Iver Warburton
ASTR 600 - Cosmology
October 2023

PHYS 600 : HW 3

1. Reviewing the Background

The following are meant to be short problems, just to review key parts of the homogeneous expansion of the Universe. If not specified, assume a cosmology with $\Omega_{m,0} = 0.3$, $\Omega_\Lambda = 0.7$ and $h = 0.7$.

Density Parameters

Calculate Ω_m and Ω_Λ at $z = 0.5$.

Luminosity and Angular Diameter distances

Make a plot of the luminosity and angular diameter distances as a function of redshift to $z = 10$. Compare these (on the same plot) to the luminosity/angular diameter distance for an $\Omega_{m,0} = 1$ universe.

Make sure to label your axes. Comment on the non-monotonicity of the angular diameter distance. ### Looking Back

At the colloquium last week, the speaker mentioned that a redshift of 2-3 roughly corresponds to 10 billion years ago. Find the redshift that is 10 billion years ago (three significant figures only).

A Λ-dominated Universe

Consider a Universe with only a cosmological constant. Find $a(t)$. What is the age of such a Universe?

2. Massive Neutrinos

(This is based on a problem from Baumann.)

Our current understanding of neutrino physics suggests that atleast two of the three neutrino species have a non-zero (although small) mass. Let us work out the cosmological implications of this.

- Assuming that the neutrinos are relativistic at decoupling, show that the energy density at decoupling is given by

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

  where T_ν is the neutrino temperature.

- Consider a series expansion for small m_ν/T_ν and show that

$$\rho_\nu \approx \rho_{\nu,0} \left(1 + \frac{5}{7\pi^2} \frac{m_\nu^2}{T_\nu^2}\right)$$

  where $\rho_{\nu,0}$ is the energy density of massless neutrinos.

- If ρ_v is "significantly" larger than $\rho_{v,0}$ at the epoch of the CMB ($z \sim 1000$), then the neutrinos can affect the CMB. Estimate the smallest neutrino mass detectable in the CMB. Feel free to make assumptions to do this, but explicitly describe these. You may assume that the CMB temperature today is 0.235 meV.

- Assume that one of the neutrino species has a mass of m_ν. Estimate the redshift at which the neutrinos go non-relativistic.

- Estimate the number density of neutrinos today. To do so, you can't just use the non-relativistic integrals from class, since the neutrinos are not in thermal equilibrium anymore. Instead, calculate the number density at neutrino decoupling, and then dilute the density with the expansion of the Universe.

- Show that these neutrinos have a energy density today of

$$\Omega_{\nu,0} h^2 \approx \frac{m_\nu}{94\,\text{eV}}$$

3. Measuring the Expansion History with Standard Candles

You are encouraged to use Mathematica or something similar to simplify/assist with the algebra in this problem, if you want. If/when you need to, assume the standard values of the cosmological parameters from the first problem.

Suppose that the standard candle method directly measures the radial comoving distance

$$\chi = \int dz \frac{c}{H_0 \left[\Omega_{m,0}(1+z)^3 + \Omega_{\Lambda,0} + (1 - \Omega_{m,0} - \Omega_{\Lambda,0})\big(1+z)^2\right]^{1/2}}$$

- Develop a series expansion of χ as a function of z expanding about $z = 0$. Work to order z^3 (i.e. include those terms).
- To what redshift is this expansion accurate to 10%?
- Explain the following statement - for very low redshift measurements, the only parameter that can be measured is H_0.
- As you increase the redshift reach (still low redshifts), you gain sensitivity to the other cosmological parameters. However, it turns out that you are largely sensitive to a combination of $\Omega_{m,0}$ and $\Omega_{\Lambda,0}$, and not both individually. What combination and why?
- Consider a survey that measures χ to 1% at $z = 0.01, 0.1, 0.2, 0.3$. Forecast the errors on H_0, $\Omega_{m,0}$ and $\Omega_{\Lambda,0}$.

1. Reviewing the Background

Assume $\Omega_{m,0} = 0.3$, $\Omega_\Lambda = 0.7$, $h = 0.7$

i.) Density Parameters

Find Ω_m and Ω_Λ at $z = 0.5$

$$\Omega_m = \frac{\rho_m}{\rho_{crit}} \longleftarrow \text{Definition of } \Omega_m$$

$$\Omega_m = \frac{\rho_{m,0}(1+z)^3}{\rho_{crit,0}} \cdot \frac{\rho_{crit,0}}{\rho_{crit}}$$

$$\Omega_m = \frac{\Omega_{m,0}(1+z)^3}{\Omega_{m,0}(1+z)^3 + \Omega_\Lambda}$$

← $\Omega_\Lambda = \Omega_{\Lambda,0} = 0.7$ to be used here. Ω_Λ has no a-relation in Friedmann. $H_0^2(\Omega_{m,0}a^{-3} + \Omega_{\Lambda,0})$

$$\Omega_m = \frac{0.3(1+0.5)^3}{0.3(1+0.5)^3 + 0.7}$$

$$\boxed{\Omega_m = 0.59124} \text{ at } z = 0.5$$

$$\Omega_\Lambda = \frac{\Omega_{\Lambda_0}}{\Omega_{\Lambda_0} + \Omega_m(1+z)^3}$$

← Actually sort of an inverse of my expression for Ω_m.

$$\Omega_\Lambda = \frac{0.7}{0.7 + 0.3(1+0.5)^3}$$

$$\boxed{\Omega_\Lambda = 0.40876} \text{ at } z = 0.5$$

Check that Ω_m and Ω_Λ add to 1:

$$\Omega_m + \Omega_\Lambda \stackrel{?}{=} 1$$

$$0.59124 + 0.40876 = 1$$

$$1 = 1 \checkmark$$

$$\boxed{\begin{aligned} &\text{At } z = 0.5, \\ &\Omega_m = 0.59124 \\ &\Omega_\Lambda = 0.40876 \end{aligned}}$$

ii.) Luminosity and Angular Diameter Distances

Mostly done in attached Jupyter Notebook. Plot is also there.

$$d_A = \frac{d_m}{1+z} \longleftarrow \text{angular diameter distance}$$

$$d_L = d_m(1+z) \longleftarrow \text{luminosity distance}$$

$$d_L = d_A(1+z)^2$$

$$d_m = \frac{c}{H_0}\int \frac{1}{(1+z)E(z)}\,dz \longleftarrow \text{comoving distance}$$

$$E(z) = \sqrt{\Omega_m(1+z)^3 + \Omega_\Lambda}$$

$$H^2 = H_0^2[\qquad]$$

$$H = H_0\sqrt{\qquad}$$

See Jupyter Notebook.

$$\boxed{\begin{aligned} d_m &= \frac{c}{H_0}\int \frac{1}{(1+z)\sqrt{\Omega_m(1+z)^3 + \Omega_\Lambda}}\,dz \\ d_L &= \frac{c(1+z)}{H_0}\int \frac{1}{(1+z)\sqrt{\Omega_m(1+z)^3 + \Omega_\Lambda}}\,dz \\ d_A &= \frac{c}{H_0(1+z)}\int \frac{1}{(1+z)\sqrt{\Omega_m(1+z)^3 + \Omega_\Lambda}}\,dz \end{aligned}}$$

← Used these in my code.

iii.) Looking Back

10 billion years ago = what redshift?

According to Ryden 6.2, (For flat + $\Omega_{\Lambda,0} > 0$)

Integrating the Friedmann equation gives:

$$H_0 t = \frac{2}{3\sqrt{1-\Omega_{m0}}} \ln\left[\left(\frac{a}{a_{m\Lambda}}\right)^{3/2} + \sqrt{1 + \left(\frac{a}{a_{m\Lambda}}\right)^3}\right] \quad (6.28)$$

where $a = \frac{1}{1+z}$ and $a_{m\Lambda} = \left(\frac{\Omega_{m,0}}{\Omega_{\Lambda,0}}\right)^{\frac{1}{3}} = \left(\frac{0.3}{0.7}\right)^{\frac{1}{3}} = 0.7539$

$$t = \frac{2}{3H_0\sqrt{1-\Omega_{m,0}}} \ln\left[\left(\frac{1}{1+z}\cdot\left(\frac{\Omega_{\Lambda,0}}{\Omega_{m,0}}\right)^{\frac{1}{3}}\right)^{\frac{3}{2}} + \sqrt{1 + \left(\frac{1}{1+z}\cdot\left(\frac{\Omega_{\Lambda,0}}{\Omega_{m,0}}\right)^{\frac{1}{3}}\right)^3}\right]$$

I was doing it on my own and changed my mind, so...

$$t = \frac{2}{3\cdot 70\cdot\sqrt{1-.3}} \ln\left[\left(\frac{1.32635}{1+z}\right)^{\frac{3}{2}} + \sqrt{1 + \left(\frac{1.32635}{1+z}\right)^3}\right]$$

$$10\times10^9 = 0.011383 \ln\left[\sqrt{\left(\frac{1.32635}{1+z}\right)^3} + \sqrt{1 + \left(\frac{1.32635}{1+z}\right)^3}\right]$$

$$8.78493\times10^{11} = \ln\left[\sqrt{\left(\frac{1.32635}{1+z}\right)^3} + \sqrt{1 + \left(\frac{1.32635}{1+z}\right)^3}\right]$$

$$e^{8.78493\times10^{11}} = \sqrt{\left(\frac{1.32635}{1+z}\right)^3} + \sqrt{1 + \left(\frac{1.32635}{1+z}\right)^3}$$

$$= \sqrt{2.33333\,a^3} + \sqrt{1 + 2.33333\,a^3}$$

Wolfram Alpha (using $x = \frac{a}{a_{m\Lambda}}$, $b = e^{\frac{3}{2}H_0 t\sqrt{1-\Omega_{m0}}}$)

Result:

$$x = \frac{(b^2-1)^{2/3}}{2^{2/3}\,b^{2/3}}$$

Substituting:

$$\frac{a}{a_{m\Lambda}} = \frac{\left(e^{3H_0 t\sqrt{1-\Omega_{\Lambda}}} - 1\right)^{2/3}}{2^{2/3}\,e^{H_0 t\sqrt{1-\Omega_{\Lambda 0}}}}$$

Unit conversions detailed below.*

$$\frac{1}{1+z}\cdot\left(\frac{\Omega_{\Lambda,0}}{\Omega_{m,0}}\right)^{\frac{1}{3}} = \frac{\left(e^{(3\cdot 70\cdot 10\times10^9\sqrt{1-0.7})} - 1\right)^{2/3}}{2^{2/3}\,e^{70\cdot 10\times10^9\sqrt{1-0.7}}}$$

$$\frac{1}{1+z}\cdot\left(\frac{0.7}{0.3}\right)^{\frac{1}{3}} = \frac{\left(e^{(3\cdot 0.390313)} - 1\right)^{2/3}}{2^{2/3}\,e^{0.390313}}$$

$$\frac{1}{1+z}(1.32635) = 0.726705$$

Ned Wright calculator: $z_{10\,Gyr} \approx 1.856 \checkmark$

$$\boxed{z = 1.825} \longrightarrow$$

It seems that 10 Gyr ago was slightly less than z = 2-3.

*Unit conversion work in exponents:

$$\frac{70\text{ km}}{\text{s Mpc}}\left|\frac{10\times10^9\text{ yr}}{}\right|\frac{\sqrt{0.3}}{}\left|\frac{\pi\times10^7\text{ s}}{1\text{ yr}}\right|\frac{1\text{ Mpc}}{3.086\times10^{19}\text{ km}} = 0.390313$$

iv.) Λ-Dominated Universe

Universe with only Λ constant. Find a(t) and age.

$$\frac{\dot a}{a} = H_0\sqrt{\Omega_{\Lambda,0}} \longleftarrow \Omega_{\Lambda,0} = 1 \text{ because } \Sigma\Omega = 1$$

$$\frac{\dot a}{a} = H_0$$

$$\frac{1}{a}\frac{da}{dt} = H_0$$

$$\int_0^1 \frac{1}{a}\,da = H_0\int_t^0 dt$$

$$\ln(1) - \ln(0) = H_0(0 - t)$$

$$0 - \boxed{\text{UND}} = -H_0 t$$

$$\boxed{H_0 t = \boxed{\text{UND}}}$$

A value for a(t) cannot exist for this Universe. The age would need to be infinitely old. Wild question, cool!

2. Massive Neutrinos

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

$$\rho_\nu = \frac{T_\nu^4}{\pi^2}\int \frac{\frac{E^2}{T^2}\sqrt{\frac{E^2}{T^2} + m_\nu^2/T_\nu^2}}{e^{\cancel{E/T}}+1}\,d\xi$$

$$\rho_\nu = \frac{T_\nu^4}{\pi^2}\int \frac{\frac{p^2+m^2}{T^2}\sqrt{\frac{p^2+m^2}{T^2} + \frac{m_\nu^2}{T_\nu^2}}}{e^{\frac{\sqrt{p^2+m^2}}{T}}+1}\,d\mathcal{E}$$

$$\xi = \frac{P}{T} = \frac{\sqrt{p^2+m^2}}{T}$$
$$dP = T\,d\xi$$

a.) Energy density at decoupling

$$\rho(T) = \frac{g}{(2\pi)^3}\int \frac{1}{e^{\frac{E-\mu}{T}}+1}E(p)\,d^3p,$$ where μ is chemical potential and is relevant only when numbers of particles in system is changing $\to \mu = 0$

$$= \frac{g}{(2\pi)^3}\iiint \frac{\sqrt{p^2+m^2}}{e^{\frac{\sqrt{p^2+m^2}}{T}}+1}\,d^3p,$$ where we treat the m in the exponent in the denominator to be negligibly small $(m \ll p)$

$$= \frac{g}{8\pi^3}\iiint \frac{\sqrt{p^2+m^2}}{e^{\frac{p}{T}}+1}\,d^3p$$

$$= 4\pi p^2 \cdot \frac{g}{8\pi^3}\int \frac{\sqrt{p^2+m^2}}{e^{\frac{p}{T}}+1}\,dp$$

2 dof

$$= \frac{gp^2}{2\pi^2}\int \frac{\sqrt{p^2+m^2}}{e^{\frac{p}{T}}+1}\,dp,$$ where $g = 2$ degrees of freedom

$$= \frac{2p^2}{2\pi^2}\int \frac{\sqrt{\xi^2T^2+m^2}}{e^{\xi}+1}\,T\,d\xi$$

$$= \frac{p^2}{\pi^2}\int T\frac{T\sqrt{\xi^2+\frac{m^2}{T^2}}}{e^{\xi}+1}\,d\xi$$

$$= \frac{\cancel{T^2}}{\pi^2}\int \frac{\xi^2T^2\sqrt{\xi^2+\frac{m^2}{T^2}}}{e^{\xi}+1}\,d\xi$$

$$= \frac{1}{\pi^2}\int \frac{\xi^2T^2T^2\sqrt{\xi^2+\frac{m^2}{T^2}}}{e^{\xi}+1}\,d\xi$$

$$\boxed{\rho(T) = \frac{T_\nu^4}{\pi^2}\int \frac{\xi^2\sqrt{\xi^2+\frac{m^2}{T_\nu^2}}}{e^{\xi}+1}\,d\xi}$$ ← Energy density at decoupling, as expected.

b.) Show energy density of neutrinos

$$\rho(T) = \frac{T_\nu^4}{\pi^2}\int \frac{\xi^2\xi\sqrt{1+\frac{m_\nu^2}{T_\nu^2\xi^2}}}{e^{\xi}+1}\,d\xi$$

where $\frac{m_\nu^2}{T_\nu^2\xi^2} \ll 1$ so we can say: $\sqrt{1+\frac{m_\nu^2}{T_\nu^2\xi^2}} \approx \sqrt{1+\text{small}}$

So we use: $(1+x)^n = 1+nx$

$1 + \frac{1}{2}\frac{m^2}{T^2\xi^2}$

$$= \frac{T_\nu^4}{\pi^2}\int \frac{\xi^3\left(1+\frac{m_\nu^2}{2T_\nu^2\xi^2}\right)}{e^{\xi}+1}\,d\xi$$

$$= \frac{T_\nu^4}{\pi^2}\int \frac{\xi^3 + \frac{m_\nu^2\xi}{2T_\nu^2}}{e^{\xi}+1}\,d\xi$$

$$= \frac{T_\nu^4}{\pi^2}\left(\int_0^\infty \frac{\xi^3}{e^{\xi}+1}\,d\xi + \int_0^\infty \frac{m_\nu^2\xi}{2T_\nu^2(e^{\xi}+1)}\,d\xi\right),$$ where we integrate from 0 to ∞.

$$= \frac{T_\nu^4}{\pi^2}\left(\frac{7\pi^4}{120} + \frac{\pi^2m_\nu^2}{24T_\nu^2}\right)$$

$$= \frac{120T_\nu^4}{7\pi^2}\left(1 + \frac{120\,m_\nu^2}{7\cdot 24\pi^2T_\nu^2}\right)$$

$$= \frac{120T_\nu^4}{7\pi^2}\left(1 + \frac{5m_\nu^2}{7\pi^2T_\nu^2}\right)$$

$$= \frac{7\pi^2T_\nu^4}{120}\left(1 + \frac{5m_\nu^2}{7\pi^2T_\nu^2}\right),$$ where we can roughly call the coefficient as ρ_0 (source: Navya)

$$\boxed{\rho_\nu(T) \approx \rho_{\nu,0}\left(1 + \frac{5m_\nu^2}{7\pi^2T_\nu^2}\right)}$$ ← This is the energy density of neutrinos.

c.) Smallest detectable neutrino mass

$\rho_\nu \gg \rho_{\nu,0}$, so let's say $\rho_\nu = 10\rho_{\nu,0}$ (on the low end)

From previously-determined equation:

From textbook (for present):
CMB$_0$ → 2.73K, 0.235 meV
Neut$_0$ → 1.97K, 0.17 meV

$$10\rho_{\nu,0} = \rho_{\nu,0}\left(1 + \frac{5m^2}{7\pi^2T_\nu^2}\right)$$

$$10 = 1 + \frac{5m^2}{7\pi^2T_\nu^2}$$

$$9 = \frac{5}{7}\frac{m^2}{\pi^2T_\nu^2}$$

$$m = \sqrt{\frac{7\cdot 9}{5}\pi^2T_\nu^2}$$

$$m_\nu = \sqrt{\frac{63}{5}}\pi T_\nu$$

$$m_\nu = \sqrt{\frac{63}{5}}\pi\left(\frac{4}{11}\right)^{\frac{1}{3}}T_{CMB,0}(1+z)$$ (0.235 meV, $z = 1000$)

$$m_\nu = \sqrt{\frac{63}{5}}\pi\left(\frac{4}{11}\right)^{\frac{1}{3}}235\text{ meV}$$

From the scaling between T_ν and T_γ in class:

$T_\nu = \left(\frac{4}{11}\right)^{\frac{1}{3}}T_{CMB}$

$T_{CMB_{1000}} = T_{CMB_0}(1+z)$

$T_{CMB_{1000}} = 0.235\text{ meV}(1+1000)$

$T_{CMB_{1000}} = 235\text{ meV}$

$T_{\nu_{1000}} = \left(\frac{4}{11}\right)^{\frac{1}{3}}235\text{ meV}$

$$\boxed{\begin{aligned} m_\nu &= 836.5\text{ meV} \\ m_\nu &= 0.837\text{ eV}\end{aligned}}$$ ← Minimum detectable neutrino mass at $z \sim 1000$.

d.) Redshift when neutrinos go non-relativistic

Neutrinos go non-relativistic when $E_{kin} = E_{mass}$

$k_BT_\nu = m_\nu c^2$

$T_\nu \propto m_\nu$

$m_\nu \propto T_\nu \propto \frac{1}{a} \propto 1+z$

$m_\nu = \frac{k_B}{c^2}T_\nu$

$k_B = 1.38\times10^{-23}\frac{m^2\,kg}{s^2\,K}$

$m_\nu = \frac{k_B}{c^2}T_{\nu_0}(1+z)$

$1\text{ meV} = \frac{1.602\times10^{-19}\text{ J}}{}\Big|\frac{kg\,m^2}{J\,s^2}$

$1+z = \frac{c^2}{k_BT_{\nu,0}}m_\nu \quad \frac{m^2}{s^2\,\text{meV}}\,\text{meV}$

$$= \frac{(3\times10^8)^2m^2}{s^2}\Big|\frac{s^2\,K}{1.38\times10^{-23}m^2\,kg}\Big|\frac{}{1.97\,K}\Big|\frac{1.602\times10^{-19}J}{1\text{ meV}}\Big|\frac{kg\,m}{J\,s^2}$$

— x —

My inclination was to do calculations like this but Navya suggested using $T = m$ rather than $T \propto m$, so I do that here.

$m_\nu = T_\nu$

$m_\nu = T_{\nu,0}(1+z)$

$m_\nu = 0.17\text{ meV}(1+z)$

$z+1 = \frac{m_\nu}{0.17\text{ meV}}$ ← from textbook value cited in 2c solution

$$\boxed{z = 5.88\,m_\nu - 1\;\frac{1}{\text{meV}}}$$ ← Leaving it in terms of m_ν because "m_ν" was the given mass. Though Baumann does state that $z_{\nu,NR}$ is when $m_\nu > 0.24$ meV. → ($z = 1.41$ with this)

I don't know how I feel about this answer. It was checked by Navya but I feel like the constants should be there for the calculation. I found a paper that says $z+1 = \frac{m_\nu}{0.528\text{ meV}}$, which doesn't match.

e) Number density of neutrinos today

Energy density at decoupling $\propto gT^3$.

T falls as $\frac{1}{a}$

Density falls as $\frac{1}{a^3}$

- Find relation between number density of photons (known) and number density of neutrinos (unknown).
- We found scaling between T_ν and n_ν today:

$n_\nu \propto \frac{3}{4}gT_\nu^3$

$n_\gamma \propto gT_\gamma^3$

$\frac{n_\nu}{n_\gamma} = \frac{\frac{3}{4}T_\nu^3}{1T_\gamma} \leftarrow T_\nu = \left(\frac{4}{11}\right)^{\frac{1}{3}}T_\gamma$

$n_{\nu,0} = \frac{3}{4}\left(\frac{4}{11}\right)T_\gamma^2\,n_{\gamma,0}$ ← in textbook: $n_{\gamma,0} = 4.107\times10^8\,\frac{1}{m^3}$ today

$n_{\nu,0} = \frac{3}{11}\cdot 4.107\times10^8\,\frac{1}{m^3}$

$$\boxed{n_{\nu,0} = 1.120\times10^8\,\frac{1}{m^3}}$$ ← A solution for $n_{\nu,0}$, number density of neutrinos today.

f.) Energy density of neutrinos today

Ωh^2 vs energy density.

Start with energy density of photons. Units: $\left[\frac{h^2\,eV}{m^3}\right]$

↳ In textbook: $1.053\times10^{10}\,\frac{h^2\,eV}{m^2}$

$\rho = \Omega_{\nu,0}h^2 = m_\nu n_{\nu,0}$ ← Definition of density ρ

$= m_\nu\frac{3}{11}n_{\gamma,0}$

$1.053\times10^{10}\,\frac{h^2\,eV}{\cancel{m^3}} = m_\nu\cdot 1.120\times10^8\,\cancel{\frac{1}{m^3}}$ ← Using $n_{\nu,0}$ found above in 2e.

$$\boxed{\Omega_{\nu,0}h^2 \approx \frac{m_\nu}{94\,eV}}$$ ← Dividing these two quantities returns roughly 94 eV in the denominator.

The energy density of neutrinos today, in terms of neutrino mass. This matches the expected expression.

3.) Measuring Expansion History

$$\chi = \int dz \frac{c}{H_0[\Omega_{m,0}(1+z)^3 + \Omega_{\Lambda,0} + (1-\Omega_{m,0}-\Omega_{\Lambda,0})(1+z)^2]^{1/2}}$$

a.) Series expansion of $\chi(z)$ about $z=0$:

$$\chi(0) + z\chi'(0) + \frac{1}{2}z^2\chi''(0)$$

Used Mathematica to expand. Result:

$$\frac{c}{H_0} + \frac{1}{2}z^2\left(\frac{3c(3\Omega_m + 2(1-\Omega_m-\Omega_\Lambda))^2}{4H_0} - \frac{c(6\Omega_m + 2(1-\Omega_m-\Omega_\Lambda))}{2H_0}\right) - \frac{cz(3\Omega_m + 2(1-\Omega_m-\Omega_\Lambda))}{2H_0}$$

b.) Used Mathematica to integrate analytically. Result:

$$\frac{cz}{H_0} - \frac{3cz^2\Omega_m}{4H_0} + \frac{cz^3(3\Omega_m + 2(1-\Omega_m-\Omega_\Lambda))^2}{8H_0} - \frac{cz^3(6\Omega_m + 2(1-\Omega_m-\Omega_\Lambda))}{12H_0} - \frac{cz^2(1-\Omega_m-\Omega_\Lambda)}{2H_0}$$

Used Python to integrate numerically. Result:

[Work is in following pages of code]

$\downarrow$

But my solution was $\boxed{z=1.333}$, as detailed in the Jupyter Notebook and plot.

c.) For very low-z measurements, we can really only measure H_0 because in the expansion:

$$\frac{c}{H_0} + \frac{1}{2}z^2\left(\frac{3c(3\Omega_m + 2(1-\Omega_m-\Omega_\Lambda))^2}{4H_0} - \frac{c(6\Omega_m + 2(1-\Omega_m-\Omega_\Lambda))}{2H_0}\right) - \frac{cz(3\Omega_m + 2(1-\Omega_m-\Omega_\Lambda))}{2H_0}$$

every term has at least one power of z except for the very first term, $\frac{c}{H_0}$.

Similarly in the result of analytical integration:

$$\frac{cz}{H_0} - \frac{3cz^2\Omega_m}{4H_0} + \frac{cz^3(3\Omega_m + 2(1-\Omega_m-\Omega_\Lambda))^2}{8H_0} - \frac{cz^3(6\Omega_m + 2(1-\Omega_m-\Omega_\Lambda))}{12H_0} - \frac{cz^2(1-\Omega_m-\Omega_\Lambda)}{2H_0}$$

every term has multiple zs multiplied together except for the first term.

We are told that z is very small and when very small things are multiplied together, the result is extremely small. Because of that, all of the terms after the first will basically go to zero.

In both of these cases, the only other measurable value in the first term is H_0, so that is the measurable term.

d.) How do $\Omega_{m,0}$ and $\Omega_{\Lambda,0}$ impact d_L when redshifts are not tiny?

$$d_L = \frac{cz}{H_0}\left(1 + z\frac{1-q_0}{2}\right), \text{ where } q_0 = \Omega_{r,0} + \frac{1}{2}\Omega_{m,0} - \Omega_{\Lambda,0}$$ $\leftarrow$ this comes from Ryden (7.45)

$$d_L = \frac{cz}{H_0}\left(1 + \frac{z}{2}\left(1 - \left(\Omega_{r,0} + \frac{1}{2}\Omega_{m,0} - \Omega_{\Lambda,0}\right)\right)\right)$$

$$= \frac{cz}{H_0}\left(1 + \frac{z}{2}\left(1 - \cancel{\Omega_{r,0}} - \frac{1}{2}\Omega_{m,0} + \Omega_{\Lambda,0}\right)\right)$$

$$d_L = \frac{cz}{H_0}\left(1 + \frac{z}{2}\left(1 - \underbrace{\frac{1}{2}\Omega_{m,0} + \Omega_{\Lambda,0}}\right)\right)$$

$\rightarrow$ d_L is sensitive to other parameters, but when it comes to $\Omega_{m,0}$ and $\Omega_{\Lambda,0}$, those impact d_L in the combination of $\Omega_{\Lambda,0} - \frac{1}{2}\Omega_{m,0}$.

ASTR 600 - Cosmology

HW 3

Iver Warburton

October 2023

In [1]:
```
# Our lovely imports

import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import quad
```

Problem 1

part ii.)

In [2]:
```
# Constants

c = 2.99 * 10**5 # km/s
h = 0.7
H_0 = h*100
#Omegam0 = 0.3
#OmegaL0 = 0.7
z_max = 10
z = np.linspace(0, z_max, 100)
```

```
# Calculate comoving distance

# Define function to build integrand
def integrand_comdist(z, Omegam0, OmegaL0):
    # Integrand expression for comoving distance that was provided in
problem set
    #X = c/(H_0*(Omega_m0*(1+z)**3 + Omega_L0 + (1 - Omega_m0 -
Omega_L0)*(1+z)**2)**(1/2))
    X_com = (c/H_0) * (1/((1+z) * (Omegam0*(1+z)**3 + OmegaL0)**
(1/2)))
    return X_com
#integrand_comdist


# Function to calculate comoving distance
def calc_comdist(z_vals, Omegam0, OmegaL0):
    comdist_vals = []
    comdist = [quad(integrand_comdist(z_vals, Omegam0, OmegaL0), 0,
z_val)[0] for z_val in z]
    comdist_vals.append(comdist)
    return comdist
```

```
# Function to calculate comoving distance
def calc_comdist(z, Omegam0, OmegaL0):
    #X_com_integrand = (c/H_0) * (1/((1+z) * (Omegam0*(1+z)**3 +
OmegaL0)**(1/2)))
    comdist_vals = []
    for z_val in z:
        comdist = quad(X_com_integrand, 0, z_val, args=(Omegam0,
OmegaL0))[0]# for z_val in z]
        comdist_vals.append(comdist)
    return comdist_vals
```

In [3]:
```
# Calculate comoving distance

# Define function to build integrand
def integrand_comdist(z, Omegam0, OmegaL0):
    # Integrand expression for comoving distance that was provided in
    #X = c/(H_0*(Omega_m0*(1+z)**3 + Omega_L0 + (1 - Omega_m0 - Omega_
    X_com = (c/H_0) * (1/((1+z) * (Omegam0*(1+z)**3 + OmegaL0)**(1/2))
    return X_com
#integrand_comdist


# Function to calculate comoving distance
def calc_comdist(z_vals, Omegam0, OmegaL0):
    #comdist_vals = []
    comdist = np.array([quad(integrand_comdist, 0, z_val, args=(Omegam
    #comdist_vals.append(comdist)
    return comdist
```

```
# Function to calculate luminosity distance
def calc_lumdist(z_vals, Omegam0, OmegaL0):  #, comdist):  #z_vals,
Omegam0, OmegaL0):
    comdist = calc_comdist(z_vals, Omegam0, OmegaL0)
    lumdist_vals = []
    for z in z_vals:
        lumdist = comdist * (1+z)
        lumdist_vals.append(lumdist)
    return lumdist
```

In [4]:
```
# Function to calculate luminosity distance
def calc_lumdist(z_vals, Omegam0, OmegaL0):  #, comdist):  #z_vals, Om
    comdist = calc_comdist(z_vals, Omegam0, OmegaL0)
    #lumdist_vals = []
    #for z in z_vals:
    lumdist = comdist * (1+z_vals)
    #lumdist_vals.append(lumdist)
    return lumdist
```

```
# Function to calculate angular distance
def calc_angdist(z_vals, Omegam0, OmegaL0):  #comdist):  #z_vals,
Omegam0, OmegaL0):
    comdist = calc_comdist(z_vals, Omegam0, OmegaL0)
    angdist_vals = []
    for z in z_vals:
        angdist = comdist / (1+z)
        angdist_vals.append(angdist)
    return angdist_vals
```

In [5]:
```
# Function to calculate angular distance
def calc_angdist(z_vals, Omegam0, OmegaL0):  #comdist):  #z_vals, Omeg
    comdist = calc_comdist(z_vals, Omegam0, OmegaL0)
    #angdist_vals = []
    #for z in z_vals:
    angdist = comdist / (1+z_vals)
    #angdist_vals.append(angdist)
    return angdist
```

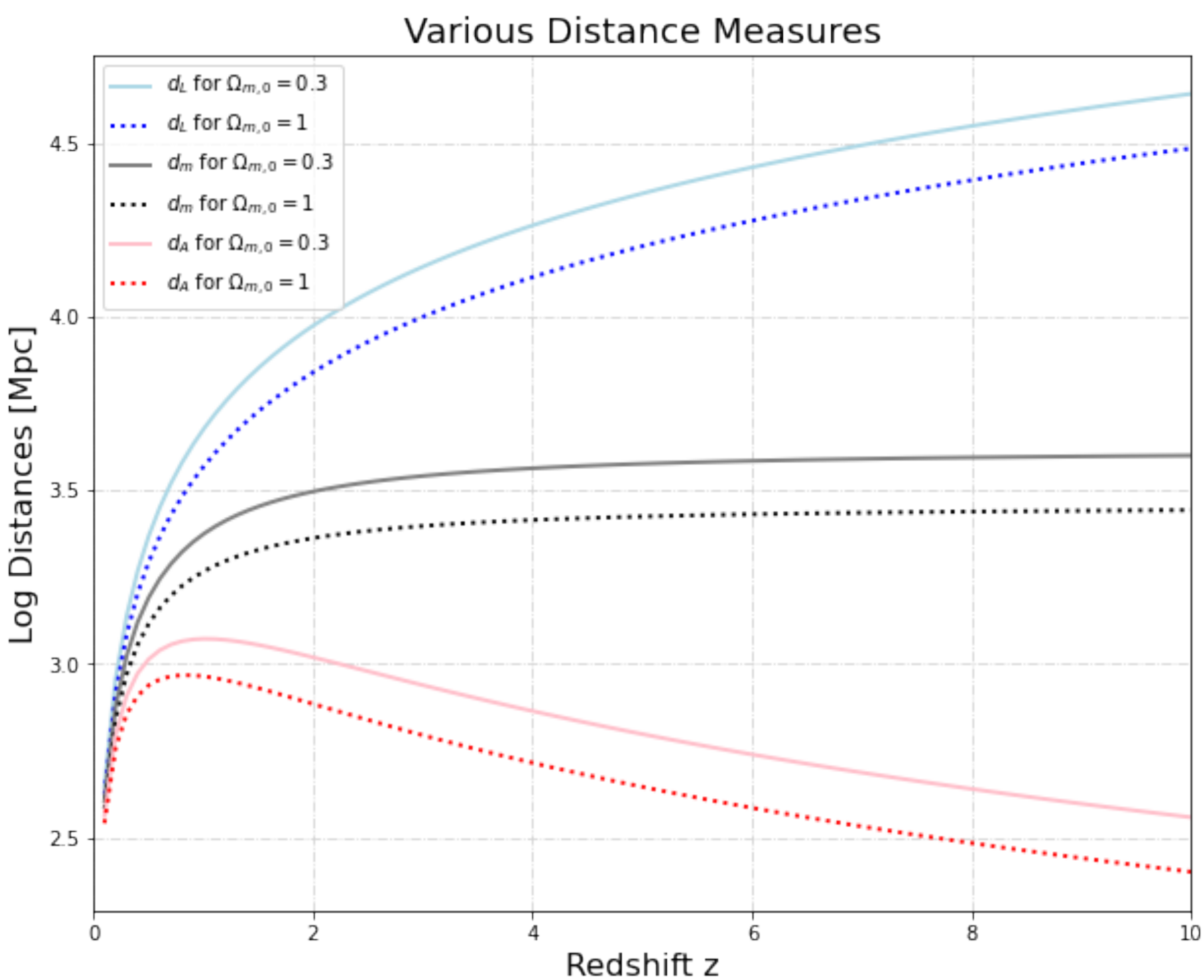
In [6]:
```
# Plot results
def dists_plot(z_vals):

    ## SETUP ##
    # Set figure
    fig, ax = plt.subplots(figsize=(10,8)) #(8,6))

    Omegam0_03 = 0.3
    Omegam0_1 = 1.0
    OmegaL0_07 = 0.7
    OmegaL0_0 = 0

    # Calculate distances
```

```
    lumdist_vals03 = calc_lumdist(z_vals, Omegam0_03, OmegaL0_07)
    lumdist_vals1 = calc_lumdist(z_vals, Omegam0_1, OmegaL0_0)

    comdist_vals03 = calc_comdist(z_vals, Omegam0_03, OmegaL0_07)
    comdist_vals1 = calc_comdist(z_vals, Omegam0_1, OmegaL0_0)

    angdist_vals03 = calc_angdist(z_vals, Omegam0_03, OmegaL0_07)
    angdist_vals1 = calc_angdist(z_vals, Omegam0_1, OmegaL0_0)


    ## PLOTTING ##
    # Plot distances
    plt.plot(z_vals, np.log10(lumdist_vals03), label='$d_L$ for $\Omeg
    plt.plot(z_vals, np.log10(lumdist_vals1), label='$d_L$ for $\Omega

    plt.plot(z_vals, np.log10(comdist_vals03), label='$d_m$ for $\Omeg
    plt.plot(z_vals, np.log10(comdist_vals1), label='$d_m$ for $\Omega

    plt.plot(z_vals, np.log10(angdist_vals03), label='$d_A$ for $\Omeg
    plt.plot(z_vals, np.log10(angdist_vals1), label='$d_A$ for $\Omega


    ## LABELLING ##
    # Label axes
    plt.xlabel('Redshift z', fontsize=16)
    plt.ylabel('Log Distances [Mpc]', fontsize=16)

    # Increse axis numbering text size
    ax.tick_params(axis='both', which='major', labelsize=10)
    ax.tick_params(axis='both', which='minor', labelsize=10)

    # Plot features
    plt.xlim(0, 10)
    plt.legend(fontsize=10)  #14)
    plt.grid(True, color='lightgrey', ls='-.')
    plt.title('Various Distance Measures', fontsize=18)


    ## SAVING ##
    # Save and show
    plt.savefig("HW3Q1DistancesPlot.pdf", format="pdf", bbox_inches="t
    plt.show()
```

In [7]:
```
dists_plot(z)
```

```
/var/folders/1w/ktxtfrr91bj5bztz50dqm0fr0000gn/T/ipykernel_32956/1624
047874.py:26: RuntimeWarning: divide by zero encountered in log10
  plt.plot(z_vals, np.log10(lumdist_vals03), label='$d_L$ for $\Omega
_{m,0} = 0.3$', ls='-', color='lightblue', lw=2)
/var/folders/1w/ktxtfrr91bj5bztz50dqm0fr0000gn/T/ipykernel_32956/1624
047874.py:27: RuntimeWarning: divide by zero encountered in log10
  plt.plot(z_vals, np.log10(lumdist_vals1), label='$d_L$ for $\Omega_
{m,0} = 1$', ls=':', color='blue', lw=2)
```

```
/var/folders/1w/ktxtfrr91bj5bztz50dqm0fr0000gn/T/ipykernel_32956/1624
047874.py:29: RuntimeWarning: divide by zero encountered in log10
  plt.plot(z_vals, np.log10(comdist_vals03), label='$d_m$ for $\Omega
_{m,0} = 0.3$', ls='-', color='grey', lw=2)
/var/folders/1w/ktxtfrr91bj5bztz50dqm0fr0000gn/T/ipykernel_32956/1624
047874.py:30: RuntimeWarning: divide by zero encountered in log10
  plt.plot(z_vals, np.log10(comdist_vals1), label='$d_m$ for $\Omega_
{m,0} = 1$', ls=':', color='black', lw=2)
/var/folders/1w/ktxtfrr91bj5bztz50dqm0fr0000gn/T/ipykernel_32956/1624
047874.py:32: RuntimeWarning: divide by zero encountered in log10
  plt.plot(z_vals, np.log10(angdist_vals03), label='$d_A$ for $\Omega
_{m,0} = 0.3$', ls='-', color='pink', lw=2)
/var/folders/1w/ktxtfrr91bj5bztz50dqm0fr0000gn/T/ipykernel_32956/1624
047874.py:33: RuntimeWarning: divide by zero encountered in log10
  plt.plot(z_vals, np.log10(angdist_vals1), label='$d_A$ for $\Omega_
{m,0} = 1$', ls=':', color='red', lw=2)
/var/folders/1w/ktxtfrr91bj5bztz50dqm0fr0000gn/T/ipykernel_32956/1624
047874.py:54: MatplotlibDeprecationWarning: savefig() got unexpected
keyword argument "overwrite" which is no longer supported as of 3.3 a
nd will become an error in 3.6
  plt.savefig("HW3Q1DistancesPlot.pdf", format="pdf", bbox_inches="ti
ght", overwrite=True)
```



This plot very closely matches the prediction plot I drew before coding this notebook.

- Luminosity distance is simply the comoving distance increased by a factor of $(1 + z)$, so this distance increases a little bit more quickly as z increases.
- Angular diameter distance is certainly the most curious result of these, as it increases with z and then suddenly slopes back down, but it's entirely possible to come up with this concept before plotting. This behaviour has to do with the way that angular sizes behave in an expanding Universe. Earlier in the process of expansion, distant objects were much closer to us and therefore covered a larger angular area on the sky. The images we see of these objects are from light that left the objects when they were much closer and larger on our sky, so this turnover point is closely related to the expansion rate of the Universe.

Problem 3

In [8]:
```
# Constants
c = 2.9979 * 10**8 # m/s
h = 0.7
H_0 = 100 * h # units of km s-1 Mpc-1
z_max = 2
z = np.linspace(0, z_max, 100)
Omega_m0 = 0.3
Omega_L0 = 0.7
```

Analytical integration

Completed in Mathematica and result is rewritten here to be plotted.

In [9]:
```
# Expression for result of analytical integration
X_analint = ((c * z)/(H_0)) - ((3 * c * z**2 * Omega_m0)/(4 * H_0)) +
```

Numerical integration

In [10]:
```
# Define function to build integrand
def integrand(z):
    # Integrand expression for comoving distance that was provided in
    X = c/(H_0*(Omega_m0*(1+z)**3 + Omega_L0 + (1 - Omega_m0 - Omega_L
    return X
```

In [11]:

```python
# Calculate the integral for comoving distance numerically
X_numint = [quad(integrand, 0, z_val)[0] for z_val in z]
#print(np.shape(X_numint))
```

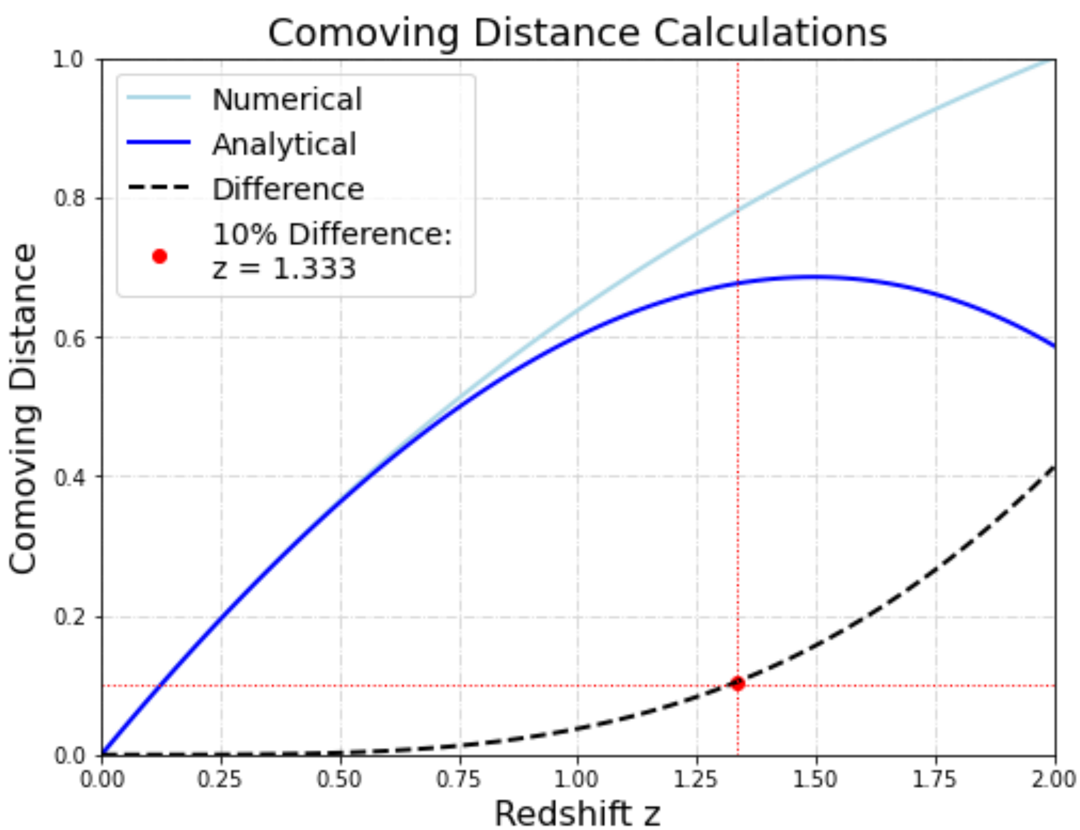
In [12]:

```python
# Plot results
def comdist_plot(z, results_numint, results_analint):

    ## PLOT ##
    # Set figure
    fig, ax = plt.subplots(figsize=(8,6))

    # Plot distances
    #plt.plot(z, results_numint, label='Numerical', ls='-', color='lig
    #plt.plot(z, results_analint, label='Analytical', ls='-', color='b
    # Plot distances normalised to numerical result
    plt.plot(z, results_numint/np.max(results_numint), label='Numerica
    plt.plot(z, results_analint/np.max(results_numint), label='Analyti

    # Plot difference between the calculations
    difference = results_numint/np.max(results_numint) - results_anali
    plt.plot(z, difference, label='Difference', ls='--', color='black'


    ## POINTS ##
    # Add point
    index_diff10 = np.where(difference >= 0.1)[0][0]
    diff10 = difference[index_diff10]
    z_diff10 = z[index_diff10]
    print("z at 10% difference:", z_diff10.round(3))
    plt.scatter(z[index_diff10], difference[index_diff10], color='red'

    # Add pointer lines
    plt.axhline(0.1, ls=':', c='red', lw=1)
    plt.axvline(z_diff10, ls=':', c='red', lw=1)


    ## LABELLING ##
    # Label axes
    plt.xlabel('Redshift z', fontsize=16)
    plt.ylabel('Comoving Distance', fontsize=16)

    # Increse axis numbering text size
    ax.tick_params(axis='both', which='major', labelsize=10)
    ax.tick_params(axis='both', which='minor', labelsize=10)

    # Plot features
    plt.xlim(0, 2)
```

```
    plt.ylim(0, 1)
    plt.legend(fontsize=14)
    plt.grid(True, color='lightgrey', ls='-.')
    plt.title('Comoving Distance Calculations', fontsize=18)


    ## SAVE ##
    # Save and show
    plt.savefig("HW3Q3Plot.pdf", format="pdf", bbox_inches="tight", ov
    plt.show()
```

In [13]:
```
# Create the plot
comdist_plot(z, X_numint, X_analint)
```

```
z at 10% difference: 1.333

/var/folders/1w/ktxtfrr91bj5bztz50dqm0fr0000gn/T/ipykernel_32956/3647
817420.py:52: MatplotlibDeprecationWarning: savefig() got unexpected
keyword argument "overwrite" which is no longer supported as of 3.3 a
nd will become an error in 3.6
  plt.savefig("HW3Q3Plot.pdf", format="pdf", bbox_inches="tight", ove
rwrite=True)
```



The analytical expansion is only accurate to 10% out to redshift $z = 1.333$. After that point, the expansion result curves sharply downward and becomes extremely unreliable. It's an excellent match out to roughly $z = 0.6$, so this is the realm in which is could be extremely useful to calculate comoving distance.

Oops