HW2 Computational Physics

10/10

Isaac Sarnoff

October 2019

1

1 Introduction

This week we were learning about numerical and computational methods for finding the solutions to or minimizing functions. We first looked at using an overrelaxation method. Then we looked at binary search and applied it to Wein's displacement law. Finally, we used the method of gradient descent to find the solutions to the line of best fit for the Schechter function by minimizing its χ^2 value with respect to the free parameters.

2 Problem 1: Overrelaxation

For this problem, we wanted to see the difference between the effectiveness of the overrelaxation method in comparison with the effectiveness of the simple relaxation method.We started by finding the error in the overrelaxation method: $x^* = x' + \epsilon'$. we also know that $x' = (1+w)f(x) - wx$ this allows us to taylor expand and get $x^* + \epsilon' = (1+w)(f(x^* + \epsilon) + \epsilon f'(x^*)) - wx$ after a bunch of algebra, we find that

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

After figuring this out, we created a method for both the overrelaxation and regular relaxation methods and used them on the function $x = 1 - e^{-2x}$. This showed that at least in some cases, overrelaxation can greatly increase computational efficiency. Finally, we asked the question of would it ever be appropriate to use an factor of w less than 0. It would. and it would be incredibly helpful to do so if a function is changing extremely rapidly. In these cases, the less than 0 w could act as a dampener that would reduce oscillation around the solution.

better to make plots.

```
1.0
0.86466473
0.82259667
0.8070247
0.8009202
0.79847467
0.7974866
0.797086
0.7969234
0.79685736
0.79683053
0.7968196
0.79681516
0.79681337
0.79681265
0.79681236
0.79681224
0.7968122
0.7968122
converged
```

(a)

```
1
0.8646647167633873
0.7968872733338224
0.7968203558771076
0.7968130311893629
0.7968122287543633
converged
```

(b)

what was value of w? How did you choose it?

Figure 1: List of values see by the relaxation, (a), and overrelaxation, (b) methods as they converge on the given function. Both start at 1 and converge relatively quickly, however, the overrelaxation method, with a learning factor of 0.5 coonverges more than twice as fast.

3 Problem 2: Binary Search

In this problem we are asked to look into the case of Wein's Displacement constant, λ. We start by taking the known form of the equation for intensity of black body radiation and taking the derivative of that equation:

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

Taking the derivative, we find the maximum at:

$$0 = 2\pi hc^2[\frac{hce^{hc/\lambda k_B T}}{\lambda^7 k_B T^7 (e^{hc/\lambda k_B T}-1)^2} + \frac{5}{\lambda^6(e^{hc/\lambda k_B T}-1)}]$$

which we can simplify down to:

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

then, by insisting that $x = hc/\lambda k_B T$: we arrive at a very simple equation of form:

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

Rearanging the equationf for x in terms of λ so that it gives λ in terms of x, we get:

$$\lambda = hc/k_B T x := b/T$$

We now formulate a binary search method that finds the value of x to a tolerance of 10^{-6} by iterating over possible values of x and removing half of the search area at every step.

```
50.0
25.0
12.5
6.25
3.125
4.6875
5.46875
5.078125
4.8828125
4.98046875
4.931640625
4.9560546875
4.96826171875
4.962158203125
4.9652099609375
4.96368408203125
4.964447021484375
4.9648284912109375
4.965019226074219
4.965114593505859
4.965066909790039
4.965090751647949
4.965102672576904
4.965108633041382
4.965111613273621
4.96511310338974
4.9651138484478
4.9651142209768295
```

plot please

Figure 2: List of middle x values checked by a binary search algorithm searching for solutions between 0 and 100. Note the rapid rate of convergence.

Finally, we apply our work to find the temperature of the sun given that it has maximum radiation intensity at a wavelength, $\lambda = 502nm$.

$$502nm = hc/k_b T(4.9651)$$

$$T = 5760K$$

This result is only 18K below the accepted result, 0.3% off.

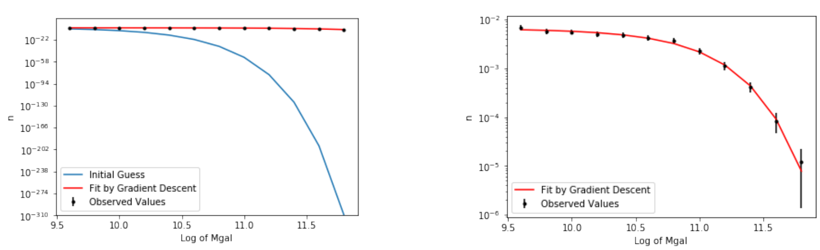
4 Problem 3: Gradient Descent

For the final problem, we used the gradient descent method to find a best fit line for the Schechter function. This took a great deal of fine tuning and we ended up using the analytical central derivative method with fixed h values because we kept getting infinite growth and division by zero. Another problem that occurred frequently was that that the there appears to be a local minimum in the parameter space corresponding to a $\chi^2 = 607$ and it was very hard to keep the gradient descent from falling into that local minimum. The last problem that we faced was that phi has to be in log space or else, what would be a small step size can end up being large enough to throw the function wildly away from where it was searching. However, once all of those problems were solved, we found that we were able to fit the Schechter function quite well and found a $\chi^2 = 2.9$. This is somewhat lower than we had expected, as we have 12 points and only 3 parameters, but it does make sense when you consider that the parameters are actually correlated with one another. Finally, we looked at our

interesting. you should show this fit.

χ

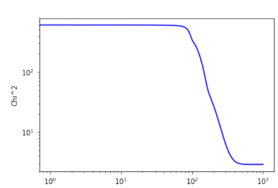
data points



(a) (b)

Figure 3: Plots of the Schechter function and the observed values. Plot a contains the initial fitting with random parameters given to the Schechter function as well as the fitting after applying gradient descent. Plot b is solely after gradient descent. Notice that the initial values are off by several hundred orders of magnitude at points.

values of χ^2 as a function of iteration to make sure that we were indeed seeing statistically meaningful convergences.



too Small. I cannot read this. Bar old.

Figure 4: χ^2 as a function of iteration. It changes very slowly in the beginning as it has a low gamma value.