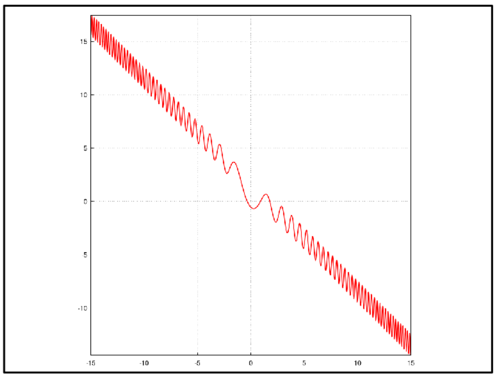
Primer parcial wxMaxima A1 [redacted]

1 [redacted]oluciones de la ecuación $x=arctan(x^2)-cos(x^2+1)$

Como se trata de una función trigonométrica, la resolveremos con find_root. Para ello, primero vamos a graficarla.

```
wxdraw2d(
   color=red,
   line_width=2,
   explicit(atan(x^2)−cos(x^2+1)−x, x, −15, 15),
   yaxis=true,
   xaxis=true,
   proportional_axes=xy,
   grid=true
);
```



Como vemos, la función hace cero en dos puntos, por lo que si la analizamos con find_root tendremos que proporcionarle dos rangos con f(x) de signos distintos. Si no, sólo obtendríamos una de las soluciones.
Esos rangos pueden ser, por ejemplo, [-2,0] y [0,1].

```
find_root(atan(x^2)−cos(x^2+1)−x, x, −2, 0);
```

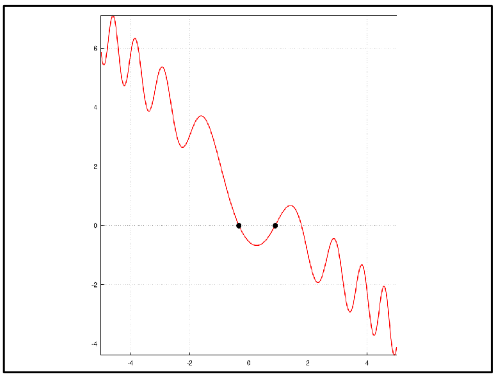
-0.3331894622672368

```
find_root(atan(x^2)−cos(x^2+1)−x, x, 0, 1);
```

0.8903239877394064

Finalmente, colocamos los puntos en la gráfica para comprobar que es correcto:

```
wxdraw2d(
   color=red,
   line_width=2,
   explicit(atan(x²)−cos(x²+1)−x, x, −5, 5),
   color=black,
   point_type=7,
   point_size=2,
   points([[−0.3331894622672368,0],
            [0.8903239877394064,0]]),
   yaxis=true,
   xaxis=true,
   proportional_axes=xy,
   grid=true
);
```



2 Define una función que calcule la suma de los n primeros múltiplos de 5. ¿Cuál es el primer natural para el que dicha suma pasa de 10.000?

Definimos la función. Esta recibe n como parámetro y hace n iteraciones con un bucle for que calcula la suma acumulativa multiplicando el iterador (i) por 5. Devuelve la suma acumulada.

```
suma_multiplos5(n):=block(
   suma:0,
   for i:1 thru n do(
      suma: suma + (i·5)
   ),
   suma
);
```

$$\text{suma\_multiplos5}(n):= \text{block}(suma:0,\text{for } i \text{ thru } n \text{ do } suma:suma+i\,5, suma)$$

Primero un ejemplo fácil de comprobar, para ver si funciona correctamente: los tres primeros múltiplos de 5 son 5, 10, 15; que suman 30.

```
suma_multiplos5(3);
```

30

La función es correcta. Ahora veamos el primer natural cuya suma acumulada es mayor que 10.000:

```
valor:0$
while suma_multiplos5(valor) < 10000 do(
   valor: valor + 1
)$
valor;
suma_multiplos5(valor);
```

63

10080

El primer n cuya suma de los n primeros múltiplos de 5 es mayor que 10000 es 63, cuya suma da 10080.

3 Calcula los puntos de corte de la recta 3x+sqrt(2)y=1 y la elipse centrada en el origen de semiejes 2sqrt(2) y sqrt(5). Representa gráficamente la recta, la elipse, y los puntos de corte.

Para encontrar los puntos de corte, resolvemos el sistema de ecuaciones de las dos funciones. La función de una elipse es $x^2/a^2 + y^2/b^2=1$, donde a y b son los semiejes (2*√2 y √5).

Por tanto, la ecuación de la elipse (simplificada) queda $x^2/8 + y^2/5 =1$.

```
to_poly_solve([x^2/8 + y^2/5 =1, 3·x+sqrt(2)·y=1],[x,y]);
```

WARNING: redefining MAXIMA::OPAPPLY in DEFMACRO
WARNING: redefining MAXIMA::OPCONS in DEFMACRO

$$\%union\left(\left[x=-\frac{18\sqrt{5}-12}{41}, y=\frac{27\,2^{3/2}\sqrt{5}+5\sqrt{2}}{82}\right], \left[x=\frac{18\sqrt{5}+12}{41}, y=-\frac{27\,2^{3/2}\sqrt{5}-5\sqrt{2}}{82}\right]\right)$$

Tomamos los puntos de corte obtenidos y graficamos todo lo que pide el ejercicio:

```
wxdraw2d(
   color=red,
   line_width=2,
   implicit(x²/(2·sqrt(2))² + y²/(sqrt(5))²=1, x, −5, 5, y, −5, 5),
   color=blue,
   implicit(3·x+sqrt(2)·y=1, x, −5, 5, y, −5, 5),
   color=black,
   point_type=7,
   point_size=2,
   points([[−(18·sqrt(5)−12)/41, (27·2^(3/2)·sqrt(5)+5·sqrt(2))/82],
            [(18·sqrt(5)+12)/41, −(27·2^(3/2)·sqrt(5)−5·sqrt(2))/82]]),
   yaxis=true,
   xaxis=true,
   proportional_axes=xy,
   grid=true
);
```

rat: replaced 1.414213562373095 by 22619537/15994428 = 1.414213562373096