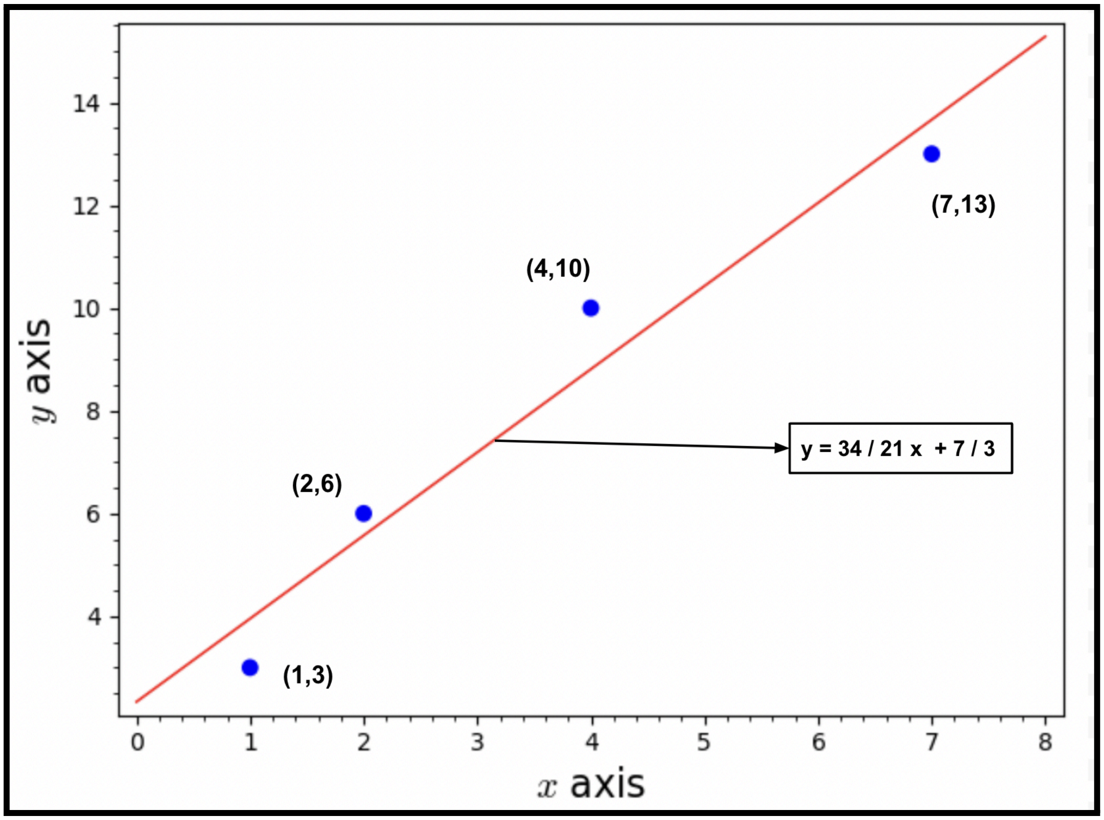
Constrained Optimization Using Lagrange Multipliers

```
var('x')
best_fit = (34/21)*x + 7/3
show('y =',best_fit)
P1 = plot(best_fit,x,0,8, frame = True,axes_labels=['$x$ axis','$y$
axis'],axes=False, color = 'red')
Q1 = point((1,3), size=50)
Q2 = point((2,6), size=50)
Q3 = point((4,10), size=50)
Q4 = point((7,13), size=50)

show(P1 + Q1 + Q2 + Q3 + Q4)
```



From the above visualization, we can see that all four points align well with the line of best fit. A condition for a good line of best fit is that the number of points should be close to the line of best fit if not on the line of best fit and there should be an approximately equal number of points on either side of the line of best fit. In this case, both conditions are fulfilled hence, we can say that our data indicates a line of best fit.

Given that d_i consists of similar constants m and b as $y = mx + b$ and we need to find these two constants we can reduce the two equations to

$$f(m, b) = \Sigma_{i=1}^{n}(yi - mxi - b)^2$$

Obj Function: $Minimize(f(m, b)$

$$\nabla f = 0$$

We will find partial derivative with respect to m

$$\nabla f = < f_m, f_b >$$

$$f_m = \Sigma_{i=1}^{n} 2(yi - mxi - b)(-xi)$$

$$f_m = -2\ \Sigma_{i=1}^{n}(yi - mxi - b)(xi)$$

Setting partial $partial\ derivatives = 0$ to find equations

$$f_m = -2\ \Sigma_{i=1}^{n}(yi - mxi - b)(xi) = 0$$

We will split the sum and take constants as common

$$\Sigma^{n}_{i=1} x_i y_i \; - \; m \, \Sigma^{n}_{i=1} x_i^{\,2} \; - \; b \, \Sigma^{n}_{i=1} x_i = \; 0$$

$$\Sigma^{n}_{i=1} x_i y_i \; = \; m \, \Sigma^{n}_{i=1} x_i^{\,2} \; + \; b \, \Sigma^{n}_{i=1} x_i \qquad (eq\ 1)$$

We will find a partial derivative with respect to b

$$f_b \; = \; \Sigma^{n}_{i=1} 2(yi \; - \; mxi \; - \; b)(-\,1)$$

$$f_b \; = \; -\,2\,\Sigma^{n}_{i=1}(yi \; - \; mxi \; - \; b)$$

Setting partial $partial\ derivatives \; = \; 0$ to find equations

$$f_b \; = \; -\,2\,\Sigma^{n}_{i=1}(yi \; - \; mxi \; - \; b) \; = \; 0$$

$$\Sigma^{n}_{i=1}(yi \; - \; mxi \; - \; b) \; = \; 0$$

$$\Sigma^{n}_{i=1} yi \; - \; m\,\Sigma^{n}_{i=1} xi \; - \Sigma^{n}_{i=1} b \; = \; 0$$

$$\Sigma^{n}_{i=1} yi \; = \; m\,\Sigma^{n}_{i=1} xi \; + \Sigma^{n}_{i=1} b$$

$$\Sigma^{n}_{i=1} yi \; = \; m \, \Sigma^{n}_{i=1} xi \; + \; nb \qquad (eq\ 2)$$

Therefore, the equations for maxima or minima are

$$\Sigma^{n}_{i=1} x_i y_i = m \Sigma^{n}_{i=1} x_i^{2} + b \Sigma^{n}_{i=1} x_i \qquad (eq\ 1)$$

$$\Sigma^{n}_{i=1} yi = m \Sigma^{n}_{i=1} xi + nb \qquad (eq\ 2)$$

To find whether this is maxima or minima, we will find the second partial derivative of f.

$$f_{mm} = 2 \Sigma^{n}_{i=1} x_i^{2}$$

$$f_{bb} = \Sigma^{n}_{i=1} 2 = 2n$$

$$f_{mb} = 2 \Sigma^{n}_{i=1} x_i$$

As $D(m, b) = f_{mm} \cdot f_{bb} - [f_{mb}]^{2}$

$$= 4n \Sigma^{n}_{i=1} x_i^{2} - [2 \Sigma^{n}_{i=1} x_i]^{2}$$

$$= 4 [n \Sigma^{n}_{i=1} x_i^{2} - (\Sigma^{n}_{i=1} x_i)^{2}]$$

Given that

$$n \Sigma^{n}_{i=1} x_i^{2} > (\Sigma^{n}_{i=1} x_i)^{2}]$$

$$D(m, b) > 0 \ and\ f_{mm} > 0 \ always.$$

Thus the solution of the equation minimizes the function,

For

$$\Sigma^{n}_{i=1} x_i y_i = m \Sigma^{n}_{i=1} x_i^{2} + b \Sigma^{n}_{i=1} x_i \ (eq\ 1)$$

$$\Sigma^{n}_{i=1} yi = m\ \Sigma^{n}_{i=1} xi + nb\ (eq\ 2)$$

Lagrange Multipliers

```
var('x,y,L')

# defining the objective and constraint functions

f(x,y) = 3*x + 4*y #objective function (maximize)
g(x,y) = x^(1/2) + y^(1/2) #constraint


# Lagrange Multiplier Method
gradf = f.gradient()
show('grad(f) =', gradf)
gradg = g.gradient()
show('grad(g) =',gradg)
s = [gradf()[i] == L*gradg()[i] for i in range(2)] #2
equations for x and y
s.append(g(x,y) == 7)
solution_set = solve(s,x,y,L)
show(solution_set)
```

$$\texttt{grad(f)} = (x, y) \mapsto (3, 4)$$

$$\texttt{grad(g)} = (x, y) \mapsto \left(\frac{1}{2\sqrt{x}}, \frac{1}{2\sqrt{y}}\right)$$

$$[[L = 24, x = 16, y = 9]]$$

Given that it is a Lagrange Multiplier, it only provides local maxima and minima.

We can do some calculations to confirm if we are getting global extrema.

Now we can use second partial derivatives tests using a Hessian matrix to confirm if we have a global maximum or minimum.

```
var('x,y,L')

# defining the objective and constraint functions
L = 24

f(x,y) = 3*x + 4*y #objective function (maximize)
g(x,y) = x^(1/2) + y^(1/2) #constraint

extrema = f - L*g
show(extrema.hessian())
```

$$\boxed{\begin{pmatrix} (x,y) \mapsto \frac{6}{x^{\frac{3}{2}}} & (x,y) \mapsto 0 \\ (x,y) \mapsto 0 & (x,y) \mapsto \frac{6}{y^{\frac{3}{2}}} \end{pmatrix}}$$

We can now use the second partial derivative test:

$D(x\,,\,y) \;=\; f_{xx} \cdot f_{yy} \;-\; [f_{xy}]^2$

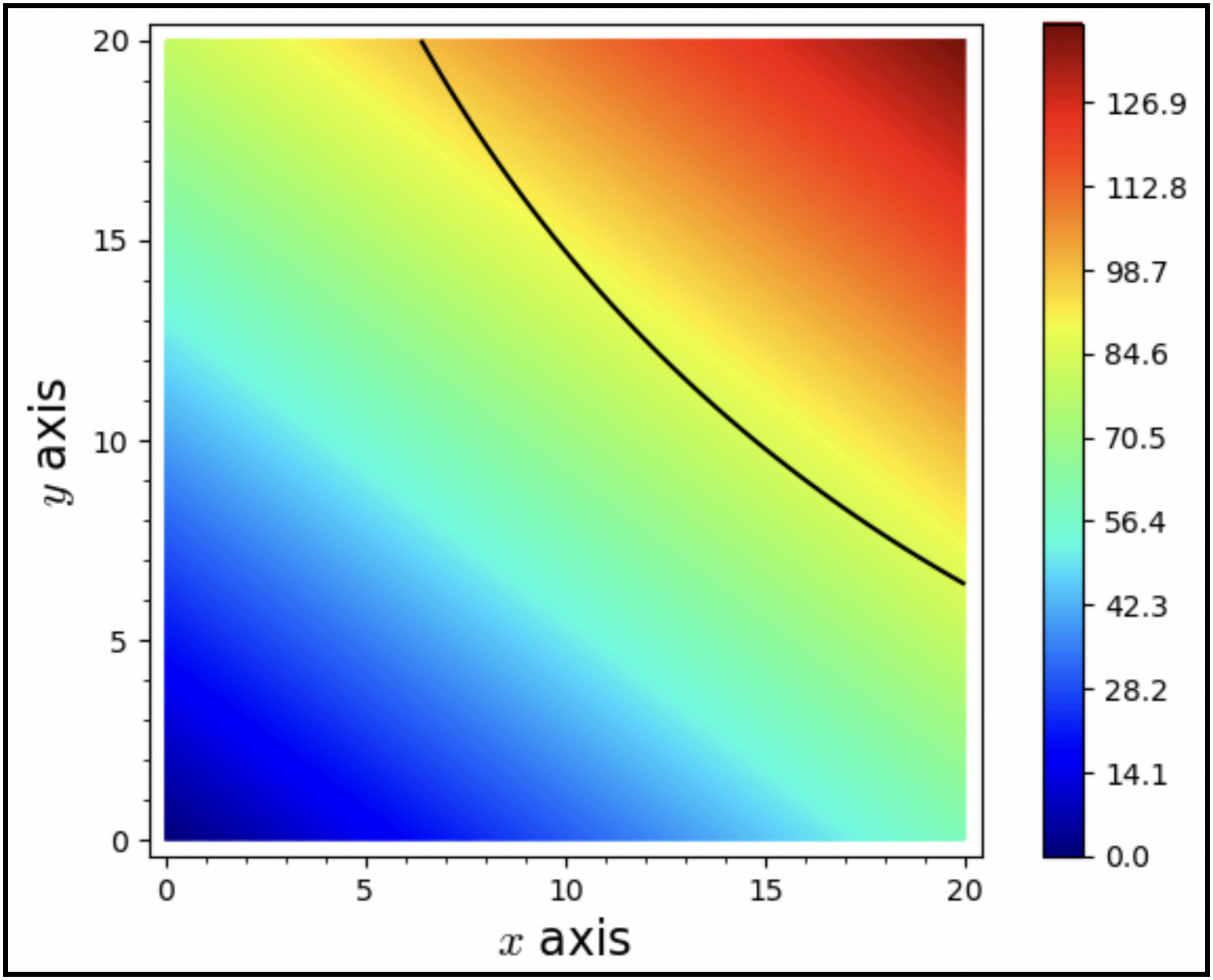
Here we can that our critical points for x and y values (16,9)

$6/16^{3/2} .\, 6/9^{3/2} \;-\; 0$

$D(x, y) \;=\; 0.2$

Given that $D(x, y) \;>\; 0 \;\; and \; f_{xx} \;>\; 0$ we can approximately say that we have a global minimum at (16,9). However, this is just an approximation and we need to check extreme points to make a final conclusion.

```
# contour plot
P1 = contour_plot(f(x,y), (x,0,20), (y,0, 20), contours = 1000, fill
= False, cmap = 'jet', colorbar = True, axes_labels=['$x$ axis','$y$
axis'])
P2 = contour_plot(g(x,y), (x,0,20), (y,0, 20), fill = False,
contours = [7])
show(P1+P2)
```



We can see multiple colors in this plot. Warmer colors like red indicate higher values while cooler colors like blue represent lower values. Our contour plot for the constraint function identified at x = 16 there is an overlap with the contour plot of our objective function and we can see a global minimum at x = 16 when y = 9. However, this might not mean it is the only global extrema so we need to check other extreme points.

c)

Given that the Lagrange Multiplier method only provides solutions for local extrema. We need to consider other extremes to check our global extrema.

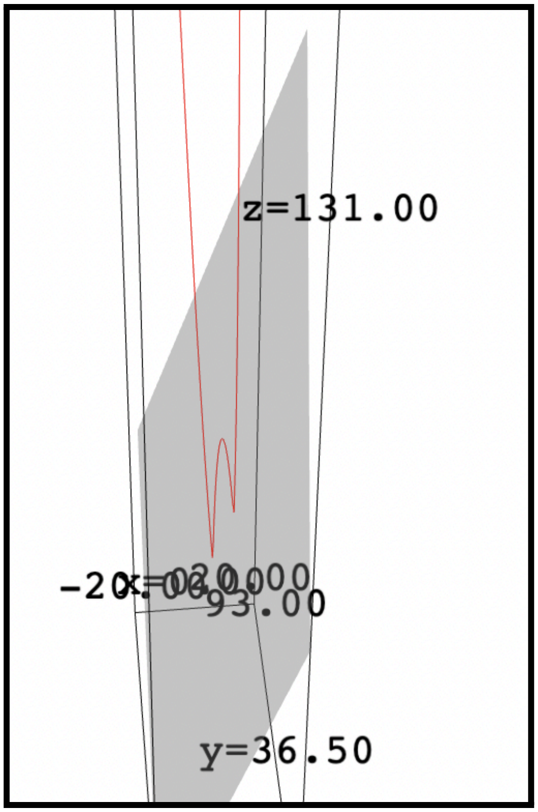
```
var('x,y,L')

# defining the objective and constraint functions
f(x,y) = 3*x + 4*y #objective
g(x,y) = x^(1/2) + y^(1/2) - 7 #constraint

# x=16, y=9
print('f(16,9) =', f(16,9), 'Global Minima')

# x=0, y=49 (extreme point)
print('f(0,49) =',f(0,49), 'Global Maxima')
print('f(49,0) =',f(49,0))

P1 = plot3d(f, (x,-20,20), (y,-20,20),viewer="threejs",
online=True, opacity=0.4, color="grey")
P2 =
parametric_plot3d([x,sqrt((7-x^2)^2),f(x,sqrt((7-x^2)^2))],
(x,-10,10),color="red")
show(P1)
```




```
f(16,9) = 84 Global Minima
f(0,49) = 196 Global Maxima
f(49,0) = 147
```

Through the parametric plot above we can see that there are three points where the constraint touches the function. Three are shown on the curve near the plane out of which the Lagrange multiplier method missed two. Therefore, we need to include the extreme values by alternatively substituting 0 in the constraint for x and y. Also, we can also consider grad(g) as zero and use its values to check for more local extrema.