

Computerøvelse: Friedmann-modeller S

Published

Assign to

Edit

Computerøvelsen for uge 6. Mere information følger til undervisningen onsdag d. 11. oktober.

Bemærk: Øvelsen må gerne afleveres i grupper. Sørg for at tilmelde jer en gruppe på Absalon eller skriv alle gruppemedlemmers navne på hver side af opgavebesvarelsen.

I denne øvelse vil vi arbejde med, hvordan man numerisk kan løse for skalafaktorens udvikling og sammenligning med data for mere komplicerede kosmologiske modeller, der inkluderer mere end et bidrag til energitætheden.

Jeg har lavet et lille eksempel (simpleexample.py), der viser, hvordan man kan lave et numerisk integral. Det er ganske givet ikke særligt elegant, så nogen af jer kan sikkert lave det meget smartere. Vi skal nu prøve at lave noget lignende med Friedmann-ligningerne.

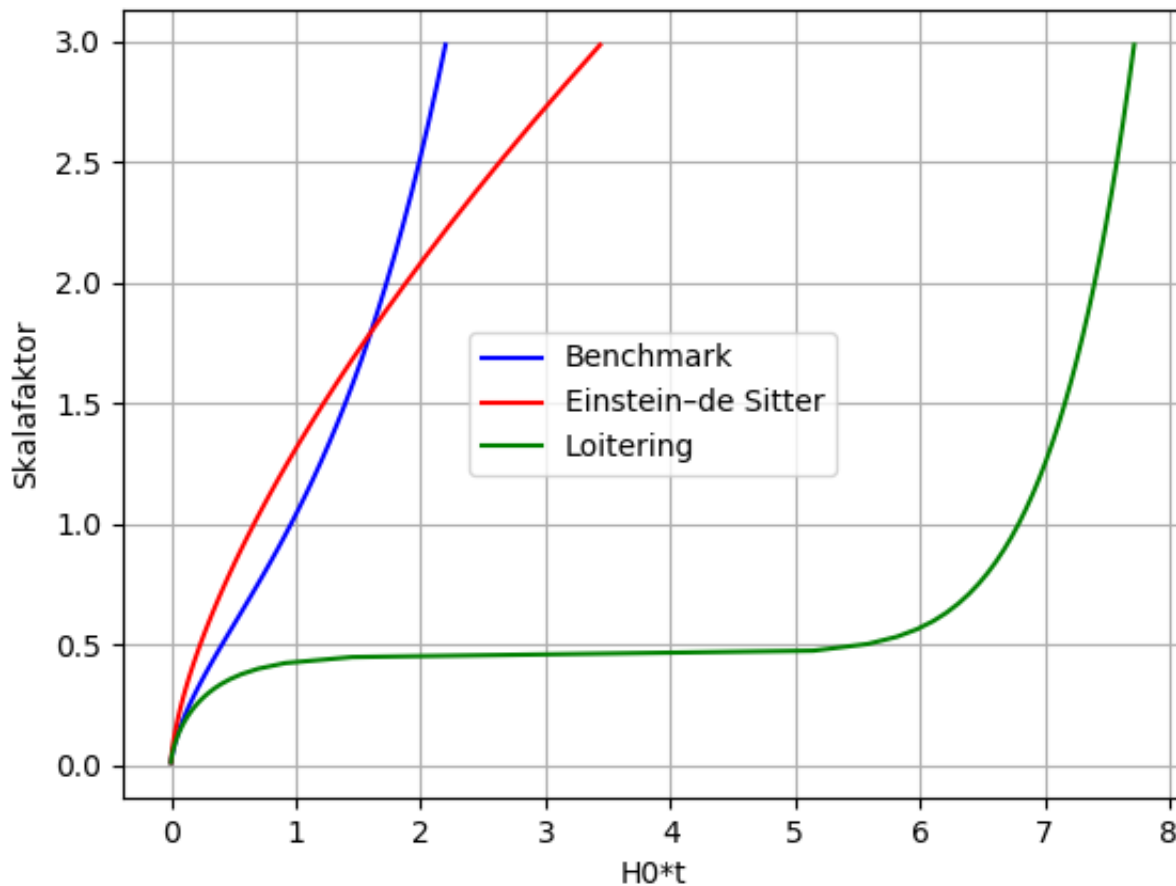
1) Start med at vise (eller læs noterne, afsnit 7.4), at

$$\left(\frac{da}{dt}\right) = \sqrt{H_0^2 \left(\frac{\Omega_{m,0}}{a} + \Omega_{\Lambda,0} a^2 + (1 - \Omega_0) \right)}$$

Lav dernæst et python program, der kan finde en numerisk løsning af denne ligning (t som funktion af a) for flg. værdier af parametrene: 1) Omega_m=0.3, Omega_Lambda=0.7, 2) Omega_m=1.0, Omega_Lambda=0.0, 3) Omega_m = 0.337, Omega_Lambda=1.7701. Omega_0 er summen af alle bidrag til Omega. Vælg evt. værdierne for a med logaritmisk spredning:

```
a = 10**(np.arange(0,100,1)/40.-2.)
```

I skulle gerne få noget ala det her:



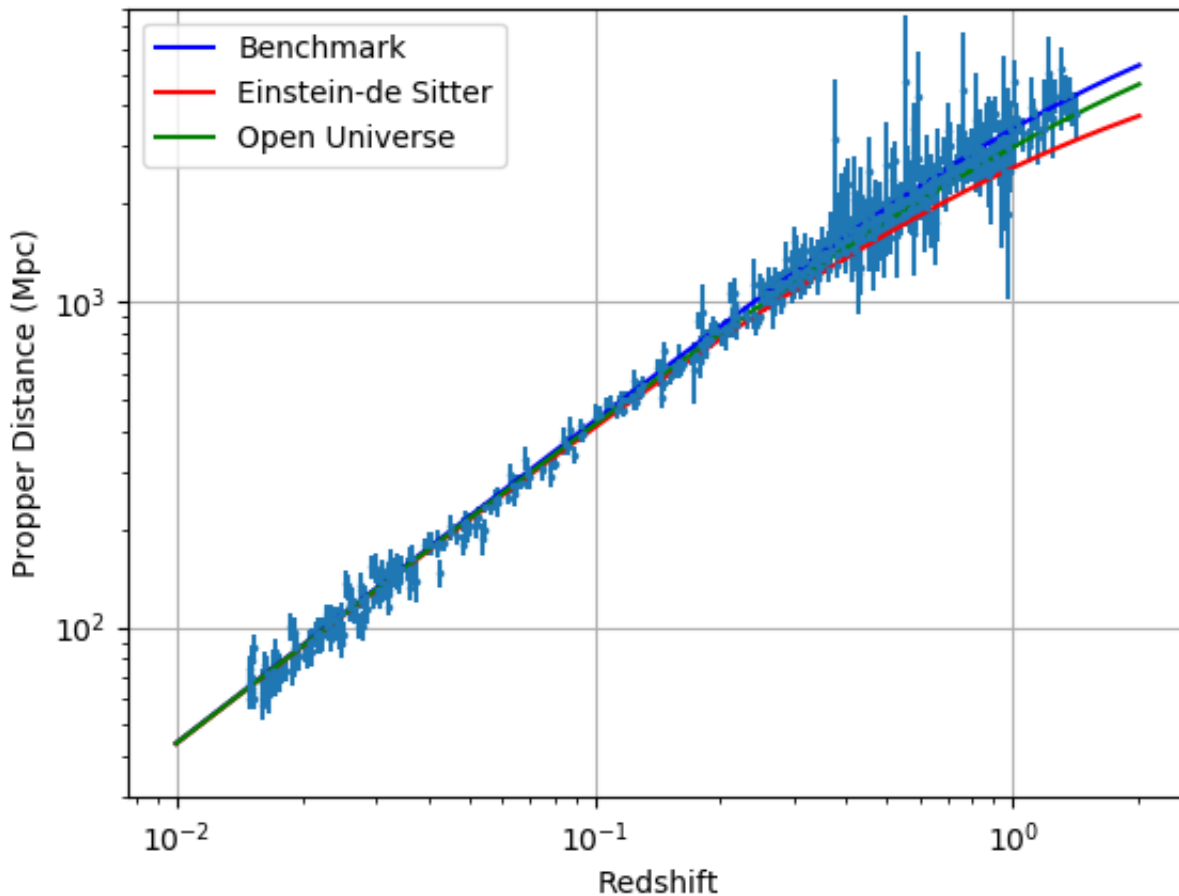
Brug evt. lidt tid på at undersøge, hvilke løsninger man får med forskellige valg af de kosmologiske parametre. Bemærk, at hvis I vil undersøge universer, der kolliderer, så skal man tænke over fortegnet på differentialligningen.

2) Udløst dernæst en formel for dr/dz (hvor r er metrik-afstanden). Brug formelen fra 1) samt formler for dr/dt og da/dz (jeg har lagt hints op). Beregn dernæst r som funktion af z for 1) $\Omega_m=0.3$, $\Omega_\Lambda=0.7$, 2) $\Omega_m=1.0$, $\Omega_\Lambda=0.0$, 3) $\Omega_m=0.3$, $\Omega_\Lambda=0.0$. Sammenlign med observationer af type Ia supernovaer i filen `SN_SCP.data`.

Data-filen læses sådan her:

```
data = np.loadtxt('SN_SCP.data')
zdata = np.array(data[:,0])
dlumdata = 10**((0.2*(np.array(data[:,1])-25)))
dlumerror = 10**((0.2*(np.array(data[:,1])+np.array(data[:,2])-25))-dlumdata)
ddata = dlumdata/(1+zdata)
derror = dlumerror/(1+zdata)
```

I skulle gerne få noget ala det her:



3) Jeg har lagt et program op (cosmology.py) som I kan bruge til at beregne afstande og tider for generelle kosmologiske modeller. Det kan I for brug for siden hen i studiet. Jeg lægger mine løsninger til 1) og 2) op efter øvelsen, så man ikke bliver desperat, hvis ikke kan få det til at fungere.

Faktisk findes der allerede en masse nyttigt i pakken cosmology som er en del af astropy. Man kan læse mere herom her: [Link → \(https://docs.astropy.org/en/stable/cosmology/\)](https://docs.astropy.org/en/stable/cosmology/)

Med denne pakke kan man f.eks. gøre sådan her:

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.integrate as spi
from astropy.cosmology import Planck15 as cosmo

plt.close('all')
```

```

#Load supernova data
data = np.loadtxt('SN_SCP.data')
zdata = np.array(data[:,0])
dlumdata = 10**(0.2*(np.array(data[:,1])-25))
dlumerror = 10**(0.2*(np.array(data[:,1])+np.array(data[:,2])-25))-dlumdata
ddata = dlumdata/(1+zdata)
derror = dlumerror/(1+zdata)

z = (np.arange(1,201,1))/100.

#Make the plot: distance vs. redshift
fig, ax = plt.subplots()
ax.grid()
ax.errorbar(zdata,ddata,yerr=derror,fmt='.',ms=3)
ax.set(xlabel='Redshift', ylabel='Propper Distance (Mpc)',yscale='log',xscale="log",ylim=(30,8000))
ax.plot(z, cosmo.luminosity_distance(z)/(1+z), 'b', label='Benchmark')
ax.legend()
plt.show()

```

Points 1

Submitting a file upload

File types pdf, ipynb, and py

Due	For	Available from	Until
12 Oct 2023	Everyone	5 Sep 2023 at 0:00	12 Oct 2023 at 23:59

+ Create rubric

🔍 Find rubric

