`pgfplots` output for `gmsh`

Sebastian Eiser

April 7, 2014

Abstract

This document shows the possibilities with the `pgfplots` patch for `gmsh`. Text such as labels, ticks or colormap descriptions do not look very nice in a screenshot/picture output of a model or post-processing view. The general idea of this patch is to take a raw (i.e. no axis, colorbars, ticks or labels) png file output and use LaTeX for decorating it. It is based on the `pgfplots` package by Christian Feuersänger.

Jumpstart

Please see the documentation at the end of this document.

Figure 1 shows the side-by-side comparison of a mesh which was saved in the `png` format (left) and the new `pgf` output.

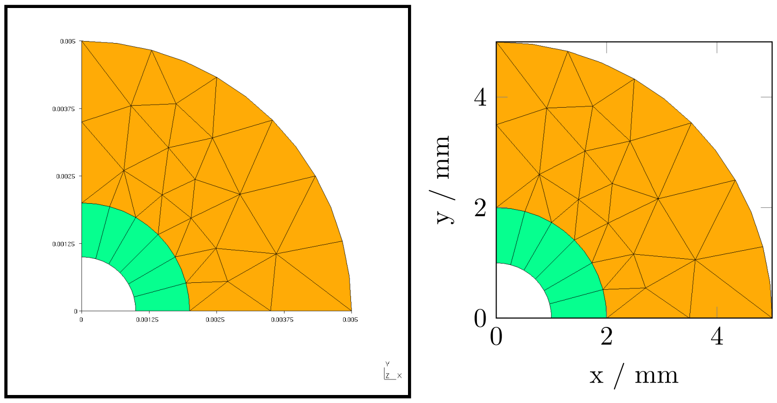


Figure 1: Flat (two-dimensional) mesh output with axes. The bounding box was automatically clipped as the tool `convert` (imagemagick) was found.

Show room / Demos

Mesh and Model

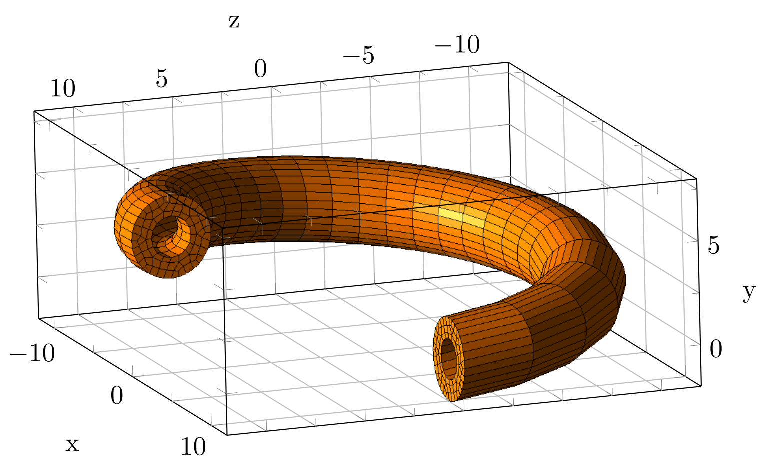


Figure 2: Three-dimensional output with axes. The background was automatically removed as the tool `convert` (imagemagick was found. The figure shows the effect of the key `3d box` (manually added to the `.pgf` file). Notice that the grid lines `behind` the figure are indeed invisible, whereas the lines completing the box in the foreground are drawn on top.

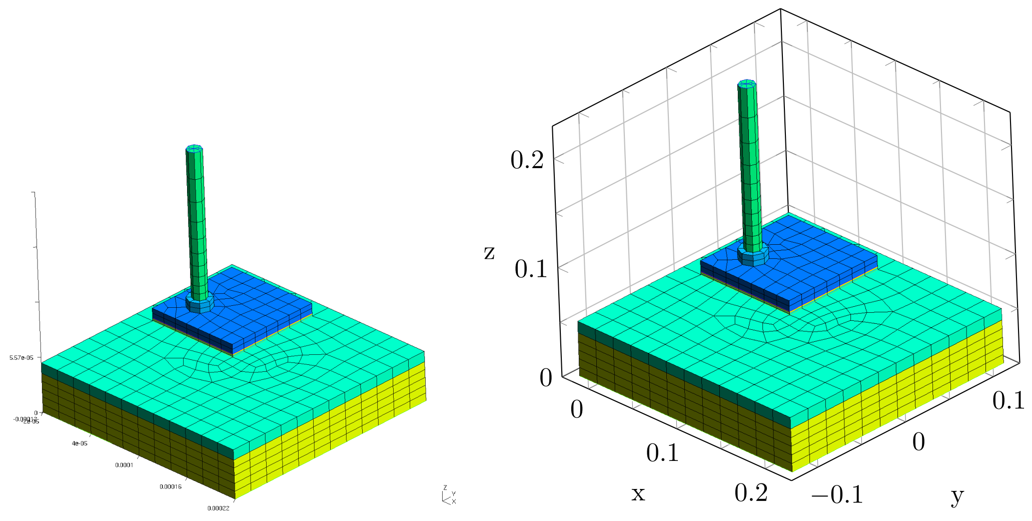


Figure 3: Side-by-side comparison of a mesh only output. To increase the axis margins, custom `xmin=..., xmax, ...` have been added to the `pgf` file.

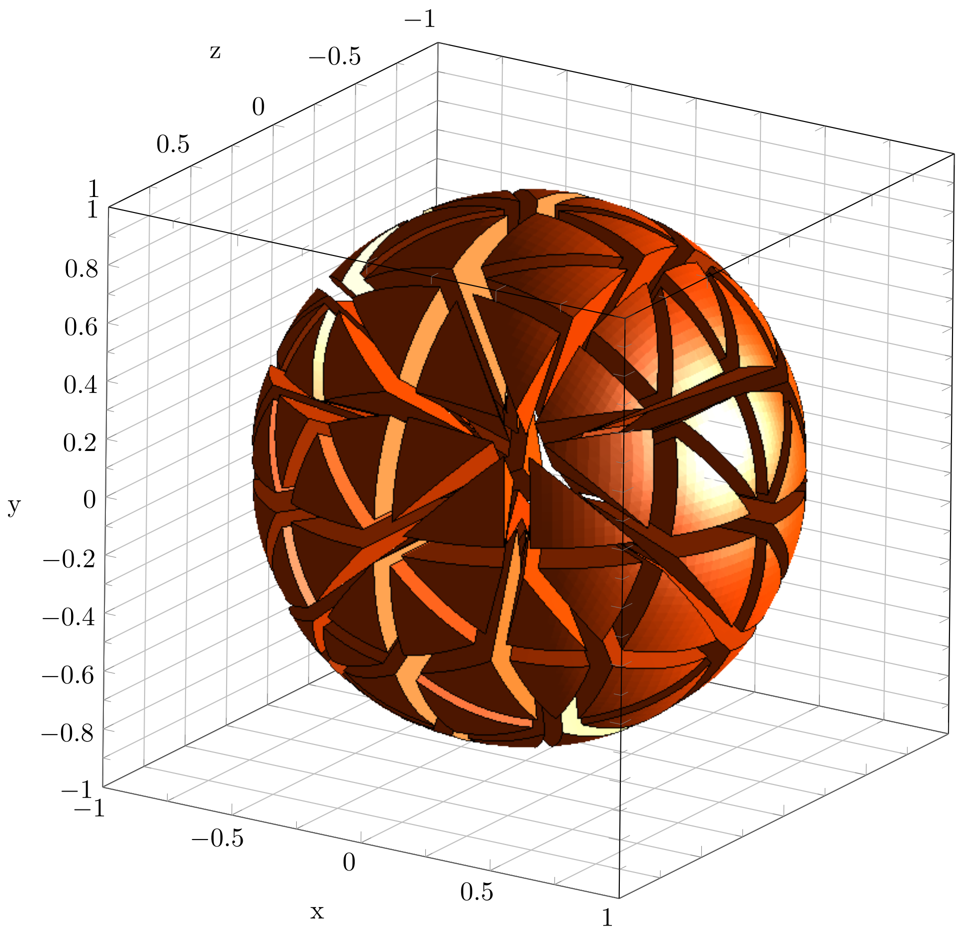


Figure 4: A higher order unit sphere mesh (see `demos/sphere.geo`) with the elements shrunk by some factor. Additionally, the key `3d box` is used.

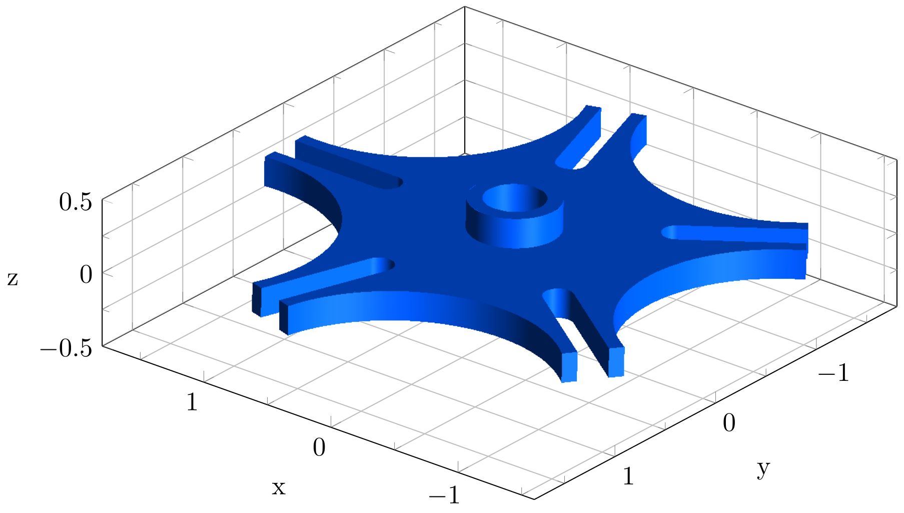


Figure 5: Another mesh (see `demos/piece-extr.geo`), but no element edges are shown.

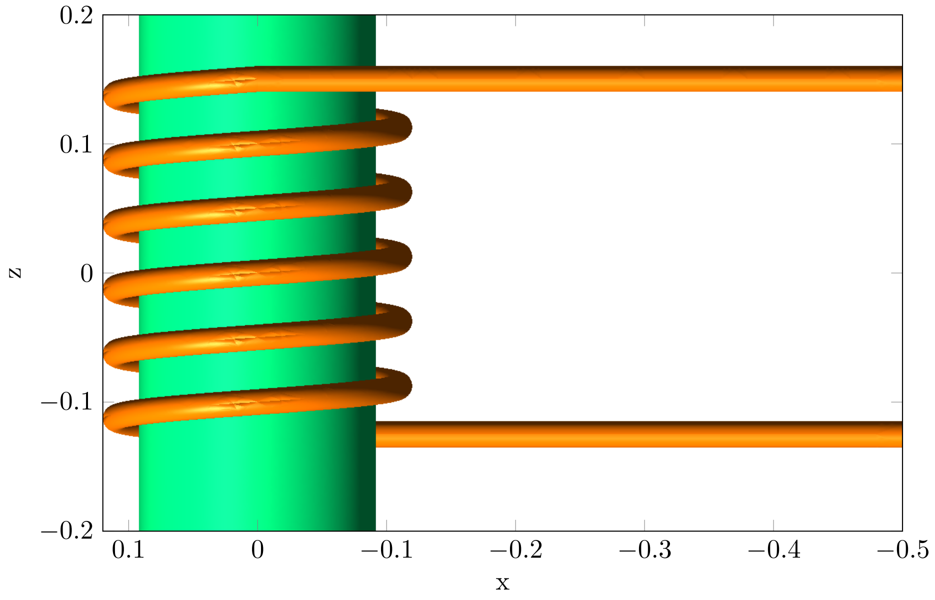


Figure 6: Flat output of a three dimensional model (see `demos/indheat.geo`). The axis labels are automatically inferred from the orientation of the model. The `gmsh` light source can be adjusted to create a pleasant plastic representation.

Post processing views

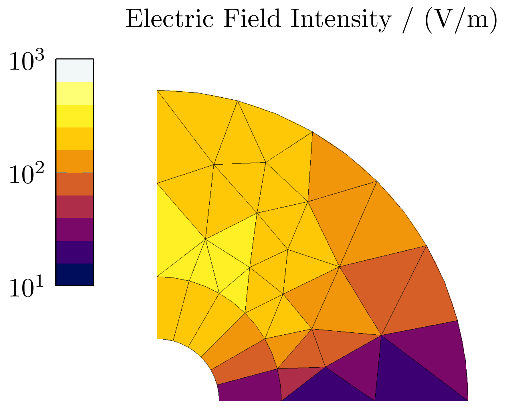


Figure 7: Flat (two-dimensional) output without axes and custom colormap size/position. Notice the *logarithmic* colormap. The colormap and its settings in `gmsh` (e.g. filled contours) are respected and exported into the `pgf` file.

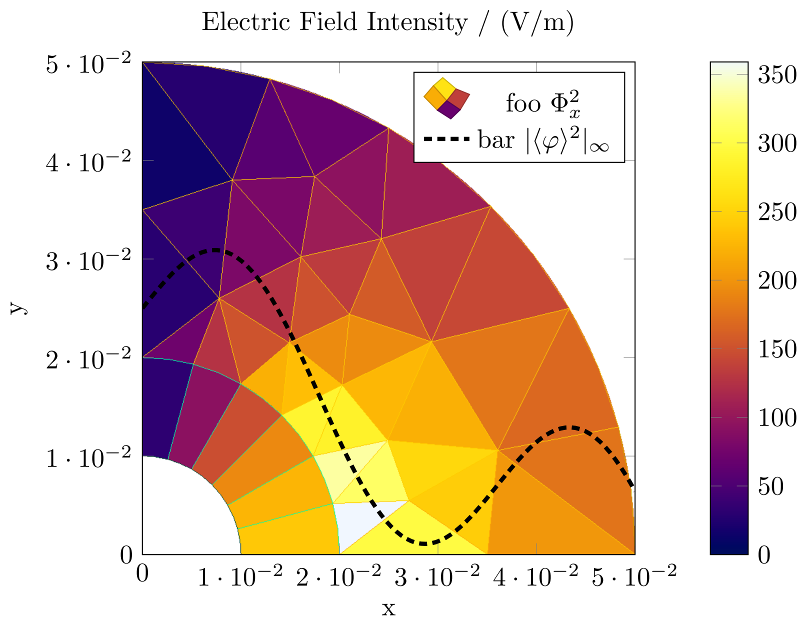


Figure 8: Flat (two-dimensional) output with axes and custom colormap. The legend for the surface (◆) and line plot (----) were added manually. See the manual for `pgfplots` (`http://pgfplots.sourceforge.net/pgfplots.pdf` for more examples.

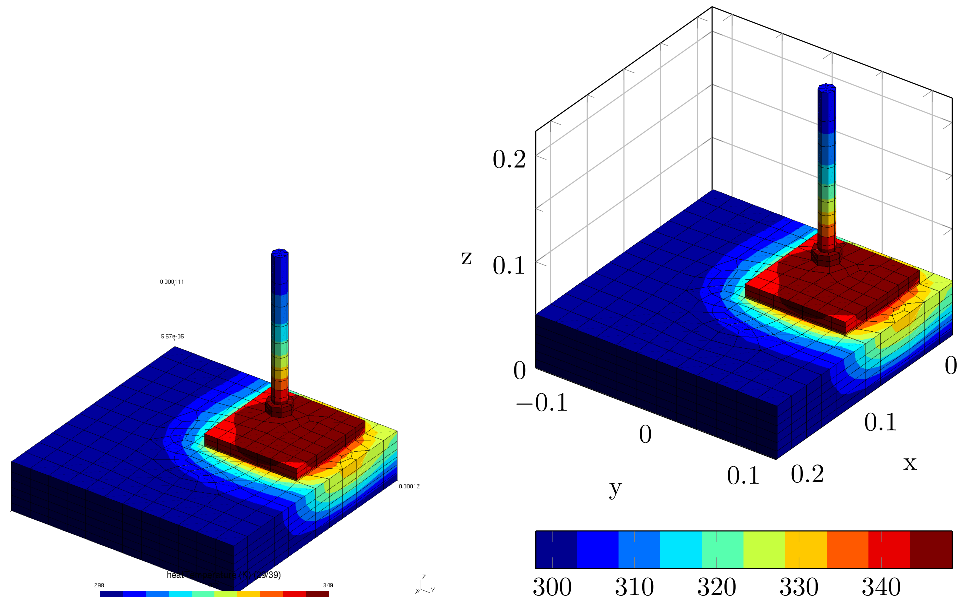


Figure 9: Side-by-side comparison of a post processing view of the *png* and the *pgf* output using the default gmsh colorbar. The option *horizontal colorbar* was selected in the file dialog.

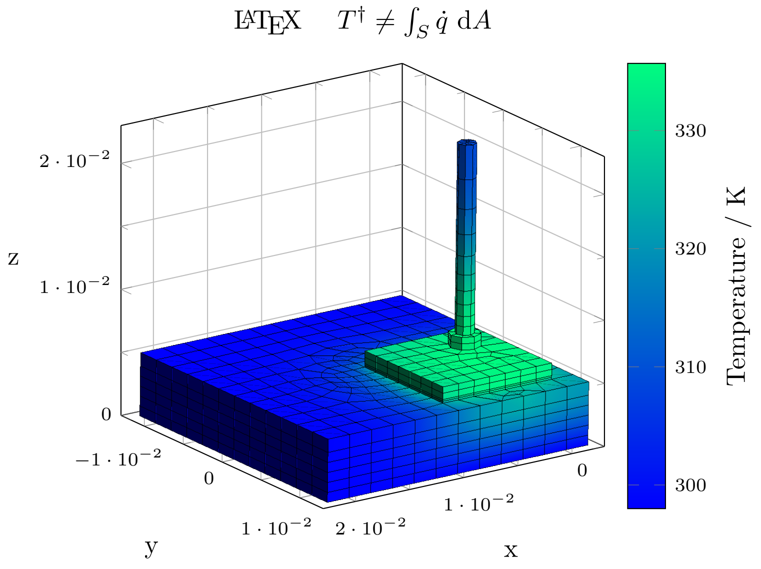


Figure 10: Demonstration of custom color export. In the title, LaTeX math is used. Also, `font=scriptsize` is used for the tick labels.

Main workings

Saving your model or post-processing view as a `.pgf` file will put **three** files in your folder.

- `.tex` file: A minimal file which allows you to get started. Run e.g. `pdflatex` on this file. Its content is simply:

  ```
  \documentclass{article}
  \usepackage{pgfplots}
  \pgfplotsset{compat=1.8}
  \begin{document}
  \input{perspective.pgf}
  \end{document}
  ```

- `.pgf` file: Included by the `tex` file. This is in fact also a `tex` file, but since there is already a tex output in `gmsh`, I selected this file extension. In this file, the custom colormap (if any) and the definitions for the plot as inferred from `gmsh` are written. You should change this file manually for adjusting the width of the figure in your document, adding legends or labels etc. The basic output seen in Fig. 1 is produced by:

  ```
  \begin{tikzpicture}
  \begin{axis}[
    width=.28\linewidth, % set figure width here
    enlargelimits=false, % tight axis, use xmin=<val>, xmax=<val> for custom bounding box
    xlabel={x / mm},
    ylabel={y / mm},
    scale only axis,
    axis equal image,
    axis on top,
    ]
      \addplot graphics[xmin=0, xmax=5, ymin=0, ymax=5]
        {cyclmsh.png};
  \end{axis}
  \end{tikzpicture}%
  ```

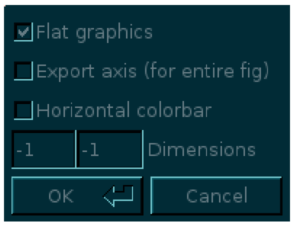
  Upon exporting a post processing view, the currently active colormap is exported to this file too.

- `.png` file: The raw png file without any axes, labels, ticks, orientation widget or colorbar. If *Imagemagick* is installed and found, the png is also automatically trimmed (flat output) or the white background is automatically replaced by a transparent layer (three-dimensional output)

Note: A message of warning is issued in case of overwriting the `.pgf` file, but non warning is given for overwriting the `.png` file. The `.tex` file is not changed if it already exists.

Prerequisites Your LaTeX distribution must come with `pgfplots` installed. MikTeX users should be fine. Ubuntu users got it installed when they installed the `texlive` package. Arch Linux has the package `texlive-pictures`. I don't know about Macs.

Options summary Upon selecting *Save as* in `gmsh` and `.pgf` is selected as file extension, the following dialog is presented:



☑ Flat graphics

☐ Export axis (for entire fig)

☐ Horizontal colorbar

-1 -1 Dimensions

Figure 1 was created using option one (*Flat graphics*) and option two (*Export axis*).

- *Flat graphics*: Selects two-dimensional (default) or three-dimensional output. Accessible via gmsh option `Print.PgfTwoDim = 1`.
  - *checked*: The raw `png` file is included, but by default no axis are drawn. If the tool `convert` (part of the *imagemagick* suite) is found, the png file is automatically trimmed (such that no borders are visible). If the tool is not installed or found, a warning is emitted. I suppose this works only on Linux and Macs.
  - *unchecked*: For three-dimensional output, orthographic projection is mandatory if you want axes. If *Export axes* is not selected, a two dimensional output is generated (see Fig. 11, left), else an error is shown. The same is true for camera views (see Fig. 11 for an example).
- *Export axes*: Draw two or three axes. This option is only useful if you plot your full figure, or if you adjust the values of the axes manually (`General->Axes->Set position and size of axes`). Accessible via gmsh option `Print.PgfExportAxis = 1`.
  - For a flat output, the minima and maxima of the current orientation are inferred from the *Euler angles*. Labels are set (if none are set in `gmsh`, the default *x,y,z* are used).
  - Because of a limitation of TEX internals, very small values need to be scaled up, and large values are scaled down. In such a case, a warning is written to the console, and the labels are appended with units (e.g. x / μm).
  - For a three-dimensional output, `pgfplots` needs to know the orientation of the axes (unit vectors) and the origin of the figure. This is not trivial and may fail in some cases. The key is to provide `pgfplots` a mapping of *world* coordinates (x, y, z) and pixel coordinates of the `png` file. This mapping is determined from the current view in the *opengl* window of `gmsh`. Hence, the `png` file originating from the three-dimensional `pgf` output must not be cropped. Instead, the white background is removed and made transparent. Again this works automatically if the tool `convert` of the *Imagemagick* suite is found on the command line. If the command fails, a warning is issued and you need to do that manually (e.g. with gimp).

    Again: This feature works well if your entire scene is visible. Should you encounter problems, check the output of your LATEX compiler for hints or see the documentation of `pgfplots` Chapter *Using External Graphics as Plot Sources*.
- *Horizontal Colorbar*: Although easily customizable int he `pgf` file, this option is provided for convenience. It causes the colorbar (if any) to be drawn horizontally. See Fig. 9. Accessible via gmsh option `Print.PgfHorizontalBar = 1`.

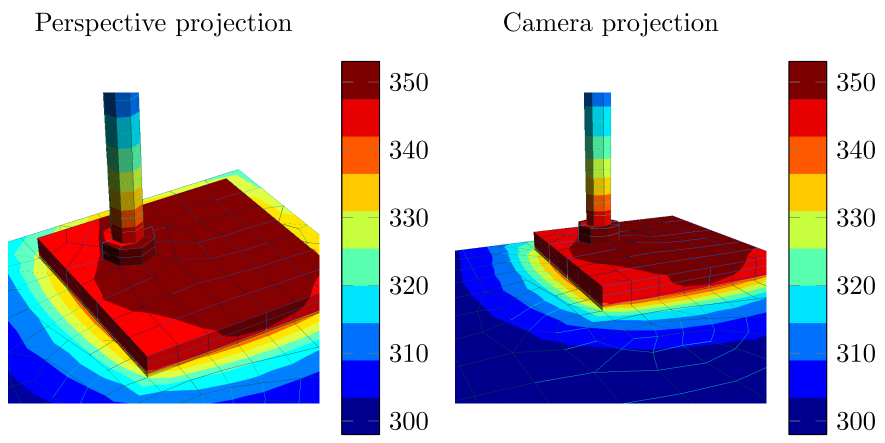


Figure 11: Three-dimensional view, but a two-dimensional output was produced, because the *prespective projection* (left) or the *camera projection* is used. No axes are possible in these scenaria.

Customization Please see the excellent pgfplots manual `http://pgfplots.sourceforge.net/pgfplots.pdf` for customization like legends, superimposed plots etc.