Problem 2

Language: Python3

A manipulator has the following Jacobian matrix.

```
J = [ -l1*s1, -l2*s12, -l2*s12*l1*c1, l2*c12*l2*c12 ]
```

Find the joint torques such that the manipulator applies a static force of 10 newton-meter in the x direction at the end-effector.

In [2]:
```
import sympy as sp
```

In [4]:
```
# define the symbols we'll need
fx, fy, t1, t2, l1, l2 = sp.symbols('fx fy t1 t2 l1 l2')

# create jacobian matrix
p2_jacobian = sp.Matrix([
    [(-l1 * sp.sin(t1)) - (l2 * sp.sin(t1+t2)), -l2 * sp.sin(t1+t2)],
    [(l1* sp.cos(t1))+(l2 * sp.cos(t1+t2)), (l2*sp.cos(t1+t2))]
])
print('jacobian matrix:')
display(p2_jacobian)

# create force matrix which
# has (10 newton meters) in x component
p2_force = sp.Matrix([10,0])

# create a torque matric that
# we'll solve for
p2_torque = sp.Matrix([t1,t2])

# solve for the joint torques
print('joint torques:')
p2_solved_torques = sp.simplify(
    sp.solve(p2_jacobian.T * p2_force, p2_torque)
)
display(p2_solved_torques)
```

```
jacobian matrix:
```

$$\begin{bmatrix} -l_1 \sin(t_1) - l_2 \sin(t_1 + t_2) & -l_2 \sin(t_1 + t_2) \\ l_1 \cos(t_1) + l_2 \cos(t_1 + t_2) & l_2 \cos(t_1 + t_2) \end{bmatrix}$$

```
joint torques:
```

```
[(0, 0), (0, pi), (-pi, pi), (pi, 0)]
```