University of Waterloo

PHYS 474 Assignment 1

Robert Burnet
rcburnet@uwaterloo.ca
20465122

September 25, 2016

1. (a)

$$\Phi(L)dL = n_* \left(\frac{L}{L_*}\right)^\alpha e^{-\frac{L}{L_*}} \frac{dL}{L_*}$$

Number density of $L > 0.1L_*$:

$$\int_{0.1L_*}^{\infty} \Phi(L)dL = \int_{0.1L_*}^{\infty} n_* \left(\frac{L}{L_*}\right)^\alpha e^{-\frac{L}{L_*}} \frac{dL}{L_*}$$

Numerically, in Python:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats

def phi(L):
    n = 0.0037
    a = -0.96
    L_star = 10.0**11.0
    return n*(L/L_star)**a * np.e**(-L/L_star) / L_star

x = np.linspace(10**10, 10**13, 1000000)

result = np.trapz(phi(x), x)

print "space density:", result
```

Which gives $\int_{0.1L_*}^{\infty} \Phi(L)dL \approx \mathbf{0.00646\ Mpc^{-3}}$

This means that in every volume of 1 Mpc^3, we'd see 0.00646 galaxies within, or we'd see approximately 1 galaxy in a volume of 154.8 Mpc^3, which corresponds to a galaxy separation of:

$$\begin{aligned} V &= \frac{4}{3}\pi R^3 \\ \Rightarrow R &= \sqrt[3]{\frac{3V}{4\pi}} \\ &= \mathbf{3.33\ Mpc^3} \end{aligned}$$

To get the number of galaxies in the universe, simply multiply the space density of galaxies by the volume of the universe:

$$\begin{aligned} \# &= 0.00646\mathrm{Mpc}^{-3}(V_{universe}) \\ &= 0.00646\left(\frac{4}{3}\pi R_H^3\right) \\ &= \mathbf{7.43{\times}10^{10}\ galaxies} \end{aligned}$$

(b) Luminosity density of $L > 0$, $\rho_{L>0}$:

$$\int_0^{\infty} \Phi(L)LdL = \int_0^{\infty} n_* \left(\frac{L}{L_*}\right)^{\alpha+1} e^{-\frac{L}{L_*}} dL$$

Luminosity density of $L > 0.1L_*$, $\rho_{L>0.1L_*}$:

$$\int_{0.1L_*}^{\infty} \Phi(L)LdL = \int_{0.1L_*}^{\infty} n_* \left(\frac{L}{L_*}\right)^{\alpha+1} e^{-\frac{L}{L_*}} dL$$

Numerically, in Python:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats

```

```
def phiL(L):
    n = 0.0037
    a = -0.96
    L_star = 10.0**11.0
    return n*(L/L_star)**(a+1) * np.e**(-L/L_star)

x_0 = np.linspace(0, 10**13, 1000000)
x_10 = np.linspace(10**10, 10**13, 1000000)

result_0 = np.trapz(phiL(x_0), x_0)
result_10 = np.trapz(phiL(x_10), x_10)

print "L > 0 luminosity density:", result_0
print "L > 0.1 L_star luminosity density:", result_10
```

Which gives $\int_0^\infty \Phi(L)LdL \approx \mathbf{3.62{\times}10^8\ L_\odot/Mpc^3}$

and $\int_{0.1L_*}^\infty \Phi(L)LdL \approx \mathbf{3.31{\times}10^8\ L_\odot/Mpc^3}$

and $\frac{\rho_{L>0.1L_*}}{\rho_{L>0}} = \mathbf{0.91}$

Therefore, they make up about 91% of the luminosity density of the universe.

(c)

$$\begin{aligned}\frac{0.3\rho_{crit}}{\rho_{L>0}} &= \frac{0.3\left(\frac{3H_0^2}{8\pi G}\right)}{3.62\times10^8}\\ &= 2.24\times10^{32}kg\mathrm{L}_\odot\\ &= \mathbf{112.7\ M_\odot/L_\odot}\end{aligned}$$

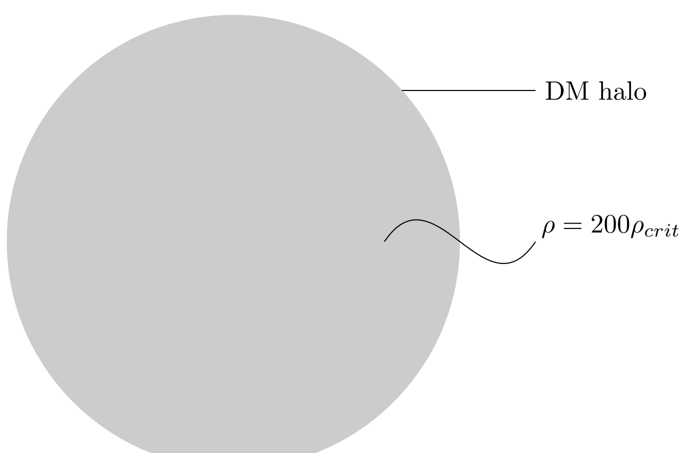
Where $H_0 = (70\text{km/s/Mpc})(\frac{3.2408\times10^{-20}\text{Mpc}}{1\text{km}}) = 2.2686\times10^{-18}\text{s}^{-1}$
and $G = (6.674\times10^{-11}\text{m}^3/\text{kg/s}^2)(\frac{3.2408\times10^{-20}\text{Mpc}}{1000\text{m}})^3 = 2.272\times10^{-78}\text{Mpc}^3/\text{kg/s}^2$

(d) The largest value for M/L_* I could obtain was 2.31 with age of 13.7 Gyrs and metallicity of -1.999. My calculated value was $\approx$ 49 times larger than the value I obtained from the model.

(e)

$$M \propto L \Rightarrow M = 112.7\frac{M_\odot}{L_\odot}L$$

$$M_{universe} = M_{DM,universe}$$



$$@L_* \Rightarrow M_* = 112.7\frac{M_\odot}{\cancel{L_\odot}}10^{11}\cancel{L_\odot}$$
$$= 1.127 \times 10^{13} M_\odot$$

$$\rho = 200\rho_{crit}$$
$$= 200(0.3\left(\frac{3H_0^2}{8\pi G}\right))$$
$$= 1.6225 \times 10^{43}\text{kg/Mpc}^3$$
$$= 8.1575 \times 10^{12} M_\odot/\text{Mpc}^3$$

$$\rho = \frac{M}{V}$$
$$8.1575 \times 10^{12} M_\odot/\text{Mpc}^3 = \frac{1.127 \times 10^{13} M_\odot}{\frac{4}{3}\pi R^3}$$
$$R = \sqrt[3]{\frac{1.127 \times 10^{13}}{8.1575 \times 10^{12}}\left(\frac{3}{4\pi}\right)}$$
$$= \mathbf{0.691\ Mpc}$$

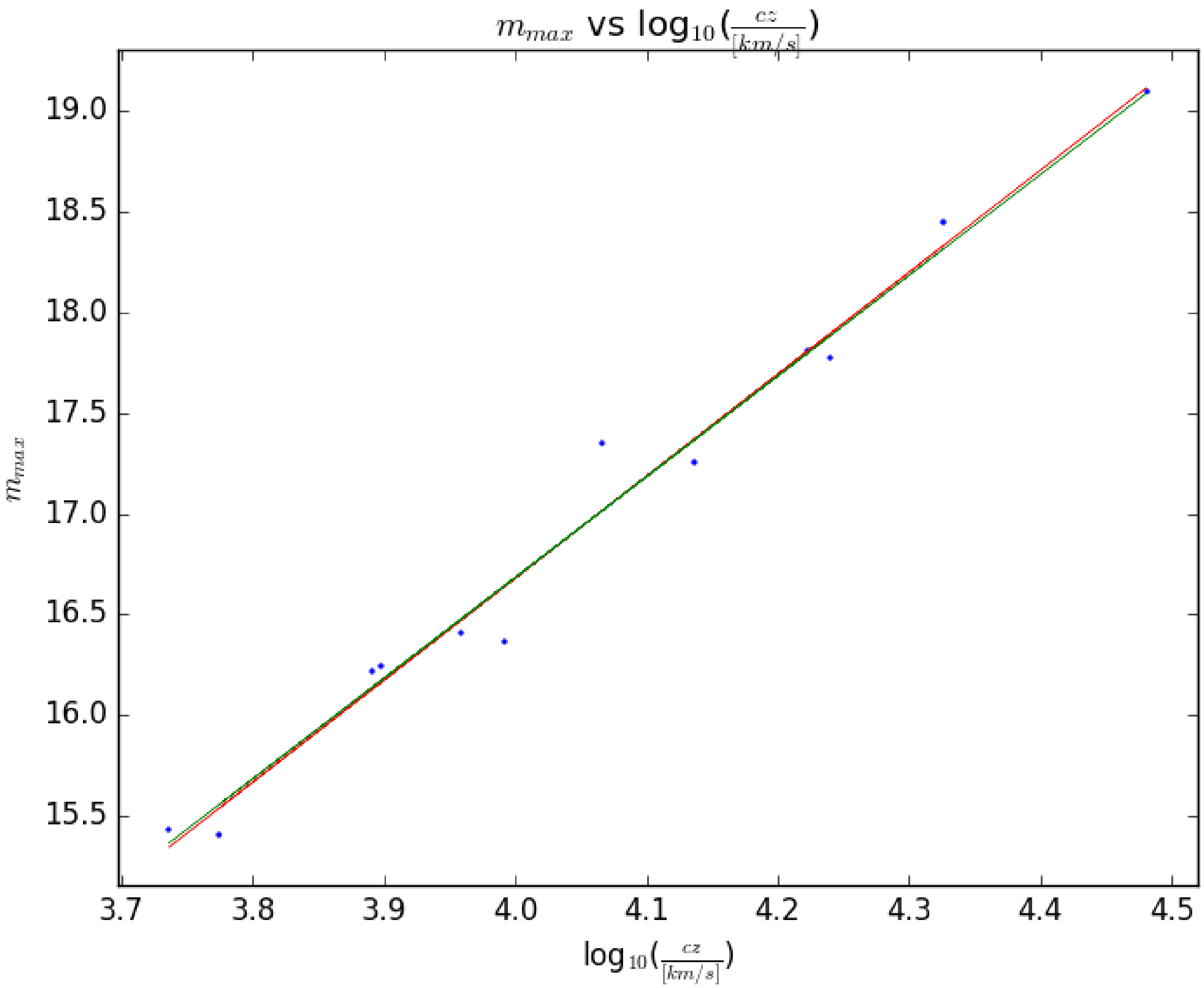
(f) $M_K = -24.0$
$M_{bol} - M_{bol,\odot} = -2.5log\left(\frac{L}{L_\odot}\right)$
$M_{bol} = BC_K - M_K$

etc...

(g)

2. (a)
```
log_cz = np.array([3.958, 4.066, 4.223, 3.992, 4.137, 3.897, 3.891, 4.481, 3.774,
    4.240, 3.736, 4.326])
m = np.array([16.4090, 17.3488, 17.8124, 16.3683, 17.2548, 16.2426, 16.2164,
    19.0955, 15.4026, 17.7777, 15.4329, 18.4524])

# Linear regression to find best fit line:
slope, intercept, r_value, p_value, std_err = stats.linregress(log_cz, m)

std_dev = np.std(m)

print "r_squared with fitted slope:", r_value**2
print "fitted slope:", slope
print "intercept with fitted slope:", intercept

# Do it for fixed slope of 5.0:
slope_fixed = 5.0
intercepts_fixed = m - slope_fixed*log_cz
intercept_fixed = np.mean(intercepts_fixed)

std_intercepts = np.std(intercepts_fixed)

print "intercept with fixed slope:", intercept_fixed

print "standard deviation of magnitudes:", std_dev
print "standard deviation of intercepts:", std_intercepts

lobf_fit = slope*log_cz + intercept
```

```
lobf_fixed = slope_fixed*log_cz + intercept_fixed

points = plt.scatter(log_cz, m, s = 2, color='b')
line1 = plt.plot(log_cz, lobf_fit)
line2 = plt.plot(log_cz, lobf_fixed)
plt.setp(line1, color='r', linewidth=0.3)
plt.setp(line2, color='g', linewidth=0.3)
plt.title('$m_{max}$ vs log$_{10}$($\\frac{cz}{[km/s]}$)')
plt.ylabel('$m_{max}$')
plt.xlabel('log$_{10}$($\\frac{cz}{[km/s]}$)')
plt.savefig('a02q2ae.png')
```



(b)

$$M = m - 5log\left(\frac{d}{10}\right)$$

Where M is constant

$$cz = H_0 d \times \frac{1\text{Mpc}}{10^6\text{pc}} \Rightarrow d = \frac{cz}{H_0} \times 10^6$$

$$\Rightarrow M_{max} = m_{max} - 5log\left(\frac{cz}{H_0}10^5\right)$$

Therefore,

$$m_{max} = 5log(cz) - 5log\left(\frac{H_0}{10^5}\right) + M_{max}$$

This is a line of m_{max} vs $log(cz)$ with a slope of 5.0 and intercept of $M_{max} - 5log\left(\frac{H_0}{10^5}\right)$

(c) as above, the intercept is $M_{max} - 5log\left(\frac{H_0}{10^5}\right)$

$$\Rightarrow m_0 = M_{max} - 5log\left(\frac{h}{10^3}\right)$$

Where $h = \frac{H_0}{100\text{km/s/Mpc}}$

(d) i. redshift and m_{max} measurements may have uncertainties which result in discrepancy between expected and actual values that aren't accounted for here.

ii. m_{max} is flux dependent, which depends on intensity of light on CCD camera. Light from different parts of the sky go through different amounts of dust and may be absorbed more or less than other light sources.

iii. the second reason is probably more significant as the data is given to us to 4 and 6 significant digits, which an uncertainty of ± 0.0005 for $log(cz)$ and ± 0.00005 for m_{max} wouldn't account for the amount of discrepancy seen, assuming all the digits reported in the table are certain.

(e) from output of code above in a):

r^2 = 0.9817 (statistical uncertainty of linear regression)
slope = 5.065 ≈ 5.1
intercept = -3.580

Our expectations are a slope of 5.0, I get a slope of 5.1 which is not consistent with out expectations.

With fixed slope of 5.0, I get an intercept of -3.316, with standard deviation (uncertainty) of 0.151.

The standard deviation in magnitudes of the points is 1.114.

See plot above in a) for plotted lines of best fit.

(f)

$$m_0 = M_{max} - 5log\left(\frac{h}{10^3}\right)$$

$$\begin{aligned} m_{max} - M_{max} &= 32.0 \text{ (for NGC 4639)} \\ 12.61 - M_{max} &= 32.0 \\ \Rightarrow M_{max} &= -19.39 \end{aligned}$$

Therefore,

$$m_0 = -19.39 - 5log\left(\frac{h}{10^3}\right)$$

I found that m_0 = -3.580 for fitted slope, and -3.316 for fixed slope.

Therefore, $h_1 \approx 0.69$ for the fitted slope (slope ≈ 5.1) and $h_2 \approx 0.61$ for the fixed slope (slope = 5.0).

$$h_{ave} = \frac{h_1 + h_2}{2} \approx \mathbf{0.65}$$

(g)

$$
\begin{aligned}
h &= 10^{\frac{-m_0+19.39}{5}+3} \\
dh &= \pm\left|\frac{\partial h}{\partial m_0}\right||dm_0| \\
&= \pm|-0.0609882 \times 10^{\frac{-m_0}{5}}||dm_0|
\end{aligned}
$$

Where $m_0 = -3.58$ or $= -3.316$ and $|dm_0| = 0.151$

Therefore,

$$
\begin{aligned}
dh_1 &= \pm 0.048 \text{ (for } m_0 = -3.58) \\
dh_2 &= \pm 0.042 \text{ (for } m_0 = -3.316)
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
dh_{ave} &= \pm\sqrt{\left(\frac{\partial h_{ave}}{\partial h_1} dh_1\right)^2 + \left(\frac{\partial h_{ave}}{\partial h_2} dh_2\right)^2} \\
&= \pm\sqrt{\frac{1}{4}\left(dh_1^2 + dh_2^2\right)} \\
&= \pm\frac{1}{2}\sqrt{dh_1^2 + dh_2^2} \\
&= \pm\mathbf{0.032}
\end{aligned}
$$