FrontISTR Installation Manual

FrontISTR Commons

October 2, 2019

Contents

1 FrontISTR Installation Manual **2**
1.1 Manuals . . . 2
1.2 List of description on this manual . . . 2
1.3 Operating Environment . . . 3
1.3.1 Required software . . . 3
1.3.2 Operation verification environment . . . 4
1.4 Extracting the Archive File . . . 5
1.5 Installation . . . 5
1.5.1 Supported installation procedure . . . 5
1.6 Compile with cmake . . . 6
1.6.1 Preparation . . . 6
1.6.2 Compiling . . . 7
1.6.3 Executing `make install` . . . 7
1.6.4 cmake options . . . 7
1.7 About test . . . 8
1.8 About documentation of source code . . . 8
1.9 Enable Debugging . . . 9
1.10 Compile with editing Makefile.conf manually . . . 9
1.10.1 Editing Makefile.conf . . . 9
1.10.2 Executing setup.sh . . . 10
1.10.3 Executing make . . . 11
1.10.4 Executing make install . . . 11
1.10.5 Installing in Windows environments . . . 11
1.11 Appendix . . . 11
1.11.1 List of `Makefile.conf` Variables . . . 11
1.11.2 Example of Makefile.conf . . . 15
1.11.3 Notes on K-computer and Fujitsu FX10 . . . 16
1.12 Appendix . . . 17
1.12.1 List of `Makefile.conf` Variables . . . 17
1.12.2 Example of Makefile.conf . . . 20
1.12.3 Notes on K-computer and Fujitsu FX10 . . . 22
1.13 Appendix : Example of installation procedure to CentOS7.6(cmake) . . . 22
1.13.1 Preparation . . . 22
1.13.2 Installing libraries . . . 22
1.13.3 Compiling FrontISTR . . . 25
1.14 Appendix : Example of installation procedure to CentOS7.6 (Makefile.conf) . . . 26
1.14.1 Preparation . . . 26
1.14.2 Installing libraries . . . 27
1.14.3 Compiling FrontISTR . . . 29
1.15 Appendix : Example of installation procedure to Ubuntu18.04(cmake) . . . 32
1.15.1 Preparation . . . 32
1.15.2 Installing libraries . . . 32
1.15.3 Compiling FrontISTR . . . 35
1.16 Appendix : Example of installation procedure to Ubuntu18.04(Makefile.conf) . . . 36
1.16.1 Preparation . . . 36

1.16.2 Installing libraries . . . 36
1.16.3 Compiling FrontISTR . . . 39
1.17 Appendix : Example of installation procedure to Windows10(Makefile.conf) . . . 42
1.17.1 Preparation . . . 42
1.17.2 Installing libraries . . . 42
1.17.3 Compiling FrontISTR . . . 46

1 FrontISTR Installation Manual

Item	Content
Name of Software	FrontISTR
Version	5.0
License	MIT License
Correnponding Clerks	FrontISTR Commons2-11-16 Yayoi, Bunkyo-ku, Tokyoc/o Institute of Engineering Innovation, School of EngineeringE-mailsupport@frontistr.com

1.1 Manuals

- Introduction
- How to install
- Theory
- User’s manual
- Tutorial

This manual explains the installation procedure for a large-scale structural analysis program using the finite element method (FEM), FrontISTR.

1.2 List of description on this manual

- PDF
- Installation
  - Required software
  - Extracting the Archive File
  - Installation
  - Compile with cmake
    - Example of installation procedure to CentOS7.6 (cmake)
    - Example of installation procedure to Ubuntu18.04 (cmake)
  - Compile with editing Makefile.conf manually
    - List of Makefile.conf Variables
    - Example of installation procedure to CentOS7.6 (Makefile.conf)

* Example of installation procedure to Ubuntu18.04 (Makefile.conf)
* Example of installation procedure to Windows10 (Makefile.conf)

1.3 Operating Environment

1.3.1 Required software

To install this software, the following software programs must have been installed in the environment where this software is to be installed. For information on how to install these software programs, refer to their installation manuals.

1.3.1.1 C, C++ and Fortran90 compilers C, C++, and Fortran90 compilers are required to install this software.

1.3.1.2 MPI This software executes parallel processing via MPI and requires an MPI library that compiles wihh the MPI-1 standards. Representative free-of-charge libraries that implement MPI include MPICH and OpenMPI. OpenMPI and MPICH can be downloaded from the following website:

Open MPI: Open Source High Performance Computing

- https://www.open-mpi.org/

High-Performance Portable MPI

- http://www.mpich.org/

1.3.1.3 METIS The domain decomposition utilities of this software use the METIS libraries to enable domain decomposition using pMETIS and kMETIS. METIS is required to use these domain decomposition functions. Supported versions of METIS are the newest version Ver.5 series and previous version Ver.4 series.

For environments where METIS is not installed, domain decomposition is still possible using the RCB algorithm.

METIS - Serial Graph Partitioning and Fill-reducing Matrix Ordering

- http://glaros.dtc.umn.edu/gkhome/metis/metis/overview

1.3.1.4 ParMETIS We plan to use the ParMETIS library as the parallel domain decomposition utilities for this software.

ParMETIS is not required at present.

ParMETIS - Parallel Graph Partitioning and Fill-reducing Matrix Ordering

- http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview

1.3.1.5 HEC-MW This software uses the HEC-MW library developed by the “Revolutionary Simulation Software” and “Research and Development of Innovative Simulation Software” projects.

HEC-MW comes bundled with the FrontISTR archive file, and is automatically compiled when this software is installed, so there is no need to install HEC-MW separately.

1.3.1.6 REVOCAP_Refiner This software is compatible with the REVOCAP_Coupler coupled analysis tool developed by the “Research and Development of Innovative Simulation Software” project. REVOCAP_Coupler is required to use the coupled analysis function. REVOCAP_Coupler can be downloaded from the following website:

- http://www.multi.k.u-tokyo.ac.jp/FrontISTR/

1.3.1.7 REVOCAP_Coupler

This software is compatible with the REVOCAP_Coupler coupled analysis tool developed by the "Research and Development of Innovative Simulation Software" project.REVOCAP_Coupler is required to use the coupled analysis function. REVOCAP_Coupler can be downloaded from the following website:

- http://www.ciss.iis.u-tokyo.ac.jp/dl/index.php

1.3.1.8 LAPACK/BLAS

This software is implemented a function of estimating condition number of preconditioned matrix.

When you use this function, LAPACK and BLAS libraries are required.

Reference implements of LAPACK can be downloaded from the following website:

- http://www.netlib.org/lapack/

Reference implements of BLAS can be downloaded from the following website:

- http://www.netlib.org/blas/

OpenBLAS is compatible, fast and Open-Sourced implements of LAPACK/BLAS. OpenBLAS can be downloaded from the following website:

- http://www.openblas.net/

If Intel MKL is installed, it is not necessary to install above software.

1.3.1.9 MUMPS

This software is compatible with a public domain parallel direct solver MUMPS (a MUltifrontal Massively Parallel sparse direct Solver). MUMPS is based on public domain software developed during the Esprit IV European project PARASOL (1996-1999). Since this first public domain version in 1999, research and developments have been supported by the following institutions: CERFACS, CNRS, ENS Lyon, INPT(ENSEEIHT)-IRIT, INRIA, and University of Bordeaux. MUMPS can be downloaded from the following website:

- http://mumps.enseeiht.fr/

1.3.1.10 ScaLAPACK

Therefore this software doesn't use this directly, MUMPS needs ScaLAPACK for compiling. ScaLAPACK can be downloaded from the following website:

- http://www.netlib.org/scalapack/

If Intel MKL is installed, it is not necessary to install above software.

1.3.1.11 ML

This software is compatible with multigrid preconditioner package ML (Multi-Level Preconditioner). ML is one of the packages developed under Trilinos project at Sandia National Laboratories. ML can be downloaded from the following website:

- https://trilinos.org/

1.3.1.12 Intel MKLMath Kernel Library

The contact analysis module of this software uses Intel MKL. If Intel MKL has not been installed in the environment where this software is to be installed, some of the contact analysis functions cannot be used.

1.3.2 Operation verification environment

Operations of this software have been verified in the following environment. However, if the software programs (described above) required for installing this software have been installed, this software should operate normally even in environments other than the environment shown below.

1.3.2.1 Operation verification environment

System	Operatingsystem	CPU	Compiler	Parallel processingenvironment
K computer	Linux	SPARC64 VIIIfx	Fujitsu compiler	Fujitsu MPI
PRIMEHPC FX100	Linux	SPARC V9 + HPC-ACE2	Fujitsu compiler	Fujitsu MPI
EARTH SIMULATOR(ES3)	SUPER UX	SX-ACE	NEC compiler	NEC MPI
UV2000	Linux (SUSE Linux Enterprise 10)	Intel Xeon	Intel compiler	SGI MPT
PC cluster	Linux (CentOS-7)	Intel Xeon	Intel compiler	Intel MPI
PC cluster	Linux (RedHat Enterprise Linux 7)	Intel Xeon	Intel compiler	OpenMPI
Desktop PC	Linux (ubuntu 16.04, 18.04)	AMD Ryzen	GNU compiler	OpenMPI
Desktop PC	Linux (ubuntu 16.04, 18.04)	AMD Ryzen	PGI compiler	OpenMPI
Desktop PC	Linux (ubuntu 16.04, 18.04)	Intel Core-i7	GNU compiler	OpenMPI
Desktop PC	Windows (7, 10)	Intel Core-i7	GNU compiler (mingw)	Microsoft MPI
Raspberry PI 3 B+	Linux (raspbian 32bit)	ARM Cortex-A53	GNU compiler	OpenMPI
Notebook PC	macOS Mojave	Intel Core i7	GNU Compiler	OpenMPI

1.4 Extracting the Archive File

The archive file has been archiving with the tar command and compressing with gzip. It can be extracted using the following command. ("$" at the beginning of the command string represents a prompt.)

```
$ tar xzf FrontISTR_V50.tar.gz
```

If the tar command in the environment where this software is to be installed does not support the z option, the archive file can be extracted using the following command:

```
$ gzip -dc FrontISTR_V50.tar.gz | tar xf -
```

If the archive file is extracted, a directory named "FrontISTR" will be created in the directory where the archive file has been extracted. (This directory is hereafter referred to a "${FSTRBUILDDIR}".)

1.5 Installation

1.5.1 Supported installation procedure

This software supports two procedure for compiling.

1.5.1.1 Compile with cmake

This software supports compiling the software using cmake.

cmake need to be installed in advance. cmake can be downloaded from the following website:

https://cmake.org/

```
$ cd `${FSTRBUILDDIR}`
$ mkdir build
$ cd build
$ cmake ..
$ make -j2
$ make install
```

cmake will search libraries and headers and gererate proper Makefiles for compiling. If you have multi-cored CPU, you will run parallel make for saving compile time.

continue… (Compile with cmake)

1.5.1.1.1 Appendix

- Appendix: Example of installation procedure to CentOS7.6(cmake)
- Appendix: Example of installation procedure to Ubuntu18.04(cmake)

1.5.1.2 Compile with editing Makefile.conf manually This software supports compiling the software manually edited Makfile.conf.

```
$ cd `${FSTRBUILDDIR}`
$ cp Makefile.conf.org Makefile.conf
$ vi Makefile.conf
  Edit Makefile.conf manually, indicate the location of libraries and headers.
$ ./setup.sh [Options]
$ make
$ make install
```

When difficult to set automatic configuration with cmake, recommend manually editing Makefile.conf.

continue… (Compile with editing Makefile.conf)

1.5.1.2.1 Appendix

- Appendix: Example of installation procedure to CentOS7.6(Makefile.conf)
- Appendix: Example of installation procedure to Ubuntu18.04(Makefile.conf)
- Appendix: Example of installation procedure to Windows10(Makefile.conf)

1.6 Compile with cmake

cmake will find libraries and headers automatically. You can also specify libraries and headers manually.

Detiled usage of cmake command, see https://cmake.org/.

1.6.1 Preparation

In advance, required software install as following directory structure.

```
$HOME
  |-- local
        |-- bin
        |-- include
        |-- lib
```

Add $HOME/local/bin in $PATH environment variable.

Next, check whether already installed cmake. You may need to install version 2.8.11 or later.

```
$ cmake --version
cmake version 2.8.12.2
```

1.6.2 Compiling

Compile FrontISTR.

```
$ cd `${FSTRBUILDDIR}`
$ mkdir build
$ cd build
$ cmake ..
$ make -j2
```

-j2 option follows make command tells make to execute simultaneously. Reduce compile time when increasing the number of simultaneous.

1.6.3 Executing `make install`

When finished to compile, execute following command.

```
$ make install
```

This software will be installed in the /usr/local/bin directory or in the directory specified by -DCMAKE_INSTALL_PREFIX.

```
$ cmake -DCMAKE_INSTALL_PREFIX=$HOME/local ..
```

Which options enabled in compiled FrontISTR(fistr1) can confirm as following command.

```
$ ./fistr1 -v
FrontISTR version 5.0.0 (2d3fdb51979459c7ea9357a7c9b790fa69dfd4e2)
MPI: Enabled
OpenMP: Enabled
HECMW_METIS_VER: 5
Compile Option: -p --with-tools --with-metis --with-mumps --with-lapack --with-ml
```

1.6.4 cmake options

You can explicitly various options at configuration time.

Options (Default)	Contents	Remarks
-DWITH_TOOLS=ON	Compiling tools such as partitioners	hecmw_part1 etc.
-DWITH_MPI=ON	Enable MPI	require libraries
-DWITH_OPENMP=ON	Enable OpenMP	require supported compiler
-DWITH_REFINER=ON	Enable REVOCAP_Refiner functionality	require libraries
-DWITH_REVOCAP=ON	Enable REVOCAP_Coupler functionality	require libraries
-DWITH_METIS=ON	Enable METIS functionality	4.0.3 and 5.1.0 supported
-DMETIS_VER_4=OFF	In case of using metis-4.0.3, specify ON	In case of using metis-5.1.0, it isn't necessary to specify it.
-DWITH_PARMETIS=ON	Enable ParMETIS functionality	3.2.0 and 4.0.3 supported
-DMETIS_VER_3=OFF	In case of using ParMetis-3.2.0, specify ON	In case of using parmetis-4.0.3, it isn't necessary to specify it.
-DWITH_MKL=ON	Enable MKL PARDISO functionality	require libraries
-DWITH_MUMPS=ON	Enable MUMPS functionality	require libraries
-DWITH_LAPACK=ON	Enable LAPACK functionality	require libraries
-DWITH_ML=ON	Enable Trilinos ML functionality	require libraries
-DWITH_DOC=OFF	Documentation of the source code	require doxygen and graphviz
-DOLD_RES_FORMAT=OFF	Save result file format if ON is specified	

The list of variables that can be set with cmake can be confirmed with the following command.

```
$ cmake -L
```

Other options are as follows.

Options	Contents	Remarks
-DBLA_VENDOR=	Specify vendor name of BLAS and LAPACK	Refer to the FindBLAS.cmake
-DBLAS_LIBRARIES=	Directly specifies BLAS library	Absolute path
-DLAPACK_LIBRARIES=	Directly specifies LAPACK library	Absolute path
-DCMAKE_INSTALL_PREFIX=	Specify installing path. Default is /usr/local	If specifies -DCMAKE_INSTALL_PREFIX=$HOME/local,binar will copy to $HOME/local/bin
-DCMAKE_C_COMPILER=	Specify C compiler	-DCMAKE_C_COMPILER=icc (ex. Intel C compiler)
-DCMAKE_CXX_COMPILER=	Specify C++ compiler	-DCMAKE_CXX_COMPILER=icpc (ex. Intel C++ compiler)
-DCMAKE_Fortran_COMPILER=	Specify Fortran compiler	-DCMAKE_Fortran_COMPILER=ifort (ex. Intel Fortran compiler)
-DCMAKE_PREFIX_PATH=	Specify path of libraries	-DCMAKE_PREFIX_PATH=$HOME/tools

1.7 About test

This software includes test scripts for validating compiled binaries.

In order to execute the test you need to install ruby in advance. If ruby is already installed, test is automatically enabled.

Test is executed as follows.

```
$ make test
```

Results as follows.

```
Running tests...
Test project /home/fistr/FrontISTR/build
      Start  1: Static_exA_Test
 1/23 Test  #1: Static_exA_Test ..................   Passed    3.54 sec
      Start  2: Static_exB_Test
 2/23 Test  #2: Static_exB_Test ..................   Passed    2.51 sec
...
```

You can output verbosed messages as follow.

```
$ make test ARGS="-VV -j4 -O test_log.txt"
```

Results will output in test_log.txt file. For more information;

```
$ ctest --help
```

1.8 About documentation of source code

For learning or investigating the source code, you can generate HTML documents that describes relationship functions from the source codes.

You need install doxygen and graphviz in advance.

```
$ cmake -DWITH_DOC=ON ..
$ make doc
```

Generated HTML can browse like as follow.

```
$ firefox doc/html/index.html
```

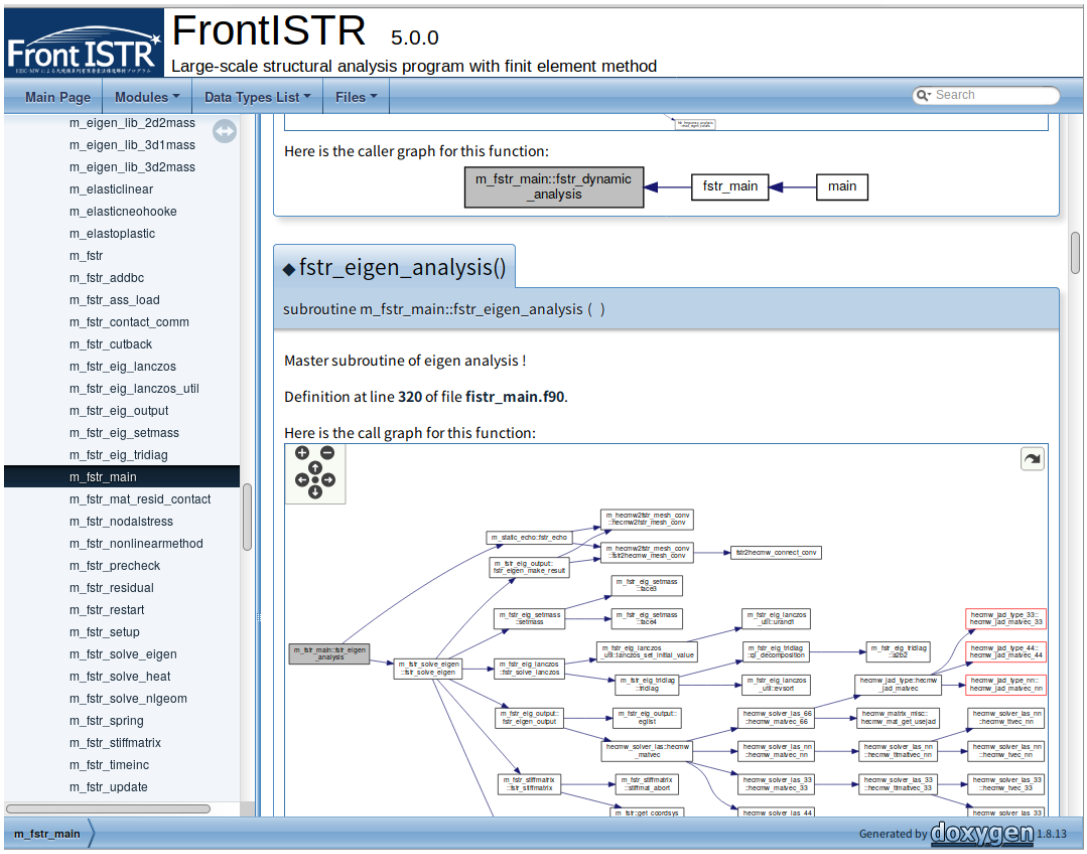

Figure 1: Doxygen

1.9 Enable Debugging

To enable debugging, specify the following option.

```
$ cmake -DCMAKE_BUILD_TYPE="DEBUG" ..
```

To perform more advanced debugging, specifying the following options is useful for investigating memory leaks.

```
$ cmake -DCMAKE_BUILD_TYPE="DEBUG" -DDEBUG_EXTRA=ON ..
```

1.10 Compile with editing Makefile.conf manually

1.10.1 Editing Makefile.conf

Create Makefile.conf by editing Makefile.conf.org in ${FSTRBUILDDIR} to match the computer environment where this software is to be installed. There are a large number of variables that can be defined, but most of them can be used "as is" (without having to change the default values). For most environments, there should be no need to change variables other than those shown below.

Variable	Description
MPIDIR	Specifies the directory where MPI has been installed

Variable	Description
PREFIX	Specifies the directory where the executable module of this software is to be installed
METISDIR	Specifies the directory where METIS has been installed
PARMETIS-DIR	Specifies the directory where ParMETIS has been installed
REFIN-ERDIR	Specifies the directory where REVOCAP_Refiner has been installed
REVO-CAPDIR	Specifies the directory where REVOCAP_Coupler has been installed
MUMPSDIR	Specifies the directory where MUMPS has been installed
CC	Specifies the C compiler start command
CPP	Specifies the C++ compiler start command
F90	Specifies the Fortran90 compiler start command

Refer to Appendix A, "List of Makefile.conf Variables" for details on all variables. Refer also to Appendix B, "Makefile.conf Setting Example" for an example of Makefile.conf settings.

1.10.2 Executing setup.sh

Create Makefile by executing the setup.sh shell script in ${FSTRBUILDDIR}, as below.

```
$ ./setup.sh
```

To generate a library for parallel computing, for example, execute setup.sh with the following options specified.

1.10.2.1 Options specified when setup.sh is executed

Option	Description	Remarks
-g or –debug	Generates a library for debugging	
-p or –parallel	Generates a library for parallel execution	
–with-tools	Generates a tool such as a partitioner	
–with-refiner	Includes REVOCAP_Refiner	
–with-revocap	Includes REVOCAP_Coupler	
–with-metis	Uses METIS	
–with-parmetis	Uses ParMETIS	Disabled at present
–with-mkl	Uses Intel MKL	
–with-mumps	Uses MUMPS	
–with-lapack	Uses Lapack	Needed for condition number estimation
–with-ml	Uses ML	
–old-res-format	Revert result file format	

Examples of setup.sh execution are shown below.

1.10.2.2 Compiling for parallel partitioner If this software is used in parallel execution environment where MPI has installed, execute setup.sh with the −p or −−parallel option specified, as below.

```
$ ./setup.sh –p
```

1.10.2.3 Generating a tool such as a partitioner If a processing or post-processing tool such as a partitioner (RCB) or visualizer is required, execute setup.sh with the −−with−tools option specified, as below.

```
$ ./setup.sh –p —with−tools
```

1.10.2.4 Using METIS If METIS is used with a partitioner in an environment where METIS has been installed, execute setup.sh with the −−with−metis option specified, as below.

```
$ ./setup.sh -p --with-tools --with-metis
```

1.10.2.5 Compiling for contact analysis If this software is used in a sequential processing with contact analysis, execute setup.sh with the −−with−mkl or −−with−mumps option specified, as below.

Sequential processing

```
$ ./setup.sh --with-mkl
```

or

```
$ ./setup.sh --with-mumps
```

If this software is used in a parallel processing with contact analysis, execute setup.sh with −p, −−with−metis and −−with−mumps options specified, as below.

```
$ ./setup.sh -p --with-metis --with-mumps
```

1.10.3 Executing make

Execute make in ${FSTRBUILDDIR} as below.

```
$ make 2 > & 1 | tee make.log
```

It may take sevral minutes to execute make, depending on the computer environment. If an error occurs during execution, take appropriate actions such as reviewing the Makefile.conf

1.10.4 Executing make install

After make has finished executing normally, execute make install to install this software in the directory specified in Makefile.conf, as below.

```
$ make install
```

1.10.5 Installing in Windows environments

In Windows environments, the following UNIX-like environments can be used to install this software according to the above procedures.

- Sequential processing version : MinGW, Cygwin
- Parallel processing version : MinGW + Microsoft MPI, Cygwin + OpenMPI

1.11 Appendix

1.11.1 List of `Makefile.conf` Variables

1.11.1.1 Setting related to MPI

Variable name	Description	Default
MPIDIR	This variable specifies the path to the directory where MPI has been installed. If an MPI compatible compiler automatically refers to the path, there is no need to set this variable together with the following variables.	None
MPIBINDIR	This variable specifies the path to the directory where MPI executable files have been installed.	None

Variable name	Description	Default
MPIINCDIR	This variable specifies the path to the directory where MPI header files have been installed.	.
MPILIBDIR	This variable specifies the path to the directory where MPI libraries have been installed.	.
MPILIBS	This variable specifies the MPI library that will be linked to C and Fortran90 object files.	None

1.11.1.2 Setting related to the installation directory

Variable name	Description	Default
PREFIX	This variable specifies the path to the directory where this software is to be installed.	$(HOME)/FrontISTR
BINDIR	This variable specifies the path to the directory where the executable files of this software are to be installed.	$(PREFIX)/bin
INCLUDEDIR	This variable specifies the path to the directory where the header files of this software are to be installed. Normally, there is no need to change the value of this variable from the default value.	$(PREFIX)/include
LIBDIR	This variable specifies the path to the directory where the libraries of this software are to be installed. Normally, there is no need to change the value of this variable from the default value.	$(PREFIX)/lib

1.11.1.3 Settings related to METIS

Variable name	Description	Default
METISDIR	This variable specifies the path to the directory where METIS has been installed.	$(HOME)/metis
METISINCDIR	This variable specifies the path to the directory where the header files (such as metis.h) of METIS have been installed. Normally, there is no need to change the value of this variable from the default value.	$(METISDIR)/include
METISLIBDIR	This variable specifies the path to the directory where the library (libmetis .a) of METIS has been installed. Normally, ther is no need to change the value of this variable from the default value.	$(METISDIR)/lib

1.11.1.4 Settings related to ParMETIS

Variable name	Description	Default
PARMETISDIR	This variable specifies the path to the directory where ParMETIS has been installed.	$(HOME)/ParMetis
PAEMETIS-INCDIR	This variable specifies the path to the directory where the header files (such as parmetis.h) of ParMETIS have been installed. Normally, there is no need to change the value of this variable from the default value.	$(PARMETISDIR)/include
PARMETISLIB-DIR	This variable specifies the path to the directory where the library (libparmetis.a) of ParMETIS has been installed. Normally, there is no need to change the value of this variable from the default value.	$(PARMETISDIR)/lib

1.11.1.5 Settings related to REVOCAP_Refiner

Variable name	Description	Default
REFINERDIR	This variable specifies the path to the directory where REVOCAP_Refiner has been installed.	$(HOME)/REVOCAP_Refiner
REFINER-INCDIR	This variable specifies the path to the directory where REVOCAP_Refiner header files have been installed. Normally, there is no need to change the value of this variable from the default value.	$(PARMETISDIR)/include
REFINERLIB-DIR	This variable specifies the path to the directory where REVOCAP_Refiner libraries have been installed. Normally, there is no need to change the value of this variable from the default value.	$(PARMETISDIR)/lib

1.11.1.6 Settings related to REVOCAP_Coupler

Variable name	Description	Default
REVOCAPDIR	This variable specifies the path to the directory where REVOCAP_Coupler has been installed.	$(HOME)/REVOCAP_Coupler
REVOCAP-INCDIR	This variable specifies the path to the directory where REVOCAP_Coupler header files have been installed. Normally, there is no need to change the value of this variable from the default value.	$(REVOCAPDIR)/include
REVOCAPLIB-DIR	This variable specifies the path to the directory where REVOCAP_Coupler libraries have been installed. Normally, there is no need to change the value of this variable from the default value.	$(REVOCAPDIR)/lib

1.11.1.7 Settings related to MUMPS

Variable name	Description	Default
MUMPSDIR	This variable specifies the path to the directory where MUMPS has been installed.	$(HOME)/MUMPS
MUMPSINCDIR	This variable specifies the path to the directory where MUMPS header files have been installed. Normally, there is no need to change the value of this variable from the default value.	$(MUMPSDIR)/include
MUMPSLIBDIR	This variable specifies the path to the directory where MUMPS libraries have been installed. Normally, there is no need to change the value of this variable from the default value.	$(MUMPSDIR)/lib

1.11.1.8 Settins related to ML

Variable name	Description	Default
MLDIR	This variable specifies the path to the directory where ML is installed.	$(HOME)/trilinos
MLINCDIR	This variable specivies the path to the directory where ML header files are installed. Normally, there is no need to change the value of this variable from the default value.	$(MLDIR)/include
MLLIBDIR	This variable specivies the path to the directory where ML libraries are installed. Normally, there is no need to change the value of this variable from the default value.	$(MLDIR)/lib

1.11.1.9 Settings related to the C compiler

Variable name	Description	Default
CC	This variable specifies the C compiler start command	mpicc
CFLAGS	This variable specifies the option to be assigned to the C compiler. Normally, there is no need to change the value of this variable from the default value.	
LDFLAGS	This variable specifies the option to be assigned to the C linker. Normally, there is no need to change the value of this variable from the default value. However, when REVOCAP_Refiner (written in C++) is to be used and C compiler is used for linking C programs, C++ standard library (e.g. -lstdc++) needs to be specified here.	−lm
OPTFLAGS	This variable specifies the optimization option (or another option) to be assigned to the C compiler.	−O3
CLINKER	This variable specifies the linker command for C program. This is used when, for example, REVOCAP_Refiner (written in C++) is to be used and C++ compiler needs to be used for linking C programs with C++ libraries.	$(CC)

1.11.1.10 Settings related to the C++ compiler

Variable name	Description	Default
CPP	This variable specifies the C++ compiler start command.	mpic++
CPPFLAGS	This variable specifies the option to be assigned to the C++ compiler. Normally, there is no need to change the value of this variable from the default value. However, if Boost Libraries are not automatically referenced from the C++ compiler, use the -I option to specify the directory that contains the include files.	−DMPICH_IGNORE_CXX_SEEK (Note: This is required for Intel compilers.)
CPPLDFLAGS	This variable specifies the option to be assigned to the C++ linker. Normally, there is no need to change the value of this variable from the default value.	None
CPPOPTFLAGS	This variable specifies the optimization option (or another option) to be assigned to the C++ compiler.	−O3

1.11.1.11 Settings related to Fortran90 compiler

Variable name	Description	Default
F90	This variable specifies the Fortran90 compiler start command.	mpif90
F90FLAGS	This variable specifies the option to be assigned to the Fortran90 compiler. Normally, there is no need to change the value of this variable from the default value.	−DMPICH_IGNORE_CXX_SEEK
F90LDFLAGS	This variable specifies the option to be assigned to the Fortran90 linker. Normally, there is no need to change the value of this variable from the default value. However, if Intel MKL is used, specify its link option. Also, when REVOCAP_Refiner (written in C++) is to be used and Fortran90 compiler is used for linking Fortran90 programs, C++ standard library (e.g. -lstdc++) needs to be specified here.	None
F90OPTFLAGS	This variable specifies the optimization option (or another option) to be assigned to the Fortran90 compiler.	−O2

Variable name	Description	Default
F90LINKER	This variable specifies the linker command for Fortran90 program. This is used when, for example, REVOCAP_Refiner (written in C++) is to be used and C++ compiler needs to be used for linking Fortran90 programs with C++ libraries. (E.g. on K-computer, "mpiFCCpx –linkfortran" needs to be specified.)	$(F90)

1.11.1.12 Settings related to UNIX commands

Variable name	Description	Default
MAKE	This variable specifies the make start command. If options are required, specify them together. Normally, there is no need to change the value of this variable from the default value.	make
AR	This variable specifies the command for creating or changing an archive file. If options are required, specify them together. Normally, there is no need to change the value of this variable from the default value.	ar ruv
CP	This variable specifies the command for copying files or directories. If options are required, specify them together. Normally, there is no need to change the value of this variable from the default value.	cp −f
RM	This variable specifies the command for deleting files or directories. If options are required, specify them together. Normally, there is no need to change the value of this variable from the default value.	rm −f
MKDIR	This variable specifies the command for creating directories. If options are required, specify them together. Normally, there is no need to change the value of this variable from the default value.	mkdir −p
MV	This variable specifies the command for moving files or directories. If options are required, specify them together. Normally, there is no need to change the value of this variable from the default value.	mv

1.11.2 Example of Makefile.conf

```
# MPI
MPIDIR     =
MPIBINDIR  =
MPILIBDIR  =
MPIINCDIR  =
MPILIBS    =

# for install option only
PREFIX     = $(HOME)/FrontISTR
BINDIR     = $(PREFIX)/bin
LIBDIR     = $(PREFIX)/lib
INCLUDEDIR = $(PREFIX)/include

# Metis
METISDIR    = $(HOME)/Metis-4.0
METISLIBDIR = $(METISDIR)
METISINCDIR = $(METISDIR)/Lib
```

```
# ParMetis
PARMETISDIR     = $(HOME)/ParMetis-3.1
PARMETISLIBDIR  = $(PARMETISDIR)
PARMETISINCDIR  = $(PARMETISDIR)/ParMETISLib

# Refiner
REFINERDIR      = $(HOME)/REVOCAP_Refiner-1.1.0
REFINERINCDIR   = $(REFINERDIR)/Refiner
REFINERLIBDIR   = $(REFINERDIR)/lib/x86_64-linux

# Coupler
REVOCAPDIR      = $(HOME)/REVOCAP_Coupler-1.6.2
REVOCAPINCDIR   = $(REVOCAPDIR)/librcap
REVOCAPLIBDIR   = $(REVOCAPDIR)/librcap

# MUMPS
MUMPSDIR      = $(HOME)/MUMPS_4.10.0
MUMPSINCDIR   = $(MUMPSDIR)/include
MUMPSLIBDIR   = $(MUMPSDIR)/lib

# ML
MLDIR      = $(HOME)/trilinos/11.8.1/ml
MLINCDIR   = $(MLDIR)/include
MLLIBDIR   = $(MLDIR)/lib

# C compiler settings
CC         = mpiicc
CFLAGS     =
LDFLAGS    = -lm
OPTFLAGS   = -O3
CLINKER    = mpiicc

# C++ compiler settings
CPP          = mpiicpc
CPPFLAGS     = -DMPICH_IGNORE_CXX_SEEK -I$(HOME)/include
CPPLDFLAGS   =
CPPOPTFLAGS  = -O3

# Fortran compiler settings
F90          = mpiifort
F90FLAGS     =
F90LDFLAGS   = -lmkl_intel_lp64 -lmkl_intel_thread -lmkl_core -liomp5
F90OPTFLAGS  = -O2
F90LINKER    = mpiifort
```

1.11.3 Notes on K-computer and Fujitsu FX10

This version includes tuned codes for K-computer and Fujitsu FX10. However, in order to maximize the performance, a part of the source code needs to be changed for corresponding environment.

File to be changed:

hecmw1/src/solver/solver_33/hecmw_tuning_fx.f90

Changes:

Set the value of parameter TotalSectorCacheSize defined in the file to

- 12 on K-computer,
- 24 on FX10.

Initially, the parameter is set for K-computer.

1.12 Appendix

1.12.1 List of `Makefile.conf` Variables

1.12.1.1 Setting related to MPI

Variable name	Description	Default
MPIDIR	This variable specifies the path to the directory where MPI has been installed. If an MPI compatible compiler automatically refers to the path, there is no need to set this variable together with the following variables.	None
MPIBINDIR	This variable specifies the path to the directory where MPI executable files have been installed.	None
MPIINCDIR	This variable specifies the path to the directory where MPI header files have been installed.	.
MPILIBDIR	This variable specifies the path to the directory where MPI libraries have been installed.	.
MPILIBS	This variable specifies the MPI library that will be linked to C and Fortran90 object files.	None

1.12.1.2 Setting related to the installation directory

Variable name	Description	Default
PREFIX	This variable specifies the path to the directory where this software is to be installed.	$(HOME)/FrontISTR
BINDIR	This variable specifies the path to the directory where the executable files of this software are to be installed.	$(PREFIX)/bin
INCLUDEDIR	This variable specifies the path to the directory where the header files of this software are to be installed. Normally, there is no need to change the value of this variable from the default value.	$(PREFIX)/include
LIBDIR	This variable specifies the path to the directory where the libraries of this software are to be installed. Normally, there is no need to change the value of this variable from the default value.	$(PREFIX)/lib

1.12.1.3 Settings related to METIS

Variable name	Description	Default
METISDIR	This variable specifies the path to the directory where METIS has been installed.	$(HOME)/metis
METISINCDIR	This variable specifies the path to the directory where the header files (such as metis.h) of METIS have been installed. Normally, there is no need to change the value of this variable from the default value.	$(METISDIR)/include
METISLIBDIR	This variable specifies the path to the directory where the library (libmetis.a) of METIS has been installed. Normally, ther is no need to change the value of this variable from the default value.	$(METISDIR)/lib

1.12.1.4 Settings related to ParMETIS

Variable name	Description	Default
PARMETISDIR	This variable specifies the path to the directory where ParMETIS has been installed.	$(HOME)/ParMetis
PAEMETIS-INCDIR	This variable specifies the path to the directory where the header files (such as parmetis.h) of ParMETIS have been installed. Normally, there is no need to change the value of this variable from the default value.	$(PARMETISDIR)/include
PARMETISLIB-DIR	This variable specifies the path to the directory where the library (libparmetis.a) of ParMETIS has been installed. Normally, there is no need to change the value of this variable from the default value.	$(PARMETISDIR)/lib

1.12.1.5 Settings related to REVOCAP_Refiner

Variable name	Description	Default
REFINERDIR	This variable specifies the path to the directory where REVOCAP_Refiner has been installed.	$(HOME)/REVOCAP_Refiner
REFINER-INCDIR	This variable specifies the path to the directory where REVOCAP_Refiner header files have been installed. Normally, there is no need to change the value of this variable from the default value.	$(PARMETISDIR)/include
REFINERLIB-DIR	This variable specifies the path to the directory where REVOCAP_Refiner libraries have been installed. Normally, there is no need to change the value of this variable from the default value.	$(PARMETISDIR)/lib

1.12.1.6 Settings related to REVOCAP_Coupler

Variable name	Description	Default
REVOCAPDIR	This variable specifies the path to the directory where REVOCAP_Coupler has been installed.	$(HOME)/REVOCAP_Coupler
REVOCAP-INCDIR	This variable specifies the path to the directory where REVOCAP_Coupler header files have been installed. Normally, there is no need to change the value of this variable from the default value.	$(REVOCAPDIR)/include
REVOCAPLIB-DIR	This variable specifies the path to the directory where REVOCAP_Coupler libraries have been installed. Normally, there is no need to change the value of this variable from the default value.	$(REVOCAPDIR)/lib

1.12.1.7 Settings related to MUMPS

Variable name	Description	Default
MUMPSDIR	This variable specifies the path to the directory where MUMPS has been installed.	$(HOME)/MUMPS
MUMPSINCDIR	This variable specifies the path to the directory where MUMPS header files have been installed. Normally, there is no need to change the value of this variable from the default value.	$(MUMPSDIR)/include
MUMPSLIBDIR	This variable specifies the path to the directory where MUMPS libraries have been installed. Normally, there is no need to change the value of this variable from the default value.	$(MUMPSDIR)/lib

1.12.1.8 Settins related to ML

Variable name	Description	Default
MLDIR	This variable specifies the path to the directory where ML is installed.	$(HOME)/trilinos
MLINCDIR	This variable specivies the path to the directory where ML header files are installed. Normally, there is no need to change the value of this variable from the default value.	$(MLDIR)/include
MLLIBDIR	This variable specivies the path to the directory where ML libraries are installed. Normally, there is no need to change the value of this variable from the default value.	$(MLDIR)/lib

1.12.1.9 Settings related to the C compiler

Variable name	Description	Default
CC	This variable specifies the C compiler start command	mpicc
CFLAGS	This variable specifies the option to be assigned to the C compiler. Normally, there is no need to change the value of this variable from the default value.	
LDFLAGS	This variable specifies the option to be assigned to the C linker. Normally, there is no need to change the value of this variable from the default value. However, when REVOCAP_Refiner (written in C++) is to be used and C compiler is used for linking C programs, C++ standard library (e.g. -lstdc++) needs to be specified here.	−lm
OPTFLAGS	This variable specifies the optimization option (or another option) to be assigned to the C compiler.	−O3
CLINKER	This variable specifies the linker command for C program. This is used when, for example, REVOCAP_Refiner (written in C++) is to be used and C++ compiler needs to be used for linking C programs with C++ libraries.	$(CC)

1.12.1.10 Settings related to the C++ compiler

Variable name	Description	Default
CPP	This variable specifies the C++ compiler start command.	mpic++
CPPFLAGS	This variable specifies the option to be assigned to the C++ compiler. Normally, there is no need to change the value of this variable from the default value. However, if Boost Libraries are not automatically referenced from the C++ compiler, use the -I option to specify the directory that contains the include files.	−DMPICH_IGNORE_CXX_SEEK (Note: This is required for Intel compilers.)
CPPLDFLAGS	This variable specifies the option to be assigned to the C++ linker. Normally, there is no need to change the value of this variable from the default value.	None
CPPOPTFLAGS	This variable specifies the optimization option (or another option) to be assigned to the C++ compiler.	−O3

1.12.1.11 Settings related to Fortran90 compiler

Variable name	Description	Default
F90	This variable specifies the Fortran90 compiler start command.	mpif90

Variable name	Description	Default
F90FLAGS	This variable specifies the option to be assigned to the Fortran90 compiler. Normally, there is no need to change the value of this variable from the default value.	−DMPICH_IGNORE_CXX_SEEK
F90LDFLAGS	This variable specifies the option to be assigned to the Fortran90 linker. Normally, there is no need to change the value of this variable from the default value. However, if Intel MKL is used, specify its link option. Also, when REVOCAP_Refiner (written in C++) is to be used and Fortran90 compiler is used for linking Fortran90 programs, C++ standard library (e.g. -lstdc++) needs to be specified here.	None
F90OPTFLAGS	This variable specifies the optimization option (or another option) to be assigned to the Fortran90 compiler.	−O2
F90LINKER	This variable specifies the linker command for Fortran90 program. This is used when, for example, REVOCAP_Refiner (written in C++) is to be used and C++ compiler needs to be used for linking Fortran90 programs with C++ libraries. (E.g. on K-computer, "mpiFCCpx –linkfortran" needs to be specified.)	$(F90)

1.12.1.12 Settings related to UNIX commands

Variable name	Description	Default
MAKE	This variable specifies the make start command. If options are required, specify them together. Normally, there is no need to change the value of this variable from the default value.	make
AR	This variable specifies the command for creating or changing an archive file. If options are required, specify them together. Normally, there is no need to change the value of this variable from the default value.	ar ruv
CP	This variable specifies the command for copying files or directories. If options are required, specify them together. Normally, there is no need to change the value of this variable from the default value.	cp −f
RM	This variable specifies the command for deleting files or directories. If options are required, specify them together. Normally, there is no need to change the value of this variable from the default value.	rm −f
MKDIR	This variable specifies the command for creating directories. If options are required, specify them together. Normally, there is no need to change the value of this variable from the default value.	mkdir −p
MV	This variable specifies the command for moving files or directories. If options are required, specify them together. Normally, there is no need to change the value of this variable from the default value.	mv

1.12.2 Example of Makefile.conf

```
# MPI
MPIDIR      =
MPIBINDIR   =
MPILIBDIR   =
```

```
MPIINCDIR =
MPILIBS   =

# for install option only
PREFIX      = $(HOME)/FrontISTR
BINDIR      = $(PREFIX)/bin
LIBDIR      = $(PREFIX)/lib
INCLUDEDIR = $(PREFIX)/include

# Metis
METISDIR     = $(HOME)/Metis-4.0
METISLIBDIR = $(METISDIR)
METISINCDIR = $(METISDIR)/Lib

# ParMetis
PARMETISDIR     = $(HOME)/ParMetis-3.1
PARMETISLIBDIR = $(PARMETISDIR)
PARMETISINCDIR = $(PARMETISDIR)/ParMETISLib

# Refiner
REFINERDIR     = $(HOME)/REVOCAP_Refiner-1.1.0
REFINERINCDIR = $(REFINERDIR)/Refiner
REFINERLIBDIR = $(REFINERDIR)/lib/x86_64-linux

# Coupler
REVOCAPDIR     = $(HOME)/REVOCAP_Coupler-1.6.2
REVOCAPINCDIR = $(REVOCAPDIR)/librcap
REVOCAPLIBDIR = $(REVOCAPDIR)/librcap

# MUMPS
MUMPSDIR     = $(HOME)/MUMPS_4.10.0
MUMPSINCDIR = $(MUMPSDIR)/include
MUMPSLIBDIR = $(MUMPSDIR)/lib

# ML
MLDIR     = $(HOME)/trilinos/11.8.1/ml
MLINCDIR = $(MLDIR)/include
MLLIBDIR = $(MLDIR)/lib

# C compiler settings
CC       = mpiicc
CFLAGS   =
LDFLAGS  = -lm
OPTFLAGS = -O3
CLINKER  = mpiicc

# C++ compiler settings
CPP          = mpiicpc
CPPFLAGS     = -DMPICH_IGNORE_CXX_SEEK -I$(HOME)/include
CPPLDFLAGS   =
CPPOPTFLAGS = -O3

# Fortran compiler settings
F90          = mpiifort
F90FLAGS     =
F90LDFLAGS   = -lmkl_intel_lp64 -lmkl_intel_thread -lmkl_core -liomp5
F90OPTFLAGS = -O2
F90LINKER    = mpiifort
```

1.12.3 Notes on K-computer and Fujitsu FX10

This version includes tuned codes for K-computer and Fujitsu FX10. However, in order to maximize the performance, a part of the source code needs to be changed for corresponding environment.

File to be changed:

hecmw1/src/solver/solver_33/hecmw_tuning_fx.f90

Changes:

Set the value of parameter TotalSectorCacheSize defined in the file to

- 12 on K-computer,
- 24 on FX10.

Initially, the parameter is set for K-computer.

1.13 Appendix : Example of installation procedure to CentOS7.6(cmake)

We will explain how to install this software and how to build external libraries required this software on CentOS7.6.

More information for building each libraries, refer to their installation manuals.

1.13.1 Preparation

At first, install the basic build toolchains and libraries as follows.

```
$ su
# yum group mark install "Development Tools"
# yum update
# yum install openmpi-devel cmake
# exit
```

Configure the MPI environment settings as follows.

```
$ module purge
$ module local mpi/openmpi-x86_64
```

If you write the above two lines in $HOME/.bash_profile, the settings will be reflected even after the next login.

Please check compilers and compiler wrappers for MPI works propery as follows.

```
$ which gcc g++ gfortran mpicc mpic++ mpifort
/usr/bin/gcc
/usr/bin/g++
/usr/bin/gfortran
/usr/lib64/openmpi/bin/mpicc
/usr/lib64/openmpi/bin/mpic++
/usr/lib64/openmpi/bin/mpifort
```

1.13.2 Installing libraries

Compile and install required libraries this software. Working directory is $HOME/work, destination directory for install is $HOME/local.

And add $HOME/local/bin to PATH environment variable as follows.

```
$ cd $HOME
$ mkdir work
$ mkdir -p local/bin local/lib local/include
$ export PATH=$HOME/local/bin:$PATH
```

1.13.2.1 Downloads

Downloads the following software and save it to working directory $HOME/work.

Software	Link
REVOCAP_Refiner-1.1.04.tar.gz	https://www.frontistr.com/
FrontISTR_V50.tar.gz	http://www.frontistr.com/
OpenBLAS-0.2.20.tar.gz	http://www.openblas.net/
metis-5.1.0.tar.gz	http://glaros.dtc.umn.edu/gkhome/metis/metis/download
scalapack-2.0.2.tgz	http://www.netlib.org/scalapack/
MUMPS_5.1.2.tar.gz	http://mumps.enseeiht.fr/
trilinos-12.14.1-Source.tar.bz2	https://trilinos.org/download/

```
$ cd $HOME/work
$ tar xvf REVOCAP_Refiner-1.1.04.tar.gz
$ cd REVOCAP_Refiner-1.1.04
$ make
$ cp lib/x86_64-linux/libRcapRefiner.a $HOME/local/lib
$ cp Refiner/rcapRefiner.h $HOME/local/include
```

```
$ cd $HOME/work
$ tar xvf OpenBLAS-0.2.20.tar.gz
$ make BINARY=64 NO_SHARED=1 USE_OPENMP=1
$ make PREFIX=$HOME/local install
```

```
$ cd $HOME/work
$ tar xvf metis-5.1.0.tar.gz
$ cd metis-5.1.0
$ make config prefix=~/local cc=gcc openmp=1
$ make
$ make install
```

```
$ cd $HOME/work
$ tar xvf scalapack-2.0.2.tgz
$ cd scalapack-2.0.2
$ mkdir build
$ cmake -DCMAKE_INSTALL_PREFIX=$HOME/local \
        -DCMAKE_EXE_LINKER_FLAGS="-fopenmp" \
        -DCMAKE_BUILD_TYPE="Release" \
        -DBLAS_LIBRARIES=$HOME/local/lib/libopenblas.a \
        -DLAPACK_LIBRARIES=$HOME/local/lib/libopenblas.a \
        ..
$ make
$ make install
```

```
$ cd $HOME/work
$ tar xvf MUMPS_5.1.2.tar.gz
$ cd MUMPS_5.1.2
$ cp Make.inc/Makefile.inc.generic Makefile.inc
```

Change the following parts of copied Makefile.inc.

```
$ vi Makefile.inc
$ cp Make.inc/Makefile.inc.generic Makefile.inc
$ vi Makefile.inc
LMETISDIR = $(HOME)/local
IMETIS    = -I$(LMETISDIR)/include
LMETIS    = -L$(LMETISDIR)/lib -lmetis

ORDERINGSF  = -Dmetis -Dpord

CC      = mpicc
FC      = mpifort
FL      = mpifort

LAPACK = -L$(HOME)/local/lib -lopenblas

SCALAP  = -L$(HOME)/local/lib -lscalapack

INCPAR = -I/usr/include/openmpi-x86_64

LIBPAR  = $(SCALAP) -L/usr/lib64/openmpi/lib -lmpi

LIBBLAS = -L$(HOME)/local/lib -lopenblas

OPTF    = -O -DBLR_MT -fopenmp
OPTC    = -O -I. -fopenmp
OPTL    = -O -fopenmp
```

Then execute make.

```
$ make
$ cp lib/*.a $HOME/local/lib
$ cp include/*.h $HOME/local/include
```

```
$ cd $HOME/work
$ tar xvf trilinos-12.14.1-Source.tar.gz
$ cd trilinos-12.14.1-Source
$ mkdir build
$ cmake -DCMAKE_INSTALL_PREFIX=$HOME/local \
        -DCMAKE_C_COMPILER=mpicc \
        -DCMAKE_CXX_COMPILER=mpic++ \
        -DCMAKE_Fortran_COMPILER=mpifort \
        -DTPL_ENABLE_MPI=ON \
        -DTPL_ENABLE_LAPACK=ON \
        -DTPL_ENABLE_SCALAPACK=ON \
        -DTPL_ENABLE_METIS=ON \
        -DTPL_ENABLE_MUMPS=ON \
        -DTPL_MUMPS_INCLUDE_DIRS=$HOME/local/include \
        -DTrilinos_ENABLE_ML=ON \
        -DTrilinos_ENABLE_Zoltan=ON \
        -DTrilinos_ENABLE_OpenMP=ON \
        -DTrilinos_ENABLE_Amesos=ON \
        -DTrilinos_ENABLE_ALL_OPTIONAL_PACKAGES=OFF \
        -DBLAS_LIBRARY_DIRS=$HOME/local/lib \
        -DLAPACK_LIBRARY_DIRS=$HOME/local/lib" \
        -DSCALAPACK_LIBRARY_DIRS=$HOME/local/lib" \
```

```
        -DBLAS_LIBRARY_NAMES="openblas" \
        -DLAPACK_LIBRARY_NAMES="openblas" \
        -DSCALAPACK_LIBRARY_NAMES="scalapack" \
        ..
$ make
$ make install
```

1.13.3 Compiling FrontISTR

Finishing compiling above libraries, compile FrontISTR.

```
$ cd $HOME/work/FrontISTR
$ mkdir build
$ cd build
$ cmake -DCMAKE_INSTALL_PREFIX=$HOME/FrontISTR \
        -DWITH_ML=ON \
        -DBLAS_LIBRARIES=$HOME/local/lib/libopenblas.a \
        -DLAPACK_LIBRARIES=$HOME/local/lib/libopenblas.a \
        ..
```

1.13.3.1 Executing make Execute make command.

```
$ make
```

When execute make simultaneously, specify option -j.

```
$ make -j4
```

Reduce compile time when increase the number of simultaneous.

```
$ make install
```

FrontISTR will be installed to $(HOME)/FrontISTR/bin.

1.13.3.3 Testing FrontISTR Run sample case in the tutorial directory and check running of FrontISTR.

```
$ cd $HOME/work/FrontISTR/tutorial
$ cd 01_elastic_hinge
$ $HOME/FrontISTR/bin/fistr1
 Step control not defined! Using default step=1
 fstr_setup: OK
 Start visualize PSF 1 at timestep 0

 loading step=     1
 sub_step= 1,    current_time=  0.0000E+00, time_inc=  0.1000E+01
 loading_factor=     0.0000000    1.0000000
### 3x3 BLOCK CG, SSOR, 1
       1     1.903375E+00
       2     1.974378E+00
       3     2.534627E+00
       4     3.004045E+00
       5     3.202633E+00
       6     3.203864E+00
```

```
...
...
```

When finished analysis, displayed message as follows.

```
...
...
   2966     1.143085E-08
   2967     1.078272E-08
   2968     1.004759E-08
   2969     9.372882E-09
### Relative residual = 9.39169E-09

### summary of linear solver
      2969 iterations       9.391687E-09
     set-up time      :     4.108060E-01
     solver time      :     6.506822E+01
     solver/comm time :     4.342469E-01
     solver/matvec    :     1.923199E+01
     solver/precond   :     2.688405E+01
     solver/1 iter    :     2.191587E-02
     work ratio (%)   :     9.933263E+01

 Start visualize PSF 1 at timestep 1
### FSTR_SOLVE_NLGEOM FINISHED!

 ================================
    TOTAL TIME (sec) :      74.93
          pre (sec) :       1.86
        solve (sec) :      73.07
 ================================
 FrontISTR Completed !!
```

1.14 Appendix : Example of installation procedure to CentOS7.6 (Makefile.conf)

We will explain how to install this software and how to build external libraries required this software on CentOS7.6.

More information for building each libraries, refer to their installation manuals.

1.14.1 Preparation

At first, install the basic build toolchains and libraries as follows.

```
$ su
# yum group mark install "Development Tools"
# yum update
# yum install openmpi-devel cmake
# exit
```

Configure the MPI environment settings as follows.

```
$ module purge
$ module local mpi/openmpi-x86_64
```

If you write the above two lines in $HOME/.bash_profile, the settings will be reflected even after the next login.

Please check compilers and compiler wrapper for MPI works propery as follow.

```
$ which gcc g++ gfortran mpicc mpic++ mpifort
/usr/bin/gcc
/usr/bin/g++
/usr/bin/gfortran
/usr/lib64/openmpi/bin/mpicc
/usr/lib64/openmpi/bin/mpic++
/usr/lib64/openmpi/bin/mpifort
```

1.14.2 Installing libraries

Compile and install required libraries this software. Working directory is $HOME/work, destination directory for install is $HOME/local.

And add $HOME/local/bin to PATH environment variable as follows.

```
$ cd $HOME
$ mkdir work
$ mkdir -p local/bin local/lib local/include
$ export PATH=$HOME/local/bin:$PATH
```

1.14.2.1 Downloads Downloads the following software and save it to working directory $HOME/work.

Software	Link
REVOCAP_Refiner-1.1.04.tar.gz	https://www.frontistr.com/
FrontISTR_V50.tar.gz	https://www.frontistr.com/
OpenBLAS-0.2.20.tar.gz	http://www.openblas.net/
metis-5.1.0.tar.gz	http://glaros.dtc.umn.edu/gkhome/metis/metis/download
scalapack-2.0.2.tgz	http://www.netlib.org/scalapack/
MUMPS_5.1.2.tar.gz	http://mumps.enseeiht.fr/
trilinos-12.14.1-Source.tar.bz2	https://trilinos.org/download/

```
$ cd $HOME/work
$ tar xvf REVOCAP_Refiner-1.1.04.tar.gz
$ cd REVOCAP_Refiner-1.1.04
$ make
$ cp lib/x86_64-linux/libRcapRefiner.a $HOME/local/lib
$ cp Refiner/rcapRefiner.h $HOME/local/include
```

```
$ cd $HOME/work
$ tar xvf OpenBLAS-0.2.20.tar.gz
$ make BINARY=64 NO_SHARED=1 USE_OPENMP=1
$ make PREFIX=$HOME/local install
```

```
$ cd $HOME/work
$ tar xvf metis-5.1.0.tar.gz
$ cd metis-5.1.0
$ make config prefix=$HOME/local cc=gcc openmp=1
$ make
$ make install
```

```
$ cd $HOME/work
$ tar xvf scalapack-2.0.2.tgz
$ cd scalapack-2.0.2
$ mkdir build
$ cmake -DCMAKE_INSTALL_PREFIX=$HOME/local \
        -DCMAKE_EXE_LINKER_FLAGS="-fopenmp" \
        -DBLAS_LIBRARIES=$HOME/local/lib/libopenblas.a \
        -DLAPACK_LIBRARIES=$HOME/local/lib/libopenblas.a \
        ..
$ make
$ make install
```

```
$ cd $HOME/work
$ tar xvf MUMPS_5.1.2.tar.gz
$ cd MUMPS_5.1.2
$ cp Make.inc/Makefile.inc.generic Makefile.inc
```

Change the following parts of copied Makefile.inc

```
$ vi Makefile.inc
$ cp Make.inc/Makefile.inc.generic Makefile.inc
$ vi Makefile.inc
LMETISDIR = $(HOME)/local
IMETIS    = -I$(LMETISDIR)/include
LMETIS    = -L$(LMETISDIR)/lib -lmetis

ORDERINGSF  = -Dmetis -Dpord

CC      = mpicc -fopenmp
FC      = mpifort -fopenmp
FL      = mpifort -fopenmp

LAPACK = -L$(HOME)/local/lib -lopenblas

SCALAP  = -L$(HOME)/local/lib -lscalapack

INCPAR = -I/usr/include/openmpi-x86_64

LIBPAR  = $(SCALAP) -L/usr/lib64/openmpi/lib -lmpi

LIBBLAS = -L$(HOME)/local/lib -lopenblas

OPTF    = -O -DBLR_MT -fopenmp
OPTC    = -O -I. -fopenmp
OPTL    = -O -fopenmp
```

Then execute make.

```
$ make
$ cp lib/*.a $HOME/local/lib
$ cp include/*.h $HOME/local/include
```

```
$ cd $HOME/work
$ tar xvf trilinos-12.14.1-Source.tar.gz
$ cd trilinos-12.14.1-Source
```

```
$ mkdir build
$ cmake -DCMAKE_INSTALL_PREFIX=$HOME/local \
        -DCMAKE_C_COMPILER=mpicc \
        -DCMAKE_CXX_COMPILER=mpic++ \
        -DCMAKE_Fortran_COMPILER=mpifort \
        -DTPL_ENABLE_MPI=ON \
        -DTPL_ENABLE_LAPACK=ON \
        -DTPL_ENABLE_SCALAPACK=ON \
        -DTPL_ENABLE_METIS=ON \
        -DTPL_ENABLE_MUMPS=ON \
        -DTPL_MUMPS_INCLUDE_DIRS=$HOME/local/include \
        -DTrilinos_ENABLE_ML=ON \
        -DTrilinos_ENABLE_Zoltan=ON \
        -DTrilinos_ENABLE_OpenMP=ON \
        -DTrilinos_ENABLE_Amesos=ON \
        -DTrilinos_ENABLE_ALL_OPTIONAL_PACKAGES=OFF \
        -DBLAS_LIBRARY_DIRS=$HOME/local/lib \
        -DLAPACK_LIBRARY_DIRS=$HOME/local/lib" \
        -DSCALAPACK_LIBRARY_DIRS=$HOME/local/lib" \
        -DBLAS_LIBRARY_NAMES="openblas" \
        -DLAPACK_LIBRARY_NAMES="openblas" \
        -DSCALAPACK_LIBRARY_NAMES="scalapack" \
        ..
$ make
$ make install
```

1.14.3 Compiling FrontISTR

Finishing compiling above libraries, compile FrontISTR.

```
$ cd $HOME/work
$ tar xvf FrontISTR_V50.tar.gz
$ cd FrontISTR
```

1.14.3.1 Editing Makefile.conf Copy template as Makefile.conf.org to Makfile.conf. Then edit Makefile.conf as follows.

```
$ cp Makefile.conf.org Makefile.conf
$ vi Makefile.conf
############################################################
#                                                          #
#     Setup Configulation File for FrontISTR               #
#                                                          #
############################################################

# MPI
MPIDIR          = /usr/lib64/openmpi
MPIBINDIR       = $(MPIDIR)/bin
MPILIBDIR       = $(MPIDIR)/lib
MPIINCDIR       = /usr/include/openmpi-x86_64
MPILIBS         = -lmpi -lmpi_cxx -lmpi_mpifh

# for install option only
PREFIX          = $(HOME)/FrontISTR
```

```
BINDIR          = $(PREFIX)/bin
LIBDIR          = $(PREFIX)/lib
INCLUDEDIR      = $(PREFIX)/include

# Metis
METISDIR        = $(HOME)/local
METISLIBDIR     = $(METISDIR)/lib
METISINCDIR     = $(METISDIR)/include
HECMW_METIS_VER= 5

# ParMetis
PARMETISDIR     = $(HOME)/local
PARMETISLIBDIR = $(PARMETISDIR)/lib
PARMETISINCDIR = $(PARMETISDIR)/include

# Refiner
REFINERDIR      = $(HOME)/local
REFINERINCDIR  = $(REFINERDIR)/include
REFINERLIBDIR  = $(REFINERDIR)/lib

# Coupler
REVOCAPDIR      = $(HOME)/local
REVOCAPINCDIR  = $(REVOCAPDIR)/include
REVOCAPLIBDIR  = $(REVOCAPDIR)/lib

# MUMPS
MUMPSDIR        = $(HOME)/local
MUMPSINCDIR     = $(MUMPSDIR)/include
MUMPSLIBDIR     = $(MUMPSDIR)/lib
MUMPSLIBS       = -ldmumps -lmumps_common -lpord -L$HOME/local/lib -lscalapack

# MKL PARDISO
MKLDIR      = $(HOME)/
MKLINCDIR   = $(MKLDIR)/include
MKLLIBDIR   = $(MKLDIR)/lib

# ML
MLDIR           = $(HOME)/local
MLINCDIR        = $(MLDIR)/include
MLLIBDIR        = $(MLDIR)/lib
MLLIBS          = -lml -lamesos -ltrilinosss -lzoltan -lepetra -lteuchosremainder -lteuchosnu
# C compiler settings
CC              = mpicc -fopenmp
CFLAGS          =
LDFLAGS         = -lstdc++ -lm
OPTFLAGS        = -O3

# C++ compiler settings
CPP             = mpic++ -fopenmp
CPPFLAGS        =
CPPLDFLAGS      =
CPPOPTFLAGS     = -O3

# Fortran compiler settings
F90             = mpif90 -fopenmp
F90FLAGS        =
F90LDFLAGS      = -lstdc++ -L$(HOME)/local/lib -lopenblas
F90OPTFLAGS     = -O2
F90FPP          = -cpp
```

```
F90LINKER       = mpif90 -fopenmp

MAKE            = make
AR              = ar ruv
MV              = mv -f
CP              = cp -f
RM              = rm -f
MKDIR           = mkdir -p
```

1.14.3.2 Executing setup.sh Finished to edit Makefile.conf, then execute setup.sh.

```
$ ./setup.sh -p --with-tools --with-refiner \
             --with-metis --with-mumps --with-lapack --with-ml
```

1.14.3.3 Executing make Execute make command.

```
$ make
```

```
$ make install
```

FrontISTR will be installed to $(HOME)/FrontISTR/bin.

1.14.3.5 Testing FrontISTR Run sample case in the tutorial directory and check running of FrontISTR.

```
$ cd $HOME/work/FrontISTR/tutorial
$ cd 01_elastic_hinge
$ $HOME/FrontISTR/bin/fistr1
 Step control not defined! Using default step=1
 fstr_setup: OK
 Start visualize PSF 1 at timestep 0

 loading step=     1
 sub_step= 1,    current_time=  0.0000E+00, time_inc=  0.1000E+01
 loading_factor=     0.0000000    1.0000000
### 3x3 BLOCK CG, SSOR, 1
       1     1.903375E+00
       2     1.974378E+00
       3     2.534627E+00
       4     3.004045E+00
       5     3.202633E+00
       6     3.203864E+00
...
...
```

When finished analysis, displayed message as follows.

```
...
...
    2966     1.143085E-08
    2967     1.078272E-08
    2968     1.004759E-08
    2969     9.372882E-09
### Relative residual = 9.39169E-09

### summary of linear solver
```

```
      2969 iterations       9.391687E-09
    set-up time       :     4.108060E-01
    solver time       :     6.506822E+01
    solver/comm time  :     4.342469E-01
    solver/matvec     :     1.923199E+01
    solver/precond    :     2.688405E+01
    solver/1 iter     :     2.191587E-02
    work ratio (%)    :     9.933263E+01

 Start visualize PSF 1 at timestep 1
### FSTR_SOLVE_NLGEOM FINISHED!

 ========================================
    TOTAL TIME (sec) :       74.93
          pre (sec) :        1.86
        solve (sec) :       73.07
 ========================================
 FrontISTR Completed !!
```

1.15 Appendix : Example of installation procedure to Ubuntu18.04(cmake)

We will explain how to install this software and how to build external libraries required this software on Ubuntu18.04.

More information for building each libraries, refer to their installation manuals.

1.15.1 Preparation

At first, install the basic build toolchains and libraries as follows.

```
$ sudo apt install build-essential gfortran cmake openmpi-bin libopenmpi-dev
```

Please check compilers and compiler wrappers for MPI works propery.

```
$ which gcc g++ gfortran mpicc mpic++ mpifort
/usr/bin/gcc
/usr/bin/g++
/usr/bin/gfortran
/usr/bin/mpicc
/usr/bin/mpic++
/usr/bin/mpifort
```

1.15.2 Installing libraries

Compile and install required libraries this software. Working directory is $HOME/work, destination directory for install is $HOME/local.

And add $HOME/local/bin to PATH environment variable as follows.

```
$ cd $HOME
$ mkdir work
$ mkdir -p local/bin local/lib local/include
$ export PATH=$HOME/local/bin:$PATH
```

Software	Link

1.15.2.1 Downloads

Downloads the following software and save it to working directory $HOME/work.

Software	Link
REVOCAP_Refiner-1.1.04.tar.gz	https://www.frontistr.com/
FrontISTR_V50.tar.gz	https://www.frontistr.com/
OpenBLAS-0.2.20.tar.gz	http://www.openblas.net/
metis-5.1.0.tar.gz	http://glaros.dtc.umn.edu/gkhome/metis/metis/download
scalapack-2.0.2.tgz	http://www.netlib.org/scalapack/
MUMPS_5.1.2.tar.gz	http://mumps.enseeiht.fr/
trilinos-12.14.1-Source.tar.bz2	https://trilinos.org/download/

```
$ cd $HOME/work
$ tar xvf REVOCAP_Refiner-1.1.04.tar.gz
$ cd REVOCAP_Refiner-1.1.04
$ make
$ cp lib/x86_64-linux/libRcapRefiner.a $HOME/local/lib
$ cp Refiner/rcapRefiner.h $HOME/local/include
```

```
$ cd $HOME/work
$ tar xvf OpenBLAS-0.2.20.tar.gz
$ make BINARY=64 NO_SHARED=1 USE_OPENMP=1
$ make PREFIX=$HOME/local install
```

```
$ cd $HOME/work
$ tar xvf metis-5.1.0.tar.gz
$ cd metis-5.1.0
$ make config prefix=$HOME/local cc=gcc openmp=1
$ make
$ make install
```

```
$ cd $HOME/work
$ tar xvf scalapack-2.0.2.tgz
$ cd scalapack-2.0.2
$ mkdir build
$ cmake -DCMAKE_INSTALL_PREFIX=$HOME/local \
        -DCMAKE_EXE_LINKER_FLAGS="-fopenmp" \
        -DBLAS_LIBRARIES=$HOME/local/lib/libopenblas.a \
        -DLAPACK_LIBRARIES=$HOME/local/lib/libopenblas.a \
        ..
$ make
$ make install
```

```
$ cd $HOME/work
$ tar xvf MUMPS_5.1.2.tar.gz
$ cd MUMPS_5.1.2
$ cp Make.inc/Makefile.inc.generic Makefile.inc
```

Change the following parts of copied Makefile.inc.

```
$ vi Makefile.inc
$ cp Make.inc/Makefile.inc.generic Makefile.inc
$ vi Makefile.inc
LMETISDIR = $(HOME)/local
IMETIS    = -I$(LMETISDIR)/include
LMETIS    = -L$(LMETISDIR)/lib -lmetis

ORDERINGSF  = -Dmetis -Dpord

CC      = mpicc -fopenmp
FC      = mpifort -fopenmp
FL      = mpifort -fopenmp

LAPACK = -L$(HOME)/local/lib -lopenblas

SCALAP  = -L$(HOME)/local/lib -lscalapack

INCPAR =

LIBPAR  = $(SCALAP)

LIBBLAS = -L$(HOME)/local/lib -lopenblas
```

Then execute make.

```
$ make
$ cp lib/*.a $HOME/local/lib
$ cp include/*.h $HOME/local/include
```

```
$ cd $HOME/work
$ tar xvf trilinos-12.14.1-Source.tar.gz
$ cd trilinos-12.14.1-Source
$ mkdir build
$ cmake -DCMAKE_INSTALL_PREFIX=$HOME/local \
        -DCMAKE_C_COMPILER=mpicc \
        -DCMAKE_CXX_COMPILER=mpic++ \
        -DCMAKE_Fortran_COMPILER=mpifort \
        -DTPL_ENABLE_MPI=ON \
        -DTPL_ENABLE_LAPACK=ON \
        -DTPL_ENABLE_SCALAPACK=ON \
        -DTPL_ENABLE_METIS=ON \
        -DTPL_ENABLE_MUMPS=ON \
        -DTrilinos_ENABLE_ML=ON \
        -DTrilinos_ENABLE_Zoltan=ON \
        -DTrilinos_ENABLE_OpenMP=ON \
        -DTrilinos_ENABLE_Amesos=ON \
        -DTrilinos_ENABLE_ALL_OPTIONAL_PACKAGES=OFF \
        -DBLAS_LIBRARY_DIRS=$HOME/local/lib \
        -DLAPACK_LIBRARY_DIRS=$HOME/local/lib" \
        -DSCALAPACK_LIBRARY_DIRS=$HOME/local/lib" \
```

```
        -DBLAS_LIBRARY_NAMES="openblas" \
        -DLAPACK_LIBRARY_NAMES="openblas" \
        -DSCALAPACK_LIBRARY_NAMES="scalapack" \
        ..
$ make
$ make install
```

1.15.3 Compiling FrontISTR

Finishing compiling above libraeies, compile FrontISTR.

```
$ cd $HOME/work/FrontISTR
$ mkdir build
$ cd build
$ cmake -DCMAKE_INSTALL_PREFIX=$HOME/FrontISTR \
        -DBLAS_LIBRARIES=$HOME/local/lib/libopenblas.a \
        -DLAPACK_LIBRARIES=$HOME/local/lib/libopenblas.a \
        ..
```

```
$ make
```

When execute make simultaneously, specify option -j.

```
$ make -j4
```

Reduce compile time when increase the number of simultaneous.

```
$ make install
```

FrontISTR will be installed to $(HOME)/FrontISTR/bin.

1.15.3.3 Testing FrontISTR

Run sample case in the tutorial directory and check running of FrontISTR.

```
$ cd $HOME/work/FrontISTR/tutorial
$ cd 01_elastic_hinge
$ $HOME/FrontISTR/bin/fistr1
 Step control not defined! Using default step=1
 fstr_setup: OK
 Start visualize PSF 1 at timestep 0

 loading step=     1
 sub_step= 1,    current_time=  0.0000E+00, time_inc=  0.1000E+01
 loading_factor=     0.0000000    1.0000000
### 3x3 BLOCK CG, SSOR, 1
       1     1.903375E+00
       2     1.974378E+00
       3     2.534627E+00
       4     3.004045E+00
       5     3.202633E+00
       6     3.203864E+00
...
...
```

When finished analysis, displayed message as follows.

```
...
...
   2966     1.143085E-08
   2967     1.078272E-08
   2968     1.004759E-08
   2969     9.372882E-09
### Relative residual = 9.39169E-09

### summary of linear solver
      2969 iterations       9.391687E-09
     set-up time      :     4.108060E-01
     solver time      :     6.506822E+01
     solver/comm time :     4.342469E-01
     solver/matvec    :     1.923199E+01
     solver/precond   :     2.688405E+01
     solver/1 iter    :     2.191587E-02
     work ratio (%)   :     9.933263E+01

 Start visualize PSF 1 at timestep 1
### FSTR_SOLVE_NLGEOM FINISHED!

 ========================================
    TOTAL TIME (sec) :       74.93
          pre (sec) :        1.86
        solve (sec) :       73.07
 ========================================
 FrontISTR Completed !!
```

1.16 Appendix : Example of installation procedure to Ubuntu18.04(Makefile.conf)

We will explain how to install this software and how to build external libraries required this software on Ubuntu18.04.

More information for building each libraries, refer to their installation manuals.

1.16.1 Preparation

At first, install the basic build toolchains and libraries as follows.

```
$ sudo apt install build-essential gfortran cmake openmpi-bin libopenmpi-dev
```

Please check compilers and compiler wrappers for MPI works propery.

```
$ which gcc g++ gfortran mpicc mpic++ mpifort
/usr/bin/gcc
/usr/bin/g++
/usr/bin/gfortran
/usr/bin/mpicc
/usr/bin/mpic++
/usr/bin/mpifort
```

1.16.2 Installing libraries

Compile and install required libraries this software. Working directory is $HOME/work, destination directory for install is $HOME/local.

And add $HOME/local/bin to PATH environment variable as follows.

```
$ cd $HOME
$ mkdir work
$ mkdir -p local/bin local/lib local/include
$ export PATH=$HOME/local/bin:$PATH
```

1.16.2.1 Downloads Downloads the following software and save it to working directory $HOME/work.

Software	Link
REVOCAP_Refiner-1.1.04.tar.gz	https://www.frontistr.com/
FrontISTR_V50.tar.gz	https://www.frontistr.com/
OpenBLAS-0.2.20.tar.gz	http://www.openblas.net/
metis-5.1.0.tar.gz	http://glaros.dtc.umn.edu/gkhome/metis/metis/download
scalapack-2.0.2.tgz	http://www.netlib.org/scalapack/
MUMPS_5.1.2.tar.gz	http://mumps.enseeiht.fr/
trilinos-12.14.1-Source.tar.bz2	https://trilinos.org/download/

```
$ cd $HOME/work
$ tar xvf REVOCAP_Refiner-1.1.04.tar.gz
$ cd REVOCAP_Refiner-1.1.04
$ make
$ cp lib/x86_64-linux/libRcapRefiner.a $HOME/local/lib
$ cp Refiner/rcapRefiner.h $HOME/local/include
```

```
$ cd $HOME/work
$ tar xvf OpenBLAS-0.2.20.tar.gz
$ make BINARY=64 NO_SHARED=1 USE_OPENMP=1
$ make PREFIX=$HOME/local install
```

```
$ cd $HOME/work
$ tar xvf metis-5.1.0.tar.gz
$ cd metis-5.1.0
$ make config prefix=$HOME/local cc=gcc openmp=1
$ make
$ make install
```

```
$ cd $HOME/work
$ tar xvf scalapack-2.0.2.tgz
$ cd scalapack-2.0.2
$ mkdir build
$ cmake -DCMAKE_INSTALL_PREFIX=$HOME/local \
        -DCMAKE_EXE_LINKER_FLAGS="-fopenmp" \
        -DWITH_MIL=ON \
        -DBLAS_LIBRARIES=$HOME/local/lib/libopenblas.a \
        -DLAPACK_LIBRARIES=$HOME/local/lib/libopenblas.a \
        ..
$ make
$ make install
```

```
$ cd $HOME/work
$ tar xvf MUMPS_5.1.2.tar.gz
$ cd MUMPS_5.1.2
$ cp Make.inc/Makefile.inc.generic Makefile.inc
```

Change the following parts of copied Makefile.inc.

```
$ vi Makefile.inc
$ cp Make.inc/Makefile.inc.generic Makefile.inc
$ vi Makefile.inc
LMETISDIR = $(HOME)/local
IMETIS    = -I$(LMETISDIR)/include
LMETIS    = -L$(LMETISDIR)/lib -lmetis

ORDERINGSF  = -Dmetis -Dpord

CC      = mpicc
FC      = mpifort
FL      = mpifort

LAPACK = -L$(HOME)/local/lib -lopenblas

SCALAP  = -L$(HOME)/local/lib -lscalapack

INCPAR =

LIBPAR  = $(SCALAP)

LIBBLAS = -L$(HOME)/local/lib -lopenblas

OPTF    = -O -DBLR_ML -fopenmp
OPTC    = -O -I. -fopenmp
OPTL    = -O -fopenmp
```

Then execute make.

```
$ make
$ cp lib/*.a $HOME/local/lib
$ cp include/*.h $HOME/local/include
```

```
$ cd $HOME/work
$ tar xvf trilinos-12.14.1-Source.tar.gz
$ cd trilinos-12.14.1-Source
$ mkdir build
$ cmake -DCMAKE_INSTALL_PREFIX=$HOME/local \
        -DCMAKE_C_COMPILER=mpicc \
        -DCMAKE_CXX_COMPILER=mpic++ \
        -DCMAKE_Fortran_COMPILER=mpifort \
        -DTPL_ENABLE_MPI=ON \
        -DTPL_ENABLE_LAPACK=ON \
        -DTPL_ENABLE_SCALAPACK=ON \
        -DTPL_ENABLE_METIS=ON \
        -DTPL_ENABLE_MUMPS=ON \
        -DTrilinos_ENABLE_ML=ON \
        -DTrilinos_ENABLE_Zoltan=ON \
        -DTrilinos_ENABLE_OpenMP=ON \
        -DTrilinos_ENABLE_Amesos=ON \
```

```
        -DTrilinos_ENABLE_ALL_OPTIONAL_PACKAGES=OFF \
        -DBLAS_LIBRARY_DIRS=$HOME/local/lib \
        -DLAPACK_LIBRARY_DIRS=$HOME/local/lib" \
        -DSCALAPACK_LIBRARY_DIRS=$HOME/local/lib" \
        -DBLAS_LIBRARY_NAMES="openblas" \
        -DLAPACK_LIBRARY_NAMES="openblas" \
        -DSCALAPACK_LIBRARY_NAMES="scalapack" \
        ..
$ make
$ make install
```

1.16.3 Compiling FrontISTR

Finishing compiling above libraries, compile FrontISTR.

```
$ cd $HOME/work
$ tar xvf FrontISTR_V50.tar.gz
$ cd FrontISTR
```

1.16.3.1 Edit Makefile.conf Copy template as Makefile.conf.org to Makefile.conf. Then edit Makefile.conf as follows.

```
$ cp Makefile.conf.org Makefile.conf
$ vi Makefile.conf
############################################################
#                                                          #
#     Setup Configulation File for FrontISTR               #
#                                                          #
############################################################

# MPI
MPIDIR          = /usr/lib/x86_64-linux-gnu/openmpi
MPIBINDIR       = /usr/bin
MPILIBDIR       = $(MPIDIR)/lib
MPIINCDIR       = $(MPIDIR)/include
MPILIBS         = -lmpi -lmpi_cxx -lmpi_mpifh

# for install option only
PREFIX          = $(HOME)/FrontISTR
BINDIR          = $(PREFIX)/bin
LIBDIR          = $(PREFIX)/lib
INCLUDEDIR      = $(PREFIX)/include

# Metis
METISDIR        = $(HOME)/local
METISLIBDIR     = $(METISDIR)/lib
METISINCDIR     = $(METISDIR)/include
HECMW_METIS_VER= 5

# ParMetis
PARMETISDIR     = $(HOME)/local
PARMETISLIBDIR = $(PARMETISDIR)/lib
PARMETISINCDIR = $(PARMETISDIR)/include
```

```
# Refiner
REFINERDIR      = $(HOME)/local
REFINERINCDIR   = $(REFINERDIR)/include
REFINERLIBDIR   = $(REFINERDIR)/lib

# Coupler
REVOCAPDIR      = $(HOME)/local
REVOCAPINCDIR   = $(REVOCAPDIR)/include
REVOCAPLIBDIR   = $(REVOCAPDIR)/lib

# MUMPS
MUMPSDIR        = $(HOME)/local
MUMPSINCDIR     = $(MUMPSDIR)/include
MUMPSLIBDIR     = $(MUMPSDIR)/lib
MUMPSLIBS       = -ldmumps -lmumps_common -lpord -L$(HOME)/local/lib -lscalapack

# MKL PARDISO
MKLDIR      = $(HOME)/
MKLINCDIR   = $(MKLDIR)/include
MKLLIBDIR   = $(MKLDIR)/lib

# ML
MLDIR           = $(HOME)/local
MLINCDIR        = $(MLDIR)/include
MLLIBDIR        = $(MLDIR)/lib
MLLIBS          = -lml -lamesos -ltrilinosss -lzoltan -lepetra -lteuchosremainder -lteuchosnu

# C compiler settings
CC              = mpicc -fopenmp
CFLAGS          =
LDFLAGS         = -lstdc++ -lm
OPTFLAGS        = -O3

# C++ compiler settings
CPP             = mpic++ -fopenmp
CPPFLAGS        =
CPPLDFLAGS      =
CPPOPTFLAGS     = -O3

# Fortran compiler settings
F90             = mpif90 -fopenmp
F90FLAGS        =
F90LDFLAGS      = -lstdc++ -L$(HOME)/local/lib -lopenblas
F90OPTFLAGS     = -O2
F90FPP          = -cpp
F90LINKER       = mpif90 -fopenmp

MAKE            = make
AR              = ar ruv
MV              = mv -f
CP              = cp -f
RM              = rm -f
MKDIR           = mkdir -p
```

1.16.3.2 Executing setup.sh Finished to edit Makefile.conf, then execute setup.sh.

```
$ ./setup.sh -p --with-tools --with-refiner \
             --with-metis --with-mumps --with-lapack --with-ml
```

1.16.3.3 Executing make

Execute make command.

```
$ make
```

```
$ make install
```

FrontISTR will be installed to $(HOME)/FrontISTR/bin.

1.16.3.5 Testing FrontISTR

Run sample case in the tutorial directory and check running of FrontISTR.

```
$ cd $HOME/work/FrontISTR/tutorial
$ cd 01_elastic_hinge
$ $HOME/FrontISTR/bin/fistr1
 Step control not defined! Using default step=1
 fstr_setup: OK
 Start visualize PSF 1 at timestep 0

 loading step=     1
 sub_step= 1,    current_time=  0.0000E+00, time_inc=  0.1000E+01
 loading_factor=     0.0000000    1.0000000
### 3x3 BLOCK CG, SSOR, 1
       1     1.903375E+00
       2     1.974378E+00
       3     2.534627E+00
       4     3.004045E+00
       5     3.202633E+00
       6     3.203864E+00
...
...
```

When finished analysis, displayed message as follows.

```
...
...
   2966     1.143085E-08
   2967     1.078272E-08
   2968     1.004759E-08
   2969     9.372882E-09
### Relative residual = 9.39169E-09

### summary of linear solver
      2969 iterations       9.391687E-09
    set-up time       :     4.108060E-01
    solver time       :     6.506822E+01
    solver/comm time  :     4.342469E-01
    solver/matvec     :     1.923199E+01
    solver/precond    :     2.688405E+01
    solver/1 iter     :     2.191587E-02
    work ratio (%)    :     9.933263E+01

 Start visualize PSF 1 at timestep 1
### FSTR_SOLVE_NLGEOM FINISHED!

 ================================
    TOTAL TIME (sec) :      74.93
          pre (sec) :       1.86
        solve (sec) :      73.07
```

```
==========================================
FrontISTR Completed !!
```

1.17 Appendix : Example of installation procedure to Windows10(Makefile.conf)

We will explain how to install this software and how to build external libraries required this software on Windows10.

More information for building each libraries, refer to their installation manuals.

1.17.1 Preparation

At first, install the basic build toolchains and libraries as follows.

1.17.1.1 Installation of compilers and toolchains for Windows First, install development environment. Development environment uses MSYS2 in this example.

https://www.msys2.org

Download 64bit installer named msys2−x86_64−xxxxxxxx.exe (xxxxxxxx is version number) and install it.

1.17.1.2 Installing binary package Finished to install above software, run windows command prompt named MSYS2 MinGW 64−bit, then install other required software.

```
(MINGW64) pacman -S base-devel mingw-w64-x86_64-toolchain \
            mingw-w64-x86_64-cmake \
            mingw-w64-x86_64-binutils \
            mingw-w64-x86_64-perl \
            git
```

Please check compilers works propery as follows.

```
(MINGW64) which gcc g++ gfortran
/mingw64/bin/gcc
/mingw64/bin/g++
/mingw64/bin/gfortran
```

1.17.2 Installing libraries

Compile and install required libraries this software. Working directory is $HOME/work, destination directory for install is $HOME/local.

And add $HOME/local/bin to PATH environment variable as follows.

```
(MINGW64) cd $HOME
(MINGW64) mkdir work
(MINGW64) mkdir -p local/bin local/lib local/include
(MINGW64) export PATH=$HOME/local/bin:$PATH
```

1.17.2.1 Installing MPI In this example, MPI libraries and runtime uses Microsoft MPI.

You can download runtime (msmpisetup.exe) and SDK (msmpisdk.msi) from the following URL.

Download Microsoft MPI v10.0

1.17.2.1.1 Generating .a format library file To link Microsoft MPI with gcc/gfortran provided from MinGW-w64, convert library format from DLL to .a.

```
(MINGW64) cd $HOME/local/lib
(MINGW64) gendef /c/Windows/System32/msmpi.dll
(MINGW64) dlltool -d msmpi.def -l libmsmpi.a -D /c/Windows/System32/msmpi.dll
(MINGW64) ls
libmsmpi.a msmpi.def
```

1.17.2.1.2 Modifying header files provided from MS-MPI Copy original header files from installation directory to current directory.

```
(MINGW64) cd $HOME/local/include
(MINGW64) cp /c/Program\ Files\ \(x86\)/Microsoft\ SDKs/MPI/Include/*.h .
(MINGW64) cp /c/Program\ Files\ \(x86\)/Microsoft\ SDKs/MPI/Include/x64/*.h .
(MINGW64) ls
mpi.h  mpif.h  mpifptr.h  mpio.h  mspms.h  pmidbg.h
```

1.17.2.2 Downloads Downloads the following software and save it to working directory $HOME/work.

Software	Link
REVOCAP_Refiner-1.1.04.tar.gz	https://www.frontistr.com/
FrontISTR_V50.tar.gz	https://www.frontistr.com/
OpenBLAS-0.2.20.tar.gz	http://www.openblas.net/
metis-5.1.0.tar.gz	http://glaros.dtc.umn.edu/gkhome/metis/metis/download
scalapack-2.0.2.tgz	http://www.netlib.org/scalapack/
MUMPS_5.1.2.tar.gz	http://mumps.enseeiht.fr/
trilinos-12.14.1-Source.tar.bz2	https://trilinos.org/download/

```
(MINGW64) cd $HOME/work
(MINGW64) tar xvf REVOCAP_Refiner-1.1.04.tar.gz
(MINGW64) cd REVOCAP_Refiner-1.1.04
(MINGW64) make
(MINGW64) cp lib/x86_64-linux/libRcapRefiner.a $HOME/local/lib
(MINGW64) cp Refiner/rcapRefiner.h $HOME/local/include
```

```
(MINGW64) pacman -S mingw-w64-x86_64-openblas
```

```
(MINGW64) cd $HOME/work
(MINGW64) tar xvf metis-5.1.0.tar.gz
(MINGW64) cd metis-5.1.0
```

Correct the following file to fit MinGW environment.

- Makefile
- GKlib/gk_arch.h
- GKlib/getopt.c

```
% vim Makefile
line:60 change from
```

```
cd $(BUILDDIR) && cmake $(CURDIR) $(CONFIG_FLAGS)
to
cd $(BUILDDIR) && cmake -G "MSYS Makefiles" $(CURDIR) $(CONFIG_FLAGS)
```

```
(MINGW64) vim GKlib/gk_arch.h
line:44 remove
  #include <sys/resource.h>
```

```
(MINGW64) vim GKlib/gk_getopt.h
line:54 remove following lines
/* Function prototypes */
extern int gk_getopt(int __argc, char **__argv, char *__shortopts);
extern int gk_getopt_long(int __argc, char **__argv, char *__shortopts,
              struct gk_option *__longopts, int *__longind);
extern int gk_getopt_long_only (int __argc, char **__argv,
              char *__shortopts, struct gk_option *__longopts, int *__longind);
```

```
(MINGW64) make config prefix=$HOME/local/ cc=gcc openmp=1
(MINGW64) make
(MINGW64) make install
```

```
(MINGW64) cd $HOME/work
(MINGW64) tar xvf scalapack-2.0.2.tgz
(MINGW64) cd scalapack-2.0.2
```

Copy template SLmake.inc.example to SLmake.inc. Then edit SLmake.inc as follows.

```
(MINGW64) cp SLmake.inc.example SLmake.inc
(MINGW64) vi SLmake.inc
```

```
#
#  The fortran and C compilers, loaders, and their flags
#

FC            = gfortran -fno-range-check
CC            = gcc
NOOPT         = -O0
FCFLAGS       = -O3 -I$(HOME)/local/include
CCFLAGS       = -O3 -I$(HOME)/local/include
FCLOADER      = $(FC)
CCLOADER      = $(CC)
FCLOADFLAGS   = $(FCFLAGS) -L$(HOME)/local/lib -lmsmpi
CCLOADFLAGS   = $(CCFLAGS) -L$(HOME)/local/lib -lmsmpi

#
#  BLAS, LAPACK (and possibly other) libraries needed for linking test programs
#

BLASLIB       = -lopenblas
LAPACKLIB     = -lopenblas
LIBS          = $(LAPACKLIB) $(BLASLIB)
```

```
(MINGW64) make
(MINGW64) cp libscalapack.a $HOME/local/lib
```

Finished to make, then copy libray.

Although an error is displayed at the end of compilation, ignore it.

```
(MINGW64) cd $HOME/work
(MINGW64) tar xvf MUMPS_5.1.2.tar.gz
(MINGW64) cd MUMPS_5.1.2
(MINGW64) cp Make.inc/Makefile.inc.generic Makefile.inc
```

Change the following parts of copied Makefile.inc.

```
(MINGW64) vi Makefile.inc
(MINGW64) cp Make.inc/Makefile.inc.generic Makefile.inc
(MINGW64) vi Makefile.inc
LMETISDIR = $(HOME)/local
IMETIS    = -I$(LMETISDIR)/include
LMETIS    = -L$(LMETISDIR)/lib -lmetis

ORDERINGSF  = -Dmetis -Dpord

CC      = gcc
FC      = gfortran -fno-range-check
FL      = gfortran

LAPACK = -lopenblas

SCALAP  = -L$(HOME)/local/lib -lscalapack

INCPAR = -I$(HOME)/local/include

LIBPAR  = $(SCALAP) $(LAPACK) -L$(HOME)/local/lib -lmsmpi

LIBBLAS = -lopenblas

LIBOTHERS = -lpthread -fopenmp
```

Then execute make.

```
(MINGW64) make
(MINGW64) cp lib/*.a $HOME/local/lib
(MINGW64) cp include/*.h $HOME/local/include
```

```
(MINGW64) cd $HOME/work
(MINGW64) tar xvf trilinos-12.14.1-Source.tar.gz
(MINGW64) cd trilinos-12.14.1-Source
(MINGW64) mkdir build
(MINGW64) cmake -G "MSYS Makefiles" \
        -DCMAKE_INSTALL_PREFIX="$HOME/local" \
        -DCMAKE_CXX_FLAGS="-I$HOME/local/include" \
        -DCMAKE_C_FLAGS="-I$HOME/local/include" \
        -DBLAS_LIBRARY_NAMES="openblas" \
        -DLAPACK_LIBRARY_NAMES="openblas" \
        -DMPI_USE_COMPILER_WRAPPERS=OFF \
        -DMPI_C_HEADER_DIR="$HOME/local/include" \
        -DMPI_CXX_HEADER_DIR="$HOME/local/include" \
        -DTPL_ENABLE_MPI=ON \
        -DTrilinos_ENABLE_OpenMP=ON \
        -DTrilinos_ENABLE_ML=ON \
        -DTrilinos_ENABLE_Zoltan=ON \
        -DTrilinos_ENABLE_ALL_OPTIONAL_PACKAGES=OFF \
        ..
```

```
(MINGW64) make
(MINGW64) make install
```

1.17.3 Compiling FrontISTR

Finishing compiling above libraries, compile FrontISTR.

```
(MINGW64) cd $HOME/work
(MINGW64) tar xvf FrontISTR_V50.tar.gz
(MINGW64) cd FrontISTR
```

1.17.3.1 Editing Makefile.conf Copy template as Makefile.conf.org to Makefile.conf. Then edit Makefile.conf as follows.

```
(MINGW64) cp Makefile.conf.org Makefile.conf
(MINGW64) vi Makefile.conf
#############################################################
#                                                           #
#      Setup Configulation File for FrontISTR               #
#                                                           #
#############################################################

# MPI
MPIDIR          = $(HOME)/local
MPIBINDIR       = "/c/Program\ Files/Microsoft\ MPI/Bin/"
MPILIBDIR       = $(MPIDIR)/lib
MPIINCDIR       = $(MPIDIR)/include
MPILIBS         = -lmsmpi

# for install option only
PREFIX          = $(HOME)/FrontISTR
BINDIR          = $(PREFIX)/bin
LIBDIR          = $(PREFIX)/lib
INCLUDEDIR      = $(PREFIX)/include

# Metis
METISDIR        = $(HOME)/local
METISLIBDIR     = $(METISDIR)/lib
METISINCDIR     = $(METISDIR)/include
HECMW_METIS_VER= 5

# ParMetis
PARMETISDIR     = $(HOME)/local
PARMETISLIBDIR  = $(PARMETISDIR)/lib
PARMETISINCDIR  = $(PARMETISDIR)/include

# Refiner
REFINERDIR      = $(HOME)/local
REFINERINCDIR   = $(REFINERDIR)/include
REFINERLIBDIR   = $(REFINERDIR)/lib

# Coupler
REVOCAPDIR      = $(HOME)/local
REVOCAPINCDIR   = $(REVOCAPDIR)/include
```

```
REVOCAPLIBDIR  = $(REVOCAPDIR)/lib

# MUMPS
MUMPSDIR       = $(HOME)/local
MUMPSINCDIR    = $(MUMPSDIR)/include
MUMPSLIBDIR    = $(MUMPSDIR)/lib
MUMPSLIBS      = -ldmumps -lmumps_common -lpord -L$HOME/local/lib -lscalapack

# MKL PARDISO
MKLDIR     = $(HOME)/
MKLINCDIR  = $(MKLDIR)/include
MKLLIBDIR  = $(MKLDIR)/lib

# ML
MLDIR          = $(HOME)/local
MLINCDIR       = $(MLDIR)/include
MLLIBDIR       = $(MLDIR)/lib
MLLIBS         = -lml -lzoltan -lws2_32

# C compiler settings
CC             = gcc -fopenmp
CFLAGS         = -D_WINDOWS
LDFLAGS        = -lstdc++ -lm
OPTFLAGS       = -O3

# C++ compiler settings
CPP            = g++ -fopenmp
CPPFLAGS       = -D_WINDOWS
CPPLDFLAGS     =
CPPOPTFLAGS    = -O3

# Fortran compiler settings
F90            = gfortran -fopenmp -fno-range-check
F90FLAGS       =
F90LDFLAGS     = -lstdc++ -lopenblas
F90OPTFLAGS    = -O2
F90FPP         = -cpp
F90LINKER      = gfortran -fopenmp

MAKE           = make
AR             = ar ruv
MV             = mv -f
CP             = cp -f
RM             = rm -f
MKDIR          = mkdir -p
```

1.17.3.2 Executing setup.sh

Finished to edit Makefile.conf, then execute setup.sh.

```
(MINGW64) ./setup.sh -p --with-tools --with-refiner \
          --with-metis --with-mumps --with-lapack --with-ml
```

1.17.3.3 Executing make

Execute make command.

```
(MINGW64) make
```

```
(MINGW64) make install
```

FrontISTR will be installed to $(HOME)/FrontISTR/bin.

1.17.3.5 Testing FrontISTR Run sample case in the tutorial directory and check running of FrontISTR.

```
(MINGW64) cd $HOME/work/FrontISTR/tutorial
(MINGW64) cd 01_elastic_hinge
(MINGW64$) $HOME/FrontISTR/bin/fistr1
 Step control not defined! Using default step=1
 fstr_setup: OK
 Start visualize PSF 1 at timestep 0

 loading step=     1
 sub_step= 1,    current_time=  0.0000E+00, time_inc=  0.1000E+01
 loading_factor=     0.0000000    1.0000000
### 3x3 BLOCK CG, SSOR, 1
       1     1.903375E+00
       2     1.974378E+00
       3     2.534627E+00
       4     3.004045E+00
       5     3.202633E+00
       6     3.203864E+00
...
...
```

When finished analysis, displayed message as follows.

```
...
...
   2966     1.143085E-08
   2967     1.078272E-08
   2968     1.004759E-08
   2969     9.372882E-09
### Relative residual = 9.39169E-09

### summary of linear solver
      2969 iterations       9.391687E-09
     set-up time       :     4.108060E-01
     solver time       :     6.506822E+01
     solver/comm time  :     4.342469E-01
     solver/matvec     :     1.923199E+01
     solver/precond    :     2.688405E+01
     solver/1 iter     :     2.191587E-02
     work ratio (%)    :     9.933263E+01

 Start visualize PSF 1 at timestep 1
### FSTR_SOLVE_NLGEOM FINISHED!

 ================================
    TOTAL TIME (sec) :      74.93
          pre (sec) :       1.86
        solve (sec) :      73.07
 ================================
 FrontISTR Completed !!
```

1.17.3.6 Supplement To run in an environment where MinGW is not installed, you need to place following files in the same directory as FrontISTR fistr1 .exe.

- libwinpthread-1.dll
- libgfortran-3.dll
- libgcc_s_seh-1.dll
- libgomp-1.dll
- libstdc++-6.dll
- libquadmath-0.dll

You can find these libraries from

C:\\mingw64\bin

You also need to setup Microsoft MPI runtime (MSMpiSetup.exe).