Solution Midterm

May 10, 2019

1 Problem 1

1.0.1 1.

The total energy $E > 0$ means the object is unbound.

1.0.2 2.

Yes. A star can achieve $E > 0$ at some point but still remain bound to the system if before it leaves the system, it goes through encounters (either two-body relaxation or strong encounters) with other objects in the system which reduce its energy to $E < 0$.

1.0.3 3.

Due to Two-Body Relaxation In a large-N system order 1 changes in the velocity occurs on a relaxation timescale ($t_{\rm relax}$) due to distant encounters. A star with $E > 0$ escapes the system at a timescale of the crossing time $t_{\rm cross}$. Thus the overall probability that a star's energy will change before it leaves the system is simply-

$$P \propto \frac{t_{\rm cross}}{t_{\rm relax}} \sim \frac{\ln N}{N} \tag{1}$$

Due to Strong Encounters The other way the velocities can change is via strong scattering. Based on our class, let's call that any encounter with an impact parameter $b \leq b_{90}$ is a strong encounter. A star moving with $v > v_{\rm esc}$ will have a strong encounter rate given by $\sim n\sigma v_r$, where n is the average number density of other stars, σ is the encounter cross-section, and v_r is the expected relative velocity between the escaping star and other stars in the system.

$$\begin{aligned}
v_r &\sim v_{\rm esc} && (2)\\
n &\sim N/R^3 && (3)\\
\sigma &\sim b_{90}^2 && (4)\\
&\sim \left(\frac{Gm_T}{v_{\rm esc}^2}\right)^2 && (5)\\
&\sim \left(\frac{GM}{N}\frac{r}{GM}\right)^2 && (6)\\
&\sim \left(\frac{R}{N}\right)^2. && (7)
\end{aligned}$$

Here, we assumed that $m_T \sim M/N$, $v_{\rm esc} \sim \frac{GM}{R}$. M is the total system mass and $m_T = m_1 + m_2$ is the total mass between the interacting objects.

Thus,

$$n\sigma v_r \sim \frac{N}{R^3} b_{90}^2 v_{\rm esc} \approx \frac{N}{R^3}\frac{R^2}{N^2} v_{\rm esc} = \frac{1}{RN} v_{\rm esc}. \tag{8}$$

So, the probability that the escaping star will have an encounter before leaving the system is -

$$P \propto P_{\rm enc} \sim n\sigma v_r t_{\rm cross} \sim \frac{1}{RN} v_{\rm esc} \frac{R}{v_{\rm esc}} = \frac{1}{N} \tag{9}$$

In general, the probability of retention is low and is inversely proportional to N. This is also called "back scattering" in N-body literature.

2 Problem 2

2.0.1 1.

Stars of different species tend to reach equipartition of kinetic energy. Thus, species m_1 moving in the potential of m_2 will try to achive-

$$\begin{aligned} m_1 < v_1^2 > &= m_2 < v_2^2 > && (10)\\ i.e., < v_1^2 > &= \frac{m_2}{m_1} < v_2^2 > && (11)\\ & && (12) \end{aligned}$$

Thus, the higher-mass species will lose energy to the lower-mass species and hence will become more bound and sink into the potential.

2.0.2 2.

Mass enclosed within r

$$M(r) = \frac{4}{3}\pi r^3 \rho \tag{13}$$

The potential at r is

$$\phi(r) = -\frac{GM(r)}{r} = \frac{4\pi G\rho}{3} r^2 \tag{14}$$

Using $x = r\cos\theta$ and $y = r\sin\theta$, we get a complete set of equations:

$$\ddot{x} = -\frac{\partial\phi}{\partial x} = -\frac{8\pi G\rho}{3} x \tag{15}$$

and

$$\ddot{y} = -\frac{\partial\phi}{\partial y} = -\frac{8\pi G\rho}{3} y \tag{16}$$

These are equations for a 2D simple harmonic oscillator.

2.0.3 3.

Turning points are where the velocity is zero. For simplicity in the next part, let's go to the r, θ coordinate system and invoke integrals of motion.

Total energy

$$E = \frac{1}{2}\left(\dot{r}^2 + r^2\dot{\theta}^2\right) + \phi(r) \tag{17}$$

Invoking, E and $r^2\dot{\theta} = L$ to be conserved, and replacing the $\dot{\theta}$ term above we get,

$$\dot{r}^2 = 2[E - \phi(r)] - \frac{L^2}{r^2} \tag{18}$$

The turning points are the solution of

$$\begin{aligned} \dot{r} &= 0 && (19) \\ i.e., r &= \pm\sqrt{2[E - \phi(r)] - \frac{L^2}{r^2}} && (20) \\ & && (21) \end{aligned}$$

2.0.4 4.

The problem essentially is to find out the confinement radius of a species that keeps losing energy to the other species. The spatial separation is simply because the gradual reduction of $< v_1^2 >$ over time and since the average kinetic energy of a species is connected to the average potential energy of the species, we can relate $< v_1^2 >$ to some confinement radius in an order of magnitude way using the virial equilibrium.

The kinetic energy in species 1 is $E_{K,1} \sim < v_1^2 >$.

The potential energy in species 1 assuming uniform density is $E_{P,1} \sim \phi_1 \sim R_1^2$, where R_1 is the characteristic radial extent of the orbits of species 1.

Invoking virial equilibrium,

$$\begin{aligned} E_{K,1} &= \frac{1}{2}E_{P,1} && (22) \\ i.e., < v_1^2 > &\sim R_1^2 && (23) \\ i.e., \frac{m_2}{m_1} < v_2^2 > &\sim R_1^2 && (24) \\ i.e., \frac{m_2}{m_1}R_2^2 &\sim R_1^2 && (25) \\ i.e., R_1 &\sim \left(\frac{m_2}{m_1}\right)^{1/2} R_2 && (26) \end{aligned}$$

Assuming that the size of the system R essentially is the size of the subsystem species 2, it follows,

$R_1 \sim \left(\frac{m_2}{m_1}\right)^{1/2} R$.

2.0.5 5.

One of the basic assumptions was that $M_1 << M_2$, as a result, we could think of this problem simply as the species 1 is sinking in the unchanging background potential of species 2. However, as R_1 decreases keeping the density of species 2 ρ_2 constant, at some point the total mass in species 2 within R_1 will become comparable to M_1. This is where our basic assumption will not hold.

Total mass of species 2 contained within R_1 is

$$M_{(2;\leq R_1)} = \rho_2 \frac{4}{3}\pi R_1^3 = \left(\frac{m_2}{m_1}\right)^{3/2} \frac{4}{3}\pi R^3 \rho_2 = \left(\frac{m_2}{m_1}\right)^{3/2} M_2 \tag{27}$$

The assumption of species 1 moving in a potential dominated by species 2 of course breaks down when

$M_1 \geq M_{(2;\leq R_1)} = \left(\frac{m_2}{m_1}\right)^{3/2} M_2$.

3 Problem 3

3.0.1 1.

For a polytropic Ergodic DF,

$\rho = \psi^n$.

If $\rho \propto r^{-\alpha}$, then $\psi \sim r^{-\alpha/n}$.

Force per unit mass is-

$$\frac{GM(r)}{r^2} = -\frac{d\psi}{dr}. \tag{28}$$

Thus, $M(r) = \frac{r^2}{G}\frac{d\psi}{dr} \sim r^{(n-3)/(n-1)}$.

3.0.2 (a)

$n = 3$ implies $M(r) \sim$ constant. This is equivalent to a central point mass, thus for any $r > 0$, the mass enclosed is the same.

3.0.3 (b)

$n \to \infty$ implies $M(r) \sim r$ and equivalently, from the ideal gas equation, $\gamma = 1 + \frac{1}{n} = 1$. Thus, the equivalent thermodynamic system will have $p = k\rho$, which is the EOS for isothermal gas.

This limit represents isothermal sphere.

3.0.4 2.

3.0.5 (a)

Polytropic Poisson's equation with $\rho = c_n \Psi^n$:

$$\frac{1}{r^2}\frac{d}{dr}\left(r^2 \frac{d\Psi}{dr}\right) + 4\pi G c_n \Psi^n = 0 \tag{29}$$

for $\Psi > 0$. The density becomes zero if $\Psi = 0$ and undefined if $\Psi < 0$.

Substituting,

$$\Psi = \Psi_0 \psi, \tag{30}$$

$$s = \frac{r}{b} \tag{31}$$

$$i.e., \frac{d}{dr} = \frac{1}{b}\frac{d}{ds}, \tag{32}$$

We can rewrite the first term of the Poisson's equation as

$$\frac{\Psi_0}{b^2}\frac{1}{s^2}\frac{d}{ds}\left(s^2\frac{d\psi}{ds}\right). \tag{33}$$

Using, $\frac{1}{b^2} = \frac{4}{3}\pi G \Psi_0^{n-1} c_n$, one can rewrite the second term as-

$$4\pi G c_n \Psi^n = 4\pi G c_n \Psi_0^n \psi^n = 3\frac{\Psi_0}{b^2}\psi^n. \tag{34}$$

Dividing both terms by Ψ_0/b^2, and remembering the limits of validity for a polytropic equation ($\Psi > 0$) we get-

$$\frac{1}{s^2}\frac{d}{ds}\left(s^2\frac{d\psi}{ds}\right) = -3\psi^n \; (\psi > 0) \tag{35}$$

$$= 0 \; (\psi \leq 0) \tag{36}$$

3.0.6 (b)

For Plummer sphere, $\phi = -\frac{GM}{\sqrt{r^2+b^2}}$. By definition, $\Psi = -\phi + \phi_0$. Defining $\phi_0 = 0$, $\Psi = \frac{GM}{\sqrt{r^2+b^2}}$.

Hence, $\Psi_0 \equiv \Psi(0) = \frac{GM}{b}$. Thus, $\psi = \frac{\Psi}{\Psi_0} = \frac{1}{\sqrt{1+r^2/b^2}} = \frac{1}{\sqrt{1+s^2}}$, where $s \equiv \frac{r}{b}$.

Putting this in the Lane-Emden equation,

$$\frac{1}{s^2}\frac{d}{ds}\left(s^2\frac{d\psi}{ds}\right) = \frac{1}{s^2}\frac{d}{ds}\left(s^2\frac{d(1+s^2)^{-1/2}}{ds}\right) \tag{37}$$

$$= -3\psi^5. \tag{38}$$

Thus, clearly, the Plummer sphere is a solution of the Lane-Emden equation with $n = 5$.

3.0.7 (c)

According to the Polytropic equation, density

$$\rho = c_5 \Psi^5 = c_5 \frac{\Psi_0^5}{(1+s^2)^{5/2}} \tag{39}$$

For $r \to \infty$, i.e., $s \to \infty$, $\rho \to c_5 \Psi_0^5 s^{-5} = c_5 \Psi_0^5 b^5 r^{-5}$. Thus, the density is nowhere zero although it reduces quite fast with increasing r.

The total mass can be obtained if we can find the force on a unit mass at $r \to \infty$ due to the system. The force on a unit mass at $r \to \infty$ is-

$$
\begin{aligned}
\left(\frac{GM}{r^2}\right)_{r\to\infty} &= \left(\frac{d\phi}{dr}\right)_{r\to\infty} && (40)\\
i.e., M &= \frac{1}{G}\left(r^2\frac{d\phi}{dr}\right)_{r\to\infty} && (41)\\
i.e., M &= -\frac{b}{G}\left(s^2\frac{d\Psi}{ds}\right)_{s\to\infty} && (42)\\
i.e., M &= -\frac{b\Psi_0}{G}\left(s^2\frac{d\psi}{ds}\right)_{s\to\infty} && (43)\\
i.e., M &= \frac{b\Psi_0}{G}\left(\frac{s^3}{(1+s^2)^{3/2}}\right)_{s\to\infty} && (44)\\
i.e., M &= \frac{b\Psi_0}{G} && (45)\\
& && (46)
\end{aligned}
$$

Hence, the total mass is finite.

An alternative way to show the above. For Plummer sphere, $\phi = -\frac{GM}{\sqrt{r^2+b^2}}$.

From Poisson's equation,

$$\frac{1}{r^2}\frac{d}{dr}\left(r^2\frac{d\phi}{dr}\right) = 4\pi G\rho. \tag{47}$$

Solving for ρ gives

$$\rho(r) = \frac{3M}{4\pi b^3}\left(1+\frac{r^2}{b^2}\right)^{-5/2}. \tag{48}$$

For $r >> b$, $\rho \sim r^{-5}$. Thus, it is non-zero everywhere, although it goes down rapidly.
The mass within r is given by-

$$
\begin{aligned}
M(r) &= \int_0^r 4\pi x^2 dx \frac{3M}{4\pi b^3}\left(1+\frac{x^2}{b^2}\right)^{-5/2} && (49)\\
&= 3M\int_0^{r/b}\frac{x^2}{b^2}d(x/b)\left(1+\frac{x^2}{b^2}\right)^{-5/2} && (50)\\
&= 3M\int_0^{r/b} y^2\left(1+y^2\right)^{-5/2}dy && (51)\\
&= 3M\frac{r^3/b^3}{3(1+r^2/b^2)^{3/2}} && (52)\\
&\approx M \text{ for r} >> \text{b} && (53)\\
& && (54)
\end{aligned}
$$

Thus, the total mass is finite.

3.0.8 (c)

The Lane-Emden equation does not have analytic solutions unless $n = 1, 5$. So, let's consider the behavior of the potential around the above potential form. Note from the above that as n increases, the potential decreases more slowly with increasing r. In the Plummer potential, $(d\phi/dr)_{r\to\infty} \sim r^{-2}$.

The total mass $M \sim r^2\frac{d\phi}{dr}$. So, if $d\phi/dr \sim r^{-2}$, the mass is finite. If $d\phi/dr \sim r^{-2+\delta}$, then clearly that mass will not be finite as $r \to \infty$, where $\delta > 0$. On the other hand, for any $\delta \leq 0$, the mass remains finite.

In addition, note that for $n = 5$ the $\frac{dM(r)}{dr} \to 0$ for $r \to \infty$. Physically this means that as r is increases, very little additional mass is obtained and the total mass asymptotes to M. For $n < 5$, based on the above order-of-magnitude argument around $n = 5$, one can show that $\frac{dM(r)}{dr} < 0$. If the extent is infinite, this would indicate that as you increase r, mass enclosed is reduced. This is clearly meaningless. So, the extent must be finite. On the other hand, for $n > 5$, $\frac{dM(r)}{dr} > 0$ for $r \to \infty$. So mass as well as extent are infinite.

Of course, a more rigorous proof can be obtained by numerically solving the Lane-Emden equation for $n \neq 1, 5$. If one plots the density for those solutions for $n < 5$, the density crosses the r axis indicating that for some finite r the density becomes negative.

3.1 Problem 4

3.1.1 1.

The three-dimensional velocity dispersion assuming isotropy is

$$v_\sigma^2 = < v^2 > = \frac{\int_{-\infty}^{\infty} d^3v\, v^2 f(\mathcal{E})}{\int_{-\infty}^{\infty} d^3v\, f(\mathcal{E})} \tag{55}$$

where all symbols have usual meaning.

$$\begin{aligned} f &= F\mathcal{E}^{n-3/2} \text{ for } \mathcal{E} > 0 && (56)\\ &= 0 \text{ for } \mathcal{E} \leq 0 && (57)\\ & && (58)\end{aligned}$$

Now notice that

$$\mathcal{E} = -\phi - \frac{1}{2}v^2, \tag{59}$$

and for Plummer sphere

$$n = 5. \tag{60}$$

$$< v^2 > = \frac{\int_0^{\sqrt{-2\phi}} dv\, v^4\, (-\phi - \frac{1}{2}v^2)^{7/2}}{\int_0^{\sqrt{-2\phi}} dv\, v^2\, (-\phi - \frac{1}{2}v^2)^{7/2}}. \tag{61}$$

Replace variables such that,

$$
\begin{aligned}
v &= \sqrt{-2\phi}\ \sin\ \theta && (62)\\
dv &= \sqrt{-2\phi}\ \cos\ \theta\ d\theta && (63)\\
& && (64)
\end{aligned}
$$

$$
< v^2 >= \frac{-2\phi \int_0^{\pi/2} \sin^4\ \theta\ (1-\sin^2\theta)^{7/2}\ cos\ \theta\ d\theta}{\int_0^{\pi/2} \sin^2\ \theta\ (1-\sin^2\ \theta)^{7/2}\ cos\ \theta\ d\theta}. \tag{65}
$$

Again replace variables-

$$
\begin{aligned}
\sin\theta &= t && (66)\\
\cos\theta\ d\theta &= dt, && (67)\\
& && (68)
\end{aligned}
$$

then,

$$
\begin{aligned}
< v^2 > &= \frac{-2\phi\ \int_0^1\ dt\ t^4\ (1-t^2)^{7/2}}{\int_0^1\ dt\ t^2\ (1-t^2)^{7/2}} && (69)\\
&\approx -2\phi\frac{7\pi/2048}{7\pi/512} && (70)\\
&= -\frac{\phi}{2} && (71)\\
&= \frac{GM}{2\sqrt{r^2+b^2}} && (72)\\
&= \frac{GM}{2b\sqrt{1+r^2/b^2}} && (73)\\
&= \left(\frac{\pi}{24}\frac{G^5M^4}{b^2}\rho\right)^{1/5} && (74)
\end{aligned}
$$

3.1.2 2.

$$
\Gamma_{\rm BS} = n\sigma v_\infty = n\pi b^2_{\rm impact} v_\infty \tag{75}
$$

The impact parameter, $b^2_{\rm impact} = a^2(1+\frac{v_c^2}{v_\infty^2})$, where a is the semimajor axis, $v_c \equiv \left[\frac{2G(m_1+m_2+m)}{a}\right]^{1/2}$ is the critical velocity. Thus,

$$
\Gamma_{\rm BS} = n\pi v_\infty a^2 \left[1+\frac{2G(m_1+m_2+m)}{av_\infty^2}\right] \tag{76}
$$

Replacing $n = \rho/ < m_\star >$, we get

$$
\Gamma_{\rm BS} = \pi\frac{\rho}{< m_\star >} v_\infty a^2 \left[1+\frac{2G(m_1+m_2+m)}{av_\infty^2}\right] \tag{77}
$$

Note: It is quite obvious that the larger the semimajor axis, the larger the cross section. The largest semimajor axis that can remain unbroken is calculated by saying that the total energy of the 3-body system is zero. i.e., equate the kinetic energy of the reduced 3-body system (E_K) is equal to the total energy of the binary (E_b).

$$E_K = E_b \tag{78}$$

$$i.e., \frac{1}{2}\frac{m(m_1+m_2)}{m+m_1+m_2}v_\infty^2 = \frac{Gm_1m_2}{2a} \tag{79}$$

$$i.e., a_{\max} = \frac{Gm_1m_2(m_1+m_2+m)}{m(m_1+m_2)v_\infty^2} \tag{80}$$

$$\tag{81}$$

3.1.3 3.

Escape speed

$$v_{\rm esc} = \sqrt{-2\phi} = \left[\frac{2GM}{(r^2+b^2)^{1/2}}\right]^{1/2} = \left(\frac{2GM}{b}\frac{1}{\sqrt{1+(r/b)^2}}\right)^{1/2} \tag{82}$$

3.1.4 4.

In general, for merging BBHs to be produced, $t_s + t_{\rm formation} + t_{\rm merger} \leq$ age of the universe, where, $t_{\rm formation}$ and $t_{\rm merger}$ are timescale for BBH formation via dynamical processes, and timescale for merger through GW radiation. Below we estimate the different timescales.

Mass segregation For a given cluster of mass M, the total number of stellar objects is $N \equiv M/ < m_\star >$ and the number of binaries is $N_b \equiv f_b N$.

Most of the interactions happen at the deepest part of the potential, i.e., $r << b$. Note, in principle, one can integrate over all r, but for simplicity, we will choose that all interactions are happening at small r. Also note that $\Gamma \propto \rho$, so, interactions will happen more frequently at small r anyway.

$t_{\rm relax} \sim \frac{N}{0.1 \ln N} t_{\rm dyn} \sim \frac{N}{0.1 \ln N}\frac{R}{<v^2>^{1/2}}$, where, $t_{\rm relax}$ and $t_{\rm dyn}$ are the local two-body relaxation timescale and dynamical timescale, respectively, R is some characteristic size, and $< v^2 >$ is the 3-D velocity dispersion.

Since Plummer sphere has infinite extent, we need to decide on some characteristic size R_x and average velocity dispersion $\overline{< v^2 >_x}$ upto R_x to estimate $t_{\rm dyn}$ such that mass enclosed within R_x, $M(R_x) = xM$, where, $x < 1$.

$$R_x = b\left(\frac{x^{2/3}}{1-x^{2/3}}\right)^{1/2} \tag{83}$$

The average velocity dispersion, $\overline{< v^2 >_x}$ is given by-

$$\overline{< v^2 >_x} = \frac{\int_0^{R_x} < v^2 > 4\pi r'^2 \rho dr'}{\int_0^r 4\pi r'^2 \rho dr'} = \frac{3GM}{2b}\left[\frac{tan^{-1}(R_x/b)}{8} + \frac{(R_x/b)^3 - R_x/b}{8(R_x/b)^4 + 16(R_x/b)^2 + 8}\right] \tag{84}$$

It is also OK to simply estimate a characteristic $< v^2 >_x$ at R_x by-

$$< v^2 >_x = \frac{GM}{2b\sqrt{1 + R_x^2/b^2}} \tag{85}$$

The dynamical time is roughly-

$$t_{\rm dyn} \approx \frac{R_x}{\sqrt{< v^2 >_{r \le r_x}}} \tag{86}$$

$$\text{or more approximately} \approx \frac{R_x}{\sqrt{< v^2 >_x}} \tag{87}$$

BHs sink over the mass segregation timescale $t_{\rm MS} \sim \frac{m_{\rm BH}}{< m_\star >} t_{\rm relax}$.

Interaction timescale Interaction rate for binary--single encounter is-

$$\Gamma = n\sigma v_\infty = n\pi a^2 \left[1 + \frac{2G(m_1 + m_2 + m)}{a v_\infty^2}\right] v_\infty \tag{88}$$

Timescale for binary--single interactions for a single binary is Γ^{-1}.
The timeacale for any interaction is $(N_b\Gamma)^{-1}$.

Of course in a cluster there is a range in a. However, most interactions happen with the largest allowed a that does not get broken in the cluster environment due to strong encounters. This limit can be calculated using the idea of critical velocity. If you equate the kinetic energy of the reduced three-body system with the binding energy of the binary, then the total energy is zero. The total energy must be > 0 is ionization is allowed. Hence, the largest binary would be where a is such that the total energy is zero.

Using this criteria, one can get the maximum $a = a_{\rm max}$ as

$$a_{\rm max} = \frac{G m_1 m_2 (m_1 + m_2 + m)}{m(m_1 + m_2) v_\infty^2} \tag{89}$$

In the numerical part we will simply use $a = a_{\rm max}$ to calculate the interaction rate.

Inspiral timescale Use Peters equation to estimate the merger timescale for any given binary. Note that the timescale to merger is a very steep function of a and e. Thus, it is best to use the smallent a that dynamics can create in the cluster environment.

If one assumes that the recoil speed $v_{\rm recoil} \sim v_{\rm orb}$, then the smallest $a = a_{\rm min}$ is given by when $v_{\rm recoil} = v_{\rm orb} = v_{\rm esc}$, where, $v_{\rm esc}$ is the escape speed from the cluster. Since the inspiral time is the shortest for $a_{\rm min}$ for a given e, we will simply use this as a characteristic $t_{\rm inspiral}$ for a cluster.

combine everything Below you can find scripts that can calculate all timescales. Essentially, clusters that have the lowest total time will dominate.

```
In [172]: import numpy as np
          import scipy
          import scipy.integrate

          constants = {'AU': 1.496e13,
```

```
             'PC': 3.086e18,
             'lightyear': 9.463e17,
             'Msun': 1.99e33,
             'Rsun': 6.96e10,
             'Lsun': 3.9e33,
             'Tsun': 5.780e3,
             'km': 1e5,
             'yr': 31556925.9936,
             'G': 6.67259e-8
             }
import numpy as np

def v_esc(M, b=1.):
    K = constants['G']*constants['Msun']/constants['PC']
    vesc = (2.*K*(M/b))**0.5
    return vesc

def v_orb(m1, m2, a):
    K = constants['G']*constants['Msun']/constants['AU']
    vorb = (K*(m1+m2)/a)**0.5
    return vorb

def get_semimajor(m1, m2, vorb):
    K = constants['G']*constants['Msun']/constants['AU']
    a = K * (m1+m2)/vorb**2.
    return a

def inspiral_time_peters(a0,e0,m1,m2):
    """
    Computes the inspiral time, in Gyr, for a binary
    a0 in Au, and masses in solar masses
    """
    coef = 6.086768e-11 #G^3 / c^5 in au, gigayear, solar mass units
    beta = (64./5.) * coef * m1 * m2 * (m1+m2)
    if e0 == 0:
        return a0**4 / (4*beta)
    c0 = a0 * (1.-e0**2.) * e0**(-12./19.) * (1.+(121./304.)*e0**2.)**(-870./2299.)
    time_integrand = lambda e: e**(29./19.)*(1.+(121./304.)*e**2.)**(1181./2299.) /
    integral,abserr = scipy.integrate.quad(time_integrand,0,e0)
    return integral * (12./19.) * c0**4. / beta


def beta(m1,m2):
    coef = 6.086768e-11 #G^3 / c^5 in au, gigayear, solar mass units
    #coef =  6.086768e-20 #G^3 / c^5 in au, year, solar mass units
    G = 3.96511851e-14 #G in au, solar mass, second units
    return (64./5.) * coef * m1 * m2 * (m1+m2)
```

```python
def c0(a0,e0):
    return a0 * (1.-e0**2.) * e0**(-12./19.) * (1.+(121./304.)*e0**2.)**(-870./2299.)

def time_integrand(e):
    return e**(29./19.)*(1.+(121./304.)*e**2.)**(1181./2299.) / (1.-e**2.)**1.5


def vsigma(M, b=1., x=0.5):
    x23 = x**(2./3.)
    rx_by_b = (x23/(1-x23))**0.5
    K = constants['G']*constants['Msun']/constants['PC']
    sigma = K*0.5*M*(1+rx_by_b**2.)**(-0.5)
    sigma = sigma**0.5
    return sigma

def ave_vsigma(M, b=1., x=0.5):
    x23 = x**(2./3.)
    rx_by_b = (x23/(1-x23))**0.5
    K = constants['G']*constants['Msun']/constants['PC']
    ave_v = np.arctan(rx_by_b/8.) + (rx_by_b**3. - rx_by_b)/(8*rx_by_b**4. + 16.*rx_
    ave_v = ave_v * 1.5 * K * M/b
    ave_v = ave_v**0.5
    return ave_v

def t_dyn(M, b=1., x=0.5):
    x23 = x**(2./3.)
    rx = b * ((x23/(1-x23))**0.5)
    ave_v = ave_vsigma(M, b=b, x=x)
    t_dyn = rx*constants['PC']/ave_v
    t_dyn = t_dyn/constants['yr']
    return t_dyn

def t_relax(M, b=1., mave=0.5, x=0.5):
    """
        Input: cluster mass M in Msun
                scale length b in pc
                average stellar mass mave in Msun
                BH mass mbh in Msun
        Return: Relaxation timescale in Gyr
    """
    N = M/mave
    tdyn = t_dyn(M, b=b, x=x)
    t = (N/(0.1*np.log(N))) * tdyn
    t = t/1e9
    return t

def t_mass_segregation(M, b=1., mave=0.5, mbh=10., x=0.5):
    """
```

```
        Input: cluster mass M in Msun
                scale length b in pc
                average stellar mass mave in Msun
                BH mass mbh in Msun
        Return: Mass segregation timescale in Gyr
    """
    trelax = t_relax(M, b=b, mave=mave, x=0.5)
    mratio = mave/mbh
    tms = trelax * mratio
    return tms

def central_density(M, b=1.):
    """
        Input: cluster mass M in Msun
                scale length b in pc
        Return: density at r<<b in Msun/pc^3
    """
    rhoc = 3*M/(4.*np.pi*b**3)
    return rhoc

def central_vsigma(M, b=1.):
    """
        Input: cluster mass M in Msun
                scale length b in pc
        Return: velocity dispersion at r<<b in km/s
    """
    K = constants['G']*constants['Msun']/constants['PC']
    vsigmac = (K * M/(2.*b))**0.5
    vsigmac = vsigmac*1e-5
    return vsigmac

def t_inspiral_vs_mass(M, b=1., e=2./3., m1=10., m2=10.):
    vesc = v_esc(M, b=b)
    a = get_semimajor(m1, m2, vesc)
    tinsp = inspiral_time_peters(a,e,m1,m2)
    return tinsp

def semimajor_for_kT_binary(vinf, m=0.5, m1=10., m2=10., kt=1.):
    """
        input: vinf:- is velocity at infinity in km/s
                m:- mass of the incoming star
                m1, m2:- masses of the binary components
                kt:- how bound the binary is relative to the incoming star
                    i.e., kt = Eb/(Ek), where Eb is the binary total energy,
                    Ek is the kinetic energy of the incoming star
        return: a:- semimajor axis of the binary in AU
    """
    K = constants['G']*constants['Msun']/(constants['km']*constants['km']*constants[
```

```
    a = (1./kt) * m1*m2*(m1+m2+m)/(m*(m1+m2)*vinf*vinf)
    a = K*a
    return a

def cross_section(M, b=1., m=0.5, m1=10., m2=10., kt=1.):
    vsigmac = central_vsigma(M, b=b) #km/s
    a = semimajor_for_kT_binary(vsigmac, m=m, m1=m1, m2=m2, kt=kt)
    #print (vsigmac, a)
    K = constants['G']*constants['Msun']/(constants['AU']*constants['km']*constants[
    gf = 2.* K * (m1+m2+m) / (a*vsigmac*vsigmac)
    crosssection = np.pi*a*a*(1.+gf) #pc^2
    return crosssection

def Gamma_interaction(M, b=1., mave=0.5, m=0.5, m1=10., m2=10., kt=1.):
    rhoc = central_density(M, b=b)
    nc = rhoc/mave  #pc^{-3}
    vsigmac = central_vsigma(M, b=b) #km/s
    cross = cross_section(M, b=b, m=m, m1=m1, m2=m2, kt=kt)
    K = constants['AU']*constants['AU']*constants['km']*constants['yr']*1e9/(constan
    gamma = K*nc*cross*vsigmac #Gyr^-1
    return gamma
```

4 Plotting

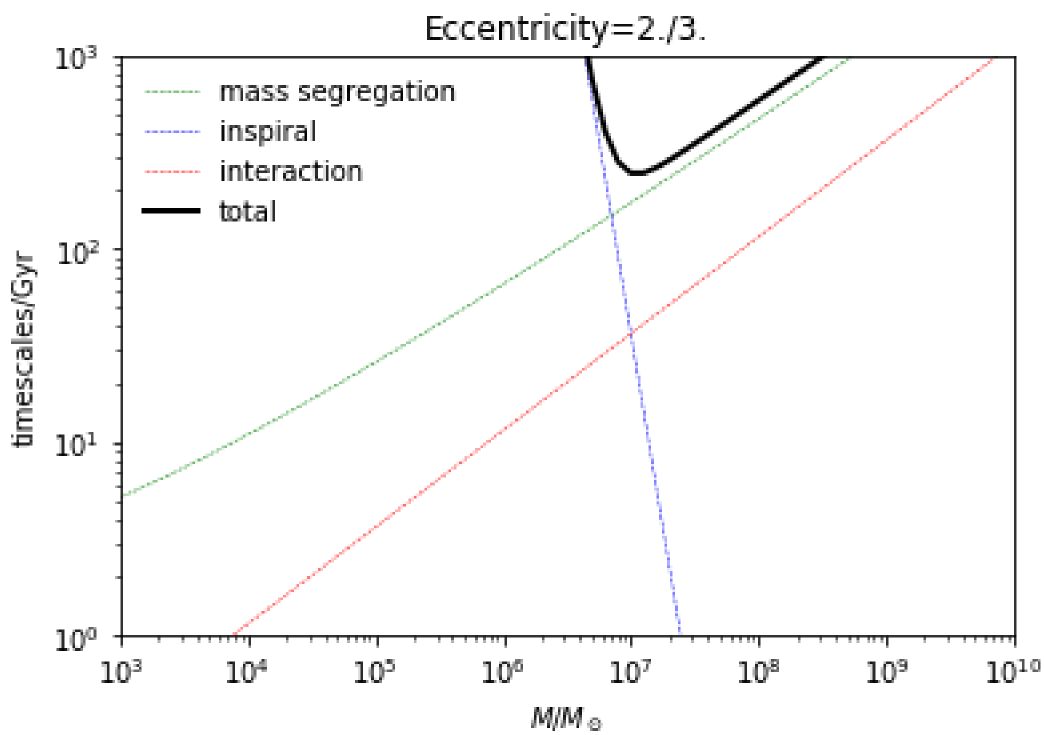
Below we will plot the different timescales and the combined timescale to get an idea where dynamical formation of merging black hole binaries would be most efficient.

```
In [173]: import matplotlib.pyplot as plt
          %matplotlib inline

          b, mave, mbh, frac, ecc, kT = 1., 10., 0.5, 0.5, 2./3., 1.
          mass = np.arange(3., 10.1, 0.1)
          mass = 10**mass
          t_r = t_relax(mass, b=b, mave=mave, x=frac)
          t_ms = t_mass_segregation(mass, b=b, mave=mave, mbh=mbh, x=frac)
          t_insp = t_inspiral_vs_mass(mass, e=ecc)
          gamma = Gamma_interaction(mass, b=b, mave=mave, m=mave, m1=mbh, m2=mbh, kt=kT)
          t_interact = gamma**(-1)
          plt.plot(mass,t_ms, ls='dashed', lw=0.5, color='green', label='mass segregation')
          plt.plot(mass,t_insp, ls='dashed', lw=0.5, color='blue', label='inspiral')
          plt.plot(mass,t_interact, ls='dashed', lw=0.5, color='red', label='interaction')
          ttot = t_interact+t_insp+t_ms
          plt.plot(mass,ttot, ls='solid', lw=2, color='black', label='total')
          plt.xscale('log')
          plt.yscale('log')
          plt.xlim([1e3,1e10])
          plt.ylim([1,1e3])
```

```
plt.xlabel(r'$M/M_\odot$')
plt.ylabel('timescales/Gyr')
plt.legend(loc='best', numpoints=1, frameon=0)
plt.title("Eccentricity=2./3.")
mmax = np.log10(mass[np.argsort(ttot)[0]])
print(r'for characteristic e=2/3 : log(M/Msun) = %g' %(mmax,))
```

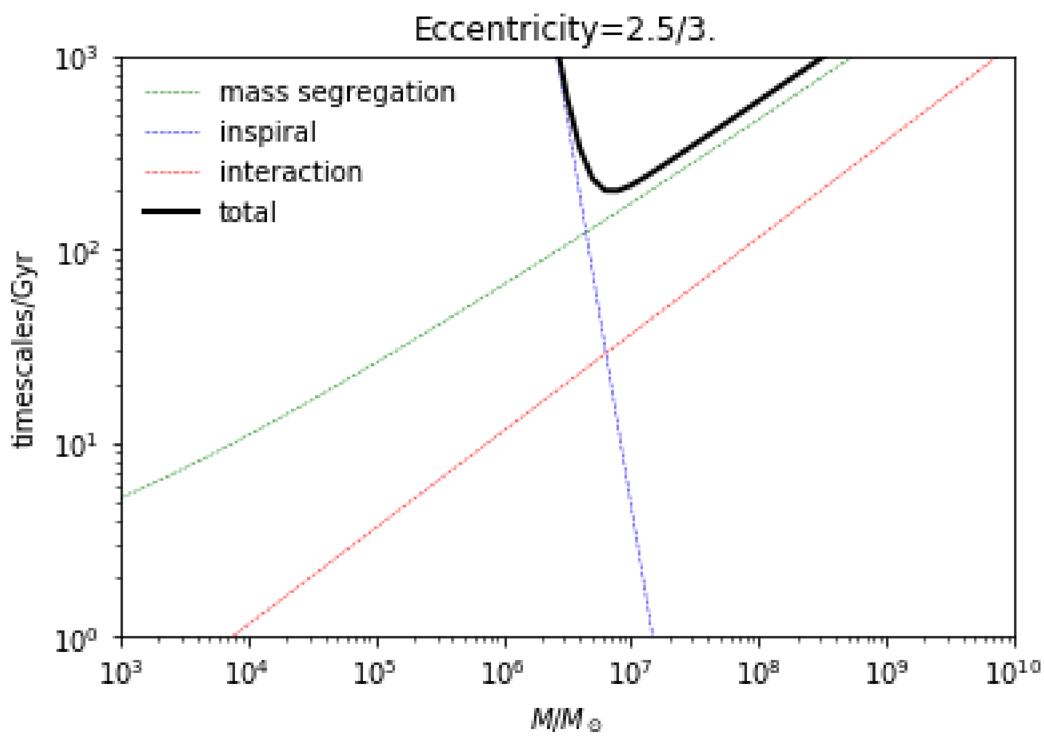
```
for characteristic e=2/3 : log(M/Msun) = 7
```




```
In [174]: b, mave, mbh, frac, ecc, kT = 1., 10., 0.5, 0.5, 2.5/3., 1.
          mass = np.arange(3., 10.1, 0.1)
          mass = 10**mass
          t_r = t_relax(mass, b=b, mave=mave, x=frac)
          t_ms = t_mass_segregation(mass, b=b, mave=mave, mbh=mbh, x=frac)
          t_insp = t_inspiral_vs_mass(mass, e=ecc)
          gamma = Gamma_interaction(mass, b=b, mave=mave, m=mave, m1=mbh, m2=mbh, kt=kT)
          t_interact = gamma**(-1)
          plt.plot(mass,t_ms, ls='dashed', lw=0.5, color='green', label='mass segregation')
          plt.plot(mass,t_insp, ls='dashed', lw=0.5, color='blue', label='inspiral')
          plt.plot(mass,t_interact, ls='dashed', lw=0.5, color='red', label='interaction')
          ttot = t_interact+t_insp+t_ms
          plt.plot(mass,ttot, ls='solid', lw=2, color='black', label='total')
```

```
plt.xscale('log')
plt.yscale('log')
plt.xlim([1e3,1e10])
plt.ylim([1,1e3])
plt.xlabel(r'$M/M_\odot$')
plt.ylabel('timescales/Gyr')
plt.legend(loc='best', numpoints=1, frameon=0)
plt.title("Eccentricity=2.5/3.")
mmax = np.log10(mass[np.argsort(ttot)[0]])
print(r'for characteristic e=2/3 : log(M/Msun) = %g' %(mmax,))
```

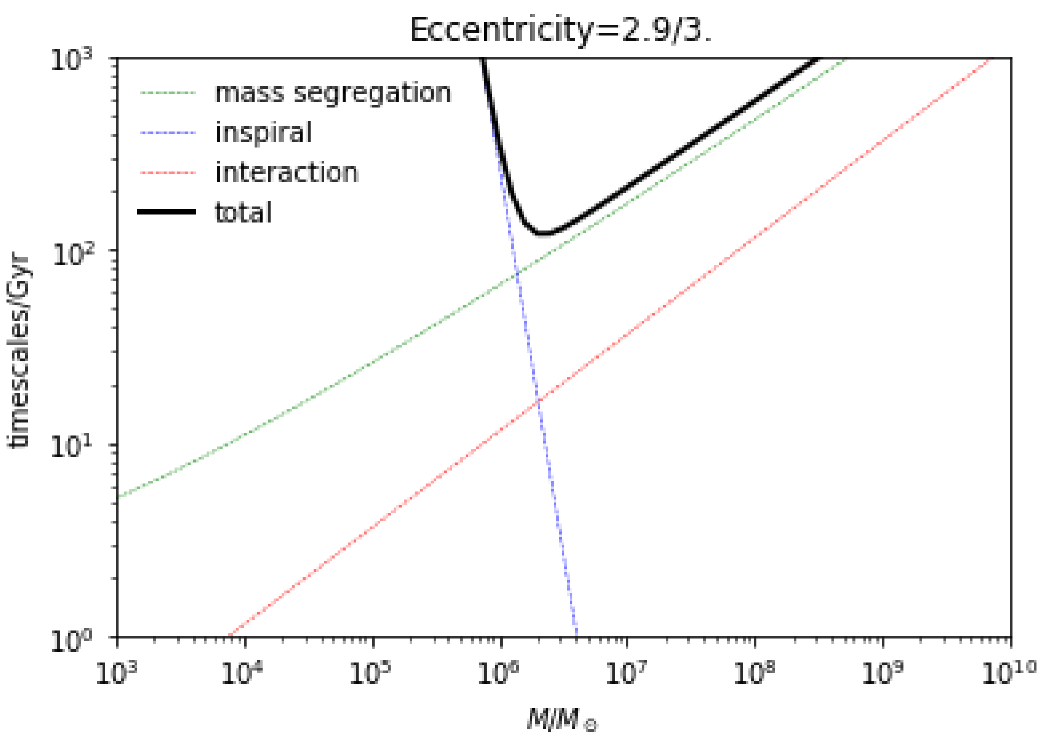
```
for characteristic e=2/3 : log(M/Msun) = 6.9
```




```
In [175]: b, mave, mbh, frac, ecc, kT = 1., 10., 0.5, 0.5, 2.9/3., 1.
          mass = np.arange(3., 10.1, 0.1)
          mass = 10**mass
          t_r = t_relax(mass, b=b, mave=mave, x=frac)
          t_ms = t_mass_segregation(mass, b=b, mave=mave, mbh=mbh, x=frac)
          t_insp = t_inspiral_vs_mass(mass, e=ecc)
          gamma = Gamma_interaction(mass, b=b, mave=mave, m=mave, m1=mbh, m2=mbh, kt=kT)
          t_interact = gamma**(-1)
          plt.plot(mass,t_ms, ls='dashed', lw=0.5, color='green', label='mass segregation')
```

```
plt.plot(mass,t_insp, ls='dashed', lw=0.5, color='blue', label='inspiral')
plt.plot(mass,t_interact, ls='dashed', lw=0.5, color='red', label='interaction')
ttot = t_interact+t_insp+t_ms
plt.plot(mass,ttot, ls='solid', lw=2, color='black', label='total')
plt.xscale('log')
plt.yscale('log')
plt.xlim([1e3,1e10])
plt.ylim([1,1e3])
plt.xlabel(r'$M/M_\odot$')
plt.ylabel('timescales/Gyr')
plt.legend(loc='best', numpoints=1, frameon=0)
plt.title("Eccentricity=2.9/3.")
mmax = np.log10(mass[np.argsort(ttot)[0]])
print(r'for characteristic e=2.9/3 : log(M/Msun) = %g' %(mmax,))
```

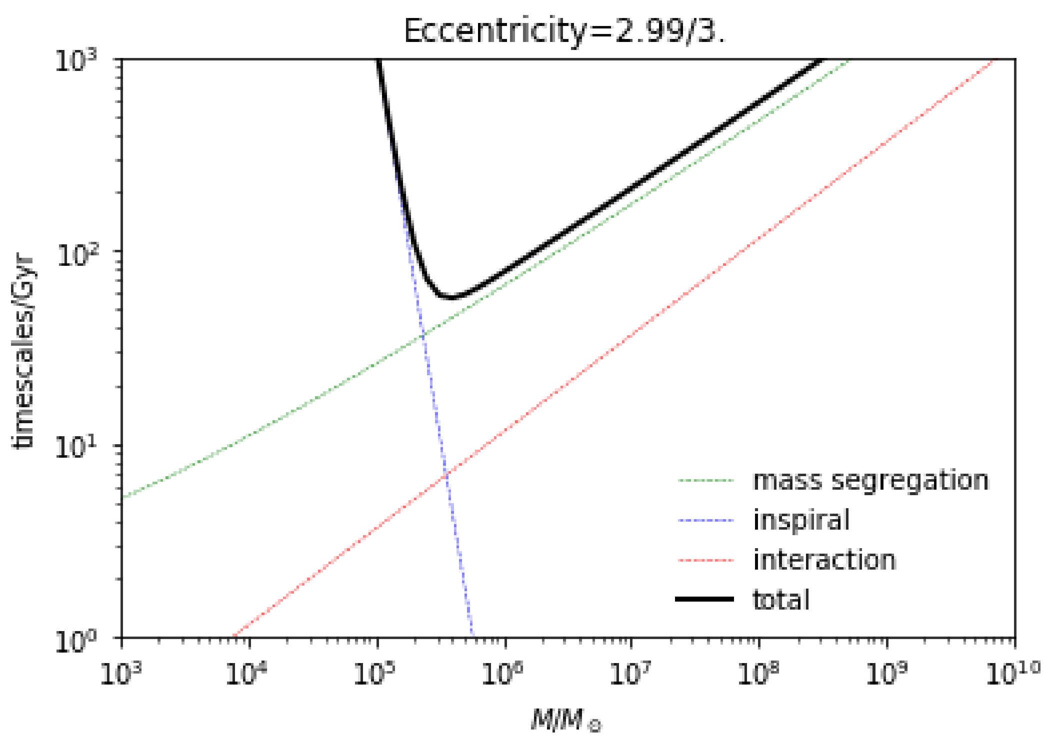
```
for characteristic e=2.9/3 : log(M/Msun) = 6.3
```




```
In [176]: b, mave, mbh, frac, ecc, kT = 1., 10., 0.5, 0.5, 2.99/3., 1.
          mass = np.arange(3., 10.1, 0.1)
          mass = 10**mass
          t_r = t_relax(mass, b=b, mave=mave, x=frac)
          t_ms = t_mass_segregation(mass, b=b, mave=mave, mbh=mbh, x=frac)
```

```
t_insp = t_inspiral_vs_mass(mass, e=ecc)
gamma = Gamma_interaction(mass, b=b, mave=mave, m=mave, m1=mbh, m2=mbh, kt=kT)
t_interact = gamma**(-1)
plt.plot(mass,t_ms, ls='dashed', lw=0.5, color='green', label='mass segregation')
plt.plot(mass,t_insp, ls='dashed', lw=0.5, color='blue', label='inspiral')
plt.plot(mass,t_interact, ls='dashed', lw=0.5, color='red', label='interaction')
ttot = t_interact+t_insp+t_ms
plt.plot(mass,ttot, ls='solid', lw=2, color='black', label='total')
plt.xscale('log')
plt.yscale('log')
plt.xlim([1e3,1e10])
plt.ylim([1,1e3])
plt.xlabel(r'$M/M_\odot$')
plt.ylabel('timescales/Gyr')
plt.legend(loc='best', numpoints=1, frameon=0)
plt.title("Eccentricity=2.99/3.")
mmax = np.log10(mass[np.argsort(ttot)[0]])
print(r'for characteristic e=2.99/3 : log(M/Msun) = %g' %(mmax,))
```

```
for characteristic e=2.99/3 : log(M/Msun) = 5.6
```



For the problem in the question, if we use $e = 2/3$, most contribution comes from $M/M_\odot \sim 10^7$.

One clear thing to note is that the two timescales that dominate the physics is mass segregation and inspiral timescales. Usually, interaction timescale is too short to make any difference. Also note that the combined timescale is nowhere shorter than the Hubble time for $e =< e >= 2/3$. This essentially means that mergers usually come from the highly eccentric part of the distribution where the inspiral timescale is significantly lower. This also tells you why clusters usually process the heaviest BHs first and then go down on mass.

A more rigorous way to do this would be to run Monte Carlo of binary semimajor axis and eccentricities drawn from distributions and then calculate the distribution of the combined timescales as above individually. Then one can obtain a distribution of merger timescales. One can choose the subset that has merger timescales below Hubble time and count the numbers. One can then plot the number of such mergers as a function of the total cluster mass. However, this is beyond the scope of this course and requires significant simulations.

If interested, please read the following paper for a detailed semi-analytic analysis of the same problem based on numerical results.

The Star Clusters That Make Black Hole Binaries across Cosmic Time