Given the following state space system

$$\frac{dx}{dt} = Ax + Bu \qquad y = Cx + Du$$

with

$$A = \begin{bmatrix} 0 & 0 & -1 \\ 2 & 1 & 2 \\ -2 & 0 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \qquad C = [\,0 \;\; 1 \;\; 0\,] \qquad D = 0$$

and $x_0 = [\,1 \;\; 2 \;\; 0\,]^T$

same state matrix from HW#2

Find the impulse response of the system using the following three methods. Show that the solutions are identical by comparing all 3 cases on a single plot and via a short summary table of numerical values over the range $0 \le t \le 2.5$ sec.

Case 1: Analytical solution using the matrix exponential approach to obtain an explicit expression for $y(t)$.

Case 2: Using matlab's built-in LTI solvers (impulse, step, lsim)

Case 3. Using numerical integration via Matlab's ode45 routine.

From prior HW, for $A = \begin{bmatrix} 0 & 0 & -1 \\ 2 & 1 & 2 \\ -2 & 0 & 1 \end{bmatrix}$

we had

$$A^{-1} = \begin{bmatrix} -\frac{1}{2} & 0 & -\frac{1}{2} \\ 3 & 1 & 1 \\ -1 & 0 & 0 \end{bmatrix}$$

$$\lambda_1 = -1 \qquad \lambda_2 = 1 \qquad \lambda_3 = 2$$

$$e^{At} = \frac{1}{3} \begin{bmatrix} 2e^{-t} + e^{2t} & 0 & e^{-t} - e^{2t} \\ -4e^{-t} + 6e^{t} - 2e^{2t} & 3e^{t} & -2e^{-t} + 2e^{2t} \\ 2e^{-t} - 2e^{2t} & 0 & e^{-t} + 2e^{2t} \end{bmatrix}$$

Case #1 Analytical Soln via Hand Calc

Analytical Soln for LTI syst

$$x(t) = e^{At}x_0 + \int_0^t e^{A(t-\tau)} B\,u(\tau)\,d\tau$$

But, for the impulse input

$$u(\tau) = \delta(\tau) \qquad \text{where } \delta(t-t_0) = \begin{cases} 0 & t \neq t_0 \\ \infty & t = t_0 \end{cases}$$

Thus, for our case, we have

$$\int_{-\infty}^{\infty} \delta(t-t_0)\,dt = 1$$

$$x(t) = e^{At}x_0 + \int_{0^-}^{t} e^{A(t-\tau)} B\,\delta(\tau)\,d\tau$$

$$\int_{-\infty}^{\infty} f(t)\,\delta(t-t_0)\,dt = f(t_0)$$

$$= e^{At}x_0 + e^{At}B$$

or $\boxed{x(t) = e^{At}(x_0 + B)}$ { an impulse input manifests itself as an initial condition on the system

$$\therefore \quad x(t) = \frac{1}{3}\begin{bmatrix} 2e^{-t}+e^{2t} & 0 & e^{-t}-e^{2t} \\ -4e^{-t}+6e^{t}-2e^{2t} & 3e^{t} & -2e^{-t}+2e^{2t} \\ 2e^{-t}-2e^{2t} & 0 & e^{-t}+2e^{2t} \end{bmatrix} \underbrace{\begin{bmatrix} 1 \\ 3 \\ 0 \end{bmatrix}}_{x_0 + B}$$

$$= \frac{1}{3}\begin{bmatrix} 2e^{-t}+e^{2t} \\ -4e^{-t}+15e^{t}-2e^{2t} \\ 2e^{-t}-2e^{2t} \end{bmatrix}$$

and

$$y(t) = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$\boxed{y(t) = -\frac{4}{3}e^{-t} + 5e^{t} - \frac{2}{3}e^{2t}}$$

Case # 2 Matlab soln

The only issue here is that the impulse command only allows zero initial conditions — so we cannot use this directly since $x_0 \neq 0$ here

Instead, we recognize from Case 1 that the impulse input manifests itself as an initial condition on the system. Thus, we can treat this by changing the given IC as follows:

$$x_{0_{new}} = x_0 + B$$

and solve the problem as an unforced system using lsim.
$u = 0$ for $t > 0$

Case # 3 Solution using ODE45

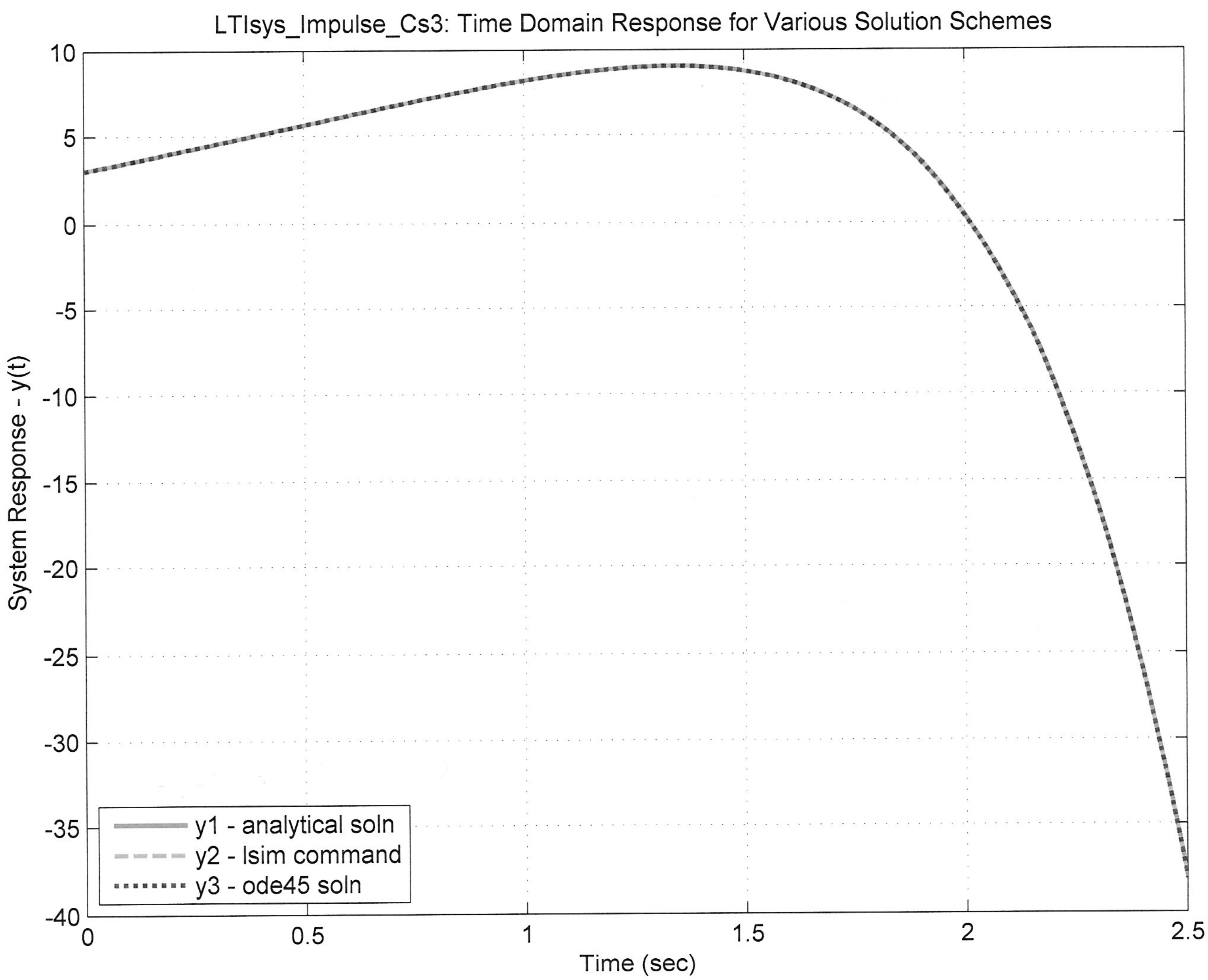
This problem is also straightforward if we actually model an unforced system, $u = 0$, with the ICs set as $x_{0_{new}} = x_0 + B$

Upon return from ODE45, we can form the desired output as $y = Cx$ or simply $y(t) = x_2(t)$ for this problem

→ OK, all three cases should work and give ~~essentially~~ the same result... — let's see

see LTIsys_impulse_cs3.m

got identical solutions OK


LTIsys_Impulse_Cs3: Time Domain Response for Various Solution Schemes
System Response - y(t)
Time (sec)
10
5
0
-5
-10
-15
-20
-25
-30
-35
-40
0
0.5
1
1.5
2
2.5
y1 - analytical soln
y2 - lsim command
y3 - ode45 soln

```
>> ltisys_impulse_cs3

      Comparison of Results for LTIsys_Impulse_Cs3 Cases

          t             y1            y2            y3
      0.00e+000    3.0000e+000   3.0000e+000   3.0000e+000
      1.00e-001    3.5051e+000   3.5051e+000   3.5051e+000
      2.00e-001    4.0208e+000   4.0208e+000   4.0208e+000
      3.00e-001    4.5468e+000   4.5468e+000   4.5468e+000
      4.00e-001    5.0817e+000   5.0817e+000   5.0817e+000
      5.00e-001    5.6227e+000   5.6227e+000   5.6227e+000
      6.00e-001    6.1654e+000   6.1654e+000   6.1654e+000
      7.00e-001    6.7032e+000   6.7032e+000   6.7032e+000
      8.00e-001    7.2266e+000   7.2266e+000   7.2266e+000
      9.00e-001    7.7228e+000   7.7228e+000   7.7228e+000
      1.00e+000    8.1749e+000   8.1749e+000   8.1749e+000
      1.10e+000    8.5603e+000   8.5603e+000   8.5603e+000
      1.20e+000    8.8502e+000   8.8502e+000   8.8502e+000
      1.30e+000    9.0073e+000   9.0073e+000   9.0073e+000
      1.40e+000    8.9841e+000   8.9841e+000   8.9841e+000
      1.50e+000    8.7206e+000   8.7206e+000   8.7206e+000
      1.60e+000    8.1409e+000   8.1409e+000   8.1409e+000
      1.70e+000    7.1501e+000   7.1501e+000   7.1501e+000
      1.80e+000    5.6290e+000   5.6290e+000   5.6290e+000
      1.90e+000    3.4293e+000   3.4293e+000   3.4292e+000
      2.00e+000    3.6607e-001   3.6607e-001   3.6605e-001
      2.10e+000   -3.7900e+000  -3.7900e+000  -3.7900e+000
      2.20e+000   -9.3232e+000  -9.3232e+000  -9.3233e+000
      2.30e+000   -1.6586e+001  -1.6586e+001  -1.6586e+001
      2.40e+000   -2.6012e+001  -2.6012e+001  -2.6012e+001
      2.50e+000   -3.8139e+001  -3.8139e+001  -3.8139e+001
>>
```

```
%
%  LTISYS_IMPULSE_CS3.M   Computes and Compares the Impulse Response
%                 for a particular LTI system using three methods  (Case 3 Data)
%
%  This file simulates a 3rd order LTI system using various solution schemes:
%     Case 1:  Analytical solution using the matrix exponential (done by hand)
%     Case 2:  Use Matlab's built-in impulse, step, or lsim commands (as needed)
%     Case 3:  Use Matlab's built-in ode solver -- ode45 (for example)
%
%  The goal here is to know how to work with all three approaches -- especially when
%  the input function is the unit impulse (a very important case indeed).  See the
%  hand calculations for the development of the equations/approaches used here...
%
%  File prepared by J. R. White, UMass-Lowell  (Feb. 2014)
%

   clear all, close all,  nfig = 0;
%
%  define system of interest
   A = [0 0 -1; 2 1 2; -2 0 1];   B = [0 1 0]';   C = [0 1 0];   D = 0;
   xo = [1 2 0]';
%
%  Case 1 -- analytical solution (see details in notes)
   t1 = 0;   t2 = 2.5;   Nt = 251;   t = linspace(t1,t2,Nt)';
   y1 = (-4/3)*exp(-t) + 5*exp(t) - (2/3)*exp(2*t);
%
%  Case 2 -- solve unforced system using lsim command with new initial conditions
   sys2 = ss(A,B,C,D);   xon2 = xo + B;   u2 = zeros(Nt,1);
   [y2,ttb] = lsim(sys2,u2,t,xon2);
%
%  Case 3  --  solve as an unforced system in ode45 with new initial conditions
   xon3 = xo + B;   options = odeset('RelTol',1.0e-6);
   ftx = @(t,x) A*x;
   [t3,x3] = ode45(ftx,t,xon3,options);   y3 = x3(:,2);
%
%  plot desired response, y(t), for all three cases
   nfig = nfig+1;   figure(nfig)
   plot(t,y1,'r-',t,y2,'g--',t,y3,'b:','LineWidth',2),grid
   title('LTIsys\_Impulse\_Cs3: Time Domain Response for Various Solution Schemes')
   xlabel('Time (sec)'),ylabel('System Response - y(t)')
   legend('y1 - analytical soln','y2 - lsim command','y3 - ode45 soln', ...
         'Location','SouthWest')
%
%  print brief table of results to screen
   fprintf('\n')
   fprintf('    Comparison of Results for LTIsys_Impulse_Cs3 Cases \n\n')
   fprintf('        t             y1             y2             y3 \n')
   for i = 1:10:Nt
     fprintf('   %10.2e  %12.4e  %12.4e  %12.4e \n',t(i), y1(i),y2(i),y3(i));
   end
%
%  end of program
```

Given $\frac{dx}{dt} = Ax + Bu \qquad y = Cx$

with $A = \begin{bmatrix} 0 & 0 & -1 \\ 2 & 1 & 2 \\ -2 & 0 & 1 \end{bmatrix} \quad x_0 = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad C = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}^T$

with $u(t) = \begin{cases} 0 & t < 0 \\ 1 & 0 < t < 1 \\ 0 & t > 1 \end{cases}$

Determine the analytic solution. In particular, create a table for $y(t)$ versus time at 0.5 second intervals out to 2.5 sec. Also use Matlab's lsim and ODE45 routines and compare...

The general solution for an LTI system can be written as

$$x(t) = e^{A(t-t_0)} x(t_0) + \int_{t_0}^{t} e^{A(t-\tau)} B\, u(\tau)\, d\tau$$

but since $u(\tau)$ is discontinuous at $\tau = 1$ sec, we have two regions

done in Notes

for $0 < t < 1$ sec $\qquad x(t) = e^{At} x_0 + A^{-1}(e^{At} - I)B \qquad$ since $u = 1$

for $t > 1$ sec $\qquad x(t) = e^{A(t-1)} x(1) \qquad$ since $u = 0$

To get numerical values, we need expressions for A^{-1} and e^{At}. However, these were already computed as part of HW#2 and they are simply reproduced here:

$$A^{-1} = \begin{bmatrix} -1/2 & 0 & -\frac{1}{2} \\ 3 & 1 & 1 \\ -1 & 0 & 0 \end{bmatrix} \qquad \text{from HW\#2}$$

$$e^{At} = \frac{1}{3}\begin{bmatrix} 2e^{-t} + e^{2t} & 0 & e^{-t} - e^{2t} \\ -4e^{-t} + 6e^{t} - 2e^{2t} & 3e^{t} & -2e^{-t} + 2e^{2t} \\ 2e^{-t} - 2e^{2t} & 0 & e^{-t} + 2e^{2t} \end{bmatrix} \qquad \text{from HW\#2}$$

for $t < 1$ sec, we need $e^{At}x_0$, $A^{-1}B$, and $A^{-1}e^{At}B$

$$e^{At}x_0 = \frac{1}{3}\begin{bmatrix} 2e^{-t} + e^{2t} \\ -4e^{-t} + 12e^{t} - 2e^{2t} \\ 2e^{-t} - 2e^{2t} \end{bmatrix}$$

$$A^{-1}B = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$A^{-1}e^{At}B = \begin{bmatrix} -\frac{1}{2} & 0 & -\frac{1}{2} \\ 3 & 1 & 1 \\ -1 & 0 & 0 \end{bmatrix}\begin{bmatrix} 0 \\ e^{t} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ e^{t} \\ 0 \end{bmatrix}$$

$\therefore$ for $t < 1$ sec

$$x(t) = \frac{1}{3}\begin{bmatrix} 2e^{-t} + e^{2t} \\ -4e^{-t} + 12e^{t} - 2e^{2t} \\ 2e^{-t} - 2e^{2t} \end{bmatrix} + \begin{bmatrix} 0 \\ e^{t} \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{3}(2e^{-t} + e^{2t}) \\ -\frac{4}{3}e^{-t} + 5e^{t} - \frac{2}{3}e^{2t} - 1 \\ \frac{1}{3}(2e^{-t} - 2e^{2t}) \end{bmatrix}$$

← This is $y(t)$ for $t < 1$

also evaluate this vector in Matlab at $t = 1$

$\therefore$

t	$x_1(t)$	$y(t) = x_2(t)$	$x_3(t)$
0	1	2	0
0.5	1.3104	4.6227	-1.4078
1.0	2.7083	7.1749	-4.6808

also since $y(t) = [0\ 1\ 0]\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = x_2(t)$

for $t > 1$ sec

$$X(t) = e^{A(t-1)} X(1)$$

$$= \frac{1}{3}\begin{bmatrix} 2e^{-(t-1)} + e^{2(t-1)} & 0 & e^{-(t-1)} - e^{2(t-1)} \\ -4e^{-(t-1)} + 6e^{(t-1)} - 2e^{2(t-1)} & 3e^{(t-1)} & -2e^{-(t-1)} + 2e^{2(t-1)} \\ 2e^{-(t-1)} - 2e^{2(t-1)} & 0 & e^{-(t-1)} + 2e^{2(t-1)} \end{bmatrix} \begin{bmatrix} 2.7083 \\ 7.1749 \\ -4.6808 \end{bmatrix}$$

note
$y(t) = [2^{nd} \text{ row}] \left[\underline{X}\right]_{t=1}$
for $t > 1$

for example, let's evaluate this at $t = 2$ sec

$$X(2) = \frac{1}{3}\begin{bmatrix} 2e^{-1} + e^{2} & 0 & e^{-1} - e^{2} \\ -4e^{-1} + 6e - 2e^{2} & 3e & -2e^{-1} + 2e^{2} \\ 2e^{-1} - 2e^{2} & 0 & e^{-1} + 2e^{2} \end{bmatrix} \begin{bmatrix} 2.7083 \\ 7.1749 \\ -4.6808 \end{bmatrix}$$

$$= \frac{1}{3}\begin{bmatrix} 8.1248 & 0 & -7.0212 \\ 0.0601 & 8.1548 & 14.0424 \\ -14.0424 & 0 & 15.1460 \end{bmatrix} \begin{bmatrix} 2.7083 \\ 7.1749 \\ -4.6808 \end{bmatrix}$$

$$= \frac{1}{3}\begin{bmatrix} 54.869 \\ -7.057 \\ -108.93 \end{bmatrix} = \begin{bmatrix} 18.29 \\ -2.352 \\ -36.31 \end{bmatrix}$$

OK agrees with matlab solution

→ see the Matlab m file and plots for the Matlab solution to this problem use lsim and ODE45

Note that lsim and ODE45 must be restarted at $t = 1$ sec because of the discontinuity at $t = 1$ sec.

Got identical solutions ...

LTISYS _ PULSE _ CS3

extra

Note:

Although the expression for $t > 1$ is relatively obvious

$$x(t) = e^{A(t-1)} x(1) \qquad \text{for } t > 1$$

This can be formally derived by integrating the analytical soln from 0 to τ, where $t > 1$

$$x(t) = e^{A(t-t_0)} x(t_0) + \int_{t_0}^{t} e^{A(t-\gamma)} B u(\gamma)\, d\gamma$$

or

$$x(t) = e^{At} x_0 + \underbrace{e^{At} \int_0^1 e^{-A\gamma} B u\, d\gamma}_{u = 1 \text{ here}} + \underbrace{e^{At} \int_1^{\tau} e^{-A\gamma} B u\, d\gamma}_{u = 0 \text{ here} \ \therefore \text{ this vanishes}}$$

$$= e^{At} x_0 + e^{At} \left[-A^{-1} e^{-A\gamma} B \right] \Big|_0^1$$

$$= e^{At} x_0 + e^{At} \left[-A^{-1} \left(e^{-A} - I \right) B \right]$$

$$= e^{At} x_0 + e^{At} \left[A^{-1} \left(I - e^{-A} \right) B \right]$$

Now, insert $I = e^{A} e^{-A} = e^{-A} e^{A}$, as appropriate

$$x(t) = e^{At} e^{-A} e^{A} x_0 + e^{At} e^{-A} e^{A} \left[A^{-1} \left(I - e^{-A} \right) B \right]$$

$$= e^{A(t-1)} \Big\{ \underbrace{e^{A} x_0 + A^{-1} \left(e^{A} - I \right) B}_{} \Big\}$$

however, this is just $x(1)$ for $t \le 1$

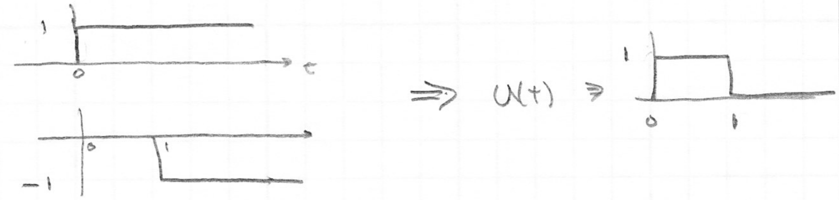
$$\therefore \boxed{x(t) = e^{A(t-1)} x(1) \qquad \text{for } t > 1}$$

ok, this is what was used in the calculations for this problem...

extra
extra

Note: One student tried to solve this problem using two unit step functions — one at $t_0 = 0$ and a second one at $t_0 = 1$ with a strength of -1

That is $u_1(t) \Rightarrow$ $u_2(t) \Rightarrow$ $\Rightarrow u(t) \Rightarrow$



Let's show that this should work just fine ...

general soln

$$x(t) = e^{A(t-t_0)} x(t_0) + e^{At}\int_{t_0}^{t} e^{-A\tau} B\, u(\tau)\, d\tau$$

for u = constant, this becomes

$$x(t) = e^{A(t-t_0)} x(t_0) + A^{-1}\left(e^{A(t-t_0)} - I\right) B u$$

Now, for input #1, $u = 1$, $t_0 = 0$, and $x(t_0) = x_0$

$$\therefore\ x(t) = e^{At} x_0 + A^{-1}\left(e^{At} - I\right) B$$

And, for input #2, $u = -1$ $t_0 = 1$, and $x(t_0) = 0$
(note that this input is totally independent of the first case — and, from its perspective, the system is at rest when this input is applied)

$$\therefore\ x(t) = 0 + A^{-1}\left(e^{A(t-1)} - I\right) B(-1)$$

Now, the desired output is the sum of the responses from the two linearly independent inputs, or

$$x(t) = e^{At} x_0 + A^{-1}\left(e^{At} - I\right) B - A^{-1}\left(e^{A(t-1)} - I\right) B$$

$$= e^{At} x_0 + A^{-1} e^{At} B - \cancel{A^{-1} B} - A^{-1} e^{A(t-1)} B + \cancel{A^{-1} B}$$

$$= e^{At} x_0 + A^{-1}\left[e^{At} - e^{A(t-1)}\right] B$$

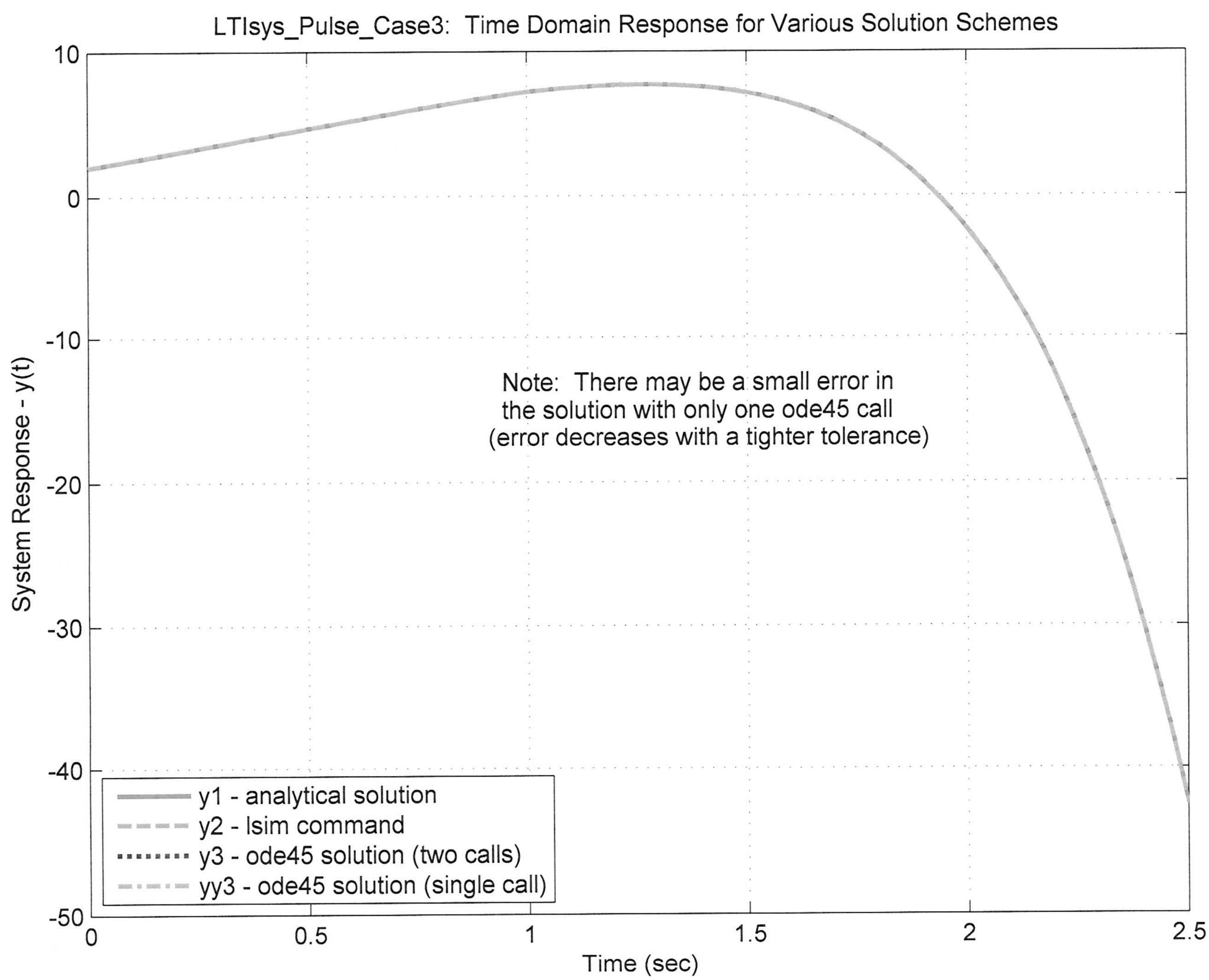
$$= e^{At} \underbrace{e^{-A} e^{A}}_{I} x_0 + A^{-1}\left[e^{At} \underbrace{e^{-A} e^{A}}_{I} - e^{A(t-1)}\right] B$$

$$= e^{A(t-1)} \Big[\underbrace{e^{A} x_0 + A^{-1}\left(e^{A} - I\right) B}_{} \Big]$$

but we know this is $x(1)$ from the first part of the problem

$$\therefore \text{ for } t \ge 1 \quad x(t) = e^{A(t-1)} x(1)$$

OK, this is the same soln as before


LTIsys_Pulse_Case3: Time Domain Response for Various Solution Schemes
10
0
-10
-20
-30
-40
-50
0
0.5
1
1.5
2
2.5
Time (sec)
System Response - y(t)
Note: There may be a small error in
the solution with only one ode45 call
(error decreases with a tighter tolerance)
y1 - analytical solution
y2 - lsim command
y3 - ode45 solution (two calls)
yy3 - ode45 solution (single call)

```
>> ltisys_pulse_cs3

    Comparison of Results for LTIsys_Pulse_Cs3 Cases (RelTol = 1.0e-005)

```

t	y1	y2	y3	yy3
0.000e+000	2.0000e+000	2.0000e+000	2.0000e+000	2.0000e+000
1.000e-001	2.5051e+000	2.5051e+000	2.5051e+000	2.5051e+000
2.000e-001	3.0208e+000	3.0208e+000	3.0208e+000	3.0208e+000
3.000e-001	3.5468e+000	3.5468e+000	3.5468e+000	3.5468e+000
4.000e-001	4.0817e+000	4.0817e+000	4.0817e+000	4.0817e+000
5.000e-001	4.6227e+000	4.6227e+000	4.6227e+000	4.6227e+000
6.000e-001	5.1654e+000	5.1654e+000	5.1654e+000	5.1654e+000
7.000e-001	5.7032e+000	5.7032e+000	5.7032e+000	5.7032e+000
8.000e-001	6.2266e+000	6.2266e+000	6.2266e+000	6.2266e+000
9.000e-001	6.7228e+000	6.7228e+000	6.7228e+000	6.7228e+000
1.000e+000	7.1749e+000	7.1749e+000	7.1749e+000	7.1719e+000
1.000e+000	7.1749e+000	7.1749e+000	7.1749e+000	7.1719e+000
1.100e+000	7.4552e+000	7.4552e+000	7.4552e+000	7.4514e+000
1.200e+000	7.6288e+000	7.6288e+000	7.6288e+000	7.6247e+000
1.300e+000	7.6574e+000	7.6574e+000	7.6574e+000	7.6529e+000
1.400e+000	7.4923e+000	7.4923e+000	7.4923e+000	7.4873e+000
1.500e+000	7.0719e+000	7.0719e+000	7.0718e+000	7.0663e+000
1.600e+000	6.3188e+000	6.3188e+000	6.3188e+000	6.3127e+000
1.700e+000	5.1363e+000	5.1363e+000	5.1363e+000	5.1295e+000
1.800e+000	3.4035e+000	3.4035e+000	3.4034e+000	3.3959e+000
1.900e+000	9.6965e-001	9.6965e-001	9.6963e-001	9.6134e-001
2.000e+000	-2.3522e+000	-2.3522e+000	-2.3522e+000	-2.3614e+000
2.100e+000	-6.7941e+000	-6.7941e+000	-6.7942e+000	-6.8043e+000
2.200e+000	-1.2643e+001	-1.2643e+001	-1.2643e+001	-1.2655e+001
2.300e+000	-2.0255e+001	-2.0255e+001	-2.0255e+001	-2.0267e+001
2.400e+000	-3.0067e+001	-3.0067e+001	-3.0067e+001	-3.0081e+001
2.500e+000	-4.2621e+001	-4.2621e+001	-4.2621e+001	-4.2636e+001

```
>>
```

```
%
%  LTISYS_PULSE_CS3.M   Computes and Compares the Response to an Input Pulse
%                  for a particular LTI system using three methods  (Case 3 Data)
%
%  This file simulates a 3rd order LTI system using various solution schemes:
%     Case 1:  Analytical solution using the matrix exponential (done by hand)
%     Case 2:  Use Matlab's built-in impulse, step, or lsim commands (as needed)
%     Case 3:  Use Matlab's built-in ode solver -- ode45 (for example)
%
%  The goal here is to know how to work with all three approaches -- especially when
%  the input function is a pulse input.  In particular, for this type of input,
%  the LTI solution techniques within Matlab are exact.  Thus, this example will
%  allow us to validate our understanding of the various solution techniques.  See
%  the hand calculations for the development of the equations/approaches used here.
%
%  File prepared by J. R. White, UMass-Lowell  (Feb. 2014)
%

   clear all, close all,  nfig = 0;
%
%  define system of interest
   A = [0 0 -1; 2 1 2; -2 0 1];   B = [0 1 0]';   C = [0 1 0];   D = 0;
   xo = [1 2 0]';
%
%  define discrete time vector -- here we will use two intervals because of the
%  discontinuous u(t) at t = 1 sec
   t1 = 0;   t2 = 1;   t3 = 2.5;
   Nta = 51;  ta = linspace(t1,t2,Nta)';
   Ntb = 76;  tb = linspace(t2,t3,Ntb)';
   t = [ta; tb];
%
%  Case 1 -- analytical solution (see details in notes)
   y1a = (-4/3)*exp(-ta) + 5*exp(ta) - (2/3)*exp(2*ta) - 1;
   xat1 = [(2/3)*exp(-t2) + (1/3)*exp(2*t2);
           (-4/3)*exp(-t2) + 5*exp(t2) - (2/3)*exp(2*t2) - 1;
           (2/3)*exp(-t2) - (2/3)*exp(2*t2)];
   dt = tb-t2;
   y1b = ((-4/3)*exp(-dt) + 2*exp(dt) - (2/3)*exp(2*dt))*xat1(1) + ...
         exp(dt)*xat1(2) + ((-2/3)*exp(-dt) + (2/3)*exp(2*dt))*xat1(3);
   y1 = [y1a; y1b];
%
%  Case 2 -- solve with lsim with two separate calls (start at local t=0 each time)
   sys = ss(A,B,C,D);   xo2a = xo;  u2a = ones(Nta,1);
   [y2a,t2a,x2a] = lsim(sys,u2a,ta,xo2a);
   xo2b = x2a(end,:)'; u2b = zeros(Ntb,1);
   [y2b,t2b,x2b] = lsim(sys,u2b,dt,xo2b);
   y2 = [y2a; y2b];
%
%  Case 3 -- solve in ode45 with two separate calls
   tol = 1.0e-5;  options = odeset('RelTol',tol);
   xo3a = xo;   uopt = 1;
   ftx1 = @(t,x) ltisys_odeeqn(t,x,A,B,uopt);
```

```
      [t3a,x3a] = ode45(ftx1,ta,xo3a,options);   y3a = x3a(:,2);
      xo3b = x3a(end,:);
      ftx2 = @(t,x) A*x;
      [t3b,x3b] = ode45(ftx2,tb,xo3b,options);   y3b = x3b(:,2);
      y3 = [y3a; y3b];
%
%   Case 3  --  solve in ode45 with single call with u(t) defined in odefile
%   Note: This case may have a small amount of error -- see plot for t > 1 sec.
%         However, the error should decrease with a tighter tolerance.
%         The moral here is that one should be careful with discontinuous inputs.
%         The best way to treat these cases is to have multiple calls to the
%         solution algorithn, with the end condition of one interval as the
%         initial condition for the next interval...
      xo3 = xo;   uopt = 2;  % uopt = 2 has a discontinuous input at t = 1 sec
      tt = t;  tt(Nta+1) = tt(Nta+1)+100*eps;  % time vector must be monotonic
      ftx1 = @(t,x) ltisys_odeeqn(t,x,A,B,uopt);
      [tt3,x3] = ode45(ftx1,tt,xo3,options);   yy3 = x3(:,2);
%
%   plot desired response, y(t), for all three (really four) cases
      nfig = nfig+1;   figure(nfig)
      plot(t,y1,'r-',t,y2,'g--',t,y3,'b:',tt3,yy3,'c-.','LineWidth',2),grid
      title('LTIsys\_Pulse\_Case3:  Time Domain Response for Various Solution Schemes')
      xlabel('Time (sec)'),ylabel('System Response - y(t)')
      legend('y1 - analytical solution','y2 - lsim command', ...
             'y3 - ode45 solution (two calls)', ...
             'yy3 - ode45 solution (single call)','Location','SouthWest')
      txt = {['  Note:  There may be a small error in    ';
              '  the solution with only one ode45 call   ';
              '(error decreases with a tighter tolerance)']};
      text(0.91,-15,txt)
%
%   print brief table of results to screen
      fprintf('\n')
      fprintf('    Comparison of Results for LTIsys_Pulse_Cs3 Cases (RelTol = %8.1e)↙
\n\n', tol)
      fprintf('        t              y1             y2             y3             yy3\n')
      for i = 1:5:Nta
        fprintf('    %10.3e  %12.4e  %12.4e  %12.4e   %12.4e \n', ...
                ta(i), y1a(i),y2a(i),y3a(i),yy3(i));
      end
      for i = 1:5:Ntb
        fprintf('    %10.3e  %12.4e  %12.4e  %12.4e   %12.4e \n', ...
                tb(i), y1b(i),y2b(i),y3b(i),yy3(Nta+i));
      end
%
%   end of program
```

```
%
%   LTISYS_ODEEQN.M  LTI State Space System with Specified Forcing Function
%
%   Used to evaluate derivative of state at some time point (called by ODE45 or ODE23)
%   The forcing function u(t) must be defined for each case of interest via the use
%   of the uopt switch.
%
      function xp = ltisys_odeeqn(t,x,A,B,uopt)
%
%   Case 1: scalar input with u(t) = 1
      if uopt == 1, u = 1; end
%   Case 2: scalar input with u(t) = 1 for t <= 1, and u(t) = 0 for t > 1
      if uopt == 2,
        u = 1;  if t > 1, u = 0;  end
      end
%
      xp = A*x + B*u;
%
%  end of routine
```