

Numerical Methods in Astrophysics

Project 3

**Two Dimensional Random Walk,  
Circular Binary and  
Hypervelocity Stars**

*Saksham Kaushal*

# Contents

<b>I</b>	<b>Two Dimensional Random Walk</b>	<b>2</b>
1	Introduction	2
2	Methods, Results and Discussions	2
3	Conclusions	5
<b>II</b>	<b>Circular Binary</b>	<b>6</b>
1	Introduction	6
2	Methods, Results and Discussions	6
3	Conclusions	10
<b>III</b>	<b>Hypervelocity Stars</b>	<b>11</b>
1	Introduction	11
2	Methods, Results and Discussions	11
3	Conclusions	22
	<b>Appendices</b>	<b>23</b>
<b>A</b>	<b>Complete Matlab Programs</b>	<b>23</b>
A.1	Problem 1 . . . . .	23
A.2	Problem 2 . . . . .	24
A.3	Problem 3 . . . . .	26
A.3.1	Problems a–e . . . . .	26
A.3.2	Part f . . . . .	28

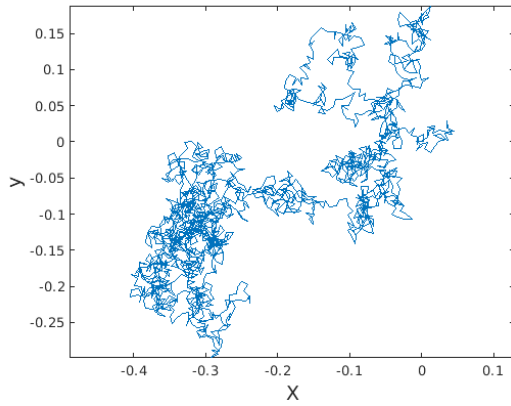
## Part I

# Two Dimensional Random Walk

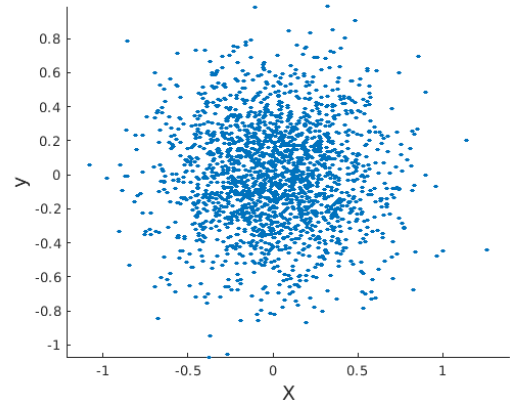
## 1 Introduction

A two dimensional 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. In this problem, such a random walk is visualized and certain aspects of it are analysed. This problem also investigates the number distribution of particles with distance, after a given number of steps in random walks are executed.

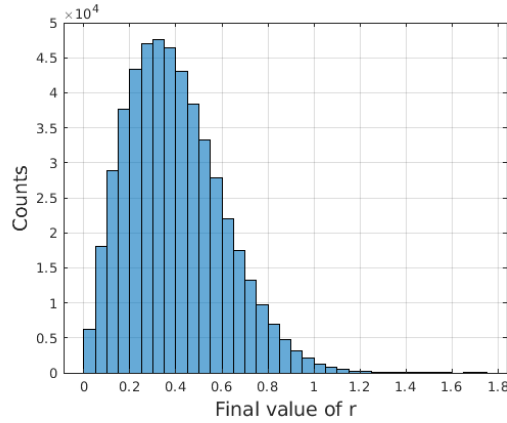
## 2 Methods, Results and Discussions



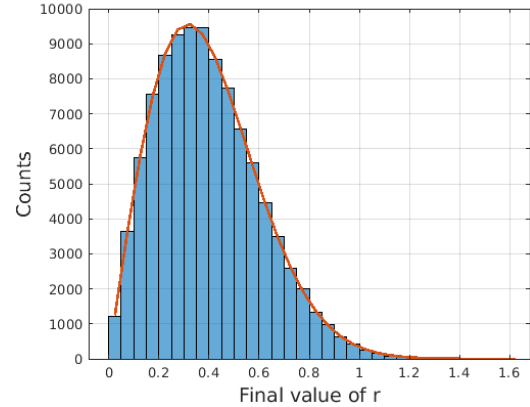
(a) Two dimensional random walk of a particle over 2000 time steps



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



(c) Number distribution with distance of  $5 \times 10^5$  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 numbers uniformly in the range  $[0, 1]$ , therefore, multiplying the obtained random number by  $2\pi$  give us the random angle  $\theta$ , 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.

```

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

```

**Code 1:** problaa.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 by assigning new variables 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:** problab.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  $5 \times 10^5$  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 adopted from code 2, with an exception of line 17, where the modulus of displacement of a particle from origin is calculated. Subsequent 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.

```

1 clear; % clear variables and functions
2 tic; % start clock
3 d = 0.01;
4 np = 5.e5; % number of particles
5 tstep = 2000
6
7 for i = 1:np
8     x = 0; % initial x at origin
9     y = 0; % initial y at origin
10    for j = 1:tstep
11        theta = 2*pi*rand(); % random theta between zero and 2pi
12        x = x+d*cos(theta); % next value of x
13        y = y+d*sin(theta); % next value of y
14    end
15    xfinal(i) = x;
16    yfinal(i) = y;
17    r(i) = sqrt(x^2+y^2);
18 end
19
20 dr = 0.05; % bin width
21 binedges = 0:dr:max(r)+dr; % bin edges. Starts at zero,
22 % step size of binwidth,
23 % ends at ceil of maximum value of r.
24 histogram(r,binedges);
25 grid on;
26 xlabel('Final value of r','FontSize',14)
27 ylabel('Counts','FontSize',14)
28 toc % stop clock

```

**Code 3:** problac.m - Number distribution with distance of  $5 \times 10^5$  particles

- (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 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,

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

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:** (Snippet of) problad.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,

$$N(r)dr = 2\pi r\rho(r)dr, \quad (2)$$

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

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

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,

$$\begin{aligned} \frac{dN}{dr} &= \frac{2N_p}{nd^2} \frac{d}{dr} \left[ r \exp\left(-\frac{r^2}{nd^2}\right) \right] = 0 \\ \Rightarrow \frac{2N_p}{nd^2} \exp\left(-\frac{r^2}{nd^2}\right) \left[ 1 - \frac{2r^2}{nd^2} \right] &= 0 \\ \Rightarrow n &= \frac{2r^2}{d^2} \end{aligned} \quad (4)$$

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,

$$n = 9.8 \times 10^{23}. \quad (5)$$

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.,

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

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

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

**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

For a large sample of random walks, the distribution of eventual positions of particles is given by equation 1. The theoretical estimates and observations made using numerical simulation seem to agree well. One example of (approximation of) random walk considered here is the motion of a photon through the layers of Sun. This causes the photon to take several years (of the order of millions) to reach the surface of the Sun before eventually escaping.

## Part II

# Circular Binary

## 1 Introduction

A system of two stars rotating in circular orbits around their common centre of mass is called a *circular binary*. The heavier star is called the primary star whereas the lighter one is called the secondary star. The orbital time period for both the stars is equal and therefore, primary star traverses a small orbit at slower velocity, while the orbit of the secondary star is larger which it traverses at a higher velocity. In most of the commonly encountered systems, correlation between the two stars is extremely significant in terms of their evolution. Such a gravitationally bound system possesses a negative total energy.

## 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 equation gives us the relation between dimensionless position of primary star and dimensionless mass of secondary 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,

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

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  $2\pi$ .

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.



(b) **Methods** – Given that,

$$\frac{m_p}{m_s} = 4 \quad (20)$$

Adding 1 on both side, we get,

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

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

$$\tilde{m}_p = 4 \times \tilde{m}_s = 0.8 \quad (22)$$

**Results** – Using relations mentioned in table 1, the initial conditions for the circular binary are defined in Matlab file *binary.m* as 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  $m_p = 0.8$  and  $m_s = 0.2$ .

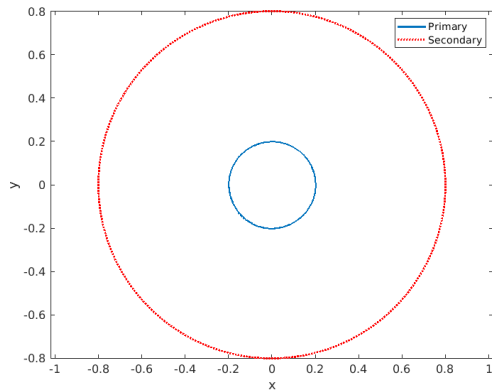
The value of **tmax** is set to  $10\pi$ . Since from equation 19, it is known that the value of dimensionless time is  $2\pi$ , this given value of **tmax** will allow simulation to run for 5 orbits of the binary.

```

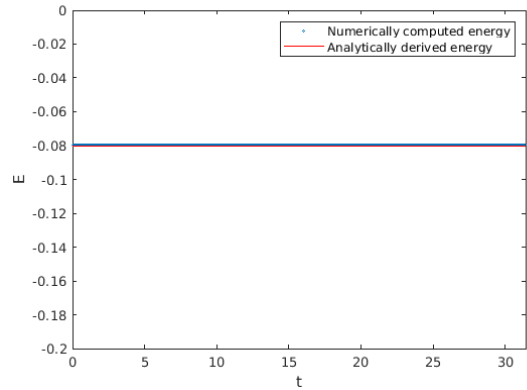
9 | h      = 1.d-2;      % time-step size
10 | Ns     = 1000 ;      % sampling
11 | mp     = 0.8 ;      % primary mass
12 | ms     = 0.2 ;      % secondary mass
13 | x(1)   = -ms ;      % primary x
14 | x(2)   = 0 ;        % primary y
15 | x(3)   = 0 ;        % primary vx
16 | x(4)   = -ms ;      % primary 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 numerical results with theoretical expectations.

**Figure 2:** Plotted results for problem 2.

- (c) **Methods** – Executing the Matlab code resulted in creation of file named *out*. Data obtained in 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,

$$E = \frac{1}{2}m_p v_p^2 + \frac{1}{2}m_s v_s^2 - \frac{Gm_p m_s}{r} \quad (23)$$

From equation 13, we can obtain relations for velocities of primary and secondary stars at any positions in their orbits as,

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

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

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

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

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

```

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:** energyplot.m - Program to plot the time evolution of analytically and numerically calculated energies of a circular binary system.

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

This initial positions and velocity components of primary and secondary stars in a circular binary have been calculated. Using fourth order Runge-Kutta method, first order ordinary differential equations are integrated to obtain values of positions and velocities of stars in a binary star system at subsequent time steps. The total energy of the bound system is constant and negative.

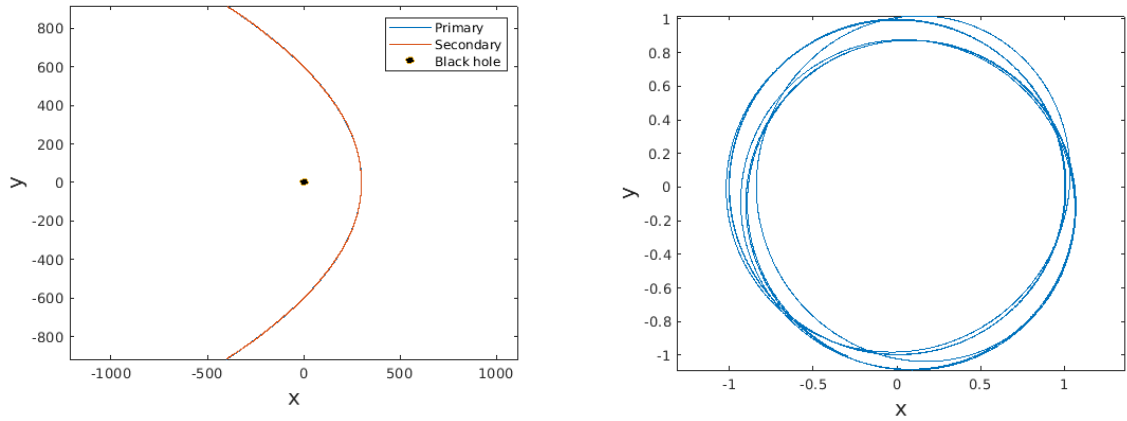
## Part III

# Hypervelocity Stars

## 1 Introduction

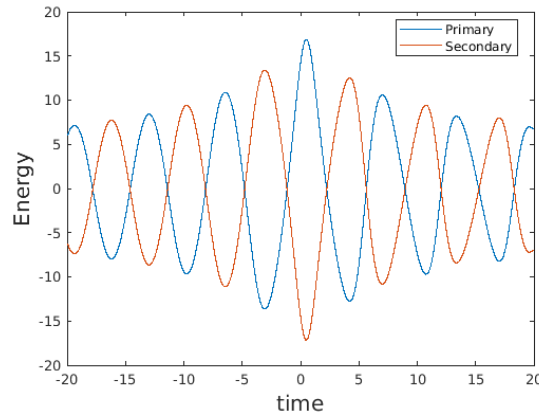
Hypervelocity stars are obtained when a binary star system interacts tidally with a black hole. Depending on several factors, one or both stars of a binary may be ejected at extremely high velocities of the order of several thousand kilometres per second. Although another possible phenomenon that might result in ejection of hypervelocity stars from a binary when one star goes supernova has been highlighted in many recent texts, this shall not be the topic of study of this problem. Here, the ejection of hypervelocity stars from binary on close encounter with a black hole and its dependence on several parameters like penetration factor, orientation of rotation, masses of stars and black hole, etc. are studied.

## 2 Methods, Results and Discussions



(a) Orbits of two stars in a binary star system in the black hole rest frame.

(b) Orbit of secondary star in rest frame of primary star.



(c) Time evolution of dimensionless energies of the two stars in a binary star system.

**Figure 3:** Plotted results for problems 3(a)–(c)

- (a) **Methods** – The function `dx dy` defined in Matlab code file `f.m` is modified to include black hole gravity terms in lines 7,8,11 and 12 of code 7. These modifications are done according to the equations,

$$\begin{aligned} E_p &= \frac{1}{2}m_p v_p^2 - \frac{Gm_p M}{r_p} - \frac{Gm_p m_s}{r_{ps}}, \\ E_s &= \frac{1}{2}m_s v_s^2 - \frac{Gm_s M}{r_s} - \frac{Gm_p m_s}{r_{ps}}, \end{aligned} \quad (27)$$

```

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

```

**Code 7:** f.m - Function defining values of velocities and accelerations of the two stars along the two axes.

```

1 Rt      = (mb)^(1/3) ; % tidal radius
2 R0      = 10 * Rt ; % initial distance between BH and binary
3 f0      = -acos(-1+(D/5)) ; % initial true anomaly (eq 44)
4 Rdot    = sin(f0) * mb^(1/3) / (sqrt(2*D)) ; % dR/dt (eq 49a)
5 Fdot    = (1+cos(f0))^2 * sqrt(2) / ...
6          (4*D^(3/2)) ; % df/dt (eq 49b)
7
8 xcmxdot = Rdot*cos(f0) - R0*Fdot*sin(f0) ; % d(xcmx)/dt (using eq 41a)
9 xcmydot = Rdot*sin(f0) + R0*Fdot*cos(f0) ; % d(xcmy)/dt (using eq 41b)
10
11 phi     = pi/2 ; % binary phase
12 rpxdot  = -ms*sin(phi+pi) ; % d(rpx)/dt (using problem 2)
13 rpydot  = ms*cos(phi+pi) ; % d(rpy)/dt (using problem 2)
14 rsxdot  = -mp*sin(phi) ; % d(rsx)/dt (using problem 2)
15 rsydot  = mp*cos(phi) ; % d(rsy)/dt (using problem 2)
16
17 t       = (sqrt(2)/3) * (D^(3/2)) * ...
18          (tan(f0/2))*(3+(tan(f0/2))^2) ; % initial time t0
19 x(1)    = (R0*cos(f0)) + (ms*cos(phi+pi)) ; % x_p
20 x(2)    = (R0*sin(f0)) + (ms*sin(phi+pi)) ; % y_p
21 x(3)    = xcmxdot+rpxdot ; % v_xp
22 x(4)    = xcmydot+rpydot ; % v_yp
23 x(5)    = (R0*cos(f0)) + (mp*cos(phi)) ; % x_s
24 x(6)    = (R0*sin(f0)) + (mp*sin(phi)) ; % y_s
25 x(7)    = xcmxdot+rsxdot ; % v_xs
26 x(8)    = xcmydot+rsydot ; % v_ys

```

**Code 8:** initialc.m - Initial conditions for the simulation.

where,  $M$  is the mass of black hole and  $r_p$  and  $r_s$  are the distances from black hole to primary and secondary stars, respectively.

In dimensionless units, the mass of black hole is given by,

$$\tilde{M} = \frac{M}{m_p + m_s} = \frac{M}{m} \quad (28)$$

In the given scenario,  $m_p = 3.2M_\odot$ ,  $m_s = 0.8M_\odot$  and  $M = 4 \times 10^6 M_\odot$ . This gives us  $m = 3.2 + 0.8 = 4M_\odot$ . Consequently, using equations 8 and 28, dimensionless masses are given by,

$$\begin{aligned} \tilde{m}_p &= 0.8 \\ \tilde{m}_s &= 0.2 \\ \tilde{M} &= 10^6 \end{aligned} \quad (29)$$

These values are assigned in the initial lines of the main code *HVS.m* to variables `mp`, `ms` and `mb`, respectively.

Next, the initial conditions are specified in the file *initialc.m*, shown in code 8. The initial position of primary star with respect to black hole is given by,

$$x_p = X_{CM} + r_p, \quad (30)$$

where,  $X_{CM}$  is the position of of centre of mass of binary from black hole, and  $r_p$  is the position of primary star with respect to centre of mass of binary. All these quantities are vectors and arrows have been dropped here (and later) for simplicity, and the approximation that centre of mass of the whole system lies within the black hole is assumed to be valid here, given that  $M \gg m_p, m_s$ .

The value of  $X_{CM}$  in equation 30 is given by,

$$X_{CM} = (R \cos(f), R \sin(f)), \quad (31)$$

where,

$$R = \frac{2R_p}{1 + \cos(f)}. \quad (32)$$

The angle  $f$ , called *true anomaly*, is the angle from point of closest approach and is a function of time.  $R_p$  is the *periastron*, which is the distance of closest approach. The initial value of true anomaly is given as,

$$f_0 = -\cos^{-1} \left( \frac{D}{5} - 1 \right), \quad (33)$$

where,

$$D = \frac{R_p}{R_t}, \quad (34)$$

is called the penetration factor.

For a given phase angle  $\phi$ , the value of  $r_p$  in equation 30 is given by,

$$r_p = (m_s \cos(\phi + \pi), m_s \sin(\phi + \pi)) \quad (35)$$

Using equations 30, 31 and 35, we obtain the coordinates of primary star as,

$$\begin{aligned} x_{px} &= R \cos(f) + m_s \cos(\phi + \pi) \\ x_{py} &= R \sin(f) + m_s \sin(\phi + \pi) \end{aligned} \quad (36)$$

Similarly, for secondary star,

$$x_s = X_{CM} + r_s, \quad (37)$$

$$r_s = (m_p \cos(\phi), m_p \sin(\phi)) \quad (38)$$

Using equations 31, 37 and 38, we obtain the coordinates of secondary star as,

$$\begin{aligned} x_{sx} &= R \cos(f) + m_p \cos(\phi) \\ x_{sy} &= R \sin(f) + m_p \sin(\phi). \end{aligned} \quad (39)$$

For the given initial conditions,

$$R_0 = 10R_t, \quad (40)$$

where  $R_t$  is the *tidal radius*, given by,

$$R_t = \left( \frac{M}{m} \right)^{\frac{1}{3}} a$$

In dimensionless units, *i.e.*, in units of  $a$ , tidal radius is given by,

$$\tilde{R}_t = \left( \tilde{M} \right)^{\frac{1}{3}}. \quad (41)$$

Therefore, from equations 40 and 41,

$$\tilde{R}_0 = 10\tilde{M}^{\frac{1}{3}} \quad (42)$$

Velocity of primary star is given by differentiating equation 30 with respect to time as,

$$v_p = \dot{x}_p = \dot{X}_{CM} + \dot{r}_p. \quad (43)$$

Time derivative of  $X_{CM}$  is given by differentiating equation 31 as,

$$\begin{aligned}\dot{X}_{CM} &= \frac{d}{dt}(R \cos(f)), \frac{d}{dt}(R \sin(f)) \\ \Rightarrow \dot{X}_{CMx} &= \dot{R} \cos(f) - R \dot{f} \sin(f), \\ \dot{X}_{CMy} &= \dot{R} \sin(f) + R \dot{f} \cos(f)\end{aligned}\tag{44}$$

Here, tilde has been dropped from dimensionless variables for simplicity and  $\dot{R}$  and  $\dot{f}$  are given as,

$$\dot{R} = \frac{M^{1/3}}{\sqrt{2D}} \sin(f),\tag{45}$$

$$\dot{f} = \frac{\sqrt{2}}{4} D^{-3/2} (1 + \cos(f))^2.\tag{46}$$

The binary star system can be assumed to be in the similar initial conditions as the one considered in part II of this project, *i.e.* a circular binary rotating counter-clockwise in x-y plane with stars initially located on x-axis (Changing the phase angle  $\phi$  updates this condition accordingly). This will give us the derivative of equation 35,  $\dot{r}_p$  as,

$$\begin{aligned}\dot{r}_{px} &= -m_s \sin(\phi + \pi), \\ \dot{r}_{py} &= m_s \cos(\phi + \pi).\end{aligned}\tag{47}$$

Similarly, equation 43 for secondary star translates to,

$$v_s = \dot{x}_s = \dot{X}_{CM} + \dot{r}_s\tag{48}$$

Value of  $X_{CM}$  is given by equation set 31, while the value of  $r_s$  can be computed for secondary star using the method similar to the one used for primary star for equation set 47, adopted from part II of the project. Using equation 38 this gives us,

$$\begin{aligned}\dot{r}_{sx} &= -m_p \sin(\phi), \\ \dot{r}_{sy} &= m_p \cos(\phi).\end{aligned}\tag{49}$$

At  $t = 0$ , the binary passes through the periastron. This implies that the evolution of code needs to begin at time  $t < 0$ , which can be obtained as a function of initial true anomaly  $f_0$  discussed in equation 33. This gives us the initial time of

$$t_0 = \frac{\sqrt{2}}{3} D^{3/2} \tan\left(\frac{f_0}{2}\right) \left(3 + \tan^2\left(\frac{f_0}{2}\right)\right).\tag{50}$$

This gives us a complete set of initial conditions that can be used for simulation – initial time, initial positions and initial velocities. These relations can be used in the Matlab file *initialc.m* given in code 8, to define the initial conditions.

To summarize,

- The dimensionless values of masses (equation set 29) and  $D = 3$  (equation 34) are initialized in the beginning of file *HVS.m*.
- In file *initialc.m*, shown in code 8, values of tidal radius,  $R_t$  (equation 41) and initial distance between binary and black hole,  $R_0$  (equation 40) are defined in lines 1 and 2.
- The values of  $\dot{X}_{CM}$  (velocity equation set 44) along the two axes are defined in lines 8 and 9.
- Calculating the values of  $\dot{X}_{CM}$  requires values of true anomaly ( $f_0$ ), time derivative of radial distance  $\dot{R}$  and time derivative of true anomaly  $\dot{f}$ , given by equations 33, 45 and 46, respectively. These values are defined in lines 3–6.
- A phase angle  $\phi = \pi/2$  is assumed and initialized in line 11.
- Initial time  $t_0$  is calculated using equation 50 in line 17–18.

```

12 plot(xp,yp,'DisplayName','Primary');
13 hold on;
14 plot(xs,ys,'DisplayName','Secondary');
15 hold on;
16 plot(0,0,'o','MarkerSize',5.5, 'MarkerFaceColor','black','DisplayName','Black hole')
17 axis equal;
18 xlabel('x','FontSize',15);
19 ylabel('y','FontSize',15);
20 legend;

```

**Code 9:** script4.m - Code snippet to plot the trajectories of stars in black hole rest frame.

```

22 plot(xs-xp,ys-yp);
23 axis equal;
24 xlabel('x','FontSize',15);
25 ylabel('y','FontSize',15);

```

**Code 10:** script4.m - Code snippet to plot the trajectory of secondary star in the rest frame of primary mass.

- Initial positions of the two stars are computed in lines 19–20 and 23–24 using equation sets 35 and 38.
- The values of initial velocities derived in equations 43 and 48 are assigned in lines 21–22 and 25–26.
- Values of velocities depend on velocity of centre of mass as well as velocity of each star in the binary. Velocity of centre of mass is already defined in lines 8–9, as mentioned above, while the velocities of each star along the two axes are defined in lines 12–15 using equation sets 47 and 49.

Using the above methods, the initial conditions are defined. Upon executing the code *HVS.m*, a file named *out* is created. The plotting of trajectories in black hole rest frame is done using Matlab code *script4.m*, a snippet of which is shown in code 9.

**Results** – The plot of obtained trajectories of the binary stars in the black hole rest frame is shown in figure 3a. The numerically computed values of initial conditions are summarised in table 2.

**Discussions** – The trajectory of the centre of mass of binary appears to be parabolic. This can be confirmed if the energy of the binary turns out to be roughly zero. Also, though the plot seems to show a single line of trajectory, on zooming in, the individual tracks of the two stars can be resolved. The structure and separation of binary does not seem to be affected by the interaction on close approach of the system at periastron.

- (b) **Methods** – Using the output file *out*, the trajectory of secondary star in units of dimensionless position in the comoving frame of primary star is plotted using the code snippet shown in code 10.

**Results** – The plot obtained is shown in figure 3b.

- (c) **Methods** – The energies of primary and secondary stars is given by,

$$\begin{aligned}
E_p &= \frac{1}{2}m_p v_p^2 - \frac{Gm_p M}{r_p} - \frac{Gm_p m_s}{r_{ps}} \\
E_s &= \frac{1}{2}m_s v_s^2 - \frac{Gm_s M}{r_s} - \frac{Gm_p m_s}{r_{ps}}
\end{aligned} \tag{51}$$

```

27 plot(t,Ep,'DisplayName','Primary');
28 hold on;
29 plot(t,Es,'DisplayName','Secondary');
30 xlabel('time','FontSize',15);
31 ylabel('Energy','FontSize',15);
32 legend('location','best');

```

**Code 11:** script4.m - Code snippet to plot evolution of energies of binary stars over time



Quantity	Variable	D=3	D=0.1
$t_0$	$t$	-19.9555	-15.1290
$x_p$	$x(1)$	-400.0000	-980.0000
$y_p$	$x(2)$	-916.7151	-199.1975
$v_{px}$	$x(3)$	37.6166	44.6972
$v_{py}$	$x(4)$	24.4949	4.4721
$x_s$	$x(5)$	-400.0000	-980.0000
$y_s$	$x(6)$	-915.7151	-198.1975
$v_{sx}$	$x(7)$	36.6166	43.6972
$v_{sy}$	$x(8)$	24.4949	4.4721

**Table 2:** Summary of initial values of various physical quantities in the simulation and their variable names. These values are enlisted for values  $D = 3$  and  $D = 0.1$ .

The energies in dimensionless units computed using *HVS.m* and *energy.m* are imported from *out* file and plotted using the code snippet shown in code 11.

**Results** – The plot of evolution of dimensionless energies of primary and secondary stars is shown in figure 3c.

**Discussions** – The total dimensionless energy of the system remains constant and roughly equal to zero over time, as can be observed from figure 3c.

- (d) **Methods** – Using all the same programs used in parts (a)–(c), the value of  $D$  is updated to a new value of 0.1 in Matlab code *HVS.m*. The plotting of data generated in *out* file is also done using the same codes.

**Results** – The plots generated are shown in figure 4. The trajectory in terms of dimensionless position of the two stars in black hole rest frame is shown in figure 4a. The orbit of the secondary star in rest frame of primary star is shown in figure 4b. The evolution of dimensionless energies of the two stars over time is shown in figure 4c. The values of initial parameters computed in the code are enlisted in table 2.

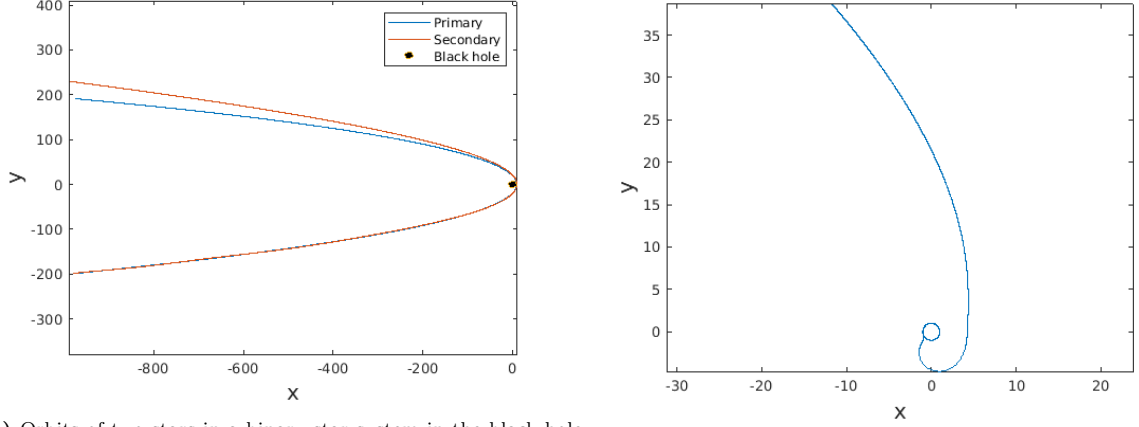
**Discussions** – The trajectory of the centre of mass of binary still appears to be parabolic. The separation of the binary is affected in later times, after the tidal interaction with the black hole. The binary separates over time as seen in the upper left corner of figure 4a. This break up of binary is essentially an effect of tidal interaction of binary with the black hole. Evidently, this becomes prominent as the binary passes very close to the super massive black hole, such as the one found in the centre of our galaxy.

The star detaches from binary after the tidal interaction. This is seen in figure 4b. In the rest frame of primary star, the secondary star is seen to initially encircle as the binary rotates, evident in lower central circular region of figure 4b. After gravitational interaction, the secondary star begins to increasingly move away from primary star.

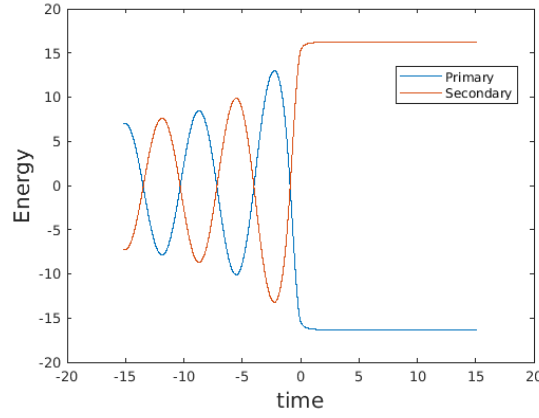
The total energy of the binary, which is the sum of the energies of the two stars is conserved over time and is roughly zero. After time  $t = 0$ , when the binary reaches the periastron, the energy of primary star becomes extremely negative, while the energy of the secondary star becomes extremely positive. This indicates that orbit of primary star becomes more bound to the black hole and vice versa. This suggests more strongly that deviations in orbits of stars of the binary are essentially due to gravitational interactions between binary and black hole, in case the point of approach is closer than some limiting value, decided by the value of penetration factor,  $D$ .

- (e) **Methods** – From previous part, we know that the star ejected from the binary after tidal interaction is the secondary star. From equation 51, we can estimate the final velocity at  $r = \infty$  using the value of velocity obtained from our numerical results in *out* file. At such large distances the following two approximations can be made safely,

$$\begin{aligned} \frac{Gm_s M}{r_s} &\approx 0, \\ \frac{Gm_p m_s}{r_{ps}} &\approx 0. \end{aligned} \tag{52}$$



(a) Orbits of two stars in a binary star system in the black hole rest frame. Note that the binary separation increases over time (b) Orbit of secondary star in rest frame of primary star. The stars are seen to depart away from each other at later times



(c) Time evolution of dimensionless energies of the two stars in a binary star system. The plot changes abruptly at  $t = 0$ , *i.e.* after tidal interaction.

**Figure 4:** Plotted results for problem 3(d).

This gives us,

$$E_s \approx \frac{1}{2} m_s v_s^2$$

$$\Rightarrow v_s \approx \sqrt{\frac{2E_s}{m_s}} \quad (53)$$

**Results** – Numerical values obtained from the simulation give an eventual dimensionless energy of  $\tilde{E}_s \approx 16.2244$  for the secondary star. Substituting for non-dimensional values in equation 53, we get,

$$v_s = \sqrt{\frac{2\tilde{E}_s G m^2}{m_s a}}$$

$$= \sqrt{\frac{2 \times 16.2244 \times (6.67 \times 10^{-11}) \times (4M_\odot)^2}{(7 \times R_\odot) \times (0.8 \times M_\odot)}} \quad (54)$$

$$= \sqrt{\frac{618.38 \times 10^{-11} M_\odot}{R_\odot}}$$

$$\approx 4204 \text{ km/s.}$$

Thus, for the secondary star, detached from the binary, we obtain a velocity of approximately 4204 km/s at a distance  $r = \infty$ .

(f) **Methods** – To answer whether the orientation of binary rotation affects the outcomes of our simulation, the orbital trajectories and energies of clockwise and counter-clockwise rotating

```

5 h = 1.d-3; % time-step size
6 Ns = 10000; % sampling
7 mb = 1.e6; % BH mass in units of the binary mass m
8 mp = 0.8; % primary star mass in units of m
9 ms = 0.2; % secondary star mass in units of m
10 for D=[ 1.0, 0.5, 0.1, 0.025, 0.0125 0.01]
11 [t, x] = initialc(D,mb,mp,ms); % Define the initial values of t and x.
12 tmax = -t;
13 tprint = t;
14 dtp = (tmax-t)/Ns;
15 filename = sprintf('%.4f',D);
16 while t < tmax
17     if t >= tprint
18         [Ep,Es] = energy(x,mb,mp,ms);
19         v = [t x(1) x(2) x(5) x(6) Ep Es];
20         save(filename, 'v', '-ascii', '-append')
21         tprint = tprint + dtp;
22     end
23     x = RK4(h,t,x,mb,mp,ms);
24     t = t+h;
25 end
26 end

```

**Code 12:** HVS.m - Program to perform simulation for different values of  $D$

```

16 rpxdot = ms*sin(phi+pi)
17 rpydot = -ms*cos(phi+pi)
18 rsxdot = mp*sin(phi)
19 rsydot = -mp*cos(phi)

```

**Code (13)** initialc.m - Initial velocities for clockwise rotating binaries.

```

16 rpxdot = -ms*sin(phi+pi)
17 rpydot = ms*cos(phi+pi)
18 rsxdot = -mp*sin(phi)
19 rsydot = mp*cos(phi)

```

**Code (14)** initialc.m - Initial velocities for counter-clockwise rotating binaries.

binaries are probed. These simulations are performed for values of  $D = 1.0, 0.5, 0.1, 0.025, 0.0125$  and  $0.01$ . Code 12 shows the program to perform simulation similar to one performed in previous section for different values of  $D$  iteratively.

The initial conditions for clockwise and counter-clockwise rotating binaries differ in their directions of initial velocities. Codes 13 and 14 can be compared to mark the differences in initial velocities of the two types of binaries. Rest of the code is exactly similar to code 8.

Once a collection of output data files for different values of  $D$  for each type of binary is obtained, codes 9 and 11 can be used to plot orbits and energy evolution for each file. A slightly efficient program to plot and save multiple figures at once is given in code 15.

**Results** – The orbital trajectories obtained for different values of  $D$ , for clockwise and counter-clockwise rotating binaries is shown in figure 6. Corresponding binary energy evolutions are shown in figure 7. The final values of energies of primary and secondary stars are listed in table 3.

**Discussions** – Figure 6 clearly demonstrates that clockwise rotating binaries are more stable to tidal disruption than counter-clockwise rotating binaries. This can be seen most prominently for cases  $D = 1.0 - 0.1$ . In the energy evolution plots of figure 7, these effects are most pronounced for  $D = 1.0$  and  $0.5$ .

D	Energies (counter-clockwise)	Energies (clockwise)
1.0	(-25.2675, 25.1855)	(-0.0890, -0.1091)
0.5	(-23.4682, 23.3859)	(9.6326, -9.9671)
0.1	(-16.3085, 16.2244)	(14.6806, -14.7687)
0.025	(-39.1484, 12.5554)	(-8.3239, -18.3726)
0.0125	( $7.4 \times 10^6$ , $-4.33 \times 10^2$ )	( $6.85 \times 10^5$ , $-4.84 \times 10^2$ )
0.01	( $2.79 \times 10^6$ , $1.65 \times 10^5$ )	( $7.36 \times 10^7$ , $3.52 \times 10^6$ )

**Table 3:** Final energies of two stars (primary, secondary) in the simulations of binaries for different values of penetration factor,  $D$ .

```

1 clear; clf;
2
3 cd retrograde;
4 files = dir('*0*');
5 i=1;
6 % tiledlayout(6,1,'Padding','tight')
7 for file = files'
8     out = load(file.name);
9
10    t=out(:,1); % time
11    xp=out(:,2); % the position of the primary star
12    yp=out(:,3);
13    xs=out(:,4); % the position of the secondary star
14    ys=out(:,5);
15    Ep=out(:,6); % the energy of the primary star
16    Es=out(:,7); % the energy of the secondary star
17
18    plot(xp,yp,'DisplayName','Primary');
19    hold on;
20    plot(xs,ys,'DisplayName','Secondary');
21    hold on;
22    plot(0,0,'o','MarkerSize',5.5, 'MarkerFaceColor', ...
23         'black','DisplayName','Black hole')
24    axis equal;
25    xlabel('x','FontSize',15);
26    ylabel('y','FontSize',15);
27    legend;
28
29    % plot(t,Ep,'DisplayName','Primary');
30    % hold on;
31    % plot(t,Es,'DisplayName','Secondary');
32    % xlabel('time','FontSize',15);
33    % ylabel('Energy','FontSize',15);
34    % legend('location','best');
35
36    title(strcat('D = ',file.name))
37    hold off
38    exportgraphics(gcf,strcat('../plots/3f/retrograde_energies/', ...
39                             string(i),'.png'),'Resolution',300)
40    figure;
41    i=i+1
42 end
43 cd ..

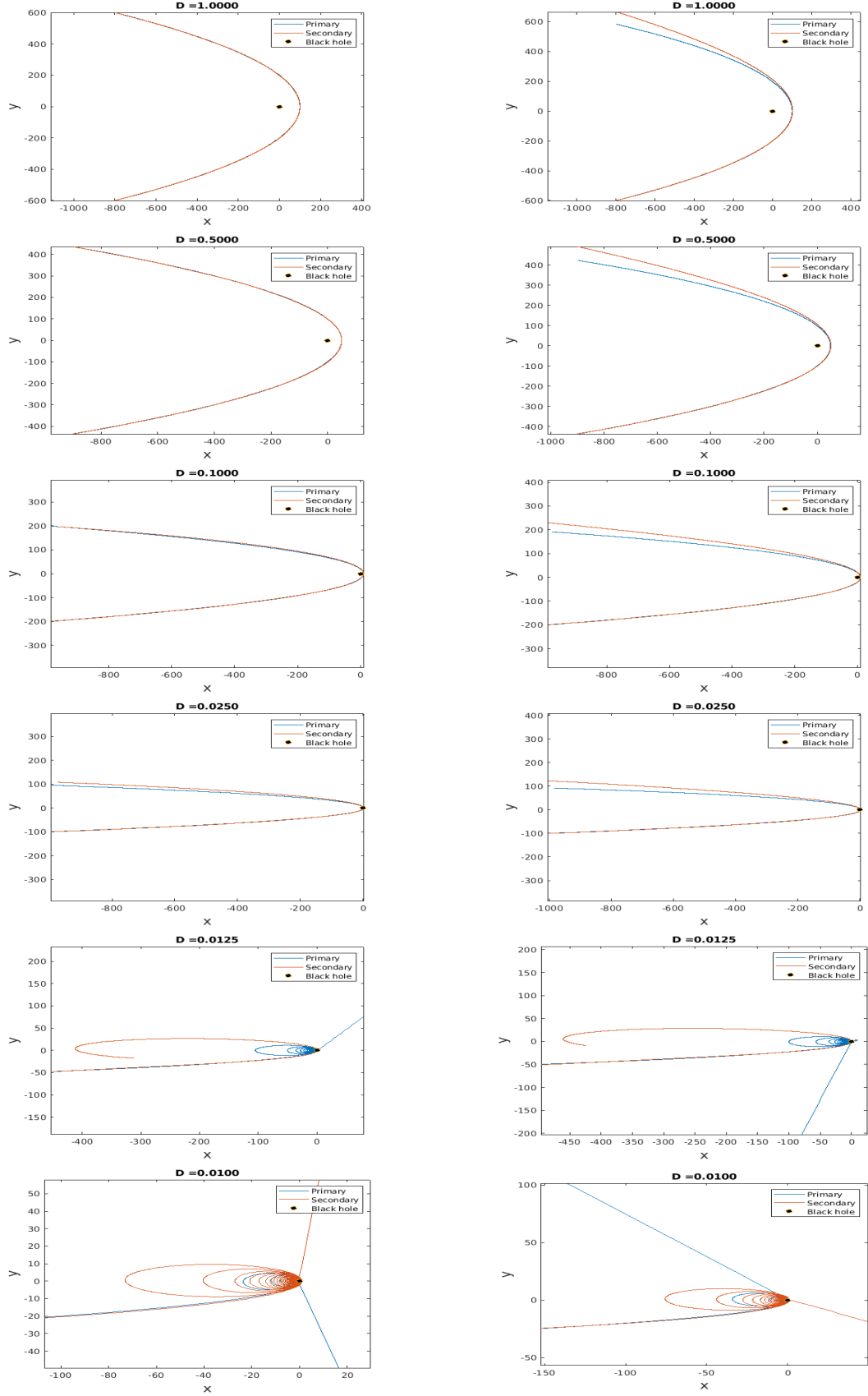
```

**Code 15:** script4.m - Program to collectively plot orbital trajectories for multiple data files in a directory. The commented lines 29–34 can be used to plot binary energy evolution instead.

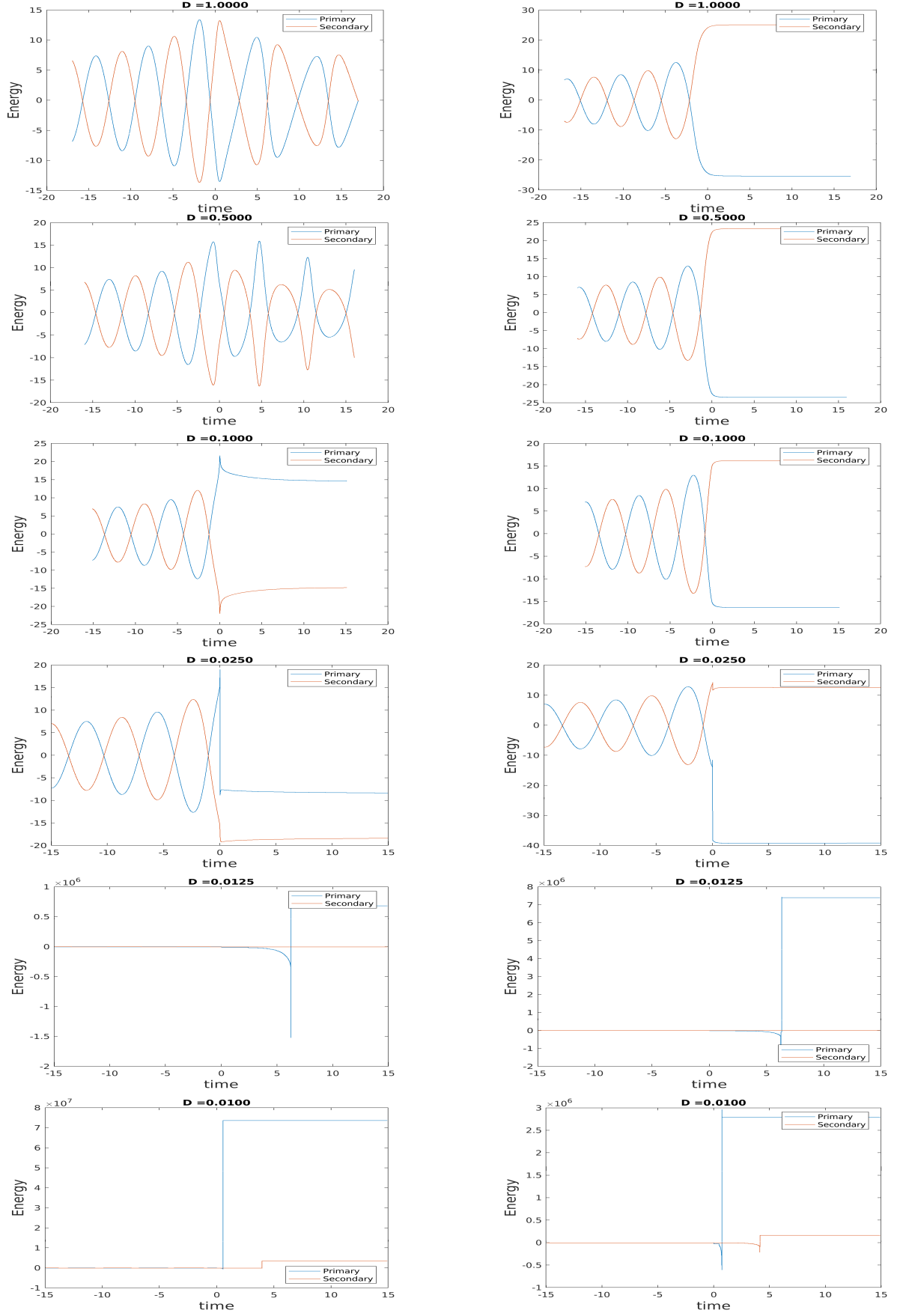
The extreme case tidal disruptions are essentially similar for both cases with both stars getting ejected, as seen for  $D = 0.01$ . One another possibility for such close interactions is the complete disruption and *swallow up* of the binary by the super massive black hole. On the other hand, in somewhat less extreme cases, such as for  $D = 0.0125$ , one of the stars is captured in highly eccentric orbit while the other one is ejected at extremely high speed.

For the case of  $D = 0.1$  considered in previous section in equation 54, if we consider the final energy obtained in table 3, the value of velocity at  $r = \infty$  for clockwise rotating binary turns out to approximately 3998 km/s instead of approximately 4200 km/s obtained for counter-clockwise rotating binary.

From these observations, it can be safely concluded that orientation of rotation of binary has a profound impact on evolution of the system in the vicinity of black hole.



**Figure 6:** A comparison of trajectories of clockwise (left) and counter-clockwise (right) binaries for different values of penetration factor  $D = 1.0, 0.5, 0.1, 0.025, 0.0125$  and  $0.01$ , for the cases of tidal breakups of binaries. Some plots are zoomed in for better visualization.



**Figure 7:** A comparison of evolution of energies of clockwise (left) and counter-clockwise (right) binaries for different values of penetration factor  $D = 1.0, 0.5, 0.1, 0.025, 0.0125$  and  $0.01$ , for the cases of tidal breakups of binaries. These energies correspond to orbits shown in figure 6.

### 3 Conclusions

Binary star systems around super massive black holes get tidally disrupted in case the distance of closest approach is less than some threshold value. If the distance of closest approach is less than that value, one star of the binary gets more strongly bound in a tighter lower energy orbit while the other one flies off in a higher energy trajectory. This phenomenon results in occurrence of hypervelocity stars. In extreme cases of very close approaches, the stars may get tidally destroyed by the black hole's gravity, thereby becoming sources of observable transient tidal disruption flares.

# Appendices

## A Complete Matlab Programs

### A.1 Problem 1

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

problaa.m

```
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
```

problab.m

```
1 clear; % clear variables and functions
2 tic; % start clock
3 d = 0.01;
4 np = 5.e5; % number of particles
5 tstep = 2000
6
7 for i = 1:np
8     x = 0; % initial x at origin
9     y = 0; % initial y at origin
10    for j = 1:tstep
11        theta = 2*pi*rand(); % random theta between zero and 2pi
12        x = x+d*cos(theta); % next value of x
13        y = y+d*sin(theta); % next value of y
14    end
15    xfinal(i) = x;
16    yfinal(i) = y;
17    r(i) = sqrt(x^2+y^2);
18 end
19
20 dr = 0.05; % bin width
21 binedges = 0:dr:max(r)+dr; % bin edges. Starts at zero,
22 % step size of binwidth,
23 % ends at ceil of maximum value of r.
24 histogram(r,binedges);
```



```

25 grid on;
26 xlabel('Final value of r','FontSize',14)
27 ylabel('Counts','FontSize',14)
28 toc % stop clock

```

problac.m

```

1 clear; % clear variables and functions
2 tic; % start clock
3 d = 0.01; % jump size at each timestep
4 np = 1.e5; % number of particles
5 tstep = 2000; % number of timesteps for each particle
6
7 for i = 1:np % for each particle
8     x = 0; % initial x at origin
9     y = 0; % initial y at origin
10    for j = 1:tstep % for each timestep
11        theta = 2*pi*rand(); % random theta between zero and 2pi
12        x = x+d*cos(theta); % next value of x
13        y = y+d*sin(theta); % next value of y
14    end
15    xfinal(i) = x;
16    yfinal(i) = y;
17    r(i) = sqrt(x^2+y^2);
18 end
19
20 dr = 0.05; % bin width
21 binedges = 0:dr:max(r)+dr; % bin edges. Starts at zero,
22 % step size of binwidth,
23 % ends at ceil of maximum value of r.
24 histogram(r,binedges);
25 grid on;
26 xlabel('Final value of r','FontSize',14)
27 ylabel('Counts','FontSize',14)
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);
36 toc % stop clock

```

problad.m

## A.2 Problem 2

```

1 % this is the main code to estimate the binary evolution
2 % two functions: energy.m and RK4.m are used in this code.
3 % See section 24, Octave tutorial for Function m-files
4 % Define mp, ms, x(1)-x(8) and tmax
5
6 clear;clf
7 h = 1.d-2; % time-step size
8 Ns = 1000 ; % sampling
9 mp = 0.8 ; % primary mass
10 ms = 0.2 ; % secondary mass
11 x(1) = -ms ; % primary x
12 x(2) = 0 ; % primary y
13 x(3) = 0 ; % primary vx
14 x(4) = -ms ; % primary vy
15 x(5) = mp ; % secondary x
16 x(6) = 0 ; % secondary y
17 x(7) = 0 ; % secondary vx
18 x(8) = mp ; % secondary vy
19 tmax = 10*pi; % final time
20 %-----
21 t = 0 ; % initial time
22 tprint = t;

```

```

23 dtp      = (tmax-t)/Ns;
24 while t < tmax
25     if t >= tprint
26         E = energy(x,mp,ms);          % Estimati binary energy
27         v = [t x(1) x(2) x(5) x(6) E];
28         save out v -ascii -append % save the v vector in the file out
29         tprint = tprint + dtp;
30     end
31     x = RK4(h,t,x,mp,ms); % Runge-Kutta method
32     t = t+h;
33 end

```

binary.m

```

1 function dxdt = f(t,x,mp,ms)
2 % The vector F. see equatiion 29
3 % input variables: t, x, mp, ms
4 % output variables: dxdt
5 r = sqrt((x(1)-x(5))^2+(x(2)-x(6))^2);
6 dxdt(1) = x(3);
7 dxdt(2) = x(4);
8 dxdt(3) = ms*(x(5)-x(1))/r^3;
9 dxdt(4) = ms*(x(6)-x(2))/r^3;
10 dxdt(5) = x(7);
11 dxdt(6) = x(8);
12 dxdt(7) = mp*(x(1)-x(5))/r^3;
13 dxdt(8) = mp*(x(2)-x(6))/r^3;

```

f.m

```

1 function xout=RK4(h,t,x,mp,ms)
2 % Runge-Kutta method
3 % input variables: h,t,x,mp,ms
4 % output variables: xout
5 % function f.m is used
6 k1 = h*f(t,x, mp,ms);
7 k2 = h*f(t+h/2,x+k1/2,mp,ms);
8 k3 = h*f(t+h/2,x+k2/2,mp,ms);
9 k4 = h*f(t+h,x+k3, mp,ms);
10 xout = x+k1/6+k2/3+k3/3+k4/6;

```

RK4.m

```

1 function Etot=energy(x,mp,ms)
2 % binary energy
3 % input variables: x, mp, ms
4 % output variables: Eot
5 vp2 = x(3)^2+x(4)^2 ; % the square of the primary star velocity
6 vs2 = x(7)^2+x(8)^2 ; % the square of the secondary star velocity
7 r = sqrt((x(5)-x(1))^2+(x(6)-x(2))^2); % binary separation
8 Etot= (mp/2)*vp2+(ms/2)*vs2-mp*ms/r; % binary energy: kinetic + potential

```

energy.m

```

1 % plot the orbits of the two stars
2 clear; clf;
3 load out % load the data file out
4 xp=out(:,2); % the primary star
5 yp=out(:,3);
6 xs=out(:,4); % the secondary star
7 ys=out(:,5);
8
9 plot(xp,yp,'LineWidth',2,'DisplayName','Primary') % plot the orbit of the primary star
10 hold on % hold ON holds the current plot and all axis properties
11 % so that subsequent graphing commands add to the existing
12 % graph without resetting the color and linestyle.
13 plot(xs,ys,'r:', 'LineWidth',2,'DisplayName','Secondary') % plot the orbit of the secondary star

```

```

14 xlabel('x','FontSize',12)
15 ylabel('y','FontSize',12)
16 ylim([-0.8 0.8]) % plotting range from x=-0.8 to 0.8.
17 set(gca, 'FontSize', 12)
18 axis equal
19 legend;

```

orbitplot.m

```

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

```

energyplot.m

## A.3 Problem 3

### A.3.1 Problems a–e

```

1 % The disruption of a binary by a massive BH
2 % the main code
3 % define mb, mp, ms. These are dimensionless masses.
4
5 clear % clear all variables and functions from memory
6 D = 0.1; % penetration factor: Rp/Rtidal
7 h = 1.d-3; % time-step size
8 Ns = 10000; % sampling
9 mb = 1.e6 ; % BH mass in units of the binary mass m
10 mp = 0.8 ; % primary star mass in units of m
11 ms = 0.2 ; % secondary star mass in units of m
12 [t, x] = initialc(D,mb,mp,ms); % Define the initial values of t and x.
13 disp(t);
14 disp(x);
15 tmax = -t;
16 tprint = t;
17 dtp = (tmax-t)/Ns;
18 while t < tmax
19     if t >= tprint
20         [Ep,Es] = energy(x,mb,mp,ms);
21         v = [t x(1) x(2) x(5) x(6) Ep Es];
22         save out v -ascii -append
23         tprint = tprint + dtp;
24     end
25     x = RK4(h,t,x,mb,mp,ms);
26     t = t+h;
27 end

```

HVS.m

```

1 function [t, x] = initialc(D,mb,mp,ms)
2 % input variables: D,mb,mp,ms. D=Rp/Rtidal is the penetration factor.
3 % out variables: t and x. Note that x is a vector.
4 % Define the initial values of t and x as functions of the input variables.
5 Rt      = (mb)^(1/3) ; % tidal radius
6 R0      = 10 * Rt    ; % initial distance between BH and binary
7 f0      = -acos(-1+(D/5)) ; % initial true anomaly (eq 44)
8 Rdot    = sin(f0) * mb^(1/3) / (sqrt(2*D)) ; % dR/dt (eq 49a)
9 Fdot    = (1+cos(f0))^2 * sqrt(2) / ...
10         (4*D^(3/2)) ; % df/dt (eq 49b)
11
12 xcmxdot = Rdot*cos(f0) - R0*Fdot*sin(f0) ; % d(xcmx)/dt (using eq 41a)
13 xcmydot = Rdot*sin(f0) + R0*Fdot*cos(f0) ; % d(xcmy)/dt (using eq 41b)
14
15 phi     = pi/2 ; % binary phase
16 rpxdot  = -ms*sin(phi+pi) ; % d(rpx)/dt (using problem 2)
17 rpydot  = ms*cos(phi+pi) ; % d(rpy)/dt (using problem 2)
18 rsxdot  = -mp*sin(phi) ; % d(rsx)/dt (using problem 2)
19 rsydot  = mp*cos(phi) ; % d(rsy)/dt (using problem 2)
20
21 t       = (sqrt(2)/3) * (D^(3/2)) * ...
22         (tan(f0/2))*(3+(tan(f0/2))^2) ; % initial time t0
23 x(1)    = (R0*cos(f0)) + (ms*cos(phi+pi)) ; % x_p
24 x(2)    = (R0*sin(f0)) + (ms*sin(phi+pi)) ; % y_p
25 x(3)    = xcmxdot+rpxdot ; % v_xp
26 x(4)    = xcmydot+rpydot ; % v_yp
27 x(5)    = (R0*cos(f0)) + (mp*cos(phi)) ; % x_s
28 x(6)    = (R0*sin(f0)) + (mp*sin(phi)) ; % y_s
29 x(7)    = xcmxdot+rsxdot ; % v_xs
30 x(8)    = xcmydot+rsydot ; % v_ys

```

initialc.m

```

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

```

f.m

```

1 function [Ep,Es]=energy(x,mb,mp,ms)
2 % the energies of the two stars
3 % input variables: x, mb, mp, ms. x is a vector variable.
4 % out variables: Ep, Es
5 rp = sqrt(x(1)^2+x(2)^2); % BH-primary separation
6 vp2 = x(3)^2+x(4)^2 ; % squire of primary velocity
7 rs = sqrt(x(5)^2+x(6)^2); % BH-secondary
8 vs2 = x(7)^2+x(8)^2 ; % squire of secondary velocity
9 rps = sqrt((x(5)-x(1))^2+(x(6)-x(2))^2); % primary-secondary
10 Ep = (mp*vp2/2)-(mb*mp/rp)-(mp*ms/rps); % primary energy
11 Es = (ms*vs2/2)-(mb*ms/rs)-(mp*ms/rps); % secondary energy

```

energy.m

```

1 function xout=RK4(h,t,x,mb,mp,ms)
2 % Runge-Kutta method
3 % input variables: h,t,x,mb,mp,ms
4 % output variables: xout
5 % a new input variable mb is added.
6 k1 = h*f(t,x, mb,mp,ms);
7 k2 = h*f(t+h/2,x+k1/2,mb,mp,ms);

```

```

8 k3 = h*f(t+h/2,x+k2/2,mb,mp,ms);
9 k4 = h*f(t+h,x+k3,mb,mp,ms);
10 xout = x+k1/6+k2/3+k3/3+k4/6;

```

RK4.m

```

1 clear; clf;
2 out=load('out'); % load the data file out
3 t=out(:,1); % time
4 xp=out(:,2); % the position of the primary star
5 yp=out(:,3);
6 xs=out(:,4); % the position of the secondary star
7 ys=out(:,5);
8 Ep=out(:,6); % the energy of the primary star
9 Es=out(:,7); % the energy of the secondary star
10
11
12 plot(xp,yp,'DisplayName','Primary');
13 hold on;
14 plot(xs,ys,'DisplayName','Secondary');
15 hold on;
16 plot(0,0,'o','MarkerSize',5.5, 'MarkerFaceColor','black','DisplayName','Black hole')
17 axis equal;
18 xlabel('x','FontSize',15);
19 ylabel('y','FontSize',15);
20 legend;
21
22 plot(xs-xp,ys-yp);
23 axis equal;
24 xlabel('x','FontSize',15);
25 ylabel('y','FontSize',15);
26
27 plot(t,Ep,'DisplayName','Primary');
28 hold on;
29 plot(t,Es,'DisplayName','Secondary');
30 xlabel('time','FontSize',15);
31 ylabel('Energy','FontSize',15);
32 legend('location','best');

```

script4.m

### A.3.2 Part f

```

1 % The disruption of a binary by a massive BH
2 % the main code
3 % define mb, mp, ms. These are dimensionless masses.
4 clear % clear all variables and functions from memory
5 h = 1.d-3; % time-step size
6 Ns = 10000; % sampling
7 mb = 1.e6 ; % BH mass in units of the binary mass m
8 mp = 0.8 ; % primary star mass in units of m
9 ms = 0.2 ; % secondary star mass in units of m
10 for D=[ 1.0, 0.5, 0.1, 0.025, 0.0125 0.01]
11 [t, x] = initialc(D,mb,mp,ms); % Define the initial values of t and x.
12 tmax = -t;
13 tprint = t;
14 dtp = (tmax-t)/Ns;
15 filename = sprintf('%.4f',D);
16 while t < tmax
17 if t >= tprint
18 [Ep,Es] = energy(x,mb,mp,ms);
19 v = [t x(1) x(2) x(5) x(6) Ep Es];
20 save(filename, 'v', '-ascii', '-append')
21 tprint = tprint + dtp;
22 end
23 x = RK4(h,t,x,mb,mp,ms);
24 t = t+h;
25 end
26 end

```

HVS.m

```

1 function [t, x] = initialc(D,mb,mp,ms)
2 % input variables: D,mb,mp,ms. D=Rp/Rtidal is the penetration factor.
3 % out variables: t and x. Note that x is a vector.
4 % Define the initial values of t and x as functions of the input variables.
5 Rt      = (mb)^(1/3) ; % tidal radius
6 R0      = 10 * Rt    ; % initial distance between BH and binary
7 f0      = -acos(-1+(D/5)) ; % initial true anomaly (eq 44)
8 Rdot    = sin(f0) * mb^(1/3) / (sqrt(2*D)) ; % dR/dt (eq 49a)
9 Fdot    = (1+cos(f0))^2 * sqrt(2) / ...
10         (4*D^(3/2)) ; % df/dt (eq 49b)
11
12 xcmxdot = Rdot*cos(f0) - R0*Fdot*sin(f0) ; % d(xcmx)/dt (using eq 41a)
13 xcmydot = Rdot*sin(f0) + R0*Fdot*cos(f0) ; % d(xcmy)/dt (using eq 41b)
14
15 phi     = pi/2 ; % binary phase
16 rpxdot  = -ms*sin(phi+pi)
17 rpydot  = ms*cos(phi+pi)
18 rsxdot  = -mp*sin(phi)
19 rsydot  = mp*cos(phi)
20
21 t       = (sqrt(2)/3) * (D^(3/2)) * ...
22         (tan(f0/2))*(3+(tan(f0/2))^2) ; % initial time t0
23 x(1)    = (R0*cos(f0)) + (ms*cos(phi+pi)) ; % x_p
24 x(2)    = (R0*sin(f0)) + (ms*sin(phi+pi)) ; % y_p
25 x(3)    = xcmxdot+rpxdot ; % v_xp
26 x(4)    = xcmydot+rpydot ; % v_yp
27 x(5)    = (R0*cos(f0)) + (mp*cos(phi)) ; % x_s
28 x(6)    = (R0*sin(f0)) + (mp*sin(phi)) ; % y_s
29 x(7)    = xcmxdot+rsxdot ; % v_xs
30 x(8)    = xcmydot+rsydot ; % v_ys

```

initialc\_prograde.m

```

1 function [t, x] = initialc(D,mb,mp,ms)
2 % input variables: D,mb,mp,ms. D=Rp/Rtidal is the penetration factor.
3 % out variables: t and x. Note that x is a vector.
4 % Define the initial values of t and x as functions of the input variables.
5 Rt      = (mb)^(1/3) ; % tidal radius
6 R0      = 10 * Rt    ; % initial distance between BH and binary
7 f0      = -acos(-1+(D/5)) ; % initial true anomaly (eq 44)
8 Rdot    = sin(f0) * mb^(1/3) / (sqrt(2*D)) ; % dR/dt (eq 49a)
9 Fdot    = (1+cos(f0))^2 * sqrt(2) / ...
10         (4*D^(3/2)) ; % df/dt (eq 49b)
11
12 xcmxdot = Rdot*cos(f0) - R0*Fdot*sin(f0) ; % d(xcmx)/dt (using eq 41a)
13 xcmydot = Rdot*sin(f0) + R0*Fdot*cos(f0) ; % d(xcmy)/dt (using eq 41b)
14
15 phi     = pi/2 ; % binary phase
16 rpxdot  = ms*sin(phi+pi)
17 rpydot  = -ms*cos(phi+pi)
18 rsxdot  = mp*sin(phi)
19 rsydot  = -mp*cos(phi)
20
21 t       = (sqrt(2)/3) * (D^(3/2)) * ...
22         (tan(f0/2))*(3+(tan(f0/2))^2) ; % initial time t0
23 x(1)    = (R0*cos(f0)) + (ms*cos(phi+pi)) ; % x_p
24 x(2)    = (R0*sin(f0)) + (ms*sin(phi+pi)) ; % y_p
25 x(3)    = xcmxdot+rpxdot ; % v_xp
26 x(4)    = xcmydot+rpydot ; % v_yp
27 x(5)    = (R0*cos(f0)) + (mp*cos(phi)) ; % x_s
28 x(6)    = (R0*sin(f0)) + (mp*sin(phi)) ; % y_s
29 x(7)    = xcmxdot+rsxdot ; % v_xs
30 x(8)    = xcmydot+rsydot ; % v_ys

```

initialc\_retrograde.m

```

1 function dxdt = f(t,x,mb,mp,ms)
2 r = sqrt((x(1)-x(5))^2+(x(2)-x(6))^2) ;
3 rp = sqrt(x(1)^2 + x(2)^2) ;
4 rs = sqrt(x(5)^2 + x(6)^2) ;

```

```

5 dxdt(1) = x(3) ; % v_px
6 dxdt(2) = x(4) ; % v_py
7 dxdt(3) = (ms*(x(5)-x(1))/r^3)+mb*(-x(1)/rp^3) ; % a_px
8 dxdt(4) = (ms*(x(6)-x(2))/r^3)+mb*(-x(2)/rp^3) ; % a_py
9 dxdt(5) = x(7) ; % v_sx
10 dxdt(6) = x(8) ; % v_sy
11 dxdt(7) = (mp*(x(1)-x(5))/r^3)+mb*(-x(5)/rs^3) ; % a_sx
12 dxdt(8) = (mp*(x(2)-x(6))/r^3)+mb*(-x(6)/rs^3) ; % a_sy

```

f.m

```

1 function [Ep,Es]=energy(x,mb,mp,ms)
2 % the energies of the two stars
3 % input variables: x, mb, mp, ms. x is a vector variable.
4 % out variables: Ep, Es
5 rp = sqrt(x(1)^2+x(2)^2); % BH-primary separation
6 vp2 = x(3)^2+x(4)^2 ; % squire of primary velocity
7 rs = sqrt(x(5)^2+x(6)^2); % BH-secondary
8 vs2 = x(7)^2+x(8)^2 ; % squire of secondary velocity
9 rps = sqrt((x(5)-x(1))^2+(x(6)-x(2))^2); % primary-secondary
10 Ep = (mp*vp2/2)-(mb*mp/rp)-(mp*ms/rps); % primary energy
11 Es = (ms*vs2/2)-(mb*ms/rs)-(mp*ms/rps); % secondary energy

```

energy.m

```

1 function xout=RK4(h,t,x,mb,mp,ms)
2 % Runge-Kutta method
3 % input variables: h,t,x,mb,mp,ms
4 % output variables: xout
5 % a new input variable mb is added.
6 k1 = h*f(t,x, mb,mp,ms);
7 k2 = h*f(t+h/2,x+k1/2,mb,mp,ms);
8 k3 = h*f(t+h/2,x+k2/2,mb,mp,ms);
9 k4 = h*f(t+h,x+k3, mb,mp,ms);
10 xout = x+k1/6+k2/3+k3/3+k4/6;

```

RK4.m

```

1 clear; clf;
2
3 cd retrograde;
4 files = dir('*0*');
5 i=1;
6 % tiledlayout(6,1,'Padding','tight')
7 for file = files'
8     out = load(file.name);
9
10     t=out(:,1); % time
11     xp=out(:,2); % the position of the primary star
12     yp=out(:,3);
13     xs=out(:,4); % the position of the secondary star
14     ys=out(:,5);
15     Ep=out(:,6); % the energy of the primary star
16     Es=out(:,7); % the energy of the secondary star
17
18     plot(xp,yp,'DisplayName','Primary');
19     hold on;
20     plot(xs,ys,'DisplayName','Secondary');
21     hold on;
22     plot(0,0,'o','MarkerSize',5.5, 'MarkerFaceColor', ...
23         'black','DisplayName','Black hole')
24     axis equal;
25     xlabel('x','FontSize',15);
26     ylabel('y','FontSize',15);
27     legend;
28
29     % plot(t,Ep,'DisplayName','Primary');
30     % hold on;
31     % plot(t,Es,'DisplayName','Secondary');

```

```

32 % xlabel('time','FontSize',15);
33 % ylabel('Energy','FontSize',15);
34 % legend('location','best');
35
36 title(strcat('D = ',file.name))
37 hold off
38 exportgraphics(gcf,strcat('../plots/3f/retrograde_energies/', ...
39                          string(i),'.png'),'Resolution',300)
40 figure;
41 i=i+1
42 end
43 cd ..

```

script4.m