Research and Development for Next-generation Information Technology of
Ministry of Education, Culture, Sports, Science and Technology
"Research and Development of Innovative Simulation Software"

CISS Free Software

FrontISTR

Ver. 4.3

Installation Manual

Corresponding Clerks:
(Engagement) The Foundation for the Promotion of Industrial Science (F.P.I.S)
4-6-1 Komaba, Meguro-ku, Tokyo 153-8505 JAPAN
(Management) Center for Research on Innovative Simulation Software,
Institute of Industrial Science (IIS), the University of Tokyo
4-6-1 Komaba, Meguro-ku, Tokyo 153-8505 JAPAN
Fax : +81-3-5452-6662
E-mail : software@ciss.iis.u-tokyo.ac.jp

Contents

1. Introduction

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

2. Operating Environment

2.1 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) C, C++, and Fortran90 compilers

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

(2) Boost Libraries

Boost Libraries are required to compile the C++ source code of this software. If Boost Libraries have not been installed in the environment where this software is to be installed, they can be downloaded from the following website:

http://www.boost.org/

(3) Intel MKL (Math 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.

(4) MPI

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

http://www.mcs.anl.gov/research/projects/mpich2

(5) 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. However, if below mentioned MUMPS is to be used, and METIS is to be used for ordering in MUMPS, Metis Ver.4 series has to be installed since MUMPS supports only Ver.4 series of METIS. For environments where METIS is not

installed, domain decomposition is still possible using the RCB algorithm. METIS can be downloaded from the following website:

http://glaros.dtc.umn.edu/gkhome/views/metis/index.html

(6) ParMETIS

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

ParMETIS is not required at present.

(7) 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.

(8) REVOCAP_Refiner

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

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

(9) 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

(10) 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://graal.ens-lyon.fr/MUMPS/

2.2 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.

Table 1: Operation verification environment

Environment (operating system)	Compiler	Parallel processing environment
K computer	Fujitsu Compiler	Fujitsu MPI
EARTH SIMULATOR (ES2)	NEC Compiler	NEC MPI
Intel Xeon Cluster CentOS 5	Intel Compiler	Intel MPI
AMD Opteron Cluster RedHat Enterprise Linux 5	Intel Compiler	OpenMPI
Intel Itanium Cluster SUSE Linux Enterprise 10	Intel Compiler	Intel MPI
AMD Opteron Cluster CentOS 4.4	Intel Compiler	MPICH 1.2.7p1
PC Windows XP, Windows 7	gnu Compiler	MPICH2-1.3.2p1
PC Windows XP x64	Intel Compiler	Intel MPI

3. 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_V42.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-V42.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 as "${FSTRBUILDDIR}".)

4. Installation

Use the following procedure to install this software.

4.1 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.

MPIDIR: Specifies the directory where MPI has been installed
PREFIX: Specifies the directory where the executable module of this software is to be installed
METISDIR: Specifies the directory where METIS has been installed
PARMETISDIR: Specifies the directory where ParMETIS has been installed
REFINERDIR: Specifies the directory where REVOCAP_Refiner has been installed
REVOCAPDIR: 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.

4.2 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.

Table 2: 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-paracon	Generates a executable module for parallel contact analysis	

Examples of setup.sh execution are shown below.

(1) Compiling for parallel processing

If this software is used in a parallel execution environment where MPI has been installed, execute setup.sh with the -p or -parallel option specified, as below.

```
$ ./setup.sh –p
```

(2) Generating a tool such as a partitioner

If a preprocessing 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
```

(3) 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
```

(4) 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, --with-mumps and --with-paracon options specified, as below

```
$ ./setup.sh  -p --with-metis --with_mumps --with_paracon
```

4.3 Executing make

Execute make in ${FSTRBUILDDIR} as below.

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

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

4.4 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
```

4.5 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

Parallel processing version: Cygwin

Appendix A: List of Makefile.conf Variables

(1) Settings related to MPI

MPIDIR

Description: 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.

Default value: None

MPIBINDIR

Description: This variable specifies the path to the directory where MPI executable files have been installed.

Default value: None

MPILIBDIR

Description: This variable specifies the path to the directory where MPI libraries have been installed.

Default value: .

MPIINCDIR

Description: This variable specifies the path to the directory where MPI header files have been installed.

Default value: .

MPILIBS

Description: This variable specifies the MPI library that will be linked to C and Fortran90 object files.

Default value: None

(2) Settings related to the installation directory

PREFIX

Description: This variable specifies the path to the directory where this software is to be installed.

Default value: $(HOME)/FrontISTR

BINDIR

Description: This variable specifies the path to the directory where the executable files of this software are to be installed. Normally, there is no need to change the value of this variable.

Default value: $(PREFIX)/bin

LIBDIR

Description: 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.

Default value: $(PREFIX)/lib

INCLUDEDIR

Description: 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.

Default value: $(PREFIX)/include

(3) Settings related to METIS

METISDIR

Description: This variable specifies the path to the directory where METIS has been installed.

Default value: $(HOME)/metis-4.0

METISLIBDIR

Description: This variable specifies the path to the directory where the library (libmetis.a) of METIS has been installed. Normally, there is no need to change the value of this variable from the default value.

Default value: $(METISDIR)

METISINCDIR

Description: 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.

Default value: $(METISDIR)/Lib

(4) Settings related to ParMETIS

PARMETISDIR

Description: This variable specifies the path to the directory where ParMETIS has been installed.

Default value: $(HOME)/ParMetis-3.1

PARMETISLIBDIR

Description: 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.
Default value: $(PARMETISDIR)

PAEMETISINCDIR
Description: 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.
Default value: $(PARMETISDIR)/ParMETISLib

(5) Settings related to REVOCAP_Refiner

REFINERDIR
Description: This variable specifies the path to the directory where REVOCAP_Refiner has been installed.
Default value: $(HOME)/ REVOCAP_Refiner

REFINERINCDIR
Description: 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.
Default value: $(REFINERDIR)/Refiner

REFINERLIBDIR
Description: 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.
Default value: $(REFINERDIR)/lib

(6) Settings related to REVOCAP_Coupler

REVOCAPDIR
Description: This variable specifies the path to the directory where REVOCAP_Coupler has been installed.
Default value: $(HOME)/ REVOCAP_Coupler

REVOCAPINCDIR
Description: 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.
Default value: $(REVOCAPDIR)/librcap

REVOCAPLIBDIR

Description: 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.
Default value: $(REVOCAPDIR)/librcap

(7) Settings related to MUMPS

MUMPSDIR
Description: This variable specifies the path to the directory where MUMPS has been installed.
Default value: $(HOME)/MUMPS

MUMPSINCDIR
Description: 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.
Default value: $(MUMPSDIR)/include

MUMPSLIBDIR
Description: 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.
Default value: $(MUMPSDIR)/lib

(8) Settings related to the C compiler

CC
Description: This variable specifies the C compiler start command.
Default value: mpicc

CFLAGS
Description: 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.
Default value: None

LDFLAGS
Description: 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.
Default value: -lm

OPTFLAGS

Description: This variable specifies the optimization option (or another option) to be assigned to the C compiler.

Default value: -O3

CLINKER

Description: 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.

Default value: [The value specified for CC]

(9) Settings related to the C++ compiler

CPP

Description: This variable specifies the C compiler start command.

Default value: mpic++

CPPFLAGS

Description: 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.

Default value: -DMPICH_IGNORE_CXX_SEEK (Note: This is required for Intel compilers.)

CPPLDFLAGS

Description: 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.

Default value: None

CPPOPTFLAGS

Description: This variable specifies the optimization option (or another option) to be assigned to the C compiler.

Default value: -O3

(10) Settings related to the Fortran90 compiler

F90

Description: This variable specifies the Fortran90 compiler start command.

Default value: mpif90

F90FLAGS

Description: 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.

Default value: None

F90LDFLAGS

Description: 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.

Default value: None

F90OPTFLAGS

Description: This variable specifies the optimization option (or another option) to be assigned to the Fortran90 compiler.

Default value: -O2

F90LINKER

Description: 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.)

Default value: [The value specified for F90]

(11) Settings related to UNIX commands

MAKE

Description: 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.

Default value: make

AR

Description: 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.

Default value: ar ruv

CP

Description: 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.
Default value: cp -f

RM
Description: 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.
Default value: rm -f

MKDIR
Setting : 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.
Default value: mkdir –p

Appendix B: Makefile.conf Setting Example

```
# 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

# 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

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

Appendix C: 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.