1 Sean las funciones

```
define(f(x),sin(x));
define(g(x),cos(x));
```

$$f(x):=\sin(x)$$

$$g(x):=\cos(x)$$

1.1 Polinomios de Taylor centrados en 0, grado 8

```
define(tf(x),ratdisrep(taylor(f(x),x,0,8)));
```

$$tf(x):=-\frac{x^7}{5040}+\frac{x^5}{120}-\frac{x^3}{6}+x$$

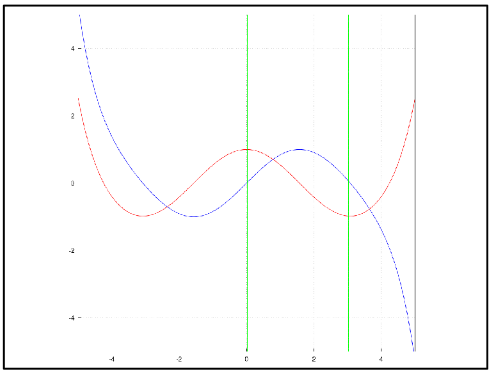
```
define(tg(x),ratdisrep(taylor(g(x),x,0,8)));
```

$$tg(x):=\frac{x^8}{40320}-\frac{x^6}{720}+\frac{x^4}{24}-\frac{x^2}{2}+1$$

1.2 Área de la región delimitada por los polinomios entre 0,3

Primero graficamos los polinomios, también marcamos el intervalo 0,3 con rectas verticales para tener una mejor referencia visual de qué tenemos que calcular:

```
wxdraw2d(
   color=blue,
   explicit(tf(x), x, −5, 5),
   color=red,
   explicit(tg(x), x, −5, 5),
   color=green,
   implicit(x=0+y·0,x,−5, 5,y,−5, 5),
   implicit(x=3+y·0,x,−5, 5,y,−5, 5),
   yrange= [−5,5],
   yaxis=true,
   xaxis=true,
   grid=true,
   proportional_axes=xy
);
```



Como una función no es estrictamente mayor que la otra (se cortan) en el rango dado, debemos encontrar el punto de corte comprendido entre 0,3. Resolvemos la ecuación, y llamamos al punto "a":

```
a:find_root(tf(x)−tg(x),x,0,3);
```

0.7853984010624282

Y ahora aplicamos la fórmula para el cálculo del área: ∫[b,a] abs(g(x) − f(x)) dx, donde g(x) es la función que está por arriba.

Calculamos el área comprendida entre 0, a (tg(x) por arriba):

```
ratprint:false$
A1:abs(integrate(tg(x)−tf(x), x, 0, a));
```

0.4142135886202007

Y ahora el área comprendida entre a,3 (tf(x) por arriba):

```
A2:abs(integrate(tf(x)−tg(x), x, a, 3));
```

2.243677874334487

Finalmente, sumamos las dos áreas:

```
A_final: A1+A2;
```

2.657891462954687

1.3 Spline en 0,1,2,3

```
load(interpol)$
```

Definimos la lista de puntos y el spline

```
puntos:makelist([0+k·5/5, f(0+k·5/5)],k,0,3);
```

$$[[0,0],[1,\sin(1)],[2,\sin(2)],[3,\sin(3)]]$$

```
define(s(x),cspline(puntos));
```

$$s(x):=\left(-\frac{\sin(3)\,x^3}{15}+\frac{2\sin(2)\,x^3}{5}-\frac{3\sin(1)\,x^3}{5}+\frac{\sin(3)\,x}{15}-\frac{2\sin(2)\,x}{5}+\frac{8\sin(1)\,x}{5}\right)\mathrm{charfun}(x<1)+$$
$$\mathrm{charfun}(1\le x \wedge x<2)\left(\frac{\sin(3)\,x^3}{3}-\sin(2)\,x^3+\sin(1)\,x^3-\frac{6\sin(3)\,x^2}{5}+\frac{21\sin(2)\,x^2}{5}-\frac{24\sin(1)\,x^2}{5}+\frac{19\sin(3)\,x}{15}-\frac{23\sin(2)\,x}{5}+\frac{32\sin(1)\,x}{5}-\frac{2\sin(3)}{5}+\frac{7\sin(2)}{5}-\frac{8\sin(1)}{5}\right)+$$
$$\mathrm{charfun}(2\le x)\left(-\frac{4\sin(3)\,x^3}{15}+\frac{3\sin(2)\,x^3}{5}-\frac{2\sin(1)\,x^3}{5}+\right.$$

$$\frac{12\sin(3)x^2}{5}-\frac{27\sin(2)x^2}{5}+\frac{18\sin(1)x^2}{5}-\frac{89\sin(3)x}{15}+\frac{73\sin(2)x}{5}-\frac{52\sin(1)x}{5}+\frac{22\sin(3)}{5}-\frac{57\sin(2)}{5}+\frac{48\sin(1)}{5})$$

Y comprobamos el error absoluto:

```
is(f(1.5)−tf(1.5)<f(1.5)−s(1.5));
```

true

El error absoluto en 1.5 es menor en el polinomio de Taylor que en el Spline.

2 Sea la ecuación:

```
define(ec(x),%e^(x^2+x−1) − (%e^(x^2)+2));
```

$$ec(x):=\%e^{x^2+x-1}-\%e^{x^2}-2$$

2.1 Calcular con NR la solución e iteraciones necesarias en [1,2] con exactitud 10^{-6}

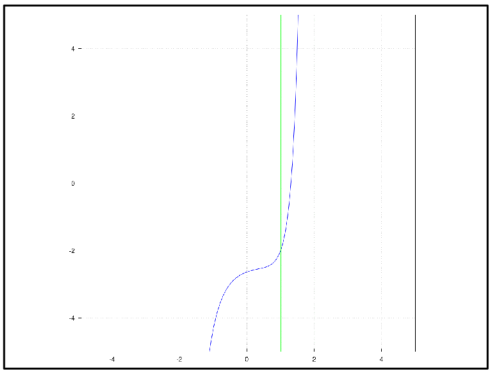
Definimos nuestro bloque con la función NR:

```
nr(expr,var,ini,err):=block(
   [x0:ini, x1, dfx0, j, tol:10^(−10)],
   numer: true,
   local(f,df),
   define(f(x),subst(x,var,expr)),
   define(df(x),diff(f(x),x)),

   for i:1 thru 25 do(
      j:i,
      dfx0:df(x0),
      if abs(df(x0))<err then
         error("escoja otro valor inicial"),
      x1:x0−f(x0)/dfx0,
      if abs(x0−x1)<tol then return(),
      x0:x1
   ),
   if j=25 then error("escoja otro valor inicial") else [x1,j]
)$
```

Graficamos la función para buscar un buen punto de inicio:

```
wxdraw2d(
   color=blue,
   explicit(ec(x), x, −5, 5),
   color=green,
   implicit(x=1+y·0,x,−5, 5,y,−5, 5),
   implicit(x=2+y·0,x,−5, 5,y,−5, 5),
   yrange= [−5,5],
   yaxis=true,
   xaxis=true,
   grid=true,
   proportional_axes=xy
);
```



Vemos que el punto 1 está tocando el eje OX, así que lo mejor será alejarnos un poco. Vamos a probar a aplicar NR empezando en 1.5:

```
nr(ec(x),x,1.5,10^−6);
```

[1.308311431058462,6]

Podemos comprobar la solución y el error absoluto resolviendo la ecuación:

```
sol:find_root(ec(x),x,0,2);
```

1.308311431058462

```
is(sol−1.308311431058462<10^−6);
```

true