

```

sage: accel_x = M*ax == Ft*cos(theta) - Ff sage: var("M ax Ft theta Ff N R alpha r") (M, ax, Ft, theta, Ff, N, R,
alpha, r) sage: accel_x = M*ax == Ft*cos(theta) - Ff sage: var("M ax ay Ft theta Ff N R alpha r") (M, ax, ay, Ft,
theta, Ff, N, R, alpha, r) sage: accel_x = M*ax = Ft*cos(theta) - Ff sage: does_not_fly_away = M*ay
= 0 sage: accel_y = 0 == Ft*sin(theta)+N-M*9.81 sage: var("M ax ay Ft theta Ff N R alpha r g") (M, ax, ay,
Ft, theta, Ff, N, R, alpha, r, g) sage: accel_x = M*ax = Ft*cos(theta) - Ff sage: does_not_fly_away =
M*ay = 0 sage: accel_y = 0 == Ft*sin(theta)+N-M*g sage: var("M ax ay Ft theta Ff N R alpha r g l") (M, ax, ay,
Ft, theta, Ff, N, R, alpha, r, g, l) sage: accel_x = M*ax = Ft*cos(theta) - Ff sage: does_not_fly_away =
M*ay = 0 sage: accel_y = 0 = Ft*sin(theta)+N-M*g sage: torquenet = I*alpha = r*Ft*cos(theta)-R*Ff
sage: rotationaltranslational = R*alpha==ax sage: solve([accel_x, accel_y, doesnotflyaway, torquenet, rotationaltranslational],
ax, ay, al .....: pha, Ff, N)

```