User-friendly web utilities for generating LaTeX output and METAPOST graphics

Troy Henderson

Abstract

There are several facets of the creation of LaTeX documents and METAPOST graphics that deter users from initially trying both LaTeX and METAPOST. These include the basic structure of the source files, the compilation of the source files, and the conversion of the output to a desired format. Furthermore, many TeX users often desire to create 2D and 3D graphs of functions for inclusion into their documents. Many of these types of graphs require considerable amounts of source code to create professional quality graphics, and this is yet another deterrent for those who might otherwise consider using METAPOST. This article introduces several free web utilities that aim to eliminate each of these obstacles and describes the usage and methods of these utilities.

1 LaTeX Previewer

LaTeX is a powerful typesetting system, but there are several reasons that discourage document preparers from using (in fact, even trying) LaTeX. Probably the most common reason is the fact that LaTeX is not a WYSIWYG word processor such as most document preparers are accustomed to using. As a result, these preparers might be interested in trying LaTeX while at the same time they might also be overwhelmed by LaTeX due to some of its characteristics. Several of these characteristics could include, for example, LaTeX's large installation size, relatively complex (source) document structure for beginners, compilation process, and lack of real-time previewing of LaTeX output.

Several years ago, the LaTeX Previewer was created to address these issues and provide beginners with a user-friendly interface to LaTeX that did not require users to download and install LaTeX and did not require any knowledge of the LaTeX document structure or compilation process. The LaTeX Previewer can be (freely) used by visiting

`http://www.tlhiv.org/ltxpreview`

Figure 1 shows the initial display for the Previewer. Beginners simply type source code and preview the corresponding output by selecting the Preview button. If a user requires the inclusion of particular LaTeX packages, then the Packages button can be used to accesses a user-friendly interface for adding and removing different packages. If the desired package is not listed, then the user is advised to send an email (and is provided a link) to request additional packages. Figure 2 illustrates an example of how the source code is rendered. If the user makes a mistake in typing the LaTeX source code and if a compilation error occurs, the user is informed of this error and is encouraged to view the compilation output log using the Log button. If compilation is successful, the (cropped) output can be downloaded in a variety of different formats including LaTeX source, Encapsulated PostScript (EPS), Portable Document Format (PDF), Portable Network Graphics (PNG), Scalable Vector Graphic (SVG), and Adobe Shock-Wave Flash (SWF).

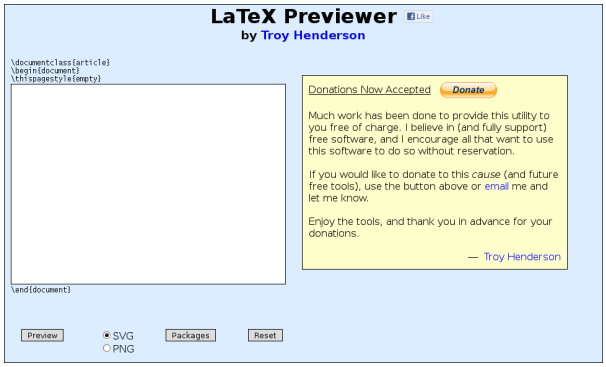

Figure 1: LaTeX Previewer Initial Display

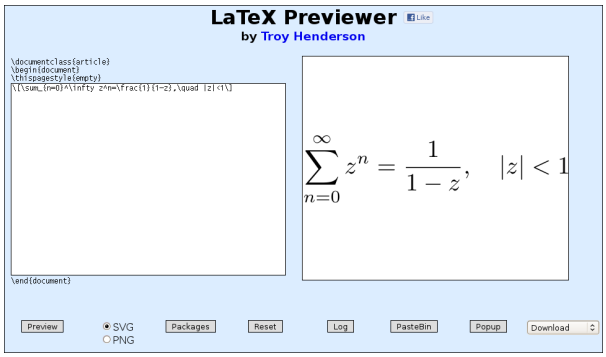

Figure 2: LaTeX Previewer Example Run

There is also a PasteBin button for sharing the source code with others as well as a Popup button for opening a larger window to view the output. Finally, the user's web browser is detected and it is automatically determined if proper SVG support is available, and if so, the default rendered output format is SVG. If the user's browser does not have proper SVG support, then the failsafe PNG rendered output is used. Furthermore, the user can override the automatic output rendering format by manually selecting either the SVG or PNG radio button.

2 METAPOST Previewer

METAPOST, like LaTeX, can be unwelcoming to beginners who wish to create professional quality graphics. These beginners, when first witnessing METAPOST examples, often wish to start creating their own graphics by typing source code. A typical METAPOST source file often needs `prologues` set appropriately to ensure that fonts used throughout the graphic are embedded in the resulting output. Furthermore, since many METAPOST beginners are experienced in LaTeX, they typically would like to typeset labels in METAPOST using LaTeX. In order to accomplish this, several commands are required in the METAPOST source. Since a METAPOST source file can accommodate multiple figures, the output naming scheme can be quite confusing to beginners, and it may also be unclear that each output is in fact an EPS file (even though each output filename extension may not be `.eps`).

These issues, as well as several others, were the primary motivation for creating the METAPOST Previewer. The METAPOST Previewer was introduced in *TUGboat* [2] in 2007, but it has since been given more features and made more user-friendly. Like the LaTeX Previewer, the goal was to provide a user-friendly interface to METAPOST that makes these issues transparent to beginners. Also, like the LaTeX Previewer, the METAPOST Previewer can be (freely) used by visiting

`http://www.tlhiv.org/mppreview`

The user interface for the METAPOST Previewer is virtually identical to that of the LaTeX Previewer, and an example is shown in Figure 3. The only noticeable difference between the two Previewers is that the Packages button for the METAPOST Previewer provides the user with the ability to select both LaTeX *and* METAPOST packages. For example, the user can choose the LaTeX package `arev` to have labels typeset in the Arev Sans (a derivative of Bitstream Vera Sans) font and also choose the METAPOST package `boxes` (for drawing a variety of boxes in METAPOST).

3 Previewer framework

Both Previewers have a user interface that is written in HTML (the main markup language for web pages) and CSS (a style sheet language for adjusting the appearance of web pages). This user interface provides dynamic interaction using JavaScript (a client-side scripting language used to enhance websites). When users enter source code into the Previewer and select Preview, the source code is posted to a CGI script (a standard method for processing HTML form data) written in Perl. The CGI script creates a LaTeX/METAPOST source file that contains the preamble presented on the Previewer's page, any user-selected packages to be included in the preamble, as well as the user-provided source code. The CGI script then compiles the source file with `latex` or `mpost` using a *halt on error* interaction method in order to determine whether compilation errors occur. If no errors occur, the LaTeX/METAPOST output (DVI/EPS) is converted to the on-screen display format, returned to the Previewer, and the output is then available for conversion to the downloadable formats listed above. Figure 4 illustrates the steps used to convert the source to each available format.

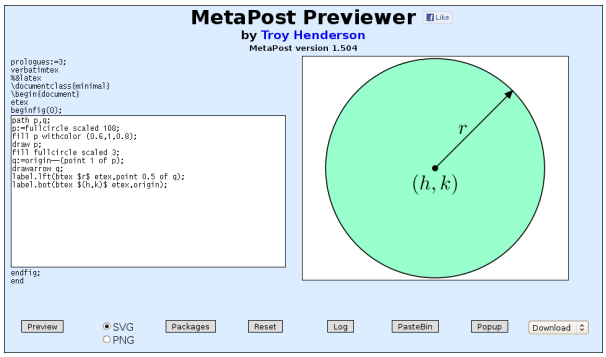

Figure 3: METAPOST Previewer Example

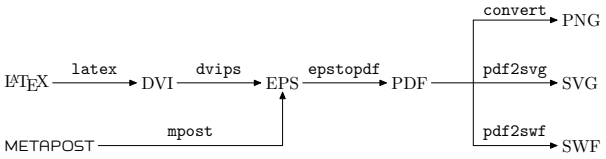

Figure 4: Previewer Conversion Process

The first few commands, namely `latex`, `dvips`, `mpost`, and `epstopdf` are included in most TeX distributions, and these sequential commands provide PDF output which is used as the basis for all other downloadable formats. The `convert` command is provided by ImageMagick [3] which (quoting from its website) "is a software suite to create, edit, compose, or convert bitmap images". Furthermore, `pdf2svg` is a utility by David Barton [1] which uses Poppler and Cairo to convert PDF documents/graphics to SVG format. Finally, `pdf2swf` is part of the SWFTools [5] suite and is a PDF to SWF converter.

4 Function Grapher

A standard type of graphic that TeX users often include in their documents are graphs of functions, curves, and surfaces. There are a variety of commercial applications available that can generate these

graphs, and each of these applications requires at least a moderate level of expertise in not only generating the graph but also exporting the graph to a TEX-friendly format. Furthermore, these graphs often appear *out of place* in TEX documents since the fonts may not be consistent with the document font and the line widths of the curves may not be consistent with default TEX document rules (lines). Of course, these issues can be addressed, but typically even more expertise is needed to accomplish this consistency. Finally, these commercial applications can cost anywhere between $100 to $2,500 (USD) which may be impractical for many users. There is an abundance of free (many open-source) applications which can produce these graphs as well, but, typically, an even greater level of expertise than for the commercial applications is needed to produce high-quality graphs.

The primary reason for creating the Function Grapher was to provide a free utility that produces publication-quality graphs with a user-friendly interface. The user interface of the Function Grapher is similar to that of the Previewers, and its output is designed to mimic MATLAB's [4] graphs with several personal preferences incorporated. Like the Previewers, the Function Grapher can be (freely) used by visiting

`http://www.tlhiv.org/mpgraph`

Figure 5 illustrates a preview of the Function Grapher, and shows the polar plot of $r(t) = \sin(8t/5)$ with $0 \leq t \leq 10\pi$.

Currently, the Function Grapher can graph (up to) three functions of a single variable simultaneously, parametric plane curves of a single variable, polar curves, contour plots of a function of two variables, slope fields of first order ordinary differential equations, parametric space curves, surface plots of (up to) three functions of two variables simultaneously, and parametric surfaces. Each type of graph has two predefined examples (accessed using the corresponding Example button) that can be used to illustrate syntax and output quality. The Options button provides users with the ability to adjust the appearance of each graph. Users can adjust the graph's aspect ratio, color scheme, axis labels, axis and grid visibility for each type of graph and can adjust mesh visibility for 3D surfaces. The output can be downloaded in a variety of formats for stand-alone graphics or for insertion into TEX documents.

5 Function Grapher framework

Like the Previewers, the user interface for the Function Grapher is written in HTML and CSS, and dynamic interaction is accomplished using JavaScript.

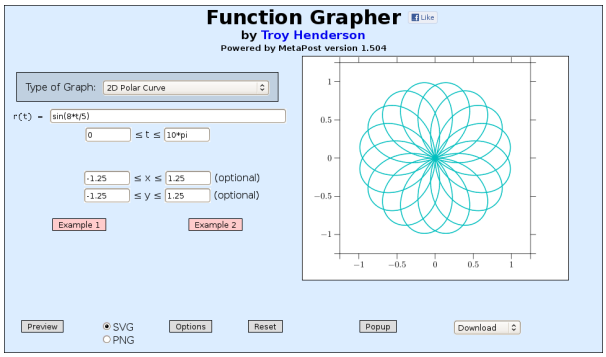

Figure 5: Function Grapher Example

Also, like the Previewers, when users enter functions to graph and select Preview, the function information is posted to a (Perl) CGI script which processes the HTML form data. The CGI script creates a uniform partition of each independent variable's interval and then evaluates the function(s) at the nodes of this partition. The density of the partition is chosen to balance quality with computation time, and the extreme values of each function are determined from these evaluations.

The tick marks for both the independent and the dependent variables are then computed using Wilkinson's algorithm [6], and clipping of the curve or surface is performed if the user specifies a range on the dependent variable(s) which is more narrow than the computed extreme values. The 3D graphics are represented using an orthographic projection of the xyz-space in which the uv-projection plane is orthogonal to a viewpoint vector w on the unit sphere. That is, $w = \langle \cos\theta\cos\phi, \sin\theta\cos\phi, \sin\phi \rangle$ where $-\pi \leq \theta < \pi$ and $-\frac{\pi}{2} \leq \phi \leq \frac{\pi}{2}$. The uv-projection plane is chosen so that the z-axis is projected to the v-axis (i.e., the z-direction is always drawn vertically). Both θ and ϕ can be specified using the Options button, and have default values of $\theta = -127.5°$ and $\phi = 20°$. These values can be changed manually by editing the corresponding HTML form fields or semi-automatically by clicking the rotation arrows overlaid on the previewed graphic. The final steps in creating each graph are accomplished by the CGI script writing a METAPOST source file and then generating output using the process described in Figure 4.

6 Function Grapher examples

This section gives several example graphs generated by the Function Grapher. Figure 6 is the graph of

three functions of a single variable, namely

$$\begin{aligned} f(x) &= \sin(2\pi x) \\ g(x) &= \sin\big(2\pi x - \tfrac{2\pi}{3}\big) \\ h(x) &= \sin\big(2\pi x + \tfrac{2\pi}{3}\big) \end{aligned}$$

representing three phase power.

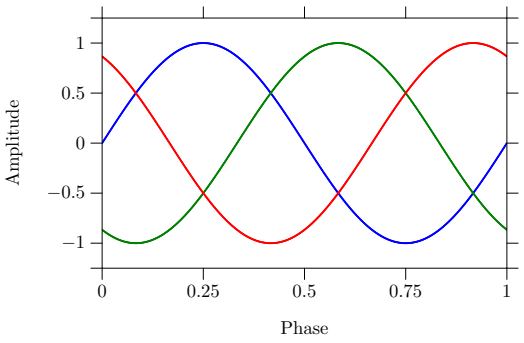

Figure 6: Three Phase Power

Figure 7 shows several contours (level curves) of the function

$$f(x,y) = 3(1-x)^2 e^{-x^2-(y+1)^2} - 10(\tfrac{x}{5} - x^3 - y^5)e^{-x^2-y^2} - \tfrac{1}{3}e^{-(x+1)^2-y^2}.$$

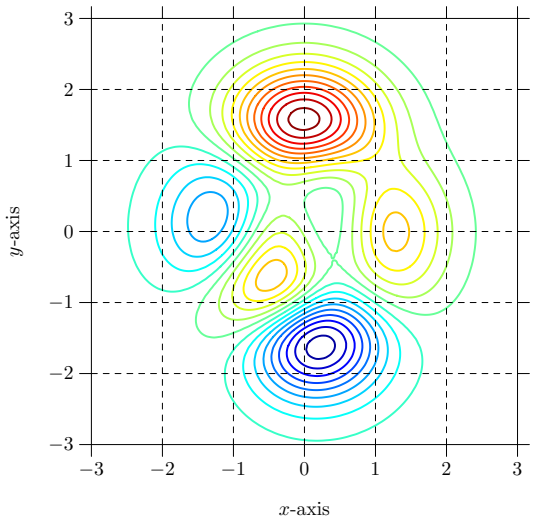

Figure 7: Contour (Level Curve) Plot

Figure 8 is the slope field plot of the ordinary differential equation

$$\begin{cases} \dfrac{dy}{dx} = x + y \\ y(0) = 0 \end{cases}$$

where the numerical solution is approximated using a fourth order Runge-Kutta method, and Figure 9 is the surface plot of the function

$$f(x,y) = e^{-x^2-y^2}.$$

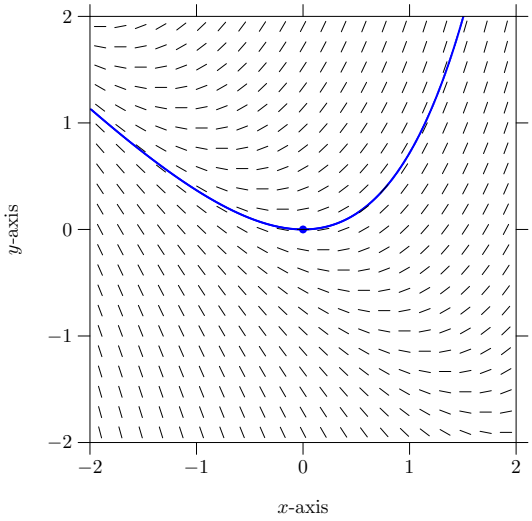

Figure 8: ODE Slope Field Plot

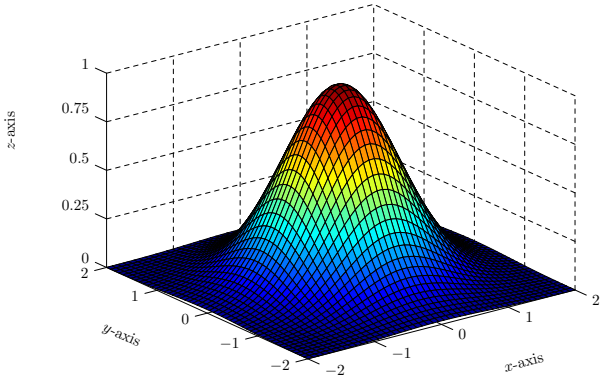

Figure 9: Surface Plot of Function of Two Variables

Figure 10 is the graph of the parametric (torus) surface parameterized by

$$\begin{aligned} x(s,t) &= (3+\cos s)\cos t \\ y(s,t) &= (3+\cos s)\sin t \\ z(s,t) &= \sin s, \end{aligned}$$

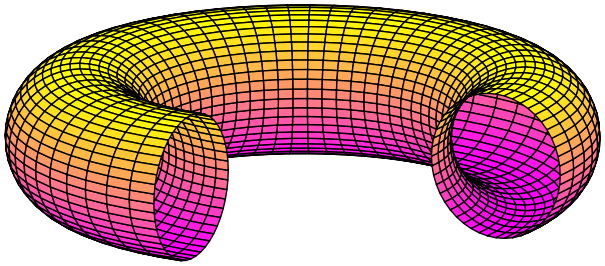

Figure 10: Surface Plot of Torus

Figure 11 is the graph of the parametric (heart) surface parameterized by

$$\begin{aligned} x(s,t) &= \tfrac{1338}{557} r(t)\cos(t)\cos(s) \\ y(s,t) &= 2\sin(s) \\ z(s,t) &= \tfrac{1338}{557}\left[r(t)\sin(t) + \tfrac{970}{557}\right]\cos(s) \end{aligned}$$

where

$$r(t) = 2 - 2\sin(t) + \frac{\sin(t)\sqrt{|\cos(t)|}}{\sin(t) + \frac{7}{5}}$$

with $-\frac{\pi}{2} \le s \le \frac{\pi}{2}$ and $-\pi \le t < \pi$.

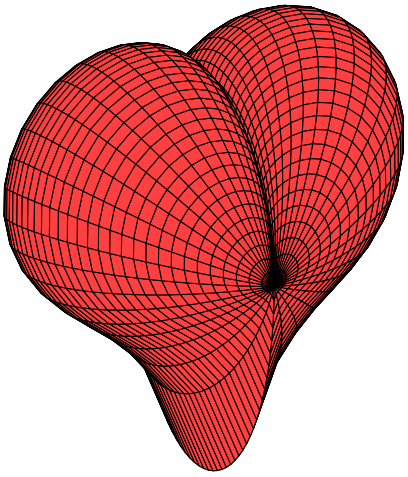

Figure 11: Surface Plot of a 3D Heart

Finally, Figure 12 is the graph of the parametric space curve parameterized by

$$\begin{aligned} x(t) &= \cos(6t) \\ y(t) &= \sin(6t) \\ z(t) &= t. \end{aligned}$$

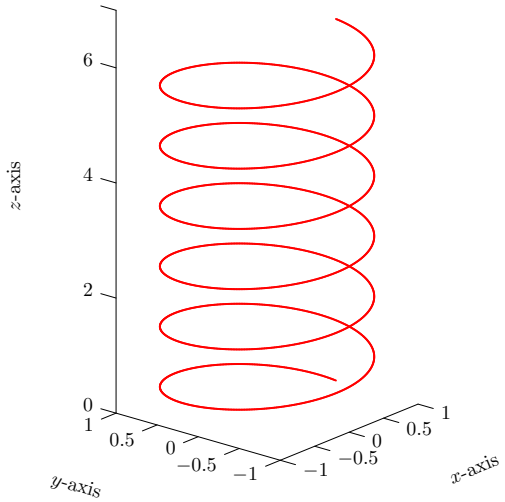

Figure 12: Plot of a Helix Space Curve

7 Conclusion

It is the author's desire that these web-based utilities will simplify the introduction to LaTeX and METAPOST and, as a consequence, help users prepare professional-quality manuscripts (both text layout and graphics). These utilities will continue to be available for free, and users are encouraged to direct feature requests and any other type of feedback to the addresses below.

References

[1] David Barton. `http://www.cityinthesky.co.uk/opensource/pdf2svg`.

[2] Troy Henderson. A beginner's guide to MetaPost for creating high-quality graphics. *TUGboat*, 28(1):85–90, 2007.

[3] ImageMagick. `http://www.imagemagick.org`.

[4] Mathworks. `http://www.mathworks.com/products/matlab`.

[5] SWFTools. `http://www.swftools.org`.

[6] Leland Wilkinson. *The Grammar of Graphics*. Springer-Verlag New York, Inc., 2005.

◇ Troy Henderson
Department of Mathematics
University of Mobile
5735 College Parkway
Mobile, AL 36613 USA
`thenderson (at) umobile dot edu`
`http://www.tlhiv.org`