Contents

Listings

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).

```
if exist('avalue','var')
    fprintf('print the value passed to this script: %f\n',avalue)
else
    fprintf('no value passed to this script\n')
end
```

```
no value passed to this script
```

Add an Odd Harmonic and Plot It

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

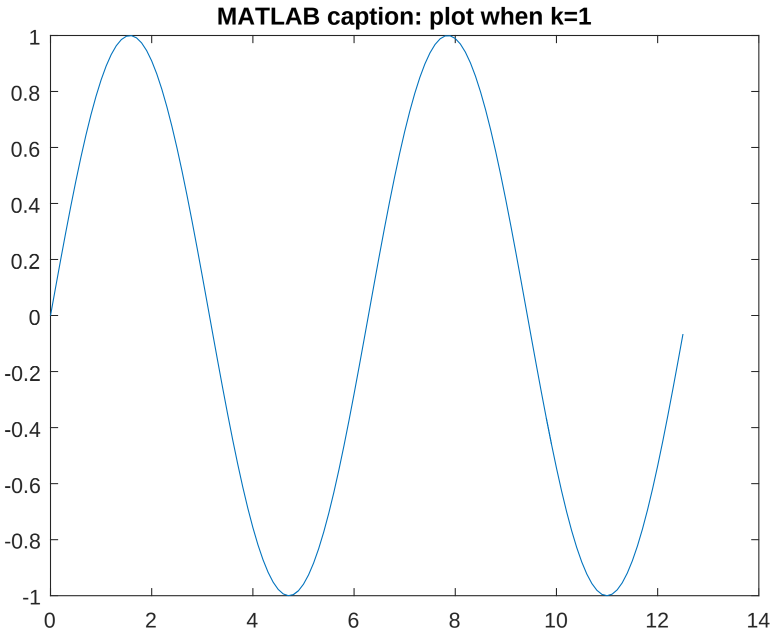

Figure 1: first harmonic

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:

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

```
%% 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
```

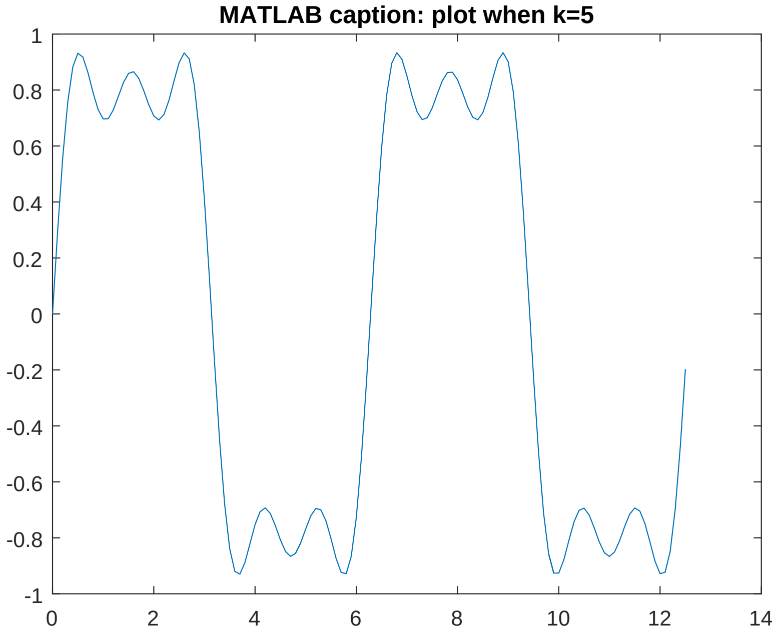

Figure 2: sum of first 5 harmonics

```
%%
%
%  copyfile(fullfile(matlabroot,'help','techdoc',...
%  'matlab_env','examples','fourier_demo2.m'),'.','f')
%% Square Waves from Sine Waves
% <latex>
% % The actual function to publish starts now
% % This text block is changed to a latex block to show the caption and reference capabilities
% %
% % the following statements insert the references to the plots:
% 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 \ref{exampleB_01.eps} on page \pageref{exampleB_01.eps} (1 harmonic),
% figure \ref{exampleB_02.eps} on page \pageref{exampleB_02.eps} (5 harmonics) and
```

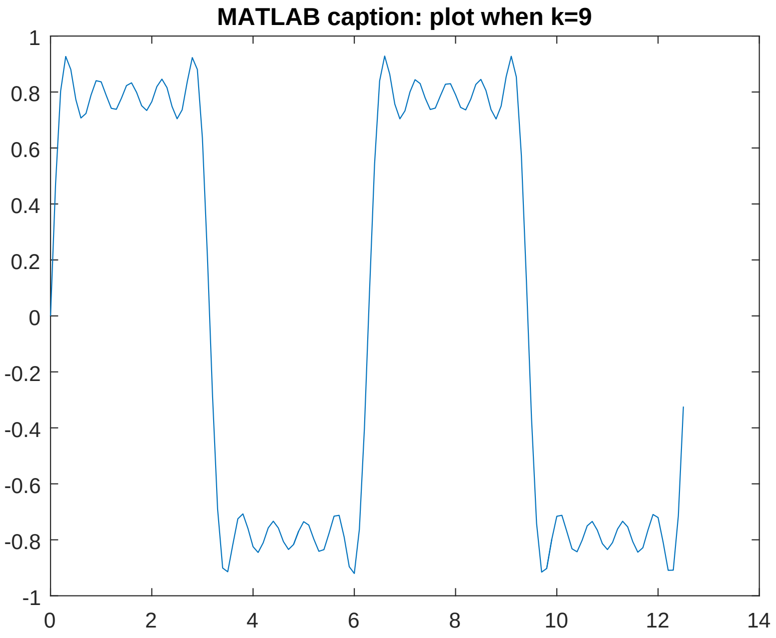

Figure 3: sum of first 9 harmonics

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

```
k    = 1 ;
y    = sin(k*t)/k;
figure(k)
plot(t,y);
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:
%

for k = 3:2:9
    y = y + sin(k*t)/k;
    if mod(k,4)==1
        figure(k)
        plot(t,y)
        title(sprintf('MATLAB caption: plot when k=%.0f',k))
    end
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
% <latex>
% % assuming m-file in directory one level higher than tex dir (using the standard html subdirectory)
% % assuming numbers and framed are not set in \usepackage and they are wanted
% % \lstinputlisting[frame=single,numbers=left]{../exampleB.m}
% % assuming numbers and framed are set in \usepackage and they are not wanted
% % \lstinputlisting[frame=none,numbers=none]{../exampleB.m}
% % assuming numbers and framed are set in \usepackage are set and wanted
% \lstinputlisting[caption=listing of exampleB script]{../exampleB.m}
% </latex>
%%
% <latex>
% \lstinputlisting[caption=listing of publish\_mpl\_examples script]{../publish_mpl_examples.m}
% </latex>
```

Listing 2: listing of publish_mpl_examples script

```
%% example1: -> pdf
% Use the function to create pdf-file.
% This is the same as using the publish user interface.
mycode = { ...                                          % example of code to execute (two lines)
                'avalue = 2;'   ...
            'exampleA'   ...
            } ;
pstruct = struct( ...                                   % publish options
    'format' , 'pdf' , ...                              % output format
    'call' , {mycode} , ...                             % code to execute (defined above)
    'newname' , 'exampleA1.pdf' );                       % new name of output file
newname = publish_mpl('exampleA', pstruct) ;             % produce the output file (pdf)

%% example2: -> latex
% Use the function to create tex-file
% with as much as possible the same layout
% as the original tex file but with references, captions
% and listings
mycode = { ...                                          % example of code to execute (one line)
            'exampleA'   ...
            } ;
pstruct = struct( ...                                   % publish options
    'format' , 'latex' , ...                            % output format latex using the new xsl file
    'call' , {mycode} , ...                             % code to execute (defined above)
    'orientation', 'portrait', ...                     % overwrite orientation (default 'landscape')
    'newname' , 'exampleA2.tex' , ...                  % new name of output file
    'prettifier_options' , '' ) ;                       % overwrite prettify options (default 'framed,numbered')

newname = publish_mpl('exampleA', pstruct) ;            % produce the output file (tex)

%% example3: -> latex
% Same as example2 but the layout is landscape and
% the MATLAB code will be in frames with numbers.
mycode = { ...                                         % example of code to execute (one line)
            'exampleA'   ...
            } ;
pstruct = struct( ...                                  % publish options
    'format' , 'latex' , ...                           % output format latex using the new xsl file
    'call' , {mycode} , ...                            % code to execute (defined above)
    'newname' , 'exampleA3.tex');                       % new name of output file
 %  'orientation', 'landscape', ...                     % use default orientation ('landscape')
 %  'prettifier_options', 'framed,numbered' , ...       % use default prettify options ('framed,numbered')

newname = publish_mpl('exampleA', pstruct) ;            % produce the output file (tex)
%% example4: -> latex
% same as example3 but listings have their own
% caption in exampleB and they are listed by
% setting 'makelstlistoflistings' to true
mycode = { ...                                         % example of code to execute (one line)
            'exampleB'   ...
            } ;
pstruct = struct( ...                                  % publish options
    'format' , 'latex' , ...                           % output format latex using the new xsl file
    'call' , {mycode} , ...                            % code to execute (defined above)
    'newname' , 'exampleB1.tex' , ...                  % new name of output file
    'pdfauthor', 'han@hanoostdijk.nl' , ...               % insert a pdf option
    'maketableofcontents', true , ...                  % create tableofcontents
    'makelstlistoflistings', true);                     % create lstlistoflistings
newname = publish_mpl2('exampleB', pstruct) ;                   % produce the output file (tex)

```

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