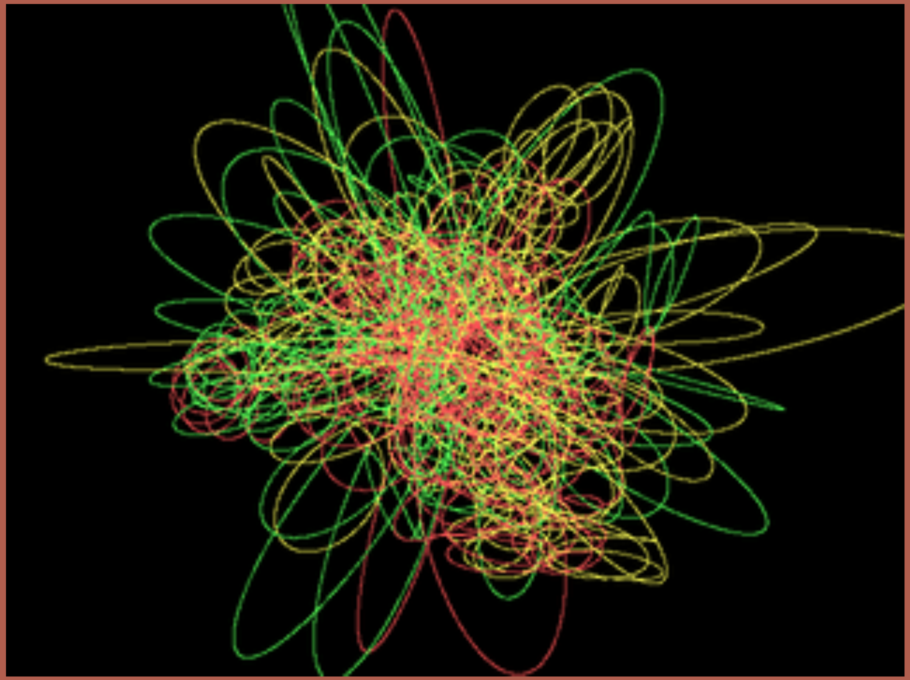
Улучшение точности численного интегрирования гравитационной системы N тел с помощью сдвоенных чисел двойной точности



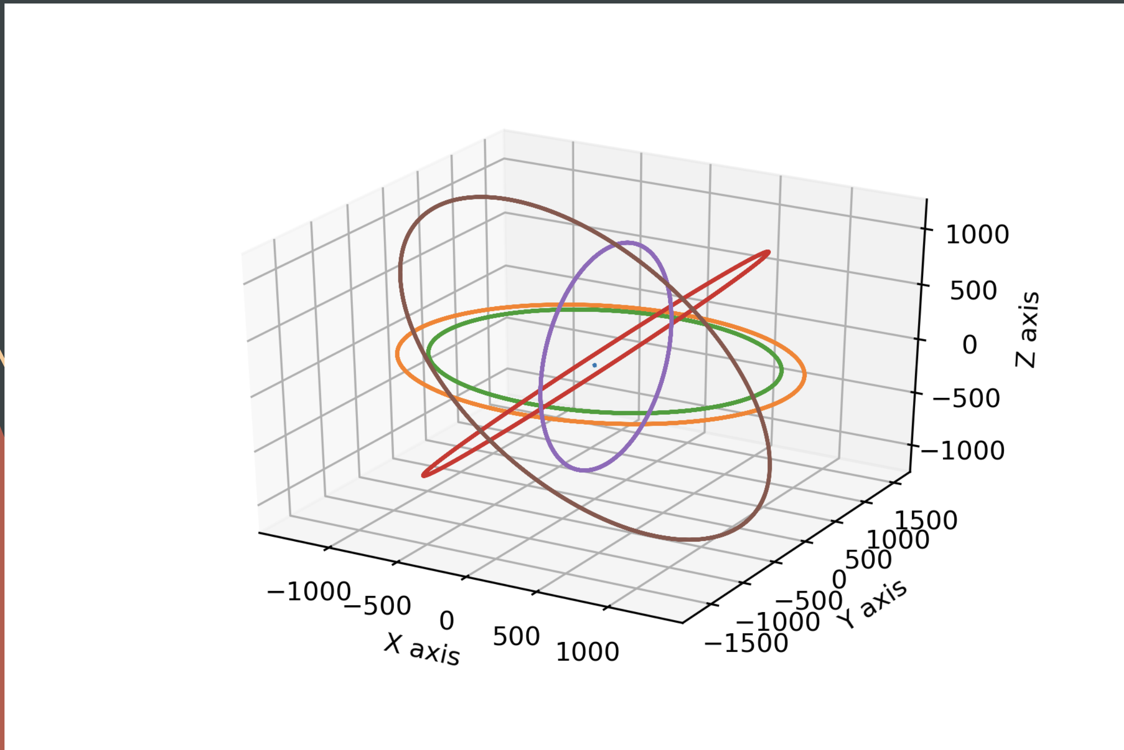
Субботин Максим 9382

Кодуков Александр 9382

Задача N тел

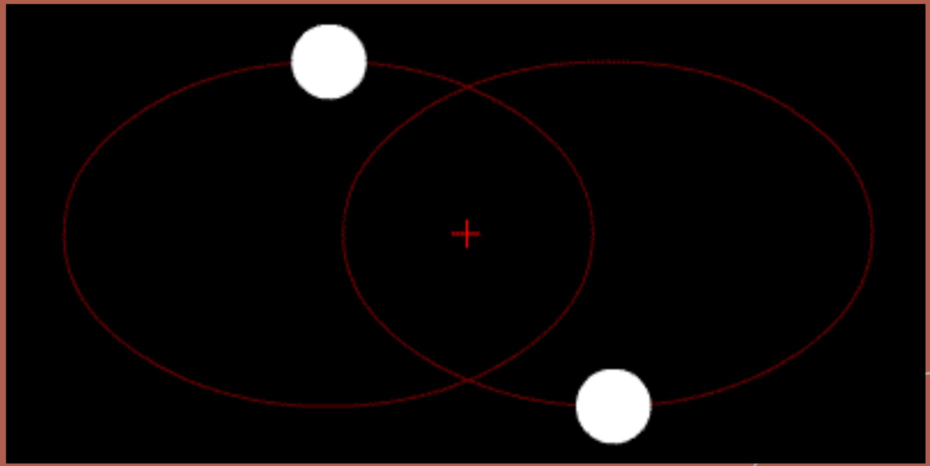
Описать эволюцию системы из N тел (материальных точек) в гравитационном поле.

Тело задается массой, положением и скоростью.



Задача N тел

- 1 тело: описывается 1 законом Ньютона
- 2 тела: существует общее решение для нахождения орбиты
- 3+ тел: невозможно выразить через уравнения от координат и скоростей в общем случае. Задача может быть решена только численными методами



Физика явления

- Сила гравитационного взаимодействия, действующая на тело *n*:

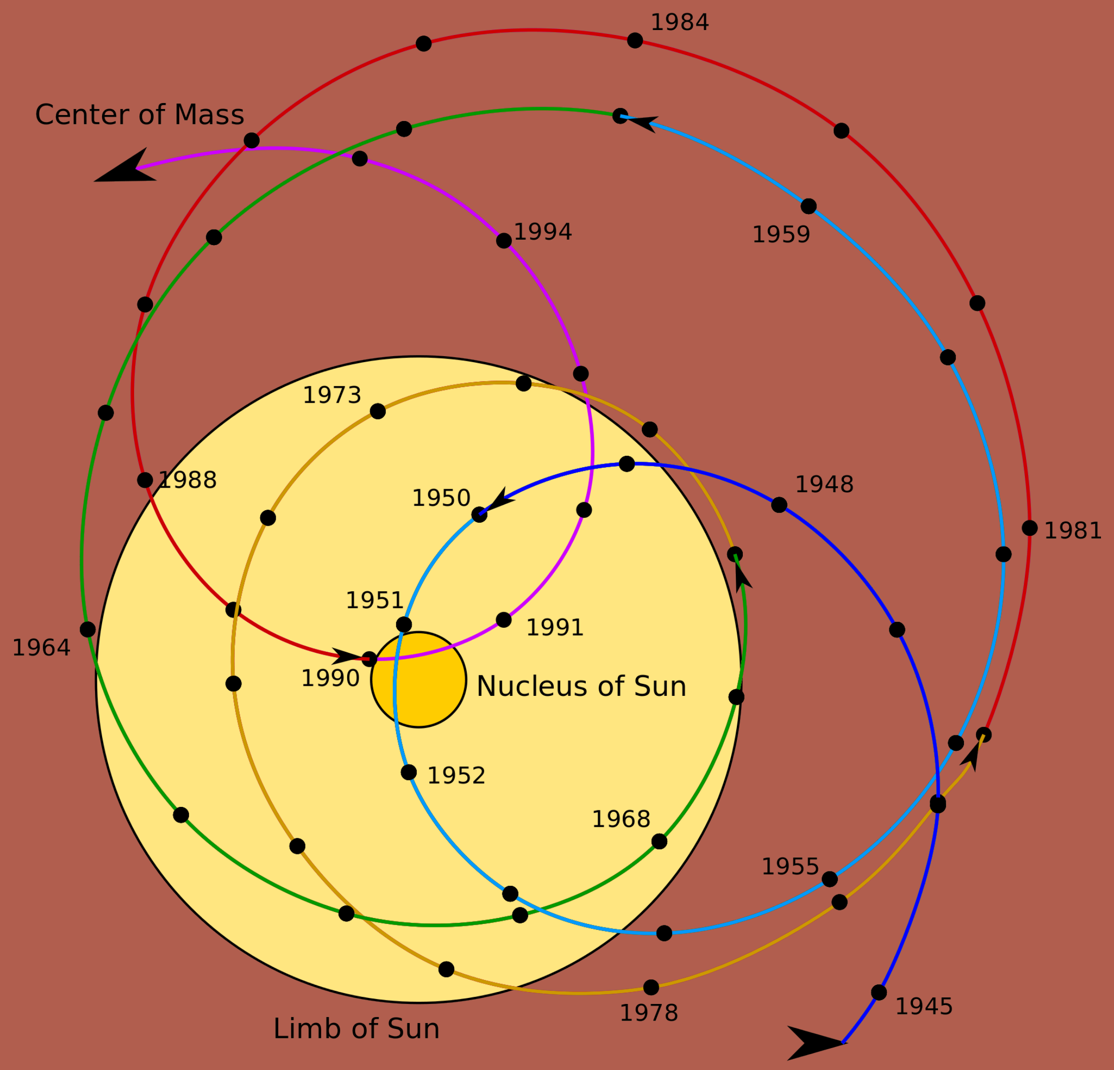
$$\vec{F}_n = -G \sum_{k \neq n} m_n m_k \frac{\vec{r}_n - \vec{r}_k}{|\vec{r}_n - \vec{r}_k|^3}$$

- Ускорение тела *n*:

$$\vec{a}_n = \vec{F}_n / m_n = -G \sum_{k \neq n} m_k \frac{\vec{r}_n - \vec{r}_k}{|\vec{r}_n - \vec{r}_k|^3}$$

Инварианты

- Полная энергия системы
- Момент импульса системы
- Положение барицентра системы



Задача Коши

Начальные данные:

$$\begin{cases} \dfrac{dx}{dy} = f(x, y) \\ y(x_0) = y_0 \end{cases}$$

Необходимо найти:

$$y(t) - ?$$

Переход к численным методам

$$\frac{\partial^2 \vec{r}_n}{\partial t^2} = f_n = -G \sum_{k \neq n} m_k \frac{\vec{r}_n - \vec{r}_k}{|\vec{r}_n - \vec{r}_k|^3}$$

$$\Rightarrow$$

$$\frac{\partial \vec{v}_n}{\partial t} = f_n$$

$$\frac{\partial \vec{r}_n}{\partial t} = \vec{v}_n$$

Семейство методов Рунге-Кутты

$$y_{n+1} = y_n + h\sum_{i=1}^{s} b_i k_i$$

$$\begin{aligned}
k_1 &= f(t_n, y_n),\\
k_2 &= f(t_n + c_2 h, y_n + h(a_{21}k_1)),\\
k_3 &= f(t_n + c_3 h, y_n + h(a_{31}k_1 + a_{32}k_2)),\\
&\vdots\\
k_i &= f\left(t_n + c_i h, y_n + h\sum_{j=1}^{i-1} a_{ij}k_j\right).
\end{aligned}$$

Таблица Бутчера:

$$\begin{array}{c|cccc}
c_1 & a_{11} & a_{12} & \dots & a_{1s}\\
c_2 & a_{21} & a_{22} & \dots & a_{2s}\\
\vdots & \vdots & \vdots & \ddots & \vdots\\
c_s & a_{s1} & a_{s2} & \dots & a_{ss}\\
\hline
 & b_1 & b_2 & \dots & b_s
\end{array}$$

Метод Эйлера

$$y_{n+1} = y_n + hf(t_n, y_n).$$

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$

Равноускоренное движение:

$$\vec{v}(t) = \vec{v}_0 + \vec{a}t$$

Метод Рунге-Кутты 4 порядка

$$y_{n+1} = y_n + \frac{1}{6}h\left(k_1 + 2k_2 + 2k_3 + k_4\right),$$
$$t_{n+1} = t_n + h$$

$$k_1 = f(t_n, y_n),$$
$$k_2 = f\left(t_n + \frac{h}{2}, y_n + h\frac{k_1}{2}\right),$$
$$k_3 = f\left(t_n + \frac{h}{2}, y_n + h\frac{k_2}{2}\right),$$
$$k_4 = f\left(t_n + h, y_n + hk_3\right).$$

$$\begin{array}{c|cccc}
0 & 0 & 0 & 0 & 0 \\
1/2 & 1/2 & 0 & 0 & 0 \\
1/2 & 0 & 1/2 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 \\
\hline
 & 1/6 & 1/3 & 1/3 & 1/6
\end{array}$$

IEEE 754

Стандарт описывает:

- формат чисел с плавающей точкой: мантисса, экспонента (показатель), знак числа;
- представление положительного и отрицательного нуля, положительной и отрицательной бесконечностей, а также *нечисла́* (англ. *Not-a-Number, NaN*);
- методы, используемые для преобразования числа при выполнении математических операций;
- исключительные ситуации: деление на ноль, переполнение, потеря значимости, работа с денормализованными числами и другие;
- операции: арифметические и другие.

Double: мантисса - 53 бита (число двойной точности)

Double-double: мантисса - 106 бит (сдвоенное число двойной точности)

Long double: мантисса - 112 бит (число четверной точности)

Арифметика double-double

Число в double-double:

$x = x_l + x_h$,где $|x_l| \leq \frac{1}{2} ulp(x_h)$

Алгоритм суммирования с учетом ошибки:

Algorithm 4.3 The Fast2Sum algorithm [108].

$s \leftarrow \mathrm{RN}(a+b)$
$z \leftarrow \mathrm{RN}(s-a)$
$t \leftarrow \mathrm{RN}(b-z)$

Сложение чисел в double-double:

Algorithm 14.1 Dekker's algorithm for adding two double-word numbers (x_h, x_ℓ) and (y_h, y_ℓ) [108]. We assume radix 2.

if $|x_h| \geq |y_h|$ **then**
  $(r_h, r_\ell) \leftarrow \mathrm{Fast2Sum}(x_h, y_h)$
  $s \leftarrow \mathrm{RN}(\mathrm{RN}(r_\ell + y_\ell) + x_\ell)$
else
  $(r_h, r_\ell) \leftarrow \mathrm{Fast2Sum}(y_h, x_h)$
  $s \leftarrow \mathrm{RN}(\mathrm{RN}(r_\ell + x_\ell) + y_\ell)$
end if
$(t_h, t_\ell) \leftarrow \mathrm{Fast2Sum}(r_h, s)$
return (t_h, t_ℓ)

Fused Multiply-Add/Substract

$$\circ(ab + c)$$

Инструкция процессора, позволяющая посчитать a * b ± c, с округлением только финального результата.

Пример использования:

Algorithm 4.7 Dekker product.

Require: $s = \lceil p/2 \rceil$

$(x_h, x_\ell) \leftarrow \mathrm{Split}(x, s)$
$(y_h, y_\ell) \leftarrow \mathrm{Split}(y, s)$
$r_1 \leftarrow \mathrm{RN}(x \cdot y)$
$t_1 \leftarrow \mathrm{RN}(-r_1 + \mathrm{RN}(x_h \cdot y_h))$
$t_2 \leftarrow \mathrm{RN}(t_1 + \mathrm{RN}(x_h \cdot y_\ell))$
$t_3 \leftarrow \mathrm{RN}(t_2 + \mathrm{RN}(x_\ell \cdot y_h))$
$r_2 \leftarrow \mathrm{RN}(t_3 + \mathrm{RN}(x_\ell \cdot y_\ell))$

Algorithm 5.1 2MultFMA(x_1, x_2)

$r_1 \leftarrow \mathrm{RN}(x_1 \cdot x_2)$
$r_2 \leftarrow \mathrm{RN}(x_1 \cdot x_2 - r_1)$

Алгоритмы суммирования

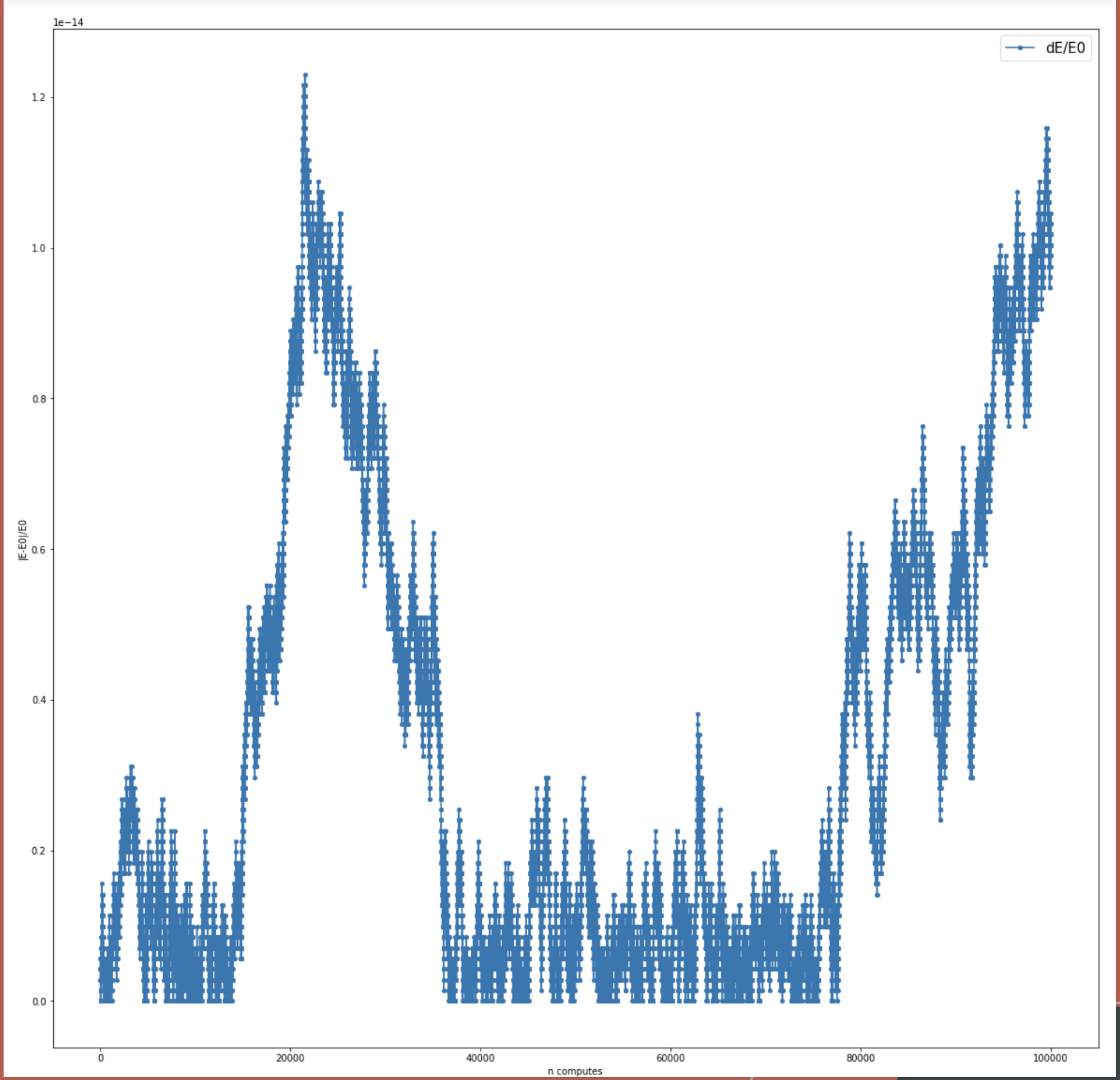
Алгоритм Кэхена:

```
function KahanSum(input)
    var sum = 0.0
    var c = 0.0

    for i = 1 to input.length do
        var y = input[i] - c
        var t = sum + y
        c = (t - sum) - y
        sum = t
    next i

    return sum
```

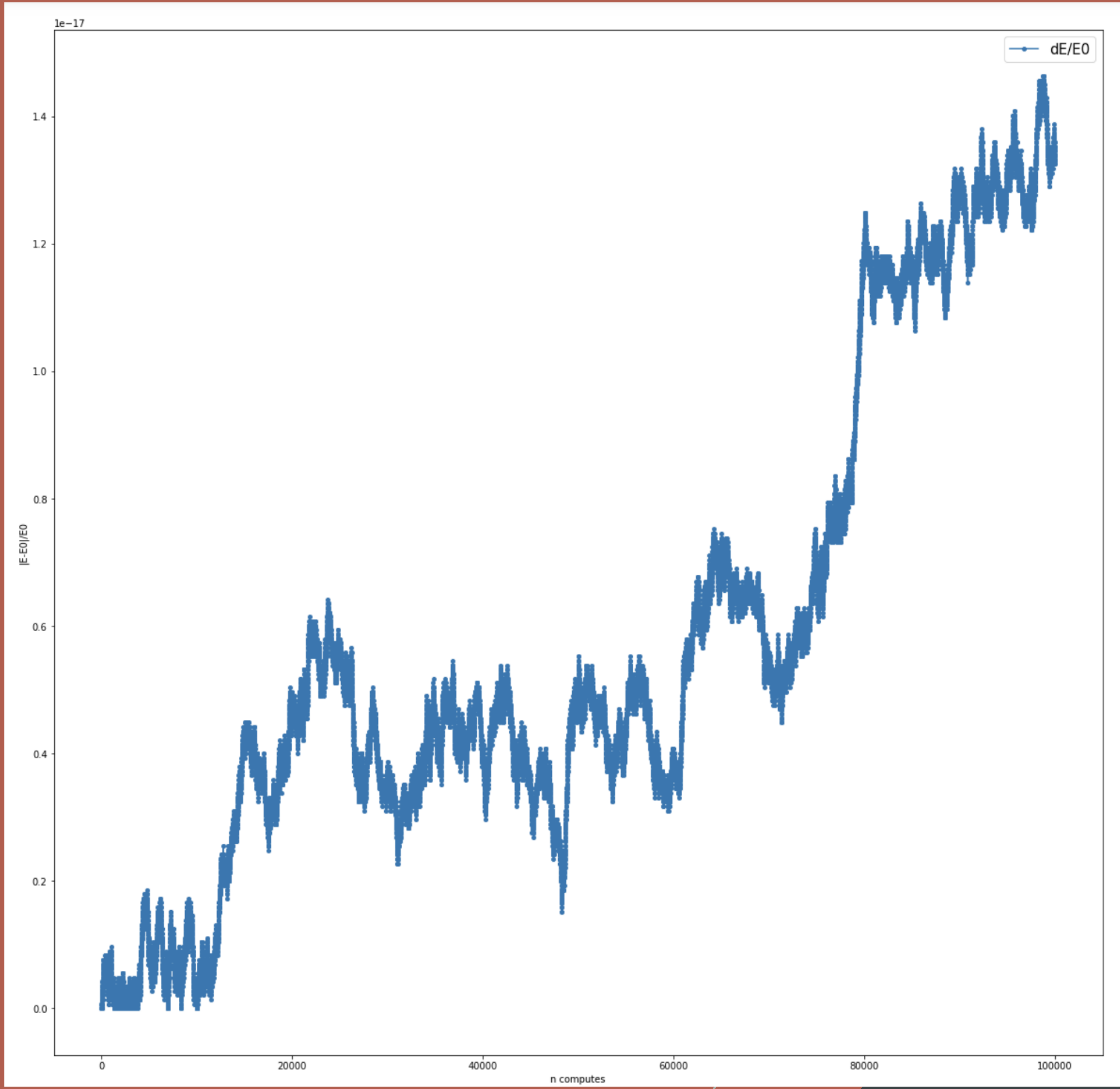
АлгоритмНоймаера:

```
function NeumaierSum(input)
    var sum = 0.0
    var c = 0.0

    for i = 1 to input.length do
        var t = sum + input[i]
        if |sum| >= |input[i]| then
            c += (sum - t) + input[i]
        else
            c += (input[i] - t) + sum
        endif
        sum = t
    next i

    return sum + c
```

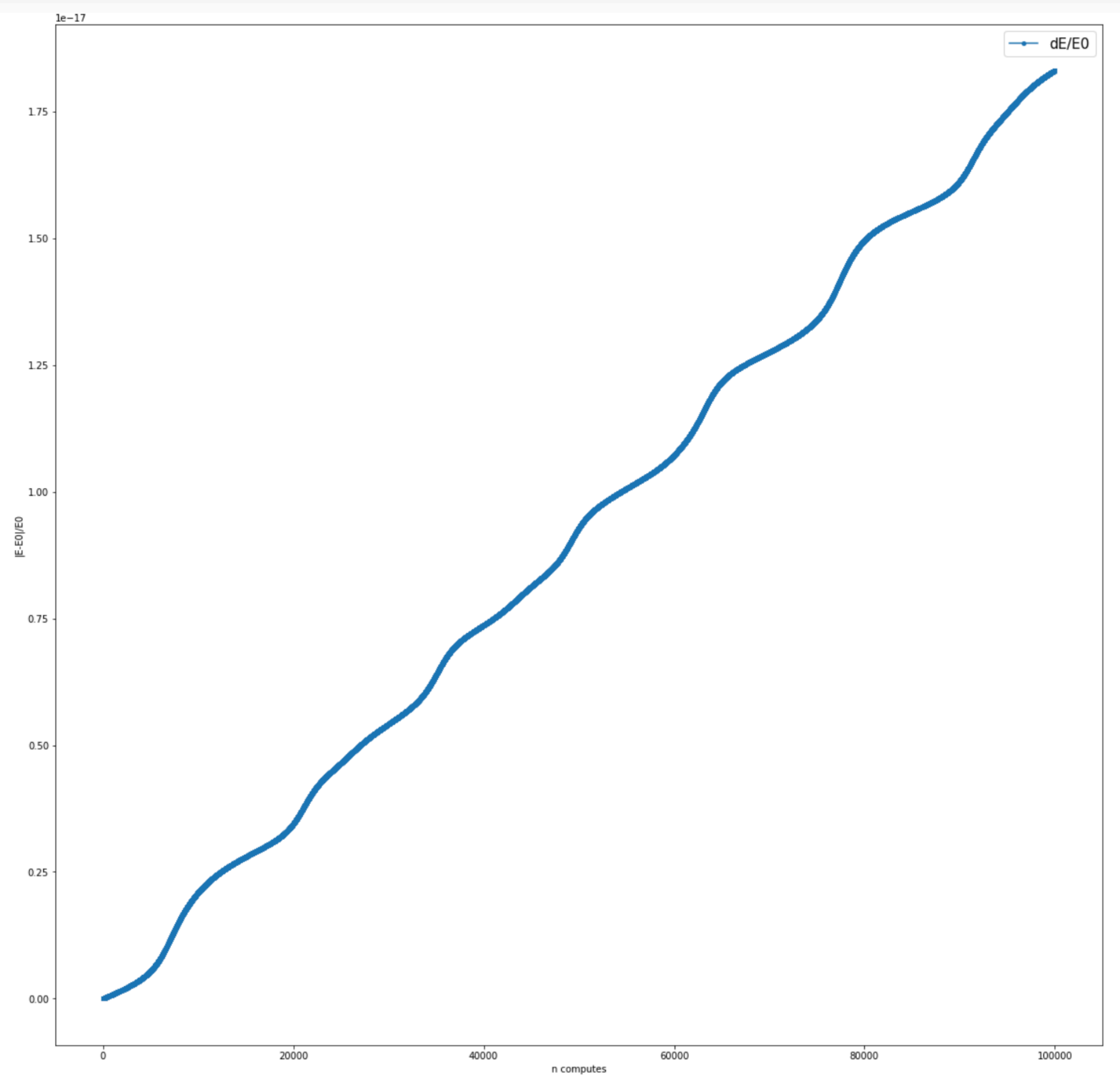
Dopri8 double Energy



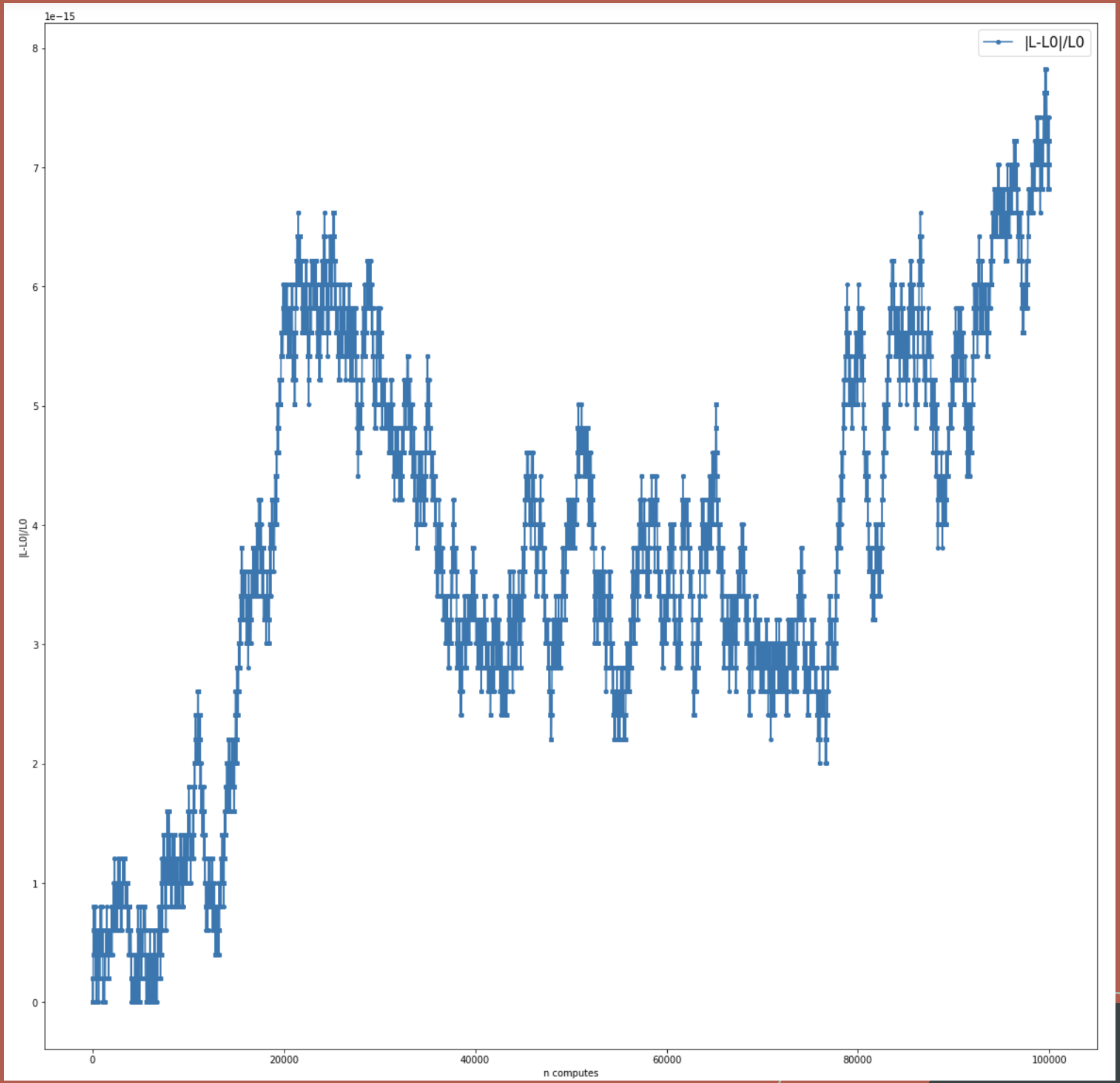
Dopri8 long double Energy



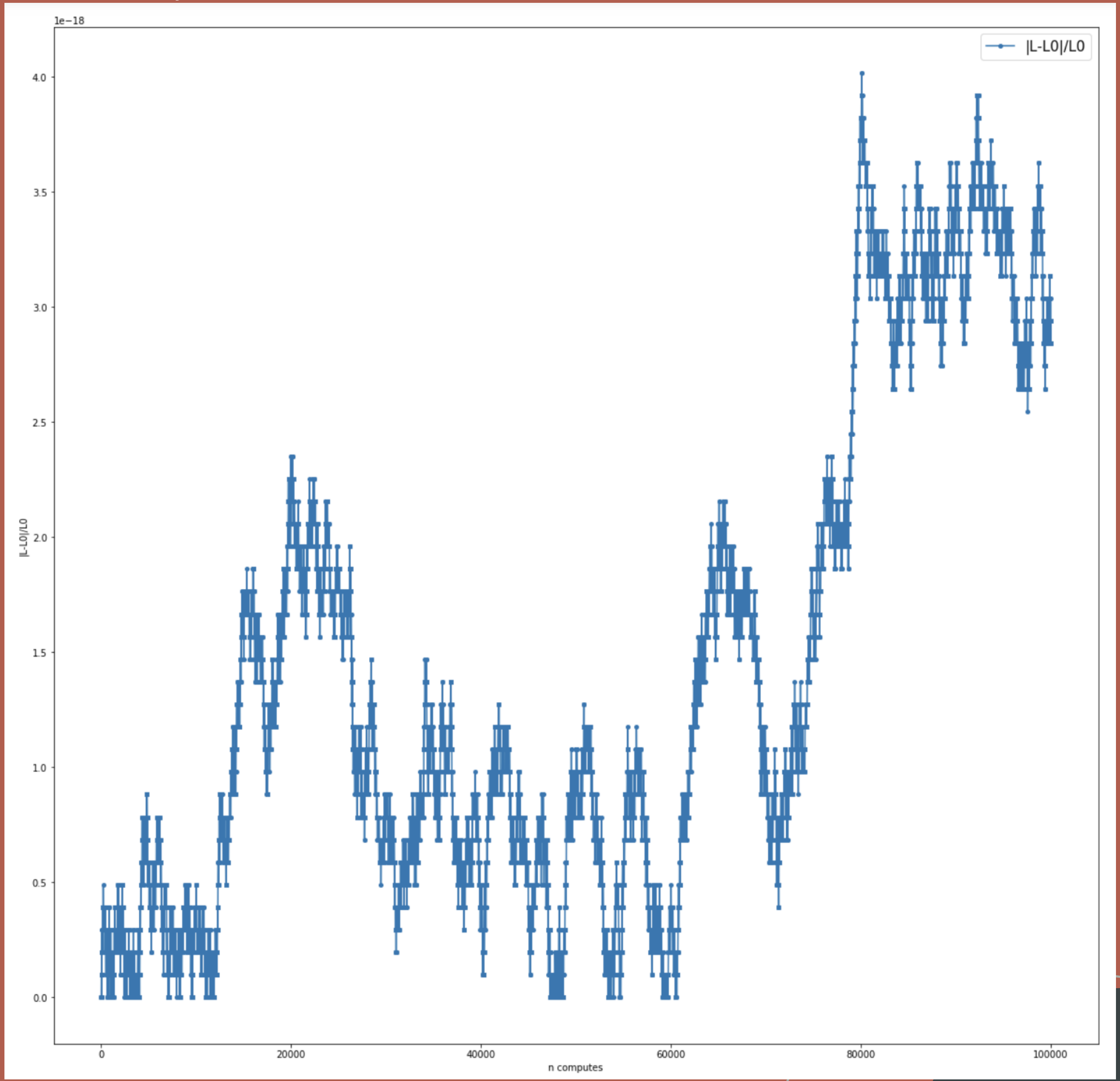
Dopri8 double-double Energy



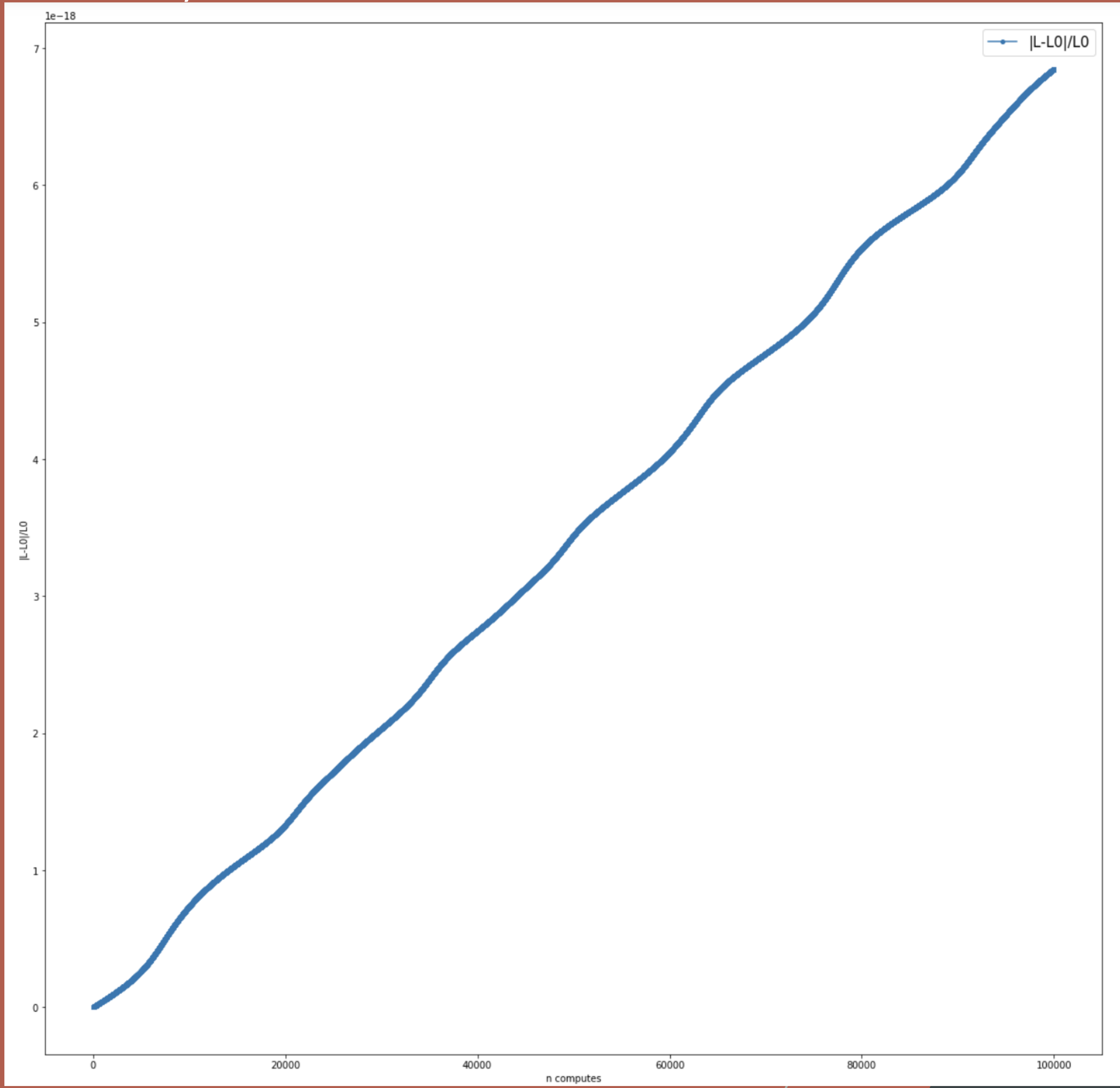
Dopri8 double момент импульса



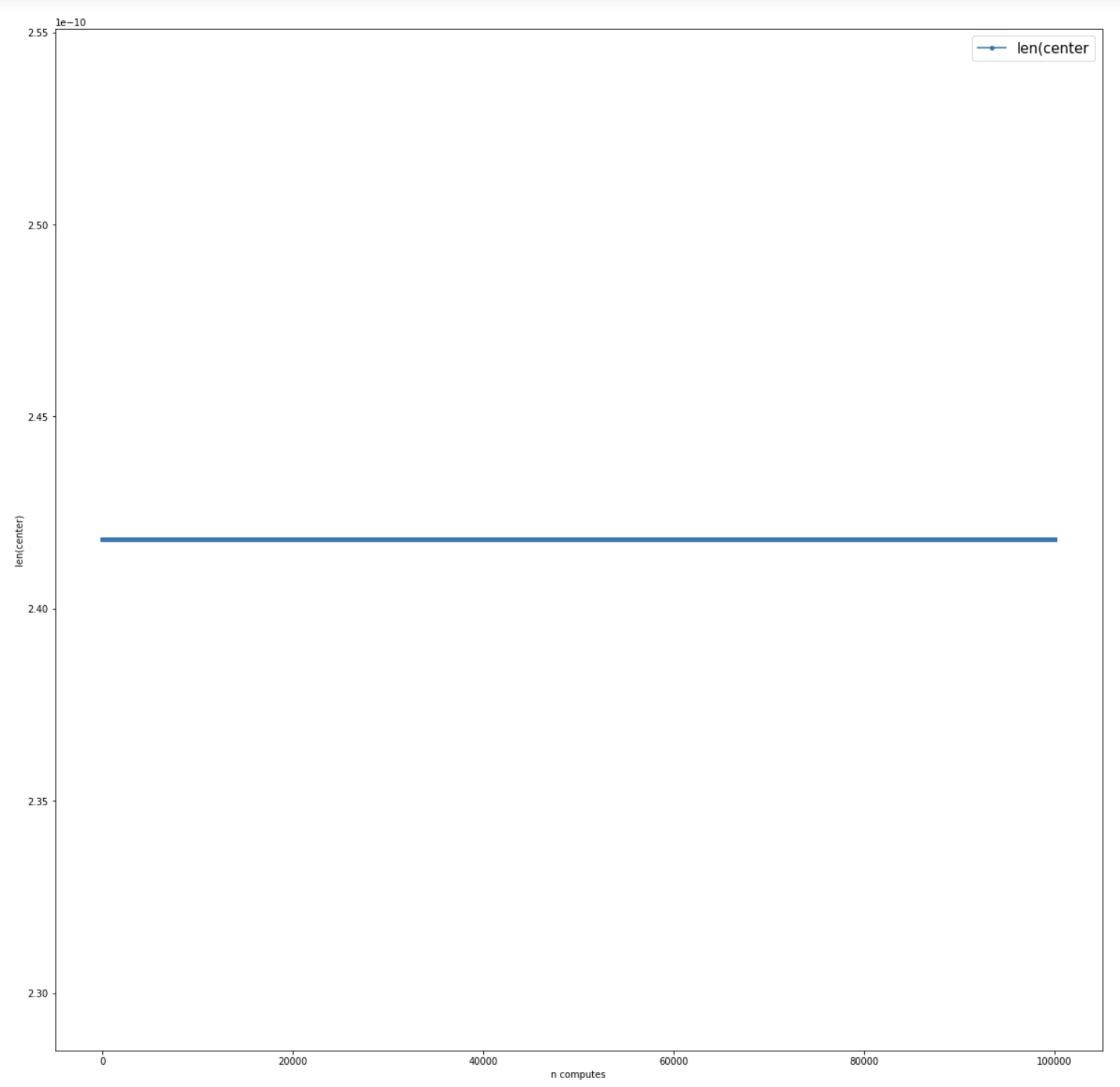
Dopri8 long double момент импульса



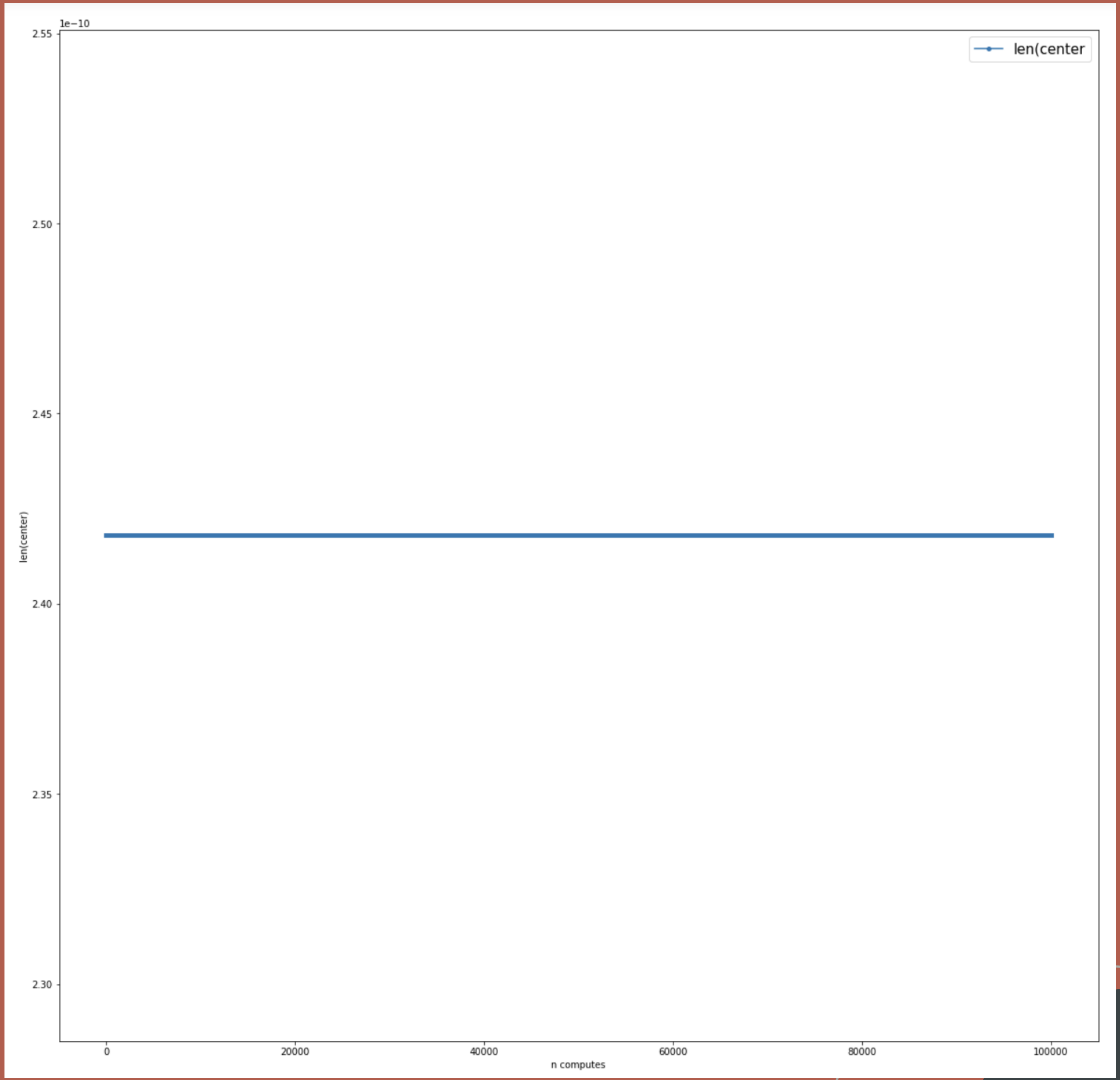
Dopri8 double-double момент импульса



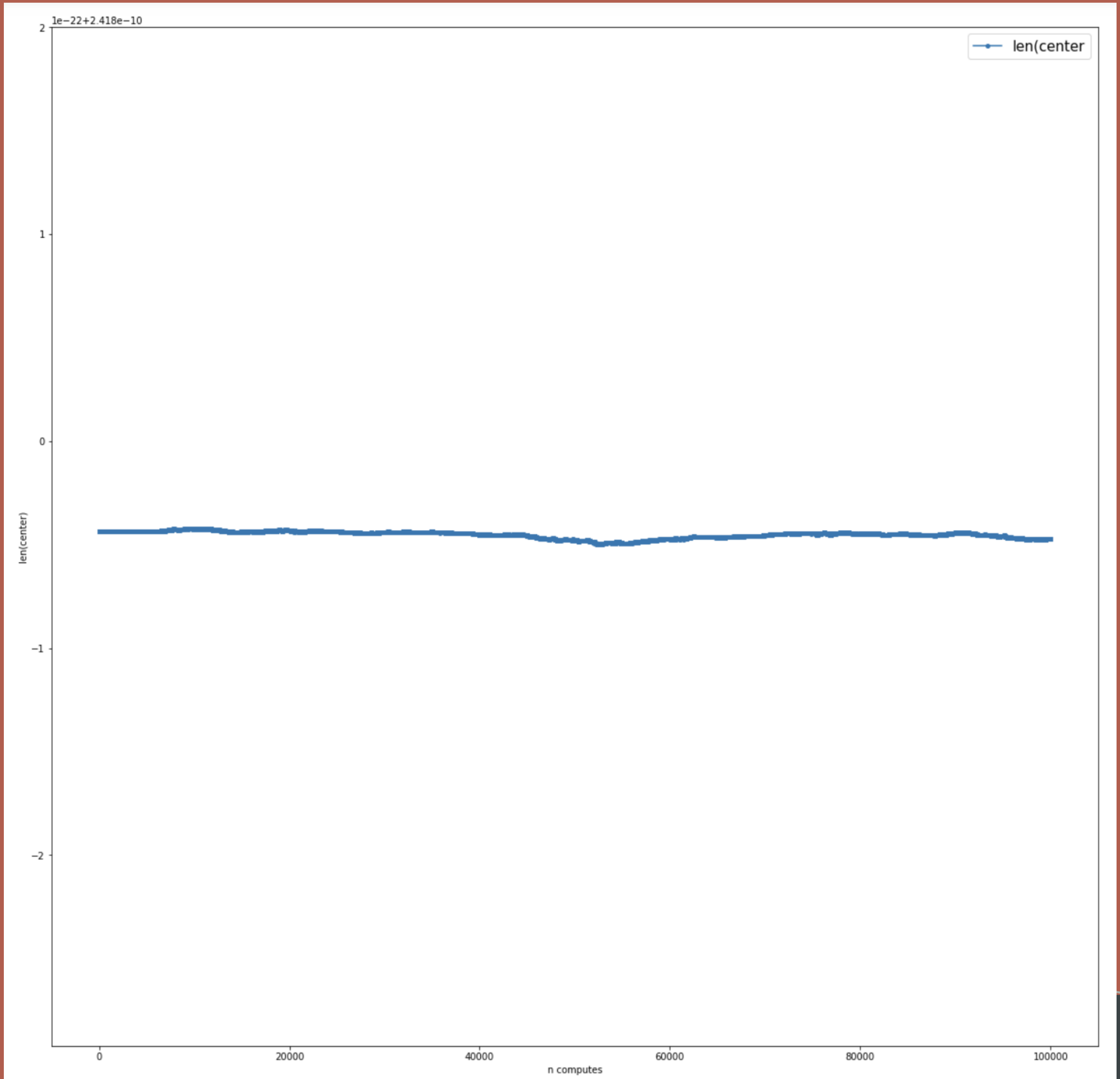
Dopri8 double-double Center



Dopri8 long double Center

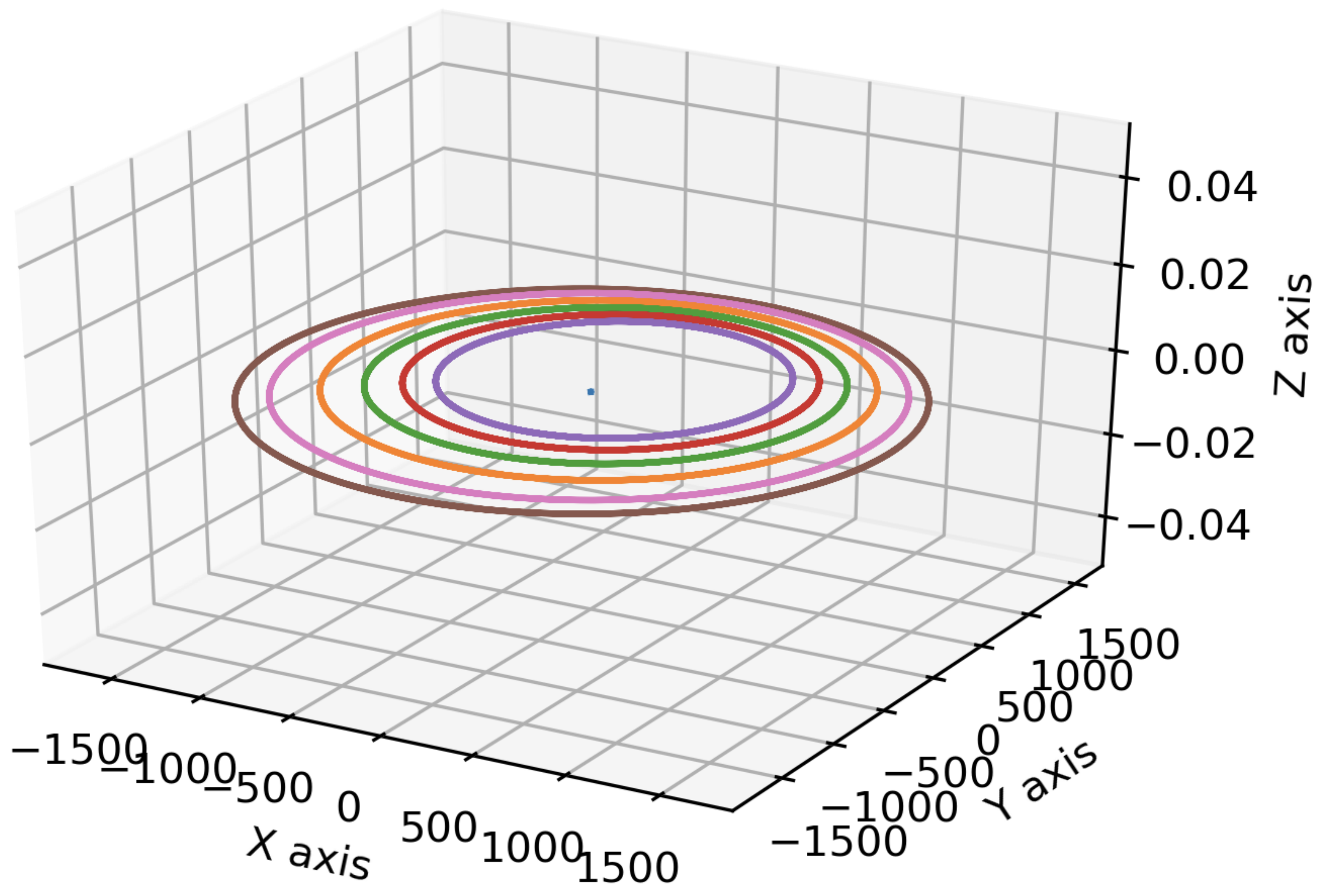


Dopri8 double Center



В микросекундах

	RK4	Dopri8
double	11111189	52421846
Long double	14074273	65553740
Double-double	38293416	136196255



0.04
0.02
0.00
−0.02
−0.04
Z axis
−1500
−1000
−500
0
500
1000
1500
X axis
1500
1000
500
0
−500
−1000
−1500
Y axis