28/28

Cosmology HW2 Connor Hainje

code for this HW at github.com/cmhainje/cosmo-hw/blob/main/hw2/homework2.ipynb

PROBLEM 1:

part 1: 1/1

$$\left(\frac{\dot a}{a}\right)^2 = H_0^2 \sum_n \Omega_n a^n \rightarrow \dot a^2 = H_0^2 \sum_n \Omega_n a^{n+2}$$

$$t(a) = \int \frac{dt}{da}\, da = \int \frac{1}{\dot a}\, da = H_0^{-1} \int_0^a \frac{da}{(\sum_n \Omega_n a^{n+2})^{1/2}}$$ ✓

part 2:

(a) 2/2 neglecting curvature $(\Omega_k = 0)$ and taking $a \ll a_{m\Lambda}$,

$$t(a) = H_0^{-1} \int \frac{da}{\sqrt{\Omega_m a^{-1} + \Omega_r a^{-2}}}$$

$a_{eq} = \frac{\Omega_r}{\Omega_m}$ so $\Omega_m a^{-1} + \Omega_r a^{-2} = a_{eq}^{-2}\left(\Omega_m a_{eq}\left(\frac{a_{eq}}{a}\right) + \Omega_r\left(\frac{a_{eq}}{a}\right)^2\right)$

$$= a_{eq}^{-2}\left[\Omega_r \left(\frac{a}{a_{eq}}\right)^{-1} + \Omega_r \left(\frac{a}{a_{eq}}\right)^{-2}\right]$$

then $t(a) = \frac{a_{eq}}{H_0\sqrt{\Omega_r}} \int \frac{da}{\sqrt{(a/a_{eq})^{-1} + (a/a_{eq})^{-2}}}$

let $x = a/a_{eq} \rightarrow da = a_{eq}\, dx$ so $t(a) = \frac{a_{eq}^2}{H_0\sqrt{\Omega_r}} \int \frac{dx}{\sqrt{x^{-1} + x^{-2}}}$

$$= \frac{a_{eq}^2}{H_0\sqrt{\Omega_r}} \int \frac{x\, dx}{\sqrt{1+x}}$$

$$= \frac{2a_{eq}^2}{3H_0\sqrt{\Omega_r}} (x-2)\sqrt{1+x} + \text{const}$$

let $t(0) = 0 \rightarrow 0 = \frac{-4a_{eq}^2}{3H_0\sqrt{\Omega_r}} + C$ so $t(a) = \frac{2a_{eq}^2}{3H_0\sqrt{\Omega_r}}\left[2 + (x-2)\sqrt{1+x}\right]$ ✓

difference in exponent of a_{eq} is due to my use of $\sqrt{\Omega_r}$ compared to solution's use of $\sqrt{\Omega_m}$

(b) 4/4 $t(a) = \frac{2a_{eq}^2}{3H_0\sqrt{\Omega_r}} (x-2)\sqrt{1+x}$

$a \ll a_{eq} \rightarrow x \ll 1$: $(x-2)\sqrt{1+x} = (-2+x)(1 + x/2 - \frac{1}{8}x^2 + O(x^3))$

$$= -2 + x - x + \frac{1}{2}x^2 + \frac{1}{4}x^2 + O(x^3)$$

$$= -2 + \frac{3}{4}x^2 + O(x^3)$$

$a \gg a_{eq} \rightarrow x \gg 1$: let $y = \frac{1}{x}$. limit is then $y \ll 1$.

$$(x-2)\sqrt{1+x} = \left(\frac{1}{y} - 2\right)\sqrt{1 + 1/y} = y^{-3/2}(1-2y)\sqrt{1+y}$$

$$= y^{-3/2}(1-2y)(1 + y/2 - \frac{1}{8}y^2 + O(y^3))$$

$$= y^{-3/2}(1 - 2y + y/2 - y^2 - \frac{1}{8}y^2 + O(y^3))$$

$$= y^{-3/2} - \frac{3}{2}y^{-1/2} + O(y^0)$$

$$= x^{3/2} - \frac{3}{2}x^{1/2} + \Omega(x^0)$$

I guess obviously it's just $x^{3/2}$ to leading order hence the limits are

$$t(a) \approx \begin{cases} \frac{a_{eq}^2}{2H_0\sqrt{\Omega_r}} \left(a/a_{eq}\right)^2 ✓ & \text{if } a \ll a_{eq} \\ \frac{2a_{eq}^2}{3H_0\sqrt{\Omega_r}} \left(a/a_{eq}\right)^{3/2} ✓ & a \gg a_{eq} \end{cases}$$

(c) $t_{eq} = t(a_{eq}) = \frac{2a_{eq}^2}{3H_0\sqrt{\Omega_r}}\left[2 + (1-2)\sqrt{1+1}\right] = \frac{2(2-\sqrt{2})}{3}\frac{a_{eq}^2}{H_0\sqrt{\Omega_r}}$ ✓

1/1

$H_0 = 67.66\ \mathrm{km/s/Mpc}$ from Planck 18 $\quad \Omega_r = 9.0\times10^{-5}$ from lecture

$\Omega_m = 0.30966$ from Planck 18 $\longrightarrow a_{eq} = 2.9\times10^{-4}$

then $t_{eq} = 5.1\times10^{4}$ year ✓ agrees with naive estimate from lecture!

part 3.

(a) neglecting curvature and taking $a \gg a_{eq}$, $\quad t(a) = H_0^{-1}\int \frac{da}{\sqrt{\Omega_m a^{-1} + \Omega_\Lambda a^{+2}}}$

2/2

defining $a_{m\Lambda} = \left(\frac{\Omega_m}{\Omega_\Lambda}\right)^{1/3}$ and $x = a/a_{m\Lambda}$, $\quad dx\, a_{m\Lambda} = da$

$$\Omega_m a^{-1} + \Omega_\Lambda a^{+2} = \Omega_\Lambda a_{m\Lambda}^3 a^{-1} + \Omega_\Lambda a^{+2} = \Omega_\Lambda a_{m\Lambda}^2 \left(x^{-1} + x^{+2}\right)$$

then $t(a) = \frac{1}{H_0\sqrt{\Omega_\Lambda}\, a_{m\Lambda}}\int\frac{da}{\sqrt{x^{-1}+x^2}} = \frac{1}{H_0\sqrt{\Omega_\Lambda}}\int\frac{dx}{\sqrt{x^{-1}+x^2}} = \frac{1}{H_0\sqrt{\Omega_\Lambda}}\int\sqrt{\frac{x}{1+x^3}}\,dx = \frac{2}{3H_0\sqrt{\Omega_\Lambda}}\operatorname{arcsinh}(x^{3/2})$ ✓

this time, setting $t(0)=0$ sets the constant to zero.

Baumann 2.177 gives $a(t) = \left(\frac{\Omega_m}{\Omega_\Lambda}\right)^{1/3}\sinh^{2/3}\left(\frac{3}{2}H_0\sqrt{\Omega_\Lambda}\,t\right)$

inverting... $\quad x^{3/2} = \sinh\left(\frac{3}{2}H_0\sqrt{\Omega_\Lambda}\,t\right)$

$$\operatorname{arcsinh}(x^{3/2}) = \frac{3}{2}H_0\sqrt{\Omega_\Lambda}\,t$$

$$\frac{2}{3H_0\sqrt{\Omega_\Lambda}}\operatorname{arcsinh}(x^{3/2}) = t \quad ✓$$

(b) $\Omega_m = 0.30966$, $\Omega_\Lambda = 0.68885$ $\quad a_{m\Lambda} = 0.766$. $\quad t_0$ is when $a=1$.

1/1

$t_0 = t(1) = t\left(x = \frac{1}{0.766}\right) = \frac{2}{3H_0\sqrt{\Omega_\Lambda}}\operatorname{arcsinh}\left(\left(\frac{1}{0.766}\right)^{3/2}\right) = 1.38\times10^{10}$ years ✓ (13.8 Gyr)

part 4. 2/2

$dt^2 = a^2 d\eta^2 \Rightarrow \frac{d\eta}{dt} = a^{-1}$ ✓ $\Rightarrow \eta(t) - \cancel{\eta(0)}^{0} = \int_0^t \frac{dt'}{a(t')}$ $\quad$ so $\eta_0 = \int_0^{t_0}\frac{dt}{a(t)}$.

using the $a(t)$ from above, $\eta_0 = \int_0^{13.8\ \mathrm{Gyr}}\left[\underbrace{a_{m\Lambda}}_{0.766}\sinh^{2/3}\left(\frac{3}{2}\underbrace{H_0\sqrt{\Omega_\Lambda}}_{5.743\times10^{-2}\ \mathrm{Gyr}}\,t\right)\right]^{-1}dt = 47.2\ \mathrm{Gyr} = 14500\ \mathrm{Mpc}$ ✓

PROBLEM 2:

part 1. $\chi(z) = \eta_0 - \eta(z)$

1/1

$$\eta(t) = \int_0^t \frac{dt}{a(t)} \rightarrow \eta(a) = \int_0^a \frac{1}{a}\frac{dt}{da}\,da = \int_0^a \frac{da}{a\,\dot a(a)}$$ ✓

$$\left(\frac{\dot a}{a}\right)^2 = H_0^2\left[\Omega_r a^{-4} + \Omega_m a^{-3} + \Omega_k a^{-2} + \Omega_\Lambda\right] \rightarrow \dot a = H_0\left(\Omega_r a^{-2} + \Omega_m a^{-1} + \Omega_k a^0 + \Omega_\Lambda a^2\right)^{1/2}$$

$$\rightarrow \eta(a) = \int_0^a \frac{da}{H_0\, a\,(\Omega_r a^{-2} + \Omega_m a^{-1} + \Omega_k a^0 + \Omega_\Lambda a^2)^{1/2}}$$ ✓

$$a = \frac{1}{1+z} \quad \text{so} \quad \chi(z) = \eta_0 - H_0^{-1}\int_0^{\frac{1}{1+z}} \frac{da}{a\,(\Omega_r a^{-2} + \Omega_m a^{-1} + \Omega_k a^0 + \Omega_\Lambda a^2)^{1/2}}$$

$$= H_0^{-1}\int_{1/1+z}^{1} \frac{da}{a\,(\Omega_r a^{-2} + \Omega_m a^{-1} + \Omega_k a^0 + \Omega_\Lambda a^2)^{1/2}}$$ ✓

note that this is a function of z since the limit is $1/1+z$. I didn't change integration variable to z.

part 2. see notebook.

3/3

part 3. $\Omega_k = \frac{-k}{(R_0 H_0)^2} \Rightarrow R_0 = \frac{1}{H_0\sqrt{|\Omega_k|}}$ for $k \neq 0$, undefined if $\Omega_k = k = 0$.

4/4

$$S_k = R_0 \begin{cases} \sin(\chi/R_0) & \text{if } k=1 \\ \chi/R_0 & k=0 \\ \sinh(\chi/R_0) & k=-1 \end{cases} = \begin{cases} \frac{1}{H_0\sqrt{|\Omega_k|}}\sin\left(H_0\sqrt{|\Omega_k|}\,\chi\right) & k=1 \\ \chi & k=0 \\ \frac{1}{H_0\sqrt{|\Omega_k|}}\sinh\left(H_0\sqrt{|\Omega_k|}\,\chi\right) & k=-1 \end{cases}$$

$$d_L = (1+z)\,S_k(\chi(z)), \qquad d_A = (1+z)^{-1}\,S_k(\chi(z))$$

see notebook.

part 4. taking $\Omega_k = 0$, $z \ll z_{eq}$ $(a \gg a_{eq})$

1/1

in this case, $\chi(z) = H_0^{-1}\int_{\frac{1}{1+z}}^{1} \frac{da}{a\,(\Omega_m a^{-1} + \Omega_\Lambda a^2)^{1/2}}$

then $d_L(z) = (1+z)\,S_0(\chi(z)) = (1+z)\,\chi(z) = H_0^{-1}(1+z)\int_{\frac{1}{1+z}}^{1} \frac{da}{a\,(\Omega_m a^{-1} + \Omega_\Lambda a^2)^{1/2}}$ ✓

using our usual tricks, $a_{m\Lambda} = \left(\frac{\Omega_m}{\Omega_\Lambda}\right)^{1/3}$ and $x = a/a_{m\Lambda}$, we can rewrite the integrand as

$$\frac{da}{a\,(\Omega_m a^{-1} + \Omega_\Lambda a^2)^{1/2}} = \frac{da}{a\sqrt{\Omega_\Lambda}\,a_{m\Lambda}\sqrt{x^{-1} + x^2}} = \frac{1}{\sqrt{\Omega_\Lambda}\,a_{m\Lambda}}\frac{dx}{\sqrt{x + x^4}}$$

then $$d_L(z) = H_0^{-1}\frac{1+z}{a_{m\Lambda}\sqrt{\Omega_\Lambda}}\int_{1/a_{m\Lambda}(1+z)}^{1/a_{m\Lambda}} \frac{dx}{\sqrt{x + x^4}}$$

$$= H_0^{-1}\underbrace{\frac{1+z}{\sqrt{\Omega_\Lambda}}\left[(1+z)\,{}_2F_1\left(\tfrac{1}{3},\tfrac{1}{2},\tfrac{4}{3},-a_{m\Lambda}^3(1+z)^3\right) - {}_2F_1\left(\tfrac{1}{3},\tfrac{1}{2},\tfrac{4}{3},-a_{m\Lambda}^3\right)\right]}_{\equiv F(z;\,\Omega_\Lambda)}$$

where ${}_2F_1$ is the hypergeometric function

in the $\Omega_r = \Omega_k = 0$ limit, we can take $\Omega_m = 1 - \Omega_\Lambda$ and $a_{m\Lambda}^3 = \frac{1}{\Omega_\Lambda} - 1$

see notebook.

part 5.

(a) $\chi^2(\Omega_\Lambda, H_0, M) = \sum_i \sigma_i^{-2} \left[5 \log_{10} F(z_i ; \Omega_\Lambda) - \mu_i + K(H_0, M) \right]^2$

2/2

we seek $\tilde{K} = \underset{K}{\operatorname{argmin}} \; \chi^2(\Omega_\Lambda, K) \Big|_{\Omega_\Lambda}$.

$$\frac{\partial}{\partial K} \chi^2 = \sum_i \sigma_i^{-2} \, 2 \left[5 \log F_i - \mu_i + \tilde{K} \right] = 0 \qquad \text{letting } F_i \equiv F(z_i ; \Omega_\Lambda).$$

$$\rightarrow \sum_i \sigma_i^{-2} \left(5 \log F_i - \mu_i \right) + \tilde{K} \sum_i \sigma_i^{-2} = 0$$

$$\rightarrow \tilde{K} = \frac{-1}{\sum_i \sigma_i^{-2}} \sum_i \sigma_i^{-2} \left(5 \log F_i - \mu_i \right) \checkmark$$

plugging this in,

$$\tilde{\chi}^2(\Omega_\Lambda) = \sum_i \sigma_i^{-2} \left[5 \log F(z_i ; \Omega_\Lambda) - \mu_i + \tilde{K} \right]^2$$

$$= \sum_i \sigma_i^{-2} \left(5 \log F(z_i ; \Omega_\Lambda) - \mu_i \right)^2 + \tilde{K}^2 \sum_i \sigma_i^{-2} + 2 \tilde{K} \sum_i \sigma_i^{-2} \left(5 \log F(z_i ; \Omega_\Lambda) - \mu_i \right)$$

$$= \sum_i \sigma_i^{-2} \left(5 \log F_i - \mu_i \right)^2 + \frac{1}{\sum_i \sigma_i^{-2}} \left[\sum_i \sigma_i^{-2} \left(5 \log F_i - \mu_i \right) \right]^2$$

$$- \frac{2}{\sum_i \sigma_i^{-2}} \sum_i \sigma_i^{-2} \left(5 \log F_i - \mu_i \right) \sum_i \sigma_i^{-2} \left(5 \log F_i - \mu_i \right)$$

$$= \sum_i \sigma_i^{-2} \left(5 \log F_i - \mu_i \right)^2 - \frac{1}{\sum_i \sigma_i^{-2}} \left[\sum_i \sigma_i^{-2} \left(5 \log F_i - \mu_i \right) \right]^2 \checkmark$$

(b) condensing notation, $\tilde{\chi}^2 = \sum_i \sigma_i^{-2} \Delta_i^2 - \frac{1}{\sum_i \sigma_i^{-2}} \left(\sum_i \sigma_i^{-2} \Delta_i \right)^2$

4/4

homework2

February 24, 2025

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
import astropy.constants as C
import astropy.units as U

import chic

from astropy.cosmology import Planck18
from scipy.integrate import quad
```

1 Problem 1

1.1 Part 2(c)

[2]:
```
H0 = Planck18.H0
Om = Planck18.Om0
Or = Planck18.Ogamma0 * 1.68

print("Constants:")
print(f"    H_0 = {H0}")
print(f"Omega_m = {Om}")
print(f"Omega_r = {Or:.3e}")

a_eq = Or / Om
t_eq = 2 * (2 - np.sqrt(2)) / 3 * a_eq**2 / (H0 * np.sqrt(Or))
t_eq = t_eq.to(U.yr)

print()
print("Calculations:")
print(f"   a_eq = {a_eq:.3e}")
print(f"   t_eq = {t_eq:.3e}")
```

```
Constants:
    H_0 = 67.66 km / (Mpc s)
Omega_m = 0.30966
Omega_r = 9.075e-05

Calculations:
```

```
a_eq = 2.931e-04
t_eq = 5.089e+04 yr
```

1.2 Part 3(b)

[3]:
```
H0 = Planck18.H0
Om = Planck18.Om0
OL = Planck18.Ode0

print("Constants:")
print(f"    H_0 = {H0}")
print(f"Omega_m = {Om}")
print(f"Omega_L = {OL:.5f}")

a_mL = (Om / OL)**(1/3)
x = 1 / a_mL
t_0 = 2 / (3 * H0 * np.sqrt(OL)) * np.arcsinh(x ** (3/2))
t_0 = t_0.to(U.yr)

print()
print("Calculations:")
print(f"   a_mL = {a_mL:.3f}")
print(f"    t_0 = {t_0:.3e}")
```

```
Constants:
    H_0 = 67.66 km / (Mpc s)
Omega_m = 0.30966
Omega_L = 0.68885

Calculations:
   a_mL = 0.766
    t_0 = 1.381e+10 yr
```

1.3 Part 4

[4]:
```
def a(t):
    x = H0 * np.sqrt(OL) * t
    x = x.to(U.dimensionless_unscaled).value
    return a_mL * np.sinh(1.5 * x)**(2/3)

def integrand(t_in_yr):
    return 1 / a(t_in_yr * U.yr)

eta_0, err = quad(integrand, 0, t_0.value)
eta_0 = eta_0 * U.yr
print(f"eta_0 = {eta_0.to(U.Gyr):.1f}")
print(f"      = {(eta_0 * C.c).to(U.Mpc):.0f}")
```

```
eta_0 = 47.2 Gyr
      = 14459 Mpc
```

2 Problem 2

2.1 Part 2

$$\eta(z) = H_0^{-1} \int_0^{1/(1+z)} \frac{\mathrm{d}a}{a\sqrt{\Omega_r a^{-2} + \Omega_m a^{-1} + \Omega_k a^0 + \Omega_\Lambda a^{+2}}}$$

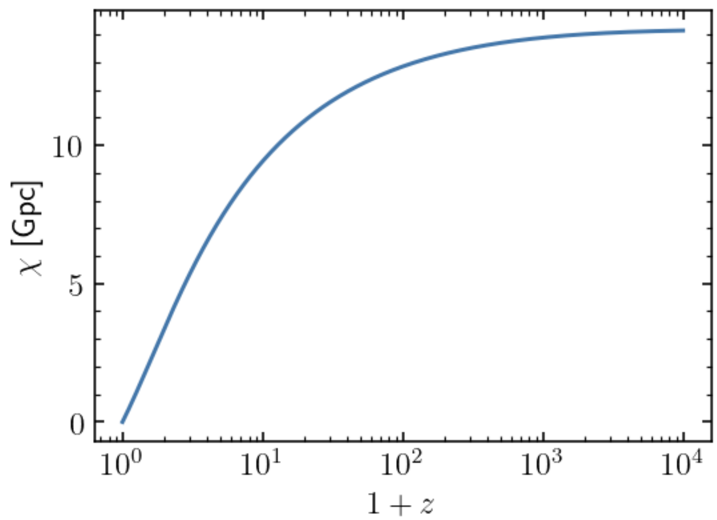
$$\implies \chi(z) = \eta(0) - \eta(z) = H_0^{-1} \int_{1/(1+z)}^{1} \frac{\mathrm{d}a}{a\sqrt{\Omega_r a^{-2} + \Omega_m a^{-1} + \Omega_k a^0 + \Omega_\Lambda a^{+2}}}$$

[5]:
```
def chi(
    z,
    H0 = Planck18.H0,
    Or = Planck18.Ogamma0 * 1.68,
    Om = Planck18.Om0,
    Ok = Planck18.Ok0,
    OL = Planck18.Ode0,
):
    def H0_eta(z):
        def integrand(a):
            return np.pow(a**2 * (
                Or * a**-2
                + Om * a**-1
                + Ok
                + OL * a**2
            ), -0.5)
        return quad(integrand, 1 / (1 + z), 1)[0]

    return (C.c / H0).to(U.Gpc) * np.array([H0_eta(_z) for _z in np.
 ↪atleast_1d(z)])

chi(0)
```

[5]: [0] Gpc

[6]:
```
zs = np.geomspace(1, 1e4 + 1, 1_000) - 1
chis = chi(zs)

plt.figure(figsize=(4, 3))
plt.plot(1 + zs, chis)
plt.xscale('log')
plt.yscale('linear')
plt.xlabel("$1 + z$")
plt.ylabel("$\\chi$ [Gpc]")
```

```
plt.tight_layout()
plt.show()
```



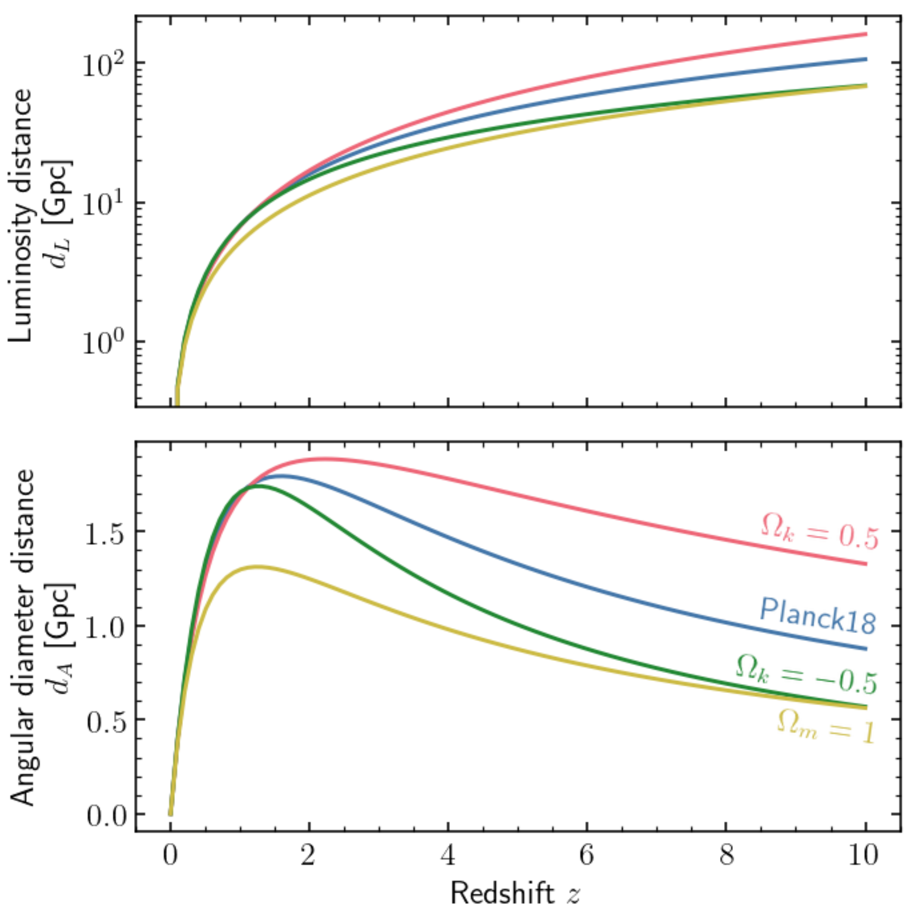
2.2 Part 3

[7]:
```
def S_k(chi, H0, Ok):
    alpha = (H0 * np.sqrt(np.abs(Ok)) / C.c).to(1 / U.Gpc)
    alpha_chi = (alpha * chi).to(U.dimensionless_unscaled).value

    if Ok < 0:    # k = +1
        return np.sin(alpha_chi) / alpha
    elif Ok > 0:  # k = -1
        return np.sinh(alpha_chi) / alpha
    else:         # k = 0
        return chi

def luminosity_distance(
    z,
    H0 = Planck18.H0,
    Or = Planck18.Ogamma0 * 1.68,
    Om = Planck18.Om0,
    Ok = Planck18.Ok0,
    OL = Planck18.Ode0,
):
    z_arr = np.atleast_1d(z)
    chi_arr = chi(z_arr, H0=H0, Or=Or, Om=Om, Ok=Ok, OL=OL)
    S_k_arr = S_k(chi_arr, H0, Ok)
    return S_k_arr * (1 + z_arr)
```

```python
def ang_diameter_distance(
    z,
    H0 = Planck18.H0,
    Or = Planck18.Ogamma0 * 1.68,
    Om = Planck18.Om0,
    Ok = Planck18.Ok0,
    OL = Planck18.Ode0,
):
    z_arr = np.atleast_1d(z)
    chi_arr = chi(z_arr, H0=H0, Or=Or, Om=Om, Ok=Ok, OL=OL)
    S_k_arr = S_k(chi_arr, H0, Ok)
    return S_k_arr / (1 + z_arr)
```

[8]:
```python
zs = np.linspace(0, 10, 100)

fig, axs = plt.subplots(2, 1, figsize=(5, 5), sharex=True)

for cosmo_kw in [
    dict(
        Ok = 0,
        H0 = Planck18.H0,
        Or = Planck18.Ogamma0 * 1.68,
        Om = Planck18.Om0,
        OL = Planck18.Ode0,
    ),
    dict(
        Ok = +0.5,
        H0 = Planck18.H0,
        Or = 0.5 * Planck18.Ogamma0 * 1.68,
        Om = 0.5 * Planck18.Om0,
        OL = 0.5 * Planck18.Ode0,
    ),
    dict(
        Ok = -0.5,
        H0 = Planck18.H0,
        Or = 1.5 * Planck18.Ogamma0 * 1.68,
        Om = 1.5 * Planck18.Om0,
        OL = 1.5 * Planck18.Ode0,
    ),
    dict(
        Ok = 0,
        H0 = Planck18.H0,
        Or = 0,
        Om = 1,
        OL = 0,
    ),
]:
```

```
    axs[0].plot(zs, luminosity_distance(zs, **cosmo_kw))
    axs[1].plot(zs, ang_diameter_distance(zs, **cosmo_kw))

axs[0].set_yscale('log')

axs[1].text(1 - 0.025, 0.49, "Planck18", color='C0',
            transform=axs[1].transAxes, ha='right', va='bottom', rotation=-8)
axs[1].text(1 - 0.025, 0.71, "$\\Omega_k = 0.5$", color='C1',
            transform=axs[1].transAxes, ha='right', va='bottom', rotation=-8)
axs[1].text(1 - 0.025, 0.34, "$\\Omega_k = -0.5$", color='C2',
            transform=axs[1].transAxes, ha='right', va='bottom', rotation=-8)
axs[1].text(1 - 0.025, 0.32, "$\\Omega_m = 1$", color='C3',
            transform=axs[1].transAxes, ha='right', va='top', rotation=-8)

axs[1].set_xlabel('Redshift $z$')

axs[0].set_ylabel('Luminosity distance\n$d_L$ [Gpc]')
axs[1].set_ylabel('Angular diameter distance\n$d_A$ [Gpc]')

fig.tight_layout()
plt.show()
```



looks right!

2.3 Part 4

$$
\begin{aligned}
F(z;\Omega_\Lambda) &= (1+z)\int_{1/(1+z)}^{1} \frac{\mathrm{d}a}{a\sqrt{\Omega_m a^{-1} + \Omega_\Lambda a^{+2}}} \\
&= \frac{1+z}{a_{m\Lambda}\sqrt{\Omega_\Lambda}} \int_{1/a_{m\Lambda}(1+z)}^{1/a_{m\Lambda}} \frac{\mathrm{d}x}{\sqrt{x+x^4}} \\
&= \frac{1+z}{\sqrt{\Omega_\Lambda}}\Big[(1+z)\,{}_2F_1\big(\tfrac{1}{3},\tfrac{1}{2},\tfrac{4}{3},-a_{m\Lambda}^3(1+z)^3\big) - {}_2F_1\big(\tfrac{1}{3},\tfrac{1}{2},\tfrac{4}{3},-a_{m\Lambda}^3\big)\Big]
\end{aligned}
$$

integration performed with Mathematica

because we're assuming that radiation is irrelevant and $k = 0$, I think we assume that $\Omega_m + \Omega_\Lambda = 1$ and thus $a_{m\Lambda} = ((1-\Omega_\Lambda)/\Omega_\Lambda)^{1/3}$.

[9]:
```
from scipy.special import hyp2f1

def F(z, OL, Om=Planck18.Om0):
    a_mL = ((1 - OL) / OL)**(1/3)
    return (1 + z) * (
        (1 + z) * hyp2f1(1./3., 1./2., 4./3., -a_mL**3 * (1 + z)**3)
                - hyp2f1(1./3., 1./2., 4./3., -a_mL**3)
    ) / np.sqrt(OL)
```

I also tried doing the integral numerically and verified the above formula and implementation

that code, for posterity:

```
def F_num(z, OL, Om=Planck18.Om0):
    a_mL = ((1 - OL) / OL)**(1/3)

    def integrand(x):
        return 1 / np.sqrt(x + x**4)

    integral = np.array([quad(integrand, 1 / (a_mL * (1 + z)), 1 / a_mL)[0] for z in np.atleas
    return (1 + z) * integral / (a_mL * np.sqrt(OL))

rng = np.random.default_rng()
zs = rng.uniform(low=0, high=2, size=(10,))
OLs = rng.uniform(low=0, high=1, size=(10,))
for z in zs:
    for OL in OLs:
        print(f"{F(z, OL) - F_num(z, OL).squeeze():+.2e}, ", end="")
```

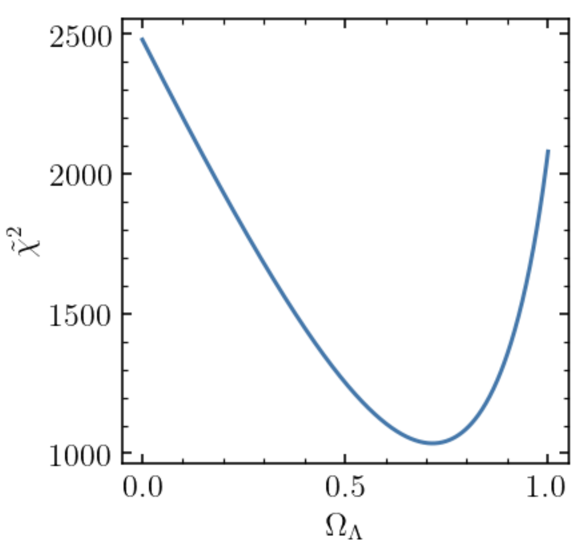
2.4 Part 5(b)

the code below downloads the data as `data.txt` if it does not already exist in the current directory

[10]:
```
![ ! -f data.txt ] && \
curl -o data.txt https://archive.stsci.edu/hlsps/ps1cosmo/scolnic/
 ↪hlsp_ps1cosmo_panstarrs_gpc1_all_model_v1_lcparam-full.txt
```

[11]:
```
data = np.loadtxt(
    'data.txt',
    dtype={
        'names': ('name', 'zcmb', 'zhel', 'dz', 'mb', 'dmb'),
        'formats': ('S16', '<f8', '<f8', '<f8', '<f8', '<f8'),
    }
)

z_i = data['zcmb']
mu_i = data['mb']
sigma_i = data['dmb']
```

[12]:
```
def chi_sq(OL):
    Delta_i = 5 * np.log10(F(z_i, OL)) - mu_i
    term1 = (sigma_i**-2 * Delta_i**2).sum()
    term2 = (sigma_i**-2 * Delta_i).sum()**2 / (sigma_i**-2).sum()
    return term1 - term2

OLs = np.linspace(0, 1, 1_000)[1:]
chi_sqs = np.array([chi_sq(OL) for OL in OLs])

plt.figure(figsize=(3, 3))
plt.plot(OLs, chi_sqs)
plt.xlabel('$\\Omega_{\\Lambda}$')
plt.ylabel('$\\tilde \\chi^2$')
plt.show()
```



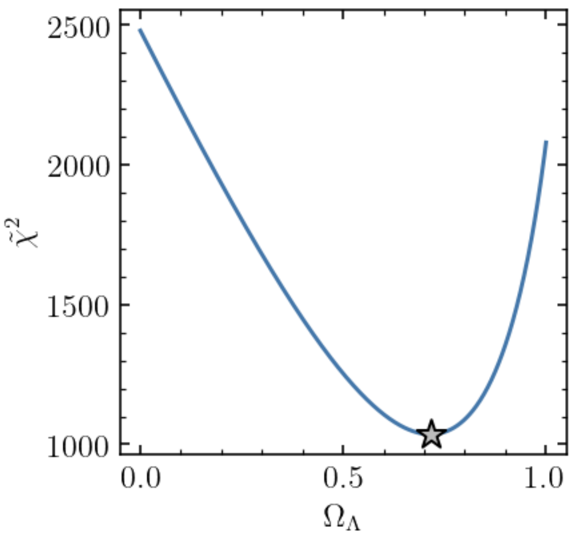
[13]:

```
from scipy.optimize import minimize_scalar

res = minimize_scalar(chi_sq, bounds=(1e-10, 1))
if not res.success:
    raise RuntimeError("did not converge!" + res.message)

print(f"minimization result:")
print(f"  Omega_L = {res.x:.3f}")
print(f"   chi_sq = {res.fun:.2e}")

print()
print(f"Planck18 reference:")
print(f"  Omega_L = {Planck18.Ode0:.3f}")
```

```
minimization result:
  Omega_L = 0.715
   chi_sq = 1.04e+03

Planck18 reference:
  Omega_L = 0.689
```

[14]:

```
def chi_sq(OL):
    Delta_i = 5 * np.log10(F(z_i, OL)) - mu_i
    term1 = (sigma_i**-2 * Delta_i**2).sum()
    term2 = (sigma_i**-2 * Delta_i).sum()**2 / (sigma_i**-2).sum()
    return term1 - term2

OLs = np.linspace(0, 1, 1_000)[1:]
chi_sqs = np.array([chi_sq(OL) for OL in OLs])

plt.figure(figsize=(3, 3))
plt.plot(OLs, chi_sqs)
plt.scatter([res.x], [res.fun], marker='*', s=128, facecolor='C6',␣
 ↪edgecolor='k', zorder=999)
plt.xlabel('$\\Omega_{\\Lambda}$')
plt.ylabel('$\\tilde \\chi^2$')
plt.show()
```


2500
2000
1500
1000
$\tilde{\chi}^2$
0.0
0.5
1.0
Ω_Λ

```
[ ]:
```