

User Guide of FFV-C

Frontflow / violet Cartesian

Ver. 0.8.7

Institute of Industrial Science
The University of Tokyo

Advanced Institute for Computational Science
RIKEN

July 2015



FFV-C	Version 0.8.7	25 July	2015
	Version 0.8.6	14 July	2015
	Version 0.8.0	22 Aug.	2013
	Version 0.7.1	12 July	2013
	Version 0.6.0	3 June	2013
	Version 0.5.0	14 July	2012

(c) Copyright 2007-2011

VCAD System Research Program, RIKEN. All rights reserved.
2-1, Hirosawa, Wako, 351-0198, JAPAN.

(c) Copyright 2012-2015

Advanced Institute for Computational Science, RIKEN. All rights reserved.
7-1-26, Minatojima-minami-machi, Chuo-ku, Kobe, 650-0047, JAPAN.
<http://www.aics.riken.jp/>

(c) Copyright 2011-2015

Institute of Industrial Science, The University of Tokyo. All rights reserved.
4-6-1 Komaba, Meguro-ku, Tokyo, 153-8505 JAPAN.
<http://www.iis.u-tokyo.ac.jp/>

目次

第 1 章	FFV-C の概要	1
1.1	特徴	2
1.2	開発経緯	2
1.3	ライセンス	3
第 2 章	インストール	4
2.1	MPI ライブラリのインストール	5
2.2	パッケージによるインストール	6
2.2.1	ffvc_package	6
2.2.2	ターゲット環境に応じた指定項目	7
2.3	個別ライブラリ/アプリのインストール	8
2.3.1	ライブラリのインストール	8
2.3.2	FFV-C のコンパイル	8
第 3 章	基礎方程式と解析方法	11
3.1	支配方程式	12
3.1.1	非圧縮性流体	12
3.1.2	非圧縮性流体への弱い圧縮性の導入	13
3.2	無次元化	14
3.2.1	非圧縮	14
	無次元化パラメータの選択	15
	純強制対流	15
	熱対流	15
	純自然対流	15
	固体熱伝導	15
	共役熱移動	15
3.2.2	弱圧縮形式	16
3.3	解法アルゴリズム	16
3.3.1	Fractional Step 法	16
	Euler Explicit	16
	Navier-Stokes equations	16
	Thermal transport equation	17
	Adams-Bashforth	17
	Navier-Stokes equations	17
	Thermal transport equation	17
3.3.2	連続の式と圧力 Poisson の二重反復	17

第 4 章	解析モデルの作成	19
4.1	大規模並列格子生成の問題点と方針	20
4.2	形状近似	21
4.3	解析モデルの作成	21
4.3.1	形状データの準備	21
4.3.2	計算領域の設定	22
4.4	組み込み例題	22
4.4.1	IP_Rect クラス	23
4.4.2	IP_PMT クラス	24
4.4.3	IP_CYLINDER クラス	25
4.5	例題	27
第 5 章	入力パラメータ	28
5.1	パラメータの記述構文	29
5.2	FFV-C の指定パラメータ	30
5.2.1	ApplicationControl	30
	CheckParameter	30
	DryRunBC	30
	Operator	30
	VariableRange	30
5.2.2	ConvectionTerm	31
5.2.3	DomainInfo	32
	UnitOfLength	32
	GlobalDivision	32
	GlobalOrigin	32
	GlobalPitch	32
	GlobalRegion	32
	GlobalVoxel	32
	ActiveSubDomainFile	32
5.2.4	FillHint	33
5.2.5	GeometryModel	34
	Source	34
	VoxelOutput	34
	Output	35
	OutputGlyph	35
5.2.6	GoverningEquation	36
	FlowEquation	36
	HeatEquation	36
	Buoyancy	36
	TimeVariation	36
	PDEtype	36
5.2.7	IntrinsicExample	37
5.2.8	Iteration	38
	Alias	39
	Class	39

MaxIteration	39
ResidualCriterion	39
ResidualNorm	39
ErrorNorm	40
DivMaxIteration	40
DivCriterion	40
DivNorm	40
5.2.9 MediumTable	41
5.2.10 MonitorList	42
5.2.11 Output	44
Log	44
Base	44
Profiling	45
WallInfo	45
Console, History	45
Data	45
Format	45
DirectoryPath	46
TimeSlice	46
BasicVariables	46
StatisticalVariables	47
FormatOption	47
SPH	47
PLOT3D	47
5.2.12 Reference	49
5.2.13 ReferenceFrame	50
5.2.14 ShapeApproximation	51
5.2.15 SolvingMethod	52
5.2.16 StartCondition	53
InitialState	53
5.2.17 TimeControl	55
Acceleration	55
TimeStep	55
Session	56
Statistic	56
5.2.18 TurbulenceModeling	58
Model	58
VelocityProfile	58
InitialPerturbation	58
5.2.19 Unit	59
第 6 章 境界条件	60
6.1 境界条件の概要	61
6.1.1 外部境界条件と局所境界条件	61
6.1.2 計算格子と内部・外部領域	61

6.1.3	BcTable ノードのパラメータ構造	62
6.1.4	境界条件の基本リスト	63
	Boundary	63
	Kind	63
	Class	63
	Medium	63
6.1.5	OuterBC	64
6.2	外部境界条件	65
6.2.1	壁面境界	65
	ThermalOption	66
	断熱境界	66
	熱流束境界	66
	熱伝達境界	66
	等温境界	69
6.2.2	対称境界	71
6.2.3	流境界	72
6.2.4	速度指定境界	73
6.2.5	周期境界	74
6.2.6	トラクションフリー境界	76
6.2.7	遠方境界	77
6.3	局所境界条件	78
6.3.1	壁面境界	78
	流れの境界条件	78
	熱境界条件	78
	断熱境界	79
	熱流束境界	80
	熱伝達境界	80
	等温壁境界	81
6.3.2	流境界条件	81
	流れの流境界	81
	熱流境界	82
6.3.3	速度指定条件	82
	流れの境界条件	82
	熱境界条件	83
6.3.4	セルボリュームに対する熱境界条件	84
	SpecifiedTemperature	84
	HeatSource	84
6.3.5	モニタ	85
第 7 章	モニタリング機能	87
7.1	パラメータファイルで指定する方法	88
7.1.1	値のサンプリング方法	90
	Nearest	90
	Interpolation	90
	Smoothing	91

7.1.2	指定パラメータの制限およびエラー処理	91
7.1.3	出力ファイルフォーマット	92
	ヘッダ領域	92
	データ領域	92
7.2	局所境界条件も合わせて指定する方法	93
7.3	モニター例	94
7.3.1	Line と Pointset	94
	初期化時の出力情報	96
	単一ファイル出力	97
	分散ファイル出力	97
7.3.2	Polygon	98
	初期化時の出力情報	99
	ファイル出力	99
7.3.3	Cylinder と Box	100
第 8 章	ファイル管理とリスタート	102
8.1	分散並列ファイル管理	103
8.1.1	CDMlib の提供するファイルフォーマット	103
8.1.2	メタファイル	103
	プロセス情報ファイル (proc.dfi)	103
	インデクスファイル (prs.dfi, vel.dfi, fvel.dfi, tmp.dfi など)	104
8.1.3	ファイルフォーマット	109
8.2	リスタート	110
8.2.1	標準リスタート	111
	平均値とリスタート	112
8.2.2	同一解像度でプロセス数が異なるリスタート	113
8.2.3	細分化リスタート	113
8.2.4	ステージング	114
第 9 章	ソルバーの実行	115
9.1	FFV-C ソルバーの実行	116
9.2	出力ファイル	117
9.2.1	出力ファイルの種類と指定	117
9.2.2	解析条件情報 [condition.txt]	118
9.2.3	領域情報 [DomainInfo.txt]	119
9.2.4	基本履歴 [history_base.txt]	120
9.2.5	コンポーネント履歴 [history_compo.txt]	121
9.2.6	流量収支履歴 [history_domainflux.txt]	122
9.2.7	反復履歴 [history_iteration.txt]	122
9.2.8	サンプリング履歴 [sampling.txt]	122
9.2.9	性能情報	123
9.2.10	ボクセルファイル	124
9.2.11	結果ファイル	124
9.2.12	メモリ使用量の情報	124
9.3	並列計算	125

9.3.1	MPI 並列	125
9.3.2	スレッド並列	125
9.4	各プラットフォームにおける実行	126
9.4.1	RICC	126
9.4.2	京	128
第 10 章	アップデート情報	129
参考文献		131
索引		132

第 1 章

FFV-C の概要

本章では，三次元非定常非圧縮熱流体解析ソルバー FFV-C について概要を説明します．

1.1 特徴

FFV-C(Frontflow/violet Cartesian) は、直交格子上で三次元非定常非圧縮性熱流体を解析するシステムです。FFV-C ソルバーは、下記のような特徴を持っています。

格子	Isotropic Cartesian (直交等間隔) Anisotropic Cartesian (各軸方向に異なる格子幅を指定可能な直交格子)
形状近似	キューブ近似 (Binary), 任意形状 (Cut)
変数配置	コロケート
離散化	有限体積法, 差分法
時間積分	一次精度 Euler 陽解法, 二次精度 Adams-Bashforth 法
空間スキーム	一次精度風上, 三次精度 MUSCL, 二次精度中心
解法	Fractional Step 法
反復法	Point SOR, 2-colored SOR-SMA (ストライドメモリアクセス版) BiCGstab (前処理つき双共役勾配法)
スタート機能	Initial(Impulsive, Smooth), 指定時刻からの再スタート, 粗格子からの内挿リスタート
入力ファイル	モデルファイル (STL フォーマット), テキストファイル (計算条件など)
出力ファイル	sph フォーマット, PLOT3D フォーマット, 履歴ファイル, モニター出力, 性能情報など
外部境界条件	固定・移動壁面, 流入, 流出, 周期, 対称, トラクションフリー
内部境界条件	壁面, 速度規定, 流出, 部分周期境界, 圧力損失, 多孔質
温度境界条件	断熱, 熱伝導, 熱伝達, 熱流束, 等温
乱流モデル	LES 局所 SGS モデル, CSM (+ 減衰関数), WALE (+ 減衰関数) 標準 Smagorinsky (+ 壁法則)
並列化	Hybrid 並列 (プロセス並列と OpenMP によるスレッド並列)
領域分割	等分割, Active Sub-Domain (活性サブドメインのみアサイン)
組み込み例題機能	キャビティフロー問題など, 解析モデルの形状ファイルが不要な問題
利用ライブラリ	MPI ライブラリ, CDMLib, CPMLib, PMLib, Polylib, TextParser

図 1.1 に FFV-C のモジュール構造を示します。

ソルバーを構築する上で必要な、パラメータハンドリング、主要な境界条件処理とパラメータの関連づけ、ファイル入出力、並列計算処理、組み込み例題など、コアアルゴリズム以外の部分は、FlowBase クラスなどにパッケージ化しています。FFV-C で利用している各ライブラリは次のような機能を提供します。

CDMLib	ファイル入出力の管理を行う機能ライブラリ
CPMLib	直交等間隔格子の並列領域管理ライブラリ
OpenMPI	プロセス並列ライブラリ
PMLib	性能測定パッケージライブラリ
Polylib	幾何形状データを並列領域で管理する機能を提供するライブラリ
TextParser	YAML に類似した形式で記述されたテキストのパラメータをパースするライブラリ

1.2 開発経緯

FFV-C は、HPCI 戦略プログラム「分野4次世代ものづくり」プロジェクトのもと、東京大学生産技術研究所において研究開発を推進しているシミュレーションソフトウェアで、理化学研究所 VCAD システム研究プログラムで開発さ

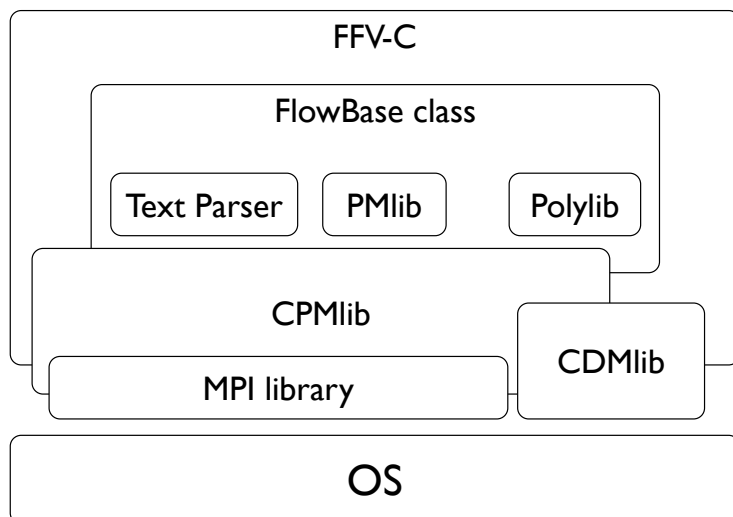


図 1.1 FFV-C の内部モジュール構造

れてきた V-Sphere::CBC ソルバークラスをオリジナルとして、大規模並列解析を実施するために改変が加えられたプログラムです。CBC の場合には V-Sphere ミドルウェアを利用して記述されていましたが、FFV-C ではよりコンパクトなアプリケーションミドルウェアである CPMlib を用いた記述になっています。また、入力パラメータの形式が XML から独自の簡単な YAML ベースの記述形式になっています。

1.3 ライセンス

ライセンスについては、2つのライセンス形態を用意しています。

- VCAD ライセンス

VCAD ライセンスについては、下記からお問い合わせください。

<http://vcad-hpsv.riken.jp/jp/>

- 二条項 BSD ライセンス

[https://ja.wikipedia.org/wiki/BSD ライセンス](https://ja.wikipedia.org/wiki/BSD_ライセンス)

第 2 章

インストール

MPI 通信ライブラリや必要な機能ライブラリなど、FFV-C に必要なライブラリ群のインストールとコンパイル，および FFV-C のコンパイルについて説明します．

FFV-C は図 1.1 に示すようなサブモジュールにより構成されており、各機能はライブラリ化されています。このため、各ライブラリのコンパイルと FFV-C のコンパイルが必要になります。

2.1 MPI ライブラリのインストール

OpenMPI^{*1}のインストールについて説明します。

1. autotools によるコンパイル

autotools(autoconf & automake)^{*2}を用いて作成されたパッケージは容易にインストールができます。典型的な場合、インストールまでの全工程が自動化され、ソースコードを展開した後、以下のコマンドを入力するだけで全てが完了します。

```
./configure && make && make install
```

この時点で autotools のバージョンが違う場合には以下のコマンドを実行し、環境を合わせます。

```
$ aclocal
$ autoconf
$ automake -a

あるいは

$ autoreconf -i
```

2. シェルスクリプトを用いたコンパイル環境の設定

configure のために、次のようなスクリプトを用意して実行します。インストールディレクトリは/opt/openmpi とします。コンパイラは、Intel Compiler icpc/ifort の利用を指定しています。

```
$ cat config_ompi.sh

#!/bin/sh
export CC=icc
export CFLAGS=-O3
export CXX=icpc
export CXXFLAGS=-O3
export F77=ifort
export FFLAGS=-O3
export FC=ifort
export FCFLAGS=-O3
#
./configure --prefix=$1

$ ./config_ompi.sh /opt/openmpi
```

3. コンパイルの実行とインストール

```
$ make
$ sudo make install
```

4. PATH の設定

^{*1} [urlhttp://www.open-mpi.org/](http://www.open-mpi.org/)

^{*2} <http://www.gnu.org/software/autoconf/>

実行時の `mpirun`^{*3} が正しいパスになっているかどうかを `which` コマンドで確認します^{*4}.

```
$ which mpirun
```

5. 環境変数の設定

実行時に必要な環境変数をホームディレクトリの `.bash_profile` などに記述しておきます.

```
export LD_LIBRARY_PATH=/opt/openmpi/lib:$LD_LIBRARY_PATH
export DYLD_LIBRARY_PATH=/opt/openmpi/lib:$DYLD_LIBRARY_PATH
```

2.2 パッケージによるインストール

2.2.1 ffvc_package

`ffvc_package` は各ライブラリと FFV-C アプリケーションを同梱したパッケージです^{*5}. このパッケージを展開すると、下記のようになり、各ライブラリの末尾 3 桁の数字はリリースバージョン番号を示しています.

```
$ tar xvfz ffv_package-2.3.8.tar.gz
$ ls
CDMLib-0.8.1.tar.gz
CPMLib-2.0.5.tar.gz
Cutlib-3.1.2.tar.gz
FFVC-2.3.8.tar.gz
PMLib-3.1.1.tar.gz
Polylib-3.5.3.tar.gz
TextParser-1.6.3.tar.gz
install_fx.sh
install_gnu.sh
install_intel.sh
```

インストールシェルは、ターゲットの環境向けに 3 種類が提供されています. 例えば `install_intel.sh` は、Intel 系の CPU に対して、Intel コンパイラと OpenMPI を用いてコンパイルする場合のシェルスクリプトです.

```
$ cat install_intel.sh
#!/bin/sh
#
#####
#
# FFV-C Install Shell
#
# Copyright (c) 2014 Advanced Institute for Computational Science, RIKEN.
# All rights reserved.
#
#####
#
# Double precision Option
#
# $ ./install_*.sh double
#
#####
# Default Precision
```

^{*3} `mpiexec` でも動きます.

^{*4} Mac OSX の場合にはデフォルトでインストールされている OpenMPI の方を見に行くので、インストールした OpenMPI の PATH を最初の方に書いておきます.

^{*5} http://avr-aics-riken.github.io/ffvc_package/ から入手できます.

```

export PRCSN1=float
export PRCSN2=4

arg=$1

if [ "${arg}" = "double" ]; then
export PRCSN1=double
export PRCSN2=8
fi

#####
# Edit MACRO for your target machine

#export FFV_HOME=hogehoge

export TMP_CCC=mpicc
export TMP_CXX=mpicxx
export TMP_F90=mpif90

#####

# library name
export TP_LIB=TextParser-1.6.3
export PM_LIB=PMlib-3.1.1
export PLY_LIB=Polylib-3.5.3
export CPM_LIB=CPMlib-2.0.5
export CDM_LIB=CDMlib-0.8.1
export FFVC=FFVC-2.3.8

...

```

`install_intel.sh` を用いてコンパイルするには、まず、環境変数とコンパイルオプションを計算機環境に合わせて指定します。

2.2.2 ターゲット環境に応じた指定項目

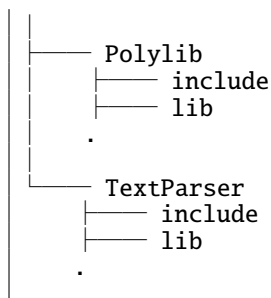
- インストールディレクトリの指定

`export FFV_HOME=hogehoge` にインストールする先のディレクトリを指定します。指定ディレクトリ配下に下記のようなディレクトリ構造が生成されます。

```

FFV
├── CDMlib
│   ├── include
│   └── lib
├── .
├── CPMlib
│   ├── include
│   └── lib
├── .
├── FFVC
│   ├── bin
│   │   ├── combsph
│   │   ├── ffvc
│   │   └── ffvc-config
│   ├── doc
│   └── include
├── .
└── PMlib
    ├── include
    └── lib

```



- コンパイラの指定

使用するコンパイラは、MPI ライブラリが提供するラッパーコンパイラを指定します。提供されるインストールシェルとは異なる環境の場合に、適切なコンパイラを指定してください。

- 倍精度計算

倍精度計算を行う場合には、インストールシェルの起動時に倍精度オプションを指定します。

```
$ ./install_*.sh double
```

2.3 個別ライブラリ/アプリのインストール

2.3.1 ライブラリのインストール

あるライブラリだけアップデートされた場合には、提供されるライブラリ毎にコンパイルを行うこともできます。インストール方法は各ライブラリのディレクトリ内にある `INSTALL` を見てください。autotools によりパッケージ化していますので、前述のような `configure && make && make install` でコンパイルとインストールができます。また、`Makefile.hand` も提供されていますので、手動でのコンパイルも可能です。

ライブラリをアップデートした場合には、FFV-C アプリケーション本体も再コンパイル（あるいは再リンク）してください。

インストールされたライブラリのバージョンは、Polylib の場合、下記の方法で調べることができます。

```
$ cd /usr/local/Polylib/bin
$ polylib-config --version

Polylib - Polygon Management Library Ver. 2.6.4 (20130627_2359)

Copyright (c) 2010-2011 VCAD System Research Program, RIKEN.
All rights reserved.

Copyright (c) 2012-2013 Advanced Institute for Computational Science, RIKEN.
All rights reserved.
```

2.3.2 FFV-C のコンパイル

本節では、FFV-C のコンパイルとインストールについて説明します。開発中の最新アーカイブ `FFVC-x.x.x.tar.gz` は、<https://github.com/avr-aics-riken/FFVC> から入手できます。

- アーカイブの解凍

```
$ tar xvfz FFVC-x.x.x.tar.gz
```


解凍すると、以下のようなファイル構成になります*⁶。

```

FFVC-x.x.x
├── AUTHORS          開発者のリスト
├── BUILD_DIR        ビルドするディレクトリ
├── COPYING          コピーライト
├── ChangeLog        リビジョンの履歴
├── INSTALL          インストール方法
├── LICENSE          ライセンスファイル
├── Makefile.am
├── Makefile.in
├── Makefile_hand
├── NEWS            リリースの履歴
├── README          最初に見るべきファイル
├── aclocal.m4
├── config
│   ├── compile
│   ├── depcomp
│   ├── install-sh
│   └── missing
├── config.h.in
├── configure
├── configure.ac
├── doc
│   ├── Makefile
│   ├── Makefile.am
│   ├── Makefile.in
│   ├── ffvc_ug.pdf   ドキュメント
│   └── readme.pdf    メモ
├── doxygen
│   └── Conf
│       ├── FB
│       ├── FFVC
│       ├── IP
│       ├── Utility
│       └── makefile
├── examples
│   ├── 2Dcavity
│   ├── 2Dcyl
│   ├── 3DCavity      三次元のキャビティフロー（バイナリモデル）
│   ├── BackStep      バックステップ流れ
│   ├── Cylinder      柱状体周りの流れ
│   ├── HeatedPlate    加熱平板
│   ├── Jet            噴流
│   ├── LDC            辺長比 1:1:2 のキャビティフロー（実験値との比較）
│   ├── PMT            性能測定用
│   ├── SHC1D          片持ち梁の定常熱伝導
│   ├── Sphere         球周りの流れ
│   └── ThermalCavity   キャビティの温度計算
├── ffvc-config.in
├── src
│   ├── ASD            Active Sub-Domain クラス
│   ├── FB            FlowBase クラス
│   ├── FFV            FFV-C クラス
│   ├── FILE_IO        ファイル入出力クラス
│   ├── F_CORE         Fortran のコアプログラム
│   ├── F_LS           線形ソルバーの Fortran ファイル
│   ├── F_VOF          VOF クラスの Fortran ファイル
│   ├── Geometry       形状処理関連クラス
│   ├── IP             組み込み例題クラス群
│   ├── Makefile.am
│   ├── Makefile.in
│   ├── Makefile_ffvc
│   ├── Makefile_hand
│   ├── Util_Combsph    分散ファイルの結合ユーティリティ
│   └── main.C

```

*⁶ doxygen ディレクトリについては、doxygen ファイルを生成するために必要な設定ファイルのみを提供しています。Conf ディレクトリ内で make を実行すると各ディレクトリに doxygen ファイルが生成されます。

```
└── make_setting
```

- **configure オプション**

提供するライブラリと FFV-C アプリケーションは、autotools を使ってパッケージしています。このために、configure 時に計算機環境に応じて適切にオプションを与える必要があります。各計算機向けのオプションの例については、展開したライブラリ内にある INSTALL を参照してください。

make install を行くと、指定したディレクトリにライブラリやアプリケーションがインストールされます。

第 3 章

基礎方程式と解析方法

FFV-C が扱う流体の基礎方程式について簡単に説明します．詳細は FFV-C の詳細な説明書 (`inside_ffvc.pdf`) を参照してください．

3.1 支配方程式

FFV-C ソルバーは、圧力や温度の変化により生じる流体の密度変化が小さく、代表的な流速が音速に比べてかなり低い場合を仮定して、非圧縮性流体および弱圧縮性の基礎方程式を用いています。

3.1.1 非圧縮性流体

解析対象とする流れの特徴を以下のように仮定し、支配方程式を記述します。

- 流れの速度が音速に比べて十分に低く、流れの運動に対する圧縮性の影響は小さいと仮定して、流れを非圧縮性として取り扱います。
- 温度場の代表的な温度差スケールが 30 °C 以下で、密度変化が小さいと仮定します。この場合、密度変化が質量保存則へ与える影響は小さく、密度変化が流れの運動に及ぼす影響を Boussinesq 近似によりモデル化できます。

支配方程式として、非圧縮性流れに対する質量保存則式 (3.1)、運動量保存則式 (3.2)、エネルギー保存則式 (3.3) を用います。 δ はクロネッカーのデルタで重力方向 ($i=3$) のときに浮力が作用します。ここで、プライム [$'$] は有次元量を表します。

$$\frac{\partial u_i'}{\partial x_i'} = 0 \quad (3.1)$$

$$\rho' \frac{\partial u_i'}{\partial t'} + \rho' \frac{\partial}{\partial x_j'} \{ (u_j' - u_j^{g'}) u_i' \} = - \frac{\partial P'}{\partial x_i'} + \frac{\partial}{\partial x_j'} \left[\mu \left(\frac{\partial u_i'}{\partial x_j'} + \frac{\partial u_j'}{\partial x_i'} \right) \right] - \rho' g' \delta_{i3} \quad (3.2)$$

$$\rho' C_p' \left[\frac{\partial \theta'}{\partial t'} + \frac{\partial}{\partial x_i'} \{ (u_i' - u_i^{g'}) \theta' \} \right] = \frac{DP'}{Dt'} + \frac{\partial}{\partial x_i'} \left(\lambda' \frac{\partial \theta'}{\partial x_i'} \right) + \mu' \Phi' + Q' \quad (3.3)$$

ρ'	[kg / m ³]	density
P'	[Pa]	pressure
C_p'	[J / (kg K)]	specific heat at constant pressure
θ'	[K]	temperature
λ'	[W / (m K)]	heat conductivity
u_j'	[m / s]	velocity components
$u_j^{g'}$	[m / s]	velocity components of a grid point
x_j'	[m]	coordinate axis
t'	[s]	time
μ'	[Pa s]	viscosity
g'	[m / s ²]	gravitational acceleration
Φ'	[1/s ²]	dissipation function
Q'	[W / m ³]	heat source

式 (3.2) は形式的に ALE(Arbitrary Lagrangian and Eulerian) で書かれていますが、速度 $u_j^{g'}$ で移動する格子系での保存則を表現しています。格子点を固定 ($u_j^{g'} = 0$) すれば Euler 表現、流体粒子と一緒に移動 ($u_j^{g'} = u_j'$) させれば Lagrangian

表現となります。ここでは、並進や回転などの任意の格子移動速度を与えるために $u_j^{g'}$ を利用します。

低マッハ数を仮定すると、散逸関数 Φ は M^2 に比例するので、その寄与は小さく圧力の全微分の項の影響も小さいので、式 (3.3) は、次のようなパッシブスカラの移流拡散方程式になります。

$$\rho' C_p' \frac{\partial \theta'}{\partial t'} + \rho' C_p' \frac{\partial}{\partial x_i'} \left\{ (u_i' - u_i^{g'}) \theta' \right\} = \frac{\partial}{\partial x_i'} \left(\lambda' \frac{\partial \theta'}{\partial x_i'} \right) + Q' \quad (3.4)$$

ここで α は温度拡散係数で $[m^2/s]$ の単位です。

$$\left. \begin{aligned} \alpha' &= \frac{\lambda}{\rho' C_p'} & [m^2/s] \\ \nu' &= \frac{\mu'}{\rho'} & [m^2/s] \\ p' &= \frac{\tilde{p}'}{\rho_0'} & [m^2/s^2] \end{aligned} \right\} \quad (3.5)$$

温度拡散係数 α が一定の場合には下記のようにになります。

$$\frac{\partial \theta'}{\partial t'} + \frac{\partial}{\partial x_i'} \left\{ (u_i' - u_i^{g'}) \theta' \right\} = \alpha' \frac{\partial}{\partial x_i'} \left(\frac{\partial \theta'}{\partial x_i'} \right) + \frac{Q'}{\rho' C_p'} \quad (3.6)$$

3.1.2 非圧縮性流体への弱い圧縮性の導入

一般的に、流れの代表的な速度が音速の 0.2~0.3 程度までは運動方程式に対する圧縮性の影響は小さいと言われています。しかしながら、代表速度は小さくても流体の圧縮性が本質的な影響を及ぼす場合があります。例えば、大温度差で駆動される自然対流は空気の体積膨張のため、また、空洞部をもつ形状の流れではヘルムホルツ共鳴を伴うので、音波を扱えるように連続の式において圧縮性を考慮する必要があります。

ここでは、低マッハ数における弱い圧縮性を考慮した定式化を考えます [1]。圧縮性を考慮した連続の式は、

$$\frac{\partial \rho'}{\partial t'} + \frac{\partial}{\partial x_i'} (\rho' u_i') = 0 \quad (3.7)$$

この式を実質微分を用いて変形し、 c を音速として等エントロピー変化を仮定すると、式 (3.7) は、

$$\frac{1}{c^2} \frac{Dp'}{Dt'} + \rho' \frac{\partial u_i'}{\partial x_i'} = 0 \quad (3.8)$$

さらに、圧力の実質微分項では空間微分の寄与が小さいと仮定すると、次式のようにになります。

$$\frac{1}{c^2} \frac{\partial p'}{\partial t'} + \rho' \frac{\partial u_i'}{\partial x_i'} = 0 \quad (3.9)$$

3.2 無次元化

代表速度 u'_0 , 代表長さ L' , 代表温度スケール $\Delta\theta'_0$ と基準温度 θ'_0 で式 (3.1), (3.2), (3.6) を無次元化します.

$$\left. \begin{aligned} u &= \frac{u'}{u'_0} \\ x &= \frac{x'}{L'} \\ p &= \frac{p' - p'_0}{\rho' u'^2_0} \\ \theta &= \frac{\theta' - \theta'_0}{\Delta\theta'_0} \end{aligned} \right\} \quad (3.10)$$

3.2.1 非圧縮

以下の式 (3.11)–(3.13) は, 単一成分の熱流動を表します.

$$\frac{\partial u_i}{\partial x_i} = 0 \quad (3.11)$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j} \left\{ (u_j - u_j^s) u_i \right\} = -\frac{\partial p}{\partial x_i} + \frac{1}{Re} \frac{\partial}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + \frac{Gr}{Re^2} \delta_{i3} \theta \quad (3.12)$$

$$\frac{\partial \theta}{\partial t} + \frac{\partial}{\partial x_i} \left\{ (u_i - u_i^s) \theta \right\} = \frac{1}{Pe} \frac{\partial}{\partial x_i} \frac{\partial \theta}{\partial x_i} + \Theta \quad (3.13)$$

$$\left. \begin{aligned} Re &= \frac{u'_0 L'}{\nu'} \\ Pr &= \frac{\mu' C'_p}{\lambda'} = \frac{\nu'}{\alpha'} \\ Gr &= \frac{g' \beta' \Delta\theta'_0 L'^3}{\nu'^2} \\ Ra &= Pr \cdot Gr \\ Pe &= Pr \cdot Re \\ \Theta &= \frac{Q'}{\rho' C'_p u'_0 \Delta\theta'_0 / L'} \end{aligned} \right\} \quad (3.14)$$

ここで,

Pr	Prandtl 数	粘性と熱の拡散率の比
Re	Reynolds 数	慣性力と粘性力の比
Gr	Grashof 数	浮力と粘性力の比
Ra	Rayleigh 数	不安定性のパラメータ
Pe	Peclet 数	対流と熱伝達のエネルギー輸送の比
Θ	-	無次元の温度変化率

式 (3.12) は強制対流と自然対流を表現し、右辺第三項が自然対流と強制対流の比を表しています。つまり、 $Gr/Re^2 \gg 1$ の場合には自然対流が支配的で、 $Gr/Re^2 \ll 1$ の場合には強制対流が支配的となります。 $Gr = 0$ つまり温度差が無い場合には純強制対流です。一方、 $Gr/Re^2 \rightarrow \infty$ の場合には純自然対流で、流れは浮力によって駆動されるため代表速度が自明ではありません。また、 $Gr > 10^9$ となるような流れは非定常性が強くなります。

無次元化パラメータの選択

FFV-C ソルバーは、支配方程式を無次元化して解いています。このため、無次元化のパラメータを選択する必要がありますが、解くべき現象に応じて適切に選択します。

■純強制対流 式 (3.12) においては $Gr = 0$ なので Re が支配パラメータとなります。無次元化のスケージングは、 $u'_0, L', \nu', \alpha' (= \lambda'/\rho' C'_p)$ を与えます。

■熱対流

浮力の効果を考慮しない場合 式 (3.12) において、純強制対流と同じく $Gr = 0$ です。式 (3.13) では Pe が支配パラメータとなります。無次元化のスケージングは、 $u'_0, L', \nu', \alpha' (= \lambda'/\rho' C'_p)$ を与えます。

浮力の効果を考慮する場合 式 (3.12) では Gr, Re が、式 (3.13) では Pe が支配パラメータとなります。無次元化のスケージングは、 $u'_0, L', \Delta\theta'_0, \beta', g', \nu', \alpha', Pr$ を与えます。

■純自然対流 浮力の効果を考慮した熱対流と同じです。ただし、 u'_0 は自明でないので、純自然対流の場合の代表流速はスケールアナリシスから推測され [2], Pr 数が小さい場合は次式のように見積もることができます。

$$u'_0 = \sqrt{g' \beta' \Delta\theta'_0 L'} \quad (3.15)$$

自然対流の場合の代表速度は式 (3.15) の関係を用いて見積もり、代表速度パラメータとして与えます。自然対流と強制対流が共存する共存対流の場合には、各々の代表スケールの平均値や大きい方の値を代表速度とします。

■固体熱伝導 式 (3.13) の形式で Pe が支配パラメータとなります。ただし、対流項の寄与はありません。無次元化のスケージングは、 $L', \Delta\theta'_0, \alpha'$ を与えます。 u'_0 には、一般に熱輸送の時間スケールと代表速度は熱流の伝播速度に相当すると考え、次式を用います。

$$u'_0 = \frac{\alpha'}{L'} \quad (3.16)$$

■共役熱移動 共役熱移動は、固体中の熱移動と流体中の熱流動を同時に扱うので、必然的に多媒質の熱移動問題となります。熱流動は浮力効果を考慮しています。

3.2.2 弱圧縮形式

式 (3.9) において、密度はほぼ一定で ρ'_0 とすると、

$$\frac{1}{\rho'_0 c^2} \frac{\partial p'}{\partial t'} + \frac{\partial u'_i}{\partial x'_i} = 0 \quad (3.17)$$

これを無次元化すると、

$$M^2 \frac{\partial p}{\partial t} + \frac{\partial u_i}{\partial x_i} = 0 \quad (3.18)$$

ここで、マッハ数 $M = u'/c$ です。

3.3 解法アルゴリズム

前節の支配方程式に対して、非圧縮性流体の解法に使われる分離解法を適用し、有限体積法で離散化します。

3.3.1 Fractional Step 法

非圧縮性の Navier-Stokes 方程式 (3.12) の解法として、Fractional step 法を用います。これは、任意のベクトル場が非回転場と湧き出し無しの直交するベクトル場に分解できる性質を利用して、二つのベクトルの和をとることにより解を求める分離解法です。

離散式のコーディングポリシーとして、各セル単位で計算を進めていきます。保存的な支配方程式を解くのでセル界面の流束を評価する方法が素直で演算量も少なくなりますが、コロケートでは固体面や境界面の処理を考える上でセル単位毎の方が計算処理がしやすいからです。

Euler Explicit

一次精度の時間進行法です。

■ Navier-Stokes equations

式 (3.12) の対流項と粘性項をそれぞれ C_i , D_i , 浮力項を外力 f_i で表すと、

$$\left. \begin{aligned} \frac{\partial u_i}{\partial t} + C_i &= -\frac{\partial p}{\partial x_i} + D_i + f_i \\ C_i &= \frac{\partial}{\partial x_j} \{ (u_j - u_j^g) u_i \} \\ D_i &= \frac{1}{Re} \frac{\partial}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \\ f_i &= \frac{Gr}{Re^2} \delta_{i3} \theta \end{aligned} \right\} \quad (3.19)$$

疑似ベクトルの予測式は、

$$u_i^* = u_i^n + \Delta t (D_i^n - C_i^n + f_i^n) \quad (3.20)$$

連続の式による拘束条件から、圧力の Poisson 方程式は、

$$\frac{\partial}{\partial x_i} \frac{\partial p^{n+1}}{\partial x_i} = \frac{1}{\Delta t} \frac{\partial u_i^*}{\partial x_i} \Big|_{face} \quad (3.21)$$

圧力ポテンシャルによるセルセンターとスタガード位置の速度ベクトルの修正式は,

$$u_i^{n+1} = u_i^* - \Delta t \frac{\partial p}{\partial x_i}^{n+1} \quad (3.22)$$

$$u_{i,face}^{n+1} = u_{i,face}^* - \Delta t \frac{\partial p}{\partial x_i}^{n+1} \quad (3.23)$$

ここで, $u_{i,face}^*$ はセルセンタの値 u_i^* から内挿します.

■Thermal transport equation

式 (3.13) の移流項と拡散項をそれぞれ Cs_i , Ds_i で表すと,

$$\left. \begin{aligned} \frac{\partial \theta}{\partial t} + Cs_i &= Ds_i + \Theta \\ Cs_i &= \frac{\partial}{\partial x_i} \left\{ (u_i - u_i^g) \theta \right\} \\ Ds_i &= \frac{1}{Pe} \frac{\partial}{\partial x_i} \frac{\partial \theta}{\partial x_i} \end{aligned} \right\} \quad (3.24)$$

$$\theta^{n+1} = \theta^n + \Delta t (Ds_i^n - Cs_i^n + \Theta^n) \quad (3.25)$$

Adams-Bashforth

二次精度ですが, 安定条件が厳しく, CFL 数は 0.1 程度です.

■Navier-Stokes equations

$$u_i^* = u_i^n + \Delta t \left[\frac{1}{2} \left\{ 3(D_i^n - C_i^n) - (D_i^{n-1} - C_i^{n-1}) \right\} + \frac{1}{2} (3f_i^n - f_i^{n-1}) \right] \quad (3.26)$$

■Thermal transport equation

$$\theta^{n+1} = \theta^n + \Delta t \left[\frac{1}{2} \left\{ 3(Ds_i^n - Cs_i^n) - (Ds_i^{n-1} - Cs_i^{n-1}) \right\} + \frac{1}{2} (3\Theta^n - \Theta^{n-1}) \right] \quad (3.27)$$

3.3.2 連続の式と圧力 Poisson の二重反復

オリジナルの HSMAC 法 [3] は, 速度の予測子に対して圧力と速度修正を同時に行う反復法であり, ポアソンを解かずスタガード変数配置上でニュートン反復を用いるアルゴリズムです. この HSMAC をコロケート変数配置に適用し, さらに毎反復毎に Poisson 方程式の右辺項を計算する方法 C-HSMAC 法 [4] が提案されています. HSMAC, C-HSMAC のアルゴリズムの利点の一つは, 収束判定条件として連続の式をそのまま適用できる点です. この点は, 収束条件に物理的な意味を与えるので計算精度をモニタする上で有用です.

一方, 反復毎にソース項の計算を行うと, 圧力の収束すべき方向を定めるソース項が毎回振れ, 計算量が増える恐れがあります. そこで, FFV-C では, ソースを固定する Fractional Step 法ベースでアルゴリズムを構成します. 補助的にスタガード位置での速度成分を配置して連続の式を評価し, 収束判定条件として連続の式を用います. このため, 連続の式を満たすまで反復するループの中で, 圧力の Poisson 反復を行う, 二重反復の形式をとります.

セルセンターの速度を $^c\mathbf{u}$, セルフェイスの速度を $^f\mathbf{u}$ とすると, 解くべき式は次のようになります.

$$\frac{\partial ^c\mathbf{u}^{n+1}}{\partial t} + \nabla(^c\mathbf{u}^c\mathbf{u})^n = -\nabla p^{n+1} + \frac{1}{Re} \nabla^2 ^c\mathbf{u}^n \quad (3.28)$$

コロケート配置では、対流項と粘性項の評価をセルセンタで行い、その値をセルフェイスに内挿します。このため、複雑な計算でも同じ定義点での値に対して行えるので、スタガード格子に比べて簡単になります。一方で、境界条件はセルセンタとセルフェイスの両方に対して必要になります。

境界条件は、 $f\mathbf{u}$, $c\mathbf{u}$ に対して必要となります。 $f\mathbf{u}$ に対しては、計算領域内の $\nabla\mathbf{u}$ 、つまり、Poisson のソース項を計算するために必要な境界条件です。

第 4 章

解析モデルの作成

解析モデルの作成方法を説明します。解析モデルの作成については、FXgen アプリケーションを用いて、ポリゴンにラベルと境界条件を付与します。また、形状データが不要な組み込み例題について説明します。

4.1 大規模並列格子生成の問題点と方針

流体計算のプロセスを分類すると、前処理、計算、後処理の3つの処理になります。このうち、前処理である格子生成は、流体計算のプロセスの中で最も工数のかかる部分です。特に、非常に複雑な形状を計算対象とする場合には、格子生成の時間割合が全プロセスの半分以上になることもあります。また、大規模な並列計算を実施する場合は、PCで大規模な計算格子を作成した後、作成したファイルを計算サーバーへ転送する作業にも時間がかかります。

したがって、複雑な形状まわりの流れを大規模並列計算するために、新しい格子生成方法が必要となります。大規模計算では、従来の格子生成のようにPCやWS上で動作させるGUIプログラムは搭載メモリサイズの制限から必要な規模の格子を生成することができません。

そこで、PCで格子は生成せず、流体計算を開始した直後のプログラムの初期化処理時に格子を作成します。この方法は、領域分割に基づいた並列処理で格子を作成できるので計算効率の点で優れています。しかしながら、アルゴリズムだけで複雑形状の格子生成に対応することは難しく、ユーザの判断が必要となる部分が出てきます。FFV-Cでは、格子生成の前処理としてFXgenプログラムを用いて自動格子生成のヒントを作成し、このヒントを元に並列で格子を作成する2段階の方法を採用します。

直交格子法では、物体表面から格子定義点までの距離を計算し、計算空間内の任意の位置にある物体を認識するような計算スキームを構築します。つまり、距離情報に基づいた物体認識のスキームが従来の格子生成の代わりとなります。そのため、非常にロバストな前処理アルゴリズムを開発する必要があります。また、同時に並列で動作するアルゴリズムにすることが要請されます。FFV-Cでは、図4.1に示すように、2ステージの処理により格子生成を行い、これらの2つの要素を両立するアルゴリズム開発を行っています。

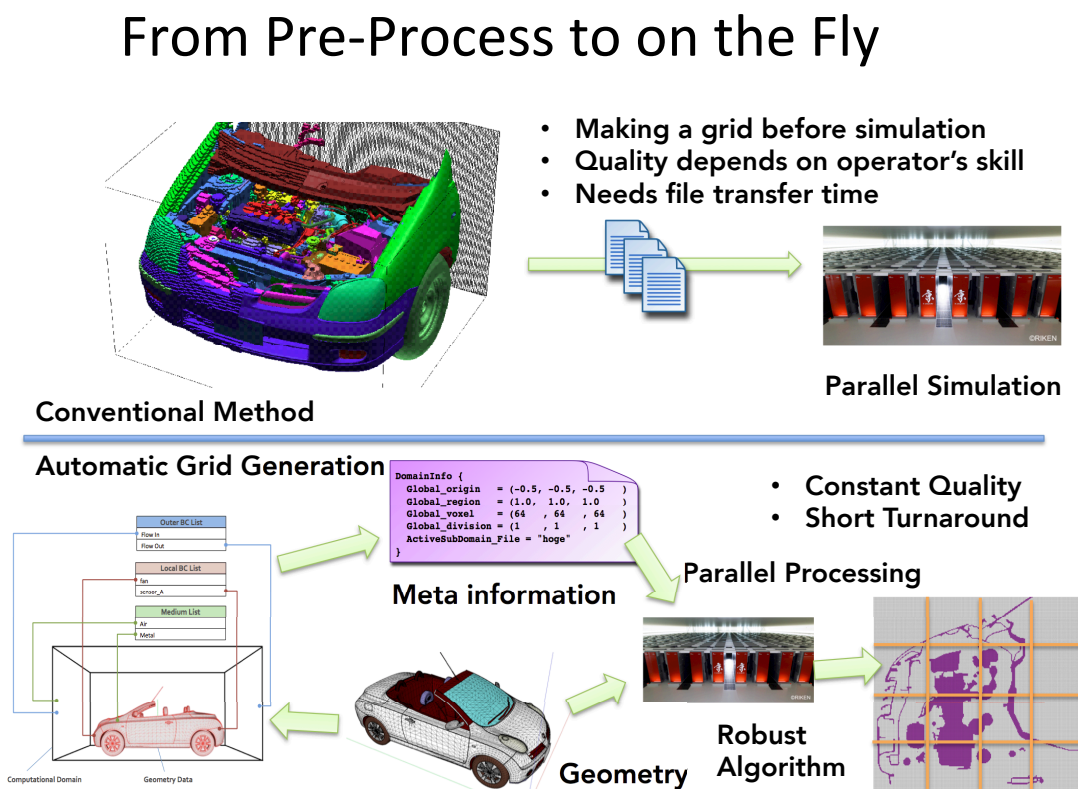


図4.1 自動格子生成のながれ。図の上部は従来のPC上のGUIプログラムを利用する方法。下部はFFV-Cで採用する2ステージの自動格子生成。

4.2 形状近似

直交格子を用いる流体解析では、解析対象となる形状の近似度を直交格子上でどのように考慮するかにより、計算のロバスト性、予測精度、計算時間、計算格子（解析モデル）の作りやすさなどの特性が異なります。一般には、図 4.2 のように分類することができます [5]。

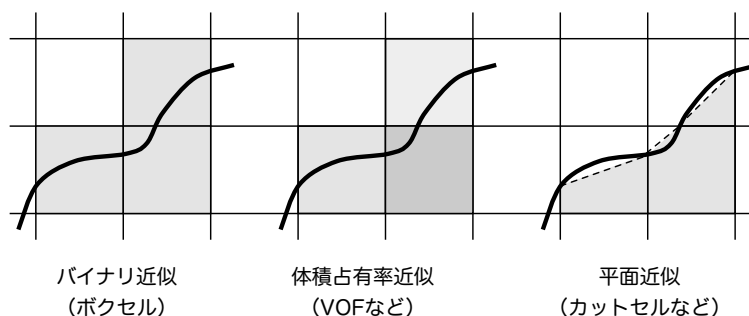


図 4.2 直交格子における形状近似度の分類

Binary Voxel モデルは、立方体のセル要素単位で形状を表現する解析モデルです。物体の形状近似としては最も簡単であり、モデル作成時のロバスト性に大きな利点があります。一方で、階段状の近似のため、流れの再現性が悪い場合もあります。

体積率モデルは、Binary Voxel の形状近似度を改善する方法の一つで、セル内における流体の占有率を考慮した計算をする場合に利用します。FFV-C ソルバーでは、圧力損失部などの境界条件の実装に用いています。

距離情報モデルは Binary Voxel に比べて流れの再現性を大きく改善できます。この形状表現モデルでは、格子の定義点から物体表面までの距離を用いて流束計算を行うスキームを用います。

解析対象となる物体形状を表す STL/Obj ファイルを入力データとして、FXgen は解析領域や格子サイズ、領域分割の参考値などを決めます。FXgen に組み込んでいる領域分割アルゴリズムは、FFV-C で利用されている CPMlib の領域分割アルゴリズムと同一であるので、FFV-C の初期時の領域分割は FXgen で試行したパラメータが再現できます。

出力パラメータとしては、下記に示すような領域サイズ、基点、分割数などになります。

```
DomainInfo {
  UnitOfLength      = "M"
  GlobalOrigin       = (-2.0, -1.0, 0.0)
  GlobalRegion       = (5.0, 2.0, 2.0)
  GlobalPitch        = (2.0e-2, 2.0e-2, 2.0e-2)
  //GlobalVoxel       = (200, 200, 200)
  //GlobalDivision    = (1, 1, 4)
```

4.3 解析モデルの作成

4.3.1 形状データの準備

形状を表すデータとして、STL フォーマットが利用できます*1。STL ファイルは、CAD やほかのモデラーから出力できます*2。

*1 形状データをハンドリングする Polylib 自体は STL, Obj のフォーマットをロード可能です。

*2 データ量が多くなる場合にはバイナリ形式を用います

FFV-C は様々な境界条件を指定できますが、境界条件は STL ファイル単位で指定します（具体的な境界条件の指定方法は6章の境界条件をご覧ください）。例えば、ダクト形状があり、その入り口と出口を境界条件で指定することを考えましょう。この場合、ダクト本体、入り口、出口の3つの STL ファイルが必要となります。

形状データの品質については、水密なデータであれば申し分ありませんが、FFV-C は非水密な形状データでも計算可能です。また、計算格子幅以下の穴は計算格子上に反映されず、水密なデータとして計算されることもあります。一方で、計算格子幅以上大きさの穴は、そのまま穴のあいた形状として計算することになります。

4.3.2 計算領域の設定

計算する対象の形状データが準備できると、次は計算領域を決めます。この作業のために、前処理用のアプリケーション FXgen を利用します。FXgen は形状データを読み込み、表示し、計算領域の大きさや分割数などを検討するアプリケーションです。従来の格子生成アプリとは異なり、格子自体の作成はしません。代わりに、格子生成に必要なパラメータを決定し、その情報をテキストファイルに出力します。FFV-C は、この計算領域情報と形状データを読み込み、計算格子を作成した後、流体計算を開始します。

FFV-C は自動生成した格子ファイルを圧縮して出力します。FXgen はこの圧縮格子ファイルを読み込む機能を備えており、ある程度の大規模データでもその様子を確認することができます。

4.4 組み込み例題

組み込み例題は、FFV-C ソルバーに組み込み済みの解析モデルです。解析モデルをプログラムによって生成するため、解析モデル形状データを作成しなくても計算ができます。ただし、表 4.1 に示すようにパラメタライズできる簡単な形状のモデルに限られます。組み込み例題の指定は Geometry セクションの Source で指示し、各モデルに固有のパラメータは IntrinsicExample セクションで指定します。

表 4.1 組み込み例題クラス

組み込みモデル名	利用クラス	説明
Rectangular	IP_Rect	計算領域が矩形で、かつ単一媒質のモデル
PerformanceTest	IP_PMT	性能測定を行うためのモデル（三次元立方体キャピティフローと同じ問題設定）
BackStep	IP_STEP	バックステップ流れのモデル
Cylinder	IP_CYLINDER	角柱と円柱周りの流れのモデル
Duct	IP_Duct	円形と正方形のダクト流れのモデル
Polygon	Intrinsic	距離情報スキームを用いる場合に、入力するポリゴンファイル名と DomainInfo の指定だけで計算するモデル

4.4.1 IP.Rect クラス

IP.Rect クラスは三次元の矩形の計算領域を表現するクラスです。計算領域は、次のように Domain ファイルで指定します。ここでは、各方向の格子幅を指定し、基点座標と計算領域の大きさを指定しています。

```
DomainInfo {
  UnitOfLength   = "M"
  GlobalOrigin   = (-0.5, -0.5, -0.5 )
  GlobalRegion   = (1.0, 1.0, 1.0 )
  GlobalPitch    = (1.5625e-02, 1.5625e-02, 1.5625e-02)
}
```

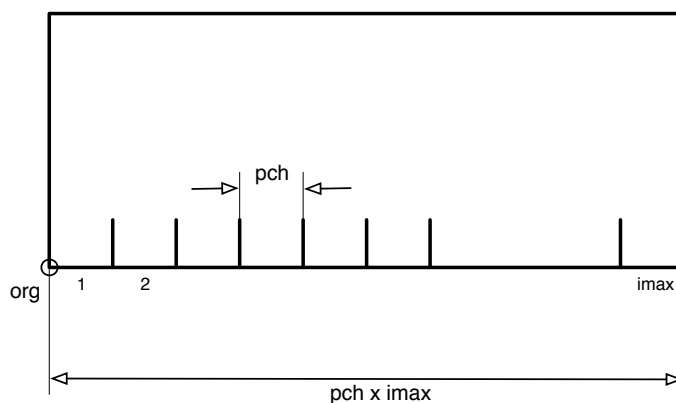


図 4.3 Rectangular クラスのパラメータ設定

表 4.2 IntrinsicExample セクションで設定可能なパラメータ

指定パラメータ	指定値	意味
CheckEven	"Yes" "No"	分割数が偶数であるかどうかをチェックする
FluidMedium	MediumTable のラベル名	流体の媒質
SolidMedium	MediumTable のラベル名	固体の媒質

表 4.2 のパラメータは、IP.Rect クラスに固有の設定項目で、IntrinsicExample セクションで指定します。

```
IntrinsicExample {
  FluidMedium = "air"
  SolidMedium = "fe"
  CheckEven   = "yes"
}
```

上記のパラメータ設定では、分割数の偶数チェックを行い、MediumTable において Air, Fe のラベル名で指定されている物性値を、それぞれ流体と固体として参照しています。

4.4.2 IP_PMT クラス

FFV-C ソルバーの基本的性能を測定するための例題クラスです。三次元立方体の空間内のキャビティフローを解きます。性能測定モードとなり、圧力反復の収束判定は行わず、反復回数は固定となります。また、初期化時のファイル出力が抑制されます。

4.4.3 IP_CYLINDER クラス

二次元と三次元の円柱・角柱まわりの流れを計算するクラスです。図 4.4(a) に三次元の単一角柱，図 4.4(b) に二次元の二角柱（円柱）の様子を示します。表 4.4.3 に指定できるパラメータを示します。

- 二次元の問題を解く場合には、Dimension=2D を指定、VoxelSize は kmax=3 となるようにしてください。二次元は xy 平面で角柱の配置を指定します。また、Geometry セクションの FillDirectionControl には、 z 方向を”Suppress”としてください。
- 三次元では、角柱・円柱の底面は $z-$ 方向に接しています。 $z+$ 方向に長さのパラメータを指定してください。
- MediumTable で指定する媒質については、二次元では固体は一つ以上、三次元では二つ以上のリストアップが必要です。

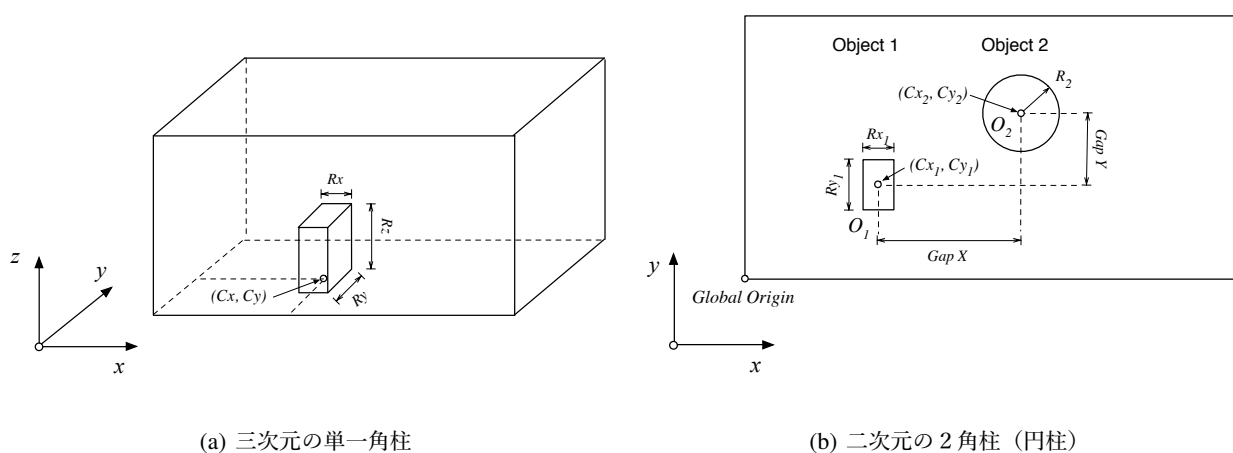


図 4.4 角柱と円柱の計算モデル。添え字 1,2 は、それぞれ前方物体と後方物体を示します。

表 4.3 柱状物体の指定パラメータ

記号	パラメータ
FluidMedium	流体の媒質名
Dimension	"2D" "3D"
UseCylinder	"Yes" "No"
Shape	"Rectangular" "Circular"
LengthX	角柱指定時の角柱の x 方向長さ R_x
LengthY	角柱指定時の角柱の y 方向長さ R_y
Radius	円柱指定時の半径
LengthZ	角柱・円柱の z 軸方向長さ R_z
PositionX	中心位置座標 C_x
PositionY	中心位置座標 C_y
SolidMedium	角柱・円柱に割り当てる固体媒質名

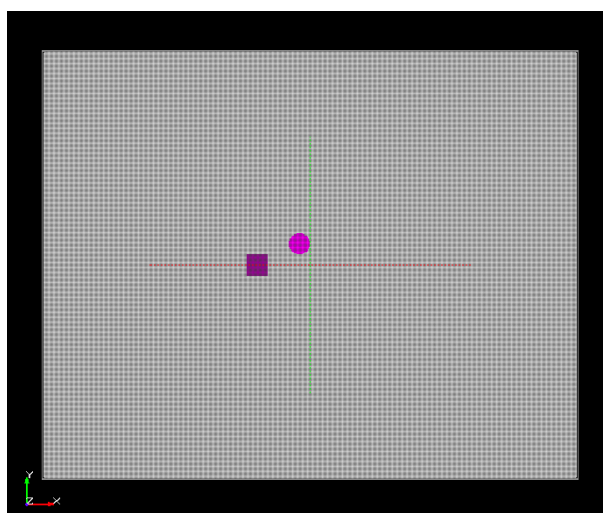


表 4.4 二次元の円柱と角柱のモデル。

下記に、二次元の角柱と円柱の場合のパラメータ例（図 4.4）を示します。IntrinsicExample セクションにパラメータを記述します。

```
IntrinsicExample {
  FluidMedium = "air"
  Dimension   = "2d"

  Cylinder1 {
    UseCylinder = "yes"
    Shape = "rectangular"
    LengthX = 1.0
    LengthY = 1.0
    PositionX = 0.0
    PositionY = 0.0
    SolidMedium = "fe1"
  }

  Cylinder2 {
    UseCylinder = "yes"
    Shape = "circular"
    Radius = 0.5
    PositionX = 2.0
    PositionY = 1.0
    SolidMedium = "fe2"
  }
}

GeometryModel {
  Source = "cylinder"
  FillMedium = "air"
  HintOfFillSeedDirection = "Xminus"
  HintOfFillSeedMedium = "air"
  VoxelOutput = "svx"
  FillDirectionControl = ("fill", "fill", "suppress")
  OutputGlyph = "off"
}
```

4.5 例題

ソースファイルの Example ディレクトリに含まれる例題について説明します。提供される例題は、組み込みモデルや簡単なボクセルモデルを同梱した例題群で、基本的な流れやソルバーの検証のために用意されています。

表 4.5 組み込み例題

Example	Class	Comment
Cavity flow 3D (Cube)	IP_Rect	三次元立方体キャビティフロー
LDC112	IP_Rect	Guermond の実験に対応する辺長が 1:1:2 のキャビティフロー
PMT	IP_PMT	性能測定を行うための例題（三次元立方体キャビティフローの例題と同じ）

表 4.6 サンプル例題

Example	Comment
---------	---------

第 5 章

入力パラメータ

FFV-C の入力パラメータは，記述性とマルチプラットフォームでの稼働を考慮し，軽量の簡易パースライブラリ（TextParser）を利用しています．構文は階層化され，テキストで容易に記述できます．本章では，FFV-C の制御と物理パラメータについて説明します．

5.1 パラメータの記述構文

TextParser ライブラリが扱うパラメータデータベースは、下記のような構造になっています。記述構造として、ノードとリーフがあります。リーフはラベルと値のペアをもち、ノードは値を持ちません。TextParser ライブラリの Examples ディレクトリにパラメータの様々な記述例があるので、参考にしてください。

```
TimeControl { // ノード
  Acceleration { //ノード
    TemporalType      = "Time" // リーフ=ラベルと値のペア
    AcceleratingTime = 0.0
  }

  TimeStep {
    Mode          = "CFLReferenceVelocity"
    DeltaT        = 0.1
  }

  Session {
    TemporalType   = "step"
    Start          = 0
    End            = 100
  }

  Average {
    TemporalType   = "step"
    Start          = 0
    End            = 0
  }
}
```

上記の TimeStep ノードは TimeControl ノード内にあります。このようなパラメータを、ディレクトリ表現のアナロジーを用いて /TimeControl/TimeStep と表記することにします。

TextParser の簡単な記述法、特徴は以下のようになります。詳細は、<http://avr-aics-riken.github.io/TextParser/> を参照してください。

- YAML, json ライクなパラメータ記述方式で、習得が容易。
- パラメータを階層構造的に記述可能。
- 右辺（値）の文字列はダブルクォーテーションで囲む。
- 文字列比較では大文字と小文字の区別はしない。
- 数字はそのまま記述。指数形式もサポート。
- ベクトル要素の記述が簡単。
- limits.h で定義されている Numerial Limits の各値が記述可能。⇒ INT_MAX
- 簡単な依存関係処理が記述可能。
- C 言語のように // や /* ~ */ でコメントが可能。

5.2 FFV-C の指定パラメータ

実行制御パラメータを記述します。

5.2.1 ApplicationControl

FFV-C アプリケーションに関するパラメータを記述します。

```
ApplicationControl {
  CheckParameter      = "off"
  DryRunBC            = "off"
  Operator            = "Kenji_Ono"
  VariableRange       = "off"
}
```

CheckParameter

CheckParameter を on にすると、ソルバー起動後、パラメータファイルを読み込み、初期化段階まで実行した後、停止します。入力パラメータの妥当性をチェックするために使用します。このとき、初期設定パラメータの内容が condition.txt に書き出されます。また、初期条件を与えたフィールドファイルが出力されるので、初期条件のチェックが可能です。

DryRunBC

DryRunBC は境界条件確認のためのオプションです。CheckParameter と異なり、指定時刻まで計算処理を行います。ただし、フィールドの値は計算せず、境界条件のみ更新します。このオプションは時間的に変化する境界条件が正しく動作しているかをチェックします。

Operator

Operator には、FFV-C の実行者名を入力します。FFV-C 実行後に出力される実行時性能測定結果ファイル (profiling.txt) に書き出されます。

VariableRange

VariableRange はデバッグ用のオプションで、値のオーバーシュートやアンダーシュートを矯正します。温度計算を実施する場合に変数値を無次元値で [0,1] の範囲に制限することを指定します。保存則を満たさなくなるため、影響を考慮して利用してください。

表 5.1 ApplicationControl の指定パラメータ

ラベル	値	コメント	必須
CheckParameter	on off	指定パラメータのチェック	
DryRunBC	on off	境界条件動作の確認	
Operator	作業者名	レポートに表示	○
VariableRange	on off	on のとき値を無次元で [0, 1] に制限	

5.2.2 ConvectionTerm

対流項のスキームに関するパラメータを指定します。

```
ConvectionTerm {
  Scheme = "O3MUSCL"
  Limiter = "minmod"
}
```

Scheme と Limiter のパラメータを表 5.2 に示します。Limiter は制限関数の種類を示し、Scheme=“O3MUSCL” の場合にのみ有効となります。非圧縮流れのように物理量の変化が連続的な場合には不要の場合もあります。

対流項スキームによってステンシルが変化するので、ガイドセルの値も異なります。ファイル出力時のオプションでガイドセル出力 GuideOut の指定に注意してください。

表 5.2 Scheme と Limiter のパラメータ

ラベル	指定スキーム	出力ガイドセルサイズ	ラベル	制限関数
O1Upwind	一次精度風上スキーム	1	Minmod	minmod 型
O2Central	二次精度中心スキーム	1	NoLimiter	—
O3MUSCL	三次精度 MUSCL スキーム	2		

5.2.3 DomainInfo

対象となる計算領域の情報を与えます。

```
DomainInfo {  
    UnitOfLength    = "NonDimensional"  
    GlobalOrigin    = (-0.5, -0.5, -0.5 )  
    GlobalRegion    = (1.0,  1.0,  1.0 )  
    GlobalVoxel     = (128   , 128   , 128   )  
  
    //GlobalPitch    = (1.5625e-02, 1.5625e-02, 1.5625e-02)  
    //GlobalDivision = (1     , 1     , 1     )  
}
```

計算領域情報については、計算領域に関するパラメータを指定します。領域パラメータの検討は、4章の解析モデルの作成で述べたように FXgen を用いて計算領域の検討を行います。また、検討結果は TextParser 形式で出力することができ、DomainInfo 情報が得られます。

UnitOfLength

DomainInfo ノードに記述された長さの単位を指定します。指定できる単位は、NonDimensional または M です。

GlobalDivision

並列計算時の各軸方向の分割数を指定します。

GlobalOrigin

計算空間における座標値の最小値を指定します。必須項目です。

GlobalPitch

各軸方向の分割幅を指定します。GlobalVoxel と排他的です。

GlobalRegion

計算領域の大きさを指定します。必須項目です。

GlobalVoxel

計算空間の各軸方向の分割数を指定します。GlobalPitch と排他的で、同時指定時には優先されます。

ActiveSubDomainFile

サブドメインの活性・不活性を指定するファイル名を指定します。ファイルがなければブランクとします。

5.2.4 FillHint

並列自動格子生成を実行するにあたり、媒質同定のフィル操作のためにヒントを与えます。与えることのできるヒントの種類は、外部境界側の媒質を指定する `OuterFace` と計算領域内部の点を指定する `Point` の2種類があります。このうち、外部境界側の媒質指定を少なくとも一つ指定してください*¹。

```
FillHint {  
  outer {  
    kind      = "outerface"  
    direction = "Xplus"  
    medium    = "air"  
  }  
  body {  
    kind      = "point"  
    coordinate = (0.029, 0.04, 0.05)  
    medium    = "tissue"  
  }  
}
```

上記の例では、外部境界の `X.Plus` 側の面を媒質“air”でフィルを開始することを指定しています。同時に、座標点 (0.029, 0.04, 0.05) を含むセルを“tissue”媒質でフィルすることを指定しています。`Point` によるフィルは、自動格子生成でフィルされない内部領域が生じ、その点の媒質が指定可能な場合に利用し、通常は指定しません。

`direction` には、`Xminus`, `Xplus`, `Yminus`, `Yplus`, `Zminus`, `Zplus` のキーワードを指定します。`medium` には、`MediumTable` に記述するキーワードを指定します。

*¹ 将来的には、外部境界条件から自動的に判断し、指定を不要にします。

5.2.5 GeometryModel

計算対象となる形状，あるいはケースを指定します。幾何形状 (STL) ファイルを指定して計算をする場合には，/BcTable/Boundary で必要な STL ファイルを指定します。組み込み例題を指定する場合には，組み込み例題クラスのキーワードを指定します。

```
GeometryModel {
  Source      = "polygon"
  VoxelOutput = "bvx"
  Output      = "off"
  OutputGlyph = "off"
}
```

Source

上記の例では，ポリゴンモデルを用いた計算を `Source = "polygon"` で指定しています。表 5.3 に FFV-C が提供する組み込み例題の一覧を示します。組み込み例題で例題固有のパラメータ設定については，`IntrinsicExample` をご覧ください。

また，具体的な例題の事例については，計算工学ナビの解析事例データベース <http://www.cenav.org> をご覧ください。

表 5.3 example ディレクトリに同梱している例題

ラベル	例題
2Dcavity	二次元キャビティ
2Dcyl	二次元の柱状体まわりの流れ（角柱，円柱，2 物体）
BackStep	バックステップ形状
Cylinder	三次元の柱状体まわりの流れ（角柱，円柱，2 物体）
Duct	直方体と円形断面のダクト
HeatedPlate	加熱平板
Jet	噴流
LDC	辺長が 1 : 1 : 2 のキャビティフロー（実験値との比較）
PerformanceTest	性能評価
Rectangular	矩形計算領域の問題
Sphere	球まわりの流れ
ThermalCavity	加熱冷却によるキャビティの自然対流

VoxelOutput

FFV-C の自動格子生成プロセスの結果を出力します。FXgen のボクセル ID の表示機能で確認します。指定可能なパラメータは次のようになります。

表 5.4 VoxelOutput のキーワード

キーワード	指定内容
svx	VXgen で読み込み可能なボクセルファイル
bvx	FXgen で読み込み可能な圧縮ボクセルファイル
off	出力しない

svx ファイルは 1 セルあたり 4 バイトを必要とするので，大規模ボクセルの場合には出力ファイルが大きくな

り VXgen では可視化することができません。

bvx ファイルはファイル圧縮機能を用いているため、大規模ボクセルも確認できます。出力ファイル名は Output の /Output/Data/DirectoryPath で指定するディレクトリ配下の CID ディレクトリに出力されます。ファイル名は固定で cid_xxxxxx.bvx となり、x には 6 桁のランク番号となります。並列計算時にサブドメイン内が全て同じ ID の場合にはファイルは出力されません。代わりに、proc.dfi ファイル内で /Process/Rank[@]/CellID=2 のように、ID が直接記述されます。

bvx ファイルを指定した場合、同時に境界条件を確認するためのファイル群が BCflag ディレクトリに出力されます。このファイルは、FXgen でセルの 6 面に与えられた境界条件 ID を確認するために利用します。ファイル名は固定で bcf_xxxxxx.bvx となり、x には 6 桁のランク番号となります。

Output

Output は、FFV-C で読み込んだ形状データを出力します。形状を確認する場合に”on”にします。出力ファイル形式は STL Binary 形式です。

1 プロセスでの計算 (Serial, OpenMP) の場合には各グループ 1 ファイル、並列プロセスで計算時にはポリゴンを保持しているサブドメインのプロセスだけがファイルを出力します。出力されるファイル名は、Duct_20130915104336.stlb のように、グループ名の後にタイムスタンプが追加されます。

OutputGlyph

自動格子生成時の交点計算の結果を確認するために、物体とセルセンター間を結ぶ線分上の交点位置を示すポリゴンを生成し、出力します。V-Isio や FXgen で形状データや格子データとともに表示し、確認を行います。

5.2.6 GoverningEquation

FFV-C の支配方程式を設定します。ここでは、支配方程式の型の選択、浮力モード、形状近似などのパラメータを指定します。

```
GoverningEquation {
  FlowEquation      = "Incompressible"
  HeatEquation      = "FlowOnly"
  Buoyancy          = "NoBuoyancy"
  TimeVariation     = "Unsteady"
  PDEType           = "NavierStokes"
}
```

FlowEquation

FlowEquation には表 5.5 に示す支配方程式の形式を示します。

表 5.5 FlowEquation のパラメータ指定

ラベル	支配方程式
Incompressible	非圧縮性
LimitedCompressibility	微小圧縮性

HeatEquation

HeatEquation には、表 5.6 に計算する問題の熱流動現象の分類（熱流動タイプ）を示します。熱伝導方程式を指定している場合（“SolidConduction”）には、FFV-C の実行初期化時に生成される condition.txt ファイル内で Heat Conduction Equation と表示されます。

Buoyancy

Buoyancy の指定は、HeatEquation が必要とする場合にのみ有効になります。

表 5.6 熱対流計算と HeatEquation および Buoyancy の関係

支配方程式	HeatEquation	Buoyancy
純強制対流	FlowOnly	—
強制熱対流（浮力なし）	ThermalFlow	NoBuoyancy
強制熱対流（浮力あり）	ThermalFlow	Boussinesq
自然対流	ThermalFlowNatural	Boussinesq
固体熱伝導	SolidConduction	—

TimeVariation

TimeVariation ラベルでは、解析する現象として定常 Steady あるいは非定常 Unsteady を指定します。

PDEtype

PDEtype で指定する方程式の型は、NavierStokes または Euler から選択します。デフォルトは NavierStokes で、Euler はテスト用のパラメータです。

5.2.7 IntrinsicExample

組み込み例題に固有のパラメータを指定します。

```
IntrinsicExample {
  FluidMedium = "air"
  SolidMedium = "fe"
}
```

指定可能なパラメータを表 5.7 に示します。指定できるパラメータは各組み込み例題毎に異なります。単位は /Unit/UnitOfInputParameter の指定に従います。

表 5.7 IntrinsicExample で指定できるパラメータ

組み込み例題	指定可能なラベル	DataType	指定値
全例題共通	FluidMedium	STRING	MediumTable 内のノード名
	SolidMedium	STRING	MediumTable 内のノード名
Cylinder, Duct, Sphere, Step	Driver	REAL	ドライバ部分の長さ [m] 値が 0 ならば、ドライバ部なしを指定
	DriverMedium	STRING	MediumTable 内のノード名
	DriverFaceMedium	STRING	MediumTable 内のノード名
Cylinder	Dimension	STRING	2D 3D
	Width	REAL	幅 [m]
	Height	REAL	高さ [m]
Duct	Shape	STRING	Circular Rectangular
	Diameter	REAL	断面径 [m]
	Direction	STRING	Xminus Xplus Yminus Yplus Zminus Zplus
Jet	Dimension	STRING	2D 3D
	Ring1/UseRing	STRING	Yes No
	Ring1/InnerRadius	REAL	Ring1 の内径 [m]
	Ring1/OuterRadius	REAL	Ring1 の外径 [m]
	Ring1/RotationFrequency	REAL	回転周波数 [Hz]
	Ring1/InletMassFlow	REAL	噴出流量 [m^3]
	Ring2/UseRing	STRING	Yes No
	Ring2/InnerRadius	REAL	Ring2 の内径 [m]
	Ring2/OuterRadius	REAL	Ring2 の外径 [m]
	Ring2/RotationFrequency	REAL	回転周波数 [Hz]
Rectangular	CheckEven	STRING	Yes No 分割数の偶数チェック
	Dimension	STRING	2D 3D
Sphere	Radius	REAL	球の半径 [m]
Step	Dimension	STRING	2D 3D
	StepLength	REAL	ステップの長さ [m]
	StepHeight	REAL	ステップの高さ [m]

5.2.8 Iteration

圧力のポアソン方程式や陰解法のように、得られる線形システムの係数行列が大型疎行列となる場合には反復解法を用います。ここでは利用できる反復法とそのパラメータを指定します。FFV-C では、圧力 Poisson 反復、速度の陰解法時の反復、温度陰解法について、反復法のパラメータを指定できます。

FFV-C の反復過程は、フラクショナルステップ法を基本としていますが、ダルシー則のような速度の関数で圧力勾配が決まるような境界条件を陰的に扱えるように2段階の反復を用いています。このため、HSMAC のように圧力のアップデートと同時に速度 \mathbf{u}^{n+1} の値を毎反復ごとにアップデートし、 $\nabla \cdot \mathbf{u}^{n+1}$ を評価し、速度の発散が指定値以下になったら収束したと判断しています。まず、圧力の Poisson 方程式の反復段階では、指定残差まで収束させます。その後、2段階めの反復では速度と圧力を同時に緩和する反復を行います。つまり、圧力 Poisson の収束と速度の発散値の収束の2つの閾値で収束判定を行っています。

```
Iteration {
  LinearSolver[@] {
    Alias          = "sor2"
    class          = "sor2sma"
    MaxIteration    = 20
    ResidualCriterion = 1.0e-4
    ResidualNorm    = "RbyB"
    ErrorNorm       = "DeltaXbyX"
    Omega          = 1.1
    CommMode        = "async"
  }

  LinearSolver[@] {
    Alias          = "sor"
    class          = "sor"
    MaxIteration    = 50
    ResidualCriterion = 1.0e-4
    ResidualNorm    = "RbyX"
    ErrorNorm       = "DeltaXbyX"
    Omega          = 1.1
  }

  LinearSolver[@] {
    Alias          = "bicg"
    class          = "bicgstab"
    MaxIteration    = 100
    ResidualCriterion = 1.0e-6
    ResidualNorm    = "RbyB"
    ErrorNorm       = "DeltaXbyX"
    Preconditioner  = "SOR2SMA" // "none" or "SOR"
    InnerIteration  = 10
    Omega          = 1.1
    CommMode        = "async"
  }

  DivMaxIteration  = 50
  DivCriterion     = 1.0e-4
  DivNorm          = "max" // "L2"

  Pressure         = "bicg"
  Velocity         = ""
  Temperature      = "sor2"
}
```

反復パラメータの指定は、2段階になっています。まず、LinearSolver[@] ノードで反復解法と収束判定基準をリストアップし、alias で任意の識別名称を与えます。次に、圧力反復 (Pressure)、速度の陰解法 (Velocity)、温度 (Temperature) の各ノードで各反復の線形ソルバーの識別名称 (Alias) を指定します。

以下に LinearSolver[@] で必須な項目を示します。ここに示される項目以外は、各反復解法固有のパラメータとなります。

Alias

収束判定基準の名称です。

Class

反復解法を指定します。

表 5.8 反復解法の指定

LinearSolver	反復法の指定
SOR	PointSOR 法（ベクトル処理不可）
SOR2SMA	ストライドメモリアクセス型の 2 色 SOR 法（ベクトル処理可）
BiCGSTAB	BiCGStab 法

MaxIteration

反復の最大回数を指定します。

ResidualCriterion

収束判定の閾値を指定します。

ResidualNorm

残差の収束判定のノルムを指定します。

表 5.9 指定可能な ResidualNorm

ラベル	指定内容
RbyB	残差を定数項ベクトルの絶対値で割った値 $\sqrt{\sum_{i,j,k}^{Max} \ \vec{r}\ _2} / \sqrt{\sum_{i,j,k}^{Max} \ \vec{b}^0\ _2}$, $\vec{r} = \vec{b} - A\vec{x}$
RbyR0	残差を初期残差で割った値 $\sqrt{\sum_{i,j,k}^{Max} \ \vec{r}\ _2} / \sqrt{\sum_{i,j,k}^{Max} \ \vec{r}^0\ _2}$

ErrorNorm

誤差のノルムを指定します。

表 5.10 指定可能な ErrorNorm

ラベル	指定内容	
DeltaX	解ベクトルの変化量の L2 ノルム	$\sqrt{\sum_{i,j,k}^{Max} \ \vec{x}^{(m+1)} - \vec{x}^{(m)}\ _2}$
DeltaXbyX	解ベクトルの変化量の L2 ノルムを 解ベクトルの自体の L2 ノルムで割った値	$\sqrt{\sum_{i,j,k}^{Max} \ \vec{x}^{(m+1)} - \vec{x}^{(m)}\ _2} / \sqrt{\sum_{i,j,k}^{Max} \ \vec{x}\ _2}$

表 5.11 は、各反復法に固有の指定パラメータを示します。

表 5.11 固有パラメータの指定

ラベル	指定内容	SOR	SOR2SMA	BiCGSTAB
Omega	加速（緩和）係数	○	○	○ *1
CommMode	袖通信のモード ”Sync ”Async”	—	○	○ *2
Preconditioner	前処理法	—	—	○ *3
InnerIteration	前処理の内部反復回数	—	—	○

- *1) Omega の指定は、前処理で SOR, SOR2SMA を指定したときのみ有効です。
 *2) 前処理法の指定で SOR2SMA を指定したときのみ有効です。
 *3) 前処理法として、SOR, SOR2SMA を指定できます。

次に、速度と圧力の同時反復の収束パラメータは、以下の 3 つを指定します。

DivMaxIteration

圧力速度反復の最大反復回数を指定します。

DivCriterion

非圧縮条件 $\nabla \cdot \vec{v}$ の収束判定閾値を指定します。

DivNorm

ノルムを指定します。 ”max” または ”L2” を指定します。

5.2.9 MediumTable

FFV-C の計算で利用する媒質の物性値テーブルを記述します。ここで記述する媒質は、解析に利用する媒質名をガイドセル部分に与える媒質も含めて、過不足なく挙げる必要があります。

```
MediumTable {
  air {
    State           = "Fluid"
    MassDensity     = 1.1763
    SpecificHeat    = 1007
    ThermalConductivity = 2.614e-02
    KinematicViscosity = 15.83e-06
    Viscosity       = 18.62e-06
    SpeedOfSound    = 340.0
    VolumeExpansion = 0.04e-3
    COLOR           = "23B7A9FF"
  }
  Fe {
    State           = "Solid"
    MassDensity     = 7870.0
    SpecificHeat    = 442.0
    ThermalConductivity = 80.3
    COLOR           = "9C4625FF"
  }
}
```

各媒質は固体と流体によって記述しなければならない物性値が異なります。指定できる項目を表 5.12 に示します。固体については、密度・比熱・熱伝導率のみの記述となります。各媒質の情報は、ノード名によって管理されます*2。

色番号は FXgen が出力するカラーテーブル値で、任意です。

表 5.12 MediumTable における流体要素と固体要素の物性値の指定

ラベル	説明	単位		ラベル	説明	単位
State	セル状態の識別	Fluid		State	セル状態の識別	Solid
MassDensity	密度	kg/m^3		MassDensity	密度	kg/m^3
SpecificHeat	定圧比熱	$kJ/(kgK)$		SpecificHeat	定圧比熱	$kJ/(kgK)$
ThermalConductivity	熱伝導率	$W/(mK)$		ThermalConductivity	熱伝導率	$W/(mK)$
KinematicViscosity	動粘性係数	m^2/s		Color	色番号（任意）	
Viscosity	粘性係数	$Pa \cdot s$				
SpeedOfSound	音速	m/s				
VolumeExpansion	体膨張率	$1/K$				
Color	色番号（任意）	—				

*2 FFV-C の内部データ構造の面からは、ノード名が示す物性値は配列に格納されます。その配列の格納番号とノード名がリンクしており、格納番号は ID と呼ぶこともあります。

5.2.10 MonitorList

ユーザが指定した物理量を指定した位置でサンプリングし、ファイルに出力する機能です。サンプリングして出力する機能は2通りの方法で実装されています。ここでは、指定した座標点で計算結果をサンプリングし、ファイルに出力する方法について説明します。詳細は、第7章をご覧ください。もう一つの指定方法は、ポリゴンに与えられたノード名を用いて CellMonitor 機能を使う方法で、これについては境界条件ノードをご覧ください。

```
MonitorList {
  Log           = "On"
  OutputMode    = "Gather"
  Sampling {
    TemporalType = "time"
    Interval     = 2.0e-5
  }

  list[@] {
    type          = "Line"
    label         = "line1"
    Variable      = "Velocity"
    SamplingMethod = "Interpolation"
    SamplingMode   = "Fluid"
    Division      = 64
    From          = (-0.5, 0.0, 0.0)
    To            = (0.5, 0.0, 0.0)
    Variables {
      velocity    = "on"
      pressure    = "off"
      temperature = "off"
      TotalPressure = "off"
    }
  }

  list[@] {
    type          = "PointSet"
    label         = "line2"
    SamplingMethod = "Interpolation"
    SamplingMode   = "Fluid"
    Division      = 64
    From          = (0.0, 0.0, -0.5)
    To            = (0.0, 0.0, 0.5)
    Variables {
      velocity    = "on"
      pressure    = "off"
      temperature = "off"
      TotalPressure = "off"
    }
  }

  list[@] {
    type          = "Polygon"
    label         = "sensor"
    SamplingMethod = "Interpolation"
    SamplingMode   = "Fluid"
    Variables {
      velocity    = "on"
      pressure    = "off"
      temperature = "off"
      TotalPressure = "off"
    }
  }
}
```

指定パラメータを表 5.14 に示します。MonitorList では、表 5.13 に示す数種類の指定方法があります。点群 (PointSet)、線分 (Line) の2種類は、指定される点毎にサンプリングを行い、サンプリング結果を記録します。一

方, Box, Cylinder, Polygon は指定される領域をサンプリングし, その平均値を記録します. また, Box, Cylinder の領域は座標値とベクトルパラメータで指定します.

表 5.13 モニタ部の指定方法

指定ラベル	指定内容
Box	矩形領域を指定
Cylinder	円筒形領域を指定
Polygon	ポリゴンによる指定
Line	線分による指定
PointSet	点群による指定

表 5.14 モニタリストでの指定パラメータ

ラベル	指定ラベル	指定内容
Log	on off	ログ出力指定
OutputMode	Gather	マスタープロセスに集約して出力
	Distribute	各プロセス毎に出力
Sampling/TemporalType	Step Time	出力形式の指定
Sampling/Interval	—	指定間隔
CellMonitor	on off	モニタ出力指定
PointSet		点群によりモニタ点を指定する
Set		点の座標を指定する
x,y,z		座標
Line		線分によりモニタ点を指定する
From		開始点座標を指定する
To		終了点座標を指定する
Division		開始点と終了点を結ぶ線分の分割数
Variable	Velocity	速度を指定
	Pressure	圧力
	Temperature	温度
	TotalPressure	全圧
	Vorticity	渦度
SamplingMethod	Nearest	モニタ指定点を含むセルの値
	Interpolation	三重線形内挿補間
	Smoothing	局所平均による平滑化
SamplingMode	All	全セルを対象とする
	Fluid	流体セルのみを対象とする
	Solid	固体セルのみを対象とする

- モニタ出力機能は, Log ラベルで on/off を指定します.
- 出力モードは OutputMode ラベルで指定します. これは並列計算時のファイル出力方式で, マスターノードに集約してファイル出力する場合には Gather を指定し, 分散ノード毎にファイル出力する場合には Distribute を指定します.
- Variable ラベルでは, サンプリングする物理量を指定します. 物理量は複数行指定可能です.
- SamplingMethod ラベルで指定されるパラメータは, サンプリング方法を指定します.
- SamplingMode で指定されるパラメータは, サンプリングモードを指定します.
- ファイル出力間隔は, Sampling/Interval で指定し, その指定単位を Sampling/TemporalType で指定します.

5.2.11 Output

各種履歴ファイルとフィールドデータ（瞬間値，平均値，派生変数）出力の制御パラメータを指定します。

```
Output {
  Log {
    Base           = "On"
    Profiling      = "On"
    WallInfo       = "Off"
    Console {
      TemporalType = "Step"
      Interval     = 1
    }
    History {
      TemporalType = "Step"
      Interval     = 1
    }
  }

  Data {
    Format          = "sph"
    DirectoryPath   = "hoge"
    TimeSlice       = "off"

    BasicVariables {
      TemporalType  = "step"
      Interval      = 100

      TotalPressure = "Off"
      Helicity      = "Off"
      Vorticity     = "Off"
      Qcriterion    = "Off"
      Divergence    = "Off"
    }

    StatisticalVariables {
      TemporalType  = "time"
      Interval      = 1.0

      VelocityStat  = "On"
      PressureStat  = "Off"
      TemperatureStat = "Off"
    }
  }

  FormatOption {
    SPH {
      GuideOut      = 2
    }

    PLOT3D {
      XYZfile       = "on"
      IblankFile    = "on"
    }
  }
}
```

Log

Log ノードは基本履歴ファイルの on/off を制御し，標準モニタ出力やコンポーネント情報，領域の流量収支履歴の出力を制御します。

■Base

標準履歴ファイルの出力を制御します。標準履歴ファイルは、”history_base.txt”, ”history_compo.txt”, ”history_domainflux.txt”, ”history_force.txt”の 4 つのファイルです。

■Profiling

Profiling は実行時に性能測定のための計時を行い、結果をレポートとして出力することを指定します。Detail を指定することにより詳細なレポートを出力します。出力項目の詳細は性能情報をご覧ください。

■WallInfo

WallInfo は壁法則を用いた場合の種々の情報を出力しますが、試験的なものです。

■Console, History

Console と History ノードでは、それぞれ標準出力（モニタ画面出力）と履歴ファイルの出力タイミングを指定します。TemporalType により、時刻単位 (step|time) を、Interval でファイル出力間隔を指定します。つまり、Console/TemporalType あるいは History/TemporalType=“Time” の場合、時刻の単位は UnitOfInputParameter で指定したモードに従います。

表 5.15 履歴ファイルの出力指定

ラベル	指定内容	値
Base	標準履歴ファイル	on off
Profiling	実行性能レポートの作成・出力	on off detail
WallInfo	壁面情報履歴	on off
CCNVfile	CCNV 機能の利用	on off
Console/TemporalType	標準出力の出力指定形式	Step (ステップ数指定) Time (時刻指定)
Console/Interval	出力間隔	ステップ数 時間
History/TemporalType	履歴ファイルの出力指定形式	Step (ステップ数指定) Time (時刻指定)
History/Interval	出力間隔	ステップ数 時間

Data

Data ノードでは基本変数 (BasicVariables)、統計値 (StatisticalVariables) のフィールドデータのファイル出力制御を記述します。

■Format

出力するファイルフォーマット形式を指定します。指定可能なファイルフォーマットを表 5.16 に示します。各ファイルフォーマットの詳細に関しては、後述の FormatOption で記述します。

表 5.16 FFV-C でサポートするファイルフォーマット

フォーマット	コメント
sph	VCAD SPH フォーマット
plot3d	NASA 開発の構造格子用のフォーマット
bov	VisIt で利用される raw データに近いフォーマット (スカラー形式のデータのみ)

■DirectoryPath

ファイル出力するディレクトリを指定します。相対パスと絶対パスの両方での指定ができます。指定がない場合には、ジョブを起動したカレントディレクトリにファイルが出力されます。

■TimeSlice

並列時に出力タイムスライス毎にディレクトリを作成します (on | off)。

■BasicVariables

圧力、速度、温度など原始変数と静圧などの派生変数の出力を指定します。Intervalによりファイル出力間隔を指定します。指定する単位は、TemporalTypeによって、時刻またはステップの時制単位を指定します。TemporalType="time"の場合、時刻の単位はUnit/UnitofInputParameterで指定したモードに従います。

下記は派生変数（基本変数から計算される変数）の生成を指定します。表 5.17 に示す各変数は、on/offのスイッチ指定により有効・無効になり、指定するタイミングでファイルに出力されます。

表 5.17 派生変数の指定

ラベル	生成する派生変数
Divergence	発散量
Helicity	ヘリシティ
Qcriterion	速度勾配テンソルの第二不変量
TotalPressure	全圧
Vorticity	渦度ベクトル

TotalPressure

全圧の計算を指定した場合には、tp*.sph のファイル名でファイルが出力されます*3。
全圧は次式で定義され、単位体積あたりのエネルギーを表します。

$$\frac{1}{2}u'^2 + \frac{P'}{\rho'} \quad [Pa] \sim [J/m^3] \quad (5.1)$$

非圧縮の場合には、

$$P_T' = \frac{1}{2}\rho'u'^2 + P' \quad [J/m^3] \quad (5.2)$$

式 (5.2) は無次元化すると、以下のようになります。

$$P_T = \frac{P_T'}{\rho'_0 u_0'^2} \quad (5.3)$$

Vorticity

渦度の計算を指定した場合には、vrt*.sph のファイル名でファイルが出力されます。

*3 ワイルドカード*には、ステップ数や並列計算時にはランク番号が入ります

Helicity

ヘリシティの計算を指定した場合には、`hty*.sph` のファイル名でファイルが出力されます。ヘリシティは速度ベクトル \vec{u} と渦度ベクトル $\vec{\omega}$ の内積として定義される量で次式により表されます。`hlt*.sph` のファイル名でファイルが出力されます。

$$H = \vec{u} \cdot \vec{\omega} \quad (5.4)$$

Qcriterion

速度勾配テンソルの第2不変量は渦構造を可視化するのに利用され、符号により単純剪断乱流の中の層状渦と管状渦を区別することができます [6]。 `qcr*.sph` のファイル名でファイルが出力されます。

Divergence

デバッグのため、速度の発散値を `div*.sph` のファイル名で出力します。

■StatisticalVariables

統計値の出力フォーマットと出力タイミングを指定します。TemporalType により、時刻単位 (step | time) を、Interval でファイル出力間隔を指定します。Reynolds Stress については、レイノルズ応力テンソル (瞬時値, 時間平均値), 生成項 (時間平均値), 散逸項 (時間平均値), 乱流拡散項 (時間平均値), 速度圧力勾配相関項 (時間平均値) を計算します。

表 5.18 SPH ファイルフォーマットの詳細オプション

ラベル	説明
VelocityStat	速度の統計量計算の指定 { on off }
PressureStat	圧力の統計量計算の指定 { on off }
TemperatureStat	温度の統計量計算の指定 { on off }
ReynoldsStress	乱流統計量計算の指定 { on off }

統計操作の開始時刻については、TimeControl で指定します。

FormatOption

各ファイルフォーマットの詳細項目を指定します。現在は、SPH 形式、PLOT3D 形式をサポートしています。

■SPH

SPH ファイル形式は、直交等間隔データを表現するファイル形式で、理研 VCAD プロジェクトで開発されました。出力ガイドセルのサイズは ConvectionTerm の項を参照して GuideOut で指定してください。

■PLOT3D

PLOT3D フォーマットで、結果を出力する場合のオプションを記述します。

Iblankfile は、計算領域中流体セルでない領域を非計算グリッドとして出力します。XYZfile ラベルは、計算領域の形状データファイルの出力の有無を指定します。

表 5.19 PLOT3D 形式ファイルの出力指定

ラベル	指定内容	コメント
XYZfile	形状ファイル出力	on off
Iblankfile	マスクファイル出力	on off

5.2.12 Reference

解析に用いる無次元化の基準量，あるいは無次元パラメータを指定します．

```
Reference {
  Length      = 0.046
  Velocity     = 50.14
  BasePressure = 0.0
  Temperature {
    Base       = 20.0
    Difference  = 35.0
  }
  Medium      = "air"
}
```

表 5.20 に示すように基準量を必要に応じて記述します．無次元パラメータである Reynolds 数と Prandtl 数は，/Unit/UnitOfInputParameter の指定が無次元のときのみ指定できます．Medium で指定する名前は，基礎方程式の無次元化に用いる物性値（密度や比熱など）を指定するためのもので，MediumTable 内にリストアップされている必要があります．固体熱伝導解析の場合には固体のラベルを指定し，それ以外の（熱）流動解析の場合には流体のラベルを指定します．

表 5.20 無次元化の基準パラメータ

ラベル	意味	単位
Length	代表長さ	<i>m</i>
Velocity	代表速度	<i>m/s</i>
BasePressure	基準圧力	<i>Pa</i>
Temperature/Base	基準温度	[K °C]
Temperature/Difference	温度差	[K °C]
Prandtl	プラントル数	— 無次元のときのみ指定
Reynolds	レイノルズ数	— 無次元のときのみ指定
Medium	代表物性値として指定する媒質ラベル	—

Temperature/Base, Difference では，温度計算を実施する場合の基準量を有次元値で指定します．基準温度 (Base) と温度差 (Difference) は，非圧縮計算のパスシブスカラーによる温度計算では温度場を特徴づける代表量となります．単位は/Unit/Temperature で指定します．

5.2.13 ReferenceFrame

観測の座標系を指定します.

```
ReferenceFrame {  
    Mode = "Stationary"  
}
```

FFV-C では, 表 5.21 に示す選択肢があります. 移動座標系を指定する場合には, 格子の移動速度の各方向成分 (有次元では $[m/s]$) を入力します. 座標系は右手系をとり, 各軸 x, y, z 方向の速度成分をそれぞれ u, v, w とします. 静止座標系と移動座標系とでは, 同じ問題を解く場合でも与える境界条件が異なるので注意します.

表 5.21 Reference_Frame の指定

ラベル	指定パラメータ	参照座標系
Stationary	—	静止座標系
Translational	u, v, w	並進運動する移動座標系

5.2.14 ShapeApproximation

このパラメータは形状近似レベルを指定します。

```
ShapeApproximation {  
  Method = "Binary"  
}
```

ShapeApproximation ラベルには、表 5.22 に示す解析モデルの形状近似モードを指定します。

表 5.22 形状近似モードの指定

ラベル	形状近似
Binary	バイナリボクセル近似
DistanceInfo	距離情報近似

5.2.15 SolvingMethod

時間積分と解法アルゴリズムの組み合わせを指定するパラメータです。

```
SolvingMethod {  
  Flow = "FS_C_EE_D_EE"  
  Heat = "C_EE_D_EE"  
}
```

表 5.23 に時間進行法と分離解法の種類の組み合わせを示します。Flow ラベルでは流動の支配方程式の時間積分法と解法アルゴリズムの組み合わせを指定します。

表 5.23 流動解析のアルゴリズム指定

ラベル	時間積分法と解法の組み合わせ
FS_C_EE_D_EE	Fractional Step 法 + 時間一次精度 Euler 陽解法（対流項と拡散項）
FS_C_AB_D_AB	FS 法 + 時間二次精度 Adams-Bashforth 陽解法（対流項と拡散項）

温度解析の場合には、Heat ラベルで温度輸送方程式の時間進行法と解法アルゴリズムの組み合わせを表 5.24 に示します。

表 5.24 温度解析のアルゴリズム指定

ラベル	時間積分法と解法の組み合わせ
C_EE_D_EE	時間一次精度 Euler 陽解法（対流項と拡散項）
C_EE_D_EI	時間一次精度 Euler 陽解法（対流項） + Euler 陰解法（拡散項）

5.2.16 StartCondition

計算のスタート条件を指定します。

```

StartCondition {
  Restart {
    Staging      = "off"

    DFIfiles {
      Velocity    = "vel0.dfi"
      Pressure    = "prs0.dfi"
      Fvelocity   = "fvel.dfi"
    }
  }

  InitialState {
    MassDensity = 1000.0
    Pressure    = 0.0
    Velocity    = (0.0, 0.0, 0.0)

    MediumTemperature {
      Fe    = 150.0
      Air   = 100.0
    }
  }
}

```

表 5.25 StartCondition のパラメータ指定

ノード	ラベル	指定値	コメント
Restart	Staging	on off	リスタート指定
	DFIfiles/Velocity		速度の DFI ファイル名
	DFIfiles/Pressure		圧力の DFI ファイル名
	DFIfiles/Temperature		温度の DFI ファイル名
	DFIfiles/Fvelocity		セルフェイス位置の速度の DFI ファイル名
InitialState			初期条件を指定
	MassDensity		密度
	Pressure		圧力
	Velocity		速度ベクトル
	MediumTemperature		熱計算をする場合に MediumTable でリストアップした媒質に対する初期温度を指定する [°C]

InitialState

物理変数の初期値を指定します。記述する初期値は有次元量で指定しますが、/GoverningEquation/HeatEquation="FlowOnly" と選択した場合のみ、無次元での指定も可能です。圧力値は、/Unit/Pressure で指定する圧力の単位に従います。各変数の無次元化は以下のようになり、添え字の 0 は代表値または基準値を意

味します.

$$\left. \begin{aligned} \rho &= \frac{\rho'}{\rho'_0} \\ p &= \frac{p' - p'_0}{\rho'_0 u'^2_0} \\ u_i &= \frac{u'_i}{u'_0} \\ \theta &= \frac{\theta' - \theta'_0}{\Delta\theta'_0} \end{aligned} \right\} \quad (5.5)$$

InitialState/MediumTemperature では, MediumTable でリストアップした媒質に対して, 初期温度を与えます.

5.2.17 TimeControl

時刻制御に関するパラメータを指定します。

```
TimeControl {
  Acceleration {
    TemporalType      = "Time"
    AcceleratingTime = 0.0
  }

  TimeStep {
    Mode              = "CFLReferenceVelocity"
    DeltaT            = 0.1
  }

  Session {
    TemporalType      = "step"
    Start             = 0
    End               = 100
  }

  Statistic {
    TemporalType      = "step"
    Start             = 0
    End               = 0
  }
}
```

Acceleration

Acceleration ノードは、イニシャルスタートの場合にのみ有効なパラメータで、一定速度になるまでの時間を指定します。TemporalType で時間の単位を指定します。指定単位が Time の場合、加速時間の値は/Unit/UnitOfInputParameter で指定するモード ("Dimensional" | "NonDimensional") に従います。計算初期の急加速による発散を防ぐため、格子の移動速度や指定流速をゼロから徐々に加速し、指定の値に漸近させる目的で利用します。加速時間を長くすると流れの発達に時間がかかるので、発散しない程度の時間を設定します。加速時間 t_0 は全計算領域を通過する時間程度が適切で、 $t_0 = L'/u'_0$ を参考にします。ここで L' は領域長さで u'_0 は代表速度とします。値として 0.0 を指定すると急加速になります。加速時間中は、参照速度 u_{Ref} に対して次式の加速曲線を与え、図 5.1 のように滑らかに一定速度に漸近させます。

$$u_{Ref} = \begin{cases} \frac{1}{2} \left(1 - \cos \left(\frac{t}{t_0} \pi \right) \right) & (t < t_0) \\ 1.0 & (t \geq t_0) \end{cases} \quad (5.6)$$

TimeStep

時間積分幅 Δt を指定します。表 5.26 に時間積分幅 Δt の指定方法を示します。拡散数 D は一次元の拡散方程式の場合 $D = \alpha' \Delta t' / h'^2$ で与えられます。 α は拡散係数で、Navier-Stokes 方程式の場合 $1/Re$ 、温度の輸送方程式の場合には $1/Pe$ となります。安定性解析から $D < 1/2$ であることが要請されます。多次元の場合には、 d_m を次元数として $\Delta t < h'^2 / (2d_m \alpha')$ となります。

DeltaT には CFL 数、または Δt を記述します。時間積分幅の選択は、熱解析ソルバの種類を示す HeatEquation のパラメータと関連があり、SolidConduction の場合には Direct のみ選択できます。

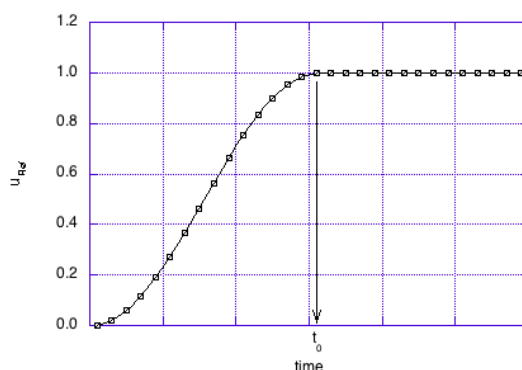


図 5.1 加速中の速度プロファイル

表 5.26 Time_Increment のパラメータ指定

Mode	時間積分幅の決定方法	DeltaT への指定数値
Direct	Δt を直接指定する	Δt
CFLReferenceVelocity	CFL 数を指定し、代表流速から Δt を決定	CFL 数
Diffusion	拡散数から Δt を決定	—
CFLDiffusionReferenceVelocity	代表流速に対する CFL 数と拡散数から Δt を決定	CFL 数

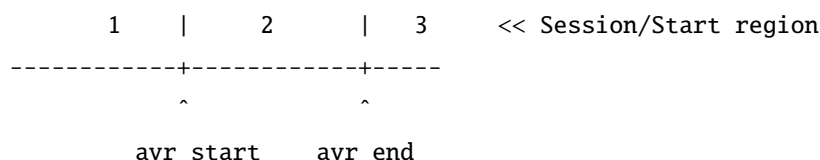
Session

計算ジョブを実行する回の計算時間を指定します。計算時間を time で指定する場合、時間の単位は UnitOfInputParameter のモードに従います。指定された単位で、開始時刻と終了時刻を指定します。Start=0 の場合はイニシャルスタートで、ゼロ以外はリスタートを指定することになります。

Statistic

時間平均操作に関するパラメータを指定します。FFV-C は非定常解析を行いますので、流れの挙動が準定常状態になったところで時間平均操作を開始し、十分な長さで時間平均操作が行われた速度場や温度場を定常解とみなします。時間平均操作の開始時刻は Start で指定し、この開始時刻以降、毎ステップごとに時間平均操作を行います。平均操作の開始時刻は、TemporalType の単位で記述します。Time を指定した場合には UnitOfInputParameter の次元に従います。つまり、UnitOfInputParameter="DIMENSIONAL" の場合には有次元時刻で時間平均操作の開始時刻を指定することになります。

平均値のリスタートでは、Session/Start が示すリスタート時刻と/Average/Start, /Average/End の示す平均区間の時刻により、次のような 1~3 のパターンに分類できます。



1. 平均操作は行わぬが、まだ指定時刻に到達していないので、平均値ファイルは存在せず、平均値のリスタートはない
2. 前セッションから継続して平均操作を行うが、既に平均値ファイルが存在する（はず）ので、平均値のリスタート処理を行う
3. 既に平均値操作の区間は終了しているので、平均操作は行わない

時間平均場の出力タイミングは、/Output/Data/AveragedVariables/Interval で指定します.

5.2.18 TurbulenceModeling

LES(Large-Eddy Simulation) のオプションパラメータを指定します。

```
TurbulenceModeling {
  Model          = "Smagorinsky"
  Cs             = 0.2
  InitialPerturbation = "on"
  VelocityProfile = "NoSlip"
}
```

Model

指定できる LES のモデルを表 5.27 に示します。Smagorinsky モデルの場合には、定数 C_S が必須パラメータです。

表 5.27 LES のモデル指定

ラベル	モデル
No	乱流モデルなし
Smagorinsky	標準スマゴリンスキーモデル (定数 C_S を指定)
CSM	Coherent Structure Model
WALE	WALE モデル

VelocityProfile

表 5.28 に示すパラメータにより、壁面の扱いを指定します*4。

減衰関数として用いる Van Driest 関数の無次元定数 A^+ は約 25 としています。

$$f_s = 1 - \exp\left(-\frac{y^+}{A^+}\right) \quad (5.7)$$

表 5.28 壁面 VelocityProfile の指定

ラベル	パラメータの説明
LawOfWall	壁法則
VanDriest	VanDriest 型の減衰関数
NoSlip	滑りなし壁面条件
Slip	滑り壁条件

InitialPerturbation

初期擾乱を与えるパラメータについては、未定。

*4 実験中

5.2.19 Unit

入力ファイルと出力ファイルで用いる単位を指定します*5.

```
Unit {
  UnitOfInputParameter = "Dimensional"
  UnitOfOutput          = "Dimensional"
  Pressure               = "Gauge"
}
```

各ラベルは、表 5.29 に示す単位の指定に用いられます。有次元のファイル出力時には、圧力単位としてゲージ圧 (Gauge Pressure) と絶対圧力 (Absolute Pressure) が選択できます。

UnitOfOutput では出力する結果ファイル (*.sph) の単位を指定し、有次元か無次元を指定できます。

式 (5.8) に示すゲージ圧を式 (5.9) により無次元化する場合に、基準圧として $p'_0 = 1.01325 \times 10^5$ [Pa] を用い、動圧が $10^0 \sim 10^3$ 程度とすると、 $p \sim O(1)$ 程度となるので、単精度計算では4桁程度有効桁が失われる場合もあります。そのような場合、有次元値のファイル出力単位としてゲージ圧 p'_g を用います (非圧縮流れの場合には圧力差が意味をもつので、ゲージ圧でもかまいません)。ゲージ圧の基準となる大気圧 p'_0 [Pa] は BasePressure で指定します。圧力単位の指定は、履歴ファイルのモニタ値にも適用されます。

$$p'_g = p' - p'_0 \quad (5.8)$$

$$p = \frac{p'_g}{\rho'_0 u'^2_0} \quad (5.9)$$

表 5.29 単位の指定

ラベル	指定値	説明
UnitOfInputParameter	Dimensional NonDimensional	入力パラメータファイルの単位を指定します (*1)
UnitOfOutput	Dimensional NonDimensional	出力ファイルの単位を指定します
Pressure	Gauge Absolute	入力パラメータの単位が有次元のときに有効となります
Temperature	Celsius Kelvin	入力パラメータの単位が有次元のときに有効となります

*5 温度単位は Celsius で固定です。

第 6 章

境界条件

FFV-C で設定できる境界条件の設定について説明します。まず境界条件と媒質を指定するパラメータの構造について述べた後、流れと熱の境界条件について説明します。

6.1 境界条件の概要

6.1.1 外部境界条件と局所境界条件

FFV-C では、境界条件を外部境界条件と局所境界条件の2つに分類して実装しています。外部境界条件は計算領域外部面に指定する境界条件で、局所境界条件は計算領域内部に指定する境界条件です。図 6.1 に示すように、計算領域を構成する6面が外部境界面で、この部分に与える境界条件が外部境界条件です。それ以外の内部領域に作用する境界条件は局所境界条件として扱います。外部境界面には、各面に対して外部境界条件を1種類のみ与えることができます。

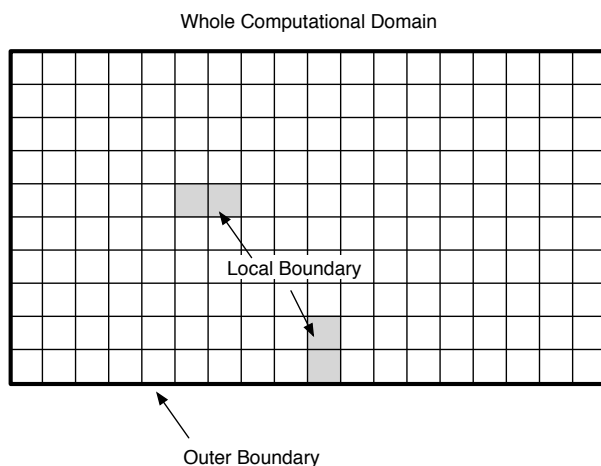


図 6.1 計算領域における外部境界と局所境界の指定場所

6.1.2 計算格子と内部・外部領域

境界条件で参照する計算領域と格子配置について説明します。計算領域とコロケート変数配置の変数のインデックスの表記を図 6.2、図 6.3 に示します。



図 6.2 計算領域のインデックス

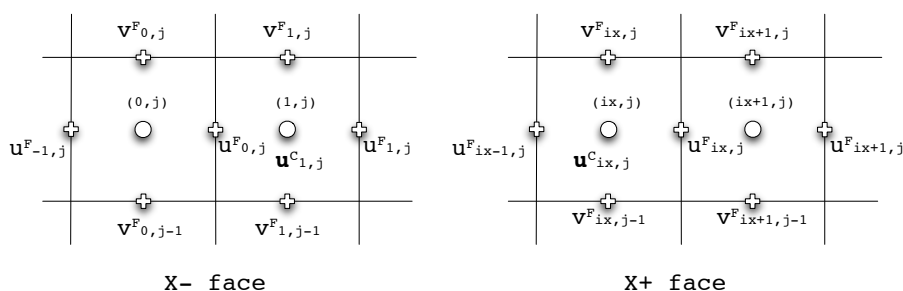


図 6.3 コロケート配置の変数のインデックス。基本変数 (u_i^C, p, θ) は全てセルセンタ位置に配置され、補助的な速度ベクトル u_i^F がスタガード位置に配置されます。

6.1.3 BcTable ノードのパラメータ構造

BcTable ノードでは、次のように Boundary と外部境界条件 OuterBC の2つのノードを記述します。

```

BcTable {
  Boundary {
    throat {
      kind          = "inner"
      class          = "SpecifiedVelocity"
      Medium         = "tissue"
      filepath       = "./throat_m.stl"

      type           = "Massflow"
      profile         = "polynomial6" // constant, harmonic, polynomial6
      //velocity      = 2.3e-04
      OrientationVector = (0.0, 0.0, 1.0)
      InOut           = "out"
      //frequency     = 0.0
      //initialphase   = 0.0
      //constantbias   = 0.0
      //temperature    = 34.0
      c6 = -2.6310
      c5 = 4.8413
      c4 = -3.4224
      c3 = 1.1680
      c2 = -0.2049
      c1 = 0.0179
    }

    cell {
      kind          = "inner"
      class          = "Obstacle"
      Medium         = "tissue"
      filepath       = "./cell_m.stl"
    }

    tfree {
      kind          = "outer"
      class          = "TractionFree"
      Medium         = "air"
    }

    outerWall {
      kind          = "outer"
      class          = "Wall"
      Medium         = "tissue"
      Type           = "fixed"
    }
  }

  OuterBC {
    Xminus = "tfree"
    Xplus  = "tfree"
    Yminus = "tfree"
    Yplus  = "outerWall"
    Zminus = "tfree"
    Zplus  = "tfree"
  }
}

```

6.1.4 境界条件の基本リスト

FFV-C の境界条件の指定方法は、次のようになります。

1. まず、境界条件の基本リストを作成します。計算に用いる境界条件の種類を過不足無くリストアップします。
2. 次に、外部境界条件について、基本リストの中から選択して指定します。

境界条件の基本リストは局所境界条件と外部境界条件について記述しますが、各々で記述する内容は異なります。まず、共通部分について説明し、その後固有部分について説明します。

Boundary

Boundary ノードでは境界条件リストの名前を指定します。境界条件の基本リストにはユニークな名前を命名します。/BcTable/Boundary 配下に境界条件名のノードを作成します。上記の例では、throat, cell, tfree, outerWall などになります。

Kind

局所境界条件か外部境界条件かの種別をしていします。局所境界条件の場合には Inner, 外部境界条件の場合には Outer を指定します。

Class

具体的な境界条件の種類を指定します。局所境界条件と外部境界条件で指定可能なキーワードが異なります。外部境界条件の場合には、表 6.1, 表 6.2 で示すキーワードを与えます。

局所境界条件の場合には、表 6.3 で示すキーワードを与えます。計算領域内に存在する局所的な境界条件を記述するノードです。

FFV-C には、実装上、利用できる媒質数と境界条件数の和に制限があり、指定境界条件数と媒質数の和は 30 個以下となります*1。CellMonitor は境界条件ではありませんが、境界条件と同じ指定方法を用いて実装しているので、このノードで指定します。

表 6.1 外部境界での境界条件の種類 (1)

Class	流れの境界条件	熱流れのときのオプション	熱境界条件
FarField	遠方境界	AmbientTemperature	遠方温度指定 *テスト実装
Intrinsic	組み込み用のカスタム条件	—	カスタム
Outflow	流出境界	—	対流流出
Periodic	周期境界	—	周期境界
SpecifiedVelocity	流入境界	Temperature	流入温度指定
Symmetric	対称境界	—	断熱
TractionFree	接線応力なし	—	流入出と断熱

Medium

この媒質名は、MediumTable ノードにリストアップされた媒質ノード名を参照します。局所境界条件に対しては、指定する形状要素の媒質を指定します。一方、外部境界条件については、ガイドセル部の媒質を指定します。

*1 これらの制限は、境界条件を効率よく実装する方法の制約から来るものです。

表 6.2 外部境界での境界条件の種類 (2)

Class	流れの境界条件	ThermalOption	熱境界条件
Wall	壁面境界	Adiabatic	断熱指定
		ConstantTemperature	一定温度
		HeatFlux	熱流束指定
		HeatTransferS	熱伝達係数と表面温度から熱伝達を計算
		HeatTransferSF	強制対流の層流・乱流熱伝達境界
		HeatTransferSN	自然対流の乱流熱伝達境界
		IsoThermal	等温指定

表 6.3 局所境界条件（コンポーネント）の種類. 表中の Cold flow は流れのみの場合（FlowOnly）を示す.

Class	境界指定位置	適用	実装形式	コンポーネントの説明
CellMonitor	セル要素	Cold/Thermal flow	—	物理量のモニター位置の指定
DirectHeatFlux	セル界面	Thermal flow	熱流束	熱流束指定
HeatSource	セル要素	Thermal flow	外力項	吸発熱指定
HeatTransferS	セル界面	Thermal flow	熱流束	熱伝達係数と表面温度により計算
HeatTransferSF	セル界面	Thermal flow	熱流束	強制対流の層流・乱流熱伝達境界
HeatTransferSN	セル界面	Thermal flow	熱流束	自然対流の乱流熱伝達境界
IsoThermal	セル界面	Thermal flow	熱流束	等温面指定
Monitor	—	Cold/Thermal flow	—	物理量のモニター位置の指定
Obstacle	セル界面	Cold/Thermal flow	熱流束マスク	断熱物体
Outflow	セル界面	Cold/Thermal flow	対流流束	流出境界
Periodic	セル界面	Cold/Thermal flow	参照値指定	部分周期境界
SolidRevolution	—	Cold/Thermal flow	—	回転体, 実験
SpecifiedTemperature	セル要素	Thermal flow	温度指定	温度指定
SpecifiedVelocity	セル界面	Cold/Thermal flow	対流流束	速度指定境界

上記の Kind, Class, Medium が必須の指定項目で, 残りのラベルは各境界条件固有のパラメータとなります.

6.1.5 OuterBC

境界条件の基本リストのグループ名を参照して, 計算領域の外部境界の各面における境界条件を指定します. 前述の例では, Y プラス方向の外部境界面に固定壁面の境界条件を与え, その他の外部境界面にはトラクションフリーの境界条件を設定しています.

6.2 外部境界条件

このセクションでは、外部境界条件の詳細について説明します。

6.2.1 壁面境界

壁面の速度境界条件では、指定する境界面の移動速度を与えます。移動速度は OrientationVector により方向を、Amplitude により大きさを指定します。壁面速度が時間的に変化する場合と一定の場合があります。移動速度ベクトルは壁面と平行なスライド成分のみで、壁面と垂直な成分はゼロである点に注意します。

壁面境界は、与えられた速度からセルフェイス位置の運動量流束を計算して、境界値として与えます。外部境界では下記のような入力パラメータで指定します。次の境界条件の例では、Inlet_1 で Y 方向に一定速度 5[m/s] 平行振動する壁の境界条件を指定し、Inlet_2 で Z 方向に 7[m/s], 2[Hz] で平行振動する壁の境界条件を指定しています。

```
Boundary {
  SlideWall_1 {
    kind          = "outer"
    class         = "Wall"
    medium        = "Fe"
    type          = "slide"
    profile       = "constant"
    velocity      = 5.0
    OrientationVector = (0.0, 1.0, 0.0)
  }

  SlideWall_2 {
    kind          = "outer"
    class         = "Wall"
    medium        = "Fe"
    type          = "slide"
    profile       = "harmonic"
    OrientationVector = (0.0, 0.0, 1.0)
    Amplitude     = 7.0
    frequency     = 2.0
    initialphase  = 0.0
    constantbias  = 0.0
  }
}
```

表 6.4 壁面の速度境界条件の指定パラメータ

ラベル	指定オプション	説明
Type	Fixed Slide	固定壁かスライド壁を指定
Profile	Constant Harmonic polynomial6	指定速度のタイプ
OrientationVector	—	方向ベクトルの成分
Velocity	—	指定単位 [m/s], Profile=Constant の場合のみ
Amplitude	—	速度, 以下のパラメータは Profile=Harmonic の場合のみ
Frequency	—	周波数 f [Hz]
InitialPhase	—	初期位相 ϕ [Rad]
ConstantBias	—	一定値 b [m/s]
ThermalOption	表 6.2 の ThermalOption	Cold flow の場合は不要

壁面の速度境界の指定パラメータを表 6.4 に示します。時間変化を伴う速度指定は Profile="Harmonic" を指定します。harmonic の場合、式 (6.1) の形式の単振動の境界条件を周期や初期位相、固定バイアスと供に与えます。時間的に

変化しない壁面境界の場合には Profile="Constant" と Velocity を指定します。

$$V = A \sin(2\pi ft + \phi) + b \quad (6.1)$$

壁面境界に対する圧力の境界条件は、Navier-Stokes 方程式から Neumann 型の圧力境界条件が得られます。高レイノルズ数流れにおいては、粘性項の寄与が小さいと仮定し粘性項を省略し $\nabla p = 0$ の形式になります。圧力の壁面境界条件については、内部と外部の扱いは同じで、スキーム中で壁面を認識し $\nabla p = 0$ が満たされるようになっていますので、明示的な境界条件の指定は必要ありません。

ThermalOption

計算領域の外部面における壁面に対する熱境界条件としては、断熱、熱流束、熱伝達、等温、固定参照温度が指定できます。熱伝達境界は、さらに幾つかの指定パターンがあります。詳細は Inside_FFVC.pdf をご覧ください。

■断熱境界

熱流束がゼロ、つまり $q' = 0$ を指定します。下記の例では、Insulator と名付けられた外部境界条件に対して、固定壁の断熱条件を指定しています。Insulator の境界面のガイドセルとして Fe の媒質属性もつことを指定しています。

```
Boundary {
  Insulator {
    Kind      = "Outer"
    Class     = "Wall"
    Medium    = "Fe"
    Type      = "fixed"
    ThermalOption = "Adiabatic"
  }
}
```

■熱流束境界

境界面で指定の熱流束 $q'[W/m^2]$ を与えます。符号は計算領域内に流入する熱流束の場合に正、流出する熱流束の場合に負とします。下記の例では heatflux1 に $12.0[W/m^2]$ で流入する熱流束をもつ面を指定しています。

```
Boundary {
  heatflux1 {
    Kind      = "Outer"
    Class     = "Wall"
    Medium    = "Fe"
    Type      = "fixed"
    ThermalOption = "Heatflux"
    Flux      = 12.0
  }
}
```

■熱伝達境界

熱伝達境界は次式の形式で熱流束を与える条件で、幾つかの種類を用意しています。熱流体解析のモードと指定できる熱伝達境界の関係を表 6.5 に示します。

表 6.5 熱伝達境界条件と HeatEquation の関係

HeatEquation	指定できる熱伝達境界の種類
FlowOnly	—
ThermalFlow ThermalFlowNatural	TypeS TypeSN TypeSF
SolidConduction	TypeS

$$q' = -H'(\theta'_{sf} - \theta'_{\infty}) \quad (6.2)$$

H'	$[W / (m^2 K)]$	Coefficient of heat transfer
θ'_{sf}	$[^{\circ}C]$	Surface temperature of solid
θ'_{∞}	$[^{\circ}C]$	Temperature at outer boundary layer

TypeS 表面温度と熱伝達係数により計算

TypeS は表面温度と熱伝達係数を与え、熱流束を計算します。式 (6.2) において、熱伝達係数 H' と固体表面温度 θ'_{sf} を与え、図 6.4 に示す固体表面に隣接する流体セルの値を θ'_{∞} として、界面での熱流束を計算します。

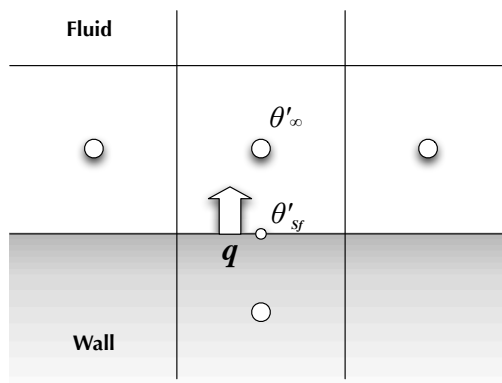


図 6.4 TypeS の熱伝達境界

以下に、熱境界部分のみパラメータ指定の一例を示します。

```
Boundary {
  Type_S {
    Kind          = "Outer"
    Class          = "Wall"
    Medium         = "Fe"
    Type           = "Fixed"
    ThermalOption  = "HeatTransferS"
    SurfaceTemperature = 300.0
    CoefOfHeatTransfer = 20.0
  }
}
```

表 6.6 熱伝達境界 TypeS の指定パラメータ

指定キーワード	パラメータの説明
CoefOfHeatTransfer	熱伝達係数 [$W/(m^2 K)$]
SurfaceTemperature	表面温度 [$^{\circ}C$]

TypeSN 自然対流の乱流熱伝達

自然対流の場合の乱流熱伝達の実験式を実装した境界条件です。文献 [7] には、平板に対する自然対流の層流と乱流の熱伝達に関する近似式が説明されています。雰囲気流体の温度に比べ加熱面の温度が非常に高い場合、平板が長くなると境界層が不安定になり、ほぼ $Ra > 10^9$ で層流から乱流へ遷移します。垂直平板に関する平均熱伝達 ($\overline{Nu_L}$, 代表長 L) は次式で整理されます。

$$\left. \begin{array}{l} \text{層流} \quad \overline{Nu_L} = 0.59 Ra_L^{1/4} \quad (10^4 < Ra_L < 10^9) \\ \text{乱流} \quad \overline{Nu_L} = 0.10 Ra_L^{1/3} \quad (10^9 < Ra_L < 10^{13}) \end{array} \right\} \quad (6.3)$$

一方、水平平板の場合には、加熱面が上面と下面にある場合で雰囲気流体の挙動が異なるため、式 (6.4) のように整理されています。

$$\left. \begin{array}{l} \text{上面加熱} \quad \overline{Nu_L} = 0.54 Ra_L^{1/4} \quad (10^4 < Ra_L < 10^7) \\ \text{上面加熱} \quad \overline{Nu_L} = 0.15 Ra_L^{1/3} \quad (10^7 < Ra_L < 10^{11}) \\ \text{下面加熱} \quad \overline{Nu_L} = 0.27 Ra_L^{1/4} \quad (10^5 < Ra_L < 10^{10}) \end{array} \right\} \quad (6.4)$$

上式を形式的にまとめると、

$$H' = \alpha' Ra_L^{\beta} \frac{\lambda'}{L'} \quad (6.5)$$

TypeSN の境界条件は、上式のパラメータを実装しています。ここでは、垂直平板と水平平板の上面は、同じ係数を用いています。

式 (6.2) は、固体壁面に隣接する流体セルの温度と SurfaceTemperature との温度差を用いて評価します。

```
Boundary {
  Type_SN {
    Kind          = "Outer"
    Class         = "Wall"
    Medium        = "Fe"
    Type          = "Fixed"
    ThermalOption = "HeatTransferSN"
    SurfaceTemperature = 100.0
    VerticalLaminarAlpha = 0.59
    VerticalLaminarBeta  = 0.25
    VerticalTurbulentAlpha = 0.1
    VerticalTurbulentBeta = 0.3333333
    VerticalRaCritical   = 1.0e9
    LowerLaminarAlpha   = 0.27
    LowerLaminarBeta    = 0.25
    LowerTurbulentAlpha = 0.27
    LowerTurbulentBeta  = 0.25
    LowerRaCritical     = 1.0e9
  }
}
```

表 6.7 熱伝達境界 TypeSN のパラメータ

パラメータタグ	記号の意味
VerticalLaminarAlpha	垂直平板と水平平板（上面）の層流時の係数 α
VerticalLaminarBeta	垂直平板と水平平板（上面）の層流時の係数 β
VerticalTurbulentAlpha	垂直平板と水平平板（上面）の乱流時の係数 α
VerticalTurbulentBeta	垂直平板と水平平板（上面）の乱流時の係数 β
VerticalRaCritical	垂直平板と水平平板（上面）の臨界 Ra 数 Ra_L
LowerLaminarAlpha	水平平板（下面）の層流時の係数 α
LowerLaminarBeta	水平平板（下面）の層流時の係数 β
LowerTurbulentAlpha	水平平板（下面）の乱流時の係数 α
LowerTurbulentBeta	水平平板（下面）の乱流時の係数 β
LowerRaCritical	水平平板（下面）の臨界 Ra 数 Ra_L
SurfaceTemperature	表面温度 [°C]

TypeSF 強制対流の層流・乱流熱伝達

強制対流の場合の層流・乱流熱伝達の実験式を実装した境界条件です。文献 [7] から、平板に対する発達した強制対流の乱流熱伝達は、実験による摩擦係数の測定結果とチルトン-コルバーンのアナロジーを用い、温度一定で平板が遷移長さよりも十分に大きいと仮定すると、式 (6.6) のように表せます。実験式を整理すると、熱伝達係数は以下のような表現ができます。

$$\overline{Nu_L} = 0.037 Re_L^{4/5} Pr^{1/3} \quad (6.6)$$

形式的に次式のように表し、パラメータを求めます。

$$H = \alpha Re_L^\beta Pr^\gamma \frac{\lambda}{L'} \quad (6.7)$$

温度差の定義にはバルク温度と隣接セルの値を用いたオプションが選択できます。以下に、パラメータ指定の一例を示します。

式 (6.2) は、固体壁面に隣接する流体セルの温度と SurfaceTemperature との温度差を用いて評価します。

```
Boundary {
  Type_SF {
    Kind          = "Outer"
    Class         = "Wall"
    Medium        = "Fe"
    Type          = "Fixed"
    ThermalOption = "HeatTransferSF"
    SurfaceTemperature = 500.0
    alpha         = 0.037
    beta          = 0.8
    gamma         = 0.333333
  }
}
```

■等温境界

等温壁境界は、指定面で温度が一定となる境界条件で、面温度を一定に保つような熱流束が発生しま

表 6.8 熱伝達境界 TypeSF のパラメータ

タグ	記号の意味
Alpha	式 (6.7) 中の係数 α
Beta	係数 β
Gamma	係数 γ
SurfaceTemperature	表面温度 [$^{\circ}\text{C}$]

す. 例えば X マイナス側の外部境界面のセル界面位置では, 次の形式の熱流束となります.

$$q'_{ISO, 1/2} = -\lambda'_1 \frac{\theta'_1 - \theta'_{sf}}{h'/2} \quad (6.8)$$

```
Boundary {
  isothermal {
    Kind      = "Outer"
    Class     = "Wall"
    Medium    = "Fe"
    Type      = "Fixed"
    ThermalOption = "IsoThermal"
    Temperature = 100.0
  }
}
```

表 6.9 等温壁の指定パラメータ

指定キーワード	パラメータの説明
Temperature	表面温度 [$^{\circ}\text{C}$]

6.2.2 対称境界

外部境界にのみ用いられる境界条件で、指定する面が対称面であると仮定します。図 6.5 に X プラス方向の境界面における対称境界面の速度ベクトルの境界条件を示します。速度については、面直な成分のみ固体壁と同じで、残りはフリーとします。圧力は勾配がゼロとします。

```
Boundary {  
  SymWall {  
    Kind    = "Outer"  
    Class   = "Symmetric"  
    Medium  = "Fe"  
  }  
}
```

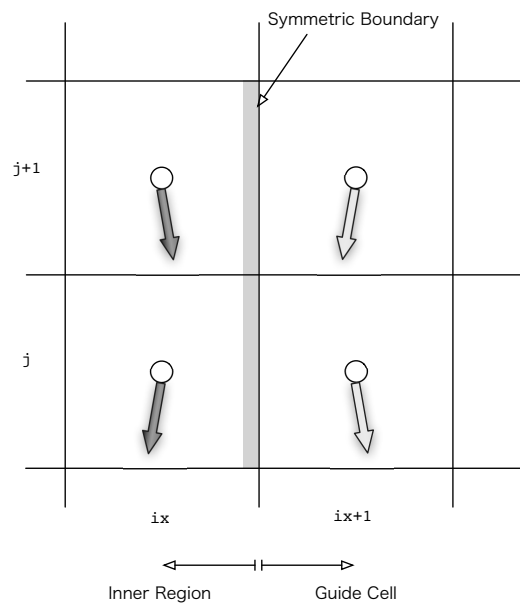


図 6.5 対称境界面における境界条件

熱計算では、対称境界が指定された面は断熱境界となります。

6.2.3 流出境界

流出境界を指定する場合には、流出方向は既知とします。外部境界では図 6.6 に示すようにガイドセルのセル属性は流体である必要があります。

```
Boundary {
  outlet {
    Kind      = "Outer"
    Class     = "outflow"
    Medium    = "air"
    PressureType = "dirichlet"
    PrsValue  = 0.0
  }
}
```

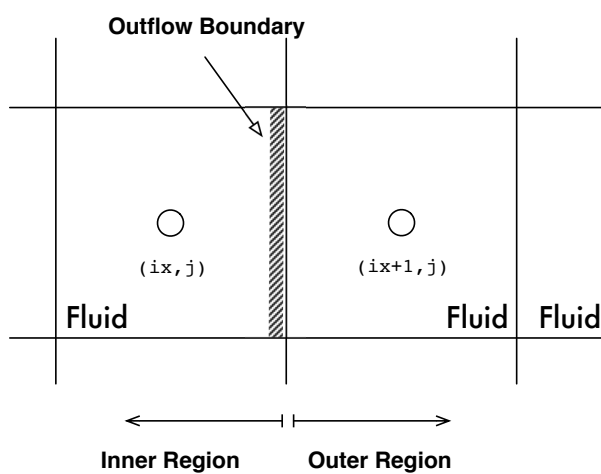


図 6.6 外部境界面における流出境界。x+ 方向の例。

圧力境界条件としては表 6.10 に示す 2 種類の境界条件を与えることができます。

熱境界も速度と同様に対流流出型の境界条件となります。逆流がある場合には、ノイマン条件、つまり計算領域に入る流束をゼロに制限します。

表 6.10 流出圧力境界条件

ラベル	パラメータ
PressureType	Dirichlet Neumann
PrsValue	指定圧力値, または圧力勾配値

6.2.4 速度指定境界

この境界条件はセル界面における運動量流束の形で実装されています。

```
Boundary {
  Inlet_1 {
    kind          = "Outer"
    class          = "SpecifiedVelocity"
    medium         = "Fe"
    type           = "Velocity"
    profile        = "constant"
    velocity       = 3.0
    OrientationVector = (0.0, 0.0, 1.0)
  }

  Inlet_2 {
    kind          = "Outer"
    class          = "SpecifiedVelocity"
    medium         = "Fe"
    type           = "Velocity"
    profile        = "harmonic"
    Amplitude      = 7.0
    Frequency      = 2.0
    InitialPhase   = 0.0
    ConstantBias   = 0.0
    OrientationVector = (1.0, 0.0, 0.0)
    Temperature    = 20.0
  }
}
```

境界面の指定方法は、表 6.11 に示すパラメータを与えます。時間変化を伴う速度指定は Profile=“Harmonic” を指定し、式 (6.1) の形式の単振動の境界条件を周期や初期位相、固定バイアスと共に与えます。時間的に変化しない壁面境界の場合には Profile=“Constant” を指定し、Velocity により速度を指定します。

圧力の境界条件は、Neumann 型の圧力境界条件 $\nabla p = 0$ が用いられます。

熱流れの問題の場合には流入温度を指定する必要があり、単位は Unit ノードの Temperature で指定した単位になります。

表 6.11 速度指定境界のパラメータ

ラベル	指定キーワード	パラメータの説明
Profile	Constant Harmonic	一定、または式 (6.1) の形式
OrientationVector	—	方向ベクトルの成分
Velocity	—	指定単位 $[m/s]$ 、Profile=Constant の場合のみ
Amplitude	—	速度、以下のパラメータは Profile=Harmonic の場合のみ
Frequency	—	周波数 $f [Hz]$
InitialPhase	—	初期位相 $\phi [Rad]$
ConstantBias	—	一定値 $b [m/s]$
Temperature	—	指定温度 $[^{\circ}C]$

6.2.5 周期境界

周期境界条件には、外部境界に対する周期境界と計算内部領域に設定する部分的な周期境界条件を併用する条件の 2 種類があります。外部境界に対する周期境界条件では、図 6.2 において、Inner Region の両端の境界が重なる状態を想定しています。

外部境界に対する周期境界条件には表 6.12 に示す 3 つのモードが指定できます。下記には、各モードの例を示します。SimpleCopy モードは、周期境界条件面の両端で、単純に計算内部領域の値を他方のガイドセルにコピーします。PressureDifference モードは、両端で圧力差を与える周期境界条件で、速度や温度については SimpleCopy モードと同じですが、圧力は指定の圧力差を与えます。上流側と下流側の設定が必要です。Driver モードは、乱流計算などで発達したチャンネル流を上流境界として与えるためのしくみで、局所境界条件との組み合わせで利用します。Driver モードの説明は局所境界条件をご覧ください。

```
Boundary {
  x-dir_periodic_1 {
    Class      = "periodic"
    Mode       = "SimpleCopy"
  }

  x-dir_periodic_2 {
    Class      = "periodic"
    Mode       = "Directional"
    FlowDirection = "Upstream"
    PressureDifference = 8.148e-3
  }

  x-dir_periodic_3 {
    Class      = "periodic"
    Mode       = "Directional"
    FlowDirection = "Downstream"
    PressureDifference = 8.148e-3
  }

  x-dir_periodic_4 {
    Class      = "periodic"
    Mode       = "driver"
    DriverDirection = "Xminus"
  }
}
```

表 6.12 周期境界条件のモード

キーワード	モードの説明
SimpleCopy	周期境界の両端で物理量をガイドセルにコピーします。
Directional	圧力差を与える周期境界条件で、上流と下流の境界面を指定します。
Driver	計算領域内で部分的な周期境界条件を設定します。

Directional モードでは、表 6.13 に示すパラメータが必要で、PressureDifference の値が、Upstream と Downstream で同じ値である必要があります。

表 6.13 Directional モードに必要なパラメータ

必要なキーワード	パラメータの説明
PressureDifference	両端にかかる圧力差 [Pa]
FlowDirection	Upstream（上流面）または Downstream（下流面）

6.2.6 トラクションフリー境界

遠方境界条件として、トラクションフリー条件を用います。

トラクションフリー条件は、外部境界に対してのみ指定できる境界条件で、計算対象の主領域から遠方の挙動を仮定した条件です。つまり、圧力の遠方条件 $p = 0$ （基準圧）を考慮し、計算外部境界において流体の内部応力の法線方向成分がゼロである仮定を用いています。ガイドセルは流体の属性である必要があります。

この境界条件は、噴流のエントレインメントの効果などを考慮できる利点がありますが、渦が流出するような境界には適用できません。

```
Boundary {  
  tfree {  
    kind          = "Outer"  
    Class         = "TractionFree"  
    Medium        = "air"  
  }  
}
```

熱流れの場合には、流出の場合には `outflow` と同じになります。

6.2.7 遠方境界

外挿境界条件で，実験的な実装です．

指定された外部境界面において，外部境界面の値を内部から外挿して与えます．

```
Boundary {  
  far_field {  
    kind          = "Outer"  
    Class          = "FarField"  
    Medium         = "air"  
    AmbientTemperature = 25.0  
  }  
}
```

熱流れの場合には，流入時に対応する温度を指定します．

6.3 局所境界条件

内部領域の境界条件の多くは、計算空間内に局所的に存在し、複雑な計算処理を行います。

表 6.3 を以下に再掲します。

表 6.14 局所境界条件（コンポーネント）の種類。表中の Cold flow は流れのみの場合（FlowOnly）を示す。

Class	境界指定位置	適用	実装形式	コンポーネントの説明
CellMonitor	セル要素	Cold/Thermal flow	—	物理量のモニター位置の指定
DirectHeatFlux	セル界面	Thermal flow	熱流束	熱流束指定
HeatSource	セル要素	Thermal flow	外力項	吸発熱指定
HeatTransferS	セル界面	Thermal flow	熱流束	熱伝達係数と表面温度により計算
HeatTransferSF	セル界面	Thermal flow	熱流束	強制対流の層流・乱流熱伝達境界
HeatTransferSN	セル界面	Thermal flow	熱流束	自然対流の乱流熱伝達境界
IsoThermal	セル界面	Thermal flow	熱流束	等温面指定
Monitor	—	Cold/Thermal flow	—	物理量のモニター位置の指定
Obstacle	セル界面	Cold/Thermal flow	熱流束マスク	断熱物体
Outflow	セル界面	Cold/Thermal flow	対流流束	流出境界
Periodic	セル界面	Cold/Thermal flow	参照値指定	部分周期境界
SolidRevolution	—	Cold/Thermal flow	—	回転体，実験
SpecifiedTemperature	セル要素	Thermal flow	温度指定	温度指定
SpecifiedVelocity	セル界面	Cold/Thermal flow	対流流束	速度指定境界

6.3.1 壁面境界

流れの境界条件

局所境界条件は幾何形状データに紐付けて、パラメータを指定します。下記の例では、plant_m.stl ファイルに物体 (Obstacle) の属性と sus の媒質名を与えています。解析モデルのポリゴンに Obstacle を指定すると、そのポリゴンと交差する位置が固体として認識されます。そのセル界面においては、指定する壁面速度から運動量流束が直接計算され、スキーム中で境界条件として与えられます。

壁面境界に対する圧力の境界条件は、Navier-Stokes 方程式から Neumann 型の圧力境界条件が得られます。高レイノルズ数流れにおいては、粘性項の寄与が小さいと仮定し粘性項を省略し $\nabla p = 0$ の形式になります。Binary 近似の場合には、固体壁面との界面で $\nabla p = 0$ を満たすようにスキームが構成されています。

```
Boundary {
  body {
    kind          = "inner"
    class         = "Obstacle"
    Medium        = "sus"
    filepath      = "geometry/plant_m.stl"
  }
}
```

熱境界条件

壁面に対する熱境界条件としては、断熱、熱流束、熱伝達、等温、温度条件を指定できます。熱境界条件の実装の詳細は inside_ffvc.pdf をご覧ください。

FFV-C では熱境界条件はポリゴンモデルに与えます。多くの場合は流体と固体の界面ですが、固体熱伝導と共役熱移動の場合には、固体-固体界面の場合もあります。熱流束の符号は、図 6.7 において、作動流体から熱が奪

われる場合を負，作動流体に熱を与える場合を正にとります*2。

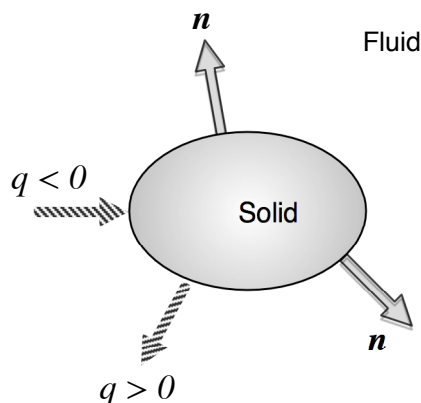


図 6.7 流体-固体界面における熱境界条件の熱流束の方向

FFV-C の熱流体解析には幾つかのモードがあります。HeatEquation の指定モードによって，計算空間内の計算対象とする部分が異なります。

- * Thermal Flow および Thermal Flow Natural
熱流動計算で流体の温度のみを解きます*3。したがって，固体部分は計算対象とはならず不活性セルとして扱います。これより，流体-固体の境界面で与える熱境界条件（熱流束）は，流体セル側のみに指定されます。
- * SolidConduction
固体部分の熱伝導のみを解くので，流体部分を不活性セルとして扱います。したがって，流体-固体の境界面で与える熱境界条件は，固体セル側のみに指定されます。
- * Conjugate Heat Transfer
流体と固体の両方の熱移動を計算します。したがって，流体-固体の境界面では熱境界条件は不要です。精度の高い計算のためには，温度境界層を解像できる格子密度が必要になります。

■断熱境界

Obstacle を指定した場合，壁面は断熱壁，つまり熱流束がゼロ $q' = 0$ となります。

```
Boundary {
  Body {
    kind    = "inner"
    Class   = "Obstacle"
    Medium  = "A1"
  }
}
```

この例では，形状モデルで Body と名付けられたポリゴン表面が断熱壁に指定されています。バイナリモデルでは，Body の固体ボクセルに隣接する流体セルの面が断熱境界条件となります。

*2 固体熱伝導の場合には，作動流体に相当する媒質となります。

*3 計算の実装上，固体部分も解いていますが，その値はマスクされ無効化されています。

■ 熱流束境界

熱流束境界は境界面で指定の熱流束を与えます。Heatflux の単位は $[W/m^2]$ です。

```
Boundary {
  exhaust {
    kind      = "inner"
    Class     = "DirectHeatFlux"
    Medium    = "Fe"
    Heatflux  = 10.0
  }
}
```

■ 熱伝達境界

熱伝達境界は次式の形式で熱流束を与える条件で、幾つかの種類があります。固体-流体セル間の熱伝達境界の与え方は、外部境界条件の熱伝達境界で説明した内容と同じです。

TypeS バルク温度と熱伝達係数により計算

TypeS は流体のバルク温度と熱伝達係数を与え、熱流束を計算します。式 (6.2) において、 θ'_∞ を固体表面に接する流体セルの値と仮定します。

```
Boundary {
  exhaust {
    kind          = "inner"
    Class         = "HeatTransferS"
    Medium        = "Fe"
    BulkTemperature = 300.0
    CoefOfHeatTransfer = 20.0
  }
}
```

TypeSN 自然対流の乱流熱伝達

自然対流の場合の乱流熱伝達の実験式を実装した境界条件です。

```
Boundary {
  catalizer {
    kind          = "inner"
    Class         = "HeatTransferSN"
    Medium        = "Fe"
    BulkTemperature = 500.0
    VerticalLaminarAlpha = 0.59
    VerticalLaminarBeta  = 0.25
    VerticalTurbulentAlpha = 0.1
    VerticalTurbulentBeta = 0.3333333
    VerticalRaCritial    = 1.0e9
    LowerLaminarAlpha    = 0.27
    LowerLaminarBeta     = 0.25
    LowerTurbulentAlpha  = 0.27
    LowerTurbulentBeta   = 0.25
    LowerRaCritial       = 1.0e9
  }
}
```


TypeSF 強制対流の層流・乱流熱伝達

強制対流の場合の層流・乱流熱伝達の実験式を実装した境界条件です。

```
Boundary {
  catalizer {
    kind          = "inner"
    Class         = "HeatTransferSF"
    Medium        = "Fe"
    SuBulkTemperature = 500.0
    Alpha         = 0.037
    Beta          = 0.8
    Gamma         = 0.33333
  }
}
```

■等温壁境界

等温壁境界は、指定面で温度が一定となる境界条件で、面温度を一定に保つような熱流束が発生します。

```
Boundary {
  HeatedWall {
    kind          = "inner"
    Class         = "IsoThermal"
    Medium        = "Fe"
    Temperature   = 300.0
  }
}
```

6.3.2 流出境界条件

流れの流出境界

計算領域内部に設定する流出境界について説明します。局所境界の場合には流出側には、固体セルがあると仮定しています。このため、媒質の指定には固体要素の媒質を指定してください。

ポリゴンが隣接するセルセンター間 $(i, j) - (i + 1, j)$ にあり、セル (i, j) は流体要素とします。このとき、セル (i, j) から見た場合、実際にセル $(i + 1, j)$ は固体要素でなくても、図 6.8 のような状況と考えます。ここで、ハッチング部分、つまり (i, j) セルの固体セルに隣接する面が流出境界として指定されます。速度の流出面における対流速度的評価方法として流出コンポーネントの平均速度を用い、流出界面の対流熱流束 \tilde{f} を一次風上の形式で評価します。流出面における圧力境界は圧力勾配ゼロとしています。圧力の指定については、外部境界面での指定と同様で表 6.10 を参照してください。

```
Boundary {
  Outlet_1 {
    kind          = "inner"
    Class         = "Outflow"
    Medium        = "Fe"
    PressureType  = "Neumann"
    PrsValue      = 0.0
  }
}
```

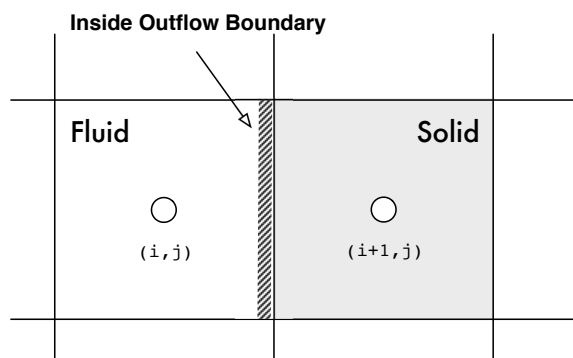


図 6.8 計算内部領域における流出境界の設定

熱流出境界

熱の流出境界は，流出界面の対流熱流束 \tilde{f} を一次風上の形式で評価します．

$$\tilde{f} = \frac{\partial}{\partial x'} (u' \theta')_{upstream_face} \quad (6.9)$$

分離解法において温度輸送方程式を解く過程では，速度は既知なので上式は直ちに計算できます．

6.3.3 速度指定条件

流れの境界条件

この境界条件は，セル界面の運動量流束の形で実装されています．

```
Boundary {
  Inlet {
    kind           = "inner"
    Class          = "SpecifiedVelocity"
    Medium         = "Fe"
    filepath       = "outlet.stl"
    OrientationVector = (0.0, 0.0, -1.0)
    Type           = "Velocity"
    Profile        = "Constant"
    Velocity       = 3.0
    InOut          = "out"
    Temperature    = 60.0
  }
}
```

境界面の指定方法は表 6.15 に示すパラメータを与えます．時間変化を伴う速度指定は Profile="Harmonic" を指定し，式 (6.10) の形式の単振動の境界条件を周期や初期位相，固定バイアスと併に与えます．時間的に変化しない壁面境界の場合には Profile="Constant" を指定し，周波数，初期位相，固定バイアス値の指定は不要です．

$$V = A \sin(2\pi f t + \phi) + b \quad (6.10)$$

InOut は，速度境界条件を与えるセルを基準として，その境界条件が流入条件であるか，あるいは流出条件であるかを指定します．いま，図 6.9 でセル (i,j) とセル (i+1,j) の間に境界条件となるポリゴンがあるとしします．この速度境界の方向を OrientationVector=(-1.0, 0.0, 0.0) と与えると，指定ベクトルは x 軸の負方向のベクトルとなります．セル (i,j) が流体でセル (i+1,j) 方向に流入速度境界を与える場合，In を指定します（図左上）．逆に，

表 6.15 コンポーネントの流束指定のパラメータ

ラベル	指定キーワード	パラメータの説明
OrientationVector	—	指定する速度ベクトルの成分（単位ベクトルでなくてもよい）
Type	Velocity Massflow	速度指定，または流量指定
Profile	Constant Harmonic polynomial6	指定速度のタイプ
Velocity	—	速度 $[m/s]$ または 流量 $[m^3/sec.]$ ，Profile=Constant の場合のみ
Amplitude	—	速度，流量，以下のパラメータは Profile=Harmonic の場合のみ
Frequency	—	周波数 $f [Hz]$
InitialPhase	—	初期位相 $\phi [Rad]$
ConstantBias	—	一定値 $b [m/s]$ または $[m^3/sec.]$
$c_1 \sim c_6$	—	Polynomial6 の係数，Profile=Polynomial6 の場合のみ
InOut	In Out	流体側を基準に流入方向か流出方向かを指定する
Temperature	—	熱計算の場合に流入温度 $[^{\circ}C]$ を指定

流出条件を与える場合には，Out とします（図右下）．境界速度を与えた反対側のセルからは， $i+1/2$ のセルフェイスは固体壁があるものとして計算されます．この実装により，ある境界の両方が流体で，片方だけに速度境界条件を与えることができます．

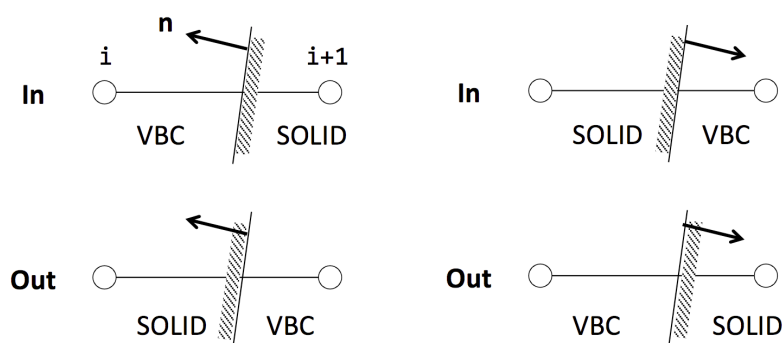


図 6.9 有効な境界条件の方向の指定. 4つのパターン.

熱境界条件

指定面での対流熱流束を式 (6.9) で評価します．

6.3.4 セルボリュームに対する熱境界条件

セル体積要素に作用するコンポーネントの熱境界条件を説明します。この境界条件は、全てのセルに対して適用可能です。

SpecifiedTemperature

以下の形式で指定温度を与えます。

```
Boundary {
  cylinder {
    kind      = "inner"
    Class     = "SpecifiedTemperature"
    Medium    = "Fe"
    Temperature = 600.0
  }
}
```

HeatSource

表 6.16 に示すように、発熱量または発熱密度を指定ポリゴンに与えることができます。発熱量を指定した場合には、ポリゴンで指定されたセルの体積を前処理で計算し発熱密度に変換します。次の例では、cylinder に 100[W] の発熱量を与えています。

```
Boundary {
  cylinder {
    kind      = "inner"
    Class     = "HeatSource"
    Medium    = "Fe"
    HeatReleaseValue = 100.0
  }
}
```

表 6.16 発熱量の指定方法

キーワード	パラメータの種類	単位
HeatReleaseValue	発熱量	[W]
HeatGenerationDensity	発熱密度	[W/m ³]

6.3.5 モニタ

局所境界条件のしくみを用いたサンプリング設定について説明します。計算空間内の任意の位置に配置されたポリゴンに対して、物理量をサンプリングします。これは形式的に局所境界条件の一つとして実装されています。

下記の例では、Sensor と名付けられたポリゴンに対応する領域をモニタ部とし、そこで速度、圧力、全圧をモニタすることを指定しています。ここで OrientationVector はモニタ面の法線方向を指定しています。

Reference は？

```
BCTable {
  LocalBoundary {
    Sensor {
      class          = "CellMonitor"
      Medium         = "air"

      shape          = "polygon"
      OrientationVector = (0.0, 0.0, -1.0)
      SamplingWidth  = "singleCell"

      Variables {
        velocity     = "on"
        pressure     = "on"
        temperature  = "off"
        TotalPressure = "on"
      }
    }
  }
}
```

表 6.17 セルモニタで指定できるパラメータ

ラベル	指定キーワード	コメント
Shape	Cylinder	円筒形状のモニタ領域
	Box	矩形のモニタ領域
	Polygon	ポリゴンによる領域指定
SamplingWidth	SingleCell	ポリゴンと交差するセルのみ
	NeighborCell	ポリゴンの両面のセル
OrientationVector	—	方向ベクトル
Reference	yes no	
Variables		
	Pressure	on off
	Velocity	on off
	Temperature	on off
	TotalPressure	on off

表 6.18 各 shape のパラメータ

Shape	ラベル	コメント
Cylinder	Center	円筒の中心座標 (x, y, z)
	Radius	円筒の半径
	Depth	円筒の深さ
Box	Center	矩形の中心座標 (x, y, z)
	Dir	方向ベクトル (V_x, V_y, V_z)
	Depth	矩形の深さ
	Width	矩形の幅
	Height	矩形の高さ

第 7 章

モニタリング機能

FFV-C ソルバーには、計算中の任意点の物理量をモニターする仕組みがあります。本章では、その指定方法について説明します。

物理量モニタリング機能は、ユーザが指定した位置で指定した物理量をファイルに出力する機能です。位置の指定には、パラメータファイルのみで指定する方法とポリゴンファイルと合わせて指定する方法の2種類があります。

7.1 パラメータファイルで指定する方法

モニタリングの指定は、MonitorList セクションに記述します。

```
MonitorList {
  Log           = "On"
  OutputMode    = "Gather"
  Sampling {
    TemporalType = "Step"
    Interval    = 10
  }

  List[@] {
    Type          = "Line"
    Label         = "grp1"
    SamplingMethod = "Nearest"
    SamplingMode   = "Fluid"
    Division      = 64
    From          = (-0.5, 0.0, 0.0)
    To            = (0.5, 0.0, 0.0)
    Variables {
      Velocity     = "On"
      Vorticity    = "On"
      TotalPressure = "On"
    }
  }

  List[@] {
    Type          = "Pointset"
    Label         = "grp2"
    SamplingMethod = "Interpolation"
    SamplingMode   = "Fluid"
    Set[@] {
      Coordinate   = (0.2, 0.2, 0.2)
      Tag          = "p1"
    }
    Set[@] {
      Coordinate   = (0.3, 0.3, 0.3)
      Tag          = "p2"
    }
    Set[@] {
      Coordinate   = (0.4, 0.4, 0.4)
      Tag          = "p3"
    }
    Variables {
      Pressure     = "On"
      Helicity     = "On"
      TotalPressure = "Off"
    }
  }

  List[@] {
    Type          = "Box"
    Label         = "grp3"
    SamplingMethod = "Nearest"
    SamplingMode   = "Fluid"
    Normal        = (0.0, 0.0, 1.0)
    Center        = (0.0, 0.0, 0.4)
    Depth         = 0.05
    OrientationVector = (1.0, 0.0, 0.0)
    Width         = 0.4
    Height        = 0.2
    Variables {
      Velocity     = "On"
      Pressure     = "Off"
    }
  }
}
```



```

        Temperature = "Off"
        TotalPressure = "Off"
    }
}

List[@] {
    Type          = "Cylinder"
    Label         = "grp4"
    SamplingMethod = "Nearest"
    SamplingMode   = "Fluid"
    Normal        = (0.0, -1.0, 0.0)
    Center        = (0.0, 0.2, 0.0)
    Depth         = 0.1
    Radius        = 0.1
    Variables {
        Velocity   = "On"
        Pressure   = "Off"
        Temperature = "Off"
        TotalPressure = "Off"
    }
}

List[@] {
    Type          = "Polygon"
    Label         = "sensor"
    SamplingMethod = "Nearest"
    SamplingMode   = "Fluid"
    Normal        = (0.0, -1.0, 0.0)
    Variables {
        Velocity   = "On"
        Pressure   = "Off"
        Temperature = "Off"
        TotalPressure = "Off"
    }
}
}

```

表 7.1 モニター機能の設定

ラベル	キーワード	説明
Log	On Off	機能の有効・無効
OutputMode	Gather Distribute	モニターログの出力モード
Sampling/TemporalType	Step Time	間隔の指定形式
Sampling/Interval	—	サンプリング間隔

MonitorList には、線分分割点 (Line)、点群 (Pointset)、直方体領域 (Box)、円柱領域 (Cylinder)、ポリゴン (Polygon)[平面を想定] の 5 種類の指定方法があります。ポリゴンのみ、局所境界条件と合わせポリゴンファイルを使用して指定します。

それぞれをグループと呼び、Pointset の構成点を Set と定義します。ファイルへの出力はグループ毎に書き出されます。MonitorList/List[@]/Label の値は、sampling_[label].txt のようにファイルの名前の中に書き出されます。Line および Pointset については、各 Sampling_[Label].txt ファイルのヘッダ部に、出力点数、出力変数、各点座標が記述されます。

sampling_[Label].txt の他に sampling_info.txt が出力され、MonitorList で設定した出力指定情報がまとめて記述されます。

出力モードは MonitorList/OutputMode ノードで指定します。出力モードは並列計算時のファイル出力様式で、マスターノードに集約してファイル出力する場合には Gather を指定し、分散ノード毎にファイル出力する場合には Distribute を指定します。ファイル名の命名ルールは以下のようになっています。

- 逐次実行時は sampling.txt に、各リストの Label の値 (文字列) を追加したファイル名となります。例えば、Pointset で Label="grp2" と設定されたグループに対しては、sampling_grp2.txt というファイル名となります。
- 並列実行時、OoutputMode="Gather" を指定した場合、ファイル名は逐次と同じになります。
- 並列実行時、OutputMode="Distribute" を指定した場合、上記のファイル名に対して、更にランク番号を追加したファイル名となります。例えば、Line で Label="grp1" と設定されたグループに対しては、sampling_grp1_*.txt となります (*にはランク番号が入ります)。sampling_info.txt については、逐次と同様 Gather モードで出力されます。

MonitorList/List[@]/Variables では、以下のキーワードによりモニタリングする物理量を指定します。

Velocity	速度
Pressure	圧力
Temperature	温度
TotalPressure	全圧
Vorticity	渦度
Helicity	ヘリシティ
Qcriterion	速度勾配テンソルの第2不変量

物理量は複数指定可能です。派生変数は、Output/Data/DerivedVariables ノードで各リーフの値が "Off" になっていても、出力可能です。MonitorList/list[@]/Variables ノードに記述していても、リーフの値が "Off" の場合は出力しません。熱計算ではない (GoverningEquation/HeatEquation="FlowOnly") 場合は、MonitorList/List[@]/Variables ノードに Temperature リーフを書くことはできません。

MonitorList/list[@]/SamplingMethod で指定されるパラメータは、以下のキーワードにより採取方法を指定します。

Nearest	モニタ点を含むセルでの値
Interpolation	三重線形補間
Smoothing	局所平均による平滑化

MonitorList/List[@]/SamplingMode で指定されるパラメータは、以下のキーワードにより採取モードを指定し、各採取方法での対象セルを指定します。

All	全セルを対象
Fluid	流体セルのみを対象
Solid	固体セルのみを対象

7.1.1 値のサンプリング方法

Nearest

モニタ点を含むセルのセル中心での値を採取します。Nearest では SamplingMode="All" に固定です。

Interpolation

モニタ点を囲む 8 つのセルセンター位置での値を採取し、xyz の 3 方向に対して線形補間を行いモニタ点での値を評価します。SamplingMode="All" の場合には、モニタ点を囲む 8 セルの状態 (流体/固体) によらず、常に三重線形補間を行います。SamplingMode="Fluid" の場合には、モニタ点を囲む 8 セル全てが流体セルの場合のみ三重線形補間を行い、それ以外の場合にはモニタ点を含むセルセンターでの値を採取します (Nearest 相当)。SamplingMode="Solid" の場合も同様に、モニタ点を囲む 8 セル全てが固体セルの場合のみ三重線形補間を行い、それ以外の場合にはモニタ点を含むセル中心での値を採取します (Nearest 相当)。

Smoothing

モニタ点を含むセルおよびその隣接セルでのセルセンタの値を採取して、その平均値 (局所平均値) を計算します。SamplingMode="All" の場合には、6つの全隣接セルを用いて、合計7セルでの値を採取して平均します。SamplingMode="Fluid" の場合には、モニタ点を含むセルと、そこに隣接する流体セルのみから平均値を計算します。SamplingMode="Solid" の場合には、モニタ点を含むセルと、そこに隣接する固体セルのみから平均値を計算します。

7.1.2 指定パラメータの制限およびエラー処理

- Pointset グループに所属するモニタ点 (Set) に計算対象領域外の座標値があると、初期化時にエラーメッセージを出力してソルバーの実行を停止します。
- Line グループで指定された線分は、必要なら計算対象領域内にクリッピングしてから、線分上の両端を含む分割点をモニタ点に定めます。線分が完全に計算対象領域外にある場合には、エラーメッセージを出力してソルバーの実行を停止します。このようなケースは図 7.1 に示すような場合が想定されます。
- 採取方法が Interpolation または Smoothing において採取モードが Fluid または Solid の場合には、次の条件を満たすモニタ点があると、ソルバー初期化時に警告メッセージが出力され、ソルバー実行中にはそのモニタ点での採取はスキップされます。

SamplingMode="Fluid" だが、モニタ点を含むセルが固体セルである

SamplingMode="Solid" だが、モニタ点を含むセルが流体セルである

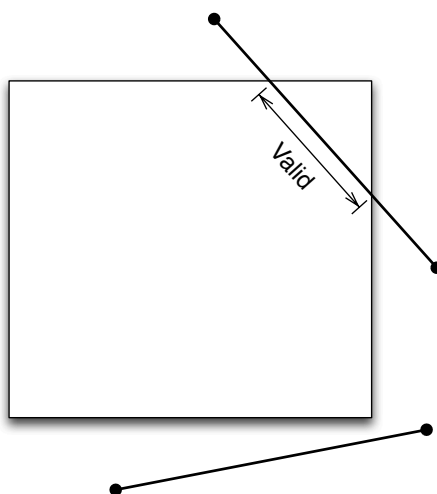


図 7.1 Line のサンプリング指定で無効なケース

7.1.3 出力ファイルフォーマット

採取された物理量は、グループ毎にファイルに出力されます。出力ファイルは、テキストファイルで、Line と PointSet については、モニタ点座標とモニタ変数を記述したヘッダ領域と、それに続く、採取したステップ数個のデータ領域からなります。ヘッダ領域とデータ領域、および、隣接するデータ領域間は1行の空行で区切られています。

Polygon については、先頭行に step, time, モニタ対象の物理量を示すキーワード (Velocity, Pressure 等) が並びます。2行目以降は、MonitorList/Sampling/Interval で指定した間隔で、それぞれの数値が出力されます。Variables の値は面積平均です。

Cylinder, Box については、物理量の値は体積平均です。

ヘッダ領域

1行目の整数 n にモニタ点数と、モニタ対象の物理量を示すキーワード (Velocity, Pressure 等) が並びます。続く n 行に、各モニタ点の座標値およびコメントが出力されます。なお、分散出力時には、 n は担当ノード内のモニタ点数になり、担当モニタ点の座標値のみを出力します。

<pre>n Velocity Pressure Temperature x1 y1 z1 #p1 x2 y2 z2 #p2 ... xn yn zn #p[n]</pre>	<p>← n 点で速度, 圧力, 温度をモニタ</p> <p>← 各モニタ点の座標とコメントを 空白区切りで出力</p>
---	---

データ領域

1行目に、採取時のステップ数 step(整数) とソルバー内部時間 time(実数) が出力されます。続く n 行に、各モニタ点で採取した値が、ヘッダ領域のキーワードの並び順に出力されます。

<pre>step time u1 v1 w1 p1 t1 u2 v2 w2 p2 t2 ... un vn wn pn tn</pre>	<p>← ステップ数=step, 時間=time</p> <p>← モニタ点毎の採取値が 空白区切りで並ぶ (u_i, v_i, w_i)=速度, p_i=圧力, t_i=温度</p>
---	---

採取値の有効桁数は、単精度計算では小数点以下7桁、倍精度計算では16桁です。

なお、採取モードの制限により採取をスキップされたモニタ点では、データ領域の該当する行には、「*NA*」の文字列が出力されます。

7.2 局所境界条件も合わせて指定する方法

Polygon によりサンプリングする場合、MonitorList 内の記述の他に、BcTable に以下のような記述が必要です。

```
BCTable {  
  Boundary {  
    sensor {  
      Kind      = "inner"  
      Class     = "Monitor"  
      Medium    = "Air"  
    }  
  }  
}
```

7.3 モニター例

7.3.1 Line と Pointset

以下の指定によって、10 ステップ毎にサンプリングする例を示します。

Line “Lx” x 軸にそって 5 点 ($x = -0.5, -0.25, 0.0, 0.25, 0.5$)

Line “Ly” y 軸にそって 5 点 ($y = -0.5, -0.25, 0.0, 0.25, 0.5$)

Line “Lz” z 軸にそって 5 点 ($z = -0.5, -0.25, 0.0, 0.25, 0.5$)

Pointset “P8” $x = \pm 0.25, y = \pm 0.25, z = \pm 0.25$ の組み合わせで 8 点

```
MonitorList {
  Log                = "On"
  OutputMode         = "Gather"
  Sampling {
    TemporalType      = "Step"
    Interval          = 10
  }

  List[@] {
    Type              = "Line"
    Label              = "Lx"
    SamplingMethod     = "Nearest"
    SamplingMode       = "Fluid"
    Division           = 4
    From               = (-0.5, 0.0, 0.0)
    To                 = (0.5, 0.0, 0.0)
    Variables {
      Velocity         = "On"
      Pressure          = "On"
      TotalPressure    = "On"
    }
  }
}

List[@] {
  Type              = "Line"
  Label              = "Ly"
  SamplingMethod     = "Nearest"
  SamplingMode       = "All"
  Division           = 4
  From               = (0.0, -0.5, 0.0)
  To                 = (0.0, 0.5, 0.0)
  Variables {
    Velocity         = "On"
    Pressure          = "On"
    TotalPressure    = "On"
  }
}

List[@] {
  Type              = "Line"
  Label              = "Lz"
  SamplingMethod     = "Interpolation"
  SamplingMode       = "All"
  Division           = 4
  From               = (0.0, 0.0, -0.5)
  To                 = (0.0, 0.0, 0.5)
  Variables {
    Velocity         = "On"
    Pressure          = "On"
    TotalPressure    = "On"
  }
}
}
```

```
List[@] {
  Type          = "Pointset"
  Label         = "P8"
  SamplingMethod = "Interpolation"
  SamplingMode   = "Fluid"
  Set[@] {
    Coordinate   = (-0.25, -0.25, -0.25)
    Tag          = "p1"
  }
  Set[@] {
    Coordinate   = ( 0.25, -0.25, -0.25)
    Tag          = "p2"
  }
  Set[@] {
    Coordinate   = (-0.25,  0.25, -0.25)
    Tag          = "p3"
  }
  Set[@] {
    Coordinate   = ( 0.25,  0.25, -0.25)
    Tag          = "p4"
  }
  Set[@] {
    Coordinate   = (-0.25, -0.25,  0.25)
    Tag          = "p5"
  }
  Set[@] {
    Coordinate   = ( 0.25, -0.25,  0.25)
    Tag          = "p6"
  }
  Set[@] {
    Coordinate   = (-0.25,  0.25,  0.25)
    Tag          = "p7"
  }
  Set[@] {
    Coordinate   = ( 0.25,  0.25,  0.25)
    Tag          = "p8"
  }
  Variables {
    Pressure     = "On"
    Helicity     = "On"
    Vorticity    = "off"
  }
}
```

初期化時の出力情報

以下に、4 並列実行時の sampling_info.txt への出力例を示します。

```
>> Sampling Information

Output Type      : Gather
Base Name of Output File : sampling_info.txt
Unit             : Non Dimensional

1 : Line        division=5 [Lx]
   Variables : Velocity Pressure TotalPressure
   Method : Nearest
   Mode : Fluid

           order :           X           Y           Z           : rank : comment
           1 :   -4.999999e-01   0.000000e+00   0.000000e+00 :     3 : point_0
           2 :   -2.499999e-01   0.000000e+00   0.000000e+00 :     3 : point_1
           3 :    2.980232e-08   0.000000e+00   0.000000e+00 :     3 : point_2
           4 :    2.500000e-01   0.000000e+00   0.000000e+00 :     3 : point_3
           5 :    4.999999e-01   0.000000e+00   0.000000e+00 :     3 : point_4

2 : Line        division=5 [Ly]
   Variables : Velocity Pressure TotalPressure
   Method : Nearest
   Mode : All

           order :           X           Y           Z           : rank : comment
           1 :    0.000000e+00  -4.999999e-01   0.000000e+00 :     2 : point_0
           2 :    0.000000e+00  -2.499999e-01   0.000000e+00 :     2 : point_1
           3 :    0.000000e+00   2.980232e-08   0.000000e+00 :     3 : point_2
           4 :    0.000000e+00   2.500000e-01   0.000000e+00 :     3 : point_3
           5 :    0.000000e+00   4.999999e-01   0.000000e+00 :     3 : point_4

3 : Line        division=5 [Lz]
   Variables : Velocity Pressure TotalPressure
   Method : Interpolation
   Mode : All

           order :           X           Y           Z           : rank : comment
           1 :    0.000000e+00   0.000000e+00  -4.999999e-01 :     1 : point_0
           2 :    0.000000e+00   0.000000e+00  -2.499999e-01 :     1 : point_1
           3 :    0.000000e+00   0.000000e+00   2.980232e-08 :     3 : point_2
           4 :    0.000000e+00   0.000000e+00   2.500000e-01 :     3 : point_3
           5 :    0.000000e+00   0.000000e+00   4.999999e-01 :     3 : point_4

4 : PointSet    division=8 [P8]
   Variables : Pressure Helicity
   Method : Interpolation
   Mode : Fluid

           order :           X           Y           Z           : rank : comment
           1 :   -2.500000e-01  -2.500000e-01  -2.500000e-01 :     0 : p1
           2 :    2.500000e-01  -2.500000e-01  -2.500000e-01 :     0 : p2
           3 :   -2.500000e-01   2.500000e-01  -2.500000e-01 :     1 : p3
           4 :    2.500000e-01   2.500000e-01  -2.500000e-01 :     1 : p4
           5 :   -2.500000e-01  -2.500000e-01   2.500000e-01 :     2 : p5
           6 :    2.500000e-01  -2.500000e-01   2.500000e-01 :     2 : p6
           7 :   -2.500000e-01   2.500000e-01   2.500000e-01 :     3 : p7
           8 :    2.500000e-01   2.500000e-01   2.500000e-01 :     3 : p8
```


単一ファイル出力

以下は、4 並列実行時のファイル出力内容 (sampling_Lz.txt) です。最初の 6 行はヘッダで、5 点のモニタ点 (2~6 行目に座標とラベルが示されています) に対して、速度 (u, v, w 成分) と圧力をモニタすることがわかります。各ステップのモニタ値にはステップ数と時刻のヘッダがつきます。

```

5 Velocity Pressure TotalPressure
0.000000e+00 0.000000e+00 -4.999999e-01 #point_0
0.000000e+00 0.000000e+00 -2.499999e-01 #point_1
0.000000e+00 0.000000e+00 2.980232e-08 #point_2
0.000000e+00 0.000000e+00 2.500000e-01 #point_3
0.000000e+00 0.000000e+00 4.999999e-01 #point_4

10 3.125000e-02
-3.8579664e-09 1.6753970e-13 -2.2118146e-17 1.3322930e-15 1.3322930e-15
-4.6404459e-08 1.4141045e-12 -1.9428935e-15 3.9968080e-14 3.9968080e-14
-3.2663900e-07 6.6729634e-12 2.9309891e-14 2.2029227e-10 2.2034691e-10
-1.2792697e-06 1.7829644e-11 6.1417521e-13 -1.2121856e-11 -1.1269204e-11
1.0845728e-04 1.1291234e-11 6.1851235e-13 -3.4674881e-11 1.1728182e-08

20 6.250000e-02
-2.5465420e-06 1.2441052e-11 -2.2204884e-15 1.2147670e-10 1.2795698e-10
-7.8328230e-06 2.8948080e-11 9.6456491e-13 2.3243271e-10 2.6312816e-10
-1.6451937e-05 3.6312862e-11 3.8983938e-11 2.7585900e-10 4.1135187e-10
-3.2475702e-05 5.2842487e-11 1.9095123e-10 -3.0830727e-09 -2.5555660e-09
7.2699040e-04 2.1998901e-11 5.5966318e-11 -4.3365205e-09 5.2417261e-07

30 9.375000e-02
-1.0547113e-05 4.4814920e-12 9.6546837e-13 2.9948526e-09 3.1060405e-09
-3.2290773e-05 2.0184070e-11 1.1753808e-10 4.9389532e-09 5.4605485e-09
-6.0378370e-05 3.8872468e-11 8.7544566e-10 -2.1239039e-09 -2.9950858e-10
-1.1264053e-04 6.8499373e-11 3.2821101e-09 -3.9409763e-08 -3.3063309e-08
2.1987632e-03 4.3962826e-11 6.9003903e-10 -5.0379306e-08 4.7841249e-06

...

```

分散ファイル出力

前述と同条件でのファイル出力例 (sampling_Lz_1.txt) です。

```

2 Velocity Pressure TotalPressure
0.000000e+00 0.000000e+00 -4.999999e-01 #point_0
0.000000e+00 0.000000e+00 -2.499999e-01 #point_1

10 3.125000e-02
-3.8579664e-09 1.6753970e-13 -2.2118146e-17 1.3322930e-15 1.3322930e-15
-4.6404459e-08 1.4141045e-12 -1.9428935e-15 3.9968080e-14 3.9968080e-14

20 6.250000e-02
-2.5465420e-06 1.2441052e-11 -2.2204884e-15 1.2147670e-10 1.2795698e-10
-7.8328230e-06 2.8948080e-11 9.6456491e-13 2.3243271e-10 2.6312816e-10

30 9.375000e-02
-1.0547113e-05 4.4814920e-12 9.6546837e-13 2.9948526e-09 3.1060405e-09
-3.2290773e-05 2.0184070e-11 1.1753808e-10 4.9389532e-09 5.4605485e-09

...

```

7.3.2 Polygon

以下の指定によって、5 ステップ毎にサンプリングする例を示します。

Polygon “Sensor” Sensor ラベルで指定した平面形状ファイルの面積平均

```
MonitorList {
  Log          = "On"
  OutputMode   = "Gather"
  Sampling {
    TemporalType = "Step"
    Interval    = 5
  }

  List[@] {
    Type        = "Polygon"
    Label       = "Sensor"
    SamplingMethod = "Nearest"
    SamplingMode = "Fluid"
    Normal       = (0.0, 0.0, -1.0)
    Variables {
      Velocity    = "On"
      Pressure    = "On"
      TotalPressure = "On"
    }
  }
}
```

Normal には、ポリゴンの法線ベクトルを記述します。

Sensor の Monitor 機能を有効にするには、局所境界条件で以下の記述が必要です。

```
BCTable {
  LocalBoundary {
    Sensor {
      Class      = "Monitor"
      Medium     = "Air"
    }
  }
}
```

また、

```
GeometryModel {
  Source = "polylib.tp"
}
```

のように、GeometryModel/Source の値を polylib.tp（名前は任意）にした場合、polylib.tp 内には以下のような STL の指定と記述が必要です。

```
Sensor {
  Class_name = "PolygonGroup"
  Filepath   = "./sensor.stl"
  Movable    = "false"
  Label      = "Air"
  Type       = "Monitor"
}
```

初期化時の出力情報

以下に、OpenMP (4 threads) 並列実行時の sampling_info.txt への出力例の一部を示します。Polygon に含まれる格子点がすべて書き出されます。

```
>> Sampling Information

Output Type           : Gather
Base Name of Output File : sampling_info.txt
Unit                  : Dimensional

1 : Polygon    division=408 [sensor]
   Variables : Velocity Pressure TotalPressure
   Method    : Nearest
   Mode      : Fluid

order :           X           Y           Z           : rank : comment
  1 : -8.999999e-03 -2.099999e-02 -1.980000e-01 :    0 : point_0
  2 : -6.999998e-03 -2.099999e-02 -1.980000e-01 :    0 : point_1
  3 : -4.999998e-03 -2.099999e-02 -1.980000e-01 :    0 : point_2
  4 : -2.999998e-03 -2.099999e-02 -1.980000e-01 :    0 : point_3
  5 : -9.999982e-04 -2.099999e-02 -1.980000e-01 :    0 : point_4

...

405 :  3.000002e-03  2.100001e-02 -1.980000e-01 :    0 : point_404
406 :  5.000002e-03  2.100001e-02 -1.980000e-01 :    0 : point_405
407 :  7.000002e-03  2.100001e-02 -1.980000e-01 :    0 : point_406
408 :  9.000002e-03  2.100001e-02 -1.980000e-01 :    0 : point_407
```

ファイル出力

以下は、逐次実行時のファイル出力内容 (sampling_sensor.txt) です。

step	time[sec]	Velocity [m/s]	Pressure [pa]	TotalPressure[pa]
5	6.662225e-06	4.3269503e-23	-8.3969249e-20	-8.3969249e-20
10	1.332445e-05	6.7001661e-13	-2.6877984e-09	-2.6877984e-09
15	1.998668e-05	6.3254566e-09	-3.3081153e-05	-3.3081153e-05
20	2.664890e-05	7.9700573e-07	-4.6390090e-03	-4.6390090e-03
25	3.331113e-05	1.3540335e-05	-8.2405679e-02	-8.2405679e-02
30	3.997335e-05	7.9157158e-05	-4.9071237e-01	-4.9071237e-01
...				

7.3.3 Cylinder と Box

以下の指定によって、5 ステップ毎にサンプリングする例を示します。

Box “Box1” 0.4×0.2×0.05 の直方体領域の体積平均

Cylinder “Cyl2” 半径 0.1, 高さ 0.1 の円筒の体積平均

```
MonitorList {
  Log                = "On"
  OutputMode         = "Gather"
  Sampling {
    TemporalType      = "Step"
    Interval          = 5
  }

  List[@] {
    Type              = "Box"
    Label             = "Box1"
    SamplingMethod     = "Nearest"
    SamplingMode       = "Fluid"
    Normal             = (0.0, 0.0, 1.0)
    Center             = (0.0, 0.0, 0.4)
    Depth             = 0.05
    OrientationVector  = (1.0, 0.0, 0.0)
    Width              = 0.4
    Height             = 0.2
    Variables {
      Velocity         = "On"
      Pressure         = "On"
      Temperature      = "On"
    }
  }
}

List[@] {
  Type              = "Cylinder"
  Label             = "Cyl2"
  SamplingMethod     = "Nearest"
  SamplingMode       = "Fluid"
  Normal             = (0.0, -1.0, 0.0)
  Center             = (0.0, 0.2, 0.0)
  Depth             = 0.1
  Radius             = 0.1
  Variables {
    Velocity          = "On"
    Pressure           = "On"
    TotalPressure      = "On"
  }
}
```

表 7.2 に、Box, Cylinder の場合のパラメータについて記述します。

表 7.2 Box, Cylinder のパラメータ

Shape	ラベル	コメント
Box	Normal	流れの方向 (OrientationVector) が x 正の方向のとき, 右手系で z の方向
	Center	Normal に垂直な底面矩形の中心座標 (x, y, z)
	Depth	Normal 方向の矩形の長さ
	OrientationVector	流体の速度ベクトル (V_x, V_y, V_z)
	Width	底面矩形の OrientationVector 方向の長さ
	Height	底面矩形のもう一辺の長さ
Cylinder	Normal	円筒上面に垂直な, 円筒深さ方向の単位ベクトル
	Center	円筒上面の中心座標 (x, y, z)
	Depth	円筒の深さ
	Radius	円筒の半径

第 8 章

ファイル管理とリスタート

FFV-C は大規模並列計算を実用計算に適用するための多くのメカニズムが組み込まれています。その中でも、分散並列ファイルの管理方法とリスタートは特徴的なもので、CDMLib が提供する機能により実現されています。本章では、FFV-C の並列ファイル管理とリスタート方法について説明します。

8.1 分散並列ファイル管理

大規模な並列計算では、多数のプロセスが計算とファイル入出力を同時に実行します。計算機のデバイスの中でも、ハードディスクの動作は、メモリやキャッシュなどに比べて桁違いに遅いため、アプリケーションの中でファイル入出力に関連する時間コストは少なくありません。特に、プロセス数が増えると、同時に書き込みや読み出しがおこるので、プロセス間のデバイス利用の競合が発生するために性能低下が避けられません。この点を改善するために、ハードウェアレベルでは RAID や Lustre ファイルシステムなど、また運用面ではグローバルファイルシステムとローカルファイルシステムを用いたステージング技術などが提案されています。

一方、ソフトウェアからのアプローチとしては、並列ファイル I/O という方法が提案されています。MPI-IO, netCDF, HDF5, Adios などがミドルウェアとしてよく用いられるものです。FFV-C では、アプリケーションの立場から扱いやすいファイル管理方法を提案しています。基本的な考え方は、次のようになります。

- 1 プロセスで 1 つのファイルを扱う基本的な方法を採用します。この方法は多くのシステムで動作し、ハードウェアの基本的な性能を引き出せます。
- 多数のファイルが生成されることとなりますが、その管理はメタファイルで行い、ユーザは個々のファイルに触らない方針です。
- もし、性能上問題が生じた場合には、複数のファイルを一つにまとめる MPI-IO 的な処置を行います。

分散されたファイル自体の数や保存方法など細かなことを内部的に処理し、ユーザに対してはメタデータの形で必要な情報のみを提示することにしています。これにより、様々なリスタート機能を提供できるとともに、リスタート時の指定が自動化されます。

8.1.1 CDMlib の提供するファイルフォーマット

CDMlib はファイルの分散管理を行うメタファイルと計算結果ファイルの実体から構成されます。計算結果ファイルは、sph フォーマット、plot3d フォーマットを選択できます*1。

8.1.2 メタファイル

分散ファイル管理を行うメタファイルには以下のものがあります。

- proc.dfi(ファイルプロセス情報ファイル)
並列プロセスのランク番号や領域分割情報を保持します。
- index.dfi(インデクスファイル)
結果の生データファイルに関する情報を保持します。sph ファイルフォーマットを選択すると、各変数毎に index ファイルが作成されます。plot3d の場合には 1 ファイルに全変数を書き出すので、index ファイルは一つです。

これらのメタファイルは、ユーザが直接記述することはありません。最初に FFV-C が実行されると自動的に生成され、リスタート時や後のデータ解析や可視化時に利用されます。

プロセス情報ファイル (proc.dfi)

並列計算時のプロセス情報と各プロセスで担当する計算領域（サブドメイン）の分担範囲の情報を持ちます。

Domain ノードは計算領域の情報を示します。これは FFV-C の DomainInfo パラメータと同じです。

MPI ノードは、並列実行したプロセスグループ数とプロセス数を示します。

*1 2015 年 7 月 25 日現在

Process ノードは、各ランク番号^{*2}のプロセスが担当する計算領域と計算機名を示します。下記の例では、[64×64×64]の領域を3等分に分割して計算しています。GlobalDivision の情報からは、3×1×1に分割、つまりx方向に3分割されており、また VoxelSize の情報から各担当計算量が分かります。HeadIndex と TailIndex は、各サブドメインが担当する領域がグローバルドメインのどの部分に相当するかを、矩形サブドメインのボクセルインデックスの最小値と最大値で表しています。

```
Domain {
  GlobalOrigin      = (-5.000000e-01, -5.000000e-01, -5.000000e-01)
  GlobalRegion      = (1.000000e+00, 1.000000e+00, 1.000000e+00)
  GlobalVoxel       = (64, 64, 64)
  GlobalDivision    = (3, 1, 1)
  ActiveSubdomainFile = ""
}

MPI {
  NumberOfRank      = 3
  NumberOfGroup     = 1
}

Process {
  Rank[@] {
    ID      = 0
    HostName = "Strontium.local"
    VoxelSize = (22, 64, 64)
    HeadIndex = (1, 1, 1)
    TailIndex = (22, 64, 64)
  }
  Rank[@] {
    ID      = 1
    HostName = "Strontium.local"
    VoxelSize = (21, 64, 64)
    HeadIndex = (23, 1, 1)
    TailIndex = (43, 64, 64)
  }
  Rank[@] {
    ID      = 2
    HostName = "Strontium.local"
    VoxelSize = (21, 64, 64)
    HeadIndex = (44, 1, 1)
    TailIndex = (64, 64, 64)
  }
}
```

インデクスファイル (prs.dfi, vel.dfi, fvel.dfi, tmp.dfi など)

FFV-C は圧力、速度、温度などの変数毎にファイルを出力します。したがって、1変数につき1つの index.dfi ファイルが生成されます。表 8.1 に、出力される dfi ファイルの標準名を示します。

下記に vela.dfi の例を示します。index.dfi ファイルはファイル情報 (FileInfo)、ファイルパス情報 (FilePath)、単位系 (Unit)、時系列データ (TimeSlice) の4つのブロックで構成されています。各ブロックの内容については、表 8.2～表 8.5 に示します。

```
FileInfo {
  DirectoryPath      = "hoge"
  TimeSliceDirectory = "off"
  Prefix            = "vela"
  FileFormat         = "sph"
  GuideCell          = 0
  DataType           = "Float32"
  Endian             = "little"
}
```

^{*2} 実行時にプロセスに割り当てられた固有の番号

表 8.1 DFI ファイルの命名規則

変数		接頭辞
瞬時値	圧力	prs
	速度 (セルセンタ)	vel
	速度 (セルフェイス)	fvel
	温度	tmp
	速度の発散値	div
平均値	圧力	prsa
	速度	vela
	温度	tmpa
派生変数	ヘリシティ	hlt
	総圧	tp
	渦度	vrt
	速度勾配テンソルの第二不変量	qcr

```

ArrayShape      = "nijk"
Component       = 3
Variable[@]{ name = "u" }
Variable[@]{ name = "v" }
Variable[@]{ name = "w" }
}

FilePath {
  Process = "./proc.dfi"
}

UnitList {
  Length {
    Unit      = "NonDimensional"
    Reference = 1.000000e+00
  }
  Pressure {
    Unit      = "NonDimensional"
    Reference = 0.000000e+00
    Difference = 1.176300e+00
  }
  Velocity {
    Unit      = "NonDimensional"
    Reference = 1.000000e+00
  }
}

TimeSlice {
  Slice[@] {
    Step = 110
    Time = 3.437500e-01
    AverageTime = 1.2
    AverageStep = 200
    VectorMinMax {
      Min = 5.323000e-06
      Max = 1.194742e-01
    }
    MinMax[@] {
      Min = -1.916740e-02
      Max = 1.194742e-01
    }
    MinMax[@] {
      Min = -2.030298e-03
      Max = 2.030273e-03
    }
    MinMax[@] {
      Min = -6.064814e-02
      Max = 5.416530e-02
    }
  }
}

```

```

    }
}
Slice[@] {
    Step = 120
    Time = 3.750000e-01
    VectorMinMax {
        Min = 5.728970e-06
        Max = 1.455714e-01
    }
    MinMax[@] {
        Min = -2.360172e-02
        Max = 1.455714e-01
    }
    MinMax[@] {
        Min = -2.727196e-03
        Max = 2.727167e-03
    }
    MinMax[@] {
        Min = -7.470629e-02
        Max = 6.425590e-02
    }
}
...
}

```

表 8.2 FileInfo ノードの記述内容

リーフラベル	指定内容
DirectoryPath	出力ディレクトリのパス名
TimeSliceDirectory	タイムスライスの指定 "on" "off"
Prefix	表 8.1 に示すプリフィクス (接頭辞) が記述される
FileFormat	"sph" "bov"
GuideCell	出力するガイドセル数
DataType	データの精度 "Float32" "Float64"
Endian	エンディアン "little" "big"
ArrayShape	配列の要素並び指定 "ijkn" "nijk" (スカラの場合は"ijkn")
Component	配列成分数 (スカラ=1 ベクトル=3)
Variable	配列成分数が 2 以上の場合, 成分名が記述されます

DirectoryPath は, 計算結果を書き出すディレクトリを指定します. FFV-C アプリケーションを起動したディレクトリを基点とした相対パス, あるいは絶対パスで指定します.

TimeSliceDirectory は, 各時刻毎に 1 つのディレクトリを DirectoryPath の下に作成することを指定します. これは非常に多数のプロセスで計算する場合に 1 つのディレクトリ内のファイル数が増えることを避けるための手段として使います.

ファイル名は prefix に従い, 次のような規則で生成されます. ext は FileFormat で指定する拡張子になります.

- 逐次実行時

[Prefix]_[ステップ番号:10 桁].[ext]

- 並列実行時

[Prefix]_[ステップ番号:10 桁]_id[RankID:6 桁].[ext]

データタイプは, 単精度 float32 と倍精度 float64 の指定が選べます. ただし, 倍精度の場合には, Cutlib, CPMlib, FFV-C のコンパイルを倍精度オプションをつけてビルドする必要があります*3.

*3 CDMlib 自体は, Int8, UInt8, Int16, UInt16, Int32, UInt32, Int64, UInt64, Float32, Float64 の指定が可能です.

エンディアンは、省略時には実行プラットフォームと同じです*4。

配列の要素並びは、3, 4 次元の配列要素の指定を示します。ijkn は Fortran で (imax,jmax,kmax,Component) と宣言された配列を示し、nijk は逆に (Component,imax,jmax,kmax) と宣言された配列であることを示します。

*4 Intel 系は little, IBM, Sparc 系は big

表 8.3 FilePath ノードの記述内容

リーフラベル	指定内容
Process	プロセスファイルのパス名

プロセスファイルへのパス名が相対パスあるいは絶対パスで記述されています。

表 8.4 UnitList ノードの記述内容

変数	リーフラベル	指定内容
Length		長さの単位
	Unit	"mm" "cm" "M" "NonDimensional"
	Reference	代表長さ
Pressure		圧力の単位
	Unit	"Pa" "NonDimensional"
	Reference	基準圧力
	Difference	代表圧力差
Velocity		速度の単位
	Unit	"m/s" "NonDimensional"
	Reference	代表速度
Temperature		温度の単位
	Unit	"C" "NonDimensional"
	Reference	基準温度
	Difference	代表温度差

FFV-C で利用可能な単位系は上記のようになります。

表 8.5 TimeSlice ノードの記述内容

リーフラベル	指定内容
Step	出力時のステップ
Time	出力時の時刻
AverageTime	平均時間が必要に応じて出力される
AverageStep	平均化したステップ数が必要に応じて出力される
VectorMinMax	ベクトルのユークリッドノルム $\sqrt{u^2 + v^2 + w^2}$
	Min ベクトルノルムの最小値
	Max ベクトルノルムの最大値
MinMax[@]	各成分の最小値と最大値
	Min 最小値
	Max 最大値

TimeSlice ノードには、時系列のファイルに関する内容が記述されています。平均時間と平均ステップ数は平均操作した時間とステップ数で、平均処理がされていない場合には出力されません。

ベクトルデータの場合には、ベクトルのユークリッドノルムが書かれ、その後に u, v, w 各成分毎の最小値と最大値が書き出されています。

8.1.3 ファイルフォーマット

FFV-C は直交格子ソルバーですので、各軸方向の格子幅は一定です。この特徴を活かして、ファイルサイズが少なくなるようなファイルフォーマットを採用しています。一つは、VCAD プロジェクトで採用した sph ファイルフォーマットで、もう一つは VisIt で用いられる bov(Brick of Value) ファイルフォーマットです。どちらも、簡単なヘッダと値の生データを記述するという方式です。sph ファイルはヘッダ自身もファイル内に記述 (ascii/binary) されていますが、bov ファイルはヘッダのみ別ファイルでアスキーで記述され、データ自体はバイナリーで記述可能です。

可視化やデータ処理には他の汎用的なフォーマットが便利ですが、それらは汎用的なためにファイルサイズが大きくなる問題点があります。そこで、PLOT3D, AVS フィールドファイル, vtk 形式へのファイルコンバータを用意しています。

ファイルフォーマットの詳細は、CDMLib ユーザガイドをご覧ください。

8.2 リスタート

FFV-C アプリケーションは、様々な活用場面が考えられます。例えば、計算の途中から実行するプロセス数や解像度を変更したい場合、あるいは計算機自体を変更する場合などがあります。このような場合には、前回計算した結果からプロセス数や解像度、計算機の変更があってもそれに対応できる仕組みが必要になります。

FFV-C では、次のようなリスタートシナリオで計算が可能です。

- 標準リスタート

計算環境とプロセス数、解像度に変更がなく、書き出したファイルをそのまま読み、リスタートします。逐次計算のケースでは、必ずこの標準リスタートになります。

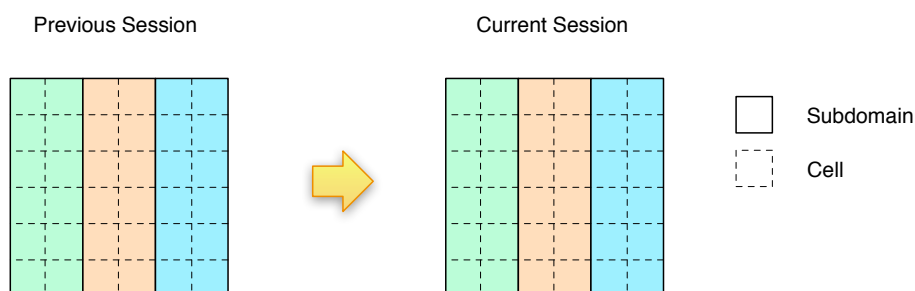


図 8.1 同一解像度、同一プロセスのリスタート、3 プロセスでの実行。

- 同一解像度でプロセス数、またはランクマッピングが異なるリスタート

同一解像度（同じボクセルサイズ）でリスタート時に、前回のセッション^{*5}とは異なるプロセス数で計算します。集合運用の計算機では、利用可能なプロセス数がバッチジョブ投入時に異なりますが、その時々で最適なプロセス数で計算ができます。

同じプロセス数でも空間分割数が異なると、各ランクが担当するサブドメインの大きさは異なるので、通常のリスタートはできません。

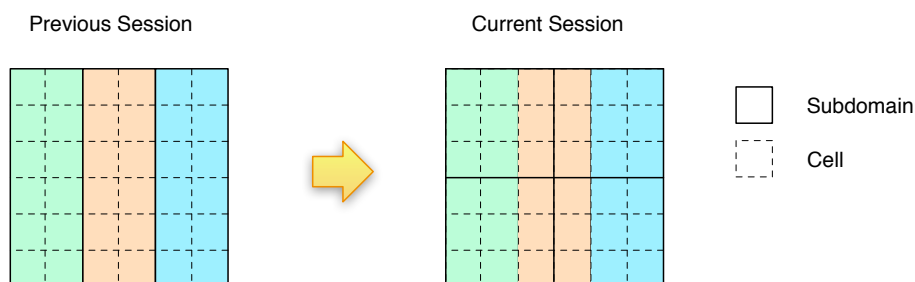


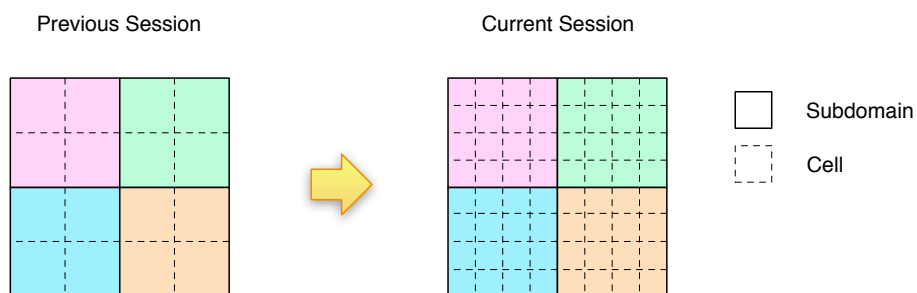
図 8.2 同一解像度、異なるプロセスマッピングでのリスタート、3 プロセスで実行した結果をロードし、4 プロセスでリスタート実行する。

- 細分化リスタート

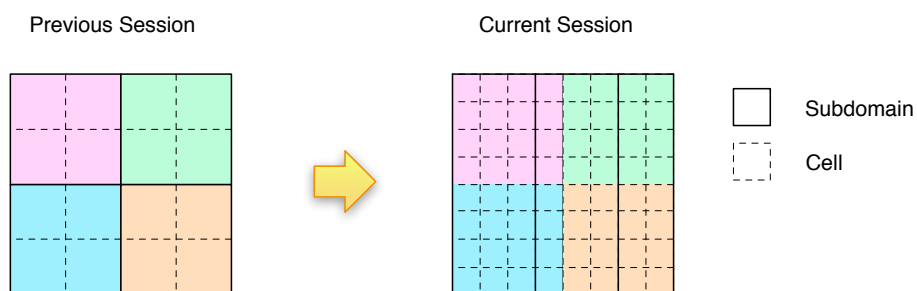
準定常状態が初期値やそれに至る非定常状態への依存性が小さいケースについては、過渡応答よりも準定常状態の流れの様子を早く知りたい場合に利用できる加速方法です。計算初期に流れを発達させる段階では粗い格子を用い、準定常的な状態に近づくに従い、細かい格子を用いてリスタートを行います。つまり、前回のセッションとは異なる解像度で計算を行います。この場合、解像度は各方向 2 倍ずつ細くなるようにします。この細分

^{*5} ある計算対象の計算を複数回のリスタートで計算する場合、1 回の計算をセッションと呼ぶことにします。

化リスタートは、三軸方向に対して内挿処理をするため、二次元問題の場合には利用できません。



(a) 同一計算領域



(b) 異なる計算領域

図 8.3 細分化リスタート。

- ステージング

京コンピュータなどのようにステージング運用を行う計算機システムの場合には、計算に先立ってリスタートに必要なファイルを予め実行予定のランクのローカルディスクにファイルを転送しておく必要があります。このためには、計算前に各ランクで必要なファイルを特定し、ローカルファイルシステムに転送する準備をしておく必要があります。

- 計算機の変更

途中から計算機を変更することは、大規模計算ではあまりないと思いますが、小規模の計算を手元で実行して、その結果を基に大規模な計算を行うようなことを想定しています。この場合には、計算機システムによって異なるエンディアンへの扱いに対応することと、もちろんソースコードをコンパイルできることが必要です。

8.2.1 標準リスタート

FFV-C の基本的なリスタート方法です。最初のセッションが終了すると、proc.dfi, index.dfi ファイルが生成されます。リスタート時には TimeControl ノードのパラメータを変更することにより、リスタート指示を与えます。

例えば、時制として step を指定し、300 ステップからリスタートすることを指定する場合には、次のように Start=300 とします。リスタート後のセッションは 2000 ステップ目で終了します。

```
TimeControl {
  ...
  Session {
    TemporalType = "step"
    Start       = 300
    End         = 2000
  }
}
```

```
Average {
  TemporalType = "step"
  Start        = 400
  End          = 1800
}
}
```

一方、時制として time を指定した場合には、リスタート時にはリスタート開始のステップ数を明示的に指定します。

```
TimeControl {
  ...
  Session {
    TemporalType = "time"
    Start        = 0.2
    End          = 20.0
    RestartStep  = 31
  }

  Average {
    TemporalType = "time"
    Start        = 2.0
    End          = 19.0
    RestartStep  = 672
  }
}
```

前セッションが終了すると、index ファイルが生成されるので、そのファイルを見ると、TimeSlice セクションに出力ファイルのステップ数と時刻が記録されているので、その値を見てリスタートの指定を行います。

```
TimeSlice {
  .
  .
  .
  Slice[@] {
    Step = 21
    Time = 0.0997
    ...
  }

  Slice[@] {
    Step = 31
    Time = 0.2
    ...
  }
  .
  .
  .
}
```

■平均値とリスタート 平均値の操作とリスタートとの関係を説明します。Average/Start で指示される開始点は、平均化操作を始めるステップです。上記の例では、400-1800 ステップの間、1400 ステップ分の時間平均を行います。リスタート時には、まだ平均操作開始点に到達していないので、平均操作は行われません。したがって、平均値のファイルは読み込みません。

一方、もし Average/Start=200 と指定されていたとしたら、リスタート時には既に平均値操作が行われ、ファイルが出力されていると見なします。したがって、平均値のファイルをロードし、平均値操作を継続します。もし、ロード時にファイルがなければエラーとなります。また、平均操作が Initial セッションで始まっていて、Restart セッションでも引き続き平均操作を行う場合は下記を追加する必要があります。

```
StartCondition {
  Restart {
    DFIfiles {
```



```

    AveragedVelocity = "vela.dfi"
    AveragedPressure = "prsa.dfi"
    AveragedTemperature = "tmpa.dfi"
  }
}
}

```

8.2.2 同一解像度でプロセス数が異なるリスタート

このリスタート方法は、解像度は同じ（同一ボクセルサイズ）で、プロセス数だけ変化する場合に利用します。リスタート時に利用できるプロセス数が違う場合です。

8.2.3 細分化リスタート

細分化リスタートは、準定常状態の流れを短時間で計算するために用いられます。通常は次の方法で細分化リスタートを繰り返します。

1. 粗い格子を用いてタイムステップを大きくとり、流れを発達させます。
2. 細分化を行う場合には、粗格子の半分のセル幅の格子を指定し、粗格子で計算した結果を初期値としてリスタート計算を行います。
3. 上記の手順を数回繰り返し、所望の格子解像度での計算を行います。

この方法は、リスタート時に粗格子から倍の解像度を持つ格子に内挿処理を行います。したがって、リスタート時の初期値は方程式を満足しないため反復回数が増加しますが、馴染ませると収束に向かいます。

計算を実行すると、*.dfi ファイルが生成されます。各出力ファイルに対して、表 8.6 に示すようなデフォルト名が付けられています。

表 8.6 dfi ファイルのファイル名

変数	dfi のファイル名
圧力	prs
速度	vel
温度	tmp
圧力平均値	prsa
速度平均値	vela
温度平均値	tmpa
発散値	div
ヘリシティ	hlt
全圧	tp
速度勾配テンソルの第二不変量 λ_2	qcr
渦度	vrt

この dfi ファイルには、8.1.2 節に示すように、並列計算時に各プロセスが書き出すファイルを管理する情報が書かれています。

粗格子の結果を用いてリスタートするケースについて説明します。最初に 10mm の格子を用い、順次 5mm, 2.5mm と細くなる場合を想定します。10mm 格子の計算が終了し、prs.dfi ができます。次に、5mm の計算を行う時には DFI ファイルを適当なファイル名にリネームし、例えば prs_0.dfi として、パラメータファイルを次のように指定します。

```

StartCondition {
  Restart {

```

```
DFIfiles {  
  Velocity    = "vel_0.dfi"  
  Pressure    = "prs_0.dfi"  
  Fvelocity   = "fvel_0.dfi"  
}  
}  
}
```

リスタート後の計算が終了すると、`prs_.dfi` ができますので、次のリスタート時には `prs_1.dfi` として、上記の手順を再帰的に繰り返して計算を進めます。

8.2.4 ステージング

京コンピュータなどのようにステージング運用を行う計算機システムの場合には、計算に先立ってリスタートに必要なファイルを予め実行予定のランクのローカルディスクにファイルを転送しておく必要があります*6。

上記のプロセス数が異なるリスタートや細分化リスタートを実行するためには、各ランクで必要なファイルを特定し、ローカルファイルシステムに転送する準備をしておく必要があります。このとき、どのランクがどのファイルを必要とするかは、前回のセッションの DFI 情報を用いて、転送するファイルをランクディレクトリにまとめます。このためのツールとして `frm` を使います。

`frm` はログインノードで利用するツールで、ログインノードで動作するようにコンパイルしておく必要があります。`frm` の実行方法については、CDMLib ユーザガイドをご覧ください。

*6 ファイル数やファイルサイズが小さい場合、あるいはプロセス数が少なく、同じファイルに各ランクが同時にアクセスしてもコストが小さい場合には、共有領域を利用する方法もあります。その場合には、ここで述べる方法を利用しなくてもよいですが、I/O 性能を高めるためには必要となります。ここでは、京コンピュータを想定しています。

第 9 章

ソルバーの実行

FFV-C ソルバーの実行方法と出力ファイルについて説明します.

9.1 FFV-C ソルバーの実行

次のようなディレクトリ構成を仮定し、3Dcavity の例題を実行します。標準の `install_*.sh` でコンパイルすると、コンパイル済みの実行モジュールは指定した FFVC ディレクトリ配下 `bin` ディレクトリに格納されます。パラメータファイルは、`cavity.tp` とします。

```
Examples
|
+- 3DCavity
|   +-cavity.tp
|
:
```

カレントディレクトリを `Examples/3DCavity` とし、実行モジュールのディレクトリにパスを通しておくと、以下のよう
に実行できます。

```
$ ffvc cavity.tp
```

9.2 出力ファイル

9.2.1 出力ファイルの種類と指定

FFV-C ソルバーを実行すると、表 9.1 に示すファイルが生成されます。また、履歴ファイルについては、Output/Log ノードで生成の有無を指定します。

表 9.1 実行時に生成されるファイル

カテゴリ	ファイル名	出力内容
解析条件情報	condition.txt	計算条件, 前処理, ソルバー起動時のログ
領域情報	DomainInfo.txt	並列計算時の計算領域の分割に関する情報
性能情報	profiling.txt	実行時間タイミングサマリー出力ファイル
基本履歴	history_base.txt	ステップ数, 時刻, 反復回数, 収束状況などの情報
コンポーネント履歴	history_compo.txt	局所境界における関連物理量出力
流量収支履歴	history_domainflux.txt	計算外部領域における流入出流量, 平均速度の情報
反復履歴	history_iteration.txt	反復解法の収束履歴
サンプリング履歴	sampling.txt	MonitorList 指定時の出力ファイル
瞬時値データ	vel_*.sph	速度の瞬時値
	prs_*.sph	圧力の瞬時値
	tmp_*.sph	温度の瞬時値
	fvel_*.sph	セルフェイスでの速度の瞬時値
平均値データ	vela_*.sph	速度の時間平均値
	prsa_*.sph	圧力の時間平均値
	tmpa_*.sph	温度の時間平均値
派生データ	tp_*.sph	全圧
	vrt_*.sph	渦度
	hlt_*.sph	ヘリシティ
	qcr_*.sph	速度勾配テンソルの第 2 不変量
リスタート用	index.dfi	各物理量のインデクスファイル
	proc.dfi	プロセス情報ファイル

表 9.1 の履歴と瞬時値・平均値のデータ出力については、出力インターバルを指定できます。出力インターバルは Output/Data ノードに記述し、各項目独立に、ステップと時刻のどちらによっても指定可能です。サンプリング履歴の出力に関しては、MonitorList で指定できます。ファイル名のアスタリスク*には、ステップ数やランク番号などの情報が入ります。また、拡張子の .sph は sph 形式を表しており、PLOT3D フォーマットもサポートします。

9.2.2 解析条件情報 [condition.txt]

ソルバーを実行すると、`condition.txt` ファイルが生成されます。このファイルはソルバーの起動時のログで、必要な境界条件の設定に係わる前処理、チェック内容などが表 9.2 に示す各セクション毎に記録されています。

表 9.2 condition.txt ファイルの表示項目

セクション名	表示内容
Tables	入力ファイルに記述された媒質数と局所境界条件数
Medium List	媒質情報
Cut Info	交点計算情報
Memory required for Cut	交点計算に必要なメモリ要求量（概算）
Global Domain Information	計算領域の寸法、配列サイズ、格子ピッチ、原点座標
Memory required for Preprocessor	前処理に必要なメモリ要求量（概算）
Components	各コンポーネントの要素数、媒質数など
Information of Scanned Voxel ID	格子生成で利用する内部ボクセル情報
Comparison of Component normal and area	ポリゴンモデルの境界条件情報と認識した情報との違い
Component Information	各コンポーネントの詳細な情報
Restart	リスタート時の情報
Library Information	リンクしたライブラリのバージョン情報
Solver Control Parameters	制御パラメータ
Simulation Parameters	物理量のパラメータ
Initial Values for Physical Variables	初期値の情報
Intrinsic Parameter	組み込み例題を用いた場合の情報
Effective cells and Open Area of Computational Domain	計算対象セル数と各外部境界面における開口面積の割合
Outer Boundary Conditions	外部境界条件
Memory required for Solver	ソルバー本体に必要なメモリ要求量（概算）

9.2.3 領域情報 [DomainInfo.txt]

領域全体の情報、および領域分割された各サブドメインの配列サイズ、領域サイズ、原点座標の情報が表示されます。また、各領域に含まれる境界条件コンポーネントの BoundingBox のインデクス情報が含まれます。

```
>> Global Domain Information

imax, jmax, kmax      =           73           28           28

(dx, dy, dz) [m] / [-] = ( 2.0000e-02 2.0000e-02 2.0000e-02) / ( 3.5714e-02 3.5714e-02 3.5714e-02)
(ox, oy, oz) [m] / [-] = ( 2.0000e-02 2.0000e-02 2.0000e-02) / ( 3.5714e-02 3.5714e-02 3.5714e-02)
(Lx, Ly, Lz) [m] / [-] = ( 1.4600e+00 5.6000e-01 5.6000e-01) / ( 2.6071e+00 1.0000e+00 1.0000e+00)

Domain    0
ix, jx, kx      [-] =           37           28           28
(ox, oy, oz) [m] / [-] = ( 2.0000e-02 2.0000e-02 2.0000e-02) / ( 3.5714e-02 3.5714e-02 3.5714e-02)
(Lx, Ly, Lz) [m] / [-] = ( 7.4000e-01 5.6000e-01 5.6000e-01) / ( 1.3214e+00 1.0000e+00 1.0000e+00)
no          Label   ID   i_st   i_ed   j_st   j_ed   k_st   k_ed
  1             Air    4     0     0     0     0     0     0

Domain    1
ix, jx, kx      [-] =           36           28           28
(ox, oy, oz) [m] / [-] = ( 7.6000e-01 2.0000e-02 2.0000e-02) / ( 1.3571e+00 3.5714e-02 3.5714e-02)
(Lx, Ly, Lz) [m] / [-] = ( 7.1999e-01 5.6000e-01 5.6000e-01) / ( 1.2857e+00 1.0000e+00 1.0000e+00)
no          Label   ID   i_st   i_ed   j_st   j_ed   k_st   k_ed
  1             Air    4    14    17     1    28     1    28

-----
Report of Whole Domain Statistics
Domain size          =           73           28           28
Number of voxels     = 1.469440e+05
Number of surface    = 2.457600e+04
Effective voxels      = 1.469440e+05 (100.00\%)
Fluid voxels         = 1.469440e+05 (100.00\%)
Wall voxels          = 0.000000e+00 ( 0.00\%)
Division :           = 2 : Equal segregation

-----
Domain Statistics per MPI process
Mean volume in each domain      = 1.469440e+05
Std. deviation of domain        = 0.000000e+00
Mean comm. in each domain       = 0.000000e+00
Std. deviation of surface       = 0.000000e+00
Mean effective volume in each domain = 1.469440e+05
Std. deviation of effective volume = 0.000000e+00

Domain :  ix  jx  kx  Volume Vol_dv[%] Surface Srf_dv[%] Fluid[%] Solid[%] Eff_Vol Eff_Vol_dv[%]
-----
    0 :  73  82  16  1.4694e+05  0.000  0.00e+00  0.00   100.0   0.0   1.46e+05   0.00
-----
```

9.2.4 基本履歴 [history_base.txt]

標準履歴ファイルは、下記のような履歴情報が出力されます。この履歴情報は選択された時間積分スキームや反復解法の収束判定ノルムの種類などにより出力項目は異なります。下記の計算例では、時間積分に Euler 陽解法を用いた流動解析を行い、圧力 Poisson 方程式の反復解の収束判定のノルムに速度の発散の最大値を用いています。各欄のラベルの説明を表 9.3 に示します。

標準出力と history_base.txt ファイルには同じ内容が出力され、時刻と速度の次元は有次元となっています。収束判定のノルムの種類については、表??を参照のこと。ノルムの次元は慣例的に無次元としています。

step	time[-]	v_max[-]	ItrVP	v_div_max[-]	ItrP	r_r0	deltaP	avrP	deltaV	avrV	time[sec]
1	3.125000e-03	0.00000e+00	1	1.2529e-06	50	1.0524e+04	2.118e-07	2.387e-16	1.050e-04	1.103e-08	1.598821e-01
2	6.250000e-03	6.1685e-07	1	5.0281e-06	50	4.0431e+03	8.023e-07	-2.057e-15	2.130e-04	5.640e-08	1.575129e-01
3	9.375000e-03	3.0604e-06	1	1.1121e-05	50	2.7914e+03	1.736e-06	-1.120e-14	3.252e-04	1.621e-07	1.545382e-01
4	1.250000e-02	8.4922e-06	1	1.9072e-05	50	2.2570e+03	3.012e-06	1.154e-15	4.414e-04	3.570e-07	1.556299e-01
5	1.562500e-02	1.8023e-05	1	2.8457e-05	50	1.9471e+03	4.625e-06	3.991e-14	5.617e-04	6.725e-07	1.569211e-01
6	1.875000e-02	3.2711e-05	1	3.8938e-05	50	1.7347e+03	6.565e-06	1.175e-13	6.859e-04	1.143e-06	1.570611e-01
7	2.187500e-02	5.3559e-05	1	5.0244e-05	50	1.5723e+03	8.807e-06	1.818e-13	8.136e-04	1.805e-06	1.582470e-01

表 9.3 履歴ファイルの出力項目

Label	説明
step	計算ステップ数
time	時刻
v_max	速度の最大値
ItrVP	速度圧力同時緩和反復回数
V_div_Max	反復の収束判定に用いるノルムの種類とその値。上記の場合は速度の発散値の最大値を用いています。指定するノルムの種類により、ヘッダの記述が変わります。
ItrP	圧力ポアソンの反復回数
r_r0	反復の収束判定に用いるノルムの種類とその値。上記の場合は残差を初期残差で割った値を用いています。指定するノルムの種類により、ヘッダの記述が変わります。
deltaP	圧力の 1 ステップの変化量の自乗和平方根 $\sqrt{\sum_{i,j,k}^{Max} \ \delta p\ _2}$
avrP	圧力の平均値
deltaV	速度の 1 ステップの変化量の自乗和平方根 $\sqrt{\sum_{i,j,k}^{Max} \ \delta v\ _2}$
avrV	速度の平均値
time[sec]	このステップの計算に要した時間（秒）

下記は、熱流動計算で速度場対流項に Euler 陽解法、温度場は対流項、拡散項に Euler 陽解法を用いた履歴の出力例を示します。deltaT は温度の 1 ステップの変化量の RMS（自乗和平方根）、avrT は、温度の平均値です。

step	time[sec]	v_max[m/s]	ItrVP	v_div_max[-]	ItrP	dx_b	deltaP	avrP	deltaV	avrV	deltaT	avrT	time[sec]
1	1.666667e-04	0.00000e+00	10	2.8400e-01	200	4.4888e-01	1.620e+01	4.483e+00	2.284e-01	5.218e-02	5.054e-05	8.679e-06	8.976793e-02
2	3.333333e-04	2.6852e+00	10	1.4303e-01	200	9.6965e-01	1.128e+01	8.870e+00	2.139e-01	9.791e-02	5.052e-05	1.736e-05	7.954216e-02
3	5.000000e-04	3.1871e+00	10	1.1606e-01	200	1.2913e+00	9.100e+00	1.327e+01	2.155e-01	1.443e-01	5.049e-05	2.603e-05	7.935309e-02
4	6.666667e-04	3.3308e+00	10	1.0078e-01	200	1.4675e+00	8.189e+00	1.767e+01	2.168e-01	1.914e-01	5.046e-05	3.470e-05	7.974982e-02
5	8.333333e-04	3.3789e+00	10	9.0329e-02	200	1.5740e+00	7.627e+00	2.207e+01	2.170e-01	2.384e-01	5.043e-05	4.337e-05	8.019686e-02
6	1.000000e-03	3.3924e+00	10	8.3161e-02	200	1.6494e+00	7.273e+00	2.649e+01	2.161e-01	2.851e-01	5.038e-05	5.204e-05	8.125401e-02
7	1.166667e-03	3.3917e+00	10	7.9153e-02	200	1.7039e+00	7.056e+00	3.093e+01	2.146e-01	3.311e-01	5.030e-05	6.070e-05	8.071899e-02

9.2.5 コンポーネント履歴 [history_compo.txt]

コンポーネントに関連する履歴を出力します。

step	time[sec]	Q[04]	Q[05]
1	4.000000e-04	-9.60000e+02	0.00000e+00
2	8.000000e-04	-8.1139e+02	1.9200e+02
3	1.200000e-03	-7.1957e+02	2.2157e+02
4	1.600000e-03	-6.4465e+02	2.8151e+02
5	2.000000e-03	-5.9062e+02	3.0965e+02
6	2.400000e-03	-5.4862e+02	3.3736e+02
7	2.800000e-03	-5.1712e+02	3.5581e+02

上記の例は、FFVC/examples/SHC1D です。condition.txt の Component Information を見ると、Heat Transfer type S が設定されている Rod に No.4 が、Isothermal が設定されている HeatedWall に No.5 が設定されていることがわかります。Q[04] は、Rod から流体に熱伝達される熱量 [W] を ll 、Q[05] は、HeatedWall から Rod に熱伝導されている熱量 [W] を毎ステップ記述したものです。

このように、history_compo.txt には、局所境界条件で設定されたコンポーネントに対して、関連する物理量の履歴が示されます。ただし、Class = “Monitor”のコンポーネントについては、MonitorList での設定にしたがい、sampling.txt に出力されます。

表 9.4 コンポーネント履歴ファイルの出力項目

カテゴリー	コンポーネント Class	表示項目
セル界面速度	SpecifiedVelocity	平均速度 V [m/s]
	Outflow	平均速度 V [m/s]
		温度指定の場合、流入熱量 Q [W]
セル界面熱計算	DirectHeatFlux	熱流束 q [W/m ²]
	HeatTransferS	移動熱量 Q [W]
	HeatTransferSF	移動熱量 Q [W]
	HeatTransferSN	移動熱量 Q [W]
	IsoThermal	移動熱量 Q [W]
	Radiation	移動熱量 Q [W]
セル要素熱計算	HeatSource	
	SpecifiedTemperature	

9.2.6 流量収支履歴 [history_domainflux.txt]

計算領域の外部境界における流量と速度の履歴を出力します。Q は断面流量 [m^3/s] を、Balance は計算内部領域への流入出する流量の和を示します。V は有効断面平均速度 [m/s] を表しますが、境界条件で述べるように流出断面を指定している場合には指定値となります。熱計算の場合、H は移動熱量 [W] を示します。

step	time	Q:X-	...	Q:Z+ >>	Balance	V:X-	...	V:Z+
756	1.890020e+00	-7.5980e-02		-1.3623e-01 >>	6.5136e-01	-1.3155e-03		-8.6816e-04
757	1.892520e+00	-7.6318e-02		-1.3660e-01 >>	6.5357e-01	-1.3214e-03		-8.7049e-04
758	1.895020e+00	-7.6656e-02		-1.3696e-01 >>	6.5578e-01	-1.3273e-03		-8.7283e-04

9.2.7 反復履歴 [history_iteration.txt]

圧力 Poisson, 圧力速度の反復履歴を示します。ノルムのタイプに VdivMax (速度の発散値の最大値) を指定している場合には、計算領域内の位置が出力されます。

step= 16	time= 5.000000e-02	Itr_VP	Div_V	Itr_P	Norm (i, j, k)
		1	1.030545e-04	50	2.347707e+03 (0, 0, 0)
		2	7.872764e-05	100	2.198585e+03 (0, 0, 0)
step= 17	time= 5.312500e-02	Itr_VP	Div_V	Itr_P	Norm (i, j, k)
		1	1.014899e-04	50	1.996019e+03 (0, 0, 0)
		2	7.596384e-05	100	1.865845e+03 (0, 0, 0)
step= 18	time= 5.625000e-02	Itr_VP	Div_V	Itr_P	Norm (i, j, k)
		1	1.008246e-04	50	1.698130e+03 (0, 0, 0)
		2	7.406585e-05	100	1.580989e+03 (0, 0, 0)
step= 19	time= 5.937500e-02	Itr_VP	Div_V	Itr_P	Norm (i, j, k)
		1	1.010439e-04	50	1.454337e+03 (0, 0, 0)
		2	7.311767e-05	100	1.346295e+03 (0, 0, 0)
step= 20	time= 6.250000e-02	Itr_VP	Div_V	Itr_P	Norm (i, j, k)
		1	1.022700e-04	50	1.263698e+03 (0, 0, 0)
		2	7.319565e-05	100	1.162129e+03 (0, 0, 0)

9.2.8 サンプル履歴 [sampling.txt]

座標値指定、ポリゴン指定、領域指定によるサンプリング結果を出力します。第7章をご覧ください。

9.2.9 性能情報

実行時のタイミングを測定し、サマリーを表示します。各項目の表示内容を表 9.5 に示します。

Report of Timing Statistics PMLib version 1.9.9									
Operator : Kenji_Ono									
Host name : masako-VirtualBox									
Date : 2013/11/07 : 13:40:28									
Parallel Mode : OpenMP (4 threads)									
Total execution time = 5.292371e+00 [sec]									
Total time of measured sections = 4.636522e+00 [sec]									
Statistics per MPI process [Node Average]									
Label	call	avr[sec]	accumulated time			flop		messages[Bytes]	
			avr[%]	sdv[sec]	avr/call[sec]	avr	sdv	speed	

Poisson_SOR2_SMA	: 27000	2.205365e+00	47.57	0.00000e+00	8.168020e-05	5.972e+09	0.000e+00	2.52	Gflops
Poisson_Src_Norm	: 13700	1.078649e+00	23.26	0.00000e+00	7.873352e-05	8.690e+09	0.000e+00	7.50	Gflops
Poisson_BC	: 27000	9.509439e-01	20.51	0.00000e+00	3.522015e-05	0.000e+00	0.000e+00	0.00	Mflops
Projection_Velocity	: 270	7.469797e-02	1.61	0.00000e+00	2.766592e-04	3.517e+08	0.000e+00	4.38	Gflops
Pseudo_Velocity	: 100	6.269217e-02	1.35	0.00000e+00	6.269217e-04	9.806e+08	0.000e+00	14.57	Gflops
Poisson_Setup_for_Itr	: 13500	5.711293e-02	1.23	0.00000e+00	4.230588e-06	0.000e+00	0.000e+00	0.00	Mflops
Divergence_of_Pvec	: 100	3.436279e-02	0.74	0.00000e+00	3.436279e-04	4.547e+07	0.000e+00	1.23	Gflops
Velocity_BC	: 270	2.522254e-02	0.54	0.00000e+00	9.341681e-05	0.000e+00	0.000e+00	0.00	Mflops
Force_Calculation	: 100	2.122140e-02	0.46	0.00000e+00	2.122140e-04	0.000e+00	0.000e+00	0.00	Mflops
Projection_Velocity_BC	: 270	1.926827e-02	0.42	0.00000e+00	7.136398e-05	0.000e+00	0.000e+00	0.00	Mflops
Poisson_Src_VBC	: 100	1.369810e-02	0.30	0.00000e+00	1.369810e-04	0.000e+00	0.000e+00	0.00	Mflops
History_Stdout	: 100	1.291752e-02	0.28	0.00000e+00	1.291752e-04	0.000e+00	0.000e+00	0.00	Mflops
Copy_Array	: 200	1.223564e-02	0.26	0.00000e+00	6.117821e-05	0.000e+00	0.000e+00	0.00	Mflops
History_Iteration	: 370	1.187801e-02	0.26	0.00000e+00	3.210274e-05	0.000e+00	0.000e+00	0.00	Mflops
Variation_Space	: 100	1.157522e-02	0.25	0.00000e+00	1.157522e-04	7.987e+07	0.000e+00	6.43	Gflops
Poisson_Norm_Div_max	: 270	1.019073e-02	0.22	0.00000e+00	3.774343e-05	1.991e+07	0.000e+00	1.82	Gflops
Search_Vmax	: 100	8.419275e-03	0.18	0.00000e+00	8.419275e-05	1.106e+07	0.000e+00	1.22	Gflops
Pseudo_Vel_Flux_BC	: 100	5.115986e-03	0.11	0.00000e+00	5.115986e-05	4.608e+05	0.000e+00	85.90	Mflops
Pseudo_Velocity_BC	: 100	4.827976e-03	0.10	0.00000e+00	4.827976e-05	0.000e+00	0.000e+00	0.00	Mflops
History_Base	: 100	3.802538e-03	0.08	0.00000e+00	3.802538e-05	0.000e+00	0.000e+00	0.00	Mflops
File_Output	: 2	3.530025e-03	0.08	0.00000e+00	1.765013e-03	2.458e+05	0.000e+00	66.39	Mflops
Pvec_Euler_Explicit	: 100	2.552986e-03	0.06	0.00000e+00	2.552986e-05	9.830e+06	0.000e+00	3.59	Gflops
assign_Const_to_Array	: 100	2.134085e-03	0.05	0.00000e+00	2.134085e-05	0.000e+00	0.000e+00	0.00	Mflops
History_Force	: 100	1.908779e-03	0.04	0.00000e+00	1.908779e-05	0.000e+00	0.000e+00	0.00	Mflops
Allocate_Arrays	: 4	1.703024e-03	0.04	0.00000e+00	4.257560e-04	0.000e+00	0.000e+00	0.00	Mflops
Domain_Monitor	: 100	4.866123e-04	0.01	0.00000e+00	4.866123e-06	0.000e+00	0.000e+00	0.00	Mflops
Restart_Process	: 1	8.821487e-06	0.00	0.00000e+00	8.821487e-06	0.000e+00	0.000e+00	0.00	Mflops

Total			4.636522e+00			1.616e+10		3.25	Gflops

表 9.5 タイミングレポートの表示内容

項目	内容
Poisson_SOR2_SMA	2-colored SOR 法の Poisson の反復計算部分のみ
Poisson_Src_Norm	Poisson 方程式のソース項のノルム計算
Poisson_BC	圧力の境界条件
Projection_Velocity	速度の射影
Poisson_Setup_for_Itr	反復処理の準備
Divergence_of_Pvec	擬似速度ベクトルの発散
Velocity_BC	速度の境界条件
Force_Calculation	力の計算
Projection_Velocity_BC	射影速度の境界条件
Poisson_Src_VBC	速度境界条件による圧力ソース項の修正
History_Stdout	履歴の標準出力
Variation_Space	空間変動量の計算
Poisson_Norm_Div_max	ノルムの計算（速度の発散値の最大値）
Search_Vmax	速度の最大値の計算
Pseudo_Vel_Flux_BC	擬似速度流束の境界条件
Pseudo_Velocity_BC	擬似速度の境界条件
History_Base	基本履歴
File_Output	ファイル出力
Pvec_Euler_Explicit	擬似速度の時間積分
assign_Const_to_Array	配列の値代入
History_Force	力の履歴出力
Allocate_Arrays	配列確保
Domain_Monitor	計算領域の流量モニターの計算
Restart_Process	リスタート処理

9.2.10 ボクセルファイル

組み込み例題の場合には、内部で生成されたボクセルファイルが `example.svx` として書き出されます。

9.2.11 結果ファイル

計算結果は、デフォルトで `sph` ファイルフォーマット出力で書き出されます。これらは、V-Isio で可視化できます。

9.2.12 メモリ使用量の情報

FFV-C ソルバーでは、実行中において必要なときに必要な量だけメモリを使用する方針です。このため、プリプロセスとメインループ（計算部分実行中）でメモリ使用量は異なります。プログラム起動中に必要な最大メモリ量を、Polygon 読み込み、格子生成、プリプロセス、ソルバー（計算部分実行中）ごとに、`condition.txt` 中表示します。

9.3 並列計算

9.3.1 MPI 並列

本節では、OpenMPI 通信ライブラリを用いたプロセス並列の実行について説明します。並列計算の実行は、`mpirun` コマンドで起動します。

```
$ mpirun -np 2 ffvc hoge hoge.tp
```

9.3.2 スレッド並列

本節では、共有メモリでのスレッド並列の実行について説明します。実行時の環境設定として、環境変数 `OMP_NUM_THREADS` にスレッド数を設定します。次の例では、`bash` で4スレッドを指定しています。

```
$ export OMP_NUM_THREADS=4
```

この後、逐次、並列実行をコマンドラインで指示します。

```
Serial  
$ ffvc hoge hoge.tp  
  
MPI  
$ mpirun -np 8 ffvc hoge hoge.tp
```

上記で並列実行の場合には、8 プロセス ×4 スレッドのハイブリッド実行となります。

9.4 各プラットフォームにおける実行

9.4.1 RICC

バッチジョブのスクリプトファイル例を示します。

```
#!/bin/sh

#----- qsub option
#MJS: -mpc
#MJS: -proc 64
#MJS: -thread 1
#MJS: -mem 1200mb
#MJS: -time 24:00:00
#MJS: -eo
#MJS: -rerun Y
#MJS: -cwd
#MJS: -parallel openmpi

#----- FTLcommand
#FTLDIR: $MJS_CWD
#FTL_SUFFIX: off
#FTL_RANK_FORMAT: 3
#FTL_NO_RANK: off
#
#<BEFORE>
#ALL: sphere_f
#0: P32E_resize_5mm_CutWS.svx
#</BEFORE>
#
#<BEFORE_R>
#ALL: xml
#</BEFORE_R>
#
#<AFTER>
#0: *.sph, *.txt, *.log
#</AFTER>

#----- Program execution
mpirun ./ffvc hoge.jp
```

次に、利用頻度の高いコマンド類を示します。

■ ジョブ投入

```
$ qsub go.sh
```

■ ジョブ状態表示

```
$ qstat -m 使用メモリ量  
$ qstat -p プライオリティ  
$ qstat -w 実行待ち理由の表示
```

■ 実行中 Job の標準出力表示

```
$ qcat REQID
```

■ Job 優先度変更

```
$ qalter -p <PRIORITY> <REQID>
```

■ mpc の計算ノード上のファイル一覧 OPTION は ls コマンドと同じである。

```
$ qls REQID[@RankID] [OPTION]
```

■ mpc の計算ノード上のファイルを取得します。

```
$ qget REQID[@RankID] file
```

9.4.2 京

バッチジョブのスクリプトファイル例を示します。

```
#!/bin/bash -x

#----- pjsub option
#!/bin/sh -x
#PJM --rsc-list "node=8"
#PJM --rsc-list "elapsed=01:00:00"
#PJM --mpi "shape=8"
#PJM --stg-transfiles all
#PJM --mpi "use-rankdir"
#PJM --stgin "rank=* ./a.out %r:./"
#PJM --stgin "rank=* ./input.dat %r:./"
#PJM --stgout "rank=* ./result.txt%r ./result.%r.txt"
#PJM -s
#
. /work/system/Env_base
export OMP_NUM_THREADS=8

#----- Program execution
mpiexec ./a.out
```

次に、利用頻度の高いコマンド類を示します。

■ ジョブ投入

```
$ pjsub job.sh
```

■ ジョブ状態表示

```
$ pjstat      ジョブ状態表示
$ pjstat -H   ヒストリー表示
$ pjstat -w   実行待ち状態の表示
```

■ ジョブ削除

```
$ pjdel JOB_ID
```

■ 実行中 Job の標準出力表示

```
$ pjcat JOB_ID
```


第 10 章

アップデート情報

本ユーザガイドのアップデート情報について記します。

- Version 0.8.5 2015/7/14
- New AICS logo
 - 解析モデル
- Version 0.8.5 2015/7/13
- ffvc version 2.3.9 に合わせてアップデート
- Version 0.8.4 2013/11/25
- モニタリング機能の加筆
 - リスタートの修正
 - ソルバーの実行を修正
- Version 0.8.3 2013/10/13
- SpecifiedVelocity > FluidDirection の加筆
 - PLOT3D オプションの記述改訂
 - /Output/Log/CCNVfile
 - ファイル管理とリスタートを加筆
- Version 0.8.2 2013/9/15
- GeometryModel/Output のパラメータを追記
- Version 0.8.1 2013/9/12
- 1, 2, 3, 5, 6 章のアップデート
- Version 0.8.0 2013/8/22
- パラメータの再構成
- Version 0.7.1 2013/7/12
- インストール方法改訂
 - パラメータ記述構造の改訂
/Steer/Iteration, /Steer/StartCondition, /OuterBoundary, /MediumTable
- Version 0.7.0 2012/10/18
- ファイル出力の指定様式変更
 - 反復解法の指定様式変更
 - スタート条件の指定様式変更
- Version 0.6.0 2013/6/3
- 東大と理研のライセンスを追記.
 - リスタートの章を追加.
- Version 0.5.2 2012/8/23
- 粗格子を用いたリスタートについて追記.
- Version 0.5.0 2012/7/14
- 0.5.0 プレリリース.

参考文献

- [1] C.W Hirt and B.D Nichols. Adding limited compressibility to incompressible hydrocodes. *Journal of Computational Physics*, Vol. 34, No. 3, pp. 390 – 400, 1980.
- [2] 中山顕, 桑原不二郎, 許国良. 熱流体力学 -基礎から数値シミュレーションまで-. 共立出版, 2002.
- [3] C. W. Hirt, B. D. Nichols, and N. C. Romero. SOLA: a numerical solution algorithm for transient fluid flows. 1975.
- [4] K. Maruyama, D. Toriu, and S. Ushijima. Parallel computation for fluid-structure thermal interactions with multiphase modeling. In *Proc. 10th WCCM/APCOM*, No. 19797, 2012.
- [5] 小林俊雄（編）. CFD ハンドブック, 第 10.4 章, pp. 538–545. 丸善, 東京, 2003. 格子形成法 (10.4).
- [6] M. Tanaka and S. Kida. Characterization of vortex tubes and sheets. *Physics of Fluids A*, Vol. 5, pp. 2079–2082, 1993.
- [7] 庄司正弘. 伝熱工学. 東京大学出版会, 1995.

索引

イニシャルスタート	55
インデクス	61
温度拡散係数	13
解析事例データベース	34
ガイドセル	31
拡散数	55
基本リスト	
境界条件の—	63
境界条件リストの名前	63
組み込み例題	34, 37
計算工学ナビ	34
座標系	50
移動—	50
静止—	50
時間積分幅	55
実行制御パラメータ	30
初期値	53
ステンシル	31
スレッド並列	125
制限関数	31
セッション	110
単振動	65, 82
定常解	56
低マッハ数	13
熱流動タイプ	36
ノード	
Text Parser の—	29
反復解法	38
ファイル	
コンポーネント履歴—	121
反復履歴—	122
標準履歴—	120
ボクセル—	124
流量履歴—	122
履歴—	44
condition—	118
sph—	124
Boussinesq 近似	12
プロセス並列	125
分離解法	16
並列計算	125
変数配置	
Collocated—	61
密度変化	12
無次元化	14

リーフ	29
Text Parser の—	29
履歴	120
Acceleration	55
ALE	12
Alias	39
Application Control	30
Base	44
Basic Variables	46
Binary Voxel	21
Boundary	62, 63
Boussinesq	12
Buoyancy	36
CFL	55
Check Parameter	30
Class	39, 63
Console	45
Convection Term	31
Data	45
Directory Path	46
Divergence	47
Div Criterion	40
Div Max Iteration	40
Div Norm	40
Domain Info	32
Dry Run BC	30
Error Norm	40
Fill Hint	33
Flow Equation	36
Format	45
Format Option	47
Fractional step	16
Geometry Model	34
Governing Equation	36
Heat Equation	36
Helicity	47
History	45
Initial Perturbation	58
Intrinsic Example	37
Iteration	38
Kind	63
Log	44
Max Iteration	39
Medium	63
Medium Table	41
Model	58
Monitor List	42
mpirun	125
Operator	30
OuterBC	62
OuterFace	33

Outer BC	64
Output	35, 44
Output Glyph	35
PDE Type	36
Point	33
Qcriterion	47
Reference	49
Reference Frame	50
Residual Criterion	39
Residual Norm	39
Session	56
Shape Approximation	51
Solving Method	52
Source	34
Start Condition	53
Statistic	56
Statistical Variables	47
Text Parser	29
Thermal Option	66
Time Control	55
Time Slice	46
Time Step	55
Time Variation	36
Total Pressure	46
Turbulence Modeling	58
Unit	59
Variable Range	30
Velocity Profile	58
Vorticity	46
Voxel Output	34