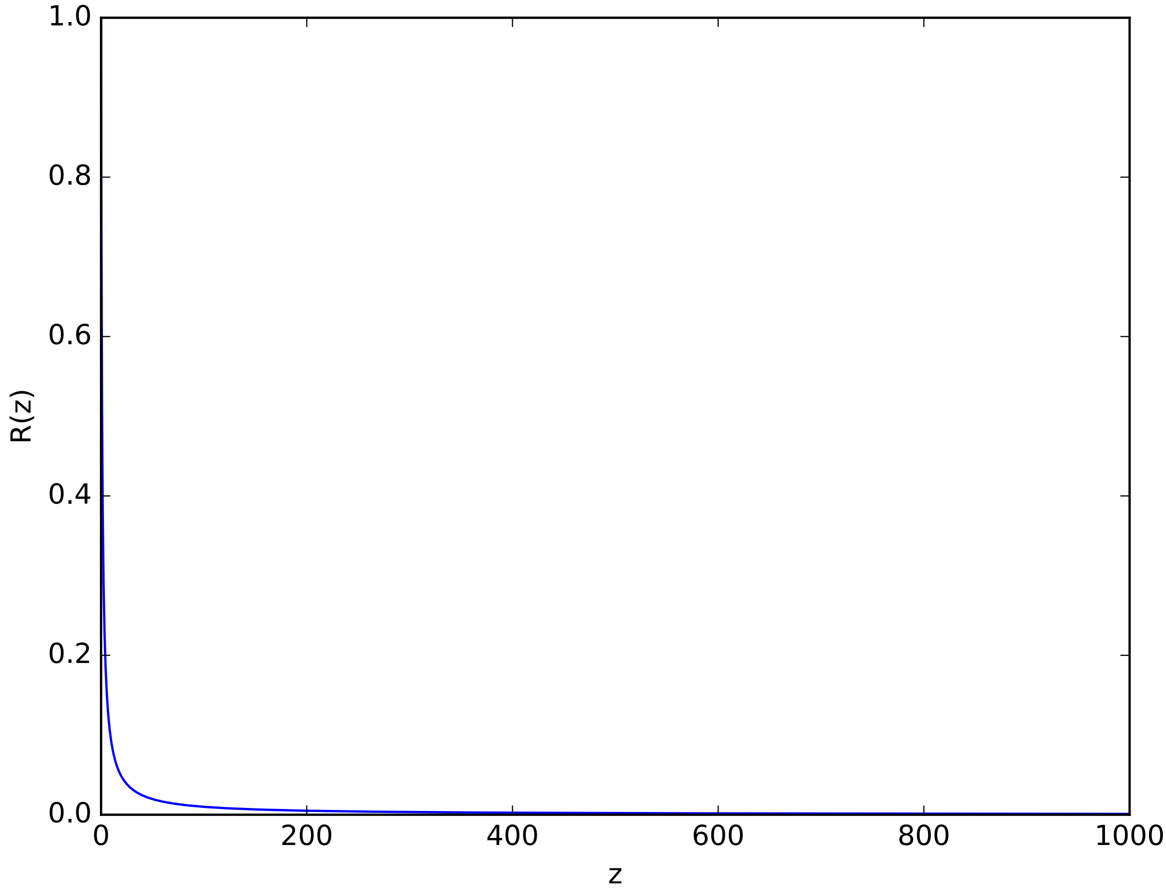
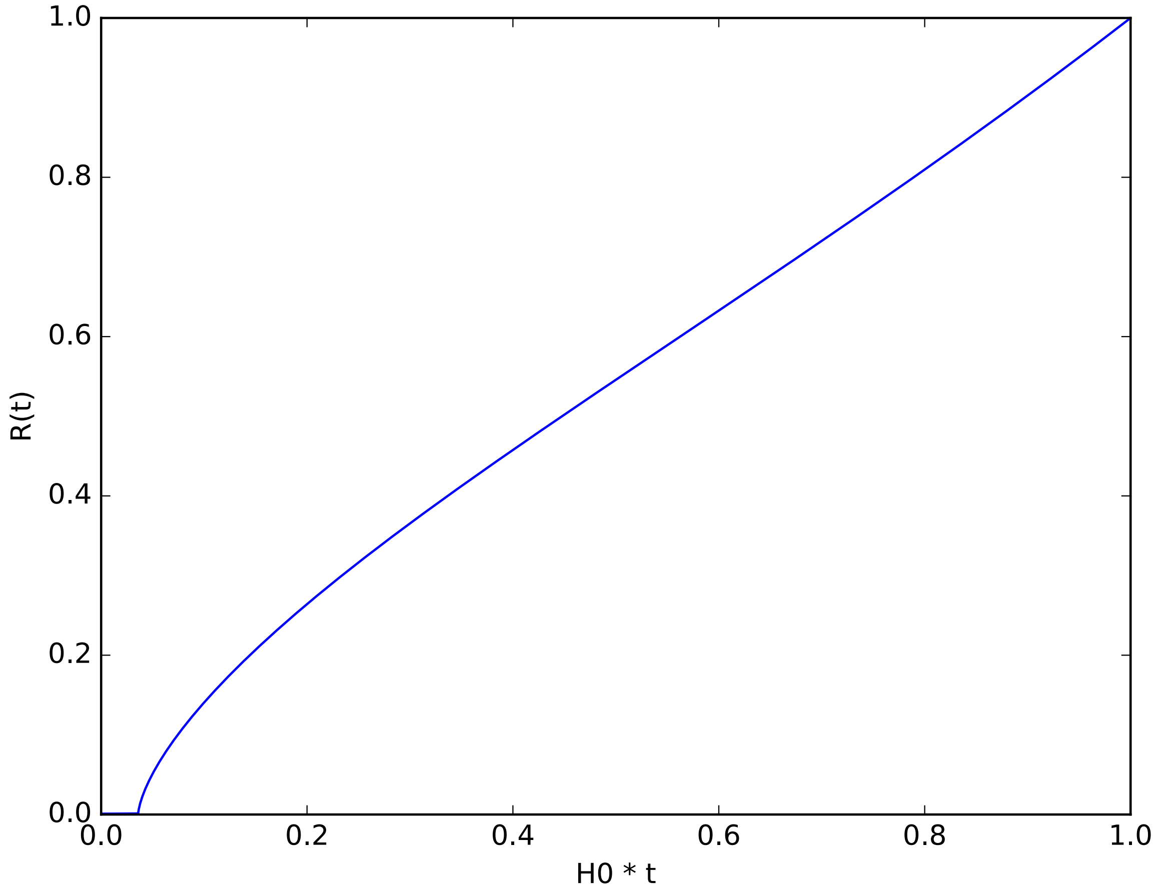
```
import matplotlib.pyplot as plt
import numpy as np
from tools import multipage

omega_m = 0.3
omega_lambda = 0.7
omega_k = 0

import matplotlib.pyplot as plt
import numpy as np

omega_m = 0.3
omega_lambda = 0.7
omega_k = 0
N=10e5
R = 1 #B.C
H0 = 70

dt = 1.0/N/H0 #i want the time scales normalized to hubbles

def func(R, H0):
    global omega_m, omega_lambda
    R_dot = H0 * R *  (omega_m * (R ** -3) + omega_lambda)**0.5
    R -= dt*R_dot #going back in time
    return max(R,0.001) , R_dot


R_arr = np.ndarray(N)
R_dot_arr = np.ndarray(N)
T_arr = np.linspace(0,1,N)
t = N*dt

for i in xrange(int(N)-1,-1,-1):
    R  , R_dot = func(R,H0)
    R_arr[i]=R
    R_dot_arr[i] = R_dot
    t -= dt
R_dot_dot_arr = np.gradient(R_dot_arr,T_arr)
z = 1/R_arr -1
# Plot it
plt.figure()
plt.plot(T_arr, R_arr)
plt.xlabel("H0 * t")
plt.ylabel("R(t)")
plt.figure()
plt.plot(z, R_arr)
plt.xlabel("z")
plt.ylabel("R(z)")
plt.figure()
plt.plot(T_arr, R_dot_arr)
plt.xlabel("H0 * t")
plt.ylabel("dR/dt (t)")
plt.figure()
plt.plot(T_arr[50000:], R_dot_dot_arr[50000:]) #dont want to see that
     nasty delta in the big bang
plt.xlabel("z")
plt.ylabel("Scale Acceleration d^2R/dt^2 (t)")
plt.grid()
#plt.show()
```



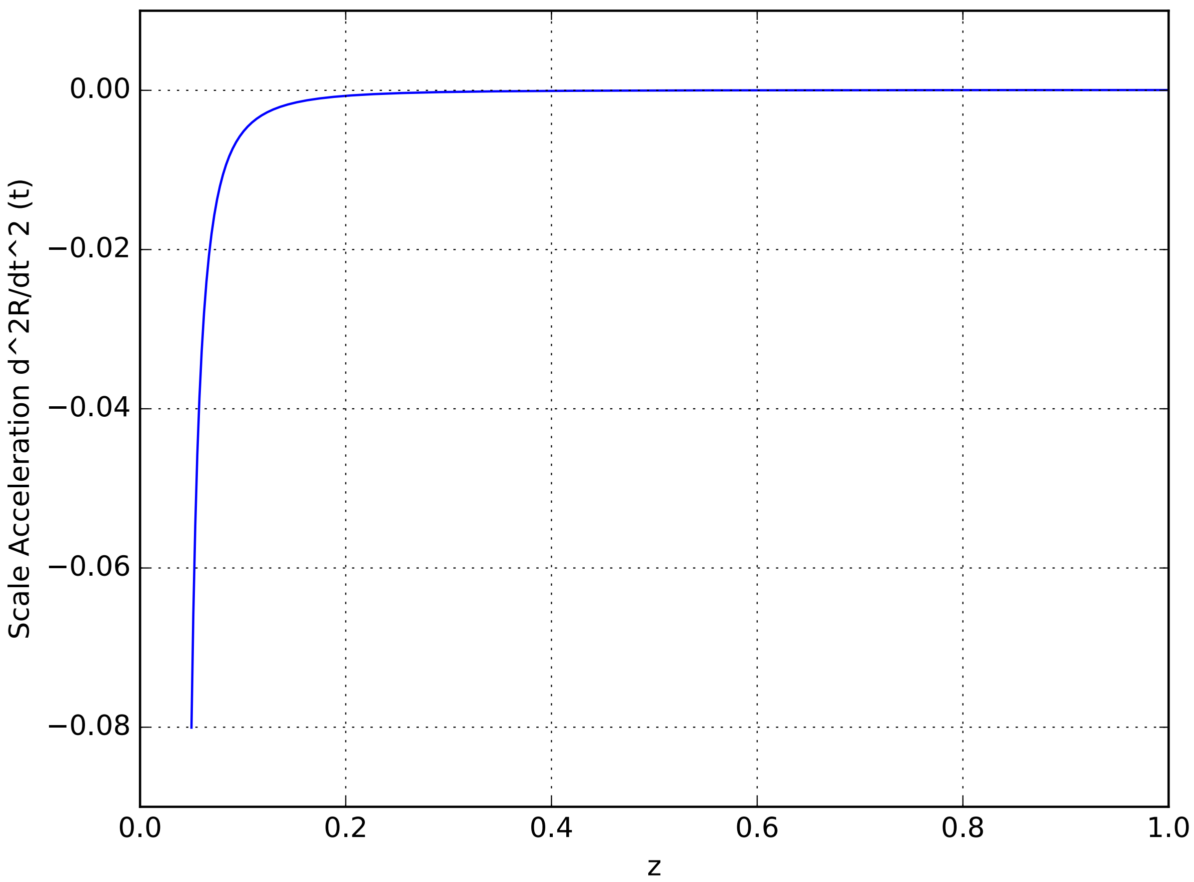
```
multipage('cosmo1_figs.pdf')
```

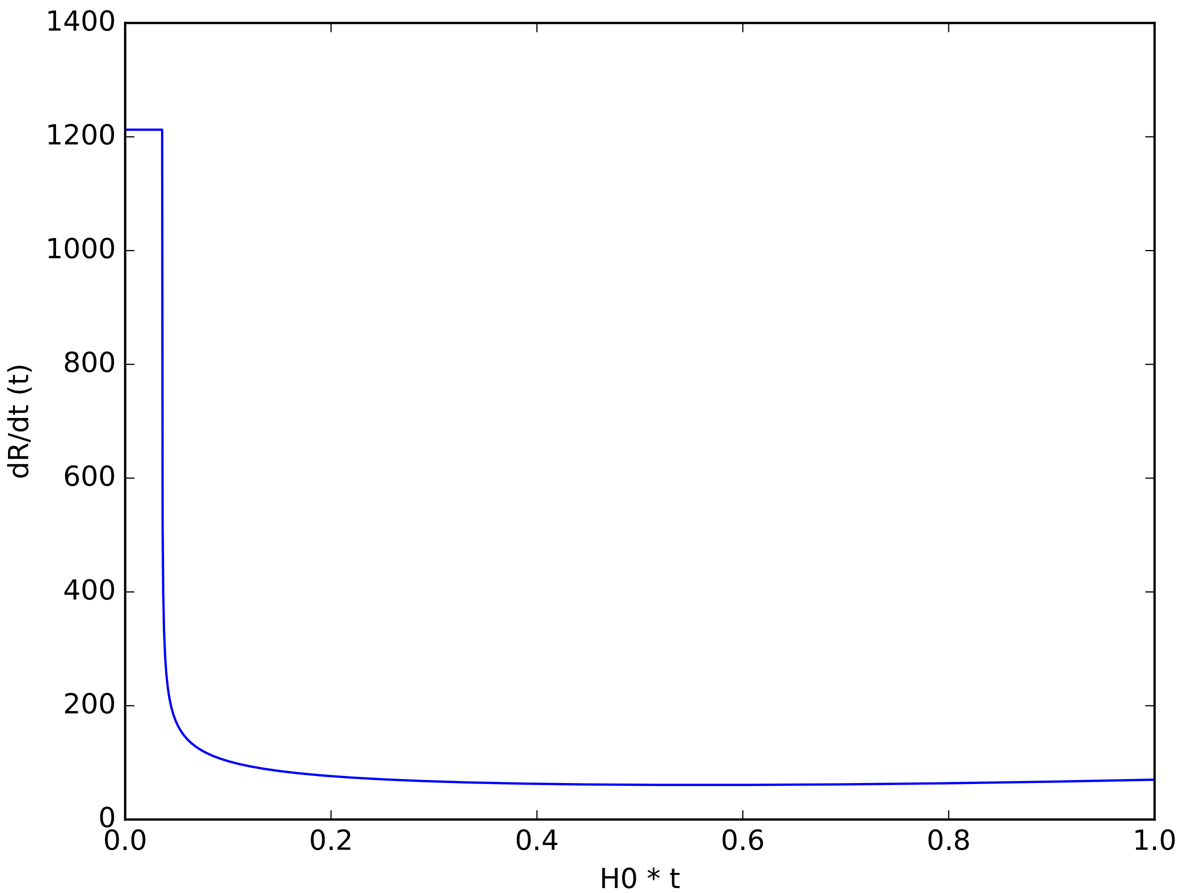


1.0
0.8
0.6
0.4
0.2
0.0
R(z)
0
200
400
600
800
1000
z



1.0
0.8
0.6
0.4
0.2
0.0
R(t)
0.0
0.2
0.4
0.6
0.8
1.0
H0 * t


Scale Acceleration d^2R/dt^2 (t)
0.00
−0.02
−0.04
−0.06
−0.08
0.0
0.2
0.4
0.6
0.8
1.0
z


dR/dt (t)
1400
1200
1000
800
600
400
200
0
0.0
0.2
0.4
0.6
0.8
1.0
H0 * t