Table of Contents

CASE I

```
close all
%setting initial condition
A = [-5.0];
B = [2.0];
C = [1.0];
D = [0];
t = 0.1;
IC = 1;
U = 0;
sys = ss(A,B,C,D);
[Yc, Tc] = initial(sys, IC, 0:t:2);

% State Discretition
Ad = expm(-5*t);
Bd = ((-2/5)*expm(-5*t)) + (2/5);
yd = [];
yd(1) = 1;
Td = [];
Td(1) = 0;
% Exact Discrete
for i = 1:length(Tc)
    yd(i+1) = Ad*yd(i) +(Bd*U);
    Td(i+1) = i; % save corresponding discrete time integer index
end
```

Part 1

```
fprintf("After discretization Ad = expm(-5*t) Bd = ((-2/5)*expm(-5*t))
 + (2/5)")
fprintf("State Equation: X[k+1] = Ad*X[k] + Bd*U[k]")
fprintf("Y[k] = X[k]")
```

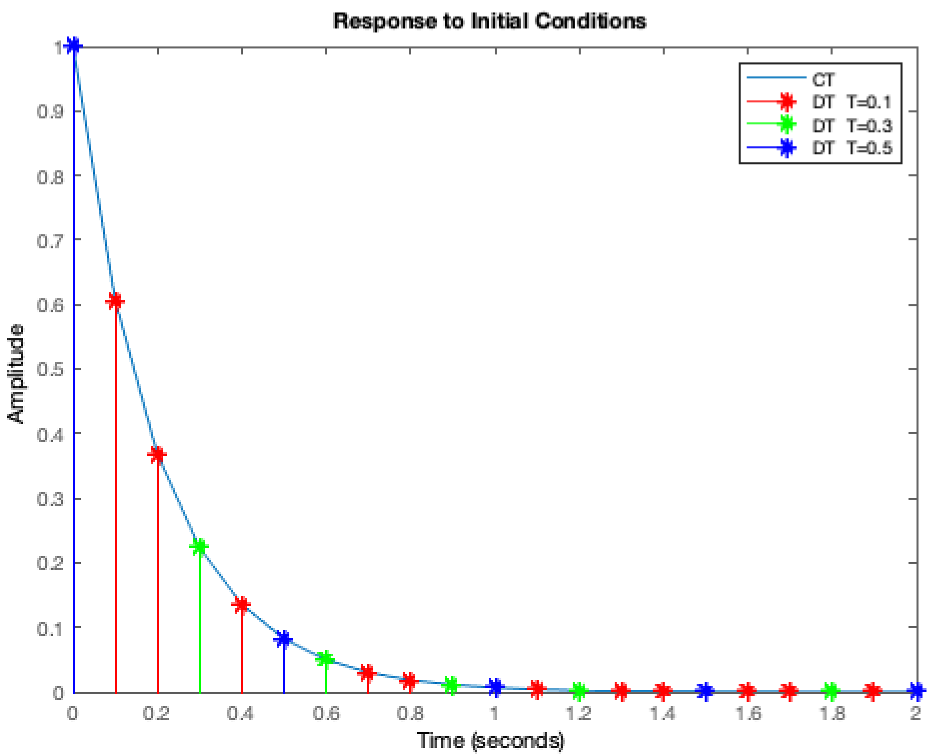
*After discretization Ad = expm(-5*t) Bd = ((-2/5)*expm(-5*t)) + (2/5)State Equation: X[k+1] = Ad*X[k] + Bd*U[k]Y[k] = X[k]*

Part 2

```
figure
hold on
initial(sys, IC, 0:0.1:2)

yd = [];
yd(1) = 1;
Td = [];
Td(1) = 0;
% Exact Discrete
t = 0.1;
Ad = expm(-5*t);
Bd = ((-2/5)*expm(-5*t)) + (2/5);
for i = 1:length(Tc)
    yd(i+1) = Ad*yd(i) +(Bd*U);
    Td(i+1) = i; % save corresponding discrete time integer index
end
ed = stem(Td*t,yd, '*', 'DisplayName', 'Exact Discrete');
ed.Color = 'red';
t = 0.3;
Ad = expm(-5*t);
Bd = ((-2/5)*expm(-5*t)) + (2/5);
for i = 1:length(Tc)
    yd(i+1) = Ad*yd(i) +(Bd*U);
    Td(i+1) = i; % save corresponding discrete time integer index
end
ed = stem(Td*t,yd, '*', 'DisplayName', 'Exact Discrete');
ed.Color = 'green';
t = 0.5;
Ad = expm(-5*t);
Bd = ((-2/5)*expm(-5*t)) + (2/5);
for i = 1:length(Tc)
    yd(i+1) = Ad*yd(i) +(Bd*U);
    Td(i+1) = i; % save corresponding discrete time integer index
end
ed = stem(Td*t,yd, '*', 'DisplayName', 'Exact Discrete');
ed.Color = 'blue';
legend('CT','DT  T=0.1',  'DT  T=0.3', 'DT  T=0.5')

hold off
fprintf("\nThe continous time signal is stable because as t approaches
 infinity, the amplitude approaches 0")
```

```
fprintf("\nThe discrete time signal is stabel for any time T > 0 can
 be seen from the graph as well for T = 0.1, 0.3, 0.5, the amplitude
 also approaches 0")
```

```
The continous time signal is stable because as t approaches infinity,
 the amplitude approaches 0
The discrete time signal is stabel for any time T > 0 can be seen
 from the graph as well for T = 0.1, 0.3, 0.5, the amplitude also
 approaches 0
```



Part 3a

```
figure
fprintf("\nClose form continuous signal is x(t) = e^(A*t) * x(0) \n
 y(t) = x(t)")
I = imread('d.jpg');
imshow(I)
```

```
Close form continuous signal is x(t) = e^(A*t) * x(0)
 y(t) = x(t)Warning: Image is too big to fit on screen; displaying at
 50%
```

$$x[k] = A_d^{k} x[0] + \sum_{m=0}^{k-1} A_d^{k-1-m} B_d \, u[m]$$

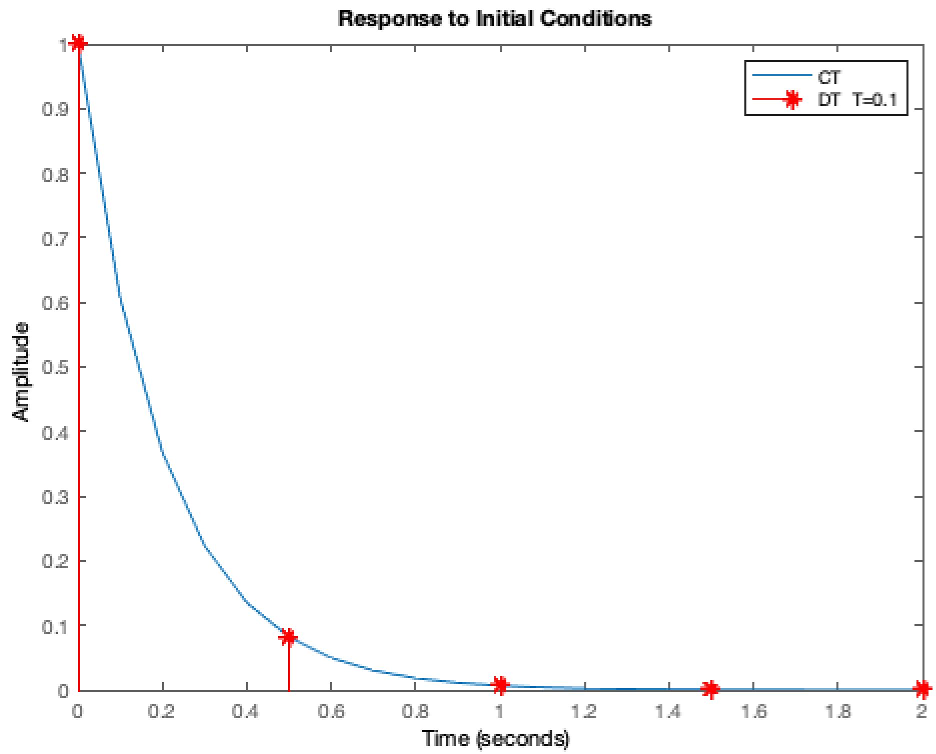
$$y[k] = x[k]$$

where $A_d = e^{-5t}$

$$B_d = -2/5 \exp^{-5t} + 2/5$$

Part 3b

```
figure
hold on
initial(sys, IC, 0:0.1:2)
ed = stem(Td*t,yd, '*', 'DisplayName', 'Exact Discrete');
ed.Color = 'red';
legend('CT','DT  T=0.1')
```

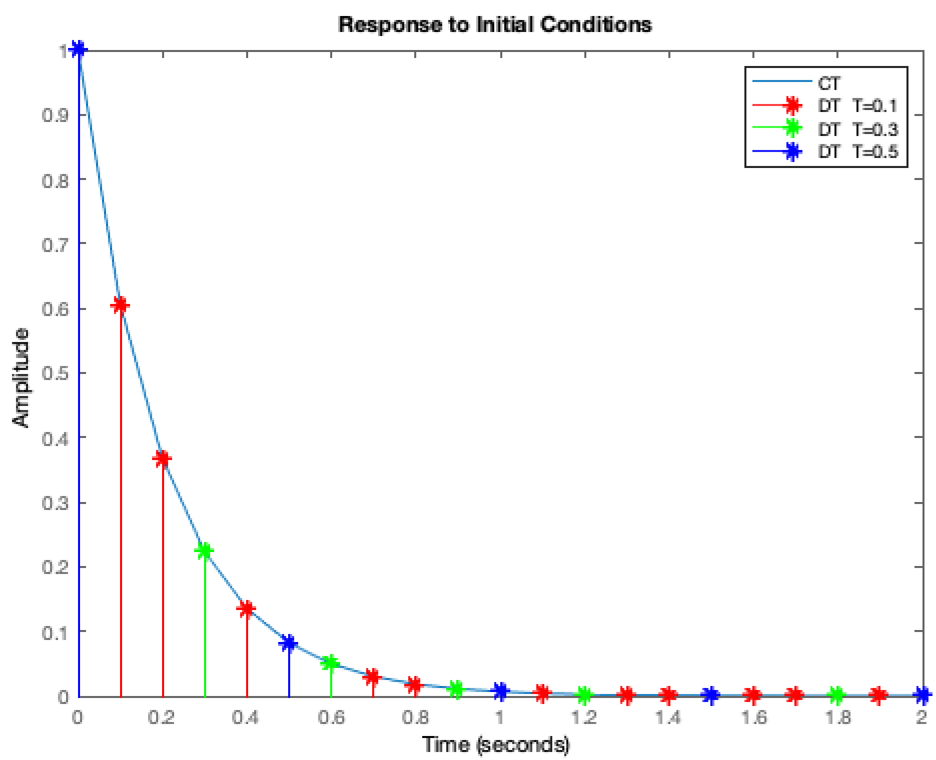


Part 3c

```
figure
hold on
initial(sys, IC, 0:0.1:2)

yd = [];
yd(1) = 1;
Td = [];
Td(1) = 0;
% Exact Discrete
t = 0.1;
Ad = expm(-5*t);
Bd = ((-2/5)*expm(-5*t)) + (2/5);
for i = 1:length(Tc)
    yd(i+1) = Ad*yd(i) +(Bd*U);
    Td(i+1) = i; % save corresponding discrete time integer index
end
ed = stem(Td*t,yd, '*', 'DisplayName', 'Exact Discrete');
ed.Color = 'red';
t = 0.3;
Ad = expm(-5*t);
Bd = ((-2/5)*expm(-5*t)) + (2/5);
for i = 1:length(Tc)
    yd(i+1) = Ad*yd(i) +(Bd*U);
```

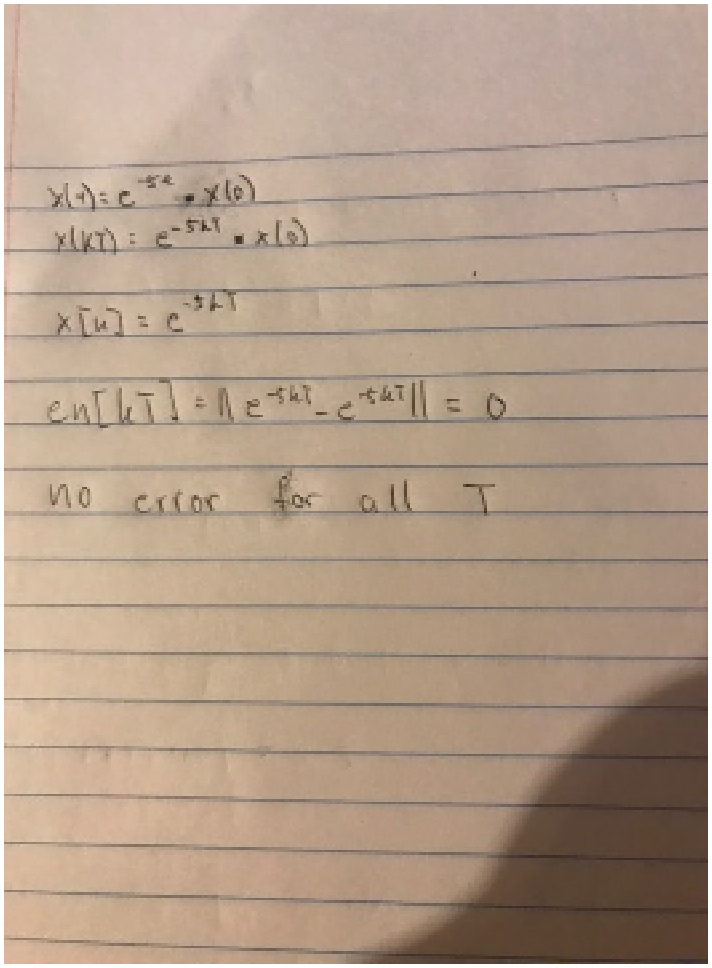
```
    Td(i+1) = i; % save corresponding discrete time integer index
end
ed = stem(Td*t,yd, '*', 'DisplayName', 'Exact Discrete');
ed.Color = 'green';
t = 0.5;
Ad = expm(-5*t);
Bd = ((-2/5)*expm(-5*t)) + (2/5);
for i = 1:length(Tc)
    yd(i+1) = Ad*yd(i) +(Bd*U);
    Td(i+1) = i; % save corresponding discrete time integer index
end
ed = stem(Td*t,yd, '*', 'DisplayName', 'Exact Discrete');
ed.Color = 'blue';
legend('CT','DT  T=0.1',  'DT  T=0.3', 'DT  T=0.5')

hold off
fprintf("\nWhen T increases the discrete time signal matches better
 with continous time. When T decreases the DT signal loses alot of
 information but in general the discrete time signal is stable for all
 T")
```

```
When T increases the discrete time signal matches better with
 continous time. When T decreases the DT signal loses alot of
 information but in general the discrete time signal is stable for all
 T
```



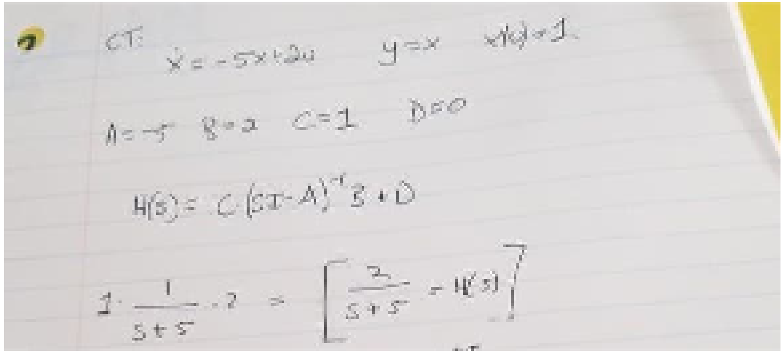
Part 3d

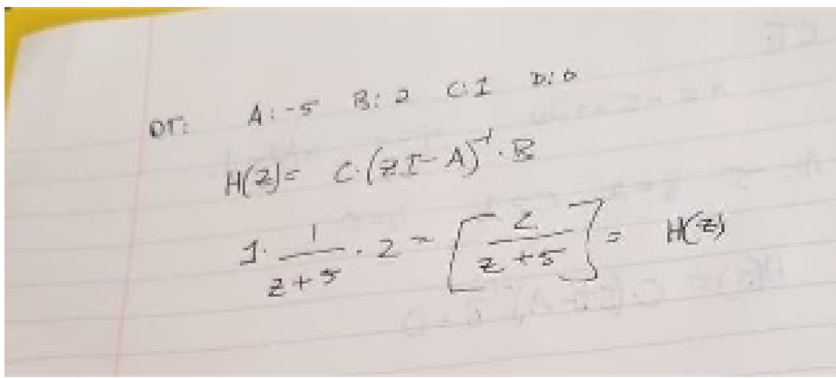
```
figure
I = imread('case1err.jpg');
imshow(I)
```



Part 4a

```
figure
I = imread('case1part4ct.jpg');
imshow(I)
figure
I = imread('case1part4dt.jpg');
imshow(I)
```

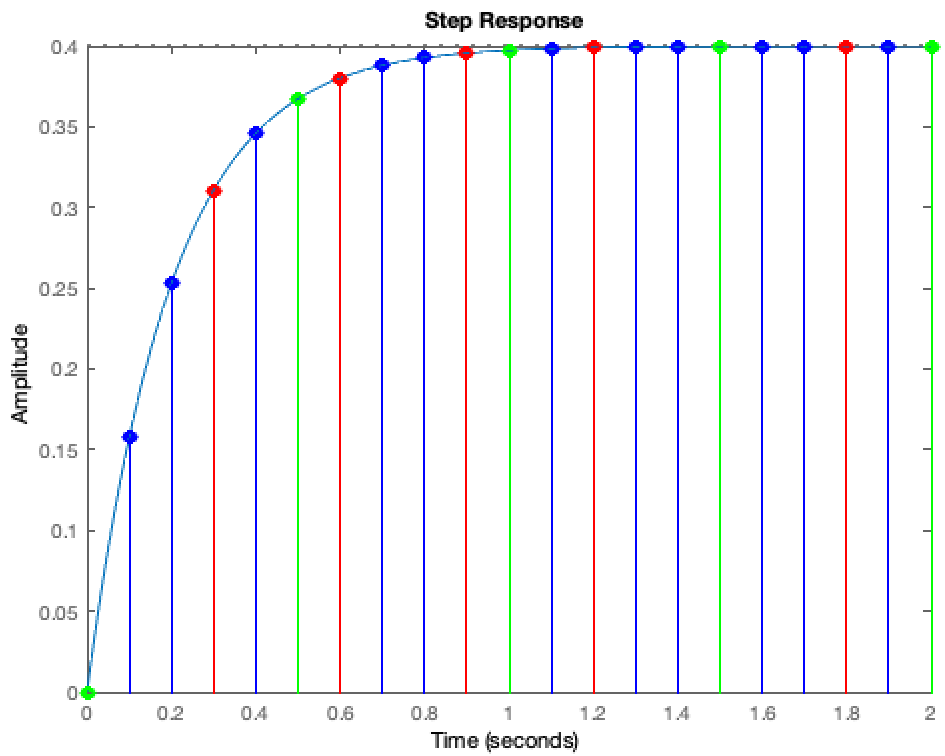




Part 4b

```
figure
step(sys, 0:0.01:2);
hold on;
U = 1;
yd(1) = 0;
t = 0.1;
Ad = expm(-5*t);
Bd = ((-2/5)*expm(-5*t)) + (2/5);
for i = 1:length(Tc)
    yd(i+1) = Ad*yd(i) +(Bd*U);
    Td(i+1) = i; % save corresponding discrete time integer index
end
ed = stem(Td*t,yd, 'filled', 'DisplayName', 'Exact Discrete');
ed.Color = 'blue';
t = 0.3;
Ad = expm(-5*t);
```

```
Bd = ((-2/5)*expm(-5*t)) + (2/5);
for i = 1:length(Tc)
    yd(i+1) = Ad*yd(i) +(Bd*U);
    Td(i+1) = i; % save corresponding discrete time integer index
end
ed = stem(Td*t,yd, 'filled', 'DisplayName', 'Exact Discrete');
ed.Color = 'red';
t = 0.5;
Ad = expm(-5*t);
Bd = ((-2/5)*expm(-5*t)) + (2/5);
for i = 1:length(Tc)
    yd(i+1) = Ad*yd(i) +(Bd*U);
    Td(i+1) = i; % save corresponding discrete time integer index
end
ed = stem(Td*t,yd, 'filled', 'DisplayName', 'Exact Discrete');
ed.Color = 'green';
```

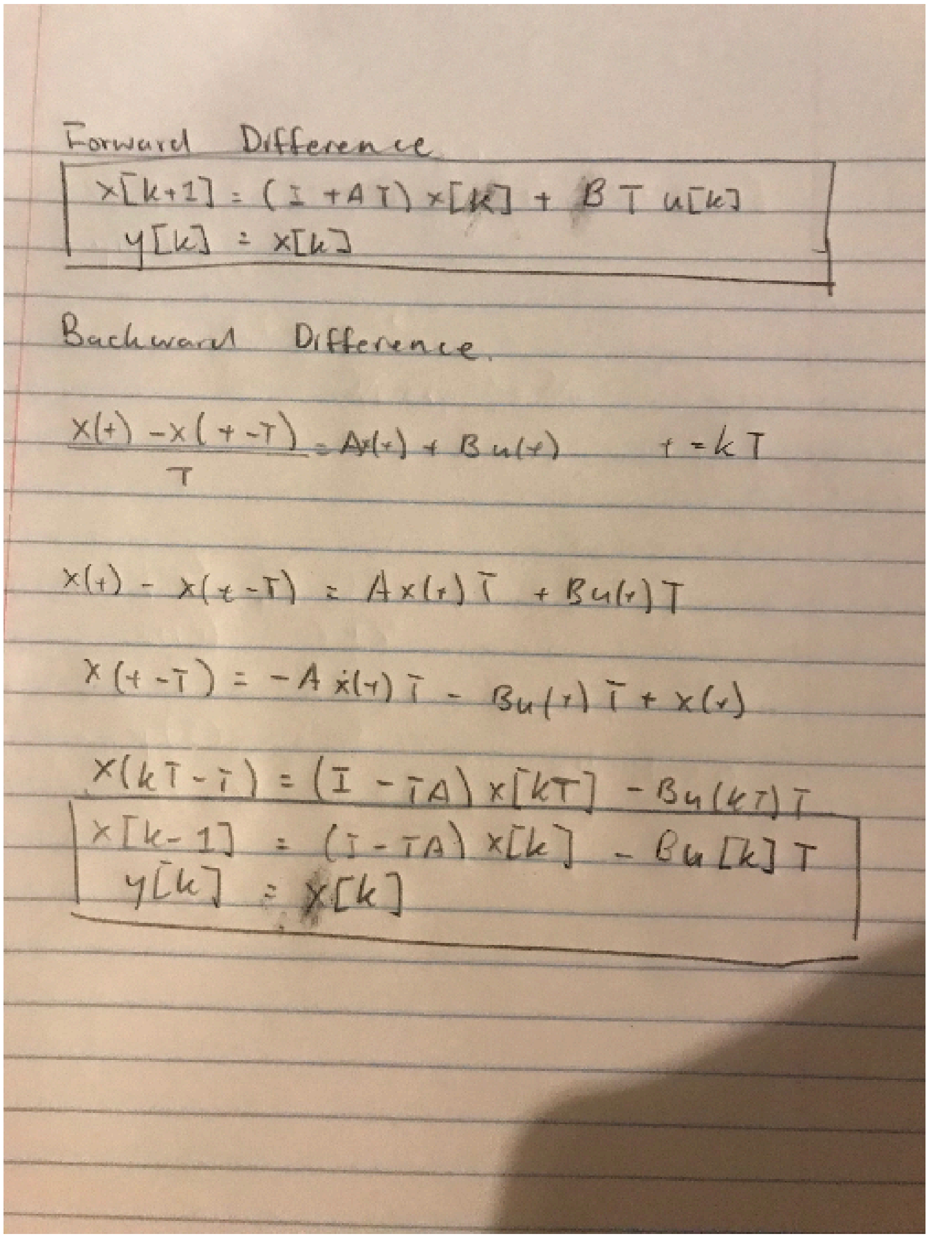


Case II

Part 1

```
figure
I = imread('case2part1.jpg');
imshow(I)
```

Warning: Image is too big to fit on screen; displaying at 33%



Forward Difference

$$\boxed{\begin{aligned} x[k+1] &= (I + AT)\,x[k] + BT\,u[k] \\ y[k] &= x[k] \end{aligned}}$$

Backward Difference

$$\frac{x(t) - x(t-T)}{T} = Ax(t) + Bu(t) \qquad t = kT$$

$$x(t) - x(t-T) = Ax(t)T + Bu(t)T$$

$$x(t-T) = -A\dot{x}(t)T - Bu(t)T + x(t)$$

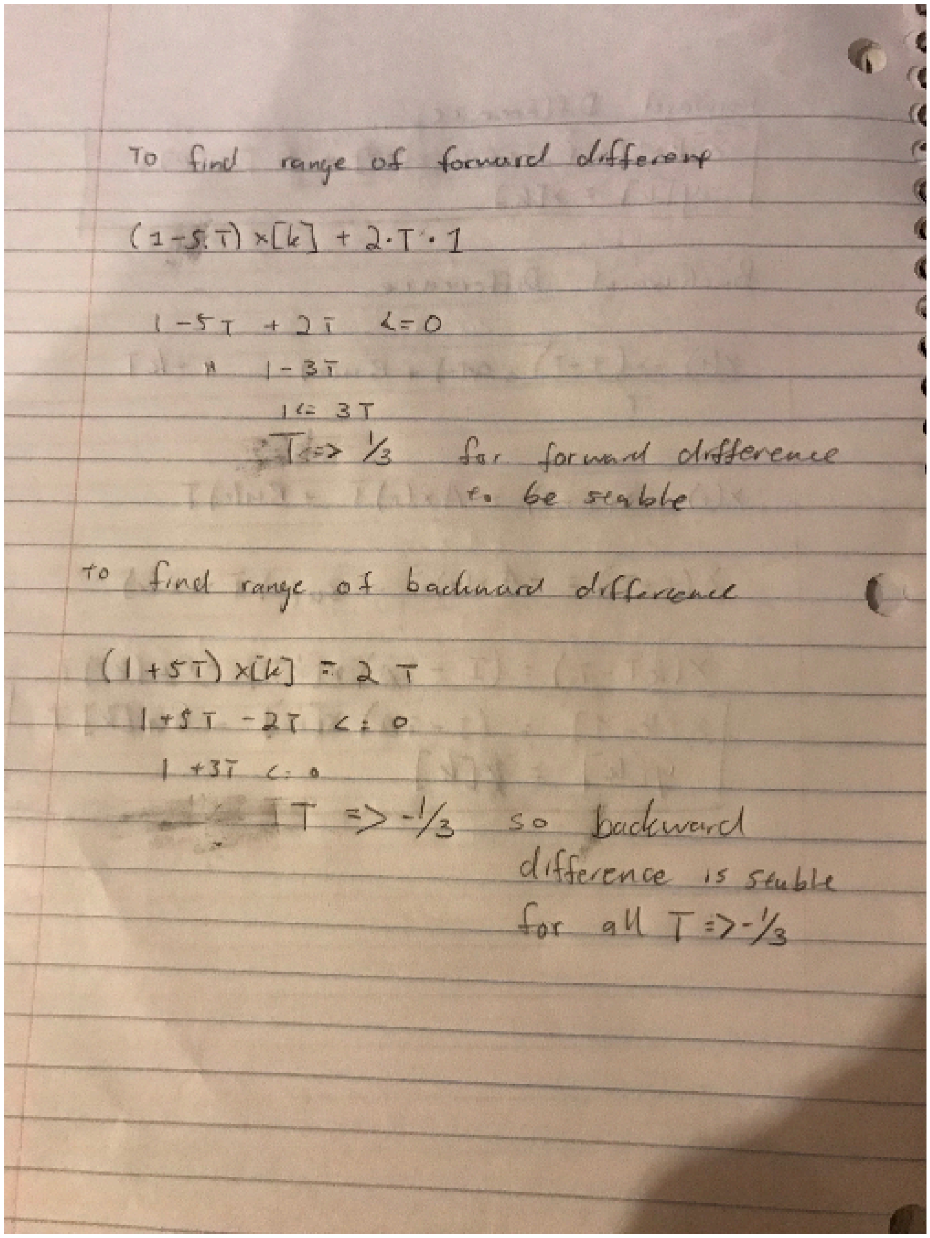
$$x(kT-T) = (I - TA)\,x[kT] - Bu(kT)T$$

$$\boxed{\begin{aligned} x[k-1] &= (I - TA)\,x[k] - Bu[k]T \\ y[k] &= x[k] \end{aligned}}$$

Part 2

```
figure
I = imread('case2part2.jpg');
imshow(I)
```

Warning: Image is too big to fit on screen; displaying at 33%



To find range of forward difference

$(1-5T)\,x[k] + 2\cdot T\cdot 1$

$1 - 5T + 2T <= 0$

$1 - 3T$

$1 <= 3T$

$T => 1/3$ for forward difference to be stable

to find range of backward difference

$(1+5T)\,x[k] = 2T$

$1 + 5T - 2T <= 0$

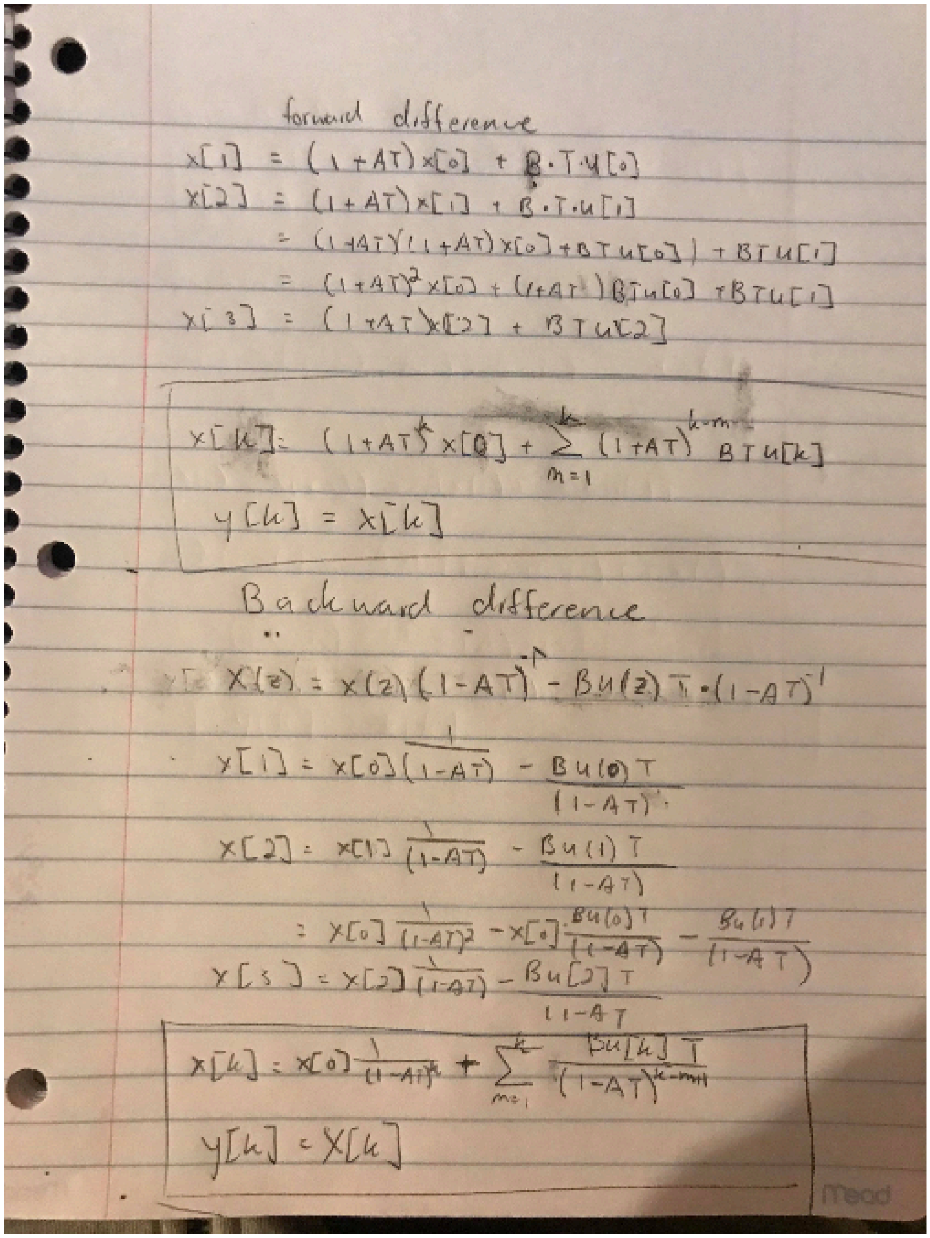
$1 + 3T <= 0$

$T => -1/3$ so backward difference is stable for all $T => -1/3$

Part 3a

```
figure
I = imread('case2part3a.jpg');
imshow(I)
```

Warning: Image is too big to fit on screen; displaying at 33%


forward difference

$x[1] = (1+AT)x[0] + B \cdot T \cdot u[0]$

$x[2] = (1+AT)x[1] + B \cdot T \cdot u[1]$

$= (1+AT)((1+AT)x[0] + BTu[0]) + BTu[1]$

$= (1+AT)^2 x[0] + (1+AT)BTu[0] + BTu[1]$

$x[3] = (1+AT)x[2] + BTu[2]$

$$\boxed{x[k] = (1+AT)^k x[0] + \sum_{m=1}^{k} (1+AT)^{k-m} BTu[k] \qquad y[k] = x[k]}$$

Backward difference

$X(z) = x(z)(1-AT)^{-1} - Bu(z)T \cdot (1-AT)^{-1}$

$x[1] = x[0]\frac{1}{(1-AT)} - \frac{Bu(0)T}{(1-AT)}$

$x[2] = x[1]\frac{1}{(1-AT)} - \frac{Bu(1)T}{(1-AT)}$

$= x[0]\frac{1}{(1-AT)^2} - x[0]\frac{Bu(0)T}{(1-AT)} - \frac{Bu(1)T}{(1-AT)}$

$x[3] = x[2]\frac{1}{(1-AT)} - \frac{Bu[2]T}{1-AT}$

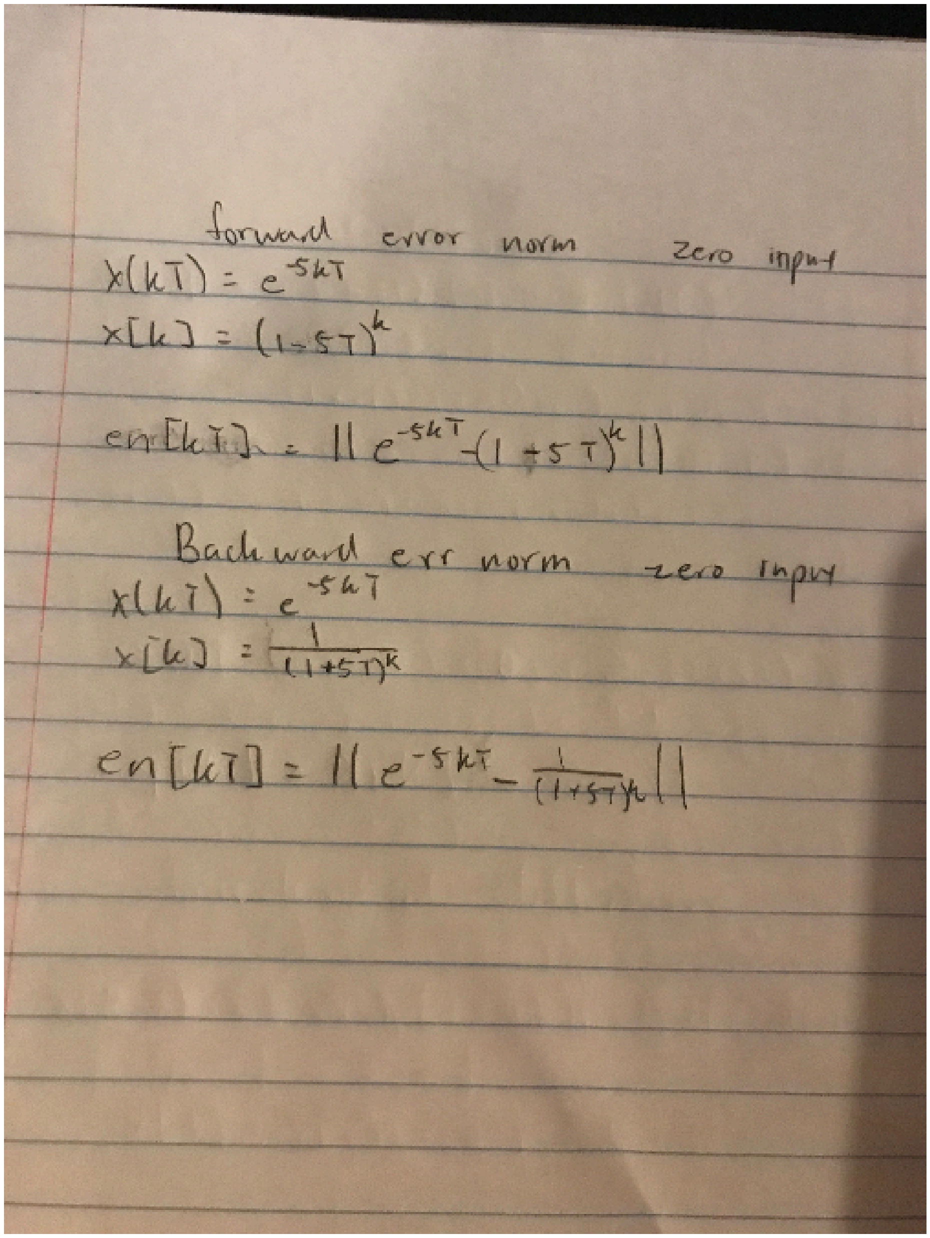
$$\boxed{x[k] = x[0]\frac{1}{(1-AT)^k} + \sum_{m=1}^{k} \frac{Bu[k]T}{(1-AT)^{k-m+1}} \qquad y[k] = X[k]}$$

Part 3b

```
figure
I = imread('case2part3b.jpg');
imshow(I)
fprintf("within the range of stability, samller the period, the higher
 the lower the error between the estimates\n")
fprintf("this is because as you sample more frequently the estimates
 gets better matched to the original signal\n")
```

```
fprintf("Larger the K value, more time for the method to converge
 hence lowering the error")
fprintf("The result for this is different comapred to Case 1 part 3d
 because 3d was an exact solution, while this is an estimate, so 3d
 has no error component, while forward and backward methods do")
fprintf("\n see next part for graph")
```

```
Warning: Image is too big to fit on screen; displaying at 33%
within the range of stability, samller the period, the higher the
 lower the error between the estimates
this is because as you sample more frequently the estimates gets
 better matched to the original signal
Larger the K value, more time for the method to converge hence
 lowering the errorThe result for this is different comapred to Case 1
 part 3d because 3d was an exact solution, while this is an estimate,
 so 3d has no error component, while forward and backward methods do
 see next part for graph
```



forward error norm zero input

$x(kT) = e^{-5kT}$

$x[k] = (1-5T)^k$

$en[kT] = || e^{-5kT} - (1+5T)^k ||$

Backward err norm zero input

$x(kT) = e^{-5kT}$

$x[k] = \frac{1}{(1+5T)^k}$

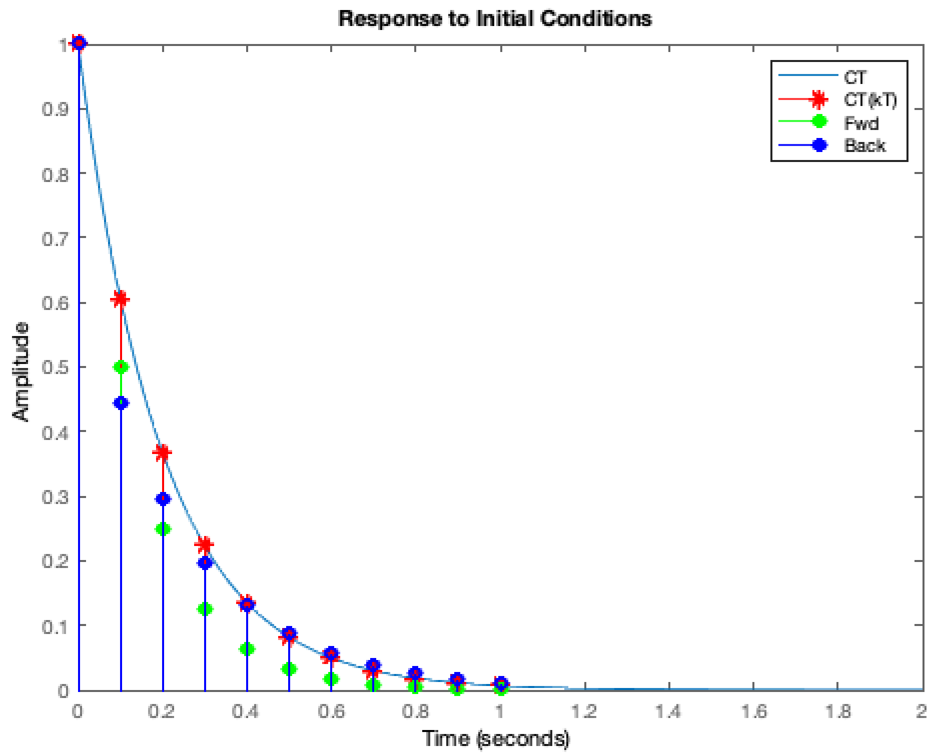
$en[kT] = || e^{-5kT} - \frac{1}{(1+5T)^k} ||$

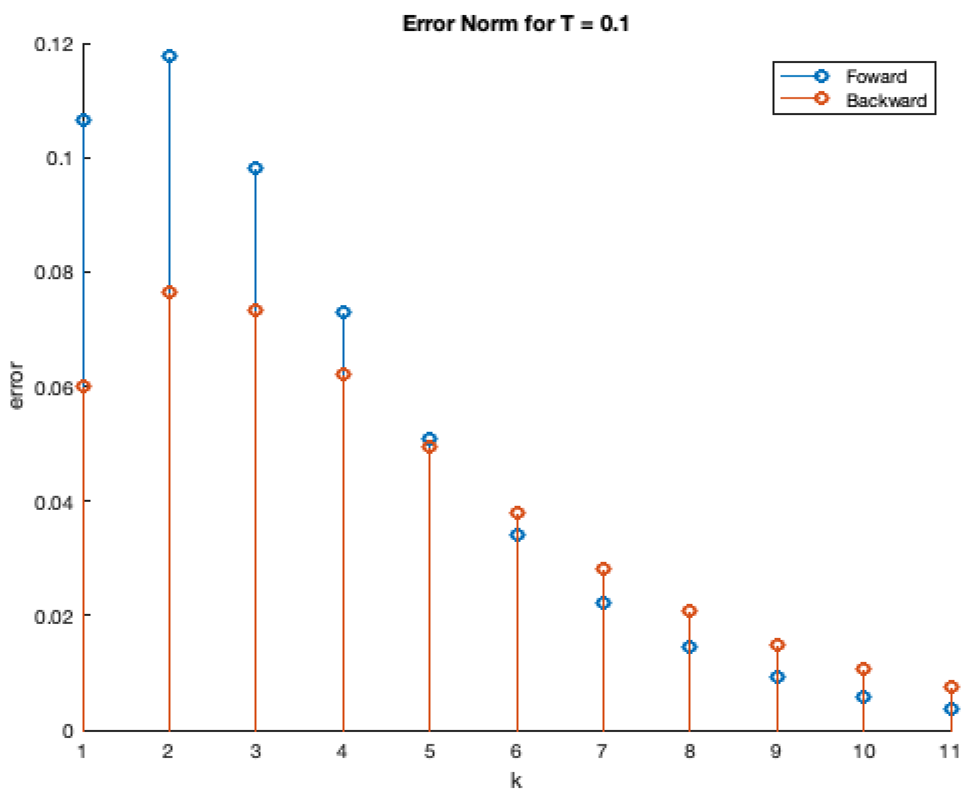
Part 3c

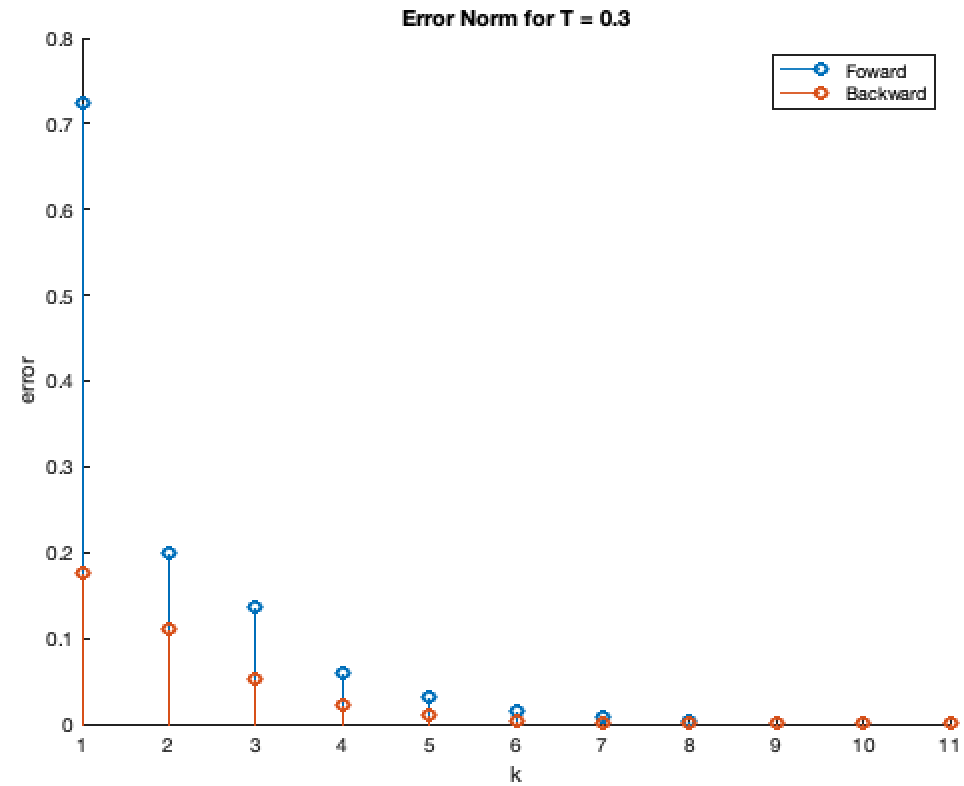
```
close all
A = [-5.0];
B = [2.0];
C = [1.0];
D = [];
t = .1;
IC = 1;
U = 0;
```

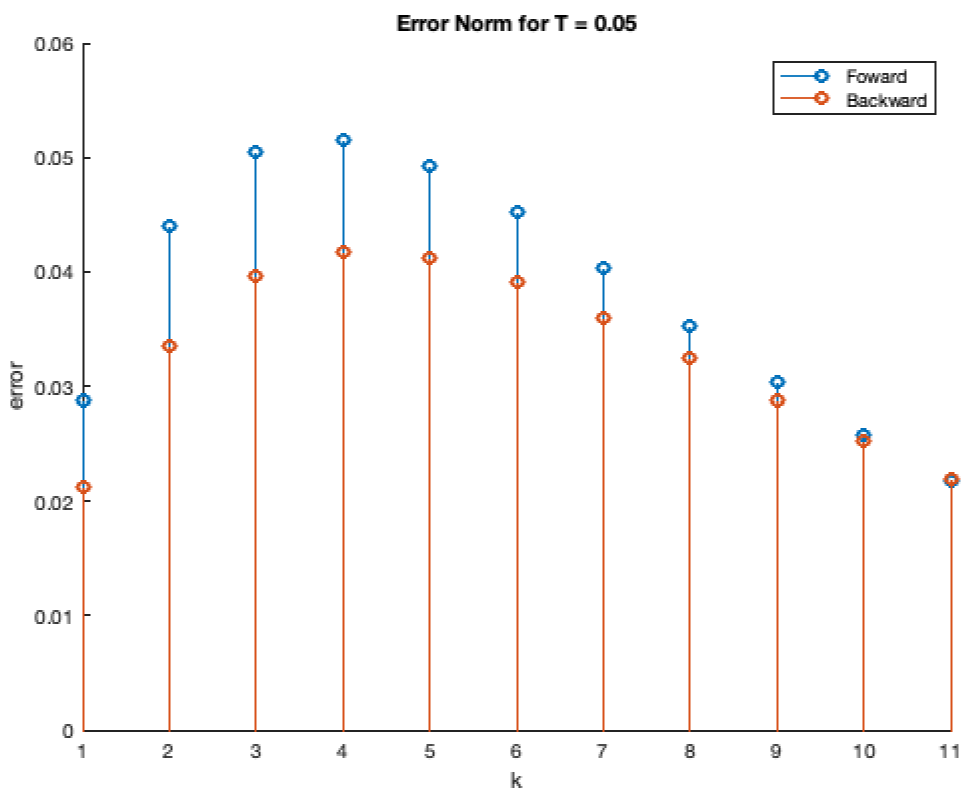
```
% Forward Difference
Ad = expm(-5*t);
Bd = ((-2/5)*expm(-5*t)) + (2/5);

sys = ss(A,B,C,D);
[Yc, Tc] = initial(sys, IC, 0:0.01:2);
hold on;
initial(sys, IC, 0:0.01:2)
ybd = [];
yd = [];
yd(1) = 1;
Td = [];
Td(1) = 0;
% Exact Discrete
for i = 1:length(Tc)
    yd(i+1) = Ad*yd(i) +(Bd*U);
    Td(i+1) = i; % save corresponding discrete time integer index
end
hold on;
ed = stem(Td(1:11)*t,yd(1:11), '*', 'DisplayName', 'Exact Discrete');
ed.Color = 'red';

yfd = [];
yfd(1) = 1;
Td = [];
Td(1) = 0;
% Forward Difference
for i = 1:length(Tc)
    yfd(i+1) = (1+(t*A))*yfd(i) +(B*t*U);
    Td(i+1) = i;
end
hold on;
fd = stem(Td(1:11)*t,yfd(1:11), 'filled','DisplayName', 'Forward
 Difference');
fd.Color = 'green';
ybd = zeros(1, length(Td));
ybd(1) = 1;
% Backward Difference
for k = 2:length(Td)
    z = 0;
    for m = 1:k
        z = z + 1/(1-A*t)^(k-m+1) * (B*t*U);
    end
    ybd(k) = ybd(1)*(1/((1-A*t)^k)) + z;
end
bd = stem(Td(1:11)*t,ybd(1:11), 'filled','DisplayName', 'Backward
 Difference');
bd.Color = 'blue';
legend('CT','CT(kT)','Fwd','Back')

en_forward = [];
en_backward = [];
t = 0.1;
```

```
for k = 1:11
    en_forward(k) = abs(exp(-5*k*t) - (1-5*t)^k);
    en_backward(k) = abs(exp(-5*k*t) - (1/(1+5*t)^k));
end
k = 1:11;
figure
hold on
stem(k, en_forward)
stem(k, en_backward)
title("Error Norm for T = 0.1")
xlabel("k")
ylabel("error")
legend("Foward", "Backward")
t = 0.3;
for k = 1:11
    en_forward(k) = abs(exp(-5*k*t) - (1-5*t)^k);
    en_backward(k) = abs(exp(-5*k*t) - (1/(1+5*t)^k));
end
k = 1:11;
figure
hold on
stem(k, en_forward)
stem(k, en_backward)
title("Error Norm for T = 0.3")
xlabel("k")
ylabel("error")
legend("Foward", "Backward")
t = 0.05;
for k = 1:11
    en_forward(k) = abs(exp(-5*k*t) - (1-5*t)^k);
    en_backward(k) = abs(exp(-5*k*t) - (1/(1+5*t)^k));
end
k = 1:11;
figure
hold on
stem(k, en_forward)
stem(k, en_backward)
title("Error Norm for T = 0.05")
xlabel("k")
ylabel("error")
legend("Foward", "Backward")
```


Response to Initial Conditions
Amplitude
Time (seconds)
CT
CT(kT)
Fwd
Back
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2

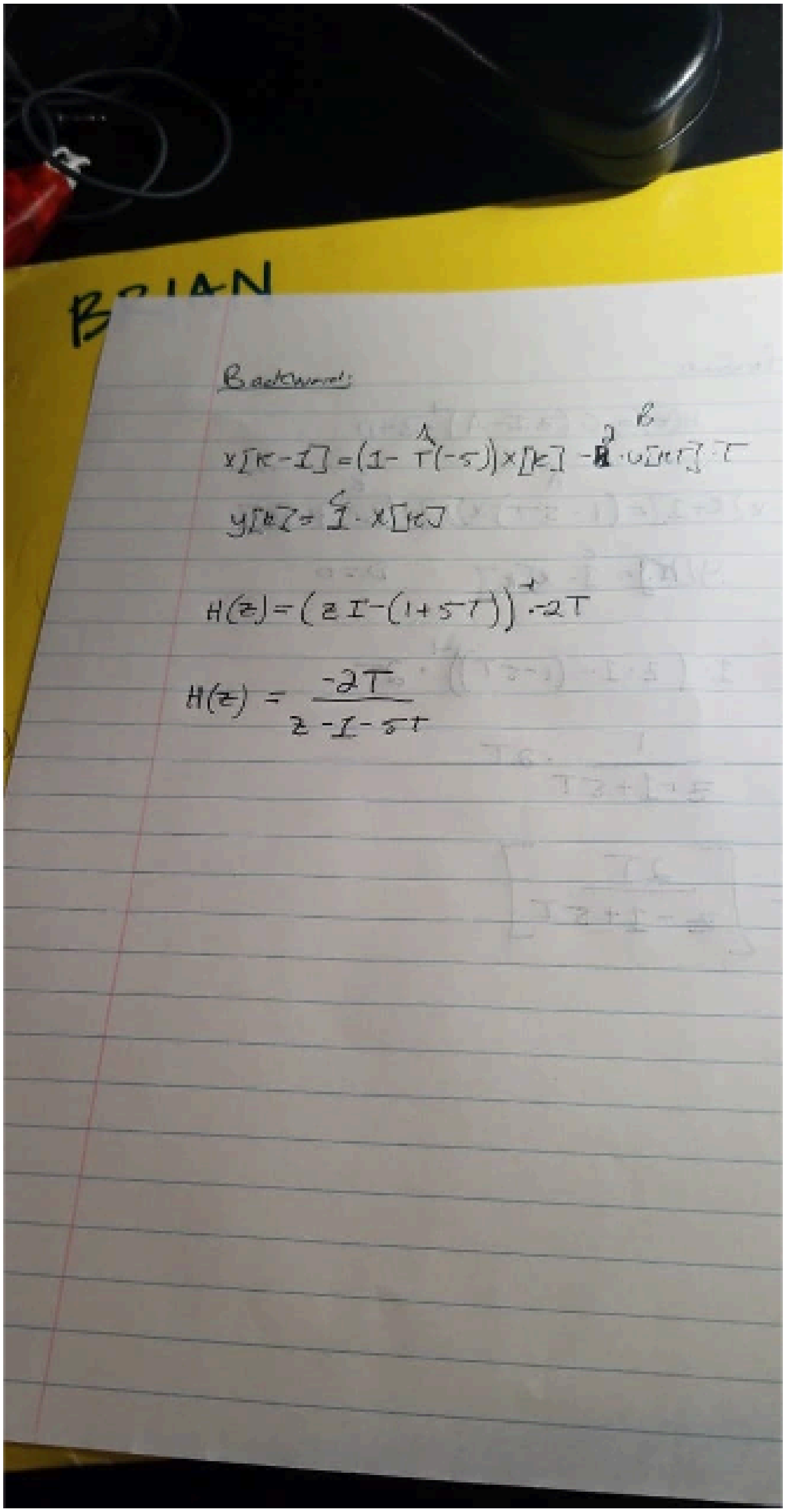

Error Norm for T = 0.1
error
k
Foward
Backward
0
0.02
0.04
0.06
0.08
0.1
0.12
1
2
3
4
5
6
7
8
9
10
11

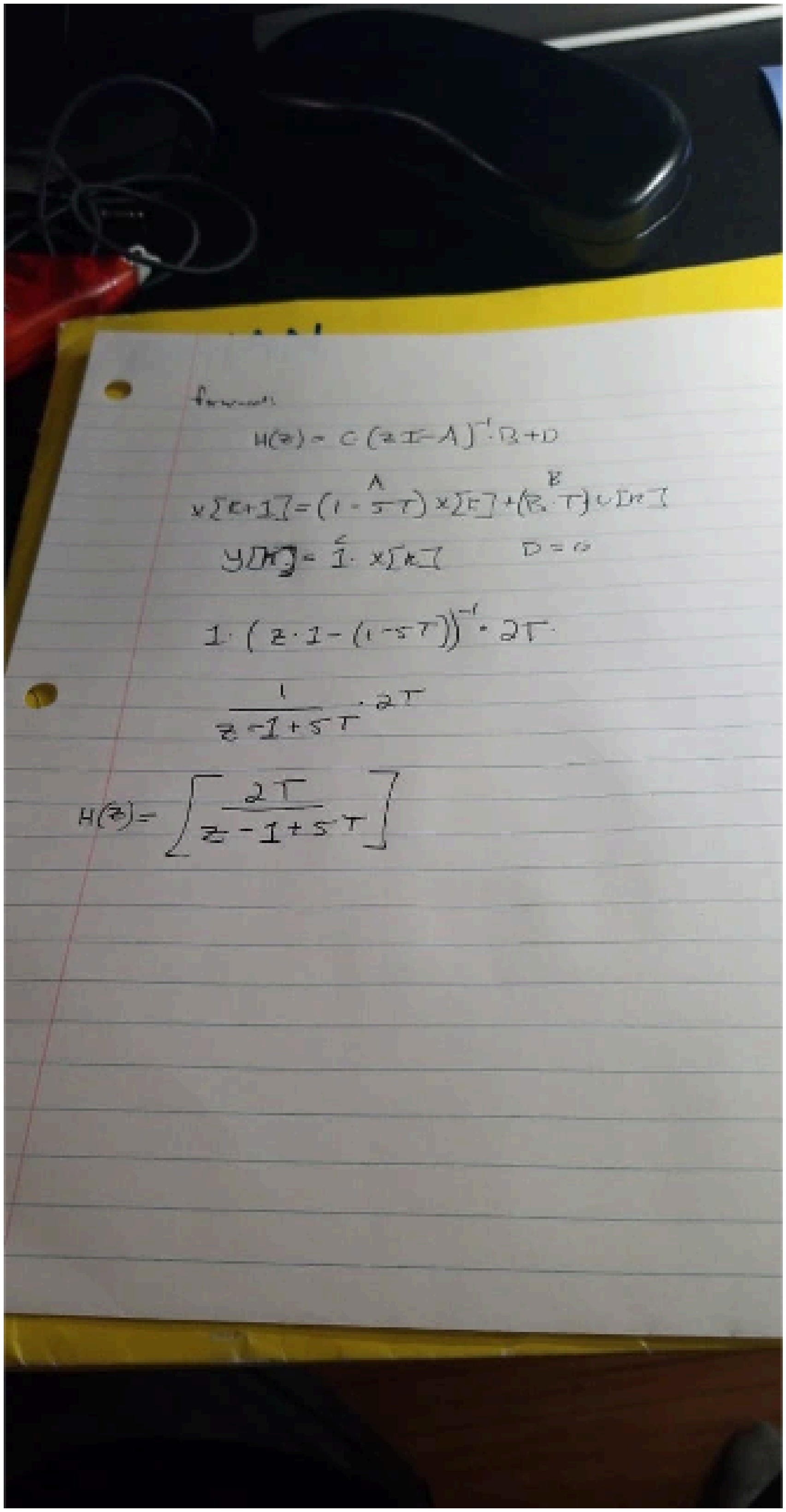

Error Norm for T = 0.3
Foward
Backward
error
k
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
1
2
3
4
5
6
7
8
9
10
11


Error Norm for T = 0.05
Foward
Backward
error
k
0
0.01
0.02
0.03
0.04
0.05
0.06
1
2
3
4
5
6
7
8
9
10
11

Part 4a

```
figure
I = imread('case2part4af.jpg');
imshow(I)
figure
I = imread('case2part4ab.jpg');
imshow(I)
```


BRIAN
Backward:
$x[k-1] = (1 - \overset{A}{T}(-5))\,x[k] - \overset{B}{2}\cdot u[k-1]\cdot T$
$y[k] = \overset{C}{1}\cdot x[k]$
$H(z) = (zI - (1+5T))^{-1}\cdot -2T$
$H(z) = \frac{-2T}{z - 1 - 5T}$



forward:

$$H(z) = C\,(zI - A)^{-1} \cdot B + D$$

$$x[k+1] = \overset{A}{(1 - 5T)}\, x[k] + \overset{B}{(2 \cdot T)}\, u[k]$$

$$y[k] = \overset{C}{1} \cdot x[k] \qquad D = 0$$

$$1 \cdot \left(z \cdot 1 - (1 - 5T)\right)^{-1} \cdot 2T$$

$$\frac{1}{z - 1 + 5T} \cdot 2T$$

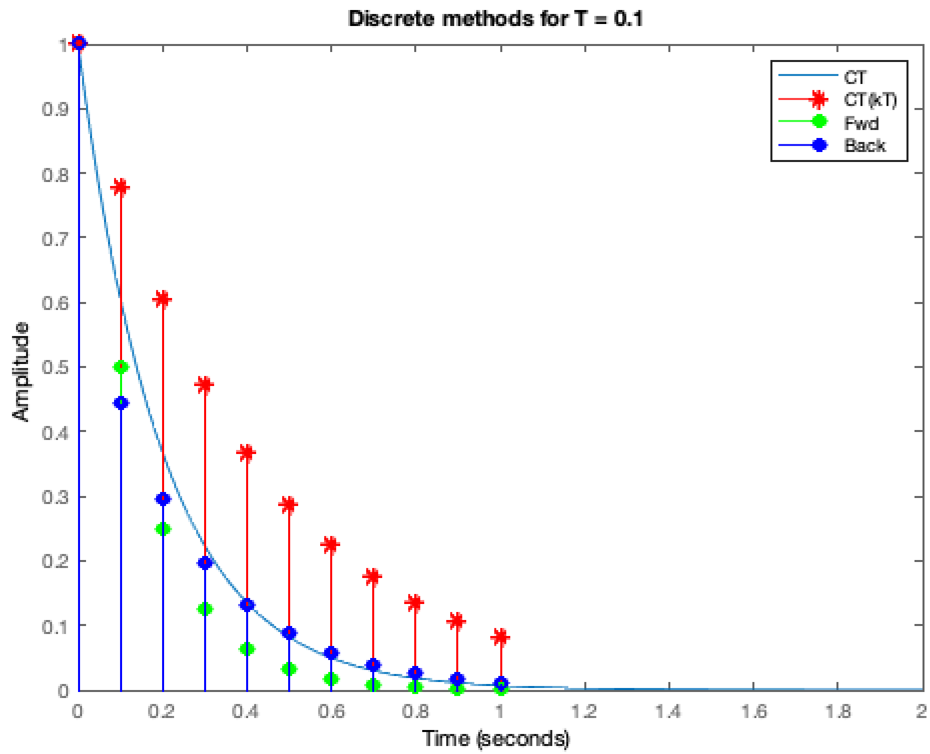
$$H(z) = \left[\frac{2T}{z - 1 + 5T}\right]$$

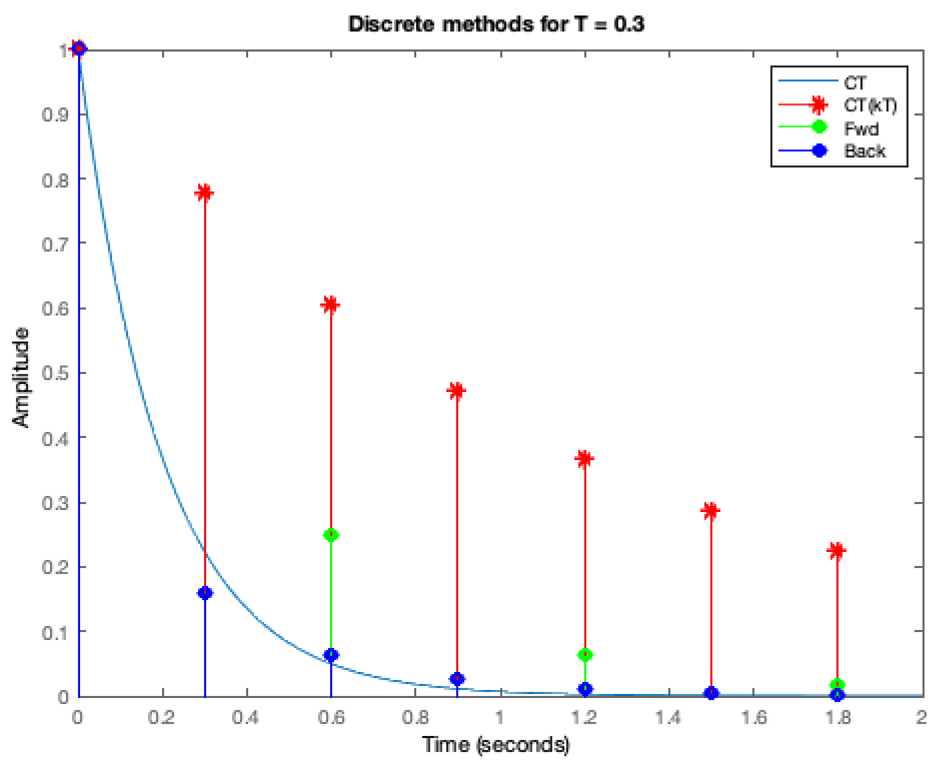
Part 4b1

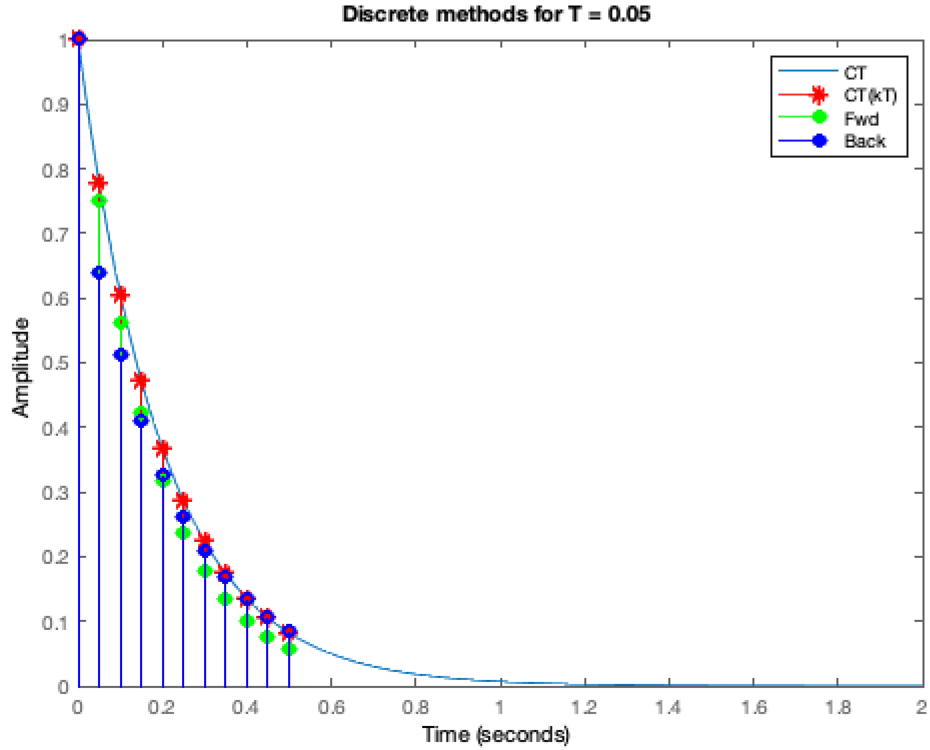
```
close all
A = [-5.0];
B = [2.0];
C = [1.0];
D = [];
IC = 1;
U = 0;
figure
% Exact Difference
Ad = expm(-5*t);
Bd = ((-2/5)*expm(-5*t)) + (2/5);

sys = ss(A,B,C,D);
[Yc, Tc] = initial(sys, IC, 0:0.01:2);
hold on;
initial(sys, IC, 0:0.01:2)
ybd = [];
yd = [];
yd(1) = 1;
Td = [];
Td(1) = 0;
% Exact Discrete
t = .1;
for i = 1:length(Tc)
    yd(i+1) = Ad*yd(i) +(Bd*U);
    Td(i+1) = i; % save corresponding discrete time integer index
end
hold on;
ed = stem(Td(1:11)*t,yd(1:11), '*', 'DisplayName', 'Exact Discrete');
ed.Color = 'red';

yfd = [];
yfd(1) = 1;
Td = [];
Td(1) = 0;
% Forward Difference
for i = 1:length(Tc)
    yfd(i+1) = (1+(t*A))*yfd(i) +(B*t*U);
    Td(i+1) = i;
end
hold on;
fd = stem(Td(1:11)*t,yfd(1:11), 'filled','DisplayName', 'Forward
 Difference');
fd.Color = 'green';
ybd = zeros(1, length(Td));
ybd(1) = 1;
% Backward Difference
for k = 2:length(Td)
    z = 0;
    for m = 1:k
        z = z + 1/(1-A*t)^(k-m+1) * (B*t*U);
```

```
    end
    ybd(k) = ybd(1)*(1/((1-A*t)^k)) + z;
end
bd = stem(Td(1:11)*t,ybd(1:11), 'filled','DisplayName', 'Backward
 Difference');
bd.Color = 'blue';
legend('CT','CT(kT)','Fwd','Back')
title('Discrete methods for T = 0.1')
figure
initial(sys, IC, 0:0.01:2)
t = .3;
for i = 1:length(Tc)
    yd(i+1) = Ad*yd(i) +(Bd*U);
    Td(i+1) = i; % save corresponding discrete time integer index
end
hold on;
ed = stem(Td(1:11)*t,yd(1:11), '*', 'DisplayName', 'Exact Discrete');
ed.Color = 'red';

yfd = [];
yfd(1) = 1;
Td = [];
Td(1) = 0;
% Forward Difference
for i = 1:length(Tc)
    yfd(i+1) = (1+(t*A))*yfd(i) +(B*t*U);
    Td(i+1) = i;
end
hold on;
fd = stem(Td(1:11)*t,yfd(1:11), 'filled','DisplayName', 'Forward
 Difference');
fd.Color = 'green';
ybd = zeros(1, length(Td));
ybd(1) = 1;
% Backward Difference
for k = 2:length(Td)
    z = 0;
    for m = 1:k
        z = z + 1/(1-A*t)^(k-m+1) * (B*t*U);
    end
    ybd(k) = ybd(1)*(1/((1-A*t)^k)) + z;
end
bd = stem(Td(1:11)*t,ybd(1:11), 'filled','DisplayName', 'Backward
 Difference');
bd.Color = 'blue';
legend('CT','CT(kT)','Fwd','Back')
title('Discrete methods for T = 0.3')
figure
t = .05;
initial(sys, IC, 0:0.01:2)
for i = 1:length(Tc)
    yd(i+1) = Ad*yd(i) +(Bd*U);
    Td(i+1) = i; % save corresponding discrete time integer index
end
```

```
hold on;
ed = stem(Td(1:11)*t,yd(1:11), '*', 'DisplayName', 'Exact Discrete');
ed.Color = 'red';

yfd = [];
yfd(1) = 1;
Td = [];
Td(1) = 0;
% Forward Difference
for i = 1:length(Tc)
    yfd(i+1) = (1+(t*A))*yfd(i) +(B*t*U);
    Td(i+1) = i;
end
hold on;
fd = stem(Td(1:11)*t,yfd(1:11), 'filled','DisplayName', 'Forward
 Difference');
fd.Color = 'green';
ybd = zeros(1, length(Td));
ybd(1) = 1;
% Backward Difference
for k = 2:length(Td)
    z = 0;
    for m = 1:k
        z = z + 1/(1-A*t)^(k-m+1) * (B*t*U);
    end
    ybd(k) = ybd(1)*(1/((1-A*t)^k)) + z;
end
bd = stem(Td(1:11)*t,ybd(1:11), 'filled','DisplayName', 'Backward
 Difference');
bd.Color = 'blue';
legend('CT','CT(kT)','Fwd','Back')
title('Discrete methods for T = 0.05')
```


Discrete methods for T = 0.1
CT
CT(kT)
Fwd
Back
Amplitude
Time (seconds)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1.2
1.4
1.6
1.8
2

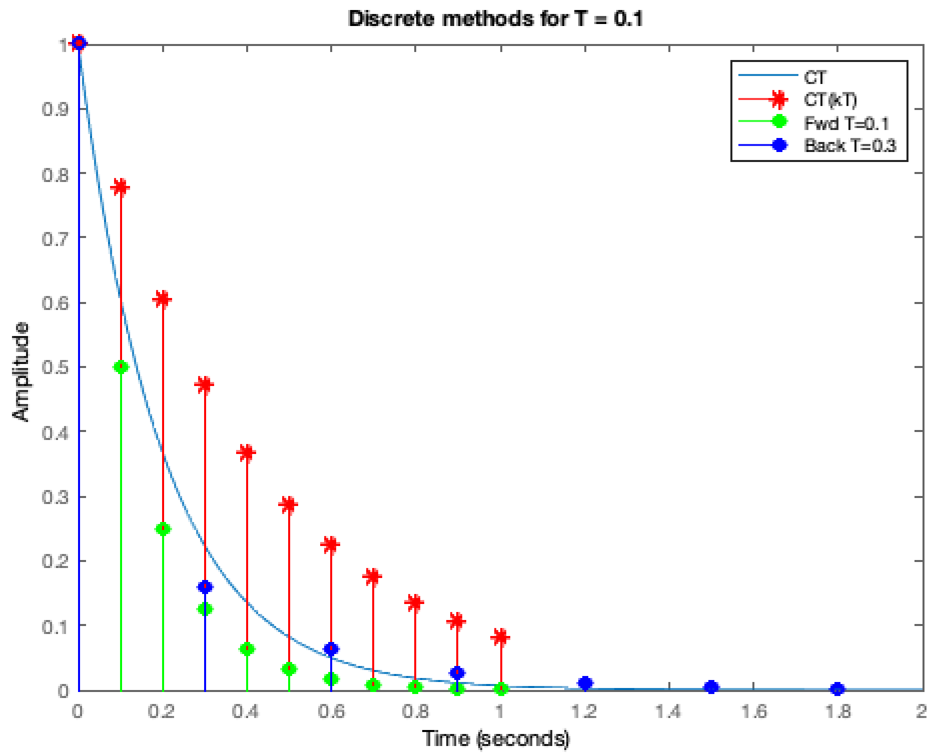

Discrete methods for T = 0.3
CT
CT(kT)
Fwd
Back
Amplitude
Time (seconds)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1.2
1.4
1.6
1.8
2



Part 4b2

```
close all
A = [-5.0];
B = [2.0];
C = [1.0];
D = [];
IC = 1;
U = 0;
figure
% Exact Difference
Ad = expm(-5*t);
Bd = ((-2/5)*expm(-5*t)) + (2/5);

sys = ss(A,B,C,D);
[Yc, Tc] = initial(sys, IC, 0:0.01:2);
hold on;
initial(sys, IC, 0:0.01:2)
ybd = [];
yd = [];
yd(1) = 1;
Td = [];
Td(1) = 0;
% Exact Discrete
t = .1;
```

```
for i = 1:length(Tc)
    yd(i+1) = Ad*yd(i) +(Bd*U);
    Td(i+1) = i; % save corresponding discrete time integer index
end
hold on;
ed = stem(Td(1:11)*t,yd(1:11), '*', 'DisplayName', 'Exact Discrete');
ed.Color = 'red';

yfd = [];
yfd(1) = 1;
Td = [];
Td(1) = 0;
% Forward Difference
for i = 1:length(Tc)
    yfd(i+1) = (1+(t*A))*yfd(i) +(B*t*U);
    Td(i+1) = i;
end
hold on;
fd = stem(Td(1:11)*t,yfd(1:11), 'filled','DisplayName', 'Forward
 Difference');
fd.Color = 'green';
ybd = zeros(1, length(Td));
ybd(1) = 1;
% Backward Difference
t = 0.3;
for k = 2:length(Td)
    z = 0;
    for m = 1:k
        z = z + 1/(1-A*t)^(k-m+1) * (B*t*U);
    end
    ybd(k) = ybd(1)*(1/((1-A*t)^k)) + z;
end
bd = stem(Td(1:11)*t,ybd(1:11), 'filled','DisplayName', 'Backward
 Difference');
bd.Color = 'blue';
legend('CT','CT(kT)','Fwd T=0.1','Back T=0.3')
title('Discrete methods for T = 0.1')
```



Part 4b3

```
% the results shows that backwards difference will work for all T >
 -1/3
% and forward difference will only work for T = 0<T<1/3
% inorder to have the same negative poles for the transfer function,
 the T
% for forward has to be less than 0.2 and the T for backward have to
 be
% greater than 0.2 so in the graph, T for forward was 0.1 and backward
 was
% 0.3 and this will have the same poles for the transfer function.

% also for forward difference the it grows very instable as T
 increases,
% and the threshhold is 0.3333 before it becomes unstable and this can
 be
% seen in the graph.
```

Published with MATLAB® R2018b