Computational Physics : Assignment 2

Moaz Abdelmaguid Mohamed (maa1142)

October 11, 2019

10/10

Question 6.13: Wien's Displacement Constant :

Plank's Radiation law:

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

(a) To obtain the the wavelength (λ) corresponding to the strongest emitted radiation, we take the derivative of planck's radition law w.r.t (λ) and set it to zero:

$$\frac{\mathrm{d}I(\lambda)}{\mathrm{d}\lambda} = 2hc^2\left(\frac{hc(e^{hc/\lambda k_B T})}{kT\lambda^7(e^{hc/\lambda k_B T} - 1)^2} - \frac{5}{\lambda^6(e^{hc/\lambda k_B T} - 1)}\right) = 0$$

Taking $\frac{2hc^2}{\lambda^6 e^{hc/\lambda k_B T} - 1)^2}$ as a common factor and dividing both sides of the equation by it:

$$5e^{hc/\lambda k_B T} + \frac{hc}{\lambda k_B T} - 5 = 0$$

Now multiply both sides of the equation by $e^{-hc/\lambda k_B T}$, we have:

$$5e^{-hc/\lambda k_B T} + \frac{hc}{\lambda k_B T} - 5 = 0$$

Therefore, the solution to this equation is the value of λ at which the emitted radiation is strongest

* Now we make the substitution $x = \frac{hc}{\lambda k_B T}$ and solve for λ in terms of x:

$$\lambda = \frac{hc}{x k_B T}$$

Now we define Wien's displacement constant (b) as follows : $b = \frac{hc}{k_B x}$

Plugging (b) in the equation for λ we can see that the wavelength of maximum radiation indeed obeys the Wien displacement law :

$$\lambda = \frac{b}{T}$$

After making the substitution in x, we will be left with the following nonlinear equation:

$$5e^{-x} + x - 5 = 0$$

(b) After applying the binary search method (code attached), the yielded value of x is 4.965114116668701 in addition to 0 Substituting in our equation for wien's displacement law:

$x = 4.965114116668701$

$K_B = 1.38064852 * 10^{-23} m^2.k^{-1}.kg.s^{-2}$

$c = 3 * 10^8 m.s^{-1}$

$h = 6.626176 * 10^{-34} J.s^{-1}$

The value of Wien's displacement constant :

$$b = \frac{hc}{k_B x} = 2.9 * 10^{-3} m.k$$

(c) The estimated value of the surface temperature would be:

$$T = \frac{b}{\lambda} = 5779.256k$$

where $b = 2.9 * 10^{-3} m.k$ and $\lambda = 502nm$

Question 6.11: Overrelaxation Method :

(a) **Deriving Error formula for the overrelaxation method:**

We have the following relation:

$$x^* = x' + \epsilon'$$

Before measuring x', the error in measurement is:

$$x^* = x + \epsilon$$

We also have the equation for overrelaxation as:

$$x' = (1+w)f(x^*) - wx^*$$

Now we can expand the overrelaxation formula around x^* which will yield the following equation when we ignore higher order terms:

$$x' = (1+w)f(x^*) + (x - x^*)f'(x^*) - wx$$

Using the last two equations along with some manipulations, we can get the following result:

$$x' = (1+w)f(x^*) - wx^* - w(x - x^*) + (1+w)(x - x^*)f'(x^*)$$

Where we have also used the following relation to arrive at the previous formula :

$$x^* = (1+w)f(x^*) - wx^*$$

Therefore, one can conclude from the previous two formulas that :

$$x' - x^* = ((1+w)f'(x^*) - w) - w)(x - x^*)$$

The last result is of extreme importance as it connects ϵ' and ϵ' as follows:

$$\epsilon' = ((1+w)f'(x^*) - w)\epsilon$$

Using the error definition, we can write the following :

$$x^* = x' + \epsilon' = x + \epsilon = x + \frac{\epsilon'}{(1+w)f'(x^*) - w)}$$

Rearranging the previous equation, we can write it as follows:

$$x' - x = \frac{\epsilon'}{(1+w)f'(x^*) - w)} - \epsilon'$$

Lastly, we take ϵ' as a common factor and divide both sides by its coefficient to get the desired result:

$$\epsilon' = \frac{x - x'}{1 - 1/((1+w)f'(x) - w)}$$

(b) We are trying to solve the equation

$$x = 1 + e^{-cx}$$

Using the attached code, the number of iterations needed to converge to a solution accurate to 10^{-6} is 14 when c = 2

(c) Using the overrelaxation method to solve the same equation, my best convergence rate was after 4 iterations only using W = 0.75 which is more than twice as fast as the normal relaxation method we used in the previous part to evaluate the same problem.

(d) The only way I can think choosing a negative value of W would be helpful if the positive value of W wasn't giving any good results and wasn't going in the right direction of our minima. This could be the case if the code wasn't converging to a minimum value after a considerably large number of iterations.

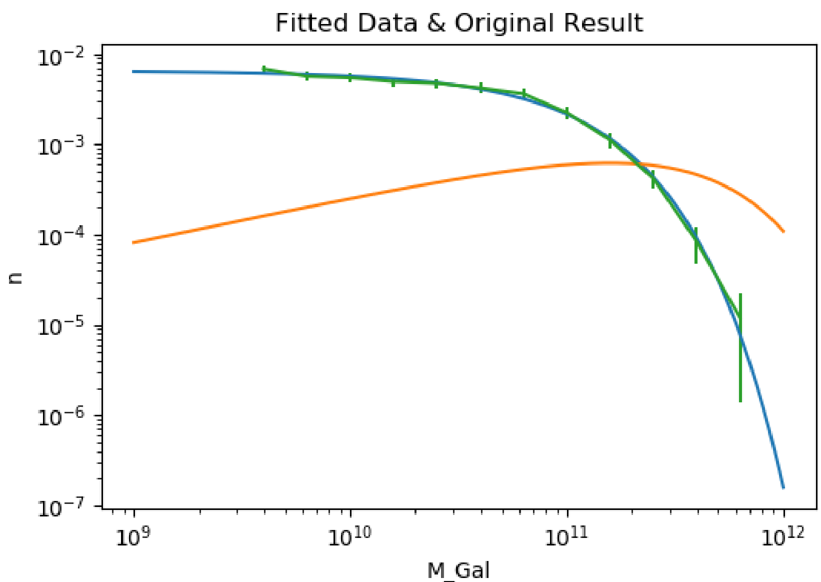
Question 3: Schechter function and Cosmology :

\left(\right)

Our Schechter function is defined as:

$$n(Mgal) = \phi^*(\frac{Mgal}{M_*})^{\alpha+1} e^{\frac{Mgal}{M_*}} ln(10)$$

Fitting the Schechter Function:

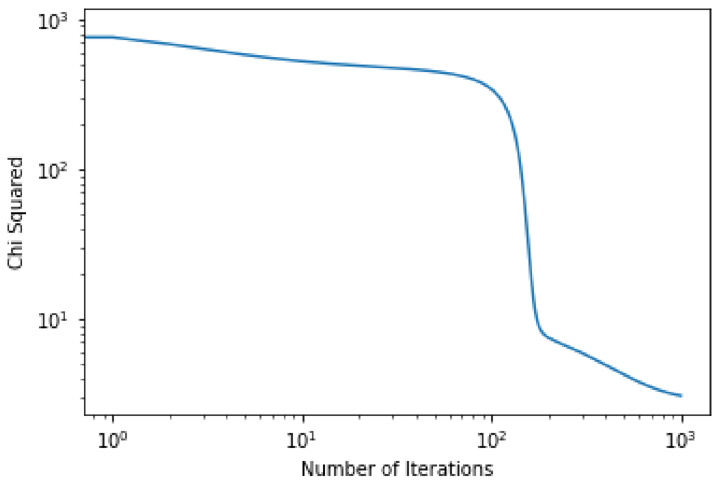


In this question, we used the gradient descend algorithm to find the minimum values of our three unknown parameters (M_*, ϕ and α) that would minimize χ^2 which should give a best fit for the provided data that has been measured. The optimal values that I got are:

$\phi = -2.59432915$
$\alpha = 1.03451465$
$M_* = 10.9928363$

The minimum value that I got for χ^2 is 2.995253755746826

I used these values afterwards and plugged it in the Schechter Function and obtained this graph for a larger range of M_gal. The orange line shows the Schechter function using random initial values other than the ones we get from the gradient descent method which is supposed to minimize the error represented by χ^2 and give us a good fit for the data, which is what happened. It is also worth mentioning that no matter what the initial values I put, it always converged to the same result (or slightly different) except when I put disastrous initial guesses which would make the code blow up at some points because it is dividing by zero or other errors that I couldn't fully comprehend.



This is a plot showing χ^2 as a function of the iterations which is an indication of how fast our function converged to its minimum value. In evaluating the derivatives in the gradient descent method, I used the extrapolated difference scheme as it was the most accurate one from the previous assignment. I've converged relatively faster than most of my colleagues I've compared my results with. I'm not sure if this is the only factor that made my algorithm converge faster. The ones I've compared my results with were using either the central difference or forward difference schemes.

interesting. not sure, but convergence also depends on γ and initial step size.