FYS3150 Project 3

Fredrik Wilhelm Holmen

The code can be found at: https://github.com/wholmen/FYS3150-Project-3

In this project I have simulated the solar system using the Newton's equation of motion: $F = \frac{M_1 M_2}{r^2}\hat{u}_r$. The initial conditions for the planets are found at http://nssdc.gsfc.nasa.gov/planetary/factsheet/.

All relevant plots are below in this PDF.

To solve Newton's equation of motion, I have used RungeKutta of fourth order. The equation of motion will be split into two coupled differential equations:

$$\frac{dx}{dt} = v_x and \frac{dv_x}{dt} = a_x$$

$$\frac{dy}{dt} = v_y and \frac{dv_y}{dt} = a_y$$

Here a can be rewritten as

$$a = \frac{F}{M_1} = \frac{M_2}{r^2}\hat{u}_r$$

I see that I get circular motions for all planets in the Solar System with my initial values. Changing those values will change the orbits.

I have numerical stability for $\Delta t \leq 10^{-2}$. RK4 has an error going as $O(h^4)$ and for the small planets, the error will grow drastically when the final time is large.

There are no friction forces acting on the planets. so the mechanical energy and the total angular momentum should be conserved. I have tested if this is the case, calculating the momentum and total energy for the earth. The total momentum is given as rv and the mechanical energy is $\frac{1}{2}mv^2 + G\frac{M_1 M_2}{r}$.

In the plots below I calculate the relative value, where I compare them to the value at $t = 0$. We see that the angular momentum differs more and more as time passes, but the energy is oscillating through time. The deviations from initial value are between 6 and 8%, which is a very large error.

By looking at the changing of total energy and angular momentum, I see that the numerical precision is a bit bad for the multi-body system, and by looking at the orbits for the innermost planets, I can see that the lines are much thicker because of numerical errors.

By trial and error I find that if I increase the initial velocity of the earth by 40%, it will escape the Sun. This happens when the kinetic energy is larger than the potential energy.

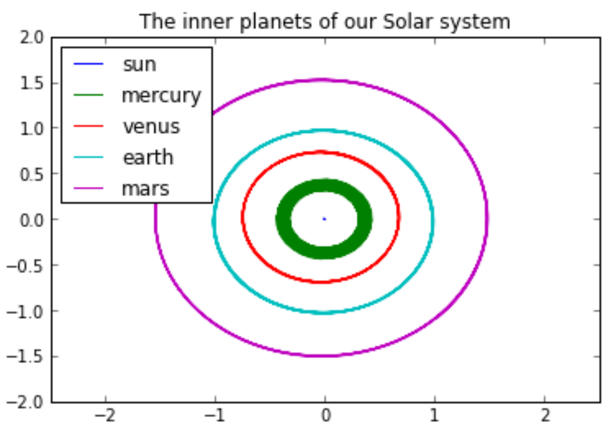
$$E_k \geq E_p$$
$$\frac{1}{2}M_e v^2 \geq G\frac{M_s M_e}{r}$$
$$v \geq \sqrt{\frac{2GM_s}{r}}$$

I get the result $42km/s$, which is 41.35% larger than the initial value. This is pretty close to the value achieved numerically.

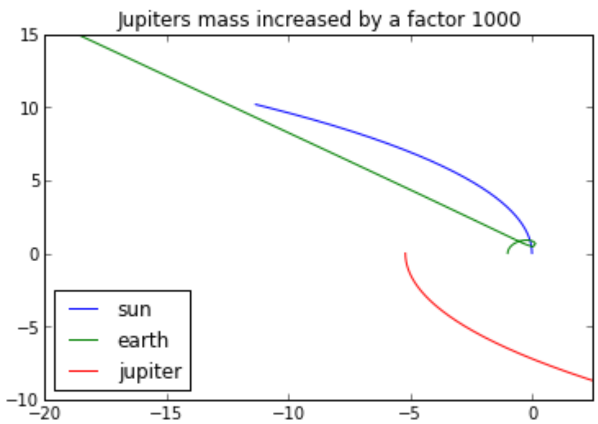
In [182]:
```
Msun = 1.99e30; G = 6.67e-11
print sqrt(2*G*Msun/AU)
print 42.125/29.8*100
```

```
42125.1777615
141.359060403
```

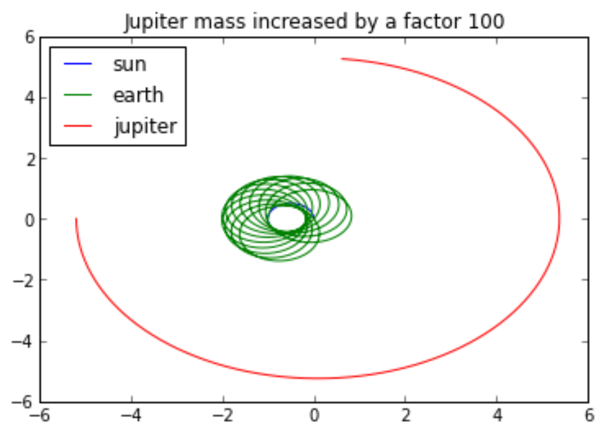
In [170]:
```
# Tfinal is 5 years for venus and mercury, and 10years for the rest.

from pylab import *

title0 = "sun10.txt"; title1 = "mercury5.txt"; title2 = "venus5.txt"; title3 = "eart
x00,y00,t00,E00 = readfile(title0); x11,y11,t11,E11 = readfile(title1); x22,y22,t22,
x44,y44,t44,E44 = readfile(title4); x55,y55,t55,E55 = readfile(title5)

plot(x00,y00,'-',label="sun")
plot(x11,y11,'-',label="mercury")
plot(x22,y22,'-',label="venus")
plot(x33,y33,'-',label="earth")
plot(x44,y44,'-',label="mars")
legend(loc="upper left")
xlim(-2.5,2.5)
title("The inner planets of our Solar system"
show()
```



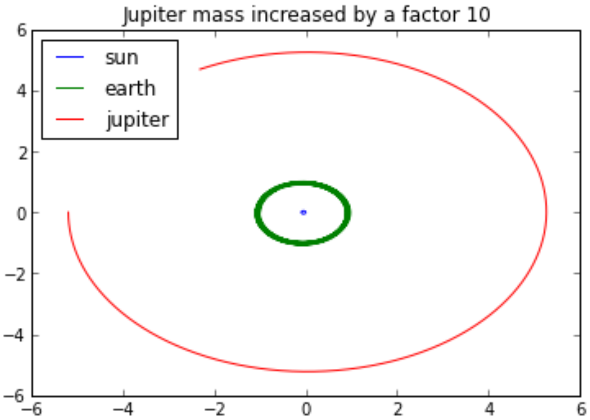
In [159]:
```
# Jupiter mass increased by a factor 1000
title0 = "sun_j1000.txt"; title1 = "earth_j1000.txt"; title2 = "jupiter_j1000.txt"
x0j, y0j, t0j, E0j = readfile(title0); x1j, y1j, t1j, E1j, L = readfile(title1); x2j

plot(x0j,y0j,label="sun")
plot(x1j,y1j,label="earth")
plot(x2j,y2j,label="jupiter")
legend(loc="lower left")
xlim(-20,2.5)
ylim(-10,15)
title("Jupiters mass increased by a factor 1000")
show()
```



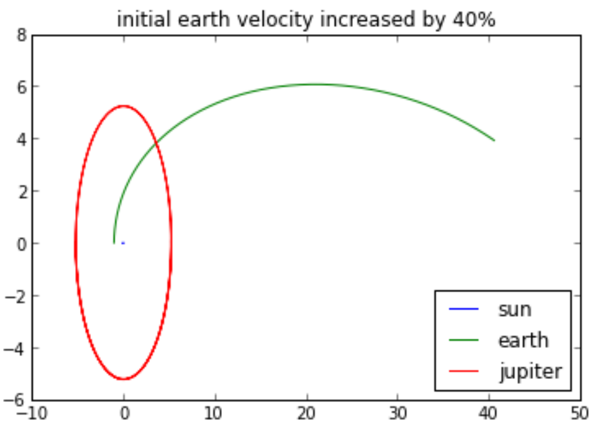
In [158]:
```
# Jupiter mass increased by a factor 100
title0 = "sun_j100.txt"; title1 = "earth_j100.txt"; title2 = "jupiter_j100.txt"
x0j, y0j, t0j, E0j = readfile(title0); x1j, y1j, t1j, E1j, L = readfile(title1); x2j

plot(x0j,y0j,label="sun")
plot(x1j,y1j,label="earth")
plot(x2j,y2j,label="jupiter")
legend(loc="upper left")
title("Jupiter mass increased by a factor 100")
show()
```



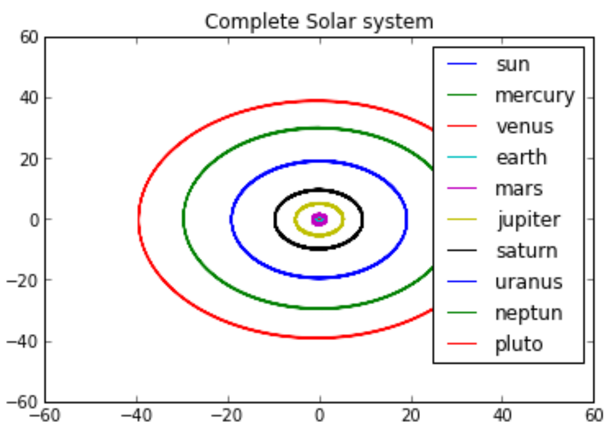
In [157]:
```
# Jupiter mass increased by a factor 10
title0 = "sun_j10.txt"; title1 = "earth_j10.txt"; title2 = "jupiter_j10.txt"
x0j, y0j, t0j, E0j = readfile(title0); x1j, y1j, t1j, E1j, L = readfile(title1); x2j

plot(x0j,y0j,label="sun")
plot(x1j,y1j,label="earth")
plot(x2j,y2j,label="jupiter")
legend(loc="upper left")
title("Jupiter mass increased by a factor 10")
show()
```



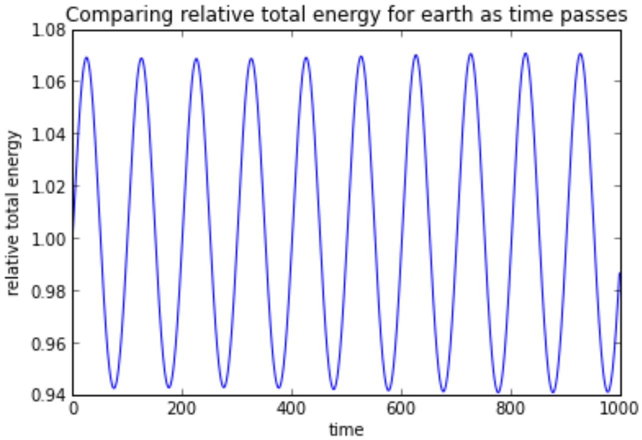
In [156]:
```
# Initial velocity for earth is increased by 40%

title0 = "sun_e.txt"; title1 = "earth_e.txt"; title2 = "jupiter_e.txt"
x0j, y0j, t0j, E0j = readfile(title0); x1j, y1j, t1j, E1j, L = readfile(title1); x2j

plot(x0j,y0j,label="sun")
plot(x1j,y1j,label="earth")
plot(x2j,y2j,label="jupiter")
legend(loc="lower right")
title("initial earth velocity increased by 40%")
show()
```



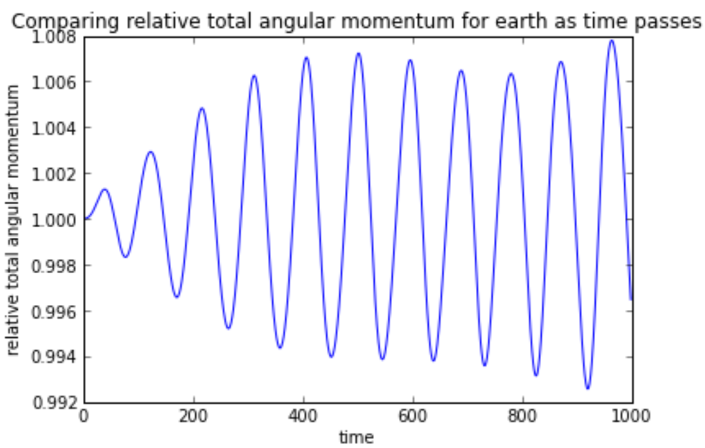
In [186]:
```
# Tfinal = 1000years. dt = 0.01

title0 = "sun.txt"; title1 = "mercury.txt"; title2 = "venus.txt"; title3 = "earth.tx
title5 = "jupiter.txt"; title6 = "saturn.txt"; title7 = "uranus.txt"; title8 = "nept
x0,y0,t0,E0 = readfile(title0); x1,y1,t1,E1 = readfile(title1); x2,y2,t2,E2 = readfi
x5,y5,t5,E5 = readfile(title5); x6,y6,t6,E6 = readfile(title6); x7,y7,t7,E7 = readfi

plot(x0,y0,'-',label="sun")
plot(x1,y1,'-',label="mercury")
plot(x2,y2,'-',label="venus")
plot(x3,y3,'-',label="earth")
plot(x4,y4,'-',label="mars")
plot(x5,y5,'-',label="jupiter")
plot(x6,y6,'-',label="saturn")
plot(x7,y7,'-',label="uranus")
plot(x8,y8,'-',label="neptun")
plot(x9,y9,'-',label="pluto")
legend(loc="upper right")
title("Complete Solar system")
xlim(-60,60)
ylim(-60,60)
show()
```



In [174]:
```
title3 = "earth10.txt"; x33,y33,t33,E33,L33 = readfile(title3)
del E33[-1]
E33 = array(E33)
plot(E33/E33[0])
xlabel("time")
ylabel("relative total energy")
title("Comparing relative total energy for earth as time passes")
show()
```



In [176]:
```
L33 = array(L33)
plot(L33/L33[0])
xlabel("time")
ylabel("relative total angular momentum")
title("Comparing relative total angular momentum for earth as time passes")
show()
```



In [106]:
```
# changing gravitational constant:
G = 6.67e-11 # m^3 kg^-1 s^-2
AU = 149597871000 #[m]
year = 3600*24*365.25 #[s]
G = G/AU**3*year**(2)
print "%e AU^3 kg^-1 year^-2" %G
```

```
1.984072e-29 AU^3 kg^-1 year^-2
```

In [107]:
```
def readfile(title):
    infile = open(title,'r')
    x = []; y = []; t = []; E = []; L = []
    for line in infile:
        lsplit = line.split()
        x.append(float(lsplit[0]))
        y.append(float(lsplit[1]))
        E.append(float(lsplit[2]))
        t.append(float(lsplit[3]))
        if len(lsplit) == 5:
            L.append(float(lsplit[4]))
    infile.close()
    if len(L) == 0:
        return array(x),array(y),array(t),E
    else:
        return array(x),array(y),array(t),E,L
```

In [ ]: