Homework 2

20/10

Stefan Waterval

October 11, 2019

1 Problem 1

1.1 Part b

✓

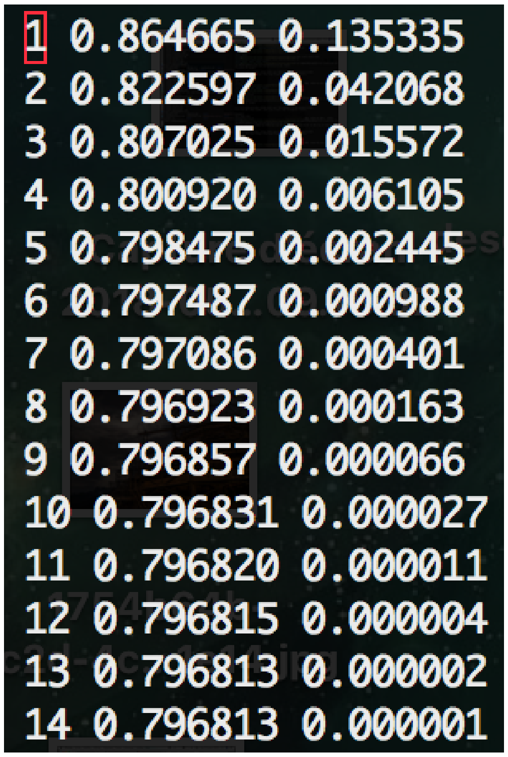
We are asked to solve the equation

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

to an accuracy of 10^{-6} using the relaxation method. In eq 1, x is the unknown and c is set to 2.
The relaxation method is a simple method where we iterate over the equation and set our new variable x_{new} as $f(x)$. For a given starting value (here set to 1), the relaxation method can be expressed as follows:

$$x_{\text{new}} = f(x). \tag{2}$$

This is then repeated as long as $|f(x) - x| > 10^{-6}$. The code can be found in `ex1_hw2_stefan_waterval.c` and the output is displayed in `ex1_611b.dat`. The columns are the iteration i, $f(x_i)$, and $|f(x_i) - x_i|$. ✓



```
1 0.864665 0.135335
2 0.822597 0.042068
3 0.807025 0.015572
4 0.800920 0.006105
5 0.798475 0.002445
6 0.797487 0.000988
7 0.797086 0.000401
8 0.796923 0.000163
9 0.796857 0.000066
10 0.796831 0.000027
11 0.796820 0.000011
12 0.796815 0.000004
13 0.796813 0.000002
14 0.796813 0.000001
```

better to show as plot

Figure 1: Results found in `ex1_611b.dat`.

From fig 1, we see that the relaxation method hits convergence after 14 iterations.

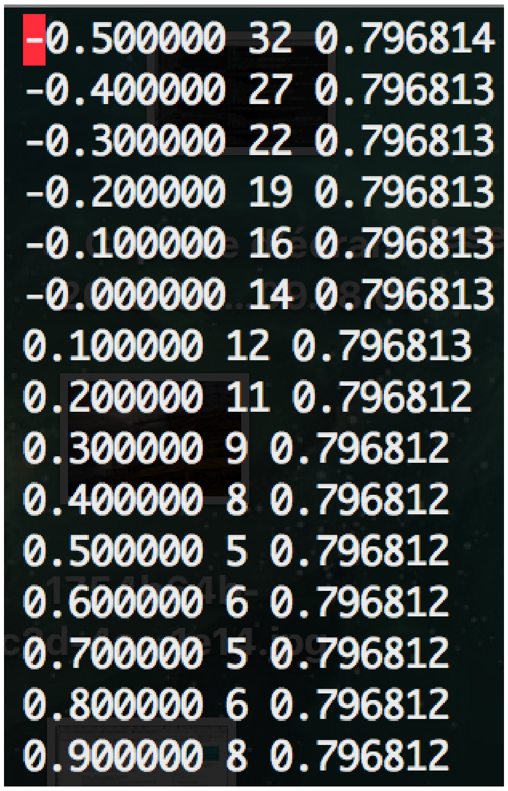
1.2 Part c

Here we solve again Eq 1 for x, setting $c = 2$ but we improve the relaxation method by introducing a new parameter ω. Formally, this is implemented as follows:

$$x_{\text{new}} = (1 + \omega) f(x) - \omega x. \tag{3}$$

We test the covergence rate for different values of ω ranging from -0.5 to 0.9 with 0.1 increments. The code is located in `ex1_hw2_stefan_waterval.c` and the output results can be found in `ex1_611c.dat`. The columns are ω, the number of iterations and the result.

nice. but also better in plot form



```
-0.500000 32 0.796814
-0.400000 27 0.796813
-0.300000 22 0.796813
-0.200000 19 0.796813
-0.100000 16 0.796813
-0.000000 14 0.796813
0.100000 12 0.796813
0.200000 11 0.796812
0.300000 9 0.796812
0.400000 8 0.796812
0.500000 5 0.796812
0.600000 6 0.796812
0.700000 5 0.796812
0.800000 6 0.796812
0.900000 8 0.796812
```

Figure 2: Results found in `ex1_611c.dat`.

Fig 2 indicates that the quickest covergence happens for $\omega = 0.5$ and $\omega = 0.7$. When $\omega = 0$, we retrieve the relaxation method and the convergence rate is hence the same as subsection 1.1.

1.3 Part d

Following the argument found in *Computational Physics* from Mark Newman, if we start say below the solution of the equation (let us assume the one-dimensional case), we can overshoot the steps with the parameter ω and hope to get to the solution faster. This assumes however that we start in the right direction (in the book they mention starting at $x_0 = 1$ and targetting $x_{\text{sol}} = 5$, bypassing the intermediate $x_1 = 3$ and going straight to $x_1 = 4$). A negative ω could then come handy if we start going in the opposite direction and then actually getting further away of x_{sol} in the first steps.

2 Problem 2

We are given the following formula for the intensity of radiation per unit area and per unit wavelength λ from a black body at temperature T:

$$I(\lambda) = \frac{2\pi hc^2\lambda^{-5}}{e^{hc/\lambda k_B T} - 1}. \tag{4}$$

2.1 Part a

The maximum radiation is optained by differentiating $I(\lambda)$ w.r.t. λ and set it equal to zero:

$$\begin{aligned} I'(\lambda) &= \frac{-10\pi hc^2\lambda^{-6}[e^{hc/\lambda k_B T} - 1] - 2\pi hc^2\lambda^{-5}e^{hc/\lambda k_B T}(-\frac{hc}{\lambda^2 k_B T})}{(e^{hc/\lambda k_B T} - 1)^2} = 0 \\ &\Rightarrow -\frac{10\pi hc^2}{\lambda^6}e^{hc/\lambda k_B T} + \frac{10\pi hc^2}{\lambda^6} + \frac{2\pi h^2c^3}{\lambda^7 k_B T}e^{hc/\lambda k_B T} = 0 \\ &\Rightarrow -5e^{hc/\lambda k_B T} + 5 + \frac{hc}{\lambda k_B T}e^{hc/\lambda k_B T} = 0 \\ &\Rightarrow 5e^{-hc/\lambda k_B T} + \frac{hc}{\lambda k_B T} - 5 = 0. \end{aligned}$$

Setting $x = \frac{hc}{\lambda k_B T}$ leads to:

$$5e^{-x} + x - 5 = 0. \tag{5}$$

✓

This equation has a trivial solution $x = 0$ but a quick look at x shows that it would mean $\lambda \to \infty$ or $T \to \infty$, which is not physical. Thus, the non trivial solution of this equation will give the maximum of $I(\lambda)$ for λ as follows:

$$\lambda = \frac{b}{T}, \tag{6}$$

where $b = hc/k_B x$ is called the *Wien displacement constant*. Eq. 6 is known as *Wien displacement law.*

2.2 Part b

We are asked to solve eq. 5 to an accuracy of $\epsilon = 10^{-6}$ using the binary search method and find the value of b. The code can be found in `ex2_hw2_stefan_waterval.c`. The starting values for x_1 and x_2 for the binary search were chosen to be 1 and 10. Running the code with these initial conditions, one finds $x = 4.965115$ and $b = 2898$ µm K. This value of b is consistent with what is found in the literature.

✓

2.3 Part c

With the results of subsection 2.2, we can now estimate the temperature of the Sun with the given information that the wavelength peak in the Sun's emitted radiation falls at $\lambda = 502$ nm. Using Wien displacement law, we get:

✓

$$T_{\text{Sun}} = \frac{b}{\lambda} = \frac{2898\ \mu\text{m K}}{502\ \text{nm}} = 5572\ \text{K}. \tag{7}$$

This temperature is a good estimation of the actual temperature of 5778 K.

3 Problem 3

In this problem, we are asked to fit the *Schechter function* from provided measured data. The code is available `ex3_hw2_stefan_waterval.py`. The Schechter function in our case is expressed as follows:

$$n(M_{\text{gal}}) = \phi^* \left(\frac{M_{\text{gal}}}{M_*}\right)^{\alpha+1} \exp\left(-\frac{M_{\text{gal}}}{M_*}\right) \ln(10). \tag{8}$$

In eq. 8 above, the three free parameters that we are trying to find are the amplitude ϕ^*, the *characteristic mass scale* M_*, where the function goes from a power-law to and exponential cutoff, and α, the low-mass slope of the function.
The optimal free parameters are obtained by minimizing the χ^2 function using a *gradient descent* method in three dimensions. The gradient descent method is implemented in the following way:

$$\begin{aligned}
a_{i+1} &= a_i - \gamma \frac{f(a_i, b_{i-1}, c_{i-1}) - f(a_{i-1}, b_{i-1}, c_{i-1})}{a_i - a_{i-1}} \\
b_{i+1} &= b_i - \gamma \frac{f(a_{i-1}, b_i, c_{i-1}) - f(a_{i-1}, b_{i-1}, c_{i-1})}{b_i - b_{i-1}} \\
c_{i+1} &= c_i - \gamma \frac{f(a_{i-1}, b_{i-1}, c_i) - f(a_{i-1}, b_{i-1}, c_{i-1})}{c_i - c_{i-1}}
\end{aligned} \tag{9}$$

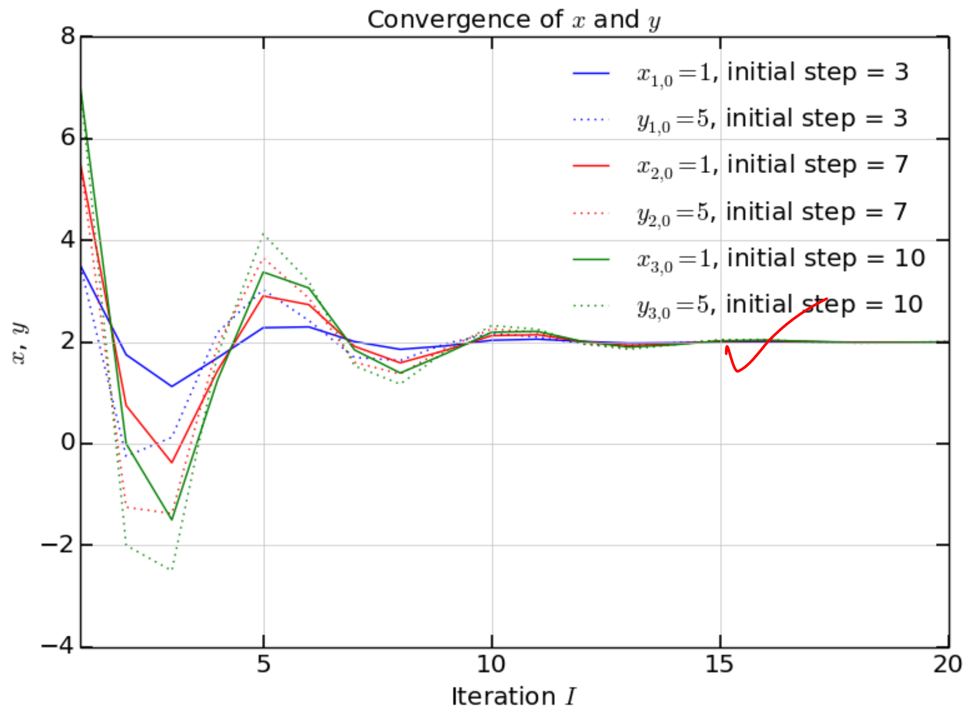
The parameter γ is set to 0.0001. We start with initial values for ϕ^*, α, and M_* (which in eq. 9 correspond to $a_{i-1} = a_0$, $b_{i-1} = b_0$, and $c_{i-1} = c_0$) and we add the same initial step to get a_1, b_1, and c_1. From there on, we can iterate the algorithm until the difference between the parameters in the i-th iteration are within 10^{-6} from the previous ones.
As mentioned before, the function we are trying to minimize (f in eq. 9) is the χ^2 which is given by:

$$\chi^2 = \sum_{i=1}^{N} \frac{(n_{\mathrm{fit},i} - n_i)^2}{\sigma_i^2}. \tag{10}$$

Here we sum over the number of measured data points provided. In other words, for every set of three free parameters, we compute the Schechter function at every point M_{gal} and calculate χ^2 using eq. 10, where n_i and σ_i are the provided n's and error on them, respectively.
In order to try our gradient descent algorithm, we first test a two-dimensional version on a simple function where the minimum is known. The function is:

$$f(x, y) = (x - 2)^2 + (y - 2)^2, \tag{11}$$

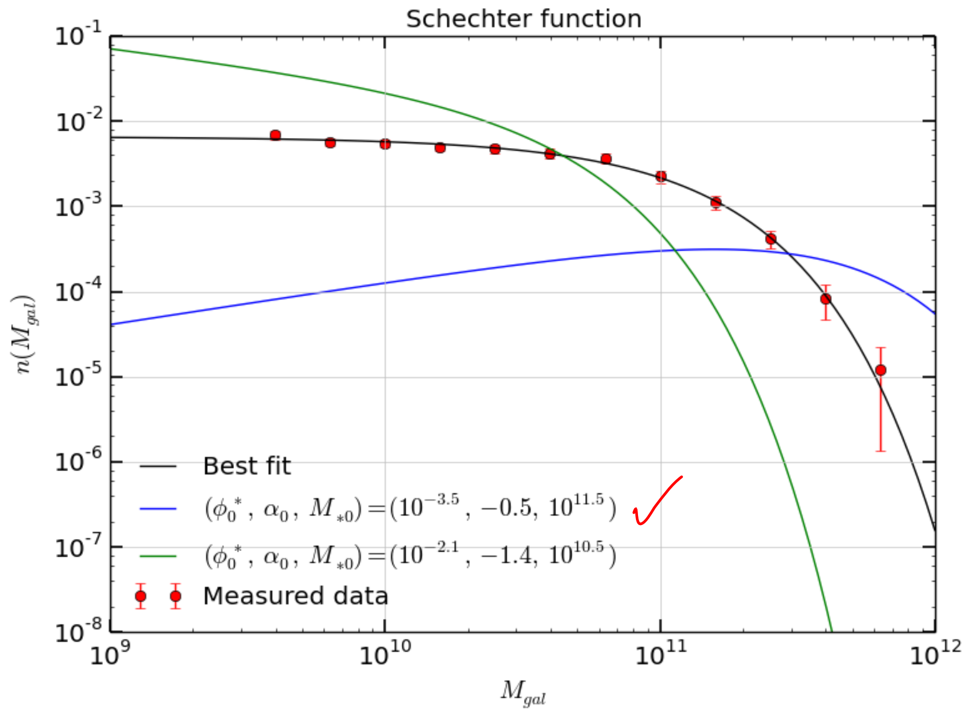
where the minimum of the function obviously is at $x = y = 2$.
The robustess of the code was tested using the same initial values $x_0 = 1$ and $y_0 = 5$ with three different initial steps 3, 7, and 10. The convergence to $x = 2$ and $y = 2$ as a function of the number of iterations is shown in fig. 3 below. The value of γ used here is 0.5.



nice.

Figure 3: Convergence of the gradient descent in two dimensions for three different initial step sizes.

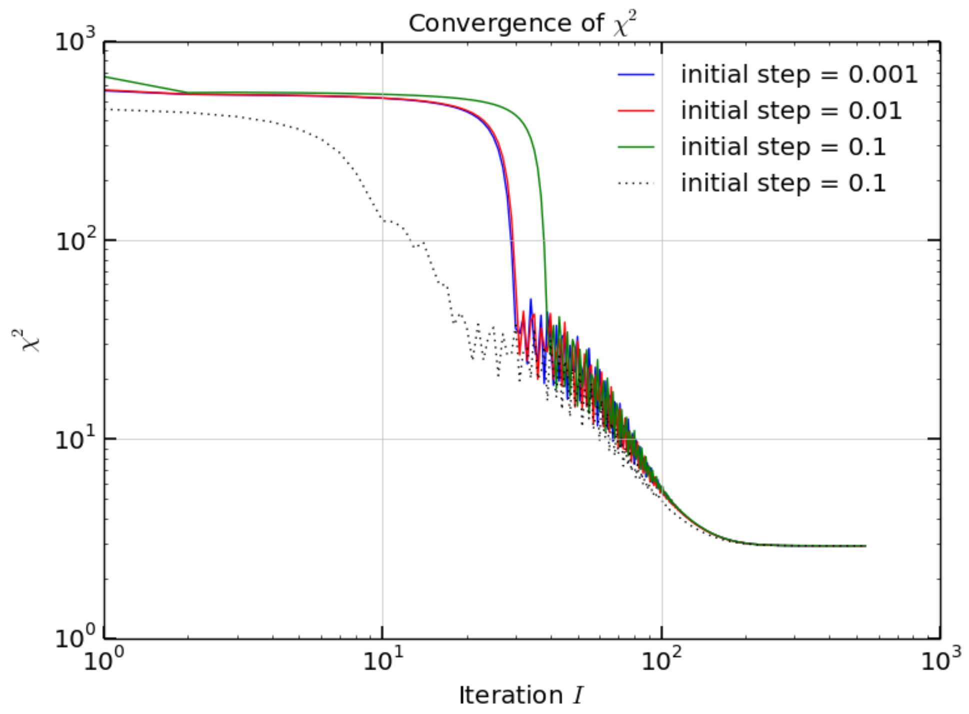
Now that our gradient descent algorithm showed to be robust, we can minimize χ^2 to find our optimal free parameters. Note that the three-dimensional gradient descent will be applied on a base 10 logspace. This means that the initial free parameters we feed in an the output will be exponents. We just have to raise 10 to the respective exponents for ϕ^* and M_* to get the appropriate parameters.
We begin with two sets of starting points $(\log\phi^*, \alpha, \log M_*)$ = (-3.5, -0.5, 11.5) and (-2.1, -1.4, 10.5). The optimal free parameters found are $(\log\phi^*, \alpha, \log M_*) \approx$ (-2.568, -1.009, 10.976) with a corresponding $\chi^2 \approx 2.908$.



nice .

Figure 4: Schechter function for the two initial free parameter sets (blue and green) as well as the best fit in black. The provided measured data are shown as red dots.

Fig. 4 shows the results obtained after running the gradient descent method. The blue and green lines are the Schechter function computed using the initial set of free parameters mentioned earlier. In black is the Schechter function plotted for the optimal values obtained from the gradient descent algorithm. The curve lies well within the measured data displayed in red.
Finally, let us have a look at the convergence of χ^2 as a function of the number of iterations i for the initial free parameter set $(\log \phi^*, \alpha, \log M_*)$ = (-3.5, -0.5, 11.5). For this test, we use the same initial values but change the initial step. We compute the χ^2 for initial step 0.001, 0.01, and 0.1. After looking at the results, it was decided to add a fourth curve (dotted black), which has initial step 0.1 but starting values $(\log \phi^*, \alpha, \log M_*)$ = (-3.5, -1.5, 10.5) to help understand the behaviour of the convergence.



very good.

Figure 5: Convergence of χ^2 for initial step size 0.001, 0.01, and 0.1 with same starting values (blue, red, green) and for a different set of initial parameters with step size 0.1 (black dotted).

We observe that all four curves hit convergence at $i \approx 200$. The main difference happens for both curve with initial step size 0.1 where χ^2 starts decreasing earlier or later. Intuitively, one could assume that a bigger step size should at least make χ^2 start decreasing earlier but this is also dependant on the initial free parameters values. Indeed, comparing the optimal values with the initial ones, we see that in the black dotted case, all our starting values are smaller than the optimal ones. Increasing the initial step therefore shifts the parameters closer to the optimal ones in the first step. On the other hand, the green curve has two initial values bigger than the optimal one, thus making a bigger step size less efficient at the beginning.