

キャビティ流れの解析、paraFoamの実習

大嶋拓也(新潟大学)

2009/5/13 OpenFOAM講習会

キャビティ流れの解析を例に、OpenFOAM・ParaViewを用いた解析をひととおり実習

- 問題説明
- OpenFOAMケースフォルダの構造
- blockMeshによるメッシュの作成
- 境界条件・実行条件の与え方
- ソルバ(icoFoam)の実行
- ParaViewによる可視化

キャビティ流れ(1)

- キャビティ: 空洞のこと
- 上端の壁が水平方向に一定速度で動き、引きずられるように内部の流体が動く
- CFDの事例として、最もポピュラーな問題の一つ
- CFDソフトウエアの検証に、頻繁に使用される
- 1982年、Ghiaらによる論文が有名



キャビティ流れ(2)

- 一辺1 mの正方形キャビティ
- 上端の壁の移動速度 *u* = 1 m/s
- 流体は密度1 kg/m³
- 動粘性係数0.01 m²/s
- 辺長およびuによるレイノルズ数Re = 100
- 非定常・非圧縮ソルバを使用



計算領域の定義

• 計算領域 (OpenFOAMメッシュは必ず3次元なので、z方向に厚みを持たせる)





• OpenFOAMは通常、ホームフォルダ以下にインストールする



ケースフォルダの構成



• メッシュ設定ファイル: constant/polyMesh/blockMeshDict





```
convertToMeters 1; 全体の縮尺の設定(原寸大なので1)
vertices 頂点座標の設定
  (000) 0番目の頂点
  (1 \ 0 \ 0)
  (1\ 1\ 0)
  (0\ 1\ 0)
  (0\ 0\ 0.1)
  (1 \ 0 \ 0.1)
  (1\ 1\ 0.1)
  (010.1) 7番目の頂点
);
```







```
patches 境界面の設定
```

```
wall movingWall movingWallという名前の壁面(wall)
   (3762) 頂点3、7、6、2を結んでできる面(内部から見て時計回り)
 wall fixedWalls 同様に、3つの面をまとめてfixedWallsと名付ける
   (0473)
   (2651)
   (1540)
 empty frontAndBack 2次元問題を3次元メッシュで解くためのダミーの壁(empty)
   (0321)
   (4567)
);
mergePatchPairs 本例題では使用しない
```

メッシュの生成(1): OpenFOAM端末の起動

- OpenFOAMのオペレーションは、コマンド入力が基本
- ここでは全て、コマンド入力で実行

端末の起動



メッシュの生成(2): blockMeshの実行



ケースフォルダへ移動

run

cd icoFoam/cavity

メッシュ生成コマンドblockMeshの実行

blockMesh

生成メッシュの確認

• 以下のメッシュファイルの生成を確認



初期条件・境界条件の設定

• 初期条件・境界条件ファイル: 0/U、0/p





初期条件・境界条件の設定(2): 速度



• 速度の初期条件・境界条件ファイル: 0/U

dimensions [01-10000]; 単位の次元。質量、長さ、時間、温度、物質量、電流、光度の順。 (長さ)・(時間)⁻¹→速度

internalField uniform (000); 内部メッシュの速度。一様に零ベクトル

```
boundaryField 境界面
 movingWall 動<壁面
           fixedValue; ディリクレ境界条件(値を固定)
   type
           uniform (100); 一様に(1,0,0)の速度ベクトルを与える
   value
 fixedWalls 固定壁面
           fixedValue; ディリクレ境界条件
   type
           uniform (000); 固定壁面なので、一様に零ベクトル
   value
 frontAndBack 3次元方向(z方向)両端の境界面
           empty; emptyという境界条件により、2次元問題を表す
   type
```

初期条件・境界条件の設定(4): 圧力

• 圧力の初期条件・境界条件ファイル: 0/p

```
dimensions [02-20000];
OpenFOAMの非圧縮ソルバでは、圧力を密度で割って与える。
圧力 = (質量)・(長さ)・(時間)<sup>-2</sup>、密度 = (質量)・(長さ)<sup>-3</sup>
→圧力/密度 = (長さ)<sup>2</sup>・(時間)<sup>-2</sup>
```

NIIGATA UNIVER

internalField uniform 0; 内部メッシュ→一様に圧力0

```
boundaryField
  movingWall
              zeroGradient; 境界面の法線方向勾配 = 0 (ノイマン条件)
    type
  fixedWalls
              zeroGradient;
    type
  frontAndBack
              empty;
    type
```

実行条件の設定(1)

• 実行条件の設定ファイル: system/controlDict



実行条件の設定(2)



| startTime | 0; 解析の開始時刻 |
|-----------------------|--|
| endTime | 5; 解析の終了時刻 |
| deltaT | 0.05; 時間刻み |
| writeControl | timeStep; データ書出し間隔の決定法 |
| writeInterval に書出す | 20; writeControlがtimeStepのとき、writeIntervalステップごと |

NIIGATA UNIVERSI

※上記以外は、ユーザーマニュアル4.3節参照



icoFoamによる解析の実行

icoFoam

icoFoamによる解析の実行(ログを残したいとき)

foamJob icoFoam

paraFoamによる可視化の開始

paraFoam

cavityケースで基本(1) データの読み込み







cavityケースで基本(3): 速度場の可視化(成分の表示)







[SHIFT] + ドラッグ





ツールバーの操作



レプレゼンテーション





Fileメニュー

- •作成した画像の保存
 - ➤Save Screenshot (画面に表示されている物を保存)
 - ▶ Save Animation (全時刻ステップの画像を連続的に保存)
- •可視化の各種設定の保存・読込み
 - ≻Save State (保存)
 - ≻Load State (読込み)





Sourcesメニュー

•paraFoamの中で、簡単な形状の作成が可能

≻Line (線分)

≻Plane (面)

➢Box (直方体)

≻Sphere (球)

. . .

≻Cylinder (円筒)

➢Annotate Time (解析上の時刻の表示)

・フィルタへの入力としても使用できる



フィルタ



Filtersメニュー

- •Filtersメニュー + Filtersツールバー
 - ▶数十種類のフィルタ
 - ▶ データ・形状の加工・抽出が可能
 - ▶複数のフィルタを組合せられる(パイプライン)
 - ▶ Pipeline Browserに、パイプラインの状態表示



フィルタの例: 速度場の可視化(ベクトルの表示)



フィルタの例: セル中心でベクトル表示



- OpenFOAMでは、セル中心で値が定義される
- Glyphフィルタでは節点への補間値を使用する
- セル中心でベクトルを描くには?



NIIGATA UNIVERSI

- interFoamで相界面を表示したいとき、gamma = 0.5の等値面を作成する
- 下はcavityケースでpの最小~最大を10分割する場合



 Contour作成後、Display→Opacityで0.2~0.5程度の値を入れると、半透明の 等値面になる

フィルタの例: Clipフィルタ



NIIGATA UNIVERS

- 切断面上の任意点、切断面の法線を指定する
- 法線の方向が残される部分になる
- キャビティの上半分を残すには

データ (*.OpenFOAM) Clip
•Origin
$$\rightarrow$$
(0.5, 0.5,
•Normal \rightarrow (0, 1, 0)

• 残ったメッシュをtriangulateしてしまう(元と異なるメッシュ、データ量の増大)

0.05)

フィルタの例: Sliceフィルタ



NIIGATA UNIVERSI

• Clipと同様

•Normal→(0, 1, 0)

multiblockのcell dataをsliceする場合には、注意が必要

▶ カラーコンター表示が崩れる場合あり

▶ 必要に応じて、Extract BlockでinternalMeshを取り出す





流線の始点となる点の集合として、Point Source、Line Sourceを使える

- Point Source: ある点を中心とするある半径内にランダムに点を分布
 ▶ 中心点、半径、点数を指定
- Line Source: ある線上に等間隔で分布
 - ▶ 始点、終点、点数を指定



- 注意: フラットセル、特異点(コーナー、境界面など)に弱い
 - ➤「Coincident points in polyline...can't compute normals」等のエラー
 - ▶ 特に、autoRefineMeshやsnappyHexMeshによるsplit-hexのポリヘドラを paraFoamが分割して出来るセル

- ➢ Initial step lengthを小さくしてみる
- ▶ foamToVTKの方が、フラットセルを回避するアルゴリズムで分割する
- ▶ ポリヘドラで流線が上手く行かない場合は、foamToVTKを試してみる



- ある断面全体を流線の始点としたい場合
- Stream Tracer (Custom Source)を使う



NIIGATA UNIVERSITY

• Origin \rightarrow (0.5, 0.5, 0.05)





- 任意の矩形面を流線の始点としたい場合
- Plane Sourceを使う
 - ➢ Origin、Point1、Point2で矩形の3頂点を指定
 - ➤ X Resolution、Y Resolutionで密度を指定



- Point2→(0.25, 0.5, 0.05)
- X Resolution→10
- Y Resolution \rightarrow 10

フィルタの例: パーティクルトレース

- Inputは、temporal datasetを出力するフィルタなら何でもよい
 - ▶ 通常、時間方向の補間が必要になるので、Temporal Interpolatorが無難

- Sourceはmultiblock datasetを使用できないので、Slice等は使えない
 - ➢ Point Source、Line Source、Plane Source等を利用
 - ▶ ParaView 3.4では改善



フィルタの例: 直線に沿ってデータをサンプリング

- 3Dばかりでなく、グラフ表示も可能
- 例: cavityケースの鉛直方向中心線に沿って、Uのy成分をプロットしたい







- もう少し、凝った輪郭を描くには?
- internalMeshのみを読み込んでいる場合

データ (*.OpenFOAM) → Extract Surface → Feature Edges

NIIGATA UNIVERSIT

フィルタの例: パッチでの流量の算出

- あるパッチを通過する流量を求めるには?
- ParaView 3.4以上が必要かもしれません

データ (*.OpenFOAM)
$$\longrightarrow$$
 Extract Block \longrightarrow Surface Flow パッチも選択 パッチを選択

- Spreadsheet ViewでSurface Flowの数値を確認
- 正確な数値が必要なら、calcMassFlowユーティリティがおすすめ
 - ➢ Surface Flowは、節点補間値から計算するため

ボリュームレンダリング



- 大規模データの可視化に有用
- 計算コストが高い
 - ➢ Projected Tetra = 比較的高速、ray cast = 高品質
- interFoam、interDyMFoamでの液相の表現にも有用(かも?)

```
▶ gamma = 0.5の前後でopacityを急峻に変える
```

・ ボリュームレンダリングの質にこだわるなら、Visltもおすすめ

まとめ

キャビティ流れの解析を例に、OpenFOAM・ParaViewを用いた解析をひととおり実習

- OpenFOAMケースフォルダの構造
- blockMeshによるメッシュの作成
- 境界条件・実行条件の与え方
- ソルバ(icoFoam)の実行
- ParaViewによる可視化