

## Contents

Acknowledgement . . . . .	1
Square Waves from Sine Waves . . . . .	1
Add an Odd Harmonic and Plot It . . . . .	2
Note About Gibbs Phenomenon . . . . .	3
Listings . . . . .	3

## Listings

1 listing of exampleB script . . . . .	3
2 listing of publish_mpl_examples script . . . . .	7

## exampleB.m : file for publish\_mpl showing extra options

As exampleA.m with the only difference that the listings get their own caption (and therefore the section headers are omitted). The listings are presented in a 'lstlistoflistings' by specifying this in the `publish` options (see example4) in `publish_mapl_examples`.

## 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 2 (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
```

no value passed to this script

## 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))
```

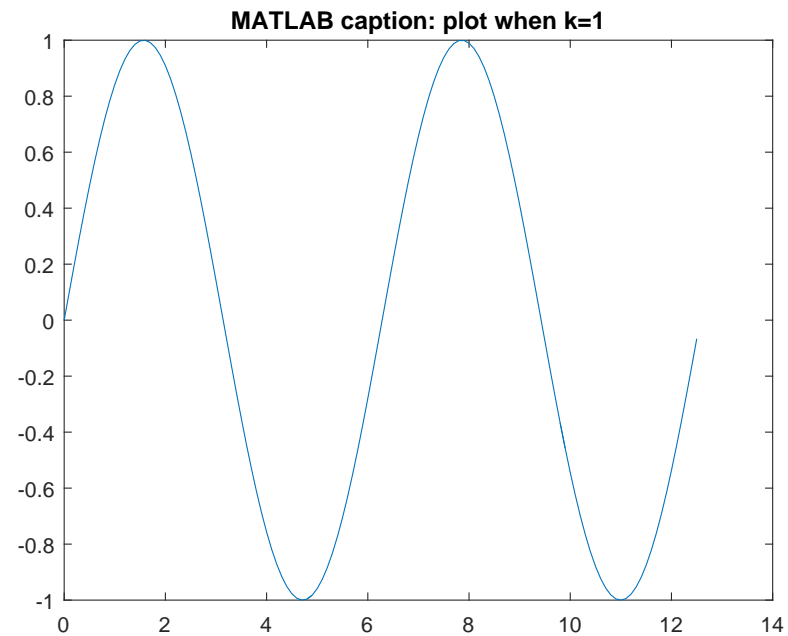


Figure 1: first harmonic

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
```

## Note About Gibbs Phenomenon

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

## Listings

Listing 1: listing of exampleB script

```
1 %% exampleB.m : file for publish_mpl showing extra options
2 % As exampleA.m with the only difference that the listings
3 % get their own caption (and therefore the section headers
4 % are omitted). The listings are presented in a 'lstlistoflistings'
5 % by specifying this in the |publish| options (see example4) in
6 % publish_mapl_examples.
7 %% Acknowledgement
8 % This file is adapted from the |fourier_demo2.m| file
9 % that is included in MATLAB and can be copied in
10 % the current directory with
```

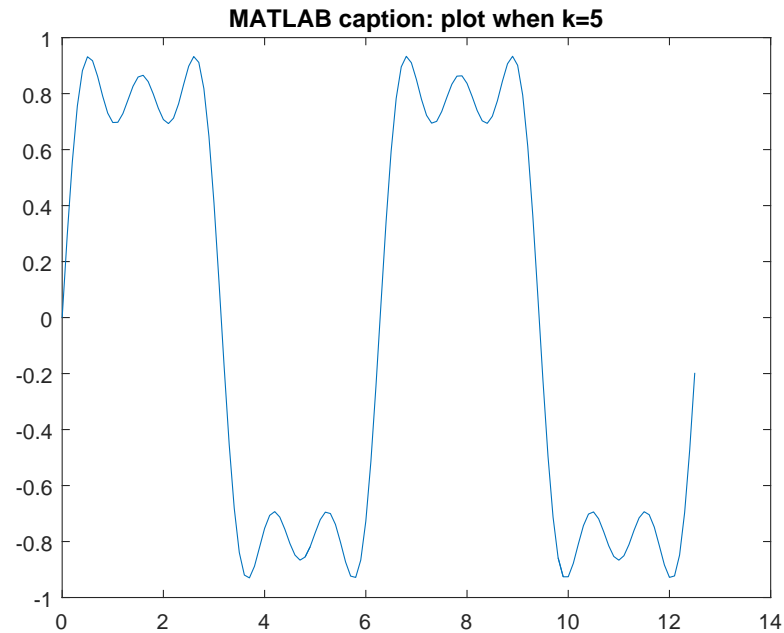


Figure 2: sum of first 5 harmonics

```

11 %%
12 %
13 %   copyfile(fullfile(matlabroot,'help','techdoc',...
14 %   'matlab_env','examples','fourier_demo2.m'),'.','f')
15 %% Square Waves from Sine Waves
16 % <latex>
17 % % The actual function to publish starts now
18 % % This text block is changed to a latex block to show the caption and reference capabilities
19 % %
20 % % the following statements insert the references to the plots:
21 % The Fourier series expansion for a square-wave is
22 % made up of a sum of odd harmonics, as shown here
23 % by the plots in figure \ref{exampleB_01.eps} on page \pageref{exampleB_01.eps} (1 harmonic),
24 % figure \ref{exampleB_02.eps} on page \pageref{exampleB_02.eps} (5 harmonics) and

```

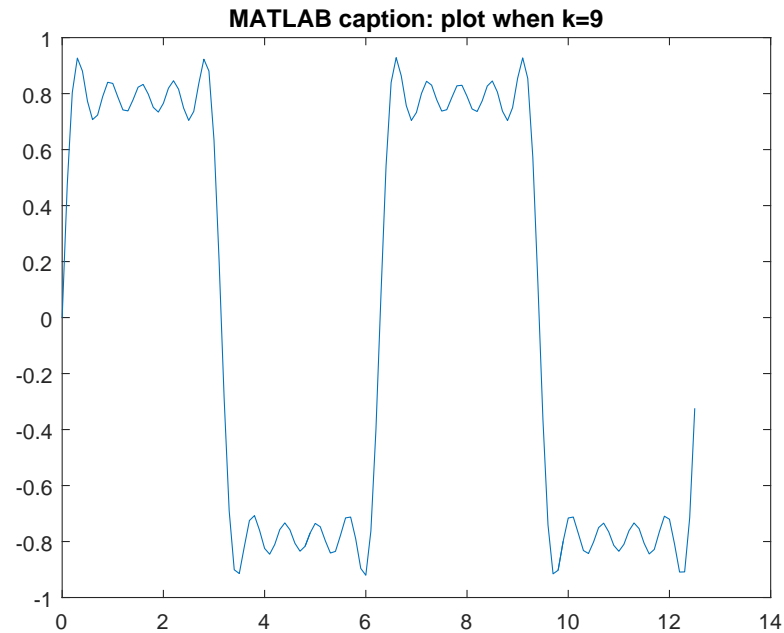


Figure 3: sum of first 9 harmonics

```

25 % figure \ref{exampleB_03.eps} on page \pageref{exampleB_03.eps} (9 harmonics).
26 % %
27 % % the following statements define the captions of the plots:
28 % \global\def\captionA{first harmonic}
29 % \global\def\captionB{sum of first 5 harmonics}
30 % \global\def\captionC{sum of first 9 harmonics}
31 % </latex>
32 if exist('avalue','var')
33     fprintf('print the value passed to this script: %f\n',avalue)
34 else
35     fprintf('no value passed to this script\n')
36 end
37 %% Add an Odd Harmonic and Plot It
38 t = 0:.1:pi*4;

```

```

39 k    = 1 ;
40 y    = sin(k*t)/k;
41 figure(k)
42 plot(t,y);
43 title(sprintf('MATLAB caption: plot when k=%.0f',k))
44
45 %%
46 % In each iteration of the for loop add an odd
47 % harmonic to y. As _k_ increases, the output
48 % approximates a square wave with increasing accuracy.
49 %
50 % Perform the following mathematical operation
51 % at each iteration:
52 %
53 % $$ y = y + \frac{\sin kt}{k} $$
54 %
55 % Display some of the plots:
56 %
57
58 for k = 3:2:9
59     y = y + sin(k*t)/k;
60     if mod(k,4)==1
61         figure(k)
62         plot(t,y)
63         title(sprintf('MATLAB caption: plot when k=%.0f',k))
64     end
65 end
66
67 %% Note About Gibbs Phenomenon
68 % Even though the approximations are constantly
69 % improving, they will never be exact because of the
70 % Gibbs phenomenon, or ringing.
71
72 %% Listings
73 % <latex>
74 % % assuming m-file in directory one level higher than tex dir (using the standard html subdirectory)
75 % % assuming numbers and framed are not set in \usepackage and they are wanted

```

```

76 % % \lstinputlisting[frame=single,numbers=left]{../exampleB.m}
77 % % assuming numbers and framed are set in \usepackage and they are not wanted
78 % % \lstinputlisting[frame=none,numbers=none]{../exampleB.m}
79 % % assuming numbers and framed are set in \usepackage are set and wanted
80 % \lstinputlisting[caption=listing of exampleB script]{../exampleB.m}
81 % </latex>
82 %%
83 % <latex>
84 % \lstinputlisting[caption=listing of publish\_mpl\_examples script]{../publish_mpl_examples.m}
85 % </latex>

```

Listing 2: listing of publish\_mpl\_examples script

```

1 %% example1: -> pdf
2 % Use the function to create pdf-file.
3 % This is the same as using the publish user interface.
4 mycode = { ... % example of code to execute (two lines)
5     'avalue = 2;' ...
6     'exampleA' ...
7 } ;
8 pstruct = struct( ... % publish options
9     'format' , 'pdf' , ... % output format
10    'call' , {mycode} , ... % code to execute (defined above)
11    'newname' , 'exampleA1.pdf' ); % new name of output file
12 newname = publish_mpl('exampleA', pstruct); % produce the output file (pdf)
13
14 %% example2: -> latex
15 % Use the function to create tex-file
16 % with as much as possible the same layout
17 % as the original tex file but with references, captions
18 % and listings
19 mycode = { ... % example of code to execute (one line)
20     'exampleA' ...
21 } ;
22 pstruct = struct( ... % publish options
23     'format' , 'latex' , ... % output format latex using the new xsl file
24     'call' , {mycode} , ... % code to execute (defined above)

```

```

25     'orientation', 'portrait', ...
26     'newname' , 'exampleA2.tex' , ...
27     'prettifier_options' , '' );
28     ')
29 newname = publish_mpl('exampleA', pstruct) ;
30
31 %% example3: -> latex
32 % Same as example2 but the layout is landscape and
33 % the MATLAB code will be in frames with numbers.
34 mycode = { ...
35     'exampleA' ...
36     } ;
37 pstruct = struct( ...
38     'format' , 'latex' , ...
39     'call' , {mycode} , ...
40     'newname' , 'exampleA3.tex');
41 % 'orientation', 'landscape', ...
42 % 'prettifier_options', 'framed,numbered' , ...
43
44 newname = publish_mpl('exampleA', pstruct) ;
45 %% example4: -> latex
46 % same as example3 but listings have their own
47 % caption in exampleB and they are listed by
48 % setting 'makelstlistoflistings' to true
49 mycode = { ...
50     'exampleB' ...
51     } ;
52 pstruct = struct( ...
53     'format' , 'latex' , ...
54     'call' , {mycode} , ...
55     'newname' , 'exampleB1.tex' , ...
56     'pdfauthor', 'han@hanoostdijk.nl' , ...
57     'maketableofcontents', true , ...
58     'makelstlistoflistings', true);
59 newname = publish_mpl2('exampleB', pstruct) ;
60
% overwrite orientation (default 'landscape')
% new name of output file
% overwrite prettify options (default 'framed,numbered')

% produce the output file (tex)

% example of code to execute (one line)

% publish options
% output format latex using the new xsl file
% code to execute (defined above)
% new name of output file
% use default orientation ('landscape')
% use default prettify options ('framed,numbered')

% produce the output file (tex)

% example of code to execute (one line)

% publish options
% output format latex using the new xsl file
% code to execute (defined above)
% new name of output file
% insert a pdf option
% create tableofcontents
% create lstlistoflistings
% produce the output file (tex)

```



```

61 %% example5: -> latex
62 % same as example3 but listings have their own
63 % caption in exampleB and they are listed by
64 % setting 'makelstlistoflistings' to true
65 mycode = { ...                                % example of code to execute (one line)
66     'exampleB' ...
67 } ;
68 pstruct = struct( ...                          % publish options
69     'format' , 'xml' , ...
70     'call' , {mycode} , ...
71     'newname' , 'exampleB21.xml' , ...
72     'pdfauthor' , 'han@hanoostdijk.nl' , ...
73     'makelstlistoflistings' , true);
74 newname = publish_mpl('exampleB', pstruct) ;
% output format latex using the new xsl file
% code to execute (defined above)
% new name of output file
% insert a pdf option
% create lstlistoflistings
% produce the output file (tex)

```