XLiFE++: an eXtended Library of Finite Elements in C++

Install and Use

Nicolas Kielbasiewicz, Nicolas Salles

Unité de Mathématiques Appliquées,
ENSTA - Paristech

June 14, 2016

Overview

1. Downloading XLIFE++
2. Installing XLIFE++
3. Using XLIFE++

1 Downloading XLIFE++

2 Installing XLIFE++
- Requirements
- About CMAKE
- Building XLIFE++
- Compiling XLIFE++

3 Using XLIFE++
- Requirements
- Running user script
- About PARAVIEW

XLiFE++ is downloadable at the following url: http://uma.ensta-paristech.fr/soft/XLiFE++

About website:

- Archives of XLiFE++ are of 3 kinds: sources only, api only, and sources/api (full)
- User documentation is available in pdf format or in HTML format
- Source documentation (API), generated by DOXYGEN, is available here
- Qt-based applications for users will be downloadable (under development)

1. Downloading XLIFE++
2. Installing XLIFE++
   - Requirements
   - About CMAKE
   - Building XLIFE++
   - Compiling XLIFE++
3. Using XLIFE++
   - Requirements
   - Running user script
   - About PARAVIEW

Requirements

- Basically, XLIFE++ compilation depends on the cross-platform builder CMAKE, available at `http://cmake.org`.
- To use the full abilities of mesh definition in XLIFE++, you may also install GMSH (`http://gmsh.info`).
- To use the full abilities of XLIFE++, you may also install ARPACK and/or UMFPACK.
- To plot solutions, you may install PARAVIEW (`http://www.paraview.org`) or GMSH.
- A C++ compiler

- Basically, XLiFE++ compilation depends on the cross-platform builder CMAKE, available at `http://cmake.org`.
- To use the full abilities of mesh definition in XLiFE++, you may also install GMSH (`http://gmsh.info`).
- To use the full abilities of XLiFE++, you may also install ARPACK and/or UMFPACK.
- To plot solutions, you may install PARAVIEW (`http://www.paraview.org`) or GMSH.
- A C++ compiler

Requirements

- Basically, XLIFE++ compilation depends on the cross-platform builder CMAKE, available at `http://cmake.org`.
- To use the full abilities of mesh definition in XLIFE++, you may also install GMSH (`http://gmsh.info`).
- To use the full abilities of XLIFE++, you may also install ARPACK and/or UMFPACK.
- To plot solutions, you may install PARAVIEW (`http://www.paraview.org`) or GMSH.
- A C++ compiler

- Basically, XLiFE++ compilation depends on the cross-platform builder CMAKE, available at `http://cmake.org`.
- To use the full abilities of mesh definition in XLiFE++, you may also install GMSH (`http://gmsh.info`).
- To use the full abilities of XLiFE++, you may also install ARPACK and/or UMFPACK.
- To plot solutions, you may install PARAVIEW (`http://www.paraview.org`) or GMSH.
- A C++ compiler

- Basically, XLiFE++ compilation depends on the cross-platform builder CMAKE, available at `http://cmake.org`.
- To use the full abilities of mesh definition in XLiFE++, you may also install GMSH (`http://gmsh.info`).
- To use the full abilities of XLiFE++, you may also install ARPACK and/or UMFPACK.
- To plot solutions, you may install PARAVIEW (`http://www.paraview.org`) or GMSH.
- A C++ compiler

> ! If you want to use multithreading capabilities, you may install a compatible compiler

- clang/clang++ does not support natively OpenMP
- on Mac, g++ is an alias to clang++
- On Mac, with Homebrew package manager, g++ compilers are not installed by default with OpenMP management
- On Windows If CodeBlocks is installed with its own MinGW compiler, it does not support OpenMP

- Basically, XLIFE++ compilation depends on the cross-platform builder CMAKE, available at `http://cmake.org`.
- To use the full abilities of mesh definition in XLIFE++, you may also install GMSH (`http://gmsh.info`).
- To use the full abilities of XLIFE++, you may also install ARPACK and/or UMFPACK.
- To plot solutions, you may install PARAVIEW (`http://www.paraview.org`) or GMSH.
- A C++ compiler

! If you want to use multithreading capabilities, you may install a compatible compiler

- clang/clang++ does not support natively OpenMP
- on Mac, g++ is an alias to clang++
- On Mac, with Homebrew package manager, g++ compilers are not installed by default with OpenMP management
- On Windows If CodeBlocks is installed with its own MinGW compiler, it does not support OpenMP

- Basically, XLiFE++ compilation depends on the cross-platform builder CMAKE, available at `http://cmake.org`.
- To use the full abilities of mesh definition in XLiFE++, you may also install GMSH (`http://gmsh.info`).
- To use the full abilities of XLiFE++, you may also install ARPACK and/or UMFPACK.
- To plot solutions, you may install PARAVIEW (`http://www.paraview.org`) or GMSH.
- A C++ compiler

! If you want to use multithreading capabilities, you may install a compatible compiler

- clang/clang++ does not support natively OpenMP
- on Mac, g++ is an alias to clang++
- On Mac, with Homebrew package manager, g++ compilers are not installed by default with OpenMP management
- On Windows If CodeBlocks is installed with its own MinGW compiler, it does not support OpenMP

- Basically, XLIFE++ compilation depends on the cross-platform builder CMAKE, available at `http://cmake.org`.
- To use the full abilities of mesh definition in XLIFE++, you may also install GMSH (`http://gmsh.info`).
- To use the full abilities of XLIFE++, you may also install ARPACK and/or UMFPACK.
- To plot solutions, you may install PARAVIEW (`http://www.paraview.org`) or GMSH.
- A C++ compiler

! If you want to use multithreading capabilities, you may install a compatible compiler

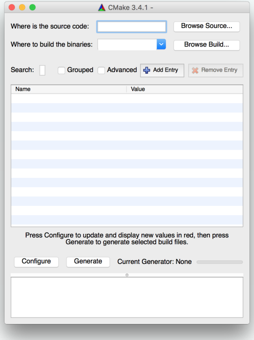
- clang/clang++ does not support natively OpenMP
- on Mac, g++ is an alias to clang++
- On Mac, with Homebrew package manager, g++ compilers are not installed by default with OpenMP management
- On Windows If CodeBlocks is installed with its own MinGW compiler, it does not support OpenMP

- Basically, XLiFE++ compilation depends on the cross-platform builder CMAKE, available at `http://cmake.org`.
- To use the full abilities of mesh definition in XLiFE++, you may also install GMSH (`http://gmsh.info`).
- To use the full abilities of XLiFE++, you may also install ARPACK and/or UMFPACK.
- To plot solutions, you may install PARAVIEW (`http://www.paraview.org`) or GMSH.
- A C++ compiler

! If you want to use multithreading capabilities, you may install a compatible compiler

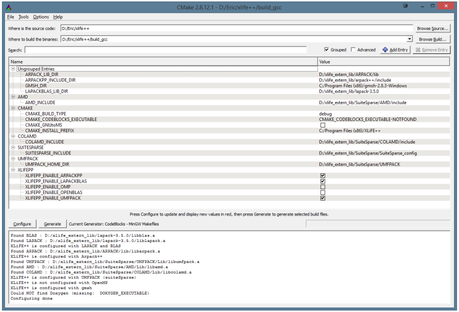
- clang/clang++ does not support natively OpenMP
- on Mac, g++ is an alias to clang++
- On Mac, with Homebrew package manager, g++ compilers are not installed by default with OpenMP management
- On Windows If CodeBlocks is installed with its own MinGW compiler, it does not support OpenMP

CMAKE can be called and used command-line or with a GUI:



Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: None



Where is the source code: D:/Eric/xlife++

Where to build the binaries: D:/Eric/xlife++/build_gcc

Name	Value
Ungrouped Entries	
ARPACK_LIB_DIR	D:/xlife_extern_lib/ARPACK/lib
ARPACKPP_INCLUDE_DIR	D:/xlife_extern_lib/arpack++/include
GMSH_DIR	C:/Program Files (x86)/gmsh-2.8.3-Windows
LAPACKBLAS_LIB_DIR	D:/xlife_extern_lib/lapack-3.5.0
AMD	
AMD_INCLUDE	D:/xlife_extern_lib/SuiteSparse/AMD/include
CMAKE	
CMAKE_BUILD_TYPE	debug
CMAKE_CODEBLOCKS_EXECUTABLE	CMAKE_CODEBLOCKS_EXECUTABLE-NOTFOUND
CMAKE_GNUtoMS	□
CMAKE_INSTALL_PREFIX	C:/Program Files (x86)/XLiFE++
COLAMD	
COLAMD_INCLUDE	D:/xlife_extern_lib/SuiteSparse/COLAMD/include
SUITESPARSE	
SUITESPARSE_INCLUDE	D:/xlife_extern_lib/SuiteSparse/SuiteSparse_config
UMFPACK	
UMFPACK_HOME_DIR	D:/xlife_extern_lib/SuiteSparse/UMFPACK
XLIFEPP	
XLIFEPP_ENABLE_ARPACKPP	☑
XLIFEPP_ENABLE_LAPACKBLAS	☑
XLIFEPP_ENABLE_OMP	□
XLIFEPP_ENABLE_OPENBLAS	□
XLIFEPP_ENABLE_UMFPACK	☑

Press Configure to update and display new values in red, then press Generate to generate selected build files.

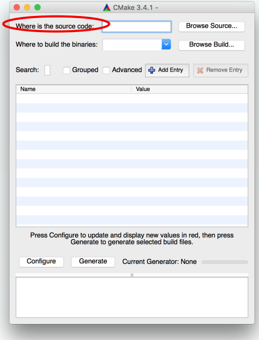
Current Generator: CodeBlocks - MinGW Makefiles

```
Found BLAS : D:/xlife_extern_lib/lapack-3.5.0/libblas.a
Found LAPACK : D:/xlife_extern_lib/lapack-3.5.0/liblapack.a
XLiFE++ is configured with LAPACK and BLAS
Found ARPACK : D:/xlife_extern_lib/ARPACK/lib/libarpack.a
XLiFE++ is configured with Arpack++
Found UMFPACK : D:/xlife_extern_lib/SuiteSparse/UMFPACK/Lib/libumfpack.a
Found AMD : D:/xlife_extern_lib/SuiteSparse/AMD/Lib/libamd.a
Found COLAMD : D:/xlife_extern_lib/SuiteSparse/COLAMD/Lib/libcolamd.a
XLiFE++ is configured with UMFPACK (suiteSparse)
XLiFE++ is not configured with OpenMP
XLiFE++ is configured with gmsh
Could NOT find Doxygen (missing:  DOXYGEN_EXECUTABLE)
Configuring done
```

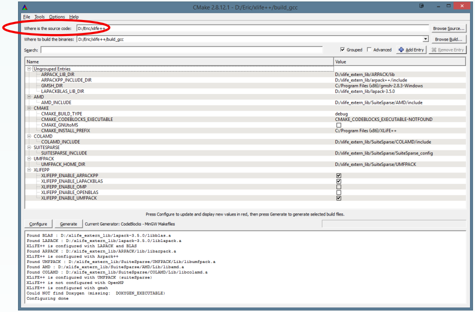
Installing XLiFE++

About CMAKE

CMAKE can be called and used command-line or with a GUI:



Press Configure to update and display new values in red, then press Generate to generate selected build files.



Where is the source code: D:/Eric/xlife++

Where to build the binaries: D:/Eric/xlife++/build_gcc

Name	Value
Ungrouped Entries	
ARPACK_LIB_DIR	D:/xlife_extern_lib/ARPACK/lib
ARPACKPP_INCLUDE_DIR	D:/xlife_extern_lib/arpack++/include
GMSH_DIR	C:/Program Files (x86)/gmsh-2.8.3-Windows
LAPACKBLAS_LIB_DIR	D:/xlife_extern_lib/lapack-3.5.0
AMD	
AMD_INCLUDE	D:/xlife_extern_lib/SuiteSparse/AMD/include
CMAKE	
CMAKE_BUILD_TYPE	debug
CMAKE_CODEBLOCKS_EXECUTABLE	CMAKE_CODEBLOCKS_EXECUTABLE-NOTFOUND
CMAKE_GNUtoMS	□
CMAKE_INSTALL_PREFIX	C:/Program Files (x86)/XLiFE++
COLAMD	
COLAMD_INCLUDE	D:/xlife_extern_lib/SuiteSparse/COLAMD/include
SUITESPARSE	
SUITESPARSE_INCLUDE	D:/xlife_extern_lib/SuiteSparse/SuiteSparse_config
UMFPACK	
UMFPACK_HOME_DIR	D:/xlife_extern_lib/SuiteSparse/UMFPACK
XLIFEPP	
XLIFEPP_ENABLE_ARPACKPP	☑
XLIFEPP_ENABLE_LAPACKBLAS	☑
XLIFEPP_ENABLE_OMP	□
XLIFEPP_ENABLE_OPENBLAS	□
XLIFEPP_ENABLE_UMFPACK	☑

Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: CodeBlocks - MinGW Makefiles

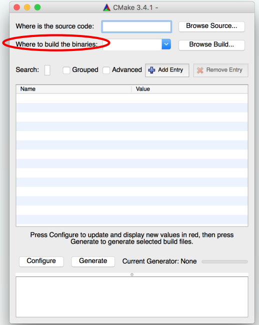
```
Found BLAS : D:/xlife_extern_lib/lapack-3.5.0/libblas.a
Found LAPACK : D:/xlife_extern_lib/lapack-3.5.0/liblapack.a
XLiFE++ is configured with LAPACK and BLAS
Found ARPACK : D:/xlife_extern_lib/ARPACK/lib/libarpack.a
XLiFE++ is configured with Arpack++
Found UMFPACK : D:/xlife_extern_lib/SuiteSparse/UMFPACK/Lib/libumfpack.a
Found AMD : D:/xlife_extern_lib/SuiteSparse/AMD/Lib/libamd.a
Found COLAMD : D:/xlife_extern_lib/SuiteSparse/COLAMD/Lib/libcolamd.a
XLiFE++ is configured with UMFPACK (suiteSparse)
XLiFE++ is not configured with OpenMP
XLiFE++ is configured with gmsh
Could NOT find Doxygen (missing:  DOXYGEN_EXECUTABLE)
Configuring done
```

Where is the source code: This should be $XLIFEPP_HOME_DIR, containing the top-level CMakeLists.txt script that CMAKE will run.

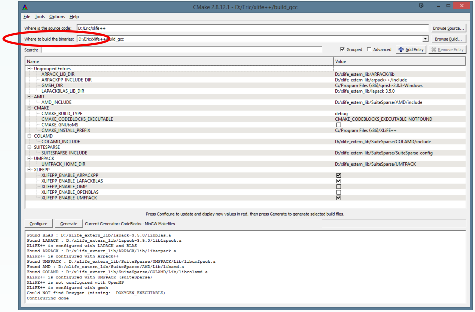
Installing XLiFE++

About CMAKE

CMAKE can be called and used command-line or with a GUI:



Press Configure to update and display new values in red, then press Generate to generate selected build files.



Where is the source code: D:/Eric/xlife++

Where to build the binaries: D:/Eric/xlife++/build_gcc

Name	Value
⊟ Ungrouped Entries	
ARPACK_LIB_DIR	D:/xlife_extern_lib/ARPACK/lib
ARPACKPP_INCLUDE_DIR	D:/xlife_extern_lib/arpack++/include
GMSH_DIR	C:/Program Files (x86)/gmsh-2.8.3-Windows
LAPACKBLAS_LIB_DIR	D:/xlife_extern_lib/lapack-3.5.0
⊟ AMD	
AMD_INCLUDE	D:/xlife_extern_lib/SuiteSparse/AMD/include
⊟ CMAKE	
CMAKE_BUILD_TYPE	debug
CMAKE_CODEBLOCKS_EXECUTABLE	CMAKE_CODEBLOCKS_EXECUTABLE-NOTFOUND
CMAKE_GNUtoMS	□
CMAKE_INSTALL_PREFIX	C:/Program Files (x86)/XLiFE++
⊟ COLAMD	
COLAMD_INCLUDE	D:/xlife_extern_lib/SuiteSparse/COLAMD/include
⊟ SUITESPARSE	
SUITESPARSE_INCLUDE	D:/xlife_extern_lib/SuiteSparse/SuiteSparse_config
⊟ UMFPACK	
UMFPACK_HOME_DIR	D:/xlife_extern_lib/SuiteSparse/UMFPACK
⊟ XLIFEPP	
XLIFEPP_ENABLE_ARPACKPP	☑
XLIFEPP_ENABLE_LAPACKBLAS	☑
XLIFEPP_ENABLE_OMP	□
XLIFEPP_ENABLE_OPENBLAS	□
XLIFEPP_ENABLE_UMFPACK	☑

Press Configure to update and display new values in red, then press Generate to generate selected build files.

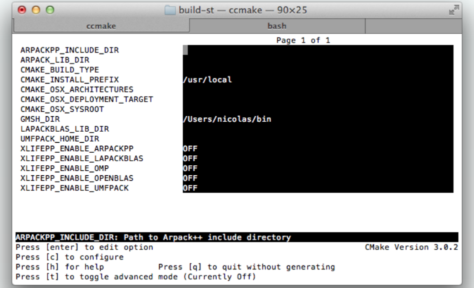
Current Generator: CodeBlocks - MinGW Makefiles

```
Found BLAS : D:/xlife_extern_lib/lapack-3.5.0/libblas.a
Found LAPACK : D:/xlife_extern_lib/lapack-3.5.0/liblapack.a
XLiFE++ is configured with LAPACK and BLAS
Found ARPACK : D:/xlife_extern_lib/ARPACK/lib/libarpack.a
XLiFE++ is configured with Arpack++
Found UMFPACK : D:/xlife_extern_lib/SuiteSparse/UMFPACK/Lib/libumfpack.a
Found AMD : D:/xlife_extern_lib/SuiteSparse/AMD/Lib/libamd.a
Found COLAMD : D:/xlife_extern_lib/SuiteSparse/COLAMD/Lib/libcolamd.a
XLiFE++ is configured with UMFPACK (suiteSparse)
XLiFE++ is not configured with OpenMP
XLiFE++ is configured with gmsh
Could NOT find Doxygen (missing:  DOXYGEN_EXECUTABLE)
Configuring done
```

Where is the source code: This should be $XLIFEPP_HOME_DIR, containing the top-level CMakeLists.txt script that CMAKE will run.

Where to build the binaries: This is the directory where CMAKE will put the build files, such as CMAKE cache, object files, ... It is often $XLIFEPP_HOME_DIR/build

CMAKE can be called with a GUI in Terminal on Unix/Linux and MacOS:



```
                                                     Page 1 of 1
 ARPACKPP_INCLUDE_DIR
 ARPACK_LIB_DIR
 CMAKE_BUILD_TYPE
 CMAKE_INSTALL_PREFIX             /usr/local
 CMAKE_OSX_ARCHITECTURES
 CMAKE_OSX_DEPLOYMENT_TARGET
 CMAKE_OSX_SYSROOT
 GMSH_DIR                         /Users/nicolas/bin
 LAPACKBLAS_LIB_DIR
 UMFPACK_HOME_DIR
 XLIFEPP_ENABLE_ARPACKPP          OFF
 XLIFEPP_ENABLE_LAPACKBLAS        OFF
 XLIFEPP_ENABLE_OMP               OFF
 XLIFEPP_ENABLE_OPENBLAS          OFF
 XLIFEPP_ENABLE_UMFPACK           OFF



ARPACKPP_INCLUDE_DIR: Path to Arpack++ include directory
Press [enter] to edit option                                                CMake Version 3.0.2
Press [c] to configure
Press [h] for help           Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently Off)
```

Where to build the binaries: The directory in which `ccmake` is called

Where is the source code: The directory containing the top-level CMakeLists.txt. It is given with a relative path to the `ccmake` command:

```
ccmake ..
```

In command line mode, options are set with -D:
Example:

```
cmake .. -DCMAKE_CXX_COMPILER=g++-4.8
```

Main CMAKE options:

CMAKE_CXX_COMPILER This option is used to set the compiler CMAKE will use to compile your code:

CMAKE_BUILD_TYPE This option is used to set the type of build (debug, release, . . .). Possible values are essentially Debug and Release. The default value is Release.

Configuring XLiFE++ with GMSH:

XLIFEPP_GMSH_DIR to specify the directory containing the GMSH binary. If GMSH is in your path, it will be automatically detected, else you can use this option.

Configuring XLIFE++ with GMSH:

XLIFEPP_GMSH_DIR to specify the directory containing the GMSH binary. If GMSH is in your path, it will be automatically detected, else you can use this option.

Configuring XLIFE++ with PARAVIEW:

XLIFEPP_PARAVIEW_DIR to specify the directory containing the PARAVIEW binary. If PARAVIEW is in your path, it will be automatically detected, else you can use this option:

Configuring XLIFE++ with GMSH:

XLIFEPP_GMSH_DIR to specify the directory containing the GMSH binary. If GMSH is in your path, it will be automatically detected, else you can use this option.

Configuring XLIFE++ with PARAVIEW:

XLIFEPP_PARAVIEW_DIR to specify the directory containing the PARAVIEW binary. If PARAVIEW is in your path, it will be automatically detected, else you can use this option:

Configuring XLIFE++ with OpenMP:

XLIFEPP_ENABLE_OMP Activates/Deactivates use of OpenMP

```
cmake .. –DXLIFEPP_ENABLE_OMP=ON
```

Configuring XLIFE++ with ARPACK:

XLIFEPP_ENABLE_ARPACK Activates/Deactivates configuration with ARPACK

Configuring XLIFE++ with ARPACK:

XLIFEPP_ENABLE_ARPACK Activates/Deactivates configuration with ARPACK

XLIFEPP_BLAS_LIB_DIR to specify the directory containing BLAS library

XLIFEPP_LAPACK_LIB_DIR to specify the directory containing LAPACK library

XLIFEPP_ARPACK_LIB_DIR to specify the directory containing ARPACK library

Configuring XLIFE++ with ARPACK:

XLIFEPP_ENABLE_ARPACK Activates/Deactivates configuration with ARPACK

XLIFEPP_BLAS_LIB_DIR to specify the directory containing BLAS library

XLIFEPP_LAPACK_LIB_DIR to specify the directory containing LAPACK library

XLIFEPP_ARPACK_LIB_DIR to specify the directory containing ARPACK library

Configuring XLIFE++ with UMFPACK:

XLIFEPP_ENABLE_UMFPACK Activates/Deactivates configuration with UMFPACK

Configuring XLIFE++ with ARPACK:

XLIFEPP_ENABLE_ARPACK Activates/Deactivates configuration with ARPACK

XLIFEPP_BLAS_LIB_DIR to specify the directory containing BLAS library

XLIFEPP_LAPACK_LIB_DIR to specify the directory containing LAPACK library

XLIFEPP_ARPACK_LIB_DIR to specify the directory containing ARPACK library

Configuring XLIFE++ with UMFPACK:

XLIFEPP_ENABLE_UMFPACK Activates/Deactivates configuration with UMFPACK

XLIFEPP_BLAS_LIB_DIR to specify the directory containing BLAS library

XLIFEPP_LAPACK_LIB_DIR to specify the directory containing LAPACK library

Configuring XLIFE++ with ARPACK:

XLIFEPP_ENABLE_ARPACK Activates/Deactivates configuration with ARPACK

XLIFEPP_BLAS_LIB_DIR to specify the directory containing BLAS library

XLIFEPP_LAPACK_LIB_DIR to specify the directory containing LAPACK library

XLIFEPP_ARPACK_LIB_DIR to specify the directory containing ARPACK library

Configuring XLIFE++ with UMFPACK:

XLIFEPP_ENABLE_UMFPACK Activates/Deactivates configuration with UMFPACK

XLIFEPP_BLAS_LIB_DIR to specify the directory containing BLAS library

XLIFEPP_LAPACK_LIB_DIR to specify the directory containing LAPACK library

XLIFEPP_UMFPACK_INCLUDE_DIR to specify the directory containing UMFPACK header

XLIFEPP_UMFPACK_LIB_DIR to specify the directory containing UMFPACK library

Configuring XLIFE++ with ARPACK:

XLIFEPP_ENABLE_ARPACK Activates/Deactivates configuration with ARPACK

XLIFEPP_BLAS_LIB_DIR to specify the directory containing BLAS library

XLIFEPP_LAPACK_LIB_DIR to specify the directory containing LAPACK library

XLIFEPP_ARPACK_LIB_DIR to specify the directory containing ARPACK library

Configuring XLIFE++ with UMFPACK:

XLIFEPP_ENABLE_UMFPACK Activates/Deactivates configuration with UMFPACK

XLIFEPP_BLAS_LIB_DIR to specify the directory containing BLAS library

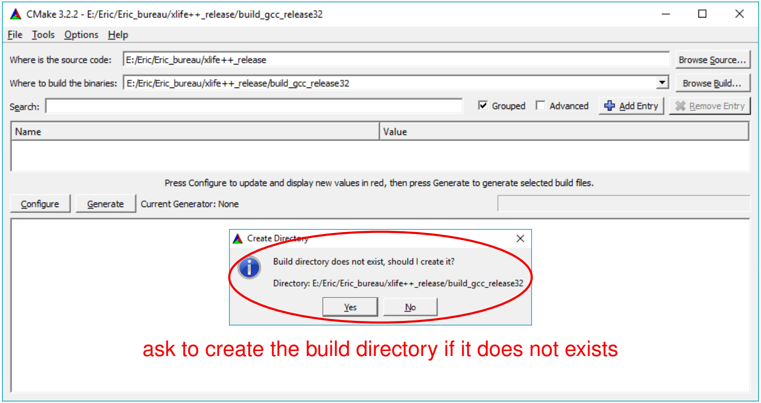
XLIFEPP_LAPACK_LIB_DIR to specify the directory containing LAPACK library

XLIFEPP_UMFPACK_INCLUDE_DIR to specify the directory containing UMFPACK header

XLIFEPP_UMFPACK_LIB_DIR to specify the directory containing UMFPACK library

XLIFEPP_SUITESPARSE_HOME_DIR to specify the home directory of SUITESPARSE, containing UMFPACK. This option is to be used if you compiled SUITESPARSE by yourself. In this case, UMFPACK will be searched in the UMFPACK subdirectory.

Building XLIFE++ - Start



CMake 3.2.2 - E:/Eric/Eric_bureau/xlife++_release/build_gcc_release32

Where is the source code: E:/Eric/Eric_bureau/xlife++_release

Where to build the binaries: E:/Eric/Eric_bureau/xlife++_release/build_gcc_release32

☑ Grouped □ Advanced

Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: None

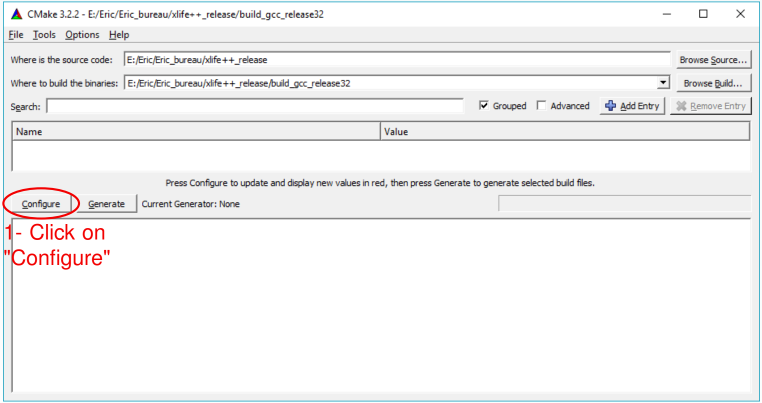
Create Directory

Build directory does not exist, should I create it?

Directory: E:/Eric/Eric_bureau/xlife++_release/build_gcc_release32

ask to create the build directory if it does not exists

Building XLIFE++ - Configure



CMake 3.2.2 - E:/Eric/Eric_bureau/xlife++_release/build_gcc_release32

Where is the source code: E:/Eric/Eric_bureau/xlife++_release

Where to build the binaries: E:/Eric/Eric_bureau/xlife++_release/build_gcc_release32

Search:

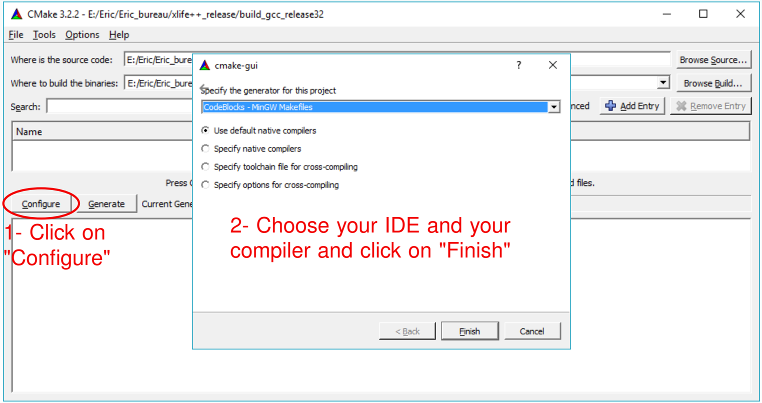
☑ Grouped □ Advanced

Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: None

1- Click on "Configure"

Building XLIFE++ - Configure



CMake 3.2.2 - E:/Eric/Eric_bureau/xlife++_release/build_gcc_release32

Where is the source code: E:/Eric/Eric_bure

Where to build the binaries: E:/Eric/Eric_bure

Search:

cmake-gui

Specify the generator for this project

CodeBlocks - MinGW Makefiles

● Use default native compilers

○ Specify native compilers

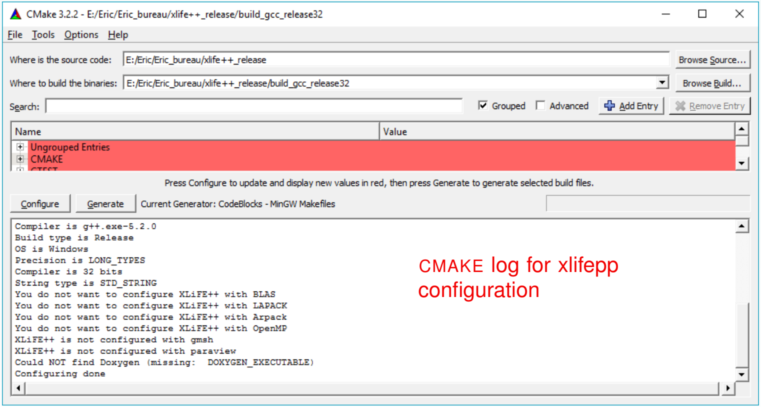
○ Specify toolchain file for cross-compiling

○ Specify options for cross-compiling

1- Click on "Configure"

2- Choose your IDE and your compiler and click on "Finish"

Building XLIFE++ - Configure



Where is the source code: E:/Eric/Eric_bureau/xlife++_release

Where to build the binaries: E:/Eric/Eric_bureau/xlife++_release/build_gcc_release32

Search:

☑ Grouped □ Advanced

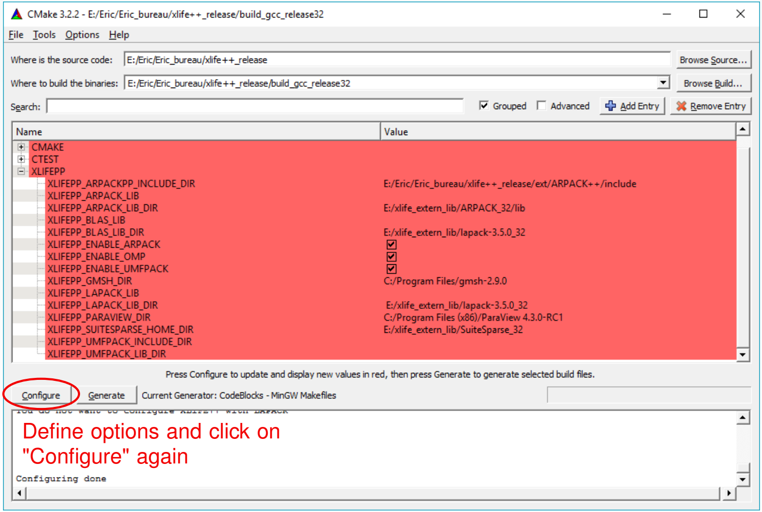
Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: CodeBlocks - MinGW Makefiles

```
Compiler is g++.exe-5.2.0
Build type is Release
OS is Windows
Precision is LONG_TYPES
Compiler is 32 bits
String type is STD_STRING
You do not want to configure XLiFE++ with BLAS
You do not want to configure XLiFE++ with LAPACK
You do not want to configure XLiFE++ with Arpack
You do not want to configure XLiFE++ with OpenMP
XLiFE++ is not configured with gmsh
XLiFE++ is not configured with paraview
Could NOT find Doxygen (missing:  DOXYGEN_EXECUTABLE)
Configuring done
```

CMAKE log for xlifepp configuration

Building XLIFE++ - Configure



Where is the source code: E:/Eric/Eric_bureau/xlife++_release

Where to build the binaries: E:/Eric/Eric_bureau/xlife++_release/build_gcc_release32

Search:

☑ Grouped ☐ Advanced

Name	Value
⊞ CMAKE	
⊞ CTEST	
⊟ XLIFEPP	
XLIFEPP_ARPACKPP_INCLUDE_DIR	E:/Eric/Eric_bureau/xlife++_release/ext/ARPACK++/include
XLIFEPP_ARPACK_LIB	
XLIFEPP_ARPACK_LIB_DIR	E:/xlife_extern_lib/ARPACK_32/lib
XLIFEPP_BLAS_LIB	
XLIFEPP_BLAS_LIB_DIR	E:/xlife_extern_lib/lapack-3.5.0_32
XLIFEPP_ENABLE_ARPACK	☑
XLIFEPP_ENABLE_OMP	☑
XLIFEPP_ENABLE_UMFPACK	☑
XLIFEPP_GMSH_DIR	C:/Program Files/gmsh-2.9.0
XLIFEPP_LAPACK_LIB	
XLIFEPP_LAPACK_LIB_DIR	E:/xlife_extern_lib/lapack-3.5.0_32
XLIFEPP_PARAVIEW_DIR	C:/Program Files (x86)/ParaView 4.3.0-RC1
XLIFEPP_SUITESPARSE_HOME_DIR	E:/xlife_extern_lib/SuiteSparse_32
XLIFEPP_UMFPACK_INCLUDE_DIR	
XLIFEPP_UMFPACK_LIB_DIR	

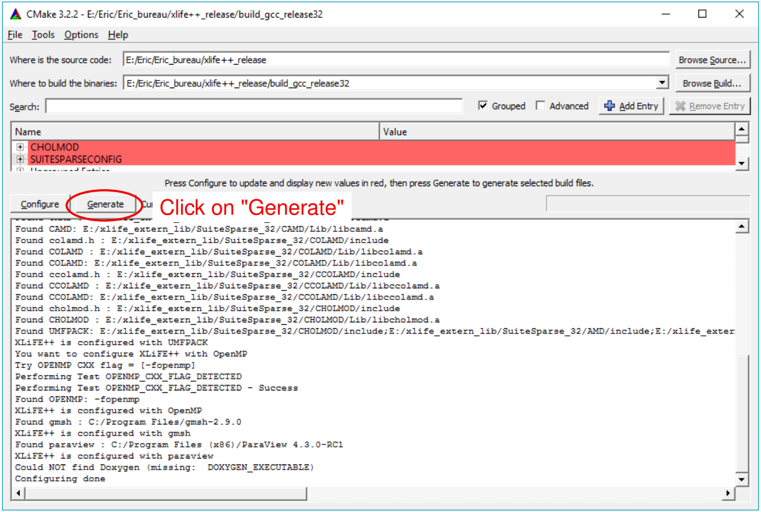
Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: CodeBlocks - MinGW Makefiles

```
You do not want to configure XLIFE++ with LAPACK
Configuring done
```

Define options and click on "Configure" again

Building XLIFE++ - Generate



CMake 3.2.2 - E:/Eric/Eric_bureau/xlife++_release/build_gcc_release32

Where is the source code: E:/Eric/Eric_bureau/xlife++_release

Where to build the binaries: E:/Eric/Eric_bureau/xlife++_release/build_gcc_release32

Search:

☑ Grouped □ Advanced

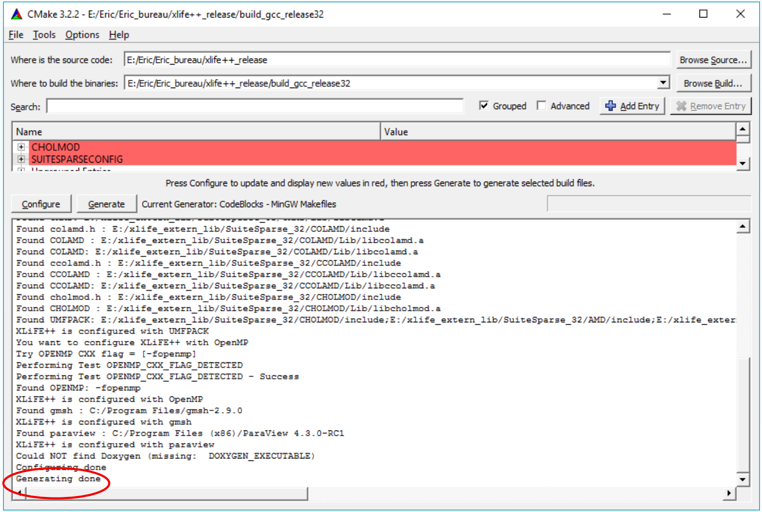
Name	Value
⊞ CHOLMOD	
⊞ SUITESPARSECONFIG	

Press Configure to update and display new values in red, then press Generate to generate selected build files.

Click on "Generate"

```
Found CAMD: E:/xlife_extern_lib/SuiteSparse_32/CAMD/Lib/libcamd.a
Found colamd.h : E:/xlife_extern_lib/SuiteSparse_32/COLAMD/include
Found COLAMD : E:/xlife_extern_lib/SuiteSparse_32/COLAMD/Lib/libcolamd.a
Found COLAMD: E:/xlife_extern_lib/SuiteSparse_32/COLAMD/Lib/libcolamd.a
Found ccolamd.h : E:/xlife_extern_lib/SuiteSparse_32/CCOLAMD/include
Found CCOLAMD : E:/xlife_extern_lib/SuiteSparse_32/CCOLAMD/Lib/libccolamd.a
Found CCOLAMD: E:/xlife_extern_lib/SuiteSparse_32/CCOLAMD/Lib/libccolamd.a
Found cholmod.h : E:/xlife_extern_lib/SuiteSparse_32/CHOLMOD/include
Found CHOLMOD : E:/xlife_extern_lib/SuiteSparse_32/CHOLMOD/Lib/libcholmod.a
Found UMFPACK: E:/xlife_extern_lib/SuiteSparse_32/CHOLMOD/include;E:/xlife_extern_lib/SuiteSparse_32/AMD/include;E:/xlife_exter
XLiFE++ is configured with UMFPACK
You want to configure XLiFE++ with OpenMP
Try OPENMP CXX flag = [-fopenmp]
Performing Test OPENMP_CXX_FLAG_DETECTED
Performing Test OPENMP_CXX_FLAG_DETECTED - Success
Found OPENMP: -fopenmp
XLiFE++ is configured with OpenMP
Found gmsh : C:/Program Files/gmsh-2.9.0
XLiFE++ is configured with gmsh
Found paraview : C:/Program Files (x86)/ParaView 4.3.0-RC1
XLiFE++ is configured with paraview
Could NOT find Doxygen (missing:  DOXYGEN_EXECUTABLE)
Configuring done
```

Building XLiFE++ - Generate



Where is the source code: E:/Eric/Eric_bureau/xlife++_release

Where to build the binaries: E:/Eric/Eric_bureau/xlife++_release/build_gcc_release32

Search:

Press Configure to update and display new values in red, then press Generate to generate selected build files.

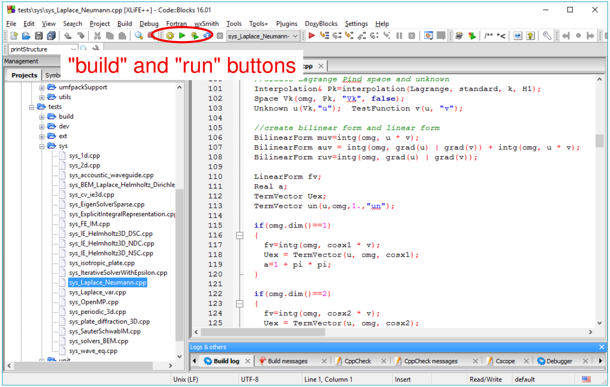
Current Generator: CodeBlocks - MinGW Makefiles

```
Found colamd.h : E:/xlife_extern_lib/SuiteSparse_32/COLAMD/include
Found COLAMD : E:/xlife_extern_lib/SuiteSparse_32/COLAMD/Lib/libcolamd.a
Found COLAMD: E:/xlife_extern_lib/SuiteSparse_32/COLAMD/Lib/libcolamd.a
Found ccolamd.h : E:/xlife_extern_lib/SuiteSparse_32/CCOLAMD/include
Found CCOLAMD : E:/xlife_extern_lib/SuiteSparse_32/CCOLAMD/Lib/libccolamd.a
Found CCOLAMD: E:/xlife_extern_lib/SuiteSparse_32/CCOLAMD/Lib/libccolamd.a
Found cholmod.h : E:/xlife_extern_lib/SuiteSparse_32/CHOLMOD/include
Found CHOLMOD : E:/xlife_extern_lib/SuiteSparse_32/CHOLMOD/Lib/libcholmod.a
Found UMFPACK: E:/xlife_extern_lib/SuiteSparse_32/CHOLMOD/include;E:/xlife_extern_lib/SuiteSparse_32/AMD/include;E:/xlife_exter
XLiFE++ is configured with UMFPACK
You want to configure XLiFE++ with OpenMP
Try OPENMP CXX flag = [-fopenmp]
Performing Test OPENMP_CXX_FLAG_DETECTED
Performing Test OPENMP_CXX_FLAG_DETECTED - Success
Found OPENMP: -fopenmp
XLiFE++ is configured with OpenMP
Found gmsh : C:/Program Files/gmsh-2.9.0
XLiFE++ is configured with gmsh
Found paraview : C:/Program Files (x86)/ParaView 4.3.0-RC1
XLiFE++ is configured with paraview
Could NOT find Doxygen (missing:  DOXYGEN_EXECUTABLE)
Configuring done
Generating done
```

Command-line: You just have to run `make libs` to compile libraries or `make` to compile libraries and tests executables

IDE mode: You select target "libs" to compile libraries, or "all" to compile libraries and test executables

To run a test:


"build" and "run" buttons

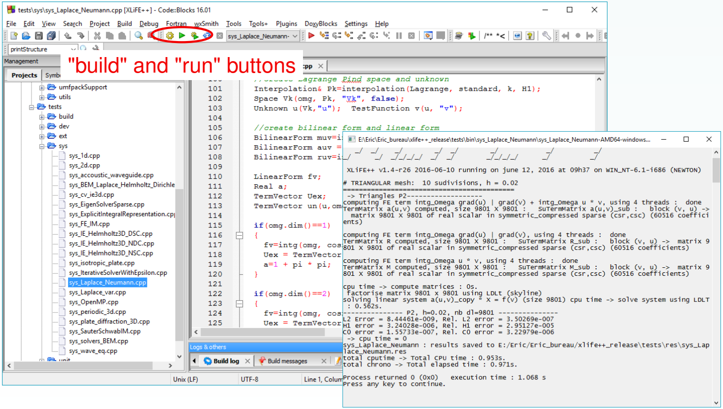
```
//create Lagrange Pind space and unknown
Interpolation& Pk=interpolation(Lagrange, standard, k, H1);
Space Vk(omg, Pk, "Vk", false);
Unknown u(Vk,"u");  TestFunction v(u, "v");

//create bilinear form and linear form
BilinearForm muv=intg(omg, u * v);
BilinearForm auv = intg(omg, grad(u) | grad(v)) + intg(omg, u * v);
BilinearForm ruv=intg(omg, grad(u) | grad(v));

LinearForm fv;
Real a;
TermVector Uex;
TermVector un(u,omg,1.,"un");

if(omg.dim()==1)
{
  fv=intg(omg, cosx1 * v);
  Uex = TermVector(u, omg, cosx1);
  a=1 + pi * pi;
}

if(omg.dim()==2)
{
  fv=intg(omg, cosx2 * v);
  Uex = TermVector(u, omg, cosx2);
```

tests\sys\sys_Laplace_Neumann.cpp [XLiFE++] - Code::Blocks 16.01
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
sys_Laplace_Neumann
printStructure
Management
Projects Symb
umfpackSupport
utils
tests
build
dev
ext
sys
sys_1d.cpp
sys_2d.cpp
sys_accoustic_waveguide.cpp
sys_BEM_Laplace_Helmholtz_Dirichle
sys_cv_ie3d.cpp
sys_EigenSolverSparse.cpp
sys_ExplicitIntegralRepresentation.cp
sys_FE_IM.cpp
sys_IE_Helmholtz3D_DSC.cpp
sys_IE_Helmholtz3D_NDC.cpp
sys_IE_Helmholtz3D_NSC.cpp
sys_isotropic_plate.cpp
sys_IterativeSolverWithEpsilon.cpp
sys_Laplace_Neumann.cpp
sys_Laplace_var.cpp
sys_OpenMP.cpp
sys_periodic_3d.cpp
sys_plate_diffraction_3D.cpp
sys_SauterSchwabIM.cpp
sys_solvers_BEM.cpp
sys_wave_eq.cpp
unit
Logs & others
Build log
Build messages
CppCheck
CppCheck messages
Cscope
Debugger
Unix (LF)
UTF-8
Line 1, Column 1
Insert
Read/Write
default

Command-line: You just have to run `make libs` to compile libraries or `make` to compile libraries and tests executables

IDE mode: You select target "libs" to compile libraries, or "all" to compile libraries and test executables

To run a test:


tests\sys\sys_Laplace_Neumann.cpp [XLiFE++] - Code::Blocks 16.01
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
sys_Laplace_Neumann-
printStructure
Management
Projects Symb
umfpackSupport
utils
tests
build
dev
ext
sys
sys_1d.cpp
sys_2d.cpp
sys_accoustic_waveguide.cpp
sys_BEM_Laplace_Helmholtz_Dirichle
sys_cv_ie3d.cpp
sys_EigenSolverSparse.cpp
sys_ExplicitIntegralRepresentation.cp
sys_FE_IM.cpp
sys_IE_Helmholtz3D_DSC.cpp
sys_IE_Helmholtz3D_NDC.cpp
sys_IE_Helmholtz3D_NSC.cpp
sys_isotropic_plate.cpp
sys_IterativeSolverWithEpsilon.cpp
sys_Laplace_Neumann.cpp
sys_Laplace_var.cpp
sys_OpenMP.cpp
sys_periodic_3d.cpp
sys_plate_diffraction_3D.cpp
sys_SauterSchwabIM.cpp
sys_solvers_BEM.cpp
sys_wave_eq.cpp
unit
Logs & others
Build log
Build messages
Unix (LF)
UTF-8
Line 1, Colum
E:\Eric\Eric_bureau\xlife++_release\tests\bin\sys_Laplace_Neumann\sys_Laplace_Neumann-AMD64-windows...

"build" and "run" buttons

```
//create Lagrange Pind space and unknown
Interpolation& Pk=interpolation(Lagrange, standard, k, H1);
Space Vk(omg, Pk, "Vk", false);
Unknown u(Vk,"u");  TestFunction v(u, "v");

//create bilinear form and linear form
BilinearForm muv=i
BilinearForm auv =
BilinearForm ruv=i

LinearForm fv;
Real a;
TermVector Uex;
TermVector un(u,om

if(omg.dim()==1)
{
  fv=intg(omg, cos
  Uex = TermVector
  a=1 + pi * pi;
}

if(omg.dim()==2)
{
  fv=intg(omg, cos
  Uex = TermVector
```

```
 XLiFE++ v1.4-r26 2016-06-10 running on june 12, 2016 at 09h37 on WIN_NT-6.1-i686 (NEWTON)

# TRIANGULAR mesh:  10 sudivisions, h = 0.02
===========================================
 -> Triangles P2------------------
computing FE term intg_Omega grad(u) | grad(v) + intg_Omega u * v, using 4 threads :  done
TermMatrix a(u,v) computed, size 9801 X 9801 :   SuTermMatrix a(u,v)_sub :   block (v, u) ->
  matrix 9801 X 9801 of real scalar in symmetric_compressed sparse (csr,csc) (60516 coeffici
ents)

computing FE term intg_Omega grad(u) | grad(v), using 4 threads :  done
TermMatrix R computed, size 9801 X 9801 :   SuTermMatrix R_sub :   block (v, u) ->  matrix 9
801 X 9801 of real scalar in symmetric_compressed sparse (csr,csc) (60516 coefficients)

computing FE term intg_Omega u * v, using 4 threads :  done
TermMatrix M computed, size 9801 X 9801 :   SuTermMatrix M_sub :   block (v, u) ->  matrix 9
801 X 9801 of real scalar in symmetric_compressed sparse (csr,csc) (60516 coefficients)

cpu time -> compute matrices : 0s.
 factorise matrix 9801 x 9801 using LDLt (skyline)
solving linear system a(u,v)_copy * X = f(v) (size 9801) cpu time -> solve system using LDLT
 : 0.562s.
--------------- P2, h=0.02, nb dl=9801 ---------------
L2 Error = 8.44461e-009, Rel. L2 error = 3.50269e-007
H1 error = 3.24028e-006, Rel. H1 error = 2.95127e-005
C0 error = 1.55733e-007, Rel. C0 error = 3.22979e-006
 -> cpu time = 0
sys_Laplace_Neumann : results saved to E:/Eric/Eric_bureau/xlife++_release\tests\res\sys_Lap
lace_Neumann.res
total cputime -> Total CPU time : 0.953s.
total chrono -> Total elapsed time : 0.971s.

Process returned 0 (0x0)   execution time : 1.068 s
Press any key to continue.
```

1 Downloading XLIFE++

2 Installing XLIFE++
- Requirements
- About CMAKE
- Building XLIFE++
- Compiling XLIFE++

3 Using XLIFE++
- Requirements
- Running user script
- About PARAVIEW

Requirements

- Basically, XLIFE++ compilation depends on the cross-platform builder CMAKE, available at `http://cmake.org`.
- To use the full abilities of mesh definition in XLIFE++, you may also install GMSH (`http://gmsh.info`).
- To plot solutions, you may install PARAVIEW (`http://www.paraview.org`) or GMSH.
- A C++ compiler
- If you use command line, you may want to add $XLIFEPP_HOME_DIR/bin to your path

Running user script

There are user scripts to prepare the user program that will use XLIFE++: xlifepp.sh on Unix/Linux and MacOS, xlifepp.bat on Windows. Scripts are physically in $XLIFEPP_HOME_DIR/bin directory

```
USAGE:
    xlifepp.sh --build [--interactive] [(--generate|--no-generate)]
    xlifepp.sh --build --non-interactive [(--generate|--no-generate)]
                       [--clean]
                       [--compiler <compiler>] [--directory <dir>]
                       [--generator-name <generator>]
                       [--build-type <build-type>]
                       [(--with-omp|--without-omp)]
    xlifepp.sh --help
    xlifepp.sh --version
```

Running user script

There are user scripts to prepare the user program that will use XLiFE++: xlifepp.sh on Unix/Linux and MacOS, xlifepp.bat on Windows. Scripts are physically in $XLIFEPP_HOME_DIR/bin directory

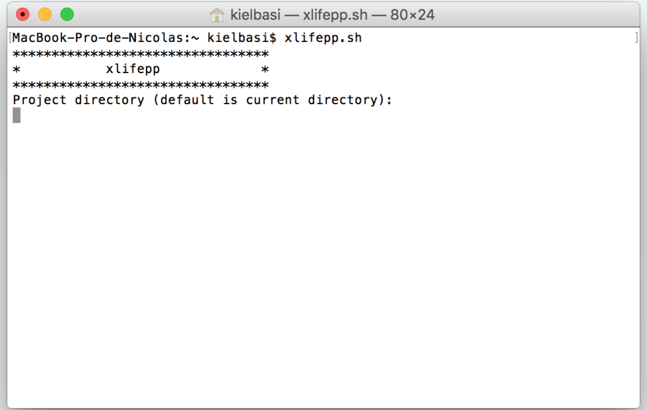
```
MAIN OPTIONS:
    --build, -b                   copy cmake files and eventually sample of
                                  main file
                                  and run cmake on it to prepare your so-called
                                  project directory. This is the default
    --clean, -cl                  clean project directory (remove CMake build files)
    --generate, -g                generate the project. Used with --build option.
                                  This is the default.
    --help, -help, -h             show the current help
    --interactive, -i             run xlifepp in interactive mode. Used with
                                  --build option. This is the default
    --non-interactive, -noi       run xlifepp in non interactive mode. Used with
                                  --build option
    --no-generate, -nog           prevent generation of your project. You will
                                  do it yourself.
    --version, -v                 print version number of XLiFE++ and its date
    --verbose-level <value>,      set the verbose level. Default value is 1
    -vl <value>
```

Running user script

There are user scripts to prepare the user program that will use XLIFE++: xlifepp.sh on Unix/Linux and MacOS, xlifepp.bat on Windows. Scripts are physically in $XLIFEPP_HOME_DIR/bin directory

```
OPTIONS FOR BUILD IN NON INTERACTIVE MODE:
    --build-type <value>,      set cmake build type (Debug, Release, ...).
    -bt <value>
    --cxx-compiler <value>,    set the C++ compiler to use.
    -cxx <value>
    --directory <dir>,         set the directory where you want to build
    -d <dir>                   your project
    --generator-name <name>,   set the cmake generator.
    -gn <name>
    -f <filename>,             copy <filename> as a main file for the user
    --main-file <filename>     project.
    -nof,                      do not copy the sample main.cpp file. This is
    --no-main-file             the default.
    --with-omp, -omp           activates OpenMP mode
    --without-omp, -nomp       deactivates OpenMP mode
```

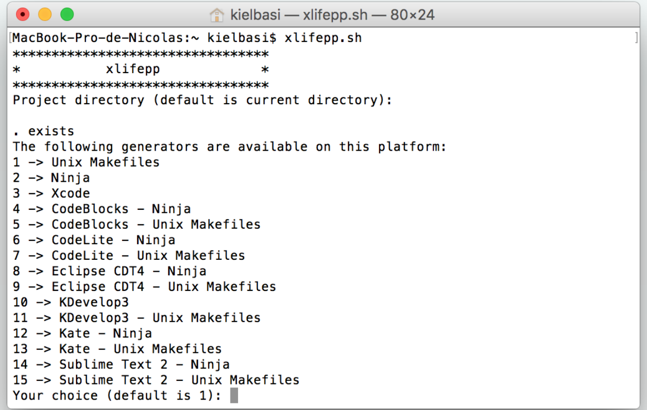
Running user script



kielbasi — xlifepp.sh — 80×24

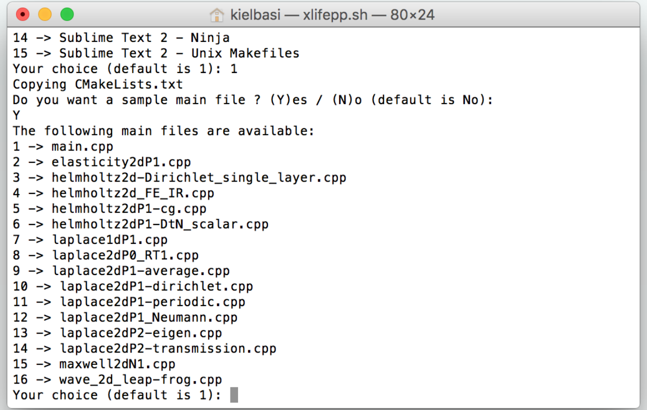
```
[MacBook-Pro-de-Nicolas:~ kielbasi$ xlifepp.sh
*********************************
*           xlifepp             *
*********************************
Project directory (default is current directory):
```

Running user script



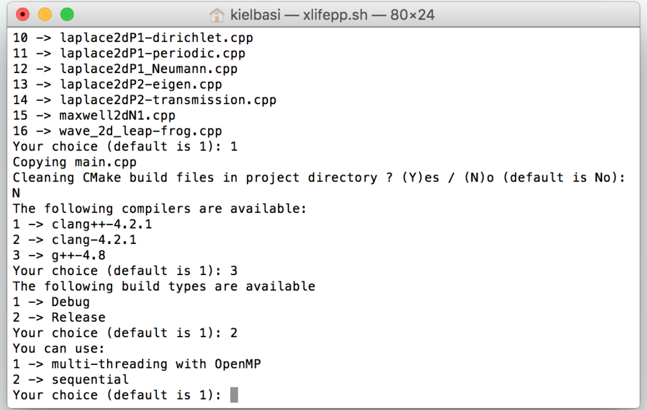
```
MacBook-Pro-de-Nicolas:~ kielbasi$ xlifepp.sh
*********************************
*           xlifepp            *
*********************************
Project directory (default is current directory):

. exists
The following generators are available on this platform:
1 -> Unix Makefiles
2 -> Ninja
3 -> Xcode
4 -> CodeBlocks - Ninja
5 -> CodeBlocks - Unix Makefiles
6 -> CodeLite - Ninja
7 -> CodeLite - Unix Makefiles
8 -> Eclipse CDT4 - Ninja
9 -> Eclipse CDT4 - Unix Makefiles
10 -> KDevelop3
11 -> KDevelop3 - Unix Makefiles
12 -> Kate - Ninja
13 -> Kate - Unix Makefiles
14 -> Sublime Text 2 - Ninja
15 -> Sublime Text 2 - Unix Makefiles
Your choice (default is 1):
```

Running user script



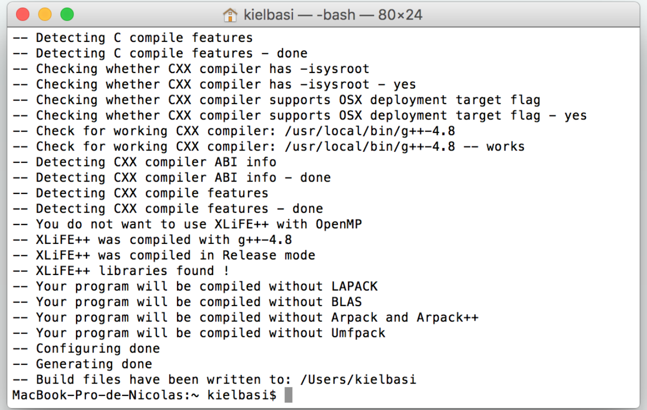
```
14 -> Sublime Text 2 - Ninja
15 -> Sublime Text 2 - Unix Makefiles
Your choice (default is 1): 1
Copying CMakeLists.txt
Do you want a sample main file ? (Y)es / (N)o (default is No):
Y
The following main files are available:
1 -> main.cpp
2 -> elasticity2dP1.cpp
3 -> helmholtz2d-Dirichlet_single_layer.cpp
4 -> helmholtz2d_FE_IR.cpp
5 -> helmholtz2dP1-cg.cpp
6 -> helmholtz2dP1-DtN_scalar.cpp
7 -> laplace1dP1.cpp
8 -> laplace2dP0_RT1.cpp
9 -> laplace2dP1-average.cpp
10 -> laplace2dP1-dirichlet.cpp
11 -> laplace2dP1-periodic.cpp
12 -> laplace2dP1_Neumann.cpp
13 -> laplace2dP2-eigen.cpp
14 -> laplace2dP2-transmission.cpp
15 -> maxwell2dN1.cpp
16 -> wave_2d_leap-frog.cpp
Your choice (default is 1): 
```

Running user script



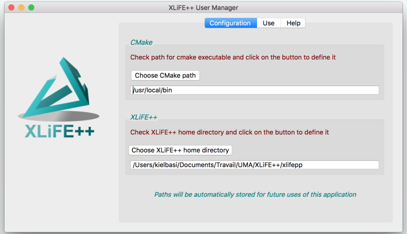
```
10 -> laplace2dP1-dirichlet.cpp
11 -> laplace2dP1-periodic.cpp
12 -> laplace2dP1_Neumann.cpp
13 -> laplace2dP2-eigen.cpp
14 -> laplace2dP2-transmission.cpp
15 -> maxwell2dN1.cpp
16 -> wave_2d_leap-frog.cpp
Your choice (default is 1): 1
Copying main.cpp
Cleaning CMake build files in project directory ? (Y)es / (N)o (default is No):
N
The following compilers are available:
1 -> clang++-4.2.1
2 -> clang-4.2.1
3 -> g++-4.8
Your choice (default is 1): 3
The following build types are available
1 -> Debug
2 -> Release
Your choice (default is 1): 2
You can use:
1 -> multi-threading with OpenMP
2 -> sequential
Your choice (default is 1):
```

Running user script



```
-- Detecting C compile features
-- Detecting C compile features - done
-- Checking whether CXX compiler has -isysroot
-- Checking whether CXX compiler has -isysroot - yes
-- Checking whether CXX compiler supports OSX deployment target flag
-- Checking whether CXX compiler supports OSX deployment target flag - yes
-- Check for working CXX compiler: /usr/local/bin/g++-4.8
-- Check for working CXX compiler: /usr/local/bin/g++-4.8 -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- You do not want to use XLiFE++ with OpenMP
-- XLiFE++ was compiled with g++-4.8
-- XLiFE++ was compiled in Release mode
-- XLiFE++ libraries found !
-- Your program will be compiled without LAPACK
-- Your program will be compiled without BLAS
-- Your program will be compiled without Arpack and Arpack++
-- Your program will be compiled without Umfpack
-- Configuring done
-- Generating done
-- Build files have been written to: /Users/kielbasi
MacBook-Pro-de-Nicolas:~ kielbasi$
```

Running user script



CMake

Check path for cmake executable and click on the button to define it

Choose CMake path

/usr/local/bin

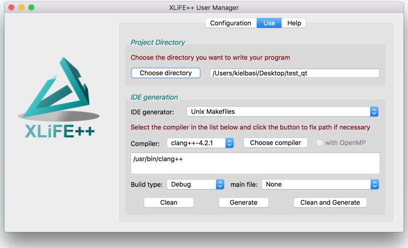
XLiFE++

Check XLiFE++ home directory and click on the button to define it

Choose XLiFE++ home directory

/Users/kielbasi/Documents/Travail/UMA/XLiFE++/xlifepp

Paths will be automatically stored for future uses of this application



Project Directory

Choose the directory you want to write your program

Choose directory

/Users/kielbasi/Desktop/test_qt

IDE generation

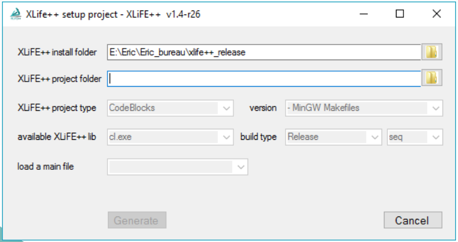
IDE generator: Unix Makefiles

Select the compiler in the list below and click the button to fix path if necessary

Compiler: clang++-4.2.1 Choose compiler □ with OpenMP

/usr/bin/clang++

Build type: Debug main file: None



XLiFE++ install folder E:\Eric\Eric_bureau\xlife++_release

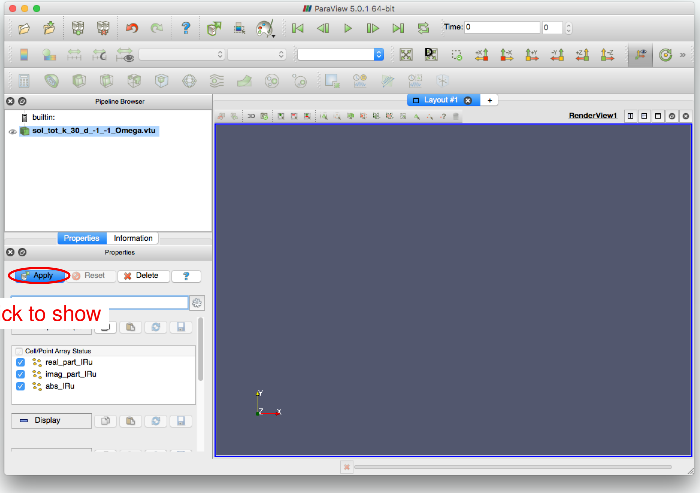
XLiFE++ project folder

XLiFE++ project type CodeBlocks version - MinGW Makefiles

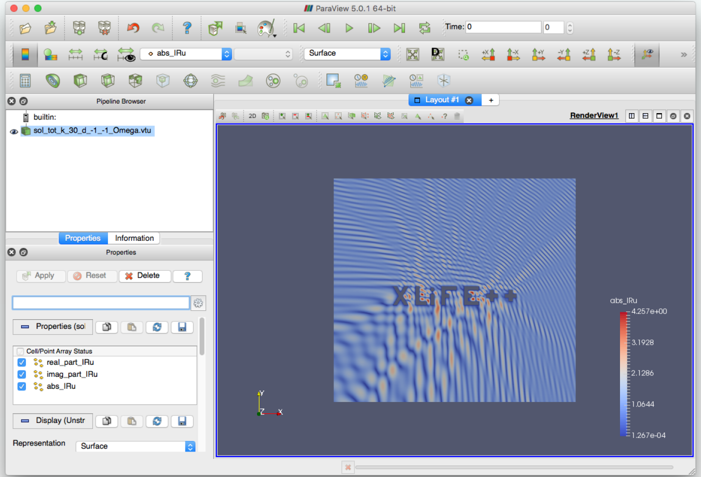
available XLiFE++ lib cl.exe build type Release seq

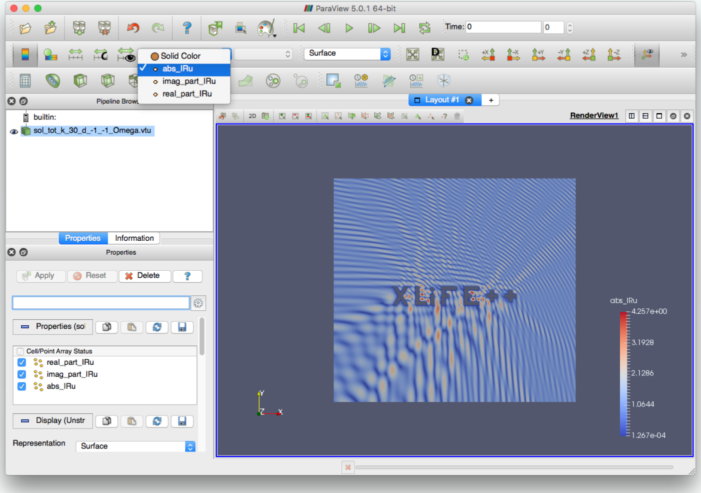
load a main file

- On MacOS, a Qt-based GUI application to manage the previous script. You can download it on the website
- On Windows, a very small application you can find in $XLIFEPP_HOME_DIR/bin



Click to show


ParaView 5.0.1 64-bit
Time: 0
abs_IRu
Surface
Pipeline Browser
builtin:
sol_tot_k_30_d_-1_-1_Omega.vtu
Layout #1
RenderView1
Properties
Information
Apply
Reset
Delete
Properties (sol
Cell/Point Array Status
real_part_IRu
imag_part_IRu
abs_IRu
Display (Unstr
Representation
Surface
XLiFE++
abs_IRu
4.257e+00
3.1928
2.1286
1.0644
1.267e-04
Y
Z
X



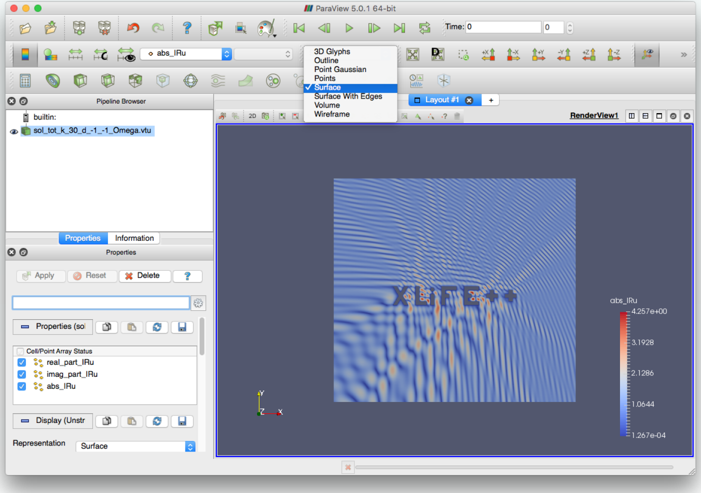
Time: 0

Surface

builtin:
sol_tot_k_30_d_-1_-1_Omega.vtu

Cell/Point Array Status
☑ real_part_IRu
☑ imag_part_IRu
☑ abs_IRu

Representation Surface



ParaView 5.0.1 64-bit

Time: 0

Pipeline Browser

builtin:

sol_tot_k_30_d_-1_-1_Omega.vtu

Cell/Point Array Status

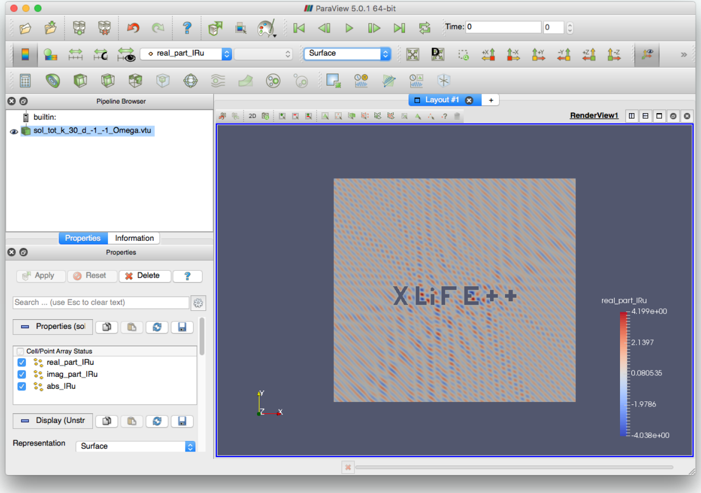
☑ real_part_lRu

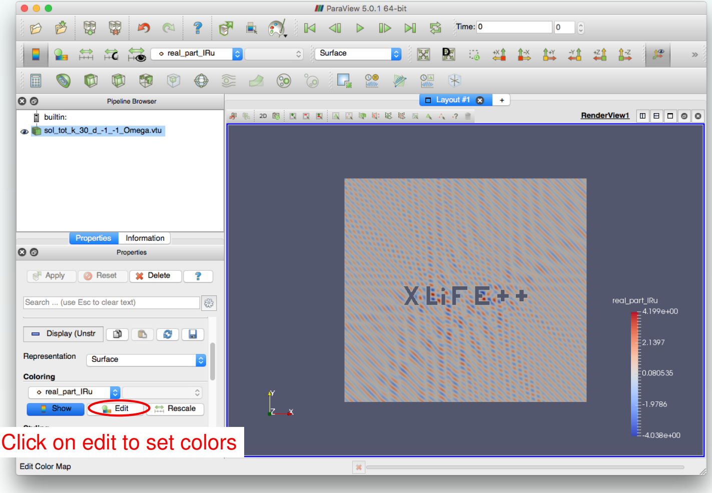
☑ imag_part_lRu

☑ abs_lRu

Representation: Surface

About ParaView

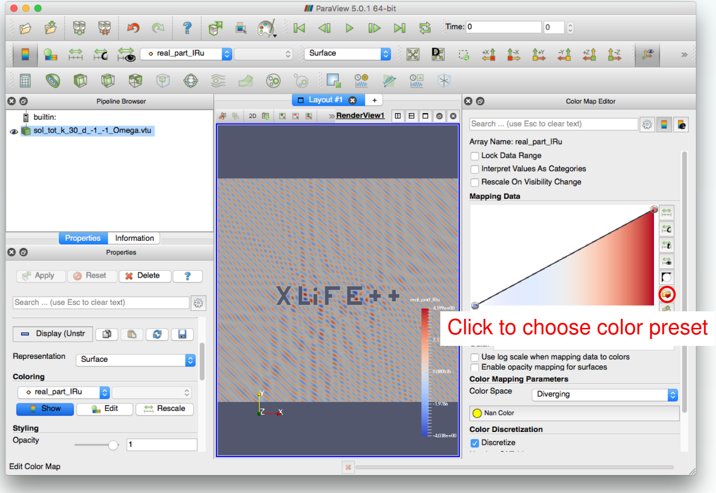




Representation: Surface

Coloring: real_part_IRu

Click on edit to set colors



ParaView 5.0.1 64-bit

Time: 0

Array Name: real_part_IRu

□ Lock Data Range
□ Interpret Values As Categories
□ Rescale On Visibility Change

Mapping Data

□ Use log scale when mapping data to colors
□ Enable opacity mapping for surfaces

Color Mapping Parameters

Color Space: Diverging

Color Discretization

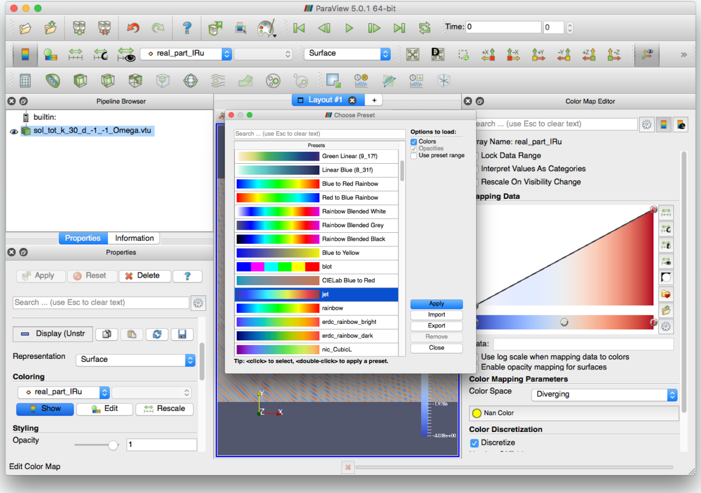
☑ Discretize

Representation: Surface

Coloring: real_part_IRu

Opacity: 1

Click to choose color preset



ParaView 5.0.1 64-bit

Time: 0

real_part_IRu

Surface

Pipeline Browser

builtin:
sol_tot_k_30_d_-1_-1_Omega.vtu

Properties

Representation: Surface

Coloring: real_part_IRu

Styling

Opacity: 1

Choose Preset

Options to load:
☑ Colors
☑ Opacities
☐ Use preset range

Presets:
- Green Linear (9_17f)
- Linear Blue (8_31f)
- Blue to Red Rainbow
- Red to Blue Rainbow
- Rainbow Blended White
- Rainbow Blended Grey
- Rainbow Blended Black
- Blue to Yellow
- blot
- CIELab Blue to Red
- jet
- rainbow
- erdc_rainbow_bright
- erdc_rainbow_dark
- nic_CubicL

Tip: <click> to select, <double-click> to apply a preset.

Color Map Editor

Array Name: real_part_IRu
☐ Lock Data Range
☐ Interpret Values As Categories
☐ Rescale On Visibility Change

Mapping Data

☐ Use log scale when mapping data to colors
☐ Enable opacity mapping for surfaces

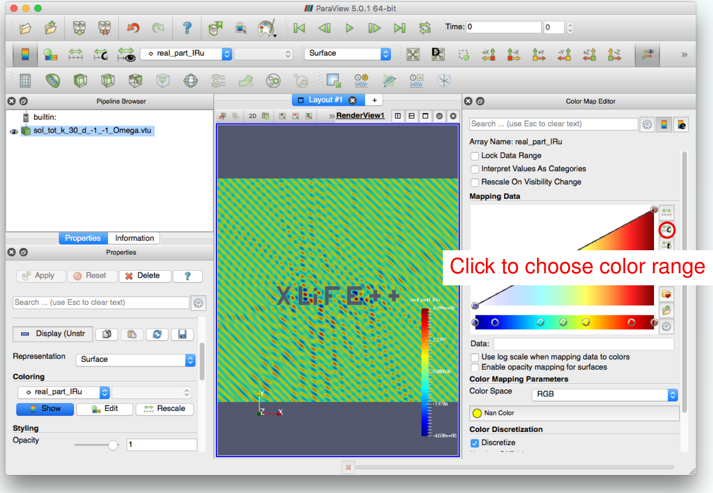
Color Mapping Parameters

Color Space: Diverging

Nan Color

Color Discretization

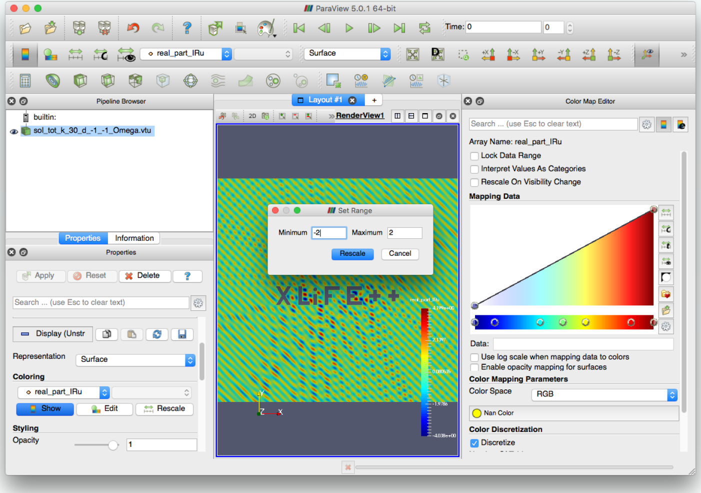
☑ Discretize



ParaView 5.0.1 64-bit

Click to choose color range

About ParaView



ParaView 5.0.1 64-bit

Time: 0

real_part_IRu

Surface

Pipeline Browser

builtin:

sol_tot_k_30_d_-1_-1_Omega.vtu

Layout #1

RenderView1

Set Range

Minimum -2 Maximum 2

Rescale Cancel

Properties Information

Apply Reset Delete ?

Search ... (use Esc to clear text)

Display (Unstr

Representation Surface

Coloring

real_part_IRu

Show Edit Rescale

Styling

Opacity 1

Color Map Editor

Search ... (use Esc to clear text)

Array Name: real_part_IRu

□ Lock Data Range

□ Interpret Values As Categories

□ Rescale On Visibility Change

Mapping Data

Data:

□ Use log scale when mapping data to colors

□ Enable opacity mapping for surfaces

Color Mapping Parameters

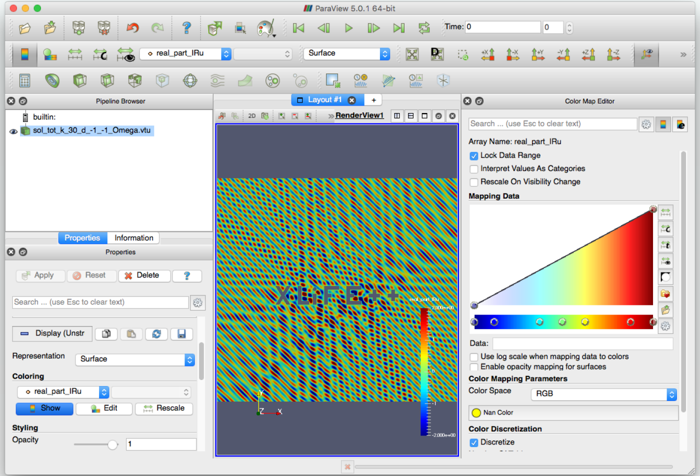
Color Space RGB

Nan Color

Color Discretization

☑ Discretize

About PARAVIEW



ParaView 5.0.1 64-bit

Time: 0

builtin:

sol_tot_k_30_d_-1_-1_Omega.vtu

Array Name: real_part_IRu

☑ Lock Data Range

☐ Interpret Values As Categories

☐ Rescale On Visibility Change

☐ Use log scale when mapping data to colors

☐ Enable opacity mapping for surfaces

Color Space: RGB

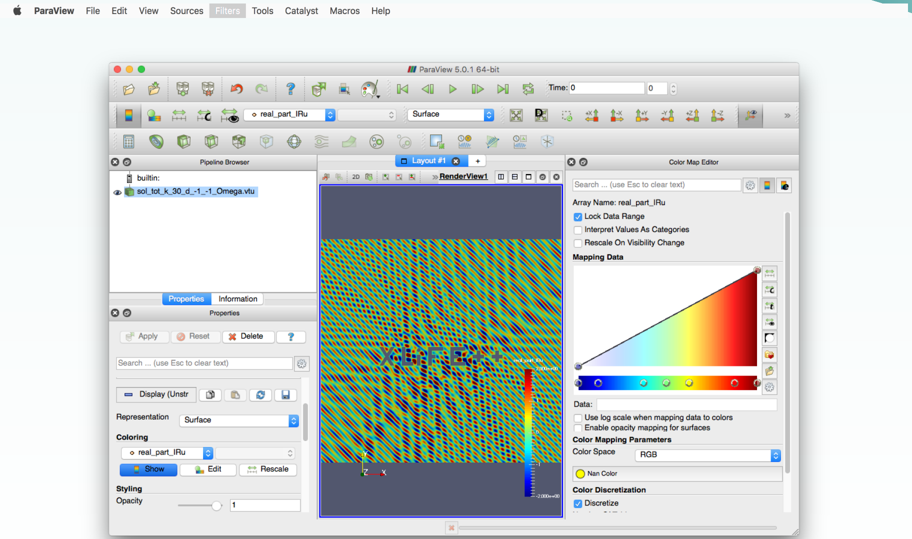
Representation: Surface

Coloring: real_part_IRu

Opacity: 1

☑ Discretize

About PARAVIEW



Time: 0 | 0

builtin:

sol_tot_k_30_d_-1_-1_Omega.vtu

Search ... (use Esc to clear text)

Display (Unstr

Representation Surface

Coloring

real_part_IRu

Styling

Opacity 1

Search ... (use Esc to clear text)

Array Name: real_part_IRu

☑ Lock Data Range

□ Interpret Values As Categories

□ Rescale On Visibility Change

Mapping Data

Data:

□ Use log scale when mapping data to colors

□ Enable opacity mapping for surfaces

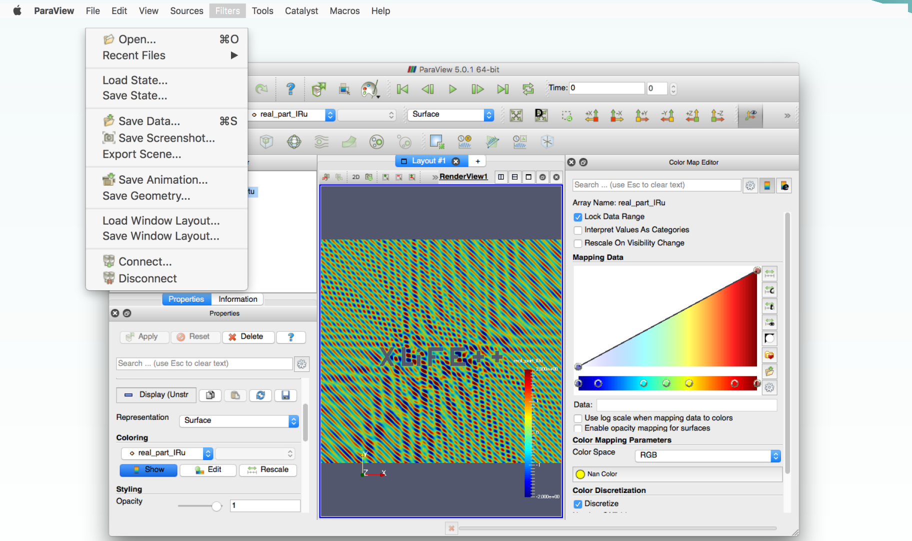
Color Mapping Parameters

Color Space RGB

Nan Color

Color Discretization

☑ Discretize



ParaView 5.0.1 64-bit

Time: 0

Representation: Surface

Coloring: real_part_IRu

Opacity: 1

Array Name: real_part_IRu

☑ Lock Data Range

□ Interpret Values As Categories

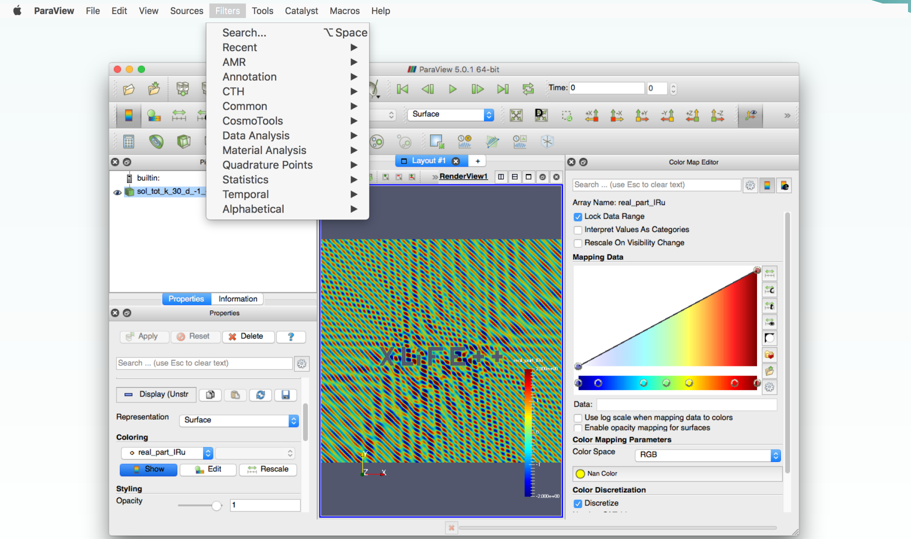
□ Rescale On Visibility Change

□ Use log scale when mapping data to colors

□ Enable opacity mapping for surfaces

Color Space: RGB

☑ Discretize



Time: 0

builtin:
sol_tot_k_30_d_-1_

Display (Unstr

Representation: Surface

Coloring: real_part_IRu

Styling

Opacity: 1

Array Name: real_part_IRu

☑ Lock Data Range

☐ Interpret Values As Categories

☐ Rescale On Visibility Change

Mapping Data

Data:

☐ Use log scale when mapping data to colors

☐ Enable opacity mapping for surfaces

Color Mapping Parameters

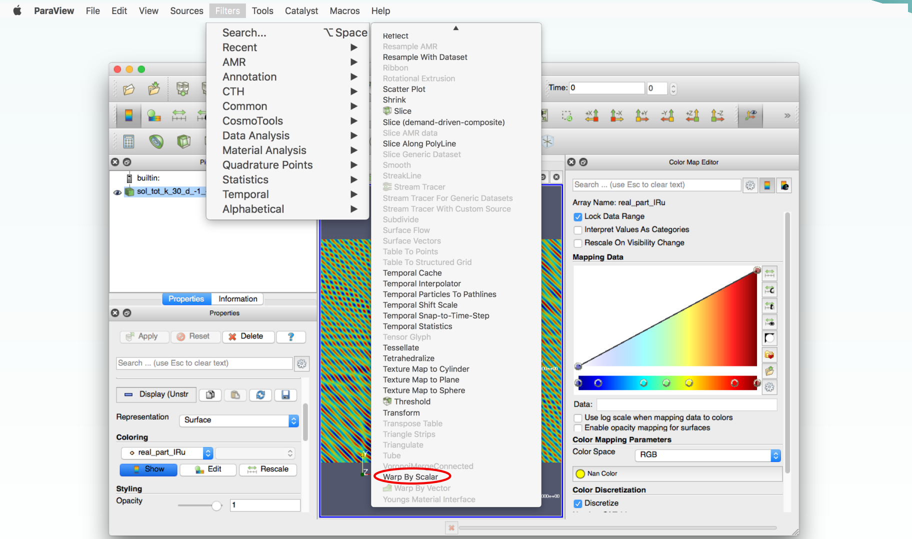
Color Space: RGB

Nan Color

Color Discretization

☑ Discretize

About PARAVIEW



Time: 0

Array Name: real_part_IRu

☑ Lock Data Range

□ Interpret Values As Categories

□ Rescale On Visibility Change

Data:

□ Use log scale when mapping data to colors

□ Enable opacity mapping for surfaces

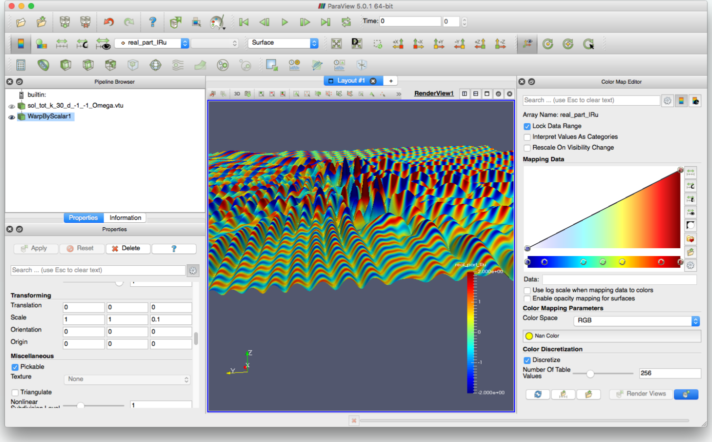
Color Space: RGB

Representation: Surface

Coloring: real_part_IRu

Opacity: 1

☑ Discretize



Time: 0 0

real_part_IRu

Pipeline Browser

- builtin:
- sol_tot_k_30_d_-1_-1_Omega.vtu
- WarpByScalar1

Properties | Information

Properties

Search ... (use Esc to clear text)

Transforming

Translation	0	0	0
Scale	1	1	0.1
Orientation	0	0	0
Origin	0	0	0

Miscellaneous

☑ Pickable

Texture: None

☐ Triangulate

Nonlinear Subdivision Level: 1

Color Map Editor

Search ... (use Esc to clear text)

Array Name: real_part_IRu

☑ Lock Data Range

☐ Interpret Values As Categories

☐ Rescale On Visibility Change

Mapping Data

Data:

☐ Use log scale when mapping data to colors

☐ Enable opacity mapping for surfaces

Color Mapping Parameters

Color Space: RGB

Nan Color

Color Discretization

☑ Discretize

Number Of Table Values: 256