

Assignment 7 - Mod Sim

Patrick
Nitschke
MITK

Problem 1

$$\dot{x} = f(x, u, t)$$

Explicit RK2: $k_1 = f(x_k, u(t_k), t)$

$$k_2 = f(x_k + a \Delta t \cdot k_1, u(t_k + c \Delta t), t_k + c \Delta t)$$

$$x_{k+1} = x_k + b_1 \Delta t k_1 + b_2 \Delta t k_2$$

Error $e_k = x_{k+1} - x(t_{k+1} | k)$, also $x(t_k) = x_k$ (assume same)

a) e_k = error is of order 3 implies RK2

$+ O(\Delta t^3)$

$$x(t_{k+1}) = x(t_k) + \Delta t \cdot \underset{\textcircled{1}}{f(x(t_k), u(t_k))} + \frac{\Delta t^2}{2} \cdot \underset{\textcircled{2}}{\ddot{f}(x(t_k), u(t_k))}$$

RK2:

$$x_{k+1} = x_k + b_1 \Delta t k_1 + b_2 \Delta t k_2 \quad \checkmark f \text{ is LTI}$$

$$k_1 = f(x_k, u(t_k), t)$$

$$k_2 = f(x_k + a \Delta t k_1, u(t_k + c \Delta t), t_k + c \Delta t)$$

$$x_{k+1} = x_k + \underbrace{b_1 \Delta t f(x_k, u(t_k))}_{\textcircled{1}} + b_2 \Delta t \underbrace{f(x_k + a \Delta t f(x_k, u(t_k)), u(t_k + c \Delta t))}_{\text{Taylor exp.}}$$

$$f(x_k + a \Delta t f(x_k, u(t_k + c \Delta t))) \quad \checkmark f \textcircled{2}$$

$$= \underbrace{f(x_k)}_{\textcircled{1}} + a \Delta t \frac{\partial f}{\partial x} \bigg|_{x_k} f(x_k, u(t_k + c \Delta t))$$

equating coefficients:

$$\textcircled{1} (b_1 + b_2) = 1$$

$$\rightarrow b_1 + b_2 = 1$$

$c = a$ for RK2

$$\textcircled{2} b_2 a \Delta t^2 = \frac{\Delta t^2}{2}$$

$$\rightarrow b_2 a = \frac{1}{2}$$

b) ERK2 have best order 2 \rightarrow This means that the global error $\in O(\Delta t^2)$ and one-step error $\in O(\Delta t^3)$.

How order = global error order?

We have N time steps, total error

$$= N \Delta t^3$$

$$N = \frac{\text{Total time}}{\Delta t} = \frac{T}{\Delta t}$$

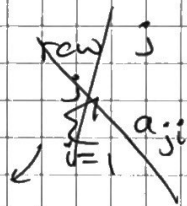
$$\left. \begin{array}{l} \\ \end{array} \right\} \text{Global error} = \underline{T \Delta t^2} = \underline{O(\Delta t^2)}$$

$$c) x(t) = x(t_k) + \sum_{i=1}^n \frac{a_i}{i!} (t-t_k)^i, \quad t \in [t_k, t_{k+1}]$$

\rightarrow Error disappears as $\underline{n \rightarrow \infty}$.

$$\rightarrow \sum_i b_i = 1$$

$\rightarrow c = \sum_i a_i$ \sum a's on that row?



\rightarrow Infer these from tables in problem 2 I guess.

Problem 2

Simulate classic system $\dot{x} = \lambda x$

a) Implement code on TestERK.m

We see that stability, when $\Delta t = 0.4$ are only apparent for ERK1, while ^{issues} ERK2, ERK3 are fine. fine.

By $t_{\text{me}} = 100$ s, the oscillations disappear.

b) ERK1 reaches marginal stability

$$\frac{\Delta t (s)}{0.1}$$

ERK2 — " —

$$\sim 0.8$$

ERK3 — " —

$$\sim 1.95$$

See plots attached.

Order of the methods as a func. of time step?

(Compare global error with Δt and plot.) - student

Global error actual:

$$\dot{x} = Ax$$

$$\frac{dx}{dt} = Ax$$

$$\int_{x_0}^{x(t)} \frac{1}{x} dx = \int_0^t A dt$$

$$\ln x \Big|_{x_0}^{x(t)} = A t \Big|_0^t$$

$$\ln x(t) - \ln(x_0) = A^t - A^{t_0}$$

$$\frac{x(t)}{x_0} = e^{A(t-t_0)}$$

$$x(t) = e^{A(t-t_0)} x_0$$

Real value at $t=100s$, $t_0=0$

$$\hookrightarrow x(t) = \begin{bmatrix} 1 \\ 1 \end{bmatrix} e^{A(100)} \begin{bmatrix} 1 \\ 1 \end{bmatrix} ??$$

We know global error:

ERK1 : g. error $\in C \cdot \Delta t$

$$: \|x(T) - x_N\| = O(\Delta t)$$

ERK2 : $\|x(T) - x_N\| = O(\Delta t^2)$

" 3 : " = $O(\Delta t^3)$

Plot this.

c) Values for λ ? $\Delta t = 0.4$.

ERK1: Marginally stable for $\lambda = 5$
Unstable for $\lambda < -5$.

ERK2: Marginally stable $\lambda = -5$,
Unstable for $\lambda < -5$.

ERK3: Marginally stable around $\lambda = -6.3$
Unstable for $\lambda < -6.3$.

Problem 3

a) We see that x, y follows a normal Van der Pol ~~plot~~ behaviour, but the Δt 's (time steps) vary quite significantly. ODE45 starts off with almost 0 (0.000025s) as the first time step but by $t=1$ second it's at ~ 0.08 s. The time steps continue to increase until just about 5 seconds where it ~~too~~ reduces to 0.01, a factor 10 decrease. This is ~~due to~~ due to ODE45 being an adaptive integrator, as in it changes its time steps according to the fast dynamics.

b) We need 154 time-steps to simulate for 25 ~~sec~~ seconds, otherwise it just goes crazy to like 10^{300} in magnitude.

$\Delta t \leq 0.1633986$ is OK. \leftarrow
 $\Delta t \geq 0.1633987$ not OK at all. $\left. \vphantom{\Delta t} \right\}$ see plot.

At $\Delta t = 0.1$ s the ERK4 does not do so bad, with a 0.01s delay on x and a 0.025s delay on y .

Once Δt gets to about 0.01s then they are almost identical, as ODE45's smallest time steps are 0.011s.

\rightarrow See plot.

Table of Contents

Test system definition - Default system now, prob 2c) done here	1
ERK 1	1
ERK2	2
ERK3	3
Plotting error vs deltaT - PROB 2b done here)	4
Mass-damper-spring parameters - Testing this for fun	7
ERK 1	7
ERK2	9
ERK3	10
Plotting error vs deltaT	11
Prob 3 - Van der Pol oscillator	15
ERK4	15
Plotting 3b deltaT = 0.1633986, part 3a is subplot 3	16
ERK4	17
Plotting 3b, deltaT = 0.1633987 (NOTE: Stopped sim after 7.06101 seconds)	18
Plotting 3b, deltaT = 0.1633987 (NOTE: Sim to 7.065 seconds)	19

Test system definition - Default system now, prob 2c) done here

```
lambda = -2;  
fTestSystem = @(t,x) lambda*x;  
x0 = 1;  
  
% Simulation parameters  
deltaT = 0.4;  
totalT = 2;  
T = linspace(0,totalT, 1 + totalT/deltaT);
```

ERK 1

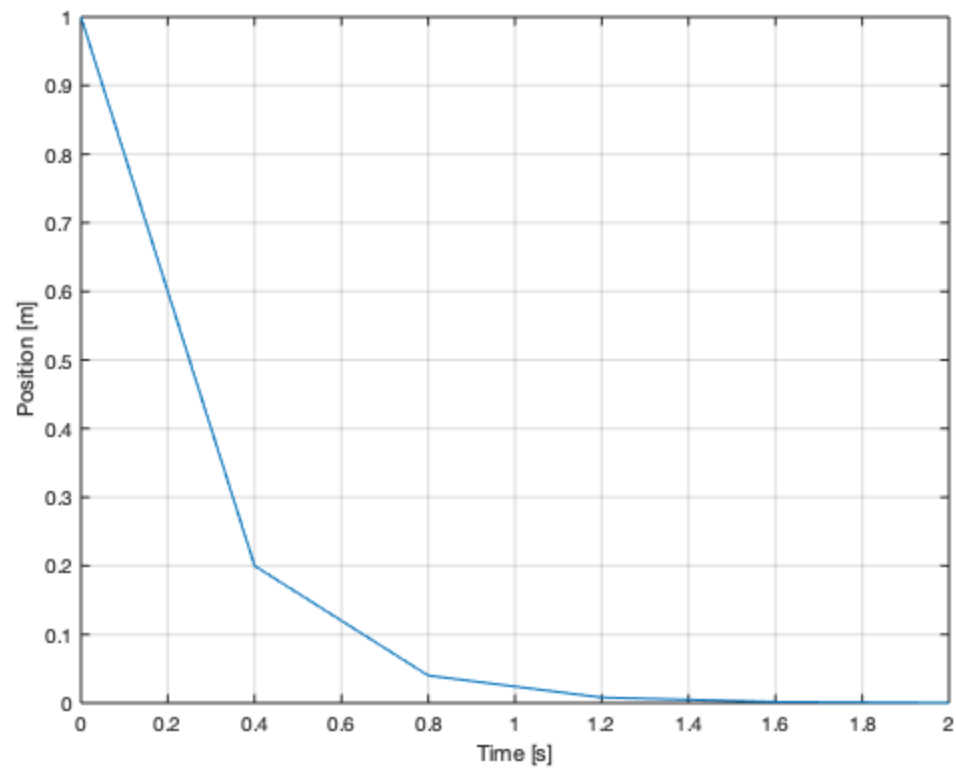
Heun's first-order method

```
A = 0;  
c = sum(A);  
b = 1;  
HeunArray = struct('A',A,'b',b,'c',c);  
  
% Simulate  
%disp('Starting ERK1');  
Nx = length(x0);  
Nstage = length(A);  
Nt = length(T);  
X1 = ERKTemplate(HeunArray,fTestSystem,T,x0,Nx,Nstage,Nt);  
% Plot  
figure(1)  
plot(T,X1)  
ylabel('Position [m]')
```

```

xlabel('Time [s]')
grid on

```



ERK2

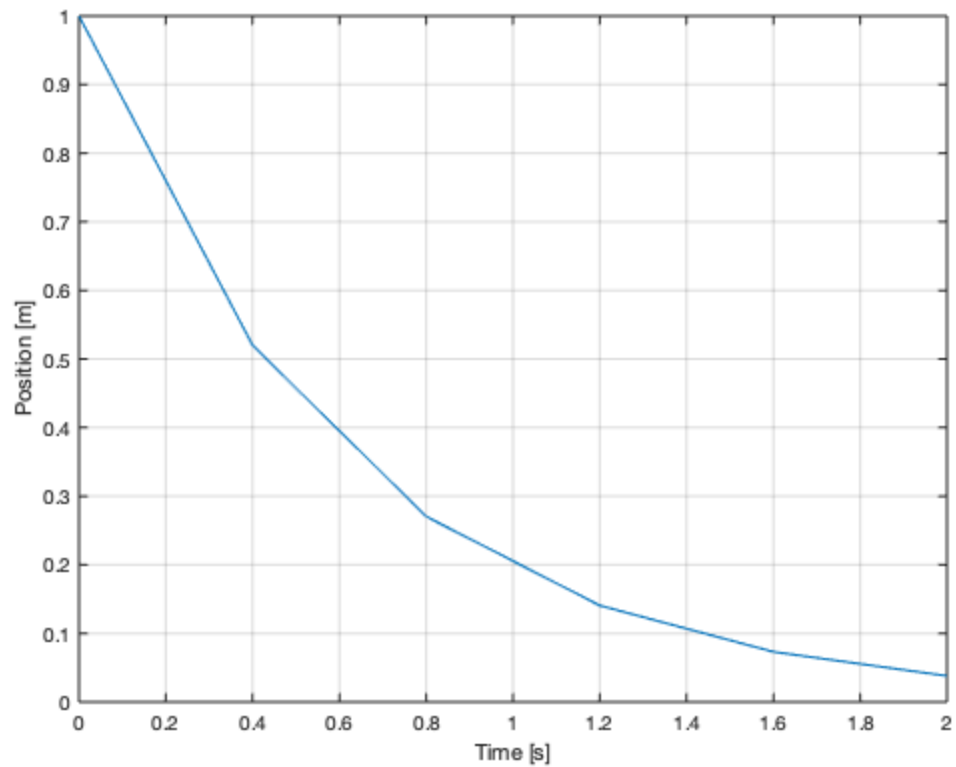
Heun's second-order method

```

A = [0  0;
     1/2 0];
c = sum(A,2);
b = [0; 1];
HeunArray = struct('A',A,'b',b,'c',c);

%Simulate
%disp('Starting ERK2');
Nx = length(x0);
Nstage = length(A);
Nt = length(T);
X2 = ERKTemplate(HeunArray,fTestSystem,T,x0,Nx,Nstage,Nt);
% Plot
figure(2)
plot(T,X2)
ylabel('Position [m]')
xlabel('Time [s]')
grid on

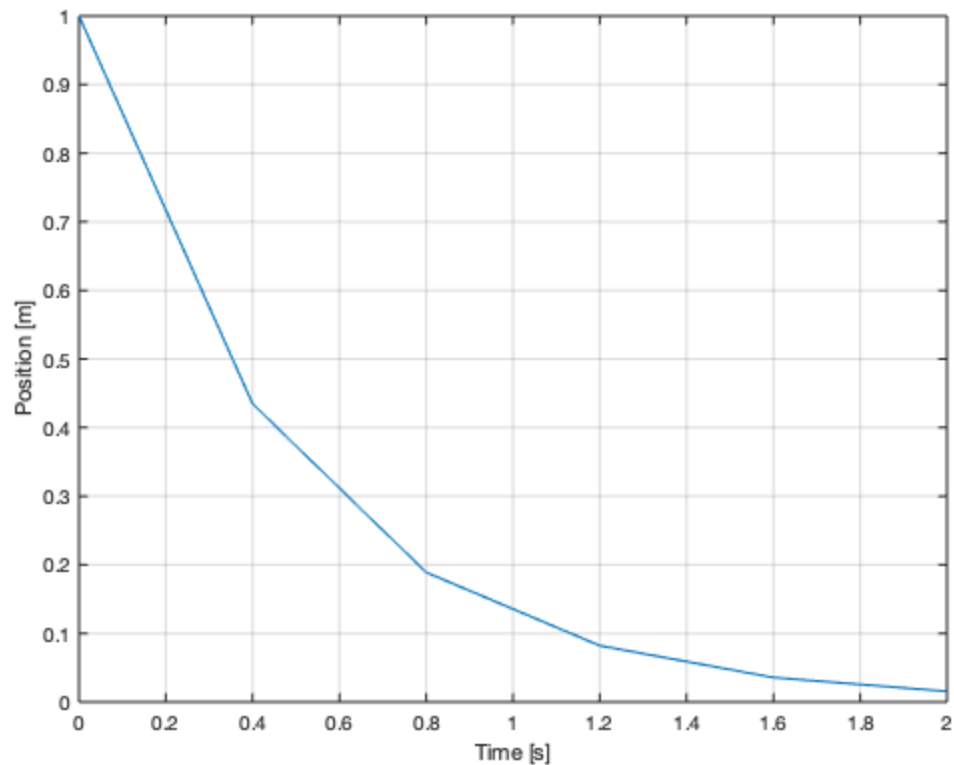
```



ERK3

Heun's third-order method

```
A = [0 0 0;  
     1/3 0 0;  
     0 2/3 0];  
c = sum(A,2);  
b = [1/4; 0; 3/4];  
HeunArray = struct('A',A,'b',b,'c',c);  
  
% Simulate  
%disp('Starting ERK3');  
Nx = length(x0);  
Nstage = length(A);  
Nt = length(T);  
X3 = ERKTemplate(HeunArray,fTestSystem,T,x0,Nx,Nstage,Nt);  
% Plot  
figure(3)  
plot(T,X3)  
ylabel('Position [m]')  
xlabel('Time [s]')  
grid on
```



Plotting error vs deltaT - PROB 2b done here)

```
%disp('Starting real answer');
tspan = [0 totalT];
[t_ans, X_ans] = ode45(fTestSystem, tspan, x0);
x_last_ans = X_ans(length(X_ans),:);

index = 1;
start_deltaT = 0.0001;
inc_deltaT = 0.04;
end_deltaT = start_deltaT + 0.4;
n_index = floor(end_deltaT-start_deltaT)/inc_deltaT;

error_deltaT = zeros(3,n_index);
deltaTs = zeros(1,n_index);

for i = start_deltaT:inc_deltaT:end_deltaT
    deltaT = i;
    T = linspace(0,totalT, 1 + totalT/deltaT);

    % ERK 1
    A = 0;
    c = sum(A);
    b = 1;
    HeunArray = struct('A',A,'b',b,'c',c);
```

```

% Simulate
%disp('Starting ERK1');
Nx = length(x0);
Nstage = length(A);
Nt = length(T);
X1 = ERKTemplate(HeunArray,fTestSystem,T,x0,Nx,Nstage,Nt);

% ERK2
A = [0 0;
     1/2 0];
c = sum(A,2);
b = [0; 1];
HeunArray = struct('A',A,'b',b,'c',c);

% Simulate
%disp('Starting ERK2');
Nx = length(x0);
Nstage = length(A);
Nt = length(T);
X2 = ERKTemplate(HeunArray,fTestSystem,T,x0,Nx,Nstage,Nt);

% ERK3
A = [0 0 0;
     1/3 0 0;
     0 2/3 0];
c = sum(A,2);
b = [1/4; 0; 3/4];
HeunArray = struct('A',A,'b',b,'c',c);

% Simulate
%disp('Starting ERK3');
Nx = length(x0);
Nstage = length(A);
Nt = length(T);
X3 = ERKTemplate(HeunArray,fTestSystem,T,x0,Nx,Nstage,Nt);

error_deltaT(1,index) = abs(X1(Nt) - x_last_ans);
error_deltaT(2,index) = abs(X2(Nt) - x_last_ans);
error_deltaT(3,index) = abs(X3(Nt) - x_last_ans);
deltaTs(index) = i;

msg = ['Iteration ', index, ' complete'];
%disp(msg);
index = index + 1;
end

% Plot
figure(4)
subplot(3,1,1)
hold on
plot(deltaTs, error_deltaT(1,:), '-rx')
grid on
ylabel('Error in Position [m]')

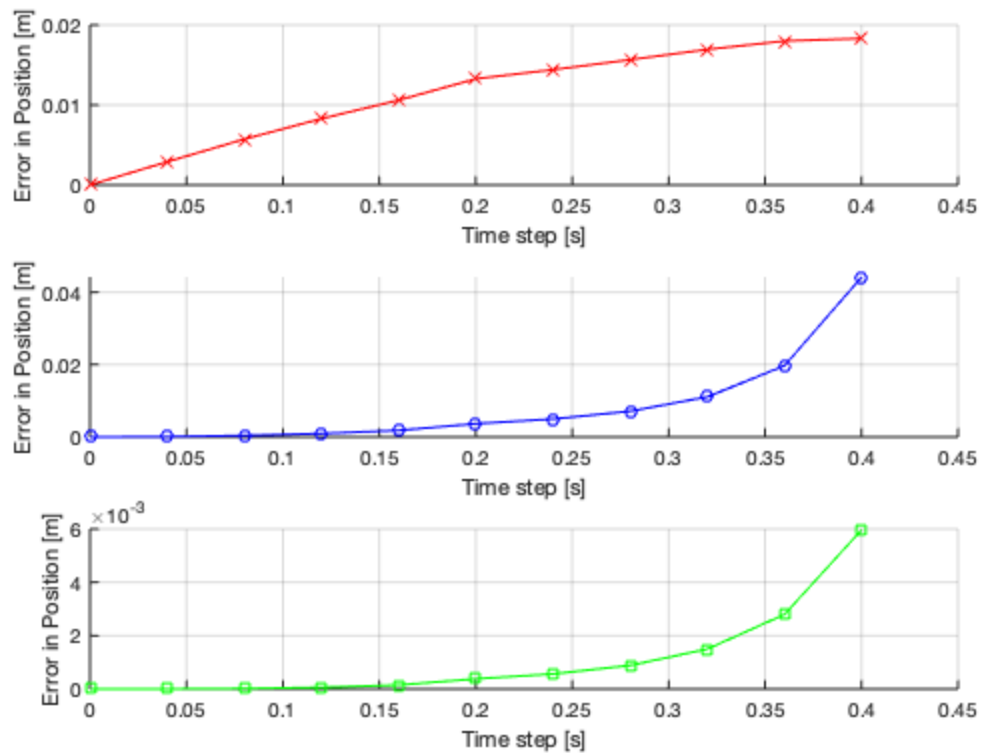
```

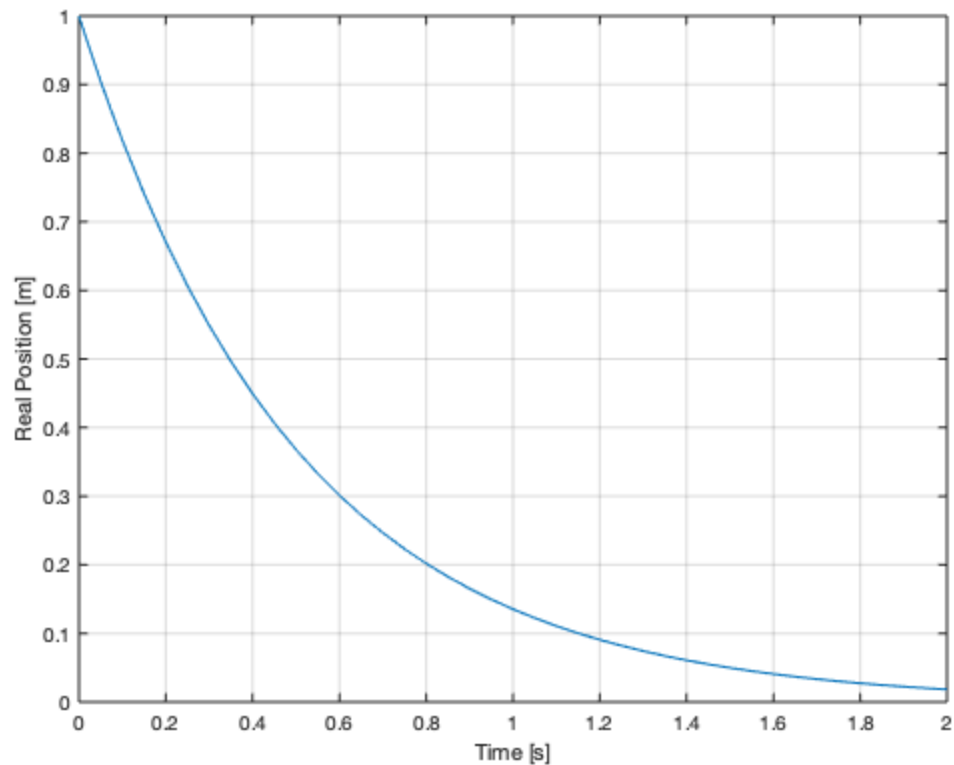
```

xlabel('Time step [s]')
subplot(3,1,2)
hold on
plot(deltaTs, error_deltaT(2,:), '-bo')
grid on
ylabel('Error in Position [m]')
xlabel('Time step [s]')
subplot(3,1,3)
hold on
plot(deltaTs, error_deltaT(3,:), '-gs')
grid on
ylabel('Error in Position [m]')
xlabel('Time step [s]')

% Plot answer
figure(5)
title('Result using ode45')
plot(t_ans,X_ans);
grid on
ylabel('Real Position [m]')
xlabel('Time [s]')

```





Mass-damper-spring parameters - Testing this for fun

```
m = 1;
d = 0.1;
k = 1;
A = [ 0      1;
      -k/m -d/m];
% Mass-damper-spring vector field
fMassDamperSpring = @(t,x) A*x;
x0 = [1;1];
% Test system
fTestSystem = @(t,x) -2*x;
%x0 = 1;

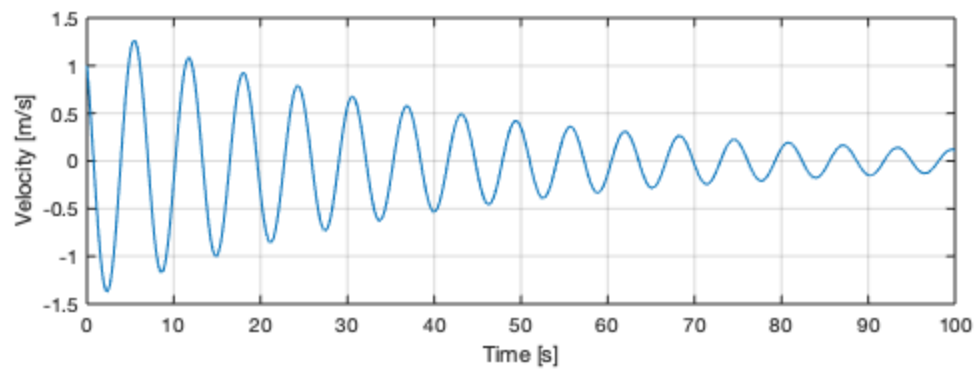
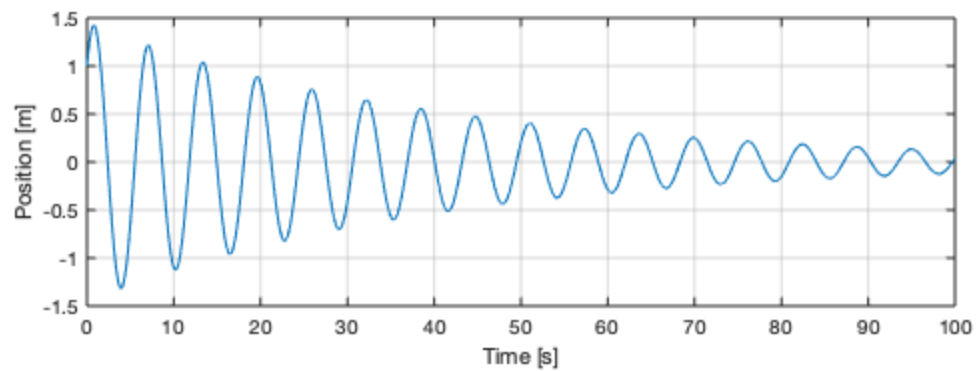
% Simulation parameters
deltaT = 0.05;
totalT = 100;
T = linspace(0,totalT, 1 + totalT/deltaT);
```

ERK 1

```
% Heun's first-order method
```

```
A = 0;
c = sum(A);
b = 1;
HeunArray = struct('A',A,'b',b,'c',c);

% Simulate
%disp('Starting ERK1');
Nx = length(x0);
Nstage = length(A);
Nt = length(T);
X1 = ERKTemplate(HeunArray,fMassDamperSpring,T,x0,Nx,Nstage,Nt);
% Plot
figure(6)
subplot(2,1,1)
plot(T,X1(1,:))
ylabel('Position [m]')
xlabel('Time [s]')
grid on
subplot(2,1,2)
plot(T,X1(2,:))
grid on
ylabel('Velocity [m/s]')
xlabel('Time [s]')
```

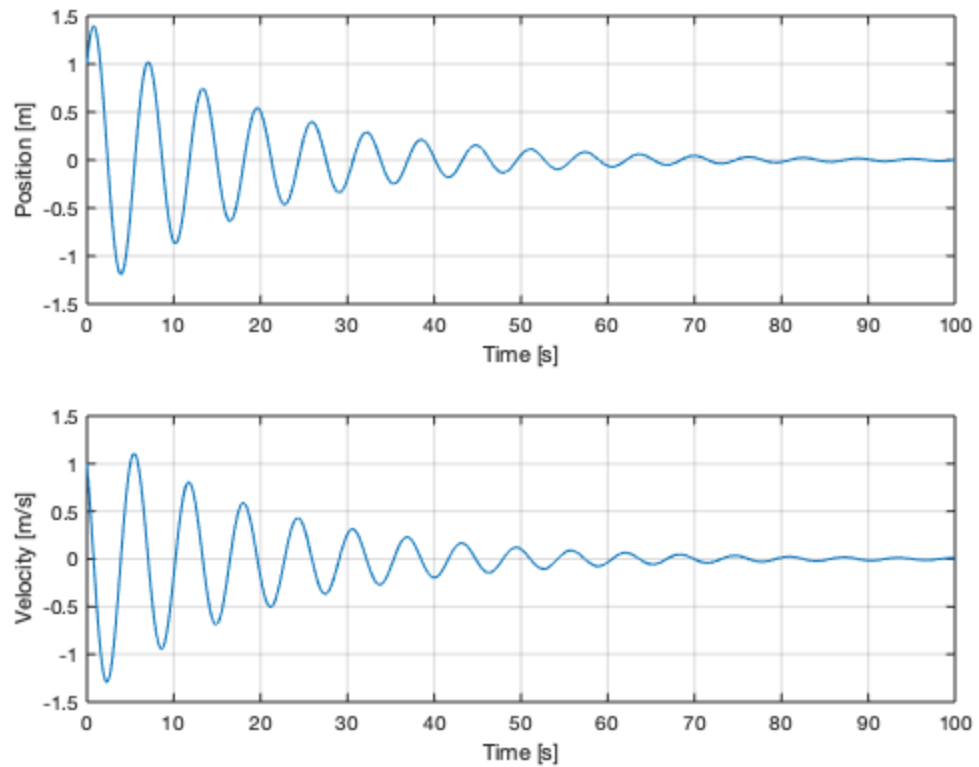


ERK2

Heun's second-order method

```
A = [0 0;
      1/2 0];
c = sum(A,2);
b = [0; 1];
HeunArray = struct('A',A,'b',b,'c',c);

%Simulate
%disp('Starting ERK2');
Nx = length(x0);
Nstage = length(A);
Nt = length(T);
X2 = ERKTemplate(HeunArray,fMassDamperSpring,T,x0,Nx,Nstage,Nt);
% Plot
figure(7)
subplot(2,1,1)
plot(T,X2(1,:))
ylabel('Position [m]')
xlabel('Time [s]')
grid on
subplot(2,1,2)
plot(T,X2(2,:))
grid on
ylabel('Velocity [m/s]')
xlabel('Time [s]')
```



ERK3

Heun's third-order method

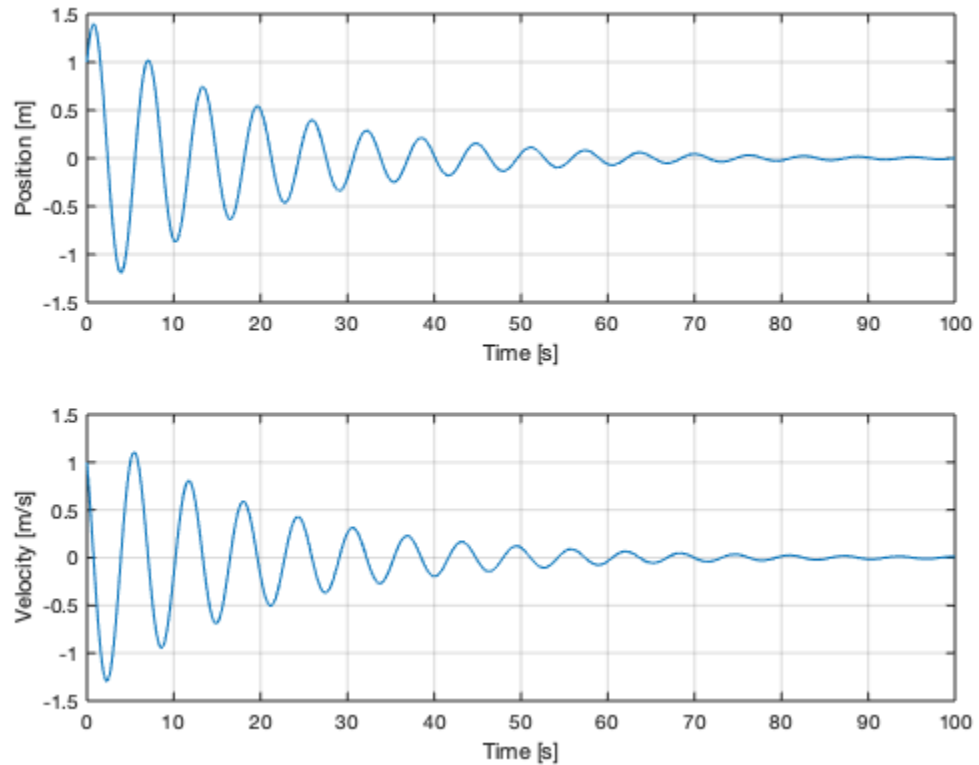
```
A = [0 0 0;
      1/3 0 0;
      0 2/3 0];
c = sum(A,2);
b = [1/4; 0; 3/4];
HeunArray = struct('A',A,'b',b,'c',c);

% Simulate
%disp('Starting ERK3');
Nx = length(x0);
Nstage = length(A);
Nt = length(T);
X3 = ERKTemplate(HeunArray,fMassDamperSpring,T,x0,Nx,Nstage,Nt);
% Plot
figure(8)
subplot(2,1,1)
plot(T,X3(1,:))
ylabel('Position [m]')
xlabel('Time [s]')
grid on
subplot(2,1,2)
plot(T,X3(2,:))
```

```

grid on
ylabel('Velocity [m/s]')
xlabel('Time [s]')

```



Plotting error vs deltaT

```

%disp('Starting real answer');
tspan = [0 totalT];
[t_ans, X_ans] = ode45(fMassDamperSpring, tspan, x0);
x_last_ans = X_ans(length(X_ans),:);

index = 1;
start_deltaT = 0.05;
inc_deltaT = 0.05;
end_deltaT = 1;
n_index = floor((end_deltaT-start_deltaT)/inc_deltaT);

error_deltaT = zeros(6,n_index);
deltaTs = zeros(1,n_index);

for i = start_deltaT:inc_deltaT:end_deltaT
    deltaT = i;
    T = linspace(0,totalT,totalT/deltaT);

    % ERK 1
    A = 0;

```

```

c = sum(A);
b = 1;
HeunArray = struct('A',A,'b',b,'c',c);

% Simulate
%disp('Starting ERK1');
Nx = length(x0);
Nstage = length(A);
Nt = length(T);
X1 = ERKTemplate(HeunArray,fMassDamperSpring,T,x0,Nx,Nstage,Nt);

% ERK2
A = [0 0;
     1/2 0];
c = sum(A,2);
b = [0; 1];
HeunArray = struct('A',A,'b',b,'c',c);

% Simulate
%disp('Starting ERK2');
Nx = length(x0);
Nstage = length(A);
Nt = length(T);
X2 = ERKTemplate(HeunArray,fMassDamperSpring,T,x0,Nx,Nstage,Nt);

% ERK3
A = [0 0 0;
     1/3 0 0;
     0 2/3 0];
c = sum(A,2);
b = [1/4; 0; 3/4];
HeunArray = struct('A',A,'b',b,'c',c);

% Simulate
%disp('Starting ERK3');
Nx = length(x0);
Nstage = length(A);
Nt = length(T);
X3 = ERKTemplate(HeunArray,fMassDamperSpring,T,x0,Nx,Nstage,Nt);

error_deltaT(1:2,index) = abs(X1(:,Nt) - x_last_ans);
error_deltaT(3:4,index) = abs(X2(:,Nt) - x_last_ans);
error_deltaT(5:6,index) = abs(X3(:,Nt) - x_last_ans);
deltaTs(index) = i;

msg = ['Iteration ', index, ' complete'];
%disp(msg);
index = index + 1;
end

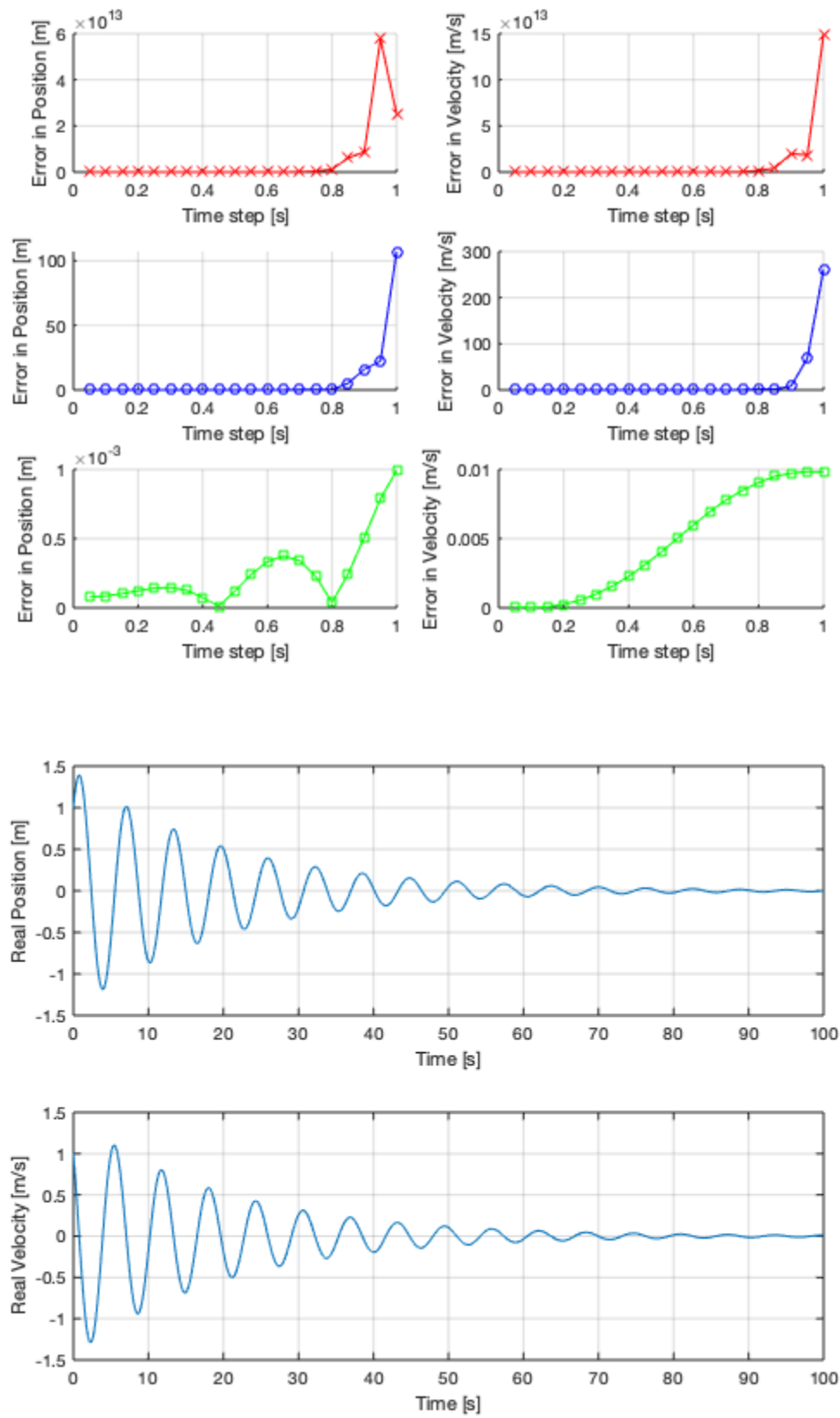
% Plot
figure(9)
subplot(3,2,1)
hold on

```

```
plot(deltaTs, error_deltaT(1,:), '-rx')
grid on
ylabel('Error in Position [m]')
xlabel('Time step [s]')
subplot(3,2,3)
hold on
plot(deltaTs, error_deltaT(3,:), '-bo')
grid on
ylabel('Error in Position [m]')
xlabel('Time step [s]')
subplot(3,2,5)
hold on
plot(deltaTs, error_deltaT(5,:), '-gs')
grid on
ylabel('Error in Position [m]')
xlabel('Time step [s]')

subplot(3,2,2)
hold on
plot(deltaTs, error_deltaT(2,:), '-rx')
grid on
ylabel('Error in Velocity [m/s]')
xlabel('Time step [s]')
subplot(3,2,4)
hold on
plot(deltaTs, error_deltaT(4,:), '-bo')
grid on
ylabel('Error in Velocity [m/s]')
xlabel('Time step [s]')
subplot(3,2,6)
hold on
plot(deltaTs, error_deltaT(6,:), '-gs')
grid on
ylabel('Error in Velocity [m/s]')
xlabel('Time step [s]')

% Plot answer
figure(10)
subplot(2,1,1)
title('Result using ode45')
plot(t_ans,X_ans(:,1));
grid on
ylabel('Real Position [m]')
xlabel('Time [s]')
subplot(2,1,2)
plot(t_ans,X_ans(:,2));
grid on
ylabel('Real Velocity [m/s]')
xlabel('Time [s]')
```



Prob 3 - Van der Pol oscillator

```
% System
% dxdt = y;
% dydt = u*(1-x^2)*y-x;

% Parameters
u = 5;
x0 = 2;
y0 = 0;
z0 = [x0; y0];
tspan = [0 25];

% Simulate
f_vanderpol = @(t,z) [z(2); u*(1-z(1)^2)*z(2)-z(1)];
[t, z] = ode45(f_vanderpol, tspan, z0);

deltaTs = zeros(length(t),1);
deltaTs(1) = t(1);
for i = 2:length(t)
    deltaTs(i) = t(i) - t(i-1);
end
```

ERK4

Following table 3 - RK4

```
A = [0    0    0    0 ;
      1/2  0    0    0;
      0    1/2  0    0;
      0    0    1    0];
c = sum(A,2);
b = [1/6; 1/3; 1/3; 1/6];
HeunArray = struct('A',A,'b',b,'c',c);

% Simulation parameters
%deltaT = 0.1635;
%totalT = 7.062;
deltaT = 0.1633986;
%deltaT = 0.02;
totalT = 25;
T = linspace(0,totalT, 1 + totalT/deltaT);

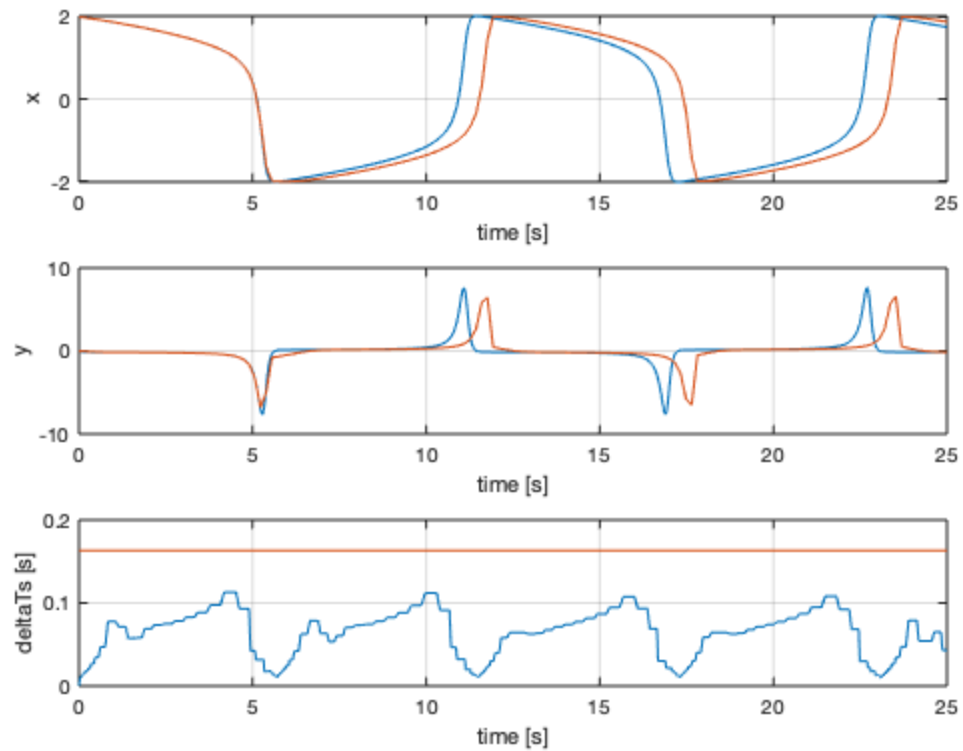
% Simulate
%disp('Starting ERK4');
Nx = length(z0);
Nstage = length(A);
Nt = length(T);
X3 = ERK_vdp(HeunArray,f_vanderpol,T,z0,Nx,Nstage,Nt);
```

Plotting 3b $\Delta T = 0.1633986$, part 3a is subplot 3

```
figure (11)
subplot(311)
plot(t,z(:,1))
hold on
plot(T,X3(1,:))
grid on
xlabel('time [s]'), ylabel('x')

subplot(312)
plot(t,z(:,2))
hold on
plot(T,X3(2,:))
grid on
xlabel('time [s]'), ylabel('y')

deltaTs2 = deltaT * ones(length(T),1);
subplot(313)
plot(t,deltaTs)
hold on
plot(T, deltaTs2)
grid on
xlabel('time [s]'), ylabel('deltaTs [s]')
```

ERK4

Following table 3 - RK4

```
A = [0 0 0 0 ;
      1/2 0 0 0;
      0 1/2 0 0;
      0 0 1 0];
c = sum(A,2);
b = [1/6; 1/3; 1/3; 1/6];
HeunArray = struct('A',A,'b',b,'c',c);

% Simulation parameters
%deltaT = 0.1635;
totalT = 7.06101;
deltaT = 0.1633987;
%deltaT = 0.02;
%totalT = 25;
T = linspace(0,totalT, 1 + totalT/deltaT);

% Simulate
%disp('Starting ERK4');
Nx = length(z0);
Nstage = length(A);
Nt = length(T);
X3 = ERK_vdp(HeunArray,f_vanderpol,T,z0,Nx,Nstage,Nt);
```

Plotting 3b, $\Delta T = 0.1633987$ (NOTE: Stopped sim after 7.06101 seconds)

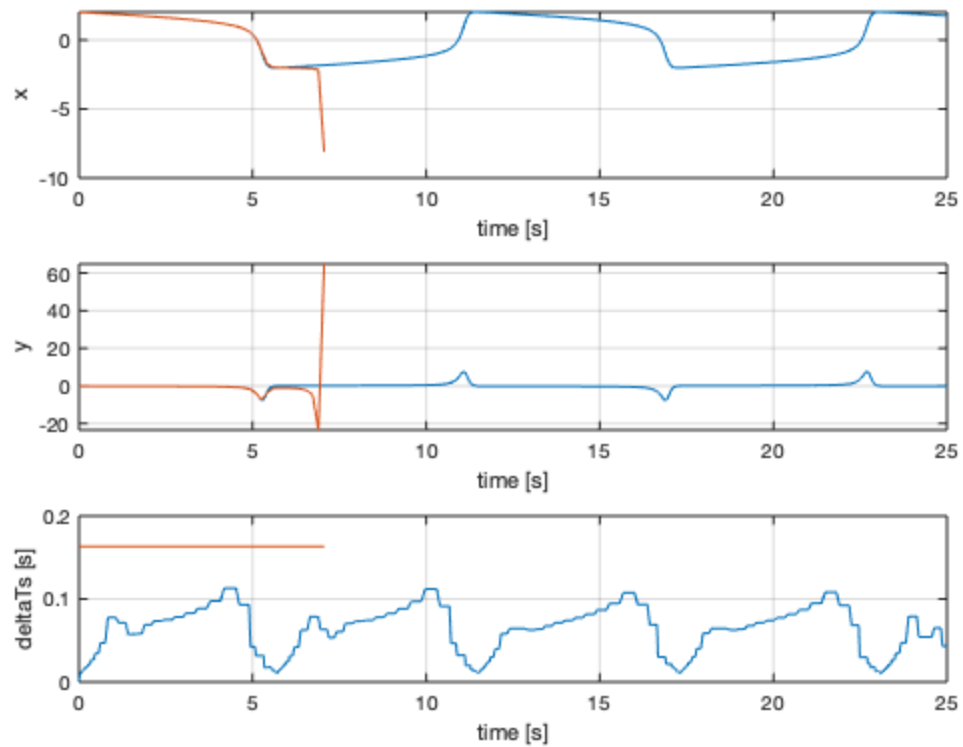
```
figure (12)
subplot(311)
plot(t,z(:,1))
hold on
plot(T,X3(1,:))
grid on
xlabel('time [s]'), ylabel('x')

subplot(312)
plot(t,z(:,2))
hold on
plot(T,X3(2,:))
grid on
xlabel('time [s]'), ylabel('y')

deltaTs2 = deltaT * ones(length(T),1);
subplot(313)
plot(t,deltaTs)
hold on
plot(T, deltaTs2)
grid on
xlabel('time [s]'), ylabel('deltaTs [s]')

% Simulation parameters
%deltaT = 0.1635;
totalT = 7.065;
deltaT = 0.1633987;
%deltaT = 0.02;
%totalT = 25;
T = linspace(0,totalT, 1 + totalT/deltaT);

% Simulate
%disp('Starting ERK4');
Nx = length(z0);
Nstage = length(A);
Nt = length(T);
X3 = ERK_vdp(HeunArray,f_vanderpol,T,z0,Nx,Nstage,Nt);
```



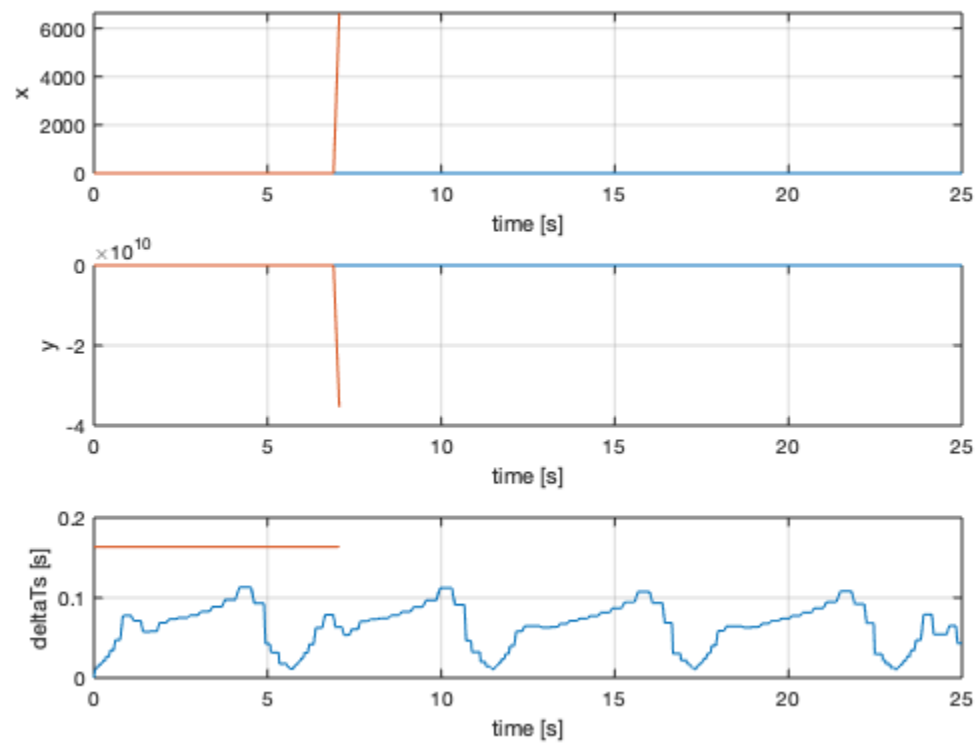
Plotting 3b, $\Delta T = 0.1633987$ (NOTE: Sim to 7.065 seconds)

```
figure (13)
subplot(311)
plot(t,z(:,1))
hold on
plot(T,X3(1,:))
grid on
xlabel('time [s]'), ylabel('x')

subplot(312)
plot(t,z(:,2))
hold on
plot(T,X3(2,:))
grid on
xlabel('time [s]'), ylabel('y')

deltaTs2 = deltaT * ones(length(T),1);
subplot(313)
plot(t,deltaTs)
hold on
plot(T, deltaTs2)
grid on
```

```
xlabel('time [s]'), ylabel('deltaTs [s]')
```



Published with MATLAB® R2019b