Reporte del Producto 4

Ana Gabriela Carretas Talamante

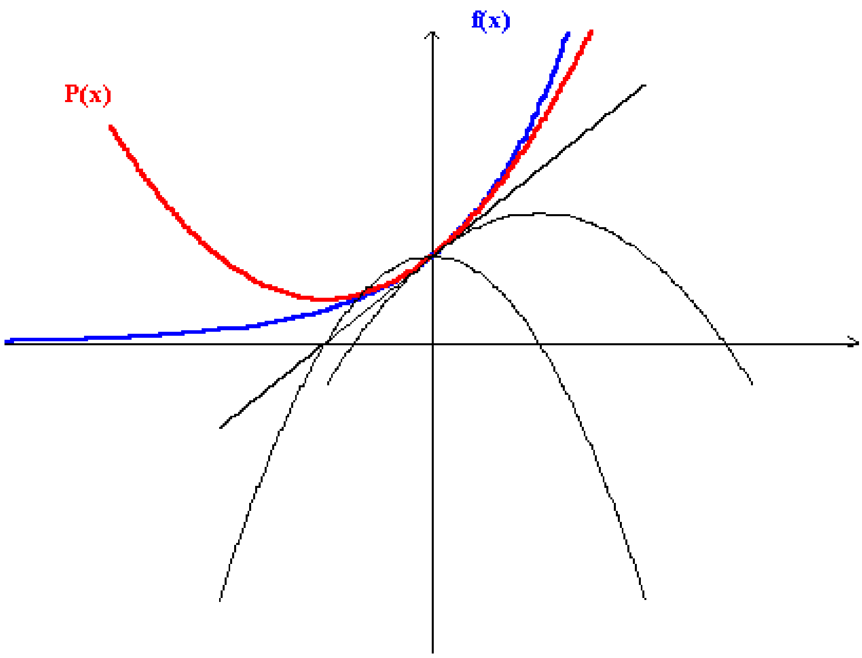
09 de marzo de 2015

1. ¿Qué es un Polinomio de Taylor?

Antes de abordar la relación que hay entre "Taylorz los polinomios, tendremos que escrutar los orígenes de esta útil herramienta, misma que precede al cálculo integral.

Sabemos que la recta tangente, como la mejor aproximación lineal a la gráfica de f en las cercanías del punto de tangencia $(x_o, f(x_o))$, es aquella recta que pasa por el mencionado punto y tiene la misma pendiente que la curva en ese punto (primera derivada en el punto), lo que hace que la recta tangente y la curva sean prácticamente indistinguibles en las cercanías del punto de tangencia.

Veamos qué sucede si en lugar de aproximarnos con una recta, tratamos de hacerlo con una parábola, es decir, tratemos de encontrar *"la parábola tangente"*.



Naturalmente a esta parábola $f(P) = a + b(x - x_o) + c(x - x_o)$ debemos pedirle que pase por el punto, que tenga la misma inclinación (primera derivada) y la misma concavidad que la parábola (segunda derivada), es decir debemos pedirle:

- $P(x_o) = f(x_o)$
- $P'(x_o) = f'(x_o)$
- $P''(x_o) = f''(x_o)$

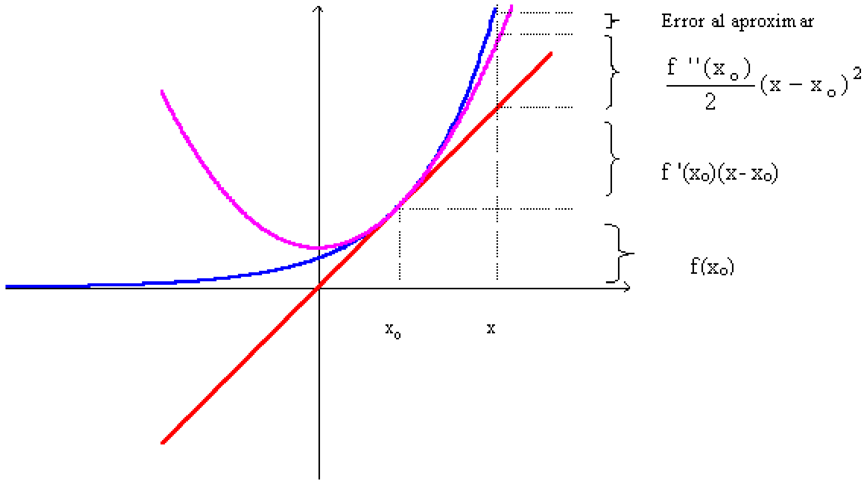


Figura 1: Ecuación de $P(x)$ gráficamente

quedando la ecuación de la parábola que mejor aproxima a la curva en las cercanías de $(x_o, f(x_o))$, como:

$$f(x_o) = f(x_o) + f'(x_o)(x - x_o) + \frac{f''(x_o)}{2}(x - x_o)$$

Dado un polinomio cualquiera podemos expresarlo en potencias de $(x - x_0)$ para cualquier x_o. Sus coeficientes estarán en función de las derivadas en x_o. Asimismo, si conocemos las derivadas en un punto x_o, podemos encontrar un polinomio de grado n que satisfaga la mejor aproximación a la función. Obteniendo entonces un **Polinomio de Taylor de grado n para f, en el punto** x_o de esta manera:

$$P_n(x) = f(x_o) + f'(x_o)(x - x_o) + \frac{f^{(2)}(x_o)}{2}(x - x_o)^2 + \ddots + \frac{f^{(n)}(x_o)}{n}(x - x_o)^n$$

Esta aproximación tiene tres ventajas importantes:

- La derivación e integración de una de estas series se puede realizar término a término, y resultan operaciones triviales.
- Se puede utilizar para calcular valores aproximados de funciones.
- Es posible calcular la optimidad de la aproximación.

2. Graficando funciones en `Maxima`

2.1. Polinomios de Taylor para $\sin(x)$

Se pidió que graficáramos la aproximación de Taylor para $\sin(x)$.

Código en `Maxima`

```
f(x):=sin(x);
T1(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(T1(x));
fortran(T3(x));
```

```
fortran(T5(x));
fortran(T7(x));
text(T1(x));
text(T3(x));
text(T5(x));
text(T7(x));
plot2d ([f(x),T1(x),T3(x),T5(x),T7(x)],[x, -3.5, 3.5], [y, -1.5, 1.5,
[grid2d,true],[color,red,green,blue,orange,gray],[legend,false],
[label,["y=sin(x)",3,0.2],["y=P1(x)",1.5,1.5],["y=P3(x)",2.3,-1],
["y=P5(x)",3,0.85],["y=P7(x)",3,-0.65]],[axes, solid],[box,false],
[xlabel,"x"], [ylabel,"y"]);
```

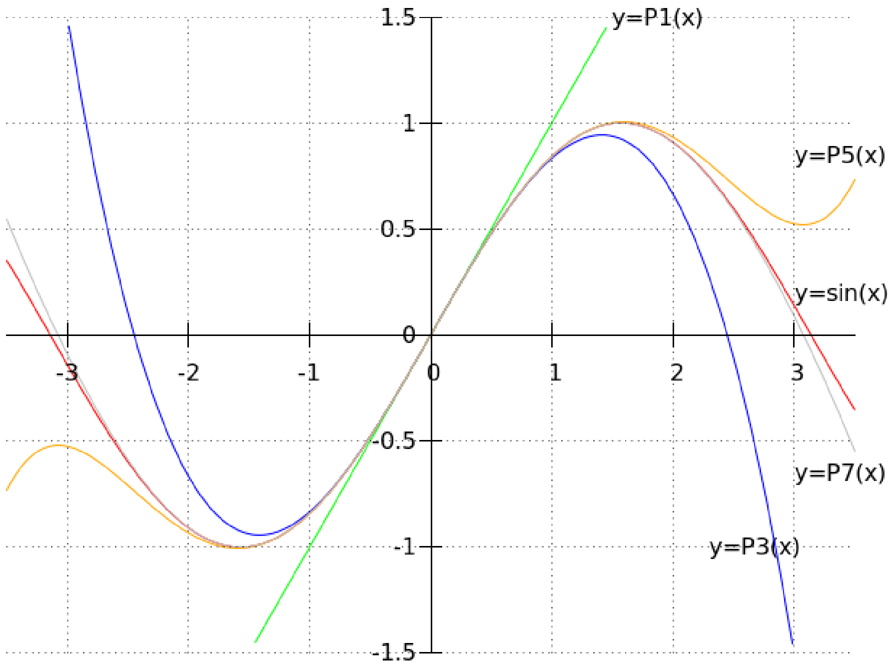


Figura 2: Gráfica de 2.1 en `gnuplot`

2.2. Polinomios de Taylor para $log(1+x)$

Se pidió que graficáramos la aproximación de Taylor para $log(1+x)$.

Código en `Maxima`

```
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);
f(x):=log(1+x);
fortran(T4(x));
fortran(T7(x));
fortran(T11(x));
fortran(T16(x));
text(T4(x));
text(T7(x));
text(T11(x));
text(T16(x));
```

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

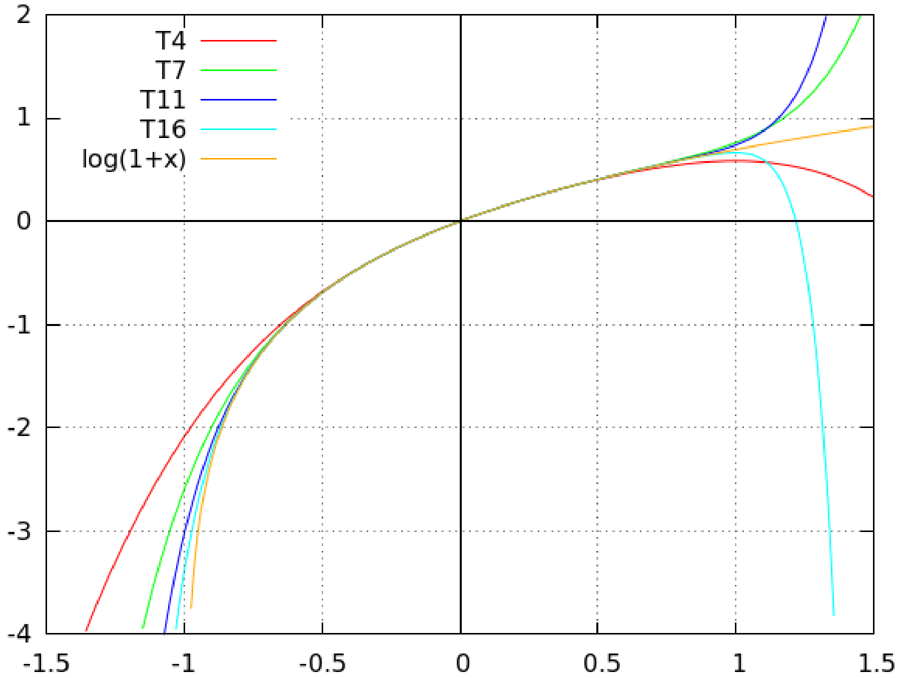


Figura 3: Gráfica de 2.2 en `gnuplot`

2.3. Polinomios de Taylor para $log(cos(x))$

También se presenta la gráfica para $log(cos(x))$.

Código con `Maxima`

```
T3(x):=taylor(f(x), x, 0, 3);
T5(x):=taylor(f(x), x, 0, 5);
T7(x):=taylor(f(x), x, 0, 7);
T9(x):=taylor(f(x), x, 0, 9);
f(x):=log(cos(x));
fortran(T3(x));
fortran(T5(x));
fortran(T7(x));
fortran(T9(x));
text(T3(x));
text(T5(x));
text(T7(x));
text(T9(x));
plot2d ([T3(x),T5(x),T7(x),T9(x),f(x)], [x, -%pi/2, %pi/2], [y, -6, 2],
[grid2d,true],[axes,solid],[color,red,green,blue,cyan,orange],
[legend,"T3","T5","T7","T9","log(cos(x))"],[xlabel,"x"],[ylabel, "y"]);
```

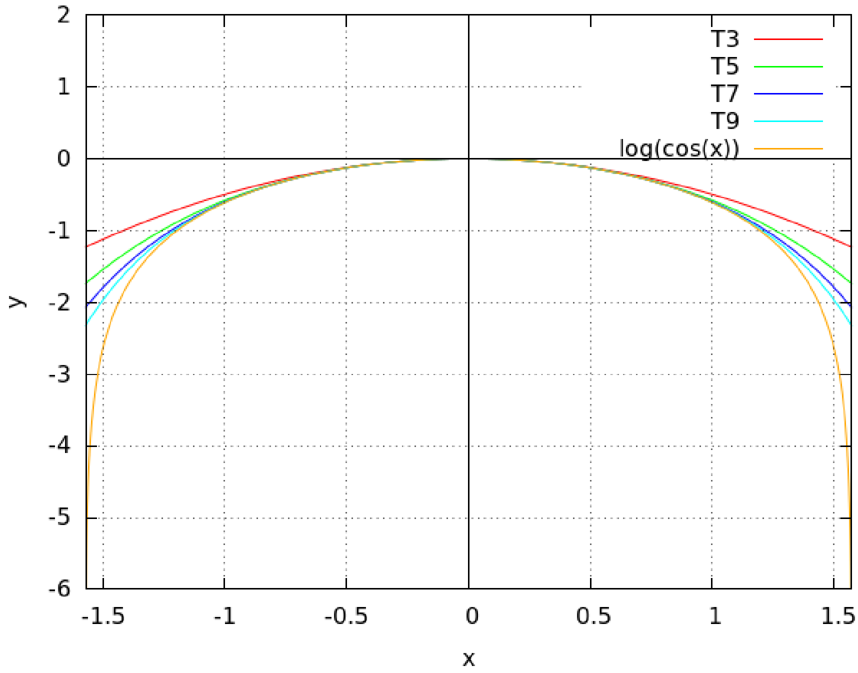


Figura 4: Gráfica de 2.3 en `gnuplot`

2.4. Polinomios de Taylor para $\frac{e^x}{\cos(x)}$

Se obtuvieron resultados de $\frac{e^x}{\cos(x)}$ en las cercanías del 0.

Código con `Maxima`

```
T3(x):=taylor(f(x), x, 0, 3);
T5(x):=taylor(f(x), x, 0, 5);
T7(x):=taylor(f(x), x, 0, 7);
T9(x):=taylor(f(x), x, 0, 9);
f(x):=exp(x)/cos(x);
fortran(T3(x));
fortran(T5(x));
fortran(T7(x));
fortran(T9(x));
text(T3(x));
text(T5(x));
text(T7(x));
text(T9(x));
plot2d ([T3(x),T5(x),T7(x),T9(x),f(x)], [x, -%pi, %pi], [y, -5, 5,
[grid2d,true],[axes,solid],[color,red,green,blue,cyan,orange],
[legend,"T3","T5","T7","T9","exp(x)/cos(x)"],[xlabel,"x"],[ylabel, "y"]);
```

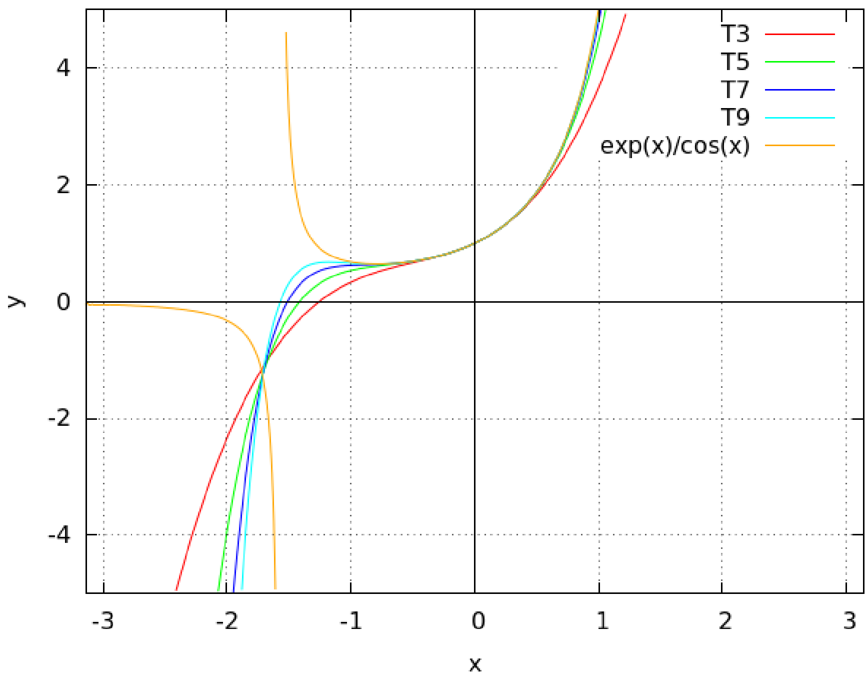


Figura 5: Gráfica de 2.4 en `gnuplot`

2.5. Polinomios de Taylor para $(1 + x)(e^x)$

Por último, tenemos la aproximación de $(1 + x)(e^x)$.

Código con `Maxima`

```
T3(x):=taylor(f(x), x, 0, 3);
T5(x):=taylor(f(x), x, 0, 5);
T7(x):=taylor(f(x), x, 0, 7);
T9(x):=taylor(f(x), x, 0, 9);
f(x):=(1+x)*(exp(x));
fortran(T3(x));
fortran(T5(x));
fortran(T7(x));
fortran(T9(x));
text(T3(x));
text(T5(x));
text(T7(x));
text(T9(x));
plot2d ([T3(x),T5(x),T7(x),T9(x),f(x)], [x, -5, 2], [y, -5, 5],[grid2d,true],
[axes,solid],[color,red,green,blue,cyan,orange],
[gnuplot_preamble, "set key left"],
[legend,"T3","T5","T7","T9","(1+x)*(exp(x))"],[xlabel,"x"],[ylabel, "y"]);
```

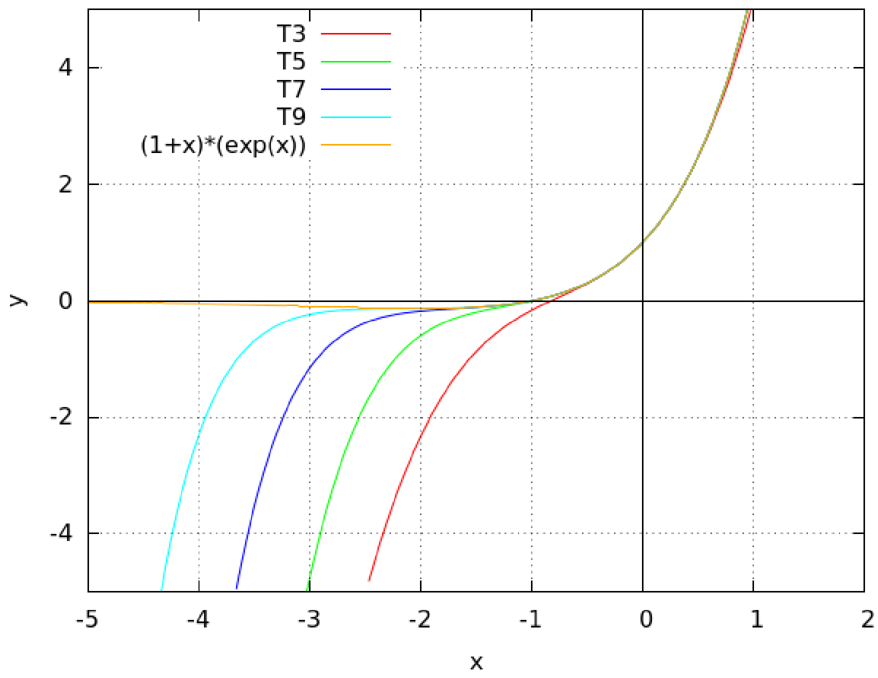


Figura 6: Gráfica de 2.5 en `gnuplot`