ASE 389P.4 Methods of Orbit Determination
Homework 0: Numerical Integration Tool

Junette Hsin
Masters Student, Aerospace Engineering and Engineering Mechanics, University of Texas, Austin, TX 78712

This assignment is designed to provide a basic introduction on how to use a numeric integrator. MATLAB was used to complete the assignment.

I. Problem 1

A. Statement

The differential equation for the harmonic oscillator is

$$\frac{d^2x}{dt^2} = -\frac{k}{m}x, \tag{1}$$

where x is the displacement, k is a positive constant, and m is the mass. The analytic solution is

$$x(t) = A\cos\left(\sqrt{\frac{k}{m}}t + \phi\right) \tag{2}$$

where A is the amplitude and ϕ is a phase constant. Write software to evaluate the analytic solution (Eq. (2)) when given inputs k/m, ϕ, and some array of times t_i, $i = 1, \dots f$. Generate a time array using the MATLAB command

```
times = linspace( 0, 20, 101 );
```

Given the values

$$A = 1.34 \qquad \phi = \pi/3.0 \qquad k/m = 1 \tag{3}$$

plot $x(t)$ from Eq. (2) over time. Provide your plot in the write-up. Be sure to label your axes and use legible fonts! (Points will be deducted for not doing so). One way of doing this is

```
ylabel('x(t)','fontsize',16);
```

with a similar command for the x-axis label. You can change the size of the tick labels using

```
set(gca(),'fontsize',12);
```

B. Solution

For a harmonic oscillator with the analytic solution

$$x(t) = A\, cos\left(\sqrt{\frac{k}{m}}t + \phi\right) \tag{1}$$

Given the values

$$A = 1.34 \qquad \phi = \pi/3.0 \qquad k/m = 1 \tag{2}$$

The plot for $x(t)$ over a time interval of 20 seconds is shown in Figure 1:

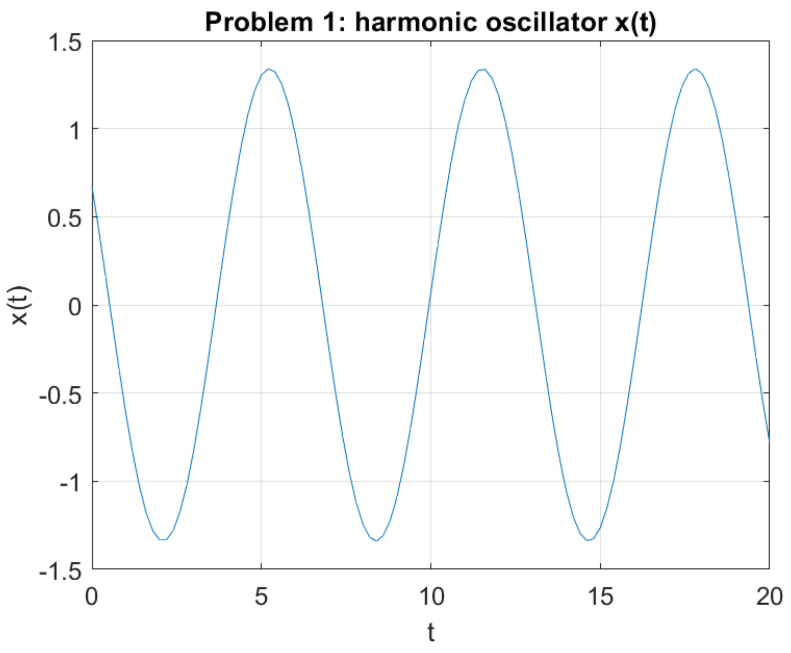

Fig. 1

II. Problem 2

A. Statement

Create a function to evaluate Eq. (1). The code should be similar to:

```
function dx = harmoscillator( t, x, kmratio );

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

where the variable `kmratio` $= k/m$. Now, set the tolerance for the integrator using (for example)

```
myoptions = odeset('RelTol',1e-12,'AbsTol',1e-20);
```

With the initial conditions

$$x(0) = A\cos(\phi) \qquad \dot{x}(0) = -A\sqrt{\frac{k}{m}}\sin(\phi)\,, \tag{4}$$

we will numerically integrate to generate a solution that closely matches the result of question 1. Call `ode45()` (or the tool of your choice) using

```
[T,Y] = ode45(@harmoscillator,times,y0,myoptions,kmratio)
```

where y0 is a 2×1 array of the initial conditions, `kmratio` is k/m, `T` is the output time, and each row of `Y` is x and $\dot{x}$ at a single time. In other words, `Y(10,:)` means the state at time `T(10)`. Now, plot the position state error from `ode45()`, i.e.,

```
error = Y(:,1)'-x;
```

where `x` is the result from Eq. (2). Provide the plot in your write-up. In the above code, `Y(:,1)` means all rows of the first column, or, alternatively, the position at all times. Again, label your axes! Your maximum errors should be on the order of 10^{-6}. Why would there be an error?

B. Solution

An analytical solution provides an exact solution but may not always be easily computed or even possible to obtain, such as in the case of some complex differential equations. Numerical solutions provide approximations which may not exactly match the analytical solution, but can come close within allowable tolerances while being more computationally efficient to obtain.

The accuracy of the MATLAB ordinary differential equations solver `ode45` can be adjusted by modifying the options structure through the function `odeset` [1]. RelTol, the relative accuracy tolerance, controls the number of correct digits in the computed answer, and AbsTol, the absolute error tolerance, controls the difference between the computed answer and the true solution [2]. Figure 2 shows the error between the numerical and analytical solutions to the displacement of the harmonic oscillator for a relative tolerance of 1e-12, which matches the order of magnitude of the error. When the relative tolerance is adjusted to 1e-6 in Figure 3, the order of magnitude of the error adjusts accordingly.

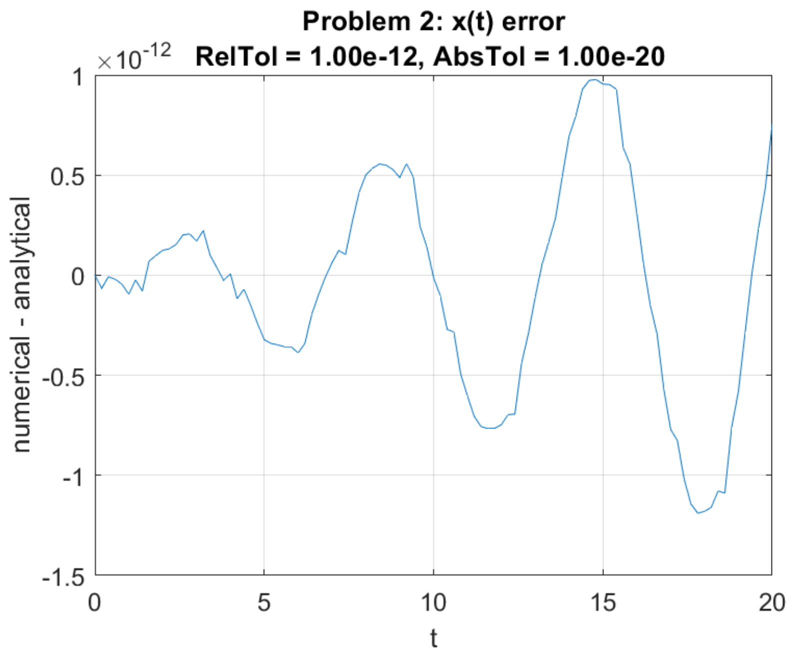

Fig. 2

The error grows over time because MATLAB ODE solvers are set up as initial value problems in which the solution at each step is obtained iteratively by propagating the state of the previous step, beginning at the initial conditions [3]. The truncation of digits and error from numerical integration is propagated as well at each step.

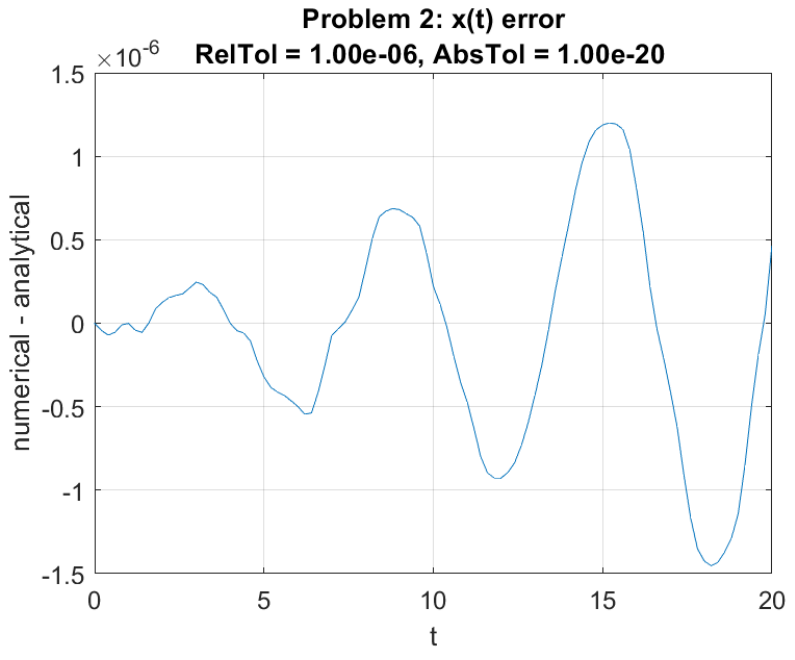

Fig. 3

III. Appendix

HW0 MATLAB code:

```
% ASE 389 Orbit Determination
% HW 0
% Junette Hsin


%% Problem 1

t    = linspace( 0, 20, 101 );
t    = 0 : 0.2 : 400;
A    = 1.34;
phi  = pi/3;
km   = 1;

x    = A * cos( sqrt( km )*t + phi );

name = 'Problem 1: harmonic oscillator x(t)';
figure('name', name);
plot(t, x);
ylabel('x(t)');
xlabel('t');
title(name);
set(gca, 'fontsize', 12);


%% Problem 2

y0 = zeros(2,1);
y0(1) = A * cos(phi);
y0(2) = - A * sqrt( km ) * sin(phi);

reltol = 1e-12;
abstol = 1e-20;
myoptions    = odeset( 'RelTol', reltol, 'AbsTol', abstol);
[t, y]       = ode45( @harmoscillator, t, y0, myoptions, km);

% numerical - analytical
error = y(:,1)' - x;

name = 'Problem 2: x(t) error';
figure('name', name);
plot(t, error);
xlabel('t')
ylabel('numerical - analytical')
title({name ;
sprintf('RelTol = %.2e, AbsTol = %.2e ', reltol, abstol) });
set(gca, 'fontsize', 12);


%% subfunctions

function dx = harmoscillator( t, x, km )

dx     = zeros(2, 1);
dx(1)  = x(2);
dx(2)  = -km * x(1);

end
```

References

[1] MathWorks, “ode45,” `https://www.mathworks.com/help/matlab/ref/ode45.html`, Feb 2021.

[2] MathWorks, “odeset,” `https://www.mathworks.com/help/matlab/ref/odeset.html`, Feb 2021.

[3] MathWorks, “Choose an ODE Solver,” `https://www.mathworks.com/help/matlab/math/choose-an-ode-solver.html`, Feb 2021.