ADVENTURE_Metis

Parallel domain decomposition of mesh into single-layered subdomains or double-layered parts and subdomains using METIS and ParMETIS as kernels

Version: 1.1

User's Manual

February 17, 2006

ADVENTURE Project

Contents

1. Summary

ADVENTURE_Metis is a domain decomposition tool (*Domain Decomposer*), which uses the Hierarchical Domain Decomposition method (HDDM) for parallel processing finite element analysis (FEA) system (ADVENTURE system) developed in the ADVENTURE project [1]. The Domain Decomposer creates groups of elements, which consist of the original finite element mesh as if those groups are to be fragments of the original mesh. ADVENTURE_Metis has the following features.

1. According to the Hierarchical Domain Decomposition method (HDDM), the original finite element mesh is decomposed into two level hierarchy of several parts each of which consists of subdomains for FEA with parallel data processing.

2. The decomposition libraries developed at Minnesota University [2] METIS and ParMETIS are used for decomposition of FE meshes into subdomains. These graph-partitioning libraries are supplied in the current ADVENTURE_Metis package.

3. The FEA mesh of the entire-type design model is used as input data for ADVENTURE_Metis, which produces HDDM-type decomposed mesh data of analysis model by the Hierarchical Domain Decomposition method (HDDM). The ADVENTURE file format is accepted for input/output data format.

4. The Message Passing Interface (MPI)[3] is used for parallel data processing. It is essential for compilation and execution of the ADVENTURE_Metis module.

5. The HDDM-type mesh data of analysis model produced by ADVENTURE_Metis can be used with the solvers released in the ADVENTURE project.

The information about the role of ADVENTURE_Metis in the ADVENTURE system can be found in the documents supplied with the solver for solid structure analysis ADVENTURE_Solid and other solvers released in the ADVENTURE project. Additional information can be also found in References [4] and [5].

2. File Input/Output

The ADVENTURE_Metis module uses the entire-type FEA model as input data to produce the HDDM-type FEA model data. The data processing flow is shown in Fig. 1.

1). The entire-type FEA model.
ADVENTURE_Metis treats the entire-type FEA model data in the binary ADVENTURE format, which was adopted as a multipurpose file format supported in the ADVENTURE project to represent FEA data. The input file is specified by arguments (required option). Detailed information can be found in [5]. The creation method of FEA model for analysis using the ADVENTURE system is discussed in documents supplied with the ADVENTURE_BCtool module. Information about the entire-type FEA model, the ADVENTURE file format, and the usage of several important tool programs is provided in the documents supplied with ADVENTURE_Soild module.

2). HDDM-type FEA model.
ADVENTURE_Metis creates the HDDM-type FEA model files in the binary ADVENTURE format. ADVENTURE_Metis performs the FEA data decomposition twice, however the number of output files with names *advhddm_in_0.adv*, *advddm_in_1.adv*, etc., is equal to the number of decomposed *Parts* at the first step of decomposition (the names will be given to the output files by default). The number at the end of the filename corresponds to the *Part's* number decomposed at the first step of decomposition. Details are given in *Appendix B*.

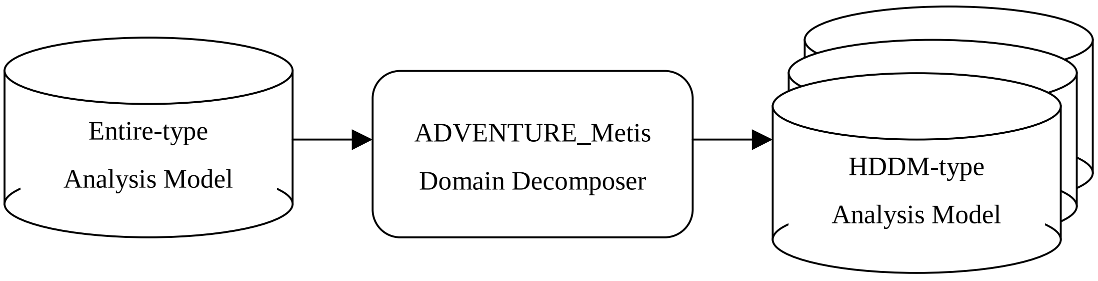


Fig. 1. Flow of Input and Output Data

3. *Program Installation and Compilation*

The ADVENTURE_Metis uses MPI libraries, which should be installed to use MPI libraries and their environments for further installation processes. In the environments without pre-installed MPI, a free software MPICH [6] can be used. Moreover, the ADVENTURE_IO library should be compiled prior to the installation of the ADVENTURE_Metis module.

To install the ADVENTURE_Metis module, execute the following commands from the ADVENTURE_Metis top directory:

1. **`./configure [`*options*`]`**
2. **`make`**
3. **`make install`**

The script **`configure`** is used to create the proper files **`Makefile`**. The following options can be used with the command **`configure`**.

- **`--prefix=`***install_dir*
  The option is used to set the target installation directory to *install_dir*. The default directory is `$HOME/ADVENTURE`.

- **`--with-advio=`***directory*
  The option is used to set a path to the ADVENTURE_IO library if it is installed in a directory other than the default directory *install_dir*.

- **`--with-mpicc=`***command*
  The option is used to set the name of MPI C compiler, which environments should be setup in advance. The compilation will not be accomplished without MPI C compiler. The default *command* option is **`mpicc`**.

- **`--with-mpi-cflags=`***CFLAGS*
  The option is used to set *flags* for the MPI C compiler. For example, if it is necessary to set the MPI *include* files, the option which can be used is **`--with-mpi-cflags=`**`"-I/usr/local/include/mpi"`. It can be also combined with the variable **`CFLAGS`**.

--with-mpi-libs=*LIBS*

The option is used to make program linking. For example, if you need to indicate which MPI library should be used, the option will be similar to **--with-mpi-libs=**”`-L/usr/local/lib/mpi –lmpi`”. It can be also combined with variable **LIBS**.

- **--enable-optimize**

  The option is used to make optimized compilation. If you are going to use additional options for optimization, this option can be associated with the option **--enable-optimize=***CFLAGS*, which sets *CFLAGS* for optimization.

The settings of MPI C compiler can be changed by the following environmental variables, which should be used prior to the command **configure**.

- **MPICC**

  Changes the name of MPI C compiler.
- **CFLAGS**

  Sets options for MPI C compiler.
- **LIBS**

  Sets libraries for linking.

For example, if *C shell* is used:

```
% setenv MPICC /usr/local/bin/mpicc
% setenv CFLAGS “-O2 -g -Wall”
% ./configure
```

If *Bourne shell* is used:

```
$ MPICC=/usr/local/bin/mpicc
$ export MPICC
$ CFLAGS=”-O2 -g -Wall”
$ export CFLAGS
$ ./configure
```

In most cases, the compilation processes are accomplished automatically. If it cannot be done automatically due to some reasons, the program can be compiled manually. To compile ADVENTURE_Metis using *Makefile*s, the sample files *Makefile.sample* are provided for reference in each directory of the ADVENTURE_Metis package. These sample files should be copied to *Makefile* and corrected in accordance with created environment. After that, the command **`make`** should be executed from the top directory.

If the compilation using the script **`configure`** is finished normally, execute the command **`make install`** from the top directory. Executable program modules and user manuals will be installed into the directory specified by the option *install_dir*. If the module is compiled without using **`configure`** script (using the copies of `Makefile.sample`), execute the command **`make install`** from the top directory. The target directory for installation of executable modules will be set according to the variable **`INSTALL_BINDIR`** specified in the file `Makefile.sample.in`. The default target directory for executable modules is `$HOME/ADVENTURE/bin`. The target installation directory for program's documentation will be set according to the variable **`INSTALL_DOCDIR`** of the file `Makefile.sample.in`. The default directory for program's documentation is `$HOME/ADVENTURE/doc/AdvMetis`.

4. *Program Handling*

4.1. Execution of Program Using MPI

The commands for program execution in parallel computing environments using MPI vary with implementations of MPI and used computing environments. In the case of MPICH, the programs can be executed in a parallel mode using the command `mpirun`. For example:

% mpirun [*mpioptions*] ***program_name*** [*options*]

In some other MPI packages, the MPI options can be passed to executable modules in a way, similar to:

% ***program_name*** [*mpioptions*] [*options*]

For more detailed explanations, refer to the MPI user's manual. A brief explanation about the program execution using MPICH (hosts specification, etc.) is given in *Appendix* of the user manual supplied with the ADVENTURE_Solid module. Depending on MPI environment, the execution commands for ADVENTURE_Metis can be:

% mpirun [*mpioptions*] **./adventure_metis** [*options*] *model_filename directory_name div_num*

or

% ./adventure_metis [*mpioptions*] [*options*] *model_filename directory_name div_num*

Model_filename, *directory_name*, and *div_num* are the necessary arguments of ADVENTURE_Metis, as described below.

4.2. Necessary Program Arguments

Execution of ADVENTURE_Metis requires three arguments, which must be always added to the command after other command options. These necessary arguments are *model_filename, directory_name*, and *div_num*.

- *model_filename*
  It is the argument to specify the filename of input entire-type FEA model data.

- *directory_name*
  It is the argument to specify the name of the top directory, where HDDM-type FEA model files will be saved. Here, a relative path should be given from the directory, where ADVENTURE_Metis will be executed. If the specified directory does not exist, ADVENTURE_Metis will automatically create this directory and the subdirectory named **`model`**. Consequently, the HDDM-type FEA model data will be saved in the subdirectory **`model`**.

- *div_num*
  It is the argument to specify the number of decomposed subdomains, which will be created after the second step of domain decomposition. ADVENTURE_Metis uses two-step hierarchical domain decomposition method (HDDM) to create subdomains of an entire-type FEA model. The number of launched ADVENTURE_Metis processes determines the number of domains after first step of decomposition (called herein the number of *Parts*). The number of second-step subdomains (called herein the number of *Subdomains*) is specified by the argument *div_num* for each *Part* created after the first-step decomposition. The following example shows a situation when 4 *Parts* will be decomposed into 400 *Subdomains* (each *Part* consists of 100 *Subdomains*).

```
% mpirun –np 4 ./adventure_metis some_model.adv some_model 100
```

  Here, the filename of entire-type FEA model is *some_model.adv* and the top directory for HDDM-type FEA model files is *some_model*. The method to determine an optimum number of *Subdomains* is described in the user manual of ADVENTURE_Solid module. One version of the command **`mpirun`** is used here as an example.

4.3. Options

- **`-HDDM`**
  This default option is used to create the domain-decomposed input data for the ADVENTURE_Solid solver using the Hierarchical Domain Decomposition method.

- **`-MAGNETIC`**
  This option is used to create the input domain-decomposed data for the ADVENTURE_Magnetic module.

- **`-difn`** *n*
  This option is used to create the input domain-decomposed data whose interface nodes have *n* degrees of freedom. Set 1 as *n* for ADVENTURE_Thermal. The default value is 3.

- **`-decomposed-file`** *filename*
  The option specifies the filename of the HDDM-type FEA model data. The filename can be specified without extension. The *Part* number, the file extension *.adv*, and underbars will be added automatically to the specified filename. The default filename is `advhddm_in`.

- **`-subdir_name`** *directory*
  The option specifies the name of subdirectory where HDDM-type FEA model data files will be created. The default name of subdirectory is `model`.

- **`-memlimit`** *n*
  The option specifies the size of memory in Mbytes, which can be used by each process. The program terminates when the value of total dynamically allocated memory is exceeded. The default value is unlimited.

- **`-ls`** *filename*
  The option specifies the name *filename* of the output log file created during operation by each processor. By default, the log will be printed to the standard error output of the start processor.

- **`-ll`** *n*

The *Log Level* can be specified to control the degree of processing details at a required time. Default value for *n* is **`2`**.

`-ll 0` : No logs are saved
`-ll 1` : The log is saved only for each routine
`-ll 2` : The detailed log is saved for each routine and subroutine
`-ll 3` : The situation of the main loop is displayed in real time.

- **`-nin`** *n*
  The option specifies a number of read processes of the input files. The default value is **`1`**.

- **`-r`**, **`-nr`**
  The option **`–r`** specifies a process of optimized renumbering. The option **`–nr`** terminates the process of optimized renumbering. The default option is **`–r`**.

- **`-h`**, **`-help`**
  Both options print display help information.

5. Helpful Hints

5.1. Disk Sharing

If large models are decomposed using a group of processors, an intensive access to a hard disk drive shared by NFS can slow down the file output processes since the file output processes in ADVENTURE_Metis are done simultaneously. One way to overcome this problem is to make the output directory symbolically linked with a local hard disk or assign the local hard disk to the output directory.

5.2. METIS Library and Decomposed FEA Models

Depending on execution environments of ADVENTURE_Metis, the decomposition done by METIS has several patterns that can be traced in the second-step domain decomposition.　Such phenomenon does not occur with ParMETIS.

5.3. Optimized Renumbering

The optimized renumbering is performed for each *Subdomain* by default.　This process may take much time if each *Subdomain* consists of a large number of nodes. The optimized renumbering can be cancelled by adding the option **`–nr`**.

Appendix

A. Operational Principles

The SIMD-type parallel execution process is employed as a basic algorithm in ADVENTURE_Metis. The parallel-distributed program execution is achieved by giving the same synchronized instructions to all processes, and only the startup process runs individually as a *Master* process. The *Master* process reads input data, decomposes the data in accordance with a number of processes, and distributes the corresponded data to CPUs sequentially. Such scheme is implemented to prevent a simultaneous access by all processes to a single input data file. The sequence of ADVENTURE_Metis operations is briefly described below. The term *Part* is used for the FEA model created by the first-step domain decomposition and the term *Subdomain* is used for the FEA model after the second-step domain decomposition.

1. Read and initialize command options.

2. File opening

In order to detect situations when abnormal program termination can cause errors (for example, when the results cannot be saved to a file), the input and output files are opened prior to start the calculations.

3. Pre-filter

The element's connectivity data are read from the FEM input file and converted into graphs.

4. Partition-1

ParMETIS performs the first-step decomposition.

5. Converter

Based on results of *Partition-1*, the element's data are redistributed and the *Part*'s graph data are corrected individually.

6. Partition-2

METIS performs the second-step decomposition of each *Part*'s graph data.

7. Post-Filter

a). The Node's data and the Boundary condition's data are read from the input file and modified according to the results of *Partition-1* process.

b). Node's numbers are changed to the *Part*'s peculiar numbers and the inner boundary conditions are set between *Parts* and between *Subdomain*s.

c). The *Part*'s data are saved to the output files.

d). The *Subdomain*’s data are saved to the output files.
e). Termination of MPI, file closing, and other program’s termination operations are performed.

B. Output File Format: Data Structure of HDDM-type FEA Model File

B.1. Domain-decomposed Data in ADVENTURE System

ADVENTURE_Metis uses the HDDM-type FEA data for the file format (which is defined as *advhddm_in* file by default). The data flow is shown in Fig. 2.

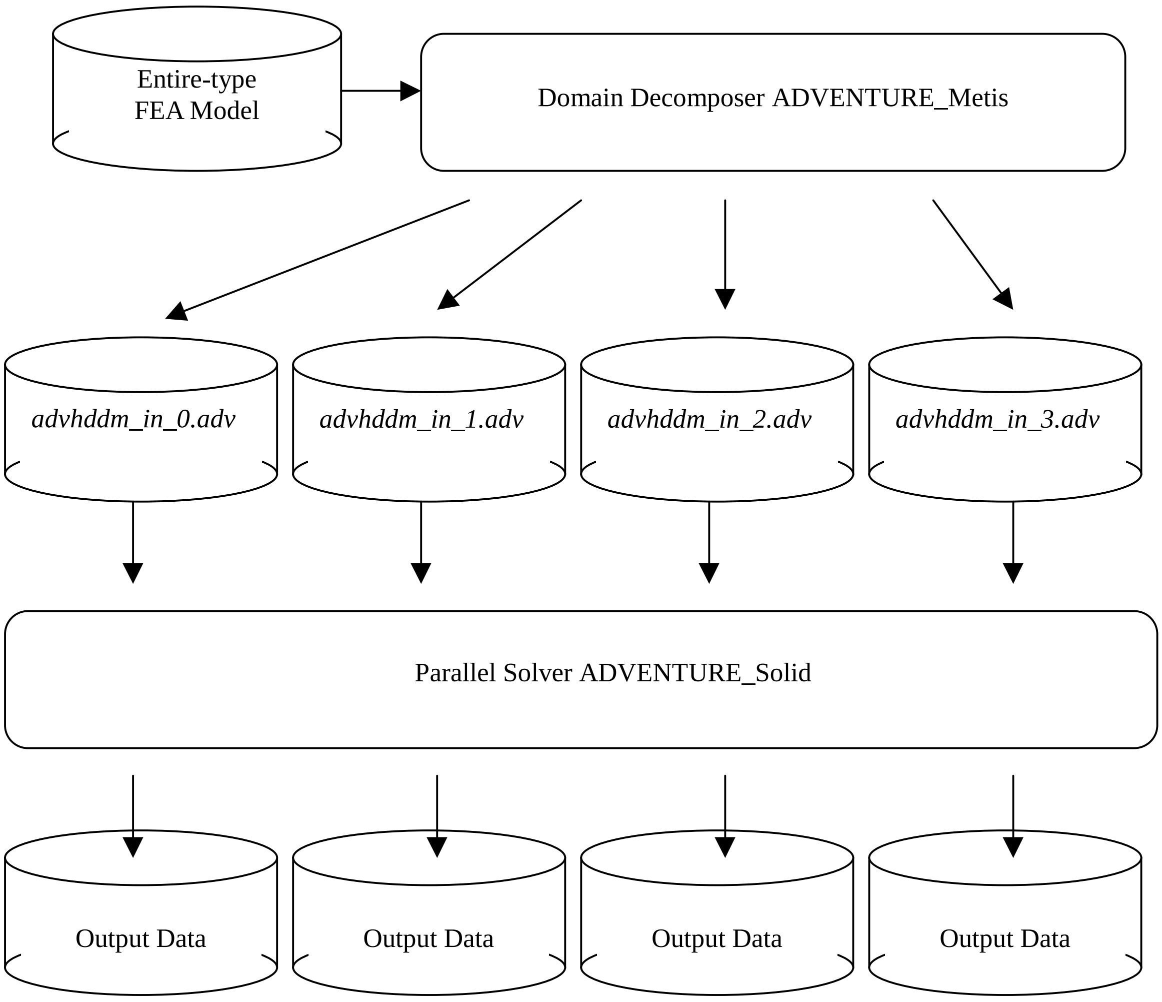


Fig. 2. Input and Output Data Flow used by ADVENTURE_Metis

B.2. Structure of HDDM-type FEA Model Data File

ADVENTURE_Metis performs two-step domain decomposition to create input files that can be used by the FEA solver module ADVENTURE_Solid, which uses the Hierarchical Domain Decomposition method (HDDM). An entire-type FEA model is divided into *Parts* at the first step of decomposition, and the *Parts* are subdivided into *Subdomains* at the second step of decomposition. The information on each *Part* and *Subdomain* is contained in the corresponded files *advhddm_in* which number is equal to the number of *Parts*. The *advhddm_in* files are prepared in accordance with the rules called ADVENTURE file format, where the information unit called *Document* is used for representation of the *Subdomain*'s data. The types of *Documents* used in the HDDM-type FEA model are listed in the following table.

Type of *Document*	*content_type**
Coordinates of the node	`Node`
Index from the node's coordinates of *Part* to the node's coordinates in the *Global* system	`FEGenericAttribute`
Information on the degrees of freedom of inner boundaries of the connected *Parts*	`HDDM_InterfaceDOF`
Connectivity of the elements	`HDDM_Element`
Index from the node's number in the *Subdomain*'s system to the node's number in the *Part*'s system	`HDDM_FEGenericAttribute`
Index from the node's number in the *Subdomain*'s system to the node's number in the *Global* system	`HDDM_FEGenericAttribute`
Inner boundary condition number	`HDDM_FEGenericAttribute`
Displacement boundary conditions	`HDDM_FEGenericAttribute`
Other information (constants, material properties, etc.)	`HDDM_FEGenericAttribute`

* "*content_type*" and *Property* are the terms used for the ADVENTURE file format [5].

The contents of *Document* are described below. "*infree*", "*midfree*", "*outfree*", "*opnu*", and "*opbd*" are the terminology related to the degree-of-freedom (DOF) of inner boundaries that will be explained in *Chapter B.3* of the current Manual.

< Structure of HDDM-type FEA model data >

```
##########################################################
  Node's coordinates (in order of node's numbers of Part)
##########################################################
[Properties]
   str <content_type> = "Node"
   int32 <num_items> = The number of nodes in the considered decomposed Part
   int32 <num_items_orig> = The number of nodes in the entire-type FEA model before decomposition
   int32 <dimension> = The dimension(s)
[Data]
   ((float64  The coordinate(s)) * The dimension(s))   * The number of nodes
----------------------------------------------------------------
##########################################################
Index from the node's number of Part to the node's number in the Global system
##########################################################
[Properties]
   str <content_type> = "FEGenericAttribute"
   int32 <num_items> = The number of nodes in the considered decomposed Part
   int32 <num_items_orig> = The number of nodes in the entire-type FEA model before decomposition
   str <fega_type> = AllNodeVariable
   str <format> = i4
   str <label> = NodeIndex_PartToGlobal
   int32 <index_byte> = 4

[Data]
   (int32  The node's number in the Global system)   * The number of nodes
----------------------------------------------------------------
##########################################################
Index from the element's number of Part to the element's number in the Global system
##########################################################
 None.
----------------------------------------------------------------
##########################################################
Information on the DOF of inner boundaries of connected Parts
##########################################################
[Properties]
   str <content_type> = "HDDM_InterfaceDOF"
   int32 <num_parts> = The total number of Parts
   int32 <dimension> = The DOF of inner boundary nodes
   int32 <num_subdomains> = The number of Subdomains in one Part
   int32 <part_number> = The identification number of the considered Part
   int32 <num_outdom> =  n_outdom
   int32 <total_num_infree> =  t_infree
   int32 <total_num_outfree> =  t_outfree
   int32 <total_num_midfree> =  t_midfree

[Data]
   (  ( int32 <op.n_mofree> )    The number of op.ibid in the considered Part
     ((int32 <op.ibid> ) x op.n_mofree)
   ) x   num_parts
----------------------------------------------------------------
##########################################################
Element's connectivity
##########################################################
[Properties]
   str <content_type> = "HDDM_Element"
   int32 <num_subdomains> = The number of Subdomains in the considered Part
   str <element_type> =  The type of element
   int32 <num_nodes_per_element> = The number of nodes per one element
   int32 <element_dimension> = The dimension of the element
   int32 <element_num_items> = The number of elements in the considered Part

[Data]
   (The number of elements in Subdomain
     (
       (int32  The node's number  ) x  The number of nodes per one element
     ) x  The number of elements in Subdomain
   ) x  The number of Subdomains
----------------------------------------------------------------
##########################################################
Index from the node's number of Subdomain to the node's number in the Part system
##########################################################
[Properties]
   str <content_type> = "HDDM_FEGenericAttribute"
   int32 <num_subdomains> = The number of Subdomains in the considered Part
   int32 <sum_items> = The total number of nodes in the considered Part
   str <fega_type> = AllNodeVariable
   str <format> = i4
   str <label> = NodeIndex_SubdomainToPart
   int32 <index_byte> = 4
```

```
        [Data]
            (The number of nodes in Subdomain
          ( int32   The node's number in the Part system )   x   The number of nodes in Subdomain
     ) x   The number of Subdomains
----------------------------------------------------------------

##########################################################
Index from the node's number of Subdomain to the node's number in the Global system
##########################################################
[Properties]
    str <content_type> = "HDDM_FEGenericAttribute"
    int32 <num_subdomains> = The number of Subdomains in the considered Part
    int32 <sum_items> = The total number of elements in the considered Part
    str <fega_type> = AllElementVariable
    str <format> = i4
    str <label> = ElementIndex_SubdomainToGlobal
    int32 <index_byte> = 4
[Data]
    (The number of elements in Subdomain
          ( int32 The element's number in the Global system )   x   The number of elements in Subdomain
    ) x   The number of Subdomains
----------------------------------------------------------------
##########################################################
The inner boundary condition's number
##########################################################
[Properties]
    str <content_type> = "HDDM_FEGenericAttribute"
    int32 <num_subdomains> = The number of Subdomains in the considered Part
    int32 <sum_items> = The total number of inner boundary conditions in the considered Part
    str <fega_type> = NodeVariable
    str <format> = i4i4i4i4
    str <label> = InterfaceDOF
    int32 <ifd_dimension> = 3
    int32 <index_byte> = 4
[Data]
    ( The number of inner boundary conditions in Subdomain
          ( index The node's identification number in Subdomain's system
            int32 <coordinate>
            int32 The identification number of the corresponding Part
            int32 The identification number of the inner boundary condition of the corresponding Part
            int32 The identification number of the inner boundary condition of the Part which contains the considered Subdomain
          )   x   The number of inner boundary conditions of the considered Subdomain
    )   x   The number of Subdomains
----------------------------------------------------------------
##########################################################
Displacement boundary conditions
##########################################################
[Properties]
    str <content_type> = "HDDM_FEGenericAttribute"
    int32 <num_subdomains> = The number of Subdomains in the considered Part
    int32 <num_items_orig> = The total num_items before decomposition
    int32 <sum_items> = The total number of "items" in the considered Part
    str <fega_type> = NodeVariable
    str <format> = i4f8
    str <label> = ForcedDisplacement
    int32 <index_byte> = 4

[Data]
    ( The number of "items" in Subdomain
          ( index The identification number of the node in Subdomain
            int32 <coordinate>
            float64   Value
          )   x   The number of items in Subdomain
    ) x   The number of Subdomains
----------------------------------------------------------------
##########################################################
Common format of HDDM_FEGenericAttribute
##########################################################
[Properties]
    str <content_type> = "HDDM_FEGenericAttribute"
    int32 <num_subdomains> = The number of Subdomains in the considered Part
    int32 <num_items_orig> = The total num_items before decomposition
    str <fega_type> = (All)(Node|Element)(Constant|Variable)
    str <format> = Format of <value>
    str <label> = Any label
    int32 <index_byte> = 4

[Data]
All(Node|Element)Variable
    (  The number of Subdomains in the considered Part
       <value>  x   The number of "items" in Subdomain
```

```
              ) x  The number of *Subdomains*

All(Node|Element)Constant
    <value>

(Node|Element)Variable
    (The number of items in the considered *Subdomain*
      ( <index>, <value> ) x   The number of “items” in the considered *Subdomain*
    ) x  The number of *Subdomains*

(Node|Element)Constant
    <value>
    (  The number of “items” in the considered *Subdomain*
      <index> x   The number of items in the considered *Subdomain*
    ) x  The number of *Subdomains*
----------------------------------------------------------------
```

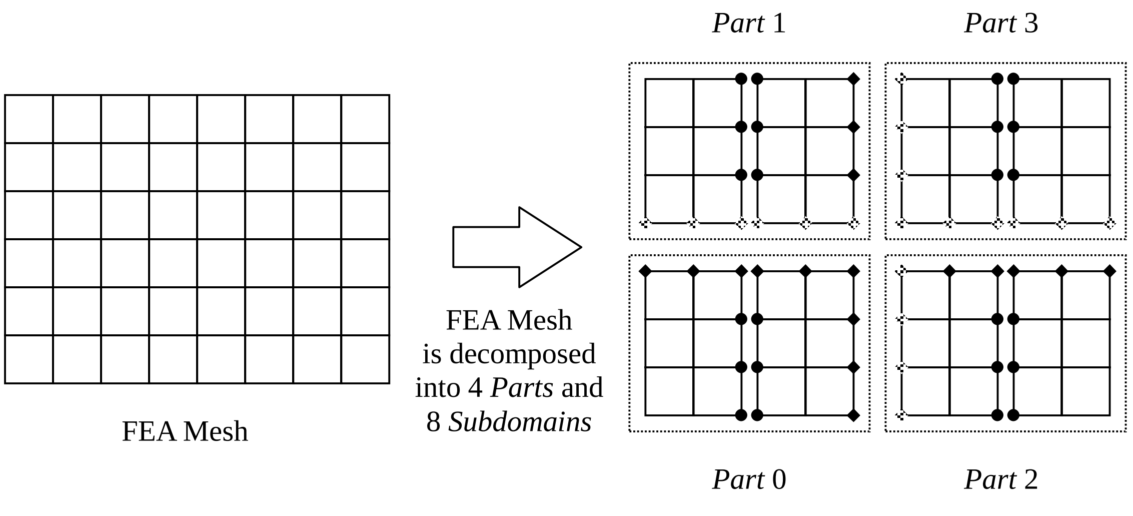
B.3. Example of HDDM-type FEA Model Data File

B.3.1. Inner Boundaries

As it was mentioned previously, in the Hierarchical Domain Decomposition method, the FEA mesh is decomposed into a number of subdomains. Each of subdomains should has information about connectivity to its adjacent subdomains because the object is originally continuous. In contrast to the Dirichlet and Neumann outer boundary conditions, which are set to the original FEA model, the HDDM method uses new boundary conditions so-called “Inner Boundary Conditions”.

B.3.2. Selection of “Parent” Responsible for Inner Boundaries

It is defined that the shared inner boundaries between the *Parts* are treated by the *Part* with smaller number. The example is shown in *Fig. 3*. The FEA mesh is decomposed into 4 *Parts* and 8 *Subdomains*. The DOF of shared boundary nodes of the *Part* 0 and *Part* 1 are treated by *Part* 0. The DOF of shared boundary nodes between *Part* 1 and *Part* 3 are treated by *Part* 1.



● Inner boundaries of *Subdomains* owned by the current *Part*

◆ Inner boundaries between neighbor *Parts* owned by the current *Part*

◇ Inner boundaries between neighbor *Parts* owned by a neighbor *Part*

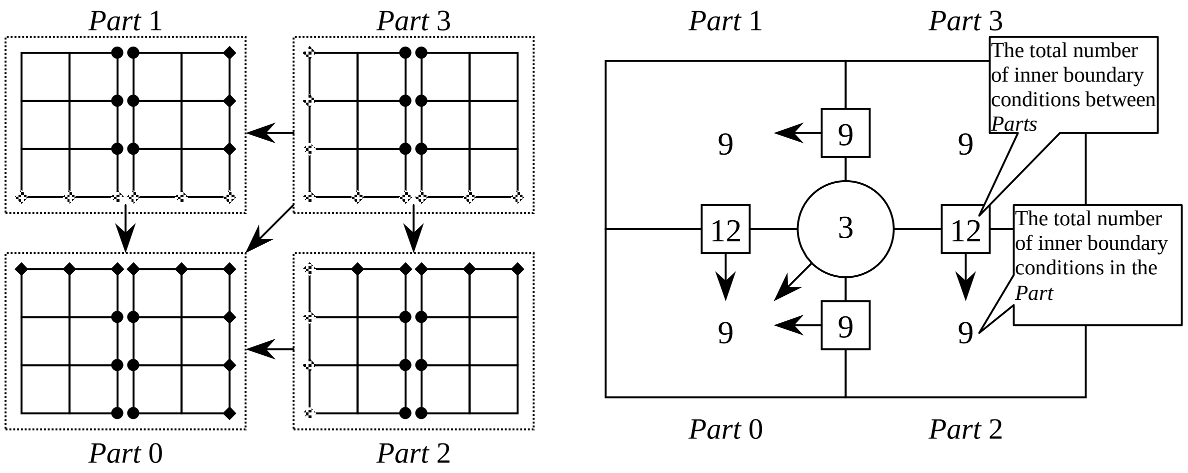
Fig. 3. Treatment of inner boundary conditions in HDDM

B.3.3. Data Related to Inner Boundaries

The following tables show the treatment of the inner boundaries in the data file *advhddm_in*.

Variable:	`t_infree`
Type of variable:	`int`
Explanation:	The total number of inner boundary conditions related to the current *Part*
Variable:	`t_outfree`
Type of variable:	`int`
Explanation:	The total number of DOF of inner boundary conditions of the current *Part* treated by the other *Parts*.
Variable:	`t_midfree`
Type of variable:	`int`
Explanation:	The total number of DOF of inner boundary conditions of the other *Part* treated by the current *Part*.
Variable:	`sum_ninbd`
Type of variable:	`int`
Explanation:	The total number (with overlapping) of DOF of inner boundary conditions of the *Subdomain* in the *Part*.
Variable:	`op[j].n_mofree`
Type of variable:	`int`
Explanation:	The total number of DOF of inner boundary conditions shared with *Part* `j`. The total number of DOF of inner boundary conditions shared within one *Part* is 0.
Variable:	`op[j].ibib[k]`
Type of variable:	`int`
Explanation:	The number of DOF of inner boundary conditions shared with *Part* `j`. This numbers are the serial numbers (`n_mofree` in total) within the *Part*, which are assigned independently for each *Part*.

The names and the types of variables are the same as read by the ADVENTURE solvers. Explanations about the counting of `t_infree`, `t_outfree`, `t_midfree`, and `op.n_mofree` are presented in the figures below.



Part 1
Part 3
Part 0
Part 2
Part 1
Part 3
9
9
9
9
12
3
12
9
9
The total number of inner boundary conditions between Parts
The total number of inner boundary conditions in the Part
Part 0
Part 2

● Inner boundaries of *Subdomains* owned by the current *Part*

◆ Inner boundaries between neighbor *Parts* owned by the current *Part*

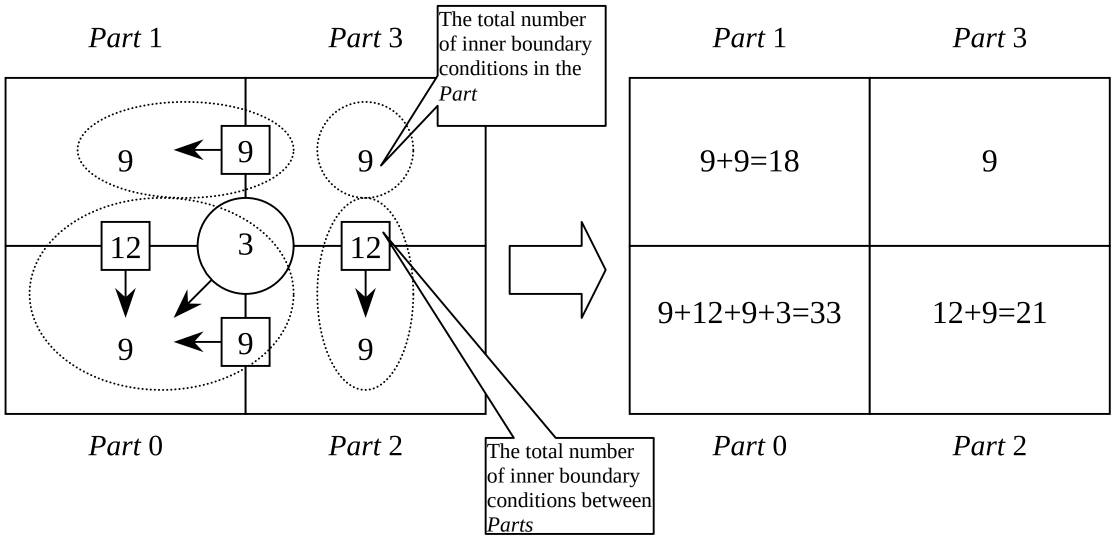
◇ Inner boundaries between neighbor *Parts* owned by a neighbor *Part*

⟵ The arrows show the *Part* responsible for the inner boundary conditions

Fig. 4. Counting of inner boundary conditions

Fig. 4 shows the way of counting of DOF of inner boundary conditions.

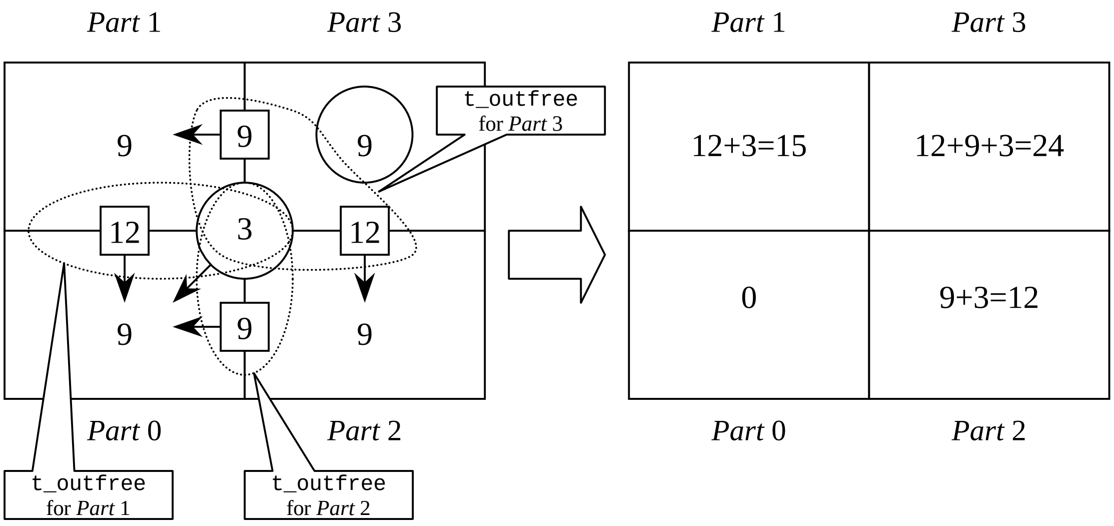
1. There are three nodes in each *Part* with inner boundary conditions. For a model without restraints, the total number of DOF of inner boundary conditions will be 9 (3 x 3 = 9).
2. For two or more *Parts* with shared inner boundaries without restraints, the DOFs will be: [*number of nodes*] x 3.
3. For the node, which is shared by three *Parts*, the total number of the DOFs will be 3. This node will be treated independently when counting `t_infree` and `t_outfree`.



Part 1
Part 3
The total number of inner boundary conditions in the *Part*
9
9
9
9
12
3
12
9
9
9
Part 0
Part 2
The total number of inner boundary conditions between *Parts*
Part 1
Part 3
9+9=18
9
9+12+9+3=33
12+9=21
Part 0
Part 2

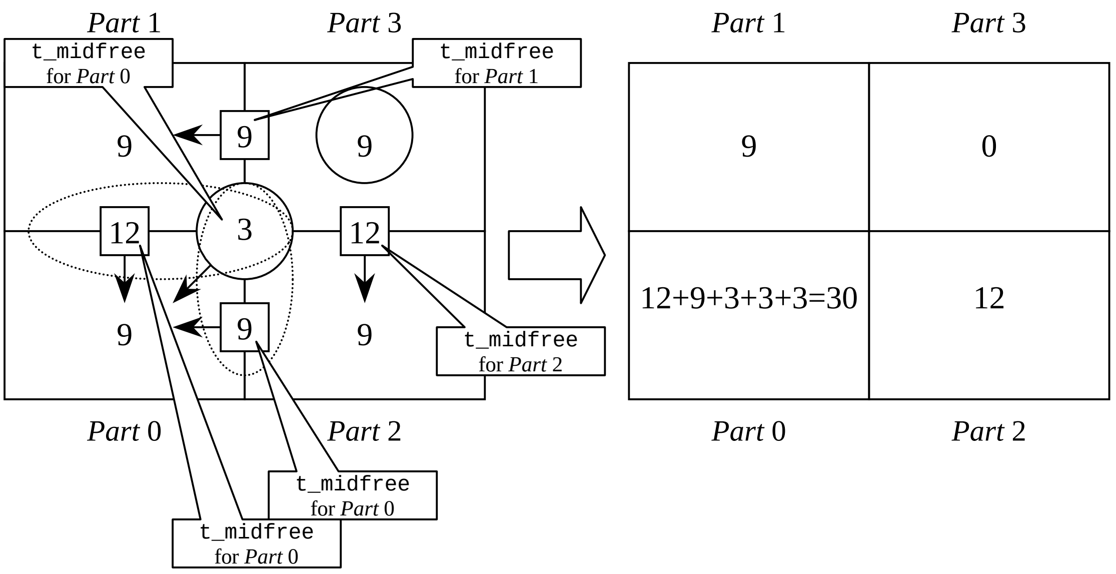
`t_infree`: the total number of DOF of inner boundary conditions (including DOF, which are not shared with other *Parts*).

Fig. 5. Counting of `t_infree`



`t_outfree`: the total number of DOF of inner boundary conditions treated by the neighbor *Part*.

Fig. 6. Counting of `t_outfree`



`t_midfree`: the total number of DOF of inner boundary conditions of the other *Part(s)* treated by the current *Part*.

Fig. 7. Counting of `t_midfree`

`t_infree`: the total number of DOF of inner boundary conditions (including the items, which are not shared with other *Parts*).

9+9	9
9+12+9+3	12+9

=

18	9
33	21

`t_outfree`: the total number of DOF of inner boundary conditions treated by a neighbor *Part*.

12+3	12+9+3
0	9+3

=

15	24
0	12

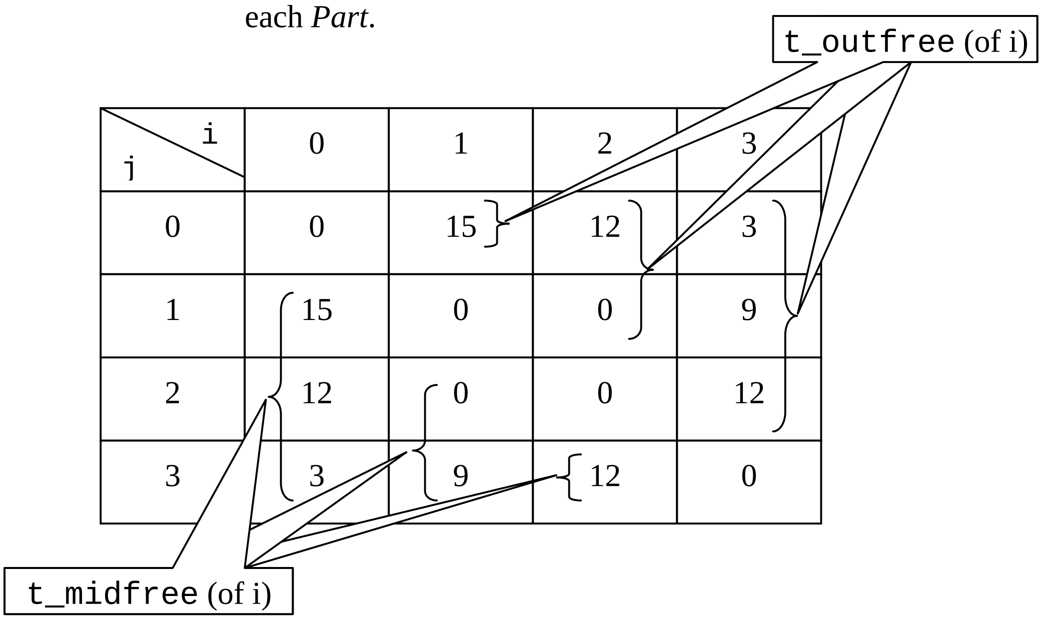
`t_midfree`: the total number of DOF of inner boundary conditions of a neighbor *Part* treated by the current *Part*.

9	0
12+9+3+3+3	12

=

9	0
30	12

`op[j].n_mofree`: The number of DOF of inner boundary conditions shared with *Part* `j`. This numbers are the serial numbers (`n_mofree` in total) within the *Parts*, which are assigned independently for each *Part*.



j \ i	0	1	2	3
0	0	15	12	3
1	15	0	0	9
2	12	0	0	12
3	3	9	12	0

Fig. 8. Counting of `m_nofree`

B.4. Contents of HDDM-type FEA Model Data File

The contents of HDDM-type FEA model data are presented below as an example. A cube (125 nodes, 64 elements) is subdivided into 2 *Parts*. Each *Part* is subsequently subdivided into 2 *Subdomains*. In this case, two binary files *advhddm_in_0.adv* and *advhddm_in_1.adv* are created in ADVENTURE format. Each file *advhddm_in* contains information about 1 *Part* and 2 *Subdomains*. These files can be found in the subdirectory *sample_data* of the ADVENTURE_Solid module.

The following text shows the contents of one of the data file *advhddm_in* processed by the program **`advshow`** supplied with the ADVENTURE_Solid as a tool for ADVENTURE files browsing.

```
[Document]
/export/work/adventure/AdvSolid-0.8beta/sample_data/cube_p2d2/model/advhddm_in_0.adv?A8138FE:Node@HDDM_Part[0]:5A85:39D89249
    size: 1800

[Properties]
  1: content_type=Node
  2: num_items=75
  3: num_items_orig=125
  4: dimension=3

[Data]
  0: 0.000000e+00 0.000000e+00 0.000000e+00
  1: 2.500000e+00 0.000000e+00 0.000000e+00
  2: 5.000000e+00 0.000000e+00 0.000000e+00
  3: 0.000000e+00 2.500000e+00 0.000000e+00
  4: 2.500000e+00 2.500000e+00 0.000000e+00
  5: 5.000000e+00 2.500000e+00 0.000000e+00
  6: 0.000000e+00 5.000000e+00 0.000000e+00
  7: 2.500000e+00 5.000000e+00 0.000000e+00
  8: 5.000000e+00 5.000000e+00 0.000000e+00
  9: 0.000000e+00 7.500000e+00 0.000000e+00
  10: 2.500000e+00 7.500000e+00 0.000000e+00
  11: 5.000000e+00 7.500000e+00 0.000000e+00
  12: 0.000000e+00 1.000000e+01 0.000000e+00
  13: 2.500000e+00 1.000000e+01 0.000000e+00
```

(Omitted)

```
  64: 2.500000e+00 2.500000e+00 1.000000e+01
  65: 5.000000e+00 2.500000e+00 1.000000e+01
  66: 0.000000e+00 5.000000e+00 1.000000e+01
  67: 2.500000e+00 5.000000e+00 1.000000e+01
  68: 5.000000e+00 5.000000e+00 1.000000e+01
  69: 0.000000e+00 7.500000e+00 1.000000e+01
  70: 2.500000e+00 7.500000e+00 1.000000e+01
  71: 5.000000e+00 7.500000e+00 1.000000e+01
  72: 0.000000e+00 1.000000e+01 1.000000e+01
  73: 2.500000e+00 1.000000e+01 1.000000e+01
  74: 5.000000e+00 1.000000e+01 1.000000e+01


[Document]
/export/work/adventure/AdvSolid-0.8beta/sample_data/cube_p2d2/model/advhddm_in_0.adv?1C2655B3:NodeIndex_PartToGlobal@HDDM_Part[0]:5A85:39D89249
    size: 300

[Properties]
  1: content_type=FEGenericAttribute
  2: num_items=75
  3: num_items_orig=125
  4: fega_type=AllNodeVariable
  5: format=i4
  6: label=NodeIndex_PartToGlobal
  7: index_byte=4

[Data]
  0: 0
  1: 1
  2: 2
  3: 5
  4: 6
  5: 7
  6: 10
```

```
           7: 11
           8: 12
  9: 15
```

(Omitted)

```
  71: 117
  72: 120
  73: 121
  74: 122

[Document]
/export/work/adventure/AdvSolid-0.8beta/sample_data/cube_p2d2/model/advhddm_in_0.adv?56A68AB7:HDDM_InterfaceDOF@HDDM_Part[0]:5A85:39D89249
    size: 308

[Properties]
  1: content_type=HDDM_InterfaceDOF
  2: num_parts=2
  3: dimension=3
  4: num_subdomains=2
  5: part_number=0
  6: num_outdom=4
  7: total_num_infree=105
  8: total_num_outfree=0
  9: total_num_midfree=75

[Data]
# Part[0]
#    num items in part
    0
# Part[1]
#    num items in part
    75
  0: 0
  1: 1
  2: 2
  3: 3
  4: 4
  5: 5
  6: 6
  7: 7
  8: 8
  9: 9
```

(Omitted)

```
  68: 68
  69: 69
  70: 70
  71: 71
  72: 72
  73: 73
  74: 74

[Document]
/export/work/adventure/AdvSolid-0.8beta/sample_data/cube_p2d2/model/advhddm_in_0.adv?341B2827:HDDM_Element@HDDM_Part[0]:5A85:39D89249
    size: 1032

[Properties]
  1: content_type=HDDM_Element
  2: num_subdomains=2
  3: element_type=3DLinearHexahedron
  4: num_nodes_per_element=8
  5: dimension=3
  6: sum_items=32
  7: num_items_orig=64

[Data]
# SubDomain[0]
# num items in SubDomain
    16
  0: 27 29 37 33 28 31 41 35
  1: 29 30 38 37 31 32 42 41
  2: 33 37 40 34 35 41 44 36
  3: 37 38 39 40 41 42 43 44
  4: 28 31 41 35 18 19 23 21
  5: 31 32 42 41 19 20 24 23
  6: 35 41 44 36 21 23 26 22
  7: 41 42 43 44 23 24 25 26
  8: 18 19 23 21 9 10 14 12
  9: 19 20 24 23 10 11 15 14
  10: 21 23 26 22 12 14 17 13
  11: 23 24 25 26 14 15 16 17
  12: 9 10 14 12 3 4 6 5
  13: 10 11 15 14 4 1 2 6
  14: 12 14 17 13 5 6 7 8
  15: 14 15 16 17 6 2 0 7
# SubDomain[1]
# num items in SubDomain
    16
  0: 27 29 37 33 28 31 41 35
  1: 29 30 38 37 31 32 42 41
  2: 33 37 40 34 35 41 44 36
  3: 37 38 39 40 41 42 43 44
```

```
          4: 28 31 41 35 18 19 23 21
          5: 31 32 42 41 19 20 24 23
  6: 35 41 44 36 21 23 26 22
  7: 41 42 43 44 23 24 25 26
  8: 18 19 23 21 9 10 14 12
  9: 19 20 24 23 10 11 15 14
  10: 21 23 26 22 12 14 17 13
  11: 23 24 25 26 14 15 16 17
  12: 9 10 14 12 3 4 6 5
  13: 10 11 15 14 4 1 2 6
  14: 12 14 17 13 5 6 7 8
  15: 14 15 16 17 6 2 0 7

[Document]
/export/work/adventure/AdvSolid-0.8beta/sample_data/cube_p2d2/model/advhddm_in_0.adv?42A1930E:HDDM_NodeIndex@HDDM_Part[0]:5A85:39D89249
   size: 368

[Properties]
  1: content_type=HDDM_FEGenericAttribute
  2: num_subdomains=2
  3: sum_items=125
  4: fega_type=AllNodeVariable
  5: format=i4
  6: label=NodeIndex_SubdomainToPart
  7: index_byte=4

[Data]
# SubDomain[0]
# num items in SubDomain
   45
  0: 68
  1: 62
  2: 65
  3: 60
  4: 61
  5: 63
  6: 64
  7: 67
  8: 66
  9: 45
```

(Omitted)

```
  38: 5
  39: 8
  40: 7
  41: 19
  42: 20
  43: 23
  44: 22
# SubDomain[1]
# num items in SubDomain
   45
  0: 74
  1: 68
  2: 71
  3: 66
  4: 67
  5: 69
  6: 70
  7: 73
  8: 72
  9: 51
```

(Omitted)

```
  39: 14
  40: 13
  41: 25
  42: 26
  43: 29
  44: 28

[Document]
/export/work/adventure/AdvSolid-0.8beta/sample_data/cube_p2d2/model/advhddm_in_0.adv?7F8062F5:InterfaceDOF@HDDM_Part[0]:5A85:39D89249
   size: 3008

[Properties]
  1: content_type=HDDM_FEGenericAttribute
  2: num_subdomains=2
  3: fega_type=NodeVariable
  4: format=i4i4i4i4
  5: label=InterfaceDOF
  6: index_byte=4
  7: ifd_dimension=3
  8: sum_items=150

[Data]
# SubDomain[0]
# num items in SubDomain
   75
  0: 0 0 0 66 66
  1: 0 1 0 67 67
  2: 0 2 0 68 68
  3: 1 0 0 60 60
```

```
          4: 1 1 0 61 61
          5: 1 2 0 62 62
  6: 2 0 0 63 63
  7: 2 1 0 64 64
  8: 2 2 0 65 65
  9: 7 0 0 102 102
```

(Omitted)

```
  70: 43 1 0 22 22
  71: 43 2 0 23 23
  72: 44 0 0 84 84
  73: 44 1 0 85 85
  74: 44 2 0 86 86
# SubDomain[1]
# num items in SubDomain
    75
  0: 0 0 0 72 72
  1: 0 1 0 73 73
  2: 0 2 0 74 74
  3: 1 0 0 66 66
  4: 1 1 0 67 67
  5: 1 2 0 68 68
  6: 2 0 0 69 69
  7: 2 1 0 70 70
  8: 2 2 0 71 71
  9: 3 0 0 99 99
```

(Omitted)

```
  68: 39 2 0 14 14
  69: 42 0 0 24 24
  70: 42 1 0 25 25
  71: 42 2 0 26 26
  72: 43 0 0 27 27
  73: 43 1 0 28 28
  74: 43 2 0 29 29

[Document]
/export/work/adventure/AdvSolid-0.8beta/sample_data/cube_p2d2/model/advhddm_in_0.adv?5D9FFE7F:HDDM_FEGA@HDDM_Part[0]:5A85:39D89249
    size: 872

[Properties]
  1: content_type=HDDM_FEGenericAttribute
  2: num_subdomains=2
  3: num_items_orig=75
  4: fega_type=NodeVariable
  5: format=i4f8
  6: label=ForceDisplacement
  7: index_byte=4
  8: sum_items=54

[Data]
# SubDomain[0]
# num items in SubDomain
    27
  0: 27 0 0.000000e+00
  1: 27 1 0.000000e+00
  2: 27 2 0.000000e+00
  3: 29 0 0.000000e+00
  4: 29 1 0.000000e+00
  5: 29 2 0.000000e+00
  6: 30 0 0.000000e+00
  7: 30 1 0.000000e+00
  8: 30 2 0.000000e+00
  9: 33 0 0.000000e+00
```

(Omitted)

```
  24: 39 0 0.000000e+00
  25: 39 1 0.000000e+00
  26: 39 2 0.000000e+00
# SubDomain[1]
# num items in SubDomain
    27
  0: 27 0 0.000000e+00
  1: 27 1 0.000000e+00
  2: 27 2 0.000000e+00
  3: 29 0 0.000000e+00
  4: 29 1 0.000000e+00
  5: 29 2 0.000000e+00
  6: 30 0 0.000000e+00
  7: 30 1 0.000000e+00
  8: 30 2 0.000000e+00
  9: 33 0 0.000000e+00
```

(Omitted)

```
  24: 39 0 0.000000e+00
  25: 39 1 0.000000e+00
  26: 39 2 0.000000e+00
```

```
         [Document]
         /export/work/adventure/AdvSolid-0.8beta/sample_data/cube_p2d2/model/advhddm_in_0.adv?1CA6F0C7:HDDM_FEGA@HDDM_Part[0]:5A85:39D89249
  size: 296

[Properties]
  1: content_type=HDDM_FEGenericAttribute
  2: num_subdomains=2
  3: num_items_orig=25
  4: fega_type=NodeVariable
  5: format=i4f8
  6: label=Load
  7: index_byte=4
  8: sum_items=18

[Data]
# SubDomain[0]
# num items in SubDomain
    9
  0: 3 2 -1.000000e-01
  1: 4 2 -1.000000e-01
  2: 1 2 -1.000000e-01
  3: 5 2 -1.000000e-01
  4: 6 2 -1.000000e-01
  5: 2 2 -1.000000e-01
  6: 8 2 -1.000000e-01
  7: 7 2 -1.000000e-01
  8: 0 2 -1.000000e-01
# SubDomain[1]
# num items in SubDomain
    9
  0: 3 2 -1.000000e-01
  1: 4 2 -1.000000e-01
  2: 1 2 -1.000000e-01
  3: 5 2 -1.000000e-01
  4: 6 2 -1.000000e-01
  5: 2 2 -1.000000e-01
  6: 8 2 -1.000000e-01
  7: 7 2 -1.000000e-01
  8: 0 2 -1.000000e-01

[Document]
/export/work/adventure/AdvSolid-0.8beta/sample_data/cube_p2d2/model/advhddm_in_0.adv?17791560:HDDM_FEGA@HDDM_Part[0]:5A85:39D89249
  size: 8

[Properties]
  1: content_type=HDDM_FEGenericAttribute
  2: num_subdomains=2
  3: num_items_orig=1
  4: fega_type=AllElementConstant
  5: format=f8
  6: label=YoungModule
  7: index_byte=4
  8: sum_items=1

[Data]
  2.100000e+04

[Document]
/export/work/adventure/AdvSolid-0.8beta/sample_data/cube_p2d2/model/advhddm_in_0.adv?22B722E9:HDDM_FEGA@HDDM_Part[0]:5A85:39D89249
  size: 8

[Properties]
  1: content_type=HDDM_FEGenericAttribute
  2: num_subdomains=2
  3: num_items_orig=1
  4: fega_type=AllElementConstant
  5: format=f8
  6: label=PoissonRatio
  7: index_byte=4
  8: sum_items=1

[Data]
  4.000000e-01

[Document]
/export/work/adventure/AdvSolid-0.8beta/sample_data/cube_p2d2/model/advhddm_in_0.adv?D53DCE0:HDDM_FEGA@HDDM_Part[0]:5A85:39D89249
  size: 8

[Properties]
  1: content_type=HDDM_FEGenericAttribute
  2: num_subdomains=2
  3: num_items_orig=1
  4: fega_type=AllElementConstant
  5: format=f8
  6: label=HardeningParameter
  7: index_byte=4
  8: sum_items=1

[Data]
  1.000000e+03

[Document]
/export/work/adventure/AdvSolid-0.8beta/sample_data/cube_p2d2/model/advhddm_in_0.adv?33E08E29:HDDM_FEGA@HDDM_Part[0]:5A85:39D89249
  size: 8

[Properties]
  1: content_type=HDDM_FEGenericAttribute
```

```
          2: num_subdomains=2
          3: num_items_orig=1
  4: fega_type=AllElementConstant
  5: format=f8
  6: label=YieldStress
  7: index_byte=4
  8: sum_items=1

[Data]
  5.000000e+02

[Document]
/export/work/adventure/AdvSolid-0.8beta/sample_data/cube_p2d2/model/advhddm_in_0.adv?77CED430:HDDM_FEGA@HDDM_Part[0]:5A85:39D89249
    size: 8

[Properties]
  1: content_type=HDDM_FEGenericAttribute
  2: num_subdomains=2
  3: num_items_orig=1
  4: fega_type=AllElementConstant
  5: format=f8
  6: label=Density
  7: index_byte=4
  8: sum_items=1

[Data]
  7.600000e+02

[Document]
/export/work/adventure/AdvSolid-0.8beta/sample_data/cube_p2d2/model/advhddm_in_0.adv?26873E3:HDDM_FEGA@HDDM_Part[0]:5A85:39D89249
    size: 24

[Properties]
  1: content_type=HDDM_FEGenericAttribute
  2: num_subdomains=2
  3: num_items_orig=1
  4: fega_type=AllElementConstant
  5: format=f8f8f8
  6: label=GravityAcceleration
  7: index_byte=4
  8: sum_items=1

[Data]
  0.000000e+00 0.000000e+00 -9.800000e+00

[Document]
/export/work/adventure/AdvSolid-0.8beta/sample_data/cube_p2d2/model/advhddm_in_0.adv?4852AF07:HDDM_FEGA@HDDM_Part[0]:5A85:39D89249
    size: 136

[Properties]
  1: content_type=HDDM_FEGenericAttribute
  2: num_subdomains=2
  3: num_items_orig=64
  4: fega_type=AllElementVariable
  5: format=i4
  6: label=ElementIndex_SubdomainToGlobal
  7: index_byte=4
  8: sum_items=32

[Data]
# SubDomain[0]
# num items in SubDomain
    16
  0: 0
  1: 1
  2: 4
  3: 5
  4: 16
  5: 17
  6: 20
  7: 21
  8: 32
  9: 33
  10: 36
  11: 37
  12: 48
  13: 49
  14: 52
  15: 53
# SubDomain[1]
# num items in SubDomain
    16
  0: 8
  1: 9
  2: 12
  3: 13
  4: 24
  5: 25
  6: 28
  7: 29
  8: 40
  9: 41
  10: 44
  11: 45
  12: 56
  13: 57
  14: 60
```

```
          15: 61

[Document]
/export/work/adventure/AdvSolid-0.8beta/sample_data/cube_p2d2/model/advhddm_in_0.adv?70B8D367:DocumentList[0]:5A85:39D89249
    size: 759

[Properties]
  1: content_type=DocumentList
  2: num_items=15
  3: num_subdomains=2
  4: num_parts=2
  5: part_number=0
  6: label=HDDM_FEA_Model

[Data]
  0: A8138FE:Node@HDDM_Part[0]:5A85:39D89249
  1: 1C2655B3:NodeIndex_PartToGlobal@HDDM_Part[0]:5A85:39D89249
  2: 56A68AB7:HDDM_InterfaceDOF@HDDM_Part[0]:5A85:39D89249
  3: 341B2827:HDDM_Element@HDDM_Part[0]:5A85:39D89249
  4: 42A1930E:HDDM_NodeIndex@HDDM_Part[0]:5A85:39D89249
  5: 7F8062F5:InterfaceDOF@HDDM_Part[0]:5A85:39D89249
  6: 5D9FFE7F:HDDM_FEGA@HDDM_Part[0]:5A85:39D89249
  7: 1CA6F0C7:HDDM_FEGA@HDDM_Part[0]:5A85:39D89249
  8: 17791560:HDDM_FEGA@HDDM_Part[0]:5A85:39D89249
  9: 22B722E9:HDDM_FEGA@HDDM_Part[0]:5A85:39D89249
  10: D53DCE0:HDDM_FEGA@HDDM_Part[0]:5A85:39D89249
  11: 33E08E29:HDDM_FEGA@HDDM_Part[0]:5A85:39D89249
  12: 77CED430:HDDM_FEGA@HDDM_Part[0]:5A85:39D89249
  13: 26873E3:HDDM_FEGA@HDDM_Part[0]:5A85:39D89249
  14: 4852AF07:HDDM_FEGA@HDDM_Part[0]:5A85:39D89249
```

References

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

[2]. METIS Home Page: <http://www-users.cs.umn.edu/~karypis/metis/>.

[3]. MPI Home Page: <http://www-unix.mcs.anl.gov/mpi/>.

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

[5]. T. Miyamura, S. Tanaka, H. Takubo, S. Yoshimura and G. Yagawa: Standardization of Input/Output Data in Large Scale Parallel Computational Mechanics System, Internet Transactions of Japan Society for Computational Engineering and Science (JSCES), No. 20000028 (2000) (in Japanese), <http://homer.shinshu-u.ac.jp/jsces/trans/trans2000/No20000028.pdf>.

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

[7]. G. Yagawa and R. Shioya: Parallel Finite Elements on a Massively Parallel Computer with Domain Decomposition, Computing Systems in Engineering, 4, Nos. 4-6 (1993), 495-503.