

# CFD の理論と ANSYS 利用法の基本

青木 宣明

## 目次

1	はじめに.....	3
2	なぜ数値計算をするのか? .....	4
3	CFD の理論の基礎.....	5
3.1	保存式.....	5
3.1.1	質量保存式 (連続の式) .....	5
3.1.2	運動量保存式 (運動方程式) .....	6
3.1.3	エネルギー保存式.....	6
3.1.4	保存式の一般形.....	7
3.2	有限体積法.....	8
3.2.1	保存式の離散化.....	8
3.2.2	スタッガード格子.....	10
3.3	CV 界面での輸送量の補間方法.....	11
3.3.1	定常 1 次元の対流と拡散.....	11
3.3.2	風上差分法.....	13
3.3.3	2 次精度中心差分法.....	14
3.4	SIMPLE 法 (圧力・速度のカップリング) .....	14
3.4.1	運動方程式.....	15
3.4.2	圧力・速度の補正.....	15
3.4.3	SIMPLE のアルゴリズム.....	16
3.5	収束計算の手法.....	17
3.5.1	緩和係数.....	17
3.5.2	離散化スキームの選択.....	19
3.5.3	収束の判定.....	19
3.6	メッシュ生成.....	19
3.7	上手くいかなかったシミュレーションの活用報.....	21
4	ANSYS®の利用法.....	22
4.1	ANSYS FLUENT の Tutorial.....	22
4.2	反応が含まれる Case Study.....	22
4.2.1	計算領域の描画.....	23
4.2.2	計算領域のメッシュへの離散化.....	25
4.2.3	計算領域の境界条件の種類の指定.....	28
4.2.4	Meshing からのファイルのエクスポートと FLUENT での読み込み.....	28
4.2.5	流れ・物質移動のモデルと物質の物性・境界条件の指定.....	29

4.2.6	離散化スキームと収束計算のモニター法の選択 .....	31
4.2.7	結果の整理 .....	32
4.2.8	Mesh Adaption .....	37
4.2.9	ユーザー定義関数の導入 .....	38
4.3	非定常計算の Case Study .....	38
4.3.1	非定常計算の設定 .....	38
4.3.2	計算結果の整理 .....	40
4.3.3	初期条件の与え方の応用編 .....	41
4.3.4	非定常計算のコツ .....	42
4.4	乱流の Case Study .....	43
4.4.1	Case Study の設定 .....	43
4.4.2	各乱流モデルによる結果 .....	44
4.5	熱交換器のシミュレーションと周期境界条件の設定 .....	44
4.5.1	問題設定 .....	44
4.5.2	形状の描画とメッシュ作成 .....	45
4.5.3	メッシュ作成と境界条件の種類の設定 .....	47
4.5.4	周期境界の設定方法 .....	49
4.5.5	モデル・物性・境界条件・収束計算の設定と結果 .....	50
4.6	DesignModeler・Meshing の使い方の補足 .....	51
4.6.1	形状から形状を切り取るまたは形状を融合して新たな複雑な形状にする方法 ...	51
4.6.2	スケッチの各部位のサイズを指定する方法 .....	52
4.6.3	点の座標を手動で指定してから形状描画を進める方法 .....	53
4.6.4	複雑な形状でのメッシュの切り方 .....	54
5	おわりに .....	57
	参考文献 .....	57
	付録 .....	58
A	実習例に対応した CFD Case ファイル一覧 .....	58
B	ANSYS 15.0 のインストール方法 .....	58
C	Journal File .....	63
C.1	Line を引くための Journal .....	63
C.2	FLUENT の設定を行う Journal File .....	64
C.3	ファイルの読み込み・緩和係数の変更・収束計算・セーブを行う Journal File ...	69
D	ユーザー定義関数 (UDF) .....	71
D.1	速度分布を設定する UDF .....	71
D.2	物性を定義する UDF .....	72
D.3	結果整理の (標準偏差) UDF .....	72
D.4	Langmuir 型吸着を考慮した反応速度式の UDF .....	73
E	非等温反応系の計算に必要な設定 .....	75
F	スラグ流再現のための設定方法 .....	76

## 1 はじめに

本書では、基本的に化学工学、とくに反応工学に応用できるように Case Study を示しながら数値流体力学 (Computational Fluid Dynamics, CFD) の理論とソフトウェアの ANSYS® (FLUENT®はその中の CFD 計算を実施するためのソフト) の使い方を説明していく。なお、本書は ANSYS® 15.0 に準拠している。CFD の理論に関する書籍は多数あるし、FLUENT®の利用法もマニュアルをよく読めば使いこなせる。しかし、CFD でできることは多様であり、各自が扱う事例は限定的なものである。化学工学では流れと反応を伴う流れを扱うことが多い。そこで、その限定された部分を修得できるテキストがあれば教える側も教わる側も便利である。本研究室独自の CFD ソフト利用法に関する知識の蓄積にもつながる点でも、確立した手法をできるだけ速く身につけてオリジナルな仕事にスムーズに入っていけるようする点でも、利用法を文書にまとめておくことは有意義であると私は考えている。

CFD の実務に関わっていてさらに学習を深めたい方にお勧めの書籍を先に紹介しておく。両方とも Kindle 電子書籍で Kindle for PC (<https://www.amazon.co.jp/dp/B011UEHYWQ/>でダウンロード) があればパソコンでも読める。CFD ソフトウェアのサポートに携わってきた方が書かれているので実務家が CFD の理論と実務の理解を深めるのに非常に役立つと筆者は考えている。

- 小林治樹；CFD ツールによるモデル化の基本，アシスト CFD 技術事務所 (2013)  
<https://www.amazon.co.jp/dp/B00HKQX1I0/>
- 小林治樹；CFD エンジニアのためのやさしい流体力学 (2013)  
<https://www.amazon.co.jp/dp/B00AP5PY2E/>

なお、学生さんの場合は無料で FLUENT®を使える ANSYS Student がある。  
<http://www.ansys.com/ja-JP/products/academic/ansys-student>  
からダウンロードできる。ただし 12 ヶ月の期間限定で、CFD のメッシュ数の上限は 512,000、サポートは専用の Web サイト上だけという制限はある。

まず、CFD シミュレーションの本題に入る前に、なぜ数値計算をするのかという疑問がわくと思う。それに答えるところから本稿をはじめていく。

※注意 本書はもともと研究室内向けのテキストであり、オンラインで公開するにあたり著作権侵害になる可能性のある ANSYS のユーザーインターフェース (UI) を含むキャプチャー画面を削除した。わかりにくくなっている部分があることを先に述べておく。

## 2 なぜ数値計算をするのか？

まず、数値計算が必要になるかという第一の理由として、流体力学の講義でも出てきた Navier–Stokes の式のような偏微分方程式は解析的に解くことがまずできないことが挙げられる。実際研究で扱う対象は、教科書の例題よりも複雑だから、対象とする現象の収支式はなおさらややこしいもので、解析的には解けない。

二つめの理由として、実験ではできないことが数値計算でできるからということも挙げられる。実験では様々な外乱が入るため、対象とする現象に効いてくる因子は何で、それがどの程度、観測している量に影響してくるかを特定することが困難なことがある。数値計算ならば、ある特定の因子だけを変えてこの変化による傾向を抽出することもできる。さらに、あまりに非現実的な設定は考えものだが、通常実験する範囲を超えて条件を振って、傾向を外挿することもできる。過酷な条件での実験が困難な場合はこのようにできることは便利である。また、反応器などの装置サイズと反応器の性能の関係を明らかにしたいときに、一部だけサイズを変えて反応器を作り直して、実験を繰り返し替えるということをするのは、時間も手間もお金もかかる。数値計算では条件の変更が容易であることが多く、計算が走り始めれば放置しても危険はないので、その間ほかのことをすることもできる。このため、作業負荷も低くなりやすく、融通もききやすい。では、数値計算はお金がかからないかということとは限らない。ANSYS®をはじめとする市販の CFD ソフトウェアもライセンス契約で使うものなので、使うのにお金がかかる。自前でプログラムを書けばお金はかからないが、プログラムを書いて、そのバグを取るのに精一杯になってしまう。研究の本当の目的は、プログラムを書くことではないはずだから、これでは目的は果たせなくなってしまう。市販の CFD ソフトウェアを用いるのも、本来の目的に集中するための投資ともいえる。

ただし、数値計算にも弱点はある。特定の影響を抽出するのに便利と述べたが、これは広い範囲を一度に扱うには向かないということでもある。また、数値計算で扱うのはあくまでモデルであり、現象を正しく記述できるモデルのないものに対しては、正確な結果を期待できない。たとえば、層流から乱流への遷移を扱うことは、理論でも不明なところが多く、CFD シミュレーションではなおのこと扱いは不可能に近い。

上記にもあるように、数値計算をするには対象とする現象のモデル化が必要となる。CFD シミュレーションでは、主に物質・運動量・熱の流れの現象を扱うため、必要になる流れのモデル化の方法を次節では説明する。

### 3 CFD の理論の基礎

本節では、本論文中に出てきた CFD シミュレーションに関する最低限の離散化・収束計算理論・方法についてふれる。まず、質量・運動量・エネルギー保存則の一般形を導出し、CFD シミュレーションの離散化手法の一つである有限体積法（Control Volume 法，CV 法）を紹介する。次に、計算領域を有限体積（CV）に分割したときに CV 境界面での輸送物性・対流項の値の補間するための方法について説明する。さらに、圧力と運動量の方程式の収束計算を安定に行うための SIMPLE 法について述べる。収束計算を安定かつ迅速に収束させるために重要な不足緩和，収束判定法，計算領域の離散化の方法と収束計算への影響についても触れる。詳細な理論については文献[1-4] や ANSYS Customer Portal からダウンロードできる Theory Guide を参照されたい。

#### 3.1 保存式

CFD において流動・伝熱などの問題を解くためには、物質・運動量・エネルギーの保存式を解く必要がある。ここではこれらの保存式を示し、保存式の一般形を導く。

##### 3.1.1 質量保存式（連続の式）

ここでは 1 成分系を考える。体積  $\Delta x \Delta y \Delta z$  の微小な六面体について物質収支をとると、

$$\begin{aligned} (\text{物質の増加速度}) &= (\text{物質の流入速度}) \\ &\quad - (\text{物質の流出速度}) + (\text{区間 } \Delta x \text{ での生成速度}) \end{aligned} \quad (1)$$

であるから、

$$\begin{aligned} \Delta x \Delta y \Delta z \frac{\partial \rho}{\partial t} &= \Delta y \Delta z \left( (\rho v_x)|_x - (\rho v_x)|_{x+\Delta x} \right) \\ &\quad + \Delta z \Delta x \left( (\rho v_y)|_y - (\rho v_y)|_{y+\Delta y} \right) \\ &\quad + \Delta x \Delta y \left( (\rho v_z)|_z - (\rho v_z)|_{z+\Delta z} \right) \\ &\quad + S_m \Delta x \Delta y \Delta z \end{aligned} \quad (2)$$

と表わせる。ここで、 $\rho$  は流体の密度、 $v$  は速度、 $S_m$  は単位体積あたりの生成速度を示す。式 (2) の両辺を  $\Delta x \Delta y \Delta z$  で割って  $\Delta x, \Delta y, \Delta z \rightarrow 0$  の極限をとれば

$$\frac{\partial \rho}{\partial t} = - \left\{ \frac{\partial}{\partial x} (\rho v_x) + \frac{\partial}{\partial y} (\rho v_y) + \frac{\partial}{\partial z} (\rho v_z) \right\} + S_m \quad (3)$$

が得られる。反応が起こらなければ、最後の生成項は 0 である。また、多成分系では拡散も考慮する必要がある。

### 3.1.2 運動量保存式（運動方程式）

体積  $\Delta x \Delta y \Delta z$  の微小な六面体における  $x$  方向の運動量収支は

$$\begin{aligned} (\text{運動量の増加速度}) = & (\text{運動量の流入速度}) - (\text{運動量の流出速度}) \\ & + (\text{外部からこの六面体に働く力の総和}) \end{aligned} \quad (4)$$

であるから、

$$\begin{aligned} \frac{\partial}{\partial t}(\rho \Delta x \Delta y \Delta z \cdot v_x) = & \\ \text{対流による運動量輸送} & \begin{cases} \Delta y \Delta z \left( (\rho v_x \cdot v_x)|_x - (\rho v_x \cdot v_x)|_{x+\Delta x} \right) \\ + \Delta z \Delta x \left( (\rho v_y \cdot v_x)|_y - (\rho v_y \cdot v_x)|_{y+\Delta y} \right) \\ + \Delta x \Delta y \left( (\rho v_z \cdot v_x)|_z - (\rho v_z \cdot v_x)|_{z+\Delta z} \right) \end{cases} \\ \text{分子運動による運動量輸送} & \begin{cases} + \Delta y \Delta z \left( \tau_{xx}|_x - \tau_{xx}|_{x+\Delta x} \right) \\ + \Delta z \Delta x \left( \tau_{yx}|_y - \tau_{yx}|_{y+\Delta y} \right) \\ + \Delta x \Delta y \left( \tau_{zx}|_z - \tau_{zx}|_{z+\Delta z} \right) \end{cases} \\ \text{圧力・重力・他の外力} & + (p|_x - p|_{x+\Delta x}) \cdot \Delta y \Delta z + \rho \Delta x \Delta y \Delta z \cdot g_x + F_x \Delta x \Delta y \Delta z \end{aligned} \quad (5)$$

この式で  $\tau$  は運動量流束,  $p$  は圧力,  $g_x$  は  $x$  方向の重力加速度,  $F_x$  はその他の体積力を表す。上式の両辺を  $\Delta x \Delta y \Delta z$  で割って  $\Delta x, \Delta y, \Delta z \rightarrow 0$  の極限をとれば

$$\begin{aligned} \frac{\partial}{\partial t}(\rho v_x) = & - \left( \frac{\partial}{\partial x} \rho v_x v_x + \frac{\partial}{\partial y} \rho v_y v_x + \frac{\partial}{\partial z} \rho v_z v_x \right) \\ & - \left( \frac{\partial}{\partial x} \tau_{xx} + \frac{\partial}{\partial y} \tau_{yx} + \frac{\partial}{\partial z} \tau_{zx} \right) \\ & - \frac{\partial p}{\partial x} + \rho g_x + F_x \end{aligned} \quad (6)$$

が得られる。本研究では重力の影響は十分小さいものとし、その他の体積力もないので圧力の項までを計算することになる。なお、同様の手順で  $y$  方向・ $z$  方向の運動方程式が導かれる。

### 3.1.3 エネルギー保存式

前節と同様に、体積  $\Delta x \Delta y \Delta z$  の微小な六面体についてエネルギー収支をとる。

$$\begin{aligned} & (\text{内部エネルギー} + \text{運動エネルギーの増加速度}) \\ & = (\text{内部エネルギー} + \text{運動エネルギーの流入速度}) \\ & - (\text{内部エネルギー} + \text{運動エネルギーの流出速度}) \end{aligned}$$

+ (外部からこの六面体に対してする仕事量) (7)

外部からの仕事には、重力・圧力・粘性散逸・化学種の拡散・反応熱などが考えられるが、ここでは圧力による仕事だけを考慮してエネルギー保存式を導く。

$$\frac{\partial}{\partial t} \left( \rho E + \frac{1}{2} \rho v^2 \right) \Delta x \Delta y \Delta z =$$

$$\begin{aligned} \text{対流によるエネルギーの出入} & \left\{ \begin{aligned} & \Delta y \Delta z \left( [v_x (\rho E + \frac{1}{2} \rho v^2)] \Big|_x - [v_x (\rho E + \frac{1}{2} \rho v^2)] \Big|_{x+\Delta x} \right) \\ & + \Delta z \Delta x \left( [v_y (\rho E + \frac{1}{2} \rho v^2)] \Big|_y - [v_y (\rho E + \frac{1}{2} \rho v^2)] \Big|_{y+\Delta y} \right) \\ & + \Delta x \Delta y \left( [v_z (\rho E + \frac{1}{2} \rho v^2)] \Big|_z - [v_z (\rho E + \frac{1}{2} \rho v^2)] \Big|_{z+\Delta z} \right) \end{aligned} \right. \\ \text{熱伝導によるエネルギーの出入} & \left\{ \begin{aligned} & + \Delta y \Delta z (q_x \Big|_x - q_x \Big|_{x+\Delta x}) \\ & + \Delta z \Delta x (q_y \Big|_y - q_y \Big|_{y+\Delta y}) \\ & + \Delta x \Delta y (q_z \Big|_z - q_z \Big|_{z+\Delta z}) \end{aligned} \right. \\ \text{圧力による仕事} & \left\{ \begin{aligned} & + ((pv_x) \Big|_x - (pv_x) \Big|_{x+\Delta x}) \cdot \Delta y \Delta z \\ & + ((pv_y) \Big|_y - (pv_y) \Big|_{y+\Delta y}) \cdot \Delta z \Delta x \\ & + ((pv_z) \Big|_z - (pv_z) \Big|_{z+\Delta z}) \cdot \Delta x \Delta y \end{aligned} \right. \end{aligned} \quad (8)$$

この式で  $E$  は単位質量あたりの内部エネルギー、 $q$  は熱流束である。上式の両辺を  $\Delta x \Delta y \Delta z$  で割って  $\Delta x, \Delta y, \Delta z \rightarrow 0$  の極限をとれば

$$\begin{aligned} \frac{\partial}{\partial t} \left( \rho E + \frac{1}{2} \rho v^2 \right) = & - \left\{ \frac{\partial}{\partial x} \left( v_x (\rho E + \frac{1}{2} \rho v^2) \right) \right. \\ & + \frac{\partial}{\partial y} \left( v_y (\rho E + \frac{1}{2} \rho v^2) \right) + \frac{\partial}{\partial z} \left( v_z (\rho E + \frac{1}{2} \rho v^2) \right) \left. \right\} \\ & - \left( \frac{\partial}{\partial x} q_x + \frac{\partial}{\partial y} q_y + \frac{\partial}{\partial z} q_z \right) \\ & - \left\{ \frac{\partial}{\partial x} (pv_x) + \frac{\partial}{\partial y} (pv_y) + \frac{\partial}{\partial z} (pv_z) \right\} \end{aligned} \quad (9)$$

となる。

### 3.1.4 保存式の一般形

これまで個々の保存式を示してきたが、これらの式は、

$$\text{(非定常項)} = \text{(対流項)} + \text{(拡散項)} + \text{(生成項)} \quad (10)$$

の形をしていることがわかる。そこで従属変数を  $\phi$  として、対流項を左辺に移項して一般的な保存式を示すと次式のようなになる。

$$\begin{aligned} \frac{\partial}{\partial t}(\rho\phi) + \frac{\partial}{\partial x}(\rho v_x\phi) + \frac{\partial}{\partial y}(\rho v_y\phi) + \frac{\partial}{\partial z}(\rho v_z\phi) \\ = \frac{\partial}{\partial x}\left(\Gamma\frac{\partial\phi}{\partial x}\right) + \frac{\partial}{\partial y}\left(\Gamma\frac{\partial\phi}{\partial y}\right) + \frac{\partial}{\partial z}\left(\Gamma\frac{\partial\phi}{\partial z}\right) + S_m \end{aligned} \quad (11)$$

つまり、

$$\frac{\partial}{\partial t}(\rho\phi) + \text{div}(\rho\mathbf{v}\phi) = \text{div}(\Gamma \text{grad}\phi) + S_m \quad (12)$$

と表される。ここで、 $\Gamma$  は交換係数とよばれ、運動量なら粘度、質量なら拡散係数、エネルギーなら熱伝導率に相当する。

## 3.2 有限体積法

CFD シミュレーションでは質量・運動量・エネルギーの保存式の離散化が必要となる。FLUENT では、この離散化を有限体積法によって行っている。有限要素法とは、図 1 に示すように、計算領域を Control Volume (CV) に分割し、CV ごとに質量・運動量・エネルギーの保存式を積分することで得られた連立代数方程式を解くという方法である。物理量は CV の中心で定義され、この値がメッシュの平均値となる。また、境界値は隣接する CV 中心の値から求める。ここでは保存則の微分方程式の離散化の方法と、CV 法の特徴でもある離散化格子の区切り方のスタッガード格子について説明する。

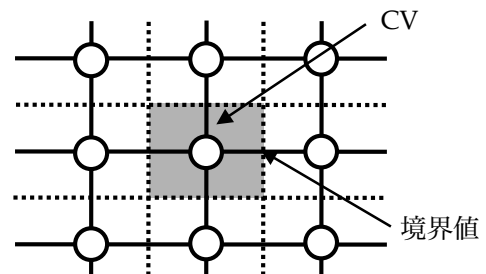


図1 有限体積法

### 3.2.1 保存式の離散化

まずは保存式の離散化の方法について、次式で表現される定常 1 次元熱伝導を例に用いて説明していく。

$$\frac{d}{dx}\left(k\frac{dT}{dx}\right) + S = 0 \quad (13)$$

ここで、 $k$  は熱伝導率、 $T$  は温度、 $S$  は単位体積・単位時間あたりの発熱速度である。この微分方程式に対する離散化方程式を導くために、図 2 に示すような 間隔  $\Delta x$  で等間隔に並んだ格子点群を用いることにする。



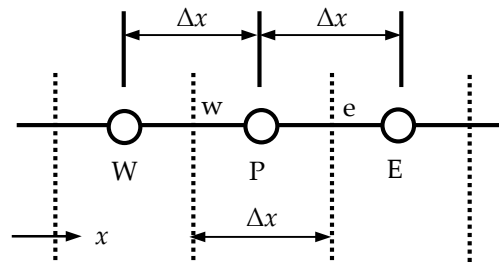


図2 1次元問題における格子点

まず、格子点 E と W に挟まれたある格子点 P に着目する。破線が CV の界面を示し、文字  $w \cdot e$  はこれらの界面を示す。この CV について式(13)を積分すると、

$$\left(k \frac{dT}{dx}\right)_e - \left(k \frac{dT}{dx}\right)_w + \int_w^e S dx = 0 \quad (14)$$

が得られる。この式の  $dT/dx$  を差分近似し、 $S$  はコントロールボリューム内において平均値  $\bar{S}$  で一定とすると、次のような離散化方程式が得られる。

$$k_e \frac{T_E - T_P}{\Delta x} - k_w \frac{T_P - T_W}{\Delta x} + \bar{S} \Delta x = 0 \quad (15)$$

$k_e \cdot k_w$  は CV 界面の熱伝導率である。さらに、この離散化方程式を次式のように表すと便利であり、この形が離散化方程式の標準形である。

$$a_P T_P = a_E T_E + a_W T_W + b \quad (16)$$

ここで、

$$a_E = \frac{k_e}{\Delta x}, \quad a_W = \frac{k_w}{\Delta x}, \quad a_P = a_E + a_W, \quad b = \bar{S} \Delta x \quad (17)$$

である。

この離散化において、式(15)からわかるように、界面の熱伝導率が必要になる。熱伝導率は各 CV の中心で定義されるので、界面での値をどう求めるかを決めなくてはならない。この方法としてまず考えられるのが、次式のような線形補間である。

$$k_e = \frac{k_E + k_P}{2} \quad (18)$$

もう一つは多層壁の熱伝導から（輸送現象 p. 114 を参照[5]），

$$q_e = \frac{T_P - T_E}{\Delta x / k_P + \Delta x / k_E} = \frac{k_e (T_P - T_E)}{\Delta x} \quad (19)$$

となるので,

$$k_e = \frac{1}{1/k_P + 1/k_E} \quad (20)$$

のように求める方法である.

### 3.2.2 スタッガード格子

一般にスカラー変数  $\varphi$  の保存則を離散化する際に, 速度は CV の界面で定義されていると都合がよい. つまり, 速度成分に対しては「ずらして配列された (スタッガード) 格子」を用いるのである.

まずは, スタッガード格子を用いる理由を定常 1 次元の定密度場を用いて説明する. この場合, 連続の式は,

$$du/dx = 0 \quad (21)$$

のようになる. 図 2 で表された CV についてこの式を積分すると,

$$u_e - u_w = 0 \quad (22)$$

となる. さらに, 界面における速度を線形補間から求めると,

$$\frac{u_P + u_E}{2} - \frac{u_W + u_P}{2} = \frac{u_E - u_W}{2} = 0 \quad (23)$$

となり, 離散化した連続の式は, 隣接した格子点ではなく一つおきの格子点で速度が等しくなればよいことを示している. このため, 図 3 で示されるような非現実的な速度場でも式(22)を満足することになる. しかし, 合理的で意味のある解とはいえない.

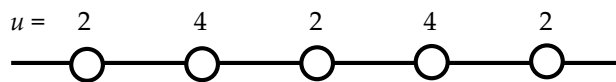


図 3 波状速度場

このような非現実的な解の導出を避ける解決策がスタッガード格子を採用することなのである. スタッガード格子では, 図 4 の矢印で示すように, 速度成分は CV の界面で定義され, 速度に対する CV は灰色になっている部分である.

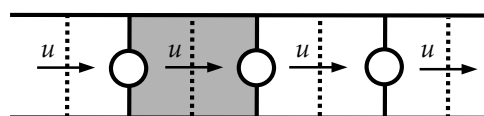


図 4 スタッガード格子

スタaggerド格子を用いる重要な利点は、標準的な CV (図 1 の灰色の部分) に対して離散化された連続の式は、隣接した格子点の速度成分の差を含むようになり、図 3 のような波状速度場が連続の式を満たすのを防いでくれる点にある。

### 3.3 CV 界面での輸送量の補間方法

前節の最初でも述べたように、有限体積法では各メッシュにおける変数の値はメッシュ中心で定義され、この値をメッシュの平均値とみなす。こうしたとき、隣接するメッシュの境界において、対流によって運ばれる量  $\varphi$  をどのように算出するかが問題になる。そこで、ここでは単に対流と拡散が存在する定常次元の場を用いて、その手法について説明する。ここでは FLUENT でも用意されている 1 次精度風上差分法、2 次精度中心差分法について扱う。

#### 3.3.1 定常 1 次元の対流と拡散

まず、質量・運動量・エネルギーの保存式の一般形について考える。保存式の式は、

$$(\text{非定常項}) = (\text{対流項}) + (\text{拡散項}) + (\text{生成項}) \quad (24)$$

の形をしている。そこで保存される量を  $\varphi$ 、生成項を  $S_m$  として、対流項を左辺に移項して一般的な保存式を示すと次式のようなになる。

$$\begin{aligned} \frac{\partial}{\partial t}(\rho\phi) + \frac{\partial}{\partial x}(\rho v_x\phi) + \frac{\partial}{\partial y}(\rho v_y\phi) + \frac{\partial}{\partial z}(\rho v_z\phi) \\ = \frac{\partial}{\partial x}\left(\Gamma \frac{\partial\phi}{\partial x}\right) + \frac{\partial}{\partial y}\left(\Gamma \frac{\partial\phi}{\partial y}\right) + \frac{\partial}{\partial z}\left(\Gamma \frac{\partial\phi}{\partial z}\right) + S_m \end{aligned} \quad (25)$$

つまり、

$$\frac{\partial}{\partial t}(\rho\phi) + \text{div}(\rho v\phi) = \text{div}(\Gamma \text{grad}\phi) + S_m \quad (26)$$

と表される。ここで、 $\Gamma$  は交換係数とよばれ、運動量なら粘度、質量なら拡散係数、エネルギーなら熱伝導率に相当する。式(26)から、単に対流と拡散が存在する定常次元の場での保存式は次式で表される。

$$\frac{d}{dx}(\rho u\phi) = \frac{d}{dx}\left(\Gamma \frac{d\phi}{dx}\right) \quad (27)$$

ここで  $u$  は  $x$  方向の速度、 $\Gamma$  は交換係数を表す。また、連続の式は、

$$\frac{d}{dx}(\rho u) = 0 \quad (28)$$

である。式(27)を図2で示す CV について積分すると、

$$(\rho u \phi)_e - (\rho u \phi)_w = \left( \Gamma \frac{d\phi}{dx} \right)_e - \left( \Gamma \frac{d\phi}{dx} \right)_w \quad (29)$$

が得られる。φ に関して界面の値を線形補間（中心差分）から求められるものとする、

$$\phi_e = \frac{\phi_E + \phi_P}{2}, \quad \phi_w = \frac{\phi_P + \phi_W}{2} \quad (30)$$

が成立する。さらに、微分の項については差分近似するものとする、式(29)は次のように書き換えられる。

$$\frac{1}{2}(\rho u)_e(\phi_E + \phi_P) - \frac{1}{2}(\rho u)_w(\phi_P + \phi_W) = \Gamma_e \frac{(\phi_E - \phi_P)}{\Delta x} - \Gamma_w \frac{(\phi_P - \phi_W)}{\Delta x} \quad (31)$$

交換係数 Γ については、[3.2.1 節](#)で述べた CV 界面における熱伝導率の補間方法のいずれかを用いるものとする。この式を簡略化するために二つの記号を定義する。

$$F \equiv \rho u, \quad D \equiv \Gamma / \Delta x \quad (32)$$

なお、式(28)から F は一定であるとわかる。また、u が速度なので F は負にもなり得る。この記号を用いると、離散化方程式を次のように表すことができる。

$$a_P \phi_P = a_E \phi_E + a_W \phi_W \quad (33)$$

ここで、

$$a_E = D_e - \frac{F_e}{2}, \quad a_W = D_w + \frac{F_w}{2}, \quad a_P = D_e + \frac{F_e}{2} + D_w - \frac{F_w}{2} \quad (34)$$

であり、 $a_P$  についてはさらに、 $F_e$  と  $F_w$  が等しいことから次式が得られる。

$$a_P = a_E + a_W. \quad (35)$$

この離散化方程式から現実的な解が求められるためには、式(32)の係数がすべて正でなければならない。  $a_E < 0$  であるような条件、たとえば、 $F_e = F_w = 1$  および  $D_e = D_w = 4$  の場合、 $a_E = -1$ 、 $a_W = 3$ 、 $a_P = 2$  となる。このとき  $\phi_E = 200$ 、 $\phi_W = 100$  ならば、 $\phi_P = 50$  となり、隣接するメッシュ値の範囲外の値になってしまい、現実的とはいえない。物理的にも、あるメッシュの温度を考えると、両側に隣接するメッシュの温度がわかっている、対流と熱伝導しか起こらないのであれば、そのメッシュの温度は両側に隣接するメッシュの温度の間になっているはず

ずである。次に、式(33)の係数がすべて正になるようなメッシュ境界での対流項の補間の方法を説明する。

### 3.3.2 風上差分法

前節のような定式化によって非現実的な離散化方程式が導かれてしまうことを回避するための方法として風上差分法がある。この方法は、拡散項はそのまま変更せずに、対流項についてだけ次の仮定によって計算する。

「ある界面での対流によって運ばれる量  $\phi$  の値はその界面の風上側の格子点での  $\phi$  の値に等しい。」

これを式で示すと、

$$\begin{cases} \phi_e = \phi_P & (F_e > 0) \\ \phi_e = \phi_E & (F_e < 0) \end{cases} \quad (36)$$

となる。 $\phi_w$ の値も同様にして決まる。この方法を用いた場合の離散化方程式の係数は次のようになる。なお、 $[A, B]$  は  $A$  と  $B$  の大きいほうを表すものとする。

$$\begin{aligned} a_E &= D_e + [-F_e, 0], & a_W &= D_w + [F_w, 0] & (37) \\ a_P &= D_e + [F_e, 0] + D_w + [-F_w, 0] & & & (38) \end{aligned}$$

$D$  は常に正であるから、この定式化では離散化方程式の係数が常に正になり、非現実的な解の導出を確実に避けられる。

この補間方法の精度を求めておく。 $F_e > 0$  とすると、式(29)は、

$$\frac{d}{dx}(\rho u \phi)_P = F \left( \frac{d\phi}{dx} \right)_P = \frac{F}{\Delta x} (\phi_E - \phi_P) \quad (39)$$

となる。 $\phi_E$  はテイラー展開を用いて、

$$\phi_E = \phi_P + \left( \frac{d\phi}{dx} \right)_P \Delta x + \frac{1}{2!} \left( \frac{d^2\phi}{dx^2} \right)_P (\Delta x)^2 + \frac{1}{3!} \left( \frac{d^3\phi}{dx^3} \right)_P (\Delta x)^3 + o((\Delta x)^4) \quad (40)$$

と表せる。これを式(39) に代入すると、

$$\frac{F}{\Delta x} (\phi_E - \phi_P) = F \left( \frac{d\phi}{dx} \right)_P + F \left\{ \frac{1}{2!} \left( \frac{d^2\phi}{dx^2} \right)_P (\Delta x) + o((\Delta x)^2) \right\} \quad (41)$$

となり、式(41)右辺第 2 項以降が離散化誤差であり、 $(\Delta x)$  のオーダーの誤差を含むことがわ

かる。よって風上差分法は 1 次精度 (FLUENT の Solution Methods では First Order Upwind と表記) であり, 粗いメッシュを用いてしまうと精度が非常に悪くなる。

### 3.3.3 2 次精度中心差分法

本節の最初のところで挙げた, 中心差分法は, メッシュ界面における対流項の補間法としては風上差分よりも精度の高い方法である。ここでは精度が高くなることを証明する。この方法では界面での  $\phi$  の値は式(30)で定義される。この定義を用いた場合, 式(29)は,

$$\frac{d}{dx}(\rho u \phi)_P = F \left( \frac{d\phi}{dx} \right)_P = \frac{F}{\Delta x} \left( \frac{\phi_E + \phi_P}{2} - \frac{\phi_P + \phi_W}{2} \right) = \frac{F}{2\Delta x} (\phi_E - \phi_W) \quad (42)$$

となる。また,  $\phi_W \cdot \phi_E$  はテイラー展開を用いて,

$$\phi_W = \phi_P - \left( \frac{d\phi}{dx} \right)_P \Delta x + \frac{1}{2!} \left( \frac{d^2\phi}{dx^2} \right)_P (\Delta x)^2 - \frac{1}{3!} \left( \frac{d^3\phi}{dx^3} \right)_P (\Delta x)^3 + o((\Delta x)^4) \quad (43)$$

$$\phi_E = \phi_P + \left( \frac{d\phi}{dx} \right)_P \Delta x + \frac{1}{2!} \left( \frac{d^2\phi}{dx^2} \right)_P (\Delta x)^2 + \frac{1}{3!} \left( \frac{d^3\phi}{dx^3} \right)_P (\Delta x)^3 + o((\Delta x)^4) \quad (44)$$

となる。これらを式(42)に代入すると,

$$\frac{F}{2\Delta x} (\phi_E - \phi_W) = F \left( \frac{d\phi}{dx} \right)_P + F \left\{ \frac{1}{3!} \left( \frac{d^3\phi}{dx^3} \right)_P (\Delta x)^2 + o((\Delta x)^3) \right\} \quad (45)$$

となり,  $(\Delta x)^2$  のオーダーの誤差を含むことがわかる。つまり, この離散化式の精度は 2 次である。このため, この方法は 2 次精度中心差分法ともよばれる (FLUENT の Solution Methods では Second Order Upwind と表記)。こちらの方法のほうが風上差分法に比べて 1 次精度が高い。ただし, 3.3.1 節で述べたように, 場合によっては現実的な解が得られないこともあるので注意が必要である。精度の次数が高い方法が必ずしも精度の高い解をもたらしてくれるとはかぎらない。

## 3.4 SIMPLE 法 (圧力・速度のカップリング)

有限体積法では, 運動方程式と連続の式は逐次的に解かれる。この計算では, 連続の式が圧力方程式として使用される。しかし, 非圧縮性流体の場合, 密度と圧力の間に直接的な関係が無いために離散化した連続の式に圧力が明示的に現れてこない (式(33)参照)。そこで, SIMPLE 法 (Semi-Implicit Method for Pressure-Linked Equation, 圧力結合方程式の半陰解法) というアルゴリズムを用いて連続の式に圧力の項を導入することによって速度と圧力を

同時に反復計算する。本節では運動方程式の離散化と、SIMPLE 法によって速度場・圧力場を求めるアルゴリズムについて説明していく。

### 3.4.1 運動方程式

ここでも 1 次元の領域を例に挙げる。図 5 で示す CV について運動方程式を積分して、離散化方程式を求めるのであるが、式(16)・(33)からわかるように、得られる離散化方程式の形は本質的には同じであり、このことは運動方程式にもあてはまる。 $u$  に対する CV の界面における拡散係数や質量流量の計算は、適切な補間を必要とするが、これも 3.2・3.3 節で述べた定式化を応用することができる。

これらを考慮すると、図 5 の灰色で示した CV における離散化方程式は次のように書ける。

$$a_w u_w = a_w u_w + a_p u_p + b + (p_w - p_p)A \quad (46)$$

ここで  $b$  は重力やその他の外力による生成項であり、右辺の最後の項が圧力勾配によって生じる項である。本来は最後の項も生成項に含まれるのだが、ここでは分離して表記する。 $A$  は圧力の作用する面積であり、この項はこの CV に作用する圧力差による力を表していることになる。

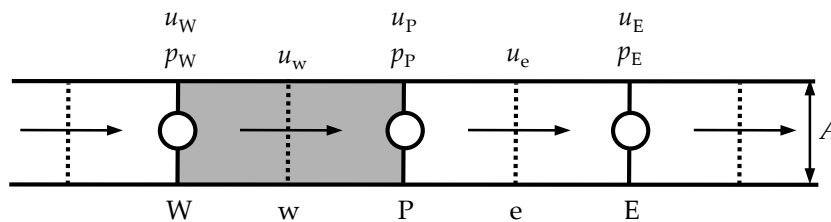


図 5  $u$  に対する CV (灰色が一つの CV に相当する)

### 3.4.2 圧力・速度の補正

収束計算では、推測した圧力場  $p^*$  を式(46)に代入して、この式を解くことによって速度場  $u^*$  が得られる。このとき、 $p^*$  と  $u^*$  は次の関係を満たしている。

$$a_w u_w^* = a_w u_w^* + a_p u_p^* + b + (p_w^* - p_p^*)A \quad (47)$$

この速度場を連続の式に代入する。一般に、 $u^*$  は連続の式を満たさないため、圧力場を推測値の補正が必要になる。正確な圧力を  $p$ 、速度を  $u$  とすると、それらは、

$$p = p^* + p', \quad u = u^* + u' \quad (48)$$

で表されるものとする。この式の  $p'$ ・ $u'$  はそれぞれ圧力・速度の補正值になる。式(46)から式(47)を引くと、

$$a_w u'_w = a_w u'_w + a_p u'_p + (p'_w - p'_p)A \quad (49)$$

が得られる。ここで右辺の第1項・第2項を落とすと、

$$u'_w = (p'_w - p'_p)d_w \quad (50)$$

$$d_w = A/a_w \quad (51)$$

となる。この2項を落としても計算上問題ないことは [3.4.3 節](#) で述べる。式(50)は速度補正方程式とよばれ、次のようにも書ける。

$$u_w = u_w^* + (p'_w - p'_p)d_w. \quad (52)$$

次に、連続の式を圧力補正のための式へと変形する。まず、1次元定常で生成項がない場合、図5でPと記したCVについての離散化方程式は、

$$\rho_e u_e A - \rho_w u_w A = 0 \quad (53)$$

となる。この式に [式\(45\)](#) で示した速度の表現を代入して整理すると、

$$\rho_e u'_e A - \rho_w u'_w A = \rho_w u_w^* A - \rho_e u_e^* A \equiv D_p^* \quad (54)$$

となり、右辺は  $u^*$  を用いたときの連続の式に対する誤差を示す。この式に速度補正方程式(52)を代入すると、

$$\rho_e (p'_p - p'_e) d_e A - \rho_w (p'_w - p'_p) d_w A = D_p^* \quad (55)$$

となり、この式も  $p'$  についての一般的な離散化方程式の形に整理できる。

$$a_p p'_p = a_w p'_w + a_e p'_e + D_p^* \quad (56)$$

$$a_e = \rho_e d_e A, \quad a_w = \rho_w d_w A, \quad a_p = a_e + a_w \quad (57)$$

これで連続の式に圧力の項が導入されたことになる。

### 3.4.3 SIMPLE のアルゴリズム

SIMPLE 法の実行手順を以下に示す。

- (1) 圧力場  $p^*$  を推測する。
- (2)  $u^*$  を得るために [運動方程式\(47\)](#) を解く。
- (3) (2) で求めた  $u^*$  を用いた場合の連続の式に対する誤差  $D_p^*$  を式(54) から求める。
- (4) [式\(56\)](#) を解いて  $p'$  を求める。
- (5) [式\(48\)](#) から  $p$  を求める。



- (6) [速度補正式\(52\)](#)を用いて  $u$  を求める。
- (7) もし他の変数（温度など）が流体の性質などを介して流れ場に影響を与えるのならば、それに対する離散化方程式を解く。
- (8) 補正された圧力  $p$  を新しく推測した圧力  $p^*$  としてステップ(2)に戻り、収束解が得られるまで全手順を繰り返す。

このアルゴリズムでは、[3.4.2 節](#)で[速度補正式\(52\)](#)を導く際に右辺の第 1 項・第 2 項を省略している。まず、この 2 項を省略する理由を説明する。これらの項を残すと、これらを  $u_P \cdot u_w$  の隣接点における圧力補正項と速度補正項で表現しなければならない。これらの隣接点の値はさらに次々とそれらの隣接点の値を必要とし、結局、速度補正式は計算領域のすべての点における圧力補正項を含んでしまう。その結果として得られる圧力補正式は取り扱い不可能であるし、運動方程式と連続の式を逐次的に解くことも不可能になってしまう。しかし、この 2 項を省略することで  $p'$  の式を一般の  $\phi$  と同様の離散化式にすることができ、一度に一つだけの変数を解くという逐次解法が可能になる。ステップ(2)から(6)をみれば、どのステップも速度場と圧力場のどちらか一方しか一度に解いていないことがわかる。

また、この方法で正しい収束解が得られる理由は、収束したといえる最後の反復計算における操作に注目するとわかる。それまでのすべての反復計算の結果として、ある圧力場を得ているものとする。これを  $p^*$  とし、ステップ(2)で  $u^*$  が求まる。ステップ(3)で  $D_p^*$  を求めることになるが、最後の反復計算において、この値は実質上 0（厳密には  $D_p^* < \epsilon$ ）となっているはずである。つまり、ステップ(2)で求めた  $u^*$  は運動方程式だけでなく、連続の式も満たしていることになり、 $p^* \cdot u^*$  はそれ自身が正しい解になっている。そのため、最後の反復計算ではステップ(4)以降を実際に解く必要はなく、最後の反復計算に使用しない  $p'$  と速度補正の式を作る際にどのような近似を導入しても収束解に影響しない。

## 3.5 収束計算の手法

安定・迅速に、かつ精度よく収束計算を収束させるための方法をここでは述べる。計算を安定・迅速に計算を行うために不足緩和法と緩和係数の設定法について述べた後、計算精度を向上させるための離散化スキームの選択法、収束計算が収束したことをどのように判断するかについて述べる。

### 3.5.1 緩和係数

前節で示した連立方程式は非線形成を有しており、また、SIMPLE により速度と圧力のカップリングをしているため、変数の補正量をコントロールして解の発散を防ぐ必要がある。発散を防ぐためには変数の変化を減速してやればよく、これを不足緩和とよぶ。

次のような一般的な離散化方程式を考える。

$$a_p \phi_p = \sum a_{nb} \phi_{nb} + b. \quad (58)$$

この式の右辺第 1 項は隣接項の和を示しており、2 次元なら四つの隣接するメッシュの値、3 次元なら六つの隣接するメッシュの値の和となる。さらに、 $\phi_P^*$  は前回の反復における  $\phi_P$  の値とする。式(58) は次のように書くこともできる。

$$\phi_P = \left( \sum a_{nb} \phi_{nb} + b \right) / a_P. \quad (59)$$

$\phi_P^*$  を右辺に加え、再び減ざると、

$$\phi_P = \phi_P^* + \left\{ \left( \sum a_{nb} \phi_{nb} + b \right) / a_P - \phi_P^* \right\}. \quad (60)$$

この式で、 $\{ \}$  内が今回の反復によって変化した  $\phi_P$  の値に相当する。緩和係数  $\alpha$  を用いてこの変化分をコントロールする。

$$\phi_P = \phi_P^* + \alpha \left\{ \left( \sum a_{nb} \phi_{nb} + b \right) / a_P - \phi_P^* \right\}. \quad (61)$$

$\alpha$  が 1 より小さいと不足緩和になる (FLUENT では 1 以下にしかできない)。 $\alpha$  が小さいほど変数の変化が小さくなるので発散を防ぐ効果が大きくなる。ただし、変数の変化が遅いため収束は遅くなる。緩和係数の最適値を選ぶ一般的な法則はなく、問題の性質や格子点の状態などに依存する。

FLUENT のデフォルトでは緩和係数を圧力は 0.3、運動量は 0.7、エネルギーは 1.0 となっている。基本的にはこの係数で計算を行い、解が発散しそうであればこの値を 0.1 ずつ小さくし、解または残差が単調な変化をするようになったら元の値に戻して収束を加速させるという方法を取るとよい。

圧力の項は発散しやすく、流路出口で Reversed Flow の表示 (場合によっては Turbulent Viscosity Ratio も) が収束計算中に表示され続けることがある (最終的には安定する計算でも反復回数が少ないうちはこれが出る)。

非等温の計算では非常に発散しやすいので、よほど温度変化や反応による発熱が弱くないかぎり、Energy の緩和係数は 0.9 以下にしておくといよい。ただし、あまり小さくしないほうがよく、計算が安定してきたら 1 に戻す。また、計算の序盤で流れだけの計算を収束させた後、流れの計算を外してエネルギーと各成分の保存式だけを解いて収束させ、最後にすべての式を同時に解きながら収束させると上手くいくことがある。流れとエネルギーが連成していると非常に発散しやすいため、このような工夫が有効に働くこともある。非等温かつ、気体の流れで密度が圧力に依存するような圧力・流れとエネルギーが強く連成している場合は、すべての緩和係数を小さくする必要がある (デフォルトの値の 2 分の 1 から 3 分の 1)。

### 3.5.2 離散化スキームの選択

3.3 節で述べたように 2 次精度中心差分 (Second Order Upwind) のほうが精度は高い (QUICK はさらに上) が, 現実的な解が得られない可能性もある。しかし, この方法でもメッシュ間隔  $\Delta x$  を細かくすれば式(32)で表される  $D$  を大きくすることができ, 式(32)の離散化係数を正にすることが可能である。十分細かいメッシュを用いているならば, 2 次精度中心差分を用いたほうが精度の高い解が得られたため, 高い精度の解が必要ならばこちらを用いるべきである。また, 解の精度がそれほど高くなくてよく, 計算機負荷を軽減するために少ないメッシュで計算したいときは 1 次精度風上差分法を用いてもよい。収束が悪いときも 1 次精度でまずは計算し, ある程度収束してきたら高次精度のスキームに変えていくとよい。

### 3.5.3 収束の判定

収束の判定の最たる指標は残差 (Residual) である。ある変数の収束計算  $n$  回目の残差は, 収束計算  $n-1$  回から  $n$  回の間各セルの変数の変化量の総和÷各セルのその変数の  $n$  回目の時点での値の総和である。残差がある一定の値以下 (一般に速度・圧力は  $10^{-3}$ , エネルギーは  $10^{-6}$ ) になったら収束とみなすという方法が一般的であるが, 残差がこの値より小さくなくてもまだ変数は変化を続けている場合が多い。そこで, 流路出口断面の質量平均温度・鍵成分の質量分率・流速, 流路入口の圧力 (厳密には流路出口圧力との差圧) などもモニタリングし, これらの値が有効数字 4 桁以上変化しなくなったら収束したと判定するほうが確実である。

非等温反応系では, 収束計算途中では断熱温度上昇以上に温度が上がってしまい, デフォルトのリミットである 5000 K に温度が張り付いてしまうときがある。このようなときは, FLUENT 画面の上部メニューの **Solve** → **Controls** を開き, 緩和係数の入力欄の下の **Limits** のボタンを押して出てくるウィンドウで, **Maximum Static Temperature** の値を一時的に低くし, 上限値の温度に張り付いているメッシュ数が減ってきたところでリミットの上限値をデフォルトに戻すとよい。

また, メッシュ数が多くなるほど収束に必要な収束計算回数は増加する。収束計算回数を固定して異なるメッシュ粗さの結果を比較しても合わないのはこのせいでもある。

## 3.6 メッシュ生成

計算領域の描画は 4 章で説明する DesignModeler, メッシュの生成, 境界条件の種類の設定は Meshing で行う。Meshing では, 2 次元では長方形・四角形・三角形, 3 次元では, 直方体・六面体・四面体の形状のメッシュで計算流域を離散化できる。長方形・四角形・直方

体・六面体メッシュはアスペクト比が 10 程度まで安定な計算をしてくれる。しかし、三角形・四面体メッシュでゆがみの大きいものになると変数の変化が不安定になりやすく、ゆがみの小さいメッシュで切ろうとするとメッシュ数が非常に多くなり、計算時間が増大する。このため、数値計算を精度よくかつ効率的に行うためには四角形・六面体メッシュのほうが優れている。このことから、複雑な形状の計算領域のできるだけ多くを六面体メッシュで切るために、計算領域を細かく分割する必要がある。また、数値計算では離散化誤差によって見掛け上拡散しているような解が得られることがある。これは数値拡散とよばれ、数値計算（とくに非定常計算）を安定させるという利点もあるが、実際の濃度勾配から生じる拡散ではないので、収束解の誤差にもつながってしまう。[3.3 節](#)で示したメッシュ界面での補間には高次の精度のものを用いたほうが数値拡散は小さくなる。また、メッシュを流れ方向に平行にしたときに数値拡散は最小になる。数値拡散を小さくするためにも長方形・直方体メッシュを利用することは有効である。

導入としては 2 次元モデルから入るのが一つの手であるが、混合部で対流効果、とくに渦発生が伴う可能性のある計算では 3 次元にしてメッシュを非常に細かくする必要がある。数百万というメッシュが必要になることもある。

精度の高い解を求めるためには変数の勾配の大きい領域に対して細かくメッシュを切る必要がある。FLUENT では、粗いメッシュで一度収束計算を収束させた後、指定した変数の勾配を用いて、勾配の大きい領域だけメッシュを細かくすること（Mesh Adaption）ができる（[4.2.8 節参照](#)）。

メッシュ数の目安については、FLUENT で粘性のモデルを「Laminar」にすると、流路全体に対して、同じ保存式を解いていることになる。境界層の発達を表現するためには、最低でも 5、できれば 10 以上流路の幅方向にメッシュを切るようにしたい。流体衝突による対流効果や分子拡散の効果を正確に計算したいときは 1 メッシュあたりのサイズを数  $\mu\text{m}$  以下にする必要がある。さらに、壁近傍のメッシュ幅を小さくとるとよい。軸方向については、反応が伴う場合は流路入口付近が最も反応が速いため、細かくメッシュを切る必要がある。非等温反応系では軸方向のメッシュを細かくするほど温度が発散しにくくなり、計算を収束させやすい。

さらに、CFD シミュレーションの結果がメッシュの切り方に依存していないことを確かめる必要がある。このために、同じ設定条件でメッシュ数だけが異なる計算を少なくとも 2 回は行うと確実である。

### 3.7 上手くいかなかったシミュレーションの活用報

CFD シミュレーションをしていて、計算が収束しない、実験結果と大きくずれている、など上手くいかない、失敗だったと感じてしまうことがある。このような状況からも引き出せる情報はある。一見上手くいっていない状況をどう活かすかを以下に挙げておく。

- ・ 実験とシミュレーションの結果がぴったり合わないから使えない、というわけではない。合わないのは実験結果を支配している因子がシミュレーションのモデルが対象としている因子の外にあるから。
- ・ 精密なモデルでも簡略化したモデルでも実験結果と合うなら、簡略化したモデルに入っている因子が結果に大きく影響することがわかる。精密なモデルだけが合うのなら、精密なモデルには入っているが簡略化したモデルにはない因子が重要だとわかる。
- ・ 上記の観点からはすべての収束解には何らかの意味がある。収束しなくても考慮すべきことがほかにあることが示されていると解釈できる。

ここまでが CFD の理論に関する基礎の部分と、それに即したメッシュの切り方・収束計算のコツの要点である。より深く CFD について学ぶためには、最後の参考文献で学習してほしい。また、CFD でできることは多様なので、個人個人でやりたいことを上手くこなすコツは経験的に身に付けていかななくてはならない部分もある。では、その経験を積むためにも、実際に次節の説明に従ってソフトを使いながら CFD ソフトの使い方を学んでいこう。

## 4 ANSYS® の利用法

前節で CFD の理論を概観したので、今度は実際に Case Study をしながら ANSYS® の使い方を学んでいく。最初に ANSYS® の User's Guide にあるチュートリアルを実施してみよう。次いで、基本 Case Study として円管層流+単一 2 次反応の定常計算を扱い、非定常シミュレーションの例として、この円管でのステップ応答の計算を行う。乱流モデルを用いた例も示す。さらに熱交換のシミュレーション、周期境界条件の設定法を述べる。最後に、複雑な計算領域の描画のコツについてもふれる。なお、計算には時間がかかるものもある。こまめにデータのバックアップを取るようにしよう。

### 4.1 ANSYS FLUENT の Tutorial

FLUENT は、プログラムの **ANSYS 15.0** → **Fluid Dynamics** → **FLUENT** から起動する。**FLUENT** のところで右クリックし、**送る** → **デスクトップ (ショートカットを作成)** でショートカットを作成しておいて立ち上げてよい。

ANSYS ユーザーサービスセンター (USCJ) <http://www2.ansys.jp/> の Web サイト左側メニューの製品情報→ANSYS Fluent→ANSYS FLUENT 12.1 製品ドキュメント (日本語版) とたどって見られるページ <http://www2.ansys.jp/uscj/products/fluent/12.1/doc/index.htm?sc=vke3748uucjsgvake6p2ntl744> にある「FLUENT 12.0 in Workbench チュートリアル」と「チュートリアルガイド」のチュートリアル 1 の内容を実施してみよう。英語版で同じ内容のものは、FLUENT 画面上部メニューに **Help** の中の **User's Guide Contents** を選ぶとウィンドウが立ち上がる、そこで表示されるページの中で **FLUENT** → **Tutorial Guide** → **Chapter 1** (Introduction to Using ANSYS FLUENT in ANSYS Workbench: Fluid Flow and Heat Transfer in a Mixing Elbow) と **Chapter 3** (Introduction to Using ANSYS FLUENT: Fluid Flow and Heat Transfer in a Mixing Elbow) が対応する (Help → PDF は機能していない模様)。Workbench での CFD シミュレーションの進め方、DesignModeler での計算領域の描画方法、Meshing でのメッシュの切り方、FLUENT でのシミュレーションの設定・収束計算・結果処理の方法が一通り経験できる。自分が実施したいシミュレーションの具体例と関連しそうなものがあればチュートリアルガイドのほかの章も実施しておくといよい。

### 4.2 反応が含まれる Case Study

反応工学に扱う理想反応器としての Plug Flow Reactor (PFR) では、幅方向に均一な速度分布が形成され、この方向には完全混合が成り立つという仮定をしている。しかし、実際の流路では幅方向に速度分布が生じる。たとえば、流れが層流のときは、放物型の速度分布が発達し、流路の幅方向に滞留時間分布が生じる。この滞留時間分布は幅方向・軸方向の拡散によっても変化し、反応の進行にも影響をあたえる。ここでは円管内層流下での均相系の単



一 2 次反応について拡散係数・管径と反応進行の関係について調べながら FLUENT の使い方を説明していく。

計算領域の流れと反応の設定を最初に述べておく。平均流速  $u = 1 \text{ m/s}$  (円管半径  $R = 100 \mu\text{m}$  のときに  $Re = 2.0$ )、平均滞留時間  $\tau = 0.01 \text{ s}$  (円管長さ  $1 \text{ cm}$ )、反応速度定数  $k = 1 \text{ m}^3 \cdot \text{kmol}^{-1} \cdot \text{s}^{-1}$  (等温系で一定) とし、式(62)に示す単一 2 次反応の CFD シミュレーションを行う。なお、反応流体の物性は FLUENT の水の物性を用いて、密度は  $998.2 \text{ kg/m}^3$ 、粘度は  $0.001 \text{ Pa} \cdot \text{s}$ 、分子量は  $18.02 \text{ kg/kmol}$ 、入口における A の濃度  $C_{A0} = 55.4 (998.2/18.02) \text{ kmol/m}^3$  とする (入口の流体がすべて A からなるものとする、FLUENT は質量ベースで計算するので物性の密度と分子量、入口に相当する境界で設定する質量分率から反応成分のモル濃度が決定される)。



上記の課題と設定の下、FLUENT を用いた CFD シミュレーションの一連の流れを説明していく。大まかな流れとしては以下のようなになる。

- DesignModeler で計算領域の形状を描く (4.2.1 節)
- Meshing で計算領域をメッシュに離散化 (4.2.2 節)
- Meshing で境界条件の種類を設定 (4.2.3 節)
- Meshing で生成したメッシュを msh ファイルに変換して FLUENT に導入 (4.2.4 節)
- FLUENT での作業に入り、実際のサイズに縮尺の変更、流れや物質移動のモデルの選択、物質・流れの物性、反応、境界条件の定義 (4.2.5 節)
- 離散化スキームと収束計算モニター方法を指定し、収束計算 (4.2.6 節)
- 収束計算の結果を確認 (4.2.7 節)
- 必要に応じて計算領域の Grid Adaption (3.6 節参照) →再度収束計算 (4.2.8 節)
- 得られた速度・濃度分布から結果の整理

ほかにも、層流の放物型速度分布といった複雑な境界条件 (付録 D.1 を参照) や物性の温度の依存性 (付録 D.2 を参照) を導入するためにユーザー定義関数 (User Defined Function, UDF) を用いることもできる (4.2.9 節)。では、計算領域を描画するために Workbench から DesignModeler を起動するところから説明していく。

### 4.2.1 計算領域の描画

今回は円管が計算領域の形状になるが、円管の中心が回転対称軸になるため長方形を描き、底辺を回転対称軸にすれば 2 次元でモデル化できる。

計算領域の形状の描画は DesignModeler で行う。DesignModeler を起動するには、まずプログラムの **ANSYS 15.0** の中にある **Workbench 15.0** をクリックする。次に、立ち上がった Workbench の画面左側 Toolbox の **灰色の直方体 + Geometry** の項目をつかんで右側の白い部分上部で離す。

※Workbench の画面左側の Toolbox に DesignModeler が表示されていない場合は、ライセンスマネージャが起動していない可能性もある (FLUENT も立ち上がらない)。この場合は

PC を再起動して、プログラムの「ANSYS, Inc. License Manager」の下にある Server ANSLIC ADMIN Utility を立ち上げ、左側の一番下にあるステータス診断でエラーが出ないことを確認する。ライセンスマネージャが起動していないようなら「ANSYS, Inc.ライセンスマネージャの起動」を選択してサーバーを起動させる（Tool→Set language option で Japanese にすると日本語表記にできる）。ライセンスマネージャが起動していても DesignModeler や FLUENT が使えない場合は「ANSYS, Inc.ライセンスマネージャの停止」をクリックした直後に「ANSYS, Inc.ライセンスマネージャの起動」を選び、ライセンスマネージャを再起動すると使えるようになる。

画面右側に灰色の直方体+Geometry の項目が生成する。二つ目の？がついた項目をダブルクリックする。すると DesignModeler が起動する（クリックした項目の左のマークが黄緑色の円のなかに DM と書かれたものになる）。サイズの単位は millimeter を選んでおく。

なお、ANSYS は 3 次元形状を前提としているためか、FLUENT の Tutorial では FLUENT のツールボックスで形状描画から CFD 計算までを連続して実施できたが、2 次元形状のまま形状描画とメッシュ形成を行い、CFD 計算に入るため、Workbench では作業が不十分と判定され、FLUENT のツールボックスでは連続的に作業ができない。このため、ここでは FLUENT の Tutorial とは異なった Workbench の使い方になっている。

実際の操作の前に DesignModeler, Meshing で描いた図形表示の変更方法を説明しておく。↑↓←→のキーで上下左右に少しずつ平行移動できる。マウスの中ボタンを上をスクロールすると拡大、下をスクロールすると縮小される。マウスの中ボタンを押したまま左右に動かすと回転する。XY 平面 ( $Z = 0$ ) の表示には上部メニューの「Look At Face/Plane/Sketch」ボタン（人が黄緑色の面を見ているアイコン）ボタンを押す。

描画の作業に入ったらまず、DesignModeler の上部メニューで「XYPlane」と表示されているところの右側にある **New Sketch** をクリックする（図 6 参照）。次に長方形を描くために、「Sketching」のタブをクリックして切り替える。Sketching Toolboxes の一番下にある Settings の Grid を選んで二つのチェックボックスに両方ともチェックを入れておく。Snap にチェックを入れると位置を指定するときに Grid に合わせてくれる。Major Grid Spacing を 10 mm, Minor-steps per Major を 1 にしておく。Toolboxes の Draw に移り、Snap の機能と Grid の 1 マスが 10 mm になっていることを使い、Rectangle を選択して一辺が 10 mm の正方形を書いてみよう。二つの頂点の位置をクリックすればよい。今指している頂点の位置は画面右下に出ている。左下の頂点が原点にくるようにしておく。ここでは正方形になっているが FLUENT に入ってから縦横の比は変更できるので今はこれで問題ない。表示が見にくいいため描きにくいこともあるので、無理に最初から正確な比で形状を描く必要はない。この操作で XYPlane の下に Sketch1 ができている。



ANSYS UI キャプチャ画面のため非公開

図6 DesignModeler の画面

今 Sketch で書いたものは計算領域の下書きのようなものなので、物理的な領域を次に描く。DesignModeler の画面上部のメニューにある **Concept** から **Surfaces From Sketches** を選ぶと **Tree Outline** に SurfaceSk1 が追加される。これで先の正方形のスケッチを Surface (形状の表面) に置き換える。Tree Outline で SurfaceSk1 をクリックすると SurfaceSk1 の詳細情報が画面左下に表示される。この中に Base Objects の欄がある。次に Tree Outline で **Sketch1** をクリックし、Base Objects (形状の下書きである Sketch をここに選ぶ) の欄にある **Apply** をクリックする。さらに Tree Outline で SurfaceSk1 を右クリックして Generate を選択すると正方形の内部が灰色になる。これで表面が完成する。

1 Part, 1 Body の左の+ボタンをクリックして作成した領域が Tree Outline に見えるようにする。左下の Details View の Fluid/Solid で領域を **Fluid** に変えておこう。デフォルトの Solid のままだとその領域に流体が流れることを再現できない。4.5 節の例題のように壁領域は Solid にする。

最後に DesignModeler の形状の入ったファイルを保存するために **File** → **Export** を選択し、拡張子はそのままで **tutorial-tube.agdb** の名前で保存しよう。DesignModeler を閉じて Workbench の画面上部メニュー **File** → **Save As** でここまでの内容を一度保存しておく。

#### 4.2.2 計算領域のメッシュへの離散化

計算領域が描いたら、その領域をメッシュに離散化する。Workbench 画面左側の Toolbox から **Mesh** をドラッグし、画面右側の空いている領域にドロップしよう。Mesh, Geometry, Mesh と表示されたカラムが出てくる。**Geometry** のところで右クリックして **Import Geometry** → **Browse** (ファイル名が直接出ていればそれを選んでよい) で先に保存したファイル **tutorial-tube.agdb** を選択する。3 段目の Mesh の文字の右側のマークが緑色に変化する (図7)。3 段目の Mesh (図7で選択されている部分) をダブルクリックすると Meshing が立ち上がる。

ANSYS UI キャプチャ画面のため非公開

図7 Workbench の画面

Meshing の画面左側の **Outline** に読み込んだ形状の一覧が表示される。2 次元形状を読み込んだ際は Geometry と Surface Body のところに「？」が表示されるが問題ない (3 次元形状を前提としたソフトのため)。画面右側に Meshing Options が出てくることがあるが変更せずにそのままでもよい (閉じてもよい)。

XY 平面を正面から見た状態にしたければ、画面右下の座標表示で「Z 軸」をクリックすればよい。形状の描画サイズを調整したければ、描画領域を一度左クリックし、マウスの中ボ

タンを上下させると拡大縮小できる。

辺 (Edge) 上にメッシュを切る間隔をまずは指定する。これは先にも述べたように、極力計算領域全体を長方形でメッシュに切り、計算精度を上げるために行う。立方体の中に矢印があるアイコンが画面上部に並んでいる内の辺が黄緑色になっているところをクリックして形状内の辺を選択できるようにしておく (図 8 で対応するアイコンが押された状態になっている)。Outline の **Mesh** をクリックした状態で図 8 のように **Mesh Control** で **Sizing** を選ぶ。Outline に出ている「**Sizing**」をクリックし、 $X=0$  mm の Y 軸に平行な辺を左クリックで選択し、画面左下の Details の Scope-Geometry の欄にある **Apply** のボタンを押すとその辺が選択される (この欄が 1 Edge になる)。Details の下のほうにある Definition で次のように設定する (書いていないものはそのままよい)。

- Type : **Number of Divisions** (その辺のメッシュへの分割数を指定する)
- Number of Divisions : **10**
- Behavior : **Hard** (こちらの設定を確実に反映させるため、Soft だと Meshing 側である程度変更されてしまう)
- Bias Type : 中央が長い線になっているもの (選択した部分の端のメッシュが小さく、中央が大きくなるようにメッシュを切ってくれる、選択肢の見たとメッシュが着られる間隔の変化が対応している、No Bias だと選択した部分を均等に分割)
- Bias Factor : **2** (最も大きい間隔 ÷ 最も小さい間隔 = 2)

$X = 10$  mm の Y 軸に平行な辺を左クリックで選択して上記と同様の設定を繰り返す。辺を選ぶときに Ctrl キーを押して二つの辺を同時に設定することもできる。平行な辺に対してメッシュを切る間隔を同じにしておくことで長方形メッシュになるようにする。

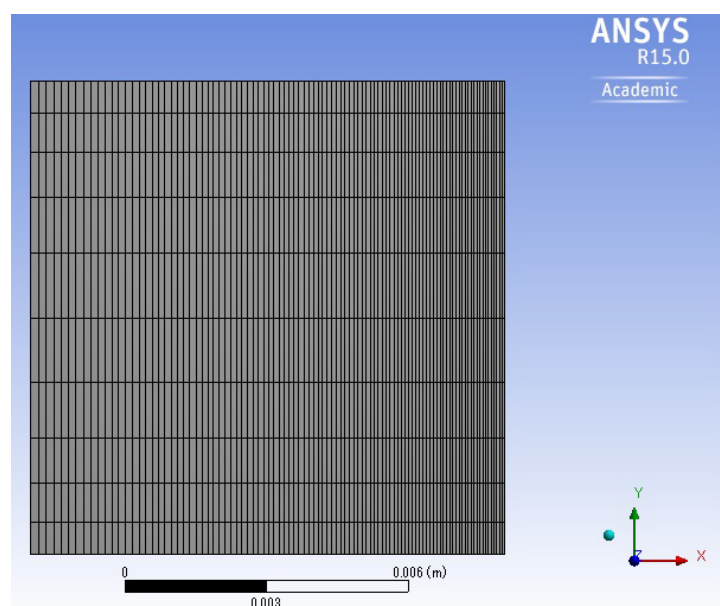


図 8 Meshing の画面 **ANSYS UI 部分を削除している**

次に X 軸上の辺を以下のように設定する。

- Type : **Number of Divisions**
- Number of Divisions : **100**
- Behavior : **Hard**
- Bias Type : 左側の線が長いもの
- Bias Factor : **3** (入口ほどメッシュが細くなる, 入口付近で反応が最も速く進むため細かいメッシュが必要になる, 反応が非常に速い場合で総メッシュ数を増やしたくないときはこの比を 10 以上にしてもよい)

さらに Y = 10 mm にある X 軸に平行な辺を設定する。

- Type : **Number of Divisions**
- Number of Divisions : **100**
- Behavior : **Hard**
- Bias Type : 右側の線が長いもの (先と向きを逆にする必要があることに注意)
- Bias Factor : **3**

Y 軸に平行な 2 辺は中央のメッシュが大きくなるように, X 軸に平行な 2 辺については X 座標が大きくなるほどメッシュが大きくなるように設定している。

次に, 面内が長方形のメッシュで分割されるように **Mesh Control** から **Mapped Face Meshing** を選択する。Outline の **Mesh** の下に **Mapped Face Meshing** の項目ができるのでそれを選択して, 正方形の面をクリックして左下に出てくる詳細の Geometry で **Apply** を押す。これで Geometry のところが 1 Face になり選択できている。ほかはそのままでよい。

最後に **Mesh** を右クリックして **Generate Mesh** を選べば図 8 のようなメッシュが引けているはずである。メッシュが Y 軸に平行になっていない場合は, X 軸に平行な 2 辺の Bias Type の選択が誤っているので再確認しよう。

メッシュ数について補足しておく。メッシュは細かいほど精度が高くなるが, メモリの使用が多くなる (基本利用が 100 MB でメッシュ数 100000 増加ごとにメモリ使用量 100 MB 程度増加が目安) のでそれだけ計算機に負荷もかかるし時間もかかる。メッシュ数に収束解が依存しなくなる最低限のメッシュ数にするのがよいが, これは経験的な感覚が必要である。

今回は長方形メッシュであるので, 四角形に歪みはないが, 形状が複雑になると歪んだ四角形や三角形のメッシュで計算領域を離散化する必要が出てくることもある。歪みの大きいメッシュを用いると計算の安定性や精度に悪影響を及ぼすこともある。そこで, 歪んだメッシュを用いる場合, メッシュの長方形・直方体からの歪み具合を調べておく必要がある。

メッシュ品質を調べるには, Outline で **Mesh** を選んでおき, 画面左下の詳細の一番下の **Statistics** で Mesh Metric で **Skewness** を選択する。Statistics には生成したメッシュ数も表示されている (ここでは 1000 のはず)。Skewness の値が 0.90 を超えるメッシュがあるようであれば, メッシュの切り方を工夫する必要がある。

### 4.2.3 計算領域の境界条件の種類指定

メッシュが引けたら、次は境界条件の種類を指定する。詳細設定は FLUENT に入ってから行う。境界条件の種類は、対応する境界を左クリックして選択してから同じ位置で右クリックを押すと出てくるメニューで **Create Named Selection** を選択し、現れる入力欄に境界条件の種類を名前として入れる。2次元平面の領域を計算するときには形状の辺 (Edge) に境界条件を設定するので、画面上部に並んでいる立方体の中に矢印があるアイコンで辺が黄緑色になっているところをクリックして形状内の辺を選択できるようにしておく。X=0 mm にある Y 軸に平行な辺を「**Velocity-inlet**」(速度を指定する入口)、X=10 mm にある Y 軸に平行な辺を「**Pressure-outlet**」(圧力を指定する出口)、X 軸上にある辺を「**Axisymmetric**」(正方形を円管にするための回転対称軸) と名づけておく。なお、**回転対称軸**は X 軸に平行に設置しないと FLUENT でエラーが出て計算できない。できあがった状態で Outline の Named Selections で Shift キーを押しながら名称を付けた部分をすべて選択したときの画面を図 9 に示す。設定した名前 (境界条件の種類) と位置の対応が一覧できる。

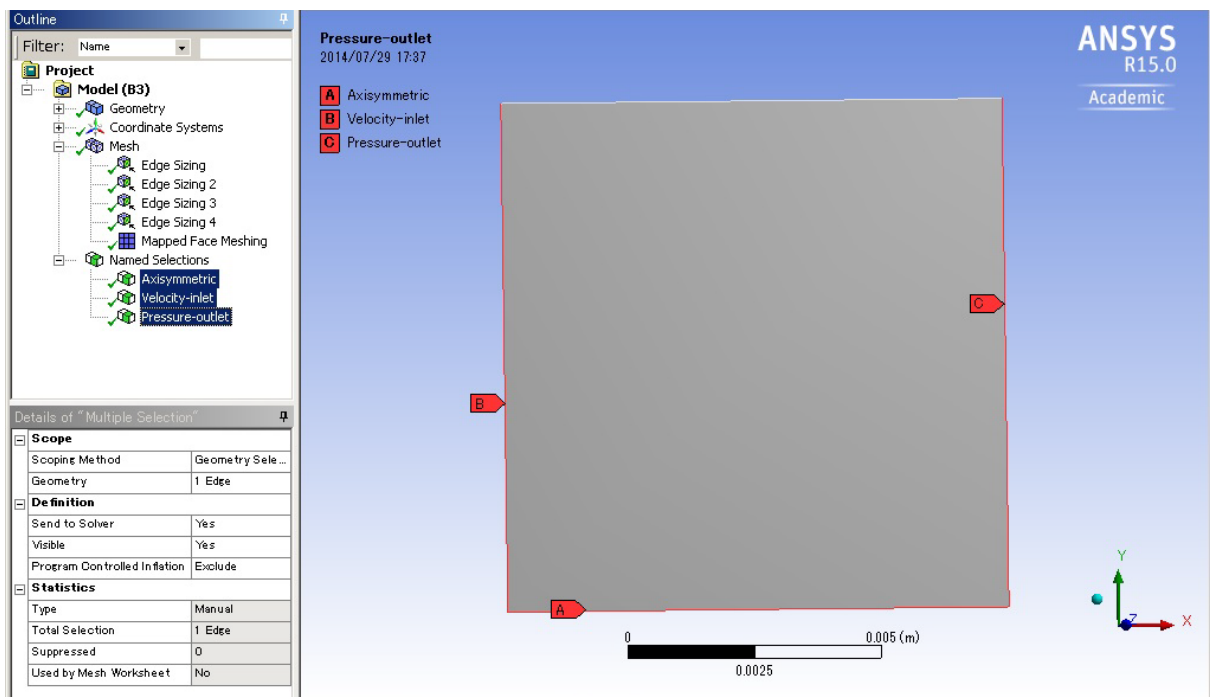


図 9 Meshing の設定終了時の画面

### 4.2.4 Meshing からのファイルのエクスポートと FLUENT での読み込み

Meshing で作成したメッシュを FLUENT で読み込む。最初に、Meshing の画面上の **File** から **Export** を選択し、**tutorial-tube.msh** で保存し、メッシュを FLUENT で読み込める形式に指定しておく。マイネットワークの現在 Meshing を使っているのとは別の PC 内に Export しようとしたら、ファイル保存に関連するフォルダ名に全角フォントが入っ

たりしていると Error が出て失敗するので注意しよう (保存するフォルダの上位フォルダに日本語が入っていてもダメ)。Workbench の Mesh のところは「？」だが、問題ない。

プログラムで **ANSYS** → **Fluid Dynamics** → **FLUENT** とたどり、起動時の画面で Dimension は **2D**、Options で **Double Precision** にチェックを入れて下部の **OK** をクリックすると FLUENT が倍精度で立ち上がる。 **File** → **Read** → **Mesh** で **tutorial-tube.msh** を選びメッシュファイルを読み込む。回転対称軸の設定がまだなので Warning が表示されるが問題ない。

※単精度よりも倍精度のほうが精度は高いが、dat ファイルサイズは倍になり、パソコンのメモリ負荷は上がり 1 回の収束計算は 20–30%遅くなる、ただし単精度よりも少ない収束計算で収束する傾向にあるのでトータルでは速くなることもある。以下の場合には倍精度使用が推奨される。

- ・混相流全般 (物性値のオーダーの差が大きい)
- ・マイクロスケールの計算 (スケールが小さい)
- ・作成したメッシュの体積差が大きい

#### 4.2.5 流れ・物質移動のモデルと物質の物性・境界条件の指定

- ・ まずは計算領域のサイズを実際に扱いたいサイズに変換する。画面左の Problem Setup の General → Mesh → Scale (または上部メニューの Mesh ☉ Scale)) を選んで現れる画面を見ると、DesignModeler でのサイズがそのまま FLUENT に読み込まれていることになる。実際のサイズは円管半径 100  $\mu\text{m}$  (y 方向のサイズ)、長さ 1 cm (x 方向のサイズ) であるから、Scaling で Specify Scaling Factors のラジオボタンをオンにして、縮尺の倍率 X : 1, Y : 0.01 を入れて Scale を押すとサイズが変換される。Close を押して Scale Mesh の画面を閉じる。
- ・ 左メニュー **Solution Setup** の **General** (または上部メニュー **Define** → **General**) で Solver にある 2D Space を **Axisymmetric** とする。これで回転対称軸が有効になる。
- ・ 次にモデルを指定していく。 **Solution Setup** の **Models** のところに **Viscous** とあるが、ここで層流か乱流かを選択する。デフォルトは層流であり、今回は層流を扱うのでこれについては **Laminar** のまま何もしない。
- ・ **Models** の **Species – Species Transport, Reactions** を選んで、選択欄の下にある **Edit** をクリックする。現れる小画面で **Species Transport** のラジオボタンを選ぶ。表示項目が増えるので Reactions の **Volumetric** にチェックを入れる。 **OK** を押してウィンドウを閉じるとまた警告が出るが **OK** を押してこれも閉じておく。
- ・ **Models** の Energy が自動で **On** になる (Species Transport モデルのときは Off にできない)。
- ・ **Solution Setup** の **Materials** で物質・流れの物性を決めていく。Fluid を選んだ状態で、選択欄の下の **Create/Edit** をクリックする。現れた画面右側の **FLUENT Database** から Material Type → **fluid**, FLUENT Fluid Materials → **water-liquid (h20<l>)** を選んで、下の **Copy** を押してから隣の **Close** で物質選択の画面を閉じる。今選んだ water-liquid が

表示されているので、Name と Chemical Formula を両方とも **a** として、下の **Change/Create** を押して、Change mixture and overwrite water-liquid? と聞かれるので「No」を選ぶ。これで原料の **a** が定義できた。生成物の **r** についても Name と Chemical Formula を両方とも **r** とした後に、下の **Change/Create** を押して、「No」を選んで定義する。これで Database に入っている水の分子量 (18.02 kg/kmol) ・密度 (998.2 kg/m<sup>3</sup>) で **a** と **r** が定義されたことになる。非等温系でさらに必要になる設定は付録 E に示している。(原系と生成系で分子量の和が同一になる必要がある (バランスしていないと警告が画面に出る)、反応式によっては分子量を調整する、たとえば、2A → B なら B の分子量は A の 2 倍にする)

- 次に対象としている系に含まれる化学種を指定する。今操作した画面で Material Type を **mixture** にして下の Properties の **Mixture Species** で **Edit** を押して編集画面に入り、先ほど作成した **a** と **r** を **Add** でリストに加え (**a** がリストの上に来るようにしておく)、最初からある 3 種の気体を **Remove** で除外する。最後に **OK** を押して編集画面を閉じる。
- 反応式を設定する (2a → 2r)。Properties の Reaction の **Edit** を押して、

Reactants 側…Species : **a**

量論係数 (Stoich. Coefficient) : **2**

反応次数 (Rate Exponent) **2**

Products 側…Species : **r**

量論係数 : **2**

としておく。さらに下の Arrhenius Rate で

Pre-Exponential Factor : **2.583e+17** (2.583×10<sup>17</sup>)

と入力して **OK** ボタンでウィンドウを閉じる。活性化エネルギーが自動で 10<sup>8</sup> J/kmol = 100 kJ/mol と設定されており、デフォルトの入口設定温度は 300 K なので、速度定数は  $k=1 \text{ m}^3/(\text{kmol}\cdot\text{s})$  と設定されたことになる。ここでの設定では原系と生成系の物性が同じ、つまり生成エンタルピーも同じなので、反応熱が 0 で 300 K の等温系になっている。

- なお、**Total Number of Reactions** の値を 2 以上にすれば複合反応も扱える。また、Species Model (Species Transport のモデルを選んだ画面) のところで **Wall Surface** にもチェックをいれておくと、この画面でも Reaction Type に Wall Surface が選べるようになり、壁表面での反応 (触媒反応に対応するとき用いる) を適用できるようになる。
- そのほかの物性を定義していく。Properties の下に行って、

Density : **volume-weighted-mixing-law**

(デフォルトの incompressible-ideal-gas のままだと気体の密度で計算され濃度が非常に低くなり反応がほとんど進まなくなるので注意)

Viscosity : **0.001**

Mass Diffusivity : **1e-9** (1×10<sup>-9</sup>)

と値を変更したら **Change/Create** を押して、**Close** で画面を閉じる。

- 境界条件の設定に移っていく。**Solution Setup** の **Cell Zone Conditions** では領域内の条件の設定を行う。今回は変更の必要がないのですぐに Boundary Conditions に移る。
- Solution Setup** で **Boundary Conditions** を選択するとすぐに右に境界の一覧が出てく



る。Meshing の Named Selections でつけた名前が並んでいる。設定しなかったところは自動で境界の種類に対応した名前になっている。

- まず **velocity-inlet** を選んでみよう。境界の一覧の下に Type の欄があり、境界条件の種類も velocity-inlet になっているはずだ。境界条件の種類はこの Type のリストを選びなおすことで変更可能である。このすぐ下にある **Edit** を押して現れるウィンドウの「**Momentum**」のタブで Velocity Magnitude を 1 [m/s] として入口流速を指定し、「**Species**」のタブの Species Mass Fractions で a の欄を 1 (a の入口濃度は密度/分子量×質量分率から決まる、Specify Species in Mole Fractions でモル分率での設定もできる) として **OK** のボタンを押してウィンドウを閉じる。ここでは成分 a の入力欄しか表れない (r をリストの先頭にした場合は r だけが現れることになる、前の項で作成した系に含まれる化学種のリストで最後になっているものは現れない)。今回は a と r の 2 成分系のため、a と r の質量分率の和が 1 であり、a の質量分率が決まれば自動的に r の質量分率も決まるためである。このように各成分の収支を質量分率ベースで FLUENT は計算するため、以降の設定画面でも同様にリストで最後になっている成分の設定は不要である。
- ほかの境界条件の設定はここでは不要である。pressure-outlet (Type も同一名) の面は何も設定しないデフォルトの状態だと、この出口面の圧力が 0 (ゲージ圧、絶対圧は基準圧力 101325 Pa を足した値) になる。基準圧力の値は Define → Operating Conditions → Operating Pressure のところから変えられる。FLUENT の収束計算とともに圧力場も計算され、入口での圧力の値が流路内の圧力損失に相当する。

#### 4.2.6 離散化スキームと収束計算のモニター法の選択

- FLUENT の画面左側で **Solution** に進み離散化スキームを選択する。**Solution Methods** で、Pressure-Velocity Coupling は **SIMPLE** (デフォルト)、Spatial Discretization は Momentum, a, Energy の収支式のところを **Second Order** とする (First Order よりも精度が高い)。化学種の収支式は a だけ表示される (化学種リストの最後の成分は現れない)。今回は a と r の 2 成分系で質量分率の和が 1 であり、a の質量分率が決まれば r も決まるためである。
- 次の **Solution Controls** で緩和係数 (3.5.1 節を参照) を設定する。今回はデフォルトの値をそのまま使う。緩和係数を入力する欄の下に Equations のボタンがあるのでこれを押すと、どの保存式を解くか選択できる。計算の難易度を下げるために一部の保存式だけに一度収束させてからすべての保存式を収束させたいときに有効である。ここでは Energy のチェックを外せば強制的に等温系にすることもできる。
- さらに収束判定の設定を行う。**Monitors** の Residuals, Statistic and Force Monitors のところで **Residuals - Print, Plot** を選択して **Edit** を押して編集画面を開く。Check Convergence のチェック (右側の列) をすべてはずす。3.5.3 節でも述べたように、デフォルトの残差の値では計算は収束していないことのほうが多いので、残差だけでは収束

判定はしない。y 方向の速度、Energy の Monitor もチェックをはずしてもよい。OK でウィンドウを閉じる。

- ・ 収束を注目している変数が変化しなくなった時点とするために、その変数の収束計算に伴う変化をモニターする必要がある。これを行うために、Monitors の Surface Monitors の白い枠の下の Create をクリックする。現れた編集画面で Report Type を Mass-Weighted Average、Report of のところで Species, Mass fraction of a、Surfaces で pressure-outlet を選択して OK を押す。これで収束計算ごとの、円管出口断面での a の質量平均の質量分率も FLUENT の画面右下に表示されるようになる。
- ・ 初期値を設定する。Solution Initialization で右に現れる Compute from を velocity-inlet に変更し、下の Initialize を押して初期条件を与える。
- ・ 次の Calculation Activity は収束計算の途中の値を保存する場合などに用いる (4.3.1 節参照)。今回は使用しないので設定はしない。
- ・ 最後に Run Calculation で収束計算 (iteration) の回数を指定して収束計算を行う。ここでは Number of Iterations を 200 回にしておく。Calculate のボタンを押せば収束計算が始まる。60 回程度計算すれば、出口の a の質量分率は変化しなくなる。入口付近から濃度分布は収束していく傾向があるので、出口の組成が変化しなくなれば、計算領域全体で濃度分布は収束していると考えてよい。収束計算中の FLUENT の画面を図 10 に示す。

ANSYS UI キャプチャー画面のため非公開

図 10 収束計算中の FLUENT の画面

- ・ 計算が終了したら Calculation Completed と表示されるので OK で閉じる。File → Write → Case & Data を選んで FLUENT の設定を含む tutorial-tube.cas と計算結果に対応する tutorial-tube.dat を保存する (.cas と .dat が同時に保存される)。

#### 4.2.7 結果の整理

まずは原料濃度の等高線図を見ていく。FLUNET 画面左の Results の下の Graphics and Animation をクリック (または上部メニューの Display @ Graphics and Animation) し、その右側の Graphic のところで Contour を選んで Set Up のボタンを押す。表示されるウィンドウで Option の Filled をチェックして Contours of で Species, Mass fraction of a を選んで Display を押すと a の質量分率が表示される。マウスの左ボタン押したままマウスを動かすことで水平移動、中ボタンを下に動かして長方形を描くとその範囲が拡大される。上に動かして長方形を描くと縮小される。Contour を表示した状態で下部にある Scene のボタンでウィンドウを開いて Contour を選択して (contour-domain-a といった名前を選ぶ)、Transform で等高線図の配置・回転・縮尺を定量的に変更できる。図 11 は Contour の例である。

Contour の画像を文書に貼る場合、Contour のウィンドウ上部の紺色になっている部分 (図 11 のメニューが出ていたあたり) で右クリックして Copy to Clipboard を選択して文書



で Ctrl+V として貼り付ける。グレースケールにしたければ、**Page Setup** を選んで開いた画面で **Gray Scale** のラジオボタンを選択し（デフォルトでは **Color**）→ **OK**（これを押さないと反映されない）→ **Close** → **Copy to Clipboard** を選択して文書に貼り付ける。

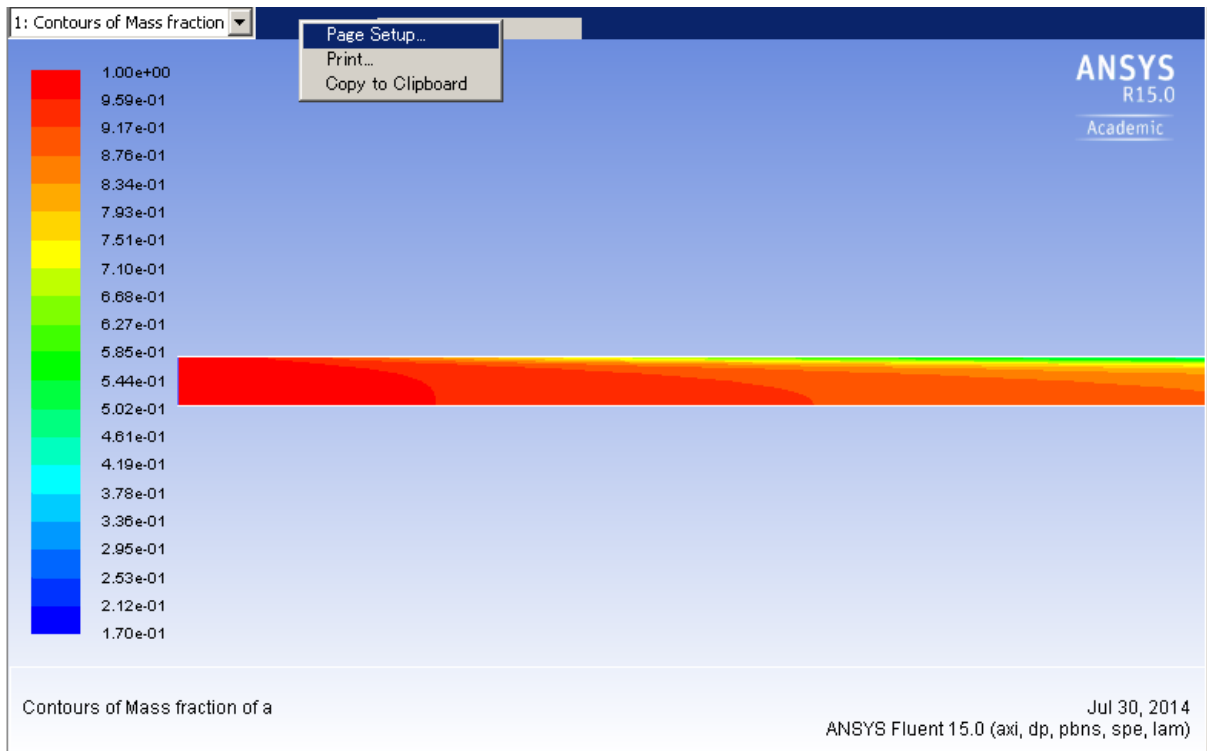
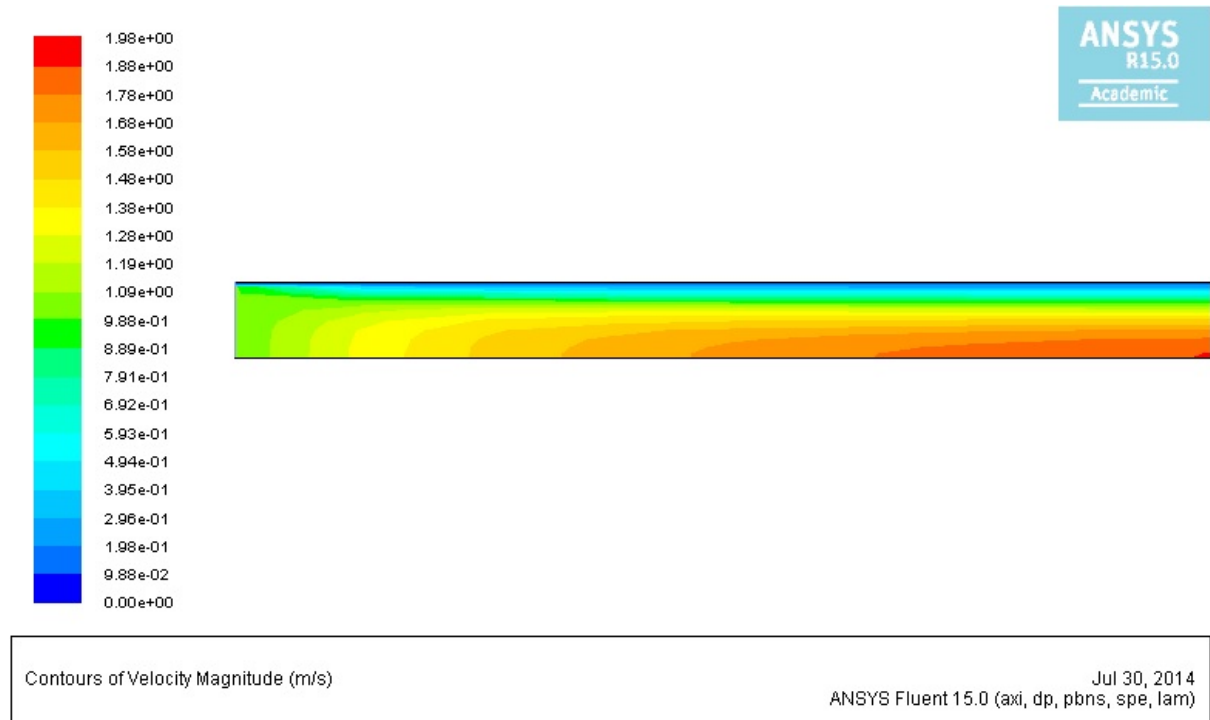


図 11 a の濃度分布

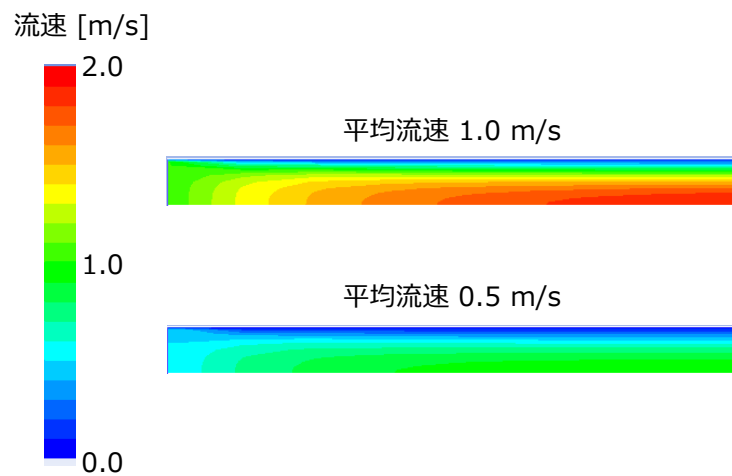
#### ※Contour を文書で示すときの注意点

(1) スケール 読者が見やすいようにスケールの変数名と単位、範囲の上限・中央・下限値をテキストボックスで示すようにする。具体例を挙げて操作方法を説明する。図 12(a)のように入口付近の速度分布の発達を見たいとする (Contours of で **Velocity, Velocity Magnitude**)。これを先の段落の方法でキャプチャーし、二つコピーする。一方をスケールだけになるようにトリミングし、もう一方を極力 Contour だけにする。図 12(b)のように、スケールの上部に変数名と単位（ここでは流速 [m/s]）、スケール右側の上端・中央・下端に流速の上限・中央・下限値を加える。

(2) 範囲の選択 異なる計算条件で得られた Contour を並べて比較したいときは、スケールの範囲を統一すると比較しやすくなる。たとえば、ここで実施した計算を入口の平均流速を 0.5 m/s に変えて計算し直し、流速の分布を Contours の画面で Auto Range にチェックを入れたまま表示すると、流速の最高値（赤色になる値）は 1.0 m/s になる。これを図 12(b)のように、入口の平均流速 1.0 m/s の結果と合わせて示したい場合、Auto Range のチェックを外し、Max の値を 2.0（チェックを外すと入力できるようになる）にして最大値を大きい側にあわせて表示させた Contour を使えばよい。



(a) 速度分布の Contour (FLUENT から Copy to Clipboard で取り出したそのまま)



(b) Contour のスケールを見やすくしたもの

図 12 Contour を見やすくする方法

計算結果の数値をプロットしたい場合は、画面左の **Results** の下の **Plots** をクリック (または上部メニューの **Display** → **Plots**) し、右側に表示される Plots の中の **XY Plot** を選択して **Set Up** をクリックする。ここでは例として、回転軸上での a の質量分率の変化をプロットしてみる。クリックして出てきたウィンドウ右側の Y Axis Function で **Species, Mass fraction of a** を選び、Surfaces を **axisymmetric** として Plot を押すと図 13 のようなグラフが表示される (先の段落の方法で画像を抽出している)。数値データとして取り出してほかのグラフソフト (Excel や Igor) でプロットしたい場合は Options で **Write to File** にチェックを入れる。すると、**Plot** のボタンが **Write** に変わるので、このボタンを押してファイル名を指定すればテキストファイルとしてデータを取り出せる。

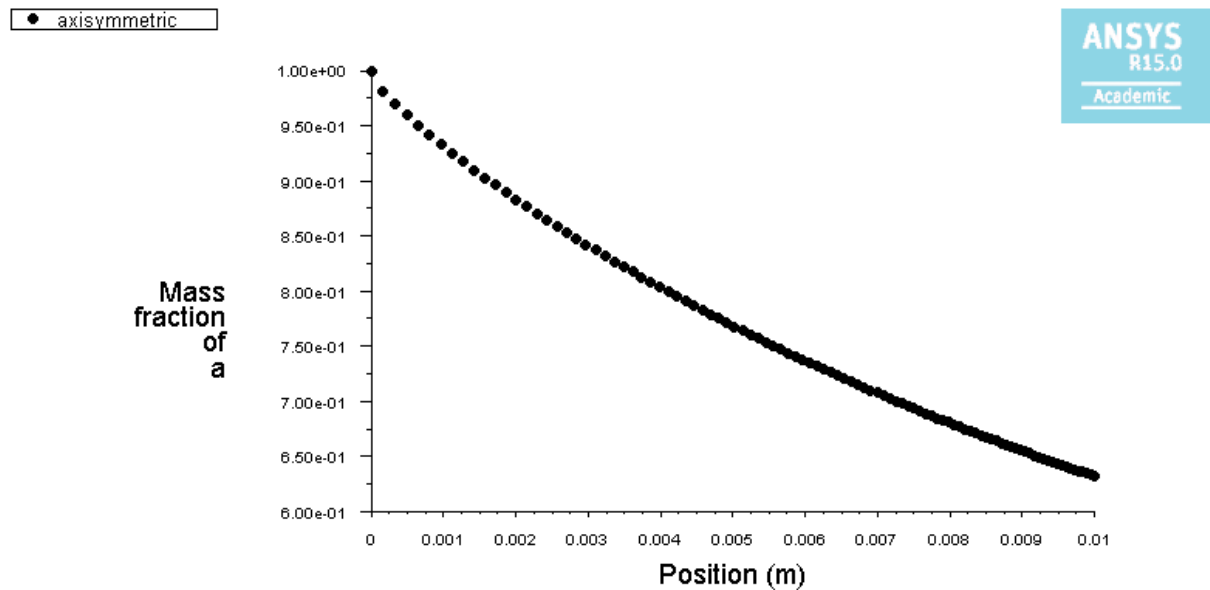


図 13 a の回転対称軸上での質量分率の変化

さらに軸方向に断面平均がどのように変化するかを追跡したい場合がある。この場合は、計算領域に Line または Plain（結果を取り出すために設置するものであり、物理的には流れに影響を及ぼさない）を設定する。今回は 2 次元なので line を設定してみる。画面上部メニューの **Surface** → **Line/Rake** と選んで出てくるウィンドウで 2 点の座標を指定すると、その間にある線分上の変数の統計量を扱える。たとえば、 $x = 0.001$  m に今回の円管の幅全体にラインを引く場合は、 $x_0 : 0.001$ ,  $x_1 : 0.001$ ,  $y_0 : 0$ ,  $y_1 : 0.0001$  と入力して Create → Close とすればよい。Surface name は任意だが、ここでは o0.001 としておく。不要な Line を消したいときは下部の Manage をクリックして消したい表面の名前を選んで Delete をクリックする。

上記の方法で Line は引けるが、内部の状態を詳しく見るには、Line を多く引く必要がある。しかし、この入力を何度も繰り返すのは面倒である。FLUENT には Journal File に作業を記述しておき、そのファイルを読み込む (**File** → **Read** → **Journal**) とその作業を FLUENT で行ってくれる。 **File** → **Write** → **Start Journal** でファイル名を指定すると、それ以降の作業内容をファイルに書き込んでくれる。書き込みをやめてファイルを保存するときは **File** → **Write** → **Stop Journal** とする。

では、実際にやってみよう。 **File** → **Write** → **Start Journal** で line1.jou と指定して、Journal の作成を開始する。  $x = 0.00002$  m にラインを引くために、 $x_0 : 0.00002$ ,  $x_1 : 0.00002$ ,  $y_0 : 0$ ,  $y_1 : 0.0001$  と入力し、New Surface Name を o0.00002 としておく（ライン名の先頭の文字はアルファベットでないと入力を受け付けてもらえない）。ウィンドウを閉じ **File** → **Write** → **Stop Journal** で Journal 作成をやめる。line1.jou をテキストエディタで見ると以下のようにになっている（WordPad で見ないとうまく改行されて表示されない）。ウィンドウを動かすことも作業として記録されてしまうのでこれより中身は多いかもしれない。

```
(cx-gui-do cx-activate-item "MenuBar*SurfaceMenu*Line/Rake...")
(cx-gui-do cx-set-real-entry-list "Line/Rake Surface*Frame4(End
Points)*Table1*RealEntry1(y0)" '(Q))
(cx-gui-do cx-set-real-entry-list "Line/Rake Surface*Frame4(End
```

```

Points)*Table1*RealEntry4(y1)" '( 1e-004)
(cx-gui-do cx-set-real-entry-list "Line/Rake Surface*Frame4(End
Points)*Table1*RealEntry1(x0)" '( 2e-005)
(cx-gui-do cx-set-real-entry-list "Line/Rake Surface*Frame4(End
Points)*Table1*RealEntry4(x1)" '( 2e-005)
(cx-gui-do cx-set-text-entry "Line/Rake Surface*TextEntry6(New Surface Name)" "o0.00002")
(cx-gui-do cx-activate-item "Line/Rake Surface*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-activate-item "Line/Rake Surface*PanelButtons*PushButton2(Cancel)")
(cx-gui-do cx-activate-item "MenuBar*WriteSubMenu*Stop Journal")

```

網掛けの部分をコピーアンドペーストして太字下線部 ( $x$  座標と線の名称) を Line の位置に合わせて変えていけば, 多数の Line を引く Journal File ができあがる (付録 C.1 参照).  $y$  座標は不変のなので最初の 1 回の入力だけするように 2~5 行目に記述がある. なお, CFD シミュレーションの設定操作も保存できる. この Case Study のメッシュを読み込んだ直後から収束計算直前までの作業を記録した Journal File を付録 C.2 に示している.

次に画面左メニューの **Results** → **Reports** → **Surface Integrals** → **Set Up** で

Report Type : **Mass-Weighted Average**

Field Variables : **Species, Mass fraction of a**

Surfaces : 作成した **Line** すべてと **pressure-outlet**

を選び **Compute** を押すと, 次のように各自軸方向位置での断面質量平均の指定量が計算される (Write でテキストファイルに出力もできる). 収束計算の過程が表示されていた領域に結果が出てくるので, これを選択して右クリックでコピー (Ctrl+C ではできない) して Excel に移せば, 軸方向の長さ (滞留時間) と A の反応率 (1 - 質量分率) の関係をプロットできる. 反応器出口での a の反応率は 0.480 であるが, plug flow では 0.525 になる. 層流による滞留時間分布により反応率が低下していることがわかる. 得られた結果をうのみにせず, 化学工学で学んだことを活かして直感的におかしくないか判断するようにしよう.

Mass-Weighted Average  
Mass fraction of a

o0.0000025	0.98576039
	(中略)
o0.009	0.54432178
pressure-outlet	0.51992708
	未反応分なので a の反応率は 0.480
-----	
Net	0.87954247

ここまで終わったら **File** → **Write** → **Case & Data** でもう一度 case と data を保存しておく.

### 4.2.8 Mesh Adaption

前節までで収束解は得られたが、収束していれば正しい解とは限らない。メッシュの切り方に依存しない解になっていることを、メッシュをさらに細かくして収束解が変化しないことで確認する必要がある。3.6 節でも述べたように FLUENT ではある変数の勾配がある一定以上のメッシュを検出し、そのメッシュだけを細かくする Mesh Adaption を行うことができる。質量分率勾配による Adaption から再度収束計算を行うまでの手順を以下に示す。

- FLUENT の画面上部メニューの **Adapt** → **Gradient** で Gradients of のところを a の質量分率 (**Species..., mass fraction of a**) に表示を切り替えて Adaption をかける基準となる勾配の変数として指定する。
- ここで下の **Compute** を押すと、指定した変数の勾配の最小値と最大値が算出される。Refine Threshold には Adaption をかけるメッシュの勾配最小値（つまりこの値以上の勾配をもつメッシュに Adaption がかかる）を入れる。この最小値は先の最小値と最大値の間の値を入れる。Adaption をかける a の濃度勾配の値を小さくするほど、Adaption の対象になるメッシュ数は大きくなる。一度に多くのメッシュに Adaption をかけると発散することもあるので、少しずつ Adaption を何段階に分けて行ったほうが安定して計算がしやすくなる。Case Study の話に戻って、ここでは Refine Threshold に  $1e-7$  と入力しよう。
- **Adapt** を押すと、メインの画面に 442 cells marked for refinement, 0 cells marked for coarsening と表示されて 442 のメッシュに Adaption がかかることが計算され、「Ok to change mesh?」という表示が現れて Adaption するか聞かれるので「Yes」を選択する。Adaption のかかるメッシュが  $x \cdot y$  方向にそれぞれ 2 分割されるのでこれで 4 分割される、つまり Adaption のかかるメッシュ数の 3 倍の分だけ総メッシュ数が増加する。これで a の質量分率の勾配による Adaption ができたことになる。
- メッシュがどのように変わったかを確認しておく。画面上部メニュー **Display** → **Mesh** と選択していく。出てきたウィンドウの Surfaces で **axisymmetric, interior-surface\_body, wall -surface\_body** を選択して **Display** を押すと図 14 のように Adaption の結果を見ることができる。管壁付近のメッシュが細かくなっていることがわかる。
- 再び **Solve** → **Run Calculation** で収束計算を 200 回ほどしてみよう。モニターの値を見ればわかるが、A の反応率は 0.479 となり、Adaption する前よりわずかに下がった程度である。非常に高い精度を求めるのでなければ、最初のメッシュでもある程度正しい結果が得られることがわかる。

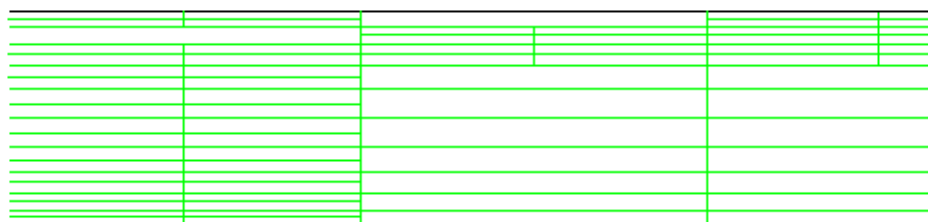


図 14 a の質量分率の勾配による Grid Adaption (入口付近)

なお、この Adaption では変数の勾配だけでなく、**Adapt** → **Region** で、計算領域の  $x \cdot y \cdot z$  座標（3次元のときだけ  $z$  座標の指定必要）を指定して **Adapt** ボタンを押すことで、指定された座標で囲まれた範囲のメッシュだけを細かくすることもできる。領域を指定する場合、デフォルトでは同じ領域は 2 度までしか Adaption できないことを留意しておこう。それ以上はメニューから **Adapt** → **Controls** で **Mesh Adaption Controls** を開き、**Max Level of Refine** の数（デフォルトは 2 になっている）を増やしてから再度 **Adapt** すればよい。

#### 4.2.9 ユーザー定義関数の導入

今の例では入口の速度分布は均一であったが、管入口から層流の速度分布が発達した状態にもできる。入口の境界条件をユーザー定義関数（User Defined Function, UDF）で定義すればよい。円管層流の速度分布  $U(r)$  は

$$U(r) = 2u(1-(r/R)^2) \quad (63)$$

で表される。ここで  $r$  は円管中心からの距離、 $u$  は平均流速である。付録 D.1 に平均流速・管半径がこの Case Study に対応した速度分布を円管入口に与える UDF を示した。この UDF の設定の仕方は、**Define** → **User-Defined** → **Functions** → **Interpreted** でファイル名を指定して UDF をコンパイルする。そして、入口の境界条件の速度を入力した欄の右側の constant のところに UDF の関数名を選択できるようになっているので、その関数名を選択すればよい。ほかにも物性の温度による関数が異なる場合（付録 D.2）や計算領域内の変数の標準偏差を求める UDF（付録 D.3）を付録 D に示してある。関数名、UDF ファイル名や保存フォルダ名に半角・全角のスペース、全角文字、-（ハイフン）が入っているとコンパイルに失敗する。

### 4.3 非定常計算の Case Study

4.2 節では定常計算を扱ってきたが、FLUENT では非定常計算も扱える。反応工学と関連して滞留時間分布（Residence Time Distribution, RTD）を求めるためにステップ応答で過渡応答を求める場合がある。ここでは、前節の円管層流の速度分布による滞留時間分布を求める場合についてシミュレーションする Case Study を扱いながら、非定常計算や結果の整理の方法について示していく。非定常計算のほかの例として、付録 F に Volume of Fluid 法による混相流（スラグ流）再現の方法も載せている。

#### 4.3.1 非定常計算の設定

円管内の層流による式(63)による速度分布によって、流路中央の流れは速いため早く円管から出て行き、管壁付近の流れは遅いので滞留時間が長くなり、幅方向に RTD が生じる。ここではこの RTD を CFD の非定常シミュレーションで求めていく。ここでも前節の円管の Case



File をそのまま用いていく。以下に手順をまとめる。

- FLUENT を **2D, Double Precision** で立ち上げる。
- **File** → **Read** → **Case** とたどり **tutorial-tube.cas** を読み込む (ファイルをダブルクリックしてもよい)。
- 非定常計算を行うために FLUENT 画面の左のメニュー-Problem Setup から **General** → **Solver** → **Time** のところを **Transient** にする。
- 上部メニュー-**Solve** → **Solution Methods** か左メニューの **Solution** → **Solution Methods** で Transient Formulation を **First Order Implicit** にしておく (一般的に 2 次精度のほうが精度は高いが解の経時変化が不安定になりやすいのでまずは 1 次にする)
- 上部メニュー-**Solve** → **Initialization** か左メニューの **Solution** → **Solution Initialization** で a の初期値を **0** にして Initialize する。これで、入口流れの a の質量分率が、時刻 0 s に 0 から 1 にステップ変化することになる。
- 次に、今回は反応を扱わないので、左メニューの **Materials** → **Mixture** → mixture-template とたどって **Create/Edit** を押し、Properties の Mechanism のところで **Edit** を押し、reaction-1 のチェックを外しておく。これで反応が計算から除外される。**Close** で Material の画面を閉じる。
- さらに、成分 a の滞留時間分布を計算するために反応器出口での a の経時変化を保存していく必要がある。これを行うためには、まず左メニューの **Monitors** → **Surface Monitors** で **Create** をクリックする。出てきたウィンドウで Options の **Write** にチェックを入れ、File Name は **a-unsteady.out** (このファイルに過渡応答が保存されていく)。X Axis は **Time Step**, Get Data Every の欄は **Time Step** にしておく。右側は以下のようにする。

Report Type : **Mass-Weighted Average**

Field Variable : **Species, Mass fraction of a**

Surfaces : **pressure-outlet**

File Name : **a-unsteady.out** (このファイルに過渡応答が保存されていく)

この方法だと各 Time Step での円管出口の a の組成しか記録されない。一定間隔の Time Step ごとに data file を保存して計算結果全体を保存したい場合は、上部メニューで **File** → **Write** → **Autosave** (左メニューで Calculation Activities からたどってもよい) と進んでいき、**Save Data File Every (Time Step)** に保存する Time Step の間隔を入力すればよい。

- 次に非定常計算の収束計算の方法を指定して計算を開始する。左メニューの **Run Calculation** で、

Time Step Size : **1e-4**

Number of Time Steps : **300**

Max Iterations/Time Step : **20**

として Calculate を押すと収束計算が開始される。0.0001 s 刻みで、300 ステップ、つまり時刻 0.03 s まで計算することになる。平均滞留時間は 0.01 s なので、この 3 倍の長さの時間の非定常計算を行ったことになる。6000 回の収束計算をすることになる。途中の時刻のデータも残したい場合は、**Solution** → **Calculation Activities** → Autosave Every (Time step) を入力すればよい。今回の例で 0.01 s 刻みで残したい場合は 2000 とする。

※途中の時刻のデータ保存の注意点を挙げておく。直前に別の case ファイルを扱っており、FLUENT を立ち上げたまま続いて新規の case で始めて途中時刻データを自動保存する場合、保存するファイル名が直前に扱っていた case ファイル名がベースになってしまうことがある。Calculation Activities のファイル名が示されている欄がこれから計算しようとしている case 名と一致しているか確認してから計算を始めるようにしよう。

### 4.3.2 計算結果の整理

前節の設定の下、非定常計算を行った結果と解析解を図 15 に示す。出口の a の質量分率経時変化はそのまま  $F$  関数になる。 $F$  関数と滞留時間分布関数  $E$  には次の関係がある。

$$F(t) = \int_0^t E(\tau) d\tau. \quad (64)$$

また、 $F$  関数の解析解は

$$F(t) = 1 - (\bar{t}/t)^2 / 4, \quad t \geq \bar{t}/2, \quad F(t) = 0, \quad t < \bar{t}/2 \quad (65)$$

である。ここで  $t$  は滞留時間、 $\bar{t}$  は平均滞留時間である。CFD での解は解析解に近いものの振動が少しあり不安定である。また、 $F(t)$  が立ち上がり始める時刻が CFD のほうが早いのは数値拡散によるものと考えられる。

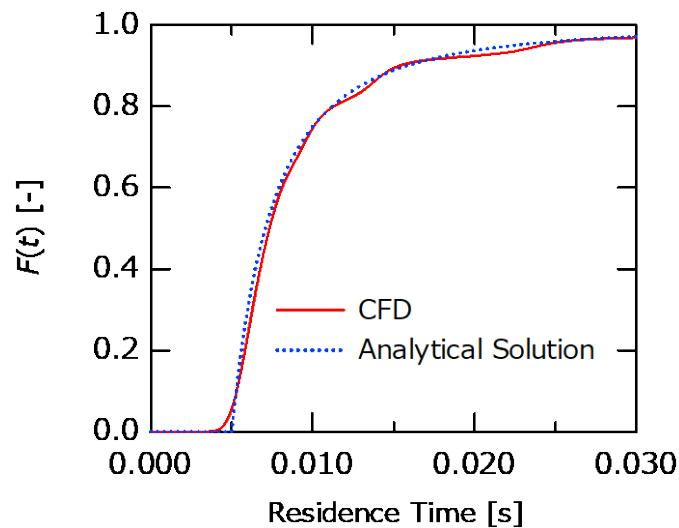


図 15 非定常 CFD シミュレーションの結果と解析解の比較

反応器内の濃度分布の変化をアニメーションで見ることが出来る。

- 左メニューで **Solution** → **Calculation Activities** → **Solution Animation** とたどり、その下部の **Create/Edit** ボタンを押してウィンドウを開く。
- Animate Sequences の値を **1** にし、アクティブになった sequence-1 の右側の Every を **2**、When を **Time Step** にして、**Define** を押す。これで 2 Time Step ごとのスナップショット



トの画像が.hmf ファイルに保存されていく。

- Define を押して表示される画面右側の Display Type で **Contours** を選び、さらに現れるウィンドウで **a** の質量分率を選んで、**Display** を押して FLUENT 右上に表示されている 1 番のウィンドウに **a** の質量分率が表示されたのを確認して **Close** を押す。Animation Sequence, Solution Animation のウィンドウも **OK** を押して閉じる（ファイルを上書きしてよいか聞かれたら Yes としておく）。Sequence-1 の左側に Active のチェックが入る。
- 先ほどと同様に非定常計算を行う。Time Step 数は **200** にしておく。
- 計算終了後、左メニューの **Results** → **Graphics and Animations** を選び、Animations の欄の **Solution Animation Playback** を選択して **Set Up** をクリックする。Playback の画面が現れるので sequence-1 を選択して右向きの横三角のボタンを押すと、**a** の質量分率の経時変化のアニメーションを見ることができる。再生のスピードは、再生ボタンの下の **Replay Speed** の bar で調節できる。
- また、このアニメーションを mpeg 形式で保存することもできる。Playback の画面の Write/Record Format で **MPEG** を選択して **Write** を押せばアニメーションが自動で再生され、FLUENT 画面上に編集状況が表示されて sequence-1.mpeg としてアニメーションが一つの mpeg 形式のファイルに収録される。この mpeg 形式のファイルを作ってしまうと.hmf ファイルは不要なので消去してもかまわない。mpeg 形式ではファイルサイズは小さくなるが、30 fps（Replay Speed は無関係、1 秒で 30 枚の画像が流れる）であるため、画像数が少ないと再生が非常に速くなってしまふことがある。

※アニメーション用のデータの保存先がシミュレーションを行っている PC と同一でないとデータ保存時に固まってしまうことがある。シミュレーションを行う PC のハードディスクから case ファイルを読み、mpeg ファイルを作成するようにしよう。

※ある case ファイルでアニメーションを作成した後、そのファイルの計算条件を変えて異なる case ファイルとして連続して別のアニメーションを作成する場合、case ファイルをわざと読み込みなおしてアニメーションの設定を最初からすること。

### 4.3.3 初期条件の与え方の応用編

Initialize で初期条件を与えると、計算領域内に均一に変数の初期値が設定されてしまう。この値を領域によって異なる状態で与えたいときもある。ここでは例として、今回の反応器で入口 5 mm は **a** だけ、後半の 5 mm は **r** だけといった初期条件を与える操作を説明する。

- 上部メニューの **Define** → **Custom Field Function** で Field Functions で **a** の質量分率を選び、左の **1** のボタンを押す。すると Definition に **1** と表示される。New Function Name に **mass\_frac\_a** と入れて **Define** のボタンを押すと **a** の質量分率が **1** であるという関数が定義される。**Close** ボタンで画面を閉じる。
- 次に、質量分率を適用する領域の座標を定義する。上部メニューの **Adapt** → **Region** で **X : 0-0.005, Y : 0-0.0001** で範囲を指定して **Mark** を押す。ここで左の **Manage** を押す。

Manage Adaption Registers のウィンドウが開き、今指定した領域が hexahedron-r0 という名前になっている。このウィンドウの下にある **Display** を押すと FLUENT 画面右上の表示が変わり、指定した領域が赤になっていることがわかる。

- Manage Adaption Registers, Region Adaption の画面を閉じる。
- **Solution** → **Solution Initialization** → **Initialize** で初期化する。
- Initialize の右側の **Patch** が選択できるようになるので、これをクリックして現れる画面で Variable : a, Use Field Function のチェックを **On** にして **mass\_frac\_a**, Registers to Patch で **hexahedron-r0** を選択して下の **Patch** ボタンを押す。 **Close** で画面を閉じる。
- **Graphics and Animations** → **Graphics** → **Contour** で a の質量分率が図 16 のようになる。

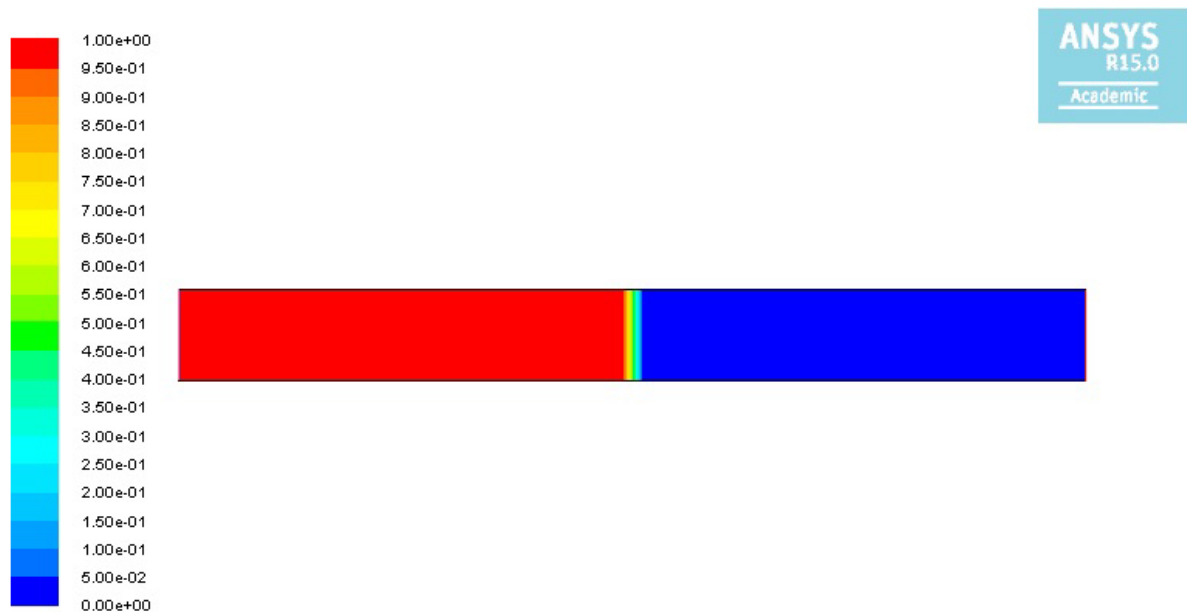


図 16 非定常 CFD 初期濃度分布の Patch の例

#### 4.3.4 非定常計算のコツ

最後に、非定常計算のコツ・注意点を述べておく。Time Step が小さいほど、またメッシュが細かいほど数値拡散（数値計算の誤差により分子拡散上に拡散が進んでしまう）による誤差は小さくなる。収束計算を安定させるために  $\text{Time Step} \leq \text{メッシュサイズ} / \text{代表速度}$ 、 $\text{タイムステップ} \leq \text{メッシュサイズの 2 乗} \div (2 \times \text{拡散係数})$  を満たしているか確認する。

1 タイムステップあたりの収束計算回数は、序盤はとくに収束しにくいので、必要に応じて大きくする必要がある。例では 20 回としたが、50–100 回程度はとると確実である（残差の変化を見て調整する、 $10^{-7}$  のオーダーになっていればよい）。

軸方向分散モデル[6]で換算して  $D = 4 \times 10^{-9} \text{ m}^2/\text{s}$  程度の数値拡散は起きてしまうようである（定常計算でも同程度の数値拡散は起きる模様）。非定常計算では、Solution

methods の Transient を 2 次精度にするのが望ましいが、今回の例でも見られたように過渡応答に振動が見られ、計算が不安定になることもある。

非定常計算には時間がかかることが多い。計算速度を少しでも速くするために Run Calculation にある Reporting Interval を "1"ではなく"100"などとして表示回数を減らすと、100 回の iteration 後に表示のプロセスを行うため表示にかかる負荷が激減する。

また、逆流のメッセージが FLUENT の text prompt に表示されている場合もこの表示時間に時間がとられている場合がある。この場合、上記設定もしくは以下のコマンドを使用して逆流のメッセージを表示しないようにすると計算速度が改善される。

**solve set flow-warnings?** (text prompt で左記を入力)

enable flow warning messages? [yes] **no** (細字の部分が出てくるので no とだけ入力)

上限値に達してしまうエラーを表示したくないときは以下を text prompt で入力する。ただし、計算終了時に設定を戻して上限値に達していないかを確認しよう。

**solve set limiter-warnings? No**

## 4.4 乱流の Case Study

これまでの例では流れは層流モデルを使ってきた。ここでは、FLUENT に備えられた各種乱流モデルで流れの数値解析を行っていく。

### 4.4.1 Case Study の設定

4.2 節で用意した case ファイルを開き、左メニューで **Problem Setup** → **Models** → Viscous と選択していき、**Edit** を押して、**k-epsilon (2 eqn)** ( $k$ - $\epsilon$  モデル) を選択する。 $k$ - $\epsilon$  モデルは乱流モデルの中で最も一般的なモデルであり、乱流を扱う場合はまずはこのモデルを選べばよい。入力項目が増えるが変更せずに下の OK を押す。

管径を大きくしてレイノルズ数を乱流域にする。左メニューで **Problem Setup** → **General** → Mesh の **Scale** のボタンを押す。表示された画面で **Specify Scaling Factors** のラジオボタンを選択してから Y 方向の縮尺に **100** と入力後 **Scale** のボタンを押すと管半径は **1 cm** になる。

境界条件については、**Problem Setup** → **Boundary Conditions** → Zone で **velocity-inlet** (流路入口に相当する部分) を選んで **Edit** をクリックし、「**Momentum**」タブの下部の Turbulence で Specification Method を **Intensity and Hydraulic Diameter**, Turbulent Intensity を **1[%]**, Hydraulic Diameter を **0.02 [m]** にしておく。**OK** で画面を閉じる。

左メニューで **Solution** → **Solution Methods** とたどり、乱流に関する Spatial Discretization の方法を **Second Order Upwind** にしておく。**Solution** → **Monitors** → Residuals - Print, Plot と選択して **Edit** をクリックし、k と epsilon の Check Convergence のチェックを外して **OK** を押して画面を閉じる。

さらに、**Solution Initialization** の表示画面で値を変更せずに **Initialize** をクリックして初期条件を与える。**Run Calculation** で収束計算回数を **2000** 回として **Calculate** をクリックして収束計算を行う。収束したらファイル名 (**tutorial-tube-ke.cas**) をつけて保存する。

乱流モデルを使用しているときは、層流のときよりも計算は収束しにくい傾向にあるので収束計算回数を多くしている。乱流計算では出口での流れが発散しやすく (reversed flow の表示がでることがある)、収束計算の序盤では Turbulence に関連する緩和係数を小さくして、出口での reversed flow が出なくなればしばらくしたら緩和係数をデフォルトの値に戻すとよい。

今度は、上部メニューを **Define** → **Model** → **Viscous** とたどり、**Reynolds Stress (5 eqn)** (レイノルズ応力モデル) を選択する。方程式数はこちらのモデルのほうが多く、より厳密なモデルである。ただし、方程式数が多い分計算にはより時間がかかる。

**Solution Methods** の Spatial Discretization で Reynolds Stresses のところを **2nd Order Upwind** にしておく。Residuals - Print, Plot のところでの **uu-stress, vv-stress, ww-stress, uv-stress** の **Check Convergence** のチェックを外す。収束計算回数を 3000 回にしておく。収束したらファイル名 (tutorial-tube-RS.cas) をつけて保存する。

#### 4.4.2 各乱流モデルによる結果

まず、各モデルでの円管出口での a の反応率を求めてみる。左メニューで **Results** → **Reports** → **Surface Integrals** とたどり、Report type を **Mass-Weighted Average**, Field Variable を **Species...**, **Mass fraction of a**, Surface を **pressure-outlet** にして **Compute** を押すと円管出口での a の質量分率が出てくる。1 からこの質量分率を引いたものが求める反応率である。円管出口での a の反応率は、 $k-\epsilon$  モデルでは 0.422, レイノルズ応力モデルでは 0.427 となる。層流と乱流モデルでは結果が大幅に変わりうるがこのことからわかる。Contour で円管内の速度分布も比較しておこう。**Graphics and Animations** で Vectors や Pathlines (入口断面を指定して底から出た流れがどのような流線に沿って流れていくかを表示) を選択して流れの向きを表示することもできるので試してみよう。

### 4.5 熱交換器のシミュレーションと周期境界条件の設定

4.2-4.4 節では円管という単純な形状をもとに CFD シミュレーションでできる主なことを一通り経験してもらった。実際研究で使う場合は、形状も設定したい境界条件もより複雑なものになるし、エネルギー収支も含む計算になることもある。ここでは ANSYS の利用法をより深めるために、前節までの例で出てこなかった発展的な計算領域形状描画と周期境界設定、非等温計算の方法を熱交換器のシミュレーションを通じて学ぶ。

#### 4.5.1 問題設定

図 17 に示す入口温度 293 K の水と入口温度 353 K の水がそれぞれ幅 2 mm の流路を 0.1 m/s で向流に流れる熱交換器をここでは想定する。周期的に温水と冷水の流路が繰り返し配置されているものとする。周期境界条件は周期の始まりと終りに設置すればよいので、ここで示した設置位置は一例ということになる。厚さ 1 mm の銅を材質とする壁が温水と冷水

を隔てており、この壁を通じて熱交換が起こる。図中の入口に示した名称は Meshing で境界に名前をつける (=境界条件の種類を決める) ときに使う。DesignModeler で形状を描く際、水が流れる流路は流体で満たされるので Fluid 領域に、壁は固体からなるので Solid 領域に指定する。流れに平行な向きを  $x$  方向、垂直な向きを  $y$  方向とする。

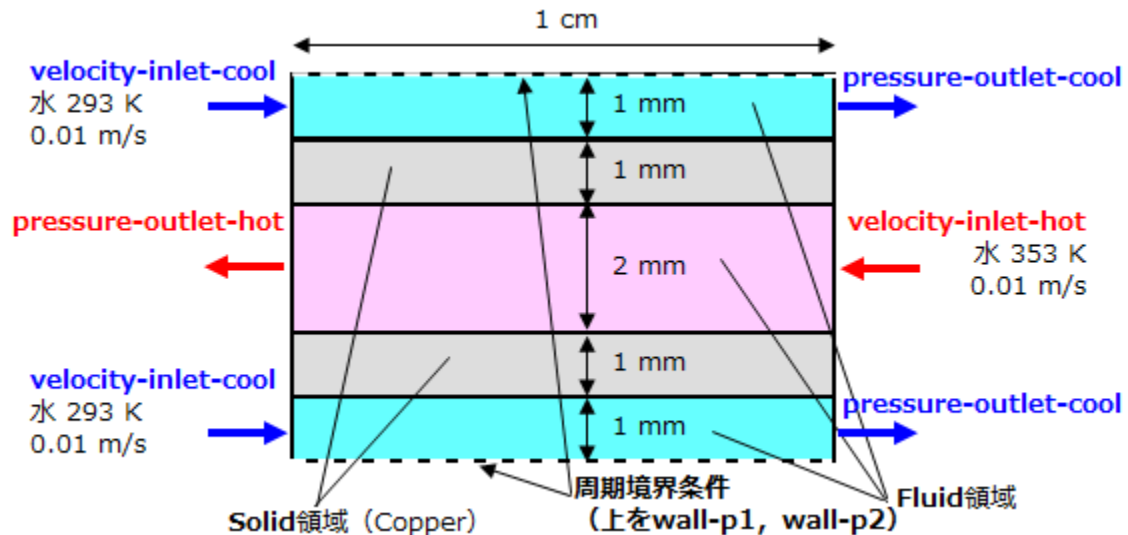


図 17 熱交換器シミュレーションの設定

#### 4.5.2 形状の描画とメッシュ作成

- Workbench を立ち上げる。ファイル名を **tutorial-heat-exchanger.wbpj** として保存する。
- DesignModeler を立ち上げ、単位の選択は **millimeter** にしておく。
- Tree Outline で **XYPlane** をクリックして選択してから、ツールバーの右上にある **Look at Face/Plane/Sketch** のボタン（人が黄緑色の面を見ている）を押して XY 平面 ( $Z = 0$ ) を表示する。この面に流路を描いていく。**Zoom** ボタン（虫眼鏡の中に+）をクリックして、図形が表示されている領域で左クリックの状態でもうすを上下させると拡大縮小できる。
- 描画に失敗したときはツールバー2段目の **Undo** で作業を戻せる。なお、Undo はスケッチの作業に対してしか使えない。
- 「**Sketching**」のタブに切り替え、Settings の Grid の二つのボックスにチェックを入れ、Major Grid Spacing を **5 mm**、Minor-Steps per Major を **5** とする。
- マウスの右ボタンをクリックした状態で原点付近を四角で囲み、ボタンを離すことで原点付近を拡大させる。Minor の Grid が表示されているようにする。
- 図 16 の左下が原点になるようにして、この図で指定したサイズに合わせて **Sketching Toolboxes** → **Draw** 内の **Rectangle** で長方形を五つ描く。長方形の左上の頂点と右下の頂点で2回クリックするとスケッチが確定する。これで XYPlane の下に Sketch1 が完成する。
- 上部メニューの **Concept** → **Surfaces from Sketches** をクリックする。SurfaceSK1 が Tree Outline に表示されるのでこれを選択する。



- ・ 左下に表示される SurfaceSK1 の Details View が表示されたら，Tree Outline で **Sketch1** をクリックして，Details View 2 項目の Base Object で **Apply** のボタンを押す。
- ・ Details View の 3 項目にある Operation を **Add Frozen** に変更しておく。

※隣接している別個の領域を描きたいときは Operation の変更が必要となる。Add Material のまま Generate すると五つの長方形の形状が連続した一つの領域として認識されてしまう (FLUENT 備え付けの Tutorial では折れ曲がり流路に細い円管を描き足したときに自動で連続した流路になったのはこのため)。

- ・ Tree Outline で **SurfaceSK1** を右クリックして現れるメニューで Generate を選択する。これで五つにわかれた長方形の領域が作成できた。Tree Outline の一番下に **5 Parts, 5 Bodies** と表示されているはずである。
- ・ 5 Parts, 5 Bodies の左の+ボタンをクリックして五つの領域が Tree Outline に見えるようにする。五つの Surface Body が出てくるので，左下の Details View の Fluid/Solid で図 17 と対応するように流体が通る領域を **Fluid** に変えておこう。選択されている領域は画面に右側で黄色くなる。
- ・ Shift キーを押しながら五つの Surface Body を選択した状態にする。この状態で右クリックして出てくるメニューに **Form New Part** があるのでこれを選択すると 5 Parts, 5 Bodies が **1 Part, 5 Bodies** に変わる。
- ・ 上部メニュー **File** → **Export** で形状を Export する。ファイル名は tutorial-heat-exchanger.agdb とする。
- ・ 以上で DesignModeler での作業は終了である。この時点で図 18 のようになっている。Workbench で **Save** しておく。

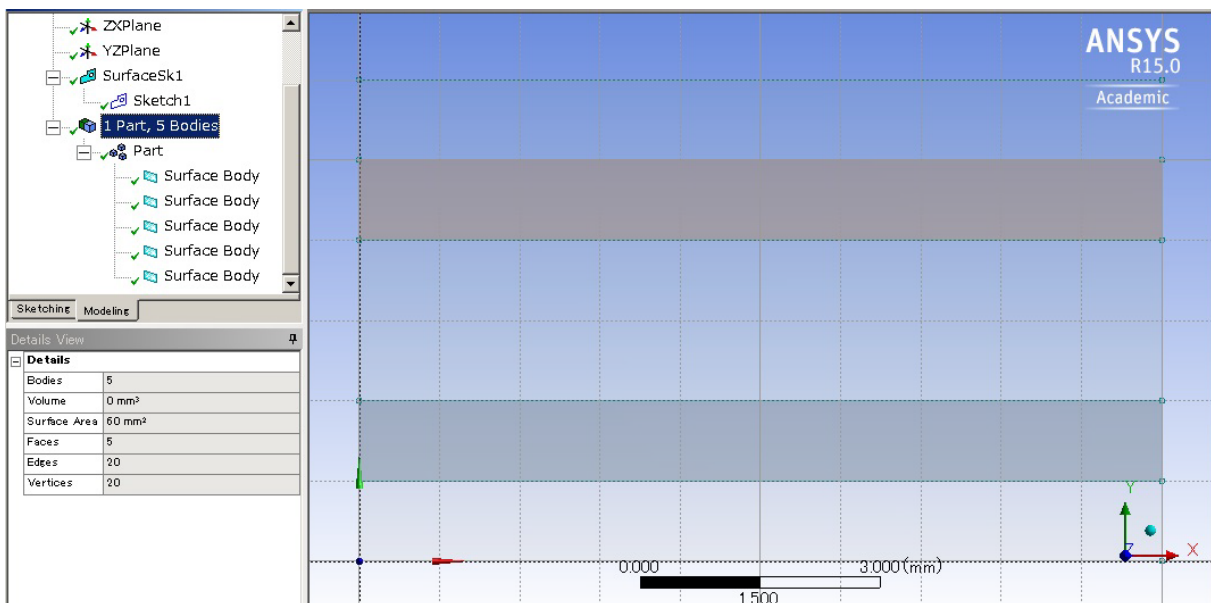


図 18 熱交換器形状描画終了時の画面

※最後の Form New Part の設定は必須である。同一 Part 内でしか熱のやりとりは起こらない仕様のため、これをしないと壁を通じた熱伝導が FLUENT で再現できなくなる。今回のシミュレーションでは流体と壁が接する面は壁の端でもあり、流体の端でもある。このように二重の性質をもつ壁を Double-Sided Wall とよぶ。FLUENT 内に形状・メッシュを読み込んだ際にも通常の壁とは異なる認識がソフト側でなされる。

### 4.5.3 メッシュ作成と境界条件の種類の設定

- Workbench で Toolbox の Mesh を画面左側に Drop し、先程作成した形状がファイル **tutorial-heat-exchanger.agdb** を取り込んでから **Mesh** をダブルクリックして Meshing を起動する。
- 右側の Meshing Option は変更しないのでそのまま閉じてよい。
- 辺が選択できるよう **Edge** ボタン (矢印が描かれた立方体で辺が黄緑にハイライトされたアイコン) を押しておく。
- まず流れに平行な方向のメッシュの間隔を設定する。画面左の Outline で **Mesh** を右クリックし、**Insert** → **Sizing** を選択する。
- Outline の **Edge Sizing** をクリックし、Ctrl キーを押しながら、流れに平行な方向の六つの辺すべてを選択し、Geometry のところの **Apply** ボタンを押す。
- 左下の詳細設定に以下のように入力する。入力が終了すると自動で辺上に黄色い点で辺が区切られる。
  - Type : **Number of Divisions**
  - Number of Divisions : **200**
  - Behavior : **Hard**
  - Bias Type : 中央の線が長いもの
  - Bias Factor : **3**

※周期境界条件を設定する対になる境界上のメッシュは数、サイズともに完全に同一になっている必要がある。このためこれらの辺は一度に選んで同時に設定するのが確実だ。

- 次に流れに垂直な方向のメッシュの間隔を設定する。画面左の Outline で **Mesh** を右クリックし、**Insert** → **Sizing** を選択する。
- Outline の **Edge Sizing** をクリックし、Ctrl キーを押しながら、流れに垂直な方向の 10 個の辺をすべて選択し、Geometry の **Apply** ボタンを押す。
- 左下の詳細設定に以下のように入力する。
  - Type : **Element Size** (1 メッシュの平均サイズを指定する)
  - Number of Divisions : **0.0001** (→自動で 1.e-004 m に表示が変わる)
  - Behavior : **Hard**
- Outline で **Mesh** を右クリックし、**Insert** → **Mapped Face Meshing** を選択する。

- ・ Outline に新しくできた **Mapped Face Meshing** を選択しておき、形状が描かれている画面右側で Shift キーを押しながら五つの面領域をクリックする。左下の詳細画面の Geometry の **Apply** ボタンを押す。
- ・ Outline で **Mesh** を右クリックして Generate Mesh を押す。領域が長方形メッシュで分割される。この時点で図 19 のようになる。
- ・ 次に境界条件の種類を設定していく。辺が選択できるよう **Edge** のボタンを押しておく。
- ・ 図 17 にしたがって、左クリックで辺を選択してから右クリックで **Create Named Selection** で現れる欄に境界の名前を入力していく。同じ名前の境界の辺は Ctrl キー+左クリックで二つとも選択してから名前をつけると二つの辺を同一名（一度に FLUENT でも設定することになる）にできる。
- ・ 境界条件の種類の設定終了時の状態は図 20 のようになる。 **File** → **Export** でファイル名を **tube-heat-exchanger.msh** としてメッシュファイルを作成する。

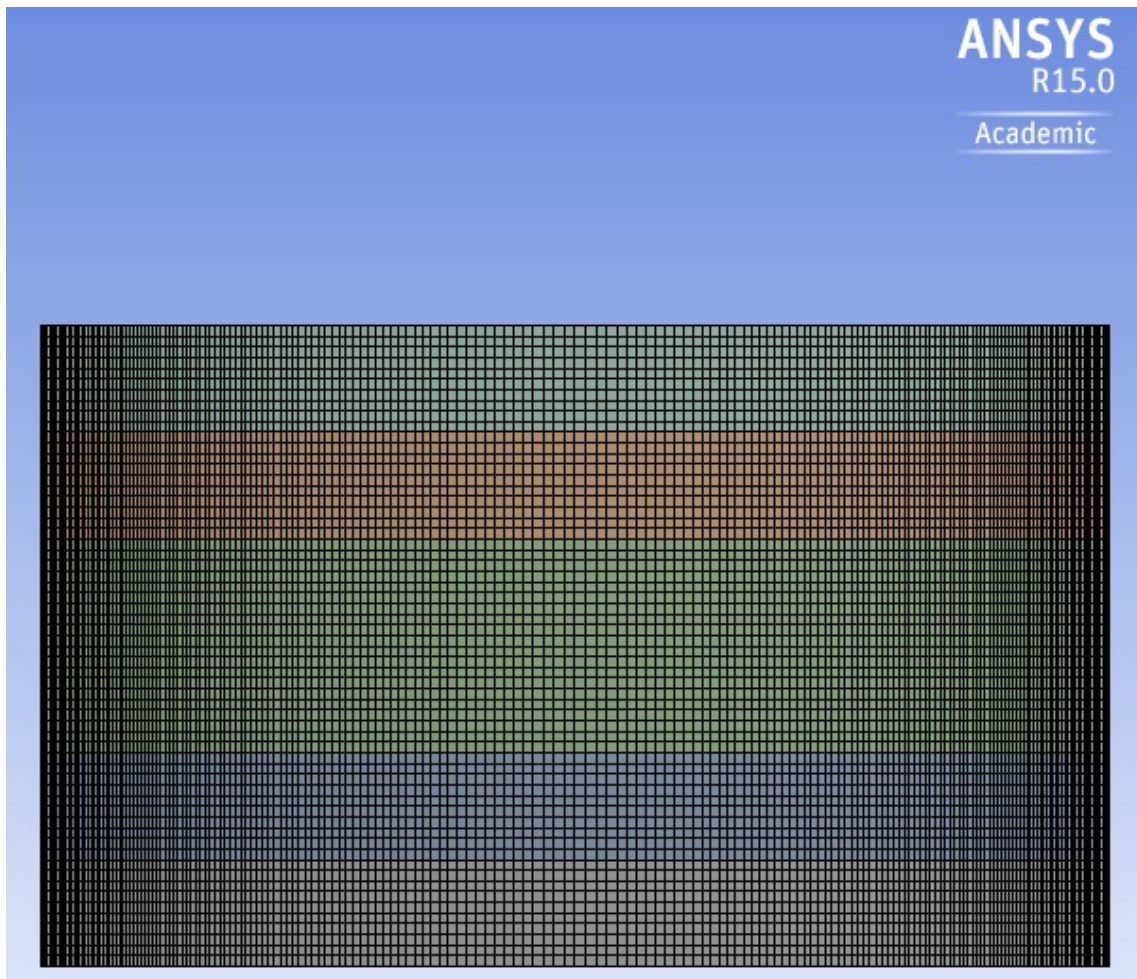


図 19 熱交換器のメッシュ



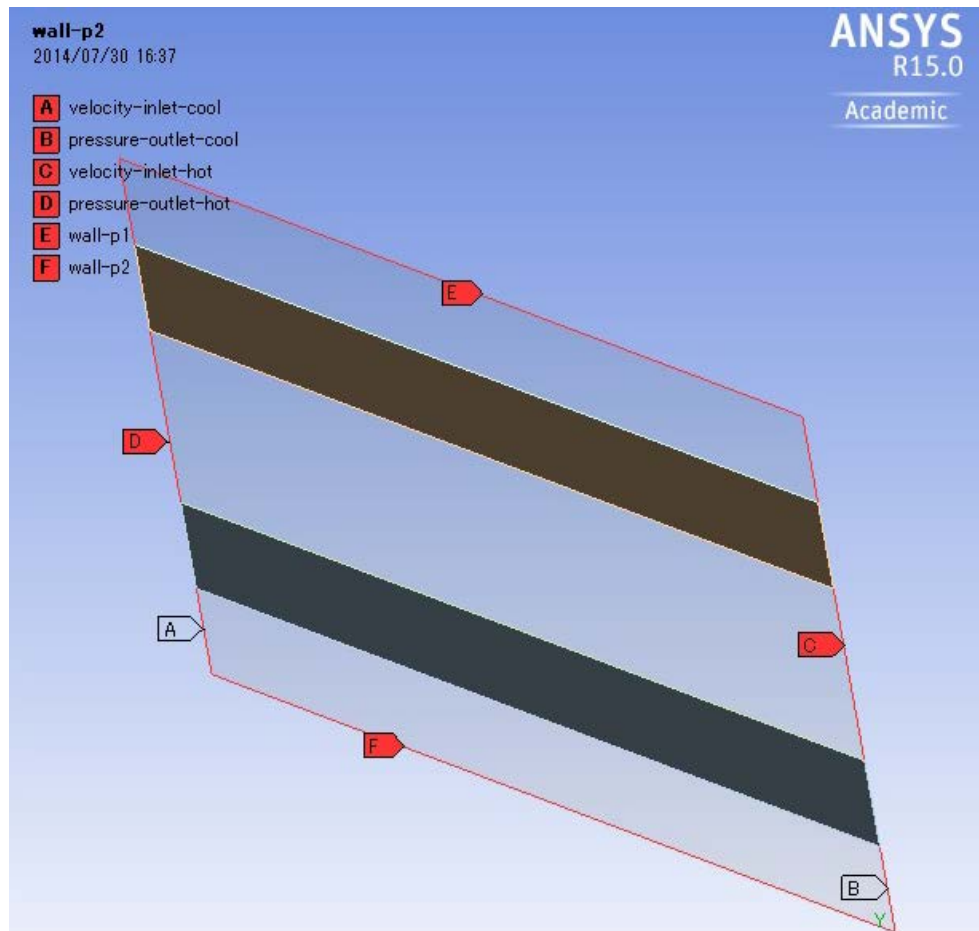


図 20 熱交換器の境界条件種の設定

#### 4.5.4 周期境界の設定方法

FLUENT での作業に入っていく。Meshing では入口と出口の境界条件の種類は設定したが、周期境界条件の設定がまだである。周期境界については FLUENT に入ってからでないと設定ができない。以下の手順で wall-p1 と wall-p2 (FLUENT で読み込んだ時点ではもちろん Type は wall) と名付けた 2 辺を周期境界に設定する。

- FLUENT を 2D, Double Precision で立ち上げ、画面上部の **File** → **Read** → **Mesh** で **tube-heat-exchanger.msh** を読み込む。
- **Problem Setup** → **Boundary Conditions** の Zone のリストで「-shadow」を含む名前をついた境界条件があることを確認しておく。ここではともに壁と流体領域の境界にある **part-part-surface\_body** と **part-part-surface\_body-shadow** があるはずだ。これがない場合は Double-Sided Wall が上手く認識されておらず、これまでの設定に何か問題がある。
- Zone の中に **wall-p1** と **wall-p2** があるはずである。これらを選択したときにリストの下部にこの境界の ID が表示されるのでこれをメモしておく。
- FLUENT の画面右下のカーソルが点滅しているところで Enter キーを押す。するとこの部分に > が表示される。> の右側にスペースを開けずに次の黄色でマーキングした太字

の部分を入力していく。二つ数字を入れているのは先にメモした周期境界の ID である。二つに分かれていた境界が統合されるため、Zone には一方の周期境界だけが残っているはずである。

> **mesh/modify-zones/make-periodic**

Periodic zone [()] **11**

Shadow zone [()] **12**

Rotational periodic? (if no, translational) [yes] **n**

Create periodic zones? [yes] **y**

Auto detect translation vector? [yes] **y**

computed translation deltas: 0.000000 -0.006000  
all 200 faces matched for zones 11 and 12.

zone 12 deleted  
created periodic zones.

>

#### 4.5.5 モデル・物性・境界条件・収束計算の設定と結果

境界条件設定以降の FLUENT 内の設定は以下のような流れになる。

##### Solution Setup

- **Models** で Energy を On にする。
- **Materials** で FLUENT Database から **water-liquid (h2o<l>)** と **copper (cu)** をコピーする。
- **Cell Zone Conditions** で Zone のリストから **part-surface body** 選択してこの名前のところでダブルクリックして編集画面を開く。Material Name を **water-liquid** に変更する。
- Solid-part-surface body の編集画面で Material Name を **copper** に変更する。
- **Boundary Conditions** で Zone のリストから **part-part-surface\_body-shadow** を選んで Edit ボタンで開く。Thermal のタブで Thermal Conditions が Coupled になっていることを確認する。固液の境界である Double sided wall の shadow 側はこれになっていないと流体と壁の間の熱伝導が正しく計算できない。
- Zone リストの velocity-inlet-cool の編集画面を名前のダブルクリックか **Edit** ボタンで開き、「**Momentum**」のタブで Velocity Magnitude を **0.01** [m/s], 「**Thermal**」のタブで Temperature を **293** [K] とする。velocity-inlet-hot も「**Momentum**」のタブで Velocity Magnitude を **0.01** [m/s], 「**Thermal**」のタブで Temperature を **353** [K] とする。

##### Solution

- **Solution Methods** で Momentum と Energy の Spatial Discretization を **Second Order Upwind** にする。
- **Monitors** の Residuals – Print, Plot で Check Convergence のチェックをすべて外す。
- **Monitors** の Surface Monitors の下の **Create** ボタンを押す。出てきた画面で Report Type

を **Mass-Weighted Average**, Field Variable を **Temperature, Static Temperature**, Surfaces を **pressure-outlet-cool** として冷水の出口温度をモニターしながら収束計算を行う。

- **Solution Initialization** で Compute from を all-zones にして **Initialize** のボタンを押す。
- **Run Calculation** で Number of Iteration を 300 にして **Calculate** のボタンを押して収束計算を行う。

## Results

- **Graphics and Animations** → **Graphics** → **Contours** を選択して Set Up ボタンを押す。表示される画面で Option の **Filled** にチェックを入れ、Contours of を **Temperature, Static Temperature** を選んで **Display** のボタンを押す。図 21 のような温度分布が得られる。
- 図 21 の温度分布をみると壁の内部がほぼ均一な温度になっている。一般には流路が小さくなると比表面積が大きくなり伝熱効率は上がる。しかし、軸方向の熱伝導も速くなり、とくに今回採用した銅のような熱伝導率が高い物質を壁に用いると壁内部の温度が軸方向に均一になってしまい、向流型と並流型の熱交換効率に差がなくなってしまうことがある。場合によっては熱伝導率の低い材料が効率的な熱交換を可能にすることもある。

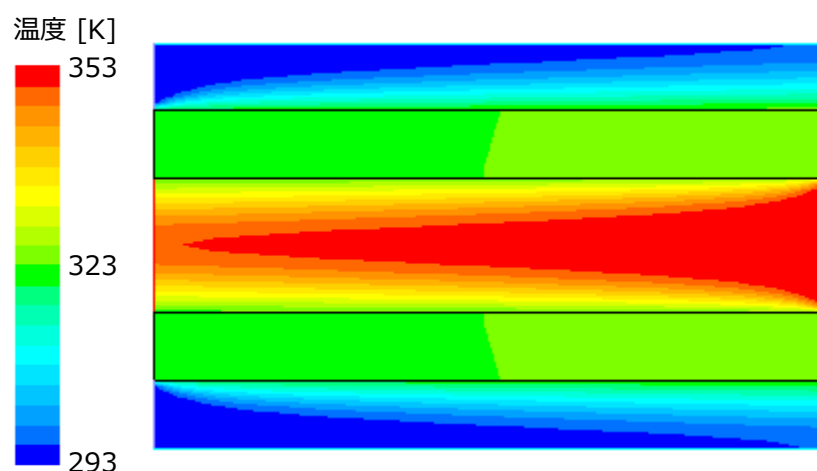


図 21 熱交換器内部の温度分布

## 4.6 DesignModeler・Meshing の使い方の補足

[4.2-4.5節](#)で触れられなかった複雑な形状・メッシュに対応するための方法をここでは示す。

### 4.6.1 形状から形状を切り取るまたは形状を融合して新たな複雑な形状にする方法

計算領域の形状が複雑になると、長方形や円などの基本形状を使うだけでは描ききれないことがある。そこで、形状から形状を切り取る、または形状を融合してより複雑な形状を描くことが有効になることがある。ここでは例として二つの円を用いてみる。

まずは「**Sketching**」タブの **Draw** の中にある **Circle** を用いて図 22(a)のように円を二つ描

く、サイズは任意で構わない。二つの円の一部分が重なるようにしよう。次に Sketching Toolboxes にある **Modify** の **Trim** を選択し、Ignore Axis をチェックする。右側の円の左側の円と重ならない円弧上でクリックするとその部分が消え図 22(b)のようになる。

Undo を押して二つの円がすべて表示された状態に戻って、今度はこの二つの円を一つの領域にする。Modify の Trim を選択した状態でお互いに相手の円と交差している円弧のところを4回クリックすると図 22(c)のようになる。

さらに左側の円のうち右側の円と重ならない領域だけを残したい場合は、右側の円の左側の円と重ならない円弧と左側の円の右側の円と重なる領域になる円弧をクリックすれば図 22(d)のようにできる。

二つの円が重ならない領域の円弧をクリックすれば、図 22(e)のように二つの円が重なった部分だけが残る。

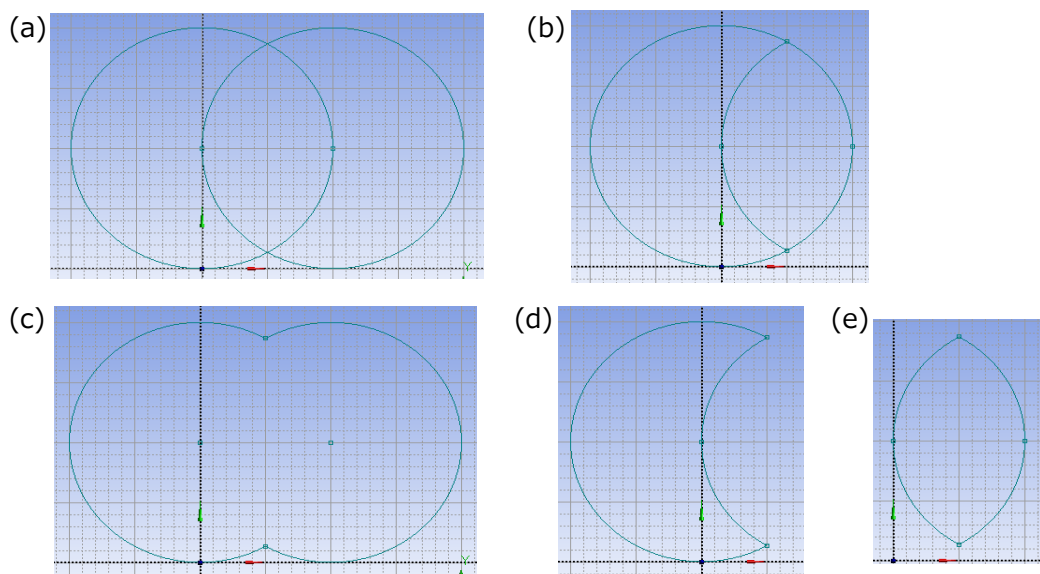


図 22 二つの形状から新たな形状の領域を作成する例

#### 4.6.2 スケッチの各部位のサイズを指定する方法

4.5 節までの例ではスケッチの各辺のサイズはマウスで位置調整していた。しかし、この方法だと、サイズの微調整ができない。サイズを細かく指定するときは、スケッチをしてからサイズを調整すればよい。「Sketching」タブの **Dimension** の中にある **General** を選択した後にサイズを指定したい部位をクリックして左下の **Details View** の **Apply** のボタンを押す。図 23 の例のように、部位の近くに V1, V2, H3 と記号が加わるので、**Details View** の **Dimensions** に部位の対応する記号の欄に指定したいサイズを入力する。

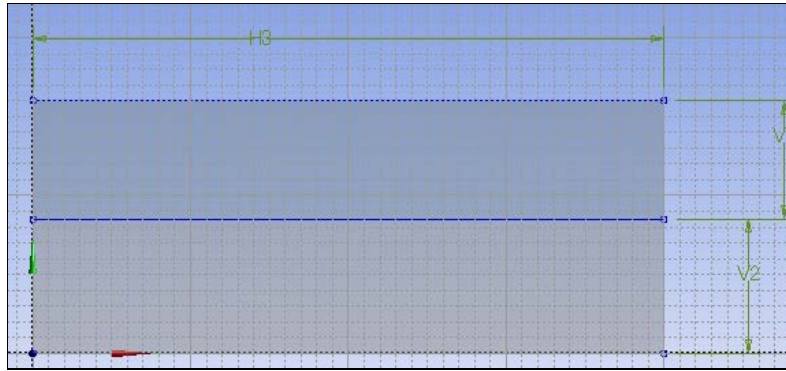


図 23 スケッチの各部位のサイズ指定

#### 4.6.3 点の座標を手動で指定してから形状描画を進める方法

形状が複雑な場合、座標の指定を詳細にしたい場合は、スケッチのマウス操作が向かない場合がある。このようなときは点の座標を手動で指定して打ち（上部メニューの **Create** → **Point**）、点から辺を（**Create** → **Lines From Points**）、辺から面を（**Create** → **Surfaces From Edges**）、面から立体を（**Create** → **Surfaces From Faces**）形成することになる。このとき、座標を逐一 DesignModeler 上で入力しなら点を打つと点数が多くなると大変になるため、座標の情報を記入したテキストファイルを読み込むようにするとよい。読み込み方は、上部メニューの **Create** から **Point** を選び、左下に出てくる Details View の中に **Coordinate File** の欄があるので、この欄をクリックして「…」のボタンを押すとファイルを選択する画面が出てくる。Tree Outline に **Point1** 出ているところで右クリックして **Generate** とすると点が表示される。座標情報のファイルには各座標について一行を使って  $abxyz$

を記述する。変数の意味は以下のとおりである。

$a$ : グループ番号 (1 以上の整数)

$b$ : ID 番号 (1 以上の整数)

$x, y, z$ :  $x, y, z$  座標 (単位は DesignModeler の設定に依存)

注意点:

- ・「#」で始まる行はコメント行となる
- ・空の行は無視される
- ・グループ番号と ID 番号の組み合わせは、同じになってはならない

たとえば次のようになる。ファイル名の拡張子は txt でよい。2次元のときは  $z$  座標をすべて 0 にしておけばよい。

```
# Group 1
11000
12010
13020
14030
```



1 5 1 3 0  
1 6 1 2 0  
1 7 2 2 0  
1 8 2 1 0  
1 9 3 1 0  
1 1 0 4 1 0  
1 1 1 4 0 0  
1 1 2 3 0 0  
1 1 3 2 0 0  
1 1 4 1 0 0

この手順で面を作ると Line Body と Surface Body が別の Part として生成してしまう（面を一つ作ると 2 Parts, 2 Bodies と表示される）。この状態で Meshing の Named Selection で境界条件の種類を設定しても FLUENT で認識されない。また、Meshing の辺に対する Sizing, Mapped Face Meshing の効果も現れない。面を作成した直後に DesignModeler の Outline で Line Body と Surface Body を選択して、右クリックで **Form New Part** を選んで 1 Part にしておく。

#### 4.6.4 複雑な形状でのメッシュの切り方

3.6 節でできるだけ多くの領域を歪みのない長方形（3D なら直方体）メッシュで切ることを勧めたが、CFD シミュレーションの対象とする形状が複雑になると形状を描画する時点で工夫が必要になることがある。ここでは実際に対象とする形状をわざと細かく分割して長方形メッシュで切られた領域を増やす例を示す。

まず、DesignModeler を立ち上げ（単位系はどれでも OK）、図 24(a)の座標に従って XYPlane 上に「Sketching」タブの Draw の中の最初にある Line で台形を頂点が  $(x, y) = (0, 0), (0, 10), (10, 10), (15, 10)$  となるように描く。Line では始点と終点でクリックすると直線ができる。**Concept** → **Surfaces from Sketches** でこのスケッチを選択して右クリックのメニューで **Generate** を選んで面にする。この形状を Meshing に読み込み、各辺を図 24(a)中の太字で示した数（底辺が 30 で残りの 3 辺は 20）に分割する（**Mesh Sizing** と **Mapped Face Meshing** を利用）。分割は等しい間隔にしておく（Mesh Sizing の詳細で Bias Type を No Bias のままにしておく）。メッシュを生成させると図 24(b)のようになる。少し歪んだ四角形メッシュで台形内が埋め尽くされる。

再び DesignModeler に戻り、今度は正方形（Draw の Rectangle を使う）と直角三角形（Line）で元の台形を図 24(c)のように描いてみよう。なお、Polygon で  $n = 3$  とすると正三角形しか描けない。**Surfaces from Sketches** で面にする際は、左下の Details View の Operation を **Add Frozen** にしてから行う（Add Material では別々の面にならない）。面が生成すると Tree Outline の一番下に **2 Parts, 2 Bodies** と表示される。二つの Surface Body を選択した状態から右クリックで **Form New Part** で一つの Part にまとめておく。この形状を Meshing に読み込み、辺上の分割数・間隔を先と同じになるように **Mesh Sizing** と **Mapped Face**

**Meshing** の設定を行う。正方形の底辺は **20** 分割、直角三角形の底辺は **10** 分割すればよい。メッシュを生成させると図 24(c) のようになる。この形状では、図 24(d) のように、右の三角になっている部分とそれ以外の領域に分けることで長方形メッシュを用いて切られた領域の割合を増やせる。

なお、この例では正方形と直角三角形の間に新たに辺が生じている。しかし、DesignModeler で二つの Surface を **Form New Part** で一つの Part にしておけば、この辺が FLUENT で壁として認識されずにすむ。Meshing からメッシュを Export して FLUENT に読み込ませて、**Boundary Conditions** の **Display Mesh** のボタンで表示する境界条件を選択表示して確認してみよう。

次に、同一直線上にある辺を分割して認識させる方法を示す。たとえば、円管で途中までは断熱条件で、それ以降は一定温度で加熱するという条件を実現したり、複雑な形状で歪みのない長方形で領域をメッシュに切るために同一直線上の辺を部分ごとに異なるメッシュ間隔を設定したりしたい場合がある。

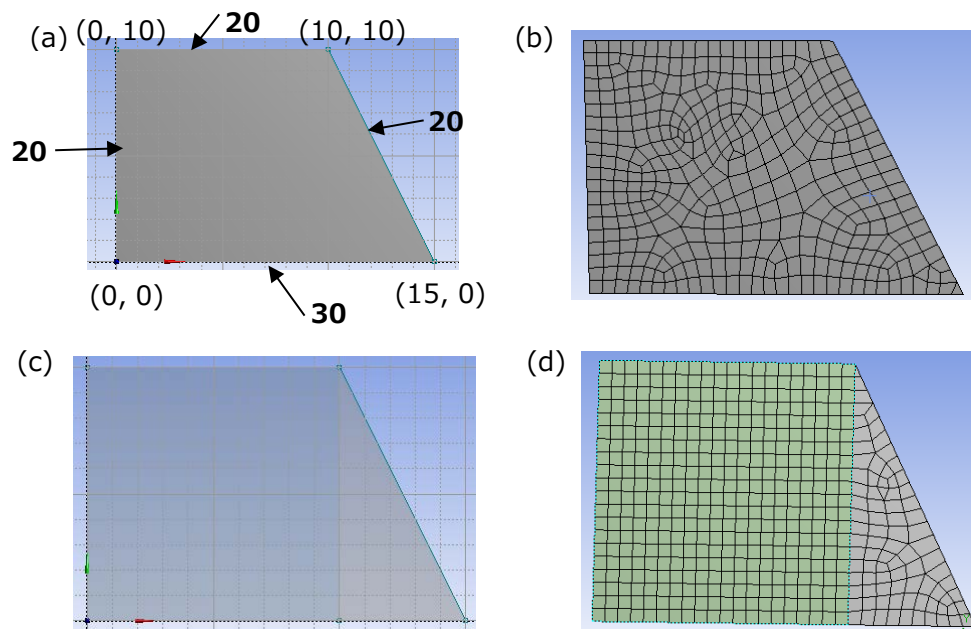



図 24 長方形メッシュをできるだけ多くの領域に用いるために領域を区切る例

実例で確認してみよう。まず、Workbench を立ち上げて(単位は mm にしておく)、Analysis Systems の **Fluid Flow (FLUENT)** をつかんで右側のスペースにドロップする。**Geometry** をクリックして DesignModeler を立ち上げる。4.6.3 節に示した点座標のテキストを DesignModeler で読み込む。図 25 のように点 (図中の◆) を結んで辺 (**Create** → **Surfaces From Edges**) にする。この際、隣り合う点を個別に辺にしていき、左下の Details View の **Operation** はすべて **Add Frozen** にする。デフォルトの Add Material だと 1 直線上にある線分が一つのもので認識されてしまう。次にすべての辺を選んで **Create** → **Surfaces From Edges** から面を作る。この時点で Tree Outline に **15 Part, 15 Bodies** と表示されているはずである。この表示の左の + ボタンを押して Body を表示して shift キーを押しながらすべて選び、右クリックで **Form New Part** を選んで **1 Part** にする。DesignModeler を閉



じる。Workbench の画面に戻って、**Mesh** をクリックして Meshing を立ち上げる。辺を操作するために  を選択しておく。Meshing の画面に先ほど描いた形状が表示される。辺の周りをクリックすると 1 mm の長さの線分単位で選択できることがわかる。これで各線分ごとにメッシュを切る間隔や境界条件の種類（名前）を設定できるようになる。

ここの例のように、形状に折れ曲がりがあり境界の線分が一直線ではない場合は、これらの境界と平行にあり、実体は一つの直線状の計算領域の境界を折れ曲がりの位置に合わせて区切るようにするとよい。また、平行に向かい合う辺上のメッシュ間隔は同一にすると長方形メッシュになる。

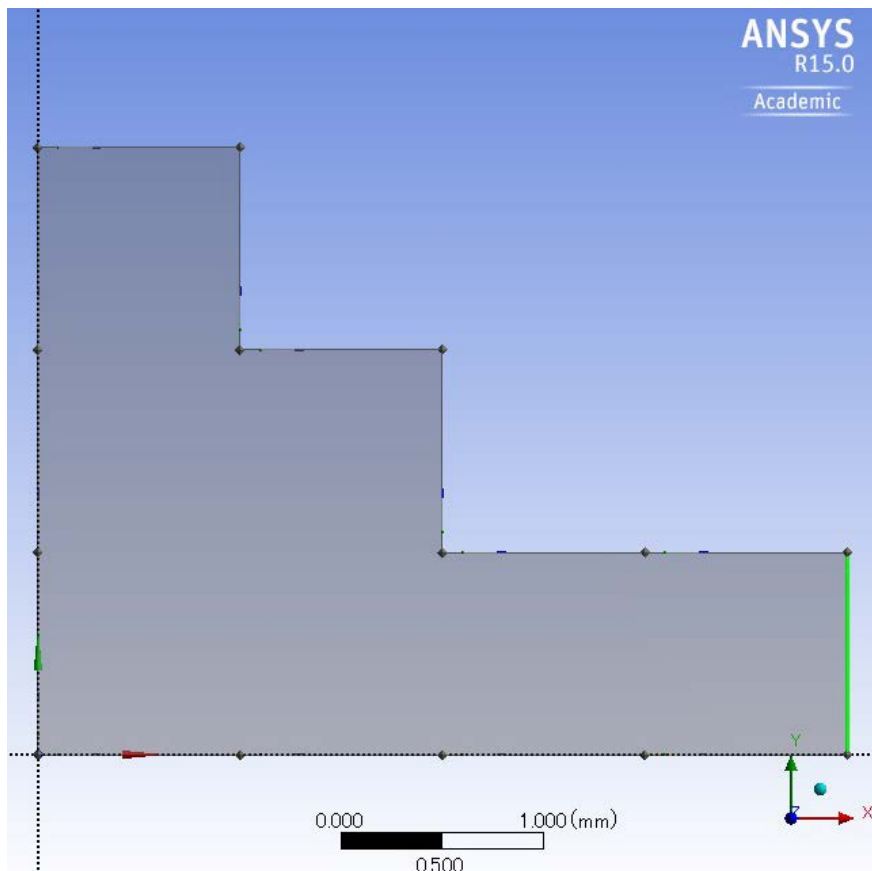


図 25 計算領域の境界の辺を分割する例

## 5 おわりに

本書では CFD の基本的な理論と、それに基づいたメッシュの切り方、収束計算のコツ、反応を伴う流れを扱う際に知っておく必要のある FLUENT の使い方を説明した。CFD の利用に関しては、メッシュの切り方や、緩和係数の決定など、経験的に決めるしかない要素も多い。本書の Case Study がその経験を積むための一助となれば幸いである。また、これから研究を通じてさらに新たな CFD の Case Study が出てくるのに合わせてさらに内容を充実させていき、CFD の経験をより効率的に積めるテキストにしていきたい。最後に、初期濃度分布のパッチ方法を教えていただき、計算結果の変数の分布から標準偏差を求める UDF をご提供くださった Tanthapanichakoon Wiroon 君、本テキストを使って実習を行って改善点を指摘してくれた吉田 篤史君、梅井 良太君、谷川 伸君に感謝いたします。

## 参考文献

- [1] S. V. パタンカー 著，水谷幸夫，香月正司 訳；コンピュータによる熱移動と流れの数値解析，北森出版 (1985)
- [2] J. H. ファーツィガー，M. ペリッチ；コンピュータによる流体力学，シュプリンガー・フェアラーク東京 (2003)
- [3] 越塚誠一；数値流体力学，培風館 (1997)
- [4] J. D. Anderson, Jr.；Computational Fluid Dynamics, McGraw-Hill (1995)
- [5] 水科篤郎，荻野文丸；輸送現象，産業図書 (1981)
- [6] O. Levenspiel；Chemical Reaction Engineering, 3rd ed., John Wiley & Sons (1998)

## 付録

### A 実習例に対応した CFD Case ファイル一覧

ファイル配布は所属研究室内だけで公開版では入手不可。

### B ANSYS 15.0 のインストール方法

ANSYS のインストール方法を以下に示しておく。この方法は、Windows®を用いており、アカデミックライセンスをサーバーで管理しながら複数 PC でシミュレーションする場合に適用できる。一部の PC (OS が Vista?) では DesignModeler と Meshing が動作しないことがある (FLUENT は使えるので CFD の収束計算専用と割り切るのも手)。グラフィックカードも後付けのものがある PC 上が望ましいが、後付けのものでも動作しないことがある。

#### 1 インストールの前に

まず、PC 選択における注意点は以下のとおりである。

- ・ パソコン工房の PC は性能に対するコストパフォーマンスがよい。
- ・ Windows®は 64bit のものを選ぶ (メモリを 4GB 以上にするときは必須, 32bit だと 3GB 程度しか認識しない)。Intel Core i シリーズの CPU は 64bit であり、このシリーズと 64bit 版 Windows®を組み合わせておくと動作が早くなるといわれている。
- ・ リモートログインができるように OS は Windows 7 Professional か Ultimate (またはそれに相当するもの) にする。
- ・ 予算の許す限り速いマシンを使おう。CPU は Intel Core i5 (クロック数は 3GHz 強, 4 コア/4 スレッド, キャッシュ 6MB) 以上のものを用いるとよい。クロック数は処理の速さ, コア数は処理速度を落とさずに並列処理できる操作の数, スレッド数は並列処理できる操作の数の最大値 (2 コア/4 スレッドだとそれぞれのコアで二つずつの処理が可能, ただし処理速度は落ちる), キャッシュは一時的に CPU 内に記憶できる領域 (この領域が大きいとメモリとデータをやり取りせずに CPU 内で情報の処理が済むので動作が早くなる) を示す。
- ・ メモリは最低でも 4GB, できれば 8GB ほしい。
- ・ グラフィックカードもできるだけ性能のよいものを選ぶ (例: グラフィックアクセラレータ GeForce GT 430)。グラフィックカードが停止すると FLUENT も強制終了になる。グラフィックカードによっては返って動作が不安定になることもある。グラフィックカードを追加した場合 (たとえばパソコン工房で上記のグラフィックカードを選択した場合), モニターへの接続口が二つある。一つはオンボードのもので, もう一つ (ハードディスク裏側の下方にあることが多い) は追加のグラフィックカードのものである。モニターは

追加したグラフィックカードの接続口につながらないと映らないので注意しよう。

- ・ 発熱が低く、高効率 = パソコンの寿命が長くなる『80PLUS 認証電源』が好ましい (よいグラフィックカードを使うと消費電力も上がるので電源の消費電力に余裕をもたせておく→マルチタップで電源をとるときは消費電力が 1500 W を超えないように!).
- ・ ハードディスクは 1TB 以上あるとよい (パーティション分割もできればしておく, 半々にしておくもよい).
- ・ まれに Workbench がまともに動作しない PC もある (グラフィックカードをオンボード以外のものを追加しても症状は治らない).

インストールは Administrator (権限の種類もログインの名前もこれになっている必要がある) でログインして行う。Administrator 以外でログインして ANSYS のインストールを開始するとエラーメッセージが表示される。Windows7 のセットアップで作成するアカウントは権限の種類は Administrator になっているが、これではインストールが上手くいかない (Prerequisite がインストールできない, インストールを完了しても DesignModeler が立ち上がらない)。ユーザー名も権限も Administrator になっているユーザーでログインできるようにするためには、「スタート」で出てくるメニューの中の「コンピューター」を右クリックして出てくる「管理」を選択する。「コンピューターの管理」という画面が出てくるので、「ローカルユーザーとグループ」→「ユーザー」を開く。「Administrator」を選択して「アカウントを無効にする」のチェックを外す。一度ログオフすると、ログイン画面にユーザー名「Administrator」が選択できるようになっている。次のページも参照するとよい。  
[http://freesoft.tvbok.com/tips/win7rc64/login\\_administrator.html](http://freesoft.tvbok.com/tips/win7rc64/login_administrator.html)  
次のページの方法でも Administrator を有効にできる。

<http://win7.jp/hk7v1134.htm>

また、PC をインターネットに接続できる状態にしておく (ライセンスサーバーの認識に必要)。Windows Update を実施して、OS を最新の状態にしておく (Service pack が入っていないとインストール時にエラーが出て止まってしまうことがある)。

VISTA, Windows 7 が入っている PC では UAC 機能を無効化してからインストールを開始する。Windows7 の場合、Administrator でログインし、「スタート」→「コントロールパネル」→「システムとセキュリティ」をクリックする。次に現れる画面で「ユーザーアカウント制御設定の変更」を選択し、画面左側のスライドバーを[通知しない]まで下げ[OK]をクリックし、OS を再起動する。

## 2 インストール

インストール用の資料が掲載された URL, 閲覧に必要なログイン ID とパスワードは ANSYS から送付されてくる資料に記載がある。この URL にある「ANSYS 15.0 インストールガイド (日本語)」を見ながら進めるとよい。

ライセンスサーバーについてだけ ANSYS LICENSE MANAGER をインストールする。Windows®と書かれた USB を PC 本体の USB 端子に差し込む。下のほうにある setup.exe を

をクリックするとインストールが始まる。出てくる画面で Select Language を English (Japanese になっけていても必ず変更すること), [Install ANSYS, Inc License Manager] を選択する。最初の警告画面は OK で閉じる。Continue を押していけばよい。途中でファイル選択の画面が出たら、メールで送られてきた license-ansys\_XXXX.txt を開く。Exit で画面を閉じていく。Server ANSLIC\_ADMIN Utility(ANSYS LICENSE MANAGER)をスタートメニューから起動し、出てくる画面の左下に「実行中」と出ていけばインストールは完了である。

次に ANSYS 自体をインストールする。ライセンスサーバーにも入れることができる。DVD をドライブに入れ、setup.exe を右クリックし、[管理者として実行]をクリックするとインストール用の画面が表示される。この画面で Select Language を English (Japanese になっけていても必ず変更すること), 「Install ANSYS, Inc. Products」を選んで ANSYS 本体をインストールしていく。基本的には変更はせずに Next を押していけばよい。CAD ソフトを使っていなければ Enter the Pro/ENGINEER information, Enter the Unigraphics NX information, Solid Works, Autodesk Inventor の画面では Skip にチェックを入れて Next を押す。進んでいくとソフトのインストールが開始する。これに 1 時間程度かかる。インストールの最終段階で Hostname を入力する欄が出てきたら、サーバーのコンピューター名を入力する。Licensing Information の確認画面が出てきて、確認が終わるとその画面下部に「Exit」をクリックできるようになるので、クリックする。最後に Finish installation の画面では Next ではなく「Finish」をクリックしてインストールを終了する。

なお、ライセンスサーバーと同じ LAN 上にある PC (ネットワークから PC 名が見れるもの) へのインストールは無制限に行うことができる。ただし、立ち上げられるのは契約したライセンス分だけである。ライセンスサーバーへの認証によって FLUENT を立ち上げることになる。このためにサーバーへ接続できるようにする必要がある。まず、ライセンスサーバーのホームグループに参加させる (Windows7 だけ可能)。ホームグループの認識を効率的にするために最初の認証のときは、同じ研究室でも有線 LAN 同士の必要がある。コントロールパネルから「ホームグループと共有に関するオプションの選択」を開くと、サーバーの Fluent-PC1 の Administrator の名が出てくる。ホームグループのパスワードを入力するとそのグループに入ることができる。パスワードはホームグループに入っている PC でコントロールパネルの同じホームグループのページを開くと、パスワードの表示を行うリンクがあるので、それをクリックすると表示される。ホームグループに入る際はライセンスサーバーに接続することになるので、別 PC への接続を遮断してしまう Norton やファイアウォールを一時的に無効にしておく。一度ホームグループに入ってしまうと、無線 LAN でも認証可で、Norton を立ち上げていても問題ない。

ホームグループに入っても FLUENT が立ち上がらない場合は、インストール時にサーバーのコンピューター名を入力していない可能性がある。このときは、ANSYS 14.5 → ANSYS Client Licensing → Client ANSLIC\_ADMIN Utility とたどって立ち上げ、「ライセンスサーバー指定」を選び、出てくる画面の下側の「サーバーを指定」のボタンを押し、ホスト名 1 にサーバー PC 名を入力したら「OK」を押せばよい。

### 3 Workbench・FLUENT の起動

FLUENT 起動時に出てくる小画面で「Show More」をクリックして FLUENT Root Path が FLUENT をインストールしたフォルダになっていることを確認しておく。

プログラムから ANSYS→Workbench とすれば Workbench が、ANSYS→Fluid Dynamics →FLUENT とすれば FLUENT が起動する。Workbench の Toolbox から FLUENT を起動させることもできる。

ライセンスサーバーのファイアウォールが有効になっていると以下のようなメッセージが出て、サーバーになっていない PC では FLUENT などが立ち上がらない。ライセンスサーバーのホームネットワークのファイアウォールを無効にしておく。ライセンスサーバーがスリープ状態になってもライセンス認証に失敗するので、サーバーPC で「コントロールパネル」→「ハードウェアとサウンド」→「電源オプション」→「プラン設定の変更」とたどり、「コンピューターをスリープ状態にする」の設定を「なし」にしておく（Windows 7 の場合）。

### 4 ライセンスサーバーのリモート操作

FLUENT をノート PC にインストールするのは負荷が大きくなりすぎることがあったり、インストールしてもソフトが動かなかったりする。サーバーの PC を遠隔で操作できるようにすれば本人がメインで使っている PC の性能が足りないときでも CFD シミュレーションができるようになる。

まず接続される側（OS が Windows 7 Professional・Ultimate であること）は、「スタート」→「コンピューター」を右クリックし「プロパティ」を選択→左ペインの「リモートの設定」をクリック→「リモート アシスタンス」で接続を許可するところチェックし、「リモートデスクトップを実行しているコンピューターからの接続を許可する」のラジオボタンを選択すればよい。

次に接続する側は、「スタート」→「全てのプログラム」→「アクセサリ」→「リモートデスクトップ接続」でクライアントプログラムを起動し、接続先 PC の PC 名、ユーザー名、パスワードを入力して「接続」ボタンを押せば接続できる。

接続する度に接続先 PC のユーザー名を入力するのが煩わしければ、接続設定を保存しておき、そのファイルをダブルクリックするだけで接続可能にもできる。このファイルを「名前を付けて保存」で自分の好きな場所に好きな名前で保存しておけばよい。デフォルトでは「Default.rdp」という名前で保存される。隠しファイルなので、フォルダオプションで隠しファイルを表示できるようにしておく。

### 5 ライセンスの更新

ライセンスの更新時にライセンスのテキストファイルが送られてくる。このファイル名を license.dat に変えておく。同時に更新の方法が書かれた PDF ファイルもくるので、これに従って Server ANSLIC\_ADMIN Utility を立ち上げてサーバーにしている PC の古い license.dat を更新する。この作業でライセンスマネージャの停止と起動を行うが、起動の直後に FLEXml



が停止中のままの場合がある。このときはサーバーを再起動する。

## 6 バージョンアップ

インストール用の資料が掲載された URL、閲覧に必要なログイン ID とパスワードは ANSYS から送付されてくる資料に記載がある。この URL にある「ANSYS 15.0 インストールガイド（日本語）」を見ながら進めるとよい。

買い取りのライセンスでない場合は新しいバージョンがリリースされると関連文書やインストール用 DVD か USB が送られてくる。

インストールの前に、Administrator（権限の種類もログインの名前もこれになっている必要がある）でログインしておく。

Administrator 権限のユーザーで VISTA, Windows7 が入っている PC では UAC 機能を無効化してからインストールを開始する。Windows 7 の場合、Administrator でログインし、「スタート」→「コントロールパネル」→「システムとセキュリティ」をクリックする。次に現れる画面で「ユーザーアカウント制御設定の変更」を選択し、画面左側のスライドバーを[通知しない]まで下げ[OK]をクリックし、OS を再起動する。最初にインストールしたときの設定のままであればこの作業は不要である。

64 ビットの Windows マシンの場合は Windows®と書かれた USB を PC 本体の端子に入れ、USB のドライブを開き、setup.exe を右クリックし「管理者として実行」する。出てくる画面で「ANSYS 製品のインストール」をクリックする。あとは2で説明した流れと同じである。この時点で新しいバージョンがプログラム一覧に出ている。旧バージョンもアンインストールしなければそのまま使える。このため、たとえば「ANSYS 14.5」「ANSYS 15.0」の両方がプログラム一覧に出ている。

次にライセンスマネージャもバージョンアップする。これをしないとバージョンアップしたほうの ANSYS が使えない。ライセンスサーバーにしている PC に USB を入れ、setupLM.exe を右クリックし「管理者として実行」する。最初の画面で言語は日本語から English に変えておく。起動中のライセンスマネージャを停止する警告が出たら「OK」を選んで閉じる。出てくる画面で Next または Continue を押していけばバージョンアップは完了する。ライセンスファイルのアップロードは既存ライセンスを使い続けるのであれば Skip する。上手くいっていれば古いバージョンも新しいものも利用可能になる。

## 7 通常時の Windows®設定の注意点

通常時の Windows®の設定の注意点も挙げておく。Windows Update の設定を“自動でインストール”にしておくと、インストール終了後に自動で再起動がかかることがある。もちろん FLUENT も強制終了になってしまう。Windows 7 の場合、「コントロールパネル」→「システムとセキュリティ」→「Windows Update」の下の「自動更新の有効化または無効化」で、重要な更新プログラムのリストを「更新プログラムをダウンロードするが、インストールするかは選択する」にしておき、インストールと必要な時の再起動は手動で行う設定にしておく。



また、Flash Player の自動更新で勝手に再起動されてしまうこともある。次のページ [http://www.macromedia.com/support/documentation/jp/flashplayer/help/settings\\_manager\\_05.html](http://www.macromedia.com/support/documentation/jp/flashplayer/help/settings_manager_05.html) (グローバル通知設定パネル, リンク切れ) で「Adobe Flash Player アップデートのリリース情報を通知します」のチェックを外しておく。

さらに、初期状態ではエラーが生じたときに Windows は自動で再起動してしまう設定になっている。「コンピューター」を右クリックしてプロパティを開き、左側の「システムの詳細設定」をクリックし、「起動と回復」の「設定」ボタンを押す。現れるウィンドウで「システムエラー」のところの「自動的に再起動する」のチェックボックスを外しておく。

また、PC が強制終了した直後に FLUENT などを立ち上げようとする、FLUENT も利用を停止したことをライセンス管理ソフトが認識していない場合がある。この場合、FLUENT などは立ち上がらない。10 分程度するとライセンスマネージャの認識が追いつくので、FLUENT も立ち上がるようになる。

なお、ライセンスサーバーの PC を「再起動」「シャットダウン後に起動」すると、ライセンスマネージャが起動していても FLUENT などが利用できなくなることがある。Server ANSLIC\_ADMIN Utility で License Manager を Stop させてから再度 Start させると使えるようになる。

## C Journal File

本文中で利用した Journal File, 知っているると便利な Journal File の使い方を以下に示していく。この Journal File は ANSYS15.0 の FLUENT でだけ動作が保障されるものであり、ソフトのバージョンが変わると通用しなく可能性があるので注意すること。

### C.1 Line を引くための Journal

本文での結果整理のための Line を引くのに用いる Journal File の一部を次に示す。y 座標の設定は軸方向に同じ幅で Line を引くのであれば最初の黄色でマーカーを付したところの記述だけでよい。各 x 座標での Line に関する記述は緑でマーカーを付したところが 1 セットにあたる。同じ操作の繰り返しなのでここでは一部の表示に留める。全体の記述が入ったファイルが別途あるので必要なら提供する。そのファイルでは Line は全部で 40 本引かれることになる。なお、3 次元シミュレーションである断面の平均値などを取り出したい場合は Surface → Plane でその面が通る任意の 3 点を指定すればよい。

```
(cx-gui-do cx-activate-item "MenuBar*SurfaceMenu*Line/Rake...")
(cx-gui-do cx-set-real-entry-list "Line/Rake Surface*Frame4(End
Points)*Table1*RealEntry1(y0)" '(0))
(cx-gui-do cx-set-real-entry-list "Line/Rake Surface*Frame4(End
Points)*Table1*RealEntry4(y1)" '(1e-04))
(cx-gui-do cx-set-real-entry-list "Line/Rake Surface*Frame4(End
```

```

Points)*Table1*RealEntry1(x0)" '( 2.5e-06))
(cx-gui-do cx-set-real-entry-list "Line/Rake Surface*Frame4(End
Points)*Table1*RealEntry4(x1)" '( 2.5e-06))
(cx-gui-do cx-set-text-entry "Line/Rake Surface*TextEntry6(New Surface Name)"
"o0.0000025")
(cx-gui-do cx-activate-item "Line/Rake Surface*PanelButtons*PushButton1(OK)")
(中略)
(cx-gui-do cx-set-real-entry-list "Line/Rake Surface*Frame4(End
Points)*Table1*RealEntry1(x0)" '( 0.009))
(cx-gui-do cx-set-real-entry-list "Line/Rake Surface*Frame4(End
Points)*Table1*RealEntry4(x1)" '( 0.009))
(cx-gui-do cx-set-text-entry "Line/Rake Surface*TextEntry6(New Surface Name)" "o0.009")
(cx-gui-do cx-activate-item "Line/Rake Surface*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-activate-item "Line/Rake Surface*PanelButtons*PushButton2(Cancel)")
(cx-gui-do cx-activate-item "MenuBar*WriteSubMenu*Stop Journal")

```

## C.2 FLUENT の設定を行う Journal File

4.2 節の Case Study の設定を行うための Journal File である。収束解のメッシュ数依存性がないことを確かめる際、Meshing でメッシュを切りなおして再計算をするようなときは、FLUENT で再度同じ設定を繰り返す必要がある。Journal File を作っておけば、同じ設定を手作業でせず自動のできるのだから、速くかつ正確に作業が進む。どの記述がどの操作に対応しているかがわかっているならば、設定が少し変わったくらいなら、Journal File を書き換えて Journal File を読み込ませることで FLUENT の設定をしたほうが効率的なこともある。メニューの選択状況が違えば動かないことがあるので FLUENT 起動直後にメッシュを読み込んだすぐ後に使うこと。※はわかりやすいように加えたものであり、実際の journal file には記載しない。実際に入力した部分も黄色のマーカを追加している。

```

(cx-gui-do cx-activate-item
"General*Frame1*Table1*Frame1(Mesh)*ButtonBox1(Mesh)*PushButton1(Scale)"※縮尺設定
(cx-gui-do cx-set-toggle-button "Scale
Mesh*Frame2(Scaling)*Table2(Scaling)*ToggleBox1*Specify Scaling Factors" #f)
(cx-gui-do cx-activate-item "Scale
Mesh*Frame2(Scaling)*Table2(Scaling)*ToggleBox1*Specify Scaling Factors")
(cx-gui-do cx-set-real-entry-list "Scale Mesh*Frame2(Scaling)*Table2(Scaling)*Frame3(Scaling
Factors)*RealEntry2(Y)" '( 0.01))
(cx-gui-do cx-activate-item "Scale
Mesh*Frame2(Scaling)*Table2(Scaling)*PushButton4(Scale)")
(cx-gui-do cx-activate-item "Scale Mesh*PanelButtons*PushButton1(Close)")
(cx-gui-do cx-set-toggle-button
"General*Frame1*Table1*Frame2(Solver)*Table2(Solver)*Frame6(2D Space)*ToggleBox6(2D
Space)*Axisymmetric" #f)※軸対称の設定

```

```
(cx-gui-do cx-activate-item
"General*Frame1*Table1*Frame2(Solver)*Table2(Solver)*Frame6(2D Space)*ToggleBox6(2D
Space)*Axisymmetric")
(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton3(Models)")
(cx-gui-do cx-set-list-selections "Models*Frame1*Table1*Frame1*List1(Models)" '( 5))
(cx-gui-do cx-activate-item "Models*Frame1*Table1*Frame1*List1(Models)")
(cx-gui-do cx-activate-item "Models*Frame1*Table1*PushButton2(Edit)") ※拡散・反応の導入
(cx-gui-do cx-set-toggle-button "Species
Model*Frame1*Table1*Frame1(Model)*ToggleBox1(Model)*Species Transport" #f)
(cx-gui-do cx-activate-item "Species
Model*Frame1*Table1*Frame1(Model)*ToggleBox1(Model)*Species Transport")
(cx-gui-do cx-set-toggle-button "Species
Model*Frame1*Table1*Frame3(Reactions)*Frame1*ToggleBox1*CheckBox1(Volumetric)"
#f)
(cx-gui-do cx-activate-item "Species
Model*Frame1*Table1*Frame3(Reactions)*Frame1*ToggleBox1*CheckBox1(Volumetric)")
(cx-gui-do cx-activate-item "Species Model*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-activate-item "Information*OK")
(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton4(Materials)") ※物質の定義
(cx-gui-do cx-set-list-selections "Materials*Frame1*Table1*Frame1*List1(Materials)" '( 5))
(cx-gui-do cx-activate-item "Materials*Frame1*Table1*Frame1*List1(Materials)")
(cx-gui-do cx-activate-item
"Materials*Frame1*Table1*Frame2*ButtonBox2*PushButton1(Create/Edit)")
(cx-gui-do cx-activate-item "Create/Edit
Materials*Frame1*Table1*Frame1*Frame3*ButtonBox3*PushButton1(FLUENT Database)")
(cx-gui-do cx-set-list-selections "Database
Materials*Frame1*Table1*Frame1*DropDownList4(Material Type)" '( 1))
(cx-gui-do cx-activate-item "Database
Materials*Frame1*Table1*Frame1*DropDownList4(Material Type)")
(cx-gui-do cx-set-list-selections "Database
Materials*Frame1*Table1*Frame1*Frame1*List1(Materials)" '( 552))
(cx-gui-do cx-activate-item "Database
Materials*Frame1*Table1*Frame1*Frame1*List1(Materials)")
(cx-gui-do cx-set-position "Database Materials" '(x 105 y 208))
(cx-gui-do cx-activate-item "Database Materials*PanelButtons*PushButton1(Copy)")
(cx-gui-do cx-activate-item "Database Materials*PanelButtons*PushButton1(Close)")
(cx-gui-do cx-set-text-entry "Create/Edit
Materials*Frame1*Table1*Frame1*Frame1*Table1*TextEntry1(Name)" "a")
(cx-gui-do cx-set-text-entry "Create/Edit
Materials*Frame1*Table1*Frame1*Frame1*Table1*TextEntry2(Chemical Formula)" "a")
(cx-gui-do cx-activate-item "Create/Edit
Materials*PanelButtons*PushButton1(Change/Create)")
(cx-gui-do cx-activate-item "Question*Cancel")
(cx-gui-do cx-set-text-entry "Create/Edit
Materials*Frame1*Table1*Frame1*Frame1*Table1*TextEntry1(Name)" "r")
```

```
(cx-gui-do cx-set-text-entry "Create/Edit
Materials*Frame1*Table1*Frame1*Frame1*Table1*TextEntry2(Chemical Formula)" "r")
(cx-gui-do cx-activate-item "Create/Edit
Materials*PanelButtons*PushButton1(Change/Create)")
(cx-gui-do cx-activate-item "Question*Cancel")
(cx-gui-do cx-set-list-selections "Create/Edit
Materials*Frame1*Table1*Frame1*Frame1*Table1*DropDownList3(Material Type)" '(0))※化
```

### 学種を選択

```
(cx-gui-do cx-activate-item "Create/Edit
Materials*Frame1*Table1*Frame1*Frame1*Table1*DropDownList3(Material Type)")
(cx-gui-do cx-activate-item "Create/Edit
Materials*Frame2(Properties)*Table2(Properties)*Frame1*Frame2*PushButton2(Edit)")
(cx-gui-do cx-set-list-selections "Species*Frame2*Table2*Frame1*List1(Available Materials)"
'(1))
(cx-gui-do cx-activate-item "Species*Frame2*Table2*Frame1*List1(Available Materials)")
(cx-gui-do cx-activate-item
"Species*Frame2*Table2*Frame2*Frame2*ButtonBox2*PushButton1(Add)")
(cx-gui-do cx-set-list-selections "Species*Frame2*Table2*Frame1*List1(Available Materials)"
'(0))
(cx-gui-do cx-activate-item "Species*Frame2*Table2*Frame1*List1(Available Materials)")
(cx-gui-do cx-activate-item
"Species*Frame2*Table2*Frame2*Frame2*ButtonBox2*PushButton1(Add)")
(cx-gui-do cx-set-list-selections "Species*Frame2*Table2*Frame2*Frame1*List1(Selected
Species)" '(0))
(cx-gui-do cx-activate-item "Species*Frame2*Table2*Frame2*Frame1*List1(Selected Species)")
(cx-gui-do cx-activate-item
"Species*Frame2*Table2*Frame2*Frame2*ButtonBox2*PushButton2(Remove)")
(cx-gui-do cx-set-list-selections "Species*Frame2*Table2*Frame2*Frame1*List1(Selected
Species)" '(1))
(cx-gui-do cx-activate-item "Species*Frame2*Table2*Frame2*Frame1*List1(Selected Species)")
(cx-gui-do cx-activate-item
"Species*Frame2*Table2*Frame2*Frame2*ButtonBox2*PushButton2(Remove)")
(cx-gui-do cx-set-list-selections "Species*Frame2*Table2*Frame2*Frame1*List1(Selected
Species)" '(0))
(cx-gui-do cx-activate-item "Species*Frame2*Table2*Frame2*Frame1*List1(Selected Species)")
(cx-gui-do cx-activate-item
"Species*Frame2*Table2*Frame2*Frame2*ButtonBox2*PushButton2(Remove)")
(cx-gui-do cx-activate-item "Species*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-activate-item "Create/Edit
Materials*Frame2(Properties)*Table2(Properties)*Frame2*Frame2*PushButton2(Edit)")※反応
```

### 式・速度式の定義

```
(cx-gui-do cx-set-list-selections "Reactions*Frame3*Frame1*Frame2*Table2*DropDownList4"
'(0))
(cx-gui-do cx-activate-item "Reactions*Frame3*Frame1*Frame2*Table2*DropDownList4")
```

```
(cx-gui-do cx-set-real-entry-list "Reactions*Frame3*Frame1*Frame2*Table2*RealEntry5" '( 2))
(cx-gui-do cx-set-real-entry-list "Reactions*Frame3*Frame1*Frame2*Table2*RealEntry6" '( 2))
(cx-gui-do cx-set-list-selections "Reactions*Frame3*Frame1*Frame4*Table4*DropDownList4"
'( 1))
(cx-gui-do cx-activate-item "Reactions*Frame3*Frame1*Frame4*Table4*DropDownList4")
(cx-gui-do cx-set-real-entry-list "Reactions*Frame3*Frame1*Frame4*Table4*RealEntry5" '( 2))
(cx-gui-do cx-set-real-entry-list "Reactions*Frame3*Frame2(Arrhenius Rate)*RealEntry1(Pre-
Exponential Factor)" '( 2.583e+017))
(cx-gui-do cx-activate-item "Reactions*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-set-list-selections "Create/Edit
Materials*Frame2(Properties)*Table2(Properties)*Frame4*Frame2*DropDownList1" '( 2)) ※流
```

### 体物性の定義

```
(cx-gui-do cx-activate-item "Create/Edit
Materials*Frame2(Properties)*Table2(Properties)*Frame4*Frame2*DropDownList1")
(cx-gui-do cx-set-real-entry-list "Create/Edit
Materials*Frame2(Properties)*Table2(Properties)*Frame7*Frame2*RealEntry3" '( 0.001))
(cx-gui-do cx-set-real-entry-list "Create/Edit
Materials*Frame2(Properties)*Table2(Properties)*Frame9*Frame2*RealEntry3" '( 1e-009))
(cx-gui-do cx-activate-item "Create/Edit
Materials*PanelButtons*PushButton1(Change/Create)")
(cx-gui-do cx-activate-item "Create/Edit Materials*PanelButtons*PushButton1(Close)")
(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton7(Boundary
Conditions)") ※境界条件の定義
(cx-gui-do cx-set-list-selections "Boundary Conditions*Frame1*Table1*Frame1*List1(Zone)"
'( 3))
(cx-gui-do cx-activate-item "Boundary Conditions*Frame1*Table1*Frame1*List1(Zone)")
(cx-gui-do cx-activate-item "Boundary
Conditions*Frame1*Table1*Frame2*Table2*Frame4*Table4*Frame1*ButtonBox1*PushButton1
(Edit)")
(cx-gui-do cx-set-real-entry-list "velocity-inlet-6-
1*Frame4*Frame3(Momentum)*Frame1*Table1*Frame8*Table8*RealEntry2(Velocity
Magnitude)" '( 1))
(cx-gui-do cx-set-real-entry-list "velocity-inlet-6-
1*Frame4*Frame6(Species)*Frame1*Table1*Frame2(Species Mass Fractions)*Table2(Species
Mass Fractions)*Frame1*Table1*RealEntry2(a)" '( 1))
(cx-gui-do cx-activate-item "velocity-inlet-6-1*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton12(Solution Methods)") ※離
```

### 散化方法の設定

```
(cx-gui-do cx-set-list-selections "Solution Methods*Frame1*Table1*Frame3(Spatial
Discretization)*Table3(Spatial Discretization)*DropDownList3(Momentum)" '( 1))
(cx-gui-do cx-activate-item "Solution Methods*Frame1*Table1*Frame3(Spatial
Discretization)*Table3(Spatial Discretization)*DropDownList3(Momentum)")
(cx-gui-do cx-set-list-selections "Solution Methods*Frame1*Table1*Frame3(Spatial
```



```

Discretization)*Table3(Spatial Discretization)*DropDownList4(a)" '( 1))
(cx-gui-do cx-activate-item "Solution Methods*Frame1*Table1*Frame3(Spatial
Discretization)*Table3(Spatial Discretization)*DropDownList4(a)")
(cx-gui-do cx-set-list-selections "Solution Methods*Frame1*Table1*Frame3(Spatial
Discretization)*Table3(Spatial Discretization)*DropDownList5(Energy)" '( 1))
(cx-gui-do cx-activate-item "Solution Methods*Frame1*Table1*Frame3(Spatial
Discretization)*Table3(Spatial Discretization)*DropDownList5(Energy)")
(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton13(Solution Controls)")※工

```

### エネルギー収支式の除外

```

(cx-gui-do cx-activate-item "Solution Controls*Frame2*Table2*PushButton3(Equations)")
(cx-gui-do cx-set-list-selections "Equations*Frame1*List1(Equations)" '( 0 1))
(cx-gui-do cx-activate-item "Equations*Frame1*List1(Equations)")
(cx-gui-do cx-activate-item "Equations*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton14(Monitors)")※収束判定

```

### 条件の設定

```

(cx-gui-do cx-activate-item "Monitors*Frame1*Table1*PushButton2(Edit)")
(cx-gui-do cx-set-toggle-button "Residual
Monitors*Frame1*Table1*Frame2*Table2*Frame1(Equations)*Table1(Equations)*CheckButto
n10" #t)
(cx-gui-do cx-activate-item "Residual
Monitors*Frame1*Table1*Frame2*Table2*Frame1(Equations)*Table1(Equations)*CheckButto
n10")
(cx-gui-do cx-set-toggle-button "Residual
Monitors*Frame1*Table1*Frame2*Table2*Frame1(Equations)*Table1(Equations)*CheckButto
n16" #t)
(cx-gui-do cx-activate-item "Residual
Monitors*Frame1*Table1*Frame2*Table2*Frame1(Equations)*Table1(Equations)*CheckButto
n16")
(cx-gui-do cx-set-toggle-button "Residual
Monitors*Frame1*Table1*Frame2*Table2*Frame1(Equations)*Table1(Equations)*CheckButto
n22" #t)
(cx-gui-do cx-activate-item "Residual
Monitors*Frame1*Table1*Frame2*Table2*Frame1(Equations)*Table1(Equations)*CheckButto
n22")
(cx-gui-do cx-set-toggle-button "Residual
Monitors*Frame1*Table1*Frame2*Table2*Frame1(Equations)*Table1(Equations)*CheckButto
n28" #t)
(cx-gui-do cx-activate-item "Residual
Monitors*Frame1*Table1*Frame2*Table2*Frame1(Equations)*Table1(Equations)*CheckButto
n28")
(cx-gui-do cx-activate-item "Residual Monitors*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-activate-item
"Monitors*Frame1*Table1*Frame4*Table4*PushButton1(Create)")※モニターの設定

```

```

(cx-gui-do cx-set-list-selections "Surface
Monitor*Frame1*Table1*Frame2*Table2*DropDownList1(Report Type)" '( 6))
(cx-gui-do cx-activate-item "Surface
Monitor*Frame1*Table1*Frame2*Table2*DropDownList1(Report Type)")
(cx-gui-do cx-set-list-selections "Surface
Monitor*Frame1*Table1*Frame2*Table2*DropDownList2(Field Variable)" '( 4))
(cx-gui-do cx-activate-item "Surface
Monitor*Frame1*Table1*Frame2*Table2*DropDownList2(Field Variable)")
(cx-gui-do cx-set-list-selections "Surface
Monitor*Frame1*Table1*Frame2*Table2*Frame5*Table5*Frame1*List1(Surfaces)" '( 2))
(cx-gui-do cx-activate-item "Surface
Monitor*Frame1*Table1*Frame2*Table2*Frame5*Table5*Frame1*List1(Surfaces)")
(cx-gui-do cx-activate-item "Surface Monitor*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton17(Solution
Initialization)" ※初期値の設定)
(cx-gui-do cx-set-toggle-button "Solution Initialization*Frame1*Table1*Frame3(Initialization
Methods)*ToggleBox3(Initialization Methods)*Standard Initialization" #f)
(cx-gui-do cx-activate-item "Solution Initialization*Frame1*Table1*Frame3(Initialization
Methods)*ToggleBox3(Initialization Methods)*Standard Initialization")
(cx-gui-do cx-set-list-selections "Solution
Initialization*Frame1*Table1*DropDownList1(Compute from)" '( 1))
(cx-gui-do cx-activate-item "Solution Initialization*Frame1*Table1*DropDownList1(Compute
from)")
(cx-gui-do cx-activate-item "Solution
Initialization*Frame1*Table1*ButtonBox8*PushButton1(Initialize)")
(cx-gui-do cx-activate-item "MenuBar*WriteSubMenu*Stop Journal")

```

### C.3 ファイルの読み込み・緩和係数の変更・収束計算・セーブを行う Journal File

ファイルの読み込み・緩和係数の変更（途中で変更）・収束計算（緩和係数の変更前後で計算回数を変えて実行）・計算結果の保存を行う Journal File の例を以下に示す。網掛けの部分  
がファイル名、太字が緩和係数を変える部分、太字下線部が収束計算回数に相当する。白抜き  
の部分は上書き保存してもいいかどうかの確認のウィンドウが現れ、OK を選択することに  
相当する。一行上のファイル名を読み込み時とは異なるものにして保存する場合は、この  
白抜きの行は不要になる。Case file によって多少記述は変わるが（本書の Case Study にはそ  
のままでは使えない）、どの部分がどの操作に対応するかというパターンを知っていれば、必  
要な部分だけを書き直してほかの Case へ転用することもできる。いきなりファイルの読み取  
りからスタートするので、保存前の case ファイル開いた状態で journal を読み込むと、上書  
き保存の確認への対応の記述がないためエラーが出続けてしまう。このため、今開いている  
case ファイルを保存してから journal を読み込むか、FLUENT 起動直後に読み込むよう  
にすればよい。また、読み込んだ journal と使用・保存する case・dat ファイルは同じフォル  
ダに保存しておくこと。



```

(cx-gui-do cx-activate-item "MenuBar*ReadSubMenu*Case & Data...")
(cx-gui-do cx-set-text-entry "Select File*Text" "file name 1.cas ")
(cx-gui-do cx-activate-item "Select File*OK")
(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton15(Solution Controls)")
(cx-gui-do cx-set-real-entry-list "Solution Controls*Frame1*Table1*Frame5(Under-Relaxation
Factors)*Table5(Under-Relaxation Factors)*RealEntry1(Pressure)" '( 0.04))
(cx-gui-do cx-activate-item "Solution Controls*Frame1*Table1*Frame5(Under-Relaxation
Factors)*Table5(Under-Relaxation Factors)*RealEntry1(Pressure)")
(cx-gui-do cx-set-real-entry-list "Solution Controls*Frame1*Table1*Frame5(Under-Relaxation
Factors)*Table5(Under-Relaxation Factors)*RealEntry4(Momentum)" '( 0.1))
(cx-gui-do cx-activate-item "Solution Controls*Frame1*Table1*Frame5(Under-Relaxation
Factors)*Table5(Under-Relaxation Factors)*RealEntry4(Momentum)")
(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton19(Run Calculation)")
(cx-gui-do cx-set-integer-entry "Run Calculation*Frame1*Table1*IntegerEntry9(Number of
Iterations)" 200)
(cx-gui-do cx-activate-item "Run Calculation*Frame1*Table1*IntegerEntry9(Number of
Iterations)")
(cx-gui-do cx-activate-item "Run Calculation*Frame1*Table1*PushButton21(Calculate)")
(cx-gui-do cx-activate-item "Information*OK")
(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton15(Solution Controls)")
(cx-gui-do cx-set-real-entry-list "Solution Controls*Frame1*Table1*Frame5(Under-Relaxation
Factors)*Table5(Under-Relaxation Factors)*RealEntry1(Pressure)" '( 0.3))
(cx-gui-do cx-activate-item "Solution Controls*Frame1*Table1*Frame5(Under-Relaxation
Factors)*Table5(Under-Relaxation Factors)*RealEntry1(Pressure)")
(cx-gui-do cx-set-real-entry-list "Solution Controls*Frame1*Table1*Frame5(Under-Relaxation
Factors)*Table5(Under-Relaxation Factors)*RealEntry4(Momentum)" '( 0.7))
(cx-gui-do cx-activate-item "Solution Controls*Frame1*Table1*Frame5(Under-Relaxation
Factors)*Table5(Under-Relaxation Factors)*RealEntry4(Momentum)")
(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton19(Run Calculation)")
(cx-gui-do cx-set-integer-entry "Run Calculation*Frame1*Table1*IntegerEntry9(Number of
Iterations)" 3000)
(cx-gui-do cx-activate-item "Run Calculation*Frame1*Table1*IntegerEntry9(Number of
Iterations)")
(cx-gui-do cx-activate-item "Run Calculation*Frame1*Table1*PushButton21(Calculate)")
(cx-gui-do cx-activate-item "Information*OK")
(cx-gui-do cx-activate-item "MenuBar*WriteSubMenu*Case & Data...")
(cx-gui-do cx-set-text-entry "Select File*Text" "file name 1.cas ")
(cx-gui-do cx-activate-item "Select File*OK")
(cx-gui-do cx-activate-item "Question*OK")
(cx-gui-do cx-activate-item "MenuBar*WriteSubMenu*Stop Journal")

```

上記の記述をコピーアンドペーストして、太字の緩和係数・収束計算回数と黄色のファイル名の部分を適宜書き換えることで、自動で多数のファイルの収束計算を進めることができる。ただし、1文字でも Journal File にミスがあると計算は止まってしまい、エラーが

出続けることになるので、途中の様子を確認するようにしたい。

## D ユーザー定義関数 (UDF)

本文中で触れたユーザー定義関数 (User Defined Function, UDF) を以下に示す。FLUENT のメニューの Help ® User's Guide Contents から見ることのできる FLUENT の下の階層にある UDF Manual の Chapter 8 にも例が多く乗っているのを自分の状況に合わせて書き換えるのがまずはよいだろう。

### D.1 速度分布を設定する UDF

Fluent では C 言語で記述したユーザー定義関数を用いて、入口断面における速度分布・物性値の温度依存性の近似式などを定義できる。ここでは平行平板間層流における流路入口の放物型速度分布の関数 (関数名: laminar\_inlet) を定義する C ファイルを示す。/\*と\*/囲んだところはコメントになり、実行には無関係になる。作成した C ファイルを FLUENT でコンパイルして入口速度・物性指定のところで関数名を指定すれば、この関数を用いて実際の計算を行ってくれる。UDF が複数ある場合は一つのファイルにすべてを記述し、一度にコンパイルを行う必要がある。UDF を用いる際に、関数名にハイフン (-) が入っていたり、フォルダ名に空白 (半角もダメ) や全角文字があったりするとコンパイルができない (Error: line 1 parse error と出る) ので注意が必要である。なお、入口断面が線 (2次元) でも面 (3次元) でも同様の UDF で設定ができる。

```
#include "udf.h"
DEFINE_PROFILE(laminar_inlet, /* function name */
               thread, /* thread */
               position) /* variable number */
{
    real x[ND_ND];
    real y;
    face_t f;
    /* loop over each of the faces of this zone */
    begin_f_loop (f,thread)
    {
        F_CENTROID(x,f,thread);
        y = x[1];
        F_PROFILE(f,thread,position) = 2.0-2.0e8*y*y;
    }
    end_f_loop (f,thread)
}
```

## D.2 物性を定義する UDF

粘度の温度依存性が次のように温度域によって異なる関数で表現される場合の UDF (関数名: cell\_viscosity) を以下に示す。Material の Viscosity で UDF から関数名を指定すれば有効になる。

$$\mu = \begin{cases} 3.3638 \times 10^{-2} - 1.9937 \times 10^{-4} T + 3.0029 \times 10^{-7} T^2 & (293 < T < 313) \\ 1.4759 \times 10^{-2} - 7.8678 \times 10^{-5} T + 1.0738 \times 10^{-7} T^2 & (313 \leq T \leq 343) \\ 7.2682 \times 10^{-3} - 3.4679 \times 10^{-5} T + 4.2762 \times 10^{-8} T^2 & (343 < T < 373) \end{cases} \quad (66)$$

```
#include "udf.h"
DEFINE_PROPERTY(cell_viscosity, cell, thread)
{
  real mu_lam;
  real temp = C_T(cell,thread);
  if (temp < 313.)
    mu_lam = .033638 -.00019937 * temp + 3.0029e-7 * pow(temp, 2.);
  else if (temp > 343.)
    mu_lam = .0072682 -.000034679 * temp + 4.2762e-8 * pow(temp, 2.);
  else
    mu_lam = .014759 -.000078678 * temp + 1.0738e-7 * pow(temp, 2.);
  return mu_lam;
}
```

なお、上記の設定は Material の Viscosity の設定で piecewise-polynomial で Range の数を 3 にし、それぞれの範囲の下限、上限温度と、0 次の係数を 1、1 次の係数を 2、2 次の係数を 3 のボックスに入力すればよい (同様に 7 次の項を 8 まで入れることができる)。温度が低いほう順に温度が高くなっていくように入力しないとうまく認識されない。ただし、この方法だと 7 次までの多項式にしか対応できないし、範囲の数も最大三つまでの制限がある。T<sup>-1</sup> の項や、範囲の分割が多い場合は UDF を使う必要がある。

また臨界点付近のように物性の変化が急激で多項式での近似が上手くいかない場合は、piecewise-linear で温度刻みを極力小さくして物性値を登録するとよい。物性値の登録は同じ作業の繰り返しなので、[付録 C](#) でも紹介した Journal File を作成して行ってもよい。物性値を 1 点登録する作業で Journal を作り、物性値登録そのものの部分をコピー&ペーストして物性値・温度といった条件だけを書き換えていく。

## D.3 結果整理の (標準偏差) UDF

計算結果の変数の分布から標準偏差を求める UDF を以下に示す。この UDF では系内に A と B が等質量で存在していて、混合の進行の経時変化を A の質量分率の計算領域内の標準偏差から追跡するのに用いる。この UDF を使うには、Define → Use-Defined → Functions → Interpreted でファイル名を指定してコンパイルした後に、Define → Use-Defined →

Functions → Execute on Demand とすると、体積と標準偏差が Fluent の画面に表示される。

```
#include "udf.h"
DEFINE_ON_DEMAND(mixing_calc)
{
Domain *d;
int i=0; /*成分によってこの値は変わる (mixture の構成要素の並び順) */
real c0,cv,dev;
real v_tot=0;
real sum=0;
Thread *c_thread;
cell_t c;
d = Get_Domain(1);
thread_loop_c(c_thread,d)
{
begin_c_loop(c, c_thread)
{
c0=C_YI(c,c_thread,i);
cv=C_VOLUME(c,c_thread);
sum += cv*(c0-0.5)*(c0-0.5);
v_tot += C_VOLUME(c,c_thread);
}
end_c_loop(c, c_thread)
}
dev = sqrt(sum/v_tot);
printf("%g %g %g\n",v_tot,dev);
}
```

#### D.4 Langmuir 型吸着を考慮した反応速度式の UDF

本テキストの演習でも見ているとおり、FLUENT の反応速度の定義は基本的にはべき乗則に対応している。流路表面に担持された触媒での反応の場合、吸着の過程が反応全体において律速段階になるとときには、分母・分子のある速度式になることがある (Langmuir の吸着等温式を仮定した場合)。この場合は以下に示すような UDF を記述し、FLUENT 上で、

Define → User-Defined → Functions → Compiled

Source Files で「Add」を押して UDF を記述したファイルを選択 → Build → Load として UDF をコンパイルして Fluent に取り込む。次に、

Define → User-Defined → Function Hooks

として現れた画面で、Surface Reaction Rate のところに対応する UDF の名称を選択しておく。これまでのものとは異なり、この UDF のファイルは C 言語をコンパイルして使用するため、C 言語のコンパイラ (たとえば Visual Studio 2005 (有償)、Windows 7 で無料でなら Visual C++2008 と Windows SDK、.NET Framework 4、下に方法とダウンロードリンク) を事前にインストールしておく必要がある。ほかの注意点としては、FLUENT のファイルと

UDF のファイルは同一フォルダに入れておき、フォルダ・ファイル名には日本語を使わないようにする。

- ・ 下記を番号順にインストールする。
- ① Visual C++2008 (Microsoft SQL Server 2008 はチェックを外してインストールを省略してよい) <http://www.microsoft.com/japan/msdn/vstudio/2008/product/express/>
- ② .NET Framework 4  
<http://www.microsoft.com/downloads/ja-jp/confirmation.aspx?FamilyID=9cfb2d51-5ff4-4491-b0e5-b386f32c0992>
- ③ Windows SDK for Windows 7.1 (インストール内容を選ぶ画面で Sample は不要なのでチェックを外す, 最後にエラーのようなものが出るが問題ない.)  
<http://www.microsoft.com/download/en/details.aspx?id=8279>
- ・ インストールが終わったら, C:\Program Files\ANSYS Inc\v150\fluent\ntbin\win64 のフォルダにある setenv.exe (Windows 7 ならスタート左下の「プログラムとファイルの検索」で検索するとよい) をクリックして環境設定をする。現れる画面で「OK」とし, Your environment has been successfully modified. と出ればよい。ちなみに以下の環境変数が加わる。

変数 : Path, 値 : c:\program files\ansys inc\v150\fluent\ntbin\win64

変数 : FLUENT\_INC, 値 : c:\program files\ansys inc\v150\fluent

- ・ 環境設定が終わったら, スタート→すべてのプログラム→Microsoft Windows SDK v7.1 →Windows SDK 7.1 Command Prompt としてコマンドプロンプトを立ち上げる。コマンドラインで fluent 2ddp (2次元倍精度) などと入れると FLUENT が立ち上がる。この状態からだと UDF のコンパイルができる。

ここでは, Langmuir の吸着等温式を仮定し,  $PH + H_2O_2 \rightarrow HQ (r_1)$ ,  $PH + H_2O_2 \rightarrow CA (r_2)$ ,  $HQ + H_2O_2 \rightarrow BQ (r_3)$  の逐次・並列反応が, B・C・D・E を吸着種として, 吸着律速で以下の速度式になる場合の UDF (関数名 : my\_rate) を示しておく。変数の意味は, rho\_w は流体の密度, y は質量分率, mw は分子量である。物質に付いた番号は Materials → Mixture Species の Selected Species で選んだ順に一致する。

$$r_1 = k_1 C_{PH} K_B C_{H_2O_2} / (1 + K_B C_{H_2O_2} + K_C C_{HQ} + K_D C_{CA} + K_E C_{BQ})$$

(Reaction Name : Reaction-1) ←速度式の記述の前の行の""内の名称と一致

$$r_2 = k_2 C_{PH} K_B C_{H_2O_2} / (1 + K_B C_{H_2O_2} + K_C C_{HQ} + K_D C_{CA} + K_E C_{BQ})$$

(Reaction Name : Reaction-2)

$$r_3 = k_3 K_C C_{HQ} K_B C_{H_2O_2} / (1 + K_B C_{H_2O_2} + K_C C_{HQ} + K_D C_{CA} + K_E C_{BQ})^2$$

(Reaction Name : Reaction-3)

```
#include "udf.h"
```

```
#define k1 10
```

```
#define k2 30
```

```

#define k3 1
#define KB 0.001
#define KC 100
#define KD 100
#define KE 100
#define rho_w 983

DEFINE_SR_RATE(my_rate,f,t,r,mw,yi,rr)
{
  Thread *t0=t->t0;
  cell_t c0=F_C0(f,t);
  real PH = yi[0];
  real H2O2 = yi[1];
  real HQ = yi[2];
  real CA = yi[3];
  real BQ = yi[4];
  PH *= rho_w/mw[0];
  H2O2 *= rho_w/mw[1];
  HQ *= rho_w/mw[2];
  CA *= rho_w/mw[3];
  BQ *= rho_w/mw[4];

  if (STREQ(r->name, "reaction-1"))
    *rr = k1 * PH * KB * H2O2 / (1. + KB*H2O2 + KC*HQ + KD*CA + KE*BQ);
  else if (STREQ(r->name, "reaction-2"))
    *rr = k2 * PH * KB * H2O2 / (1. + KB*H2O2 + KC*HQ + KD*CA + KE*BQ);
  else if (STREQ(r->name, "reaction-3"))
    *rr = k3 * KC * HQ * KB * H2O2 / pow((1. + KB*H2O2 + KC*HQ + KD*CA + KE*BQ), 2);
}

```

## E 非等温反応系の計算に必要な設定

本文中の例では等温系を例に用いたが、エネルギー収支も合わせて計算できるので非等温反応系も扱うことができる。非等温になったときにさらに必要になる設定を列挙しておく。

- Define ◎ Model ◎ Energy で Energy Equation にチェックを入れる (Species Transport モデルのときはデフォルトでチェックが入っている)。
- Solve ◎ Control ◎ Solutions で Equations のところの Energy を選択する。  
反応熱を定義する。ただし、反応熱という項目はない。Define ◎ Materials ◎ Material type: Fluid で各成分の Standard State Enthalpy を設定することで定義する。単位に注意する。 $10^6 \text{ j/kgmol} = 10^6 \text{ J/kmol} = 1 \text{ kJ/mol}$  である。A ◎ B,  $\Delta H = -100 \text{ kJ/mol}$  としたい場合、A の Standard State Enthalpy を  $H_A$ , B のそれを  $H_B$  とすると、



$$\Delta H = (\text{生成系のエンタルピー}) - (\text{原系のエンタルピー}) = H_B - H_A \quad (67)$$

の関係が成立するので、たとえば、 $H_A = 10^8 \text{ j/kgmol}$ ,  $H_B = 0 \text{ j/kgmol}$  とすればよい。

- Define ◎ Materials ◎ Material type: Mixture で熱伝導率・熱容量を入力する。
- 物性の温度依存性を定義する。全温度領域で単一の多項式で温度依存性が表現されるなら、物性定義のオプションで piecewise-linear や piecewise-polynomial を選択して、多項式に対応するように温度依存性の係数を入力すればよい。温度域によって温度依存性の式が変わるのであれば、付録 D.2 のように UDF をコンパイルしてから、オプションで User-defined を選んでからその関数名を指定することになる。
- 必要に応じて Energy の緩和係数を 1 未満にする。
- 発散を抑制するために Solve ◎ Controls ◎ Limits で上限温度を設定する（断熱温度上昇よりも高い温度にはならないはずなのでこれを目安に）。デフォルトの 5000 K で上限値に達するとともに収束することはまずない。温度上限値に達しているという表示が出ていつまでも消えない場合は、緩和係数を小さくすることも視野に入れておく。

## F スラッグ流再現のための設定方法

混相流のシミュレーションも FLUENT ではできる。スラッグ流を再現するための設定の概要を記しておく。

- T 字路を描いておく。非定常計算になるので、計算負荷を下げるためにまずは 2 次元形状にしておく。流路幅は 1 mm 以下、合流後の流路長さは幅の 10 倍以上は確保しておく。入口は **Velocity-inlet**、出口は **Pressure-outlet** にしておく。
- FLUENT で Solution Setup → Models で **Multiphase** を選択した状態で **Edit** を押す。**Volume of Fluid** のラジオボタンを選んで混相流のモデルを選ぶ。自動で非定常計算になる。Solution Setup → General で Solver の Time のラジオボタンが Transient になっている。
- **Multiphase** のウィンドウで左下の **Implicit Body Force** にチェックを入れる。収束性が向上する。右上の **Number of Eulerian Phases** に相の数を入れる。二相流ならデフォルトの 2 のままでよい。
- **Materials** で水相と油相の物質（たとえば水とトルエン）を定義しておく。
- **Phases** の **Edit** ボタンを押して相の名前と Primary phase にする相の成分を指定する。
- **Interaction** ボタンを押して出てくる画面で「**Surface Tension**」タブに移り、**Surface Tension Force Modeling** のチェックを入れる。画面の表示が変わったら Adhesion Options のところで **Wall Adhesion** にチェックを入れる（接触角の設定が有効になる）。Surface Tension Coefficient のところに二相の表面張力比（左の成分の表面張力 ÷ 右の成分の表面張力）を入力する。
- **Boundary Conditions** の設定を次にする。各境界で **Phase** の欄で流体全体（mixture）と各相を設定できるようになっている。**Velocity-inlet** になっている境界は全体の設定に対

応する **mixture** で流速を設定する。Phase を変更して Primary phase 以外の相の Volume Fraction を「**Multiphase**」のタブ設定する。

- Wall 境界では Phase を **mixture** にして「**Momentum**」のタブで **Contact Angle** を設定する。左側に壁に親和性が高い成分がくる。
- Solution → **Solution Methods** で **Momentum** だけ **Second Order Upwind** にしておく。Volume Fraction は Geo-Reconstruct のままでよい。
- **Calculation Activities** で、途中過程のデータやアニメーションを作成したければ [4.3.1](#)・[4.3.2](#) 節で説明した方法で設定をしておく。
- **Run Calculation** で Time Step Size は  $10^{-4}$  s, Number of Time Steps は流体が入口から流れ始めて十分に合流後の流路までたどり着ける、つまりスラグの形成が起こる時間を含められるようにする、Max Iterations/Time Step (1 タイムステップあたりの収束計算回数) はデフォルトの 20 でまずは計算してみる。
- 計算が終わったら、Results → **Graphics and Animations** → Graphics の中の **Contours** → **Set Up** → Contours of のところで **Phase • Volume fraction • 相** を指定して、各相体積分率の Contour を表示してスラグ流が形成できていることを確認する。たとえば図 26 のようになる。

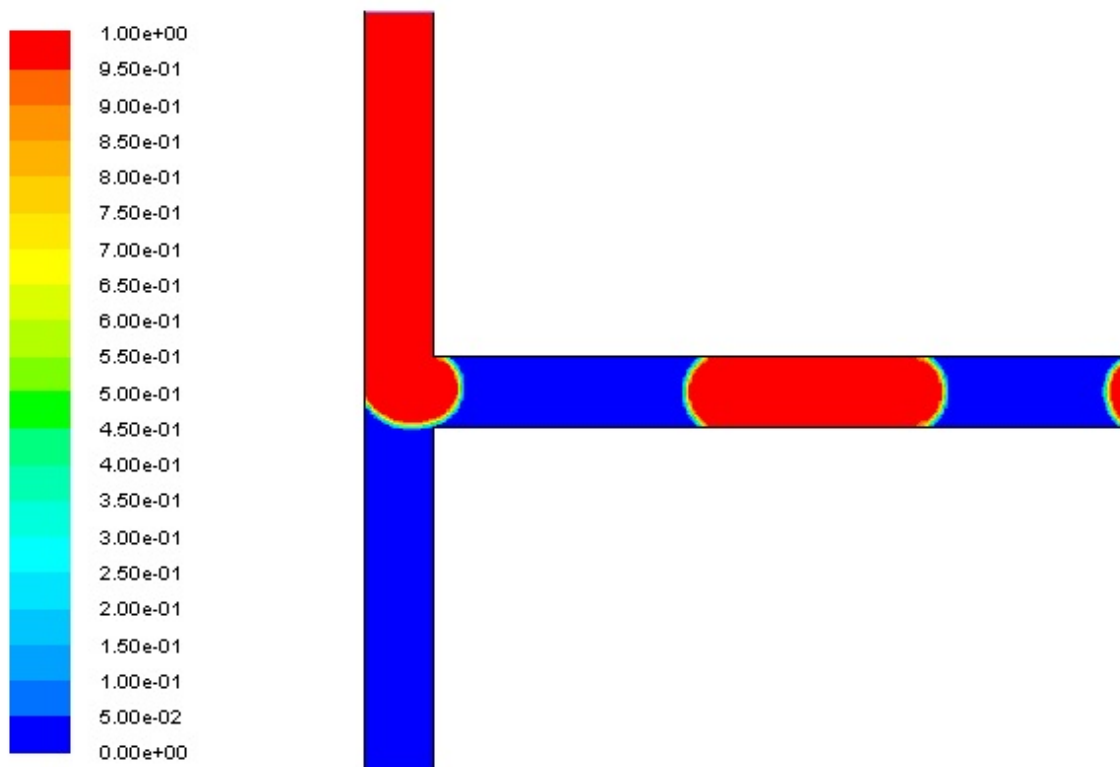


図 26 スラグ流の計算結果の例

**CFDの理論とANSYS利用法の基本**

---

2017年7月24日 2017年度版第1刷 発行  
2017年12月17日 2017年度版第3刷 発行

著者・発行者：青木 宣明

---

© 2017 青木 宣明

メールアドレス：

nobuaki-aoki@panoramic-view.info

自由に使用していただいて構いませんが本書を使用したことによる不利益の責任は一切負いかねます。間違いなどの指摘は歓迎します。

ANSYS, ANSYS Workbench, FLUENTは米国 Ansys, Inc.の米国およびその他の国における登録商標です。

Printed in Japan.