Computational Physics Homework 2

Zubair Bhatti

Fall 2019

8.5/10

1 Relaxation and Overrelaxation

$$x = 1 - e^{-cx} \tag{1}$$

Solving non-linear equations may not be analytically possible for all cases. However, the relaxation method can solve equations like (1). For example, with a tolerance of 10^{-7} it can compute a solution within 17 iterations. However, with a slight adjustment, introducing the overrelaxation method, that number dropped to 6 iterations. The relaxation method takes smaller steps for each new guess of for a solution to the equation. In contrast, the overrelaxation method has a hyper parameter ω that allows the user to tune the size of jumps made during each iteration towards a solution. If $\omega = 0$ the original relaxation method is recovered. If $\omega < 0$ then the process is under relaxation and could help find solutions to highly varied nonlinear equations where overrelaxation divereges. However, not all is perfect with the overrelaxation method, since its error scales a bit more complicated than that of the relaxation method.

$$\begin{aligned}
assume f'(x)\epsilon &= \epsilon' \\
\frac{df(x)}{dx}\epsilon &= \epsilon' \\
\frac{dx'}{dx}\epsilon &= \epsilon' \\
\frac{dx'}{dx}(x^* - x) &= \epsilon'
\end{aligned}$$

dont put text into equations. but fyi: use {\rm ...} to get roman font.

$$
\begin{aligned}
\frac{dx'}{dx}(x^* - x' - x + x') &= \epsilon' \\
(x^* - x')\frac{dx'}{dx} - (x - x')\frac{dx}{dx} &= \epsilon' \\
\epsilon'\frac{dx'}{dx} - \epsilon' &= (x - x')\frac{dx'}{dx} \\
\epsilon' &\simeq \frac{x - x'}{1 - \frac{1}{\frac{dx'}{dx}}} \\
\epsilon' &= \frac{x - x'}{1 - \frac{1}{[(1+w)f'(x) - w]}} \\
\epsilon' &= \frac{x - x'}{1 - \frac{1}{[(1+w)f'(x^*) - w]}}
\end{aligned}
$$

okay, but where's the rest of the problem?

-1.5

2 Wien's Displacement Constant

$$I(\lambda) = \frac{2\pi hc^2\lambda^{-5}}{e^{hc/\lambda k_b T} - 1} \tag{2}$$

$$\frac{d}{d\lambda}I(\lambda) = 0 = -5 + \frac{hc}{\lambda k_b T} + 5e^{hc/\lambda k_b T} \tag{3}$$

$$5e^{-x} + x - 5 = 0, where x = hc/\lambda k_b T \tag{4}$$

Planck's radiation law for black bodies (2) has a maximum when its derivative with respect to λ is taken to be 0 (3). This is useful to find the the peak wavelength omitted from a source. After finding the equation for the maximum in terms of x, (4) we see we are left with a nonlinear equation to solve. Using the binary search method which splits up sections to search in half, we can converge on a solution where $x \simeq 4.965$ so Wien's constant is 0.00289 m*K. Taking this value and comparing it to the sun's peak radiation, we can backtrack into the sun's temperature to be 5772.45 K.

3 Fitting the Schechter Function

The Schechter function describes the density of galaxies in a volume with respect to the luminosity output of the region.

$$n(M_{gal}) = \phi^* \frac{M_{gal}}{M_*}^{\alpha+1} exp(\frac{-M_{gal}}{M_*}) ln(10) \tag{5}$$

The Schechter function takes one variable M_{gal}, the mass of the galaxy and three parameters ϕ^*, the amplitude, M_*, the characteristic mass of galaxies, and α, which describes the power law slope. Since the Schechter function is not universal, it is necessary to take observations of a region in space in order to find the right parameters that define the Schechter function for the region. For a given region, the parameters that best fit the data can be found by taking a measure of good fit of an initial guess of parameters and fiddling them until a model is produced that fits the data well. The way we approach this is to minimize χ^2 through gradient descent.

$$f = \chi^2 = \sum_i^N \frac{[n_{obs}(M_{gal,i}) - n_{est}(M^i_{gal,i})]^2}{\sigma_i^2} \tag{6}$$

Algorithm 1 Gradient Descent

$\vec{x}_0 \leftarrow (\phi^*_0, M_{*,0}, \alpha_0)$
$\vec{x}_i \leftarrow \vec{x}_0$
$\epsilon \leftarrow 10^{-6}$
$s \leftarrow 1$
$s_{max} \leftarrow 10^6$
$\gamma \leftarrow 10^{-5}$
while $s < s_{max}$ **do**
  $\vec{x}_{i+1} \leftarrow \vec{x}_i - \gamma \nabla f(\vec{x}_i)$
  if $|f(\vec{x}_{i+1}) - f(\vec{x}_i)| < \epsilon$ **then**
    return
  end if
  $\vec{x}_i \leftarrow \vec{x}_{i+1}$
end while

Due to the non-convex nature of the Schechter function, some difficulty arises when trying to choose starting parameters. For 2 dimensional, it is possible to try different combinations of parameters for the function you will try and minimize and visualizing through contour plots will help you find a good starting point. However, in higher dimensions, finding the right starting points is just as difficult as running gradient descent.

Going through the process of fitting the Schechter function elucidated next steps to improve the provided gradient descent algorithm. For example, γ is the same for all components of $\vec{x}_i$. Further, for some added sophistication, instead of using the central method for calculating the gradient, one could introduce using smarter ways of taking numerical derivatives and experimenting with an adaptive learning rate so γ can vary from step to step.

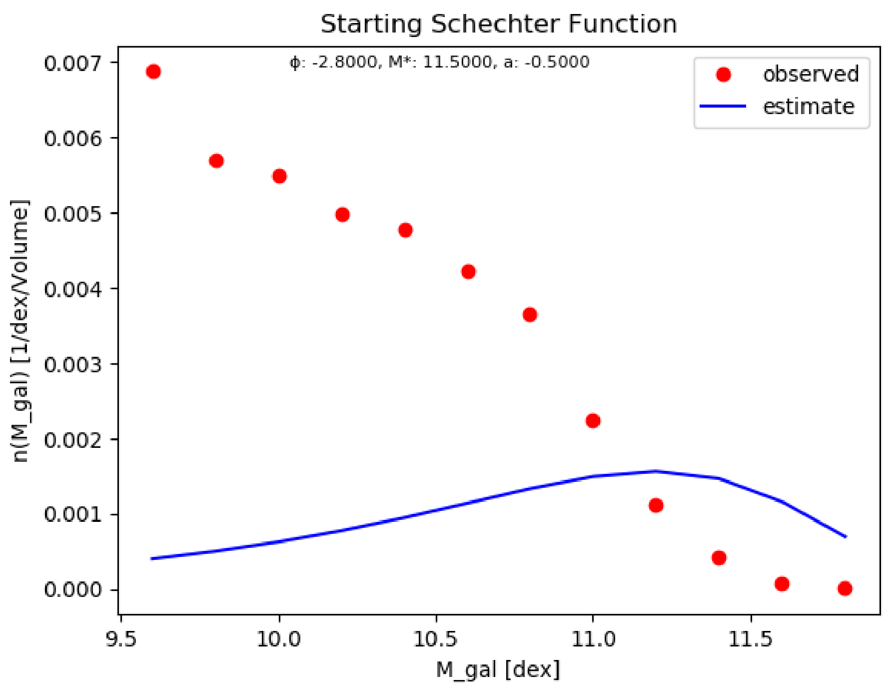


Figure 1: Plot of fit of Schechter function before running gradient descent

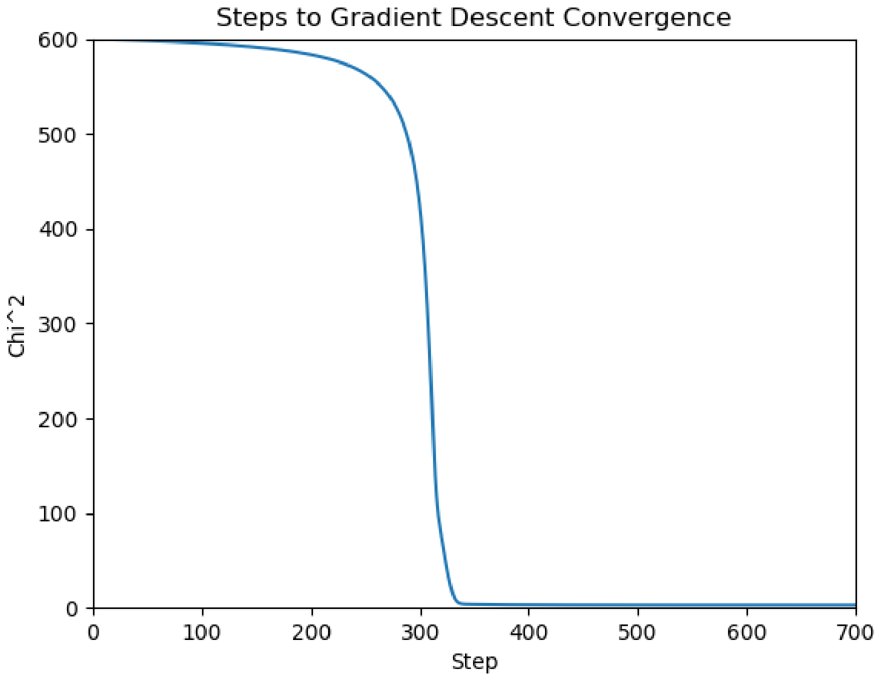
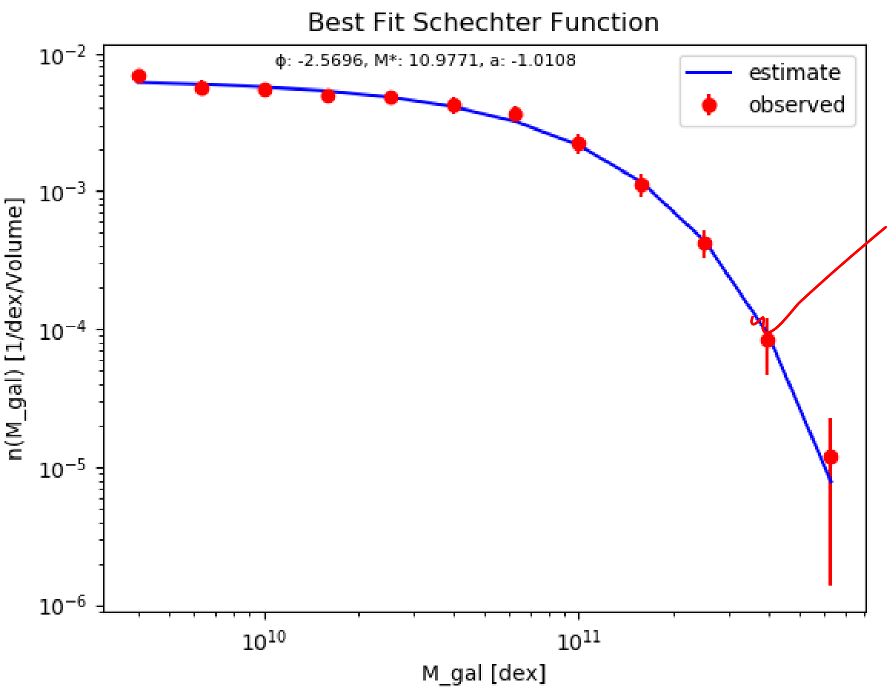


Figure 2: Plot of χ^2 as gradient descent proceeds



very good.

Figure 3: Plot of fit of Schechter function after running gradient descent