

1. シミュレーションの前に、基礎事項、専門用語いろいろ

1.1. シミュレーションとは

シミュレーションとは何か? →ぶっちゃけ「微分方程式をコンピュータを使って解くこと」

世の中の様々な物理現象の多くは微分方程式によって表すことができる。

1.2. 微分方程式とは

微分方程式とは? →「未知の関数の導関数を含む等式」

微分方程式の例:

点Pがx軸上を等速度 a で運動する場合、時刻 t ($t \geq 0$) におけるPの座標を x とすると、 x は t の関数であって、

$$\frac{dx}{dt} = a \quad \dots(1)$$

を満たす。このように未知の関数の導関数を含む等式を微分方程式という。



図 1.1: x 軸上を動く点P

では、「微分方程式を解く」とは? →微分方程式を満たす関数の式(解析解)、または値(数値解)を求めること。

上記の例を解く:

上の例の微分方程式(1)を式の形で解くには、 x を t の関数として表すことになる。そのためには、式(1)の両辺を t で積分すればよい。そうすると、

$$x(t) = at + C \quad \dots(2)$$

ただし、 C は積分定数である。

1.3. 初期条件とは

ただし、式(2)には積分定数 C が含まれている。この C を定めるには、さらに条件が必要である。一般にシミュレーションでは、時刻 $t = 0$ のときの x の値、すなわち $x(0)$ をあらかじめ決めておく(与える)ことで C の値が定まるようにする。この時刻 $t = 0$ において与える条件のことを初期条件という。

上記の例の C を定める:

例えば、 $t = 0$ において $x(0) = 1$ と決め、これを式(2)に代入すると $C = 1$ 。よって $x(t)$ は以下のように定まる。

$$x(t) = at + 1$$

1.4. 解析領域とは

室内の音の伝搬を調べる場合の室内、あるいはエンジンシリンダー内の燃焼を調べる場合のシリンダー内部のように、解析を行う対象の空間を解析領域という。コンピュータシミュレーションではコンピュータの演算能力やメモリ容量に限界があるので、無限の領域を解析することは出来ない。したがってどのような解析でも、解析領域を必ず何らかの形で決めなければならない。

1.5. 境界条件とは

解析領域の端部では、例えば固定された硬い壁は「壁の法線方向の速度が0」といったように、端部の物理的な性質を数学的に表すような条件の指定が必要である。これを境界条件という。

1.6. コンピュータは微分方程式をどのように解くか？

実際にシミュレーションで解かれる微分方程式は式(1)より遥かに複雑なので、一般的には式(2)のように解析解を求めることは出来ない。したがって数値解を求めることになる。

数値解を求めるには、まず、微分方程式に含まれる微分を、微分の定義式から \lim を外した形、すなわち式(1)であれば

$$\frac{dx}{dt} = \lim_{\Delta t \rightarrow 0} \frac{x(t+\Delta t) - x(t)}{\Delta t} \approx \frac{x(t+\Delta t) - x(t)}{\Delta t} \quad \dots(3)$$

のような近似式に置き換える。この式の形を差分と言い、微分方程式の微分を差分に置き換えることを離散化と言う。ここで Δt (時間刻み)は解析を行う者が適当に決める。

式(1)の数値解を求める:

式(3)を式(1)に代入して離散化すると

$$\frac{x(t+\Delta t) - x(t)}{\Delta t} = a$$

さらに両辺に Δt をかけ、整理すると

$$x(t+\Delta t) = x(t) + a\Delta t \quad \dots(4)$$

が得られる。

例えば、 $a=2$ 、 $\Delta t=0.1$ とし、さらに初期条件として、 $t=0$ のとき $x(0)=1$ を与えると、式(4)から

$$\begin{aligned} x(0.1) &= 1 + 0.2 = 1.2 \\ x(0.2) &= 1.2 + 0.2 = 1.4 \\ x(0.3) &= 1.4 + 0.2 = 1.6 \\ &\dots \end{aligned}$$

のように、 $t=0.1$ 、 $t=0.2$ 、 $t=0.3$ 、 \dots の値を順に求めていくことが出来る。

1.7. メッシュ(格子)、セル

式(1)は時間のみが微分変数であるが、実際にシミュレーションで解かれる方程式には空間座標(x, y, z)も微分変数として含まれる。従って空間に対しても離散化を行い、解析領域を網の目のように細かく分割する。分割された空間をメッシュ(格子)、一つ一つの細かく分割された空間(網の目)

をセルと言う。

1.8. シミュレーション手順の概要

大まかなシミュレーションの手順としては、図 1.2 のように、解析対象の形状入力・メッシュ作成、初期・境界条件設定、解析実行、解析結果の表示となる。全ての段階を一つのプログラムで行うシミュレーションソフトもあるが、OpenFOAM では各段階で別々のプログラムを使うのが一つの特徴となっている。

各プログラムを呼び出す方法として、OpenFOAM では「FoamX」というアプリケーションを使って操作する方法、キーボードからコマンドを入力して操作する方法の 2 種類がある。上級者の多くはコマンド入力で操作するが、ここでは比較的初心者向けの FoamX を使った操作を行う。

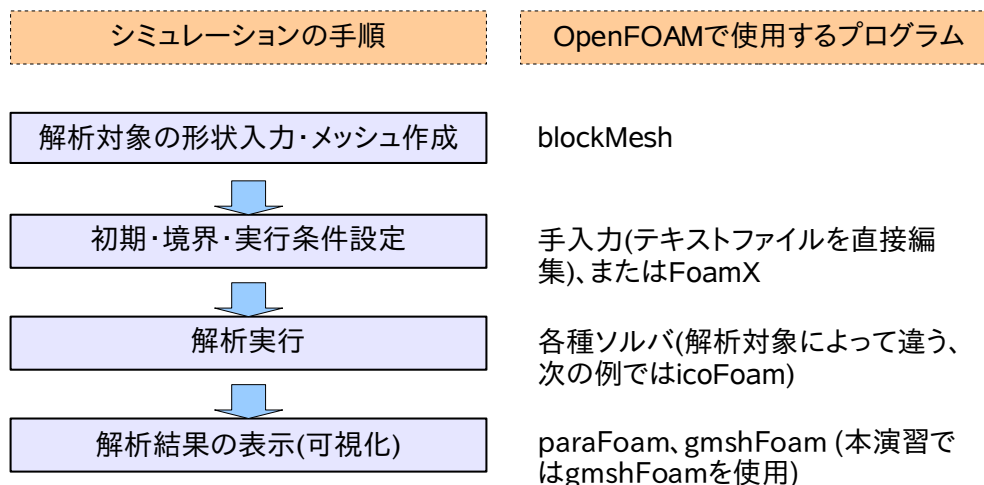


図 1.2.: シミュレーションの手順

2. シミュレーション「キャビティ流れ問題」

2.1. 概要

「キャビティ」とは空洞のことで、空洞の周囲の壁の一部を動かしたときに空洞内部の流体がどのように動くかをシミュレーションする(図 2.1)。流体の流れを解くシミュレーション(略称して CFD -- Computational Fluid Dynamics; 数値流体力学--と呼ぶ)として、最もポピュラーな解析ケース(解析する個々の問題のことを「ケース」という)である。シミュレーションソフトの性能評価や操作法習熟のための例題として、ほぼ必ず使われる問題である。

2.2. CFD とは

流体とは？ →外から加えられた力に対して容易に変形する物質のこと。一般には液体、気体。

CFD とは？ →流体の力学的な挙動を解析すること(先端的な応用では化学的な挙動も同時に解析する。燃焼問題など)。

具体的に何を解くのか？ →流体の力学的な状態は、流体の速度、および圧力によって表されることから、これらが変数となる。従って CFD では、メッシュ中の全てのセルでの速度と圧力の値を解く。

実際に解く微分方程式はどのような式か？ →座標 (x, y, z) 、時刻 t における流体の速度を

$\vec{U}(x, y, z, t) = (u, v, w)$ 、圧力 $p(x, y, z, t)$ とすると、以下の連立偏微分方程式となる。

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0,$$

$$\frac{\partial u}{\partial t} + \frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} + \frac{\partial uw}{\partial z} = \frac{\partial p}{\partial x} + \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right),$$

$$\frac{\partial v}{\partial t} + \frac{\partial uv}{\partial x} + \frac{\partial v^2}{\partial y} + \frac{\partial vw}{\partial z} = \frac{\partial p}{\partial y} + \nu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right),$$

$$\frac{\partial w}{\partial t} + \frac{\partial uw}{\partial x} + \frac{\partial vw}{\partial y} + \frac{\partial w^2}{\partial z} = \frac{\partial p}{\partial z} + \nu \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \right)$$

ここで ν は「動粘性係数」と呼ばれ、流体の「粘り気」を表す量である。値が大きい程「粘り気」が強いことを表す。空気の場合、 $1.5 \times 10^{-5} \text{ m}^2/\text{s}$ 、水は $1.0 \times 10^{-6} \text{ m}^2/\text{s}$ である。

2.3. 解析対象

図 2.1 のとおり、正方形の解析領域。上端の壁が x 軸方向に 1.0 m/s の速度で動き、その他の壁は動かない。

動粘性係数 $\nu = 0.01 \text{ m}^2/\text{s}$ とする。

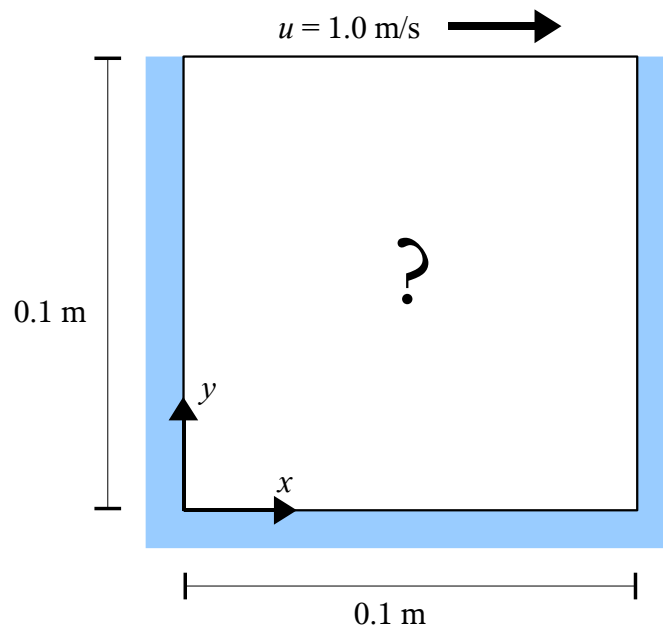


図 2.1: キャビティ流れの解析領域

2.4. 解析の実行手順

OpenFOAM では基本的に、解析ケースごとにケース専用のフォルダを作成する(以下、ケースフォルダと呼ぶ)。本解析ケースの場合は、解析に必要な入力データのほとんどが既に準備されており、以下のフォルダに存在する。

```
c:\cygwin\home\ユーザ名\OpenFOAM\ユーザ名-1.3\run\tutorials\icoFoam\cavity
```

ここで「ユーザ名」は Windows にログオンする際のユーザ名である。各ケースフォルダの下は、図 2.2 のような構造になっている。

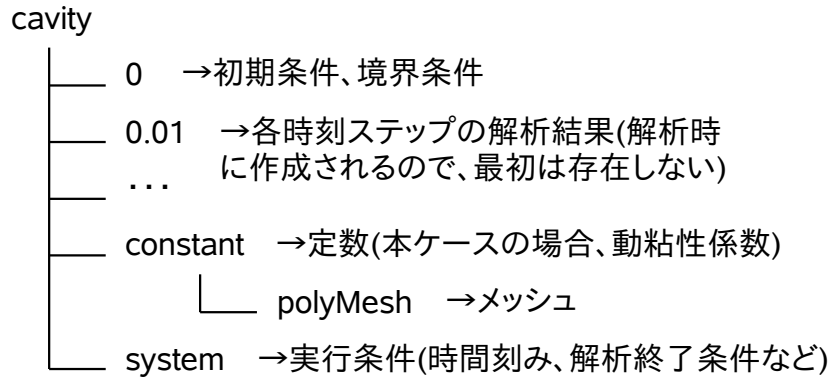


図 2.2.: ケースフォルダの構造

2.4.1. キャビティ流れの解析ケースを開く

まず、FoamX を起動する。「スタート」→「プログラム」→「OpenFOAM」→「FoamX」を選択する。

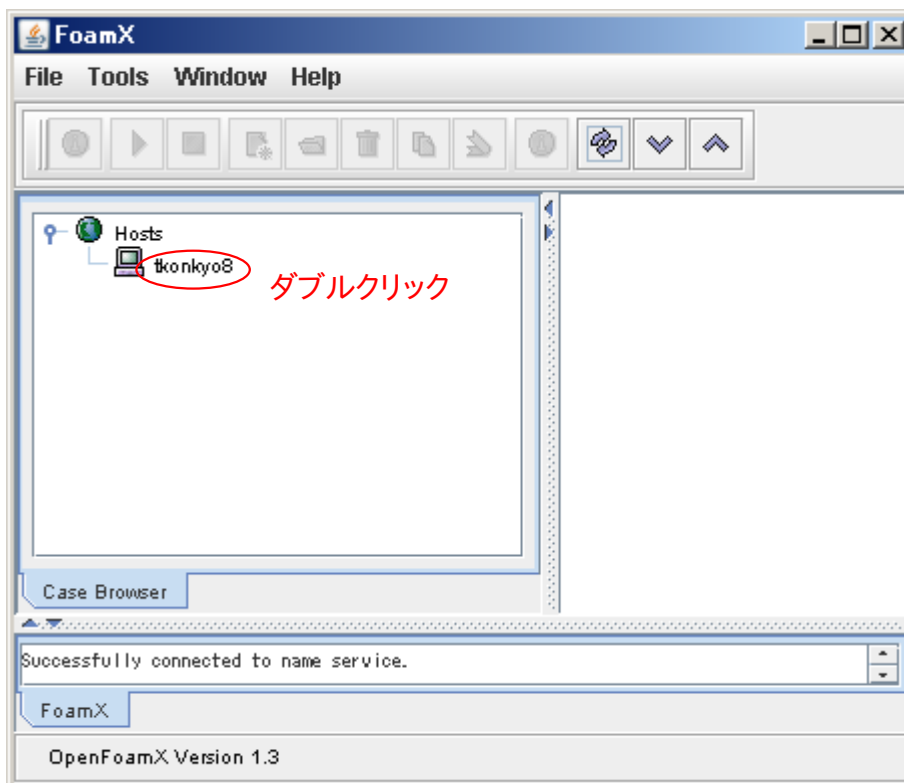


図 2.3: FoamX 起動直後の画面

FoamX が起動したら、ホスト名をダブルクリックし、さらに「\$FOAM_RUN/tutorials/icoFoam」→「cavity」をダブルクリックすると、キャビティ流れの解析ケースが開く。

2.4.2. メッシュの作成

ここでは図 2.1 の解析領域を x 軸、 y 軸方向ともに 20 分割したメッシュを作成する。また、図 2.1 のとおり、解くべき問題自体は 2 次元であるが、OpenFOAM のメッシュは必ず 3 次元で作成する必要がある。そのため 2 次元問題のメッシュを作成する場合には、 z 方向の高さを適当に決めてセル数を 1 にする。ここでは z 方向の高さを 0.01 とする。

解析領域の端点に番号を付ける必要があるため、直方体形の8つの端点に0~7(0番から始まることに注意)の番号を図2.4のように割り当てる。またOpenFOAMでは全ての境界面に名前を付ける必要があるため、図2.4に示した名前を付ける。

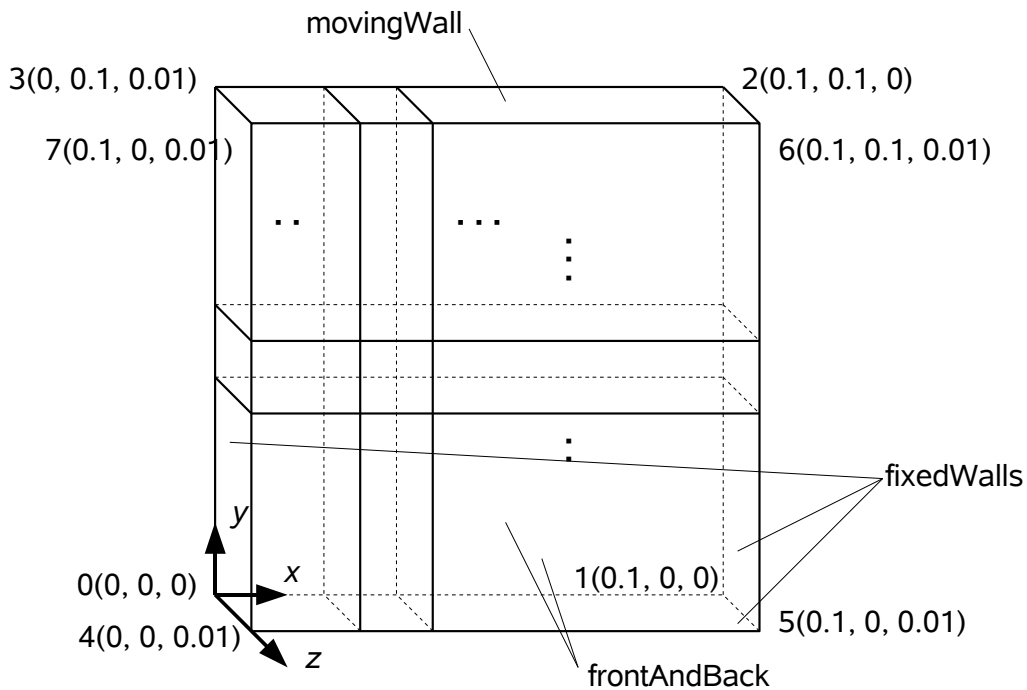


図 2.4: 解析領域のメッシュ分割

以上をOpenFOAMのメッシュ設定ファイルに記述すると以下ようになる。なおこのファイルは、ケースフォルダの中の

constant¥polyMesh¥blockMeshDict

である。

```

/*-----*/
| ===== |
|  \ \      /  F i e l d           | OpenFOAM: The Open Source CFD Toolbox |
|  \ \      /  O p e r a t i o n   | Version: 1.3 |
|   \ \ /    A n d                 | Web: http://www.openfoam.org |
|    \ \ /    M a n i p u l a t i o n |
/*-----*/

FoamFile
{
    version      2.0;
    format       ascii;

    root         "";
    case         "";
    instance     "";
    local        "";

    class        dictionary;

```


blockMeshDict の内容を確認したら、図 2.5 のとおり、FoamX 画面で「cavity」をマウスの右ボタンでクリック→「Foam Utilities」→「mesh」→「generation」→「blockMesh」を選択して、blockMesh プログラムを走らせる。

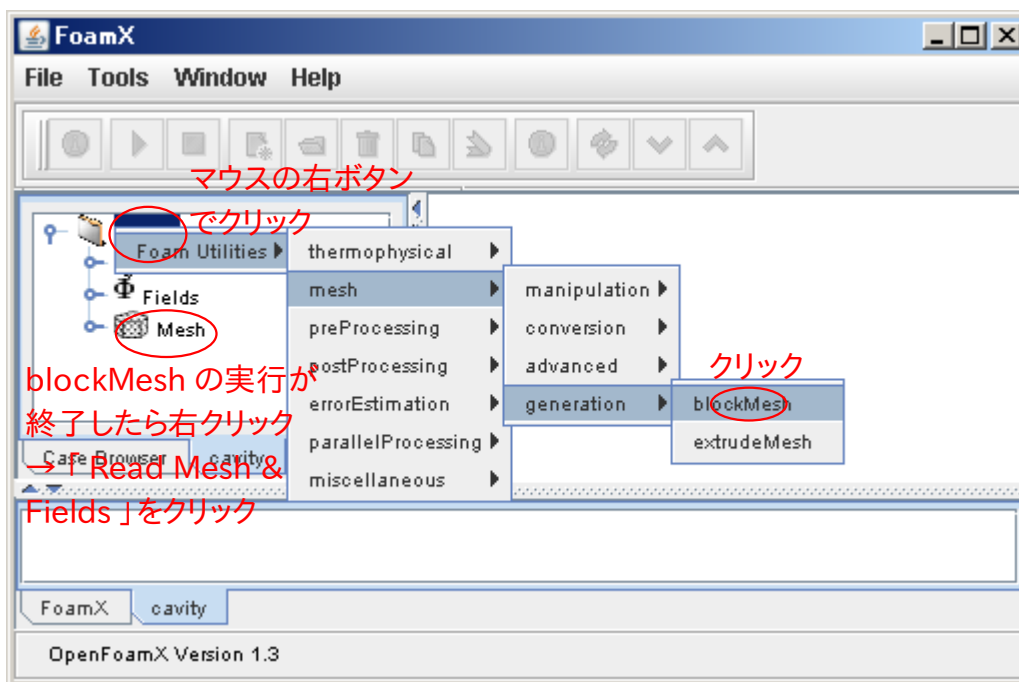


図 2.5: blockMesh プログラムの選択

すると図 2.6 の画面が開くので、「Execute」ボタンをクリックする(ここで「Edit dictionary」ボタンを押すと blockMeshDict ファイルを編集することが出来るが、ここでは編集不要)。blockMesh の実行が終わったら、ケースフォルダの下に constant/polyMesh ディレクトリに「boundary」、「faces」、「neighbor」、「owner」、「points」というファイルが作成される。これらがメッシュファイルである。「Mesh」を右クリックして「Read Mesh&Fields」をクリックし、メッシュを読み込む。

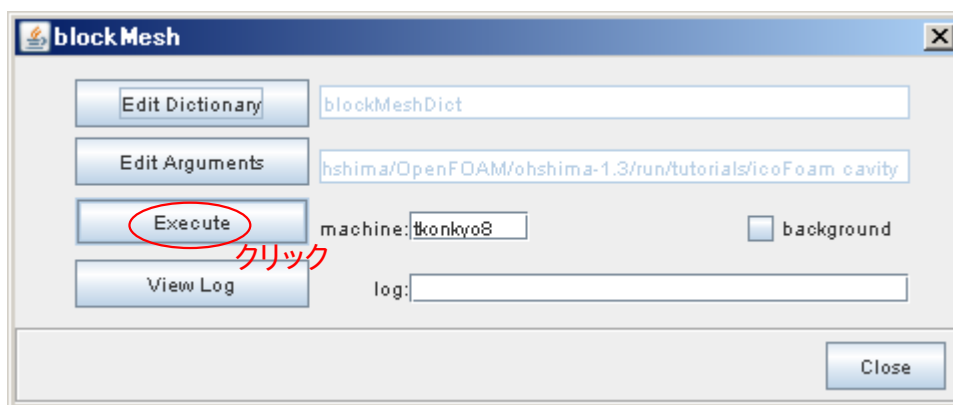


図 2.6: blockMesh 起動画面

2.4.3. 境界条件・初期条件の作成

メッシュを読み込んだら、図 2.7 に示す「Mesh」の左側の鍵マークをクリックして、さらに「Patches」の左側の鍵マークをクリックする。「movingWall」、「fixedWalls」、「frontAndBack」をダブルクリック

して、それぞれメッシュ生成時に blockMeshDict で指定したとおり、boundary type が「wall」、「wall」、「empty」となっていることを確認する。これらが境界条件の種類となる。

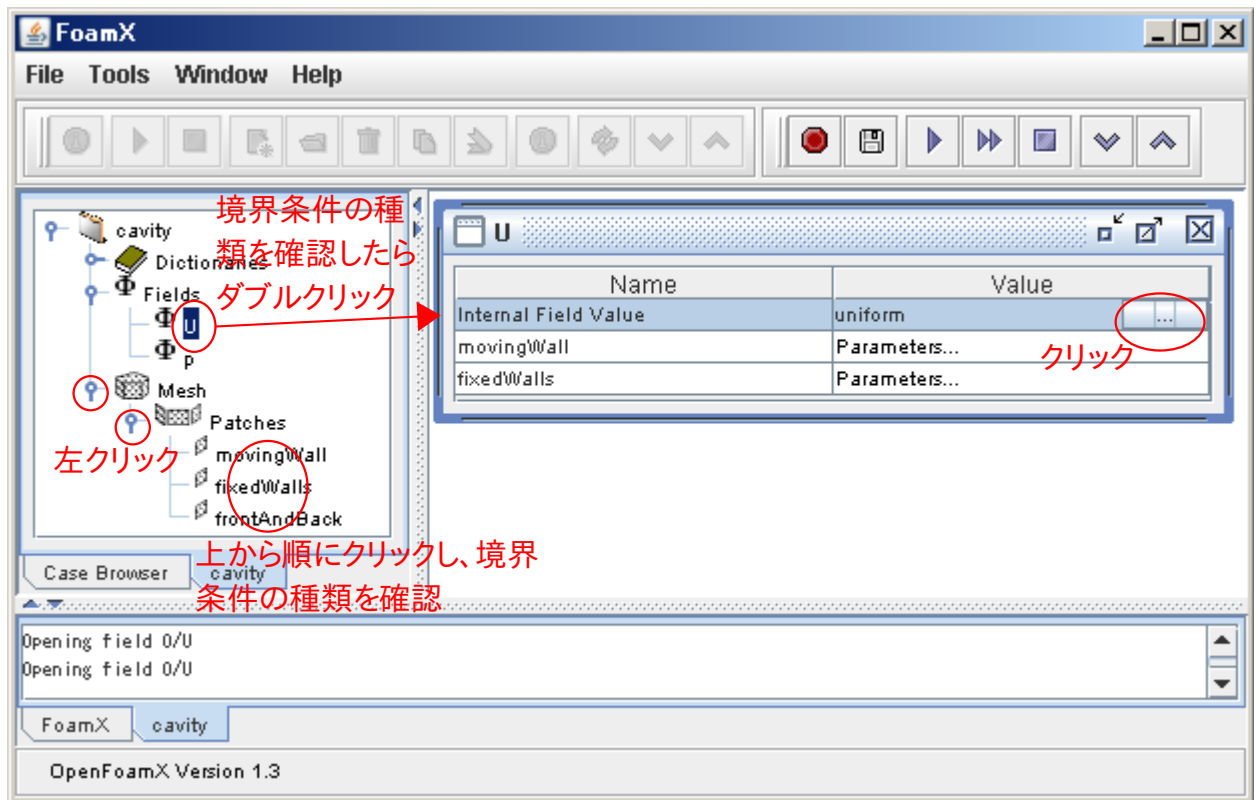


図 2.7: 境界条件・初期値の指定

つぎに境界値と解析領域内部の初期値を指定する。「Fields」の左側の鍵マークをクリックすると、「U」、「p」の2項目が現れる。「U」は速度、「p」は圧力に関する項目である。まず「U」をダブルクリックし、「Internal Field Value」をクリック、さらに右端の「...」をクリックして現れる画面で「uniform 0.0 0.0 0.0」となっていることを確認する。これは解析領域内部の速度の初期値が全て0であることを示している。次に同様に「movingWall」をクリック、今度は速度の x 成分が1.0となっていることを確認する。これは x 軸方向に1.0 m/sで動く壁を境界条件として指定したこと示す。「fixedWall」は固定された壁という境界条件なので、全て0であることを確認する。

「p」については初期値のみを指定する。「Internal Field Value」が0であることを確認する。

これらの境界条件・初期条件はケースフォルダの下の「0」というフォルダに作成される。各変数につき1つのファイルが、変数の名前をファイル名として作成される。本ケースでは「U」、「p」が作成される。

2.4.4. 動粘性係数の設定

動粘性係数の設定は「Dictionaries」、「transportProperties」の左側の鍵マークをクリックし、「nu」をダブルクリックすると現れる画面で「0.01」が指定されていることを確認する。

2.4.5. 実行条件の設定

時間刻みなどの実行条件は「Dictionaries」→「controlDict」で設定する。設定項目は多数あるが、特に重要なのは「endTime」、「deltaT」、「writeInterval」の3つである。

「endTime」では、解析を終了する解析上の時刻を設定する。ここでは「0.5」(0.5秒)を設定する。

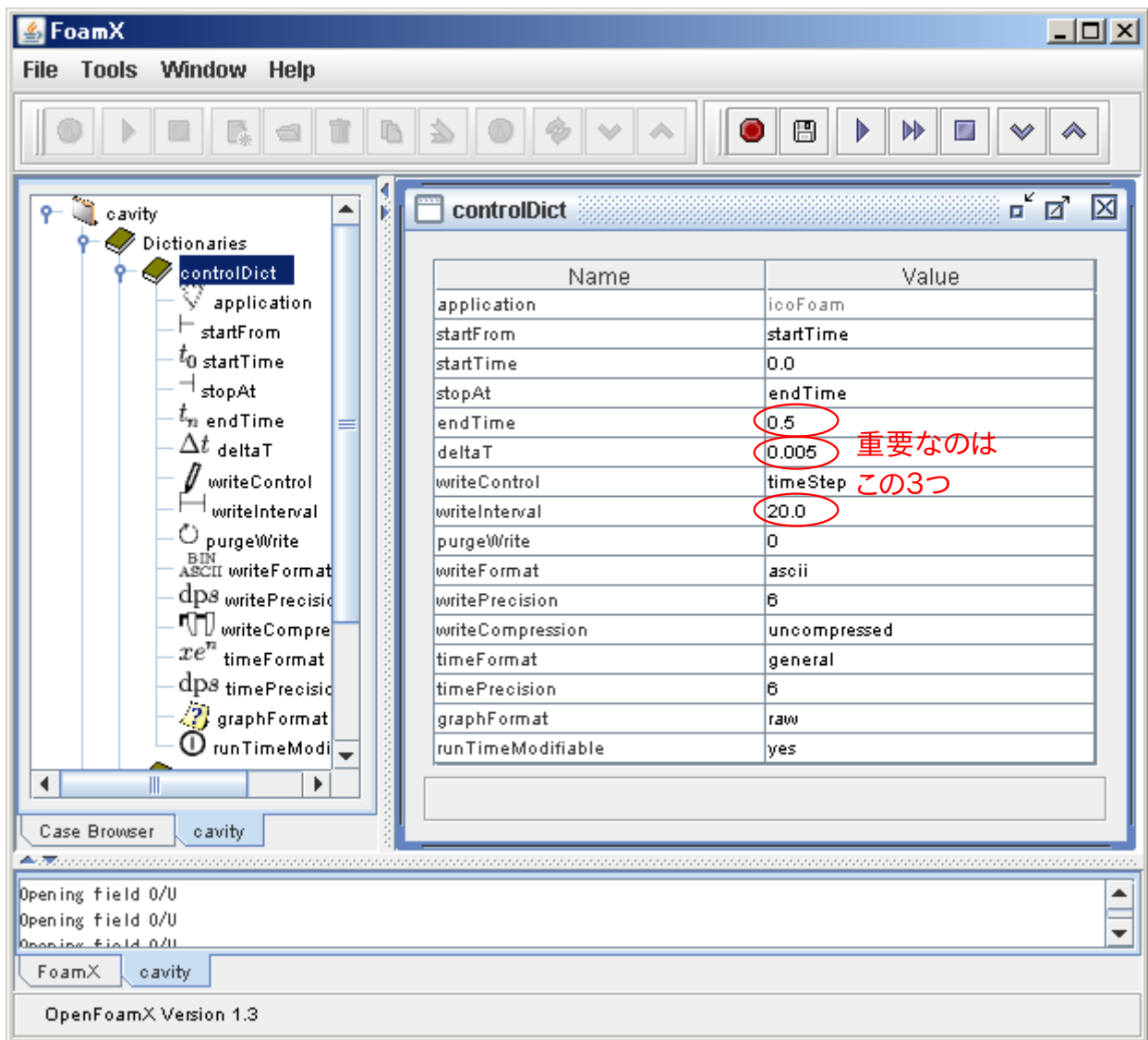


図 2.8: controlDict の設定項目

「deltaT」では時間刻みを設定する。一般には時間刻みを大きく取れば計算時間は少なくて済むが、シミュレーションの精度は悪くなる。また本ケースでは、以下の式を満たすように時間刻みを設定しなければならない。

$$C = \frac{|\vec{U}| \Delta t}{\Delta x} < 1 \quad \dots (5)$$

ここで $|\vec{U}|$ は解析領域内の速度の最大値、 Δt は時間刻み、 Δx は流体が流れる方向のセルの幅である。この C の値をクーラン数と言う。式(5)は、直感的に解釈すると「時間刻みごとの流体の移動距離が、セルの大きさを超えてはならない」ことを表している。本ケースに限らず、多くのシミュレーションでクーラン数による制限が存在する。本ケースでは $|\vec{U}|$ の最大値を movingWall の速さの 1.0 m/s、 Δx を $0.1/20 = 0.005$ m と考え、 Δt を逆算すると

$$\Delta t < \frac{\Delta x}{|\vec{U}|} = \frac{0.005}{1.0} = 0.005 \text{ [秒]}$$

となるので、この値をセットする。

「writeInterval」では、時間刻みを何回進めるごとに解析結果を保存するかを指定する。例えば 20 と設定すると、計算が $0.005 \times 20 = 0.1$ [秒] 進むごとに解析結果を保存することになる。時間刻みを進めるごとに解析結果を保存するとデータ量が大量になり、また保存処理にも余分な時間がか

かるので、通常はこの程度の値を使用する。

2.4.6. データの保存

以上の設定が終わったら、FoamX 画面上端のフロッピーディスクのアイコンをクリックして設定を保存する。

2.4.7. メッシュの確認

これで解析を実行する準備が整ったが、その前にメッシュと初期条件を表示して確認しておこう。図 2.5 と同様にして「cavity」で右クリック→「Foam Utilities」→「gmshFoam」をクリックし、さらに「Execute」をクリックすると、図 2.9 のような画面が現れる。

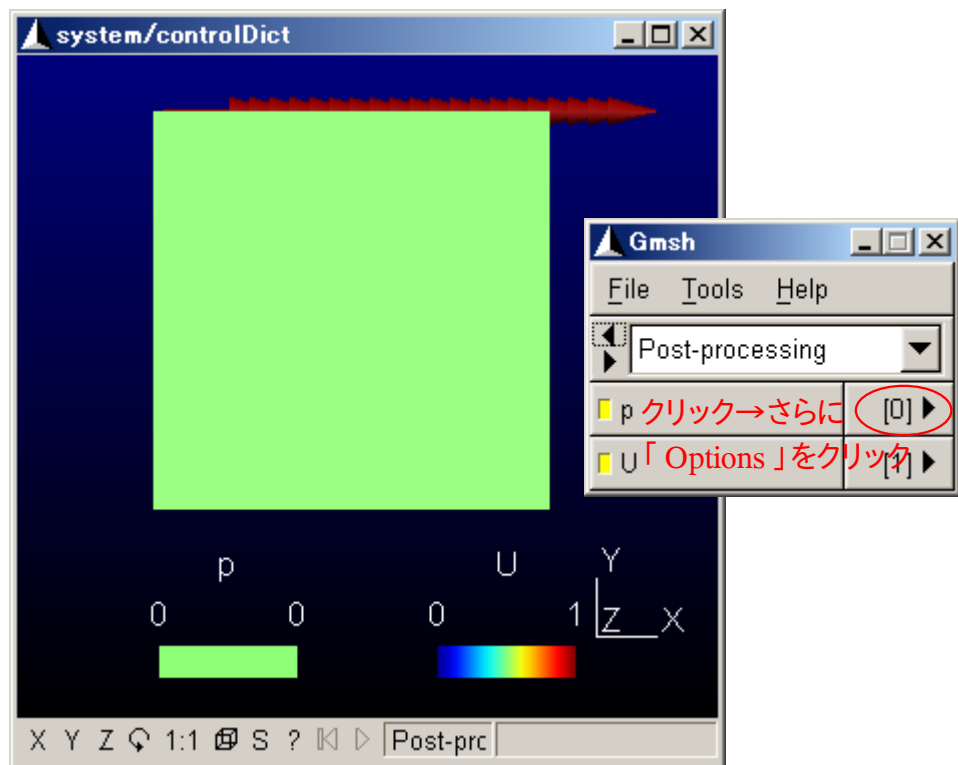


図 2.9: gmshFoam 初期画面

図 2.9 の赤丸で囲んだ部分をクリックし、開いたメニューの一番下の項目「Options...」をクリックすると、図 2.10 の画面となる。図で示した通りに順にクリックしていくと、図 2.11 のようにメッシュが表示される。図 2.11 では、メッシュの色が圧力の値、矢印の大きさと色が速度の値を表している。圧力は解析領域全体で 0 となっており、速度は movingWall の境界面速度のみ矢印の色から 1 となっていることがわかる。よってメッシュ、初期条件、境界条件が正しくセットされていることがわかる。

gmshFoam を終了するには、「File」→「Quit」を選択する。

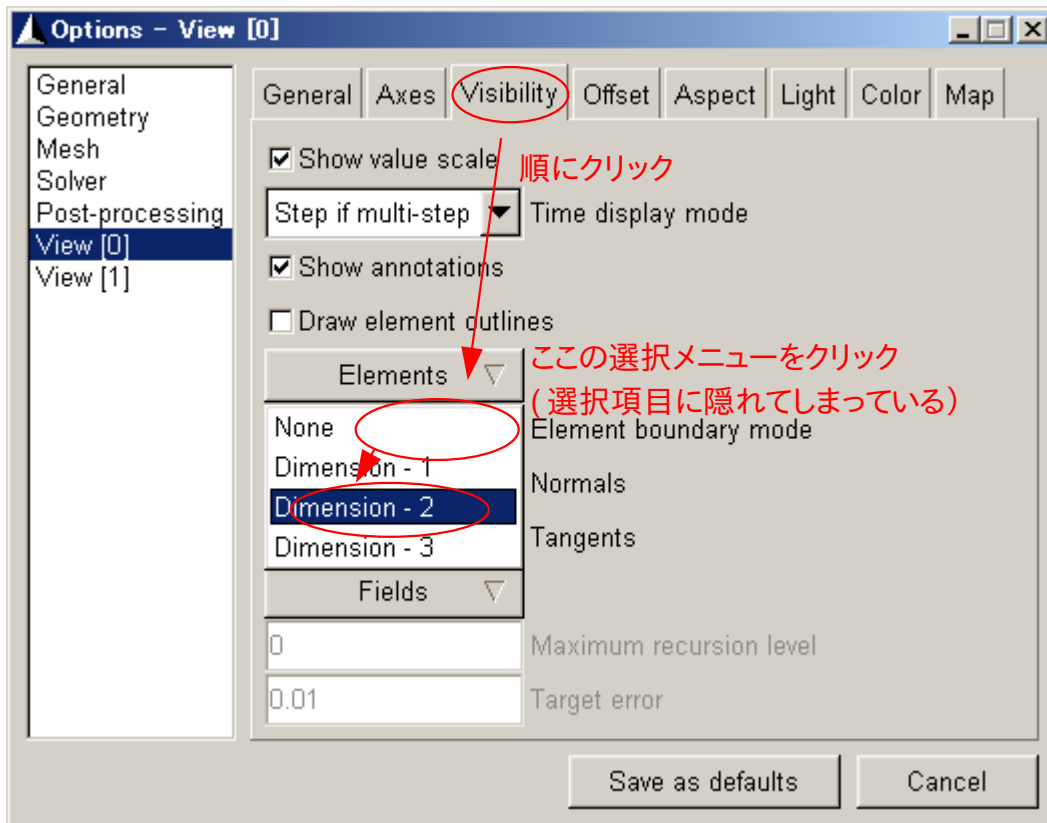


図 2.10: Options 画面

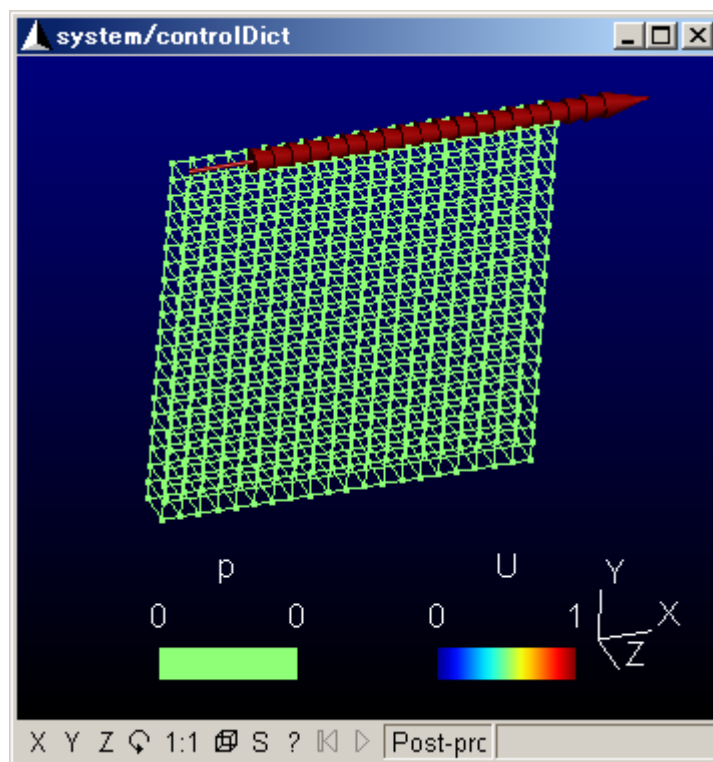


図 2.11: メッシュ表示

2.4.8. 解析の実行

解析を実行するには、FoamX 画面上端の「>>」をクリックする。本ケースは非常に小さなケースな

ので、解析は数秒で終了する。

2.4.9. 結果の表示

解析が終了したら、結果を表示させてみよう。2.4.7 節に述べたのと同じ手順で `gmshFoam` を起動する。起動時の設定では解析上の時刻が判りにくいので、2.4.7 節と同様にして Option 画面を開き、「Value if multi-step」を選択する。「View[1]」についても、同様に操作する。

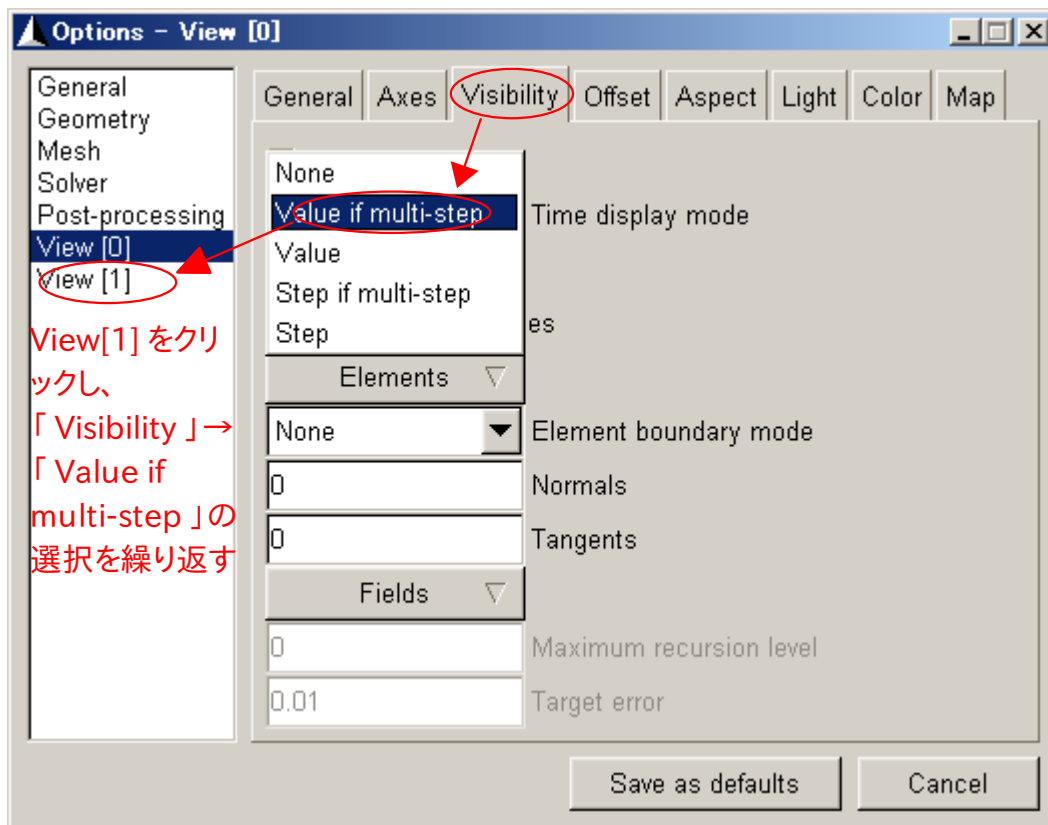


図 2.12: 解析上の時刻の表示

時刻を進める・戻すのは、図 2.14 に示すように、グラフィック表示ウインドウ下端のボタンで操作する。本ケースの場合、時刻 0.1 秒以降は解析結果がほとんど変わらないが、計算終了時刻の 0.5 秒まで進めてみよう。

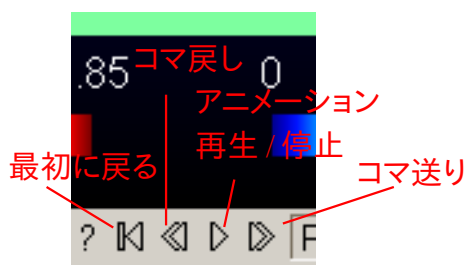


図 2.14: アニメーション再生ボタン

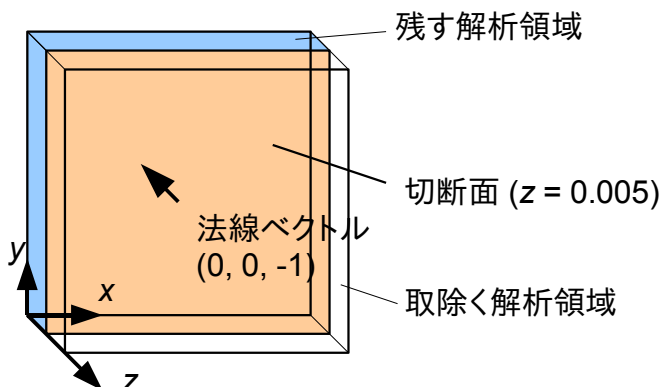


図 2.13: 切断面の設定

さらに解析領域境界面に囲まれて内部が見えないので、図 2.13 のように、解析領域を切断面によって薄切りにして内部が見えるようにする。解析領域の切断面を設定するには、まず切断面の方程式を

$$Ax + By + Cz + D = 0$$

の形で決める必要がある。ここでは z 軸方向の解析領域中心である $z = 0.005$ の面で切断することとすると、 A, B, C, D の組合せは、 C, D の符号によって

$$A=0, B=0, C=1, D=-0.005 \quad \text{または} \quad A=0, B=0, C=-1, D=0.005$$

の 2とおりが存在することになる。gmshFoam の場合、「平面の法線ベクトル(A, B, C)の向いている方向の解析領域が切断後に残され、法線ベクトルと逆方向の解析領域が切り取られる」仕組みになっているので、画面手前側が切り取られるように法線ベクトルとしては $(0, 0, -1)$ を指定することになる。従って A, B, C, D の組合せは

$$A=0, B=0, C=-1, D=0.005$$

と決まる。

このように決まった切断面を入力するには、図 2.15 のように「Tool」メニューから「Clipping Planes」を選択し、続いて現れる図 2.16 の画面で A, B, C, D の値を入力する。

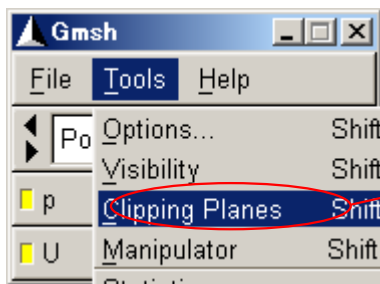


図 2.15: 切断面の入力画面を開く

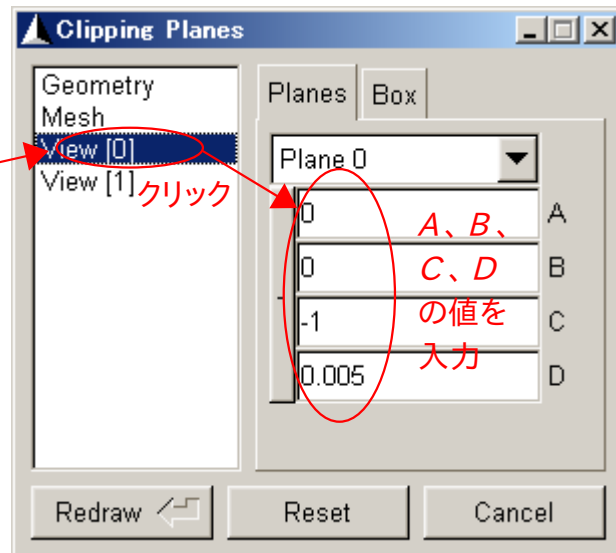


図 2.16: 切断面の入力

すると図 2.17 のように、内部の速度ベクトルが見えるようになる。

図中の解析領域内の塗り潰しは、色で圧力を表している。青色が圧力の低い部分を、赤色が圧力の高い部分を表している。矢印は流体の速度ベクトルで、矢印の大きさで速度の大きさ(速さ)を表している。圧力は解析領域左上端で小さく、画面右端で大きくなっていることが判る。速度ベクトルの表示からは流体が解析領域全体を周回する渦を描いており、渦の中心は解析領域やや上寄りにあることがわかる。速さは解析領域上端の movingWall の近くが大きく、解析領域左右下端で小さくなっていることがわかる。

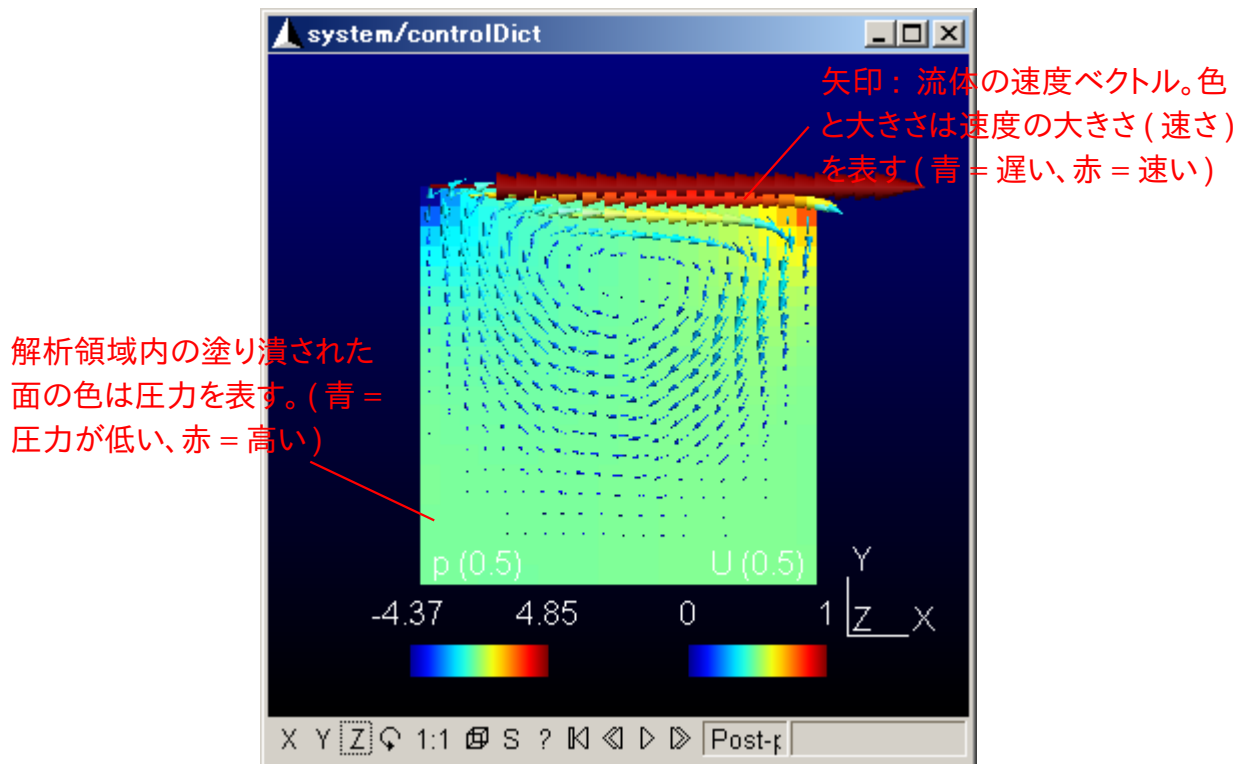


図 2.17: 解析結果の切断表示

2.4.10. 流線を描く

今度は解析領域内に質量の無い粒子を散らしたとき、粒子が流れの中でどのように動くか、その軌跡を描いてみよう。このような軌跡のことを流線という。流線の計算原理は、粒子の初期位置 $(x, y, z) = (x_0, y_0, z_0)$ を適当に決定し、式(4)を x, y, z それぞれについて計算する。さらに各時刻における粒子位置の間を直線で結ぶことで、近似的に流線が計算される。ただし、式(4)では定数であった速度 a は、各時刻での粒子の位置に対応したセルの速度の値が使用される。

まず、流線が見やすいように図 2.19 のようにして速度・圧力の描画を消去し、図 2.18 の流線描画設定画面を表示させる。粒子は点 (X_0, Y_0, Z_0) を起点とする 2 つのベクトル $\vec{U} = (X_1, Y_1, Z_1) - (X_0, Y_0, Z_0)$ 、 $\vec{V} = (X_2, Y_2, Z_2) - (X_0, Y_0, Z_0)$ によって張られる平面に均等に配置される。ここでは解析領域全体に均等に粒子を配置することになると、

$$(X_0, Y_0, Z_0) = (0, 0, 0.005), (X_1, Y_1, Z_1) = (0.1, 0, 0.005), (X_2, Y_2, Z_2) = (0, 0.1, 0.005)$$

とするのがよい。粒子の個数は \vec{U} 、 \vec{V} 方向ともに 10 個ずつ、平面全体で合計 $10 \times 10 = 100$ 個とする。時間刻み Δt は式(5)と同様の理由から 0.005 秒とし、100 時刻ステップを計算することになると、ちょうど解析終了時刻の 0.5 秒まで計算されることになる。

以上の数値を図 2.18 のように入力し、最後に「Run」をクリックする。さらに、このままでは解析領域の境界が判らず見にくいので、図 2.20 に示すボタンをクリックしてオプション設定画面を表示させ、図 2.21 のとおり入力して箱型の座標軸を表示させる。すると最終的には図 2.22 の画面となるので、再生ボタンを押すと流線描画のアニメーションが開始される。

ここをクリックし、さらに「Plugins」→
「Stream Lines...」をクリック
クリックして消す

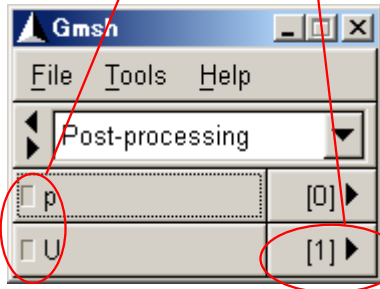


図 2.19: 速度・圧力描画の
消去

ここをクリックし、さらに「Options...」
をクリック

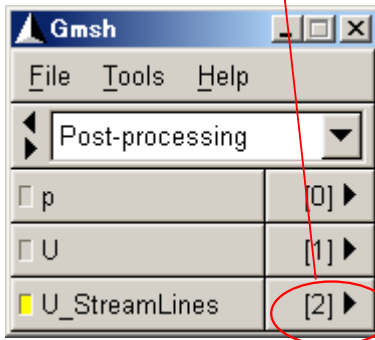


図 2.20: オプション設定画
面の呼出し

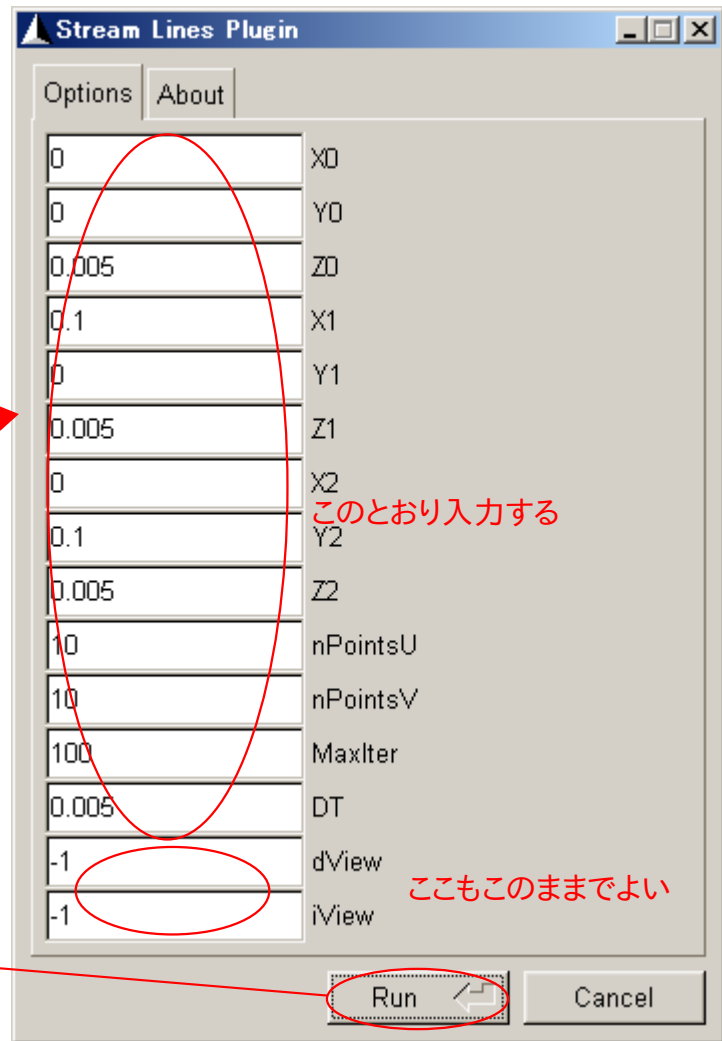


図 2.18: 流線描画の設定

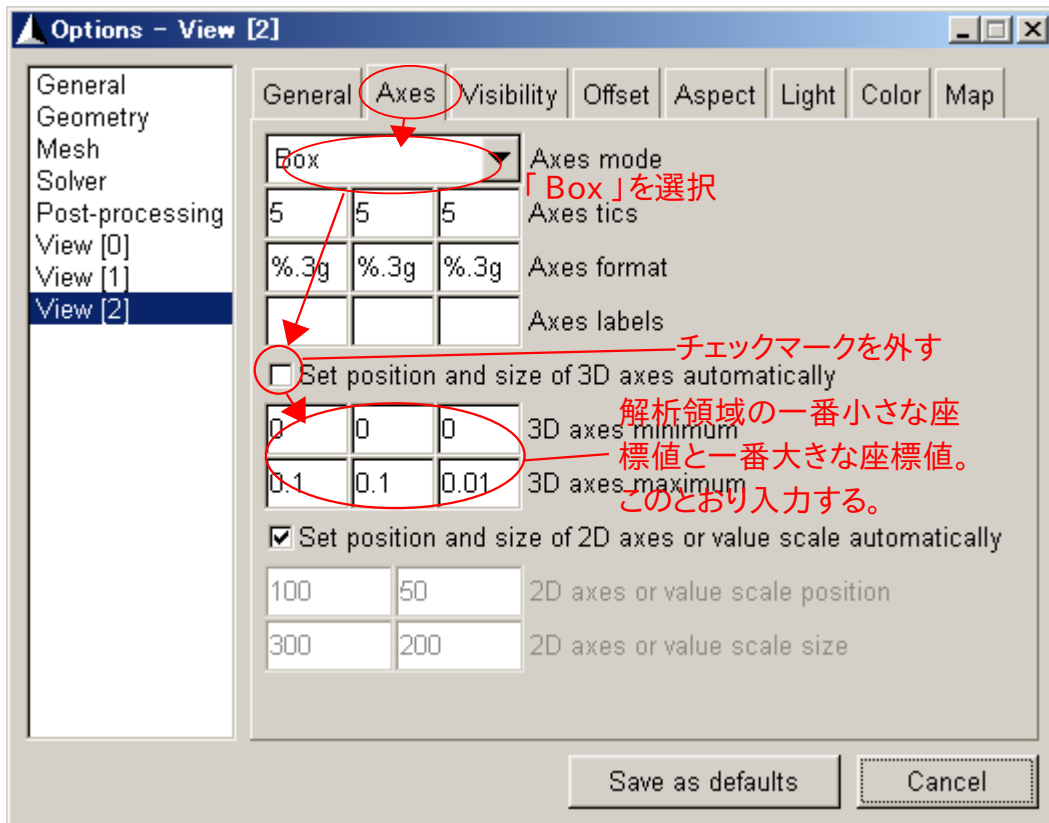


図 2.21: 座標軸の表示設定

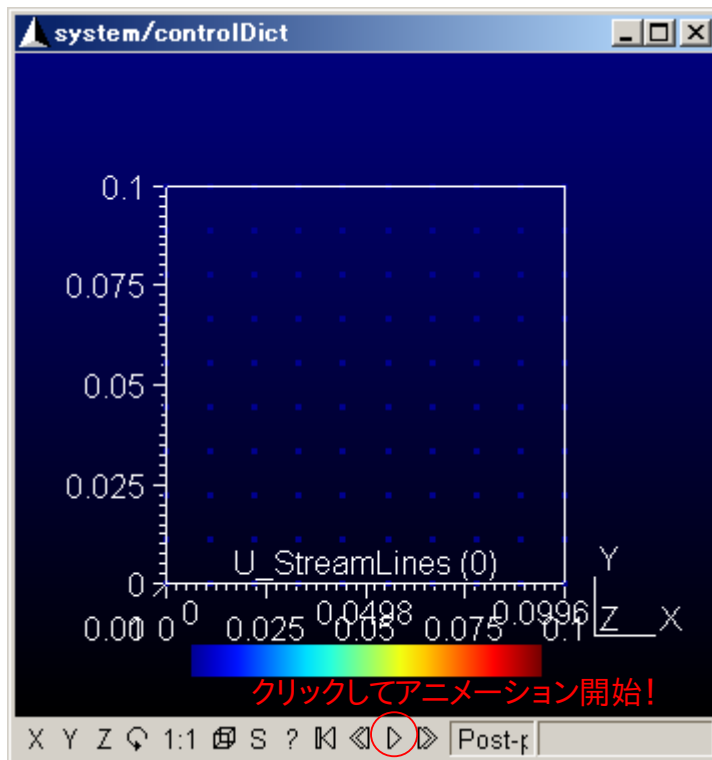


図 2.22: 流線表示画面(初期状態)

100 ステップ経過後の最終状態は図 2.23 のようになる。色は各粒子の初期位置からの移動距離で、青は移動距離の短い部分、赤は移動距離の長い部分を表す。画面上端の movingWall 以外の壁面近くでは一般に流れが遅く、また画面左右下端のほか、中央やや上部の渦中心部にも、

流体がほとんど動かず、淀みとなっている部分があることがわかる。

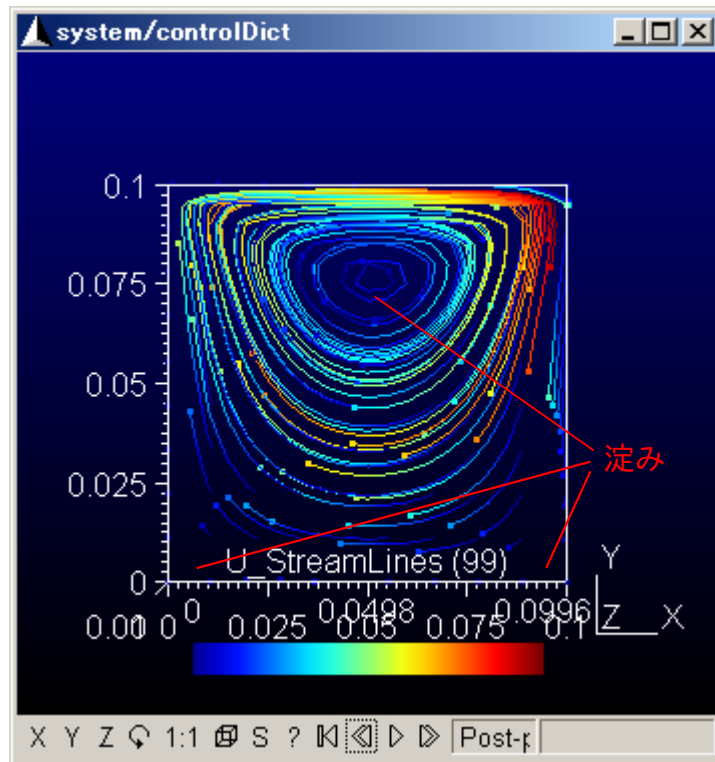


図 2.23: 流線表示画面(最終状態)

画像を保存するときは、「File」→「Save As...」をクリックして現れる画面で「Format:」には「PNG (*.png)」を選択し、「Filename:」欄に保存したいファイルの名前(ファイル名の最後を必ず「.png」とすること)を入力する。

以上で、シミュレーションの手順をひとつおりのり行ったことになる。

3. 演習問題

2章で行ったシミュレーションを、 x 軸、 y 軸方向のセル数を倍の 40×40 として行ってみよう。流線を描くときも、 \vec{U} 、 \vec{V} 方向の粒子数を倍の 20×20 とすること。

解析ケースを図 3.1 に従って「cavity」フォルダから「cavityFine」フォルダにコピーして行うこと。

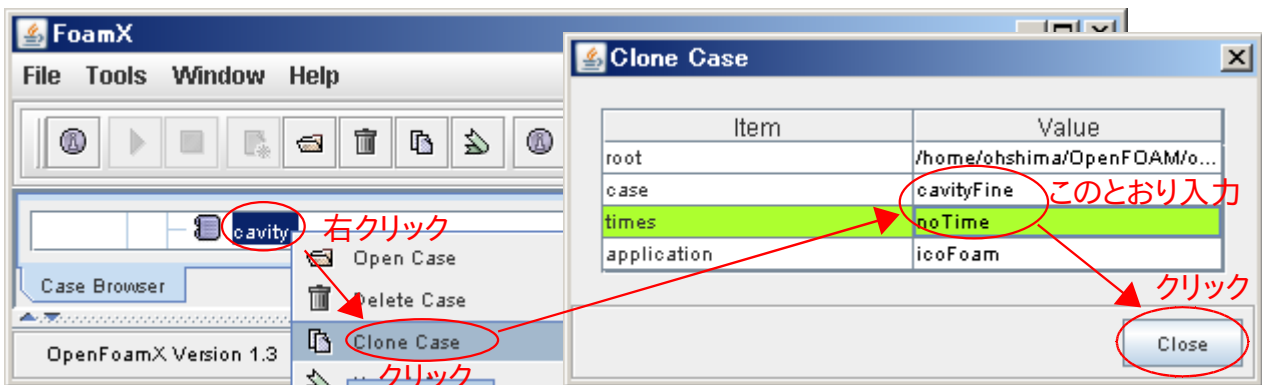


図 3.1: 解析ケースのコピー

境界条件(movingWall の速度)、解析・流線計算の時間刻みの設定、流線計算の繰り返し回数に気を付けること。cavity と cavityFine の解析結果の違いで気づいた点をレポートにまとめよう。