

New Stable Periodic Solutions to the n -Body Problem

Robert J. Vanderbei

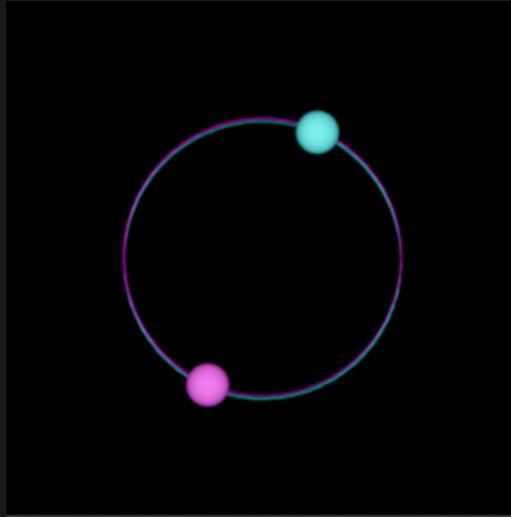
2014 Feb 17

Program in Applied and Computational Mathematics
Princeton University

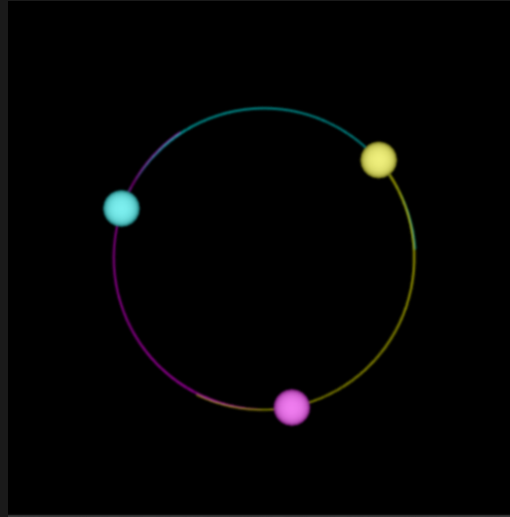
<http://www.princeton.edu/~rvdb>

Lagrange Style Orbits

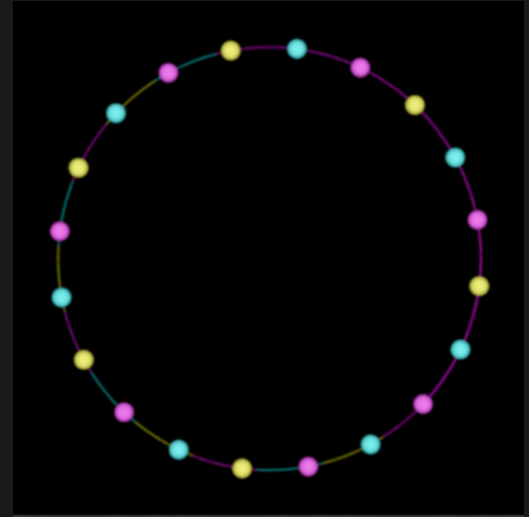
Click anywhere below for WebGL simulation



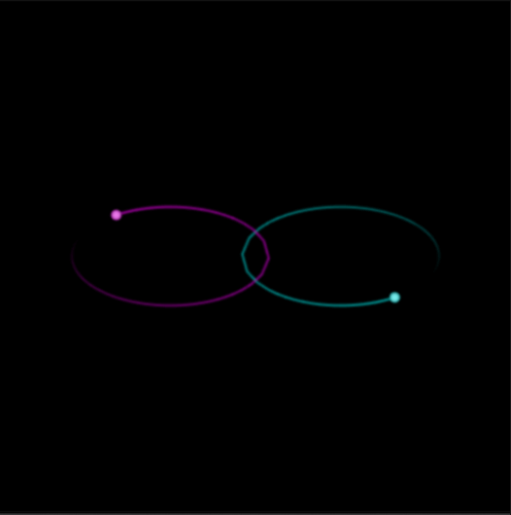
Reset Pause/Run Select an orbit: Lagrange2



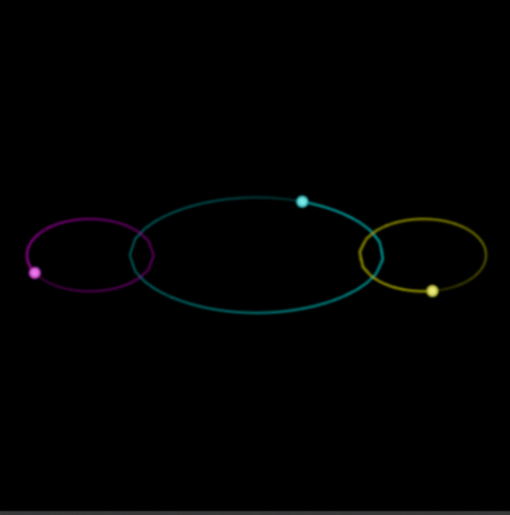
Reset Pause/Run Select an orbit: Lagrange3



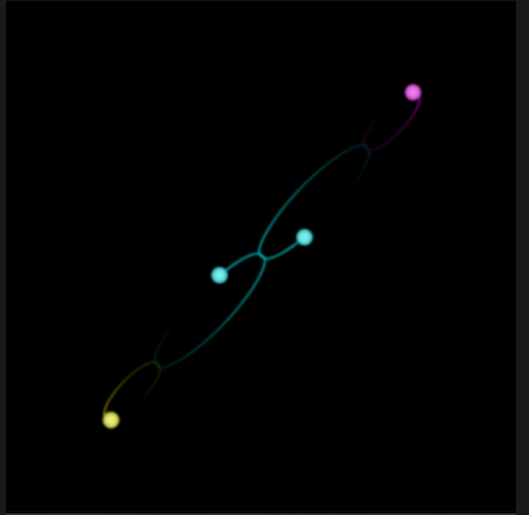
Reset Pause/Run Select an orbit: Lagrange20



Reset Pause/Run Select an orbit: Double Ellipse

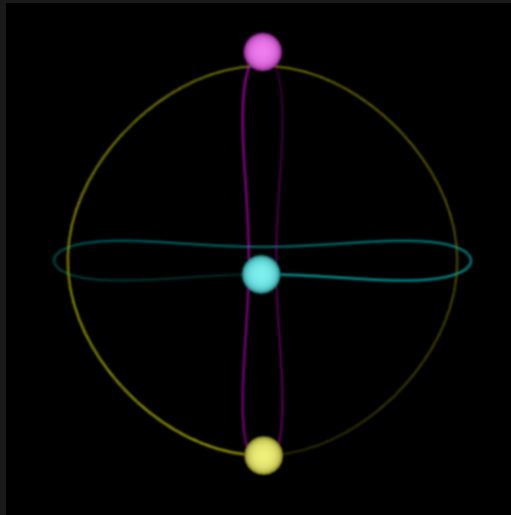


Reset Pause/Run Select an orbit: Triple Ellipse

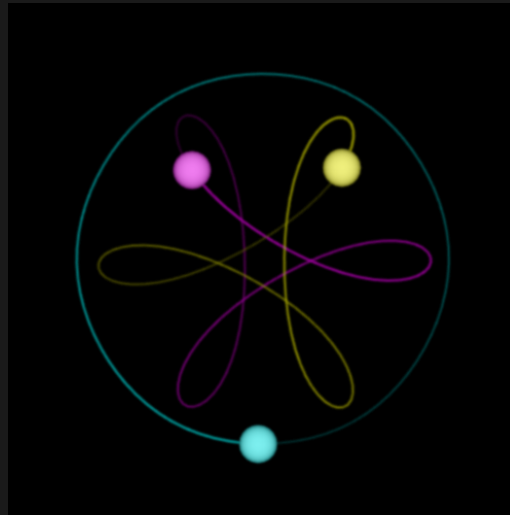


Reset Pause/Run Select an orbit: Quad Ellipse

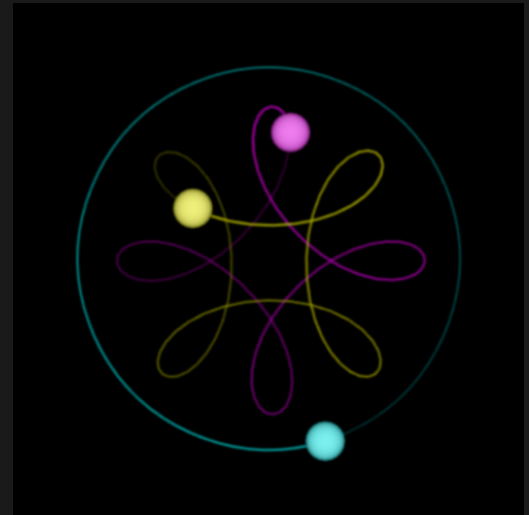
Hill-Type and Double/Doubles



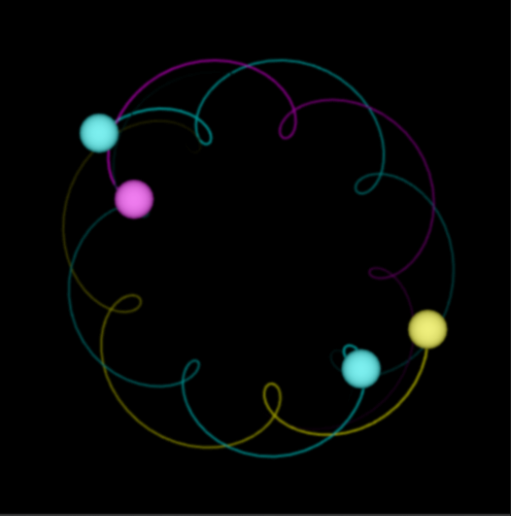
Reset Pause/Run Select an orbit:



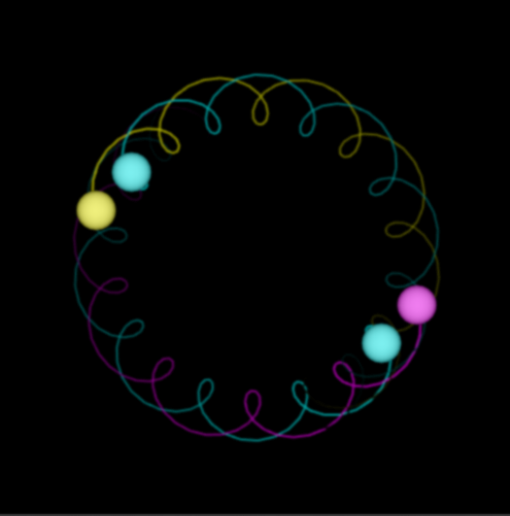
Reset Pause/Run Select an orbit:



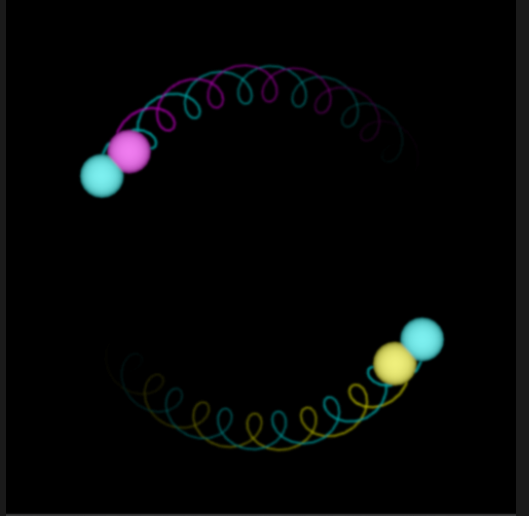
Reset Pause/Run Select an orbit:



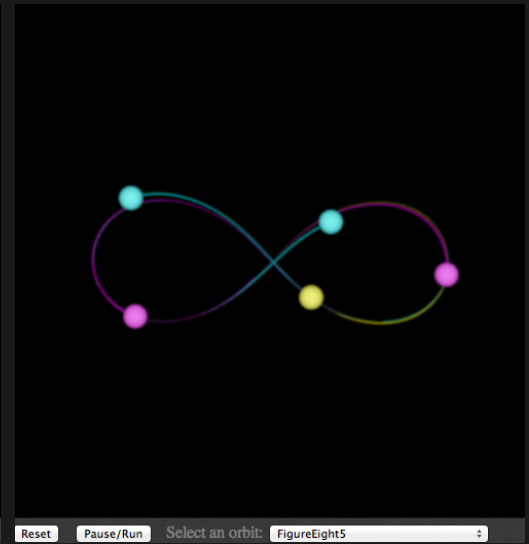
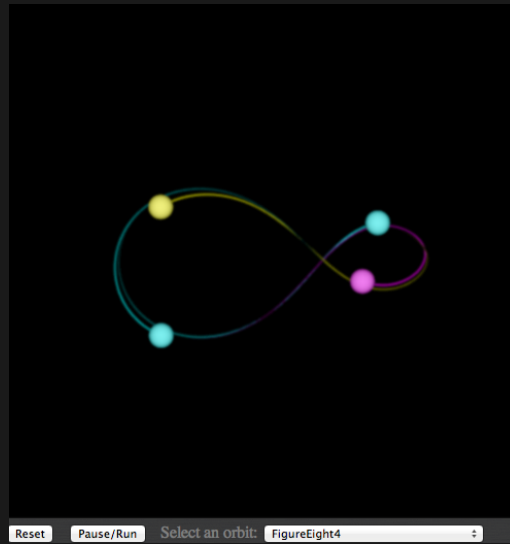
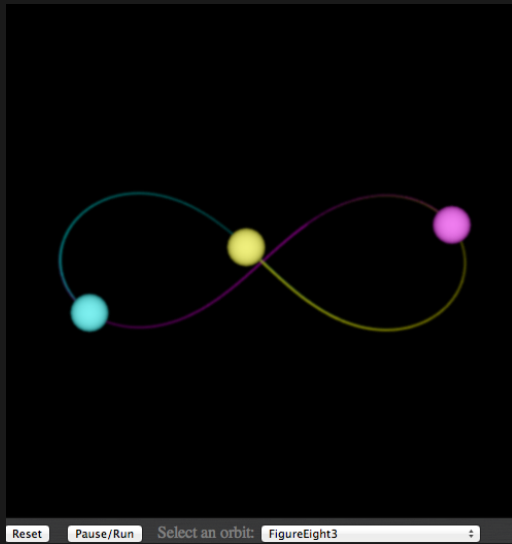
Reset Pause/Run Select an orbit:



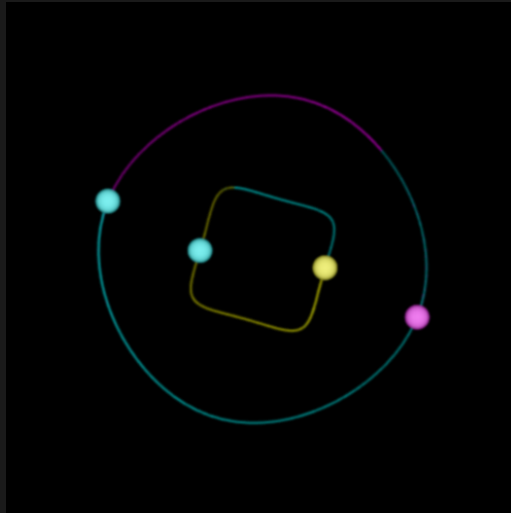
Reset Pause/Run Select an orbit:



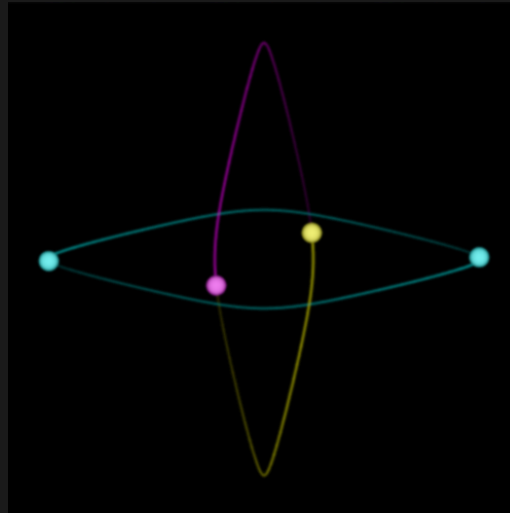
Reset Pause/Run Select an orbit:



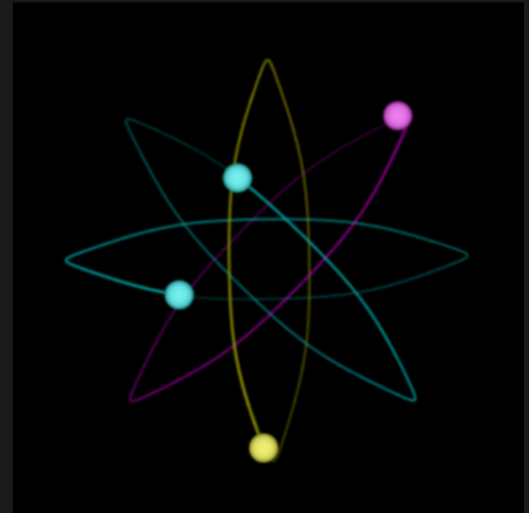
Exotic But Unstable



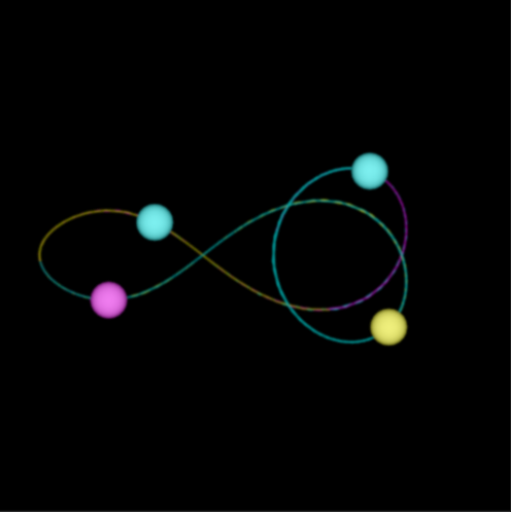
Reset Pause/Run Select an orbit: PlateSaucer4



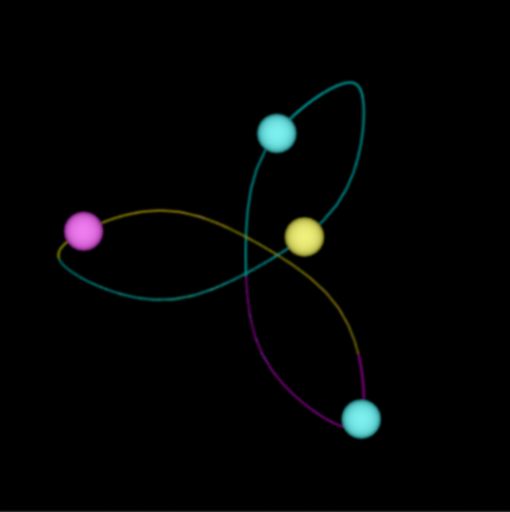
Reset Pause/Run Select an orbit: 1Month1Year



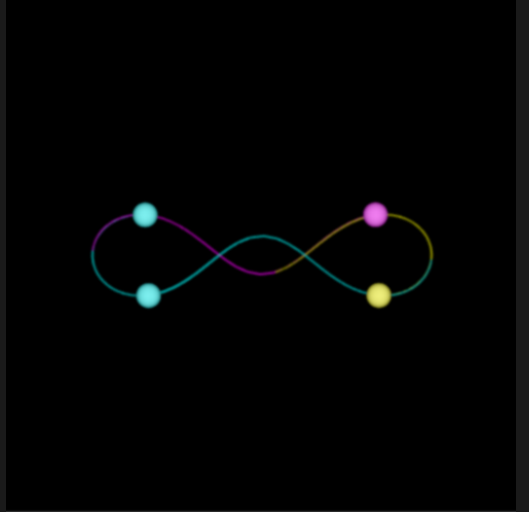
Reset Pause/Run Select an orbit: 1Month1YearAgain



Reset Pause/Run Select an orbit: FoldedTriLoop4

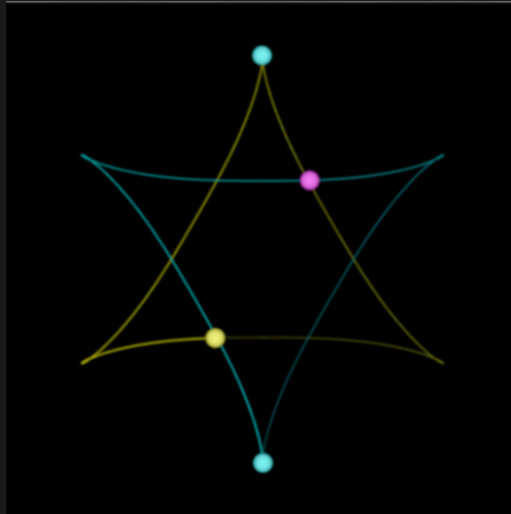


Reset Pause/Run Select an orbit: Trefoil4

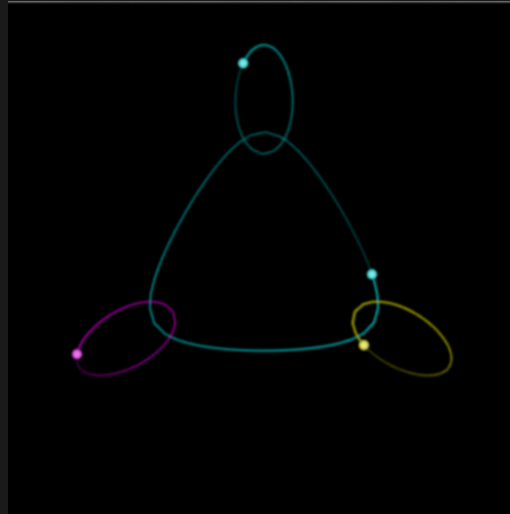


Reset Pause/Run Select an orbit: Braid4

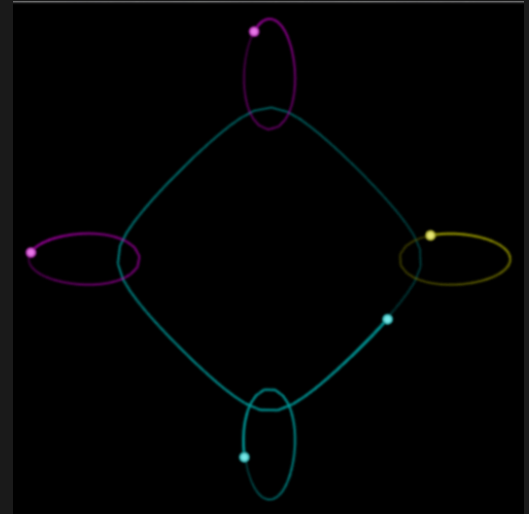
New Ones



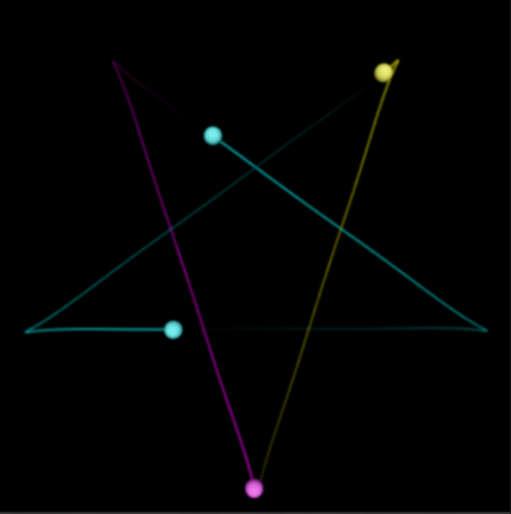
Reset Pause/Run Select an orbit: Star of David



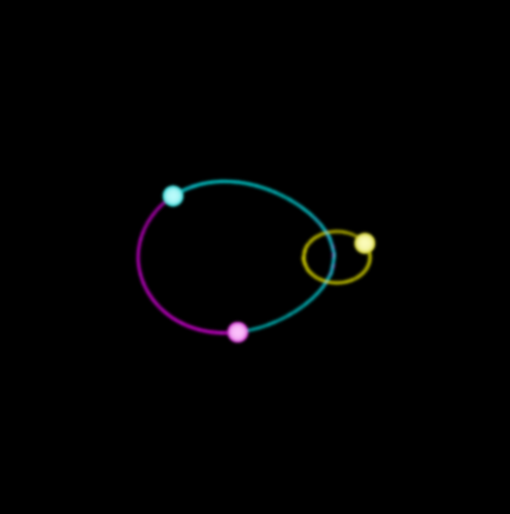
Reset Pause/Run Select an orbit: Triangle



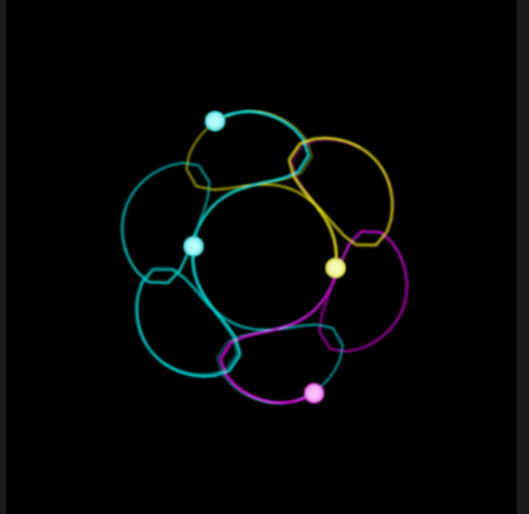
Reset Pause/Run Select an orbit: Four Corners



Reset Pause/Run Select an orbit: five point star



Reset Pause/Run Select an orbit: BinaryEllipse3



Reset Pause/Run Select an orbit: Hexagon

Optimization

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && b \leq h(x) \leq b + r, \\ & && l \leq x \leq u \end{aligned}$$

- *Linear Programming (LP)*: f and h are linear.
 - *Convex Optimization*: f is convex, each h_i is concave, and $r = \infty$.
 - *Nonlinear Optimization*: f and each h_i is assumed to be twice differentiable
-
- Generally, we seek a *local solution* in the vicinity of a given starting point.
 - If problem is convex (which includes LP), any local solution is automatically a *global solution*.

Reference: *Linear Programming: Foundations and Extensions, 4th Edition, Springer-Verlag*

Least Action Principle

Given: n bodies.

Let:

- m_j denote the mass and
- $z_j(t)$ denote the position in $\mathbb{R}^2 = \mathbb{C}$ of body j at time t .

Action Functional:

$$A = \int_0^{2\pi} \left(\sum_j \frac{1}{2} m_j \|\dot{z}_j\|^2 + \sum_{j,k:k < j} \frac{Gm_j m_k}{\|z_j - z_k\|} \right) dt.$$

Equation of Motion

First Variation:

$$\begin{aligned}\delta A &= \int_0^{2\pi} \sum_{\alpha} \left(\sum_j m_j \dot{z}_j^{\alpha} \delta \dot{z}_j^{\alpha} - \sum_{j,k:k < j} G m_j m_k \frac{(z_j^{\alpha} - z_k^{\alpha})(\delta z_j^{\alpha} - \delta z_k^{\alpha})}{\|z_j - z_k\|^3} \right) dt \\ &= - \int_0^{2\pi} \sum_j \sum_{\alpha} \left(m_j \ddot{z}_j^{\alpha} + \sum_{k:k \neq j} G m_j m_k \frac{z_j^{\alpha} - z_k^{\alpha}}{\|z_j - z_k\|^3} \right) \delta z_j^{\alpha} dt\end{aligned}$$

Setting first variation to zero, we get:

$$m_j \ddot{z}_j^{\alpha} = - \sum_{k:k \neq j} G m_j m_k \frac{z_j^{\alpha} - z_k^{\alpha}}{\|z_j - z_k\|^3}, \quad j = 1, 2, \dots, n, \quad \alpha = 1, 2$$

Note: If $m_j = 0$ for some j , then the first order optimality condition reduces to $0 = 0$, which is *not* the equation of motion for a massless body.

Lagrange Orbit

Consider two bodies of equal mass m on a circular orbit of radius r :

$$z(t) = r e^{i\omega t \pm \pi/2}.$$

Differentiating twice, we get

$$\ddot{z}(t) = -r\omega^2 e^{i\omega t \pm \pi/2}.$$

From Newton's laws of motion, we get

$$-r\omega^2 e^{i\omega t \pm \pi/2} = \ddot{z}(t) = -\frac{Gm}{(2r)^2} e^{i\omega t \pm \pi/2}.$$

Solving for r , we get

$$r = \left(\frac{Gm}{4\omega^2} \right)^{1/3}.$$

Double/Double Style Initialization

Start with a pair of bodies each of mass $2m$ in a circular orbit:

$$z(t) = \pm r_0 e^{i\omega_0 t}$$

where

$$r_0 = \left(\frac{2Gm}{4\omega_0^2} \right)^{1/3} .$$

Then assume that each mass is actually a 2-body system in a circular orbit with radius r_1 with given initial phase shifts:

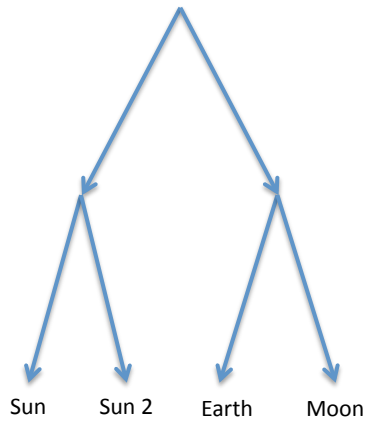
$$z(t) = \pm r_0 e^{i\omega_0 t} \pm r_1 e^{i\omega_1 t + i\phi_{\pm}} .$$

Here,

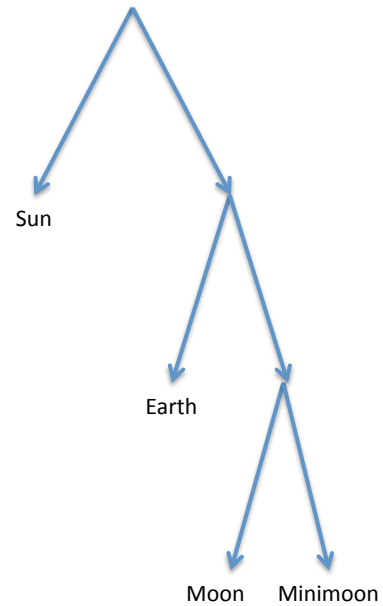
$$r_1 = \left(\frac{Gm}{4\omega_1^2} \right)^{1/3} .$$

Hierarchies of 2-Body Systems

Double/Double



Mini/Month/Year



An AMPL Model

```
param N := 4;      # number of masses
param M := 40000; # number of terms in numerical approx to integral

param G := 1;

param period := 2*pi/omega0; # temporal length of the orbit segment
param dt := period / M;

var x {j in 0..N-1, i in 0..M-1} >= -5, <= 5;
var y {j in 0..N-1, i in 0..M-1} >= -5, <= 5;

var xdot {j in 0..N-1, i in 0..M-1} = (x[j,(i+1) mod M]-x[j,i])/dt;
var ydot {j in 0..N-1, i in 0..M-1} = (y[j,(i+1) mod M]-y[j,i])/dt;

var K {i in 0..M-1}
    = 0.5 * sum {j in 0..N-1} (xdot[j,i]^2 + ydot[j,i]^2);

var P {i in 0..M-1}
    = - sum {j in 0..N-1, k in 0..N-1: k<j}
        1/sqrt((x[j,i]-x[k,i])^2 + (y[j,i]-y[k,i])^2);

minimize Action: sum {i in 0..M-1} (K[i] - P[i])*dt;
```

A Double/Double Initialization

```
param omega0 := 1./2;
param omega1 := 1./1;
param phi_p := -pi/2;
param phi_m := pi/2;
param r0 := (G/(2*omega0^2))^(1./3.);
param r1 := (G/(4*omega1^2))^(1./3.);
param t {i in 0..M-1} := i*dt;

let {k in 0..M-1} x[0,k] := r0*cos(omega0*t[k]) + r1*sin(-omega1*t[k] + phi_p);
let {k in 0..M-1} y[0,k] := r0*sin(omega0*t[k]) + r1*cos(-omega1*t[k] + phi_p);

let {k in 0..M-1} x[1,k] := r0*cos(omega0*t[k]) - r1*sin(-omega1*t[k] + phi_p);
let {k in 0..M-1} y[1,k] := r0*sin(omega0*t[k]) - r1*cos(-omega1*t[k] + phi_p);

let {k in 0..M-1} x[2,k] := -r0*cos(omega0*t[k]) + r1*cos(-omega1*t[k] + phi_m);
let {k in 0..M-1} y[2,k] := -r0*sin(omega0*t[k]) + r1*sin(-omega1*t[k] + phi_m);

let {k in 0..M-1} x[3,k] := -r0*cos(omega0*t[k]) - r1*cos(-omega1*t[k] + phi_m);
let {k in 0..M-1} y[3,k] := -r0*sin(omega0*t[k]) - r1*sin(-omega1*t[k] + phi_m);

solve;
```

Limitations of the Model

- The integral gets discretized to a finite sum.
- Masses must be positive.
- Solutions can occur at local maxima and at saddle points. Looking only for local minima, we miss these.

Alternate Approach: Solve Equations of Motion

Instead of minimizing the action functional, which is an unconstrained optimization problem...

$$\begin{aligned} &\text{minimize} && \int_0^{2\pi} \left(\sum_j \frac{1}{2} m_j \|\dot{z}_j\|^2 + \sum_{j,k:k < j} \frac{Gm_j m_k}{\|z_j - z_k\|} \right) dt, \\ &\text{subject to} && \text{no constraints,} \end{aligned}$$

how about simply looking for trajectories that satisfy Newton's laws:

$$\text{minimize } 0,$$

$$\text{subject to } m_j \ddot{z}_j^\alpha = - \sum_{k:k \neq j} Gm_j m_k \frac{z_j^\alpha - z_k^\alpha}{\|z_j - z_k\|^3}, \quad j = 1, 2, \dots, n, \quad \alpha = 1, 2.$$

AMPL Model for the Equations of Motion

```
var x {i in 0..N-1, j in 0..M-1} >= -5, <= 5;
var y {i in 0..N-1, j in 0..M-1} >= -5, <= 5;

var xdot {i in 0..N-1, j in 0..M-1} = (x[i,(j+1) mod M]-x[i,j])/dt;
var ydot {i in 0..N-1, j in 0..M-1} = (y[i,(j+1) mod M]-y[i,j])/dt;

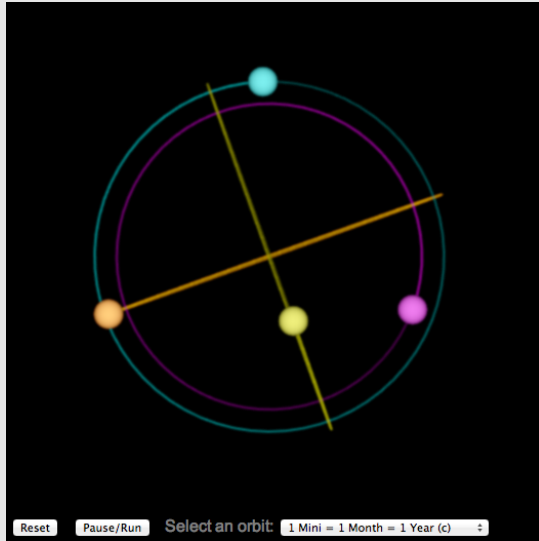
var xdot2 {i in 0..N-1, j in 0..M-1} = (xdot[i,j]-xdot[i,(M+j-1) mod M])/dt;
var ydot2 {i in 0..N-1, j in 0..M-1} = (ydot[i,j]-ydot[i,(M+j-1) mod M])/dt;

minimize zero: 0;

subject to F_equals_ma_x {i in 0..N-1, k in 0..M-1}:
    xdot2[i,k]
    = sum {j in 0..N-1: j != i}
        (x[j,k]-x[i,k]) / ((x[j,k]-x[i,k])^2+(y[j,k]-y[i,k])^2)^(3/2);

subject to F_equals_ma_y {i in 0..N-1, k in 0..M-1}:
    ydot2[i,k]
    = sum {j in 0..N-1: j != i}
        (y[j,k]-y[i,k]) / ((x[j,k]-x[i,k])^2+(y[j,k]-y[i,k])^2)^(3/2);
```

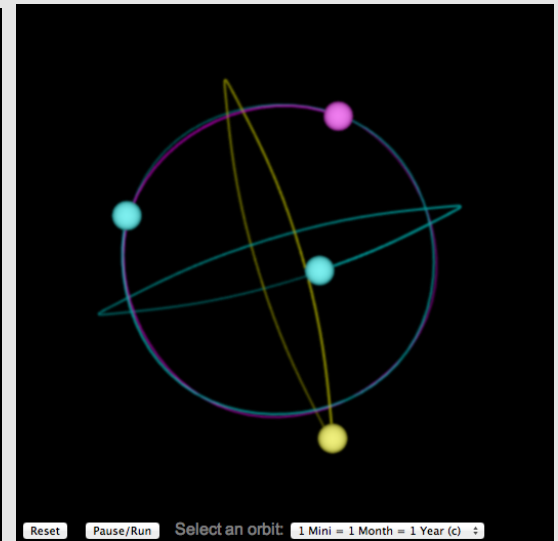
New Solution Via Equations of Motion



A Mini/Month/Year Design



Action Minimization



Equations of Motion

NOTE: Action minimization found an orbit. But, it is immediately unstable as the middle figure shows.

Sensitivity Analysis

Let

$$\xi^*(t) = \begin{bmatrix} x^*(t) \\ v^*(t) \end{bmatrix}$$

be a solution to

$$\dot{\xi} = A(\xi)$$

where

$$A \left(\begin{bmatrix} x(t) \\ v(t) \end{bmatrix} \right) = \begin{bmatrix} v(t) \\ a(x(t)) \end{bmatrix}$$

and

$$a(x) = \begin{bmatrix} a_1(x) \\ \vdots \\ a_n(x) \end{bmatrix}$$

and

$$a_j(x) = - \sum_{k:k \neq j} \frac{x_j - x_k}{\|x_j - x_k\|^2}, \quad j = 1, 2, \dots, n.$$

Note: we've chosen units so that $G = 1$ and all masses are unit masses.

Sensitivity Analysis —Continued

Consider a nearby solution $\xi(t)$:

$$\begin{aligned}\dot{\xi}(t) &= A(\xi(t)) \\ &\approx A(\xi^*(t)) + A'(\xi^*(t))(\xi(t) - \xi^*(t)) \\ &= \dot{\xi}^*(t) + A'(\xi^*(t))(\xi(t) - \xi^*(t)).\end{aligned}$$

Put $\Delta\xi = \xi - \xi^*$. Then $\dot{\Delta\xi} = A'(\xi^*(t))\Delta\xi$. A finite difference approximation yields

$$\begin{aligned}\Delta\xi(t+h) &= \Delta\xi(t) + \Delta t A'(\xi^*(t))\Delta\xi(t) \\ &= (I + \Delta t A'(\xi^*(t))) \Delta\xi(t).\end{aligned}$$

Iterating around one period, we get:

$$\Delta\xi(T) = \left(\prod_{i=0}^{n-1} (I + \Delta t A'(\xi^*(t_i))) \right) \Delta\xi(0),$$

where $\Delta t = T/n$ and $t_i = i\Delta t$.

Linear Stability

Stability: all eigenvalues of

$$\Lambda = \prod_{i=0}^{n-1} (I + \Delta t A'(\xi^*(t_i)))$$

must be at most one in magnitude.

Numerical Results – Leap-Frog Stable Orbits

| Name | $\max(\lambda_i(\Lambda))$ |
|------------------|----------------------------|
| Lagrange2 | 1.0000 |
| Double Ellipse | 1.0001 |
| Triple Ellipse | 1.0000 |
| Quad Ellipse | 5.7921 |
| Ducati3 | 1.0000 |
| Broucke-Henon | 1.0000 |
| Hill 2 months/yr | 1.2430 |
| Hill 3 months/yr | 1.4187 |
| Star of David | 7.1140 |
| DoubleDouble5 | 18.8167 |
| DoubleDouble10 | 8.8401 |
| DoubleDouble20 | 1.0000 |
| FigureEight3 | 1.0000 |
| Five Point Star | 1.0000 |

Numerical Results – Leap-Frog Unstable Orbits

| Name | $\max(\lambda_i(\Lambda))$ |
|------------------|----------------------------|
| Lagrange3 | 85.0138 |
| Lagrange20 | 3.3644e+13 |
| FigureEight4 | 168.38 |
| FigureEight5 | 2126.0 |
| PlateSaucer4 | 3658 |
| 1Month1Year | 17.5830 |
| 1Month1YearAgain | 1.7697 |
| FoldedTriLoop4 | 7.4657e+04 |
| Trefoil4 | 4.1726e+04 |
| Braid4 | 7.7932e+03 |
| Triangle | 25.6724 |
| Four Corners | 269.0443 |
| Binary Ellipse | 1.4053 |
| Hexagon | 9.9829e+04 |

Making Stability an Objective

To the *Equations of Motion* model, add these perturbation parameters:

```
param dx := 1e-3;
param dy := 1e-3;
param dvx := 1e-3;
param dvy := 1e-3;
```

These new variables and objective function:

```
# perturbed trajectories
var xp {l in 0..4*N-1, i in 0..N-1, j in -1..M};
var yp {l in 0..4*N-1, i in 0..N-1, j in -1..M};
var xp_dot {l in 0..4*N-1, i in 0..N-1, j in -1..M-1} = (xp[l,i,j+1]-xp[l,i,j])/dt;
var yp_dot {l in 0..4*N-1, i in 0..N-1, j in -1..M-1} = (yp[l,i,j+1]-yp[l,i,j])/dt;
var xp_dot2 {l in 0..4*N-1, i in 0..N-1, j in 0..M-1} = (xp_dot[l,i,j]-xp_dot[l,i,j-1])/dt;
var yp_dot2 {l in 0..4*N-1, i in 0..N-1, j in 0..M-1} = (yp_dot[l,i,j]-yp_dot[l,i,j-1])/dt;

minimize instability:
  sum {l in 0..N-1, j in 0..N-1: l != j} (xp[l,j,M-1] - x[j,M-1])^2
+ sum {
      j in 0..N-1
    } (xp[j,j,M-1] - dx- x[j,M-1])^2
+ sum {l in 0..N-1, j in 0..N-1: l != j} (yp[l,j,M-1] - y[j,M-1])^2
+ sum {
      j in 0..N-1
    } (yp[j,j,M-1] - dy- y[j,M-1])^2
+ sum {l in 0..N-1, j in 0..N-1: l != j} (xp_dot[l,j,M-1] -xdot[j,M-1])^2
+ sum {
      j in 0..N-1
    } (xp_dot[j,j,M-1]-dvx-xdot[j,M-1])^2
+ sum {l in 0..N-1, j in 0..N-1: l != j} (yp_dot[l,j,M-1] -ydot[j,M-1])^2
+ sum {
      j in 0..N-1
    } (yp_dot[j,j,M-1]-dvy-ydot[j,M-1])^2
```

And...

And these constraints defining the perturbed trajectories:

```
subject to F_eq_ma_x_pert {l in 0..4*N-1, i in 0..N-1, k in 0..M-1}:
    xp_dot2[l,i,k]
    = sum {j in 0..N-1: j != i}
        (xp[l,j,k]-xp[l,i,k]) / ((xp[l,j,k]-xp[l,i,k])^2+(yp[l,j,k]-yp[l,i,k])^2)^(3/2);

subject to F_eq_ma_y_pert {l in 0..4*N-1, i in 0..N-1, k in 0..M-1}:
    yp_dot2[l,i,k]
    = sum {j in 0..N-1: j != i}
        (yp[l,j,k]-yp[l,i,k]) / ((xp[l,j,k]-xp[l,i,k])^2+(yp[l,j,k]-yp[l,i,k])^2)^(3/2);

subject to x_pert_init {i in 0..N-1}: xp[i,i,0] = x[i,0] + dx;
subject to y_pert_init {i in 0..N-1}: yp[i,i,0] = y[i,0] + dy;

subject to xdot_pert_init {i in 0..N-1}: xp_dot[i,i,0] = xdot[i,0] + dvx;
subject to ydot_pert_init {i in 0..N-1}: yp_dot[i,i,0] = ydot[i,0] + dvy;
```

Numerical Results

| Name | Obj. Func. Value |
|---------------------|------------------|
| Ducati3 | 1.4e-5 |
| Star of David | 1.6e-5 |
| 1Month1Year | 1.4e-5 |
| 1Mini1Month1Year | 5.5e-5 |
| 1Mini1Month1Year2 | 2.5e-1 |
| 1Mini1Month1Year(c) | 4.2e-2 |
| Five Point Star | 8.9e-6 |

Leap-Frog Midpoint Integrator (using a Spring)

Differential equation:

$$\ddot{x} = -x$$

Given: $x(0), v(0)$

Compute:

$$\begin{aligned}a(0) &= -x(0) \\v(h/2) &= v(0) + (h/2)a(0)\end{aligned}$$

For $t = h, 2h, \dots$

$$\begin{aligned}a(t) &= -x(t) \\v(t + h/2) &= v(t - h/2) + ha(t) \\x(t + h) &= x(t) + hv(t + h/2)\end{aligned}$$

| t | x | v | a |
|-----|-------|--------|--------|
| 0.0 | 1.000 | 0.000 | -1.000 |
| | | -0.050 | |
| 0.1 | 0.995 | | -0.995 |
| | | -0.150 | |
| 0.2 | 0.980 | | -0.980 |
| | | -0.248 | |
| 0.3 | 0.955 | | -0.955 |
| | | -0.343 | |
| 0.4 | 0.921 | | -0.921 |
| | | -0.435 | |
| 0.5 | 0.877 | | -0.877 |

The Midpoint Integrator

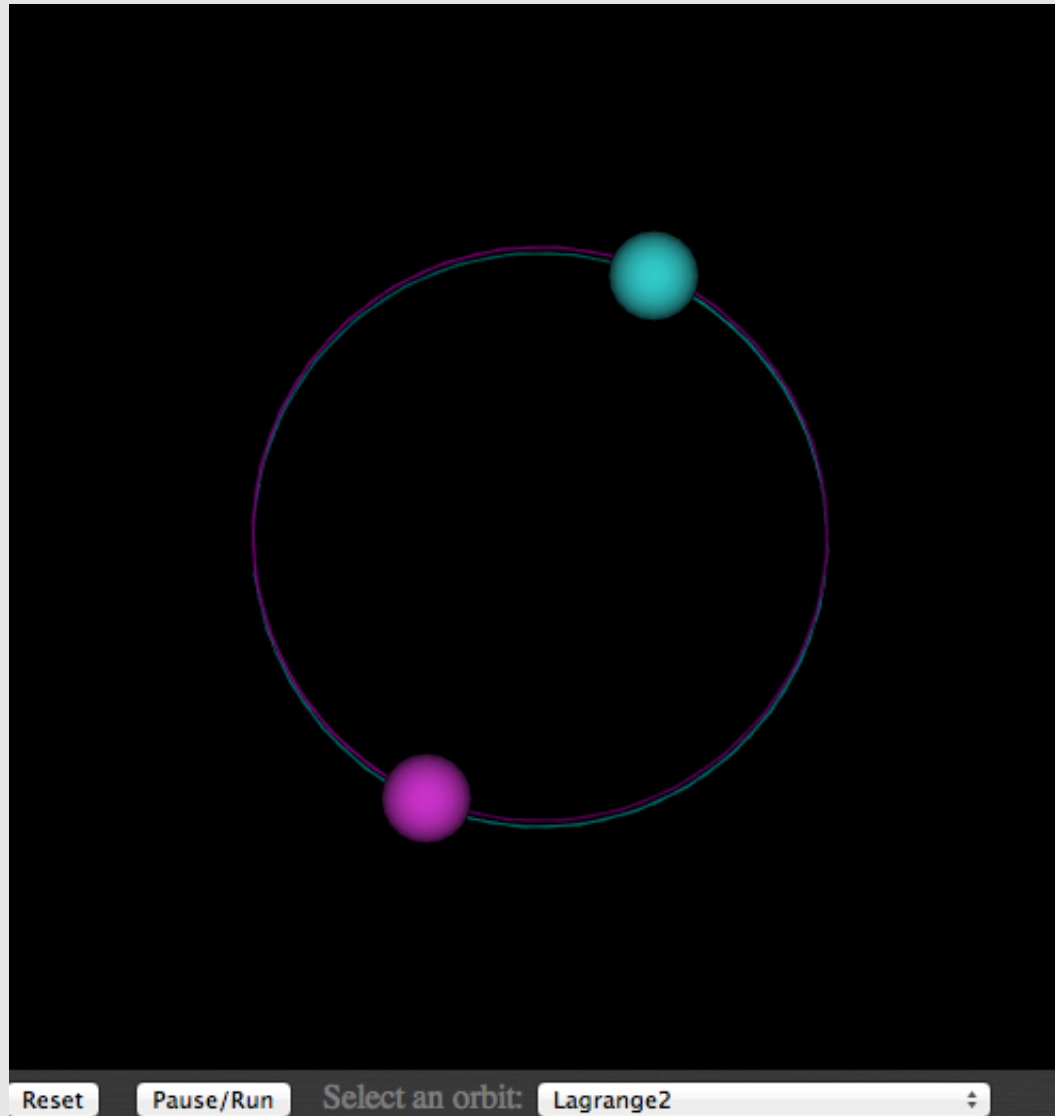
```
if (integrator == MIDPOINT) {
    for (j=0; j<n; j++) {
        p[j].x += p[j].vx * dt;
        p[j].y += p[j].vy * dt;
    }
    for (j=0; j<n; j++) {
        p[j].ax = 0; p[j].ay = 0;
        for (i=0; i<n; i++) {
            if (i != j) {
                double r3 = dist3(p[i], p[j]);
                if (r3<r03) r3=r03;
                p[j].ax -= G * p[i].m * (p[j].x - p[i].x)/r3;
                p[j].ay -= G * p[i].m * (p[j].y - p[i].y)/r3;
            }
        }
    }
    for (j=0; j<n; j++) {
        p[j].vx += p[j].ax * dt;
        p[j].vy += p[j].ay * dt;
    }
}
```

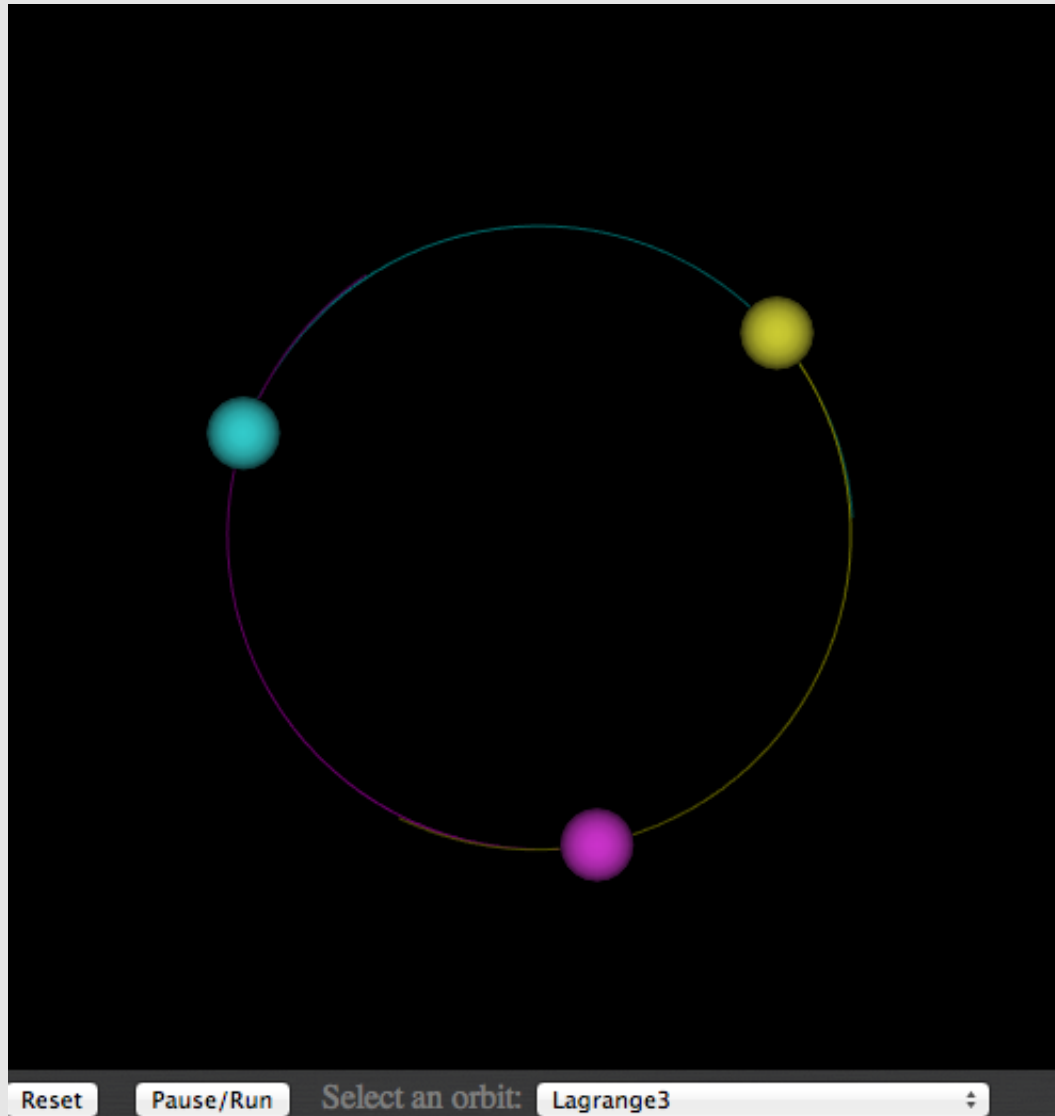
THANK YOU!

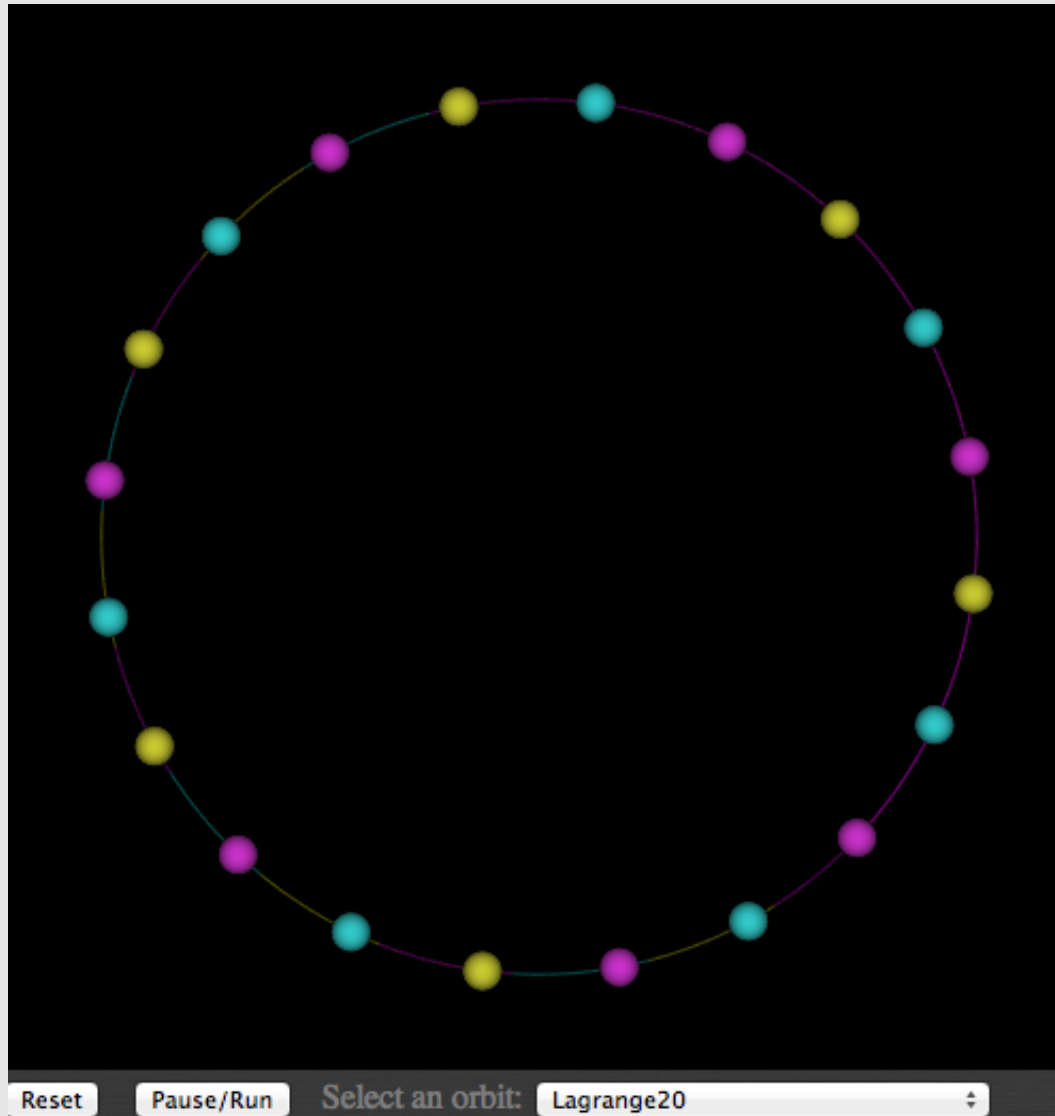
Backup Slides

Second Variation

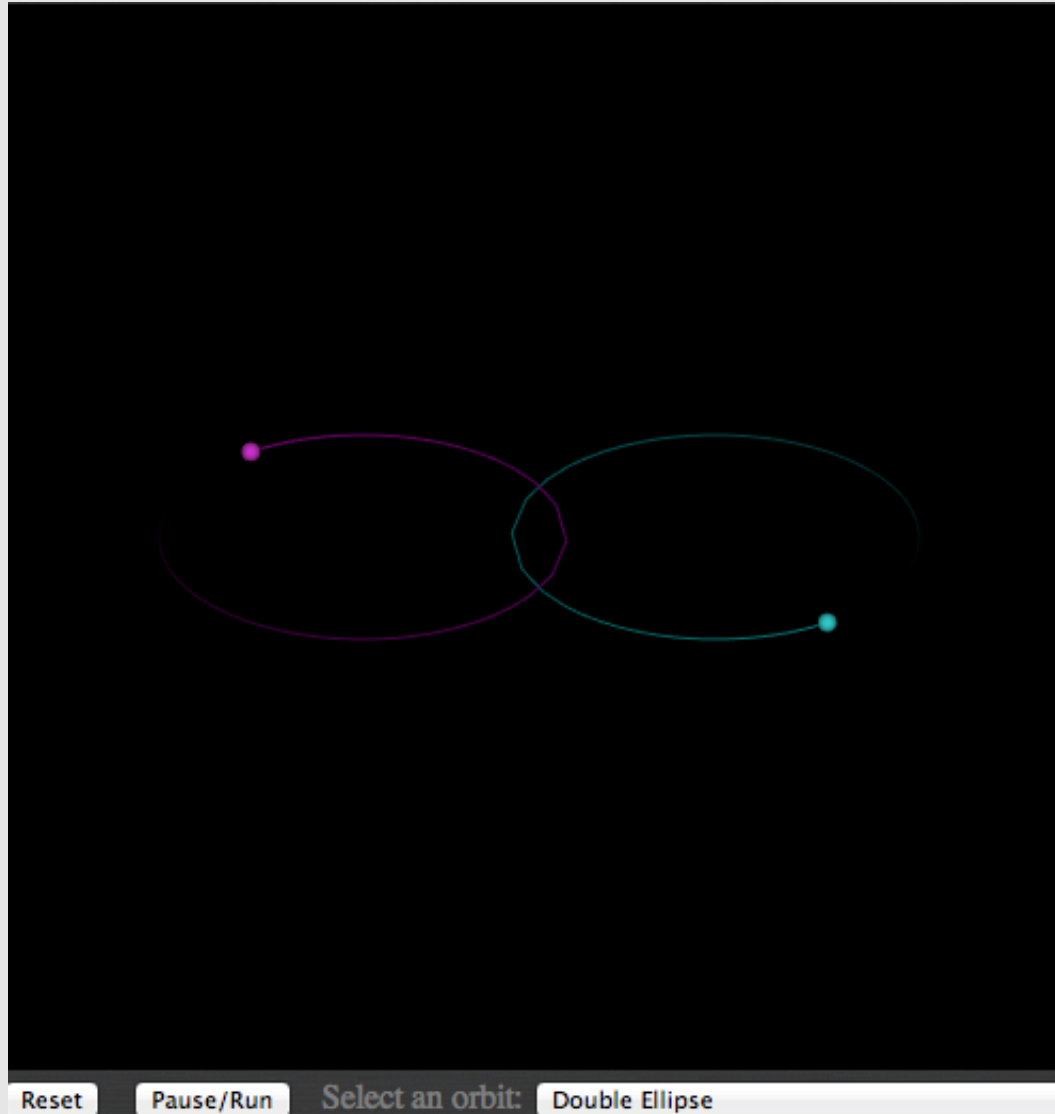
$$\begin{aligned}\delta^2 A &= \int_0^{2\pi} \sum_j \sum_\alpha (\dot{\delta z}_j^\alpha)^2 dt \\ &+ 3 \int_0^{2\pi} \sum_{j,k:j \neq k} \sum_{\alpha,\beta} \frac{(z_j^\alpha - z_k^\alpha)(z_j^\beta - z_k^\beta)(\delta z_j^\beta - \delta z_k^\beta)}{\|z_j - z_k\|^5} \delta z_j^\alpha dt \\ &- \int_0^{2\pi} \sum_{j,k:j \neq k} \sum_\alpha \frac{\delta z_j^\alpha - \delta z_k^\alpha}{\|z_j - z_k\|^3} \delta z_j^\alpha dt\end{aligned}$$



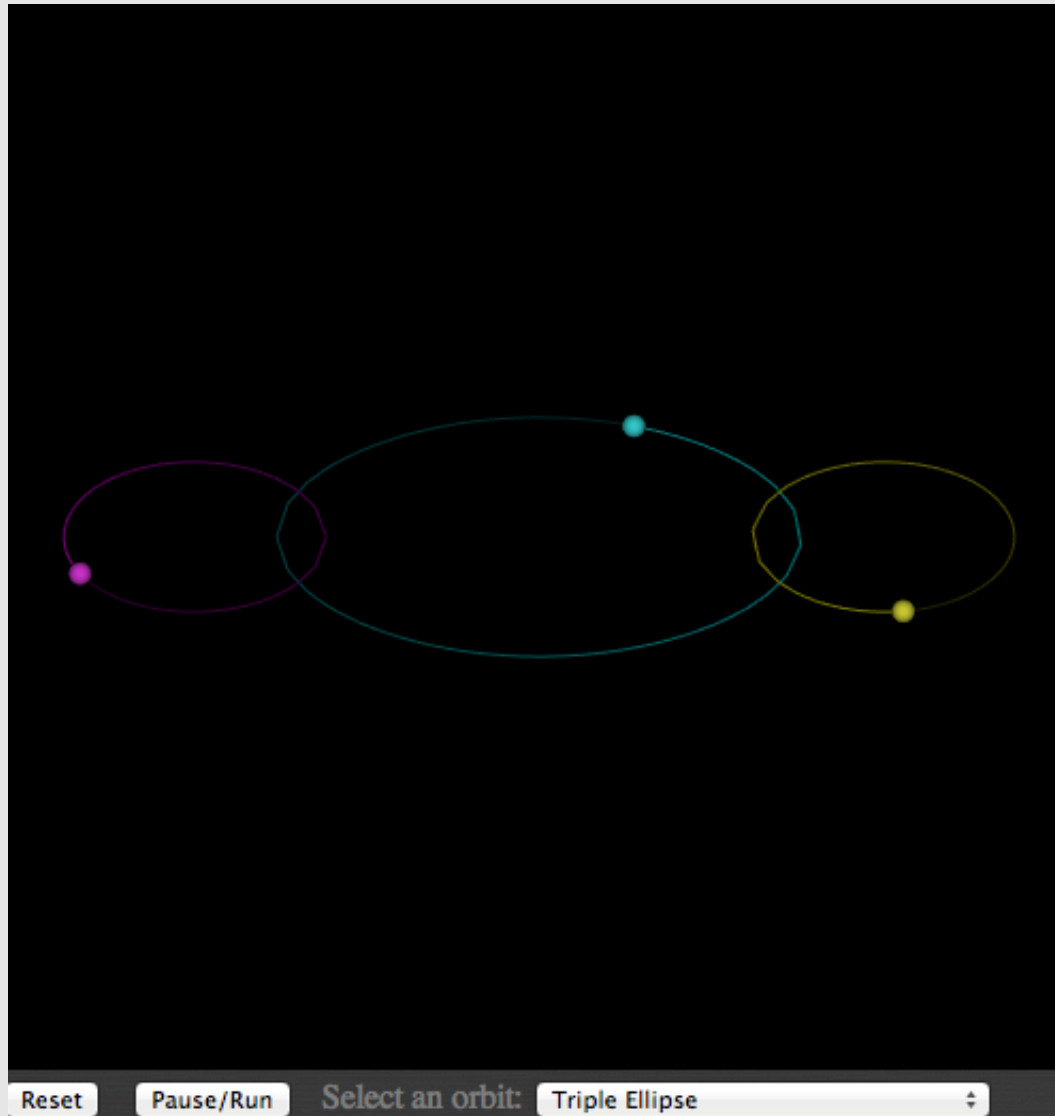


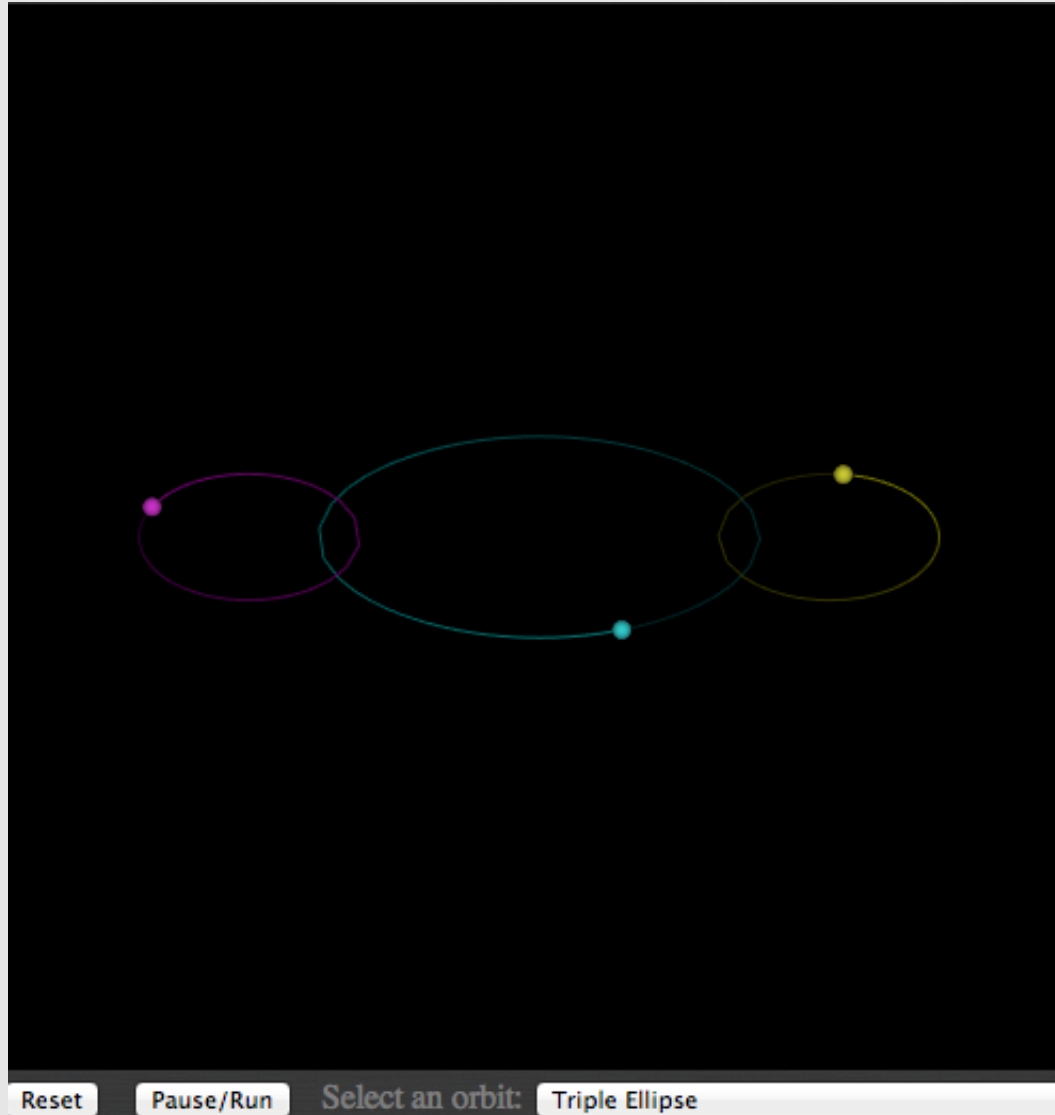


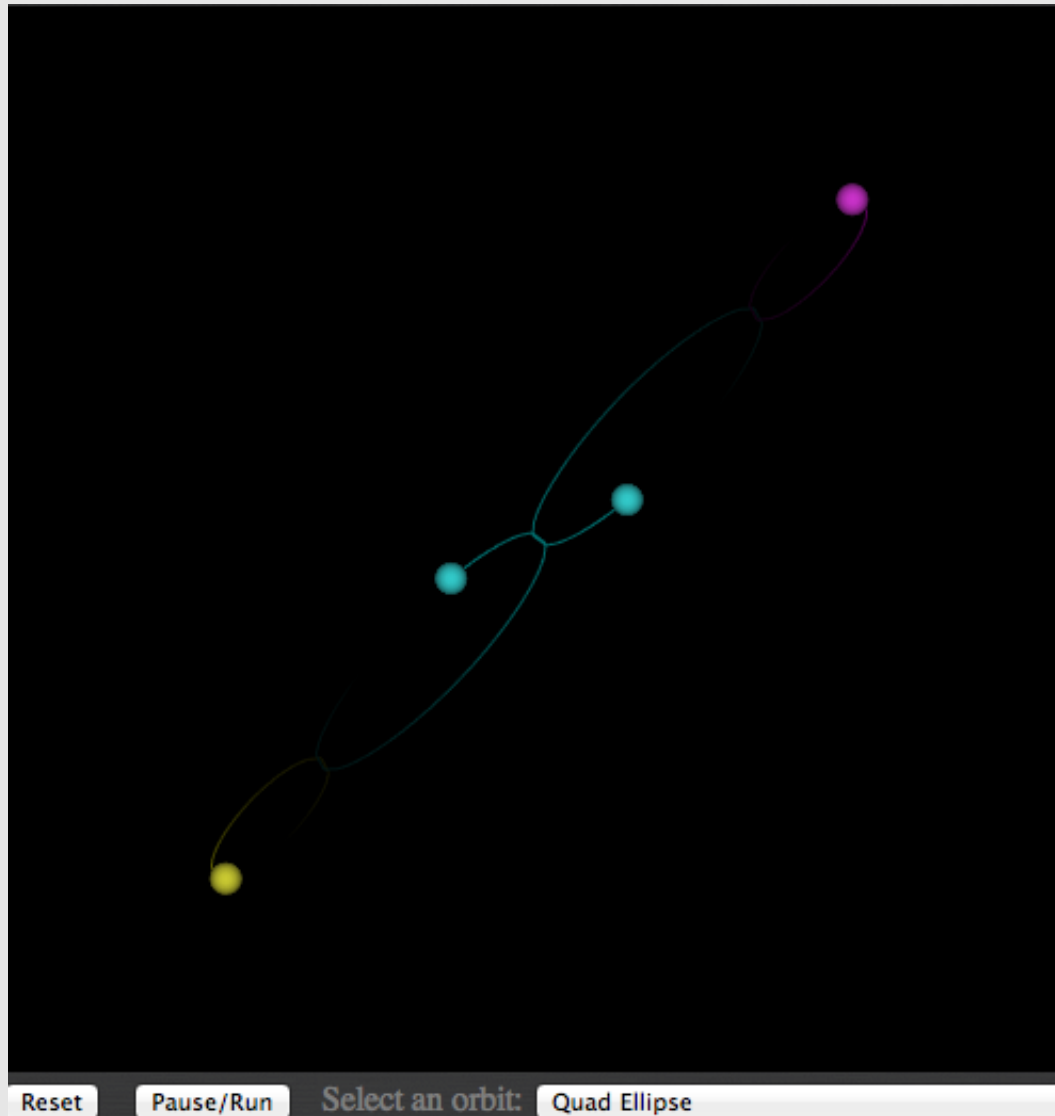
Double Ellipse



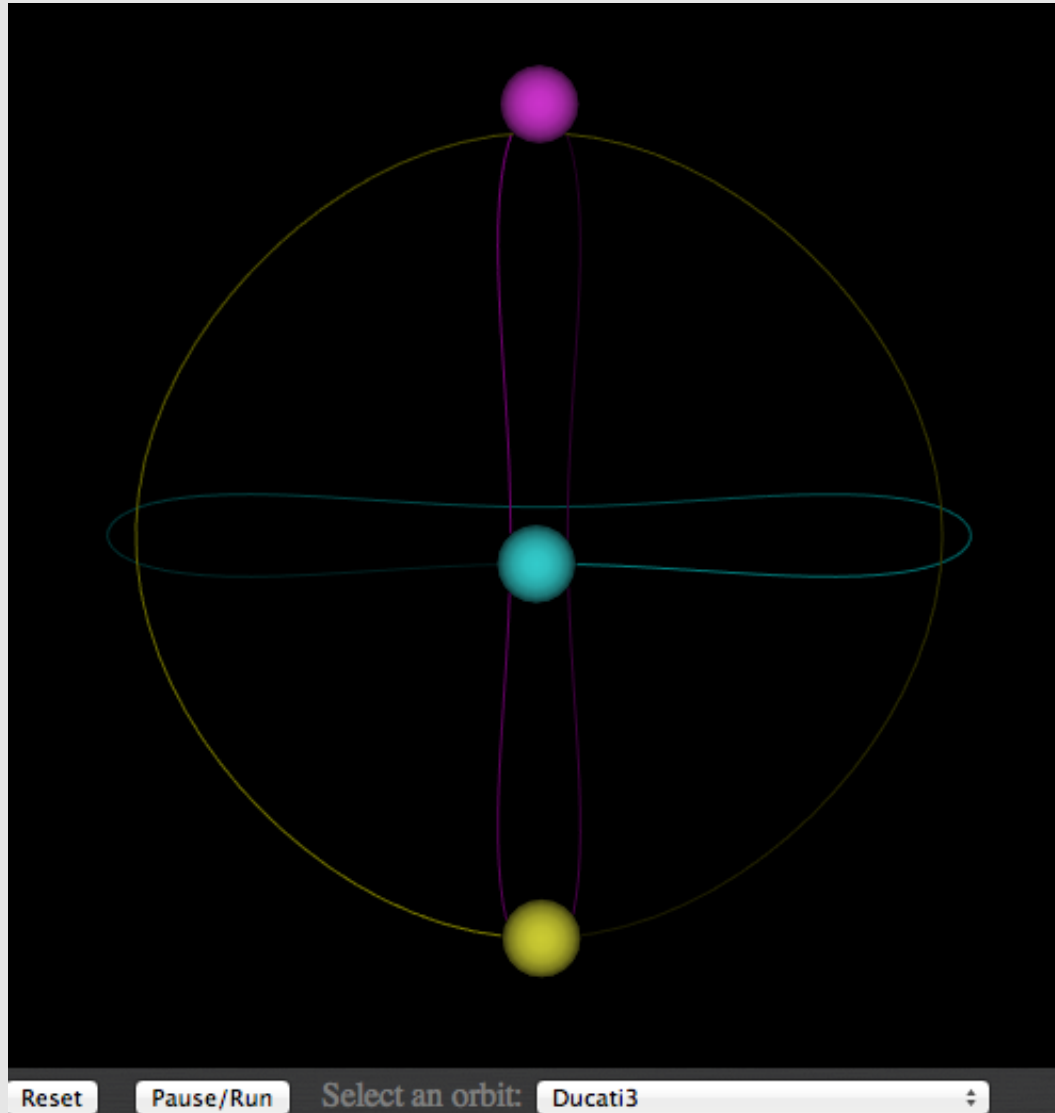
Triple Ellipse



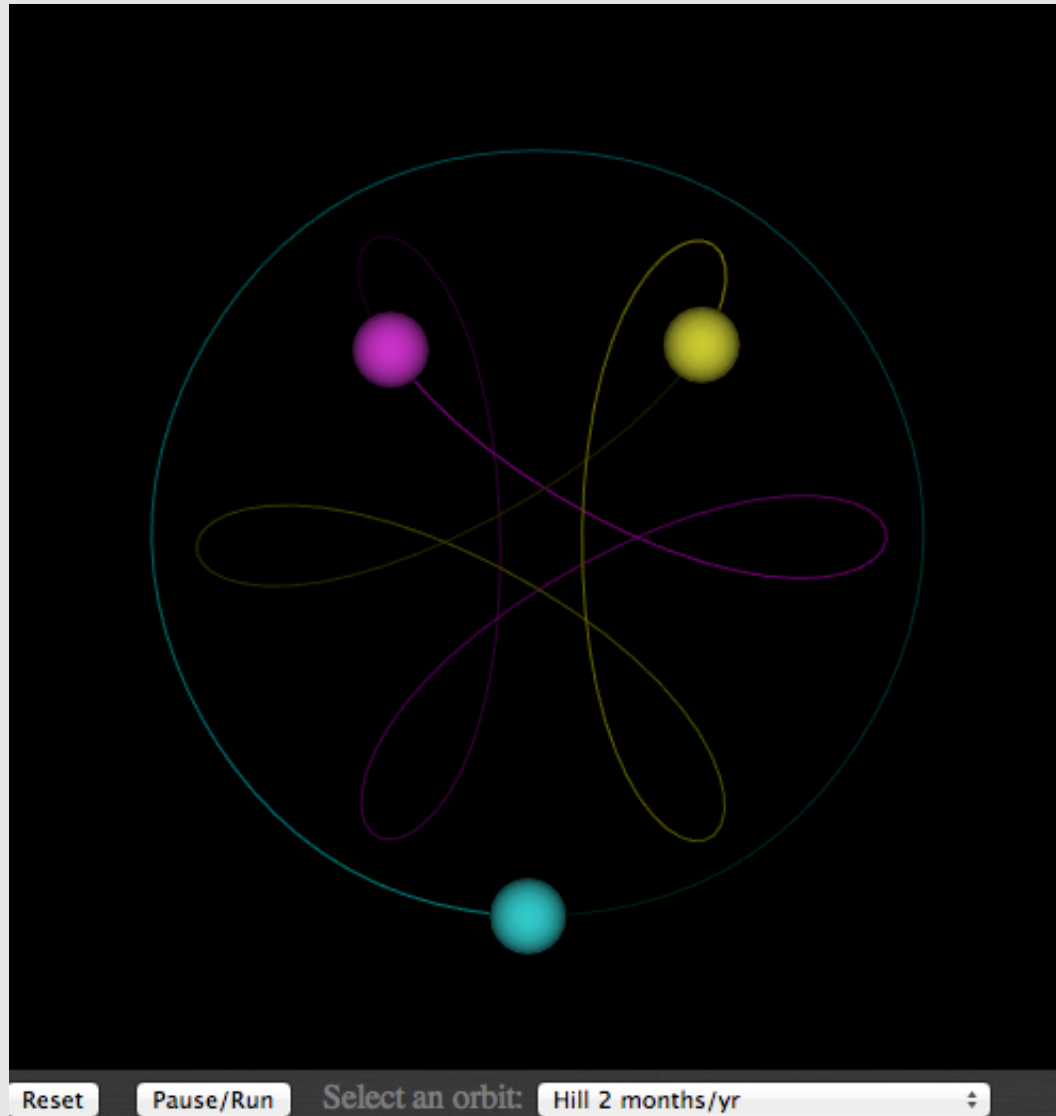




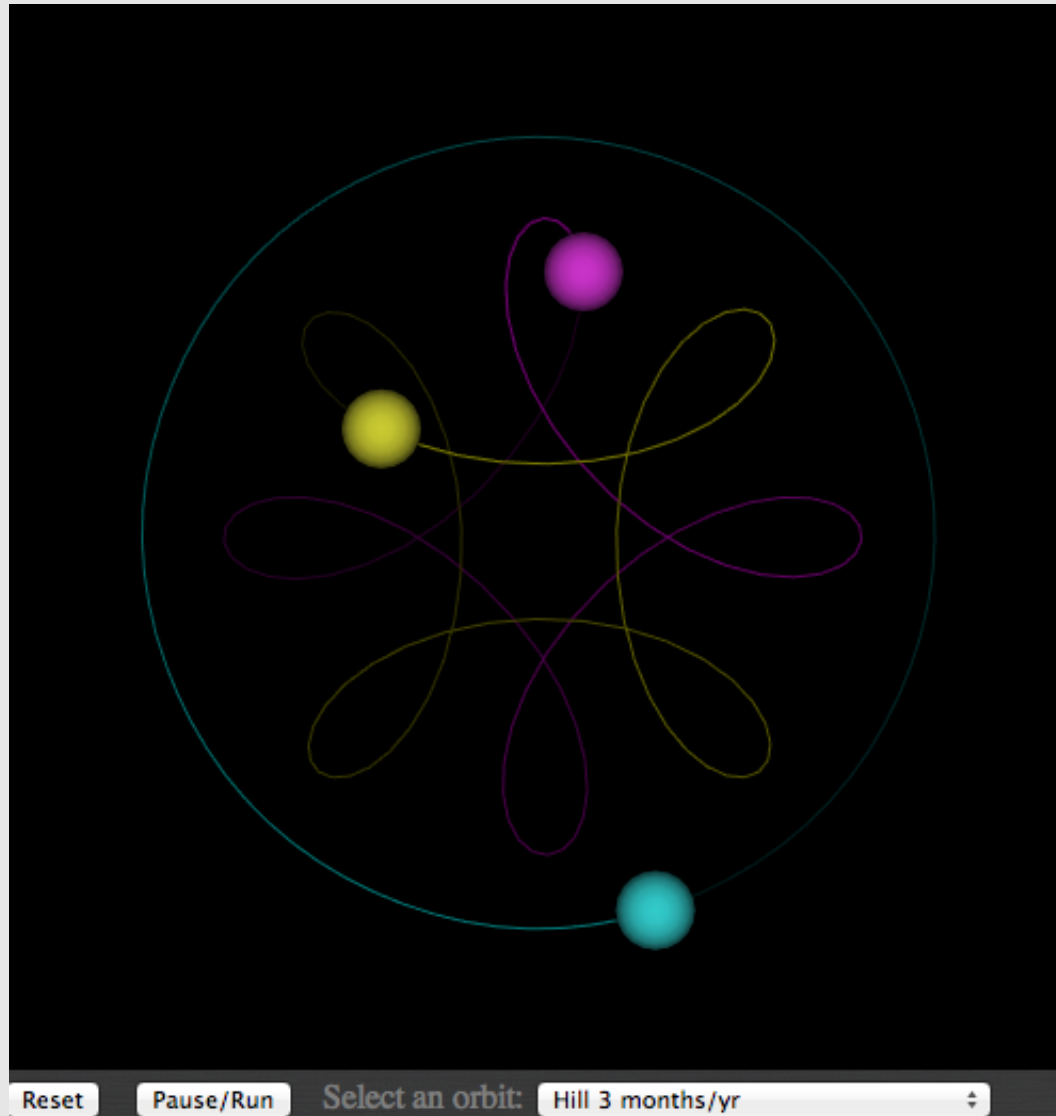
Ducati

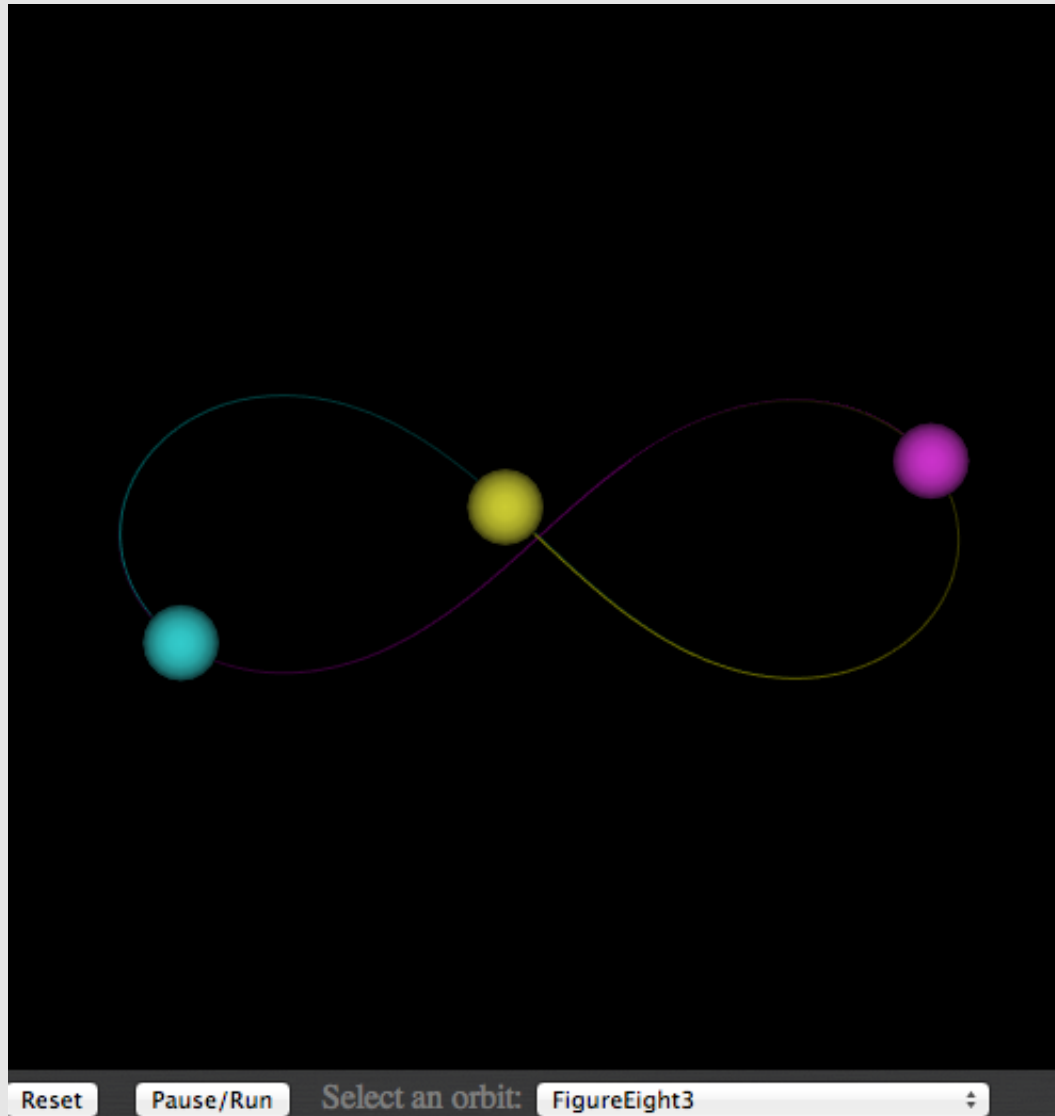


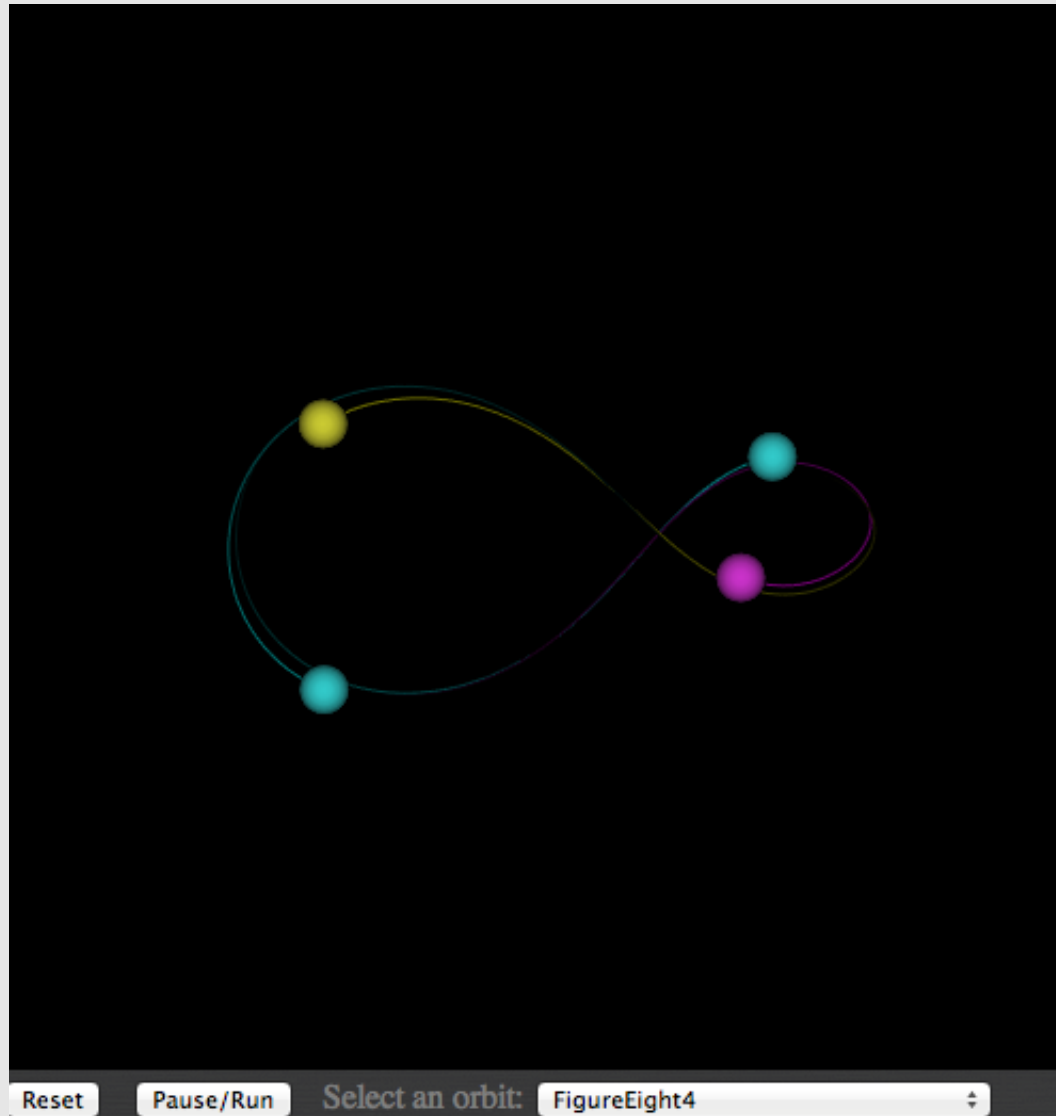
Hill – 2 Months/Year

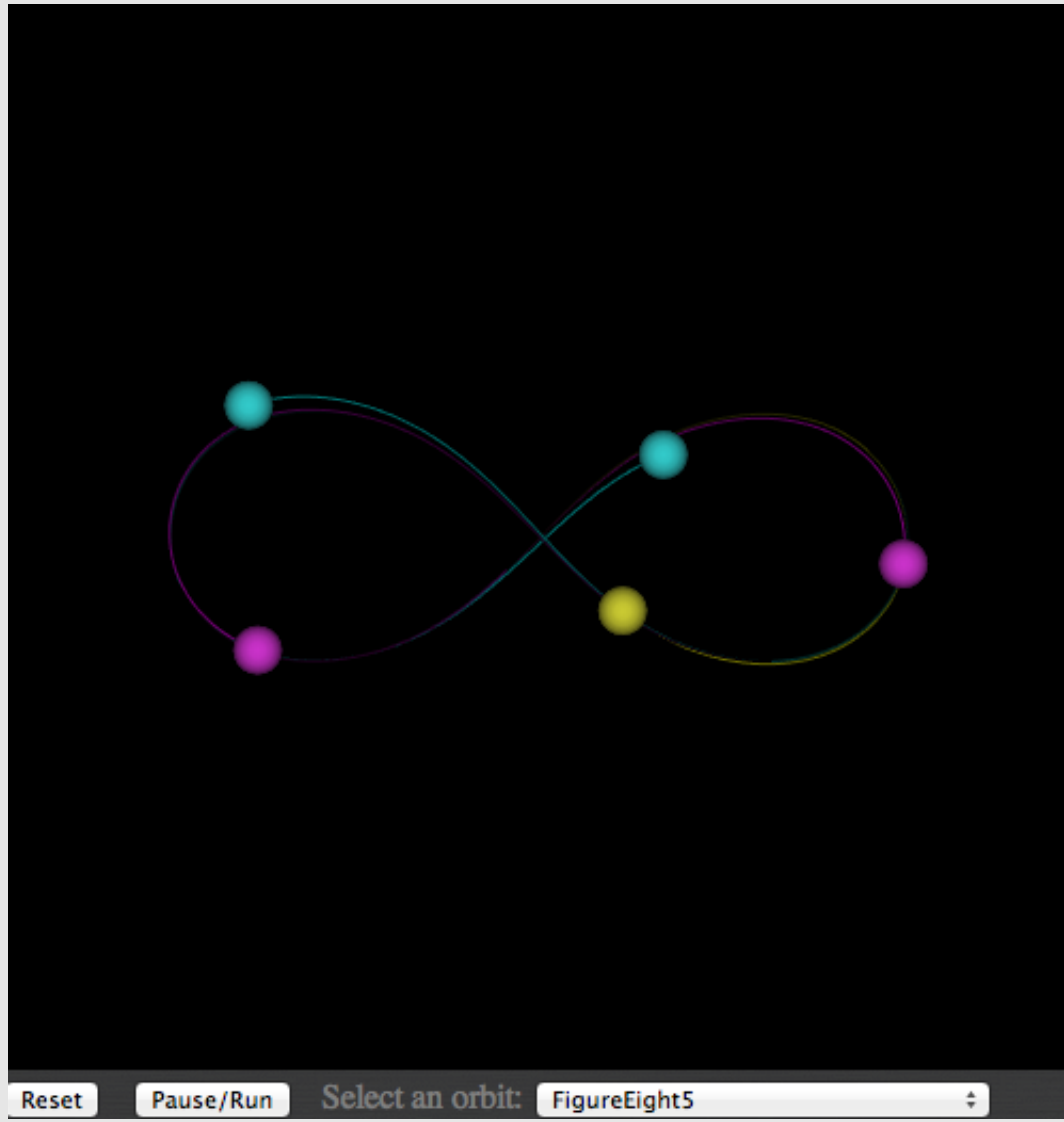


Hill – 3 Months/Year

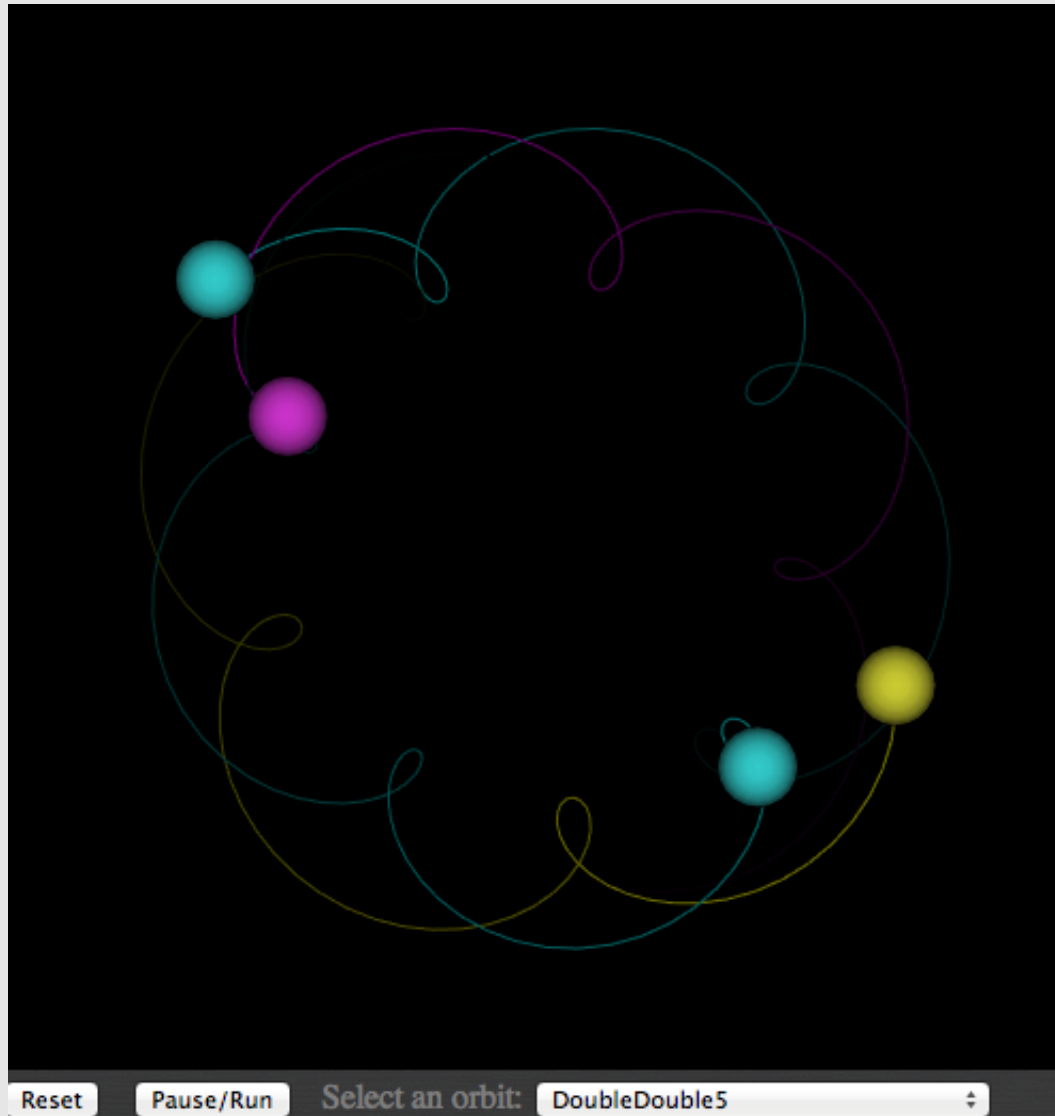




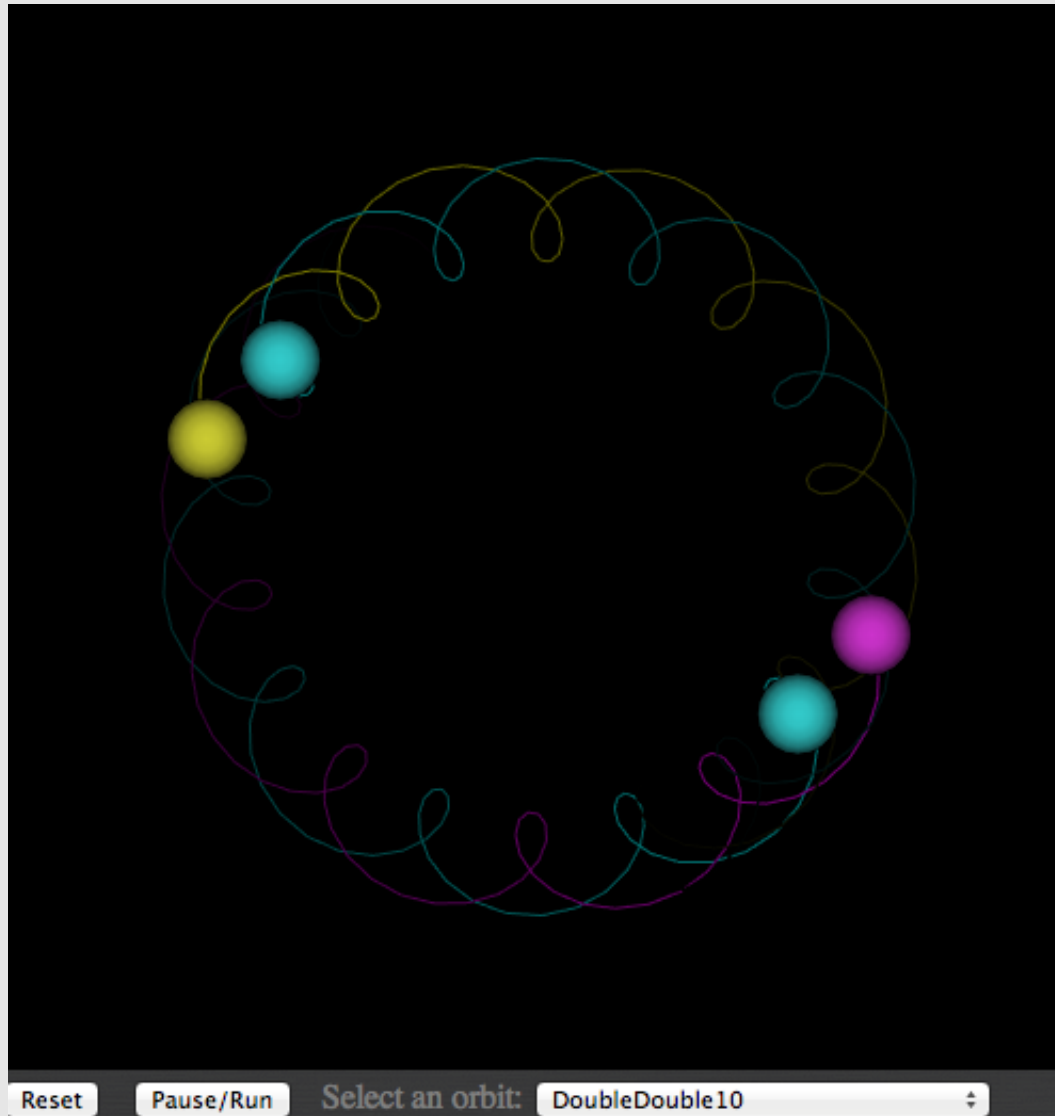




Double/Double – 5 Months/Year



Double/Double – 10 Months/Year



Double/Double – 20 Months/Year

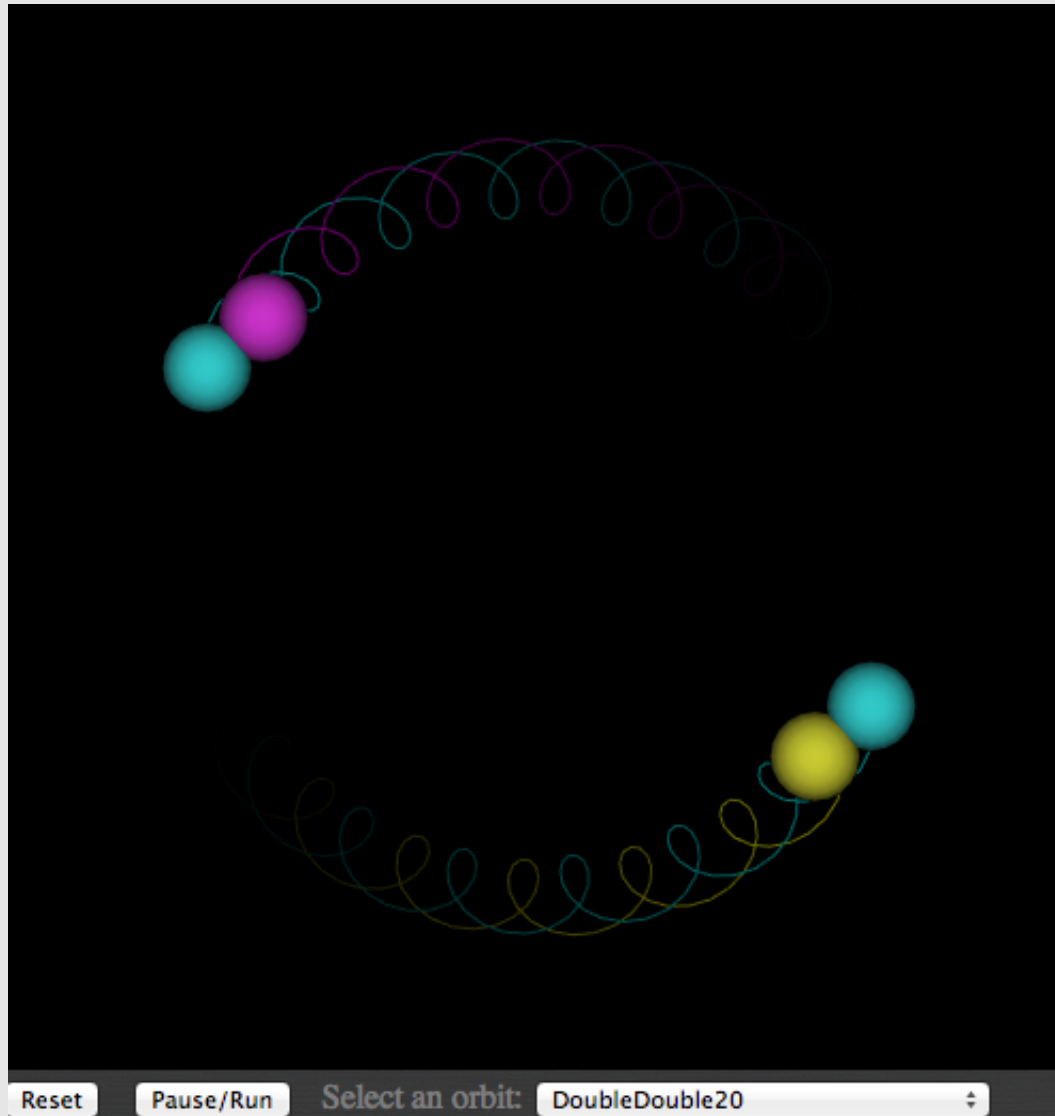
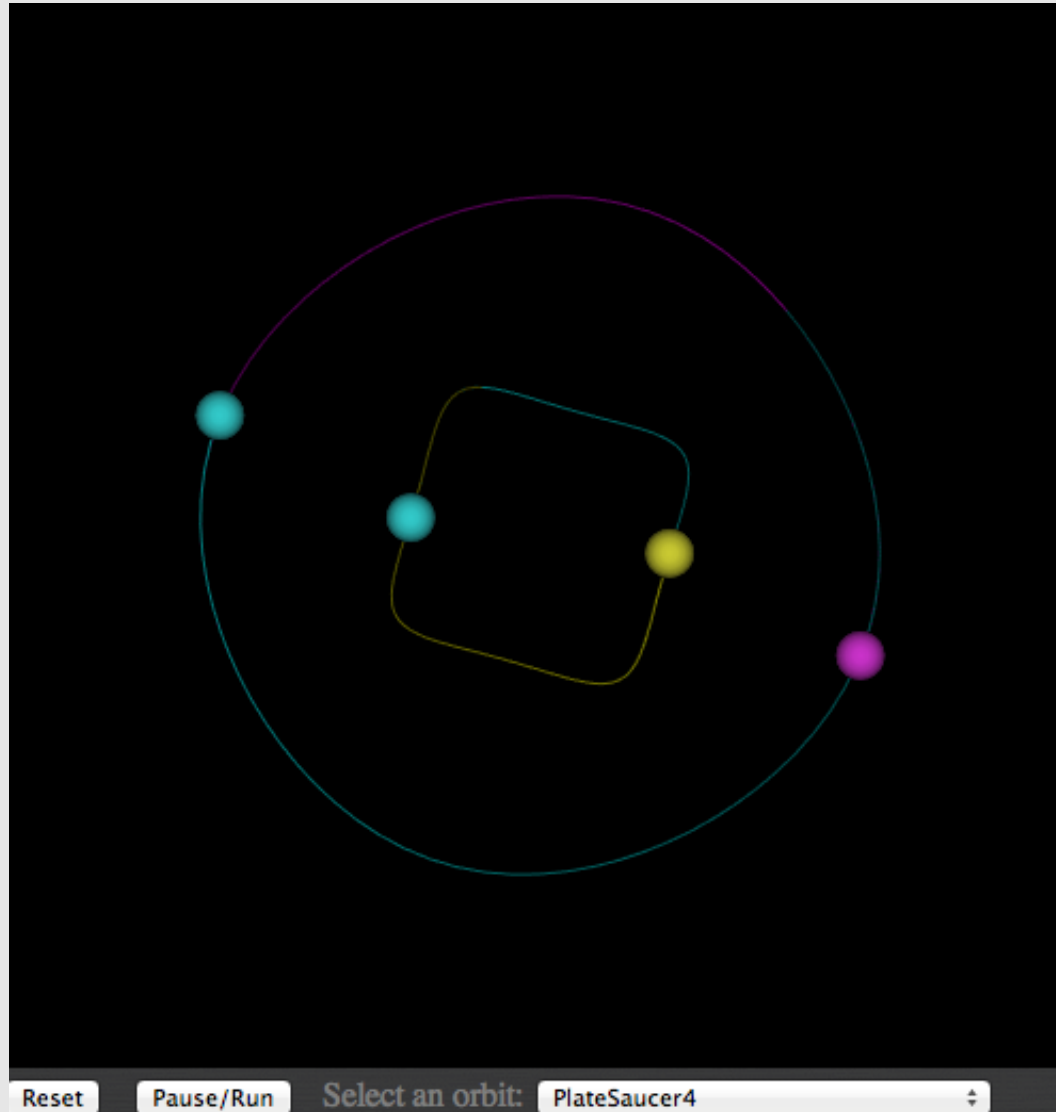
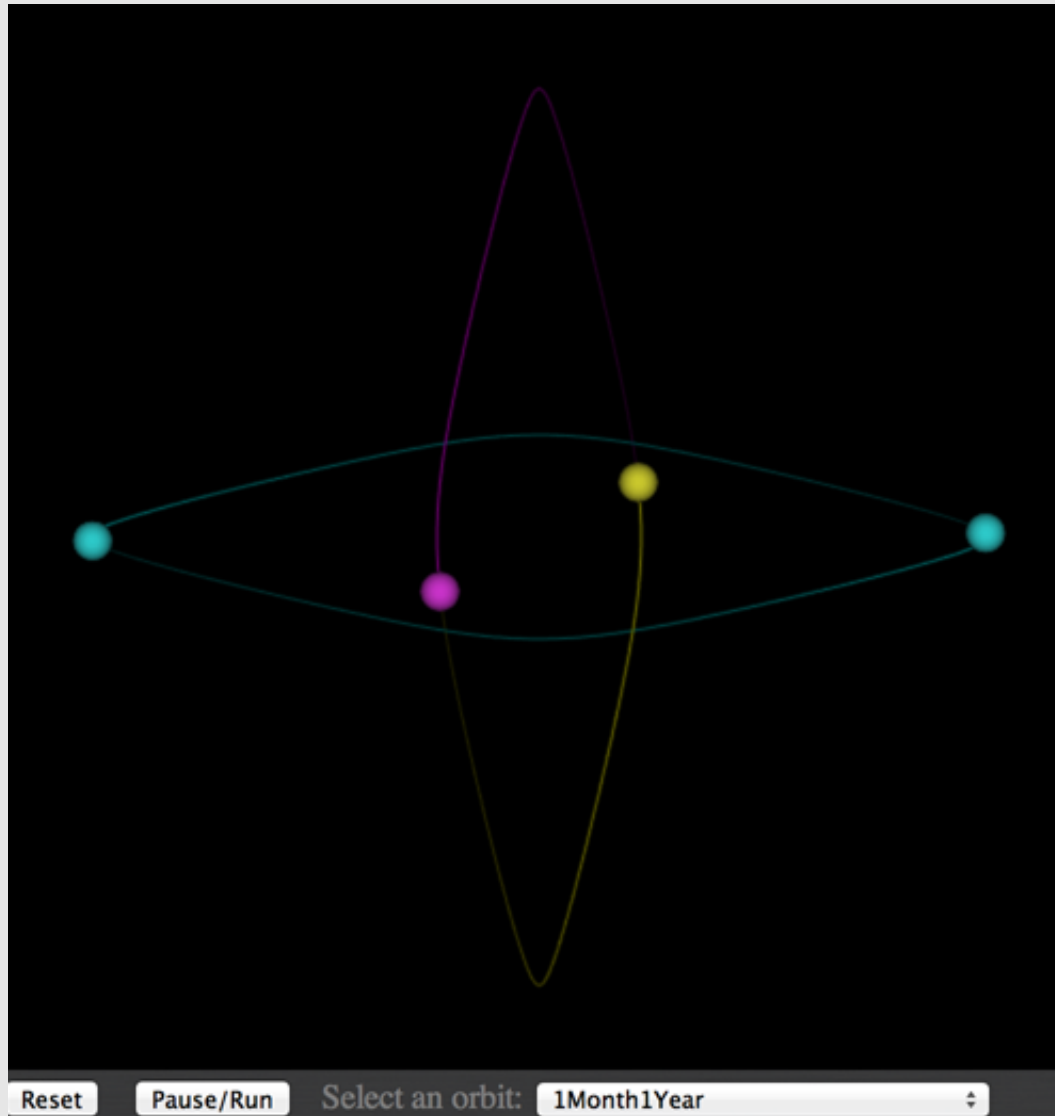
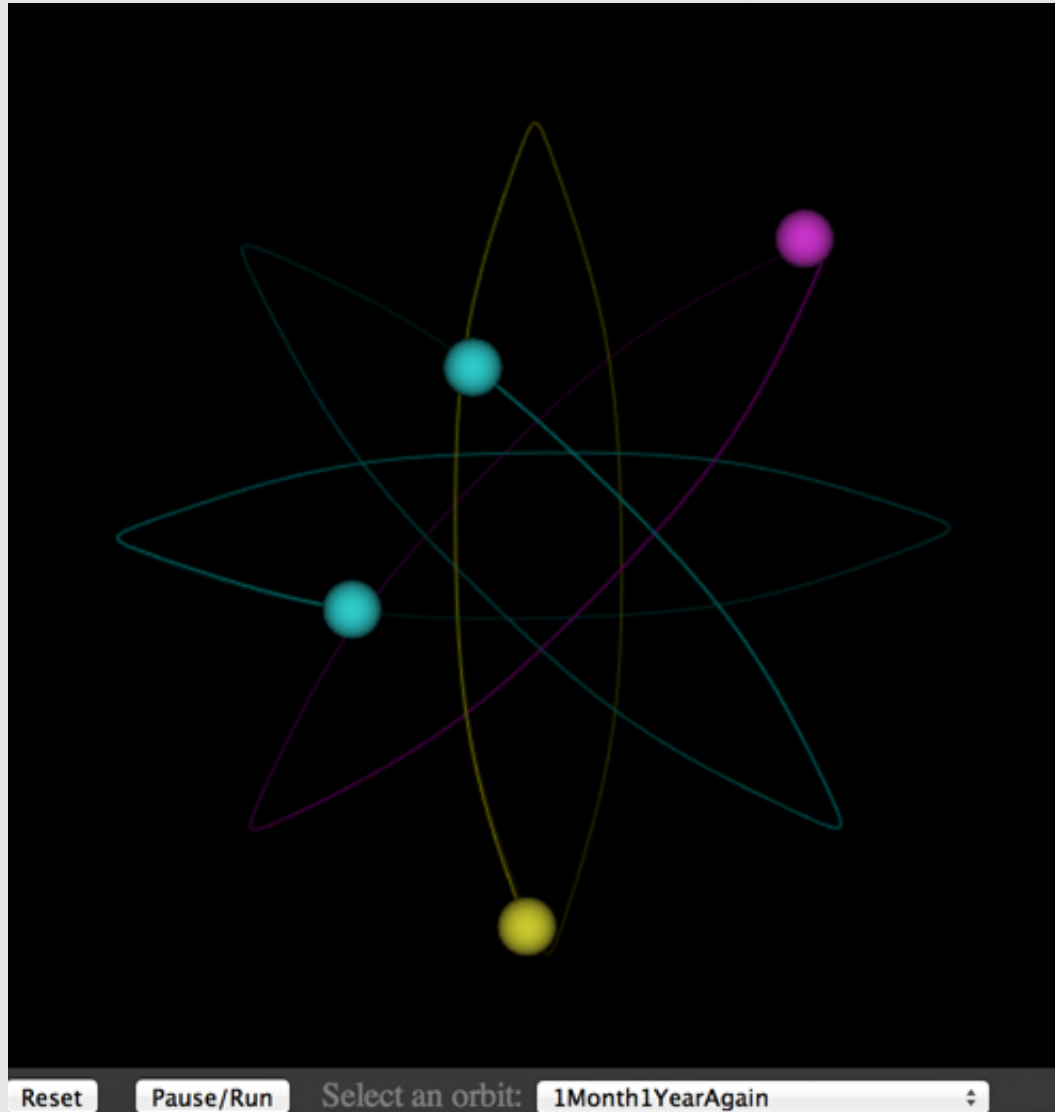


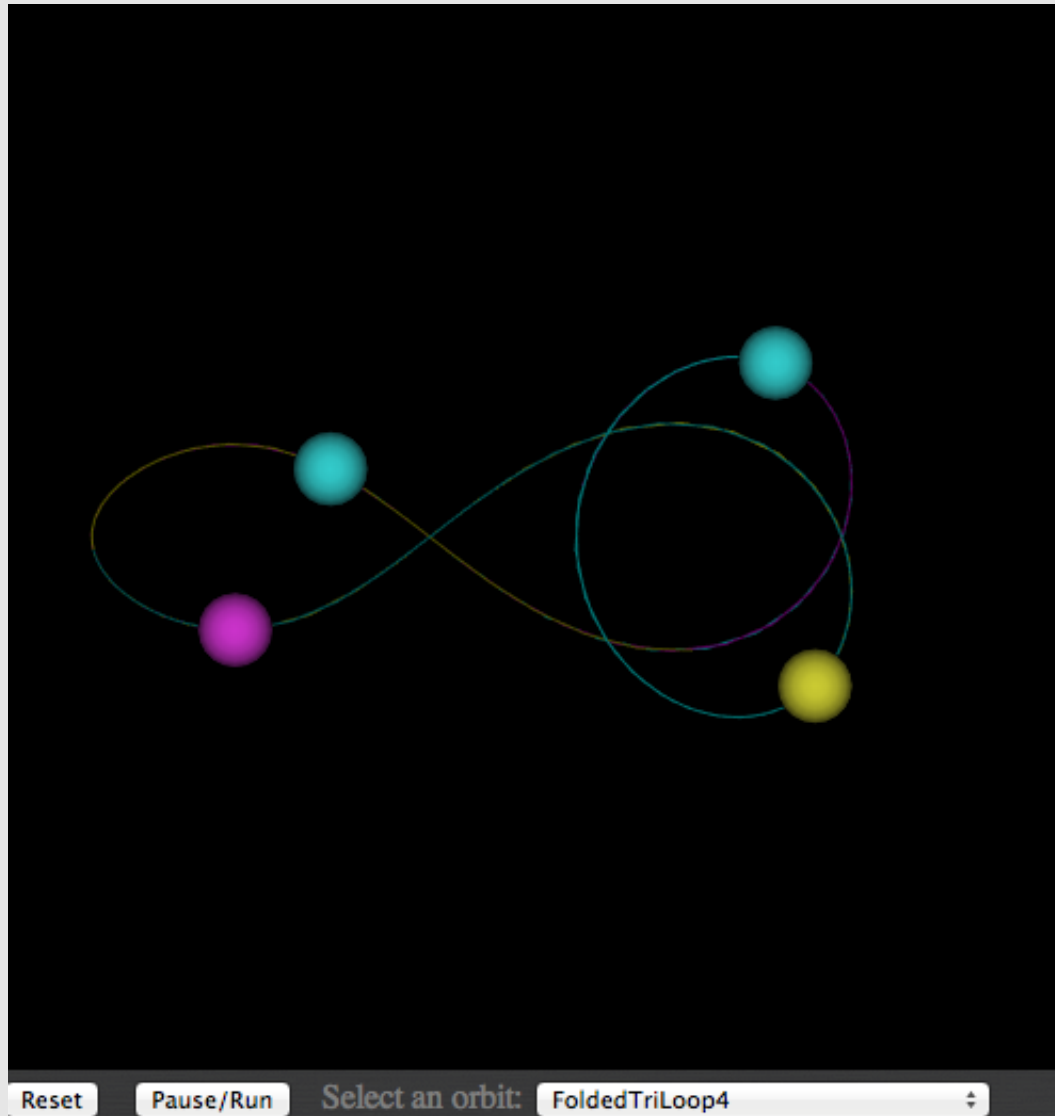
Plate and Saucer

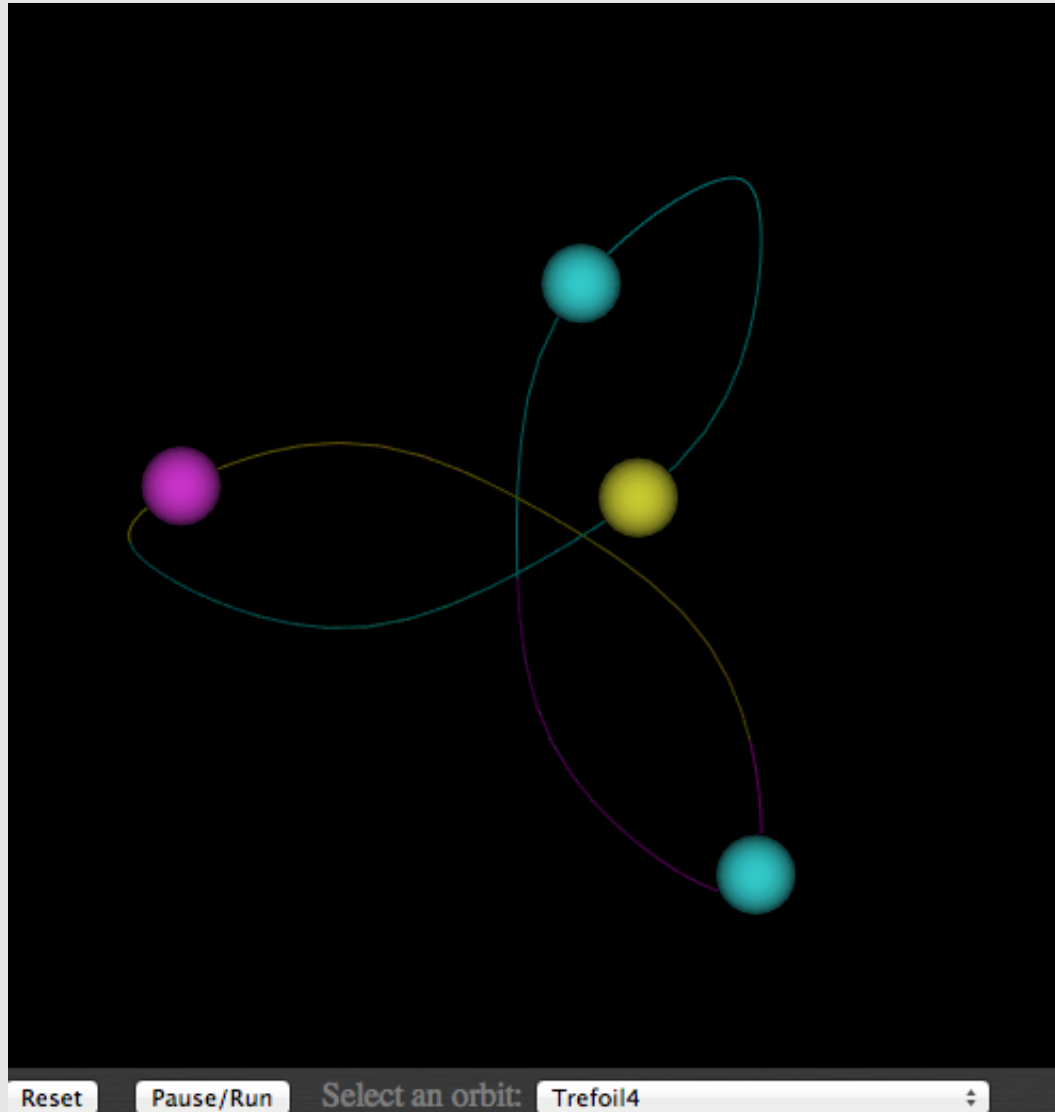


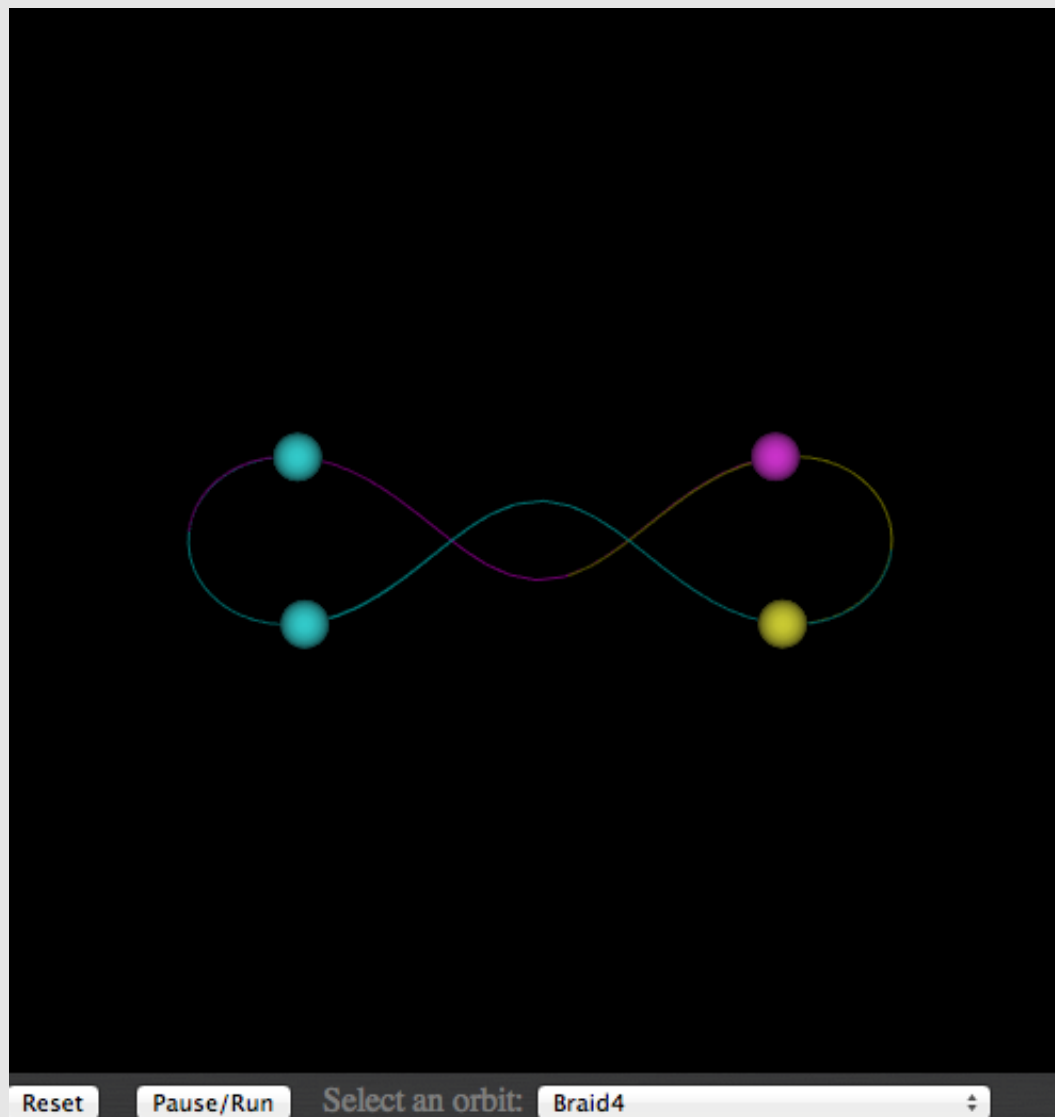
Ortho Quasi Ellipse

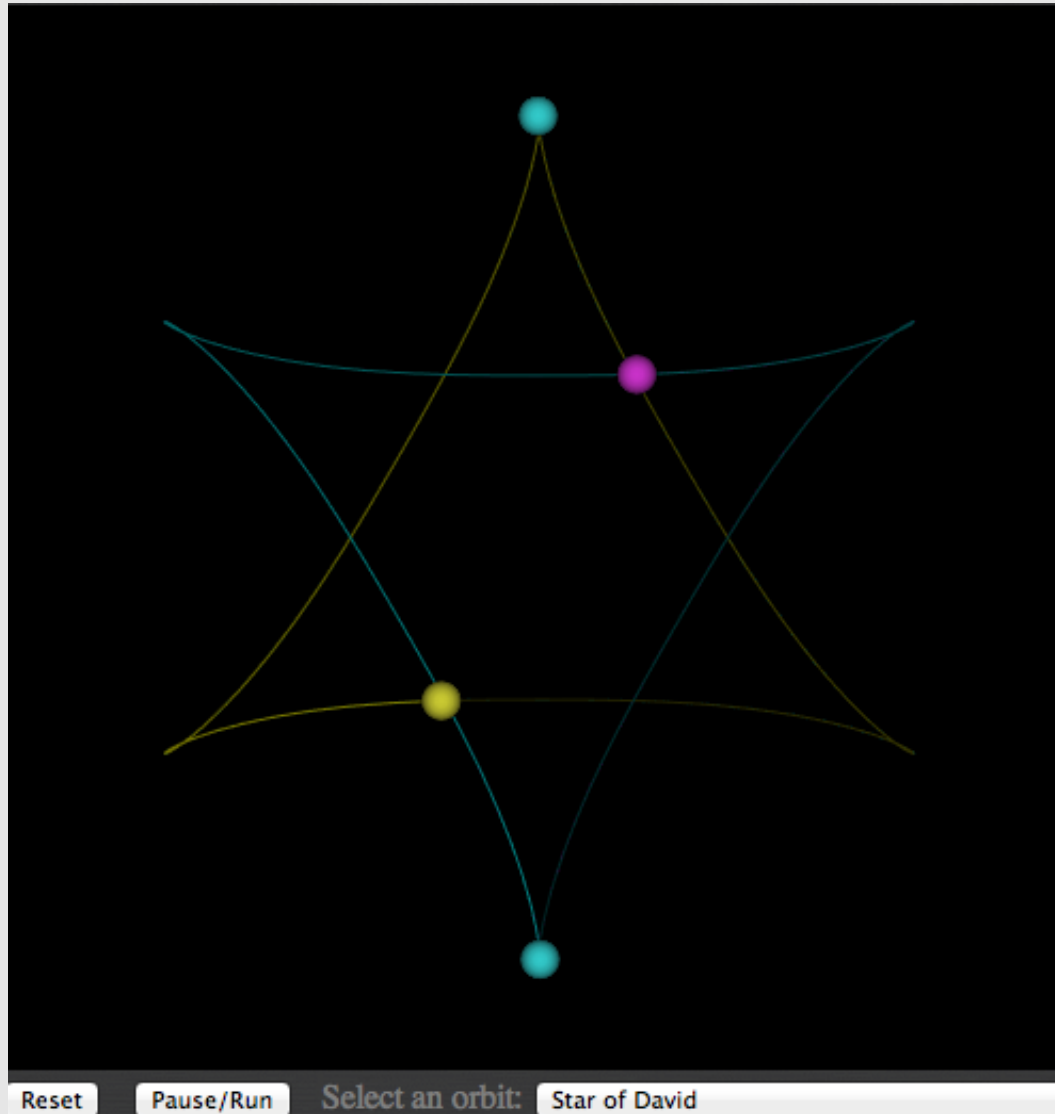


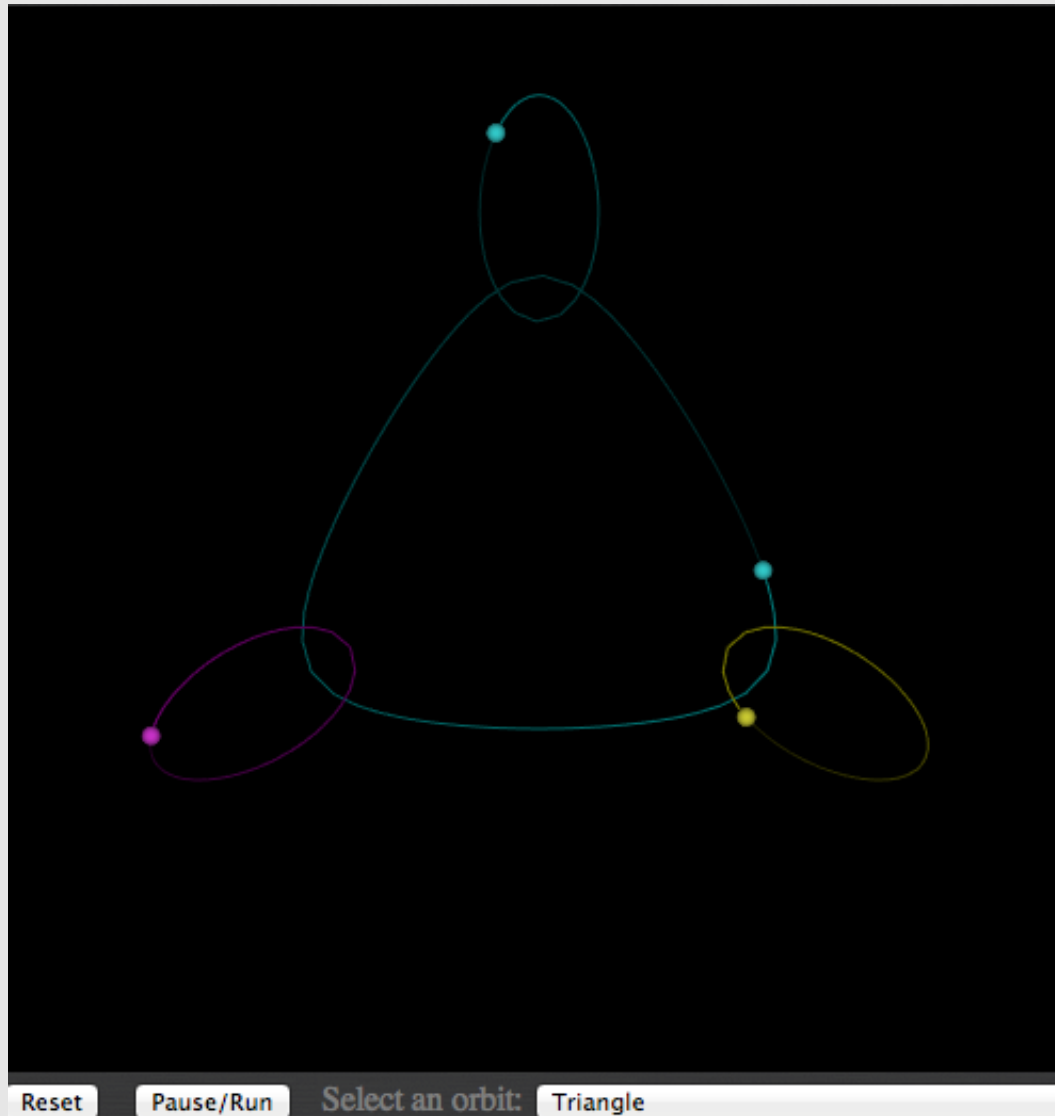


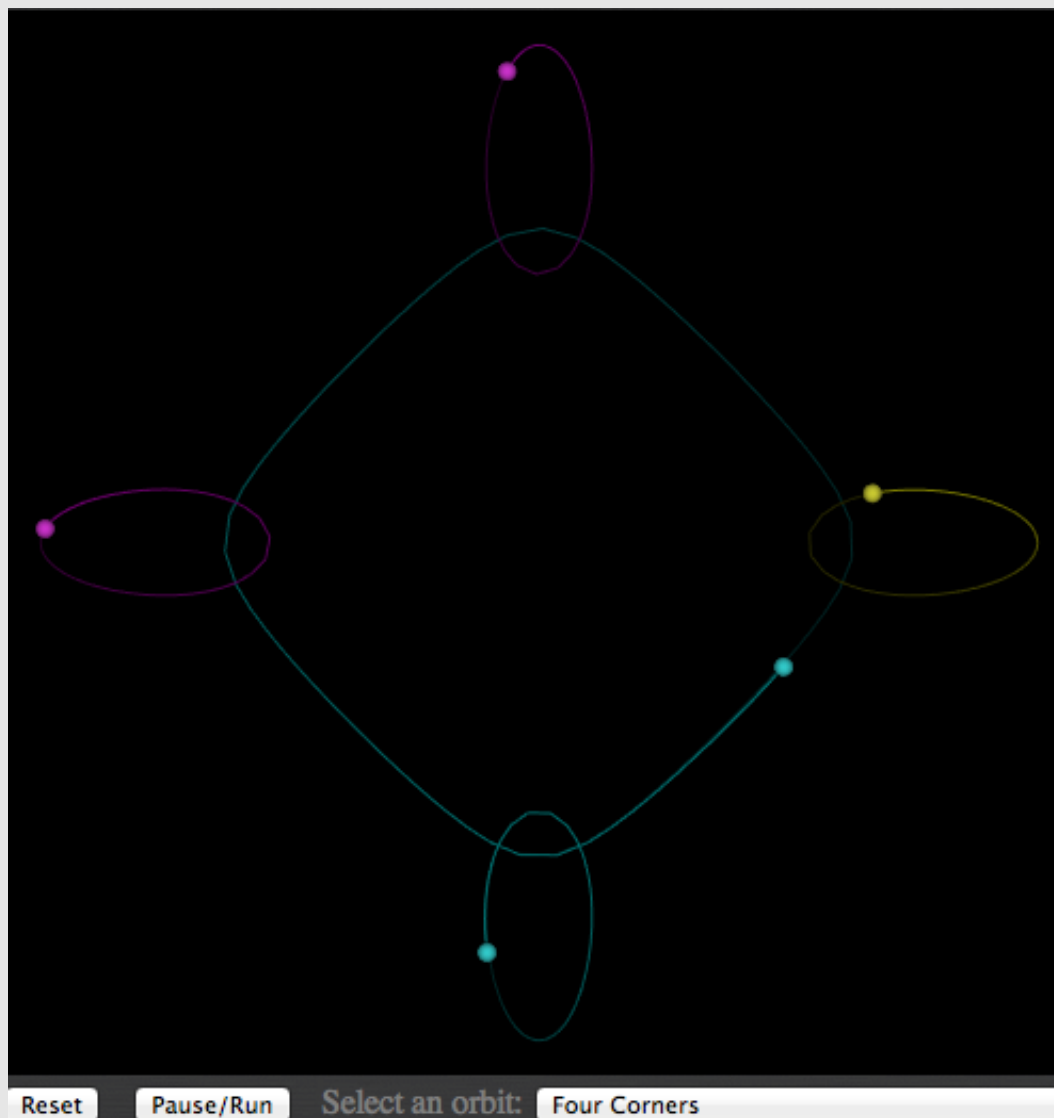


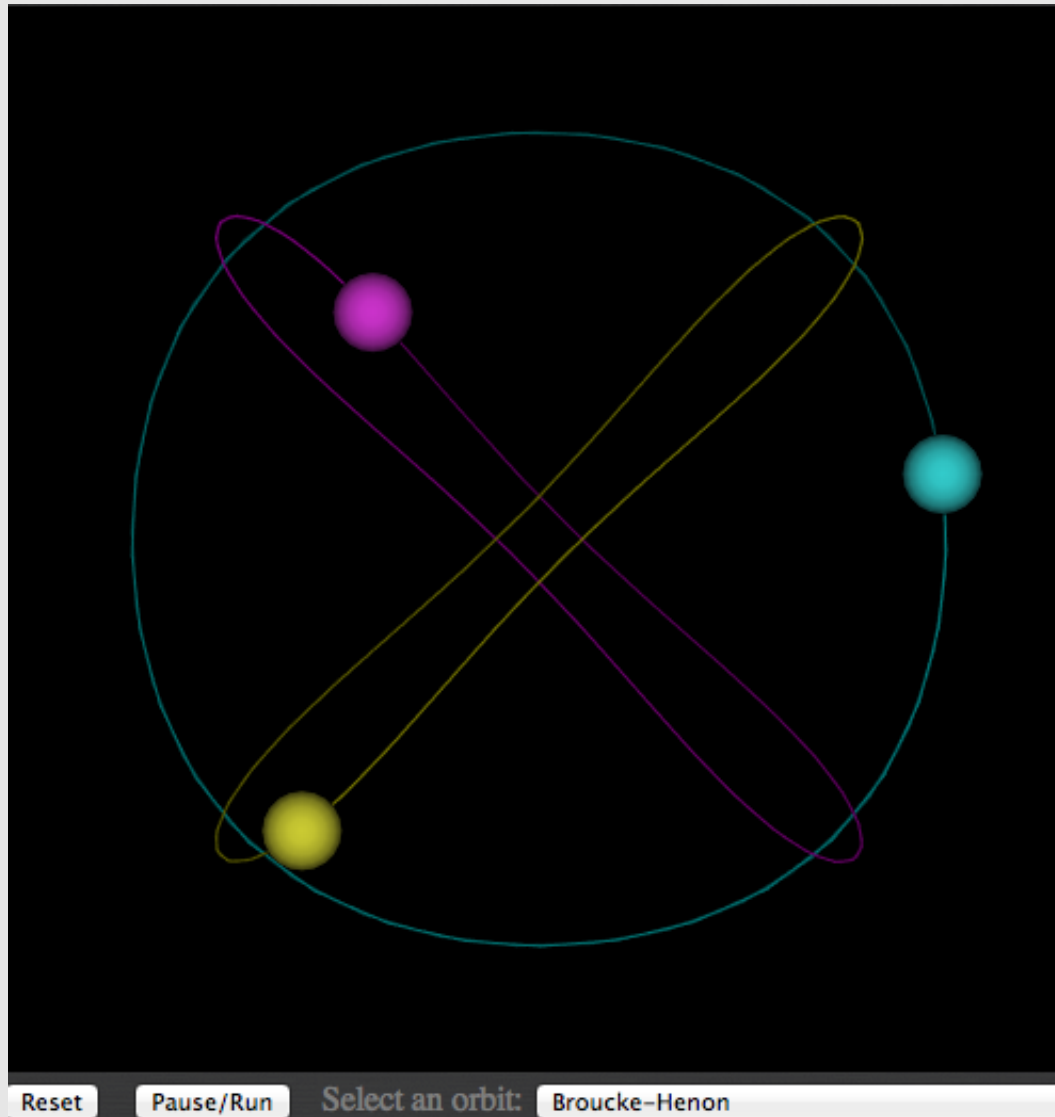


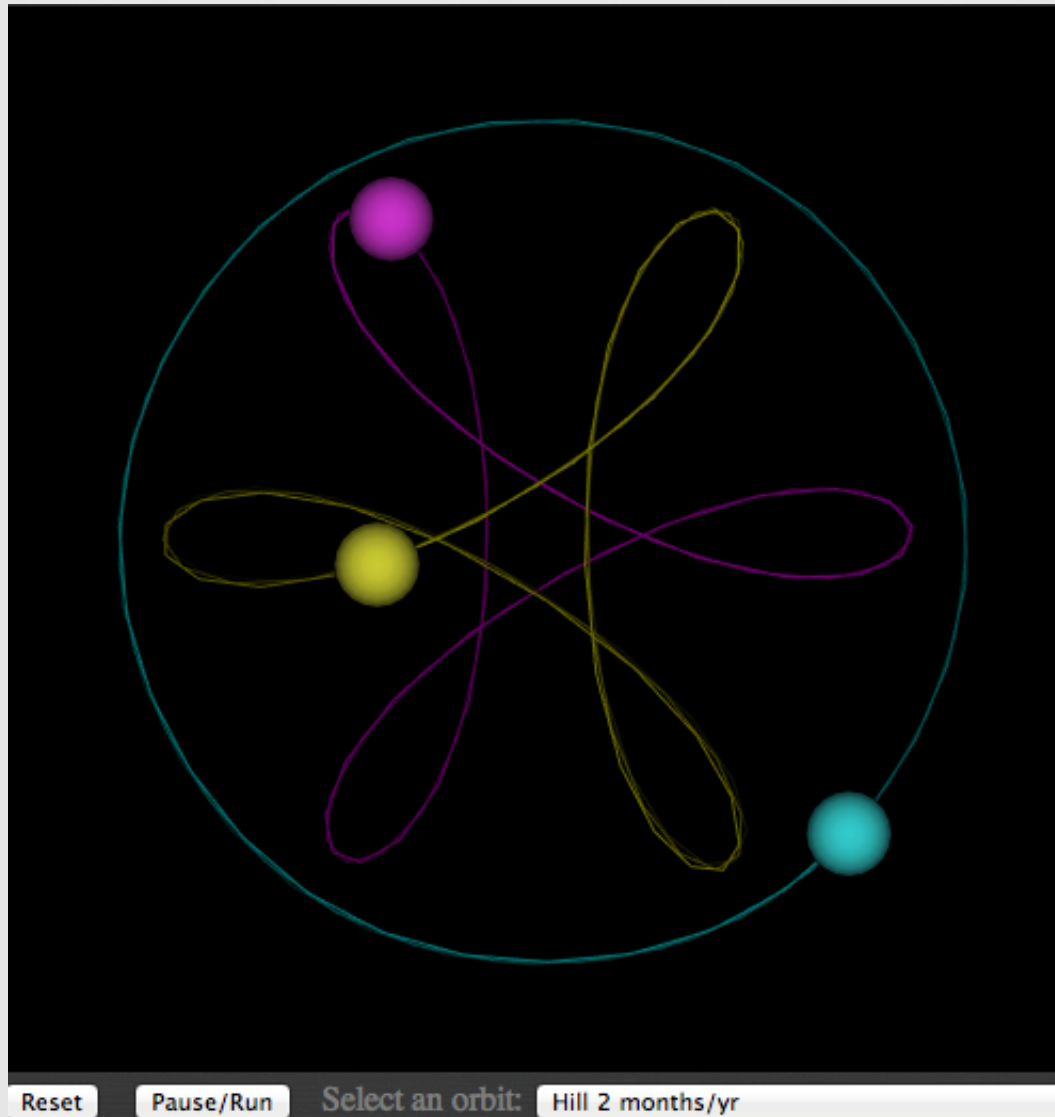


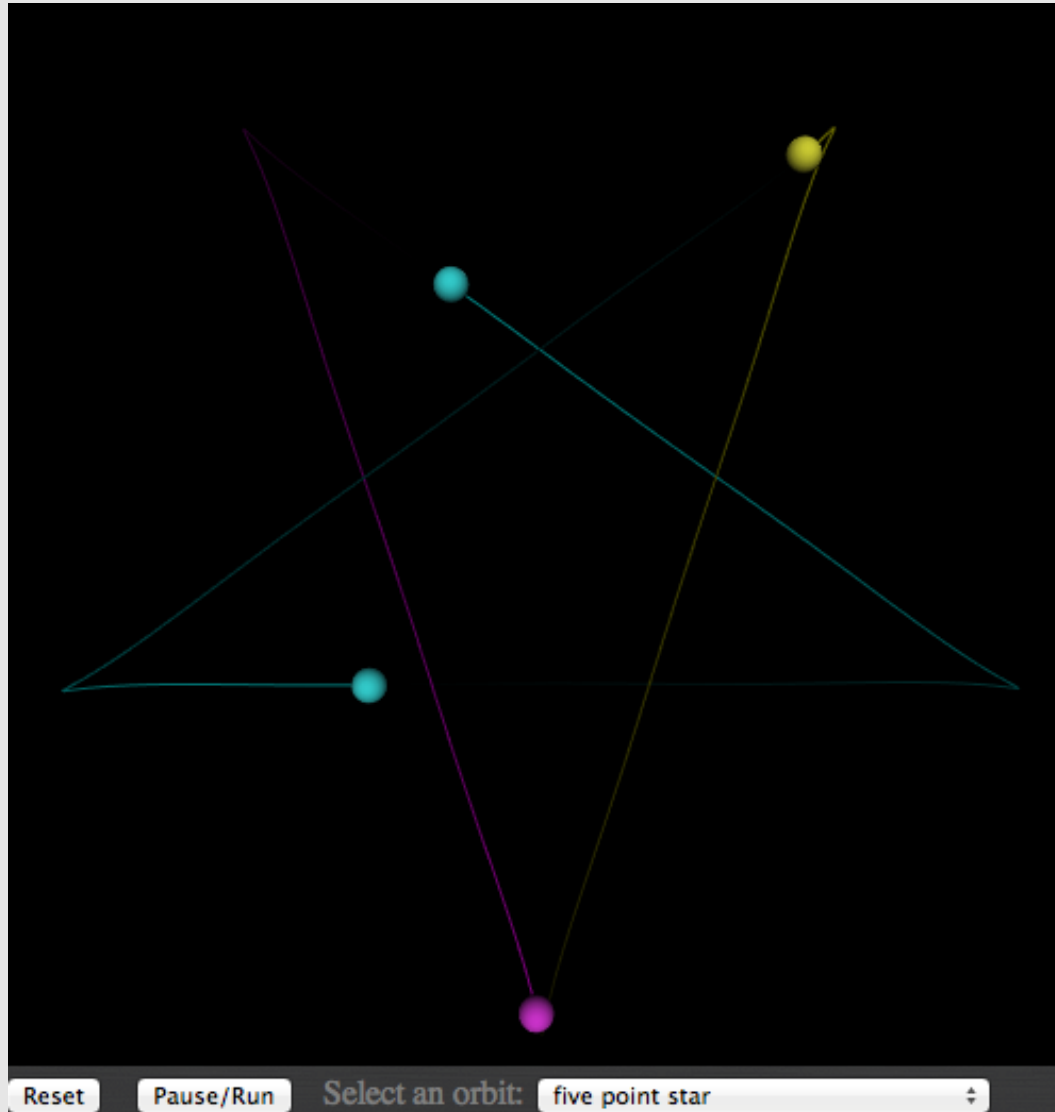




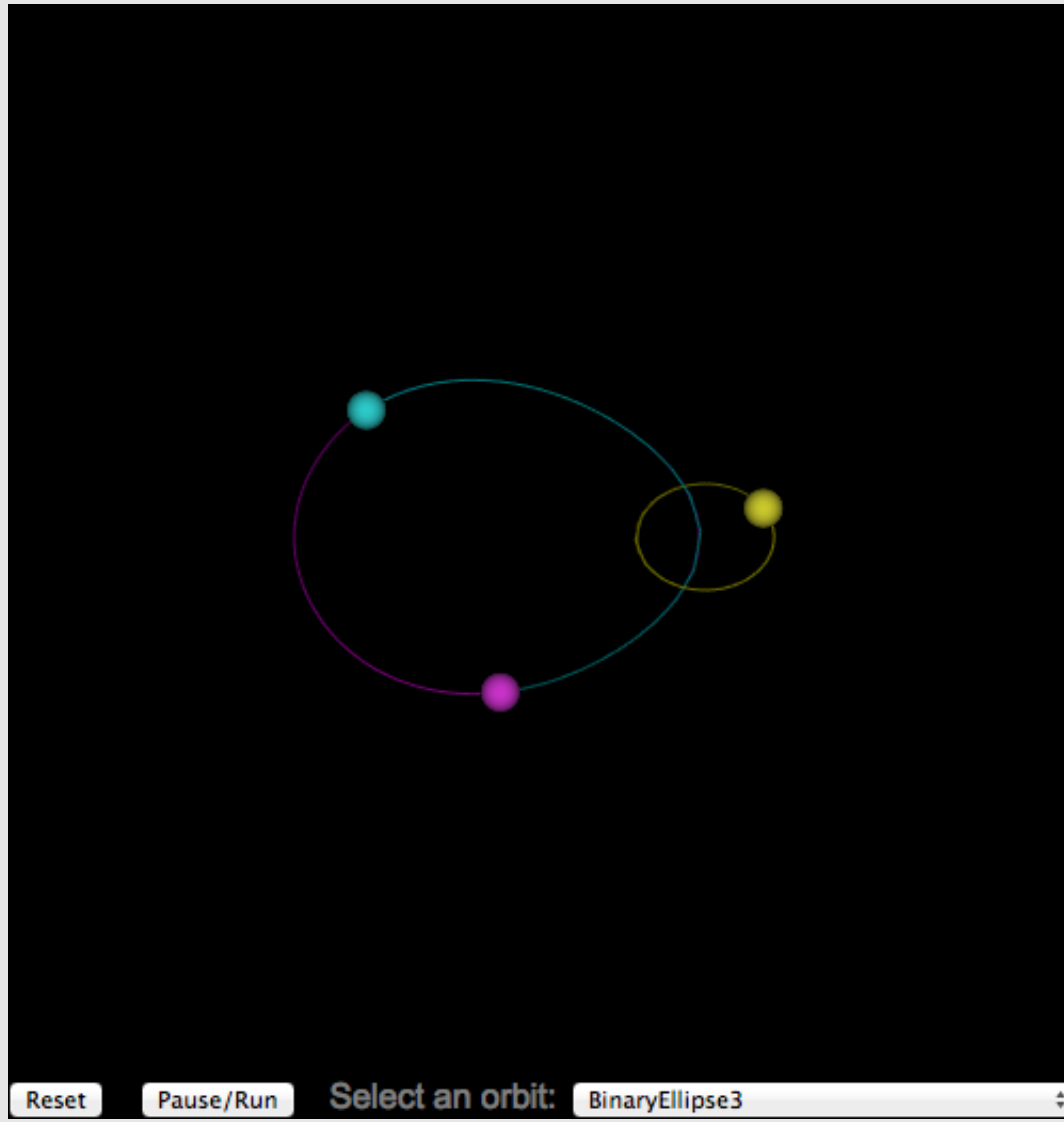








Binary Ellipse 3



Hexagon

