

ADVENTURE SYSTEM

# **ADVENTURE\_Fluid**

Imcompressible thermal fluid analysis

Version: beta - 0.41

## **モジュール使用マニュアル**

May 18, 2005

**ADVENTURE Project**

# 目次

<b>1</b>	<b>概要</b>	<b>3</b>
<b>2</b>	<b>機能</b>	<b>4</b>
2.1	ソルバー . . . . .	4
2.1.1	adventure_fluid_tet . . . . .	4
2.1.2	adventure_fluid_hex . . . . .	4
2.1.3	adventure_thermal_fluid_hex . . . . .	4
2.2	プレ処理ツール . . . . .	4
2.2.1	advfluid_pre_cavity . . . . .	4
2.2.2	advfluid_pre_cavity_nc . . . . .	4
2.2.3	advfluid_pre_pillar . . . . .	5
2.3	データ変換ツール . . . . .	5
2.3.1	advfluid_mesh2ucd . . . . .	5
2.3.2	advfluid_p_mesh2ucd . . . . .	5
2.3.3	advfluid_rest2ucd . . . . .	5
2.3.4	advfluid_p_rest2ucd . . . . .	5
2.3.5	advfluid_rest2ucd_nc . . . . .	5
<b>3</b>	<b>コンパイルとインストール</b>	<b>6</b>
3.1	必要なライブラリ . . . . .	6
3.2	コンパイル . . . . .	6
3.3	インストール . . . . .	6
<b>4</b>	<b>実行方法</b>	<b>7</b>
4.1	ソルバー . . . . .	7
4.1.1	adventure_fluid_tet . . . . .	7
4.1.2	adventure_fluid_hex . . . . .	7
4.1.3	adventure_thermal_fluid_hex . . . . .	8
4.2	プレ処理ツール . . . . .	9
4.2.1	advfluid_pre_cavity . . . . .	9
4.2.2	advfluid_pre_cavity_nc . . . . .	9
4.2.3	advfluid_pre_pillar . . . . .	9
4.3	データ変換ツール . . . . .	11
4.3.1	advfluid_mesh2ucd . . . . .	11
4.3.2	advfluid_p_mesh2ucd . . . . .	11
4.3.3	advfluid_rest2ucd . . . . .	12
4.3.4	advfluid_p_rest2ucd . . . . .	12

4.3.5	advfluid_rest2ucd_nc . . . . .	13
<b>5</b>	<b>ファイルフォーマット</b>	<b>14</b>
5.1	メッシュデータ . . . . .	14
5.1.1	四面体 (P1-P1) 要素 . . . . .	14
5.1.2	六面体 (Q1-P0) 要素 . . . . .	14
5.1.3	節点 . . . . .	15
5.1.4	初期条件 . . . . .	16
5.1.5	境界条件:No-slip 境界 . . . . .	16
5.1.6	境界条件:流速指定境界 . . . . .	17
5.2	リスタートデータ . . . . .	17
5.2.1	流速場 . . . . .	17
5.2.2	圧力場 (P1-P1 要素の場合) . . . . .	18
5.2.3	圧力場 (Q1-P0 要素の場合) . . . . .	19
5.2.4	温度場 . . . . .	19
5.3	コントロールデータ . . . . .	20
5.3.1	adventure_fluid_tet . . . . .	20
5.3.2	adventure_fluid_hex . . . . .	21
5.3.3	adventure_thermal_fluid_hex . . . . .	21
<b>6</b>	<b>解析例</b>	<b>23</b>
6.1	立方キャビティ流れ解析 . . . . .	23
6.1.1	解析の手順 . . . . .	23
6.2	立方キャビティ流れ解析 (並列版) . . . . .	24
6.2.1	解析の手順 . . . . .	24
	<b>参考文献</b>	<b>27</b>

## 図目次

1	Parameters for advfluid_pre_pillar . . . . .	10
2	Control data for lid-driven cubic cavity flow . . . . .	23
3	Analysis model of 3-dimensional cavity flow . . . . .	25
4	Analysis result of 3-dimensional cavity flow . . . . .	26

## 1 概要

ADVENTURE\_Fluid モジュールは、ADVENTURE Project で開発された非圧縮性熱流体解析モジュールであり、以下のような特徴を持っている。

- 有限要素法による非圧縮性熱流体解析を行うことができる。
- 六面体 Q1-P0 要素と四面体 P1-P1 要素を用いることができる。
- Element-by-Element なアルゴリズムを採用しており、大規模解析向けである。
- MPI が動作する全ての並列分散環境で動作する。
- 解析コードおよび専用のプレ処理ツール、データ変換ツール類から構成されている。

## 2 機能

### 2.1 ソルバー

#### 2.1.1 `adventure_fluid_tet`

四面体 P1-P1 要素用の非圧縮性流体解析ソルバーであり、SUPG (Streamline-upwind/Petrov-Galerkin) 法と PSPG (Pressure-stabilized/Petrov-Galerkin) 法による安定化を行なっている。時間方向の離散化には Crank-Nicolson 法を用いており、流速場と圧力場を同時に非対称行列ソルバにより解いている。ソルバは、Bi-CG STAB 法、GPBi-CG 法、Bi-CG STAB2 法、GMRES(m) 法から選ぶことが出来る。

#### 2.1.2 `adventure_fluid_hex`

六面体 Q1-P0 要素用の非圧縮性流体解析ソルバーであり、BTD (Balancing Tensor Diffusivity) 項を付加することにより安定化を行なっている。時間方向の離散化には MAC 法を採用しており、圧力ポアソン方程式を CG 法により解いている。

#### 2.1.3 `adventure_thermal_fluid_hex`

六面体 Q1-P0 要素用の非圧縮性熱流体解析ソルバーである。エネルギー方程式を解いている点と、Navier-Stokes 方程式にブシネスク近似による浮力項が加わっている点以外は `adventure_fluid_hex` と同様である。

### 2.2 プレ処理ツール

#### 2.2.1 `advfluid_pre_cavity`

3次元立方キャビティ問題の専用プレツールである。六面体用と四面体用のどちらのモデルも作成することができる。コントロールファイルのテンプレートも自動的に作成される。

#### 2.2.2 `advfluid_pre_cavity_nc`

`advfluid_pre_cavity` と同様なモデルに自然対流 (natural convection) 問題用の境界条件が付与されている。`adventure_thermal_fluid_hex` 用のデータ作成ツールである。コントロールファイルのテンプレートも自動的に作成される。

### 2.2.3 advfluid\_pre\_pillar

一様流れ場に置かれた角柱の周りの流れ解析用の専用ツールである。各部分の長さや要素分割数を細かく指定することができる。また、X11 による簡易節点分布チェック機能もついている。

## 2.3 データ変換ツール

### 2.3.1 advfluid\_mesh2ucd

作成されたメッシュを AVS の UCD 形式に変換するツールである。これにより、AVS Express や Micro AVS 等を用いてメッシュや境界条件のチェックを行うことができる。

### 2.3.2 advfluid\_p\_mesh2ucd

advfluid\_mesh2ucd と機能は全く同じであるが、領域分割後のデータ用である。領域分割された複数のデータを一括して変換することができる。

### 2.3.3 advfluid\_rest2ucd

解析結果 (restart データ) を AVS の UCD 形式に変換するツールである。これにより、AVS Express や Micro AVS 等を用いて解析結果の速度場や温度場をチェックすることができる。

### 2.3.4 advfluid\_p\_rest2ucd

advfluid\_rest2ucd と機能は全く同じであるが、領域分割後のデータ用である。領域分割された複数のデータを一括して変換することができる。

### 2.3.5 advfluid\_rest2ucd\_nc

advfluid\_rest2ucd と機能は全く同じであるが、熱流体解析のデータ用である。adventure\_thermal\_fluid\_hex の解析結果を AVS の UCD 形式に変換するツールである。

## 3 コンパイルとインストール

### 3.1 必要なライブラリ

ADVENTURE\_Fluid モジュールを使うためには、以下のライブラリが必要である。

- ADVENTURE\_IO モジュール
- MPI ライブラリ (mpich, LAM など)

### 3.2 コンパイル

ADVENTURE\_Fluid モジュールは configure スクリプトに対応している。ほとんどのシステムでは、コンパイルするために以下のコマンドを実行するだけである。

```
% ./configure
```

```
% make
```

### 3.3 インストール

全てのプログラムを\$(HOME)/ADVENTURE/bin にインストールするためには、

```
% make install
```

としてやればよい。また\$(HOME)/ADVENTURE/bin から全てのプログラムを削除し、コンパイルした全てのプログラムとオブジェクトモジュールを削除するためには、

```
% make clean
```

としてやればよい。

## 4 実行方法

### 4.1 ソルバー

#### 4.1.1 adventure\_fluid\_tet

adventure\_fluid\_tet を実行するには、以下のようにすればよい。

```
% mpirun -np <np> adventure_fluid_tet <log> <ctrl> <mesh> <rest_out> [<rest_in>]
<np>      : <np>   (プロセッサ数)
<log>     : <log>  (ログファイル名)
<ctrl>    : <ctrl> (コントロールファイル名)
<mesh>    : <mesh>_<np>.adv (メッシュファイル名)
<rest_out> : <rest_out>_<num_steps>_<np>.adv (リスタート出力ファイル名)
<rest_in> : <rest_in>_<np>.adv (リスタート入力ファイル名)
```

この時、以下の点に注意する必要がある。

- 1PE で実行する場合にも、`mpirun -np 1` を指定する。
- ログファイル名には任意のファイル名を指定する。
- コントロールファイル名には次章で述べるコントロールファイルを指定する。
- メッシュファイル名には次章で述べるメッシュファイルを指定する。また、その際に領域番号\_<np>と拡張子.advを除いたファイル名の先頭部分のみを指定する。
- リスタート出力ファイル名には任意のファイル名を指定する。実際に出力されるファイル名は、ここで指定したファイル名の先頭部分に時間ステップ数\_<num\_steps>、領域番号\_<np>、拡張子.advが付加される。
- リスタート入力ファイル名はリスタート計算を行う場合のみ指定する。実際にリスタート計算を行うかどうかはコントロールファイルの中で指定する。入力するリスタートファイル名から領域番号\_<np>と拡張子.advを除いたファイル名の先頭部分のみを指定する。

#### 4.1.2 adventure\_fluid\_hex

adventure\_fluid\_hex を実行するには、以下のようにすればよい。

```
% mpirun -np <np> adventure_fluid_hex <mesh> <ctrl> <rest_out> [<rest_in>]
<np>      : <np>   (プロセッサ数)
<mesh>    : <mesh>_<np>.adv (メッシュファイル名)
<ctrl>    : <ctrl> (コントロールファイル名)
```



<rest\_out> : <restout>\_<num\_steps>\_<np>.adv (リスタート出力ファイル名)  
<rest\_in> : <rest\_in>\_<np>.adv (リスタート入力ファイル名)

この時、以下の点に注意する必要がある。

- 1PE で実行する場合にも、mpirun -np 1 を指定する。
- メッシュファイル名には次章で述べるメッシュファイルを指定する。また、その際に領域番号\_<np>と拡張子.advを除いたファイル名の先頭部分のみを指定する。
- コントロールファイル名には次章で述べるコントロールファイルを指定する。
- リスタート出力ファイル名には任意のファイル名を指定する。実際に出力されるファイル名は、ここで指定したファイル名の先頭部分に時間ステップ数\_<num\_steps>、領域番号\_<np>、拡張子.advが付加される。
- リスタート入力ファイル名はリスタート計算を行う場合のみ指定する。実際にリスタート計算を行うかどうかはコントロールファイルの中で指定する。入力するリスタートファイル名から領域番号\_<np>と拡張子.advを除いたファイル名の先頭部分のみを指定する。
- adventure\_fluid\_hex にはログをファイルに落とす機能がないため、コマンドラインの最後にリダイレクトj&を付けてログをファイルに記録する必要がある。

#### 4.1.3 adventure\_thermal\_fluid\_hex

adventure\_thermal\_fluid\_hex を実行するには、以下のようにすればよい。

```
% mpirun -np <np> adventure_thermal_fluid_hex <mesh> <ctrl> <rest_out> [<rest_in>]
<np>      : <np> (プロセッサ数)
<mesh>    : <mesh>_<np>.adv (メッシュファイル名)
<ctrl>    : <ctrl> (コントロールファイル名)
<rest_out> : <restout>_<num_steps>_<np>.adv (リスタート出力ファイル名)
<rest_in> : <rest_in>_<np>.adv (リスタート入力ファイル名)
```

この時、以下の点に注意する必要がある。

- 引数は全て adventure\_fluid\_hex と同じである。
- コントロールファイルの書式のみ異なる (次章参照)

## 4.2 プレ処理ツール

### 4.2.1 advfluid\_pre\_cavity

advfluid\_pre\_cavity を実行するには、以下のようにすればよい。

```
% advfluid_pre_cavity <tet | hex> <num_division_per_direction> <write file name>
```

この時、以下の点に注意する必要がある。

- 1 番目の引数には、tet(四面体) または hex(六面体) を指定する。
- 2 番目の引数には、立方体の一辺の分割数を指定する。ここで指定した数を  $N$  とすると、要素数は四面体  $5N^3$ 、六面体  $N^3$  となり、節点数は  $(N + 1)^3$  となる。
- 3 番目の引数には、任意の出力ファイル名を指定する。

### 4.2.2 advfluid\_pre\_cavity\_nc

advfluid\_pre\_cavity\_nc を実行するには、以下のようにすればよい。

```
% advfluid_pre_cavity_nc <num_division_per_direction> <write file name>
```

この時、以下の点に注意する必要がある。

- このプレツールは adventure\_thermal\_fluid\_hex 専用である。
- 1 番目の引数には、立方体の一辺の分割数を指定する。ここで指定した数を  $N$  とすると、要素数は四面体  $5N^3$ 、六面体  $N^3$  となり、節点数は  $(N + 1)^3$  となる。
- 2 番目の引数には、任意の出力ファイル名を指定する。

### 4.2.3 advfluid\_pre\_pillar

advfluid\_pre\_pillar を実行するには、以下のようにすればよい。

```
% advfluid_pre_pillar <filename>
```

この時、以下の点に注意する必要がある。

- 引数には、任意の出力ファイル名を指定する。
- 実行すると 5 つのモデルサイズパラメータを聞かれるので、図 1 を参考にして入力する。

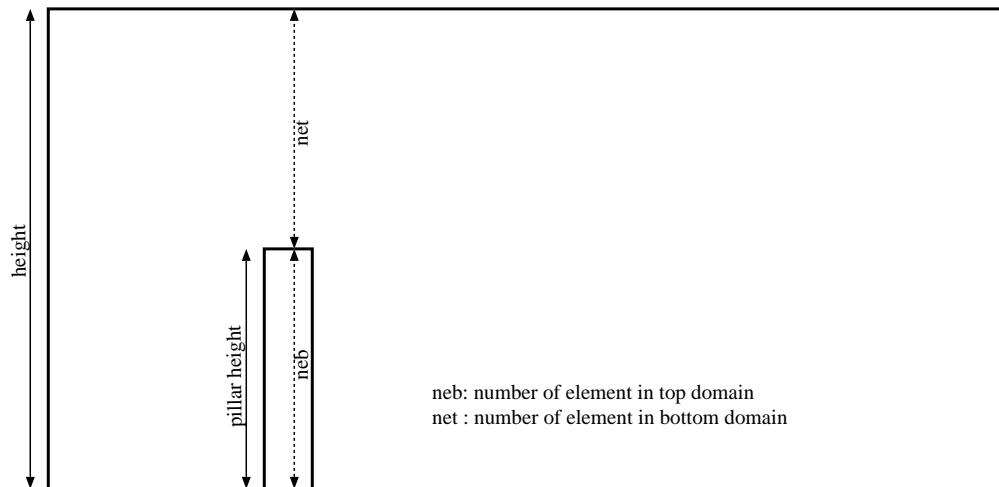
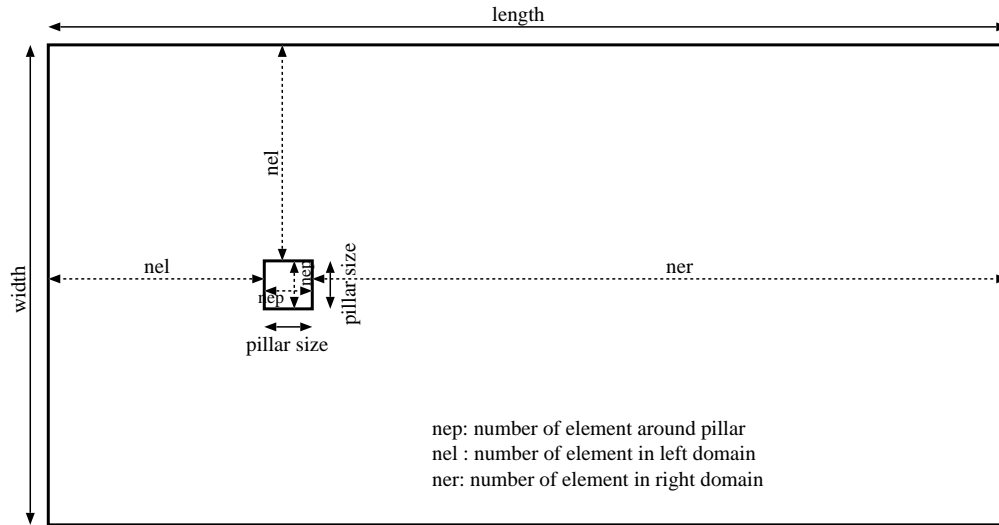


图 1: Parameters for advfluid\_pre\_pillar

- 次に、3つの X-Y 方向の要素分割パラメータを聞かれるので、図1を参考にして入力すると、X-Y 平面の節点分布が現れるので、節点分布を確認して OK なら y を入力する。
- 次に、2つの X-Z 方向の要素分割パラメータを聞かれるので、図1を参考にして入力すると、X-Z 平面の節点分布が現れるので、節点分布を確認して OK なら y を入力すると、メッシュデータが作成される

## 4.3 データ変換ツール

### 4.3.1 advfluid\_mesh2ucd

advfluid\_mesh2ucd を実行するには、以下のようにすればよい。

```
% advfluid_mesh2ucd [-M] <mesh_file> <ucd_file>
-M           : Micro AVS mode(including I.C and B.C data)
<mesh_file> : input mesh file name(<mesh_file>)
<ucd_file>  : output UCD file name(<ucd_file>.inp)
```

この時、以下の点に注意する必要がある。

- 初期条件・境界条件を同時に出力する場合には、-M オプションを付ける。MicroAVS ではこれがないと表示できない。
- <mesh\_file>には変換元のメッシュデータファイルを指定する。
- <ucd\_file>には変換先の UCD データファイルを指定する。この時、.inp の拡張子は自動的に付け加えられる。

### 4.3.2 advfluid\_p\_mesh2ucd

advfluid\_p\_mesh2ucd を実行するには、以下のようにすればよい。

```
% advfluid_p_mesh2ucd [-M] <np> <mesh_file> <ucd_file>
-M           : Micro AVS mode(including I.C and B.C data)
<np>        : number of PEs(number of domain)
<mesh_file> : input mesh file name(<mesh_file>_<n>.adv)
<ucd_file>  : output UCD file name(<ucd_file>_<n>.inp)
```

この時、以下の点に注意する必要がある。

- 初期条件・境界条件を同時に出力する場合には、-M オプションを付ける。MicroAVS ではこれがないと表示できない。

- `<np>`には領域数を指定する。
- `<mesh_file>`には変換元のメッシュデータファイルを指定する。指定するファイル名は、領域番号と`.adv`の拡張子を除いた部分である。
- `<ucd_file>`には変換先のUCDデータファイルを指定する。この時、領域番号と`.inp`の拡張子は自動的に付け加えられる。従って、指定するファイル名は、領域番号と`.inp`の拡張子を除いた部分である。

### 4.3.3 advfluid\_rest2ucd

`advfluid_rest2ucd` を実行するには、以下のようにすればよい。

```
% advfluid_rest2ucd <mesh_file> <rest_file> <ucd_file>
<mesh_file> : input mesh file name(<mesh_file>)
<rest_file> : input restart file name(<rest_file>)
<ucd_file>  : output UCD file name(<ucd_file>.inp)
```

この時、以下の点に注意する必要がある。

- `<mesh_file>`には変換元のメッシュデータファイルを指定する。
- `<rest_file>`には変換元のリスタートデータ (計算結果) ファイルを指定する。
- `<ucd_file>`には変換先のUCDデータファイルを指定する。この時、`.inp`の拡張子は自動的に付け加えられる。

### 4.3.4 advfluid\_p\_rest2ucd

`advfluid_p_rest2ucd` を実行するには、以下のようにすればよい。

```
% advfluid_p_rest2ucd <np> <mesh_file> <rest_file> <ucd_file>
<np>          : number of PEs(number of domain)
<mesh_file>   : input mesh file name(<mesh_file>_<n>.adv)
<rest_file>   : input restart file name(<rest_file>_<n>.adv)
<ucd_file>    : output UCD file name(<ucd_file>_<n>.inp)
```

この時、以下の点に注意する必要がある。

- `<np>`には領域数を指定する。
- `<mesh_file>`には変換元のメッシュデータファイルを指定する。指定するファイル名は、領域番号と`.adv`の拡張子を除いた部分である。

- `<rest_file>`には変換元のリスタートデータ (計算結果) ファイルを指定する。指定するファイル名は、領域番号と `.adv` の拡張子を除いた部分である。
- `<ucd_file>`には変換先の UCD データファイルを指定する。この時、領域番号と `.inp` の拡張子は自動的に付け加えられる。従って、指定するファイル名は、領域番号と `.inp` の拡張子を除いた部分である。

#### 4.3.5 advfluid\_rest2ucd\_nc

`advfluid_rest2ucd_nc` を実行するには、以下のようにすればよい。

```
% advfluid_rest2ucd_nc <mesh_file> <rest_file> <ucd_file>
<mesh_file> : input mesh file name(<mesh_file>)
<rest_file> : input restart file name(<rest_file>)
<ucd_file>  : output UCD file name(<ucd_file>.inp)
```

この時、以下の点に注意する必要がある。

- `<mesh_file>`には変換元のメッシュデータファイルを指定する。
- `<rest_file>`には変換元のリスタートデータ (計算結果) ファイルを指定する。
- `<ucd_file>`には変換先の UCD データファイルを指定する。この時、`.inp` の拡張子は自動的に付け加えられる。

## 5 ファイルフォーマット

メッシュデータファイル及びリスタートデータファイルのフォーマットは ADVENTURE\_IO を用いた Document 形式になっている。また、コントロールデータファイルはテキスト形式で記述する。

### 5.1 メッシュデータ

メッシュデータファイルには、要素データ、節点データ、初期条件データ、境界条件データが含まれている。各データはそれぞれ1つ以上の Document 形式になっており、以下のようなフォーマットで記述されている。

#### 5.1.1 四面体 (P1-P1) 要素

四面体 (P1-P1) 要素データのドキュメントは以下のようなフォーマットで記述されている。

[プロパティ部分]

```
content_type      =      "Element"
num_items         =      <num_element>
num_items_orig   =      <num_element_before_decomposed>
format           =      "i4i4i4i4"
num_node_per_element =      4
element_type     =      "3DLinearTetrahedron"
dimension        =      3
index_byte       =      4
```

[マスデータ部分] (要素-節点コネクティビティデータ)

```
nop[0][0], nop[0][1], nop[0][2], nop[0][3]
nop[1][0], nop[1][1], nop[1][2], nop[1][3]
...
```

(要素数 x4 個の int 型データがこの順序で並んでいる。)

#### 5.1.2 六面体 (Q1-P0) 要素

六面体 (Q1-P0) 要素データのドキュメントは以下のようなフォーマットで記述されている。

[プロパティ部分]

```
content_type      =      "Element"
num_items         =      <num_element>
num_items_orig   =      <num_element_before_decomposed>
format           =      "i4i4i4i4i4i4i4i4"
num_node_per_element =      8
element_type     =      "3DLinearHexahedron"
dimension        =      3
index_byte       =      4
```

[マスデータ部分] (要素-節点コネクティビティデータ)

```
nop[0][0], nop[0][1], nop[0][2], nop[0][3],
  nop[0][4], nop[0][5], nop[0][6], nop[0][7]
nop[1][0], nop[1][1], nop[1][2], nop[1][3],
  nop[1][4], nop[1][5], nop[1][6], nop[1][7]
```

...

(要素数 x8 個の int 型データがこの順序で並んでいる。)

### 5.1.3 節点

節点データのドキュメントは以下のようなフォーマットで記述されている。

[プロパティ部分]

```
content_type      =      "Node"
num_items         =      <num_node>
num_items_orig   =      <num_node_before_decomposed>
format           =      "f8f8f8"
dimension        =      3
index_byte       =      4
```

[マスデータ部分] (節点座標データ)

```
x[0], y[0], z[0]
x[1], y[1], z[1]
```

...



(節点数 x3 個の double 型データがこの順序で並んでいる。)

#### 5.1.4 初期条件

初期条件データのドキュメントは以下のようなフォーマットで記述されている。但し、これは解析領域全体に一様な流速場を与える場合である。。

[プロパティ部分]

```
content_type      =      "FEGenericAttribute"  
num_items         =      1  
format            =      "f8f8f8"  
fega_type        =      "AllNodeConstant"  
label             =      "VelocityIC"  
coordinate       =      7  
dd_option        =      "dirichlet"  
index_byte       =      4
```

[マスタデータ部分] (初期流速場データ)

u, v, w

#### 5.1.5 境界条件:No-slip 境界

No-slip 境界条件のデータのドキュメントは以下のようなフォーマットで記述されている。

[プロパティ部分]

```
content_type      =      "FEGenericAttribute"  
num_items         =      <num_node>  
format            =      "f8f8f8"  
fega_type        =      "NodeConstant"  
label             =      "VelocityBC"  
coordinate       =      7  
dd_option        =      "dirichlet"  
index_byte       =      4
```

[マスタデータ部分] (流速値及び、指定する節点番号の並び)

0.0, 0.0, 0.0

n[0], n[1], ...

(<num\_node>個の節点番号データが任意の順序で並んでいる。)

### 5.1.6 境界条件:流速指定境界

流速指定境界条件のデータのドキュメントは以下のようなフォーマットで記述されている。

[プロパティ部分]

```
content_type      =      "FEGenericAttribute"
num_items         =      <num_node>
format           =      "f8f8f8"
fega_type        =      "NodeConstant"
label            =      "VelocityBC"
coordinate       =      7
dd_option        =      "dirichlet"
index_byte       =      4
```

[マスタデータ部分] (流速値及び、指定する節点番号の並び)

u, v, w

n[0], n[1], ...

(<num\_node>個の節点番号データが任意の順序で並んでいる。)

## 5.2 リスタートデータ

リスタートデータファイルには、流速場データ、圧力場データが含まれている。また、adventure\_thermal\_fluid\_hexのリスタートデータのみ温度場データも含まれている。各データはそれぞれ1つのDocument形式になっており以下のようなフォーマットで記述されている。

### 5.2.1 流速場

流速場データのドキュメントは以下のようなフォーマットで記述されている。

[プロパティ部分]

```

content_type      =      FEGenericAttribute
num_items        =      <num_node>
format           =      f8f8f8
fega_type       =      AllNodeVariable
label           =      VelocityRestartData
index_byte      =      4
num_steps       =      <time_steps>
time            =      <time>

```

[マスデータ部分] (流速場データ)

```

u[0], v[0], w[0]
u[1], v[1], w[1]
...

```

(節点数 x3 個の double 型データがこの順序で並んでいる。)

### 5.2.2 圧力場 (P1-P1 要素の場合)

P1-P1 要素の流速場データのドキュメントは以下のようなフォーマットで記述されている。

[プロパティ部分]

```

content_type      =      FEGenericAttribute
num_items        =      <num_node>
format           =      f8
fega_type       =      AllNodeVariable
label           =      PressureRestartData
index_byte      =      4
num_steps       =      <time_steps>
time            =      <time>

```

[マスデータ部分] (圧力場データ)

```

p[0], p[1], ...

```

(節点数 x1 個の double 型データがこの順序で並んでいる。)

### 5.2.3 圧力場 (Q1-P0 要素の場合)

Q1-P0 要素の流速場データのドキュメントは以下のようなフォーマットで記述されている。

[プロパティ部分]

```
content_type      =      FEGenericAttribute
num_items         =      <num_element>
format            =      f8
fega_type         =      AllElementVariable
label             =      PressureRestartData
index_byte        =      4
num_steps         =      <time_steps>
time              =      <time>
```

[マスデータ部分] (圧力場データ)

p[0], p[1], ...

(要素数 x1 個の double 型データがこの順序で並んでいる。)

### 5.2.4 温度場

温度場データのドキュメントは以下のようなフォーマットで記述されている。

[プロパティ部分]

```
content_type      =      FEGenericAttribute
num_items         =      <num_node>
format            =      f8
fega_type         =      AllNodeVariable
label             =      TemperatureRestartData
index_byte        =      4
num_steps         =      <time_steps>
time              =      <time>
```

[マスデータ部分] (温度場データ)

t[0], t[1], ...

(節点数 x1 個の double 型データがこの順序で並んでいる。)

### 5.3 コントロールデータ

コントロールデータファイルには、レイノルズ数や時間刻み幅などの各種解析条件がテキスト形式で記述されており、それらはソルバーによって若干の違いがある。具体的には以下の通りである。

#### 5.3.1 adventure\_fluid\_tet

adventure\_fluid\_tet 用のコントロールデータは以下のようなフォーマットで記述されている。

[adventure\_fluid\_tet 用コントロールファイル例]

```
dt          1.000000e-02
t_end       1.000000e+00
mu          0.01
rho         1.0
solver_type 1
max_cg      1000
eps_cg      1.000000e-10
diag_scaling_do 1
rest_in_do  0
rest_intvl  100
```

それぞれの変数の意味は以下の通りである。

dt	時間刻み幅
t_end	解析終了時間
mu	粘性係数
rho	密度
solver_type	0:Bi-CGSTAB 法、1:GPBi-CG 法、 2:Bi-CGSTAB2 法、m(3 以上):GMRES(m) 法
max_cg	ソルバ最大反復数
eps_cg	ソルバ収束判定用イプシロン
diag_scaling_do	0:対角スケーリングをしない、1:対角スケーリングをする。
rest_in_do	リスタート計算用フラグ(0; 最初から計算、1; リスタート計算)
rest_intvl	リスタートデータを出力するインターバル(時間ステップ数)

### 5.3.2 adventure\_fluid\_hex

adventure\_fluid\_hex 用のコントロールデータは以下のようなフォーマットで記述されている。

[adventure\_fluid\_hex 用コントロールファイル例]

```
dt          2.000000e-02
t_end       1.000000e+01
max_cg      1000
eps_cg      1.000000e-6
eps_mac     1.000000e-6
rest_in_do  0
rest_intvl  1000
re          100
hourg_level 2
```

それぞれの変数の意味は以下の通りである。

dt	時間刻み幅
t_end	解析終了時間
max_cg	CG 法最大反復数
eps_cg	CG 法収束判定用イプシロン
eps_mac	MAC 法収束判定用イプシロン
rest_in_do	リスタート計算用フラグ (0; 最初から計算、1; リスタート計算)
rest_intvl	リスタートデータを出力するインターバル (時間ステップ数)
re	レイノルズ数
hourg_level	アワーグラスコントロールのレベル (0; なし、1;Gresho、2;Okuda et al.)

### 5.3.3 adventure\_thermal\_fluid\_hex

adventure\_thermal\_fluid\_hex 用のコントロールデータは以下のようなフォーマットで記述されている。

[adventure\_thermal\_fluid\_hex 用コントロールファイル例]

```
dt          2.000000e-02
t_end       1.000000e+01
max_cg      1000
eps_cg      1.000000e-6
eps_mac     1.000000e-6
```

```
rest_in_do      0
rest_intvl     1000
ra             1.000000e+5
pr             7.100000e+0
hourg_level    2
```

それぞれの変数の意味は以下の通りである。

```
dt             時間刻み幅
t_end         解析終了時間
max_cg        CG 法最大反復数
eps_cg        CG 法収束判定用イプシロン
eps_mac       MAC 法収束判定用イプシロン
rest_in_do    リスタート計算用フラグ (0; 最初から計算、 1; リスタート計算)
rest_intvl    リスタートデータを出力するインターバル (時間ステップ数)
ra            レーリー数 (Ra)
pr            プランドル数 (Pr)
hourg_level   アワーグラスコントロールのレベル (0; なし、 1; Gresho、 2; Okuda et al.)
```

## 6 解析例

### 6.1 立方キャビティー流れ解析

ここでは、四面体 P1-P1 要素用の非圧縮性流体解析ソルバー `advfluid_tet` を用いて、立方キャビティー流れ解析を行なう手順を示す。

#### 6.1.1 解析の手順

立方キャビティー流れ解析を行なうには、以下のようにすれば良い。

1. プレツールを用いてメッシュデータを作成する。

```
% advfluid_pre_cavity tet 8 tet8
```

この例では、一辺を 8 分割するモデルを作成している。

2. データ変換ツールを用いて作成されたメッシュデータを確認する。

```
% advfluid_mesh2ucd -M tet8 tet8
```

これにより、AVS 用 UCD データファイル (`tet8.inp`) が作成される。(図 3)

3. メッシュデータ (`tet8`) と同時に作成されるコントロールファイル (`sample.ctrl`) の中身を、適当なエディタを用いて例えば図 2 のような値に編集する。

```
% vi sample.ctrl
```

```

                                dt                2.500000e-02
                                t_end            2.500000e+00
                                mu               1.000000e-02
                                rho              1.000000e+00
solver_type                      1
                                max_cg          1000
                                eps_cg          1.000000e-08
diag_scaling_do 1
                                rest_in_do     0
                                rest_intvl     50
```

図 2: Control data for lid-driven cubic cavity flow

4. 領域分割を行なわないで解析を行なう場合 (並列解析ではない場合) には、作成されたメッシュデータに領域番号 0 と拡張子 `.adv` を手動で付けてやる必要がある。具体的には、以下のようなシンボリックリンクを張ればよい。

```
% ln -s tet8 tet8_0.adv
```



5. 解析を実行する。

```
% mpirun -np 1 adventure_fluid_tet log sample.ctrl tet8 tet8
```

単一 PE で計算する場合にも mpirun を用いる点に注意されたい。また、実際には adventure\_fluid\_tet コマンドを絶対パスで指定しないと mpirun からコマンドが見つからない点にも注意されたい。

6. 図 2 のデータを用いて実行すると、50 ステップ毎にリスタートデータを出力するので、解析終了時には以下の 2 つのデータが作成されている。

```
tet8_000050_0.adv  
tet8_000100_0.adv
```

7. データ変換ツールを用いて作成されたリスタートデータを変換する。

```
% advfluid_rest2ucd tet8 tet8_000050_0.adv tet8_000050  
% advfluid_rest2ucd tet8 tet8_000100_0.adv tet8_000100
```

これにより、以下の名前の AVS 用 UCD データファイルが作成される。

```
tet8_000050.inp (50 ステップ計算時の解析結果)  
tet8_000100.inp (100 ステップ計算時の解析結果)(図 4)
```

## 6.2 立方キャビティー流れ解析 (並列版)

ここでは、四面体 P1-P1 要素用の非圧縮性流体解析ソルバー adventure\_fluid\_tet を用いて、立方キャビティー流れの解析を領域分割法を用いた並列処理で行なう手順を示す。

### 6.2.1 解析の手順

立方キャビティー流れの並列解析を行なうには、以下のようにすれば良い。

1. プレツールを用いてメッシュデータを作成する。

```
% advfluid_pre_cavity tet 8 tet8
```

この例では、一辺を 8 分割するモデルを作成している。

2. ADVENTURE\_Metis を用いて領域分割する。

最初に領域分割されたデータが出力されるディレクトリ (out) を作成する。

```
% mkdir out
```

次に、adventure\_metis に -DYS オプションを付けてメッシュデータを領域分割する。-DYS オプションは領域間反復を伴わない単層の領域分割用に用意されたオプションである。引数には、入力データファイル名、出力データファイル名 (ベース部分のみ) を指定する。

```
% mpirun -np 2 adventure_metis -DYS tet8 tet8
```

これにより、以下のファイルが作成される。

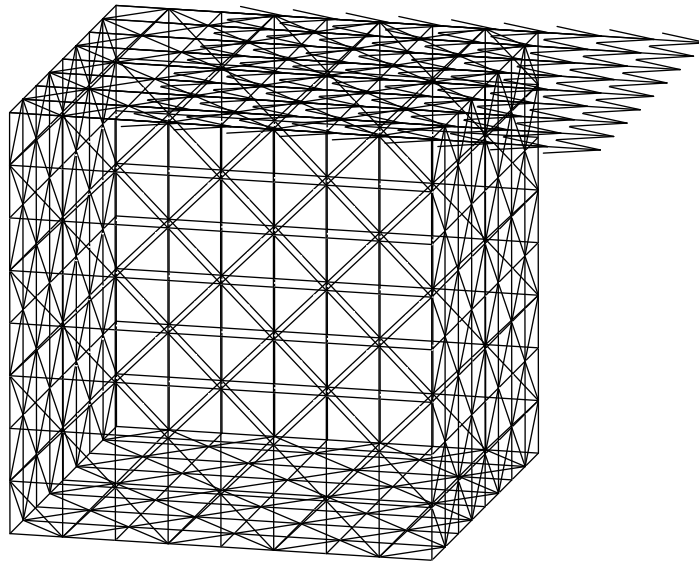


図 3: Analysis model of 3-dimensional cavity flow

```
out/tet8_0.adv  
out/tet8_1.adv
```

3. データ変換ツールを用いて作成されたメッシュデータを確認する。

```
% advfluid_p_mesh2ucd -M 2 out/tet8 out/tet8  
これにより、AVS 用 UCD データファイルが作成される。  
out/tet8_0.inp  
out/tet8_1.inp
```

4. メッシュデータ (tet8) と同時に作成されるコントロールファイル (sample.ctrl) の中身を、適切なエディタを用いて例えば図 2 のような値に編集する。

5. 解析を実行する。

```
% mpirun -np 2 adventure_fluid_tet log sample.ctrl out/tet8 out/tet8  
この時、adventure_fluid_tet コマンドを絶対パスで指定しないと mpirun からコマンド  
が見つからない点に注意されたい。
```

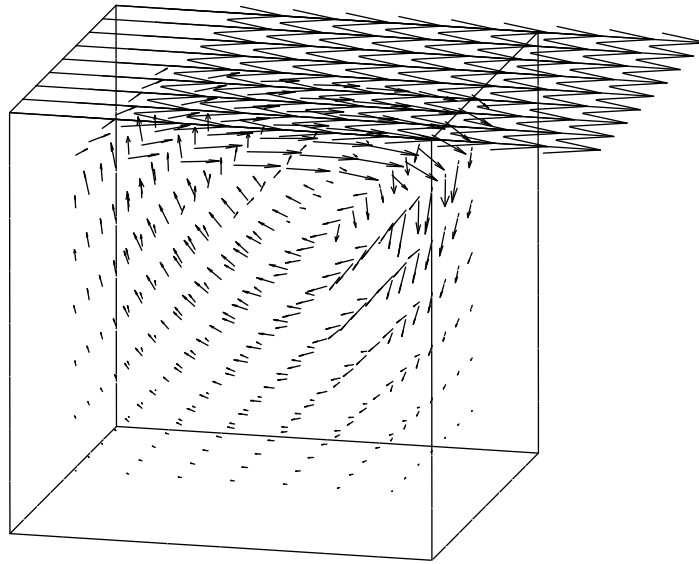


図 4: Analysis result of 3-dimensional cavity flow

6. 図 2 のデータを用いて実行すると、50 ステップ毎にリスタートデータを出力するので、解析終了時には以下の 4 つのデータが作成されている。

```
out/tet8_000050_0.adv
out/tet8_000050_1.adv
out/tet8_000100_0.adv
out/tet8_000050_1.adv
```

7. データ変換ツールを用いて作成されたリスタートデータを変換する。

```
% advfluid_prest2ucd tet8 tet8_000050 tet8_000050
% advfluid_prest2ucd tet8 tet8_000100 tet8_000100
```

これにより、以下の名前の AVS 用 UCD データファイルが作成される。

```
tet8_000050_0.inp (50 ステップ計算時の領域 0 の解析結果)
tet8_000050_1.inp (50 ステップ計算時の領域 1 の解析結果)
tet8_000100_0.inp (100 ステップ計算時の領域 0 の解析結果)
tet8_000100_1.inp (100 ステップ計算時の領域 1 の解析結果)
```

## 参考文献

- [1] ADVENTURE Project Home Page, <http://adventure.q.t.u-tokyo.ac.jp>
- [2] Yagawa, G., Nakabayashi, Y. and Okuda, H., “Large-Scale Finite Element Fluid Analysis by Massively Parallel Processors”, *Parallel Computing*, Vol.23, pp.1365-1377 (1997).
- [3] Nakabayashi, Y., Okuda, H. and Yagawa, G., “Parallel Finite Element Fluid Analysis on an Element-by-Element Basis”, *Computational Mechanics*, Vol.18, pp.377-382 (1996).
- [4] Okuda, H. and Yagawa, G., “A One-point Quadrature Technique with a New Hour-glass Controller for Large Scale Finite Element Fluid Analysis.”, *Proc. of 2nd Japan-US Symposium on FEM in Large-Scale CFD*, pp.125-128 (1994).
- [5] Gresho, P.M., Chan, S.T., Lee, R.L. and Up-son, C.D., “A Modified Finite Element Method for Solving the Time-Dependent Incompressible Navier-Stokes Equations by a Fractional Step Method.”, *Int. J. Num. Meth. in Fluids*, Vol.4, pp.557-598 (1984).
- [6] Tezduyar, T.E., Aliabadi, S., Behr, M., Johnson, A., Kalro, V. and Litke, M., “Flow Simulation and High Performance Computing”, *Computational Mechanics*, Vol.18, pp.397-412 (1995).
- [7] Kalro, V., Aliabadi, S., Garrard, W., Tezduyar, T.E., Mittal, S. and Stein, K., “Parallel Finite Element Simulation of Large Ram-Air Parachutes”, *Compt. Meth. Appl. Mech. Fluids* (1997).