Computational Astrophysics Project 3

Howard Kinsman

May 6, 2016

1 Problem 1B: Heat Conduction

1.1 Part a

In order to solve this problem it is first necessary to set the initial and boundary conditions. Variables A, B and C were defined according to equation 15 in the notes. Variable A was set to μ because of μu_{j-1}^n in equation 15. In Octave this is simply:

A = mu;

Variable B was set to:

$$B = 1 - (2*mu*(1+(1/(j-1))));$$

This is in order to satisfy $1 - 2\mu(1 + \frac{1}{j-1})u_j^n$ in equation 15. Finally C was set to:

$$C = mu*(1+(2/(j-1)));$$

This satisfies the last part of equation 15: $\mu(1+\frac{2}{j-1})\mu_{j+1}^n$.

The boundary condition at the surface was set to zero:

$$unew(N) = 0;$$

The boundary condition at the centre was set to the value above it on each iteration i.e. the boundary condition at the centre cannot be calculated so it is just assumed to be the same as the value above it. In Octave this was coded as:

$$unew(1) = unew(2);$$

The problem required the distributions to be between $\tau = 0$ and $\tau = 0.2$ so I modified the total time steps to be 2400 because $\Delta_x = 1/(N-1) = \frac{1}{49}$ and so $\Delta_\tau = \mu \Delta_x^2 = \frac{1}{12005}$ which leads to $n = \frac{\tau_n}{\Delta_\tau} + 1$ and so n = 2402.

The plot of the temperature distribution is shown in Fig. 1. Each line represents a timestep of 100. At time 1 the temperature gradient is almost vertical as there has been no time for cooling but at a time of 100 the temperature gradient forms a steep curve. The curve gets gradually less steep as time goes on until at time 2400 the curve has flattened considerably and the temperature at the centre has cooled to approximately 0.3.

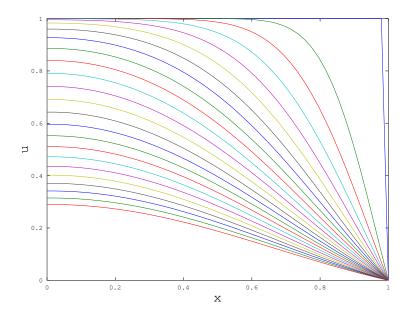


Figure 1: Plot of temperature distribution

1.2 Part b

The following lines of code in Octave retrieve the temperature when it has cooled to exactly half its initial value. The first line retrieves all values at position 2 i.e. the centre (the actual centre is position 1 but this was just set to be the same as position 2 in the boundary conditions). The second line finds the smallest timestep value (n) where temperature (u) is less than 1/2.

```
u1 = approx(2,:);

nc = min(find(u1 < 0.5))
```

The timestep value n retrieved from the above has a value of 1629. $\tau_n = (n-1)\Delta_{\tau} = \frac{1628}{12005}$. From the notes τ is in units of $t/(a^2/\eta)$. Therefore time in seconds= τ (a^2/η) where a=100,000m and $\eta=10^{-6}m^2sec$, giving the time it takes the centre to cool to half its initial temperature as 4.3×10^7 years.

Full source code listing for Problem 1B is in Appendix 1.

2 Problem 2: Circular Binary

2.1 Part a

For a binary system where the centre of mass is at the origin then: $m_1\mathbf{r_1} = m_2\mathbf{r_2}$ and we can express the stars positions using just their mass. The more massive star would lie closer to the centre of mass at a radius that is inversely proportional to its mass. The equation of motion (for the primary star) is given by:

$$m_p \frac{d^2 \mathbf{x_p}}{dt^2} = \frac{G m_p m_s}{|\mathbf{x_s} - \mathbf{x_p}|^3} (\mathbf{x_s} - \mathbf{x_p})$$
(1)

If we substitute $\mathbf{m_p} = m\tilde{\mathbf{m_p}}$, $\mathbf{x_p} = a\tilde{\mathbf{x_p}}$, $\mathbf{x_s} = a\tilde{\mathbf{x_s}}$ and $t = \tilde{t}\sqrt{a^3/Gm}$ into equation 1 then this leads to:

$$m\tilde{\mathbf{m}}_{\mathbf{p}} \frac{d^2 a \tilde{\mathbf{x}}_{\mathbf{p}}}{d \left(\tilde{t} \sqrt{a^3 / Gm} \right)^2} = \frac{G m \tilde{\mathbf{m}}_{\mathbf{p}} m \tilde{\mathbf{m}}_{\mathbf{s}}}{|a \tilde{\mathbf{x}}_{\mathbf{s}} - a \tilde{\mathbf{x}}_{\mathbf{p}}|^3} (a \tilde{\mathbf{x}}_{\mathbf{s}} - a \tilde{\mathbf{x}}_{\mathbf{p}})$$
(2)

Rearranging, we get:

$$\frac{a\tilde{\mathbf{m}}_{\mathbf{p}}Gm^2d^2\tilde{\mathbf{x}}_{\mathbf{p}}}{a^3d\tilde{t}^2} = \frac{Gm^2\tilde{\mathbf{m}}_{\mathbf{p}}\tilde{\mathbf{m}}_{\mathbf{s}}a}{|a\tilde{\mathbf{x}}_{\mathbf{s}} - a\tilde{\mathbf{x}}_{\mathbf{p}}|^3}(\tilde{\mathbf{x}}_{\mathbf{s}} - \tilde{\mathbf{x}}_{\mathbf{p}})$$
(3)

The $\frac{a\tilde{\mathbf{m}}_{\mathbf{p}}Gm^2}{a^3}$ cancel from both sides leaving:

$$\frac{d^2 \tilde{\mathbf{x}_p}}{d\tilde{t}^2} = \frac{\tilde{\mathbf{m}_s} a}{|\tilde{\mathbf{x}_s} - \tilde{\mathbf{x}_p}|^3} (\tilde{\mathbf{x}_s} - \tilde{\mathbf{x}_p})$$
(4)

The primary star is positioned on the negative side of the x axis, hence $m_s cos(\pi)$, whilst the secondary star is on the positive side, hence $m_p cos(0)$. They both sit on the x axis initially so y=0. Both of their velocities are zero in the x direction i.e. $-m_s sin(\pi)$ and $-m_p sin(0)$ respectively. The stars initial velocities are the time derivatives of the initial positions. The primary star has negative velocity i.e. downwards given by $m_s cos(\pi)$ whilst the secondary star has a positive velocity: $m_p cos(0)$, so the orbit is anticlockwise. The initial conditions for this problem are given below. These are assuming an initial phase of $\phi=0$.

$$\mathbf{r_p} = (m_s cos(\pi), m_s sin(\pi)) \tag{5}$$

$$\mathbf{r_s} = (m_p cos(0), m_p sin(0)) \tag{6}$$

$$\mathbf{\dot{r_p}} = (-m_s sin(\pi), m_s cos(\pi)) \tag{7}$$

$$\mathbf{\dot{r}_s} = (-m_p sin(0), m_p cos(0)) \tag{8}$$

The orbital period is given by:

$$T = 2\pi \sqrt{\frac{a^3}{G\left(M_1 + M_2\right)}}\tag{9}$$

and so in dimensionless units of $\sqrt{(a^3/Gm)}$ the period is simply 2π .

2.2 Parts b + c

Assuming that $\frac{m_p}{m_s} = 3$, then $m_p = 0.75$ and $m_s = 0.25$. I set the final time to be 6π to allow the stars to orbit three times. In Octave the intial conditions were set as follows:

```
% primary mass
        .75 ;
mp
         .25 ; % secondary mass
ms
     = ms*cos(pi);
                     % primary x
x(1)
x(2)
     = ms*sin(pi);
                     % primary y
     = -ms*sin(pi); % primary vx
x(3)
                     % primary vy
x(4)
     = ms*cos(pi);
x(5)
     = mp*cos(0);
                    % secondary x
                     % secondary y
     = mp*sin(0);
x(6)
     = -mp*sin(0); \% secondary vx
x(7)
     = mp*cos(0); \% secondary vy
x(8)
         6*pi; % final time
tmax
```

The full source code listing for Problem 2 is in Appendix 2.

The plot of the two stars is shown in Fig. 2. As can be seen the orbits are circular with the more massive primary star (in blue) orbiting closer to the centre of mass. The less massive secondary star (in red) lies much further out.

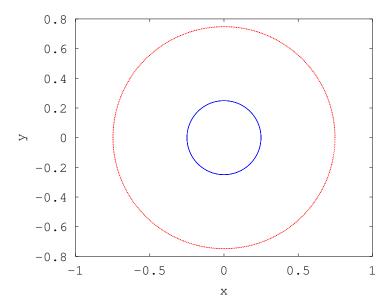


Figure 2: Plot of circular binary system

2.3 Part d

The plot of energy in the system is shown in Fig. 3. The plot shows that the derived energy in the system (blue circles) and the exact energy (red line) match. The total energy in the system is constant and negative. The exact energy in the system was calculated using $E_{exact} = \frac{-m_p m_s}{2}$. This is derived as below, for total energy in orbit (potential and kinetic):

$$E = -\frac{Gm_1m_2}{r} + \frac{1}{2}mv^2 \tag{10}$$

Eliminate v by equating it with gravity:

$$E = -\frac{Gm_1m_2}{r} + \frac{1}{2}G\frac{m_1m_2}{r} \tag{11}$$

Kinetic energy therefore is -1/2 of potential energy, so this simplifies to:

$$E = -G\frac{m_1 m_2}{2r} \tag{12}$$

 $E_{exact} = \frac{-m_p m_s}{2}$ is in dimensionless units of Gm^2/a .

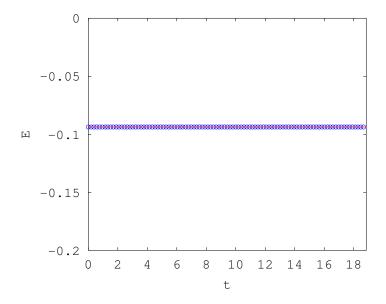


Figure 3: Plot of energy in binary system

3 Problem 3: Hypervelocity Stars

3.1 Part a

In order to plot the stellar orbits in the black hole rest frame it is first necessary to set the initial conditions. As the binary system is moving in a parabola the equations of motion for the centre of mass of parabola are:

$$X_{cm} = (R\cos(f), R\sin(f)), \qquad R = \frac{2R_p}{1 + \cos(f)}$$
(13)

This leads to the following equation for the initial angle given that the initial radius is $10R_{tidal}$ and $D = R_p/R_{tidal} = 3$:

$$f_0 = -\cos^{-1}\left(-1 + \frac{D}{5}\right) \tag{14}$$

because $\frac{2R_p}{10R_{tidal}} = 1 + cos(f)$. This was encoded into Octave as:

$$f = -a\cos(-1+D/5);$$

Initial time was set according to the following equation:

$$t_0 = \frac{\sqrt{2}}{3} D^{3/2} tan(f_0/2)(3 + tan^2(f_0/2))$$
 (15)

In Octave this was set as follows:

$$t = (sqrt(2)/3)*(D^{(3/2)})*tan(f/2)*(3+(tan(f/2))^2);$$

The initial radius is equal to $10R_{tidal}$ and $R_{tidal} = (M/m)^{1/3}a$ so in units of a, R was calculated as follows in Octave:

$$Rt=mb^{(1/3)};$$

$$R=Rt*10;$$

The initial positions of the stars relative to the black hole are given by the position of the centre of mass plus the positions of the stars as they orbit each other in circles. The initial position of the centre of mass was calculated as follows:

The initial positions of the stars were calculated in a similar fashion to Problem 2, however a phase of pi/2 was used. In order to calculate the initial velocities of the stars we first need the initial velocity of the centre of mass.

This was achieved by taking the derivatives of equation 13 and by using the derivative of the true anomaly, the angle f:

$$\dot{f} = \frac{\sqrt{2}}{4}D^{-3/2}(1 + \cos(f))^2 \tag{16}$$

The above equation is derived from:

$$\frac{df}{dt} = \sqrt{\frac{GM}{R_p^3}} \frac{\sqrt{2}}{4} (1 + \cos(f))^2 \tag{17}$$

Here the $\frac{GM}{R_p^3}=D^{-3/2}$ in dimensionless units. We note that $M=R_{tidal}^3$. So therefore it follows that:

$$\sqrt{\frac{GM}{R_p^3}} = \sqrt{\frac{GR_{tidal}^3}{R_p^3}} \tag{18}$$

and so, dropping the G for dimensionless units, we have:

$$\sqrt{\frac{R_{tidal}^3}{r_p^3}} = \left(\frac{R_p}{R_{tidal}}\right)^{-3/2} = D^{-3/2} \tag{19}$$

The derivative of equation 13 is as follows:

$$\dot{R} = \frac{2R_p \dot{f} \sin(f)}{(1 + \cos(f))^2} \tag{20}$$

which gives:

$$\dot{R} = \frac{2R_p \frac{\sqrt{2}}{4} D^{-3/2} \sin(f) (1 + \cos(f))^2}{(1 + \cos(f))^2}$$
(21)

This simplies immediately to:

$$\dot{R} = \frac{R_p D^{-3/2} sin(f)}{\sqrt{2}} \tag{22}$$

This can be rewritten as:

$$\dot{R} = \frac{R_p sin(f)}{D\sqrt{2D}} \tag{23}$$

As $D = \frac{R_p}{R_{tidal}}$ and $R_{tidal} = M^{1/3}$ we finally get:

$$\dot{R} = \frac{M^{1/3}sin(f)}{\sqrt{2D}}\tag{24}$$

Substituting \dot{R} and \dot{f} into equation 13 and applying the product rule gives us the following equations for \dot{X}_x and \dot{X}_y :

$$\dot{X}_x = \dot{R}cos(f) - R\dot{f}sin(f) \tag{25}$$

$$\dot{X}_y = R\dot{f}cos(f) + sin(f)\dot{R} \tag{26}$$

This was encoded in to Octave as follows:

```
 \begin{array}{l} {\rm rdot} = & ({\rm mb}\,\hat{}\,(1/3)/(\,{\rm sqrt}\,(2*D))) * \sin{(\,f\,)}; \\ {\rm fdot} = & ({\rm sqrt}\,(2)/4) * ({\rm D}\,\hat{}\,(-3/2)) * (1 + \cos{(\,f\,)})\,\hat{}\,2; \\ {\rm xdotx} = & ({\rm rdot} * \cos{(\,f\,)}) - ({\rm R}* {\rm fdot} * \sin{(\,f\,)}); \\ {\rm xdoty} = & ({\rm R}* {\rm fdot} * \cos{(\,f\,)}) + (\sin{(\,f\,)}* {\rm rdot\,}); \end{array}
```

Finally the initial conditions are ready to be set as follows:

The full source code listing for Problem 3 is in Appendix 3.

Fig. 4 shows the plot of the stars as the approach the black hole in a parabola. With a penetration factor of $D = \frac{R_p}{R_{tidal}} = 3$ they can be seen to be orbitting each other and successfully maintain their orbits as they pass the black hole at periastron.

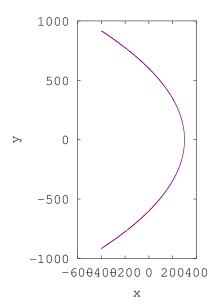


Figure 4: Plot of binary system in BH rest frame where D=3

3.2 Part b

Fig. 5 shows a plot of the same data as in Problem 1 except the plot is in the comoving frame of the secondary star. This was achieved by subtracting the position of the primary star from the position of the secondary star. The was coded into Octave as follows:

$$plot\left(\left.xs{-}xp\right.,\ ys{-}yp\right.,\ 'r:')$$

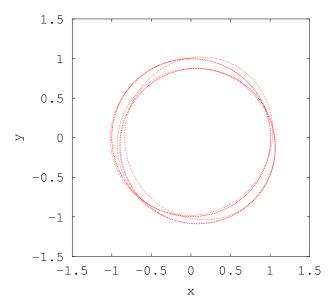


Figure 5: Plot of primary star in comoving frame of secondary star D=3

The binary system is clearly not destabilised by the black hole.

3.3 Part c

Fig. 6 is a plot of the energy of the binary system. It shows the energy is oscillating from positive to negative between the two stars as they orbit each other. The amplitude of the plot can be seen to be increasing as the stars approach the black hole reaching a maximum at periastron before starting to decrease as they move away from the influence of the black hole. The system has a total energy of zero and the energy of the one star is always the negative of the other.

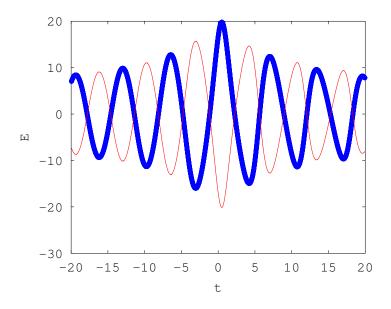


Figure 6: Plot of energy in binary system D=3

3.4 Part d

The code was then re-run but this time with $R_p = 0.1 R_{tidal}$. Fig. 7 is a plot again in the rest frame of the black hole. The binary system is clearly disrupted by the black hole. The two stars are orbitting each until the closest approach to the black hole where they separate.

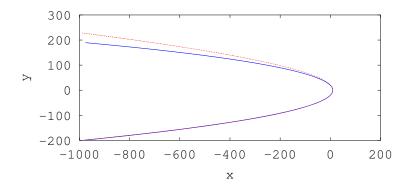


Figure 7: Plot of binary system in BH rest frame where D=0.1

A plot of the secondary star in the comoving frame of the primary star, Fig. 8, shows the secondary star to be initially in orbit with the primary star but then leaves the binary system on a separate trajectory.

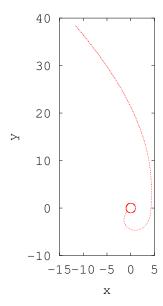


Figure 8: Plot of primary star in comoving frame of secondary star D=0.1

A plot of the energy of the system, Fig. 9, shows that the energy is

initially oscillating as in Part c but after the closest approach to the black hole the energy no longer oscillates between the two stars after the system is disrupted. The primary star (blue) ends up with a high negative energy whilst the secondary star (red) has a high positive energy.

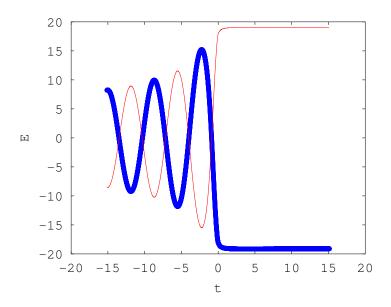


Figure 9: Plot of energy in binary system D=0.1

3.5 Part e

The velocity of the ejected star (secondary star in this case) can be calculated by using $E_k = 1/2mv^2$ so

$$v = \sqrt{2E/M} \tag{27}$$

From Fig. 9, and from the output file, the energy of the ejected star tends towards 19 as r approaches ∞ . E here is in units of Gm^2/a so, in kms units, $Gm^2/a = 5.4 \times 10^{40}$ and so $E = 10^{41}J$. From equation 27 we get 1,017,849m/s, and so the ejected star is travelling at approximately 1000km/s at $r = \infty$.

3.6 Part f

Throughout this project I had assumed that the lower mass star would always be ejected from the system, so I chose to investigate this further to see if my assumptions were correct. I ran the above simulation 8 times whilst varying the initial phase of the binary system i.e. pi/4, pi/2, 3pi/4, pi, 5pi/4, 3pi/2,

7pi/4 and 2pi. The results were surprising because in exactly half the runs the lower mass star was ejected and half the larger mass star. With an initial phase of pi/4 to pi the lower mass star was ejected, whilst from 5pi/4 to 2pi the larger mass star was ejected, see Fig. 10. This suggests that this process is independent of the mass of the star.

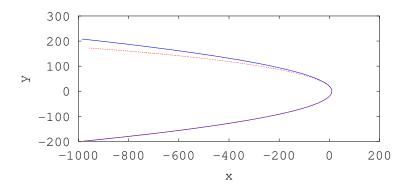


Figure 10: Plot of binary system in BH rest frame where D=0.1 and Phase=2Pi

This may be possibly due to the energy in the system. The total energy in the system is zero so the energy of the one star is always equal to the negative of the other. The star which is ejected may be the one that is in the phase with positive energy at the time of closest approach to the black hole.

Astronomical observations could confirm whether indeed we do detect similar numbers of high mass hypervelocity stars as we do low mass hypervelocity stars.

4 Appendix 1 - Problem 1B code

```
% problem1B.m
% Time evolution of temperature distribution u: spherical system
% Define A, B, C, unew(1) and unew(N)
% When you write code, it is a good practice to add comments that
% describe the code. Comments allow others to understand your code,
% and can refresh your memory when you return to it later.
% Add comments to Octave code using the percent (%) symbol.
% Comment lines can appear anywhere in a program file, and you can
% append comments to the end of a line of code. see section 26, Octave to
                   \% mu=dt/dx<sup>2</sup>
      = 0.2;
mu
                   % grid points
Ν
      = 50;
                   % total time-steps
steps = 2400;
      = ones (N,1); % initial condition: u = 1 anywhere
u(N) = 0;
                   % boundary condition at the surface
                   % empty array
approx = [];
                   % time evolution
for n=1:steps
    approx(:,n)=u; % Distribution u at each step saved in matrix: approx
                   % see section 5, Octave tutorial for the colon.
                   % Add a new column to approx matrix in each timestep.
                   % k-th column of approx gives u at timestep k.
    for j = 2:N-1
                  % see Eq (15) and define A, B, C
        A = mu;
        B = 1 - (2*mu*(1+(1/(j-1))));
        C = mu*(1+(2/(j-1)));
        unew(j)= A*u(j-1)+B*u(j)+C*u(j+1);
    end
    unew(1) = unew(2); % boundary condition at the centre
    unew(N) = 0; % boundary condition at the surface
                 % update the distribution
    u = unew;
end
                       % time evolution of temperature at the center j=1
u1 = approx(2,:);
nc = min(find(u1 < 0.5)) % critical timestep: central temp becomes < 0.5.
                       % generate N points between 0 and 1.
x = linspace(0,1,N);
plot(x, approx(:,[1:100:steps])) % plot temperture distributions
                                 \% distributions at n=1, 101, 201, 301...
```

% see section 5 and 6, Octave tutorial

xlabel('x', 'FontSize',20)
ylabel('u', 'FontSize',20)

5 Appendix 2 - Problem 2 code

```
% this is the main code to estimate the binary evolution
% two functions: energy.m and RK4.m are used in this code.
% See section 24, Octave tutorial for Function m-files
\% Define mp, ms, x(1)-x(8) and tmax
clear; clf
h
       = 1.d-2; % time-step size
       = 100 ; % sampling
Ns
%-
         .75 ;
                % primary mass
mp
         .25; % secondary mass
ms
x(1) = ms*cos(pi);
                     % primary x
     = ms*sin(pi);
                     % primary y
x(2)
     = -ms*sin(pi); % primary vx
x(3)
     = ms*cos(pi); \% primary vy
x(4)
                    % secondary x
x(5) = mp*cos(0);
x(6) = mp*sin(0); \% secondary y
    = -mp*sin(0); \% secondary vx
x(7)
    = mp*cos(0); % secondary vy
x(8)
tmax
         6*pi; % final time
%-
          0 ; % initial time
tprint = t;
       = (tmax-t)/Ns;
while t < tmax
     i f
        t >= tprint
         E = energy(x, mp, ms);
                                    % Estimat binary energy
         v = [t x(1) x(2) x(5) x(6) E];
         save out v -ascii -append % save the v vector in the file out
         tprint = tprint + dtp;
     end
     x = RK4(h,t,x,mp,ms); % Runge-Kutta method
     t = t+h;
end
function dxdt = f(t, x, mp, ms)
% The vector F. see equation 29
% input variables: t, x, mp, ms
% output variables: dxdt
```

```
r = \mathbf{sqrt}((x(1)-x(5))^2+(x(2)-x(6))^2;
  dxdt(1) = x(3);
  dxdt(2) = x(4);
  dxdt(3) = ms*(x(5)-x(1))/r^3;
  dxdt(4) = ms*(x(6)-x(2))/r^3;
  dxdt(5) = x(7);
  dxdt(6) = x(8);
  dxdt(7) = mp*(x(1)-x(5))/r^3;
  dxdt(8) = mp*(x(2)-x(6))/r^3;
function Etot=energy (x,mp,ms)
\% binary energy
% input variables: x, mp, ms
% output variables: Eot
vp2 = x(3)^2 + x(4)^2; % the square of the primary star velocity
vs2 = x(7)^2 + x(8)^2; % the square of the secondary star velocity
    = \mathbf{sqrt}((x(5)-x(1))^2+(x(6)-x(2))^2); \% \text{ binary separation}
Etot= (mp/2)*vp2+(ms/2)*vs2-mp*ms/r; % binary energy: kinetic + potential
function xout=RK4(h,t,x,mp,ms)
% Runge-Kutta method
% input variables: h,t,x,mp,ms
% output variables: xout
% function f.m is used
k1
     = h*f(t,x,
                         mp, ms);
     = h*f(t+h/2,x+k1/2,mp,ms);
k2
     = h*f(t+h/2,x+k2/2,mp,ms);
k3
     = h * f (t+h, x+k3,
k4
                        mp, ms);
xout = x+k1/6+k2/3+k3/3+k4/6;
\% plot the orbits of the two stars
clear; clf;
             % load the data file out
load out
xp=out(:,2); % the primary star
yp = out(:,3);
xs=out(:,4); % the secondary star
ys = out(:,5);
plot(xp,yp) % plot the orbit of the primary star
            % hold ON holds the current plot and all axis properties
hold on
            % so that subsequent graphing commands add to the existing
            % graph without resetting the color and linestyle.
```

```
plot(xs, ys, 'r:') % plot the orbit of the secondary star
xlabel('x', 'FontSize', 20)
ylabel('y', 'FontSize', 20)
ylim ([-0.8 \ 0.8]) % plotting range from x=-0.8 to 0.8.
set (gca, 'Fontsize', 20)
axis equal
% plot the evolution of the binary energy (it should be a constant).
% Defin mp, ms, and tmax
clear; clf;
%-----
mp = .75;
ms
     = .25;
tmax = 6*pi;
%—
load out % load the data file out
      = out(:,1); \% time
      = out(:,6); % energy
exact = -mp*ms/2; % analytic estimate
plot(t,E,'o') % numerical energy
hold on
plot([0 tmax],[exact exact],'r') % exact line
xlabel('t', 'FontSize',20)
ylabel('E', 'FontSize',20)
xlim ([0 tmax])
ylim ([-0.2 \ 0])
set (gca, 'Fontsize', 20)
```

6 Appendix 3 - Problem 3 code

```
function [t, x] = initialc(D, mb, mp, ms)
% input valiables: D,mb,mp,ms. D=Rp/Rtidal is the penetration factor.
\% out variables: t and x. Note that x is a vector.
% Define the initial values of t and x as functions of the input valiable
  phase=2*\mathbf{pi};
  f = -a\cos(-1 + D/5);
t = (\mathbf{sqrt}(2)/3)*(D^{(3/2)})*tan(f/2)*(3+(tan(f/2))^2); \% initial time
Rt=mb^(1/3);
R=Rt*10;
Xx=R*cos(f);
Xy=R*sin(f);
rdot = (mb^{(1/3)}/(sqrt(2*D))*sin(f);
fdot = (sqrt(2)/4)*(D^{(-3/2)})*(1+cos(f))^2;
xdotx = (rdot * cos(f)) - (R*fdot * sin(f));
xdoty = (R*fdot*cos(f)) + (sin(f)*rdot);
x(1) = Xx + (ms*cos(phase+pi)); \% x : primary star
x(2) = Xy + (ms * sin(phase + pi)); \% y
x(3) = xdotx + (-ms*sin(phase+pi)); \% dx/dt
x(4) = xdoty + (ms*cos(phase+pi)); \% dy/dt
x(5) = Xx + (mp*cos(phase)); \% x : secondary star
x(6) = Xy + (mp*sin(phase)); \% y
x(7) = xdotx + (-mp*sin(phase)); \% dx/dt
x(8) = xdoty + (mp*cos(phase)); \% dy/dt
function dxdt = f(t, x, mb, mp, ms)
\% a new {f input} variabel mb has been added.
% include the BH gravity terms. see eq 38.
  rp = sqrt(x(1)^2+x(2)^2);
rs = sqrt(x(5)^2+x(6)^2);
  r = \mathbf{sqrt}((x(5)-x(1))^2+(x(6)-x(2))^2;
  dxdt(1) = x(3);
  dxdt(2) = x(4);
dxdt(3) = ((mb/rp^3)*(-x(1))) + ((ms/r^3)*(x(5)-x(1)));
dxdt(4) = ((mb/rp^3)*(-x(2))) + ((ms/r^3)*(x(6)-x(2)));
  dxdt(5) = x(7);
  dxdt(6) = x(8);
dxdt(7) = ((mb/rs^3)*(-x(5))) + ((mp/r^3)*(x(1)-x(5)));
dxdt(8) = ((mb/rs^3)*(-x(6))) + ((mp/r^3)*(x(2)-x(6)));
```

```
% The disruption of a binary by a massive BH
% the main code
% define mb, mp, ms. These are dimensionless masses.
                   % clear all variables and functions from memory
clear
      = 0.1;
D
                     % penetration factor: Rp/Rtidal
      = 1.d-3;
h
                   % time-step size
      = 10000:
                   % sampling
Ns
                  \% BH mass in units of the binary mass m
mb
      = 1e6;
                  % primary star mass in units of m
mp
      = .75;
      = .25;
                  % secondary star mass in units of m
_{
m ms}
[t, x] = initialc(D, mb, mp, ms); % Define the initial values of t and x.
     = -t:
tmax
tprint = t;
      = (tmax-t)/Ns;
dtp
while t < tmax
     if t >= tprint
         [Ep, Es] = energy(x, mb, mp, ms);
          v = [t x(1) x(2) x(5) x(6) Ep Es];
          save out v -ascii -append
          tprint = tprint + dtp;
     end
     x = RK4(h, t, x, mb, mp, ms);
     t = t+h;
end
clear; clf;
out=load('out'); % load the data file out
t = out(:,1); \% time
xp=out(:,2); % the position of the primary star
yp = out(:,3);
xs=out(:,4); % the position of the secondary star
ys = out(:,5);
Ep=out(:,6); % the energy of the primary star
Es=out (:,7); % the energy of the secondary star
\%plot (xp, yp);
%hold on
plot (xs-xp, ys-yp, 'r:')
xlabel('x', 'Fontsize', 20)
ylabel('y', 'Fontsize', 20)
```

```
set (gca, 'Fontsize', 20)
axis equal
%plot(t,Ep, 'o')
%hold on
%plot(t,Es,'r')
%xlabel('t', 'Fontsize', 20)
%ylabel('E', 'Fontsize', 20)
%set (gca, 'Fontsize', 20)
function xout=RK4(h,t,x,mb,mp,ms)
% Runge-Kutta method
% input variables: h,t,x,mb,mp,ms
% output variables: xout
% a new input variable mb is added.
k1
     = h * f(t, x,
                          mb, mp, ms);
k2
     = h*f(t+h/2,x+k1/2,mb,mp,ms);
k3
     = h*f(t+h/2,x+k2/2,mb,mp,ms);
k4
     = h * f (t+h, x+k3,
                          mb, mp, ms);
xout = x+k1/6+k2/3+k3/3+k4/6;
function [Ep, Es] = energy (x, mb, mp, ms)
% the energies of the two stars
% input variables: x, mb, mp, ms. x is a vector variable.
% out variables: Ep, Es
rp = \mathbf{sqrt}(x(1)^2 + x(2)^2);
                                 % BH-primary separation
vp2 =
           x(3)^2 + x(4)^2;
                                 % squre of primary velocity
rs = \mathbf{sqrt}(x(5)^2 + x(6)^2);
                                % BH-secondary
            x(7)^2+x(8)^2; % squre of secondary velocity
vs2 =
rps = \mathbf{sqrt}((x(5)-x(1))^2+(x(6)-x(2))^2); % primary-secondary
Ep = (mp/2)*vp2-mb*mp/rp-mp*ms/rps; % primary energy
Es = (ms/2)*vs2-mb*ms/rs-mp*ms/rps; % secondary energy
```