Reporte de práctica 4

Alejandro Guirado García

17 de marzo de 2015

1. Introducción

El objetivo de esta actividad es aprender a crear gráficas de funciones mediante el programa gnuplot. gnuplot es un programa muy flexible para generar gráficas de funciones y datos.

Este programa es compatible con los sistemas operativos más populares (Linux, UNIX, Windows, Mac OS X...). El origen de gnuplot data de 1986. Ademas, puede producir sus resultados directamente en pantalla, así como en multitud de formatos de imagen, como PNG, EPS, SVG, JPEG, etc. Se puede usar interactivamente o en modo por lotes (batch), usando scripts. En esta actividad lo usaremos para calcular series de taylor. En matemáticas, una serie de Taylor es una aproximación de funciones mediante una serie de potencias o suma de potencias enteras de polinomios como:

$$(x-a)^n$$

La serie de Taylor de una función real o compleja 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 suma:

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

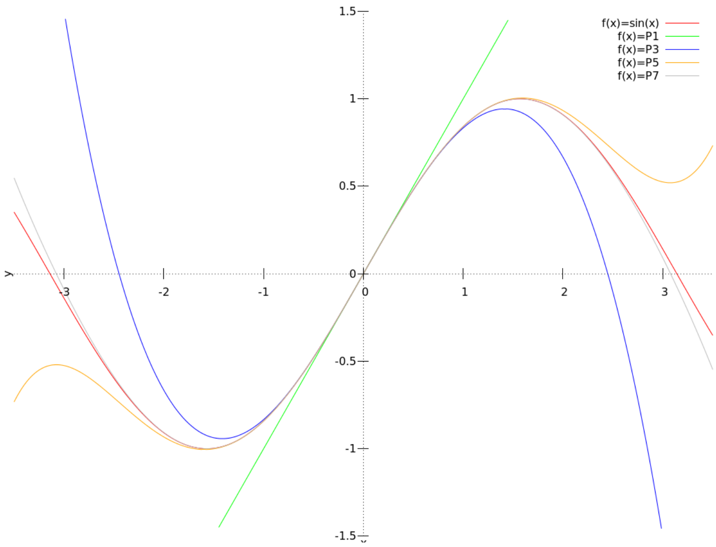
2. Función sin(x)

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

2.1. Código en Maxima

```
f(x):=sin(x);
p1(x):=taylor(f(x), x, 0, 1);
p3(x):=taylor(f(x), x, 0, 3);
p5(x):=taylor(f(x), x, 0, 5);
p7(x):=taylor(f(x), x, 0, 7);
fortran(p1(x));
fortran(p3(x));
```

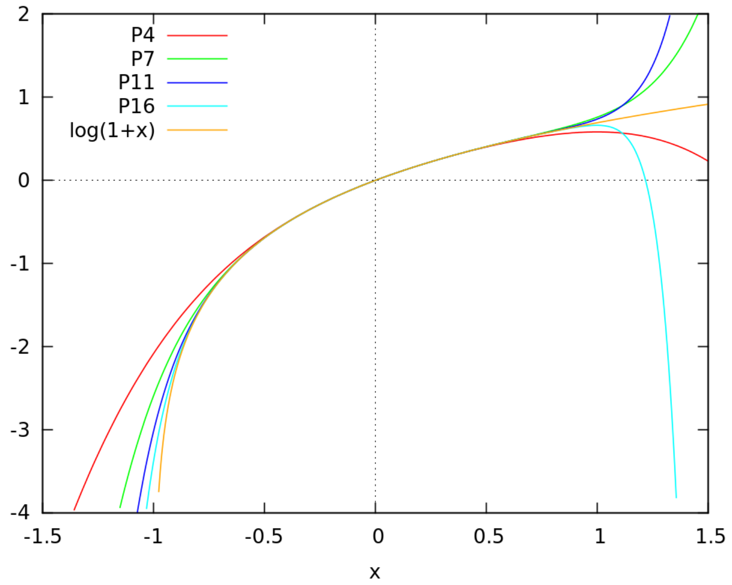
```
fortran(p5(x));
fortran(p7(x));
tex(p1(x));
tex(p3(x));
tex(p5(x));
tex(p7(x));
plot2d ([f(x),p1(x), p3(x), p5(x), p7(x)], [x, -3.5, 3.5], [y, -1.5, 1.5],
[color,red,green,blue,orange,gray],[legend, "f(x)=sin(x)", "f(x)=P1", "f(x)=P3", "f(x)=P5", "f(x)=
[axes, true], [ylabel,"y"], [xlabel,"x"],[box, false],
[gnuplot_preamble,"set ylabel 'y'; set xlabel 'x' "]);
```



3. Función Log(x+1)

Se aproximara con polinomios de grado 4, 7, 11 y 16.

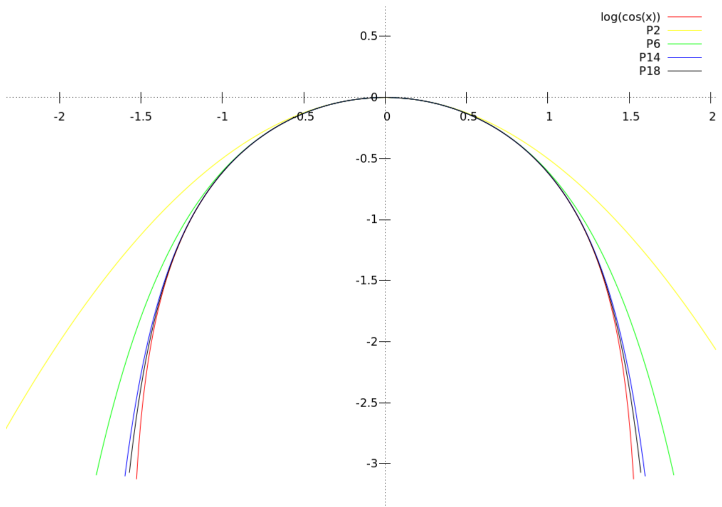
```
f(x):= (1+x)*exp(x);
T3(x):=taylor(f(x), x, 0, 3);
T9(x):=taylor(f(x), x, 0, 9);
T13(x):=taylor(f(x), x, 0, 13);
T15(x):=taylor(f(x), x, 0, 15);
fortran(T3(x));
fortran(T9(x));
fortran(T13(x));
fortran(T15(x));
tex(T3(x));
tex(T9(x));
tex(T13(x));
tex(T15(x));
plot2d ([f(x),T3(x), T9(x), T13(x), T15(x)], [x, -6, 2], [y, -2, 6],[legend, "(1+x)exp(x)", "P3",
```



4. Función log[cos(x)]

Se aproximara con polinomios de grado 2, 6, 14 y 18. En el rango [-pi/2 a pi-2]

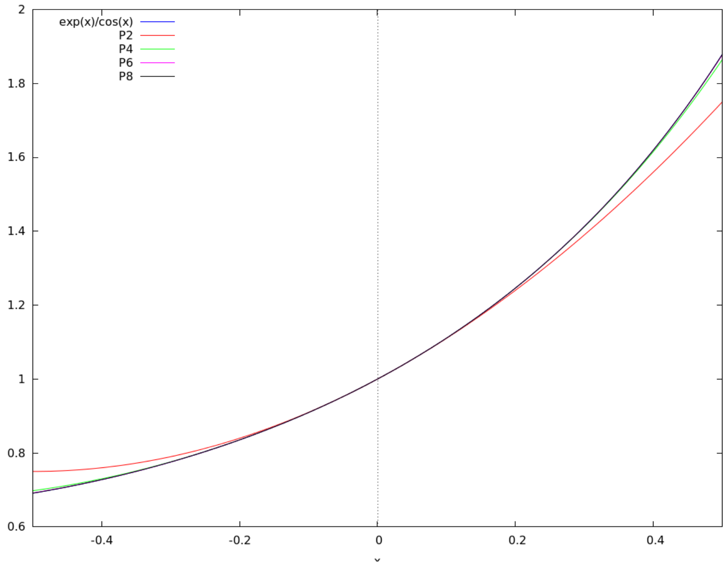
```
f(x):= log(cos(x));
P2(x):=taylor(f(x), x, 0, 2);
P6(x):=taylor(f(x), x, 0, 6);
P14(x):=taylor(f(x), x, 0, 14);
P18(x):=taylor(f(x), x, 0, 18);
fortran(P2(x));
fortran(P6(x));
fortran(P14(x));
fortran(P18(x));
tex(P2(x));
tex(P6(x));
tex(P14(x));
tex(P18(x));
plot2d ([f(x),P2(x), P6(x), P14(x), P18(x)],[x, -%pi, %pi],[y,-%pi, %pi],[color,red,yellow ,green,
```



5. Función exp(cos(x))

Se pide aproximar en las cercanias de x=0. En este caso se tomaron valores de (2,4,6,8) para los polinomios de Taylor.

```
f(x):=exp(x)/cos(x);
P2(x):=taylor(f(x), x, 0, 2);
P4(x):=taylor(f(x), x, 0, 4);
P6(x):=taylor(f(x), x, 0, 6);
P8(x):=taylor(f(x), x, 0, 8);
fortran(P2(x));
fortran(P4(x));
fortran(P6(x));
fortran(P8(x));
tex(P2(x));
tex(P4(x));
tex(P6(x));
tex(P8(x));
plot2d ([f(x),P2(x), P4(x), P6(x), P8(x)], [x, -0.5, 0.5],[legend, "exp(x)/cos(x)", "P2", "P4", "P
```



6. Función exp(1+x)

Se pide aproximar en las cercanias de x=0. En este caso se tomaron valores de (3,9,13,15) para los polinomios de Taylor.

```
f(x):= (1+x)*exp(x);
P3(x):=taylor(f(x), x, 0, 3);
P9(x):=taylor(f(x), x, 0, 9);
P13(x):=taylor(f(x), x, 0, 13);
P15(x):=taylor(f(x), x, 0, 15);
fortran(P3(x));
fortran(P9(x));
fortran(P13(x));
fortran(P15(x));
tex(P3(x));
tex(P9(x));
tex(P13(x));
```

```
tex(P15(x));
plot2d ([f(x),P3(x), P9(x), P13(x), P15(x)], [x, -5, 3], [y, -3, 5],[legend, "(1+x)exp(x)", "P3",
```