

環境構築方法

2021.09.30 稲葉

対象：

オープン CAE シンポジウム 2021 トレーニング Room C / 10:30-12:00

『Python による OpenFOAM のハンドリング (題材： 形状や計算条件のベイズ最適化)』

はじめに

本セッションでは、Python で OpenFOAM を取り扱うために『OpenFOAM が実行可能な環境で Python を実行する』必要があります。ご受講の前に、本資料の『A』『B』『C』のいずれかに従い環境を用意した後、D の動作確認をお願いいたします。導入には時間がかかる可能性があるため、講義直前に準備を始めた場合に講習に間に合わなくなってしまう可能性がありますのでお気を付けください。

目次

- A. Docker を使用する場合 (予想時間：数分～数十分)
- B. VirtualBox を使用する場合 (予想時間：数分～数時間)
- C. ご自身の環境を使用する場合 (予想時間：不明)
- D. 動作確認方法

A. Docker を使用する場合

1. インターネットに接続された状態で以下を実行します。(時間がかかります)

```
docker pull inabower/opencae-training2021-ofpy
```

2. 作業時には作業したいディレクトリにて以下を実行します。

```
docker run --rm -v ${PWD}:/home/work -p 1111:8888 inabower/opencae-training2021-ofpy
```

2. Chrome や FireFox などの WEB ブラウザの URL バーに『localhost:1111/lab』を入力し接続します。
(『1111』は任意に変更可能です)

→ D. 動作確認へ

B. VirtualBox を使用する場合

1. 『<http://dexcs.gifu-nct.ac.jp/download/>』から "DEXCS Advance" のインストールイメージ "DEXCS_Advance.iso" をダウンロードします
2. 『http://www.opencae.or.jp/wp-content/uploads/2018/06/Install_Dexcs2015_on_Windows10.pdf』を参考に、 "DEXCS_Advance.iso" をインストールし起動します。
3. スタートメニューから "OF-v2006Terminal" を実行します

※以降 Python や Jupyter を実行する際はこのターミナルを用いて実行します。

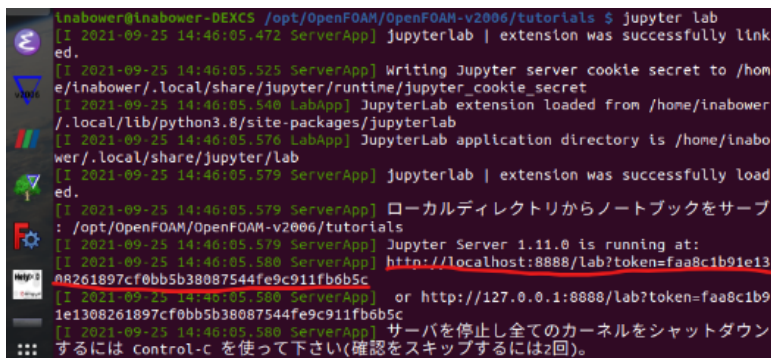


4. 以下を実行し必要なライブラリをインストールします

```
sudo apt install -y python3-pip  
sudo pip3 install jupyterlab optuna PyFoam
```

5. 以下を実行し表示される URL をクリックします

```
cd (任意の場所)  
jupyter lab
```



6. FireFox で Jupyter が表示されたことを確認します。

→ D. 動作確認へ

C. ご自身の環境を使用する場合

本セッションでは『A』もしくは『B』で環境を構築することを前提として講習させていただきます。
ご自分の環境で参加を希望される方は、バージョンや設定の差異により講習内容を再現できない可能性もあることをご了承ください。

それを踏まえたうえで、以下に示す本セッションで用いるライブラリの導入をお願いします。また、環境構築のヒントとして [Dockerfile](#) もご参考にしてください。

※最重要：OpenFOAMの\$FOAM_ETC/bashrcが読み込まれた状態でJupyterを実行する必要があります。

Dの動作確認が再現されない場合にはAもしくはBをご利用ください。

- OpenFOAM-v2006
- Python3 (3.8.10)
 - Jupyter lab (3.1.13)
 - Optuna (2.9.1)
 - PyFoam (2021.6)

→ D. 動作確認へ (必ず動作確認を行ってください)

D. 動作検証

1. OpenFOAM 実行可能な Jupyter Lab を開きます。

(A の場合)

```
cd (任意の場所)
docker run --rm -v ${PWD}:/home/work -p 1111:8888 inabower/opencae-training2021-ofpy
```

→ WEB ブラウザで”localhost:1111/lab”

(B の場合)

スタートメニューから”OF-v2006Terminal を起動

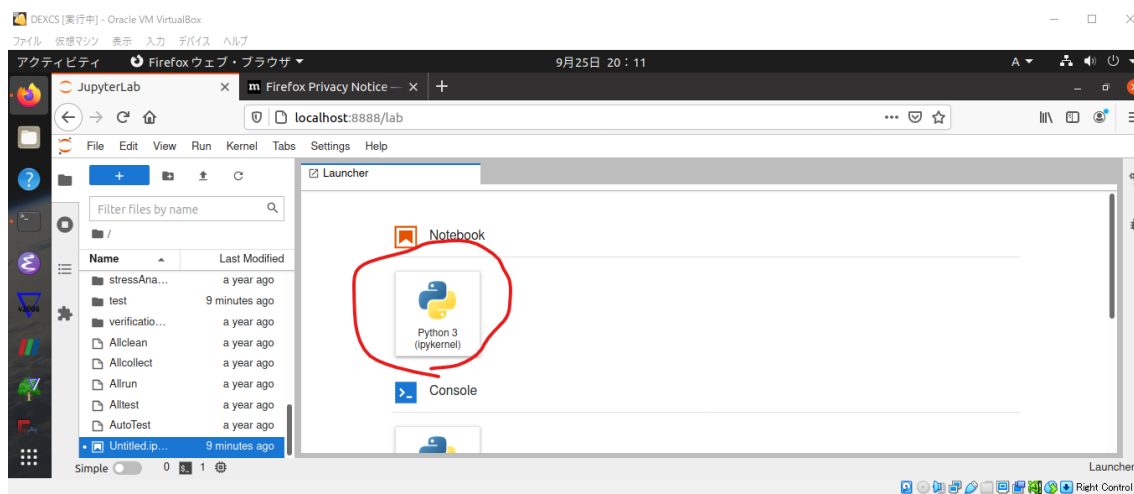
```
cd (任意の場所)
jupyter lab
```

→ 表示されるリンクをクリック

(C の場合)

```
cd (任意の場所)
source ???/etc/bashrc # 各自の OF の環境設定ファイルを読み込む
jupyter lab
```

2. Python3 の Notebook を新規作成します。



- 3.
4. 以下のコマンドをセルに入れて(Shift+Enter)などで実行したときに OpenFOAM 由来のエラーメッセージが出力されることを確認します。

```
import subprocess
subprocess.call(['simpleFoam'])
```

```
[17]: import subprocess
subprocess.call(["simpleFoam"])

-----
| \ \ / \ / F i e l d      | OpenFOAM: The Open Source CFD Toolbox
| \ \ / \ / O p e r a t i o n | Version: v2006
| \ \ / \ / A n d             | Website: www.openfoam.com
| \ \ / \ / M a n i p u l a t i o n |
| \ \ / \ /                   |
-----
Build : 845f816f5b-20200629-OPENFOAM-2006
Arch : "LSB;label=32;scalar=64"
Exec : simpleFoam
Date : Sep 25 2021
Time : 14:50:39
Host : inabower-DEKCS
PID : 29631
I/O : uncollated
Case : /opt/OpenFOAM/OpenFOAM-v2006/tutorials
nProcs : 1
trapFpe: Floating point exception trapping enabled (FOAM_SIGFPE).
fileModificationChecking: Monitoring run-time modified files using timeStampMaster (fileModificationSkew 5, maxFileModificationPolls 20)
allowSystemOperations: Allowing user-supplied system call operations

// * * * * *
Create time

--> FOAM FATAL ERROR:
cannot find file "/opt/OpenFOAM/OpenFOAM-v2006/tutorials/system/controlDict"

From virtual Foam::autoPtr<Foam::Istream> Foam::fileOperations::uncollatedFileOperation::readStream(Foam::regIOobject&, const Foam::file&, const Foam::word&, bool) const
in file global/fileOperations/uncollatedFileOperation/uncollatedFileOperation.C at line 546.

FOAM exiting
```

上のようなメッセージが出たら OK。(Python 上で simpleFoam を実行可能であると判断できる)

5. その他の講習で扱うライブラリが使用可能であることも確認します。

```
import PyFoam, optuna
```

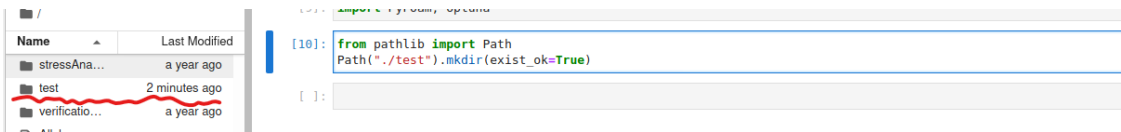
```
[9]: import PyFoam, optuna
```

```
[ ]:
```

上のように何も表示されなければ OK。(optuna と PyFoam を実行可能であると判断できる)

6. 現在のディレクトリにファイルの作成などが可能であることも確認します。

```
from pathlib import Path
Path('./test').mkdir(exist_ok=True)
```



```
[10]: from pathlib import Path
Path('./test').mkdir(exist_ok=True)

[ ]:
```

“test”ディレクトリが作成されていれば OK。(Python でファイルの入出力が可能と判断できる)

7. 以上です。お疲れさまでした。

以上