Symbolic manipulation

In [1]:
```
f(x) = x^2+x-2
show(f(x))
```

$x^2 + x - 2$

In [2]:
```
f
```

Out[2]:
```
x |--> x^2 + x - 2
```

In [3]:
```
type(f)
```

Out[3]:
```
<class 'sage.symbolic.expression.Expression'>
```

In [4]:
```
 f(500)
```

Out[4]:
```
250498
```

In [5]:
```
f(-pi)
```

Out[5]:
```
-pi + pi^2 - 2
```

In [6]:
```
f(-pi).n()
```

Out[6]:
```
4.72801174749956
```

In [7]:
```
solve(f(x)==0,x)
```

Out[7]:
```
[x == 1, x == -2]
```

In [8]:
```
solve(f(x)==0,x,solution_dict=True)
```

Out[8]:
```
[{x: 1}, {x: -2}]
```

In [9]:
```
solve(f(x)==0,x,solution_dict=False)
```

Out[9]:
```
[x == 1, x == -2]
```

In [10]:
```
f.coefficients()
```

Out[10]:
```
[[x |--> -2, x |--> 0], [x |--> 1, x |--> 1], [x |--> 1, x |--> 2
```

In [11]:
```
 f.roots()
```

Out[11]:
```
[(1, 1), (-2, 1)]
```

In [12]:
```
f.factor()
```

Out[12]:
```
x |--> (x + 2)*(x - 1)
```

In [13]:
```
a = (x^2+2*x+1).roots()
a
```

Out[13]:
```
[(-1, 2)]
```

In [14]:
```
a = (x^2+x+1).roots()
a
```

Out[14]:
```
[(-1/2*I*sqrt(3) - 1/2, 1), (1/2*I*sqrt(3) - 1/2, 1)]
```

In [15]:
```
show(a)
```

$$\left[\left(-\frac{1}{2} i \sqrt{3} - \frac{1}{2}, 1\right), \left(\frac{1}{2} i \sqrt{3} - \frac{1}{2}, 1\right)\right]$$

In [16]:
```
show((x^5+x+1).roots())
```

$$\left[\left(-\frac{1}{6}\left(\frac{1}{2}\right)^{\frac{1}{3}}\left(3\sqrt{23}\sqrt{3}-25\right)^{\frac{1}{3}}\left(i\sqrt{3}+1\right) - \frac{\left(\frac{1}{2}\right)^{\frac{2}{3}}\left(-i\right.}{3\left(3\sqrt{23}\sqrt{\right.}}\right.\right.$$

$$\left(-\frac{1}{6}\left(\frac{1}{2}\right)^{\frac{1}{3}}\left(3\sqrt{23}\sqrt{3}-25\right)^{\frac{1}{3}}\left(-i\sqrt{3}+1\right) - \frac{\left(\frac{1}{2}\right)^{\frac{2}{3}}\left(i\right.}{3\left(3\sqrt{23}\right.}\right.$$

$$\left(\frac{1}{3}\left(\frac{1}{2}\right)^{\frac{1}{3}}\left(3\sqrt{23}\sqrt{3}-25\right)^{\frac{1}{3}} + \frac{2\left(\frac{1}{2}\right)^{\frac{2}{3}}}{3\left(3\sqrt{23}\sqrt{3}-25\right)^{\frac{1}{3}}} + \frac{1}{3}, 1\right), \left(-\frac{1}{2} i\right.$$

In [17]:
```
var('a,b,c')
sol = solve(a*x^2+b*x+c==0,x)
show(sol)
```

$$\left[x = -\frac{b + \sqrt{b^2 - 4\,ac}}{2\,a}, x = -\frac{b - \sqrt{b^2 - 4\,ac}}{2\,a}\right]$$

In [18]:
```
var('x,y')
solve([x+y==6,x-y==4],[x,y])
```

Out[18]:
```
[[x == 5, y == 1]]
```

In [19]:
```
var('x,y')
solve([x+y==6,x+y==7],[x,y])
```

Out[19]:
```
[]
```

In [20]:
```
solve([x+y==5],[x,y])
```

Out[20]:
```
[[x == -r1 + 5, y == r1]]
```

Solving a system of non linear equations

In [21]:
```
s = solve([x^2+y^2==1, x*y==1/4],[x,y],solution_dict=True)
show(s)
```

$$\left[\left\{x : -\frac{1}{2}\sqrt{\sqrt{3}+2}, y : \frac{1}{2}\sqrt{\sqrt{3}+2}\left(\sqrt{3}-2\right)\right\}, \left\{x : \frac{1}{2}\sqrt{\sqrt{3}+2}, y\right.\right.$$
$$\left\{x : -\frac{1}{2}\sqrt{-\sqrt{3}+2}, y : -\frac{1}{4}\sqrt{3}\sqrt{2} - \frac{1}{4}\sqrt{2}\right\}, \left\{x : \frac{1}{2}\sqrt{-\sqrt{3}+\right.$$

In [22]:
```
show( s[0])
```

$$\left\{x : -\frac{1}{2}\sqrt{\sqrt{3}+2}, y : \frac{1}{2}\sqrt{\sqrt{3}+2}\left(\sqrt{3}-2\right)\right\}$$

In [23]:
```
show(solve(x^2-2*x-1>8,x))
```

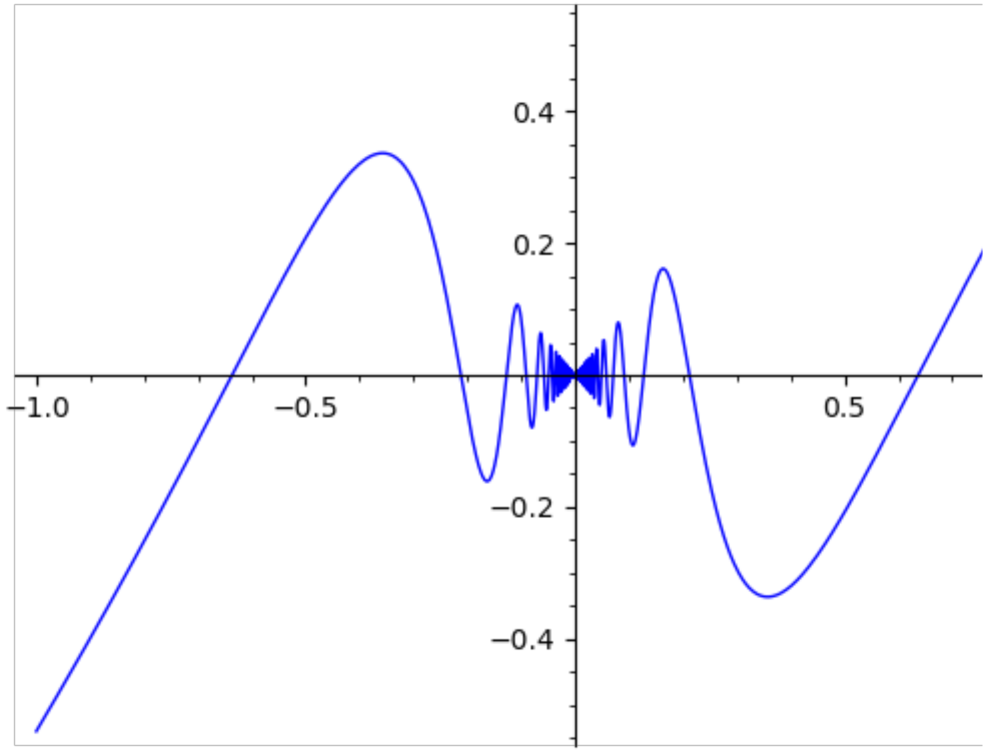
$$\left[\left[x < -\sqrt{10}+1\right], \left[x > \sqrt{10}+1\right]\right]$$

Graph of explicit functions

In [24]:
```
var('x')
f(x) = x*cos(1/x)
```

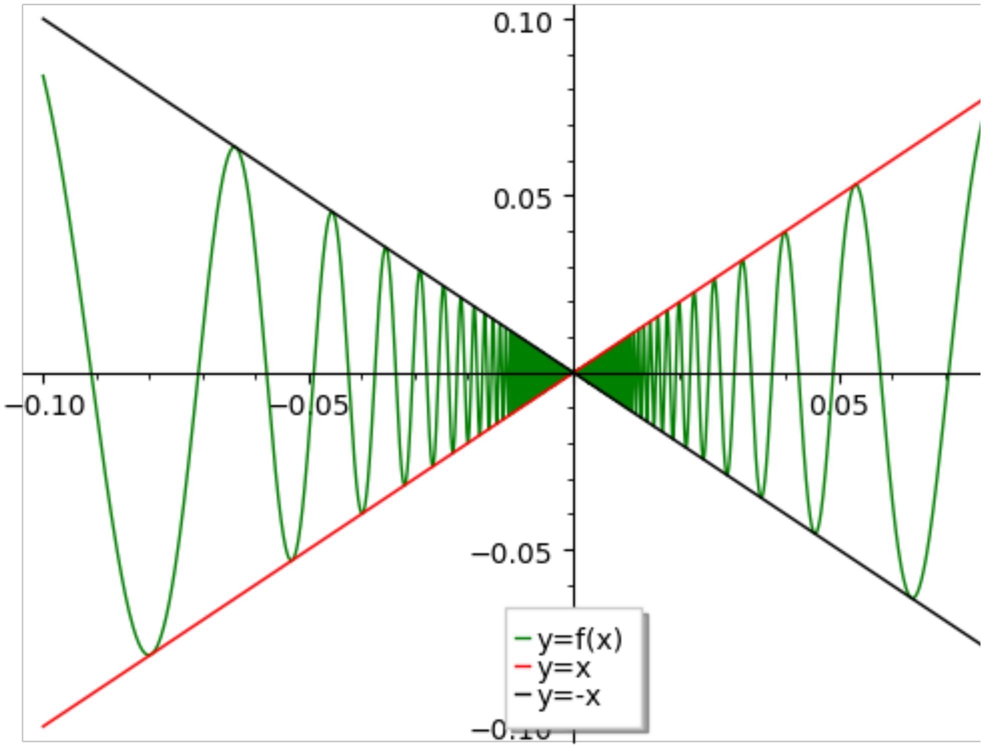
In [25]:
```
f.plot()
```

Out[25]:



Plotting multiple graphs together Example: Plot the graph of y = xcos1/x,y = x and
together.

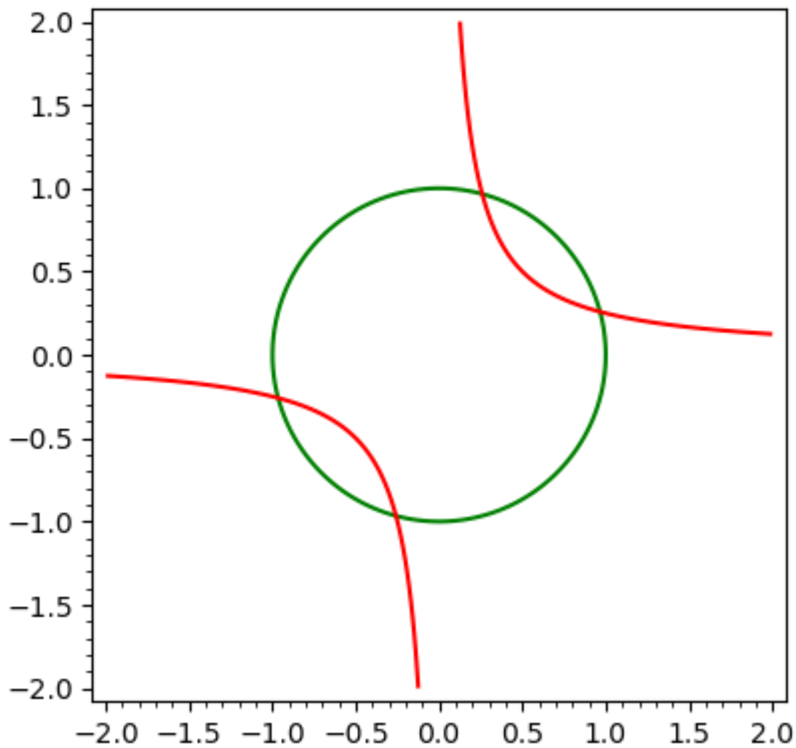
In [26]:
```
p = plot(f, (x, -0.1,0.1),figsize = 6,color = 'green',legend_labe
p1 = plot(x, -0.1,0.1, color='red',legend_label='y=x')
p2 = plot(-x, -0.1,0.1, color='black',legend_label='y=-x')
show(p+p1+p2,figsize=6)
```



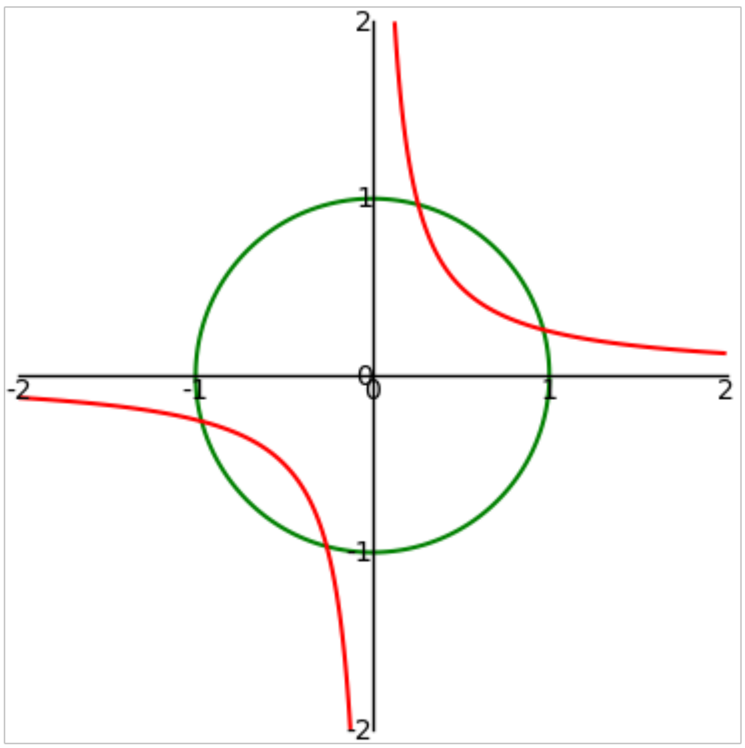
Implicit Plot

In [27]:

```
var('x y')
f(x, y) = x^2 + y^2 - 1
g(x, y) = x * y - 1/4

p = implicit_plot(f, (x, -2, 2), (y, -2, 2), figsize=6, color='gr
p1 = implicit_plot(g, (x, -2, 2), (y, -2, 2), color='red', legend
show(p+p1)
```



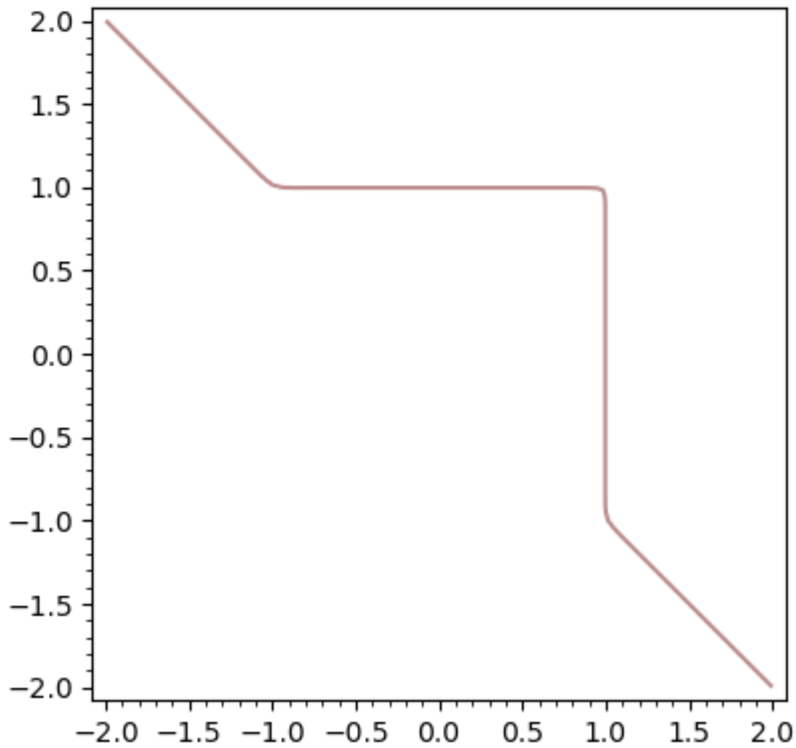
In [28]:
```
var('x y')
f(x, y) = x^2 + y^2 - 1
g(x, y) = x * y - 1/4

p = implicit_plot(f, (x, -2, 2), (y, -2, 2), figsize=6, color='gr
p1 = implicit_plot(g, (x, -2, 2), (y, -2, 2), color='red', legend

# Create axis lines
axis_x = line([(-2, 0), (2, 0)], color='black')
axis_y = line([(0, -2), (0, 2)], color='black')

# Create ticks for x and y axes
ticks_x = [text(str(i), (i, 0), color='black', vertical_alignment
ticks_y = [text(str(i), (0, i), color='black', horizontal_alignme

# Combine plots and hide default axes
final_plot = p + p1 + axis_x + axis_y + sum(ticks_x) + sum(ticks_
final_plot.show(frame=False, axes=False, figsize=6)
```



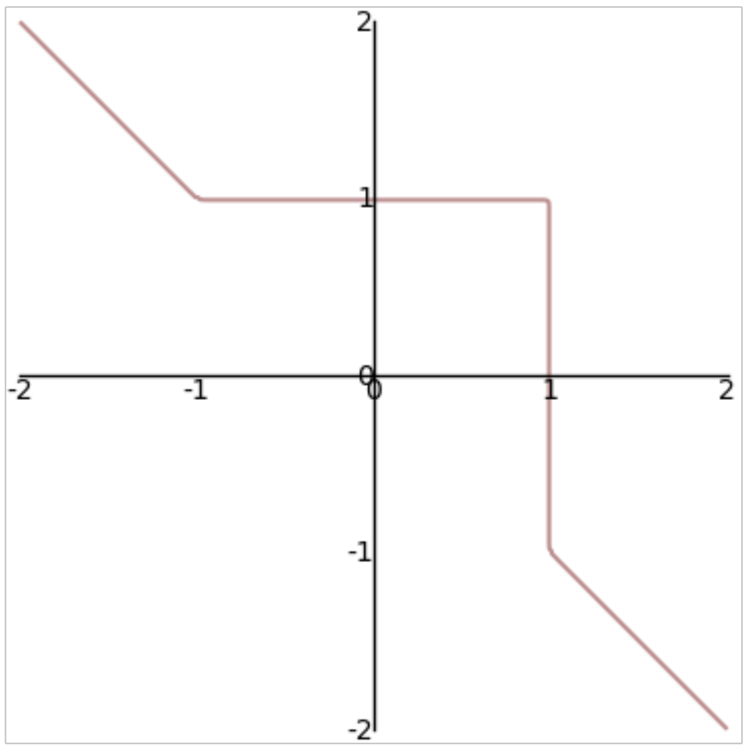
Example: Let us draw a curve given by implicitely defined function x^n +y^n = 1 f
values of n.

In [29]:
```
var('x,y')
n = 41
f(x,y) = x^n+y^n-1
implicit_plot(f,(x,-2,2),(y,-2,2),color='rosybrown')
```

Out[29]:

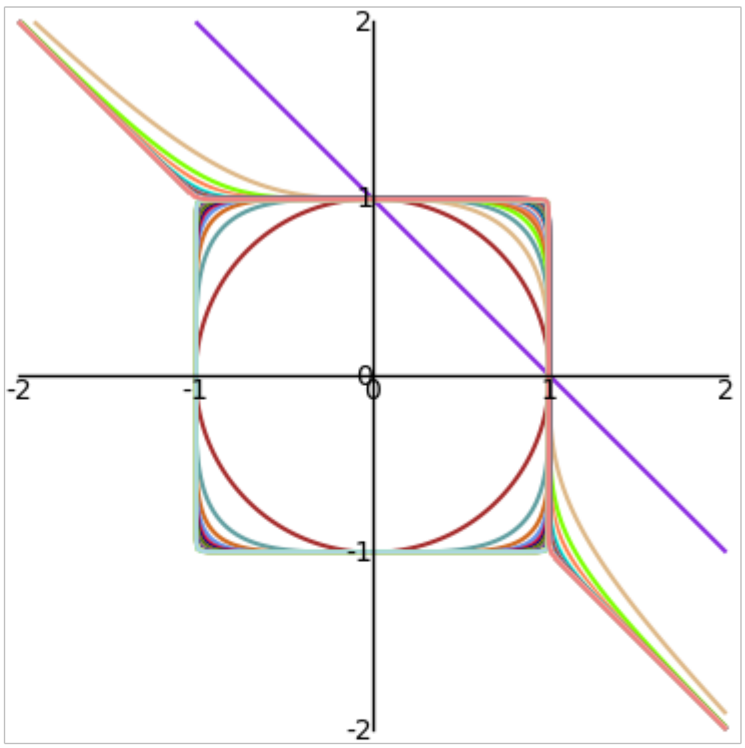


In [30]:
```
var('x,y')
n = 101
f(x,y) = x^n+y^n-1
p = implicit_plot(f,(x,-2,2),(y,-2,2),color='rosybrown')

# Create axis lines
axis_x = line([(-2, 0), (2, 0)], color='black')
axis_y = line([(0, -2), (0, 2)], color='black')

# Create axis lines
ticks_x = [text(str(i), (i, 0), color='black', vertical_alignment
ticks_y = [text(str(i), (0, i), color='black', horizontal_alignme

# Combine plots and hide default axes
final_plot = p +  axis_x + axis_y + sum(ticks_x) + sum(ticks_y)
final_plot.show(frame=False, axes=False, figsize=6)
```



In [31]:

```
var('x,y')
plt = []
for i in range(1,60):
    n = i
    f(x,y) = x^n+y^n-1
    p = implicit_plot(f,(x,-2,2),(y,-2,2),color=list(colors.keys(
    plt.append(p)
p1 = sum(plt)
# Create axis lines
axis_x = line([(-2, 0), (2, 0)], color='black')
axis_y = line([(0, -2), (0, 2)], color='black')

# Create axis lines
ticks_x = [text(str(i), (i, 0), color='black', vertical_alignment
ticks_y = [text(str(i), (0, i), color='black', horizontal_alignme

# Combine plots and hide default axes
final_plot = p1 +  axis_x + axis_y + sum(ticks_x) + sum(ticks_y)
final_plot.show(frame=False, axes=False, figsize=6)
```



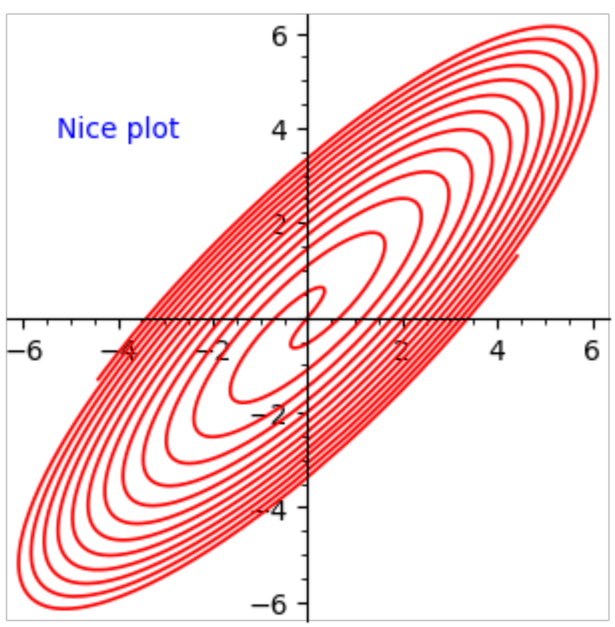
In [32]:
```
colors
```

Out[32]:
```
{'automatic': RGB color (0.6784313725490196, 0.8470588235294118,
137255), 'aliceblue': RGB color (0.9411764705882353, 0.9725490196
1.0), 'antiquewhite': RGB color (0.9803921568627451, 0.9215686274
31372549019608), 'aqua': RGB color (0.0, 1.0, 1.0), 'aquamarine':
(0.4980392156862745, 1.0, 0.8313725490196079), 'azure': RGB color
05882353, 1.0, 1.0), 'beige': RGB color (0.9607843137254902, 0.96
02, 0.8627450980392157), 'bisque': RGB color (1.0, 0.894117647058
274509803922), 'black': RGB color (0.0, 0.0, 0.0), 'blanchedalmon
r (1.0, 0.9215686274509803, 0.803921568627451), 'blue': RGB color
1.0), 'blueviolet': RGB color (0.5411764705882353, 0.168627450980
2745098039215), 'brown': RGB color (0.6470588235294118, 0.1647058
0.16470588235294117), 'burlywood': RGB color (0.8705882352941177,
4509804, 0.5294117647058824), 'cadetblue': RGB color (0.372549019
6196078431372549, 0.6274509803921569), 'chartreuse': RGB color (0
62745, 1.0, 0.0), 'chocolate': RGB color (0.8235294117647058, 0.4
529, 0.11764705882352941), 'coral': RGB color (1.0, 0.49803921568
7254901960784), 'cornflowerblue': RGB color (0.39215686274509803,
4901961, 0.9294117647058824), 'cornsilk': RGB color (1.0, 0.97254
0.8627450980392157), 'crimson': RGB color (0.8627450980392157, 0.
```

Parametric Plot in 2D

Example: Plot the graph of the function given by parametric coordinates x = tsin(
tcos(t^2), −2π ≤t≤2π.

In [33]:
```
var('t')
p = parametric_plot([t*sin(1-t^2),t*cos(t^2)],(t,-2*pi,2*pi),colo
txt = text('Nice plot',(-4,4))
show(p+txt,figsize=5)
```



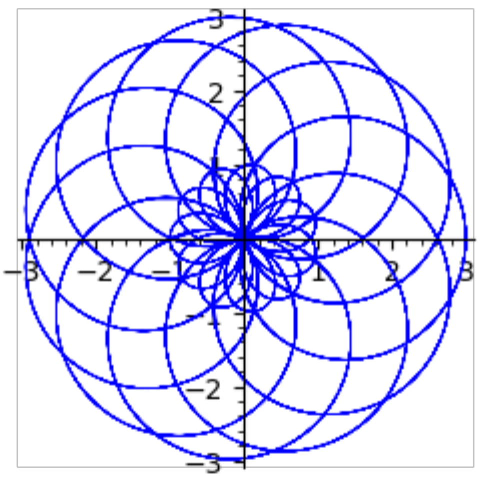
Polar Plot

Example: Plot the graph of the function g by polar curve

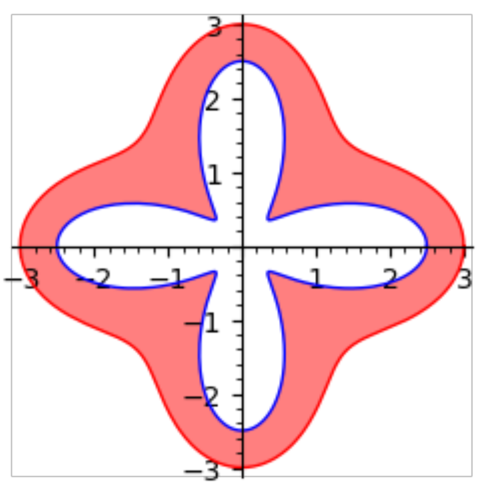
```
r = 1+2cos(sθ), 0≤ θ ≤200π
for various values of s = 1,1.01,1.02,1.05,1.5 etc
```

In [34]:
```
var('theta')
s = 1.1
polar_plot(1+2*cos(s*theta),(theta,0,200*pi),plot_points=5000,thi
```

Out[34]:



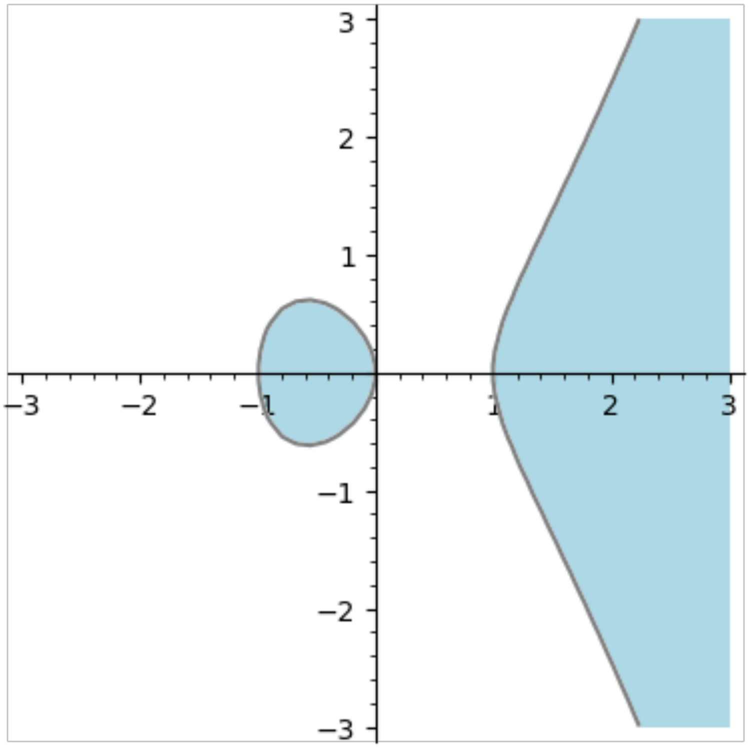
In [35]:
```
var('t')
p1 = polar_plot(0.5 * cos(4*t) + 2.5, 0, 2*pi,color='red')
p2 =polar_plot(cos(4*t) + 1.5, 0, 2*pi, fill=0.5 * cos(4*t) + 2.5
show(p1+p2,figsize=4)
```



Region Plot

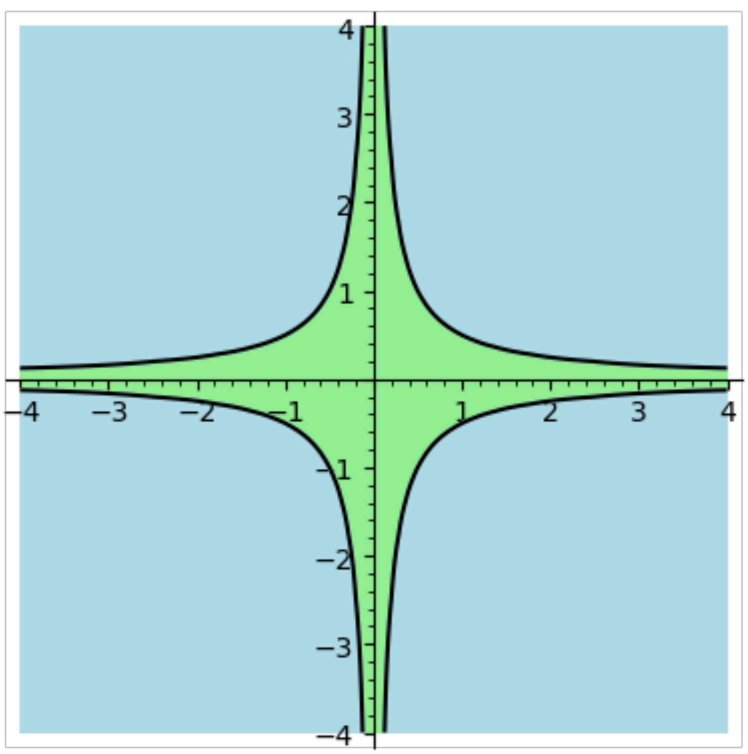
In [36]:
```
region_plot(x*(x-1)*(x+1) -y^2 > 0, (x,-3,3), (y,-3,3),incol='lig
```

Out[36]:

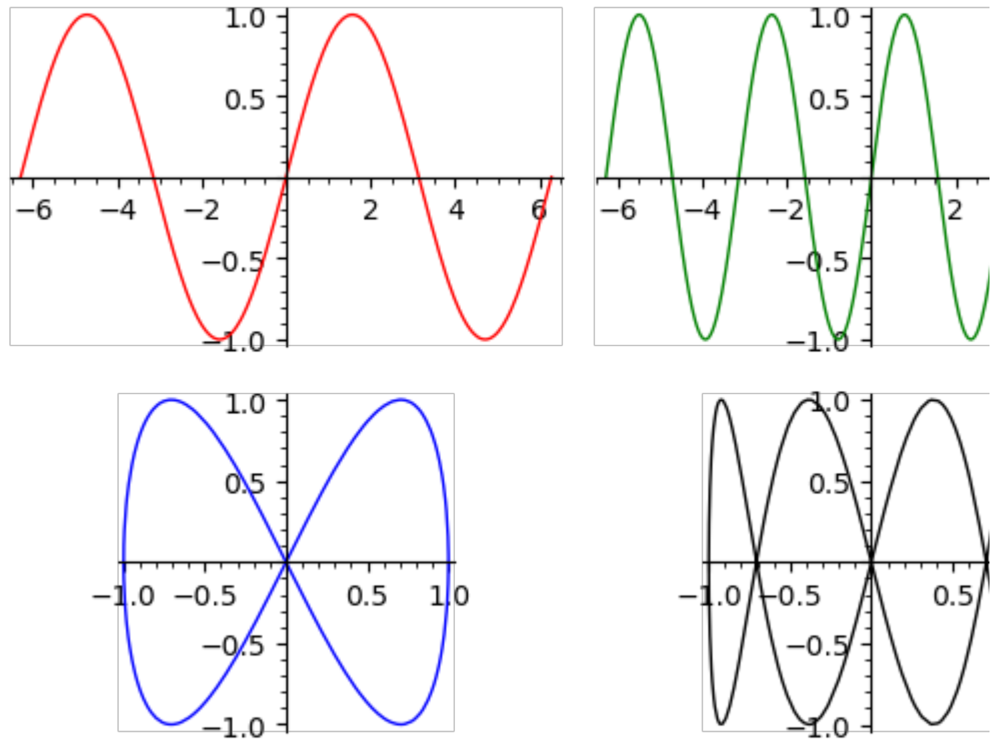


In [37]:
```
region_plot([x*y<=1/2 , x*y>=-1/2],(x,-4,4),(y,-4,4),incol='light
```

Out[37]:



In [38]:
```
f(x) = sin(x)
g(x) = sin(2*x)
h(x) = sin(4*x)
p1 = plot(f,(-2*pi,2*pi),color='red')
p2 = plot(g,(-2*pi,2*pi),color='green')
p3 = parametric_plot((f,g),(0,2*pi),color='blue')
p4 = parametric_plot((f,h),(0,2*pi),color='black')
L = [p1,p2,p3,p4]
G=graphics_array(L,2)
G.show(figsize=6)
```

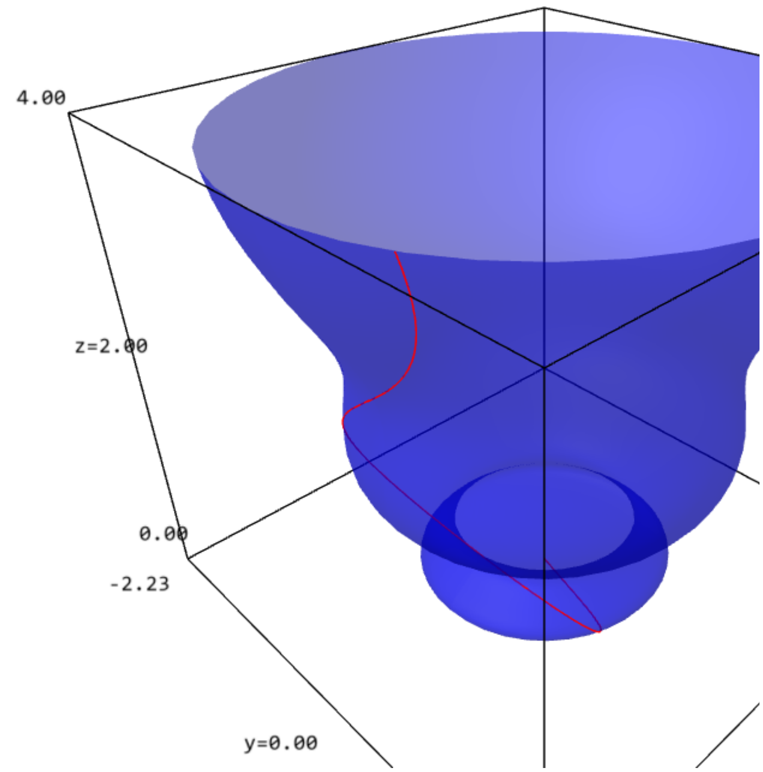


Plotting 3d graphs in SageMath

Example: Plot the graph of f(x,y) = xsin(x + y)cos(x−y) in the domain −2 ≤ x ≤ 2 aı

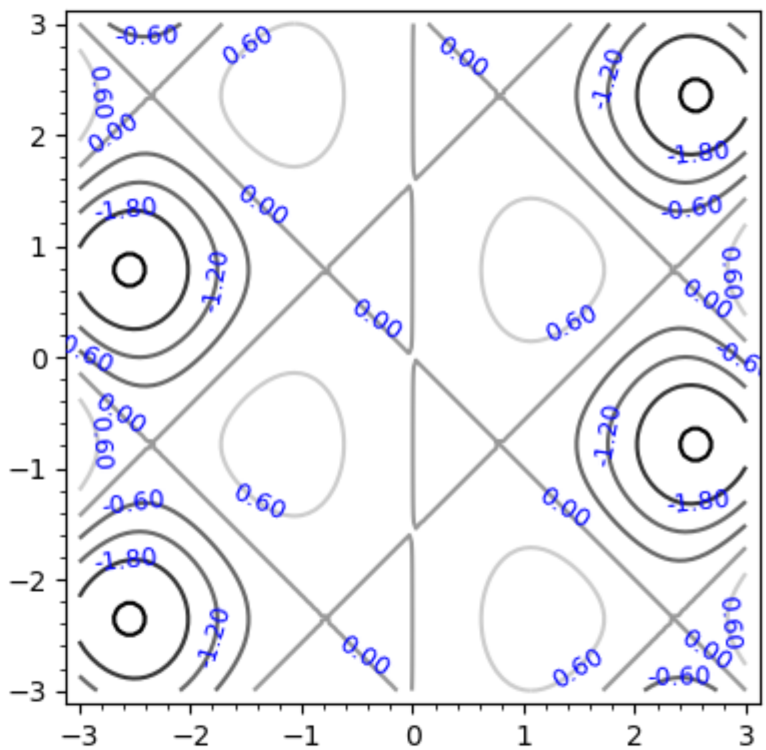
In [39]:
```
var('x,y')
f(x,y) = x*sin(x+y)*cos(x-y)
plot3d(f(x,y), (x,-3,3),(y,-3,3))
```

Out[39]:



In [40]:
```
contour_plot(f(x,y),(x,-3,3),(y,-3,3),fill=False,contours=5,label
```

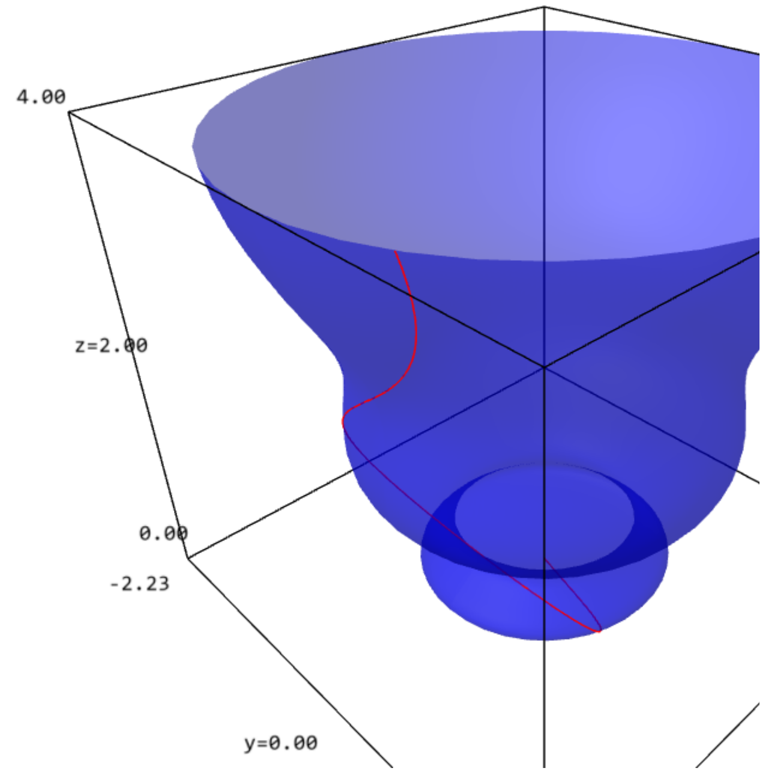
Out[40]:



In [41]:
```
plot3d??
```

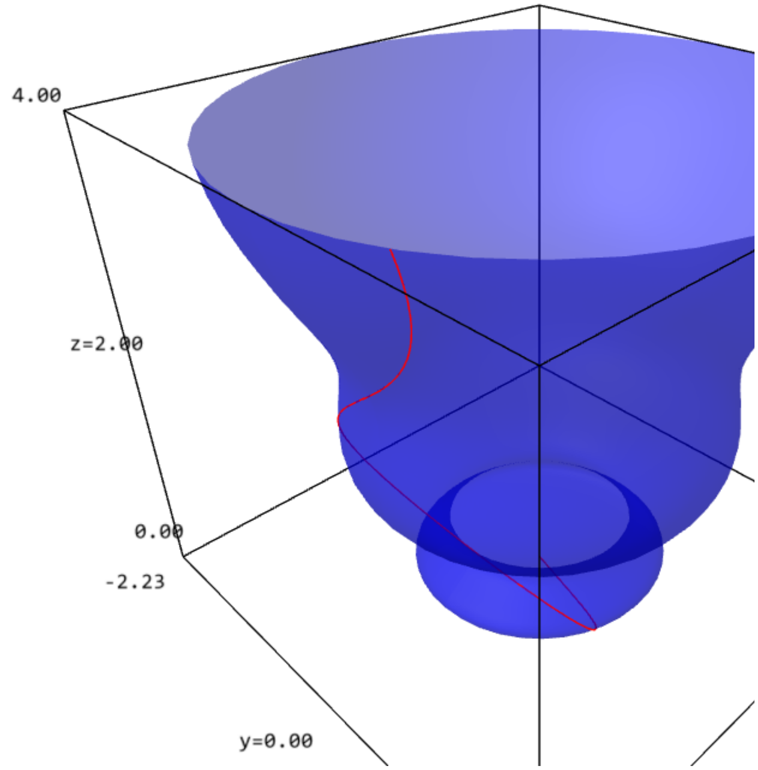
In [42]:
```
plot3d(f(x,y), (x,-2,2),(y,-2,2))+sphere(color='green')
```

Out[42]:

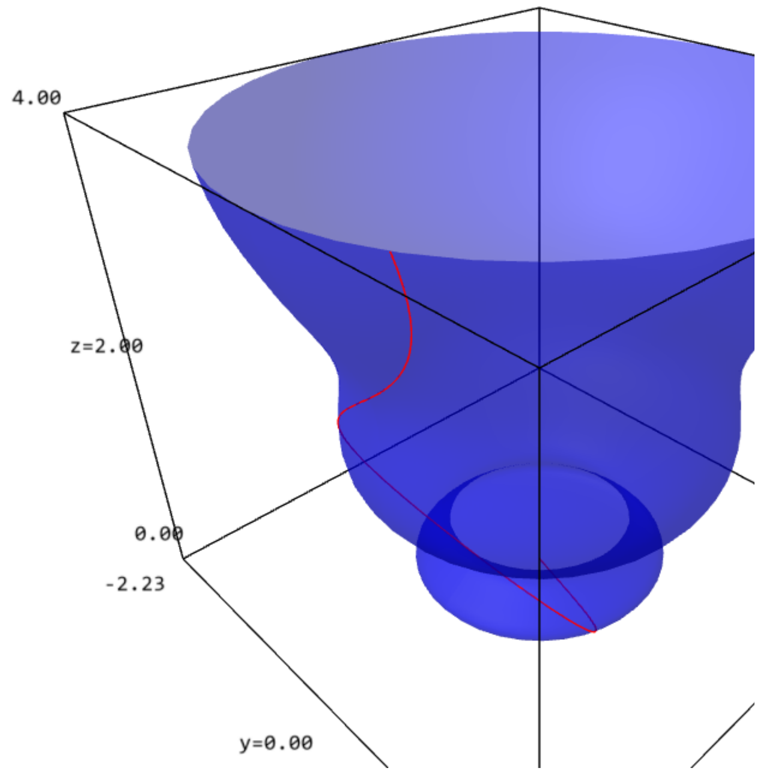


In [43]:
```
var('x,y,z')
implicit_plot3d(x^2+y^2+z^2==4, (x,-3,3), (y,-3,3), (z,-3,3))
```

Out[43]:



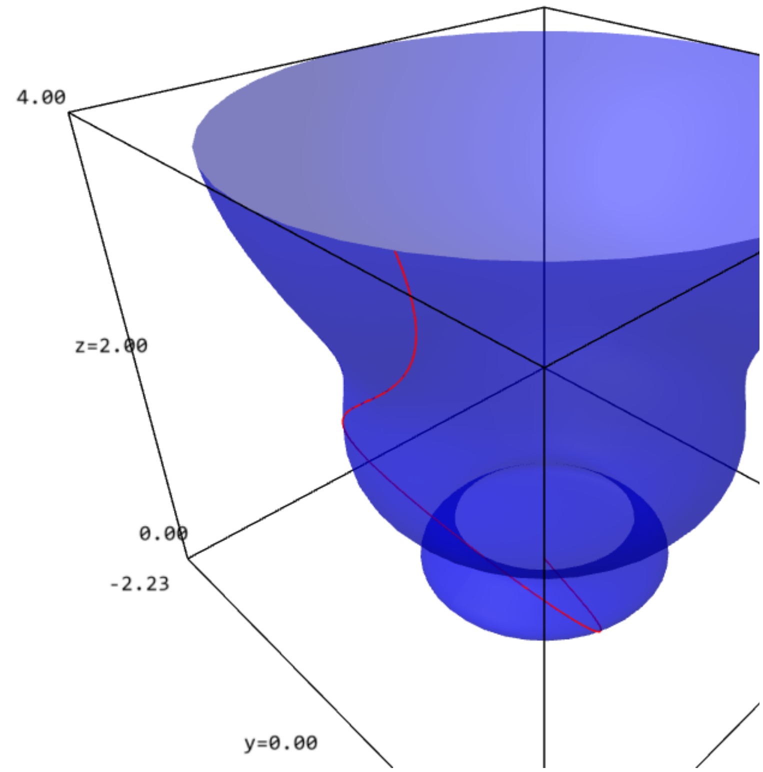
In [44]:
```
F = (x^2+9/4*y^2+z^2-1)^3 - x^2*z^3 - 9/(80)*y^2*z^3
r = 1.5
V = implicit_plot3d(F, (x,-r,r), (y,-r,r), (z,-r,r), plot_points=
show(V)
```



A hyperboloid

In [45]:
```
var('z')
implicit_plot3d(x^2 + y^2 - z^2 -0.3, (x,-2,2), (y,-2,2), (z,-1.8
```

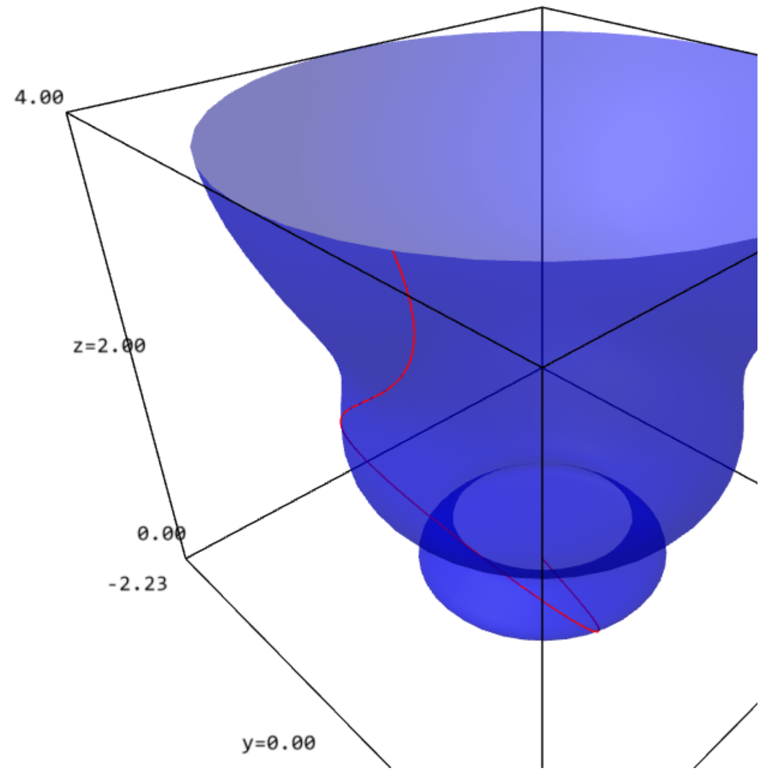
Out[45]:



Cube

In [46]:
```
n = 100
implicit_plot3d(x^n + y^n + z^n - 1, (x,-2,2), (y,-2,2), (z,-2,2)
```

Out[46]:

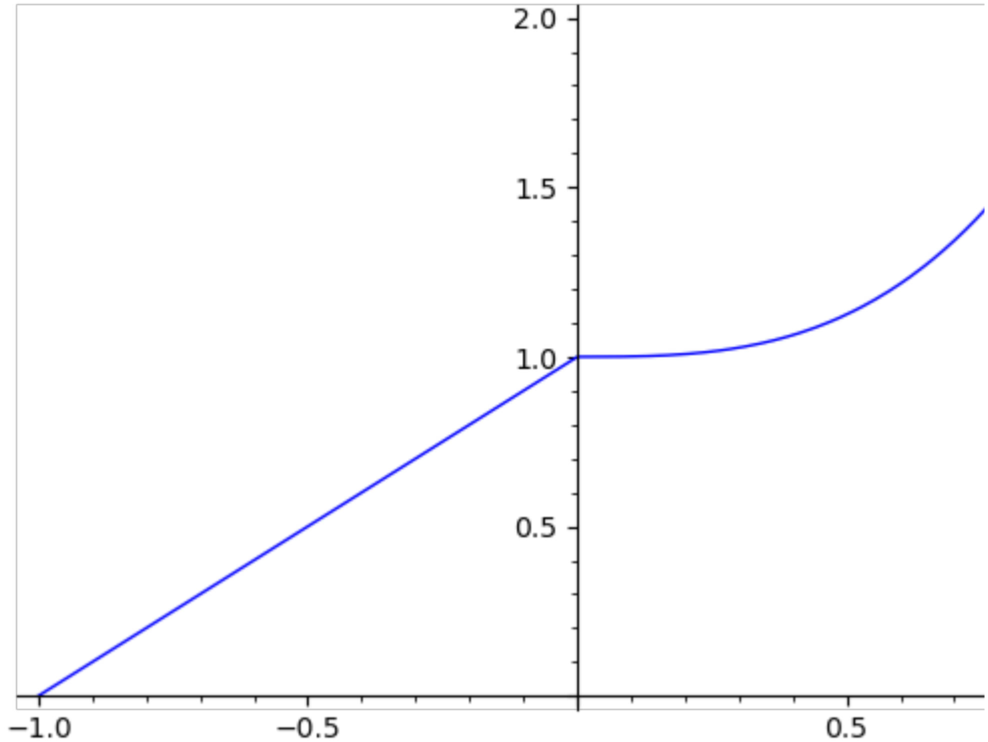


Piecewise functions

Example: Plot the graph of the function f(x) which is 1+ x^3 in [0,1] and 1+ x in [−

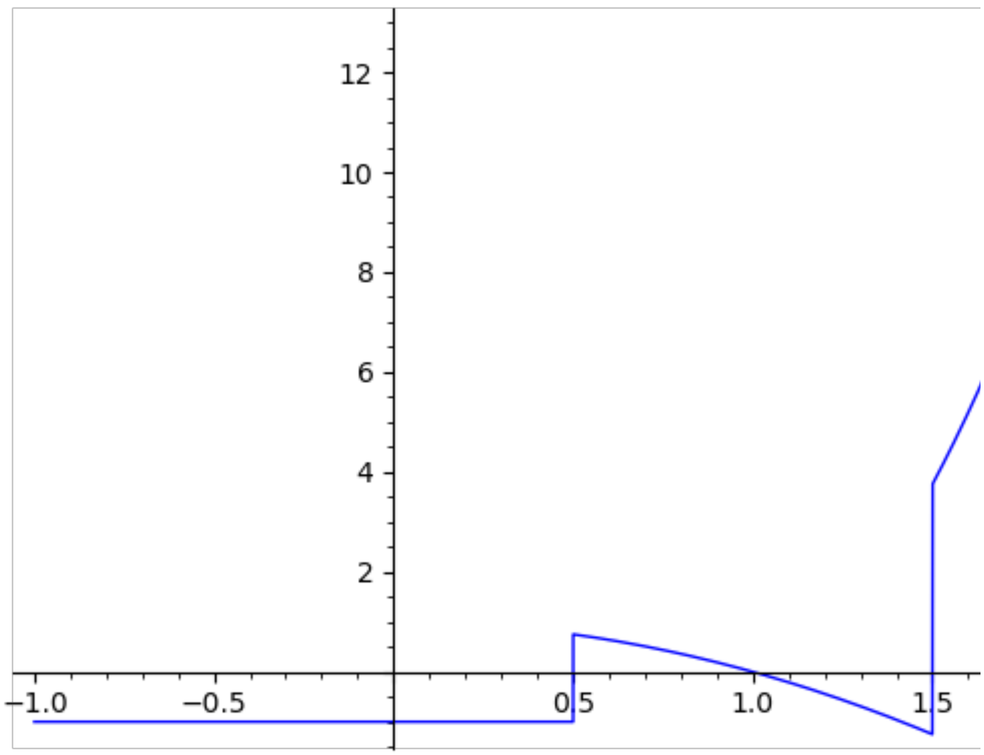
In [47]:
```
f = piecewise([((0,1), 1+x^3), ([-1,0], 1+x)]);
f.plot(-1,1)
```

Out[47]:



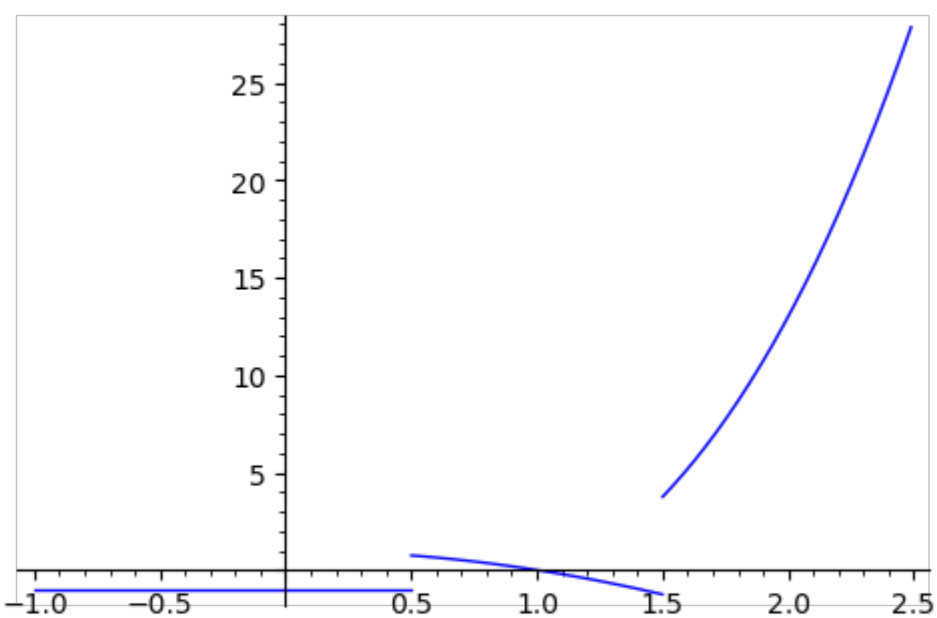
In [48]:
```
f1(x) = -1
f2(x) = 1-x^2
f3(x) = 2*x^3-3
f = piecewise([[(-1,0.5),f1],[(0.5,1.5),f2],[(1.5,2.5),f3]])
f.plot(-1,2)
```

Out[48]:



In [49]:
```
f.plot(-1,2.5,exclude = [0.5,1.5],figsize=5)
```

Out[49]:



One Variable Calculus with SageMath

In [50]:
```
f(x)=sin(x)
```

In [51]:
```
a = 0
f.limit(x=a)
```

Out[51]: x |--> 0

In [52]:
```
f(x)=sin(x)/x
a = 0
f.limit(x=a)
```

Out[52]: x |--> 1

In [53]:
```
f.limit(x=a,dir='-')
```

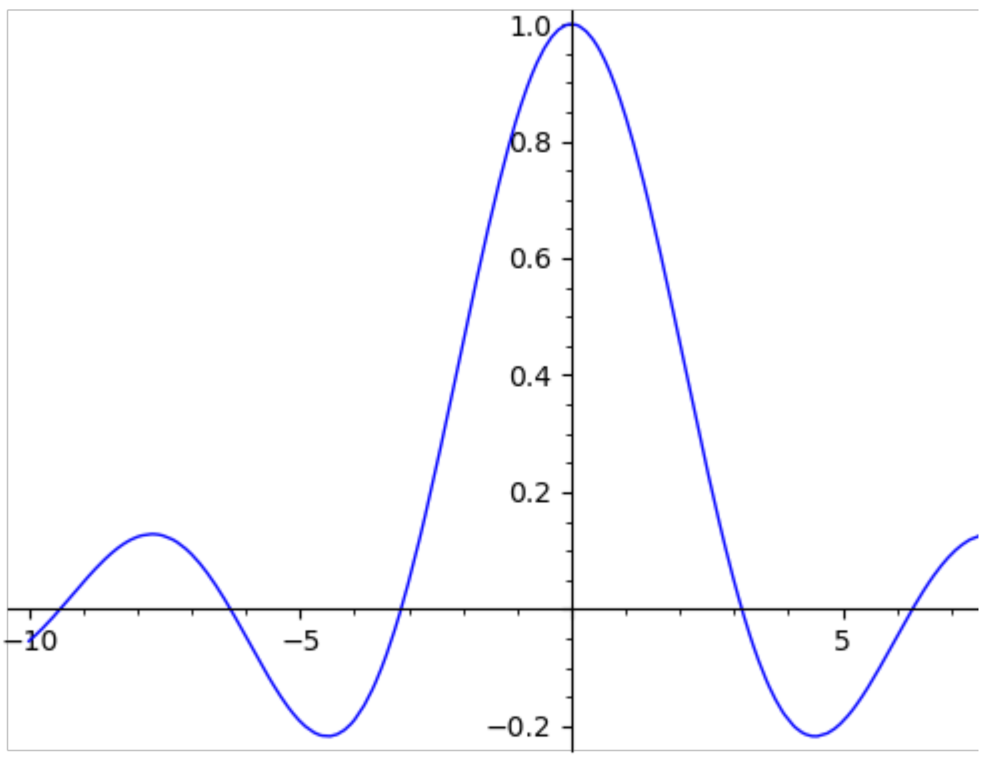
Out[53]: x |--> 1

In [54]:
```
f.limit(x=a,dir='+')
```

Out[54]: x |--> 1

In [55]:
```
f.plot((x,-10,10))
```

Out[55]:



In [56]:
```
f(x)=sin(x)
df = f.diff()
show(df)
```

$x \mapsto \cos(x)$

In [57]:
```
f(x)=sin(x)/x
df = f.diff()
show(df)
```

$$x \mapsto \frac{\cos(x)}{x} - \frac{\sin(x)}{x^2}$$

In [58]:
```
d2f = f.diff(2)
show(d2f)
```

$$x \mapsto -\frac{\sin(x)}{x} - \frac{2\cos(x)}{x^2} + \frac{2\sin(x)}{x^3}$$

In [59]:
```
d2f = f.diff(20)
show(d2f)
```

$$x \mapsto \frac{\sin(x)}{x} + \frac{20\cos(x)}{x^2} - \frac{380\sin(x)}{x^3} - \frac{6840\cos(x)}{x^4} + \frac{116280\sin(x)}{x^5}$$
$$- \frac{390700800\cos(x)}{x^8} + \frac{5079110400\sin(x)}{x^9} + \frac{60949324800\ \mathrm{c}}{x^{10}}$$
$$- \frac{6704425728000\cos(x)}{x^{12}} + \frac{60339831552000\sin(x)}{x^{13}} + \frac{4827186524160}{x^{14}}$$
$$- \frac{20274183401472000\cos(x)}{x^{16}} + \frac{101370917007360000\sin(x)}{x^{17}} -$$
$$- \frac{1216451004088320000\sin(x)}{x^{19}} - \frac{2432902008176640000\cos(x)}{x^{20}}$$

In [60]:
```
f(x)=sin(x)
show(f.integral(x))
```

$$x \mapsto -\cos(x)$$

In [61]:
```
f.integral(x,0,1)
```

Out[61]:
```
-cos(1) + 1
```

In [62]:
```
f(x) = x^2*sin(2*x)+x^2*exp(-x)+x^2-x+3
show(f(x))
```

$$x^2 e^{(-x)} + x^2\sin(2x) + x^2 - x + 3$$

In [63]:
```
a = oo
f.limit(x=a)
```

Out[63]:
```
x |--> +Infinity
```

In [64]:
```
limit(f(x),x=1)
```

Out[64]:
```
(e*sin(2) + 3*e + 1)*e^(-1)
```

In [65]:
```
limit(f(x),x=1).n()
```
Out[65]:
```
4.27717686799712
```

In [66]:
```
df = f.diff()
show(df)
```
$$x \mapsto 2\,x^2 \cos(2\,x) - x^2 e^{(-x)} + 2\,x e^{(-x)} + 2\,x \sin(2\,x) + 2\,x - 1$$

In [67]:
```
f.derivative()
```
Out[67]:
```
x |--> 2*x^2*cos(2*x) - x^2*e^(-x) + 2*x*e^(-x) + 2*x*sin(2*x) +
```

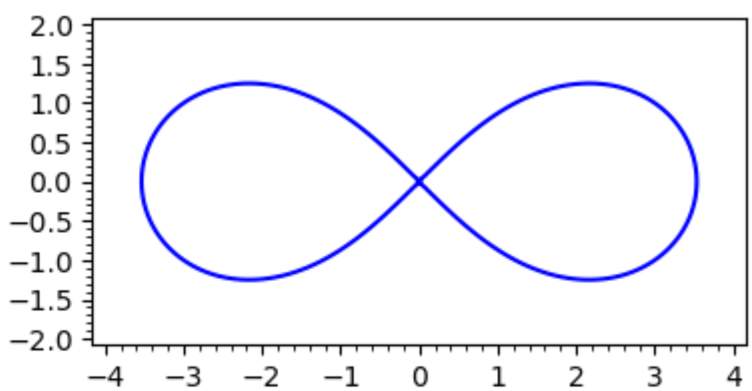
In [68]:
```
show(f.derivative())
```
$$x \mapsto 2\,x^2 \cos(2\,x) - x^2 e^{(-x)} + 2\,x e^{(-x)} + 2\,x \sin(2\,x) + 2\,x - 1$$

In [69]:
```
show(f.integral(x))
```
$$x \mapsto \frac{1}{3}\,x^3 - \frac{1}{2}\,x^2 - \frac{1}{4}\,\left(2\,x^2 - 1\right)\cos(2\,x) - \left(x^2 + 2\,x + 2\right)e^{(-x)} + \frac{1}{2}\,x \sin$$

In [70]:
```
f.integral(x,0,1).n()   # lower limit is 0 amd upper limit is 1
```
Out[70]:
```
3.30262155002575
```

Implicit Derivative

Example: Find the derivative dy/dx from 2(x^2 +y^2)^2 = 25(x^2−y^2)

In [71]:
```
var('x,y')
f(x,y)= 2*(x^2+y^2)^2-25*(x^2-y^2)
curve = implicit_plot(f(x,y),(x,-4,4),(y,-2,2))
curve.show(figsize=4)
```


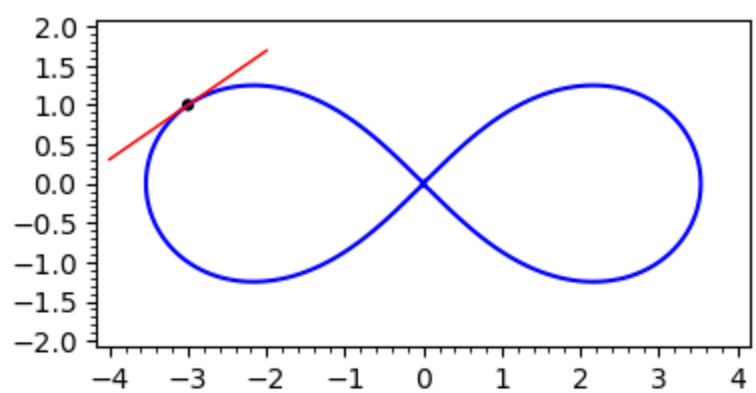
In [72]:
```
dyx = f.implicit_derivative(y,x)
show(dyx)
```

$$-\frac{4\left(x^2+y^2\right)x-25\,x}{4\left(x^2+y^2\right)y+25\,y}$$

In [73]:
```
dxy = f.implicit_derivative(x,y)
show(dxy)
```

$$-\frac{4\left(x^2+y^2\right)y+25\,y}{4\left(x^2+y^2\right)x-25\,x}$$

In [74]:
```
#Grapical interpretation of derivative
a,b = -3,1
m = dyx.subs(x=a,y=b)
T = b+m*(x-a)
pt = point((a,b),size=20,color='black')
tgt = plot(T,-4,-2,color='red')
show(curve+pt+tgt,figsize=4)
m
```



Out[74]: 9/13

Higher order partial derivatives

In [75]:
```
var('x,y,z')
f(x,y)=4*x*y*exp(-x^2-y^2)
show(f(x,y))
```

$$4\,xye^{\left(-x^2-y^2\right)}$$

In [76]:
```
show(f.diff(x)(x,y))
```

$$-8\,x^2ye^{\left(-x^2-y^2\right)}+4\,ye^{\left(-x^2-y^2\right)}$$

In [77]:
```
show(f.diff(y)(x,y))
```

$$-8\,xy^2e^{\left(-x^2-y^2\right)} + 4\,xe^{\left(-x^2-y^2\right)}$$

In [78]:
```
show(f.diff(x,2)(x,y))
```

$$16\,x^3ye^{\left(-x^2-y^2\right)} - 24\,xye^{\left(-x^2-y^2\right)}$$

In [79]:
```
show(f.diff(y,x)(x,y))
```

$$16\,x^2y^2e^{\left(-x^2-y^2\right)} - 8\,x^2e^{\left(-x^2-y^2\right)} - 8\,y^2e^{\left(-x^2-y^2\right)} + 4\,e^{\left(-x^2-y^2\right)}$$

In [80]:
```
show(f.diff(x,y)(x,y))
```

$$16\,x^2y^2e^{\left(-x^2-y^2\right)} - 8\,x^2e^{\left(-x^2-y^2\right)} - 8\,y^2e^{\left(-x^2-y^2\right)} + 4\,e^{\left(-x^2-y^2\right)}$$

In [81]:
```
bool(f.diff(x,y)(x,y)==f.diff(y,x)(x,y))
```

Out[81]: True

Talyor’s Theorem

In [82]:
```
f(x)=sin(x)
a,n = 0,5
tn(x) = f.taylor(x,a,n)
show(tn(x))
```

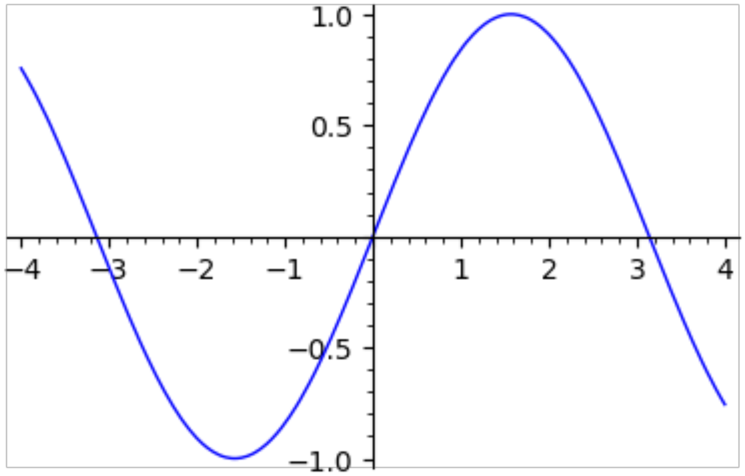
$$\frac{1}{120}\,x^5 - \frac{1}{6}\,x^3 + x$$

In [88]:
```
f(x)=sin(x)
a,n = pi,5
gn(x) = f.taylor(x,a,n)
show(gn(x))
```

$$\pi + \frac{1}{120}\,(\pi - x)^5 - \frac{1}{6}\,(\pi - x)^3 - x$$

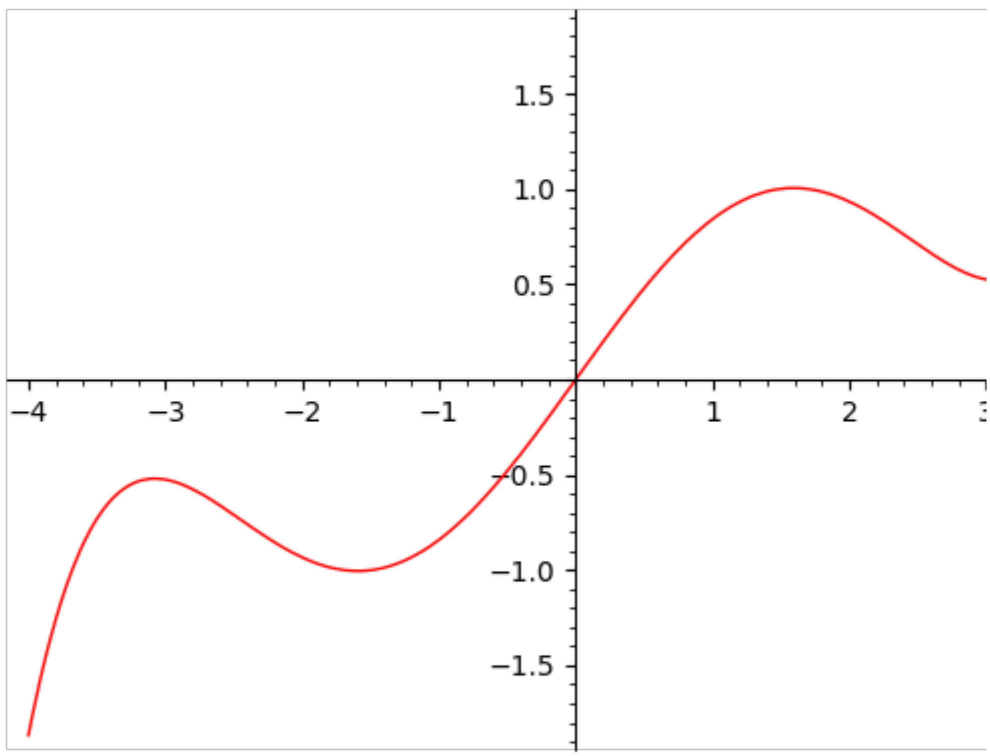
In [89]:
```
pf=f.plot(-4,4,figsize=4)
pf
```

Out[89]:



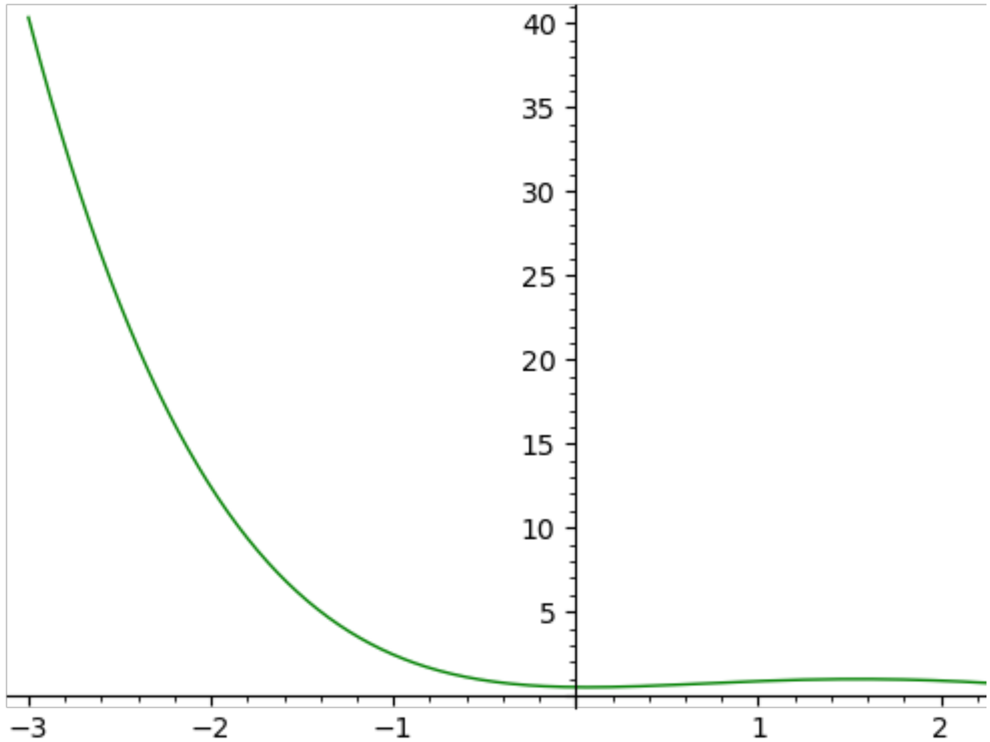
In [90]:
```
ptn=tn.plot(-4,4,color='red')
ptn
```

Out[90]:

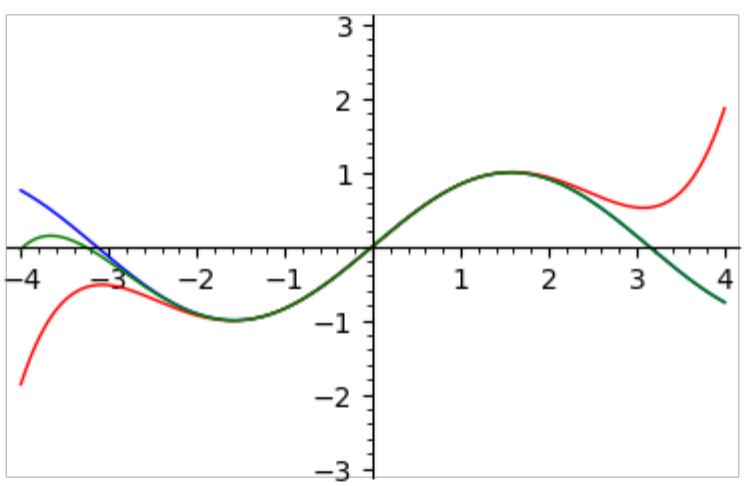


In [91]:
```
ptn1=gn.plot(-3,3,color='green')
ptn1
```

Out[91]:



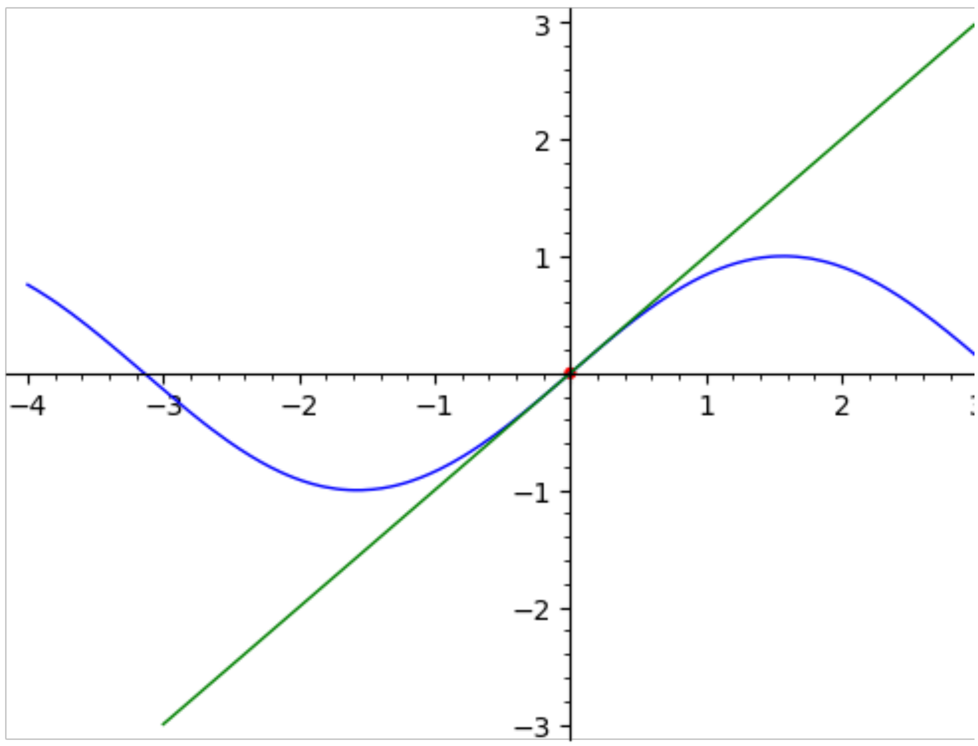
In [92]:
```
a,n = 0,5
b,m = pi,15
tn(x) = f.taylor(x,a,n)
gn(x) = f.taylor(x,b,m)
pf=f.plot(-4,4,figsize=4)
ptn=tn.plot(-4,4,color='red')
ptn1= gn.plot(-4,4,color='green')
show(pf+ptn+ptn1,figsize=4,ymax=3,ymin=-3)
```



In [93]:
```
var('x')
x0 = 0
f = sin(x)
@interact
def _(order=[1..10],x0=(0,(0,2))):
    tn = f.taylor(x,x0,order)
    p = plot(f,-4,4, thickness=1)
    dot = point((x0,f(x=x0)),pointsize=20,rgbcolor=(1,0,0))
    pt = plot(tn,x0-3,x0+3, color='green', thickness=1)
    show(dot + p + pt, ymin = -3,ymax=3)
```

order 1

x0 0

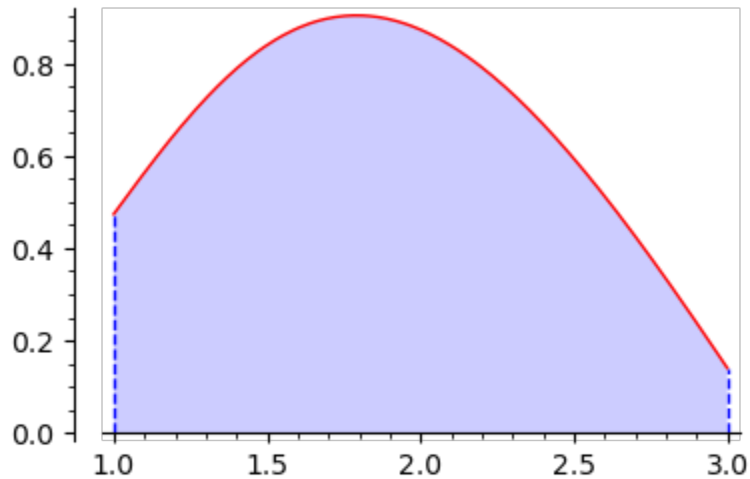


Applications of Integrals

Areas Problem: Find the area under the curve f(x) = sinx−xe^(−x^2) between x = the x-axis.

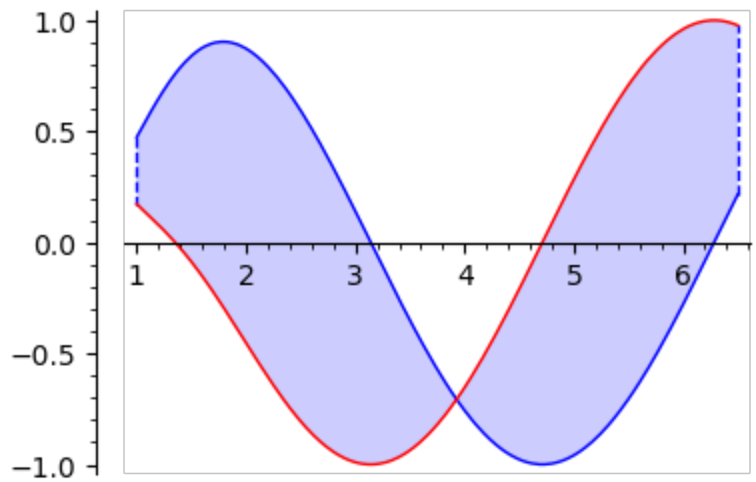
In [94]:
```
f(x) = sin(x)-x*exp(-x^2)
a, b = 1,3
l1 = line([(a,0),(a,f(a))],linestyle='--')
l2 = line([(b,0),(b,f(b))],linestyle='--')
p = plot(f(x),(x,a,b),color='red',fill=0,fillcolor='blue', fillal
show(f(x))
show(l1+l2+p)
A= integrate(f(x),x,a,b).n()
print(f"The are a under f and x-axis and between {a} and {b} is {
```

$-xe^{\left(-x^2\right)} + \sin(x)$




```
The are a under f and x-axis and between 1 and 3 is 1.34641678678
```

In [95]:
```
f(x) = sin(x)-x*exp(-x^2)
g(x) = cos(x)-x*exp(-x^2)
a, b = 1,6.5
l1 = line([(a,f(a)),(a,g(a))],linestyle='--')
l2 = line([(b,f(b)),(b,g(b))],linestyle='--')
p1 = plot(f(x),(x,a,b),fill=g(x),fillcolor='blue', fillalpha=0.2)
p2 = plot(g(x),(x,a,b),color='red')
show(l1+l2+p1+p2,figsize=4)
A= integrate(f(x)-g(x),x,a,b).n()
print(f"The area under f and g between {a} and {b} is {A}")
```




```
The area under f and g between 1 and 6.50000000000000 is 0.190065
```

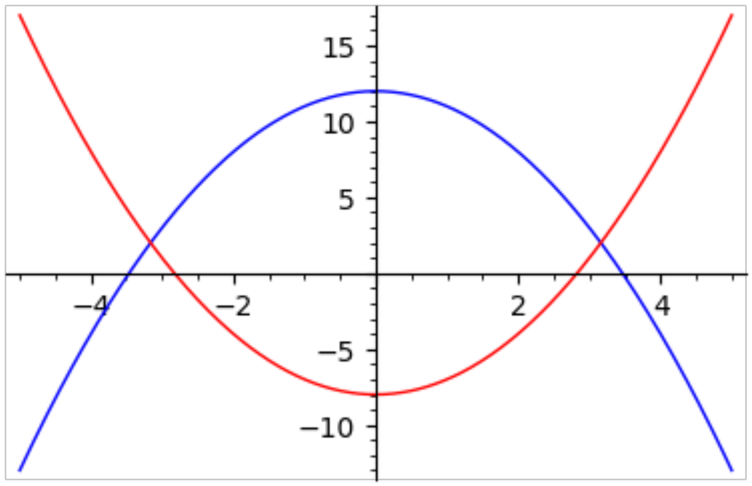
In [96]:
```
#correct Ans
c1 = find_root(f(x)-g(x),1,6.5)
integrate(f(x)-g(x),x,a,c1).n()+integrate(g(x)-f(x),x,c1,b).n()
```

Out[96]:
```
5.40190802923807
```

Find the area eclosed between two curves y = 12−x^2 and y = x^2−8

In [97]:
```
f(x)=12-x^2
g(x) = x^2-8
plot(f(x),-5,5)+plot(g(x),-5,5,color='red',figsize=4)
```

Out[97]:



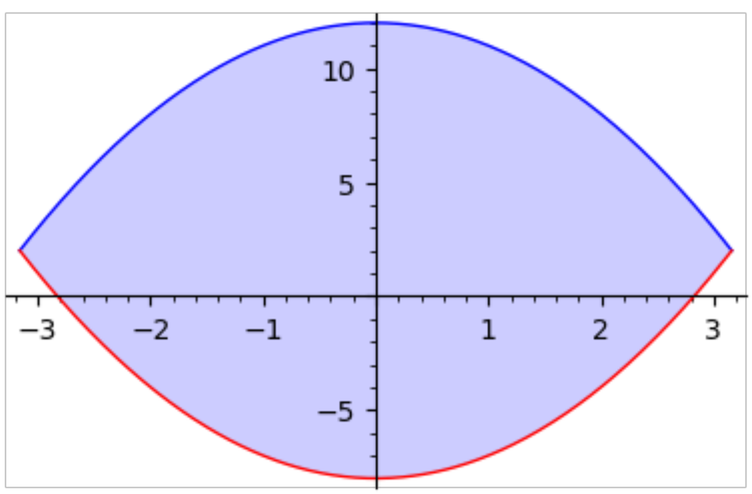
In [98]:
```
solve(f(x)==g(x),x,solution_dict=True)
```

Out[98]:
```
[{x: -sqrt(10)}, {x: sqrt(10)}]
```

In [99]:
```
 S = solve(f(x)==g(x),x,solution_dict=True)
a,b = S[0][x],S[1][x]
a,b
```

Out[99]:
```
(-sqrt(10), sqrt(10))
```

In [100]:
```
p1 = plot(f(x),(x,a,b),fill=g(x),fillcolor='blue', fillalpha=0.2)
p2 = plot(g(x),(x,a,b),color='red')
show(p1+p2,figsize=4)
```



In [101]:
```
A = integral(f(x)-g(x),x,a,b)
show(A.n())
```

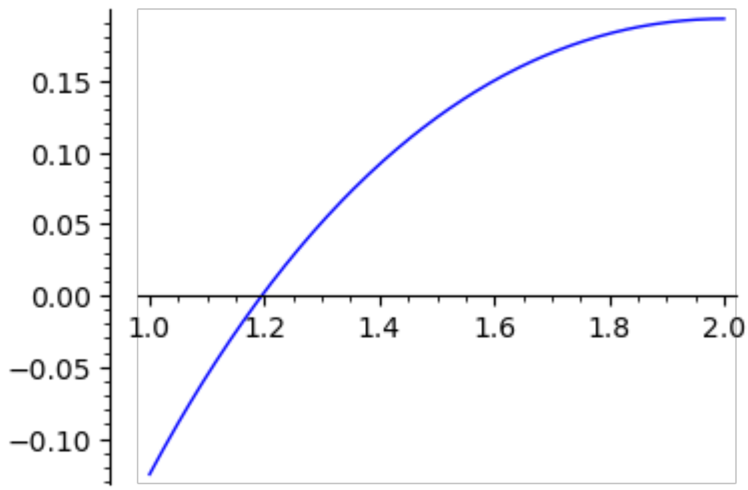
84.3274042711568

Arc Length (Rectification)

Find the arc length of the curve y = log(x)−x^2/8, 1≤ x ≤2.

In [102]:
```
f(x)=ln(x)-x^2/8
plot(f(x),(x,1,2),figsize=4)
```

Out[102]:



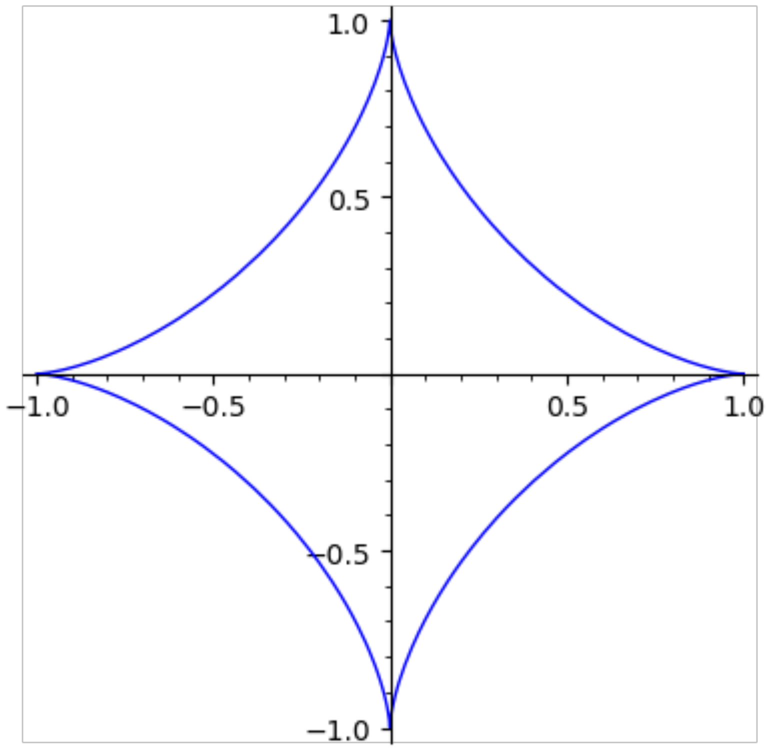
In [103]:
```
integral(sqrt(1+derivative(f,x)^2),x,1,2).n()
```

Out[103]:
```
1.06814718055995
```

Example: Find the arc length of the curve x = cos^3 t,y = sin^3 t, 0≤t≤2π.

In [104]:
```
var('t')
f(t) = cos(t)^3
g(t) = sin(t)^3
parametric_plot((f(t),g(t)),(t,0,2*pi))
```

Out[104]:



In [105]:
```
f1(t)= f.diff()(t)
g1(t)= g.diff()(t)
S = f1(t)^2+g1(t)^2
S.trig_simplify()
```

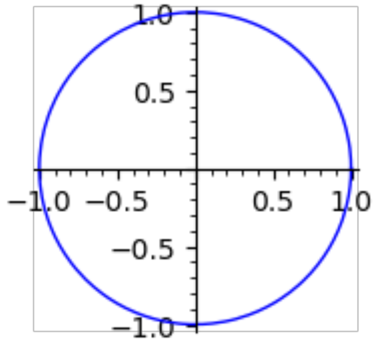
Out[105]: -9*cos(t)^4 + 9*cos(t)^2

In [106]:
```
L = 4*numerical_integral(sqrt(S),0,pi/2)[0]
L
```

Out[106]: 5.999999999999998

In [107]:
```
var('theta,r')
r(theta)=1
polar_plot(r, 0, 2*pi,figsize=3)
```

Out[107]:



In [108]:
```
dr = r.diff()
L = integral(sqrt(r^2+dr^2),theta,0,2*pi)
L
```

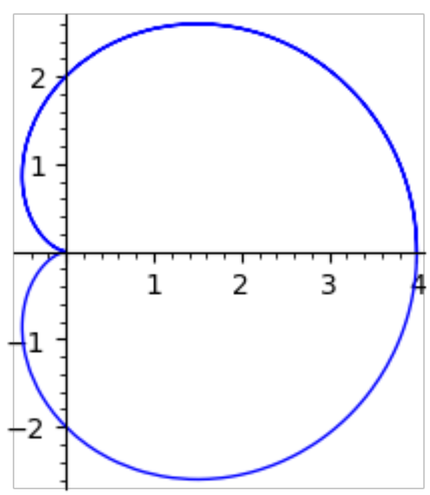
Out[108]: 2*pi

Area

Find the area enclosed by the limacon r = 2+2cosθ, 0≤ θ ≤2π.

In [109]:
```
var('theta')
r = 2 + 2*cos(theta)
polar_plot(r, 0, 3*pi,figsize=4)
```

Out[109]:

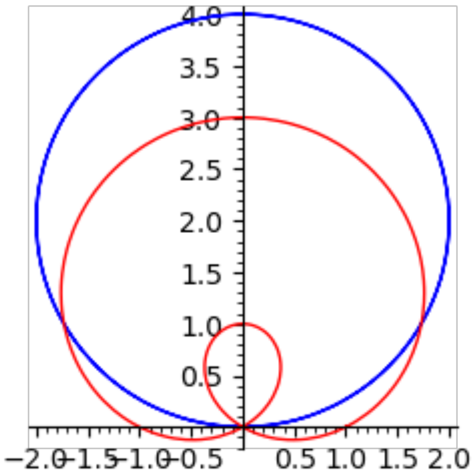


In [110]:
```
A = integral(1/2*r^2,theta, 0, 2*pi)
A
```

Out[110]: 6*pi

Problem: Find the area of the region inside the circle r = 4sinθ and outside the ca
2sinθ−1.

In [111]:
```
r1 = 4*sin(theta)
r2 = 2*sin(theta)+1
c1 = polar_plot(4*sin(theta), 0, 2*pi,figsize=4)
c2 = polar_plot(2*sin(theta)+1, 0, 2*pi,color='red',figsize=4)
show(c1+c2)
```



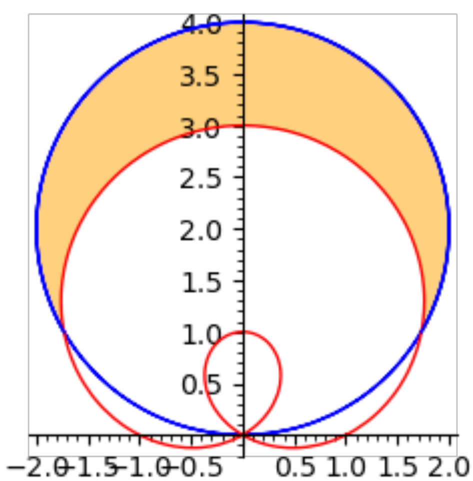
In [112]:
```
solve(4*sin(theta)==2*sin(theta)+1,theta)
```

Out[112]: [theta == 1/6*pi]

In [113]:
```
theta1 = pi-pi/6
theta2 = pi/6
```

In [114]:
```
c1+c2+polar_plot(r1, theta2,theta1, fill=r2, fillcolor='orange')
```

Out[114]:



In [115]:
```
A = 1/2*integral(r1^2-r2^2,theta, theta2,theta1)
A.n()
```

Out[115]: 4.36996235219855

Differential equation

Solve the differential equaton dy/dx = x(1−x^2)/ y(2−y)

In [116]:
```
var('x,y')
f(x,y)=(x*(1-x^2))/(y*(2-y))
x=var('x')
y = function('y')(x)
de = diff(y,x)==f(x,y)
sol=desolve(de,y)
show(sol)
```

$$\frac{1}{3}\,y(x)^3 - y(x)^2 = \frac{1}{4}\,x^4 - \frac{1}{2}\,x^2 + C$$

In [117]:
```
x=var('x')
y = function('y')(x)
sol=desolve(5*diff(y,x,2)+4*diff(y,x)+17*y == 0,y)
show(sol)
```

$$\left(K_2 \cos\left(\frac{9}{5}\,x\right) + K_1 \sin\left(\frac{9}{5}\,x\right)\right)e^{\left(-\frac{2}{5}\,x\right)}$$

In [118]:
```
x=var('x')
y = function('y')(x)
sol=desolve(5*diff(y,x,2)+4*diff(y,x)+17*y == 0,y,[0,-1,2])
show(sol)
```

$$-\frac{1}{9}\left(9\,\cos\left(\frac{9}{5}\,x\right) - 8\,\sin\left(\frac{9}{5}\,x\right)\right)e^{\left(-\frac{2}{5}\,x\right)}$$

Matrix

In [119]:
```
A = matrix([[1,2,3,4],[4,3,2,1,],[6,17,8,9],[9,8,17,6]])
show(A)
```

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \\ 6 & 17 & 8 & 9 \\ 9 & 8 & 17 & 6 \end{pmatrix}$$

In [120]:
```
show(A.characteristic_polynomial())
A1 = A.echelon_form()
A1
```

$$x^4 - 18x^3 - 150x^2 - 100x - 1500$$

Out[120]:
```
[ 1  2  3  4]
[ 0  5  0 15]
[ 0  0 10  0]
[ 0  0  0 30]
```

In [121]:
```
 A.rref()
```

Out[121]:
```
[1 0 0 0]
[0 1 0 0]
[0 0 1 0]
[0 0 0 1]
```

In [122]:
```
show(A.transpose())
```

$$\begin{pmatrix} 1 & 4 & 6 & 9 \\ 2 & 3 & 17 & 8 \\ 3 & 2 & 8 & 17 \\ 4 & 1 & 9 & 6 \end{pmatrix}$$

In [123]:
```
show(A.det())
```

$$-1500$$

In [124]:
```
A.dimensions()
```

Out[124]:
```
(4, 4)
```

In [125]:
```
A.diagonal()
```

Out[125]:
```
[1, 3, 8, 6]
```

In [126]:
```
A.adjugate()
```

Out[126]:
```
[-150 -600  100   50]
[ 300  150 -150    0]
[ 150  300    0 -150]
[-600 -150   50  100]
```

In [127]:
```
A.inverse()
```

Out[127]:
```
[ 1/10   2/5 -1/15 -1/30]
[ -1/5 -1/10  1/10     0]
[-1/10  -1/5     0  1/10]
[  2/5  1/10 -1/30 -1/15]
```

In [128]:
```
(1/(A.det()))*(A.adjugate())
```

Out[128]:
```
[ 1/10   2/5 -1/15 -1/30]
[ -1/5 -1/10  1/10     0]
[-1/10  -1/5     0  1/10]
[  2/5  1/10 -1/30 -1/15]
```

In [129]:
```
D,P = A.eigenmatrix_right()
show(D,P)
```

$$\begin{pmatrix} -6.776756183823781? & 0 & \\ 0 & 24.41467974370986? & \\ 0 & 0 & 0.181038220056 \\ 0 & 0 & \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 & \\ -0.5247290581069334? & 0.6130034396511035? & -0.284137190772 \\ 2.249578064465175? & 2.864083300655234? & -0.7568306825602 \\ -3.369008065251360? & 3.399105740610487? & 0.50495116232 \end{pmatrix}$$

In [130]:
```
A=matrix([[4,-1,6],[2,1,6],[2,-1,8]])
show(A)
show(A.eigenspaces_right())
show(A.eigenvalues())
```

$$\begin{pmatrix} 4 & -1 & 6 \\ 2 & 1 & 6 \\ 2 & -1 & 8 \end{pmatrix}$$

$$\left[\left(9, \mathrm{RowSpan}_{\mathbf{Q}}\begin{pmatrix} 1 & 1 & 1 \end{pmatrix}\right), \left(2, \mathrm{RowSpan}_{\mathbf{Q}}\begin{pmatrix} 1 & 0 & -\frac{1}{3} \\ 0 & 1 & \frac{1}{6} \end{pmatrix}\right)\right]$$

$$[9, 2, 2]$$

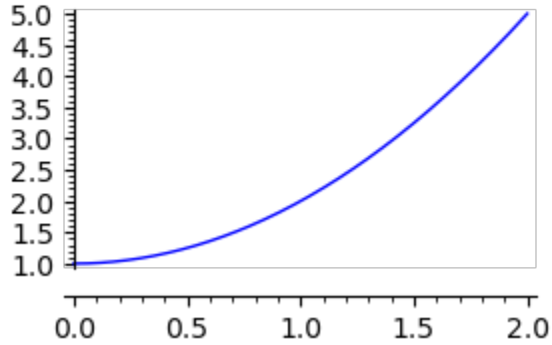
In [131]:
```
D,P = A.eigenmatrix_right()
show(D,P)
```

$$\begin{pmatrix} 9 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix}\begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & -\frac{1}{3} & \frac{1}{6} \end{pmatrix}$$

Surface of revolution

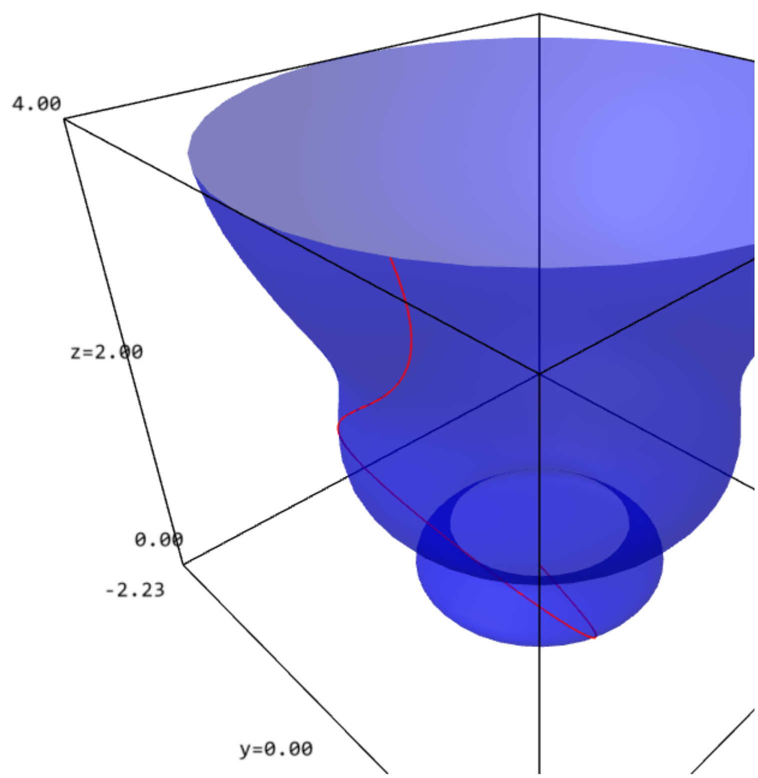
In [132]:
```
var('u')
f = 1+u^2
fp = plot(f,(u,0,2),figsize=3)
fp
```

Out[132]:



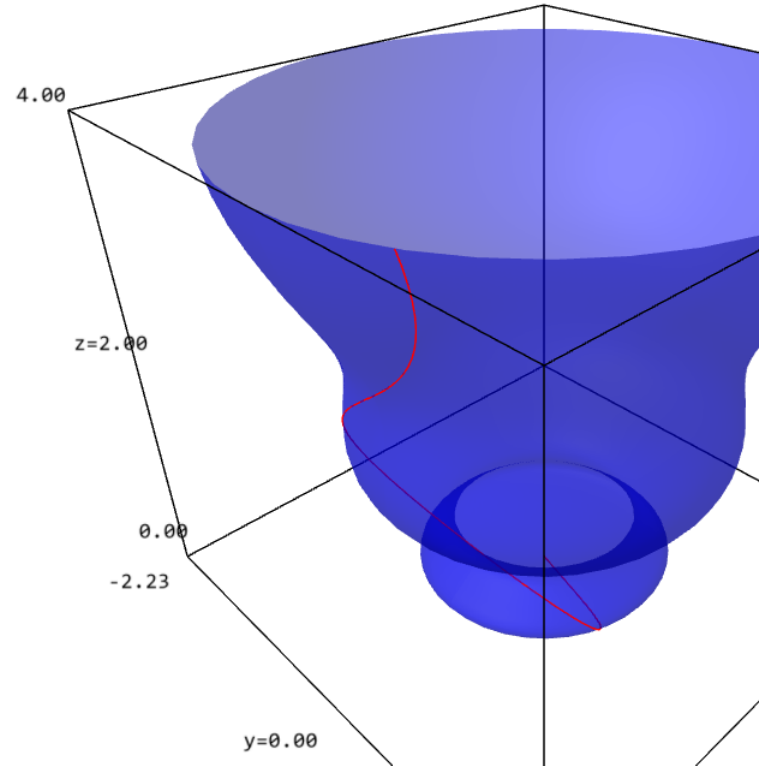
In [133]:
```
revolution_plot3d(f, (u,0,2),parallel_axis='x',show_curve=True,f
```

Out[133]:



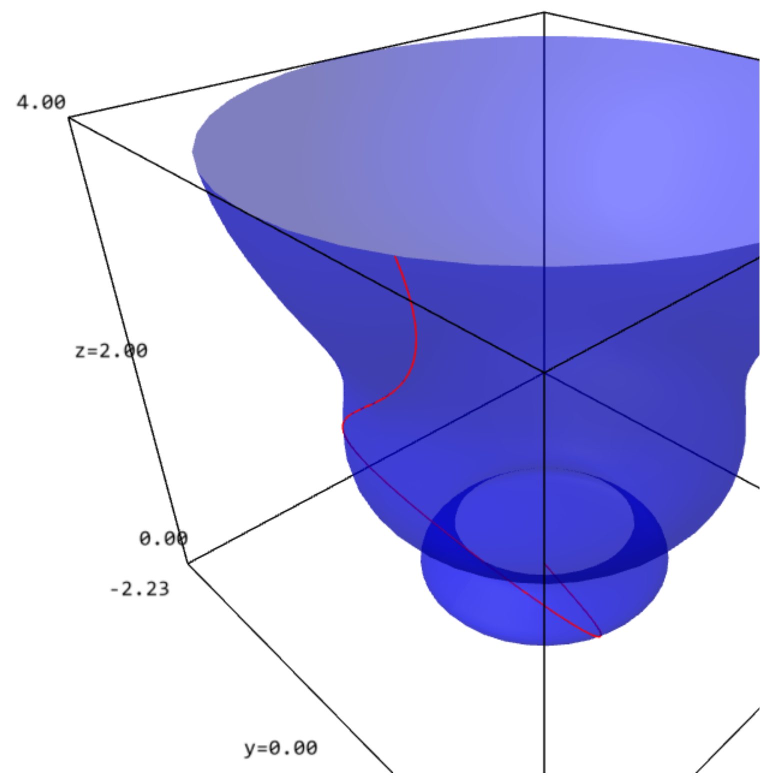
In [134]:
```
revolution_plot3d(f, (u,0.5,2),parallel_axis='z',show_curve=True
```

Out[134]:



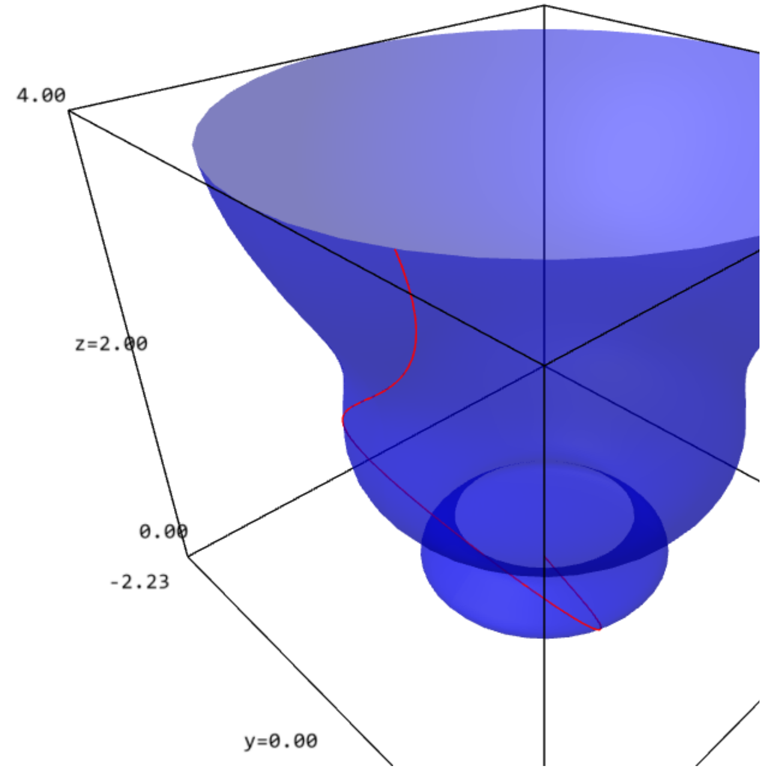
In [135]:
```
revolution_plot3d(f, (u,0,2),parallel_axis='y',show_curve=True,fr
```

Out[135]:

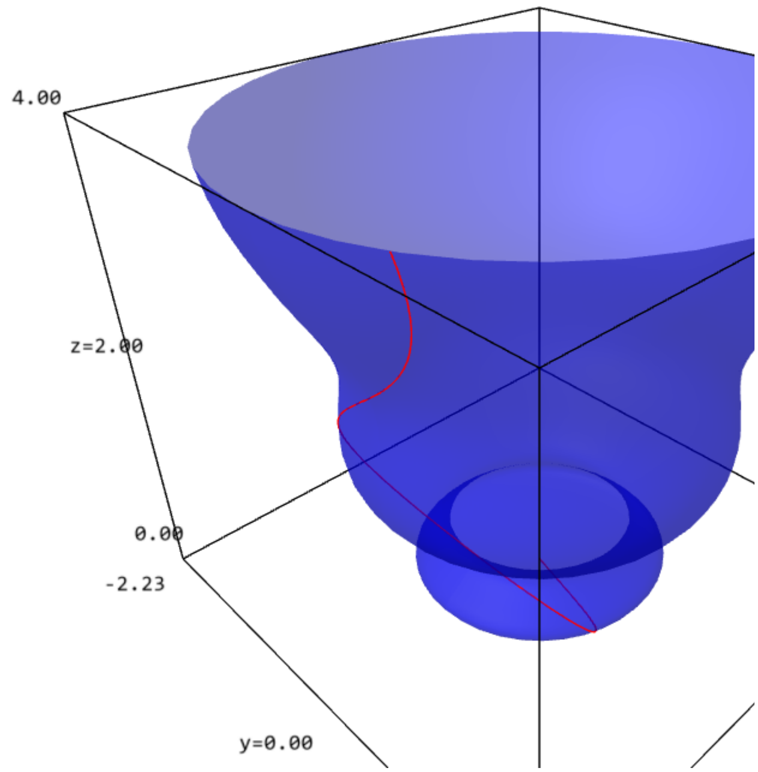


In [136]:
```
revolution_plot3d(u, (u,0,2),parallel_axis='x',show_curve=True,fi
```

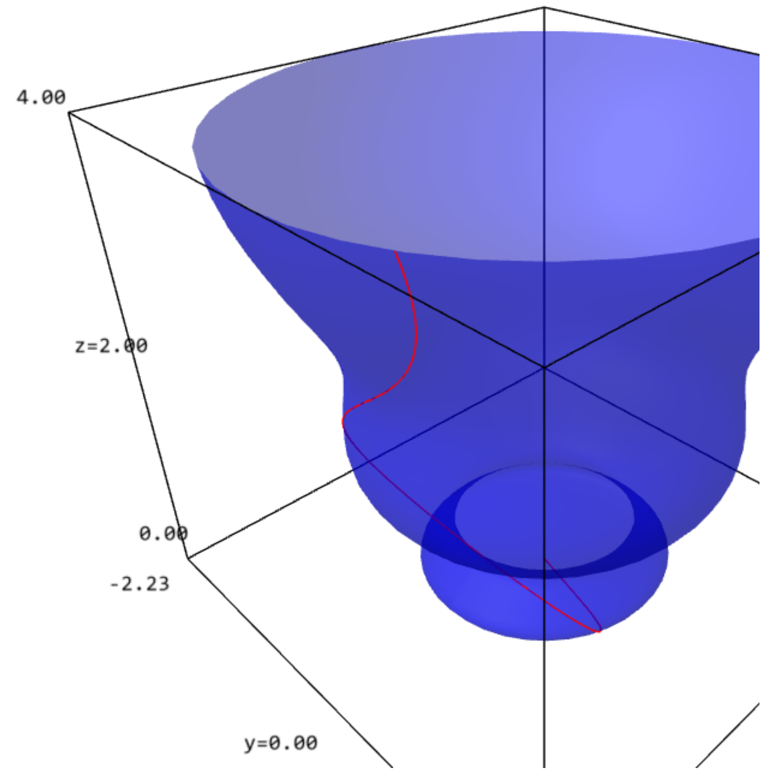
Out[136]:



In [137]:
```
u = var('u')
curve = (u, sin(4*u), u^2)
curve3d = parametric_plot3d(curve,(u,0,2))
show(curve3d)
```



In [138]:
```
P = revolution_plot3d(curve, (u,0,2),(0, 2*pi),show_curve=True, p
show(P,aspect_ratio=1)
```



Number Theroy

In [139]:
```
a = 6543
```

In [140]:
```
a.divisors()
```

Out[140]: [1, 3, 9, 727, 2181, 6543]

In [141]:
```
a.factor()
```

Out[141]: 3^2 * 727

In [142]:
```
a.is_prime()
```

Out[142]: False

In [143]:
```
11.is_prime()
```

Out[143]: True

In [144]:
```
a.is_perfect_power()
```

Out[144]: False

In [145]:
```
36.is_perfect_power()
```
Out[145]:
```
True
```

In [146]:
```
a.is_integer()
```
Out[146]:
```
True
```

In [147]:
```
a.next_prime()
```
Out[147]:
```
6547
```

In [148]:
```
a.previous_prime()
```
Out[148]:
```
6529
```

In [149]:
```
a.digits()
```
Out[149]:
```
[3, 4, 5, 6]
```

In [150]:
```
a, b = 78 , 18
```

In [151]:
```
a//b
```
Out[151]:
```
4
```

In [152]:
```
a/b
```
Out[152]:
```
13/3
```

In [153]:
```
a/b.n()
```
Out[153]:
```
4.33333333333333
```

In [154]:
```
a%b
```
Out[154]:
```
6
```

In [155]:
```
q , r =a.quo_rem(b)
print(q)
print(r)
```
```
4
6
```

In [156]:
```
gcd(a,b)
```
Out[156]:
```
6
```

In [157]:
```
a.gcd(b)
```
Out[157]:
```
6
```

In [158]:
```
d, p,q = xgcd(a,b)
```

In [159]:
```
d, p,q
```

Out[159]:
```
(6, 1, -4)
```

In [160]:
```
d==p*a +q*b
```

Out[160]:
```
True
```

In [161]:
```
prime_pi(5)
```

Out[161]:
```
3
```

In [162]:
```
prime_pi(a)
```

Out[162]:
```
21
```

In [163]:
```
[i for i in range(79) if gcd(i,a)==1]
```

Out[163]:
```
[1,
 5,
 7,
 11,
 17,
 19,
 23,
 25,
 29,
 31,
 35,
 37,
 41,
 43,
 47,
 49,
 53,
 55,
 59,
 61,
 67,
 71,
 73,
 77]
```

In [164]:
```
len([i for i in range(79) if gcd(i,a)==1])
```

Out[164]:
```
24
```

In [165]:
```
c=a.factorial()
c
```

Out[165]:
```
1132428117820629783145752115873204622873174957948825199004896282520076624508621317734400000000000000000000
```

In [166]:
```
c.ndigits()
```

Out[166]:
```
116
```

In [ ]:

In [ ]: