

2024/02/26

都市デジタルツインの実現に向けた研究開発及び実証調査業務

熱流体シミュレーションシステムの開発

要件定義資料

(要件定義書／基本設計書及び内部設計書)

別紙

シミュレーションモデル仕様書

株式会社構造計画研究所

目 次

I 本書の概要	1
I-1. 本書の位置づけ	1
I-2. 本書が前提とするソフトウェア・ライブラリ	1
II 熱流体解析手法	2
II-1. 基礎方程式	3
II-2. 採用ソルバー	3
II-3. OpenFOAM のインストール方法	4
II-4. OpenFOAM の処理手順	5
III OpenFOAM の入力仕様	8
III-1. OpenFOAM のフォルダ構成	8
IV 入力データ	9
IV-1. system フォルダ (メッシュ生成条件)	12
IV-2. constnat フォルダ (物性値)	14
IV-3. 0.orig フォルダ (初期値、境界条件)	15
V 出力データ	21
V-1. 計算結果ファイル	21
V-2. ログファイル	21
VI 熱流体解析ソルバの切り替え手順	22
VI-1. OpenFOAM 用テンプレートの作成	22
VI-2. OpenFOAM の実行スクリプト	27
VI-3. OpenFOAM 用テンプレートのアップロードと利用	28

I 本書の概要

I-1. 本書の位置づけ

本書は、熱流体シミュレーションシステムの開発 要件定義資料（要件定義書／基本設計書及び内部設計書）の別紙であり、内部モジュールインターフェースである熱流体解析シミュレーションモデル仕様について記述する。

I-2. 本書が前提とするソフトウェア・ライブラリ

要件定義資料[III-3-②.利用するソフトウェア・ライブラリ]記載の通り、本システムでは熱流体解析ソフトウェアとして「OpenFOAM (ESI-OpenCFD)」を採用する。

OpenFOAM は下記のサイトから入手可能であり、各種ドキュメントについても参照する事が可能である。

<https://www.openfoam.com/>

II 熱流体解析手法

本システムで計算する熱流体解析の概念図を下図に示す。

日射と風の流入条件（温度、風速）を与え、さらに建物や地表面では発熱や日射吸収を考慮しながら熱環境を計算するものである。

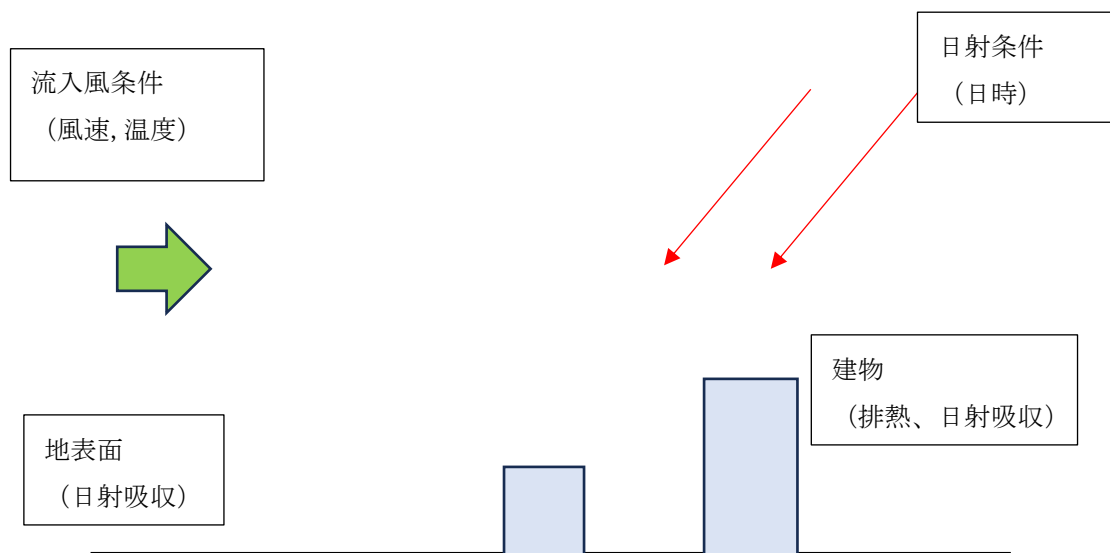


図 1 温熱解析の概要

II-1. 基礎方程式

熱流体解析は下記の運動方程式、連続式、エネルギー式を用いて流れと温度変化の計算を行う。OpenFOAM では、これらの基礎方程式を有限体積法で離散化して計算を行っている。

- ・運動方程式

$$\frac{\partial \rho u}{\partial t} + \nabla(\rho u u) = -\nabla p + \nabla[\mu\{\nabla u + (\nabla u)^T\}] - \nabla\left(\frac{2}{3}\mu\nabla u\right) + \rho g$$

- ・連続式

$$\frac{\partial \rho}{\partial t} + \nabla(\rho u) = 0$$

- ・エネルギー方程式

$$\frac{\partial}{\partial t}(\rho E) + \nabla(\rho E u) = -\nabla(\rho u) + \nabla(k\nabla T)$$

II-2. 採用ソルバー

OpenFOAM には下記に示すような複数の熱流体解析ソルバーが存在する。下記の熱流体解析ソルバーから、日射を考慮できる buoyantSimpleFoam を標準ソルバーとして採用する。

表 1 OpenFOAM の主な熱流体解析ソルバー

ソルバー名	特徴
buoyantBoussinesqSimpleFoam	定常非圧縮性熱流体解析
buoyantBoussinesqPimpleFoam	非定常非圧縮性熱流体解析
buoyantSimpleFoam	定常圧縮性熱流体解析、日射考慮可能
buoyantPimpleFoam	非定常圧縮性熱流体解析、日射考慮可能

II-3. OpenFOAM のインストール方法

OpenFOAM は、Linux (ubuntu) 上で動作するプログラムであり、CUI コマンドの下記手順でインストールを行う。

```
# Add the repository
curl https://dl.openfoam.com/add-debian-repo.sh | sudo bash

# Update the repository information
sudo apt-get update

# Install preferred package. Eg,
sudo apt-get install openfoam2306-default
```

また、コマンドサーチパスや環境変数を設定する下記の設定を ~/.bashrc に記述する。

```
source /usr/lib/openfoam/openfoam2306/etc/bashrc
```

II-4. OpenFOAM の処理手順

OpenFOAM は、STL や OBJ などの三角形パッチの表面形状から 3 次元メッシュファイルを生成し、各種境界条件や計算条件から熱流体計算を実施するものである。

下図に OpenFOAM の主な処理の流れを示す。

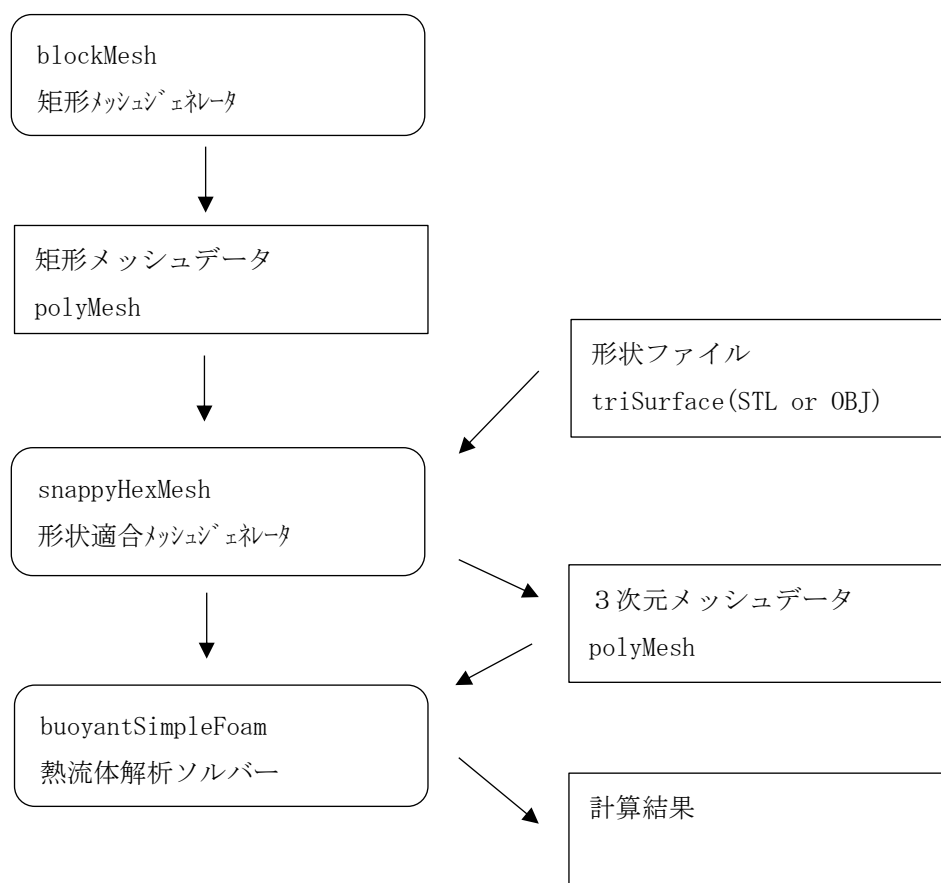
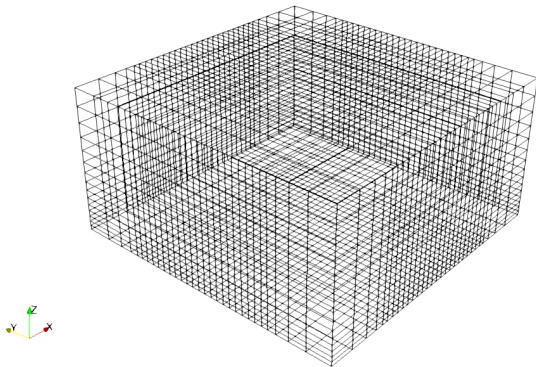
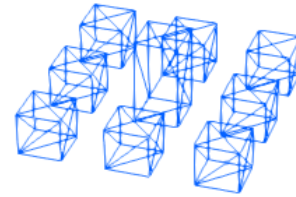


図 2 OpenFOAM の処理の流れ

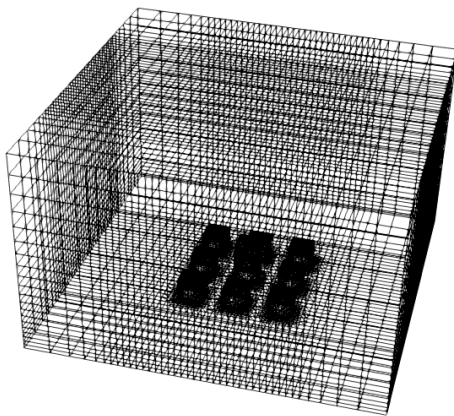
①blockMesh で作成した矩形メッシュ



②STL 形式または OBJ 形式で作成した
建物や地形形状
(CityGML から変換する)



③snappyHexMesh で作成した
計算用メッシュ



建物周辺のメッシュ拡大図

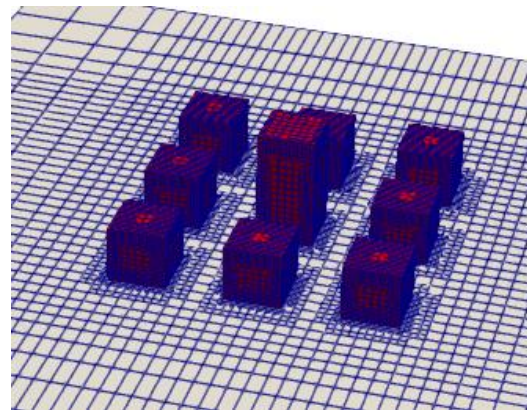


図 3 OpenFOAM のメッシュ作成イメージ

OpenFOAM では面の名前に対して各種境界条件を設定する。本システムでは blockMesh で作成した矩形メッシュの各面に下記のような名前を設定している。

ユーザ指定の流入・流出条件はこれらの面の名前を利用して境界条件を設定している。

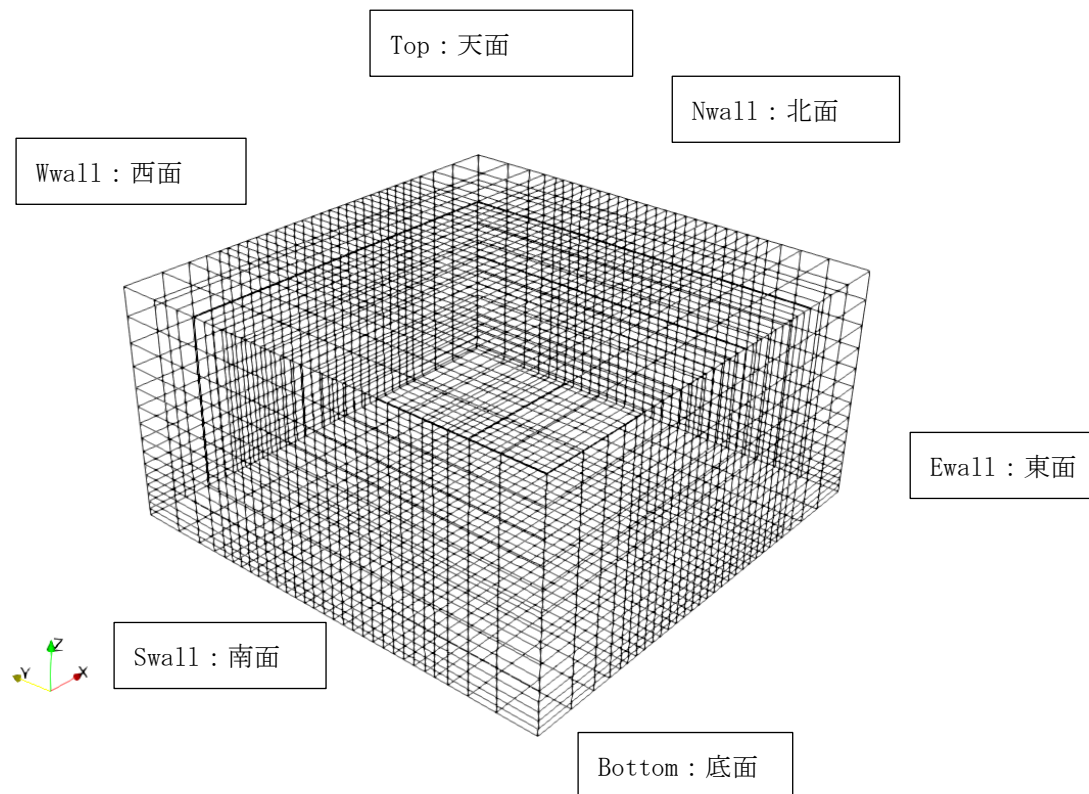


図 4 矩形メッシュの面の命名

III OpenFOAM の入力仕様

III-1. OpenFOAM のフォルダ構成

OpenFOAM の標準的なフォルダ構成を以下に示す。形状ファイルや解析条件を対応フォルダに格納した後に Allrun スクリプトを実行する事により、メッシュ生成から計算までが連続的に実行される。

計算結果は 100 などの収束計算回数の数値名のフォルダに出力される。

Allrun	# 実行スクリプト
Allclean	# 初期化スクリプト
---system	# メッシュ生成条件、解析制御条件の格納フォルダ
+---constant	# 物性値フォルダ
¥---triSurface	# 形状ファイル (STL や OBJ ファイル)
¥---polyMesh	# 計算用メッシュファイル
+---0. orig	# 初期条件および境界条件の格納フォルダ
---0	# 初期条件の格納フォルダ
---100	# 計算結果の格納フォルダ
---nnn	# 計算結果の格納フォルダ (nnn は収束計算回数)

図 5 OpenFOAM のフォルダ構成

IV 入力データ

本章では、メッシュ作成や解析条件などの設定ファイルについて説明する。

OpenFOAM で入力する主なフォルダ構成と解析条件ファイルとを以下に示す。

```
|
|
| ¥---system                                # メッシュ生成およびソルバー制御
|     blockMeshDict                        # 矩形メッシュ作成条件
|     controlDict                          # ソルバー制御データ（計算回数、 $\Delta t$ 、出力制御）
|     fvSchemes                            # 計算スキーム
|     fvSolution                           # 収束パラメータ
|     snappyHexMeshDict                    # 自動メッシュ分割条件
|     surfaceFeatureExtractDict            # 特徴線抽出指定
|
| +---constant                             #物性値フォルダ
|     RadiationProperties                  # 日射条件
|     boudaryRadiationProperties           # 日射吸収率
|
| +---0.orig                               # 初期条件、境界条件フォルダ
|     U                                  # 速度
|     p_rgh                              # 圧力（動圧圧力）
|     p                                  # 圧力（全圧力）
|     T                                  # 温度
|     epsilon                             # 乱流散逸率  $\epsilon$ 
|     k                                  # 乱流エネルギーk
|     nut                                  # 乱流動粘性
|     alphasat                            # 熱拡散率
```

図 6 OpenFOAM の主な解析条件ファイルとフォルダ構成

OpenFOAM の入力ファイルは非常に多くの項目がある。ユーザによるカスタマイズの容易さや、システムの保守性を向上させるために、ユーザ入力部分をインクルードファイル形式で別ファイルとして定義する事とし。

下記の例では、計算領域の大きさと分割数をインクルードファイルとした例である。

```
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       blockMeshDict;
}
scale 1;

// *** Include User Data ***
#include "userBlockMesh"

vertices
(
    ($minx $miny $minz)
    ($maxx $miny $minz)
    ($maxx $maxy $minz)
    ($minx $maxy $minz)
    ($minx $miny $maxz)
    ($maxx $miny $maxz)
    ($maxx $maxy $maxz)
    ($minx $maxy $maxz)
);

blocks
(
    hex (0 1 2 3 4 5 6 7) ($nx $ny $nz) simpleGrading (1 1 3)
)
```

図 7 OpenFOAM の入力ファイル例 (blockMeshDict)

```
minx -100;  
miny -100;  
minz 0.0;  
maxx 100;  
maxy 100;  
maxz 100;  
  
nx 40;  
ny 40;  
nz 20;
```

図 8 インクルードファイル例 (userBlockMesh)

OpenFOAM のすべての入力データを記述する事は困難であるので、以下ではユーザ指定により変更されるインクルードファイルの内容について説明する事とする。

IV-1. system フォルダ (メッシュ生成条件)

(1) userSurfaceFeatureExtract

surfaceFeatureExtractDict ファイルに対するユーザ定義ファイルである。

ユーザ入力形状ファイル名を下記の構造で記述する。

```
building.stl // User Input stl or obj
{
    extractionMethod    extractFromSurface;
    includedAngle        150;

    subsetFeatures
    {
        nonManifoldEdges    no;
        openEdges            yes;
    }
}
```

(2) userBlockMesh

blockMeshDict ファイルに対するユーザ定義ファイルである。

計算対象地域に適応した平面直角座標系で計算領域の西南端と北東端の座標値 (m) と熱流体計算領域の最低標高(m)と最高標高(m)を下記の構造で定義する。最高標高は対象地域の建物よりも十分大きく (2 倍～3 倍程度) する必要がある。さらに、x 方向、y 方向、z 方向の格子分割数を設定する。

座標値は、UI で設定した緯度経度から自動的に変換される。

分割数についても計算領域の範囲から自動的に設定される。

```
minx -100.0; // 南西隅と北東隅の座標値と標高値
miny -100.0;
minz -0.0;
maxx 100.0;
maxy 100.0;
maxz 100.0;

nx 30; // x 方向分割数
ny 30; // y 方向分割数
nz 15; // z 方向分割数
```

(3) userSnappyHexMesh_1

snappyHexMeshDict ファイルに対するユーザ定義ファイルである。

メッシュ細分化レベルとユーザ入力形状ファイル名と面の名前を下記の構造で記述する。

```
mlevel 1;           // Mesh Refine Level 1 - 3

// Add User STL File
geometry
{
    building.stl
    {
        type triSurfaceMesh;
        name building;
    }
}
```

(4) userSnappyHexMesh_2

snappyHexMeshDict ファイルに対するユーザ定義ファイルである。

ユーザ入力形状ファイルの拡張子を除いた面の名前を下記の構造で記述する。

```
features
(
    {
        file "building.eMesh";
        level $mlevel;
    }

);
```

(5) userSnappyHexMesh_3

snappyHexMeshDict ファイルに対するユーザ定義ファイルである。

計算領域の内部点座標値を設定する。

座標値は UI で設定した計算領域の大きさから自動的に設定される。

```
locationInMesh (50.0 50.0 20.0);
```

IV-2. constan フォルダ (物性値)

(1) userRadiationProperties

radiationProperties ファイルに対するユーザ定義ファイルである。

日射条件として、計算する日時と座標値 (緯度、経度) を設定する。

UI で設定した計算領域と計算日時から設定される。

```
localStandardMeridian  +9;    // GMT offset (hours)
startDay                242;    // day of the year
startTime              10;    // time of the day (hours decimal)
longitude              135.243683; // longitude (degrees)
latitude              34.052235; // latitude (degrees)
```

(2) userBoundaryRadiationProperties

boundaryRadiationProperties ファイルに対するユーザ定義ファイルである。

ユーザ入力形状ファイルの拡張子を除いた面の名前と日射吸収率を設定する。

UI で設定したユーザ入力形状ファイルと日射吸収率が設定される。

```
building
{
    type    opaqueDiffusive;
    wallAbsorptionEmissionModel
    {
        type    multiBandAbsorption;
        absorptivity  (0.7 0.7);
        emissivity    (0.7 0.7);
    };
};
```


IV-3. 0.orig フォルダ (初期値、境界条件)

(1) userU_1

u ファイルに対するユーザ定義ファイルである。

ユーザ入力の風向と風速から速度ベクトル成分が設定される。

```
internalField    uniform (2 0 0);    // initial Velocity
```

(2) userU_2

u ファイルに対するユーザ定義ファイルである。

風向きを考慮した流入面に対してユーザ入力の風速値を基に、べき指数分布の風速分布を設定する。さらに流出面の設定を行う。

地表面や建物面に対しては noslip の境界条件を設定する。

```
Wwall                                // inlet wall
{
    type            exprFixedValue;
    value            $internalField;
    U0               -2.0;           // User Input Velocity (NEGATIVE)
    MINZ             0;              // User Input MINZ
    NO               0.27;           // Power CONSTAT Value
    ZO               10.0;           // (m)    CONSTAT Value
    valueExpr        "$U0*pow((pos().z()-$MINZ)/$ZO,$NO)*face()/area()";
}

Ewall                                // outlet wall
{
    type            inletOutlet;
    inletValue       uniform (0 0 0);
    value            $internalField;
}

building                                // building
{
    type            noSlip;
}
```

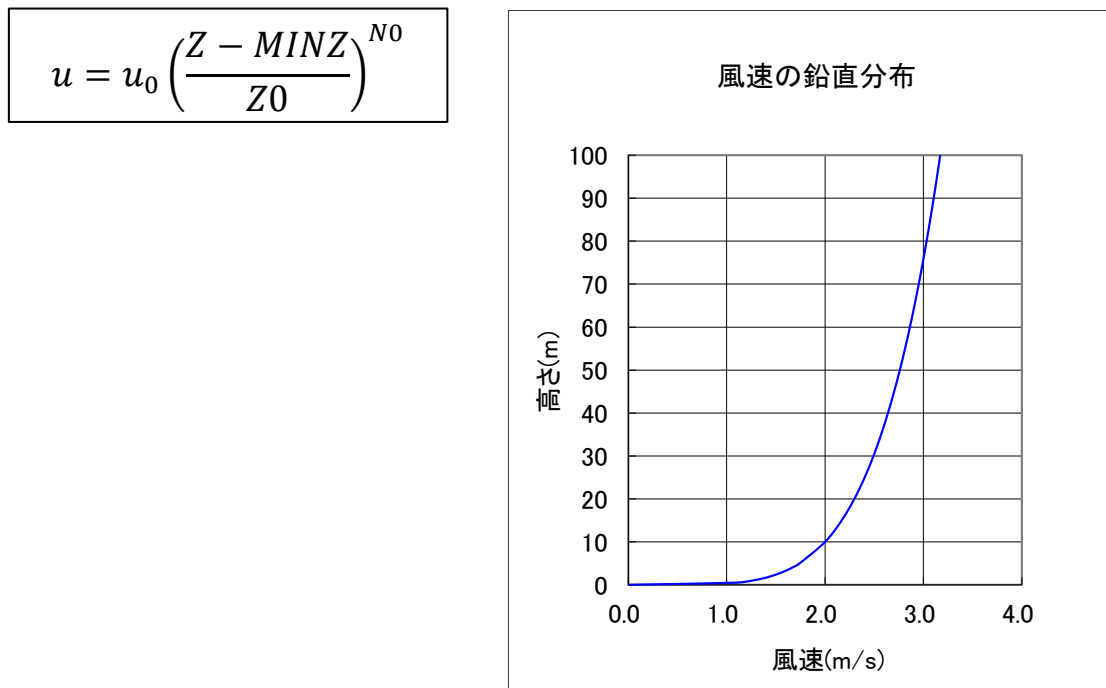


図 9 ベキ指数式による流入速度の定義例

(3) userPrgh

p_rgh ファイルに対するユーザ定義ファイルである。

流出面に対して、大気圧を設定している。

\$internalField にはデフォルト値として大気圧が設定されている。

```
Ewall // outlet wall
{
    type      fixedValue;
    value     $internalField;
}
```

(4) userT_1

T ファイルに対するユーザ定義ファイルである。

ユーザの設定した流入温度を絶対温度の単位 (K) で記述する。

```
internalField    uniform 300;
```

(5) userT_2

T ファイルに対するユーザ定義ファイルである。

ユーザの設定した風向から、流入面と流出面を設定する。また建物や地表面では対応した発熱量(W/m²)を設定する。

```
Wwall            // inlet wall
{
    type          fixedValue;
    value         $internalField;
}

Ewall            // outlet wall
{
    type          inletOutlet;
    inletValue    $internalField;
    value        $internalField;
}

building
{
    type          externalWallHeatFluxTemperature;
    mode          coefficient;
    kappaMethod   fluidThermo;
    h             10;                // W/m2 K
    Ta            $internalField;    // Ref. Temperature
    qr            qr;
    q             100;               // User input Heat Flux W/m2
    value        $internalField;
}
```

(6) userEpsilon

epsilon ファイルに対するユーザ定義ファイルである。

流入面と建物や地表面に乱流散逸条件を設定する。

```
Wwall // inlet wall
{
    type          exprFixedValue;
    value          $internalField;
    U0             2.0;           // User Input Velocity (POSITIVE)
    MINZ           0;             // User Input minZ
    Z0             10.0;          // (m) CONSTAT Value
    N0             0.27;          // Power CONSTAT Value
    ZG             550.0;         // (m) CONSTAT Value
    CM             0.3;           // CONSTAT Value
    valueExpr      "$CM*$U0/$Z0*$N0*pow((pos().z()-$MINZ)/$Z0,$N0-1.0) ¥
                    * pow(0.1*pow((pos().z()-$MINZ)/$ZG,-$N0-0.05) ¥
                    * $U0*pow((pos().z()-$MINZ)/$Z0,$N0) ,2.0)";
}

building // building
{
    type          epsilonWallFunction;
    value          $internalField;
}
```

(7) userK

k ファイルに対するユーザ定義ファイルである。

流入面と建物や地表面に乱流エネルギー条件を設定する。

```
Wwall // inlet wall
{
    type            exprFixedValue;
    value           $internalField;
    U0              2.0;           // User Input Velocity
    MINZ            0;             // User Input minZ
    Z0              10.0;          // (m) CONSTAT Value
    N0              0.27;          // Power CONSTAT Value
    ZG              550.0;         // (m) CONSTAT Value
    valueExpr       "pow( 0.1*pow((pos().z()-$MINZ)/$ZG, -$N0-0.05) ¥
                    * $U0*pow((pos().z()-$MINZ)/$Z0, $N0) , 2.0)";
}

building // building
{
    type            kqRWallFunction;
    value           $internalField;
}
```

(8) userNut

nut ファイルに対するユーザ定義ファイルである。

建物や地表面の乱流粘性条件を設定する。

```
building // building
{
    type            atmNutmWallFunction;
    z0              uniform 0.1;
    value           $internalField;
}
```

(9) userAlphat

alphat ファイルに対するユーザ定義ファイルである。

建物や地表面の熱拡散率の条件を設定する。

```
building
{
    type            compressible::alphatWallFunction;
    Prt             0.85;
    value           $internalField;
}
```

V 出力データ

V-1. 計算結果ファイル

計算結果フォルダには物理量毎に結果が保存されている。

U	# 速度 (m/s)
T	# 温度 (K)
p	# 全圧力 (Pa)
p_rgh	# 動圧力 (Pa)
alphat	# 熱拡散率
epsilon	# 乱流散逸率
k	# 乱流エネルギー
nut	# 乱流粘性
qr	# 日射量 (W/m2)

V-2. ログファイル

Allrun スクリプトで実行される各種プログラムの出力ログがプロジェクトのルートフォルダに作成される。

log.blockMesh	# blockMesh の出力ログ
log.surfaceFeatureExtract	# surfaceFeatureExtract の出力ログ
log.snappyHexMesh	# snappyHexMesh の出力ログ
log.buoyantSimpleFoam	# buoyantSimpleFoam (熱流体解析ソルバー) のログ

VI 熱流体解析ソルバの切り替え手順

VI-1. OpenFOAM 用テンプレートの作成

OpenFOAM の標準的なフォルダ構成を以下に示す。下図のフォルダ構成を tar 形式に圧縮したファイルが、要件定義資料における「熱流体解析ソルバー式圧縮ファイル」となる。

Allrun	# 実行スクリプト
Allclean	# 初期化スクリプト
---system	# メッシュ生成条件、解析制御条件の格納フォルダ
controlDict	# ソルバー制御データ（計算回数、 Δt 、出力制御）
fvSchemes	# 計算スキーム
fvSolution	# 収束パラメータ
+---constant	# 物性値フォルダ
boundaryRadiationProperties	# 日射吸収率条件
radiationProperties	# 日射条件
thermophysicalProperties	# 流体物性
turbulenceProperties	# 乱流条件
g	# 物体力（重力）
¥---triSurface	# 形状ファイル（STL や OBJ ファイル）
+---0.orig	# 初期条件および境界条件の格納フォルダ

図 10 OpenFOAM の入力構成

ユーザが UI で指定する速度や温度などの情報は IV 章で示したようなインクルードファイルで作成されるため、これらのインクルードファイルを読み込むように境界条件ファイルを作成する事により、本システムの UI を利用しながら、異なるソルバーを選択する事が可能となる。

変更可能な代表的なソルバーは下記のソルバーである。

表 2 変更可能な代表的ソルバー

ソルバー名	特徴
buoyantBoussinesqSimpleFoam	定常非圧縮性熱流体解析
buoyantBoussinesqPimpleFoam	非定常非圧縮性熱流体解析
buoyantSimpleFoam	定常圧縮性熱流体解析、日射考慮可能
buoyantPimpleFoam	非定常圧縮性熱流体解析、日射考慮可能

OpenFOAM にはこれ以外にも様々な熱流体解析ソルバーが用意されており、必要なファイルを用意すれば同様に実行可能と考えられる。

ソルバーを変更する以外にも細かい設定を変更する事も可能である。代表的な変更方法を以下に示す。

(1) controlDict

利用するソルバー名や最大収束回数などを設定する事が可能である。

詳細は、OpenFOAM のドキュメントを参照されたい。

application	buoyantSimpleFoam;	# ソルバー名
startFrom	startTime;	
startTime	0;	
stopAt	endTime;	
endTime	1000;	# 最大収束回数
deltaT	1;	# 時間増分
writeControl	timeStep;	
writeInterval	100;	# 出力タイミング
purgeWrite	0;	
writeFormat	ascii;	
writePrecision	6;	
writeCompression	off;	
timeFormat	general;	
timePrecision	6;	

図 11 controlDict の設定例

(2) fvSchemes

差分近似の手法などが変更可能である。詳細は OpenFOAM のドキュメントを参照されたい。

```
ddtSchemes
{
    default      steadyState;
}
gradSchemes
{
    default      Gauss linear;
}
divSchemes
{
    default      none;
    div(phi,U)    bounded Gauss limitedLinear 0.2;
    energy        bounded Gauss limitedLinear 0.2;
    div(phi,K)    $energy;
    div(phi,h)    $energy;
    turbulence    bounded Gauss limitedLinear 0.2;
    div(phi,k)    $turbulence;
    div(phi,epsilon) $turbulence;
    div(phi,omega) $turbulence;
    div(((rho*nuEff)*dev2(T(grad(U))))) Gauss linear;
}
laplacianSchemes
{
    default      Gauss linear uncorrected;
}
interpolationSchemes
{
    default      linear;
}
snGradSchemes
{
    default      uncorrected;
}
```

図 12 fvSchemes の設定例

(3) fvSolution

物理量毎に連立方程式の解き方や緩和係数などを変更可能である。詳細は OpenFOAM のドキュメントを参照されたい。

```
solvers{
  p_rgh {
    solver      GAMG;
    tolerance   1e-7;
    relTol      0.01;
    smoother    DICGaussSeidel;
  }
  "(U|h|k|epsilon|omega)" {
    solver      PBiCGStab;
    preconditioner DILU;
    tolerance   1e-7;
    relTol      0.01;
  }
}
SIMPLE{
  momentumPredictor no;
  nNonOrthogonalCorrectors 0;
  pRefCell      0;
  pRefValue     0;

  residualControl {
    p_rgh      1e-3;
    U          1e-4;
    h          1e-4;
    "(k|epsilon|omega)" 5e-3;
  }
}
relaxationFactors{
  rho      1.0;
  p_rgh    0.7;
  U        0.3;
  h        0.7;
  "(k|epsilon|omega)" 0.3;
}
```

図 13 fvSolution の設定例

VI-2. OpenFOAM の実行スクリプト

OpenFOAM の実行スクリプト (Allrun) の記述例を以下に示す。

システムで計算結果を表示するために、ソルバー実行後にセル中心座標を出力するコマンドを記述する必要がある。

```
#!/bin/sh
cd "${0%/*}" || exit                # Run from this directory
. ${WM_PROJECT_DIR:?}/bin/tools/RunFunctions  # Tutorial run functions
#-----

runApplication surfaceFeatureExtract
runApplication blockMesh
runApplication snappyHexMesh -overwrite
restore0Dir

runApplication $(getApplication)

#-----

# Write Cell Center Coordinate
$(getApplication) -postProcess -func writeCellCentres -latestTime
```

図 14 Allrun スクリプトの設定例

OpenFOAM の初期化スクリプト (Allclean) の記述例を以下に示す。

```
#!/bin/sh
cd "${0%/*}" || exit                # Run from this directory
. ${WM_PROJECT_DIR:?}/bin/tools/CleanFunctions  # Tutorial clean functions
#-----

cleanCase0
rm -rf constant/extendedFeatureEdgeMesh

#-----
```

図 15 Allclean スクリプトの設定例

VI-3. OpenFOAM 用テンプレートのアップロードと利用

作成・変更した OpenFOAM 用テンプレートを、無圧縮の tar ファイルとする。ウェブアプリの熱流体解析ソルバ一覧画面より、tar ファイルをアップロードする。

上記の操作により、シミュレーションモデル編集画面でアップロードしたソルバを指定できるようになる。

以上