Research and Development for Next-generation Information Technology of
Ministry of Education, Culture, Sports, Science and Technology
"Research and Development of Innovative Simulation Software"

CISS Free Software

FrontFlow/blue version.8.1
FrontFlow/blue-ACOUSTICS version.2.3

User Manual

Corresponding Clerks:
(Engagement) The Foundation for the Promotion of Industrial Science (F.P.I.S)
4-6-1 Komaba, Meguro-ku, Tokyo 153-8505 JAPAN
(Management) Center for Research on Innovative Simulation Software,
Institute of Industrial Science (IIS), the University of Tokyo
4-6-1 Komaba, Meguro-ku, Tokyo 153-8505 JAPAN
Fax : +81-3-5452-6662
E-mail : software@ciss.iis.u-tokyo.ac.jp

Contents

Chapter 1

Introduction

1.1 Program Functions

FrontFlow/blue (FFB) is a general purpose, explicit finite element code that calculates incompressible unsteady flows in arbitrarily shaped geometries. The governing equations are the unsteady incompressible Navier-Stokes equations in Cartesian coordinates.

1.2 Implemented Program Features

The following features, among many others, are implemented in this version of the program.

- Large Eddy Simulations
- Overset Flow Computations
- Parallel Computations
- Implicit Method for the Momentum Equations
- Heat Transter
- Cavitation Model
- Multi phase Flow
- Aeroacoustics Noise
- REFINER
- Utility Programs

1.2.1 Large Eddy Simulations

Static and dynamic Smagorinsky models are currently implemented for Large Eddy Simulations (LES) using an appropriate numerical method and wall boundary conditions.

1.2.2 Multi-frame of Reference Dynamic Overset Flow Computations

This version of FFB supports multi-frame of reference dynamic overset flow field computations. Namely, it will overset the velocity vectors and pressures of the nodes and elements specified in the overset conditions data file, using the values interpolated in the elements specified by the overset conditions data, considering the difference in velocities of the reference frames. Velocities of reference frames are defined by the frame attributes of each element (i.e. the frame number) and the velocity of each frame. An arbitrary number of reference frames and frame velocities can be defined, including a rotating frame of reference. FFB will read or generate overset condition data at every JSET time step(if any). Thus, it is possible to compute flow fields around any number of arbitrarily moving bodies by preparing overset condition data that appropriately correspond to the motions of interest.

1.2.3 Parallel Computations

Parallel computations are available using the domain-decomposition programming model. At the beginning of execution, FFB queries the domain number IPART that it should compute. When this number is greater than zero, FFB recognizes it is running in parallel mode and it will perform the necessary communications with the other processes computing other domains. All communications are done within the framework of boundary condition transactions. Hence, parallel-mode executions can be regarded as a serial mode execution extended to deal with "communication boundary conditions". Actually, a single code runs in both serial and parallel modes, depending on the read-in boundary condition data, although parallel-mode execution requires that executables be linked with the basic communication library. For both of execution modes, FFB is fully tuned for generic vector architecture machines as well as for RISC architecture machines.

1.2.4 Implicit Method for the Momentum Equations

To maintain temporal accuracy, Direct Numerical Simulations (DNS) and Large Eddy Simulations (LES) usually use explicit time advancement methods for turbulent flows to keep temporal accuracy. However, there are several situations in which the limit on the time increment using explicit methods is very strict and the computational cost is very high. One example is wall-bounded flow; if the grid space in the near-wall region is small, because of the limit on the diffusion number, the time increment is very small. In this situation at least a semi-implicit scheme is required. Another case is flow around an airfoil. Near the leading edge of anairfoil, the flow is accelerated so that the time increment is restricted by the limit on the local CFL number. In this case, a fully implicit method is very useful to remove the limit on the CFL condition. The Euler backward scheme and the Crank-Nicolson scheme are implemented in this version. The Euler backward scheme provides a slightly faster solution than the Crank-Nicolson scheme, it is less accurate. The former is first-order in time and is associated wit numerical dissipation proportional to the local flow velocity multiplied by the square of the time increment, while the latter is second-order in time and dissipation free. In LES, the Crank-Nicolson scheme is recommended.

1.2.5 Heat Transfer

The flow solver can compute heat transfer in both fluids and solids. Buoyancy can be treated but heat radiation is not supported.

1.2.6 Cavitation Model

This version of FFB implements a cavitation model based on one-phase flow with a source/sink term. The transform equation of liquid fraction will be computed. Generation and disappearance of cavitation are be coupled with the continuity equation in which we assume weak compressibility (low-Mach assumption).

1.2.7 Aeroacoustics Noise Prediction (Curles's Equation)

Predictions of aeroacoustics noise, base on Lighthill's analogy, are supported. Sound source data can be computed by the flow solver in FFB. Curle's equations are implemented as utility programs to calculate sound field using sound source data.

1.2.8 Aeroacoustics Noise Prediction (Acoustic Solver)

Acoustic analysis code is also available to predict aeroacoustics noise in FFB-A [1], which is a subsystem of FFB. One of features of the acoustic solver is that it supports large-scale acoustics computation using up to 100 million grids. Sound fields with complex geometries or high frequencies can be computed by using a large scale grid.

1.2.9 REFINER

The auto mesh refinement function (REFINER) is supported. The REFINER divides a element into eight elements in the flow solver and can be used repeatedly. If REFINER is used three times (four times), the original mesh size will be 512 times (4096 times) in the flow solver. Theoretically, there are no limits

[1] FrontFlow/blue-ACOUSTICS

in computation size with using this function. Users can specify the region to be refined (local refine); therefore, it is possible to use a small-size grid only for the regions near walls.

1.2.10 Utility Programs

The FFB system includes not only the flow solvers but many other types of utility programs to support preparing data, decomposing, analyzing, converting and so on. These utility programs can be accessed by the user interface FrontWorkBench (FWB).

Chapter 2

Installation

To install FrontFlow/blue (FFB), the archived file "FrontFlow_blue.8.1.tar.gz" must be uncompressed and restored as follows;

```
%gzip -d  FrontFlow_blue.8.1.tar.gz
%tar -xvf  FrontFlow_blue.8.1.tar
```

This creates a directory 'FFB8' whose contents are as follows;

Directory	Content
bin	execute files
data	sample data
doc	user manual (Japanese and English)
include	header files
lib	source codes for library and library files
make	file for compiling
util	source codes for utilities

FFB supports two installation methods; auto mode and manual mode. Auto mode is recommended for the first installation because it is relatively simpler; manual-mode may be useful when you want to partially re-compile.

In this chapter, the auto mode will is explained first (2.1), then the manual mode is explained (2.2) Notes on installation appear in the last section (2.3).

2.1 Install by Auto mode

The procedure for an auto mode installation are as follows;

1. Set the environment variable ${LES3DHOME}
2. Set platform information (OPTION)
3. Make library files and execution files using a make-script
4. Make a parallel mode version (needed.)

2.1.1 Set Environment Variable : LES3DHOME

The enviroment variable LES3DHOME identifies the local home directory for FFB. If FFB8 is at /home/yourname/ and you are using the C-shell, then set the variables as follows;

```
%setenv LES3DHOME /home/yourname/FFB8
```

If you are using a Bourne-shell, set as following;

```
%LES3DHOME=/home/yourname/FFB8
```

You can use the following command to confirm that the variable was properly set.

```
%echo $LES3DHOME
```

.

2.1.2 Set Platform Information

Before compiling, you have to specify platform information. The information which you need specify is

CPP	command name to invoke C Pre-Processor
POPT	options for C pre-processor
CCOM	command name to invoke C-compiler
COPT	options for C compiler
FCOM	command name to invoke Fortran-compiler
FOPT	options for Fortran compiler
INCDIR	directories searched for the system including files
LIBDIR	directories searched for system library files

This information will be specified in the file named 'OPNTION', and it is at ${LES3DHOME}/make . A sample OPTIONS file is shown in Figure 2.1.

FFB has been used on various platforms including AIX, HI-UXMPP8,HP-UX, IRIX, OSF1, SunOS, UNIX_System_V Eart Simulaotor, BG_P, HA8000, FX-10, K and Linux.

```
# Compilers and Option flags
CPP  = /lib/cpp
CCOM = cc
POPT = -P
COPT =
FCOM = f77
FOPT = -O2 -w0

# Locations of include file(s) and library file(s) for IRIX
INCDIR = /usr/local/mpi/include
LIBDIR  = /usr/local/mpi/lib/hmpp2/cml
```

Figure 2.1: Example of an OPTIONS file for setting platform information

2.1.3 Execute Make script

If you properly set ${LES3DHOME} and the platform information, then you can install FFB by just executing a make script as follows:

```
%cd ${LES3DHOME}/make
%Makeall
```

This may take some time. [1] If no problems are encountered, five library files will be created at ${LES3DHOME}/lib, and about 100 execution files will be created at ${LES3DHOME}/bin.

It is preferable for teh library files to be created properly, but serious problems do not arise if all execution files are not created. Actually, some execution files may not be created owing to the memory limitations of your platform.

2.1.4 Install Parallel Version (if needed)

The parallel version of FFB will not be created by the auto mode make script, because all platforms do not necessarily support the MPI library. However, if MPI is installed on your platform, you can install the parallel version of FFB using the following procedure. [2]

1. Make the library file for MPI library handling

2. Make the execution file : les3x.mpi

[1] The time needed depends on the platform and specified options in the OPTION file. Roughly speaking compiling and linking will take about 5-30 minutes.

[2] Note that the source codes are exactly the same in the serial and the parallel version of FFB, but the library files to be linked differ.

```
%cd $LES3DHOME/lib/src/dd_mpi
%make lib
```

```
%cd $LES3DHOME/util/les3x.mpi
%make
```

2.2 Install by Manual-mode

The procedure for installation using manual-mode is as follows:

1. Set the environment Variable (${LES3DHOME})
2. Set platform information (OPTION)
3. Make library files
4. Make the execution file for les3x
5. Make the execution files for other utility

2.2.1 Set Environment Variable : LES3DHOME

The procedures for setting the environment variable are completely the same as those described in section 2.1.1. See 2.1.1 for details.

2.2.2 Set Platform Information

The procedure for setting the platform informations is completely the same as that described in section 2.1.2. See 2.1.2 for details.

2.2.3 Make Library Files

The following 11 directories in $(LES3DHOME)/lib/src/,

```
dd_dmy/ dd_mpi/  fort/ gf/ gf2/ msl2/ tetra/ tetra/ multi/
REVOCAP_Refiner-0.4.3/  ParMetis-3.1
```

You can make library files corresponding to each directory as follows :

```
%cd ${LES3DHOME}/lib/src/dd_dmy
%make lib
%cd ${LES3DHOME}/lib/src/dd_mpi
%make lib
%cd ${LES3DHOME}/lib/src/fort
%make lib
%cd ${LES3DHOME}/lib/src/gf
%make lib
%cd ${LES3DHOME}/lib/src/gf2
%make lib
%cd ${LES3DHOME}/lib/src/msl2
%make lib
%cd ${LES3DHOME}/lib/src/tetra
%make lib
%cd ${LES3DHOME}/lib/src/multi
%make lib
%cd ${LES3DHOME}/lib/src/REVOCAP_Refiner-0.4.3 ; make Refiner
%cd ${LES3DHOME}/lib/src/ParMetis-3.1/METISLib/; make
%cd ${LES3DHOME}/lib/
%ln -s ./src/REVOCAP_Refiner-0.4.3/lib/x86_64-linux/libRcapRefiner.a  .
%ln -s ./src/ParMetis-3.1/libmetis.a
```

Nine library files will be created at $LES3DHOME/lib. Note that the MPI library should be installed on your platform before creating a library file corresponding to dd_mpi.

A script file ${LES3DHOME}/lib/src/Makeall is available to create library files except the file for MPI.

2.2.4 Make Execution File for les3d, les3c, and les3ct

You can make the les3d, les3c, les3ct and les3x files as follows:

```
% cd $(LES3DHOME)/proj/les3d
% make
% cd $(LES3DHOME)/proj/les3c
% make
% cd $(LES3DHOME)/proj/les3ct
% make
% cd $(LES3DHOME)/proj/les3x
% make
```

and you can make the les3d.mpi execute file as follows:

```
% cd $(LES3DHOME)/proj/les3d.mpi
% make
% cd $(LES3DHOME)/proj/les3c.mpi
% make
% cd $(LES3DHOME)/proj/les3ct.mpi
% make
% cd $(LES3DHOME)/proj/les3x.mpi
% make
```

.

This is a parallel execution mode of les3x. These two excution files are compiled from the same codes, but the library file to be linked for domain decomposing is different. The library file to be linked is

- les3x ← ${LES3DHOME}/lib/dd_dmy
- les3x.mpi ← ${LES3DHOME}/lib/dd_mpi

2.2.5 Make Execution Files for Other Utilityies

Except for les3x and les3x.mpi, there are ten directories in ${LES3DHOME}/util/ as follows :

```
bcmod/    check/    convert/  gfcat/
hscat/    mesh/     mis/      sound/
tetra/
ddrgb/    ddrgbt/    ddelm/    ddelmt/  metis/
```

You can create utility excution file as follows:

```
% cd ${LES3DHOME}/util/'each-directory'
% make  PROG='utility-name'
```

where 'each-directory' is a directory among those at ${LES3DHOME}/util/ except for les3d and les3d.mpi, and 'utility-name' is the name of the utility which you want to create.

For example, if you want to create 'chknd', whose source code is at ${LES3DHOME}/util/check, the following commands are needed:

```
% cd ${LES3DHOME}/util/check
% make  PROG=chknd
```

See Chapter **??** for the details regarding each utility.

Make-scripts such as
${LES3DHOME}/util/Makeall
${LES3DHOME}/util/'each-directory'/Makeall
are available to compile automatically.

2.3 Notes

- All execution files will be created in ${LES3DHOME}/bin. Hence, it is recommended that ${LES3DHOME}/bin be added to your PATH, via one of following command.

  For a C-shell,

```
%set path = ($path ${LES3DHOME/bin} )
```

  and for a Bourne-sell,

```
%PATH= $PATH: ${LES3DHOME/bin}
```

Chapter 3

File System

In the framework of FFB, all files except the parameter file use the same file format. This file format is the original one to FFB; it is called the general file format (GF). Data flow in the FFB system is shown in Figure 3.1. FFB [1] reads data from the parameter and GF files and outputs GF files. In this chapter, teh format used in the parameter file and the GF files are explained.

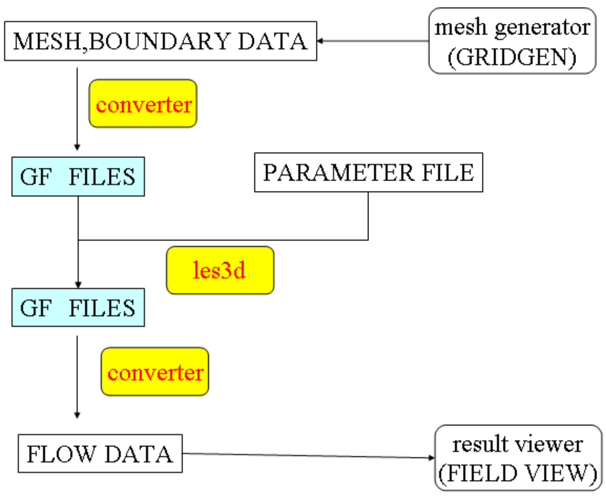

Figure 3.1: Data Flow of basic functions in FFB

[1]In this manual, les3d is assumed to be the name of utility and FFB is assumed to be the name of the system

3.1 Parameter File

The FFB system contains four flow solvers: les3d, les3c, les3ct, les3x. In this section, the details regarding the parameter file for les3x are explained.

3.1.1 control parameters for les3x

The flow solver les3x reads the simple ASCII format file shown in Fig. 3.2. [2] Parameters in this file can be categorized as shown in the table 3.1.

Table 3.1: Category of paremeters for les3x

Line Num.	Contents
1 ∼ 2	File Header
3 ∼ 4	Analysis Mode steady/unsteady, turbulence model, numerical scheame
5 ∼ 6	Normalization
7 ∼ 8	Physical Model heat transfer flag, solid treatment flag
9 ∼ 10	overset flag, midnode flag, multi-frame flag etc.
11 ∼ 14	Physical Properties viscosity, density, specific heat, heat conductivity
15 ∼ 20	Time Integration and Matirx Solver time increment, restart flag, criteria for convergence
21 ∼	Optional Output optional output flag, output interval
After Sample Setting	File name filename of mesh B.C., flow, etc.

See table 3.2、3.3、3.4 for locations of each parameter.

[2] Note：Line numbers of in the left colum of Figure 3.2 do not apper in an actual parameter file.

```
#FFB_V07
PARAMETER FILE FOR LES3X IN FFB VER.7.1
#GIVE ITRANS IMODEL IFORM  IPRESS FSMACH
      1      0      4      1      0.00
#GIVE DREFF  UREFF  TREFF  TOUT
      1.0    30.0   270.0  1.0
#GIVE IHEAT  ISOLID ICAV   IBUSNQ
      0      0      0      0
#GIVE NFRAME ISET   JSET   OMEGA
      0      0      0      0.0
#GIVE VISCM  SIGMA  RHOF   RHOS
      1.54e-5 0.0   1.0    10.0
#GIVE CONDF  CONDS  CPF    CPS
      1.0E-1 1.0E-0 10.0   1.0
#GIVE ISTART NTIME  DT     NMAXSE NMAXSI
      0      500    0.01   100    1
#GIVE NMAXT  NMAXP  EPST   EPSP   EPSS
      10     40     1.0E-6 1.0E-6 1.0E-20
#GIVE TFINAL UFINAL VFINAL WFINAL
      1.0    0.00   0.00   0.00
#GIVE IOUT   INTFSV INTSFV
      0      0      0
#GIVE NSMPL  LSMPL  XSMPL  YSMPL ZSMPL
      6
1 0.5 0.0 0.15
2 0.5 0.0 0.15
3 0.5 0.0 0.15
4 0.5 0.0 0.15
1 0.5 0.0 0.25
4 0.5 0.0 0.25
#GIVE MESH         FILE NAME
../../../DATA/data1/MESH
#GIVE B.C.         FILE NAME
../../../DATA/data1/BOUN
#GIVE INITIAL FLOW FILE NAME
DUMMY
#GIVE FINAL  FLOW FILE NAME
FLOW
#GIVE HISTORY      FILE NAME
HISTORY
#GIVE ATTRIBUE     FILE NAME
../../../DATA/data1/ATTR
#GIVE MID-NODE     FILE NAME
../../../DATA/data1/MID
#GIVE OVERSET      FILE NAME
../../../DATA/data1/OVER
#GIVE AVERAGE      FILE NAME
AVE
#GIVE RMS          FILE NAME
RMS
#GIVE STRESS       FILE NAME
STR
#GIVE CURRENT FLOW FILE NAME
FLOWS
#GIVE SURFACE PRS. FILE NAME
SURF
#END OF INPUT DATA

#OPTIONS
#MONITOR
x#STOPNOW
#OPTIONE
```

Figure 3.2: Example of a parameter file (PARMLES3X)

File Header

- `#FFB_V07` : File Head
  File Head `#FFB_V07` should be specified. The flow solver will stop without the file header.

Settings for Analysis Mode

- ITRANS: Steady/Unsteady Flag Unsteady computations based on the fractional step or steady computations based on either the SIMPLE method or SIMPEC method can be selected. Possible values for ITRANS are 1, 2, or 3, as follows.

ITRANS	pressure method	steady/unsteady
1	Fractional Step	unsteady
2	SIMPLE	steady
3	SIMPLEC	steady

- IMODEL: Turbulence Model
  The turbulence model can be one selected from those listed below.

IMODEL	turbulence model	steady/unsteady
0	no turbulence model	steady, unsteady
1	LES: Standard Smagorinsky Model (SSM)	unsteady
2	LES: Dynamic Smagorinsky Model (DSM)	unsteady
12	RANS: Low-Re type $k-\epsilon$ (Launder-Sharma model)	steady
13	RANS: Low-Re type $k-\epsilon$ (Chien model)	steady

Note that LES must be selected in unsteady computation, RANS must be selected in steady computation.

- IFORM: Momentum Equation Flag In unsteady computations, the Crank Nicolson method is used. Balancing tensor diffusivity (BTD) and the Euler backward method (only for diffusion term) are implemented as stabilizing functions in the momentum equations. Values for the parameter IFORM control these methods as follows.

IFROM	Scheme for Momentum Equation
1	Crank Nicolson method
2	Crank Nicolson method + BTD term
3	Crank Nicolson method Euler backward method (only for diffusion term)
4	Crank Nicolson Method + BTD term Euler backward method (only for diffusion term)

- IPRESS: Pressure Method Flag This parameter controls pressure method in unsteady computations.

  - IPRESS=1：Fractional-step method
  - IPRESS=2:Fractional-step method with Low-Mach assumption. The convection term in the pressure equation is computed explicitly based on the Galerkin method.

- FSMACH: Mach number in low-Mach assumption

Settings for Normalization

The flow solver‘ les3x ’uses dimensional data in SI units. The solver reads dimensional data, normalizes that dimensional data data, computes the flow, converts the normalized data back to dimensional data, and outputs dimensional data. The flow solver uses the parameters D000, U000 and T000 for normalization. Normalized data can be used by setting these parameters to 1.0.

- D000: Characteristic Length
  Set the characteristic length [m] in the computational model.
- U000: Characteristic Velocity
  Set the characteristic velocity [m/s] in the computational model.
- T000: Temperature Difference
  Set the temperature difference between maximum and minimum temperatures [K] in the computational model.
- TREF: Characteristic Temprature
  Set the characteristic temperature [K] in the computational model.

Settings for the Physical Model

- IHEAT: Heat Transfer Flag
  Heat transfer computations will be active when IHEAT is set to 1.
- ISOLID: Solid Treatment Flag
  Heat transfer computation in both fluids and solids will be active when both ISOLID and IHEAT are set to 1.
- ICAV : Cavitation Model Flag
  This parameter is not used.
- IBUSNQ: Boussinesq Approximation Flag
  This parameter is note used, beacuase the Boussinesq approximation is applied regardless of the value set for this parameter.

Settings for Overset, Midnode, Multi-frame

- NFRAME: Multi-frame Function Flag This parameter controls the multi-frame function and must be set to 0, -1 or 1. A single frame computation will be done when NFRAME is set to either 0; then all element are assumed to be in a static frame of reference. The multi frame function will be active when NFRAME is set to -1 or 1. In the multi frame mode, some elements are assumed to be in a rotational frame (NFRAME = -1) or in an acceleration frame (NFRAME=1). The frame of each element is specified in the GF attribute files.
- ISET: Midnode Flag This parameter controls midnode computations and must be set to either 0, -1 or 1. Midnode computations will be active when ISET is set to -1 or 1. Midnode data will be read when ISET=1; otherwise, midnode data, will be generated by the flow solver when ISET = -1.
- JSET: Overset Flag This parameter controls the overset computations. The overset function will be active whne this parameter is set to not zero. The overset data will be read at every " JSET " step, will be generated by the flow solver at every " -JSET " step.
- OMEGA: Angular Momentum

Settings for Physical Properties

- VISCM: Molecular Viscosity (m^2/s)
- SIGMA: Cavitation Number This parameter is not used.
- RHOF: Density of Fluid (kg/m^3)
- RHOS: Density of Solid (kg/m^3)
- CONDF: Heat Conductivity of Fluid (W/m/K)
- CONDS: Heat Conductivity of Solid (W/m/K)
- CPF: Specific Heat of Fluid (J/kg/K)
- CPS: Specific Heat of Solid (J/kg/K)

Settings for Time Integration and Matrix Solver

- ISTART: Restart Flag The flow solver les3x will read initial flow data when this parameter is set to 1.
- NTIME: Number of Time Steps in Unsteady Computations
- DT: Time Increment

- NMAXSE: Maxim Number of Steps in Steady Computations
- NMAXSI: Number of Inner Iterations in SIMPLE(C) Methods
- NMAXT: Maximum Number of Iterations for Transfer Equation This parameter is applied to momentum equations and transfer equations of temperature and variables for RANS.
- NMAXP: Maximum Number of Pressure Equation
- EPST: Criteria of Convergence for Transfer Equations
- EPSP: Criteria of Convergence for Pressure Equations
- EPSS: Criteria of Convergence for SIMPLE(C) Method
- TFINAL: Parameter for Time Relaxation
  All the values listed below are exponentially developed by the following function:

  $$1.0\text{-exp}\ \{\ \text{-TIME/TFINAL}\ \}$$

  where TIME is the current time of the integration and TFINAL is a user epscified parameter used to control development of the flow field. If no further development of the flow field is desired, simply set TFINAL to zero and this function will take a constant value of one. When TFINAL = 0. the divide exception will be internally suppressed.

  – inlet boundary velocities
  – moving wall boundary velocities
  – angular velocity
- UFINAL, VFINAL, WFINAL: Acceleration Terms Added to X, Y, Z
  The following acceleration will be added.

  – UFINAL*exp(-TIME/TFINAL)
  – VFINAL*exp(-TIME/TFINAL)
  – WFINAL*exp(-TIME/TFINAL)

  where UFINAL, VFINAL, and WFINAL are user specified parameters to control development of the flow field. Note that these acceleration terms are consistent with the above development function.

Settings for Optional Output Data

- IOUT: Optional Output Flag This parameter controls optional output as follows.

IOUT	Flow Files to be Output
0	final flow file
1	final flow and time averaged flow file
2	final flow, time averaged, RMS and stress flow file

- INTFSV: Interval of Flow File Output
  An unsteady flow file will be output every INTFSV step. Unsteady flow file will not be output when INTFSV is set to 0.

- INTSFV: Interval of Surface Pressure File Output
  An unsteady surface pressure file will be output every INTPSV step. Unsteady surface pressure file will not be output when INTPSV is set to 0.

- NSMPL: Number of Sampling Data

- LSMPL: Type of Sampling Data

- XSMPL, YSMPL, ZSMPL: Coordinates of Sampling Data

LSMPL	Type of Variable
1	Velocity-U
2	Velocity-V
3	Velocity-W
4	Pressure
5	Liquid Fraction
6	Temprature
7	Turburlent Viscosity
8	Kinetic Energy (RANS)
9	Turbulent Dissipation

Table 3.2: Format of Parameter File(PARMLES3X)

FILE HEADER				
COMMENT				
ITRANS	IMODEL	IFORM	IPRESS	FSMACH
COMMENT				
D000	U000	T000	TREF	
COMMENT				
IHEAT	ISOLID	ICAV	IBUSNQ	
COMMENT				
NFRAME	ISET	JSET	OMEGA	
COMMENT				
VISCM	SIGMA	RHOF	RHOS	
COMMENT				
CONDF	CONDS	CPF	CPS	
COMMENT				
ISTART	NTIME	DT	NMAXSE	NMAXSI
COMMENT				
NMAXT	NMAXP	EPST	EPSP	EPSP
COMMENT				
IOUT	INTFSV	INTSFV		
COMMENT				
NSMPL				
LSMPL	XSMPL	YSMPL	ZSMPL	*(NSMPL lines)
COMMENT				
FILEMS	(FILE NAME TO READ MESH DATA)			
COMMENT				
FILEBC	(FILE NAME TO READ BOUNDARY CONDITION)			
COMMENT				
FILEIF	(FILE NAME TO READ INITIAL FLOW FIELD)			
COMMENT				
FILEFF	(FILE NAME TO WRITE FINAL FLOW FIELD)			
COMMENT				
FILEHS	(FILE NAME TO WRITE TIME HISTORIES)			
COMMENT				
FILEAT	(FILE NAME TO READ ELEMENT ATTRIBUTE)			
COMMENT				
FILEMN	(FILE NAME TO READ MID-NODE DATA)			
COMMENT				
FILEOS	(FILE NAME TO READ OVERSET DAT)			
COMMENT				
FILEAV	(FILE NAME TO WRITE AVERAGE FIELD)			
COMMENT				
FILERM	(FILE NAME TO WRITE RMS FIELD)			
COMMENT				
FILEST	(FILE NAME TO WRITE STRESS FIELD)			
COMMENT				
FILEFS	(FILE NAME TO WRITE CURRENT FIELD)			
COMMENT				
FILEPS	(FILE NAME TO WRITE SURFACE PRESSURE)			

Table 3.3: List of arameters used in parameter file PARMLES3X (1/2)

Name	Contents	Value
ITRANS	Unsteady/Steady Flag	0, 1, 2
IMODEL	Turbulence Model Flag	0, 1, 2, 12, 13
IFORM	Scheme for Momentum Equation	1, 2, 3, 4
IPRESS	Scheme for Pressure Equation	1, 2
FSMACH	Mach Number	$0.01 \sim 0.1$
D000	Characteristic Length	$\neq 0$
U000	Characteristic Velocity	$\neq 0$
T000	Temperature Difference	$\neq 0$
TREF	Characteristic Temprature	> 0
IHEAT	Heat Transfer Flag	0,1
ISOLID	Solid Greatment Flag	0,1
ICAV	Cavitation Model Flag	0,1
IBUSNQ	Boussinesq approximation Flag	0,1
NFRAME	Multi-Frame Function Flag	-1, 0, 1
ISET	Midnode Flag	0, 1
JSET	Overset Flag	0, 1
OMEGA	Angular Momentum	REAL
VISCM	Molecular Viscosity (m^2/s)	REAL
SIGMA	Cavitation Number	REAL
RHOF	Density of Fluid (kg/m^3)	REAL
RHOS	Density of Solid (kg/m^3)	REAL
CONDF	Heat Conductivity of Fluid (W/m/K)	REAL
CONDS	Heat Conductivity of Solid (W/m/K)	REAL
CPF	Specific Heat of Fluid (J/kg/K)	REAL
CPS	Specific Heat of Solid (J/kg/K)	REAL
ISTART	Restart Flag	0, 1, 2
NTIME	Number of Time Steps in Unsteady Computations	INTEGER
DT	Time Increment	REAL
NMAXSE	Maxim number of Steps in Steady Computations	INTEGER
NMAXSI	Number of Inner Iterations of SIMPLE(C) Methods	INTEGER(>0)
NMAXT	Maximum Number of Iterations for Transfer Equation	$10 \sim 50$
NMAXP	Maximum Number of Pressure Equation	$50 \sim 200$
EPST	Criteria of convergence for Transfer Equations	1.0E-8 $\sim$ 1.0e-4
EPSP	Criteria of convergence for Pressure Equations	1.0E-8 $\sim$ 1.0e-4
EPSS	Criteria of convergence for SIMPLE(C) Method	0 $\sim$ 1.0e-6
TFINAL	Parameter for Time Relaxation	
UFINAL	Acceleration Terms Added to X Direction	
VFINAL	Acceleration Terms Added to Y Direction	
WFINAL	Acceleration Terms Added to Z Direction	
IOUT	Optional Output Flag	0, 1, 2
INTFSV	Interval of Flow File Output	0, 1, 2 $\sim$ N
INTSFV	Interval of Surface Pressure File Output	0, 1, 2 $\sim$ N
NSMPL	Number of Sampling Data	
LSMPL	Type of Sampling Data	$1 \sim 9$
XSMPL	X-Coordinates of Sampling Data	
YSMPL	Y-Coordinates of Sampling Data	
ZSMPL	Z-Coordinates of Sampling Data	

Table 3.4: List of arameters used in parameter file PARMLES3X (1/2)

名前	内容	(IN/OUT)	Condition
FILEMS	File Name of Mesh File	IN	Default
FILEBC	File Name of B.C. File	IN	Default
FILEIF	File Name of Initial Flow File	IN	Active when ISTART=1
FILEFF	File Name of Final Flow File	OUT	Default
FILEHS	File Name of History File	OUT	Default
FILEAT	File Name of Attribute File	IN	Active when NRAME $\neq$0 or ISOLOD=1
FILEMN	File Name of Midnode File	IN	Active when ISET $\geq$1
FILEOS	File Name of Overset File	IN	Active when JSET $\geq$1
FILEAV	File Name of Time Avergaed Flow File	OUT	Active when IOUT $\geq$1
FILERM	File Name of RMS Flow File	OUT	Active when IOUT=2
FILEST	File Name of Stress Flow File	OUT	Active when IOUT=2
FILEFS	File Name of Unsteady Flow File	OUT	Active when INTFSV$\geq$1
FILEPS	File Name of Unsteady Surface Pressure File	OUT	Active when INTPSV$\geq$1

3.2 GF format

A GF file starts with "file_type_header", and ends with W#ENDFILEW.
The file type is specified by setting "file_type_header" to either "#A_GF_V1" or "#U_GF_V1". When "#A_GF_V1" is used, the file format is in ASCII mode; otherwise when "#U_GF_V1" is used, the file format is in FORTRAN UNFORMATTED mode. There are some datasets between 'file_type_header' and '#ENDFILE'.
Possible datasets include "#NEW_SET", comment, and some DATA.
The format of DATA is

- "array_type_headder"
  "#FLT_ARY" or "#INT_ARY", which indicates array type.
- key word
  A character string identifying the data
- comment
  A character string identifying the data name
- element number of array
  The first and the scond numbers of a 2D-array are indicated by num2 and num.
- 2D-ARRY(num2,num)

Mainly the following DATA type are used in the flow slover "les3x".

- Mesh data file
- Boundary data file
- Flow data file
- Overset data file
- Attribute data file
- Domain Decompose Ddesctiption (DDD) file
- History data file

A summary of the GF data format is shown in Figure 3.3. The dataset format and details regarding each are explained in a later section.

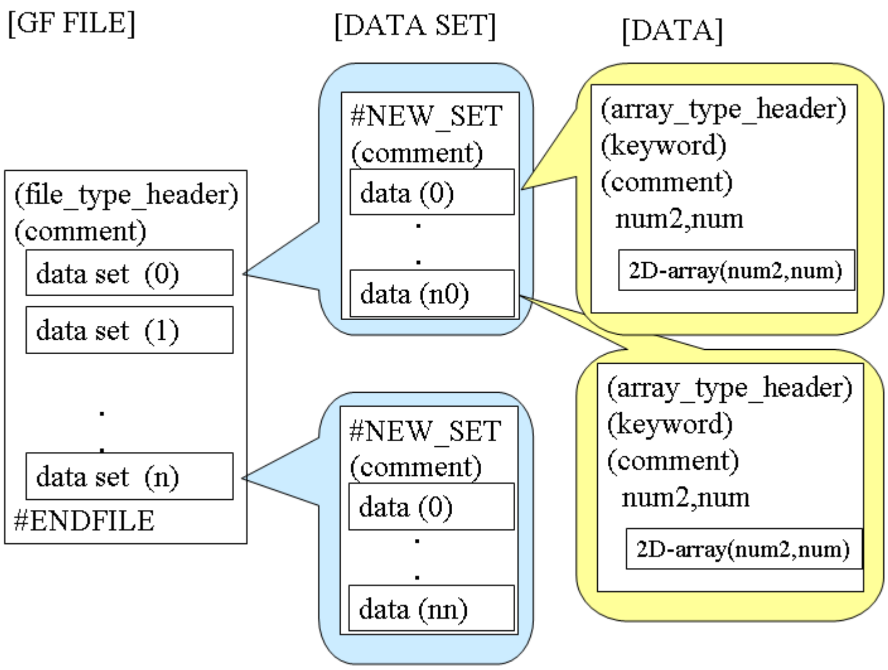

Figure 3.3: GF data format

3.2.1 Mesh Data File

The FFB system assumes hexahedron finite elements which have eight nodes. The GF mesh file includes a node table and node coordinates for each direction. The data format is shown in Table 3.5.

Table 3.5: Mesh data format

Type	Keyword	num	num2	Contents
#FLT_ARY	*GRID_2D	2	NP	x-coordinate of nodes
				y-coordinate of nodes
#FLT_ARY	*GRID_3D	3	NP	x-coordinate of nodes
				y-coordinate of nodes
				z-coordinate of nodes
#INT_ARY	*NODE_2D	4	NE	node table (2D)
#INT_ARY	*NODE_3D	8	NE	node table (3D)
#INT_ARY	*ELM_TYP	1	NE	List specifying element

Note that NP indicates the number of nodes and NE indicate the number of elements.

The GF mesh data file supports both 2D and 3D mesh data. Two dimensional (three-dimensional) mesh data can be handled using the keywords *GRID_2D and *NODE_2D (*GRID_3D and *NODE_3D). You will usually use keywords *GRID_3D and *NODE_3D, because FFB assumes 3D flows. The list specifying element type specifying z(*ELM_TYP) does not have to be specified in the current version.

3.2.2 Boundary Data File

The GF boundary data file supports inlet connect boundary, wall connect boundary, symmetric connect boundary, cyclic connect boundary, body surface connect boundary, free boundary, moving Wall connect boundary, and interconnect boundary.

These boundary conditions are as follows:

- Inlet Boundary
  The inlet velocity (U,V,(W)) is specified at each inlet boundary node.
- Wall Boundary
  A no-slip condition is implemented for turbulence computations.
- Symmetric Boundary
  A symmetric condition is assumed at each symmetric boundary node.
- Cyclic Boundary
  A cyclic condition is assumed at a pair of cyclic boundary nodes. The pressure difference for the cyclic boundary will be controlled (thus it need not be specified) so as to maintain the given flow rate, except when the flow rate is set to zero. In this case, a given constant pressure difference is applied to the cyclic boundary.
- Body Surface Boundary
  The fluid force is calculated at a body surface boundary. Wall shear components of the body force are directly calculated based on the local total viscosity and local velocity gradient without taking condiering the wall velocities or the wall function used for momentum integration.
- Free Boundary
  Traction free and zero pressure are aussumed for the free boudary.
- Moving Wall Boundary
  A moving aall condition is assumed at each moving wall boundary node. The current program assumes that the direction in which a wall moves is parallel to the local wall surface. When this assumption does not hold, the solution is not guaranteed.
- InterConnect Boundary
  An interconnect boundary condition includes the data that each domain needs for communication in parallel execution.

A summary of the GF boundary data format is shown in table 3.2.2.
Note that 'NP***' indicates the node number of each boundary condition.

Table 3.6: Boundary data format

Type	Keyword	num	num2	Contents
#INT_ARY	*BC_INLT	1	NPINLT	inlet boundary nodes
#FLT_ARY	*BC_IV2D	2	NPINLT	inlet boundary velocities(U)
				inlet boundary velocities(V)
#FLT_ARY	*BC_IV3D	3	NPINLT	inlet boundary velocities(U)
				inlet boundary velocities(V)
				inlet boundary velocities(W)
#INT_ARY	*BC_WALL	1	NPWALL	wall boundary nodes
#INT_ARY	*BC_SYMT	1	NPSYMT	symmetric boundary nodes
#INT_ARY	*BC_CYCL	1	NPCYCL	cyclic boundary nodes
#FLT_ARY	*BC_FACT	1	NEDAMP	eddy viscosity damping coefficient
#INT_ARY	*BC_BODY	1	NPBODY	body surface nodes
#INT_ARY	*BC_FREE	1	NPFREE	free boundary nodes
#INT_ARY	*BC_MWAL	1	NPMWAL	moving wall boundary nodes
#FLT_ARY	*BC_WV2D	1	NPMWAL	moving wall boundary u-velocities
				moving wall boundary v-velocities
#FLT_ARY	*BC_WV3D	1	NPMWAL	moving wall boundary u-velocities
				moving wall boundary v-velocities
				moving wall boundary w-velocities
#INT_ARY	*BC_INTR	3	NPINTR	inter-connect boundary nodes
				corresponding sub-domain numbers
				node number in the corresponding sub-domains
#INT_ARY	*BC_CNER	3	NPCNER	corner boundary node

3.2.3 Flow Data File

The FFB system assumes an eighte finite element with linear velocity interpolation and constant pressure at each element. The GF flow data file includes time, time step, velocities at nodes, and pressure at elements. A Summary of the GF flow data format is shown in table 3.7.

Table 3.7: Flow data format

Type	Keyword	num	num2	Contents
#FLT_ARY	*TIME_PS	1	1	time
#INT_ARY	*STEP_PS	1	1	step number
#FLT_ARY	*VELO_2D	2	NP	U-velocity at nodes
				V-velocity at nodes
#FLT_ARY	*VELO_3D	3	NP	U-velocity at nodes
				V-velocity at nodes
				W-velocity at nodes
#FLT_ARY	*PRES_2E	1	NE	element pressure
#FLT_ARY	*PRES_3E	1	NE	element pressure

Note that NP indicates the number of nodes and NE indicates the number of elements.

The GF flow data file supports both 2D and 3D data. Two-dimensional (three-dimensional) flow data can be handled using keywords *VELO_2D and *PRES_2E (*VELO_3D and *PRES_3E). You will usually use keywords *VELO_3D and *PRES_3E, because FFB assumes 3D flows.

The flow solver les3x will output the final flow field (velocities and pressure) to the GF flow data, which is the default output file. As optional output files, the flow solver les3x will output the following data:

- GF average file
  Time averaged velocities and pressure
- GF RMS file
  RMS of velocities and pressure
- GF stress file
  Shear stress of velocity, namely $((vw)', (wu)', (uv)')$, and time-averaged viscosity. Note that the viscosity is the sum of molecular viscosity and eddy viscosity calculated at each element.

Note that the data formats for the above files are the same as that for the GF flow file. The GF-average-file will be output when IOUT ≥ 1 in the parameter file. The GF-RMS-file and the GF-stress-file will be output when IOUT $= 2$ in the parameter file.

A summary of the GF-flow type data format is shown in table 3.8.

Table 3.8: Flow-type data format

File	Node			Element
GF-flow-file	U	V	W	P
GF-average-file	ave. of U	ave. of V	ave. of W	ave. of P
GF-RMS-file	rms of U	rms of V	rms of W	rms of P
GF-stress-file	$(vw)'$	$(wu)'$	$(uv)'$	ave. of viscosity

3.2.4 Overset Data File

When overset calculations are performed, the GF overset data file will be read in by les3x, Actually, when JSET > 1, les3x will obtain the information; necessary for overset calculation, from the GF overset data file at every JSET step. See 3.1 for details regarding the parameter file.

A summary of the GF overset data file is shown in table 3.9.

Table 3.9: Overset data format

Type	Keyword	num	num2	Contents
#FLT_ARY	*TIME_PS	1	1	time
#INT_ARY	*STEP_PS	1	1	step number
#INT_ARY	*BC_PSET	3	NBP	overset boundary nodes
				element number to calculate overset boundary
				domain number to send/receive overset value
#FLT_ARY	*BC_PGET	3	NBP	gxi -coordinate in interpolating element
				zeta-coordinate in interpolating element
				eta -coordinate in interpolating element
#INT_ARY	*BC_ESET	3	NBE	overset elements
				element number to calculate overset boundary
				domain number to send/receive overset value
#FLT_ARY	*BC_EGET	3	NBE	local gxi -coordinate in interpolating element
				local zeta-coordinate in interpolating element
				local eta -coordinate in interpolating element

Note that NBP and NBE indicate numbers of nodes and elements, respectively, which are overset.

3.2.5 Attribute Data File

The GF attribute file contains element frame attribution. This file will be read by les3d, when multi-frame is assumed. Actually, les3x will read this file if NFRAME is set to non-zero. See 3.1 for details regarding the parameter file.

A summary of GF attribution data file is shown in table 3.10.

Table 3.10: Attribute data format

Type	Keyword	num	num2	Contents
#INT_ARY	*ELM_ATRE	1	NE	element frame attributes

The following element numbers are currently available:

frame number(iframe)	type of reference frame
-1	rotating frame (default)
0	stationary frame
1, 2,..., NFRAME	translating frames

See "NFRAME" in parameter file in section 3.1 for details. Note that NE indicates the number of elements.

3.2.6 Domain Decompose Description (DDD) File

The "DDD" file provides a description of domain decompose description. The GF DDD file is used to decompose and unify the GF mesh, boundary, overset, and attribute files for parallel execution mode.

A summary of the GF DDD file is shown in table 3.11.

Table 3.11: Domain decomposing description data format

Type	Keyword	num	num2	Contents
#INT_ARY	*PT_NODE	1	NBP	global node number of local node
#INT_ARY	*PT_ELEM	1	NBE	global element number of local element
#INT_ARY	*BC_INTR	3	NBP	inter-connect boundary nodes
				corresponding sub-domain numbers
				node number in the corresponding sub-domain

Note that

- NBP indicates the number of local nodes and NBE indicates the number of local elements
- The number of domains and the number of datasetT in the DDD file must be the same.

3.2.7 History Data File

The GF history file is a default output file of les3x. This file includes 21 kinds of default output data. Optional output data can be specified by the user. The data format of the GF history file by les3x is shown in the table 3.12.

Note that

- NSMPL indicates the number of optional sampling data and NTIME indicates the number of total time steps. These parameters can be specified in the parameter file. See 3.1 for details.
- The GF history file can include velocities (U,V,W) and pressure at the points you specify as optional output data. See NSMPL,LSMPL in section 3.1 for details.
- Fluid force will be calculated around the surface that is specified by the body boundary nodes. Hence, if the body boundary is not specified, the fluid force will be zero. See 3.2.2 for details.

Table 3.12: History data format of les3x

type	keyword	num	num2	contents
#FLT_ARY	*HISTORY	21 + NSAMPL	NTIME	TIME
				time
				maximum divergent
				average element eddy viscosity
				iterations done for pressure equation
				l2-norm residual of pressure equation
				fluid force acting in x direction
				fluid force acting in y direction
				fluid force acting in z direction
				iterations done for u-equation
				iterations done for v-equation
				iterations done for w-equation
				iterations done for t-equation
				iterations done for k-equation
				iterations done for e-equation
				l2-norm residual of u-equation
				l2-norm residual of v-equation
				l2-norm residual of w-equation
				l2-norm residual of t-equation
				l2-norm residual of k-equation
				l2-norm residual of e-equationv
				volume integration of tempratur
				(user specified infomation)