LABORATORIO MATLAB

Liz Ángel Núñez Torres

4 de mayo de 2024

Preparación de datos y funciones.

1. Definir la función a explorar.
2. Definir el conjunto de x a definir.
3. Utiliza MATLAB para definir funciones.

Definir funciones:

Hemos definido las siguientes funciones en nuestro editor:

- ”trigo ”donde calcularemos el seno y coseno de los valores que queramos analizar.

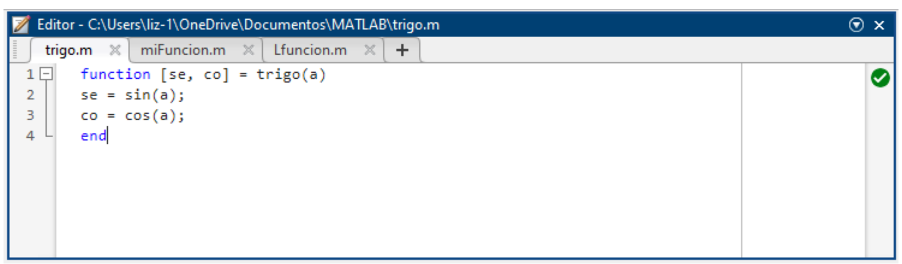


Figura 1: Función **trigo** Tomada de MATLAB

- ”miFuncion ”función donde tenemos un polinomio.

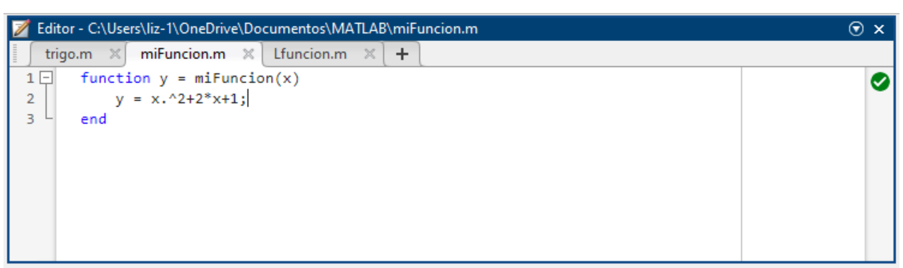


Figura 2: Función **miFuncion** Tomada de MATLAB

- "Lfuncion "función donde tenemos un nuevo polinomio.

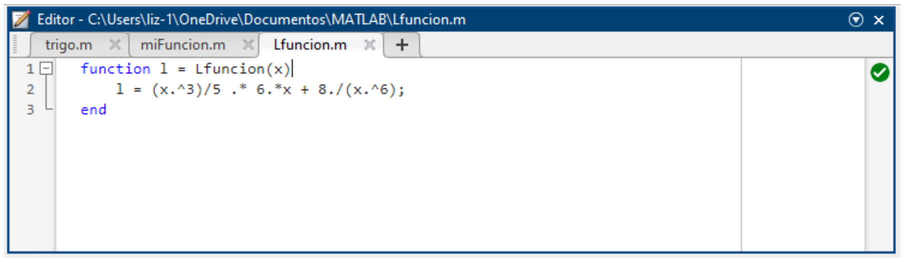
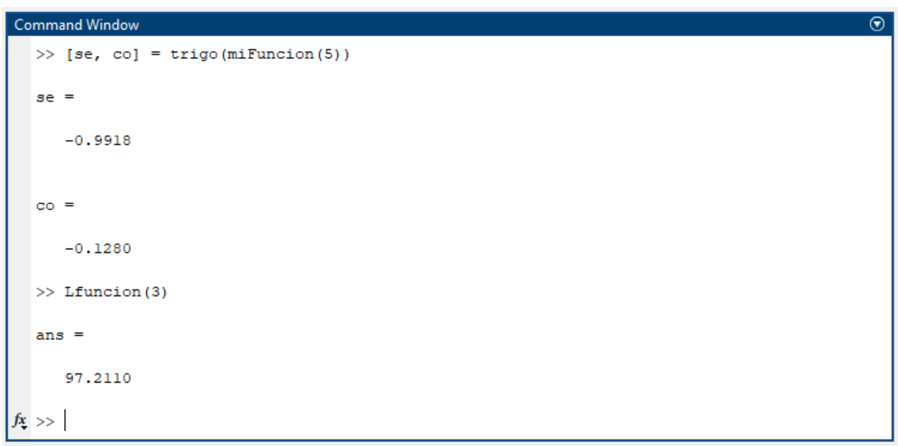


Figura 3: Función **Lfuncion** Tomada de MATLAB

Ahora que hemos definido nuestras funciones, vamos a evaluarlas con diferentes valores.



Como podemos apreciar, en la función *"miFuncion "* hemos evaluado el seno y coseno con la función *"trigo "* para valores de $x = 5$.

Después hemos evaluado en nuestra función *"Lfuncion "* con valores de $x = 3$.

Cálculos Básicos.

1. Utiliza las funciones definidas y las funciones apropiadas de Matlab para calcular derivadas y sus gráficas, la integral en un rango del dominio de terminado, el área de la región acotada por la gráfica seleccionada, el eje $y = 0$ y dos valores apropiados x_0,x_1 que acoten la región.

2. Utiliza las funciones de MATLAB, derivadas (diff) de diversos órdenes, para encontrar la segunda y tercera derivada de la función seleccionada.

3. Verifica los resultados con cálculos a mano o utilizando herramientas de cálculo simbólico si es posible.

- Vamos a desarrollar nuestro primer punto.

```
Command Window
>> syms x;
>> diff(miFuncion(x))

ans =

2*x + 1

>> miDerivada = ans

miDerivada =

2*x + 1

>> diff(Lfuncion(x))

ans =

(24*x^3)/5 - 48/x^7

>> Lderivada = ans

Lderivada =

(24*x^3)/5 - 48/x^7

>>
```

Hemos calculado las derivadas de nuestras dos funciones *"miFuncion "* y *"Lfuncion "* para a continuación definir estas derivadas.

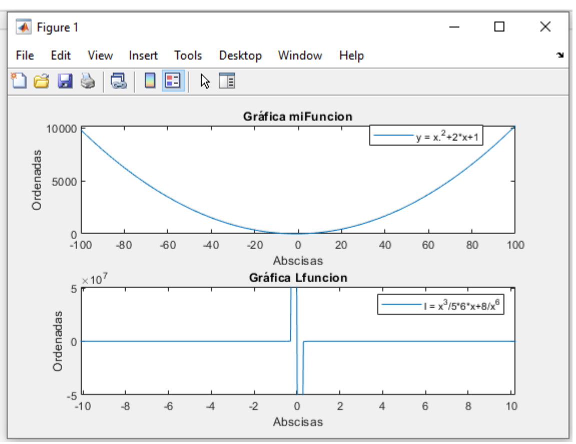
Función:	**Derivada:**
miFuncion	*miDerivada*
Lfuncion	*Lderivada*

Después de identificar nuestras derivadas como *"miDerivada "* y *"Lderivada "* vamos a transformarlas en funciones anónimas; de esta forma al llamar las mismas, MATLAB las podrá reconocer como númericas y no como variables simbólicas.

```
Command Window
>> miDerivada = matlabFunction(miDerivada)

miDerivada =

  function_handle with value:

    @(x)x.*2.0+2.0

>> Lfuncion = matlabFunction(Lderivada)

Lfuncion =

  function_handle with value:

    @(x)x.^3.*(2.4e+1./5.0)-1.0./x.^7.*4.8e+1

>>
```

Una vez con estos parámetros establecidos podemos olvidarnos de las derivadas y proceder a graficar ambas funciones.

```
Command Window
>> x = linspace(-100, 100, 1000);
>> y = miFuncion(x);
>> z = Lfuncion(x);
>> subplot(2,1,1);
>> plot(x, y)
>> xlabel('Abscisas')
>> ylabel('Ordenadas')
>> title('Gráfica miFuncion')
>> legend('y = x.^2+2*x+1')
>> subplot(2,1,2)
>> plot(x, z)
>> xlabel('Abscisas')
>> ylabel('Ordenadas')
>> title('Gráfica Lfuncion')
>> legend('l = x^3/5*6*x+8/x^6')
>>
```



En la siguiente imagen analizaremos las integrales en el dominio de la función *"miFuncion "* junto con el área de la región acotada por la gráfica que hemos obtenido anteriormente.

```
Command Window
>> a = -15; %Valor dentro del dominio de nuestra función
>> b = 15; %Valor dentro del dominio de nuestra función
>> c = -100; %Área negativa de la región acotada por nuestra anterior gráfica
>> d = 100; %Área positiva de la región acotada por nuestra anterior gráfica
>> syms x
>> int(miFuncion(x), a, b)

ans =

2280

>> Integral_dominio = ans

Integral_dominio =

2280

>> int(miFuncion(x), c, d)

ans =

2000600/3

>> Integral_area = ans

Integral_area =

2000600/3
```

Es entonces cuando al obtener nuestros cálculos creamos variables simbólicas.

En la siguiente imagen obtendremos $y = 0$.

```
Command Window
>> d = 1; %Valor de x^2
>> f = 2; %Valor de 2x
>> g = 1; %$Valor de 1
>> roots([d, f, g])

ans =

    -1
    -1

>> y = ans

y =

    -1
    -1

>>
```

Hemos obtenido las raíces de x para conocer los valores que hacen $x = 0$ y de esta manera obtener $y = 0$

Para finalizar, obtendremos el punto de inflexión de la función que hemos venido desarrollando y con este resultdo cálcularemos una integral en el intervalo x_0, x_1 que acote un área determinada de nuestra función.

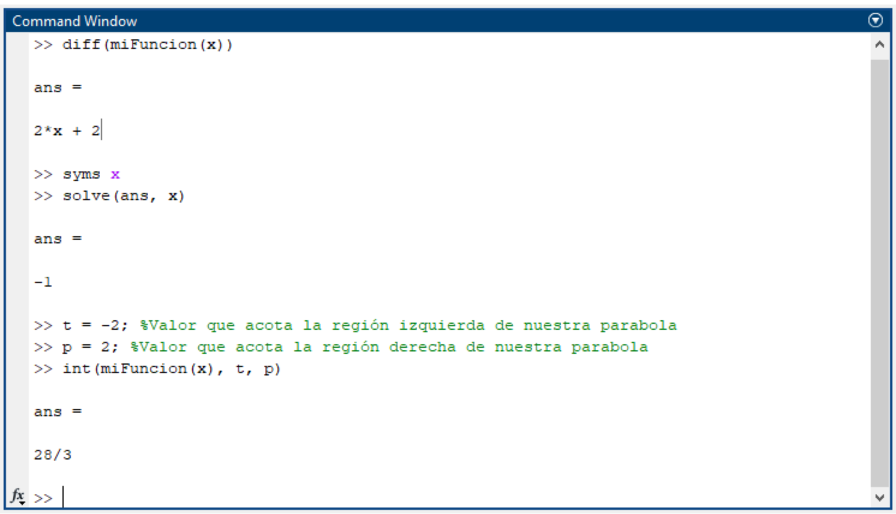
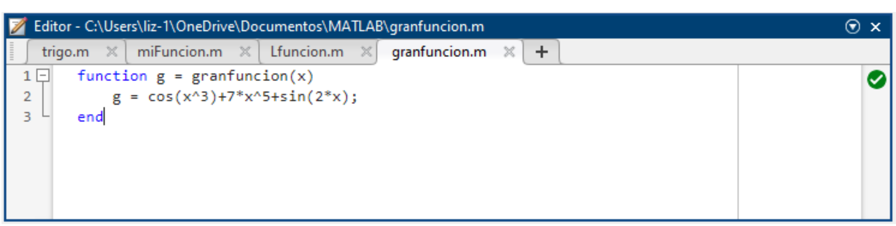


Figura 4: Los valores t, p corresponden a x_0, x_1

- En este punto cálcuraemos derivadas de diversas ordenes, para esto, crearemos una nueva función *"granfuncion "*.



A continuación, nuestras derivadas.

```
>> diff(granfuncion(x))

ans =

2*cos(2*x) - 3*x^2*sin(x^3) + 35*x^4

>> first_dev = ans

first_dev =

2*cos(2*x) - 3*x^2*sin(x^3) + 35*x^4

>> diff(first_dev)

ans =

140*x^3 - 6*x*sin(x^3) - 9*x^4*cos(x^3) - 4*sin(2*x)

>> sec_dev = ans

sec_dev =

140*x^3 - 6*x*sin(x^3) - 9*x^4*cos(x^3) - 4*sin(2*x)

>> diff(sec_dev)

ans =

27*x^6*sin(x^3) - 6*sin(x^3) - 54*x^3*cos(x^3) - 8*cos(2*x) + 420*x^2

>> third_dev = ans

third_dev =

27*x^6*sin(x^3) - 6*sin(x^3) - 54*x^3*cos(x^3) - 8*cos(2*x) + 420*x^2
```

En la anterior gráfica hemos obtenido las derivadas una a una para ir definiendo las variables simbólicas como podemos apreciar.

- En el último punto vamos a verificar nuestra derivada del anterior punto de la manera $h^3(x)+j^3(x)+k^3(x)$ para evaluar nuestro anterior resultado.

Command Window

```
>> syms x;
>> h = cos(x^3);
>> j = 7*x^5;
>> k = sin(2*x);
>> primera_devh = diff(h);
>> primera_devj = diff(j);
>> primera_devk = diff(k);
>> segunda_devh = diff(primera_devh);
>> segunda_devj= diff(primera_devj);
>> segunda_devk = diff( primera_devk);
>> tercera_devh = diff(segunda_devh);
>> tercera_devj = diff(segunda_devj);
>> tercera_devk = diff(segunda_devk);
>> % Ahora obtendremos los resultados de cada derivada de tercer grado
>> tercera_devh

tercera_devh =

27*x^6*sin(x^3) - 54*x^3*cos(x^3) - 6*sin(x^3)

>> tercera_devj

tercera_devj =

420*x^2

>> tercera_devk

tercera_devk =

-8*cos(2*x)

>> tercera_devh + tercera_devj + tercera_devk

ans =

27*x^6*sin(x^3) - 6*sin(x^3) - 54*x^3*cos(x^3) - 8*cos(2*x) + 420*x^2

>> 
```

Es así, como evaluando la derivada de cada función de forma independiente, llegamos a la sumatoria de funciones derivadas de tercer orden y constatamos que obtenemos el mismo resultado que en el punto pasado.

VISUALIZACIÓN

- Utiliza la función plot de MATLAB para visualizar las funciones definidas.
- Agrega etiquetas a los ejes, título y leyendas si es necesario.
- Experimenta con diferentes estilos de líneas, colores y marcadores para mejorar la visualización. Agrega puntos de interés como mínimos, máximos, puntos de inflexión, etc., utilizando MATLAB para calcular estos puntos.

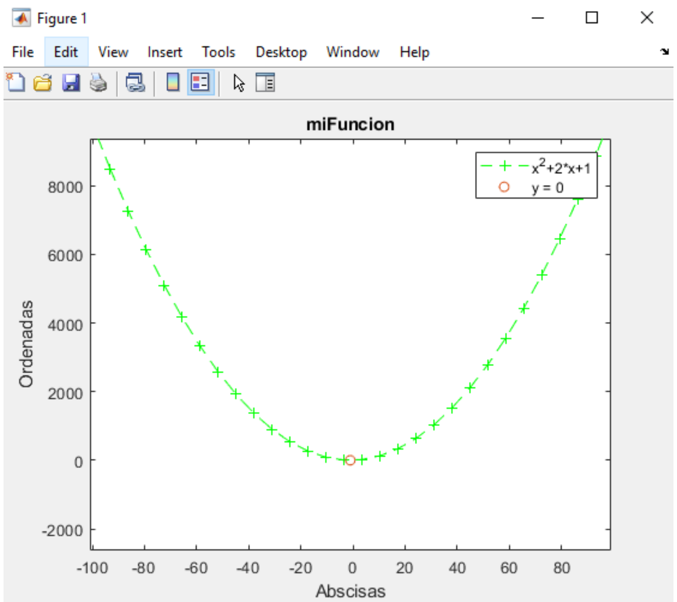
Este punto podemos resolverlo con el siguiente script.

```
Command Window
>> x = linspace(-100, 100, 30);
y1 = miFuncion(x);
plot(x, y1, 'g--+')
xlabel('Abscisas')
ylabel('Ordenadas')
title('miFuncion')
legend ('x^2+2*x+1')
%Acontinuación, pondremos el punto de inflexión de esta función
hold on
x1_inflexion = -1; y_inflexion = 0;
scatter(x1_inflexion,y_inflexion)
hold off
figure
y2 = Lfuncion(x);
plot(x, y2, 'b-.d')
xlabel('Abscisas')
ylabel('Ordenadas')
title('Lfuncion')
legend ('(x^3)/5 * 6*x + 8/(x^6)')
figure
y3 = granfuncion(x);
plot(x, y3, 'r:o')
xlabel('Abscisas')
ylabel('Ordenadas')
title('granfuncion')
legend ('cos(x.^3)+7*x.^5+sin(2*x)')
>>
```

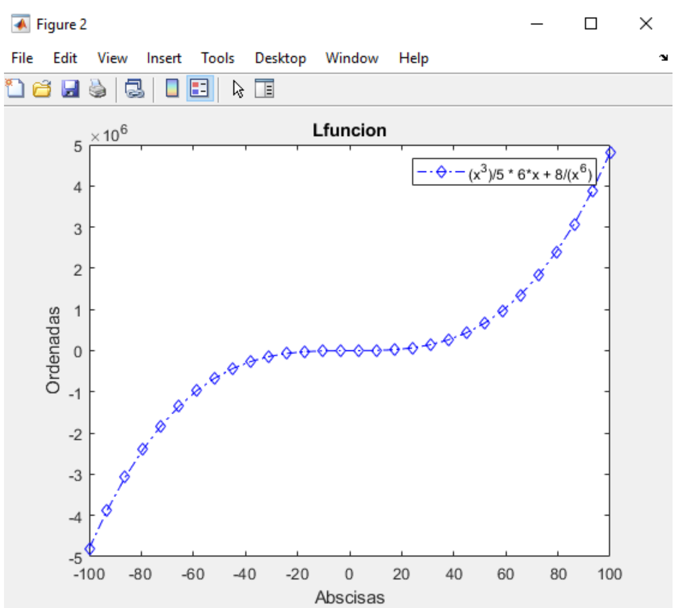
Es entonces que obtenemos nuestras tres gráficas, la primera con $y = 0$. Las otras funciones hemos modificado el color, las lineas y leyendas.

Las gráficas son las siguientes:

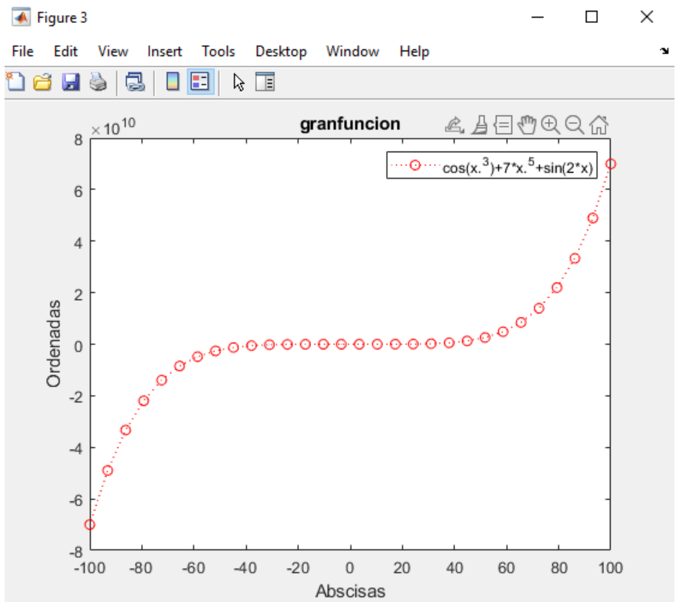
- **miFuncion**



- **Lfuncion**



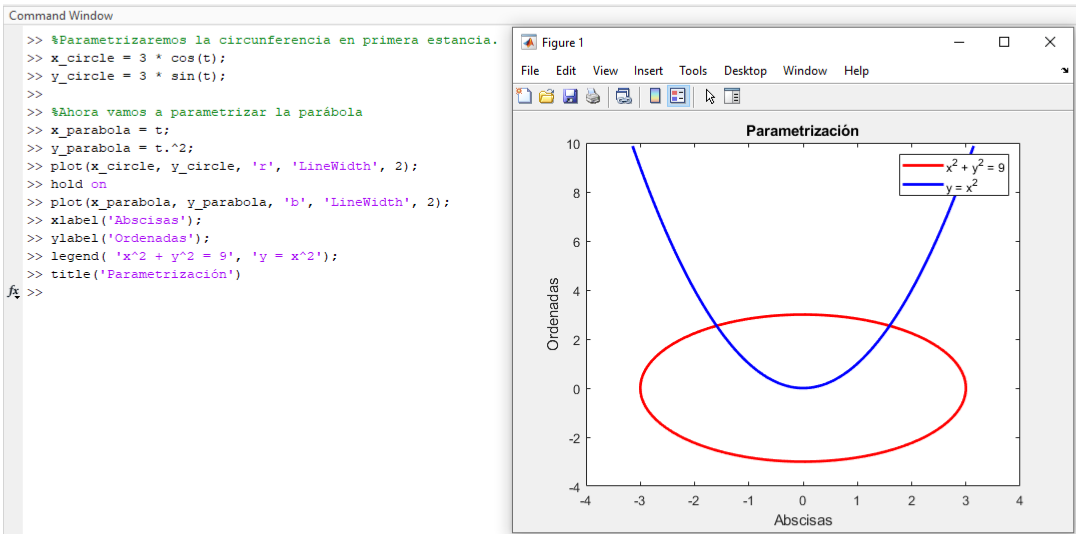
- **granfuncion**



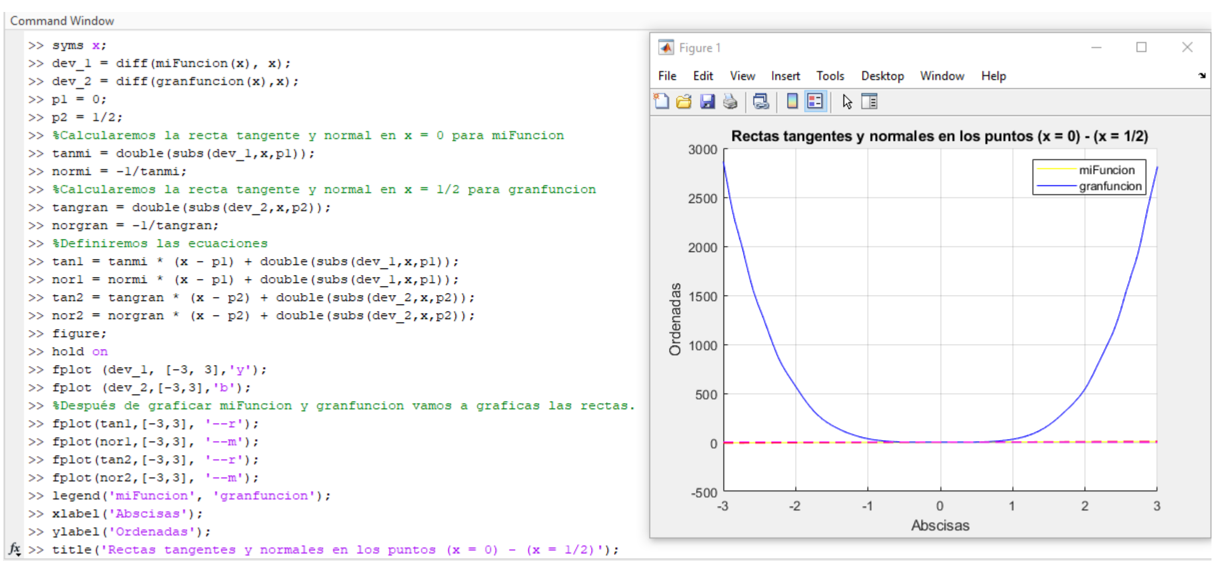
Curvas en el Plano

- Parametriza las curvas $x^2 + y^2 = 9$ y $y = x^2$ en el intervalo $[-4, 4]$.
- Utiliza MATLAB para graficar ambas curvas simultáneamente utilizando *plot*.
- Calcula la recta tangentes y normales a las curvas en los puntos $x = 0$ y $x = 1/2$ utilize el comando diff. Representelas gráficamente, puede usar *quiver* u otras funciones de visualización.

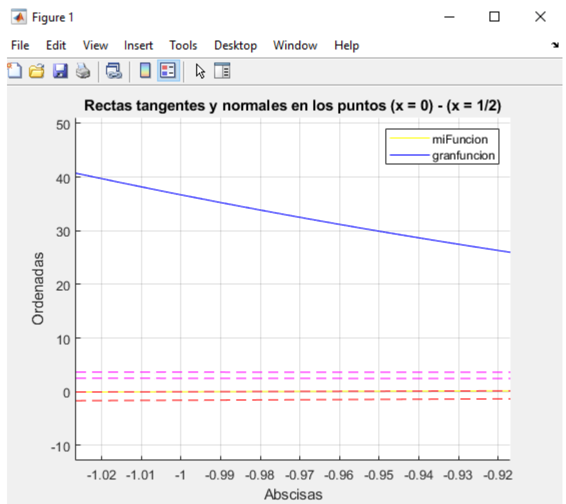
Vamos a desarrollar los dos puntos en la siguiente imagen, mostrando el proceso de parametrización en un intervalo cerrado junto con las gráficas de la circunferencia y la parabola



Ahora trazaremos las rectas tangentes y normales en los puntos $x = 0$ y $x = \frac{1}{2}$ en las funciones *"miFuncion "* y *"granfuncion "*.



Si ampliamos la gráfica podemos ver que aunque parecian solapadas varias de nuestras rectas, estás se comportan de manera paralela la una de otra.



Superficies y Curvas en el espacio

- Considere el paraboloide hiperbólico $z = y^2 - x^2$ para respresentar superficies en el espacio tridimensional y represéntalo gráficamente.
- Utiliza funciones como *surf* o *mesh* para visualizar estas superficies.
- Paramétrice la curva interseccion del paraboloide anterior con el plano $y = 2$ y grafique esa curva.
- Calcule el plano tangente a la superficie en el punto p(1, 2, 3).
- Visualiza los vectores tangentes y normales utilizando quiver3 u otras funciones de visualización en 3D.

Vamos presentar el código **donde hemos aunado todos los puntos a analizar** en tres diferentes imagenes donde hemos distribuido todos los puntos de manera lineal.

Representación del Paraboloide Hiperbólico , trazas en z = 0, y = 0, x = 0.

Command Window

```
>> x = linspace(-60, 60, 100);
y = linspace(-60, 60, 100);
%Usamos mesh
[X, Y] = meshgrid(x, y);
Z = Y.^2 - X.^2;
figure;
surf(X, Y, Z);
hold on;
% Trazas con z = 0 (plano xy)
plot3(x, y, zeros(size(x)), 'r', 'LineWidth', 2);
% Trazas con y = 0 (plano xz)
plot3(x, zeros(size(x)), -x.^2, 'g', 'LineWidth', 2);
% Trazas con x = 0 (plano yz)
plot3(zeros(size(y)), y, y.^2, 'b', 'LineWidth', 2);
xlabel('Profundidad');
ylabel('Largo');
zlabel('Alto');
title('Paraboloide Hiperbólico: z = y^2 - x^2');
t = linspace(-60, 60, 100);
x = t;
y = 2 * ones(size(t));
z = 4 - t.^2;
%Intersección
plot3(x, y, z, 'm', 'LineWidth', 2);
legend('Paraboloide Hiperbólico', 'Trazas con z = 0', 'Trazas con y = 0', 'Trazas con x = 0', 'Curva de Intersección con el Plano (y = 2)', 'Location', 'north
p1 = 1;
p2 = 2;
p3 = 3;
```

Parametrización de la curva de intersección con el plano Y = 2 y Cálculo del plano Tangente en el Punto (1, 2, 3).

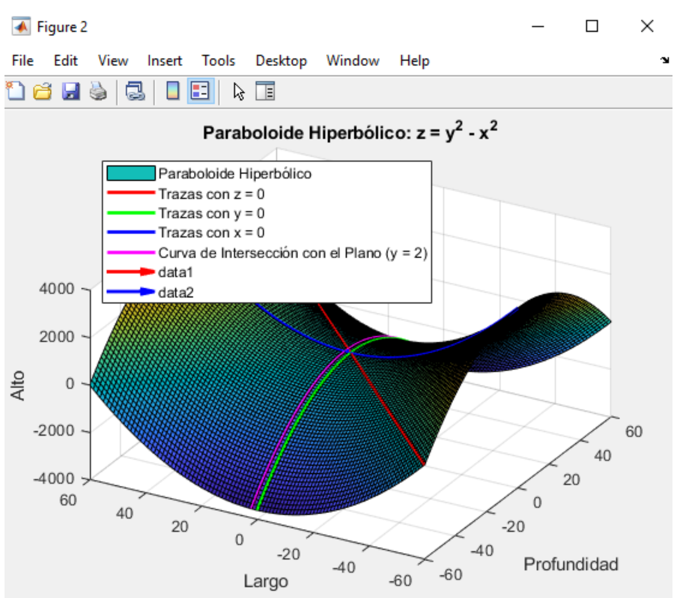
Command Window

```
dz_dx = -2 * p1;
dz_dy = 2 * p2;
dx = 0;
dy = 0;
%Usamos derivadas parciales
plano_tangente = dz_dx * (X - p1) + dz_dy * (Y - p2) + (p3 - dz_dx * p1 - dz_dy * p2);x = linspace(-60, 60, 100);
y = linspace(-60, 60, 100);
%Usamos mesh
[X, Y] = meshgrid(x, y);
Z = Y.^2 - X.^2;
figure;
surf(X, Y, Z);
hold on;
% Trazas con z = 0 (plano xy)
plot3(x, y, zeros(size(x)), 'r', 'LineWidth', 2);
% Trazas con y = 0 (plano xz)
plot3(x, zeros(size(x)), -x.^2, 'g', 'LineWidth', 2);
% Trazas con x = 0 (plano yz)
plot3(zeros(size(y)), y, y.^2, 'b', 'LineWidth', 2);
xlabel('Profundidad');
ylabel('Largo');
zlabel('Alto');
title('Paraboloide Hiperbólico: z = y^2 - x^2');
t = linspace(-60, 60, 100);
x = t;
y = 2 * ones(size(t));
z = 4 - t.^2;
```

Visualización de Vectores Tangentes y Normales.

```
%Intersección
plot3(x, y, z, 'm', 'LineWidth', 2);
legend('Paraboloide Hiperbólico', 'Trazas con z = 0', 'Trazas con y = 0', 'Trazas con x = 0', 'Curva de Intersección con el Plano (y = 2)', 'Location', 'north
p1 = 1;
p2 = 2;
p3 = 3;

dz_dx = -2 * p1;
dz_dy = 2 * p2;
dx = 0;
dy = 0;
%Usamos derivadas parciales
plano_tangente = dz_dx * (X - p1) + dz_dy * (Y - p2) + (p3 - dz_dx * p1 - dz_dy * p2);
%calcular vectores
tangent_vector = [dx, dy, dz_dx];
%normalizar
tangent_vector = tangent_vector / norm(tangent_vector);
% Vector normal
normal_vector = cross([1, 0, dz_dx], [0, 1, dz_dy]);
normal_vector = normal_vector / norm(normal_vector);
x_points = [p1, p1 + tangent_vector(1)];
y_points = [p2, p2 + tangent_vector(2)];
z_points = [p3, p3 + tangent_vector(3)];
quiver3(x_points(1), y_points(1), z_points(1), tangent_vector(1), tangent_vector(2), tangent_vector(3), 'r', 'LineWidth', 2, 'MaxHeadSize', 0.5);

x_points = [p1, p1 + normal_vector(1)];
y_points = [p2, p2 + normal_vector(2)];
z_points = [p3, p3 + normal_vector(3)];
quiver3(x_points(1), y_points(1), z_points(1), normal_vector(1), normal_vector(2), normal_vector(3), 'b', 'LineWidth', 2, 'MaxHeadSize', 0.5);
```

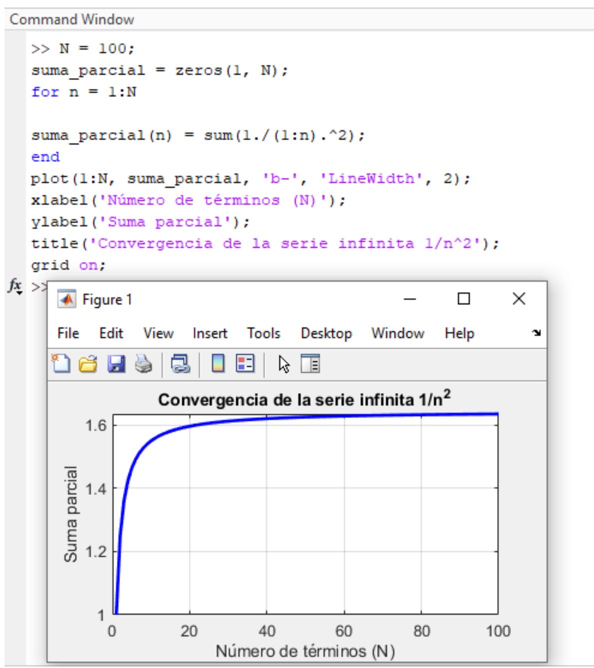
Representación gráfica de todos los puntos.



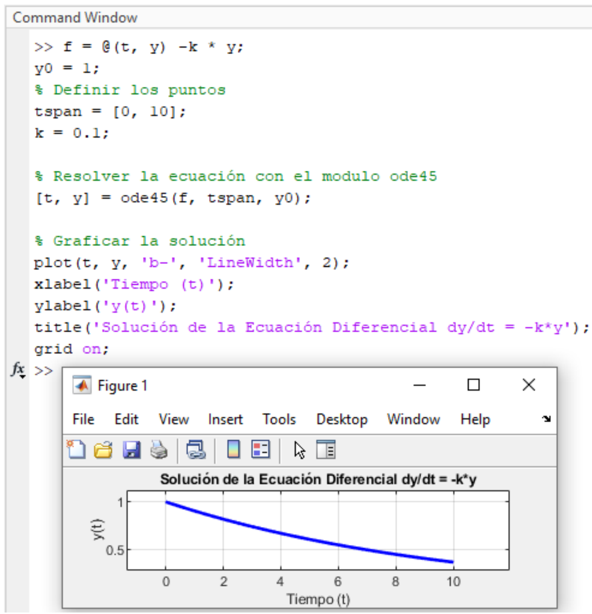
Análisis Adicional.

- Realiza análisis adicionales según tus intereses. Por ejemplo, puedes explorar la convergencia de series infinitas, solución de ecuaciones diferenciales, optimización de funciones, etc.
- Utiliza las herramientas de MATLAB adecuadas para estos análisis, como *sum*, *ode45*, *fmincon*, etc.

Para estos puntos opcionales hemos hecho una convergencia de serie infinita usando el comando *sum*.

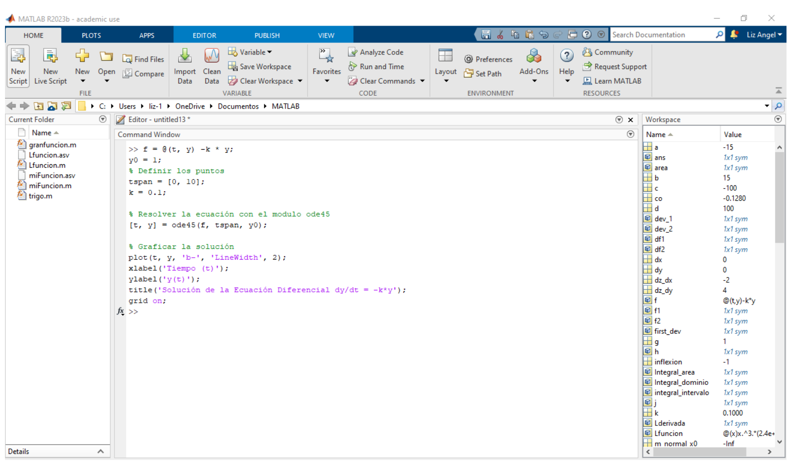


Y para finalizar, hemos desarrollado un problema de Ecuaciones Diferenciales usando el comando *ode45*.



Siendo entonces, este el último punto de nuestras actividades.

Dejamos para finalizar, la imagen de todo nuestro MATLAB después de haber terminado cada uno de los puntos.



Gracias.