

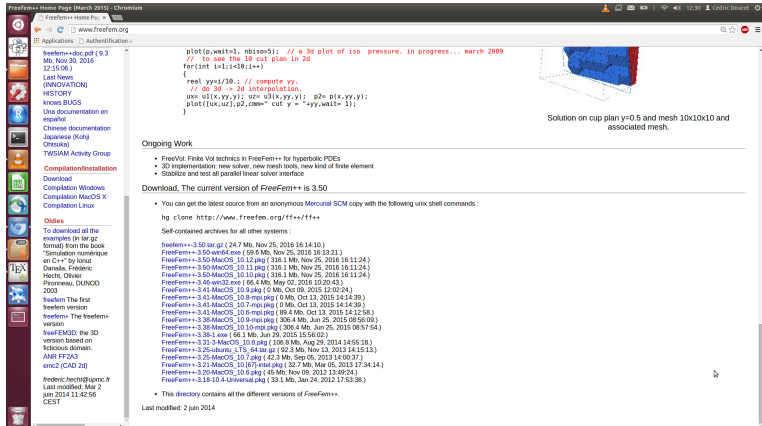
CMake support in FreeFem++

Cédric Doucet

Inria Paris

December 8, 2016

Current installation of FreeFem++



FreeFem++ Home Page [March 2015] - Chromium

FreeFem++ Home Page

www.freefem.org

Applications: Authentication

freefem-3d.pdf (9.3 Mb, Nov 30, 2015 12:15:06)

Last News (INNOVATION)

HISTORY

knows BUGS

Una documentation en espagnol

Chinese documentation

Japanese (Kohji Ohtsuka)

TWISAM Activity Group

Compilation/Installation

Download

Compilation Windows

Compilation MacOS X

Compilation Linux

Oldies

To download all the examples (in tar.gz format) from the book "Simulation numerique en C++" by Ivarut Dannele, Frederic Hecht, Olivier Pironneau, DUNOD 2003

freefem The first freefem version

freefem The freefem++ version

freeFEM3D: the 3D version based on fictitious domain.

AMR FF2A3

emc2 (CAD 2d)

frederic.hecht@univ-c.fr

Last modified: Mar 2 jun 2014 11:42:56 CEST

```
plot(p,wait=1,nbiso=5); // a 3d plot of iso pressure. In progress... march 2009
// to see the 10 cut plan in 2d
for(int i=1;i<10;i++)
{
    real yy=i/10.; // compute yy.
    // do 3d -> 2d interpolation.
    uxx= u(x,yy,y); uyy= u(y,yy,y); p2= p(x,yy,y);
    plot([ux,uy],p2,cmm=" cut y = "+yy,wait= 1);
}
```

Solution on cup plan y=0.5 and mesh 10x10x10 and associated mesh.

Ongoing Work

- FreeVot: Finite Vol techniques in FreeFem++ for hyperbolic PDEs
- 3D implementation: new solver, new mesh tools, new kind of finite element
- Stabilize and test all parallel linear solver interface

Download, The current version of FreeFem++ is 3.50

- You can get the latest source from an anonymous [Mercurial SCM](#) copy with the following unix shell commands:

```
hg clone http://www.freefem.org/ff++/ff++
```

Self-contained archives for all other systems:

File	Size	Version	Release Date
freefem++-3.50.tar.gz	24.7 Mb	Nov 25, 2016	16:14:10
FreeFem++-3.50.macosx	59.6 Mb	Nov 25, 2016	16:13:21
FreeFem++-3.50-MacOS_10.12.pkg	316.1 Mb	Nov 25, 2016	16:11:24
FreeFem++-3.50-MacOS_10.11.pkg	316.1 Mb	Nov 25, 2016	16:11:24
FreeFem++-3.50-MacOS_10.10.pkg	316.1 Mb	Nov 25, 2016	16:11:24
FreeFem++-3.46-wm32.exe	66.4 Mb	May 02, 2016	10:20:43
FreeFem++-3.41-MacOS_10.9.pkg	0 Mb	Oct 09, 2015	12:02:24
FreeFem++-3.41-MacOS_10.8.pkg	0 Mb	Oct 13, 2015	14:14:39
FreeFem++-3.41-MacOS_10.7.pkg	0 Mb	Oct 13, 2015	14:14:39
FreeFem++-3.41-MacOS_10.6.pkg	89.4 Mb	Oct 13, 2015	16:12:58
FreeFem++-3.38-MacOS_10.9-mpl.pkg	306.4 Mb	Jun 25, 2015	08:56:09
FreeFem++-3.38-MacOS_10.10-mpl.pkg	306.4 Mb	Jun 25, 2015	08:57:54
FreeFem++-3.35-1.exe	66.1 Mb	Jun 20, 2015	15:56:02
FreeFem++-3.31-3-MacOS_10.8.pkg	106.8 Mb	Aug 29, 2014	14:55:18
FreeFem++-3.25-ubuntu_LTS_64.tar.gz	92.3 Mb	Nov 13, 2013	14:15:13
FreeFem++-3.25-MacOS_10.7.pkg	42.3 Mb	Sep 05, 2013	14:00:37
FreeFem++-3.21-MacOS_10.67-intel.pkg	32.7 Mb	Mar 05, 2013	17:34:14
FreeFem++-3.20-MacOS_10.6.pkg	45 Mb	Nov 08, 2012	13:49:24
FreeFem++-3.18-10.4.4-universal.pkg	33.1 Mb	Jun 24, 2012	17:53:38

- This directory contains all the different versions of FreeFem++.

Last modified: 2 jun 2014

Current installation of FreeFem++

Installation from binaries/packages:

- ▶ very easy
- ▶ Mac, Ubuntu and Windows

Installation from sources:

- ▶ building with minimal dependencies
- ▶ automatic installation of many scientific modules
- ▶ platform-dependent (Linux, Mac, Windows)

Maintenance and evolution:

- ▶ development of Autoconf/configure is almost stopped
- ▶ manual porting on different platforms
- ▶ difficult to customize (31 pages configuration script)

Could we do better?

CMake

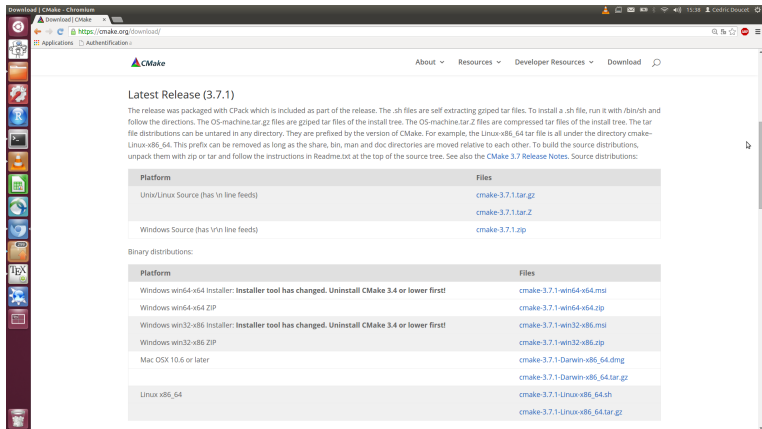
CMake: Cross-platform Make

- ▶ cross-platform
- ▶ based on Make
- ▶ free and open-source
- ▶ customizable
- ▶ Kitware product
- ▶ stable (15 years)
- ▶ well established
- ▶ active support

Other Kitware products: ITK, Paraview, VTK

Other young but promising tools: Scons, Waf, ...

Installation of CMake



The screenshot shows the CMake website's download page. The browser address bar shows the URL <https://cmake.org/download/>. The page title is "CMake" and the navigation bar includes links for "About", "Resources", "Developer Resources", and "Download".

Latest Release (3.7.1)

The release was packaged with CPack which is included as part of the release. The .sh files are self extracting gzipped tar files. To install a .sh file, run it with /bin/sh and follow the directions. The OS-machine.tar.gz files are gzipped tar files of the install tree. The OS-machine.tar.Z files are compressed tar files of the install tree. The tar file distributions can be untarred in any directory. They are prefixed by the version of CMake. For example, the Linux-x86_64 tar file is all under the directory cmake-Linux-x86_64. This prefix can be removed as long as the share, bin, man and doc directories are moved relative to each other. To build the source distributions, unpack them with zip or tar and follow the instructions in Readme.txt at the top of the source tree. See also the [CMake 3.7 Release Notes](#). Source distributions:

Platform	Files
Unix/Linux Source (has \n line feeds)	cmake-3.7.1.tar.gz cmake-3.7.1.tar.Z
Windows Source (has \r\n line feeds)	cmake-3.7.1.zip

Binary distributions:

Platform	Files
Windows win64-x64 Installer: Installer tool has changed. Uninstall CMake 3.4 or lower first!	cmake-3.7.1-win64-x64.msi
Windows win64-x64 ZIP	cmake-3.7.1-win64-x64.zip
Windows win32-x86 Installer: Installer tool has changed. Uninstall CMake 3.4 or lower first!	cmake-3.7.1-win32-x86.msi
Windows win32-x86 ZIP	cmake-3.7.1-win32-x86.zip
Mac OSX 10.6 or later	cmake-3.7.1-Darwin-x86_64.dmg cmake-3.7.1-Darwin-x86_64.tar.gz
Linux x86_64	cmake-3.7.1-Linux-x86_64.sh cmake-3.7.1-Linux-x86_64.tar.gz

- + installed by default on many systems
- + available as a package (apt-get, brew, conda, port, yum, ...)

Why is CMake better?

Reason 1: it is the same... or almost

Current installation from sources:

```
./configure --prefix=/home --enable-download=true  
make  
make check  
make install
```

CMake installation from sources:

```
cmake -D CMAKE_INSTALL_PREFIX=/home -D ENABLE_DOWNLOAD=true .  
make  
make check  
make install
```



```
cd /freeen/ubuntu16-all-packages-runner -demo/ff++/cmake_build
cmake -D ENABLE_DOWNLOAD=true -D FF_DOWNLOAD_DIR=/builds/download ..
-- The C compiler identification is GNU 5.4.0
-- The CXX compiler identification is GNU 5.4.0
-- The Fortran compiler identification is GNU 5.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working CXX compiler: /usr/bin/c++
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working Fortran compiler: /usr/bin/gfortran
-- Check for working Fortran compiler: /usr/bin/gfortran -- works
-- Detecting Fortran compiler ABI info
-- Detecting Fortran compiler ABI info - done
-- Checking whether /usr/bin/gfortran supports Fortran 90
-- Checking whether /usr/bin/gfortran supports Fortran 90 -- yes
-- Found MPI_C: /usr/lib/openmpi/lib/libmpi.so
-- Found MPI_CXX: /usr/lib/openmpi/lib/libmpi_cxx.so; /usr/lib/openmpi/lib/libmpi.so
-- Found MPI_Fortran: /usr/lib/openmpi/lib/libmpi_usempi_ignore_tkr.so; /usr/lib/openmpi/lib/libmpi_usempi_fh.so; /usr/lib/openmpi/lib/libmpi.so
-- Could NOT find PkgConfig (missing: PKG_CONFIG_EXECUTABLE)
-- Found CXX: /usr/include (found version "2.1")
-- Found HDF5: /usr/lib/x86_64-linux-gnu/hdf5/serial/lib/libhdf5.so; /usr/lib/x86_64-linux-gnu/libpthread.so; /usr/lib/x86_64-linux-gnu/libz.so; /builds/miniconda3/lib/libz.so; /usr/lib/x86_64-linux-gnu/libz.so; /usr/lib/x86_64-linux-gnu/libn.so (found version "1.8.16")
-- Looking for Fortran dgemv - found
-- Looking for pthread.h - found
-- Looking for pthread_create
-- Looking for pthread_create - not found
-- Looking for pthread_create in pthreads
-- Looking for pthread_create in pthreads - not found
-- Looking for pthread_create in pthread
-- Looking for pthread_create in pthread - found
-- Found Threads: TRUE
-- A library with BLAS API found.
-- Looking for Fortran cheev
-- Looking for Fortran cheev - found
-- A library with LAPACK API found.
-- Found SCOTCH: /usr/include/scotch
-- Configuring done
-- Generating done
-- Build files have been written to: /builds/demo/ff++/cmake_build
cd /freeen/ubuntu16-all-packages-runner -demo/ff++/cmake_build
```

Figure: Output of cmake

```

j@freefem-ubuntu16-all-packages-runner-jdemo/ff++/cnake_build$ cd /home/jdemon/cnake_builds
j@freefem-ubuntu16-all-packages-runner-jdemo/ff++/cnake_build$ make
Cleaning dependencies of target banglib
[60] Building COX object src/banglib/CMakeFiles/banglib.dir/Metric.cpp.o
[61] Building COX object src/banglib/CMakeFiles/banglib.dir/MeshQuad.cpp.o
[13] Building COX object src/banglib/CMakeFiles/banglib.dir/MeshRead.cpp.o
[18] Building COX object src/banglib/CMakeFiles/banglib.dir/R2.cpp.o
[19] Building COX object src/banglib/CMakeFiles/banglib.dir/MeshWrite.cpp.o
[26] Building COX object src/banglib/CMakeFiles/banglib.dir/SetOfE4.cpp.o
[23] Building COX object src/banglib/CMakeFiles/banglib.dir/wrIte_xdmf.cpp.o
[37] Building COX object src/banglib/CMakeFiles/banglib.dir/wrIte_hdf5.cpp.o
[30] Building COX object src/banglib/CMakeFiles/banglib.dir/MeshIo.cpp.o
[38] Building COX object src/banglib/CMakeFiles/banglib.dir/MeshGen.cpp.o
[40] Building COX object src/banglib/CMakeFiles/banglib.dir/MeshRaw.cpp.o
[40] Building COX object src/banglib/CMakeFiles/banglib.dir/QuadTree.cpp.o
[40] Building COX object src/banglib/CMakeFiles/banglib.dir/Mesh2.cpp.o
[50] Linking static library libbanglib.a
[50] Built target banglib
Cleaning dependencies of target lbfff
[60] Building COX object src/lbfff/CMakeFiles/lbfff.dir/_fentlb/P012_2d.cpp.o
[61] Building COX object src/lbfff/CMakeFiles/lbfff.dir/_fentlb/P012_1d.cpp.o
[76] Building COX object src/lbfff/CMakeFiles/lbfff.dir/_fentlb/fen.cpp.o
[76] Building COX object src/lbfff/CMakeFiles/lbfff.dir/_fentlb/FSpace.cpp.o
[76] Building COX object src/lbfff/CMakeFiles/lbfff.dir/_fentlb/splittimesp.cpp.o
[80] Building COX object src/lbfff/CMakeFiles/lbfff.dir/_fentlb/QquadTree.cpp.o
[80] Building COX object src/lbfff/CMakeFiles/lbfff.dir/_fentlb/Mesh3Dn.cpp.o
[80] Building COX object src/lbfff/CMakeFiles/lbfff.dir/_fentlb/BangFreeFem.cpp.o
[90] Building COX object src/lbfff/CMakeFiles/lbfff.dir/_fentlb/Mesh2Dn.cpp.o
[96] Building COX object src/lbfff/CMakeFiles/lbfff.dir/_fentlb/Pk3order.cpp.o
[100] Building COX object src/lbfff/CMakeFiles/lbfff.dir/_fentlb/glbs.cpp.o
[100] Building COX object src/lbfff/CMakeFiles/lbfff.dir/_fentlb/CheckPrt.cpp.o
[106] Building COX object src/lbfff/CMakeFiles/lbfff.dir/_fentlb/QquadTree.cpp.o
[110] Building COX object src/lbfff/CMakeFiles/lbfff.dir/_fentlb/Element_RT.cpp.o
[110] Building COX object src/lbfff/CMakeFiles/lbfff.dir/_fentlb/Drawing.cpp.o
[110] Building COX object src/lbfff/CMakeFiles/lbfff.dir/_fentlb/Element_P2H.cpp.o
[120] Building COX object src/lbfff/CMakeFiles/lbfff.dir/_fentlb/FSpacecn.cpp.o
[120] Building COX object src/lbfff/CMakeFiles/lbfff.dir/_fentlb/P012_3d.cpp.o
[120] Building COX object src/lbfff/CMakeFiles/lbfff.dir/_fentlb/mshgls.cpp.o
[130] Building COX object src/lbfff/CMakeFiles/lbfff.dir/_fentlb/mshglsFormular.cpp.o
[130] Building COX object src/lbfff/CMakeFiles/lbfff.dir/_fentlb/MeshIndn.cpp.o
[140] Building COX object src/lbfff/CMakeFiles/lbfff.dir/lxex.cpp.o
[140] Building COX object src/lbfff/CMakeFiles/lbfff.dir/Ignat.cpp.o
[146] Building COX object src/lbfff/CMakeFiles/lbfff.dir/array_long.cpp.o
[150] Building COX object src/lbfff/CMakeFiles/lbfff.dir/array_real.cpp.o
[150] Building COX object src/lbfff/CMakeFiles/lbfff.dir/tqfen.cpp.o
[160] Building COX object src/lbfff/CMakeFiles/lbfff.dir/environment.cpp.o
[160] Building COX object src/lbfff/CMakeFiles/lbfff.dir/P11toValue.cpp.o
[160] Building COX object src/lbfff/CMakeFiles/lbfff.dir/Ignesh.cpp.o
[170] Building COX object src/lbfff/CMakeFiles/lbfff.dir/Ignesh3.cpp.o
[170] Building COX object src/lbfff/CMakeFiles/lbfff.dir/Function.cpp.o
[170] Building COX object src/lbfff/CMakeFiles/lbfff.dir/CoeffAllnc.cpp.o
[180] Building COX object src/lbfff/CMakeFiles/lbfff.dir/load.cpp.o
[180] Building COX object src/lbfff/CMakeFiles/lbfff.dir/glmesh2D.cpp.o
[190] Building COX object src/lbfff/CMakeFiles/lbfff.dir/mis993Tar.cpp.o
[190] Building COX object src/lbfff/CMakeFiles/lbfff.dir/WFFPack_Solver.cpp.o
[200] Building COX object src/lbfff/CMakeFiles/lbfff.dir/problem.cpp.o
[200] Building COX object src/lbfff/CMakeFiles/lbfff.dir/initFuncnt.cpp.o
[200] Building COX object src/lbfff/CMakeFiles/lbfff.dir/string_def.cpp.o
[200] Building COX object src/lbfff/CMakeFiles/lbfff.dir/array_complex.cpp.o

```

Figure: Output of make

Reason 2: generators

```
cmake -G "Visual Studio 15 2017 Win64" ..
```

Command-Line tools:

- ▶ Borland
- ▶ JOM
- ▶ MinGW
- ▶ MSYS
- ▶ Ninja
- ▶ NMake
- ▶ Unix
- ▶ Watcom

IDEs:

- ▶ Codeblocks
- ▶ CodeLite
- ▶ Eclipse CDT4
- ▶ Kate
- ▶ KDevelop3
- ▶ SublimeText 2
- ▶ Visual Studio
- ▶ XCode

Note: some IDEs can parse CMake scripts (e.g. NetBeans)

Reason 3: expressive high-level language

CMakeLists.txt:

```
CMAKE_MINIMUM_REQUIRED(VERSION 2.8)
PROJECT(HELLO C)
FILE(GLOB SOURCE_FILES *.c)
ADD_EXECUTABLE(hello ${SOURCE_FILES})
```

Makefile:

```
CC=gcc
EXEC=hello
SRC= $(wildcard *.c)
OBJ= $(SRC:.c=.o)

all: $(EXEC)

hello: $(OBJ)
$(CC) -o $@ $^ $(LDFLAGS)

main.o: hello.h

%.o: %.c
$(CC) -o $@ -c $< $(CFLAGS)
```

configure.ac:

```
ff_uname='uname'
case $ff_uname in
    CYGWIN*|MINGW*|MSYS_NT*)
        ff_suffix_dylib="dll"
        ff_suffix_dylib_a="dll.a";;
    Darwin)
        ff_suffix_dylib="dylib"
        ff_suffix_dylib_a="dylib";;
    *)
        ff_suffix_dylib="so";
        ff_suffix_dylib_a="so";
esac
```

Reason 4: packaging is made easy

`cpack .`

Linux:

- ▶ Debian
- ▶ Fedora
- ▶ Red Hat
- ▶ Ubuntu

Mac:

- ▶ Bundle
- ▶ DMG

Windows:

- ▶ Cygwin
- ▶ NSIS
- ▶ WIX

Note: cross-compilation is possible

```
CMAKE_MINIMUM_REQUIRED(VERSION 2.6)

PROJECT(FreeFem)

ADD_EXECUTABLE(ff++ freefem.cxx)
INSTALL(TARGETS ff++ DESTINATION freefem++/bin)

SET(CPACK_GENERATOR "DEB")
SET(CPACK_DEBIAN_PACKAGE_MAINTAINER "Cedric Doucet")

INCLUDE(CPack)

# To use this:
# make package
# sudo dpkg -i ff++-0.1.1-Linux.deb

# This will result in the file:
#/usr/freefem++/bin/ff++
```

Note: it seems possible to let CMake list dependencies

Reason 5: Customisable detection of modules

FindScotch.cmake:

```
SET(SCOTCH_FOUND False)
FIND_PATH(SCOTCH_INCLUDES NAMES scotch.h
          PATHS /usr/include
          PATH_SUFFIXES scotch)
FIND_LIBRARY(SCOTCH_LIBRARIES NAMES scotch
            PATHS /usr/lib)
IF(SCOTCH_INCLUDES AND SCOTCH_LIBRARIES)
  SET(SCOTCH_FOUND True)
ENDIF(SCOTCH_INCLUDES AND SCOTCH_LIBRARIES)
```

Usage:

```
FIND_PACKAGE(Scotch)
IF(SCOTCH_FOUND)
  ADD_LIBRARY(scotch SHARED scotch.cpp)
  INCLUDE_DIRECTORIES(/usr/include/scotch)
ENDIF()
```

Note: CMake provides such files for BLAS, LAPACK, MPI, ...

Reason 5: Customisable installation of modules

InstallIPOPT.cmake:

```
SET(LIBNAME IPOPT)
SET(URL http://www.coin-or.org/download/source/Ipopt/Ipopt-3.12.4.tgz)
SET(URL_MD5 12a8ecaff8dd90025ddea6c65b49cb03)
SET(CONFIGURE_COMMAND ../src/configure --prefix=<INSTALL_DIR>
                        CXX=${CMAKE_CXX_COMPILER})
SET(BUILD_COMMAND make)
SET(INSTALL_COMMAND make install)
```

Usage:

```
INSTALL_PACKAGE(IPOPT)
ADD_LIBRARY(ff-Ipopt SHARED ff-Ipopt.cpp)
ADD_DEPENDENCIES(ff-Ipopt ipopt)
INCLUDE_DIRECTORIES(${IPOPT_INCLUDES})
TARGET_LINK_LIBRARIES(${IPOPT_LIBRARIES})
```

Note: targets and variables are automatically generated

When will CMake support be available?

CMake support is still in progress...

Core building without any download	Done
Automatic installation process	Done
Core building with downloads	Done
Test building	Done
Recipe files for external libraries	Doing
Find files for external libraries	Doing
Porting	Doing
Continuous integration	Doing
Automatic testing	Todo
Packaging	Todo
Documentation	Todo

but will be available soon!