

# MODYLAS Tutorial

Version 1.0.2  
4 Nov. 2014

## **CONTENTS**

1. Overview of Directories and Files
    - 1.1 Directory structure
    - 1.2 Types of executable files
    - 1.3 Types of input/output files
  2. Installation
    - 2.1 Machine environment
    - 2.2 Compilation
  3. Sample 1: Water
    - 3.1 Overview of input files
    - 3.2 Execution of calculations
      - 3.2.1 Energy minimization
      - 3.2.2 NVT
      - 3.2.3 NPT
    - 3.3 Monitor of calculation results
      - 3.3.1 Monitor of Hamiltonian, temperature and pressure
      - 3.3.2 Display animation of trajectory
  4. Sample 2: Proteins in aqueous solution (10-million atom system)
    - 4.1 Overview of input files
    - 4.2 Execution of calculations
    - 4.3 Monitor of calculation results
- [Appendix] Calculation Using 65,536 Nodes of the K-Computer

## 1. Overview of Directories and Files

Decompressing the downloaded MODYLAS file "MODYLAS1.0.0.tar.gz" generates the directories and files described in the following sections.

```
%> tar zxvf MODYLAS0.1.0.0.tar.gz
```

### 1.1 Directory structure

(1) Under MODYLAS\_1.0.0/

- binary/ Directory containing pre-compiled MODYLAS execution module
- document/ Directory containing documents
- sample/ Directory containing sample input/output files
- src/ Directory containing MODYLAS program source files
- LISENCE.pdf File in which license information is recorded

(2) Under binary/

- K/ modylas Pre-compiled execution module that runs on the K-computer

(3) Under document/

- manual/ Directory containing the manual  
ReferenceManual\_1.0.0.pdf ... Reference Manual
- tutorial/ Directory containing the Tutorial  
Tutorial\_1.0.0.pdf ... Tutorial

(4) Under sample/

- water/ Input file for water (with tens of thousands of atoms)
- pyp111/ Input file for protein system (with tens of thousands of atoms)
- pyp888/ Input file for protein system (with 10 million atoms)

### 1.2 Types of executable files

- modylas Main program of MODYLAS
- modylas-text2bin Program to convert an ASCII input file to a binary one
- modylas-mdtrj2xyz Program to convert the trajectory of the binary output file "sessionname.mdtrj.bin" to an ASCII XYZ-format file

(For the paths to these files, see the preceding section.)

### 1.3 Types of input/output files

All input/output files of MODYLAS must have a common name except for the extension. The common name is expressed as "sessionname" . "sessionname" may be a user-specified string, but must not begin with a numerical character. Input/output files are distinguished from each other by the extension. The following section lists the input/output files:

## [Input files]

The "sessionname.mddef" file is used to define the details of the calculation conditions. The "sessionname.mdff" file is used to define force field parameters of each atom, and the "sessionname.mdxyz" file is used to define initial coordinates and velocities. As the input files, both ASCII-format and binary-format can be used. Normally, the use of ASCII-format input files is recommended. For the calculation of a very large system, however, the use of binary-format input files is recommended in order to reduce the time required to read the input files. If both ASCII-format and binary-format input files are found, the binary-format input files will override the ASCII-format input files.

- sessionname.mddef  
    Calculation conditions (ASCII format)
- sessionname.mdff  
    Force field information (ASCII format)
- sessionname.mdff.bin  
    Binary-format version of sessionname.mdff (binary format)
- sessionname.mdxyz  
    Atomic coordinates and velocities, and periodic-cell information (ASCII format)
- sessionname.mdxyz.bin  
    Binary-format version of sessionname.mdxyz (binary format)
- sessionname.posiconst (option)  
    External input file for specifying the constraint of atomic position (position constraint) (ASCII format). The constraint of atomic position can be specified with the force field input file (sessionname.mdff). However, if the sessionname.posiconst file exists, the sessionname.posiconst file will override the force field input file, and the specification in sessionname.mdff will be ignored. For the calculation of a very large system, the use of this external input file is strongly recommended, because MODYLAS execution module takes long time to read the position constraint information in the sessionname.mdff. When sessionname.mdff.bin is used instead of sessionname.mdff, this external input file must be created because any position constraint information are not converted into sessionname.mdff.bin.

## [Output files]

- sessionname.mdrun  
Information on the time spent for the calculation (ASCII format)
- sessionname.mdmntr  
Monitored values of physical quantities such as the Hamiltonian, temperature, and pressure (ASCII format)
- sessionname.mdtrj.bin  
Trajectory (atom coordinates and velocities) information for analysis (binary format)
- sessionname.restart.bin  
Atomic coordinates and velocities at the last time step for restart (binary format).  
This file has the same format as the input file of sessionname.mdxyz.bin.
- sessionname.restart.asc  
Atomic coordinates and velocities at the last time step for restart (ASCII format), which are in the same format as the input file of sessionname.mdxyz. This file is created by default, but is not created if "ascii=no" is specified between the <output> tags in sessionname.mddef., which are output in the same format as the input file (sessionname.mdxyz) (ASCII format). The data in this file is output by default, but is not output when "ascii=no" is specified between the <output> tags in sessionname.mddef.

(Note) Trajectory information file in the DCD format (i.e., sessionname.dcd) will be supported in near future.

## 2. Installation

### 2.1 Machine environment

- MODYLAS has been tested in the following systems:

K-computer, FX10, and PC cluster

- Compiler

The commands for compiling MODYLAS with MPI depend on the system. Therefore, you need to know the compiler installed in your system.

- [Fujitsu] frtpx (e.g., mpifrtpx or mpifrt)
- [Intel] ifort (e.g., mpif90 or mpiifort)
- [PGI] pgf90 (e.g., mpif90)

- MPI library

Normally, suitable MPI libraries are automatically linked by the compilation command for MPI.

### 2.2 Building MODYLAS

#### (A) Building modylas, modylas-text2bin, and modylas-mdtrj2xyz together

```
tar xvzf modylas-1.0.0.tar.gz
```

```
cd modylas-1.0.0/source/
```

```
setenv FC MPIF90CMD
```

Set FC environment variable only when the value set by the configure command is not suitable.

MPIF90CMD is the suitable Fortran 90 compiler for MPI.

```
setenv FCFLAGS FLAGS
```

Set FCFLAGS environment variable only when the value set by the configure command is not suitable.

In FLAGS, specify an optimization level.

```
./configure --with-kind-fortran-compiler=XXX
```

In XXX, specify the preset prepared for your system. The systems with preset are listed below. If your system is not listed below, specify whichever most closely corresponds to your system.

K	For the K-computer
FX10	For Fujitsu FX10
INTEL	For a system that uses an Intel compiler
PGI	For a system that uses a PGI compiler

After that, set the above environment variables as needed, and then execute the

configure command again or directly edit src/Makefile.

```
cd src
make -j N
```

N is the number of jobs to compile simultaneously. When the -j option is omitted, source files are compiled one by one.

#### (B) Building only modylas-text2bin and modylas-mdtrj2xyz without MPI

```
tar xvzf modylas-1.0.0.tar.gz
cd modylas-1.0.0/source/
setenv FC F90CMD
```

Set FC environment variable only when the value set by the configure command is not suitable.

F90CMD is the suitable Fortran 90 compiler.

```
setenv FCFLAGS FLAGS
```

Set FCFLAGS environment variable only when the value set by the configure command is not suitable.

In this environment variable, you need to describe necessary settings, such as optimization level.

```
./configure --with-kind-fortran-compiler=XXX --disable-mpi
```

In XXX, specify the preset prepared for your system. The systems with preset are listed below. If your system is not listed, specify whichever most closely corresponds to your system.

K	For the K-computer
FX10	For Fujitsu FX10
INTEL	For a system that uses an Intel compiler
PGI	For a system that uses a PGI compiler

After that, set the above environment variable as needed, and then execute the configure command again or directly edit src/Makefile.

```
cd src
make clean
make -j N
```

N is the number of jobs to compile simultaneously. When the -j option is omitted, source files are compiled one by one.

### 3. Sample 1: Water

This chapter shows the procedure for calculation of MODYLAS using a simulation of water molecules as a simple example. In order to obtain the trajectory of the thermal equilibrium state of water molecules, some standard procedures which are energy minimization, calculation of NVT ensemble at room temperature, and calculation of NPT ensemble at room temperature and atmospheric pressure, are carried out.

#### 3.1 Overview of input files

Sample input files for a molecular system with 4428 water molecules are put under the `sample/water/` directory. The input files for the initial structure and force fields can be created by using Nano-Ignition (input generator software) or VMD software with an output plug-in for MODYLAS (to be distributed soon) applied. The following sections give an example of the calculation procedure beginning from the step where input files have already been prepared. For information about the tags and keywords written in the input files, see the Reference Manual.

#### 3.2 Execution of calculations

For convenience, first of all, copy the MODYLAS execution module into the directory that contains the input files, or link the module to the directory.

```
%> cd sample/water/
%> ln -s (directory containing MODYLAS module)/modylas ./
```

##### 3.2.1 Energy minimization

Files with the session name "water\_opt" are the input files for energy minimization. Execution of the program must be conducted in an appropriate way depending on the computer system.

(1) In the case of a computer system which does not use batch jobs, type the following command:

```
%> mpiexec -n 8 ./modylas ./water_opt
```

(2) In the case of the computer system which uses batch jobs, submit a job with a specified command after preparing job script file. For example, on a K-computer, check the number of nodes that you want to use and file names in job script file, "run\_K.sh" (see ① to ⑧ for an example of job script file in Section 6.1 of the Reference Manual), and then execute the following command to submit the job:

```
%> pjsub ./run_K.sh
```

To check the status of a job, execute the following command:

```
%> pjstat
```

To delete a job, execute the following command:

```
%> pjdel (job-id-number)
```

### 3.2.2 NVT

Next, using the calculation result obtained from the energy minimization described in Section 3.2.1, perform the calculation for NVT ensemble. Execute the following commands to generate the three input files with session name, "water\_nvt" ,

```
%> cp water_opt.mddef water_nvt.mddef
%> cp water_opt.mdff water_nvt.mdff
%> cp water_opt.restart.asc water_nvt.mdxyz
```

The input files for the force field and initial configuration do not need to be modified. The water\_nvt.mddef file, however, needs to be modified with a text editor, such as vi or emacs, to change the calculation conditions.

```
%> vi water_nvt.mddef
```

or

```
%> emacs water_nvt.mddef
```

The keywords to be modified are of the ensemble, setting temperature, heat bath, initial velocity, and the number of steps (detailed explanation is omitted). For reference, see "water.mddef\_for\_nvt" as a modified input file put in the same directory. Modify the water\_nvt.mddef file referring to the water.mddef\_for\_nvt file.

(1) In the case of a computer system which does not use batch jobs, type the following command:

```
%> mpiexec -n 8 ./modylas ./water_nvt
```

(2) On the K-computer, modify the file name in job script file "run\_K.sh" (from "water\_opt" to "water\_nvt"), and then, submit the MODYLAS job as the following command:

```
%> pjsub ./run_K.sh
```

### 3.2.3 NPT

Using the calculation result obtained from the calculation for NVT ensemble, perform the calculation for NPT ensemble. As with the procedure described in the previous section, execute the following commands to generate the three input files with session name, "water\_npt":

```
%> cp water_nvt.mddef water_npt.mddef
%> cp water_nvt.mdff water_npt.mdff
```

```
%> cp water_nvt.restart.asc water_npt.mdxyz
```

To change the calculation conditions, modify the `water_npt.mddef` file with a text editor.

```
%> vi water_npt.mddef
```

or

```
%> emacs water_npt.mddef
```

The keywords to be changed are of ensemble, setting pressure, and pressure bath (detailed explanation is omitted). For reference, see "water.mddef\_for\_npt" as a modified input file located in the same directory. Modify the `water_npt.mddef` file by referring to the `water.mddef_for_npt` file.

(1) In the case of a computer system which does not use batch jobs, type the following command:

```
%> mpiexec -n 8 ./modylas ./water_npt
```

(2) On the K-computer, modify the file name in job script file "run\_K.sh" (from "water\_nvt" to "water\_npt"), and then, submit the MODYLAS job as the following command:

```
%> pjsub ./run_K.sh
```

### 3.3 Monitor of Calculation Results

This section briefly describes how to check the output data obtained from the calculations described in Section 3.2.

#### 3.3.1 Monitor of Hamiltonian, temperature and pressure

Physical quantities such as the Hamiltonian, temperature, and pressure values are written to the `sessionname.mdmntr` file sequentially with the step interval specified in the interval variable between the `<monitor>` and `</monitor>` tags in the `sessionname.mddef` file. The physical quantities are printed in the `sessionname.mdmntr` file and their units are shown in the header of the file. The physical quantities and units are also described in the Reference Manual. You can check the output values in the `sessionname.mdmntr` file with a text editor, such as `vi` or `emacs`, or a command, such as "less". If you want to display the output values as a graph, the `sessionname.mdmntr` file can be used directly as the input data for graphic display software. For example, graph of the output values can easily be displayed using `gnuplot`. In an environment that `gnuplot` is installed, type the following command to start `gnuplot`:

```
%> gnuplot
```

You can display a graph on the screen by typing commands after "gnuplot>". For example, to check the time course of potential energy in the energy minimization calculation, type the following command,

```
gnuplot> plot 'water_opt.mdmntr' using 4 with lines
```

where, the string between single quotations ( ' ') specifies the file name, "4" after "using" specifies display of the data (potential energy) in column 4 in the water\_opt.mdmntr file, and "with lines" specifies the connection of the displayed data by lines. To see the time course of temperature in the calculation in NPT ensemble, type the following command,

```
gnuplot> plot 'water_npt.mdmntr' using 2:7 with lines
```

where, "2:7" after "using" specifies display of the data in column 7 for the data in column 2.

### 3.3.2 Display animation of trajectory

Trajectory information (atomic coordinates and velocities) are output at specified time-step intervals to a file with an extension of ".mdtrj.bin" by the calculation of MODYLAS. As an example, this section describes how to display animation of the trajectory from the water\_npt.mdtrj.bin file obtained from the calculation in NPT ensemble.

Since the water\_npt.mdtrj.bin file is a binary-format file, first, you have to convert the coordinate data into ASCII format. The conversion program "modylas-mdtrj2xyz" can be used to convert into an xyz-format trajectory file. Move into the directory containing the input/output files, and set a link of the conversion program execution module "modylas-mdtrj2xyz" to the directory as follows,

```
%> ln -s (directory path)/modylas-mdtrj2xyz ./
```

Next, execute the following command:

```
%> ./modylas-mdtrj2xyz ./water_npt
```

Then, type the number of atoms to be printed, and press the Enter key. If you want to specify the total number of atoms, enter a negative value, e.g., "-1". Next, you are asked to specify the frame interval number at which data will be printed. If you want to convert all the frames of the output data, type "1", and then press the Enter key. This procedure generates a file named "water\_npt.xyz". If you want to display the animation of the file, use a visualization software for molecules, for example, a representative software, VMD (requiring separate installation). To display the animation, use the following command:

```
%> vmd ./water_npt.xyz
```

## 4. Sample 2: Proteins in aqueous solution (10 million atom system)

This chapter shows how to calculate a very large system (10 million atoms system) by using the K-computer. Here, a molecular system of proteins in aqueous solution is used as a sample.

### 4.1 Overview of input files

If you use ASCII-format input files as in the case of sample 1 when handling a very large system like a 10 million atom system, merely reading the input files will require a considerable time. To avoid this, convert the input files into binary format. A set of sample input files is put under the sample/pyp888/ directory.

### 4.2 Execution of calculations

For convenience, first, copy the MODYLAS execution module into the directory that contains the input files, or link the module to the directory.

```
%> cd sample/pyp888/
%> ln -s (directory containing MODYLAS module)/modylas ./
```

The files with session name, "pyp888", are the input files for this calculation. To execute the program, check the number of nodes that you want to use (set to 64 nodes, for example) and the file name in job script file "run\_K.sh" (see ① to ⑧ for an example of job script file in Section 6.1 of the Reference Manual), and then execute the following command to submit the job:

```
%> pjsub ./run_K.sh
```

To check the status of a job, use the following command:

```
%> pjstat
```

To delete a job, use the following command:

```
%> pjdel (job-id-number)
```

### 4.3 Monitor of calculation results

The monitoring of the calculation results and the visualization of the trajectories are the same as those explained for the calculation of the water sample in the previous chapter. Refer to the previous chapter, if necessary.

## [Appendix] Calculation Using 65,536 Nodes of the K-Computer

This appendix describes how to set up the calculation that uses MODYLAS and 65,536 nodes of the K-computer.

### (1) System design

As described in Section 8.2 of the Reference Manual, MODYLAS assigns the cell blocks formed by domain decomposition to individual processes. A cell block must contain one or more subcells, and the total number of subcells must be  $8^n$ . Based on these restrictions, " $n$ " must be 6 or more ( $8^5 < 65,536 < 8^6$ ) when 65,536 nodes are used. On the other hand, there is another restriction on the length of a side of a subcell, such that the length is not less than half of the cutoff radius of the Lennard-Jones potential. For these reasons, the length of a side of a unit cell must be set to  $\text{cutoff}/2 * 2^6$  or more. In an isothermal-isobaric  $NPT$  ensemble, the length of a side of a unit cell fluctuates with simulation time. Therefore, the length of a side of a unit cell should be set with a certain margin.

### (2) Example of mdef specifications

Usually, some calculation conditions, such as the time step ( $dt$ ), are independent of the number of MPI process. Thus, we explain only the calculation conditions that should be modified for the calculation using 65,536 nodes.

```
<mpi>
division=manual #manual/auto
nxdiv=64      # omitted, when division=auto
nydiv=32      # omitted, when division=auto
nzdiv=32      # omitted, when division=auto
</mpi>
```

First, change the process dividing method from automatic operation ( $\text{division}=\text{auto}$ ) to manual operation ( $\text{division}=\text{manual}$ ). Then, specify "64," "32," and "32" ( $64 \times 32 \times 32 = 65,536$ ) as the number of process divisions in the  $x$ ,  $y$ , and  $z$  directions (" $\text{nxdiv}$ ," " $\text{nydiv}$ ," and " $\text{nzdiv}$ "), respectively.

```
type=fmm
<fmm> nmax=4 ULswitch=3 ncell=64 </fmm>
```

Specify "fmm" as the electrostatic interaction calculation method ( $\text{type}$ ). Specify "64" as the number of cell divisions ( $\text{ncell}$ ) in each axial direction and also, specify "3" for "ULswitch" to

optimize communications.

### (3) Example of execution script

The following shows an example of execution script for a calculation using 65,536 nodes with 65,536 parallel processes, assuming that the language environment of the K-computer is K-1.2.0-15:

```
#!/bin/sh
#PJM --rsc-list "node=48x54x32:strict"
#PJM --rsc-list "elapse=1:00:00"
#PJM --rsc-list="rscgrp=huge"
#PJM --stg-transfiles all
#PJM --mpi "use-rankdir"
#PJM --mpi "rank-map-hostfile=./hostfile-48x54x32-64x32x32-all"
#PJM --stgin "rank=* ./modylas %r:./"
#PJM --stgin "rank=* ./pyp888.mddef %r:./"
#PJM --stgin "rank=* ./pyp888.mdff.bin %r:./"
#PJM --stgin "rank=* ./pyp888.mdxyz.bin %r:./"
#PJM --stgin "rank=* ./hostfile-48x54x32-64x32x32-all %r:./"
##
#PJM --stgout "rank=0 %r:./pyp888.mdmntr ./ "
#PJM --stgout "rank=0 %r:./pyp888.mdrun ./ "
#PJM --stgout "rank=0 %r:./pyp888.restart.bin ./ "
#PJM --stgout "rank=0 %r:./pyp888.mdtrj.bin ./ "
#PJM -s

# path
./work/system/Env_base_1.2.0-15

# Parameters
NPROCS=65536
NTHREADS=8
export PARALLEL=${NTHREADS}
export OMP_NUM_THREADS=${NTHREADS}
LPG="/opt/FJSVxosmmm/sbin/lpgparm -t 4MB -s 4MB -h 4MB -d 4MB -p 4MB"
```

```

# Endian
export FORT90L='-Wl,-T'

# MCA Parameters
MCA_PARAM="--mca common_tofu_fastmode_threshold 0"
MCA_PARAM="${MCA_PARAM} --mca common_tofu_max_fastmode_procs 40"

LD="/modylas ./pyp888"
mpiexec -n ${NPROCS} ${MCA_PARAM} ${LPG} ${LD}

```

Two special operations are required to execute the calculation using the 65,536 nodes of the K-computer at the maximum efficiency of MODYLAS.

- (1) Changing the logical node shape: In the default configuration, the logical node shape of the K-computer is set to "48 × 54 × 32" and it has to be changed to "64 × 32 × 32". To change this setting, use the map-hostfile specified in #PJM --mpi "rank-map-hostfile= ". For details on specification format of map-hostfile, see the manual for the K-computer.
- (2) Using all nodes exclusively: Changing the logical node shape to "64 × 32 × 32" by using map-hostfile requires the exclusive use of all nodes. Specify the exclusive use in #PJM --rsc-list "node=48x54x32:strict".

In addition to these two operations, it is necessary to ensure that the process shape (nxdiv, nydiv, nzdiv) specified in the mdef file are compatible with the new logical node shape.

In the above example of execution script, the set of the input files to be staged-in and the binary execution module (modylas) are specified in the #PJM --stgin line. Meanwhile, the set of output files to be staged-out is specified in the #PJM --stgout line. The number of processes and the number of threads are specified as usual in the NPROCS and NTHREADS parameters, respectively. When the input binary files are created in the little endian format, '-Wl,-T' must be set in environment variable "FORT90L". MODYLAS and the input file are specified in the LD parameter (LD="/modylas ./pyp888"), and parallel operation is executed by the mpiexec command (mpiexec -n \${NPROCS} \${MCA\_PARAM} \${LPG} \${LD}).

For more information, see the manual for the K-computer.