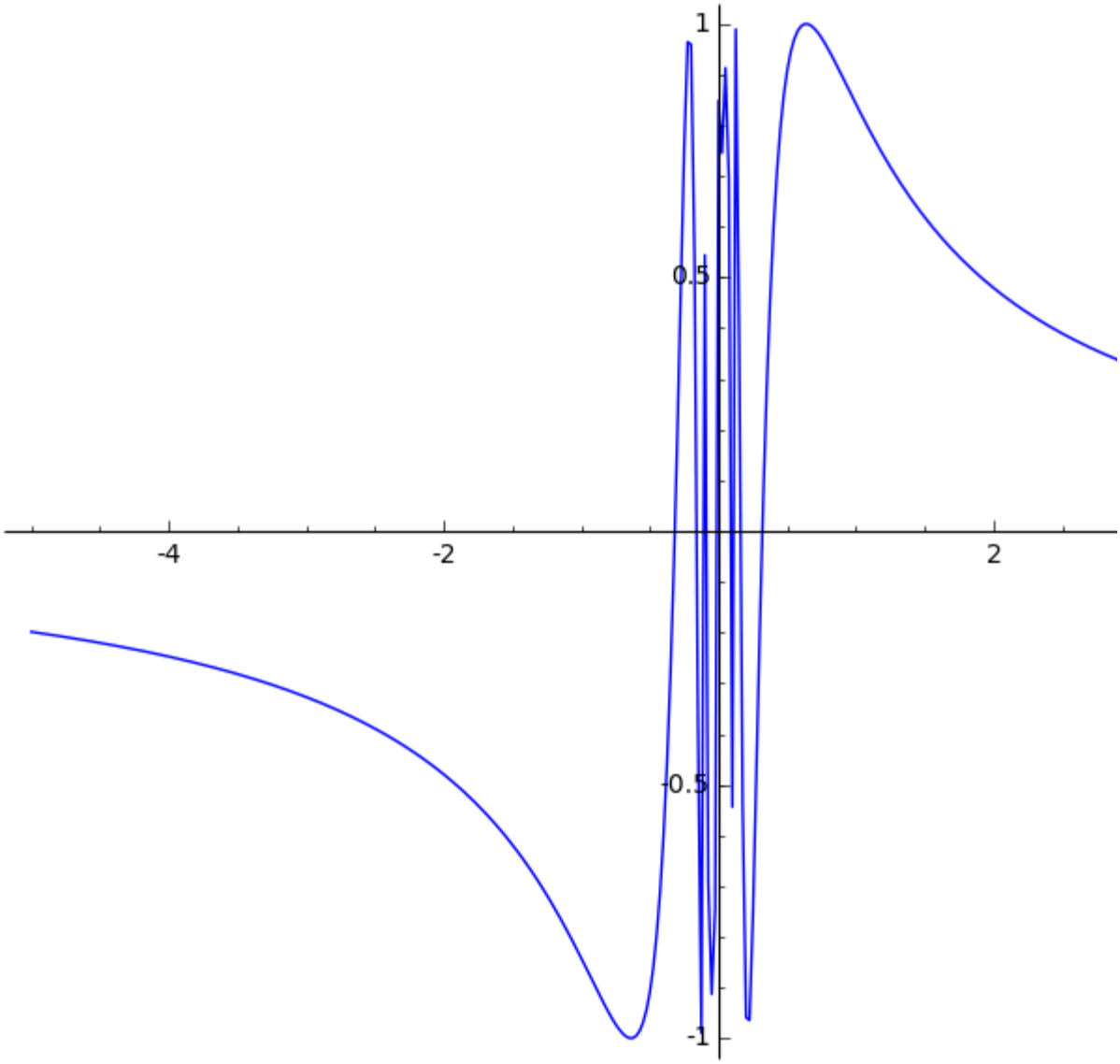
plot-libgiac

Very Basic conversion of graphics is supported from giac to sage.

The lines, circles, ... conics are not implemented.

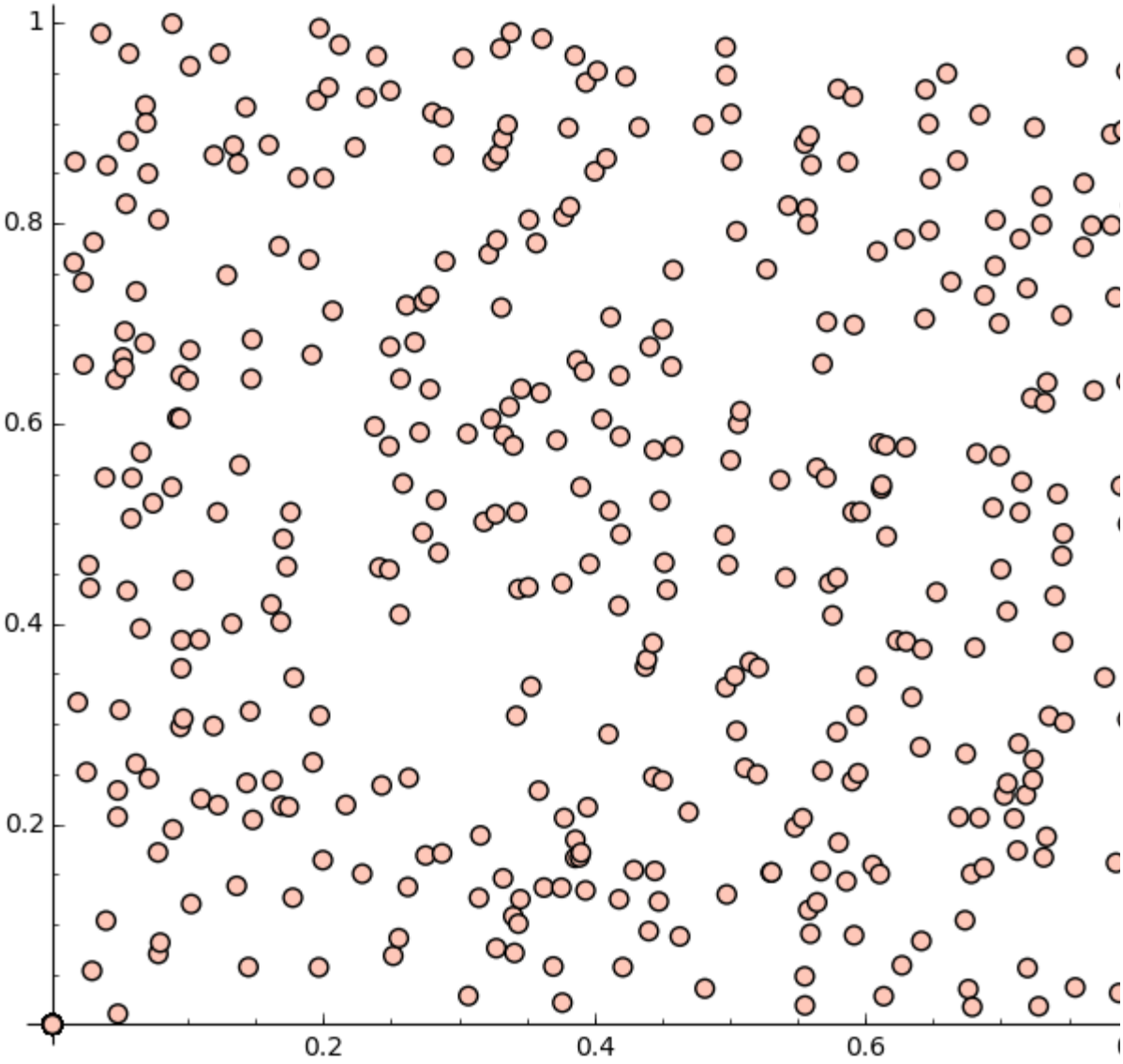
```
from giacpy import *;
x,y,z=libgiac('x,y,z');
f=sin(1/x);
f.plot().mplot()
```

```
// Giac share
root-directory:/home/fred/dev/sage/git-trac-command/local/share
/
// Using keyword file
/home/fred/dev/sage/git-trac-command/local/share/giac/doc/fr/ke
s
// Giac share
root-directory:/home/fred/dev/sage/git-trac-command/local/share
/
Help file
/home/fred/dev/sage/git-trac-command/local/share/giac/doc/fr/ai
s not found
Added 0 synonyms
```



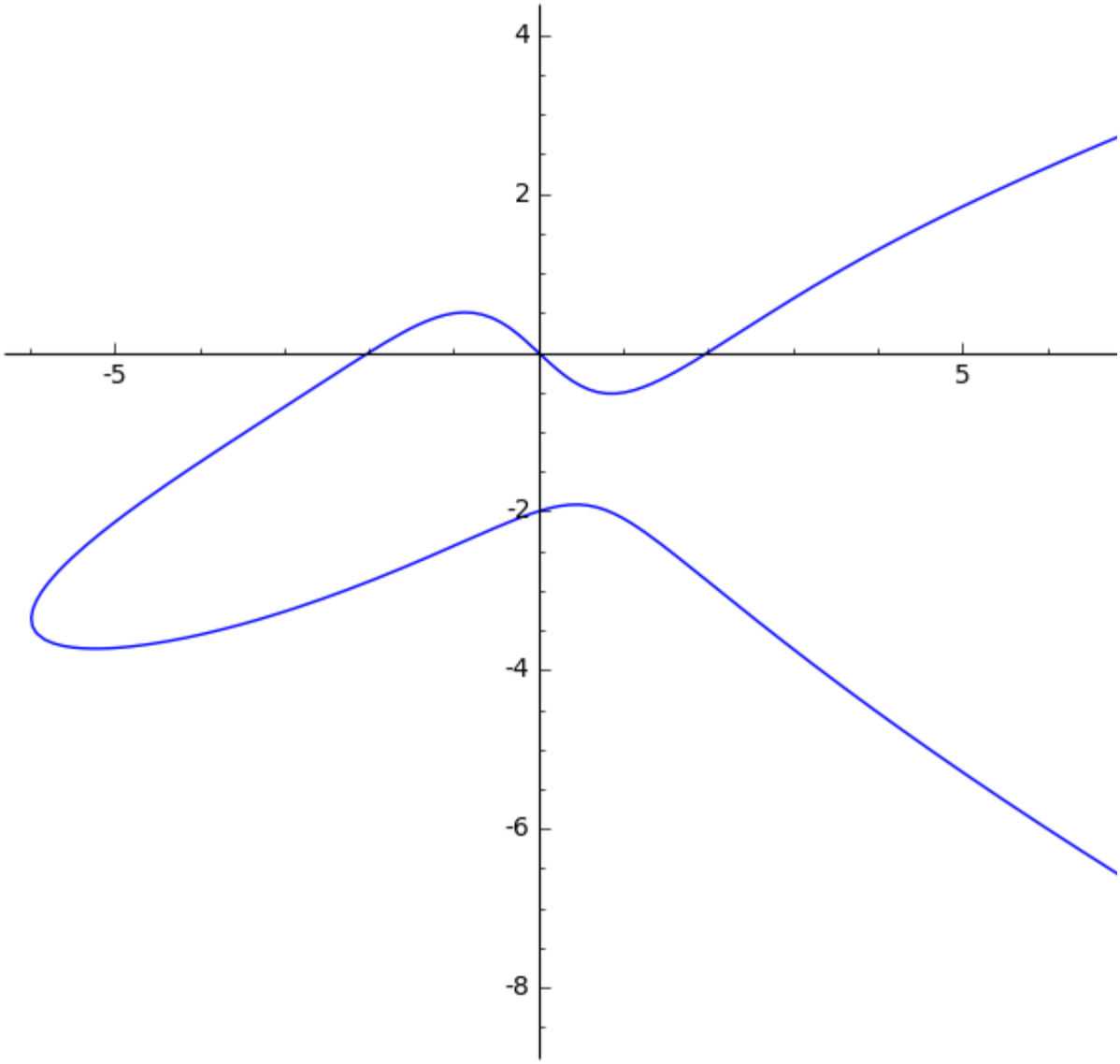

```
M=libgiac.ranm(500,2,'0..1')
#p1=plotlist(M)
p2=libgiac.scatterplot(M)
```

```
p2.mplot()
```




```
x,y=libgiac('x,y');
d=(x^4+y^3+2*x*y^2-y+x).plotimplicit('x=-5..5,y=-5..5');
```

```
(d*i*2).mplot()
```



Giac and sage 5.10 answers are different on this example of integration.

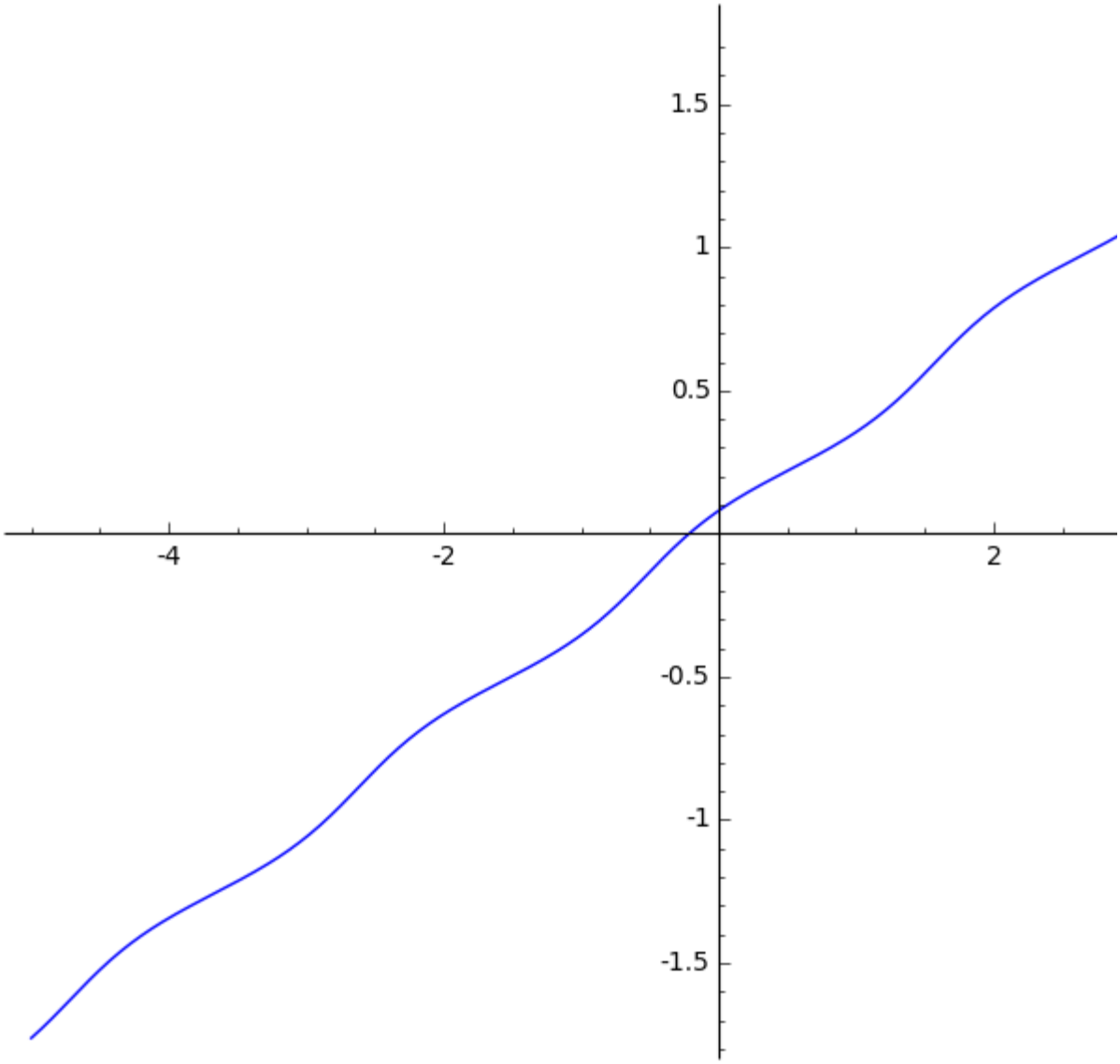
Giac answer is more complicated, but the constant is correct, indeed, the primitive must be continuous.

```
x=var('x');f=1/(3+sin(3*x))
```

```
I1=(libgiac(f)).int();I1   # primitive with giac
```

```
2/3*2/(sqrt(2)*4)*(atan((3*tan(3*x/2)+1)/(2*sqrt(2)))+pi*floor(
/pi+1/2))
```

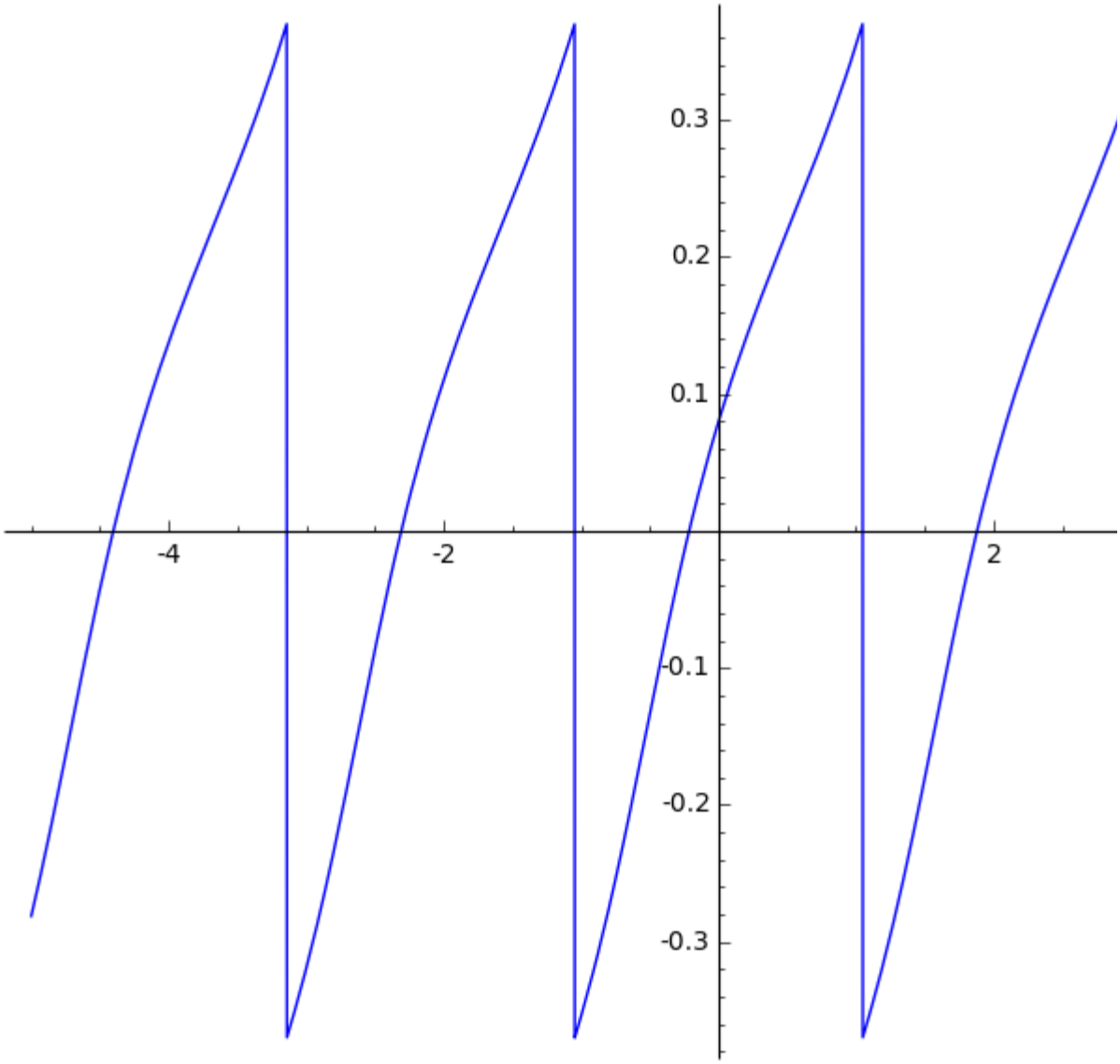
```
I1.plot().mplot()
```




```
I2=integral(f,x);I2   # the primitive with sage 6.3beta
```

```
    1/6*sqrt(2)*arctan(1/4*sqrt(2)*(3*sin(3*x)/(cos(3*x) + 1) + 1))
```

```
plot(I2,x,-5,5)
```


0.3
0.2
0.1
-0.1
-0.2
-0.3
-4
-2
2