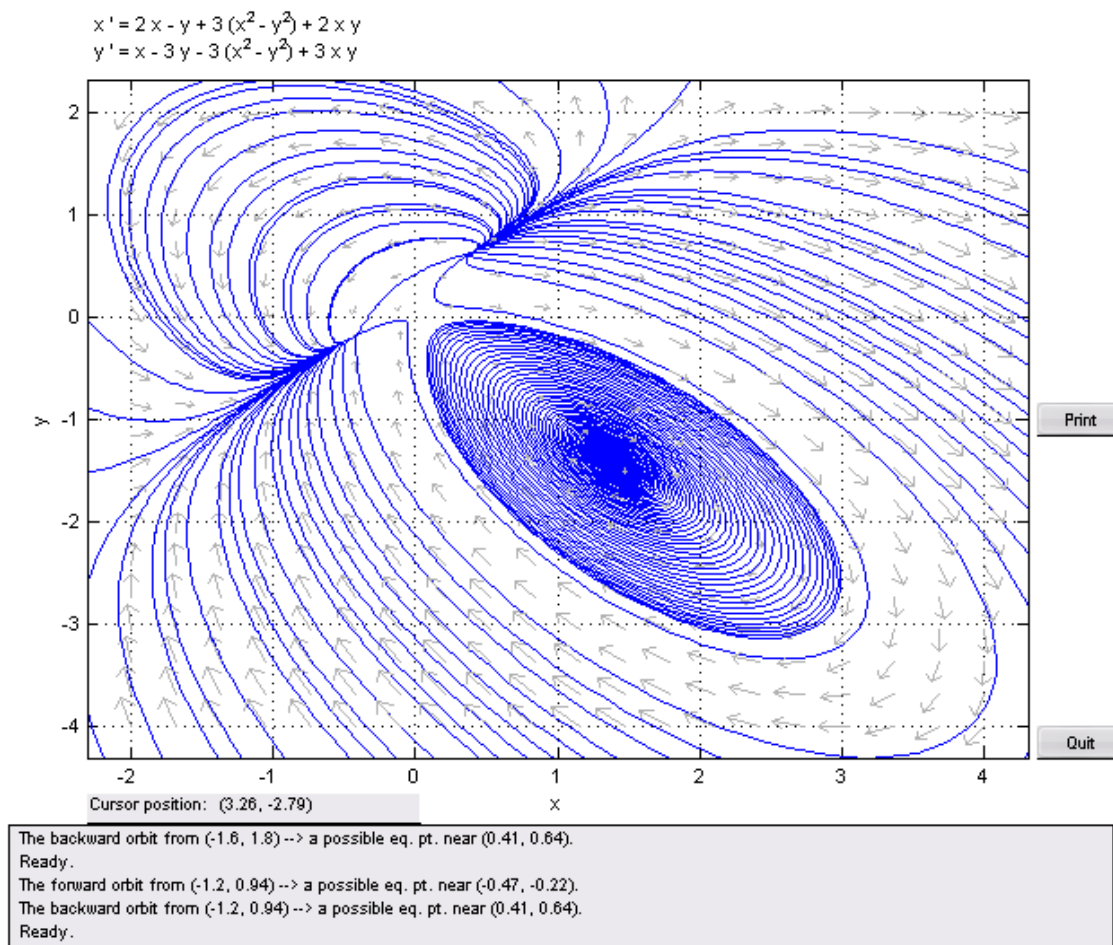


## Contents

1	introduccion	2
2	Que es matlab	2
3	aplicaciones de matlab	3
3.1	transformada de fourier . . . . .	3
3.2	sonido . . . . .	4
3.3	serie de fourier . . . . .	5
3.4	interpolacion . . . . .	5
3.5	transformada de laplace . . . . .	6
4	Ecuaciones diferenciales ordinarias	8
4.1	introduccion mecanica clasica . .	9



## 1 introduccion

resolver ecuaciones diferenciales ordinarias usando el ordenador y matlab. es un material de apoyo para el texto ecuaciones diferenciales ordinarias. el lector podra simular y correr la solucion numerica de varios problemas introductorios dentro de la fisica y adquirir el conocimiento y las herramientas para resolver problemas mas complejos. la primera parte del texto ofrece una introduccion a la interfaz de MATLAB y hace referencia a la sintaxis. la segunda parte habla sobre el toolbox ODE ( ordinary differential equation ). Agradesco la colaboracion de los lectores que sus comentarios ayuden a mejorar el texto para futuras ediciones.

1. command window

2. workspace window

falta poco para empezar tu primera corrida en matlab. todas las funciones, declaracion de constantes se teclea en el command window. mientras todos los resultados y variables son almacenadas en el workspace. Asi pues teclea el siguiente codigo en command window y observa el workspace almacena tu informacion.

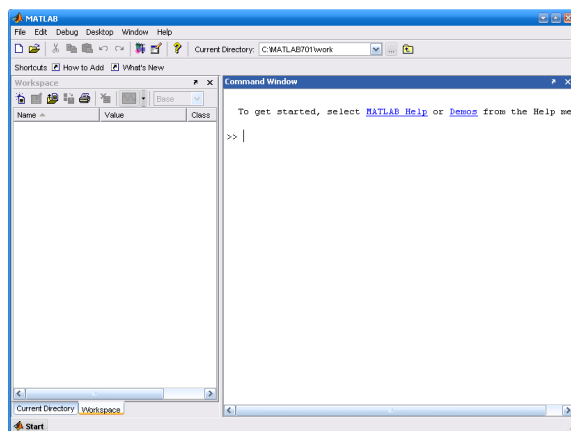
```
x = linspace(0,5);
y = sin(2*pi*x);
plot(y)
```

nota : el operador punto y coma ; se escribe para indicar el final de linea.

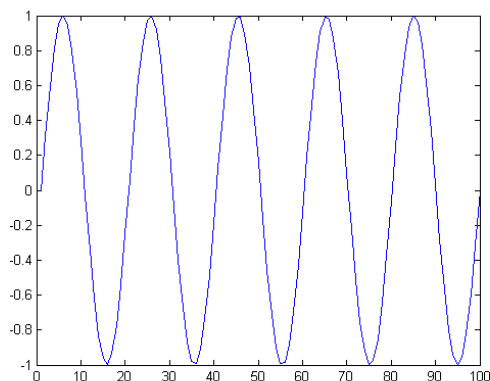
## 2 Que es matlab

Matlab es un programa para resolver problemas numericos su poder radica en el manejo de matrices en forma eficiente tambien incluye su propio compilador lo cual permite extender su uso permitiendo al usuario crear sus propios commands, clases y funciones. Es compatible con uno de los mas usados lenguajes de programacion C y Fortran.

la interfaz es amigable al usuario



hagamos una pausa y observa la pantalla principal de matlab que consta de dos importantes ventanas.



yay tu primera grafica en matlab la famosa funcion sin(x) como te daras cuenta matlab contiene un gran repertorio para graficacion. para mencionar algunos se encuentra graficacion de funciones parametricas, 2D, 3D, imagenes, videos, histogramas, campos vectoriales, etc. hasta este momento haz dominado el commando "plot(datos)" graficaremos funciones parametricas en matlab un ejemplo clasico es :

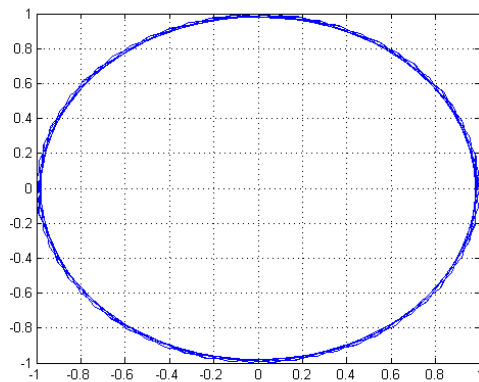
$$f(t) = \begin{cases} \cos(t) & 0 < t < 2\pi \\ \sin(t) & 0 < t < 2\pi \end{cases}$$

ahora teclea el siguiente codigo. si tienes muchas variables en el workspace puedes bor-

ralas tecleando `clear` en el command window

```
t = linspace(0,2*pi);
y = sin(2*pi*t);
x = cos(2*pi*t);
plot(x,y)
```

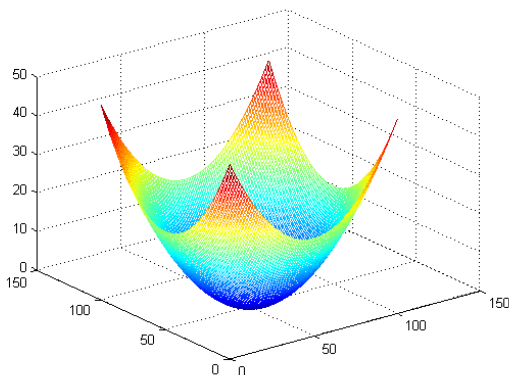
la grafica parametrica se muestra a continuacion



la grafica de un circulo representado por funciones parametricas. esto puede servirnos para graficar la trayectoria de un proyectil.

Ahora un ejemplo de graficas en 3D

```
[x,y]=meshgrid(-5:.01:5);
z= (x.^2+y.^2);
mesh(z)
```



El comando `mesh(datos)` grafica en forma de un mechado una matriz de datos. es muy importante recordad el uso del operador `.` matlab

trabaja en standar usando algebra de matrices, al usar `.` matlab reconoce es una multiplicacion normal y no entre dos matrices

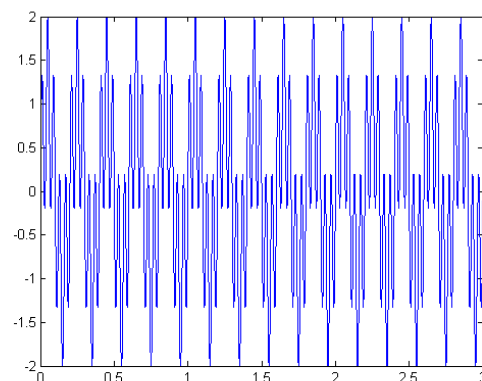
## 3 aplicaciones de matlab

podria seguir hablando otro tipos de graficacion pero tardaria un monton de tiempo. a continuacion muestro trabajos e realizado con matlab en la universidad y el codigo

### 3.1 transformada de fourier

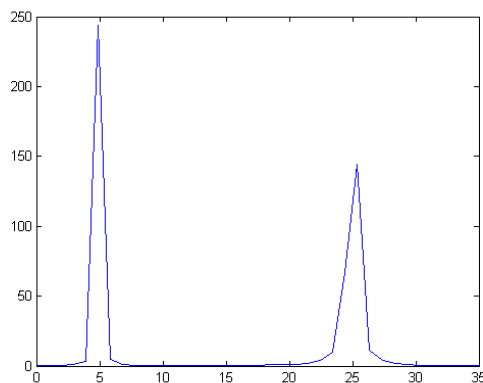
bueno este es algo que todos utilizamos en algun momento en la universidad. no mencionare matematicamente la transformada de fourier pero si su uso. el poder de fourier radica en interpretar cualquier funcion en una serie de  $\sin(x)$  y  $\cos(x)$ . estas dos funciones son muy importantes por que se encuentra en la naturaleza. la propagacion del sonido, corriente alterna, vibraciones y resonancia. incluso es solucion a la ecuacion de onda y ecuacion de flujo de calor etc. es mas facil analizar fenomenos en otro dominio (el dominio de fourier).

```
t = 0:1/1000:3;
y = sin(2*pi*5*t);
y1 = sin(2*pi*25*t);
plot(t,y+y1)
```



la senal anterior es la suma de dos ondas, la primera con una frecuencia de 5 Hz y la segunda de 25 Hz . pero espera como puedo saber esa informacion de antemano ?. Si observas el codigo puedes indentificar las frecuencias y que sucede si no tienes el codigo a la mano ? si una persona observa la grafica anterior y preguntas : cuantas senales y de que frecuencias se compone? bueno eso esta dificil en el dominio de tiempo pero en el dominio de fourier es relativamente facil.

```
FFT=fft(y,1024);
FFT =
FFT.*conj(FFT)/1024;
f = 1000*(0:256)/1024;
plot(f,FFT(1:257));
```

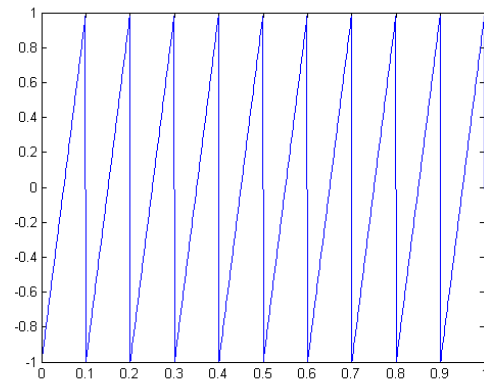


los dos picos muestran el valor de frecuencia en 5 y 25 Hz. aqui van unos tips al momento realisar fourier

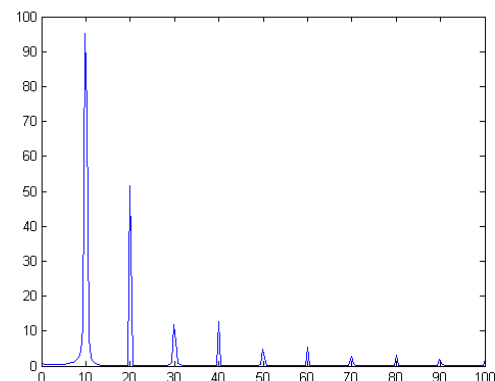
1. la funcion `fft(dato,n)` en matlab se usa para la transformada de fourier
2. siempre usa la funcion `fft` con multiplos de dos por ejemplo : 2, 4, 16, 32, 64, 128, 256, 512, 1024, etc
3. la frecuencia de adquisicion de datos debe ser por lo menos el doble de la frecuencia mas alta en la senal
4. agregar ceros a la senal no afecta a la transformada de fourier pero si da mas resolucion a la grafica

pero que pasa si agrego una senal que no sea de la forma  $\sin(\omega t) + \sin(\omega pt)$

```
t = 0:1/1000:8*pi;
y = sawtooth(2*pi*10*t);
plot(t,y)
```



apliquemos la transformada de fourier a la onda en forma de sierra con  $f = 10$  Hz



Fijate que la frecuencia fundamental es el pico de mayor amplitud que es 10 Hz claro tiene sentido si nuestra funcion tiene una  $f = 10$  Hz , los otros picos representan armonicos. pero que son los armonicos ?

### 3.2 sonido

la senal de sierra puede ser la representacion del sonido produce un violin. si tocamos otros instrumentos musicales la forma de la onda

seria diferente. sin embargo el oido escucharia la frecuencia fundamental de 10 hz mas los armonicos. como los dos instrumentos producen la misma frecuencia fundamental para la misma nota deberan escucharse casi igual y se dice tiene un "pitch" identico. como la amplitud de los armonicos es diferente el sonido no es identico y se dice tienen diferente "timbre". a pesar que el piano solo puede reproducir 20% de las frecuencia el ser humano puede escuchar (4Khz a 20 Khz) puede producir mas del 70% de la informacion los humanos pueden persivir ( 7 de 10 octavas )

en la vida real no podemos usar la transformada de fourier eficientemente ya que en su dominio aparecen un monton de armonicos y frecuencia. en ese caso podemos implementar filtros digitales y analogos.

### 3.3 serie de fourier

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos\left(\frac{n\pi x}{L}\right) + b_n \sin\left(\frac{n\pi x}{L}\right) \quad (1)$$

$$a_n = \int_{-L}^L f(x) \cos\left(\frac{n\pi x}{L}\right) dx \quad (2)$$

$$b_n = \int_{-L}^L f(x) \sin\left(\frac{n\pi x}{L}\right) dx \quad (3)$$

esta es la serie de fourier. resolvamos la solucion para la funcion de tipo sierra.

$$b_n = \int_0^{2L} x \sin\left[\frac{n\pi x}{L}\right] dx \quad (4)$$

$$b_n = \frac{L^2 \sin(2\pi n)}{\pi^2 n^2} - \frac{2L^2 \cos(2\pi n)}{\pi n} \quad (5)$$

sustituyendo el valor de bn en la ecuacion (1)

$$\frac{A}{L} \sum_{n=1}^{\infty} b_n \sin\left[\frac{n\pi x}{L}\right] \quad (6)$$

como la funcion  $f(x) = x$  es impar el valor de los coeficientes "an" es igual a cero. "L" representa el periodo de la onda y "A" es la amplitud. ejemplo: Genera una funcion tipo

sierra usando la ecuacion (6). Amplitud = 5 y una frecuencia de 2 Hz y desarrolla los primeros 10 terminos de la serie.

fijate que en matlab no podemos usar esta tecnica por las siguientes razones

- no tenemos un espacio ilimitado de memoria para extender la serie hasta infinito
- las variables son discretas
- no conocemos la funcion  $f(x)$  represente nuestros datos

En la vida real solo trabajamos con datos numericos. en ocasiones es imposible generar una interpolacion y definir una funcion  $f(x)$ . eso es la realidad.

### 3.4 interpolacion

la idea de este metodo es generar un polinomio de grado "n" para unir puntos en un patron de datos. por ventaja matlab hace esto automaticamente. teclea "cftool" en el command window

- Para generar un polinomio de grado "n" necesitas por lo menos "n" puntos en el plano
- cftool es la interfaz visual facilita matlab para interpolacion. exponencial, fourier, polinomio, lineal , etc

para iniciar la interpolacion debes exportar datos del workspace. los siguientes datos son aleatorios y se usaran para definir un ejemplo :

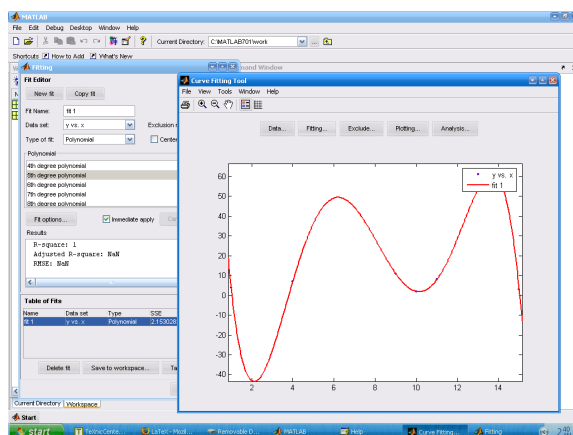
$$(1,4) (4,7) (9,11) (10,2) (11,8) (15,3)$$

el primer digito corresponder a "x" y el segundo digito a "y" (x,y)

insertamos los datos en el command window.  
un vector para los puntos en "x" y un vector  
para los puntos en "y"

```
x = [ 1 4 9 10 11 15 ];
y = [4 7 11 2 8 3 ];
cftool
```

graficamos. matlab cuenta con muchas  
formas de interpolar es este caso usaremos un  
polinomio de grado "5"



Linear model Poly5:

$$f(x) = p1*x^5 + p2*x^4 + p3*x^3 + p4*x^2 + p5*x + p6$$

Coefficients:

```
p1 = -0.02875 p2 = 1.155 p3 =
-16.76 p4 = 105.4 p5 = -262.4 p6
= 176.6
```

Goodness of fit: SSE: 2.153e-022 R-square:  
1 Adjusted R-square: NaN RMSE: NaN  
ahora sabes interpolar datos. continuamos con  
los usos y herramientas que ofrece matlab

### 3.5 transformada de laplace

usada la mayor parte del tiempo en resolver  
circuitos electricos. por que la usamos y su  
funcion ? . laplace permite extendernos del  
dominio del tiempo al domonio complejo "s".  
Otra manera definimos a la serie de fourier es

en su forma compleja por

$$X(w) = \int_{-\infty}^{\infty} x(t)e^{-iwt} dt \quad (7)$$

la ecuacion (7) puede expanderse en la transfor-  
mada de laplace, multiplicando la senal en el  
tiempo por un termino exponencial

$$X(\sigma, w) = \int_{-\infty}^{\infty} [x(t)e^{-\sigma t}]e^{-iwt} dt \quad (8)$$

$$X(\sigma, w) = \int_{-\infty}^{\infty} x(t)e^{-(\sigma+iw)t} dt \quad (9)$$

para localizar valores en el plano complejo se  
puede representar por letra  $s = \sigma + iw$  y la  
ecuacion (9) se puede escribir en una forma mas  
compacta

$$X(s) = \int_{-\infty}^{\infty} x(t)e^{-st} dt \quad (10)$$

tambien existe la transformada inversa de  
laplace :

$$L^{-1}[F(s)] = \frac{1}{2\pi i} \int_{\sigma-i\infty}^{\sigma+i\infty} F(s)e^{st} ds \quad (11)$$

la ecuacion (11) es la forma formal de la trans-  
formada de laplace pero no la usaremos. ya  
existe unas tablas que facilitan ese trabajo.y lo  
mejor es que matlab puede sacar la transfor-  
mada de laplace. a continuacion se menciona  
un ejemplo.

la tranformada de laplace de la funcion

$$f(t) = te^{-2t}$$

```
syms t s
f=t*exp(-2*t)
laplace(f)
```

y a la salida obtenmos el resultado

$$F(s) = 1/(s^3 - s^2 + s - 1)$$

matlab contiene comandos muy simples.  
estos son parte de un paquete "symbolic  
toolbox" para realizar los siguiente ejemplos  
necesitas tener instalado el paquete

encuentra la transformada inversa de laplace

$$F(s) = 2t^2 e^{-t} \cos(4t) + \sqrt{t}$$

```
syms t s
F =
2*(t.^2)*exp(-t)
.*cos(4*t)+t.^(1/2)
R = laplace(F)
pretty(R)
```

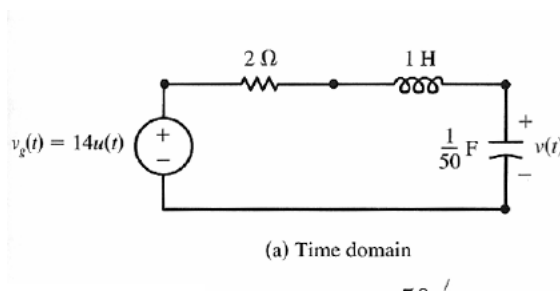
la salida del codigo anterior es

$$\frac{4(s+1)(s^2+2s-47)}{(s^2+2s+17)^3} + \frac{\sqrt{\pi}}{512s^{3/2}} \quad (12)$$

resuelve la siguiente ecuacion diferencial usando laplace con las siguientes condiciones  $y(0)=2$   $y'(0)=2$   $y''(0)=0$

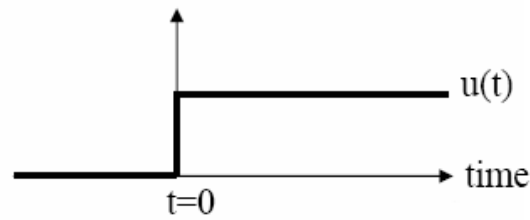
$$y''' + 6y'' + 15y' + 50y = te^{-5t} + t^3 \quad (13)$$

el poder de fourier radica su uso en electronica y disenio de filtros digitales. los dos tipos IIR ( infinite impulse response) y FIR ( finite impulse response ). el filtro FIR se aplica usando un proceso llamado convolucion ( este termino se asocia en este caso a proceso digital de senales) mientras el IIR es aplicado mediante laplace y la transformada "z" eso es de esperarse. recuerden un decaimiento exponencial de la forma  $e^{-t}$  tiende a cero en forma de asintota.



resolvamos el circuito RCL anterior con la ayuda de laplace y encontremos su step response ( funcion de impulso ) y grafiquemos su resultado en matlab

$$u(t) = \begin{cases} 0 & -\infty < t < 0 \\ 1 & 0 \leq t < \infty \end{cases}$$



la grafica anterior muestra una senal de entrada de corriente directa

resolvamos usando la ley conservacion de la energia y dado que las condiciones iniciales del sistemas son cero

$$\begin{aligned} V(s) &= \frac{50/s}{50/s + s + 2} V_s(s) = \frac{50}{s^2 + 2s + 50} \left( \frac{14}{s} \right) \\ V(s) &= \frac{700}{s(s^2 + 2s + 50)} = \frac{K_0}{s} + F(s) \rightarrow K_0 = 14 \\ V(s) &= \frac{14}{s} - \frac{14s + 28}{s^2 + 2s + 50} = \frac{14}{s} - \frac{14(s+1)}{(s+1)^2 + 7^2} - \frac{(2)(7)}{(s+1)^2 + 7^2} \end{aligned}$$

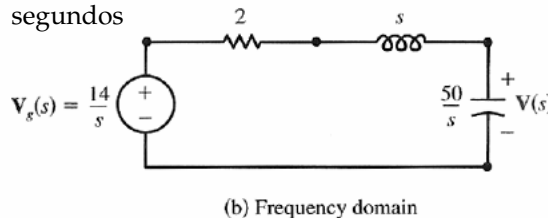
ahora el resultado final es :

$$f(s) = \frac{700}{s(s^2 + 2s + 50)} \quad (14)$$

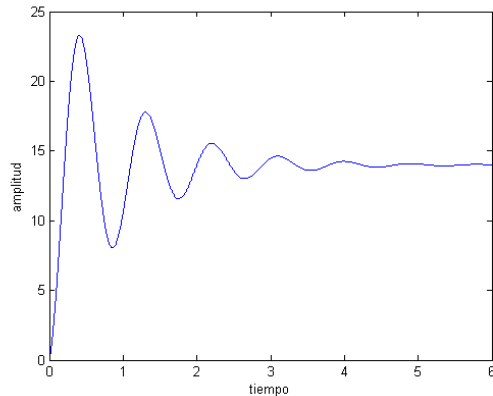
aplicando la transformada inversa de laplace a  $f(s)$  encontramos la step function en el dominio del tiempo

$$f(t) = 14 - e^{-t}(14\cos(7t) - 2\sin(7t)) \quad (15)$$

ahora tenemos que graficar  $f(t)$  y observar como el circuito se comporta en los primeros 6 segundos



observamos el decaimiento exponencial. la forma de un oscilador



aunque la forma mas compacta representar la step fuction del circuito RCL anterior es por medio de la ecuacion (14) el uso de variables complejas dificulta y hace casi imposible generar una grafica la solucion es separar las parte real e imaginaria de la variable "s". los siguientes tips son utiles para saber en que se puede aplicar laplace

1. la transformada de laplace solo es util si se analizan funciones consisten de senos y exponenciales
2. filtros analogos se disenian usando la transformada de laplace y recursive filtros mediante la transformada "z"

## 4 Ecuaciones diferenciales ordinarias

Ahora entremos a metodos para resolver dos tipos de ecuaciones diferenciales ordinarias :

1. ecuaciones diferenciales de segundo orden con coeficientes constantes
2. sistema de nxn ecuaciones diferenciales de primer orden

Matlab cuenta con un paquete llamada "ODE toolbox" en su siglas en ingles (ordinary differential equation) tambien cuenta con un paquete

PDE ( partial differential equation ) en este texto solo se mencionare el ODE y en futuras ediciones se agregara el PDE. Para poder resolver una ecuacion diferencial en matlab primero debe crearse un M-file en otras palabras una funcion donde se introduce la ecuacion diferencial a resolver y debera ser invocada en el command window insertando condiciones iniciales, valores de tiempo , etc. La ecuaciones de un oscilador con friccion se define por

$$my'' + by' + ky = 0 \quad (16)$$

la ecuacion anterior es muy facil de resolver a mano pero sera un buen ejemplo para usar el ODE de matlab

```
function dydt =
oscilador(t,y)
dydt = [y(2);
-0.1*y(2)-1*y(1)];
```

haciendo la sustitucion  $y' = y(2)$   $y = y(1)$  una ecuacion diferencial de segundo orden puede transformarse a una ecuacion de 1er orden. invoquemos la m-file anterior tecleando en el command window lo siguiente

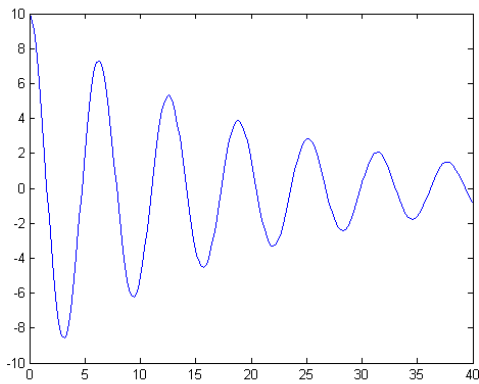
```
[t,y] =
ode45(@oscilador,[0
40],[5; 5]);
plot(t,y(:,1)+y(:,2));
```

matlab regresa una matriz "y" de 2 columnas. cada columna representa 1 de las 2 soluciones de la ecuaciones diferencial. recuerda la solucion de estas ecuaciones diferenciales es una combinacion lineal de la forma:

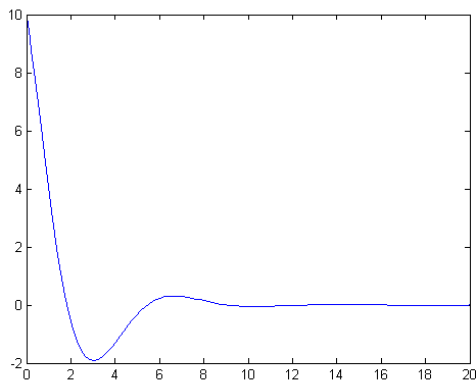
$$y(t) = C_1 Y_1 + C_2 Y_2 \quad (17)$$

la grafica del codigo anterior con  $m=1$  ,  $b=.1$  ,  $k=1$  y  $t$  de 0 a 40 segundos se muestra a continuacion

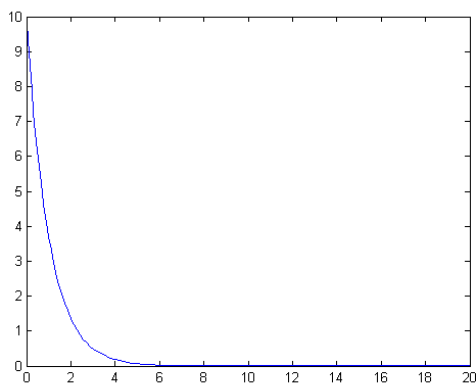




si cambiamos el valor de "b", "m" y "k" encontramos diferentes soluciones. ahora editemos el m-file y sea el coeficiente de friccion  $b = 1$



al aumentar la friccion el decaimiento exponencial es mayor y la oscilacion tiende a cero mas rapido. Si hacemos aun mayor el valor de "b" no se observan oscilaciones en el sistema, sea  $b = 2$  en el siguiente ejemplo :



## 4.1 introduccion mecanica clasica

la mecanica clasica ofrece un sin numero de problemas de sistemas de ecuaciones diferenciales describiendo movimiento. El orden de la ecuacion diferencial correspondiente al movimiento del sistema puede ser enorme. Aqui se tendra en cuenta lo siguiente para ayudar a reducir lo mas posible el numero de ecuaciones

- Usar la energia cinetica y potencial para resolver el movimiento
- No necesitas resolver para aceleraciones
- Necesitas resolver velocidades inerciales

Le ecuacion de Lagrange para sistemas conservativos

$$\frac{\partial L}{\partial \dot{q}} \frac{\partial}{\partial t} = \frac{\partial L}{\partial q} \quad (18)$$

en donde  $q$  es la coordenada generalizada y los grados de libertad del sistema y " $L$ " se define como el lagrangiano y es la resta de la energia cinetica y potencial :

$$L(q, \dot{q}, t) = T(\dot{q}) - U(q) \quad (19)$$

El metodo de newton requiere que encuentres todas las fuerzas del sistemas en 3 direcciones y reducir el problema. el metodo de lagrange nos permite reducir el problema resolviendo el numero de ecuaciones y usando algebra elemental.

- el lagrangiano requiere de saber posicion y velocidad del sistema

Otra manera de interpretar el movimiento es mediante la mecanica de hamilton la cual usa el momento y posicion para poder resolver el sistema . La energia cinetica debe estar en funcion del momento y no de la velocidad

$$T(p) = \frac{p^2}{2m} \quad (20)$$

recuerda que esta es la energia cinetica en terminos de fisica clasica y no relativista El hamiltoniano es entonces la transformacion del la-

grangiano y se define por la transformacion de legendre :

$$H(q_j, p_j, t) = \sum_j \dot{q}_j p_j - L(q, \dot{q}, t) \quad (21)$$

para cada velocidad hay un momento generalizado

$$p_j = \frac{\partial L}{\partial \dot{q}_j} \quad (22)$$

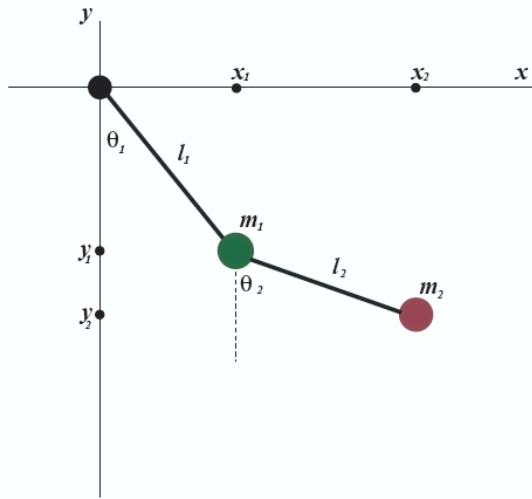
asi obtenemos las ecuaciones definen al hamiltoniano

$$\frac{\partial H}{\partial p_j} = \dot{q}_j \quad (23)$$

$$\frac{\partial H}{\partial q_j} = -\dot{p}_j \quad (24)$$

$$\frac{\partial H}{\partial t} = -\frac{\partial L}{\partial t} \quad (25)$$

apliquemos las anteriores ecuaciones al pendulo doble



encontremos el potencial y energia cinetica del sistema

$$U(q) = m_1 g y_1 + m_2 g y_2 \quad (26)$$

$$T(\dot{q}) = \frac{m_1(\dot{y}_1^2 + \dot{x}_1^2)}{2} + \frac{m_2(\dot{y}_2^2 + \dot{x}_2^2)}{2} \quad (27)$$

haciendo la sustitucion

$$x_1 = l_1 \sin(\theta_1) \quad (28)$$

$$y_1 = -l_1 \cos(\theta_1) \quad (29)$$

$$x_2 = x_1 + l_2 \sin(\theta_2) \quad (30)$$

$$y_2 = y_1 - l_1 \cos(\theta_2) \quad (31)$$

reducimos terminos para el potencial

$$U(q) = \quad (32)$$

$$-(m_1 + m_2)gl_1 \cos \theta_1 - m_2 gl_2 \cos \theta_2.$$

reducimos terminos para la energia cinetica

$$T(\dot{q}) = \quad (33)$$

$$\frac{1}{2}m_1 l_1^2 \dot{\theta}_1^2 + \frac{1}{2}m_2 [l_1^2 \dot{\theta}_1^2 + l_2^2 \dot{\theta}_2^2 + 2l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \cos(\theta_1 - \theta_2)]$$

usando ecuacion (19)

$$L(q, \dot{q}, t) \quad (34)$$

$$L = \frac{1}{2}(m_1 + m_2)l_1^2 \dot{\theta}_1^2 + \frac{1}{2}m_2 l_2^2 \dot{\theta}_2^2 + m_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \cos(\theta_1 - \theta_2)$$

$$+ (m_1 + m_2)gl_1 \cos \theta_1 + m_2 gl_2 \cos \theta_2$$

usando la ecuacion (18)y resolviendo para el lagrangiano en (34)

$$\frac{\partial L}{\partial \theta_1} = m_1 l_1^2 \ddot{\theta}_1 + m_2 l_1^2 \ddot{\theta}_1 + m_2 l_1 l_2 \ddot{\theta}_2 \cos(\theta_1 - \theta_2)$$

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_1} \right) = (m_1 + m_2)l_1^2 \ddot{\theta}_1 + m_2 l_1 l_2 \ddot{\theta}_2 \cos(\theta_1 - \theta_2) - m_2 l_1 l_2 \dot{\theta}_2 \sin(\theta_1 - \theta_2)(\dot{\theta}_1 - \dot{\theta}_2)$$

$$\frac{\partial L}{\partial \theta_1} = l_1(m_1 + m_2)g \sin \theta_1 - m_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2),$$

resolviendo entonces para

$$\theta_1 = \quad (35)$$

$$(m_1 + m_2)l_1^2 \ddot{\theta}_1 + m_2 l_1 l_2 \ddot{\theta}_2 \cos(\theta_1 - \theta_2) + m_2 l_1 l_2 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) + l_1(m_1 + m_2)g \sin \theta_1 = 0$$

resolviendo ahora para la siguiente derivada parcial

$$C_1 \equiv \frac{p_1 p_2 \sin(\theta_1 - \theta_2)}{l_1 l_2 [m_1 + m_2 \sin^2(\theta_1 - \theta_2)]}$$

$$\frac{\partial L}{\partial \dot{\theta}_2} = m_2 l_2^2 \dot{\theta}_2 + m_2 l_1 l_2 \dot{\theta}_1 \cos(\theta_1 - \theta_2)$$

$$C_2 \equiv \frac{l_2^2 m_2 p_1^2 + l_1^2 (m_1 + m_2) p_2^2 - l_1 l_2 m_2 p_1 p_2 \cos(\theta_1 - \theta_2)}{2 l_1^2 l_2^2 [m_1 + m_2 \sin^2(\theta_1 - \theta_2)]^2} \sin[2(\theta_1 - \theta_2)]$$

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_2} \right) = m_2 l_2 \ddot{\theta}_2 + m_2 l_1 l_2 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - m_2 l_1 l_2 \dot{\theta}_1 \sin(\theta_1 - \theta_2) (\dot{\theta}_1 - \dot{\theta}_2)$$

$$\frac{\partial L}{\partial \theta_2} = m_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) - l_2 m_2 g \sin \theta_2,$$

despues de obtener el lagrangiano y resolver el algebra ya podemos usar el siguiente codigo y obtener la solucion numerica : recuerden la computadora solo es una maquina realiza un proceso varias veces no puede obtener y generar ecuaciones por si sola.

resolviendo para

$$\theta_1 = \quad (36)$$

$$m_2 l_2^2 \ddot{\theta}_2 + m_2 l_1 l_2 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - m_2 l_1 l_2 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) + l_2 m_2 g \sin \theta_2 = 0.$$

La ecuacion (35) y (36) son ecuaciones diferenciales de segundo orden. podemos usar la transformacion de legendre y obtener el hamiltoniano.asi obtener un sistema de 4 ecuaciones diferenciales de primer orden y resolver usando el ODE de matlab.

$$ecuaciones \quad (37)$$

$$\dot{\theta}_1 = \frac{\partial H}{\partial p_1} = \frac{l_2 p_1 - l_1 p_2 \cos \theta_1 - \theta_2}{l_1^2 l_2 [m_1 + m_2 \sin^2(\theta_1 - \theta_2)]}$$

$$\dot{\theta}_2 = \frac{\partial H}{\partial p_2} = \frac{l_1 (m_1 + m_2) p_2 - l_2 m_2 p_1 \cos(\theta_1 - \theta_2)}{l_1 l_2^2 m_2 [m_1 + m_2 \sin^2(\theta_1 - \theta_2)]}$$

$$\dot{p}_1 = -\frac{\partial H}{\partial \theta_1} = -(m_1 + m_2) g l_1 \sin \theta_1 - C_1 + C_2$$

$$\dot{p}_2 = -\frac{\partial H}{\partial \theta_2} = -m_2 g l_2 \sin \theta_2 + C_1 - C_2$$

no dare todo el proceso paso a paso como obtener el hamiltoniano por que estoy cansado de escribir en latex y el algebra es tediosa nota : por falta de espacio se agrega C1 y C2 que tiene el siguiente valor

El siguiente código genera la matriz de 4 ecuaciones diferenciales de primer orden ver ecuación (37)

```

1 function xprime = doublependulum(t,x,flag,g,l1,l2,m1,m2)
2 C1 = (x(3).*x(4).*sin(x(1)-x(2)))./...
3     l1*l2*(m1+m2*(sin(x(1)-x(2))).^2);
4 C2 = ((l2^2*m2*(x(3)).^2+l1^2*(m1+m2)*...
5     (x(4)).^2-l1*l2*m2*x(3).*x(4).*cos(x(1)-x(2)))./...
6     /2*l1^2*l2^2*(m1+m2*sin((x(1)-x(2))).^2).^2)...
7     .*sin(2*(x(1)-x(2))));
8
9 xprime=zeros(4,1);
10 xprime(1)=(l2*x(3)-l1*x(4).*cos(x(1)-x(2)))./...
11     l1^2*l2*(m1+m2*(sin(x(1)-x(2))).^2);
12 xprime(2)=(l1*(m1+m2)*x(4)-l2*m2*x(3).*cos(x(1)-x(2)))./...
13     l1*l2^2*m2*(m1+m2*(sin(x(1)-x(2))).^2);
14 xprime(3) = -(m1+m2)*g*l1*sin(x(1))-C1+C2;
15 xprime(4) = -m2*g*l2*sin(x(2))+C1-C2;

```

Debemos salvar el código en formato m-file e invocar la función. El siguiente código se debe teclear en el command window sea  $m1 = 1$   $m2 = 1$   $l1 = 1$  y  $l2 = 1$  para el siguiente ejemplo (puedes variar las condiciones iniciales, masas y longitud del pendulo:)

#### Command Window

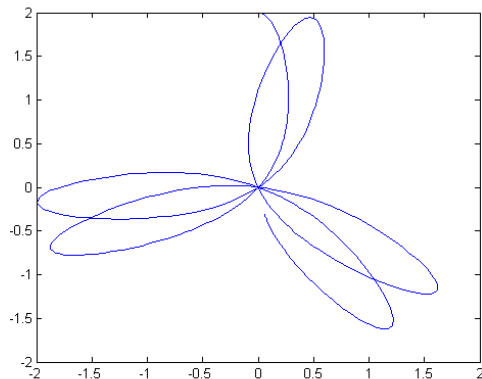
To get started, select [MATLAB Help](#) or [Demos](#) from the Help menu.

```

>> [t,x]=ode45('doublependulum',[0,0.7],[pi;pi;10;-8.898788999],[],9.8,1,1,1,1);
[t,x]=ode45('doublependulum',[0,1],[pi;pi;20;-50],[],9.8,1,1,1,1);
[t,x]=ode45('doublependulum',[0,25],[pi;pi;0;0],[],9.8,1,1,1,1);
x1=sin(x(:,1));
y1 = -cos(x(:,1));
x2 = x1+sin(x(:,2));
y2 = y1-cos(x(:,2));
plot(x2,y2)
axis([min(-2) max(2) ...
min(-2) max(2)])

```

invocando el primer ODE que se encuentra en el código anterior



## References

- [1] Buerger, David J., *TEX for scientists and engineers*, (New York: McGraw-Hill, 1990).

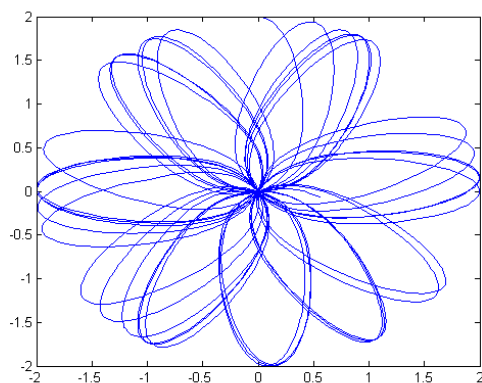
<http://www.wam.umd.edu/petersd/246/matlabode2.html>

<http://www.math.montana.edu/doc/dfieldtut.html>

<http://online.redwoods.cc.ca.us/instruct/darnold/>

<http://www.mathworks.com/>

invocando el segundo ODE que se encuentra en el código anterior



ESTE TEXTO FUE CREADO POR :  
OSCAR GUERRERO MIRAMONTES  
ING.FISICA  
MEXICO - UACJ  
E-MAIL:  
SLOWMUSICIAN@YAHOO.COM.MX

invocando el tercer ODE que se encuentra en el código anterior invocando el segundo ODE que se encuentra en el código anterior

