2nd Year Calculus (Mr. P Solver)

Video Link: https://youtu.be/Teb28OFMVFc (https://youtu.be/Teb28OFMVFc)

Codes: https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqa2szcEtxY0l4REVhR0xiVUV5MUlyQzl4bW1Cd3xBQ3J
(https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqa2szcEtxY0l4REVhR0xiVUV5MUlyQzl4bW1Cd3xBQ3J

In [54]:
```
import numpy as np
import matplotlib.pyplot as plt
import scipy as sp
from scipy.integrate import quad, quad_vec
import sympy as smp
from sympy import *
from sympy.vector import *
```

In [55]:
```
x,y,z,t,u1,u2,u3,v1,v2,v3 = smp.symbols('x y z t u_1 u_2 u_3 v_1 v_2 v_3')
```

Vectors and Geometry

In [56]:
```
a = np.array([2,3,7])
b = np.array([2,4,1])
u = smp.Matrix([u1,u2,u3])
v = smp.Matrix([v1,v2,v3])
```

Addition and Multiplication

In [57]:
```
print(2*a +5*b)
display(2*u +5*v)
```

```
[14 26 19]
```

$$\begin{bmatrix} 2u_1 + 5v_1 \\ 2u_2 + 5v_2 \\ 2u_3 + 5v_3 \end{bmatrix}$$

Dot products

In [58]:
```
print(np.dot(a,b))
display(u.dot(v))
```

```
23
```

$u_1v_1 + u_2v_2 + u_3v_3$

Cross products

In [59]:
```
print(np.cross(a,b))
display(u.cross(v))
```

```
[-25  12   2]
```

$$\begin{bmatrix} u_2 v_3 - u_3 v_2 \\ -u_1 v_3 + u_3 v_1 \\ u_1 v_2 - u_2 v_1 \end{bmatrix}$$

Length of vector

In [60]:
```
print(np.linalg.norm(a))

unorm = u.norm()
display(unorm)
display(unorm.subs([(u1,2), (u2,3),(u3,7)]))
```

```
7.874007874011811
```

$$\sqrt{|u_1|^2 + |u_2|^2 + |u_3|^2}$$

$$\sqrt{62}$$

Vector projection

Projection of u on v,

$$\text{proj}_v(u) = (u.\,\hat{v})\hat{v} = \frac{u \cdot v}{|v|^2} v$$

In [61]:
```
projab = np.dot(a,b)*b/np.linalg.norm(b)**2
print(projab)

projuv = u.dot(v)*v/v.norm()**2
display(projuv)
```

```
[2.19047619 4.38095238 1.0952381 ]
```

$$\begin{bmatrix} \frac{v_1(u_1 v_1 + u_2 v_2 + u_3 v_3)}{|v_1|^2 + |v_2|^2 + |v_3|^2} \\ \frac{v_2(u_1 v_1 + u_2 v_2 + u_3 v_3)}{|v_1|^2 + |v_2|^2 + |v_3|^2} \\ \frac{v_3(u_1 v_1 + u_2 v_2 + u_3 v_3)}{|v_1|^2 + |v_2|^2 + |v_3|^2} \end{bmatrix}$$

Lines

$$\vec{r}(t) = \vec{r}_0 + t\vec{v}$$

In [62]:
```
r0 = smp.Matrix([1,1,1])
v = smp.Matrix([4,3,4])
r = r0 + t*v
r
```

Out[62]:
$$\begin{bmatrix} 4t+1 \\ 3t+1 \\ 4t+1 \end{bmatrix}$$

Planes

$$\vec{n} \cdot (P_0 - \langle x, y, z \rangle) = 0$$

In [63]:
```
n = smp.Matrix([3,2,3])
P0 = smp.Matrix([2.2,3,2])
r = smp.Matrix([x,y,z])
n.dot(P0 - r)
```

Out[63]: $-3x - 2y - 3z + 18.6$

Example: Find unit vector parallel to the line of intersection of the two planes $3x - 6y - 2z = 15$ and $2x + y - 2z = 5$. (Hint: It's going to be perpendicular to both normal vectors)

In [64]:
```
n1 = np.array([3,-6,-2])
n2 = np.array([2,1,-2])
vec = np.cross(n1,n2)
ans = vec/np.linalg.norm(vec)
ans
```

Out[64]:
```
array([0.67909975, 0.09701425, 0.72760688])
```

Vector Calculus

Vector derivatives

In [65]:
```
r = smp.Matrix([4*t,6*smp.cos(5*t),t**3])
display(r)
diffr = smp.diff(r,t)
display(diffr)
```

$$\begin{bmatrix} 4t \\ 6\cos(5t) \\ t^3 \end{bmatrix}$$

$$\begin{bmatrix} 4 \\ -30\sin(5t) \\ 3t^2 \end{bmatrix}$$

Example: Find the angle between the velocity and acceleration as a function of time $\theta(t)$ and also find the angle at $t = 4s$. Plot t vs $\theta(t)$ graph.

In [66]:
```python
v = smp.diff(r,t)
a = smp.diff(v,t)
theta = smp.acos(v.dot(a)/(v.norm()*a.norm()))
theta.simplify()
```

Out[66]:

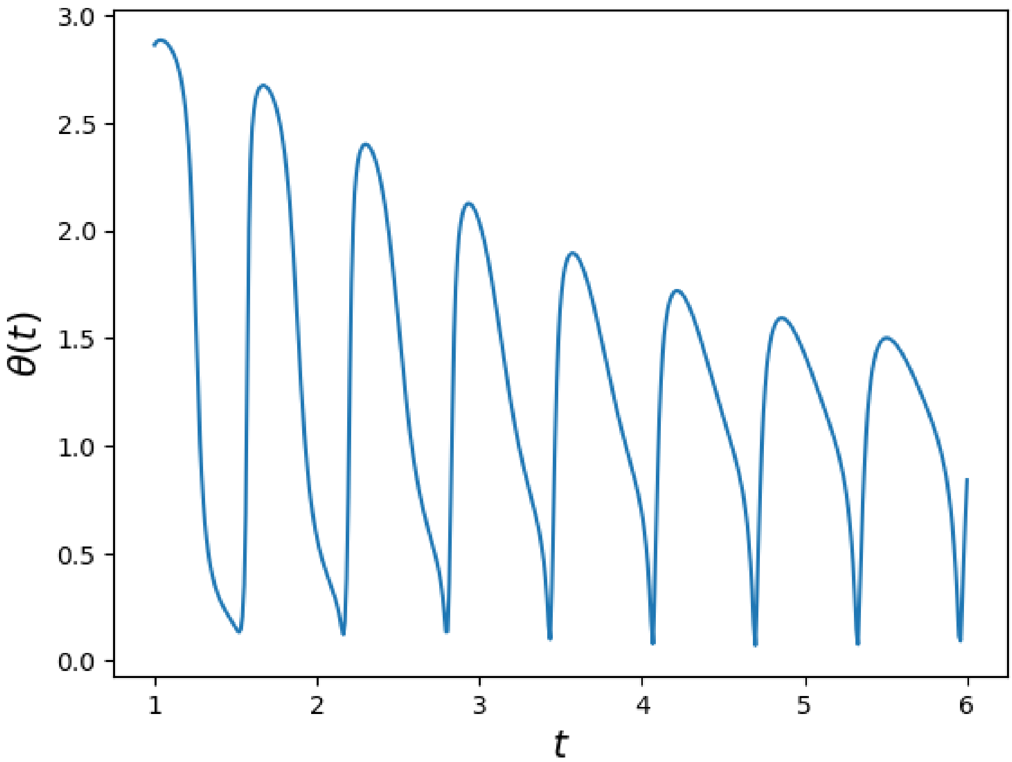
$$\operatorname{acos}\left(\frac{3\left(t^{3}+125\sin\left(10t\right)\right)}{\sqrt{\left|t\right|^{2}+625\left|\cos\left(5t\right)\right|^{2}}\sqrt{9\left|t^{2}\right|^{2}+900\left|\sin\left(5t\right)\right|^{2}+16}}\right)$$

In [67]:
```python
theta.subs(t,4).evalf()  # evalf() evaluates a float value
```

Out[67]: 0.681852695830224

In [68]:
```python
thetaf = smp.lambdify([t], theta)  # function

tt = np.linspace(1,6,500)
tht = thetaf(tt)
plt.plot(tt,tht)
plt.xlabel('$t$', fontsize=15)
plt.ylabel(r'$\theta(t)$', fontsize=15)
plt.show()
```



Vector Integrals

In [69]:
```
r = smp.Matrix([smp.exp(-t**3), smp.sin(t), 5*t**3 + 4*t])
I = smp.Integral(r,t)
display(I)
display(I.doit())  # performs the integration
```

$$\int \left[\begin{matrix}e^{-t^{3}}\\ \sin{\left(t \right)}\\ 5t^{3} + 4t\end{matrix}\right] dt$$

$$\left[\begin{matrix}\frac{\Gamma\left(\frac{1}{3}\right)\gamma\left(\frac{1}{3}, t^{3}\right)}{9\Gamma\left(\frac{4}{3}\right)}\\ - \cos{\left(t \right)}\\ \frac{5t^{4}}{4} + 2t^{2}\end{matrix}\right]$$

Some cases integrals can't be solved analytically. So we need to solve them numerically.

In [70]:
```
r1 = smp.Matrix([smp.exp(-t**2)*smp.cos(t)**3, smp.exp(-t**4), 1/(3+t**2)])
I1 = smp.Integral(r1, (t,0,1))
I1
```

Out[70]:

$$\int\limits_{0}^{1} \left[\begin{matrix}e^{-t^{2}} \cos^{3}{\left(t \right)}\\ e^{-t^{4}}\\ \frac{1}{t^{2} + 3}\end{matrix}\right] dt$$

In [71]:
```
from scipy.integrate import quad_vec

rf = smp.lambdify([t],r1)
quad_vec(rf, 0,1)  # integration and error
```

Out[71]:
```
(array([[0.53525785],
        [0.84483859],
        [0.30229989]]),
 3.5151979041265046e-14)
```

Arclength

$$L = \int_a^b \sqrt{dx^2 + dy^2 + dz^2} = \int_a^b \sqrt{(dx/dt)^2 + (dy/dt)^2 + (dz/dt)^2} dt$$

Find arclength of $\langle 0, 3t, 2t^2 \rangle$ from $t = 0$ to $t = 1$.

In [72]:
```
r2= smp.Matrix([0, 3*t, 2*t**2])
display(r2)

f1 = smp.diff(r2,t).norm()
L= smp.integrate(f1, (t,0,1))
display(L)

# numerical
f1f = smp.lambdify([t], f1)
print('numerical solution in (0,1) is,', quad(f1f,0,1)[0])
```

$$\begin{bmatrix} 0 \\ 3t \\ 2t^2 \end{bmatrix}$$

$$\frac{9\,\mathrm{asinh}\left(\frac{4}{3}\right)}{8} + \frac{5}{2}$$

```
numerical solution in (0,1) is, 3.735938824751624
```

Other Relevant Quantities

If $ds = \sqrt{dx^2 + dy^2 + dz^2}$ is the arclength element; the velocity will be $ds/dt = \sqrt{(dx/dt)^2 + (dy/dt)^2 + (dz/dt)^2} = |d\vec{r}/dt| = |\vec{v}|$. The other quantities of importance are;

1. Unit tangent vector: $\vec{T} = \frac{d\vec{r}}{dt}\frac{1}{|d\vec{r}/dt|} = \vec{v}/|\vec{v}|$
2. Curvature: $\kappa = |\frac{d\vec{T}}{dt}|\frac{1}{|\vec{v}|}$
3. Unit normal vector: $\vec{N} = \frac{d\vec{T}/dt}{|d\vec{T}/dt|}$

Example: Find all these for $\vec{r}(t) = \langle a\cos(t)e^t, b\sin(t), ct\rangle$.

In [73]:
```
t, a, b, c = smp.symbols('t a b c', pos=True, real=True)
r = smp.Matrix([a*smp.cos(t)*smp.exp(t), b*smp.sin(t), c*t])
display('path', r)
v = smp.diff(r,t)
modv = v.norm()
display('velocity vector', v, 'magnitude of velocity', modv)
```

```
'path'
```

$$\begin{bmatrix} ae^t\cos(t) \\ b\sin(t) \\ ct \end{bmatrix}$$

```
'velocity vector'
```

$$\begin{bmatrix} -ae^t\sin(t) + ae^t\cos(t) \\ b\cos(t) \\ c \end{bmatrix}$$

```
'magnitude of velocity'
```

$$\sqrt{b^2\cos^2(t) + c^2 + \left(ae^t\sin(t) - ae^t\cos(t)\right)^2}$$

Get $\vec{T}$, κ and $\vec{N}$,

In [74]:
```
T = v/modv
kappa = T.diff(t).norm()/modv
N = T.diff(t)/T.diff(t).norm()
```

In [75]:
```
print('at (t,a,b,c) = (2,3,4,5),')
display('unit tangent vector',T.subs([(t,2),(a,3),(b,4),(c,5)]).evalf(6))
display('curvature',kappa.subs([(t,2),(a,3),(b,4),(c,5)]).evalf(6))
display('unit normal vector',N.subs([(t,2),(a,3),(b,4),(c,5)]).evalf(6))
```

```
at (t,a,b,c) = (2,3,4,5),
```

```
'unit tangent vector'
```

$$\begin{bmatrix} -0.984293 \\ -0.0557647 \\ 0.167503 \end{bmatrix}$$

```
'curvature'
```

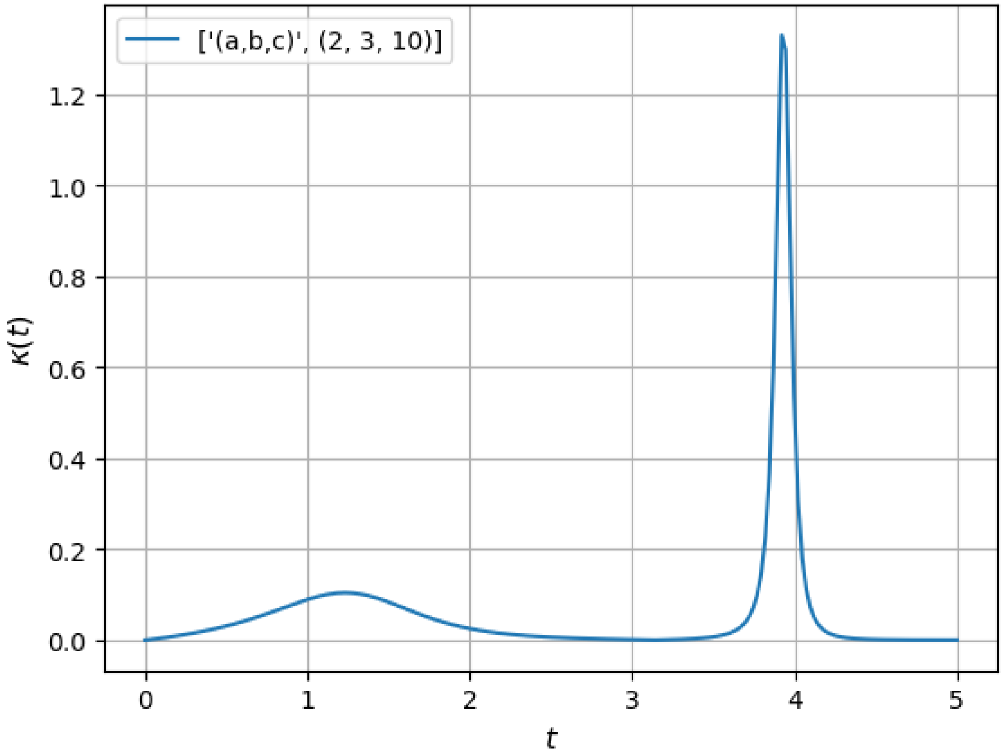
$$0.00775459$$

```
'unit normal vector'
```

$$\begin{bmatrix} -0.152946 \\ -0.204518 \\ -0.96684 \end{bmatrix}$$

Plot of the curvature:

```
In [76]: kf = smp.lambdify([t,a,b,c], kappa)
a1,b1,c1 = 2,3,10   # values of (a,b,c)
tt = np.linspace(0,5,200)
kk = kf(tt,a1,b1,c1)
plt.plot(tt,kk, label=['(a,b,c)', (a1,b1,c1)])
plt.legend()
plt.xlabel('$t$', fontsize=12)
plt.ylabel('$\kappa(t)$', fontsize=12)
plt.grid()
plt.show()
```



Partial/Directional Derivatives

```
In [77]: x, y, z = smp.symbols('x y z')
```

Partial derivatives $\frac{\partial f}{\partial x}$, $\frac{\partial f}{\partial y}$ and $\frac{\partial^3 f}{\partial x y^2}$ of $f(x, y) = y^2 \sin(x + y)$.

In [78]:
```python
fxy = y**2 * smp.sin(x+y)
display('function, f', fxy)
display('f_x', smp.diff(fxy, x))
display('f_y', smp.diff(fxy, y))
display('f_xyy', smp.diff(fxy, y, y, x))
```

'function, f'

$$y^2 \sin(x+y)$$

'f_x'

$$y^2 \cos(x+y)$$

'f_y'

$$y^2 \cos(x+y) + 2y \sin(x+y)$$

'f_xyy'

$$-y^2 \cos(x+y) - 4y \sin(x+y) + 2\cos(x+y)$$

The Chain Rule

Suppose x,y and z are functions of t and $w = w(x, y, z)$. Find dw/dt.

In [79]:
```python
t = smp.symbols('t')
x, y, z, w = smp.symbols('x y z w', cls = smp.Function)
x = x(t)
y = y(t)
z = z(t)
w = w(x,y,z)
display(w)
display('dw/dt', w.diff(t))
```

$$w(x(t), y(t), z(t))$$

'dw/dt'

$$\frac{d}{dx(t)} w(x(t), y(t), z(t)) \frac{d}{dt} x(t) + \frac{d}{dy(t)} w(x(t), y(t), z(t)) \frac{d}{dt} y(t) + \frac{d}{dz(t)} w(x(t), y(t), z(t)) \frac{d}{dt} z(t)$$

For some particular functions;

In [80]:
```python
w1 = x* smp.sin(y)* smp.exp(-z**2)
display('function w1',w1,'dw1/dt',smp.diff(w1,t))
dw1dt = smp.diff(w1,t).subs([(x, 1/t**2), (y,14*t), (z, 2*t)])
display('for a given x(t), y(t) and z(t),', dw1dt.doit())
```

'function w1'

$$x(t)e^{-z^2(t)} \sin(y(t))$$

'dw1/dt'

$$-2x(t)z(t)e^{-z^2(t)} \sin(y(t)) \frac{d}{dt} z(t) + x(t)e^{-z^2(t)} \cos(y(t)) \frac{d}{dt} y(t) + e^{-z^2(t)} \sin(y(t)) \frac{d}{dt} x(t)$$

'for a given x(t), y(t) and z(t),'

$$-\frac{8e^{-4t^2} \sin(14t)}{t} + \frac{14e^{-4t^2} \cos(14t)}{t^2} - \frac{2e^{-4t^2} \sin(14t)}{t^3}$$

Gradients (∇f)

Now we are dealing with particular coordinate systems.

In [81]:
```
C = CoordSys3D('')
display(C, C.y, C.k)
```

$$\mathrm{CoordSys3D}\left(,\left(\left[\begin{matrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{matrix}\right], \hat{\mathbf{0}}\right)\right)$$

$\mathbf{y}$

$\hat{\mathbf{k}}$

In [82]:
```
f1 = C.x*smp.sin(C.y)
display('function', f1)

gradf1 = gradient(f1)  # gradient
gradf1m = gradf1.to_matrix(C)  # matrix form
display('gradient', gradf1, gradf1m)

display('for y=pi',gradf1.subs(C.y,smp.pi), gradf1m.subs(C.y,smp.pi))
```

```
'function'
```

$\mathbf{x} \sin\left(\mathbf{y}\right)$

```
'gradient'
```

$(\sin\left(\mathbf{y}\right))\hat{\mathbf{i}} + (\mathbf{x} \cos\left(\mathbf{y}\right))\hat{\mathbf{j}}$

$$\left[\begin{matrix}\sin\left(\mathbf{y}\right)\\\mathbf{x}\cos\left(\mathbf{y}\right)\\0\end{matrix}\right]$$

```
'for y=pi'
```

$(-\mathbf{x})\hat{\mathbf{j}}$

$$\left[\begin{matrix}0\\-\mathbf{x}\\0\end{matrix}\right]$$

Directional Derivatives

$$D_u f = \nabla f \cdot u$$

In [83]:
```
uvec = 6*C.i +3*C.j -5*C.k  # writing a vector
u = uvec.normalize()  # making unit vector
display('u', u)
Duf1 = gradient(f1).dot(u) # directional derivative
display('directional derivative', Duf1)
```

'u'

$$(\frac{3\sqrt{70}}{35})\hat{\mathbf{i}} + (\frac{3\sqrt{70}}{70})\hat{\mathbf{j}} + (-\frac{\sqrt{70}}{14})\hat{\mathbf{k}}$$

'directional derivative'

$$\frac{3\sqrt{70}\mathbf{x}\cos(\mathbf{y})}{70} + \frac{3\sqrt{70}\sin(\mathbf{y})}{35}$$

Maxima and Minima of a 2D function

Extreme values of $f(x, y)$ can occur at;

1. Boundary points of the domain of $f(x, y)$.
2. Critical points ($f_x = f_y = 0$)

At a point(a,b);

1. Local maxima: $f_{xx} < 0$ and $f_{xx}f_{yy} - f_{xy}^2 > 0$.
2. Local minima: $f_{xx} > 0$ and $f_{xx}f_{yy} - f_{xy}^2 > 0$.
3. Saddle point: $f_{xx}f_{yy} - f_{xy}^2 < 0$.
4. Inconclusive: $f_{xx}f_{yy} - f_{xy}^2 = 0$.

In [84]:
```
x, y = smp.symbols('x y', real=True)
f = x**2 -y**3 + x*y**2
display('function', f)
```

'function'

$$x^2 + xy^2 - y^3$$

In [85]:
```
# solving df/dx = df/dy = 0
smp.solve([f.diff(x),f.diff(y)])
```

Out[85]: [{x: -9/2, y: -3}, {x: 0, y: 0}]

In [86]:
```
fxx = f.diff(x,x)
fyy = f.diff(y,y)
fxy = f.diff(x,y)
```

In [87]:
```
x1, y1 = -9/2, -3   # input the point
fxx1 = fxx.subs([(x,x1),(y,y1)]).evalf()
D1 = (fxx*fyy-fxy**2).subs([(x,x1),(y,y1)]).evalf()
print('Given point is', (x1,y1))
display('fxx', fxx1)
display('fxx*fyy - fxy**2', D1)
```

```
Given point is (-4.5, -3)
```

```
'fxx'
```

2.0

```
'fxx*fyy - fxy**2'
```

-18.0

In [88]:
```
if fxx1 < 0 and D1 > 0:
    print('local maxima')
elif fxx1 > 0 and D1 > 0:
    print('local minima')
elif D1 < 0:
    print('saddle point')
else:
    print('nothing can be said')
```

```
saddle point
```

Lagrange Multipliers

Minimize $f(x, y, z)$ subject to the constraint $g(x, y, z) = 0$. It requires to solve 2 equations $\nabla f = \lambda \nabla g$ and $g(x, y, z) = 0$.

Example: A space probe has the shape of an ellipsoid $4x^2 + y^2 + 4z^2 = 16$ and after sitting in the sun for an hour, the temperature on its surface is given by $T(x, y, z) = 8x^2 + 4yz - 16z + 600$. Find the hottest point on the surface.

Solution: Here, the function is $f = T = 8x^2 + 4yz - 16z + 600$ and the constraint is $g = 4x^2 + y^2 + 4z^2 - 16 = 0$.

In [89]:
```
C = CoordSys3D('')

lam = smp.symbols('\lambda')
f = 8*C.x**2 +4*C.y*C.z -16*C.z +600
g = 4*C.x**2 +C.y**2 +4*C.z**2 -16

eq1 = gradient(f) - lam*gradient(g)
eq1m = eq1.to_matrix(C)
eq2 = g
display('f',f,'g',g, 'equation 1',eq1,eq1m, 'equation 2',eq2)
```

'f'

$$8\mathbf{x}^2 + 4\mathbf{y}\mathbf{z} - 16\mathbf{z} + 600$$

'g'

$$4\mathbf{x}^2 + \mathbf{y}^2 + 4\mathbf{z}^2 - 16$$

'equation 1'

$$(-8\mathbf{x}\lambda + 16\mathbf{x})\hat{\mathbf{i}} + (-2\mathbf{y}\lambda + 4\mathbf{z})\hat{\mathbf{j}} + (4\mathbf{y} - 8\mathbf{z}\lambda - 16)\hat{\mathbf{k}}$$

$$\begin{bmatrix} -8\mathbf{x}\lambda + 16\mathbf{x} \\ -2\mathbf{y}\lambda + 4\mathbf{z} \\ 4\mathbf{y} - 8\mathbf{z}\lambda - 16 \end{bmatrix}$$

'equation 2'

$$4\mathbf{x}^2 + \mathbf{y}^2 + 4\mathbf{z}^2 - 16$$

In [90]:
```
sols = smp.solve([eq1m,eq2]) # use the matrix to solve
for sol in sols:
    print('\n (x,y,z,lambda) =', sol)
    print('value of local maxima =',f.subs(sol).evalf())
print('\ncompare the values of local maxima and find the highest one,'
      ,'i.e. the highest temperature here')
```

```

 (x,y,z,lambda) = {.x: -4/3, .y: -4/3, .z: -4/3, \lambda: 2}
value of local maxima = 642.666666666667

 (x,y,z,lambda) = {.x: 0, .y: -2, .z: -sqrt(3), \lambda: sqrt(3)}
value of local maxima = 641.569219381653

 (x,y,z,lambda) = {.x: 0, .y: -2, .z: sqrt(3), \lambda: -sqrt(3)}
value of local maxima = 558.430780618347

 (x,y,z,lambda) = {.x: 0, .y: 4, .z: 0, \lambda: 0}
value of local maxima = 600.000000000000

 (x,y,z,lambda) = {.x: 4/3, .y: -4/3, .z: -4/3, \lambda: 2}
value of local maxima = 642.666666666667

compare the values of local maxima and find the highest one, i.e. the highest tempe
rature here
```

Multiple Integrals

In rare cases it can be solved symbolically.

$$\int_0^1 \int_0^{1-x^2} \int_3^{4-x^2-y^2} x^2 e^x \, dz dy dx$$

In [91]:
```
x, y, z = smp.symbols('x y z')
f1 = x**2*smp.exp(x)
smp.integrate(f1, (z,3, 4-x**2-y**2), (y,0,1-x**2), (x,0,1))
```

Out[91]: $-\frac{40252}{3} + 4936e$

We need to do this numerically for most of the cases.

Example:

$$\int_0^1 \int_0^{1-x^2} \int_3^{4-x^2-y^2} xe^{-y} \cos(z) dz dy dx$$

```
# no result
x, y, z = smp.symbols('x, y, z')
f = x*smp.exp(-y)*smp.cos(z)
smp.integrate(f, (z, 3, 4-x**2-y**2), (y, 0, 1-x**2), (x, 0, 1))
```

`tplquad` : function to perform triple integrals in the module `scipy.integrate` .

In [92]:
```
from scipy.integrate import tplquad
f = lambda z,y,x: x*np.exp(-y)*np.cos(z)
x1, x2 = 0, 1
y1, y2 = 0, lambda x: 1 -x**2
z1, z2 = 3, lambda x, y: 4 -x**2 -y**2
tplquad(f, x1, x2, y1, y2, z1, z2)[0]
```

Out[92]: -0.09109526451447894

Integrals and Vector Fields

Line Integrals (Scalar)

Given curve, $\vec{r}(t) = \langle g(t), h(t), k(t) \rangle$. The line integral of $f(x, y, z)$ along the curve is,

$$\int_C f(x, y, z) ds = \int_a^b f(g(t), h(t), k(t)) |d\vec{r}/dt| dt$$

In [93]:
```
t = smp.symbols('t', real=True)
x,y,z,f = smp.symbols('x y z f', cls=smp.Function, real=True)
x = x(t)
y = y(t)
z = z(t)
f = f(x,y,z)
r = smp.Matrix([x,y,z])

integrand = f*r.diff(t).norm()
smp.Integral(integrand, (t, a, b))
```

Out[93]: $\int_a^b \sqrt{\left|\frac{d}{dt} x(t)\right|^2 + \left|\frac{d}{dt} y(t)\right|^2 + \left|\frac{d}{dt} z(t)\right|^2} f(x(t), y(t), z(t)) \, dt$

Suppose,

1. $\vec{r}(t) = \langle \cos(t), \sin(t), t \rangle$; (Helix)
2. $f(x, y, z) = 2xy + \sqrt{z}$

We are going from $t = 0$ to $t = 2\pi$.

In [94]:
```
integrand1 = integrand.subs([(f,2*x*y + smp.sqrt(z)),
                             (x, smp.cos(t)),
                             (y, smp.sin(t)),
                             (z, t)]).doit().simplify()
display(smp.Integral(integrand1, (t, 0, 2*smp.pi)))
smp.integrate(integrand1, (t, 0, 2*smp.pi)).simplify()
```

$$\int_0^{2\pi} \sqrt{2}\left(\sqrt{t} + \sin(2t)\right)\, dt$$

Out[94]: $\frac{8\pi^{\frac{3}{2}}}{3}$

Most of the cases can't be solved symbolically.

Example: Given,

1. $\vec{r}(t) = \left\langle 3\cos(t), 2\sin(t), e^{t/4} \right\rangle$
2. $f(x, y, z) = 2xy + \sqrt{z}$

We are going from $t = 0$ to $t = 2\pi$.

In [95]:
```
integrand2 = integrand.subs([(f, 2*x*y+smp.sqrt(z)),
                             (x, 3*smp.cos(t)),
                             (y, 2*smp.sin(t)),
                             (z, smp.exp(t/4))]).doit().simplify()
display(smp.Integral(integrand2, (t, 0, 2*smp.pi)))
#smp.integrate(integrand2, (t, 0, 2*smp.pi)).simplify()
# no answer
```

$$\int_0^{2\pi} \frac{\left(e^{\frac{t}{8}} + 6\sin(2t)\right)\sqrt{e^{\frac{t}{2}} + 80\sin^2(t) + 64}}{4}\, dt$$

Integration using `quad` function of `scipy.integrate` :

In [96]:
```
from scipy.integrate import quad
integrand2f = smp.lambdify([t], integrand2)
quad(integrand2f, 0, 2*np.pi)[0]
```

Out[96]: `24.294733741870633`

Line Integrals (Vector)

Given, $\vec{r}(t) = \langle g(t), h(t), k(t) \rangle$. The line integral of $\vec{F}(x, y, z)$ along the curve is;

$$\int_C \vec{F}(x, y, z) \cdot d\vec{r} = \int_a^b \vec{F}(g(t), h(t), k(t)) \cdot \frac{d\vec{r}}{dt} dt$$

In [97]:
```python
t = smp.symbols('t', real=True)
x,y,z,F1,F2,F3 = smp.symbols('x y z F_1 F_2 F_3',cls=smp.Function,real=True)
x, y, z = x(t), y(t), z(t)
F1, F2, F3 = F1(x,y,z), F2(x,y,z), F3(x,y,z)
r = smp.Matrix([x, y, z])
F = smp.Matrix([F1, F2, F3])

integrand = F.dot(r.diff(t))
display(smp.Integral(integrand, (t,a,b)).simplify())
```

$$\int_a^b \left(F_1(x(t), y(t), z(t))\frac{d}{dt}x(t) + F_2(x(t), y(t), z(t))\frac{d}{dt}y(t) + F_3(x(t), y(t), z(t))\frac{d}{dt}z(t)\right) dt$$

Example: Find line integral of $\vec{F} = \left\langle \sqrt{z}, -2x, \sqrt{y} \right\rangle$ along the curve $\vec{r}(t) = \left\langle t, t^2, t^4 \right\rangle$ from $t = 0$ to $t = 1$.

In [98]:
```python
integrand1 = integrand.subs([(F1,smp.sqrt(z)),(F2,-2*x),(F3,smp.sqrt(y)),
                             (x,t),(y,t**2),(z,t**4)]).doit().simplify()
display(smp.Integral(integrand1, (t,0,1)))
smp.integrate(integrand1, (t,0,1))
```

$$\int_0^1 t^2 \cdot (4t\,|t| - 3)\; dt$$

Out[98]: $-\frac{1}{5}$

Many of the integrals can't be solved symbolically and we must do that numerically.

Example: Find line integral of $\vec{F} = \left\langle \sqrt{|z|}, -2x, \sqrt{|y|} \right\rangle$ along the curve $\vec{r}(t) = \left\langle 3\cos^2(t), t^2, 2\sin(t) \right\rangle$ from $t = 0$ to $t = 2\pi$.

In [99]:
```python
integrand2 = integrand.subs([(F1,smp.sqrt(smp.Abs(z))),
                             (F2,-2*x),
                             (F3,smp.sqrt(smp.Abs(y))),
        (x,3*smp.cos(t)**2),(y,t**2),(z,2*smp.sin(t))]).doit().simplify()
display(smp.Integral(integrand2, (t,0,2*smp.pi)))
# can't be solved symbolically
```

$$\int_0^{2\pi} 2\left(-6t\cos(t) - 3\sqrt{2}\sin(t)\sqrt{|\sin(t)|} + |t|\right)\cos(t)\, dt$$

In [100]:
```python
from scipy.integrate import quad
integrand2f = smp.lambdify([t], integrand2)
quad(integrand2f, 0, 2*np.pi)[0]
```

Out[100]: -118.4352528130723

Surface Integrals (Scalar)

Area of a surface is given by;

$$A = \iint_S \left|\frac{d\vec{r}}{du} \times \frac{d\vec{r}}{dv}\right| dudv; \quad where\ \vec{r} = \vec{r}(u, v)$$

$\vec{r}$ denotes the surface and it's a function of 2 variables.

The surface integral of a scalar function $G(\vec{r})$ is given by;

$$\iint_S G(\vec{r}(u,v)) \left| \frac{d\vec{r}}{du} \times \frac{d\vec{r}}{dv} \right| dudv$$

In [101]:
```
#  r = r(u=rho, v=theta)
rho, th = smp.symbols('\\rho \\theta', pos=True, real=True)
x, y, z, G = smp.symbols('x y z G', cls=smp.Function, real=True)
x, y, z = x(rho,th), y(rho,th), z(rho,th)
G = G(x,y,z)
r = smp.Matrix([x,y,z])

integrand = G* r.diff(rho).cross(r.diff(th)).norm()
integrand
```

Out[101]:

$$\sqrt{\left|\frac{\partial}{\partial \rho}x(\rho,\theta)\frac{\partial}{\partial \theta}y(\rho,\theta) - \frac{\partial}{\partial \theta}x(\rho,\theta)\frac{\partial}{\partial \rho}y(\rho,\theta)\right|^2 + \left|\frac{\partial}{\partial \rho}x(\rho,\theta)\frac{\partial}{\partial \theta}z(\rho,\theta) - \frac{\partial}{\partial \theta}x(\rho,\theta)\frac{\partial}{\partial \rho}z(\rho,\theta)\right|^2 +$$

Example: 2D parabola is given by $\vec{r}(x,y) = \left\langle x, y, x^2 + y^2 \right\rangle$ and thus $\vec{r}(\rho,\theta) = \left\langle \rho\cos\theta, \rho\sin\theta, \rho^2 \right\rangle$. The surface density is given by $G(x,y,z) = x^2 + y^2$. Find surface integral for $0 < \rho < 1$ and $0 < \theta < 2\pi$.

In [102]:
```
integrand1 = integrand.subs([(G,x**2+y**2),
    (x,rho*smp.cos(th)),(y,rho*smp.sin(th)),(z,rho**2)]).doit().simplify()
display(integrand1)
smp.integrate(integrand1, (rho,0,1), (th,0,2*smp.pi)).simplify()
```

$$\rho^2\sqrt{4\rho^2+1}\,|\rho|$$

Out[102]:

$$\frac{\pi\left(1+25\sqrt{5}\right)}{60}$$

Complicated integrals can be solved numerically ***(try it)****.*

Surface Integtals(Vectors)

The surface integral of a vector function $\vec{G}(\vec{r})$ is given by;

$$\iint_S \vec{G}(\vec{r}(u,v)) \cdot \left(\frac{d\vec{r}}{du} \times \frac{d\vec{r}}{dv} \right) dudv$$

This is also known as flux of the vector field $\vec{G}$ through the surface $\vec{r}$.

In [103]:

```python
# u = rho and v = theta
rho, th = smp.symbols('\\rho \\theta', pos=True, real=True)
x,y,z,G1,G2,G3 = smp.symbols('x y z G_1 G_2 G_3',cls=smp.Function,real=True)
x, y, z = x(rho,th), y(rho,th), z(rho,th)
G1, G2, G3 = G1(x,y,z), G2(x,y,z), G3(x,y,z)
r = smp.Matrix([x,y,z])
G = smp.Matrix([G1,G2,G3])

integrand = G.dot(r.diff(rho).cross(r.diff(th)))
integrand
```

Out[103]:

$$\Big(\frac{\partial}{\partial\rho}x(\rho,\theta)\frac{\partial}{\partial\theta}y(\rho,\theta) - \frac{\partial}{\partial\theta}x(\rho,\theta)\frac{\partial}{\partial\rho}y(\rho,\theta)\Big)\, G_3\left(x(\rho,\theta), y(\rho,\theta), z(\rho,\theta)\right) + \Big(-\frac{\partial}{\partial\rho}x(\rho,\theta)\frac{\partial}{\partial\theta}z(\rho,$$

$$\left(x(\rho,\theta), y(\rho,\theta), z(\rho,\theta)\right) + \Big(\frac{\partial}{\partial\rho}y(\rho,\theta)\frac{\partial}{\partial\theta}z(\rho,\theta) - \frac{\partial}{\partial\theta}y(\rho,\theta)\frac{\partial}{\partial\rho}z(\rho,\theta)\Big)\, G_1\left(x(\rho,\theta), y(\rho,\theta), z(\rho,\theta\right.$$

Example: 2D parabola is given by $\vec{r}(x,y) = \left\langle x, y, x^2+y^2 \right\rangle$ and thus $\vec{r}(\rho,\theta) = \left\langle \rho\cos\theta, \rho\sin\theta, \rho^2 \right\rangle$. The vector field is given by $\vec{G}(x,y,z) = \left\langle y^2, z, 0 \right\rangle$. Find the flux of $\vec{G}$ for $0 < \rho < 1$ and $0 < \theta < \pi$ (through half of the surface).

In [104]:

```python
integrand1 = integrand.subs([(G1,y**2),(G2,z),(G3,0),
    (x,rho*smp.cos(th)),(y,rho*smp.sin(th)),(z,rho**2)]).doit().simplify()
display(integrand1)
smp.integrate(integrand1, (rho,0,1), (th,0,smp.pi))
```

$$-2\rho^4\left(\frac{\sin(2\theta)}{2}+1\right)\sin(\theta)$$

Out[104]: $-\frac{4}{5}$

*Complicated integrals can be solved numerically **(try it)**.*

Explicit sympy Functionality

In [105]:

```python
from sympy.vector import ParametricRegion
```

Find the mass of a cylinder with radius a and height h centered at origin with density $\rho_V(x,y) = x^2+y^2$.

In [106]:

```python
C = CoordSys3D('')
a,h,r,th,z = smp.symbols(r'a h r \theta z',pos=True)
cylinder = ParametricRegion((r*smp.cos(th),r*smp.sin(th),z),
                            (r,0,a),(th,0,2*smp.pi),(z,0,h))
vector_integrate(C.x**2+C.y**2,cylinder)
```

Out[106]: $\frac{\pi a^4 h}{2}$

This cannot be done numerically. Using `vector_integrate` by defining field `CoordSys3D` and region `ParametricRegion` is the only way. See more in the link, https://docs.sympy.org/latest/modules/vector/vector_integration.html (https://docs.sympy.org/latest/modules/vector/vector_integration.html)

In [ ]:

```

```