

Publishing M-Files in MATLAB

Lisa Oberbroeckling

January 27, 2012

Contents

1	Introduction	2
2	Basic Scripts or M-files	2
3	Publishing M-files	3
4	Using Cells	4
4.1	Using Cells for publishing	4
4.2	Cells for running/debugging files	10
5	Formatting text	10
5.1	Basic text formatting	11
5.2	Lists	11
5.3	HTML Links	12
5.4	Inserting Images	13
5.5	Preformatted Text	14
5.6	Inserting HTML Code	14
5.7	Inserting \LaTeX equations	15
5.8	M-file and HTML file for this section	16
6	Summary of M-Files/Links	16

1 Introduction

This document is written to give students and others information on publishing M-files to HTML files using MATLAB's publishing feature. A good reference on publishing M-files is found at MATHWORKS.COM:

http://www.mathworks.com/help/techdoc/matlab_env/f6-22451.html

This document has many clickable links with it to either specific reference pages or to published webpages for examples. Thus it may be easiest to read this document online.

2 Basic Scripts or M-files

Here I will explain some of the syntax needed to publish m-files to webpages (HTML files). If you are doing this for an assignment or project, the specific directions of where to save these files, etc. are given separately.

First, you must understand the difference between a BASIC SCRIPT FILE (m-file) and a PUBLISHABLE SCRIPT FILE. A script file is just an ASCII (American Standard Code for Information Interchange, i.e., basic text) file with an extension of .m. Within that file are MATLAB commands as if you had typed them in at the command prompt within the Command Window. You can put comments (and should, especially for long files) to better read/debug the file. Comments start with "%". Here is an example:

basicMFile.m

```
1 % Example of Basic Script file
2 % Lisa Oberbroeckling, Spring 2012
3 % first problem
4 clc
5 x=linspace(-pi,pi);
6 y=sin(x);
7 plot(x,y)
8 % second problem
9 A=[1 2 3;4 5 6;7 8 9];
10 B=[A(1,:); -4*A(1,:) + A(2,:); A(3,:)]
11 B2=[B(1,:); B(2,:); -7*B(1,:) + B(3,:)]
12 C=[B2(1,:); -1/3*B2(2,:); B2(3,:)]
```

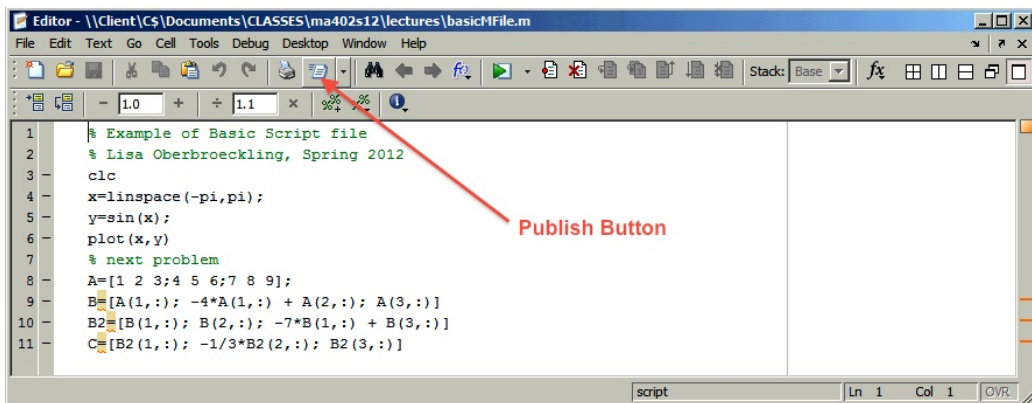
3 Publishing M-files

There are several different ways you can publish your m-files to HTML files using MATLAB. One way is to enter the following on the command line within the Command Window: `publish('filename.m')`

You can also type: `publish('filename.m', 'html')` One would do this if you've changed the publishing settings to be "latex", for example.

The other way of publishing, which is more common, is to do it within MATLAB's Editor Window and press the "Publish" button (or "Save and Publish" button if changes have been made without saving). It is between the "Print" and "Find text" (binoculars) buttons (see Figure 1).

Figure 1: Publish Button in Editor Window



This will create a folder named "html" in the current working directory (if there isn't a folder of that name already there), and put the `filename.html` file and any other files necessary for the webpage (PNG files for images, for instance) in the "html" folder. If we publish the above file, it wouldn't be very pretty. See

LINK 1: <http://math.loyola.edu/~loerbpro/matlab/html/basicMFile.html>.

Notice that the output that shown on the webpage is out of order of the commands given; the output of the next problem appears BEFORE the plot. Thus we want to format the comments in a special way so that when it is published, the MATLAB commands and output appear in order. This is the topic of the next section.

4 Using Cells

In order to format our m-file to make it better for publishing, we want to break up the commands in the m-file in CELLS. Each CELL is broken up by “%%”. When you do this within the Editor Window, you’ll notice lines appearing between each cell. Cells are useful not only for publishing, but for running and debugging scripts. Cells are used in publishing to signify different sections of the webpage (like for homework assignments to have different sections for each problem). Cells also determine how/where output for lines of code are displayed. For more detailed information, go to http://www.mathworks.com/help/techdoc/matlab_env/brqxeeu-259.html.

4.1 Using Cells for publishing

If you have text following the “%%” on the same line, this also creates a SECTION with that text as the section title. In addition, a bulleted list is created with those linked section titles at the top of the webpage. For the following we just added “%%” to two lines (lines 1 and 7); compare with the file `basicMFile.m` on page 2.

publishMFile1.m

```
1 %% Example of Basic Script file
2 % Lisa Oberbroeckling, Spring 2012
3 clc
4 x=linspace(-pi,pi);
5 y=sin(x);
6 plot(x,y)
7 %% second problem
8 A=[1 2 3;4 5 6;7 8 9];
9 B=[A(1,:); -4*A(1,:) + A(2,:); A(3,:)]
10 B2=[B(1,:); B(2,:); -7*B(1,:) + B(3,:)]
11 C=[B2(1,:); -1/3*B2(2,:); B2(3,:)]
```

LINK 2: <http://math.loyola.edu/~loberbro/matlab/html/publishMFile1.html>

The above is better than the published page without cells, but can be better. We may want the page to start with a title. This is done by adding a line “%%” after our title (and another comment line(s) for other introductory text, like my name).

publishMFile2.m

```
1 %% Example of Basic Script file
2 % Lisa Oberbroeckling, Spring 2012
3 %%
4 clc
5 x=linspace(-pi,pi);
6 y=sin(x);
7 plot(x,y)
8 %% second problem
9 A=[1 2 3;4 5 6;7 8 9];
10 B=[A(1,:); -4*A(1,:) + A(2,:); A(3,:)]
11 B2=[B(1,:); B(2,:); -7*B(1,:) + B(3,:)]
12 C=[B2(1,:); -1/3*B2(2,:); B2(3,:)]
```

LINK 3: <http://math.loyola.edu/~loberbro/matlab/html/publishMFile2.html>

If you look at the published webpage, you'll notice that we have a section link and title for the "second problem" but not for the first. So we probably want to change line 3 to include a section title:

publishMFile3.m

```
1 %% Example of Basic Script file
2 % Lisa Oberbroeckling, Spring 2012
3 %% first problem
4 clc
5 x=linspace(-pi,pi);
6 y=sin(x);
7 plot(x,y)
8 %% second problem
9 A=[1 2 3;4 5 6;7 8 9];
10 B=[A(1,:); -4*A(1,:) + A(2,:); A(3,:)]
11 B2=[B(1,:); B(2,:); -7*B(1,:) + B(3,:)]
12 C=[B2(1,:); -1/3*B2(2,:); B2(3,:)]
```

LINK 4: <http://math.loyola.edu/~loberbro/matlab/html/publishMFile3.html>

When publishing m-files, each time a new cell is started, MATLAB displays the output created by the commands of the previous cell. The difference between `publishMFile3.html` and `publishMFile4.html` is where the output is displayed for the second problem.

publishMFile4.m (partial view)

```
8 %% second problem
```

```

9 % problem 2a
10 A=[1 2 3;4 5 6;7 8 9];
11 B=[A(1,:); -4*A(1,:) + A(2,:); A(3,:)]
12 %%%
13 % problem 2b
14 B2=[B(1,:); B(2,:); -7*B(1,:) + B(3,:)]
15 %%
16 % problem 2c
17 C=[B2(1,:); -1/3*B2(2,:); B2(3,:)]
18 %%
19 % problem 2d
20 x=linspace(-10,10);
21 y=exp(x); % can comment after a command, too
22 plot(x,y)
23 title('Another Example')

```

LINK 5: <http://math.loyola.edu/~loberbro/matlab/html/publishMFile4.html>

Another important place to insert a cell break is when you want to have text following Matlab commands, but within the same section. If you just include comments after the Matlab commands, they will be formatted as comments within the displayed code, not as text. Instead, insert a cell break (without a section title) and then the comment block that will be the text:

publishMFile4.m (partial view)

```

24 %%% third problem
25 A=[1 2 3;4 5 6;7 8 9;eye(3)]
26 % If I don't have a cell break above this comment, this
27 % text just appears as comments within the command lines,
28 % and not text on the webpage.
29 %%%
30 % Instead, have a cell break or a section break

```

LINK 5: <http://math.loyola.edu/~loberbro/matlab/html/publishMFile4.html>

Note that you can also have section titles without having cell breaks. This is done by having the line start with “%%” along with the section text. This will have the text and/or Matlab commands be within the sections, but the output of those commands at the next cell break, which may not have the desired effect.

publishMFile4.m (partial view)

```

33 %% Next section
34 % this section does not have a cell break. This may
35 % or may not be useful depending on how you want the
36 % output displayed on the published webpage. It works
37 % here because this is the last section and cell.
38 x=linspace(0.0001,10);
39 y=log(x);
40 plot(x,y)

```

LINK 5: <http://math.loyola.edu/~loberbro/matlab/html/publishMFile4.html>

Using cells is especially important for m-files with multiple plots. Remember, MATLAB only shows the last plotting command (like `plot`, `...`, `mesh`, `surf`, etc.). You can have multiple commands appear on the same figure by using the `hold on` and `hold off` commands. But if you want to display multiple figures (not in the same window), you have to either use the `pause` command of the `figure` command. This first example uses the `pause` command:

publishMFile5.m

```

1 %% Example of Basic Script with pause
2 % Lisa Oberbroeckling, Spring 2012
3 %%
4 x=linspace(-pi,pi);
5 y=sin(x);
6 plot(x,y)
7 hold on
8 y=cos(x);
9 plot(x,y,'r')
10 hold off
11 title('First Plot')
12 pause
13 [x,y]=meshgrid(linspace(-10,10));
14 z=sin(x).*cos(y);
15 mesh(x,y,z)
16 xlabel('x'),ylabel('y'),zlabel('z')
17 title('Second Plot')

```

LINK 6: <http://math.loyola.edu/~loberbro/matlab/html/publishMFile5.html>

If you run the file `publishMFile5.m`, the first figure will appear and then Matlab will be paused. The second will appear after any key is pressed. When this is published, even the publishing will be on pause after the first figure is created until you press a key. But if you look at the webpage, only

the last figure is actually shown on the webpage. As discussed above, when publishing, Matlab runs each cell as a block and then displays any output. At the end of the cell the only output that Matlab sees as being created by the cell of commands is the second figure. The second figure replaces the first figure, so it is not shown on the webpage. Thus, we need to create a cell for each figure we want on the webpage. When we publish the m-file, we have to remember to “press any key” for the publishing can continue, which is really annoying so you may want to take the `pause` command out or comment it out. In the following example we created a new cell without a section title.

publishMFile5b.m

```
1 %% Example of Basic Script with pause
2 % Lisa Oberbroeckling, Spring 2012
3 %%
4 close all
5 clc
6 % first plot
7 x=linspace(-pi,pi);
8 y=sin(x);
9 plot(x,y)
10 hold on
11 y=cos(x);
12 plot(x,y,'r')
13 hold off
14 title('First Plot')
15 % pause
16 %%
17 [x,y]=meshgrid(linspace(-10,10));
18 z=sin(x).*cos(y);
19 mesh(x,y,z)
20 xlabel('x'),ylabel('y'),zlabel('z')
21 title('Second Plot')
```

LINK 7: <http://math.loyola.edu/~loberbro/matlab/html/publishMFile5b.html>

The next group of files use the `figure` command.

publishMFile6.m

```
1 %% Example of Basic Script with figure
2 % Lisa Oberbroeckling, Spring 2012
3 %%
4 close all
```



```

5 clc
6 % first plot
7 figure(1)
8 x=linspace(-pi,pi);
9 y=sin(x);
10 hold on
11 y=cos(x);
12 plot(x,y,'r')
13 hold off
14 plot(x,y,'r')
15 title('First Plot')
16 % Second plot
17 figure(2)
18 [x,y]=meshgrid(linspace(-10,10));
19 z=sin(x).*cos(y);
20 mesh(x,y,z)
21 xlabel('x'),ylabel('y'),zlabel('z')
22 title('Second Plot')
23 % third plot
24 figure(3)
25 [x,y]=meshgrid(linspace(-10,10));
26 z=x.*cos(y);
27 mesh(x,y,z)
28 xlabel('x'),ylabel('y'),zlabel('z')
29 title('Third Plot')
30 % fourth plot
31 figure(4)
32 [x,y]=meshgrid(linspace(-10,10));
33 z=x.*y;
34 mesh(x,y,z)
35 xlabel('x'),ylabel('y'),zlabel('z')
36 title('Fourth Plot')

```

LINK 8: <http://math.loyola.edu/~loberbro/matlab/html/publishMFile6.html>

When the above file is published, it puts each figure side-by-side on one line. Depending on how many figures you have this may not have the desired effect, so you may want to have each figure within its own cell.

publishMFile6b.m

```

1 %% Example of Basic Script with figure
2 % Lisa Oberbroeckling, Spring 2012
3 %%
4 close all

```

```

5  clc
6  %% first plot
7  figure(1)
8  x=linspace(-pi,pi);
9  y=sin(x);
10 hold on
11 y=cos(x);
12 plot(x,y,'r')
13 hold off
14 plot(x,y,'r')
15 title('First Plot')
16 %% Second plot
17 figure(2)
18 [x,y]=meshgrid(linspace(-10,10));
19 z=sin(x).*cos(y);
20 mesh(x,y,z)
21 xlabel('x'),ylabel('y'),zlabel('z')
22 title('Second Plot')
23 %% third plot
24 figure(3)
25 [x,y]=meshgrid(linspace(-10,10));
26 z=x.*cos(y);
27 mesh(x,y,z)
28 xlabel('x'),ylabel('y'),zlabel('z')
29 title('Third Plot')
30 %% fourth plot
31 figure(4)
32 [x,y]=meshgrid(linspace(-10,10));
33 z=x.*y;
34 mesh(x,y,z)
35 xlabel('x'),ylabel('y'),zlabel('z')
36 title('Fourth Plot')

```

LINK 9: <http://math.loyola.edu/~loberbro/matlab/html/publishMFile6b.html>

4.2 Cells for running/debugging files

This section is still in progress.

5 Formatting text

You can format your m-file by clicking on the “Cell” menu item within the Editor Window. By clicking on one of the items on the Cell menu, Matlab

will insert text into your m-file for that purpose. This includes inserting text for things already discussed above, like inserting a cell break, title, etc. There are also buttons on the toolbar within the editor window for some of these items. You also have the ability to customize your toolbar to add others.

Going to the menu and/or using the buttons on the toolbar can take extra time after awhile, so it is also useful to know how to just type in the formatting. Note that these are presented not in the order of the menu.

5.1 Basic text formatting

In order to create a new line or new paragraph of text, have a blank comment line in between the lines.

```
1 % Here is a paragraph that just goes on, and on, and on,  
2 % and on, and on,  
3 % and on, and on, and on, and on, and on, and on. Still ...  
   going on,  
4 % and on, and on, and on, and on, and on.  
5 %  
6 % Then we have a new line/paragraph.
```

Text can be formatted to be **bold**, *italic*, **monospaced**, or combinations like ***bold, italic***.

```
1 % Text can be *bold*, _italic_, and/or |monospaced|. One can  
2 % also combine these formats like:  
3 %  
4 % _*BOLD, ITALIC TEXT*_  
5 %  
6 % |_ITALIC, MONOSPACED TEXT_|
```

5.2 Lists

One can have an unordered, or bulleted list.

```
1 %% Unordered (Bulleted) List  
2 %%  
3 % * first item  
4 % * second item blah blah
```

```
5 %
```

Keep in mind that you must have a cell break before the list, with or without a section title. You also must have a blank comment line to end the list. Here's another example.

```
1 %%  
2 %  
3 % * item number 1  
4 % * item number 2  
5 %
```

You can also have an ordered (numbered list) using the same formatting as above, but with “#” instead of “*” for each list item.

```
1 %% Ordered (Numbered) List  
2 %  
3 % # first item  
4 % # second item blah blah  
5 %
```

As in the bulleted list, one must have a cell break, with or without a section title. Second example without section title:

```
1 %%  
2 %  
3 % # blah blah blah blah blah blah blah blah blah blah ...  
    blah blah blah blah blah blah blah blah  
4 % # yadda yadda yadda yadda yadda yadda yadda yadda yadda ...  
    yadda yadda yadda yadda yadda yadda yadda yadda yadda ...  
    yadda yadda  
5 %
```

5.3 HTML Links

You can have the links display the URL or display other text.

URL as the link:

```
1 % Here's an example of using the URL as
```

```
2 % the text of the link: <http://www.mathworks.com>
```

Other text as the link:

```
1 % Other text as the link: <http://www.mathworks.com MathWorks>
```

LINKING THE M-FILE: It may be nice (required!) to link the M-File that was published to create the webpage. The easiest way to do it is like it is done below rather than using the entire URL.

```
1 % *LINKING THE M-FILE* It may be nice (required!) to link ...  
   the M-File that  
2 % was published to create the webpage. The easiest way to ...  
   do it is  
3 % like this <../publishMFile7.m M-File for this page>.
```

The “../” before the file name means to go back one folder from where the HTML file is located, which is where the M-file is located.

IMPORTANT: the text within the “<” and “>” must be on one line within the file. The editor might automatically wrap the text if the URL and/or the text for the link is long. If this is the case, go back and make it one line.

5.4 Inserting Images

Any figures that the MATLAB code creates will automatically be saved as PNG files and inserted on the webpage. You can also include images like the following example.

```
1 %  
2 % <<working2hard.gif>>  
3 %
```

The image must be on its own line - no text can appear before or after it for the image to be shown correctly on the page. Also note that the above is assuming the file for the image is located in the “html” folder into which the HTML file the M-file produced is located.

5.5 Preformatted Text

Preformatted text is a way to display text on the webpage EXACTLY as it appears in the editor, including extra spaces, line breaks in exactly the same place, etc. This is commonly used for displaying lines of code in programming.

IMPORTANT: notice that the code below doesn't appear any different than plain comments. In order for it to be preformatted, you must use the menu item to insert the lines and change it. Here is what the menu item inserts:

```
1 %  
2 %  PREFORMATTED  
3 %  TEXT  
4 %
```

Then you change the words “PREFORMATTED” AND “TEXT” and add lines if necessary:

```
1 %  
2 %  preformatted text  
3 %  displayed  
4 %  exactly  
5 %  as it appears in the editor  
6 %  (commonly used to display lines of code)  
7 %  _____  
8 %  function y = myexample(x)  
9 %  % MYEXAMPLE(X) is a function for example purposed only  
10 %  %  
11 %  y = x.^2;  
12 %  _____  
13 %
```

5.6 Inserting HTML Code

You can insert other HTML code (such as tables) into your document.

```
1 %  
2 % <html>
```

```

3 % <table border="1" cellspacing="0" ...
   cellpadding="3"><tr><td>one</td><td>two</td></tr></table>
4 % </html>
5 %

```

You may notice at the top of some M-files I insert the following.

```

1 %%
2 % <html>
3 % <style type="text/css">
4 % body { font-family: Helvetica, Arial, sans-serif;}
5 % </style>
6 % </html>

```

This is just to change the font of the entire webpage to a sans serif font, rather than the default font (like “Times New Roman”).

5.7 Inserting \LaTeX equations

Basic \LaTeX equations can be displayed on the webpage. Only basic mathematics can be displayed when publishing to HTML mode, and some symbols don’t display correctly. For example, the not equals “ \neq ” symbol doesn’t work (hopefully will in later versions). Technically speaking, when these equations are published to an HTML file, the equation is saved as a PNG image (in the “html” folder or same folder the HTML file is saved) and that image file is displayed on the webpage.

```

1 % One can insert basic LaTeX commands and equations. One ...
   can have LaTeX
2 % code with the paragraph like
3 % $$ e^{\pi i} + 1 = 0 $$
4 % or on a separate line.
5 %
6 % $$ y_1 = \frac{d}{dx} \int_0^{\pi x} e^{\theta^2} d\theta $$
7 %
8 % Notice that the "not equals" symbol since doesn't work:
9 % $$ 0 \neq 1 $$

```

5.8 M-file and HTML file for this section

The m-file that includes examples of all of this text formatting can at LINK 10: <http://math.loyola.edu/~loberbro/matlab/html/publishMFile7.html>

6 Summary of M-Files/Links

All of the following M-files can be found at:

<http://math.loyola.edu/~loberbro/matlab/>.

The published webpages can be found at:

<http://math.loyola.edu/~loberbro/matlab/html/>.

1. `basicMFile.html`: example of a basic M-file or SCRIPT file.
2. `publishMFile1.html`: example of minimally changing `basicMFile.m` for publishing.
3. `publishMFile2.html`: example of putting in a title and introductory text.
4. `publishMFile3.html`: adding section title.
5. `publishMFile4.html`: examples of placing cell breaks for displaying output.
6. `publishMFile5.html`: basic example of plots and the `pause` command.
7. `publishMFile5b.html`: modifying output of plots using cell breaks.
8. `publishMFile6.html`: basic example of plots and the `figure` command.
9. `publishMFile6b.html`: modifying output of plots using cell breaks and `figure` command.
10. `publishMFile7.html`: examples of formatting text.