Graficando series de Taylor

Rios Quijada Danira

28 de Febrero de 2015

1. Teorema de Taylor

En cálculo, el teorema de Taylor, recibe su nombre del matemático británico Brook Taylor, quien lo enunció en 1712. Este teorema permite obtener aproximaciones polinómicas de una función en un entorno de cierto punto en que la función sea diferenciable.

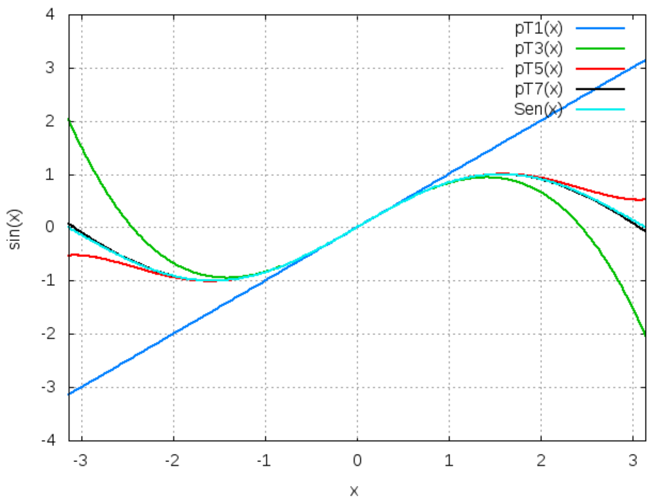
En esta actividad hicimos varias series de taylor, de distintos grados para distintas funciones, graficando los resultados con maxima.

2. sin(x)

2.1. Código

```
 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);
pts([[0,0]]);
tex(P1(x));
tex(P3(x));
tex(P5(x));
tex(P7(x));
plot2d ([P1(x), P3(x), P5(x), P7(x), f(x)], [x, -%pi, %pi],
[color, blue, green, red, black, cyan],
[box,true],[gnuplot_preamble, "set grid"],
[style,[lines,2]],[legend,"pT1(x)","pT3(x)", "pT5(x)", "pT7(x)", "Sen(x)"],
[axes, true],[ylabel, "sin(x)"]);
```

2.2. Gráfica

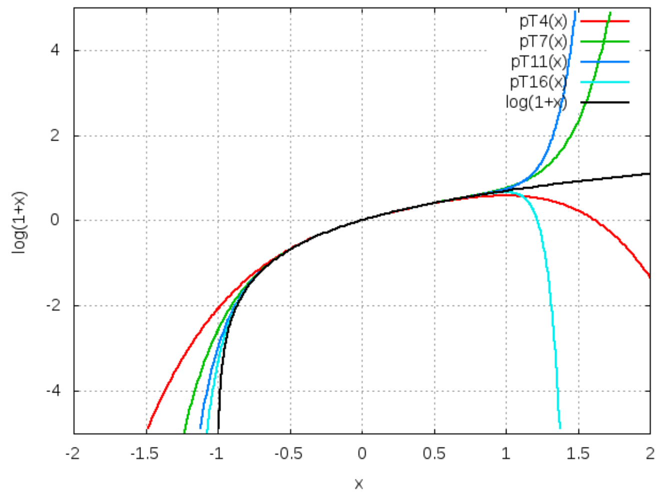


3. log (1+x)

3.1. Código

```
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);
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],[style,[lines,2]],[color, red, green, blue, cyan, orange],
[box,true],[gnuplot_preamble, "set grid"],
[legend, "pT4(x)", "pT7(x)", "pT11(x)", "pT16(x)", "log(1+x)"],
[axes, true],[xlabel,"x"], [ylabel, "log(1+x)"]);
```

3.2. Gráfica

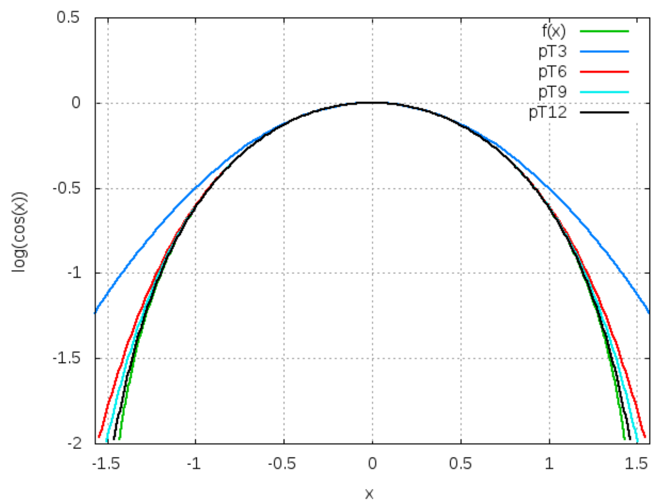


4. log(cos(x))

4.1. Código

```
f(x):=log(cos(x));
t3(x):=taylor(f(x), x, 0, 3);
t6(x):=taylor(f(x), x, 0, 6);
t9(x):=taylor(f(x), x, 0, 9);
t12(x):=taylor(f(x), x, 0, 12);
tex(t3(x));
tex(t6(x));
tex(t9(x));
tex(t12(x));
plot2d ([f(x), t3(x), t6(x), t9(x), t12(x)], [x, -0.5*%pi, 0.5*%pi], [y, -2,0.5],
[color, green, blue, red, cyan, orange],[box,true],[style,[lines,2]],
[legend, "f(x)", "pT3", "pT6", "pT9", "pT12"],[gnuplot_preamble, "set grid"],
 [axes, true], [xlabel,"x"], [ylabel, "log(cos(x))"]);
```

4.2. Gráfica

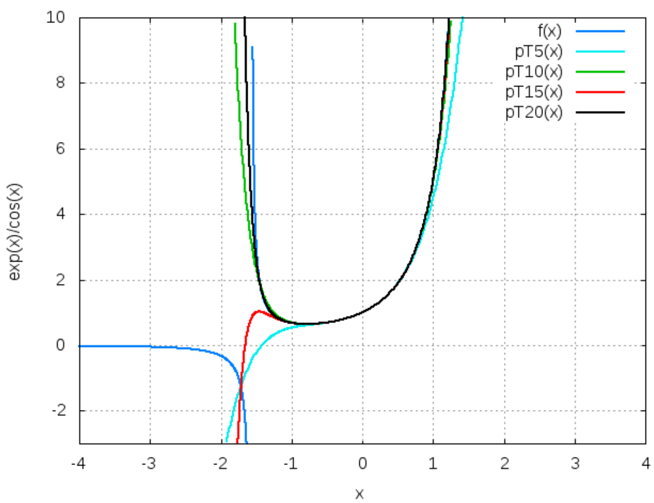


5. exp(x)/cos(x)

5.1. Código

```
f(x):=exp(x)/cos(x);
p5(x):=taylor(f(x), x, 0, 5);
p10(x):=taylor(f(x), x, 0, 10);
p15(x):=taylor(f(x), x, 0, 15);
p20(x):=taylor(f(x), x, 0, 20);
tex(p5(x));
tex(p10(x));
tex(p15(x));
tex(p20(x));
plot2d ([f(x), p5(x), p10(x), p15(x), p20(x)], [x, -4, 4], [y, -3,10],
[color, blue, cyan, green, red, orange],[box,true],[style,[lines,2]],
[legend, "f(x)", "pT5(x)", "pT10(x)", "pT15(x)","pT20(x)"],
[gnuplot_preamble, "set grid"],
[axes, true,centering], [xlabel,"x"], [ylabel, "exp(x)/cos(x)"]);
```

5.2. Gráfica



6. (1+x)(exp(x))

6.1. Código

```
f(x):=(1+x)*exp(x);
t5(x):=taylor(f(x), x, 0, 5);
t10(x):=taylor(f(x), x, 0, 10);
t15(x):=taylor(f(x), x, 0, 15);
t20(x):=taylor(f(x), x, 0, 20);
tex(t5(x));
tex(t10(x));
tex(t15(x));
tex(t20(x));
plot2d ([t5(x), t10(x), t15(x), t20(x), f(x)], [x, -16, 16], [y, -16,16],
[box,true],[style,[lines,2]],[gnuplot_preamble, "set grid"],
[color, red, green, blue, cyan, orange],
[legend, "f(x)", "pT5", "pT10", "pT15", "pT20"],
[axes, true], [xlabel,"x"], [ylabel, "(1+x)*exp(x)"]);
```

6.2. Gráfica