10/10

Computational Physics
Homework 2

Matthew Drnevich

1. (a) Let ϵ be the error on our current estimate of the solution to the equation, i.e. $x^* = x + \epsilon$. Then let ϵ' be the error on the next estimate such that $x^* = x' + \epsilon'$. Then we have that

$$\begin{aligned}
x' &= (1+\omega)f(x) - \omega x \\
\implies \frac{x'-x}{1+\omega} + x &= f(x) \\
&= f(x^*) + (x - x^*)f'(x^*) + \ldots \\
&\approx x^* - \epsilon f'(x^*) \\
\implies \frac{x'-x}{1+\omega} &= \epsilon - \epsilon f'(x^*) \\
\implies x' - x &= \epsilon + \epsilon(\omega - (1+\omega)f'(x^*)) \\
\implies x - x' &= \epsilon((1+\omega)f'(x^*) - \omega) - \epsilon \\
\implies \frac{x-x'}{\epsilon((1+\omega)f'(x^*)-\omega)} &= 1 - 1/[(1+\omega)f'(x^*) - \omega] \\
\implies \frac{x-x'}{-(x'-x-\epsilon)} &= 1 - 1/[(1+\omega)f'(x^*) - \omega] \\
\implies \frac{x-x'}{\epsilon'} &= 1 - 1/[(1+\omega)f'(x^*) - \omega] \\
\implies \epsilon' &= \frac{x-x'}{1 - 1/[(1+\omega)f'(x^*) - \omega]} \\
\implies \epsilon' &\approx \frac{x-x'}{1 - 1/[(1+\omega)f'(x) - \omega]}
\end{aligned}$$

✓

(b) Using normal relaxation we get a solution with less than 10^{-6} error in 14 iterations.

(c) Using overrelaxation with $\omega = 0.37$ we get a solution with less than 10^{-6} error in 7 iterations. If we increase to $\omega = 0.38$ this converges in 1 iteration.

(d) Yes, having $\omega < 0$ is usual when you wish to decrease the amount that you use the derivative to take steps, which is particularly usual when you have a solution x^* where $f'(x)$ is very large nearby. this could cause your method to oscillate around the solution unless you use $\omega < 0$ to reduce this effect by using less of the values of $f(x)$. ✓

2. (a) We have that $I(\lambda) = (2\pi hc^2\lambda^{-5})(e^{hc/\lambda k_B T} - 1)^{-1}$ so product rule tells us that to

maximize this we need

$$
\begin{aligned}
\frac{\partial I}{\partial \lambda} = (-10\pi hc^2\lambda^{-6})(e^{hc/\lambda k_BT}-1)^{-1} + (2\pi h^2c^3\lambda^{-7}/k_BT)e^{hc/\lambda k_BT}(e^{hc/\lambda k_BT}-1)^{-2} &= 0 \\
\implies (-10\pi hc^2\lambda^{-6})(e^{hc/\lambda k_BT}-1) + (2\pi h^2c^3\lambda^{-7}/k_BT)e^{hc/\lambda k_BT} &= 0 \\
\implies -5(e^{hc/\lambda k_BT}-1) + (hc/\lambda k_BT)e^{hc/\lambda k_BT} &= 0 \\
\implies -5(1-e^{-hc/\lambda k_BT}) + \frac{hc}{\lambda k_BT} &= 0 \\
\implies 5e^{-hc/\lambda k_BT} + \frac{hc}{\lambda k_BT} - 5 &= 0
\end{aligned}
$$

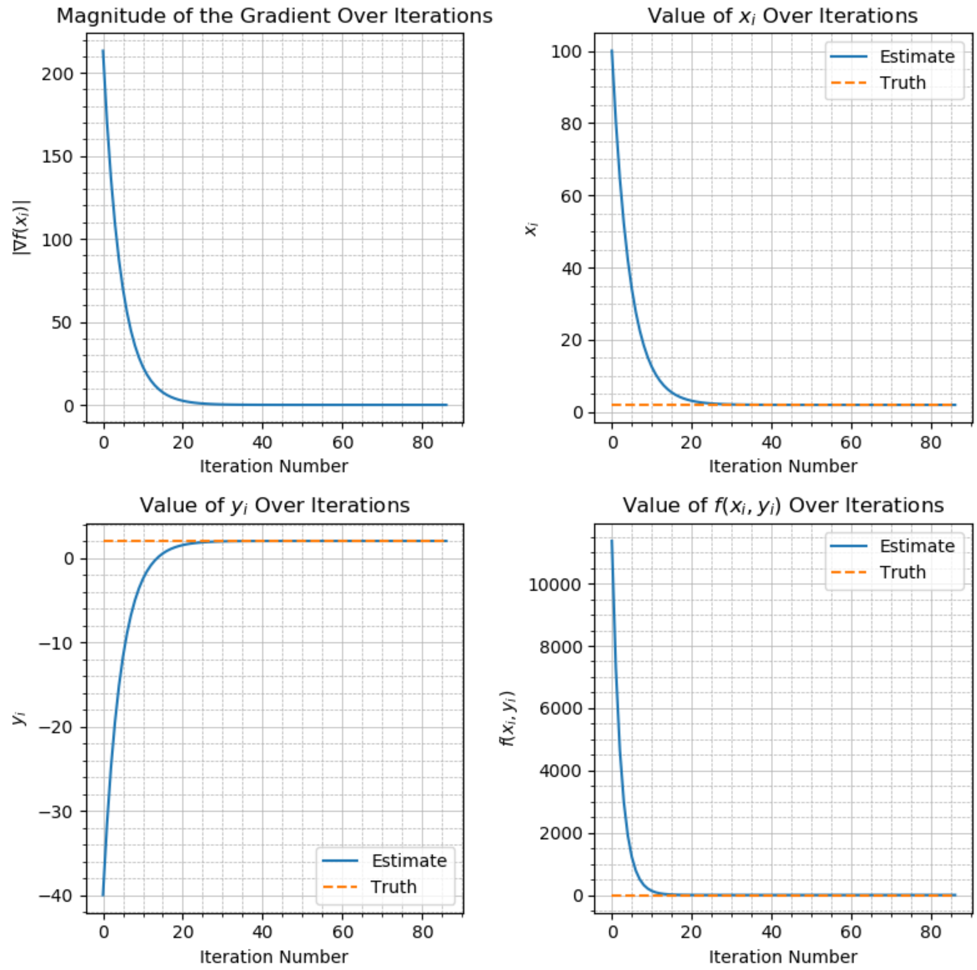
Thus if $x = hc/\lambda k_BT$ then $\lambda = hc/xk_BT = b/T$ where $b = hc/k_Bx$ and we now need to solve the equation

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

(b) Using our solution from the Python code, we get a value of $b = 2.897769 \times 10^{-3}$.

(c) This is simply that $T = b/\lambda = 5770$ Kelvin.

3. The code demonstrating the solution to this problem can be viewed in my Github repo for Homework 2 in the Jupyter notebook 'hw2.ipynb'. The first plot here demonstrates the success of my gradient descent code for the trial function $f(x, y) = (x-2)^2 + (y-2)^2$.



good

Figure 1: Evolution of parameters by gradient descent for the trial function.

Next, when using gradient descent to determine the Schechter function, I used gradient descent on the log of the χ^2 function to determine the parameters $\log(M_*)$, $\log(\phi^*)$, and α. My parameters for gradient descent were a learning rate of $\gamma = 3 \times 10^{-3}$, an error threshold of $\epsilon_{error} = 10^{-6}$, and a derivative step size for the extrapolated difference method of $h = 0.1$. In order to show robustness, three trials were conducted, each with considerably different initializations of the parameters, which were

Trial 1: $\log(M_*) = 11.5$, $\log(\phi^*) = -3.2$, and $\alpha = -0.5$

Trial 2: $\log(M_*) = 12.5$, $\log(\phi^*) = -1$, and $\alpha = -4$

Trial 3: $\log(M_*) = 10.5$, $\log(\phi^*) = -0.5$, and $\alpha = 1.5$

below you can see how these initial parameters evolved over the course of gradient descent.

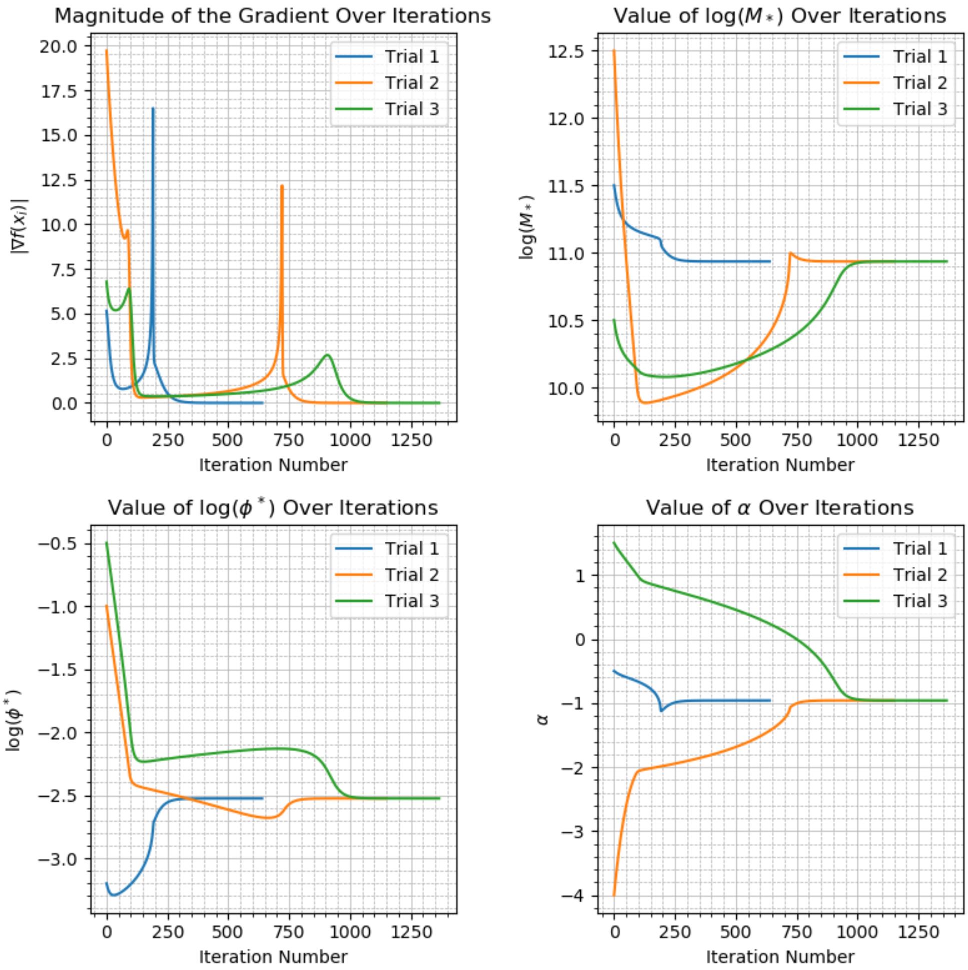


Figure 2: Evolution of parameters by gradient descent on the log of χ^2 of the Schechter function.

Finally, you can see that all three of these trials converged to the same solution, just at different rates, which is visible in the following graphs.

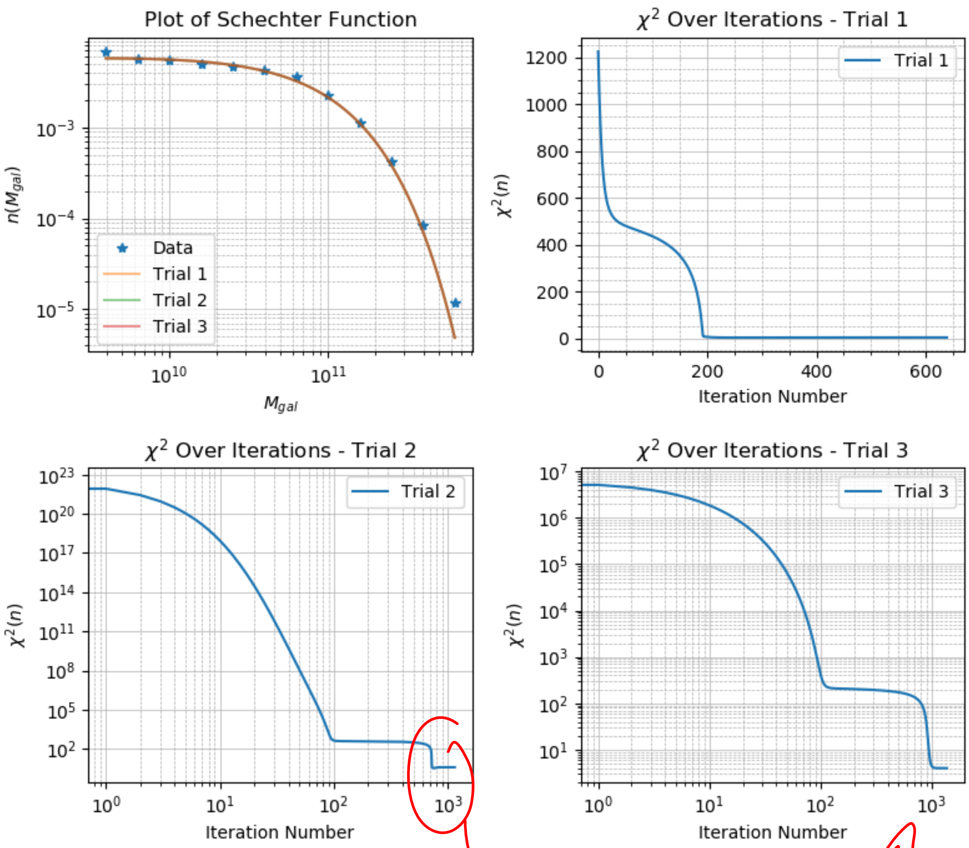


Figure 3: Evolution of parameters by gradient descent on the log of χ^2 of the Schechter function.

Oh, I was confused at first but now I see different y-scalings. Very robust!