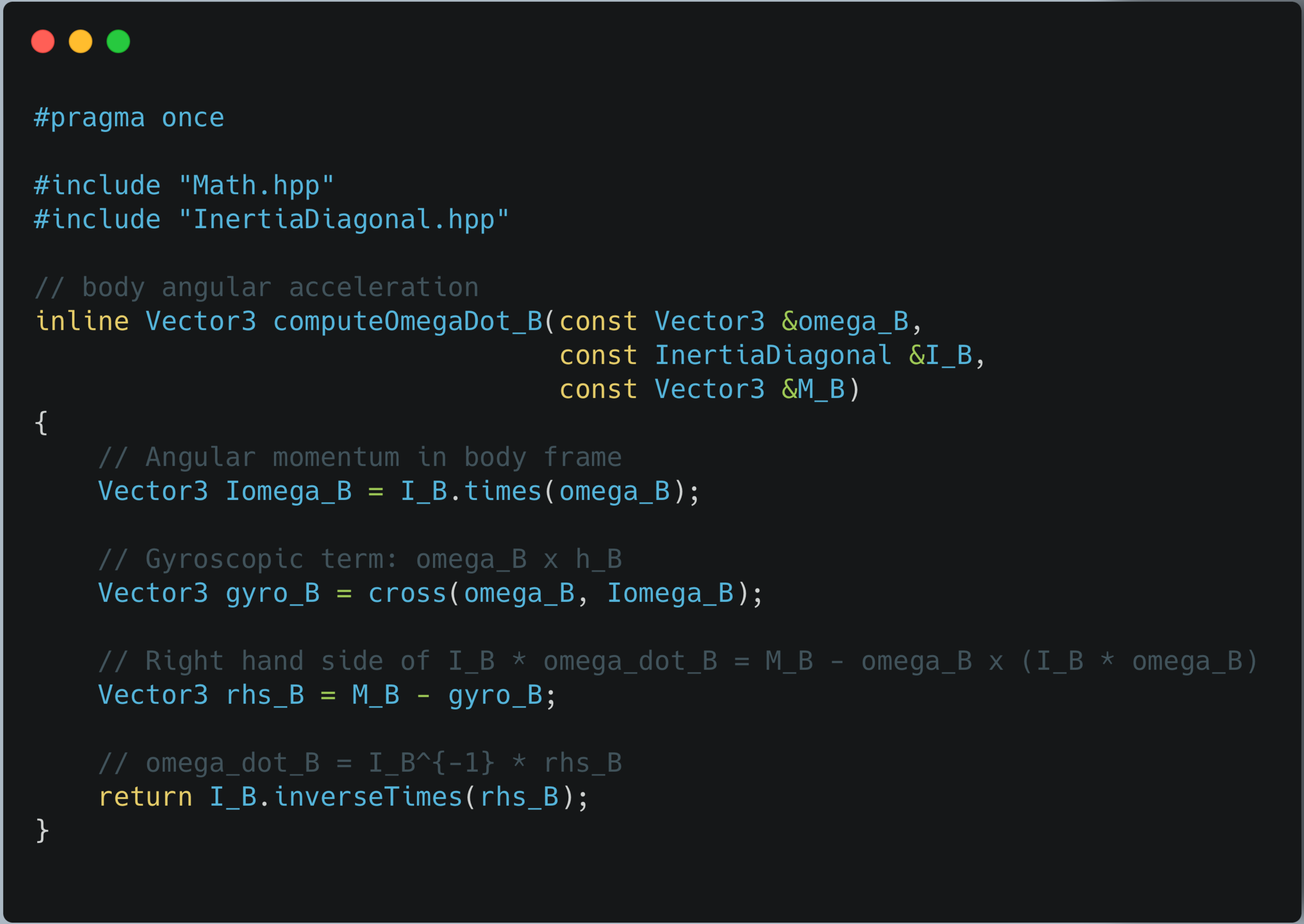

```
#pragma once

#include "Math.hpp"
#include "InertiaDiagonal.hpp"

// body angular acceleration
inline Vector3 computeOmegaDot_B(const Vector3 &omega_B,
                                 const InertiaDiagonal &I_B,
                                 const Vector3 &M_B)
{
    // Angular momentum in body frame
    Vector3 Iomega_B = I_B.times(omega_B);

    // Gyroscopic term: omega_B x h_B
    Vector3 gyro_B = cross(omega_B, Iomega_B);

    // Right hand side of I_B * omega_dot_B = M_B - omega_B x (I_B * omega_B)
    Vector3 rhs_B = M_B - gyro_B;

    // omega_dot_B = I_B^{-1} * rhs_B
    return I_B.inverseTimes(rhs_B);
}
```