ADVENTURE_sFlow

Stationary Navier-Stokes Solver with HDDM
Version 0.11 (beta)

User's manual

April 1, 2007
ADVENTURE Project

Contents

1 To Start

ADVENTURE_sFlow is a finite element solver developed in the ADVENTURE Project [1] to perform the stationary incompressible viscous flow problem analysis using the Hierarchical Domain Decomposition Method (HDDM) for parallel processing.

The first chapter contains an outline of ADVENTURE_sFlow and some explanations about the execution of operating procedures. Program analysis functions are introduced in Chapter 2.

1.1 Module Features

ADVENTURE_sFlow has the following features.

- Parallel processing which performs load-balancing using Hierarchical Domain Decomposition Method (HDDM) can be obtained.
- A stationary problem can be analyzed.
- Using iterative methods as BiCGSTAB method, GP-BiCG method, and BiCGSTAB2 method.
- Supporting linear tetrahedral elements.
- A stabilized finite element method is used.

1.2 Operational Environment

ADVENTURE_sFlow operates in the following environment.

Operating Systems: UNIX, Linux
Parallel processing library: MPI

MPICH and LAM/MPI are the most commonly used MPI libraries. In recent Linux distributer, either library may be pre-installed but otherwise, you need to install it.

· MPICH destination:
http://www-unix.mcs.anl.gov/mpi/mpich/
· LAM destination:
http://www.lam-mpi.org/

1.3 Compilation and Installation

1.3.1 Compilation

The C compiler, the MPI environment and the ADVENTURE_IO module are necessary to compile ADVENTURE_sFlow module.
The following procedure is accepted for the compilation of ADVENTURE_sFlow.

(1) Extraction of the archive file.

gunzip -c advsflow0.11beta.tar.gz | tar xvf –

(2) After moving to the extracted directory, the compilation can be done.

Makefile.in. is edited in that directory.
The following macros are matched to the environment.

ADVSYS_DIR	Top directory of ADVENTURE
ADVIO_CONFIG	Full path to script advsys-config of ADVENTURE_IO
MPI_CC	MPI C compiler
MPI_LINKER	MPI C link
CC	C compiler
LINKER	C link
CFLAGS	Optimization options

Afterwards, execute the following command:

make

1.3.2 Installation

If the compilation is accomplished successful, the installation will be performed with the following command.

make install

This procedure has to be done by the user, who can write files to the installation directory. To set the target installation directory to *install_dir*, the following command should be executed.

make install prefix=*install_dir*

However, install_dir should specify the absolute path of the installed directory.

1.4 Execution of the programs

There are two versions of the ADVENTURE_sFlow executable module :

- **advsflow-p** version for static load distribution of parallel processing
- **advsflow-h** version for dynamic load distribution of parallel processing

Depending on the environment, one these modules should be used when performing an analysis. Refer to the paragraph 2.1 for detailed information about each module.
Though MPI is used for parallel version, the execution procedures will be described with MPICH.

There are many types of MPI implementations. Compilation & execution methods for each kind will be different. Therefore you should refer to the technical manual, in order to know how to change your method (compilation & execution) into the appropriate way.

mpirun [*options for mpirun*] **advsflow-p** [*options*] *data_dir*

Or

mpirun [*options for mpirun*] **advsflow-h** [*options*] *data_dir*

[*options for mpirun*]
Among a number of options for MPICH, the following options are frequently used. Refer to the MPICH manual for details.

- **-np** *number_of_hosts*
  The option is used to specify the number of MPI processor to be involved
- **machinefile** *machine_file*
  The list of machine name which perform parallel computing is specified. Otherwise, the default file set with the system is used.

[*options*]

The options of ADVENTURE_sFlow are basically the same for both execution modules. These options are used for selection of analysis types and other output parameters. Detailed information is described later.

2 Function of Parallel Processing and Functions in Nonlinear analysis

2.1 Function of Parallel Processing

In ADVENTURE_sFlow, Parallel Processing using the Hierarchical Domain Decomposition Method is possible. Fig.1 shows the hierarchical domain decomposition of the entire model. Large decomposed unit of the first hierarchy level will be referred as "Part", and smaller units of the decomposed "Part" (2^{nd} hierarchy level) will be referred as "Subdomains". Domain Decomposition have to be performed by ADVENTURE_Metis prior to operation of ADVENTURE_sFlow.

MPI is used as parallel libraries for ADVENTURE_sFlow and at the starting time of execution, two or more processes are launched according to the user's specifications (environment settings). Commonly, one process is assigned for one CPU, however a number of processes can also be assigned to one CPU. Here, the terms "node", "process", and "CPU" are used for convenience without any special distinctions.

Moreover, two binary executions are prepared in Parallel Processing. The parallel processing methods that can be used are described below :

(1) Static load distribution version (advsflow-p)

As it is shown in Fig. 4, the calculations are performed in parallel and the processes are assigned statically. One process is automatically assigned for one " Part " prior to program execution, and, in this case, to decrease network communications between nodes, the number of " Parts " should be equal to the number of network nodes.

※ Because the volume of network communications is restrained in static load distribution version, this version works efficiently if all nodes have the same performance in the parallel computing system.

(2) Dynamic load distribution version (advsflow -h)

The treatment of "Parts" and "Domains" is accomplished dynamically when the domain-decomposed data prepared by ADVENTURE_Metis are read by ADVENTURE_sFlow and subdivided between CPUs in the way that one "Part" will be assigned for one CPU and the remaining CPUs will be used for treatment of "Domains" (Fig. 5). Here, such sharing will be referred as the "Parent" process and the "Child" process. The "Child" processes do necessary calculations of "Domains" and, later, the "Parent" process handles the calculated results. The number of "Parts" should be less than the total number of CPUs considered for calculation. Since the communications will be concentrated on the "Parent" nodes, the efficiency will be decreased if the number of the "Child" nodes is much larger than the number of the "Parent" nodes. To overcome this problem, the "Parent" nodes are also considered to work in parallel and processing of "Parents" can also be distributed.

Parallel method	command	Number of parts in Decomposition domain
Static load distribution Dynamic load distribution	advsflow-p advsflow-h	Number of parts=number of network nodes Number of parts=number of "parent" process

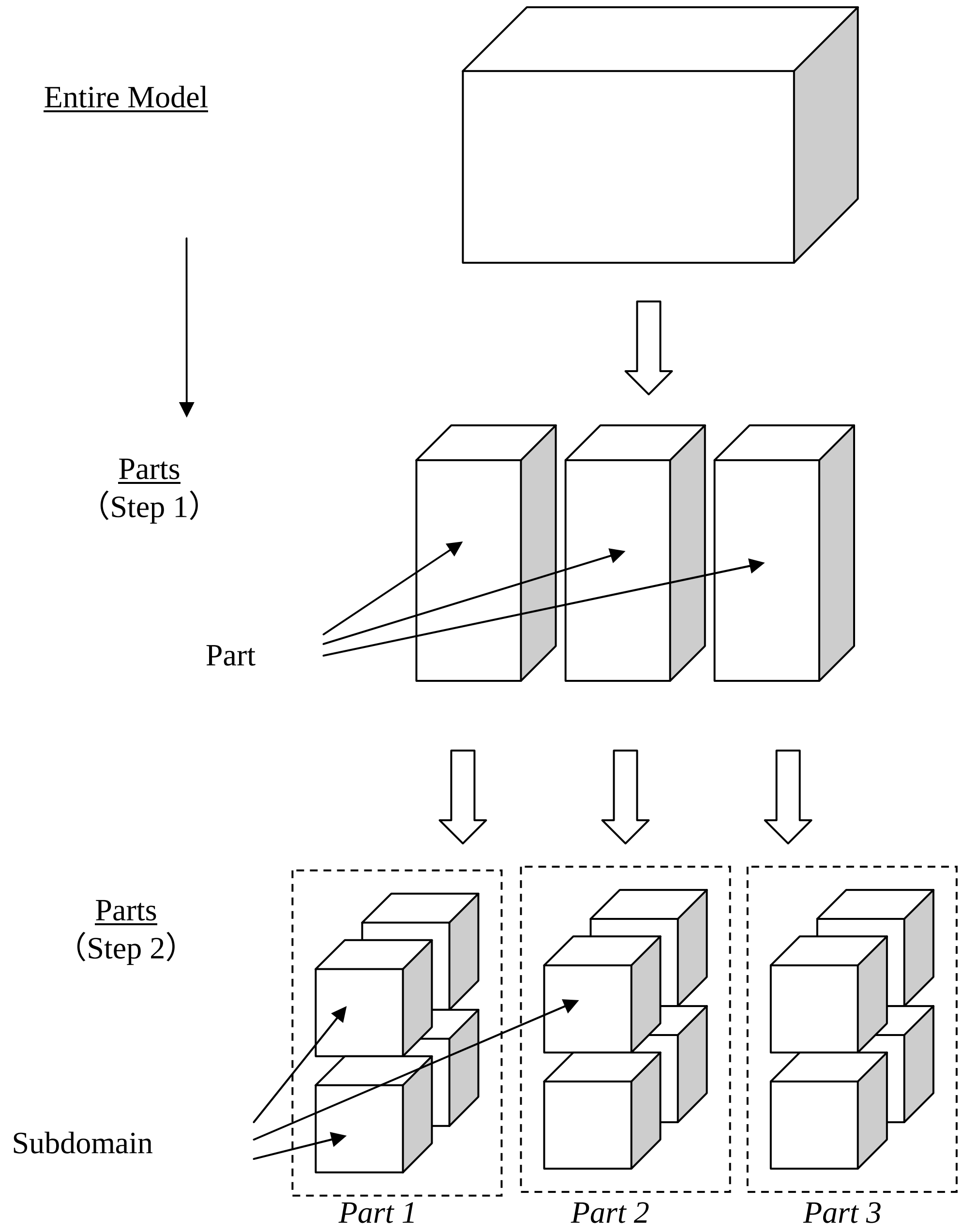

Fig.1 Hierarchical Domain Decomposition of model

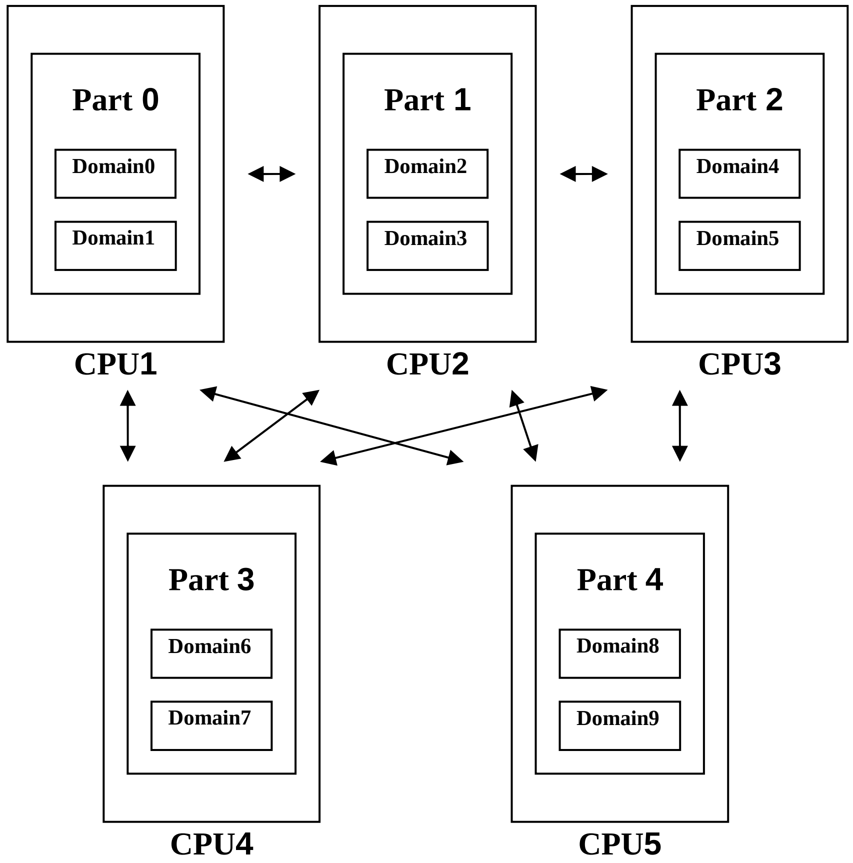

Fig.2 ：Adjustement of Domains to CPUs (static load distribution version)

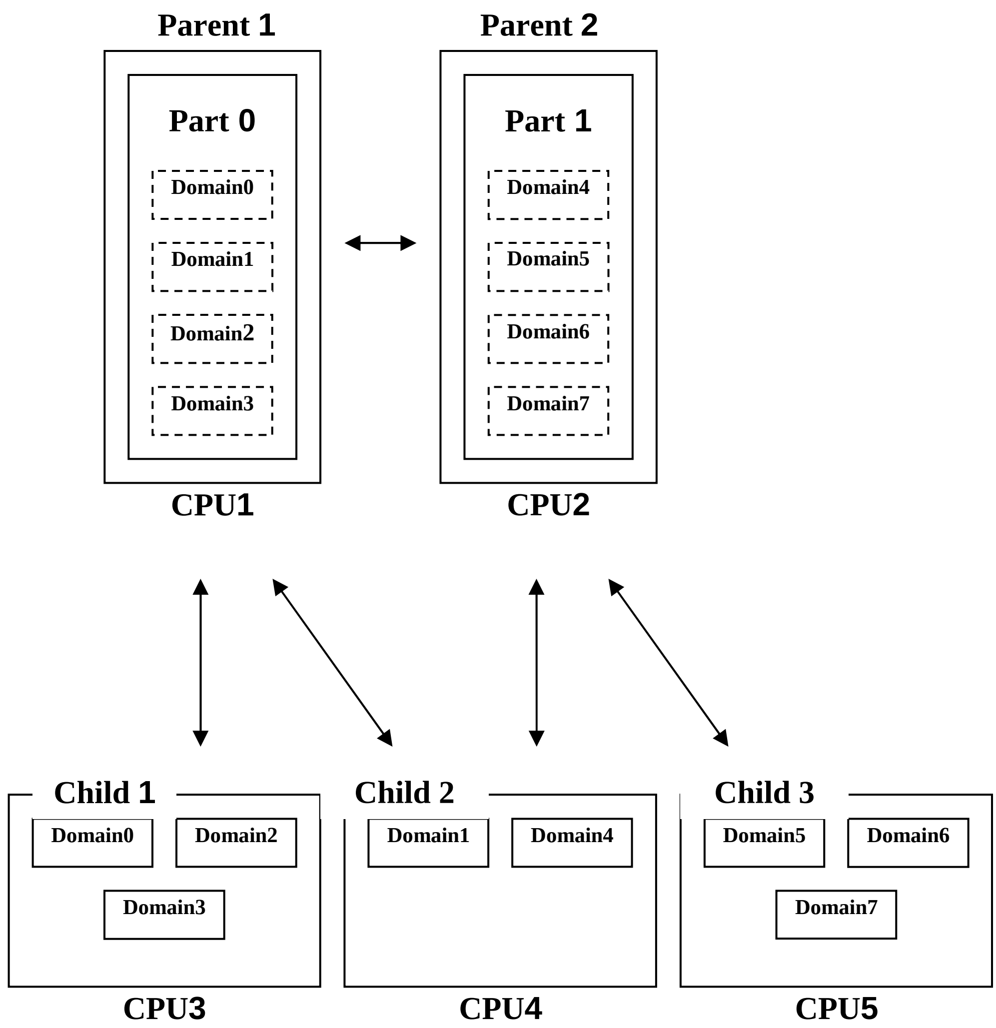

Fig.3 ：Adjustement of Domains to CPUs (Dynamic load distribution version)

2.2 Functions in Nonlinear Analysis

The Newton method is used for nonlinear analysis. The flowchart of processes is presented in Fig.4. The data processing is performed by two large loops. The outside loop is for nonlinear iterations. Iterations used in HDDM are performed by BiCGSTAB method, GP-BiCG method or BiCGSTAB2 method in the inside loop.

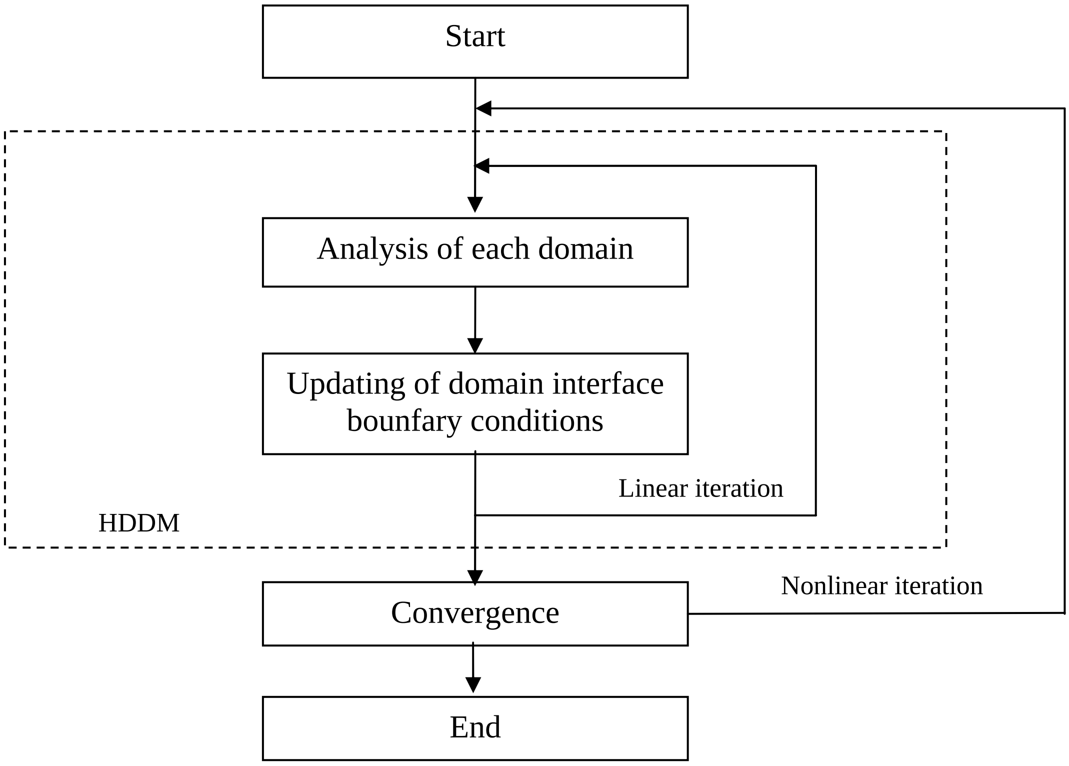

Fig.4 Flowchart of processes for nonlinear analysis

2.3 Domain Decomposition (ADVENTURE_Metis)

If very detailed decomposition by ADVENTURE_Metis is done, some elements might not be included into the domain. When such a domain is found during the execution of ADVENTURE_sFlow, a warning is put out and the execution ends. Moreover, if you do a excessively rough decomposition as regard to the number of total elements, the amount of calculations might increase to such an extent that the memory becomes insufficient. Consequently, it will result in errors and termination of the ADVENTURE_sFlow module.

Good performance depends on an appropriate domain decomposition. Basically, the number of parts should be decided taking into account the method used for parallel processing, the number of nodes used in a network and the computing environment. The number of subdomains should be decided being based on the memory used by computation processes. Obviously, as more detailed domain decomposition is done, as less memory is required. Compared with the static load distribution version, there are much more transfer data between the "parent" and the "child" in the case of dynamic load distribution. Consequently, the static load distribution method is more efficient for uniform computing environment.

※ Furthermore, in order to fix the number of DOF of inner boundary nodes, the option (-difn 4) is used during the execution of ADVENTURE_Metis, as described in section 3.1.

3 Analysis Functions

A stationary problem can be analyzed by using parallel processing. Various analysis can be performed and the accurate functions are described below.

3.1 The Operational Flow

The operational flowchart of ADVENTURE System around ADVENTURE_sFlow is shown in Fig.5.

(1) Preparation of the mesh data (ADVENTURE _TetMesh)
It performs mesh decomposition of the analyzed object

(2) Setup of the boundary conditions (ADVENTURE_BCtool)
Boundary conditions added to the mesh of the analyzed object are set up.

(3) Conversion of the analysis model
fgr_getnode and sflow_makefem (ADVENTURE_sFlow attachements) are used to make the entire-type analysis model file. These functions are used to prepare data for each node which belongs to boundary surfaces and to create the entire-type analysis model analysis file. The detailed operating procedures are described later in the appendix.

(4) Domain Decomposition (ADVENTURE_Metis)
It decomposes the entire-type analysis model. You should add the option −difn 4 when executing it, like in the following example.

mpirun [*mpi options*] **adventure_metis** -difn 4 [*options*] *model_filename directory_name div_num*

By using this option, the number of degrees of freedom (DOF) of inner boundary nodes is set up to 4. In flow problem analysis, the number of DOF of nodes is assumed to be 4 whereas in static solid analysis, the number of DOF of node displacement is 3. This option is indispensable to prepare the input model for ADVENTURE_sFlow.

(5) The *stationary flow problem analysis (ADVENTURE_sFlow)*
The ADVENTURE_sFlow module performs finite element analysis using the decomposed analysis model as an input.

(6) Visualization (ADVENTURE_Visual)
Analysis results can be visualized by using this module.

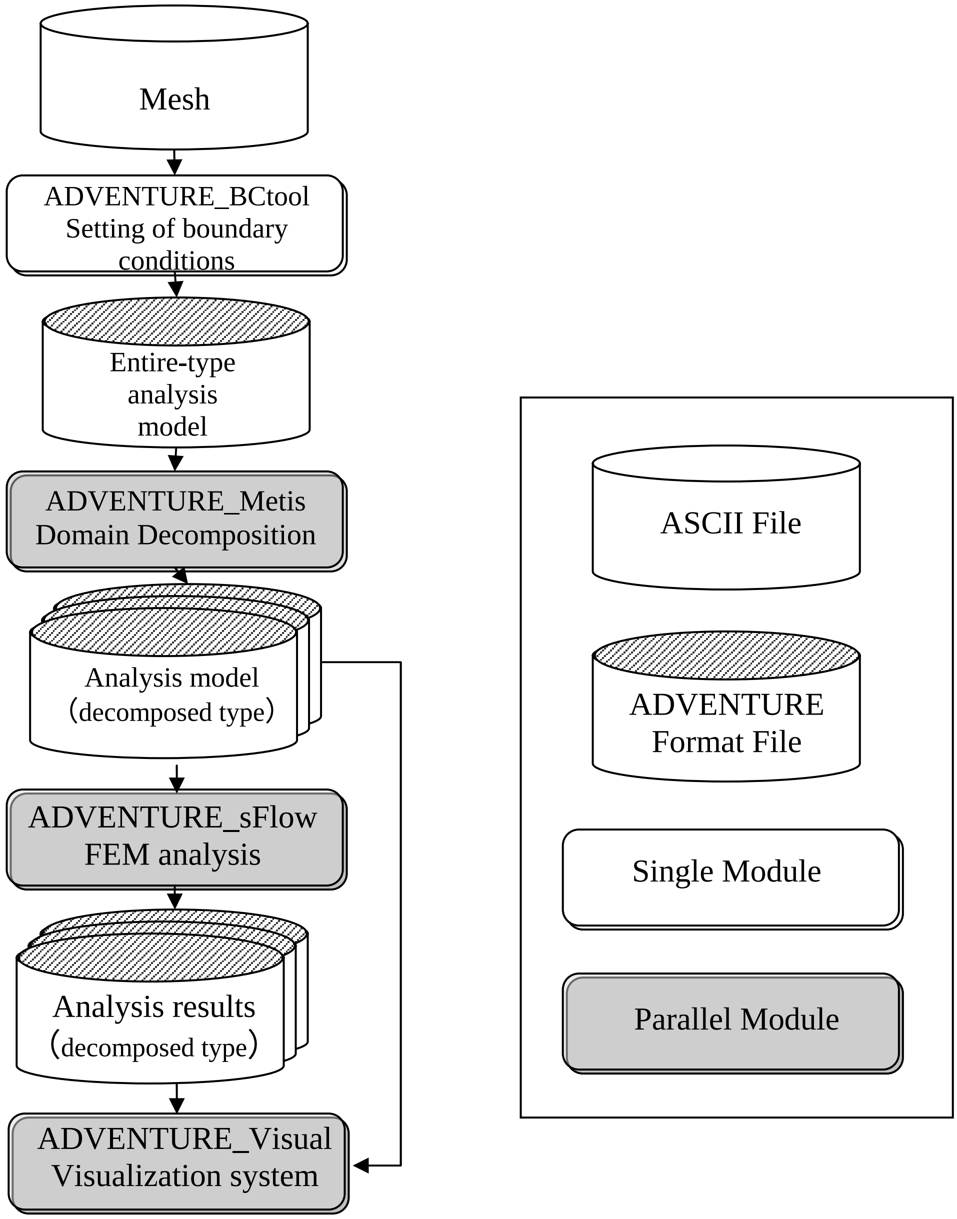

Fig.5 Operational flowchart

3.2 Input and Output Data

The input and output date processing by ADVENTURE_sFlow are shown in Fig.6. Except the output log that can be displayed on the view screen, all data files are saved using the ADVENTURE binary data format. A file is created for each part.

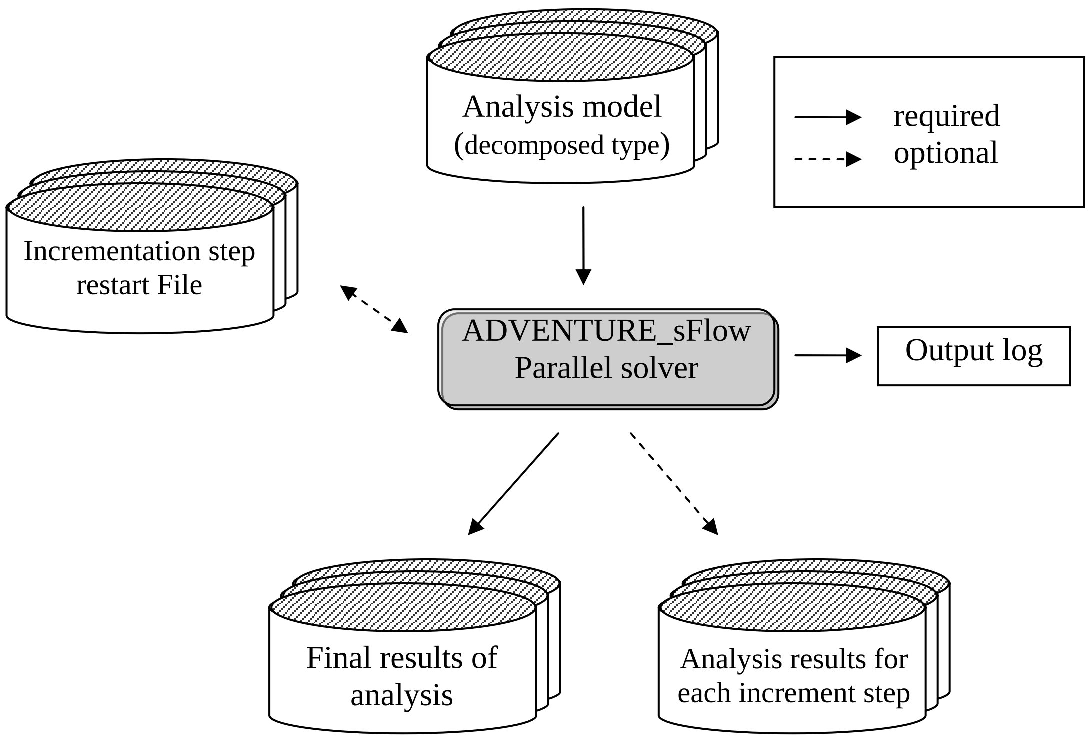

Fig.6 Output and input files

The files containing the decomposed analysis model, previously prepared with ADVENTURE_Metis, are used as input. The output files contain the information on physical quantities of node pressure and velocity. In nonlinear analysis, the output can be done for each increment step.

The restart function is provided for the cases of limited continuation of execution by saving the data into files and restarting the program from the moment of interruption of the calculations. The restart file that can be used includes the nonlinear repetition restart file.

3.3 About the Unit System

In ADVENTURE_sFlow, the values of velocity and pressure normalized by density are used as DOF. Conversion functions of unit system are not included . Thereby, a compatible unit system should be used when the input data are prepared.

3.4 Boundary Conditions

The following boundary conditions can be applied

- Node density specification
- Natural boundary conditions (without surface force)

3.5 Material Properties

The following isotropic material property can be applied :

- viscosity

3.6 Output Analysis Results

The analysis results are printed out into HDDM-type files. The values of pressure and velocity for each node are output.

The analysis model is recorded in the ADVENTURE File format for each part.

4 Execution Method

The parallel version requires MPI for execution. In the case of MPICH, the programs can be executed in a parallel mode using the command mpirun :

mpirun [*options for mpirun*] **advsflow-p** [*options*] *data_dir*

mpirun [*options for mpirun*] **advsflow-h** [*options*] *data_dir*

Here, [options for mpirun] are the command options for mpirun. [options] are the command options used by ADVENTURE_sFlow. These options are used for selection of analysis types and various settings. (see 4.2 for details)

data_dir is the path to the top directory where the input/output data are located. That parameter is indispensable.

4.1 Filenames for Input and Output

The following input/output filenames are set up by default.

- Analysis models file
  data_dir/model/advhddm_in_*P*.adv
- Analysis results file
  data_dir/result/advhddm_out_*P*.adv
- Restart file (File containing the analysis results by increment step during the nonlinear analysis)
  data_dir/result/advhddm_out_*S*_*P*.adv

Here, P is the number of part and S is increment step number for the Newton method.

4.2 Program Options

The following options can be used for program execution.

4.2.1 Specification of stationary analysis

- *-ns*
  The option is used to perform the nonlinear analysis. If this option is not specified, it becomes a Stokes problem. In addition, the following sub-option can be specified behind the option –ns.

- *--ns-tol x*
  The tolerance x to judge about the convergence is specified. This value is the relative variation in the Newton method. The computations are supposed to converge if the relative variation becomes smaller than the tolerant value. The default tolerance is $1.0*10^{-4}$.

- *--step n*
  The maximum number of iterations for the Newton method is specified. The default value is 20.

- *--out-interval n*
  The output of the analysis results will be done for the last step n. By default, no output will be done.

- *--use-resin n*
  The analysis will restart from the n increment step of the Newton method if the output data was recorded in the restart file before.

4.2.2 Control options for iteration methods

In order to solve linear equation, the iterations can be performed by using BiCGSTAB method, GP-BiCG method, or BiCGSTAB2. A few options are used in ADVENTURE_sFlow to control the iterations.

- -solver [bicgstab, gpbicg or bicgstab2]
  BiCGSTAB method, GP-BiCG method, or BiCGSTAB2 method can be used. The default method is BiCGSTAB2.

- -cg-tol x
  The tolerance x to judge about the convergence is specified. The iterations are supposed to converge if the relative residual error is smaller than the residual error obtained at the first step. The default tolerance is $1.0*10^{-6}$.

- -cgloop-max n
  The maximum number of iterations is specified. The default value is 10000.

4.2.3 Options for changing of input and output filenames

- **-model-file** *file*
  The input analysis filename is assigned by *file*. The new extension is _P.adv. The default file is advhddm_in.

- **-model-dir** *dir*
  dir is the name of the sub-directory containing the input analysis files. The default name is model.

- **-result-file** *file*
  The analysis results filename is assigned by *file*. The extension _P.adv is added to the filename. The default filename is advhddm_out.

- **-result-dir** *dir*
  dir is the name of the sub-directory containing the output analysis results files. The default name is result.

- **-ns-resin-file** *file*
  The filename of the input increment step restart file is assigned by *file*. The extension _S_P.adv will be added to the filename automatically. The default name is advhddm_out.

- **-ns-resin-dir** *dir*
  dir is the name of the sub-directory containing input increment step restart file. The default name is result.

4.2.4 Others options

- **-memlimit *n***
  The option specifies the size of memory in Mbytes, which can be used by each process. The program terminates when the value of allocated memory is exceeded. The default value is 1024 (MBytes)

- **-help or-h**
  The main help messages can be displayed.

- **-version or –v**
  The version of the module can be displayed.

- **-help-ns**
  The help messages for nonlinear iteration control options can be displayed.

- **-help-iter**
  The help messages for iteration method control options can be displayed.

4.2.5 Setting by defs.h

- **NORM [0 or 1]**

  The maximum norm of the relative residual error of iteration method is considered by 0. With 1, the 2-norm is chosen. The default norm is the 2-norm.

- **DIAG_SIGN [0 or 1]**

  If the parameter is equal to 0 (which is done by default), pre-processing for diagonal scaling of the iteration is done with signs (+ / -). But, if the indicator (parameter) is equal to 1, pre-processing for diagonal scaling of the iteration is done with absolute value (|.|). BiCGSTAB2 tends to diverge with absolute value calculation (indicator = 1).

Appendix

A. Information on elements

B. Regarding boundary conditions

C. About tools

D. Analysis examples

A. Information on elements

ADVENTURE_sFlow supports linear tetrahedral elements.

A.1 The tetrahedral element

The number of nodes is 4 and the arrangement of the node numbers in the element connectivity is shown in Fig.7.

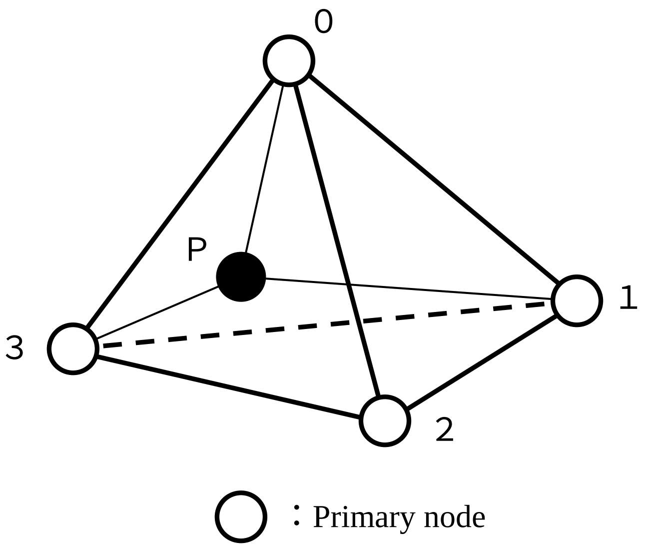

Fig. 7：Linear tetrahedral element

B. Regarding boundary conditions

The boundary condition for ADVENTURE_sFlow can be written in the format below.

B.1 Boundary condition data file

```
[Properties]
1: content_type=FEGenericAttribute
2: num_items=81
3: fega_type=NodeVariable
4: label=DIrichletBC
5: format=i4f8
6: index_byte=4
[Data]
0 0 0.000000e+00
0 1 0.000000e+00
0 2 0.000000e+00
1 0 0.000000e+00
1 1 0.000000e+00
1 2 0.000000e+00
2 0 0.000000e+00
2 1 0.000000e+00
3 2 0.000000e+00
3 0 0.000000e+00
…

58 0 1.000000e+02
59 0 1.000000e+02
60 0 1.000000e+02
...
```

<u>Fig. 8 : Format of boundary conditions file</u>

From left to right, you can read the node number, the direction and either the value of velocity or pressure. However, as each node has 4 degrees of freedom in stationary flow problem analysis, you need to consider the three components of the flow velocity (0, 1, 2) and the pressure component (3).

B.2 Material properties data file (viscosity)

```
[Properties]
1: content_type=FEGenericAttribute
2: num_items=1
3: fega_type=AllElementVariable
4: label=kinematicViscosity
5: format=i4f8f8
6: index_byte=4
[Data]
1.000000e-02
```

Fig. 9 : Format of data file with material properties

In this case, only the dynamic viscosity value is specified.

C. About tools

The following tools are included in ADVENTURE_sFlow archive.

C.1 Entire type analysis model conversion filter : sflow_makefem

sflow_makefem is a tool used for stationary flow problem analysis that converts the analysis condition file (the extension is .cnd) obtained with ADVENTURE_BCtool . The following conversion can concretely be done.

- Displacement Boundary Conditions ⇒ boundary conditions with a given number of DOF

The execution command is:

sflow_makefem <kinematic viscosity> *mshFile datFile cndFile advFile*

Here, *mshFile* is the mesh data prepared by ADVENTURE_Tet_Mesh and *cndFile* is made by ADVENTURE_BCtool.
<kinematic viscosity> specifies the dynamic viscosity. However, though the analysis is possible if you increase the Reynolds number up to 1,000 in the case of 300,000 DOF, you can only consider the Reynolds number up to 50 in the case of about 1 million DOF (an unexpected divergence of linear iterations occurs). This is true only if the Stokes flow is considered as the initial value. The name of the advFile can be specified as it pleases you.

advFile for ADVENTURE_sFlow analysis is completed by executing this command.

C.2 Velocity and pressure components for sflow_hddmmrg

This program **sflow_hddmmrg** is designed to merge the domain-decomposed ADVENTURE format file containing velocity and pressure results data.
The following command is used to execute sflow_hddmmrg :

sflow_hddmmrg [*Pressure or Velocity***]** *directory_for_analysis*

When either the velocity field or the pressure field is requested, the right file is selected. *directory_for_analysis* is the name of the directory where both the analysis results obtained by ADVENTURE_sFlow and the domain-decomposed model done with ADVENTURE_Metis are located.

Pressure.dat and Velocity.dat are obtained by executing this command. The values of the velocity and the pressure of each node can be known. These files are used when executing the data converter introduced in the next chapter.

C.3 Data converter (advsflow_p_rest2ucd)

This tool converts the analysis results into the UCD format of Micro_AVS. With this tool, it's possible to check the velocity fields and the pressure fields of the analysis results by using MicroAVS.

The execution command is :

advsfolw_p_rest2ucd *advFile ucdFile*

advFile is the file obtained by the entire type analysis model conversion filter sflow_makefem.
ucdFile is the name of the output UCD file. You can name the UCD file as it pleases you. However, please remember not to input the extension of the ucdFile and advFile when executing it. Moreover, the output files obtained with sflow_hddmmrg (introduced in the previous paragraph) Pressure.dat and Velocity.dat have to be in the same directory as the advFile.

C.4 Pre-Processing Tools

advmodel is the special pre-processing tool for the lid-driven cavity problem. The analysis model file with the extension .adv is generated
The usage of advmodel is as follows:

advmodel *<kinematic viscosity> <Num_division_per_direction>* *advFile*

- To the first argument, <*kinematic viscosity*>, assign the value of the dynamic viscosity.
- To the second argument, <*Num_division_per_direction*>, assign the number of division in one direction of the lid-driven cubic. For example, when you assign 49, the number of DOF is about 500,000 and when the number of divisions is 62, the number of DOF is roughly 1,000,000.
- To the third argument, *advFile*, assign an arbitrary file name (for example, test.adv)

D Analysis examples

D.1 Example of stationary flow problem analysis

In this part, an analysis example using ADVENTURE_sFlow and ADVENTURE_Metis Domain Decomposition is presented. Each module of ADVENTURE_TriPatch, ADVENTURE_TetMesh and ADVENTURE_BCtool are used to draw up the model.

Below, each procedure, and the command which is used for the execution of this example are shown. Concerning the details of each command, refer to the manual of each system.

(1) *Preparation of IGES files*

First of all, IGES files are prepared by using Business CAD. Please consult the manual of ADVENTURE_TriPatch for entity limitations of the IGES format.

Here, a subway station is considered as an analysis example. Moreover, the file *station.igs* is created.

(2) *Preparation of surface patch*

ADVENTURE_TriPatch makes the surface patch data file from the IGES files created at the first step.

First of all, the node density file should be prepared in a suitable editor as shown in Fig.10.

station.ptn

```
BaseDistance
1.0
```

Fig. 10: Example of a node density control file

With the following command, *station.pcm*, *station.pcg*, *station.wrl* are created.

% ADVENTURE_TriPatch station station

(3) *Preparation of the mesh data*

With the following command, *stationc.pcc, stationc.ptn* are output and stationc.msh is created by the next command.

% advtmesh9p station –d
% advtmesh9m stationc

(4) *Setup of the boundary conditions*

Boundary conditions are added by using ADVENTURE_BCtool.

stationc_3.fgr ， *stationc_3.pch* ， *stationc_3.pcg* are created with the following command

% msh2pch stationc.msh 3

After startup of BcGUI with the following command, a window like Figure 11 will appear on the screen.

% bcGUI stationc_3.pch stationc_3.pcg

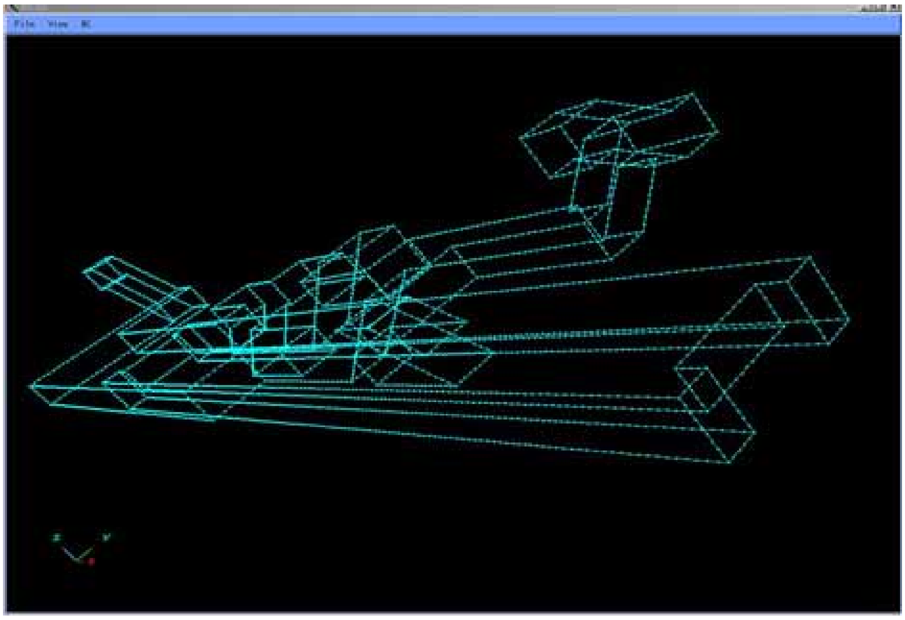

Fig. 11: this window is displayed by bcGUI command

First of all, the speed (it's assumed to be 3m/s here) is set on the green faces of the subway station shown in Fig.12. However, the speed is substituted for Displacement.

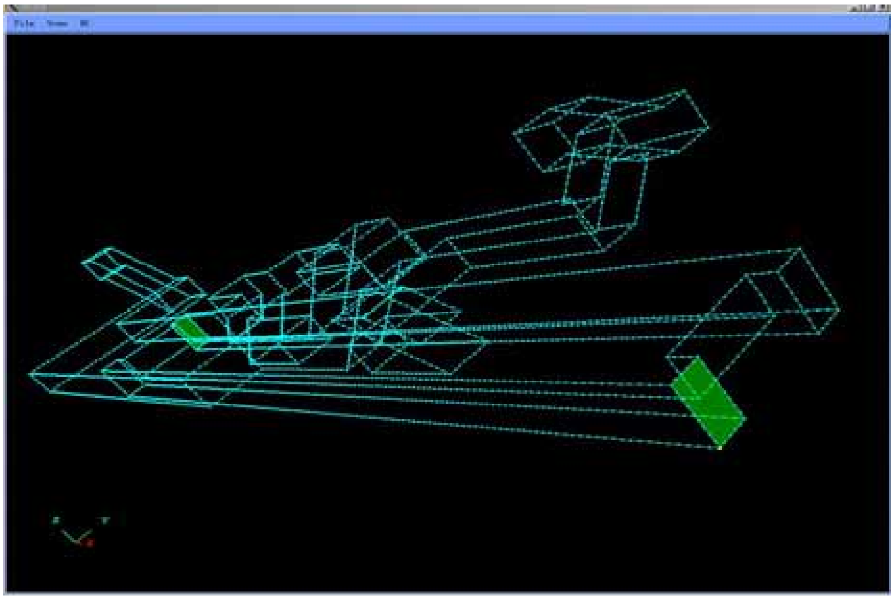

Fig.12: Setting of boundary conditions 1

Then, as it is shown in Fig.13, the speed components x, y and z are set to 0 for each green face.

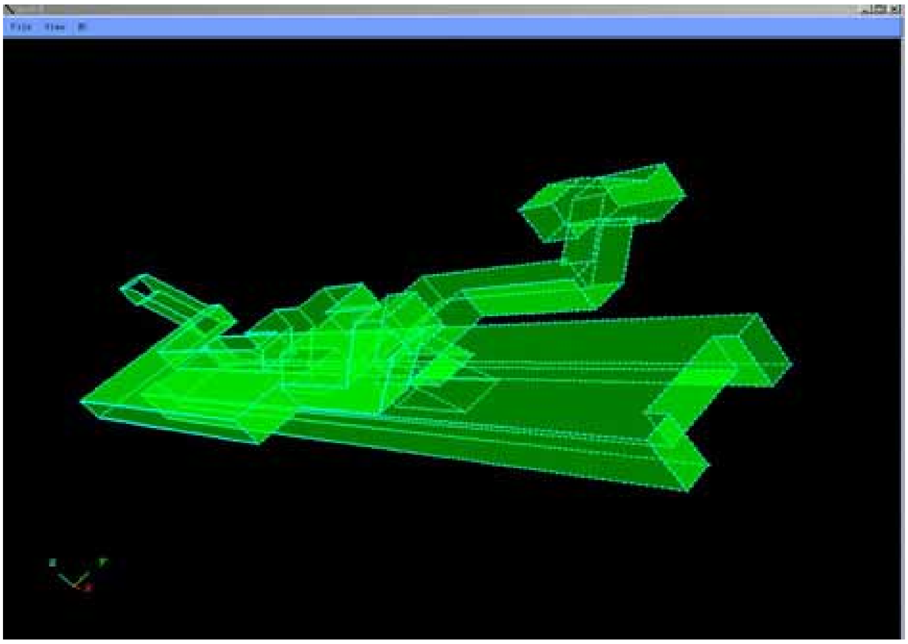

Fig.13: Setting of boundary conditions 2

However, the parts shown in Fig.14 (green faces) are set as the natural boundary.

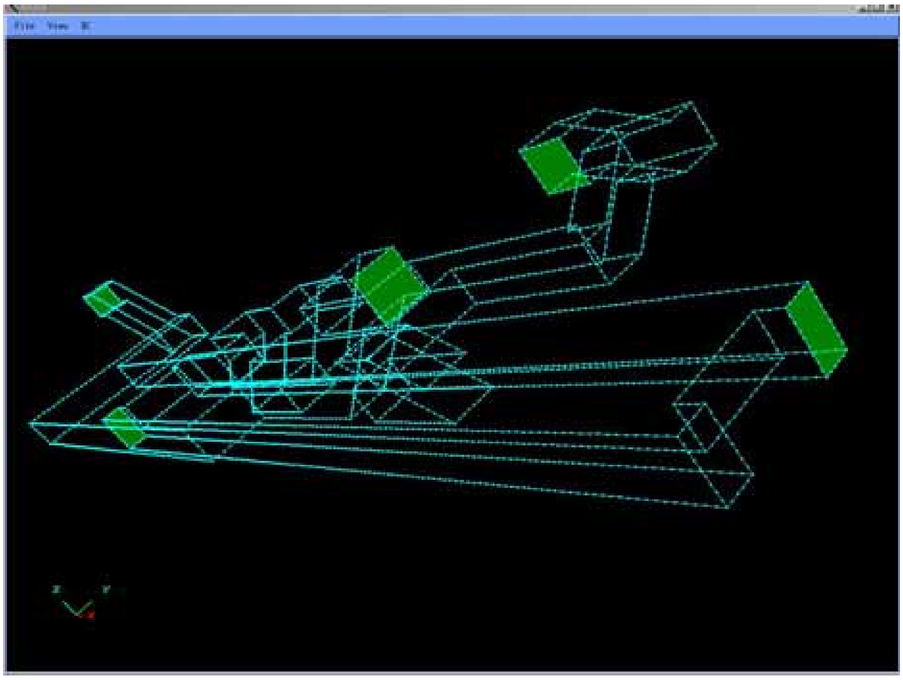

Fig.14: Setting of boundary conditions 3

When all the boundary conditions are set, the file with the extension .cnd "*station.cnd*" is output.

(5) *Preparation of the entire type analysis model*

sflow_makefem is a tool that converts the analysis condition file obtained by ADVENTURE_BCtool for the stationary flow problem analysis.
First of all, the list of the nodes of each face is realized with fgr_getnode. The following command is input. The output file is “*station.dat*”.

% fgr_getnode stationc_3.fgr station.dat

Then, the entire type analysis model is created by the following command. The output file is “*station.adv*”.

% sflow_makefem 0.01 stationc.msh station.dat station.cnd station.adv

(6) *Decomposition of domain*

ADVENTURE_Metis uses the entire-type analysis model as input data to produce the HDDM-type model data. When executing it, you need to use the option –difn 4. Indeed, in stationary flow problem, each node has 4 degrees of freedom.
First of all, the number of parts and the number of subdomains have to be determined to achieve the Hierarchical Domain Decomposition. In this case, we consider that the analysis is performed by using eight PC in the static load distribution version. Therefore, the number of parts is assumed to be eight.
Here, the number of elements of the analysis model is 40,981 and, taking into account that the number of subdomains in a part is 100,

40,981 (number of element) ÷ 8 (number of parts) ÷ 100 (number of subdomains in a part) = 51.227 is assumed to be the number of subdomains in a part. Furthermore, the number of subdomains in the entire model is :

(Part) * (Number of subdomains in a part) = 408.
The decomposition domain is performed in the same way as the following example of the command.

% mpirun -np 8 -machinefile machinefile adventure_metis -difn 4 station.adv out 51

-machinefile is an option for MPI. out is a directory for input and output data.

(7) *Execution of the analysis*

The decomposed analysis model is analyzed as an input by using ADVENTURE_sFlow module. The analysis is performed in the same way as the following example of the command.

% mpirun -np 8 -machinefile machinefile advsflow-p -ns out

The ADVENTURE_sFlow option –ns is required to perform a non-linear analysis.

(8) *Visualization of the analysis results*

We introduce in this section how to visualize the analysis results with MicroAVS. The UCD file for MicroAVS can be prepared by using the command advsflow_p_rest2ucd (ADVENTURE_sFlow attachment).
First of all, pressure and velocity data are taken of the output file of the analysis results. The execution method is as follows.

% sflow_hddmmrg   Pressure out
% sflow_hddmmrg Velocity out

As a result, the values of both velocity and pressure can be read on each node. Then, the UCD file is made with the output data (Pressure.dat, Velocity.dat) and station.adv. An example of the command is shown as follows.

% advsflow_p_rest2ucd station station

The former ADV file and the UCD format file (the output file is station.inp) are obtained. The files, station.adv, Pressure.dat and Velocity.dat have to be in the same directory.

The velocity field is visualized in Fig.15 by using MicroAVS.

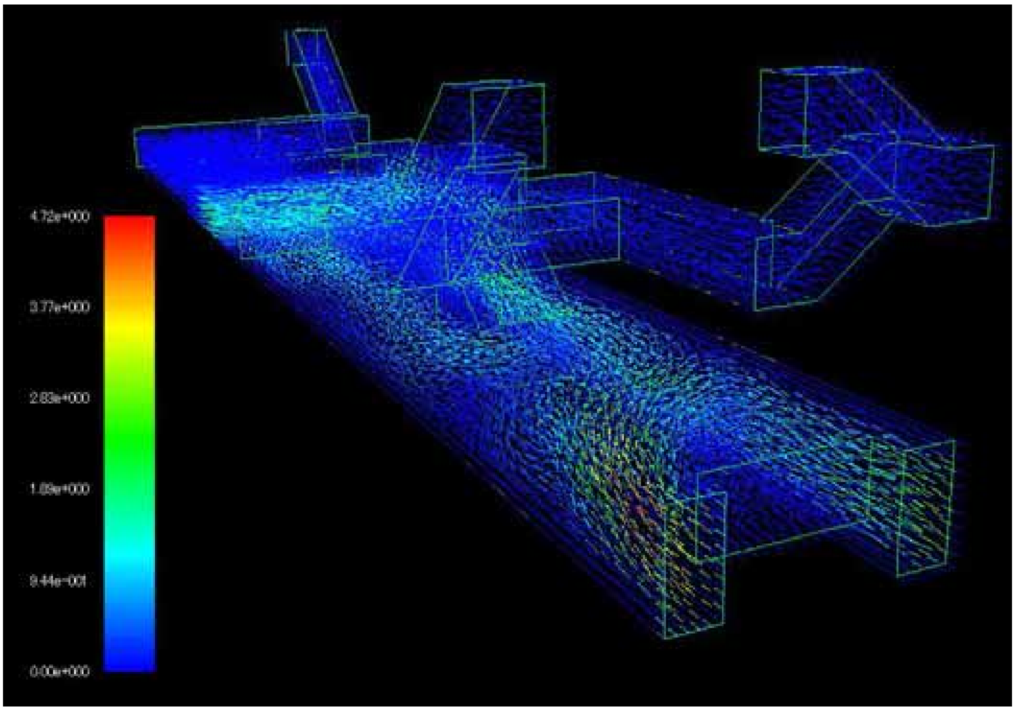

Fig. 15 : Visualization of the velocity field

References

[1] ADVENTURE Project: http://adventure.q.t.u-tokyo.ac.jp

[2] Yagawa, G., and Shioya, R.: Parallel finite elements on a massively parallel computer with domain decomposition, *Computing Systems in Engineering*, 4, Nos. 4-6, pp. 495-503(1993).

[3] Yagawa, G., and Shioya, R.: Massively Parallel Finite Element Analysis, Asakura-Shoten, (1998) (in Japanese)（[10]と同じ）

[4] Miyamura, T., Noguchi, H., Shioya, R., Yoshimaura, S., and Yagawa, G. : Massively parallel elastic-plastic finite element analysis using the hierarchical domain decomposition method, *Transactions of Japan Society of Mechanical Engineers (JSME)*, 65-A, No.634, pp. 1201-1208(1999) (in Japanese).

[5] MPI: http://www-unix.mcs.anl.gov/mpi/

[6] MPICH: http://www-unix.mcs.anl.gov/mpi/mpich/

[7] Mandel, J.: Balancing domain decomposition, *Communications on Numerical Methods in Engineering*, 9, 233-241(1993)

[8] Shioya, R., Kanayama, H., Mukaddes,A.M.M., and Ogino, M. : Heat conductive analysis with balancing domain decomposition method, *Journal of Theoretical and Applied Mechanics*,Vol.52,pp.43-53(2003).

[9] Quarteroni, A. and Vali, A. : Domain Decompostion Methods for Partial Differential Equations, Clarendon Press Oxford(1999).

[10] 矢川元基, 塩谷隆二; 超並列有限要素解析，計算科学シリーズ，朝倉書店(1998)

[11] Franca, L.-P. and Frey S.-L. : Stabilized finite element methods: Ⅱ. The incompressible Navier-Stokes equations, *Computer Methods in Applied Mechanics and Engineering*, Vol. 99, pp.209-233 (1992).

[12] Brooks, A. N., and Hughes, T. J. R. : Streamline upwind / Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations, *Computer Methods in Applied Mechanics and Engineering*, Vol. 32, pp.199-259 (1982).

[13] Gerard, L. G. S. and Diederik, R.-F. : BiCGSTAB(*L*) for linear equations involving unsymmetric matrix with complex spectrum, *Electronic Transactions on Numerical Analysis*, Vol. 1, pp. 11-32(1993).

[14] Hansbo, P. and Szepessy, A. : A velocity-pressure streamline diffusion finite element method for the incompressible Navier-Stokes equations, *Computer Methods in Applied Mechanics and Engineering*, Vol. 84, pp.175-192(1990).

[15] Hughes, T. J. R. and Brooks, A. N. : A theoretical framework for Petrov-Galerkin methods with discontinuous weighting functions : application to the streamline-upwind procedure, *in Finite Elementsin Fluids*, Fallagher, R. H., Norrie, D.H., Oden, H.T., and Zienkiewicz, O.C., eds., Vol.4, pp. 47-65(1982).

[16] Kanayama, H. and Toshigami, K.: Three-dimensional air flow analysis in clean rooms by a finite element method, *Theoretical and Applied Mechanics*, Vol. 36, pp.35-46(1988).

[17] Kanayama, H., Toshigami, K., and Motoyama, H. : A partial upwind finite element approximation for the stationary Navier-Stokes equations, *Computational Mechanics*, Vol. 5, pp.209-216 (1989).

[18] Tabata, M., and Suzuki, A. : A stabilized finite element method for the Rayleigh-Benard equations with infinite Prandtl number in a spherical shell, *Computer Methods in Applied Mechanics and Engineering*, Vol. 190, pp. 387-402(2000).

[19] Tezduyar, T. E., Mittal, S., and Shih, R. : Time accurate incompressible flow computations with quadrilateral velocity-pressure elements, *Computer Mehtods in Applied Mechanics and Engineering*, Vol. 87, pp. 363-384(1991).

[20] Zhou, T.-X. and Feng, M.-F. : A least squares Petrov-Galerkin finite element method for the stationary Navier-Stokes equations, *Mathematics of Computation*, Vol. 60, pp. 531-543(1993).

[21] Saad, Y. and Schultz, M. H., GMRES : A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SLAM Journal on Scientific and Statistical Computing*, Vol.7, pp.856-869(1986).

[22] Shioya, R. and Yagawa, G. : Iterative domain decomposition FEM with preconditioning technique for large scale problem, *ECM'99, Progress in Experimental and Computational Mechanics in Engineering and Material Behavior*, pp, 255-260(1999).

[23] Kanayama, H., Tagami, D., Araki, T., and Kume, H. : A stabilization technique for steady flow problems, *International Journal of Computational Fluid Dynamics*,Vol.18,No4, pp.297-301(2004).

[24] Kanayama, H., Tagami, D., and Chiba, M. : Stationary incompressible viscous flow analysis by a domain decomposition method, *Domain Docomposition Methods in Science and Engineering XVI*, pp.611 – 618(2006).