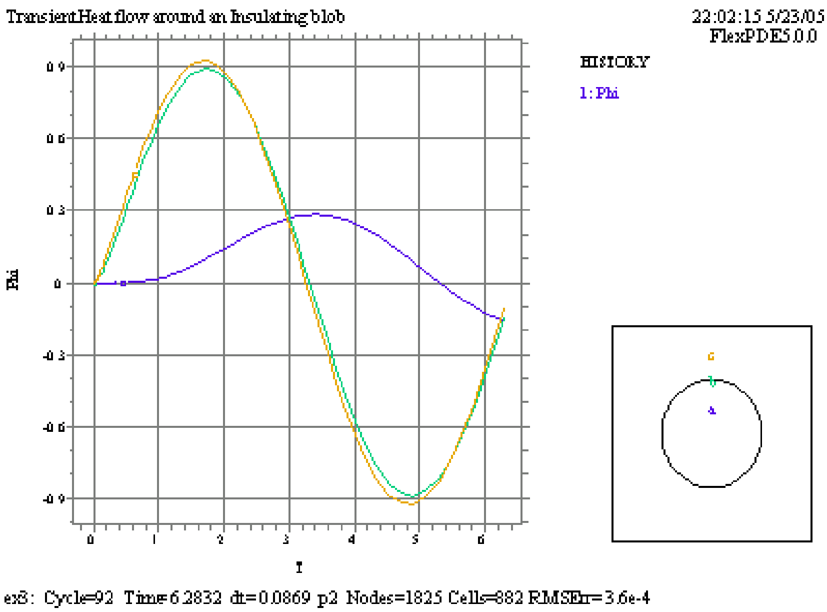
問題をより興味深いものとするために、上端のプレートの温度を $\sin(t)$ という形で変動させることにします。また内部の 3 点における温度変化を HISTORY プロットの形で出力させるコードも追加します。

```
TITLE 'Transient Heat flow around an Insulating blob'
VARIABLES
  Phi (threshold=0.01)    { the temperature }
DEFINITIONS
  K = 1             { default conductivity }
  C = 1             { default heat capacity }
  R = 1/2
EQUATIONS
  Div(-K*grad(phi)) + C*dt(Phi) = 0
BOUNDARIES
  REGION 1 'box'
    START(-1,-1)
    VALUE(Phi)=0         LINE TO (1,-1)
    NATURAL(Phi)=0       LINE TO (1,1)
    VALUE(Phi)=sin(t)    LINE TO (-1,1)
    NATURAL(Phi)=0       LINE TO CLOSE
  REGION 2    'blob'    { the embedded blob }
    K = 0.001
    C = 0.1
    START(R,0)
    ARC(CENTER=0,0) ANGLE=360
```

```
TIME 0 TO 2*pi
PLOTS
  FOR T = pi/2 BY pi/2 TO 2*pi
    CONTOUR(Phi)
    VECTOR(-K*grad(Phi))
    ELEVATION(Phi) FROM (0,-1) to (0,1)
HISTORIES
  HISTORY(Phi) AT (0,r/2) (0,r) (0,3*r/2)
END
```

最終時点 $(t = 2 \cdot \pi)$ における等温線図、及び HISTORY プロットは次のようになります。





4.7.1　時間依存型問題における注意事項

時間依存型問題において初期条件に矛盾があったり、あるいは境界値を開始時点（または別の時点）で瞬時に設定したりすると、系内に強い過渡的なシグナルが誘発されることになります。結果としてタイムステップや場合によってはメッシュサイズが非常に小さな値となってしまうことがあります。

このような急激な変化を細部にわたって調べるという特別な理由がない限りは、定常解と整合性のある初期条件を設定するようにしてください。さらに熱源とか外部からの流束等の境界値を設定する場合には、意味のある時間間隔の中で値を増加（あるいは減少）させる形で設定してください。

熱源や電源等をオンにするという動作をスムーズなオペレーションであると考えるのは誤りです。瞬時に値を変化させると数学的には超高周波を伴うことになります。そのような急激な変化に対する挙動を調べることが目的であるなら致し方ありません。しかし現実的に考えるなら、熱源や電源をオンにするといった操作はある有限時間をかけて行われるべきです。確かにこの時間は電気的なスイッチであればミリ秒、半導体スイッチであればマイクロ秒のオーダです。しかし 1 秒とか 2 秒経過した後の状態についてのみ問題にするのであれば、スイッチオンのプロセスは多少ぼかしても問題ないはずです。

4.8 固有値とモード解析

FlexPDE を使って偏微分方程式の**固有値**や**固有関数**を求めることもできます。

上で示した時間依存型の同次熱流体方程式について考えてみましょう。

$$C\frac{\partial\phi}{\partial t} - \nabla\cdot K\nabla\phi = 0$$

境界条件についても同次型のもの、すなわち

$$\phi = 0$$
$$\frac{\partial\phi}{\partial n} + \alpha\phi = 0$$

を想定します。

今、次のような変数分離型の解を求めたいとします。

$$\phi(x,y,t) = \psi(x,y)\cdot\exp(-\beta t)$$

このとき偏微分方程式は次のようになります。

$$\nabla\cdot K\nabla\psi + \lambda\psi = 0$$
$$\lambda = -C\beta$$

この方程式が自明以外の解を持つときの λ と ψ をそれぞれ系の固有値、固有関数と呼びます。偏微分方程式に対する一般解は、非同次境界条件を満たす一つの特解とこれら固有関数を組み合わせたものとして表現できます。

FlexPDE に固有値問題を解かせるには定常状態用のベーシックなスクリプトに対し次のような変更を加える必要があります。

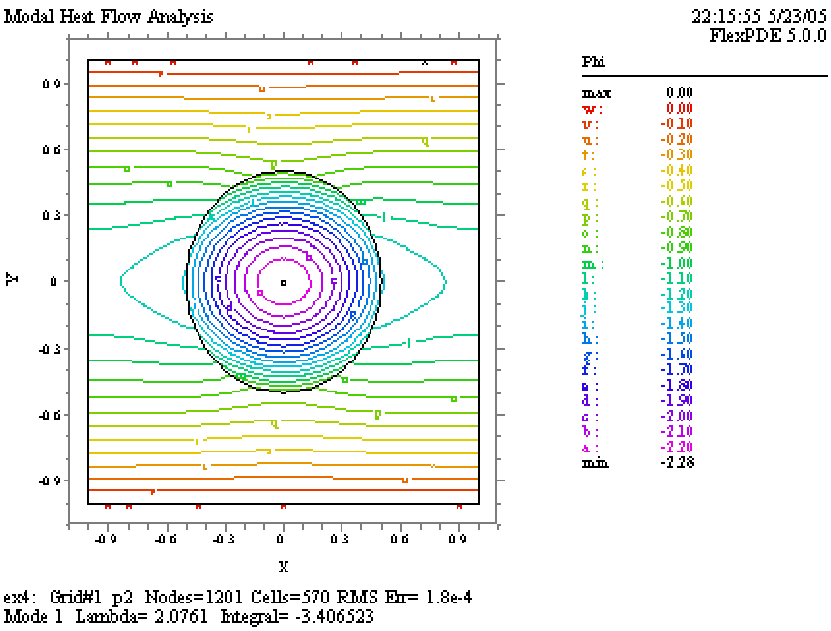
- `SELECT` セクションの `MODES` パラメータに対し値を設定します。この数によっていくつの λ の値を計算するかが決定されます。大きさの小さいものから順に使用されます。
- 方程式の記述に際しては固有値を表す予約語 `LAMBDA` を使用してください。その際、`LAMBDA` はすべて正の値となるよう方程式を調整する必要があります。さもないと求解に際しての順番に問題が生じることになります。

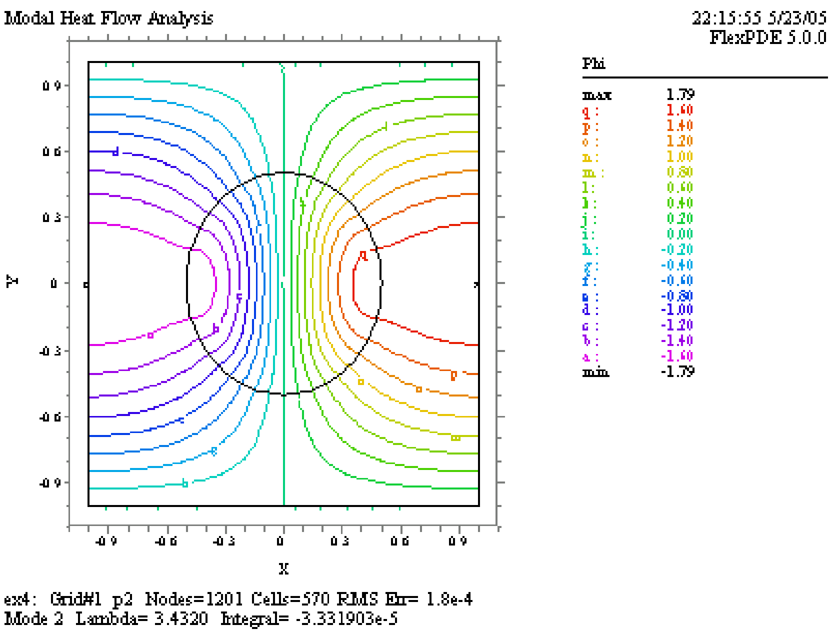
```
TITLE 'Modal Heat Flow Analysis'
SELECT
  modes=4
VARIABLES
  Phi             { the temperature }
DEFINITIONS
  K = 1           { default conductivity }
  R = 0.5         { blob radius }
EQUATIONS
  Div(k*grad(Phi)) + LAMBDA*Phi = 0
BOUNDARIES
  REGION 1 'box'
    START(-1,-1)
    VALUE(Phi)=0      LINE TO (1,-1)
    NATURAL(Phi)=0    LINE TO (1,1)
    VALUE(Phi)=0      LINE TO (-1,1)
    NATURAL(Phi)=0    LINE TO CLOSE
  REGION 2  'blob'    { the embedded blob }
    k = 0.2           { This value makes more interesting pictures }
    START 'ring' (R,0)
    ARC(CENTER=0,0) ANGLE=360 TO CLOSE
PLOTS
  CONTOUR(Phi)
  VECTOR(-k*grad(Phi))
  ELEVATION(Phi) FROM (0,-1) to (0,1)
  ELEVATION(Normal(-k*grad(Phi))) ON 'ring'
END
```

FlexPDE によって出力される解は次のような特徴を持ったものとなります。

- すべてのプロットは要求されたモードごとに出力される。
- 固有値の一覧を示すプロットページが追加される。
- 個々のプロットにはモード番号と固有値に関する情報が付加される。
- LAMBDA という名称は数式中でも使用できる。

最初の 2 つの固有値に対応した等温線図は次のようになります。


Modal Heat Flow Analysis
22:15:55 5/23/05
FlexPDE 5.0.0
Phi
max 0.00
w: 0.00
v: -0.10
u: -0.20
t: -0.30
s: -0.40
r: -0.50
q: -0.60
p: -0.70
o: -0.80
n: -0.90
m: -1.00
l: -1.10
k: -1.20
j: -1.30
i: -1.40
h: -1.50
g: -1.60
f: -1.70
e: -1.80
d: -1.90
c: -2.00
b: -2.10
a: -2.20
min -2.28
Y
X
0.9
0.6
0.3
0
-0.3
-0.6
-0.9
ex4: Grid#1 p2 Nodes=1201 Cells=570 RMS Err= 1.8e-4
Mode 1 Lambda= 2.0761 Integral= -3.406523


Modal Heat Flow Analysis
22:15:55 5/23/05
FlexPDE 5.0.0
Phi
max 1.79
q: 1.60
p: 1.40
o: 1.20
n: 1.00
m: 0.80
l: 0.60
k: 0.40
j: 0.20
i: 0.00
h: -0.20
g: -0.40
f: -0.60
e: -0.80
d: -1.00
c: -1.20
b: -1.40
a: -1.60
min -1.79
Y
X
0.9
0.6
0.3
0
-0.3
-0.6
-0.9
ex4: Grid#1 p2 Nodes=1201 Cells=570 RMS Err= 1.8e-4
Mode 2 Lambda= 3.4320 Integral= -3.331903e-5

4.8.1　固有値サマリ

固有値問題を実行すると、算出された固有値の一覧を示すプロットページが自動的に出力されます。

ユーザが SUMMARY プロットを指定した場合にはそちらが優先されることになるので、固有値リストに独自の情報を付加することができます。

例えば前のスクリプトに次の指定を加えてみましょう。

```
SUMMARY
  REPORT(lambda)
  REPORT(integral(phi))
```

出力される情報は次のようになります。

```
Modal Heat Flow Analysis                    22:15:55 5/23/05
                                              FlexPDE 5.0.0

  Eigenvalues:

  Mode  1: lambda= 2.076144  integral(phi)=-3.408079
  Mode  2: lambda= 3.431960  integral(phi)=-4.340801e-6
  Mode  3: lambda= 5.704378  integral(phi)=-1.050399
  Mode  4: lambda= 6.752271  integral(phi)= 9.194491e-4
```

第 5 章

より高度な問題への適用

正方形領域上での熱流が解析対象のすべてであるなら恐らく FlexPDE を持ち出すまでもないでしょう。FlexPDE の能力はほとんど任意の関数形が材質パラメータや数式中、あるいは出力指定項の中で使用できる点にあります。また幾何形状は非常に複雑なものであっても構いません。出力指定様式は簡潔であると同時に強力です。

後続のセクションでは現実的な問題を解く過程で遭遇する種々の状況について考察し、それらに FlexPDE をどう適用するかについて見て行くことにします。

5.1 非線形係数と方程式

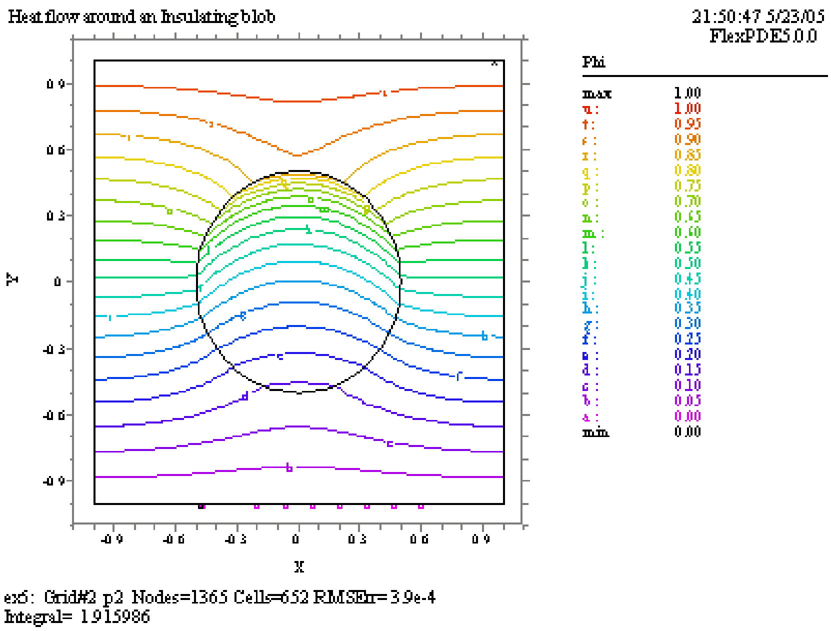
方程式や材質特性がシステム変数の複雑な関数となることが良く起こります。FlexPDE はそれに対し万全の対応を図っています。

一例として、これまで見てきた例における'blob' 中の熱伝導率が実際には温度に強く依存した関数である場合について考えてみましょう。ここでは $K = \exp(-5\phi)$ という関数を想定することにします。解は簡単なものになるはずがありません。しかし素直に入力して実行させてみましょう。

```
...
REGION 2    'blob'    { the embedded blob }
k = exp(-5*phi)
...
```

非線形性が加わったことは自動的に検知され専用のソルバが起動されます。非線形性に伴う種々の問題については FlexPDE が自動的に対処します。

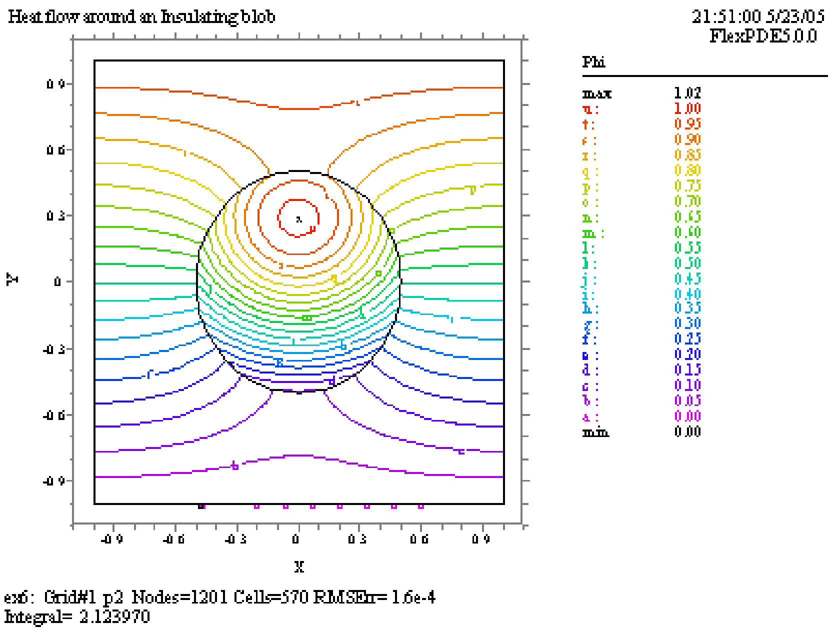
結果はすぐに得られ次のようになります。



方程式中に非線形項を加えることも容易に行えます。例えば熱源に $\phi\sin(\phi)$ という非線形項を設定してみましょう。

```
EQUATIONS
  Div(k*grad(phi)) + 0.01*phi*sin(phi) = 0
```

"run" をクリックするだけで次のような解を得ることができます。



5.1.1　**非線形性に伴う問題点**

実際には非線形問題への対応は上で述べたほど簡単ではありません。

- 非線形問題の場合、解は一つとは限りません。
- 非線形問題には解が全く存在しないこともあります。

FlexPDE は非線形系の問題を解くのに**ニュートン-ラフソン反復法**（Newton-Raphson iteration process）を使用します。この手法は解の**初期推定値**に極めて敏感です。初期条件が実際の解からかけ離れたものであった場合には、たとえ別の点からは簡単に求められるものであっても、全く解が求められないといったことも起り得ます。

非線形問題を扱う場合には次の点に留意ください。

- `INITIAL VALUES` セクションでできるだけ良い**初期値**を与える。
- **境界条件**に矛盾がないことを確認する。
- **ステージング**の機能を使って線形系から非線形系に移行する。これによって線形系での解を初期条件として利用することができます。
- 問題を**時間依存型**として設定する（時間軸を人為的な relaxation 次元として利用する）。
- `SELECT CHANGELIM` を使って余計な探索を抑制し、解への収束性を改善する。

- `MONITORS` の機能を使って情報を表示し、問題点の特定を図る。

扱いの難しい非線形問題の扱いについては再度第 11 章で後述します。

5.2 自然境界条件

自然境界条件という用語は通常変分法の中で用いられます。しかし有限要素法も本質的には誤差汎関数（functional）を最小化するものであることから、この用語は有限要素法との兼ね合いにおいても使用されます。

しかしこの用語に対してはより直感的でわかりやすい意味付けも可能です。

ラプラスの方程式

$$\nabla \cdot \nabla u = 0$$

について考えてみましょう。**発散定理**によれば左辺の量を平面領域 A 上で面積分した値は流束の法線方向成分を境界線上で線積分した値に等しくなります。

$$\iint\limits_A \operatorname{div}\left(\operatorname{grad}(u)\right) dA = \oint\limits_S n \cdot \operatorname{grad}(u) dl$$

（これは 2 次元の式ですが、3 次元でも同様のことが言えます。）

境界上での値 $n \cdot \operatorname{grad}(u)$ は実はラプラス（およびポアッソン）方程式の自然境界条件でもあります。これは境界内部の系が外部の系とどうインタラクトするかを示すものです。すなわち系の境界を横切る u の流れ（の符号を反転させたもの）の量を表しています。

発散定理はより一般的な**部分積分公式**

$$\int_a^b u dv = uv\Big|_a^b - \int_a^b v du$$

の姿を変えたものと見ることもできます。

項 uv を積分境界上で評価することによって境界上の積分式が出てきます。div の積分にこの公式を適用することによって発散定理の式が導かれます。

FlexPDE は偏微分方程式中でシステム変数の 2 階微分を含むすべての項にこの部分積分公式を適用します。もちろんラプラス方程式の場合にはすべての項がこれに該当するわけです。

ラプラス方程式の解を求めるためには、境界上のすべての点において変数 u の値を指定するか、$n \cdot \operatorname{grad}(u)$ の値を指定する必要があります。

FlexPDE の記法で言うと、

`VALUE(u)=u1`	が前者を
`NATURAL(u)=F`	が後者を規定することになります。

言い換えるなら

> FlexPDE における自然境界条件は境界面/境界線上での流束の値を指定します。その流束は偏微分方程式の部分積分によって規定されるものでもあります。

流束の値は非線形なものでも構いません。

例えば熱せられた物体からの放射損失は温度の 4 乗に比例します。従って次のステートメント

```
NATURAL(u) = -k*u^4
```

はラプラス方程式の境界条件として全く妥当なものと言えます。

5.2.1　代表的な例

部分積分は数学上の基本操作の一つであるため、その適用が数多くの基本的物理法則（例えばアンペールの法則）につながっていたとしても驚くには当たりません。

このため**自然境界条件**は多くの応用分野における基本的な保存則を記述することになる場合がしばしばあります。しかしラプラス方程式よりも複雑な方程式においてはその意味が常に自明であるとは限りません。

そこでまずいくつかの基本的な項とその自然境界条件による寄与を見て行くことにしましょう（ここでは 2 次元の場合について述べますが、3 次元への拡張も容易に行えます)。

- 項 $\dfrac{\partial f(u)}{\partial x}$ に部分積分を適用すると

$$\iint \frac{\partial f(u)}{\partial x} dxdy = \oint f(u)dy = \oint f(u)\alpha dl$$

  が導かれます。ここに α は境界線上の法線の x 方向余弦、dl は微小径路長を表します。FlexPDE は部分積分を 2 階微分の項にしか適用しないので、この公式は関数 $f(u)$ にさらに u の導関数が含まれている場合にのみ適用されます。x 以外の座標変数に対しても同様です。

- 項 $\dfrac{\partial^2 f(u)}{\partial x^2}$ に部分積分を適用すると

  $$\iint \frac{\partial^2 f(u)}{\partial x^2} dxdy = \oint \frac{\partial f(u)}{\partial x} dy = \oint \frac{\partial f(u)}{\partial x} \alpha dl$$

  が導かれます。この項は 2 階微分であるため、常に自然境界条件への寄与項が導かれます。

- 項 $\nabla \cdot \boldsymbol{F}(u)$ に部分積分を適用すると発散定理

  $$\iint \nabla \cdot \boldsymbol{F}(u) dxdy = \oint \boldsymbol{F}(u) \cdot \widehat{n} dl$$

  が導かれます。ここに $\widehat{n}$ は境界上の外向き単位法線ベクトルを表します。最初の例の場合と同様、ベクトル $\boldsymbol{F}$ 自身に u の導関数が含まれていない場合には部分積分は適用されません。

- 項 $\nabla \times \boldsymbol{F}(u)$ に部分積分を適用すると回転定理

  $$\iint \nabla \times \boldsymbol{F}(u) dxdy = \oint \widehat{n} \times \boldsymbol{F}(u) dl$$

  が導かれます。

これらの公式を用いて自然境界条件の意味をチェックしておきましょう。

熱拡散方程式

```
Div(-k*grad(Temp)) + Source = 0
Natural(Temp) = 外向き法線方向流束 = normal(-k*grad(Temp))
```

Note: 偏微分方程式中、熱流束に対しては負の符号を付してある点に注意してください。この符号を付けなかった場合には Natural の指定も反転させる必要があります。

1 次元熱拡散方程式

```
dx(-k*dx(Temp)) + Source = 0
Natural(Temp) = 流束の外向き法線成分 = (-k*dx(temp)*nx)
```

ただし nx は境界上の法線の x 方向余弦。x 以外の座標変数に対しても同様。

磁場方程式

```
curl(curl(A)/mu) = J
Natural(A) = H の接線方向成分 = tangential(curl(A)/mu)
```

対流方程式

```
dx(u)-dy(u)=0
```

2 階の項がないため Natural(u) は定義されない。

5.2.2 流束型境界条件の例

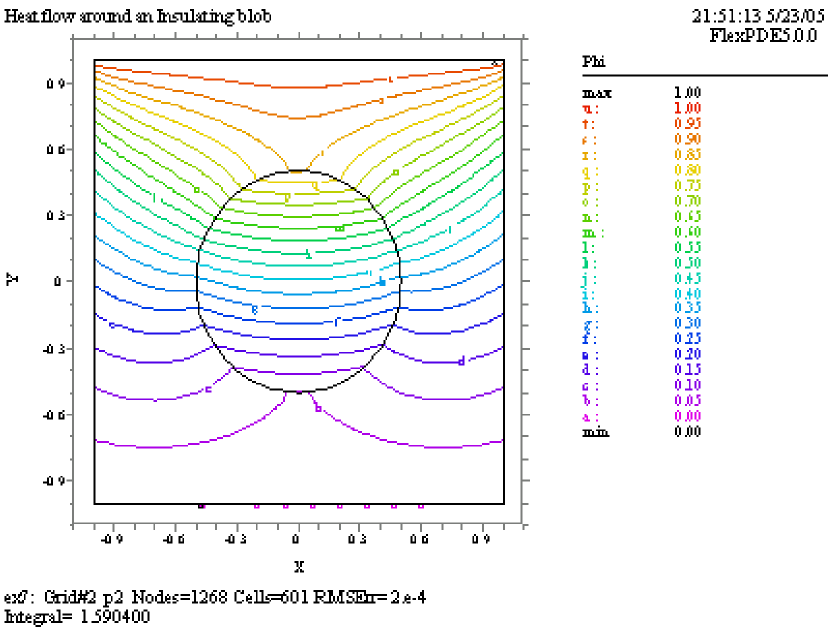
我々の熱流束の問題に戻り、自然境界条件による影響について見てみましょう。最初の例では左右の境界上で Natural(Phi)=0 という指定を行いました。これは境界上で流束が 0 であることに対応しています。一方、対流による熱損失がある場合の境界条件は

```
Flux = -K*grad(Phi) = Phi -- Phi0
```

と表現できます。ここで Phi0 は基準の冷却温度を意味します。今 Phi0=0 として対流による熱損失がある場合の境界条件を記述してみましょう。

```
REGION 1 'box'
  START(-1,-1)
  VALUE(Phi)=0        LINE TO (1,-1)
  NATURAL(Phi)=Phi    LINE TO (1,1)
  VALUE(Phi)=1        LINE TO (-1,1)
  NATURAL(Phi)=Phi    LINE TO CLOSE
```

結果は次に示すように両側で等温線（isotherms）が上向きとなります。



5.3 不連続な変数

デフォルトの場合、FlexPDE はすべての変数値が材質境界をまたがって連続であると仮定します。これは境界上のノードは両側の材質によって共用されるとする有限要素法のモデルからして自然な帰結と言えます。

しかしスクリプト中で CONTACT とか JUMP というキーワードを用いることによって材質境界上での変数の不連続性を取扱うことが可能になります。

CONTACT(V) は NATURAL 境界条件の特殊形です。この指定によって変数値は境界上で 2 重化されたノード内に格納されるようになるため、値の 2 重性に対応できるようになります。

JUMP(v) の場合には、境界領域の内側から外側に移動したときに変数"v"の値が瞬時に変化するような設定となります。材質'1' と'2' の境界を考えたとき、材質'1' 中での JUMP(V) は (V2-V1) を、材質'2' 中での JUMP(V) は (V1-V2) を意味します。

JUMP の想定される用途としては境界内側の CONTACT 境界条件中で使用する形態です。このように CONTACT と JUMP を併用することによって、2 つの値の差に比例した線状、あるいは面状の熱源を表現することができます。

JUMP は他の境界条件中でも使用できますが、JUMP の引数は CONTACT で指定したものと同一であると仮定されます。用例については
"Samples | Misc | Discontinuous_Variables | Contact_Resistance_Heating.pde"
を参照ください。

JUMP 演算子の解釈は次節で説明する接触抵抗のモデルに則ったものです。

5.3.1 接触抵抗

2 つの伝導体間の**接触抵抗**の問題はモデル変数の不連続性を伴う典型例です。

この問題の場合、非常に薄い抵抗層が境界の両側で温度や電位のジャンプを引き起こします。ジャンプの大きさは抵抗層を横切る熱流や電流に比例します。顕微鏡で見れば抵抗物質に物理的広がりがあるわけですが、有限要素法モデルで忠実にモデル化し得るだけの厚みがあるわけではありません。

接触抵抗の場合、材質'1' と'2' の間の抵抗層を横切る熱流束は'1' の側から見て

```
F1 = -K1*dn(T) = -(T2-T1)/R
```

と定式化されます。ここに

- F1 は外向きの熱流束の値
- K1 は熱伝導率
- dn(T) は T の外向き法線微分
- R は境界層の抵抗
- T1, T2 は境界における 2 つの温度値

を意味します。

一方、'2' の側から見たときは

```
F2 = -K2*dn(T) = -(T1-T2)/R = -F1
```

となります。法線は逆の符号を持つため'2' からの流れは'1' からの流れの符号を反転させたものとなります（エネルギー保存)。

熱拡散方程式

```
div(-K*grad(T)) = H
```

の自然境界条件は発散定理より外向き熱流束を表す

```
Natural(T) = -K*dn(T)
```

で与えられます。

NATURAL の代りに CONTACT 境界条件を使用することにより、この熱流束を不連続変数と関係付けることができます。

FlexPDE における表現 JUMP(T) は材質'1' 中では（T2-T1）と、材質'2' 中では（T1-T2）と定義されます。従って接触抵抗の境界条件は次のように表されます。

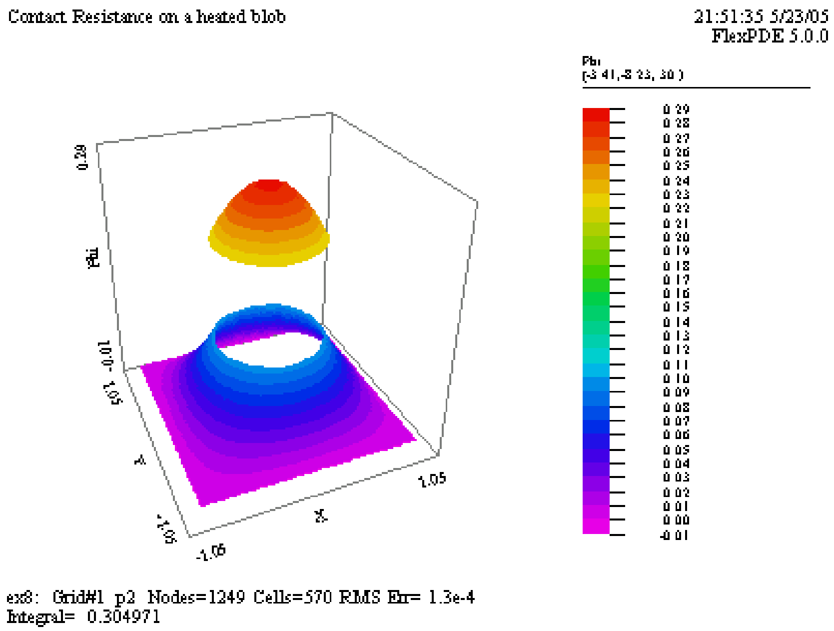
```
CONTACT(T) = -JUMP(T)/R
```

このステートメントは境界をシェアする双方の材質において同じことを意味します。（JUMP に付された符号は発散項の符号を反映したものです。）

これまでの例にジャンプを引き起こす熱源を加えた形でスクリプトを変更してみましょう。なお、正方形領域の 4 辺とも今回は温度 0 に保つものとします。

```
TITLE 'Contact Resistance on a heated blob'
VARIABLES
  Phi            { the temperature }
DEFINITIONS
  K = 1          { default conductivity }
  R = 0.5        { blob radius }
  H = 0          { internal heat source }
  Res = 0.5      { contact resistance }
EQUATIONS
  Div(-k*grad(phi)) = H

BOUNDARIES
  REGION 1 'box'
    START(-1,-1)
    VALUE(Phi)=0    { cold outer walls }
    LINE TO (1,-1) TO (1,1) TO (-1,1) TO CLOSE
  REGION 2    'blob'    { the embedded blob }
    H = 1           { heat generation in the blob }
    START 'ring' (R,0)
    CONTACT(phi) = -JUMP(phi)/Res
    ARC(CENTER=0,0) ANGLE=360 TO CLOSE
PLOTS
  CONTOUR(Phi)
  SURFACE(Phi)
  VECTOR(-k*grad(Phi))
  ELEVATION(Phi) FROM (0,-1) to (0,1)
  ELEVATION(Normal(-k*grad(Phi))) ON 'ring'
END
```

FlexPDE によって出力された曲面（surface）プロットには温度の不連続性が明確に表現されています。



5.3.2　連結性解除

接触抵抗のモデルを使うと隣接したリージョン間で変数値の連結を実質的に解除することができます。前の例においてジャンプ境界条件を次のように設定してみましょう。

```
CONTACT(phi) = 0*JUMP(phi)
```

この場合、接触抵抗の値が無限となるためリージョン間での流束の交換は行われなくなります。

Note:　JUMP ステートメントは特別な扱いを受けます。右辺の値は明らかに 0 であるわけですが、簡略化の過程で除去されることはありません。

5.3.3　複数変数の扱い

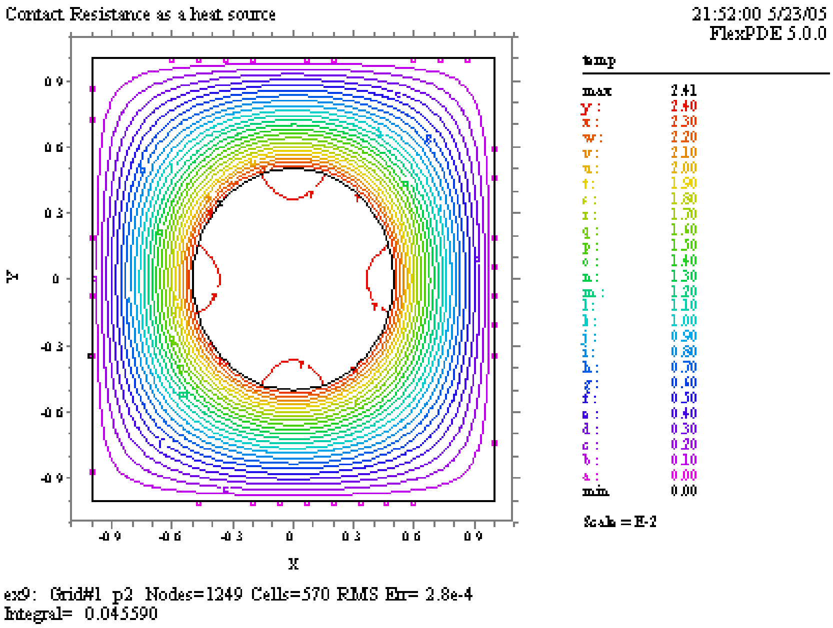
JUMP(V) という表現は、変数 V に関し CONTACT 境界条件が設定されている境界上であれば、任意の境界条件ステートメント中で使用できます。

例えば電気抵抗の場合、電位が接触抵抗の両端でジャンプすることになるので、そこを流れる電流が熱流方程式に対する熱源となります。次の例は厳密に言うと物理的に実現できるわけではありませんが、テクニックの紹介には役に立ちます。Temp に対する Natural 境界条件中に熱源を表す項として JUMP(Phi) という記述が出てくる点に注意してください。Phi は CONTACT 境界条件中に規

定されているので、2重に値を持つ量として記録されています。従ってその JUMP が Temp に対する境界条件でも使えるようになるわけです。Temp は CONTACT 境界条件中に出てこないので、接合の境界上で一つの値しか取りません。

```
TITLE 'Contact Resistance as a heat source'
VARIABLES
  Phi            { the voltage }
  Temp           { the temperature }
DEFINITIONS
  Kd = 1         { dielectric constant }
  Kt = 1         { thermal conductivity }
  R = 0.5        { blob radius }
  Q = 0          { space charge density }
  Res = 0.5      { contact resistance }
EQUATIONS
  Phi:     Div(-kd*grad(phi)) = Q
  Temp:    Div(-kt*grad(temp) = 0
BOUNDARIES
  REGION 1 'box'
    START(-1,-1)
    VALUE(Phi)=0    { grounded outer walls }
    VALUE(Temp)=0   { cold outer walls }
    LINE TO (1,-1) TO (1,1) TO (-1,1) TO CLOSE
  REGION 2    'blob'    { the embedded blob }
    Q = 1          { space charge in the blob }
    START 'ring' (R,0)
    CONTACT(phi) = -JUMP(phi)/Res
    { the heat source is the voltage difference times the current }
    NATURAL(temp) = -JUMP(Phi)^2/Res
    ARC(CENTER=0,0) ANGLE=360 TO CLOSE
PLOTS
  CONTOUR(Phi)    SURFACE(Phi)
  CONTOUR(temp)   SURFACE(temp)
END
```

等温線図には境界上の熱源の効果が示されています。



第 6 章

1 次元の問題への適用

FlexPDE は 1 次元の問題を 2 次元のそれの縮退したものとして扱います。問題の記述の仕方はこれまでのものと大きく変わりませんが、次のような 1 次元固有の配慮が必要となります。

- COORDINATES の指定は CARTESIAN1, CYLINDER1, SPHERE1 のいずれかを使用します。
- 座標点の指定には次のように 1 次元座標を用います。

  ```
  START(0) LINE TO (5)
  ```

- 境界を規定するパスはドメインそのものに一致してしまいます。このため境界条件の規定はその形では行えません。ドメインの両端における境界条件の指定には既存の POINT VALUE, POINT LOAD を使用してください。

  ```
  START(0) POINT VALUE(u)=0 LINE TO (5) POINT LOAD(u)=1
  ```

- 1 次元においては ELEVATION と HISTORY のみが意味を持つプロットとなります。

次は球面座標系を使用した例です。2 つの球状の熱源の間に球殻上の非伝導層がはさまったモデルを想定しています。

```
TITLE 'Heat flow through an Insulating shell'
COORDINATES
  Sphere1
VARIABLES
  Phi       { the temperature }
DEFINITIONS
  K = 1     { default conductivity }
  R1 = 1    { the inner reservoir }
  Ra = 2    { the insulator inner radius }
  Rb = 3    { the insulator outer radius }
  R2 = 4    { the outer reservoir }
EQUATIONS
  Div(-k*grad(phi)) = 0

BOUNDARIES
  REGION 1          { the total domain }
    START(R1)       POINT VALUE(Phi)=0
    LINE TO (R2)    POINT VALUE(Phi)=1
    { note:  no 'Close'!  }
  REGION 2    'blob'    { the embedded layer }
    k = 0.001
    START (Ra) LINE TO (Rb)
PLOTS
  ELEVATION(Phi) FROM (R1) to (R2)
END
```

第 7 章

3 次元の問題への適用

3 次元の問題を扱うのは容易なことではありません。FlexPDE の使いやすさはこれまでも改良されてきましたが、それでも 3 次元の問題のセットアップと実行には 2 次元の問題に FlexPDE を適用する際の数多くのノウハウが前提となります。先行するセクションをスキップして直接ここに飛び込むのはお勧めできません。

FlexPDE は 2 次元のドメインを第 3 の次元に押し出す（extrude）形で 3 次元ドメインを構成します。この第 3 の次元をさらにレイヤ分けすることが可能です。その場合、それぞれのレイヤに別個の材質特性や境界条件を持たせることが可能です。レイヤ間の境界面は平面でなくても構いませんが、その形状には多少の制約が付きます。

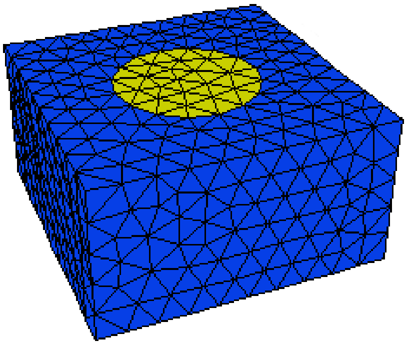
ドメイン定義プロセスとは異なり、FlexPDE によってこのような 3 次元ドメイン内に構成される有限要素モデル自体には汎用性があります。

7.1　押出しの概念

押出し（extrusion）に関する基本的な考え方は単純です。正方形を第 3 の次元に押出せば立方体に、円を押出せば円柱になります。球面座標系を想定するなら円は球になります。

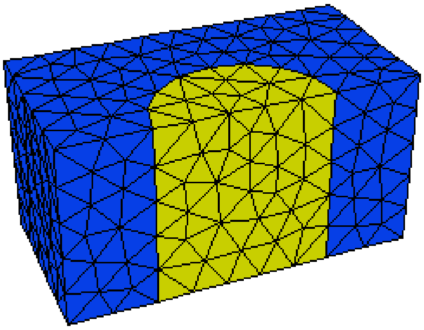
Note: 与えられた問題の特質を良く考え、押出しに最も適した方向を見極めることが大切です。

これまでの 2 次元熱流体問題を第 3 の次元に押出したらどうなるでしょうか。押出す長さを上下の平板間の距離の半分とすると、煉瓦中に円柱が埋め込まれたような形状を得ることができます。



任意の z の点で横断面を作成するとオリジナルの 2 次元の図形が現れます。

$y = 0$ での横断面には押出された構造が明白に示されています。



7.2　押出しの設定

上記のような押出しを行わせるには 2 次元用スクリプトに対し 3 つの基本的変更を加える必要があります。

- COORDINATES セクションでは CARTESIAN3 を指定します。
- 押出しのレイヤ構成を指定するため EXTRUSION セクションを追加する必要があります。
- 切断面の表示を行わせるために PLOTS, MONITORS に対する指定を変更します。

EXTRUSION セクションの記法には詳細型と短縮型の 2 種類があります。どちらの場合にもモデルのレイヤは Z の小さい方から大きい方へと積み重なって行きます。

詳細型の場合、**分割面**SURFACE とその間の**レイヤ**LAYER に対しそれぞれ名称を設定します（分割面に対してはその数式も指定します)。

Note: SURFACE という用語に対し 2 重の意味を持たせてある点に注意してください。プロットコマンドの中ではデータを表示するための曲面を意味します。これに対し押出しとの関連においてはレイヤ間の分割面という意味合いで使用されます。どちらを指すかは文脈から明白なはずです。

上記の単純な円柱の場合には EXTRUSION セクションの指定は次のようになります。

```
EXTRUSION
  SURFACE 'Bottom'     z=0
  LAYER 'Everything'
  SURFACE 'Top'        z=1
```

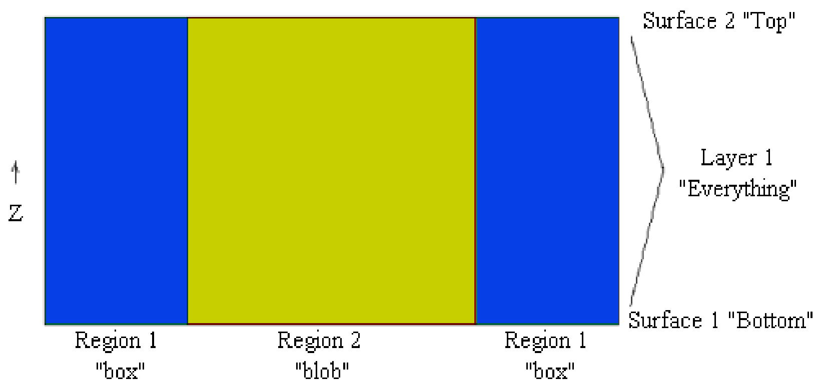
頂部と底部の分割面（平面）に対し名称とその数式が指定されています。この分割面間のレイヤがドメイン全体に対応するため、ここでは'Everything' と名付けられています。

短縮型の場合、単に分割面の数式のみを指定します。

```
EXTRUSION    z = 0, 1
```

この場合、レイヤと分割面を参照するには番号を使用することになります。最初の分割面 $z = 0$ は"SURFACE 1"という形で、第 2 の分割面 $z = 1$ は"SURFACE 2"という形で参照されます。

レイヤ定義に関する限り、円柱内の部分と円柱外の部分とに区別はない点に注意してください。これを区別するためには**レイヤ**の概念と 2 次元平面上での**リージョン**の概念とを組み合わせることになります。次の図は縦方向の横断面を表示したもので、レイヤとリージョンの区分けが示されています。



円柱領域は'blob' リージョンと'Everything' レイヤとの積集合としてユニークに特定されます。

7.3 レイヤリング

今度は全長にわたる円柱ではなくキャニスターをモデル化したいものとしましょう。そのためにはリージョン 2 上のスタックを 3 つの部分、すなわちキャニスター、そしてその上下に位置する'box'の延長部とに分割する必要があります。

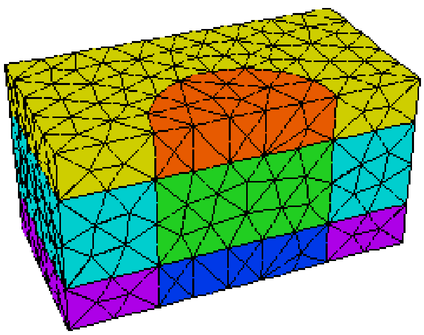
次の設定では 3 つのレイヤを規定していますが、そのうち中央のレイヤがキャニスターを含むレイヤです。

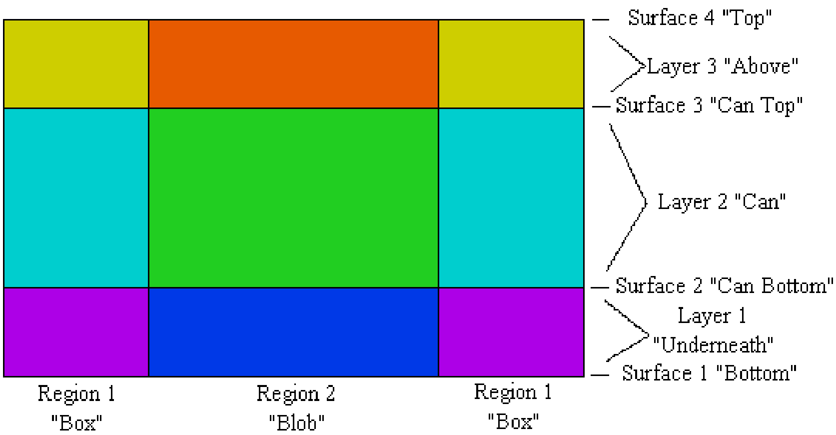
```
EXTRUSION
  SURFACE "Bottom"        z=-1/2
    LAYER "Underneath"
  SURFACE "Can Bottom"    z=-1/4
    LAYER "Can"
  SURFACE "Can Top"       z=1/4
    LAYER "Above"
  SURFACE "Top"           z=1/2
```

これによって 3 次元のオブジェクトを 6 つの**区画**（compartments）に分けたことになります。すなわち 2 つのリージョンと 3 つのレイヤの組合せです。

これらの区画に対してはそれぞれ独自の材質特性と境界条件を設定することができます。

断面図を示すと次のようになります。





一見すると 9 つの区画が存在するように見えますが、リージョン 1 は円柱部を完全に取り囲んでいる点に注意してください。個々のレイヤ内において左右にあるリージョン 1 の領域はつながっているのです。上の図において 6 色の色分けが施されていますが、各々が 6 個の区画に対応しています。

それぞれの区画（compartment）を別個の entity として規定する必要はありませんし、またそうすることは誤りでもあることを強調しておきます。個々の材質区画ごとにレイヤやリージョンの指定を行う必要はありません。基盤平面上でのリージョンの定義、あるいは extrusion 中でのレイヤの定義を繰り返すことは混乱の元となります。

任意の区画は基盤平面上での `REGION` と extrusion 中の `LAYER` の組合せで特定できるのです。

7.4　材質特性の設定

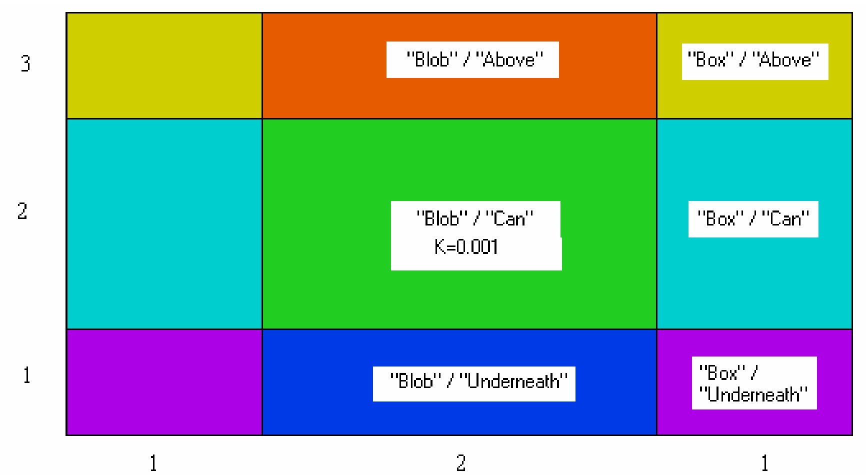
2 次元の問題の場合には blob の熱伝導率を blob に対する `REGION` 定義の中で指定しましたが、このアプローチは 3 次元の場合にも適用されます。

違いは適用対象となる `LAYER` についても指定が必要となる点です。具体的には次のように `LAYER` 修飾した文節内で伝導率を指定します。

```
REGION 2    'blob'    { the embedded blob }
  LAYER 'Can' K = 0.001
  START 'ring' (R,0)
  ARC(CENTER=0,0) ANGLE=360
```

このような LAYER 修飾型文節を使わなかった場合には、指定された伝導率はリージョン 2 上に位置する全レイヤに適用されることになります。結局パラメータ定義がどの REGION 内で行われたか、及びそのときの LAYER 修飾項が適用対象となる 3 次元区画（compartment）を特定するわけです。

次の図では 6 つの異なる区画に対して（Region, Layer）の組みを示してあります。



このように BOUNDARIES セクション内でそれぞれの区画に対するパラメータ値を再定義する場合には次のような指定方法を取ってください。

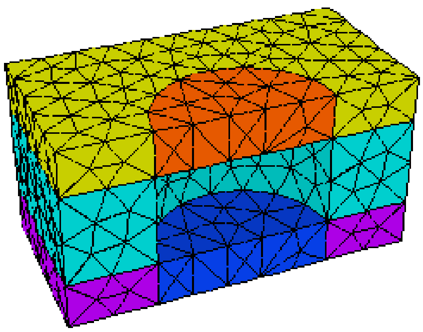
```
BOUNDARIES

REGION 1
  params(1,all)
  { parameter redefinitions for all layers of region 1 }
  LAYER 1
    params(1,1)
    { parameter redefinitions restricted to layer 1 of region 1 }
  LAYER 2
    params(1,2)
    { parameter redefinitions restricted to layer 2 of region 1 }
  LAYER 3
    params(1,3)
    { parameter redefinitions restricted to layer 3 of region 1 }
  START(,)  ....   TO CLOSE    { trace the perimeter }
```

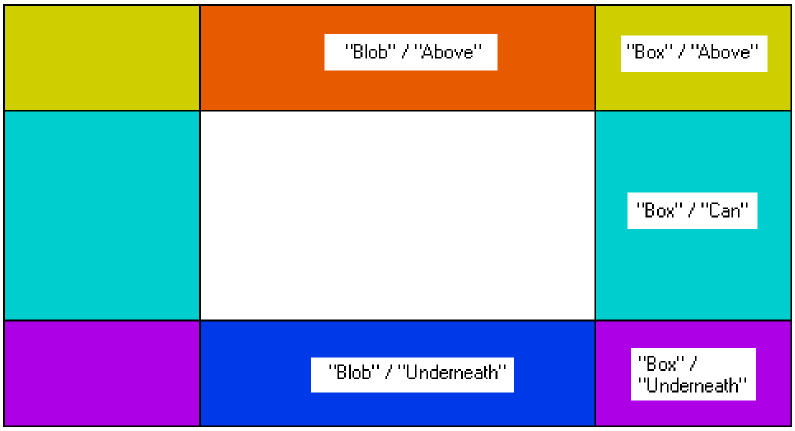
```
REGION 2
  params(2,all)
  { parameter redefinitions for all layers of region 2 }
  LAYER 1
    params(2,1)
    { parameter redefinitions restricted to layer 1 of region 2 }
  LAYER 2
    params(2,2)
    { parameter redefinitions restricted to layer 2 of region 2 }
  LAYER 3
    params(2,3)
    { parameter redefinitions restricted to layer 3 of region 2 }
  START(,)  ....   TO CLOSE    { trace the perimeter }

{ ...  and so forth for all regions }
```

7.5 空の区画

予約語 VOID は構文的にはパラメータ再定義と同様に扱われます。もしこの語が上記の LAYER 修飾項のいずれかに置かれていた場合、該当する**区画**（compartment）（(Region, Layer）の組み）はドメインから除去されることになります。

```
REGION 2    'blob'    { the embedded blob }
  LAYER 'Can' VOID
  START 'ring' (R,0)
  ARC(CENTER=0,0) ANGLE=360
```





具体例については”Samples ｜ Misc ｜ 3D_Domains ｜ 3D_Void.pde”を参照ください。

7.6 限定リージョン

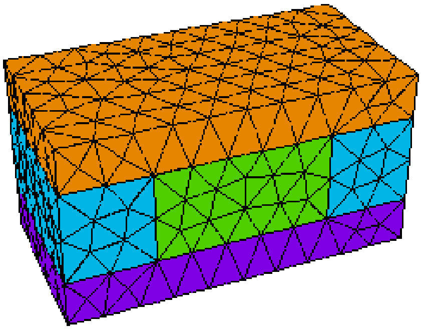
これまでの議論では 2 次元の基盤平面上で定義されたリージョンの構成がそのまま extrude された第 3 の次元にも伝播していました。しかし LIMITED REGION という指定を使うとその設定を特定のレイヤ、あるいは表面に限定することができます。

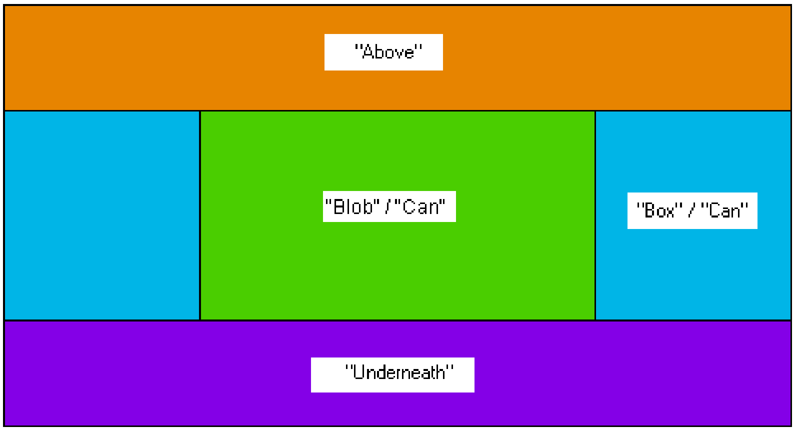
LIMITED REGION の場合、extrude された次元全体に適用されるのではなく、宣言中で明示されたレイヤ、あるいは表面上にのみ存在することになります。レイヤ（layers）を指定した場合、LIMITED REGION はそのレイヤとその境界となる表面上に設定されます。一方、表面（surfaces）を指定した場合には LIMITED REGION はその表面上に設定されます。

次は LIMITED REGION の設定例です。

```
LIMITED REGION 2    'blob'    { the embedded blob }
  LAYER 'Can' K = 0.001
  START 'ring' (R,0)
  ARC(CENTER=0,0) ANGLE=360 TO CLOSE
```

この設定の場合、キャニスターは”Above”と”Underneath”のレイヤには伝播しなくなります。





7.7　切断面上のプロット

2 次元の問題においては CONTOUR, SURFACE, VECTOR, GRID 機能を用いることによって、平面上における計算値をグラフ化することができました。

3 次元の場合、3 次元オブジェクトに対する任意の切断面上で同様の表示を行うことができます。具体的には**切断面**を規定する数式をプロットコマンド中の'ON' という修飾詞に続けて書くだけです。

```
PLOTS
  CONTOUR(Phi) ON x=0
```

Note:　ON クローズのさらなる用法については後述します。

GRID コマンドを使用すると計算グリッド（と暗黙的にはドメイン構造）を表示させることができます。

```
GRID(x,z) ON y=0
```

このコマンドは FlexPDE で使用されている四面体のメッシュ構造とプロット平面との交差をグラフ化します。その際、切断面上に存在する材質の特性に応じて色分けが行われます。従って同一の特性を持った 3 次元区画（compartments）は同一の色で表示されることになります。GRID プロットの引数は横軸と縦軸に表示する値を示しています。引数を変更することによって歪んだグリッドの表示も可能です。

7.8　3D キャニスター用スクリプト

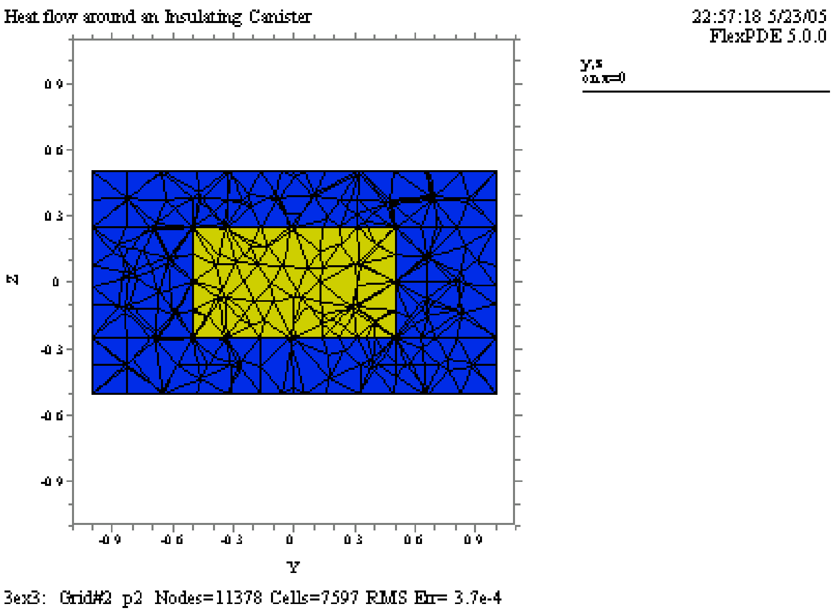
以上の変更をすべて加えた 3D キャニスター分析用スクリプトを次に示します。

```
TITLE 'Heat flow around an Insulating Canister'
COORDINATES
  Cartesian3
VARIABLES
  Phi            { the temperature }
DEFINITIONS
  K = 1          { default conductivity }
  R = 0.5        { blob radius }
EQUATIONS
  Div(-k*grad(phi)) = 0
EXTRUSION
  SURFACE 'Bottom'        z=-1/2
    LAYER 'underneath'
  SURFACE 'Can Bottom'    z=-1/4
    LAYER 'Can'
  SURFACE 'Can Top'       z=1/4
    LAYER 'above'
  SURFACE 'Top'           z=1/2
```

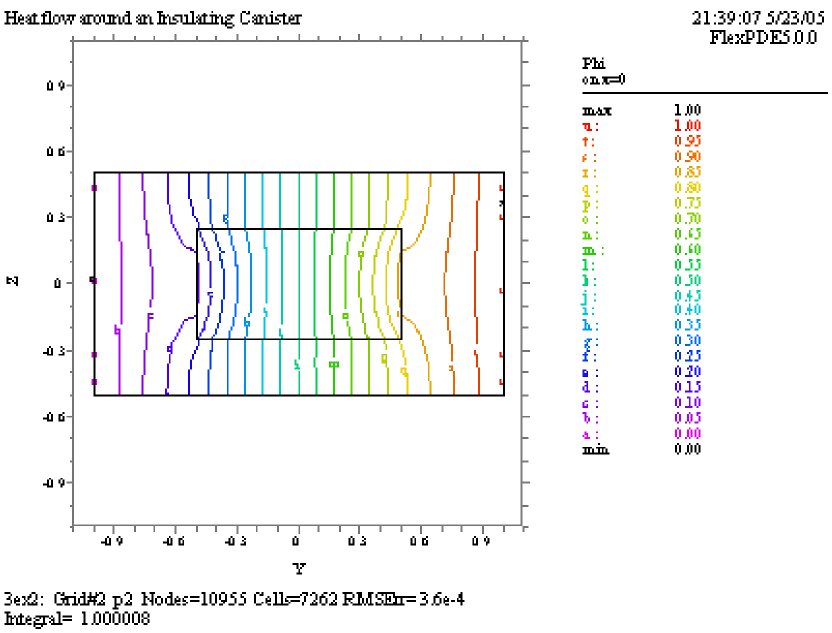
```
BOUNDARIES
  REGION 1 'box'
    START(-1,-1)
    VALUE(Phi)=0        LINE TO (1,-1)
    NATURAL(Phi)=0      LINE TO (1,1)
    VALUE(Phi)=1        LINE TO (-1,1)
    NATURAL(Phi)=0      LINE TO CLOSE
  LIMITED REGION 2    'blob'    { the embedded blob }
    LAYER 2 k = 0.001   { the canister only }
    START 'ring' (R,0)
    ARC(CENTER=0,0) ANGLE=360 TO CLOSE
PLOTS
  GRID(y,z) ON x=0
  CONTOUR(Phi) ON x=0
  VECTOR(-k*grad(Phi)) ON x=0
  ELEVATION(Phi) FROM (0,-1,0) to (0,1,0)    { note 3D coordinates }
END
```

Extrude された上面と下面については境界条件が指定されていないので、流束 0 という条件が仮定されます。これは自然境界条件に基づく標準的なデフォルトです。

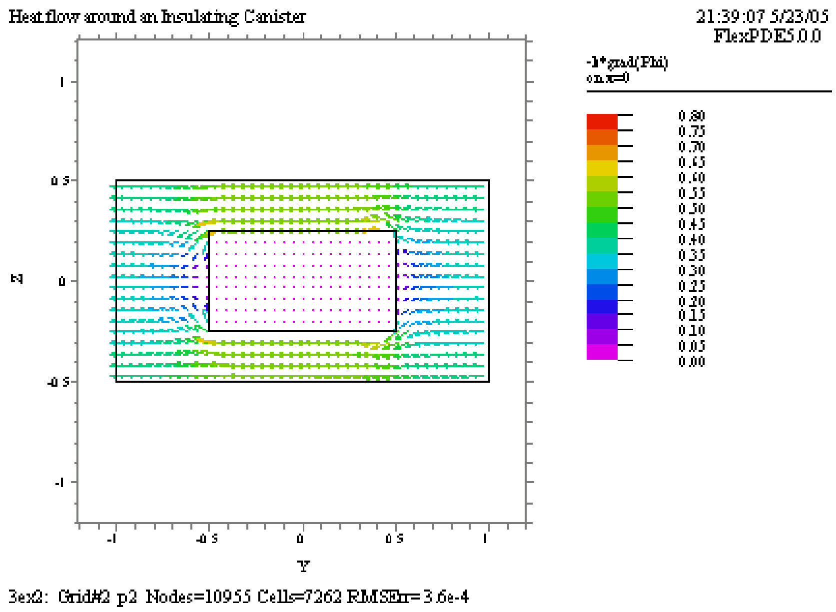
PLOTS コマンドで指定された項のうち最初の 3 つのグラフは次のようになります。



Heat flow around an Insulating Canister
22:57:18 5/23/05
FlexPDE 5.0.0
Y,Z
on x=0
Z
Y
3ex3: Grid#2 p2 Nodes=11378 Cells=7597 RMS Err= 3.7e-4



Heat flow around an Insulating Canister
21:39:07 5/23/05
FlexPDE5.0.0
Phi
on x=0
max 1.00
u : 1.00
t : 0.95
s : 0.90
r : 0.85
q : 0.80
p : 0.75
o : 0.70
n : 0.65
m : 0.60
l : 0.55
k : 0.50
j : 0.45
i : 0.40
h : 0.35
g : 0.30
f : 0.25
e : 0.20
d : 0.15
c : 0.10
b : 0.05
a : 0.00
min 0.00
Z
Y
3ex2: Grid#2 p2 Nodes=10955 Cells=7262 RMSErr=3.6e-4
Integral= 1.000008

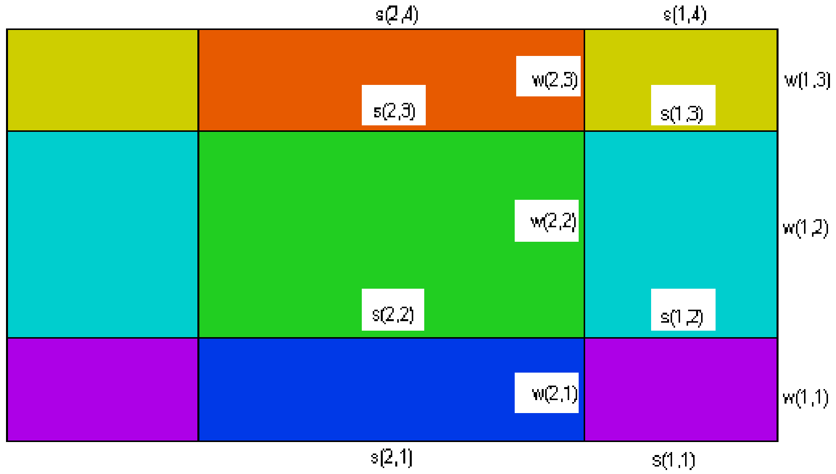


7.9　境界条件の設定（3 次元）

3 次元の問題における境界条件の指定方法は 2 次元におけるアプローチを拡張したものです。

- 2 次元における境界線上に設定された境界条件は、その曲線を extrude することによって生成される側面に適用されます。
- `LAYER` <number>, あるいは `LAYER` <name> という修飾詞を用いることにより、境界条件の適用を側面上の特定のレイヤに限定することができます。
- Extrude された表面に対する境界条件は、リージョンに対する、あるいは 2 次元ドメイン全体に対するパラメータ再定義と同じ考え方で設定されます。その場合、境界条件の定義の前に修飾詞 `SURFACE` <number>, または `SURFACE` <name> を置く必要があります。

次の図では別個に境界条件が設定し得る種々の表面に対しラベルを付してあります。レイヤ間の境界面に対しては`"s"`というラベルを、側面に対しては壁という意味合いで`"w"`というラベルを用いています。この図では断面図に現れる側面しか示してありませんが、実際には基盤平面上の境界線を表す個々の線分/円弧が別個の側面を形成し得るわけで、それらすべてに対し`"w"`タイプの境界条件を設定できます。



これらの境界面/側面に対する境界条件をすべて設定するとすれば次のような構成となります。

```
BOUNDARIES
  SURFACE 1
    s(all, 1)  { boundary conditions on surface 1 over full domain }
  SURFACE 2
    s(all, 2)  { boundary conditions on surface 2 over full domain }
    {...other surfaces }
  REGION 1
    SURFACE 1
      s(1,1)  { boundary conditions on surface 1, restricted to region 1 }
    SURFACE 2
      s(1,2)  { boundary conditions on surface 2, restricted to region 1 }
      ...
      START(,)  { -- begin the perimeter of region m }
        w(1,..)  { boundary conditions on following segments of sidewall
                   of region 1 on all layers }
        LAYER 1
          w(1,1)  { boundary conditions on following segments of sidewall
                    of region 1, restricted to layer 1 }
        LAYER 2
          w(1,2)  { boundary conditions on following segments of sidewall
                    of region 1, restricted to layer 2 }
```

```
      ...
    LINE TO ....
      { segments of the base plane boundary with above BC's }
      LAYER 1
        w(1,1)  { new boundary conditions on following segments of
                  sidewall of region 1, restricted to layer 1 }
    ...
    LINE TO ....
      { continue the perimeter of region 1 with modified boundary
        conditions }
    TO CLOSE
REGION 2
  SURFACE 1
    s(2,1)  { boundary conditions on surface 1, restricted to region 2 }
  SURFACE 2
    s(2,2)  { boundary conditions on surface 2, restricted to region 2 }
    ...
  START(,)  { -- begin the perimeter of region m }
    w(2,..)  { boundary conditions on following segments of sidewall of
               region 2 on all layers }
    LAYER 1
      w(2,1)  { boundary conditions on following segments of sidewall of
                region 2, restricted to layer 1 }
    LAYER 2
      w(2,2)  { boundary conditions on following segments of sidewall of
                region 2, restricted to layer 2 }
    ...
  LINE TO ....
    { segments of the base plane boundary with above BC's }
    LAYER 1
      w(2,1)  { new boundary conditions on following segments of sidewall
                of region 2, restricted to layer 1 }
      ...
  LINE TO ....
  {continue the perimeter of region 2 with modified boundary conditions}
  TO CLOSE
```

2次元の場合と同様、スクリプト中で後に現れるリージョンは前に出てきたリージョンの一部、もしくはすべてを上書きすることになります。例えば基盤平面上の REGION 1 の広がりは、その外周によって規定されるもののうちで他のリージョンには含まれない部分となります。

例えば今、キャニスターの表面に一定の温度"Tcan"を設定したいとしましょう。ドメイン中のキャニスター部は3つの表面、すなわち底面、上面、側面から構成されます。

キャニスターの底面、上面を規定するレイヤ境界の表面には'Can Bottom', 'Can Top' という名称を付けてあります。今温度を設定したい領域はこれらの表面の中で REGION 2 の上部に位置する部分だけなので、REGION 2 の定義内において SURFACE という修飾詞を付けて境界条件を設定します。

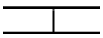
一方、キャニスターの側面は REGION 2 の境界線を extrude したもののうち、'Can' というレイヤに含まれる部分です。従ってここでは LAYER という修飾詞を付けて REGION 2 の境界線に対する境界条件を設定します。

BOUNDARIES セクション上での設定は次のようになります。

```
BOUNDARIES
  REGION 1 'box'
    START(-1,-1)
    VALUE(Phi)=0      LINE TO (1,-1)
    NATURAL(Phi)=0    LINE TO (1,1)
    VALUE(Phi)=1      LINE TO (-1,1)
    NATURAL(Phi)=0    LINE TO CLOSE
  REGION 2    'blob'    { the embedded blob }
    SURFACE 'Can Bottom' VALUE(Phi)=Tcan
    SURFACE 'Can Top' VALUE(Phi)=Tcan
    { parameter redefinition in the 'Can' layer only:  }
    LAYER 2 k = 0.001
    START 'ring' (R,0)
    { boundary condition in the 'Can' layer only: }
    LAYER 'Can' VALUE(Phi)=Tcan
    ARC(CENTER=0,0) ANGLE=360 TO CLOSE
```

7.10　レイヤ境界の変形

先にレイヤ境界は平面に限らないと述べました。しかし FlexPDE はレイヤ境界に対しある種の仮定を置いているため、設定できる形状には多少制約が付きます。

- 3 次元オブジェクトは側面とレイヤ境界を持ち、extrude された形状を維持したものでなくてはならない（側面はのばしたり縮めたりすることはできない)。
- レイヤ境界面はリージョン境界をまたがって連続でなくてはならない。垂直なジャンプを伴う境界面の場合、それは複数のレイヤに分ける必要がある。その際、ジャンプはリージョン境界で発生するものとし、個々のレイヤはジャンプの両側にまたがる形で設定する（ ではなく ）。
- 境界面同士はマージしても良いが、逆転することがあってはならない。（上下の）逆転を防ぐためには面の定義において `MAX`, `MIN` 関数を使用すると良い。

これらのルールに従ってキャニスターを球（上下の半球からなる）に変更してみましょう。

半球面を与える数式としては次のいずれかが使用できます。

```
Z = Zcenter + sqrt( R^2 -- x^2 -- y^2)
```

または

```
Z = Ztop -- R + sqrt(R^2 -- x^2 -- y^2)
```

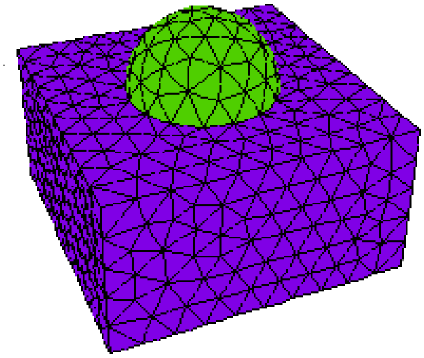
- 全体をカバーする直方体の上面、下面にこれら半球が接触することを回避するために、extrusion の範囲を -1 から 1 まで拡張します。
- 計算時のエラーを防ぐため、`sqrt` 中の引数は負にならないように工夫をします。

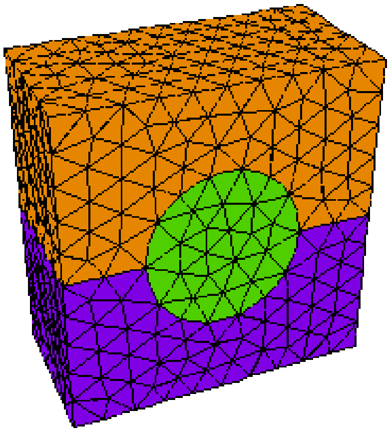
新たなスクリプトは次のようになります。

```
TITLE 'Heat flow around an Insulating Sphere'
COORDINATES
  Cartesian3
VARIABLES
  Phi          { the temperature }
```

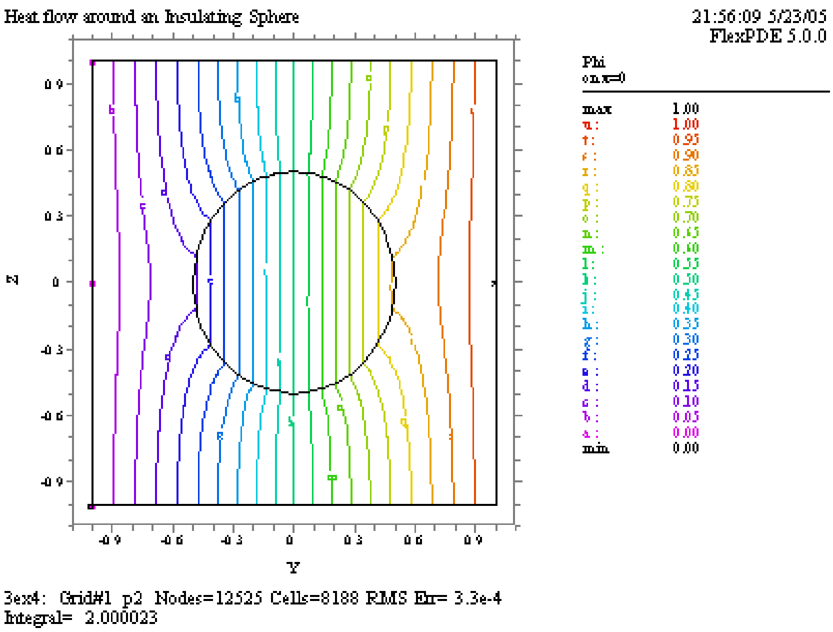
```
DEFINITIONS
  K = 1          { default conductivity }
  R = 0.5        { sphere radius }
  { shape of hemispherical cap:  }
  Zsphere = sqrt(max(R^2-x^2-y^2,0))
EQUATIONS
  Div(-k*grad(phi)) = 0
EXTRUSION
  SURFACE 'Bottom'           z=-1
    LAYER 'underneath'
  SURFACE 'Sphere Bottom'    z = -max(Zsphere,0)
    LAYER 'Can'
  SURFACE 'Sphere Top'       z = max(Zsphere,0)
    LAYER 'above'
  SURFACE 'Top'              z=1
BOUNDARIES
  REGION 1 'box'
    START(-1,-1)
    VALUE(Phi)=0      LINE TO (1,-1)
    NATURAL(Phi)=0    LINE TO (1,1)
    VALUE(Phi)=1      LINE TO (-1,1)
    NATURAL(Phi)=0    LINE TO CLOSE
  LIMITED REGION 2    'blob'    { the embedded blob }
    LAYER 2 K = 0.001
    START 'ring' (RSphere,0) ARC(CENTER=0,0) ANGLE=360
    TO CLOSE
PLOTS
  GRID(y,z) on x=0
  CONTOUR(Phi) on x=0
  VECTOR(-k*grad(Phi)) on x=0
  ELEVATION(Phi) FROM (0,-1,0) to (0,1,0)
END
```

左側の図は上半分から外周部を取り去ったもの、右側の図は平面 $x = 0$ による切断面です。





等温線図は次のようになります。



対象とした 3 次元オブジェクトの対称性から、このプロットは 2 次元の等温線図を回転させたものと類似の形状となっています。

7.11　3 次元における積分

3 次元の問題の場合、リージョンとレイヤとによって選択された 3 次元区画（volume compartments）上で**体積積分**が実行できます。

- `Result = VOL_INTEGRAL(<integrand>)`
  ドメイン全体を対象に `integrand` の積分を計算します。
- `Result = VOL_INTEGRAL(<integrand>, <region name>)`
  指定されたリージョン内の全レイヤを対象に `integrand` の積分を計算します。
- `Result = VOL_INTEGRAL(<integrand>, <layer name>)`
  指定されたレイヤ内の全リージョンを対象に `integrand` の積分を計算します。
- `Result = VOL_INTEGRAL(<integrand>, <region name>, <layer name>)`
  リージョン名、レイヤ名で規定される 3 次元区画を対象に `integrand` の積分を計算します。
- `Result = VOL_INTEGRAL(<integrand>, <region number>, <layer number>)`
  リージョン番号、レイヤ番号で規定される 3 次元区画を対象に `integrand` の積分を計算します。

選択された面上で**面積分**が実行できます。種々の修飾詞名を解析することにより、FlexPDE は面積分のステートメントでどの面が対象かを推定します。平面ではなかった場合には実際の表面積によって重み付けが行われます。

- `Result = SURF_INTEGRAL(<integrand>)`
  ドメインの外周を形成する面上で `integrand` の積分を計算します。
- `Result = SURF_INTEGRAL(<integrand>, <surface name> [, <layer_name>] )`
  指定された extrusion 面に含まれる全リージョン上で `integrand` の積分を計算します。オプショナルなレイヤ名によってレイヤを特定することもできます。
- `Result = SURF_INTEGRAL(<integrand>, <surface name>, <region name> [, <layer_name>] )`
  指定された extrusion 面のうち特定のリージョンに範囲を限定して `integrand` の積分を計算します。オプショナルなレイヤ名によってレイヤを特定することもできます。
- `Result = SURF_INTEGRAL(<integrand>, <region name>, <layer name>)`
  リージョン名、レイヤ名で規定される 3 次元区画のすべての面を対象に `integrand` の積分を計算します。値の評価は 3 次元区画内を前提に行われます。

- `Result = SURF_INTEGRAL(<integrand>, <boundary name>`
  `[, <region_name>] )`
  指定された基盤平面上の曲線を extrude することによって生成される側面に含まれるすべてのレイヤを対象に `integrand` の積分を計算します。オプショナルなリージョン名が指定された場合には、それはどちらの表面を対象に積分計算を行うかを規定します。指定されたレイヤに隣接しない面は計算の対象とはなりません。

- `Result = SURF_INTEGRAL(<integrand>, <boundary name>, <layer name>`
  `[, <region_name>] )`
  指定された基盤平面上の曲線を extrude することによって生成される側面のうち指定されたレイヤの部分を対象に `integrand` の積分を計算します。オプショナルなリージョン名が指定された場合には、それはどちらの表面を対象に積分計算を行うかを規定します。指定されたレイヤに隣接しない面は計算の対象とはなりません。

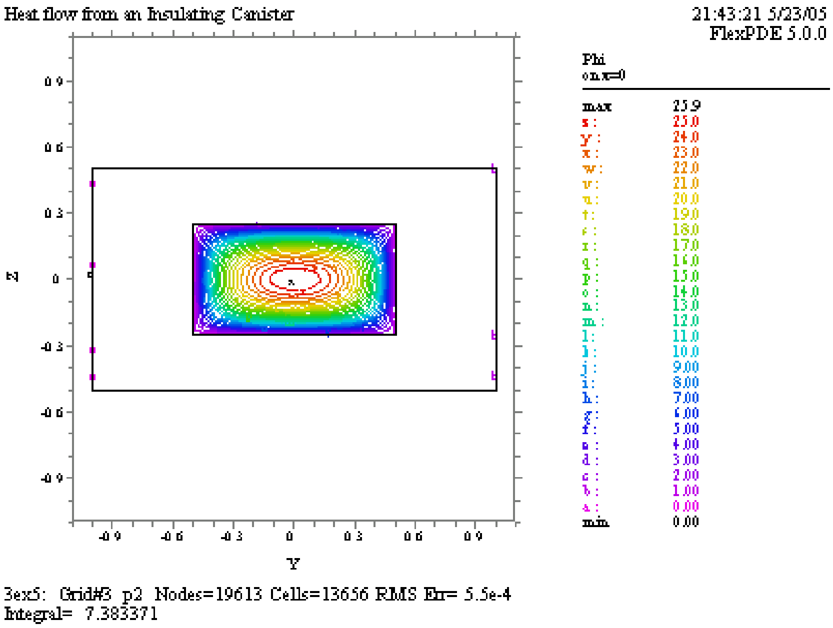
Note: これらの積分の用例については”`Samples | Misc | 3D_Integrals.pde`”を参照してください。

ここではキャニスターの問題に熱源を加え、その熱量の体積積分値を流束の面積分値と対比させます。これによって解の精度がチェックできます。

```
TITLE 'Heat flow from an Insulating Canister'
COORDINATES
  Cartesian3
VARIABLES
  Phi             { the temperature }
DEFINITIONS
  K = 1           { default conductivity }
  R = 0.5         { blob radius }
  S = 0
EQUATIONS
  Div(-k*grad(phi)) = S
```

```
EXTRUSION
  SURFACE 'Bottom'        z=-1/2
    LAYER 'underneath'
  SURFACE 'Can Bottom'    z=-1/4
    LAYER 'Can'
  SURFACE 'Can Top'       z=1/4
    LAYER 'above'
  SURFACE 'Top'           z=1/2
BOUNDARIES
  REGION 1 'box'
    START(-1,-1)
    VALUE(Phi)=0        LINE TO (1,-1)
    NATURAL(Phi)=0      LINE TO (1,1)
    VALUE(Phi)=1        LINE TO (-1,1)
    NATURAL(Phi)=0      LINE TO CLOSE
  REGION 2    'blob'    { option:  could be LIMITED }
    LAYER 2 k = 0.001   { the canister only }
    S = 1               { still the canister }
    START 'ring' (R,0)
    ARC(CENTER=0,0) ANGLE=360 TO CLOSE
PLOTS
  GRID(y,z) on x=0
  CONTOUR(Phi) on x=0
  VECTOR(-k*grad(Phi)) on x=0
  ELEVATION(Phi) FROM (0,-1,0) to (0,1,0)

SUMMARY
  REPORT(Vol_Integral(S,'blob','can')) AS 'Source Integral'
  REPORT(Surf_Integral(NORMAL(-k*grad(Phi),'blob','can')))
        AS 'Can Heat Loss'
  REPORT(Surf_Integral(NORMAL(-k*grad(Phi))))
        AS 'Box Heat Loss'
  REPORT(Vol_Integral(S,'blob','can')-Surf_Integral(NORMAL(-k*grad(Phi))))
        AS 'Energy Error'
END
```

等温線図は次のようになります。



サマリページは積分計算の結果を示しています。

```
SUMMARY

Source Integral=  0.392690
Can Heat Loss=  0.392680
Box Heat Loss=  0.392680
Energy Error=  1.048038e-5
```

Note: 等温線図下部に表示されている積分値はプロットプロセスがデフォルトで計算する `Area_Integral(Phi)` の値を示しています。

7.12 より高度なプロット制御

切断面上でのプロットの指定方法については既に議論しました。必要であれば積分の場合と同様なやり方でプロット範囲に制限を加えることができます。

さらに extrusion 面（レイヤ境界面）上でのプロットも指定することができます（平面である必要はありません)。

プロットに対する基本制御メカニズムは ON <thing> ステートメントです。例えばステートメント

```
CONTOUR(Phi) ON 'Sphere Top' ON 'Blob'
```

は extrusion 面'Sphere Top' 上のリージョン'Blob' の範囲においてポテンシャル Phi の等高線図作成を要求します。一方、

```
CONTOUR(NORMAL(-K*GRAD(Phi))) ON 'Sphere Top' ON 'Blob' ON 'Can'
```

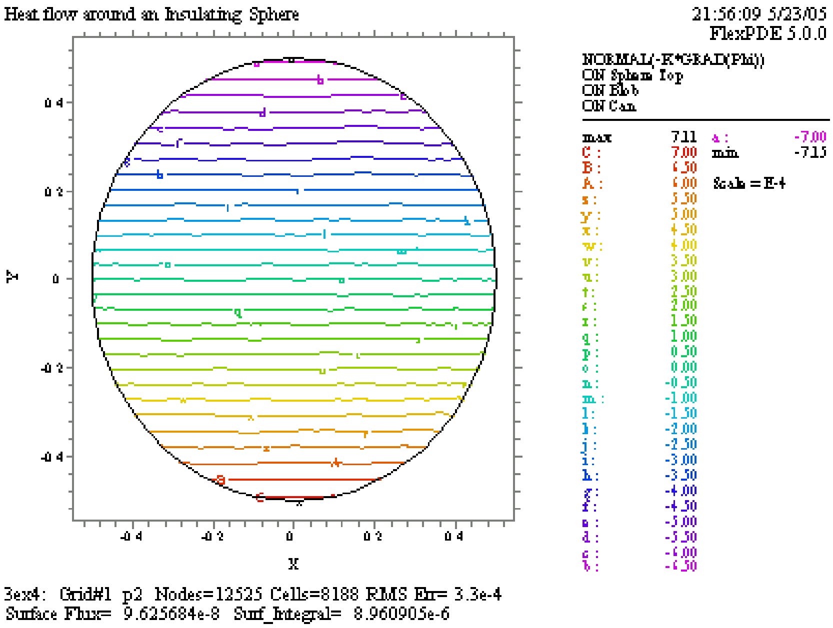
は上側の半球面上における熱流束の法線成分の値について等高線図の作成を要求します。ただしその場合、値の評価はレイヤ'Can' 側で、すなわち球面内部で行われます。

- 一般に修飾詞 ON <name> は指定されたオブジェクト名称のタイプに応じてプロット範囲のローカライズを行います。
- ON REGION <number> という指定の場合には番号によってリージョンが選択されます。
- ON SURFACE <number> という指定の場合には番号によってレイヤ境界面が選択されます。
- ON LAYER <number> という指定の場合には番号によってレイヤが選択されます。

一例として上に示した球面上での熱流束のプロットを要求してみましょう。このコマンドを PLOTS セクションに追加すると共に、同一曲面上での積分値も念のため計算しておくことにします。プロットジェネレータはプロットグリッド上で自動的に積分値を計算します。本来これは計算メッシュ上の積分値を用いた SURF_INTEGRAL の結果と一致すべきものです。

```
CONTOUR(NORMAL(-K*GRAD(Phi))) ON 'Sphere Top' ON 'Blob' ON 'Can'
REPORT(surf_integral(NORMAL(-k*GRAD(Phi)),'Sphere Top','Blob','Can'))
      AS 'Surface Flux'
```

結果は次の通りです。



この例の場合、積分値は 7e-4 前後の値がキャンセルした結果であるため、レポートされた 9.6e-8 という値はデフォルトの誤差許容値 `ERRLIM=0.001` を十分満たしています。一方、プロットグリッド上での計算値”`Surf_Integral`”はやや大きめな値 8.96e-6 を示しています。これはプロット用グリッドが計算用のグリッドに比べて目が粗いことによるものです。

第 8 章

メッシュ移動

FlexPDE は計算の実行中にドメイン境界と計算用メッシュを移動させる機能をサポートしています。

この機能は既存のスクリプト言語を拡張する形で実現されています。メッシュ移動に関わる定義変更は次の 3 つです。

- 移動させたい座標変数ごとに**代替変数**（surrogate variable）を宣言します。

  ```
  VARIABLES
    Xm = MOVE(x)
  ```

- 代替変数用の方程式を記述します。

  ```
  EQUATIONS
    dt(xm) = umesh
  ```

- 代替変数用の境界条件を記述します。

  ```
  BOUNDARIES
    START (0,0) VELOCITY(xm) = umesh
  ```

通常の方程式の記述は境界やメッシュの動きによって影響されません。EQUATIONS は常に**オイラー形式**（実験室形式）で記述します。FlexPDE は移動補正項をシンボリックな形で方程式に適用します。結果は ALE **モデル**（Arbitrary Lagrange/Eulerian model）に従ったものとなるわけですが、その場合、ユーザはメッシュ速度として次のものを選択できます。

- メッシュ速度を流体の速度にロックさせる。結果は**ラグランジュモデル**になります（FlexPDE はよじれたメッシュを元に戻すメカニズムを持っていないので、動きが激しい場合にこのモデルを使用することは推奨されません）。
- メッシュ速度を流体の速度とは別に保つ。これによってメッシュの integrity を維持しつつ、境界面の変形や流れの大筋の運動（bulk motion）への追従は可能になります。

- メッシュの動きが指定されなかった場合にはデフォルトのオイラーモデルとなります。

8.1 メッシュバランシング

移動するドメイン境界内において計算用メッシュを均等化させる便利な方法は座標系自体、またはメッシュ速度を拡散させることです。

一例としてサイズが $Rm = 0.5 + 0.25\cos(t)$ という形で振動する球をモデル化するものとします。

メッシュ座標系の拡散

まず代替座標系を定義します。

```
VARIABLES
  Phi
  Xm = MOVE(x)
  Ym = MOVE(y)
```

メッシュ座標変数に対する EQUATIONS としては内部の点をスムーズに分散させるべく単純な拡散方程式を使用します（実際の動きは境界条件によってドライブされることを期待）。

```
Div(Grad(Xm)) = 0
Div(Grad(Ym)) = 0
```

Blob 表面上のメッシュ座標系に対しては境界速度（R の時間微分から誘導）を直接適用することができます。

```
VELOCITY(Xm) = -0.25*sin(t)*x/r
VELOCITY(Ym) = -0.25*sin(t)*y/r
```

メッシュ速度の拡散

別のアプローチとして、代替座標系に加えてメッシュ速度変数を定義する方法もあります。

```
VARIABLES
  Phi
  Xm = MOVE(x)
  Ym = MOVE(y)
  Um
  Vm
```

メッシュ座標変数に対する EQUATIONS としては単に速度の関係式を規定します。

```
dt(Xm) = Um
dt(Ym) = Vm
```

メッシュ速度としては内部の点をスムーズに分散させるために単純な拡散方程式を使用します。

```
div(grad(Um)) = 0
div(grad(Vm)) = 0
```

Blob 上でのメッシュ速度に関する境界条件は上の場合と同一です。

```
VALUE(Um) = -0.25*sin(t)*x/r
VALUE(Vm) = -0.25*sin(t)*y/r
```

境界上のノードにおいて適用される有限要素法の等式はセルを通しての平均となるため、blob 境界上のメッシュ座標変数に対しては速度を明示する必要があります。

```
VELOCITY(Xm) = Um
VELOCITY(Ym) = Vm
```

8.2 脈動する球体

前記のメッシュバランシングの手法を用いて作成したスクリプトを以下に示します。

```
TITLE 'Heat flow around an Insulating blob'
VARIABLES
  Phi              { the temperature }
  Xm = MOVE(x)     { surrogate X }
  Ym = MOVE(y)     { surrogate Y }

DEFINITIONS
  K = 1            { default conductivity }
  R0 = 0.75        { initial blob radius }

EQUATIONS
  Phi:             Div(-k*grad(phi)) = 0
  Xm:              div(grad(Xm)) = 0
  Ym:              div(grad(Ym)) = 0
```

```
BOUNDARIES
  REGION 1 'box'
    START(-1,-1)
    VALUE(Phi)=0
    VELOCITY(Xm)=0 VELOCITY(Ym)=0
      LINE TO (1,-1)
    NATURAL(Phi)=0
      LINE TO (1,1)
    VALUE(Phi)=1
      LINE TO (-1,1)
    NATURAL(Phi)=0
      LINE TO CLOSE
  REGION 2    'blob'    { the embedded blob }
    k = 0.001
    START 'ring' (R,0)
    VELOCITY(Xm) = -0.25*sin(t)*x/r
    VELOCITY(Ym) = -0.25*sin(t)*y/r
    ARC(CENTER=0,0) ANGLE=360 TO CLOSE
TIME 0 TO 2*pi
PLOTS
  FOR T = pi/2 BY pi/2 TO 2*pi
    GRID(x,y)
    CONTOUR(Phi)
    VECTOR(-k*grad(Phi))
    ELEVATION(Phi) FROM (0,-1) to (0,1)
    ELEVATION(Normal(-k*grad(Phi))) ON 'ring'
END
```

Flash player がインストールされている環境であれば出力はアニメーションとして表示されます。

関連するフォームについては以下を参照ください。

- Samples | Moving_Mesh | 2D_Position_Blob.pde
- Samples | Moving_Mesh | 2D_Velocity_Blob.pde
- Samples | Moving_Mesh | 3D_Position_Blob.pde
- Samples | Moving_Mesh | 3D_Velocity_Blob.pde

第 9 章

メッシュ密度の制御

FlexPDE によって生成されるメッシュ中のセル密度を制御する方法にはいくつかあります。

暗黙の密度

生成されるメッシュのセル密度は境界セグメント上での節点の間隔に従います。非常に小さなセグメントの場合、その近傍には小さなセルが生成されることになります。

最大密度

グローバルコマンド

```
SELECT NGRID = <number>
```

によって最大セルサイズを制御することができます。最も大きな次元に対しては概ね NGRID 個のセルとなるよう、そしてより小さな次元に対しては対応するサイズでメッシュが生成されます (他の条件によってサイズは変わる可能性があります)。

明示的な密度制御

最初に生成されるメッシュのセル密度はパラメータ MESH_SPACING, MESH_DENSITY によって制御できます。MESH_SPACING は最大セルの大きさを制御するのに対し、MESH_DENSITY はその逆で単位長当りの最小セル数を制御します。メッシュジェネレータはセルサイズに影響を及ぼす多くの要請をチェックし、その中での最小値に基づきセルサイズを決定します。従って MESH_SPACING, MESH_DENSITY による制御はそれらが他の要請に比べて最小の値を与える場合にのみ効果を発揮するので、大きな間隔を要求しても実質的に無視されます。

MESH_SPACING, MESH_DENSITY による制御は定義パラメータとして、あるいは境界条件として使用できます。

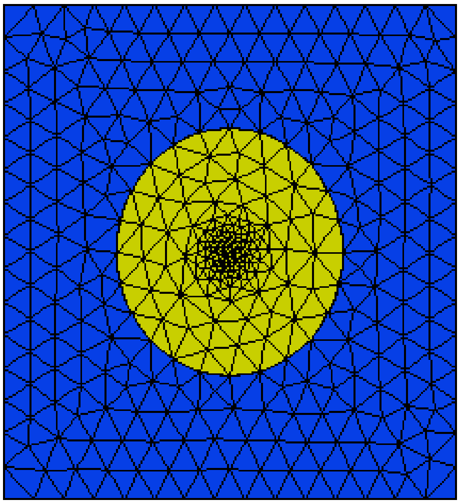
定義パラメータとしての用法の場合、それらは DEFINITONS セクションに置くこともできますが、リージョンごとの再定義セクション中で使用することもできます。これによってドメイン全体に対するメッシュ密度（2 次元/3 次元）の他に、リージョン対応のメッシュ密度（2 次元/3 次元）も制御できるようになります。

境界セグメントに沿った形でセル密度を制御したい場合には、MESH_SPACING, MESH_DENSITY による制御を境界条件に関する構文中で使用します。この場合、これらの制御は境界線、あるいは境界面上でのセル間隔を規定することになります。

MESH_SPACING, MESH_DENSITY で指定する値は空間座標変数の関数でも構いません。セクション 3.7“メッシュの生成”で説明した例において次のような指定を行うことも可能です。

```
REGION 2    'blob'    { the embedded 'blob' }
  MESH_DENSITY = 50*EXP(-50*(x^2+y^2))
  START(1/2,0)
  ARC(CENTER=0,0) ANGLE=360
```

この場合、最初に生成されるメッシュは次のようになります。



他の用例については”Samples | Misc | Mesh_Control | Mesh_Spacing.pde”と
”Samples | Misc | Mesh_Control | Mesh_Density.pde”を参照ください。

動的調整（Adaptive Refinement）

一旦最初のメッシュが生成されると、FlexPDE は誤差の推定を続け、目標の演算精度を達成できるよう必要に応じてメッシュの調整を行います。時間依存型の問題の場合には、変数の初期値に対しても動的調整機能が適用され、変数に急激な変化が生じたときにメッシュの調整が行われます。この動的調整プロセスによって生成されたセルはその後併合されることもありますが、明示的な密度制御によって初期設定されたセルは恒久的なものです。

Note: 動的調整のプロセスは既存メッシュ内部の離散点での種々の値の評価に依存しています。熱源等のソースの広がりが極端に小さかった場合（例えば薄膜状、あるいは点状の関数だった場合)、それらは見落とされる可能性があります。それらの効果に対しては、該当する位置に明示的にグリッドフィーチャを配置する等の措置が必要になります。`REGIONS` あるいは `FEATURES` の機能を使用してください。

他の制御機能については `FRONT`, `RESOLVE` ステートメントも参考にしてください。

第 10 章

データのエクスポート

FlexPDE には種々のデータエクスポート機能が用意されているので、そのデータを他のアプリケーションや視覚化ソフトで利用することが可能です。

EXPORT 修飾詞

最も簡単な方法はプロットコマンドに”EXPORT”または”PRINT”という修飾詞を付加することです。この場合、プロットデータは既定の様式でテキストファイル中に出力されるため、テーブル入力機能を使った他の FlexPDE プログラムでそれを利用することができます。ELEVATIONS または HISTORIES の場合、出力は時刻、あるいは x-, y-, z-座標値のリストにデータ値のリストが続く形となります（テーブル入力機能の項を参照）。2 次元プロットの場合、等間隔の矩形グリッドが構成され、データはテーブル入力形式で出力されます。

FORMAT ストリング

EXPORT 修飾詞によって生成されるテキストファイルの様式は FORMAT "string"という修飾詞を含めることによって制御できます。

この修飾詞が EXPORT や PRINT 修飾詞と共に使用された場合、ファイルにはグリッドの各点ごとに 1 行のテキスト行が挿入されます。行の内容は <string> によって指定されたものとなります。

- "#"を除くすべての文字はそのまま出力行にコピーされる。
- "#"はエスケープ文字として解釈され、種々のオプションが"#"に続く文字によって選択される。
  - #x, #y, #z 及び#t はデータ点の空間座標値、時間座標値を出力する。
  - #1 から#9 はプロット関数リスト中の対応する要素の値を出力する。
  - #b はタブ制御文字を出力する。

- #r は残りのフォーマットストリングをプロットリスト中のプロット関数ごとに繰り返すことを指示する。
- 繰返しのストリング内での#i はプロット関数リスト中での現行要素の値を出力する。

用例については”export_format”, ”export_history”を参照ください。

FORMAT 指定のエクスポートの場合、ファイルの出所に関する情報がヘッダとして出力されます。ヘッダは"{"と"}"によって仕切られます。2 次元のグリッドにおいては、テーブル中でドメイン外の点についても { } が付加され、外部の点であることが示されます。これらのコメント形式がインポート側のアプリケーションにとって不都合な場合には、データファイルを手で編集してください。

テーブル出力

テーブル形式のエクスポートファイルを生成するには TABLE プロットコマンドを使用することもできます。このコマンドは EXPORT 修飾詞付きの CONTOUR コマンドと等価ですが、グラフィック出力を伴わない点が異なります。テーブル出力の場合にも FORMAT "string"修飾詞を使用できます。

他の FlexPDE プログラムへのデータ転送

FlexPDE は有限要素メッシュ上で定義されたデータを直接転送する機能も備えています。
TRANSFER 出力機能は現状のメッシュ構成と指定されたデータ値を ASCII テキストファイル中に出力します。別の FlexPDE プログラムからは TRANSFER 入力機能を用いることによってこのファイルを読むことができます。転送されたデータは作成元の基底関数を持った出力メッシュ上に補間されます。TRANSFER 入力メッシュは必要な領域をカバーできる限り計算用のメッシュと同一である必要はありません。

TRANSFER ファイルのデータ形式は TECPLOT ファイル（下記参照）と同様です。しかし TRANSFER ファイルの場合には計算に際しての 2 次式、3 次式基底関数が保持されるのに対し、TECPLOT ファイルの場合には 1 次式の基底関数に変換されてしまいます。通常の ASCII テキストファイルなのでユーザアプリケーションとのデータ転送にも利用できます。TRANSFER ファイルのデータ形式についてはコマンドリファレンスマニュアルを参照ください。

視覚化ソフトへの出力

FlexPDE は third-party の視覚化ソフトに対するデータエクスポート機能をサポートしています。データのエクスポートはプロットコマンド中で指定しますが、その場合、CONTOUR 等のプロットタイプを指定する項でエクスポート様式を指定することになります。現状 CDF, TECPLOT, VTK の 3 種類に対応しています。

CDF

CDF(arg1 [,arg2,...]) という指定の場合、netCDF v3 の形式が選択されます。CDF は"common data format"の略であり、SlicerDicer (www.visualogic.com) 等、いくつかの製品によってサポートされています。CDF、及びそれをサポートするソフトウェア製品の一覧については www.unidata.ucar.edu/packages/netcdf をご参照ください。

CDF データは等間隔の矩形メッシュに制限されるため、不等間隔のドメインの場合には一部の矩形が欠落する場合があります。そのため材質境界の定義情報が抜け落ちたりすることがあります。細部の解像度を維持するためにはドメインに対する ZOOM 指定を検討ください。

CDF プロットステートメントに対しては ZOOM あるいは"ON SURFACE"という修飾詞を付加することができます。またメッシュ密度は POINTS 修飾詞によって制御できます。グリッドサイズをグローバルな形で制御するには"SELECT CDFGRID=n"というステートメントを使用してください。これによってグリッドサイズはすべて n にセットされます。デフォルトのグリッドサイズは 50 です。

引数の数に制限はなく、そのすべてが同一ファイル上にエクスポートされます。出力ファイルのデフォルトは"<problem>_<sequence>.cdf"ですが、FILE 修飾詞を使えば特定のファイル名称を設定することもできます。

TECPLOT

TECPLOT(arg1 [,arg2,...]) という指定の場合、TecPlot 形式での出力が選択されます。
TecPlot は有限要素法のデータ形式をサポートした視覚化ソフトであるため、FlexPDE で定義された材質境界が保持されます。このため ZOOM とか POINTS の制御は特に使用する必要はありません。計算メッシュ全体が材質番号でグルーピングされた形でエクスポートされます。TecPlot ではこれらのグループを有効にするかどうかが選択できます。引数の数に制限はなく、そのすべてが同一ファイル上にエクスポートされます。出力ファイルのデフォルト

は”<problem>_<sequence>.dat”ですが、FILE 修飾詞を使えば特定のファイル名称を設定することもできます。

TecPlot に関する情報については www.amtec.com をご参照ください。

VTK

VTK(arg1 [,arg2,...]) という指定の場合、Visual Tool Kit 形式での出力が選択されます。VTK はフリーの視覚化ソフトであり、多くの視覚化ソフトの基盤として採用され始めています。このファイル形式は VisIt (www.llnl.gov/visit) のような VTK をベースとしていない一部のソフトでも読むことができます。この形式でも有限要素法のメッシュ構造は保持されるため、FlexPDE で定義された材質境界も保持されます。このため ZOOM とか POINTS の制御は特に使用する必要はありません。計算メッシュ全体がエクスポートされます。ただし視覚化ソフトの機能の中には FlexPDE で制御できないものもあります。引数の数に制限はなく、そのすべてが同一ファイル上にエクスポートされます。出力ファイルのデフォルトは”<problem>_<sequence>.vtk”ですが、FILE 修飾詞を使えば特定のファイル名称を設定することもできます。

VTK 形式では 2 次式の基底関数をサポートしていますが、3 次式には対応していません。3 次式の基底関数に基づく計算結果をエクスポートする場合には VTKLIN を使用してください。

VTKLIN(arg1 [,arg2,...]) によって生成される VTK 形式のファイルでは、FlexPDE の計算用セルが 1 次式の有限要素法セルに変換されています。

VTK に関する情報については public.kitware.com/VTK/をご参照ください。

用例

用例については以下を参照ください。

- Samples | Misc | Import-Export | Export.pde
- Samples | Misc | Import-Export | Export_Format.pde
- Samples | Misc | Import-Export | Export_History.pde
- Samples | Misc | Import-Export | Transfer_Out.pde
- Samples | Misc | Import-Export | Transfer_In.pde
- Samples | Misc | Import-Export | Table.pde

Note: 他社製品への言及があったからといってそれは PDE Solutions Inc. のエンドースメントを意味するものではありません。

第 11 章

非線形問題への対応

FlexPDE は問題が非線形であるかどうかを自動的に認識し、解法戦略をそれに適した形に変更します。FlexPDE で使用される解法は**修正ニュートン-ラフソン反復法**（modified Newton-Raphson iteration procedure）と呼ばれるものです。これは下降法であり、エネルギー汎関数の勾配を下ることによってその最小値を求めようとするものです（最小点では汎関数の勾配は 0 となります）。単純な拡散型問題の場合がその一例ですが、汎関数が 2 次関数に近いものであった場合には、この探索法は 2 次関数的に収束します（反復を繰り返すごとに相対誤差は 2 乗の形で小さくなります）。FlexPDE で実装されているデフォルトの探索法は通常ユーザの助けがなくても解を見つけることができます。しかし非線形性が強いときには、FlexPDE が正しい解に向かうようユーザのガイドが必要となる場合があります。そのためのテクニックにはいくつかあります。

時間依存型問題

非線形の時間依存型問題の場合、いかなる非線形性もタイムステップコントローラによって感知され、そのタイムステップの調整が初期条件からの系の正確な展開を保証するとの仮定に基づき、それぞれのタイムステップではニュートン法のステップを 1 つしか進めないというのがデフォルトの設定です。この場合、解の微分値はタイムステップの開始時点でのみ計算され、アップデートされることはありません。

より難しい局面にも頑強に対応できるような（従って演算時間を要する）選択肢がいくつか用意されています。キーとなるのは `PREFER_STABILITY` という選択肢ですが、これを指定した場合には各タイムステップにおいてニュートン法の反復が最大 3 回まで行われるようになります。その場合、微分値は反復ごとに再計算されます。またそれは誤差の重み付けスキームを変更し、より局所的な変動に重きを置くようにします。また `PREFER_STABILITY` は `NRUPDATE` と `TNORM` の値をリセットします。

定常問題

非線形の定常問題の場合には状況はさらに複雑化します。系がユニークな解を持つという保証がないばかりか、仮にそうだとしても FlexPDE がそれを見出せるという保証がないからです。

(1) 良い初期値からスタートする

真の解に近い初期値を与えることは解探索にとってこの上ない助けとなります。ただしその初期値は境界条件に合致したものとしてください。そうでない場合にはあらぬ方向へ求解が流れることとなり解探索に支障を来たします。

(2) ステージング機能を使って非線形項をアクティベートする

FlexPDE のステージング機能を使えば非線形項の強度を徐々に増やしてゆくことができます。まずは線形に近い系からスタートし、FlexPDE が境界条件を満たす解を見つけられるように仕向けます。それを起点にしてより非線形性の強い系に対する解探索を行わせます。ステージングをうまく使うと非常にたちの悪い問題の解にも近づいて行くことが可能です。

(3) 人為的な拡散項を利用して解を安定化させる

信号処理において不連続な信号をフーリエ成分から再構成しようとすると Gibbs 現象が発生します。この現象の特徴は顕著なもので、復元された信号中に上下に振動する数多くのピークが混入する結果となります。メッシュ密度が十分でない状態で急激な変位をモデル化しようとすると、有限要素法においても同様の現象が発生します。信号処理の場合には窓関数（window function）を用いることによって信号を平滑化することができます。これは実質的に信号中の高周波成分を取り除くローパスフィルタとして機能します。偏微分方程式の問題においては拡散演算子 $\mathrm{div}(\mathrm{grad}(u))$ が同様のローパスフィルタの役割を果たします。技術的な詳細についてはテクニカルノート”Smoothing Operators in PDE”を参照ください。要は PDE 中に $eps \cdot \mathrm{div}(\mathrm{grad}(u))$ という項を導入することによって空間的広がり D を持った振動を平滑化することができます。ここに $eps = D^2/\pi^2$（定常問題)、または $eps = 2Dc/\pi$（時間依存型問題）です（c は信号伝播速度)。ただし方程式の他の部分と次元上の整合性を維持するため、必要に応じて項のスケールは調整してください。このような項の使用は、解の空間周波数成分を有限要素メッシュ上でうまく扱える範囲に抑える効果を持ちます。

(4) CHANGELIM による刻みの制御

セレクタ `CHANGELIM` を選択することによって、ニュートン-ラフソン法の各ステップにおいて節点の値が変化する量に制限を加えることができます。1 次元のニュートン法の場合と同様、近似解が汎関数の極大点の近傍にあった場合には、次の近似解への刻み（ステップ）が非常に大きな値と

なってしまうことがあります。FlexPDE は節点における変化の量を CHANGELIM*（変数の平均値）以下に抑えるよう制御します。CHANGELIM のデフォルト値は 0.5 ですが、この値の場合には、初期値（あるいは途中の近似解）の位置が真の解の位置から遠いときに解探索に大きな乱れ（wild excursions）を生じることがあります。CHANGELIM の値を 0.1 とか、場合によっては 0.01 にまで小さくしてみてください。適切な初期値が与えられた場合には CHANGELIM = 0.01 でも驚くほど短時間に解に収束することができます。CHANGELIM はそれぞれの局所値ではなく RMS の平均値に乗ぜられるため、その効果は解に到達した時点で消え、最後の収束は依然 2 次関数的に推移します。

(5) 負の値に注意

FlexPDE は解を近似するのに区間ごとに定義された多項式（piecewise polynomials）を使用します。解が単一セル上で急激に変動するような場合には、早い段階ではほぼ間違いなく under-shoot が発生します。このため、たとえ変数値は正に留まるはずという仮定があったとしてもそうはなりません。負の値となったときに方程式が成立しなくなるような場合には、該当変数を対数化した上で方程式を組みなおしてみてください。この場合、対数値はたとえ負になることがあっても、実際の変数値は正に留まります。

(6) 時間依存型問題への変形

定常問題はどのようなものであっても時間依存型問題の無限時間後の状態と考えることができます。定常問題の解からの乖離を小さくする方向に誘導する時間微分項を偏微分方程式に追加してみてください。（良い例が時間依存型の拡散方程式 DIV(K*GRAD(U)) = DT(U) です。div が負値であることは解が極大の状態にあることを意味するので、その値は減少する方向に推移します。）この場合、時間は架空の変数であるわけですが、このように時間依存型の方程式にすることによってタイムステップコントローラを利用した形での解の誘導が可能になります。

第 12 章

技術サポート

ライトストーン技術サポート

FlexPDE に関する操作上のご質問は (株) ライトストーンまでお気軽にお問い合わせ下さい。

E-mail:	tech@lightstone.co.jp
Tel:	03-5600-7202
Fax:	03-5600-6671

索引

ALE モデル, 85
ARC, 12
AREA_INTEGRAL, 26, 31
AUTOSTAGE, 31

BINTEGRAL, 26
BOUNDARIES, 6, 12

CDF, 95
CHANGELIM, 45, 98
CLOSE, 12
compartments, 62, 63, 65
CONTACT, 50, 53
CONTOUR, 19
COORDINATES, 31, 57, 60

DEFINITIONS, 6, 17

ELEVATION, 19
END, 6
EQUATIONS, 6, 11
ERRLIM, 25, 34
EXPORT, 93
EXTRUSION, 60
extrusion, 59

FORMAT, 93

GRID, 68

HISTORY, 29, 34

INITIAL VALUES, 45
INTEGRAL, 26

JUMP, 50, 53

LAMBDA, 38
LAYER, 60
LIMITED REGION, 66
LINE, 12
LINE_INTEGRAL, 26

MESH_DENSITY, 89
MESH_SPACING, 15, 89
MODES, 38
MONITORS, 6, 19

NATURAL, 8, 18
NATURAL 型境界条件, 8, 18

ON, 67, 82

PDE ソルバ, 3
PLOTS, 6, 19
POINT LOAD, 57
POINT VALUE, 57
PREFER_STABILITY, 97
PRINT, 93

REGION, 12
REPORT, 27

SELECT, 6
SPLINE, 12
STAGED, 29
STAGES, 30
START, 12
SUMMARY, 19, 28, 41
SURF_INTEGRAL, 31, 78
SURFACE, 19, 60

TABLE, 94
TECPLOT, 95
THRESHOLD, 34
TITLE, 6
TRANSFER, 94

VALUE, 8, 18
VALUE 型境界条件, 8, 18
VARIABLES, 6, 11
VECTOR, 19

VOL_INTEGRAL, 26, 31, 78
VTK, 96

XCYLINDER, 31

YCYLINDER, 31

エクスポート, 93
エクスポートモジュール, 6
円柱座標系, 31
　　積分計算, 31

オイラー形式, 85
大文字/小文字, 11
押出し, 59
　　設定, 60

回転定理, 48

ガイドライン, 10

記法, 11
境界条件, 8, 18
　　3 次元, 71

区画, 62, 63
　　空の, 65

グラフィックス出力, 19
グラフィックス出力モジュール, 6
グリッド
　　計算用, 68
　　プロット用, 82

計算精度, 25

限定リージョン, 66

コメント, 7
固有関数, 38
固有値, 38
固有値サマリ, 41
固有値問題, 4

誤差許容値, 25

サマリプロット, 28

材質特性, 63
材質パラメータ, 17

自然境界条件, 18, 46, 47
自然言語, 3
初期推定値, 45
初期値, 45

時間依存型問題, 4
時間依存性, 34, 37
磁場方程式, 48

数式アナライザ, 5
スクリプト, 6
スクリプトエディタ, 5
ステージング, 29, 45, 98

精度, 25
積分
　　3 次元, 78
積分計算, 26
接触抵抗, 50
切断面, 67
線形, 5
線積分, 26

素材特性, 5

体積積分, 26, 78
タイムステップ制御, 34
タイムステップ調整機能, 5
対流方程式, 48

代替変数, 85

定常状態問題, 4

データ転送, 94

統合環境, 3

ドメイン, 12

ニュートン-ラフソン法, 5, 45, 97

熱拡散方程式, 48

発散定理, 46, 48

パラメータ, 17

ヒストリ, 29
非線形, 5
非線形問題, 43, 45, 97

微分表記, 11

不連続変数, 50, 53

部分積分, 46, 47
分割面, 60

プロット制御, 82

変分法, 46

メッシュ, 15
メッシュ移動, 85
メッシュ生成モジュール, 5
メッシュ調整機能, 5, 91
メッシュバランシング, 86
メッシュ分割, 25
メッシュ密度, 89
面積分, 26, 78

モード解析, 38
問題設定のガイドライン, 10
問題の設定, 9

有限要素法モデルビルダ, 3
有限要素法数値演算モジュール, 5
有限要素メッシュ, 15

ラグランジュモデル, 85

リージョン, 5, 12, 61

レイヤ, 60
レイヤ境界
　　変形, 75
レイヤリング, 62
連結性解除, 53