Numerical Methods in Astrophysics

Project 3

Two Dimensional Random Walk, Circular Binary and Hypervelocity Stars

Saksham Kaushal

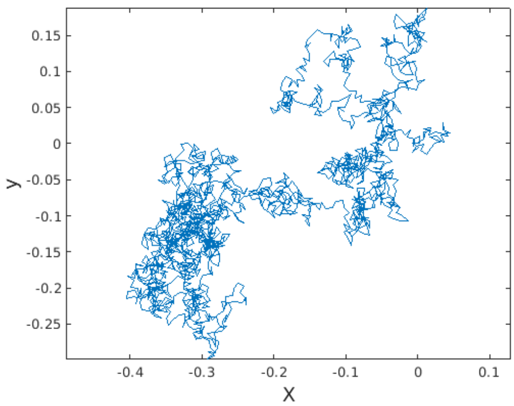
Contents

Part I
Two Dimensional Random Walk

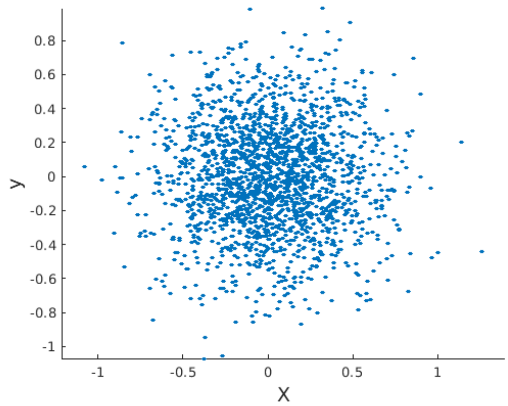
1 Introduction

A random walk is a process that describes the path of an object in a two dimensional plane consisting of successive random steps. In physics, these processes play an important role in study of polymers, Brownian motion, diffusion, etc.

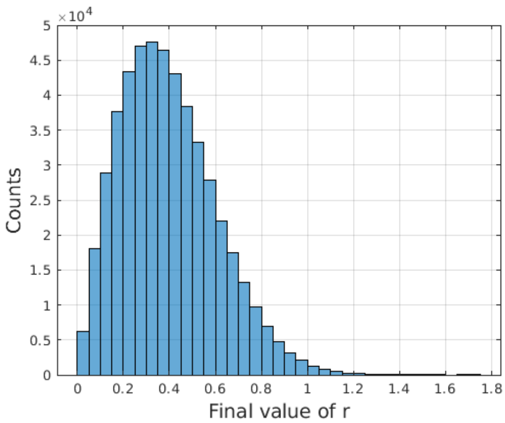
2 Methods, Results and Discussions



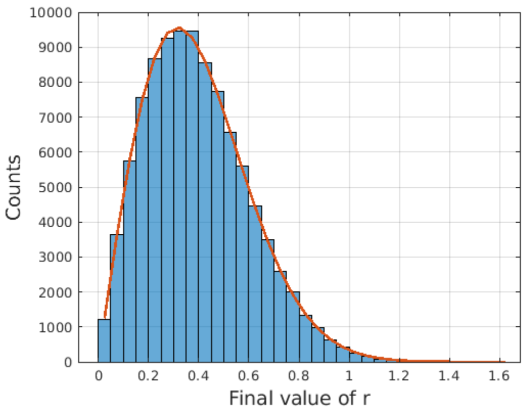
(a) Two dimensional random walk of a particle over 2000 time steps



(b) Distribution of final coordinates of 2000 random walk particles



(c) Number distribution with distance of 5×10^5 random walk particles



(d) Theoretical estimates and observed distribution of 10^5 random walk points. The blue histogram represents the observed number distribution , while the red line is a plot of analytical estimates.

Figure 1: Plotted results for problem 1.1A.

(a) **Methods** – Random walk of a particle which is initially located at origin (0,0), is computed for 2000 time steps. At each time step, the object moves a distance, $d = 0.01$ units in a random direction which is mathematically represented using angle, $\theta \in [0, 2\pi]$, computed in the Matlab program with the help of inbuilt function, `rand()`. This function generates random number uniformly in the range $[0, 1]$, therefore, multiplying the obtained random number by 2π give us the random angle θ, in line 6 of the code 1 Successive movements in the two dimensional plane are computed in lines 7 and 8 of code 1, which are enclosed in a *for loop* defined in line 5, which repeats the process for each of the 2000 time steps.

```
clear;
x(1)     = 0;                          % initial x at origin
y(1)     = 0;                          % initial y at origin
d        = 0.01;
for i = 1:2000                         % for loop with 2000 steps
    theta   = 2*pi*rand();             % random theta between zero and 2pi
    x(i+1)  = x(i)+d*cos(theta);       % next value of x
    y(i+1)  = y(i)+d*sin(theta);       % next value of y
end
plot(x,y);                             % plot the random walk
axis equal;                            % equal dimensions for axes
xlabel('X','FontSize',14);             % x-axis label
ylabel('y','FontSize',14);             % y-axis label
```

Code 1: prob1aa.m - 2D random walk of a particle

Results – The plot of two dimensional random walk of a particle obtained using Matlab code given in code 1 is shown in figure 1a.

(b) **Methods** – Random walk for 2000 time steps, like the one performed in previous part is performed for 2000 different particles. Accordingly, the crux of code 1 is executed in a *for loop*, 2000 times, i.e. once for each particle, as shown in line 4 of code 2. Instead of visualizing the complete path of random walk, this time only the final coordinates of the particle after finishing the random walk are considered and stored separately in lines 12 and 13 of code 2.

```
1  clear;
2  d   = 0.01;
3
4  for i = 1:2000                         % for loop with 2000 steps
5      x = 0;                             % initial x at origin
6      y = 0;                             % initial y at origin
7      for j = 1:2000
8          theta   = 2*pi*rand();         % random theta between zero and 2pi
9          x       = x+d*cos(theta);      % next value of x
10         y       = y+d*sin(theta);      % next value of y
11     end
12     xfinal(i)   = x;
13     yfinal(i)   = y;
14 end
15 scatter(xfinal,yfinal,10,"filled") ;   % plot the random walk
16 axis equal;                            % equal dimensions for axes
17 xlabel('x','FontSize',14);             % x-axis label
18 ylabel('y','FontSize',14);             % y-axis label
```

Code 2: prob1ab.m - Scatter of final positions of 2000 2D random walks

Results – A scatter plot of final positions after 2000 time steps, of 2000 particles undergoing two dimensional random walk is shown in figure 1b.

(c) **Methods** – Using elements from code 2, final positions of 5×10^5 particles are computed. The functions `tic` and `toc` are used to compute the time elapsed in execution of the code. With the help of these functions, the total number of particles is chosen, based purely on estimate of the maximum computations that can be performed in feasible time. So, The initial 18 lines of code 3 are essentially adopted from code 2, with an exception of line 17, where the modulus of displacement of a particle from origin is calculated. The following lines evaluate and plot the histogram of number distribution of particles with distance travelled. Using a bin width of $dr = 0.05$, binedges are computed and the histogram is plotted on line 24.
An alternate method in which counts in each bin are calculated to produce the histogram, was used to confirm results, and is not shown in code 3.

Results – The histogram showing particle number distribution, $N(r, r + \Delta r)$, of 500000 particles for $\Delta r = 0.05$ is given in figure 1c.

(d) **Methods** – Initial lines of code 4, till line 27, are adopted from code 3 and the value of `np` is set to 10^5, i.e. histogram is generated for 10^5 particles. Theoretically expected number distribution

```
clear;                                          % clear variables and functions
tic;                                            % start clock
d        = 0.01;
np       = 5.e5;                                % number of particles
tstep    = 2000

for i = 1:np
    x = 0;                                      % initial x at origin
    y = 0;                                      % initial y at origin
    for j = 1:tstep
        theta    = 2*pi*rand();                 % random theta between zero and 2pi
        x        = x+d*cos(theta);              % next value of x
        y        = y+d*sin(theta);              % next value of y
    end
    xfinal(i)    = x;
    yfinal(i)    = y;
    r(i)         = sqrt(x^2+y^2);
end

dr           = 0.05;                            % bin width
binedges     = 0:dr:max(r)+dr;                  % bin edges. Starts at zero,
                                                % step size of binwidth,
                                                % ends at ceil of maximum value of r.
histogram(r,binedges);
grid on;
xlabel('Final value of r','FontSize',14)
ylabel('Counts','FontSize',14)
toc                                             % stop clock
```

Code 3: prob1ac.m - Number distribution with distance of 5×10^5 particles

is then computed for each bin in lines 30-33, and the obtained curve is overlayed on the histogram. Analytically calculated number distribution of particles with distance is given by equation (3) of project notes as,

$$N(r, r + \Delta r) = N_p \exp\left(-\frac{r^2}{nd^2}\right)\left\{1 - \exp\left(-\frac{\Delta r(2r + \Delta r)}{nd^2}\right)\right\}, \tag{1}$$

where, $N(r, r + \Delta r)$ is the number of particles that have eventually travelled an absolute value of displacement in the range $[r, r + \Delta r]$, N_p is the total number of particles, n is the number of time steps and d is the jump size at each time step. This equation 1 is encoded in code 4 in lines 31 and 32.

```
28 hold on;
29 midpoints       = binedges+dr/2;         % midpoints of bins, for plotting
30 for i = 1:length(binedges)
31     n(i) = np * exp(-(binedges(i)^2)/(tstep*d*d))...
32            * (1-exp((-(2*binedges(i)*dr+dr*dr))/(tstep*d*d)));
33 end
34
35 plot(midpoints,n,'LineWidth',2);
```

Code 4: (Part of) prob1ad.m - Analytical estimates and observed number distributions with distance of 10^5 particles

Results – The histogram for 10^5 particles is shown in figure 1d. The analytical estimates is overlayed on the same plot.

Discussion – For a large number of particles, the observed number distribution seems to coincide well with the theoretical estimates, as seen in figure 1d.

(e) **Methods** – The number distribution of photons in this case is given by,

$$N(r)dr = 2\pi r \rho(r) dr, \tag{2}$$

where, $\rho(r)$ is the particle density distribution, given by,

$$\rho(r) = \frac{N_p}{\pi n d^2} \exp\left(-\frac{r^2}{nd^2}\right) \tag{3}$$

Substituting equation 3 in equation 2, we obtain,

$$N(r)dr = \frac{2rN_p}{nd^2} \exp\left(-\frac{r^2}{nd^2}\right) dr.$$

At peak number density, derivative of n with respect to displacement r equals zero. From this we obtain,

$$\frac{dN}{dr} = \frac{2N_p}{nd^2} \frac{d}{dr}\left[r \exp\left(-\frac{r^2}{nd^2}\right)\right] = 0$$

$$\Rightarrow \frac{2N_p}{nd^2} \exp\left(-\frac{r^2}{nd^2}\right)\left[1 - \frac{2r^2}{nd^2}\right] = 0$$

$$\Rightarrow n = \frac{2r^2}{d^2} \tag{4}$$

Using value of $r = R_\odot = 7 \times 10^8$ m and $d = 1$ mm $= 10^{-3}$ m in equation 4, we get, number of time steps,

$$n = 9.8 \times 10^{23}. \tag{5}$$

Total distance travelled by the photon before reaching the surface is, $s =$ number of time steps $\times$ distance travelled in each time step, i.e.,

$$s = 9.8 \times 10^{23} \times 10^{-3} = 9.8 \times 10^{20}\text{m}. \tag{6}$$

With photon travelling at speed of light, c, the time elapsed is given by,

$$t = \frac{s}{c} \approx 3.267 \times 10^{12}\text{sec} \approx 10^5\text{years} \tag{7}$$

Discussion – The rough estimate of time taken by photon considers an ideal case with several approximations. Despite that, the calculation gives us a reasonable order of magnitude estimate in equation 7, which is quite close to the value of a few million years, usually obtained for a realistic case.

3 Conclusions

Part II
Circular Binary

1 Introduction

2 Methods, Results and Discussions

(a) **Methods** – The dimensionless masses and positions of primary and secondary stars are respectively given by,

$$\begin{aligned} \tilde{m}_p &= \frac{m_p}{m} \text{ and } \tilde{m}_s = \frac{m_s}{m}, \\ \tilde{x}_p &= \frac{x_p}{a}, \text{ and } \tilde{x}_s = \frac{x_s}{a}, \end{aligned} \tag{8}$$

where, m and a are the total mass and separation of binary respectively. For a circular binary, the gravitational force experienced by each star equals the centripetal force on the star. Mathematically,

$$\begin{aligned} \frac{Gm_s m_p}{a^2} &= m_p x_p \omega^2 = m_s x_s \omega^2, \\ \Rightarrow x_s &= \frac{x_p m_p}{m_s}. \end{aligned} \tag{9}$$

Total separation of binary is the sum of positions of primary and secondary stars with respect to their centre of mass, *i.e.*

$$x_p + x_s = a. \tag{10}$$

Using equations 9 and 10, we obtain,

$$\begin{aligned} x_p + \frac{x_p m_p}{m_s} &= a \\ \Rightarrow x_p \left(\frac{m}{m_s}\right) &= a \\ \Rightarrow \tilde{x}_p &= \tilde{m}_s \end{aligned} \tag{11}$$

This relation gives us the relation for dimensionless position of primary star. Similarly, we can also obtain the relation for dimensionless position of secondary star as,

$$\begin{aligned} x_s + \frac{x_s m_s}{m_p} &= a \\ \Rightarrow \tilde{x}_s &= \tilde{m}_p \end{aligned} \tag{12}$$

If we substitute the value of angular velocity, $\omega = v_p/r_p = v_s/r_s$, in equation 9 for the two stars, we get,

$$\begin{aligned} \frac{Gm_s m_p}{a^2} &= \frac{m_p v_p^2}{x_p} = \frac{m_s v_s^2}{x_s} \\ \Rightarrow \frac{v_p}{v_s} &= \frac{m_s}{m_p} \end{aligned} \tag{13}$$

Multiplying both numerator and denominator of RHS by $1/m$, we obtain,

$$\frac{v_p}{v_s} = \frac{\tilde{m}_s}{\tilde{m}_p} \tag{14}$$

If we use the first relation of equation 13, we get,

$$\begin{aligned} \frac{Gm_s m_p}{a^2} &= \frac{m_p v_p^2}{x_p} \\ \Rightarrow v_p^2 &= \frac{Gm_s x_p}{a^2} \\ \Rightarrow v_p^2 &= \frac{Gm}{a}\tilde{m}_s \tilde{x}_p = \frac{Gm}{a}\tilde{m}_s^2, \end{aligned} \tag{15}$$

Parameter (Primary Star)	**Value**	**Parameter** (Secondary star)	**Value**
$\tilde{x}_{px}$	$-\tilde{m}_s$	$\tilde{x}_{sx}$	$\tilde{m}_p$
$\tilde{x}_{py}$	0	$\tilde{x}_{sy}$	0
$\tilde{v}_{px}$	0	$\tilde{v}_{sx}$	0
$\tilde{v}_{py}$	$-\tilde{m}_s$	$\tilde{v}_{sy}$	$\tilde{m}_p$

Table 1: A summary of initial conditions for the simulation. All quantities are mentioned in dimensionless units, for a binary in x-y plane rotating counter-clockwise, with centre of mass at origin and primary star on the negative x-axis and secondary star on positive x-axis initially.

where, we have substituted values from equation set 8 and equation 11. In units of $\sqrt{Gm/a}$, we obtain the value of dimensionless velocity of primary star as,

$$\tilde{v}_p = \tilde{m}_s \tag{16}$$

By using equations 13 and 16, we get dimensionless velocity of secondary as,

$$\tilde{v}_s = \tilde{m}_p \tag{17}$$

Given that angular velocity $\omega = 2\pi/T$, where T is the period of the circular binary, we can find the dimensionless time by using equation 9, as,

$$\begin{aligned}\frac{Gm_s m_p}{a^2} &= m_p x_p \omega^2 \\ \Rightarrow \frac{Gm_s}{a^2} &= \frac{4\pi^2 x_p}{T^2}\end{aligned}$$

Using equations 8 and 11, we get,

$$\begin{aligned}\frac{Gm_s}{a^3} &= \frac{4\pi^2 m_s}{T^2 m} \\ \Rightarrow T &= 2\pi\sqrt{\frac{a^3}{Gm}}.\end{aligned} \tag{18}$$

In units of $\sqrt{a^3/Gm}$, we therefore derive dimensionless time of a circular binary as,

$$\tilde{T} = 2\pi \tag{19}$$

Results – In the given scenario, the stars lie in x-y plane, with their centre of mass at origin. Primary star has initial coordinates $(-x_{px}, 0)$ while the secondary stars lies initially at $(x_{py}, 0)$. The stars rotate counter-clockwise, which gives us the velocity of primary star, $-v_{py}$ and the velocity of secondary star, v_{sy}. Using equations 11, 12, 16 and 17, the dimensionless values of initial conditions of the stars are summarized in table 1.
The value of dimensionless time period of binary is derived in equation 19 as 2π.

(b) **Methods** – Given that,

$$\frac{m_p}{m_s} = 4 \tag{20}$$

Adding 1 on both side, we get,

$$\begin{aligned}\frac{m_p}{m_s} + 1 &= 5 \\ \Rightarrow \frac{m_p + m_s}{m_s} &= \frac{m}{m_s} = 5 \\ \Rightarrow \tilde{m}_s &= \frac{m_s}{m} = \frac{1}{5} = 0.2\end{aligned} \tag{21}$$

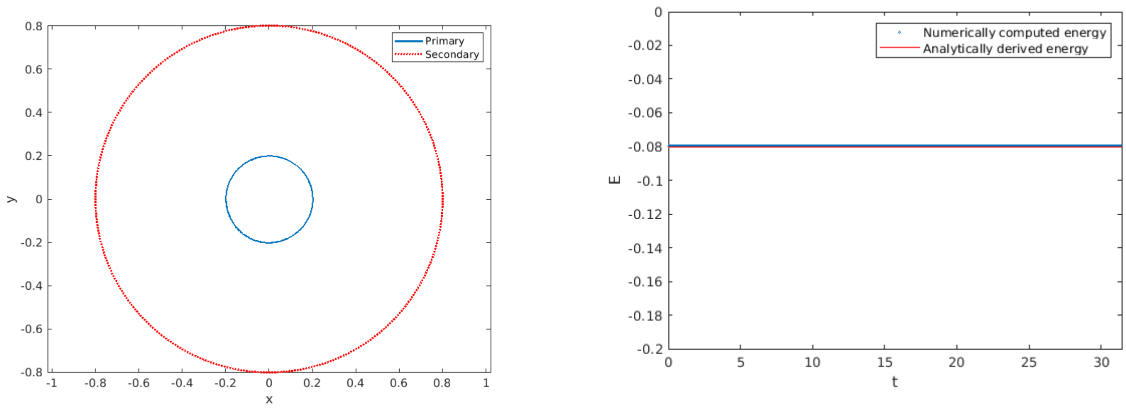
Using equations 8, 20 and 21, we can obtain the dimensionless mass of primary star as,

$$\tilde{m_p} = 4 \times \tilde{m_s} = 0.8 \tag{22}$$

Results – Using relations mentioned in table 1, the initial conditions for the circular binary are defined in Matlab file *binary.m* a shown in lines 13–20 of code 5. Using equations 21 and 22, the values of dimensionless masses of two stars are assigned in lines 11 and 12 as `mp = 0.8` and `ms = 0.2`.
The value of `tmax` is set ot 10π. Since from equation 19, it is known that the value of dimensionless time is 2π, this given value of `tmax` will allow simulation to sun for 5 orbits of the binary.

```
 9 h      = 1.d-2;       % time-step size
10 Ns     = 1000 ;       % sampling
11 mp     = 0.8  ;       % primary mass
12 ms     = 0.2  ;       % secondary mass
13 x(1)   = -ms  ;       % primary x
14 x(2)   = 0    ;       % primary y
15 x(3)   = 0    ;       % primary vx
16 x(4)   = -ms  ;       % primary vy
17 x(5)   = mp   ;       % secondary x
18 x(6)   = 0    ;       % secondary y
19 x(7)   = 0    ;       % secondary vx
20 x(8)   = mp   ;       % secondary vy
21 tmax   = 10*pi;       % final time
```

Code 5: binary.m - Initial conditions of circularly rotating binary for a simulation of five orbital rotations.



(a) Trajectories of primary and secondary stars in a circular binary system over the first five rotations.

(b) Analytically estimated and numerically calculated evolution of orbital energy over five rotations. The lines are seen to overlap, indicating good agreement of theoretical expectations with numerical results.

Figure 2: Plotted results for problem 2.

(c) **Methods** – Executing the Matlab code resulted in creation of file named *out*. This file is then used to plot the orbits of the two stars using the pre-existing program *orbitplot.m*.

Results – The obtained plot, which shows the trajectories of the two stars over 5 rotations is shown in figure 2a.

Discussions – It can be observed that dimensionless orbital radii of primary and secondary stars are 0.2 and 0.8 respectively. The orbits are circular around the centre of mass lying at origin.

(d) **Methods** – The energy of binary star system is given by,

$$E = \frac{1}{2}m_p v_p^2 + \frac{1}{2}m_s v_s^2 - \frac{G m_p m_s}{r} \tag{23}$$

From equation 13, we van obtain relations for velocities of primary and secondary stars at any

```
1  mp      = 0.8   ;
2  ms      = 0.2   ;
3  tmax    = 10*pi ;
4  load out                          % load the data file out
5  t       = out(:,1)  ;             % time
6  E       = out(:,6)  ;             % energy
7  exact   = -mp*ms/2  ;             % analytic estimate
8  
9  plot(t,E,'o','MarkerSize',0.9,...
10     'DisplayName','Numerically computed energy')    % numerical energy
11 hold on
12 plot([0 tmax],[exact exact],'r',...
13     'DisplayName','Analytically derived energy')    % exact line
14 xlabel('t','FontSize',12)
15 ylabel('E','FontSize',12)
16 xlim([0 tmax])
17 ylim([-0.2 0])
18 set(gca, 'Fontsize', 10)
19 legend
```

Code 6: second code

positions in their orbits as,

$$\begin{aligned} \frac{Gm_s m_p}{r^2} = \frac{m_p v_p^2}{x_p} = \frac{m_s v_s^2}{x_s} \\ \Rightarrow v_p^2 = \frac{Gm_s x_p}{r^2} \text{ and } v_s^2 = \frac{Gm_p x_s}{r^2} \end{aligned} \tag{24}$$

Substituting these values in equation 23 and using binary separation, $r = a = x_p + x_s$ from equation 10, we get

$$\begin{aligned} E &= \frac{Gm_p m_s x_p}{2a^2} + \frac{Gm_s m_p x_s}{2a^2} - \frac{Gm_p m_s}{a} \\ &= \frac{Gm_p m_s}{2a} - \frac{Gm_p m_s}{a} \\ &= -\frac{Gm_p m_s}{2a} \end{aligned} \tag{25}$$

In units of Gm^2/a, the dimensionless energy is given by,

$$\tilde{E} = \frac{-\dfrac{Gm_p m_s}{2a}}{\dfrac{Gm^2}{a}} = -\frac{\tilde{m}_p \tilde{m}_s}{2}. \tag{26}$$

The values of m_p, m_s and $tmax$ are assigned to variables `mp`, `ms` and `tmax` in lines 1-3 of code 6. The derived value of dimensionless energy is specified in line 7 of the code. The program in code 6 plots the time evolution of numerically and analytically calculated energies of the binary.

Results – The plot generated using code 6 is shown in figure 2b. The analytically estimated value of energy of the binary is given by equation 26.

Discussions – From equation 26 we can conclude that the dimensionless energy of binary star system is constant and less than zero. In this case, its value from equation 26 turns out to be -0.08. Figure 2b shows that numerically obtained energy of the system is similar to analytically calculated energy, both showing no variation in value over time.

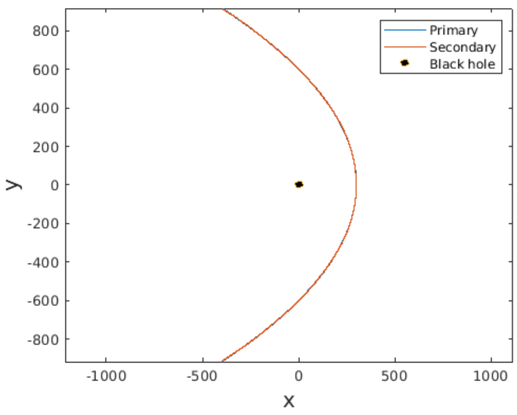
3 Conclusions

Part III

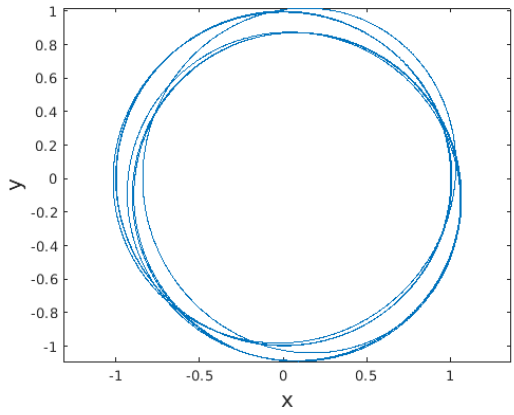
Hypervelocity Stars

1 Introduction

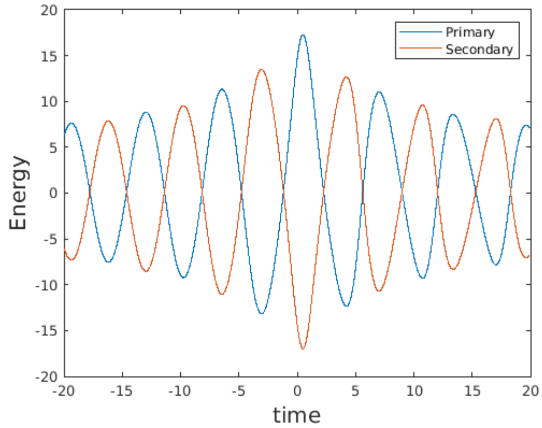
2 Methods, Results and Discussions



(a) 3a



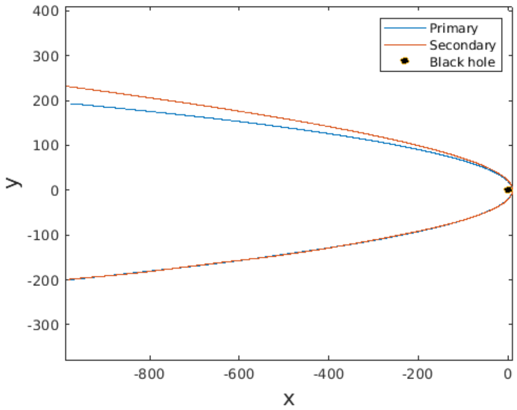
(b) 3b



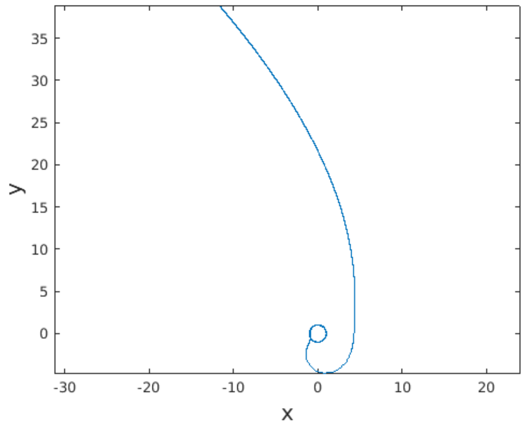
(c) 3c

Figure 3: Problem 3 - fig 1

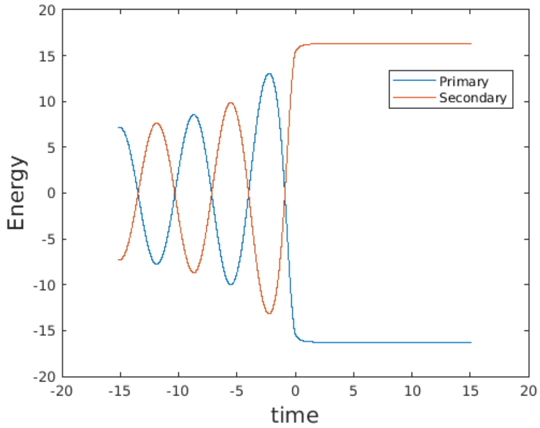
```
Rt       = (mb)^(1/3)                          ;   % tidal radius
R0       = 10 * Rt                             ;   % initial distance between BH and binary
Rp       = 3 * Rt                              ;   % periastron radius
f0       = -acos(-1+(D/5))                     ;   % initial true anomaly (eq 44)
Rdot     = sin(f0) * mb^(1/3) / (sqrt(2*D))    ;   % dR/dt (eq 49a)
Fdot     = (1+cos(f0))^2 * sqrt(2) / ...
           (4*D^(3/2))                         ;   % df/dt (eq 49b)

xcmxdot = Rdot*cos(f0) - R0*Fdot*sin(f0)       ;   % d(xcmx)/dt (using eq 41a)
xcmydot = Rdot*sin(f0) + R0*Fdot*cos(f0)       ;   % d(xcmy)/dt (using eq 41b)

phi      = pi/2                                ;   % binary phase
rpxdot   = -ms*sin(phi+pi)                     ;   % d(rpx)/dt (using problem 2)
rpydot   = mp*cos(phi+pi)                      ;   % d(rpy)/dt (using problem 2)
rsxdot   = -mp*sin(phi)                        ;   % d(rsx)/dt (using problem 2)
rsydot   = ms*cos(phi)                         ;   % d(rsy)/dt (using problem 2)

t        = (sqrt(2)/3) * (D^(3/2)) * ...
           (tan(f0/2))*(3+(tan(f0/2))^2)       ;   % initial time t0
```



(a) 3da



(b) 3db



(c) 3dc

Figure 4: Problem 3 - fig 2

Quantity	Variable	D=3	D=0.1
t_0	`t`	-19.9555	-15.1290
x_p	`x(1)`	-400.0000	-980.0000
y_p	`x(2)`	-917.3151	-199.7975
v_{px}	`x(3)`	37.6166	44.6972
v_{py}	`x(4)`	24.4949	4.4721
x_s	`x(5)`	-400.0000	-980.0000
y_s	`x(6)`	-916.3151	-198.7975
v_{sx}	`x(7)`	36.6166	43.6972
v_{sy}	`x(8)`	24.4949	4.4721

Table 2: Table of variable values for two values of D

```
x(1)     = (R0*cos(f0)) + (mp*cos(phi+pi))   ;   % x_p
x(2)     = (R0*sin(f0)) + (mp*sin(phi+pi))   ;   % y_p
x(3)     = xcmxdot+rpxdot                    ;   % v_xp
x(4)     = xcmydot+rpydot                    ;   % v_yp
x(5)     = (R0*cos(f0)) + (ms*cos(phi))      ;   % x_s
x(6)     = (R0*sin(f0)) + (ms*sin(phi))      ;   % y_s
x(7)     = xcmxdot+rsxdot                    ;   % v_xs
x(8)     = xcmydot+rsydot                    ;   % v_ys
```

Code 7: code 1

```
function dxdt = f(t,x,mb,mp,ms)
  r = sqrt((x(1)-x(5))^2+(x(2)-x(6))^2)            ;
  rp = sqrt(x(1)^2 + x(2)^2)                       ;
  rs = sqrt(x(5)^2 + x(6)^2)                       ;
  dxdt(1) = x(3)                                   ;   % v_px
  dxdt(2) = x(4)                                   ;   % v_py
  dxdt(3) = (ms*(x(5)-x(1))/r^3)+mb*(-x(1)/rp^3)  ;   % a_px
  dxdt(4) = (ms*(x(6)-x(2))/r^3)+mb*(-x(2)/rp^3)  ;   % a_py
  dxdt(5) = x(7)                                   ;   % v_sx
  dxdt(6) = x(8)                                   ;   % v_sy
  dxdt(7) = (mp*(x(1)-x(5))/r^3)+mb*(-x(5)/rs^3)  ;   % a_sx
  dxdt(8) = (mp*(x(2)-x(6))/r^3)+mb*(-x(6)/rs^3)  ;   % a_sy
```

Code 8: code 2

(a) **Methods** –
Results –
Discussions –

(b) **Methods** –
Results –
Discussions –

(c) **Methods** –
Results –
Discussions –

(d) **Methods** –
Results –
Discussions –

3 Conclusions