

ON THE OSS LIBRARIES BEHIND ELMERGUI

Mikko Lyly

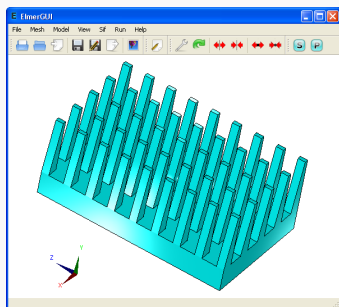
CSC - IT Center for Science
P.O. Box 405
FI-02101 Espoo, Finland

May 25, 2009



INTRODUCTION

ElmerGUI is the new graphical user interface for Elmer. It replaces the old interface, ElmerFront, which has slowly become deprecated. ElmerGUI is capable of importing finite element mesh files, generating finite element partitionings for CAD models, setting up PDE-systems to solve, and exporting model data and results for the postprocessor to visualize. Intended for tutorials, education, and for scetching.



WORK FLOW

ElmerGUI it self has no CAD-modeling features (only import). The work flow of a typical analysis is more or less the following:

- ▶ Modeling the geometry with external tools:
 - ▶ GiD - powerful reasonably priced CAD program and mesh generator
 - ▶ Gmsh - open source tool for CAD modeling and mesh generation
 - ▶ Salome - GUI for OpenCASCADE, several built in mesh generators
- ▶ Geometry or mesh import followed by (re)meshing
- ▶ Setting up the PDE (equations, params, boundary conditions, ...)
- ▶ Generating and saving input files for ElmerSolver
- ▶ Launching ElmerSolver as an external process
- ▶ Postprocessing the results (ElmerPost or VTK)

The design and programming of ElmerGUI was initiated in the early 2008. The basic requirements were the following:

- ▶ Portability. The GUI should compile and run on all major desktop environments without drastic platform dependent modifications.
- ▶ The GUI should have a modern polished look, and it should be easy to use and extend.
- ▶ It should be possible to utilize external libraries to a large extent.
- ▶ Productivity. Efficient high level classes should be readily available.
- ▶ The GUI should be fast (involves complex algorithms and visualization of large data sets).
- ▶ The licences of all external components should be compatible (GPL).

Based on the above considerations, it was decided to write the application based on the Qt library from Trolltech (part of the Nokia corporation since 2008):

- ▶ One code - all platforms.
- ▶ Well documented high quality toolkits, modules, and helper classes.
- ▶ Productivity.
- ▶ Native OpenGL support (see also VTK below).
- ▶ Written in C++ (like most of the external libs).
- ▶ Powerful graphical tools for UI design.
- ▶ Incorporates with all major IDEs (Visual Studio, Eclipse, Code Blocks,...)
- ▶ Three possible licences: commercial, LGPL or GPL (have to choose before starting to develop your application - trap!).

For more details, see <http://www.qtsoftware.com/>

CAD AND MESH IMPORT LIBRARIES

The basic task of ElmerGUI is to import CAD models and mesh files generated by external modeling tools. Not much choice here:

- ▶ ElmerGrid is the native import library and mesh manipulation tool of the Elmer software suite. ElmerGUI links statically against ElmerGrid, which provides the basic functionality for importing FE mesh files. Written in C.
- ▶ The CAD import functionality is based on the libraries of OpenCASCADE. OpenCASCADE is a HUGE package written in the early 90s by Matra Datavision (fused with Dassault in 1998), and since then developed by OpenCASCADE SAS (owned by Principia since 2003). LGPL-type licence. Written in C++.
- ▶ There is also the ACIS kernel developed by the Spatial Corporation. Extremely powerful, but incompatible licence.



MESH GENERATORS

Another fundamental task for ElmerGUI is related to mesh generation. Eventhough there are numerous packages and libraries available all over the net, only a few of them are capable of dealing with complicated 3D geometries and produce high quality Delaunay partitionings for the FEM:

- ▶ Tetgen. High quality 3D generator provided by the Weierstrass Institute for Applied Analysis and Stochastics. Very fast, but comes with incompatible licence. Possible to compile as a plugin for ElmerGUI. Written in C++.
- ▶ Netgen. High quality 2D and 3D generator by Joachim Schöberl (currently developed in Linz, Austria, and in Aachen, Germany). Compatible LGPL licence. Default generator in ElmerGUI. Written in C++.

POSTPROCESSING

There are several external tools available for postprocessing the numerical results produced by ElmerSolver. The main tool is ElmerPost, which is an external program written in C and TCL. Another possibility is to export the results from ElmerSolver in DX-, GiD-, or VTK-formats, and use programs like ParaView to draw surfaces, contours, etc.

The internal postprocessor of ElmerGUI is based on the Visualization Toolkit (VTK) by Kitware Inc. The library of VTK is extremely powerful, licenced under GPL, and relatively easy to integrate with GUIs written in C++. ParaView is the canonical GUI for VTK.

VTK provides algorithms e.g. for drawing partially transparent surfaces, volume rendering, particle tracing, glyphs, and so on. It is very well suited for FEM with unstructured meshes. Works also in parallel.

SUMMARY

ElmerGUI is a graphical OSS interface for the Elmer-software suite. It uses several external OSS libraries for its core functions:

- ▶ Qt for the GUI, XML-parsing, databases, threading, ECMA script, ...
- ▶ OpenCASCADE for CAD import.
- ▶ ElmerGrid, Netgen, and Tetgen, for meshing.
- ▶ QWT for drawing functions and curves during solution.
- ▶ VTK for postprocessing.
- ▶ PythonQt for scripting.

THOUGHTS

Writing all the above from scratch in a small project is practically impossible:

- ▶ OSS provides enabling technologies for programming efficient GUIs (pre- and postprocessors) for the FEM.
- ▶ OSS makes it possible to save both money and time.
- ▶ Possible to (try to) understand and control what happens under the hood.

On the other hand:

- ▶ There is not much to lean on - if something goes wrong, or simply does not work as expected, the worst case scenario is that you're on your own.
- ▶ Not much choice, really (3 libraries for the GUI, 1 for CAD, 2 for mesh generation).

EOF. Thank you for your attention.