PHYS 600: Homework 3

Yarone Tokayer

October 8, 2023

Problem 1 Reviewing the Background

Density Parameters In general, $\rho_{\rm m}$ scales as $(1+z)^3$ and ρ_Λ is constant. We can now evaluate the desired quantities:

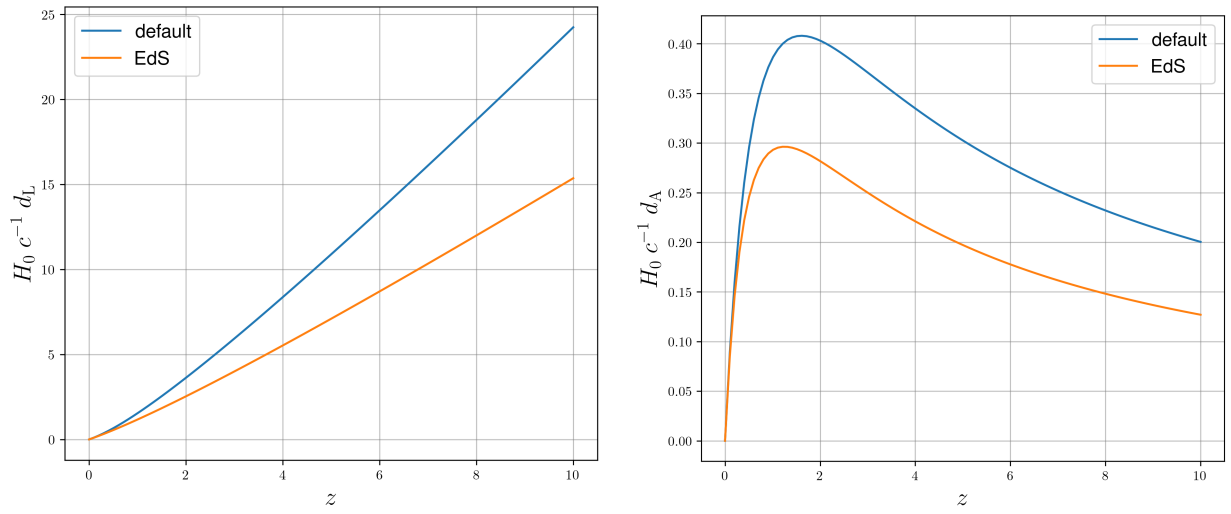
$$\begin{aligned}
\Omega_{\rm m}(z=0.5) &= \frac{\rho_{\rm m}(z)}{\rho_{\rm c}(z)} \\
&= \Omega_{\rm m,0}(1+z)^3\frac{H_0^2}{H^2(z)} \\
&= \frac{\Omega_{\rm m,0}(1+z)^3}{\Omega_{\rm m,0}(1+z)^3+\Omega_{\rm r,0}(1+z)^4+\Omega_{\rm k,0}(1+z)^2+\Omega_\Lambda} \\
&= \frac{0.3\cdot(1+0.5)^3}{0.3\cdot(1+0.5)^3+0+0+0.7} = \boxed{0.591} \\
\Omega_\Lambda(z=0.5) &= \frac{\rho_\Lambda}{\rho_{\rm c}(z)} \\
&= \Omega_\Lambda\frac{H_0^2}{H^2(z)} \\
&= \frac{\Omega_\Lambda}{\Omega_{\rm m,0}(1+z)^3+\Omega_{\rm r,0}(1+z)^4+\Omega_{\rm k,0}(1+z)^2+\Omega_\Lambda} \\
&= \frac{0.7}{0.3\cdot(1+0.5)^3+0+0+0.7} = \boxed{0.409}
\end{aligned}$$

We have approximated $\rho_{\rm r}=0$ at all times extending back to at least $z=0.5$, and assumed a flat universe with $\Omega_{\rm k}(z)=0$. These results are confirmed with `Astropy` (see appendix).

Luminosity and Angular Diameter Distances The luminosity distance from us as a particular redshift, $d_{\rm L}$ is given by

$$d_{\rm L} = (1+z)d_{\rm M}; \qquad d_{\rm M} = \int_0^z \frac{{\rm d}z'}{H(z')}$$

$$\implies d_{\rm L}(z) = (1+z)\int_0^z \frac{{\rm d}z'}{H_0\left[\Omega_{\rm m,0}(1+z')^3+\Omega_{\rm r,0}(1+z')^4+\Omega_{\rm k}(1+z')^2+\Omega_\Lambda\right]^{1/2}}$$



(a) (b)

Figure 1: The luminosity distance and angular diameter distance as a function of redshift for two different cosmologies.

$$\approx (1+z)\int_0^z \frac{\mathrm{d}z'}{H_0\left[\Omega_{\mathrm{m},0}(1+z')^3+\Omega_\Lambda\right]^{1/2}}$$

The angular diameter distance is given by

$$\begin{aligned} d_\mathrm{A} &= \frac{d_\mathrm{M}}{(1+z)} \\ &\approx (1+z)^{-1}\int_0^z \frac{\mathrm{d}z'}{H_0\left[\Omega_{\mathrm{m},0}(1+z')^3+\Omega_\Lambda\right]^{1/2}} \end{aligned}$$

We have approximated $\rho_\mathrm{r} = 0$ at all times extending back to at least $z = 10$, and assumed a flat universe with $\Omega_\mathrm{k}(z) = 0$.

To plot these, we use the `numpy.trapz` method for numerical integration, which uses a trapezoidal approximation.

To make the quantities dimensionless, we will compute $H_0 d_\mathrm{L}$ and $H_0 d_\mathrm{A}$. See plots in Fig. 1. Code can be found in the appendix.

Note in Fig. 1b, that there is a critical point at $z \approx 0.7$. This corresponds to the epoch when the Universe goes from matter dominated to Λ dominated.

Looking Back In general, we can find the lookback time from the comoving distance (we will explicitly include c to make getting units of time easier):

$$\mathrm{d}d_\mathrm{M} = \frac{c}{H(z')}\,\mathrm{d}z$$

Now, in each differential interval $\mathrm{d}z$, the distance light must travel is $(1+z)^{-1}\,\mathrm{d}d_{\mathrm{M}}$. We get:

$$t_{\mathrm{L}} = \frac{1}{H_0}\int_0^z \frac{\mathrm{d}z'}{(1+z')\left[\Omega_{\mathrm{m},0}(1+z')^3+\Omega_\Lambda\right]^{1/2}}.$$

We use a bisection method to invert this function and solve for z numerically. See the code in the appendix. Our results is $\boxed{z = 1.86}$. This is verified using Astropy.

A Λ-dominated Universe We will use the first Friedmann equation to solve for $a(t)$:

$$\begin{aligned}\left(\frac{\dot{a}}{a}\right)^2 &= \frac{\Lambda}{3}\\ \implies \frac{\mathrm{d}a}{\mathrm{d}t} &= \sqrt{\frac{\Lambda}{3}}a\\ \implies a(t) &= Ae^{\sqrt{\Lambda/3}\ t}\end{aligned}$$

We throw away the negative square root, since we observe $\dot{a}$ to be positive. To find the age of the Universe, we look for t such that $a(t) = 0$. But there is no such finite value in this model (i.e., it requires $t = -\infty$). This implies an infinitely old Universe.

. . . . . .

Problem 2 Massive Neutrinos

- In general, the energy density of a species in an equilibrium gas is given by (e.g., 3.9 in Baumann):

$$\rho = \frac{g}{(2\pi)^3}\int \mathrm{d}^3p\ f(p,T)E(p)$$

Where $E(p) = \sqrt{m^2+p^2}$ and $f(p,T)$ is the distribution function over phase space. Using the Fermi-Dirac distribution (neutrinos are Fermions), and setting the chemical potential to zero for very early times, we get

$$\rho_\nu = \frac{g_\nu}{2\pi^2}T_\nu^4\int_0^\infty \mathrm{d}\xi \frac{\xi^2\sqrt{\xi^2+m_\nu^2/T_\nu^2}}{\exp\left[\sqrt{\xi^2+m_\nu^2/T_\nu^2}\right]+1}$$

with $\xi = p/T$. For relativistic neutrinos, we have $m_\nu \ll T\nu$, and we will use $g = 2$ for the spin degrees of freedom for a single neutrino species (e.g. `https://physics.stackexchange.com/questions/335061/degrees-of-freedom-of-neutrinos`). We get

$$\rho_\nu = \frac{T_\nu^4}{2\pi^2}\int_0^\infty \mathrm{d}\xi \frac{\xi^2\sqrt{\xi^2+m_\nu^2/T_\nu^2}}{e^\xi+1}$$

- In general, we have the expansion $\sqrt{1+\epsilon^2} = 1 + \frac{\epsilon^2}{2} - \cdots$. We manipulate as follows:

$$\begin{aligned}
\xi^2\sqrt{\xi^2 + m_\nu^2/T_\nu^2} &= \xi^3\sqrt{1 + \frac{m_\nu^2}{\xi^2 T_\nu^2}} \\
&= \xi^3\left[1 + \frac{m_\nu^2}{2\xi^2 T_\nu^2} + \cdots\right] \\
\implies \rho_\nu &\approx \frac{T_\nu^4}{2\pi^2}\int_0^\infty \mathrm{d}\xi \frac{\xi^3\left[1 + \frac{m_\nu^2}{2\xi^2 T_\nu^2}\right]}{e^\xi + 1} \\
&= \frac{T_\nu^4}{2\pi^2}\int_0^\infty \mathrm{d}\xi \frac{\xi^3}{e^\xi + 1} + \frac{T_\nu^4}{2\pi^2}\frac{m_\nu^2}{2T_\nu^2}\int_0^\infty \mathrm{d}\xi \frac{\xi}{e^\xi + 1} \\
&= \underbrace{\frac{T_\nu^4}{2\pi^2}\frac{7\pi^4}{120}}_{\equiv\rho_{\nu,0}} + \frac{T_\nu^4}{2\pi^2}\frac{m_\nu^2}{2T_\nu^2}\frac{\pi^2}{12} \\
&= \rho_{\nu,0}\left[1 + \frac{5}{7\pi^2}\frac{m_\nu^2}{T_\nu^2}\right]
\end{aligned}$$

  I used Wolfram Alpha to evaluate the integrals.

- Let's suppose that in order the be detectable, we require $\rho_{\nu,\mathrm{CMB}} \gtrsim 10\rho_{\nu,0}$. We will also assume that the relativistic approximation used above is still valid at the epoch of the CMB. We get

$$\begin{aligned}
1 + \frac{5}{7\pi^2}\frac{m_\nu^2}{T_\nu^2} &\gtrsim 10 \\
m_\nu^2 &\gtrsim \frac{63\pi^2}{5}T_\nu^2 \\
m_\nu &\gtrsim \sqrt{\frac{63}{5}}\pi T_\nu
\end{aligned}$$

  In class, we found that after neutrino decoupling, we have $T_\nu = (4/11)^{1/3}T_\gamma$:

$$\begin{aligned}
m_\nu &\gtrsim \sqrt{\frac{63}{5}}\pi\left(\frac{4}{11}\right)^{1/3} T_{\gamma,0}(1+z) \\
&\approx \sqrt{\frac{63}{5}}\pi\left(\frac{4}{11}\right)^{1/3} 0.235\text{ meV} \times 1000 \\
&\approx \boxed{1.87\text{ eV}}
\end{aligned}$$

- When a particle goes non-relativistic, its thermal energy becomes much smaller than its rest mass energy. We look for the redshift at which $m_\nu \sim T_\nu$:

$$T_\nu \sim m_\nu$$

$$
\begin{aligned}
T_{\nu,0}(1+z) &\sim m_\nu \\
\left(\frac{4}{11}\right)^{1/3} T_{\gamma,0}(1+z) &\sim m_\nu \\
\implies z_{\text{non-rel}} &\sim \left(\frac{11}{4}\right)^{1/3} \frac{m_\nu}{0.235 \text{ meV}} - 1
\end{aligned}
$$

- Our strategy will be to find n_ν at the redshift of decoupling from the photon bath, and then track its evolution in reference to photons, since we know that photons have remained relativistic since then. We know that neutrinos decoupled during radiation domination, so for Fermions we can use

$$
\begin{aligned}
n_{\nu,\text{dec}} &= \frac{3\zeta(3)}{4\pi^2} g_\nu T_{\text{dec}}^3 \\
&= \frac{3\zeta(3)}{2\pi^2} T_{\text{dec}}^3 \\
\implies n_{\nu,0} &= n_{\nu,\text{dec}}(1+z)^{-3} \\
&= \frac{3\zeta(3)}{2\pi^2} T_{\text{dec}}^3 (1+z)^{-3} \\
&= \frac{3\zeta(3)}{2\pi^2} T_{\nu,0}^3 (1+z)^3 (1+z)^{-3} \\
&= \frac{3\zeta(3)}{2\pi^2} T_{\nu,0}^3 \\
&= \frac{3\zeta(3)}{2\pi^2} \left(\frac{4}{11}\right) T_{\gamma,0}^3 \\
&= 8.62159 \times 10^{-13} \text{ eV}^3
\end{aligned}
$$

  From CMB measurements we used $T_{\gamma,0} = 0.235$ meV.

-

$$
\begin{aligned}
\Omega_{\nu,0} &= \frac{\rho_{\nu,0}}{\rho_{\text{c},0}} \\
&= \frac{m_\nu n_{\nu,0}}{\rho_{\text{c},0}} \\
&= m_\nu \frac{8.62159 \times 10^{-13} \text{ eV}^3}{8.07 \times 10^{-11} \text{ eV}^4 h^2} \\
\implies \Omega_{\nu,0} h^2 &= \frac{m_\nu}{93.6 \text{ eV}}
\end{aligned}
$$

. . . . . .

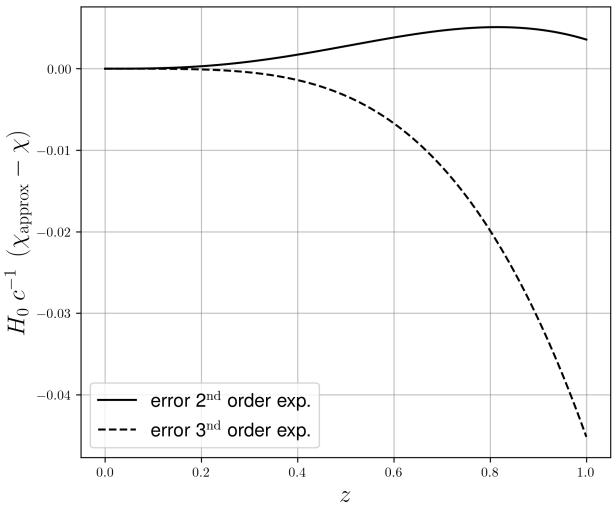


Figure 2: The luminosity distance and angular diameter distance as a function of redshift for two different cosmologies.

Problem 3 Measuring the Expansion History with Standard Candles

- In order to expand χ to third order in z, we expand the integrand to second order. Using Mathematica (see code in the appendix), we get:

$$\begin{aligned} \mathrm{d}\chi &= \frac{1}{H_0\left[\Omega_{\mathrm{m},0}(1+z)^3 + \Omega_{\Lambda,0} + (1-\Omega_{\mathrm{m},0}-\Omega_{\Lambda,0})(1+z)^2\right]^{1/2}} \\ &\approx \frac{1}{H_0}\left[1 + \frac{-2+2\Omega_{\Lambda,0}-\Omega_{\mathrm{m},0}}{2}z + \frac{8-20\Omega_{\Lambda,0}+12\Omega_{\Lambda,0}^2+4\Omega_{\mathrm{m},0}+3\Omega_{\mathrm{m},0}^2-12\Omega_{\Lambda,0}\Omega_{\mathrm{m},0}}{8}z^2\right] \end{aligned}$$

- Now we go back to Python to find the z value at which our third order approximation of χ diverges from the exact integral by 10%. We again implement a bisection method. See the Python code in the appendix.

  We find that this occurs at $\boxed{z = 1.204}$. Note however, that after about $z \approx 0.45$, the second order approximation actually performs better than the third order approximation. The second order approximation doesn't diverge by 10% until about $z = 2.05$. See Fig. 2.

- We can use the same procedure to find where the first order approximation $\chi = \frac{z}{H_0}$ (which does not involved the cosmological density parameters) differs by the exact integral by at least 10%. See the Python code in the appendix. We find that this occurs at $z = 0.386$.

  Our statement can be explained using two approaches:

  Intuitively On an intuitive level, at low redshifts (the local universe), the Hubble parameter is roughly equal to what it is at present day. We are hardly probing

the epochs at which it would be different, so we should only expect to find a constant relationship between recession velocity and distance. Since the cosmological parameters are related to H's *evolution*, we do not probe them in the local universe.

Mathematically Using the expansion we just derived, we find that only for z not close to 0 does χ significantly deviate from the first order expansion, which is sensitive only to H_0. Cosmological density parameters are only probed for higher z.

- As we increase the redshift, the deviation of the first order expansion (i.e., $\chi \approx z$) relative to the second order expansion becomes significant, so we start to probe that term, which is proportional to $-2 + 2\Omega_{\Lambda,0} - \Omega_{\mathrm{m},0}$. This is to combination of $\Omega_{\Lambda,0}$ and $\Omega_{\mathrm{m},0}$ that we are sensitive to. To break the degeneracy between them, we need to measure χ at higher z so that higher order deviations become significant. In those terms, we have different combinations of the two variables, so that we have at least two independent equations to solve for the two unknowns.

- To put a lower bound on σ_i^2 for $i = H_0, \Omega_{\mathrm{m},0}, \Omega_{\Lambda,0}$, we compute the associated Fischer matrix elements. We have for our data $(z_1, z_2, z_3, z_4) = (0.01, 0.1, 0.2, 0.3)$. We assume a Gaussian likelihood function with

$$\mu = \begin{pmatrix} \chi(z_1) \\ \vdots \\ \chi(z_4) \end{pmatrix} \qquad C = 0.01^2 \times \mathrm{diag}\left(\chi(z_1)^2, \cdots, \chi(z_4)^2\right)$$

$$\implies C^{-1} = 0.01^{-2} \times \mathrm{diag}\left(\chi(z_1)^{-2}, \cdots, \chi(z_4)^{-2}\right)$$

Since $z \leq 0.3$, we use our second-order approximation throughout:

$$\begin{aligned} \chi &\approx \frac{1}{H_0} \int_0^z \mathrm{d}z' \left[1 + \frac{-2 + 2\Omega_{\Lambda,0} - \Omega_{\mathrm{m},0}}{2} z'\right] \\ &= \frac{1}{H_0} \left[z + \frac{-2 + 2\Omega_{\Lambda,0} - \Omega_{\mathrm{m},0}}{4} z^2\right] \end{aligned}$$

We will compute $F_{ii} = \mu_{,i} C^{-1} \mu_{,i} + \mathrm{Tr}[C^{-1} C_{,i} C^{-1} C_{,i}]$. These derivatives can be calculated using Mathematica...

. . . . . .

A Python code

Phys 600: HW 3

Yarone Tokayer

October 8, 2023

In [2]:
```
# Install packages

from astropy.cosmology import FlatLambdaCDM
from astropy import units as u
import numpy as np
import matplotlib.pyplot as plt
```

In [3]:
```
plt.rcParams.update({
    "text.usetex": True,
    "font.family": "Helvetica"
})
```

Problem 1

In [4]:
```
# Set up an Astropy cosmology to check our answers

H0 = 70.0
Om0 = 0.3

cosmo = FlatLambdaCDM(H0=H0, Om0=Om0)

Ode0 = cosmo.Ode0
```

In [5]:
```
Ode0
```

Out[5]:
```
0.7
```

Density Parameters

Check our work for $\Omega_{\rm x}(z)$ using Astropy

In [6]:
```
z = 0.5

cosmo.Om(z), cosmo.Ode(z)
```

Out[6]:
```
(0.5912408759124087, 0.4087591240875912)
```

Agrees!

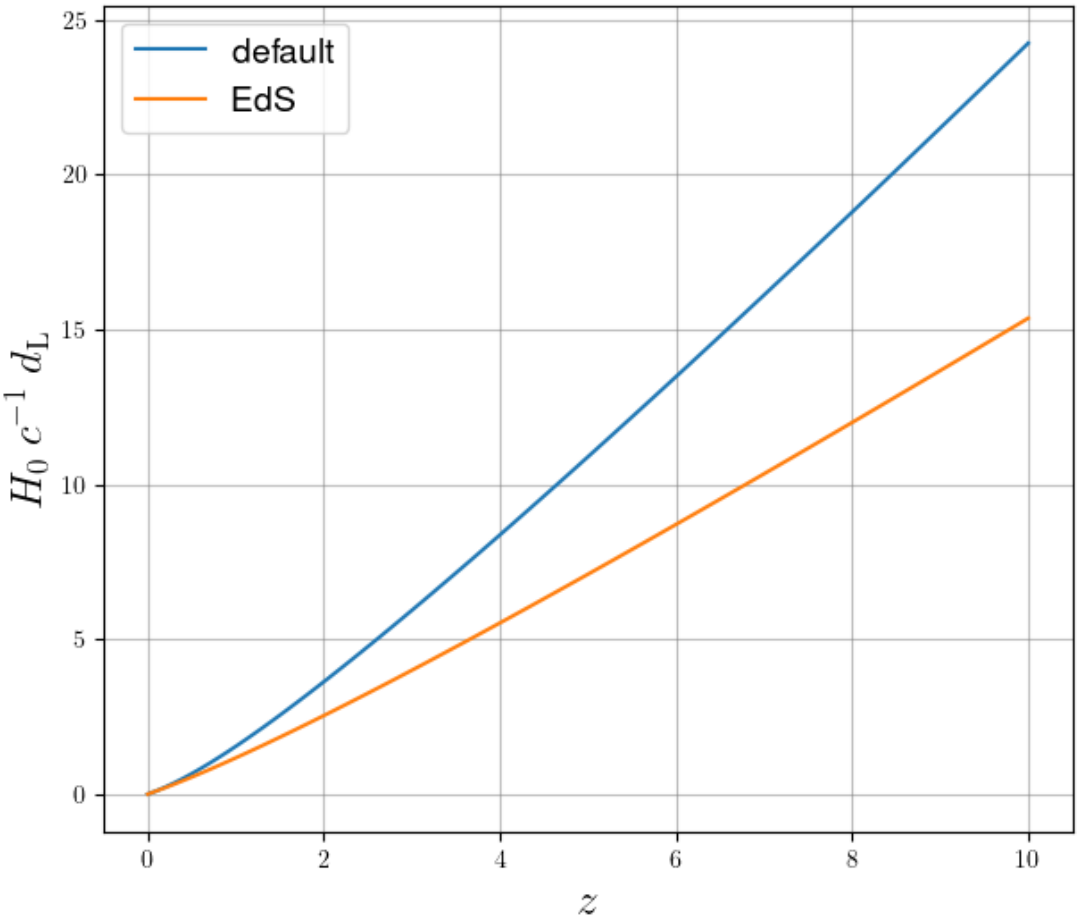
Luminosity and Angular Diameter Distances

In [7]:
```
res = 100 # Resolution of plot
z_axis = np.linspace(0, 10, res)

d_l = np.zeros(len(z_axis))
d_a = np.zeros(len(z_axis))

d_l_eds = np.zeros(len(z_axis))
d_a_eds = np.zeros(len(z_axis))
```

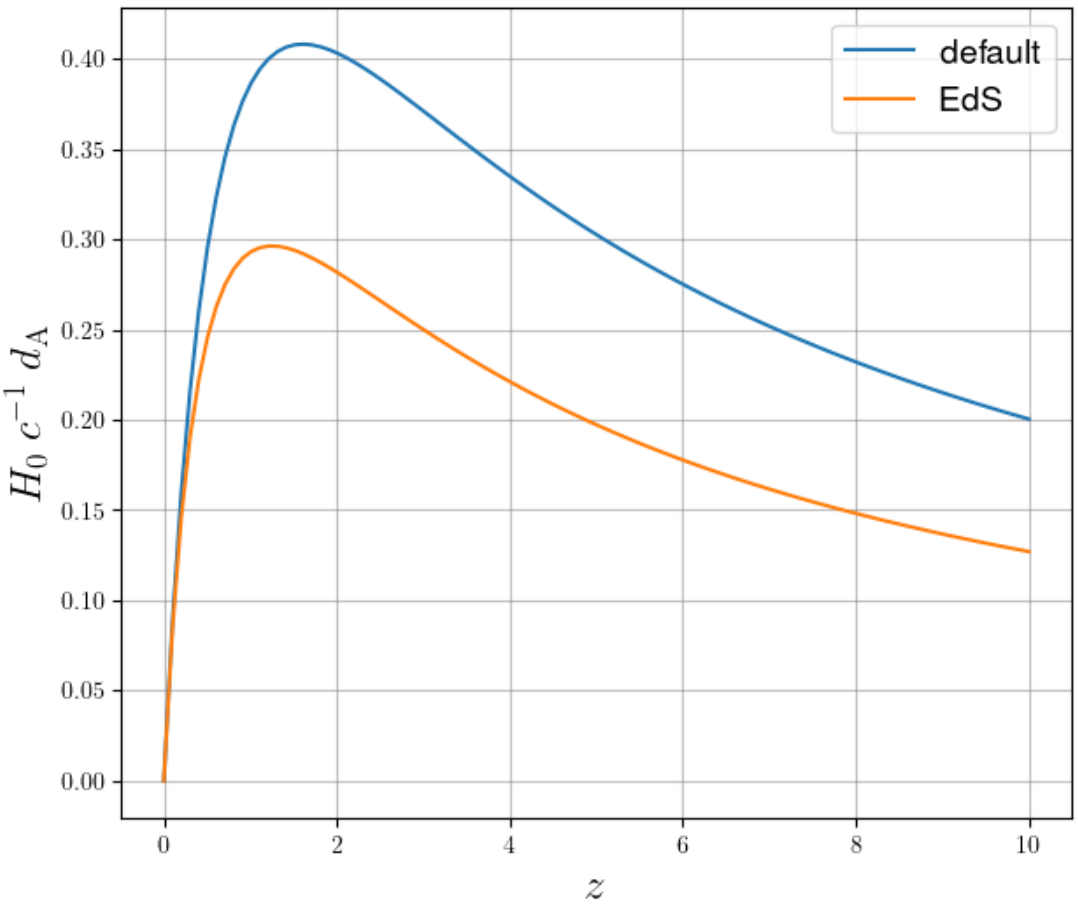
In [8]:
```
# Calculate d_l and d_a for the default cosmology

for i in range(res):
    z = z_axis[i]
    Ode0 = 0.7

    # Calculate comoving distance from z=0 to z
    z_array = np.linspace(0, z, 1000)
    integrand = 1 / ( np.sqrt(Om0 * (1 + z_array)**3 + Ode0) )
    d_m = np.trapz(integrand, x=z_array)

    # Convert to d_l and d_a
    d_l[i] = (1 + z) * d_m
    d_a[i] = d_m / (1 + z)
```

In [9]:
```
# Calculate d_l and d_a for a matter only universe (EdS)

for i in range(res):
    z = z_axis[i]

    # Calculate comoving distance from z=0 to z
    z_array = np.linspace(0, z, 1000)
    integrand = 1 / ( np.sqrt((1 + z_array)**3) ) # Om0=1, Omde0=0
    d_m = np.trapz(integrand, x=z_array)

    # Convert to d_l and d_a
    d_l_eds[i] = (1 + z) * d_m
    d_a_eds[i] = d_m / (1 + z)
```

In [10]:
```
fig, ax = plt.subplots(1, 1, figsize=(7,6))

ax.plot(z_axis, d_l, label='default')
ax.plot(z_axis, d_l_eds, label='EdS')

ax.set_axisbelow(True)
ax.xaxis.grid(color='gray', alpha=0.5, linestyle='-')
ax.yaxis.grid(color='gray', alpha=0.5, linestyle='-')

ax.set_xlabel('$z$', fontsize=18)
ax.set_ylabel(r'$ H_0\ c^{-1}\ d_\mathrm{L}$', fontsize=18)

ax.legend(fontsize=14)

fig.savefig('/Users/yaronetokayer/Yale Drive/Classes/PHYS 600/phys600 hw/phy
           dpi=300, bbox_inches='tight')
```



In [11]:
```
fig, ax = plt.subplots(1, 1, figsize=(7,6))

ax.plot(z_axis, d_a, label='default')
ax.plot(z_axis, d_a_eds, label='EdS')

ax.set_axisbelow(True)
ax.xaxis.grid(color='gray', alpha=0.5, linestyle='-')
ax.yaxis.grid(color='gray', alpha=0.5, linestyle='-')

ax.set_xlabel('$z$', fontsize=18)
ax.set_ylabel(r'$ H_0\ c^{-1}\ d_\mathrm{A}$', fontsize=18)

ax.legend(fontsize=14)

fig.savefig('/Users/yaronetokayer/Yale Drive/Classes/PHYS 600/phys600 hw/phy
           dpi=300, bbox_inches='tight')
```



Looking Back

In [12]:
```python
def t_l(z, h, Om0, Ode0):
    '''
    Function to compute the lookback time as a function of z

    Inputs:
    z - redshift
    h - H0/(100 km/s/Mpc)
    Om0 - Fractional matter density at z=0
    Ode0 - Fractional dark energy density at z=0

    Returns:
    t_l - lookback time in years
    '''

    H0 = h * 100 * u.km / u.s / u.Mpc

    z_array = np.linspace(0, z, 1000)

    integrand = 1 / H0 / ( 1 + z_array ) / np.sqrt(Om0 * ( 1 + z_array )**3
    return np.trapz(integrand, x=z_array).to(u.year)
```

In [13]:
```
def t_l_to_z(t_l_target, h, Om0, Ode0, tolerance=1e-6):
    '''
    Function to find redshift (z) given the lookback time (t_l) using a bise

    Inputs:
    t_l_target - Lookback time (astropy units expected)
    h - H0/(100 km/s/Mpc)
    Om0 - Fractional matter density at z=0
    Ode0 - Fractional dark energy density at z=0
    tolerance - Tolerance for the bisection method convergence

    Returns:
    z - Redshift corresponding to the given lookback time
    '''

    # Initialize the search interval for z
    z_low, z_high = 0.0, 1e3

    # Perform the bisection search
    while z_high - z_low > tolerance:
        z_mid = (z_low + z_high) / 2
        t_mid = t_l(z_mid, h, Om0, Ode0)
        if t_mid > t_l_target:
            z_high = z_mid
        else:
            z_low = z_mid

    # The bisection method has converged; return the redshift
    return z_mid
```

In [14]:
```
# Compute z for t_l = 1e10 years

h = 0.7
Om0 = 0.3
Ode0 = 0.7
t_l_target = 1e10 * u.year

t_l_to_z(t_l_target, h, Om0, Ode0)
```

Out[14]:
```
1.8555903807282448
```

Check our work for $t_{\rm t}$ using Astropy

In [15]:
```
z=1.8555903807282448
cosmo.lookback_time(z)
```

Out[15]: $9.9999956\ \mathrm{Gyr}$

We are correct to 3 significant figures, as required.

Problem 3

To what redshift is our expansion accurate to 10%? To answer this, we create two functions: one that computes the exact numerical integral $\chi(z)$, and another that computes the series expansion to third order. Then we use a bisection method to find the z at which the expansion differs by 10%. We also define a function to compute the second order approximation.

Define functions

In [16]:
```
def chi(z, Om0, Ode0):
    '''
    Function to compute the dimensionless comoving distance as a function of

    Inputs:
    z - redshift
    Om0 - Fractional matter density at z=0
    Ode0 - Fractional dark energy density at z=0

    Returns:
    chi - dimensionless comoving distance (H_0 \chi * c^-1)
    '''

    z_array = np.linspace(0, z, 1000)

    integrand = 1 / np.sqrt( Om0 * ( 1 + z_array )**3 + Ode0 + (1 - Om0 - Oc
    return np.trapz(integrand, x=z_array)
```

In [22]:
```
def chi_thirdorder(z, Om0, Ode0):
    '''
    Function to compute the approximate dimensionless comoving distance to t
    as a function of z

    Inputs:
    z - redshift
    Om0 - Fractional matter density at z=0
    Ode0 - Fractional dark energy density at z=0

    Returns:
    chi - dimensionless comoving distance (H_0 \chi * c^-1)
    '''

    z_array = np.linspace(0, z, 1000)

    integrand = (
        1
        + ( ( -2 + 2*Ode0 - Om0 ) / 2 ) * z_array
        + ( ( 8 - 20*Ode0 + 12*Ode0**2 + 4*Om0 + 3*Om0**2 - 12*Ode0*Om0 ) /
    )

    return np.trapz(integrand, x=z_array)

def chi_secondorder(z, Om0, Ode0):
    '''
```

```
    Function to compute the approximate dimensionless comoving distance to t
    as a function of z

    Inputs:
    z - redshift
    Om0 - Fractional matter density at z=0
    Ode0 - Fractional dark energy density at z=0

    Returns:
    chi - dimensionless comoving distance (H_0 \chi * c^-1)
    '''

    z_array = np.linspace(0, z, 1000)

    integrand = (
        1
        + ( ( -2 + 2*Ode0 - Om0 ) / 2 ) * z_array
    )

    return np.trapz(integrand, x=z_array)
```

Find the z at which the error exceed 10%

For the third order approximation:

In [18]:
```
# Initialize the search interval for z
z_low, z_high = 0.0, 2

tolerance = 1e-6 # Precision of z
target = 0.1 # Looking for delta to be within 10%

# Perform the bisection search
while z_high - z_low > tolerance:
    z_mid = (z_low + z_high) / 2
    delta_mid = np.abs(( chi_thirdorder(z_mid, Om0, Ode0) - chi(z_mid, Om0,
    if delta_mid > target:
        z_high = z_mid
    else:
        z_low = z_mid

z_mid
```

Out[18]:
```
1.2042932510375977
```

For the second order approximation:

In [63]:
```
# Initialize the search interval for z
z_low, z_high = 0.0, 2.5

tolerance = 1e-6 # Precision of z
target = 0.1 # Looking for delta to be within 10%

# Perform the bisection search
while z_high - z_low > tolerance:
```

```
    z_mid = (z_low + z_high) / 2
    delta_mid = np.abs(( chi_secondorder(z_mid, Om0, Ode0) - chi(z_mid, Om0,
    if delta_mid > target:
        z_high = z_mid
    else:
        z_low = z_mid

z_mid
```

Out[63]: 2.0541709661483765

For the first oder approximation:

This will help us see at what redshift we begin to probe the cosmological density parameters.

In [59]:
```
# Initialize the search interval for z
z_low, z_high = 0.0, 2

tolerance = 1e-6 # Precision of z
target = 0.1 # Looking for delta to be within 10%

# Perform the bisection search
while z_high - z_low > tolerance:
    z_mid = (z_low + z_high) / 2
    delta_mid = np.abs(( z_mid - chi(z_mid, Om0, Ode0) ) / chi(z_mid, Om0, O
    if delta_mid > target:
        z_high = z_mid
    else:
        z_low = z_mid

z_mid
```

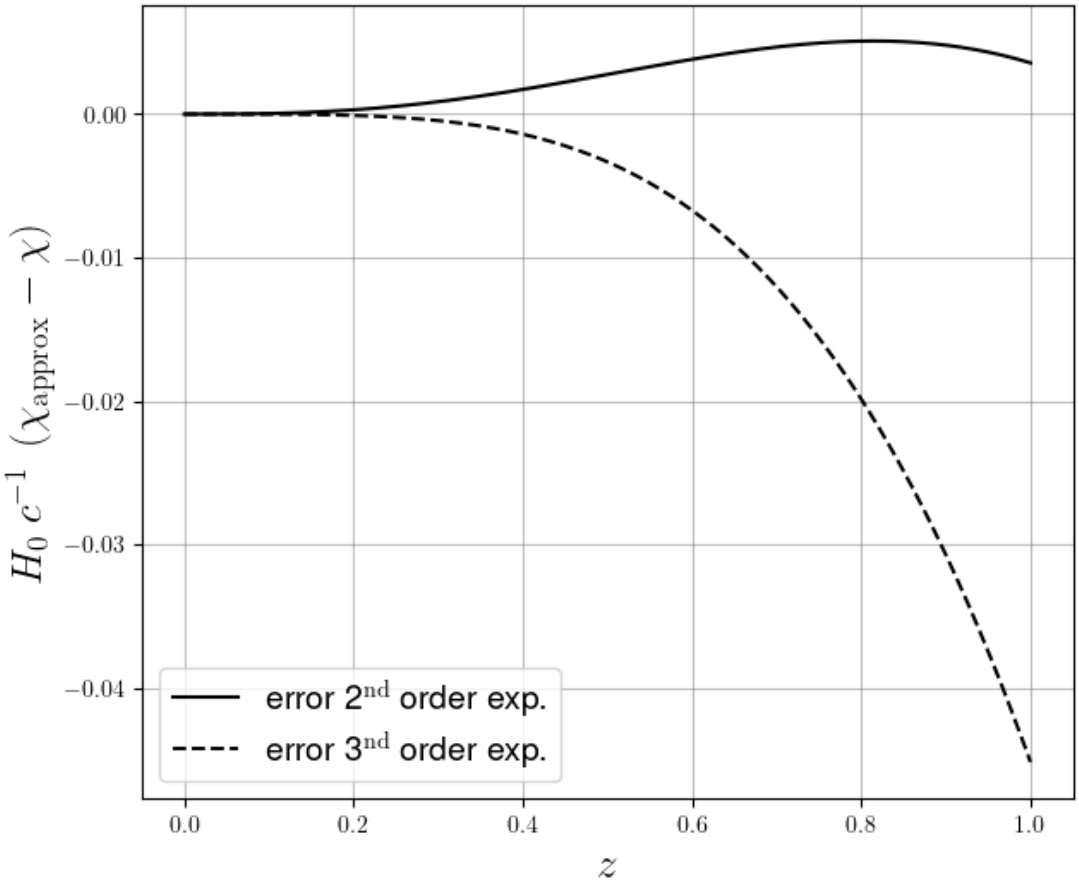
Out[59]: 0.3857755661010742

Surprisingly, we find that the third order approximation diverges at a smaller redshift than the second order approximation. Let's plot each term of the expansion relative to the exact value

In [61]:
```
z = np.linspace(0,1,1000)
exact = chi(z, Om0, Ode0)
third = chi_thirdorder(z, Om0, Ode0)
second = chi_secondorder(z, Om0, Ode0)
first = z

error2 = second - exact
error3 = third - exact
```

In [64]:
```
fig, ax = plt.subplots(1, 1, figsize=(7,6))

# ax.plot(z, exact, ls='-', color='black', label='exact')
# ax.plot(z, first, ls='--', color='black', label='one term exp.')
# ax.plot(z, second, ls=':', color='black', label='two term exp.')
```

```
# ax.plot(z, third, ls='-.', color='black', label='three term exp.')

ax.plot(z, error2, ls='-', color='black', label='error 2$^\mathrm{nd}$ order
ax.plot(z, error3, ls='--', color='black', label='error 3$^\mathrm{nd}$ orde

ax.set_axisbelow(True)
ax.xaxis.grid(color='gray', alpha=0.5, linestyle='-')
ax.yaxis.grid(color='gray', alpha=0.5, linestyle='-')

ax.set_xlabel('$z$', fontsize=18)
ax.set_ylabel(r'$ H_0\ c^{-1}\ \left(\chi_\mathrm{approx} - \chi\right)$', f

ax.legend(fontsize=14)

fig.savefig('/Users/yaronetokayer/Yale Drive/Classes/PHYS 600/phys600 hw/phy
            dpi=300, bbox_inches='tight')
```



In [ ]:

B Mathematica code

PHYS 600: HW 3

Problem 3

Expand the integrand for χ to second order:

```
In[•]:= dχ[z_] := c / (h0 √(Ωm (1 + z)^3 + Ωlam + (1 - Ωm - Ωlam) (1 + z)^2))

In[•]:= dχSeries = Series[dχ[z], {z, 0, 2}]
```

Out[•]=

$$\frac{c}{h0} + \frac{c\,(-2 + 2\,\Omega lam - \Omega m)\,z}{2\,h0} + \frac{c\left(8 - 20\,\Omega lam + 12\,\Omega lam^2 + 4\,\Omega m - 12\,\Omega lam\,\Omega m + 3\,\Omega m^2\right)z^2}{8\,h0} + O[z]^3$$