

**形状最適化機能  
adjointShapeOptimizationFoamの  
チュートリアルを調べてみた  
+ その他話題提供**

OpenCAE学会 SH

# 発表内容

- **adjointShapeOptimizationFoamについて**
  - adjointShapeOptimizationFoamって何？
  - CAEでの構造最適化手法全般について
  - Adjoint法 (随伴変数法) は何もの
  - OpenFOAM の adjointShapeOptimizationFoamを調べる
- **FRIENDSHIP-Frameworkの無料ソフトCaEseFreeを試してみる**
  - CaEseFree とは？
  - インストール方法等
  - 簡単な操作実施例
- **レベルセット法を用いた自由表面流が解けるオープンソースCFD REEF3D を試す**
  - REEF3Dとは？
  - レベルセット法とVOF法
  - インストール方法等
  - チュートリアルDamBreakを試す
- **OpenFOAMで音響解析できるか調べてみた**
  - OpenAcoustics について
  - OpenFOAM の PotentialWaveFoam について

# adjointShapeOptimizationFoamについて①

- a. **adjointShapeOptimizationFoamって何？**  
**OpenFOAM V2.0.0 から実装された形状最適化を有する非圧縮性流体ソルバ→**  
<http://www.openfoam.org/version2.0.0/>  
**Other: There are other new solvers including adjointShapeOptimizationFoam, for optimisation against pressure loss using an adjoint formulation**  
**(↑ということで圧力損失を最小化するものらしいです)**
- b. **Tutorial: /opt/OpenFOAM/OpenFOAM-2.3.x/tutorials/incompressible/adjointShapeOptimizationFoam/pitzDaily**
- c. **流体計算そのものはSimpleFoamと同じ(と思われる)**
- d. **Adjoint法(日本語では随伴変数法とか訳される)による形状最適化機能を有する**
- e. **配管などの内部流れのみ適用可能**  
→ **最適化オタクとして興味があったのだがいままで未着手  
今回初めて調べてみた**

# シミュレーション最適化手法の概要

## 最適化手法全体の概論

### ー最適化手法の例

**制御工学：変分法、最大原理、動的計画法**

**応用数学：**

**線形計画法 LP (Simplex法, 内点法)**

**非線形最適化法 NLP**

**ー制約条件無し：最急降下法, Newton法, 共役勾配法 (CG法)**

**ー制約条件有り：ペナルティ関数法, 逐次2次計画法 (SQP法)**

**経営工学：整数計画法, ミニマックス問題, ゲーム理論**

**情報工学：ニューラルネットワーク (NN), 進化戦略 (ES),**

**遺伝的アルゴリズム (GA), 遺伝的プログラミング (GP)**

**人工生命, シミュレーテッドアニーリング (SA), PSO**

# シミュレーション最適化手法の概要

## 解析分野から見た最適化手法の分類

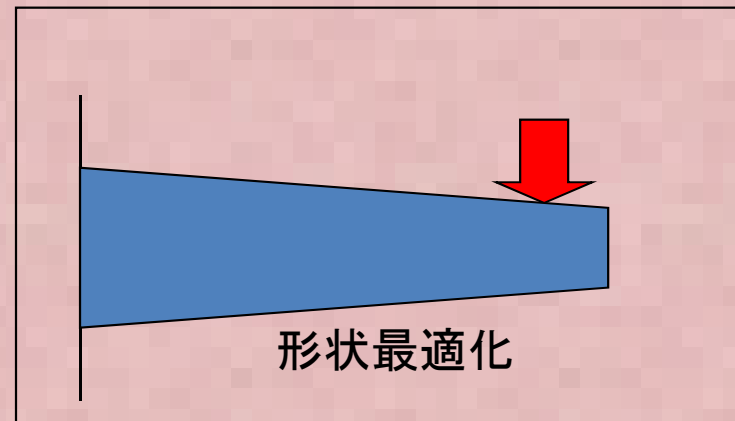
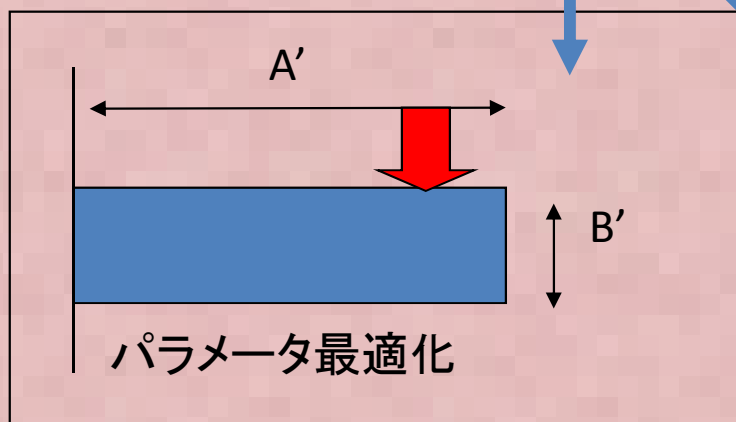
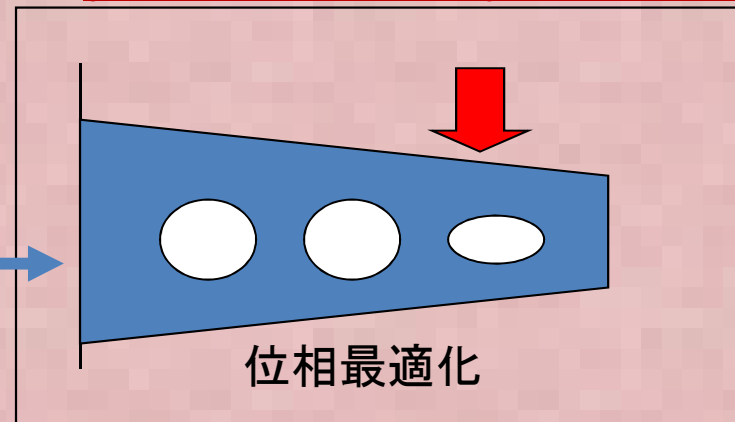
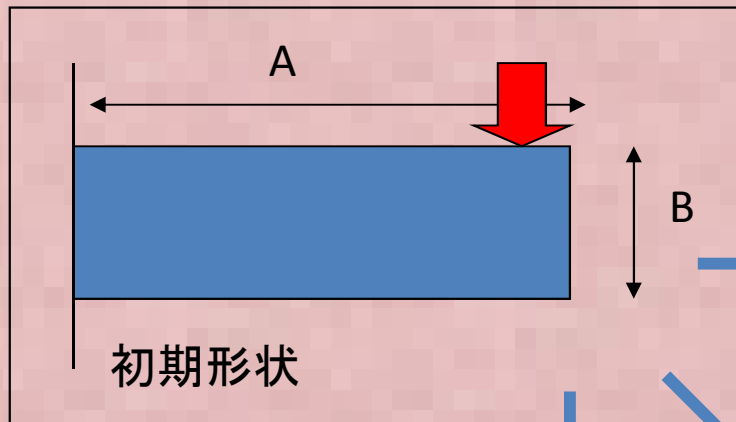
形態による分類：

- パラメータ（寸法）最適化問題：寸法、材料物性などを最適化
- 形状最適化問題：位相を維持しつつ形状を最適化
- 位相最適化問題：位相幾何学形状を最適化

ノンパラメトリック最適化  
Adjoint法はこっち



DAKOTAやOpenMDAOはパラメータ最適化  
(設計変数が有限個)のソフトである



# シミュレーション最適化手法の概要

## ー構造解析分野から見た最適化手法の分類②ー

具体的な解析手法：

ーパラメータ（寸法）最適化問題（有限個のパラメータ）：

1) 近似モデルを用いない

- ・ GAなどの評価関数の勾配を用いないアルゴリズム  
+ FEMプログラムの自動反復による最適化（汎用的）

2) 近似モデルを用いる

- ・ 応答曲面法（RSM）、品質工学（タグチメソッド）
- ・ 感度解析、確率有限要素法、テーラ級数近似（線形問題）

ー形状最適化問題：

- ・ Basis Vector法 (Nastranなどのプログラムに装備)

ベシスベクトル法は、設計が考えられる基本形状候補  
(ベシスベクトル)の組み合わせにより最適形状を求める手法

- ・ 力法 (Azegami et al., 1994) システム応力の一様化物質境界を変数

ー位相最適化問題

密度法・均質化法(菊池 他, Quint) 平均コンプライアンスの最小化

[→類似手法:FSD(全応力設計法) 構成部材応力を許容値以下]

# adjointShapeOptimizationFoamについて②

- **Adjoint法** (随伴変数法/単純に随伴法と呼ぶこともある) は何もの?  
→ 制約条件付きの非線形最適化問題の解法の一つで**ラグランジュの未定乗数法**の一種(というかそのものです)

## 非線形最適化法 NLP

-制約条件無し:最急降下法, Newton法, 共役勾配法 (CG法) → 評価関数の設計変数に対する微分を使う: 例 最急降下法  $y_{n+1} = y_n + K(\Delta y / \Delta x)$

-制約条件有り:ペナルティ関数法, 逐次2次計画法 (SQP法)

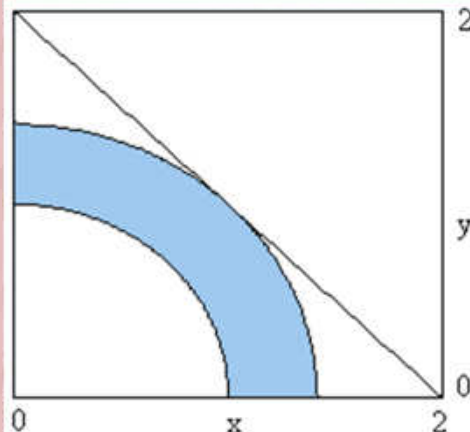
→ このほかにラグランジュ未定乗数法がある

ペナルティ関数法(制約条件式を逆数等にしてペナルティ項として評価関数に足して拡張評価関数に対する制約条件なしの最適化アルゴリズムを適用する方法)

とラグランジュ未定乗数法(制約条件式に新たに定義した変数=ラグランジュ乗数を掛けて、評価関数に足す)は考えは別ものですが、最終的な定式化はほぼ同じような感じになります。(SQP法はより具体的な計算アルゴリズム)

<http://ja.wikipedia.org/wiki/%E9%9D%9E%E7%B7%9A%E5%BD%A2%E8%A8%88%E7%94%BB%E6%B3%95>

(非線形計画法のWIKIページから引用)



## 2次元の例

制約空間(水色)と目的関数(直線)の接点が解である。  
単純な問題の例として、以下の制約条件群があり、

$$x_1 \geq 0 \quad x_2 \geq 0 \quad x_1^2 + x_2^2 \geq 1 \quad x_1^2 + x_2^2 \leq 2$$

次の目的関数を最大化する問題を示す。

$$f(x) = x_1 + x_2 \text{ ここで } x = (x_1, x_2) \text{ である。}$$

# adjointShapeOptimizationFoamについて③

- ラグランジュ未定乗数法計算の簡単な例

制約条件式に新たに定義した変数=ラグランジュ乗数を掛けて、評価関数に足す

<http://mcweb.nagaokaut.ac.jp/~kurahashi/lecture/optimal-design2.pdf>

## 変分法に基づく最適変分法に基づく最適設計

(制約条件付きの汎関数の最小化問題) 倉橋殿資料↓から引用

### 1. ラグランジュの未定乗数法

問)  $N=AB$ という関係のもと、  
AとBの和を最小とするA,Bの値を求めなさい。

評価関数

$$J = A + B$$

制約条件式

$$N = AB \Rightarrow N - AB = 0$$



ラグランジュ関数

$$J^* = A + B + \lambda(N - AB) \quad \lambda : \text{ラグランジュの未定乗数}$$

ラグランジュ関数を最小とするA,Bの値を求める。

ラグランジュ関数が最小となる変数を探すため、全ての変数に対する微分が零となる式を誘導する。

$$\begin{cases} \frac{\partial J^*}{\partial A} = 1 - \lambda B = 0 \Rightarrow B = \frac{1}{\lambda} \\ \frac{\partial J^*}{\partial B} = 1 - \lambda A = 0 \Rightarrow A = \frac{1}{\lambda} \\ \frac{\partial J^*}{\partial \lambda} = N - AB = 0 \end{cases}$$

↓ 代入

$$N - AB = 0 \Rightarrow N - \frac{1}{\lambda^2} = 0 \Rightarrow \lambda = \pm \sqrt{\frac{1}{N}}$$

$$\therefore A, B = \pm \frac{1}{\sqrt{\frac{1}{N}}}$$



# adjointShapeOptimizationFoamについて④

- ラグランジュ未定乗数  $\lambda$  = 随伴変数  
随伴変数法の特徴 (畔上先生 ↓ 下記資料から抜粋)  
<http://www.az.cs.is.nagoya-u.ac.jp/class/adaptive-systems/nuide.html>

## 直接微分法

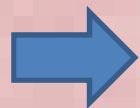
1.2.3 (直接微分法の特徴) 直接微分法は次のような特徴をもつ。

- (1) 直接微分法では、(1.2.5) の計算を  $n$  回繰り返す必要がある。このことは、多設計変数 ( $n > 1$ ) のときに不利となる。もしも、逆行列  $K^{-1}$  を保存しないで、設計変数ごとに  $K^{-1}$  を求め直すときには、差分による断面積微分の計算法と同程度の計算量となる。
- (2) 一方、(1.2.5) の計算を終えて、 $u_p^T$  をいったん求めてしまえば、 $u_p^T$  はすべての評価関数  $f_0, \dots, f_m$  に対して共通に使える。このことは、評価関数が沢山 ( $m > 1$ ) のときに有利となる。  
→ (注  $u_p^T$  というのはこの問題では変位 (構造解析の問題) で最適条件を満たす最適条件を満たす主問題 ( $Ku = f$ ) の解のこと。つまり一度最適解を求めてしまえば評価関数が沢山あっても使い回しできるという意味)

## 随伴変数法

1.2.5 (随伴変数法の特徴) 随伴変数法は次のような特徴をもつ。

- (1) 随伴変数法では、随伴問題の数は評価関数の数  $m + 1$  と一致する。このことは、評価関数が沢山 ( $m > 1$ ) のときに不利になる。
- (2) 一方、随伴問題の数は  $m + 1$  は設計変数の数  $n$  に依存しない。そこで、多設計変数 ( $n > 1$ ) のときには有利になる。
- (3) さらに、随伴問題の変数の数は状態方程式の変数の数  $n$  と一致する。このことは、状態変数が時間領域や物理的空間領域で定義された関数になった場合 (状態空間は無限次元空間になる) でも、随伴変数法は成り立つことを意味する。



つまり随伴変数法は形状最適化や位相最適化など理論的には設計変数の自由度が無限個になるような問題(実際はFEM,FVMなどで離散化するので自由度は有限個になる)に有利である。逆に設計自由度が寸法数個など少ないパラメータで表せるような単純なパラメータ最適化問題には向いていないと考えられる。

# adjointShapeOptimizationFoamについて⑤

- 随伴変数法の特徴 その2:問題点・課題 (大林先生↓下記資料から)

<http://www.ifs.tohoku.ac.jp/edge/publications/chap2.pdf> [CFDによる最適設計の動向科  
研究報告「市場競争力に基づく新しい航空機設計法の研究」平成14年3月第2章] より

1回の解析ですべての設計変数の勾配情報を計算できる画期的な方法として、Adjoint法がある。この方法は上の随伴方程式を利用する。まず流れの支配方程式(2.2)を解く。その情報を利用して随伴方程式(2.4)((2.3)の間違いと思われる)を解きラグランジュ係数 $\xi$ を決定する。そして最適性条件式(2.4)によって $\partial J/\partial g$ を計算する。こうして、得られた勾配情報を用いて非線形計画法を実行する。設計変数の次元がどんなに増えようと1度にすべての設計変数の勾配計算ができるというのは、多くの設計変数を必要とする3次元の複雑形状の設計に対してすばらしい利点である。ただし、目的関数や設計変数を変えるたびに定式化からコーディングまでやり直さなければならないという重大な欠点を持つ。

目的関数を $J(Q, g)$  ( $Q$ :流れの変数、 $g$ :設計変数)とする。流れの変数は当然流れの支配方程式を満たす必要があるので、流れの支配方程式 $F(Q, g) = 0$ を考慮する必要がある。通常 $J$ の値を求めるためにまず $F = 0$ を解いているのでこれが拘束条件であることを意識しないことが多いが、設計上の拘束条件がなくても流れの支配方程式が拘束条件となることに注意しよう。

最適解であることの数学的な条件を求めてみよう[1]。適当な内積を導入しラグランジュ係数 $\xi$ を用いると、ラグランジュ関数

$$L(Q, g, \xi) = J(Q, g) - \langle F(Q, g), \xi \rangle \quad (2.1)$$

が定義できて、変分法によってこの最適解の条件が、

$$\text{流れの支配方程式: } F(Q, g) = 0 \quad (2.2)$$

$$\text{随伴方程式: } \left( \frac{\partial F}{\partial Q} \right)^* \xi - \left( \frac{\partial J}{\partial Q} \right)^* = 0 \quad (2.3)$$

$$\text{最適性条件: } \left( \frac{\partial F}{\partial g} \right)^* \xi - \left( \frac{\partial J}{\partial g} \right)^* = 0 \quad (2.4)$$



目的関数や設計変数の追加変更が必要になった場合、プログラム自体を作りかえる必要がある

# adjointShapeOptimizationFoamについて⑥

- OpenFOAM の adjointShapeOptimizationFoamを調べる その1

ととりあえずtutorial を実行してみる。pitzDaily の1個だけ。

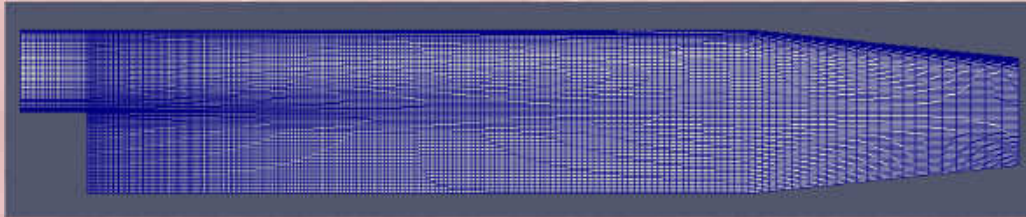
→

`$FOAM TUTORIALS/incompressible/adjointShapeOptimizationFoam/pitzDaily`

- Allrunが入っていないので  
blockMeshを実行後、adjointShapeOptimization を実行する。
- 2D問題なのですぐに計算が終わる。結果をparaFoam で可視化すると次のような各分布図が得られる。通常の流速 $U$ , 圧力 $p$ の他に謎の変数 $\alpha$  (アルファ) と $\alpha$ のついた流速 $U_\alpha$ , 圧力 $p_\alpha$  が計算されている→さてこれは何でしょう？
- OpenFOAMの他例と同様に詳しい説明が書かれた資料は同封されておらずさてどうしたものか？→が！ネット検索したら結構一杯資料が出てきた。

# adjointShapeOptimizationFoamについて⑦

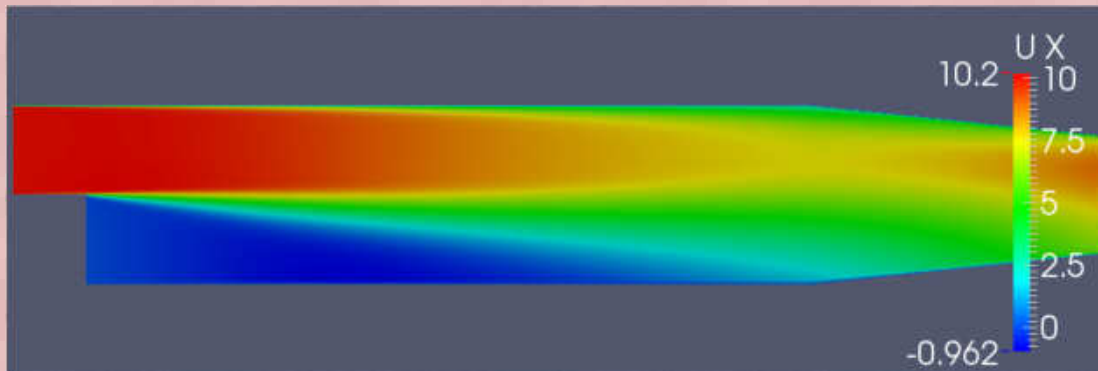
- OpenFOAM の adjointShapeOptimizationFoamを調べる その2  
とくにあえずtutorial/pitzDaily を計算した結果



オリジナルのメッシュ



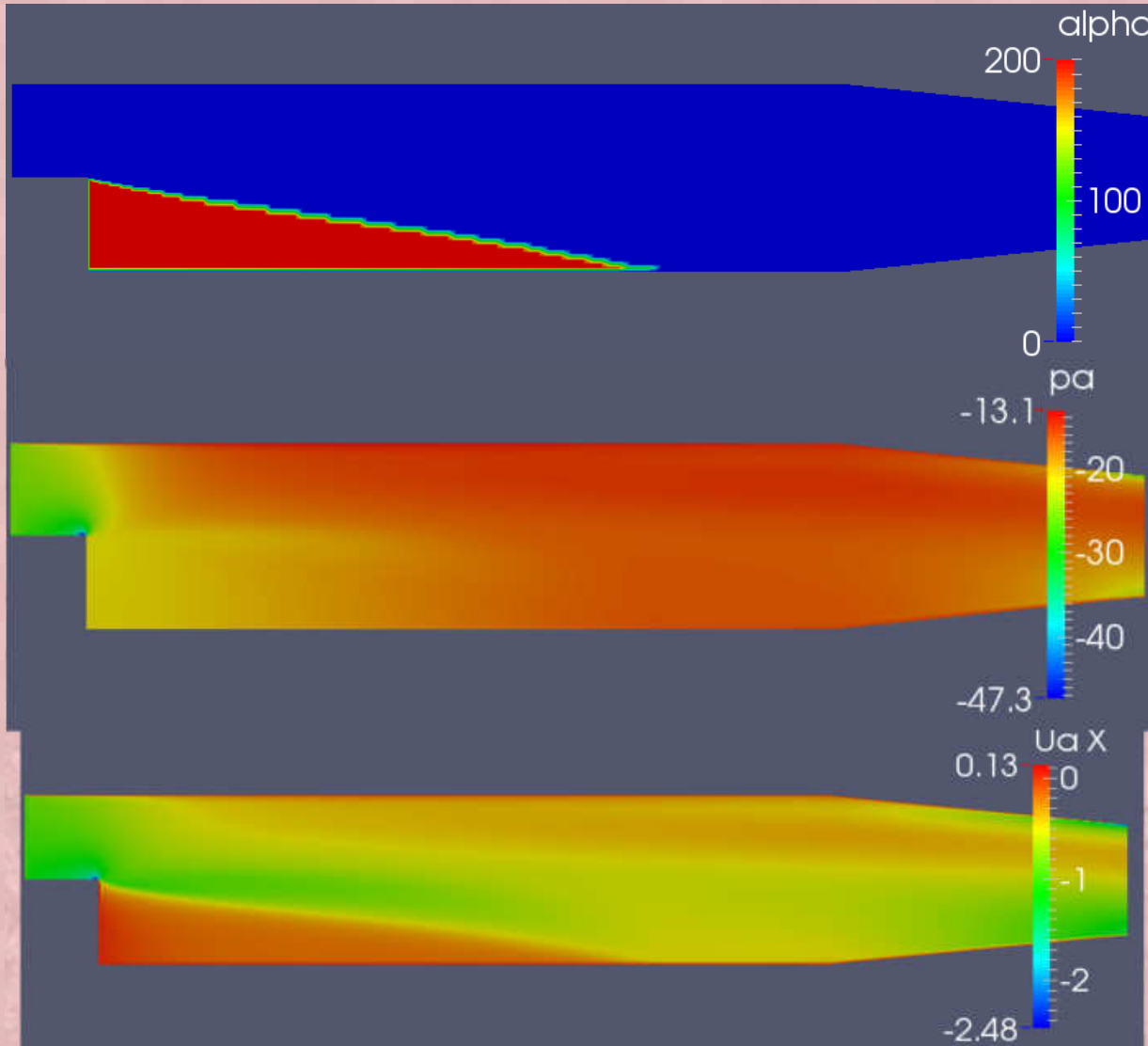
圧力分布 t=1000



流速分布 t=1000

# adjointShapeOptimizationFoamについて⑧

- OpenFOAM の adjointShapeOptimizationFoamを調べる その3  
とくにあえずtutorial/pitzDaily を計算した結果2



なぞの変数:  $\alpha$

圧力?のようなPa  
分布 t=1000

流速?のような  
Uax分布 t=1000



# adjointShapeOptimizationFoamについて⑨

- OpenFOAMの他例と同様に詳しい説明が書かれた資料は同封されておらず  
さてどうしたものか？ → ネット検索したら結構一杯資料が出てきた。

[https://groups.google.com/forum/#!msg/openfoam/n6RtyDJjt2g/rppkHt\\_bT6wJ](https://groups.google.com/forum/#!msg/openfoam/n6RtyDJjt2g/rppkHt_bT6wJ)

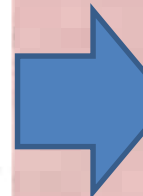
(Google Openfoam group での大淵さんと桜木さんのやりとり) ↓

OpenFOAM初心者のsakumaです。

Openfoamのチュートリアルにある  
adjointShapeOptimizationFoamについてですが、  
計算を流してみるとUa, Paが計算されます。この速度、圧力はど  
のような意味を示  
しているのでしょうか。

また初期値の設定において、inlet側の設定においてマイナス値を入  
れるのは  
adjointShapeOptimizationFoamの流儀と言うか書式でしょうか。

よろしくお願いたします。



大淵さんのコメントから“a”のついた変数は  
随伴変数つまり、ラグランジュ乗数になるとい  
うことが分かった。Aはadjointのa  
(ただし、論文はネットでは概要のみで本文は  
無料公開はされていなかったなので入手断念)

大淵さん： こんにちは。ソルバソースコードに記載されてい  
る文献に詳しい説明があります。

00032 References:

00033 ¥verbatim

00034 "Implementation of a continuous adjoint for  
topology optimization of

00035 ducted flows"

00036 C. Othmer,

00037 E. de Villiers,

00038 H.G. Weller

00039 AIAA-2007-3947

00040

[http://pdf.aiaa.org/preview/CDReadyMCFD07\\_1379/PV2007\\_3947.pdf](http://pdf.aiaa.org/preview/CDReadyMCFD07_1379/PV2007_3947.pdf)

00041 ¥endverbatim

このソルバは、adjoint法(随伴法)で圧力損失を最小化させ  
る最適化ソルバです。

aがついた変数は、それぞれ元になる変数の随伴変数にな  
ります。

以上、ご参考まで。

# adjointShapeOptimizationFoamについて⑩

- より詳しい説明資料：OpenCAE関西の高木先生の資料

[http://ofbkansai.sakura.ne.jp/data/takagi\\_adjoint.pdf](http://ofbkansai.sakura.ne.jp/data/takagi_adjoint.pdf)

→ 随伴変数や定式化について説明あり。これから $\alpha$  は"porosity"「空孔率」とでも訳す？であるとわかる。意味合いとしては密度の逆数であり、この値が高いところは、不要であるということで削るべき場所となる。OpenFOAM の adjointShapeOptimization では 各セルの空孔率 $\alpha$ を設計変数として(従ってセルを細かくすると $\infty$ に設計変数が増える)、目的関数である圧力損失を最小化する計算を行っていることが理解できる。

## Formulation of adjoint method

- Optimization problem

Minimize  $J = J(\alpha, \mathbf{v}, p)$  subject to  $R(\alpha, \mathbf{v}, p) = 0$

where

$J$  : cost function

$\alpha$  : porosity

$\mathbf{v}$  : velocity

$p$  : pressure

## State equations

Incompressible, steady-state Navier-Stokes equations with porosity

$$(R_1, R_2, R_3)^T = (\mathbf{v} \cdot \nabla) \mathbf{v} + \nabla p - \nabla \cdot (2\mu \mathbf{D}(\mathbf{v})) + \alpha \mathbf{v}$$

$$R_4 = -\nabla \cdot \mathbf{v}$$

where  $R$  is the state equations,

$$R = (R_1, R_2, R_3, R_4)^T$$

Introduce a Lagrangian function  $L$ ,

$$L := J + \int_{\Omega} (\mathbf{u}, q) R d\Omega$$

$$(\mathbf{u}, q) = (u_1, u_2, u_3, q) \quad (\text{Lagrangian multipliers})$$

# adjointShapeOptimizationFoamについて⑪

- より詳しい説明資料 :

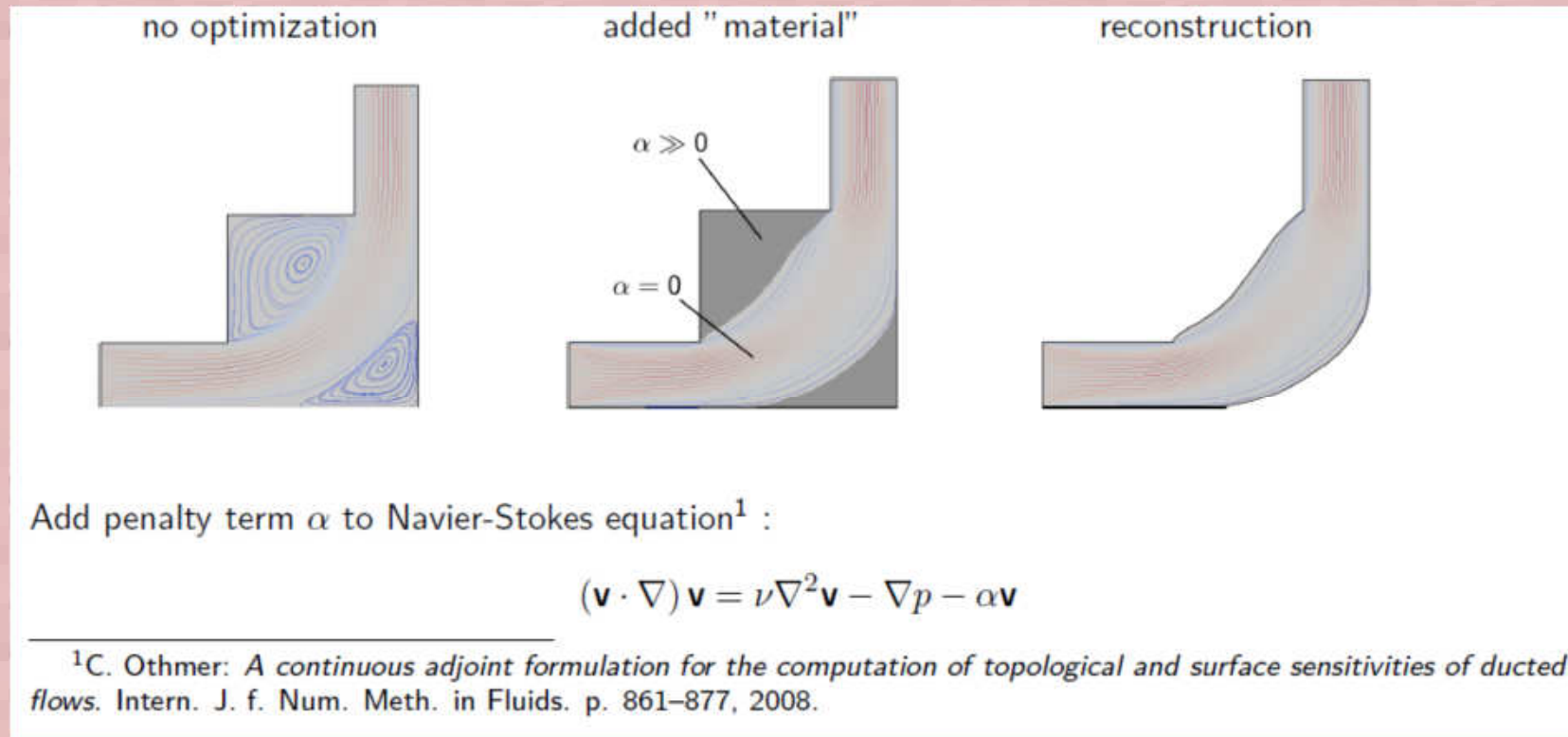
An (more) effective Discrete Adjoint Model for OpenFOAM

Markus Towara, Uwe Naumann

Software and Tools for Computational Engineering Science

RWTH Aachen University

EuroAD 2013, Oxford, 10. December 2013





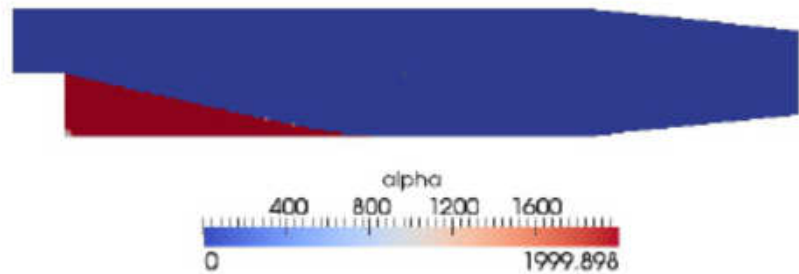
# adjointShapeOptimizationFoamについて⑫

- より詳しい説明資料：

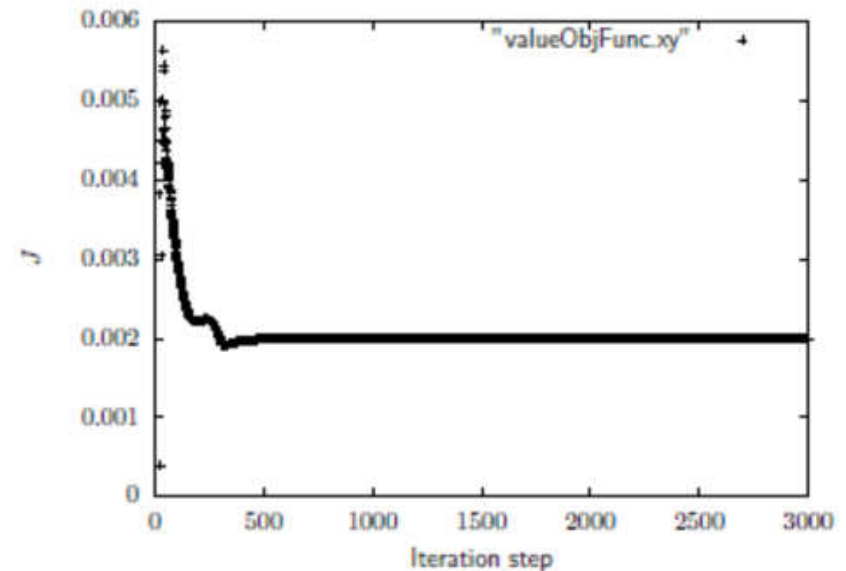
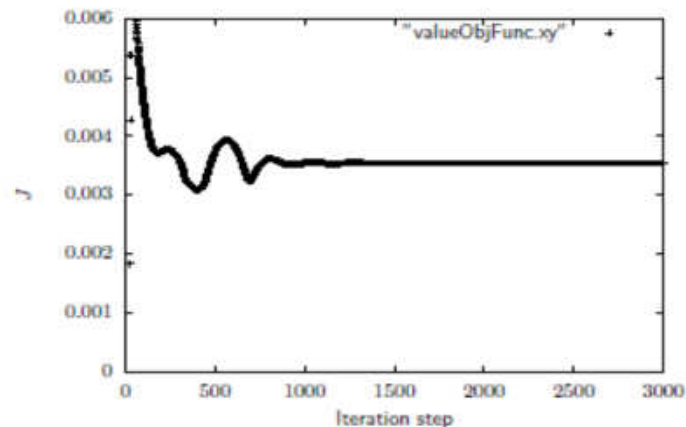
[http://www.tfd.chalmers.se/~hani/kurser/OS\\_CFD\\_2013/](http://www.tfd.chalmers.se/~hani/kurser/OS_CFD_2013/)

↑ソースコード解説とカスタマイズ例、追加チュートリアルがある。

pitzDaily porosity field



pitzDaily value of the objective function, without porosity update



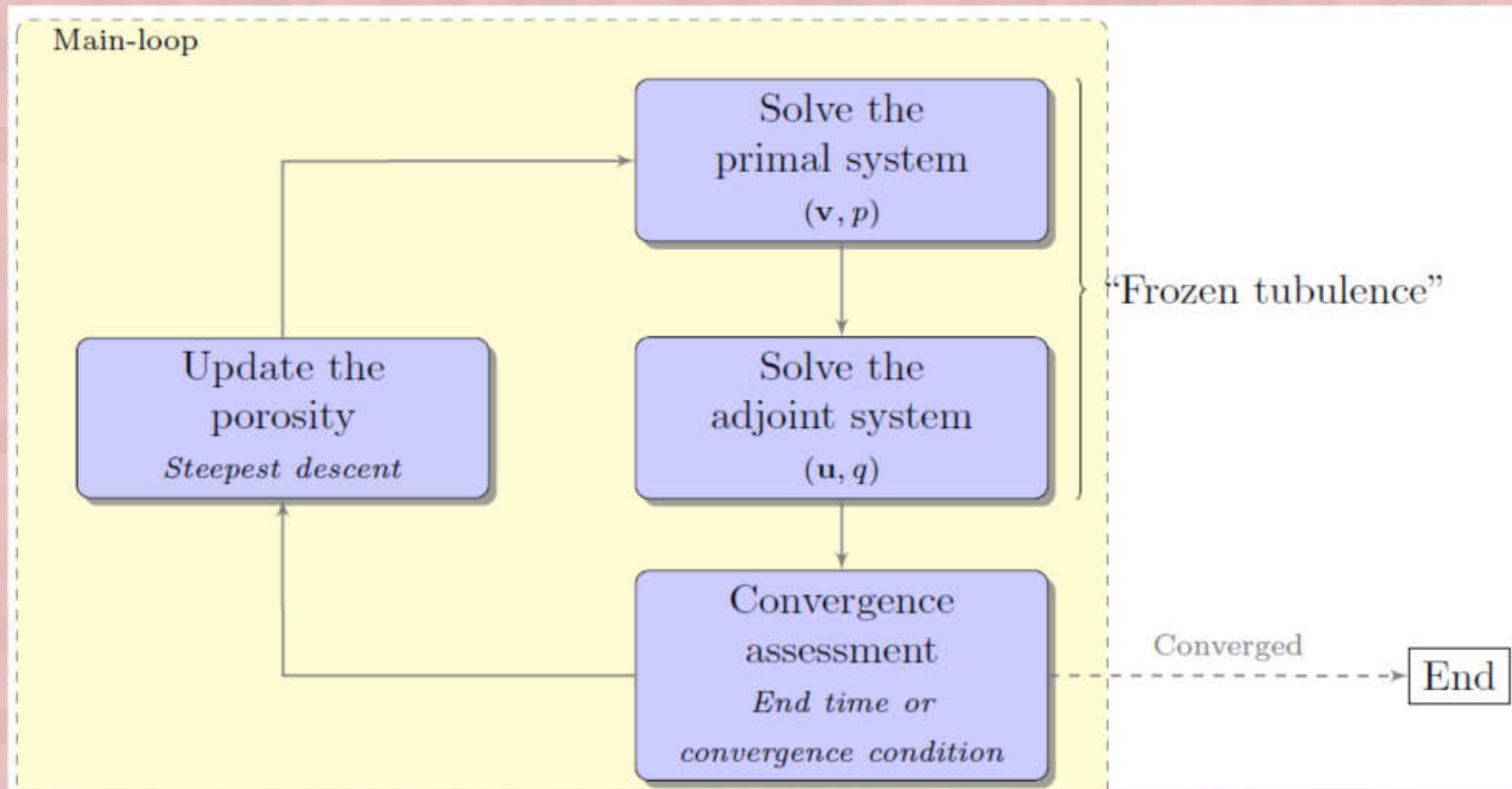
添付のカスタマイズコード：  
UNAdjointShapeOptimizationFoam  
(porosity をupdate しない)をコンパイルして  
みる→展開後、端末でwmakeする。  
OpenFOAM 2.2用のようだが、2.3で問題  
なくコンパイルできた

# adjointShapeOptimizationFoamについて⑬

- より詳しい説明資料：

[http://www.tfd.chalmers.se/~hani/kurser/OS\\_CFD\\_2013/](http://www.tfd.chalmers.se/~hani/kurser/OS_CFD_2013/)

↑ソースコード解説とカスタマイズ例、追加チュートリアルがある。



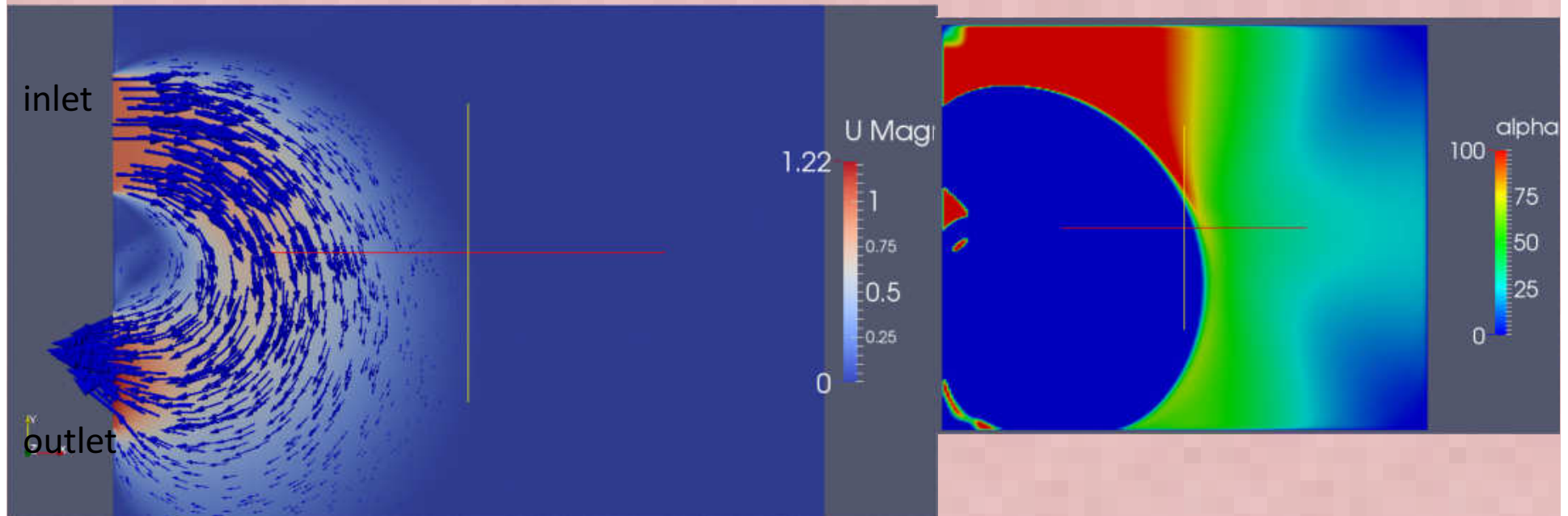
adjointShapeOptimizationFoam の計算手順

# adjointShapeOptimizationFoamについて⑬

- より詳しい説明資料：

[http://www.tfd.chalmers.se/~hani/kurser/OS\\_CFD\\_2013/](http://www.tfd.chalmers.se/~hani/kurser/OS_CFD_2013/)

↑追加チュートリアル実施例



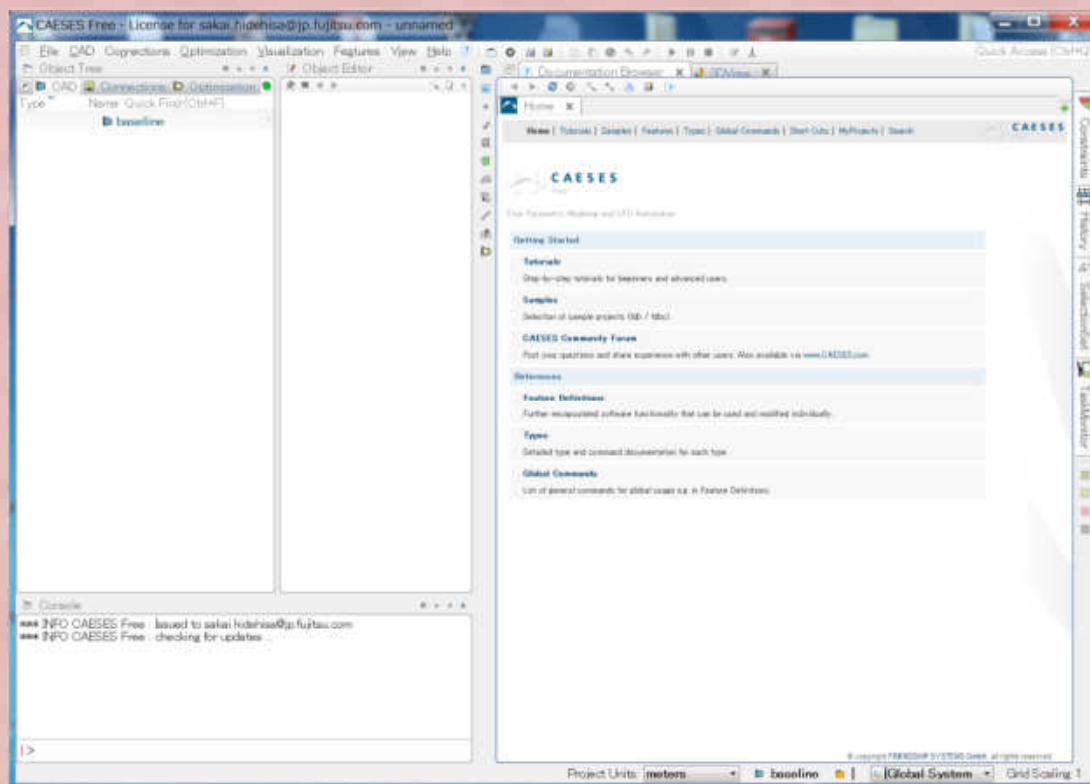
添付のtutorial Pipebending:UNAdjointShapeOptimizationFoam で計算する。Inlet の上側エリアが不要と判断されている模様

# adjointShapeOptimizationまとめ

- 各セルのporosity(空孔率)  $\alpha$  を設計変数として圧力損失を目的関数Jとして形状最適化計算を行うもの。構造解析では位相最適化に相当するものであり、当初境界形状を移動させる形状最適化機能と思っていたが違っていた。
- 随伴変数 (aつきの変数 = ラグランジュ乗数) と空孔率  $\alpha$  が基本変数 (速度、圧力) に加えて同時に計算される。
- 目的関数を圧力損失以外の変数や設計変数をporosity(空孔率)  $\alpha$  以外に追加したい場合はソースコードの変更が必要

# FRIENDSHIP-Frameworkの無料ソフト CAEseFreeを試してみる①

- CAEseFree とは？ OpenFOAMなどのモデル形状作成に無料で使えるCADソフト/プリポストのようなもの？ 商用版CAESE/FFWの機能制限無料版という位置づけ。商用版ではCFDと自動連携して形状最適化ができるみたい？ FREE版でもパラメータSTUDYはできる



形状をパラメトリックに変更可能なように、形状最適化やパラメータスタディに便利のような感じである。



SLIDE SHARE にFNさんの資料が上がっているので参照



Fumiya Nozaki

最終更新日: 2014年6月1日



# FRIENDSHIP-Frameworkの無料ソフト CaeseFreeを試してみる②

## ・インストール方法など

### ▶ [ダウンロードサイト](#)

#### Free Download

If you are a new user, just download your CAESES® Free version from the given links below and register with your name and email address during the installation process. That's it!



↓  
CAESES®/ FPW 3.1 for 32bit Windows  
Tested on Vista, Win7 and Win8.

↓  
CAESES®/ FPW 3.1 for 64bit Windows  
Tested on Vista 64bit, Win7 64bit and Win8 64bit.

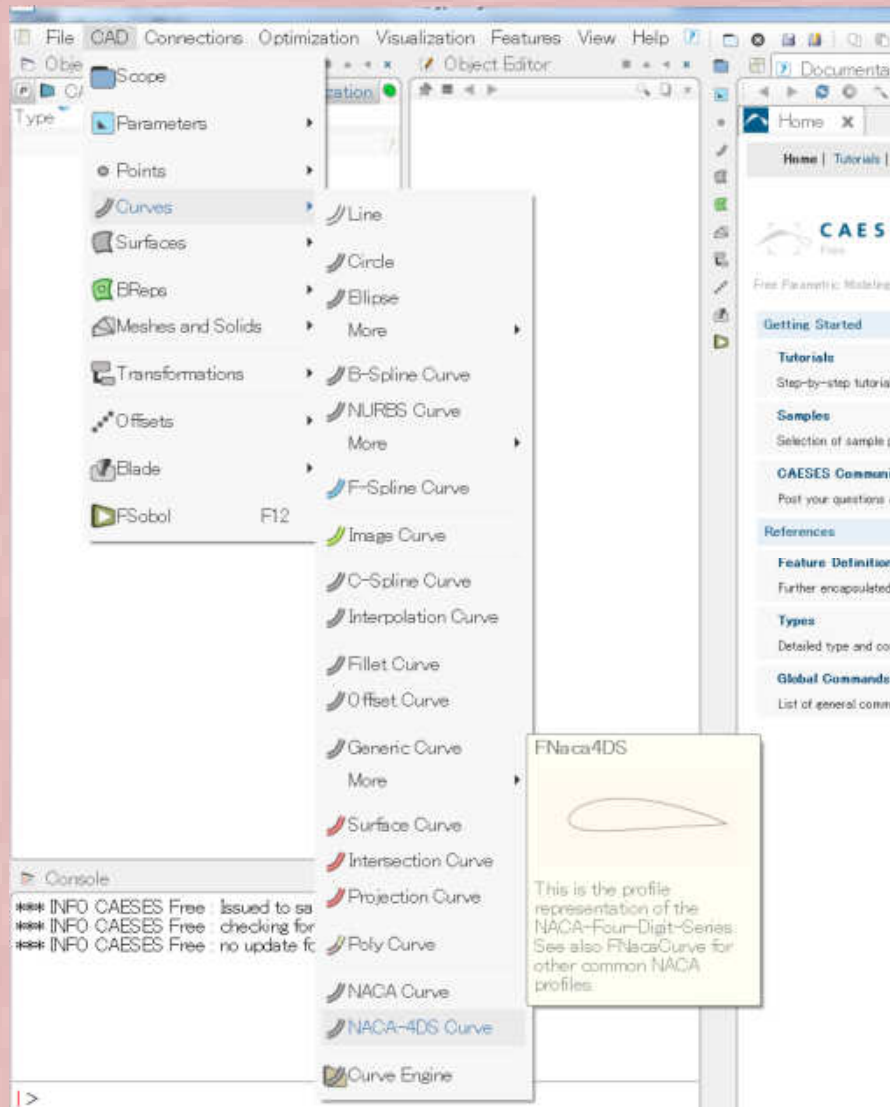
↓  
CAESES®/ FPW 3.1 for 32bit Linux distributions

↓  
CAESES®/ FPW 3.1 For 64bit Linux distributions  
Good results expected with Debian Squeeze, Ubuntu 10.04, 12.04, Suse Enterprise 10.3+ (like SLES12), RedHat 5.4+ (like RHEL5, RHEL 6, SL 6), some other 2.6er kernel distributions. Please install a proprietary 3D GPU driver on your Linux distribution (OpenGL GLX extension required). You have to have a running DBUS message bus.

Windows, Linux ともに  
32bit, 64bit の両方に  
対応しています。

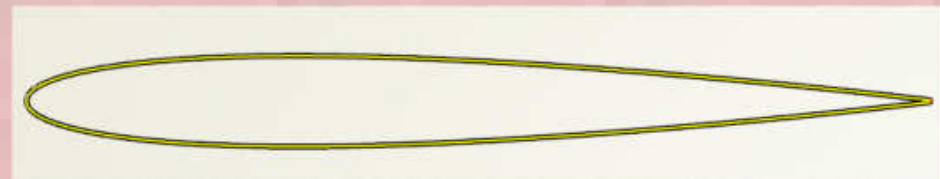
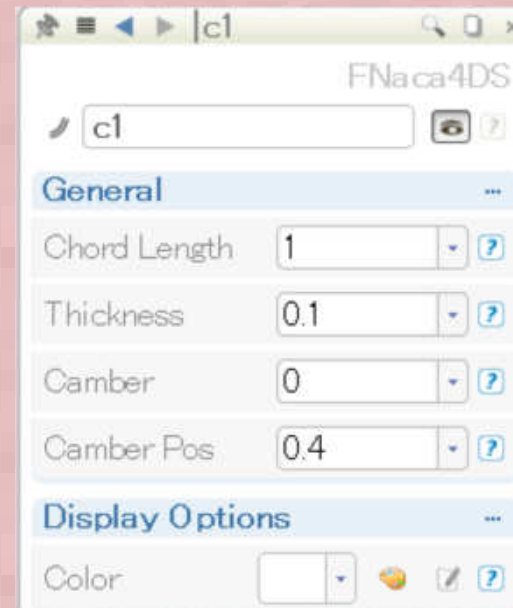
- Windows, Linux の32, 64bit 環境で動作する(Windows7 64bit, Ubuntu12.04LTS 64bit で動作することは確認した)
- Install はHPからモジュールをダウンロードしてWindows はexe を実行、linux は圧縮ファイルを展開するだけ
- 初回起動時にユーザ登録 (Mail アドレス)が必須で、二回目以降もライセンスを確認しにサーバにアクセスするのでネット環境必須

# FRIENDSHIP-Frameworkの無料ソフト CaeseFreeを試してみる③

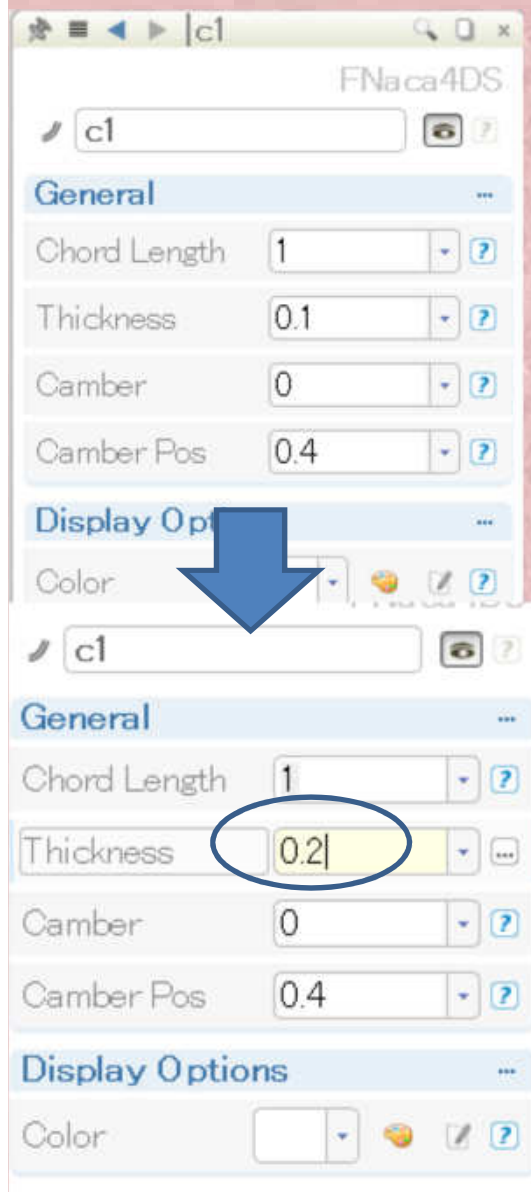


簡単な操作例:

- ① CADメニューのCurves, NACA 4DS カーブを選択
- ② パラメタ(デフォルト)と断面図が出てくる

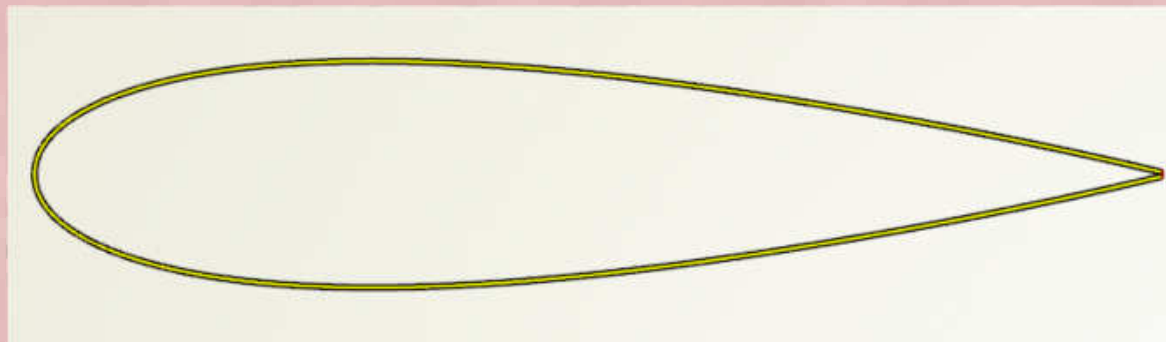
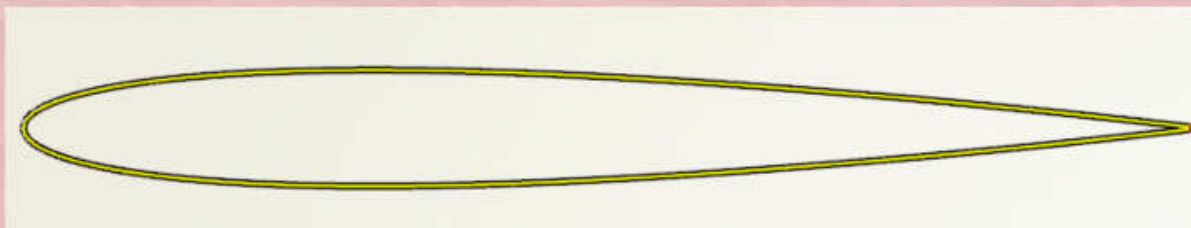


# FRIENDSHIP-Frameworkの無料ソフト CaeseFreeを試してみる④



簡単な操作例:

③ パラメタを変えてみる

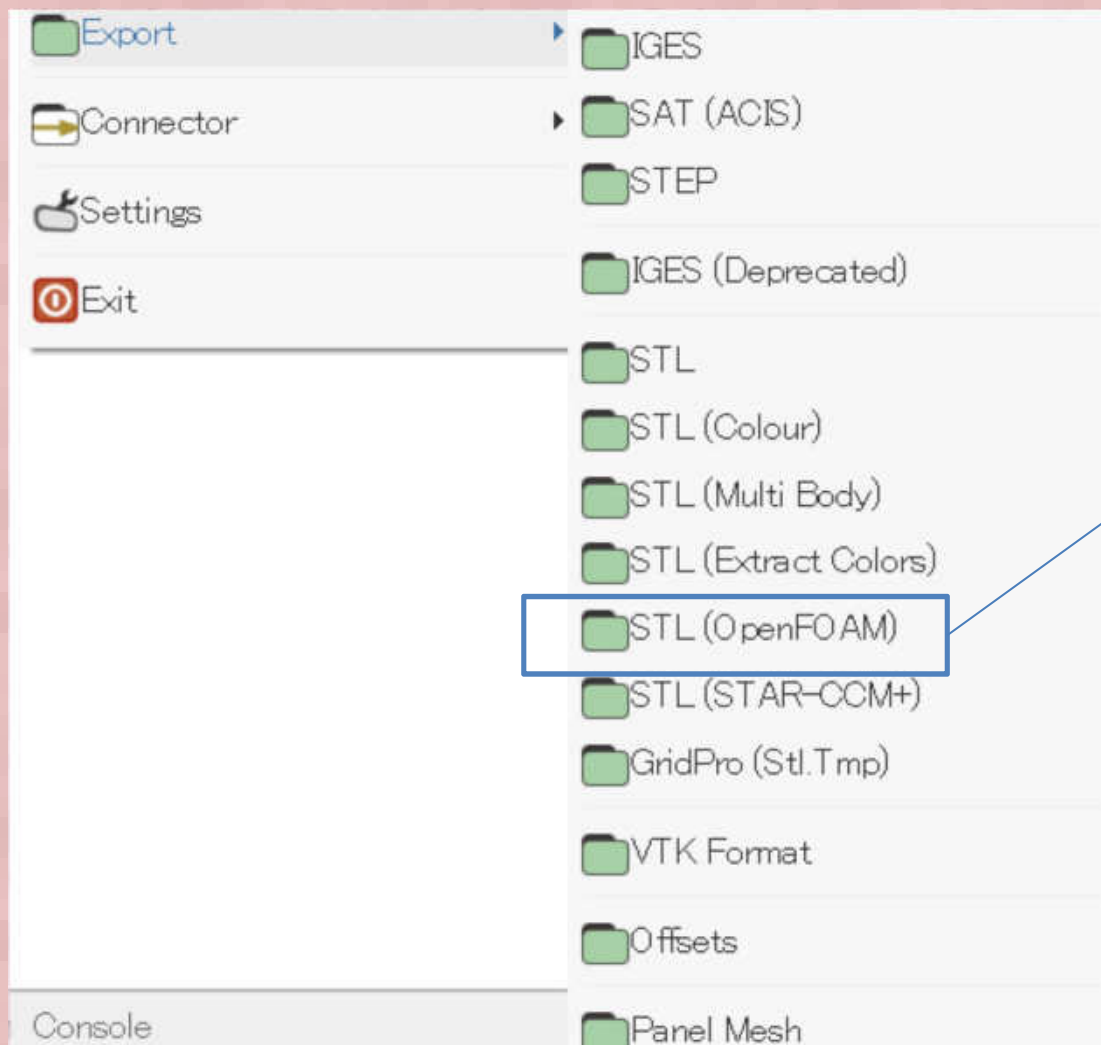


断面形状も勝手にかわる



# FRIENDSHIP-Frameworkの無料ソフト CaeseFreeを試してみる⑤

データは基本的にSTLで出力する模様



形状を作ったらこれで出力すればよいらしい？

すいません今回はこれで終了。  
ネットワークにつながないと動かないので、結構使い勝手が悪い。

# レベルセット法を用いた自由表面流が解けるオープンソースCFD REEF3D を試す - REEF3Dとは？ -

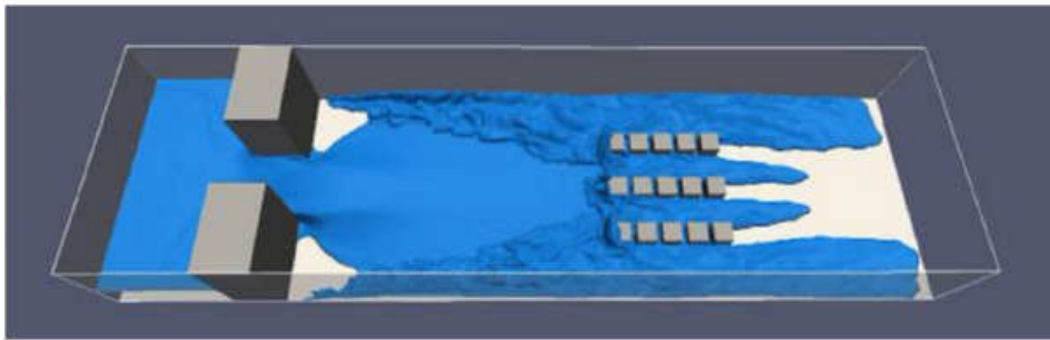
レベルセット法にもとづく自由表面流れ計算可能な流体計算ソフト

 **FN365**  
@FN365

[フォロー](#)

【オープンソースCFDソフト REEF3D】レベルセット法を採用し、自由表面流れ計算に強み。直交格子を使用し、境界の取り扱いはGCIBM。並列計算にも対応。補間にはWENOも使える。 [bit.ly/1ryhNza](https://bit.ly/1ryhNza)

← ↻ ★ ...



FN365 さん↓

[https://twitter.com/](https://twitter.com/fn365/status/550092886587768832)

[fn365/status/550092886587768832](https://twitter.com/fn365/status/550092886587768832)

1/10 のOpenCAE@岐阜勉強会で柴田先生から紹介があった。

# レベルセット法を用いた自由表面流が解けるオープンソースCFD REEF3D を試す

## -レベルセット法とVOF法-

<http://www3.u-toyama.ac.jp/okumura/fem/pdf/text/second-day/Tanaka2.pdf>

### 自由表面流れ

桜庭 雅明  
日本工営株式会社 中央研究所  
田中 聖三  
東京大学 地震研究所

サマースクール2012  
有限要素法による流れ解析の基礎と応用

### 自由表面流れとは？



越波を伴う流れ



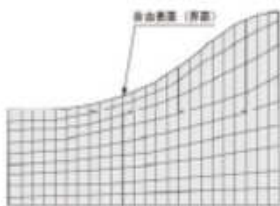
砕波を伴う流れ



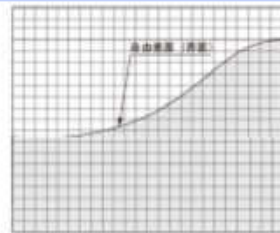
跳水を伴う流れ

- 自由表面流れ: 液体と気体の界面(Interface)が存在する流れ
- 代表的な挙動: 波動, 跳水, スロッシング等の流体の現象

### 自由表面流れ解析手法の分類



界面追跡法 (Interface-Tracking method)

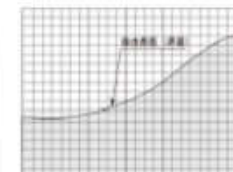
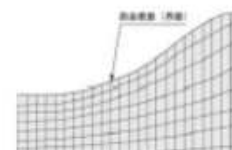


界面捕捉法 (Interface-Capturing method)

- 界面(Interface)の位置を直接的(界面追跡), 間接的(界面捕捉)により表現する
- 移動メッシュ(Lagrange記述), 固定メッシュ(Euler記述),

### 自由表面流れ解析手法の分類

名称	界面の表現方法	特徴	手法
界面追跡法	領域境界として直接的に表現	界面の位置が正確, 解析メッシュの破壊	ALE法 Space-Time法
界面捕捉法	識別子を用いて間接的に表現	不連続な現象に適用可能, 界面が不鮮明となる	VOF法 Level set法



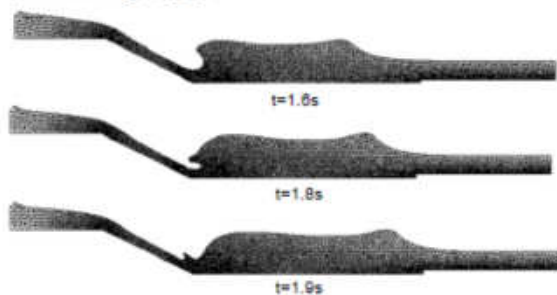
# レベルセット法を用いた自由表面流が解けるオープンソースCFD REEF3D を試す

## -レベルセット法とVOF法②-

<http://www3.u-toyama.ac.jp/okumura/fem/pdf/text/second-day/Tanaka2.pdf>

### 界面追跡法の応用

複雑な自由表面形状の流れへの適用にはメッシュの再構築 (Re-Meshing) が必要。



### 界面追跡法のまとめ

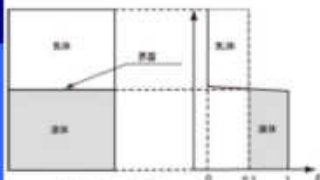
#### □ 界面追跡法の特徴

- メッシュによる移動境界 (自由表面) の表現
- メッシュの移動座標系により小分類される
- 直接的な表現のため、界面が明確に定義可能
- 複雑な自由表面形状に適用は難しい
- 大変形する問題に関しては、メッシュの再構築 (Re-Meshing) が必要

-Level set 関数使うかVOF関数を使うかの違い。

← 特徴は左下参照

### 界面捕捉法 VOF (Volume Of Fluid) 法



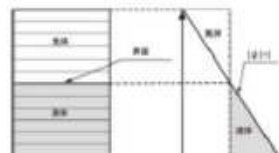
- 長所
- アルゴリズムが容易
- 短所
- 体積補正に工夫を要する
  - 界面付近の平滑化に工夫を要する

VOF関数の定義

VOF関数を用いた密度・粘性係数

$$\begin{cases} \rho = \rho_f \phi + \rho_g (1 - \phi) \\ \mu = \mu_f \phi + \mu_g (1 - \phi) \end{cases}$$

### 界面捕捉法 Level set 法



- 長所
- 界面の安定性・鋭敏性を保つ
  - 体積補正の導入が比較的容易
- 短所
- Level set関数の再初期化が必要

Level set関数の定義

Heaviside関数を用いた密度・粘性係数

$$\begin{cases} \rho = 0.5(\rho_f + \rho_g) + H_D(\rho_f - \rho_g) \\ \mu = 0.5(\mu_f + \mu_g) + H_D(\mu_f - \mu_g) \end{cases} \quad H_D, \text{ Heaviside関数}$$

# レベルセット法を用いた自由表面流が解けるオープンソースCFD REEF3D を試す

## - インストール方法等 -

- <https://reef3d.wordpress.com/source-code/>  
→ このリンク: Sourceforge からソースをDLする
- Mpic++ でmake する。
- Linux ubuntu 12.04LTS と  
Windowsでも cygwin64 でmake できた。  
(何も考えず、ソースを展開してmake)
- Mpic++ を使ってコンパイルする仕様(mpi.hをインクルード)にしているのでmpiライブラリが入っていないとコンパイルすらできない(cygwin64 でもmpi 関連ライブラリ導入が必須)。
- メッシュは専用メッシャーDiveMeshで作成する。こちらはg++でコンパイルする
- 実行はシングル実行の場合はコマンドを直接実行。並列の場合はmpirun でプロセッサ数を指定して実行



# チュートリアルDamBreakを試す

- OpenFOAM のInterFoam で有名なダムブレイクと同様のチュートリアルがあるので、これを実行してみる。

コマンド実行は左のソルバ、メッシュ向けのコントロールファイル2個をおいたディレクトリにてメッシュャー、ソルバの順番に実行する

```
% divemesh
```

```
% reef3d
```

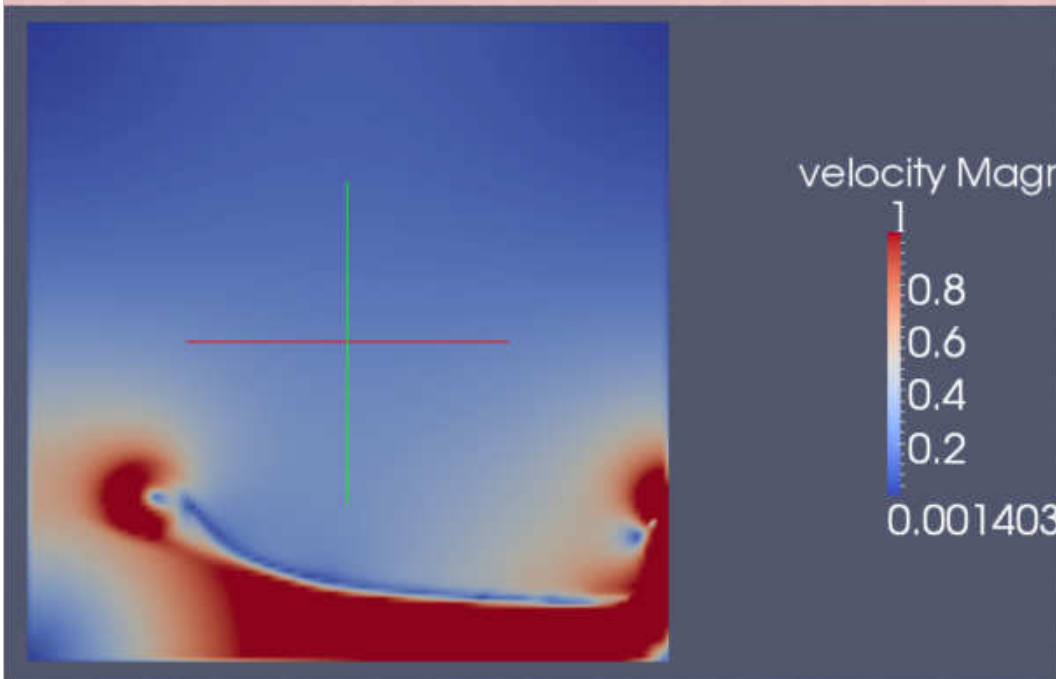
(この辺はOpenFOAM ライク仕様)

Mesher設定: control.txt

```
C 11 21 // left side: wall boundary
C 12 3 // side: symmetry plane
C 13 3 // side: symmetry plane
C 14 21 // right side: wall boundary
C 15 21 // bottom: wall boundary
C 16 21 // top: wall boundary
B 1 0.005 // mesh size dx
B 10 0.0 0.6 0.0 0.005 0.0 0.6 //
rectangular domain size domain size
M 10 4 // number of processors
```

Solver設定:ctrl.txt

```
B 5 1 // rough wall law
B 50 0.0001 // wall roughness ks
D 10 4 // Conservative WENO discretization for velocities
D 20 2 // implicit diffusion treatment
D 30 1 // projection method for the pressure
F 30 3 // 3rd-order Runge-Kutta Scheme for Level Set Time Treatment
F 40 3 // 3rd-order Runge-Kutta Scheme for Reinitialization Time Treatment
F 50 4 // Level set function is not fixed at inlet or outlet
F 54 0.15 // x-coordinate for end fluid phase one
F 56 0.3 // z-coordinate for end fluid phase one
N 10 3 // BiCGStab solver with Jacobi scaling
N 40 1 // 2nd-order Adams-Bashforth time treatment for velocities
N 45 25000 // maximum number of iterations
N 47 0.1 // factor for CFL criterion
M 10 4 // number of parallel processes
P 10 1 // turn on .vtu print out
P 30 0.01 // print out interval based on simulation time
W 22 -9.81 // gravity
```



Tutorial: DamBreak の実行結果

# Reef3D のまとめ

- レベルセット法の自由界面の計算が可能な reef3D を試した
  - mpicc があれば問題なくコンパイル可能
  - コードがコンパクトでカスタマイズも簡単ぽい?
  - 計算設定やメッシャー設定が割と簡単  
(チュートリアル of 全行にコメントがあるのは良い)
  - 結果は VTU フォーマットで出力 ParaView で可視化
  - そこそこ複雑な乱流モデルなども含まれるようで、簡単な自由界面の計算には良いかも？

# OpenFOAMで音響解析できるか調べてみた①

- OpenAcoustics について
  - 先日富山のOpenCAE勉強会にてOpenFOAMにて音解析ができるか？どうかという話題が上がったが、音響関連のHPにてOpenFOAMでできる？という記述で見た気がしたので、再度調べてみた。
- 音響関連のHPとはOpenAcousticsのHPでした。 <http://www.openacoustics.org/>
- 音響数値解析のオープンソース(Pythonベース)を公開しており、その中に既存のオープンソースの音響解析機能を調査したページがある。Elmer, CodeAsterなどを調べており、調査対象にOpenFOAM も入っている



# OpenFOAMで音響解析できるか調べてみた②

OpenAcoustics の他ツールのサーベイ結果

<http://www.openacoustics.org/> から引用

音響教育向けの音響解析専用オープンソースコードとしては、以下のものがあります。

- [Lambda](#), Institut für H'ortechnik & Audiologie Germany, 2D-TLM (Transmission Line Matrix) Method ([the 19th ICA Paper](#))

音響解析が可能なプロダクションレベルの主要オープンソースマルチフィジックスコードとしては、以下のものがあります。

- [Elmer](#), CSC Finland, FEM Helmholtz + BEM Helmholtz (non-FMM-accelerated)、[サーベイ結果](#)
- [OpenFOAM](#), OpenCFD Ltd., United Kingdom, FVTD (requires customization)、[サーベイ結果](#)
- [Code Aster](#)、フランス電力公社、FEM、[サーベイ結果](#)
- [ESP-r](#), University of Strathclyde United Kingdom, 残響式等の基本的な計算式に基づいて計算、[サーベイ結果](#)

## [Code Aster](#)

### 概要

- 開発機関: EDF (フランス電力公社)、フランス
- 解析手法: FEM
- ライセンス: GPL
- 参考: 理論マニュアルのセクションR4 (R4 : Méthodes d'analyse) 中、[R4.02](#)が音響解析関連。利用可能な有限要素の[一覧](#)。サンプルケースは[AHLV100](#)、[AHLV101](#)。

## 1.5. Elmerによる音響問題解析のサンプル

### (1) [散乱](#)

- Helmholtz方程式
- 音源は正弦振動境界
- 散乱体表面は反射境界、外周は吸音境界

### (2) [ドップラー効果](#)

- Helmholtz方程式
- 音源は正弦振動要素
- 媒質が一定速度で一様に流れる

### (3) [振動—音響連成問題](#)

- 無限大平板上の正弦振動する弾性梁(固定端—自由端)からの音響放射
- 線形弾性方程式とHelmholtz方程式の連成

-音響解析機能を持つオープンソースとしてLambda(知らない?), Elmer, CodeAster, OpenFOAMなどを調査

-Elmer, CodeAster ではサンプル事例の実行結果まで詳しく解説している

# OpenFOAMで音響解析できるか調べてみた③

OpenFOAM survey 結果について

<http://www.openacoustics.org/> から引用↓

## OpenFOAM

### 概要

-開発機関: OpenCFD Ltd.、英国

-解析手法: FVM (FVTD)

-ライセンス: GPL  
OpenFOAMでは、OpenAcoustics開発者の一人が提供するカスタマイズソルバによって、時間領域音響解析が可能です。potentialWaveFoamというそのソルバと、テストケースをFOACプロジェクトのSubversionレポジトリからダウンロード可能です。

### svn co

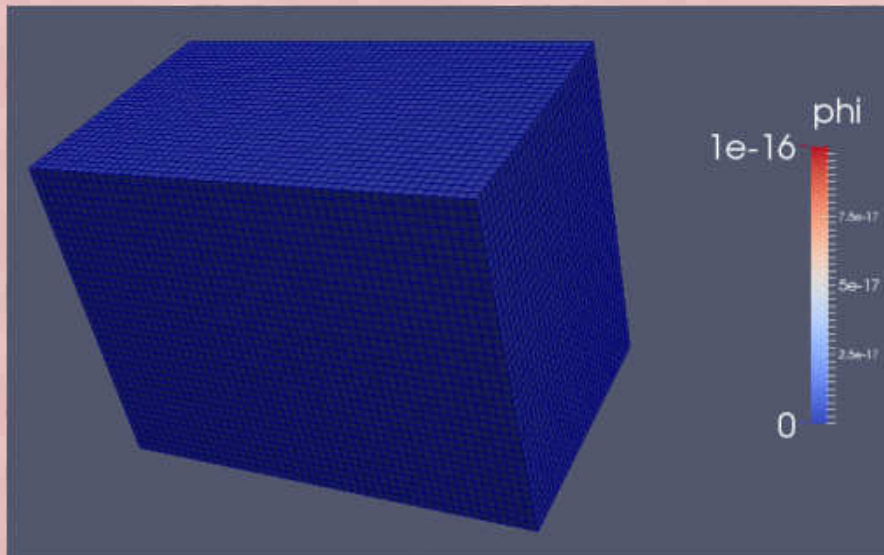
<https://foac.svn.sourceforge.net/svnroot/foac/trunk/survey/OpenFOAM/potentialWaveFoam>

↑ 上記リンクは古く、別のレポジトリへ行くように指示されるが、そちらを指定すれば特に問題なく、ソースをダウンロードできる(詳細次ページ)。

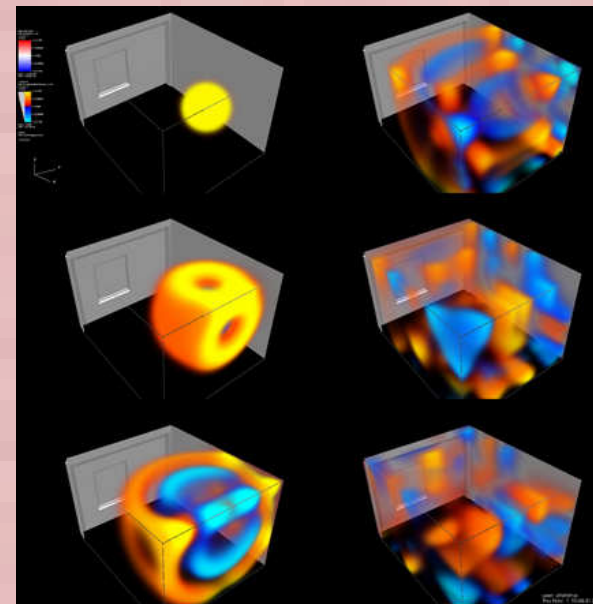
- OpenFOAM の環境設定された端末で“Allrun”を実行するとMake からサンプルファイルの実行まで実行されるようになっているらしい？

# OpenFOAMで音響解析できるか調べてみた④

- ・Allrunを実行してみたが、OpenFOAM V1.5 とかなり古いバージョンをベースに開発されているようで、OpenFOAM2.3.0 で実行してみたが、そのままではエラーになる。
- ・ソルバ部分のpotentialWaveFoam自体はwmake すると特に問題なく完了するので、手動にてSample のチュートリアルファイルを実行すると計算自体は特に問題なく完了するようであるが、何も結果が出力されておらず、やはり何か問題ありそう



計算結果？



本当はこんな感じになるらしい？

現在のレポジトリ

<https://svn.code.sf.net/p/foac/code/trunk/survey/OpenFOAM/potentialWaveFoam>