The following people have participated in creating these solutions: Nicolaas E. Groeneboom, Magnus Pedersen Lohne, Karl R. Leikanger *NOTE: There might be errors in the solution. If you find something which doens't look right, please let me know*

Partial solutions to problems: Lecture 23-24

Problem 1

1. Hubble's law gives $v = Hr$. We express the Hubble constant in km/s/AU: $H = 71km/s/Mpc = 71km/s/(2.1 \times 10^{11}AU) \approx 3.4 \times 10^{-10}km/s/AU$ We thus have $v = 3.4 \times 10^{-10}km/s/AU \times 1AU = 3.4 \times 10^{-7}m/s$.

2. $s = vt$ gives roughly 1/3 AU.

3. The FRW metric was derived assuming a homogeneous and isotropic universe. The universe is only homogeneous and isotropic on large scales, i.e., for distances of several Mpc. The FRW metric can therefore not be used to describe physics within smaller areas of the universe. All the results in the lectures on cosmology will therefor only be valid for objects very far apart.

Problem 2

1. For a sphere $x^2 + y^2 + z^2 = R^2$

2. $dr/d\theta = \cos\theta$

3. From the figure which you have drawn you see that $x_A = R\sin\theta\cos\phi_A = R\sin\theta$ and $x_B = R\sin(\theta + \Delta\theta)\cos\Delta\phi_{AB} \approx R\sin(\theta + \Delta\theta)$

4. Taylor expanding we have $sin(x + \epsilon) \approx \sin(x) + \cos(x)\epsilon$ for small ϵ.

5. Again from your drawing you find $y_A = R\sin\theta\sin\phi_A \approx 0$ and $y_B = R\sin\theta\sin\Delta\phi_{AB} \approx R\sin\theta\Delta\phi_{AB}$.

6. Combine previous expressions for Δx_{AB} and Δy_{AB}.

7. $z = R\cos\theta = R\sqrt{1 - \sin^2\theta} = R\sqrt{1 - r^2}$.

8. $$\frac{dz}{dr} = \frac{-rR}{\sqrt{1 - r^2}}$$

9. Combine previous results.

Problem 3

Only solutions to the numerical part are presented:

We use the expressions for the parametrization of $R(t), k = +1$ from the text:

$$R_+(x) = \frac{1}{2}\frac{\Omega_0}{\Omega_0 - 1}(1 - \cos x)$$
$$t_+(x) = \frac{1}{2}\frac{\Omega_0}{H_0(\Omega_0 - 1)^{3/2}}(x - \sin x)$$

And for $R(t), k = -1$:

$$R_-(x) = \frac{1}{2}\frac{\Omega_0}{1 - \Omega_0}(\cosh x - 1)$$
$$t_-(x) = \frac{1}{2}\frac{\Omega_0}{H_0(1 - \Omega_0)^{3/2}}(\sinh x - 1)$$

To calculate the expansion of a flat universe $k = 0$ we use the expression:

$$R(t) = \left(\frac{9}{4}\right)^{1/3} H_0^{2/3} t^{2/3}$$

This is one possible way of solving the problem.

```
#Remember to import scitools
from scitools.all import *

#Hubble's parameter
H=<...>

"""
Ex. 23.3.2
***
Make plots of different models for R(t) in the same figure.
"""

#Declare an array containing all values for x in [0,pi)
x=linspace(0,2*pi,100)

#Calculate R(x), t(x) for omega = 1.1 and k=+1.
#Make arrays containing all values of R,t.
omega=1.1
R_plus=<...>
t_plus=<...>
```

```
#Calculate R(x), t(x) for omega = 0.9 and k=-1.
#Make arrays containing all values of R,t.
omega=0.9
R_minus=<...>
t_minus=<...>

#Plot expantion of closed universe with Omega = 1.1.
plot(t_plus,R_plus)

#Use the hold() function to make multiple plots in the same figure.
hold('on')

#Plot expantion of the open universe.
plot(t_minus,R_minus)

#Plot expantion of the flat universe.
#Make a new array t, with values between 0 and max(t_plus)
#(the largest element in array t_plus)
t=linspace(0,max(t_plus),100)
R_flat=<...>
plot(t,R_flat)

#Set plot details
legend('closed universe','open universe','flat universe')
xlabel('<...>')
ylabel('<...>')

#Set the axis.
axis([0,130,0,40])

"""
Ex. 23.3.3
***
Make a plot of R(t) for a pressureless universe with omega=1.5 and k=+1,
and find the time in yr's from Big Bang to Big Crunch.
"""

#Calculate R(x), t(x) for omega = 1.1 and k=+1.
#Make arrays containing all values of R,t.
omega=1.5
R=<...>
t=<...>

#plot solution in a new window. The figure() function opens a new figure
```

```
figure()
plot(t,R)

#Set plot details
legend('closed universe')
xlabel('<...>')
ylabel('<...>')

#Calculate and print lifetime of the closed universe. Convert to yr's.
lifetime = <...>
print 'Lifetime: %f yr' %lifetime
# (approx 149*10^9yr)
```

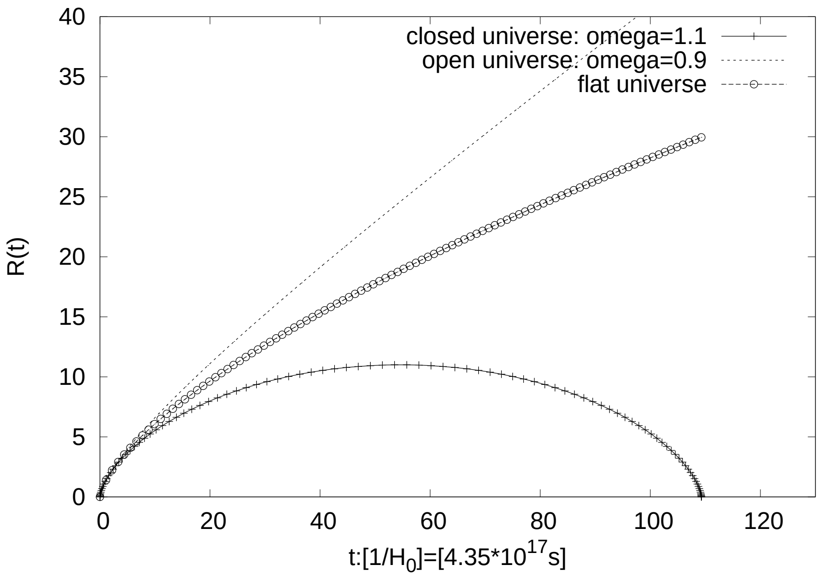
Problem 4-5

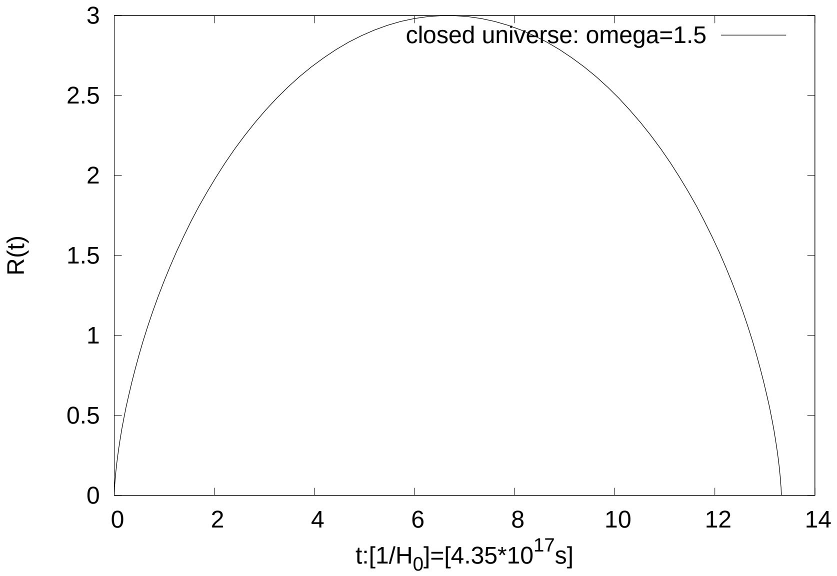
Following the same procedure as for similar derivations in the text, these two problems should be straightforward and no solutions are presented.

Problem 6

1. Insert $q_0 = 1/2$ (flat universe) in the expression
2. Result given in the question.
3. Result given in the question.
4. We get
$$r = \int_t^{t_0} \frac{dt'}{R(t')}$$
Insert $R(t) = R(t_0)(t/t_0)^{2/3}$ for a flat universe and integrate.
5. For simplicity we set $R(t_0) = 1$. We have deduced in the lecture that $R(t_0)/R(t) = 1 + z$ such that for a flat universe $t/t_0 = (1 + z)^{-3/2}$. We also deduced in the lecture an expression for $R(t)$ and $\dot{R}(t)$ in a flat universe. Combining these expressions we find $H_0 = 2/(3t_0)$ which we use to eliminate t_0 in the expression for r. Combining this information you easily arrive at the correct result.
6. Result given in the question.
7. See below.
8. Nothing to be done.

The plots: Problem 3, lecture 23-24.





9. Comparing with the solar luminosity one finds

$$M - M_{\odot} = -2.5\log_{10}(\frac{L}{L_{\odot}}) = -2.5(\log_{10}(4\pi\sigma) + 4\log_{10}(T) + 2\log_{10}(v) + 2\log_{10}(\Delta t) - \log_{10}(L_{\odot})).$$

In the following you will find examples on how to code the solutions to the numerical problems:

The files `supernovadata_i.txt` for i=1,2,3 must be saved in the same folder as your program for this examples to run.

As always; the `scitools` library and gnuplot are necessary to run most examples.

Remember: These hints are just one way of many to solve the problems.
Problem 23.6.7: We define a function which takes an array with values for z, and the deceleration parameter q_0 as imput, and returns an array with values for $(m-M)_{model}$.

```
...
#calculate theoretical mM values.
#Imput: array x(redshift), float q_0(decleration parameter).
def mMmodel(x,q_0):
   return  <...>
...
```

We make three plots for $\Omega_0 = (0.3, 1.0, 1.5)$ in the same figure, with $(m-M)_{model}$ values between 37 and 46 and z values between 0 and 2.

```
...
#Open a new figure, and use the hold function to make multiple plots
#in the same figure.
figure()
hold('on')

#Make an array that contains values for the redshift
x=linspace(0,2,100)

#Loop over the different values of Omega_0
#and plot mMtheoretic against x
for omega_0 in ( 0.3, 1.0, 1.5):

   #calculate q_0 as a function omega_0
   q_0= <...>

   #Find mMmodel for q_0,x and plot it
   mMtheoretic=mMmodel(x,q_0)
   plot(x,mMtheoretic)

   #Plot legend: print q_0 with two decimals

#Set axis and other plot details
xlabel('z')
ylabel('m-M')
axis([0,2,37,46])
```

```
legend('q_0=%.2f' %q_0)
...
```

Problem 23.6.10: We read datafiles with 5 data columns, and then sort the columns in arrays. The different columns contains data from measurements of supernovae as specified in the problem text. We then calculate $(m - M)_{data}$ and plot the results in the same figure as the theoretical values $(m - M)_{model}$ for $\Omega = 0.3, 1, 1.5$.

The following function take the infilename as an argument, and reads and sorts the data. The function can be copypasted directly into your program.

```
...
#Function reads infile, and sorts the data in arrays.
#Input: The name of the datafile
#Returns the columns of the datafile arranged in arrays.
def readfile(infilename):

   #Open the datafile for reading
   infile=open(infilename,'r')
   #The readlines() function arranges all the lines of the datafile in a list
   lines=infile.readlines()
   infile.close()

   #Declare arrays with the same length as lines
   n=len(lines)

   z=zeros(n,float) #redshift
   m=zeros(n,float) #apparent magnitude
   T=zeros(n,float) #surf temp K of shell
   v=zeros(n,float) #shell velocity km/s
   deltat=zeros(n,float) #time after explotion (days)

   #Write the columns of the datafile to arrays.
   #The split() function splits the lines into ''words''.
   #Remember to convert from datatype string to float

   i=0
   for line in lines:
      temp=line.split()

      z[i]=float(temp[0])
      m[i]=float(temp[1])
      T[i]=float(temp[2])
      v[i]=float(temp[3])
      deltat[i]=float(temp[4])
      i+=1

   #convert from days to seconds
   deltat*=24*60*60
   #convert from km/s to m/s
   v*=1000

   return (z,m,T,v,deltat)
...
```

Now, when we have read the datafiles, $(m - M)_{data}$ is easily calculated:

```
...
#Calculate experimental values for mMdata
mMdata =  <...>
...
```

We plot the data in the same figure as the plots of $(m - M)_{model}$. We define a function which will do the job:

```
...
#Plot mMdata as a function of redshift z, in the same figure as the theoretical
#plots of mM for different values of q_0.
#Takes mMdata and z (from datafile) and the infilename
#as  argumnts.
def plotdata(z,mMdata,infilename):

   #Open a new figure,
   figure()

   #Plot the values of mMdata from the datafile.
   plot(z,mMdata)
   legend('%s' %infilename)

   hold('on')

   ...
   <MAKE THE PLOTS FOR mMmodel IN THE SAME WAY AS WE DID IN THE
   PREVIOUS EXERCISE (23.6.7). INSERT ALGORIHTM HERE.>
   ...
...
```

To solve the exersice the functions must be implemented in a program. Here is one example on how to do it:

```
...
```

```
#Infilenames: supernovadata_i.dat
infilename = <...>

#Read datafile and sort data in arrays by calling function readfile()
(z,m,T,v,deltat)=readfile(infilename)

#Calculate experimental values for mMdata
mMdata =  <...>

#plot mMdata against z, in the same plot as the theorethical functions
#for mM, by calling the function plotdata()
plotdata(z,mMdata,infilename)
...
```

This lines must be placed after (under) the other functions in the program in order to work. We see that the first data set gives observations which are closest to the line in the middle which is the line for the flat universe. Our universe is flat so this is probably the set of observations made in our universe.

Problem 23.6.12: We define a function `delta()` that calculates

$$\Delta(q_0) = \sum \left((m-M)_{data} - (m-M)_{model}\right)^2$$

One way of doing this is to write.

```
...
#find Delta(q_0). Takes arrays z,mMdata (from datafile), and
#some value of q_0 as imput. Return float Delta(q_0).
def delta(z,q_0,mMdata):
   #Calculate mMmodel for the same values of the
   #redshift as the experimental data points.
   mMtheoretical=mMmodel(z,q_0)

   #Elementvise operation.
   delta_q_0=(mMdata-mMtheoretical)**2

   #The sum() function sums all array elements.
   return sum(delta_q_0)
...
```

Now we make an array of $\Delta(q_0)$ for values of $q_0 \in [-0.25, 2)$. We assume that arrays for mM and z are avaliable. The following lines give us an array $\Delta(q_0)$ with $n = 1000$:

```
...
#Make a delta(q) array
```

```
n=1000
q_0=linspace(-.25,2,n)
Deltaq=zeros(n,float)
for i in range(n):
   #Call delta() function for different values of q_0 but with
   #the same arrays z,mMdata as imput.
   Deltaq[i]=delta(z,q_0[i],mMdata)
...
```

To find the index of the minima of `Deltaq` is easy. We simply call the `scitools` function `argmin()`, whitch takes an array as input, and returns the index of (one of) the smallest value(s) in the array. We find the corresponding value of the deceleration parameter q_0 that minimizes `deltaq` (gives the best least square fit), by calling `z[argmin(deltaq)]`.

```
...
#Find the index  of the smallest element 'minima' of Deltaq
minima=argmin(deltaq)
#Print the value of q_0 in the minima (q_0[minima]) with 4 decimals
print 'least square fitting: q_0 = %.4f' %q_0[minima]
...
```

The results are $\Omega = 2\times 0.5 = 1$, $\Omega = 2\times(-0.1) = -0.2$ and $\Omega = 2\times 0.9 = 1.8$ for the three models. As expected from our by-eye inspection, the first universe is a flat universe. The second universe gave a surpise: a negative Ω is not possible within the universe models which we have studied. This is the same surprise as the researchers got in 1998. I model of the univeerse with dark energy is needed. In such a model the relation $\Omega = 2q_0$ is not valid and this is why we get an unphysical result for Ω. We know that our universe is dark energy dominated, so this is the set of observations made in our universe, not the first one as we first thought.

Problem 23.6.14 Now we are going to do least square fitting only with the data from the supernovae with redshift less than 0.2. First we need to sort the data, and make new arrays `mMsorted` and `zsorted` with data only from the relevant supernova observations. This function takes `mMdata ,z` as input, and returns the sorted arrays `mMsorted` and `zsorted`.

```
...
#Makes two new arrays with all values of z smaller than
#maxz, and the corresponding values of mMdata
def sort(z,mMdata):
   #declare arrays with 0 elements
   zsorted=array([],float)
   mMsorted=array([],float)
   #The maximum value of z
   maxz=0.2

   #Find all elements of z smaller than maxz
   #and add the elements to zsorted,
   #add the corresponding elements of mMdata to mMsorted.
   for i in range(len(z)):
      if z[i]<maxz:
         #The append() function adds a new element to an array
         zsorted=append(zsorted,z[i])
         mMsorted=append(mMsorted,mMdata[i])

   return (zsorted,mMsorted)
...
```

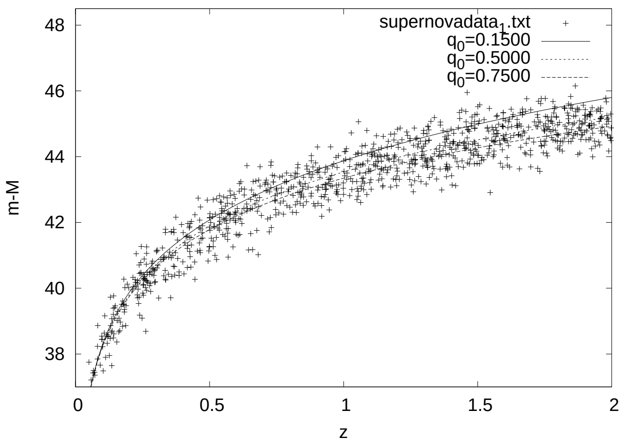
Now, call the function:

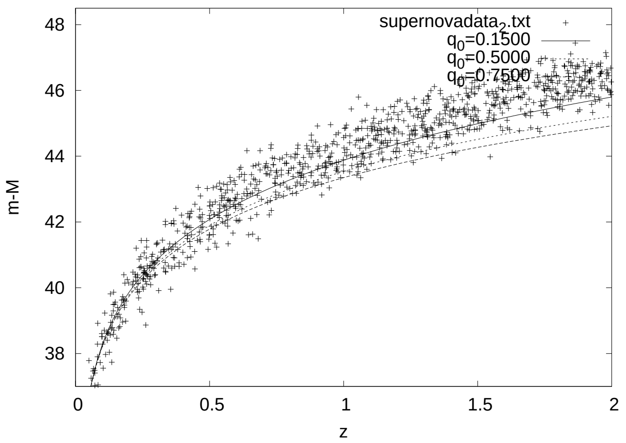
```
...
(zsorted,mMsorted)=sort(z,mMdata)
...
```

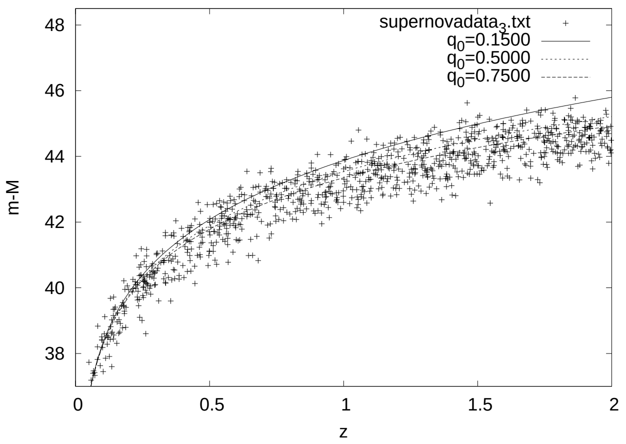
And repeat the least square fitting from the previous exercise. You find $q_0 \approx 0.1$ which is a positive value for the second universe (the one which had a negative q_0 when you included observations at higher redshifts). We see that the conclusion that the universe contains dark energy is based only on the supernovae at high redshift.

Problem 23.6.15 When looking at supernovae at very high redshift, we look at supernova explosions which took place a very long time ago. These supernovae exploded at a time when the population II and III stars were the most common stars (whereas today the population I stars are the most common). We learned in the lectures on the 'end state of stars' that population II and III stars have much less metals and the physics of the supernova explosions could therefore be quite different from the supernova explosions that we observer in the nearby universe. The methods to find the absolute magnitude of distant supernovae is based on observations of nearby supernovae, but since these explosions may take place in stars which are very different we cannot be sure that our method is correct.

The plots: Problem 6, lecture 23.


supernovadata_1.txt
q_0=0.1500
q_0=0.5000
q_0=0.7500
m-M
z


supernovadata_2.txt
q_0=0.1500
q_0=0.5000
q_0=0.7500
m-M
z


supernovadata_3.txt
q_0=0.1500
q_0=0.5000
q_0=0.7500
m-M
z