

# Computational Physics Homework 2

9.5/60

Michael Albergo

11 October 2019

## 1 Problem 1

Problem 6.11 from the Newman textbook. This problem wants us to study the rate of convergence for the over-relaxation method (much akin to the derivation of the relaxation method in the book).

a) We know from the book the

$$x' = (1 + \omega)f(x) - \omega x \quad (1)$$

Following the convention from the relaxation method derivation, we say that the error on the solution  $x^*$  for estimate  $x$  is  $\epsilon$  and the error for estimate  $x'$  is  $\epsilon'$ . We want to solve for  $\epsilon'$ .

Solving for  $f(x)$ :

$$f(x) = \frac{x' - x}{1 + \omega} \quad (2)$$

we Taylor expand  $f(x)$  around the solution:

$$f(x) = f(x^*) + f'(x^*)(x - x^*) \quad (3)$$

$$\approx x^* - \epsilon f'(x^*) \quad (4)$$

Plugging in and solving and multiplying by  $1 + \omega$  gives:

$$x' - x = \epsilon + \epsilon(\omega - (1 + \omega)f'(x^*)) \quad (5)$$

$$x - x' = -1 * (\epsilon + \epsilon(\omega - (1 + \omega)f'(x^*))) \quad (6)$$

So then we can divide by  $\epsilon((1 + \omega)f'(x^*) - w)$  such that

$$\frac{x - x'}{\epsilon((1 + \omega)f'(x^*) - w)} = 1 - \frac{1}{((1 + \omega)f'(x^*) - w)} \quad (7)$$

$$\frac{x - x'}{-x' + x + \epsilon} = 1 - \frac{1}{((1 + \omega)f'(x^*) - w)} \quad (8)$$

$$\frac{x - x'}{\epsilon'} = 1 - \frac{1}{((1 + \omega)f'(x^*) - w)} \quad (9)$$

$$\epsilon' = \frac{x - x'}{1 - \frac{1}{((1 + \omega)f'(x^*) - w)}} \quad (10)$$

$$\epsilon' \approx \frac{x - x'}{1 - \frac{1}{((1 + \omega)f'(x) - w)}} \quad (11)$$

for  $x$  near  $x^*$ , such that  $f'(x) \approx f'(x^*)$ .

b) Using the normal relaxation method, I see convergence to error  $\leq 10^{-6}$  in around 13 steps (See the jupyter notebook).

c) Using the over-relaxation method with  $\omega = 0.7$ , I see convergence to error  $\leq 10^{-6}$  in around 5 steps. (See the jupyter notebook). Moving around the  $\omega$  value generally changes the convergence to around 5-9 steps. This fits with the question's suggestion of improving to around at least half as many steps.

d) Yes, there are circumstances where under-relaxation would be beneficial. Because the approximation accuracy (and convergence) is dependent on the derivative of the function, the step you take could continuously oscillate around the solution if the derivative is large. So when  $f'(x)$  for  $x$  near  $x^*$  is very large, you might want to take smaller steps so as to not continuously overshoot.

## 2 Problem 2

$$I(\lambda) = \frac{2mhc^2\lambda^{-5}}{e^{hc/\lambda k_B T} - 1} \quad (12)$$

a) This question is basically asking us to find the critical points of  $I(\lambda)$ . This is done by taking the derivative of  $I$  with respect to  $\lambda$  and setting it equal to zero. Doing so should give us the equation they ask for.

$$\frac{\partial I}{\partial \lambda} = \frac{2\pi m h^2 c^3 \lambda^{-7} e^{\frac{hc}{\lambda k_B T}}}{k_B T (e^{\frac{hc}{\lambda k_B T}} - 1)^2} - \frac{10mhc^2\lambda^{-6}}{(e^{\frac{hc}{\lambda k_B T}} - 1)} = 0 \quad (13)$$

$$= \frac{2\pi m h^2 c^3 \lambda^{-7}}{k_B T} e^{\frac{hc}{\lambda k_B T}} - \frac{10mhc^2\lambda^{-6}}{(e^{\frac{hc}{\lambda k_B T}} - 1)} = 0 \quad (14)$$

$$\rightarrow \frac{hc}{k_B T \lambda} - 5(1 - e^{-\frac{hc}{k_B T \lambda}}) = 0 \quad (15)$$

$$\rightarrow 5e^{-\frac{hc}{k_B T \lambda}} + \frac{hc}{k_B T \lambda} - 5 = 0 \quad (16)$$

If we say  $x = \frac{hc}{k_B T \lambda}$ , then

$$5e^{-x} + x - 5 = 0 \quad (17)$$

as desired.

b) See corresponding jupyter notebook:  $b = 0.0028978$ .

c) See corresponding jupyter notebook:  $T = \frac{b}{\lambda}$  and for  $\lambda = 502 \times 10^{-9}$  meters,  $T \approx 5772$  Kelvin.

## 3 Problem 3

a) **Implement multivariate gradient descent and demonstrate that it works.**

First, here is evidence that the gradient descent algorithm implemented in the jupyter notebook functions properly. Shown in Figure 1 is the convergence of the parameters  $(x, y)$  for minimizing  $f(x, y) = (x - 2)^2 + (y - 2)^2$ . The implementation quickly finds the optimal parameters  $(x^*, y^*)$  that minimize the function (2 and 2, respectively).

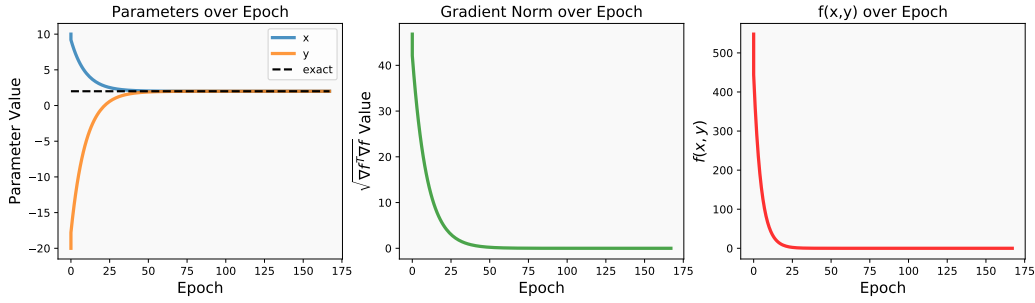


Figure 1: Convergence of parameters while minimizing the test function  $f$ .

b) **Show the Schechter function best fit and the chi-square convergence.**

This approach is then applied to the Schechter function data. The new goal is to minimize the  $\chi^2$  of:

$$n(M_{gal}) = \phi^* \left( \frac{M_{gal}}{M_*} \right)^{\alpha+1} e^{\frac{M_{gal}}{M_*}} \ln 10 \quad (18)$$

Some of the data provided was exponentiated (originally given in log base 10 form). In Figure 2 the

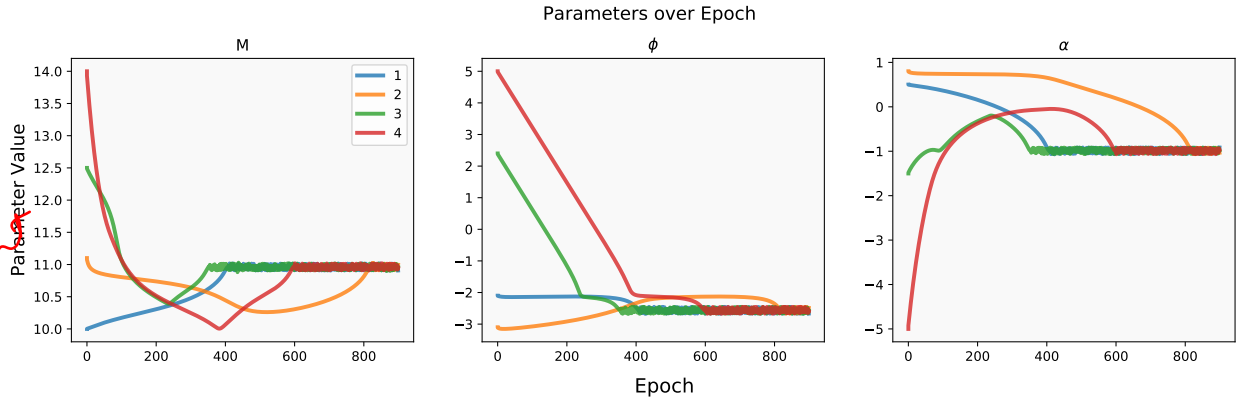


Figure 2: Convergence of parameters while minimizing the test function  $n(M_{gal})$ .

optimal values of  $M$ ,  $\phi$ , and  $\alpha$  were found via gradient descent for 4 different trials, differing by parameter initialization. The initialization of the parameters governed the speed for which convergence was reached. These values can be found in the jupyter notebook, and were chosen to be reasonable enough that the exponents they helped define were not unstable. Plots of the  $\log \chi^2(n)$  over epoch are shown in Figure 3, for which it can be seen that the  $\chi^2(n)$  value goes to and oscillates around 0. Using the converged parameters, a best fit for the Schechter was constructed for each trial. These are illustrated compared to the real data provided for the homework in Figure 4.

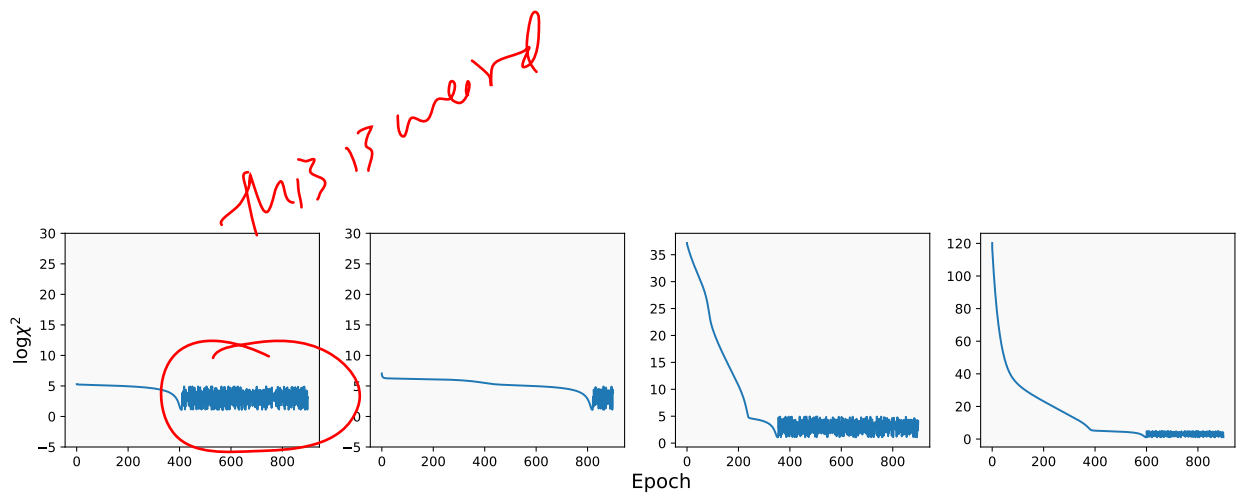


Figure 3: Minimization of  $\chi^2(n)$  for the four different trials.

-1/2

not actually  
converging.  
are you  
fixing h in  
your  
derivatives?  
steps too big  
relative to  
minimum.

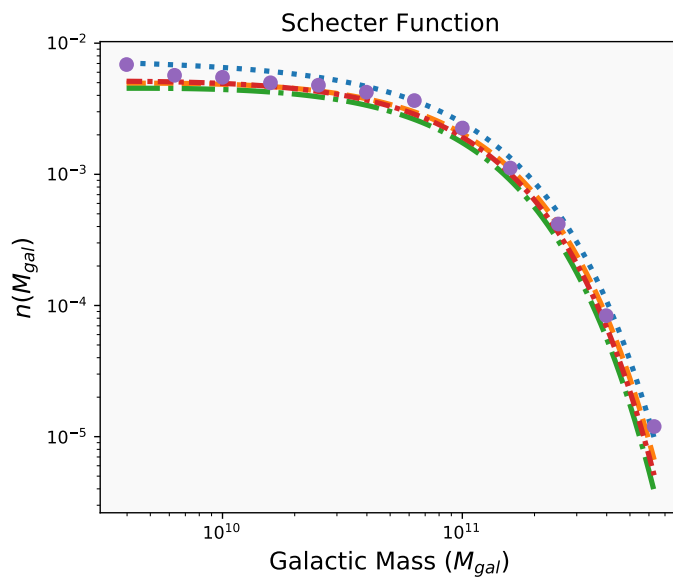


Figure 4: Schechter function best fit for the four different trials. Actual data shown as purple dots.