A Tool to Support LaTeX Markup in MATLAB
publish2latex
Version 1.1

Matthew Harker and Paul O'Leary
Institute for Automation
University of Leoben
A-8700 Leoben,Austria
URL: automation.unileoben.ac.at

Original: January 9, 2013

Last Modified: June 25, 2013

Contents

Abstract

This tool extends the concept of publishing MATLAB code by enabling the full functionallity of LaTeX. Presently the tool is no compatible with the `publish` tool in MATLAN, it is an independent toolbox. The `m` code is execued, the figures and the MATLAB screen outputs are captured. The MATLAB code is placed as a *listing*, the screen output is then added as a *verbatim* and the figures are implemented as eps files. A table of contents is also produced. All LaTeXcommands and environments are available, including references to Bibtex etc. This document has been produced using this tool.

1 Introduction

This tool extends the concept of publishing MATLAB code by enabling the full functionallity of LaTeX. Presently the tool is no compatible with the `publish` tool in MATLAN, it is an independent toolbox[1].

The concept is you use LaTeXmarkup in comment lines, whereby the full functionality of LaTeXis available.

We take advantage of the cell structure provided by MATLAB. Each file is segmented into the individual cells. The first cell mus be the file header. Between the file header and the second cell we automatically insert the file *LatexDefinitions.tex*, which is contained in the library. This enables the definition of addition macros and packages you might wish to use.

Before starting the MATLAB run we set default values for fonts and figur sizes. This ensures tha the formating of the text in the figures and the size of the figures is consistent.

This tool has been used to generate the documentation for the discrete orthogonal polynomial toolbox DOPbox and the ordinary differential equation toolbox ODEbox which are available at MATLAB Central

```
http://www.mathworks.com/matlabcentral/fileexchange/41250
http://www.mathworks.de/matlabcentral/fileexchange/41354
```

You may find a collection of m-code examples and resulting PDF documentation in that library.

2 Basic Structure

2.1 Cell 1

The first cell in the m-file must contain the Latex document header containing the fololowing information. The present document provided an example.

```
\documentclass[12pt]{article}

\title{.....}

\author{.....}
```

[1] We are presently working on a converter from MATLAB publish to publish2latex

An empty template for `publish2latex` can be found in the directory where this tool resides, it has the file name `publishTemplate.m`.

2.2 Intermediate cells

Each cell in the m-file is considered to consist of three parts:

1. Pre-m-code lines: these are interpreted as being LATEX commands.
2. m-code: Each cell contains only one code block and it starts with the first non-comment line in the cell. All the comments in the m-code part are dealt with as m-code comments and not interpreted as Latex commands.
3. post-m-code: the last comment lines in a cell may be used to a caption to the preceeding graphics figures. No other Latex code should be used in this section.

The lines between the end of the code and the start of the next cell can be used to define a caption (no other LaTeXcode should be added here) for the figures produced by the code.

Lines which start with `%%` will be passed to LaTeXas a comment line.

2.3 Last cell

The last cell in the m-file defines the bibliography style and bibtex files, e.g.

```
\bibliographystyle{IEEETran}
\bibliography{myBibliography}
```

3 Calling `publish2Latex`

After installation of the package the tool is simply activated using the command `publish2Latex`. This starts a simple pupup menu which is self explanatory.

The tool requires no external programs to generate a LaTeXfile. Consequently, this section of code should also function on Apple 8hoever has not bee tested).

The following programs must be installed if the tool is to automatically generate PDF documents:

latex To compile the .tex file to a .dvi file

bibtex Generates the necessary bibtex temporary files

dvips To convert .dvi to .ps

ps2pdf14 to convert .ps to .pdf

We have chosen to generate postscript and them PDF because we feel this is the most advantageous way in the long run. You may choose to generate postscript and thes distill with other settings.

4 Listings Package

This tool uses the *listings* package, consequently all the standard listings environments are available. You may need to install the listings package, see:

`ftp://ftp.tex.ac.uk/tex-archive/macros/latex/.../listings/listings.pdf`

for details.

Complete MATLAB files can be included in the final documentation if desired using the listings package, for example,

```
\lstinputlisting[caption=test]{Listings/fitLine.m}
```

Code inserted in this manner is not executed by this tool.

Inline code can be implemented using,

```
\lstinline{for k=1:n}
```

to generate `for k=1:n`

5 Getting started

Preparing the MATLAB environment.

```
1 clear all;
2 close all;
3 publishFigureFormat;
```

5.1 Latex Equations

The following equations are just to test the possability of havig both latex,

$$b = a^2 \tag{1}$$

and matlab equations. Note that line 7 in the code has no semicolon and produces output to the screen, which is captured in this documentation.

```
4 a=1:10;
5 b = a.^2;
6 for k = 1:5
7    c(k) = a(k) * b(k)
8 end;
```

```
c =
     1
c =
     1     8
c =
     1     8    27
c =
     1     8    27    64
c =
     1     8    27    64   125
```

5.2 Generating Figures and Captions

Each call of the `figure` command generated a figure in the output, and after the figure a caption can be added.

```
 9 h = figure;
10 plot( a,b,'k');
```

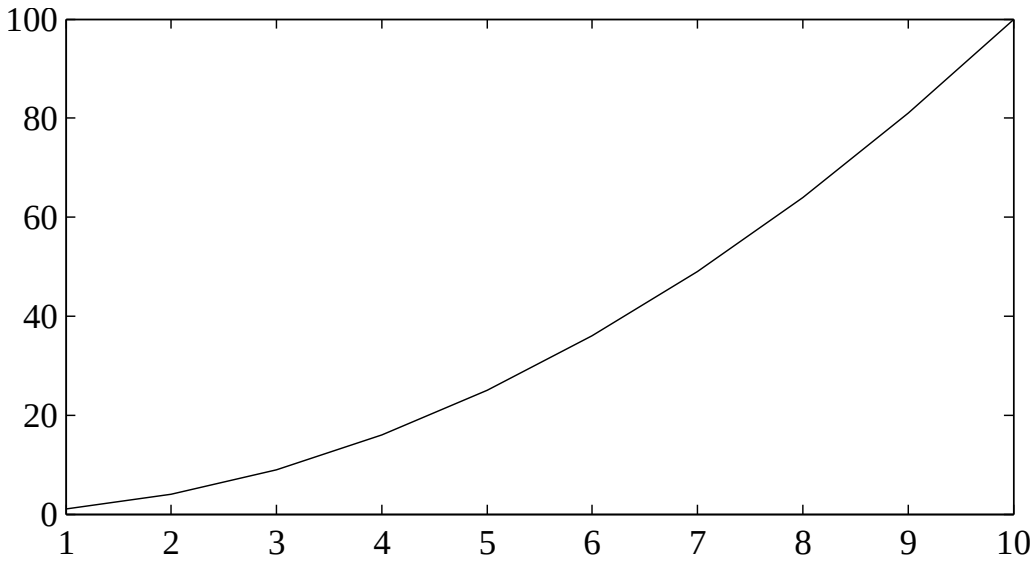
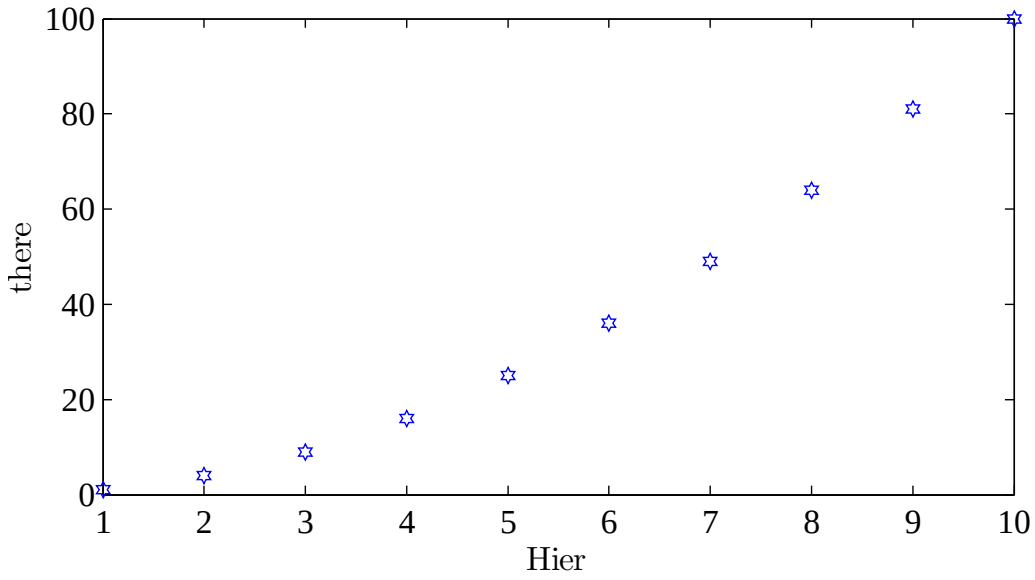


Figure 1: testCaption

6 A Figure Without a Caption

Why? just for fun.

```
11 figure
12 plot(a,b,'h');
13 xlabel('Hier');
14 ylabel('there');
```

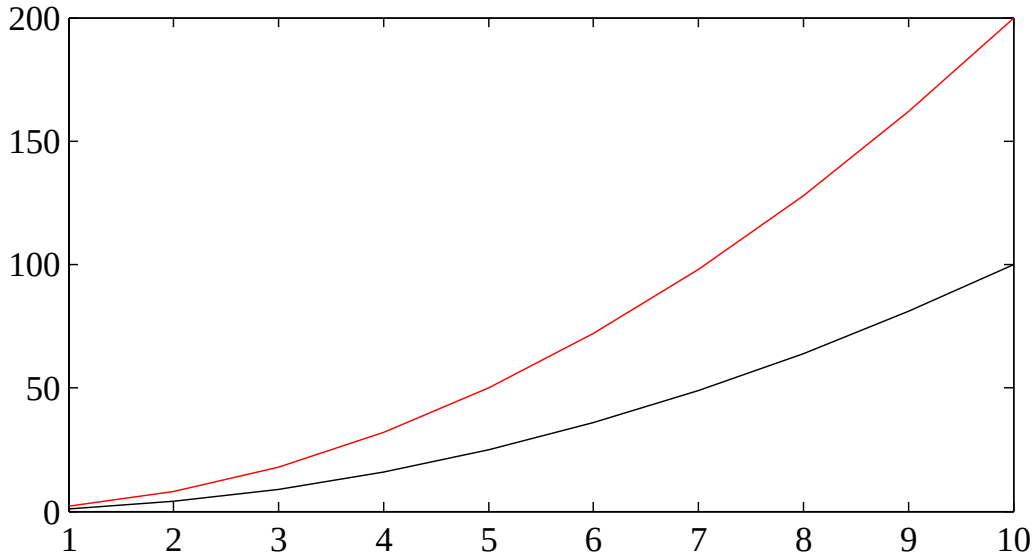



```
15 close;
```

7 Recalling a Graphic and Bibliography Citation

The command `figure(h)` with a handel recalls a figure and generates a new outpur figure. The figures are sequentially numbered and can be referenced by the label, e.g. `Figure 1`.

```
16 figure(h);
17 hold on;
18 plot( a,2*b,'r');
```



Test a reference to a bibliography [1].
and now do a loop with figure calls to a handle this should only generate one additional figure.

```
19 for k=1:2
20    figure(h);
21    plot( a, 5 * k * b, 'k');
22 end;
```

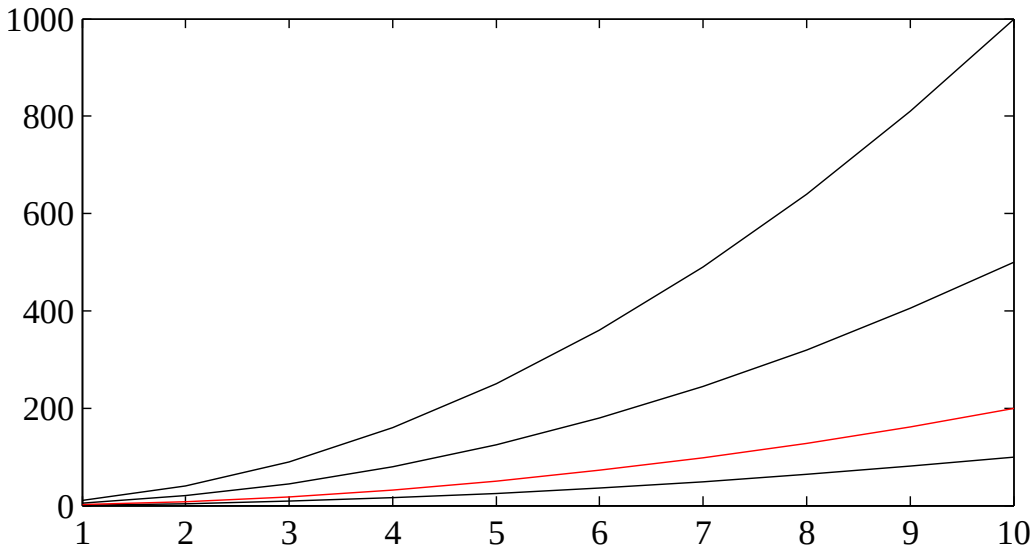


Figure 2: Figure with handel

and now do a loop with figure withpout a handle this should generate two additional figures.

```
23 for k=1:2
24    figure;
25    plot( a, 5 * k * b, 'k');
26 end;
```

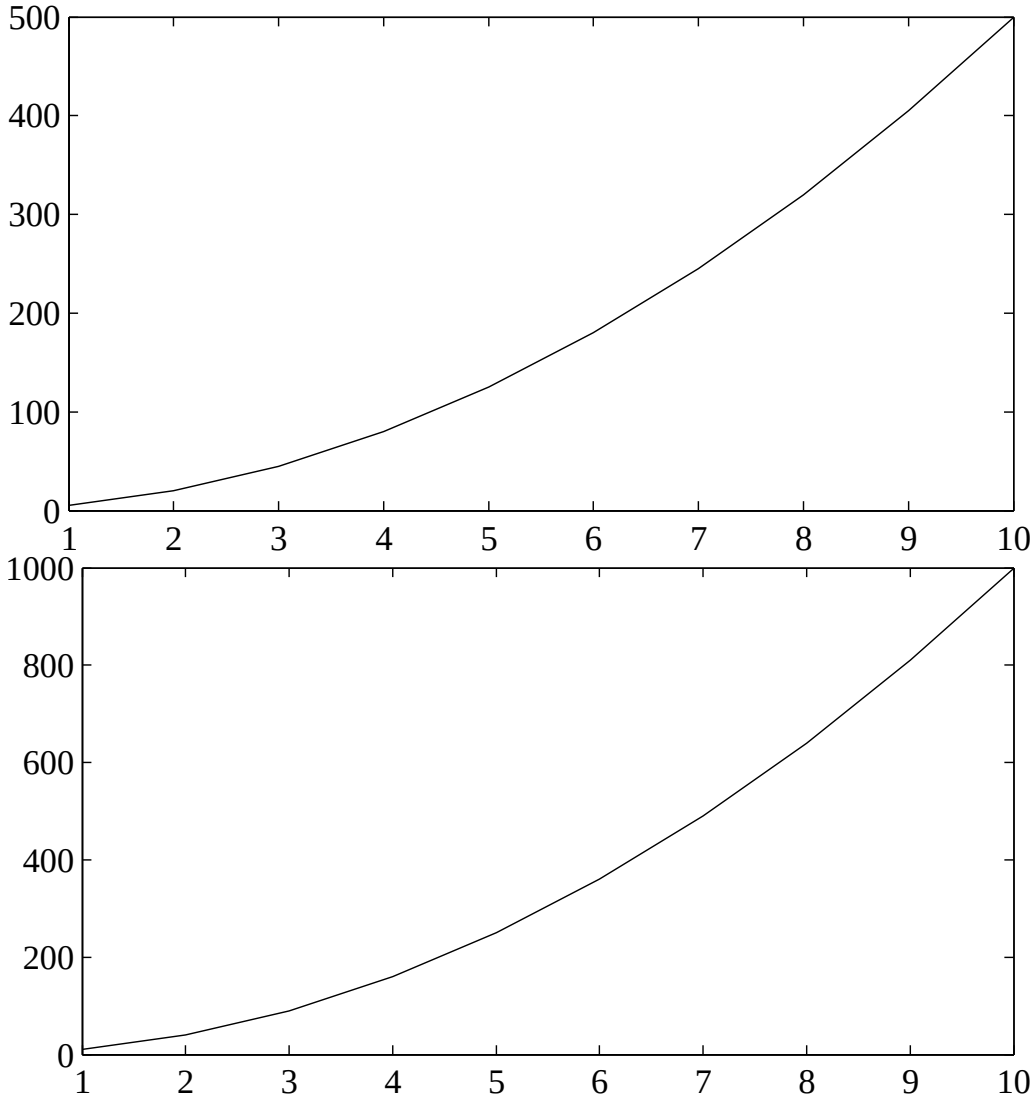


Figure 3: Loop of figures

8 Files You May Wish to Modify

There are two files in the library you may wish to modify:

1. *LatexDefinitions.tex* This file may be used ot define additional macros or include packages you may wish to use.

2. *publishFigureFormat.m* This file defines the fonts to be used in figures, sets the default interpreter in MATLAB to LaTeXand defines a default figure size.

9 Finding Errors

Finding errors in your LaTeXwhich is automatically generated can be cumbersome. Our recommendation is to generate the .TEX file and then compile this independently in your LaTeXenvironment. This will enable you to more quickly find your error.

References

[1] Paul O'Leary and Matthew Harker. Polynomial approximation: An alternative to windowing in Fourier analysis. In *International Instrumentation and Measurement Technology Conference (I2MTC 2011)*, 5 2011.