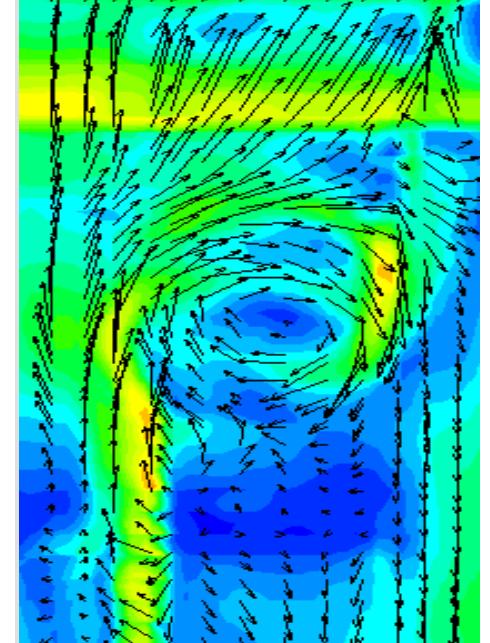


PTM

Consulting Engineers & Architects



OpenCAE Workshop 2012

OpenFOAMと空調シミュレータの連成解析

Coupling simulation between OpenFOAM and a compression heat pump program

小縣 信也(森村設計),

中村 北斗(森村設計), 村田 博道(森村設計),

江口 寛航(早大院), 大野 慶祐(早大基幹理工),

齋藤 潔(早大基幹理工),

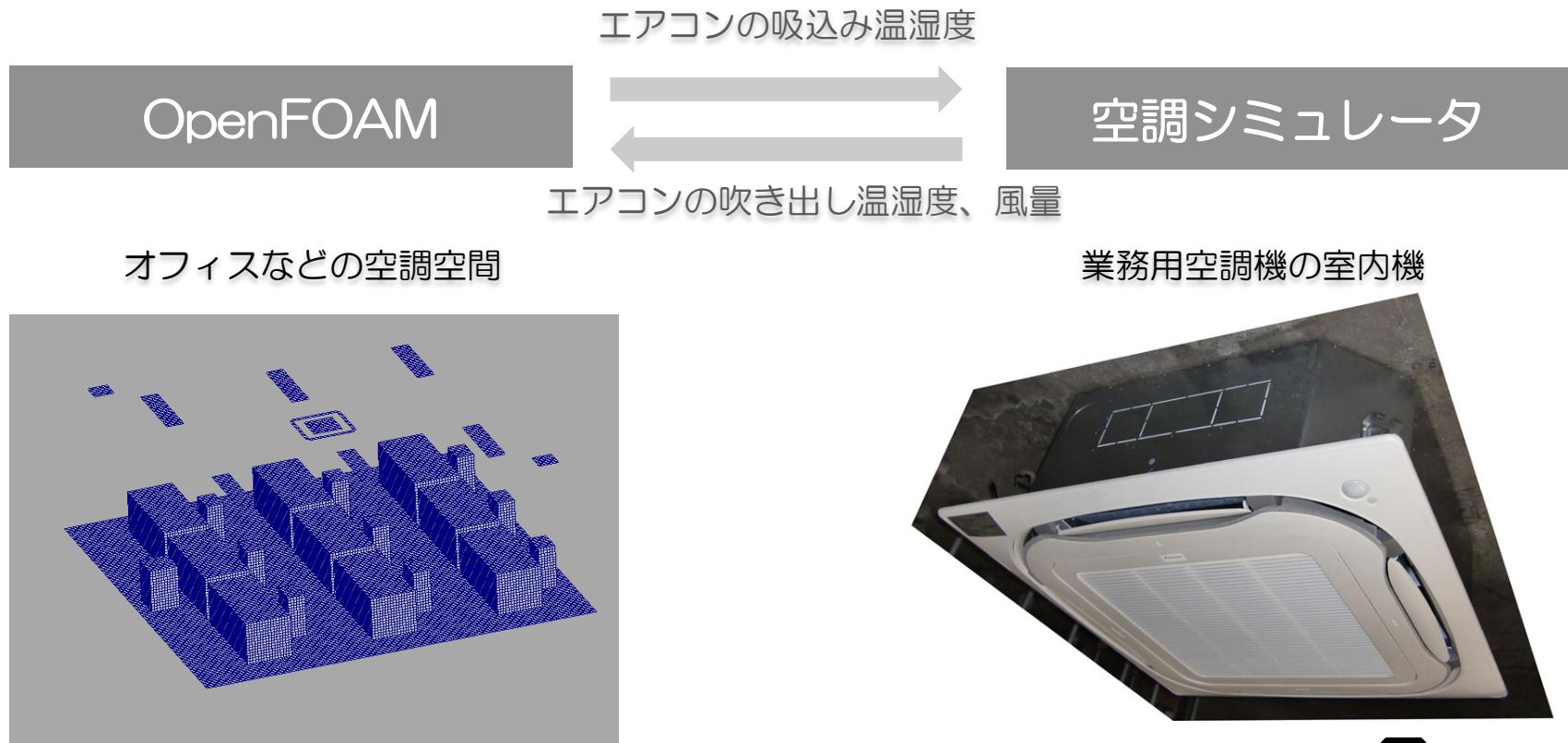
上野 清隆(関西電力), 松本 邦康(関西電力)

株式会社森村設計

P.T.Morimura & Associates, ltd.

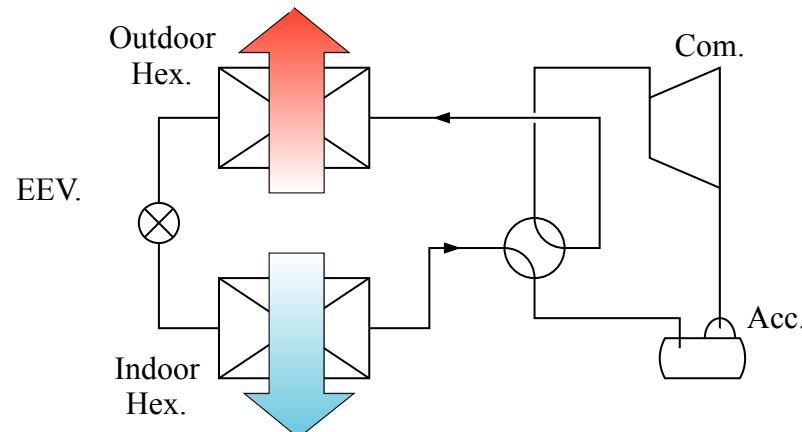
発表概要

- OpenFOAMと外部プログラムを連成させる手法とその手法を用いて行った計算例を紹介する
- 今回連成させたのは、OpenFOAMと空調シミュレータ
- これらを連成させ、空調空間と空調機の相互影響をシミュレーションによって再現した



研究の背景

- 現在、家庭用、業務用空調機として、圧縮式ヒートポンプが広く普及している
- 圧縮式ヒートポンプは、エネルギー消費効率が高く、特に、定格～中間負荷の効率はこれまでに大きく向上した
- 年間を通した運転では、低負荷での運転も多く、今後はこれまであまり検討が進められていなかった、低負荷（断続運転が発生しやすい）での効率向上が求められている



研究の背景

- 低負荷時は、ON・OFFを繰り返す断続運転状態となる

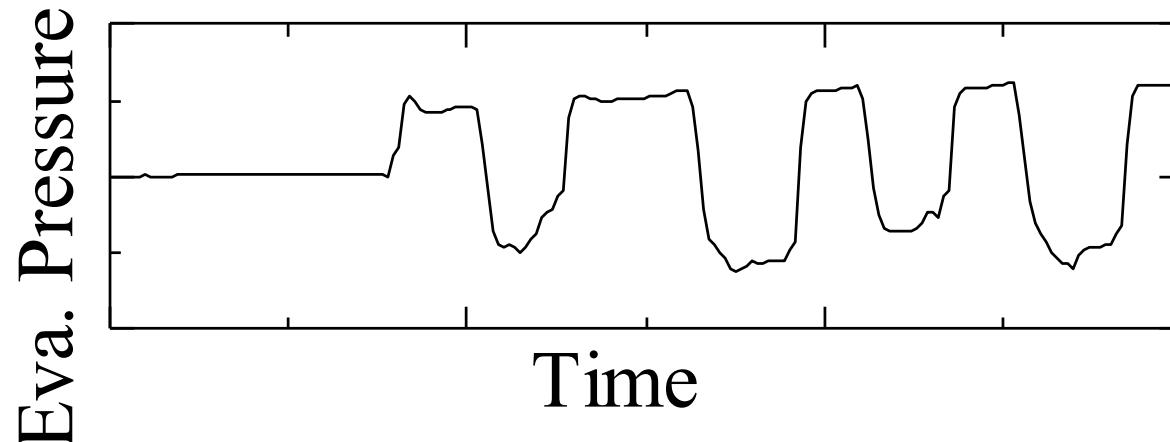


Fig. Intermittent driving

- 断続運転時の制御性能は空調空間の影響を大きく受ける
(例、室内機設置場所、センサー位置、建物躯体性能)

研究の目的

目的

- 圧縮式ヒートポンプと空調空間CFDを連成させ、空調空間を含めた機器の非定常特性を解明すること
(特に断続運転が起こるメカニズム)



最終目標

- ヒートポンプの最適な制御手法の確立

対象とする圧縮式ヒートポンプ

構成要素

- ✓ 圧縮機
- ✓ 凝縮器
- ✓ 蒸発器
- ✓ 膨張弁
- ✓ アキュームレータ
- ✓ 四方弁
- ✓ 接続配管

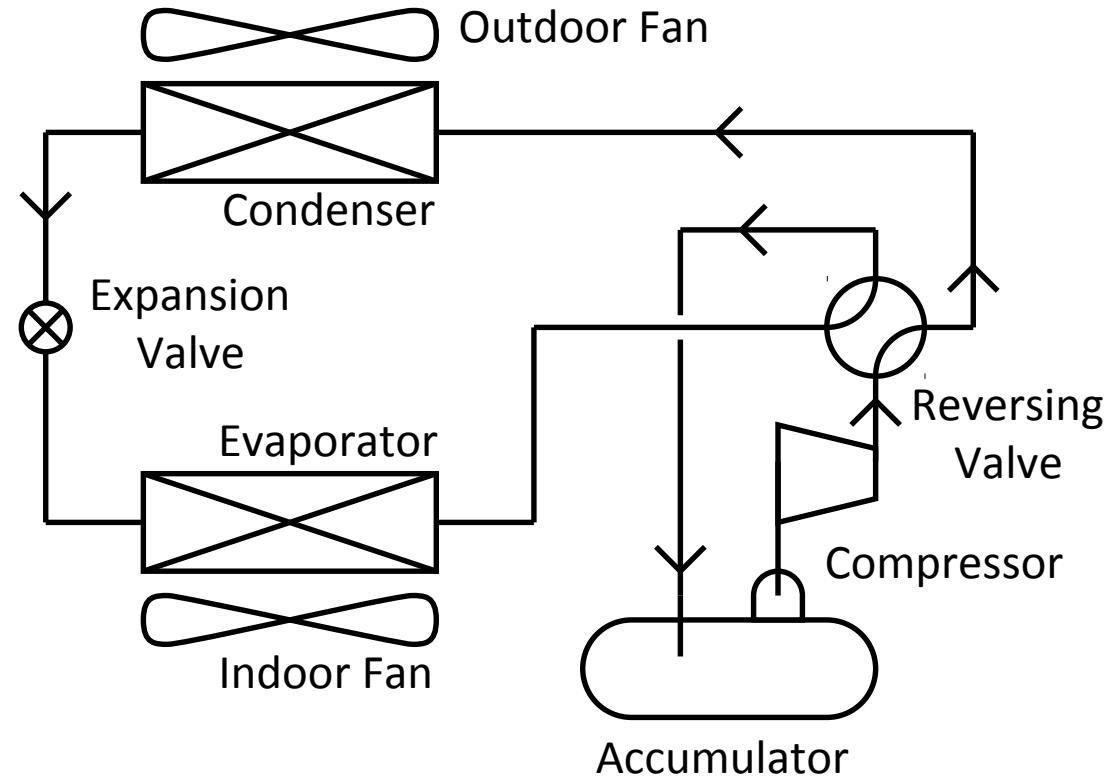


Fig. Schematic flow of
compression type heat pump

空調シミュレータの概要

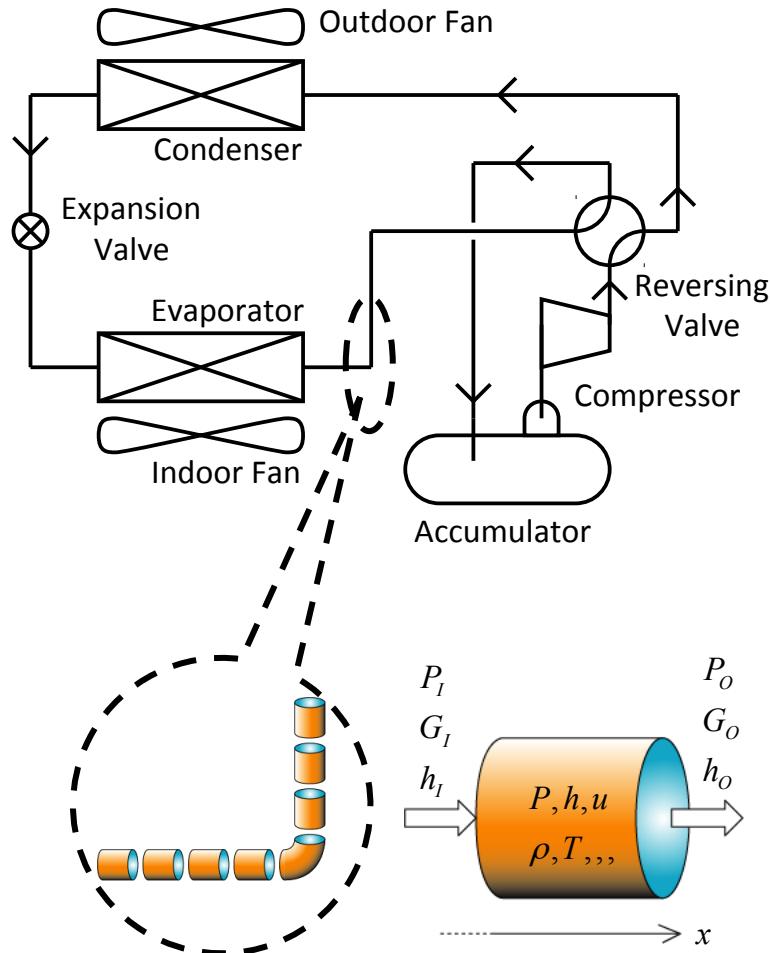


Fig. System flow

冷媒側

連続の式 :

$$\frac{\partial \rho_R}{\partial t} + \frac{\partial (\rho_R v_R)}{\partial z} = 0$$

エネルギー方程式 :

$$\frac{\partial (\rho_R u_R)}{\partial t} S_1 + \frac{\partial (\rho_R v_R h_R)}{\partial z} S_1 = -\pi D_1 q_{in}$$

圧力損失 :

$$\frac{\partial P_R}{\partial z} = 0$$

伝熱管

エネルギー方程式 :

$$\frac{\partial (\rho_M u_M)}{\partial t} S_2 = \pi D_1 q_{in} - \gamma \pi D_2 q_{out}$$

空気側

連続の式 :

$$\rho_{AO} v_{AO} S_{AO} - \rho_{AI} v_{AI} S_{AI} = -J$$

エネルギー方程式 :

$$\rho_{AO} v_{AO} h_{AO} S_{AO} - \rho_{AI} v_{AI} h_{AI} S_{AI} = \gamma \pi D_2 L q_{out} - h_{LIQ} J$$

圧力損失 :

$$P_{AO} = P_{AI}$$

水蒸気バランス :

$$\rho_{AO} v_{AO} X_{AO} S_{AO} - \rho_{AI} v_{AI} X_{AI} S_{AI} = -J$$

伝熱

管内 :

$$q_{in} = K_{in} (T_R - T_M)$$

管外 :

$$q_{out} = K_{out} \left(T_M - \frac{T_{AI} + T_{AO}}{2} \right)$$

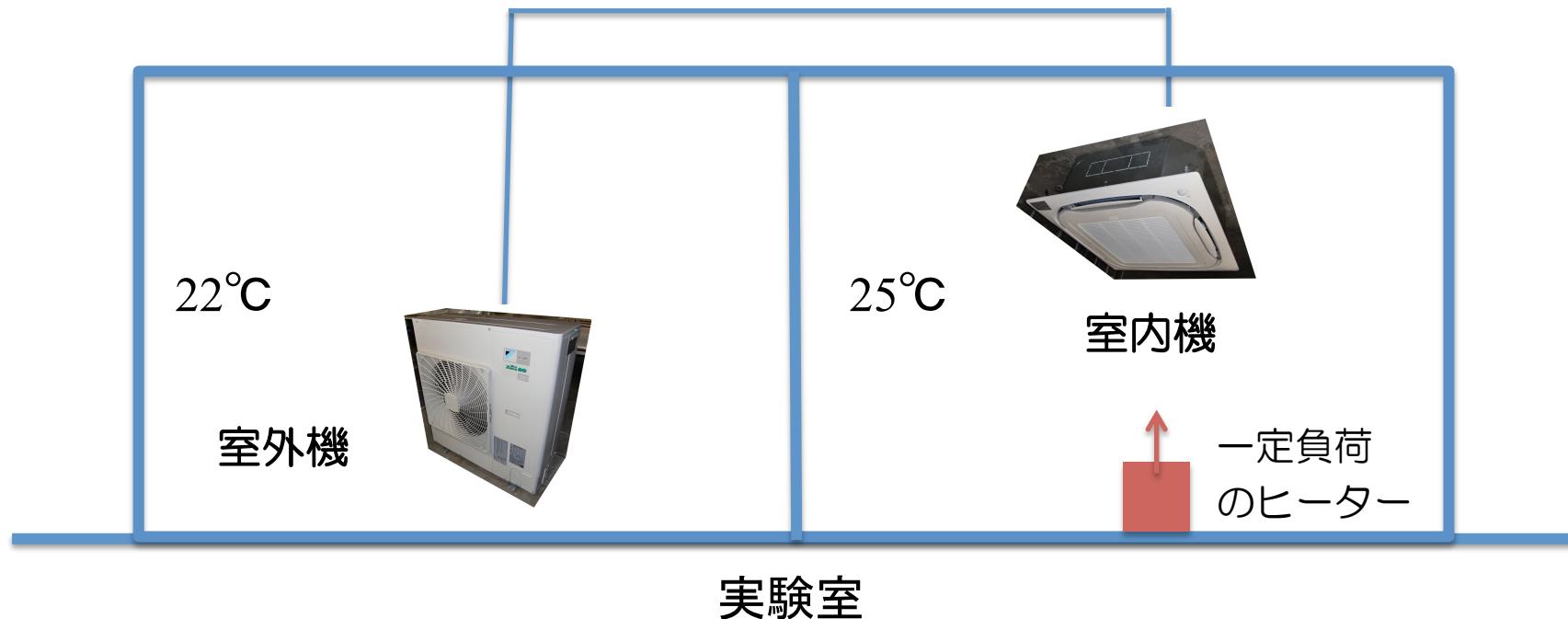
空調シミュレータの計算精度

◆実験概要

空調シミュレータの計算精度を確認するため、恒温恒湿槽室で実験をおこなった

空調機仕様

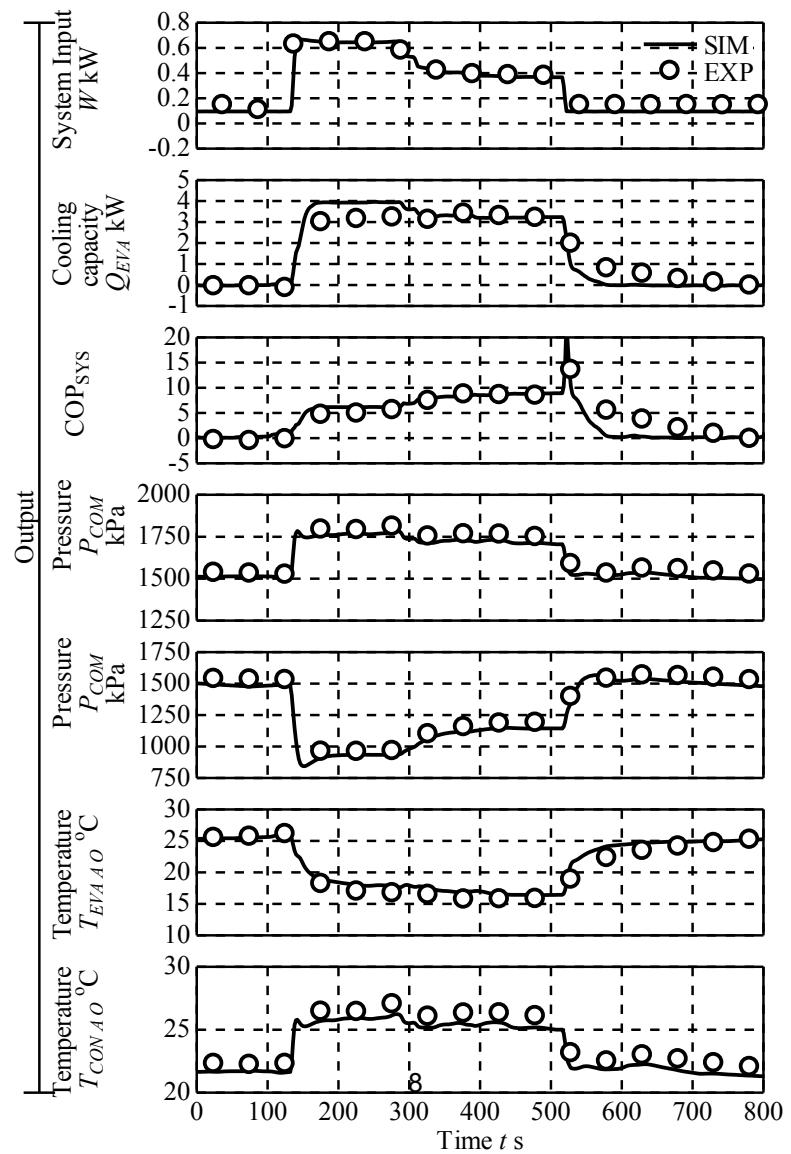
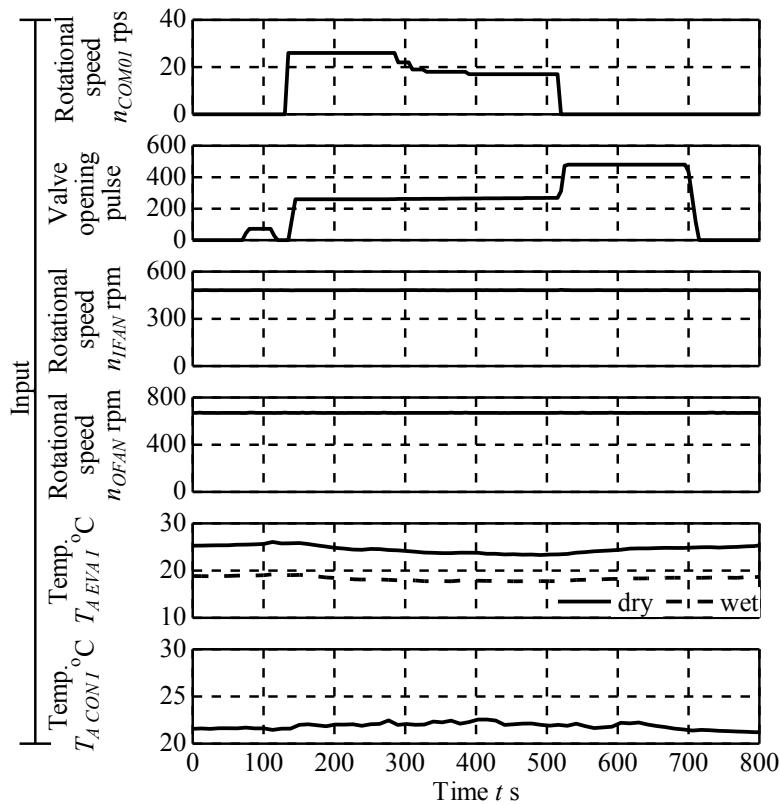
✓定格冷房能力	7.1 kW
✓システム入力	1.81 kW
✓システムCOP	3.92
✓冷媒	R410A



空調シミュレータの計算精度

断続運転解析結果

シミュレーション結果は実験値
と良く対応している



OpenFOAMの概要

- buoyantBoussinesqPimpleFoamに湿度輸送方程式を追加したソルバーを使用

基礎方程式

Continuity Equation

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0$$

Momentum Equation

$$\begin{aligned} \frac{\partial \bar{u}_i}{\partial t} + \frac{\partial}{\partial x_j} (\bar{u}_j \bar{u}_i) = \\ - \frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j} \left(\nu \frac{\partial \bar{u}_i}{\partial x_j} - \bar{u}_i \bar{u}_j \right) - g_i \beta (\bar{\theta} - \theta_{ref}) \end{aligned}$$

Temperature Equation

$$\frac{\partial \bar{\theta}}{\partial t} + \frac{\partial}{\partial x_j} (\bar{u}_j \bar{\theta}) = \frac{\partial}{\partial x_j} (\alpha \frac{\partial \bar{\theta}}{\partial x_j})$$

Humidity Equation

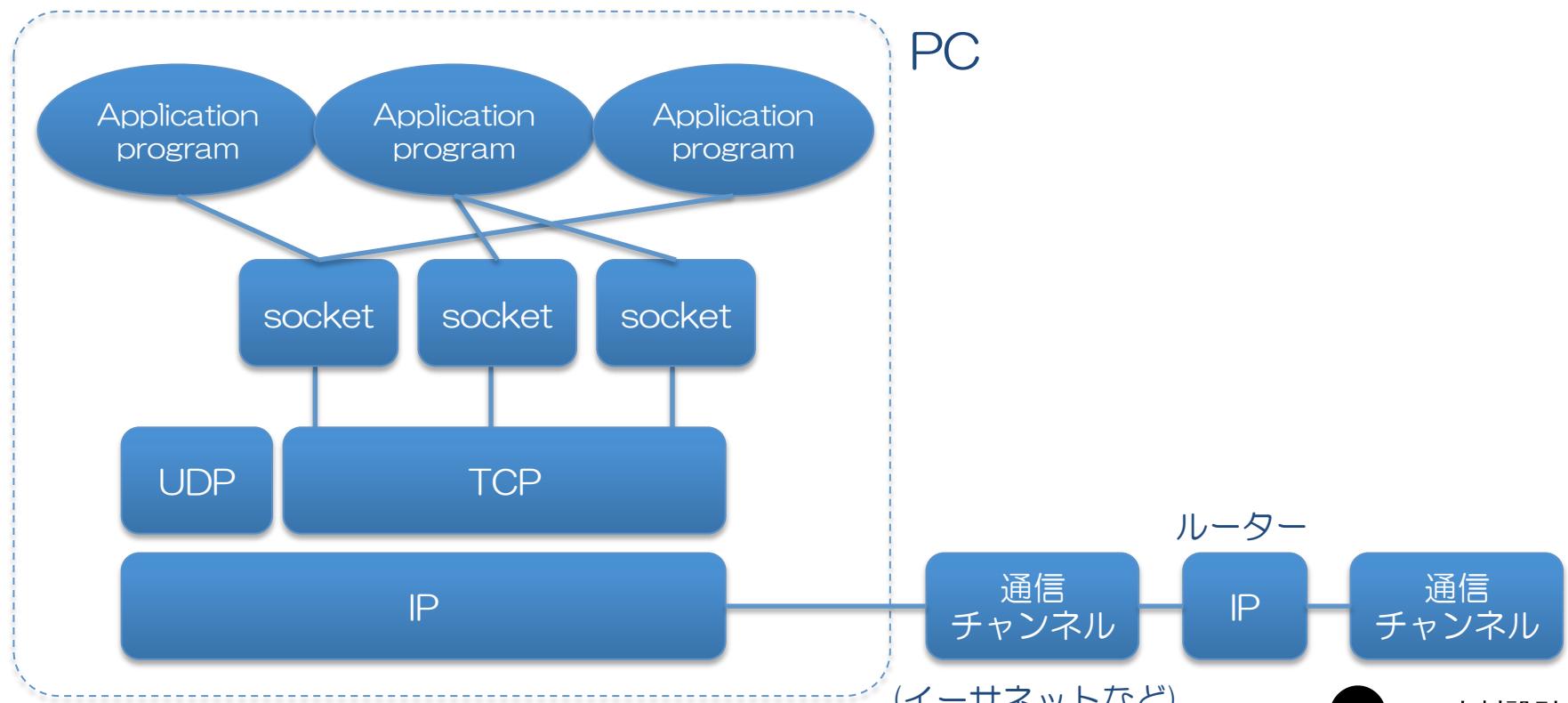
$$\frac{\partial \bar{X}}{\partial t} + \frac{\partial}{\partial x_j} (\bar{u}_j \bar{X}) = \frac{\partial}{\partial x_j} (\alpha_X \frac{\partial \bar{X}}{\partial x_j}) + Spsi$$

Turbulence Model

Standard k-epsilon

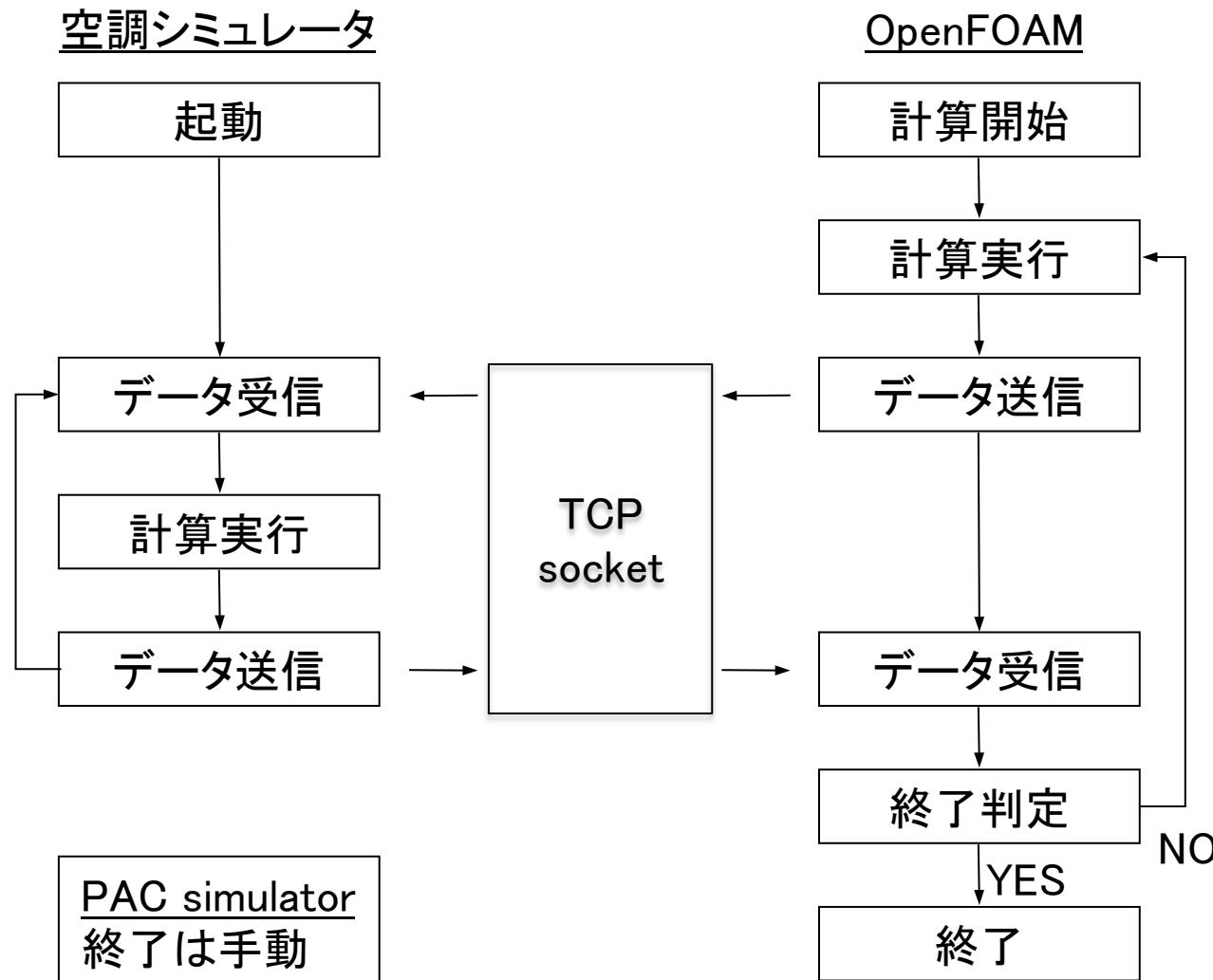
連成手法

- OpenFOAMと空調シミュレータの通信には、TCPソケットを用いている
- TCPとはネットワーク通信で用いられる標準的なプロトコル
- TCPソケットを用いると、アプリケーション間でデータの受け渡しができる



連成時の計算実行手順

計算フロー



連成手法

連成に必要な改良

空調シミュレータ

既存プログラムに通信部分の追加

OpenFOAM

通信ライブラリを作成し、
wmake libsoでコンパイル

controlDictに通信条件を記入

連成手法

TCPサーバー

```
#include <sys/socket.h>
```

socket()を実行してTCPソケットを作成する

bind()を実行してソケットにポート番号を割り当てる

listen()を実行し、クライアントからの接続要求を待機

send()とrecv()を実行して通信を行う

TCPクライアント

```
#include <sys/socket.h>
```

socket()を実行してTCPソケットを作成する

connect()を実行してサーバーへの接続を確立する

send()とrecv()を実行して通信を行う



空調シミュレータ



OpenFOAM

通信ライブラリ

TCPサーバー側(空調シミュレータ) 通信部分の構成

```
#include <sys/socket.h>
int main(){
    ...
    receiveValue(clientSocket, controlT); //センサー検出端温度
    receiveValue(clientSocket, outletT); //PAC吸い込み温度
    receiveValue(clientSocket, outletX); //PAC吸い込み絶対湿度
    ...
    ...
    sendValue(clientSocket, inletT); //PAC吹き出し温度
    sendValue(clientSocket, inletX); //PAC吹き出し絶対湿度
    sendValue(clientSocket, inlet_flow); //PAC吹き出し流量
    sendValue(clientSocket, outlet_flow); //PAC吸い込み流量
    ...
}
```

通信ライブラリ

TCPクライアント側(OpenFOAM) 通信部分の構成1/2

```
#include <sys/socket.h>
bool Foam::cbcFunctionObject::execute()
{
    ...
    sendValue(controlT);
    sendValue(outletT);
    sendValue(outletX);

    // 温度
    const scalar inletT = static_cast<scalar>(receiveValue());
    forAll(inletPatchIDs_, patchI)
    {
        T.boundaryField()[inletPatchIDs_[patchI]] == inletT;
    }

    // 絶対湿度
    const scalar inletX = static_cast<scalar>(receiveValue());
    forAll(inletPatchIDs_, patchI) {
        X.boundaryField()[inletPatchIDs_[patchI]] == inletX;
    }
    ...
}
```

通信ライブラリ

TCPクライアント側(OpenFOAM) 通信部分の構成2/2

```
//風量
const double inlet_flow = receiveValue(); //吹き出し風量
const double outlet_flow = receiveValue(); //吸い込み風量

...
const double theta = M_PI / 4;
const double A_inlet = 0.024 * cos(theta);
const double A_outlet = 0.25;

...
inletU0[0] = inlet_flow_part / A_inlet * sin(theta);
inletU0[1] = 0;
inletU0[2] = - inlet_flow_part / A_inlet * cos(theta);

...
U.boundaryField()[inletPatchIDs_[0]] == inletU2;
U.boundaryField()[inletPatchIDs_[1]] == inletU3;
U.boundaryField()[inletPatchIDs_[2]] == inletU1;
U.boundaryField()[inletPatchIDs_[3]] == inletU0;

...
Ü.boundaryField()[outletPatchIDs_[patchI]] == outletU;
}
```

OpenFOAMの通信部分は、libraryとして作成し、
wmake libsoでコンパイルする

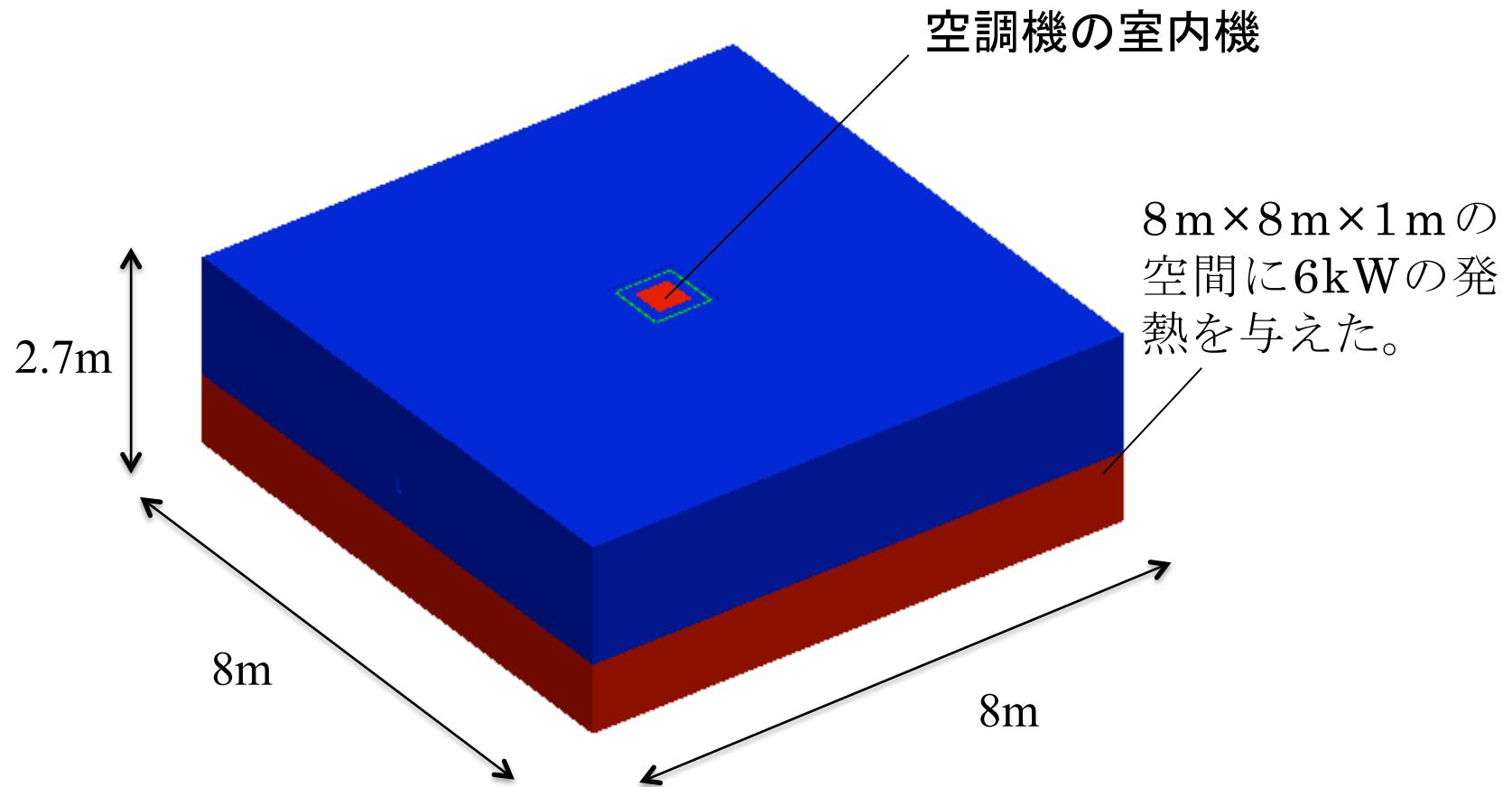
通信ライブラリ

controlDictの記述

```
//controlDict
functions
(
    cbcclnstance1
    {
        type cbccl;
        functionObjectLibs ("libcbccl_office.so");
        connectTo localhost:1234; //TCPポート番号
        controlPatchName AirCon_RA; //制御センサー検出端にする境界名
        outletPatchNames AirCon_RA; //PAC吸い込み境界名
        inletPatchNames (AirCon_SA1 AirCon_SA2 AirCon_SA3 AirCon_SA4);
            //PAC吹き出し境界名
    }
);
```

計算例

OpenFOAM計算モデル



計算例

室内機モデル

- ラウンドフロータイプ
- 吸込口1箇所に対して吹出口4箇所
- 吸込口、吹出口において風速分布は一様



Table. Indoor-Unit model

Inlet	500mm × 500mm
Outlet	30mm × 800mm × 4
Blow off angle	45°

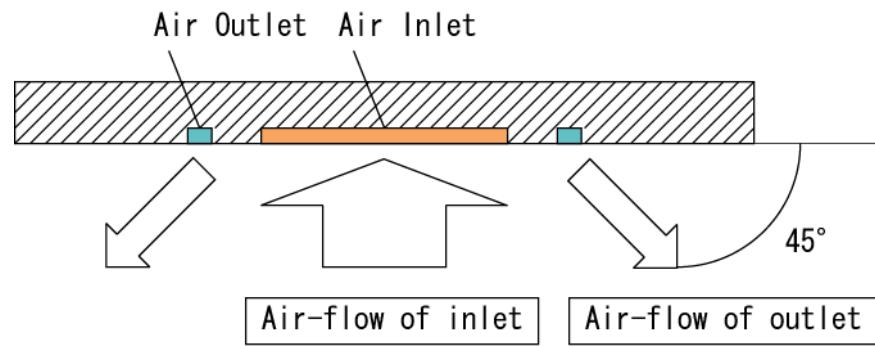
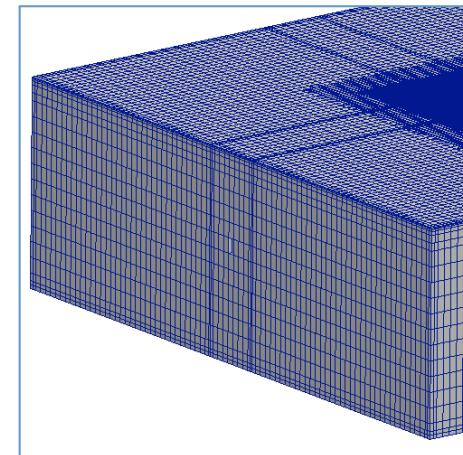
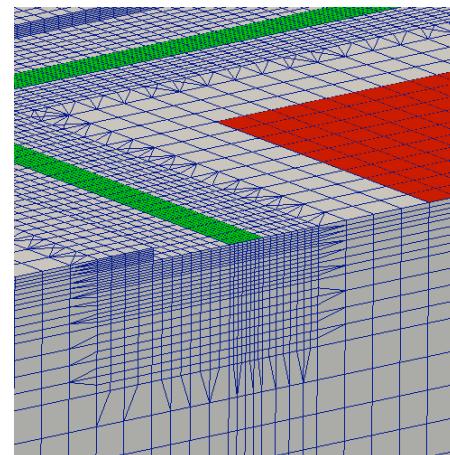
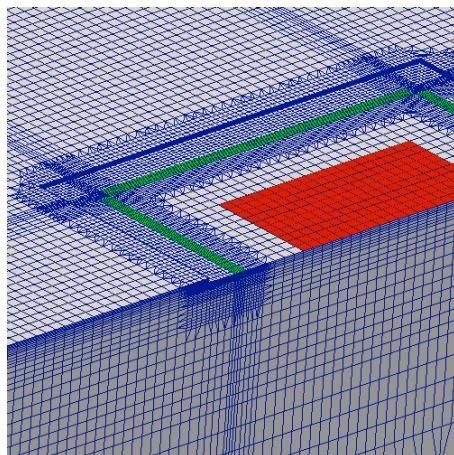
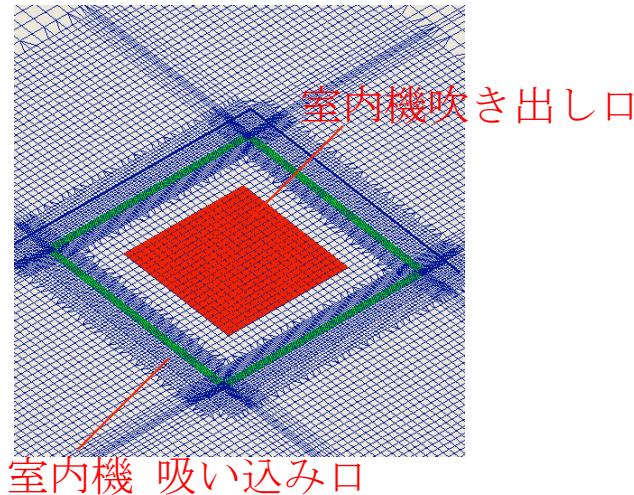
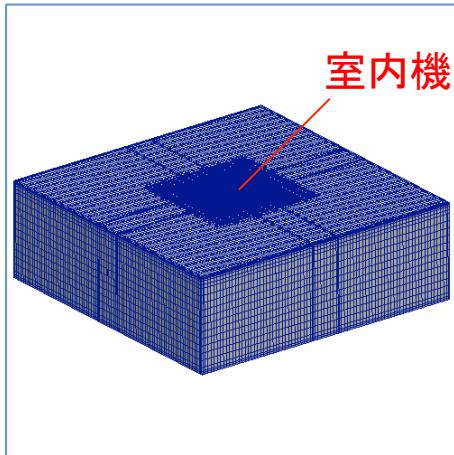


Table. Specification

Appellation	Specification
Indoor unit	Rated blast volume: 21.5m ³ /min Fan: turbofan × 1
Outdoor unit	Compressor: Closed swing type × 1 Rated blast volume: 59m ³ /min Fan: Propeller fan
Refrigerator of system	Rated cooling output: 7.1kW Cooling electric input: 1.81kW Rated heating output: 8.0kW Heating electric input: 1.60kW Refrigerant: R410A

計算例

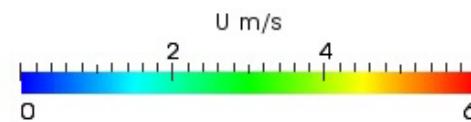
OpenFOAM計算格子



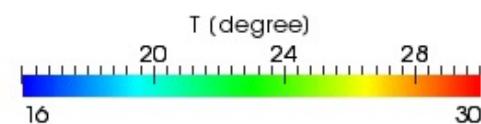
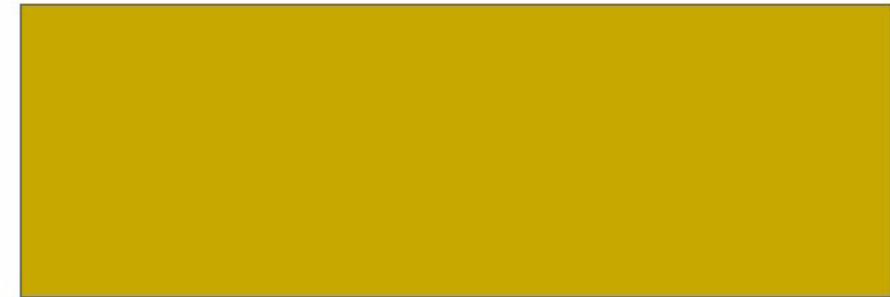
計算例

計算結果(動画)

諸条件
冷房運転
初期室内温度 : 27°C
空調目標温度 : 27°C
内部発熱 : 6kW



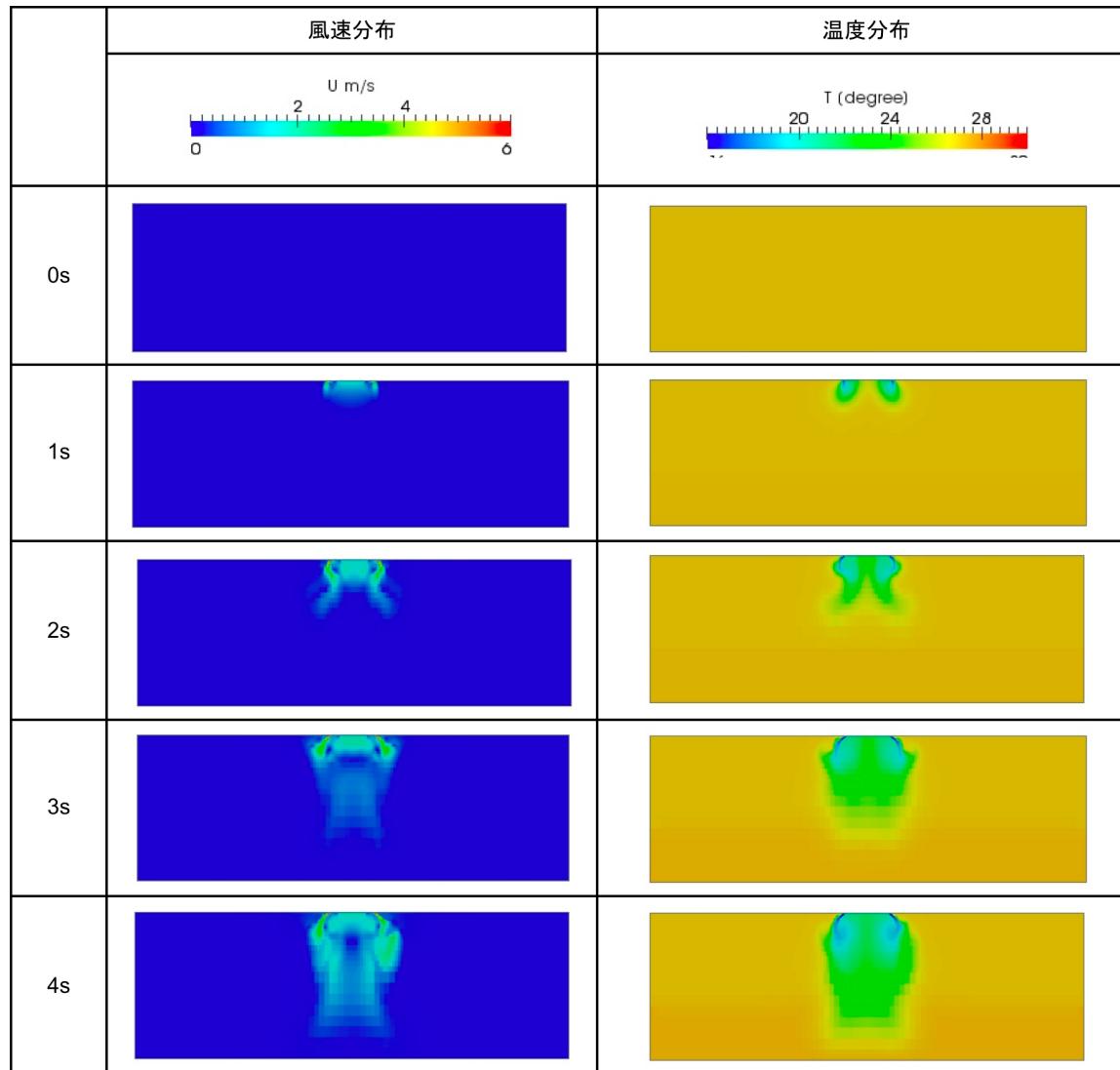
風速分布



温度分布

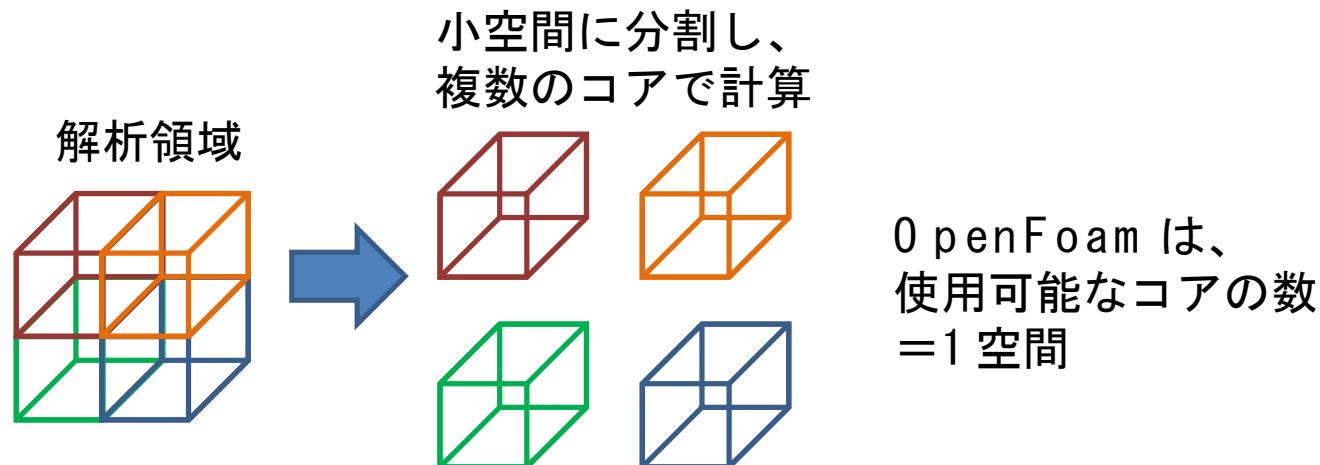
計算例

計算結果



通信ライブラリの並列化

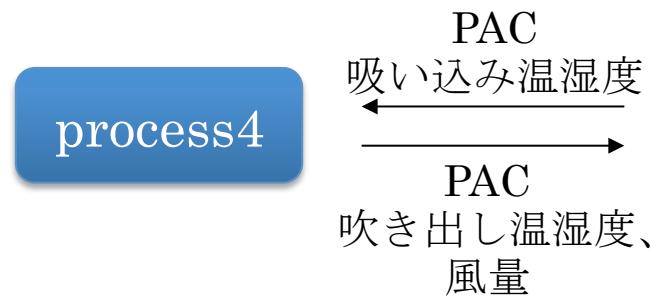
- 空調シミュレータの計算時間は、約120秒/step
- OpenFOAMの計算時間は、約1600秒/step
- 通信ライブラリの開発当初、連成時においてOpenFOAM側の並列計算ができず、OpenFOAM側がボトルネックになっていた
- OpenFOAM側を並列計算に対応できるように改良



通信ライブラリの並列化

- OpenFOAMの並列計算を行うprocessのうち、マスタープロセスだけに通信を担わせるように改良
- 計算時間の大幅な短縮を実現

空調シミュレータ



OpenFOAM

CPU4コアを用いた並列計算時

まとめと今後の展望

まとめ

- TCPソケットを用いて、OpenFOAMと外部プログラムを連成させる方法とそれを用いた計算例を紹介した
- OpenFOAMの並列計算に対応できた

今後の展望

- 軀体蓄熱を考慮させる
→ 固体・流体の連成
- 連成モデルの精度検証
- 制御系の最適な設計について検討を行う

謝辞

今回の通信ライブラリ開発において、以下の方々に多大なるご協力を賜りました。この場をお借りしてお礼申し上げます。

《通信ライブラリの開発》

新潟大学 大嶋助教

東京大学 今野助教

《通信ライブラリの並列化》

株式会社 計算力学研究センター 金田 誠

ご清聴ありがとうございました。