ADVENTURE_sFlow/ThermalConvection

Non-stationary Thermal Convection Solver with HDDM

Version 1.0

User's manual

March 25, 2015

ADVENTURE Project

Table of contents

1. Introduction

This is the user manual for ADVENTURE_sFlow/ThermalConvection (hereinafter called “ADVENTURE_sFlow”) that is finite element solver to analyze non-stationary incompressible viscous flow problem (Navier-Stokes equation) by parallel processing using Hierarchical Domain Decomposition Method (hereinafter called HDDM) under development in ADVENTURE Project [1]. In Chapter 1, outline and the operational procedure to the implementation of ADVENTURE_sFlow are explained. After Chapter 2, analysis functions of this program and others are introduced.

1.1. Features of this module

ADVENTURE_sFlow has the following features.

- It is possible to analyze non-stationary problem of thermal convection equation.
- It responds to tetrahedral piecewise linear elements
- Stabilized finite element method is used.
- Finite element method in which Lagrange derivative is approximated by characteristic curve method is used.
- It is formulated as a weak coupling problem that solve heat-transfer problems and flow problems separately.
- Parallel processing with load distribution by hierarchical domain decomposition method is possible. (For solution of flow problem only.)
- Parallel processing with load distribution by domain decomposition method is possible. (For solution of heat-transfer problem only.)

1.2. Operational environment

Operations of this program are checked in the following environment.

Corresponding platform: UNIX, Linux
Parallel communication library: MPI
Parallel linear algebraic solver: MUMPS, Lis

MPICH and OpenMPI are well-known free MPI libraries. One of them may be installed in recent Linux distributors, however, if they are not included, installation is required.

- MPICH official web site http://www-unix.mcs.anl.gov/mpi/mpich/
- OpenMPI official web site http://www.open-mpi.org/

MUMPS and Lis are parallel linear algebraic solver libraries that are called from TryDDM library used to packaging of thermal analysis processing. Refer to Appendix E for acquisition method and installation procedure. Since TryDDM library is included in the source code of ADVENTURE_sFlow, its downloading is unnecessary.

1.3. Compilation and installation

1.3.1. Compilation

For compilation of module of ADVENTURE_sFlow, C compiler, MPI compilation environment and ADVENTURE_IO should be installed. Follow the following procedure to compile ADVENTURE_sFlow module.

(1) Extract the archive file.

```
gunzip -c AdvsFlow-1.0.tar.gz | tar xvf -
```

(2) Move to extracted directory and implement compilation.
The initial state of Makefile.in assumes the module is installed to ${HOME}/ADVENTURE. If you want to install them to other places, open Makefile.in by a text editor and set the following macro by adjusting to the environment.

ADVSYS_DIR	Top directory of ADVENTURE system.
ADVIO_CONFIG	Full path to ADVENTURE_IO's script advsys-config.
MPI_CC	C compiler of MPI.
MPI_LINKER	C linker of MPI.
CC	C compiler.
LINKER	C linker.
CFLAGS	Optimization option.
EXT_LIB_BASE_DIR	Install directory of external library such as MUMPS.

Also, the initial state of Makefile.in is on the premise that all external libraries used by ADVENTURE_sFlow are installed to directories specified by EXT_LIB_BASE_DIR. If you install these libraries individually, the following macros should be set individually.

MUMPS_DIR	Directory that has MUMPS header and library.
SCALAPACK_DIR	Directory full path that has Scalapack header and library.
BLAS_DIR	Directory that has BLAS header and library.
PARMETIS_DIR	Directory that has ParMetis header and library.
LIS_INC_DIR	Directory that has Lis header.

LIS_LIB_DIR	Directory that has Lis library.

After that, implement make.

```
make
```

1.3.2. Install

When compilation succeeds, installation is implemented by the following command.

```
make install
```

It should be implemented by a user who has permission to write for directory to install.

To change directory to install, implement the following command.

```
make install prefix=install_dir
```

However, full path of directory to install should be specified for *install_dir*.

1.4. How to operate

There is execution module of parallel version static load distribution “advsflow-p” in ADVENTURE_sFlow.

Although MPI is used for parallel version, implementation method for MPICH is introduced as an example this time. There are various packaging for MPI and compilation and implementation method are dependent on packaging system. Therefore implementation is possible by referring each packaging system manual and replacing applicable area accordingly.

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

[*options for mpirun*]

Main options of mpirun are shown below. Refer to manual of MPICH for details.

- **-np** *number_of_hosts*

  Specify number of MPI process that are activated to number_of_hosts.

- **-machinefile** *machine_file*

  Specify a list of machine names used to parallel computation. If it is not specified, default file set in the system is used.

[*options*]

Specify types of analysis or set various items by specifying option of ADVENTURE_sFlow. Details are described later.

2. Features of solver

2.1. Introduction of a symmetric solver by the characteristic curve method and weak coupling problems

It is known that if Galerkin approximation is introduced into Navier-Stokes problem, there is a problem on treatment of the non-linear term. In ADVENTURE_sFlow, only stabilization method was applied up to Ver.0.5b to respond to this problem. In such a case, since simultaneous linear equations solved at each step of backward Euler method become asymmetric, asymmetric solvers such as GPBi-CG, Bi-CGSTAB and Bi-CGSTAB2 method were applied.

Other ADVENTURE solver modules such as ADVENTURE_Solid realize large increase in speed by applying strong preprocessing method such as BDD preprocessing to a symmetric solver. Therefore in Ver.1.0, method was revised to the one that a symmetric solver is used by approximating Lagrange derivative (material derivative) with the characteristic curve method.

Also, although it was formulated as a strong coupling problem that solves flow problems and heat-transfer problems as a single equation up to Ver.0.5b, it was revised to the one that it is formulated as a weak coupling problem that solves them separately in Ver.1.0.

2.2. Parallel processing capability

ADVENTURE_sFlow enables parallel treatment by using hierarchical domain decomposition method. Fig. 1 shows a pattern diagram of hierarchical domain decomposition method. Large decomposition unit of the first layer is called “Part” and fine decomposition unit of the second layer is called “Subdomain”. This domain decomposition shall be implemented prior to ADVENTURE_Metis.

ADVENTURE_sFlow apply MPI as parallel library and multiple processes (in some environment, threads) are invoked depending on specification by users. Since 1 process is generally activated per 1 node (CPU), in the following part, we use terms such as process, node and CPU without distinction for easier understanding. You can also allocate multiple processes to 1 node, of course.

Static load distribution version (advsflow-p) is ready for parallel treatment as implementation binary. As shown in Fig. 2 , calculation procedures are implemented parallel by allocating 1 Part to 1 process statically. Since number of Part and number of implementation process in domain decomposition are the same, ADVENTURE_Metis should adjust number of Part to number of process that are used for static load

distribution in advance.

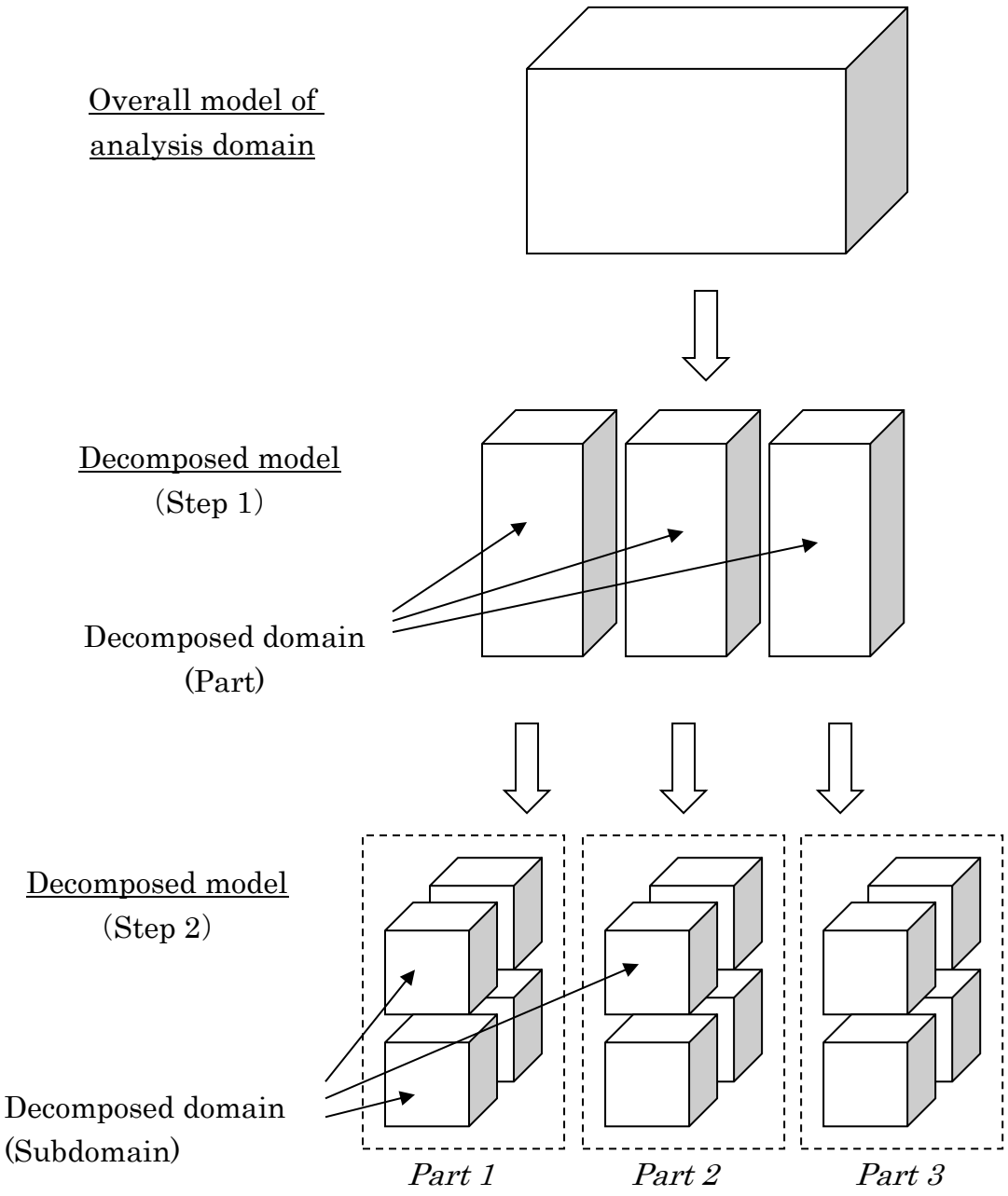


Fig. 1 : Hierarchical domain decomposition

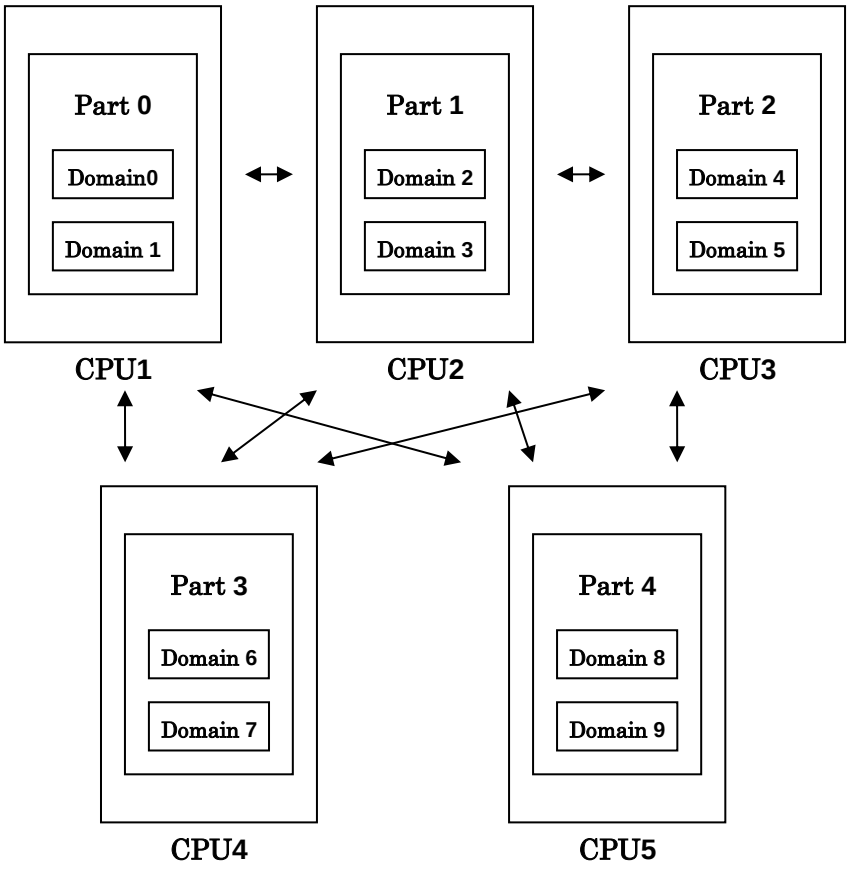


Fig. 2 : Allocation to domain CPU – Static load distribution version

2.3. Non-stationary analysis capability

Backward Euler method is used for non-stationary analysis. Also, heat-transfer problems and flow problems are solved separately step by step (weak coupling method). Fig. 3 shows general flow of these processing. Whole areas are double loops: outer loop is non-stationary iteration and inner loop is iteration to solve interface problems on hierarchical domain decomposition method. Although a figure is omitted this time, there actuary exists iterative calculation by CG method for solution processing of heat-transfer problems.

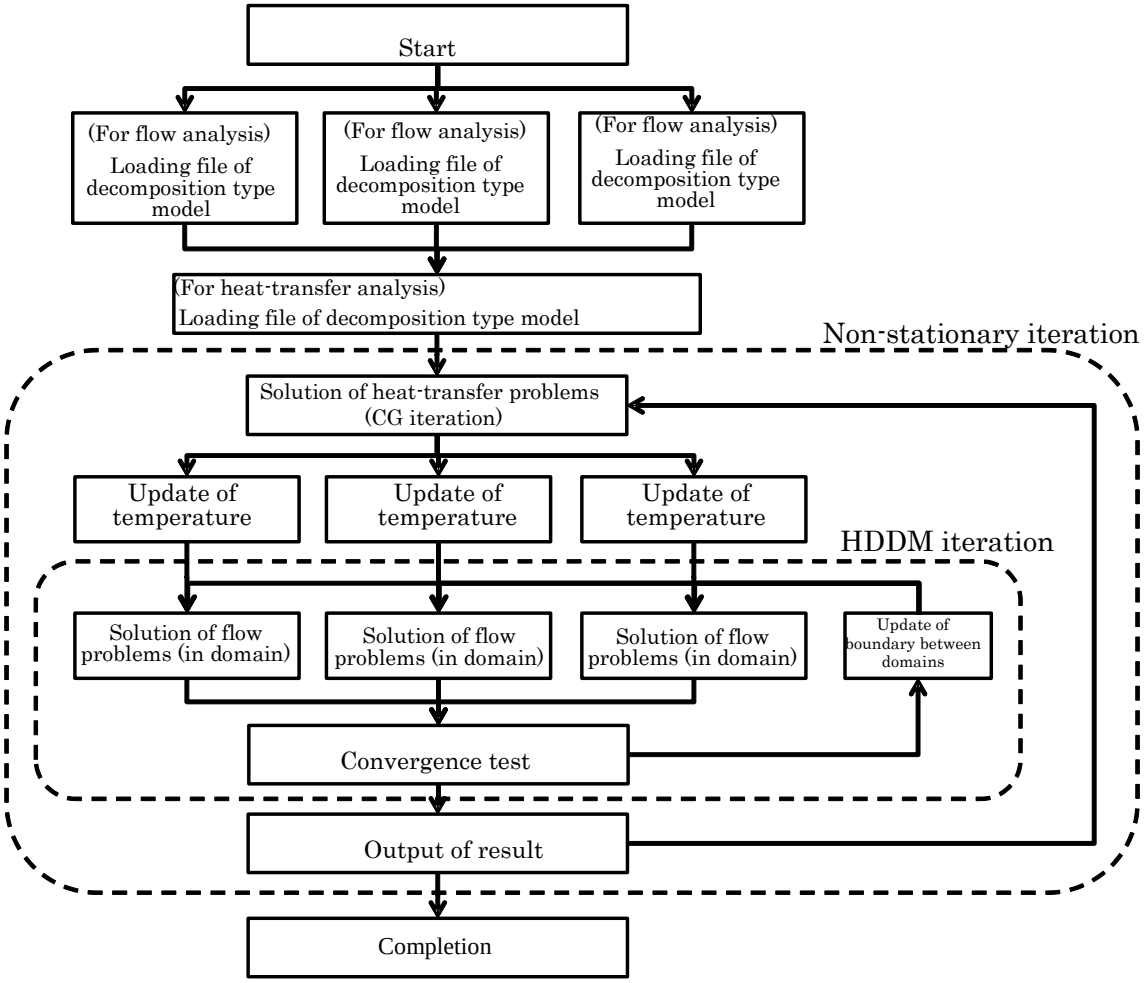


Fig. 3 : Process of analysis

2.4. Domain decomposition (ADVENTURE_Metis)

2.4.1. Parameter of domain decomposition

When analysis model is decomposed too finely, domain that includes no element may be created in ADVENTURE_Metis. When such domain is found at implementation in

ADVENTURE_sFlow, it is aborted after alert. Also, if decompositions are so rough compared to total number of elements, amount of calculation increase and it causes insufficient memory. It is also aborted after alert.

Like this, computation performance of ADVENTURE_sFlow module is depending on proper domain decomposition. Basically, number of “Part” is determined by parallel processing method, number of node used by network and computation machine environment. Number of “Subdomain” is determined according to memory necessary to computation processing. It is obvious that used amount of memory per Subdomain can be decreased by finer decomposition.

2.4.2. Two domain decomposition models

In ADVENTURE_sFlow / ThermalConveciton Ver. 1.0, different analysis model files are used for flow analysis and heat-transfer analysis.

Domain decomposition model for flow analysis can implement decomposition to multiple “Part.” At this time, option (-difn 4) is required to designate degree of freedom of node on internal boundary to 4.

In domain decomposition model for heat-transfer analysis, it is necessary to implement decomposition for 1Part only due to restriction of using library (TryDDM). Also, option (-difn 1) is required to specify degree of freedom of node on internal boundary to 1.

Details of implementation method, etc. of ADVENTURE_Metis are described in 3.1.

3. Analysis capabilities

ADVENTURE_sFlow can analyze non-stationary thermal convection problems by parallel processing. Functions of these analyses are shown below.

3.1. Analysis procedure

In ADVENTURE system, the processing flow of ADVENTURE_sFlow, preprocessing and post-processing are shown in Fig. 4.

(1) Generation of mesh data (ADVENTURE_TetMesh)
Generate mesh from a surface patch file.

(2) Assignment of boundary conditions (ADVENTURE_BCtool)
Assign boundary conditions to mesh of analysis object.

(3) Analysis model transformation (makefem3 or fgr_getnode, sFlow_makefem_c, makebc)
Use makefem3 that is accessory tool of ADVENTURE_BCtool Ver.2.1 to create integrated analysis model file.
If ADVENTURE_BCtool Ver.2.1 cannot be used, use “fgr_getnode” and “sFlow_makefem_c” that are accessory tools of ADVENTURE_sFlow. They are for generation of node data belongs to each boundary surface and for creation of integrated type analysis model file. Use “makebc” of ADVENTURE_BCtool Ver.2 for heat flux and heat- transfer boundary condition. Refer to Appendix for detailed implementation method.

(4) Domain decomposition (ADVENTURE_Metis)
Create domain decomposed model from integrated analysis model.
2 types of domain decomposition models, that is to say, for flow analysis and for heat analysis should be prepared. Both models are created from same integrated type analysis file by using domain decomposition module ADVENTURE_Metis.
For example, creation of domain decomposition model for flow analysis should be created by using, -difn 4 that is option of adventure_metis.

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

This option ”-difn 4” is used to designate degree of freedom of node on internal boundary to 4. This is because degree of freedom of node solved by flow problem

analysis is 4 while degree of freedom of node displacement solved by solid static analysis is 3. "-difn 4" is a required option for input model creation to ADVENTURE_sFlow.

For example, creation of domain decomposition model for heat analysis should be created by using option "-difn 1" of adventure_metis.

```
adventure_metis -difn 1 [options]
                model_filename directory_name_th div_num
```

This option "-difn 1" is used to specify degree of freedom of node on internal boundary to 1. This is because degree of freedom (temperature) solved by heat analysis is 1 degree of freedom per 1 node. "-difn 1" is a required option for input model for heat-transfer problem in ADVENTURE_sFlow.

Also, note that output directory should be different from the one at implementation of adventure_metis for flow analysis to differentiate it from domain decomposition analysis file for flow analysis and "mpirun" is unnecessary because decomposition is 1 Part only.

(5) Non-stationary thermal convection problem analysis (ADVENTURE_sFlow)

Finite element analysis is implemented with decomposed analysis model as input.

However, note that Ver.1.0 uses 2 types of decomposed analysis files as input files, that is to say, for flow problems and for heat-transfer problems.

(6) Visualization (ParticleViewerHPC/AVS/ParaView, etc.)

Analysis results are visualized.

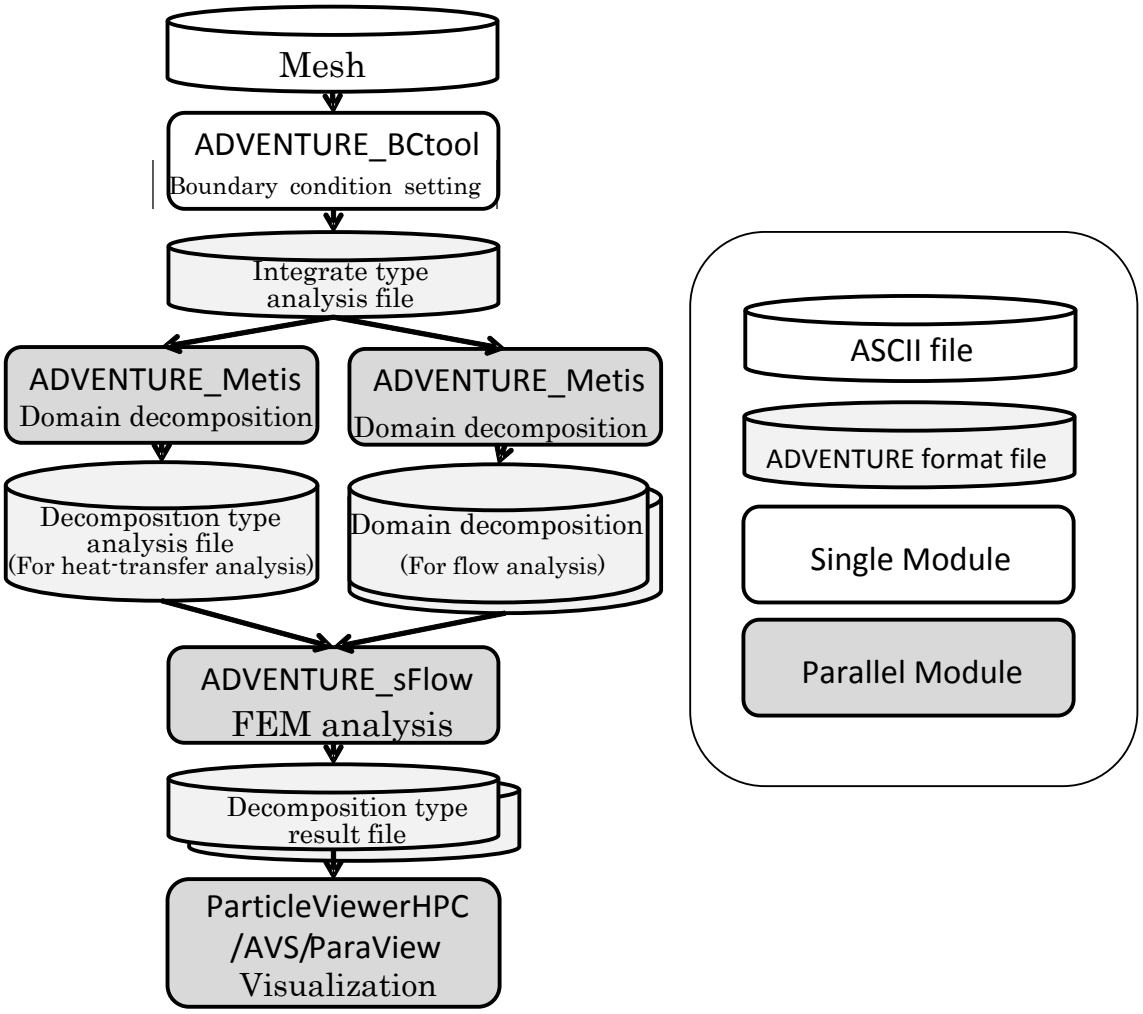


Fig. 4 : Flow of analysis

3.2. Input/ output data

Fig. 5 shows input and output file of ADVENTURE_sFlow. All files other than log output on the screen are ADVENTURE format and they are 1 file per Part. However, decomposition type analysis file to solve heat-transfer problems is used simultaneously separately from decomposition file for flow analysis.

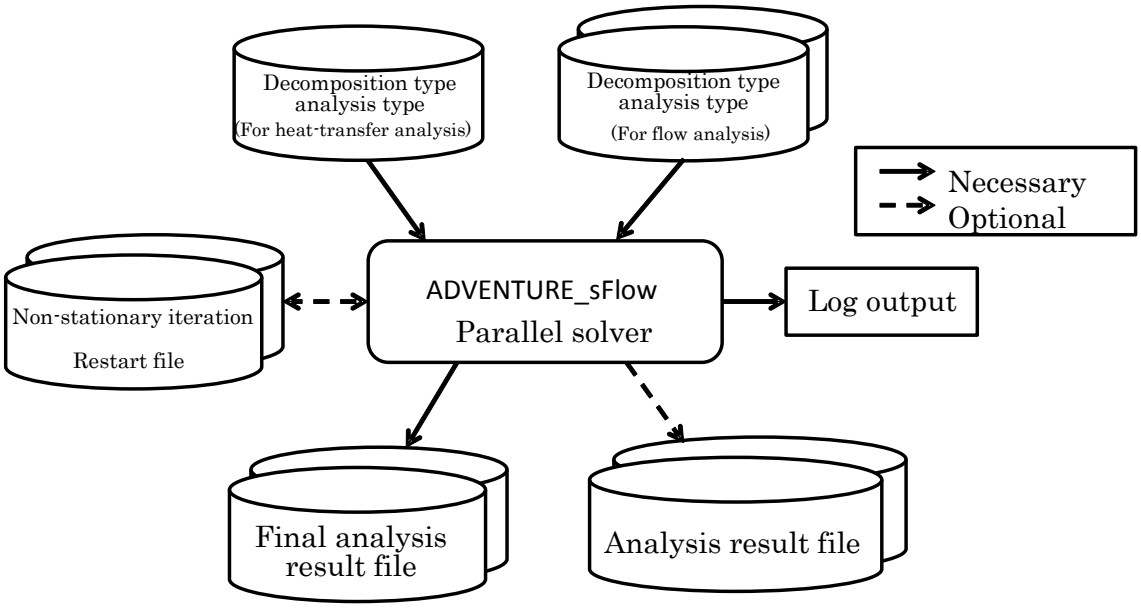


Fig. 5 : Input and output file

Output files are also given by hierarchical domain decomposition type and it has physical quantity of velocity, pressure and temperature. At non-stationary analysis, output time step by time step is also possible. These output data are decomposed into parts and subdomains, too.

There is restart function which save computation result halfway once and it restarts computation from that point to enable analysis in the environment where time to continue implementation is limited. There is non-stationary iteration restart file in available restart files.

3.3. Unit system

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

3.4. Boundary conditions

The following environment conditions can be added.

- Dirichlet boundary conditions (designation of node / flow velocity, pressure, temperature)
- Natural boundary conditions. (No surface force, no heat flux)
- Others. (with heat flux, heat-transfer)

3.5. Physical properties

The following physical property values can be used to isotropic material properties.

- Viscosity.
- Thermal conductivity. (Corresponding to thermal diffusion coefficient)
- Thermal expansion ratio.
- Reference temperature.

Although gravity acceleration is not technically physical properties, it can be defined by the similar format.

3.6. Analysis result outputs

Analysis results are given by hierarchical domain decomposition style and velocity, pressure and temperature at each node is output.

As well as analysis model, one file of ADVENTURE format is output per Part.

4. How to run

Regarding activation command, MPI is necessary for parallel version activation, and mpich invocation command is shown below.

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

Here [*options for mpirun*] are options of mpirun command.
[*options*] are options of ADVETURE_sFlow itself and designation of these options allows specification of analysis types and various settings. (Refer to 4.2 for details.)
data_dir is a required option and it designates the top directory of input and output data file. Directories and file names below are described in the following section.

4.1. Names for output/ input

Defaults of input and output file are as shown below. data_dir is the top directory. Each file is placed under this data_dir.

- Decomposed analysis model file (For flow analysis) :
  data_dir/model/advhddm_in_P.adv
- Decomposed analysis model file (For heat-transfer analysis) :
  data_dir/model_th/advhddm_in_0.adv
- Analysis result file:
  data_dir/result/advhddm_out_P.adv
- Restart file(File during the process of non-stationary analysis):
  data_dir/result/advhddm_out_S_P.adv

Here P stands for Part No. and S stands for step No. of backward Euler method.

4.2. Run-time options

Available options at implementation are shown below.

4.2.1. Options for non-stationary analysis

(Common to flow analysis and heat-transfer)

- `-ns`

  Implement non-stationary analysis. If this option is not applied, Stokes problems may occur. Also, the following sub option can be designated after –ns.

- `--ns-tol x`

  Specify tolerance for convergence. This is relative change of backward Euler method. When relative change becomes smaller than it, it is regarded that stationary solution is achieved.

  Default is set to 1.0×10^{-4} .

- `--step n`

  Specify the upper limit of number of steps of backward Euler method to n times. Default is set to 20. (Default value is for reference.)

- `--out-interval n`

  Output analysis result file every n times of steps other than the final step of backward Euler method. It is not output at default.

- `--use-resin n`

  Restart file of step n of backward Euler method that was output at the last implementation is read and analysis restarts from that point.

- `--dt n`

  Set time interval to “n”.

(For flow analysis)

- `--init_v` V_{x0} V_{y0} V_{z0}

  Specify the initial velocity. If it is not set, it is set to `(0.0, 0.0, 0.0)`.

(For heat-transfer analysis)

- `--init_t` T_0

  Specify the initial temperature. If it is not set, it is set to 0.0.

4.2.2. Options for iteration method control

In ADVENTURE_sFlow, iterative computation based on hierarchical domain decomposition method is implemented and there are some options to control it.

(Common to flow analysis and heat-transfer analysis)

- `-cg-tol x`

  Specify tolerance for convergence test. This is relative tolerance respond to the first residual of iteration. When relative tolerance becomes smaller than it at iteration, it is regarded as convergence. Default is set to 1.0×10^{-6} .

(For flow analysis)

- `-precon [bdd | bdd-diag | diag | none]`

  Set one of BDD pretreatment (bdd), BDD-diag pretreatment(bdd-diag), diagonal scaling pretreatment (diag) or no pretreatment (none) as pretreatment used to solve interface problems of hierarchical domain decomposition method. Default is set to no pretreatment (none).
- `-cgloop-max n`

  Specify upper limit of number of iterations. Default is set to 10,000.
- `-cg-norm [ Euclid | Max ]`

  When the value is Max, set the relative residual of iteration method by Max norm. When Euclid is applied, set the relative residual of iteration method by 2 norm. Default is set to Euclid (2 norm).

Note that CG iteration parameter is fixed as follows in Ver.1.0.

- Upper limit of number of CG iteration: 300,000 times
- The relative residual norm of iteration method: Euclid (2 norm)

4.2.3. Options for changing input/ output file name

When designating files used to input and output, basically designate only the top directories of them and use default values of files and directory names after that. When you change these items, use the following options. S stands for step No. of time and P stand for Part No.

(For flow analysis)

- `-model-file file`

  Input analysis model file name for flow analysis shall be a file. "_P.adv" is added to the actual file name. Default is set to advhddm_in.
- `-model-dir dir`

  Sub directory name of input analysis model file shall be "dir". Default is set to model.

(For heat-transfer analysis)

- `-thermal-model-file file`

  Input analysis model file for heat-transfer analysis shall be a file. Unlike file

designation for flow analysis, a perfect filename including directory name shall be designated (however, exclude ".adv" suffix).

(Common to flow analysis and heat-transfer analysis)

- `-result-file file`

  Analysis result file name shall be a file. _P.adv is added to the actual file name. Default is set to advhddm_out.

- `-result-dir dir`

  Subdirectory name of analysis result file shall be "dir". Default is set to result.

- `-ns-resin-file file`

  Time step restart input file name shall be a file. _S_P.adv is added to the actual file name. Default is set to advhddm_out.

- `-ns-resin-dir dir`

  Subdirectory name of time step restart input file shall be dir. Default is set to result.

4.2.4. Other options

(Common to flow analysis and heat-transfer analysis)

- `-memlimit n`

  Upper limit of memory that each process uses shall be n [MByte] and if the value exceeds it, implementation is stopped at that time. Default is set to 1,024 [MByte].

- `-help` or `-h`

  Main help message is displayed.

- `-version` or `-v`

  Version of module is displayed.

- `-help-ns`

  Help message to designate control option of non-stationary iteration is displayed.

- `-help-iter`

  Help message to designate control option of iterative method is displayed.

Appendix

A Elements

B Boundary conditions and physical properties

C Tools

D Analysis examples

E Installation procedure of dependent libraries

A Elements

ADVENTURE_sFlow responds to tetrahedral piecewise linear elements.

A.1 Tetrahedral piecewise linear elements

Number of node is 4. Fig. 6 shows the order of node No. of each node at element connectivity.

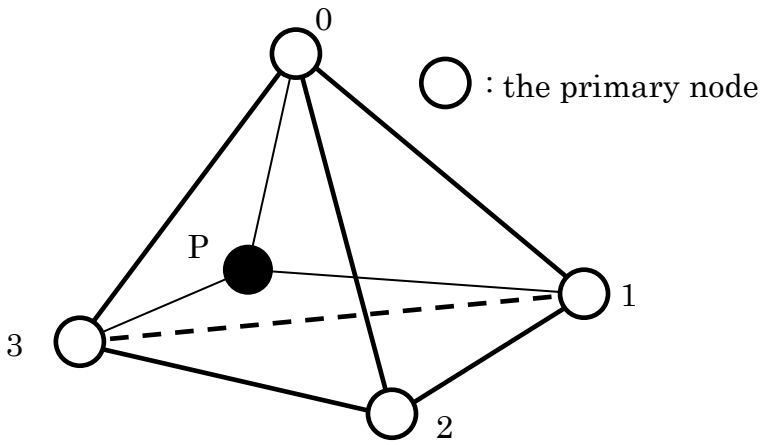


Fig. 6 : Tetrahedral piecewise linear elements

Fig. 7 shows correspondence of node No. and element surface No.

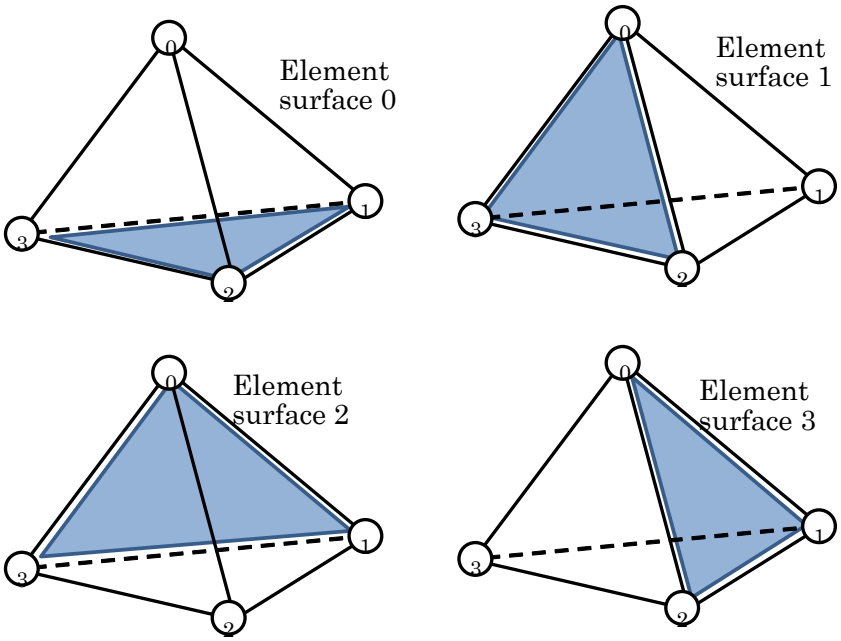


Fig. 7 : correspondence of node No. and element surface No.

B Boundary conditions and physical properties

Examples of available formats of boundary conditions in ADVENTURE_sFlow are shown below.

B.1 Flow specified boundary conditions

```
[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 1.000000e+00
1 0 1.000000e+00
3 0 1.000000e+00
58 0 1.000000e+00
59 0 1.000000e+00
60 0 1.000000e+00
...
...
```

Fig. 8 : Examples of boundary conditions to designate degree of freedom

Node No., direction component and set value [pressure and velocity] are shown from the left. However, direction component requires 3 as pressure component besides flow velocity direction component 0, 1 and 2 because degree of freedom per 1 node of non-stationary thermal convection problem analysis is 4 (heat analysis is solved separately).

B.2 Temperature specified boundary conditions

```
[Properties]
1: content_type=FEGenericAttribute
2: num_items=81
3: fega_type=NodeVariable
4: label=Temperature
5: format=i4f8
6: index_byte=4
[Data]
0 0 1.000000e+00
1 0 1.000000e+00
3 0 1.000000e+00
...
...
```

Fig. 9 : Examples of format of temperature specification boundary conditions

Node No., direction component and temperature are shown from the left. However, direction components are ignored inside solver. This format is the same as temperature specification boundary conditions of heat analysis solver ADVENTURE_Thermal that is under the development of ADVENTURE project.

You can also use Label name ”DirichletBC1”.

B.3 Heat flux boundary conditions

```
[Properties]
1: content_type=FEGenericAttribute
2: num_items=81
3: fega_type=NodeVariable
4: label=HeatFlux
5: format=i4f8
6: index_byte=4
[Data]
0 0 1.000000e+00
59 0 1.000000e+00
60 0 1.000000e+00
...
...
```

Fig. 10 : Examples of format of temperature specification boundary conditions

Node No., direction component and thermal flux are shown from the left. However, direction components are ignored inside solver. This format is the same as thermal flux boundary conditions of heat analysis solver ADVENTURE_Thermal that is under the development of ADVENTURE project.

B.4 Heat transfer boundary conditions

```
[Properties]
1: content_type=FEGenericAttribute
2: num_items=81
3: fega_type=ElementVariable
4: label=HeatConvection
5: format=i4f8f8
6: index_byte=4
[Data]
0 0 1.000000e+02 1.000000e+00
1 0 0.000000e+00 1.000000e+00
...
...
```

Fig. 11 : Examples of format of temperature specification boundary conditions

Element No., Element surface No., external temperature and heat-transfer coefficient are shown from the left.

This format is the same as heat- transfer boundary conditions of heat analysis solver ADVENTURE_Thermal that is under the development of ADVENTURE project.

B.5 Physical properties

```
[Properties]
1: content_type=FEGenericAttribute
2: num_items=1
3: fega_type=AllElementVariable
4: label=KinematicViscosity
5: format=f8
 [Data]
1.0000000e-2
```

Fig. 12 : Examples of format of physical properties

Only values of kinematic viscosity coefficient are designated here.

B.6 Gravity acceleration

```
[Properties]
1: content_type=FEGenericAttribute
2: num_items=1
3: fega_type=AllElementVariable
4: label=gravity
5: format=f8f8f8
 [Data]
0.0000000e+0 0.0000000e+0 -9.8000000e+0
```

Fig. 13 : Examples of format of gravity acceleration

Set values of gravity acceleration for each X/Y/Z component.

B.7 Boundary condition settings for BCtool Ver.2.1

In makefem3 after BCtool Ver.2.1, it is possible to set boundary conditions and gravity acceleration for ADENTURE_sFlow. The following formats are used for each.

- Boundary conditions to designate flow velocity
  velocOnFaceGroup <fgrId> 0 <dir> <value>
- Pressure boundary conditions
  presOnFaceGroup <fgrId> 0 3 <value>
- Temperature specification boundary conditions
  tempOnFaceGroup <fgrId> 0 0 <value>
- Thermal flux boundary conditions
  fluxOnFaceGroup <fgrId> 0 0 <value>
- Heat-transfer boundary conditions
  convOnFaceGroup <fgrId> <ref_temp> <coeff>
- Gravity acceleration
  gravity <gx> <gy> <gz>

Refer to BCtool manual for setting method and details of format of cnd file.

C Tools

In archive of ADVENTURE_sFlow, the following tools are included besides main body module.

C.1 Conversion filter of integrated analysis model: sFlow_makefem_c

sFlow_makefem_c is a tool that convert analysis condition file (suffix: cnd) for structure analysis obtained by ADVENTURE_BCtool to into the one for non-stationary thermal convection problems. Specifically, the following conversion is possible.

- Displacement boundary conditions ⇒ Boundary conditions to designate degree of freedom (Flow velocity, pressure and temperature)

Implementation method is as follows.

```
sFlow_makefem_c <kinematic viscosity> <heat conductivity> <beta> <Tr>
mshFile datFile cndFile advFile
```

Here mshFile means mesh data generated from ADVENTURE_TetMesh, datFile means surface group node list data generated from fgr_getnode (described later) and cndFile means boundary condition data generated from ADVENTURE_BCtool.

Designate kinematic viscosity coefficient in <kinematic viscosity>, designate temperature thermometric conductivity in <heat conductivity>, designate thermal expansion rate in <beta> and designate reference temperature in <Tr>. You can specify advFile name whatever you like.

By implementing this command, advFile for ADVENTURE_sFlow analysis is completed. Since physical properties are converted by this tool, you don't need to create integrated type file by makefem command of ADVENTURE_BCtool.

Also, you can thermal flux boundary conditions and heat-transfer boundary conditions, by creating adv file of only boundary conditions by makebc command included in BCtool Ver.2 and couple it with adv file that is created by sFlow_makefem_c

using advcat command.

C.2 Tool for creating face group node list: fgr_getnode

This is a tool that create list of nodes belonging to surface group from surface group file (fgr file) created by msh2pch command of ADVENTURE_BCtool.
Implementation method is as follows.

```
fgr_getnode fgrFile datFile
```

Here fgrFile is surface group definition data generated by msh2pch and datFile is surface group node list data.

C.3 Tool for coupling speed, pressure and temperature : hddmmrg_ts

This function couples decomposed velocity, pressure and temperature components that are obtained by analysis using ADVENTURE_sFlow after decomposition by ADVENTURE_Metis to one respectively.
Implementation method is as follows.

```
hddmmrg_ts -step <num> [Pressure, Temperature or Velocity]
directory_for_analysis
```

In [Pressure, Temperature or Velocity], select Pressure to find pressure field, select Temperature to fine temperature field and select Velocity to find speed field. Regarding directory_for_analysis, designate directory name that models decomposed by ADVENTURE_Metis and results analyzed by ADVENTURE_sFlow are included.

By implementation of this command, Pressure.dat, Temperature.dat or Velocity.dat can be obtained and you can know pressure, temperature and speed values of each node. Also, they are used when implementing data conversion tool introduced in the next section.

C.4 Tool for data conversion: advsflow_p_rest2ucd

This tool converts analysis results into UCD format of AVS. By this, you can check velocity field, pressure field and temperature field of analysis results by using Micro AVS, etc.
Implementation method is as follows.

```
advsflow_p_rest2ucd advFile ucdFile step del_time
```

advFile is advFile that is found by integrated analysis model conversion filter non_stationary_makefem. ucdFile is output UCD file name. You can specify file name whatever you like.

At implementation, note that not to input suffix of advFile and ucdFile. Also, Pressure.dat, Temperature.dat and Velocity.dat that are output by hddmmrg_ts introduced in the previous section should be in the same directory with advFile.

C.5 Tool for data conversion: sflow_vtkconv

This tool converts analysis results into vtk format. By this, you can check velocity field, pressure field and temperature field of analysis results by using ParaView, etc.

Implementation method is as follows.

```
sflow_vtkconv_ns.sh  mshFile vtk_prefix interval label1 dim1 [label2 dim2 …]
```

mshFile is ADVENTURE format mesh file.

vtk_prefix is prefix of output vtk file. For example, if ”step” is specified as a string, set of file such as ”step_0.vtk””step_1.vtk” is output (figures are step No.).

interval is output interval of time step and the same figures with ―out-interval” at implementation of advsflow-p should be designated.

For labelN and dimN, designate variable name that is output to vtk file and dimension number of that variable value. For example, if you convert velocity field, designate ”Velocity 3” and if you convert temperature field, designate ”Temperature 1”. When you convert more than one variable at once, enumerate variable names and dimension numbers to the number of variables. File of variable that is output by hddmmrg_ts introduced in the section before the last section should exist.

C.6 Tool for data conversion: sflow_pvconv

This tool converts analysis results into ParticleViewerHPC format. By this, you can check velocity field, pressure field and temperature field of analysis results by using ParticleViewerHPC, etc.

Implementation method is as follows.

```
sflow_pvconv.sh  mshFile lst_prefix start_step end_step interval
```

When it is implemented, lst file for ParticleViewerHPC is generated along with dat file output from hddmmrg. Copy these files and msh files to the same directory and select “Open the result of dynamic analysis” of ParticleViewerHPC for visualization.

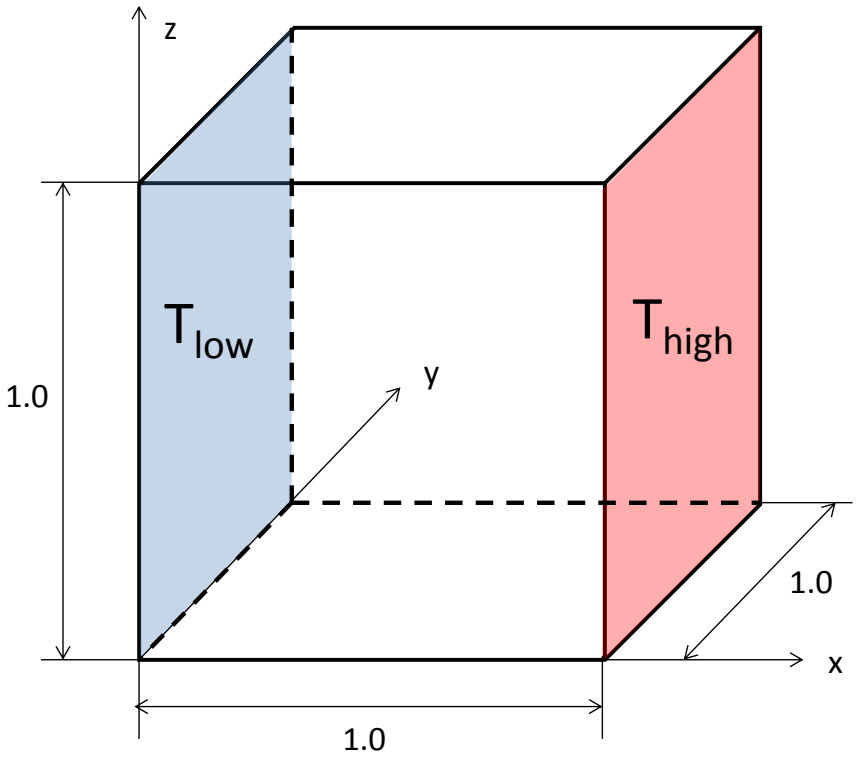
D Analysis examples

D.1 Analysis examples of non-stationary convection problems

In this section, procedures for modeling and domain decomposition using ADVENTURE System and analysis examples by ADVENTURE_sFlow are shown. Use each module of ADVENTURE_CAD , ADVENTURE_TetMesh and ADVENTURE_BCtool for modeling.

Procedures and commands used for this implementation are shown below. Refer to manuals of each system for details of each command.

Fig. 14 shows analysis model shape and boundary conditions.



Temperature specification :
T_{high} = 1.0[K], T_{low} = 0.0[K]
Velocity specification :
[x=0.0, x=1.0] (u, v, w) = (0, 0, 0) [m/s]
[y=0.0, y=1.0] v = 0 [m/s]
[z=0.0, z=1.0] (u, v, w) = (0, 0, 0) [m/s]

Fig. 14 : Thermal cavity analysis model shape and boundary conditions

(1) Creating shape of model

Here an analysis example of the thermal cavity problem of simple shape model is shown. At first, create geometry file: thermal_cavity.gm3d.

thermal_cavity.gm3d

```
box 0 0 0 1 1 1
```

Fig. 15 : Creation example of shape definition file

(2) Creating surface patch

Create surface patch using ADVENTURE_CAD based on geometry file.

At first, create node density file by appropriate editor. Fig. 16 shows a creation example.

thermal_cavity.ptn

```
BaseDistance
0.05
```

Fig. 16 : Creation example of node density file

Next, thermal_cavity.pcm is created by the next command. Node density is set to 0.05 here.

```
advcad thermal_cavity.gm3d thermal_cavity.pcm 0.05 -pcm
```

(3) Creating mesh data

Generate tetrahedral mesh and input node element information.

thermal_cavityc.pcc is created by the next command. Subsequently thermal_cavityc.msh is created.

```
advtmesh9p thermal_cavity
advtmesh9m thermal_cavityc
```

(4) Adding boundary conditions

Add boundary condition by using ADVENTURE_BCtool.

thermal_cavityc_3.fgr, thermal_cavityc_3.pch, thermal_cavityc_3.pcg and thermal_cavityc_3.trn are created by the next command.

```
msh2pch thermal_cavityc.msh 3
```

Then, when you implement bcGUI by the next command, the window as shown in Fig. 17 is opened.

```
bcGUI thermal_cavityc_3.pch thermal_cavityc_3.pcg
```

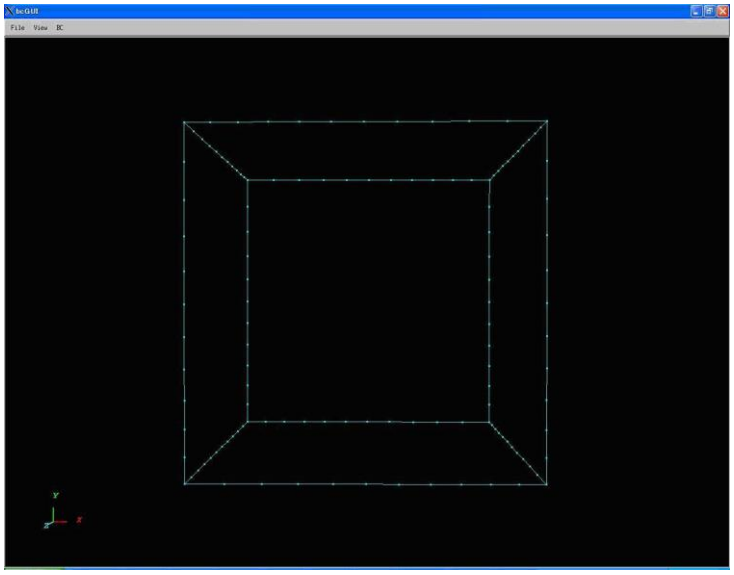


Fig. 17 : Window display by bcGUI command

At first, specify velocity of x, y and z to 0 for the selected surface (yellow) shown in Fig. 18. However, use “Displacement” as substitute of boundary conditions of velocity.

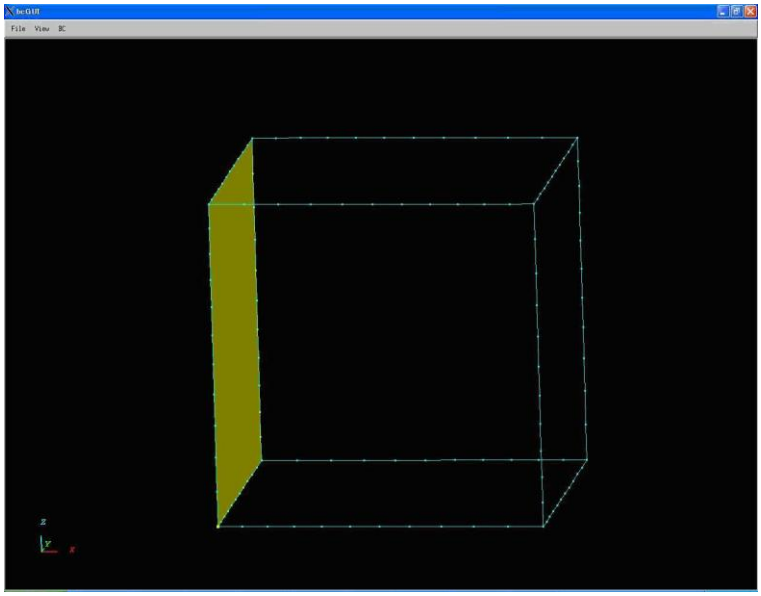


Fig. 18 : Specification example of boundary condition 1

Next, set appropriate boundary conditions to 6 surfaces shown in Fig. 19 respectively. Boundary condition setting of temperature will be explained in following.

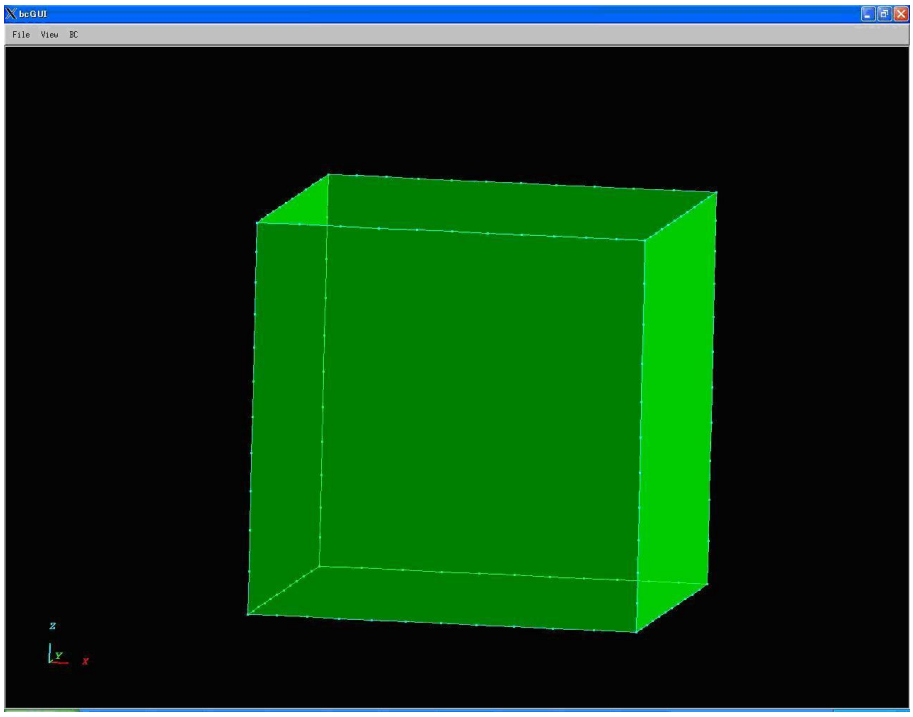


Fig. 19 : Specification example of boundary condition 2

After setting of all boundary conditions, output them by file name "thermal_cavity.cnd" .

Temperature boundary conditions should be input directly. In this case, as shown in Fig. 20, the lowest and the second lowest temperatures of 16 boundary conditions show temperature boundary conditions. The first No. of four numbers shows surface No. and 4 as the third No. shows temperature boundary condition. Then the fourth No. shows temperature 1 .0[K] and 0.0[K]. In this example, relationship between surface coordinate value and surface No. is as shown below. Since relationship may vary depending on environments, check it every time.

- Surface No. [0] : x=0
- Surface No. [1] : z=0
- Surface No. [2] : z=1
- Surface No. [3] : y=1
- Surface No. [4] : y=0
- Surface No. [5] : x=1

```
gravity 0 0 -9.8
boundary 16
dispOnFaceGroup 0 0 0 0
dispOnFaceGroup 0 0 1 0
dispOnFaceGroup 0 0 2 0
dispOnFaceGroup 1 0 0 0
dispOnFaceGroup 1 0 1 0
dispOnFaceGroup 1 0 2 0
dispOnFaceGroup 2 0 0 0
dispOnFaceGroup 2 0 1 0
dispOnFaceGroup 2 0 2 0
dispOnFaceGroup 3 0 1 0
dispOnFaceGroup 4 0 1 0
dispOnFaceGroup 5 0 0 0
dispOnFaceGroup 5 0 1 0
dispOnFaceGroup 5 0 2 0
dispOnFaceGroup 5 0 4 1
dispOnFaceGroup 0 0 4 0
```

Fig. 20 : Format example of analysis condition file (suffix cnd)

If makefem3 of ADVENTURE_BCtool Ver.2.1 or later is used, describe boundary conditions and physical property values as follows.

```
gravity 0 0 -9.8
boundary 16
velocOnFaceGroup 0 0 0 0
velocOnFaceGroup 0 0 1 0
velocOnFaceGroup 0 0 2 0
velocOnFaceGroup 1 0 0 0
velocOnFaceGroup 1 0 1 0
velocOnFaceGroup 1 0 2 0
velocOnFaceGroup 2 0 0 0
velocOnFaceGroup 2 0 1 0
velocOnFaceGroup 2 0 2 0
velocOnFaceGroup 3 0 1 0
velocOnFaceGroup 4 0 1 0
```

```
velocOnFaceGroup 5 0 0 0
velocOnFaceGroup 5 0 1 0
velocOnFaceGroup 5 0 2 0
tempOnFaceGroup 5 0 4 1
tempOnFaceGroup 0 0 4 0
```

Fig. 21 : Boundary condition file when using makefem3

```
KinematicViscosity    0.71
HeatConductivity     1.0
beta                  710
Tr                    0.5
```

Fig. 22 : Physical property value file when using makefem3

(5) Creating integrated analysis model

sFlow_makefem_c is a tool to convert analysis condition file obtained from ADVENTURE_BCtool into file for non-stationary thermal convection problem analysis.

At first, create node list of each surface by fgr_getnode.

Input the following command. Output file is “thermal_cavity.dat”.

```
fgr_getnode thermal_cavityc_3.fgr thermal_cavity.dat
```

Next, create integrated analysis model by the following command. However, output file is thermal_cavity.adv.

```
sFlow_makefem_c   0.71   1.0   710   0.5   thermal_cavityc.msh
thermal_cavity.dat thermal_cavity.cnd thermal_cavity.adv
```

In this problem, since heat flux boundary conditions and heat transfer boundary conditions are unnecessary, this is completion of integrated analysis model.

If makefem3 of ADVENTURE_BCtool Ver.2.1 or later is used, create integrates analysis model by the following command. Analysis model for sFlow is created by using “-sflow” option.

```
makefem3 -sflow thermal_cavityc.msh thermal_cavityc_3.fgr
thermal_cavity.cnd mp.dat thermal_cavity.adv
```

(6) Domain decomposition (For flow analysis)

Create the domain decomposed model files from the integrated analysis model using ADVENTURE_Metis. –difn 4 should be used at implementation. Because degree of freedom per one node is 4 in convection problem (since heat problem is [weak coupling] that is solved separately from fluid, it is not counted as degree of freedom at this point). Note that -difn 5 was designated in Ver.0.5b because thermal convection problems were solved by integrated equation (strong coupling). At first, to make layer domain decomposition, determine number of Part and number of Subdomain. Here perform analysis by static load distribution version by using 2 PCs. Therefore number of Part is 2. Element number of analysis model is 95,182 here and if number of element per 1 Part range is 500,

95,307 (number of element)÷2 (number of Part)÷500 (number of element per 1 Part range)=95.307. So number of Subdomain per 1 Part is 96.

Since (number of Part)x(number of Subdomain per 1 Part range), number of Subdomain for whole analysis domain is 192. Domain decomposition is implemented as the following command example.

```
mpirun -np 2 -machinefile machinefile adventure_metis -difn 4
thermal_cavity.adv . 96
```

Here –machinefile is option of MPI. ”.“ means designation of current directory as input / output directory.

(7) Domain decomposition (For heat transfer analysis)

For heat transfer problems, create the domain decomposed model files from the integrated analysis model using ADVENTURE_Metis, too.

Option `-difn 1` should be used at implementation. Because degree of freedom per 1 node is 1 (node temperature) in heat transfer problems.

Regarding decomposition of heat transfer analysis model in Ver.1.0, number of Part should be 1 regardless of number of PC used. Number of element of analysis model is 95,307 and if number of 1 Part range is 100,

95,307 (number of element) ÷1 (number of Part)÷500 (number of element per 1 Part range) = 190.614. So number of Subdomain par 1 Part is 190. Like this,

there is no problem even if the number of whole subdomain of flow analysis and the number of subdomain of heat transfer analysis are not the same. Also, to differentiate decomposition type analysis model file for flow analysis from it, output directory shall be changed from default value. Domain decomposition is implemented as the following command example.

```
 adventure_metis -difn 1 –subdir_name model_th thermal_cavity.adv .
950
```

Here –machinefile is option of MPI. Also, output directory is changed to standard from ”model/” to ”model_th” by using ”-subdir_ name model_th” option. ”.“ means designation of current directory as input / output directory.

(8) Analysis execution

Perform analysis with decomposed analysis model as input by using module of ADVENTURE_sFlow. Perform analysis as the following command example.

```
mpirun -np 2 -machinefile machinefile advsflow-p -ns –-ns-tol 0.6
--dt 0.01 --step 61 --out-interval 1 –thermal-model-file
model_th/advhddm_in_0 –cg-norm Euclid -precon diag .
```

Here –ns is option for ADVENTURE_sFlow and this is necessary option to perform non-stationary analysis. Analysis finish time is determined by smaller time between --ns-tol (0.6) and number of step x time interval (61×0.01＝0.61). In this example, diagonal scaling preconditioner is applied (-precon option). Analysis file for heat transfer analysis is designated at ”-thermal-model-file” option.

(9) Visualization of analysis results (ParticleViewerHPC)

The example to visualization by ParticleViewerHPC is introduced here. You can create lst file and dat file that can be used at ParticleViewerHPC by using sflow_pvconv.sh that is bundled with ADVENTURE_sFlow. Command example is shown below.

```
sflow_pvconv.sh 0 61 1 thermal_cavityc.msh .
```

The first and the second options designate the first and last steps to be converted, the third option designates interval of step to be converted, the fourth option

"thermal_cavityc.msh" designates mesh file and the fifth option designates places that directories of both model and result exist.

Fig. 24 shows the visualization result of temperature distribution of XZ surface by ParticleViewerHPC.

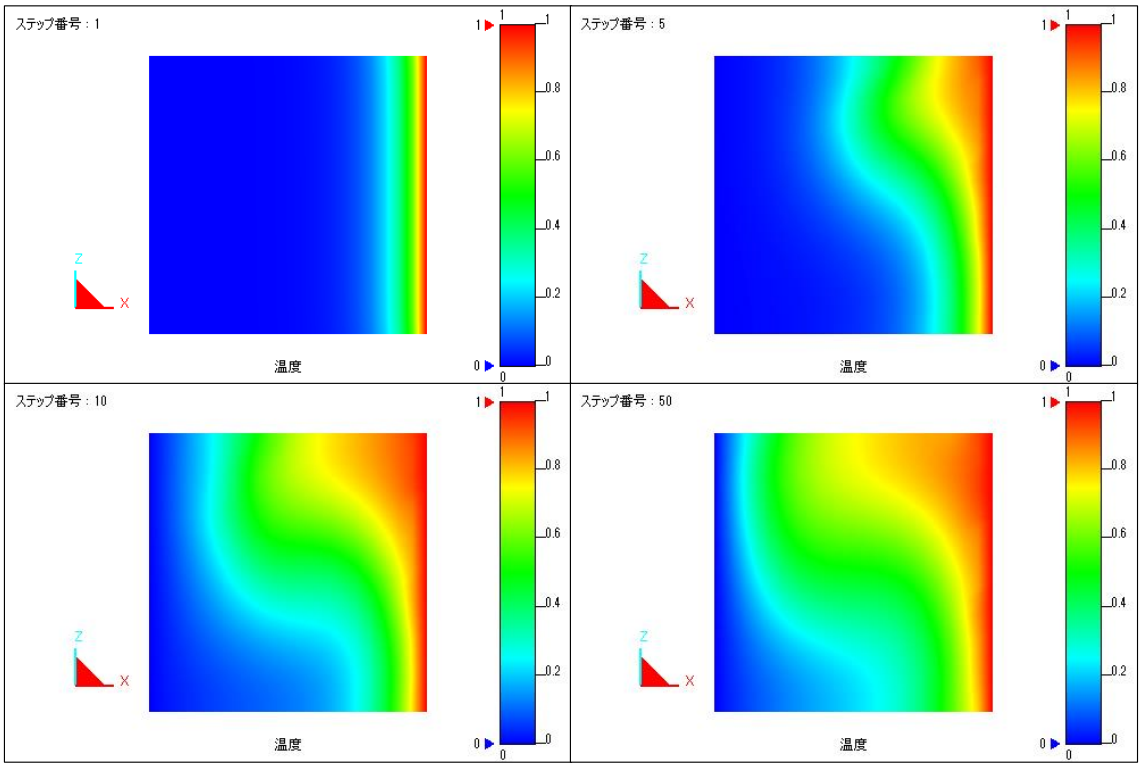


Fig. 23：Visualization of temperature distribution (ParticleViewerHPC)

(10) Visualization of analysis results (ParaView)

Here the example until visualization by ParaView is introduced. By using sflow_vtkconv_ns.sh that is bundled to ADVENTURE_sFlow, you can create VTK file that can use for ParaView, etc.

At first, retrieve data of pressure, speed and pressure from output file of analysis results. Implementation method is shown below.

```
hddmmrg_ts -step 60 Pressure .
hddmmrg_ts -step 60 Velocity .
hddmmrg_ts -step 60 Temperature .
```

By this, pressure, speed and temperature value for each node can be known.

Next, create VTK file from these output data (Pressure.dat, Velocity.dat,

Temperature.dat) and file of thermal_cavityc.msh. Command example is shown below.

```
sflow_vtkconv_ns.sh thermal_cavityc.msh step 61 1 Velocity 3
Pressure 1 Temperature 1
```

The first option “thermal_cavityc.msh” means mesh file, the second option “step” means number except step No. of VTK file, the third option “61” means number of step to be converted, the fourth option “1” means the same figure as “--out-interval” option at implementation of solver and subsequent options mean output item name (label name) and dimension numbers for each item. Fig. 24 shows visualization result of temperature distribution of XZ surface by ParaView.

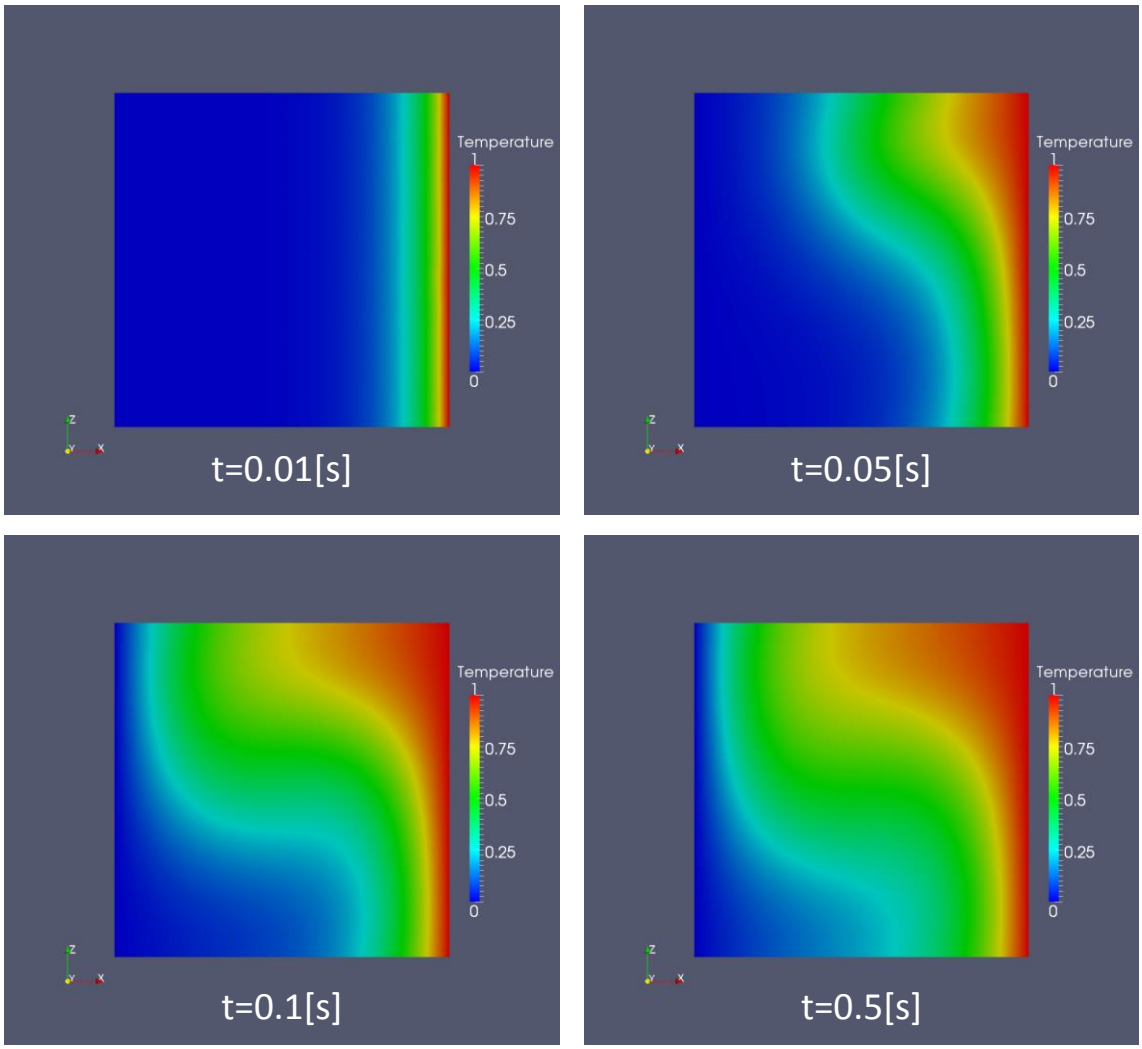


Fig. 24 : Visualization of temperature distribution

If visualization is implemented by AVS, convert file into UCD file by the following command.

```
advsflow_p_rest2ucd thermal_cavity thermal_cavity 61 0.01
```

Here the former thermal_cavity is ADV file name and the latter thermal_cavity is UCD type file name (output is thermal_cavity.inp).
Note that thermal_cavity.adv, Pressure.dat, Velocity.dat and Temperature.dat are in the same directory.

E Installation procedure of dependent libraries

ADVENTURE_sFlow Ver.1.0 is the implementation using functions of the following libraries.

1. MUMPS
2. Lis

Also, MUMPS uses many functions of external libraries.

In this section, installation procedure of these libraries are explained.

In the following explanation, installation location of these libraries is ${HOME}/lib/ as a premise.

E.1 Lis installation

Download source code of Lis from formal website (www.ssisc.org/lis/index.ja.html). The latest version as of 2015/2/24 is lis-1.5.51.tar.gz.

By deployment with tar+gzip, source code is deployed in lis-1.5.51 directory.

After moving to lis-1.5.51 directory, implement compilation and installation as follows.

```
$ CC=mpicc ./configure –prefix=${HOME}/ADVENTURE/extlib/lis/ --enabl
e-mpi
$ make
$ make install
```

Since TryDDM uses “lis_system.h” that is not installed in “make install,” move entire lis-1.5.51 directory to ${HOME}/lib.

E.2 MUMPS installation

MUMPS (a MUltifrontal Massively parallel sparse direct Solver) is parallel sparse matrix direct method solver library that is developed by INRIA, etc.

MUMPS uses ScaLAPACK, BLAS and BLACS. Also, AMD, AMF, PORD, METIS/ParMETIS or SCOTCH/PT-SCOTCH is available for ordering.

If Linux distribution is applied, MUMPS can easily be installed by using package control system of each distribution.

- Ubuntu Linux
  Install universe/mumps by apt-get, etc.
- Debian GNU/Linux
  Install libmumps-dev by apt-get, etc.
- Fedora Linux
  Install MUMPS-devel by yum.
- RedHatEnterpriseLinux/CentOS
  Add EPEL (Extra Packages for Enterprise Linux) repository [26] and install MUMPS-devel, etc. by yum.
- OpenSUSE
  Add science repository [27] and install mumps-devel, etc. by yast.

MUMPS is often preinstalled in the recent super computers. Like this, if pre-compiled MUMPS is available, we recommend using it.

The following description is a brief summary of the procedure of installation of MUMPS and all external libraries that MUMPS use from source code.

Compilation of dependent library (1): BLAS

Download source code of BLAS from website of netlib (www.netlib.org/blas/). blas.tgz (Apr. 19th 2011 release) is the latest version (as of 2014/11/13).

By deployment with tar+gzip, etc. after downloading, source code is deployed in ”BLAS” directory. It is already set for Linux+gfortran in this state. When you use other OS or fortran compiler, it is necessary to revise make.inc as needed. By implementing ”make” after that, blas_LINUX.a is created.

Set up symbolic link so that ” ln –s blas_LINUX.a libblas.a” can refer blas_LINUX.a as libblas.a because it will be required later.

After the completion of these operation, perform installation as ”cp –r BLAS ${HOME}/lib/”.

Compilation of dependent library (2) : LAPACK

When deploying lapack-3.3.1.tgz that is downloaded from netlib, lapack-3.3.1

directory is generated. After moving to lapack-3.3.1 directory, copy blas_LINUX.a that was created at the previous subsection (because test program bundled with LAPACK needs it) and rename make.inc.sample to make.inc. By implementing make after that, compile link is performed by GNU Fortran compiler.

After completion of compilation, perform installation by ”cp –r lapack-3.5.0 ${HOME}/lib/”.

Compilation of dependent library (3) : BLACS

Although there are multiple packages in BLACS depending on communication library to be used, use MPI version (mpiblacs.tgz) this time. When deploying mpiblacs.tgz that is downloaded from netlib, BLACS directory is generated.

After moving to BLACS directory, copy BMAKE/Bmake.MPI-LINUX to Bmake.inc and revise Bmake.inc according to environment. Examples of revision are shown below.

```
(Omission)
  BTOPdir = $(HOME)/lib/BLACS
(Omission)
   MPIdir = /usr/lib
(Omission)
#  ------------------------------------------------------------------
-----------
#  The Fortran 77 to C interface to be used.  If you are unsure of the
 correct
#  setting for your platform, compile and run BLACS/INSTALL/xintfac
e.
#  Choices are: Add_, NoChange, UpCase, or f77IsF2C.
#  ------------------------------------------------------------------
-----------
  INTFACE = -DAdd_
(Omission)
#====================================================================
===========
#  The following macros specify compilers, linker/loaders, the archi
ver,
#  and their options.  Some of the fortran files need to be compiled
with no
```

```
#  optimization.  This is the F77NO_OPTFLAG.  The usage of the remain
ing
#  macros should be obvious from the names.
#====================================================================
===========
   F77           = g77
   F77NO_OPTFLAGS =
   F77FLAGS       = $(F77NO_OPTFLAGS) -O
   F77LOADER      = $(F77)
   F77LOADFLAGS   =
   CC            = gcc
   CCFLAGS        = -O4
   CCLOADER       = $(CC)
   CCLOADFLAGS    =
```

When implementing “make mpi” after revision of Bmake.inc, compile link is performed by GNU C/Fortran compiler.

After completion of compilation, perform installation by ”cp –r BLACS ${HOME}/lib/”.

Compilation of dependent library (4) : ScaLAPACK

When deploying scalapack.tgz that is downloaded from netlib, scalapack-1.8.0 directory is generated. After moving to scalapack-1.8.0 directory, copy INSTALL/SLmake.LINUX to SLmake.inc and revise SLmake.inc according to installation location of MPI or BLAS/LAPACK to be used.

When implementing make after revision of SLmake.inc, compile link is performed by GNU Fortran compiler.

After completion of compilation, perform installation by ”cp –r scalapack-2.0.2 ${HOME}/lib/”.

Compilation of dependent library (5) : METIS/ParMETIS

Since MUMPS doesn’t respond to the latest ParMETIS, ParMETIS3.2.0 is used here. Since METIS is bundled with ParMETIS, compile ParMETIS.

Download source code from website of Karypis Lab (http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview). When deploying downloaded ParMETIS-3.2.0.tar.gz, ParMETIS-3.2.0 directory is created. When implementing make after moving to ParMETIS-3.22.0 directory, compile link is

performed by gcc.

After completion of compilation, perform installation by ”cp –r ParMETIS-3.2.0 ${HOME}/lib/”.

Compilation of MUMPS

When deploying MUMPS_4.10.0.tar.gz, MUMPS_4.10.0 directory is generated.
In Make.inc directory, copy Makefile.gfortran.PAR file to Makefile.inc according to the description of README, and revise it according to environment.

After revision, compile link is completed by implementing “make alllib”.

After completion of compilation, perform installation by”cp –r MUMPS_4.10.0 ${HOME}/lib/”.

References

[1] ADVENTURE Project: http://adventure.sys.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] Genki, Y. and Ryuji, S.: Massively Parallel Finite Element Analysis, Computational Science Series, Asakura Syoten (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).

[25] Yao, O., Kanayama, H., A Coupling Analysis of Thermal Convection Problems Based on a Characteristic Curve Method, Theoretical and Applied Mechanics Japan, 59, pp.257-264(2011)

[26] Fedora Project, EPEL/ja, https://fedoraproject.org/wiki/EPEL/ja

[27] openSUSE:Science team, openSUSE:Science Repositories,
http://en.opensuse.org/openSUSE:Science_Repositories