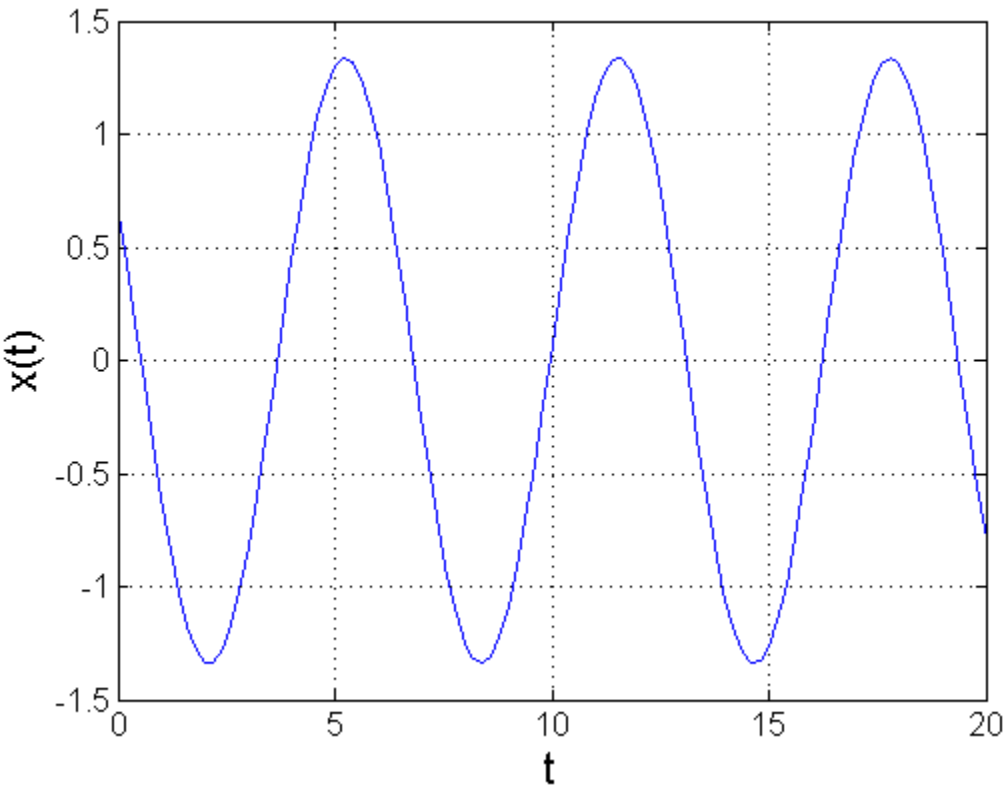
HW0 Problem 1: Harmonic Oscillator (Analytic)

```
fprintf('\n');
clearvars -except function_list pub_opt
close all

times = linspace(0, 20, 101);
plot(times, analytic_harmonic_oscilator(1.34, pi/3, 1, times));
xlabel('t', 'fontsize', 16);
ylabel('x(t)', 'fontsize', 16);
set(gca(),'fontsize',12);
grid on
```

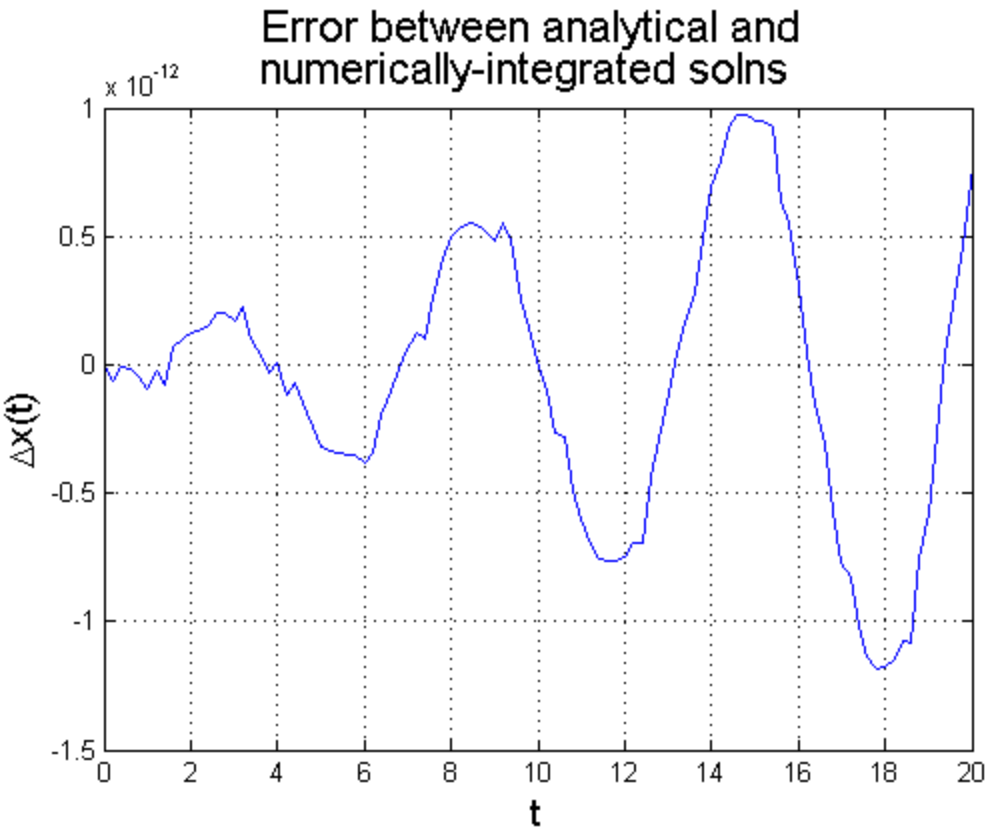


Published with MATLAB® R2013b

HW0 Problem 2: Harmonic Oscillator (Numerical Integration)

```
fprintf('\n');
clearvars -except function_list pub_opt
close all

ode_opts = odeset('RelTol', 1e-12, 'AbsTol', 1e-20);
A = 1.34;
phase = pi/3;
km_rat = 1;
x = [A*cos(phase); -A*sqrt(km_rat)*sin(phase)];

times = linspace(0, 20, 101);

[T,X] = ode45(@harmoscillator, times, x, ode_opts, km_rat);

error = X(:,1)' - analytic_harmonic_oscilator(A, phase, km_rat, times);
plot(times, error);
title('Error between analytical and \newlinenumerically-integrated solns',...
    'fontsize', 16);
xlabel('t', 'fontsize', 14);
ylabel('\Deltax(t)', 'fontsize', 14);
set(gca(),'fontsize',10);
grid on

fprintf('Why would there be an error?\n')
fprintf(...
    ['Ans: The analytical solution is exact and not dependent on \n',...
    '\t previous calculations, while numerical integration is subject\n',...
    '\t to computational error which can compound on previous errors \n',...
    '\t (depending on integrator, time step(s), and model).\n'])
```

```
Why would there be an error?
Ans: The analytical solution is exact and not dependent on
  previous calculations, while numerical integration is subject
  to computational error which can compound on previous errors
  (depending on integrator, time step(s), and model).
```



Published with MATLAB® R2013b

HW 0 Master Script

Table of Contents

Initialize

```
if ispc
    addpath('C:\Users\John\Documents\ASEN5070_SOD\tools')
end
clear all
clc

% Cell array to track what functions are used, so they can be published
% later
global function_list;
function_list = {};

% publishing options
pub_opt.format = 'pdf';
pub_opt.outputDir = '.\html';
pub_opt.imageFormat = 'bmp';
pub_opt.figureSnapMethod = 'entireGUIWindow';
pub_opt.useNewFigure = true ;
pub_opt.maxHeight = Inf;
pub_opt.maxWidth = Inf;
pub_opt.showCode = true;
pub_opt.evalCode = true;
pub_opt.catchError = true;
pub_opt.createThumbnail = true;
pub_opt.maxOutputLines = Inf;
```

Run Problem scripts and publish them

```
% Problem 1
publish('HW0_P1', pub_opt);

% Problem 2
publish('HW0_P2', pub_opt);
```

Publishing tools and support code

```
pub_opt.outputDir = '.\tools';
pub_opt.evalCode = false;
```

```
%Publish all used functions
function_list = ...
    [function_list; 'C:\Users\John\Documents\ASEN5070_SOD\tools\fcnPrintQueue'];
for idx = 1:length(function_list)
    publish(function_list{idx}, pub_opt);
end
```

Published with MATLAB® R2013b

```
function output = analytic_harmonic_oscilator(amplitude, phase, ...
    ang_freq_2, time_array)
%analytic_harmonic_oscilator   Return output for harmonic oscilator given
%   amplitude, phase, (angular frequency)^2, and array of time.
fcnPrintQueue(mfilename('fullpath')) % Add this code to code appendix

output = amplitude*cos(sqrt(ang_freq_2)*time_array + phase);
```

Published with MATLAB® R2013b

```
function fcnPrintQueue( filename )
global function_list;
if exist('function_list', 'var')
    file_in_list = 0;
    for idx = 1:length(function_list)
        if strcmp(function_list(idx), filename);
            file_in_list = 1;
            break
        end
    end
    if ~file_in_list
%         fprintf('%s\n', filename);
        function_list = [function_list; filename];
    end
end
end
```

Published with MATLAB® R2013b

```
function dx = harmoscillator( t, x, kmratio )
%harmoscillator  Return output for
fcnPrintQueue(mfilename('fullpath')) % Add this code to code appendix

dx = zeros(2,1);
if length(x) ~= 2
    return
end

dx(1) = x(2);
dx(2) = -kmratio*x(1);
```

Published with MATLAB® R2013b