LaTeX in Matlab

Aaron Biddings

Electrical and Computer Engineering
Auburn University

July 26, 2010

Outline

1. Figure Annotation
   - Matlab Function Syntax
   - Example
2. M-File Publishing
   - What is it?
   - Output Format
3. Matrices

Outline

1. Figure Annotation
   - Matlab Function Syntax
   - Example
2. M-File Publishing
   - What is it?
   - Output Format
3. Matrices

Figure Annotation

LaTeX in Matlab

The manipulation of figure annotation is very simple and straightforward. One only needs to utilize the basic Matlab functions such as *title*, *xlabel*, *ylabel*, and *text*. In addition, Matlab's text interpreter must be set to handle LaTeX coding. This is done by simply supplying the necessary properties to the afore mentioned functions.

Figure Annotation

Here are examples showing the syntax for each function. All are the same with the exception that coordinates must also be specified for the figure text function.

title('sting','PropertyName1','PropertyValue1', ...)
xlabel('sting','PropertyName1','PropertyValue1', ...)
ylabel('sting','PropertyName1','PropertyValue1', ...)
text(X, Y,'sting','PropertyName1','PropertyValue1', ...)

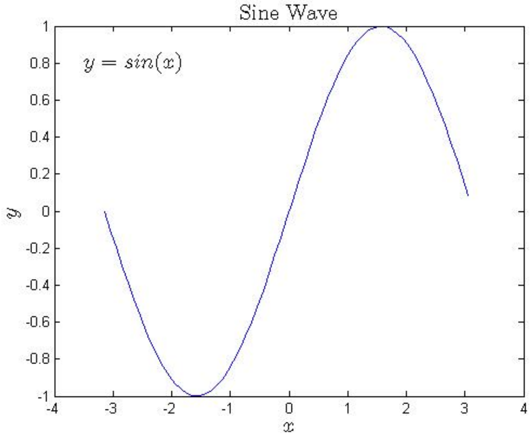
Figure Annotation

Here is shown some simple coding for a Matlab plot exemplifying one possible application of each function:

```
close
x = -pi:.1:pi;
y = sin(x);
plot(x,y)
title(['Sine Wave'],'interpreter','latex')
xlabel(['x'],'interpreter','latex', 'FontSize', 15)
ylabel(['y'],'interpreter','latex', 'FontSize', 15)
text(-3.5, 0.8,'y = sin(x)', 'interpreter', 'latex', 'FontSize', 15)
```

Figure Annotation

Results:



Note the mathematical variables and equation within the labels and text generated by LaTeX syntax.

Outline

1. Figure Annotation
   - Matlab Function Syntax
   - Example
2. M-File Publishing
   - What is it?
   - Output Format
3. Matrices

M-File Publishing

The MATLAB product allows you to quickly publish your M-file code to enable you to describe and share your code with others who may or may not have MATLAB software. Acceptable formats for publishing are: HTML, XML, LaTeX, Microsoft Word®, and Microsoft PowerPoint®.

M-File Publishing

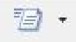
The benefit of publishing an M-File is that the generated document appears polished rather than as a mere text file of code. There are two ways to format an M-File in order to specify how the published results will appear:

1. Use drop-down menu options to automatically inserts the text markup symbols
    - **Cell > Insert Text Markup > LaTeX Markup**
2. Type the markup symbols directly in the comments

Note: Using the drop-down menu will produce a default commented text which can then be edited using LaTeX syntax.

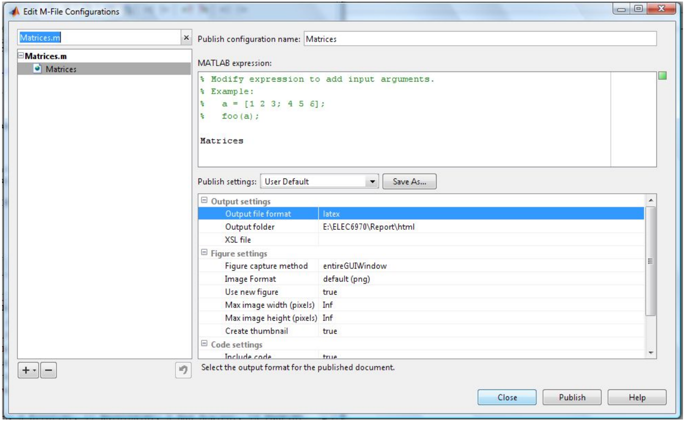
M-File Publishing

After creating the M-File, but before publishing it, the output format must be set to handle LaTeX code. In the M-File editor, click the little black arrow which is adjacent the publish button in order to display it's drop down menu.



Then select *Edit Publish Configuration for filename.m*. This will open the "Edit M-File Configurations" window.

M-File Publishing



Publish configuration name: Matrices

MATLAB expression:

```
% Modify expression to add input arguments.
% Example:
%   a = [1 2 3; 4 5 6];
%   foo(a);

Matrices
```

Publish settings: User Default

Output settings	
Output file format	latex
Output folder	E:\ELEC6970\Report\html
XSL file	
Figure settings	
Figure capture method	entireGUIWindow
Image Format	default (png)
Use new figure	true
Max image width (pixels)	Inf
Max image height (pixels)	Inf
Create thumbnail	true
Code settings	
Include code	true

Select the output format for the published document.

Next, double-click the *Output file format* property in the "Publish settings" properties table and select *latex* from the drop-down menu.

M-File Publishing

Next, select *User Default* in the Publish Settings drop-down menu and click "Save As...".
Finally, click *Publish* and *Close*.

M-File Publishing

By default, MATLAB publishes the M-file to an */html* subdirectory of the directory containing the M-file. This is where it stores the generated .tex file (which has the same name as the .m file) which now opens in the Matlab M-File editor. This file can now be compiled from a LaTeX editor to generate a LaTeX document.

Outline

1. Figure Annotation
   - Matlab Function Syntax
   - Example
2. M-File Publishing
   - What is it?
   - Output Format
3. Matrices

Matrices

The following is an example of using LaTeX code in Matlab to generate a published document showing a formatted matrix.

Matrices

Commented text was first inserted into an M-File using the drop-down menus (i.e. **Cell** > **Insert Text Markup** > **LaTeX Markup**). Then the inserted text was edited to produce a matrix using LaTeX syntax.

```
%%
%
% <latex>
% \[ \left[
% \begin{array}{ c c c }
% 1 & 2 & 3
% 4 & 5 & 6
% 7 & 8 & 9
% \end{array} \right]
% \]
% </latex>
```

Matrices

Next, publish the M-File to generate a .tex file which will automatically be saved in the newly created */html* subdirectory.

Matrices

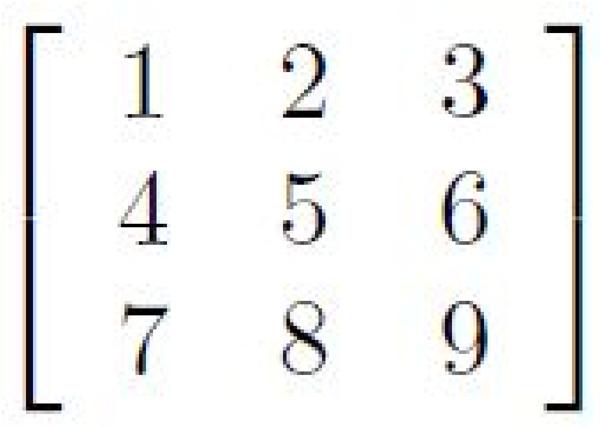
```
% This LaTeX was auto-generated from an M-file by MATLAB.
% To make changes, update the M-file and republish this doc
\documentclass{article}
\usepackage{graphicx}
\usepackage{color}

\sloppy
\definecolor{lightgray}{gray}{0.5}
\setlength{\parindent}{0pt}

\begin{document}
     \begin{par}
  \[ \left[ \begin{array}{ c c c } 1 & 2 & 3\\ 4 & 5 & 6\\
  7 & 8 & 9 \end{array} \right] \]
\end{par} \vspace{1em}
\end{document}
```

Matrices

Finally, compiling the .tex file produced the following output in a .pdf document:



$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

References

1. Matlab Help Menu (Help > Product Help)
2. http://upload.wikimedia.org/wikipedia/commons/2/2d/LaTeX.pdf
3. www.google.com

Thank You