efarmoges_sta_mathimatika

Τυχαίοι αριθμοί

```
random()
```

```
0.3675276790228389
```

```
[random() for i in range(10)]
```

```
[0.11050328624094419,
 0.3565093135705997,
 0.015316127115346467,
 0.04269228263803637,
 0.3816442062718678,
 0.28519852876845597,
 0.5199427440819391,
 0.30993137310732854,
 0.034924334590274886,
 0.007696019131140153]
```

```
randint(50,80)
```

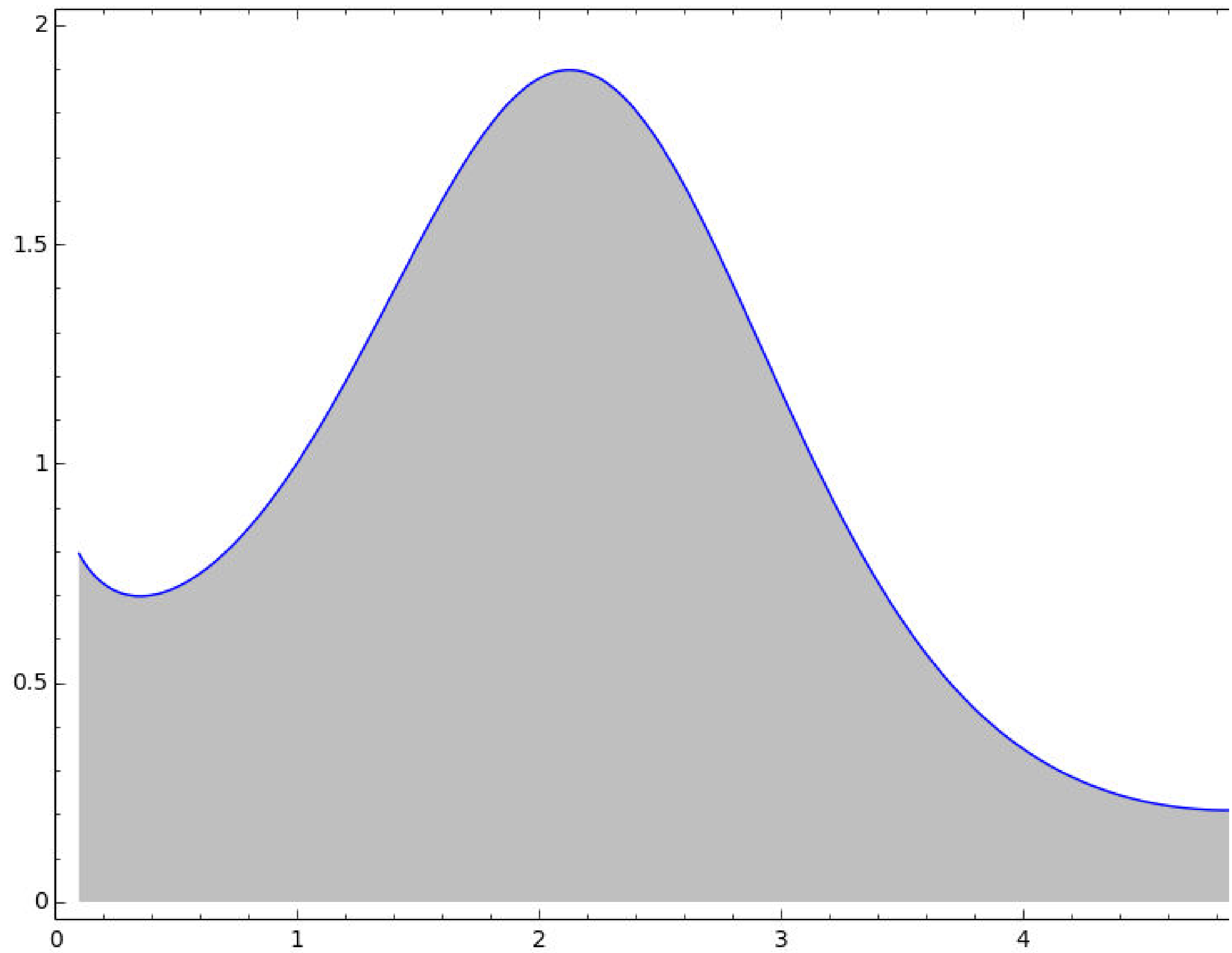
```
74
```

```
[randint(10,20) for i in range(10)]
```

```
[16, 12, 15, 10, 16, 13, 17, 14, 20, 14]
```

Υπολογισμός Ολοκληρώματος

```
f(x)=x^sin(x)
plot(f,x,0.1,5,frame=True,axes=False,fill=True,xmin=0.1,xmax=5.,ymin=0.,ymax=2.)
```




```
f(x)=x^sin(x);
pin=0
nop=20000
x1=0.1;x2=5.;y1=0.;y2=2.;
for i in range(nop):
    xx=x1+(x2-x1)*random()
    yy=y1+(y2-y1)*random()
    if yy<f(xx):
        pin+=1
print (x2-x1)*(y2-y1)*float(pin/nop)
```

4.71380000000000

```
numerical_integral(x^sin(x),0.1,5.)
```

(4.773813346559158, 3.0854447345393065e-13)

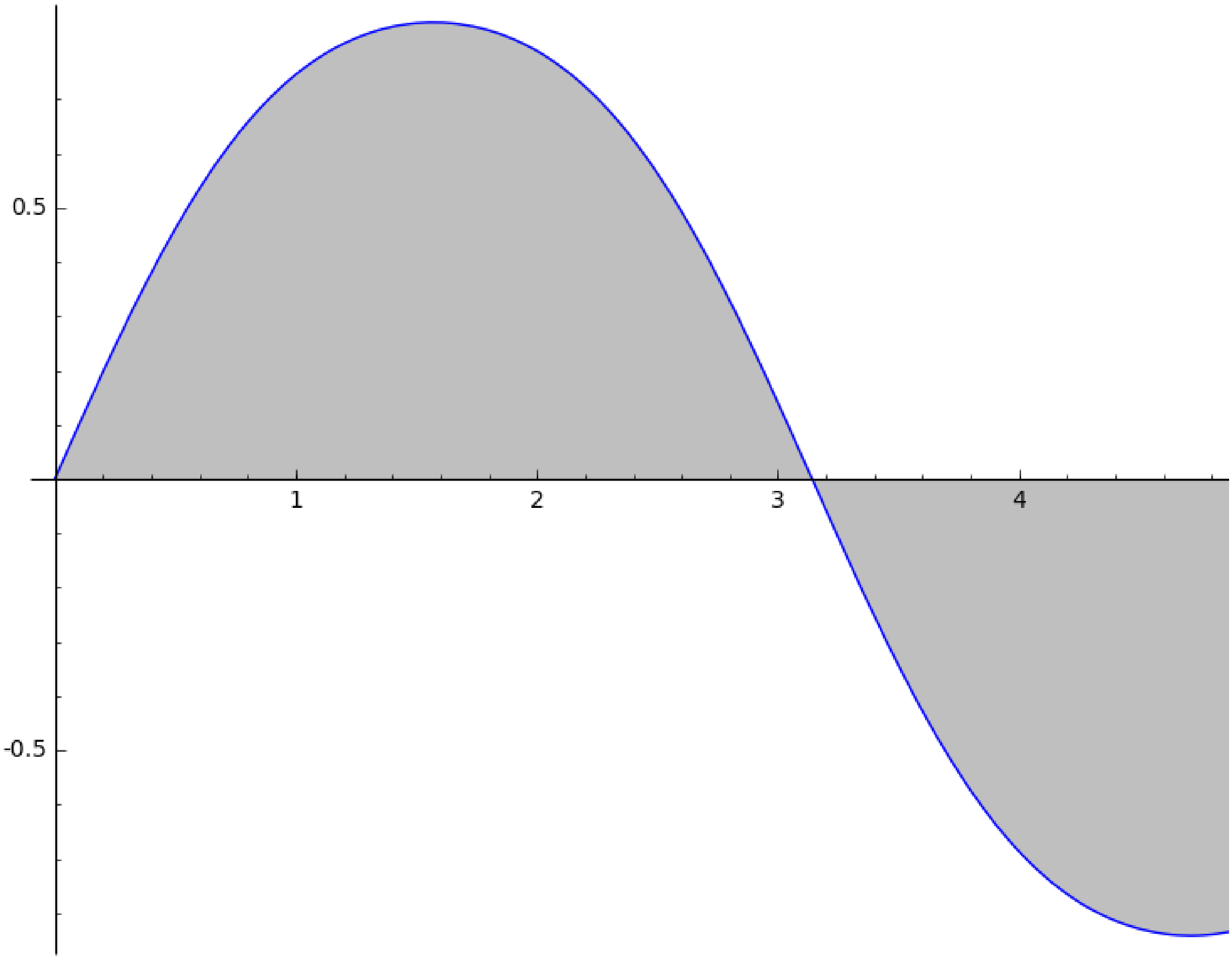
```
f(x)=sin(sin(x))
nop=200000
sf=0.
x1=0.;x2=5.;
for i in range(nop):
    xx=x1+(x2-x1)*random()
    sf+=f(xx)
fm=sf/float(nop)
print fm*(x2-x1)
```

0.646797054094543

```
numerical_integral(f(x),x1,x2)
```

$\left(0.653401711337, 3.23916970455 \times 10^{-14}\right)$

```
plot(sin(sin(x)),x,0,5,fill=True)
```

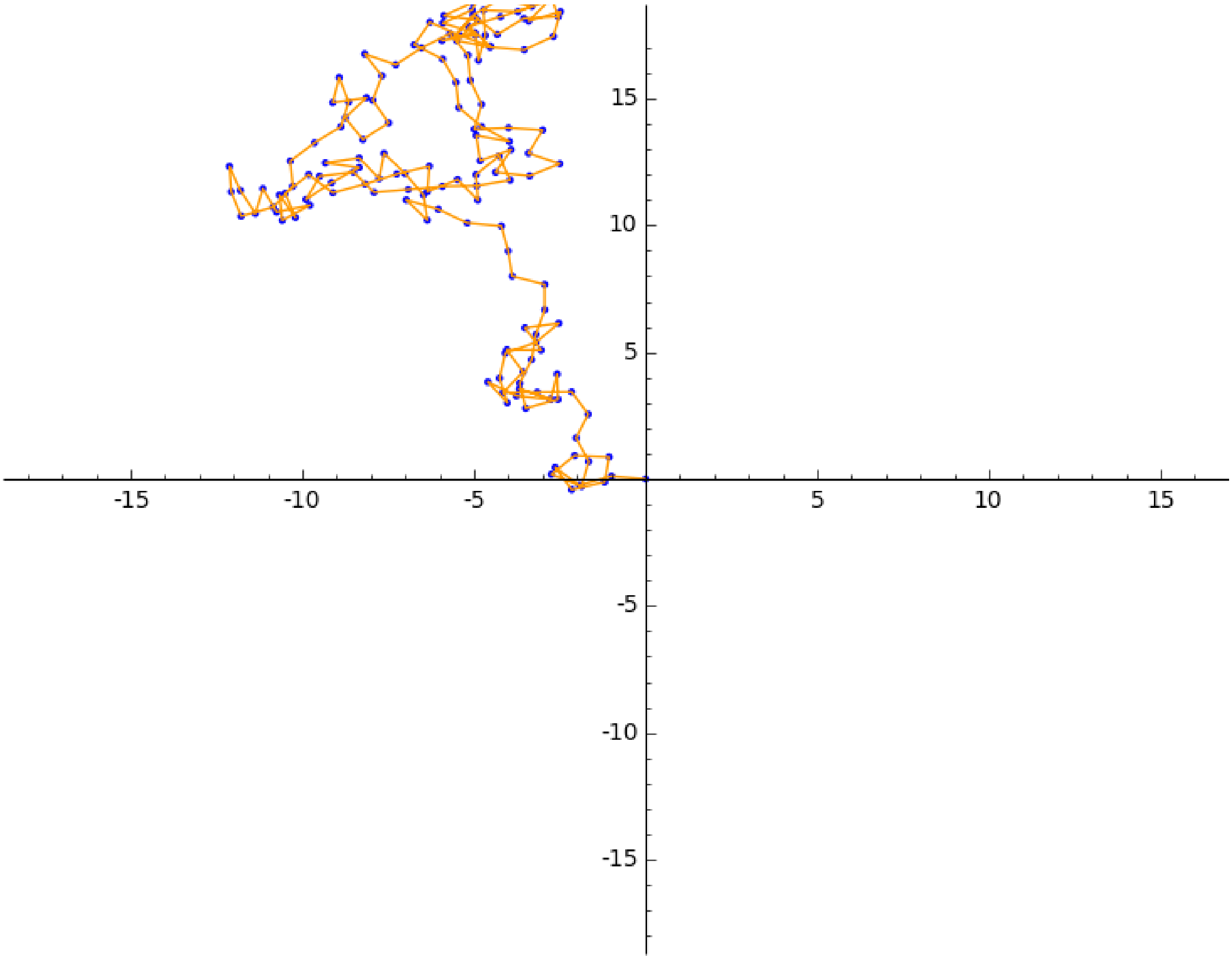


Τυχαίος Περίπατος

```
x=0;y=0
nop=900
data=[[x,y]]
for i in range(nop):
    x+=-1.0+2.0*random()
    y+=-1.0+2.0*random()
    data.append([x,y])
```

```
x=0;y=0
nop=300;l=1
data=[[x,y]]
for i in range(nop):
    th=n(2.0*pi*random())
    x,y=x+l*cos(th),y+l*sin(th)
    data.append([x,y])
```

```
g=line(data,hue=0.1)+points(data);
mm=int(sqrt(nop))+1
g.show(xmin=-mm,xmax=mm,ymin=-mm,ymax=mm)
```

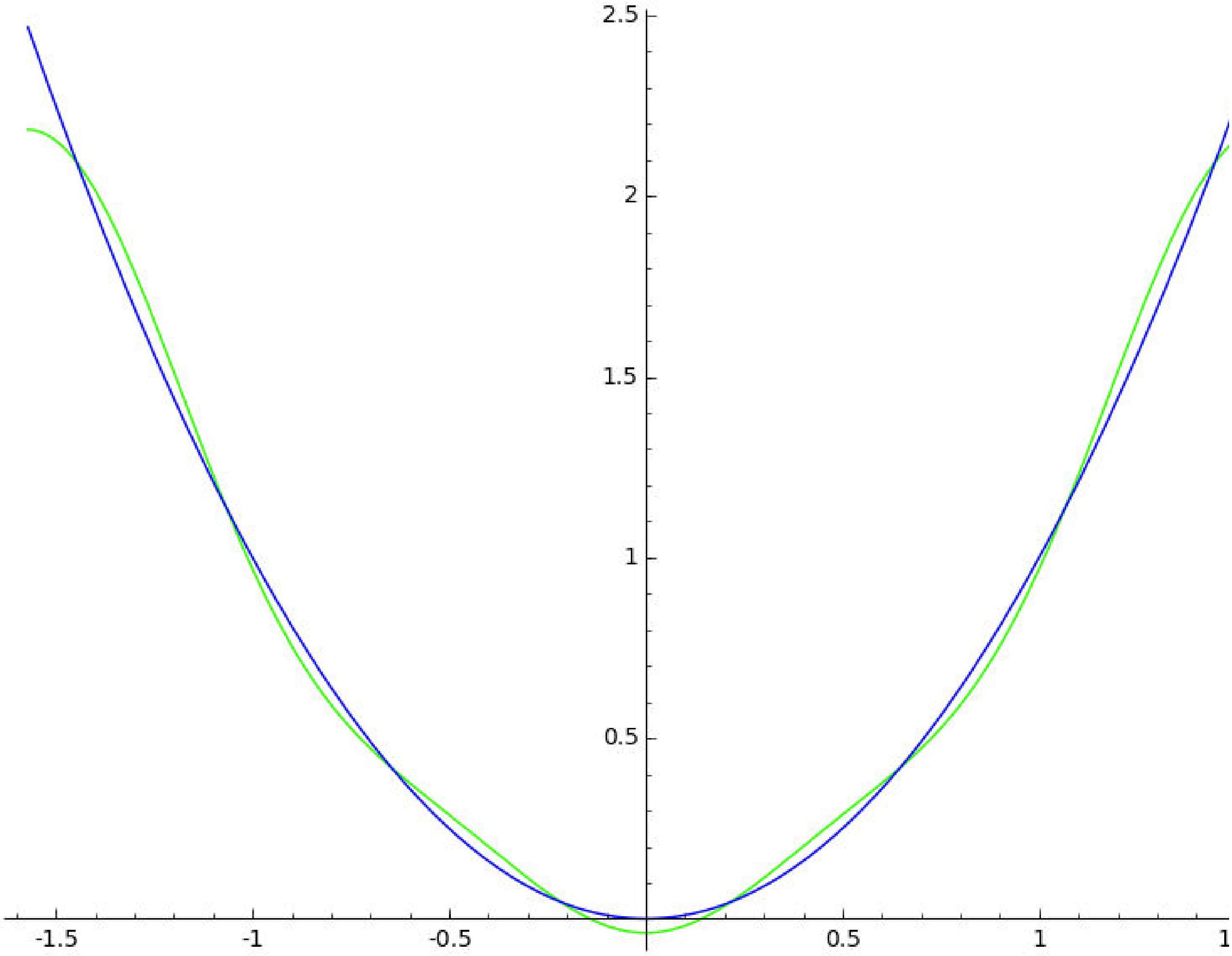


Ανάπτυγμα Fourier περιοδικής συνάρτησης

```
f(x)=x^2
fp = Piecewise([[(-pi/2,pi/2),f]])
ft=fp.fourier_series_partial_sum(4,pi/2);ft
```

    1/12*pi^2 - 1/9*cos(6*x) + 1/4*cos(4*x) - cos(2*x)

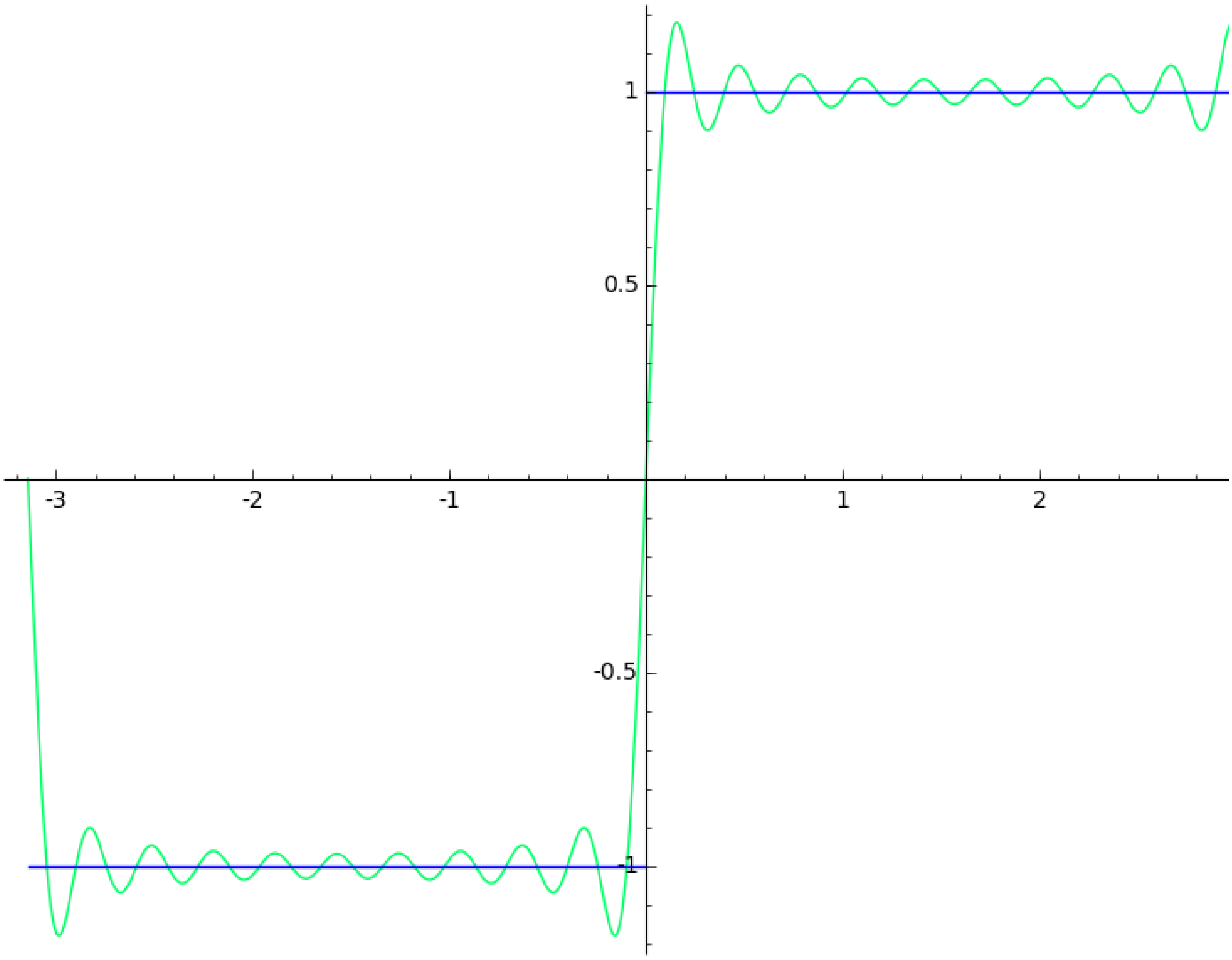
```
plot(ft,x,-pi/2,pi/2,hue=0.3)+fp.plot()
```




```
x=var('x');
f1(x)=-1.;f2(x)=1.
f =  Piecewise([[(-pi,0),f1],[(0,pi),f2]])
ft=f.fourier_series_partial_sum(20,pi);ft
```

```
0.210526315789474*sin(19*x)/pi + 0.235294117647059*sin(17*x)/pi +
0.266666666666667*sin(15*x)/pi + 0.307692307692308*sin(13*x)/pi +
0.363636363636364*sin(11*x)/pi + 0.444444444444444*sin(9*x)/pi +
0.571428571428571*sin(7*x)/pi + 0.8*sin(5*x)/pi +
1.33333333333333*sin(3*x)/pi + 4.0*sin(x)/pi
```

```
plot(ft,x,-pi,pi,hue=0.4)+f.plot()
```



Το πρόβλημα των γενεθλίων

```
def co(nn):
    s=0
    m=[randint(1,365) for i in range(nn)];
    p=[m.count(i) for i in m ]
    k=nn-p.count(1)
    if k>0:
        s+=1
    return s
```

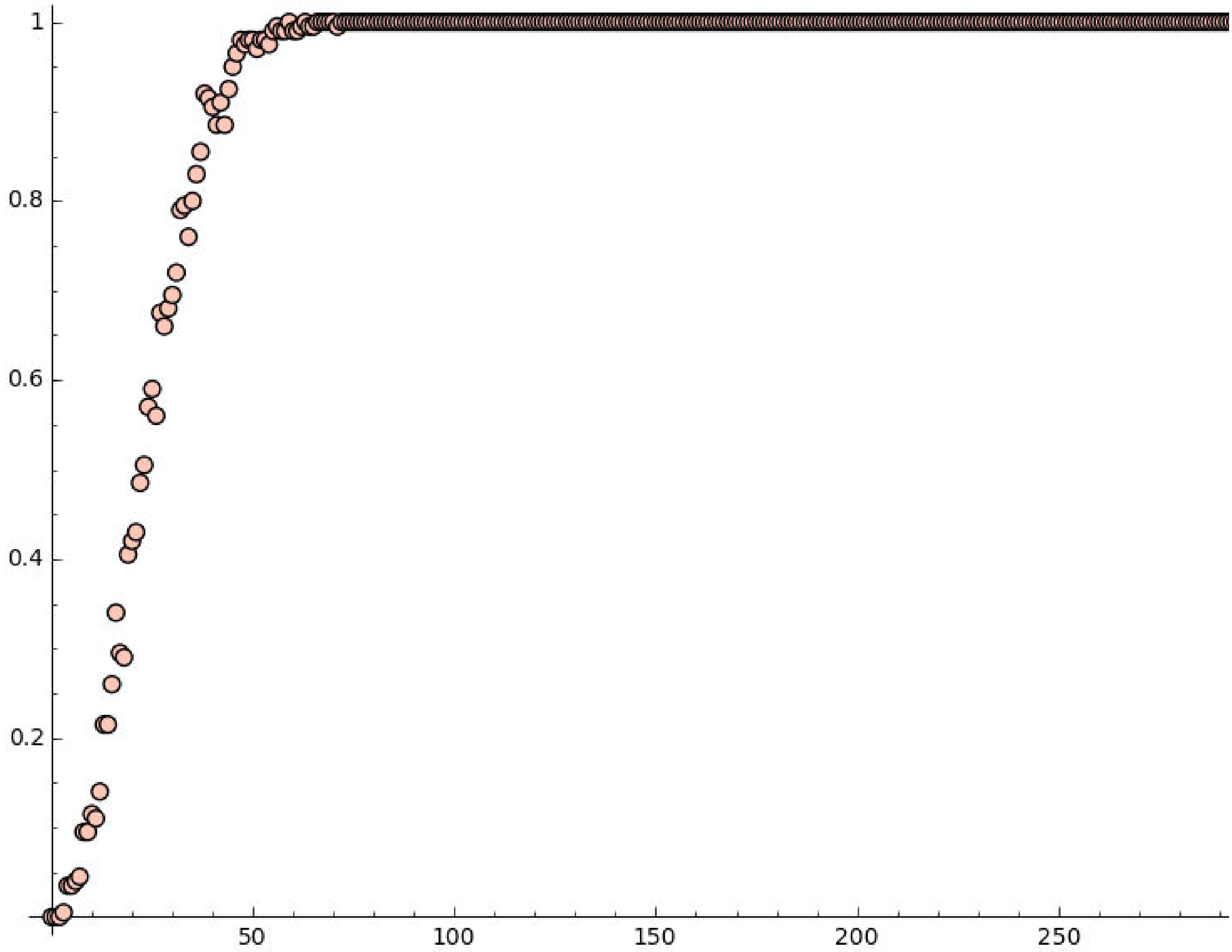
```
def pithan(kk):
    nop=200;
    cc=[co(kk) for ii in range(nop)].count(1)
    pit=float(cc/nop)
    return pit
```

```
pithan(40)
```

    0.895

```
tt=[ [oo,pithan(oo)] for oo in range(0,100)];
```

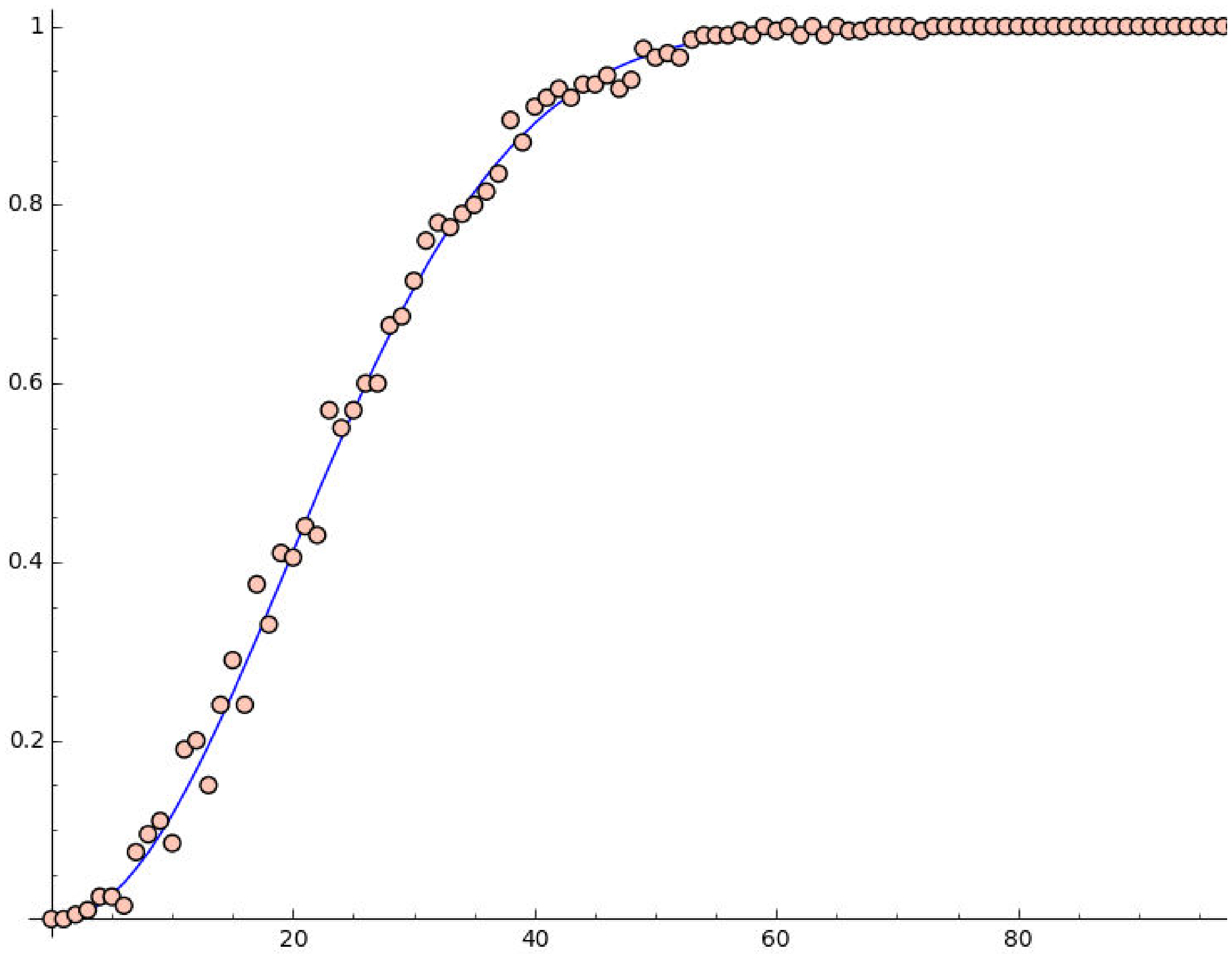
```
scatter_plot(tt)
```




```
theory(n)=1-factorial(n)*binomial(365,n)/365^n
```

```
the=[ [oo,N(theory(oo))] for oo in range(0,100)];
```

```
line(the)+scatter_plot(tt)
```



Αριθμητική εύρεση ρίζας εξίσωσης με τη μέθοδο Newton–Raphson

```
f(x)=-exp(cos(x))+2*x^2
df=diff(f,x)
acc=10^-9
x0=-50.
x=x0
mstep=100
istep=0
for istep in range(mstep):
    x= x-f(x)/df(x)
    print istep+1,x , f(x)
    if abs(f(x))<acc:
        break
```

```
1 -24.9267895484845 1240.02823577110
2 -12.4218277977656 305.913529175461
3 -6.21666402618705 74.5815468477795
4 -3.19550045175173 20.0540320360537
5 -1.62413274127288 4.32752933339756
6 -1.04273093093250 0.519472648979054
7 -0.949977469622775 0.0158345741002841
8 -0.946964316999087 0.0000175229240304553
9 -0.946960975172041 2.15887308030460e-11
```

Πρώτοι Αριθμοί

```
for p in range(0,100):
    if is_prime(p):
        print p
```

```
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
53
59
61
67
71
73
79
83
89
97
```

```
prime_range(0,100)
```

$[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]$

```
prime_range(1821,1900)
```

$[1823, 1831, 1847, 1861, 1867, 1871, 1873, 1877, 1879, 1889]$

```
primes_first_n(20)
```

$[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71]$

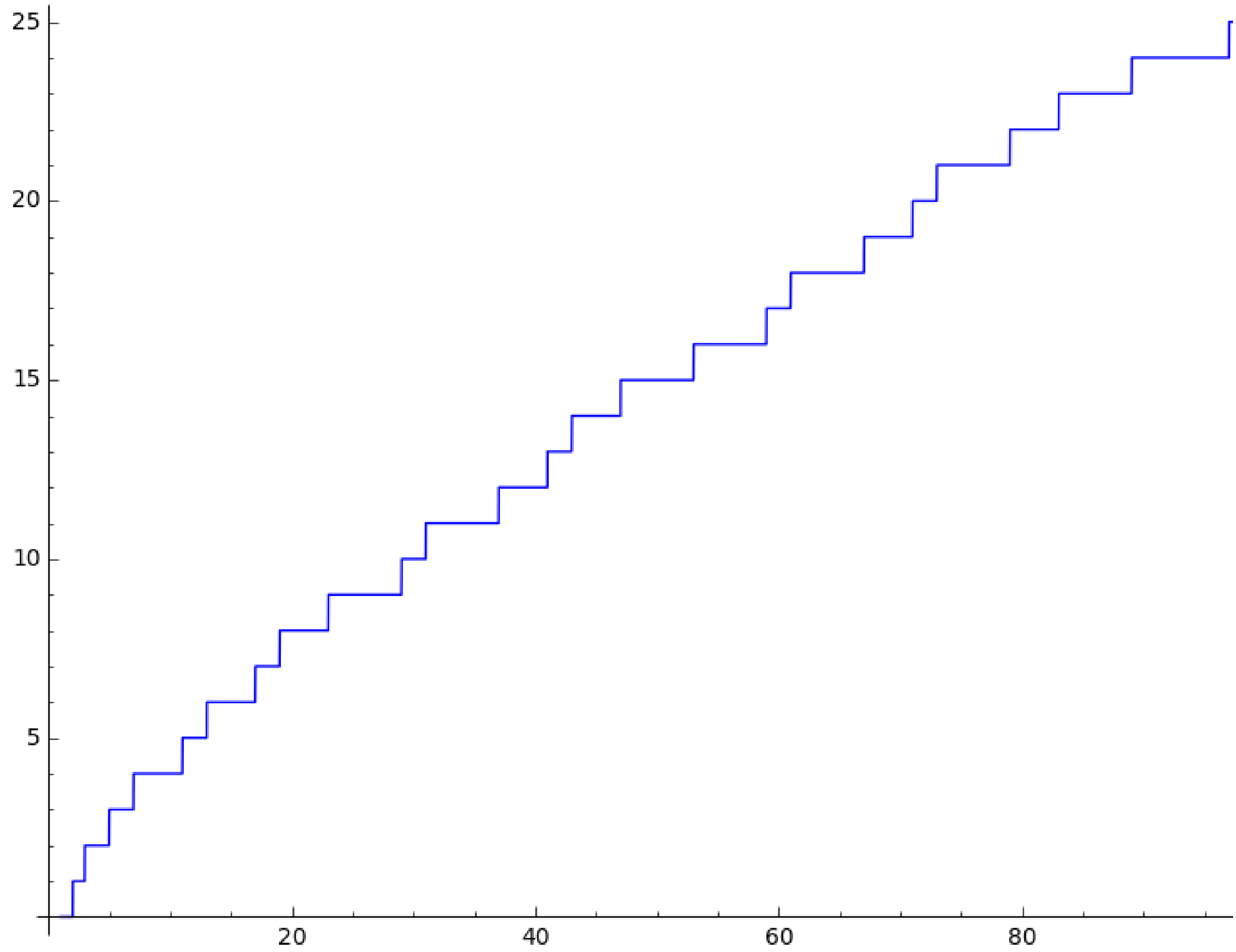
```
prime_pi(100)
```

25

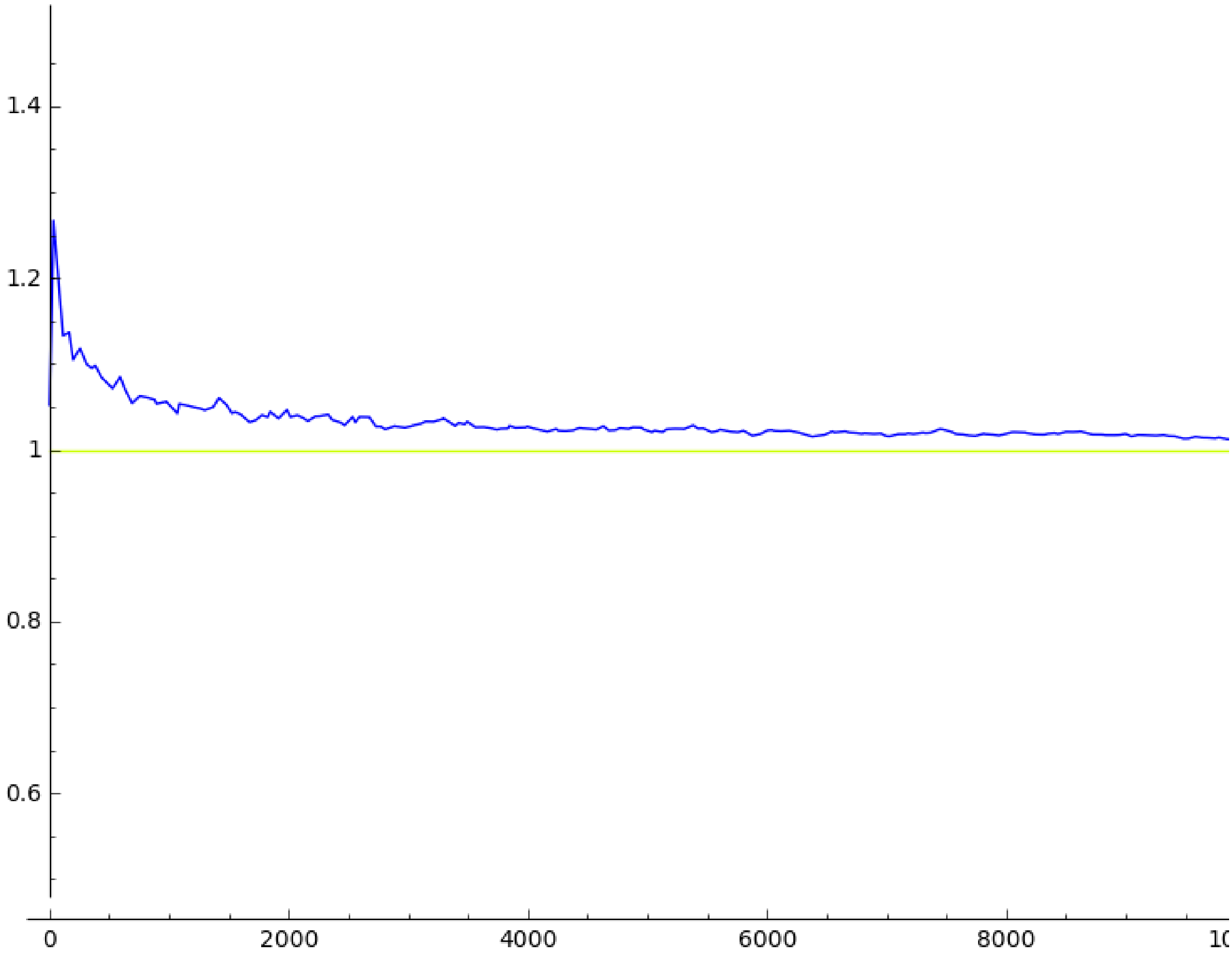
```
prime_pi(10^6)
```

78498

```
plot(prime_pi(x),x,1,100)
```




```
plot(log_integral(x)/prime_pi(x),x,1,10000,\
    ymin=0.5,ymax=1.5)+ plot(1,x,0,10000,hue=0.2)
```




```
for p in primes(1000,1100):
    print p, sum(map(int,str(p)))
```

```
1009 10
1013 5
1019 11
1021 4
1031 5
1033 7
1039 13
1049 14
1051 7
1061 8
1063 10
1069 16
1087 16
1091 11
1093 13
1097 17
```

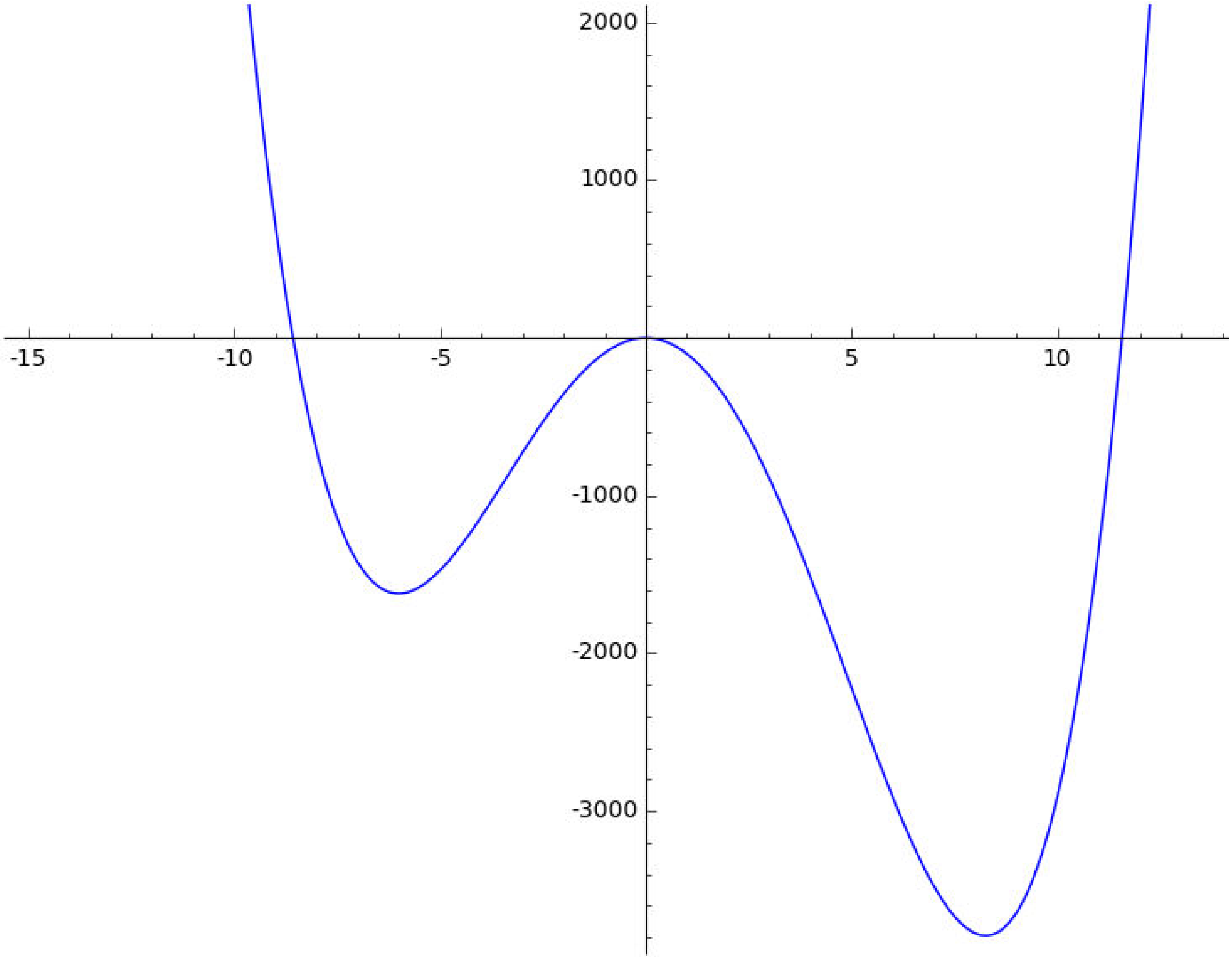
```
lps=[];
for p in primes(0,1000):
    ps=sum(map(int,str(p)))
    if ps==20:
        lps.append(p)
```

```
lps
```

$[389, 479, 569, 587, 659, 677, 839, 857, 929, 947, 983]$

Ακρότατα συναρτήσεων

```
x,y=var('x y');
f=x^4-3*x^3-99*x^2-1
plot(f,x,-15,15,ymax=2000)
```




```
fx=diff(f,x);
akro=solve(fx==0,x,solution_dict=True);
akro
```

$\left[\left\{x : \frac{33}{4}\right\}, \{x : -6\}, \{x : 0\}\right]$

```
fxx=diff(f,x,2);
fxx
```

$12\,x^2 - 18\,x - 198$

```
[fxx.subs(a) for a in akro]
```

$\left[\frac{1881}{4}, 342, -198\right]$

```
minimize(f,[2])
```

```
Optimization terminated successfully.
         Current function value: -3791.230469
         Iterations: 4
         Function evaluations: 10
         Gradient evaluations: 10
```

(8.25000000005)

```
minimize_constrained(f,[(0,15)],[2])
```

(8.25)

```
minimize_constrained(f,[(-15,0)],[-2])
```

(-6.0000000001)

```
x,y=var('x y');
f=100*(y-x^2)^2+(1-x)^2
```

```
fx=diff(f,x);fy=diff(f,y);
```

```
rr=solve([fx==0,fy==0],x,y,solution_dict=True);rr
```

$[\{y : 1, x : 1\}]$

```
[f.subs(r) for r in rr]
```

$[0]$

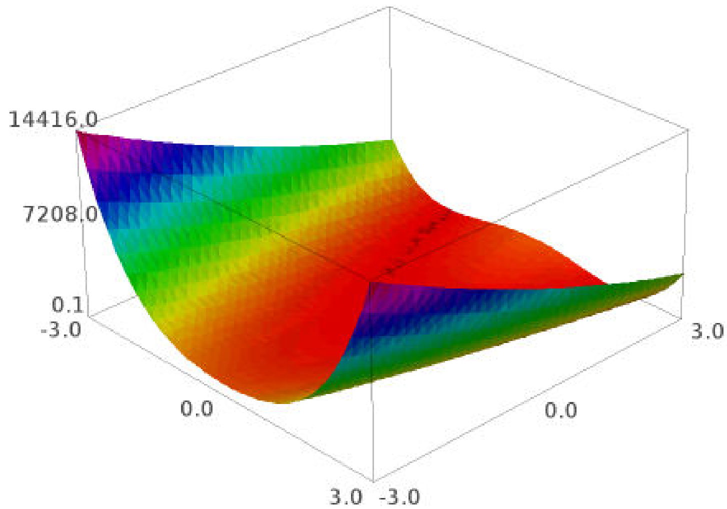
```
ee=f.hessian();ee
```

$$\begin{pmatrix} 1200\,x^2 - 400\,y + 2 & -400\,x \\ -400\,x & 200 \end{pmatrix}$$

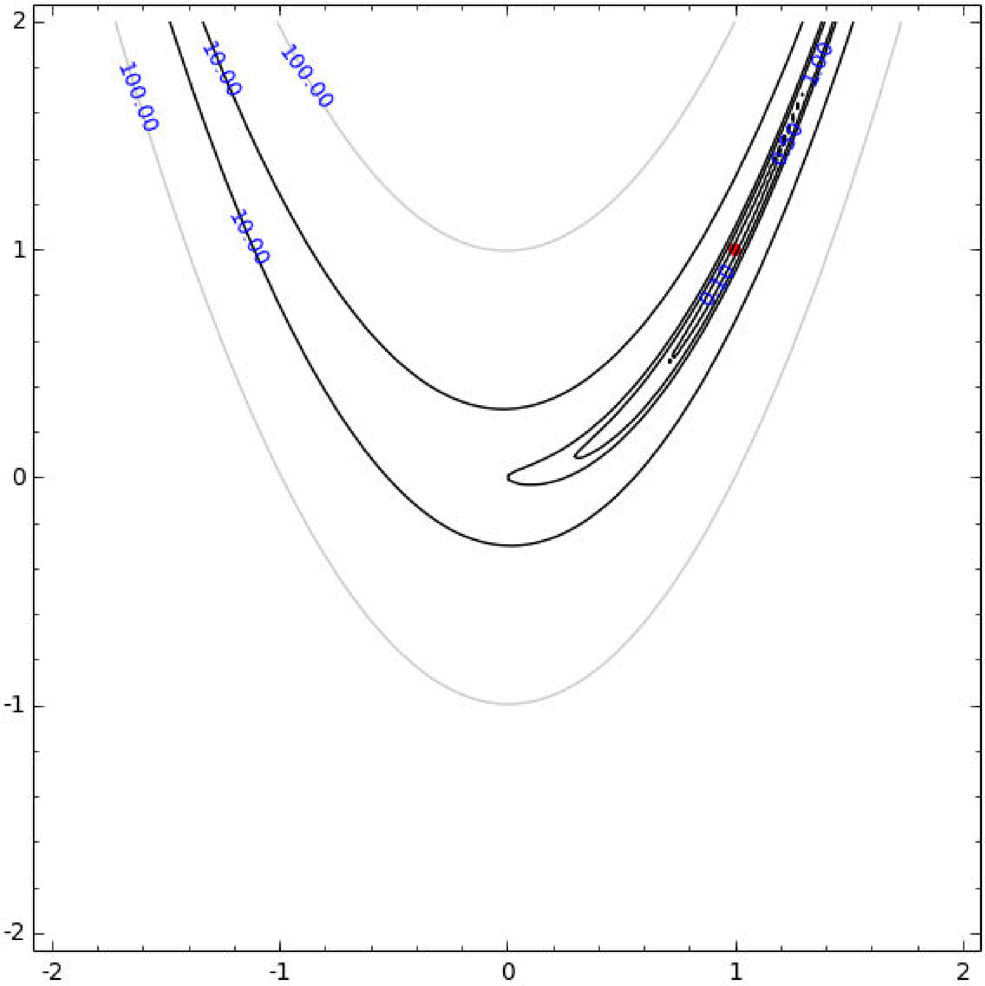
```
[map(n,ee.subs(r).eigenvalues()) for r in rr]
```

$[[0.399360767487622, 1001.60063923251]]$

```
plot3d(f,(x,-3,3),(y,-3,3),adaptive=True,mesh=True)
```




```
contour_plot(f,(x,-2,2),(y,-2,2),fill=False,contours=[1/10,1
/2,1,10,100],labels=True,plot_points=200)+point2d([[1,1]],size=30,color='red')
```



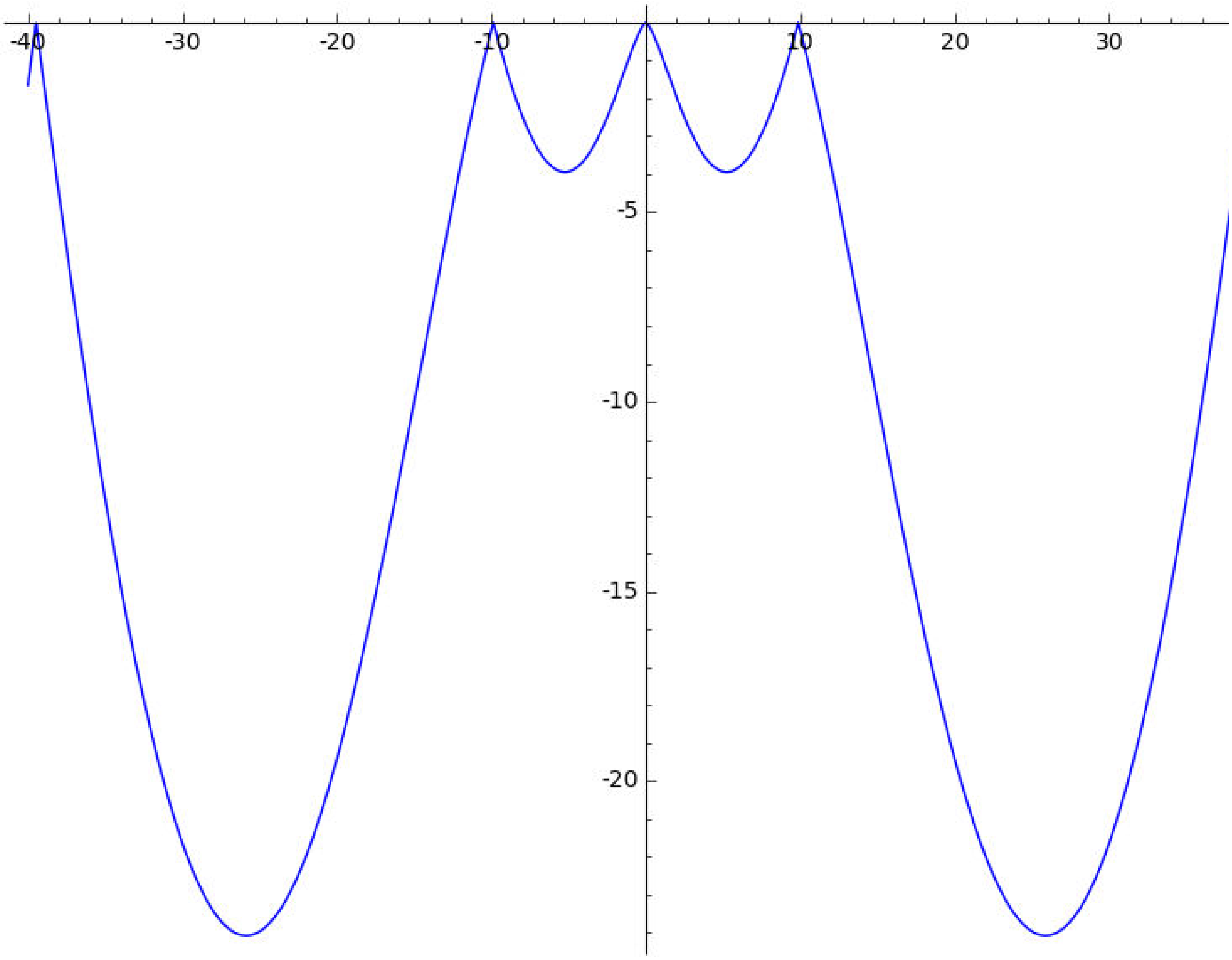

```
minimize(f,[5,5])
```

```
Optimization terminated successfully.
         Current function value: 0.000000
         Iterations: 46
         Function evaluations: 66
         Gradient evaluations: 66
```

(1.00000000069, 1.00000000136)

```
f=-abs(x*sin(sqrt(abs(x))))
```

```
plot(f,x,-40,40)
```




```
minimize(f,[1])
```

```
Optimization terminated successfully.
         Current function value: -3.945302
         Iterations: 4
         Function evaluations: 6
         Gradient evaluations: 6
```

(5.23919401903)

```
f=(3/2-x*(1-y))^2+(9/4-x*(1-y^2))^2+(21/8-x*(1-y^3))^2
```

```
minimize(f,[1,1])
```

```
Optimization terminated successfully.
         Current function value: 0.000000
         Iterations: 15
         Function evaluations: 21
         Gradient evaluations: 21
```

$(3.00000030268, 0.50000011869)$

```
fx=diff(f,x);
fy=diff(f,y);
```

```
s=solve([fx==0,fy==0],[x,y],solution_dict=True);s
```

$\left[\left\{y : -\frac{1}{14}\, i\, \sqrt{307} - \frac{13}{14}, x : 0\right\}, \left\{y : \frac{1}{14}\, i\, \sqrt{307} - \frac{13}{14}, x : 0\right\}, \{y : 1, x : 0\}, \left\{y : \frac{1}{2}, x : 3\right\}, \{y : -0.511076540822\right.$

```
rs=[[x.subs(r),y.subs(r)]  for r in s if x.subs(r) in RR and y.subs(r) in
RR];rs
```

$\left[[0, 1], \left[3, \frac{1}{2}\right], [0.100537933593, -2.64451367255]\right]$