

Solutions to Astr 540 HW #1

1) a. Using interactive software of your choice (eg., IDL, spreadsheet, Mathematica,...) fit these data with a two-component model using a bulge and a disk. You should turn in a plot showing your fit, the functional forms you used, and the fitting parameters. Do not be tempted to use one of the canned bulge/disk fitting programs that are relatively easy to find, as part of the value of this exercise is for you to see how tricky it can be to perform this type of fitting, and to see why there have been debates about disk central surface brightnesses, and so on. Don't worry about trying to produce the very best fit, but get close and indicate what you think is a plausible range that encompasses the best fit parameters.

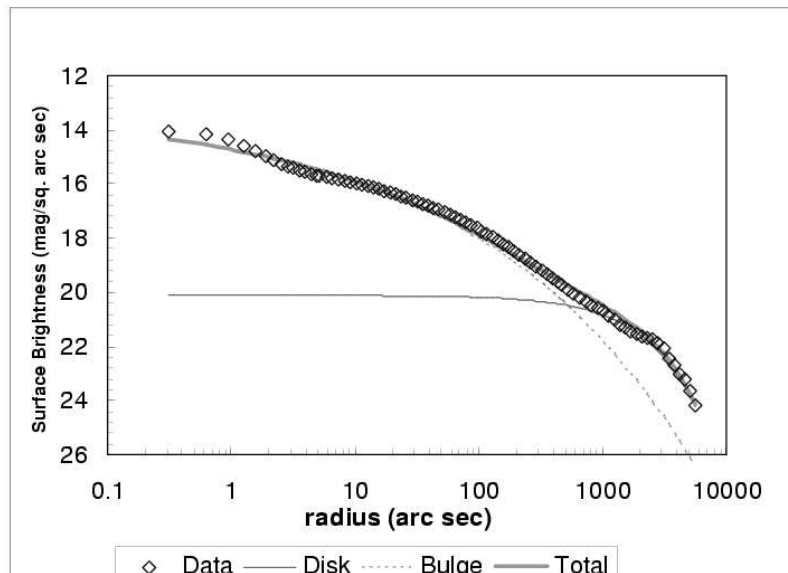
Fit the bulge using $\log I(r)/I(r_e) = -3.33[(r/r_e)^{1/4} - 1]$ (see slide 4 from Lec 5). (note that there are alternative formulations that are also acceptable)

Fit the disk using $I(r) = I_0 e^{-r/h}$ (see slide 9 from Lec 2).

Typical fit parameters: (in magnitudes):

$r_e = 875''$ $I(r_e) = 10.70$ mags/sq. arc sec (recall that $I(r_e)$ is flux within r_e)

$I_0 = 20.11$ mags/sq. arc sec $r_0 = 1475''$



b. Knowing that these data are from M31, what are the physical lengths associated with your fitting parameters?

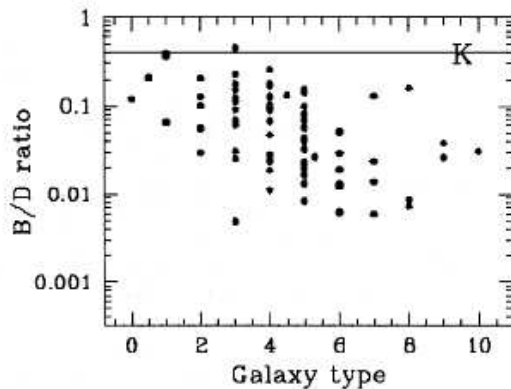
$D = 0.78 \text{ Mpc}$ $1'' = 4.84 \times 10^{-6} \text{ rad}$ which corresponds to 3.8 pc at M31 so $r_e = 3.3 \text{ Kpc}$ and $r_0 = 5.6 \text{ kpc}$.

c. Using data out to ~ 5000 arc sec, what's the bulge-to-disk luminosity ratio for M31? Does this match its Hubble type

Compute using your fit:
$$\frac{B}{D} = \frac{\sum_r \frac{[BFit(r_2) + BFit(r_1)]}{2} * \pi * (r_2^2 - r_1^2)}{\sum_r \frac{[DFit(r_2) + DFit(r_1)]}{2} * \pi * (r_2^2 - r_1^2)}$$
 - need to sum

over the flux x area product

Fits above give $B/D = 0.37$ - shown as a line below so this looks on the high side but plausible for an early-type spiral (should be more careful about specifying what wavelength was used for the observation - plot is for K!).



d. If the stars in the M31 bulge have $M/L = 3$, over what distance would a nuclear black hole with mass $= 10^7 M_{Sun}$ dominate the gravitational field?

This amounts to determining at what r the gravitational field of the black hole equal the field from the enclosed stellar mass.:

$$F_{BH} = \frac{GM_{BH}}{r^2} = \frac{GM_r(*)}{r^2}$$

$\frac{M}{L} = 3$ in solar units so for $M_{BH} = 1 \times 10^7 M_{Sun}$

$$L = 3.3 \times 10^6 L_{Sun}$$

To complete the calculation, the observed magnitudes need to be converted to luminosities - the homework didn't specify a wavelength for the

data which would be needed to answer this question rigorously. For an estimate type of calculation, note that $m-M$ for M31= 24.5 and that $M_V = +4.7$ for the Sun. We want the r for which the integrated flux equals the L above. The luminosity quoted above corresponds to an absolute magnitude $M_V = -2.5 \log(3.3 \times 10^6 L_{Sun}) + 4.7 = -11.6$.

Apparent magnitude is $V = 24.5 + -11.6 = 12.9$

Looking at the data and finding the point where the type of sum computed for b) but from the total observed flux, the r will be around $0.8'' = 3.0$ pc. However, the stellar population in M31's nucleus doesn't have the same color as the Sun so realistic bolometric corrections should be applied - actual r is likely to be slightly smaller than what we've computed here. Nonetheless, this makes the point that the black hole does not directly influence much of the surrounding galaxy.

2) Rotation curve for a generalized NFW halo.

a. For a spherical dark matter halo density profile given by

$$\rho(r) = \rho_0 \frac{1}{(r/r_s)^\alpha (1 + r/r_s)^{3-\alpha}} \quad (1)$$

derive the disk rotation curve in terms of rotation speed at the virial radius v_{200} , scaled radius $x \equiv r/r_s$ and concentration $c \equiv r_{200}/r_s$ (along with inner slope α). Assume that the mean density within the virial radius is 200 times the critical density. You can ignore the contributions to the rotation curve from baryonic components. Show your work.

Assuming spherical symmetry and no baryonic component, we can write $v_c(r) = \sqrt{GM(<r)/r}$. To obtain the mass contained within radius r , $M(<r)$, we integrate the radial density profile given above:

$$M(<r) = \int_0^r \rho(r') 4\pi r'^2 dr' \quad (2)$$

Substituting $x = r/r_{200}$ and $c = r_{200}/r_s$ gives $r/r_s = xc$, and if we define $y \equiv r'/r_s$ then we obtain

$$M(<r) = 4\pi r_s^3 \int_0^{xc} \rho_0 \frac{y^2}{y^\alpha (1+y)^{3-\alpha}} dy. \quad (3)$$

Defining the integral

$$\mu(x) = \int_0^x y^{2-\alpha}(1+y)^{\alpha-3} dy, \quad (4)$$

we can write

$$M(< r) = 4\pi r_s^3 \rho_0 \mu(xc) \quad (5)$$

Now we note that $v_{c,200} = \sqrt{GM(< r_{200})/r_{200}}$, where

$$M(< r_{200}) = 4\pi r_s^3 \rho_0 \mu(c), \quad (6)$$

since $x = 1$. Hence

$$v_c = v_{200} \sqrt{\frac{\mu(xc)r_{200}}{\mu(c)r}} \quad (7)$$

$$= v_{200} \sqrt{\frac{\mu(xc)}{x\mu(c)}}. \quad (8)$$

This is the formula given in equation 6 of Swaters et al. (2003).

b. Write a C program to input v_{200} , c , and α , and output the rotation curve out to the virial radius, in units of kpc vs. km/s. You can email the C program to Romeel; it should compile under gcc. If you do not know C, this is a good chance to learn!

To write such a program with actual values requires knowing r_{200} , in order to scale x to physical units. One way to obtain this is from Mo, Mao and White (1998), who give $r_{200} = v_{200}/10H(z)$. We will assume $z = 0$ and $H_0 = 70$ km/s/Mpc. A program to do this calculation using a 2nd-order accurate integrator (trapezoidal rule) is shown below (sorry about the messed up indentation– apparently latex doesn't like tabs in the verbatim environment; too lazy to fix).

```
#include <stdlib.h>
#include <stdio.h>
#include <math.h>

#define NSTEPS 200 /* number of integration steps */
#define H0 0.070 /* in km/s/kpc */
```

```

float mu\_int(float x, float alpha) // integrate using trapezoidal rule
{
float y,dy,mu,int0,int1;

dy = x/NSTEPS;
for(y=0, mu=0; y<x; y+=dy) {
    int0 = pow(y,2-alpha)*pow(1+y,alpha-3); // integrand at start
    int1 = pow(y+dy,2-alpha)*pow(1+y+dy,alpha-3); // integrand at end
    mu += 0.5*(int0 + int1)*dy; // average to obtain 2nd order accuracy
}
return mu;
}

int main(int argc,char **argv)
{
float v200,c,alpha;
float x,dx,v;

if( argc != 4 ) { // argc = # of command-line values
    fprintf(stderr,"usage: vrot v200 c alpha\n");
    exit(-1);
}
else {
    sscanf(argv[1],"%g",&v200); // argv = array containing parsed
    sscanf(argv[2],"%g",&c); // strings of command-line input
    sscanf(argv[3],"%g",&alpha);
}

dx = 1./NSTEPS; // width of integration steps
/* calculate rotation curve */
for (x=dx; x<=1.+0.5*dx; x+=dx) {
    v = v200 * sqrt(mu\_int(x*c,alpha)/(x*mu\_int(c,alpha)));
    fprintf(stdout, "%f\t%f\n", x*v200/10./H0, v); // x -> r
}

exit(0);
}

```