**Polinomio de Taylor**

Se utiliza para aproximar el comportamiento de una función, *f(x)*, alrededor de un punto, ***a***, a través de un polinomio de grado *n* de la forma:

Parte de la premisa:

Vamos a pensar en el polinomio más fácil de todos:

Vamos a pensar ahora en un polinomio más complejo, el de grado 1:

Vamos a pensar ahora en un polinomio de grado 2:

Generalizando estas ideas:

Para un Polinomio de Taylor de grado 1 (n=1)

|  |  |
| --- | --- |
| i = 0 | i = 1 |
|  |  |
|  |  |

Para un Polinomio de Taylor de grado 2 (n=2)

|  |  |  |
| --- | --- | --- |
| i = 0 | i = 1 | i = 2 |
|  |  |  |
|  |  |  |

Para un Polinomio de Taylor de grado 3 (n=3)

|  |  |  |  |
| --- | --- | --- | --- |
| i = 0 | i = 1 | i = 2 | i = 3 |
|  |  |  |  |

**Ejemplo 1**

Calcular los Polinomios de Taylor de grado 1, 2 y 3 de la siguiente función alrededor de 0. (**a=0**)

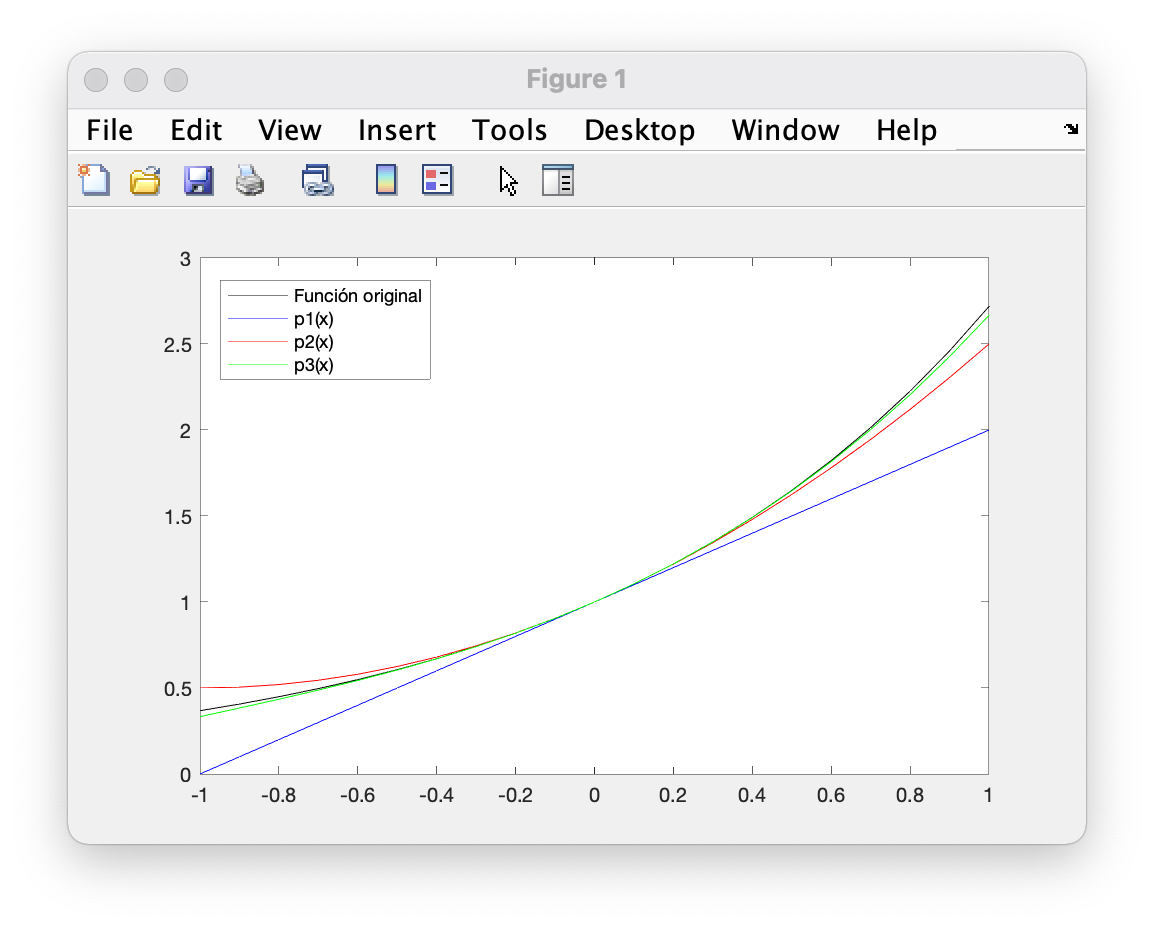
**Polinomio de Maclaurin**

**Ejemplo 2**

Calcular los Polinomios de Taylor de grado 1, 2 y 3 de la siguiente función alrededor de 0, utilizando Matlab y graficar la función y los polinomios en una misma figura en el intervalo de [-1, 1].

Si evalúo una variable simbólica dentro de una función inline o función anónima obtengo como resultado una variable simbólica.

|  |
| --- |
| f=inline('exp(x)');  f=@(x) (exp(x));    syms x  f1=diff(f(x)); %La primer derivada de f  f1=inline(f1);    a=0;    %Polinomio grado 1  p1=f(a)+f1(a)\*(x-a)    %Polinomio grado 2  f2=diff(f(x),2); %La segunda derivada de f  f2=diff(f1(x)); %La primer derivda de la primer derivada de f  f2=inline(f2);    p2=p1+f2(a)/factorial(2)\*(x-a)^2    %Polinomio grado 3  f3=diff(f(x),3);  f3=inline(f3);    p3=p2+f3(a)/factorial(3)\*(x-a)^3    %Gr·fica  x=-1:0.1:1;  y=f(x);  plot(x,y,'k') %FunciÛn original    hold on  p1=inline(p1);  p2=inline(p2);  p3=inline(p3);    plot(x,p1(x),'b')  plot(x,p2(x),'r')  plot(x,p3(x),'g')    legend('Función original','p1(x)','p2(x)','p3(x)') |



**Ejemplo 3**

Encontrar el Polinimio de Taylor de almenos 4 términos de la siguiente función alrededor de 1.2. Estimar el error relativo porcentual en 2 y graficar en el intervalo [0, 2].

|  |
| --- |
| f=inline('3.4\*sin(x)');  a=1.2;    %Polinomio de grado 1  syms x  f1=diff(f(x));  f1=inline(f1);    p1=f(a)+f1(a)\*(x-a);  p1=vpa(expand(p1),5)    %Polinomio de grado 2  f2=diff(f(x),2);  f2=inline(f2);    p2=p1+f2(a)/factorial(2)\*(x-a)^2;  p2=vpa(expand(p2),5)    %Polinomio de grado 3  f3=diff(f(x),3);  f3=inline(f3);    p3=p2+f3(a)/factorial(3)\*(x-a)^3;  p3=vpa(expand(p3),5)    %Gr·fica  x=0:0.1:2;  y=f(x);  plot(x,y,'k') %FunciÛn original    hold on  p1=inline(p1);  p2=inline(p2);  p3=inline(p3);    plot(x,p1(x),'b')  plot(x,p2(x),'r')  plot(x,p3(x),'g')    legend('FunciÛn original','p1(x)','p2(x)','p3(x)')      %Obteniendo errores  vr=f(2);  va1=p1(2);  e1=abs( (vr-va1) / vr ) \* 100    va2=p2(2);  e2=abs( (vr-va2) / vr ) \* 100    va3=p3(2);  e3=abs( (vr-va3) / vr ) \* 100 |

