

文部科学省次世代IT基盤構築のための研究開発  
「革新的シミュレーションソフトウェアの研究開発」

RSS21 フリーソフトウェア

マルチフィジックス流体シミュレーション

FrontFlow/red ver 3.1

## 使用手順説明

本ソフトウェアは文部科学省次世代IT基盤構築のための研究開発「革新的シミュレーションソフトウェアの研究開発」プロジェクトによる成果物です。本ソフトウェアを無償でご使用になる場合「RSS21フリーソフトウェア使用許諾条件」をご了承頂くことが前提となります。営利目的の場合には別途契約の締結が必要です。これらの契約で明示されていない事項に関して、或いは、これらの契約が存在しない状況においては、本ソフトウェアは著作権法など、関係法令により、保護されています。

お問い合わせ先

(公開／契約窓口) (財)生産技術研究奨励会

〒153-8505 東京都目黒区駒場4-6-1

(ソフトウェア管理元) 東京大学生産技術研究所 計算科学技術連携研究センター

〒153-8505 東京都目黒区駒場4-6-1

Fax : 03-5452-6662

E-mail : software@rss21.iis.u-tokyo.ac.jp

## 目次

1. 概要 .....	1
1.1. 実行モジュール .....	1
1.2. 手順の概要 .....	1
2. FrontFlow/red のための前処理 : prefflow の実行 .....	2
2.1. prefflow に関連するファイル群 .....	2
2.2. 事前時準備するディレクトリ及びファイル .....	3
2.3. prefflow の実行 .....	4
2.4. prefflow のエラーメッセージと計算制御ファイルの設定修正 .....	6
2.5. メインソルバに必要なファイル群 .....	9
3. メインソルバの実行 : fflowS, fflowHPC の実行 .....	10
3.1. メインソルバの実行 .....	10
3.2. 計算を途中で止める .....	14
3.3. 計算の途中経過を出力する .....	15
3.4. 計算をリスタートする .....	15
4. ユーザサブルーチンの使用法 .....	17
5. 後処理のためのコンバート .....	18
5.1 解析結果のコンバート : ffr2viz の実行 .....	18
5.2 粒子追跡データのコンバート : ffr2vizp の実行 .....	20

## 1. 概要

### 1.1. 実行モジュール

FrontFlow/red は，以下の実行モジュールで構成されます．

Table 1 FrontFlow/red の実行モジュール

prefflow	格子のチェック FrontFlow/red 用のコントロールボリュームの構築 設定ファイルのチェック コントロールボリュームから壁面までの距離計算等
fflowS	単一 CPU 用のメインソルバ
fflowHPC	並列 CPU 用のメインソルバ
ffr2viz ffr2vizp	可視化ソフトウェアで読み込むための計算結果データの コンバート

ここに，一般的な流体解析に必要な，格子生成ソフトウェア並びに可視化ソフトウェアは含まれておりません．

### 1.2. 手順の概要

FrontFlow/red を用いた流体解析一連の流れは以下ようになります．

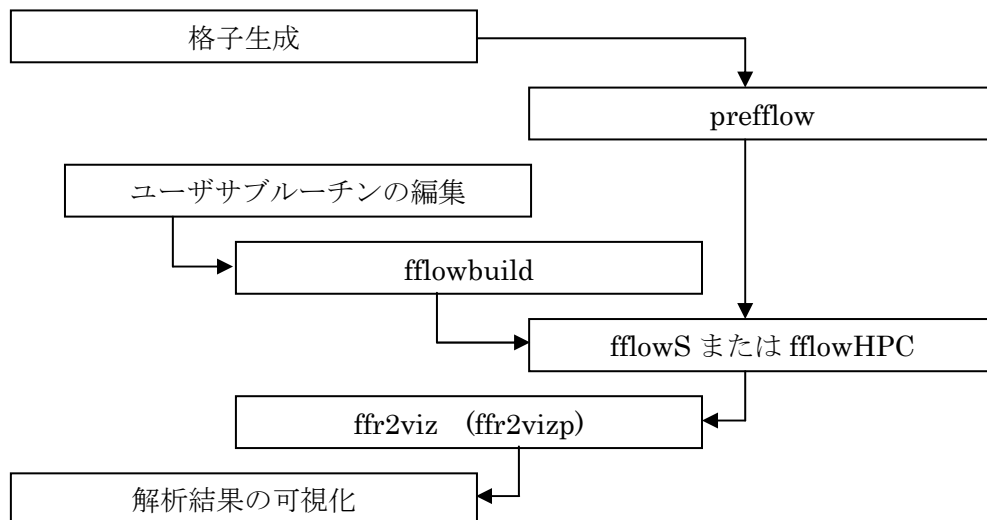


Fig.1 FrontFlow/red を用いた流体解析一連の流れ

## 2. FrontFlow/red のための前処理：prefflow の実行

### 2.1. prefflow に関連するファイル群

prefflow は、メインソルバで読み込むための格子情報などを作成する実行モジュールです。用意した格子データのチェック、節点情報やコネクティビティの確認から、fflowS や fflowHPC で用いるコントロールボリューム(以下、CV と称します)の作成、CV から壁面までの距離の計算などを行います。また並列計算の際には、prefflow に内蔵された METIS<sup>1</sup>によって領域分割を行います。

prefflow に関するファイルは、Table 2 及び Table 3 の通りです。

Table2 単一 CPU 解析(fflowS)に必要なファイル群

ファイル名 (例)	ファイルの名称	内容	I/O	A/B	備考
fflowctl	計算制御ファイル	計算制御のための諸パラメータ	I	A	○
grid.grd*	格子ファイル	格子データ	I	A	○
dimen.h_serial	配列のディメンジョン ファイル	配列の大きさを指定	O	A	-
distance	壁面距離ファイル	壁面からの距離を保存	O	B	-
geom	ジオメトリファイル	fflowS 用形状データ	O	B	-
geom.frontflow	ジオメトリ変換ファイル	可視化用格子データ	O	B	-
MAT_CVG	マテリアルファイル	マテリアル(流体属性)情報 保存データ	O	B	-

○fflowctl は固定名です。他のファイル名を用いることはできません。

○ファイル名やファイル数は格子データのフォーマットに依ります。

Table3 並列 CPU 解析(fflowHPC)に必要なファイル群

ファイル名 (例)	ファイルの名称	内容	I/O	A/B	備考
fflowctl	計算制御ファイル	計算制御のための諸パラメータ	I	A	○
grid.grd*	格子ファイル	格子データ	I	A	○
dimen.h_serial	配列のディメンジョン ファイル	配列の大きさを指定	O	A	-
dimen.parm	配列のディメンジョン ファイル	配列の大きさを指定(並列計算用)	O	A	†
dimen.h_hpc	配列のディメンジョン	配列の大きさを指定(並列計	O	A	-

<sup>1</sup> METIS : “A Fast and Highly Quality Multilevel Scheme for Partitioning Irregular Graphs”. George Karypis and Vipin Kumar. SIAM Journal on Scientific Computing, Vol. 20, No. 1, pp. 359—392, 1999.

	ファイル	算用)			
distance	壁面距離ファイル	壁面からの距離を保存	O	B	-
geom	ジオメトリファイル	FrontFlow/Red 用形状データ	O	B	-
geom.frontflow	ジオメトリ変換ファイル	可視化用格子データ	O	B	-
MAT_CVG	マテリアルファイル	マテリアル情報保存データ	O	B	-
test.m.part.xx	並列計算用一時ファイル	-	O	B	-
Bound_hpc	並列計算用境界ファイル	-	O	B	†
Grid_hpc	並列計算用格子ファイル	-	O	B	†
reod	並列計算用一時ファイル	-	O	B	†
wall_hpc	並列計算用一時ファイル	-	O	B	†
geom_hpc	並列計算用一時ファイル	-	O	B	†
work	並列計算用一時ファイル	-	O	B	†

○fflowctl は固定名です。他のファイル名を用いることはできません。

○ファイル名やファイル数は格子データのフォーマットに依ります。

† 事前に用意したサブディレクトリ内に、分割数の数だけ生成されます。

## 2.2. 事前時準備するディレクトリ及びファイル

計算は通常、作業ディレクトリを作成し、作業ディレクトリ内において実行します。作業ディレクトリは任意の場所に設定できますが、パスに全角文字や半角スペースが含まないようにしてください。ここでは「work」というディレクトリで作業するものとします。

まず、計算実行に必要なファイルを作業ディレクトリに用意します。計算をはじめるためには、作業ディレクトリに最低限以下のファイルが必要です。

work/	└	fflowctl	計算制御ファイル
	└	grid.grd	格子データ（ファイル名は計算制御ファイルで決定）

並列計算を行う際には、事前に分割数に応じて設定ファイルで指定するサブディレクトリを用意しておく必要があります。デフォルトのサブディレクトリ名は”hpc”です。例えば 16 並列で計算する場合、『hpc\_0000, hpc\_0001~hpc\_0015』の 16 個のディレクトリを用意します。

work/	└	fflowctl	計算制御ファイル
	└	grid.grd	格子データ（ファイル名は計算制御ファイルで決定）
	└	hpc_0000/	
	└	...	
	└	hpc_xxxx/	並列計算用サブディレクトリ（xxxx は CPU 番号）*

格子の作成、設定ファイルの作成は、別紙にて説明いたします。

### 2.3. prefflow の実行

はじめに、作業ディレクトリが並列計算に使用するすべての端末から全く同じパスで見えていることを確認してください。基本的には、作業ディレクトリは NFS などネットワーク共有されたボリューム上に配置してください。次に、作業ディレクトリでプリプロセッサ **prefflow** を実行します。

```
% cd work [Enter ｷｰ]
% prefflow [Enter ｷｰ] *
```

\* インストールの際に環境変数 `PATH` を設定しなかった場合は、下記のように `prefflow` プログラムを `FrontFlow/Red` インストールディレクトリから作業ディレクトリにコピーして実行してください。なお、ここではインストールディレクトリが `[/home/user_id/FrontFlow Red/]` であった場合を仮定しています。

```
% cp /home/user_id/FrontFlow_Red/bin/prefflow ./ [Enter キー]
% ./prefflow [Enter キー]
```

**prefflow** を起動すると、以下のようなメッセージが表示されます(表示される内容は、お使いの **FrontFlow/Red** のバージョン、および計算制御ファイルの設定内容によって異なることがあります)。

```
=====|
|                                     |Welcome to FrontFlow|                                     |
|=====|

#####
###      Scalar Information          #####
###                                           #####
####      Scalar No.           Scalar Name    #####
|-----|

number of chemical species :   1
1 air      mi = 0.29000E-01 [kg/mol]   Ri = 0.28669E+03 [J/(K kg)]
            cp = 0.10040E+04  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00

|=====|
|                                     |LIST SERIAL CPU MESH|                                     |
|=====|

###      Initialing... Please wait.
###      Reading FFR Grid element type field...
###      Reading FFR Grid boundary type field...
###      Reading FFR Grid vertices coordinate field...
###      Reading FFR Grid Boundary Face field...
        .
        -   (以下略)
        .
```

```

prefflow の実行には、計算条件によって数分～数時間かかることがあります。

```

正常に動作が終了すると、以下のようなメッセージが表示されます。また、作業ディレクトリ内に **dimen.h\_serial**、**dimen.h\_hpc**（インクルードファイル）などが生成されます。またサブディレクトリ **hpc\_xxxx** 内にも **dimen.parm** ファイルや分割格子データ等のファイルが生成されます。

```
.
. (表示の続き)
.

=====|
###   hpc_0000/dimen.parm file has been created successfully
=====|
###   hpc_0001/dimen.parm file has been created successfully
=====|
###   hpc_0002/dimen.parm file has been created successfully
=====|
###   hpc_0003/dimen.parm file has been created successfully
=====|
###   REORDERING COMM FILE ...
%%%   RUN TIME (S) :    0.000000E+00
%%%   Pre-Exec  terminated normally
=====|

*****
METIS 4.0.1 Copyright 1998, Regents of the University of Minnesota

Graph Information -----
Name: test.m, #Vertices: 214853, #Edges: 1466885, #Parts: 4

K-way Partitioning... -----
4-way Edge-Cut:    22469, Balance:    1.01

Timing Information -----
I/O:                1.037
Partitioning:        0.796    (KMETIS time)
Total:               1.835
*****
```

計算制御ファイルの設定に何か問題がある場合は、エラーメッセージを表示して途中で停止します。この場合は、計算制御ファイルに必要な修正を行い、**prefflow** を再実行してください。エラーメッセージ及び修正については次節の説明をご覧ください。なお、**prefflow** を再実行したときに、次のようなメッセージが表示されて入力待ち状態になることがあります。

```
.
. (表示の続き)
.

### ICALL=          2 At SF=      15318 -0.539797687845408
### SF between ICVA=    2310 and ICVB=    4124
### LIST PAIR .....
### TOUCH-INLET BC FACE NUMBER :          0
### READ_INITIAL...
### READ_SOURCE...
### list_wall_distance...
#### Wall Distance File existed,
Will you calculate again and over write it ? (n/y)
```

**prefflow** は各格子点から最近接壁面までの距離を計算して、壁面距離ファイルに書き出し

ます。作業ディレクトリ内に壁面距離ファイルが存在する場合は、壁面距離を再計算するか、既存のファイルのデータを使用するか選択することができます。再計算する場合には「y」を入力します。また、すでに壁面距離が正しく計算されている場合には、「n」を入力して壁面距離計算をスキップすることができます。

## 2.4. prefflow のエラーメッセージと計算制御ファイルの設定修正

本節では計算制御ファイルの設定が正しくないときに表示される **prefflow** のエラーメッセージと修正方法の例を説明します。**prefflow** が正常に終了するまで、計算制御ファイルの修正と **prefflow** の再実行を繰り返してください。

### \* 節点数の設定エラー

**prefflow** 実行時には、計算格子に含まれる格子節点数を計算制御ファイルの **sizes** 変数群に記述する必要があります。設定された節点数が格子データと一致してしない場合、次のようなメッセージが表示されます（格子データのフォーマットによっては、文面が少し異なることがあります）。

```
.  
.   
.   
  
(pre_read_griddata)Reading field: Nodes  
(pre_read_griddata)Number of vertex is not matched.      43002      1  
nvrtx=      43002 => Reset [nvrtx] in fflow.ctf file  
(pre_main)  
(pre_FRONTFLOW)  
###   EXECUTION TERMINATED ABNORMALLY  
9999
```

この場合、計算制御ファイルの **sizes** 変数群の **nvrtx** の値を 43002 に修正し、**prefflow** を再実行してください。格子生成ソフトウェアやフォーマットによって節点の数え方に差異がある場合があります。

### \* 要素数の設定エラー

**prefflow** 実行時には、計算格子に含まれる要素数を計算制御ファイルの **sizes** 変数群に記述する必要があります。設定された要素数が格子データと一致してしない場合、次のようなメッセージが表示されます（格子データのフォーマットによっては、文面が少し異なることがあります）。

```
.  
.   
.
```



```

      .
      .
      .
(pre_read_griddata)Number of cell is not matched.      222412      1
ncell=      222412 => Reset [ncell] in fflow.ctl file
(pre_main)
(pre_FRONTFLOW)
###   EXECUTION TERMINATED ABNORMALLY
9999

```

この場合、計算制御ファイルの **sizes** 変数群の **ncell** の値を 222412 に修正し、**prefflow** を再実行してください。

### \* 内部配列サイズの不足エラー

**prefflow** 実行時には、計算制御ファイルの設定にしたがってメモリ上にデータ配列が確保されます。この配列サイズは **sizes** 変数群の **m** から始まる変数 (**mcell**, **mvrtx** など) で設定します。これらの値は格子データの規模にあわせて適宜増減する必要があります（大きい値を設定すれば、大規模の格子データを読み込むことができますが、**prefflow** の使用メモリサイズは増大し実行パフォーマンスが低下します）。配列サイズが不足した場合は以下のようなメッセージが表示されます。

```

      .
      .
      .
### error : data error
mcell =      200000
it must be >=      432473
(nml_sizes)
(pre_namelist_admin)
at pre_namelist_admin

```

この場合、計算制御ファイルの **sizes** 変数群の **mcell** 変数の値を 432473 以上に修正し、**prefflow** を再実行してください。

なお、**sizes** 変数群の配列データサイズ変数 (**m** で始まる変数) の設定値は **prefflow** の実行時に使用されるだけです。メインソルバ実行時には、**prefflow** が見積もった最適値サイズで配列が確保されます。したがって、これら配列データサイズ変数 (**m** で始まる変数) を必要最小値よりも大きい値に設定してもメインソルバ実行時のメモリ使用量や演算パフォーマンスには影響ありません。

## \* 境界条件の設定エラー

prefflow 実行時には、計算格子に含まれる境界領域すべてについて計算制御ファイルの boundary 変数群に記述する必要があります。設定された境界条件が格子データと一致していない場合、次のようなメッセージが表示されます（格子データのフォーマットによっては、文面が少し異なることがあります）。

エラーメッセージ(1) 境界条件設定の不足
<pre>. . .  (pre_read_griddata)Boundary name: [velocity-inlet] is NOT defined in fflow.ctl file. *** please check "boundary" in fflow.ctl  1.miss type or missing in fflow.ctl 2.wrong charactor codes(CR&lt;-&gt;CR+LF etc. wrong FTP mode(ASCII/Binary) causes this error) 3.wrong file, etc.  (pre_main) (pre_FRONTFLOW)  ### EXECUTION TERMINATED ABNORMALLY  9999</pre>

このメッセージは、格子データに含まれている「velocity-inlet」境界領域に対する境界条件設定が計算制御ファイルに記述されていないことを示します。この場合には、計算制御ファイルに「velocity-inlet」境界領域に対する boundary 変数群を追加してください。

- 1) boundary変数群は格子データに含まれる境界領域ごとに記述する必要があります。
- 2) boundary変数群のname変数には、格子データファイル内で定義されている境界領域名を正しく記述してください。大文字小文字も区別されます。
- 3) boundary変数群のkind変数には、FrontFlow/Redで決められている文字列を正しく入力してください。大文字小文字も区別されます。kind変数に与えることができる文字列についてはboundary変数群の説明をご覧ください。
- 4) boundary変数群が設定されているのにエラーが表示される場合は、改行コードが正しくない可能性があります。ftp等を使用して計算制御ファイルを転送する時には、ASCIIモードで転送してください。

エラーメッセージ(2) 境界条件変数の設定値エラー
<pre>. . .</pre>

```
### error : data error  
sum of ys(i-j) = 0.000000000000000E+000  
BC no. = 6  
BC kind :wall  
it must be > 0  
i,j = 1 1  
(module_boundary) (inputdata)  
(pre_namelist_admin)  
at pre_namelist_admin
```

このメッセージは、境界領域番号 6 の境界条件として設定された配列変数 **ys** の合計値が正しくないことを示します。この場合は境界領域 6 の変数 **ys** の設定値を修正してください。

## 2.5. メインソルバに必要なファイル群

大型計算機を使用して大規模並列計算をする場合、手元のコンピュータで **prefflow** を実行してメインソルバのみを大型計算で実行することが想定されます。この時は、大型計算機に必要なファイルをアップデートする必要がありますが、**prefflow** に関連するファイル全てがメインソルバで必要になるわけではありません。アップデートの時間節約のために、メインソルバに最低限必要なファイル群を示します。

並列計算用ソルバ **fflowHPC** に必要なファイルは以下の通りです。

- **MAT\_CVG**
- **dimen.h\_serial**
- **fflowctl**
- **hpc\_xxxx** ディレクトリ及びその中身全て

なお、可視化用コンバータ **ffr2viz** を実行する際には、

- **geom.frontflow**
- **test.m.part.X** (X は並列数)

が必要になります。

### 3. メインソルバの実行：fflowS, fflowHPC の実行

#### 3.1. メインソルバの実行

##### \* fflowS

単一CPU用ソルバである fflowS の実行は、基本的に作業ディレクトリに移動し、prefflow を実行したディレクトリ内で fflowS を実行するだけです。

カレントディレクトリを作業ディレクトリに変更し、fflowS を起動します。

```
% fflowS [Enter キー] *
```

\* インストールの際に環境変数 PATH を設定しなかった場合（使用説明書へインストール編へ参照）は、下記のように fflowS プログラムを FrontFlow/Red インストールディレクトリから作業ディレクトリにコピーして実行してください。なお、ここではインストールディレクトリが「/home/user\_id/FrontFlow\_Red/」であった場合を仮定しています。

```
% cp /home/user_id/FrontFlow_Red/bin/fflowS ./ [Enter キー]
% ./fflowS [Enter キー]
```

メインソルバ実行中は、計算ログが標準出力に出力されます。

```
#####
####                                     ####
####                                     WELCOME TO FrontFlow/red               ####
####                                     FrontFlow/red Version 2.0               ####
####                                     Year:2004 Mon :07 Date:15                ####
####                                     Hour:0010 Min :07 Sec :12                 ####
####                                     #####
#####
Domain number      : 1
transport model    : constant
viscosity [Pa s]   : 0.182E-04 (mu0)
Schmidt number    : 1.000 (Sc)
Prandtl number     : 0.720 (Pr)
viscosity          : m = mu0
conductivity       : l = Cp*m/Pr (Cp:isopiestic specific heat)
diffusivity        : D = m/(r*Sc) (r:mass density)
-----
number of chemical species : 1
1 air      mi = 0.29000E-01 [kg/mol] Ri = 0.28669E+03 [J/(K kg)]
           cp = 0.10040E+04 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 Formation enthalpy :
0.00000E+00
#### Wall Distance File existed.
Will you calculate again and over write it ? (n/y)
```

prefflow と同様、壁面距離を再計算するか、既存のファイルのデータを使用するかを選択を入力します。通常は prefflow で壁面距離が正しく計算されているので、「n」を入力して壁面距離計算をスキップします。再計算する場合には「y」を入力します。

```
n [Enter キー]
```

計算が正常に進行している間は、下記のようにモニター値が各時間ステップで出力されます。

TIME STEP :	6							
TIME :	0.6000E+00	TIME INTERVAL:	0.1000E+00	MAX. COURANT :	0.5760E+00	AT VERTEX :	289	
	U	V	W	P	T	K	EPS	COMP_1
MONI :	-.8981E+00	0.2698E+00	-.7791E+00	0.1000E+02	0.3007E+03	0.0000E+00	0.0000E+00	0.1000E+01
MIN :	-.6090E+01	-.7673E+01	-.1350E+02	-.9018E+01	0.2270E+03	0.1000E+26	0.1000E+26	0.1000E+01
MAX :	0.5570E+01	0.8503E+01	0.1107E+02	0.7349E+02	0.5416E+03	-.1000E+26	-.1000E+26	0.1000E+01
OUTER-ITER NO. :	2							
INNER-ITER :								
ITER :	2	2	2	14	2	0	0	0
RES :	0.1000E-05	0.1000E-05	0.1000E-05	0.1000E-05	0.1000E-05	0.0000E+00	0.0000E+00	0.1000E+01
ABS :	0.1000E-05	0.1000E-05	0.1000E-05	0.1000E-05	0.5802E-02	0.0000E+00	0.0000E+00	0.1000E-05
OUTER-ITER ERROR :								
ERR :	0.7360E-01	0.9620E-01	0.1416E+00	0.1360E+00	0.4005E-01	0.0000E+00	0.0000E+00	0.0000E+00

## 参考

計算ログをファイルに残したい場合は、下記のようにリダイレクトを行ってください。

```
% fflows >& log.txt [Enter キー]
```

```
n [Enter キー]
```

標準出力、標準エラー出力はファイル `log.txt` に保存され、画面には何も表示されなくなります。通常、プログラムは前述の壁面距離計算の選択入力待ちになりますので、起動後数秒おいて `n` を入力してください。続いてプログラムをバックグラウンドに移します。

```
[Ctrl-z キー]
```

```
% bg [Enter キー]
```

`Control+z` キーを押すとプログラムは一時停止(`suspended`)し、プロンプトが戻ります。

コマンド `bg` により、プログラムをバックグラウンドで動作を再開します。

前述の「`n`」入力が処理されていないと、下記メッセージが表示され入力待ちで一時停止します。この場合には、コマンド `fg` によりプログラムを一度フォアグラウンドに戻し、再度「`n`」を入力してください。

```
[1] + Suspended (tty input) fflows (←入力待ちで一時停止した)
```

```
% fg
```

```
n [Enter キー]
```

```
[Ctrl-z キー]
```

```
% bg [Enter キー]
```

`tail` コマンドを使用すると、書き出し中のログファイルを表示することができます。

```
% tail -f log.txt
```

これらのコマンドの詳細については、UNIX のマニュアルや書籍をご覧ください。

計算が正常に終了すると、以下のようなメッセージが表示されます。

```

      .
      .
      .
*****
      output restart file, iter=(      23), time=( 2.300000+00)
*****
*****
      execution terminated normally
*****
      Wall time : 0000:00:05 | Stop in Year:2004 Mon :07 Day :15 Hour:11 Min :17 Sec :30
STOP Subroutine name: FRONTFLOW

```

### \*fflowHPC

カレントディレクトリを作業ディレクトリに変更し、コマンドプロンプトより MPI 上で fflowHPC を起動します。

```
% mpirun -np 4 fflows [Enter キー] *
```

\* mpirun のオプション -np に続く整数は、並列実行時の並列度です。この値は計算制御ファイルの hpc 変数群の ncpu 変数及び hpc\_cntl 変数群の NPE 変数に指定した値と同一値としてください。詳しくは MPI および mpirun コマンドのマニュアルをご覧ください。

メインソルバ実行中は、計算ログが標準出力に出力されます。

```
#####
####
####                                WELCOME TO FrontFlow/red                                ####
####                                ####
####                                FrontFlow/red Version 2.0                                ####
####                                Year:2004 Mon :12 Date:07                                ####
####                                ####
####                                Hour:0010 Min :37 Sec :06                                ####
####                                ####
#####
-----
Domain number      :      1
transport model    :      constant
viscosity [Pa s]   :      0.182E-04 (mu0)
Schmidt number     :      1.000 (Sc)
Prandtl number     :      0.720 (Pr)
  viscosity        : m = mu0
  conductivity     : l = Cp*m/Pr (Cp:isopiestic specific heat)
  diffusivity      : D = m/(r*Sc) (r:mass density)
-----
### MSG: [nostop=0] : ICCG should stop at      1000
number of chemical species :      1
1 air      mi = 0.29000E-01 [kg/mol]
      .
      .
      .

```

計算が正常に進行している間は、下記のようにモニター値が各時間ステップで出力されます。

```

      .
      .

```

=====																	
TIME STEP :		6															
TIME : 0.6000E+00		TIME INTERVAL: 0.1000E+00		MAX. COURANT : 0.5760E+00		AT VERTEX :		289									
-----																	
U		V		W		P		T	K	EPS		COMP_1					
-----																	
MONI :		-.8981E+00		0.2698E+00		-.7791E+00		0.1000E+02		0.3007E+03		0.0000E+00		0.0000E+00		0.1000E+01	
MIN :		-.6090E+01		-.7673E+01		-.1350E+02		-.9018E+01		0.2270E+03		0.1000E+26		0.1000E+26		0.1000E+01	
MAX :		0.5570E+01		0.8503E+01		0.1107E+02		0.7349E+02		0.5416E+03		-.1000E+26		-.1000E+26		0.1000E+01	
-----																	
OUTER-ITER NO. :		2															
-----																	
INNER-ITER :																	
ITER :		2		2		2		14		2		0		0		0	
RES :		0.1000E-05		0.1000E-05		0.1000E-05		0.1000E-05		0.1000E-05		0.0000E+00		0.0000E+00		0.1000E+01	
ABS :		0.1000E-05		0.1000E-05		0.1000E-05		0.1000E-05		0.5802E-02		0.0000E+00		0.0000E+00		0.1000E-05	
-----																	
OUTER-ITER ERROR :		CNVERGENCE TOLERANCE : 0.2000E+00															
ERR :		0.7360E-01		0.9620E-01		0.1416E+00		0.1360E+00		0.4005E-01		0.0000E+00		0.0000E+00		0.0000E+00	
-----																	
.																	
.																	
.																	

### 参考

計算ログをファイルに残したい場合は、下記のようにリダイレクトを行ってください。

```
% mpirun -np 4 ./fflowHPC >& log.txt & [Enter キー]
```

プログラムはバックグラウンドで実行され、標準出力、標準エラー出力はファイル `log.txt` に保存されます。なお、お使いのシステムによっては、バックグラウンド実行時、標準入力にリダイレクトを要求する場合があります。この場合には、

```
% mpirun -np 4 ./fflowHPC < /dev/null >& log.txt & [Enter キー]
```

などとしてください。詳しくはお使いのシステムにインストールされている `MPI` および `mpirun` のマニュアルをご覧ください。システム管理者にご相談ください。

また、`tail` コマンドを使用すると、書き出し中のログファイルを表示することができます。

```
% tail -f log.txt
```

`tail` コマンドの詳細については、UNIX のマニュアルや書籍をご覧ください。

計算が正常に終了すると、以下のようなメッセージが表示されます。

```

      .
      .
      .
*****
      output restart file, iter=(      10), time=( 5.000000-04Sec.)
*****
      execution terminated normaly
*****
      Wall time : 0000:01:51 | Stop in Year:2004 Mon :12 Day :07 Hour:10 Min :38 Sec :57
*****

```

\* バッチジョブファイルの一例（お使いのシステムによりオプションなど異なりますので詳しくはシステム管理者にお問い合わせください）

```
#!/bin/sh
#PBS -q P04
#PBS -N sample
#PBS -l ncpus=4
#PBS -l nodes=1:ppn=4

cd $PBS_O_WORKDIR

mpirun -machinefile ${PBS_NODEFILE} -np 4 ./fflowHPC > log.txt

exit
```

### 3.2. 計算を途中で止める

メインソルバの実行を途中で止めるためにオペレーティングシステムの強制終了コマンドを使用すると、最終ステップにおける計算結果やリスタートデータが出力されません。これらのファイルを出力して正常終了させるには、作業ディレクトリに「**stop**」という名前のファイルを作成します。作成するとメインソルバは次のステップで正常終了します。ファイルの内容は必要ありません。また作成方法も任意です。作成方法の例を示します。

作業ディレクトリで「**touch stop**」と入力することで、作業ディレクトリに **stop** という名前の空ファイルを作成することができます。

```
% touch stop [Enter キー]
```

なお、この方法ではメインソルバは次の計算ステップで停止します。プログラムが暴走している場合等、直ちに停止させたい場合はオペレーティングシステムの強制終了コマンドを使用してください。

---

### 参考

オペレーティングシステムの強制終了コマンドを使用してプログラムを停止するには、次のように行ってください。詳しくはお使いのオペレーティングシステムの説明書または解説書などをご覧ください。

「**control-c**」キーを押すことで、実行中のプログラムを強制終了することができます。プログラムをバックグラウンドで実行している場合はフォアグラウンドにしてから「**control-c**」キーを押してください。これで強制終了できない場合は、「**kill**」コマンドを使用します。はじめに強制終了するプロセスのプロセス番号(PID)を調べます。

```
% ps [Enter キー]
```



PID	TTY	TIME	CMD
8297	pts/2	00:00:00	bash
8366	pts/2	00:00:00	fflowS
8367	pts/2	00:00:00	ps

異なるシェルで実行中のプロセスを表示するには「`-u userid`」などのオプションが必要となる場合があります。詳しくは `ps` コマンドのマニュアルをご覧ください。

次に `kill` コマンドでプロセスを強制終了します。

```
% kill -9 8366 [Enter キー]  (+8366 は上で調べたプロセス番号です。-9 は強制終了のオプションです)
```

これで強制終了できない場合は、システム管理者にご相談ください。

### 3.3. 計算の途中経過を出力する

作業ディレクトリ内に“output”という名前のファイルを検知すると、実行中のタイムステップで解析結果とリスタートファイルを出力します。

作業ディレクトリで「`touch output`」と入力することで、作業ディレクトリに `output` という名前の空ファイルを作成することができます。

```
% touch output [Enter キー]
```

### 3.4. 計算をリスタートする

作業ディレクトリにリスタートデータファイルがある場合、計算をリスタートすることができます。リスタートデータのファイル名は計算制御ファイルの `files` 変数群の `restart` 変数で指定した文字列にステップ数を加えたものです（通常、`restart_3000` など）。リスタートデータはメインソルバが正常終了したステップの他、計算制御ファイルの `output` 変数群で指定した間隔で書き出されています。

計算をリスタートするには、リスタートデータファイルだけではなく、計算制御ファイル、格子データファイルやその他の中間データファイル類も必要です。作業ディレクトリを変えてリスタートする場合には、これらのファイルもすべてコピーしてください。

ここでは、3000 ステップからのリスタートを行う場合を例として説明します。

1. 作業ディレクトリにリスタートを行うステップのリスタートデータが存在するか確認してください。ここではリスタートデータのファイル名は `restart_3000` とします。
2. 計算制御ファイルの編集を行います。`time` 変数群の `start` 変数を、リスタートする時刻に修正します。`end` 変数も必要に応じて修正します。`files` 変数群の `initial` 変数に、リスタートデータファイルのパスを設定します。

```
(例)
&time
```

```

start = 3000
end = 20000,
flowcon = 2,
toltime = 1.d-3

/
&files
  gdformat = 'FV'
  vertex = gridfile.grd'
  result = './result',
  restart = './restart'
  initial = './restart_3000'
  force = './presforce'

/

```

3. メインソルバを実行します。

並列計算の場合は、計算制御ファイルの変更は以下のようになります。単一CPUの場合と異なり、読み込むリスタートファイルの名前は、`hpc_cntl`変数群で定義します。

```

(例)
&time
  start = 3000
  end = 20000,
  flowcon = 2,
  toltime = 1.d-3

/
&hpc_cntl
  NPE = 4
  mtsfil = 'test.m'
  subdir = 'hpc'
  hpc_vertex = 'Grid_hpc'
  hpc_boundary = 'Bund_hpc'
  hpc_comm = 'Comm_hpc'
  hpc_source = 'source'
  hpc_result = 'result'
  hpc_restart = 'restart'
  hpc_anim = 'result_anim'
  hpc_initial = 'restart_3000'
  walflg = 0
  wall = './wall.law'
  ucdflg = 0
  ucdfile = 'test.inp'

/

```

#### 4. ユーザサブルーチンの使用法

ユーザー定義サブルーチンを利用して、初期値設定、境界条件設定、ソース項設定、出力処理などを行うことができます。必要に応じてこれらのユーザーサブルーチンファイルを編集・修正します。

修正したユーザー定義サブルーチンを含んだ実行モジュールのコンパイルを行うには、まず `src_user` に移動します。ここに、ユーザが編集可能なユーザサブルーチンのファイルが入っていますので、必要に応じて編集します。

次に、`src_main/src/` に移動します。必要に応じて `Makefile` を修正して下さい。その後、`make all` によってコンパイルして下さい。

```
% cd xxx/FFR_V3.1.002/user_src
***** modify the USER_xxx.f *****
%cd ../src_main/src
%make all
```

コマンド実行後、1CPU 用ならびに並列用の実行モジュールがビルドされます。この実行モジュールを作業ディレクトリにコピーして実行します。

また、ユーザー定義サブルーチンを使用する際には、計算制御ファイルの `usrsub` 変数群にて、使用するサブルーチンの設定を正しく行う必要があります。

## 5. 後処理のためのコンバート

### 5.1 解析結果のコンバート : ffr2viz の実行

メインソルバの実行が正常終了すると、計算結果ファイルが書き出されます。計算結果ファイルは、1CPU 計算の場合計算制御ファイルの `files` 変数群の `result` 変数で指定した文字列に、拡張子「.frontflow」を追加したファイル名で、作業ディレクトリに書き出されます。並列計算の場合は、計算制御ファイルの `hpc_cntl` 変数群の `hpc_result` 変数で指定した文字列に、拡張子「.frontflow」を追加したファイル名で、作業ディレクトリ内の各並列サブディレクトリ内に書き出されます。計算結果ファイルはメインソルバが終了したステップの他、`output` 変数群で指定した間隔でも書き出されています。データ書き出しごとに上書きをする設定では最後に書き出された時刻ステップのデータのみ記録されています。上書きをしない設定ではファイル名の末尾に出力ステップ数が追加されます。

計算結果を市販の可視化ツールを使って可視化するには、`ffr2viz` コマンドを使用して計算結果ファイルを変換します。`FrontFlow/red` の `ffr2viz` は、`FieldView` 形式、`AVS` 形式および `Paraview` 形式への変換に対応しています。

`ffr2viz` は作業ディレクトリで実行します。変換には計算結果ファイルの他、計算制御ファイル、格子データファイルが必要です。並列計算の結果を変換する場合はメインソルバ実行時の一時データファイルの一部も必要となります。

`ffr2viz` プログラムの使用方法およびオプション説明は以下のとおりです。

<code>ffr2viz -r 結果ファイル -g 格子データファイル -gf 格子データ形式 -rf 変換形式 -o 出力ファイル名</code>
---

#### オプション

- `-r resultfile`    計算結果ファイル(\*.frontflow)を指定します。必須オプションです。(–nr オプションを指定した場合のみ省略できます)
- `-g gridfile`    格子データファイル(\*.grd / \*.ffg)を指定します。必須オプションです。
- `-gf FV/GF/FF`    格子データの形式を指定します。`FieldView` 形式の場合は「FV」、`FFR-GF` 形式の場合は「GF」、`FrontFlow native` 形式の場合は「FF」と指定します。デフォルトは FV です。
- `-rf FV/FVA/AVS`    変換出力ファイルの形式を指定します。`FieldView UNFORMATTED` 形式の場合は「FV」、`FieldView ASCII` 形式の場合は「FVA」、`AVS` 形式の場合は「AVS」、`Paraview` 形式の場合は「VTK」と指定します。デフォルトは FV です。
- `-o filename`    変換出力ファイルのファイル名を指定します。必須オプションです。
- `-nr`    変換出力ファイルに計算結果を含めず、格子データのみ出力します。
- `-m int-value`    並列計算結果を変換する場合に並列度を指定します。並列計算結果を変換するとき必須です。
- `-d subdir`    並列計算サブディレクトリ名を指定します。デフォルトは `hpc` です。

-gs 実数                      グリッドファイルのスケーリングファクター

以下にコマンド入力例を示します。ここでは、計算結果ファイルを **result.frontflow**、格子データを **geom.frontflow**、格子データ形式を **FrontFlow native** 形式、出力ファイルを **result\_cnv** とします。

FieldView UNFORMATTED 形式に変換するには、作業ディレクトリにおいて、下記のオプションで **ffr2viz** を実行します。並列度 8 の例です。

```
% ffr2viz -m 8 -r result.frontflow -g geom.frontflow -gf FF -o result_cnv -rf FV [Enter キー]
```

FieldView ASCII 形式に変換するには、下記のように実行します。

```
% ffr2viz -m 8 -r result.frontflow -g geom.frontflow -gf FF -o result_cnv -rf FVA [Enter キー]
```

### AVS 形式への変換（1CPU 計算結果）

AVS 形式に変換するには、作業ディレクトリにおいて、下記のオプションで **ffr2viz** を実行します。

```
% ffr2viz -r result.frontflow -g geom.frontflow -gf FF -o result_cnv -rf AVS [Enter キー]
```

### AVS 形式への変換（並列計算結果）

AVS 形式に変換するには、作業ディレクトリにおいて、下記のオプションで **ffr2viz** を実行します。並列度 8 の例です。

```
% ffr2viz -m 8 -r result.frontflow -g geom.frontflow -gf FF -o result_cnv -rf AVS [Enter キー]
```

いずれの場合も、**ffr2viz** を実行し正常終了すると下記のようなメッセージが表示されます（オプションや格子データ形式によりメッセージは若干異なります）。

```
////////////////////////////////////
////////////////////////////////////
* FFR2VIZ *
ver. 2.02 (07/29/2004)
////////////////////////////////////
////////////////////////////////////

Convert Single CPU data.
### Reading FFR Grid element type field...
### Reading FFR Grid boundary type field...
### Reading FFR Grid vertices coordinate field...
### Reading FFR Grid Boundary Face field...
.
. (略)
.

#### Variables number and name : 7 velo_v
#### Variables number and name : 8 velo_w
#### writing FV Header completed.
#### Writing FV variable data ...
#### Hexa elements= 19380
#### Appending data for FV completed.

END: ffr2viz conversion is successfully completed.
```

## 5.2 粒子追跡データのコンバート : ffr2vizp の実行

計算制御ファイルの `model` 変数群で粒子追跡計算を有効にしていると、投入粒子の各時間ステップごとの位置がデータとして出力されます。投入粒子位置データは「PT\_RESULTxxxxxx」というファイル名(ユーザ定義可能です)で、作業ディレクトリ内に出力されます。xxxxxx は出力した時間ステップを表します。

FieldView の `particle path` としてインポートするには `ffr2vizp` プログラムを使って投入粒子位置データを時系列にまとめた FieldView の `particle path format` に変換します。変換後は 1 つのファイルにまとめられます。

`ffr2vizp` は作業ディレクトリで実行します。変換には投入粒子位置データの外、変換制御ファイルが必要です。変換制御ファイルは「`ffr2fvpc_info.txt`」という名前で、粒子位置ファイル数、出力ファイル名と粒子位置ファイル名を記述します。出力ファイル名は 20 文字以内で指定します。

```
(変換制御ファイルの例、「path.fvp」は出力ファイル名)
file 1 path.fvp

PT_RESULT003000
```

複数の時間ステップの粒子位置データを変換するときは次のように記述します。

```
(変換制御ファイルの例、「path.fvp」は出力ファイル名)
file 3 path.fvp

PT_RESULT001000
PT_RESULT002000
PT_RESULT003000
```

複数のファイルを変換する際には、時系列に沿って順番に記述する必要があります。時系列に並んでいないと FieldView に読み込みできません。

投入粒子位置データファイル、変換制御ファイルのある作業ディレクトリで

```
% ffr2vizp [Enter キー]
```

を実行すると、FieldView の `particle path format` に変換されたデータファイルが指定されたファイル名で出力されます。コンバートしてできたデータの可視化の方法については FieldView のマニュアルをご覧ください。