

engineTCMFoamの開発(その1)

moritam51@gmail.com

はじめに

- engineFoamはOpenFoamの予混合火炎エンジン用の標準ソルバーであるが、実用エンジンに適用するには、メッシュ生成や移動、吸排気バルブ移動、燃焼モデルなど数々の課題がある。
- メッシュは境界の移動に伴い、初期設定したメッシュが伸縮するメッシュが利用できるが、複雑形状では計算途中の発散や計算精度の悪化する。
- TopoChangerMeshクラスを使って移動境界に合わせたメッシュ生成消滅ができる。(moveEngineTopoChangerMesh2をご参照)
- [moveEngineTopoChangerMesh](#)のプログラムをベースにengineFoamの物理モデルを実装したメッシュ生成・消滅が可能な[engineTCMFoam](#)を開発した。

移動境界がある場合の方程式

境界での流束は流速差で補正

$$\frac{d}{dt} \int_V \phi dV + \oint_S d\mathbf{s} \cdot (\mathbf{u} - \mathbf{u}_b) \phi = \oint_S d\mathbf{s} \cdot \mathbf{q}_\phi + \int_V s(\phi) dV$$

計算領域の体積は境界移動分で変化 $\frac{d}{dt} \int_V dV - \oint_S d\mathbf{s} \cdot \mathbf{u}_b = 0$

$$\oint_S d\mathbf{s} \cdot \mathbf{u}_b = \sum_f \int_{S_f} d\mathbf{s} \cdot \mathbf{u}_b = F_m$$

開発手順

- ① moveEngineTopoMeshフォルダをコピー⇒engineTCMFoamに変更
- ② engineFoam.CをコピーしengineTopoChangerMeshに関する改造（赤字3か所）
- ③ 移動境界による支配方程式の変更は、rhoPimpleDyMFoam.Cを参考にする。（青字6か所）
- ④ ヘッダファイルの追加。rhoPimpleDyMFoamフォルダのヘッダファイルを追加
 correctPhi.H createControls.H readControls.H
- ⑤ Make/files、Make/optionsの修正
- ⑥ wclean
- ⑦ wmake
- ⑧ fvSolutionへのpcorrの追加 rhoPimpleDyMFoamのtutorialを参考にする。

engineTCMFoam.C

```
#include "fvCFD.H"
#include "engineTime.H"
// #include "engineMesh.H"
#include "engineTopoChangerMesh.H"
#include "psiuReactionThermo.H"
#include "turbulentFluidThermoModel.H"
#include "laminarFlameSpeed.H"
#include "ignition.H"
#include "Switch.H"
#include "OFstream.H"
#include "mathematicalConstants.H"
#include "pimpleControl.H"
#include "fvIOoptionList.H"
#include "CorrectPhi.H" // add
//using namespace Foam;

// * * * * * //
// Main program:
int main(int argc, char *argv[])
{
    # include "setRootCase.H"
    # include "createEngineTime.H"
    # include "createEngineTopoChangerMesh.H"
    // #include "createEngineMesh.H"

    pimpleControl pimple(mesh);
    #include "readCombustionProperties.H"
    #include "createFields.H"
    #include "createMRF.H"
    #include "createFvOptions.H"
    #include "createRhoUf.H"
    #include "initContinuityErrs.H"
    #include "readEngineTimeControls.H"
    #include "compressibleCourantNo.H"
    #include "setInitialDeltaT.H"
    #include "startSummary.H"
    #include "createControls.H"
}
```

engineTCMFoam.C

```
// ***** //
```

```
Info<< "¥nStarting time loop¥n" << endl;
```

```
while (runTime.run())
```

```
{
```

```
    #include "readEngineTimeControls.H"
```

```
    #include "setDeltaT.H"
```

```
    #include "readControls.H"
```

```
    // Store divrhoU from the previous mesh so that it can be mapped
```

```
    // and used in correctPhi to ensure the corrected phi has the
```

```
    // same divergence
```

```
    volScalarField divrhoU
```

```
    (
```

```
        "divrhoU",
```

```
        fvc::div(fvc::absolute(phi, rho, U))
```

```
    );
```

```
    runTime++;
```

```
    Info<< "Crank angle = " << runTime.theta() << " CA-deg" << endl;
```

```
        // Store momentum to set rhoUf for introduced faces.
        volVectorField rhoU("rhoU", rho*U);
```

```
        mesh.update();
```

```
    if (mesh.changing() && correctPhi)
```

```
    {
```

```
        // Calculate absolute flux from the mapped surface velocity
        phi = mesh.Sf() & rhoUf;
```

```
        #include "correctPhi.H"
```

```
        // Make the fluxes relative to the mesh-motion
        fvc::makeRelative(phi, rho, U);
```

```
    }
```

Make/options

EXE_INC = ¥

- I\$(LIB_SRC)/OpenFOAM/lnInclude ¥
- I\$(LIB_SRC)/dynamicMesh/lnInclude ¥
- I\$(LIB_SRC)/fvMotionSolver/lnInclude ¥
- I\$(LIB_SRC)/meshTools/lnInclude ¥
- I\$(LIB_SRC)/finiteVolume/lnInclude ¥
- I\$(FOAM_SOLVERS)/combustion/XiFoam ¥
- I\$(LIB_SRC)/fvOptions/lnInclude ¥
- I\$(LIB_SRC)/sampling/lnInclude ¥
- I\$(LIB_SRC)/thermophysicalModels/specie/lnInclude ¥
- I\$(LIB_SRC)/transportModels/compressible/lnInclude ¥
- I\$(LIB_SRC)/thermophysicalModels/basic/lnInclude ¥
- I\$(LIB_SRC)/thermophysicalModels/reactionThermo/lnInclude ¥
- I\$(LIB_SRC)/TurbulenceModels/turbulenceModels/lnInclude ¥
- I\$(LIB_SRC)/TurbulenceModels/compressible/lnInclude ¥
- I\$(LIB_SRC)/thermophysicalModels/laminarFlameSpeed/lnInclude ¥
- I\$(LIB_SRC)/finiteVolume/cfdTools ¥
- I\$(HOME)/OpenFOAM/mori-3.0.1/lib/lnInclude

EXE_LIBS = ¥

- lOpenFOAM ¥
- ldynamicMesh ¥
- lfvMotionSolvers ¥
- lmeshTools ¥
- lfiniteVolume ¥
- lfvOptions ¥
- lsampling ¥
- lturbulenceModels ¥
- lcompressibleTurbulenceModels ¥
- lcompressibleTransportModels ¥
- lfluidThermophysicalModels ¥
- lreactionThermophysicalModels ¥
- lspecie ¥
- llaminarFlameSpeedModels ¥
- \$(FOAM_USER_LIBBIN)/mylibengine.so

例題 (Ubuntu14.04 OpenFOAM ver3.0)

<エンジン諸元>

ボア×ストローク
=102φ×106mm
行程容積=866cc
圧縮比=11.5
(engineCompRatioで確認)

<運転条件>

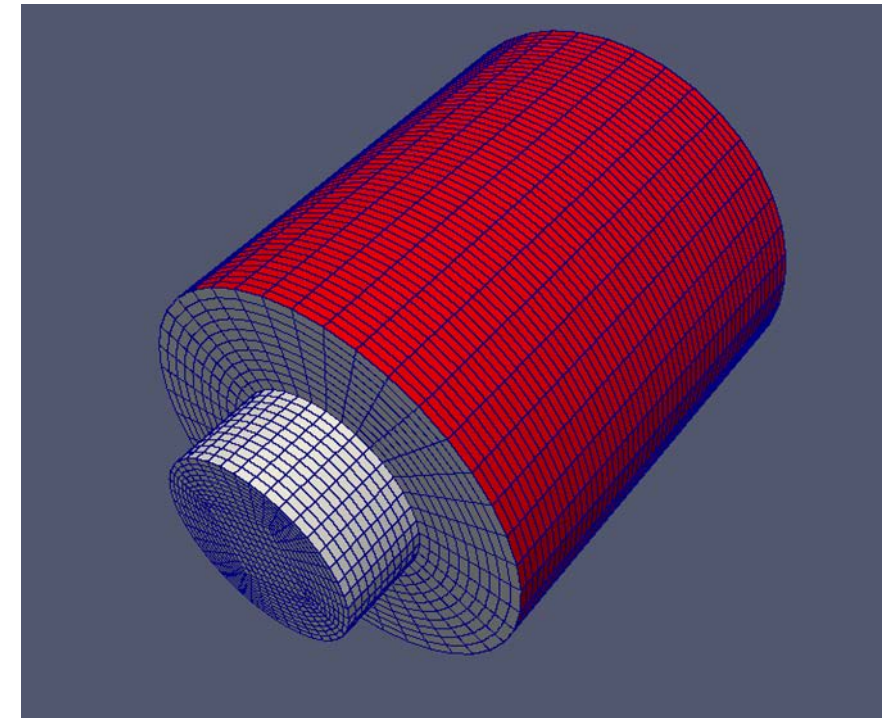
Ne=1200rpm
天然ガス
ストイキ燃焼 (φ=1)

<初期条件>

θ=-180°
P=0.09MPa
T=350K
スワール比2.2(engineSwirlで設定)
k=15.4m²/s²
ε=1.35×10³m²/s³

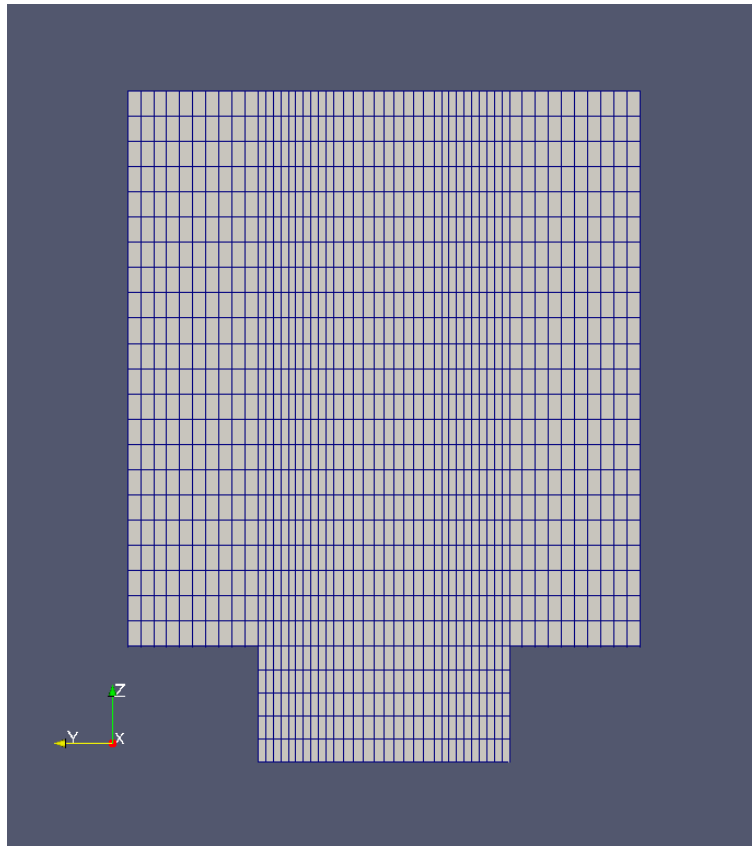
<点火時期>

θ_{ig}=10°BTDC、点火期間7°
点火位置 (0.025 0 0.131)
点火直径2mm 強度4

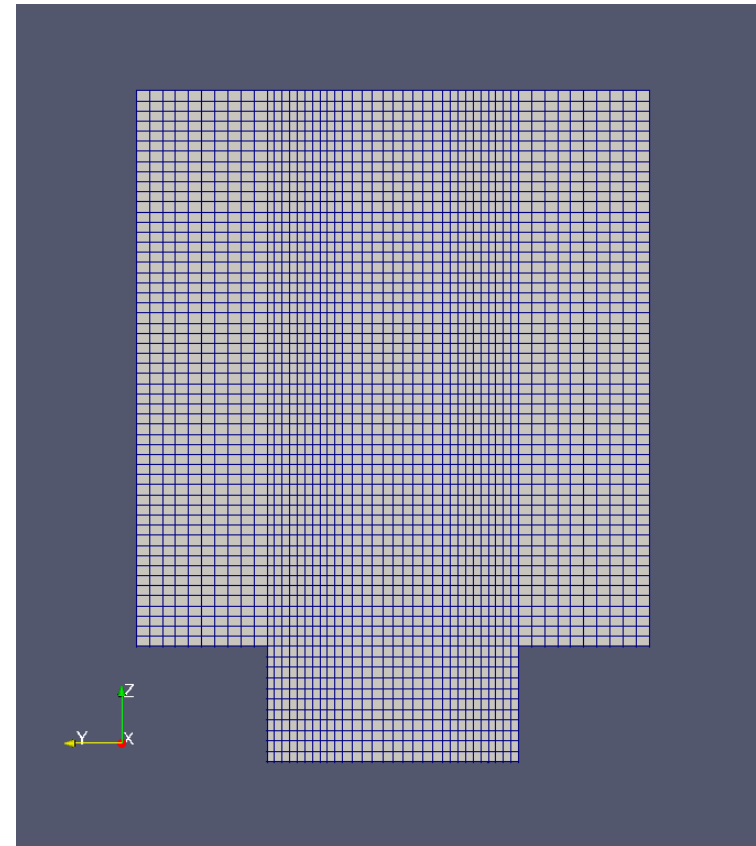


θ=540° 構造メッシュ 55,000Cells
段付きピストン

メッシュ

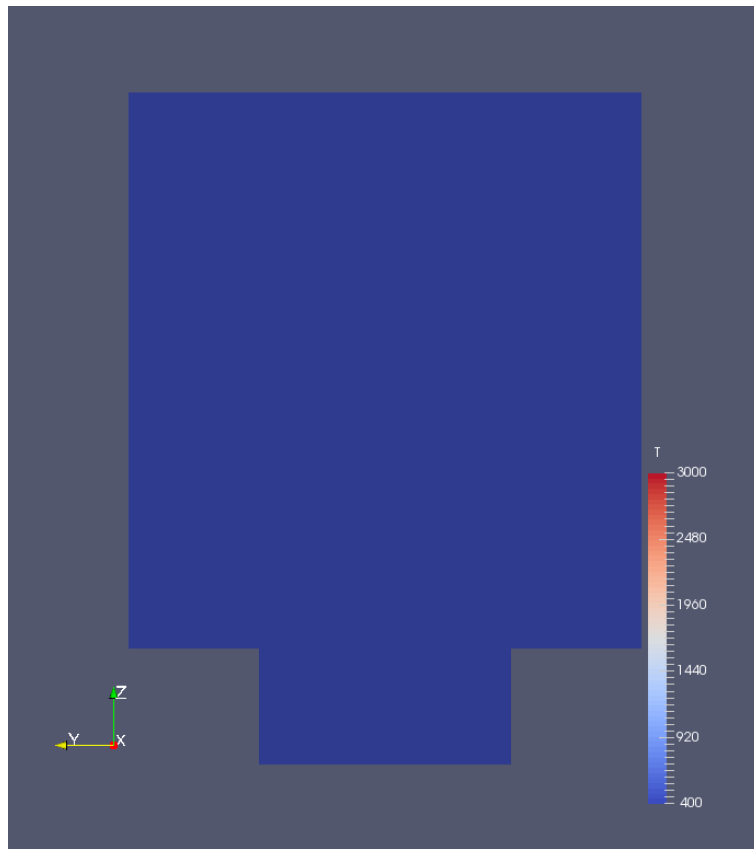


伸縮メッシュ

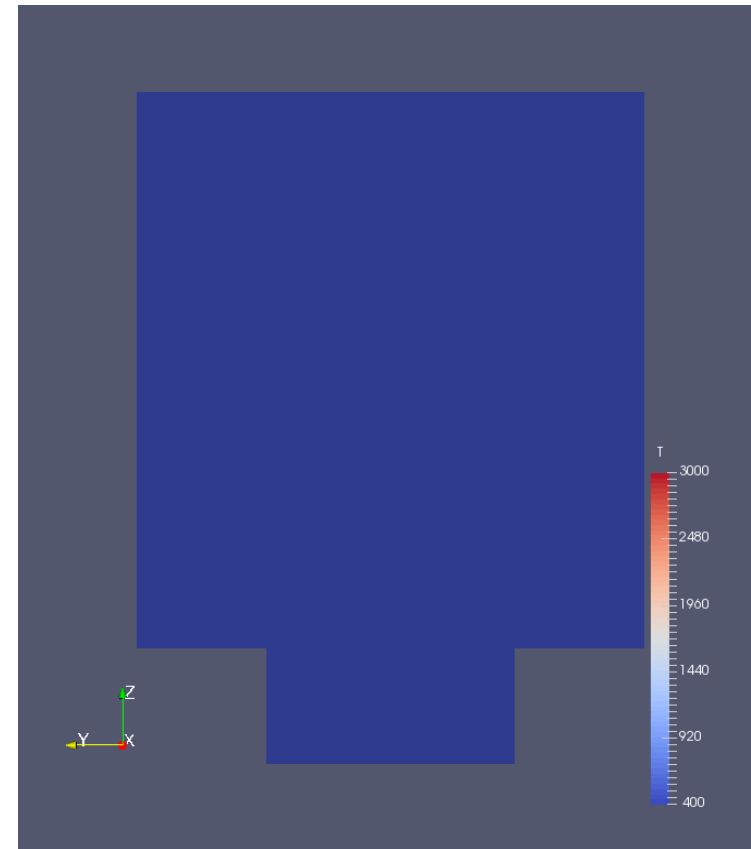


生成消滅メッシュ

温度分布



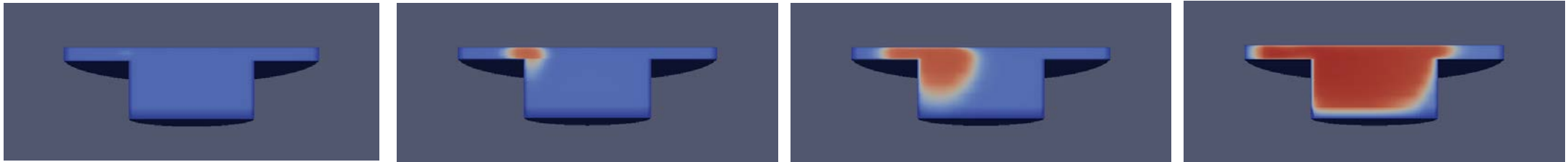
engineFoam



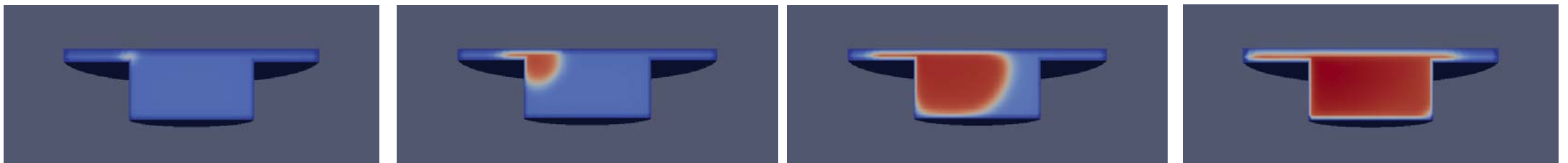
engineTCMFoam2

温度分布

engineFoam(伸縮メッシュ)



engineTCMFoam2(生成消滅メッシュ)



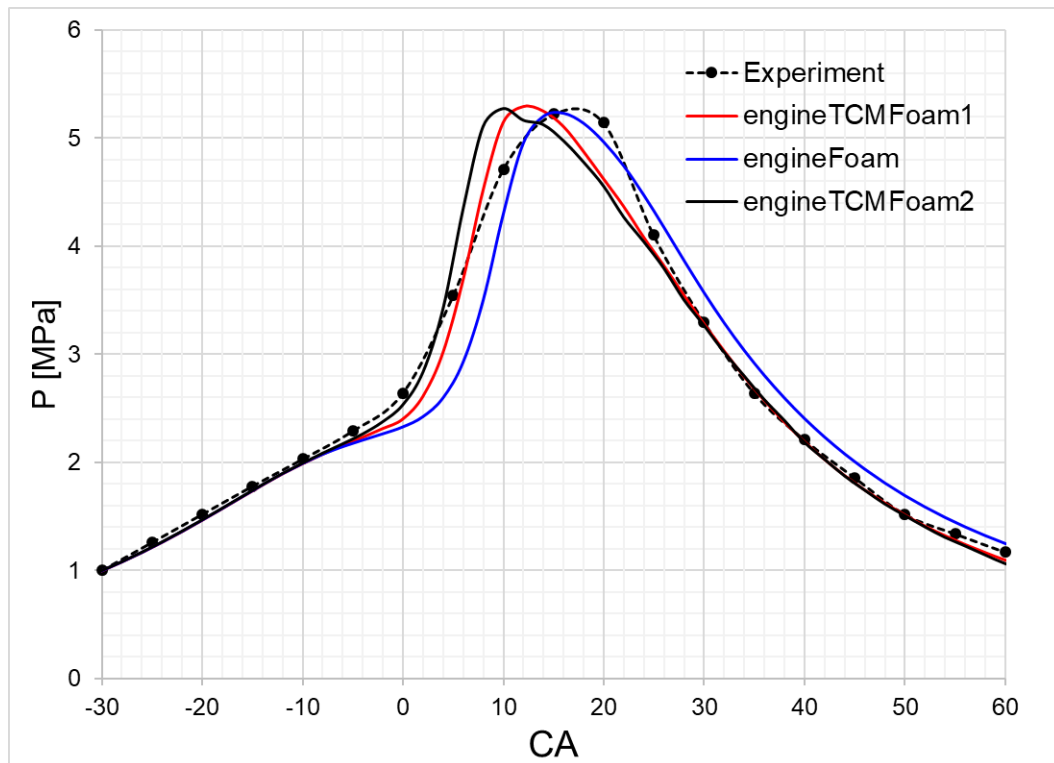
712CA

718CA

724CA

730CA

計算結果



ソルバー	メッシュ生成消滅	移動境界補正	エネルギー式	XC
engineFoam	×	×	エンタルピー	0.78
engineTCMFoam1	○	×	エンタルピー	0.76
engineTCMFoam2	○	○	内部エネルギー	0.68

- 標準 $k\epsilon$ 、b方程式algebraicモデル、都市ガスモデル
- P_{max} が実験値とほぼ同じになるようにXCを決定
- 移動境界補正がないと内部エネルギー式では発散
- エンタルピー式でもXCを選べば、近い結果を再現
- メッシュ生成消滅の効果は、膨張行程の後期で顕著
- engineTCMFoam2は計算が不安定

終わりに

- ・火花点火エンジン燃焼解析用ソルバーとしてメッシュ生成・消滅可能なengineTCMFoamを開発
- ・algebraicモデルを使って段差付ピストン形状（京大エンジン）を題材に実験値との比較を実施
- ・後期燃焼についてモデル改善があったが、メッシュ構造や燃焼モデルパラメータによって計算が不安定になったり、発散しやすいことが課題。

参考文献

- (1) OpenFOAM Workshop11 (2016、ポルトガル) 内燃機関向けのメッシュ作成トレーニング
<https://drive.google.com/folderview?id=0BwfuSMqexhZxN1VXNGxwQ25FQ0k&usp=sharing>
- (2) 「塩路、川那辺、池上、加瀬、乱流予混合燃焼の層流火炎片モデル、日本機械学会論文集 (B編)、65巻631号 (1999-3) 1089-1094」
- (3)
http://web.student.chalmers.se/groups/ofw5/Advanced_Training/DynamicMesh.pdf