HW 3 - Astrophysical Dynamics

a)

Defining variables

In[13]:=
```
n = 128 (*Number of zones = n^3*);
G = 6.674 * 10^-11 (*Gravitational constant*);
M = 10^12 * 1.99 * 10^30 (*Mass of sphere*);
pc = 3.086 * 10^16 (*Parsecs to meters*);
R = 30 * 10^3 pc (*Radius of the sphere*);
L = 90 * 10^3 pc (*Length of the box*);
```

Green’s function

In[19]:=
```
fG = If[#1 == #2 == #3 == 1, 0,
    -L^2 G / (n^2 (Min[#1, 2 n + 1 - #1]^2 + Min[#2, 2 n + 1 - #2]^2 +
          Min[#3, 2 n + 1 - #3]^2)^.5)] &;
mG = Array[fG, {2 n, 2 n, 2 n}];
```

The array mG stores the values of the the Green’s function in real space.

Fourier transform of the Green’s function

In[21]:=
```
FmG = Fourier[mG];
```

The array FmG stores the values of the the Green’s function in fourier space.

b)

Density function

In[22]:=
```
P = If[(#1 - L/2)^2 + (#2 - L/2)^2 + (#3 - L/2)^2 ≤ R^2, M/(4/3 Pi R^3), 0] &;
mn = ConstantArray[0, {2 n, 2 n, 2 n}];
mn[[1 ;; n, 1 ;; n, 1 ;; n]] =
  Table[P[i L/n, j L/n, k L/n], {i, 1, n}, {j, 1, n}, {k, 1, n}];
```

The array mn stores the values of the density in real space.

Fourier transform of density function

In[25]:=
```
Fmn = Fourier[mn];
```

Fmn is the fourier transform of the density array.

Inverse fourier transform of the product

In[26]:=
```
mSol = Re[InverseFourier[Fmn * FmG]][[1 ;; n, 1 ;; n, 1 ;; n]];
```

The array mSol now contains the values of gravitational potential.

Analytical solution

In[27]:=
```
V[r_] := If[Abs[r] < R, -G M (3 R^2 - r^2) / (2 R^3), -G M / Abs[r]]
```

Plot of analytic vs numerical solution along x-axis

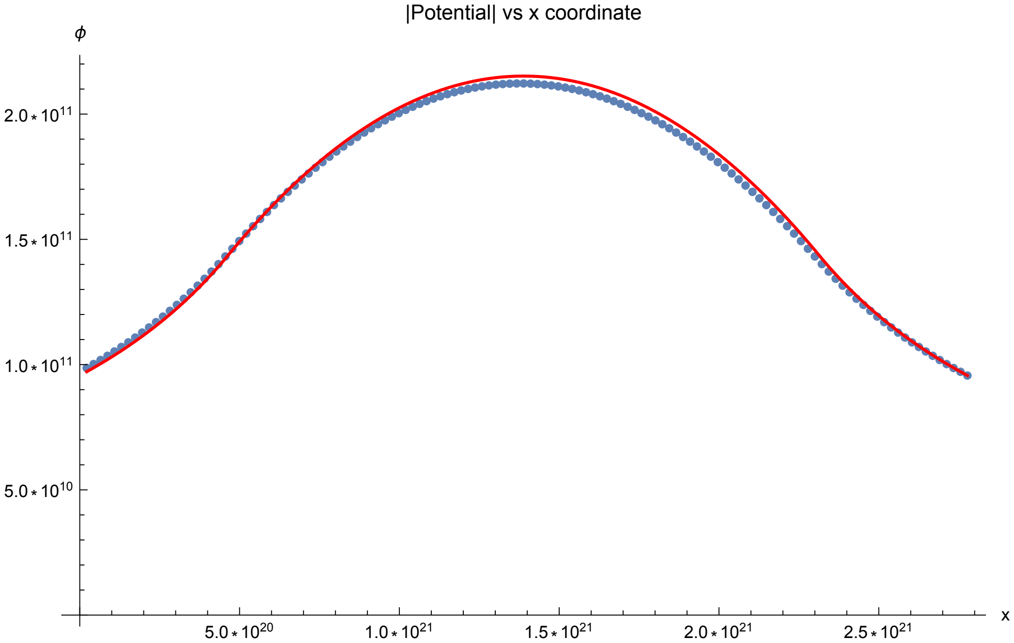
In[28]:=
```
A = mSol[[1 ;; n, Floor[n/2], Floor[n/2]]];
B = V /@ (Range[n] L / n - L/2); X = Range[n] L / n;
```

The numerical solution only matches with the analytical solution after multiplying by a constant factor (4177.68) due to some unknown reason. The numerical solution is scaled by this constant for the following plots.

Potential along the x-axis passing through the center of the sphere

In[29]:=
```
Show[ListPlot[Transpose[{X, -A Max[B] / Max[A]}],
  AxesLabel → {HoldForm["x"], HoldForm["ϕ"]},
  PlotLabel → HoldForm["|Potential| vs x coordinate"]],
 ListLinePlot[Transpose[{X, -B}], Frame → True, PlotStyle → Red]]
```

Out[29]=



Averaging radial potential

Calculating the zones within shells of radius 2jL/n and 2L(j+1)/n

In[30]:=
```
grid = Flatten[
   Table[{Floor[Sqrt[1.0 (i - n / 2 + .5) ^2 + (j - n / 2 + .5) ^2 + (k - n / 2 + .5) ^2]],
     mSol[[i, j, k]]}, {i, 1, n}, {j, 1, n}, {k, 1, n}], 2];
```

Calculating average potential of all zones within a shell

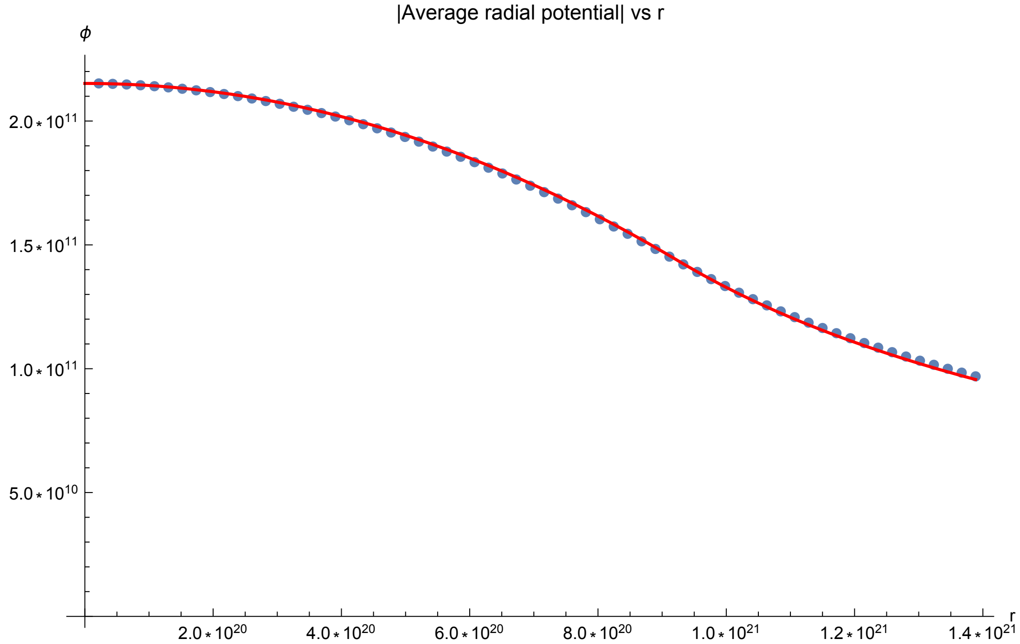
In[31]:=
```
Vr = Table[Mean[Select[grid, #[[1]] == j &][[All, 2]]], {j, 1, n / 2}];
```

The array Vr now contains the average radial potential measured over steps of 2L/n.

Plot of analytic vs numerical solution of average radial potential

In[36]:=
```
Show[ListPlot[Transpose[{Range[Length[Vr]] L / n, Abs[Vr] (-V[0] / Max[Abs[Vr]])}],
   AxesLabel → {HoldForm["r"], HoldForm["ϕ"]},
   PlotLabel → HoldForm["|Average radial potential| vs r"]],
  Plot[-V[x], {x, 0, L / 2}, PlotStyle → Red]]
```

Out[36]=



c)

Defining constants

In[70]:=
```
pc = 3.086 * 10^16 (*Parsecs to meters*);
n = nxy = 128 (*Number of zones along x and y*);
nz = n; (*Number of zones along z*);
G = 6.674 * 10^-11 (*Gravitational constant*);
Lxy = 30 * 10^3 pc (*Length of the box along x and y directions*);
Lz = 1 * 10^3 pc (*Length of the box along z direction*);
```

In[33]:=
```
p0 = 8 * 10^10 * 1.99 * 10^30 / (10^3 pc) ^3;
r0 = 10^3 pc;
h = .1 * 10^3 pc;
```

Green’s function

In[36]:=
```
fkG = If[#1 == #2 == #3 == 1, 0,
    -G Lxy / n Lxy / n Lz / nz / ( (Lxy^2 / n^2 Min[#1, 2 n + 1 - #1] ^2 + Lxy^2 / n^2
          Min[#2, 2 n + 1 - #2] ^2 + Lz^2 / nz^2 Min[#3, 2 nz + 1 - #3] ^2) ^.5)] &;
mkG = Array[fkG, {2 n, 2 n, 2 nz}];
```

The array mkG stores the values of the the Green’s function in real space.

Fourier transform of the Green’s function

In[38]:=
```
FmkG = Fourier[mkG];
```

The array FmkG stores the values of the the Green’s function in fourier space.

Density function

In[39]:=
```
Pk = p0 (1 + ((#1 - Lxy / 2) ^2 + (#2 - Lxy / 2) ^2) / r0^2) ^ (-3 / 2)
    Exp[-Abs[#3 - Lz / 2] / h] &;
mkn = ConstantArray[0, {2 n, 2 n, 2 nz}];
mkn[[1 ;; n, 1 ;; n, 1 ;; nz]] =
  Table[Pk[i Lxy / n, j Lxy / n, k Lz / nz], {i, 1, n}, {j, 1, n}, {k, 1, nz}];
```

The array mkn stores the values of the density in real space.

Fourier transform of density function

In[42]:=
```
Fmkn = Fourier[mkn];
```

Fmkn is the fourier transform of the density array.

Inverse fourier transform of the product

In[43]:=
```
mkSol = Re[InverseFourier[Fmkn * FmkG]][[1 ;; n, 1 ;; n, 1 ;; nz]];
```

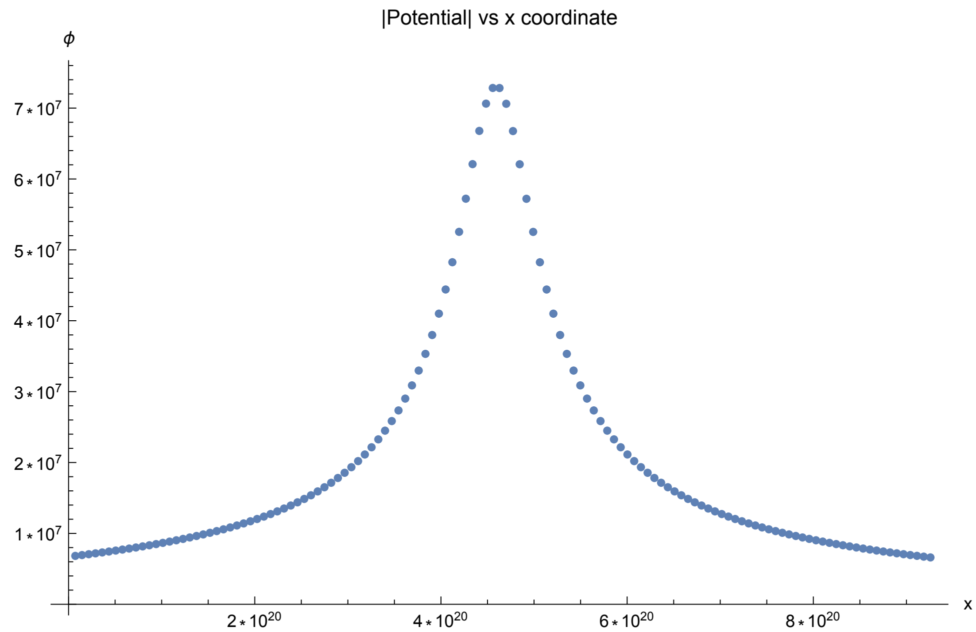
The array mkSol now contains the values of gravitational potential.

Plot of potential along x-axis passing through the center of the disk

In[44]:=
```
S = mkSol[[n/2, 1 ;; n, nz/2]]; X = Range[n] Lxy / n;
```

In[45]:=
```
ListPlot[Transpose[{X, -S}], AxesLabel → {HoldForm["x"], HoldForm["ϕ"]},
 PlotLabel → HoldForm["|Potential| vs x coordinate"]]
```

Out[45]=



Finding average radial potential on the disk

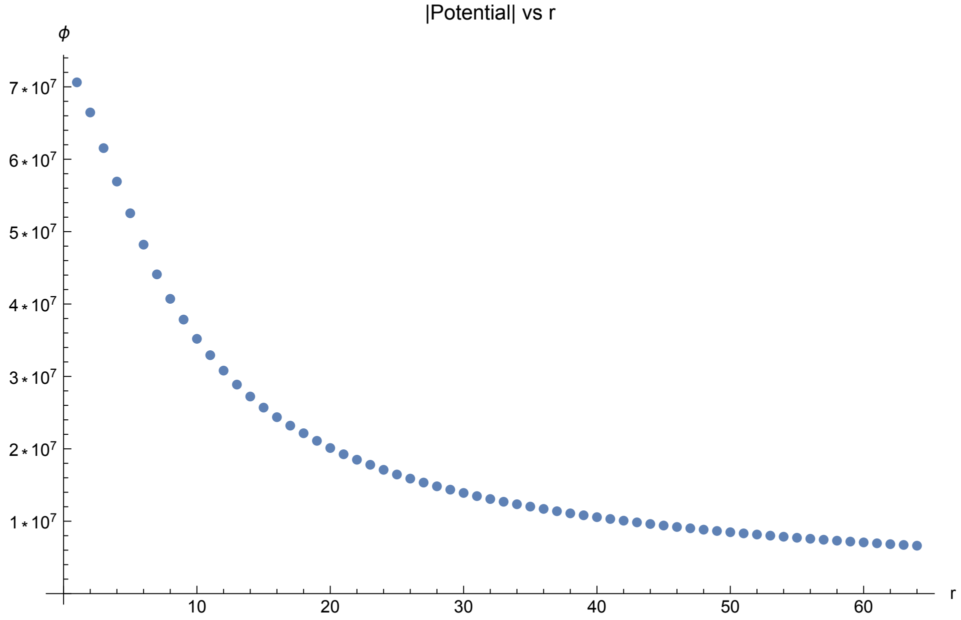
In[46]:=
```
gridk = Flatten[Table[{Floor[Sqrt[1.0 (i - n/2 + .5)^2 + (j - n/2 + .5)^2]],
    mkSol[[i, j, nz/2]]}, {i, 1, n}, {j, 1, n}], 1];
```

Calculating average potential of all zones within a shell

In[47]:=
```
Vkr = Table[Mean[Select[gridk, #[[1]] == j &][[All, 2]]], {j, 1, n/2}];
```

```
In[48]:= ListPlot[-Vkr, AxesLabel → {HoldForm["r"], HoldForm["ϕ"]},
          PlotLabel → HoldForm["|Potential| vs r"]]
```

Out[48]=



Finding an interpolating function passing through these points

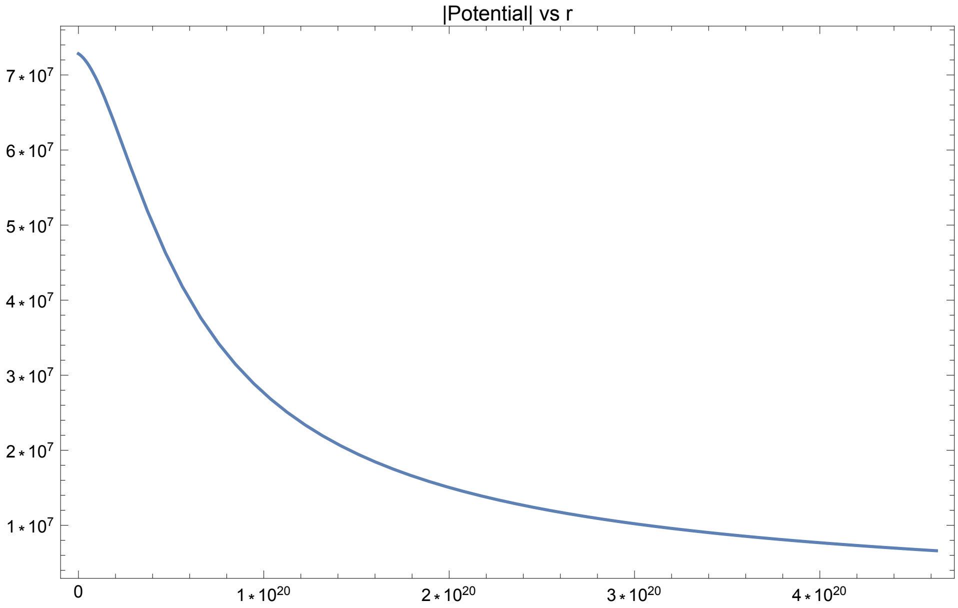
Interpolation to find derivatives

```
In[49]:= fR = Interpolation[Transpose[
            {(Range[n/2, n] - n/2) Lxy/n, -mkSol[[Floor[n/2] ;; n, n/2, nz/2]]}]]

Out[49]= InterpolatingFunction[ Domain: {{0., 4.63*10^20}}
                                Output: scalar ]
```

In[50]:=
```
Plot[fR[x], {x, 0., 4.629 * 10^20}, Frame → True,
 AxesLabel → {HoldForm["r"], HoldForm["ϕ"]},
 PlotLabel → HoldForm["|Potential| vs r"]]
```

Out[50]=



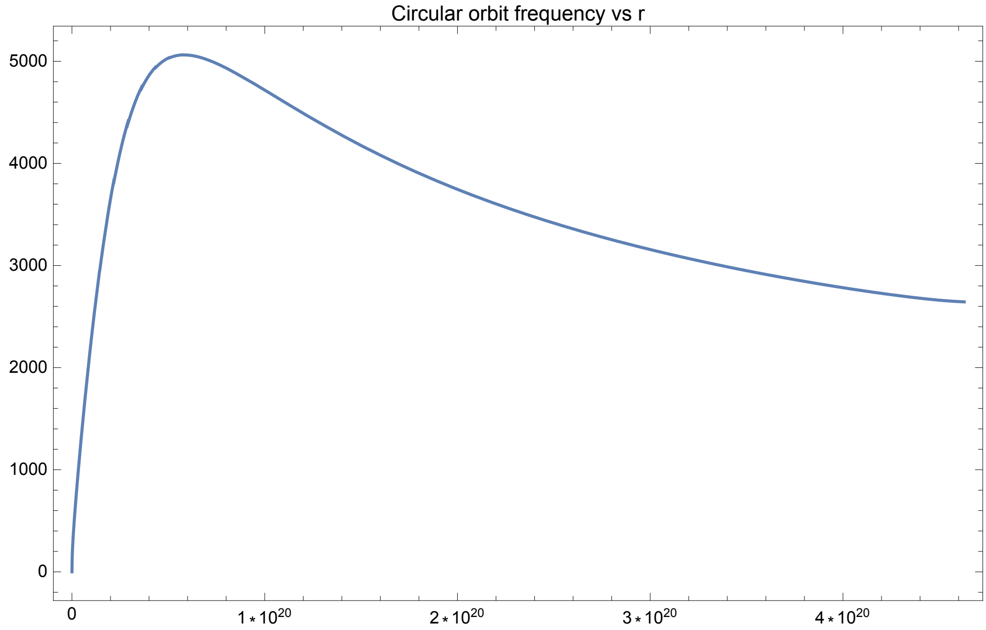
Circular orbit frequency

In[51]:=
```
wc[R_] = Sqrt[R D[-fR[R], R]]
```

Out[51]= $\sqrt{\left(-R\ \text{InterpolatingFunction}\left[\ \boxed{\text{Domain: } \{\{0., 4.63*10^{20}\}\}\ \text{Output: scalar}}\ \right][R]\right)}$

In[52]:=
```
Plot[wc[x], {x, 0.`, 4.629`*^20}, Frame → True,
 AxesLabel → {HoldForm["r"], HoldForm["ϕ"]},
 PlotLabel → HoldForm["Circular orbit frequency vs r"]]
```

Out[52]=



Epicyclic frequency

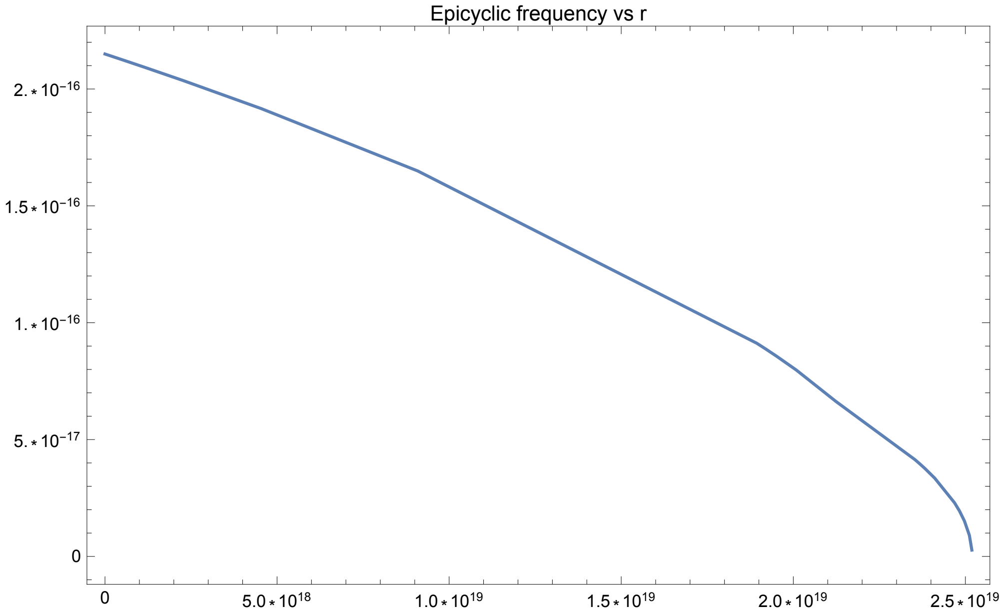
The epicyclic frequency k(R) is given by:

In[53]:=
```
k[R_] = Sqrt[ D[-fR[R], {R, 2}]]
```

Out[53]=
```
√( -InterpolatingFunction[ Domain: {{0., 4.63*10^20}} Output: scalar ][R] )
```

In[63]:=
```
Plot[{k[x]}, {x, 0.`, 4.629`*^20}, Frame → True,
 AxesLabel → {HoldForm["r"], HoldForm["ϕ"]},
 PlotLabel → HoldForm["Epicyclic frequency vs r"], PlotRange → All]
```

Out[63]=



Corotation radius

Ω_b = wb

In[64]:=
```
wb = 10^-7 / (365 24 * 60 * 60.0)
```

Out[64]= $3.170979198 * 10^{-15}$

The corotation radius (Rc) is that value of R at which k(R)=wb

Inner Lindbald resonance occurs when k(R)>wb => R<Rc

Outer Lindbald resonance occurs when k(R)<wb => R>Rc