Usando Gnuplot

Olga María Fimbres Morales

Aproximación de Taylor

Una serie de Taylor es una aproximación de funciones mediante una serie de potencias o suma de potencias enteras de polinomios como

$$(x-a)^n$$

llamados términos de la serie, dicha suma se calcula a partir de las derivadas de la función para un determinado valor o punto a suficientemente derivable sobre la función y un entorno sobre el cual converja la serie. Si esta serie está centrada sobre el punto cero, a=0, se le denomina serie de McLaurin. La serie de Taylor de una función f real o compleja (x) infinitamente diferenciable en el entorno de un número real o complejo a es la siguiente serie de potencias:

$$f(a)+\frac{f'(a)}{1!}(x-a)+\frac{f''(a)}{2!}(x-a)^2+\frac{f^{(3)}(a)}{3!}(x-a)^3+\cdots$$

que puede ser escrito de una manera más compacta como la siguiente sumatoria:

$$sum_{n=0}\frac{f^{(n)}(a)}{n!}(x-a)^n,$$

donde:
n! es el factorial de n
f (n)(a) denota la n-ésima derivada de f para el valor a de la variable respecto de la cual se deriva.

Desarrollo

Para realizar las gráficas es necesario desarrollar la instrucción para cada polonomio de Taylor que desamos hacer de la siguiente manera:

$$l(x) := taylor(f(x), x, 0, n);$$

donde la n representa el grado de la aproximación, el cual puede tomar los valores que deseemos, recordando que cada polonomio debe estar nombrado de forma diferente: l(x), g(x), d(x), etc.
En seguida, debemos especificar las caracteristicas que deseamos posea nuestra gráfica, como el dominio, el rango, los colores, etc.; para eso escribirmos las especificaciones en la instruccion de plot2d, las cuales pueden ser muy variadas. Por ejemplo, si deseamos especificar un dominio para nuestra gráfica lo haremos de la siguiente forma:

$$[x, n, m]$$

donde n siempre sera el menor valor de nuestro dominio. De igual forma el rango se espeficica por:

$$[y, n, m]$$

Podemos especificar diversas cosas en plot2d, por ejemplom, si quisieramos editar el grosor de las lineas el comando seria:

$$[style, [lines, n]]$$

siendo n un número que corresponde a un diferente grosor.

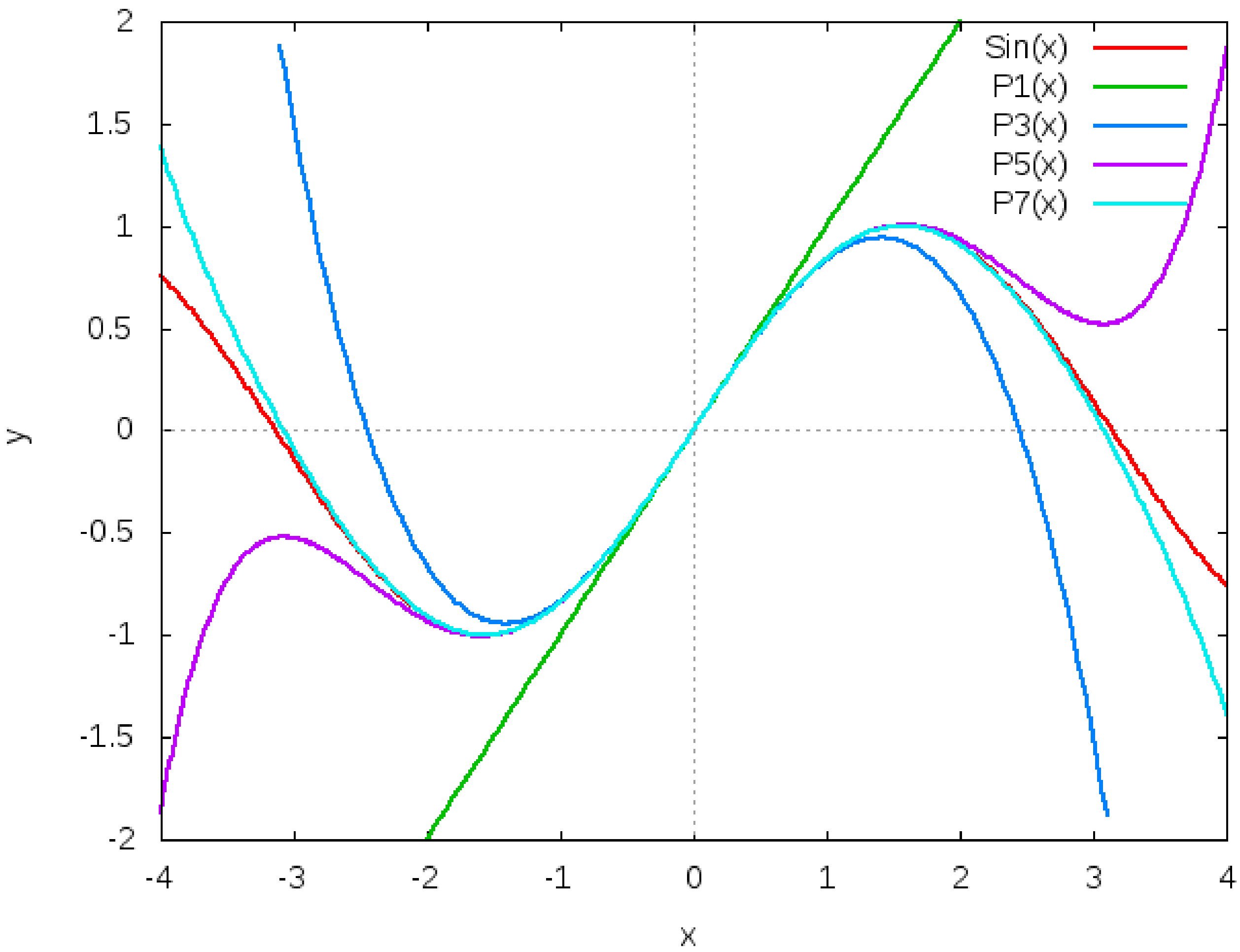
Los colores que acepta para las lineas que graficamos con rojo, verde, azul, magenta, negro y celeste. Si no espeficicamos que color deseamos para cada uno, comienza con el orden preestablecido y lo repite de ser necesario. También podemos especificar la leyenda de cada gráfica en la caja de datos con la instrucción

$$[legend, n1, n2, n3...]$$

donde n1, n2, etc. son todas las funciones que graficamos, siendo cuidadosos de seguir el mismo orden que llevamos al comenzar el documento.

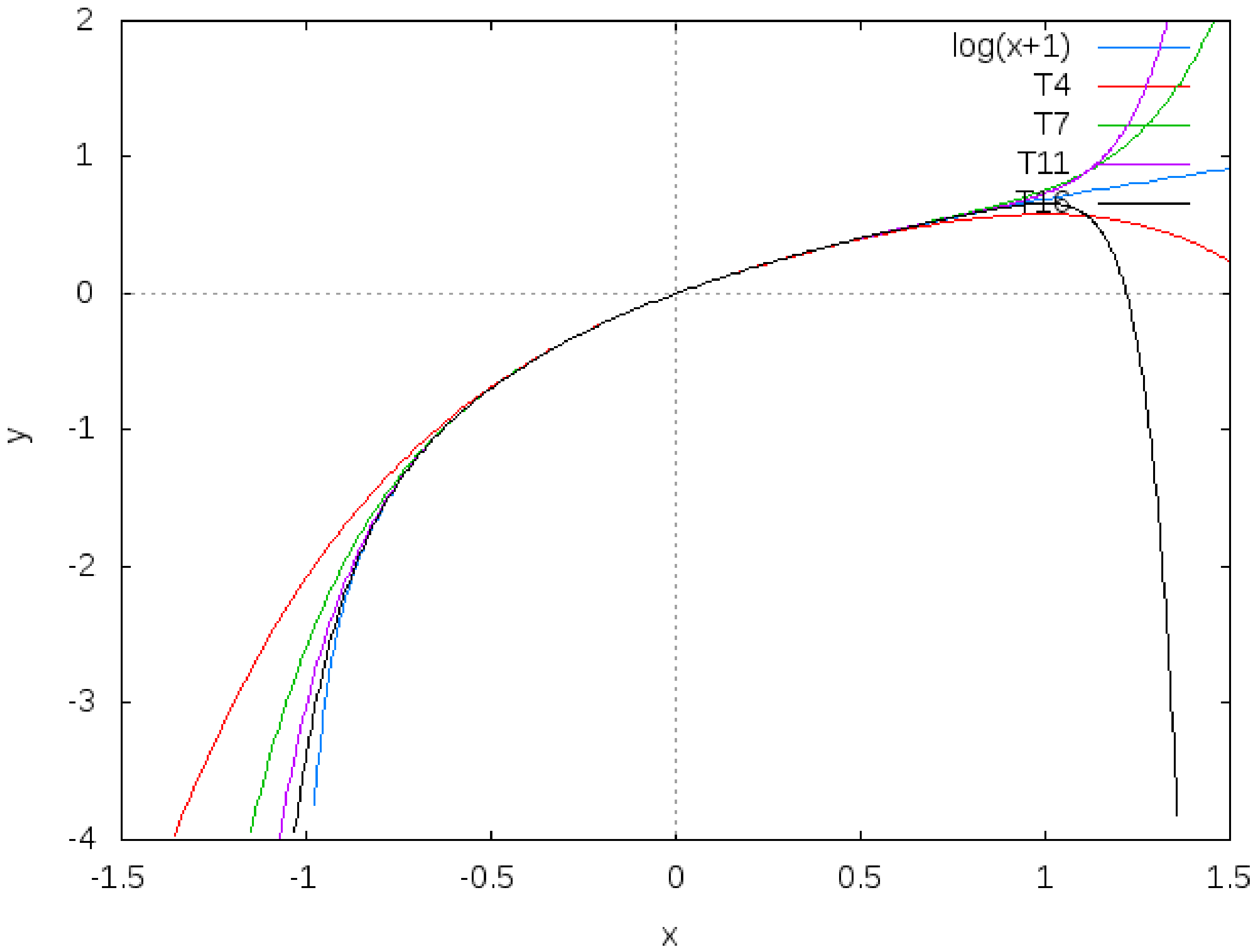
1.- Se pide reproducir exactamente la siguiente gráfica de la aproximación de Taylor de la función sin(x), de aproximación 1, 3, 5 y 7.

```
f(x):= sin(x);
l(x):=taylor(f(x), x, 0, 1);
p(x):=taylor(f(x), x, 0, 3);
t(x):=taylor(f(x), x, 0, 5);
s(x):=taylor(f(x), x, 0, 7);
fortran(f(x));
fortran(p(x));
fotran(t(x));
fortran(l(x));
fotran(s(x));
tex(f(x),p(x),t(x),l(x),s(x));
plot2d ([f(x),l(x),p(x),t(x),s(x)],
        [x, -4, 4], [y, -2, 2],
        [style, [lines,2]],
        [color, red, green, blue, magenta, cyan],
```



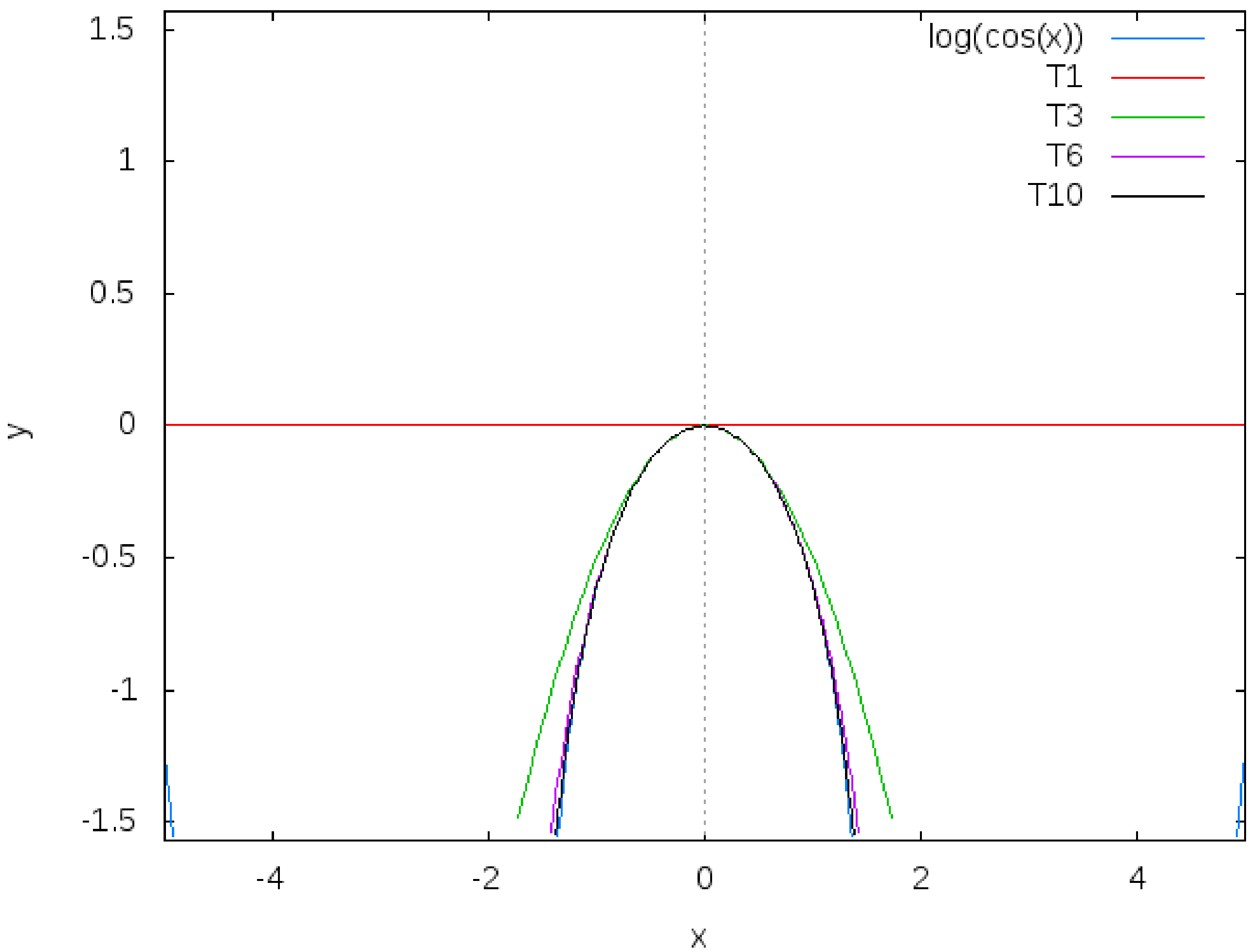
2.- Se pide ahora obtener la aproximación de Taylor de la función log(1+x), para las aproximaciones mostradas.

```
f(x):= log(1+x);
t(x):=taylor(f(x), x, 0, 4);
l(x):=taylor(f(x), x, 0, 7);
m(x):=taylor(f(x), x, 0, 11);
s(x):=taylor(f(x), x, 0, 16);
fortran(t(x);
fortran(l(x));
fotran(m(x));
fortran(s(x));
tex(f(x),p(x),t(x),l(x),s(x));
plot2d ([f(x),t(x),l(x),m(x),s(x)],
   [x, -1.5, 1.5], [y, -4, 2],
   [legend, log(1+x),T4, T7, T11, T16]);
```



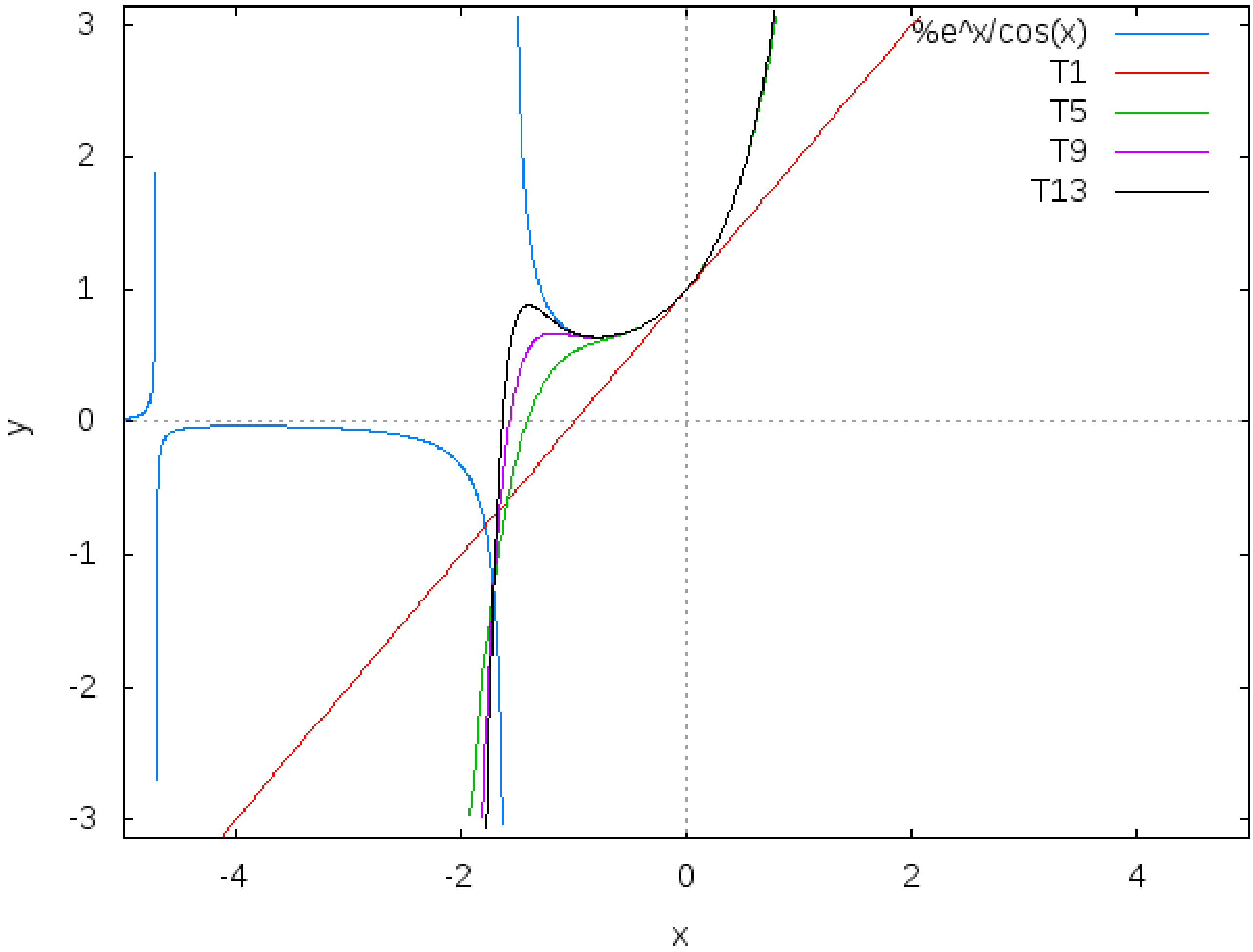
3.- De la misma forma, calcula la aproximación de la serie de Taylor de la función log(cos(x)), alrededor del punto x=0, en el rango (-pi/2, pi/2), usando polinomios de varios grados.

```
f(x):= log(cos(x));
t(x):=taylor(f(x), x, 0, 1);
l(x):=taylor(f(x), x, 0, 3);
m(x):=taylor(f(x), x, 0, 6);
s(x):=taylor(f(x), x, 0, 10);
fortran(t(x);
fotran(l(x));
fotran(m(x));
fotran(s(x));
tex(f(x),p(x),t(x),l(x),s(x));
plot2d ([f(x),t(x),l(x),m(x),s(x)],
   [x, -5, 5], [y, -%pi/2, %pi/2],
   [legend, log(cos(x)), T1, T3, T6, T10]);
```



4.- Calcula las aproximaciones de Taylor de la función exp(x)/cos(x), alrededor de x=0.

```
f(x):= (exp(x))/(cos(x));
t(x):=taylor(f(x), x, 0, 1);
l(x):=taylor(f(x), x, 0, 5);
m(x):=taylor(f(x), x, 0, 9);
s(x):=taylor(f(x), x, 0, 13);
fortran(t(x));
fotran(l(x));
fortran(m(x));
Fortran(s(x));
tex(f(x),p(x),t(x),l(x),s(x));
plot2d ([f(x),t(x),l(x),m(x),s(x)],
    [x, -5, 5], [y, -%pi, %pi],
    [legend, (exp(x)/cos(x)), T1, T5, T9, T13]);
```



5.- De la misma forma, ejemplifica la aproximación de Taylor de la función (1+x) exp(x), alrededor de x=0.

```
f(x):= (1+x)*(exp(x));
t(x):=taylor(f(x), x, 0, 1);
l(x):=taylor(f(x), x, 0, 3);
m(x):=taylor(f(x), x, 0, 6);
s(x):=taylor(f(x), x, 0, 10);
fortran(t(x));
fotran(l(x));
fotran(m(x));
fotran(s(x));
tex(f(x),p(x),t(x),l(x),s(x));
plot2d ([f(x),t(x),l(x), m(x), s(x)],
   [x, -3, 3], [y, -4, 4],
   [legend, f(x), T1, T3, T6, T10]);
```