

## GB005: 1次元/2次元過渡問題

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。

ここでは熱伝導方程式に時間依存の項が含まれるケース（過渡問題、transient problems）について考察します。時間依存型の問題の場合には GB004 で扱った方程式

$$\nabla \cdot \mathbf{f} = h(x, y) \quad (1)$$

に対し時間微分項が加わる形となります。

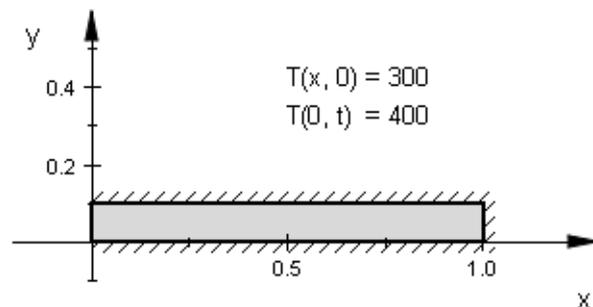
$$\nabla \cdot \mathbf{f} + \rho c_p \frac{\partial T}{\partial t} = h \quad (2)$$

ここに  $\rho$  は物質の密度を、 $c_p$  は比熱（specific heat capacity）を表すパラメータです。数式 (2) は体積要素中におけるエネルギー収支を表現したものとと言えます。なお、この数式において  $T$  は絶対温度を、 $t$  は時間を表す変数である点に注意してください。

以下においてはまず 1 次元の問題を扱います。すなわち数式モデル上、空間変数は  $x$  のみであるわけですが、FlexPDE 上は 2 次元の問題として表現されている点に注意してください。

## 1 Case1 - 鉄棒中の熱伝導 [1]

本来は 1 次元の熱伝導なので  $y$  軸方向の幅は 0 として考えるべきかも知れませんが、FlexPDE の適性を考慮し、ここでは右図のような 2 次元の広がりを持った鉄棒をモデル化することにします。鉄棒の温度を  $T(x, t)$  で表すことにしたとき、その初期状態  $T(x, 0)$  は  $300^\circ\text{K}$  であるとし、この鉄棒の左端に  $t = 0$  以降熱を加え、 $400^\circ\text{K}$  に保ったとしたとき、鉄棒内の温度分布が時間と共にどう変化するかを FlexPDE で計算してみます。



実は鉄棒が無限に長い場合には解析解が知られていて

$$T(x, t) = (T_i - T_0) \operatorname{erf} \left( \frac{x}{2\sqrt{\lambda t / \rho c_p}} \right) + T_0 \quad (3)$$

という数式で  $T$  が表現されます。ここに  $T_i$  は初期状態における温度、 $T_0$  は  $x = 0$  における温度、 $\lambda$  は熱伝導率を表すパラメータです。また  $\operatorname{erf}$  は誤差関数

$$\operatorname{erf}(y) = \frac{2}{\sqrt{\pi}} \int_0^y \exp(-\eta^2) d\eta$$

を意味し、FlexPDE 中にも用意されています。ここでは FlexPDE による数値解がこの理論値とどの程度的一致を見せるかについても検証してみます。

### 1.1 Problem descriptor [ transient01a.pde ]

まずタイトルを設定します。

#### TITLE

```
'Bar with a Temperature Step' { transient01a.pde }
```

次に演算精度に関するセレクトをセットします。デフォルトは 0.002 なのですが、ここでは精度を多少高めに設定します。

#### SELECT

```
Errlim = 3e-4
```

従属変数を定義します。

#### VARIABLES

```
temp { Temperature }
```

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。解析対象のドメイン上に熱源がないため上記数式中における  $h$  は 0、また鉄の電導率は 82 に、 $\rho \cdot c_p$  の値は  $7.87e3 \cdot 449$  に設定されています。一方、temp\_ex は厳密解を表す数式です。

#### DEFINITIONS

```
Lx = 1.0 Ly = 0.1
heat = 0 { Heat source }
k = 82 rcp = 7.87e3*449 { Iron parameters }
temp_i = 300 temp_0 = 400 { Initial and boundary temperatures }
fluxd_x = -k*dx(temp) fluxd_y = -k*dy(temp)
fluxd = vector(fluxd_x, fluxd_y) fluxd_m = magnitude(fluxd)
temp_ex = (temp_i - temp_0)*erf[x/(2*sqrt(k*t/rcp))] + temp_0
```

ここで時間依存型問題の場合に必要な初期状態の規定を行っておきます。

#### INITIAL VALUES

```
temp = tempi
```

次に方程式を上記 (2) の形で定義します。DEFINITIONS セクション中における fluxd の定義と合せると温度 temp に関する 2 階の偏微分方程式が導かれることがわかります。

#### EQUATIONS

```
div(fluxd) + rcp*dt(temp) = heat
```

今度は境界の形状と境界条件を設定します。鉄棒の左端については Value 文で定数値を指定しますが、その他の境界については Natural 文にて断熱型の境界を設定しています。

#### BOUNDARIES

```
Region 1
Start(0,0)
Natural(temp) = 0 Line to (Lx, 0) to (Lx, Ly) to (0, Ly)
Value(temp) = temp0 Line to Close
```

これも時間依存型問題に固有の指定項ですが、TIME セクションを用いて解析対象の時間帯を指定します。

#### TIME

```
from 0 to 3000
```

最後に出力すべき情報を指定します。ここでは鉄棒の中心軸上における温度 temp の値を、その理論値 temp\_ex と合せて elevation プロットの形でプロットするわけですが、それを  $t = 100$  から 3000 まで 100 秒単位に出力すべく、for ... by ... to クローズが使用されている点にご注意ください。

#### PLOTS

```
for t = 100 by 100 to 3000
Elevation(temp, temp_ex) from (0, Ly/2) to (Lx, Ly/2)
```

#### END

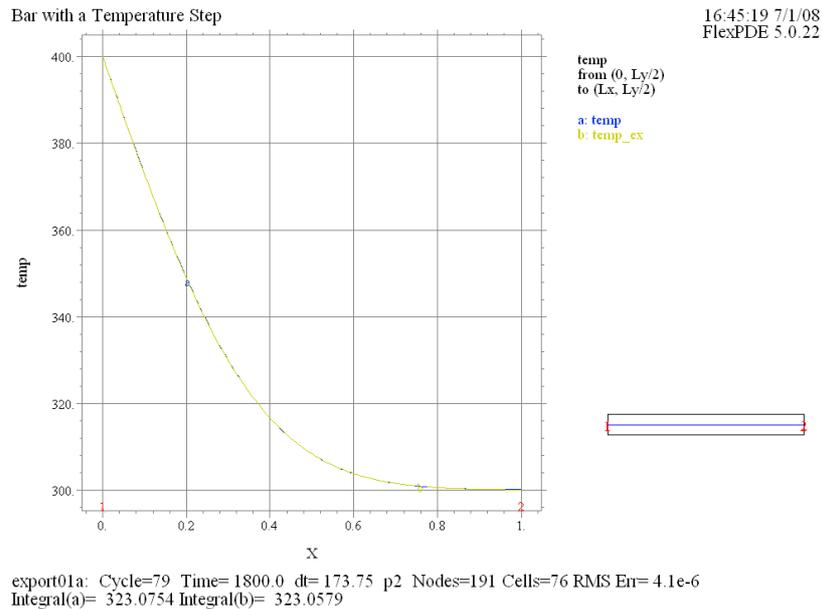
## 1.2 実行結果

上記スクリプトをそのまま FlexPDE に実行させるとすべてのプロットが一気に生成され、 $t = 3000$  に対応したプロットが表示された状態で処理が終了します。最初からすべてのプロットを確認するには「File」メニュー：「View File」と操作し、該当する .pg5 ファイルを選択してください。「View」メニュー：「Next」あるいは「Back」と操作することにより手動で前後に進めることができます。一方、「View」メニュー：「Movie」と操作すれば画面切替えは一定時間間隔で自動的に行われるようになります。

$t = 0$  における温度分布には不連続性が含まれており ( $x = 0$  においては  $400^\circ$ 、その点以外では  $300^\circ$ )、その扱いには難しい面があるのですが、FlexPDE はそれを無難に処理し、厳密解と良く一致したプロットを出力してきています。ここでは  $t = 1800$  と  $t = 3000$  のときのプロットを掲載しておくことにします。

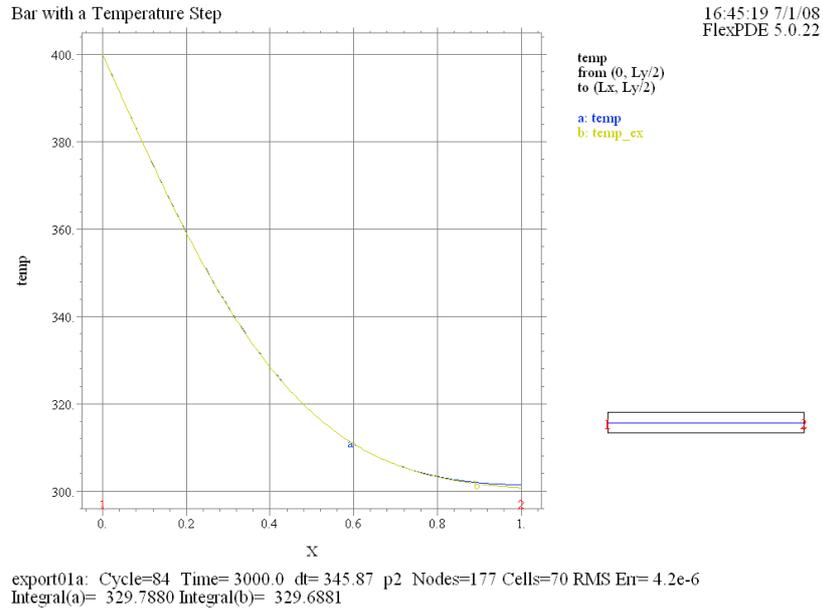
(1) Elevation(temp, temp\_ex) from (0, Ly/2) to (Lx, Ly/2) at  $t = 1800$

$t = 1800$  の時点における elevation プロットです。temp と temp\_ex のグラフが異なる色で 2 本描かれているわけですが、実質的に重なってしまっているため 1 本の曲線のように見えます。プロット下部に示されている積分値についても良い一致が見られます。



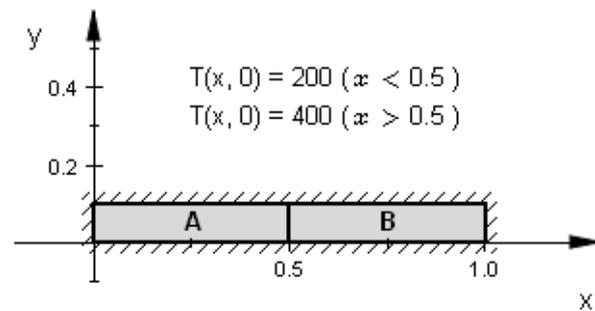
(2) Elevation(temp, temp\_ex) from (0, Ly/2) to (Lx, Ly/2) at  $t = 3000$

$t$  が 3000 に近づくとつれ、鉄棒右端における温度分布には理論値との乖離が見られるようになります。これは数値演算に伴う誤差に起因するものではありません。右端における境界条件の違いによるものです。厳密解は鉄棒が無限遠まで続いているとしたときに成り立つものである点に注意してください。FlexPDE が計算したのは有限長の鉄棒中における熱伝導です。

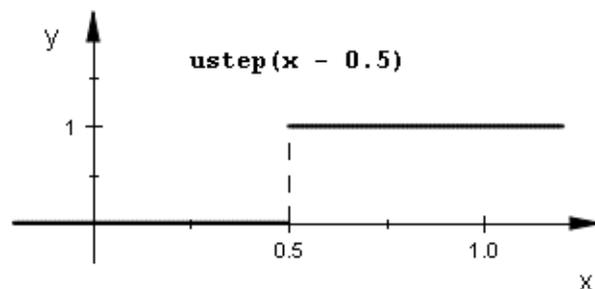


## 2 Case2 - 鉄棒中の熱伝導 [2]

今度はそれぞれ  $200^{\circ}\text{K}$ 、 $400^{\circ}\text{K}$  に保った鉄棒 A, B を時刻  $t = 0$  で接触させたときの温度分布の変化を調べてみることにします。ただし鉄棒の周囲はすべて熱的に遮断されているものとします。この例でキーとなるのは初期状態の設定に際し `ustep` というステップ関数 (unit step function、Heaviside function) を用いる点です。



例えば  $\text{ustep}(x - 0.5)$  という関数は  $x < 0.5$  のとき 0、 $x > 0.5$  のときに 1 という値を取る右図のような関数となります。



## 2.1 Problem descriptor [ transient01b.pde ]

基本形は transient01a.pde と変わりません。

```

TITLE
  'Temperature Step in Bar'    { transient01b.pde }

SELECT
  Errlim = 1e-4

VARIABLES
  temp                { Temperature }

DEFINITIONS
  Lx = 1.0  Ly = 0.1
  heat = 0                { Heat source }
  k = 82  rcp = 7.87e3*449 { Iron parameters }
  fluxd_x = -k*dx(temp)  fluxd_y = -k*dy(temp)
  fluxd = vector(fluxd_x, fluxd_y)  fluxd_m = magnitude(fluxd)

```

初期状態をステップ関数を用いて設定します。

```

INITIAL VALUES
  temp = 200 + 200*ustep(x - Lx/2) { Unit step function }

EQUATIONS
  div(fluxd) + rcp*dt(temp) = heat

```

境界条件としてはすべて断熱型を使用します。境界条件がすべて Natural 型だった場合には関数値がユニークに決まらないことがあります(GB001 参照) 今回は初期値を指定しているため問題は起こりません。

```

BOUNDARIES
  Region 1
  Start(0,0)
  Natural(temp) = 0
  Line to (Lx, 0) to (Lx, Ly) to (0, Ly) to Close

TIME
  from 0 to 100000

```

プロットを出力するタイミングは対数スケールを意識して設定します。

#### PLOTS

```
for t = 10, 30, 100, 300, 1000, 3000, 10000, 30000, 100000
Elevation(temp) from (0, Ly/2) to (Lx, Ly/2)
Contour(temp) painted
```

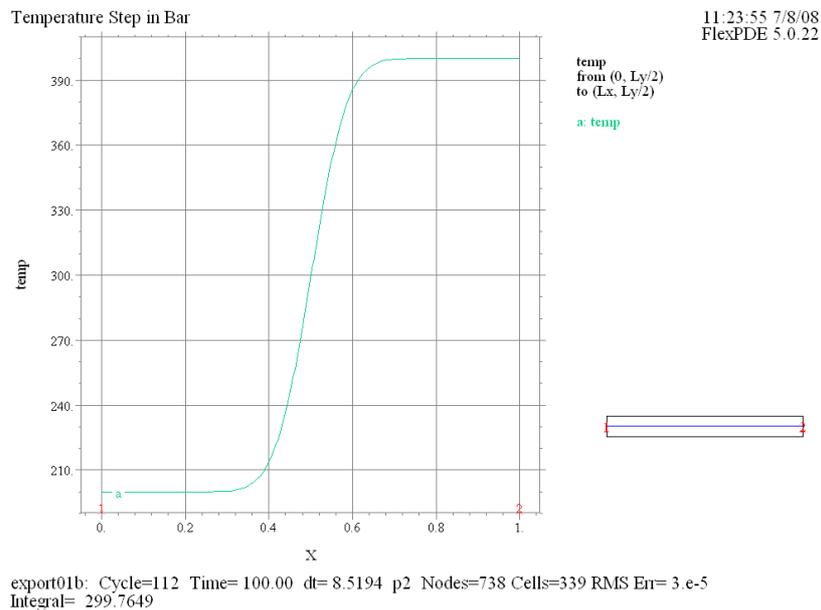
END

## 2.2 実行結果

今回、プロットを出力するタイミングは `for ... by ... to` クローズを使用せず、すべてを明示的な形で設定しています。それぞれの時点における Elevation プロット、Contour プロットが計 18 枚出力されるわけですが、ここではその一部を示すに留めます。

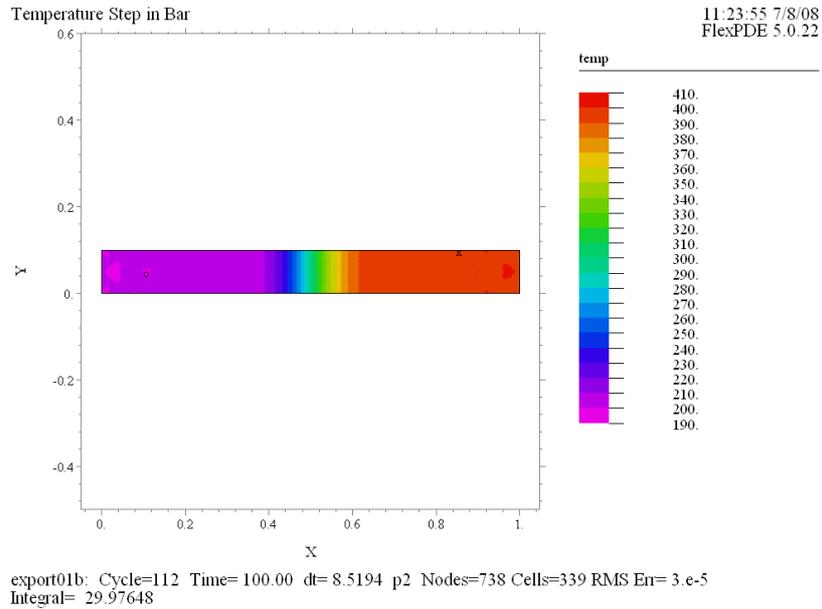
(1) Elevation(temp) from (0, Ly/2) to (Lx, Ly/2) at t = 100

2 本の鉄棒を接触させてから 100 秒経過した時点における鉄棒軸上での温度分布をプロットしたものです。鉄棒の左側の部分と右側の部分とは  $200^\circ$  の温度差があります。



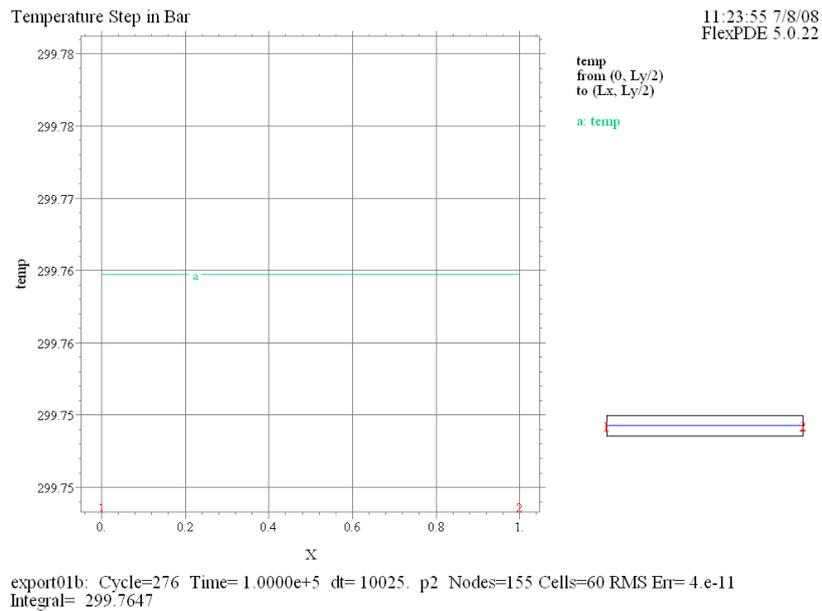
(2) Contour(temp) painted at  $t = 100$

$t = 100$  の時点における温度分布を等温線図の形でプロットしたものです。



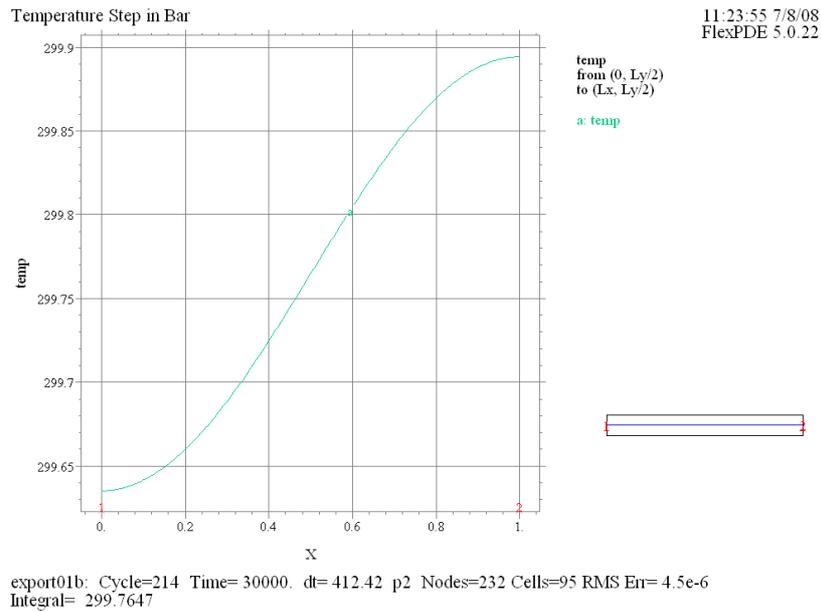
(3) Elevation(temp) from  $(0, Ly/2)$  to  $(Lx, Ly/2)$  at  $t = 100000$

$t = 100000$  における温度分布のグラフです。十分な時間の経過後であるため、一様な温度分布となっています。



(4) Elevation(temp) from  $(0, Ly/2)$  to  $(Lx, Ly/2)$  at  $t = 30000$

(3) より一つ前の時点である  $t = 30000$  における温度分布を示すグラフです。Movie 再生すると最後のフレームである (3) とその一つ前のフレーム (4) の間で、グラフ形状に突然の変化が生じるように見えますが、表示されている temp のスケールに注意してください。例えば下図の場合、299.62 から 299.91 の範囲だけが拡大されて表示されているため S 字状の曲線に見えますが、これを (1) と同じ 200 から 400 のスケールでプロットすれば全くの直線に見えるはずで

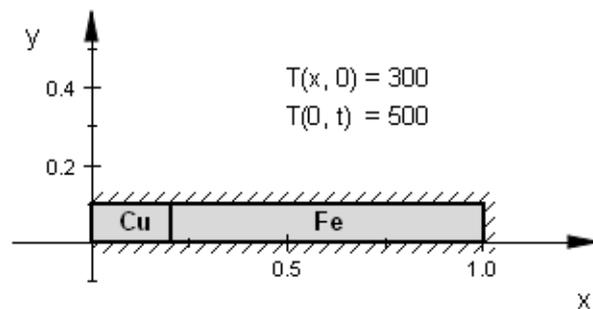


なお、プロット下部に表示されている温度の積分値（この場合は線積分）が  $t$  の値によらずほぼ一定値を保っている点に注意してください。これはエネルギー保存則に基づく帰結と言えます。

### 3 Case3 - 鉄銅溶接棒中の熱伝導

今度は右図に示されるような鉄と銅が溶接されているケースについて熱伝導を調べてみます。初期条件、境界条件は Case1 の場合とほとんど変わりませんが、唯一左端の温度が  $500^{\circ}\text{K}$  となっている点が異なります。

また、グラフ形状の時間的変化を浮き彫りにするために History プロットの機能を使用します。



### 3.1 Problem descriptor [ transient01c.pde ]

基本形は transient01a.pde と変わりません。

```

TITLE
  'Transient Conduction in Cu/Fe'    { transient01c.pde }

SELECT
  Errlim = 1e-4

VARIABLES
  temp                                { Temperature }

DEFINITIONS
  Lx = 1.0  Ly = 0.1
  heat = 0                                { Heat source }
  k rcp                                { Iron parameters }
  fluxd_x = -k*dx(temp)  fluxd_y = -k*dy(temp)
  fluxd = vector(fluxd_x, fluxd_y)  fluxd_m = magnitude(fluxd)
  tempi = 300  temp0 = 500

INITIAL VALUES
  temp = tempi

EQUATIONS
  div(fluxd) + rcp*dt(temp) = heat

```

以下の BOUNDARIES セクションにおいて、Region 1 ではドメイン全体としての形状とデフォルトの材質特性を設定、Region 'copper' では銅の部分の範囲と固有パラメータの設定を行います。

```

BOUNDARIES
  Region 1  k = 82  rcp = 7.87e3*449          { Fe, default }
  Start(0,0)
  Natural(temp) = 0  Line to (Lx, 0) to (Lx, Ly) to (0, Ly)
  Value(temp) = temp0  Line to Close
  Region 'copper'  k = 400  rcp = 8.96e3*385  { Cu }
  Start(0,0)
  Line to (0.2*Lx, 0) to (0.2*Lx, Ly) to (0, Ly) to Close

TIME
  from 0 to 10000

```

プロットを出力するタイミングは対数スケールを意識して設定します。金属棒軸上における温度分布、及び熱流束密度ベクトルの  $x$  成分の値を各時点ごとにプロットします。

#### PLOTS

```
for t = 10, 30, 100, 300, 1000, 3000, 10000
  Elevation(temp) from (0, Ly/2) to (Lx, Ly/2)
  Elevation(fluxd_x) from (0, Ly/2) to (Lx, Ly/2)
```

次の指定は金属棒軸上の 5 点において温度が時間的にどう変化して行くかを History プロットの形で出力させるものです。

#### HISTORIES

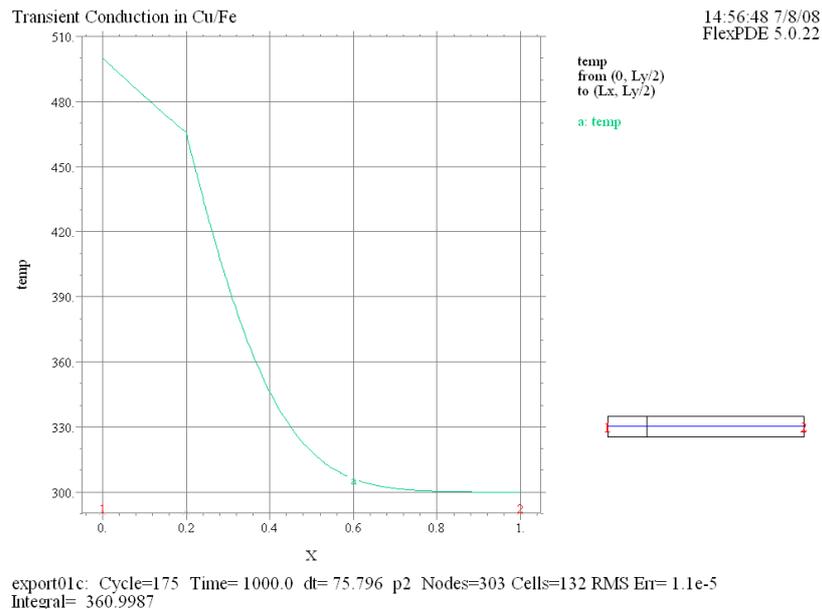
```
History(temp) at (0.1*Lx, Ly/2) at (0.2*Lx, Ly/2)
               at (0.3*Lx, Ly/2) at (0.4*Lx, Ly/2)
               at (0.5*Lx, Ly/2)
```

END

## 3.2 実行結果

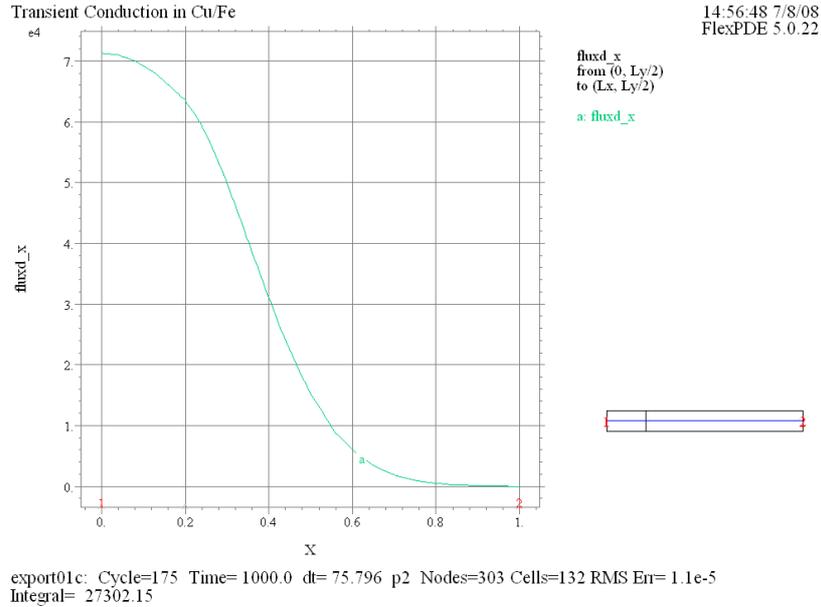
(1) Elevation(temp) from (0, Ly/2) to (Lx, Ly/2) at t = 1000

$t = 1000$  の時点における金属棒軸上での温度分布を示したものです。銅と鉄の間での熱伝導率の違いにより、銅と鉄の境界面  $x = 0.2$  において曲線の傾きが急に変化しています。



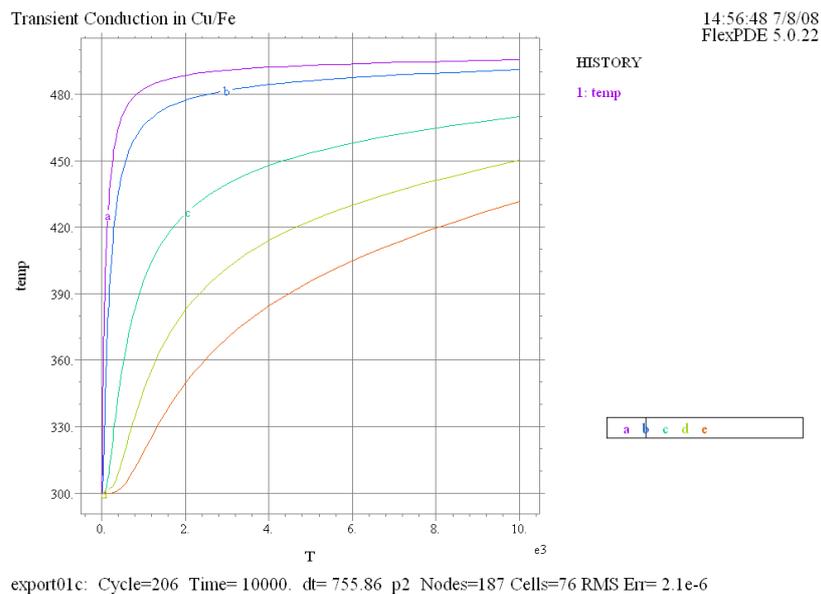
(2) Elevation(fluxd\_x) from (0, Ly/2) to (Lx, Ly/2) at t = 1000

これに対し、熱流束密度ベクトルの  $x$  成分の値  $f_x$  は銅と鉄の境界面で連続的に変化している点にご注意ください。これは理論からの要請でもあります。



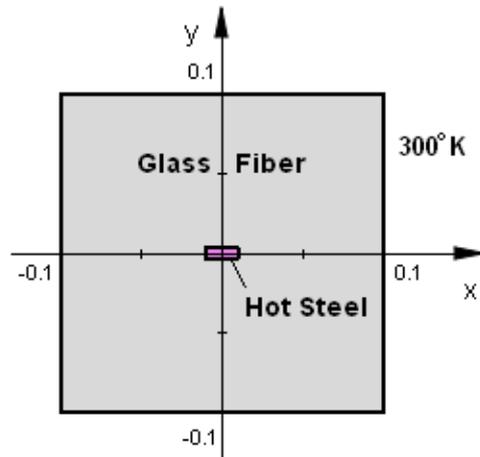
(3) History(temp) at t = 10000

最終時刻  $t = 10000$  における History プロットのグラフです。プロット右側のインサート部に a, b, c, d, e の位置が示されています。グラフにはそれぞれの点において温度（縦軸）が時間（横軸）と共にどう変化したかが示されています。



## 4 Case4 - 鉄棒からの2次元熱拡散

ここまでは本質的に1次元の問題を便宜的に2次元のモデルとして扱ってきたわけですが、今回は2次元における熱伝導の問題を正面から扱うことにします。右図中央にあるのは断面が長方形の鉄製の棒で、電磁誘導によって熱せられるものとし、周囲は熱伝導率の低いガラス繊維によって囲まれており、その外側を囲う正方形の箱は一定温度  $300^{\circ}\text{K}$  に保たれているものと仮定します。



以下に示すスクリプトの中では従属変数 `temp` に対し `threshold = 0.1` という指定が行われている点に注意してください。これより小さな温度変動に対しては興味がないことを示すもので、計算時間の短縮に効果があります。

### 4.1 Problem descriptor [ transient01d.pde ]

基本形は `transient01a.pde` と変わりません。

```
TITLE
  'Transient Heating of Steel Bar'    { transient01d.pde }

SELECT
  Errlim = 1e-3
```

計算効率を考慮し、従属変数 `temp` に関する  $0.1$  以下の変動には関心がないことを宣言します。

```
VARIABLES
  temp(threshold = 0.1)    { Absolute tolerance }
```

全体のエンクロージャのサイズは  $20\text{cm} \times 20\text{cm}$ 、鉄棒断面のサイズは  $2\text{cm} \times 2/3\text{cm}$  とします。また熱源のパワー密度を  $1e7$  に設定します。鉄棒から発せられる単位時間当りの熱量はこれに断面積をかけた値となります。

## DEFINITIONS

```

L = 100e-3  L0 = 10e-3
heat                { Heat source }
k  rcp              { Heat parameters }
tempi = 300         { Initial and boundary temperatures }
fluxd_x = -k*dx(temp)  fluxd_y = -k*dy(temp)
fluxd = vector(fluxd_x, fluxd_y)  fluxd_m = magnitude(fluxd)
power_d = 1e7       { Power density }
power = (2*L0)*(2*L0/3)*power_d { Power from bar }

```

温度の初期値は 300°K とします。

## INITIAL VALUES

```
temp = tempi
```

今回の場合、鉄棒部分では右辺は 0 とはなりません。

## EQUATIONS

```
div(fluxd) + rcp*dt(temp) = heat
```

以下の BOUNDARIES セクションにおいて、Region 1 ではドメイン全体としての形状とデフォルトの材質特性を設定、Region 'steel' では鉄棒部分の範囲と固有パラメータの設定を行います。

## BOUNDARIES

```

Region 1  heat = 0  k = 0.5  rcp = 2e6  { Glass fiber, default }
  Start(-L,-L)
    Value(temp) = tempi  Line to (L, -L) to (L, L)
                        to (-L, L) to Close
Region 'steel'  heat = power_d  k = 45  rcp = 3.5e6
                { Stainless steel }
  Start(-L0, -L0/3)  Line to (L0, -L0/3) to (L0, L0/3)
                    to (-L0, L0/3) to Close

```

## TIME

```
from 0 to 1e6
```

プロットを出力するタイミングは対数スケールを意識して設定します。等温線図については鉄棒周囲のみを拡大した図も作成します。

## PLOTS

```

for t = 100, 300, 1e3, 3e3, 1e4, 3e4, 1e5, 3e5, 1e6
Contour(temp)
Contour((temp - temp_i)*rcp) Report(power*t)
Contour(temp) zoom(-5*L0, -5*L0, 10*L0, 10*L0)

```

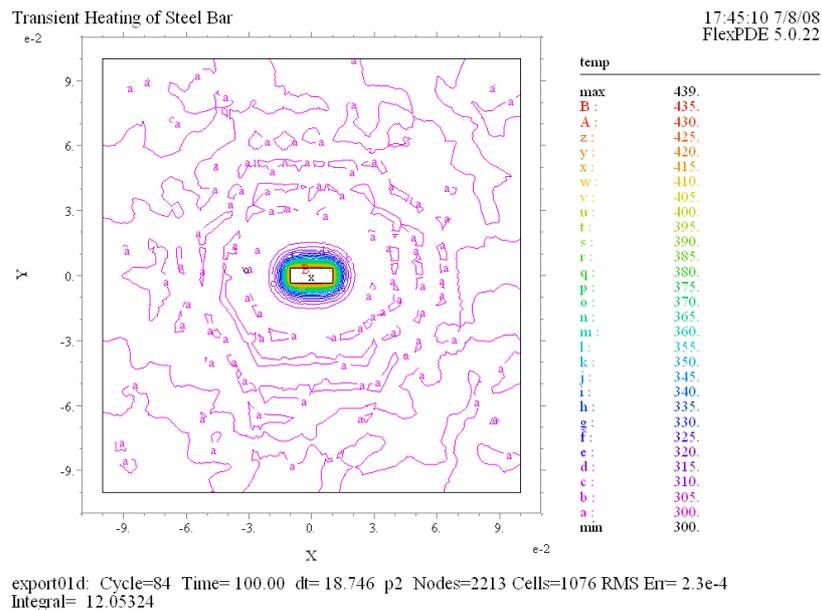
END

## 4.2 実行結果

ここでは出力結果の一部のみを紹介するに留めます。

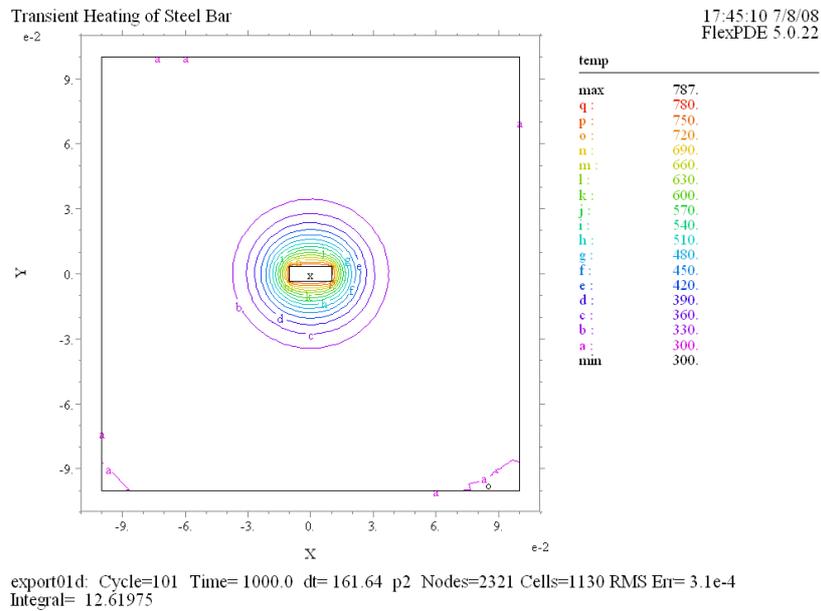
(1) Contour(temp) at  $t = 100$

$t = 100$  の時点では周囲のガラス繊維の温度はほとんどが  $300^\circ\text{K}$  のままです。

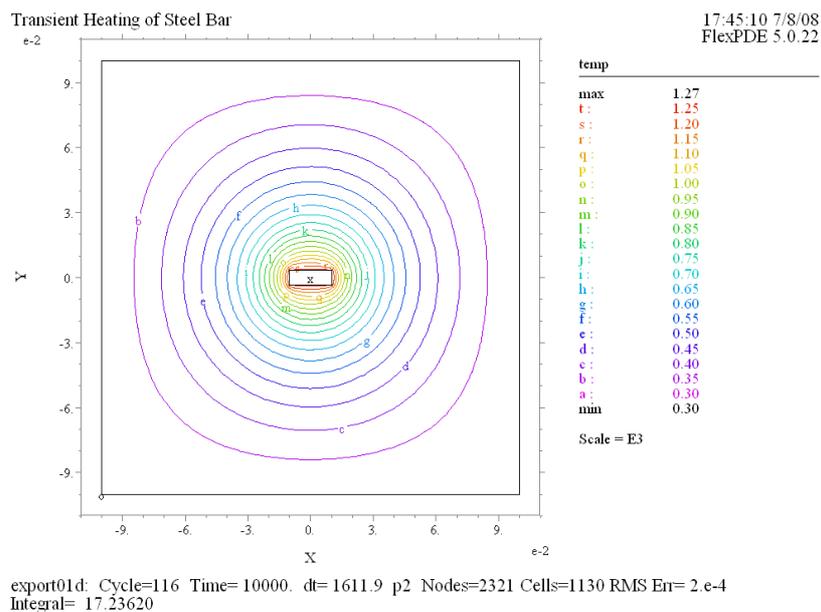


(2) Contour(temp) at  $t = 1000$ 

$t = 1000$  の時点における等温線図です。鉄棒の温度は  $780^\circ\text{K}$  に達しています。

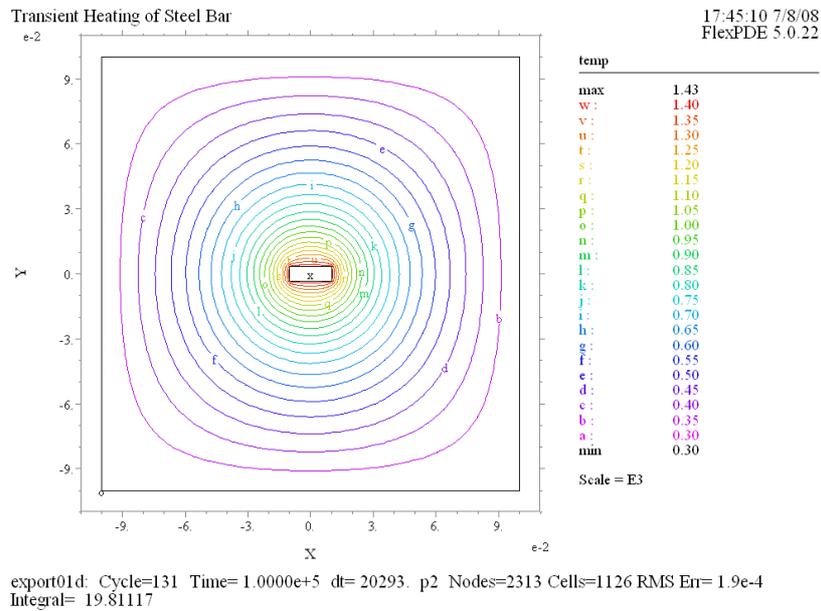
(3) Contour(temp) at  $t = 10000$ 

$t = 10000$  の時点における等温線図です。エンクロージャに近い部分の等温線の形状は角の丸い正方形に近づいて行きます。鉄棒の温度は  $1300^\circ\text{K}$  まで上昇しています。

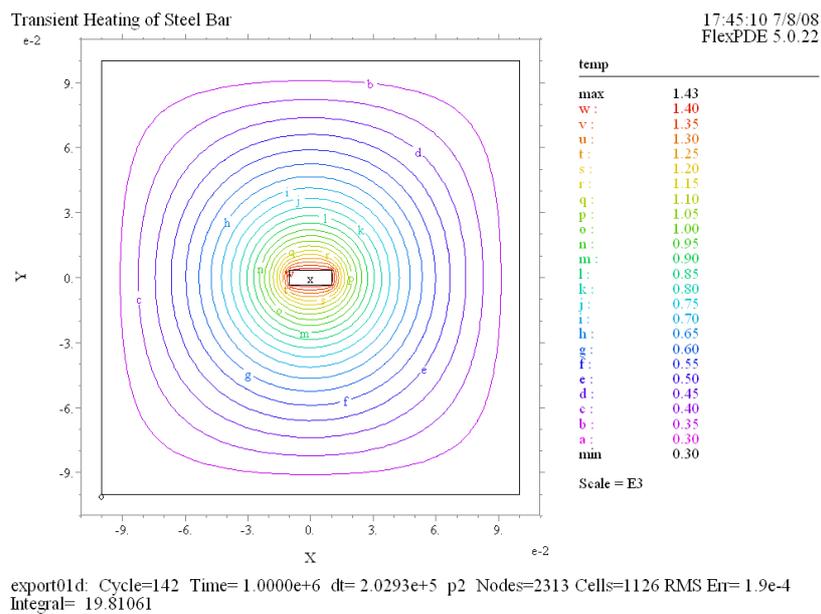


(4) Contour(temp) at  $t = 1e5$ 

$t = 100000$  の時点における等温線図です。

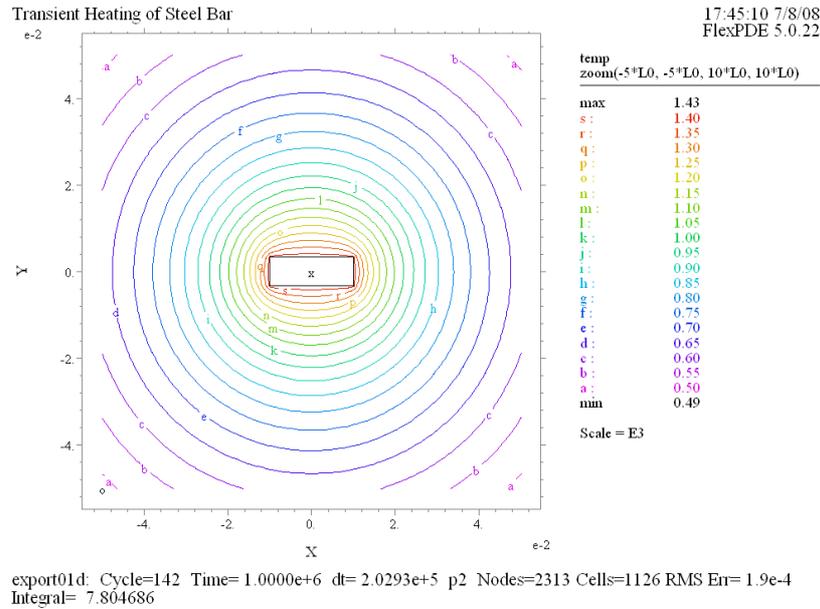
(5) Contour(temp) at  $t = 1e6$ 

$t = 1000000$  の時点における等温線図です。(4) と比べてほとんど変化が見られなくなっています。鉄棒部分の最高温度は  $1430^{\circ}\text{K}$  です。



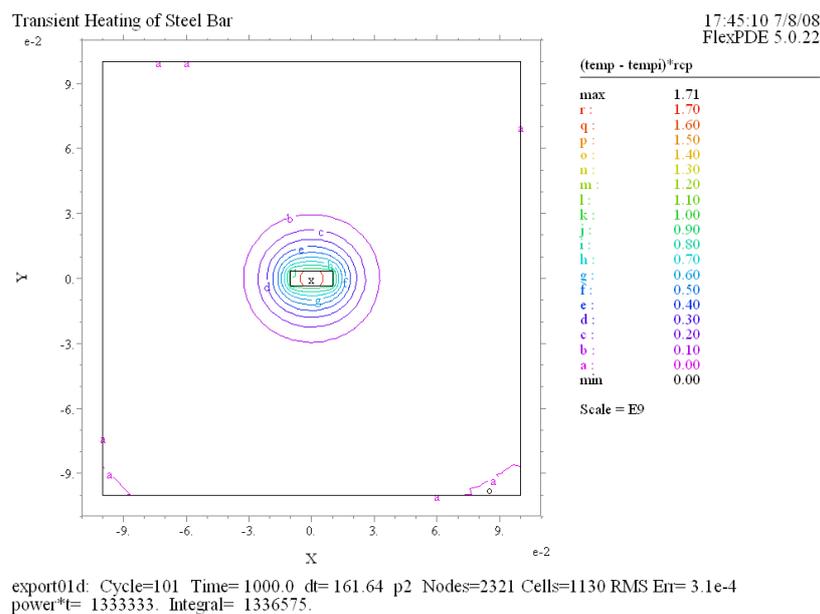
(6) Contour(temp) zoom(-5\*L0, -5\*L0, 10\*L0, 10\*L0) at t = 1e6

最終時点における等温線図 (5) の中心部分を拡大表示したものです。鉄棒に比較的近い部分で等温線の形状はほぼ円形になっていることがわかります。



(7) Contour((temp - temp\_i)\*rcp) Report(power\*t) at t = 1000

これはドメイン全体でのエンタルピー（熱エネルギー）増分を示す図です。Report 文によってプロット下部に鉄棒から発せられた熱量 (power\*t) の積分値が示されていますが、 $t = 1000$  あたりまでは良い一致が見られます。



(8) Contour((temp - tempi)\*rcp) Report(power\*t) at t = 1e6

これに対し  $t = 1000000$  の時点では積分値の乖離が非常に大きくなっています。これは外部境界から熱が失われていることによるものです。境界条件として断熱 (Natural(temp) = 0) という条件を設定すれば積分値は一致するようになります。

