1. Реализовать решение задачи N тел:

1.1. методом Верле на Python,

1.2. методом Верле, распараллелив вычисления с помощью multiprocessing,

1.3. методом Верле, реализованным на Cython,

1.4. методом Верле, реализованным на OpenCL,  
1.5. решение методом odeint (или solve\_ivp).

2. Написать тест, проверяющий, что все решатели из п.1 выдают адекватный результат на примере моделирования движения планет Солнечной системы и расчета методом odeint на всем отрезке. Провести визуализацию движения планет Солнечной системы в двумерном случае.   
На одном изображении привести графики погрешностей для всех методов по сравнению с odeint.

3. Привести графики

- времени работы методов из п.1 (всех, кроме odeint) для N (N=50, 100, 200) сгенерированных случайным образом частиц.

- ускорения по сравнению с последовательной версией метода Верле

для N (N=50, 100, 200) сгенерированных случайным образом частиц.

N частиц генерируется один раз, после этого расчет проводится для всех методов, за временной результат берется среднее по 3 запускам время работы.

1. 实施N体问题的解决方案。

1.1. Python中的Verle方法。

1.2. Verle方法，用多处理方法进行平行计算。

1.3. 用Cython实现的Verle方法。

1.4. 用OpenCL实现的Verle方法。

1.5. 用ODEINT方法（或SOLIS\_IVP）进行求解。

2. 写一个测试，检查第1项中的所有求解器是否对太阳系行星的运动进行建模并以整段的颂词法进行计算的例子给出足够的结果。在二维情况下对太阳系行星的运动进行可视化。

在一张图片上给出了所有方法与ODEINT的对比误差图。

3. 提供地块

- 第1项中的方法（所有方法，除ODEINT外）对N（N=50，100，200）个随机产生的粒子的操作时间。

- 与Verle方法的顺序版本相比，加速了。

为N（N=50，100，200）个随机产生的粒子。

N个粒子生成一次，之后对所有方法进行计算，取3次运行的时间平均值作为时间结果。