La Transformada de Laplace en ED

Capítulo 2 -Libro (2.7) **+** ***Hoja de problemas 7*** **+** ***Complementos***

Resumen de contenidos:

- **La transformada de Laplace y su inversa**
- **Aplicaciones a problemas de valores iniciales con ED y sistemas: modelos de resortes lineales ("fuerzas instantáneas")**
- **Aplicaciones a problemas de contorno: modelos de vigas ("cargas puntuales")**
- **Algunas soluciones con "*dsolve*"**

% En este cuaderno se aplica la transformada de Laplace para resolver ecuaciones diferenciales. laplace(f) nos da la transformada de Laplace de una función f de la variable simbólica t; laplace(f) es una función de la variable simbólica s. Recuperamos f con ilaplace(laplace(f)).

% Su aplicación en ED (sistemas de ED) lineales, de coeficientes constantes, nos lleva a resolver ecuaciones algebraicas (sistemas algebraicos, respectivamente). Es conveniente usar la transformada de Laplace en la resolución de modelos con ED, por ejemplo, cuando aparecen términos independientes en los que intervienen ``funciones" Heaviside y delta de Dirac. Si se puede, comparamos la solución obtenida aplicando esta transformada y con "dsolve". También, si hay términos independientes en la ED que son funciones discontinuas, analizamos la regularidad de la solución.

% Por defecto t es la variable para definir f y s la variable para definir la transformada inversa

>> syms t

>> syms s

>> laplace(exp(t)) *% transformada de Laplace de exp(t)*

ans =

1/(s - 1)

```
>> laplace(sin(25*t)*exp(78*t)) % transformada de Laplace de f=sin(5t)exp(78t)

ans =

25/((s - 78)^2 + 625)

>> pretty(ans)

       25
  ---------------
         2
  (s - 78)  + 625

>> ilaplace(ans) % recuperamos la función de partida

ans =

sin(25*t)*exp(78*t)

 >> laplace(1)

Undefined function 'laplace' for input arguments of type 'double'.

% no entiende que f=1 es una función, para indicárselo empleamos sym(1)

>> laplace(sym(1)) % transformada de Laplace de f=1

ans =

1/s

>> laplace(heaviside(t))

ans =

1/s

>> laplace(dirac(t))

ans =

1
```

>> laplace(dirac(t-pi))

ans =

1/exp(pi*s)

>> help laplace

```
--- help for sym/laplace ---

 laplace Laplace transform.
    L = laplace(F) is the Laplace transform of the scalar sym F with
    default independent variable t.  The default return is a function
    of s.  If F = F(s), then laplace returns a function of t:  L = L(t).
    By definition L(s) = int(F(t)*exp(-s*t),0,inf), where integration
    occurs with respect to t.

    L = laplace(F,t) makes L a function of t instead of the default s:
    laplace(F,t) <=> L(t) = int(F(x)*exp(-t*x),0,inf).

    L = laplace(F,w,z) makes L a function of z instead of the
    default s (integration with respect to w).
    laplace(F,w,z) <=> L(z) = int(F(w)*exp(-z*w),0,inf).

    Examples:
       syms a s t w x
       laplace(t^5)            returns   120/s^6
       laplace(exp(a*s))       returns   1/(t-a)
       laplace(sin(w*x),t)     returns   w/(t^2+w^2)
       laplace(cos(x*w),w,t)  returns   t/(t^2+x^2)
       laplace(x^sym(3/2),t)  returns   3/4*pi^(1/2)/t^(5/2)
       laplace(diff(sym('F(t)')))   returns   laplace(F(t),t,s)*s-F(0)

    See also sym/ilaplace, sym/fourier, sym/ztrans.

    Reference page in Help browser
       doc sym/laplace
```

>> help dirac

```
dirac  Delta function.
     dirac(X) is zero for all X, except X == 0 where it is infinite.
     dirac(X) is not a function in the strict sense, but rather a
     distribution with int(dirac(x-a)*f(x),-inf,inf) = f(a) and
     diff(heaviside(x),x) = dirac(x).
     See also heaviside.
```

>> help heaviside

```
heaviside    Step function.
     heaviside(X) is 0 for X < 0, 1 for X > 0, and .5 for X == 0.
     heaviside(X) is not a function in the strict sense.
     See also dirac.
```

% ***Pasos para resolver una ED con la transformada de Laplace:*** *de la ecuación diferencial con incógnita y(t) hay que llegar a una ecuación algebraica con incógnita laplace(y). Para ello tomamos transformada de Laplace en la ED, y, aplicamos las propiedades de derivación y linealidad de dicha transformada. Luego, se despeja laplace(y) y se identifica, si se puede, con transformadas de Laplace de funciones conocidas (de manera general se utilizan tablas de transformadas; aquí el comando ilaplace en Matlab nos las proporciona)*

% Resolvemos ejercicios de la hoja 7 de problemas. Con Matlab seguimos los pasos:
1.- Definimos las variables simbólicas s, t, ly. Por defecto: t variable de definición de f, s variable de laplace(f), ly va a almacenar la transformada de Laplace de la solución de la ED (s es lambda en el libro de apuntes).
2.- Se introduce la ecuación algebraica y se resuelve: usamos "solve" para despejar ly.
3.- Se utiliza "ilaplace" para encontrar la transformada inversa

% ***ED de primer orden***. ***Ejercicio 1.1:*** *y'-y=1, y(0)=45*

>> syms s t ly

>> solve(ly*s-45-ly-laplace(sym(1)),ly) *% es la transformada de Laplace de la solución*

ans =

(1/s + 45)/(s - 1)

>> ilaplace(ans) *% es la solución*

ans =

46*exp(t) – 1

% obteniendo la solución con dsolve

>> dsolve('Dy-y=1','y(0)=45')

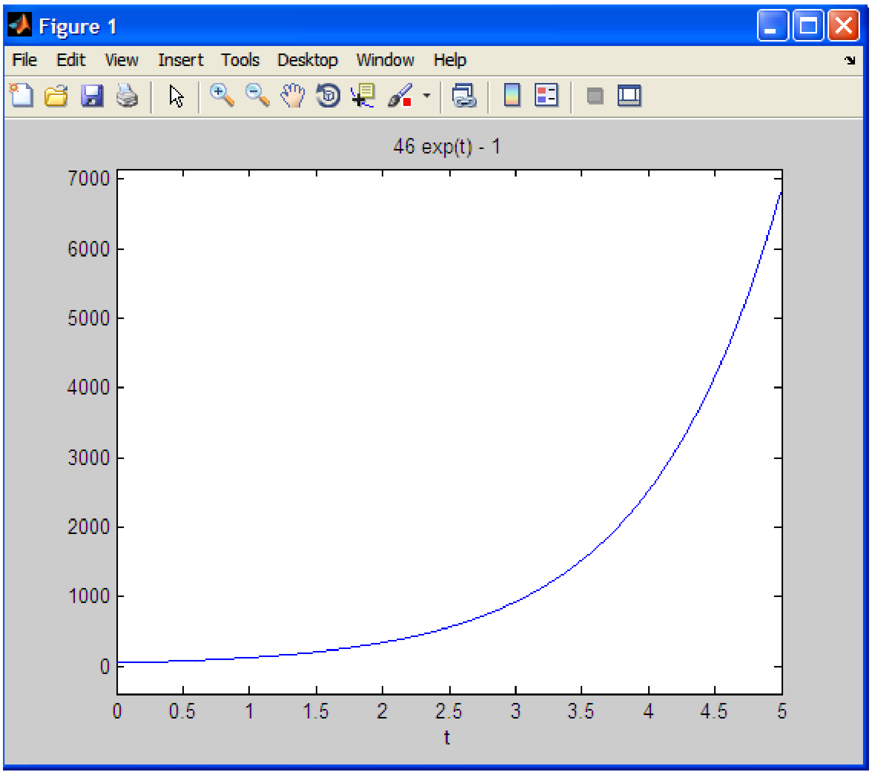
ans =

46*exp(t) - 1

% vemos que da el mismo resultado que con transformada de Laplace.

% Hacemos la gráfica en un intervalo en que esté definida la solución

>> ezplot(ans,[0,5])



*% **Ecuaciones diferenciales de segundo orden**, lineales, de coeficientes constantes, que intervienen, por ejemplo, en modelos de resortes mecánicos o circuitos eléctricos*

ay''+by'+cy=f.

La manera de proceder es: tomando transformada de Laplace en ambos miembros, llegamos a la ecuación algebraica a resolver

a(s^2*ly-s*y(0)-y'(0))+ b*(s*ly-y(0))+c*ly=laplace(f);*

para a=1, despejamos ly con
*solve(s^2*ly-s*y(0)-y'(0)+ b*(s*ly-y(0))+c*ly-laplace(f),ly)*
Finalmente ilaplace(ans) nos da la solución. Se trata de problemas de valores iniciales.

*% **Ejercicio 1.2**: D2y-3*Dy+2*y=4*t*exp(3*t), (0)=1, Dy(0)=-1*

>> syms t s ly

>> solve(ly*s^2-s+1-3*(s*ly-1)+2*ly-laplace(4*t*exp(3*t)),ly)

ans =

(s + 4/(s - 3)^2 - 4)/(s^2 - 3*s + 2)

>> sol= ilaplace(ans)

sol =

2*exp(2*t) - 3*exp(3*t) + 2*exp(t) + 2*t*exp(3*t)

% comparando con solución exacta

>> dsolve('D2y-3*Dy+2*y=4*t*exp(3*t)','y(0)=1','Dy(0)=-1')

ans =

2*exp(2*t) + 2*exp(t) + 4*exp(3*t)*(t - 1) - exp(3*t)*(2*t - 1)

>> ans-sol

ans =

3*exp(3*t) - 2*t*exp(3*t) + 4*exp(3*t)*(t - 1) - exp(3*t)*(2*t - 1)

% Obviamente, se tiene la misma solución con dsolve y usando la transformada de Laplace, dado que los datos son funciones continuas; para verlo utilizamos "simplify "o "simple"

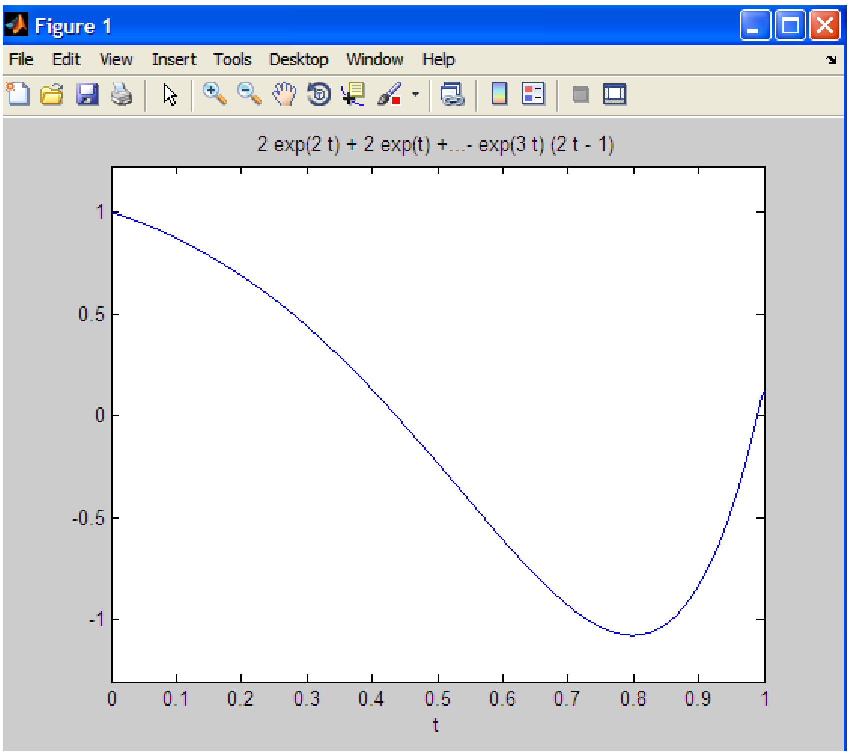
>> simplify (ans)

ans =

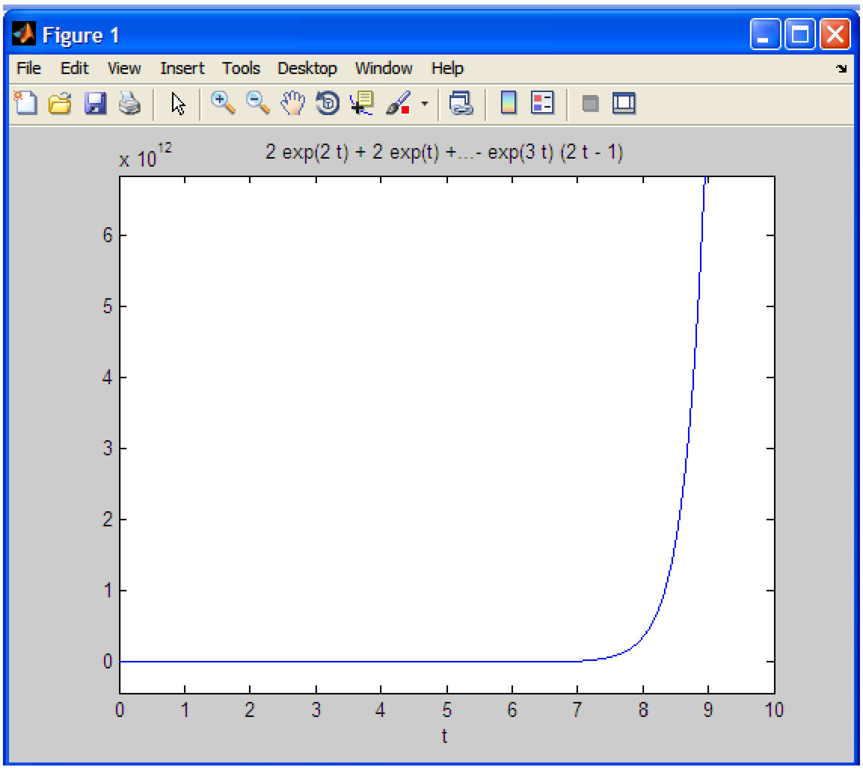
0

% Hacemos una gráfica de la solución en distintos intervalos

>> ezplot(sol,[0,1])



>> ezplot(sol,[0,10])



*% __**Ejercicio 1.3**__: y''+2y'+2y=3*delta(t-pi), y(0)=1, y'(0)=0.*
***Modelo de resorte lineal** al que se le aplica una **fuerza instantánea** en t=pi*

>> clear all

>> syms s t ly

>> solve(s^2*ly-s*1+ 2*(s*ly-1)+2*ly-laplace(3*dirac(t-pi)),ly)

ans =

(s + 3/exp(pi*s) + 2)/(s^2 + 2*s + 2)

>> ilaplace(ans)

ans =

(cos(t) - sin(t))/exp(t) + (2*sin(t))/exp(t) - 3*exp(pi - t)*sin(t)*heaviside(t - pi)

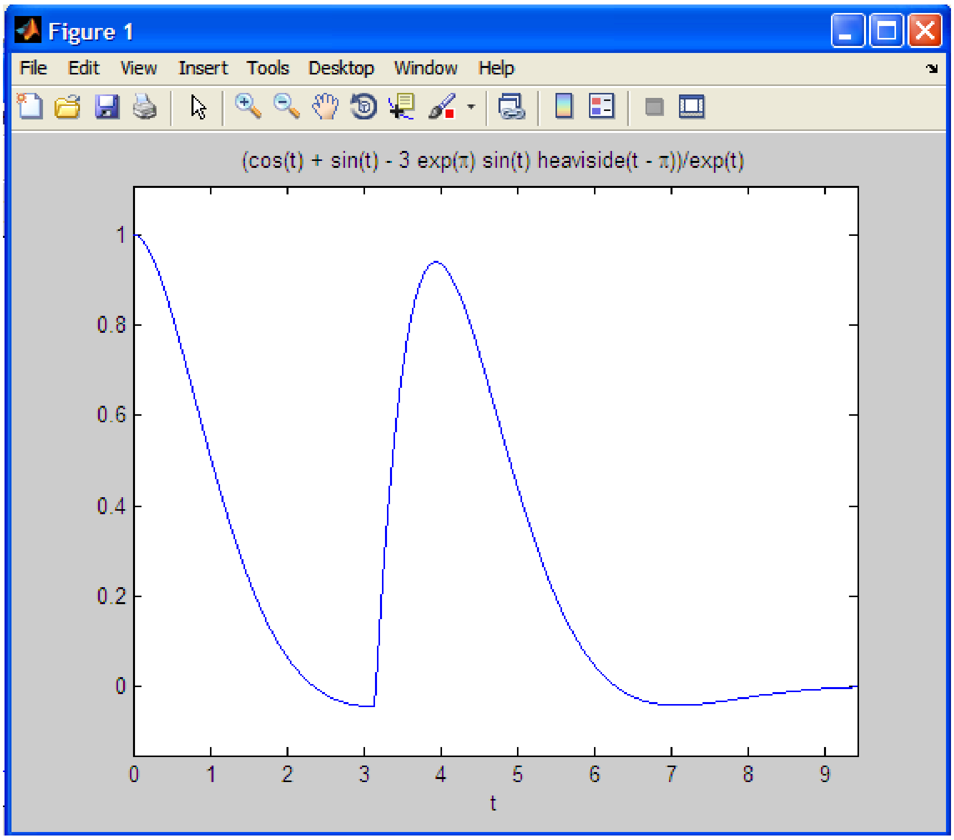
>> solucion= simplify(ans)

ans =

(cos(t) + sin(t) - 3*exp(pi)*sin(t)*heaviside(t - pi))/exp(t)

% la solucion toma el valor:
cos(t)+sin(t) si t< pi,
*(cos(t) + sin(t) - 3*exp(pi)*sin(t))/exp(t) si t> pi.*
Se trata de una función continua en cualquier intervalo, derivable con continuidad para t<pi y t>pi, pero no en t=pi como se comprueba abajo

>> ezplot(ans, [0,3*pi])



% la gráfica muestra que se trata de una función continua, derivable con continuidad para t<pi y t>pi, y no en t=pi (debido a la "fuerza" aplicada en este instante). No obstante, se comprueba también derivando: demostramos que todas las funciones que aparecen son continuas para t distinto de pi

>> diff(solucion)

ans =

- (cos(t) + sin(t) - 3*exp(pi)*sin(t)*heaviside(t - pi))/exp(t) - (sin(t) - cos(t) + 3*exp(pi)*cos(t)*heaviside(t - pi) + 3*exp(pi)*sin(t)*dirac(t - pi))/exp(t)

>> subs(ans,pi)

ans =

-Inf

*% **Ejercicio 1.4**: y''+y= - u(t)+u(t-1), y(0)=0, y'(0)=0*
***Modelo de resorte** no amortiguado, sobre el que actúa una fuerza solo durante un intervalo de tiempo: t<1.*

% Resolvemos primero con la versión 2008 de Matlab, luego con la versión 2011, y con dsolve, comparando soluciones (que van a coincidir)

>> syms s t ly

>> solve(s^2*ly+ly+laplace(heaviside(t))-laplace(heaviside(t-1)),ly)

ans =

(-1+exp(-s))/s/(s^2+1)

>> ilaplace(ans)

ans =

-1+cos(t)+2*heaviside(t-1)*sin(1/2*t-1/2)^2

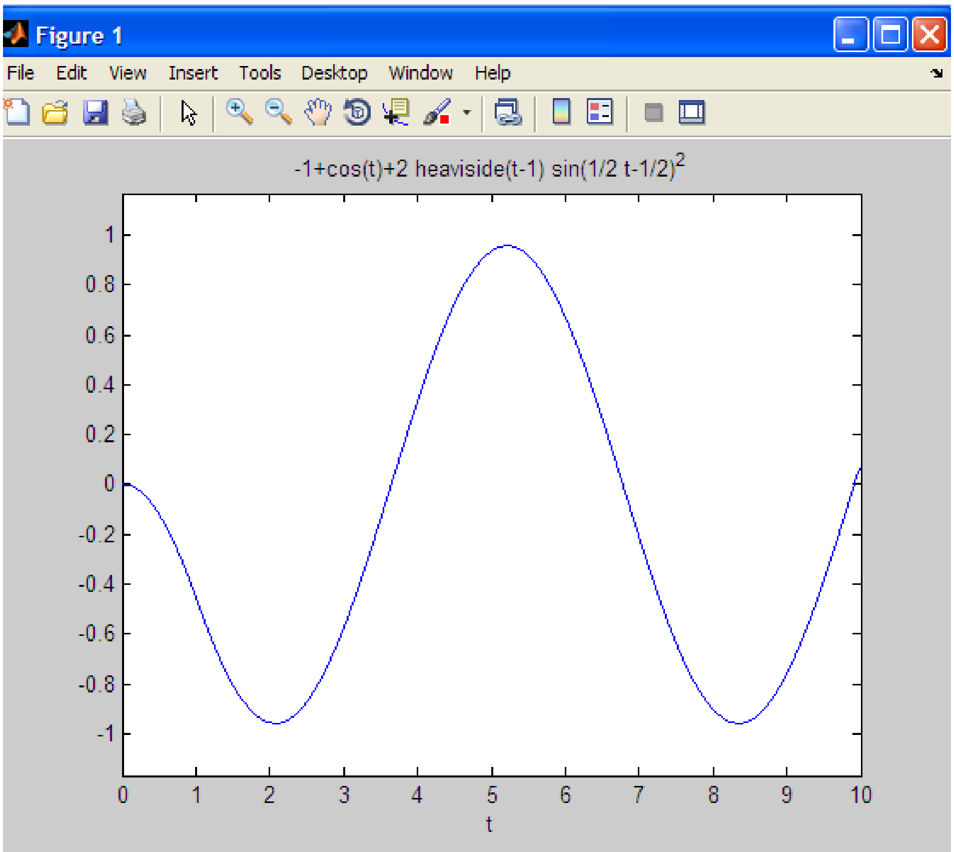
% función que vale
-1+cos(t) si t<1,
-1+cos(t)+2 sin(1/2*t-1/2)^2 si t>1.*
Se trata de una función continua, que vale -1+cos(1) en t=1; las derivadas en t=1 por la derecha y por la izquierda también coinciden: es derivable con continuidad. Sin embargo, la derivada segunda no es continua en t=pi, como comprobamos abajo

>> solucion1= ans

solucion =

-1+cos(t)+2*heaviside(t-1)*sin(1/2*t-1/2)^2

>> ezplot(solucion1,[0,10])



% en la gráfica, la solución parece una función bastante regular, pero no lo es tanto…

>> diff(solucion1)

ans =

-sin(t)+2*dirac(t-1)*sin(1/2*t-1/2)^2+2*heaviside(t-1)*sin(1/2*t-1/2)*cos(1/2*t-1/2)

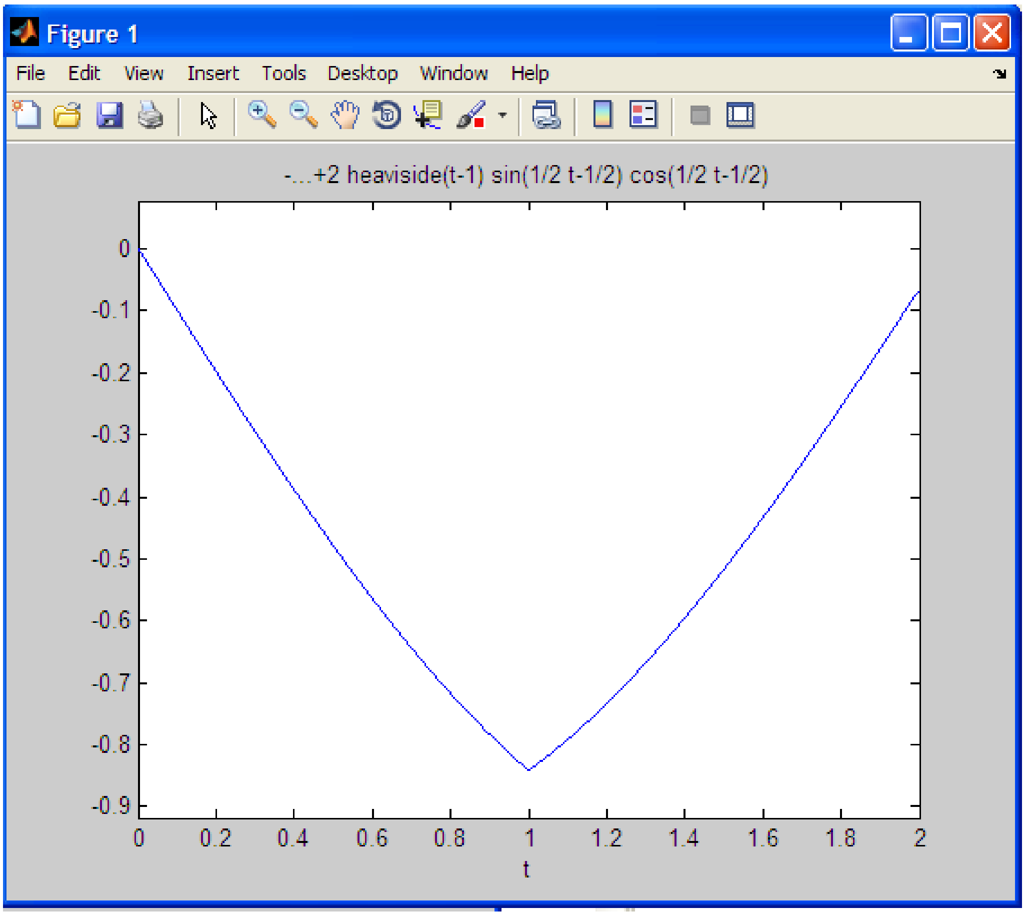
>> simplify(ans)

ans =

-sin(t)+2*heaviside(t-1)*sin(1/2*t-1/2)*cos(1/2*t-1/2)

% diff(solucion1) es continua pero no es derivable con continuidad en t=1, como se visualiza en la siguiente gráfica

>> ezplot(diff(solucion1),[0,2])



% Repetimos el ejercicio con la versión 2011 de Matlab para detectar problemas que pueden ocurrir con esta versión

>> syms s t ly

>> LY=solve(s^2*ly+ly+laplace(heaviside(t))-laplace(heaviside(t-1)),ly)

LY =

-(exp(s) - 1)/(s^3*exp(s) + s*exp(s))

>> ilaplace(LY)

ans =

ilaplace(1/(s^3*exp(s) + s*exp(s)), s, t) - ilaplace(exp(s)/(s^3*exp(s) + s*exp(s)), s, t)

% vemos que Matlab no ha calculado la transformada inversa, sin embargo, sí que lo hace, si le pedimos, primero, que simplifique LY, y luego que calcule la inversa

>> simplify(LY)

ans =

(1/exp(s) - 1)/(s*(s^2 + 1))

>> solucion=ilaplace(ans)

solucion =

cos(t) - heaviside(t - 1)*(cos(t - 1) - 1) – 1

% Nota : Matlab 2011 no reconoce directamente :
*ilaplace(solve(s^2*ly+ly+laplace(heaviside(t))-laplace(heaviside(t-1)),ly)) ;*
hay que simplificar la expresión primero. Además, expresa la solución de manera distinta que con la versión 2008 (aunque es la misma solución). Comprobamos, igual que antes, que se trata de una función continua y derivable en t=1

% la solución vale:
cos(t) – 1 para t<1,
cos(t) - cos(t - 1) para t>1
cos(1)- 1 , en t=1

>> diff(solucion)

ans =

heaviside(t - 1)*sin(t - 1) - dirac(t - 1)*(cos(t - 1) - 1) - sin(t)

>> simplify(ans)

ans =

heaviside(t - 1)*cos(1)*sin(t) - sin(t) - heaviside(t - 1)*sin(1)*cos(t)

% Demostramos que esta derivada es continua en t=1, pero no derivable en ese punto:

>> diff(ans)

ans =

heaviside(t - 1)*cos(1)*cos(t) - cos(t) + dirac(t - 1)*cos(1)*sin(t) - dirac(t - 1)*sin(1)*cos(t) + heaviside(t - 1)*sin(1)*sin(t)

>> simplify(ans)

ans =

cos(t - 1)*heaviside(t - 1) - cos(t)

% obviamente discontinua en t=1

*% En este ejercicio, como se ha indicado antes, no parece tenerse la misma solución con las dos versiones de Matlab , pero se demuestra que sí se tiene: la solución coincide en ambas versiones, como se comprueba sin más que considerar la diferencia cos(t)-1+2*heaviside(t-1)*sin(1/2*t-1/2)^2 -(cos(t) - heaviside(t - 1)*(cos(t - 1) -1) -1), esto es:*

>> solucion1-solucion

ans =

2*heaviside(t-1)*sin(1/2*t-1/2)^2+heaviside(t-1)*(cos(t-1)-1)

% que aparentemente no nos lo iguala a 0, pero, analizando el resultado, se ve que es 0.
Con los comandos simplify y simple, se ve más rápido

>> simplify(ans)

ans =

heaviside(t-1)*(1-2*cos(1/2*t-1/2)^2+cos(t-1))

>> simple(ans) *% nos da cero*

....

ans =

0

% Resolvemos con dsolve

>> u=dsolve('D2y+y=-heaviside(t)+heaviside(t-1)','y(0)=0','Dy(0)=0')

u =

- sin(t)*(heaviside(t)*sin(t) + heaviside(t - 1)*(sin(1) - sin(t))) - cos(t)*(heaviside(t)*(cos(t) - 1) + heaviside(t - 1)*(cos(1) - cos(t)))

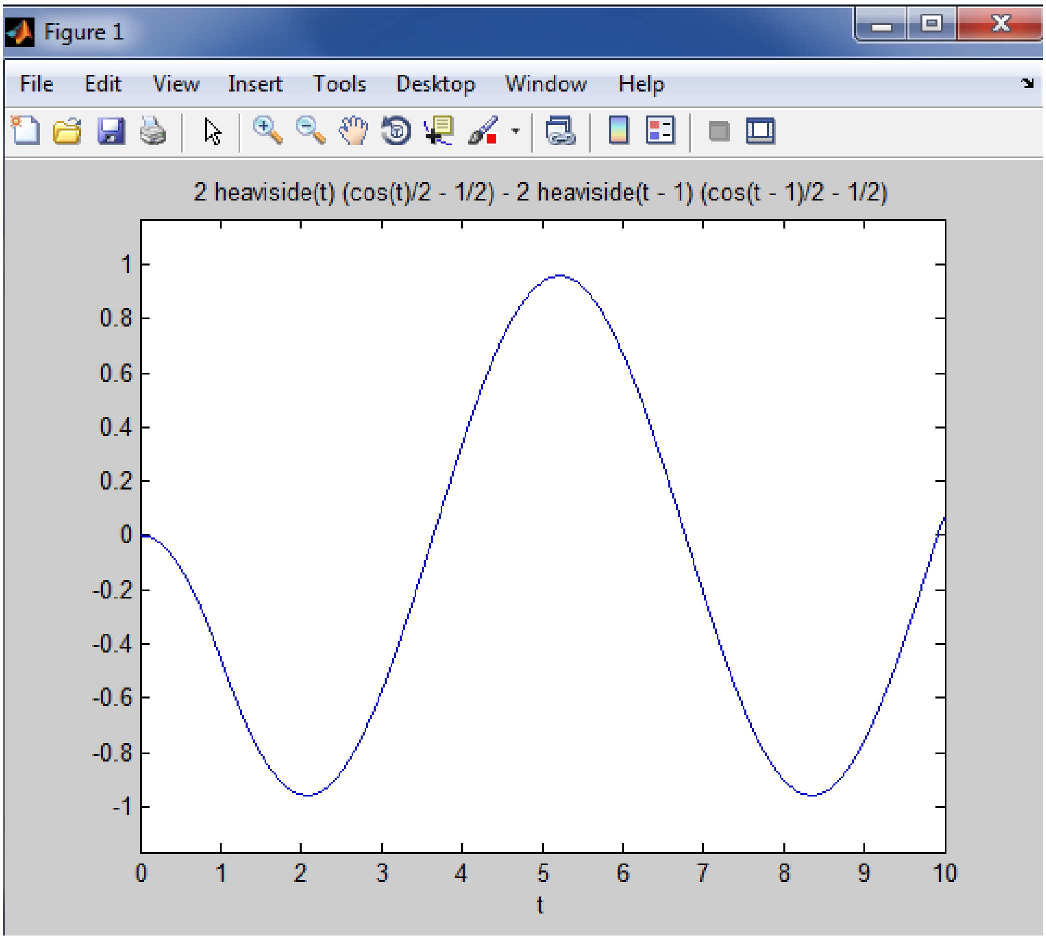
>> simple(u)

....

ans =

2*heaviside(t)*(cos(t)/2 - 1/2) - 2*heaviside(t - 1)*(cos(t - 1)/2 - 1/2)

>> ezplot(ans,[0,10]) *% nos da la misma grafica de antes*



% las funciones u / solucion / solucion1 tienen aparentemente la misma gráfica en[0,10], pero para comprobar que coinciden utilizamos

>> simple(u-solucion)

……………………

ans =

(cos(t) - 1)*(heaviside(t) - 1)

% que, de nuevo observamos no es cero, pero sí lo es para t>0, es decir, ambas soluciones obtenidas con dsolve y con transformada de Laplace coinciden para t>0; también coinciden para t=0.

*% **Problema de contorno. Ejercicio 1.5**: y’’= f(x), y(0)=0, y(3)=0, donde f(x)= 2x (heaviside(x)-heaviside(x-1)) + (3-x) (heaviside(x-1)-heaviside(x-3))*

% para ver que tiene solución única, aplicamos el teorema de la alternativa de Fredholm, demostrando que el problema homogéneo asociado (y’’=0, y(0)=y(3)=0) tiene solo la solución trivial: utilizamos dsolve

>> dsolve('D2y=0','y(0)=0','y(3)=0')

ans =

0

% aplicamos la transformada de Laplace para resolver y’’=f, y(0)=0, y’(0)=a, donde la constante a se determinará imponiendo y(3)=0.

>> syms s t ly

>> syms a

% a es una constante que almacena y'(0), ly denota la transformada de Laplace de la solución; ly depende de a y es función de la variable s

>> LY=solve(s^2*ly-a-laplace(2*t * (heaviside(t)-heaviside(t-1)) + (3-t)* (heaviside(t-1)-heaviside(t-3))), ly)

LY =

(1/exp(3*s) - 3/exp(s) + a*s^2 + 2)/s^4

>> solu_a=ilaplace(LY)

solu_a =

a*t - (heaviside(t - 1)*(t - 1)^3)/2 + (heaviside(t - 3)*(t - 3)^3)/6 + t^3/3

% solución de la ED que depende de a. Imponemos que y(3)=0

>> subs(solu_a,t,3)

ans =

3*a + 5

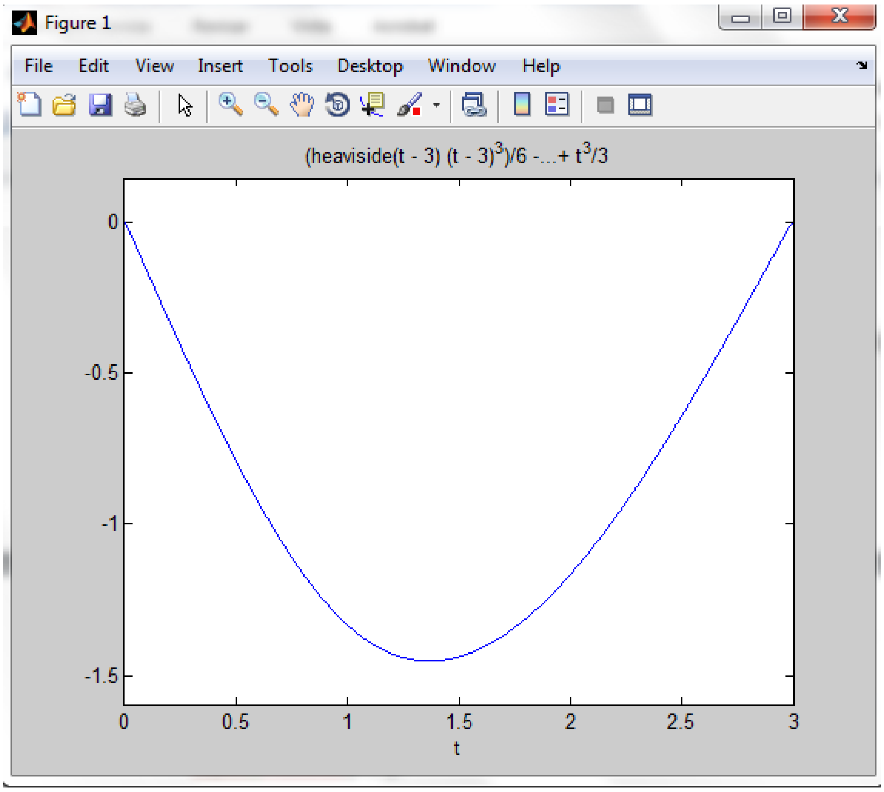
% despejando a, a=-5/3, se substituye en la solución encontrada

>> solucion=subs(solu_a,a,-5/3)

solucion =

(heaviside(t - 3)*(t - 3)^3)/6 - (heaviside(t - 1)*(t - 1)^3)/2 - (5*t)/3 + t^3/3

>> ezplot(solucion,[0,3])



% Problemas de contorno: ***Modelos de vigas con cargas puntuales.***

% Modelo de segundo orden: y''=delta(x-1), y(0)=y(2)=0.
Sea A=y'(0) (a determinar en el proceso)

>> syms A s t ly

>> solve(s^2*ly-A-laplace(dirac(t-1)),ly)

ans =

(A + 1/exp(s))/s^2

>> solucionA=ilaplace(ans)

solucionA =

A*t + heaviside(t - 1)*(t - 1)

% solución que verifica y(0)=0, y'(0)=A; encontramos A tal que y(2)=0.
Primero se impone que y(2)=0, y de aquí se despeja A, que se substituye en solucionA

>> subs(solucionA,t,2)

ans =

2*A + 1

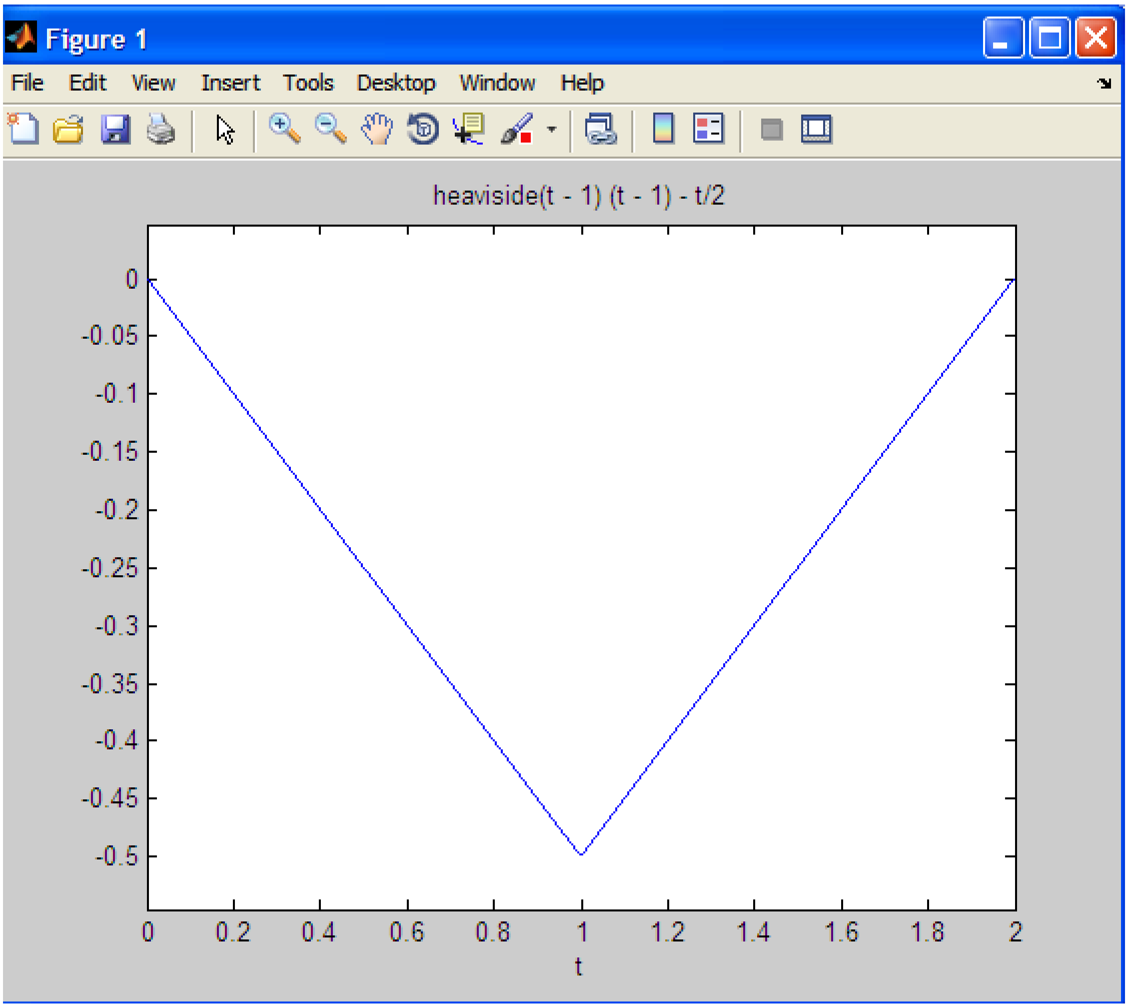
% así A=-1/2

>> solucion=subs(solucionA,A,-1/2)

solucion =

heaviside(t - 1)*(t - 1) - t/2

>> ezplot(solucion,[0,2]) *% intervalo determinado por la longitud de la viga*



*% **ED de cuarto orden. Ejercicio 1.6:** y''''=delta(x-1), y(0)=y'(0)=0, y''(2)=y'''(2)=0*

% Para empezar con ED de cuarto orden y su resolución mediante la transformada de Laplace, proponemos resolver un problema de Cauchy:
y''''=delta(t-pi), y(0)=0, y'(0)=0, y''(0)=1, y'''(0)=1

>> clear all

>> syms t s ly

>> solve(s^4*ly-s-1-laplace(dirac(t-1)),ly)

ans =

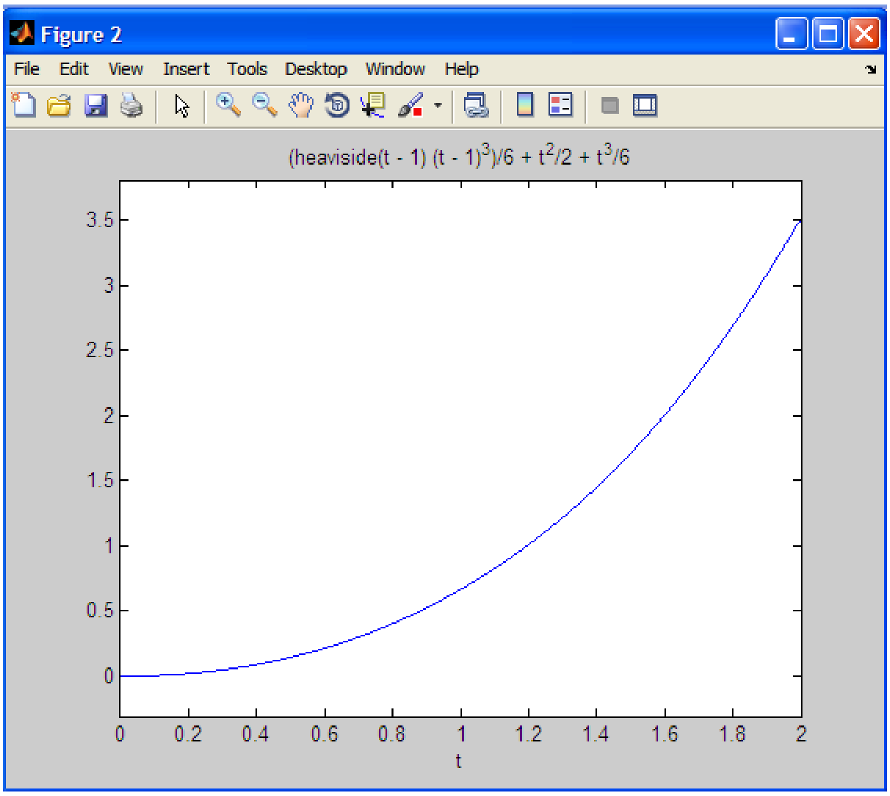
(s + 1/exp(s) + 1)/s^4

>> ilaplace(ans)

ans =

(heaviside(t - 1)*(t - 1)^3)/6 + t^2/2 + t^3/6

>> ezplot(ans,[0,2])



*% El problema de contorno y^(4)= delta(t-1), y(0)=0,y'(0)=0,y''(2)=0,y'''(2)=0,
se reduce a uno de Cauchy, con condiciones iniciales y(0)=0, y'(0)=0, y''(0)=A, y'''(0)=B.
Se resuelve en términos de las constantes A y B, y se determinan estas constantes de forma que se verifique y''(0)=0 e y'''(0)=0*

>> syms A B s t ly

>> solve(s^4*ly-A*s-B-laplace(dirac(t-1)),ly)

ans =

(B + 1/exp(s) + A*s)/s^4

>> soluA_B=ilaplace(ans)

soluA_B =

(A*t^2)/2 + (B*t^3)/6 + (heaviside(t - 1)*(t - 1)^3)/6

*% soluA_B es la solución de y''''=dirac(t-1), verificando y(0)=y'(0)=0, y''(0)=A, y'''(b)=B,
para constantes A y B que determinamos imponiendo que y''(2)=0 e y'''(2)=0*

>> diff(soluA_B,2)

ans =

A + B*t + dirac(t - 1)*(t - 1)^2 + (dirac(t - 1, 1)*(t - 1)^3)/6 + (heaviside(t - 1)*(2*t - 2))/2

>> subs(diff(soluA_B,2),t,2)

ans =

A + 2*B + 1

>> subs(diff(soluA_B,2),t,3)

ans =

A + 3*B + 2

*% igualando a cero estas dos ecuaciones algebraicas
A + 2*B + 1=0
A + 3*B + 2=0*

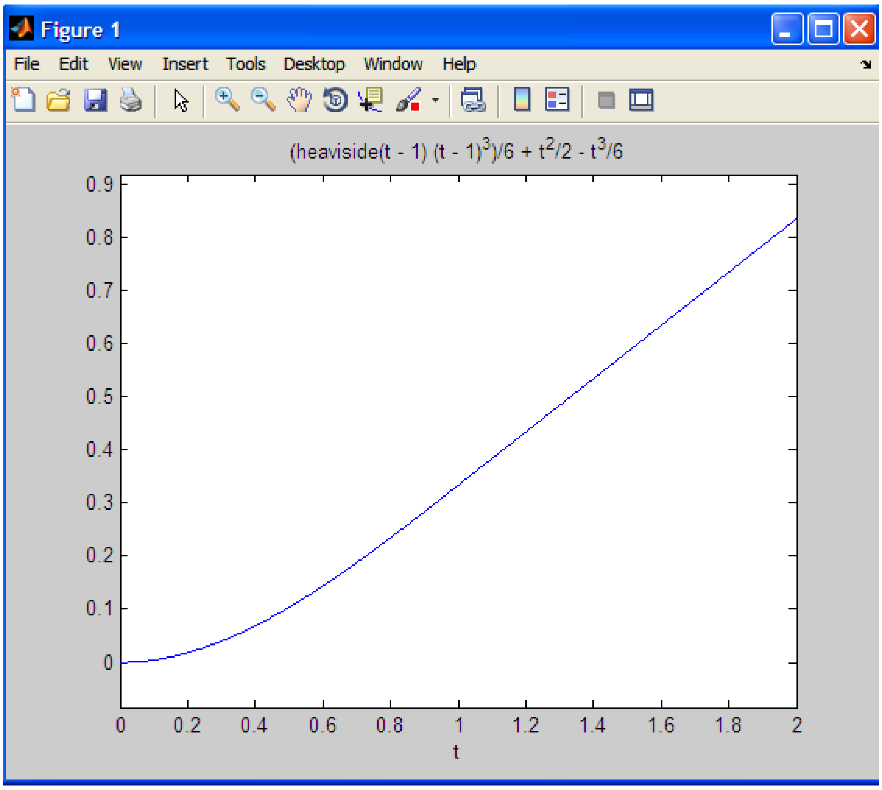
% y despejando, tenemos B=-1 y A=1. Introducimos en la solución soluA_B estos valores

>> solucion=subs(subs(soluA_B,B,-1),A,1) *% es la solución buscada*

solucion =

(heaviside(t - 1)*(t - 1)^3)/6 + t^2/2 - t^3/6

>> ezplot(solucion,[0,2])



% Resolución con dsolve (en la versión 2011 de Matlab) y comparación de soluciones

>> u=dsolve('D4y=dirac(t-1)','y(0)=0','Dy(0)=0','D2y(2)=0','D3y(2)=0')

u=

t/2 + ((heaviside(t - 1) - 1)*(t - 1)^3)/6 - 1/6

>> ezplot(u,[0,2]) *% nos da la misma gráfica de arriba*

>> u-solucion *% comprobamos que, en efecto, u y solución son la misma función*

ans =

t/2 + ((heaviside(t - 1) - 1)*(t - 1)^3)/6 - (heaviside(t - 1)*(t - 1)^3)/6 - t^2/2 - t^3/6 - 1/6

>> simple(ans)

ans =

0

*% **Observación:** A partir de la versión 2007 de Matlab, dsolve puede resolver ED en que intervienen funciones de Heaviside o delta de Dira; con versiones anteriores dsolve puede dar problemas, y dependiendo de la versión, puede ser necesario entender o interpretar lo que nos da para comprobar que la solución es la misma que la que se obtiene usando la transformada de Laplace. Para los problemas planteados, la transformada de Laplace no da problemas en versiones previas (desde Matlab 5.3). Abajo, ejemplos con dsolve y Matlab 6.5*

>> dsolve('D4y=dirac(t-pi)', 'y(0)=0','Dy(0)=0','D2y(2)=0','D3y(2)=0')

Warning: Explicit solution could not be found.
> In C:\MATLAB6p5\toolbox\symbolic\dsolve.m at line 326

ans =

[empty sym]

>> dsolve('D2y+2*Dy+2*y=dirac(t-pi)', 'y(0)=0','Dy(0)=0')

ans =

int(cos(t)*dirac(t-pi)*exp(t),t)*exp(-t)*sin(t)-int(sin(t)*dirac(t-pi)*exp(t),t)*exp(-t)*cos(t)-int(cos(a)*dirac(a-pi)*cosh(a)+cos(a)*dirac(a-pi)*sinh(a),a = `` .. 0)*exp(-t)*sin(t)+int(sin(a)*dirac(a-pi)*cosh(a)+sin(a)*dirac(a-pi)*sinh(a),a = `` .. 0)*exp(-t)*cos(t)

```
>> dsolve('D2y+y=heaviside(t)+heaviside(t-1)', 'y(0)=0','Dy(0)=0')

ans =

int(-sin(t)*(heaviside(t)+heaviside(t-1)),t)*cos(t)+int(cos(t)*(heaviside(t)+heaviside(t-1)),t)*sin(t)+int(sin(a)*(heaviside(a)+heaviside(a-1)),a = `` .. 0)*cos(t)-int(cos(a)*(heaviside(a)+heaviside(a-1)),a = `` .. 0)*sin(t)
```

% vemos que no resuelve los problemas planteados o lo deja en términos integrales. Estas mismas instrucciones devuelven la solución con la versión 2008 de Matlab.

*% **Sistema diferencial lineal**. **Ejercicio 1.7**: y+z'=exp(-t), 3y+y'=z-3z', y(0)=1,z(0)=1.*

% Aplicamos la transformada de Laplace a la resolución de un problema de Cauchy para un sistema diferencial, de primer orden, con incógnitas y(t) y z(t). Se llega a un sistema algebraico de dos ecuaciones con dos incógnitas.
lz almacena la transformada de laplace de z(t).
ly almacena la transformada de laplace de y(t).

```
>> syms s t ly lz

>> [LY,LZ]=solve(ly+s*lz-1-laplace(exp(-t)), 3*ly+s*ly-1-lz+3*(s*lz-1),ly,lz)

LY =

(s^2+2-s)/(1+s+s^2+s^3)


LZ =

(s^2+s+2)/(1+s+s^2+s^3)


>> ilaplace(LY) % nos da la incógnita y(t)

ans =

2*exp(-t)-cos(t)
```

>> ilaplace(LZ) *% nos da la incógnita z(t)*

ans =

exp(-t)+sin(t)
% demostramos que esta solución coincide con la solución explícita que nos da dsolve

>> [Y,Z]=dsolve('y+Dz=exp(-t)','3*y+Dy=z-3*Dz','y(0)=1','z(0)=1')

Y =

2/exp(t) - cos(t)

Z =

1/exp(t) + sin(t)

% Dibujamos la solución y(t), z(t) en [0,10]

>> ezplot(Y,[0,10])

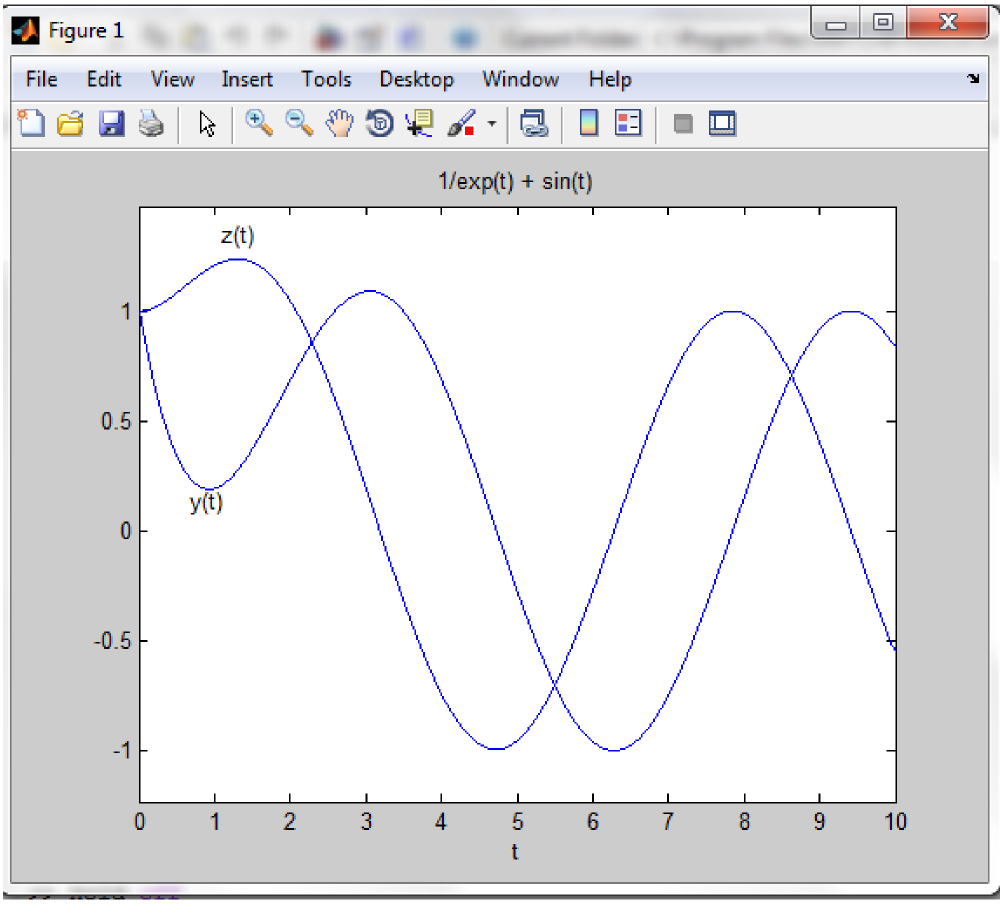
>> gtext y(t)

>> hold on

>> ezplot(Z,[0,10])

>> gtext z(t)

>> hold off



% Nota: dsolve, para este sistema concreto, se puede utilizar con la versión 2011 de Matlab. Con versiones anteriores de Matlab (2008 o anteriores) hay que despejar y y z para resolver con dsolve. En dichas versiones la resolución con transformada de Laplace no cambia.

% Finalmente observamos que la transformada de Fourier se utiliza de forma análoga a la de Laplace. Las variables simbólicas a utilizar por defecto son x y w como se ve con

>> help fourier *% con la versión 2008 de Matlab*

% el mismo resultado, nos devuelve help sym/fourier en la versión 2011 de Matlab

>> help sym/fourier

```
--- help for sym/fourier ---

FOURIER Fourier integral transform.
  F = FOURIER(f) is the Fourier transform of the sym scalar f
  with default independent variable x.  The default return is
  a function of w.
  If f = f(w), then FOURIER returns a function of t:  F = F(t).
  By definition, F(w) = int(f(x)*exp(-i*w*x),x,-inf,inf), where
  the integration above proceeds with respect to x (the symbolic
  variable in f as determined by FINDSYM).

  F = FOURIER(f,v) makes F a function of the sym v instead of
     the default w:
  FOURIER(f,v) <=> F(v) = int(f(x)*exp(-i*v*x),x,-inf,inf).

  FOURIER(f,u,v) makes f a function of u instead of the
     default x. The integration is then with respect to u.
  FOURIER(f,u,v) <=> F(v) = int(f(u)*exp(-i*v*u),u,-inf,inf).

  Examples:
   syms t v w x
   fourier(1/t)   returns   i*pi*(Heaviside(-w)-Heaviside(w))
   fourier(exp(-x^2),x,t)   returns   pi^(1/2)*exp(-1/4*t^2)
   fourier(exp(-t)*sym('Heaviside(t)'),v)   returns   1/(1+i*v)
   fourier(diff(sym('F(x)')),x,w)   returns   i*w*fourier(F(x),x,w)

  See also sym/ifourier, sym/laplace, sym/ztrans.
```

% Las transformadas de Laplace y de Fourier se pueden utilizar para resolver ecuaciones en derivadas parciales en dominios no acotados (por ejemplo, cuadrantes o semiplanos): ver ejercicios de la hoja 7, relativos a la difusión del calor y las vibraciones de vigas, y otros, en el capítulo 6 del libro de apuntes