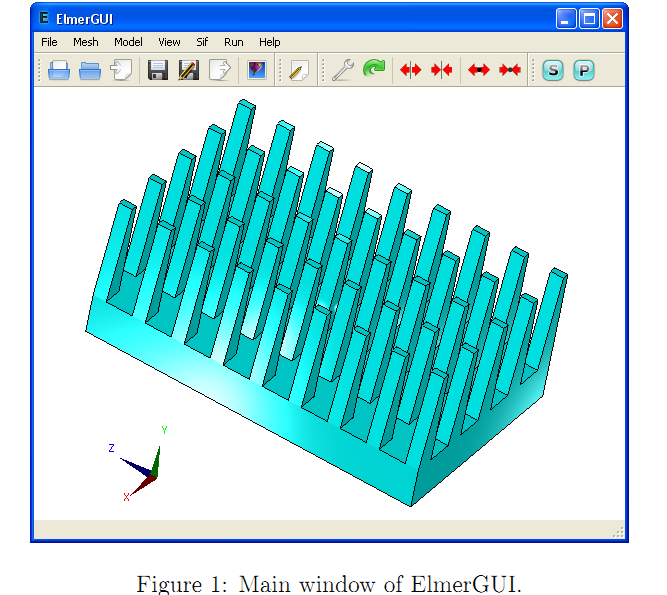
# ElmerGUI manual v. 0.4

## 1 介绍

ElmerGUI是Elmer软件套件的一个图形用户界面。这个程序能输入不同格式的有限单元网格文件，为不同的几何体输入文件生成有限单元集群，建立要求解的偏微分方程系统，输出模型数据和结果以供ElmerSolver和ElmerPost进行可视化显示。ElmerGUI也有一个内置的后处理器，可代替ElmerPost用来绘制彩色表面、等值线、矢量场和可视化时间相关的数据。



ElmerGUI主要特征之一就是并行求解器ElmerSolver\_mpi的接口。图形用户界面向用户隐藏了很多操作，这些操作一般由域分解、启动并行流程、合并结果相关的各种外部工具的命令行指示。这使得多核处理器下使用ElmerSolver得以实现，甚至能在交互式桌面环境中使用。

ElmerGUI的菜单是可编程的，个性化菜单相对简单。附录A中有一个个性化菜单的例子。

ElmerGUI依赖QtSoftware的Qt4跨平台框架，使用Qwt5库去描绘科学数据。ElmerGUI的内置后处理器基于可视化工具包（VTK）。CAD导入特性用OpenCASCADE加以实现。这个程序也可以用Tetgen和Netgen作为有限单元网格生成器。

## 2 安装

ElmerGUI的源代码能够在GitHub上的Git资源库中找到，被GPL许可的源代码可以通过执行如下命令下载：

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

或者

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

这会检索整个Elmer套件的最新版本。

### 2.1 Linux

在开始编译之前，请确保你的系统上装了Qt4开发包（库、头文件、程序开发工具等）。建议安装Qt版本4.3及以上。你也可以安装Qwt5、VTK5（编译支持Qt4）和OpenCASCADE 6.3用于附加功能。

Cmake能用来生成编译所需的makefiles。ElmerGUI编译激活首先要设置相应的Cmake变量，如-DWITH\_ELMERGUI:BOOL=TRUE。其他影响程序编译、也能类似地用-D进行设置的逻辑变量有WITH\_QWT, WITH\_VTK, WITH\_OCC, WITH\_PYTHONQT, WITH\_MATC和WITH\_PARAVIEW。

一旦build process结束，它能建立环境变量ELMERGUI\_HOME，把它加入PATH：

$ export ELMERGUI\_HOME=/usr/local/bin

$ export PATH=$PATH:$ELMERGUI\_HOME

用如下命令启动程序：

$ ElmerGUI

## 3 输入文件

### 3.1 几何模型输入文件和网格生成

ElmerGUI能够导入有限单元网格文件，在有分段线性边界的有界域生成二维或三维有限单元划分。可以使用如下网格生成器的一种：

ElmerGrid (built-in)

Tetgen (optional)

Netgen (built-in)

默认的输入口和网格生成器是ElmerGrid。Tetgen是一个可选的模块，能不能使用取决于安装（安装和编译说明能在能从Elmer的源代码树trunk/misc中找到）。

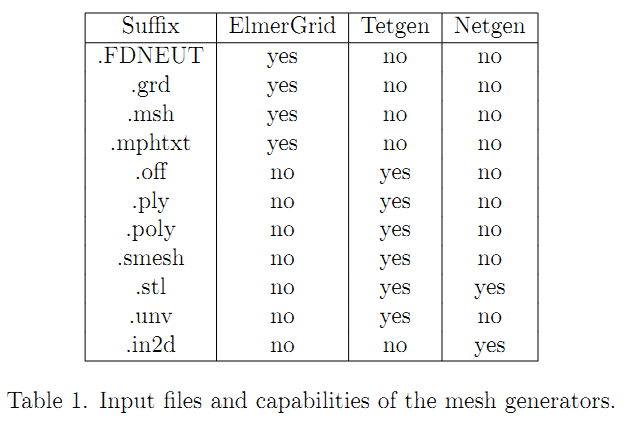
当一个几何结构输入文件打开时，ElmerGUI会自动选择一个输入过滤器或者一个网格生成器：

File🡪 Open…

这个选择根据表1中所示的输入文件后缀进行。如果两个或更多生成器都能处理同一格式，那么会使用用户定义的“偏爱生成器（preferred generator）”。偏爱生成器如下定义：

Mesh 🡪 Configure…

一旦输入文件已经打开，可能会为了更好的精确度或计算效率修改网格参数、重新对几何体进行网格划分。网格划分参数能从如下目录Mesh 🡪Configure…找到。关于Tetgen的控制字符串已经在Tetgen的用户指南中讨论、解释过。



网格生成器从菜单Mesh中再次激活，选择：

Mesh 🡪Remesh

出现问题时，可以执行如下来终止网格划分线程：

Mesh 🡪Terminate meshing

### 3.2 Elmer网格文件

Elmer网格文件包含以下四个（文件格式详细描述见附录B）：

mesh.header

mesh.nodes

mesh.elements

mesh.boundary

Elmer网格文件能被加载和保存，只需从文件目录File中打开网格目录：

File 🡪Load mesh...

和

File🡪Save as...

### 3.3 Project文件

一个ElmerGUI project由一个project目录组成，包含Elmer网格文件和一个描述当前状态和设置的.xml格式文件egproject.xml。

Projects可以从文件目录File加载或者保存，只需如下选择：

File 🡪 Load project...

和

File 🡪Save project...

当一个ElmerGUI project被加载时，一个新的求解器输入文件将会通过如下定义中的sif字生成并保存进project目录下：

Model🡪Setup…

如果新的文件名与已经存在的求解器输入文件名相同，已经存在的文件就会被重写覆盖。

典型project目录包含如下：

case.sif

egproject.xml

ELMERSOLVER\_STARTINFO

mesh.boundary

mesh.elements

mesh.header

mesh.nodes

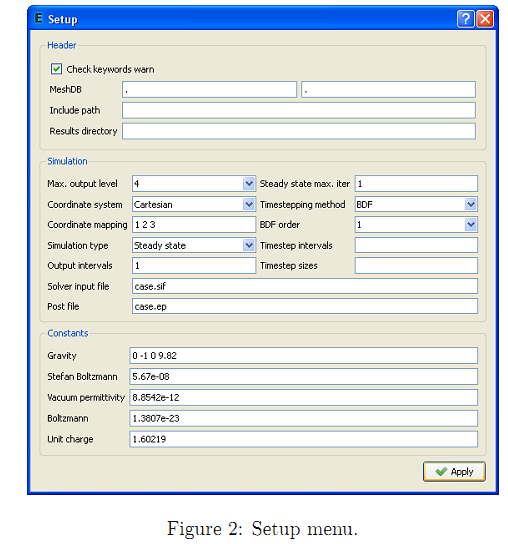
## 4 模型定义

### 4.1 Setup菜单

通用的setup菜单在如下目录下可以找到：

Model 🡪Setup...

这个菜单定义求解器输入文件的“Header”、“Simulation”和“Constants”模块的基本变量。这些模块的内容已经在Elmer求解器手册（SolverManual of Elmer）中有详细讨论。



### 4.2 方程菜单（Equation menu）

第一个由ElmerGUI定义文件（见附录A）构造的“动态菜单”是：

Model 🡪Equation

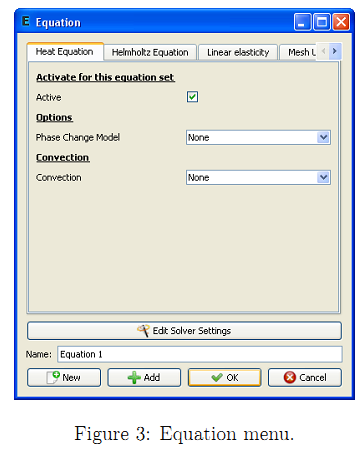
这个菜单定义了将要求解的偏微分方程系统（PDE-system）和求解中使用的数值方法和参数。它将会被使用来在求解器输入文件中生成“Solver”模块。

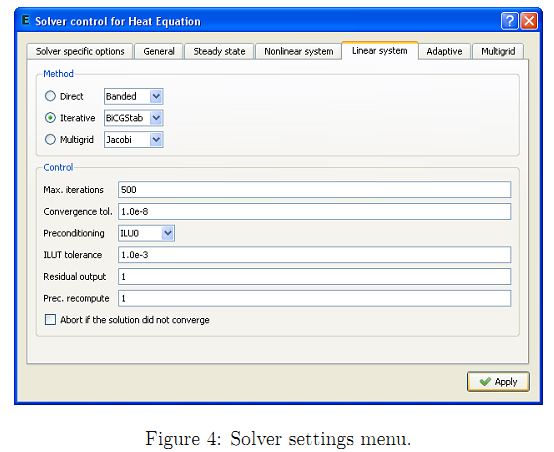
一个PDE-system按如下选择定义：

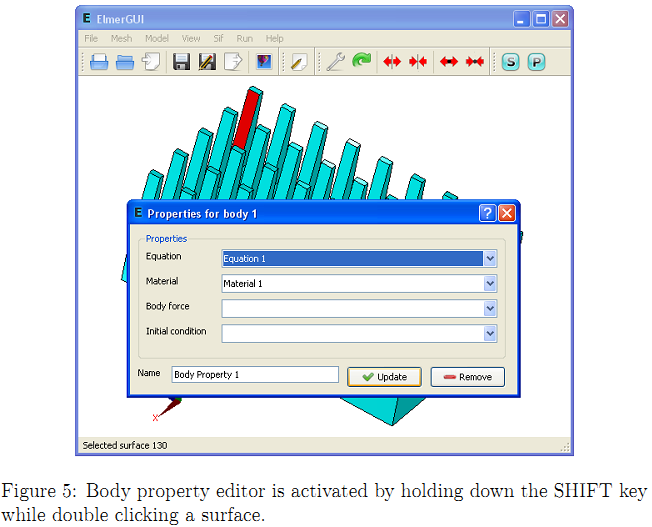
Model 🡪Equation 🡪Add...

一旦PDE-system通过激活个体方程加以定义了，数值方法和参量都能通过按“Edit Solver Settings”求解器设置编辑按钮来加以选择和调整。PDE-system的名字在标为“Name”的行编辑框中定义。在按完Ok按钮之后，方程在Model菜单下仍然是可视化、可编辑的。

也可以通过按住SHIFT键并双击几何体表面中的一个，以此赋予一个体相应方程。会出现一个上托目录单，列表了所有能被选择的属性。







### 4.3 Material menu

这个菜单跟材料和模型参数相关：

Model 🡪Material

这个菜单将会被用来在求解器输入文件中生成材料模块。

为了定义一个材料参数集并赋给体，按如下选择：

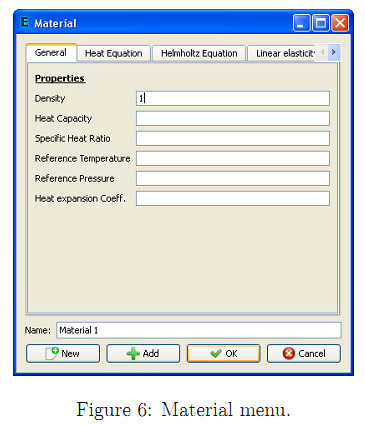
Model🡪Material🡪 Add...

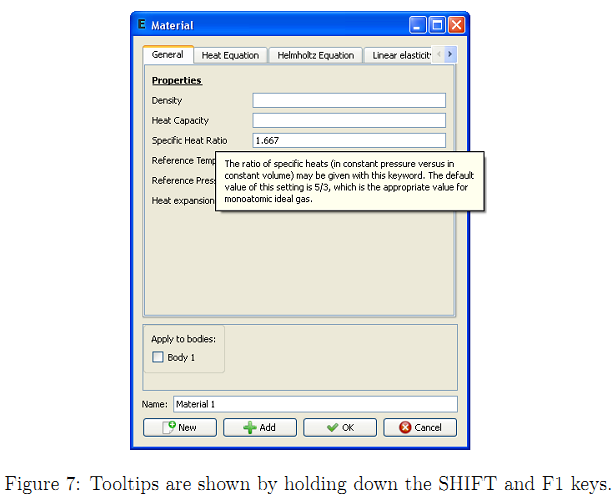
相似地，可以通过按住SHIFT键并双击几何体的边界，以此赋予一个体相应材料。

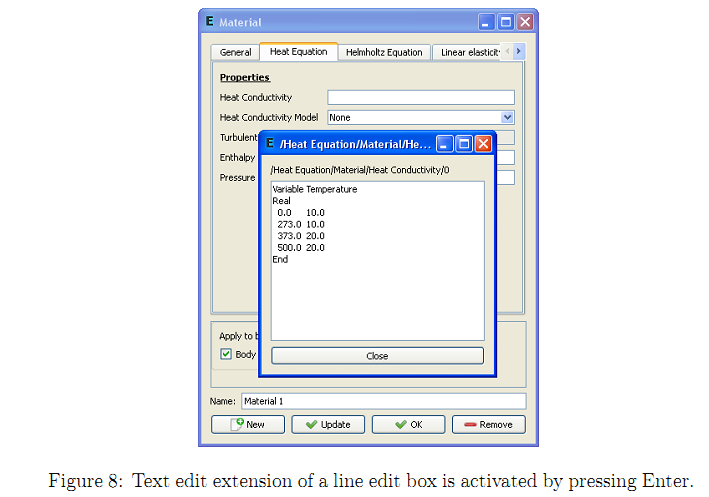
注意：密度的值务必在“General”标签下进行定义。这一块不要被落下没被定义。

注意：如果你聚焦在动态菜单的行编辑框中，按Enter，一个小的文本编辑对话框会浮现。这允许相较常数而言更复杂的表达输入。比如，进入Model🡪Material，选择Add…，把光标停留在“Heat conductivity”行编辑“Heat equation”框，按Enter。你就能定义热传导系数成一个温度相关的分段线性函数。图4.3展示了一个例子。在这个案例中，热传导系数在温度273度以下时为10，在温度273和373度之间为升至20，在373度以上时为常数20。

如果用户按SHIFT和F1，活跃小插件的一个工具提示将会被展示。







### 4.4 Body force menu

轮到Body force menu菜单了：

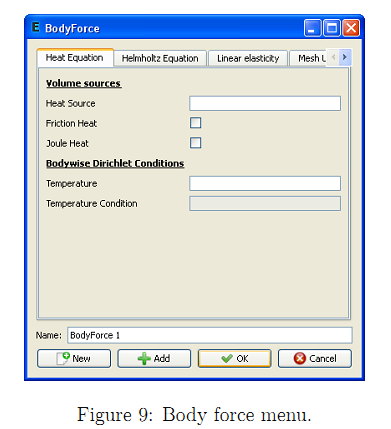
Model 🡪Body force

这个菜单被用来在求解器输入文件中构造“Body force”模块。

再一次，选择：

Model 🡪 Body force 🡪Add...

来定义一组体力并赋给体。



### 4.5 初始状态菜单（Initial condition menu）

最后跟体属性相关的菜单是：

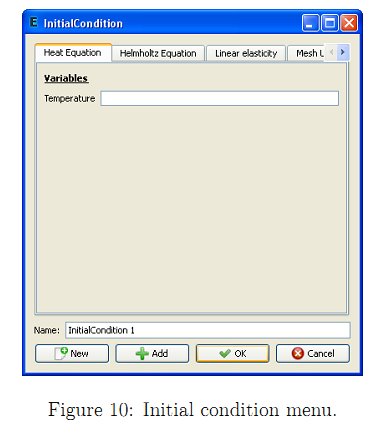
Model 🡪 Initial condition

再一次，选择：

Model 🡪Initial condition 🡪 Add...

来定义一组初始条件并赋给体。

这个菜单是用来构造求解器输入文件的“Initial condition”模块。



### 4.6 Boundary condition menu

最后，有一个为建立边界条件的菜单条目：

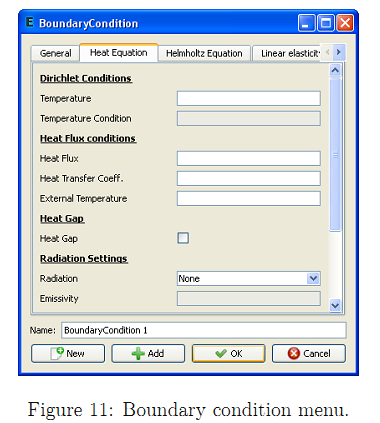
Model🡪 Boundary condition

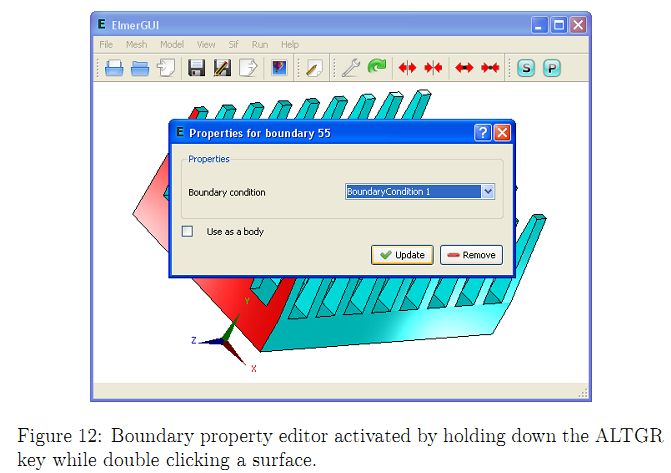
选择

Model 🡪Boundary condition 🡪Add...

来定义一组边界条件并赋给边界。

可以通过按住ALT或者ALTGR键同时双击一个表面或者边缘来赋给一个边界相应的边界条件。一个上托项目单会出现，列举所有可供选择的边界条件。从组合框中选择一个条件，最后按OK。





## 5通用功能（Utility functions）

### 5.1 边界划分和整合

表1中列举的有些输入文件格式并不能很好地适合有限单元计算（FE-calculations），尽管被广泛使用。针对像.stl一样的格式，其缺点为不能区分携带不同属性的不同边界部分，此外这种格式用离散的三角形近似边界，不能满足通常的有限单元相容性条件。

ElmerGUI提供一套轻量化的边界划分和整合的工具。划分基于“锐边检测”。如果两条法向量之间的夹角超过某一值（默认20度），两个边界单元之间的一条边缘就会被认为是锐利的。The sharp edges are then used as a mortar to divide the surface into parts.用户可以执行一次锐边检测和边界划分，从Mesh菜单中选择：

Mesh 🡪Divide surface...

在2D中相应的操作是：

Mesh 🡪Divide edge...

结果部分从第一个自由索引开始枚举。

有时，以上的进程产生了太多不同部分，最终需要被统一。这能通过按住CTRL键同时双击选择一组表面，并选择：

Mesh 🡪 Unify surface...

在2D中同一操作为：

Mesh 🡪Unify edge...

结果会继承选择的组中最小索引。

不属于闭环的锐边可以被移除：

Mesh 🡪Clean up

这个操作对边界划分没有任何影响，但是有时会使结果看起来更好。

### 5.2 保存图片

显示区域绘制的模型可以扫描成24位RGB图像并以好几种图片文件格式保存：

File 🡪Save picture as...

函数支持.bmp、 .jpg、 .png、 .pbm、.pgm和 .ppm文件扩展。

### 5.3 视图菜单（View menu）

视图菜单为控制ElmerGUI的视觉行为提供好几种公用功能。功能名都很能解释自己的功能。

## 6 求解器输入文件

Model菜单的内容以求解器输入文件的形式传递给了求解器。求解器输入文件如下生成：

Sif🡪Generate

文件内容都是可编辑的：

Sif 🡪Edit...

新的sif文件激活之前需要保存。建议的方法是：

File 🡪Save project...

通过这种方式当前的网格划分和project文件保存在同一目录下，避免后面可能发生的不一致。

## 7 求解和后处理

### 7.1 运行求解器

一旦求解器输入文件生成，project被保存，就可以开始求解这个问题了：

Run 🡪Start solver

这会触发ElmerSolver单线程（标量解）或者ElmerSolver\_mpi多重的MPI进程（并行解），这取决于下面的定义：

Run 🡪 Parallel settings...

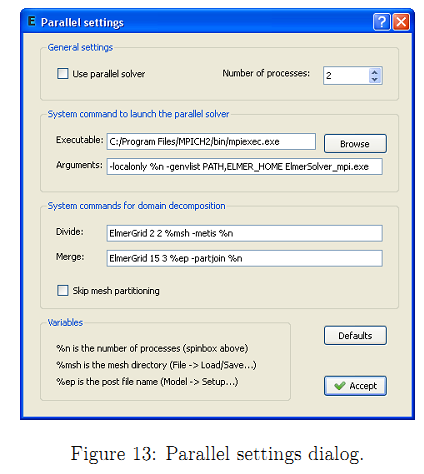
并行菜单有三个分组框。通常用户只会接触其中的一组“General settings”，选择要执行的进程数目。其他两组处理启动MPI进程的系统命令和域分解的外部工具。如果ElmerSolver\_mpi不在start-up里并行菜单会是灰色的。

当求解器运行时，伴随着迭代过程会有一个日志窗口和一个收敛监测器。以防发散和其他问题，求解器可以选择如下终止：

Run 🡪Kill solver

求解器最后会为ElmerPost在project目录下写一个结果文件。这个ep文件名字在如下定义：

Model 🡪 Setup...



### 7.2 后处理1（ElmerPost）

ElmerGUI为绘制、显示和操作结果提供了两种不同的后处理器。

第一种可以按如下激活：

Run 🡪Start postprocessor

这能启动ElmerPost，读入结果文件，展示一个代表结果的等值线图。如果结果是并行求解器产生的，计算中使用的域分解将会被展示。

### 7.3 后处理2（VTK）

第二种后处理程序基于可视化工具包，VTK。它从如下激活：

Run 🡪Postprocessor (VTK)...

一个新的窗口会出现，提供绘制表面、等值线、向量和流线的方法。

#### 7.3.1 Python interface接口

如果ElmerGUI用支持PyhthonQt的工具编译的，就会有一个可供脚本撰写的Python控制台。这个控制台位于Edit菜单下：

Edit 🡪PythonQt console...

控制台提供如下类的接口：

egp // ElmerGUI post processor

matc // MATC language interpreter

preferences // controls for preferences

surfaces // controls for surface plots

vectors // controls for vector fields

isoContours // controls for isocontours

isoSurfaces // controls for isosurfaces

streamLines // controls for streamlines

colorBar // cotrols for the colorbar

timeStep // controls for transient results

text // text annotation

上述每一个类都提供一些对于数据和显示操纵有用的方法。比如，假设我们想要读入结果文件“case.ep”，作为一个半透明曲面图显示温度场。指令如下（更多例子可见7.3.3部分）：

py> egp.ReadPostFile("case.ep")

py> egp.SetSurfaces(True)

py> surfaces.SetFieldName("Temperature")

py> surfaces.SetOpacity(50)

py> egp.Redraw()

#### 7.3.2 Public methods

#### 7.3.3 Example scripts

#### 7.3.4 ECMAScript interface

## 附录

### \* A.ElmerGUI initialization file

ElmerGUI初始化文件位于ELMERGUI\_HOME/edf，名为egini.xml。

你可以编辑这个文件来改变ElmerGUI默认的行为。比如，改变<splshscreen>的值从1到0，可以关掉启动时的启动画面。在一些非常复杂的模型中你也可以增加求解器、方程等默认值的个数。

### \* B.ElmerGUI material database

文件 ELMERGUI\_HOME/edf/egmaterials.xml 定义了ElmerGUI的材料数据库。这个文件的格式如下：

### \* C.ElmerGUI definition files

目录ELMERGUI\_HOME包含了一个名为edf 的子目录。这是所有的ElmerGUI定义文件（ed-files）所在地方。定义文件都是XML格式的文本文件，定义了Model菜单的内容和外观。

当ElmerGUI启动，这些ed-files就会被反复从edf目录下加载。后来，可以通过以下选择查看和编辑他们的内容：

File 🡪Definitions...

一个ed-file有如下结构：

### \* D.Elmer mesh files

mesh.header、mesh.nodes、mesh.elements、mesh.boundary

### \* E.Adding menu entries to ElmerGUI

因为ElmerGUI基于Qt4，它在个性化菜单和对话窗口方面相对简单。例如，一个新的菜单项可以按如下添加：

### \* F.ElmerGUI mesh structure