

```

1 # E(z) en andere kosmische parameters berekenen
2 #
3 # een paar bibliotheken klaarzetten
4 #
5 import numpy as np
6 import matplotlib.pyplot as plt
7 import math
8 #
9 #
10 const=977.8 #(1 hubble is bijna 1000 miljard jaar)
11 hubble=67.3 # LCMD
12 #
13 age_const=const/hubble #(1/hubble) = Hubble time
14 #
15 dist_const=3e5/hubble # in Mpc
16 dist_const2=dist_const*1000 # in kpc
17
18 #
19 dz = 0.001 # stap redshift
20 z_start = 0 # begin van de redshift
21 z_end = 20 # eind van de redshift
22 n_steps = int ( round (( z_end - z_start )/dz)) # number of timesteps
23 print(n_steps)
24 #
25 z_arr = np. zeros ( n_steps ) # array of zeros
26 #
27 Ez_arr = np. zeros ( n_steps) # array of zeros
28 Comoving_arr = np. zeros ( n_steps ) # array of zeros
29 App_diam_arr = np. zeros ( n_steps ) # array of zeros
30 diam_arr = np. zeros ( n_steps ) # array of zeros
31 Lum_dist_arr = np. zeros ( n_steps ) # array of zeros
32 Dist_mod_arr = np. zeros ( n_steps ) # array of zeros
33 Lookback_arr = np. zeros ( n_steps ) # array of zeros
34 #
35 Ez_arr2 = np. zeros ( n_steps) # array of zeros
36 Comoving_arr2 = np. zeros ( n_steps ) # array of zeros
37 App_diam_arr2 = np. zeros ( n_steps ) # array of zeros
38 diam_arr2 = np. zeros ( n_steps ) # array of zeros
39 Lum_dist_arr2 = np. zeros ( n_steps ) # array of zeros
40 Dist_mod_arr2 = np. zeros ( n_steps ) # array of zeros
41 Lookback_arr2 = np. zeros ( n_steps ) # array of zeros
42
43
44 # eerste berekening
45 Omega_m=0.3
46 Omega_l=0.7
47 Omega_r=0.0 # is eigenlijk 0.0001
48
49 str_m1=str(Omega_m)
50 str_l1=str(Omega_l)
51 labell='lambda='+str_l1+',matter='+str_m1
52
53 Omega_k=1-Omega_m-Omega_l-Omega_r
54 print(Omega_k,Omega_m,dist_const)
55
56 #
57 # calculalting Ez
58 for i in range (0, n_steps):
59     z = z_start + i * dz
60     z_arr [i] = z
61     Ez = Omega_r * ((z+1)**4) + Omega_m * ((z+1)**3) + (Omega_k * (z+1)**2) + Omega_l
62     Ez_arr [i] = np.sqrt(Ez)
63
64 #
65 integral=0
66 integral2=0
67 for i in range (0, n_steps):
68     z = z_arr [i]
69     integral = integral + dz/(Ez_arr [i])
70     Comoving = integral * dist_const2 #in kpc
71     App_diam_arr [i] = Comoving/(z+1)
72     Comoving_arr [i] = integral * dist_const #in Mpc

```

```

73     Lum_dist_arr [i] = Comoving_arr [i]*(z+1)
74     Dist_mod_arr [i] = 5 * math.log(Lum_dist_arr [i], 10) + 25
75     diam_arr [i] = 1/App_diam_arr [i] # in radialen
76     diam_arr [i] = diam_arr [i]*57.3*3600 # in arcsec/kpc
77     integral2 = integral2 + dz/((z+1)* Ez_arr [i])
78     Lookback_arr [i] = integral2 * age_const
79
80
81 # tweede berekening
82 Omega_m=1.0
83 Omega_l=0.0
84 Omega_r=0.0
85
86 str_m2=str(Omega_m)
87 str_l2=str(Omega_l)
88 label2='lambda='+str_l2+',matter='+str_m2
89
90 Omega_k=1-Omega_m-Omega_l-Omega_r
91 print(Omega_k,Omega_m,dist_const)
92
93 #
94 # calculating Ez2
95 for i in range (0, n_steps):
96     z = z_arr [i]
97     Ez = Omega_r * ((z+1)**4) + Omega_m * ((z+1)**3) + (Omega_k * (z+1)**2) + Omega_l
98     Ez_arr2 [i] = np.sqrt(Ez)
99
100 #
101 integral=0
102 integral2=0
103 for i in range (0, n_steps):
104     z = z_arr [i]
105     integral = integral + dz/(Ez_arr2 [i])
106     Comoving = integral * dist_const2 #in kpc
107     App_diam_arr2 [i] = Comoving/(z+1)
108     Comoving_arr2 [i] = integral * dist_const #in Mpc
109     Lum_dist_arr2 [i] = Comoving_arr2 [i]*(z+1)
110     Dist_mod_arr2 [i] = 5 * math.log(Lum_dist_arr2 [i], 10) + 25
111     diam_arr2 [i] = 1/App_diam_arr2 [i] # in radialen
112     diam_arr2 [i] = diam_arr2 [i]*57.3*3600 # in arcsec/kpc
113     integral2 = integral2 + dz/((z+1)* Ez_arr2 [i])
114     Lookback_arr2 [i] = integral2 * age_const
115
116 #
117 fig = plt. figure (1)
118 plt . plot (z_arr , App_diam_arr , linewidth = 2, color ='blue', label = label1)
119 plt . plot (z_arr , App_diam_arr2 , linewidth = 2, color ='red', label = label2)
120 plt.title ("App_diam distance", fontsize = 12)
121 plt.grid ( True )
122 plt.xlabel ('redshift')
123 plt.ylabel ('value (kpc)')
124 plt.legend ()
125 #plt.axis ([0, 20 , 0, 14])
126
127 #
128 fig = plt. figure (2)
129 plt . plot (z_arr , Comoving_arr , linewidth = 2, color ='blue', label = label1)
130 plt . plot (z_arr , Comoving_arr2 , linewidth = 2, color ='red', label = label2)
131 plt.title ("Comoving distance", fontsize = 12)
132 plt.grid ( True )
133 plt.xlabel ('redshift')
134 plt.ylabel ('value (Mpc)')
135 plt.legend ()
136 #plt.axis ([0, 20 , 0, 0.1])
137
138 #
139 fig = plt. figure (3)
140 plt . plot (z_arr , diam_arr , linewidth = 2, color ='blue', label = label1)
141 plt . plot (z_arr , diam_arr2 , linewidth = 2, color ='red', label = label2)
142 plt.title ("apparent diameter", fontsize = 12)
143 plt.grid ( True )
144 plt.xlabel ('redshift')

```

```

145 plt.ylabel ('arcsec/kpc')
146 plt.legend ()
147 plt.axis ([0, 20 , 0, 2])
148
149 #
150 fig = plt. figure (4)
151 plt . plot (z_arr , Lum_dist_arr , linewidth = 2, color ='blue', label = label1)
152 plt . plot (z_arr , Lum_dist_arr2 , linewidth = 2, color ='red', label = label2)
153 plt.title ("luminosity distance", fontsize = 12)
154 plt.grid ( True )
155 plt.xlabel ('redshift')
156 plt.ylabel ('value (Mpc)')
157 plt.legend ()
158 # plt.axis ([0, 20 , 0, 2])
159
160 #
161 fig = plt. figure (5)
162 plt . plot (z_arr , Dist_mod_arr , linewidth = 2, color ='blue', label = label1)
163 plt . plot (z_arr , Dist_mod_arr2 , linewidth = 2, color ='red', label = label2)
164 plt.title ("distance modulus", fontsize = 12)
165 plt.grid ( True )
166 plt.xlabel ('redshift')
167 plt.ylabel ('value (mag)')
168 plt.legend ()
169 plt.axis ([0, 20 , 40, 55])
170
171 #
172 fig = plt. figure (6)
173 plt . plot (z_arr , Lookback_arr , linewidth = 2, color ='blue', label = label1)
174 plt . plot (z_arr , Lookback_arr2 , linewidth = 2, color ='red', label = label2)
175 plt.title ("Lookback time", fontsize = 12)
176 plt.grid ( True )
177 plt.xlabel ('redshift')
178 plt.ylabel ('value (Gjaar)')
179 plt.legend ()
180
181 #
182 plt . show ()
183
184 # end
185
186

```