*Zach Warren*

*ASTR 3800*

*Homework 4*

*Code:*

**import** numpy **as** np  
**import** math **as** math  
**import** matplotlib.pyplot **as** plt  
**import** scipy.integrate **as** integrate  
  
f, plts = plt.subplots(3,2,figsize=(23,15))  
  
**def** Dc(m,lam,w,z):  
 w = 1+ w  
 E = **lambda** y: 1/(np.sqrt((m\*math.pow(y+1,3))+ (lam\*math.pow(y+1,w))))  
 Dc = [3\*(integrate.quad(E, 0, x))[0] **for** x **in** z]  
 **return** Dc  
  
**def** DL(z,Dc):  
 DL = [(1+z)\*Dc **for** z,Dc **in** zip(z,Dc)]  
 **return** DL  
**def** DA(z,Dc):  
 DA = [Dc/(1+z) **for** z,Dc **in** zip(z,Dc)]  
 **return** DA  
**def** VC(Dc):  
 Vc = [(4.0/3)\*math.pi\*math.pow(x,3) **for** x **in** Dc]  
 **return** Vc  
**def** tl(m,lam,w,z):  
 w = 1+ w  
 E = **lambda** y: 1/((1+y)\*(np.sqrt((m\*math.pow(y+1,3))+ (lam\*math.pow(y+1,w)))))  
 tl = [9.78\*(integrate.quad(E, 0, x))[0] **for** x **in** z]  
 **return** tl  
  
z = np.linspace(0,3,100)  
  
*#part1*Dc1 = Dc(1,0,0,z)  
DL1 = DL(z,Dc1)  
DA1 = DA(z,Dc1)  
Vc1 = VC(Dc1)  
tl1 = tl(1,0,0,z)  
  
plts[0,0].plot(z,Dc1)  
plts[0,1].plot(z,DL1)  
plts[1,0].plot(z,DA1)  
plts[1,1].plot(z,Vc1)  
plts[2,0].plot(z,tl1)  
  
*#part2*Dc2 = Dc(.25,.75,-1,z)  
DL2 = DL(z,Dc2)  
DA2 = DA(z,Dc2)  
Vc2 = VC(Dc2)  
tl2 = tl(.25,.75,-1,z)  
  
plts[0,0].plot(z,Dc2, **'r'**)  
plts[0,1].plot(z,DL2, **'r'**)  
plts[1,0].plot(z,DA2, **'r'**)  
plts[1,1].plot(z,Vc2, **'r'**)  
plts[2,0].plot(z,tl2, **'r'**)  
  
*#part3*Dc3 = Dc(.25,.75,-.8,z)  
DL3 = DL(z,Dc3)  
DA3 = DA(z,Dc3)  
Vc3 = VC(Dc3)  
tl3 = tl(.25,.75,-.8,z)  
  
plts[0,0].plot(z,Dc3, **'c'**)  
plts[0,1].plot(z,DL3, **'c'**)  
plts[1,0].plot(z,DA3, **'c'**)  
plts[1,1].plot(z,Vc3, **'c'**)  
plts[2,0].plot(z,tl3, **'c'**)  
  
*#part4*Dc4 = Dc(.25,.75,-1.2,z)  
DL4 = DL(z,Dc4)  
DA4 = DA(z,Dc4)  
Vc4 = VC(Dc4)  
tl4 = tl(.25,.75,-.12,z)  
  
plts[0,0].plot(z,Dc4, **'g'**)  
plts[0,1].plot(z,DL4, **'g'**)  
plts[1,0].plot(z,DA4, **'g'**)  
plts[1,1].plot(z,Vc4, **'g'**)  
plts[2,0].plot(z,tl4, **'g'**)  
  
*#set up plots*plts[0,0].legend([**'1'**,**'2'**,**'3'**,**'4'**])  
plts[0,0].set\_title(**'Comoving Distance vs Redshift'**)  
plts[0,0].set\_xlabel(**'z'**)  
plts[0,0].set\_ylabel(**'Dc (h-1 Gpc)'**)  
  
plts[0,1].legend([**'1'**,**'2'**,**'3'**,**'4'**])  
plts[0,1].set\_title(**'Luminosity Distance vs Redshift'**)  
plts[0,1].set\_xlabel(**'z'**)  
plts[0,1].set\_ylabel(**'DL (h-1 Gpc)'**)  
  
plts[1,0].legend([**'1'**,**'2'**,**'3'**,**'4'**])  
plts[1,0].set\_title(**'Angular Diameter Distance vs Redshift'**)  
plts[1,0].set\_xlabel(**'z'**)  
plts[1,0].set\_ylabel(**'DA (h-1 Gpc)'**)  
  
plts[1,1].legend([**'1'**,**'2'**,**'3'**,**'4'**])  
plts[1,1].set\_title(**'Comoving volume vs Redshift'**)  
plts[1,1].set\_xlabel(**'z'**)  
plts[1,1].set\_ylabel(**'Vc (h-3 Gpc3)'**)  
  
plts[2,0].legend([**'1'**,**'2'**,**'3'**,**'4'**])  
plts[2,0].set\_title(**'Lookback time vs Redshift'**)  
plts[2,0].set\_xlabel(**'z'**)  
plts[2,0].set\_ylabel(**'tl (Gyr)'**)  
  
  
plt.show()  
  
  
