Numerical Recipes: Hand-In 2

Rens Kievit (s1948415)

March 28, 2023

Abstract

In this report we present the problems, solutions and scripts for the exercises from the second handout for the course Numerical Recipes.

Plotting styles in this report are set using the following code

```python
def set_styles():
    """For consistent plotting scheme"""
    plt.style.use('default')
    mpl.rcParams['axes.grid'] = True
    plt.style.use('seaborn-darkgrid')
    mpl.rcParams['font.family'] = 'serif'
    mpl.rcParams['lines.linewidth'] = 1.5
    mpl.rcParams['font.size'] = 14
```

Listing 1: Matplotlib Plotting Styles

1 Satellite Galaxies Around a Massive Central

All dynamically written results from the code corresponding to this section can be found in `results/satellite_galaxies_results.txt`.

In this section we will investigate the spherical distribution of satellite galaxies around a massive central galaxies. Their density distribution n can be described

$$n(x) = A \langle N_{sat} \rangle \left(\frac{x}{b} \right)^{a-3} \exp \left[- \left(\frac{x}{b} \right)^c \right] \tag{1}$$

Where we take $a = 2.4$, $b = 0.25$, $c = 1.6$ and $\langle N_{sat} \rangle = 100$. x is the radius relative to the virial radius, i.e. $x \equiv r/r_{vir}$. A is a normalization constant that we do not a priori know, but is set such that the three dimensional spherical from $x = 0$ to $x_{max} = 5$ is equal to the average number of satellites, $\langle N_{sat} \rangle$:

$$\int \int \int_V n(x) dV = \langle N_{sat} \rangle . \tag{2}$$

1.1 Normalization Constant

Looking at equation 1 we can see that it is independent of angle, and only depends on the radial distance. This allows us to transform the integral in equation 2 to a spherical integral over θ, ϕ, x as

$$\begin{aligned}
\int \int \int_V n(x) dV &= \int_0^{x_{max}} \int_0^{\pi} \int_0^{2\pi} n(x) x^2 \sin(\theta) d\phi d\theta dx \\
&= 4\pi \int_0^{x_{max}} n(x) x^2 dx \\
&= 4\pi \int_0^{x_{max}} x^2 A \langle N_{sat} \rangle \left(\frac{x}{b} \right)^{a-3} \exp \left[- \left(\frac{x}{b} \right)^c \right] dx
\end{aligned}$$

Here we begin by setting $A = 1$, and using the result of this integral to calibrate A such that equation 2 holds. We can evaluate this integral using a simple one dimensional numerical integrator. We apply a Romberg integrator with order 10 to ensure sufficient accuracy. We have to account for the fact that $n(x)$ is not defined at $x = 0$ because $\left(\frac{x}{b} \right)^{a-3}$ causes a division by zero if $a < 3$, which is the case here. To combat this we set $h_{start} = \frac{a-b}{2}$ instead of $h_{start} = a - b$ to avoid evaluating $n(x = 0)$.

We find $\int_V n(x) dV = 10.88$, which means we have to set the normalization constant $A = \frac{\langle N_{sat} \rangle}{10.88} = \frac{100}{10.88} = 9.19$. We will use this value for A throughout the rest of this work.

1.2 Distribution Sampling

We want to sample the 3D distribution of satellites such that they statistically follow the distribution in equation 1. This means that the the probability distribution should be $p(x) dx = N(x) dx / \langle N_{sat} \rangle$. Here $N(x) dx$ is the number of satellites in a spherical shell of size dx at relative radius x from the center of
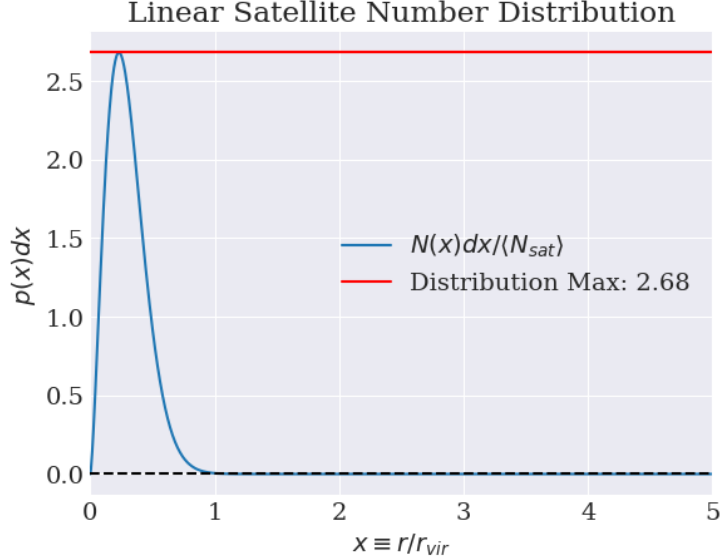
Figure 1: Analytical distribution of the number of satellite galaxies in a shell around a massive galaxy with thickness dr at radius r, described by equation 3.

the massive galaxy. This is equal to the result we saw for the spherical integral in the previous section, thus the probability distribution we want to sample is

$$p(x)dx = x^2 A \left(\frac{x}{b}\right)^{a-3} \exp\left[-\left(\frac{x}{b}\right)^c\right]dx \tag{3}$$

In this work we will use rejection sampling to simulate this distribution. The implementation in this work first generates two random numbers generated using `32-bit multiply with carry`, we call the first number x and shift it to a uniform distribution in log-space in the range $\left[-4, ^{10}\log(5)\right]$ (corresponding to the range $\left[10^{-4}, 5\right]$ in linear space), and the second we call y and shift it to the range $[0, 2.68]$ where 2.68 is the maximum of the distribution given in equation 3 as shown in figure 1. If the (x_i, y_i) point falls below the curve described by equation 3 we accept it, otherwise we reject it[1]. After each step we use the last generated random number as `x0` for the next number. We repeat this entire process, until we have $N = 10^4$ samples in total. We note that rejection sampling is far from the most efficient method to sample a distribution such as this because the majority of possible points in our (x, y) range will be rejected as is evident in figure 1. To slightly combat this we have sampled x in log- instead of linear space which means small values (< 1) are sampled at a higher rate, and the algorithm finds itself in the "interesting" region of the distribution more often. Nevertheless, on average the rejection sampling algorithm still takes ~ 7 attempts (14 random numbers generated) untill it finds an accepted point.

We plot the results from our sampling algorithm as a histogram with 20 equally spaced bins in log-space in figure 2. We divided the height of each bin by 10^3 to account for the difference between $\langle N_{sat} \rangle$ and N. We can see in

---

[1]In the code, we have replaced each x in equation 3 with 10^x to account for the fact we sample in log-space.
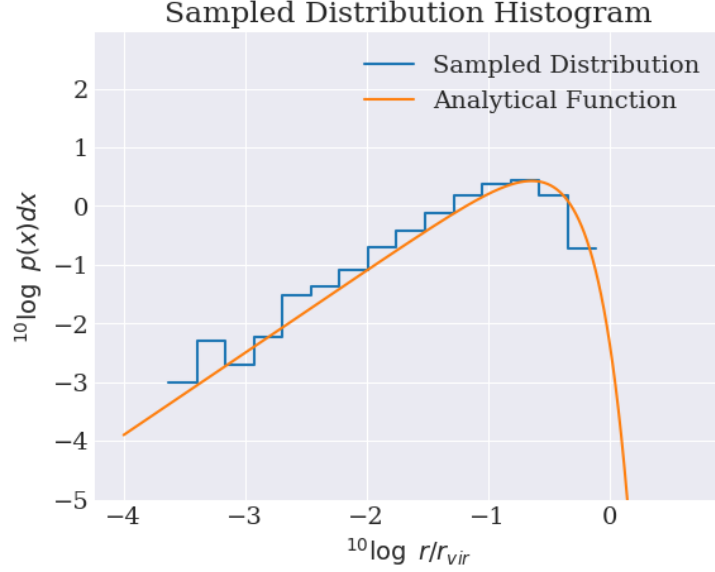
Figure 2: Analytical distribution of satellite galaxies around a massive central overplotted on the same distribution sampled by our rejection algorithm with $N = 10^4$ described in the text. The histgoram bin heights are scaled down by a factor 10^3 to enforce the same normalization.

the figure that the analytical distribution, and our sampled distribution match almost perfectly everywhere except the edges. These edges are missing from this plot because the sampler is often unable to sample points in these regions due to the large value for $p(x)dx$ there.

1.3 Random Selection

We want to look at the cumulative distribution function (CDF) of the sampled probability distribution shown in figure 2. To do this we first select 100 random satellite galaxies following the criteria given on the handout: each galaxy is selected with equal probabilty, no galaxy is selected twice, and we do not reject any draws. The simple (but maybe relatively overboard) method we use to ensure these criteria are upheld is to first shuffle the array of 10^4 galaxies in a random order, and then select the first 100.

We perform the random shuffling by generating 10^4 random numers using our random number generator mentioned earlier, and use these numbers as keys in our `merge_sort` algorithm on which to "sort" an array of indices. We then apply these "sorted" indices on the sampled array to shuffle it, and select the first 100 instances. Finally, we sort these 100 galaxies using the same algorithm on their radii directly. We then use this sorted array to plot a CDF, normalized to 1, and present this in figure 3.

In correspondence with the PDF we saw in figure 2, we can see there are almost no satellites present at either edges of the interval. The most satellite galaxies are found at $\sim x = 0.1$, a result we also saw earlier.
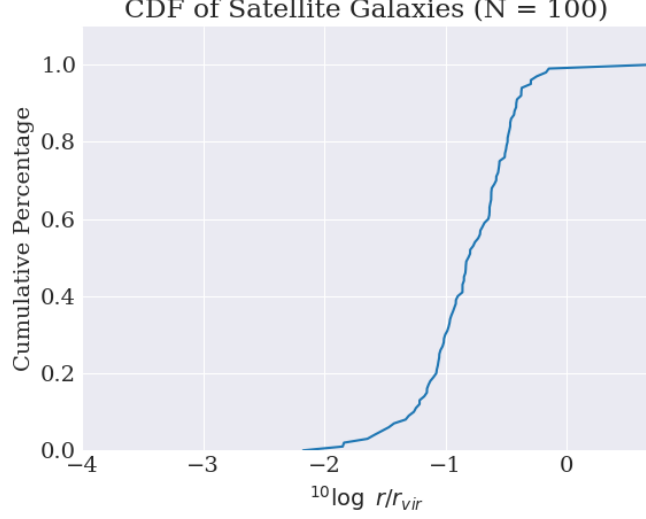
4

Figure 3: Cumulative distribution function normalized to 1 of $p(x)dx$ sampled using rejection sampling with $N = 10^4$, created using only a random subset of 100 satellites.

1.4 Derivative

To conclude this section we will look at the derivative of $n(x)$ (equation 1, not $N(x)$!) at $x = 1$. To compute this derivative we have used Ridder's Method with $h_{start} = 0.1$. Similarly to the Romberg integration earlier, we cannot choose $h_{start} = 1$ because then we would have to sample $n(x = 0)$, where the function is not defined. In agreement with the assignment, we set our target accuracy at 10^{-12}. This means we stop if our uncertainty, defined as the absolute difference between the current and previous best estimates, drops below this threshold. We compare our findings to the analytical derivative of $n(x)$ which we find to be

$$\frac{dn}{dx}_{(analytical)} = -A \langle N_{sat} \rangle \exp\left[-\left(\frac{x}{b}\right)^c\right] \frac{b^3 \left(\frac{x}{b}\right)^a \left(c\left(\frac{x}{b}\right)^c - a + 3\right)}{x^4} \quad (4)$$

Now comparing the two results, using Ridder's Method we find

$$\frac{dn}{dx}_{(ridder)} = -0.625328061833 \pm 2.220 \times 10^{-14}$$

While analytically we find

$$\frac{dn}{dx}_{(analytical)} = -0.625328061833$$

These two values match exactly up to 12 digits, indicating the strength of the numerical differentiator implemented here.

5

2 Heating and Cooling in HII regions