15

$$\frac{d\underline{x}}{dt} = \underline{v} \qquad \frac{d\underline{v}}{dt} = \frac{-\gamma M}{|\underline{x}|^3}\,\underline{x}$$

a) $$\frac{\underline{x}(t+\Delta t) - \underline{x}(t)}{\Delta t} = \underline{v}$$

$$\boxed{\underline{x}(t+\Delta t) = \underline{v}\cdot\Delta t + \underline{x}(t)}$$

$|x| = \sqrt{x_1^2 + x_2^2 + x_3^2}$

$$\frac{\underline{v}(t+\Delta t) - \underline{v}(t)}{\Delta t} = \frac{-\gamma M\,\underline{x}(t)}{|\underline{x}(t)|^3}$$

$$\boxed{\underline{v}(t+\Delta t) = -\frac{\gamma M\,\underline{x}(t)}{|\underline{x}(t)|^3}\,\Delta t + \underline{v}(t)}$$

b) $$\underline{x}(t+\Delta t) = \frac{\Delta t}{2}\left(\underline{v}(t) + \frac{-\gamma M\,\underline{x}(t)}{|\underline{x}(t)|^3}\Delta t + \underline{v}(t)\right) + \underline{x}(t)$$

$$\underline{v}(t+\Delta t) = -\frac{\Delta t}{2}\,\frac{\gamma M}{|\underline{x}(t)|^3}\left(\underline{x}(t) + \underline{v}\,\Delta t + \underline{x}(t)\right) + \underline{v}(t)$$

c) $$E = \frac{1}{2} m v_0^2 - \frac{\gamma M m}{d}$$

$$\longrightarrow v_0 = \pm\sqrt{\left(E + \frac{\gamma M m}{d}\right)\frac{2}{m}}$$

i) $E > 0 \qquad v_0 > 42128.23 \frac{m}{s}$

ii) $E = 0 \qquad v_0 = 42128.23 \frac{m}{s}$

iii) $\frac{-\gamma M_m}{2d} < E < 0 \qquad 29,789.16 \frac{m}{s} < v_0 < 42128.23 \frac{m}{s}$

iv) $E = \frac{-\gamma M_m}{2d} \qquad v_0 = 29789.16 \frac{m}{s}$

e) $$\underline{x}(t+\Delta t) = \underline{x}(t) + \Delta t \left(\frac{-\gamma M \underline{x}(t)}{|\underline{x}(t)|^3} \frac{\Delta t}{2} + \underline{v}(t) \right)$$

$$\underline{v}(t+\Delta t) = \underline{v}(t) - \frac{\gamma M}{|\underline{x}(t)|^3} \Delta t \left(\underline{x}(t) + \frac{\Delta t}{2} \underline{v}(t) \right)$$

f) $u_1 = x \qquad u_4 = v_x$

$u_2 = y \qquad u_5 = v_y$

$u_3 = z \qquad u_6 = v_z$

$$f_i = \begin{pmatrix} \dot{u}_1 \\ \dot{u}_2 \\ \dot{u}_3 \\ \dot{u}_4 \\ \dot{u}_5 \\ \dot{u}_6 \end{pmatrix} = \begin{pmatrix} u_4 \\ u_5 \\ u_6 \\ -\frac{\gamma M u_1}{\sqrt{u_1^2 + u_2^2 + u_3^2}^3} \\ -\frac{\gamma M u_2}{z^3} \\ -\frac{\gamma M u_3}{z^3} \end{pmatrix}$$

for simplicity
$z = \sqrt{u_1^2 + u_2^2 + u_3^2}$

$$f = \begin{pmatrix} u_4 + \Delta t \left(-\dfrac{\gamma M u_1}{z^3} \right) \\ u_5 + \Delta t \left(-\dfrac{\gamma M u_2}{z^3} \right) \\ u_6 + \Delta t \left(-\dfrac{\gamma M u_3}{z^3} \right) \\ -\dfrac{\gamma M u_1}{z^3} - \dfrac{\Delta t \, \gamma M}{z^3} u_4 \\ -\dfrac{\gamma M u_2}{z^3} - \dfrac{\Delta t \, \gamma M}{z^3} u_5 \\ -\dfrac{\gamma M u_3}{z^3} - \dfrac{\Delta t \, \gamma M}{z^3} u_6 \end{pmatrix}$$

Untitled

November 9, 2017

1 15. Celestial mechanics

In [1]:
```
using PyPlot
```

In [2]:
```
d=1.496e11 #[m]
m=5.972e24 #[kg]
M=1.9891e30 #[kg]
g=6.67408e-11 #[m^3 kg^-1 s^-2]
E=-g*M*m/(2*d)
t0=3600 #1hour, also dt
tend=t0*24*365*5 #5*365days
rsun=6.957e8 # [m]
v0=sqrt((E+g*M*m/d)*2./m)
```

Out[2]: 29789.157181120405

1.1 b)

In [3]:
```
function predictor(d,m,M,g,v,x,dt,t)
    steps=floor(Int,t/dt)
    memoryx=zeros(steps+1,3)
    memoryv=zeros(steps+1,3)
    memoryx[1,:]=x
    memoryv[1,:]=v
    for i in 1:steps
        #t values
        xi=x
        vi=v
        #t+1
        x=dt/2*(vi+-g*M*xi*dt/norm(xi)^3+vi)+xi
        v=-dt/2*g*M/norm(xi)^3*(xi+vi*dt+xi)+vi
        memoryx[i+1,:]=x
        memoryv[i+1,:]=v
    end
    #plot
    i= 1:steps+1
    figure(1,figsize=(8,8))
    title(L"v_0="*string(memoryv[1,2]))
```

```
        ylabel(L"$y$ $[AU]$")
        xlabel(L"$x$ $[AU]$")
        plot(memoryx[i,1]/d,memoryx[i,2]/d,linewidth=1,label="trajectory")

        #displaying the sun in the center, with its proper ratio
        f2(x)=rsun*sqrt(abs(1.-x.^2.))
        x=-1:0.001:1
        fill_between(x*rsun/d,0,f2(x)/d,color="red")
        plot(x[5]*rsun/d,f2(x[5])/d,linewidth=1,color="red",label="sun")
        fill_between(x*rsun/d,0,-f2(x)/d,color="red")
        legend()
    end
```

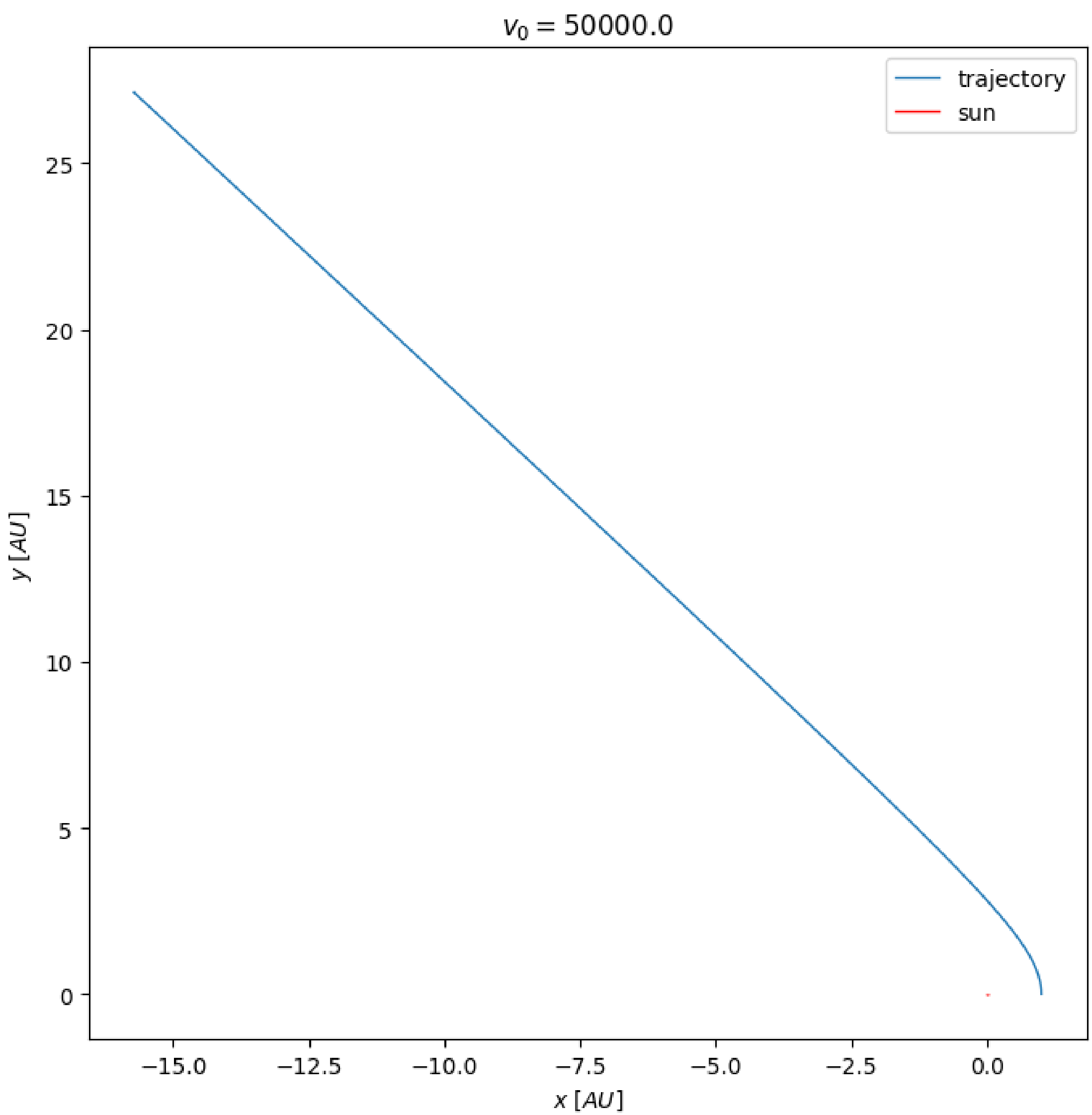
```
Out[3]: predictor (generic function with 1 method)
```

1.2 c)

For the values of v_0 take a look at my calculations

1.3 i)

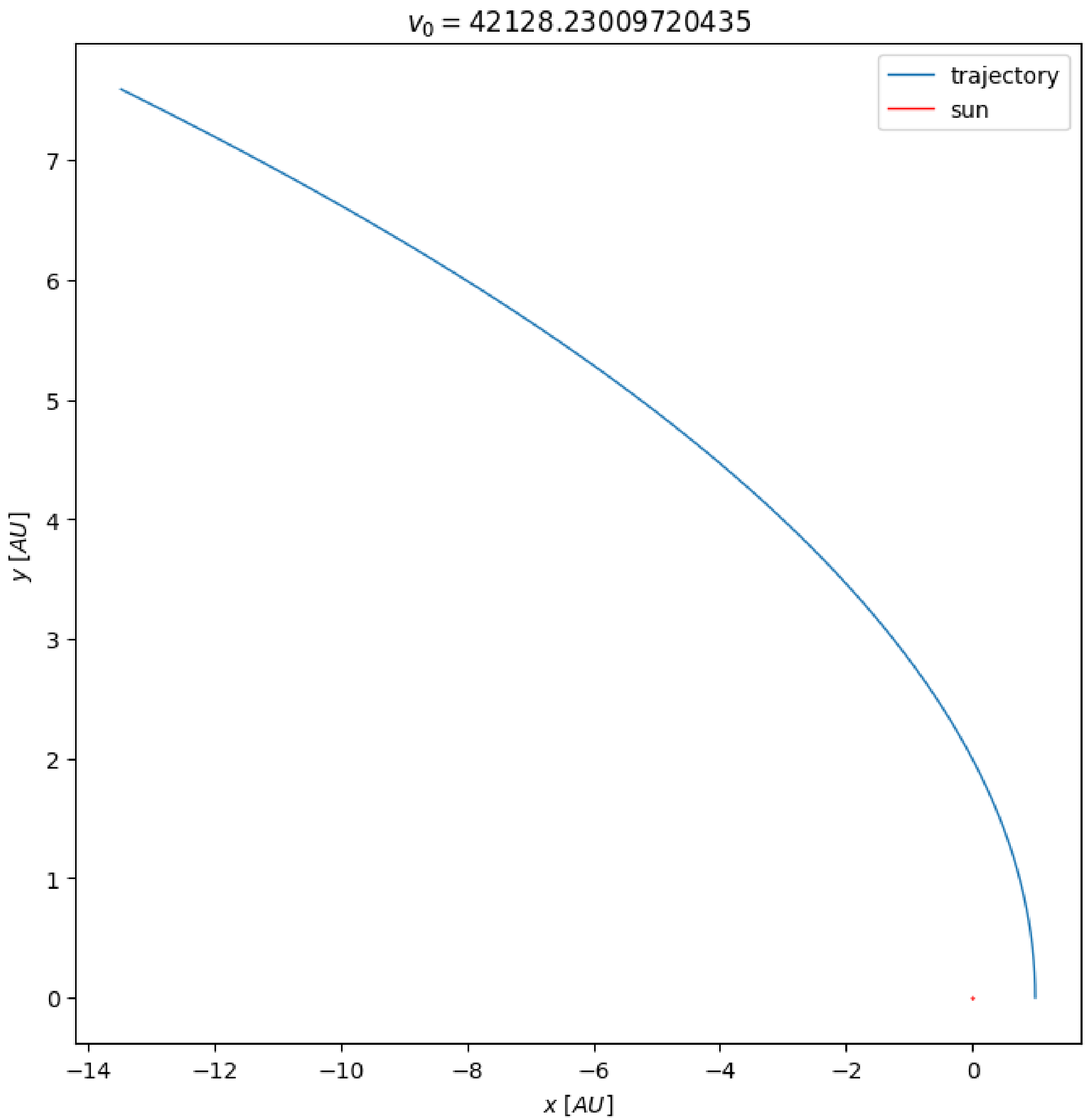
```
In [4]: x=[d, 0., 0.]
        v=[0., 50000, 0]
        predictor(d,m,M,g,v,x,t0,tend)
```



Out[4]: PyObject <matplotlib.legend.Legend object at 0x7fed708f4a50>

1.4 ii)

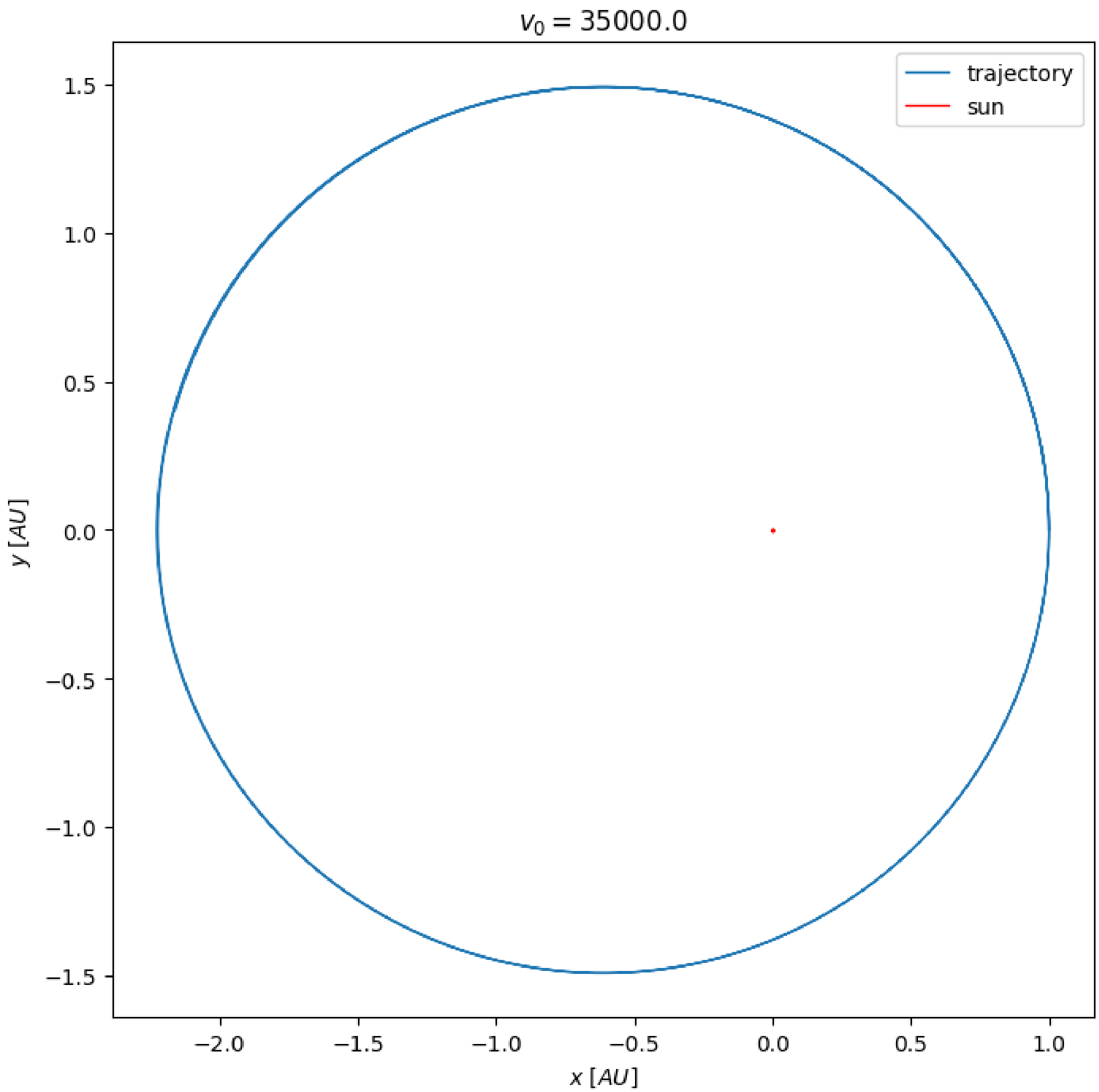
In [5]:
```
x=[d, 0., 0.]
E=0
v0=sqrt((E+g*M*m/d)*2./m)
v=[0., v0, 0]
predictor(d,m,M,g,v,x,t0,tend)
```



Out[5]: PyObject <matplotlib.legend.Legend object at 0x7fed70549410>

2 iii)

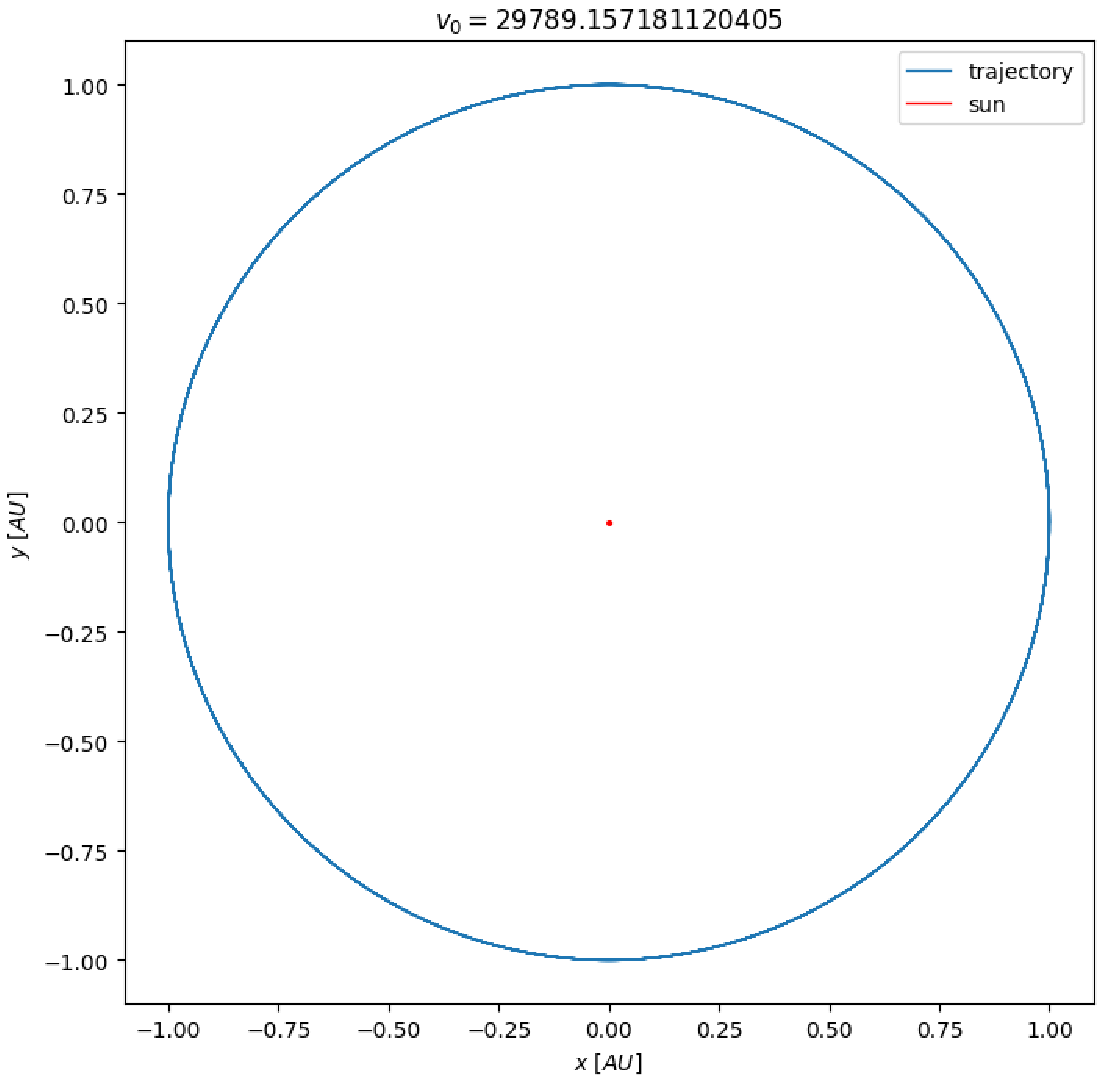
In [6]:
```
x=[d, 0., 0.]
v=[0., 35000, 0]
predictor(d,m,M,g,v,x,t0,tend)
```



Out[6]: PyObject <matplotlib.legend.Legend object at 0x7fed70432950>

3 iv)

In [7]:
```
x=[d, 0., 0.]
E=-g*M*m/(2*d)
v0=sqrt((E+g*M*m/d)*2./m)
v=[0., v0, 0]
predictor(d,m,M,g,v,x,t0,tend)
```



Out[7]: PyObject <matplotlib.legend.Legend object at 0x7fed70397110>

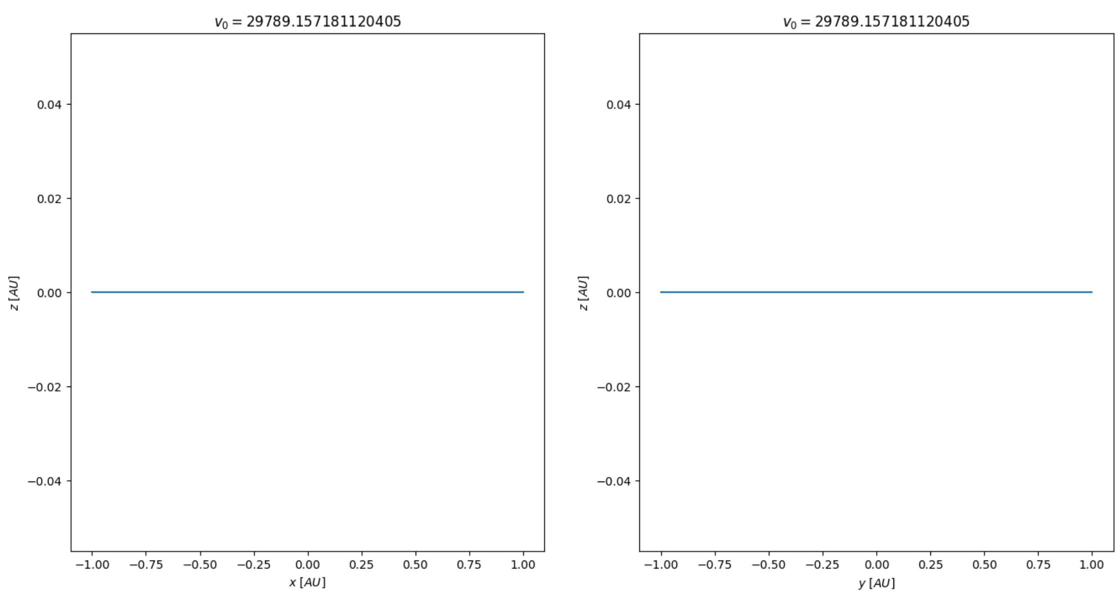
3.1 d)

In [8]:
```
#same as predictor, but shows x-z and y-z plane
function predictor2(d,m,M,g,v,x,dt,t)
    steps=floor(Int,t/dt)
    memoryx=zeros(steps+1,3)
    memoryv=zeros(steps+1,3)
    memoryx[1,:]=x
    memoryv[1,:]=v
    for i in 1:steps
        xi=x
        vi=v
```

```
        x=dt/2*(vi+-g*M*xi*dt/norm(xi)^3+vi)+xi
        v=-dt/2*g*M/norm(xi)^3*(xi+vi*dt+xi)+vi
        memoryx[i+1,:]=x
        memoryv[i+1,:]=v
    end
    i= 1:steps+1
    figure(1,figsize=(16,8))
    z=121
    subplot(z)
    title(L"v_0="*string(memoryv[1,2]))
    ylabel(L"$z$ $[AU]$")
    xlabel(L"$x$ $[AU]$")
    plot(memoryx[i,1]/d,memoryx[i,3]/d,linewidth=1,label="trajectory")



    subplot(z+1)
    title(L"v_0="*string(memoryv[1,2]))
    ylabel(L"$z$ $[AU]$")
    xlabel(L"$y$ $[AU]$")
    plot(memoryx[i,2]/d,memoryx[i,3]/d,linewidth=1,label="trajectory")

end
```

Out[8]: predictor2 (generic function with 1 method)

In [9]:
```
x=[d, 0., 0.]
E=-g*M*m/(2*d)
v0=sqrt((E+g*M*m/d)*2./m)
v=[0., v0, 0]
predictor2(d,m,M,g,v,x,t0,tend)
```



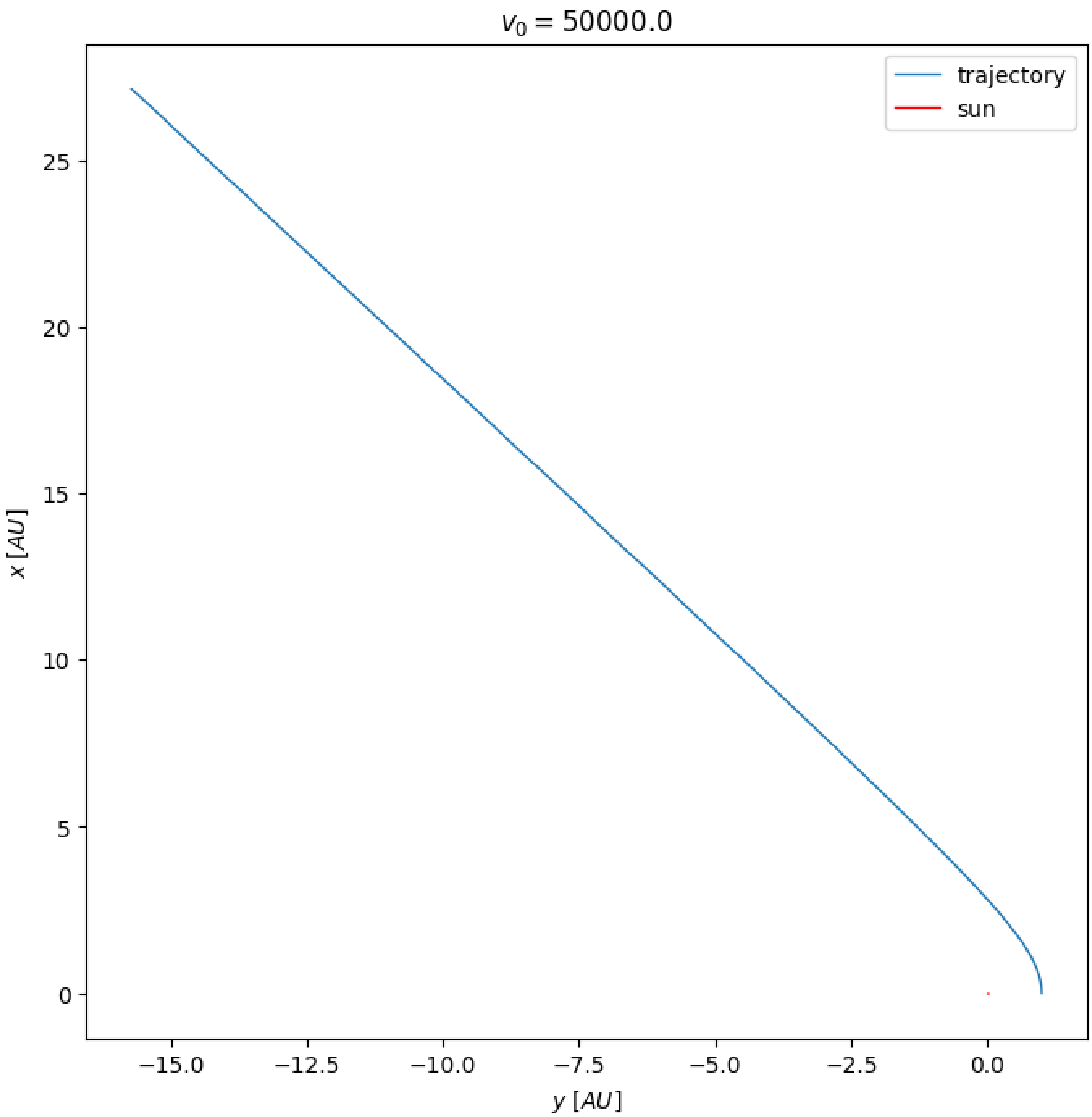
```
Out[9]: 1-element Array{Any,1}:
         PyObject <matplotlib.lines.Line2D object at 0x7fed70208110>
```

3.2 e)

```
In [10]: function rk2(d,m,M,g,v,x,dt,t)
             steps=floor(Int,t/dt)
             memoryx=zeros(steps+1,3)
             memoryv=zeros(steps+1,3)
             memoryx[1,:]=x
             memoryv[1,:]=v
             for i in 1:steps
                 #t values
                 xi=x
                 vi=v
                 #t+1 as is in the script
                 x=xi+dt*(-g*M*xi/norm(xi)^3*dt/2+vi)
                 v=vi-g*M/norm(xi)^3*dt*(xi+dt/2*vi)
                 memoryx[i+1,:]=x
                 memoryv[i+1,:]=v
             end
             i= 1:steps+1
             figure(1,figsize=(8,8))
             title(L"v_0="*string(memoryv[1,2]))
             ylabel(L"$x$ $[AU]$")
             xlabel(L"$y$ $[AU]$")
             plot(memoryx[i,1]/d,memoryx[i,2]/d,linewidth=1,label="trajectory")

             #displaying the sun in the center
             f2(x)=rsun*sqrt(abs(1.-x.^2.))
             x=-1:0.001:1
             fill_between(x*rsun/d,0,f2(x)/d,color="red")
             plot(x[5]*rsun/d,f2(x[5])/d,linewidth=1,color="red",label="sun")
             fill_between(x*rsun/d,0,-f2(x)/d,color="red")
             legend()
         end

Out[10]: rk2 (generic function with 1 method)
```

3.3 i)

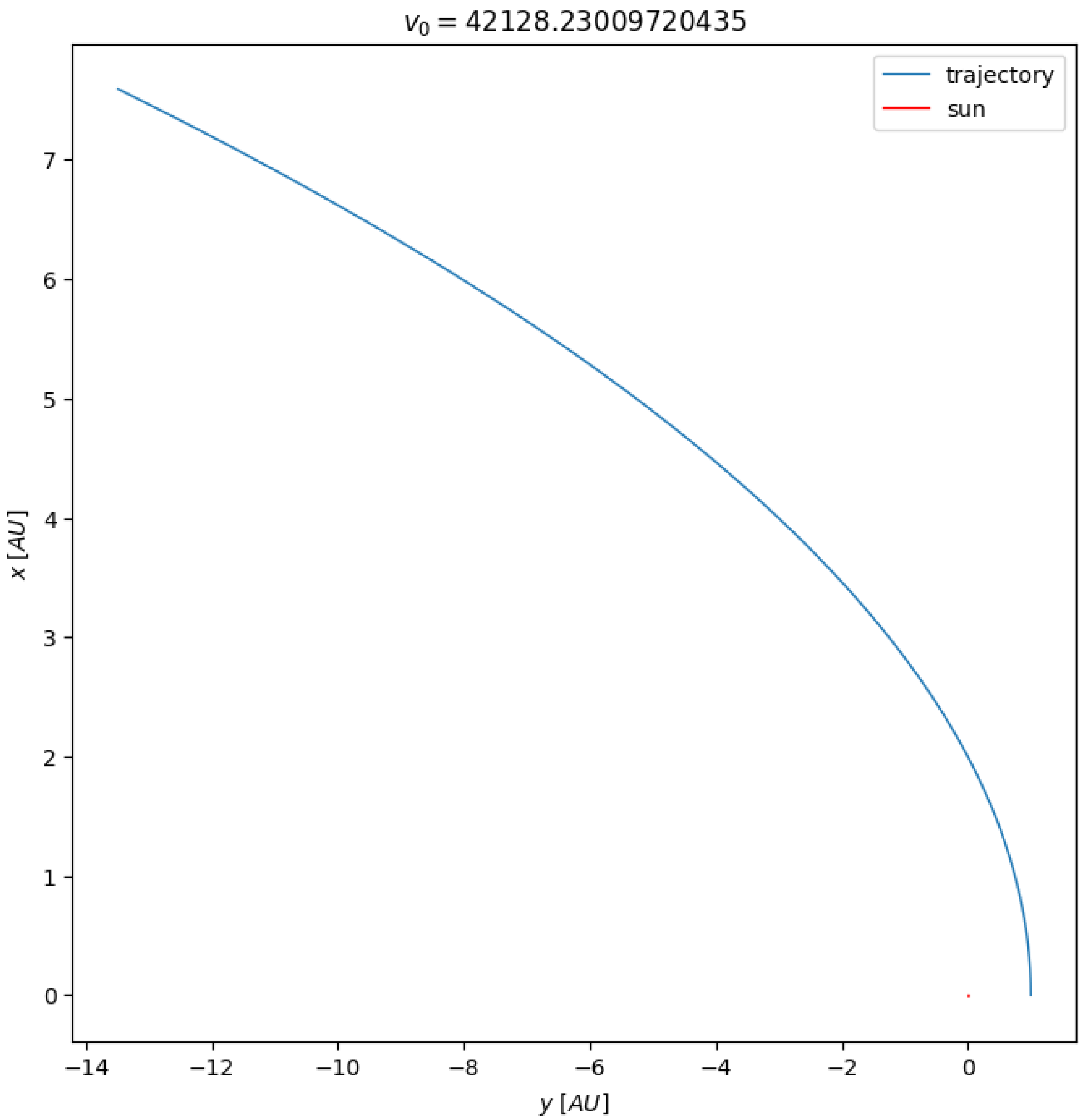
```
In [11]: x=[d, 0., 0.]
         v=[0., 50000, 0]
         rk2(d,m,M,g,v,x,t0,tend)
```




```
Out[11]: PyObject <matplotlib.legend.Legend object at 0x7fed7010a110>
```

3.4 ii)

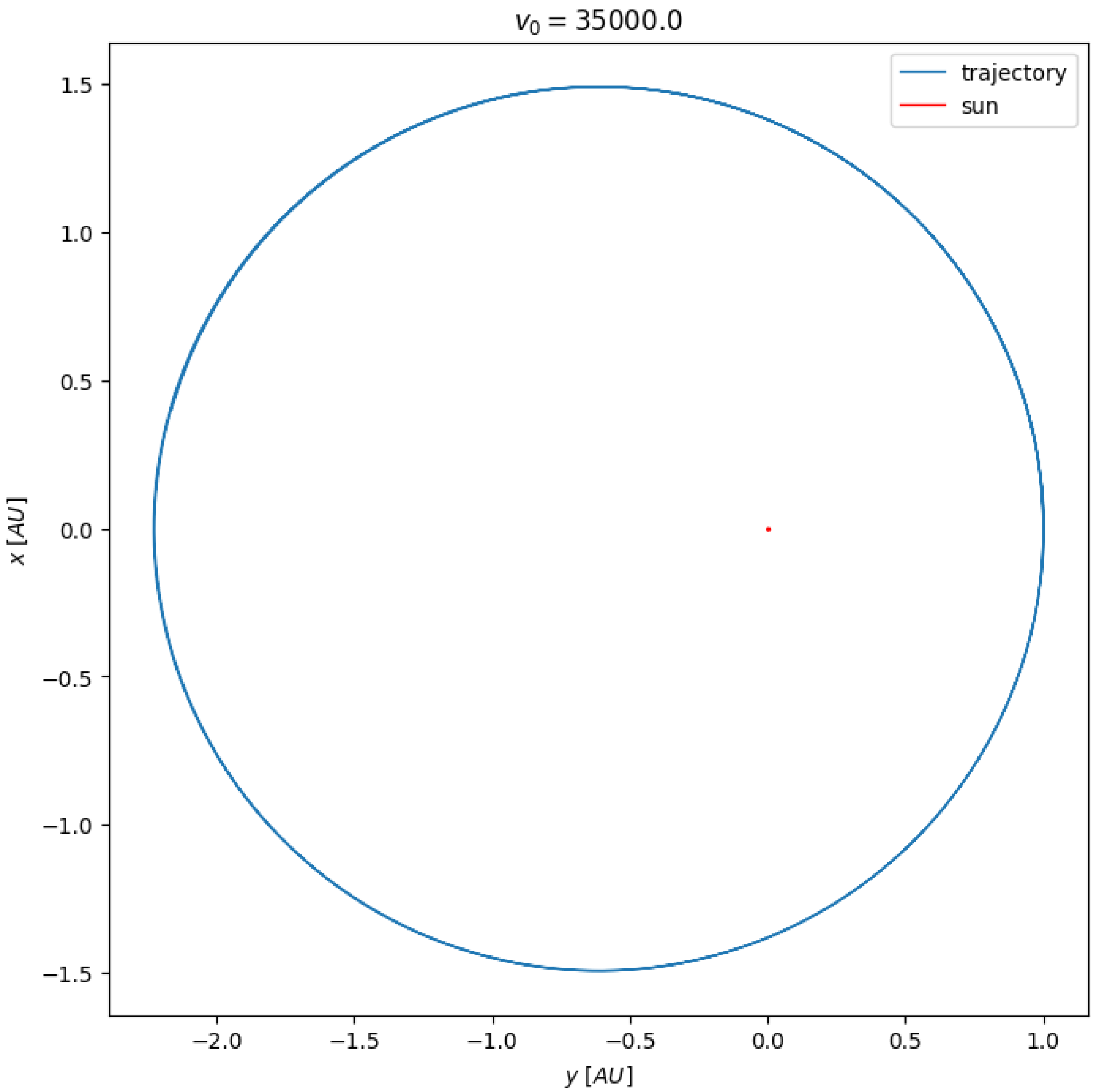
```
In [12]: x=[d, 0., 0.]
         E=0
         v0=sqrt((E+g*M*m/d)*2./m)
         v=[0., v0, 0]
         rk2(d,m,M,g,v,x,t0,tend)
```



Out[12]: PyObject <matplotlib.legend.Legend object at 0x7fed700474d0>

3.5 iii)

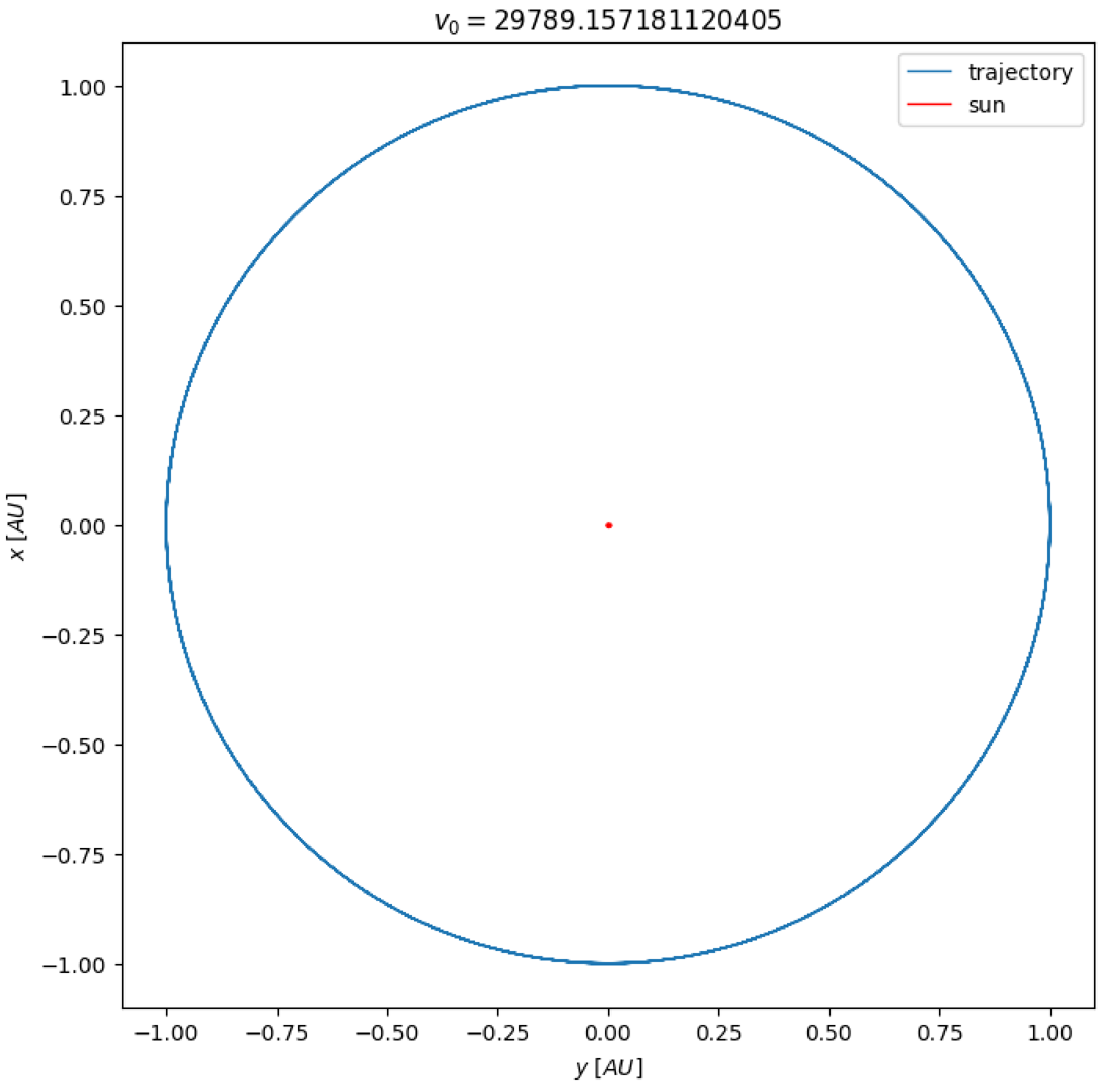
In [13]:
```
x=[d, 0., 0.]
v=[0., 35000, 0]
rk2(d,m,M,g,v,x,t0,tend)
```



Out[13]: PyObject <matplotlib.legend.Legend object at 0x7fed6ff3a190>

3.6 iv)

In [14]:
```
x=[d, 0., 0.]
E=-g*M*m/(2*d)
v0=sqrt((E+g*M*m/d)*2./m)
v=[0., v0, 0]
rk2(d,m,M,g,v,x,t0,tend)
```



Out[14]: PyObject <matplotlib.legend.Legend object at 0x7fed6fe98050>

3.7 f)

In [15]:
```
function rk4(d,m,M,g,v,x,dt,t)
    steps=floor(Int,t/dt)
    memoryx=zeros(steps+1,3)
    memoryv=zeros(steps+1,3)
    memoryx[1,:]=x
    memoryv[1,:]=v
    #function for the 2nd deriviation
    f(z)=-g*M*z/norm(x)^3
    for i in 1:steps
        #k values as on the sheet
```

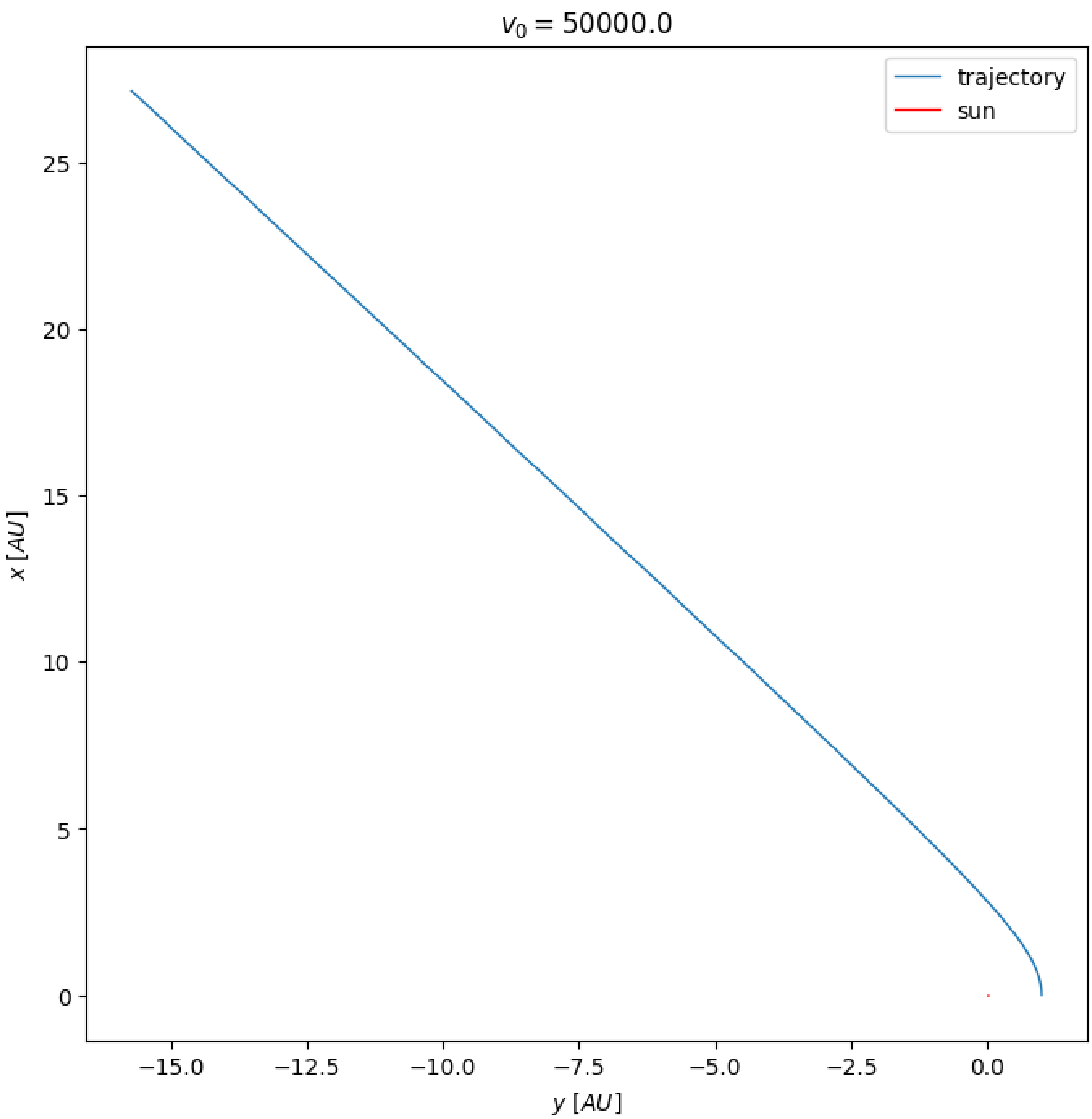
```
    k11=dt*(v[1])
    k12=dt*(v[2])
    k13=dt*(v[3])
    k14=dt*(f(x[1]))
    k15=dt*(f(x[2]))
    k16=dt*(f(x[3]))

    k21=dt*(v[1]+k14/2)
    k22=dt*(v[2]+k15/2)
    k23=dt*(v[3]+k16/2)
    k24=dt*f(x[1]+k11/2)
    k25=dt*f(x[2]+k12/2)
    k26=dt*f(x[3]+k13/2)

    k31=dt*(v[1]+k24/2)
    k32=dt*(v[2]+k25/2)
    k33=dt*(v[3]+k26/2)
    k34=dt*f(x[1]+k21/2)
    k35=dt*f(x[2]+k22/2)
    k36=dt*f(x[3]+k23/2)

    k41=dt*(v[1]+k34)
    k42=dt*(v[2]+k35)
    k43=dt*(v[3]+k36)
    k44=dt*f(x[1]+k31)
    k45=dt*f(x[2]+k32)
    k46=dt*f(x[3]+k33)

    #the corrseponding u values
    x[1]+=1/6*(k11+2*k21+2*k31+k41)
    x[2]+=1/6*(k12+2*k22+2*k32+k42)
    x[3]+=1/6*(k13+2*k23+2*k33+k43)

    v[1]+=1/6*(k14+2*k24+2*k34+k44)
    v[2]+=1/6*(k15+2*k25+2*k35+k45)
    v[3]+=1/6*(k16+2*k26+2*k36+k46)

    memoryx[i+1,:]=x
    memoryv[i+1,:]=v
end
#plot
i= 1:steps+1
figure(1,figsize=(8,8))
title(L"v_0="*string(memoryv[1,2]))
ylabel(L"$x$ $[AU]$")
xlabel(L"$y$ $[AU]$")
plot(memoryx[i,1]/d,memoryx[i,2]/d,linewidth=1,label="trajectory")
```

```
    #displaying the sun in the center
    f2(x)=rsun*sqrt(abs(1.-x.^2.))
    x=-1:0.001:1
    fill_between(x*rsun/d,0,f2(x)/d,color="red")
    plot(x[5]*rsun/d,f2(x[5])/d,linewidth=1,color="red",label="sun")
    fill_between(x*rsun/d,0,-f2(x)/d,color="red")
    legend()
end
```

Out[15]: rk4 (generic function with 1 method)

3.8 i)

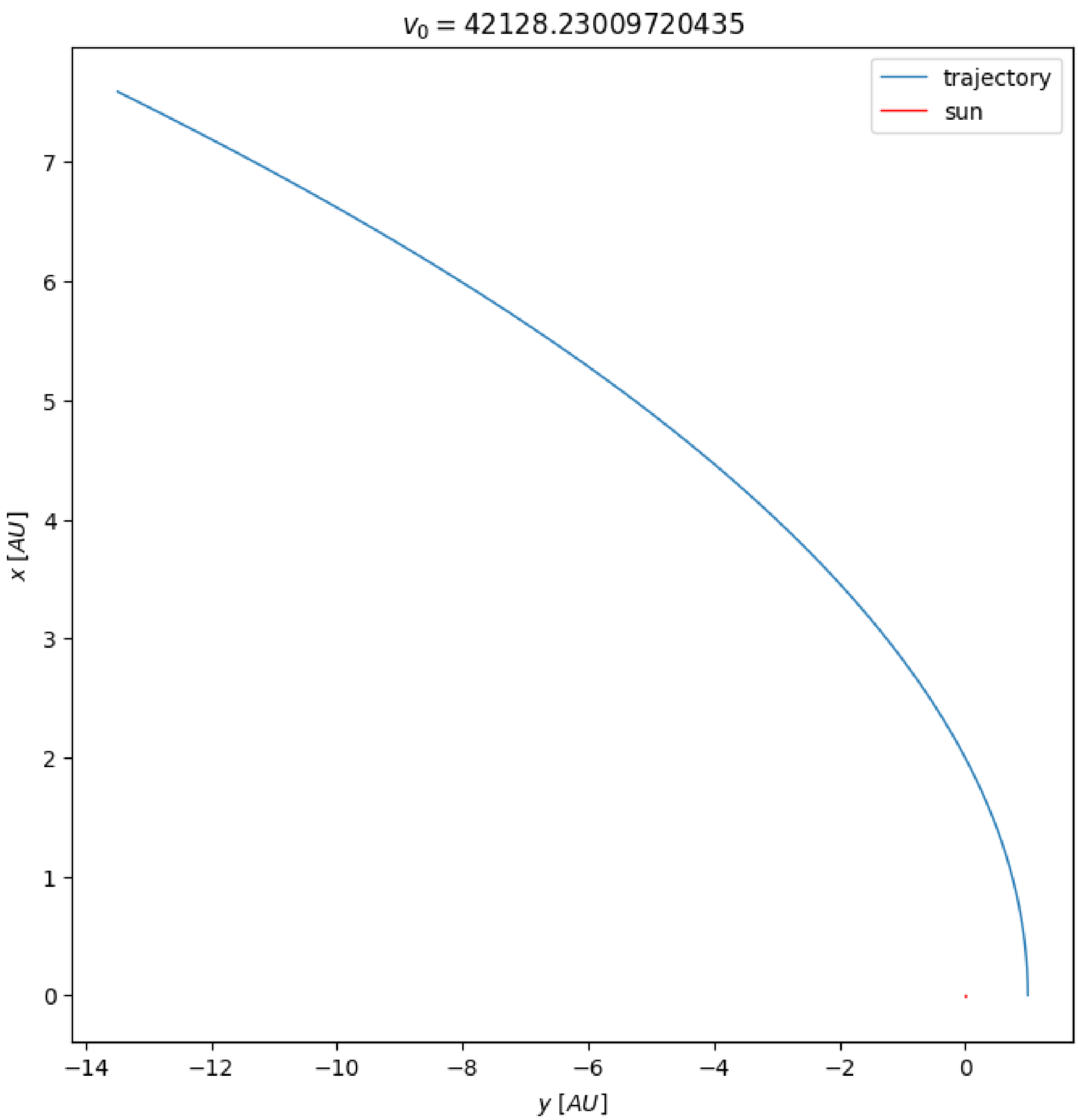
In [16]:
```
x=[d, 0., 0.]
v=[0., 50000, 0]
rk4(d,m,M,g,v,x,t0,tend)
```



Out[16]: PyObject <matplotlib.legend.Legend object at 0x7fed6fd837d0>

3.9 ii)

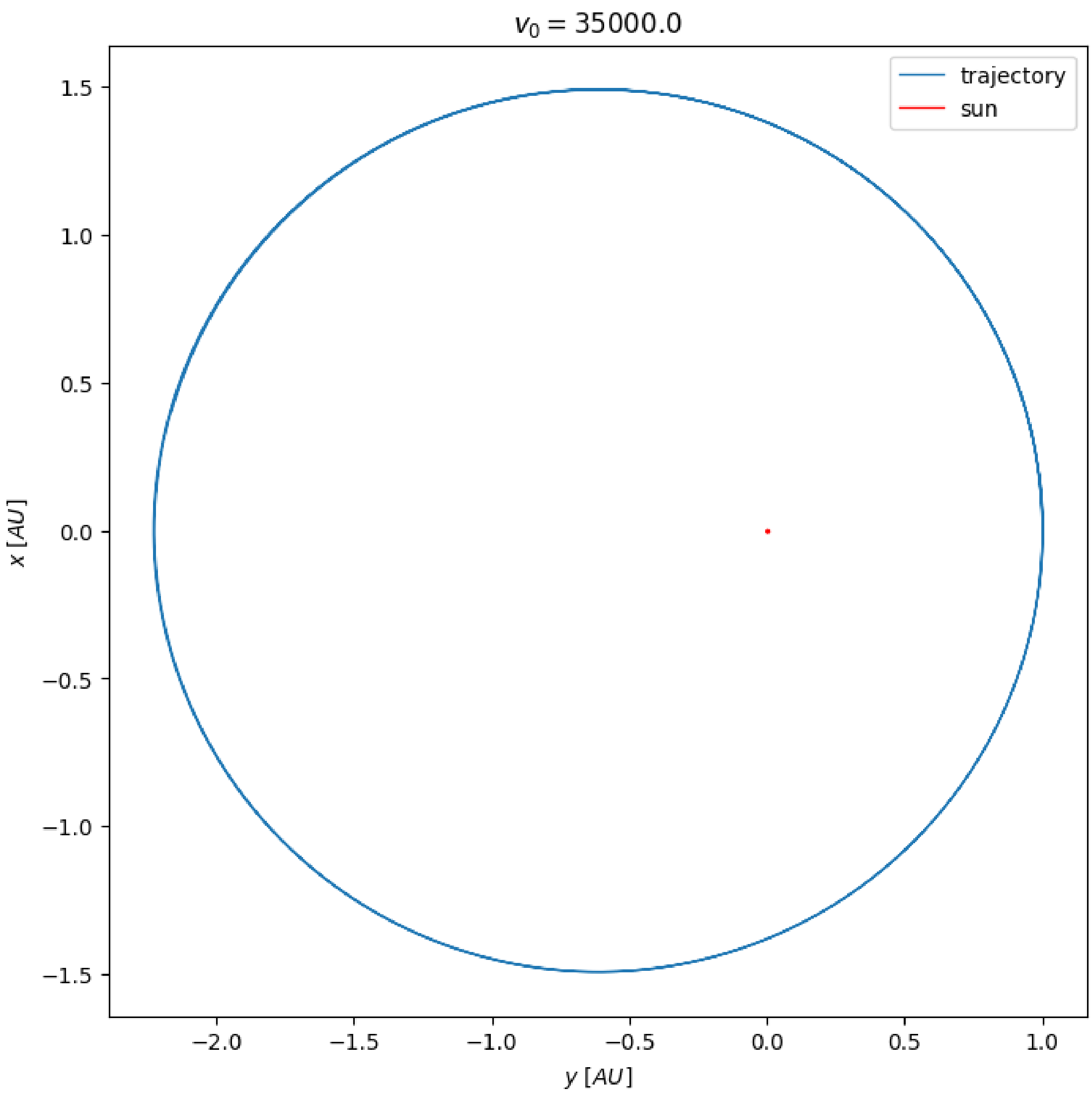
In [17]:
```
x=[d, 0., 0.]
E=0
v0=sqrt((E+g*M*m/d)*2./m)
v=[0., v0, 0]
rk4(d,m,M,g,v,x,t0,tend)
```



Out[17]: PyObject <matplotlib.legend.Legend object at 0x7fed6fce7090>

3.10 iii)

In [18]:
```
x=[d, 0., 0.]
v=[0., 35000, 0]
rk4(d,m,M,g,v,x,t0,tend)
```

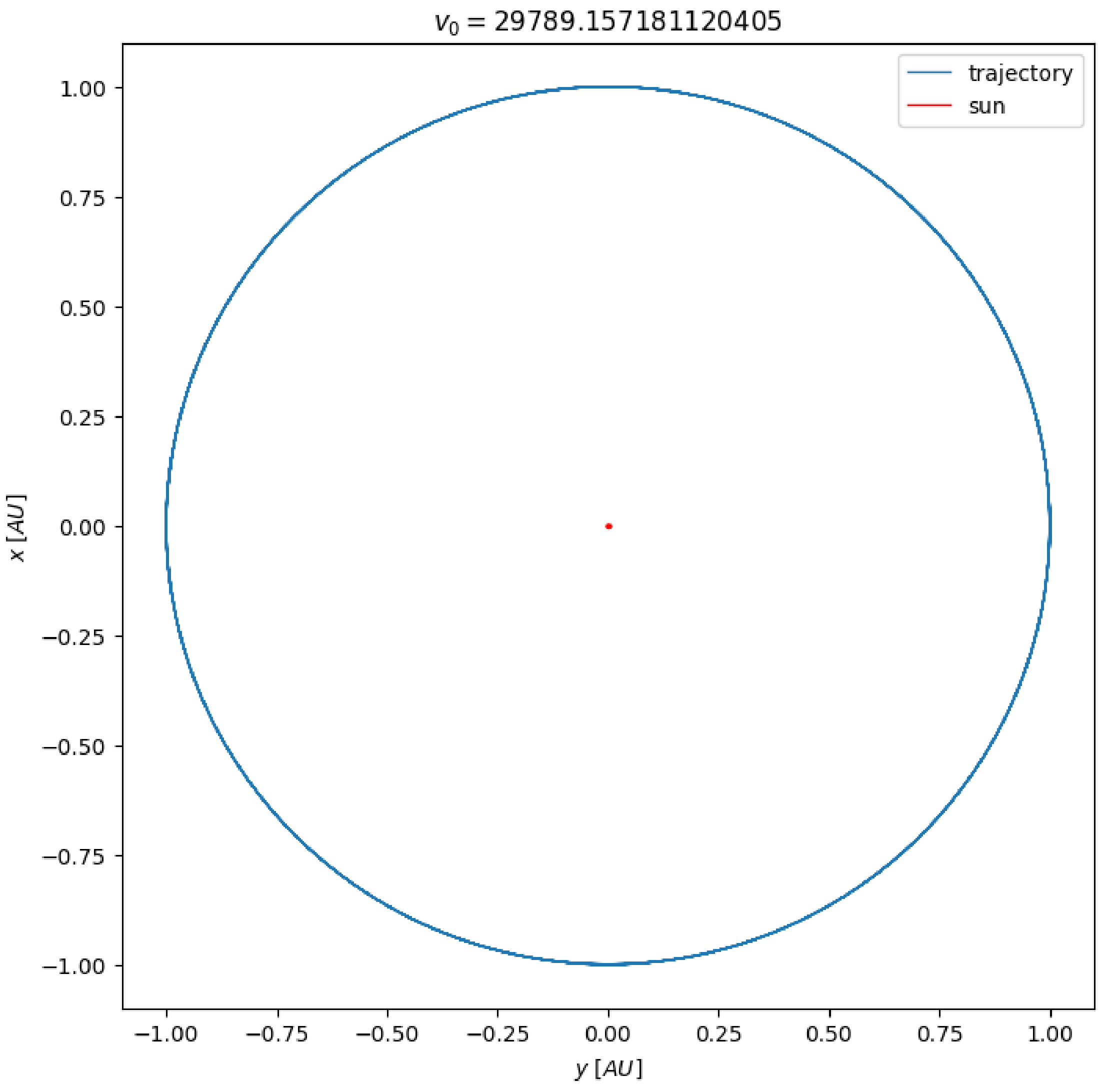


Out[18]: PyObject <matplotlib.legend.Legend object at 0x7fed6fbbb890>

3.11 iv)

In [19]:
```
x=[d, 0., 0.]
E=-g*M*m/(2*d)
v0=sqrt((E+g*M*m/d)*2./m)
```

```
v=[0., v0, 0]
rk4(d,m,M,g,v,x,t0,tend)
```




```
Out[19]: PyObject <matplotlib.legend.Legend object at 0x7fed6fb1fd50>
```