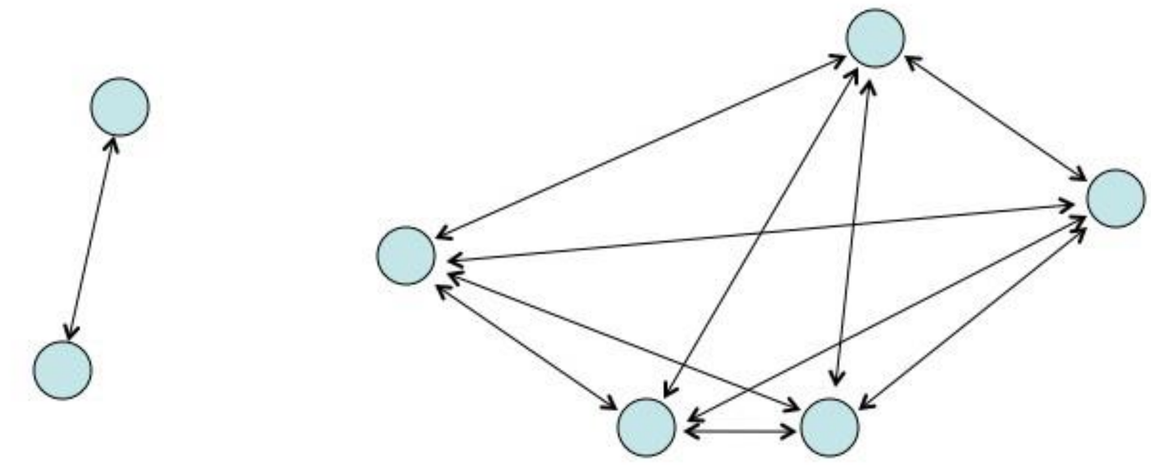
The N-Body Problem

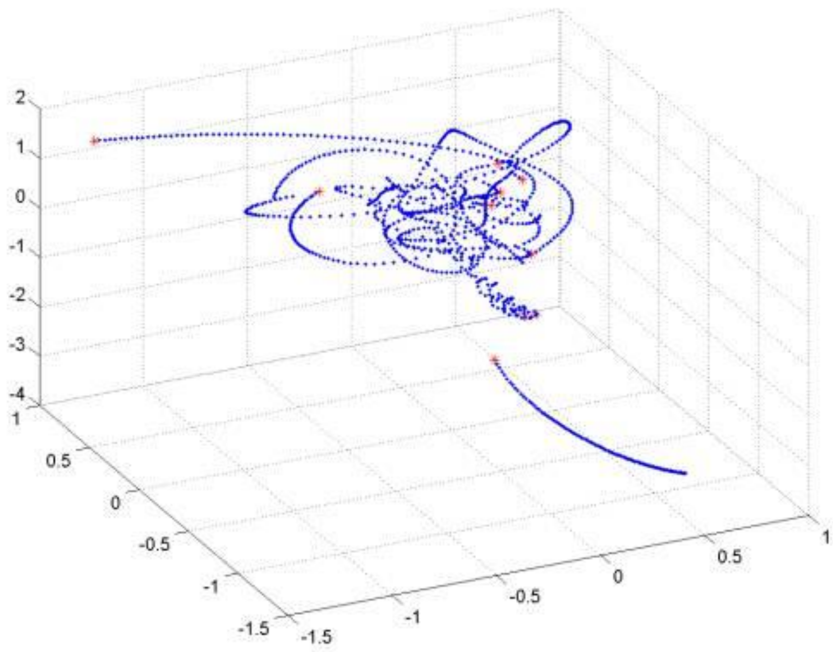


A 3D Numerical Exploration by Nicholas Cemenenkoff

Problem Statement

Given an empty 3D space and a set of masses with distinct positions and well-defined initial velocities,

if accounting only for the gravitational force between the bodies (and not allowing for collisions)…



…what sort of orbital trajectories result? Are they stable? Which factors play a critical role in stability? Is there chaos in the system?

Mathematical Formalism

We must solve Newton's equations of motions for n separate bodies in 3D.

Given a set of positions, the equation below shows how to obtain the 3D acceleration experienced by body i in the presence of j other bodies.

The accelerations are integrated to find velocities, and then the velocities are integrated to find positions.

$$\boldsymbol{a}_{ij} = \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{G m_j \left(\boldsymbol{r}_j - \boldsymbol{r}_i\right)}{\left\|\boldsymbol{r}_j - \boldsymbol{r}_i\right\|^3}$$

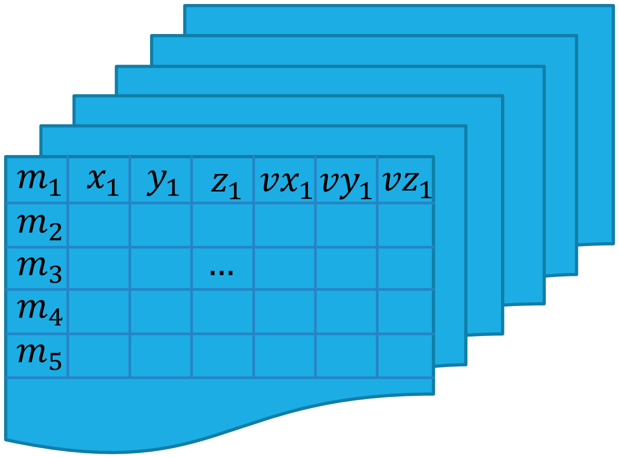
The positions give the potential energy, while the velocities give the kinetic energy.

$$U = -\sum_{1 \leq i \leq j \leq n} \frac{G m_i m_j}{\left\|\boldsymbol{r}_j - \boldsymbol{r}_i\right\|}$$

Numerical Implementation

We employ 8 different stepping methods:

1. Euler
2. Euler-Cromer (EC)
3. 2nd Order Runge-Kutta (RK2)
4. 4th Order Runge-Kutta (RK4)
5. Velocity Verlet (VV)
6. Position Verlet (PV)
7. Velocity Extended Forest-Ruth-Like (VEFRL)
8. Position Extended Forest-Ruth-Like (PEFRL)




```
if method == 'euler':
    for i in tqdm(range(steps)):
        r[i+1] = r[i] + dt*v[i]
        v[i+1] = v[i] + dt*accel(r[i])
```

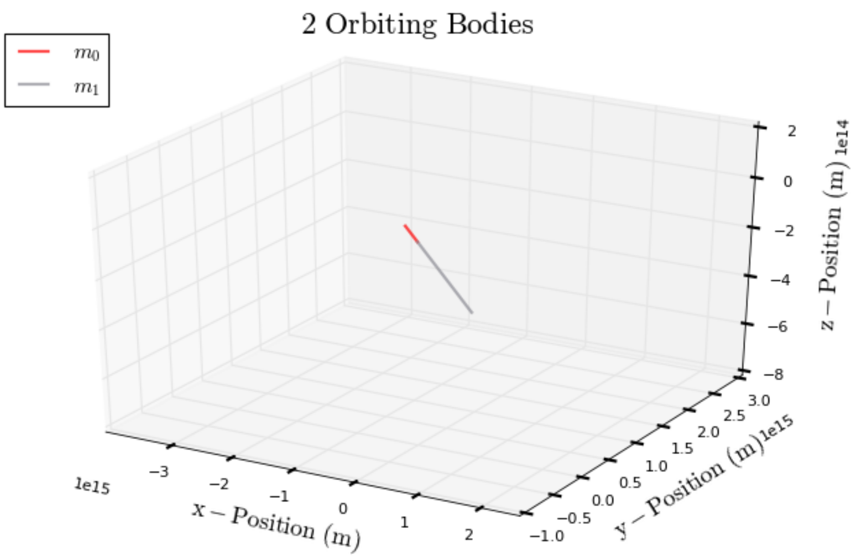
```
if method == 'vv':
    for i in tqdm(range(steps)):
        v_iphalf = v[i] + (dt/2)*accel(r[i])
        r[i+1] = r[i] + dt*v_iphalf
        v[i+1] = v_iphalf + (dt/2)*accel(r[i+1])
```

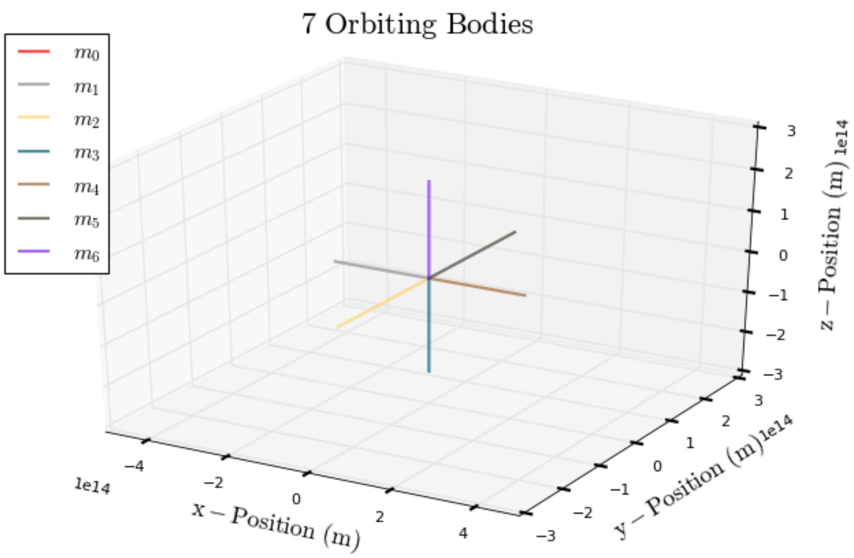
The code is cleaner with 3D arrays.

```
if method == 'rk2':
    for i in tqdm(range(steps)):
        v_iphalf = v[i] + accel(r[i])*(dt/2)
        r_iphalf = r[i] + v[i]*(dt/2)
        v[i+1] = v[i] + accel(r_iphalf)*dt
        r[i+1] = r[i] + v_iphalf*dt
```

```
if method == 'pv':
    for i in tqdm(range(steps)):
        r_iphalf = r[i] + (dt/2)*v[i]
        v[i+1] = v[i] + dt*accel(r_iphalf)
        r[i+1] = r_iphalf + (dt/2)*v[i+1]
```

Checking the Model with Symmetry





These system behaves exactly as your intuition might expect, so we can have confidence in the model.

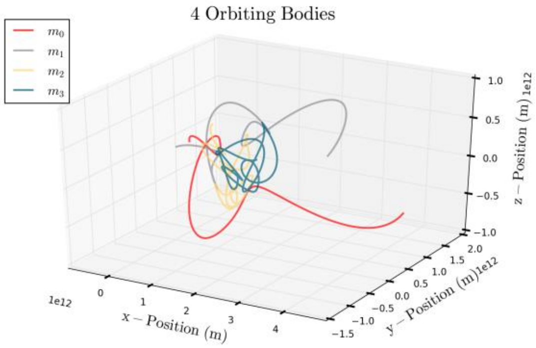
Numerical Results for N=4

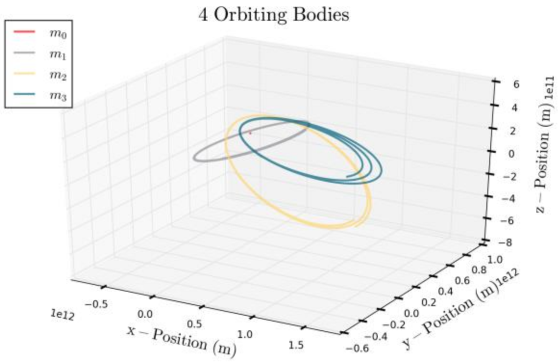
We have two sets of initial conditions:

IC1 produces stable ellipses while IC2 is more chaotic.

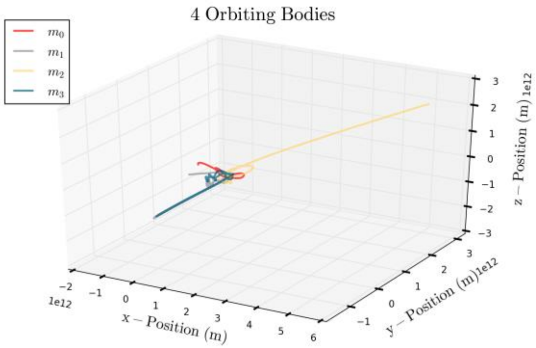
IC2 IC1

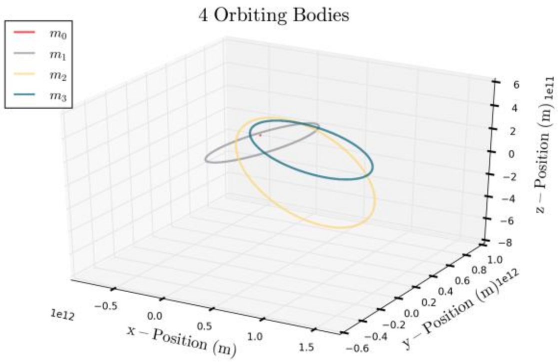
Euler



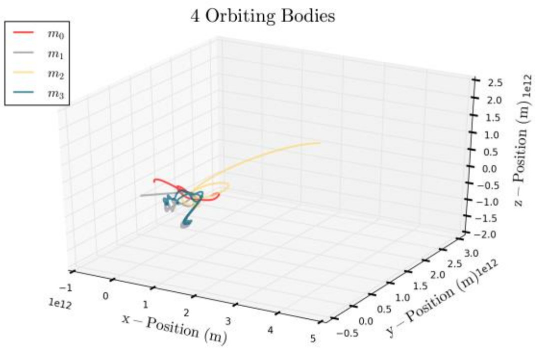


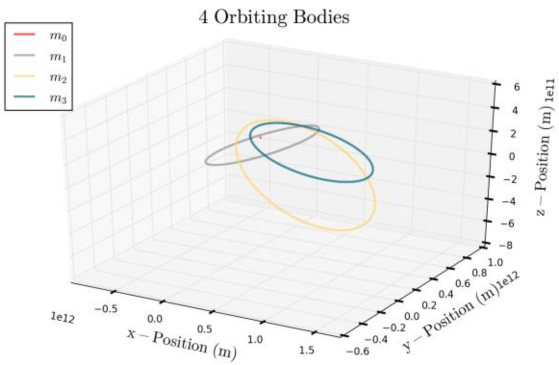
PV





PEFRL





Investigating Energy

A few questions to ponder:

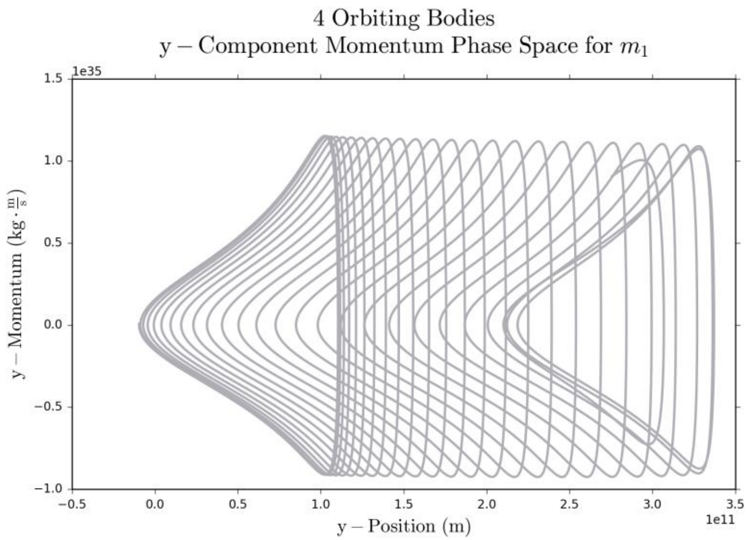
Which one is IC2?

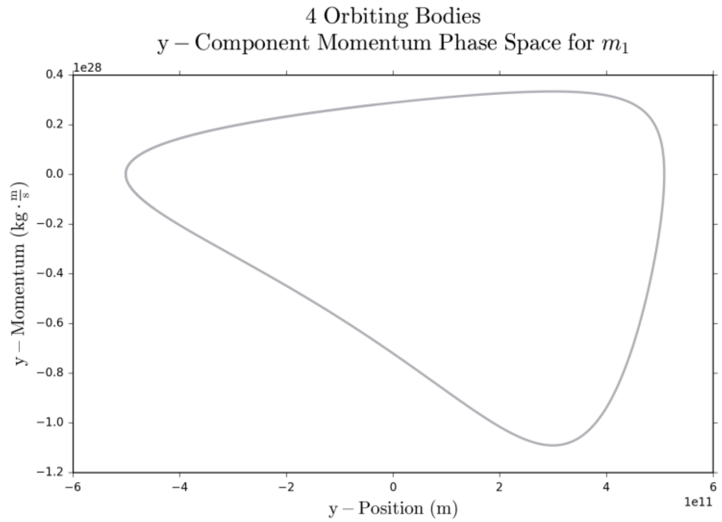
Which one is IC1?

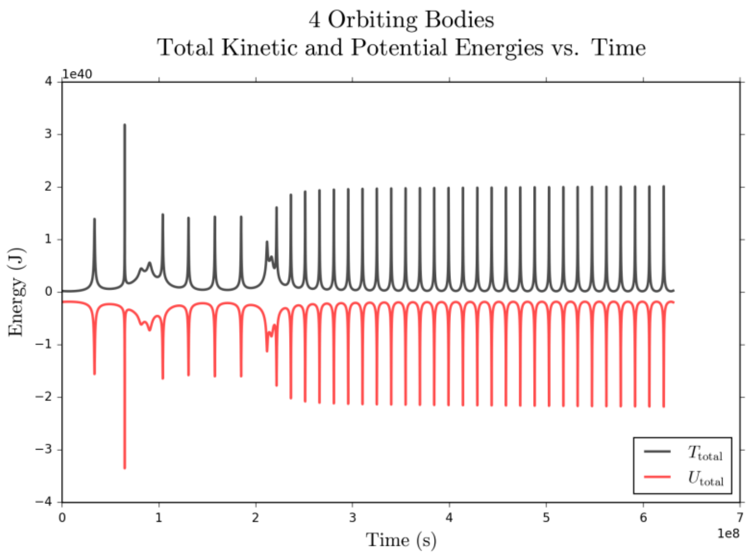
How can we tell?

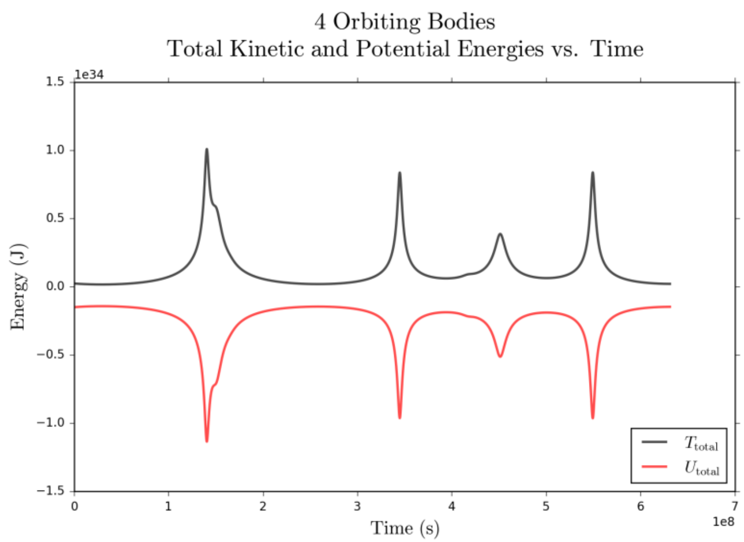
Is energy conserved?

What do the peaks mean?

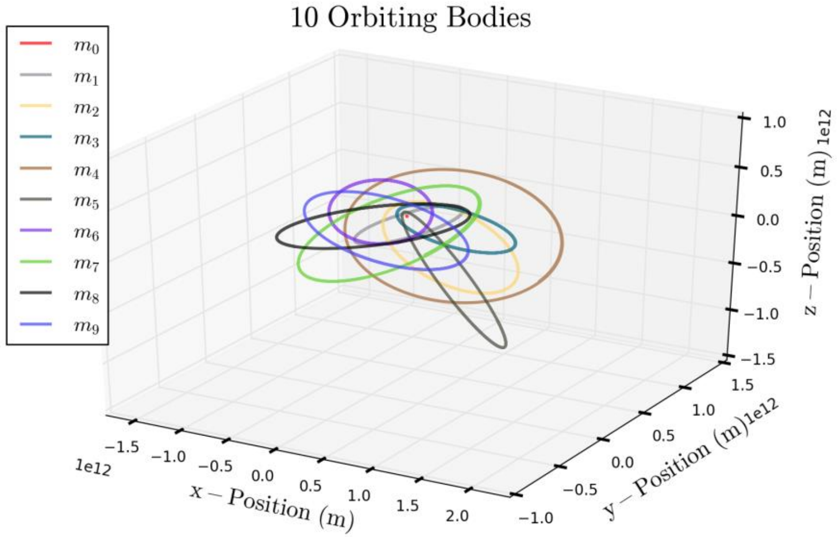


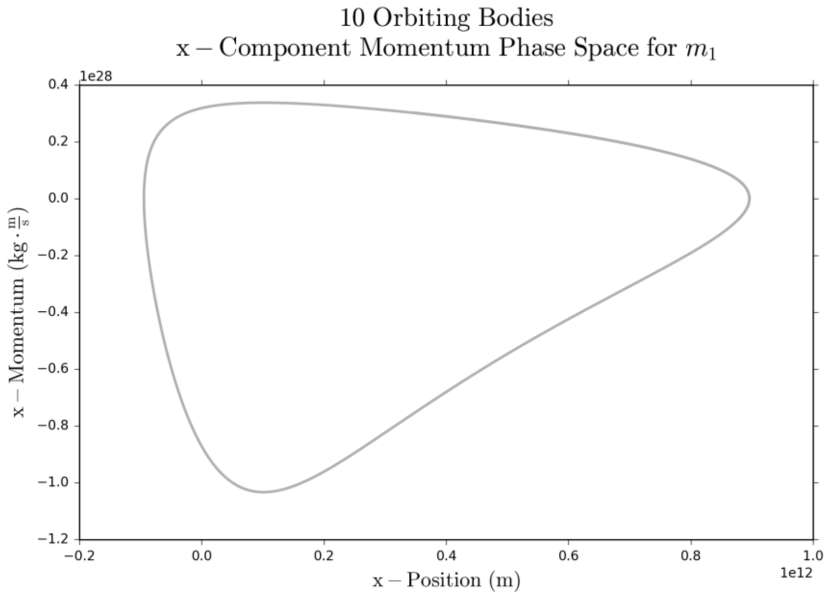


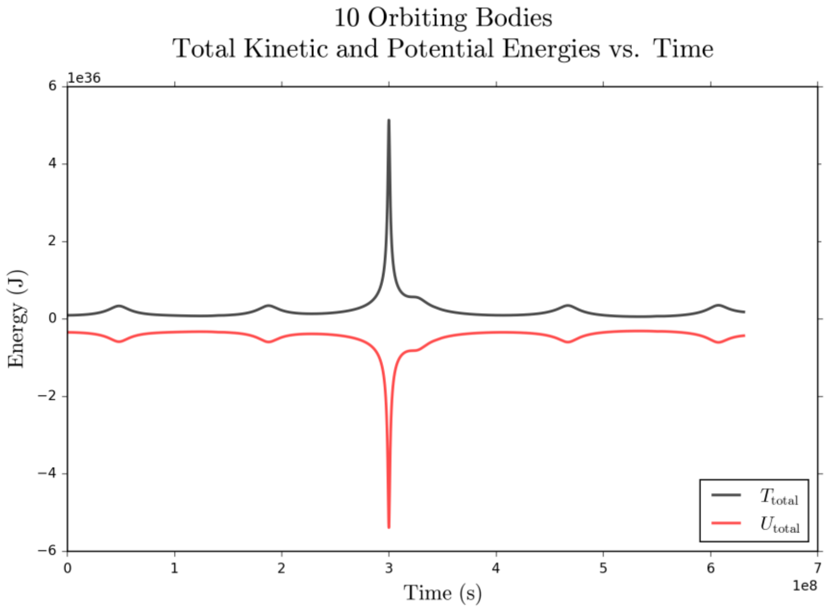


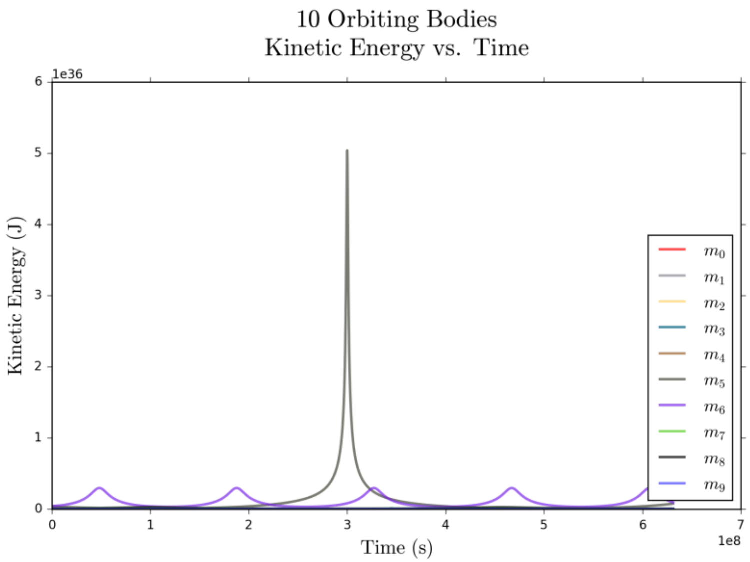


10 Bodies Similar to the Solar System

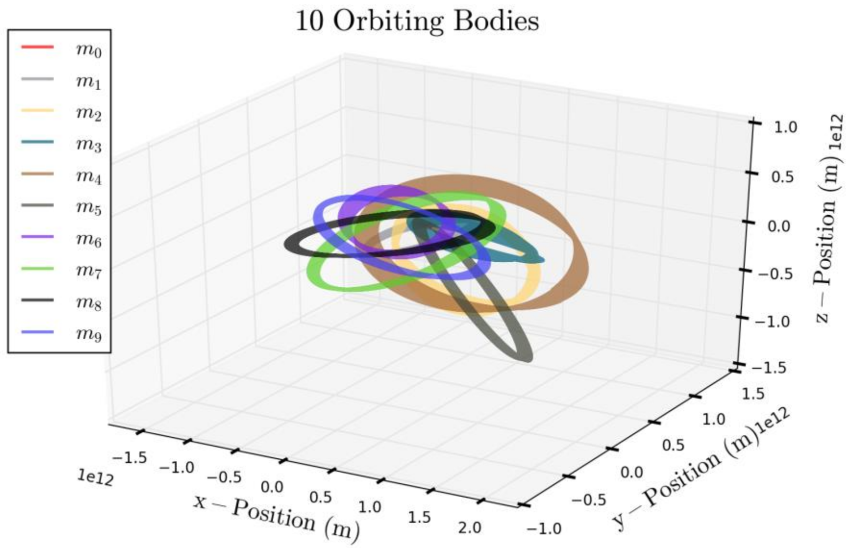


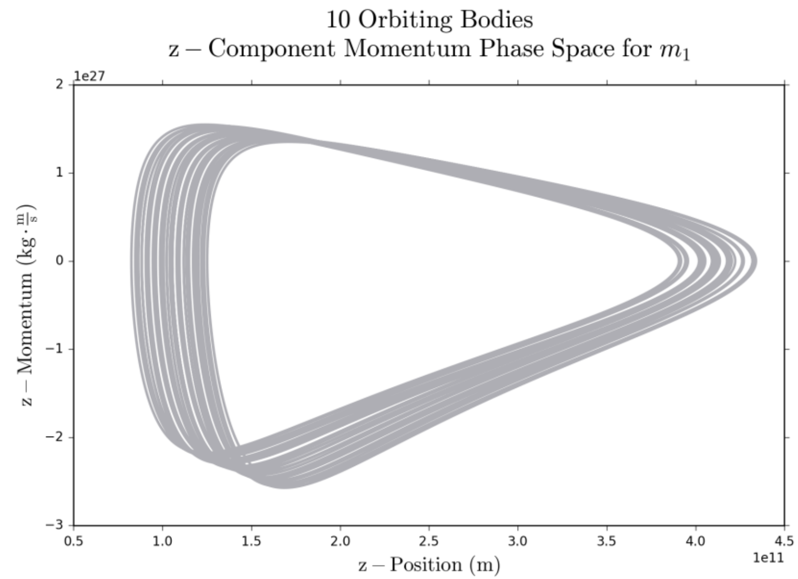


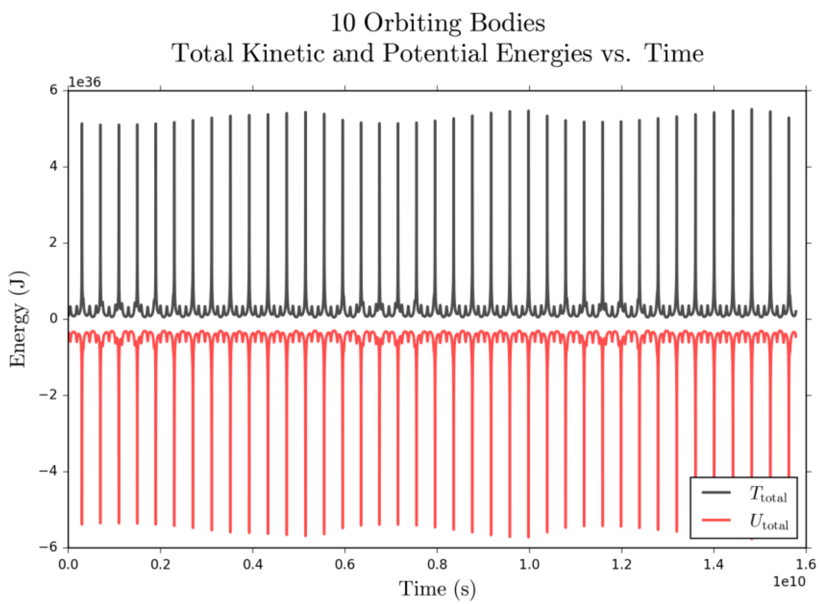


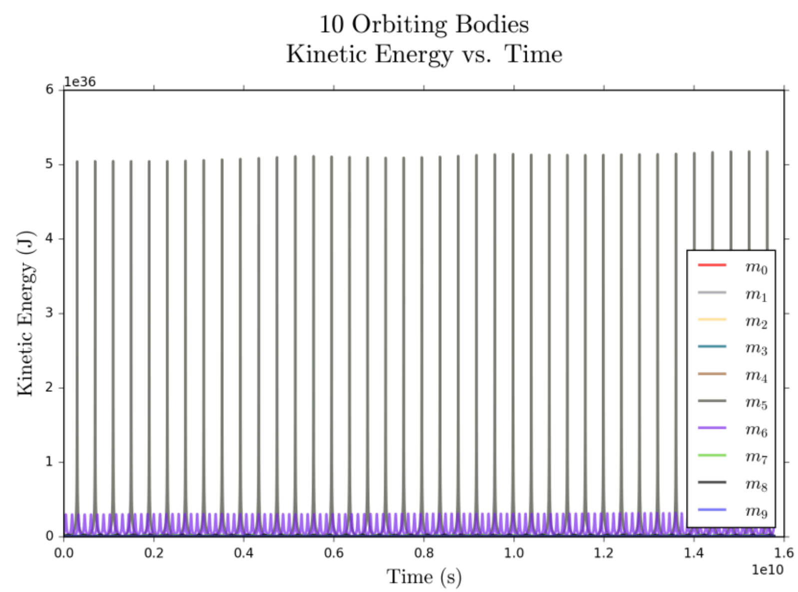


10 Bodies Extended for 500 Years









Thank You! Questions?

References:

General Mathematics

- https://en.wikipedia.org/wiki/N-body_problem

PEFRL and VEFRL Algorithms

- https://arxiv.org/pdf/cond-mat/0110585.pdf

Table 1: Initial Conditions Set 1 (IC1)

body #	mass	x	y	z	v_x	v_y	v_z
1	1.989×10^{30}	1.0	3.0	2.0	0.0	0.0	0.0
2	3.285×10^{23}	6.0	-5.0	4.0	7.0	0.5	2.0
3	4.867×10^{24}	7.0	8.0	-7.0	-4.0	-0.5	-3.0
4	5.972×10^{24}	8.0	6.0	-2.0	7.0	0.5	2.0
5	6.417×10^{23}	8.8	9.8	-6.8	4.8	1.3	4.8
6	1.898×10^{27}	9.8	3.8	-7.8	1.8	1.2	-5.8
7	5.683×10^{26}	-3.8	1.8	4.8	2.8	11.3	1.4
8	8.681×10^{25}	7.8	-2.2	1.8	3.8	10.3	2.4
9	1.024×10^{26}	6.8	-4.1	3.8	4.8	9.3	-1.4
10	1.309×10^{22}	5.8	-9.3	5.8	5.8	0.3	-2.4

Table 2: Initial Conditions Set 2 (IC2)

body #	mass	x	y	z	v_x	v_y	v_z
1	1×10^{30}	1.0	3.0	2.0	-2.0	0.5	5.0
2	2×10^{30}	6.0	-5.0	4.0	7.0	0.5	2.0
3	3×10^{30}	7.0	8.0	-7.0	-4.0	-0.5	-3.0
4	2.5×10^{30}	8.0	6.0	-2.0	7.0	0.5	2.0