

STEP1

[here](#)

STEP2

使用sympy简化J及计算J的导数[here](#)

演算结果:

```
script git:(master) X python3 polynomial_function_test.py
1.0*(1.0*T**4 + 4.0*T**2*v_x_0**2 + 4.0*T**2*v_y_0**2 + 4.0*T**2*v_z_0**2 + 12.0*T*p_x_0*v_x_0 - 12.0*T*p_x_f*v_x_0 + 12.0*T*p_y_0*v_y_0 - 12.0*T*p_y_f*v_y_0 + 12.0*T*p_z_0*v_z_0 - 12.0*T*p_z_f*v_z_0 + 12.0*p_x_0**2 - 24.0*p_x_0*p_x_f + 12.0*p_x_f**2 + 12.0*p_y_0**2 - 24.0*p_y_0*p_y_f + 12.0*p_y_f**2 + 12.0*p_z_0**2 - 24.0*p_z_0*p_z_f + 12.0*p_z_f**2)/T**3
-----
1.0*(1.0*T**4 - 4.0*T**2*v_x_0**2 - 4.0*T**2*v_y_0**2 - 4.0*T**2*v_z_0**2 - 24.0*T*p_x_0*v_x_0 + 24.0*T*p_x_f*v_x_0 - 24.0*T*p_y_0*v_y_0 + 24.0*T*p_y_f*v_y_0 - 24.0*T*p_z_0*v_z_0 + 24.0*T*p_z_f*v_z_0 - 36.0*p_x_0**2 + 72.0*p_x_0*p_x_f - 36.0*p_x_f**2 - 36.0*p_y_0**2 + 72.0*p_y_0*p_y_f - 36.0*p_y_f**2 - 36.0*p_z_0**2 + 72.0*p_z_0*p_z_f - 36.0*p_z_f**2)/T**4
script git:(master) X
```

J导数求根及J值计算[here](#)

效果如下

