

ADVENTURE_Thermal

Steady / Non-steady Heat Conduction Analysis with HDDM

Version: 2.0

プログラム使用マニュアル

March, 2016

ADVENTURE Project

Contents

1.	はじめに	6
1.1.	本モジュールの特徴	6
1.2.	動作環境	6
1.3.	コンパイルとインストール	6
1.3.1.	ファイルの展開	6
1.3.2.	ディレクトリの構成	7
1.3.3.	コンパイル手順	7
1.3.4.	実行ファイルのインストール	8
1.4.	プログラムの実行	8
2.	並列処理機能及び線形方程式ソルバについて	10
2.1.	並列処理機能	10
2.2.	ソルバの特徴	12
2.3.	疎行列格納形式	13
2.4.	領域分割(ADVENTURE_Metis)について	14
3.	解析機能	15
3.1.	非定常解析	17
3.2.	入出力データ	18
3.3.	温度の単位	18
3.4.	境界条件	19
3.5.	物性値	19
3.6.	出力値	19
4.	コンパイルとインストール	20
4.1.	コンパイル	20
4.2.	実行モジュールのインストール	22
5.	実行方法	23
5.1.	入出力ファイル名	23
5.2.	実行時オプション	24
5.2.1.	非定常解析オプション	24
5.2.2.	要素に関するオプション	24
5.2.3.	反復法のコントロールオプション	24
5.2.4.	疎行列格納形式に関するオプション	25
5.2.5.	線形ソルバの指定と BDD ソルバのオプション	26
5.2.6.	入出力ファイル名の変更オプション	27
	その他のオプション	28
5.2.7.		28
Appendix		30
A.	要素について	30
A.1.	4 面体 1 次要素	30
A.2.	4 面体 2 次要素	31
A.3.	3 角形 1 次要素	32
A.4.	3 角形 2 次要素	33

B.	境界条件と物性値について	34
B.1.	温度規定境界条件	34
B.2.	熱流束規定境界条件	35
B.3.	熱伝達規定境界条件	36
B.4.	熱ふく射規定境界条件	36
B.5.	物性値	37
C.	ツール類について	38
C.1.	一体型解析モデルファイル作成ツール <code>makefem_thermal</code>	38
C.2.	<code>pfemsolv</code>	39
D.	<code>Libsparse</code>	39
E.	解析例	46
E.1.	温度規定境界条件による解析例	47
E.2.	熱流束規定境界条件の解析例	50
E.3.	熱伝達規定境界条件による解析列	53
E.4.	内部発熱の解析例	56
E.5.	複数材料モデルの解析例	59
E.6.	表面情報を扱うモデルの解析例	63
E.7.	大規模解析例	66
6.	参考文献	67

List of Figures

図 1.	階層型領域分割	10
図 2.	領域の CPU への割り当て (シングル版)	11
図 3.	領域の CPU への割り当て (静的負荷分散版)	11
図 4.	領域の CPU への割り当て (動的負荷分散版)	12
図 5.	<code>ADVENTURE_Thermal</code> を使った解析手順	16
図 6.	非定常解析の計算フロー	17
図 7.	入出力ファイル	18
図 8.	4 面体 1 次要素	30
図 9.	4 面体 2 次要素	31
図 10.	3 角形 1 次要素	32
図 11.	3 角形 2 次要素	33
図 12.	温度規定境界条件の設定例	48
図 13.	<code>Advauto_thermalview</code> による温度分布の可視化結果	49
図 14.	解析モデル (円筒の断面)	50
図 15.	<code>Advauto_thermalview</code> による温度分布の可視化結果	52
図 16.	熱伝達規定境界条件の設定例	54
図 17.	<code>Advauto_thermalview</code> による温度分布の可視化結果	55
図 18.	熱伝達規定境界条件の設定例	57
図 19.	<code>Advauto_thermalview</code> による温度分布の可視化結果	58

図 20. 複数部材の表示	60
図 21. 熱伝達規定境界条件の設定例	61
図 22. Advauto_thermalview による温度分布の可視化結果	62
図 23. 冷却ステーブの一部	63
図 24. Advauto_thermalview による温度分布の可視化結果	65
図 25. HTTR モデルの領域分割	66
図 26. Advauto_thermalview による温度分布の可視化結果	66

List of Tables

表 1. ディレクトリ構成	7
表 2. 4 面体 1 次要素の積分点	30
表 3. 4 面体 2 次要素の積分点 (4 点積分)	31
表 4. 4 面体 2 次要素の積分点 (5 点積分)	32
表 5. 3 角形 1 次要素の積分点	32
表 6. 3 角形 2 次要素の積分点	33

バージョン 1.0 からの変更点

AdvThermal-2.0 に下記の新しい機能を追加された.

- 複数の疎行列格納形式に対応する疎行列ライブラリ “libsparse” を追加した.
- Compressed Sparse Row (CSR) をデフォルト格納形式にした.
- シングル版 advthermal-s に BDD ソルバを追加した.
- シングル版 advthermal-s に CG ソルバを追加した.
- 非定常熱伝導解析に Galerkin 法を追加した.
- Makefem_thermal では, 熱伝達規定境界条件と熱ふく射規定境界条件を設定できる.
- Makefem_thermal では, 要素毎に発熱量を設定できる.
- 新しいサンプル問題を追加した.
- 出力データ指定オプションの追加 (結果を出力しない, 表面結果のみ出力する, 或いは表面—インターフェイス結果を出力する)
- 表面及びインターフェイス節点情報によって内部節点を計算する新しいツール “pfemsolv” を追加した.

1. はじめに

このドキュメントは、ADVENTURE Project [1] で開発された固体定常/非定常熱伝導解析のための有限要素法ソルバ ADVENTURE_Thermal の使用マニュアルである。階層型領域分割法（Hierarchical Domain Decomposition Method, 以下 HDDM）に基づいた並列計算方法を採用している。

1.1. 本モジュールの特徴

ADVENTURE_Thermal は以下のような特徴を持っている。

- HDDM に基づく並列計算
 - 全ての計算を 1 プロセスで行うシングルバージョンも含む。
- HDDM ソルバの前処理としてバランシング領域分割法(BDD)[8]
- “libsparse” ライブラリを用いた複数の疎行列格納形式
- 定常及び非定常熱伝導解析
- 四面体 1 次要素と四面体 2 次要素を利用可能
- Linux を含む UNIX 環境で動作可能
- 並列データ処理のために Message Passing Interface (MPI) [6] ライブラリを使用

1.2. 動作環境

ADVENTURE_Thermal は、以下の環境で動作確認を行っている。

オペレーティングシステム	Unix, Linux
並列通信ライブラリ	MPI

1.3. コンパイルとインストール

ADVENTURE_Thermal をインストールするには、MPI のコンパイル環境と ADVENTURE_IO ライブラリがインストールされている必要がある。

1.3.1. ファイルの展開

インストールに必要なファイルは AdvThermal-2.0.tar.gz に含まれている。以下の手順でファイルを展開すると、1.3.2 にあるようなディレクトリが作成される。

```
gunzip -c AdvThermal-2.0.tar.gz | tar xvf -
```

1.3.2. ディレクトリの構成

AdvThermal-2.0.tar.gz を展開すると、表 1 に示すディレクトリ構成となる。

表 1. ディレクトリ構成

ディレクトリ名	内容
hddmsrc	ADVENTURE_Thermal のソースコード
doc	ドキュメント類(このマニュアルを含む)
tools	境界条件設定ツールなど
libfem	有限要素ライブラリのソースコード

表 2 に挙げたディレクトリ以外にも、AdvThermal-2.0 ディレクトリ内に自動設定のためのファイルが作成される。

1.3.3. コンパイル手順

- (1) ADVENTURE_IO ライブラリをマニュアルに従ってインストールする
- (2) AdvThermal-2.0 ディレクトリに移動し、以下のコマンドを実行する。

```
% ./configure
% make
```

configure は、環境に依存する部分を解決し、適切な Makefile を作成するためのシェルスクリプトである。configure に渡せる主なオプションを以下に示す。ただし、以下で用いるディレクトリの指定には、絶対パスを指定する必要がある。

--with-advio=directory

ADVENTURE_IO がインストールされたディレクトリを指定する。デフォルトは "\$HOME/ADVENTURE" である。

--with-mpicc=command

MPI のための C コンパイラのコマンド名を指定する。デフォルトは "mpicc" である。

並列版の ADVENTURE_Thermal をコンパイルするには、MPI 用 C コンパイラが必須である。

--prefix=<install_dir>

ADVENTURE_Thermal のインストール先を指定する。デフォルトは "\$HOME/ADVENTURE" である。

他の configure オプションは第 4 章で説明する。

1.3.4. 実行ファイルのインストール

インストールするには `make install` を実行する。

```
% make install
```

デフォルトのインストール先は `$HOME/ADVENTURE/` である。インストール先を変更する場合は、以下のように実行する。

```
% make install prefix=<install_dir>
```

<install_dir> オプションはインストールディレクトリをフルパスで指定する必要がある。

以下のファイルがインストールされる。

<code>bin/advthermal-s</code>	← 実行ファイル
<code>bin/advthermal-p</code>	← 実行ファイル
<code>bin/advthermal-h</code>	← 実行ファイル
<code>bin/mkbc4th</code>	← 境界条件設定ツール
<code>bin/makefem_thermal</code>	← 解析モデル作成ツール
<code>bin/so2th</code>	← 境界条件データ変換ツール
<code>bin/pfemsolv</code>	← 内部自由度解析ツール
<code>doc/AdvThermal/manual-jp.pdf</code>	← 日本語マニュアル
<code>doc/AdvThermal/manual-en.pdf</code>	← 英語マニュアル
<code>doc/AdvThermal/README.eucJP</code>	← 日本語の簡易情報
<code>doc/AdvThermal/README</code>	← 英語の簡易情報
<code>doc/AdvThermal/copyright</code>	← 著作権規定

1.4. プログラムの実行

ADVENTURE_Thermal には、3 種類の実行モジュールがある。これらの実行モジュールはそれぞれ以下のように実行する。

- ・ シングル版

```
% advthermal_s [options] data_dir
```

- ・ 並列版 (静的負荷分散)

```
% mpirun [options for mpirun] advthermal-p [options] data_dir
```

- ・ 並列版 (動的負荷分散)

```
% mpirun [options for mpirun] advthermal-h [options] data_dir
```


[*options for mpirun*]の部分には、`mpirun` に対する実行オプションを設定する。`[options]`の部分には、`ADVENTURE_Thermal` の実行オプション(詳細は *Section 5.2* を参照のこと)を設定する。`data_dir` オプションは、解析のための入出力データファイルがあるディレクトリを設定する。

`mpirun` の実行に必要な代表的なオプションは以下のようなものがある。詳細は利用している MPI 処理系のマニュアルを参照のこと。

- `-np n`
起動する MPI プロセスの数。
- `-machinefile filename`
計算に使用するマシンのリストファイル。

`ADVENTURE_Thermal` の実行オプション
3 種類の実行モジュール `adventure_s`, `adventure_p` と `adventure_h` の実行オプションは、詳細は *Section 5.2* を参照のこと。

2. 並列処理機能及び線型方程式ソルバについて

ADVENTURE_Thermal は、CPU 間で並列データ処理技術を用いて、定常及び非定常熱伝導解析を行うことができる。これらの特徴は、以下に説明する。

2.1. 並列処理機能

ADVENTURE_Thermal では、階層型領域分割法を用いることで並列処理を実現している。図 1 に階層型領域分割を模式的に示す。1 階層目の大きな分割単位を「部分」(“Part”)と呼び、2 階層目の細かい分割単位を「部分領域」(“Subdomain”)と呼ぶことにする。ADVENTURE_Thermal の実行に先立って、ADVENTURE_Metis による 2 段階の領域分割を行う必要がある。領域分割の詳細は ADVENTURE_Metis のマニュアルをご覧ください。

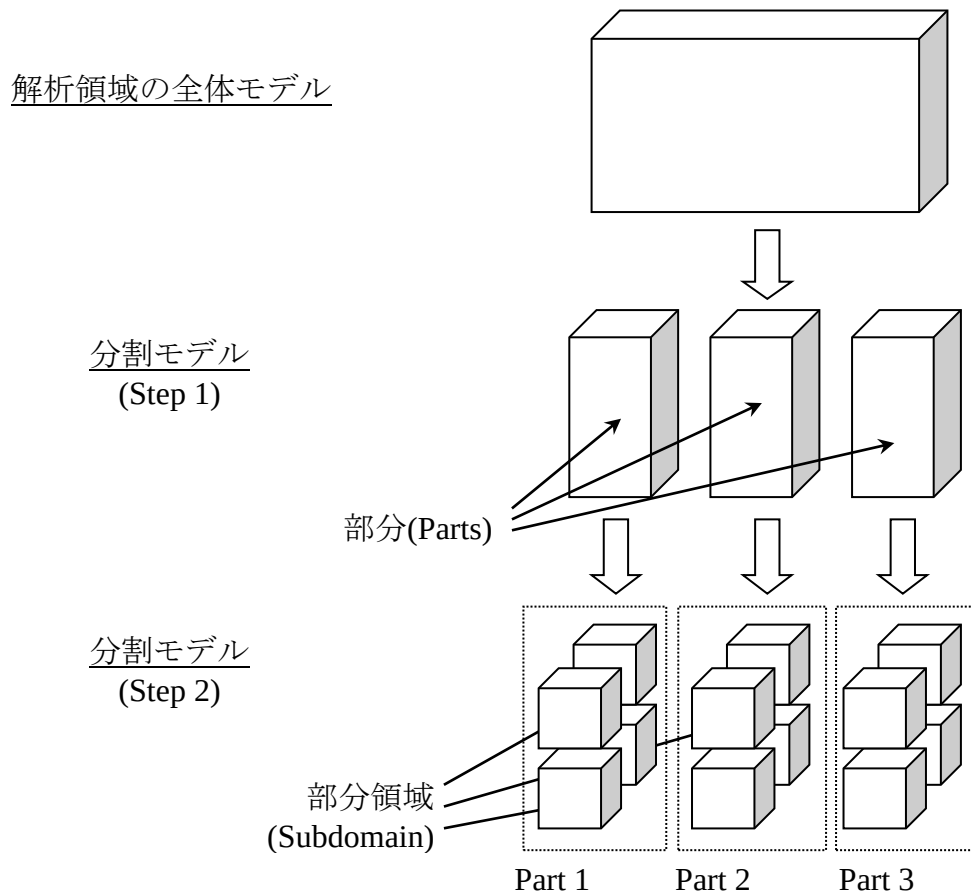


図 1. 階層型領域分割

ADVENTURE_Thermal では、効率的に計算するため、複数の負荷分散手法をサポートしている。並列データ処理のため MPI ライブラリを使用する。配布パッケージには、以下の実行モジュールが含まれている。

(1) シングル版 (advthermal-s)

並列計算は行わずに、全ての計算を一つのプロセスとして実行する。MPI なしでコンパイル・実行が可能である。部分数、領域数に関する制限はなく、並列用に領域分割した解析モデルをそのまま使用して実行できる(図 2)。基本的に、次の静的負荷分散版において各部分に対して並列実行される計算を、1 プロセス内で順に行うのと同じである。そのため、並列計算が実行できないときのデータチェックにも使える。

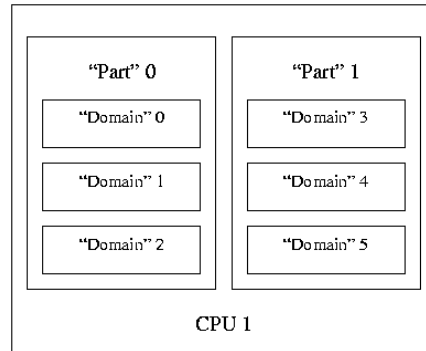


図 2. 領域の CPU への割り当て (シングル版)

(2) 静的負荷分散版 (advthermal-p)

図 3 に示すように、1 つの部分をも 1 つのプロセスに静的に割り当てることで並列に計算を行う。領域分割における部分数と実行プロセス数が同じであるため、ADVENTURE_Metis における部分分割数を静的負荷分散で使用するプロセス数に揃える必要がある。

動的負荷分散版と比べて通信量が少ないため、各ノードの性能が均質な並列計算環境においては、この静的負荷分散版が有効である。

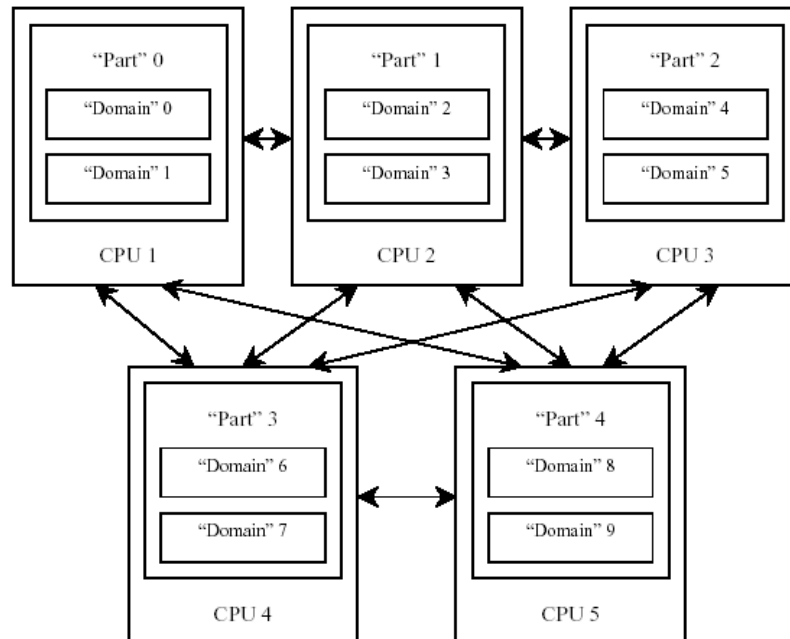


図 3. 領域の CPU への割り当て (静的負荷分散版)

(3) 動的負荷分散版 (advthermal-h)

図 4 に示すように、各プロセスを各領域の計算を行う“子”と部分単位で取りまとめを行う“親”とに分け、“子”への領域の割り当てを動的に行うことで動的な負荷分散を図る並列方式である。各親には 1 つの部分単位が静的に割り当てられるため、全プロセスのうち部分数個が親となり、残りのプロセスが子となる。そのため、起動するプロセス数より少ない部分数で領域分割をしておく必要がある。

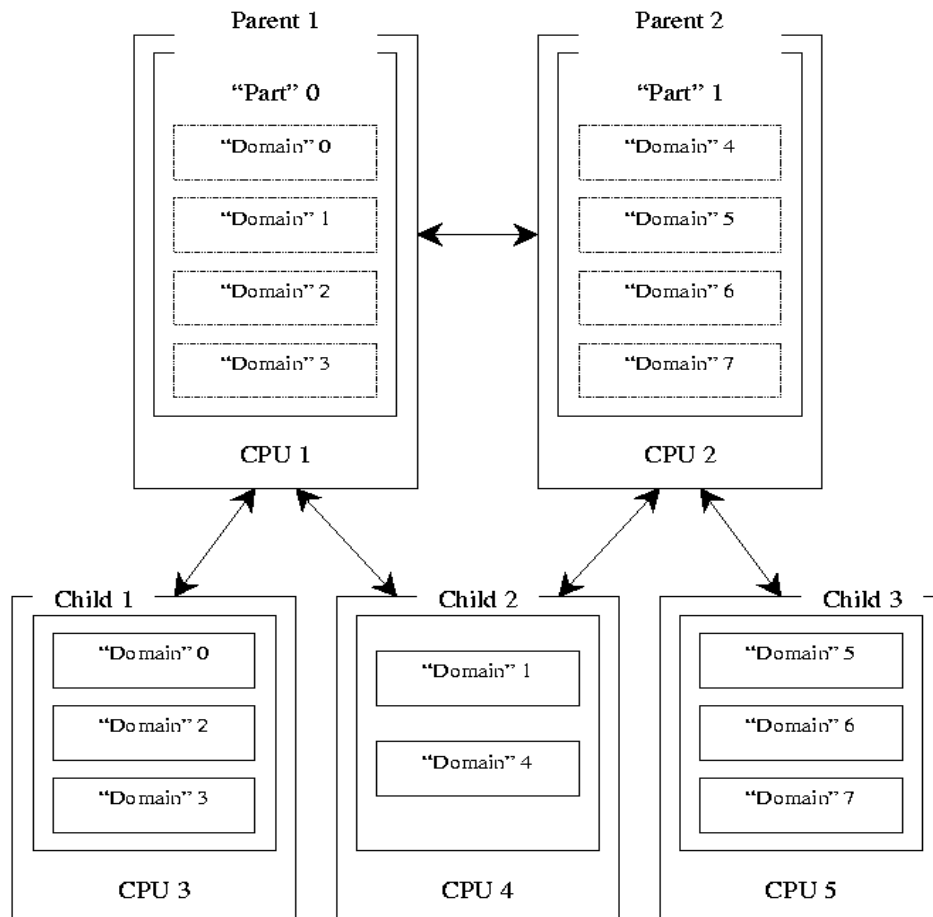


図 4. 領域の CPU への割り当て (動的負荷分散版)

2.2. ソルバの特徴

ADVENTURE_Thermal では、階層型領域分割法を使った有限要素解析を実行するため、4 種類の線型方程式ソルバが実装されている。これらのソルバは、反復法の前処理によって分類される。

- HDDM ソルバ:

HDDM ソルバは、反復法の前処理として対角スケーリング前処理を使用したモードである。

- BDD ソルバ:

ADVENTURE_Thermal では、部分領域内の温度分布を直接法ソルバで、部分領域間の境界の温度を反復法(CG)ソルバで求めている。大規模問題の計算を行うためには、前処理による反復回数の削減が必要となる。BDD ソルバは、CG 前処理として強力であるバランシング領域分割法(BDD)[8]を使用したモードである。BDD 法は Neumann-Neumann 前処理の改良であり、各 CG 反復において解析領域よりも少ない自由度をもつ「コース問題」を解いている。熱伝導解析において、コース問題は各部分領域につき 1 自由度によって構築されており[9]、コース問題における係数行列であるコースマトリックスは部分領域数と同じ次元数を持つ。なお、コース問題は並列直接法ソルバによって解いている。BDD は Neumann-Neumann 前処理の一種であり、1 部分領域ごとに 1 自由度を持たせたコース問題を CG 反復ごとに解く。

BDD ソルバは、HDDM ソルバに比べて反復回数を大きく減少させることができ、計算時間の高速化が期待できる。ここで、初期 CG 反復時に前処理行列を形成するが、前処理の計算コストは対角スケーリングに比べて大きくなるため、その時間は全体の計算時間のおおよそ 15-25%程度となる。ただし、反復回数の減少が大きいいため、解析時間全体では高速化が期待できる。また、使用メモリ量は HDDM ソルバよりも大きくなる。十分なメモリを確保できる計算機環境のユーザーは BDD ソルバの使用を薦める。

BDD ソルバでは CG 反復ごとに Neumann-Neumann 問題を解く必要があるが、部分領域から作られるコースマトリックスは正則行列にならない場合がある。本実装では、正則化パラメータを使用する手法を採用している。

- BDD-DIAG ソルバ:

BDD-DIAG ソルバは、BDD 法における Neumann-Neumann 前処理を対角スケーリングに置き換えた修正 BDD 法を使用している。BDD ソルバに比べ、反復回数が増加することがあるが、BDD ソルバより少ないメモリ使用量で実行することができる。計算モデルによっては、BDD ソルバと計算時間を異なる場合がある。BDD ソルバを使用するのに十分なメモリを確保できない場合は BDD-DIAG ソルバの使用を薦める。これまでの経験上、BDD-DIAG ソルバは BDD ソルバに比べ、約 70%のメモリ量しか必要としないことが分かっている。

- CG ソルバ:

CG ソルバは問題領域全体を CG ソルバで解くモードである。advthermal-s でのみ使用可能である。このモードを使用するときは、領域分割時に 1 パートあたりの部分領域数を 1 にする必要がある。

2.3. 疎行列格納形式

疎行列をメモリ上に保持する場合、値がゼロである成分が多いという点を利用した、特別な形式を用いる必要がある。本モジュールでは、対称な疎行列を

扱うため、下三角部分のみを保持する。メモリ使用量と計算時間を考慮しながら非零成分(*nnz*)のみを保持することを目的とした、いくつかの格納形式を以下に示す。

- Compressed Sparse Row (CSR)
- Coordinate Storage (COO)
- Diagonal Coordinate Storage (DCOO)
- Compressed Sparse Column (CSC)
- Modified Compressed Sparse Row (MSR)
- Incremental Compressed Sparse Row (ICSR)
- Variable Block Compressed Sparse Row (VBCSR)
- Diagonal Block Compressed Sparse Row (DBCSR)

これらの格納形式の詳細については、「Appendix Dlibsparse」を参照のこと。

2.4. 領域分割(ADVENTURE_Metis)について

ADVENTURE_Thermal の計算性能は、ADVENTURE_Metis による領域分割に依存する。ADVENTURE_Metis を実行する前に、部分数と領域数を決めておく必要がある。基本的に、部分数は並列処理の手法、使用するノード数や計算機環境に基づいて決定される。領域数は、計算処理に必要とされるメモリに基づいて決定される。細かく分割すればメモリの使用量を少なくできることは自明のことである。静的負荷分散版 (advthermal-p) の場合、静的負荷分散版の場合、1 領域あたりの要素数が 180-370 のとき BDD ソルバまたは BDD-DIAG ソルバを使用することで最良の性能を得ることができる。一方、動的負荷分散の場合その数は 350-450 である。この値はこれまでの経験値であるため、必ずしも全てのモデルにおいてこの範囲が最適な領域分割になるとは限らない。BDD ソルバもしくは BDD-DIAG ソルバを使用した場合、反復回数と総領域数との間に関連性はない。

ADVENTURE_Metis では、部分数 N_{part} と、1 部分あたりの領域数 $N_{subdomain}$ を指定して実行するため、総要素数を $N_{element}$ とすると、1 領域あたりの要素数 n は次式によって与えられる。

$$n = N_{element} / (N_{part} * N_{domain})$$

$N_{element}$: 総要素数

N_{part} : 部分数

N_{domain} : 1 部分あたりの部分領域数

静的負荷分散の場合と比べると、動的負荷分散のほうが親と子の間のデータ転送が多くなる。そのため、均一な計算機環境では静的負荷分散のほうが計算性能が良い。

3. 解析機能

ADVENTURE_Thermal による解析の手順を図 5.に示す。

- (1) メッシュデータの作成 (ADVENTURE_TetMesh)
ADVENTURE_TetMesh モジュールを用いて、解析モデルのメッシュ分割を行う。
- (2) 境界条件の付加 (ADVENTURE_BCtool)
メッシュに対して ADVENTURE_BCtool を用いて境界条件を付加する。
- (3) 解析用モデルデータの変換
ADVENTURE_BCtool Ver.2.0 の **makefem3** を用いて全体のメッシュデータと境界条件を、一体型 FEA モデルデータ (ADVENTURE フォーマット) に変換する。AdvThermal の **makefem_thermal** ツールも同じデータの変換に使用できる。
※なお、本バージョンからは、**makefem_thermal** で熱伝達境界条件や熱ふく射を含め全ての境界条件を設定可能となったため、旧バージョンで使用した **so2th** や **mkbc4th** は不要になった。
- (4) 領域分割 (ADVENTURE_Metis)
ADVENTURE_Metis モジュールを用いて、一体型の解析モデルより階層型に領域分割されたモデルを作成する。なお、実行時にはオプション **-difn 1** を用いる必要がある。
例えば、以下のように実行する。


```
% mpirun [mpi_options] adventure_metis -difn 1 [options]
model_filename directory_name div_num
```

上記は表記上 2 行に分かれているが、実際には 1 行のコマンドとして入力する。
オプション **-difn 1** は、内部境界上節点の自由度を 1 に指定するためのものである。これは、固体静解析で求める節点変位の自由度は 3 であるのに対し、熱伝導解析で求める節点温度の自由度が 1 であるためである。ADVENTURE_Thermal への入力モデル作成において、**-difn 1** は必須オプションである。
- (5) 熱伝導解析 (ADVENTURE_Thermal)
ADVENTURE_Thermal モジュールを用いて、分割された解析モデルを入力として、有限要素解析を行う。
- (6) 結果の可視化 (ADVENTURE_POSTtool or ADVENTURE_Auto)

ADVENTURE_POSTtool モジュールを用いて、解析結果を可視化する。または、ADVENTURE_Auto に含まれる advauto_thermalview によっても温度のスカラー分布を可視化することができる。

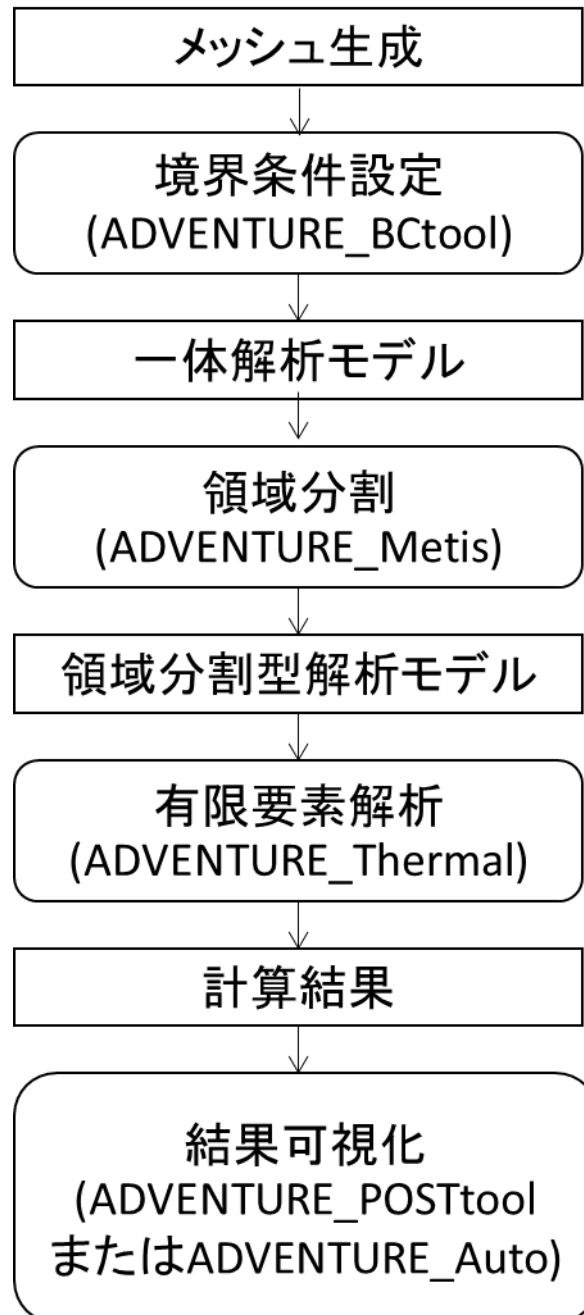


図 5. ADVENTURE_Thermal を使った解析手順

3.1. 非定常解析

非定常解析には、後方有限差分近似とクランク・ニコルソン法が用いられている。図 6 にそのアルゴリズムを示す。このアルゴリズムには 2 つの反復がある。外側の反復は時間積分であり、内側の反復は階層型領域分割法に基づく CG 法によるものである。

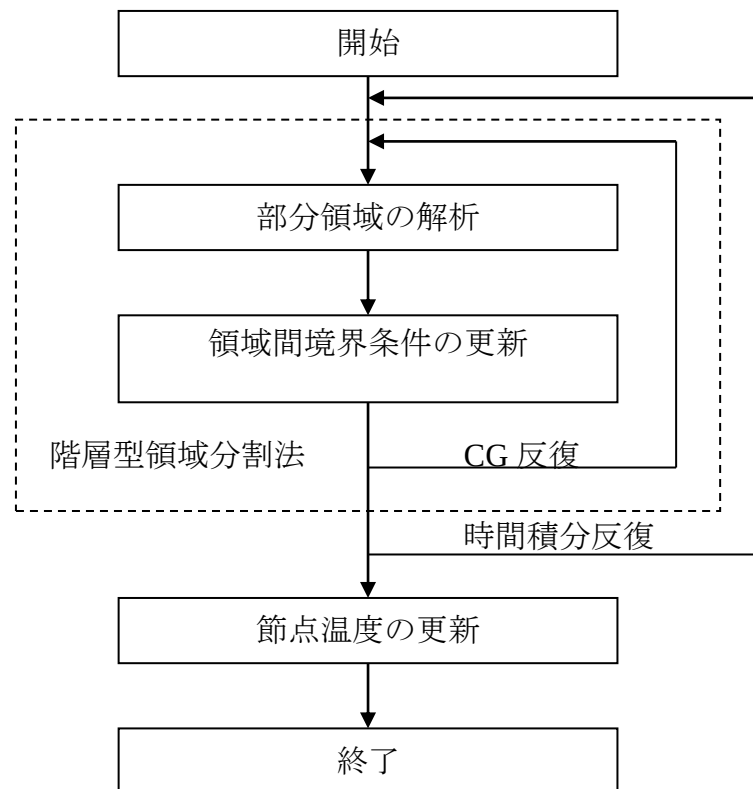


図 6. 非定常解析の計算フロー

3.2. 入出力データ

ADVENTURE_Thermal が用いる入出力ファイルを図 7 に示す。ログファイル以外はすべて ADVENTURE フォーマットのバイナリファイルである。1 ファイル中に 1 パート分のデータが存在する。

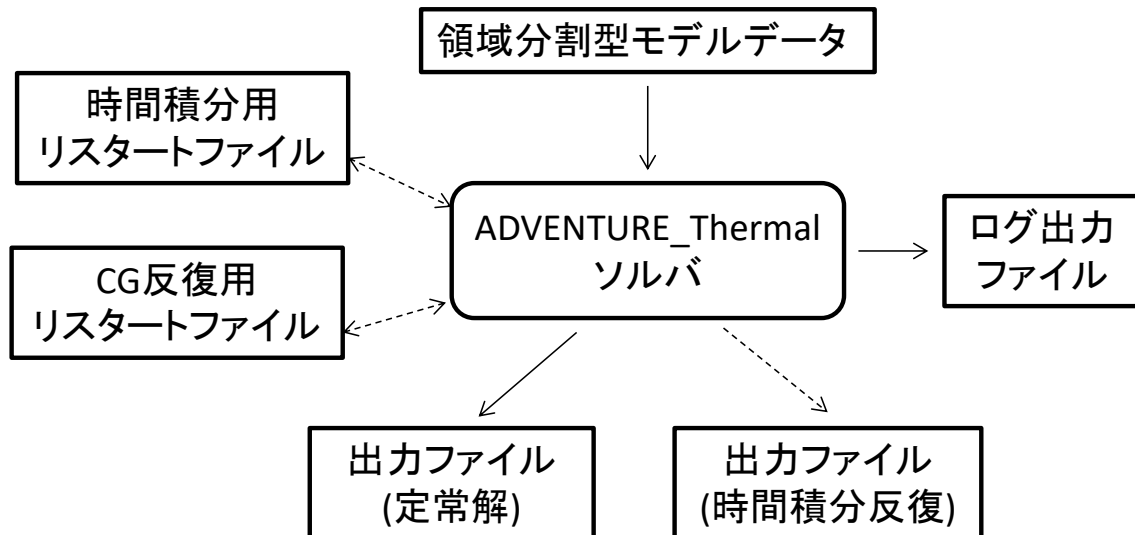


図 7. 入出力ファイル

ADVENTURE_Thermal モジュールは、ADVENTURE_Metis によって作成された階層型領域分割型データファイルを入力として使用する。計算されたすべての節点の温度も、階層型領域分割型データファイルとして保存される。時間積分ステップごとの計算結果も出力可能である。

リスタートファイルを出力して計算を中断したり、リスタートファイルを使って計算を再開することもできる。2 種類のリスタートファイルが使用可能である。

- 1). CG 反復用リスタートファイル
定常解析で使用
- 2). 時間積分反復用リスタートファイル
非定常解析で使用

BDD ソルバを利用する場合は、コース行列(LU 分解)のためのリスタートファイルも使用することができる。

3.3. 温度の単位

ADVENTURE_Thermal では、温度データは摂氏を単位として取り扱う。他の温度単位はサポートしていない。

3.4. 境界条件

以下の境界条件が設定可能である。

- 温度規定 (節点に設定)
- 熱流束規定 (節点に設定)
- 熱伝達規定 (要素面に設定)
- 熱ふく射規定 (要素面に設定: 非定常解析のみ)

熱ふく射規定境界条件は、非定常解析の時のみ使用可能である。

3.5. 物性値

以下の等方性物性値が使用可能である。

- 熱伝導率
- 比熱 (非定常解析で必要)
- 質量密度 (非定常解析で必要)
- ステファン・ボルツマン定数 (熱ふく射規定境界条件の設定時に必要)
- 発熱量

3.6. 出力値

下記の値をファイルに保存できる。

- 全ての節点温度
- 全ての節点流束
- 表面とインターフェース節点温度
- 表面とインターフェース節点流束
- 表面節点温度
- 表面節点流束

1つの ADVENTURE バイナリファイルには、1 パート分の結果が保存される。

4. コンパイルとインストール

4.1. コンパイル

ADVENTURE_Thermal モジュールをコンパイルするには、MPI 環境の設定と ADVENTURE_IO ライブラリのインストールが必要である。以下の手順で ADVENTURE_Thermal のコンパイルを行う。

1. `./configure`
2. `make`

ただし、どちらのコマンドも ADVENTURE_Thermal モジュールのトップディレクトリで実行する必要がある。

configure は、環境に依存する部分を解決し、適切な Makefile を作成するためのシェルスクリプトである。configure に渡せる主なオプションを以下に示す。ただし、ディレクトリの指定には、絶対パスを指定する必要がある。

--with-advio=directory

ADVENTURE_IO が prefix の指定(あるいはデフォルトの) 場所と異なる場所にインストールされている場合に、そのインストール先ディレクトリを指定する。デフォルトは \$HOME/ADVENTURE である。

--with-mpicc=command

MPI の C コンパイラ名を指定する。デフォルトは mpicc である。MPI の C コンパイラが見つからない場合は、ADVENTURE_Thermal の並列版はコンパイルされない。

--with-mpi-cflags=CFLAGS

MPI プログラムをコンパイルするための C コンパイルオプションを指定する。例えば、MPI インクルードファイルを指定する必要がある場合には、**--with-mpi-cflags=" -I/usr/local/include/mpi"** などとなる。MPI プログラムのコンパイル時には、ここで指定したものに加えてシングル版用に設定されているものの両方が用いられる。

--with-mpi-libs =LIBS

ライブラリをリンクする場合に必要なオプションを指定する。例えば、MPI ライブラリを明示する必要がある場合には、**--with-mpi-libs=" -L/usr/local/lib/mpi -lmpi"** などとなる。MPI プログラムのリンク時には、ここで指定したものに加えてシングル版用に必要なものの両方が用いられる。

--enable-optimize

コンパイル時に最適化を行う。これにより設定されるオプション以外の最適化オプションをつけてコンパイルしたい場合は、下の書式を使用する。

--enable-optimize=CFLAGS

CFLAGS を最適化用のオプションとして、コンパイル時に最適化を行う。

--prefix=install_dir

インストール先のトップディレクトリを `install_dir` にする。実行モジュールは、`install_dir/bin` にインストールされる。デフォルトは `$HOME/ADVENTURE` である。

`configure` を使用することで多くの環境ではコンパイル可能と思われるが、うまくいかない場合には `Makefile` のサンプルが各ディレクトリに用意されてあるので、それを用いてコンパイルする。`Makefile.sample` が存在する各ディレクトリにおいて、`Makefile.sample` を `Makefile` にコピーする。加えて、`ADVENTURE_Thermal` モジュールのトップディレクトリにおいて、`Mekefile.in.sample` を `Makefile.in` にコピーする。

計算機環境に応じて、以下に示す `Makefile.in` 中のマクロを変更する必要がある。

<code>ADVSYS_DIR</code>	← <code>ADVENTURE</code> システムがインストールされているディレクトリ
<code>ADVIO_CONFIG</code>	← <code>advsys-config</code> スクリプトのフルパス
<code>MPI_CC</code>	← MPI 用の C コンパイラ
<code>MPI_LINKER</code>	← MPI 用の C リンカ
<code>CC</code>	← C コンパイラ
<code>LINKER</code>	← C リンカ
<code>CFLAGS</code>	← 最適化オプション

`Makefile.in` を変更した後に、トップディレクトリで `make` を実行する。

```
% make
```

なお、各ディレクトリにおいて手動で `make` を実行する場合は、`hddmsrc` ディレクトリをコンパイルする前に `libfem` をコンパイルしておく必要がある。

4.2. 実行モジュールのインストール

コンパイルに成功したら、以下によりモジュールのインストールが行われる。

```
% make install
```

インストール先は `configure` で指定した場所である。インストールディレクトリを `configure` 時と変更したい場合は、以下のコマンドを実行すればよい。

```
% make install prefix=<install_dir>
```

ただし、`install_dir` にはインストールするディレクトリの絶対パスを指定する必要がある。

以下のファイルがインストールされる。

<code>bin/advthermal-s</code>	← シングル版実行モジュール
<code>bin/advthermal-p</code>	← 静的負荷分散版実行モジュール
<code>bin/advthermal-h</code>	← 動的負荷分散版実行モジュール
<code>bin/mkbc4th</code>	← 境界条件設定ツール
<code>bin/makefem_thermal</code>	← 一体型解析モデルファイル作成ツール
<code>bin/so2th</code>	← 解析データ変換ツール
<code>bin/pfemsolv</code>	← 境界条件とインターフェース自由度を用いた内部自由度解析ツール
<code>doc/AdvThermal/manual-jp.pdf</code>	← 日本語ユーザーマニュアル
<code>doc/AdvThermal/manual-en.pdf</code>	← 英語ユーザーマニュアル
<code>doc/AdvThermal/README.eucJP</code>	← 日本語の簡易情報
<code>doc/AdvThermal/README</code>	← 英語の簡易情報
<code>doc/AdvThermal/copyright</code>	← 著作権規定

5. 実行方法

ADVENTURE_Thermal は並列処理方法によって 3 つのモジュールがある。それぞれ次のコマンドで実行を行う。

- シングル版の場合

```
% advthermal_s [options] data_dir
```

- 静的負荷分散による並列版の場合

```
% mpirun [options for mpirun] advthermal-p [options] data_dir
```

- 動的負荷分散による並列版の場合

```
% mpirun [options for mpirun] advthermal-h [options] data_dir
```

ここで、[options for mpirun] は mpirun コマンドに対するオプションであり、[options] は ADVENTURE_Thermal に対するオプションである。このオプション指定により、解析種類の指定や種々の設定を行う。(詳しくは 5.2 参照)

data_dir は必須オプションであり、入出力データファイルのトップディレクトリを指定する。

5.1. 入出力ファイル名

デフォルトの各入出力ファイル名は、以下に示すとおりである。これらのファイルは、data_dir で指定されたディレクトリの下に置かれる。ファイル名中の "P" は部分番号を、"S" は時間積分ループのステップ番号をそれぞれ示す。

- HDDM 型の解析モデル入力ファイル:
data_dir/model/advhddm_in_P.adv
- 解析結果出力ファイル (定常解析時):
data_dir/result/advhddm_out_P.adv
- 解析結果出力ファイル (非定常解析時):
data_dir/result/advhddm_out_S_P.adv
- CG リスタート入出力ファイル:
data_dir/cg-res/advhddm_in_P.adv
- 時間積分リスタート入出力ファイル:
data_dir/result(right?)/advhddm_out_S_P.adv

5.2. 実行時オプション

以下に示すオプションが設定可能である。

5.2.1. 非定常解析オプション

- `-ns` 非定常解析を行うときに指定する。以下のサブオプションが使用可能であり、`-ns` に続けて指定することができる。
- `--cn` クランク・ニコルソン法による時間積分を行う。指定しない場合は後退差分近似を行う。
- `--gn` Galerkin 法における時間積分を行う。
- `--step n` 時間積分の反復回数の上限を n 回に指定する。デフォルトは 10 回。
- `--out-interval n` 時間積分ステップ n 回ごとに解析結果ファイルを出力する。デフォルト値はない；最終ステップの結果のみ出力する。
- `--dt x` 時間方向の刻み幅を x に指定する。デフォルトは 1.0。
- `--init x` 全節点における初期温度を $x^{\circ}\text{C}$ に指定する。デフォルトは 0°C 。
- `--use-resin n` 以前の実行において出力された時間積分ステップ n におけるリスタートファイルを読み込み、そこから解析を再開する。

5.2.2. 要素に関するオプション

- `-tet10-integ5` 4 面体 2 次要素の積分において、5 点積分を行う。指定しない場合は 4 点積分である。

5.2.3. 反復法のコントロールオプション

ADVENTURE_Thermal では、線形方程式を解くために CG 法による反復計算を行っており、それらをコントロールするためにいくつかのオプションがある。

- `-cg-tol x` 収束判定基準を指定する。これは、CG 反復における残差ベクトルの初期残差ベクトルに対するノルムの相対誤差であり、相対誤差がこの値より小さく

なった時点で CG 法が収束したと判定する。デフォルトは 1.0×10^{-6} である。

- `-cgloop-max n` CG 反復回数の上限を指定する。反復回数がこの値を超えた場合は、収束に至る前でもプログラムは終了する。デフォルトは **1000** である。
- `-use-cg-resin` 以前に作成した CG リスタートファイルを読み込み、その時点から解析を再開する。定常解析においてのみ有効である。指定しない場合は読み込まない。
- `-resout-cglast` CG リスタートファイルの出力を最後の CG ステップで行う。収束時、および収束せずに反復回数の上限に達した場合に出力を行う。指定しない場合は出力しない。

5.2.4. 疎行列格納形式に関するオプション

- `-keep-kmat-csr` 領域 FEM 計算のための行列を CSR 形式で格納する。デフォルトはこの形式である。
- `-keep-kmat-coo` 領域 FEM 計算のための行列を COO 形式で格納する。
- `-keep-kmat-msr` 領域 FEM 計算のための行列を MSR 形式で格納する。
- `-keep-kmat-csc` 領域 FEM 計算のための行列を CSC 形式で格納する。
- `-keep-kmat-icrsr` 領域 FEM 計算のための行列を ICSR 形式で格納する。
- `-keep-kmat-vbcsr` 領域 FEM 計算のための行列を VBCSR 形式で格納する。
- `-keep-kmat-dcoo` 領域 FEM 計算のための行列を DCOO 形式で格納する。
- `-keep-kmat-dbcsr` 領域 FEM 計算のための行列を DBCSR 形式で格納する。このオプションは `ADVENTURE_Solid` のみ利用できる。

- `-keep-kmat-sky` 領域 FEM 計算のための行列を SKY 形式で格納する。

これらの格納形式の詳細については、付録の libsparse ライブラリ(Appendix D)を参照のこと。

5.2.5. 線形ソルバの指定と BDD ソルバのオプション

ADVENTURE_Thermal では、線形方程式を解くために BDD ソルバによる反復計算を行うことが可能であり、それらをコントロールするためにいくつかのオプションがある。

- `-solver cg` 線形ソルバに CG を指定する (advthermal-s にのみ使用可能)。
- `-solver hddm` 線形ソルバに HDDM を指定する。
- `-solver bdd` 線形ソルバに BDD を指定する。
- `-solver bdd-diag` 線形ソルバに BDD-DIAG を指定する。
- `-solver bdd -iLU` 線形ソルバに IBDD を指定する。
- `-solver bdd-diag -iLU` 線形ソルバに IBDD-DIAG[11]を指定する。
- `-resout-bdd-cmat` LU 分解後のコースマトリックスを出力する。指定しない場合は出力しない。
- `-use-bdd-cmat` 以前に作成した LU 分解後のコースマトリックスを読み込み、利用する。指定しない場合は読み込まない。
- `-bdd-dir dir` コースマトリックスの入出力データディレクトリ名を指定する。入出力用ディレクトリ下のサブディレクトリ名を指定する。デフォルトは bdd である。
- `-bdd-cmat-file file` コースマトリックスの入出力ファイル名を指定する。指定した名前に `_P.adv` を付けた(P はプロセッサ番号)がファイル名となる。デフォルトは、`advhbdd_cmat_*` である。ここで “*” はプロセス番号である。このオプションは、`-resout-bdd-cmat` オプションの後で指定する必要がある。

- `-ginv-alpha x` Neumann–Neumann 前処理時に用いる非負の正則化パラメータを指定する。デフォルトは 10^{-3} である。BDD もしくは IBDD を線形ソルバとして使用する場合にのみ有効である。

5.2.6. 入出力ファイル名の変更オプション

通常は、ユーザは入出力データがあるトップディレクトリを指定するだけでよいが、サブディレクトリ名やファイル名を変更したい場合は以下のオプションを使用する。ただし、*S* は時間積分ステップ番号、*P* は部分番号を示す。

- `-model-file file` 解析モデル入力ファイル名を *file* とする。実際のファイル名は、これに `_P.adv` をつけたものになる。デフォルトは `advhddm_in` である。
- `-model-dir dir` 解析モデル入力ファイルのあるサブディレクトリ名を *dir* とする。デフォルトは `model` である。
- `-result-file file` 解析結果出力ファイル名を *file* とする。実際のファイル名は、定常解析時はこれに `_P.adv` をつけたものに、非定常解析時はこれに `_S_P.adv` をつけたものになる。デフォルトは `advhddm_out` である。
- `-result-dir dir` 解析結果出力ファイルのあるサブディレクトリ名を *dir* とする。デフォルトは `result` である。
- `-ns-resin-file file` 時間積分ステップリスタート入力ファイル名を *file* とする。実際のファイル名は、これに `_S_P.adv` をつけたものとなる。デフォルトは `advhddm_out` である。
- `-ns-resin-dir dir` 時間積分ステップリスタート入力ファイルのあるサブディレクトリ名を *dir* とする。デフォルトは `result` である。
- `-cg-resin-file file` CG リスタート入力ファイル名を *file*

- とする。実際のファイル名は、これに `_P.adv` をつけたものとなる。デフォルトは `advhddm_cgres` である。
- `-cg-resin-dir dir` CG リスタート入力ファイルのあるサブディレクトリ名を *dir* とする。デフォルトは `cg-res` である。
 - `-cg-resout-file file` CG リスタート出力ファイル名を *file* とする。実際のファイル名は、これに `_P.adv` をつけたものとなる。デフォルトは `advhddm_cgres` である。
 - `-cg-resout-dir dir` CG リスタート入力ファイルのあるサブディレクトリ名を *dir* とする。デフォルトは `cg-res` である。
 - `-result-surface-interface` 表面とインターフェース結果を出力する。このオプションを使用するには、`advFile(metis 分割前)`は `makefem_thermal` の `-with-surface` オプションによる作成する必要がある。
 - `-result-surface` 表面結果を出力する。このオプションを使用するには、`advFile(metis 分割前)`は `makefem_thermal` の `-with-surface` オプションによる作成する必要がある。
 - `-no-result` 結果を出力しない。

5.2.7. その他のオプション

- `-file-para` ファイル入出力を各プロセスが並列に行う。指定しない場合は各プロセスが同時にディスクアクセスを起こさないように順番に行う。
- `-memlimit n` 各プロセスが使用するメモリの上限を *n* [Mbyte] とし、これを越えた場合、その時点で実行を停止する。デフォルトは 256 [Mbyte] である。
- `-help or -h` メインのヘルプメッセージを表示する。
- `-version or -v` モジュールのバージョンを表示する。

- `-help-ns` 非定常解析のコントロールオプション指定のためのヘルプメッセージを表示する。
- `-help-cg` CG 法のコントロールオプション指定のためのヘルプメッセージを表示する。
- `-help-bdd` BDD ソルバのコントロールオプション指定のためのヘルプメッセージを表示する。

Appendix

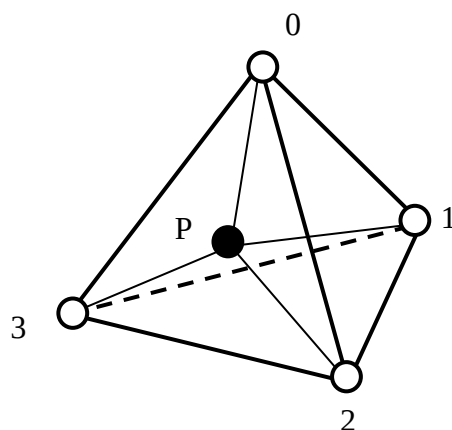
A. 要素について

ADVENTURE_Thermal は、4 面体 1 次要素及び 2 次要素に対応している。ただし、境界条件として熱伝達境界あるいは熱ふく射境界を扱う場合には、境界面ごとに剛性行列を作成する必要がある。このような面に対しては、4 面体 1 次要素の場合は 3 三角形 1 次要素を、4 面体 2 次要素の場合は 3 三角形 2 次要素を抽出し、積分を行っている。以下に各要素の積分番号、積分点について示す。

A.1. 4 面体 1 次要素

(1). 節点

節点数は 4 であり、要素コネクティビティでの各節点の節点番号の並び順は図 8 のようになっている。



○ 一次節点

図 8.4 面体 1 次要素

(2). 積分点

積分点数は 1 であり、積分点は体積座標 (L_0, L_1, L_2, L_3) を用いて表 2 のようになっている。体積座標は、図 8 で示されている点 P を以下の式で表す座標系である。

$$L_0 = 4 \text{ 面体 } P123 \text{ の体積 } / 4 \text{ 面体 } 0123 \text{ の体積} \quad (1)$$

$$L_1 = 4 \text{ 面体 } P023 \text{ の体積 } / 4 \text{ 面体 } 0123 \text{ の体積} \quad (2)$$

$$L_2 = 4 \text{ 面体 } P013 \text{ の体積 } / 4 \text{ 面体 } 0123 \text{ の体積} \quad (3)$$

$$L_3 = 4 \text{ 面体 } P012 \text{ の体積 } / 4 \text{ 面体 } 0123 \text{ の体積} \quad (4)$$

表 3.4 面体 1 次要素の積分点

積分点番号	L_0	L_1	L_2	L_3
0	1/4	1/4	1/4	1/4

A.2.4 面体2次要素

(1). 節点

節点数は10であり、要素コネクティビティでの各節点の節点番号の並び順は図9のようになっている。

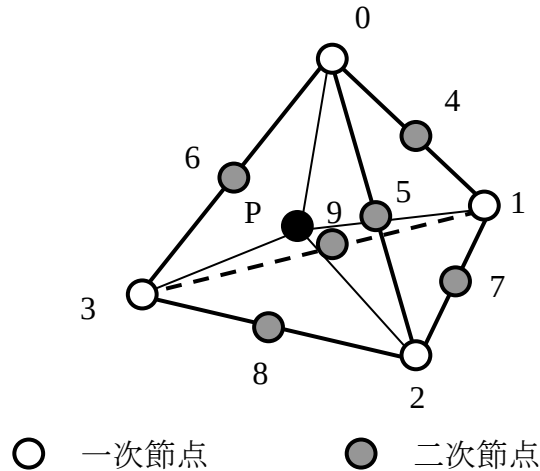


図 9.4 面体2次要素

(2). 積分点

積分点数はデフォルトでは4であり、積分点は体積座標 (L_0, L_1, L_2, L_3) を用いて表3のようになっている。ここで体積座標は図9で示されている点Pを以下の式で表す座標系である。

$$L_0 = 4 \text{ 面体 } P123 \text{ の体積 } / 4 \text{ 面体 } 0123 \text{ の体積} \quad (5)$$

$$L_1 = 4 \text{ 面体 } P023 \text{ の体積 } / 4 \text{ 面体 } 0123 \text{ の体積} \quad (6)$$

$$L_2 = 4 \text{ 面体 } P013 \text{ の体積 } / 4 \text{ 面体 } 0123 \text{ の体積} \quad (7)$$

$$L_3 = 4 \text{ 面体 } P012 \text{ の体積 } / 4 \text{ 面体 } 0123 \text{ の体積} \quad (8)$$

表 4.4 面体2次要素の積分点 (4点積分)

積分点番号	L_0	L_1	L_2	L_3
0	β	α	β	β
1	β	β	α	β
2	β	β	β	α
3	α	β	β	β

ただし、

$$\alpha = 0.58541019662496845446$$

$$\beta = 0.13819660112501051518$$

また、実行時のオプション指定により、5点積分を行うことが可能である。この場合の積分点は、同様に体積座標を用いて表4のようになっている。

表 5.4 面体 2 次要素の積分点 (5 点積分)

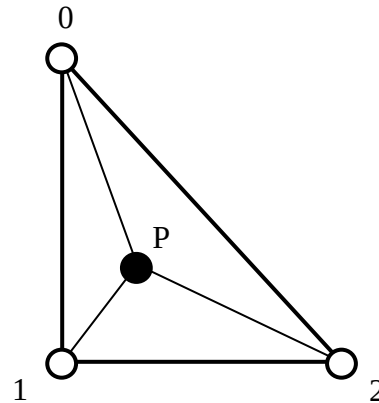
積分点番号	L_0	L_1	L_2	L_3
0	1/4	1/4	1/4	1/4
1	1/6	1/2	1/6	1/6
2	1/6	1/6	1/2	1/6
3	1/6	1/6	1/6	1/2
4	1/2	1/2	1/6	1/6

A.3. 3 角形 1 次要素

4 面体 1 次要素を使用する場合、熱伝達境界あるいは熱ふく射境界面の積分には 3 角形 1 次要素を用いる。

(1). 節点

節点数は 3 であり、要素コネクティビティでの各節点の節点番号の並び順は図 10 のようになっている。



○ 一次節点

図 10.3 角形 1 次要素

(2). 積分点

積分点数は 1 であり、積分点は面積座標 (L_0, L_1, L_2) を用いて表 5 のようになっている。ここで面積座標は図 10 で示されている点 P を以下の式で表す座標系である。

$$L_0 = 3 \text{ 角形 } P12 \text{ の面積 } / 3 \text{ 角形 } 012 \text{ の面積} \quad (9)$$

$$L_1 = 3 \text{ 角形 } P02 \text{ の面積 } / 3 \text{ 角形 } 012 \text{ の面積} \quad (10)$$

$$L_2 = 3 \text{ 角形 } P01 \text{ の面積 } / 3 \text{ 角形 } 012 \text{ の面積} \quad (11)$$

表 6.3 角形 1 次要素の積分点

積分点番号	L_0	L_1	L_2
0	1/3	1/3	1/3

A.4. 3 角形 2 次要素

4 面体 2 次要素を使用する場合、熱伝達境界あるいは熱ふく射境界面の積分には 3 角形 2 次要素を用いる。

(1). 節点

節点数は 6 であり、要素コネクティビティでの各節点の節点番号の並び順は図 11 のようになっている。

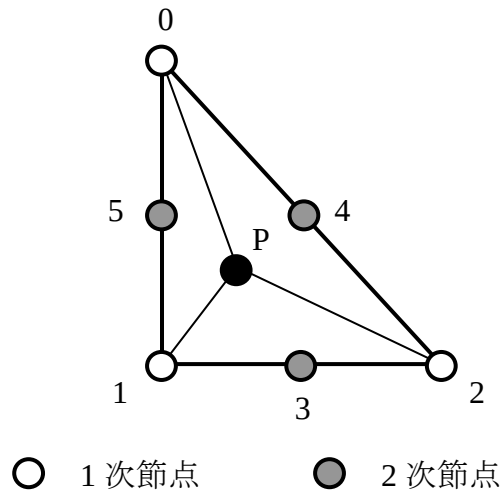


図 11. 3 角形 2 次要素

(2). 積分点.

積分点数は 3 であり、積分点は体積座標(L_0, L_1, L_2) を用いて表 6 のようになっている。ここで面積座標は図 11 で示されている点 P を以下の式で表す座標系である。

$$L_0 = 3 \text{ 角形 } P12 \text{ の面積} / 3 \text{ 角形 } 012 \text{ の面積} \quad (12)$$

$$L_1 = 3 \text{ 角形 } P02 \text{ の面積} / 3 \text{ 角形 } 012 \text{ の面積} \quad (13)$$

$$L_2 = 3 \text{ 角形 } P01 \text{ の面積} / 3 \text{ 角形 } 012 \text{ の面積} \quad (14)$$

表 7. 3 角形 2 次要素の積分点

積分点番号	L_0	L_1	L_2
0	1/2	1/2	0
1	0	1/2	1/2
2	1/2	0	1/2

B. 境界条件と物性値について

ADVENTURE_Thermal で利用可能な境界条件および物性値のフォーマットの例を以下に示す。

B.1. 温度規定境界条件

```
[Properties]
1: content_type=FEGenericAttribute
2: num_items=81
3: fega_type=NodeVariable
4: label=Temperature
5: format=i4f8
6: index_byte=4
[Data]
0 0 1.000000e+02
1 0 1.000000e+02
3 0 1.000000e+02
58 0 1.000000e+02
59 0 1.000000e+02
60 0 1.000000e+02
. . . .
. . . .
. . . .
```

[Data]フィールドは左から順に、節点番号、方向成分、温度を示している。ただし、方向成分に関しては、熱伝導解析では1節点あたりの温度の自由度が1のために、構造解析のように方向成分を指定する必要がないため、全境界条件で0と指定しておく。

B.2. 熱流束規定境界条件

```
[Properties]
1: content_type=FEGenericAttribute
2: num_items=81
3: fega_type=NodeVariable
4: label=HeatFlux
5: format=i4f8
6: index_byte=4
[Data]
0 0 0.000000e+00
1 0 3.333333e+01
3 0 0.000000e+00
58 0 6.666666e+01
59 0 6.666666e+01
60 0 6.666666e+01
. . . .
. . . .
. . . .
```

[Data]フィールドは左から順に、節点番号、方向成分、熱流束を示している。ここでの熱流束は面指定に対する熱流束(単位面積あたりの値で指定)から節点集中熱流束に変換したものである。具体的には以下のような変換を想定してある。この面に面分布熱流束 q が与えられている時、各節点での節点集中熱流束は面の面積を S とすれば、4 面体 2 次要素の場合は以下のように表すことができる。

- P0 $q_0 = 0$
- P1 $q_1 = 0$
- P2 $q_2 = 0$
- P3 $q_3 = q \times S/3$
- P4 $q_4 = q \times S/3$
- P5 $q_5 = q \times S/3$

B.3. 熱伝達規定境界条件

```
[Properties]
1: content_type=FEGenericAttribute
2: num_items=8
3: fega_type=ElementVariable
4: label=HeatConvection
5: format=i4f8f8
6: index_byte=4
[Data]
0 1 1.000000e+02 1.231002e+02
5 3 1.000000e+02 1.231002e+02
. . . .
. . . .
. . . .
```

[Data]フィールドは左から順に、要素番号、面番号、外部接触温度、熱伝達係数を示している。ここで、面番号とは各要素を形成する面にそれぞれ与えられるものであり、対する頂点の節点番号と等しくなるような番号付けを行っている。例えば、面番号0に対する頂点の番号は0である。

B.4. 熱ふく射規定境界条件

```
[Properties]
1: content_type=FEGenericAttribute
2: num_items=8
3: fega_type=ElementVariable
4: label=HeatRadiation
5: format=i4f8f8f8
6: index_byte=4
[Data]
0 1 1.000000e+02 1.000000e+00 1.000000e+00
5 3 1.000000e+02 1.000000e+00 1.000000e+00
. . . .
. . . .
. . . .
```

[Data]フィールドは左から順に、要素番号、面番号、放射源の温度、放射率、形状係数を示している。

B.5. 物性値

(1). 単一材料モデルの例

```
HeatConductivity 200
Density 10.0
SpecificHeat 100.0
StefanBoltzmanConstant 5.67e-6
InternalHeatGeneration 0.0
```

上から順に、熱伝導率、密度(非定常解析時)、比熱(非定常解析時)、ステファ
ン・ボルツマン定数(熱ふく射解析時)、内部発熱である。

(2). 複数材料モデルの例

```
#materialInfo
materialN 2
propertyN 5
HeatConductivity 100
Density 5000
SpecificHeat 41.78
StefanBoltzmanConstant 5.67e-6
InternalHeatGeneration 0.0
HeatConductivity 50
Density 2500
SpecificHeat 20.0
StefanBoltzmanConstant 5.67e-6
InternalHeatGeneration 0.0
#volumeInfo
volumeN 2
1
0
```

C. ツール類について

C.1. 一体型解析モデルファイル作成ツール *makefem_thermal*

makefem_thermal は、熱伝導解析用の一体型解析モデルファイル作成ツールである。以下の境界条件に対応している。

1. 温度
2. 熱流束
3. 熱伝達
4. 熱ふく射

入力ファイル

メッシュデータファイル (拡張子 *msh*)
 メッシュ表面データファイル (拡張子 *fgr*)
 境界条件ファイル (拡張子 *cnd*)
 物性値ファイル (拡張子 *dat*)

出力ファイル

一体型解析モデルファイル (拡張子 *adv*)

以下に実行方法を示す。

```
% makefem_thermal mshFile fgrFile cndFile matFile advFile [options]
```

mshFile : メッシュデータファイル (入力)
fgrFile : メッシュ表面データファイル (入力)
cndFile : 解析条件ファイル (入力)
matFile : 物性値ファイル (入力)
advFile : 一体型解析モデルファイル (出力)

以下に、解析条件ファイル(*cndFile*)のフォーマットを示す。

<code>gravity 0 0 0</code>	← 実際には使用しない、見せかけの値
<code>boundary 2</code>	← 境界条件のデータ数
<code>tempOnFaceGroup 0 1 10</code>	← 面グループ 0 の温度が 10[°C]
<code>fluxOnFaceGroup 5 1 100</code>	← 面グループ 5 の熱流束が 100[W/m ²]

`tempOnFaceGroup` は表面グループの温度規定を表す。

`fluxOnFaceGroup` は表面グループの熱流束規定を表す。

ADVENTURE_Thermal ver.2.0 以降に含まれる *makefem_thermal* では、熱伝達規定境界条件と熱ふく射規定境界条件にも対応している。以下に、熱伝達規定境界条件の例を示す。

<code>gravity 0 0 0</code>	← 実際には使用しない、見せかけの値
<code>boundary 2</code>	← 境界条件のデータ数
<code>transOnFaceGroup 0 10.0 100.0</code>	← 表面グループ 0 の周囲温度が 10[°C] 熱伝達係数が 100 (W/m ² .°C)

Option:

-with-surface	表面情報を advFile に保存する
-elm-src-file file name	ファイルから全要素の内部発熱量を入力

発熱量ファイルは以下のように設定する。

```
#HeatSourceInfo
SourceN 34
6.95e+04
--
--
--
```

C.2. pfemsolv

pfemsolv は、部分領域内部の未知数を解くツールである。各部分領域の境界条件は、外部表面とインターフェース条件である。このツールは、解析モデル入力の model ディレクトリからメッシュデータを読み込み、result_surface ディレクトリから境界条件を読み込む。

以下に実行方法を示す。

```
% mpirun [options for mpirun] pfemsolv data_dir
```

入力ファイル：
 領域分割型解析モデルファイル：
 data_dir/model/advhddm_in_P.adv

境界条件：
 data_dir/result_surface/advhddm_out_P.adv

解析結果（定常解析）：
 data_dir/result/advhddm_out_P.adv

D. Libsparse**LIBSPARSE-0.2b**

libsparse は、疎行列を扱う C ライブラリであり、特に様々な疎行列形式を用いて行列とベクトルの掛け算を行う。libsparse を用いて、skyline 形式から他の疎行列形式に変換できる。その逆操作も同様にできる。現在、このライブラリは ADVENTURE システムのいくつかのモジュールの対称行列のみをサポートしている。

libsparse の関数を使う際に、“advlas.h” をインクルードする必要がある。

libsparse の特徴

1. skyline 形式から他の疎行列形式に変換できる。その逆操作も同様にできる。
2. 疎行列とベクトルの掛け算を行える($y += Ax$)。
3. ADVENTURE システムに基づき、いくつかの ADVENTURE モジュールのソースファイル内のファイルに依存する。
4. ADVENTURE_Thermal と ADVENTURE_Solid の対称行列のみ利用できる。
5. 節点コネクティビティ情報を用いて異なる疎行列のインデックスを作成することができる。

認識されるマトリクス形式[10]は以下に示す：

CSR: Compressed Sparse Row

CSC: Compressed Sparse Column

COO: Coordinate format

DCOO: Diagonal Coordinate format

MSR: Modified Compressed Sparse Row

SKY: Skyline format

ICSR: Incremental Compressed Sparse Row

VBCSR: Variable Block Compressed Sparse Row

DBCSPR: Diagonal Block Compressed Sparse Row (ADVENTURE_Solid 用)

基本的な関数：

libsparse ライブラリで使用する基本的な関数を下の表に示す。

NewDMatrix()	疎形式のインデックスを定義する。
advlas_mkindex()	部分領域のメッシュデータを読み込み、疎形式のインデックスを準備する。異なる疎形式には、異なる関数を使用する。
advlas_cpmat_sky2nz()	skyline 形式から他の形式に変換する。異なる疎形式には、異なる関数を使用する。
advlas_ldl_decomposit()	skyline 行列を ldl 形式に分解する。
advlas_matmult_vec_add()	行列とベクトルの掛け算を行い、その結果を他のベクトルに保存する。
advlas_ldl_solve()	前進消去と後退代入法を用いて線形方程式を解く。

上記の関数について以下に述べる：

Dmatrix を作成する関数

NewDMatrix(matdim, node_dim, matrix type)

Output: dmat

matrix index を作成する関数**advlas_mkindex**(nnd, nel, nd_elm, nop, node_dim, dmat)

Output: dmat

Skyline 行列を非零成分のみ形式にコピーする関数**advlas_cpmat_sky2nz**(skymat, nzmat, precon_sw)

Output: nzmat

行列とベクトルの掛け算を行う関数**advlas_matmult_vec_add**(dmat, temp, reac)

Output: reac

skyline 行列を ldl 形式に分解する関数**advlas_ldl_decompose**(dmat, work)

Output: dmat (decomposed form)

ldl 形式の skyline 行列を解く関数**advlas_ldl_solve**(dmat, solution)

Output: solution

Dmatrix を削除する関数**DeleteDMatrix**(&dmat)引数 :int **nnd** – 節点数int **nel** – 要素数int **nd_elm** – 要素ごとの節点数int **nop** – 要素内の節点コネクティビティint **node_dim** – 各節点における未知数の数 (Thermal は 1、Solid は 3)int **precon_sw** – 前処理のスイッチint ***temp** – 入力ベクトルint ***reac** – 出力ベクトルstructure DMatrix **dmat** - 行列インデックスと値の構造体structure DMatrix **skymat** - skyline 行列インデックスと値の構造体structure DMatrix **dmat** - 非零成分のみの行列インデックスと値の構造体

実行オプション

- **-keep-kmat-csr** 部分領域の行列を CSR 形式で保存する。デフォルトはこの形式である。

- `-keep-kmat-coo` 部分領域の行列を COO 形式で保存する。
- `-keep-kmat-msr` 部分領域の行列を MSR 形式で保存する。
- `-keep-kmat-csc` 部分領域の行列を CSC 形式で保存する。
- `-keep-kmat-icmr` 部分領域の行列を ICSR 形式で保存する。
- `-keep-kmat-vbcsr` 部分領域の行列を VBCSR 形式で保存する。
- `-keep-kmat-dcoo` 部分領域の行列を DCOO 形式で保存する。
- `-keep-kmat-sky` 部分領域の行列を SKYLINE 形式で保存する。

疎行列格納形式

疎行列をメモリ上に保持する場合、値がゼロである成分が多いという点を利用した、特別な形式を用いる必要がある。本モジュールでは、対称な疎行列を扱うため、下三角部分のみを保持する。メモリ使用量と計算時間を考慮しながら非零成分(*nnz*)のみを保持することを目的とした、いくつかの格納形式を以下に示す。

Compressed Sparse Row (CSR)

Compressed Sparse Row (以下”CSR”)は、疎行列のためのもっとも一般的な格納形式である。行列は *data*, *col_ind*, *row_ptr* の 3 つの配列変数で表現される。

$$A = \begin{bmatrix} 1.0 & 2.0 & 0.0 & 6.0 & 0.0 \\ 2.0 & 3.0 & 4.0 & 7.0 & 9.0 \\ 0.0 & 4.0 & 5.0 & 0.0 & 0.0 \\ 6.0 & 7.0 & 0.0 & 8.0 & 10.0 \\ 0.0 & 9.0 & 0.0 & 10.0 & 11.0 \end{bmatrix}$$

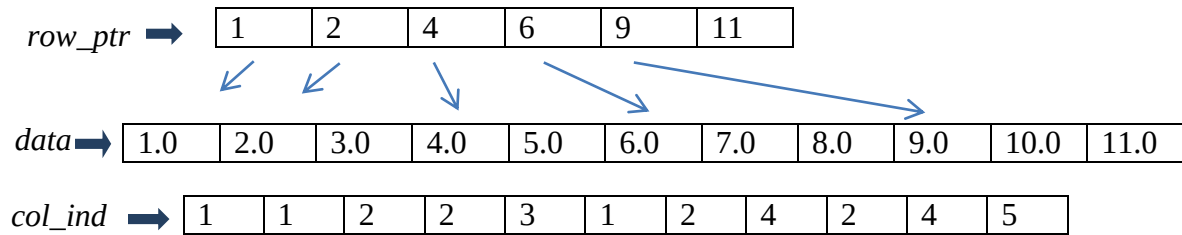
data : 長さ *nnz* の実数配列で、行列 *A* の非零成分の値を行単位で並べたものである。

col_ind : 長さ *nnz* の整数配列で、配列 *data* の各要素の列番号を表す。

row_ptr : 行列 *A* の行数+1 を長さとする整数配列で、各行の最初の成分が *data* および *col_ind* のどの位置かを表す。

row_ptr の最後の成分は *data* および *col_ind* の長さと同じ値を代入する。こうすることで、*i* 行目の非零成分の数が *row_ptr*[*i*+1] - *row_ptr*[*i*] で求められる。

CSR 形式で対称行列 A を表現する場合の例を以下に示す。



Coordinate Storage (COO)

もっとも単純な疎行列の表現形式が Coordinate Storage (以下”COO”)である。COO では、いずれも長さ nnz の、 $data$ 、 row_ind および col_ind の 3 つの配列を使用する。

$data$: 非零成分の値を表す。

row_ind : 対応する要素の行列中の位置(行番号)を表す

col_ind : 対応する要素の行列中の位置(列番号)を表す。

COO 形式で対称行列 A を表現する場合の例を以下に示す。

<i>data</i>	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0
<i>row_ind</i>	1	2	2	3	3	4	4	4	5	5	5
<i>col_ind</i>	1	1	2	2	3	1	2	4	2	4	5

対角成分を最初に保持する形式は Diagonal Coordinate Storage (DCOO)と呼ばれる。

Compressed Sparse Column (CSC)

Compressed Sparse Column (以下”CSC”)は、CSR 形式と似ているが、列単位で値を保持する。行列は $data$ 、 row_ind 、 col_ptr の 3 つの配列変数で表現される。

$data$: 長さ nnz の実数配列で、行列 A の非零成分の値を列単位で並べたものである。

row_ind : 長さ nnz の整数配列で、配列 $data$ の各要素の行番号を表す。

col_ptr : 行列 A の列数+1 を長さとする整数配列で、各列の最初の成分が $data$ および row_ind のどの位置かを表す。

col_ptr の最後の成分は $data$ および row_ind の長さと同じ値を代入する。こうすることで、 i 列目の非零成分の数が $col_ptr[i+1] - col_ptr[i]$ で求められる。

CSC 形式で対称行列 A を表現する場合の例を以下に示す。

<i>col_ptr</i>	1	4	8	9	11	11
----------------	---	---	---	---	----	----

<i>data</i>	1.0	2.0	6.0	3.0	4.0	7.0	9.0	5.0	8.0	10.0	11.0
<i>row_ind</i>	1	2	4	2	3	4	5	3	4	5	5

Modified Sparse Row (MSR)

Modified sparse row (以下”MSR”)は、いずれも長さ($Nz+1$)の 2 つの配列 (実数配列 *data* と整数配列 *col_ind*) からなる。

行列のサイズを n としたとき、配列 *data* の最初の n 個の要素は、行列の対角成分の値を表す。 $n+1$ 番目の要素は使われないが、行列に関する情報を格納することもできる。 $n+2$ 番目以降の要素は、対角成分以外の非零成分を行単位に並べて表す。配列 *col_ind* の最初の $n+1$ 個の要素は 配列 *data* 中の行の先頭要素を表す。残りの部分は、配列 *data* の対応する要素の列番号を表す。

MSR 形式で対称行列 A を表現する場合の例を以下に示す。

<i>data</i>	1.0	3.0	5.0	8.0	11.0		2.0	4.0	6.0	7.0	9.0	10.0
<i>col_ind</i>	1	2	4	6	9	11	1	2	1	2	2	4

MSR 形式を使用できるのは、係数行列の対角成分が非零のときだけという制限がある。

Variable Block Sparse Row (VBR)

Variable Block Sparse Row (以下”VBR”)形式の考えは、隣り合う非零成分をまとめてしまう(ブロック化)というものである。固定サイズのブロック化とは異なり、ブロックのサイズは可変である。固定サイズのブロックを使用した場合と同様、ブロックの最初の非零成分の列番号がわかれば、他の成分の列番号もわかる。VBR 形式では、CSR 形式で使用する 3 つの配列のほかに、ブロック数を長さとする整数配列 *nz_ptr* が必要である。

data : 長さ nnz の実数配列で、行列 A の非零成分の値を行単位で並べたものである。

col_ind : ブロック数を長さとする整数配列で、ブロックの最初の非零成分の列番号を表す。

row_ptr : 行列 A の行数を長さとする整数配列で、行の最初に位置するブロックを表す。*row_ptr* の最後の成分はブロック数を表す。

nz_ptr : 配列 *data* 中のブロックの最初の成分の位置を表す。

VBR 形式で対称行列 A を表現する場合の例を以下に示す。

<i>nz_ptr</i>	1	2	4	6	8	9	10				
<i>data</i>	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0

<i>col_ind</i>	1	1	2	1	4	2	4
<i>row_ptr</i>	1	2	3	4	6	7	

Incremental Compressed Sparse Row (ICSR)

Incremental Compressed Sparse Row (以下”ICSR”)は、MSR 形式の拡張形式である。下三角行列を 1 次元配列に並べ替えた上で、一つ前の非零成分の位置との差を記憶する。ICSR 形式では下記の 2 つの配列を使用する。

data : 長さ $nnz + 1$ の実数配列で、非零成分の値を表す。

incr : 長さ $nnz + 1$ の整数配列で、一つ前の非零成分の位置との差を表す。

ICSR 形式で対称行列 A を表現する場合の例を以下に示す。

<i>data</i>	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0
<i>incr</i>	0	1	1	2	1	1	1	2	2	2	1

Skyline or Variable Band (SKY)

Skyline はピボット交換が不要な場合に直接法ソルバで一般的に使われる形式である。対称行列の場合は、下三角行列のみ記憶することで、使用メモリ量を半分にすることができる。行列は *data*, *col_ind*, *row_ptr* の 3 つの配列変数で表現される。

data : 行単位で成分を表す。

col_ind : 各行の最初の非零成分の列番号を表す。

row_ptr : 配列 *data* 中の各行の最初の成分の位置を表す。

Skyline 形式で対称行列 A を表現する場合の例を以下に示す。

<i>row_ptr</i>	1	2	4	6	9							
<i>data</i>	1.0	2.0	3.0	4.0	5.0	6.0	7.0	0.0	8.0	9.0	0.0	10.0
<i>col_ind</i>	1	1	2	1	2							

これまでの形式と違い、Skyline 形式では零成分の一部も記憶する。

Diagonal Block Compressed Sparse Row (DBCSPR)

Diagonal Block Compressed Sparse Row (以下”DBCSPR”)形式の考えは、固定サイズのブロック化した行列というものである。構造問題にはこのタイプの行列がある。サポートするブロックの形状は 3×3 である。DBCSPR 形式では、一つのブロック内の最初の非零成分の列番号がわかれば、他の成分の列番号もわかる。

diag: 実数配列で、対角ブロックの非零成分の値を行単位で並べたものである。

data: 実数配列で、3x3 ブロック行列の値を行単位で並べたものである。

index_brow: 整数配列で、最初のブロックの最初成分の列番号を表す。

index_bcol: 整数配列で、各 3x3 ブロックの最初成分の列番号を表す。長さはブロック数を表す。

DBCSR 形式で対称行列 A を表現する場合の例を以下に示す。

```

1
2  3
4  5  6
7  8  9 10
11 12 13 14 15
16 17 18 19 20 21
                22 23 24 25
                26 27 28 29 30
                31 32 33 34 35 36
37 38 39                                40
41 42 43                                44 45
46 47 48                                49 50 51

```

diag	1	2	3	4	5	6	10	14	15	19	20	21	25	29	30	34	35	36	40
-------------	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----

44	45	49	50	51
----	----	----	----	----

data	7	8	9	11	12	13	16	17	18	22	23	24	26	27	28	31	32	33
-------------	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

37	38	39	41	42	43	46	47	48
----	----	----	----	----	----	----	----	----

index_brow	0	0	9	18	27
-------------------	---	---	---	----	----

index_bcol	0	3	0
-------------------	---	---	---

E. 解析例

ADVENTURE_Thermal による簡単な解析例を挙げる。簡略のため、熱伝導問題に限定し、4 面体 2 次要素を使用した。こちらのモデルは、sample_data/manual_example ディレクトリの下にある。

E.1. 温度規定境界条件による解析例

問題定義

図 12 のような炉壁の温度分布を解析する。外側の幅は $W=2.0\text{m}$ で壁厚 L は 0.5m である。炉は耐火れんが出てきており、熱伝導率 k は $40\text{ W/(m}\cdot^{\circ}\text{K)}$ である。内壁部と外壁部にそれぞれ $t_i = 600^{\circ}\text{C}$ and $t_o = 60^{\circ}\text{C}$ の一様な温度分布を設定する。

解析手順

1. 形状定義ファイルと節点密度ファイルの作成

temp.gm3d

Box	0	0	0	2	2	2
Box	0.5	0.5	2	1	1	2
subtract						

2. 表面パッチ作成 (ADVENTURE_CAD)

```
% advcad temp.gm3d temp.pch 0.1
```

3. メッシュ生成 (ADVENTURE_TetMesh)

- 3a. 表面パッチ修正

```
% advtmesh9p temp
```

- 3b. メッシュ生成(4 面体 1 次要素)

```
% advtmesh9m tempc
```

- 3c. 4 面体 2 次要素への変換

```
% advtmesh9s tempc
```

4. メッシュ表面の抽出 (ADVENTURE_BCtool-2.0)

```
% msh2pch tempcs.msh 3
```

5. 境界条件設定 (ADVENTURE_BCtool-2.0)

```
% ADVENTURE_BcGUI_Ver_2_0 tempcs_3.pch tempcs_3.pcg
```

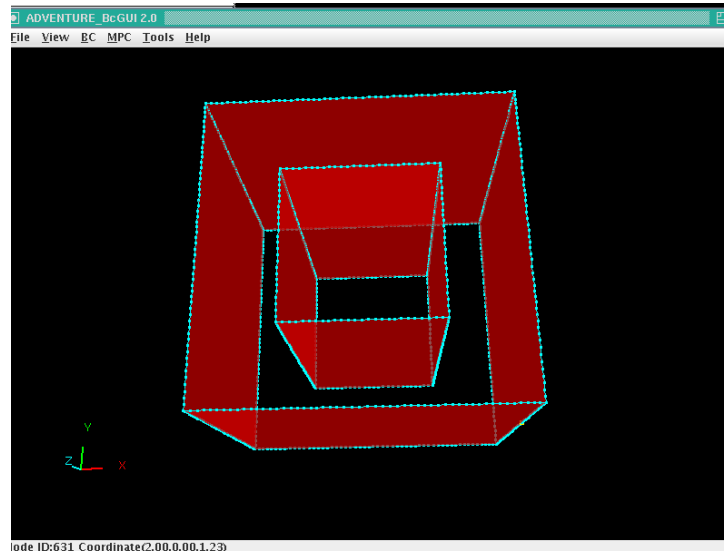


図 12. 温度規定境界条件の設定例

ADVENTURE_BcGUI_2_0 の出力ファイル
temp.cnd

```
gravity 0.0 0.0 0.0
boundary 8
tempOnFaceGroup 0    0    0    60.0
tempOnFaceGroup 1    0    0    60.0
tempOnFaceGroup 4    0    0    60.0
tempOnFaceGroup 5    0    0    600.0
tempOnFaceGroup 6    0    0    600.0
tempOnFaceGroup 7    0    0    600.0
tempOnFaceGroup 8    0    0    600.0
tempOnFaceGroup 9    0    0    60.0
```

6. 物性値ファイルの作成

s_mat.dat

```
HeatConductivity 40
```

7. 一体型解析モデルファイルの作成 (ADVENTURE_BCtool-2.0)

```
% makefem3 tempcs.msh tempcs_3.fgr temp.cnd s_mat.dat temp.adv
(temp.adv が出力される)
```

8. 領域分割型解析モデルファイルの作成 (ADVENTURE_Metis)

```
% mpirun -np 4 adventure_metis -difn 1 temp.adv temp 700
("temp"ディレクトリ内に分割データが保存される)
```


9. 熱解析 (ADVENTURE_Thermal)

```
% mpirun -np 4 advthermal-p temp
```

(“temp”ディレクトリ内に”result”ディレクトリが作成される)

10. 温度データファイルの作成

```
% hddmmrg Temperature temp
```

(Temperature.dat ファイルが出力される)

11. 温度分布の可視化 (ADVENTURE_Auto)

```
% advauto_thermalview tempcs.msh tempcs_3.fgr
```

(Temperature.dat ファイルが自動的に読み込まれる)

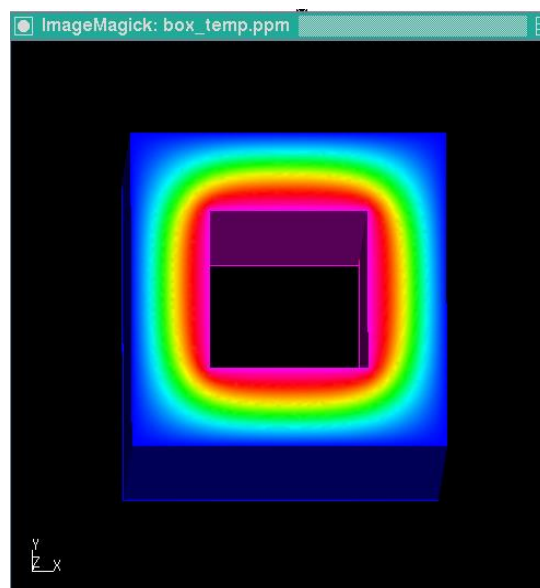


図 13. Advauto_thermalview による温度分布の可視化結果

E.2. 熱流束規定境界条件の解析例

問題定義

図 14 のような円筒の断面の熱伝達を考える。熱伝導率は $k = 50 \text{ [W/mm}\cdot\text{K]}$ 、円筒の内径は 125 mm、外径は 250 mm である。外側表面の温度は $T_1 = 10 \text{ [}^\circ\text{C]}$ で一定とし、内部表面では熱が $100 \text{ [W/mm}^2\text{]}$ で流れるものとする。他の表面は自然境界条件のままとする。以上の条件で ADVENTURE_Thermal を使って定常温度分布を求める。

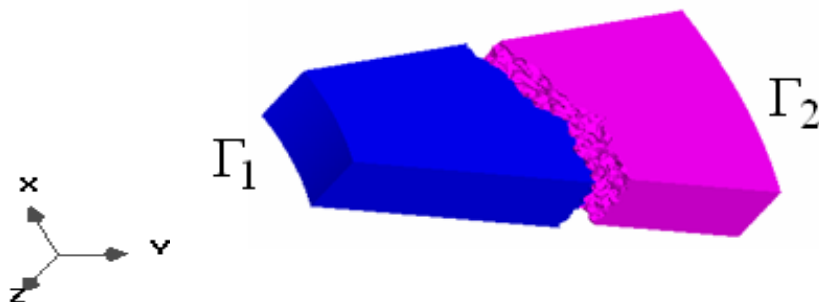


図 14. 解析モデル (円筒の断面)

解析手順

1. 形状定義ファイルと節点密度ファイルの作成

flux.igs
(CAD を使用して作成)

flux.ptn

BaseDistance 4.0

2. 表面パッチファイルの作成 (ADVENTURE_TriPatch)

```
% ADVENTURE_TriPatch flux flux
```

3. メッシュ作成 (ADVENTURE_TetMesh)

- 3a. 表面パッチ修正

```
% advtmesh9p flux
```

- 3b. メッシュ生成 (4 面体 1 次要素)

```
% advtmesh9m fluxc
```

3c. 4 面体 2 次要素への変換

```
% advtmesh9s fluxc
```

4. メッシュ表面データの抽出 (ADVENTURE_BCtool-2.0)

```
% msh2pch fluxcs.msh 3
```

5. 境界条件設定 (ADVENTURE_BCtool-2.0)

```
% ADVENTURE_BcGUI_Ver_2_0 fluxcs_3.pch fluxcs_3.pcg
```

ADVENTURE_BcGUI_2_0 の出力ファイル

```
flux.cnd
```

```
gravity 0.0 0.0 0.0
boundary 2
tempOnFaceGroup 1 0 0 10
fluxOnFaceGroup 4 0 0 100
```

6. 物性値ファイルの作成

```
s_mat.dat
```

```
HeatConductivity 40
```

7. 一体型解析モデルファイルの作成 (ADVENTURE_BCtool-2.0)

```
% makefem3 fluxcs.msh fluxcs_3.fgr flux.cnd s_mat.dat flux.adv
(flux.adv が出力される)
```

8. 領域分割型解析モデルファイルの作成 (ADVENTURE_Metis)

```
% mpirun -np 2 adventure_metis -difn 1 flux.adv flux 700
(“flux”ディレクトリ内に分割データが保存される)
```

9. 熱解析 (ADVENTURE_Thermal)

```
% mpirun -np 2 advthermal-p flux
(“flux”ディレクトリ内に”result”ディレクトリが作成される)
```

10. 温度データファイルの作成

```
% hddmmrg Temperature flux
(Temperature.dat ファイルが作成される)
```

11. 温度分布の可視化 (ADVENTURE_Auto)

```
% advauto_thermalview fluxcs.msh fluxcs_3.fgr
(Temperature.dat ファイルが自動的に読み込まれる)
```

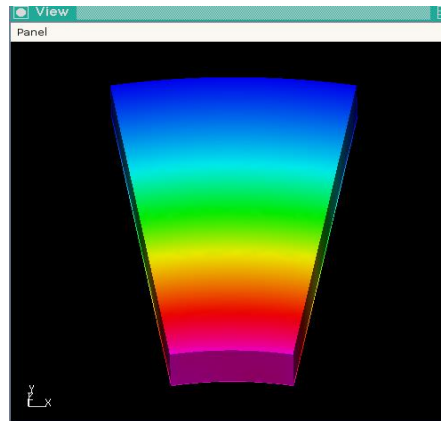


図 15. Advauto_thermalview による温度分布の可視化結果

E.3. 熱伝達規定境界条件による解析列

問題定義

熱伝導率が $8.64 \times 10^{-2} \text{ W/(mm} \cdot \text{K)}$ の円筒の断面の熱伝導を考える。円筒内部の媒体の温度は 38°C 、媒体と内部表面の間の熱伝達係数は $2.8372 \times 10^{-3} \text{ W/(mm}^2 \cdot \text{K)}$ である。円筒外部の媒体の温度は -18°C 、媒体と外部表面の間の熱伝達係数は $1.4186 \times 10^{-3} \text{ W/(mm}^2 \cdot \text{K)}$ である。以上の条件で ADVENTURE_Thermal を使って定常温度分布を求める。

解析手順

1. 形状定義ファイルと節点密度ファイルの作成

conv.igs
(CAD を使用して作成)

conv.ptn

BaseDistance 4.0

2. 表面パッチファイルの作成 (ADVENTURE_TriPatch)

```
% ADVENTURE_TriPatch conv conv
```

3. メッシュ作成 (ADVENTURE_TetMesh)

3a. 表面パッチ修正

```
% advtmesh9p conv
```

3b. メッシュ生成 (4 面体 1 次要素)

```
% advtmesh9m convc
```

3c. 4 面体 2 次要素への変換

```
% advtmesh9s convc
```

4. メッシュ表面データの抽出 (ADVENTURE_BCtool-2.0)

```
% msh2pch convcs.msh 3
```

5. 境界条件設定 (ADVENTURE_BCtool-2.0)

```
% ADVENTURE_BcGUI_Ver_2_0 convcs_4.pch convcs_3.pcg
```

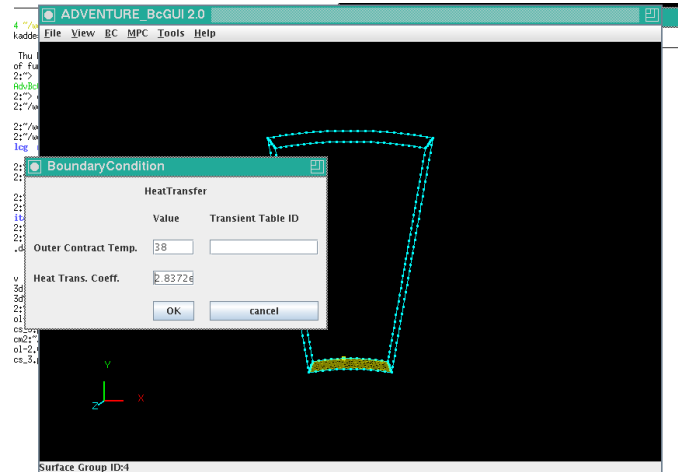


図 16. 熱伝達規定境界条件の設定例

ADVENTURE_BcGUI_2_0 の出力ファイル
conv.cnd

```
gravity 0.0 0.0 0.0
boundary 2
transOnFaceGroup 1 -18 .0014186
transOnFaceGroup 4 38 .0028372
```

6. 物性値ファイルの作成

s_mat.dat

```
HeatConductivity 0.086475
```

7. 一体型解析モデルファイルの作成 (ADVENTURE_BCtool-2.0)

```
% makefem3 convcs.msh convcs_3.fgr conv.cnd s_mat.dat conv.adv
```

(conv.adv が出力される)

8. 領域分割型解析モデルファイルの作成 (ADVENTURE_Metis)

```
% mpirun -np 2 adventure_metis -difn 1 conv.adv conv 71
```

(“conv”ディレクトリ内に分割データが保存される)

9. 熱解析 (ADVENTURE_Thermal)

```
% mpirun -np 2 advthermal-p conv
```

(“conv”ディレクトリ内に”result”ディレクトリが作成される)

10. 温度データファイルの作成

```
% hddmmrg Temperature conv
```

(Temperature.dat ファイルが出力される)

11. 温度分布の可視化 (ADVENTURE_Auto)

% **advauto_thermalview** convcs.msh convcs_3.fgr
(Temperature.dat ファイルが自動的に読み込まれる)

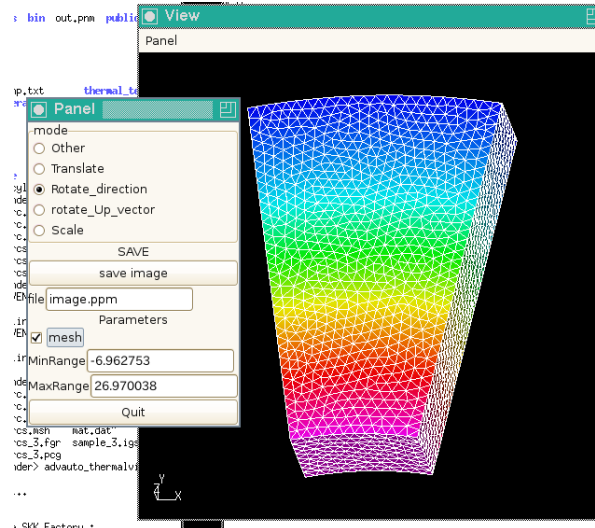


図 17. Advauto_thermalview による温度分布の可視化結果

E.4. 内部発熱の解析例

問題定義

半径 0.1m の円柱の熱伝導を考える。熱伝導率は $40 \text{ W/(m} \cdot ^\circ\text{K)}$ で、電流により熱エネルギー $4 \times 10^6 \text{ (W/m}^3)$ が発生する。熱は円柱表面から周囲の媒体(温度 20°C 一定)に発散していく。熱伝達係数は $400 \text{ W/(m}^2 \cdot ^\circ\text{K)}$ である。以上の条件で ADVENTURE_Thermal を使って定常温度分布を求める。

解析手順

1. 形状定義ファイルと節点密度ファイルの作成

igen.gm3d

```
circle 0 0 0.1 0 0 0 1 16
extrude 0 0 0.5
```

2. 表面パッチ作成 (ADVENTURE_CAD)

```
% advcad igen.gm3d igen.pch 0.01
```

3. メッシュ生成 (ADVENTURE_TetMesh)

3a. 表面パッチ修正

```
% advtmesh9p igen
```

3b. メッシュ生成(4 面体 1 次要素)

```
% advtmesh9m igenc
```

3c. 4 面体 2 次要素への変換

```
% advtmesh9s igenc
```

4. メッシュ表面の抽出 (ADVENTURE_BCtool-2.0)

```
% msh2pch igencs.msh 3
```

5. 境界条件設定 (ADVENTURE_BCtool-2.0)

```
% ADVENTURE_BcGUI_Ver_2_0 igencs_3.pch igencs_3.pcg
```

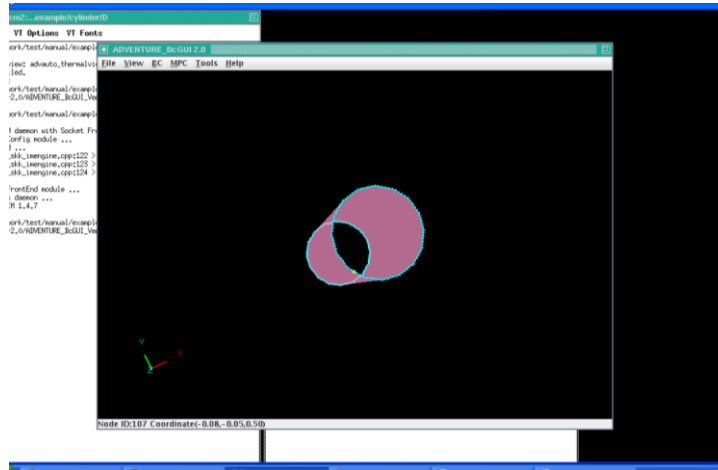



図 18. 熱伝達規定境界条件の設定例

ADVENTURE_BcGUI_2_0 の出力ファイル

igen.cnd

```
gravity 0.0 0.0 0.0
boundary 1
transOnFaceGroup 0    20    400
```

6. 物性値ファイルの作成

s_mat.dat

```
HeatConductivity 40
InternalHeatGeneration 4.0E6
```

7. 一体型解析モデルファイルの作成 (ADVENTURE_BCtool-2.0)

```
% makefem3 igencs.msh igencs_3.fgr igen.cnd s_mat.dat igen.adv
(igen.adv が出力される)
```

8. 領域分割型解析モデルファイルの作成 (ADVENTURE_Metis)

```
% mpirun -np 2 adventure_metis -difn 1 igen.adv igen 400
(“igen”ディレクトリ内に分割データが保存される)
```

9. 熱解析 (ADVENTURE_Thermal)

```
% mpirun -np 2 advthermal-p igen
(“igen”ディレクトリ内に”result”ディレクトリが作成される)
```

10. 温度データファイルの作成

```
% hddmmrg Temperature igen
(Temperature.dat ファイルが出力される)
```

11. 温度分布の可視化 (ADVENTURE_Auto)

% **advauto_thermalview** igencs.msh igencs_3.fgr
(Temperature.dat ファイルが自動的に読み込まれる)

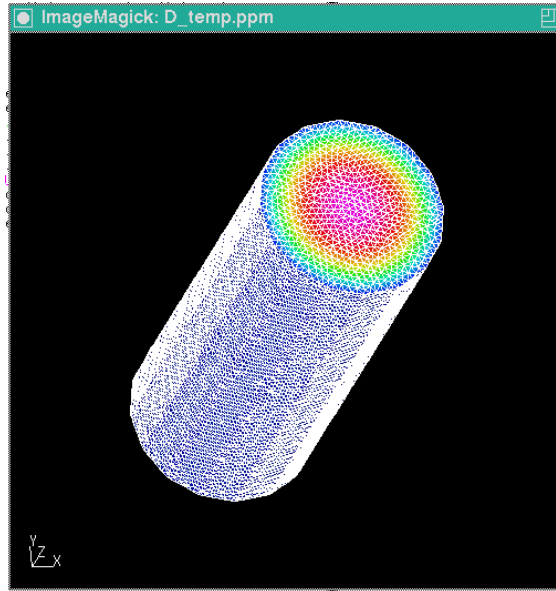


図 19. Advauto_thermalview による温度分布の可視化結果

E.5. 複数材料モデルの解析例

問題定義

核燃料棒の定常温度分布を計算する。燃料棒(Cylinder)の形状は半径 $R=15\text{mm}$ の細長い円柱に近似する。燃料棒は厚さ $L=3\text{mm}$ の鉄製の保護材(Tube)で覆われている。核燃料と鉄の熱伝導率はそれぞれ $k_f = 30 \text{ W}/(\text{m}\cdot^\circ\text{K})$ 、 $k_s = 36 \text{ W}/(\text{m}\cdot^\circ\text{K})$ とする。核燃料中の単位体積当たりの内部発熱量は $f = 4 \times 10^6 \text{ W}/\text{m}^3$ 、周囲媒体との熱伝達係数は $h_c = 100 \text{ W}/(\text{m}^2\cdot^\circ\text{K})$ 、周囲媒体の温度は $t_c = 300^\circ\text{C}$ である。

解析手順

1. 形状定義ファイルと節点密度ファイルの作成

Cylinder

cylinder.igs
(CAD を使用して作成)

cylinder.ptn

BaseDistance 3.0

Tube

tube.igs
(CAD を使用して作成)

tube.ptn

BaseDistance 3.0

2. 表面パッチファイルの作成 (ADVENTURE_TriPatch)

2a. 個別の表面パッチの作成

```
% ADVENTURE_TriPatch cylinder cylinder
% ADVENTURE_TriPatch tube tube
```

2b. `cylinder`と`tube`を結合して`multi`にする

```
% mrapach cylinder.pcm cylinder.pcg tube.pcm tube.pcg
-o multi.pcm -g multi.pcg
```

3. メッシュ作成 (ADVENTURE_TetMesh)

3a. 表面パッチ修正

```
% advtmesh9p multi
```

3b. メッシュ生成 (4 面体 1 次要素)

```
% advtmesh9m multi
```

3c. 4 面体 2 次要素への変換

```
% advtmesh9s multi
```

4. メッシュ表面データの抽出 (ADVENTURE_BCtool-2.0)

```
% msh2pch multics.msh 4
```

5. 境界条件設定 (ADVENTURE_BCtool-2.0)

5a. 複数部材モデルの表示

```
% msh2pcm multics.msh
```

```
% ADVENTURE_BcGUI_Ver_2_0 multics_V.pcm multics_V.pcg
```

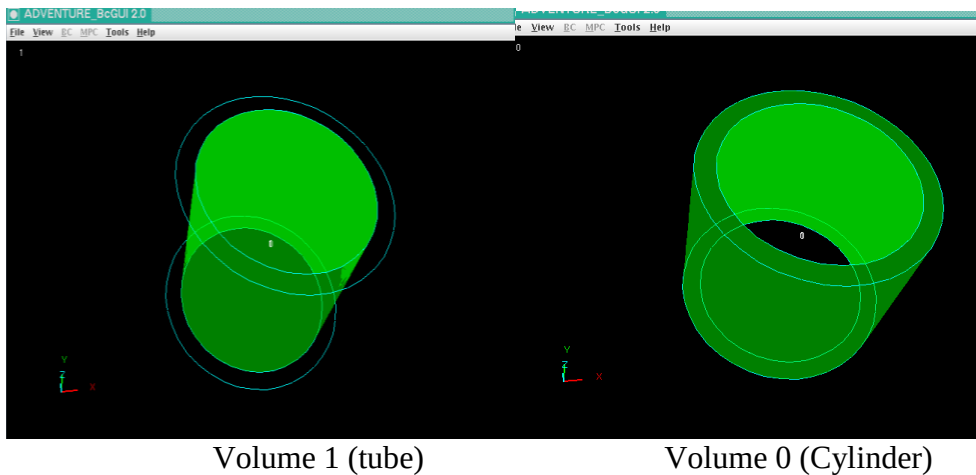


図 20. 複数部材の表示

5b. 境界条件の設定

```
% ADVENTURE_BcGUI_Ver_2_0 multics_4.pch multics_4.pcg
```

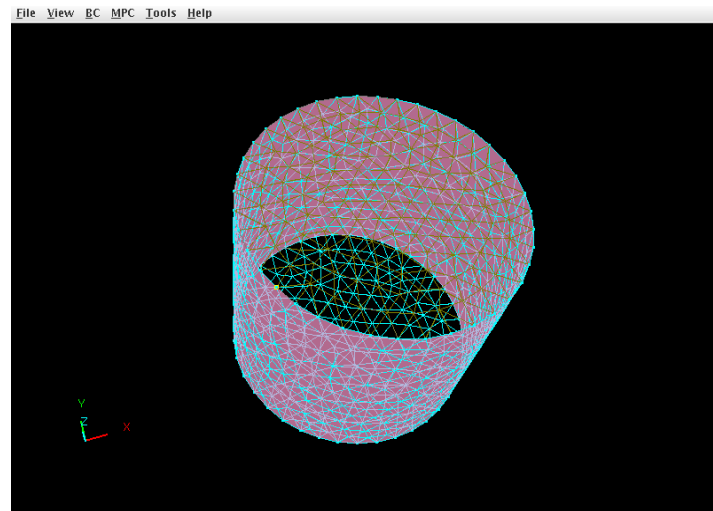


図 21. 熱伝達規定境界条件の設定例

ADVENTURE_BcGUI_2_0 の出力ファイル
multi.cnd

```
gravity 0.0 0.0 0.0
boundary 1
transOnFaceGroup 0    300    100e-6
```

6. 物性値ファイルの作成

m_material.dat

```
#materialInfo
materialN 2
propertyN 2
HeatConductivity 36.0e-3
InternalHeatGeneration 0.0
HeatConductivity 30.0e-3
InternalHeatGeneration 4.0e-3
#volumeInfo
volumeN 2
0
1
```

7. 一体型解析モデルファイルの作成 (ADVENTURE_BCtool-2.0)

```
% makefem3    multics.msh    multics_4.fgr    multi.cnd    m_material.dat
multi.adv
(multi.adv が出力される)
```

8. 領域分割型解析モデルファイルの作成 (ADVENTURE_Metis)

```
% mpirun -np 2 adventure_metis -difn 1 multi.adv multi 40
```

(“multi”ディレクトリ内に分割データが保存される)

9. 熱解析 (ADVENTURE_Thermal)

```
% mpirun -np 2 advthermal-p multi
```

(“multi”ディレクトリ内に”result”ディレクトリが作成される)

10. 温度データファイルの作成

```
% hddmmrg Temperature multi
```

(Temperature.dat ファイルが出力される)

11. 温度分布の可視化 (ADVENTURE_Auto)

```
% advauto_thermalview multics.msh multics_4.fgr
```

(Temperature.dat ファイルが自動的に読み込まれる)

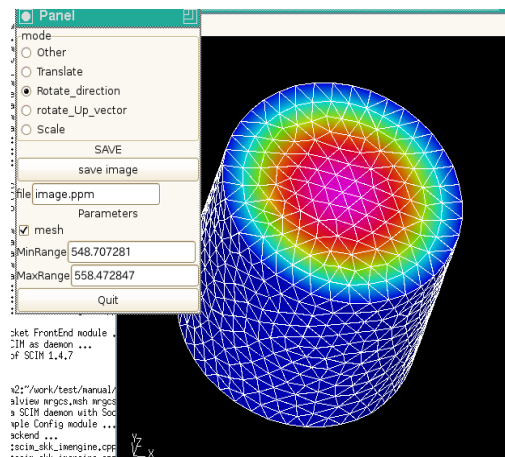


図 22. Advauto_thermalview による温度分布の可視化結果

E.6. 表面情報を扱うモデルの解析例

問題定義

図 23 のような冷却ステープの一部の定常温度分布を計算する。計算条件は以下のように示す。

空気の温度は 50°C, 水の温度は 30°C, 高温ガスの温度は 1600°C とする。

熱対流係数

炉シェルと大気との間：12 W/(m²K)

水と冷却ステープ内側との間：8000 W/(m²K)

高温ガスと炉シェルとの間：260 W/(m²K)

熱伝導率：

炉シェルとステープ本体：52.2 W/(m K)

充填材料：0.35 W/(m K)

無効なレンガやライニング材：21 W/(m K)

なお、スラグの形成が考慮されていない。

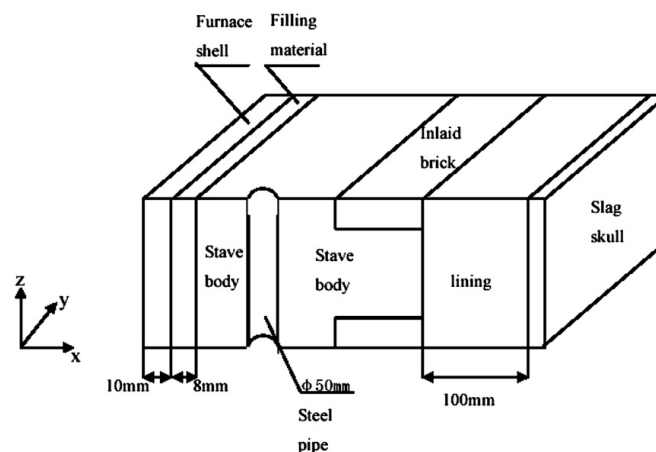


図 23. 冷却ステープの一部

解析手順

下の 1-3 ステップに従って、上記の複数材料モデルの解析用メッシュファイル partstavecs.msh を作成できる。

1. メッシュ表面データの抽出 (ADVENTURE_BCtool-2.0)

```
% msh2pch multics.msh 3
```

2. 境界条件設定 (ADVENTURE_BCtool-2.0)

```
% ADVENTURE_BcGUI_Ver_2_0 partstavecs_3.pch partstavecs_3.pcg
```

ADVENTURE_BcGUI_2_0 の出力ファイル

thermal.cnd

```
gravity 0.0 0.0 0.0
boundary 3
transOnFaceGroup 4    30.0    0.0080
transOnFaceGroup 6    1600    2.6E-4
transOnFaceGroup 7    50.0    0.12E-4
```

3. 物性値ファイルの作成

m_material.dat

```
materialN 3
propertyN 1
HeatConductivity 52.2e-3
HeatConductivity 0.35e-3
HeatConductivity 21e-3
volumeN 7
0
2
2
2
2
1
0
```

4. 一体型解析モデルファイルの作成 (makfem_thermal)

```
% makefem_thermal partstavecs.msh partstavecs_4.fgr thermal.cnd
m_material.dat partstave.adv -with-surface
```

5. 領域分割型解析モデルファイルの作成 (ADVENTURE_Metis)

```
% mpirun -np 4 adventure_metis -difn 1 partstave.adv partstave 200
(“partstave” ディレクトリ内に分割データが保存される)
```

6. 熱解析 (ADVENTURE_Thermal)

```
% mpirun -np 4 advthermal-p -result-surface-interface partstave
(“partstave” ディレクトリ内 “result_surface” ディレクトリが作成される)
```

7. 内部温度を計算する。

```
% mpirun -np 4 pfemsolv partstave
(“partstave” ディレクトリ内 “result” ディレクトリが作成される)
```

8. 温度データファイルの作成

```
% hddmmrg Temperature partstave
(Temperature.dat ファイルが出力される)
```

9. 温度分布の可視化 (ADVENTURE_Auto)

```
% advauto_thermalview partstavecs.msh partstavecs_4.fgr
```

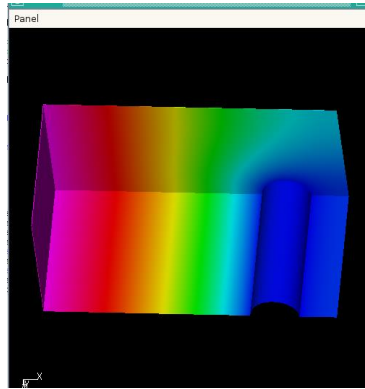



図 24. Advauto_thermalview による温度分布の可視化結果

E.7. 大規模解析例

高温工学試験研究炉(High Temperature Test Reactor)で使用する燃料集合体の200万自由度規模モデル(HTTRモデル)を、ADVENTURE_Thermalを使って解析した。境界条件は、モデル底部の高い温度を、上部に低い温度を設定した。図26がADVENTURE_Thermalで解析した温度分布である。

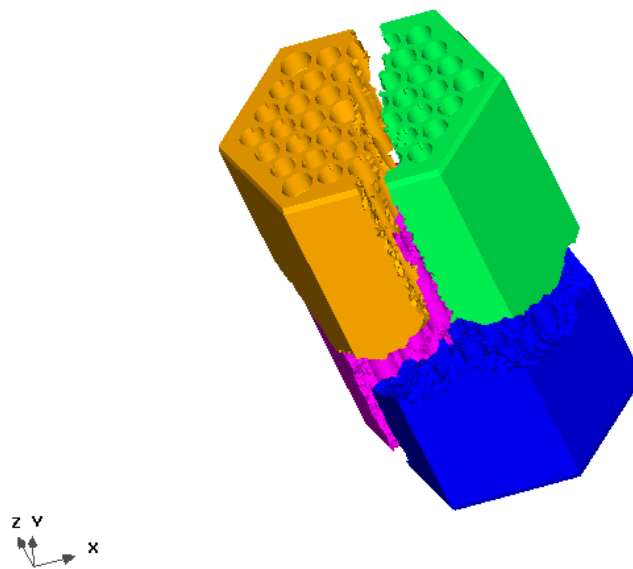


図 25. HTTR モデルの領域分割

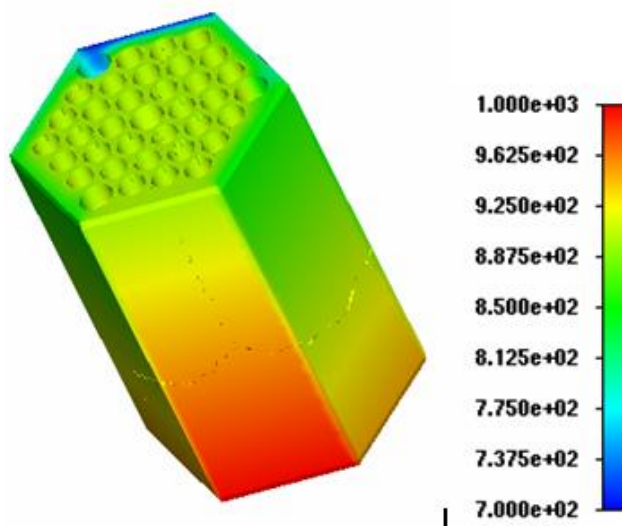


図 26. Advauto_thermalview による温度分布の可視化結果

6. 参考文献

- [1]. ADVENTURE Project: <http://adventure.sys.t.u-tokyo.ac.jp>
- [2]. G.Yagawa and R.Shioya: Parallel Finite Elements on a Massively Parallel Computer with Domain Decomposition, *Computing Systems in Engineering*, 4, Nos. 4-6 (1993), pp. 495-503.
- [3]. G.Yagawa and R.Shioya: Massively Parallel Finite Element Analysis, Asakura-Shoten, (1998) (in Japanese).
- [4]. T.Miyamura, H.Noguchi, R.Shioya, S.Yoshimura and G.Yagawa: Massively Parallel Elastic-Plastic Finite Element Analysis Using the Hierarchical Domain Decomposition Method, *Transactions of Japan Society of Mechanical Engineers (JSME)*, 65-A, No.634(1999), pp. 1201-1208 (in Japanese).
- [5]. R.Shioya, H.Kanayama, D.Tagami and E.Imamura: A Domain Decomposition Approach for Non-steady Heat Conductive Analysis, *Advances in Computational Engineering & Science*, 189.pdf, pp. 1-6, 2001.
- [6]. MPI: <http://www-unix.mcs.anl.gov/mpi/>
- [7]. MPICH: <http://www-unix.mcs.anl.gov/mpi/mpich/>
- [8]. Jan Mandel: Balancing Domain Decomposition, *Communications on Numerical Methods in Engineering*, 9(1993), 233-241
- [9]. R.Shioya, H. Kanayama, A.M.M.Mukaddes and M. Ogino: Heat Conductive Analysis with Balancing Domain Decomposition Method, *Journal of Theoretical and Applied Mechanics*, 52(2003), 43-53.
- [10]. Abul Mukid Mohammad Mukaddes, Masao Ogino and Ryuji Shioya, Performance Evaluation of Domain Decomposition Method with Sparse Matrix Storage Schemes in Modern Supercomputer, *International Journal of Computational Methods*, Volume 11, Issue Supp 01, Nov. 2014
DOI: 10.1142/S0219876213440076.
- [11]. A M M Mukaddes, M. Ogino, H. Kanayama, and R. Shioya, A Scalable Balancing Domain Decomposition Based Preconditioner for Large Scale Heat Transfer Problems, *JSME International Journal, B-Fluid T.* 49-2(2006), 533-540.