ADVENTURE_Magnetic

Electromagnetic Field Analysis with HDDM

Version 2.0.0

User's Manual

March 04, 2025

ADVENTURE Project

Contents

1. Introduction

The current document is a user manual for ADVENTURE_Magnetic, a parallel finite element analysis solver for electromagnetic field analysis currently being developed in the ADVENTURE Project [1].

In this chapter the overview of ADVENTURE_Magnetic and the operating procedures for execution are outlined, and from chapter 2 onwards the analysis features of this module are introduced.

1. 1. Program Features

ADVENTURE_Magnetic has the following features.

- ADVENTURE_Magnetic supports the electromagnetic field analyses with the finite element method.
  - (Non-linear) magnetostatic problem
  - Time-harmonic eddy current problem
  - Non-steady eddy current problem.
  - High-frequency electromagnetic field problem
- ADVENTURE_Magnetic has a solver that supports serial processing, or the load distribution of CPUs in parallel computing environments using the Hierarchical Domain Decomposition method (HDDM)[2][3][4][5].
  - Single mode : Serial processing
  - Shared-memory parallel mode : Parallel processing using OpenMP [6]
  - Distributed-memory parallel mode : Parallel processing using MPI [7]
  - Hybrid parallel mode : Distributed-memory parallel processing using MPI, and Shared-memory parallel processing using OpenMP in each MPI node
- ADVENTURE_Magnetic can analyze over 100 billion Degrees of freedom (DOFs) models [8][9]

1.2. Operational Environments

The ADVENTURE_Magnetic has been confirmed to operate in the following operational environments.

Supported platforms	: Linux, UNIX
Data processing library	: MPI (not used in single mode and shared-memory parallel mode)
Others	: C compiler, ADVENTURE_IO

1.2.1. Supported platforms

The ADVENTURE_Magnetic has been developed with the assumption that it will be executed on a supercomputer. Therefore, the supported platforms are Linux and Unix. It may also be possible to execute it on a Linux or Unix emulation environment, such as Windows Subsystem for Linux (SWL, SWL2) or Cygwin on Windows.

1.2.2. Data processing library

MPI must be installed to perform parallel analysis in distributed-memory parallel mode or hybrid parallel mode. Well-known free MPI libraries include MPICH [10] and OpenMPI [11]. In recent Linux distributions, it may be possible to install MPI library from the installation and configuration tool, but if not, it must be installed separately.

- Where to get MPICH
  https://www.mpich.org/
- Where to get OpenMPI
  https://www.open-mpi.org/

1.2.3. Others

OpenMP is required for shared-memory parallel mode and hybrid parallel mode. Check the compiler manual to see if the compiler supports OpenMP.

Also, there is an archive of ADVENTURE_IO on the ADVENTURE Project homepage

https://adventure.sys.t.u-tokyo.ac.jp/

So download this and install it according to the ADVENTURE_IO manual in advance.

1. 3. Program Compilation and Installation

To compile the ADVENTURE_Magnetic modules, you must have a C compiler, an MPI compilation environment (not required for single mode and shared-memory parallel mode), and ADVENTURE_IO installed.

Follow the steps below to compile and install the ADVENTURE_Magnetic modules.

(1) File Extraction from Archive

```
% tar zvfx AdvMagnetic-2.0.0.tar.gz
```

Here, “%” represents the command prompt, so it does not need to be entered. When the archive file is extracted, the AdvMagnetic-2.0.0 directory is created. In addition, AdvMagnetic-2.0.0 contains the following subdirectories:

HDDM	: Source files of HDDM solvers
common	: Common source files
doc	: Documents
lib	: Libraries
sample_data	: Sample data
tools	: Tools

(2) Edit “Makefile.in”

Go to the extracted directory and edit Makefile.in.

```
# ***********************************************************************
#  Copyright (C) 2000, 2001, 2002 Shinobu Yoshimura,
#              The University of Tokyo,
#              the Japan Society for the Promotion of Science (JSPS)
#  Copyright (C) 2003, 2005, 2007, 2014, 2015, 2016, 2017, 2018, 2021,
#              2024, 2025, ADVENTURE Project,
#  All Rights Reserved
# ***********************************************************************

# Include file for each Makefile
# Please modify for your own environment

# path for ADVENTURE_IO system
ADVSYSD       =     $(HOME)/ADVENTURE/bin      ← (A)

# path for install directory
INSTALL_DIR       =     $(HOME)/ADVENTURE       ← (B)
INSTALL_BINDIR    =     $(INSTALL_DIR)/bin
INSTALL_DOCDIR    =     $(INSTALL_DIR)/doc
INSTALL_DOCMAGDIR =     $(INSTALL_DOCDIR)/AdvMag

# C compiler & linker
CC      =     gcc              ← (C)
LINKER  =     $(CC)
AR      =     ar
ARFLAGS =     cr

# parallel C compiler & linker
MPI_CC     =    mpicc          ← (D)
MPI_LINKER =    $(MPI_CC)


# Compiler options
CFLAGS   = -O2                 ← (E)
OMPFLAGS = -fopenmp -lgomp
```

(A) Specify the absolute path to the directory where ADVENTURE_IO's advsys-config is installed. advsys-config is installed in the bin directory directly under the directory specified as the installation destination of ADVENTURE_IO. (Change the part in red to the directory specified as the installation destination of ADVENTURE_IO)

```
# path for ADVENTURE_IO system
ADVSYSD          =      $(HOME)/ADVENTURE/bin
```

(B) Specify the absolute path to the directory where ADVENTURE_Magnetic modules and manuals are to be installed. (Change the part in red to the installation destination)

```
# path for install directory
INSTALL_DIR        =      $(HOME)/ADVENTURE
```

(C) Specify the C compiler to be used. (Change the part in red to the C compiler to be used)

```
# C compiler & linker
CC      =     gcc
LINKER  =     $(CC)
```

(D) Specify the MPI compiler according to the MPI environment to be used.
(Change the part in red to the MPI compiler to be used)
Not necessary if only single mode or shared-memory parallel mode is used.

```
# parallel C compiler & linker
MPI_CC     =    mpicc
MPI_LINKER =    $(MPI_CC)
```

(E) Specify the compiler optimization options (`CFLAGS`).
Also, if the shared-memory parallel mode or hybrid parallel mode is used, specify the corresponding options (`OMPFLAGS`).
(Change the parts in red to the options you want to use.)

```
# Compiler options
CFLAGS   = -O2
OMPFLAGS = -fopenmp -lgomp
```

(3) Compile

To compile all parallel mode solvers and tools, execute the following command in the same directory as Makefile.in.

```
% make
```

To compile only the single mode, shared-memory parallel mode, distributed-memory parallel mode, or hybrid parallel mode, add a space to the above command, then add "`single`", "`s_omp`", "`parallel`", or "`p_omp`", respectively, and execute it. For example, the command to compile only the distributed-memory parallel mode is as follows:

```
% make parallel
```

By executing this command, the solver for distributed-memory parallel mode and tools will be compiled.

(4) Install

If the compilation is successful, install them with the following command.

```
% make install
```

Create a "bin" directory in the directory specified in (B) of "(2) Edit "Makefile.in"", and copy the executable files for the solvers and tools on to there. The above command should be executed by a user who has write permissions for the installation directory. Also, set the path to the installation directory according to the procedure for each environment (shell).

To install only the single mode, shared-memory parallel mode, distributed-memory parallel mode, or hybrid parallel mode, add "`-s`", "`-s_omp`", "`-p`", or "`-p_omp`" after "`install`" in the above command. For example, the command to install only the distributed-memory parallel mode is as follows:

```
% make install-p
```

By executing this command, the solver for distributed-memory parallel mode and tools will be installed.

If the solvers and tools have not compiled yet, they are compiled and then installed.

The executable files that will be installed are as follows:

Electromagnetic field analysis modules (HDDM)

- `advmag2_HDDM_Electromagnetic-s` : Single mode
- `advmag2_HDDM_Electromagnetic-s_omp` : Shared-memory parallel mode
- `advmag2_HDDM_Electromagnetic-p` : Distributed-memory parallel mode
- `advmag2_HDDM_Electromagnetic-p_omp` : Hybrid parallel mode

Tools

- `advmag2_advshw` : Tool for displaying ADVENTURE format file (binary)
- `advmag2_makefem_Electromangetic` : Tool for making FEA model for electromagnetic analysis
- `advmag2_hddmmrg` : Tool for integrating domain decomposed data and creating visualization file
- `advmag2_SourceVector` : Tool for computing source vector
- `advmag2_compare_results` : Tool for comparing analysis results
- `advmag2_makedat` : Tool for converting to material properties data file
- `advmag2_mesh_convert_file` : Tool for converting file format
- `advmag2_mesh_property` : Tool for analyzing mesh information
- `advmag2_mesh_cut_part` : Tool for cutting part of mesh
- `advmag2_mesh_merge` : Tool for merging meshes
- `advmag2_mesh_translation` : Tool for affine transformation (translation)
- `advmag2_mesh_scaling` : Tool for affine transformation (scaling)
- `advmag2_mesh_rotation` : Tool for affine transformation (rotation)
- `advmag2_mesh_mirror` : Tool for affine transformation (mirror)
- `advmag2_mesh_separate_elem` : Tool for creating visualization file of elements
- `advmag2_mesh_separate_dd` : Tool for creating visualization files of domain decomposition
- `advmag2_mesh_make_line` : Tool for generating mesh of line elements
- `advmag2_mesh_make_tri` : Tool for generating mesh (rectangular) of triangular elements
- `advmag2_mesh_make_quad` : Tool for generating mesh (rectangular) of quadrilateral elements
- `advmag2_mesh_make_tetra` : Tool for generating mesh (rectangular prism) of tetrahedral elements
- `advmag2_mesh_make_hexa` : Tool for generating mesh (rectangular prism) of hexahedral elements
- `advmag2_mesh_make_prism` : Tool for generating mesh (rectangular prism) of triangular prism elements
- `advmag2_mesh_make_pyramid` : Tool for generating mesh (rectangular prism) of square pyramid elements

Manuals: doc/AdvMag directory is created in the same location as the bin directory, and manuals are installed there.

- `manual-jp.pdf` : User's Manual in Japanese
- `manual-eg.pdf` : User's Manual in English

1. 4. Program Execution

Each electromagnetic field analysis module can perform either (non-linear) magnetostatic analysis, time-harmonic eddy current analysis, non-steady eddy current analysis, or high-frequency electromagnetic field analysis for each execution. Therefore, which analysis to perform must be specified at execution time. Specification is done by writing the following keywords after the executable file.

- (Non-linear) magnetostatic analysis : `Magnetostatic`
- Time-harmonic eddy current analysis : `TH_Eddy`
- Non-steady eddy current analysis : `NS_Eddy`
- High-frequency electromagnetic field analysis : `HF_EM`

Uppercase and lowercase letters will be recognized as different characters, so enter them as shown above.

When performing analysis, please use one of the four modes depending on your computer environment. For details on each mode, please refer to Section 2.1. Below are examples of execution in each mode.

1.4.1. Single mode

To perform the (non-linear) magnetostatic analysis with single mode, execute it as follows:

```
% advmag2_HDDM_Electromagnetic-s Magnetostatic [options]
```

[*options*] are command options. Specify as necessary. If nothing is specified, data will be read from the "data" directory, and the analysis results will be output to this directory. This is the same for other modes. For more information on command options, see Section 6.

1.4.2. Shared-memory parallel mode

To perform the time-harmonic eddy current analysis with shared-memory parallel mode, execute it as follows:

```
% advmag2_HDDM_Electromagnetic-s_omp TH_Eddy [options]
```

In this mode, the number of threads must be set as an environment variable, in advance. The setting method for typical environments (shell) are shown below. *n* is the number of threads.

- sh

  `% OMP_NUM_THREADS=`*n*
- csh or tcsh

  `% setenv OMP_NUM_THREADS` *n*
- bash

  `% export OMP_NUM_THREADS=`*n*

If it has been set in the configuration file, you do not need to run this command.

1.4.3. Distributed-memory parallel mode

To perform the non-steady eddy current analysis with distributed-memory parallel mode, execute it as follows:

```
% mpirun [options for mpirun] advmag2_HDDM_Electromagnetic-p NS_Eddy [options]
```

[*options for mpirun*] are options for mpirun. The main options are as follows. For details, please refer to the manual of MPICH or other MPI library.

- **-np** *n*

  The number of machines (corresponding to the number of parts).
- **-machinefile** *machine_file*

  The files contain the name of network machines.

Note that "mpirun" may also be called other names, such as "mpiexe" or "mpiexec". Please refer to the manual for each MPI library.

1.4.4. Hybrid parallel mode

To perform the high-frequency electromagnetic field analysis with hybrid parallel mode, execute it as follows:

```
% mpirun [options for mpirun] advmag2_HDDM_Electromagnetic-p_omp HF_EM [options]
```

As with the shared-memory parallel mode, the number of threads must be set as an environment variable, in advance. In addition, it must be set not only on the computer where you execute the above command, but on all computers where the MPI processes are started up.

2. Parallel Processing and Analysis Solver

2. 1. Parallel Processing

ADVENTURE_Magnetic uses the Hierarchical Domain Decomposition method[2][3][4][5] to provide parallel processing of analysis data. An entire-type model is decomposed in two steps (Fig. 1) by the ADVENTURE_Metis module prior to execution of ADVENTURE_Magnetic. A large decomposed unit of the first hierarchy level is called Part, and smaller units of the decomposed Part (second hierarchy level) are called Subdoamins. The details are given in the User's Manual of the ADVENUTRE_Metis module.

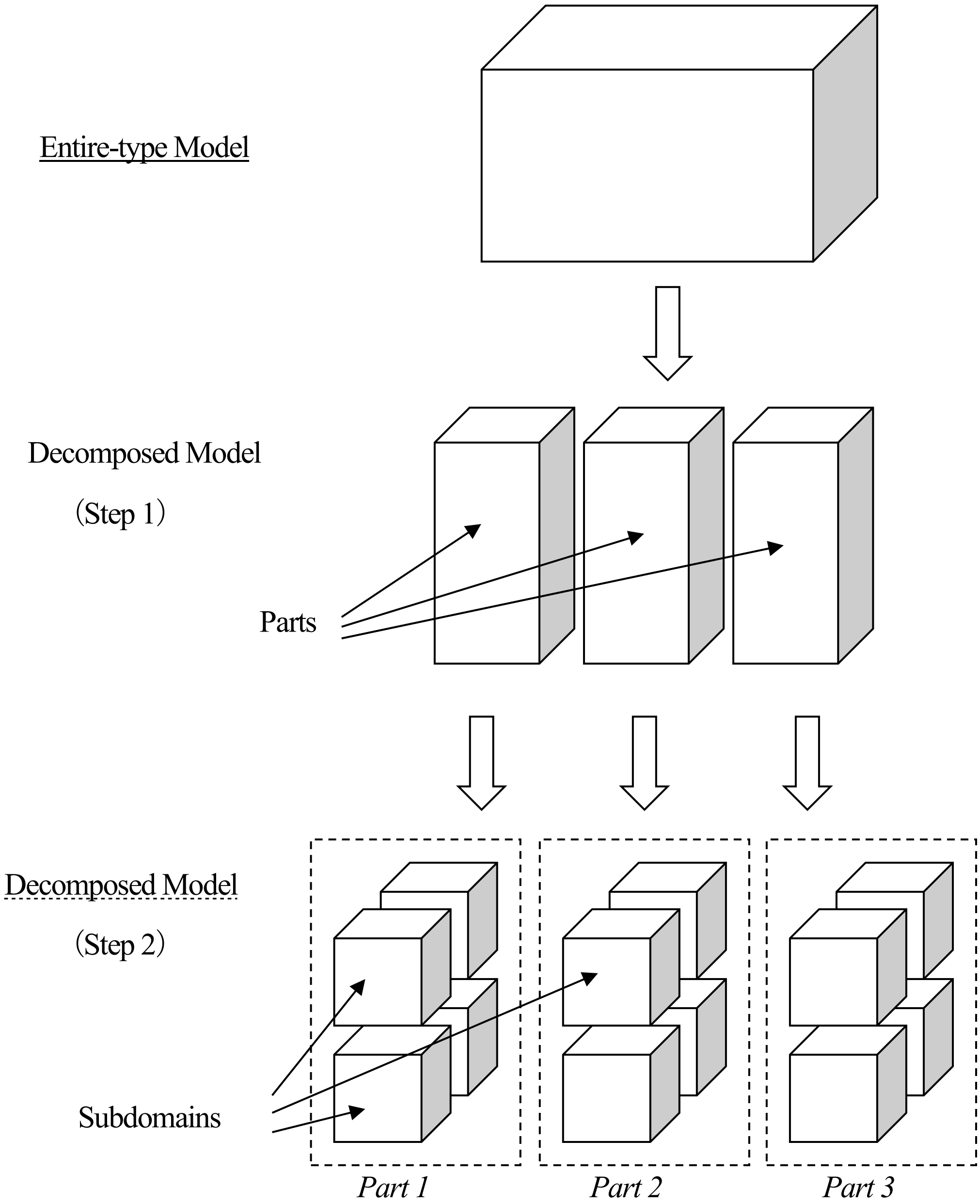


Fig. 1. Hierarchical Domain Decomposition.

ADVENTURE_Magnetic's distributed-memory parallel mode (`advmag2_HDDM_Electromagnetic-p`) uses MPI [7] as the parallel library. At startup, multiple MPI processes are launched according to the specifications. Since it is common to launch one process per node (CPU), for the sake of clarity, the following will not distinguish between process, node, and CPU. Of course, it is also possible to assign multiple MPI processes to one node.

In the distributed-memory parallel mode, parallel computing is performed by assigning one “Part” to one MPI process, as shown in Fig. 2. The number of “Parts” should correspond to the number of processes, it is necessary to set in ADVENTURE_Metis to the same number of processes used in the distributed-memory parallel mode. It is possible to run with more MPI processes than the number of “Parts”, but the surplus processes will not do anything, which will waste computational resources. Also, if the number of MPI processes is fewer than the number of “Parts”, an error message will be output and the program will be forcibly terminated.

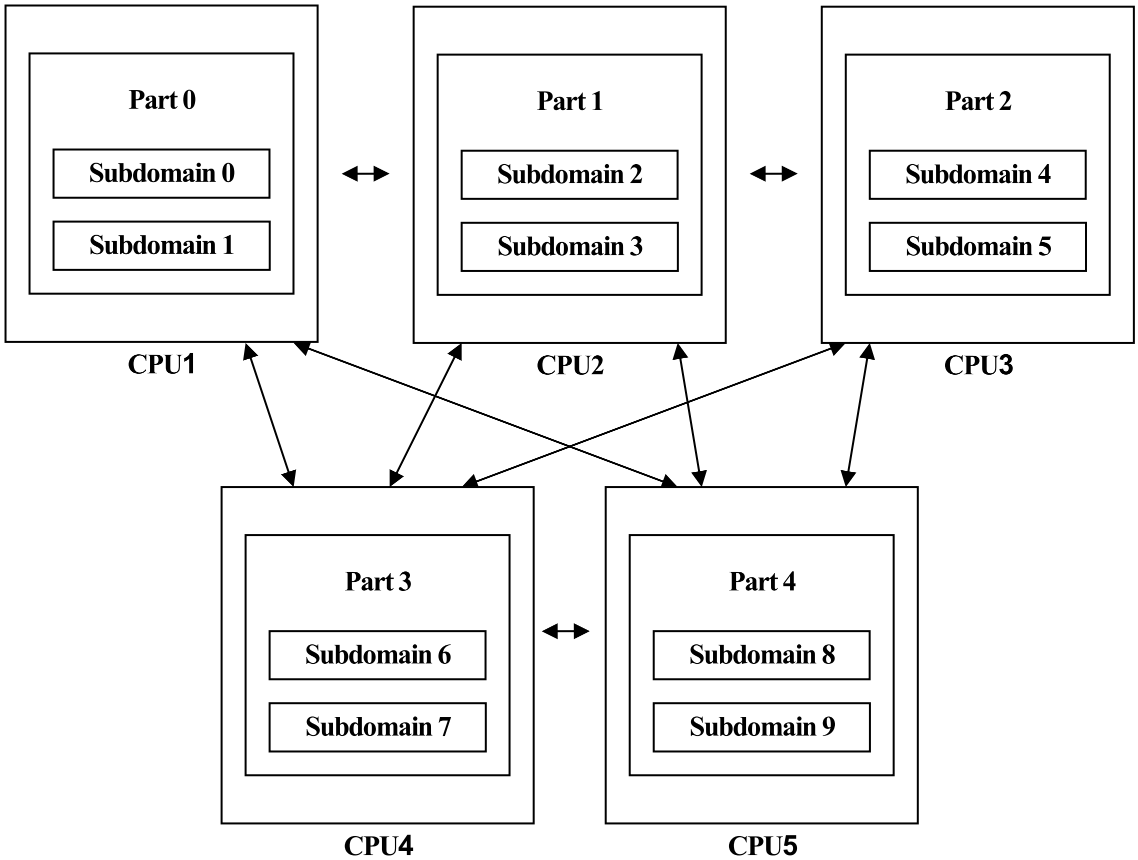


Fig. 2. Adjustment of Domains to CPUs (Distributed-memory parallel mode).

In the hybrid parallel mode (`advmag2_HDDM_Electromagnetic-p_omp`), the distributed-memory parallel mode is hybridized by using OpenMP to perform shared memory parallelism within the MPI process. One "Part" is assigned to one MPI process in the same way as in the distributed-memory parallel mode, but the processing related to the subdomains within the part are divided among OpenMP threads (Fig. 3).

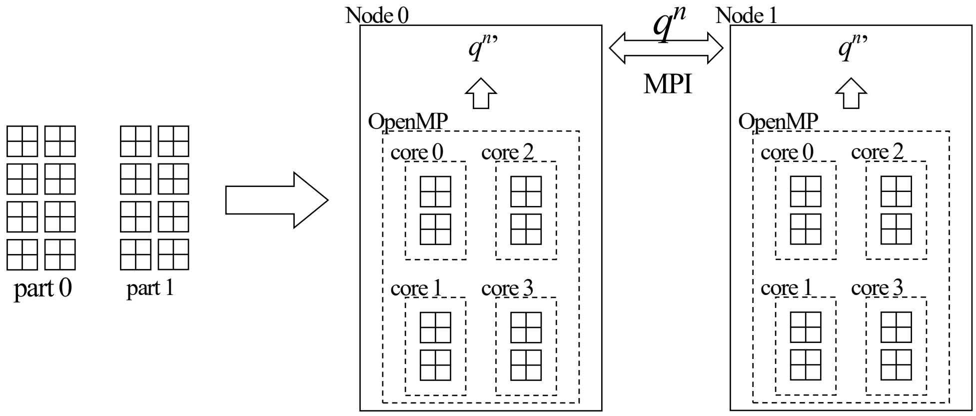


Fig. 3. Hybrid parallel mode.

In the single mode (`advmag2_HDDM_Electromagnetic-s`), no parallel computations are performed, and all computations are performed in one process. Compilation and execution is possible without the MPI library. Essentially, it is the same as the computations that are performed in parallel for each "Part" in the distributed-memory parallel mode being performed sequentially within one process. After performing processing within a "Part" sequentially, processing between "Parts" is performed without communication. Therefore, there are no restrictions on the number of "Parts", and analysis models that have been decomposed for parallel computation can be used as is to execute them (Fig. 4).

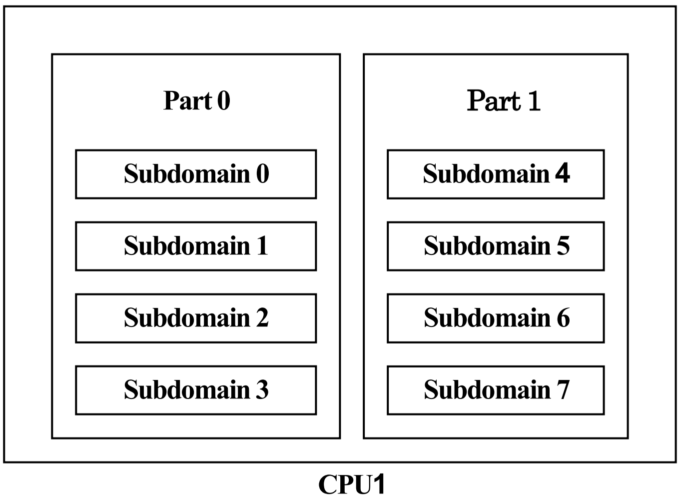


Fig. 4. Adjustment of Domain to CPUs (Single mode).

In the shared-memory parallel mode (`advmag2_HDDM_Electromagnetic-s_omp`), the single mode is shared memory parallelized by using OpenMP. "Subdomains" for each "Part" are divided among OpenMP threads.

2. 2. ADVENTURE_Metis

In the ADVENTURE_Metis, if the mesh is decomposed too finely, some "Subdomains" may not contain any elements. Also, if the mesh is decomposed too roughly compared to the total number of elements, it may take a long computational time, and amount of memory may be run out.

The computational performance of the ADVENTURE_Magnetic depends on the number of "Subdomains". Basically, the number of "Parts" is determined based on the computer environment, such as the parallel processing mode and the number of nodes used. The number of "Subdomains" is determined based on the computational time and memory usage. It has been confirmed that the ADVENTURE_Magnetic achieves relatively good performance in terms of computational time and memory usage when the number of elements per subdomain is around 100 to 200 [12].

The ADVENTURE_Metis is executed by specifying N_{part} (the number of "Parts") and $N_{subdomain}$ (the number of "Subdomains" per "Part"). If the total number of elements is $N_{element}$, $n_{element}$ (the number of elements per subdomain) is given by the following:

$$n_{element} = \frac{N_{element}}{N_{part} \times N_{subdomain}} \,. \tag{1}$$

N_{part} is determined by the parallel processing mode and computer environment, and $N_{element}$ is determined by the mesh of the model to be analyzed. Therefore, $N_{subdomain}$ is determined by the following:

$$N_{subdomain} = \frac{N_{element}}{N_{part} \times n_{element}} \,. \tag{2}$$

For $n_{element}$, use a value between 100 and 200 as shown above.

For example, if $n_{element}$ is set to 100 and a mesh with 800,000 elements (approximately 1 million DOFs) is analyzed using 4 MPI processes (4 "Parts") in the distributed-memory parallel mode, the number of "Subdomains" in each "Part" is obtained as follows:

$$\frac{800{,}000}{4 \times 100} = 2{,}000 \,. \tag{3}$$

In this case, ADVENTURE_Metis is executed as follows:

```
% mpirun -np 4 adventure_metis -HDDM -difn 1 data/model_one/input.adv data 2000
```

N_{part} and $N_{subdomain}$ are in red. Here, the MPICH is used as the MPI library, the list of machine names set in the system is used, and it is assumed that input and output is to the "data" directory. For details on the MPI library and the ADVENTURE_Metis, please refer to the respective manuals.

3. Flow of analysis

The flow of analysis using the ADVENTURE_Magnetic's modules is shown in Fig. 5.

(1) Creation of mesh data.

Mesh of the entire-type model data is prepared by ADVENTURE_CAD, ADVENTURE_TriPatch, ADVENTURE_TetMesh, commercial CAD, etc.

(2) Setting of boundary conditions.

Boundary conditions are set to mesh using the pre-processor module ADVENTURE_BCtool. The data of the extracted mesh surface groups (FaceGroup) are converted into GUI input binary format by using the `msh2pch` command. Then the boundary conditions are set up by the `ADVENTURE_BcGUI` command. For more details, see the manual of ADVENTURE_BCtool and Appendix A.3.

(3) Creation of the entire-type FEA model file.

The boundary conditions and material properties attached to mesh can be saved in an entire-type FEA model of the ADVENTURE binary format. In order to perform this operation, `advmag2_makefem_Electromagnetic` tool is used. See Appendix A.3for details of this tool. This tool is distributed with the current version of ADVENTURE_Magnetic.

(4) Domain decomposition.

Domain decomposition of the entire-type analysis model is done by ADVENTURE_Metis. When ADVENTURE_Metis is executed, the option `-difn 1` should be specified. This option specifies the DOFs of the internal boundary nodes to 1. For more information on domain decomposition, see Section 2.2.

(5) Electromagnetic field analysis

The HDDM-type model data are analyzed by finite element analysis solver ADVENTURE_Magnetic.

(6) Visualization of analysis results

The visualization files are created using the tool `advmag2_hddmmrg` included with ADVENTURE_Magnetic. In addition to files in ADVENTURE format, VTU files (ASCII) and legacy VTK files (Binary) can be created, which can be read by ParaView.

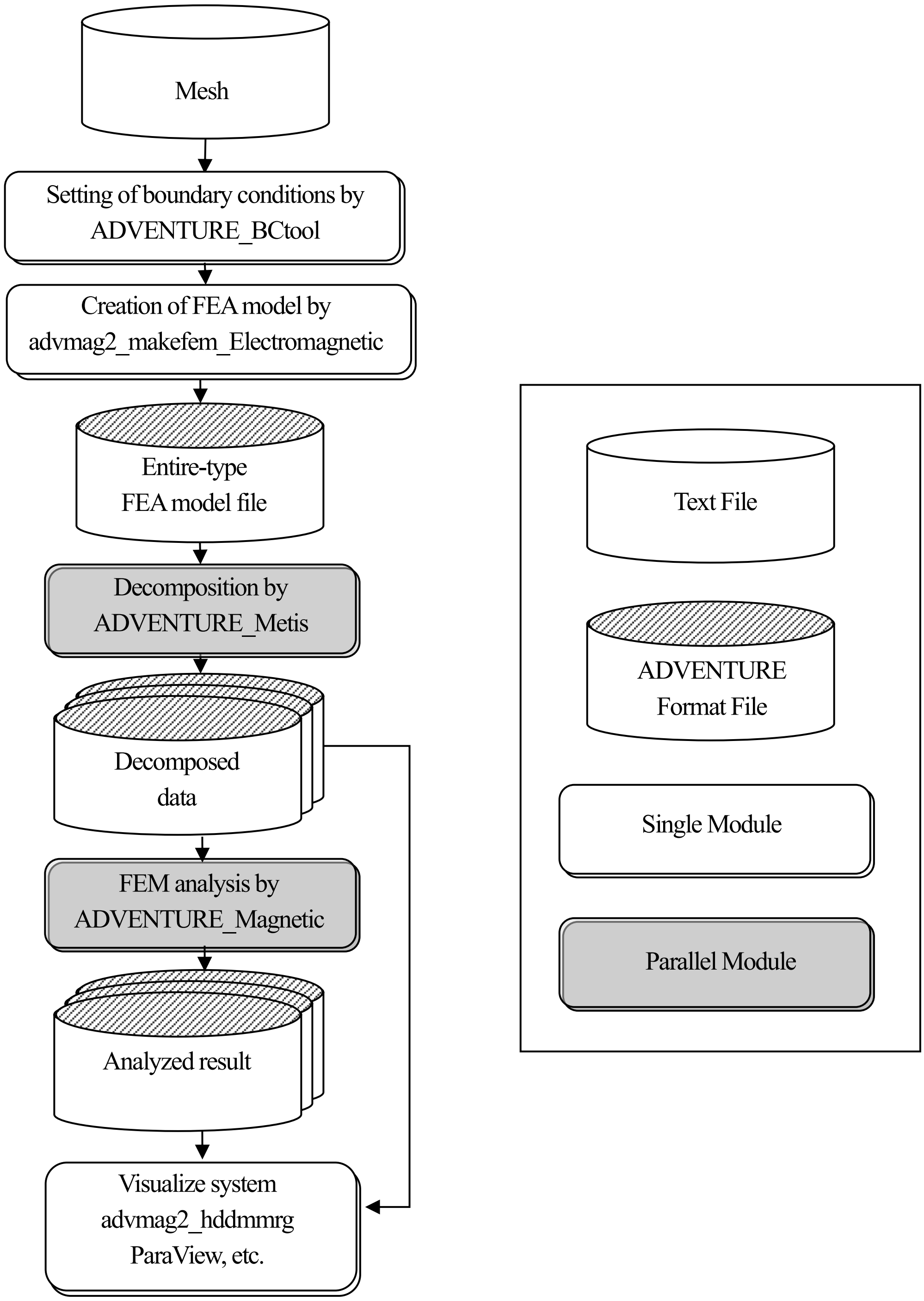


Fig. 5. Algorithm of Analysis Using ADVENTURE_Magnetic Module.

4. Input / Output files

4. 1. File names

Each input / output file name is as follows by default. "*data*" is the top directory name of the input / output files, and the input / output files related to the analysis are basically placed in this directory. This is called an analysis directory. These directory names and file names can be changed by command options. (See Section 6.2)

- Analysis derectory : *data*
- FEA model file : *data*/**model_one/input.adv**
- HDDM-type analysis model file : *data*/**model/advhddm_in_*P*.adv**
- Analysis result output settings file : *data*/**result/advhddm_out.adv**
- Analysis result output file : *data*/**result/advhddm_out_*P*.adv**
- Non-steady analysis result output file : *data*/**result/advhddm_out_*T*_*P*.adv**
- Initial values of non-steady analysis, settings file : *data*/**initial/advhddm_out.adv**
- Initial values of non-steady analysis file : *data*/**initial/advhddm_out_*P*.adv**
- Restart settings file : *data*/**restart*R*/advhddm_restart.adv**
- Restart data file : *data*/**restart*R*/advhddm_restart_*P*.adv**
- Material data file : *data*/**mtrl.dat**
- Convergence history file : *data*/**calc_log/log_g_***

Here, "***P***", "***T***" and "***R***" represent the part number, time step, and number of restarts, respectively.

In addition to these files, there are input files whose file names are specified within the material data file as shown below. File names of these files should be specified using a relative path from the subdirectory containing material data file (the directory specified in the options "`-mtrdat-dir`").

- Excitation current density data file
- Magnetization vector data file
- Shape definition file
- Characteristic curve file

For information on the format of these files, please refer to "Appendix B. Format of Input / Output files".

4. 2. Files required for each analysis

The files required for each analysis module are as follows:

- Common
  - Input files
    - HDDM-type analysis model file
    - Material data file
  - Output files
    - Analysis result output settings file
    - Convergence history file
  - Input / Output files
    - Restart settings file
    - Restart data file
- (Non-linear) magnetostatic analysis
  - Input files
    - (Optional) Excitation current density data file (coils)
Magnetization vector data file (permanent magnets)
Shape definition file (coils or permanent magnets)
: One or more files are required as electromagnetic field sources.
    - (Optional) Characteristic curve file : It is required for non-linear analysis
  - Output files
    - Analysis result output file
- Time-harmonic eddy current analysis
  - Input files
    - (Optional) Excitation current density data file (coils)
Magnetization vector data file (permanent magnets)
Shape definition file (coils or permanent magnets)
: One or more files are required as electromagnetic field sources.
  - Output files
    - Analysis result output file
- Non-steady eddy current analysis
  - Input files
    - Shape definition file (coils or permanent magnets)
    - (Optional) Initial values of non-steady analysis, settings file
Initial values of non-steady analysis file
: Output files of magnetostatic analysis or time-harmonic eddy current analysis
    - (Optional) Characteristic curve file : It is required for non-linear analysis
  - Output files
    - Non-steady analysis result output file
- High-frequency electromagnetic field analysis
  - Input files
    - (Optional) Excitation current density data file (coils)
Magnetization vector data file (permanent magnets)
Shape definition file (coils or permanent magnets)
: One or more files are required as electromagnetic field sources.
  - Output files
    - Analysis result output file

5. System of units

There is no function for specifying units in the input files (except for some angle unit specifications). Moreover, there is no function for converting units within the program. Therefore, it is necessary to determine a consistent system of units when creating input data.

When units are mentioned in this manual, such as in input/output file formats, they are all in the International System of Units (SI). If some of the base units are changed, derived units, etc. must also be converted without any inconsistencies.

Table 1. Conversion of units: when using mm instead of m as the unit of length

	Unit on this manual	After conversion	
Magnetic reluctivity	m/H	mm/H	1 m/H = 1,000 mm/H
Electrical conductivity	S/m	S/mm	1 S/m = 0.001 S/mm
Magnetic flux density / Magnetization vector	T = Wb/m^2	Wb/mm^2	1 T = 10^{-6} Wb/mm^2
Magnetic field	A/m	A/mm	1 A/m = 0.001 A/mm
Excitation current density	A/m^2	A/mm^2	1 A/m^2 = 10^{-6} A/mm^2

6. Command Options

The options available at execution are as follows. Here, *n*, *x*, and *s* following the option indicate that they are to be specified as an integer, real number, and string, respectively. For many options, the default values differ depending on the module.

6. 1. Common Options to All Modules

- `-memlimit` *n*

  This option specifies the upper limit of memory used by each process to *n* [MByte]. If this limit is exceeded, execution will be halted. The usage is limited to prevent system failures caused by greatly exceeding the installed memory amount, and significant increases in computational time due to paging. The default value is 1,000 for all modules. Please change it according to your environment and needs.

- `-v` or `-version`

  These options are used to display the version of modules.

- `-h` or `-help`

  - These options are used to display the help information.

- `-s` or `-settings`

  The setting values will be confirmed. Once this option has been processed, each module will stop execution.
  If the executable file is run with only this option, the default values for each option will be displayed.
  Also, if various options are given to the executable file and this option is attached at the end, you can check whether the settings are correct without performing any analysis.

- `-op-sw` *s*

  Within a module, the same option may be used for multiple functions or purposes. In such cases, this option switch is used to specify for which function or purpose the option is to be used. In most cases, each module has only one option switch, but some modules have multiple. If a module has multiple options, option switches are explained in the explanation of each module in Appendix A, so please refer to that.

6. 2. Options for Input / Output File names

The files used for each analysis were shown in Chapter 4. If you want to change the directory names or file names, use the following options. Here, "*dir*" is the directory name, and "*file*" is the file name. The names in parentheses after them are the default names. Also, "*P*" and "*T*" indicate the part number and time step, respectively.

These options cannot be specified for all modules, but can only be specified if the module uses the respective files.

- `-data-dir` *dir* (data)

  This option specifies the name of the analysis directory, which is the top directory for input / output files, to *dir*.

- `-onedata-dir` *dir* (model_one)

  This option specifies the name of the subdirectory containing the FEA model file, to *dir*.

- `-onedata-file` *file* (input)

  This option specifies the name of the FEA model file. The extension ".adv" will be added to *file*.

- `-model-dir` *dir* (model)

  This option specifies the name of the subdirectory containing the HDDM-type analysis model file, to *dir*.

- `-model-file` *file* (advhddm_in)

  This option specifies the name of the HDDM-type analysis model file. The string "_*P*.adv", consisting of the part number and the extension, is added to *file*.

- `-result-dir` *dir* (result)

  This option specifies the name of the subdirectory containing the analysis result output settings file, analysis result output file and non-steady analysis result output file, to *dir*.

- `-result-file` *file* (advhddm_out)

  The name of analysis result output settings file will be *file* with the extension ".adv".

  The name of analysis result output file will be *file* with the string "_*P*.adv" consisting of the part number and extension.

  The name of non-steady analysis result output file will be *file* with the string "_*T*_*P*.adv" consisting of the time step, part number, and extension.

- `-inivalue-dir` *dir* (initial)

  This option specifies the name of the subdirectory containing the initial values of non-steady analysis, settings file and initial values of non-steady analysis file, to *dir*.

- `-inivalue-file` *file* (advhddm_out)

  The name of initial values of non-steady analysis, settings file will be *file* with the extension ".adv".

  The name of initial values of non-steady analysis file will be *file* with the string "_*P*.adv" consisting of the part number and extension.

- `-mtrldat-dir` *dir* (./)

  This option specifies the name of the subdirectory containing the material data file, to *dir*.

- `-mtrldat-file` *file* (mtrl.dat)

  This option specifies the name of the material data file, to *file*.

- `-calc-log-dir` *dir* (calc_log)

  This option specifies the name of the subdirectory containing the convergence history file, to *dir*.

- `-calc-log-file` *file* (log_g)

  The name of convergence history file will be *file* with a keyword indicating the type of analysis.
  For example, for the time-harmonic eddy current analysis, the default name is "log_g_HDDM_TH_Eddy".

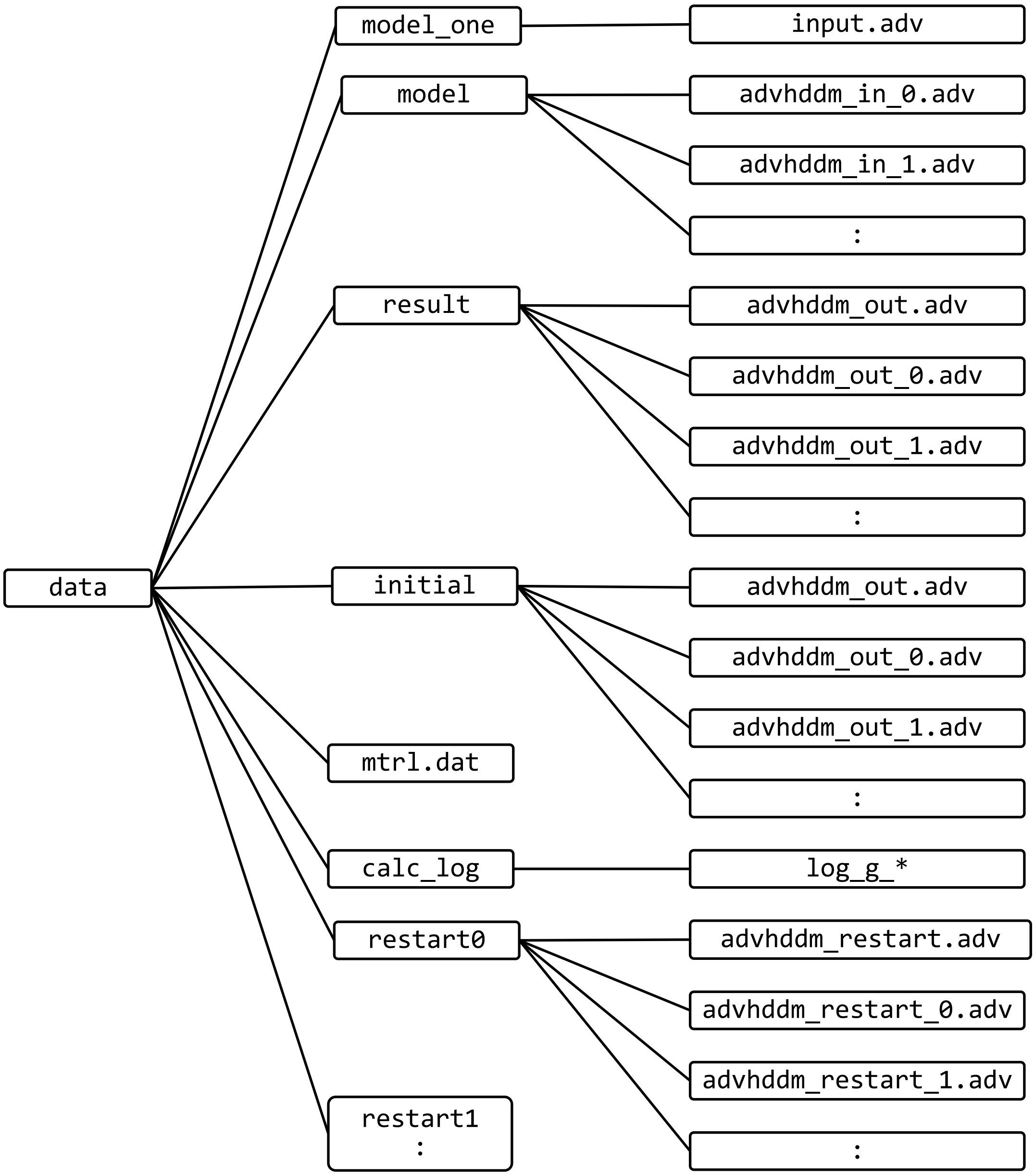


Fig. 6. Hierarchy within the analysis directory.
The directory names and file names are the default values.
The files in the "result" directory show the cases of steady analysis and quasi-steady analysis.

Fig. 6 shows the hierarchy within the analysis directory. The excitation current density data file, magnetization vector data file, shape definition file, and characteristic curve file, whose file names are specified within the material data file "`mtrl.dat`" using relative paths from the subdirectory containing the material data file (the directory specified with the option "`-mtrldat-dir`"), are not listed here. For information about "`restart`" at the bottom, see Section 6.5.1.

6. 3. Options for Non-steady analysis

These are options used when performing the ***N***on-***S***teady analysis with the solver module. The default values differ for each module, and even for each function in the same module, so please check it by using the option "`-s`" or "`-settings`".

- `-ns-delta-t` *x*

  This option specifies the time step size Δt, to *x*.

- `-ns-end-step` *n*

  This option specifies the final time step to be analyzed, to *n*.

- `-ns-start-step` *n*

  This option specifies the number of time steps at which to start the analysis, to *n*.

  Normally, the non-steady analysis starts from the first step using either 0 or the values read from the initial values of non-steady analysis file as the initial values (values for the 0th step). On the other hand, this option is used when you want to increase the number of time steps after finishing the previous analysis. When restarting the non-steady analysis, the analysis result output settings file and non-steady analysis result output file for the *n*-1th step of the previous analysis are required.

- `-ns-inivalue-type` *s*

  This option specifies the type of initial values for the non-steady analysis with the keyword *s*. The keywords that can be specified are as follows.

  - `zero` : Sets the initial values of all unknowns to 0.
  - `static` : The analysis result output file of the static analysis is used as the initial values of non-steady analysis file.
  - `real` : The analysis result output file of the time-harmonic analysis is used as the initial values of non-steady analysis file.
    The results obtained from the time-harmonic analysis are complex numbers.
    If this keyword is specified, the real part of the complex numbers will be used.
  - `imaginary` : The analysis result output file of the time-harmonic analysis is used as the initial values of non-steady analysis file.
    If this keyword is specified, the imaginary part of the complex numbers will be used.

- `-ns-out-interval` *n*

  This option specifies the interval at which the result files of the non-steady analysis are output, to *n*.

  This can be used to save disk space. However, please note that data for steps that were not output cannot be obtained unless the analysis is reperformed.

6.4. Options for Non-linear analysis

These are options used when performing the ***N***on-***L***inear analysis with the solver module. The default values differ for each module, and even for each function in the same module, so please check it by using the option "`-s`" or "`-settings`".

- `-nl-method` *s*

  This option specifies the solution method to be used for the non-linear analysis with the keyword *s*. The keywords that can be specified are as follows.
  - `None` : No non-linear analysis is performed; only a linear analysis is performed.
  - `Newton` : A non-linear analysis is performed using Newton's method (Newton-Raphson method).
  - `Picard` : A non-linear analysis is performed using Picard iteration.
  - `Explicit` : When performing a non-steady analysis that also supports non-linear analysis, an explicit non-linear analysis is performed to determine physical quantities using the results of the previous time step.

  The solution methods available depend on the analysis function.

- `-nl-param` *x*

  This option specifies the parameter for non-linear analysis, to *x*. It is not currently used by any module.

6.4.1. Options for non-linear analysis using Newton's method or Picard iteration

- `-nl-max-loop` *n*

  This option specifies the maximum number of non-linear iterations, to *n*.

- `-nl-conv` *x*

  This option specifies the convergence criterion for non-linear iterations, to *x*.

- `-nl-div` *x*

  This option specifies the divergence criterion for non-linear iterations, to *x*.

6.4.2. Options for non-linear analysis using Newton's method

- `-nl-newton-hor` *s*

  This option specifies the characteristic curve to be used in Newton's method the keyword *s*. The keywords that can be specified are as follows.
  - `Original` : A graph using the horizontal axis values as is, is used for the characteristic curve.
  - `Squared` : A graph using the squared values for horizontal axis is used for the characteristic curve. This graph is made from values entered from the characteristic curve data file within the module.

6.5. Options for Sparse Matrix Solvers

The solver module uses the parallel sparse matrix solvers for solving the interface problem in the HDDM and the non-parallelized sparse matrix solvers for solving the sparse matrices in the subdomains.

6.5.1. Options for Parallel Sparse Matrix Solvers

- `-hddm` *s*

  This option specifies the parallel sparse matrix solver for solving the interface problem in the HDDM with the keyword *s*. The keywords that can be specified are as follows. If the available matrices are limited, the types of matrices are also specified.

  - Common solvers
    - `CG` : Conjugate Gradient method (Real symmetric matrices, Hermitian matrices)
    - `CR` : Conjugate Residual method (Real symmetric matrices, Hermitian matrices)
    - `BiCG` : Bi-Conjugate Gradient method
    - `BiCR` : Bi-Conjugate Residual method
    - `COCG` : Conjugate Orthogonal CG method (Complex symmetric matrices)
    - `COCR` : Conjugate Orthogonal CR method (Complex symmetric matrices)
    - `MINRES` : Minimal Residual method (Real symmetric matrices, Hermitian matrices)
    - `MINRES-like_CS` : MINRES-like_CS method (Complex symmetric matrices)
    - `QMR` : Quasi-Minimal Residual method (Real symmetric matrices, Hermitian matrices)
    - `QMR_SYM` : QMR_SYM method (Complex symmetric matrices)
  - Product-type Krylov subspace methods : Squared type
    - `CGS` : CG Squared method
    - `CRS` : CR Squared method
    - `COCGS` : COCG Squared method (Complex symmetric matrices)
    - `COCRS` : COCR Squared method (Complex symmetric matrices)
  - Product-type Krylov subspace methods : Stabilized squared type
    - `SCGS` : Stabilized CGS method
    - `SCRS` : Stabilized CRS method
    - `SCOCGS` : Stabilized COCGS method (Complex symmetric matrices)
    - `SCOCRS` : Stabilized COCRS method (Complex symmetric matrices)
  - Product-type Krylov subspace methods : Stabilized type
    - `BiCGSTAB` : BiCG Stabilized method
    - `BiCRSTAB` : BiCR Stabilized method
    - `COCGSTAB` : COCG Stabilized method (Complex symmetric matrices)
    - `COCRSTAB` : COCR Stabilized method (Complex symmetric matrices)
  - Product-type Krylov subspace methods : Generalized type
    - `GPBiCG` : Generalized Product-type methods based on BiCG method
    - `GPBiCR` : Generalized Product-type methods based on BiCR method
    - `GPCOCG` : Generalized Product-type methods based on COCG method (Complex symmetric matrices)
    - `GPCOCR` : Generalized Product-type methods based on COCR method (Complex symmetric matrices)
  - Product-type Krylov subspace methods : STAB2 type
    - `BiCGSTAB2` : BiCGSTAB2 method
    - `BiCRSTAB2` : BiCRSTAB2 method
    - `COCGSTAB2` : COCGSTAB2 method (Complex symmetric matrices)
    - `COCRSTAB2` : COCRSTAB2 method (Complex symmetric matrices)

- `-hddm-param` *x*

  This option specifies the parameter of the parallel sparse matrix solver, to *x*. It is not currently used by any module.

- `-hddm-max-loop` *n*

  This option specifies the maximum number of iterations for the parallel sparse matrix solver, to *n*.

- `-hddm-conv` *x*

  This option specifies the convergence criterion of iterations for the parallel sparse matrix solver, to *x*.

- `-hddm-div` *x*

  This option specifies the divergence criterion of iterations for the parallel sparse matrix solver, to *x*.

- `-hddm-pc` *s*

  This option specifies the preconditioner for the parallel sparse matrix solver with the keyword *s*. The keywords that can be specified are as follows.

  - `None` : No preconditioner is used.
  - `Diag` : Simplified diagonal scaling is used as the preconditioner.

- `-hddm-pc-param` *x*

  This option specifies the parameter of the preconditioner for the parallel sparse matrix solver, to *x*. It is not currently used by any module.

- `-hddm-mat-type` *s*

  This option specifies the matrix storage format for the parallel sparse matrix solver with keyword *s*. The keywords that can be specified are as follows.

  - `implicit` : The matrix of the interface problem is not generated explicitly.

- `-hddm-log` / `-hddm-no-log`

  Whether or not to output the convergence history of the parallel sparse matrix solver to a convergence history file.

- `-hddm-keep-dom-mat` ／ `-hddm-no-keep-dom-mat`

  This option controls whether subdomain matrices are stored in memory or regenerated for each iteration of the parallel sparse solver. Storing them can reduce computational time but requires more memory.

- `-hddm-restart-out`

  The restart data required to restart the iteration of the parallel sparse matrix solver is output. The convergence criterion of the previous execution is not sufficient to produce accurate results, the iteration can be restarted with stricter convergence criterion. If no restart data is available, the iteration must be started from the beginning. Restarts can be repeated.

- `-hddm-restart` *n*

  This option specifies which restart data is used to restart iteration, to *n*. The restart count starts from 0 (not yet restarted). For example, the convergence criterion is updated and the execution is repeated as follows:

```
% advmag2_Electromagnetic HF_EM -hddm-conv 1.0e-05 -hddm-restart-out
% advmag2_Electromagnetic HF_EM -hddm-conv 1.0e-07 -hddm-restart-out -hddm-restart 0
% advmag2_Electromagnetic HF_EM -hddm-conv 1.0e-08 -hddm-restart-out -hddm-restart 1
```

  Note that the previous settings will be read from the restart setting file. Therefore, settings other than the convergence criterion and maximum number of iterations cannot be changed. In the above commands, the convergence history file names will be `log_g_HDDM0_HF_EM`, `log_g_HDDM1_HF_EM`, and `log_g_HDDM2_HF_EM`, respectively.

- `-hddm-restart-dir` *dir* (restart)

  This option specifies the name of the subdirectory containing the restart settings file and restart data file, to *dir* with the restart count. For each restart data recorded, it will be `restart0`, `restart1`, `restart2`, ...

- `-hddm-restart-file` *file* (advhddm_restart)

  The name of restart settings file will be *file* with the extension ".adv".

  The name of restart data file will be *file* with the string "_*P*.adv" consisting of the part number and extension.

6.5.2. Options for Non-parallelized Sparse Matrix Solvers

- `-solver` *s*

  This option specifies the non-parallelized sparse solvers for solving the sparse matrices in the subdomains with the keyword *s*. In addition to the keywords that can be specified for the parallel sparse matrix solver for solving the interface problem in the HDDM (option `-hddm`), the following keywords can be specified. If the available matrices are limited, the types of matrices are also specified.
  - Direct solvers
    - `LDL` : LDL^T (LDL^H) decomposition
      (Real symmetric matrices, Complex symmetric matricesHermitian matrices)
    - `LU` : *LU* decomposition
    - `LUp` : *LU* decomposition with pivoting.

  Note that if the direct method is applied to a matrix with singular, the computation will fail midway. Also, the memory usage will generally be much larger than with the iterative method.

- `-solver-param` *x*

  This option specifies the parameter of the non-parallelized sparse matrix solver, to *x*. It is not currently used by any module.

- `-solver-max-loop` *n*

  This option specifies the maximum number of iterations for the non-parallelized sparse matrix solver, to *n*.

- `-solver-conv` *x*

  This option specifies the convergence criterion of iterations for the non-parallelized sparse matrix solver, to *x*.

- `-solver-div` *x*

  This option specifies the divergence criterion of iterations for the non-parallelized sparse matrix solver, to *x*.

- `-solver-pc` *s*

  This option specifies the preconditioner for the non-parallelized sparse matrix solver with the keyword *s*. The keywords that can be specified are as follows. If the available matrices are limited, the types of matrices are also specified.
  - `None` : No preconditioner is used.
  - `Diag` : Diagonal scaling is used as the preconditioner.
  - `ICC` : Incomplete Cholesky factorization is used as the preconditioner.
    (Real symmetric matrices, Complex symmetric matricesHermitian matrices)
  - `iLU` : Incomplete *LU* factorization is used as the preconditioner.

- `-solver-pc-param` *x*

  This option specifies the parameter of the preconditioner for the non-parallelized sparse matrix solver, to *x*. It is used as an acceleration factor when the preconditioner is the Incomplete Cholesky factorization or the Incomplete *LU* factorization.

- `-solver-mat-type` *s*

  This option specifies the matrix storage format for the non-parallelized sparse matrix solver with keyword *s*. The keywords that can be specified are as follows.
  - `AIJ` : Matrices are recorded in AIJ format.

- `-solver-log` / `-solver-no-log`

  Whether or not to output the convergence history of the non-parallelized sparse matrix solver to a convergence history file.

Appendix

A. Details of how to use each module

A. 1. Electromagnetic field analysis modules (HDDM): advmag2_HDDM_Electromagnetic-*

There are four types of executable files depending on the parallelization method.

- `advmag2_HDDM_Electromagnetic-s` : Single mode
- `advmag2_HDDM_Electromagnetic-s_omp` : Shared-memory parallel mode
- `advmag2_HDDM_Electromagnetic-p` : Distributed-memory parallel mode
- `advmag2_HDDM_Electromagnetic-p_omp` : Hybrid parallel mode

Please refer to Chapter 4 for information about the input / output files of this module.

Local options:

- `-specificate-bc` *file*

  The analysis will be performed using the boundary conditions read from the boundary condition file (extension ".cnd", see the ADVENTURE_BCtool manual for details) specified by *file*. The boundary conditions read from the HDDM-type analysis model file will be discarded.

  To use this function, be sure to output FaceGroup data when creating the FEA model file with the tool for making FEA model `advmag2_makefem_Electromagnetic`.

- `-output-bc`

  The boundary conditions used in the analysis are output to the analysis result output file or the non-steady analysis result output file.

The following sections provide details on each electromagnetic field analysis.

A. 1. 1. (Non-linear) magnetostatic analysis

The linear or non-linear magnetostatic analysis is performed. The matrix is real symmetric.

When performing non-linear magnetostatic analysis, a characteristic curve data file that records a *B-H* characteristic curve (*B* is the magnetic flux density [T], *H* is the magnetic field [A/m]) is required. Based on the *B-H* characteristic curve, the nonlinearity of the magnetic reluctivity ν [m/H] (the reciprocal of the magnetic permeability μ [H/m]) is considered. Newton's method or Picard iteration can be used. In the Newton's method, if "`Original`" is specified with "`-nl-newton-hor`", the ν-B curve, and if "`Squared`" is specified, the ν-B^2 curve are calculated from the *B-H* characteristic curve read from the characteristic curve data file.

The *A* method with the magnetic vector potential *A* [Wb/m] as the unknown is solved in the magnetostatic analysis.

In the magnetostatic analysis, the following items are output to the analysis result output file. The units are the case when all input is in the International System of Units. Please convert them as necessary. The string in parentheses is the label in the ADVENTURE file.

- Unknown *A* (MagneticVectorPotential)
- Material ID for each element (MaterialID)
- Excitation current density given to the coils (CurrentDensity) [J/m^2] : If given
- Magnetization vector given to the permanent magnets (MagnetizationVector) [T] : If given
- Magnetic flux density (MagneticFluxDensity) [T]
- Nodal force (NodalForce) [N]
- Nodal force for material ID (NodalForce_MatID) [N]
- Magnetic reluctivity for each element (MagneticReluctivity) [m/H] (the reciprocal of the magnetic permeability)

A. 1. 2. Time-harmonic eddy current analysis

The time-harmonic eddy current analysis considering such as commercial power sources with relatively low frequencies, where the voltage changes like a sinusoid is performed. The time-harmonic problem (quasi-steady problem) is solved. In the time-evolution problem, a real matrix must be solved in each time step. However, that is converted to the quasi-steady problem that requires solving a complex matrix only once by replacing the time differential term $\partial/\partial t$ to $-i\omega$ or $i\omega$ (i: imaginary unit, ω: angular frequency of current [rad/s]). The matrix is complex symmetric.

Either the A method with the magnetic vector potential A [Wb/m] as the unknown, or the A-ϕ method with the electric scalar potential ϕ [V] is solved in the time-harmonic eddy current analysis. It is specified by the following options:

- `-formulation A` : A method
- `-formulation APhi` : A-ϕ method

Generally, the A-ϕ method uses several tens of percent more memory, but the iterative method converges faster and the computational time is shorter.

In the time-harmonic eddy current analysis, the following items are output to the analysis result output file. The units are the case when all input is in the International System of Units. Please convert them as necessary. The string in parentheses is the label in the ADVENTURE file.

- Unknown A (MagneticVectorPotential) : Complex numbers
- Unknown ϕ (ElectricScalarPotential) : Complex numbers, only in A-ϕ method
- Material ID for each element (MaterialID)
- Real part of excitation current density given to the coils (CurrentDensityReal) [J/m^2] : If given
- Imaginary part of excitation current density given to the coils (CurrentDensityImaginary) [J/m^2] : If given
- Real part of magnetization vector given to the permanent magnets (MagnetizationVectorReal) [T] : If given
- Imaginary part of magnetization vector given to the permanent magnets (MagnetizationVectorImaginary) [T] : If given
- Magnetic flux density (MagneticFluxDensity) [T] : Complex numbers
- Eddy current density (EddyCurrentDensity) [J/m^2] : Complex numbers
- Internal heat generation for each element (InternalHeatGeneration) [W/m^3]

A. 1. 3. Non-steady eddy current analysis

Time evolution problem considering such as commercial power sources with relatively low frequencies is considered. The matrix is real symmetric.

Either the A method, or the A-ϕ method is solved in the non-steady eddy current analysis. They can be specified in the same way as for time-harmonic eddy current analysis.

It is possible to perform explicit non-linear analysis, which determines physical quantities using the results of the previous time step. When performing a non-linear analysis, a characteristic curve data file that records the B-H characteristic curve (B is magnetic flux density [T], H is magnetic field [A/m]) is required. Based on the B-H characteristic curve, the nonlinearity of magnetic reluctivity ν [m/H] (the reciprocal of the magnetic permeability μ [H/m]) is considered.

In the non-steady eddy current analysis, the following items are output to the non-steady analysis result output file. The units are the case when all input is in the International System of Units. Please convert them as necessary. The string in parentheses is the label in the ADVENTURE file.

- Unknown A (MagneticVectorPotential)
- Unknown ϕ (ElectricScalarPotential) : Only in A-ϕ method
- Material ID for each element (MaterialID)
- Excitation current density given to the coils (CurrentDensity) [J/m^2] : If given
- Magnetization vector given to the permanent magnets (MagnetizationVector) [T] : If given
- Magnetic flux density (MagneticFluxDensity) [T]
- Eddy current density (EddyCurrentDensity) [J/m^2]
- Internal heat generation for each element (InternalHeatGeneration) [W/m^3]
- Nodal force (NodalForce) [N]
- Nodal force for material ID (NodalForce_MatID) [N]
- Magnetic reluctivity for each element (MagneticReluctivity) [m/H] (the reciprocal of the magnetic permeability)

A. 1. 4. High-frequency electromagnetic field analysis

High-frequency electromagnetic field analysis considering electromagnetic waves in the MHz to GHz range is performed. The time-harmonic problem (quasi-steady problem) is solved. The matrix is complex symmetric.

The E method with the electric field E [V/m] as the unknown is solved in the high-frequency electromagnetic field analysis.

In the high-frequency electromagnetic field analysis, the following items are output to the analysis result output file. The units are the case when all input is in the International System of Units. Please convert them as necessary. The string in parentheses is the label in the ADVENTURE file.

- Unknown E (ElectriFieldOnEDGE) : Complex numbers
- Material ID for each element (MaterialID)
- Real part of excitation current density given to the coils (CurrentDensityReal) [J/m^2] : If given
- Imaginary part of excitation current density given to the coils (CurrentDensityImaginary) [J/m^2] : If given
- Real part of magnetization vector given to the permanent magnets (MagnetizationVectorReal) [T] : If given
- Imaginary part of magnetization vector given to the permanent magnets (MagnetizationVectorImaginary) [T] : If given
- Magnetic field (MagneticField) [A/m] : Complex numbers
- Eddy current density (EddyCurrentDensity) [J/m^2] : Complex numbers
- Electric field (ElectricField) [V/m] : Complex numbers
- Internal heat generation for each element (InternalHeatGeneration) [W/m^3]

A. 2. Tool for displaying ADVENTURE format file (binary): `advmag2_advshow`

This tool extends `advshow`, which included with ADVENTURE_Solid, to be able to display complex numbers and quadruple-precision floating point numbers. The type of data stored in the Data area is determined by the “order” key, not the “format” key. The ADVENTURE file output by ADVNTURE_Magnetic contains both the “format” key and the “order” key, so it can also be displayed in `advshow`. However, in `advshow`, complex numbers are broken down into real and imaginary parts and displayed as two real numbers. Also, `advshow` cannot display quadruple-precision floating point numbers correctly.

Here's how to execute `advmag2_advshow`:

```
% advmag2_advshow [input ADVENTURE format file] [options]
```

`[input ADVENTURE format file]` : The name of the ADVENTURE file whose contents are to be displayed.

Local options:

- `-exponential-digit` *n*

  This option specifies the number of decimal places in exponential notation, to *n*. The default is 6, i.e. 7 significant digits.

- `-only-property`

  The Data area is not displayed, only the Property area is displayed. It works the same as `advinfo` included with ADVENTURE_IO.

- `-cpl2real`

  Complex numbers are displayed as two real numbers just like in `advshow`.

- `-output-file` *file*

  The contents are not displayed on the console but written to file.

- `-select-document` *s*

  Only Document containing the keyword specified by *s* is displayed. *s* is found in the “content_type” or “label” in Property area.

A. 3. Tool for making FEA model: `advmag2_makefem_Electromagnetic`

This is a tool to create the FEA model file that will be input to ADVENTURE_Metis.
Here's how to execute `advmag2_makefem_Electromagnetic`:

```
% advmag2_makefem_Electromagnetic [mesh] [fgr] [cnd] [mat] [adv] [options]
```

`[mesh]` : (Input) File that records the mesh (element connectivity, nodal coordinates, volume information).
Either MSH file (.msh), MSHX file (.mshx), or ADVENTURE file (.adv).
* The file format is determined from the file extension,
so please make sure there is no discrepancy between the file contents and the extension.

`[fgr]` : (Input) File that records the FaceGroups on the mesh surface.
Either FGR file (.fgr) or FGRX file (.fgrx).
* The file format is determined from the file extension,
so please make sure there is no discrepancy between the file contents and the extension.

`[cnd]` : (Input) Boundary condition file (.cnd) that describes the boundary conditions.
For more information, please refer to the ADVENTURE_BCtool manual.

`[mat]` : (Input) Material properties data file (.dat).
For more information, please refer to the ADVENTURE_BCtool manual.

`[adv]` : (Output) FEA model file (.adv).

Local options:

- `-dim` *n*

Since it is not possible to determine the dimension of a mesh from the MSH file, when reading a mesh that is not 3D from the MSH file, the dimension should be specified as *n* with this option.

- `-crd-magnification` *x*

This option specifies the magnification factor as *x* when converting the unit of node coordinates. If the length unit of the design drawing is mm and the CAD data is also created in mm, you can convert it to m. For example, conversion from mm to m can be specified using one of the following:

```
-crd-magnification 0.001
-crd-magnification 1.0e-03
```

- `-bc-on-ef`

This option outputs boundary conditions as data on ***e**lement **f**aces*. It is incompatible with "`-bc-on-nd`". This option must be specified to use mesh refinement during domain decomposition.

- `-bc-on-nd`

This option outputs boundary conditions as data on ***n**o**d**es*. It is incompatible with "`-bc-on-ef`".

- `-no-need-bc`

This option does not output boundary conditions. Since the size of the FEA model file is reduced, it is expected that the execution time of ADVENTURE_Metis will be shorter, but it is necessary to specify the "`-specificate-bc`" option when executing the solver module.

- `-face-group` / `-wo-face-group`

These options output/do not output FaceGroup data to the FEA model file. If FaceGroup data is present, the "`-specificate-bc`" option can be specified when executing the solver module. Changing the boundary conditions eliminates the need to recreate the FEA model file and redo the domain decomposition.

A. 4. Tool for integrating domain decomposed data and creating visualization file: `advmag2_hddmmrg`

The analysis result output file and non-steady analysis result output file output by the solver module contains domain decomposed data, just like the input HDDM-type analysis model file. This tool integrates the domain decomposed mesh and analysis results, and either the ADVENTURE file, the VTU file (text), or the Legacy VTK file (binary) is output. The VTU and Legacy VTK files can be read by visualization software such as ParaView.

The settings are read from the analysis result output settings file. Therefore, when using the analysis results of ADVENTURE_Magnetic, there is no need to set the final step of the non-steady analysis, etc. again. If you are using analysis results other than ADVENTURE_Magnetic, there is no analysis result output settings file, so please provide the necessary settings in the command options along with the local options "`-sw-NonSteady`" and "`-sw-DomainDecomposition`".

Here's how to execute `advmag2_hddmmrg`:

`% advmag2_hddmmrg [output file] [Num. of labels (0: all labels)] (labels...) [`*`options`*`]`

`[output file]` : Output file name. The file type is determined by the extension.

.adv: ADVENTURE file.

A new analysis directory will be created from the output file name given here, and files will be output there.

Ex.) If the file is "hoge.adv", then "hoge" will become the analysis directory name.

Real numbers will be the same precision as output by the solver module.

.vtu: VTU file (text). Real numbers are expressed as exponents with 7 significant digits.

.vtk: Legacy VTK file (binary). Real numbers are single-precision.

`[Num. of labels (0: all labels)] (labels...)` :

After specifying the number of labels for the physical quantities you want/don't want to output, line up that many labels.

If the number of labels is …

positive : only the physical quantities of the specified labels will be output.

negative : physical quantities other than the specified labels will be output.

0 : all physical quantities will be output.

Ex. 1) Output "MagneticFluxDensity" and "EddyCurrentDensity" to "hoge.adv".

```
% advmag2_hddmmrg hoge.adv 2 MagneticFluxDensity EddyCurrentDensity
```

Ex. 2) Output all quantities except the unknown "MagneticVectorPotential" to "hoge.vtu".

```
% advmag2_hddmmrg hoge.vtu -1 MagneticVectorPotential
```

Ex. 3) Output all physical quantities to "hoge.vtk".

```
% advmag2_hddmmrg hoge.vtk 0
```

Local options:

- `-no-output`

  No file is output. The following uses are expected.

  - For a mesh with large degrees of freedom, creating a visualization file for the entire mesh would make the file size too large or make visualization difficult, so a visualization file for only a portion of the mesh is created using "`-cut-part`" or "`-make-plane`" below.
  - When only analyzing the analysis results using "`-analyze-result`" below.

Note that even if no file is output, [output file] cannot be omitted.

- `-cut-part` *n*

  Only the element group in the material ID specified by *n* and the physical quantities related to the nodes belonging to that element group will be extracted and output to a file. The file name will be created from the name specified in `[output file]` as follows.

  Ex.) If *n* is specified as 3: hoge.vtu -> hoge_MatID3.vtu

  Material IDs are assumed to be 0 or greater. As a special case, if *n* is specified as "-1", a file will be output for each material ID.

  Ex.) If the material IDs in the mesh are 0 to 2,
  hoge_MatID0.vtu, hoge_MatID1.vtu, and hoge_MatID2.vtu will be output.

  Note that if *n* is specified as "-2", extraction by material ID will be disabled.

- `-make-plane` *s x*

  Only the physical quantities on the surface cut by a plane perpendicular to one of the coordinate axes are output to the file. However, the specified plane must be the boundary of the mesh (where the element faces line up neatly). In other words, there is no function to cut elements with a plane and interpolate, so it is impossible to cut with an arbitrary surface. If the analysis domain is a rectangular prism and the mesh is automatically generated, it is possible to create a plane only on the outer six faces. If you want to cut in the middle of the analysis domain, you must split it with the surface you want to cut when creating the CAD data, and then generate the mesh. If the mesh is created by stacking rectangular prisms, the cut will be made at the boundary between the rectangular prisms.

  s is the coordinate axis, and *x* is the coordinate value. They are specified as follows.

  Ex.) To cut with the plane of $y = 10.0$: `-make-plane y 10.0`
  File name: hoge.vtu -> hoge_y_10.0.vtu
  * The coordinate values are reflected in the file name as they are entered.

- `-analyze-result`

  The maximum, minimum and average values of the physical quantities are displayed in the console. If the physical quantity is a vector, the maximum, minimum and average values of each component are also displayed.

- `-sw-NonSteady`, `-sw-DomainDecomposition`

  This option is used when using analysis results other than ADVENTURE_Magnetic. For non-steady analysis, "`-sw-NonSteady`" should be specified. Since there is no analysis result output settings file, the settings for the non-steady analysis cannot be read from the file. Therefore, if the time step size, final time step, and interval for outputting results are different from the default values, they must be set with the respective command options (`-ns-delta-t`, `-ns-end-step`, `-ns-out-interval`). Also, if the data is decomposed, "`-sw-DomainDecomposition`" must be specified.

  Note that these options are necessary for analysis results from ADVENTURE_Magnetic Ver.1.9.2 and earlier, as the specifications of the analysis result output settings file are different.

A. 5. Tool for computing source vector: `advmag2_SourceVector`

This is a tool to check whether the shape definition file is written correctly before performing an analysis. It outputs the value of the source vector (excitation current density or magnetization vector in the electromagnetic field analysis module) at the specified coordinates. It does not read the mesh, so it does not screen by material IDs.

Here's how to execute `advmag2_SourceVector`:

```
% advmag2_SourceVector [Shape definition file] [Nodes file] [SourceVector file] [options]
```

`[Shape definition file]` : (Input) Shape definition file. Specify the file to be checked.

`[Nodes file]` : (Input) File containing coordinates of point group.
The values of source vector on the points recorded in this file will be output.
Enter the number of points and their respective coordinates (3D) as follows.
* The parts in red are explanations and will not be entered in actuality.
* ":" indicates omission.

10			<- Num. of points
0.0	0.0	0.0	<- Coordinate of 0th point
1.0	0.0	0.0	<- Coordinate of 1st point
:			
:			
0.0	0.0	1.0	<- Coordinate of 9th point

`[SourceVector file]` : (Output) The values of source vector will be output in the same format as `[Nodes file]`
(Read "Coordinate of …th point" as "Value of source vector of …th point")
If the extension is set to ".adv", it will be output in ADVENTURE format.

Local options:

- `-complex-sv`

  For quasi-steady problems that solve complex matrices, the source vector must also be given as a complex number. If you want to check the shape definition file for quasi-steady problems, use this option. It is not compatible with "`-time-evolution`".

  The name of the output file will be the file name specified in `[SourceVector file]` with "_re" (for the real part) or "_im" (for the imaginary part) appended.

  Ex.) hoge.dat -> hoge_re.dat, hoge_im.dat

- `-time-evolution`

  Use this option if you want to check the shape definition file for non-steady analyses. If the time step size and the final time step are different from the default values, set them with the respective command options (`-ns-delta-t`, `-ns-end-step`). Note that this option is incompatible with "`-complex-sv`".

  The name of the output file will be the file name specified in `[SourceVector file]` with the number of time steps appended.

  Ex.) hoge.dat -> hoge_step0.dat, hoge_step1.dat, …

- `-each-point`

  If "`-time-evolution`" is specified at the same time, the time history of the source vector will be output to a file for each point. The name of the output file will be the file name specified in `[SourceVector file]` with the point number appended.

  Ex.) hoge.dat -> hoge_point0.dat, hoge_point1.dat, …

Each line describes the time, and the *x*, *y*, and *z* values of the source vector, so it can be visualized using gnuplot, etc.

Note that even if the extension is set to ".adv", the data will still be output as text. Also, files for each time step will not be output.

A. 6. Tool for comparing analysis results: `advmag2_compare_results`

This tool compares the results of analyses performed under different conditions using the same mesh. The relative error of each physical quantity is output as a percentage.

Here's how to execute `advmag2_compare_results`:

```
% advmag2_compare_results [Num. of labels (0: all labels)] (labels...) [options]
```

`[Num. of labels (0: all labels)] (labels...)` :

After specifying the number of labels for the physical quantities you want to compare/not compare, line up that many labels.

If the number of labels is …

- positive : only the physical quantities of the specified labels will be compared.
- negative : physical quantities other than the specified labels will be compared.
- 0 : all physical quantities will be compared.

Ex. 1) Compare "MagneticFluxDensity" and "EddyCurrentDensity".

```
% advmag2_ compare_results 2 MagneticFluxDensity EddyCurrentDensity
```

Ex. 2) Compare all quantities except the unknown "MagneticVectorPotential".

```
% advmag2_ compare_results -1 MagneticVectorPotential
```

Ex. 3) Compare all physical quantities.

```
% advmag2_ compare_results 0
```

This tool has two option switches.

- `Exact` : Switch for the reference solution.
- `Approximation` : Switch for the comparison target.

Ex.) When a time-harmonic eddy current analyses with the analysis directory "cake" are performed using the COCG and COCR methods, the relative error of the eddy current density (EddyCurrentDensity) [J/m^2] is computed using the results from the COCG method as the reference solution.

```
% advmag2_HDDM_Electromagnetic-s TH_Eddy -data-dir cake -hddm COCG
```

* Time-harmonic eddy current analysis is performed using the COCG method (`-hddm COCG`) and the cake directory as the analysis directory (`-data-dir cake`).

```
% mv cake/result/ cake/COCG
```

* The subdirectory in which the analysis results are stored is renamed.

```
% advmag2_HDDM_Electromagnetic-s TH_Eddy -data-dir cake -hddm COCR
```

* Time-harmonic eddy current analysis is performed using the COCR method (`-hddm COCR`) and the cake directory as the analysis directory.

```
% advmag2_compare_results 1 EddyCurrentDensity -op-sw Exact -data-dir cake -result-dir
```

* The analysis directory for the reference solution is set to cake, and the subdirectory storing the analysis results is set to COCG. (`-op-sw Exact -data-dir cake -result-dir COCG`)

* The analysis directory to be compared will be set to cake. The subdirectory storing the analysis results will be the default "result". (`-op-sw Approximation -data-dir cake`)

This tool has no local options.

A. 7. Tool for converting to material properties data file: `advmag2_makedat`

This tool creates a material properties data file, which is input to the tool for making FEA model, from a material data file. If physical quantities are managed using a spreadsheet program, it is often easier to convert to a material data file. In such cases, it is assumed that a material data file will first be created, and then this tool will be used to convert it to a material properties data file. However, since the electromagnetic field analysis module does not reference the physical quantities described in the material properties data file, this tool is used to create material properties data files to be used in analysis modules other than ADVENTURE_Magnetic.

Here's how to execute `advmag2_makedat`:

```
% advmag2_makedat [original MTRL file] [output mat file] [options]
```

`[original MTRL file]` : (Input) Material data file.
`[output mat file]` : (Output) Material properties data file

This tool has no local options.

A. 8. Tool for converting file format: `advmag2_mesh_convert_file`

This tool converts the format of files that record mesh data or FaceGroup data on mesh surfaces. If mesh data is input, it can also output FaceGroup data on the mesh surface. However, unlike `msh2pch` included with ADVENTURE_BCtool, it outputs all surface patches as a single FaceGroup.

The format of the input and output files is determined by the file extension. It will not work correctly if there is a discrepancy between the contents of the input file and the file extension.

In addition, it does not support HDDM-type analysis model files.

The following combinations of input and output files can be converted using `advmag2_mesh_convert_file`.

Combination 1:

Input : Mesh data (extensions: .msh, .mshx or .adv)

Output : Mesh data (extensions: .msh, .mshx or .adv)

* If the ADVENTURE file contains data other than mesh data,
all data other than the mesh data will be lost when converted to an MSH or MSHX file.

FaceGroup data (extensions: .fgr or .fgrx)

Visualization file (extensions: .vtu or .vtk)

* If the ADVENTURE file contains data other than mesh data,
all data other than the mesh data will be lost when converted to an VTU or Legacy VTK file.

Combination 2:

Input : FaceGroup data (extensions: .fgr or .fgrx)

Output : FaceGroup data (extensions: .fgr or .fgrx)

Here's how to execute `advmag2_mesh_convert_file`:

```
% advmag2_mesh_convert_file [input file] [output file] [options]
```

`[input file]` : Name of input file. See above for possible input files.

`[output file]` : Name of output file. See above for possible output files.

Local options:

- `-dim` *n*

  Since it is not possible to determine the dimension of a mesh from the MSH file, when reading a mesh that is not 3D from the MSH file, the dimension should be specified as *n* with this option.

- `-no-output`

  No file is output. This option is specified when a file of only a part of the mesh is created using "`-cut-part`" or "`-make-plane`" below, or when only a visualization file of FaceGroup data is created using "-fgr". Note that even if no file is output, `[output file]` cannot be omitted.

- `-cut-part` *n*

  This option is only available if the input is mesh data.

  Only the element group in the material ID specified by *n* and the nodes belonging to that element group will be extracted and output to a file. The file name will be created from the name specified in `[output file]` as follows.

  Ex.) If *n* is specified as 3: hoge.vtu -> hoge_MatID3.vtu

  Material IDs are assumed to be 0 or greater. As a special case, if *n* is specified as "-1", a file will be output for each material ID.

  Ex.) If the material IDs in the mesh are 0 to 2,
  hoge_MatID0.vtu, hoge_MatID1.vtu, and hoge_MatID2.vtu will be output.

  Note that if *n* is specified as "-2", extraction by material ID will be disabled.

- `-make-plane` *s x*

  This option is only available if the input is mesh data.

  Only the element faces on the surface cut by a plane perpendicular to one of the coordinate axes are output to the file. However, the specified plane must be the boundary of the mesh (where the element faces line up neatly). In other words, there is no function to cut elements with a plane and interpolate, so it is impossible to cut with an arbitrary surface. If the analysis domain is a rectangular prism and the mesh is automatically generated, it is possible to create a plane only on the outer six faces. If you want to cut in the middle of the analysis domain, you must split it with the surface you want to cut when creating the CAD data, and then generate the mesh. If the mesh is created by stacking rectangular prisms, the cut will be made at the boundary between the rectangular prisms.

  s is the coordinate axis, and *x* is the coordinate value. They are specified as follows.

  Ex.) To cut with the plane of $y = 10.0$: `-make-plane y 10.0`

  File name: hoge.vtu -> hoge_y_10.0.vtu

  * The coordinate values are reflected in the file name as they are entered.

- `-fgr` *file0 file1*

  This option is used to create a visualization file of FaceGroup data. The input file is mesh data, and *file0* is the name of the file that records the FaceGroup data of that mesh data. *file1* should be the name of the visualization file (extension .vtu or .vtk).

- `-fgr-vtu` ／ `-fgr-vtk`

  If these options are specified when converting mesh data to FaceGroup data, the visualization file of FaceGroup data will be output in VTU format or Legacy VTK format. The file names are as follows.

  Ex.) Name specified as output file: hoge.fgr -> hoge_FGR.vtu / hoge_FGR.vtk

A. 9. Tool for analyzing mesh information: `advmag2_mesh_property`

This is a tool to analyze mesh data information. It outputs the following information about the input mesh:

- Number of elements, number of nodes (total number, number of 1ry nodes, number of 2ry nodes)
- Element type
- Range in which the mesh exists (minimum and maximum values of *x*-coordinate, *y*-coordinate, and *z*-coordinate)
- Mesh volume
- Information on the above four items for each material ID

Note that HDDM-type analysis model files are not supported.

Here's how to execute `advmag2_mesh_property`:

```
% advmag2_mesh_property [input file] [options]
```

`[input file]` : Name of the input file.
Either an MSH file (.msh), an MSHX file (.mshx), or an ADVENTURE file (.adv).

Local options:

- `-dim` *n*

Since it is not possible to determine the dimension of a mesh from the MSH file, when reading a mesh that is not 3D from the MSH file, the dimension should be specified as *n* with this option.

- `-fgr` *file*

The FaceGroup data stored in the file specified by *file* will be analyzed. FGR files (.fgr) and FGRX files (.fgrx) can be input. If the FaceGroup data is not from the mesh data specified by `[input file]`, it will not work correctly.

A. 10. Tool for cutting part of mesh: `advmag2_mesh_cut_part`

This tool extracts only the element group with a specified property number and the data related to the nodes that belong to that element group from the original mesh data, constructs a new mesh, and outputs it. The tool for converting file format `advmag2_mesh_convert_file` can also extract property numbers, but this tool is more versatile.

Note that HDDM-type analysis model files are not supported.

Here's how to execute `advmag2_mesh_cut_part`:

```
% advmag2_mesh_cut_part [input file] [output file] [Num. of Volumes] (Vol.num...) [options]
```

`[input file]` : Name of the input file.
Either an MSH file (.msh), an MSHX file (.mshx), or an ADVENTURE file (.adv).
* If the ADVENTURE file contains data other than mesh data,
all data other than the mesh data will be lost.

`[output file]` : Name of the output file.
Either an MSH file (.msh), an MSHX file (.mshx), or an ADVENTURE file (.adv).
* In the case of ADVENTURE files, the indexes of element numbers, node numbers,
and material IDs with the original mesh are recorded with the labels
"CutPart_ElementIndex", "CutPart_NodeIndex", and "CutPart_MaterialID_Index",
respectively. "`20: 369`" means that "20" is "369" in the original mesh.

`[Num. of Volumes] (Vol.num...)` :
After specifying the number of volumes you want to cut/not cut, line up that many material IDs.
If the number of volumes is …
positive : only meshes with the material IDs will be cut out.
negative : meshes other than the material IDs will be cut out.

Ex. 1) Cut out mesh with two material IDs 5 and 6 from hoge.msh and output it to hoge.adv.
```
% advmag2_mesh_cut_part hoge.msh hoge.adv 2 5 6
```
Ex. 2) Cut out mesh other than material IDs 0 and 1 from hoge.adv and output it to hoge.mshx.
```
% advmag2_mesh_cut_part hoge.adv hoge.mshx -2 0 1
```

Local options:

- `-dim` *n*

Since it is not possible to determine the dimension of a mesh from the MSH file, when reading a mesh that is not 3D from the MSH file, the dimension should be specified as *n* with this option.

- `-indexes` *file*

The indexes of element numbers, node numbers, and material IDs with the original mesh are output to *file*. If the file extension is .adv, it is output in ADVENTURE format, otherwise it is output as text data. This is used when the output file `[output file]` is an MSH file (.msh) or MSHX file (.mshx) and the indexes are required.

A. 11. Tool for merging meshes: advmag2_mesh_merge

This tool combines two meshes and outputs them as one mesh. Note that HDDM-type analysis model files are not supported.

It is not possible to combine linear and quadratic elements. If meshes of different orders are entered, an error message will be displayed and the module will be forcibly terminated.

Nodes with the same coordinate between the two meshes will be combined as the same node. To account for numerical errors, the length of the shortest element side is multiplied by a value (`ratio`) that can be set with the option "`-ratio-threshold`", and the squared value is used as the threshold value to determine whether the coordinates match. In other words, if the node coordinates of mesh 0 and 1 are ($x0, y0, z0$) and ($x1, y1, z1$), if

$\{(x0 - x1)^2 + (y0 - y1)^2 + (z0 - z1)^2\}$ < (length of shortest element side× `ratio`)^2,

then they will be considered to be the same node. If the combination is not successful, try changing `ratio`.

If the mesh generated by this tool is to be used in finite element analysis, the element face shapes within the joining surface of the two meshes must match. However, this tool does not check whether the element faces match. Therefore, it is possible to connect elements that do not have connecting faces or whose element faces are not supposed to match, such as a tetrahedron and a hexahedron. Please use this function with full understanding.

Here's how to execute `advmag2_mesh_merge`:

```
% advmag2_mesh_merge [input file 0] [input file 1] [output file] [options]
    [input file 0], [input file 1]   :
```

: The names of the two input files you want to merge.
Either an MSH file (.msh), an MSHX file (.mshx), or an ADVENTURE file (.adv).
* If the ADVENTURE file contains data other than mesh data,
all data other than the mesh data will be lost.

`[output file]` : The names of the two input files you want to merge.
Either an MSH file (.msh), an MSHX file (.mshx), or an ADVENTURE file (.adv).
If a VTU file (.vtu) or Legacy VTK file (.vtk) is specified,
only the visualization file will be output.
* In the case of ADVENTURE files, the indexes of element numbers, node numbers,
and material ID for the mesh before merging are recorded with the labels
"Merge_ElementIndex", "Merge_NodeIndex", and "Merge_MaterialID_Index",
respectively.
"`350: 1 25`" means that "350" after merging is "25" in `[input file 1]`.
The indexes for the node numbers before merging and after merging are also recorded
with the labels "Merge_NodeIndex_Mesh0ToMerged"
and "Merge_NodeIndex_Mesh1ToMerged", respectively.
"`15: 26`" means that "15" before merging is "26" after merging.

Local options:

- `-dim` *n*

  Since it is not possible to determine the dimension of a mesh from the MSH file, when reading a mesh that is not 3D from the MSH file, the dimension should be specified as *n* with this option.

- `-ratio-threshold` *x*

  This option specifies the value *x* to be used to judge whether the coordinates match. See above for details on how to judge whether a coordinate matches.

- `-fgr` *file0 file1 file2*

  FaceGroup data (extensions: .fgr, .fgrx) will also be merged along with the meshes. Specify the names of the file that stores the FaceGroup data of `[input file 0]` as *file0*, and `[input file 1]` as *file1*. Also specify the name of the file to output the merged FaceGroup data as *file2*. Unless the next option "`-fgr-no-delete`" is given, the FaceGroup data on the merged faces will be deleted.

- `-fgr-no-delete`

  This option leaves the FaceGroup data on the merging faces intact when combining FaceGroup data using the option "`-fgr`".

- `-indexes` *file*

  The indexes of element numbers, node numbers, and material IDs for the mesh before merging are output to *file*. Also the index of the node numbers before merging and after merging is output. If the file extension is .adv, the output is in ADVENTURE format, otherwise it is output as text data. This is used when the output file `[output file]` is an MSH file (.msh) or MSHX file (.mshx) and the indexes are required.

A. 12. Tools for affine transformation of meshes

There are tools for performing affine transformation of meshes include a tool for translation, a tool for scaling a tool for rotation, and a tool for mirror. There is no tool for shearing, as this may significantly degrade the quality of the mesh. Also, HDDM-type analysis model file s are not supported.

These tools have common local options.

Common local options:

・ `-dim` *n*

Since it is not possible to determine the dimension of a mesh from the MSH file, when reading a mesh that is not 3D from the MSH file, the dimension should be specified as *n* with this option.

A. 12. 1. Tool for translation: `advmag2_mesh_translation`

This tool performs translation, one of the affine transformations for meshes.

Here's how to execute `advmag2_mesh_translation`:

```
% advmag2_mesh_translation [input file] [output file] [Tx] [Ty] [Tz] [options]
```

`[input file]` : Name of input file. Either MSH file (.msh), MSHX file (.mshx), or ADVENTURE file (.adv).

`[output file]` : Name of output file. Either MSH file (.msh), MSHX file (.mshx), or ADVENTURE file (.adv). Also, if a VTU file (.vtu) or Legacy VTK file (.vtk) is specified, only the visualization file will be output.

`[Tx] [Ty] [Tz]` : Enter the amount of translation in the *x*, *y*, and *z* directions, respectively.

A. 12. 2. Tool for scaling: `advmag2_mesh_scaling`

This tool performs scaling, one of the affine transformations for meshes.

Here's how to execute `advmag2_mesh_scaling`:

```
% advmag2_mesh_scaling [input file] [output file] [Sx] [Sy] [Sz] [options]
```

`[input file]` : Name of input file. Either MSH file (.msh), MSHX file (.mshx), or ADVENTURE file (.adv).

`[output file]` : Name of output file. Either MSH file (.msh), MSHX file (.mshx), or ADVENTURE file (.adv). Also, if a VTU file (.vtu) or Legacy VTK file (.vtk) is specified, only the visualization file will be output.

`[Sx] [Sy] [Sz]` : Specify magnification ratios in the x, y, and z directions that are greater than 0. 0 and negative values cannot be specified as they will cause the mesh to fail. If the same value is specified for all magnification ratios, it can also be used to convert units of coordinates.

A. 12. 3. Tool for rotation: `advmag2_mesh_rotation`

This tool performs rotation, one of the affine transformations for meshes.
Here's how to execute `advmag2_mesh_rotation`:

```
% advmag2_mesh_rotation [input file] [output file] [axis of rotation (x, y or z)]
 [unit of angle (rad or deg)] [angle of rotation] [options]
```

`[input file]` : Name of input file. Either MSH file (.msh), MSHX file (.mshx), or ADVENTURE file (.adv).

`[output file]` : Name of output file. Either MSH file (.msh), MSHX file (.mshx), or ADVENTURE file (.adv). Also, if a VTU file (.vtu) or Legacy VTK file (.vtk) is specified, only the visualization file will be output.

`[axis of rotation (x, y or z)]` : Specify the axis of rotation as either “*x*”, “*y*”, or “*z*”.

`[unit of angle (rad or deg)]` : Enter the unit of the angle to be entered after this in either “`rad`” (radian) or “`deg`” (degree).

`[angle of rotation]` : Specify the angle to rotate by.

A. 12. 4. Tool for mirror: `advmag2_mesh_mirror`

This tool performs reflection, one of the affine transformations for meshes. It can create a mirror image on a plane perpendicular to the *x*-axis, *y*-axis, or *z*-axis. It can transform not only coordinates but also element connectivity.

Here's how to execute `advmag2_mesh_mirror`:

```
% advmag2_mesh_mirror [input file] [output file] [direction (x, y or z)] [coordinate]
[options]
 [unit of angle (rad or deg)] [angle of rotation] [options]
```

`[input file]` : Name of input file. Either MSH file (.msh), MSHX file (.mshx), or ADVENTURE file (.adv).

`[output file]` : Name of output file. Either MSH file (.msh), MSHX file (.mshx), or ADVENTURE file (.adv). Also, if a VTU file (.vtu) or Legacy VTK file (.vtk) is specified, only the visualization file will be output.

`[direction (x, y or z)]` : Specify the axis to consider the perpendicular plane with either “*x*”, “*y*”, or “*z*”.

`[coordinate]` : Specify the coordinate of the plane perpendicular to the axis.
Ex.) To create a mirror image at the $y = 10.0$ plane.

```
% advmag2_mesh_mirror in.adv out.adv y 10.0
```

Local options: In addition to “`-dim`”, there are other local options.

- `-fgr` *file*

When creating a mirror image, element connectivity is also converted, so the element face numbers within the element also change. As a result, if left as is, there will be discrepancies with the FaceGroup data. By using this option, the FaceGroup data read from *file* will be converted to match the mirror image and output to a file. The file name will be the file name specified in `[output file]` with the extension changed to the same as *file*.

Ex.) `[output file]` is mirrored.adv, and *file* is original.fgr -> mirrored.fgr

A. 13. Tool for creating visualization file of elements: advmag2_mesh_separate_elem

This is a tool to create a visualization file to check how the mesh is divided into elements. It does not support HDDM-type analysis model files.

Here's how to execute advmag2_mesh_separate_elem:

```
% advmag2_mesh_separate_elem [input file] [output file] [ratio (0 - 1)] [options]
```

[input file] : Name of input file. Either MSH file (.msh), MSHX file (.mshx), or ADVENTURE file (.adv).
[output file] : Name of output file. Either VTU file (.vtu) or Legacy VTK file (.vtk).
[ratio (0 - 1)] : Enter a value greater than 0 and less than 1.
The smaller the value, the greater the spacing between elements.

Local options:

- -dim *n*

Since it is not possible to determine the dimension of a mesh from the MSH file, when reading a mesh that is not 3D from the MSH file, the dimension should be specified as *n* with this option.

A. 14. Tool for creating visualization files of domain decomposition: advmag2_mesh_separate_dd

This tool creates visualization files to check how the domains of HDDM-type analysis model files are decomposed. It creates visualization files for part decomposition and subdomain decomposition for each part. HDDM-type analysis model files are read from the analysis directory.

Here's how to execute advmag2_mesh_separate_dd:

```
% advmag2_mesh_separate_dd [output file] [ratio (0 - 1)] [options]
```

[output file] : Name of the output file. Either VTU file (.vtu) or Legacy VTK file (.vtk).
The name of the visualization file for the subdomain decomposition for each part will be the name specified here with the part number added.
Ex.) File for part 2: hoge.vtu -> hoge_part2.vtu
[ratio (0 - 1)] : Enter a value greater than 0 and less than 1.
The smaller the value, the greater the spacing between parts or subdomains.

This tool has no local options.

A. 15. Tools for mesh generation

These tools generate relatively simple and regular meshes.

The meshes generated by these tools are based on straight line, square, or cubic segments.

- In 1D, a straight line segment is divided into one line element.
- In 2D, a square segment is divided into two triangular elements
  or one quadrilateral element.
- In 3D, a cube segment is divided into six tetrahedral elements,
  one hexahedral element,
  two pentahedral (triangular prism) elements,
  or six pentahedral (square pyramid) elements.

By stacking these segments, the appearance becomes a straight line in 1D, a rectangle in 2D, and a rectangular prism in 3D. They can generate linear or quadratic elements for each.

The size of the entire mesh is determined by the number of segments and `[BaseDistance]`, which is the length of the segment sides.

The file name is fixed for each type of element.

Ex.) When generating tetrahedral elements using `advmag2_mesh_make_tetra`.

Mesh data: `tetra.mshx`   FaceGroup data: `tetra.fgrx`

These tools have common local options.

Common local options:

- `-quadratic`

  Quadratic elements are generated. If this option is not specified, linear elements are generated.

- `-fgr`

  FaceGroup data is generated and output.

A. 15. 1. Tool for generating mesh of line elements: advmag2_mesh_make_line

This tool generates linear or quadratic line elements.

Here's how to execute `advmag2_mesh_make_line`:

```
% advmag2_mesh_make_line [segments in x axis] [BaseDistance] [options]
```

`[segments in x axis]` : Give the number of segments in the *x* axis direction.

`[BaseDistance]` : Give the length of the side of the segment.

A. 15. 2. Tool for generating mesh (rectangular) of triangular elements: advmag2_mesh_make_tri

This tool generates linear or quadratic triangular elements.
Here's how to execute `advmag2_mesh_make_tri`:

```
% advmag2_mesh_make_tri [segments in x axis] [segments in y axis] [BaseDistance]
[options]
```

`[segments in x axis] [segments in y axis]` :
Give the number of segments in the x-axis and y-axis directions, respectively.

`[BaseDistance]` : Give the length of the side of the segment.

A. 15. 3. Tool for generating mesh (rectangular) of quadrilateral elements: advmag2_mesh_make_quad

This tool generates linear or quadratic quadrilateral elements.
Here's how to execute `advmag2_mesh_make_quad`:

```
% advmag2_mesh_make_quad [segments in x axis] [segments in y axis] [BaseDistance]
[options]
```

`[segments in x axis] [segments in y axis]` :
Give the number of segments in the x-axis and y-axis directions, respectively.

`[BaseDistance]` : Give the length of the side of the segment.

A. 15. 4. Tool for generating mesh (rectangular prism) of tetrahedral elements: **advmag2_mesh_make_tetra**

This tool generates linear or quadratic tetrahedral elements.
Here's how to execute advmag2_mesh_make_tetra:

% advmag2_mesh_make_tetra [segments in x axis] [segments in y axis] [segments in x axis] [BaseDistance] [*options*]

[segments in x axis] [segments in y axis] [segments in z axis] :
Give the number of segments in the *x*-axis, *y*-axis and *z*-axis directions, respectively.

[BaseDistance] : Give the length of the side of the segment.

A. 15. 5. Tool for generating mesh (rectangular prism) of hexahedron elements: **advmag2_mesh_make_hexa**

This tool generates linear or quadratic hexahedron elements.
Here's how to execute advmag2_mesh_make_hexa:

% advmag2_mesh_make_hexa [segments in x axis] [segments in y axis] [segments in x axis] [BaseDistance] [*options*]

[segments in x axis] [segments in y axis] [segments in z axis] :
Give the number of segments in the *x*-axis, *y*-axis and *z*-axis directions, respectively.

[BaseDistance] : Give the length of the side of the segment.

A. 15. 6. Tool for generating mesh (rectangular prism) of triangular prism elements: **advmag2_mesh_make_prism**

This tool generates linear or quadratic triangular prism elements.
Here's how to execute advmag2_mesh_make_prism:

% advmag2_mesh_make_prism [segments in x axis] [segments in y axis] [segments in x axis] [BaseDistance] [*options*]

[segments in x axis] [segments in y axis] [segments in z axis] :
Give the number of segments in the *x*-axis, *y*-axis and *z*-axis directions, respectively.

[BaseDistance] : Give the length of the side of the segment.

A. 15. 7. Tool for generating mesh (rectangular prism) of square pyramid elements: **advmag2_mesh_make_pyramid**

This tool generates linear or quadratic square pyramid elements.
Here's how to execute advmag2_mesh_make_pyramid:

% advmag2_mesh_make_pyramid [segments in x axis] [segments in y axis] [segments in x axis] [BaseDistance] [*options*]

[segments in x axis] [segments in y axis] [segments in z axis] :
Give the number of segments in the *x*-axis, *y*-axis and *z*-axis directions, respectively.

[BaseDistance] : Give the length of the side of the segment.

B. Format of Input / Output files

ADVENTURE_Magnetic uses the following files.

- FEA model file : ADVENTURE format
- HDDM-type analysis model file : ADVENTURE format
- Analysis result output settings file : ADVENTURE format
- Analysis result output file : ADVENTURE format
- Non-steady analysis result output file : ADVENTURE format
- Initial values of non-steady analysis, settings file : ADVENTURE format
- Initial values of non-steady analysis file : ADVENTURE format
- Restart settings file : ADVENTURE format
- Restart data file : ADVENTURE format
- Material data file : Text file
- Convergence history file : Text file
- Excitation current density data file : Text file
- Magnetization vector data file : Text file
- Shape definition file : Text file
- Characteristic curve file : Text file

There are other files in addition to those mentioned above, but when they appear in this manual, it indicates which module's manual the details are in, so please refer to that manual. Also, for the names of each file, please refer to Section 4.1.

Here, after describing the ADVENTURE format, the formats of the files mentioned above are described.

B. 1. ADVENTURE format

The ADVENTURE project modules use the ADVENTURE_IO library for inputting and outputting binary files. This library was developed for the purposes of handling large volumes of data, use in distributed parallel environments, improving input/output efficiency, and eliminating endian differences due to computer environments. Files input and output with this library are called ADVENTURE files, and their format is called the ADVENTURE format.

Regarding endian differences, they are recorded in little endian, regardless of whether each computer is little endian or big endian. When reading an ADVENTURE file recorded in little endian, the ADVENTURE_IO library compiled on each computer is used, and data is converted to the endian of each computer and stored in memory.

ADVENTURE files are input and output in data units called Documents. Each Document is composed of the following.

- Document ID : An ID to uniquely specify a Document.
- Property : An area that records Document attributes, the type and number of data stored in the Data area, etc.
- Data : An area that records coordinates, element connectivity, physical quantities, etc. as binary data.

"key"s that indicate the type of information and its value "val"s are recorded in the property area.

"key" that represents the attribute of a Document is "`content_type`", and "val"s include "`Element`" or "`Node`", which indicate that element connectivity and coordinates are stored in the Data area, and "`FEGenericAttribute`", which indicates that various data such as material IDs and physical quantities are stored in the Data area.

"`num_items`" is a "key" that indicates the amount of data stored in the Data area. If "`content_type=Element`", it is the number of elements, and if "`content_type=Node`", it is the number of nodes.

"`format`" and "`order`" are "key"s that indicate the type of data stored in the Data area. "`i`", "`f`", and "`c`" indicate integers, real numbers, and complex numbers, respectively, and the value following them is the number of bytes assigned to those numbers. For example, "`i2f8c4`" indicates that each piece of data consists of a 2-byte integer, an 8-byte real number (double-precision floating-point number), and a complex number with 4 bytes each for the real and imaginary parts (a total of 8 bytes). If the mesh is not decomposed, "`i2f8c4`" x "`num_items`" is the size of the data stored in the Data area. Note that "`c`" can only be used in "`order`", and "`c8`" in "`order`" is represented as two real numbers with the real and imaginary parts separated, like "`f8f8`" in "`format`".

"fega_type" is a "key" that indicates the attributes of the data in "content_type=FEGenericAttribute". These include "AllElementVariable" / "AllNodeVariable" which has data for all elements or nodes, "ElementVariable" / "NodeVariable" which has data for some elements or nodes, and "AllElementConstant" / "AllNodeConstant" which have the same value applied to all elements or nodes.

"label" is a "key" that indicates the name of the type of data stored in "content_type=FEGenericAttribute".

For more details, please refer to the sections on each file.

B. 2. FEA model file

This file is an ADVENTURE file created by advmag2_makefem_Electromagnetic. The main properties in the Property area of this file are shown below. Some of these will change depending on the options given when advmag2_makefem_Electromagnetic is executed.

```
#########################################################
  Element connectivity
#########################################################
[Properties]
  1: content_type=Element
  2: num_items=(Num. of elements)
  3: num_nodes_per_element=10
  4: dimension=3
  5: element_type=3DQuadraticTetrahedron
  6: format=i4i4i4i4i4i4i4i4i4i4

#########################################################
  Coordinates
#########################################################
[Properties]
  1: content_type=Node
  2: num_items=(Num. of nodes)
  3: dimension=3
  4: format=f8f8f8

#########################################################
  Material IDs
#########################################################
[Properties]
  1: content_type=FEGenericAttribute
  2: num_items=(Num. of elements)
  3: fega_type=AllElementVariable
  4: label=MaterialID
  5: format=i4
```

```
##################################################
  FaceGroup data
##################################################
[Properties]
  1: content_type=FEGenericAttribute
  2: num_items=(Num. of element faces)
  3: fega_type=ElementVariable
  4: label=FaceGroup
  5: format=i4
  6: order=i4
  7: ElementFace=1
  8: index_byte=4
  9: FaceGroup=(FaceGroup 番号)

##################################################
  Boundary conditions
##################################################
[Properties]
  1: content_type=FEGenericAttribute
  2: num_items=(Num. of boundary conditions)
  3: fega_type=ElementVariable
  4: label=DirichletBCs_Axn0_EF
  5: format=i4
  6: order=i4
  7: ElementFace=1
  8: index_byte=4

##################################################
  Settings, etc.
##################################################
[Properties]
  1: content_type=FEGenericAttribute
  2: num_items=0
  3: fega_type=AllElementConstant
  4: label=Options
  5: format=
  6: order=
  7: ADVMAG_NAME=ADVENTURE_Magnetic
  8: N_VERSION=(version of the module that created this file)
  9: DirichletBCs_Axn0=NO_NEED
  10: DirichletBCs_Axn0_EF=NEED
```

B. 3. HDDM-type analysis model file

This file is an ADVENTURE file created by ADVENTURE_Metis from an FEA model file. Mesh data with hierarchical domain decomposition, physical quantities decomposed accordingly, nodes shared on boundaries between domains, communication tables, etc. are recorded for each part.

Subdomain decomposition is performed within a part, and with some exceptions, data is written to the Data area on a subdomain basis. For this reason, the Property area has “key”s that were not present in the FEA model file. For example, "num_subdomains" indicates the number of subdomains in the part, "num_items_orig" was "num_items" in the FEA model file, and "sum_items" is the sum of the number of data for each subdomain. "num_items_orig" is the original number of elements or number of nodes before domain division. If "sum_items" is data about elements that are not shared between domains, it will be the total number of elements in the part. If it is data about nodes shared between domains, overlapping nodes on boundaries will be counted multiple times.

"content_type=Element" becomes "content_type=HDDM_Element", and the number of elements in the subdomain is recorded in the Data area first, followed by the element connectivity in the subdomain. This is repeated for the number of subdomains (Fig. 7). "content_type=FEGenericAttribute" becomes "content_type=HDDM_FEGenericAttribute" decomposed according to the mesh domain decomposition if "fega_type" is "AllElementVariable" / "AllNodeVariable", "ElementVariable" / "NodeVariable", "ElementConstant" / "NodeConstant", and data is written in the Data area on a subdomain basis, just like "content_type=HDDM_Element". However, if the data is related to nodes, the number of nodes in the subdomain is recorded first. If "fega_type" does not have "All", the number of elements or nodes to which information is given is recorded. Also, "fega_type" of "AllElementConstant" / "AllNodeConstant" and "Void" are not decomposed.

```
[Properties]
 1: content_type=HDDM_Element
 2: num_subdomains=65
 3: element_type=3DQuadraticTetrahedron
 4: num_nodes_per_element=10
 5: dimension=3
 6: sum_items=6545
 7: num_items_orig=13075

[Data]
# SubDomain[0]
# num items in SubDomain
  101
 0: 162 158 168 189 156 157 163 159 169 161
 1: 168 158 213 189 159 170 169 160 188 161
        :
 99: 175 184 189 199 177 173 178 186 192 187
 100: 194 189 184 199 191 185 193 186 187 192
# SubDomain[1]
# num items in SubDomain
  100
 0: 171 192 166 186 173 167 174 168 169 188
 1: 181 192 171 186 183 172 175 173 174 188
        :
```

Fig. 7. "content_type=HDDM_Element".

Examples of nodes and communication tables shared on boundaries between domains include "label=ElementIndex_SubdomainToGlobal", which is an index from element numbers in the subdomain to element numbers in the entire mesh, "label=NodeIndex_SubdomainToPart", which is an index from node numbers in the subdomain to node numbers in the part, "label=NodeIndex_PartToGlobal", which is an index from node numbers in the part to node numbers in the entire mesh, "HDDM_InterfaceDOF", which are communication tables between parts, and "InterfaceDOF", which is information about nodes shared between domains among the nodes in the subdomain. Unique "key"s include "num_parts", which indicates the number of parts, and "part_number", which indicates the part number.

For more detailed information about this file, please refer to the ADVENTURE_Metis manual.

B. 4. Analysis result output settings file

This file is an ADVENTURE file created by the solver module. This file records the analysis conditions when the analysis result output files or non-steady analysis result output files in the same directory were obtained.

Fig. 8 shows the analysis result output settings file output when a non-steady eddy current analysis was performed (see Appendix C.3). "`Analysis`" indicates the type of analysis, and "`Formulation`" indicates the type of equation, which are "key"s that indicate what kind of problem was solved. "key"s beginning with "`NS`" indicate the settings for non-steady analysis, and "key"s beginning with "`NL`" indicate the settings for non-linear analysis. "key"s beginning with "`HDDM`" indicate the settings for the parallel sparse matrix solver, and "key"s beginning with "`Solver`" indicate the settings for the non-parallelized sparse matrix solver for the subdomain. The numbers following these, the first one indicates the type of analysis ("2" is a non-steady eddy current analysis). The second one indicates that "0" is the setting for the electromagnetic field analysis, and "1" is the setting for the analysis to compute the correction amount of the current density. Real values are recorded in the Data area. The "key"s that indicate the type of these values are "`Data`" with a number next to it. "`Data0=NS_delta_t`" indicates that the 0th value in the Data area, "`0: 8.333333e-04`", is the time step size for the non-steady analysis, and "`Data1=HDDM_conv2_0`" indicates that the 1st value in the Data area, "`1: 1.000000e-03`", is the convergence criterion for the parallel sparse matrix solver used in the electromagnetic field analysis.

B. 5. Analysis result output file

This file is an ADVENTURE file created by the solver module. It records the analysis results of steady or quasi-steady analysis. In addition to unknowns, the material IDs, the excitation current density of the coils and the magnetization vectors of the permanent magnets which are the electromagnetic field sources, physical quantities computed from the unknowns, and the boundary conditions used in the analysis depending on the command options are recorded.

B. 6. Non-steady analysis result output file

This file is an ADVENTURE file created by the solver module. It is an analysis result output file created for each time step when a non-steady analysis is performed.

B. 7. Initial values of non-steady analysis, settings file / Initial values of non-steady analysis file

These files are used as the initial values for a non-steady analysis. The analysis result output settings file and analysis result output files for steady or quasi-steady analysis are copied and used along with the subdirectories in which they are stored.

B. 8. Restart settings file

This is an ADVENTURE file created by the solver module. It records the settings of the parallel sparse matrix solver given the restart data file in the same directory.

B. 9. Restart data file

This is an ADVENTURE file created by the solver module. It records the scalars and vectors needed to restart the parallel sparse matrix solver.

```
[Properties]
 1: content_type=FEGenericAttribute
 2: num_items=11
 3: fega_type=AllElementConstant
 4: label=Options
 5: format=f8
 6: order=f8
 7: NonSteady=Yes
 8: DomainDecomposition=Yes
 9: Settings=Yes
 10: FloatPrecision=2
 11: Analysis=NS_Eddy
 12: Formulation=APhi
 13: NS_end_step=20
 14: NS_out_interval=1
 15: Data0=NS_delta_t
 16: NS_inivalue_type=real
 17: NL_method=None
 18: HDDM2_0_for=EM
 19: HDDM2_0=CG
 20: HDDM_mat2_0=implicit
 21: HDDM_pc2_0=Diag
 22: Data1=HDDM_conv2_0
 23: Data2=HDDM_div2_0
 24: HDDM_max_loop2_0=-1
 25: Solver2_0_for=EM
 26: Solver2_0=CG
 27: Solver_mat2_0=AIJ
 28: Solver_pc2_0=ICC
 29: Data3=Solver_pc_icc_param2_0
 30: Data4=Solver_conv2_0
 31: Data5=Solver_div2_0
 32: Solver_max_loop2_0=-1
 33: HDDM2_1_for=Jo
 34: HDDM2_1=CG
 35: HDDM_mat2_1=implicit
 36: HDDM_pc2_1=Diag
 37: Data6=HDDM_conv2_1
 38: Data7=HDDM_div2_1
 39: HDDM_max_loop2_1=-1
 40: Solver2_1_for=Jo
 41: Solver2_1=LDL
 42: Solver_mat2_1=AIJ
 43: Solver_pc2_1=ICC
 44: Data8=Solver_pc_icc_param2_1
 45: Data9=Solver_conv2_1
 46: Data10=Solver_div2_1
 47: Solver_max_loop2_1=-1

[Data]
 0: 8.333333e-04
 1: 1.000000e-03
 2: 1.000000e+10
 3: 1.200000e+00
 4: 1.000000e-09
 5: 1.000000e+10
 6: 1.000000e-10
 7: 1.000000e+10
 8: 1.200000e+00
 9: 1.000000e-12
 10: 1.000000e+10
```

Fig. 8. “`sample_data/ns_eddy/done/result/advhddm_out.adv`”.

B. 10. Material data file

This file is a text file created by the user. It is used to set physical quantities for each material ID. This file combines a Basic Configuration for setting general physical quantities, a Global Common Configuration for setting physical quantities common to the entire analysis domain, an Electromagnetic Field Source Configuration for setting electromagnetic field sources, and a Non-linear Configuration for setting up non-linear analysis.

B. 10. 1. Basic Configuration

The Basic Configuration begins with a keyword (*keyword*) that represents a physical quantity, followed by the number of material IDs (*num*) to be written here. After that, the material ID number and physical quantity are written as a set for the number of material IDs (*num*) (Fig. 9).

- The material IDs do not have to be listed in order starting from 0.
- It is okay if not all material IDs in the mesh are listed,
  but material IDs not listed here will be given a physical quantity of 0.
- If there are duplicated material IDs, the one written later will take priority.

List this configuration as many types of physical quantities as are required.

```
keyword num
 0 7.95e+05
 5 1.2
 2 0.5
   :
```

Fig. 9. Basic Configuration of material data file.

The physical quantities given to the electromagnetic field analysis module in the Basic Configuration are as follows. The string in parentheses after the physical quantity name is a keyword (*keyword*).

- Magnetic reluctivity (`MagneticReluctivity`) [m/H]
  - Required for (non-linear) magnetostatic analysis, time-harmonic eddy current analysis, non-linear eddy current analysis, high-frequency electromagnetic field analysis.
  - It can also be given as magnetic permeability (`MagneticPermeability`) [H/m] or relative permeablity (`RelativePermeability`) [-].
    - If multiple values are given to the same material ID, the order of priority is magnetic reluctivity, magnetic permeability, and relative permeability.

- Electrical conductivity (`ElectricalConductivity`) [S/m]
  - The keyword for electrical conductivity can also be “`Conductor`”, which was used up to ADVENTURE_Magnetic Ver.1.9.2.
  - Required for time-harmonic eddy current analysis, non-linear eddy current analysis, high-frequency electromagnetic field analysis.
  - The region with a material ID where electrical conductivity is given is treated as a conductor, and the region with a material ID where electrical conductivity is not given is treated as a non-conductor.

- Permittivity (`Permittivity`)
  - Required for high-frequency electromagnetic field analysis. Required, when considering the displacement current term in time-harmonic eddy current analysis, non-linear eddy current analysis.
  - It can also be given as relative permittivity (`RelativePermittivity`).
    - If multiple values are given to the same material ID, the order of priority is permittivity and relative permittivity.

B. 10. 2. Global Common Configuration

The Global Common Configuration is used when setting physical quantities common to the entire analysis domain, such as the angular frequency in a quasi-steady problem (Fig. 10). This configuration consists only of keyword (*keyword*) that represent physical quantity, and a value (*value*).

```
keyword
 value
```

Fig. 10. Global Common Configuration of material data file.

The physical quantities given to the electromagnetic field analysis module in the Global Common Configuration are as follows. The string in parentheses after the physical quantity name is a keyword (*keyword*).

- Angular frequency (`AngularFrequency`) [rad/s]
  - The keyword for angular frequency can also be "`CoilOmega`", which was used up to ADVENTURE_Magnetic Ver.1.9.2.
  - Required for time-harmonic eddy current analysis, high-frequency electromagnetic field analysis.
  - It can also be given as period (`Period`) [s] or frequency (`Frequency`) [Hz].
    - If multiple values are given to the same material ID, the order of priority is angular frequency, period and frequency.

B. 10. 3. **Electromagnetic Field Source Configuration**

In the Electromagnetic Field Source Configuration, settings related to the electromagnetic field source, such as the excitation current density flowing through the coil and the magnetization vector of the permanent magnet, are made. This configuration begins with a keyword (*`keyword`*) that represents the physical quantity, followed by the number of material IDs (*`num`*) to be written here. After that, the material ID number (*`id`*), the electromagnetic field source setting mode (*`mode`*), and the file (*`file`*) containing the electromagnetic field source setting information are written as a set for the number of material IDs (*`num`*) (Fig. 11). The number of material IDs (*`num`*) is the number of regions that are coils or permanent magnets. In addition, the file (*`file`*) containing the electromagnetic field source setting information should be written using a relative path from the subdirectory where the material data file is located (the directory specified by the option "`-mtrldat-dir`").

```
keyword num
 id mode file
   :
```

Fig. 11. Electromagnetic Field Source Configuration of material data file.

The electromagnetic field sources given to the electromagnetic field analysis module in the Electromagnetic Field Source Configuration are as follows. The string in parentheses after the electromagnetic field source name is a keyword (*`keyword`*).

- Coil (`Coil`)
- Permanent magnet (`PermanentMagnet`)

At least one electromagnetic field source is required in the electromagnetic field analysis module. Set at least one of the material IDs to coil or permanent magnet.

The electromagnetic field source setting mode (*`mode`*) can be selected from "`rf`" (*Read from File*) and "`md`" (*Make from Definition*).

If "`rf`" is selected, the file (*`file`*) that describes the subsequent electromagnetic field source setting information will be excitation current density data files for coils, or magnetization vector data files for permanent magnets. The values of excitation current density or magnetization vector are directly read from these files. For (non-linear) magnetostatic analysis, the file name will have the string specified in "*`file`*" followed by "`s`". For time-harmonic eddy current analysis and high-frequency electromagnetic filed analysis, which are quasi-steady problems, real and imaginary part files are required, and the string will have "`r`" and "`I`" respectively. This setting mode cannot be selected for non-steady eddy current analysis.

If "`md`" is selected, the file (*`file`*) that describes the subsequent electromagnetic field source setting information will be shape definition files. The excitation current density or magnetization vector is computed based on the loaded definitions and used in the analysis. The file name is used as written here. This setting mode must be selected for non-steady eddy current analysis.

B. 10. 4. Non-linear Configuration

In the Non-linear Configuration, the material IDs (`id`) of the regions where nonlinearity is considered and the name of the characteristic curve file (`file`) are given. This configuration begins with the keyword "`NonLinear`", followed by the number of material IDs (`num`) to be written here. After that, the material ID (id) of the region where nonlinearity is considered and the name of the characteristic curve file (`file`) are written as a set for the number of material IDs (`num`) (Fig. 12). The number of material IDs (`num`) is the number of regions where nonlinearity is considered. The name of the characteristic curve file (`file`) should be written as a relative path from the subdirectory where the material data file is located (the directory specified by the option "`-mtrldat-dir`").

```
NonLinear num
 id file
   :
```

Fig. 12. Non-linear Configuration of material data file.

B. 10. 5. Multiple identical keywords, comments

When ADVENTURE_Magnetic reads a material data file, it searches for the keyword from the beginning of the file, and when it finds the keyword, it judges that keyword to be the beginning of the configuration and reads values.

If there are multiple identical keywords in the file, ADVENTURE_Magnetic will read all the configurations, but only the first configuration will be used for analysis. With this in mind, you can intentionally place multiple keywords (by changing the order by copying and pasting each time you analyze to analyze under various conditions), but be careful not to make a mistake.

Comments can also be inserted, but they cannot be inserted in the middle of a configuration. Insert them at the beginning or end of the file, or between configurations. There is no rule that requires a symbol to be placed at the beginning of a line. However, please avoid using the same character string as a keyword. If there is one or more characters without being separated by a half-width space or line break, it will be judged to be a different string. For example, if a "`#`" is added like a hashtag, such as "`#NonLinear`", it will be considered a different string. This enables you to place multiple identical keyword configurations and then comment out keywords other than those are used for analysis by putting a "`#`" in front of them.

B. 11. Excitation current density data file / Magnetization vector data file

This file is a text file created by the user. The excitation current density data file is used to give the excitation current density [A/m^2], and the magnetization vector data file is used to give the magnetization vector [T] to the nodes. These files cannot be used in non-steady eddy current analysis.

In this file, first write the number of nodes (`num`) to which the excitation current density or magnetization vector is to be given. Then write the node number (`id`) and the *x*, *y*, and *z* components (`x y z`) of the excitation current density or magnetization vector as a set for the number of nodes (`num`) (Fig. 13).

```
num
 id x y z
   :
```

Fig. 13. Excitation current density data file / Magnetization vector data file.

B. 12. Shape definition file

This file is a text file created by the user. The excitation current density [A/m^2] or magnetization vector [T] is computed based on the loaded definitions and used in the analysis. This file includes the "Shape Definition"s and "Time Change Definition"s. In steady and quasi-steady analyses, only the shape is defined by Shape Definition, while in non-steady analyses the excitation current density computed by the Shape Definition is changed over time by the Time Change Definition. Note that since the magnetization vector is used by permanent magnets, it cannot be changed by the Time Change Definition.

The Shape Definition is written first in the file, followed by the Time Change Definition if necessary. It will not work correctly if the order of the Shape Definition and Time Change Definition are swapped or mixed.

There is a tool for computing source vector, `advmag2_SourceVector`, as a tool to check whether the shape definition file has been created correctly. See Appendix A.5 for details.

B. 12. 1. Shape Definition

There are two types of Shape Definitions:

- Double sectorial cylinder (all or part of a cylinder)
- Parallelepiped (rectangular prism, cube, etc.)

Multiple of these definitions can be combined. In addition, whether or not to give the excitation current density or magnetization vector is controlled by the material ID. Therefore, they will not be given to nodes outside the region, so set the zone a little larger so that nodes at the end face of the region are not judged to be outside the region due to numerical error. As a result, at nodes where the zones of multiple definitions overlap, the one defined earlier in the file will take precedence.

B. 12. 1. 1. **Shape Definition**

The Shape Definition of a double sectorial cylinder sets a part or the whole of the cylinder as the zone. The structure of this definition is shown in Fig. 14. Some of the components are shown in Fig. 15 and Fig. 16.

```
DoulbeSectorialCylinder
 x y z
 dir height
 unit θ1 θ2
 radius_in radius_out
 magnitude
```

Fig. 14. Shape Definition: double sectorial cylinder.

The first line is the keyword for the Shape Definition of a double sectorial cylinder. The second line specifies the coordinates [m] of the cardinal point on the central axis of the cylinder in 3D (*x y z*). The plane on which this point exists becomes the bottom surface of the cylinder. The third line specifies the direction of the central axis (*dir*) and the height of the cylinder (*height*) [m]. Specify the direction as "*x*", "*y*", or "*z*" (Fig. 15). The fourth line specifies two angles shown in Fig. 16 to determine the range of the sector. First, specify the unit of the angle in "rad" if using radian, or "deg" if using degree. Both angles are specified in the same unit. Note that which axis the angle from becomes θ_1 is determined by the direction of the central axis (*dir*) specified in the third line.

- If "*x*" is selected as the direction of the central axis (*dir*),
  the bottom surface is in the *yz* plane, and θ_1 is the angle from the *y* axis.
- If "*y*" is selected as the direction of the central axis (*dir*),
  the bottom surface is in the *zx* plane, and θ_1 is the angle from the *z* axis.
- If "*z*" is selected as the direction of the central axis (*dir*),
  the bottom surface is in the *xy* plane, and θ_1 is the angle from the *x* axis (Fig. 16).

In the fifth line, write the inner radius (*radius_in*) [m] and the outer radius (*radius_out*) [m] of the double sectorial cylinder. In the sixth line, write the magnitude of the excitation current density [A/m^2], or the magnitude of the magnetization vector [T]. <u>The magnitude has a direction</u>. When it rotates in a right-handed screw direction with respect to the central axis (*dir*) (Fig. 15), it is positive, and when it rotates in the opposite direction, it is negative. For time-harmonic eddy current analysis and high-frequency electromagnetic field analysis, which are quasi-steady problems, complex numbers must be entered. Therefore, enter two values, the real part and the imaginary part. The imaginary unit is not necessary. The real part should be the value at the antinode of the sinusoidal wave, and the imaginary part should be the value at the node 90° away from there, or the real part should be the value at the node of the sinusoidal wave, and the imaginary part should be the value at the antinode 90° away from there. In the case of a three-phase current, decide on one of the phases as described above, and enter values that are out of phase with respect to that for the remaining two phases.

Ex.) If the real part of the first phase is the value at the antinode (90°) and the imaginary part is the value at the node (180°), the second phase will have a real part at -30° and an imaginary part at 60°, and the third phase will have a real part at 210° and an imaginary part at 300°.

For a specific example of the Shape Definition of a double sectorial cylinder, see Section B.12.1.3, Appendix C.1 or C.2.

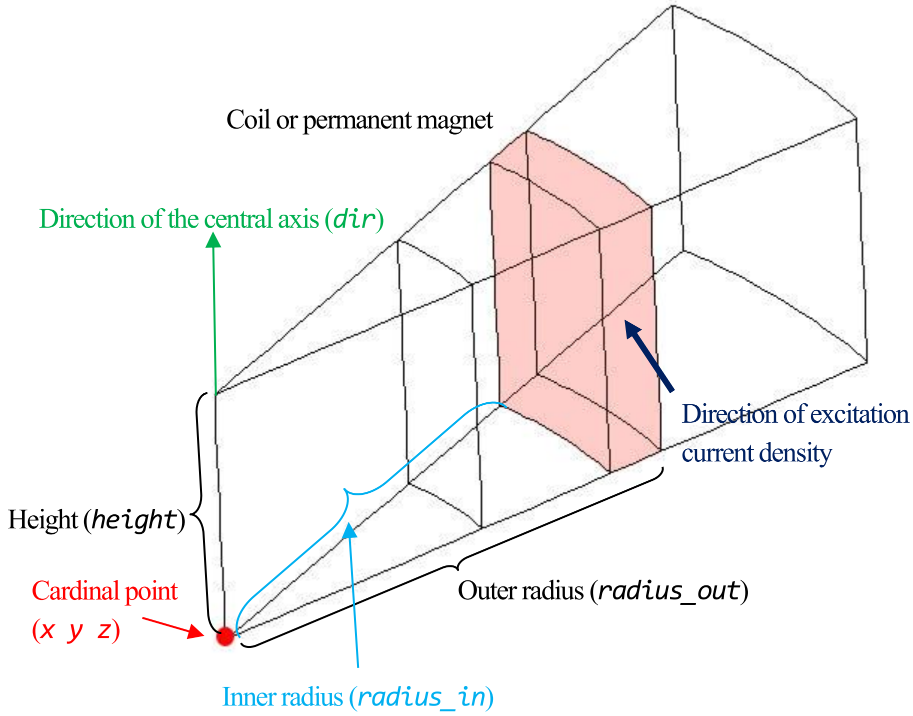


Fig. 15. Double sectorial cylinder: names of each part.

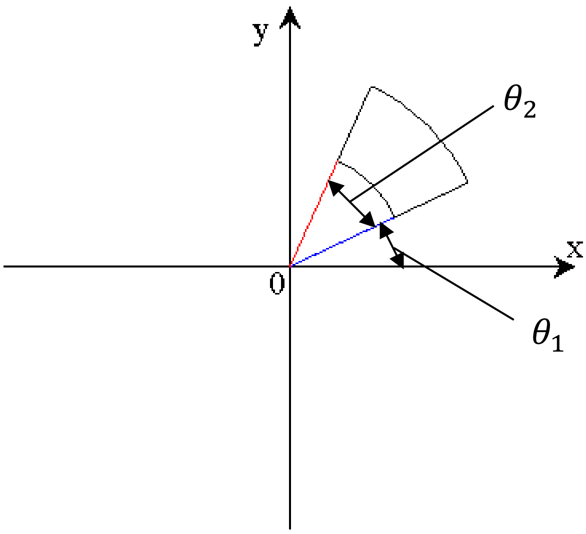


Fig. 16. Double sectorial cylinder: $\boldsymbol{\theta_1}$ and $\boldsymbol{\theta_2}$ (When the z-axis is selected as the direction of the central axis).

B. 12. 1. 2. **Parallelepiped**

The Shape Definition of a parallelepiped defines the space enclosed by three pairs of opposing parallel faces as the zone. A rectangular prism and a cube are types of parallelepipeds. The structure of this definition is shown in Fig. 17. Some of the components of each are shown in Fig. 18.

The first line is the keyword for the Shape Definition using a parallelepiped. On the second line, select one of the vertices of the parallelepiped as the cardinal point and write its coordinates (x_0 y_0 z_0). It does not matter which vertex you select. On the third to fifth lines, write the coordinates (x_1 y_1 z_1, x_2 y_2 z_2, x_3 y_3 z_3) of the three vertices adjacent to the cardinal point (blue circles in Fig. 18). As long as they are adjacent, the order is arbitrary. Two of these and the cardinal point identify the three faces of the parallelogram to which the cardinal point belongs and their range. The remaining three faces opposite them are determined automatically. On the sixth line, enter the excitation current density [A/m^2] or magnetization vector [T] as a 3D vector (v_x v_y v_z). Since complex numbers must be used in time-harmonic eddy current analysis and high-frequency electromagnetic field analysis, which are quasi-steady problems, enter two 3D vectors with real and imaginary parts (v_{xr} v_{yr} v_{zr} v_{xi} v_{yi} v_{zi}). The imaginary unit is not necessary. The part of the sinusoidal wave to which the real and imaginary parts should be assigned is the same as in the Shape Definition of a double sectorial cylinder.

For a specific example of the Shape Definition of a parallelepiped, see Section B.12.1.3.

```
Parallelepiped
 x0 y0 z0
 x1 y1 z1
 x2 y2 z2
 x3 y3 z3
 vx vy vz (or vxr vyr vzr vxi vyi vzi)
```

Fig. 17. Shape Definition: parallelepiped.

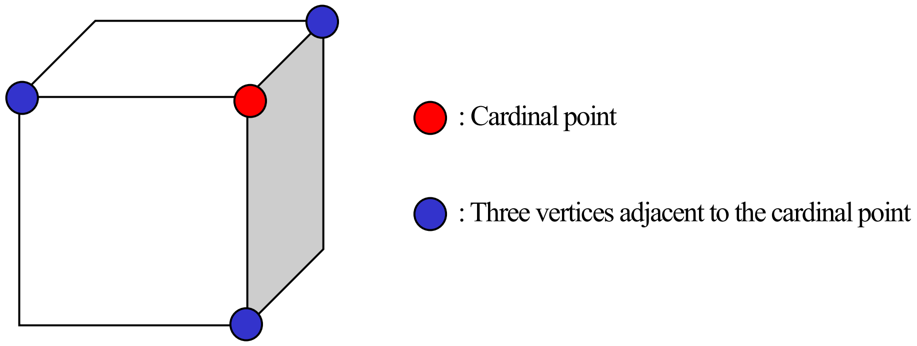


Fig. 18. Parallelepiped: cardinal point and three vertices.

B. 12. 1. 3. Specific Example of Shape Definition

Here, a specific example of Shape Definition is shown based on TEAM Workshop Problem 7 (Fig. 19, TEAM7) [13]. TEAM7 is a model for verifying eddy current analyses, and a conductor plate with holes asymmetric with respect to the coil is placed.

Fig. 20 shows the shape definition file for giving the excitation current density to the coil of this model. In Fig. 19, the unit of length is mm, but in the shape definition file, it is assumed that the mesh is made in m. The current flows counterclockwise when viewed from the *z*-axis direction as shown in Fig. 19. The magnitude of the current density is 1.0986e+06 [A/m^2] for the real part and 0 [A/m^2] for the imaginary part.

When viewed from the z-axis direction, the straight parts are defined by the Shape Definition of a parallelepiped, and the R parts are defined by the Shape Definition of a double sectorial cylinder. The values related to the dimensions are set to be approximately 0.005 m larger. The angle of R is also set to be 10° larger at both ends. This is to prevent the nodes on the end faces of the region from being judged to be outside the region due to numerical errors, as mentioned above. However, the dimensions are correct for the surface where the parallelepiped is in contact with R. This is based on the idea that the excitation current density given by the parallelepiped within R is clearly different from the actual one, and by making the double sectorial cylinder in the R part a little larger, the nodes on the boundary are not missed, but in the straight line parts beyond the boundary, the definition of the parallelepiped is written first so that it takes priority. Note that the excitation current density is given as a magnitude in the double sectorial cylinder, so it is always "`1.0986e+06 0.0`", but in the parallelepiped, it is given as vectors so it differs depending on the location.

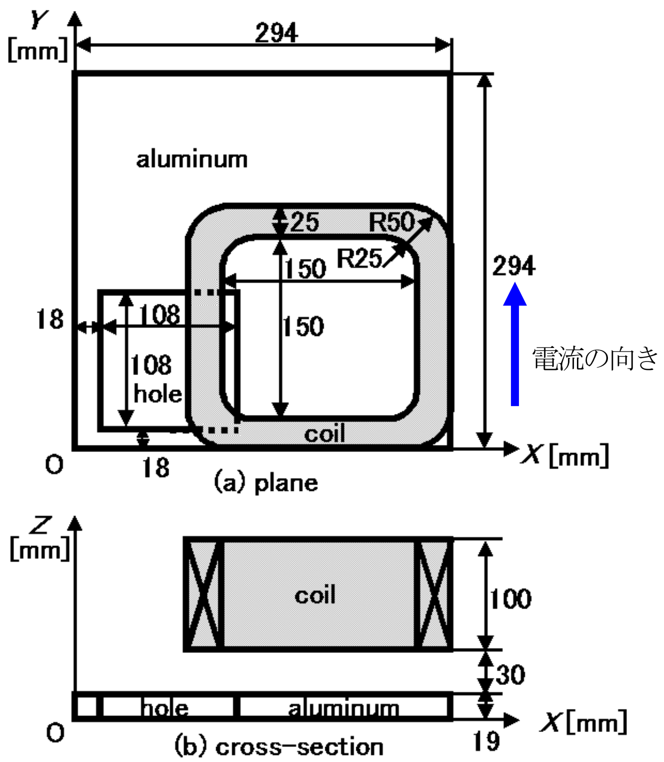


Fig. 19. TEAM Workshop Problem 7.

```
Parallelepiped
 0.089  0.050  0.044
 0.089  0.050  0.154
 0.089  0.150  0.044
 0.124  0.050  0.044
 0.0  -1.0986e+06  0.0   0.0  0.0  0.0

Parallelepiped
 0.144  -0.005  0.044
 0.144  -0.005  0.154
 0.144   0.030  0.044
 0.244  -0.005  0.044
 1.0986e+06  0.0  0.0   0.0  0.0  0.0

Parallelepiped
 0.264  0.050  0.044
 0.264  0.050  0.154
 0.264  0.150  0.044
 0.299  0.050  0.044
 0.0  1.0986e+06  0.0   0.0  0.0  0.0

Parallelepiped
 0.144  0.170  0.044
 0.144  0.170  0.154
 0.144  0.205  0.044
 0.244  0.170  0.044
 -1.0986e+06  0.0  0.0   0.0  0.0  0.0
```

```
DoubleSectorialCylinder
 0.144  0.050  0.044
 z  0.110
 deg   170.0  110.0
 0.020  0.055
 1.0986e+06  0.0

DoubleSectorialCylinder
 0.244  0.050  0.044
 z  0.110
 deg  -100.0  110.0
 0.020  0.055
 1.0986e+06  0.0

DoubleSectorialCylinder
 0.244  0.150  0.044
 z  0.110
 deg   -10.0  110.0
 0.020  0.055
 1.0986e+06  0.0

DoubleSectorialCylinder
 0.144  0.150  0.044
 z  0.110
 deg    80.0  110.0
 0.020  0.055
 1.0986e+06  0.0
```

Fig. 20. Shape definition file of TEAM Workshop Problem 7.

B. 12. 2. Time Change Definition

The Time Change Definition is used in non-steady analyses to change the excitation current density flowing through the coil. The Time Change Definition specifies the magnification of the excitation current density specified in the Shape Definition. The Time Change Definition is performed for each range of the set time using the following two:

- Sinusoidal wave
- Straight line

Multiple of these definitions can be combined. When multiple definitions are made, they will be superimposed. Waves can be expressed as the sum of sinusoidal waves using a Fourier series, so it is possible to define the time changes of various currents (express the cosine term of the Fourier series by shifting the phase of the sinusoidal wave ($\cos\theta = \sin(\theta + \pi/2)$)).

The same change will be made within coils with the same material ID. If different changes are to be made, separate material IDs are required. This requires that the regions be separated from the CAD data creation stage.

B. 12. 2. 1. Range of the set time

The range of the set time starts with the keyword “`TimeEvolution`” and continues until the next “`TimeEvolution`” or the end of the file. After each “`TimeEvolution`”, write the end of the set time in seconds (Fig. 21). If there are multiple “`TimeEvolution`”, they will be sorted in order of the earliest end time, so the order of the actual time can be swapped. In the case of Fig. 21, the Time Change Definition “*`Definition 1`*” of “`TimeEvolution 1.0`”, which has the earliest end time, is applied to time t in the range of $0.0 \leq t < 1.0$. The next earliest is “`TimeEvlution 2.0`”, so its Time Change Definition “*`Definition 3`*” is applied in the range of $1.0 \leq t < 2.0$. The Time Change Definition “*`Definition 2`*” of “`TimeEvolution 3.0`”, which is the slowest, is basically applied in the range of $2.0 \leq t < 3.0$, but if the analysis continues after 3.0 seconds from the start of the analysis, “*`Definition 2`*” will continue to be applied.

```
TimeEvolution 1.0
  Definition 1
TimeEvolution 3.0
  Definition 2
TimeEvolution 2.0
  Definition 3
```

Fig. 21. Range of the set time.

B. 12. 2. 2. **Sinusoidal wave**

The Time Change definition of a sinusoidal wave is expressed as follows:

$$a\sin(\omega t+\varphi)+C. \tag{4}$$

Here, a is the magnification, ω is the angular frequency, φ is the initial phase, and C is a constant, which are written in the shape definition file. t is the time at each time step. The structure of this definition is as shown in Fig. 22.

The first line is the keyword for defining the Time Change Definition due to a sinusoidal wave. On the second line, give the unit of angular frequency (*unit1*) and the angular frequency (ω). If the unit of angular frequency is [rad/s], write "rad", and if it is [deg/s], write "deg". It is also possible to give the frequency instead of the angular frequency. In that case, write "Hz" in "*unit1*" and the frequency in place of "ω". On the third line, write the unit of the initial phase (*unit2*) and the initial phase (φ). If the unit of the initial phase is in radians, write "rad", and if it is in degrees, write "deg". On the fourth line, write the magnification (a) and the constant (C).

For specific examples of the Time Change Definition of a sinusoidal wave, see Section B.12.2.4 or Appendix C.3.

```
TimeEvolutionSinusoidal
  unit1 ω
  unit2 φ
  a C
```

Fig. 22. Time Change Definition: sinusoidal wave.

B. 12. 2. 3. **Straight line**

The Time Change definition of a straight line is expressed as follows:

$$(\beta - \alpha)\frac{t - t_1}{t_2 - t_1} + \alpha + C. \tag{5}$$

Here, t_1 and t_2 are the start and end times of the set time range, α and β are the magnifications at the start and end times, respectively, and C is a constant; these are written in the shape definition file. Also, t is the time at each time step. In this definition, the magnification changes linearly from $\alpha + C$ at the start time t_1 of the set time range to $\beta + C$ at the end time t_2 (Fig. 23). The structure of this definition is shown in Fig. 24.

The first line is the keyword for defining the Time Change Definition due to a straight line. The second line describes the magnification (α) at the start time and the magnification (β) at the end time. The third line describes the constant (C).

For a specific example of the Time Change Definition of a straight line, see Section B.12.2.4.

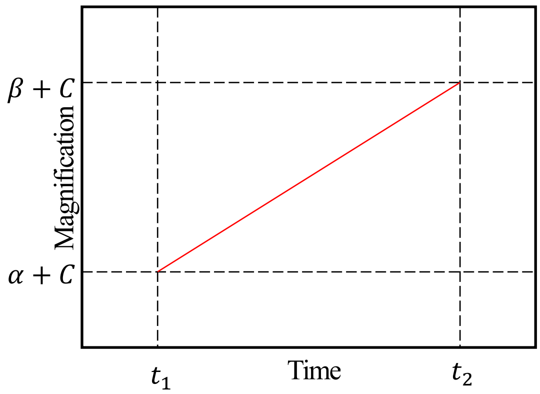


Fig. 23. Time Change Definition: magnification of the straight line.

```
TimeEvolutionSinusoidal
  α β
  C
```

Fig. 24. Time Change Definition: straight line.

B. 12. 2. 4. Specific example of Time Change Definition

As specific examples of Time Change Definition, the change from a sinusoidal wave to a straight line (Fig. 25), the superposition of sinusoidal waves (Fig. 26), and the superposition of a sinusoidal wave and a straight line (Fig. 27) are shown. In each figure, the Time Change Definition part of the shape definition file is written on the left. On the right is a graph when the magnitude of the excitation current density is given as 50 [A/m^2] in the shape definition. The horizontal axis is time [s] and the vertical axis is the magnitude of the excitation current density [A/m^2].

```
TimeEvolution 1.0
TimeEvolutionSinusoidal
  Hz 1
  deg 90.0
  1.0 0.0

TimeEvolution 2.0
TimeEvolutionLinear
  1.0 -1.0
  0.0
```

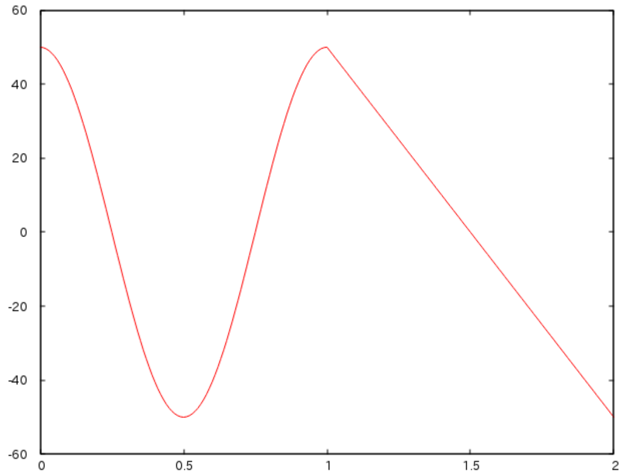


Fig. 25. Change from a sinusoidal wave to a straight line.

```
TimeEvolution 1.0
TimeEvolutionSinusoidal
  Hz 1
  deg 90.0
  1.0 0.0
TimeEvolutionSinusoidal
  Hz 2
  deg 0.0
  1.0 0.0
```

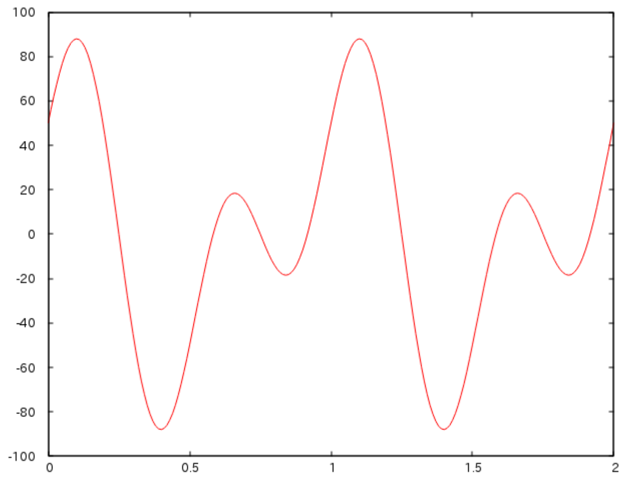


Fig. 26. Superposition of sinusoidal waves.

```
TimeEvolution 1.0
TimeEvolutionSinusoidal
  rad 31.4159
  rad 0.0
  1.0 0.0
TimeEvolutionLinear
  -1.0 1.0
  0.0
```

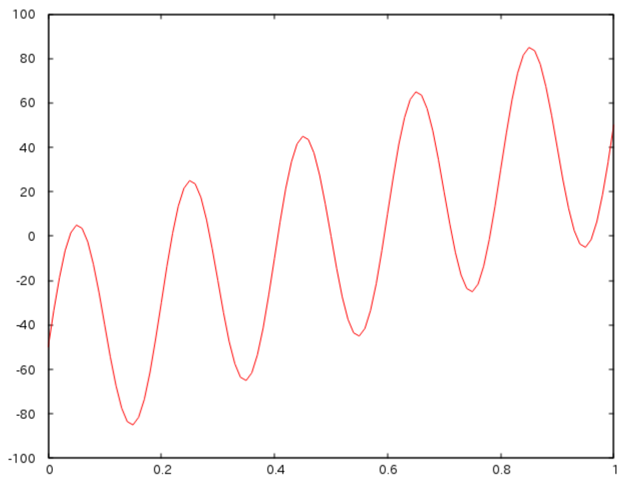


Fig. 27. Superposition of a sinusoidal wave and a straight line.

B. 12. 3. Multiple identical keywords, comments

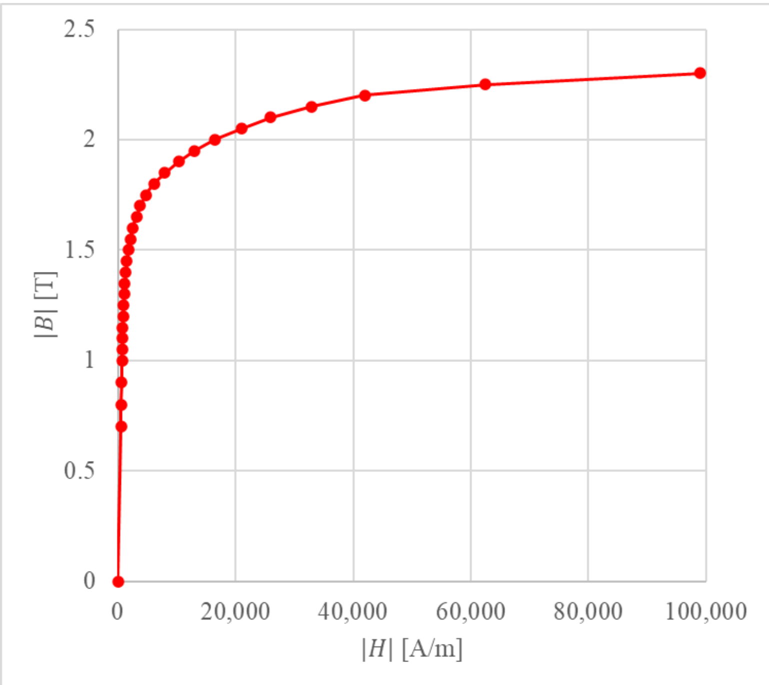
As described above, multiple Shape Definitions and Time Change Definitions can be used in combination. Even if there are multiple identical keywords, there is no distinction between those that are used in the analysis and those that are not, as in the material data file. All will be used.

Comments can be inserted, but they cannot be inserted in the middle of the definition structure. Insert them at the beginning or end of the file, or between definitions. There is no rule that requires a symbol to be placed at the beginning of a line. However, please avoid using the same character string as a keyword. If there is one or more characters without being separated by a half-width space or line break, it will be considered a different character string. For example, if a "`#`" is added like a hashtag, such as "`#TimeEvolution`", it will be considered a different character string. Therefore, it is possible to place multiple identical keyword definitions and comment out keywords other than those used in the analysis by adding a "`#`" to the beginning of them. However, please note that if "`TimeEvolution`" is commented out, the sinusoidal wave and straight line keywords for that range of the set time will be valid for the range of the set time of the previous "`TimeEvolution`" unless they are also commented out.

B. 13. Characteristic curve file

This file is a text file created by the user. It is used to give non-linear characteristics of physical quantities in non-linear analysis.

In the electromagnetic field analysis module, a *B-H* characteristic curve is given to take into account the non-linear characteristics of magnetic reluctivity [m/H]. When a *B-H* characteristic curve is graphed, the horizontal axis is the magnitude of the magnetic field (|*H*| [A/m]) and the vertical axis is the magnitude of the magnetic flux density (|*B*| [T]). The curve is represented by a cloud of points, and the number of points is recorded first in the file, and then the horizontal and vertical axis values of the points are recorded in sets in order, starting from the point closest to the origin (Fig. 28).




```
31
0.00e+0 0.00
5.30e+2 0.70
5.70e+2 0.80
6.30e+2 0.90
7.00e+2 1.00
7.40e+2 1.05
7.90e+2 1.10
8.40e+2 1.15
9.00e+2 1.20
9.70e+2 1.25
1.05e+3 1.30
1.17e+3 1.35
1.32e+3 1.40
1.52e+3 1.45
1.78e+3 1.50
2.09e+3 1.55
2.50e+3 1.60
3.15e+3 1.65
3.80e+3 1.70
4.76e+3 1.75
6.10e+3 1.80
8.00e+3 1.85
1.03e+4 1.90
1.30e+4 1.95
1.65e+4 2.00
2.10e+4 2.05
2.60e+4 2.10
3.30e+4 2.15
4.20e+4 2.20
6.25e+4 2.25
9.90e+4 2.30
```

Fig. 28. *B-H* characteristic curve (left), and its characteristic curve file (right).

B. 14. Convergence history file

This file is a text file created by the solver module. Fig. 29 shows an example of a convergence history file for a time-harmonic eddy current analysis (see Appendix C.1 for details). The following are recorded in this file:

- Parallel sparse matrix solver's conditions (`HDDM solver's condtions`)
- Degrees of freedom of interface problem (`DOF on interface`)
- Non-parallelized sparse matrix solver's conditions for solving the sparse matrices in the subdomains (`Linear solver's condtions`)
- Start time of parallel sparse matrix solver from start of module execution
- Convergence history
- End time of parallel sparse matrix solver execution

Since all data except for the convergence history are preceded by a "#", Gnuplot can plot the convergence history without modifying the file.

The file name is preceded by "`log_g_HDDM_`" followed by the keyword of the analysis. In a non-linear analysis, the convergence history of all non-linear steps is recorded in the same file. In a non-steady analysis, the convergence history is recorded for each time step in a file with the time step number added, such as"`log_g_HDDM_NS_Eddy_10`". The files with "`Jo`" record the convergence histories when analysis to compute the correction amount of the current density is performed. In a non-steady analysis, analysis is performed by material ID, so the material ID is added to the file name, such as "`log_g_HDDM_Jo_2`". For analyses other than non-steady analyses, coils are analyzed together, so only one file, "`log_g_HDDM_Jo`", is recorded.

```
#  HDDM solver's condtions for Time-harmonic Eddy Current Analysis
#    HDDM                   : COCG
#    Matrix format          : implicit
#    Preconditioner         : Diag
#    Convergence Criterion  : 0.001
#    Divergence Criterion   : 1e+10
#    Maximum number of loop : -1
#    Messages from HDDM will be outputted
#    Matrixes in subdomains are kept
#    HDDM restart file will not be outputted
#  DOF on interface : 5680
#  Linear solver's condtions for current analysis
#    Solver                 : COCG
#    Matrix format          : AIJ
#    Preconditioner         : ICC
#      Acceleration factor  : 1.2
#    Convergence Criterion  : 1e-09
#    Divergence Criterion   : 1e+10
#    Maximum number of loop : -1
#    Messages from solver will not be outputted
# time 0.0147901 [s]
0 1.000000e+00  # Norm 3.136951e-04
1 2.220798e+00
2 2.381916e+00
    :
    :
    :
70 2.120706e-03
71 1.492400e-03
72 8.941657e-04
# time 0.927617 [s]
```

Fig. 29. "`sample_data/cake/done/calc_log/log_g_HDDM_TH_Eddy`".

B. 15. **Other files**

B. 15. 1. **MSH file**

This is a text file that records the mesh. For the format, see section 6.3 of the ADVENTURE_TetMesh manual.

B. 15. 2. **FGR file**

This is a text file that records FaceGroup data. For the format, see section 9.4 of the ADVENTURE_BCtool manual.

B. 15. 3. MSHX file

This is a text file that records the mesh. It is an extension of the MSH file format and can record mixed elements. Mixtures of elements with different dimensions are not allowed. This file can handle the following elements for each dimension: The values in parentheses are the types of element described below.

1D	: Linear line element (2)	Quadratic line element (3)
2D	: Linear triangular element (3)	Quadratic triangular element (6)
	Linear quadrilateral element (4)	Quadratic quadrilateral element (8)
3D	: Linear tetrahedral element (4)	Quadratic tetrahedral element (10)
	Linear hexahedral element (8)	Quadratic hexahedral element (20)
	Linear triangular prism element (6)	Quadratic triangular prism element (15)
	Linear square pyramid element (5)	Quadratic square pyramid element (13)

The format of this file is as shown in Fig. 30. At the beginning of the file, the dimension of mesh (*dimension*) is recorded. Below that, element connectivity, node coordinates, and volume information are recorded in that order.

For element connectivity, the number of elements (*NumOfElements*) is recorded first, and below that, the types of element (*TypeOfElement*) and the nodes that make up that element (*Node0 Node1 ...*) are recorded as a set for the number of elements. The number of nodes that make up each element matches the type of element (*TypeOfElement*).

For node coordinates, the number of nodes (*NumOfNodes*) is recorded first, and below that, the node coordinates (*CoordinateX CoordinateY ...*) are recorded for the number of nodes. The coordinate dimensions match the dimension (*dimension*) at the beginning.

For the number of volumes, the number of volumes (*NumOfVolumes*) is recorded first. Below that, the information for each volume is recorded for the number of volumes. For example, the information for the 0th volume first lists the number of elements that make up the 0th volume (*NumOfElementsInVolume0*), and below that the 0th element (*Element0_0*), the 1st element (*Element1_0*), and so on up to "*NumOfElementsInVolume0*-1"th of the volumes that make up the 0th volume.

```
dimension
NumOfElements
  TypeOfElement Node0 Node1 …
    :
NumOfNodes
  CoordinateX CoordinateY …
    :
NumOfVolumes
NumOfElementsInVolume0
  Element0_0
  Element1_0
    :
NumOfElementsInVolume1
  Element0_1
  Element1_1
    :
```

Fig. 30. MSHX file.

B. 15. 4. **FGRX file**

This is a text file which records FaceGroup data. It is the FGR file format with the information that can be obtained from MSH and MSHX files reduced.

The format of this file is as shown in Fig. 31. At the beginning of the file, the number of FaceGroups recorded in this file (*NumOfFaceGroups*) is recorded. Below that is the FaceGroup information.

For example, in the FaceGroup information for the 0th FaceGroup, first the number of element faces that make up the 0th FaceGroup (*NumOfFacesInFaceGroup0*) is recorded. Below that, as information for the 0th element face that makes up the 0th FaceGroup, the element number to which that element face belongs (*Element0_0*) and the element face number within the element (*FaceInElement0_0*) are recorded as a set. This is repeated up to the "*NumOfFacesInFaceGroup0*-1"th element face. After recording the information for the 0th FaceGroup, the information for the 1st FaceGroup is recorded, and so on up to the "*NumOfFaceGroups*-1"th FaceGroup.

```
NumOfFaceGroups
NumOfFacesInFaceGroup0
  Element0_0 FaceInElement0_0
  Element1_0 FaceInElement1_0
    :
    :
NumOfFacesInFaceGroup1
  Element0_1 FaceInElement0_1
  Element1_1 FaceInElement1_1
    :
    :
```

Fig. 31. FGRX file.

C. Analysis examples: from model creation to analysis

This examples show model creation using the ADVENTURE System, domain decomposition using ADVENTURE_Metis, and analysis of this model using ADVENTURE_Magnetic. Commercial CAD, ADVENTURE_TriPatch, ADVENTURE_TetMesh, and ADVENTURE_BCtool modules are used to create the models. The versions of each module used in this analysis example are as follows. For details, refer to the manual for each module.

- ADVENTURE_TriPatch : 1.8
- ADVENTURE_TetMesh : 0.91b
- ADVENTURE_BCtool : 2.1.1
- ADVENTURE_Metis : 1.1

In addition, when creating a model with the ADVENTURE System, ADVENTURE_BCtool may fail to extract the mesh surface, making it impossible to create the model. Therefore, in this example, a standard model creation method is introduced in C.1, and also an example of how to avoid model creation failure is introduced in C.2.

C. 1. Standard model creation method: time-harmonic eddy current analysis

As an analysis example, cake model [4][14] is used, which is a model for verifying the accuracy of eddy current analysis using a solenoidal coil with unlimited length in Fig. 32. As shown in Fig. 33, taking into account the symmetry of the problem, the model to be analyzed is a region with a central angle of 20° and a height of 0.1 [m]. The radius of the conductor is 0.1 [m]. The magnetic reluctivity ν is $1/(4\pi\times10^{-7})$ [m/H] for the entire analysis region, the electrical conductivity σ of the conductor is 7.7×10^{6} [S/m], and the angular frequency ω is $2\pi\times60$ [rad/s]. The magnitudes of the real and imaginary parts of the excitation current density J flowing through the coil are 50 and 0 [A/m^2]. The boundary conditions are imposed as $A\times n=0$ on the faces of $\theta=0°$ (green) and $\theta=20°$ (red) as shown in Fig. 34.

For reference, the file containing all the steps that follow can be found in

```
sample_data/cake/done/
```

PNG files of the visualization results using ParaView can also be found in

```
sample_data/cake/done/result/
```

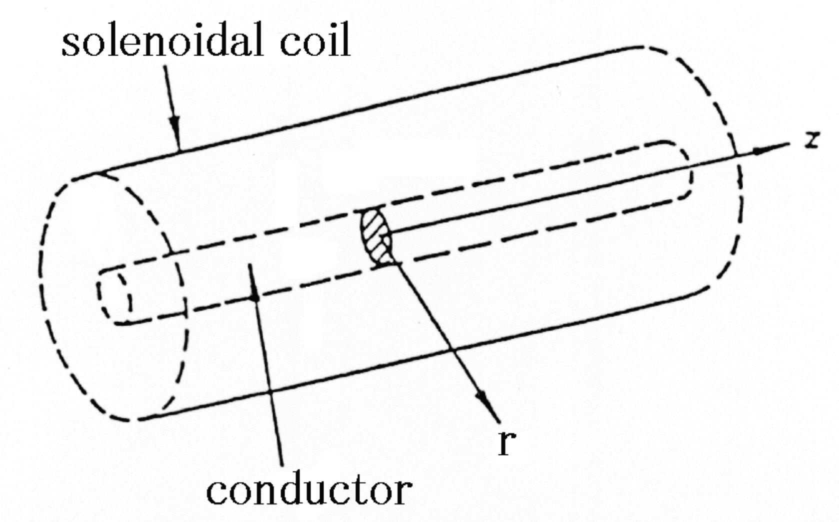


Fig. 32. Solenoidal coil with unlimited length.

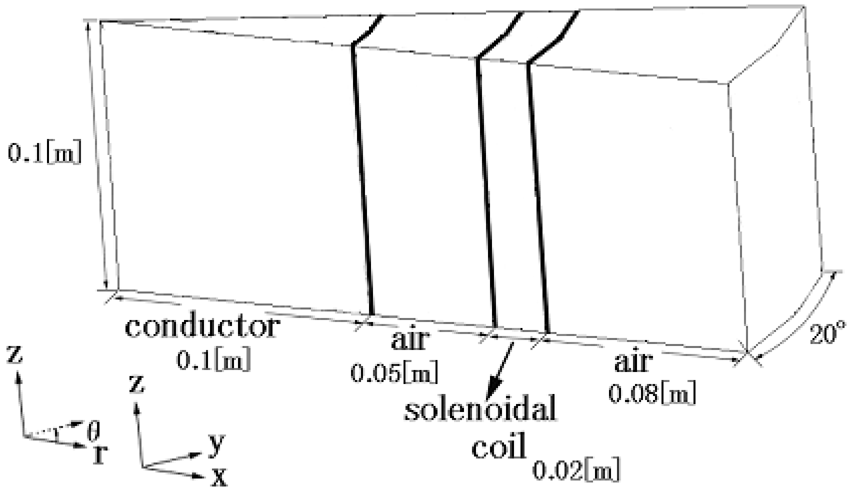


Fig. 33. Cake model.

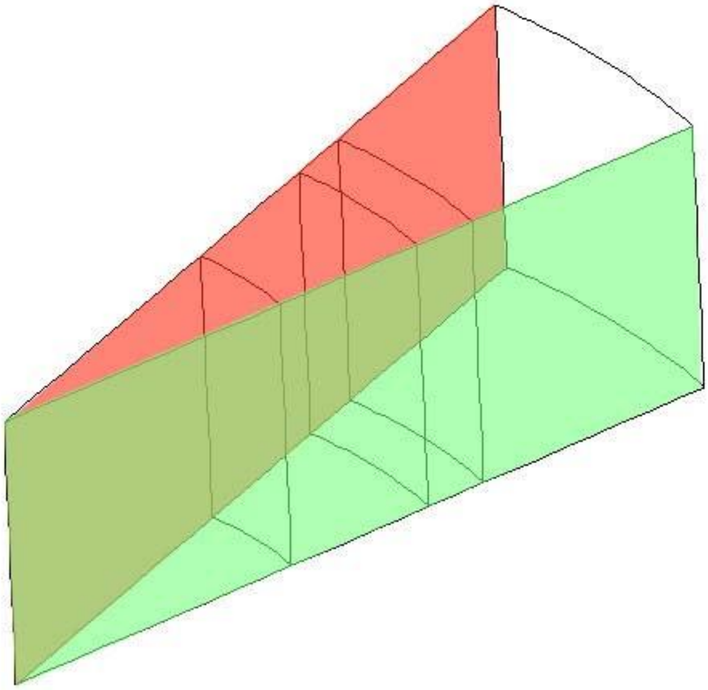


Fig. 34. Boundary conditions.

(1) Preparing IGES files

First, prepare IGES files using a commercial CAD program. IGES files are input to ADVENTURE_TriPatch, but not all IGES files are accepted. For the requirements for IGES files, see section 6.1 of the ADVENTURE_TriPatch manual.

Here, the files in

```
sample_data/cake/igs/
```

are used.

- `conductor.igs`	: Conductor region	$(0 \leq r \leq 0.1)$
- `air01.igs`	: Inner air region	$(0 \leq r \leq 0.1)$
- `coil.igs`	: Coil region	$(0 \leq r \leq 0.1)$
- `air02.igs`	: Outer air region	$(0 \leq r \leq 0.1)$
- `cake.ptn`	: Node density data file	

The node density data file is used to set the mesh fineness (average length of element sides, coarseness/density). For details, see section 6.4 of the ADVENTURE_TriPatch manual.

(2) Creating face patches

Surface patches are created using ADVENTURE_TriPatch based on each IGES file, and then merged using the surface patch merging program `mrpach`. For details on creating and merging surface patches, please refer to the ADVENTURE_TriPatch manual.

- Creating (copy) node density data files and create surface patches:

  Surface patch data files (.pcm) and surface patch data group files (.pcg) are created.

  (Sections 6.2 and 6.3 of the ADVENTURE_TriPatch manual)

  - Conductor region

```
% cp cake.ptn conductor.ptn
% ADVENTURE_TriPatch conductor conductor
```

  - Inner air region

```
% cp cake.ptn air01.ptn
% ADVENTURE_TriPatch air01 air01
```

  - Coil region

```
% cp cake.ptn coil.ptn
% ADVENTURE_TriPatch coil coil
```

  - Outer air region

```
% cp cake.ptn air02.ptn
% ADVENTURE_TriPatch air02 air02
```

- Merging surface patches: Merge from the innermost one while recording it in temporary files.

  Select the patches to be merged so that they are adjacent to each other.

  Also, the volume numbers are assigned starting from 0 in the order of merging.

```
% mrpach conductor.pcm conductor.pcg air01.pcm air01.pcg –o tmp01.pcm –g tmp01.pcg
% mrpach tmp01.pcm tmp01.pcg coil.pcm coil.pcg –o tmp02.pcm –g tmp02.pcg
% mrpach tmp02.pcm tmp02.pcg air02.pcm air02.pcg –o cake.pcm –g cake.pcg
```

(3) Generating mesh data

ADVENTURE_TetMesh is used to generate a mesh based on the surface patch. Since ADVENTURE_Magnetic treats quadratic tetrahedral elements, be sure to execute `advtmesh9s`. For details on generating mesh data, refer to the ADVENTURE_TetMesh manual. MSH files are explained in Section 6.3.

- A surface mesh data file (.pcc) is generated.
  A corrected node density control file (c.ptn) used by `advtmesh9m` is also generated.
  `% advtmesh9p cake -d`

- A linear tetrahedral element mesh (c.msh) is generated
  `% advtmesh9m cakec`

- Generating quadratic nodes. An MSH file (s.msh) is generated.
  `% advtmesh9s cakec`

(4) Setting boundary conditions

Boundary conditions are set using ADVENTURE_BCtool. The boundary conditions given here are as shown in Fig. 34. However, to create an FEA model file, the tools included with ADVENTURE_Magnetic will be used as described in the next section, so `makefem3` from BCtool will not be used. For details on setting boundary conditions, please refer to the ADVENTURE_BCtool manual.

- Extraction of mesh surfaces (.pch, .pcg, .trn) and creation of FaceGroup data (.fgr).
  For the format of each file, please refer to sections 9.5, 9.7, 9.8, and 9.4 of the ADVENTURE_BCtool manual.
  `% msh2pch cakecs.msh 9`

- Setting boundary conditions: When starting from the command line, start with the following command.
  `% BcGUI2 cakecs_9.pch cakecs_9.pcg`

If you start from a shortcut on the desktop, you must select the same file as the one passed in the above command after starting. (“`File`” -> “`Open file`”, Fig. 35, Fig. 36)

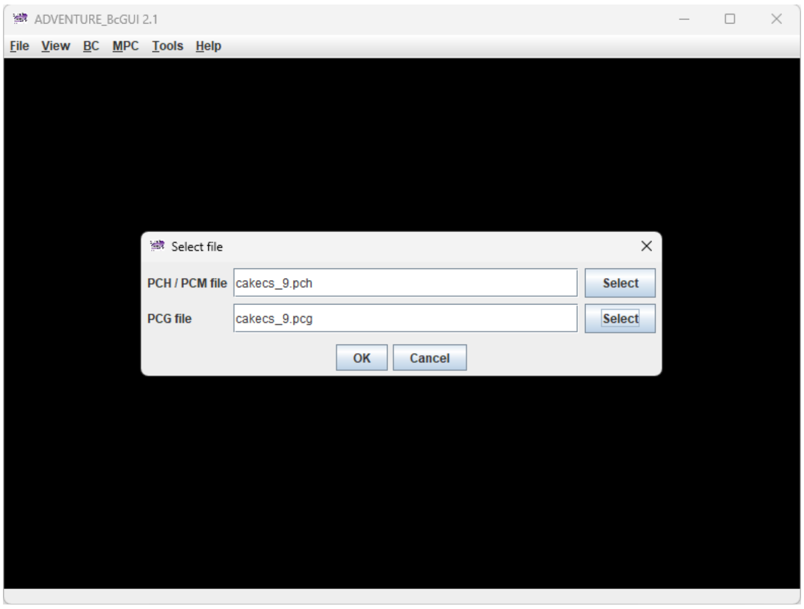


Fig. 35. Loading files.

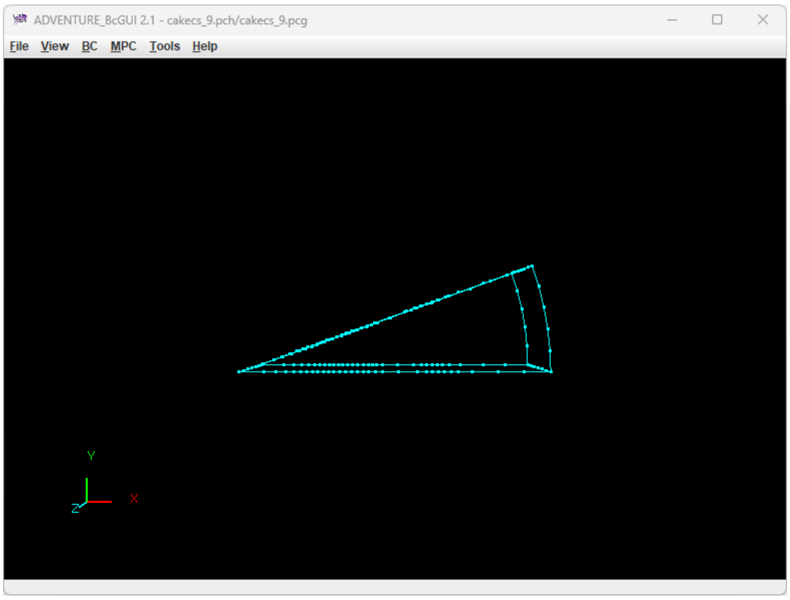


Fig. 36. Immediately after launching (immediately after loading files).

First, set the boundary condition on the $\theta = 0°$ surface (green surface in Fig. 34). Rotate the object, etc., to select the nodes on the edge of the surface with left click, and right-click until the target surface turns yellow (Fig. 37).

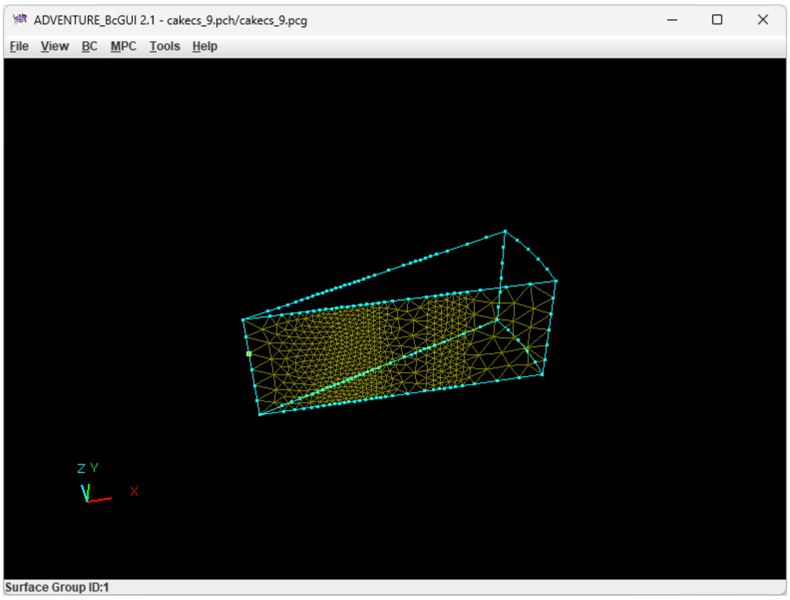


Fig. 37. Selecting the $\boldsymbol{\theta} = \mathbf{0}°$ surface.

Select the type of boundary condition (“BC” -> “`BC (Magnetic)`” -> “`Add Magnetic Vector Potential`”), check the box to the left of “`Normal`” (Fig. 38), and click “`OK`”.

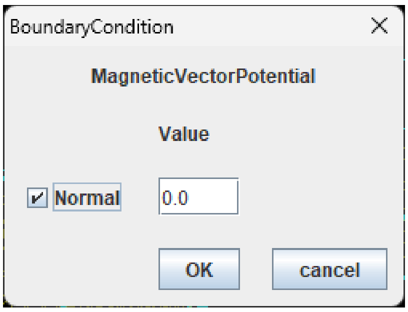


Fig. 38. Boundary condition $\boldsymbol{A} \times \boldsymbol{n} = \mathbf{0}$.

Next, set the same boundary condition to the $\theta = 20°$ surface (red surface in Fig. 34).

There are two ways to check whether the boundary conditions have been set appropriately. One is to display the contents of the boundary condition file that is output (“`View`” -> “`Boundary Condition`” -> “`Cnd format`”, Fig. 39). The other is to display the surfaces to which the boundary conditions are set (“`View`” -> “`Boundary Condition`” -> “`View Magnetic Vector Potential`”, Fig. 40). The surfaces to which the selected boundary conditions are set are colored. If the surface is selected, it will not be colored, so if it is selected, right-click to deselect it.

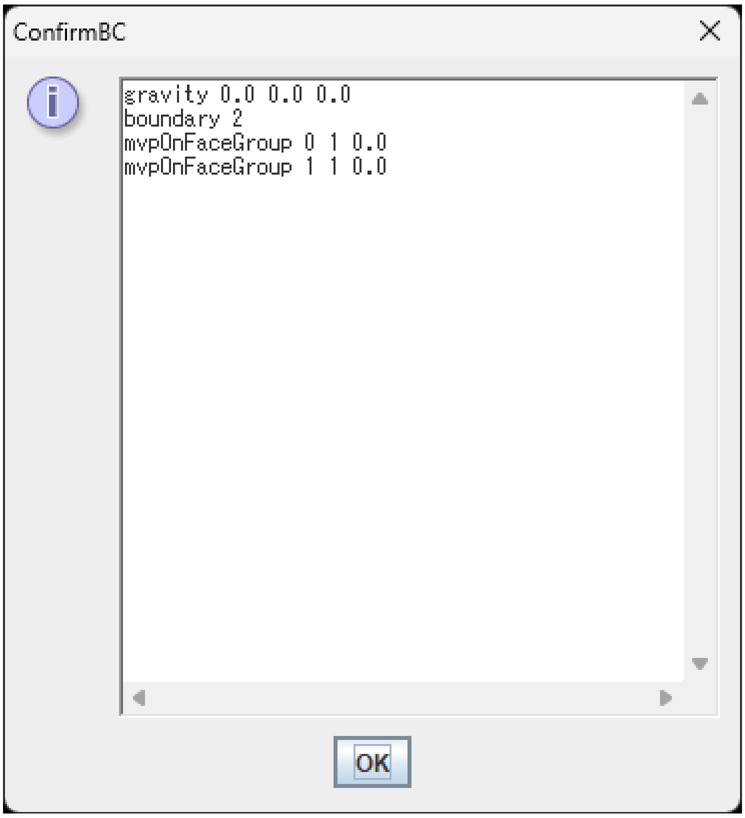


Fig. 39. Displaying the contents of the boundary condition file.

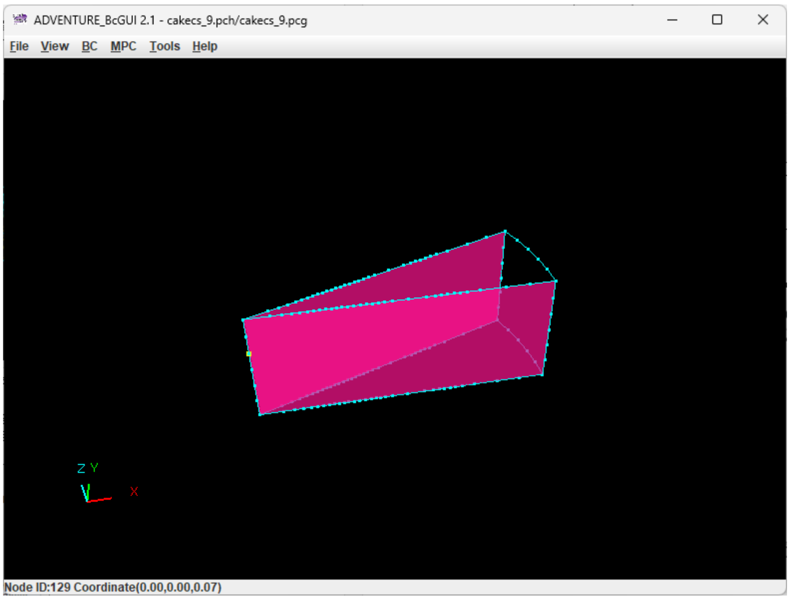


Fig. 40. Displaying the surfaces to which the boundary conditions are set.

Save the boundary condition file from "`File`" -> "`Save condition`". In this example, the file name is "`cake.cnd`" (Fig. 41). When you have finished, close `BcGUI2` from "`File`" -> "`Quit`".

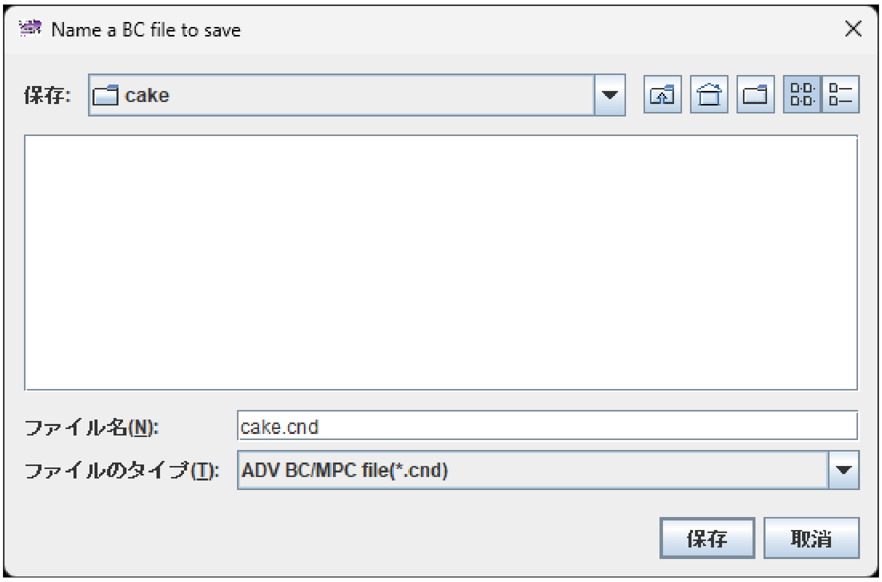


Fig. 41. Saving the boundary condition file.

(5) Creating an FEA model file

Tool for making FEA model `advmag2_makefem_Electromangetic` of ADVENTURE_Magnetic creates an FEA model file in ADVENTURE format from the mesh, physical quantites, and boundary conditions.

First, create the material property file “`cake.dat`”. For details on material property file, refer to section 9.9 of the ADVENTURE_BCtool manual. In ADVENTURE_Magnetic, physical quantities are given in the material data file. For this reason, the physical quantities setting function of the material property file is not used, and only the conversion function of the volume numbers to the material IDs is used.

The volume numbers of the mesh generated by ADVENTURE_TetMesh are assigned in the order in which the surface patches are merged with `mrpach`. In addition, if the mesh surface and volume boundary are extracted with the tool `msh2pcm` of ADVENTURE_BCtool, the volume numbers can also be confirmed in `BcGUI2`.

- Extraction of mesh surface and volume boundary (_V.pcm, _V.pcg).
  For the format, refer to Sections 9.6 and 9.7 of the ADVENTURE_BCtool manual.
  ```
  % msh2pcm cakecs.msh
  ```

- Starting `BcGUI2`.
  ```
  % BcGUI2 cakecs_V.pcm cakecs_V.pcg
  ```

When `BcGUI2` starts, each volume is numbered as shown in Fig. 42. If you are not sure where the numbers are, try rotating the model.

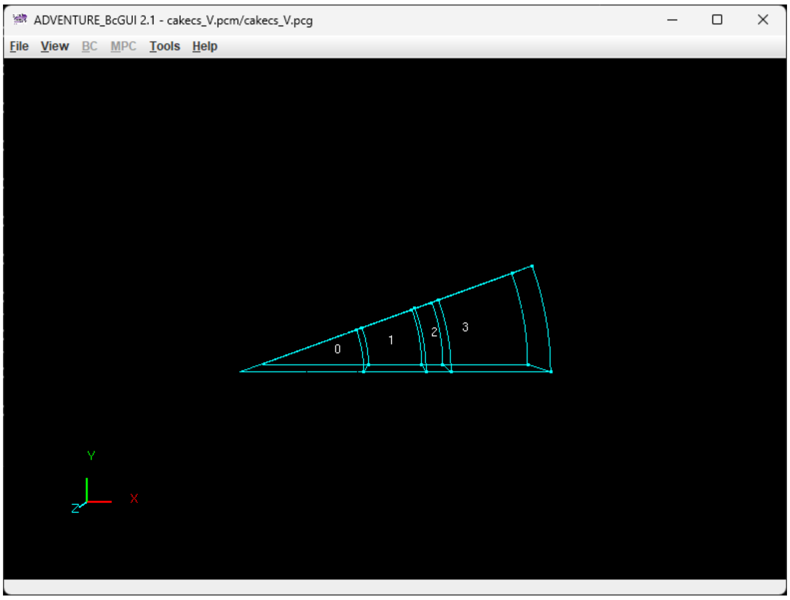


Fig. 42. Confirming the volume numbers.

The volume number and the correspondence of each region are as follows.

0 : Conductor region
1 : Inner air region
2 : Coil region
3 : Outer air region

Here, the volume numbers are used as the material IDs. An example where the volume numbers and material IDs do not match is shown in C.2.

"cake.dat" will look like Fig. 43. "#materialInfo" is the part related to the function of setting physical quantities and is not used in ADVENTURE_Magnetic. Therefore, "propertyN", which is the number of types of physical quantities, is "0". "#volumeInfo" is the part related to the function of converting volume numbers to material IDs. "volumeN" is the number of volumes the mesh has. Below this, list the material IDs to be assigned to each volume, the number of volumes.

```
#materialInfo
materialN 4
propertyN 0

#volumeInfo
volumeN 4
0
1
2
3
```

Fig. 43. Contents of "cake.dat".

Now it's ready, create the FEA model file "input.adv" with advmag2_makefem_Electromangetic.

- Creating an FEA model file

```
% advmag2_makefem_Electromangetic cakecs.msh cakecs_9.fgr cake.cnd cake.dat input.adv
```

(6) Creating an analysis directory

After creating a file for specifying the physical quantities, store it in the analysis directory along with the FEA model file.

The main physical quantities are specified in the material data file. The default name of this file is "`mtrl.dat`". The excitation current density to be applied to the coil is specified in the shape definition file. The name of this file is arbitrary, but here it is called "`coil.dat`". As mentioned at the beginning of this section, the magnetic reluctivity (`MagneticReluctivity`) is $1/(4\pi\times10^{-7}) \fallingdotseq$ 7.957747e+05 [m/H] for the entire analysis domain, the electrical conductivity (`ElectricalConductivity`) of the conductor region is 7.7×10^6 = 7.7e+06 [S/m], and the angular frequency (`AngularFrequency`) is $2\pi\times60 \fallingdotseq 376.99$ [rad/s]. The coil has material ID number 2, and uses a shape definition file, so the file type is "`md`". Enter these in "`mtrl.dat`" (Fig. 44).

```
MagneticReluctivity 4
 0 7.957747e+05
 1 7.957747e+05
 2 7.957747e+05
 3 7.957747e+05

Coil 1
 2 md coil.dat

ElectricalConductivity 1
 0 7.7e+06

AngularFrequency
 376.99
```

Fig. 44. Contents of "`mtrl.dat`".

Next, create the shape definition file "`coil.dat`". Since the coil region is sector-shaped, use "`DoubleSectorialCylinder`". Set the region a little larger so that the nodes at the end faces of the coil region are not judged to be outside the region due to numerical error. Since time-harmonic eddy current analysis treats complex numbers, give the real and imaginary parts of the current density. The magnitudes of the real and imaginary parts of the excitation current density flowing through the coil are 50 and 0 [A/m^2]. The real part corresponds to the antinode of the sinusoidal wave, and the imaginary part corresponds to the node 90° from there. Enter these in "`coil.dat`" (Fig. 45).

```
DoubleSectorialCylinder
 0.0 0.0 -0.05
 z 0.2
 deg -10.0 40.0
 0.14 0.18
 50.0 0.0
```

Fig. 45. Contents of "`coil.dat`".

Finally, create an analysis directory and a subdirectory to store the FEA model file, and move each file to the designated location. The analysis directory name will be "`cake`".

```
% mkdir -p cake/model_one
% mv input.adv cake/model_one/
% mv mtrl.dat cake/
% mv coil.dat cake/
```

(7) Domain decomposition

A hierarchically decomposed model is created using ADVENTURE_Metis based on the FEA model created so far. The option "`-difn 1`" should be specified when executing. This option is to specify the degrees of freedom on the internal boundary nodes as 1.

First, the numbers of parts and subdomains should be determined to decompose mesh hierarchically. Here, two PCs will be used to analyze in distributed-memory parallel mode. Therefore, the number of parts will be 2. The number of subdomains will be determined by formula (2) in section 2.2, so first check the number of elements $N_{element}$ in the FEA model file with `advinfo`.

```
% advinfo cake/model_one/input.adv
```

The "`num_items`" in "`content_type=Element`" is the number of elements, which is 13,075 in this case. If the number of elements per subdomain, $n_{element}$, is 100, then

$$\frac{13{,}075}{2 \times 100} = 65.375 \ . \tag{6}$$

The exact value obtained here does not have to be used. Any round number can be used. Here, the number of subdomains is set to 65.

```
% mpirun –np 2 adventure_metis –difn 1 cake/model_one/input.adv cake 65
```

Note that in shared-memory parallel mode and hybrid parallel mode, processing is assigned to threads on a subdomain basis, so if the number of subdomains is a multiple of the number of threads, there will be less waste of computational resources.

(8) Executing the analysis

Perform a time-harmonic eddy current analysis (`TH_Eddy`) using the electromagnetic field analysis module. The option "`-data-dir`" is required because the analysis directory name is different from the default.

```
% mpirun –np 2 advmag2_HDDM_Electromagnetic-p TH_Eddy -data-dir cake
```

(9) Creating a visualization file

Create a visualization file using `advmag2_hddmmrg`. Unknowns "`MagneticVectorPotentail`" and "`ElectromagneticScalarPotential`", which are singular (integral constants are not determined) will be excluded. The name of the visualization file will be "`cake.vtu`".

```
% advmag2_hddmmrg cake.vtu -2 MagneticVectorPotentail ElectromagneticScalarPotential
-data-dir cake
```

(10) Visualization

Here is an example of visualization using ParaView.

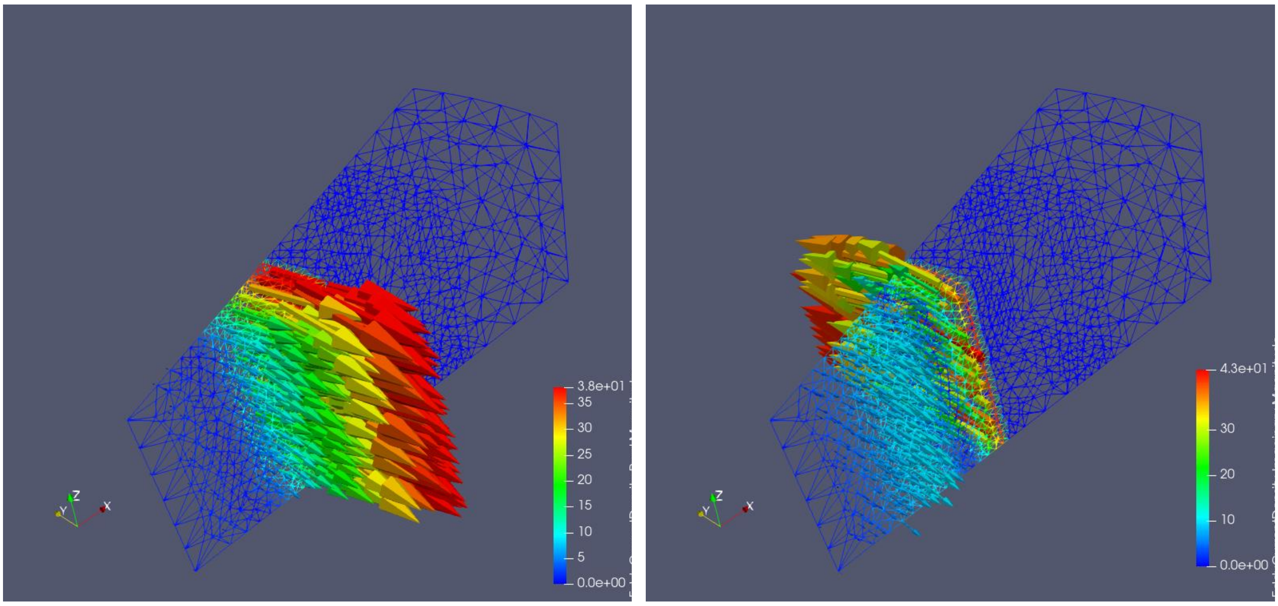
The ParaView settings after loading “`cake.vtu`” and pressing “`Apply`” are as follows.

- Data type : “`Solid Color`” -> Check each figure.
- Model surface : “`Representation`” -> “`Wireframe`”
- Color bar type : “`Choose Preset`” -> “`Blue to Red Rainbow`”
- Vector display : “`Filters`” -> “`Common`” -> “`Glyph`”
  - Scale mode : “`Vector Scale Mode`” -> “`Scale by Magnitude`”
  - Scale factor : “`Scale Factor`” -> Check each the title of the figure

The VTU files used to create Fig. 46 and Fig. 47, as well as the PNG files of these figures, are stored in

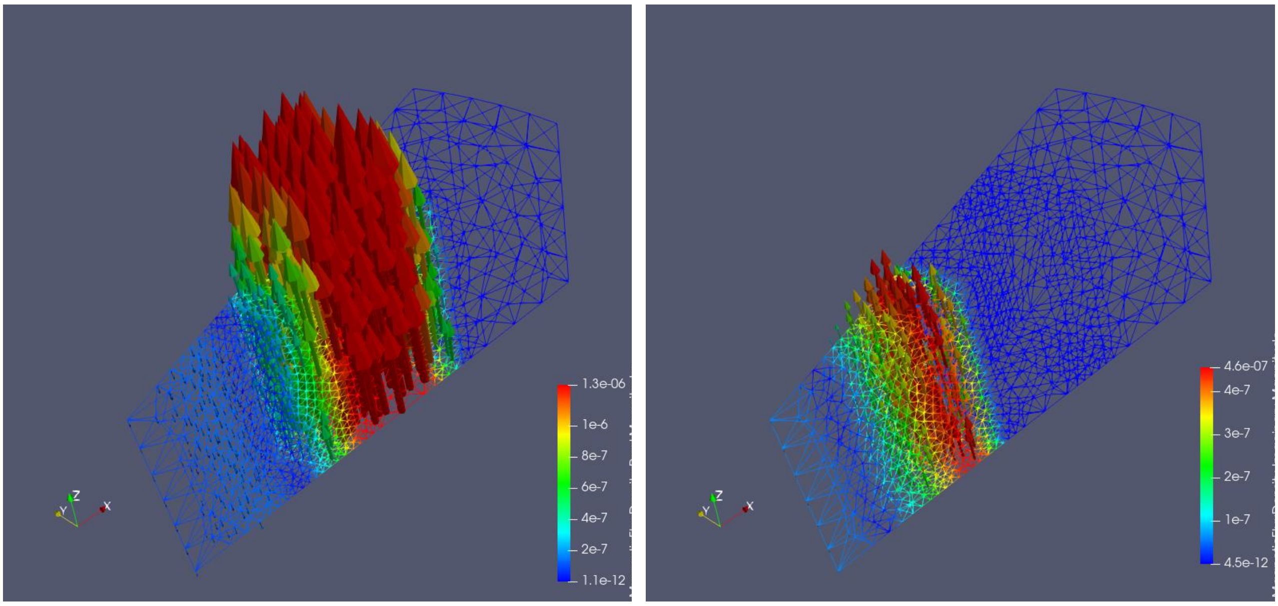
`sample_data/cake/done/result/`

The PNG files can be saved from “`File`” -> “`Save Screenshot`”.



(a) Real part (`EddyCurrentDensity-Real`) (b) Imaginary part (`EddyCurrentDensity-Imaginary`)

Fig. 46. Eddy current density (scale factor: 0.002).



(a) Real part (`MagneticFluxDensity-Real`) (b) Imaginary part (`MagneticFluxDensity-Imaginary`)

Fig. 47. Magnetic flux density (scale factor: 50,000).

C. 2. Model creation method when the bonding surfaces of materials do not match: Non-linear magnetostatic analysis

Most models used in electromagnetic field analysis are composed of multiple materials. Therefore, multiple surface patches must be merged using the surface patch merging program `mrpach` in ADVENTURE_TriPatch. However, if the shapes of the merging surfaces of the materials do not match, as in the red surface in Fig. 48, they will not be merged properly. In such cases, it may be possible to succeed by dividing material <3> into <4> and <5>, as in Fig. 49, and creating an IGES file to match the shapes of the merging surfaces. This section explains such cases with specific examples.

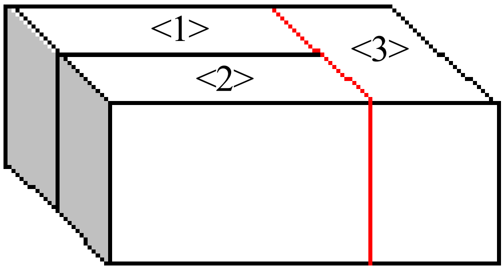


Fig. 48. Merging surfaces of materials.

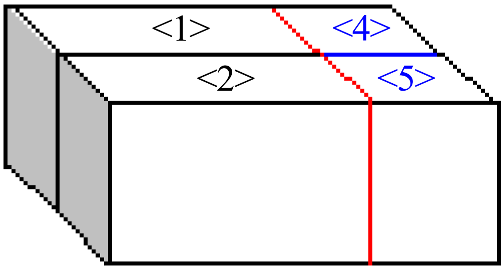


Fig. 49. Dividing a material to match the shapes of the merging surfaces.

As an analysis example, the axisymmetric model [14] in Fig. 50 is used. This model is an attracting magnet with a gap where the upper and lower iron cores have different shapes and face each other, and Fig. 50 is its cross-section. The analysis target is the region in Fig. 50 rotated 10 degrees around the *z*-axis. The excitation current density in the coil region is assumed to be 3×10^7 [A/m^2] flowing perpendicular to the cross section of Fig. 50. The magnetic resistivity ν is $1/(4\pi)\times10^7$ [m/H] in the air and coil region, and the material of the annular magnetic body is SS41P. A non-linear analysis is performed in this region with nonlinearity as shown in Fig. 51. The boundary conditions are $A \times n = 0$ on all surfaces.

In this model, the merging surfaces of the materials do not match on the right side of the coil (x = 85 mm in Fig. 50). Therefore, part of the coil region is divided as shown in Fig. 52, and the merging surfaces are made to match.

The specific steps are shown below.

For reference, the file containing all the steps that follow can be found in

`sample_data/shaft/done/`

The PNG file showing the visualization results using ParaView can also be found in

`sample_data/shaft/done/result/`

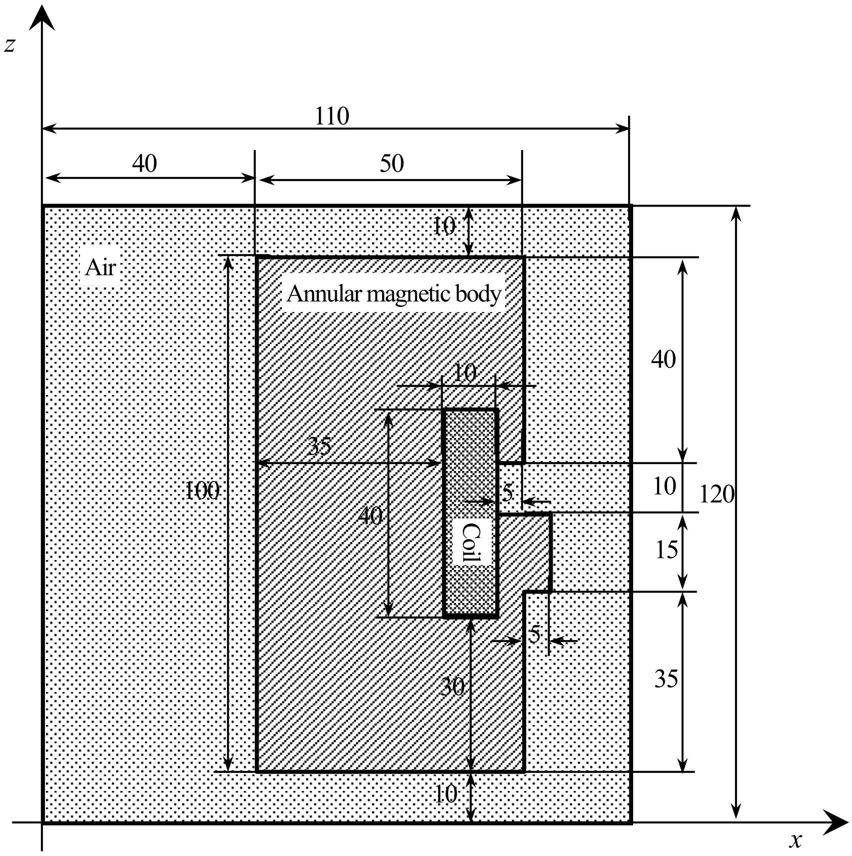


Fig. 50. Cross-section of the axisymmetric model (unit: mm).

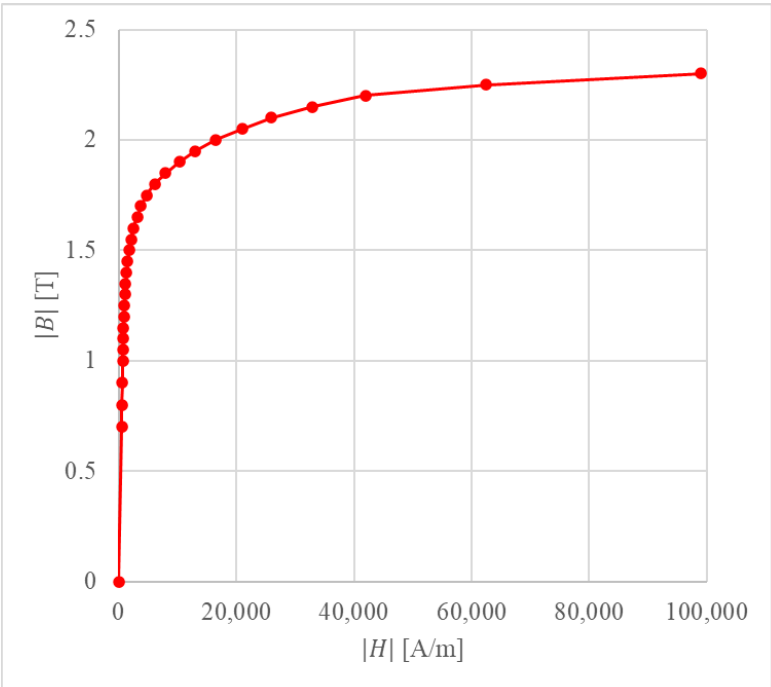


Fig. 51. *B-H* characteristic curve.

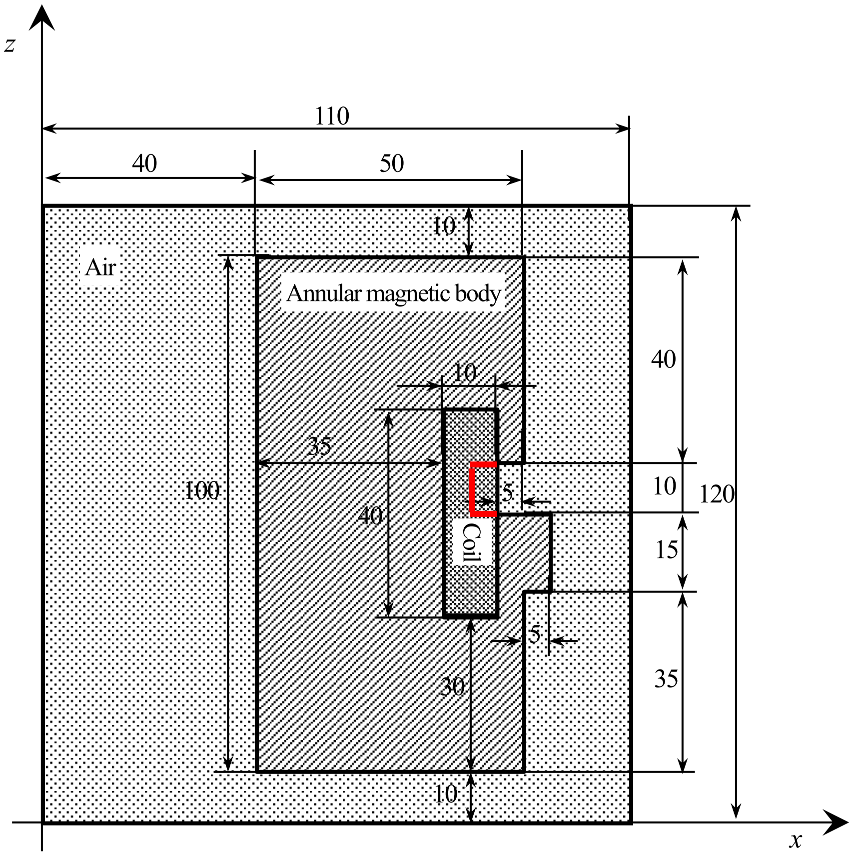


Fig. 52. Dividing of coil region.

(1) Preparing IGES files

First, prepare IGES files using a commercial CAD program. IGES files are input to ADVENTURE_TriPatch, but not all IGES files are accepted. For the requirements for IGES files, see section 6.1 of the ADVENTURE_TriPatch manual.

Here, the files in

```
sample_data/shaft/igs/
```

are used. In Fig. 50 and Fig. 52, the dimensions are written in mm, but <u>these IGES files have been created by converting the dimensions to m</u>.

- `coil01.igs` : Coil region (other than the area surrounded by red in Fig. 52)
- `coil02.igs` : Coil region (area surrounded by red in Fig. 52)
- `mag.igs` : Annular magnetic body region
- `air.igs` : Air region
- `shaft.ptn` : Node density data file

The node density data file is used to set the mesh fineness (average length of element sides, coarseness/density). For details, see section 6.4 of the ADVENTURE_TriPatch manual.

(2) Creating face patches

Surface patches are created using ADVENTURE_TriPatch based on each IGES file, and then merged using the surface patch merging program `mrpach`. For details on creating and merging surface patches, please refer to the ADVENTURE_TriPatch manual.

- Creating (copy) node density data files and create surface patches:
  Surface patch data files (.pcm) and surface patch data group files (.pcg) are created.
  (Sections 6.2 and 6.3 of the ADVENTURE_TriPatch manual)
  - Coil region (other than the area surrounded by red in Fig. 52)
```
% cp shaft.ptn coil01.ptn
% ADVENTURE_TriPatch coil01 coil01
```
  - Coil region (area surrounded by red in Fig. 52)
```
% cp shaft.ptn coil02.ptn
% ADVENTURE_TriPatch coil02 coil02
```
  - Annular magnetic body region
```
% cp shaft.ptn mag.ptn
% ADVENTURE_TriPatch mag mag
```
  - Air region
```
% cp shaft.ptn air.ptn
% ADVENTURE_TriPatch air air
```

- Merging surface patches: Merge from the innermost one while recording it in temporary files.
  Select the patches to be merged so that they are adjacent to each other.
  Also, the volume numbers are assigned starting from 0 in the order of merging.
```
% mrpach coil01.pcm coil01.pcg coil02.pcm coil02.pcg -o tmp01.pcm -g tmp01.pcg
% mrpach tmp01.pcm tmp01.pcg mag.pcm mag.pcg -o tmp02.pcm -g tmp02.pcg
% mrpach tmp02.pcm tmp02.pcg air.pcm air.pcg -o shaft.pcm -g shaft.pcg
```

(3) Generating mesh data

ADVENTURE_TetMesh is used to generate a mesh based on the surface patch. Since ADVENTURE_Magnetic treats quadratic tetrahedral elements, be sure to execute `advtmesh9s`. For details on generating mesh data, refer to the ADVENTURE_TetMesh manual. MSH files are explained in Section 6.3.

- A surface mesh data file (.pcc) is generated.
  A corrected node density control file (c.ptn) used by `advtmesh9m` is also generated.
```
% advtmesh9p shaft -d
```

- A linear tetrahedral element mesh (c.msh) is generated
```
% advtmesh9m shaftc
```

- Generating quadratic nodes. An MSH file (s.msh) is generated.
```
% advtmesh9s shaftc
```

(4) Setting boundary conditions

Boundary conditions are set using ADVENTURE_BCtool. Here, boundary conditions are applied to all faces. However, to create an FEA model file, the tools included with ADVENTURE_Magnetic will be used as described in the next section, so `makefem3` from BCtool will not be used. For details on setting boundary conditions, please refer to the ADVENTURE_BCtool manual.

- Extraction of mesh surfaces (.pch, .pcg, .trn) and creation of FaceGroup data (.fgr).
  For the format of each file, please refer to sections 9.5, 9.7, 9.8, and 9.4 of the ADVENTURE_BCtool manual.

```
% msh2pch shaftcs.msh 18
```

- Setting boundary conditions: When starting from the command line, start with the following command.

```
% BcGUI2 shaftcs_18.pch shaftcs_18.pcg
```

If you start from a shortcut on the desktop, you must select the same file as the one passed in the above command after starting. (“`File`” -> “`Open file`”, Fig. 53, Fig. 54)

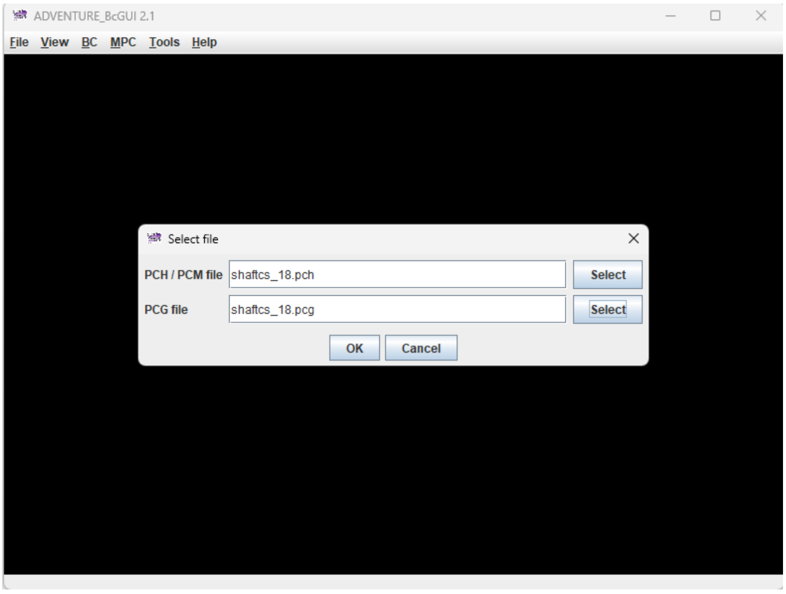


Fig. 53. Loading files.

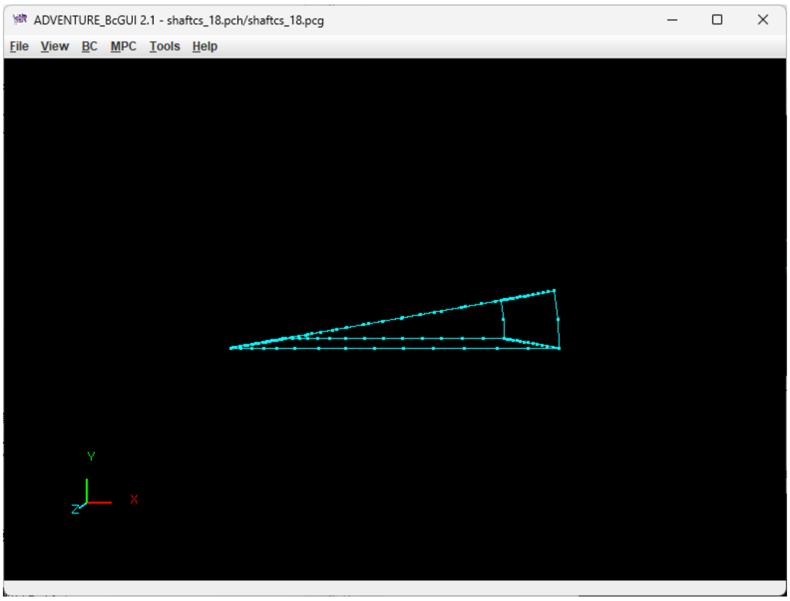


Fig. 54. Immediately after launching (immediately after loading files).

First, set the boundary condition on the $\theta = 0°$ surface. Rotate the object, etc., to select the nodes on the edge of the surface with left click, and right-click until the target surface turns yellow (Fig. 55).

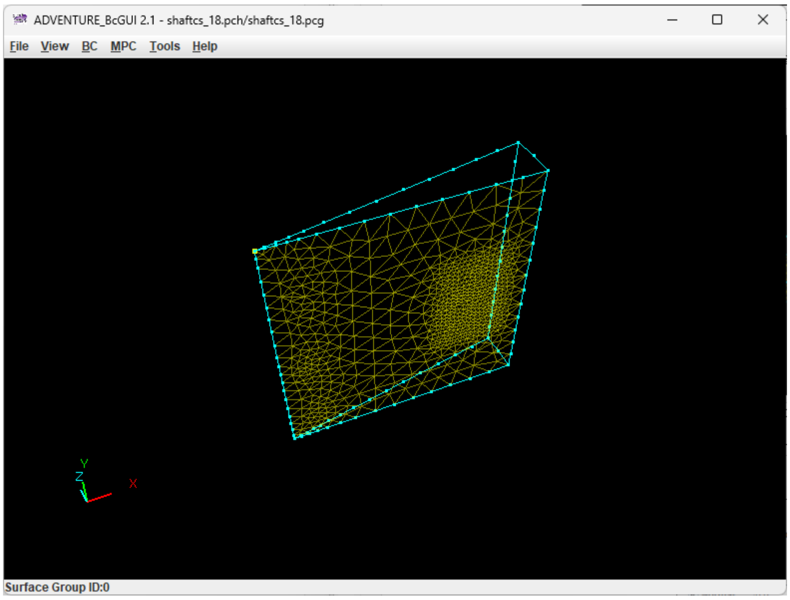


Fig. 55. Selecting the $\boldsymbol{\theta} = \mathbf{0}°$ surface.

Select the type of boundary condition (“`BC`” -> “`BC (Magnetic)`” -> “`Add Magnetic Vector Potential`”), check the box to the left of “`Normal`” (Fig. 56), and click “`OK`”.

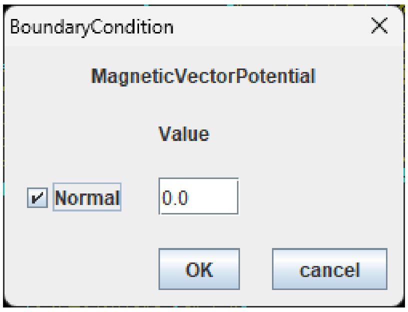


Fig. 56. Boundary condition $\boldsymbol{A} \times \boldsymbol{n} = \mathbf{0}$.

Similar boundary conditions are set on the other four faces.

There are two ways to check whether the boundary conditions have been set appropriately. One is to display the contents of the boundary condition file that is output (“`View`” -> “`Boundary Condition`” -> “`Cnd format`”, Fig. 57). The other is to display the surfaces to which the boundary conditions are set (“`View`” -> “`Boundary Condition`” -> “`View Magnetic Vector Potential`”, Fig. 58). The surfaces to which the selected boundary conditions are set are colored. If the surface is selected, it will not be colored, so if it is selected, right-click to deselect it.

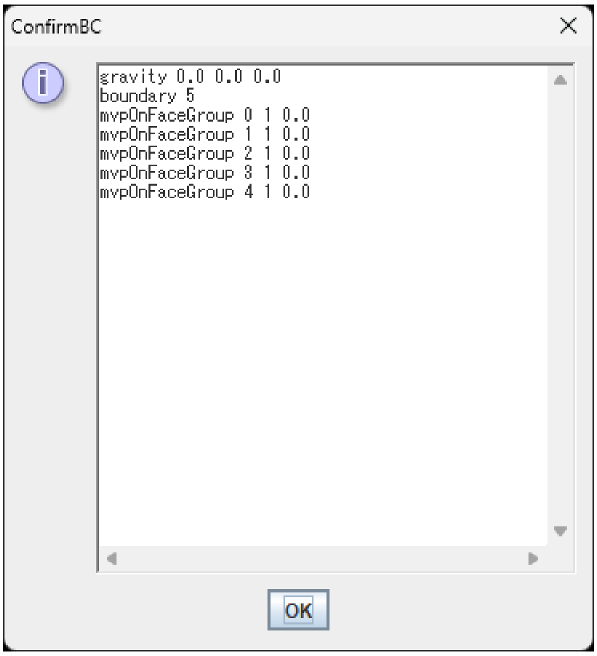


Fig. 57. Displaying the contents of the boundary condition file.

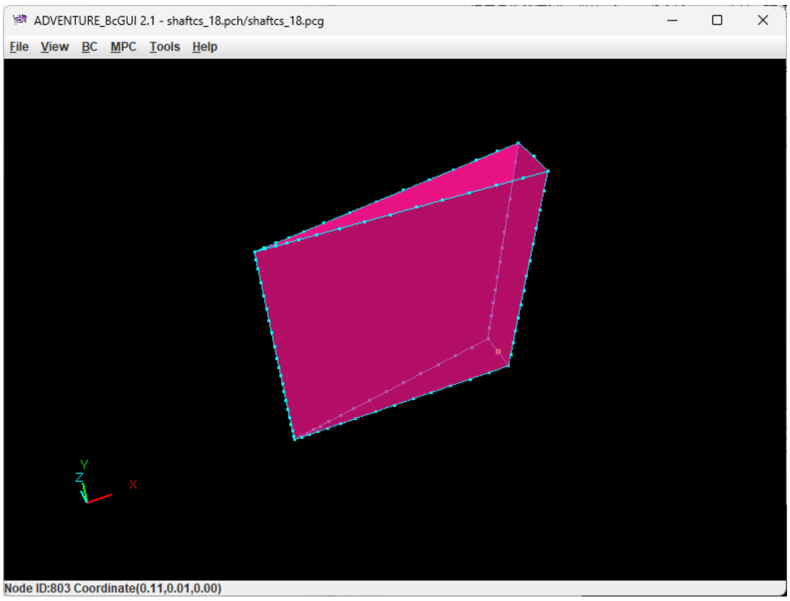


Fig. 58. Displaying the surfaces to which the boundary conditions are set.

Save the boundary condition file from "`File`" -> "`Save condition`". In this example, the file name is "`shaft.cnd`" (Fig. 59). When you have finished, close `BcGUI2` from "`File`" -> "`Quit`".

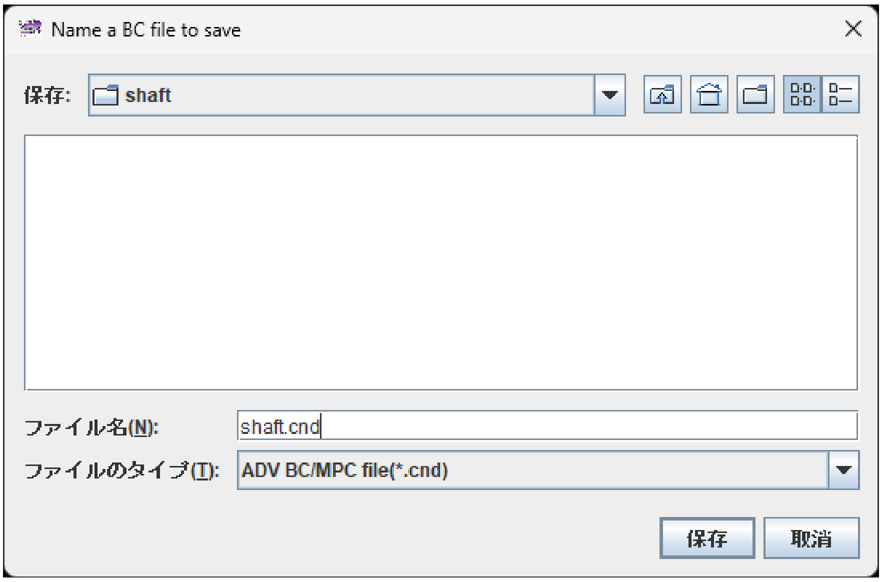


Fig. 59. Saving the boundary condition file.

(5) Creating an FEA model file

Tool for making FEA model `advmag2_makefem_Electromangetic` of ADVENTURE_Magnetic creates an FEA model file in ADVENTURE format from the mesh, physical quantites, and boundary conditions.

First, create the material property file "`shaft.dat`". For details on material property file, refer to section 9.9 of the ADVENTURE_BCtool manual. In ADVENTURE_Magnetic, physical quantities are given in the material data file. For this reason, the physical quantities setting function of the material property file is not used, and only the conversion function of the volume numbers to the material IDs is used.

The volume numbers of the mesh generated by ADVENTURE_TetMesh are assigned in the order in which the surface patches are merged with `mrpach`. In addition, if the mesh surface and volume boundary are extracted with the tool `msh2pcm` of ADVENTURE_BCtool, the volume numbers can also be confirmed in `BcGUI2`.

- Extraction of mesh surface and volume boundary (_V.pcm, _V.pcg).

For the format, refer to Sections 9.6 and 9.7 of the ADVENTURE_BCtool manual.

```
% msh2pcm shaftcs.msh
```

- Starting `BcGUI2`.

```
% BcGUI2 shaftcs_V.pcm shaftcs_V.pcg
```

When `BcGUI2` starts, each volume is numbered as shown in Fig. 60. If you are not sure where the numbers are, try rotating the model. If there are many small parts, it may be difficult to see them even when rotating the model. In such cases, switch the displayed volume by selecting "`View`" -> "`Select the volume to draw`" (Fig. 61).

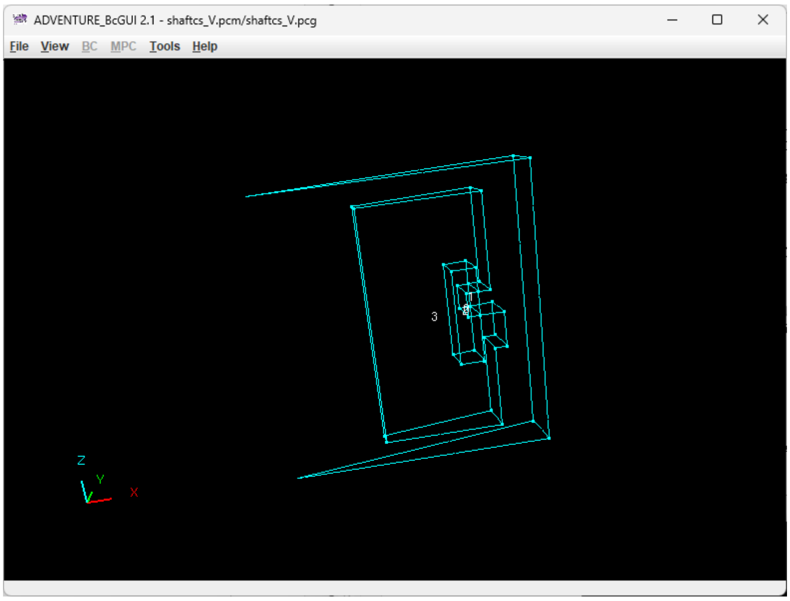


Fig. 60. Confirming the volume numbers.

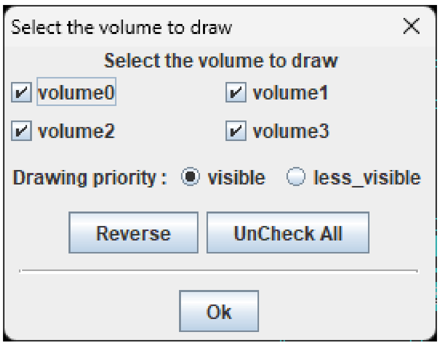


Fig. 61. Selecting the volume to draw.

The volume number and the correspondence of each region are as follows.

0 : Coil region (other than the area surrounded by red in Fig. 52)
1 : Coil region (area surrounded by red in Fig. 52)
2 : Annular magnetic body region
3 : Air region

Since the coil region is divided into two, these are combined and the material ID is set to 0. The remaining two regions are assigned the numbers 1 and 2, respectively.

"`shaft.dat`" will look like Fig. 62. "`#materialInfo`" is the part related to the function of setting physical quantities and is not used in ADVENTURE_Magnetic. Therefore, "`propertyN`", which is the number of types of physical quantities, is "`0`". "`#volumeInfo`" is the part related to the function of converting volume numbers to material IDs. "`volumeN`" is the number of volumes the mesh has. Below this, the material IDs to be assigned to each volume are listed for the number of volumes. The first two are combined into one as coils, so both material ID are "`0`". The latter two region a annular magnetic body region and an air region, so "`1`" and "`2`" are given them, respectively. Note that there are three material IDs, so "`materialN`" in "`#materialInfo`" is set to "`3`".

```
#materialInfo
materialN 3
propertyN 0

#volumeInfo
volumeN 4
0
0
1
2
```

Fig. 62. Contents of "`shaft.dat`".

Now it's ready, create the FEA model file "`input.adv`" with `advmag2_makefem_Electromangetic`.

・ Creating an FEA model file

```
% advmag2_makefem_Electromangetic shaftcs.msh shaftcs_18.fgr shaft.cnd shaft.dat
input.adv
```

(6) Creating an analysis directory

After creating a file for specifying the physical quantities, store it in the analysis directory along with the FEA model file.

The main physical quantities are specified in the material data file. The default name of this file is "`mtrl.dat`". The excitation current density to be applied to the coil is specified in the shape definition file. The name of this file is arbitrary, but here it is called "`coil.dat`". A characteristic curve file is also required to consider the nonlinearity of the magnetic reluctivity. The name of this file is also arbitrary, but here it is called "`bh_curve`". As stated at the beginning of this section, the magnetic reluctivity is $1/(4\pi\times10^{-7}) \fallingdotseq 7.957747e+05$ [m/H] in the air and coil regions. The magnetic reluctivity of the annular magnetic body is the initial value for the non-linear analysis, but here 7.571e+02 [m/H] will be used. The magnetic reluctivity of the annular magnetic body will become a different value for each element as the non-linear analysis progresses. Additionally, the coil has a material ID of "`0`", and since a shape definition file is used, the file type is "`md`". The annular magnetic body for which a non-linear analysis is performed has a material ID of "`1`". These are entered into "`mtrl.dat`" (Fig. 63).

```
MagneticReluctivity 3
 0 7.957747e+05
 1 7.571e+02
 2 7.957747e+05

Coil 1
 0 md coil.dat

NonLinear 1
 1 bh_curve
```

Fig. 63. Contents of "`mtrl.dat`".

Next, create the shape definition file "`coil.dat`". Since the coil region is sector-shaped, use "`DoubleSectorialCylinder`". Set the region a little larger so that the nodes at the end faces of the coil region are not judged to be outside the region due to numerical error. The magnitude of the excitation current density flowing through the coil is 3×10^7 [A/m^2]. Enter these in "`coil.dat`" (Fig. 64).

```
DoubleSectorialCylinder
 0.0 0.0 0.035
 z 0.05
 deg -10.0 30.0
 0.07 0.09
 3.0e+07
```

Fig. 64. Contents of "`coil.dat`".

Create a characteristic curve file "`bh_curve`". The *B-H* characteristic curve in Fig. 51 is the initial magnetization curve for SS41P. This will be approximated with a broken line. First write the number of points, then enter the horizontal and vertical axis values as a set (Fig. 65).

Finally, create an analysis directory and a subdirectory to store the FEA model file, and move each file to the designated location. The analysis directory name will be "`shaft`".

```
% mkdir -p shaft/model_one
% mv input.adv shaft/model_one/
% mv mtrl.dat shaft/
% mv coil.dat shaft/
% mv bh_curve shaft/
```

```
31
0.00e+0 0.00
5.30e+2 0.70
5.70e+2 0.80
6.30e+2 0.90
7.00e+2 1.00
7.40e+2 1.05
7.90e+2 1.10
8.40e+2 1.15
9.00e+2 1.20
9.70e+2 1.25
1.05e+3 1.30
1.17e+3 1.35
1.32e+3 1.40
1.52e+3 1.45
1.78e+3 1.50
2.09e+3 1.55
2.50e+3 1.60
3.15e+3 1.65
3.80e+3 1.70
4.76e+3 1.75
6.10e+3 1.80
8.00e+3 1.85
1.03e+4 1.90
1.30e+4 1.95
1.65e+4 2.00
2.10e+4 2.05
2.60e+4 2.10
3.30e+4 2.15
4.20e+4 2.20
6.25e+4 2.25
9.90e+4 2.30
```

Fig. 65. Contents of "bh_curve".

(7) Domain decomposition

A hierarchically decomposed model is created using ADVENTURE_Metis based on the FEA model created so far. The option "-difn 1" should be specified when executing. This option is to specify the degrees of freedom on the internal boundary nodes as 1.

First, the numbers of parts and subdomains should be determined to decompose mesh hierarchically. Here, two PCs will be used to analyze in distributed-memory parallel mode. Therefore, the number of parts will be 2. The number of subdomains will be determined by formula (2) in section 2.2, so first check the number of elements $N_{element}$ in the FEA model file with advinfo.

```
% advinfo shaft/model_one/input.adv
```

The "num_items" in "content_type=Element" is the number of elements, which is 13,431 in this case. If the number of elements per subdomain, $n_{element}$, is 100, then

$$\frac{13{,}431}{2 \times 100} \cong 67.155\ . \tag{7}$$

The exact value obtained here does not have to be used. Any round number can be used. Here, the number of subdomains is set to 65.

```
% mpirun –np 2 adventure_metis –difn 1 shaft/model_one/input.adv shaft 65
```

Note that in shared-memory parallel mode and hybrid parallel mode, processing is assigned to threads on a subdomain basis, so if the number of subdomains is a multiple of the number of threads, there will be less waste of computational resources.

(8) Executing the analysis

Perform a non-linear magnetostatic analysis (Magnetostatic) using the electromagnetic field analysis module. The option "-data-dir" is required because the analysis directory name is different from the default. In addition, to perform a non-linear analysis, the option "-nl-method" is used to specify the non-linear analysis method. Here, the Newton's method is used.

```
% mpirun –np 2 advmag2_HDDM_Electromagnetic-p Magnetostatic -data-dir shaft -nl-method
Newton
```

(9) Creating a visualization file

Create a visualization file using advmag2_hddmmrg. Unknown "MagneticVectorPotentail", which is singular (integral constants are not determined) will be excluded. The name of the visualization file will be "shaft.vtu".

```
% advmag2_hddmmrg shaft.vtu -1 MagneticVectorPotentail -data-dir shaft
```

In addition, to check the magnetic reluctivity in the annular magnetic body taking into account nonlinearity, the option "-cut-part" is used to create a visualization file for only the annular magnetic body (material ID "1") and only the magnetic reluctivity. The option "-no-output" is also used to avoid overwriting the "shaft.vtu" created by the above command.

```
% advmag2_hddmmrg shaft.vtu 1 MagneticReluctivity -data-dir shaft -cut-part 1 -no-output
```

This creates "shaft_MatID1.vtu".

(10) Visualization

Here is an example of visualization using ParaView.

The ParaView settings after loading “`shaft.vtu`” and pressing “`Apply`” are as follows.

- Data type : “`Solid Color`” -> Check each figure.
- Model surface : “`Representation`” -> “`Wireframe`”
- Color bar type : “`Choose Preset`” -> “`Blue to Red Rainbow`”
- Vector display : “`Filters`” -> “`Common`” -> “`Glyph`”
  - Scale mode : “`Vector Scale Mode`” -> “`Scale by Magnitude`”
  - Scale factor : “`Scale Factor`” -> “`0.02`”

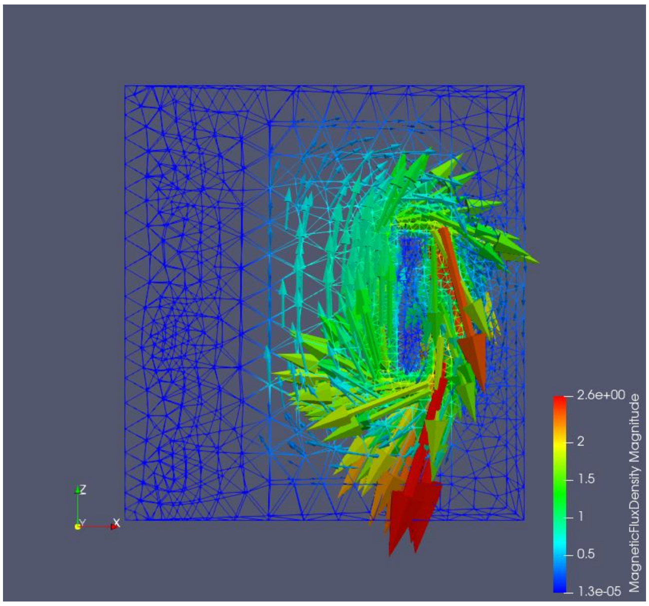


Fig. 66. Magnetic flux desity (`MagneticFluxDensity`).

Next, the ParaView settings after loading “`shaft_MatID1.vtu`” and pressing “`Apply`” are as follows.

- Data type : “`Solid Color`” -> Check each figure.
- Model surface : “`Representation`” -> “`Surface`”
- Color bar type : “`Choose Preset`” -> “`Blue to Red Rainbow`”

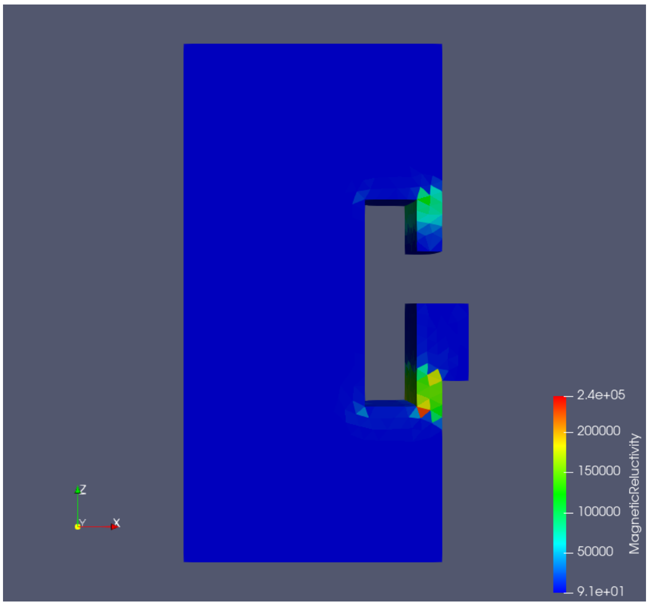


Fig. 67. Magnetic reluctivity (`MagneticReluctivity`).

The VTU files used to create Fig. 66 and Fig. 67, as well as the PNG files of these figures, are stored in

`sample_data/shaft/done/result/`

The PNG files can be saved from “`File`” -> “`Save Screenshot`”.

C. 3. Non-steady eddy current analysis

As an analysis example, the model used in "C.1 Standard model creation method: time-harmonic eddy current analysis" will be used. In addition, the real part of the results of the time-harmonic eddy current analysis will be used as the initial value for the non-steady analysis.

For reference, the file containing all the steps that follow can be found in

```
sample_data/ns_eddy/done/
```

The PNG file showing the visualization results using ParaView can also be found in

```
sample_data/ns_eddy/done/result/
```

(1) Data preparation

When using the results of time-harmonic eddy current analysis or non-linear magnetostatic analysis as the initial values, the same HDDM-type analysis model files are used. Also, the analysis result output settings file and analysis result output files are used as the initial values of non-steady analysis, settings file and initial values of non-steady analysis file for the non-steady analysis. To do this, rename the subdirectory that stores them.

```
% mv cake/result/ cake/initial
```

Next, create the material data file "`mtrl_ns.dat`" and shape definition file "`coil_ns.dat`" for the non-steady eddy current analysis. Since the name of the material data file is different from the default name, the option "`-mtrldat-file`" is required when executing the electromagnetic field analysis module. Here, copy and use the "`mtrl.dat`" and "`coil.dat`" used in the time-harmonic eddy current analysis.

```
% cp cake/mtrl.dat mtrl_ns.dat
% cp cake/coil.dat coil_ns.dat
```

In "`mtrl_ns.dat`", the angular frequency (`AngularFrequency`) that is not necessary for non-steady eddy current analysis is deleted, and the name of the shape definition file for is changed (Fig. **68**).

```
MagneticReluctivity 4
 0  7.957747e+05
 1  7.957747e+05
 2  7.957747e+05
 3  7.957747e+05

Coil 1
 2  md coil_ns.dat

ElectricalConductivity 1
 0 7.7e+06
```

Fig. 68. Contents of "`mtrl_ns.dat`".

```
DoubleSectorialCylinder
 0.0 0.0 -0.05
 z 0.2
 deg -10.0 40.0
 0.14 0.18
 50.0


TimeEvolution 1.0

TimeEvolutionSinusoidal
  Hz 60
  deg 90.0
  1.0 0.0
```

Fig. 69. Contents of "`coil_ns.dat`".

In "`coil_ns.dat`", the magnitude of the imaginary part of the forced current density is deleted, and the Time Change Definitions are added. Since the Time Change Definition is one sinusoidal wave, the end time of "`TimeEvolution`" can be any number greater than 0.0, but here it is set to "`1.0`". The frequency of the sinusoidal wave is set to 60 Hz (`Hz 60`), which corresponds to the angular frequency (`AngularFrequency`) used in the time-harmonic eddy current analysis. Also, shift the phase so that it matches the excitation current density in the real part of the time-harmonic eddy current analysis, which uses the analysis results as the initial value. The magnitude of the excitation current density in the real part is 50 [A/m^2], and the magnitude given here is also 50 [A/m^2], so set it so that the sinusoidal wave becomes 1 at 0 seconds (`deg 90.0`). The current of the sinusoidal wave 90° ahead ($\sin 180° = 0$) becomes 0, and it matches the magnitude of the imaginary part of the excitation current density in the time-harmonic eddy current analysis. The multiplier is 1, and the constant is 0 (`1.0 0.0`). Enter these in "coil_ns.dat" (Fig. 69).

These files are included in

`sample_data/ns_eddy/`

Finally, move these files to the analysis directory.

```
% mv mtrl_ns.dat cake/
% mv coil_ns.dat cake/
```

(2) Executing the analysis

A non-steady eddy current analysis (`NS_Eddy`) is performed using the electromagnetic field analysis module. The analysis directory name and material data file name are different from the defaults, so the options "`-data-dir`" and "`-mtrldat-file`" are required. In addition, "`-ns-inivalue-type`" is required to specify the type of initial value, "`-ns-delta-t`" to specify the time step size, and "`-ns-end-step`" to specify the final time step to be analyzed. Here, one period of a sinusoidal wave will be followed in 20 steps. Since 60 Hz is divided into 20 parts, the time step size $\Delta t = 1/(60 \times 20) \cong 8.333333e-04$ [s].

```
% mpirun -np 2 advmag2_HDDM_Electromagnetic-p NS_Eddy -data-dir cake -mtrldat-file mtrl_ns.dat -ns-inivalue-type real -ns-delta-t 8.333333e-04 -ns-end-step 20
```

(3) Creating a visualization file

Create a visualization file using `advmag2_hddmmrg`. Unknowns "`MagneticVectorPotentail`" and "`ElectromagneticScalarPotential`", which are singular (integral constants are not determined) will be excluded. The name of the visualization file will be "`cake_ns.vtu`".

```
% advmag2_hddmmrg cake_ns.vtu -2 MagneticVectorPotentail ElectromagneticScalarPotential -data-dir cake
```

In a non-steady analysis, files with the time step added to the file name, such as "`cake_ns_10.vtu`", are output for the number of time steps analyzed. Also, if the file name extension given to `advmag2_hddmmrg` is .vtu, a ParaView Data file (.pvd) listing the time of each time step will also be output. The name of this file will be the same as the file name given to `advmag2_hddmmrg` with the extension changed, such as "`cake_ns.pvd`".

(4) Visualization

Here is an example of visualization using ParaView.

The results of a non-steady analysis, which have consecutive numbers, will be displayed as "`cake_ns..vtu`" in the list in the window that appears when files are opened in ParaView, so if this is selected and loaded, the consecutive files will be loaded all at once. Also, if there is a ParaView Data file (.pvd), loading this will result in the same state as when consecutive files are loaded all at once. The ParaView settings after loading files and pressing "`Apply`" are almost the same as for a time-harmonic eddy current analysis, so here those that are unique to non-steady analyses are described.

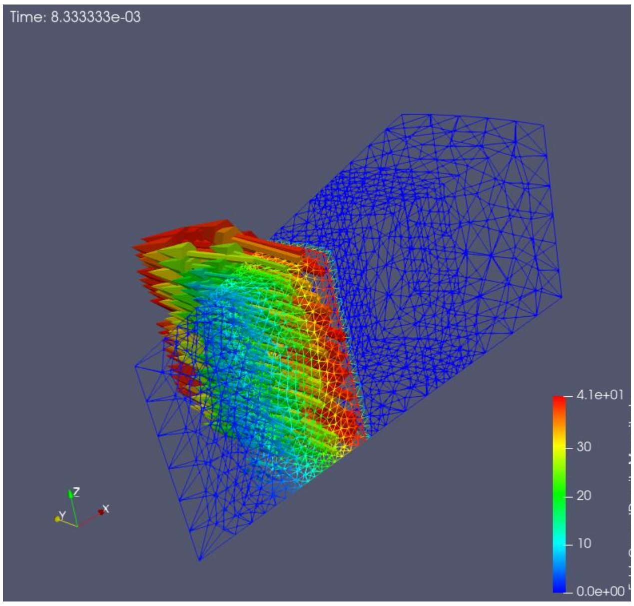
- Time display : The filter should be selected according to how the files are loaded.
  - ➢ Sequential file or PVD file : "`Filters`" -> "`Display`" -> "`Annotation`" -> "`Annotation Global Data`"
    - ✧ "`Prefix`" -> "`Time: `"
    - ✧ "`Format`" -> "`%8.6e`"
  - ➢ PVD file : "`Filters`" -> "`Display`" -> "`Annotation`" -> "`Annotation Time Filter`"
    - ✧ "`Format`" -> "`Time: {time:e}`"

Once settings have been finished, press the "`Play`" button to watch an animation of the changes.

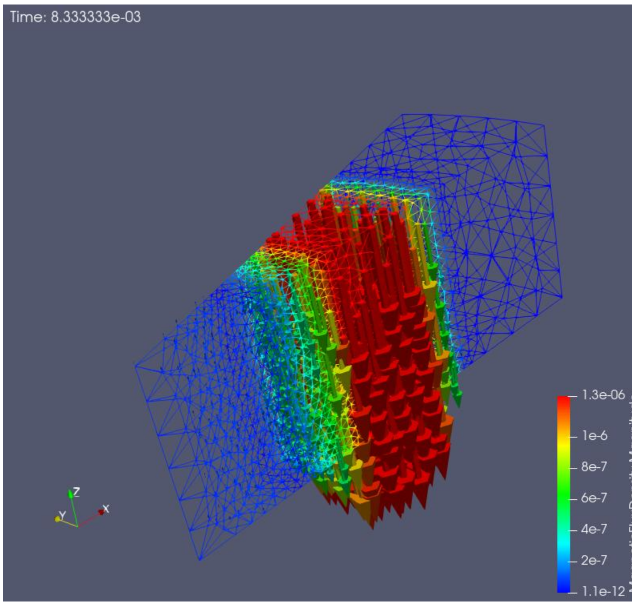
Fig. 70 shows the eddy current density and magnetic flux density at the 10th step (time 8.333333e-03 [s]). The VTU file used to create Fig. 70 and the PNG files for these figures are stored in

`sample_data/ns_eddy/done/result/`

The PNG files for each time step can be saved from "`File`" -> "`Save Screenshot`". Animations can also be saved from "`File`" -> "`Save Animation`". Change "`File of type`" appropriately before saving.



(a) Eddy current density
(`EddyCurrentDensity`, scale factor 0.002)



(b) Magnetic flux density
(`MagneticFluxDensity`, scale factor 50,000)

Fig. 70. 10th step (time 8.333333e-03 [s]) of non-steady analysis.

D. Differences from the previous version

In this chapter, the features added from the previous version are outlined.

D.1. Differences from Ver.1.9.2

- Not all the functions of Ver.1.9.2 have been inherited.
- The modules that were previously independent for magnetostatic analysis, time-harmonic eddy current analysis, and non-steady eddy current analysis have been integrated into one, and high-frequency electromagnetic field analysis functionality has also been added.
- There are four parallel modes: single mode, shared-memory parallel mode, distributed-memory parallel mode (previously static load distribution version), and hybrid parallel mode. The dynamic load distribution version has not been implemented.
- As new sparse matrix solvers, the BiCG method, BiCR method, and 20 kinds of product-type Krylov subspace methods [15] have been added.
- Boundary conditions can now be changed by simply loading a boundary condition file (.cnd) when executing a solver module, without redo creating an FEM model file and domain decomposition.
- Physical quantities are computed from unknowns within the electromagnetic field analysis module, including processing between interfaces. As a result, post-processing has changed from computing physical quantities and outputting visualization files after integrating domain decomposition data with `advmag_makeUCD` and `advmag_nodalforce` to integrating domain decomposition data and outputting visualization files with `advmag2_hddmmrg`.
- Due to the above, the tools have been revamped.
- It is now possible to insert comments into material data files and shape definition files.
- Many command options have been changed.

References

[1] ADVENTURE Project Home Page : https://adventure.sys.t.u-tokyo.ac.jp/

[2] Ryuji SHIOYA and Genki YAGAWA, Iterative domain decomposition FEM with preconditioning technique for large scale problem, *ECM'99 Progress in Experimental and Computational Mechanics in Engineering and Material Behaviour*, pp.255-260, 1999.

[3] Hiroshi KANAYAMA, Ryuji SHIOYA, Daisuke TAGAMI and Satoshi MATSUMOTO, 3-D eddy current computation for a transformer tank, *COMPEL*, Vol.21, No.4, pp.554-562, 2002.

[4] Hiroshi KANAYAMA and Shin-ichiro SUGIMOTO, Effectiveness of A-ϕ method in a parallel computation with an iterative domain decomposition method, *COMPUMAG2005*, 2005.

[5] Hiroshi KANAYAMA, Hongjie ZHENG and Natsuki MAENO, A domain decomposition method for large-scale 3-D nonlinear magnetostatic problems, *Theoretical an Applied Mechanics*, 52, pp.247-254, 2003.

[6] OpenMP: https://www.openmp.org/

[7] Message Passing Interface Forum: http://www.mpi-forum.org/

[8] Shin-ichiro SUGIMOTO, Amane TAKEI and Masao OGINO, Finite element analysis with tens of billions of degrees of freedom in a high-frequency electromagnetic field, *Mechanical Engineering Letters*, Vol.3, p.16-0067, 2017.

[9] Shin-ichiro SUGIMOTO, Amane TAKEI and Masao OGINO, High-frequency electromagnetic field analysis with 130 billion of degrees of freedom, *The 38th JSST Annual Conference, International Conference on Simulation Technology*, pp.290-295, 2019.

[10] MPICH: https://www.mpich.org/

[11] OpenMPI: https://www.open-mpi.org/

[12] Shin-ichiro SUGIMOTO, Hiroshi KANAYAMA, Shuuji ASAKAWA and Shinobu YOSHIMURA, "Time-harmonic eddy current analysis of 44 million complex DOF problem with hierarchical domain decomposition method", *Transactions of JSCES*, 2007027, 2007 (in Japanese).

[13] Kohji FUJIWARA and Takayoshi NAKATA, Results for benchmark problem 7 (asymmetric conductor with a hole), *COMPEL*, Vol.9, No.3, pp.137-154, 1990.

[14] Shin-ichiro SUGIMOTO, Masao OGINO, Hiroshi KANAYAMA and Shinobu YOSHIMURA, "Introduction of a Direct Method at Subdomains in Non-linear Magnetostatic Analysis with HDDM", *2010 International Conference on Broadband, Wireless Computing, Communication and Applications*, pp.304-309, 2010.

[15] Shin-ichiro SUGIMOTO, Tomohiro SOGABE, Shao-Liang ZHANG, Masao OGINO and Amane TAKEI, Product-type Krylov subspace methods for complex symmetric matrices in electromagnetic field analysis, *IEEJ Transactions on Industry Applications*, Vol.145, No.2, pp.87-97, 2025 (in Japanese).