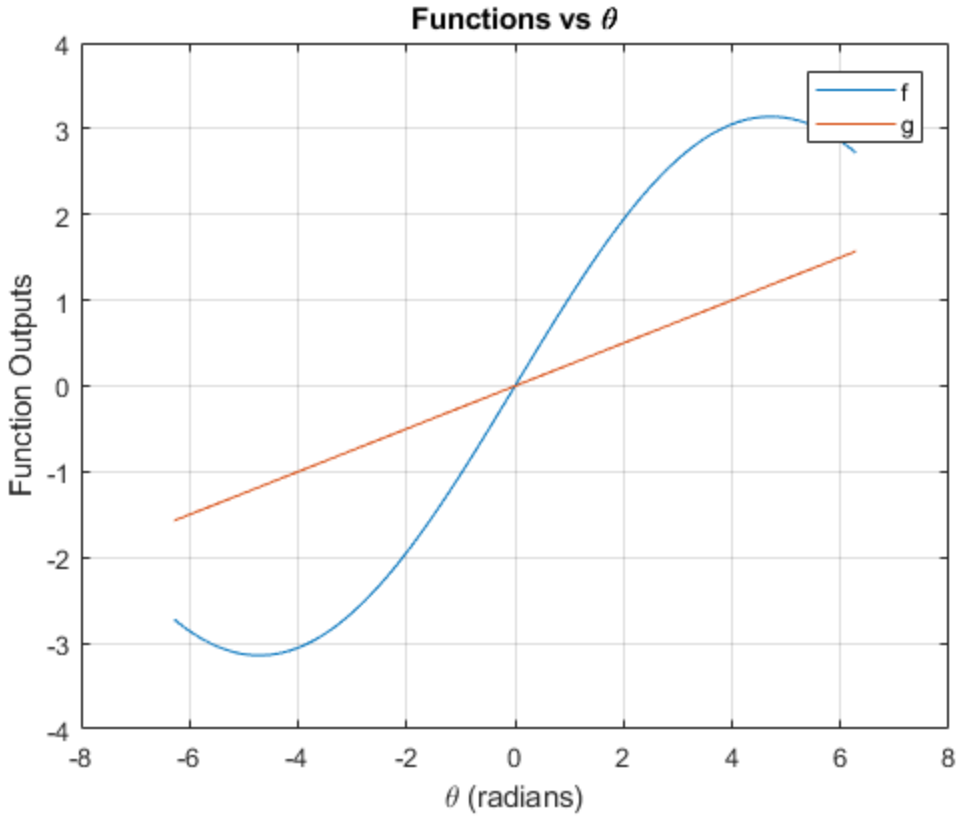
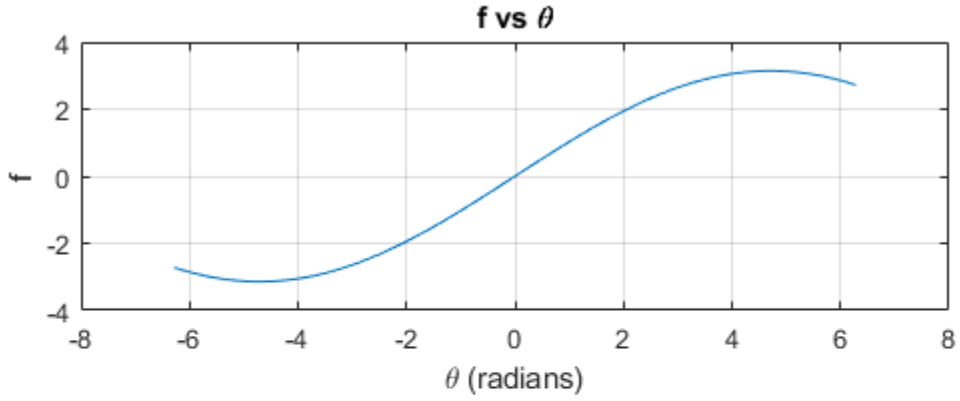
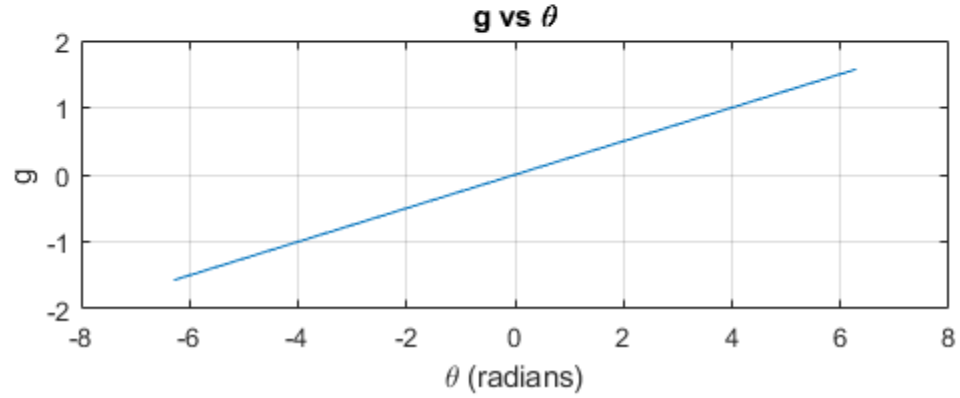
Table of Contents

1

```
   theta = linspace( -2*pi ,  2*pi , 130 ) ; %Creating theta values
for input
   f = pi * sin( theta ./ 3 ) ; %Calculating f for each value of
theta
   g = theta ./ 4 ; %Calculating g for each value of theta

   figure %plots both f and g outputs against theta inputs
   plot( theta , f , theta , g )
   legend( 'f' , 'g' ) %label which lines are which function
   title( 'Functions vs \theta' )
   xlabel( '\theta (radians)' )
   ylabel( 'Function Outputs' )
   grid

   figure % f vs theta is plotted in a graph above the graph of g vs
theta
   subplot( 2 , 1 , 1 )
   plot( theta , f )
   title( 'f vs \theta' )
   xlabel( '\theta (radians)' )
   ylabel( 'f' )
   grid

   subplot( 2 , 1 , 2 )
   plot( theta , g )
   title( 'g vs \theta' )
   xlabel( '\theta (radians)' )
   ylabel( 'g' )
   grid
```


Functions vs θ
f
g
Function Outputs
θ (radians)


f vs θ
f
θ (radians)


g vs θ
g
θ (radians)

2

```
   x = linspace( 0 , 10 , 100000 ) ; %Creates x for inputs
   r1 = zeros( size( x ) ) ; %preallocate r1 with zeros the same size
as x
   r2 = zeros( size( x ) ) ; %preallocate r2 with zeros the same size
as x


   ts_r1 = tic ; %starting timer
   for i = 1:length( x ) %running through loop for every element of x
       r1(i) = 4 *x(i) ^ 3 - 2 * x(i) ^ 2 - 1 ; %calculating r1 using
the corresponding element of x
   end
   t_r1 = toc( ts_r1 ) ; %finds elapsed time of loop


   ts_r2 = tic ; %starting timer
   r2 = 4 * x .^ 3 - 2 * x .^ 2 - 1 ; %Using the whole vector x as
the input variable to calculate r2
   t_r2 = toc( ts_r1 ) ; %finds elapsed time of calculation

   t1 = [ 'The elapsed time to calculate r1 was ' ,
num2str( t_r1 ) , ' s' ] ; %Formatting the output time for r1
   t2 = [ 'The elapsed time to calculate r2 was ' ,
num2str( t_r2 ) , ' s' ] ; %Formatting the output time for r2

   %displaying the times for each calculation
   disp( t1 )
   disp( t2 )
   disp( 'Calculating using a for loop seems to be more effecient for
matlab based on the shorter time to calculate r1' )
```

```
The elapsed time to calculate r1 was 0.034287 s
The elapsed time to calculate r2 was 0.040064 s
Calculating using a for loop seems to be more effecient for matlab
 based on the shorter time to calculate r1
```

3

```
   n = 600 ; %n is number of numbers to be averaged
   data = zeros( 1 , n ) ; %data is data to be averaged, this line
preallocates it
   data = rand( [ 1 , 600 ] ) ; %Fill data with random numbers
   sum_n = 0 ; %preallocating sum

   for i = 1:n
       sum_n = sum_n + data(i) ; %sum of all of the elements in data
   end

   average = sum_n / n ; %finding the average by dividing the sum by
the number of numbers
```

```
   mavg = mean( data ) ; %finding the average using built in matlab
command

   %displaying data
   disp( [ 'My average is ' , num2str( average ) ] )
   disp( [ 'The average calculated by MATLAB is ' ,
num2str( mavg ) ] )
```

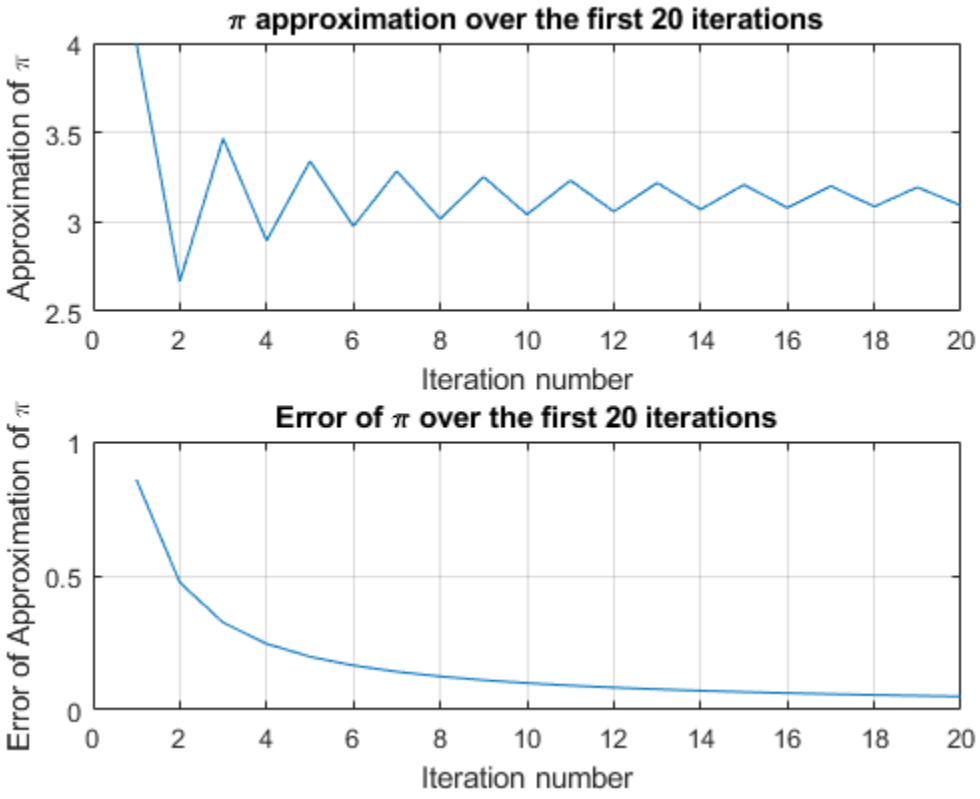
My average is 0.49577
The average calculated by MATLAB is 0.49577

4

```
 %Setting initial values for variables
   p = 0 ;
   pi_a = 0 ;
   k = 2 ;
   e = pi ;

   while e( k-1 ) > 10^-4 %approximates pi to within 10^-4
       p( k ) = 4 * ( ( ( -1 ) ^ ( k ) ) / ( 2 * ( k - 1 ) -
1 ) ) ; %Calculates each term of infinite series to estimate pi
       pi_a( k ) = sum( p ) ; %Sums all terms of series
       e( k ) = abs( pi - pi_a( k ) ) ; %calculates error of pi
approximation
       k = k + 1 ; %counts iteration
   end

   %Display value for pi and number of iterations it took
   disp( [ 'My calculation of pi is ' , num2str( pi_a( k - 1 ) ) , '
after ' , num2str( k - 1 ) , ' steps' ] )

   n = 20 ; %Sets how many iterations used for the following
calculations
   for ii = 1 : n
       pi_n(ii) = pi_a( ii + 1 ) ; %puts the first n iterations of pi
in a seperate vector
       e_n(ii) = e( ii + 1 ) ; %puts the first n iterations of error
in a seperate vector
       k_n = linspace( 1 , n , n ) ;
   end

   %plots approximation of pi and error for the initial n interations
   figure
   subplot( 2 , 1 , 1 ) %plot of pi is on the top
   plot( k_n , pi_n ) %plots iterations against approximation for pi
   title( '\pi approximation over the first 20 iterations' ) %Titles
graph
   xlabel( 'Iteration number' ) %titles x axis
   ylabel( 'Approximation of \pi' ) %title y axis
   grid %puts a grid on graph

   subplot( 2 , 1 , 2 ) %plot of error is below
```

```
    plot ( k_n , e_n ) %plots iterations against error
    title( 'Error of \pi over the first 20 iterations' ) %titles graph
    xlabel( 'Iteration number' ) %titles x axis
    ylabel( 'Error of Approximation of \pi' ) %titles y axis
    grid %puts a grid on graph
```

My calculation of pi is 3.1415 after 10001 steps



Published with MATLAB® R2017b