1. The differential equation is

\[ \frac{d^2 y(t)}{dt^2} + 5 \frac{dy(t)}{dt} + 6y(t) = u(t) = \delta(t) \]

Find the homogeneous and particular solutions:

**Homogeneous:**
Assume \( y(t) = Ye^{st} \). Then

\[ 5^2 Y + 5s Y + 6Y = 0 \]

\[ \Rightarrow 5^2 + 5s + 6 = 0 \]

\[ \Rightarrow (5 + 2)(5 + 3) = 0 \]

\[ \Rightarrow s_1 = -2, \ s_2 = -3 \]

The homogeneous solution is therefore

\[ y_h(t) = a e^{-2t} + b e^{-3t} \]

**Particular:**
Since \( u(t) = \delta(t) \), \( u(t) \equiv 1 \equiv \text{constant for } t > 0 \). Therefore, assume

\[ y_p(t) = c = \text{constant} \]

Plugging into the differential equation,

\[ 6c = 1 \Rightarrow c = \frac{1}{6} \]

**Total solution:**

The total solution is

\[ y(t) = y_p(t) + y_h(t) = \frac{1}{6} + a e^{-2t} + b e^{-3t} \]
The ICs are \( y(0) = 0 \), \( y'(0) = 0 \). Therefore,

\[
a + b = -\frac{1}{6} \\
-2a - 3b = 0
\]

Solving,

\[
a = -\frac{1}{2} \\
b = \frac{1}{3}
\]

Therefore,

\[
s(t) = \frac{1}{6} - \frac{1}{2} e^{-2t} + \frac{1}{3} e^{-3t}, \quad t \geq 0
\]

\[
s(t) = 0, \quad t < 0
\]

2. \( g(t) = \frac{d}{dt} s(t) \)

\[
= e^{-2t} - e^{-3t}, \quad t > 0
\]

\[
= 0, \quad t < 0
\]

\[
= 0, \quad t = 0
\]

The last part is because \( s(t) \) has no discontinuity at \( t = 0 \). Therefore,

\[
g(t) = \sigma(t) \left[ e^{-2t} - e^{-3t} \right]
\]

3. Since the input is an exponential, it makes sense to guess

\[
y(t) = ce^{-2t}
\]
If we plug this into the D.E, we obtain
\[ 4 ce^{-2t} - 10 ce^{-2t} + 6 ce^{-2t} = e^{-2t} \]
\[ \Rightarrow 0 = e^{-2t} \]

But this is not possible. So our guess doesn't work.

As we'll see below, a better guess is

\[ y_p(t) = c + e^{-2t} \]

4. \[ y(t) = \int_0^t g(t-\tau) u(\tau) d\tau \]
\[ = \int_0^t [e^{-2(t-\tau)} - e^{-3(t-\tau)}] e^{-2\tau} d\tau \]
\[ = \int_0^t [e^{-2t} - 3 e^{-3t+\tau}] d\tau \]
\[ = e^{-2t} \int_0^t dt - 3 e^{-3t} \int_0^t e^{\tau} d\tau \]
\[ = e^{-2t} \cdot t - 3 e^{-3t} \cdot (e^t - 1) \]

Therefore,

\[ y(t) = \left[ 3 e^{-2t} - 3 e^{-3t} + t e^{-2t} \right] \delta(t) \]

In particular homogeneous

So the response to an exponential is not always an exponential—sometimes it includes a secular term (one with a factor of \( t \))
(a) \[ x(t) = e^{-\alpha t} \frac{\tau}{t} \]
\[ h(t) = e^{-\beta t} \frac{\tau}{t} \]

\[ y(t) = h(t) * x(t) \]
\[ = \int_{-\infty}^{\infty} h(t-\tau) x(\tau) \, d\tau \]
\[ = 0, \quad t < 0 \]
\[ = \int_{0}^{t} e^{-\beta (t-\tau)} e^{-\alpha \tau} \, d\tau, \quad t > 0 \]
\[ = \int_{0}^{t} e^{-\beta t} e^{(\beta-\alpha)\tau} \, d\tau \]

If \( \beta \neq \alpha \), then
\[ y(t) = \int_{0}^{t} e^{-\beta t} e^{(\beta-\alpha)\tau} \, d\tau \]
\[ = e^{-\beta t} \left. \frac{1}{\beta-\alpha} e^{(\beta-\alpha)\tau} \right|_{\tau=0}^{t} \]
\[ = \frac{1}{\beta-\alpha} e^{-\beta t} \left( e^{(\beta-\alpha)t} - 1 \right) \]
\[ = \frac{1}{\beta-\alpha} e^{-\alpha t} - \frac{1}{\beta-\alpha} e^{-\beta t}, \quad t > 0 \]

If \( \beta = \alpha \),
\[ y(t) = \int_{0}^{t} e^{\alpha t} e^{\alpha \tau} \, d\tau \]
\[ = t e^{\alpha t}, \quad t > 0 \]
In either case, the result will look generally like

\[ y(t) \]

\[ \chi(t) = \delta(t) - 2 \delta(t-2) + \delta(t-5) \]

\[ h(t) = e^{2t} \delta(1-t) \]

\[ y(t) = h(t) \ast \chi(t) \]

Use flip & slide to get feel for answer:
Depending on the value of $t$, there are 4 cases:

$t > 6$: For this case, there is no overlap, so

$$y(t) = 0, \quad t > 6$$

$3 < t < 6$: For this case,

$$y(t) = \int_{t-1}^{5} e^{2(t-\tau)} (-1) \, d\tau$$

$$= e^{2t} \int_{t-1}^{5} (-1) e^{-2\tau} \, d\tau$$

$$= e^{2t} \left[ \frac{1}{2} e^{-2\tau} \right]_{\tau=t-1}^{\tau=5}$$

$$= e^{2t} \left[ \frac{1}{2} \left( e^{-10} - e^{-2(t-1)} \right) \right]$$

$1 < t < 3$: For this case,

$$y(t) = \int_{t-1}^{2} e^{2(t-\tau)} (1) \, d\tau + \int_{2}^{5} e^{2(t-\tau)} (-1) \, d\tau$$

$$= -\frac{1}{2} e^{2t} e^{-2\tau} \bigg|_{\tau=t-1}^{\tau=2} + \frac{1}{2} e^{2t} e^{-2\tau} \bigg|_{\tau=2}^{\tau=5}$$

$$= \frac{1}{2} e^{2t} \left( e^{-2(t-1)} - e^{-4} \right) + \frac{1}{2} e^{2t} \left( e^{-10} - e^{-4} \right)$$

$t < 1$: For this case,

$$y(t) = \int_{0}^{2} e^{2(t-\tau)} (1) \, d\tau + \int_{2}^{5} e^{2(t-\tau)} (-1) \, d\tau$$
\[ y(t) = \frac{1}{2} e^{2t} \left( e^0 - e^{-4} \right) + \frac{1}{2} e^{2t} \left( e^{-10} - e^{-4} \right) \]

Simplifying,

\[
y(t) = \begin{cases} 
\frac{1}{2} e^{2t} \left( 1 - 2e^{-4} + e^{-10} \right), & t < 1 \\
\frac{1}{2} e^{2} + \frac{1}{2} e^{2t} \left( e^{-10} - 2e^{-4} \right), & 1 < t < 3 \\
\frac{1}{2} e^{2t-10} - \frac{1}{2} e^{2}, & 3 < t < 6 \\
0, & t > 6
\end{cases}
\]

Sketch of \( y(t) \):

\[ h(t) \]

(c) \[ \chi(t) \]

Flip \( h(t) \): \[ h(t-\tau) \]
There are 4 cases:

1. \( t < 1 \): In this case, there is no overlap, so
   \[ y(t) = 0 \]

2. \( 1 < t < 3 \): \[ y(t) = \int_0^{t-1} z \cdot \sin \pi z \, dz \]
   \[ = -\frac{2}{\pi} \cos \pi z \Bigg|_0^{t-1} \]
   \[ = -\frac{2}{\pi} \left[ \cos \pi(t-1) - 1 \right] \]
   \[ = \frac{2}{\pi} \left[ 1 + \cos \pi t \right] \]

3. \( 3 < t < 5 \): \[ y(t) = \int_{t-3}^2 z \cdot \sin \pi z \, dz \]
   \[ = -\frac{2}{\pi} \cos \pi z \Bigg|_{t-3}^2 \]
   \[ = -\frac{2}{\pi} \left[ \cos \pi z - \cos \pi(t-3) \right] \]
   \[ = -\frac{2}{\pi} \left[ 1 + \cos \pi t \right] \]

4. \( t > 5 \): There is no overlap, so \( y(t) = 0 \).

Therefore,

\[ y(t) = \begin{cases} 
\frac{2}{\pi} \left( 1 + \cos \pi t \right) & 1 < t < 3 \\
-\frac{2}{\pi} \left( 1 - \cos \pi t \right) & 3 < t < 5 \\
0 & \text{else}
\end{cases} \]
(d) \( y(t) = h(t) \ast x(t) \)

\( x(t) = a + b t \)

\( h(t) = \frac{1}{\pi} \left[ \delta(t) - \delta(t - 1) \right] - \frac{1}{3} \delta(t - 2) \)

\[
y(t) = \int_{0}^{1} \frac{1}{\pi} \left[ a + b(t - \tau) \right] d\tau + \int -\frac{1}{3} \delta(t - 2) \left[ a + b(t - \tau) \right] d\tau
\]

\[
= \frac{1}{\pi} \left[ a + b t \right]_{\tau=0}^{1} - \frac{2}{3} b \tau^2 \bigg|_{\tau=0}^{1}
\]

\[
- \frac{1}{3} \left[ a + b(t - 2) \right]
\]

\[
= \frac{4}{3} \left[ a + b t \right] - \frac{2}{3} b - \frac{1}{3} a + \frac{2}{3} b - b t
\]

\[
= a + b t
\]

So \( y(t) = x(t) \) ! ! !

(e) Flip & Slide:
when \( h(t) \) overlaps a positive pulse,

\[
y(t) = \int 1 \cdot h(t-t') \, dt' = 1/3
\]

when \( h(t) \) overlaps a negative pulse

\[
y(t) = -1/3
\]

If \( h(t) \) is convolved with a step, \( r(t) \), the result is

\[
h(t) \ast \sigma(t) = \frac{1}{3}
\]

Therefore, \( h(t) \ast x(t) \) should look like

![Graphical representation of convolution result](image-url)
1. Because \( g(t) \) and \( u(t) \) are piecewise constant, \( y(t) \) will be continuous and piecewise linear. We can find \( y(t) \) at the "corners" by evaluating at \( t = \text{integer} \), since the corners of \( g(t) \) and \( u(t) \) occur at the integers. So do flip & slide:

\[
y(t) = g(t-x)u(x)
\]

There is no overlap for \( t < -5 \) or \( t > 5 \). So do \( t = -4, -3, -2, \ldots, 4 \):

\( t = -4 \):

\[
y(-4) = \int g(t-x)u(x) \, dx = 1
\]

\( t = -3 \):

\[
y(-3) = \int g(t-x)u(x) \, dx
\]
So \( y(-3) = \int g(t-2)u(t) \, dt = 0 \)

Continuing in this fashion, we have

<table>
<thead>
<tr>
<th>( t )</th>
<th>( y(t) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>-4</td>
<td>1</td>
</tr>
<tr>
<td>-3</td>
<td>0</td>
</tr>
<tr>
<td>-2</td>
<td>1</td>
</tr>
<tr>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
</tr>
</tbody>
</table>

So \( y(t) \) is:

2. By linearity and time invariance, delaying \( u(t) \) by \( T \) will delay \( y(t) \) by \( T \), so the convolution \( g(t) \times u(t-T) \) is as above, shifted right by \( T \).
3. \( T \) is easily identified as the time at which the max of \( y(t) \) occurs.
1. \( G(s) = \int_0^\infty t e^{-at} e^{-st} \, dt \)
   \[ = \int_0^\infty t e^{-(s+a)t} \, dt \]

Integrate by parts:

\( u = t \quad \Rightarrow \quad du = dt \)

\( dv = e^{-(s+a)t} \, dt \quad \Rightarrow \quad v = -\frac{1}{s+a} e^{-(s+a)t} \)

Therefore,

\[ G(s) = uv - \int v \, du \]

\[ = -\frac{t}{s+a} e^{-(s+a)t} \bigg|_0^\infty + \frac{1}{s+a} \int_0^\infty e^{-(s+a)t} \, dt \]

\[ = 0 + \frac{1}{(s+a)^2} \quad \text{if} \quad \text{Re}[s] > -a \]

\[ G(s) = \frac{1}{(s+a)^2}, \quad \text{Re}[s] > -a \]

2. \( G(s) = \int_0^\infty t^2 e^{-at} e^{-st} \, dt \)

Integrating by parts,

\[ G(s) = \frac{-t^2}{s+a} e^{-(s+a)t} \bigg|_0^\infty + \frac{1}{s+a} \int_0^\infty 2t e^{-(s+a)t} \, dt \]

\[ = 0, \quad \text{Re}[s] > -a \]
The second term above is known from part (1) above. Therefore,

\[ G(s) = \frac{2}{(s+a)^3} , \ Re[s] > -a \]

3. The pattern should be clear. In general,

\[ L \left[ t^n e^{-at} \right] = \frac{n!}{(s+a)^{n+1}} , \ Re[s] > -a \]

4. For

\[ f(t) = e^{-(t-a)^2/2b^2} \]

the LT is

\[ F(s) = \int_{-\infty}^{\infty} e^{-(t-a)^2/2b^2} e^{-st} \ dt \]

\[ = \int_{-\infty}^{\infty} e^{-\frac{(t-a)^2}{2b^2} - st} \ dt \]

The exponent is

\[ -\frac{(t-a)^2}{2b^2} - st = -\frac{t^2}{2b^2} + \frac{at}{b^2} - \frac{a^2}{2b^2} - st \]

\[ = -\frac{t^2}{2b^2} + \left(\frac{a}{b^2} - s\right)t - \frac{a^2}{2b^2} \]

Complete the square to obtain

exponent = \(-\frac{1}{2b^2} \left(t - [a - s b^2]\right)^2 + \frac{s^2 b^2}{2} - as\)

Therefore,

\[ F(s) = \int_{-\infty}^{\infty} e^{-\frac{(t-(a-sb^2))^2}{2b^2}} e^{-\frac{s^2 b^2}{2} - as} \ dt \]
\[ G(s) = e^{s^2b^{2/2} - as} \int_{-\infty}^{\infty} e^{-\left[ t - (a-sb^2) \right]^{2}/2b^2} \, dt \]

The integral above has integrand which is a Gaussian. Therefore,

\[ G(s) = e^{s^2b^{2/2} - as} \cdot \sqrt{2\pi} \cdot b \]

Factors required to normalize Gaussian.

The integral converges for all \( s \), because the exponent is dominated by \(-t^2/2b^2\). Therefore,

\[ F(s) = \sqrt{2\pi} \cdot b \cdot e^{s^2b^{2/2} - as}, \quad \text{all } s \]
1. What are doubly linked lists? What is the record declaration for a node in a doubly linked list?

Doubly linked lists have two pointers instead of the single pointer seen in singly linked lists. The pointers point to both the previous node in the list as well as the next node in the list.

```plaintext
type Listnode is
  record
    Element : Elementtype;
    Next    : Listptr;
    Prev    : Listptr;  -- this is the change made to singly linked lists
  end record;
```

b. Write an algorithm to insert a node into a sorted doubly linked list. Use a diagram to show the sequence of operations that have to be performed to carry out the insertion step.

Hint: Extend the approach used in class/notes for singly linked lists.

Preconditions:
1. User passes the list (called List) and the element to be inserted (called Element) to the insert procedure
2. List is already sorted

Postconditions:
1. Procedure returns the list with the element inserted in the correct position
2. List remains sorted

Algorithm:

Create three temporary Listptrs Current, Previous and NewNode

Previous := null
Current := List.Head;
NewNode := new Listnode;
NewNode.Element := Element
Consider the doubly linked list shown in Figure 1. The insertion of a node with element 12 in the list, the insert operation is shown in Figures 2 and 3.

Figure 2. Prior to Insertion
NewNode.Next := Current

NewNode.Prev := Previous

Previous.Next := NewNode
Current.Prev := NewNode

![Diagram of after insertion operation]

Figure 3. After Insertion Operation

c. Implement your algorithm as an Ada95 program.

Package Specification


Checking: c:/docume~1/jayaka~1/mydocu~1/16070/code/doubly_linked_list.ads (source file time stamp: 2004-03-31 20:46:40)

1. -- Specification for doubly-linked lists
2. -- Specified: Jayakanth Srinivasan
3. -- Last Modified: February 11, 2004
4. --
5. package Doubly_Linked_List is
6. subtype Elementtype is Integer;
7. type Listnode is
8. record
9. Element : Elementtype;
10. Next : Listptr;
11. Prev : Listptr; -- this is the change made to singly linked lists
12. end record;
13. type Listptr is access Listnode;
14. type List is
15. record
16. Head : Listptr;
17. end record;
18. type List is
19. record
20. Head : Listptr;
21. end record;
procedure Makeempty (  
    L : in out List );
-- Pre:  L is defined
-- Post: L is empty

function Isempty (  
    L : in List )
return Boolean;
-- Pre:  L is defined
-- Post: returns True if L is empty, False otherwise

procedure Display (  
    L : in List );
-- Pre: L may be empty
-- Post: displays the contents of L's Element fields, in the
-- order in which they appear in L

procedure Initialize (  
    L : in out List );
-- Pre: L may be empty
-- Post: Elements inserted into the list at correct position

procedure Insert_In_Order (  
    L       : in out List;
    Element : in Elementtype );
-- Post: Elements inserted into the list at correct position

end Doubly_Linked_List;


---

Package Implementation


-------------------------------------------------------------------------------
-- Implementation for doubly-linked lists
-- Programmer: Jayakanth Srinivasan
-- Last Modified: Feb 11, 2004
-------------------------------------------------------------------------------

with Ada.Text_Io;
with Ada.Integer_Text_Io;
with Ada.Unchecked_Deallocation;

use Ada.Text_Io;
use Ada.Integer_Text_Io;

package body Doubly_Linked_List is
  -- create an instance of the free procedure
  procedure Free is
    new Ada.Unchecked_Deallocation(Listnode, Listptr);
  -- check if list is empty. List.Head will be null
  function Isempty (  
    L : in List )
return Boolean is

---
begin
if L.Head = null then
    return True;
else
    return False;
end if;
end Isempty;

-- free all allocated memory at the end of the program
procedure Makeempty ( L : in out List ) is
    Temp : Listptr;
begin
    loop
        exit when Isempty(L);
        Temp := L.Head;
        L.Head := Temp.Next;
        Free(Temp);
    end loop;
    L.Head := null;
end Makeempty;

-- initialize the list by setting the head pointed to null
procedure Initialize ( L : in out List ) is
begin
    L.Head := null;
end Initialize;

-- displays the contents of the list
procedure Display ( L : in List ) is
    Temp : Listptr;
begin
    -- set the pointer to the head of the node
    Temp:= L.Head;
    while Temp /= null loop
        Put(Temp.Element);
        Put(", ");
        -- move pointer to the next node
        Temp := Temp.Next;
    end loop;
    New_Line;
end Display;

-- insert elements in ascending order
-- this procedure added :)
procedure Insert_In_Order ( L : in out List;
    Element : in Elementtype ) is
Current, Previous, Newnode : Listptr;
begin
    Current := L.Head;
    Previous := null;

80. -- create a node and set the data to element
81. Newnode := new Listnode;
82. Newnode.Element := Element;
83. -- check if the list is empty.
84. if Isempty(L) = False then
85.   loop
86.     -- need two separate exits, otherwise there will be
87.     -- an exception at runtime
88.     exit when Current = null;
89.     exit when Current.Element > Element;
90.     Previous := Current;
91.     Current := Current.Next;
92.   end loop;
93. end if;
94. -- do insertion
95. Newnode.Prev := Previous;
96. Newnode.Next := Current;
97. if Previous = null then
98.   -- list is empty
99.   L.Head := Newnode;
100. else
101.   Previous.Next := Newnode;
102.   if Current /= null then
104. end if;
105. end if;
106. end Insert_In_Order;
107.
108. end Doubly_Linked_List;

108 lines: No errors

Test Program


Compiling: c:/docume~1/jayaka~1/mydocu~1/16070/code/doubly_list_test.adb (source file time stamp:
2004-03-31 20:49:02)

1. -----------------------------------------------
2. -- Program to test the doubly linked list package
3. -- Programmer: Jayakanth Šrīnivasan
4. -- Date Last Modified : Feb 11, 2004
5. -----------------------------------------------
6.
7. with Doubly_Linked_List;
8. use Doubly_Linked_List;
9.
10. procedure Doubly_List_Test is
11.   My_List : List;
12.
13. begin
14.   -- initialize the list
15.   Initialize(My_List);
16.
17.   -- insert 1,3,2,4 into the list and display at each stage
18.   Insert_In_Order(My_List, 1);
2. What is the Shortest Path through the graph shown below using Dijkstra’s algorithm.

Assume node A is the start node.

Show all the steps in the computation of the shortest path.

Initialize

\[ V = \{a, b, c, d, e\} \]
\[ E = \{(a,b), (a,c), (b,e), (b,c), (c,b), (c,d), (c,e), (d,a), (d,e), (e,d)\} \]
\[ S = \{\rightarrow\} \]
\[ Q = \{a, b, c, d, e\} \]

\[ D = [0, \infty, \infty, \infty, \infty] \]
\[ \text{Previous} = [0, 0, 0, 0, 0] \]

Start at A

- Relax (a,b,10)
- Relax (a,c,5)

\[ S = \{a\} \]
\[ Q = \{b, c, d, e\} \]
D = \[0, 10, 5, \infty, \infty\]
Previous = \[0,a, a, 0, 0\]

Move to C
Relax(c,b,2)
Relax (c,d,2)
Relax(c,e,9)

\[S = \{a,c\}\]
\[Q = \{b, d, e\}\]

D = \[0, 7, 5, 7, 14\]
Previous = \[0,c, a, c, c\]

Move to B (you can also move to D)
Relax(b,e,1)
Relax(b,c,3)

\[S = \{a, c, b\}\]
\[Q = \{d, e\}\]

D = \[0, 7, 5, 7, 8\]
Previous = \[0, c, a, c, b\]

Move to D
Relax(d,a,7)
Relax(d,e,6)

\[S = \{a, c, b, d\}\]
\[Q = \{e\}\]

D = \[0, 7, 5, 7, 8\]
Previous = \[0, c, a, c, b\]

Move to E
Relax(e,d,4)

\[S = \{a, c, b, d, e\}\]
\[Q = \{\}\]

D = \[0, 7, 5, 7, 8\]
Previous = \[0, c, a, c, b\]

3. Define the following terms (as applied to graphs):
For a graph $G = (V, E)$, where $V$ is a finite nonempty set of vertices and $E$ is the set of edges,

a. **Walk**
   A walk is a sequence of vertices $(v_1, v_2, \ldots, v_k)$ in which each adjacent vertex pair is an edge.

b. **Path**
   A path is a walk with no repeated nodes.

c. **Eulerian Path**
   An Eulerian path in a graph is a path that uses each edge precisely once.

d. **Cycle**
   A cycle is a path that begins and ends with the same vertex.

e. **Degree of a vertex**
   The degree of a vertex in an undirected graph is the number of edges incident with it, except that a loop at a vertex contributes twice to the degree of that vertex.

   In a directed graph, the degree of the vertex is partitioned into **indegree** (number of edges entering a vertex) and **outdegree** (number of edges leaving a vertex). Note that a loop contributes to both the indegree and the outdegree.