



第 20 回オープン CAE 初心者勉強会 @ 岐阜

# OpenFOAMの利用についての試み

2013年3月16日

春日 悠

# 目次

- 自己紹介
- OpenFOAMの利用についての試み

# 自己紹介

# 自己紹介

- 氏名: 春日 悠 (かすが ゆう)
- 出身地: 福岡県春日市
- 所属: 九州の設計会社の数値解析部署
- 業務: 流体解析、システム開発

# PENGUINITIS

PENGUINITIS!



## 更新情報

- OpenFOAM: [ヘルプの表示](#) を追加 (2013.3.14)
- OpenFOAM: [非定常解析ソルバーによる定常解析](#) を追加 (2013.3.13)
- OpenFOAM: [OpenFOAM インストールメモ](#) を更新 (2013.3.7)

## 気になるニュース

2013年3月11日

<http://www.geocities.jp/penguinitis2002/>

# Penguinitis?

Linus Torvalds 氏が、ペンギンに噛まれて “penguinitis” になったとかなんとか

"The Story Behind Tux the Penguin", wired.com



2002年ホームページ開設  
・Linux ユーザーとして Web の  
情報に大変お世話になったの  
で、なにか恩返しをしたかった。

PENGUINITIS (たぶん 2003 年)

- ・Linux、プログラミング、お勉強ノート
- ・現在もテーマ変わらず

いまだに HTML 手打ち



アデリーペンギン親子 (GIMPによる)

# PENGUINITIS以前

- Linux 万歳！オープンソース万歳！
- Slackware, Vine Linux
- Bash/Vim
- C, FORTRAN77

# 現在

- Linux 万歳！オープンソース万歳！でもない
- Linux Mint
- Zsh/Vim
- C, Python, VBA

# 出合い - OpenFOAM

- 2004年: FOAM のオープンソース化当時、構造解析業務を担当
- 2005年: 流体解析チームに配置換え
- 2006年: 上司から OpenFOAM 1.3 の存在を知らされる
- 以降、プライベートにてお勉強目的でちょこちょこおさわりする程度

# OpenFOAM メモ

- Linux メモのひとつとしてインストールメモを置いたのがはじまり
- 知らんうちに増殖
- 結果、“オープン CAE” を掲げるアヤシイおじさんたちに目をつけられた

## 一般

- [OpenFOAM 情報](#)
- [OpenFOAM 事例集](#)

## OpenFOAM 入門

- [OpenFOAM 入門 \(Windows 超特急版\)](#)
- [OpenFOAM 入門](#)
- [OpenFOAM 入門 中級編](#)

## セットアップ

- [OpenFOAM のインストール](#)
- [ParaView のコンパイル](#)
- [OpenFOAM Extend Project 版のインストール](#)
- [OpenFOAM を Intel C++ コンパイラでコンパイルす](#)

## DEXGS

- [VirtualBox 上への DEXGS2012 for OpenFOAM C](#)
- [VirtualBox 上への DEXGS2010-OpenFOAM のイ](#)

# OpenFOAM メモ

- 公開してよかったか？
- 情報をまとめて管理できる、メモの質が多少上がる、どこからでも見られる
- 「会社の保有技術」とか言い出される前に、安全のために先に出しちゃう
- 情報を隠すことで儲けようとする連中をけん制しておく
- 本にならないような細かい情報を集めたい
- 同じようにメモを公開する人が出てこないかなーという淡い期待

## 一般

- [OpenFOAM 情報](#)
- [OpenFOAM 事例集](#)

## OpenFOAM 入門

- [OpenFOAM 入門 \(Windows 超特急版\)](#)
- [OpenFOAM 入門](#)
- [OpenFOAM 入門 中級編](#)

## セットアップ

- [OpenFOAM のインストール](#)
- [ParaView のコンパイル](#)
- [OpenFOAM Extend Project 版のインストール](#)
- [OpenFOAM を Intel C++ コンパイラでコンパイルす](#)

## DEXCS

- [VirtualBox 上への DEXCS2012 for OpenFOAM C](#)
- [VirtualBox 上への DEXCS2010-OpenFOAM のイ](#)

# OpenFOAM メモ

- 後悔: メモを作ることが目的になってしまった
- メンテナンスが大変
- もうやめた

## 一般

- [OpenFOAM 情報](#)
- [OpenFOAM 事例集](#)

## OpenFOAM 入門

- [OpenFOAM 入門 \(Windows 超特急版\)](#)
- [OpenFOAM 入門](#)
- [OpenFOAM 入門 中級編](#)

## セットアップ

- [OpenFOAM のインストール](#)
- [ParaView のコンパイル](#)
- [OpenFOAM Extend Project 版のインストール](#)
- [OpenFOAM を Intel C++ コンパイラでコンパイルす](#)

## DEXGS

- [VirtualBox 上への DEXGS2012 for OpenFOAM C](#)
- [VirtualBox 上への DEXGS2010-OpenFOAM のイ](#)

# OpenFOAM メモ – 今後

- ひっそり更新

## 一般

- [OpenFOAM 情報](#)
- [OpenFOAM 事例集](#)

## OpenFOAM 入門

- [OpenFOAM 入門 \(Windows 超特急版\)](#)
- [OpenFOAM 入門](#)
- [OpenFOAM 入門 中級編](#)

## セットアップ

- [OpenFOAM のインストール](#)
- [ParaView のコンパイル](#)
- [OpenFOAM Extend Project 版のインストール](#)
- [OpenFOAM を Intel C++ コンパイラでコンパイルす](#)

## DEXCS

- [VirtualBox 上への DEXCS2012 for OpenFOAM C](#)
- [VirtualBox 上への DEXCS2010-OpenFOAM のイ](#)

# 記事紹介: OpenFOAM たんけんたい

## 内容

---

1. [最小のプログラム](#)
2. [引数リスト](#)
3. [時間](#)
4. [メッシュ](#)
5. [blockMesh](#)
6. [移動メッシュ](#)
7. [スカラーフィールド](#)
8. [熱伝導](#)
9. [勾配](#)
10. [時間微分](#)
11. [代数方程式ソルバー](#)
12. [データの読み込み](#)
13. [代数方程式ソルバー 2](#)
14. [simpleFoam](#)
15. [熱物性](#)
16. [入出力オブジェクト](#)
17. [輸送特性の追加](#)
18. [輸送特性](#)

- ソースコード探検
- 狭くて深い記事を読みたかった  
たので、自分で書いてしまった  
感じ

# 記事紹介: OpenFOAM プログラミングメモ

- OpenFOAM らしからぬプログラミングをするためのメモ
- ふざけて作って見たが、どうもかなり重要な知識だったらしい? という実感

## セルループ

```
forAll(mesh.cells(), cid)
{
    const cell &c = mesh.cells()[cid];
    ...
}
```

## セルフェイスループ

```
forAll(mesh.cells(), cid)
{
    const cell &c = mesh.cells()[cid];
    forAll(c, cfid)
    {
        label fid = c[cfid];
        ...
    }
}
```

# 情報公開の考え方

- 整理されていなくてもよいので、とにかくたくさんの方が情報がほしい！（と自分に言い聞かせる）
- 整理はグーグルさんにまかせる。
- まだ“ググレ”とは言えない…。
- しばらくは初学者を（ガマンして）丁寧に扱うしかないか？

# OpenFOAMの利用についての試み

# OpenFOAM (再) 入門

- OpenFOAM (Open source Field Operation And Manipulation)
- GNU General Public License (GPL) のもとで公開されているオープンソースの数値流体力学 (CFD) ツールボックス
- 偏微分方程式ソルバー開発用のクラスライブラリ
- 標準ソルバーやツールを多数備える
- 数値流体解析ソルバーというわけではない (そうでないわけでもない)

# OpenFOAM (再) 入門

C++ のシンタックスをフルに活用して高い記述性と拡張性を実現している (ように見える)

```
solve
(
    fvm::ddt(T)
    + fvm::div(phi, T)
    - fvm::laplacian(DT, T)
);
```

ひょっとしたらこれは誤解のもとかもしれない、と最近思います...

どう簡単になっているのか？ がわからないと簡単にならない (実はそんなに簡単でもないし、でもやはり簡単でもあるし)

# OpenFOAM の歴史

- 英国 Imperial College で開発 “FOAM”
- Nabla社、FOAM を販売
- 2004年 OpenCFD社 から OpenFOAM としてオープンソースで公開
- 2011年8月、OpenCFD 社を SGI が買収、OpenFOAM は OpenFOAM Foundation から公開
- 2012年9月、OpenCFD 社を ESI Group が買収

# OpenFOAM は業務で使えるか？

- 機能的な面で使えるか？
- 利用環境
- メッシュ作成
- 設定
- 解析機能
- ポスト処理

# 利用環境

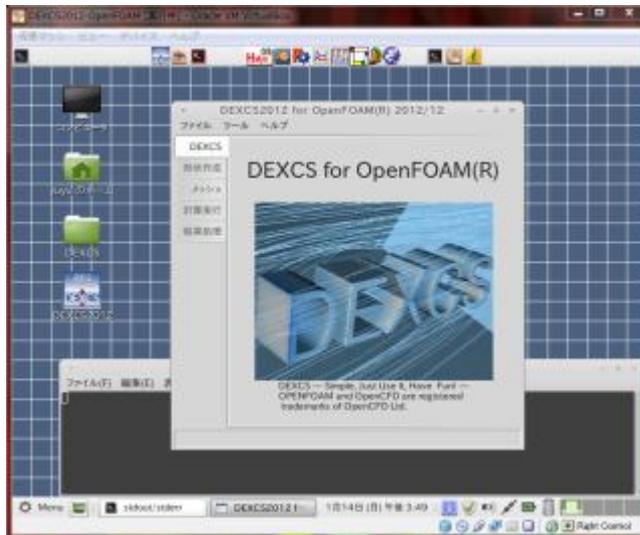
- Linuxマシン
- Windows上のバーチャルマシン
- Windows版バイナリ

# 利用環境: Linux マシン

- 一般の Linux: Ubuntu, openSUSE, Fedora (以上バイナリあり)、Linux Mint (Ubuntu バイナリが使える)、Red Hat Enterprise Linux (いばらの道)
- OpenFOAM インストール済み Linux: DEXCS for OpenFOAM (他にもあるが、特に知る必要はない！)

# 利用環境: バーチャルマシン

- VMWare Player (商用、オープンソース)
- VirtualBox (オープンソース)
- どっちがよい?: 知らん。VirtualBox でよいんでない?
- 遅くない?: 最近の CPU は仮想化支援機能があるのでそうでもない (機能が OFF になっていたらその限りではない)



DEXCS for OpenFOAM on VirtualBox

# 利用環境: Windows 版バイナリ

- Symscape (Caedium の会社) がパッチを公開
- 有志によるバイナリ版がいくつかある
- 周辺スクリプトが使えない (paraFoam とか)
- ソルバー開発を行うには、クロスコンパイル環境が必要

# 利用環境: まとめ

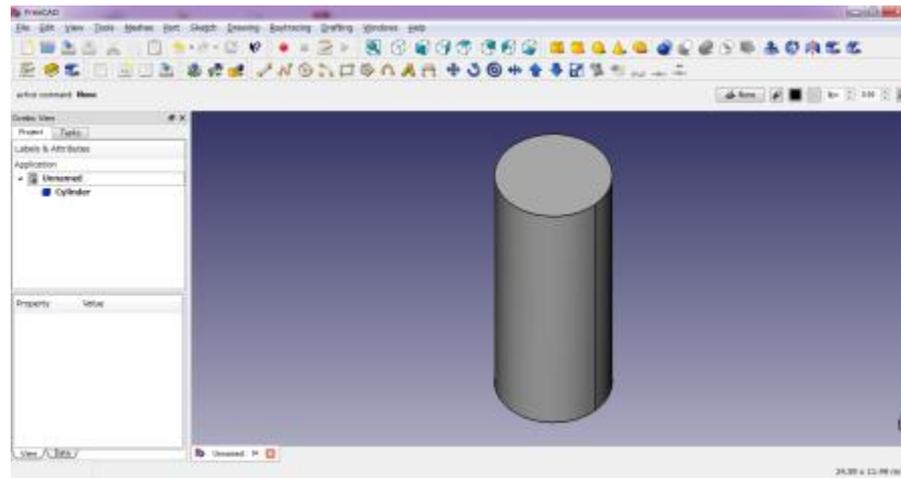
- 理想: Windows 版 (解析専任でない人にとっては特に。設定環境整備が課題)
- 現実 (& 最速): Linux マシン
- 妥協: Windows 上のバーチャルマシン
- おすすめ?: ネットにつながった SUSE 機か? (実際は商用ソルバーの制限やマシンサポートの制限で Red Hat になったり、社内の制限でネットにつなげられなかったりする。こうなるともう大変)

# メッシュ作成

- モデラー
- メッシャー

# メッシュ作成: モデラー

- Blender (3D CG 作成ソフト): かんべんしてほしい。
- FreeCAD: 有力候補。まだ幼い、今後に期待。
- モデラーくらい商用でよいかも (3D CAD くらいあるでしょ?)。



FreeCAD

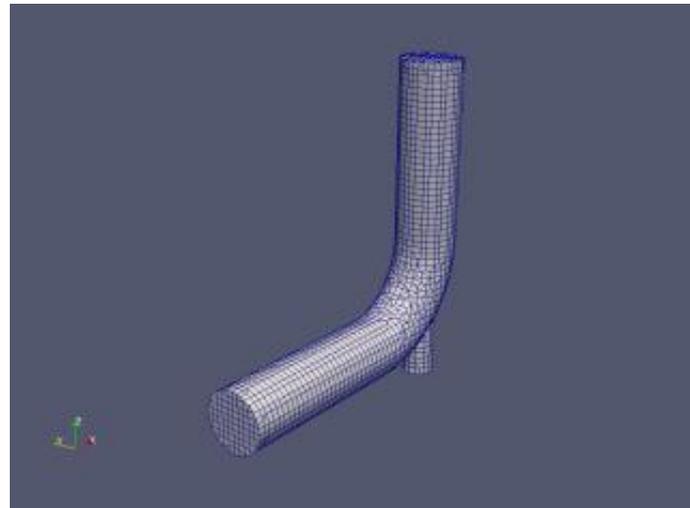
# メッシュ作成: メッシャー

- blockMesh: かんたんなものなら
- snappyHexMesh: 自動メッシャー、使うの難しい
- Netgen: テトラ、悪くない
- enGrid: テトラ、好きになれない
- Gmsh: 考慮すらしてない (考慮ぐらいすべき?)

# メッシュ作成: snappyHexMesh

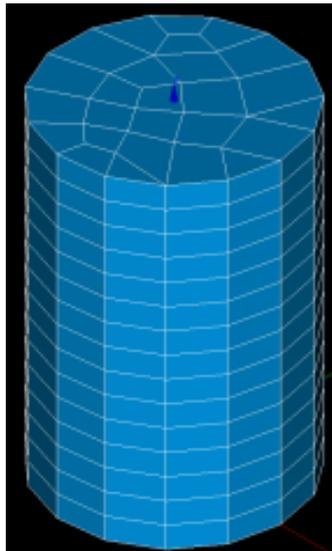
- OpenFOAM 標準自動ヘキサメッシャー (実はこっそりテトラやポリヘドラが混ざるけど)
- STL ファイル (三角形面集合) をもとにする
- 内部流れには向かない?
- 空力解析とか大気解析には便利そう

snappyHexMesh & snapEdge



# メッシュ作成: モデラー&メッシャー

- SwiftSnap: Blender プラグイン、snappyHexMesh を使う (Blender は嫌)
- SALOME: モデル作成もメッシュ作成もいける。テトラ&ヘキサ OK。高機能。使い方が少し独特? 個人的にちょっとしんどい (肩が凝る)。



SALOME によるヘキサメッシュ

# メッシュ作成: 商用

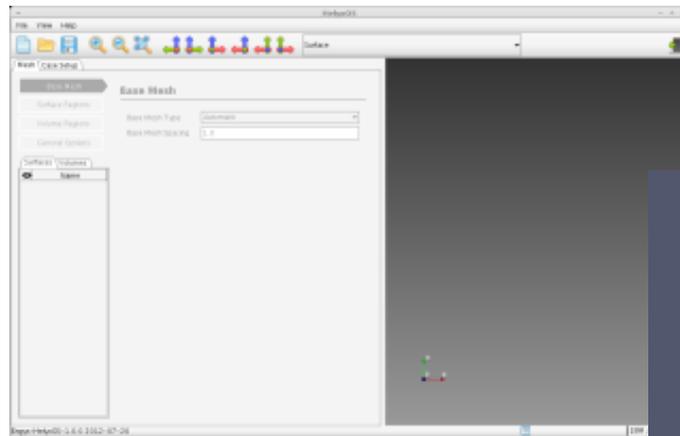
- Cubit: 個人的なお気に入り。いろいろ便利な機能があり、作業効率はよい(慣れれば！)。

# メッシュ作成: まとめ

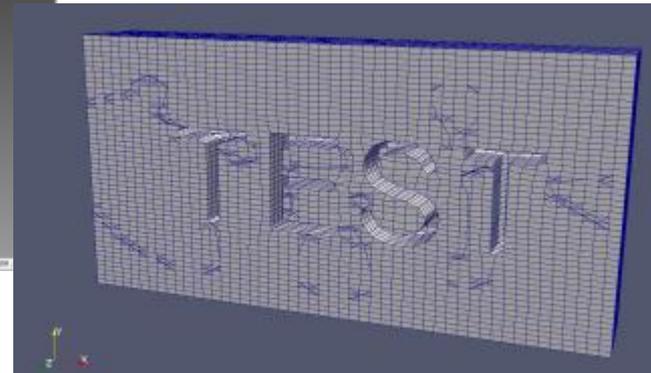
- 手切り派: Cubit
- 手切り派無料主義: SALOME
- 自動派: 商用 CAD + snappyHexMesh
- 自動派無料主義: FreeCAD + snappyHexMesh

# メッシュ作成: HelyxOS

- OpenFOAM 用設定 GUI
- snappyHexMesh の設定をよい感じにやってくれる
- 自動派にはこれで十分？



HelyxOS とメッシュ



# 設定

- テキスト
- GUI

# 設定: GUI v.s. テキスト

- OpenFOAM の設定は基本的にテキストベース
- GUI がぜひともほしい？ GUI のほうが効率がよい？ → そうとも限らない！
- 大規模メッシュ、多数の境界を含むようなモデル: GUI → 読み込み待ち・動作もっさり、テキスト → 該当ファイルの該当箇所だけエディタで編集・速い！
- プログラマの場合、テキストのほうが効率を上げられる場合がある (プレーンテキストがもっとも処理しやすい、場合によっては GUI 備え付けのスクリプトよりも)

# 設定: テキスト

- 手でちまちま: 時間がかかる
- PyFoam: OpenFOAM の設定ファイルをいじる便利なコマンドがある
- Python: 自力でテキスト処理

# 設定: GUI 化

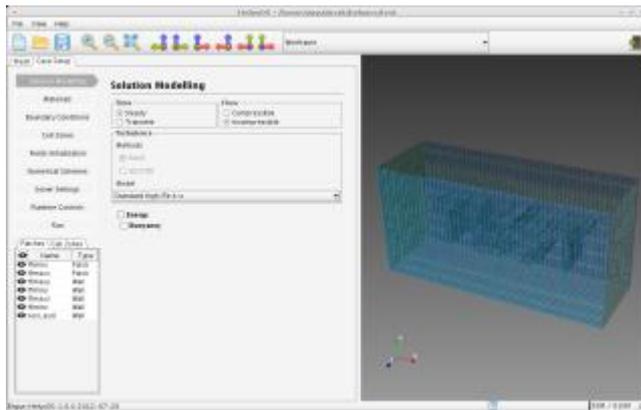
- OpenFOAM は汎用ソルバーとは言い難い: 問題固有の GUI を作り込んではいかぬか？
- OpenOffice.org (LibreOffice) の Calc でフロントエンドを作れば、Excel チックに使えて、現場のおじさんたちも大喜び？ (あるいは Windows 版と Excel で?)

	A	B	C	D	E	F	G
1	キャビティ	OpenFOAM 1.7.1					
2							
3	ソルバー	icoFoam					
4	ベースケース	\$FOAM_TUTORIALS/incompressible/icoFoam/cavity					
5							
6	ケース名	case1					
7							
8	高さ	1.00E-01 m					
9	幅	1.00E-01 m					
10							
11	密度	1.00E+00 kg/m <sup>3</sup>					
12	粘性係数	1.00E-01 Pa·s					
13	nu	1.00E-01 m <sup>2</sup> /s					
14							
15	上面速度	1 m/s					
16							
17	時間刻み	5.00E-03 s					
18	計算回数	100					
19	保存間隔	20					
20	計算時間	0.5 s					
21							
22		計算実行	ポスト実行				
23							
24							

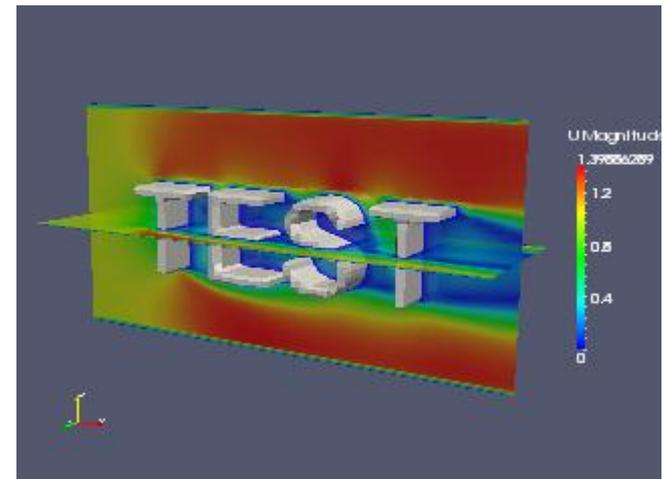
OpenOffice.org によるフロントエンド例  
(OpenOffice.org Basic による)

# 設定: GUI

- HelyxOS: OpenFOAM 用設定 GUI
- 設定簡単、細かい設定はよい感じにしてくれる。
- 難点: 対応ソルバーが限られる → ベース設定を作るのには使えるかも



HelyxOS による設定と計算



# 設定: まとめ

- 汎用: HelyxOS と PyFoam, Python の併用
- 専用: OpenOffice.org (LibreOffice) などによる専用 GUI 作成
- 要検討: Caedium などの利用は？

# 解析機能

- 一般熱流体
- 混相流
- 圧縮性とか: 経験が少ないからノーコメント
- 上記以外

# 解析機能: 一般熱流体

- simpleFoam
- pimpleFoam
- buoyantBoussinesqSimpleFoam
- buoyantBoussinesqPimpleFoam
- buoyantSimpleFoam
- buoyantPimpleFoam
- HelyxOS で済んでしまう？

# 解析機能: 混相流

- VOF (interFoam/interDyMFoam) は結構使える
- 熱計算対応ソルバーがない? のが難点
- まだ調査不足

# 解析機能: 上記以外

- 燃焼: エンジニアリング的には渦消散モデルがほしい。
- ないもの: 頑張って作るしかない。

# 解析機能: カスタマイズ

- ある意味カスタマイズし放題
- 数値流体力学、C++、OpenFOAMの実装、などの知識がいる
- OpenFOAM は (中身いじるなら) 自製ソルバーの一種と思って扱ったほうがよさそう

# 解析機能: まとめ

- 一般熱流体: HelyxOS でよいかも。不安なら商用ソフトも考慮
- 混相流: 等温 VOF なら...要調査
- その他: 自製?

# ポスト処理

- ParaView: 文句なし！
- 情報が少ない...情報の整備が必須。

# OpenFOAM は業務で使えるか？

- 機能的な面で使えるか？
- 一般熱流体はいけそう。混相流も等温なら。
- それ以外は頑張る必要がありそう。
- 解析部分以外を商用ソフトで補助してもよい。

# OpenFOAM は業務で使えるか？

- 費用的な面で使えるか？
- OpenFOAM は安いのか？

# OpenFOAM は安いのか？

- HelyxOS でできる範囲なら安いかも。
- それ以外は？ 鍵↓
- 情報がすぐに手に入るか？
- 作業効率は？
- 解析難度は？
- 解析規模は？

# 情報がすぐに手に入るか？

- ググっても出てこない。
- ユーザーグループは (過度には) 期待できない。
- 有償サポートを利用すべきかも。

# 作業効率は何？

- ハマると作業効率は著しく落ちる。
- 作業費増える！ 商用ライセンスのほうがマシかも？  
→これはお金の出所 (というか勘定のしかた) によるかも。各企業の事情によるはず。
- 外注に頼むと、作業費そのまま乗っかっちゃう。
- 何とかして効率化を図るか (GUI 化など)...別の方法で回収を図るか

# 解析難度は？

- 標準ソルバーでできない？ → 作るしかない。
- 結局、OpenFOAM の問題というより、自製ソルバーを取るか商用ソルバーを取るかの問題。
- 自社の技術力を横目で見つつ、ながーい目と寛容な心で先を見て決めるしかないか。

# 解析規模は？

- 規模は小さい？ → 商用ソルバーならすぐ終わる
- 規模は大きい？ → 大きなマシンが使えれば、商用ソルバーの大規模並列ライセンスの皮算用分を丸儲けできる！
- 解析規模が大きい場合、必要な出費を躊躇しなければ、トータルコストで得する可能性あり。

# OpenFOAM は業務で使えるか？

- 費用的な面で使えるか？
- 簡単な解析なら、タダでできると考えてよいかも。
- ちょっと難しい解析なら、高くつくかも。それを挽回できる方法 (ツール開発への出費など) か、それでもやる意義がなにかなければ、商用でよい。
- 大規模解析であれば、環境 (クラスタとか外部のスパコンとかクラウドサービスとか) を用意できれば、トータルコストで安くできる可能性がある (結局は高くなる可能性はあるが、「より安く対応力を強化できる」と考える)。

# その他

- 情報の問題

# 情報の問題

- 情報、公開してくれませんか？
- 学生さん: 難易度的に難しいか...
- 企業さん: 企業秘密的に難しいか...
- どうにかありません？

# 情報の問題: 企業は情報を出せるか？

- 企業は基本的にそんなにホイホイ情報を出せない。
- オープンソースを使っているのに、許せない？ → むしろ情報を隠すためにオープンソースを使う可能性もある (特殊な改造をソルバーに施したいが、商用ベンダーに情報を出したくないし、自製もめんどくさいとか言った場合)。法的な問題はなんもないし...
- 事例なら... → 事例聞いたってたいして役に立ちやしねえ！
- 必ずしも業務の内容に触れる部分がほしいわけではない。プリミティブな部分を切り出してもらえないか？ → めんどくさいし、何の得が？
- プライベートを犠牲にしない、気楽な楽しい仕組み？
- 結局、有志待ち？

# 情報の問題: 種類の問題

- ユーザーはまちまち: 解析専任者、設計者、研究者、学生、プログラマ
- 求めるものもまちまち: 境界条件の設定の方法、結果の出力の方法、理論の実装、理論そのもの、必要なアクセッサ
- レベルも分野もまちまち: 研究者 v.s. 学生、建築 v.s. プラント
- まとめらんない!
- OpenFOAM、“オープンCAE”、といった枠組みに少し無理がある?
- 課題: 教育の問題...

# 情報の問題: OpenFOAM の問題

- 必要な知識: 工学、数値流体力学、C++、クラスライブラリ → 各分野のユーザー自身は必ずしも必要としないものを含む
- ぶっちゃけ、ユーザーグループや勉強会で議論されていることにあまり興味がない: 逆も真でありうる
- 興味のすれ違い、コミュニケーションの困難さ: 教育か、それとも通訳か？

おわり