Polinomios de Taylor en Gnuplot

Paulina Valenzuela Coronado

Marzo 2015

1. El polinomio de Taylor

TEOREMA DE TAYLOR. Sea f continua en [a, b] y con derivadas hasta de orden n continuas también en este intervalo cerrado; supóngase que f (n+1) (x) existe en (a,b), entonces para x y xoÎ (a,b) se tiene: (a, x), entonces se cumple que:

$$P_T(x) = f(x_o) + f'(x_o)(x - x_o) + \frac{f^{(2)}(x_o)}{2!}(x - x_o)^2 + \cdots + \frac{f^{(n)}(x_o)}{n!}(x - x_o)^n + E_n$$

Donde:

$$E_n = \frac{f^{(n+1)}(c)}{(n+1)!}(x - x_o)^{n+1}$$

2. Aproximaciones con el polinomio de Taylor

Esta actividad consistió en aproximar diferentes polinomios de Taylor para una función. Usamos el programa xMaxima para estimar el polinomio de grado n, después de esto se uso la herramienta Gnuplot para gráficar.

2.1. Aproximación de la función $Sin(x)$

Código en Maxima

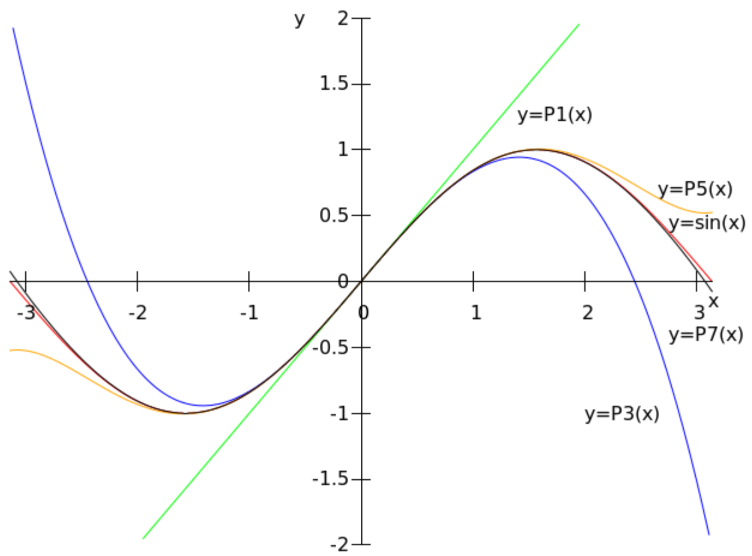
```
f(x):= sin(x);
P1(x):=taylor(f(x), x, 0, 1);
P3(x):=taylor(f(x), x, 0, 3);
P5(x):=taylor(f(x), x, 0, 5);
P7(x):=taylor(f(x), x, 0, 7);

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, -%pi, %pi],
[y, -2, 2], [color, red, green, blue, orange, black],
[axes, solid],[label,["y=P1(x)",1.4,1.27],
["y=P5(x)",2.65,0.7],["y=sin(x)",2.75,0.45],
["y=P7(x)",2.75, -0.4], ["y=P3(x)",2 ,-1],
["y",-0.6,2],["x", 3.1,-0.15]],
[ylabel,"Sin(x)"], [xlabel,"x"],[box,false],
[legend, false]);
```

Gráfica:



2.2. Aproximación de la función $log(1 + x)$

Código en Maxima

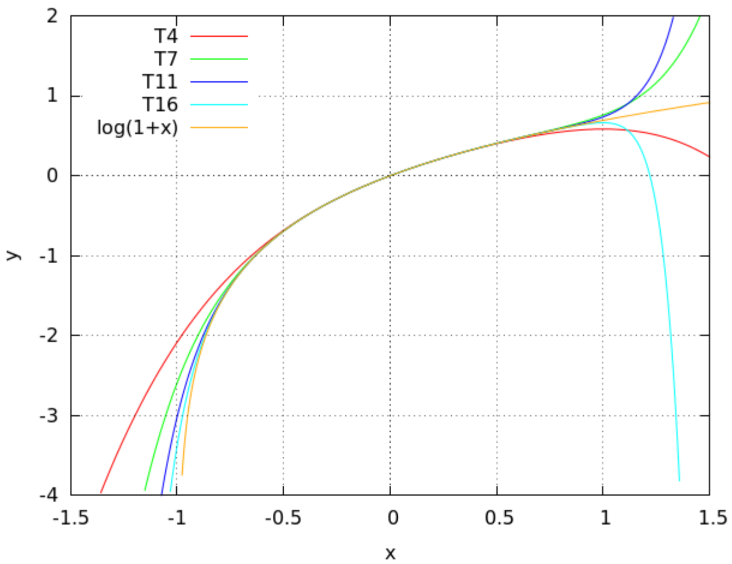
```
f(x):=log(1+x);
T4(x):=taylor(f(x), x, 0, 4);
T7(x):=taylor(f(x), x, 0, 7);
T11(x):=taylor(f(x), x, 0, 11);
T16(x):=taylor(f(x), x, 0, 16);

fortran(T4(x));
fortran(T7(x));
fortran(T11(x));
fortran(T16(x));

tex(T4(x));
tex(T7(x));
tex(T11(x));
tex(T16(x));

plot2d ([T4(x), T7(x), T11(x), T16(x), f(x)], [x, -1.5, 1.5],
[y, -4,2], [color, red, green, blue, cyan, orange], grid2d,
[gnuplot_preamble, "set key left"], [axes, true],
[xlabel,"x"], [ylabel, "y"],
 [legend, "T4", "T7", "T11", "T16", "log(1+x)"]);
```

Gráfica:



2.3. Aproximación de la función $log(cos(x))$

Código en Maxima

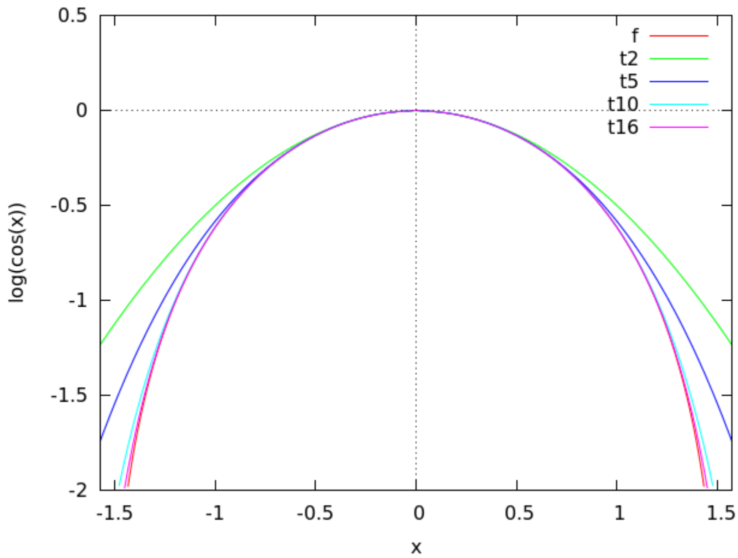
```
f(x):=log(cos(x));
t2(x):=taylor(f(x), x, 0, 2);
t5(x):=taylor(f(x), x, 0, 5);
t10(x):=taylor(f(x), x, 0, 10);
t16(x):=taylor(f(x), x, 0, 16);

fortran(t2(x));
fortran(t5(x));
fortran(t10(x));
fortran(t16(x));

tex(t2(x));
tex(t5(x));
tex(t10(x));
tex(t16(x));

plot2d ([f(x), t2(x), t5(x), t10(x), t16(x)],
 [x, -0.5*%pi, 0.5*%pi], [y, -2,0.5], [axes, true],
 [xlabel,"x"], [ylabel, "log(cos(x))"],
 [color, red, green, blue, cyan, magenta],
 [legend, "f", "t2", "t5", "t10", "t16"]);
```

Gráfica:



2.4. Aproximación de la función $\frac{e^x}{cos(x)}$

Código en Maxima

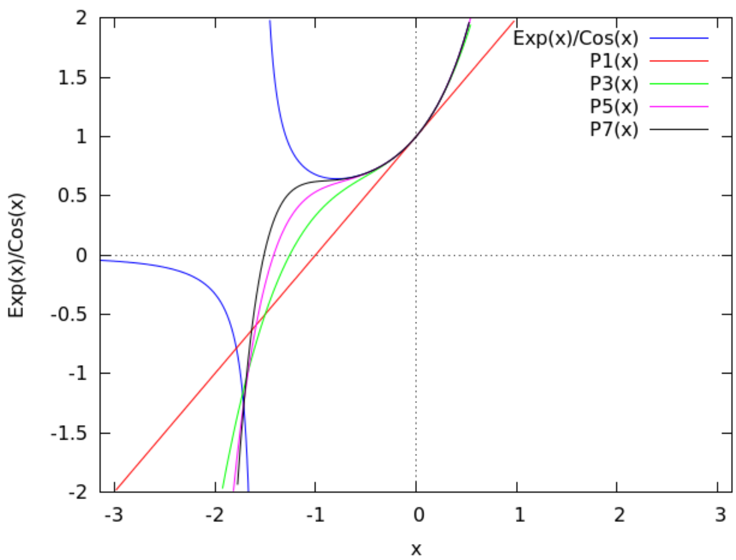
```
f(x):= exp(x)/cos(x);
t(x):=taylor(f(x), x, 0, 1);
t2(x):=taylor(f(x), x, 0, 3);
t3(x):=taylor(f(x), x, 0, 5);
t4(x):=taylor(f(x), x, 0, 7);

fortran(f(x));
fortran(t(x));
fortran(t2(x));
fortran(t3(x));
fortran(t4(x));

tex(f(x));
tex(t(x));
tex(t2(x));
tex(t3(x));
tex(t4(x));

plot2d ([f(x),t(x),t2(x),t3(x),t4(x)],
[x, -%pi, %pi], [y, -2, 2],
[legend, "Exp(x)/Cos(x)","P1(x)","P3(x)","P5(x)","P7(x)"] ,
 [xlabel,"x"], [ylabel,"Exp(x)/Cos(x)"])
```

Gráfica



2.5. Aproximación de la función $e^x(x+1)$

Código en Maxima

```
f(x):=(1+x)*exp(x);
t(x):=taylor(f(x), x, 0, 1);
t3(x):=taylor(f(x), x, 0, 3);
t5(x):=taylor(f(x), x, 0, 5);
t7(x):=taylor(f(x), x, 0, 7);

fortran(t(x));
fortran(t3(x));
fortran(t5(x));
fortran(t7(x));

tex(t(x));
tex(t3(x));
tex(t5(x));
tex(t7(x));

plot2d ([f(x), t(x), t3(x), t5(x), t7(x)], [x, -6, 2],
[y, -2, 6], [gnuplot_preamble, "set key left"],
[axes, true], [xlabel,"x"], [ylabel, "(1+x)*exp(x)"],
 [color, red, green, blue, magenta, cyan],
 [legend, "(1+x)*exp(x)", "t", "t3", "t5", "t7"], [box, true]);
```

Gráfica: