

Моделирование эволюции двойной системы с потерей массы одной из компонент

Санкт-Петербургский государственный университет

Кафедра небесной механики

Специальный практикум по небесной механике / физике планетных систем

Работа №1 по теме:

Численное моделирование двойной звездной системы, в которой одна из компонент теряет массу.

Студент 5 курса: Тимофеев Игорь Вадимович

Функция правых частей задачи N тел

```
void operator()(const matrix_data& R, matrix_data& A, data_type t){
    A = zero_matrix<data_type>(number_bodies,dim);
    subrange(A,0,number_bodies,0,dim/2) = subrange(R,0,number_bodies,dim/2,dim);
    matrix_data norm_r(number_bodies,number_bodies);

    norm(R,norm_r);
    for(uint j=0;j<number_bodies;++j){
        for(uint k=0;k<j;++k){
            subrange(A,j,j+1,dim/2,dim) += m[k] *
            (subrange(R,k,k+1,0,dim/2) - subrange(R,j,j+1,0,dim/2)) /
            (norm_r(k,j) * norm_r(k,j) * norm_r(k,j));
        }
        for(uint k=j+1; k < number_bodies; ++k){
            subrange(A,j,j+1,dim/2,dim) += m[k] *
            (subrange(R,k,k+1,0,dim/2) - subrange(R,j,j+1,0,dim/2)) /
            (norm_r(k,j) * norm_r(k,j) * norm_r(k,j));
        }
    }
    subrange(A,0,number_bodies,dim/2,dim) *= G;
}
```

Элементы орбиты в задаче N тел

```
void get_orbital_elements(const vector_data &R, uint i, uint j,
    vector_data& orbital_elements,const matrix_data &norm_r){
    data_type kappa_quad,r,v;
    data_type a,e,inclination;

    kappa_quad=G*(m[i]+m[j]);
    r=norm_r(i,j);
    v=norm_2(subrange(R,dim/2,dim));

    data_type h = v*v/2 - kappa_quad/r;

    vector_data c(3);
    if(dim == 6)
        c = cross_product(subrange(R,0,dim/2),subrange(R,dim/2,dim));
    else{
        c(0) = c(1) = 0;
        c(2) = R(0)*R(3) - R(1)*R(2);
    }

    if(2.*h*inner_prod(c,c)/(kappa_quad*kappa_quad) + 1 < 0){
        assert(2.*h*inner_prod(c,c)/(kappa_quad*kappa_quad)
+ 1 > -1.0e-10 && "Negative sqrt");
        e = 0;
    }else
        e = sqrt(1.+2.*h*inner_prod(c,c)/(kappa_quad*kappa_quad));
    a = -kappa_quad/2/h;

    if(norm_2(c) != 0)
        inclination = acos(c(2)/norm_2(c));
    else
        inclination = 0;

    orbital_elements(0) = a;
    orbital_elements(1) = e;
    orbital_elements(2) = inclination*180/pi;
    orbital_elements(3) = (1-e)*a;
}
```

В задаче использовался интегратор Рунге-Кутты пятого порядка.
Для описания системы использовалась библиотека boost и язык Си++.

```
RungeKutta5_Fehlberg<data_type, matrix_data> rk5;  
number_steps = integrate(rk5, std::ref(Integrator),  
initial, 0.0, T/number_steps, number_steps, std::ref(Integrator));
```

Результаты. Полуось растёт, а орбита становится более вытянутой.

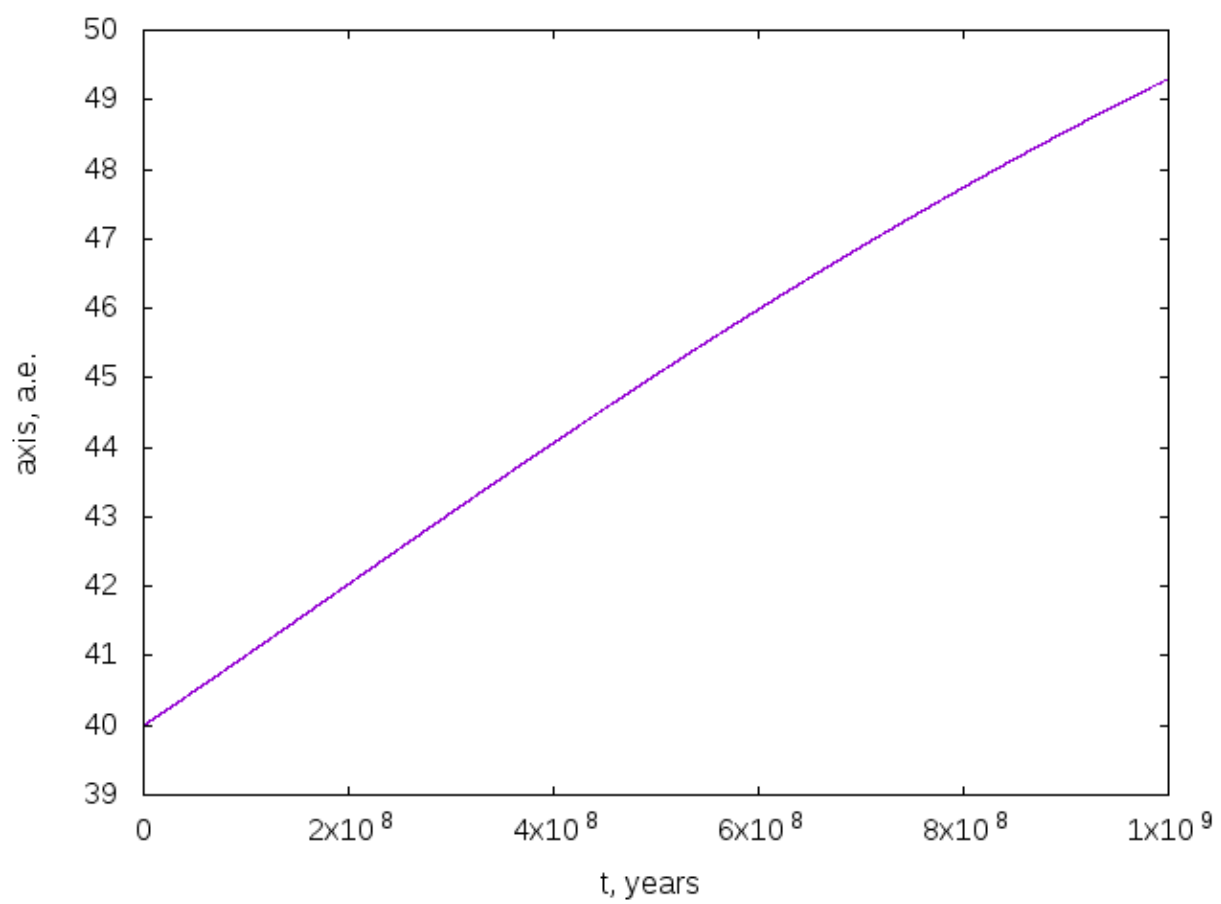


Рис. 1: Зависимость большой полуоси от времени

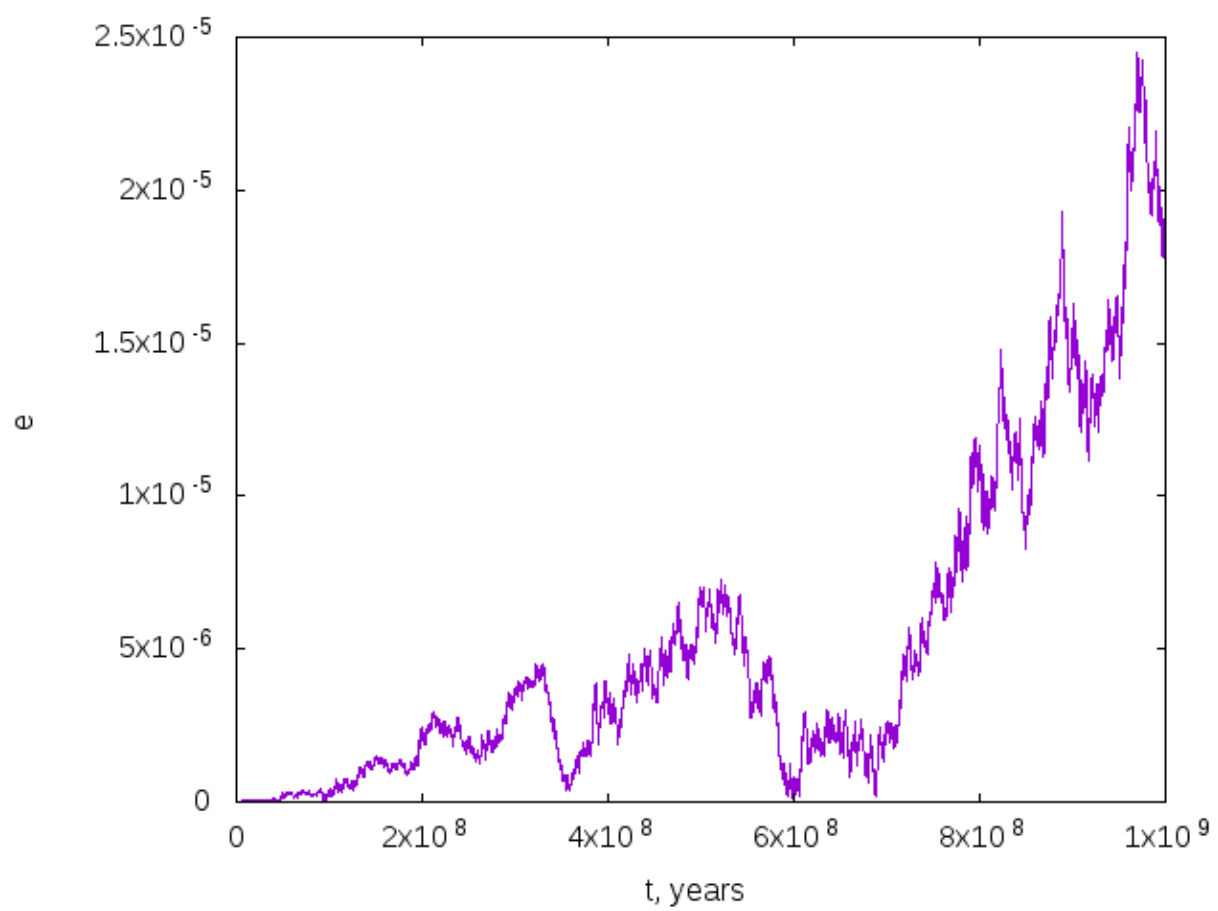


Рис. 2: Зависимость эксцентриситета от времени