

# *Elmer/Ice* Git + *cmake*

Thomas Zwinger

Elmer/Ice user meeting, EGU  
GA 2015



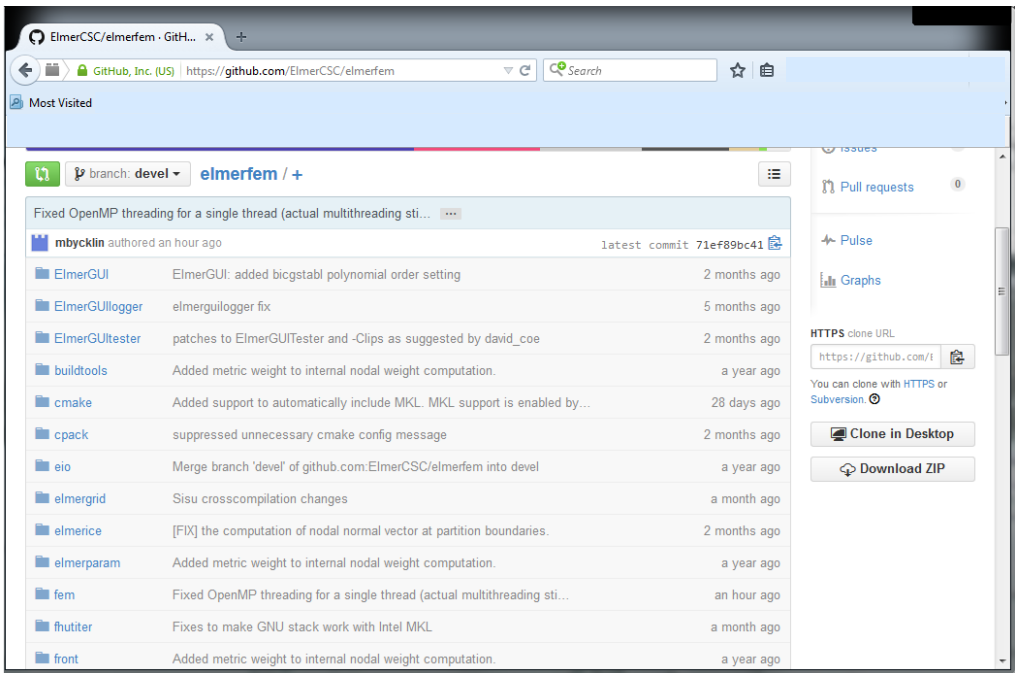
CSC-IT CENTER FOR SCIENCE

CSC

# *New distribution system*

- SVN (on SourceForge) is frozen since February 2015
- Git: <http://github.com> under: ElmerCSC/elmerfem
- Hence: <https://github.com/ElmerCSC/elmerfem>
- Git works different than SVN:
  - Distributed repositories (including local repositories)
  - Works with branches
    - We have a few of them in Elmer, the main Elmer branch is *devel*
    - Recommended to use *elmerice* branch, as Elmer/Ice related updates preferentially are pushed to this branch

# *New distribution system*

A screenshot of a web browser displaying the GitHub repository page for 'ElmerCSC/elmerferm'. The browser's address bar shows the URL 'https://github.com/ElmerCSC/elmerferm'. The repository page shows the current branch as 'devel'. A commit message is visible: 'Fixed OpenMP threading for a single thread (actual multithreading sti...'. Below this, a list of recent commits is shown, each with a folder icon, the commit message, and the time since it was made. The commits are: 'mbycklin authored an hour ago' (latest commit 71ef89bc41), 'ElmerGUI' (added bicgstabl polynomial order setting, 2 months ago), 'ElmerGUIlogger' (elmerguilogger fix, 5 months ago), 'ElmerGUITester' (patches to ElmerGUITester and -Clips as suggested by david\_coe, 2 months ago), 'buildtools' (Added metric weight to internal nodal weight computation., a year ago), 'cmake' (Added support to automatically include MKL. MKL support is enabled by..., 28 days ago), 'cpack' (suppressed unnecessary cmake config message, 2 months ago), 'eio' (Merge branch 'devel' of github.com:ElmerCSC/elmerferm into devel, a year ago), 'elmergrid' (Sisu crosscompilation changes, a month ago), 'elmerice' ([FIX] the computation of nodal normal vector at partition boundaries., 2 months ago), 'elmerparam' (Added metric weight to internal nodal weight computation., a year ago), 'fem' (Fixed OpenMP threading for a single thread (actual multithreading sti..., an hour ago), 'fnutiter' (Fixes to make GNU stack work with Intel MKL., a month ago), and 'front' (Added metric weight to internal nodal weight computation., a year ago). On the right side of the page, there are sections for 'Pull requests' (0), 'Pulse', 'Graphs', 'HTTPS clone URL' (https://github.com/t), and buttons for 'Clone in Desktop' and 'Download ZIP'.

# *How to obtain source code?*

- “I am just interested in obtaining the source code, what do I have to do?”
  - Previous: `svn checkout svn://svn.code.sf.net/p/elmerfem/code/trunk/`
  - New: `git clone https://github.com/ElmerCSC/elmerfem.git`
- “I want to contribute to the development, hence also submit changes to the repository”
  1. Tell us (=CSC) your GitHub ID
  2. Under certain circumstances we add you to the developer list
  3. Get yourself familiar (if not already) with Git
  4. If you touch stuff outside `elmerice -directory`, sign the IP waiver agreement with CSC

# Git

- Fetching repository and linking local *elmerice* branch to original:

```
git clone git://www.github.com/ElmerCSC/elmerfem
```

- Check which branch you are in: `git status` or

```
git branch -a
```

- Changing to other branch (and tying it to the GitHub):


```
git checkout --track origin/elmerice
```

- How to update?: `git pull`

# *Git policies*

1. Elmer/Ice specific changes (i.e., inside sub-directory *elmerice*) to *elmerice*-branch
2. also changes that are induced by Elmer/Ice to out-of-*elmerice* sub-directories should preferably be pushed from within *elmerice* branch
3. everyone CSC already trusted with GitHub access can do a merge from *devel* to *elmerice* branch (just check stuff in advance)
4. merges back from *elmerice* to *devel* branch should be only made by people from inside CSC. You can make a **pull request** (<https://help.github.com/articles/creating-a-pull-request/> )
5. try to avoid procedures like cherry-picking and rebasing – if necessary, talk to people at CSC first

# *New configuration system - Cmake*

- Moving from GNU autotools to cmake
  - Inside elmerfem-directory, make a build-directory:  
`mkdir builddir`
  - From within builddir try 
  - This requires that you have
    1. C and F90 compiler
    2. Hypre + Mumps installed
  - Thereafter:  
`make && make install`

```
cmake ../ \
-DMAKE_INSTALL_PREFIX=/elmer/installation/path \
-DMAKE_C_COMPILER=/path/to/compiler/executable \
-DMAKE_Fortran_COMPILER=/path/to/compiler/executable \
-DWITH_MPI:BOOL=TRUE \
-DWITH_Mumps:BOOL=TRUE \
-DWITH_Hypre:BOOL=TRUE \
-DWITH_Trilinos:BOOL=FALSE \
-DWITH_ELMERGUI:BOOL=FALSE \
-DWITH_ElmerIce:BOOL=TRUE
```

# Cmake

- Toolchain files can be used mainly in cross-platform compilations
  - Examples under `elmerfem/cmake/Toolchains`
  - Include with  
-  
`DCMAKE_TOOLCHAIN_FILE=/path/to/toolchainfile.cmake`
- Else use pre-cache files
  - Include with:  
`-C file.cmake`

```
# CMake toolchain file for building with GNU compilers
#
# Author: Mikko Byckling, CSC - IT Center for Science Ltd.
# Version: 0.1

SET(CMAKE_SYSTEM_NAME Linux)
SET(CMAKE_SYSTEM_PROCESSOR x86_64)
SET(CMAKE_SYSTEM_VERSION 1)

# Specify the cross compilers (serial)
SET(CMAKE_C_COMPILER gcc)
SET(CMAKE_Fortran_COMPILER gfortran)
SET(CMAKE_CXX_COMPILER g++)

# Specify the cross compilers (parallel)
SET(MPI_C_COMPILER mpicc)
SET(MPI_CXX_COMPILER mpic++)
SET(MPI_Fortran_COMPILER mpif90)

# Compilation flags (i.e. with optimization)
SET(CMAKE_C_FLAGS "-O3 -g -m64 -ftree-vectorize \
-funroll-loops" CACHE STRING "")
SET(CMAKE_CXX_FLAGS "-O3 -g -m64 -ftree-vectorize\
-funroll-loops" CACHE STRING "")
SET(CMAKE_Fortran_FLAGS "-O3 -g -m64 -ftree-vectorize\
-funroll-loops" CACHE STRING "")
```

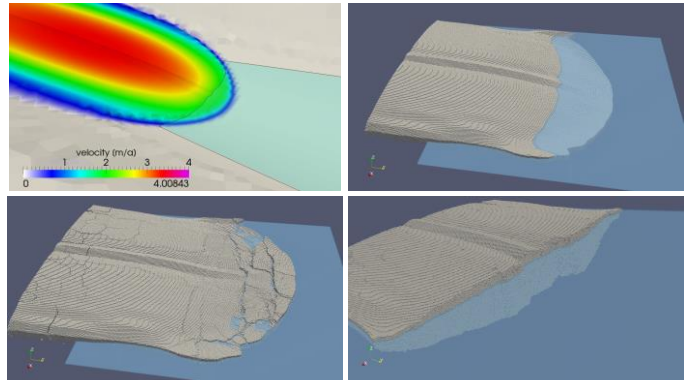


# *Cmake worked on ...*

- Linux Mint 17 (including Hypre and MUMPS, but Debian Trilinos did not work)
- Fedora (on CSC's HP-blade cluster taito.csc.fi)
- Cray Compute Linux (cross compilation) on CSC's Cray-XE cluster
- Windows 7
- OS X Maverik (instructions will follow in near future)
- If you have a toolchain or pre-cache file for a certain platform, please share ([zwinger@csc.fi](mailto:zwinger@csc.fi))

# Coupling of continuum model and DPM

- ❑ Glacier built-up with Elmer/Ice (synthetic "Svalbard-ish" setup)
- ❑ Transfer of tongue geometry into DPM
- ❑ DPM run (~1 Mio. Particles, 64 processes, 4h)
- ❑ Connectivity algorithm to detect new ice-front
- ❑ Back-transfer into Elmer/Ice



# *Urgently needed developments in Elmer/Ice?*

- **Transport dominated physics:**

- Passive tracer transport
- Age/depth
- Kinematic free surface equation
- FreeSurfaceSolver
  - cheating by using standard Galerkin
- Discontinuous Galerkin (troublesome)
- Semi-Lagrangian method
  - Issues with limiters
  - Existing code, needs to be adapted

$$\frac{\partial \Psi}{\partial t} + \mathbf{u} \cdot \nabla \Psi = 0$$
$$w + a_{\perp} = 1$$

# Acknowledgements

- Mikko Byckling, Sami Ilvonen, Peter Råback, Juha Ruokolainen, Juhani Kataja (CSC)
- Olivier Gagliardini (LGGE), Pierre-Marie Lefevre (U. Oslo), Hakime Seddik (ILTS) -> Mac

- NCoE eSTICC



norden

NordForsk

- Supported porting on large machines in Nordic countries

- NCoE SVALI

