

新TreeFoamの紹介

(TreeFoam ver 3.20.231101)

<変更内容>

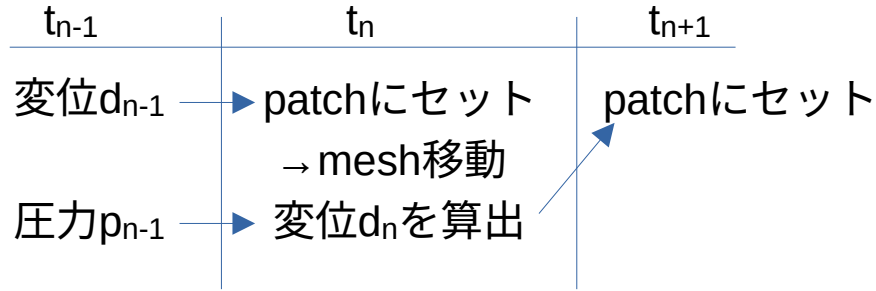
1. 連成計算用codeを変更
2. process間通信を廃止
3. OpenFOAM、FrontISTRの交互計算を追加
同時計算（高速）に加え、交互計算（安定）を追加。

1. 連成計算用codeを変更

<従来>

field「p」とfield「pointDisplacement」の連成するpatchTypeをcodedに変更して、この中に処理内容を記述。時間Loopの途中から処理が始まる。

前回 t_{n-1} に確定した変位 d_{n-1} が、
現時点 t_n のpatchに設定→mesh移動
前回圧力 p_{n-1} を使って変位 d_n を計算し、
次回 t_{n+1} に d_n をセット

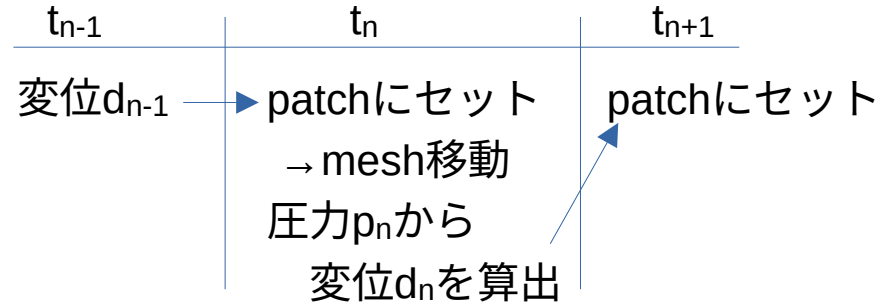


圧力 p_{n-1} から算出した変位が時間 t_{n+1} にセット

<今回>

controlDictのfunctions内にcodedを記述。ここに処理内容を記述。時間Loopの最後に処理が始まる。
(OF-9系は、時間Loopの最初に処理)

時間 t_n の圧力 p_n から算出した変位 d_n が
時間 t_{n+1} のpatchにセットする。



現時点の圧力から算出した変位が次回に設定される
圧力変化が速やかに変位に反映される。

<従来> 「p」 fieldの設定例

```
internalField    uniform 0;

boundaryField
{
    inlet
    {
        type        zeroGradient;
        value       uniform 0;
    }
    outlet
    {
        type        fixed;
        value       uniform 0;
    }
    flap
    {
        type        codedMixed;
        refValue    uniform 0;
        refGradient uniform 0;
        valueFraction uniform 0;
        value       uniform 0;
        name        codedMixed pressCalcDisp;
        code
        #{
            //face中心座標を取得
            // file名を設定
            int pNo = Pstream::myProcNo();
            char procNo[10];
            sprintf(procNo, "%d", pNo);
            int np = Pstream::nProcs();
            char nProcs[10];
            sprintf(nProcs, "%d", np);
            scalar timeValue = this->db().time().value();
            scalar deltaT = this->db().time().deltaTValue();
            std::string tVal = std::to_string(timeValue);
            :
        }
    }
}
```

ptchTypeをcodedに変更

codeを記述

<今回> 「controlDict」 の設定例

```
writeCompression off;
timeFormat        general;
timePrecision     6;
runTimeModifiable true;
adjustTimeStep    no;
//adjustTimeStep  yes;
maxCo             10;
maxDeltaT         0.1;
functions
{
    couplingSource
    {
        #include
        "${FOAM_CASE}/coupling_FrontISTR/python/cont_couplingSource_fsi"
    }
}

couplingToFistr
{
    couplingInterval
    {
        pressure 5;
        temperature 1;
    }
    jobWaitingTime
    {
        maxWaitingTime 6000;
    }
    couplingField
    {
        pressField p;
        pointField pointDisplacement;
    }
    couplingPatch flap;
}
```

処理内容が記述されている

couplingのfield名とpatch名

2. process間通信を廃止

process間通信は、

数値を文字に変換しdataBlock作成、相手のタイミングを確認して送信
dataBlock受信後、数値に変換

の処理が必要で、処理が複雑で、data数が多くなると処理時間が掛かりそう。

↓

process間通信は、全体制御processとOpenFOAMが起動したpythonScript間で通信を行っていたが
全体処理processを廃止し、

OpenFOAMが起動したpythonScriptに、全体制御の処理

FrontISTR起動、変位の計算結果読み取り、patchFaceにmapping
を行う様に修正。→ process間通信を廃止

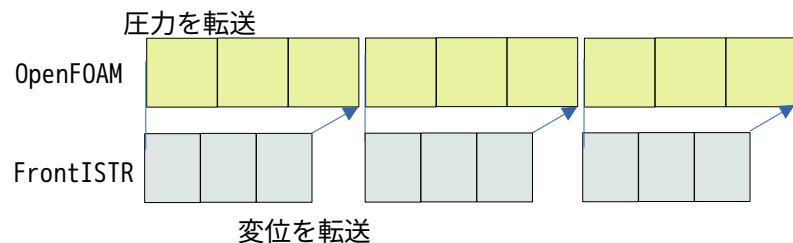
さらに、fileでやり取りしているdata（mappingで使用しているfaceやnodeの座標等）に関しては、
変数を直接binaryで保存し、読込も直接変数に読み込める様に、pickleモジュールを使用
する様に修正。（数値をテキストに変換して保存、テキストを読込み数値に変換の処理が不要）

3. OpenFOAM、FrontISTRの交互計算を追加

<従来>

連成計算は、OpenFOAMとFrontISTRを同時に実行して、各々の待ち時間を減らしている。

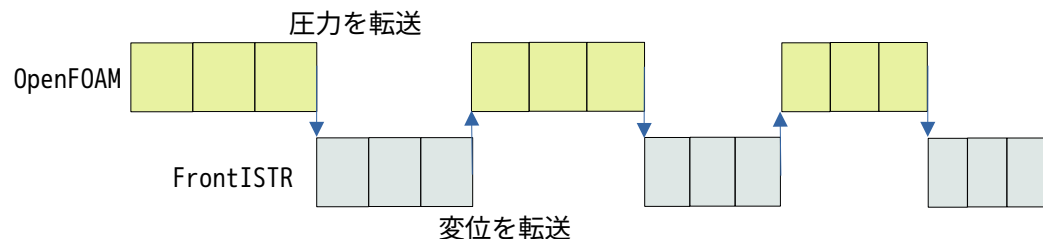
- 計算時間が早くなるが、 Δt は一定値で計算が進む。



<今回>

OpenFOAMとFrontISTRの同時計算（高速）に加え、交互計算を追加。

- maxCo制御で計算が可能。
流速に応じて Δt が変化するので、
計算時間が掛かるが安定する。



流速が変動する様なモデルに関しては、交互計算に設定して、

「adjustTimeStep yes;」で、maxCoを設定して計算させる

方が、安定して計算でき、計算時間も同時計算より早くなる場合がある。

同時計算、交互計算の設定方法

同時計算、交互計算を選択する

同時計算：deltaT：一定

交互計算：controlDict内の

adjustTimeStep: yes に設定し
maxCo制御で計算させる。

process間通信用のbufferSizeを廃止

流体-構造の連成解析

指定したOpenFOAM側のpatch名とFrontISTR側のSGRP名を通じて連成する。
計算は、startTime, latestTimeからも再開できる。

連成解析の内容
基本Dir: ...ingFistr/flap_perp_OF9v2206v2306-FrontISTR/flap_perp_OpenFOAM-FrontISTR/Fluid

連成する項目

fsi: 流体-構造の連成 (流体圧力->構造変位を計算)
 cht: 流体-固体の熱連成 (熱移動を計算) chtss: 固体の熱ひずみを計算

OpenFOAM(流体)とFrontISTR(個体)の計算方法

同時に計算 (高速) 交互に計算 (安定:maxCo制御で計算可能)

OpenFOAMの設定

連成するfield, patchを選択

p fieldの係数: 1.0 p fieldのshift量: 0.0

scalarField: p

pointField: pointDisplacement

patch名
inlet
outlet
flap
upperWall

並列計算する

FrontISTRの設定

連成するsolidDir, SGRPを選択

command 「easyistr」 「fistr1」 のpath設定

easyistr: /home/caeuser/easyIstr 参照...
fistr1: /usr/bin/fistr1 参照...
solidDir: ../Solid 参照...

SGRP名
pressWall
otherS

thread並列する nThreads: 4

連成計算するタイミング (deltaTの倍数:step数を設定)

流体-構造連成計算用のstep数 nStepsFsi: 5
熱移動、熱ひずみ計算用のstep数 nStepsHeat: 1

各Jobの待ち時間

最大待ち時間: 60 sec

結果クリア 設定取得・保存 連成計算開始 Fistrのlog
EasyISTR起動 流体folder開く 固体folder開く 閉じる