

## OpenFOAM コントリビュート活動

稲葉竜一<sup>1†</sup> 松原 大輔<sup>2</sup> @tkoyama010<sup>1</sup>

<sup>1</sup>OpenFOAM-jp <sup>2</sup> オープン CAE 勉強会

## OpenFOAM Contributing Activities

Ryuichi INABA<sup>\*†</sup> Daisuke MATSUBARA<sup>\*\*</sup> @tkoyama010<sup>\*</sup>

<sup>\*</sup>OpenFOAM-jp <sup>\*\*</sup>OpenCAE Local user group

### Abstract

The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.

**Keywords:** GitHub, git, Pull Request, OpenFOAM, develop

### 1. はじめに

現状日本からの OpenFOAM へのコントリビュートは少ない。そこで有志により”OpenFOAM-jp”といグループを立ち上げ、本家の開発やドキュメントの翻訳、チュートリアル作成などのコントリビュート活動を始めた。本報告ではその活動内容などについて発表する。

### 2. Foundation 版と ESI 版の違い

OpenFOAM は、Imperial Collage で開発された商用 CFD コード FOAM(Field Operation And Manipulation) が、2004 年に OpenFOAM と名前を変えてオープンソースとしてリリースされたことから始まる。[1][2] 2011 年には OpenFOAM を運営している OpenCFD 社が SGI 社に買収され、またその 2 社により OpenFOAM の商標を譲渡した OpenFOAM Foundation Inc. を設立した。これ以降 OpenFOAM は OpenFOAM Foundation Inc. より開発と配布が行われており、2019 年 11 月現在 OpenFOAM-V7 を最新版とするフォークが”Foundation 版”[3][4] と呼ばれている。

一方で OpenFOAM には多くのフォークが存在する。2012 年以降は ESI 社が OpenCFD 社の買収という形から開発に加わるようになり、2016 年には OpenFOAM-v3.0 からフォークした OpenFOAM-v3.0+ をリリースした。2019 年 11 月時点ではこの最新版は OpenFOAM-v1906 であり、このフォークが”ESI 版”[5][6] と呼ばれる。本報告ではこの Foundation 版と ESI 版の 2 つについてコントリビュートの流れなどを報告する。

### 3. ESI 版コントリビュート方法

ESI 版の開発は GitLab で行われている。[develop.openfoam.com/Development/OpenFOAM-plus](develop.openfoam.com/Development/OpenFOAM-plus)

### 4. Foundation 版コントリビュート方法

Foundation 版の Repository は Github で公開されているが、issue の管理は Foundation の HP の issue Tracking のページで行われる。

メールを送る → Henry G. Weller から登録用紙 (THE OPENFOAM FOUNDATION CONTRIBUTOR AGREEMENT) が送られてくる → 同意できる場合はサインと共に返送 → reporter として登録される。

<sup>†</sup> E-mail address of corresponding author: inabower@gmail.co.jp

## 5. Git の使い方

本節では <https://github.com/OpenFOAM-jp/OpenFOAM-jp.git> のリポジトリにコントリビュートの練習をする方法について解説します。環境は Ubuntu18.04 の環境を想定します。事前に GitHub のアカウントを作成してください。コントリビュートをする際にはまず、Issue を立て自分が加えたい変更について議論します。

機能の簡単な説明\*\*機能リクエストは問題に関連していますか？ 記述してください。 \*\* 問題が何であるかの明確で簡潔な説明。

\*\*希望するソリューションを説明してください\*\* あなたが何をしたいのかについての明確で簡潔な説明。

\*\*検討した代替案を説明してください\*\* 検討した代替ソリューションまたは機能の明確で簡潔な説明。

\*\*追加のコンテキスト\*\* 機能リクエストに関する他のコンテキストまたはスクリーンショットをここに追加します。

OSS のコントリビュートのバージョン管理ソフトには Git が一般的に使用されています。まずは、バージョン管理ソフト git をインストールします。

```
$ sudo apt install git
```

必要なのはソースだけで貢献しないのであれば、次のコマンドでクローンすることで済みます。

```
$ git clone https://github.com/OpenFOAM-jp/OpenFOAM-jp.git
```

貢献をしたい場合は、OpenFOAM-jp のリポジトリを自分のアカウントにフォークします。

TODO: フォークの際の画面キャプチャを挿入する。

フォーク後は自分のアカウントのリポジトリをクローンします。

```
$ git clone https://github.com/your_account_name/OpenFOAM-jp.git
```

クローンをしたら自分の環境でテストを実行してください。

TODO: テスト実行のコマンドを記述する。

テストが全てパスしたらソースを変更します。master ブランチで直接変更することはできないため、ファイルを変更する前に開発ブランチを作成してください。

```
$ git branch branch_name
```

```
$ git checkout branch_name
```

branch\_name には任意の名前を入れてください。自分のアカウント名と加えたい変更について言及されると分かりやすいです。最初のコマンドでブランチを作成し、2 番目のコマンドでブランチに移動します。これにより、変更を行う準備はほぼ完了です。変更のラベルを付けるために、連絡先の名前と電子メールを以下のコマンドで指定します。

```
$ git config --global user.name "Your Name Comes Here"
```

```
$ git config --global user.email you@yourdomain.example.com
```

もし src/toto.cc というファイルをいくつか変更したり、新しいファイルとして追加したら、ローカルのコミットは次のコマンドで行います。"Your extensive commit message here #1" には変更に関するメッセージを追加します。#1 の部分は自分が追加したイシューの番号としてください。

```
$ git add src/toto.cc
```

```
$ git commit -m "Your extensive commit message here #1"
```

この段階ではコミットはあなたのローカルリポジトリで行われていますが、GitHub リポジトリでは行われません。十分なテストで変更を検証したら、以下のコマンドで GitHub の自分のアカウントのリポジトリに変更を移すことができます。

```
$ git push origin branch_name
```

TODO: コマンドのメッセージを含める

このコマンドのメッセージに図のような URL が表示されます。URL にアクセスしプルリクエストを作成します。

TODO: GetFEM のドキュメントから引用を行っているため言及する。

GetFEM++ のマスターブランチにマージすることは許可されていないので、あなたの役割はここで終わります。プルリクエストのページで管理者や他の開発者と議論することができます。管理者が承認した場合には変更

がマージされます。

いくつかの便利な git コマンドを示します。

\$ git status : status of your repository / branch

\$ git log -follow "filepath" : Show all the commits modifying the specified file (and follow the eventual change of name of the file).

\$ gitk -follow filename : same as previous but with a graphical interface

## 6. GitHub および Travis による継続的インテグレーション

Travis CI は、GitHub 上のソフトウェアのビルドやテストを行う、オンラインで分散型の継続的インテグレーション (CI) サービスです。継続的インテグレーションサービスとは、ビルドやテストを継続的に自動実行するサービスです。継続的インテグレーションを行うことにより、機能追加やリファクタリングによるデグレードを防ぐことができます。以下に、GetFEM++ で使用している .travis.yml を示します。リポジトリに yml ファイルを追加することで、Travis によるインテグレーションテストが実行されます。今後 OpenFOAM のリポジトリでも CI が実行できるようにする予定です。

```
language: python
python: - "3.6"
sudo: false
dist: bionic
cache:
  directories: - $HOME/.cache/pip
before_install: - sudo apt-get install -y --no-install-recommends automake - sudo apt-get install -y --no-install-recommends libtool - sudo apt-get install -y --no-install-recommends make - sudo apt-get install -y --no-install-recommends g++ - sudo apt-get install -y --no-install-recommends libqtd-dev - sudo apt-get install -y --no-install-recommends libqhull-dev - sudo apt-get install -y --no-install-recommends libmumps-seq-dev - sudo apt-get install -y --no-install-recommends liblapack-dev - sudo apt-get install -y --no-install-recommends libopenblas-dev - sudo apt-get install -y --no-install-recommends libpython3-dev - sudo apt-get install -y --no-install-recommends ufw - sudo apt-get install -y --no-install-recommends imagemagick - sudo apt-get install -y --no-install-recommends fig2dev - sudo apt-get install -y --no-install-recommends texlive - sudo apt-get install -y --no-install-recommends xzdec - sudo apt-get install -y --no-install-recommends fig2ps - sudo apt-get install -y --no-install-recommends gv - pip install -r requirements.txt
addons: apt: update: true
script: - bash autogen.sh - export CXXFLAGS=-coverage - export LDFLAGS=-coverage - export CPPFLAGS=-coverage - export CFLAGS=-coverage - export FCFLAGS=-coverage - ./configure --with-pic - make -j8 - make -j8 check - (cd doc/sphinx; make html)
after_success: - bash <(curl -s https://codecov.io/bash)
```

TODO Travis CI について Wikipedia からの引用であることを明記する。

TODO 継続的インテグレーションサービスについて Wikipedia からの引用であることを明記する。

## 7. まとめ

### 参考文献

- [1] ESI Group, OpenCFD Inc. OpenFOAM History, <https://www.openfoam.com/history/>, (accessed 2019-11-24).
- [2] 三邊 考志 OpenFOAM 開発・ファンディングプロジェクトの概要と事例 オープン CAE シンポジウム 2015 梗概集, GP23, 2015. [http://www.opencae.or.jp/wp-content/uploads/2016/01/OpenCAESymposium2015\\_GP23\\_Abstract.pdf](http://www.opencae.or.jp/wp-content/uploads/2016/01/OpenCAESymposium2015_GP23_Abstract.pdf), (accessed 2019-11-24).
- [3] OpenFOAM Foundation Inc. openfoam.org, <https://openfoam.org/>, (accessed 2019-11-24).
- [4] OpenFOAM Foundation Inc. OpenFOAM-dev, <https://github.com/OpenFOAM/OpenFOAM-dev>, (accessed 2019-11-24).
- [5] ESI Group, OpenCFD Inc. openfoam.com, <https://openfoam.com/>, (accessed 2019-11-24).
- [6] ESI Group, OpenCFD Inc. OpenFOAM-plus, <https://develop.openfoam.com/Development/OpenFOAM-plus>, (accessed 2019-11-24).