



HAL
open science

Introduction to FreeFem++-cs

Antoine Le Hyaric

► **To cite this version:**

| Antoine Le Hyaric. Introduction to FreeFem++-cs. 2015. hal-01169630

HAL Id: hal-01169630

<https://hal.science/hal-01169630>

Preprint submitted on 1 Jul 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Introduction to FreeFem++-cs

Le Hyaric Antoine^{*†}

June 29, 2015

Abstract

This is a description the FreeFem++-cs software, which implements a graphical interface for the FreeFem++ language dedicated to the finite elements method. We detail how FreeFem++-cs helps users all along the development cycle. Each step in the cycle is described with examples and external links for more information. For advanced users who need to go further than all the functionalities presented here, a list of external software tools is proposed.

1 Introduction

The mathematical bases for the FreeFem++ language [1, 2] are very well documented and available online [3]. But there is more to know to run a simulation successfully. This step-by-step guide proposes one way to minimize the FreeFem++ learning curve through the use of the FreeFem++-cs [4] graphical interface. “cs” stands for “client and server”, where the client is an integrated development environment, and the server runs the FreeFem++ program behind the scenes. This document will mostly focus on features that FreeFem++-cs brings and list references to the documentation [3] for other details.

2 FreeFem++ or FreeFem++-cs?

FreeFem++ is as full-featured standalone application. So one may perfectly choose to use FreeFem++ without a graphical interface. Many of the links proposed here are valid for both. What is the difference between FreeFem++ and FreeFem++-cs? FreeFem++ is run from a terminal, and opens a separate graphical window called “ffgut”. Whereas FreeFem++-cs gathers the editor, the graphics and the output in one single window with the following extra features :

^{*}CNRS, UMR 7598, Laboratoire Jacques-Louis Lions, F-75005, Paris, France

[†]Sorbonne Universités, UPMC Univ Paris 06, UMR 7598, Laboratoire Jacques-Louis Lions, F-75005, Paris, France

- FreeFem++ (fig. 1) contains everything you need to run simulations in a command-line application. **FreeFem++-cs** is a **graphical shell** around it. First-time users usually feel more at ease with a graphical interface. And it is always possible to switch between FreeFem++-cs and FreeFem++ at any time (FreeFem++-cs can be installed alongside FreeFem++ without any interference between the two).
- FreeFem++-cs (fig. 2) includes **inline documentation** (manual, examples, reference card), which is very useful from a new-user point of view.
- Most FreeFem++ users want to concentrate on mathematical and numerical questions, not computing issues. FreeFem++-cs has been developed with the goal of **minimizing the learning curve**, with many commands explicitly listed in standard menus.

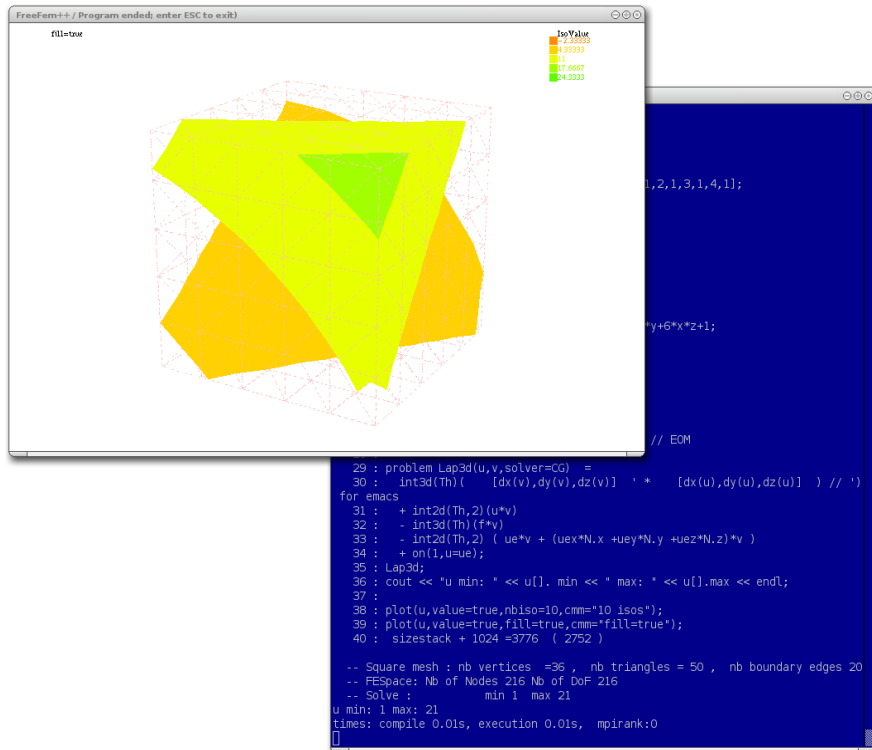


Figure 1: FreeFem++

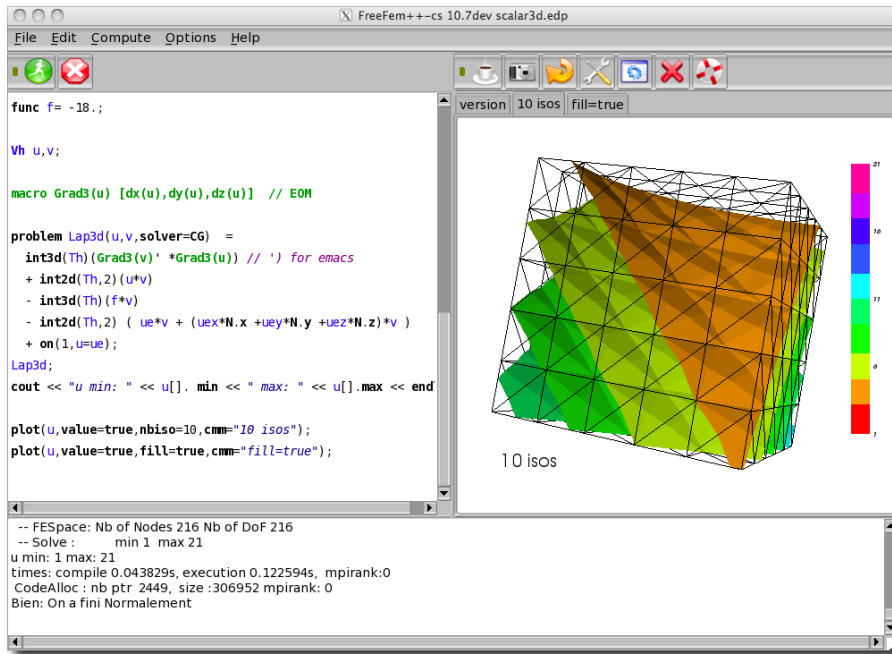


Figure 2: FreeFem++-cs

3 Which machine?

The machine capability constraints are the same as FreeFem++ for obvious reasons (FreeFem++ is the underlying engine). One can use a machine of any size for learning to use FreeFem++ or FreeFem++-cs. On top of this, FreeFem++-cs is now able to run simulations on remote machines.

- **Coarse 2D or 3D meshes** : any machine (even a palmtop) can be used (fig. 3).
- **Fine 2D meshes** : a fast laptop is usable.
- **Fine 3D meshes** : no machine will be too performant, both in terms of RAM and CPU. MPI programming may be required to get a result in realistic time. For examples of bigger test cases check Kirill Pichon Gostaf's work [5].

The operating system can be Windows, Linux or MacOS. FreeFem++ and FreeFem++-cs work fine under all of these.

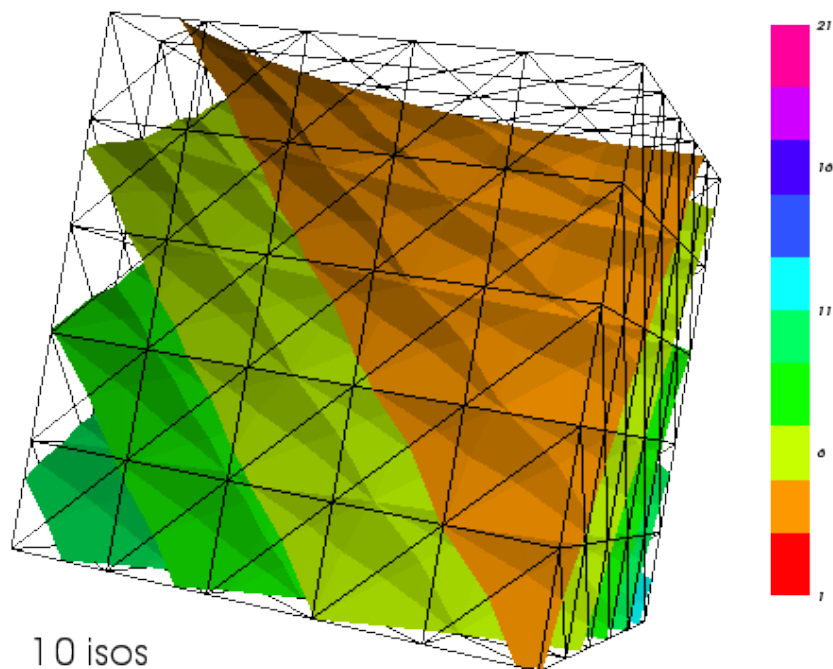


Figure 3: Simulations such as this one (6^3 vertices, P1 finite elements) will run fine on any machine

3.1 Installing FreeFem++-cs

A lot of work has been done to install FreeFem++-cs easily. FreeFem++-cs contains an automatic install procedure. No other software is needed. It can be installed in your own account without administrator privileges. Just follow the installation instructions [4].

3.2 Installing FreeFem++

If you choose not to install FreeFem++-cs, you need to install FreeFem++ from [6]. If you want to have access to the same 3D graphics capabilities as FreeFem++-cs, you may install FreeFem++-vtk (fig. 4). FreeFem++-vtk is the VTK-based [7] graphical part of FreeFem++-cs, cut out as a replacement for FreeFem++'s graphical front-end ("ffglut").

4 Learning the basics

The mathematical basis for the FreeFem++ language is the finite element method [8]. More precisely, to write a working FreeFem++ simulation program, one first needs to convert partial differential equations into their varia-

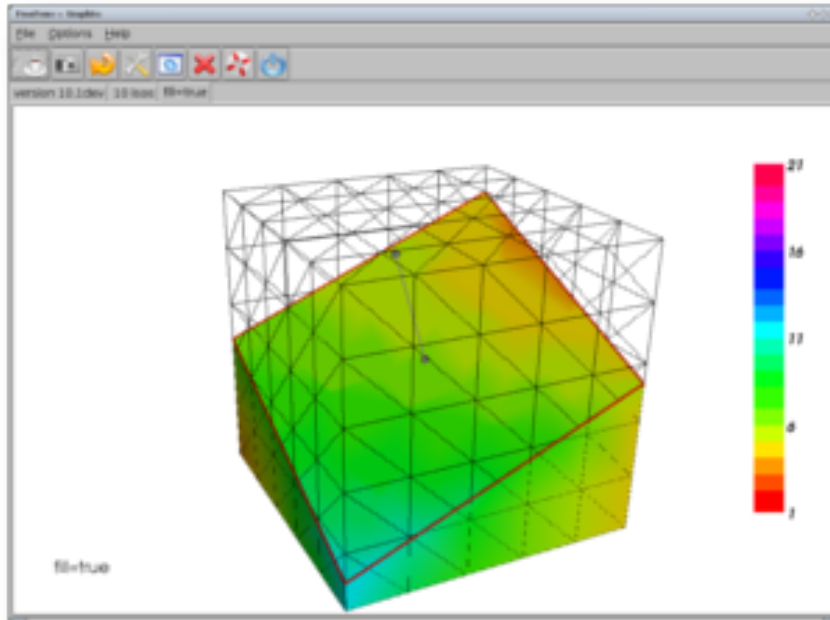


Figure 4: FreeFem++-vtk

tional formulation counterpart. This is explained in chapter “Getting Started” in the FreeFem++ documentation [3]. You may also find text books about finite elements in the FreeFem++ Documentation Bibliography. The FreeFem++ language shares its grammar with C. You will find more links to English and French C programming tutorials in [9]. FreeFem++-cs adds a few commands to navigate through the user interface. They are listed in the Commands Summary of [4].

4.1 Learning from examples

The FreeFem++ documentation [3] contains some very detailed tutorial examples, described from their mathematical definition to the resulting FreeFem++ program. Here is the table of contents of chapter 3, Learning by Examples :

- Membranes
- Heat Exchanger
- Acoustics
- Thermal Conduction

- Irrotational Fan Blade Flow and Thermal effects
- Pure Convection : The Rotating Hill
- A Projection Algorithm for the Navier-Stokes equations
- The System of elasticity
- The System of Stokes for Fluids
- A Large Fluid Problem
- An Example with Complex Numbers
- Optimal Control
- A Flow with Shocks
- Classification of the equations

4.2 The list of all FreeFem++ examples

Starting from an existing example makes it easier to try something easily and avoid the blank page syndrom. All FreeFem++ examples (including shared libraries and 3D) are available through :

- the list of all FreeFem++ examples on the web site :
 - examples++-tutorial
 - examples++
 - examples++-3d
 - examples++-chapt3
 - examples++-eigen
 - examples++-load
 - examples++-mpi
 - examples++-other
 - examples++-bug
- the "File/Open Example..." menu option in FreeFem++-cs (fig. 5).

Please note that since FreeFem++ is the underlying computational engine of FreeFem++-cs, all examples give exactly the same results in FreeFem++ and FreeFem++-cs (bugs included!).

5 Writing a program

After reading a few examples, one is ready to try to write a test program. This is where the integrated interface of FreeFem++-cs is really useful, since it contains all three steps of the Development Cycle (edit,run,visualize) in one window (fig. 6).

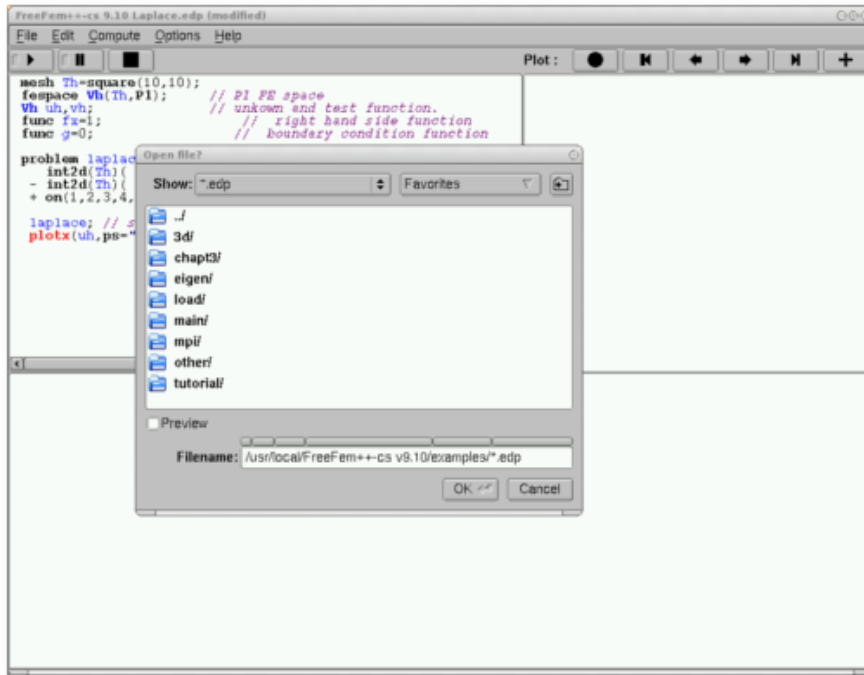


Figure 5: File/Open Example...

5.1 The mesh

5.1.1 Describing the mesh programmatically

One can describe the mesh as part of the FreeFem++ script or in an external CAD tool. Describing the mesh inside the FreeFem++ program is very handy, since changing the shape of the mesh or the size of elements is achieved just by changing a parameter value. It can be done in 2D and 3D, but is more easily achievable in 2D. In the FreeFem++ documentation [3], everything is described in chapter Mesh Generation.

5.1.2 External CAD tools

Using a CAD tool becomes necessary when the mesh is in 3D or contains many intersections between geometrical objects. In 3D, meshes can still be defined and generated in the FreeFem++ program, but taking care of all the details (especially when intersections of surfaces are involved) becomes difficult. So it is advised to use an external CAD tool like Gmsh [10] :

- The geometry is drawn and the mesh is built in Gmsh. Then the mesh is exported in the Medit [11] format.

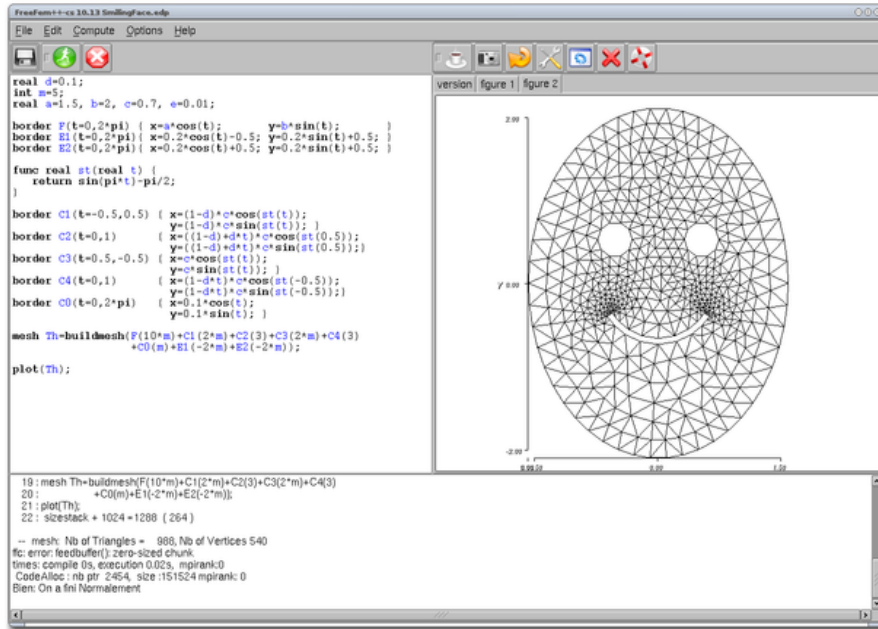


Figure 6: Development cycle

- Loading the mesh in Medit and displaying the material properties of hand-picked points on the boundaries allows to isolate the numerical labels of the boundaries where FreeFem++ will apply a boundary condition.
- The FreeFem++ program loads the mesh with `readmesh3()`.

For learning purposes it may be sufficient to build meshes out of basic geometrical shapes like cubes and spheres, like the ones found in the list of FreeFem++ 3D examples in [4].

5.2 Editing and running

One key point of FreeFem++-cs is that it contains a color-coded editor with an interesting feature: undefined variables are colored in red (fig. 7). Color codes are :

- **bold black** : all standard FreeFem++ keywords.
- **blue** : user-defined objects. User-defined types are in **bold face**. Character strings are in *"dark blue"*.
- **green** : macros and include files.

- **bold red** : spelling mistakes, syntax errors, undefined objects, macros and include files containing errors. For instance, there are two mistakes in figure 7 : "plotx" does not exist (it should be "plot") and "f" is not defined because it was mistyped as "fx".
- **purple** : comments.
- Once the program is complete, it can be run by clicking on the "run" button (no need to save it) and if FreeFem++ produces a error the line containing the error will be highlighted.

```

FreeFem++-cs 9.10 Laplace.edp (modified)
File Edit Compute Options Help
mesh Th=Square(10,10);
fespace Vh(Th,P1); // P1 FE space
Vh uh,vh; // unkown and test function.
func fx=1; // right hand side function
func g=0; // boundary condition function

problem laplace(uh,vh,solver=GMRES,tgv=1e5) =
- int2d(Th)( dx(uh)*dx(vh) + dy(uh)*dy(vh) ) // bilinear form // definition of the problem
- int2d(Th)( f*vh ) // linear form
+ on(1,2,3,4,uh=g) ; // boundary condition form

laplace; // solve the problem plot(uh); // to see the result
plotx(uh,ps="Zsplace.eps",value=true);

```

Figure 7: FreeFem++-cs editor

5.3 Visualizing

Visualizing 3D data in FreeFem++-cs is done thanks to the VTK [7] library. The plot() command applied to a 3D variables produces a figure like fig. 8. All VTK visualization techniques could be exploited but it is not realistic to to implement all of them like in ParaView [12]. The list of available graphical options is on figure 9. The most intuitive visualization options have been linked to plot() parameters. The current plan is to develop more options (but without reaching the level of completeness of ParaView). All of these options will eventually be modifiable from the FreeFem++ program through plot() options (fig. 10).

6 Going further

6.1 Looking for information

The official FreeFem++ documentation can be accessed in many different ways :

- as a printed book [1].
- as a PDF in [6].

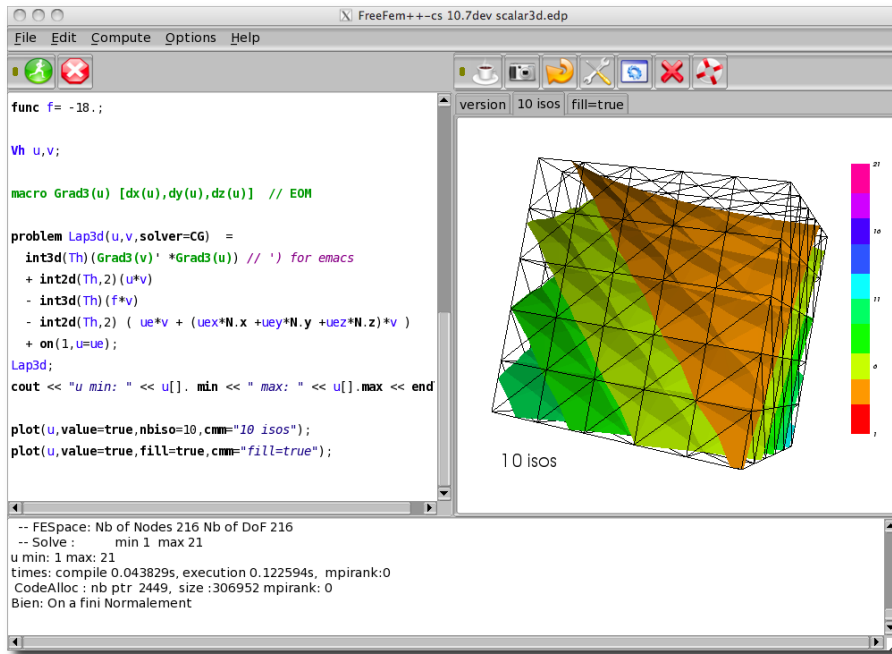


Figure 8: Visualizing 3D data

- in HTML from the web [3].
- through other documents in the Manuals section of the FreeFem++-cs website [4].

There is also a dedicated mailing list [13] to ask questions to the community.

6.2 More powerful tools

6.2.1 3D graphical tools

The goal of FreeFem++-cs is not to replace all the very powerful graphical tools that exist on the net. Here are some 3D graphical tools that are easily connected to FreeFem++ and FreeFem++-cs. These tools may become useful when developing more powerful simulations :

- **Medit** : link with Medit in the FreeFem++ documentation describes how to call Medit from FreeFem++. The full Medit documentation is accessible online [11].
- **VTK output** : very useful to connect to e.g. ParaView [12]. The trick is to save FreeFem++ data into the VTK format, via the FreeFem++ `iovtk.cpp` load module example or "File/Save Image As.../VTK" in the FreeFem++-cs window.

- **Gmsh** : Gmsh [10] has already been quoted as a CAD tool to generate a 3D mesh. Gmsh also contains a useful visualisation module which may be called from FreeFem++, as described in <http://www.geuz.org/pipermail/gmsh/2010/005844.html>.

6.2.2 Editors

The FreeFem++-cs editor has one big advantage for new users : it highlights mistyped variables in red. Apart from that, it will never have as many features as a complete development editor like Emacs. So FreeFem++-cs is able to deal with a program text edited in an different editor. A file which is changed externally is automatically reloaded into FreeFem++-cs, and it can also be run automatically :

- "File/Auto Reload" to reload a file into FreeFem++-cs automatically when it has changed.
- "Compute/Auto Run" to run program again every time a change is detected.
- "Edit/Hide Editor Panel" to put the internal editor window out of sight and leave more space for the graphics window.

6.2.3 Load modules

FreeFem++ "load modules" are dynamic libraries written in C++. Because of the universal nature of C++, load modules can be used to implement any functionality missing from FreeFem++. The downside is that they can be complex to implement. Compiling FreeFem++ load modules requires "ff-c++", which is not included in the FreeFem++-cs package at the moment because the "ff-c++" command has strict requirements on where the different headers and object files should be. To develop new modules, it is therefore required to switch to FreeFem++. Once a module is built, it may be used in FreeFem++-cs.

6.2.4 Parallel programming

Multicores are supported automatically through the use of numerical libraries capable of "Shared Memory Parallelism" (e.g. multithreaded Atlas). "Distributed Memory Parallelism" (several distinct machines working together) is also possible but it requires MPI programming. Load modules and parallel programming are not simple in any language, so FreeFem++-cs is not tuned to do this for the moment. As of this writing, the default precompiled version of FreeFem++-cs does not include a multithreaded BLAS version, and does not run MPI programs either. To run parallel programs, FreeFem++ must be downloaded and compiled with the correct configuration options. This is described in MPI programming in the FreeFem++ documentation [3].

6.3 Conclusion

Through the years, the FreeFem++-cs feature which appealed to FreeFem++ users was its single window and simplicity. FreeFem++-cs will keep following FreeFem++ developments, while making sure that users can focus on mathematics and not on computing issues. Now that 3D is implemented, we will focus on quality and stability in future developments. In continuing with this approach, we are implementing the remote execution of programs onto an external server, where FreeFem++ is run on a powerful remote server, and FreeFem++-cs displays the graphics on a local workstation. We also plan to develop a Javascript version which works straight from the internet with no need for software installation on the client machine [14].

References

- [1] F Hecht, O Pironneau, A Le Hyaric, and K Ohtsuka. *FreeFem++ Manual, Version 2.21.*, 2006.
- [2] F. Hecht. *New development in FreeFem++*. *J. Numer. Math.*, 20(3-4):251–265, 2012.
- [3] *FreeFem++ Wiki Manual*. <http://www.um.es/freefemv3/ff++/pmwiki.php>. [accessed 15-June-2015].
- [4] Antoine Le Hyaric. *FreeFem++-cs, an integrated environment for FreeFem++*. <http://www.ann.jussieu.fr/~lehyaric/ffcs>. [accessed 15-June-2015].
- [5] Kirill Pichon GOSTAF. *MFD2010 congress page*. <http://www.ljll.math.upmc.fr/~gostaf/mfd2010>. [accessed 15-June-2015].
- [6] *FreeFem++ website*. <http://www.freefem.org/ff++>. [accessed 13-June-2015].
- [7] *VTK - The Visualization Toolkit*. <http://www.vtk.org>. [accessed 15-June-2015].
- [8] Wikipedia. *Finite Element Method*. http://en.wikipedia.org/wiki/Finite_element_method. [accessed 13-June-2015].
- [9] *Logiciels gratuits pour le Calcul Scientifique*. <https://www.ljll.math.upmc.fr/lehyaric/freesoft>. [accessed 13-June-2015].
- [10] C. Geuzaine and J.-F. Remacle. *Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities*. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.

- [11] Pascal Frey. *Medit 3.0, OpenGL-based scientific visualization software*. <http://www.ann.jussieu.fr/frey/software.html>. [accessed 15-June-2015].
- [12] Ahrens, James, Geveci, Berk, Law, Charles. *ParaView: An End-User Tool for Large Data Visualization*. *Visualization Handbook*, Elsevier, 2005.
- [13] *FreeFem++ Mailing List*. <http://www.ljll.math.upmc.fr/cgi-bin/mailman/listinfo/freefempp>. [accessed 15-June-2015].
- [14] Antoine Le Hyaric. *FreeFem++ and Javascript*. <http://www.ann.jussieu.fr/~lehyaric/ffjs>. [accessed 15-June-2015].

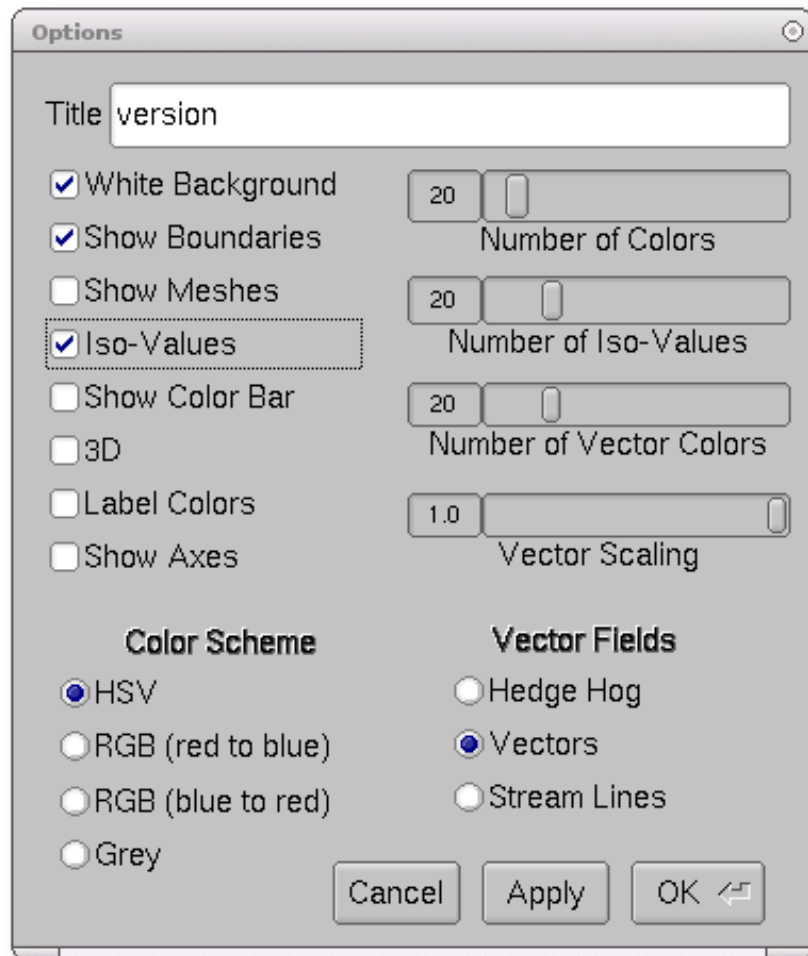


Figure 9: VTK options panel

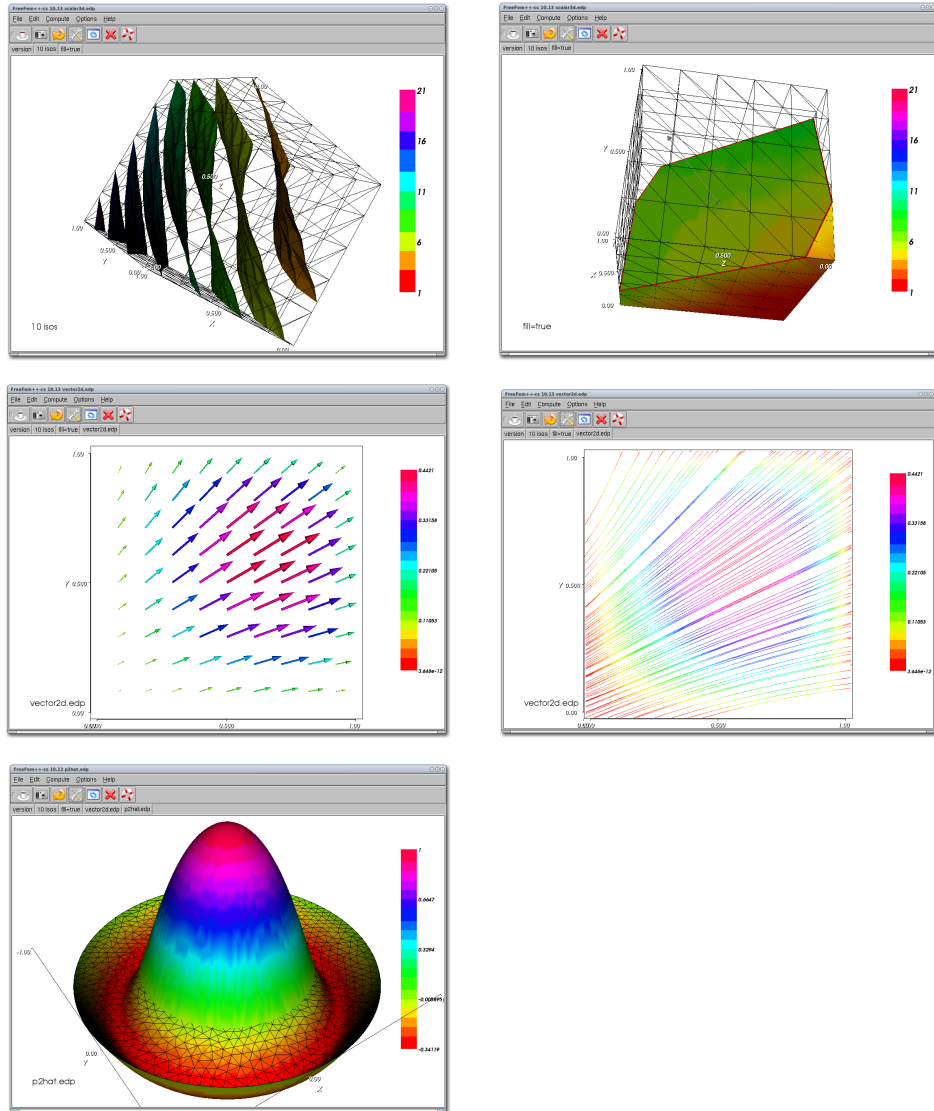


Figure 10: Examples of VTK rendering : 3D isovalues, cut through a 3D scalar, vector field, stream lines, 3D surface