A guide to Matlab publish

Table of Contents

Introduction

Matlab offers a very nice way to present work, combining code, comments, figures, etc. This facility, named publish, is a neat way to prepare assignments for handing in for this course. A short discussion follows. We recommend that you look at both the .m file and the .html output to see which commands do what.

If you are reading the .m file, then simply type "publish('publish_guide.m')" in the Matlab command window. This command will publish this file and once done, the command window will display the path where the corresponding .html file has been generated. Copy this path and paste it in the url bar of your favorite internet browser (firefox for instance!) and see the published output.

If you are already reading this in .html format, we recommend that you open the .m file and publish it again to make sure that it works.

The use of %%

If we start a line with %% Matlab's publish would consider that as a signal to make a new "cell" for the code and comments that follow. Now there are two ways we can use %%. If we begin a line with %% and then write something afterwards on the same line, that something becomes the heading of a new chapter and this chapter heading is added to the table of contents shown in the bginning of the published output. The chapter we are reading right now was created by writing "%% The use of %%" in the first line of this chapter. To create a new chapter called "More on the use of %%", we say:

More on the use of %%

If on the other hand we only put %% in a line and then start our work from the next line with a single % in the beginning, this would generate a new paragraph/section. Here is an example:

Comments with %%

Let us declare a variable x:

```
x = 1;
```

This is an important comment about x and it written with a %% line in the beginning, hence this does not appear as part of the code.

Comments without %%

Let us declare a variable xx:

```
xx = linspace(-1,1, 101);
% This is an important comment about xx but written without a %% line
 in the
% beginning and hence it appears within the code block.
```

Results and figures

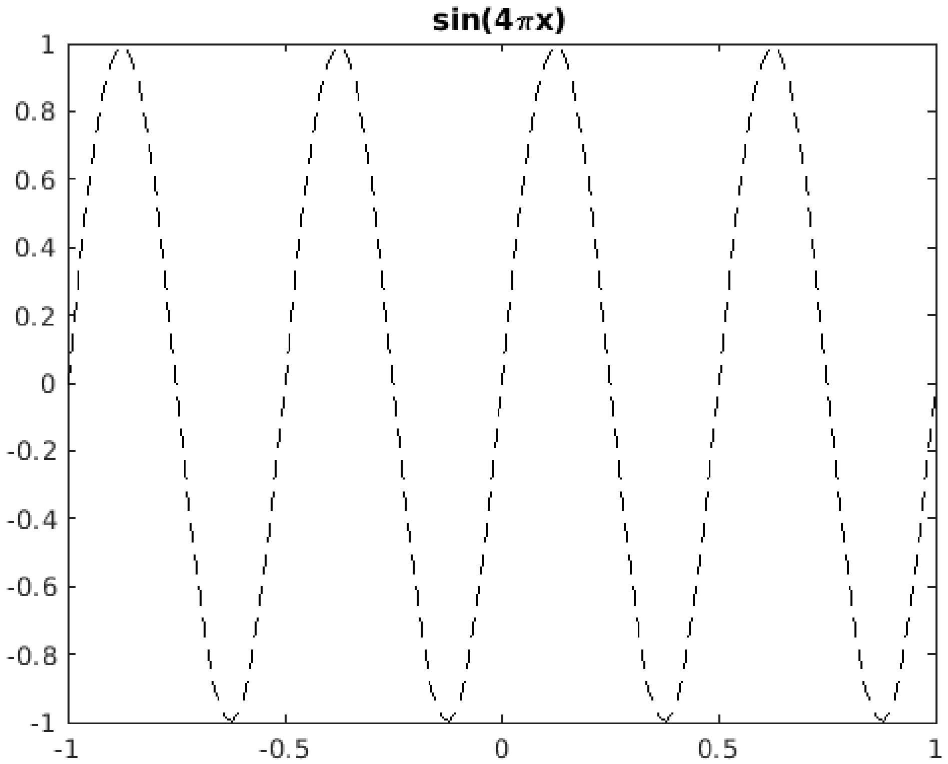
A great benefit of using publish is the automatic inclusion of figures and calculation results in your files. Below we find the value of the 20th entry of the vector xx:

```
xx20 = xx(20) % Semicolon omitted, so, result will be shown in the
 output.
```

```
xx20 =
   -0.6200
```

Figures get put in the correct place automatically:

```
plot(xx,sin(4*pi*xx),'--k'), title('sin(4\pix)')
```



Making Lists

To make a numbered list, insert a blank line with a % and then for each item in the list start a new line with "% #" in the beginning. Here is an example:

1. ITEM 1

2. ITEM 2

To make a bulleted list, use the same pattern but for each bullet use "% *" in the beginning of each line. For example:

- ITEM1

- ITEM2

Adding listing of functions

The type command allows you to see a listing of a function so you can use this within publish to add the text of a function you have written. An example using the Matlab function primes is:

```
type primes
```

```

function p = primes(n)
%PRIMES Generate list of prime numbers.
%   PRIMES(N) is a row vector of the prime numbers less than or
%   equal to N.  A prime number is one that has no factors other
%   than 1 and itself.
%
%   Class support for input N:
%      float: double, single
%      integer: uint8, int8, uint16, int16, uint32, int32, uint64,
 int64
%
%   See also FACTOR, ISPRIME.

%   Copyright 1984-2013 The MathWorks, Inc.

if ~isscalar(n)
  error(message('MATLAB:primes:InputNotScalar'));
elseif ~isreal(n)
  error(message('MATLAB:primes:ComplexInput'));
end
if n < 2
  p = zeros(1,0,class(n));
  return
elseif isfloat(n) && n > flintmax(class(n))
  warning(message('MATLAB:primes:NGreaterThanFlintmax'));
  n = flintmax(class(n));
end
n = floor(n);
p = true(1,double(ceil(n/2)));
q = length(p);
```

```
if (isa(n,'uint64') || isa(n,'int64')) && n > flintmax
  ub = 2.^(nextpow2(n)/2);  %avoid casting large (u)int64 to double
else
  ub = sqrt(double(n));
end
for k = 3:2:ub
  if p((k+1)/2)
     p(((k*k+1)/2):k:q) = false;
  end
end
p = cast(find(p)*2-1,class(n));
p(1) = 2;
```

How to publish in other formats

Finally, it's worth mentioning that it's possible to publish in other format than .html, such as .tex. For changing the format and other options, from the Matlab editor, go to File -> Publish configuration for ... -> Edit publish configuration for ... or simply type "help publish" to see what other options are available.

Published with MATLAB® R2020b