MATLAB Autumn Coursework 2018-19

Table of Contents

Question 1

Write a brief summary of the program here.

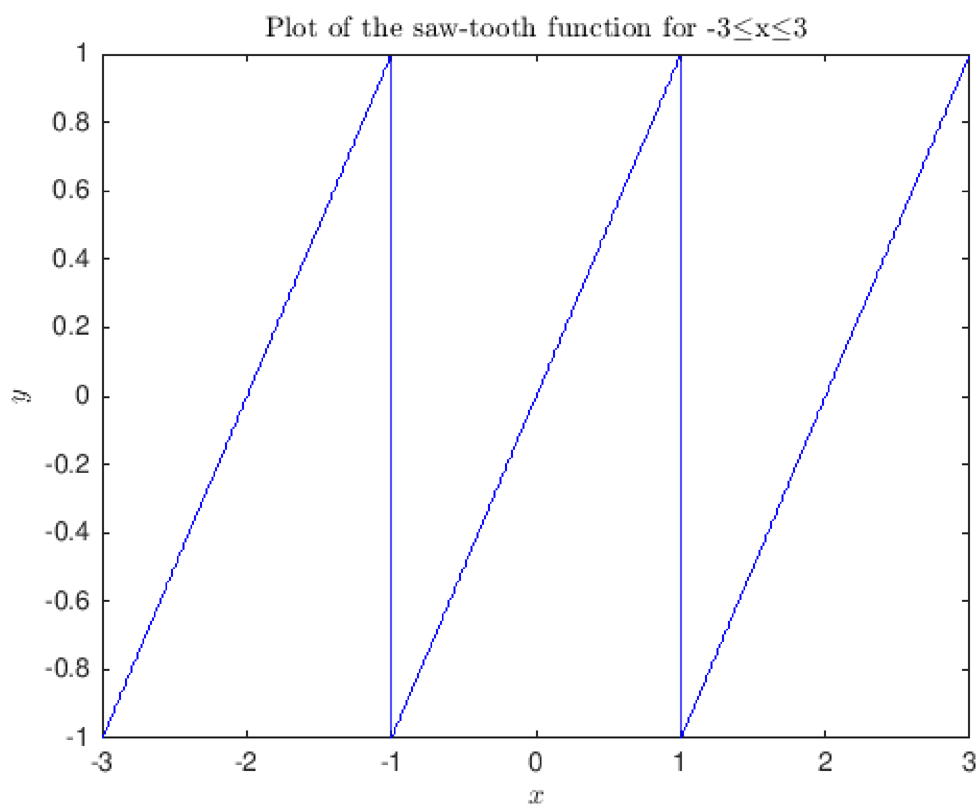
```
%This program takes the user input and displays the prime
 factorisation of this input, with each prime and its power displayed
 on separate lines
% Add comments to the code below to explain what it does.

x = 0; % initialise a variable ready for user input

% Comment out the line with the "input" statement below when you
 publish
% this script to avoid errors.
%x = input('Enter an integer greater than 1: ');%
%the input line prompts user to enter an integer >1
if x < 2 %if x is less than 2
    disp('Your integer must be at least 2. Try again!')%breaks
 program/stops user from inputting any integer less than or equal to 1
elseif floor(x) ~= x %floor function finds the closest integer less
 than or equal to x, if its not equal then prints below
    disp('You must enter and a whole number that is 2 or greater. Try
 again!')%breaks program/stops user from entering a non integer
else
    for i=2:x %meaning i=2 then i=3 and so on to i=x
        count=0;%initialise variable for the for loop
        while rem(x,i)==0
            x = x/i;
            count = count+1;%calculates remainder when x divided by i,
 and if x is a multiple of i divides through and adds 1 to the count
 then repeats, otherwise moves to next i
        end
        if count > 0
            disp([num2str(i), '^', num2str(count)])%displays the i
 that divided x to the power of the number of times it can divide x
        end
        if x == 1 %breaks after all the potential divisors have been
 tested
            break
        end
    end
end
```

Your integer must be at least 2. Try again!

Question 2

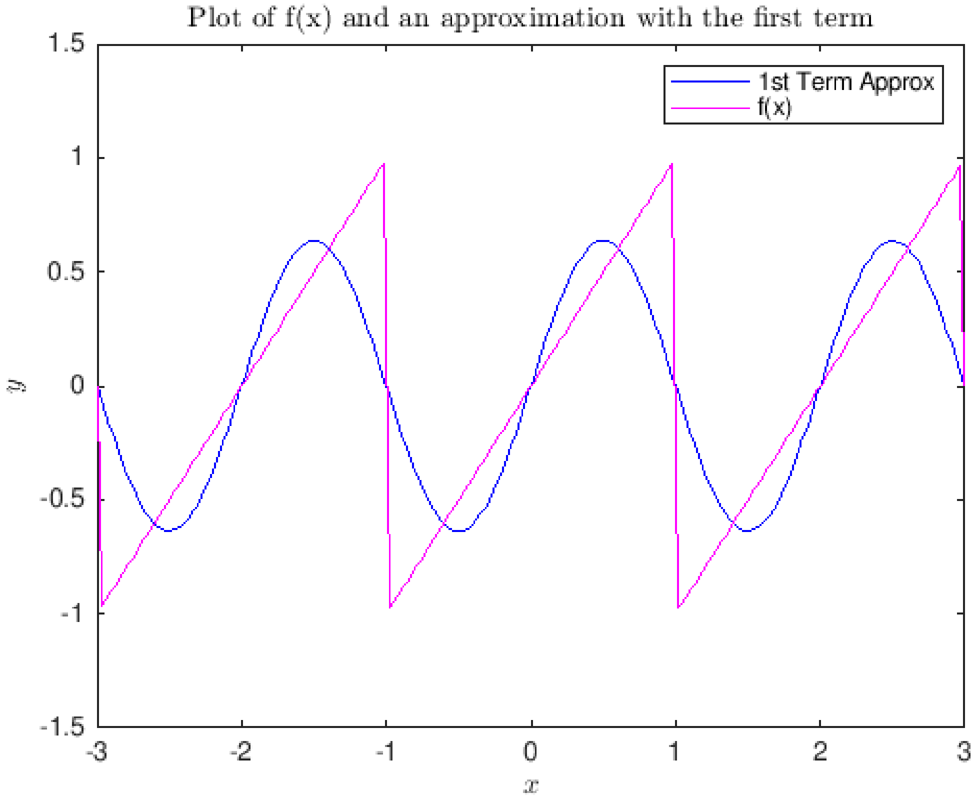
(a) Write a sentence here explaining what your code for Question 2(a) does.

```
%My code plots the saw-tooth function with a period of 2 over the
 domain [-3,3],reproducing the plot given, coded as a piecewise
 function
% Write your code for Question 2(a) here.
x0=-3:0.01:-1;%defines a domain for the first piece between -3 and 1
y0=x0+2;%the equation for the line between -3 and 1
x1=-1:0.01:1;%defines domain for piece between -1 and 1
y1=x1; %the equation for the line between -1 and 1
x2=1:0.01:3;%defines domain for piece between 1 and 3
y2=x2-2;%the equation for the line between 1 and 3
plot(x0,y0,'-b',x1,y1,'-b',x2,y2,'-b');%plots all three pieces defined
 thus far on the same figure
xline(-1,'-b');
xline(1,'-b');%these functions define the plots for the vertical lines
 at x=-1 and x=1
axis([-3 3 -1 1]);%set axis limits for the figure
set(groot,'DefaultTextInterpreter','latex');
title('Plot of the saw-tooth function for -3$\leq$x$\leq$3');
ylabel('$y$');
xlabel('$x$')%these functions allow the title and axis to be labelled
 with LaTex formatting
```



(b) Write a sentence here explaining what your code for Question 2(b) does.

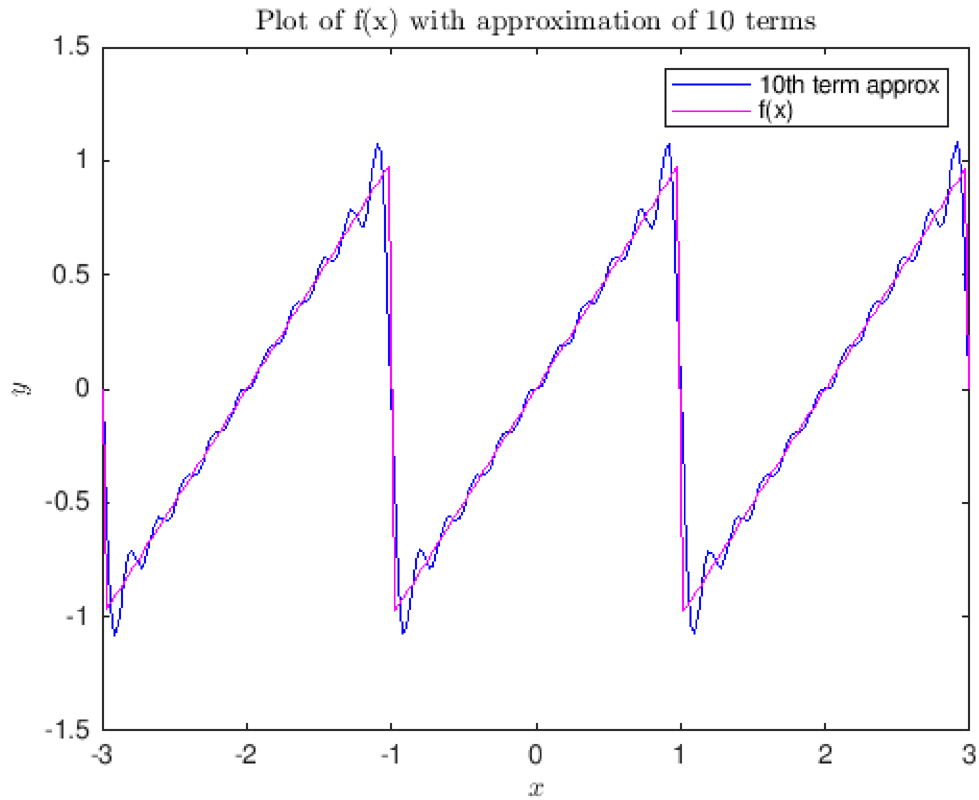
```
%My code plots the fourier series given with a large number of terms
 along with an approximation for the first term in the series
% Write your code for Question 2(b) here.
x=linspace(-3,3,200);%gives a load of points in the desired domain
yapprox=(2./pi)*sin(pi*x);%this is the formula for the first term in
 the series
plot(x,yapprox,'-b');hold on%plot command to plot the approximation
 (blue line) with hold on action allows the second graph to be plotted
 on the same figure
y=0;%initialise variable for the for loop next
N=1000;%large number of terms since f(x) is an infinite series
for n=1:N%for loop, starts from n=1, calculates y then moves to the
 next n until n=N(which is 1000)
   y=y+(2./(n*pi))*(-1)^(n+1)*sin(n*pi*x);%makes the sum of the terms
 of the fourier series
end
plot(x,y,'-m');hold off%plots the second graph which is f(x) with a
 magenta line
axis([-3,3,-1.5,1.5]);%set axis limits so that you can see every point
 that the approximation makes
set(groot,'DefaultTextInterpreter','latex');
title('Plot of f(x) and an approximation with the first term');
ylabel('$y$');
xlabel('$x$');
legend('1st Term Approx','f(x)')%these commands format the axes, title
 and label each plot according to their colour
```



(c) Write a sentence here explaining what your code for Question 2(c) does.

```
%My code plots f(x) and an approximation of f(x) with 10 terms
% Write your code for Question 2(c) here.
term=10;%number of terms in the approximation
x=linspace(-3,3,200);%gives us a load of points in the domain of -3
 and 3
yapprox=0;%initialises this variable for the for loop
for n=1:term%n initially 1, calculates large N and adds it to the
 equation making the formula for the approximation line
    N=(2./(n*pi))*(-1)^(n+1)*sin(n*pi*x);%general formula for each
 term
    yapprox=yapprox+N;%makes yapprox into the sum of the terms
end
plot(x,yapprox,'-b');hold on%plots the approximation (blue line) and
 holds action so that f(x) can also be plotted on the same axes
y=0;%initialise variable for second for loop
termf=1000;%large number of terms so that f(x) is accurate
 representation of the infinite sum
for n=1:termf%same as above but the number of terms is vastly larger
    y=y+(2./(n*pi))*(-1)^(n+1)*sin(n*pi*x);
end
plot(x,y,'-m');hold off%plots f(x) with magenta line on same set of
 axes as the approximation
axis([-3 3 -1.5 1.5]);%sets axis limits as some points in the
 approximation may be greater than 1 or -1
set(groot,'DefaultTextInterpreter','latex');
```

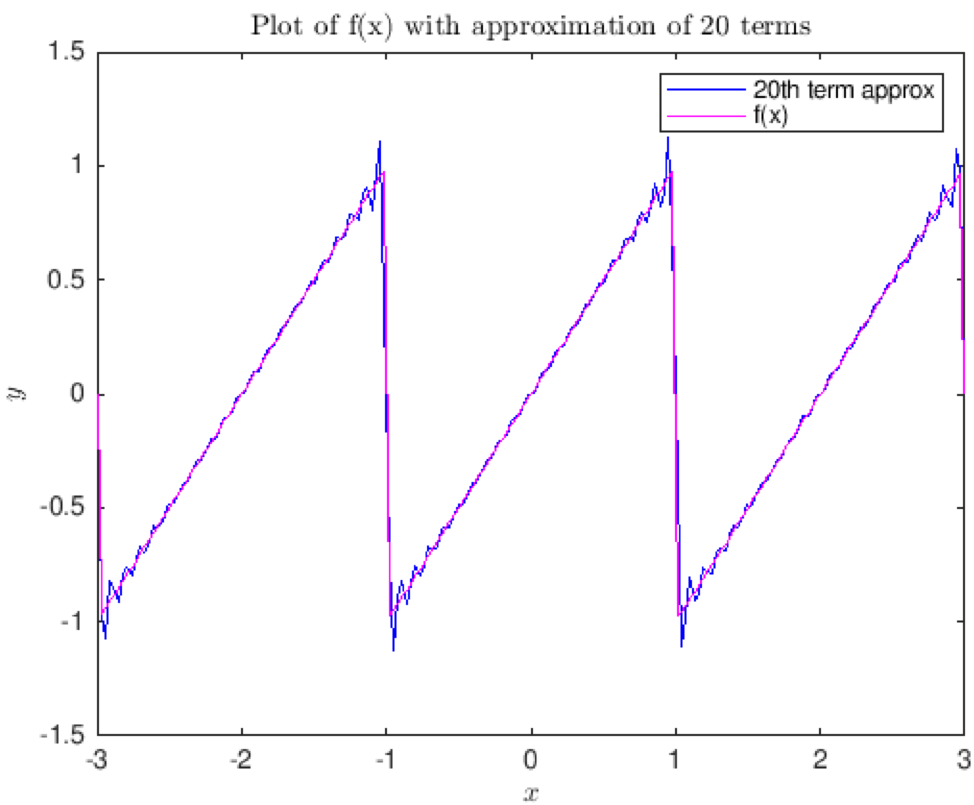
```
title('Plot of f(x) with approximation of 10 terms');
ylabel('$y$');
xlabel('$x$');
legend('10th term approx','f(x)')%labels figure with key to show which
 line is the approximation
```



(d) Write a sentence here explaining what your code for Question 2(d) does.

```
%My code plots f(x) along with an approximation of 20 terms on the
 same set of axes
% Write your code for Question 2(d) here.
term=20;%New number of terms in the approximation
x=linspace(-3,3,200);%A load of points in our domain
yapprox=0;%Initialise variable for for loop
for n=1:term%starts with n=1, finds that term and adds to the
 approximation y to make a new formula with 20 terms
    N=(2./(n*pi))*(-1)^(n+1)*sin(n*pi*x);%general formula for each
 term
    yapprox=yapprox+N;
end
plot(x,yapprox,'-b');hold on%plots approximation as a blue line and
 holds action so f(x) can also be plotted on the same axes
y=0;%initialise this variable for the for loop
termf=1000;%large number of terms so f(x) representative of infinite
 sum
for n=1:termf%starts with n=1, finds the term and adds to y making a
 formula to be plotted
```

```
    y=y+(2./(n*pi))*(-1)^(n+1)*sin(n*pi*x);
end
plot(x,y,'-m');hold off%plots f(x) as a magenta line alongside the
 approximation thanks to the hold action
axis([-3 3 -1.5 1.5]);%set axis limits so the range of the
 approximation can be seen
set(groot,'DefaultTextInterpreter','latex');
title('Plot of f(x) with approximation of 20 terms');
ylabel('$y$');
xlabel('$x$');
legend('20th term approx','f(x)')%new labels for the key to show which
 line is approximation and which is f(x)
```



(e) Write your answer to part Question 2(e) here.

```
%As number of terms included increases the points converge quickly to
 those in f(x). There is also much less oscillation
%near to the points of discontinuity, as this is where there is the
 largest difference between points. For the approximation
%with 10 terms, there are points above and below points on f(x) near
 the points of discontinuity however with 20 terms the
%terms are either above or below, following one clear curve whilst the
 first approximation is more confusing.
```

Published with MATLAB® R2018b