

exampleA.m : file for publish_mpl showing extra options

This file will be used to demonstrate the possibilities of the new function `publish_mpl`. The new function expands the possibilities of the standard `publish` function with regard to the LaTeX format. Motivation for creating this new function is that I want more control over the output than the pdf and html format can offer. So LaTeX is the obvious choice but at the same time I want to avoid manual editing of the tex file handle as much as possible. By using an adapted xsl file, the package `matlab-prettifier` created by Julien Cretel and using additional `publish` options we can achieve the following:

1. determine the documentclass and layout of the document
2. show MATLAB code (and also listings of mfiles) in a nice layout
3. specify hyperref options that determine the pdf attributes
4. determine how the header of the document is presented (titel, author, list of figures and listings)
5. include captions and references

Contents

- Acknowledgement
- Square Waves from Sine Waves
- Add an Odd Harmonic and Plot It
- Note About Gibbs Phenomenon
- Listing of this script
- Listing of `publish_mpl_examples.m`

Acknowledgement

This file is adapted from the `fourier_demo2.m` file that is included in MATLAB and can be copied in the current directory with

```
copyfile(fullfile(matlabroot,'help','techdoc',...  
'matlab_env','examples','fourier_demo2.m'),'.','f')
```

Square Waves from Sine Waves

The Fourier series expansion for a square-wave is made up of a sum of odd harmonics, as shown here by the plots in figure 1 on page 3 (1 harmonic), figure 2 on page 4 (5 harmonics) and figure 3 on page 5 (9 harmonics).

```
1 if exist('avalue','var')  
2     fprintf('print the value passed to this script: %f\n',avalue)  
3 else
```

```

4     fprintf('no value passed to this script\n')
5 end

```

print the value passed to this script: 2.000000

Add an Odd Harmonic and Plot It

```

1 t = 0:.1:pi*4;
2 k = 1 ;
3 y = sin(k*t)/k;
4 figure(k)
5 plot(t,y);
6 title(sprintf('MATLAB caption: plot when k=%.0f',k))

```

In each iteration of the for loop add an odd harmonic to y . As k increases, the output approximates a square wave with increasing accuracy.

Perform the following mathematical operation at each iteration:

$$y = y + \frac{\sin kt}{k}$$

Display some of the plots:

```

1 for k = 3:2:9
2     y = y + sin(k*t)/k;
3     if mod(k,4)==1
4         figure(k)
5         plot(t,y)
6         title(sprintf('MATLAB caption: plot when k=%.0f',k))
7     end
8 end

```

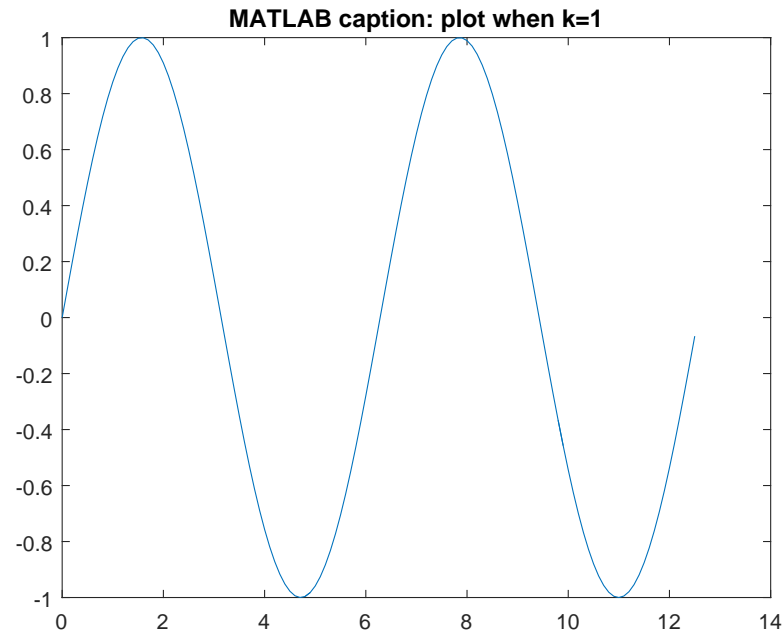


Figure 1: first harmonic

Note About Gibbs Phenomenon

Even though the approximations are constantly improving, they will never be exact because of the Gibbs phenomenon, or ringing.

Listing of this script

```

1 %% exampleA.m : file for publish_mpl showing extra options
2 % This file will be used to demonstrate the possibilities
3 % of the new function |publish_mpl|. The new function
4 % expands the possibilities of the standard |publish| function
5 % with regard to the LaTeX format. Motivation for creating
6 % this new function is that I want more control over the output than the
7 % pdf and html format can offer. So LaTeX is the obvious choice

```

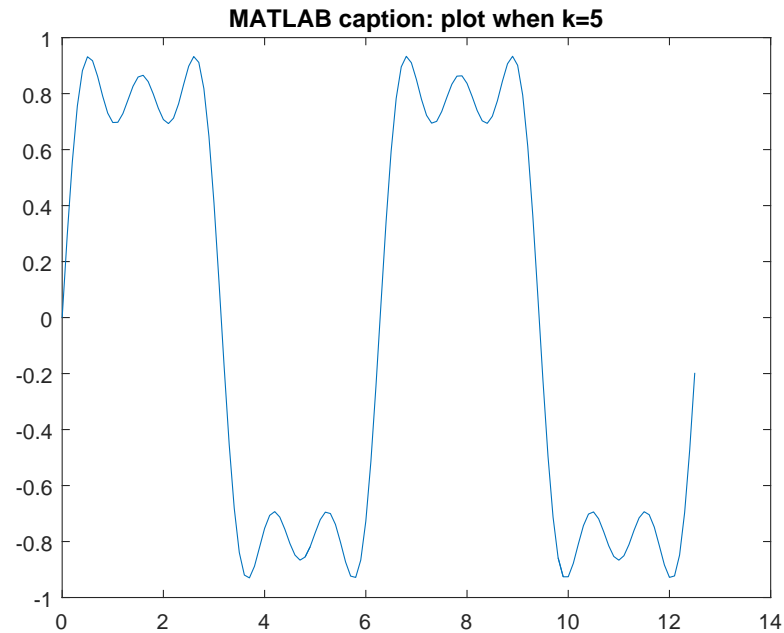


Figure 2: sum of first 5 harmonics

```

8 % but at the same time I want to avoid manual editing of the
9 % tex file handle as much as possible. By using an adapted
10 % xsl file, the package matlab-prettifier created by Julien Cretel
11 % and using additional |publish| options we can achieve the following:
12 %
13 % # determine the documentclass and layout of the document
14 % # show MATLAB code (and also listings of mfiles) in a nice layout
15 % # specify hyperref options that determine the pdf attributes
16 % # determine how the header of the document is presented (titel, author, list of figures and listings)
17 % # include captions and references
18 %% Acknowledgement
19 % This file is adapted from the |fourier_demo2.m| file
20 % that is included in MATLAB and can be copied in
21 % the current directory with

```

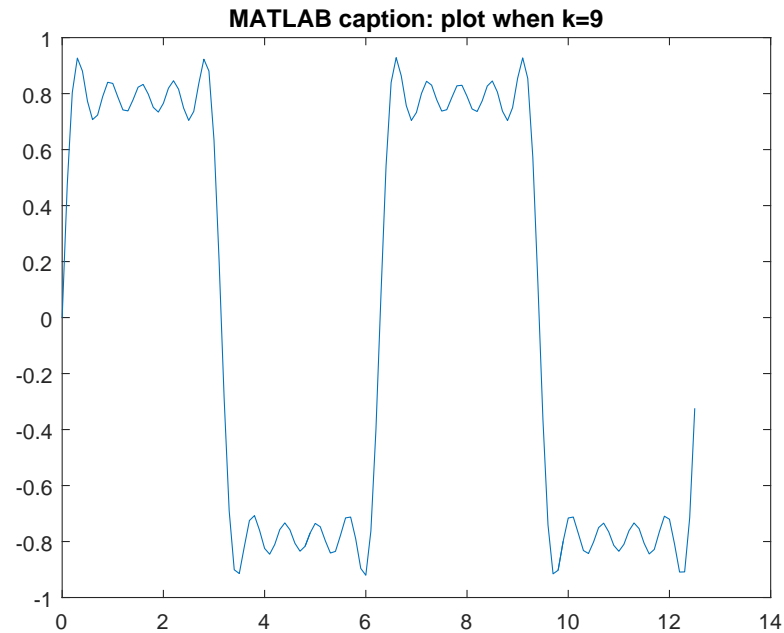


Figure 3: sum of first 9 harmonics

```

22 %%
23 %
24 %   copyfile(fullfile(matlabroot,'help','techdoc',...
25 %   'matlab_env','examples','fourier_demo2.m'),'.','f')
26 %%
27 %% Square Waves from Sine Waves
28 % <latex>
29 % % The actual function to publish starts now.
30 % % This text block is changed to a latex block to show the caption and reference capabilities
31 % %
32 % % the following statements insert the references to the plots:
33 % The Fourier series expansion for a square-wave is
34 % made up of a sum of odd harmonics, as shown here
35 % by the plots in figure \ref{exampleA_01.eps} on page \pageref{exampleA_01.eps} (1 harmonic),

```

```

36 % figure \ref{exampleA_02.eps} on page \pageref{exampleA_02.eps} (5 harmonics) and
37 % figure \ref{exampleA_03.eps} on page \pageref{exampleA_03.eps} (9 harmonics).
38 % %
39 % % the following statements define the captions of the plots:
40 % \global\def\captionA{first harmonic}
41 % \global\def\captionB{sum of first 5 harmonics}
42 % \global\def\captionC{sum of first 9 harmonics}
43 % </latex>
44 if exist('avalue','var')
45     fprintf('print the value passed to this script: %f\n',avalue)
46 else
47     fprintf('no value passed to this script\n')
48 end
49 %% Add an Odd Harmonic and Plot It
50 t = 0:.1:pi*4;
51 k = 1 ;
52 y = sin(k*t)/k;
53 figure(k)
54 plot(t,y);
55 title(sprintf('MATLAB caption: plot when k=%.0f',k))
56
57 %%
58 % In each iteration of the for loop add an odd
59 % harmonic to y. As _k_ increases, the output
60 % approximates a square wave with increasing accuracy.
61 %
62 % Perform the following mathematical operation
63 % at each iteration:
64 %
65 % $$ y = y + \frac{\sin kt}{k} $$
66 %
67 % Display some of the plots:
68 %
69
70 for k = 3:2:9
71     y = y + sin(k*t)/k;
72     if mod(k,4)==1

```

```

73     figure(k)
74     plot(t,y)
75     title(sprintf('MATLAB caption: plot when k=%.0f',k))
76 end
77 end
78
79 %% Note About Gibbs Phenomenon
80 % Even though the approximations are constantly
81 % improving, they will never be exact because of the
82 % Gibbs phenomenon, or ringing.
83 %% Listing of this script
84 % <latex>
85 % % assuming m-file in directory one level higher than tex dir (using the standard html subdirectory)
86 % % assuming numbers and framed are not set in \usepackage and they are wanted
87 % % \lstinputlisting[frame=single,numbers=left]{../exampleA.m}
88 % % assuming numbers and framed are set in \usepackage and they are not wanted
89 % % \lstinputlisting[frame=none,numbers=none]{../exampleA.m}
90 % % assuming numbers and framed are set in \usepackage are set and wanted
91 % \lstinputlisting{../exampleA.m}
92 % </latex>
93 %% Listing of publish_mpl_examples.m
94 % <latex>
95 % \lstinputlisting{../publish_mpl_examples.m}
96 % </latex>

```

Listing of publish_mpl_examples.m

```

1  addpath('../code')
2  %% example1: -> pdf
3  % Use the function to create pdf-file.
4  % This is the same as using the publish user interface.
5  mycode = { ...                                % example of code to execute (two lines)
6              'avalue = 2;'    ...
7              'exampleA'      ...
8          } ;
9  pstruct = struct( ...                        % publish options

```

```

10     'format' , 'pdf' , ...           % output format
11     'call' , {mycode} , ...         % code to execute (defined above)
12     'newname' , 'exampleA1.pdf' );   % new name of output file
13 newname = publish_mpl('exampleA', pstruct) ; % produce the output file (pdf)
14
15 %% example2: -> latex
16 % Use the function to create tex-file
17 % with as much as possible the same layout
18 % as the original tex file but with references, captions
19 % and listings
20 mycode = { ...                      % example of code to execute (one line)
21     'exampleA' ...
22 } ;
23 pstruct = struct( ...               % publish options
24     'format' , 'latex' , ...         % output format latex using the new xsl file
25     'call' , {mycode} , ...         % code to execute (defined above)
26     'orientation' , 'portrait' , ... % overwrite orientation (default 'landscape')
27     'newname' , 'exampleA2.tex' , ... % new name of output file
28     'prettifier_options' , '' ) ;    % overwrite prettify options (default 'framed,numbered')
29
30 newname = publish_mpl('exampleA', pstruct) ; % produce the output file (tex)
31
32 %% example3: -> latex
33 % Same as example2 but the layout is landscape and
34 % the MATLAB code will be in frames with numbers.
35 mycode = { ...                      % example of code to execute (one line)
36     'exampleA' ...
37 } ;
38 pstruct = struct( ...               % publish options
39     'format' , 'latex' , ...         % output format latex using the new xsl file
40     'call' , {mycode} , ...         % code to execute (defined above)
41     'newname' , 'exampleA3.tex');    % new name of output file
42 % 'orientation' , 'landscape' , ... % use default orientation ('landscape')
43 % 'prettifier_options' , 'framed,numbered' , ... % use default prettify options ('framed,numbered')
44
45 newname = publish_mpl('exampleA', pstruct) ; % produce the output file (tex)

```



```

46 %% example4: -> latex
47 % same as example3 but listings have their own
48 % caption in exampleB and they are listed by
49 % setting 'makelstlistoflistings' to true
50 mycode = { ...                                % example of code to execute (one line)
51     'exampleB' ...
52 } ;
53 pstruct = struct( ...                          % publish options
54     'format' , 'latex' , ...                    % output format latex using the new xsl file
55     'call' , {mycode} , ...                     % code to execute (defined above)
56     'newname' , 'exampleB1.tex' , ...            % new name of output file
57     'pdfauthor' , 'han@hanoostdijk.nl' , ...    % insert a pdf option
58     'makelstlistoflistings' , true);             % create lstlistoflistings
59 newname = publish_mpl('exampleB', pstruct);     % produce the output file (tex)
60
61 %% example5: -> latex
62 % same as example4 but now with a regular LaTeX contents
63 % by setting 'maketableofcontents' to true
64 mycode = { ...                                % example of code to execute (one line)
65     'exampleB' ...
66 } ;
67 pstruct = struct( ...                          % publish options
68     'format' , 'latex' , ...                    % output format latex using the new xsl file
69     'call' , {mycode} , ...                     % code to execute (defined above)
70     'newname' , 'exampleB2.tex' , ...            % new name of output file
71     'pdfauthor' , 'han@hanoostdijk.nl' , ...    % insert a pdf option
72     'maketableofcontents' , true , ...          % create tableofcontents
73     'makelstlistoflistings' , true);            % create lstlistoflistings
74 newname = publish_mpl('exampleB', pstruct);     % produce the output file (tex)
75
76 %% example6: -> xml
77 % same as example3 but now to xml format
78 mycode = { ...                                % example of code to execute (one line)
79     'exampleB' ...
80 } ;
81 pstruct = struct( ...                          % publish options
82     'format' , 'xml' , ...                      % output format latex using the new xsl file

```

```
83     'call' , {mycode} , ...           % code to execute (defined above)
84     'newname' , 'exampleB3.xml' , ... % new name of output file
85     'pdfauthor', 'han@hanoostdijk.nl' , ... % insert a pdf option
86     'makelstlistoflistings', true);    % create lstlistoflistings
87 newname = publish_mpl('exampleB', pstruct) ; % produce the output file (tex)
```