Homework 2 Computational Physics

Giacomo Recine*

7 October 2025

Contents

*gr2640@nyu.edu

1 Introduction

In this homework, we will first explore how to solve numerically non-linear equations in one variable, meaning equations of the form $x = f(x)$ within a chosen accuracy. To illustrate this, we will solve Exercise 6.11 from the Neumann book, as described in Sec. 2. In particular, we will use two simple but widely used methods: the relaxation and overrelaxation methods. Then, in Sec. 3, we will present a physical application of solving non-linear equations, specifically to determine the so-called Wien's displacement constant. Here, we will first use the bisection (binary) method, since instead of solving $x = f(x)$ directly, we can find the root of the equivalent equation $x - f(x) = 0$. From this, we will infer the temperature of the Sun. Finally, in Sec. 4, we will explore the gradient descent method. We will first test the method on a simple function to verify that our implementation works correctly, and then apply it to a physical problem: fitting the measurements of the galaxy stellar mass function from the COSMOS survey, provided in a data file. In this case, we will minimize the χ^2 function to determine the free parameters that best fit the data, using the gradient descent method. All results will be obtained with an accuracy of 10^{-6}.

2 Exercise 6.11 from Newman

2.1 A brief review: relaxation and over-relaxation methods

In this section we implement the code for the relaxation and over-relaxation methods. Let us briefly recall the relaxation method. In general, if we want to solve a nonlinear equation of the form

$$x = f(x), \tag{1}$$

we can proceed iteratively as follows: starting from an initial guess x_0, we compute $x_1 = f(x_0)$, then $x_2 = f(x_1)$, and so on. By repeating this process, the sequence $\{x_n\}$ may, if we are lucky, converge to a fixed point of $f(x)$, which corresponds to the desired solution. This is the so called relaxation method.

Of course, this simple method can present some difficulties. For instance, if the equation admits more than one solution, the relaxation method may converge to only one of them, providing no information about the others. Moreover, there exist certain solutions that cannot be reached by this method regardless of the chosen initial value. Even when starting from a point close to such a solution, the iteration will fail to converge. Nevertheless, relaxation is often a fast and convenient way to obtain solutions to nonlinear equations. In some cases, the convergence can be further accelerated by employing the technique of over-relaxation. In the over-relaxation method, the idea is to approach the true solution more quickly. It can be shown that the iteration can be written as

$$x' = (1 + \omega)f(x) - \omega x, \tag{2}$$

which is the form most commonly used. For the method to be effective, the relaxation parameter ω must be chosen carefully. There is generally some flexibility: while an optimal value exists, nearby values often provide similarly good convergence. Unfortunately, no general theory prescribes the optimal choice of ω, and in practice it is usually determined by trial and error.

2.2 Solution of the exercise

2.2.1 Solution question a

We begin by deriving the expression for the error in the relaxation method. It can be shown that, after one iteration, the error is approximately

$$\epsilon' \simeq \frac{x - x'}{1 - \dfrac{1}{(1 + \omega)f'(x) - \omega}}. \tag{3}$$

Indeed, starting from

$$x' = (1 + \omega)f(x) - \omega x, \tag{4}$$

Let us choose x close to the true solution x^*. Expanding $f(x)$ around x^* we obtain

$$x' = (1+\omega)\big(f(x^*) + (x - x^*)f'(x^*) + \dots\big) - \omega x, \tag{5}$$

since $x^* = f(x^*)$, this reduces to

$$x' = (1+\omega)\big(x^* + (x - x^*)f'(x^*) + \dots\big) - \omega x, \tag{6}$$

hence

$$x' - x^* = (x - x^*)\big((1+\omega)f'(x^*) - \omega\big). \tag{7}$$

Defining the errors $\epsilon = x - x^*$ and $\epsilon' = x' - x^*$, we find

$$\epsilon' = \epsilon\big((1+\omega)f'(x^*) - \omega\big), \tag{8}$$

now, from the definition of the error we can write

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

Since $x^* = x' + \epsilon'$, we obtain

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

rearranging, and since x^* is closed to x, gives

$$\epsilon' \simeq \frac{x - x'}{1 - \dfrac{1}{(1+\omega)f'(x) - \omega}}, \tag{11}$$

which is the desired result for the error in the overelaxation method.

2.2.2 Solution question b

We now implement the relaxation method, as reviewed above, to solve the following nonlinear equation:

$$x = 1 - e^{-2x}. \tag{12}$$

After 10 iterations, choosing as starting point $x = 1$, we have the solution $x = 0.79683050$. In particular, we are interested in determining the number of iterations required for the accuracy to fall below a chosen tolerance of 10^{-6}. Using the error estimate given by

$$\epsilon' \simeq \frac{x - x'}{1 - 1/f'(x)}, \tag{13}$$

we find, using again as starting point $x = 1$, that the desired accuracy (meaning that $|\epsilon'| < 10^{-6}$) is achieved within 14 iterations, and we get

Starting point x	Solution x	Iterations
1	0.79681263	14

Table 1: Results of the relaxation method: solution and number of iterations within an accuracy of 10^{-6}.

2.2.3 Solution questions c and d

An interesting observation is to examine how the number of iterations required to reach an accuracy better than 10^{-6} changes when using the over-relaxation method and as discussed previously, in this case the error is given by Eq. (11). There is also no precise prescription for choosing the optimal value of ω: it is typically determined by trial and error, in order to minimize the number of iterations. For instance, by trying different values of ω we obtain the following results

ω	Solution x	Iterations
0.50	0.79681516	12
1.53	0.79681336	13
1.25	0.79681263	14
0.68	0.79748664	6

Table 2: Results of the over-relaxation method: solution and number of iterations for different values of ω.

From this table we see that the best choice is $\omega = 0.68$, which achieves an accuracy better than 10^{-6} in only 6 iterations, roughly half the number required by the relaxation method.

Furthermore, there are circumstances under which choosing a value of $\omega < 0$ allows us to reach the solution faster than with the ordinary relaxation method. We previously found that the error at a new step, denoted by ϵ', depends on the error at the previous step, ϵ, as follows:

$$\epsilon' = \epsilon\big((1+\omega)f'(x^*) - \omega\big). \tag{14}$$

This implies that, in order to achieve convergence, we must have

$$\big|(1+\omega)f'(x^*) - \omega\big| < 1, \tag{15}$$

which can be rewritten as

$$-1 - f'(x^*) < \omega\,(f'(x^*) - 1) < 1 - f'(x^*), \tag{16}$$

or equivalently,

$$\frac{-1 - f'(x^*)}{f'(x^*) - 1} < \omega < \frac{1 - f'(x^*)}{f'(x^*) - 1}. \tag{17}$$

From this, we see that a negative value of ω can ensure convergence in cases where $1 - f'(x^*) < 0$, as in this situation both sides in the previous inequality are negative.

3 Exercise 6.13 from Newman: Wien's displacement constant

Since we will use the binary method, let us briefly recap how it works.

3.1 Binary Method

Binary search, also known as the bisection method, is a robust technique to find the root of a function. The method works by repeatedly narrowing down an interval in which a solution exists. If there is a single solution within the specified interval, the bisection method is guaranteed to find it. To solve an equation of the form $f(x) = 0$ using the bisection method, follow these steps:

1. choose initial interval: select two initial points x_1 and x_2 such that $f(x_1)$ and $f(x_2)$ have opposite signs, i.e., $f(x_1) \cdot f(x_2) < 0$. Decide on a target accuracy ϵ for the solution;
2. compute midpoint: calculate the midpoint of the interval:

   $$x' = \frac{x_1 + x_2}{2},$$

   and evaluate $f(x')$.
3. update interval:
   - if $f(x')$ has the same sign as $f(x_1)$, set $x_1 = x'$;
   - otherwise, set $x_2 = x'$;

4. check accuracy: if $|x_2 - x_1| > \epsilon$, return to step 2. Otherwise, compute the midpoint one last time

$$x_{\text{root}} = \frac{x_1 + x_2}{2},$$

and take this as the final estimate of the root.

The bisection method has some limitations. For example, if the function has an even number of roots within the chosen interval, so that $f(x)$ crosses the zero line an even number of times, the method may fail to find a root (there are additional limitations and potential issues that are not discussed here). The method is also applicable only to single-variable equations, unlike other methods such as the relaxation method, which can handle systems of equations.

3.2 Solution of the Exercise

3.2.1 Solution question a: Wien's displacement constant

The Planck radiation law describes the spectral intensity of a blackbody at temperature T. The intensity of radiation per unit area and per unit wavelength λ is given by:

$$I(\lambda) = \frac{2\pi hc^2 \lambda^{-5}}{e^{hc/(\lambda k_B T)} - 1}, \tag{18}$$

where h is Planck's constant, c is the speed of light, and k_B is Boltzmann's constant. Differentiating with respect to λ, we obtain

$$\frac{d}{d\lambda}\left(\frac{2\pi hc^2\lambda^{-5}}{e^{hc/(\lambda k_B T)} - 1}\right) = 2\pi hc^2\left(-5\frac{\lambda^{-6}}{e^{hc/(\lambda k_B T)} - 1} + \frac{\lambda^{-5}(hc/\lambda^2 k_B T)e^{hc/(\lambda k_B T)}}{(e^{hc/(\lambda k_B T)} - 1)^2}\right) \tag{19}$$

$$= 2\pi hc^2\lambda^{-6} \cdot \frac{-5(e^{hc/(\lambda k_B T)} - 1) + (hc/\lambda k_B T)e^{hc/(\lambda k_B T)}}{(e^{hc/(\lambda k_B T)} - 1)^2}. \tag{20}$$

In particular, to find the wavelength λ at which the emitted radiation is strongest, we need to find the maximum of $I(\lambda)$. From the above equation, this leads to

$$5e^{-hc/(\lambda k_B T)} + \frac{hc}{\lambda k_B T} - 5 = 0, \tag{21}$$

that gives us the wavelength at which the radiation is maximal. Now, making the substitution $x = \frac{hc}{\lambda k_B T}$, we can invert this relation to find that the maximum wavelength satisfies

$$\lambda_{\max} = \frac{b}{T}, \tag{22}$$

where

$$b = \frac{hc}{k_B x}, \tag{23}$$

is the so-called *Wien displacement constant*, and x is the solution of the following non-linear equation

$$5e^{-x} + x - 5 = 0. \tag{24}$$

Therefore, we must solve the latter equation in order to determine the value of x, and then use this solution to compute the wavelength λ that maximizes the radiation intensity $I(\lambda)$. To find the value of x that solves this equation, we apply the binary method.

3.2.2 Solution questions b and c

By implementing the binary search according to the steps discussed in the previous section, we wrote a code to solve the nonlinear equation in Eq. (24) to an accuracy of $\epsilon = 10^{-6}$ (meaning that the binary search method was implemented such that the algorithm iterates until $|x_{i+1} - x_i| < \epsilon$ where x_{i+1} is the solution at step $i+1$ and x_i is the solution at step i). In this case, there are in principle two solutions. Indeed, for $x \to \pm\infty$ we have $f(x) = 5e^{-x} + x - 5 \to +\infty$, and the function has a negative minimum. Therefore, it must intersect the x-axis twice. To find both solutions, we apply the binary search method on two distinct intervals (otherwise the first step described in Sec. 3.1 would not be satisfied):

- starting from $x_1 = -1$ and $x_2 = 2$, the solution is
$$x = 0.0000001192;$$
- starting from $x_1 = 2$ and $x_2 = 5$, the solution is
$$x = 4.9651144743,$$

with an accuracy of 10^{-6}. Now we can estimate Wien's displacement constant using Eq. (23). The calculation requires using the solution far from zero, corresponding to the maximum of the Planck distribution, and in particular we obtain

$$b = \frac{hc}{k_B x} \approx 0.0028970 \text{ m}\cdot\text{K}, \tag{25}$$

where the constants have the values

$$h = 6.626 \times 10^{-34} \text{ J} \cdot \text{s}, \quad c = 2.998 \times 10^{8} \text{ m/s}, \quad k_B = 1.381 \times 10^{-23} \text{ J/K}.$$

This method is commonly used to estimate the surface temperatures of astronomical bodies, such as the Sun. Indeed, since the wavelength peak in the Sun's emitted radiation occurs at $\lambda_{\max} = 502\,\text{nm}$, we can use Eq. (22) to infer the Sun's temperature. Using our values, we obtain

$$T \approx 5771.07\,\text{K}, \tag{26}$$

which is consistent with the known value for the Sun's surface temperature.

4 Exercise 3: Gradient Descent Method and its application

In this section, we discuss how the gradient descent method works, and we also present a physical application. Specifically, we show how to fit measurements of the galaxy stellar mass function using the so-called Schechter function, with all calculations performed using the gradient descent method.

4.1 Gradient Descent Method

The gradient descent method is essentially a technique used to find the minima of a function. More generally, a maximum or minimum of a function $f(x)$ can be found by differentiating it and setting the derivative equal to zero:

$$f'(x) = 0. \tag{27}$$

This shows that minima (and maxima) are nothing other than the roots (i.e., zeros) of the derivative function f'. Among these, the most efficient is Newton's method, where the fundamental formula for the Gauss-Newton method is

$$x = x - \frac{f'(x)}{f''(x)}. \tag{28}$$

It is relatively rare, however, that the derivatives of the function $f(x)$ can be calculated exactly, so the Gauss-Newton method is only occasionally used. If only the first derivative of a function can be computed, an approximate version of the Gauss-Newton method can still be applied by writing:

$$x' = x - \gamma f'(x), \tag{29}$$

where γ is a constant that represents a rough estimate of $1/f''(x)$. This latter approach is known as the gradient descent method. But the derivative can also be estimated numerically. Then, the solution at step $i+1$ is given by

$$x_{i+1} = x_i - \gamma \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}. \tag{30}$$

So far, we have considered only the one-dimensional case. However, the method can be extended to the n-dimensional case, where the solution at step $i+1$ is given by

$$\vec{x}_{i+1} = \vec{x}_i - \gamma \vec{\nabla} f(\vec{x}), \tag{31}$$

where $\vec{\nabla} f(\vec{x})$ is the gradient of the function with respect to all components of the vector $\vec{x}$. In the following, we will show explicitly how this works in the two- and three-dimensional cases.

4.1.1 Test function

We first apply the gradient descent method to a test function, given by

$$f(x, y) = (x-2)^2 + (y-2)^2, \tag{32}$$

in order to find its minimum value. Using Eq. (31), in this two-dimensional case we obtain

$$\begin{aligned} x_{i+1} &= x_i - \gamma \, \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}, \\ y_{i+1} &= y_i - \gamma \, \frac{f(y_i) - f(y_{i-1})}{y_i - y_{i-1}}. \end{aligned} \tag{33}$$

Note that we must first choose an initial value x_0. Then, the next point x_1 can be computed using the backward finite difference approximation of the derivative:

$$\begin{aligned} x_1 &= x_0 - \gamma, \frac{f(x_0, y_0) - f(x_0 - h, y_0)}{h}, \\ y_1 &= y_0 - \gamma, \frac{f(x_0, y_0) - f(x_0, y_0 - h)}{h}, \end{aligned} \tag{34}$$

and then subsequent iterations are then carried out using Eq. (33).

Notice also that we have to choose a tolerance of 10^{-6}, meaning that we iterate these formulas until the condition

$$|x_{i+1} - x_i| < 10^{-6} = \epsilon_{\text{tol}} \tag{35}$$

is satisfied. Implementing this procedure, tor the test function under consideration, after 48 iterations and choosing as starting point $(x_0, y_0) = (1, 1)$, we obtain the minimum at

$$x = 1.999996, \quad y = 1.999996, \tag{36}$$

which is consistent with the analytical result for the minimum, namely

$$(x, y) = (2, 2). \tag{37}$$

Another way to demonstrate that the algorithm works well is to plot the values of x_{i+1} and y_{i+1} at each step, and observe that they converge to the true minimum $(x, y) = (2, 2)$. Indeed we have

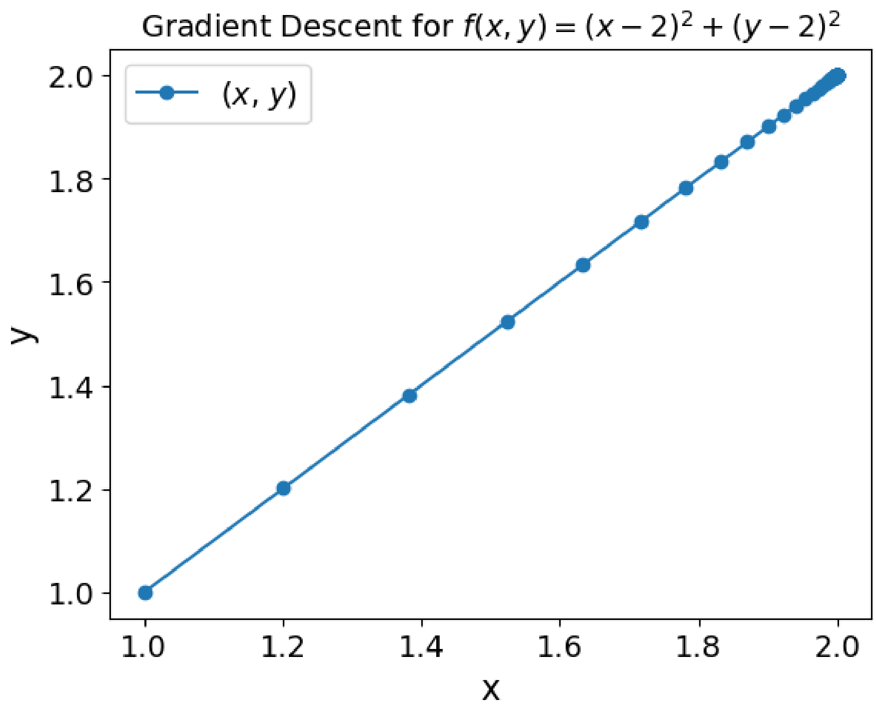

Figure 1: Plot of (x_{i+1}, y_{i+1}) for all i. We observe that, as the iterations increase, the values gradually converge to the true minimum at $(x, y) = (2, 2)$.

We see that the points converge to the true minimum, confirming that the gradient descent implementation works correctly.

4.2 Schechter function Fit

Now that we have verified the correct implementation of the gradient descent method, we can apply it to a practical astrophysical problem: fitting the measurements of the galaxy stellar mass function from the COSMOS survey, provided in a data file. The dataset is organized into three columns:

1. $\log M_{\rm gal}$ [dex];
2. $n(M_{\rm gal})$ [1/dex/Volume];
3. uncertainty in $n(M_{\rm gal})$;

Here, "dex" denotes the base-10 logarithm of the stellar mass, and the volume is expressed in units of $(\mathrm{Mpc}/h)^3$. Plotting this data in a log-log plot, we have the following behavior

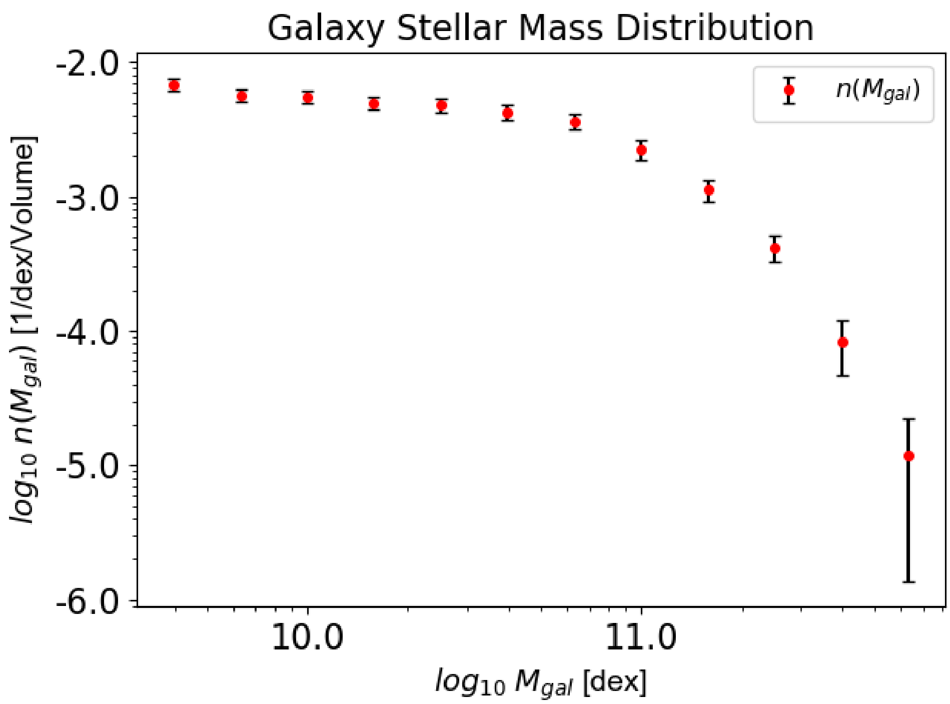

Figure 2: Plot of the data contained in the file: they represent the galaxy mass distribution with relative error.

Measurements of this type are typically modeled with a Schechter function, which in this case has the form

$$n(M_{gal}) = \phi^* \left(\frac{M_{gal}}{M^*}\right)^{\alpha+1} \exp\left(-\frac{M_{gal}}{M^*}\right) \ln(10), \tag{38}$$

where in this formulation, ϕ^* represents the amplitude, M^* the characteristic mass scale at which the function transitions from a power law to an exponential cutoff, and α the low-mass slope. These three quantities are free parameters.

To fit the data with the Schechter function, we minimize the χ^2 statistic, defined as

$$\chi^2 = \sum_i \frac{(y_i - f(x_i))^2}{\sigma_i^2}, \tag{39}$$

where y_i are the measured data points corresponding to x_i, σ_i their associated uncertainties, and $f(x_i)$ the model prediction at x_i. In our case, $f(x_i)$ is given by the Schechter function in Eq. (38). A good fit is obtained by finding the set of parameters that minimizes the χ^2 value in Eq. (39). To find this minimum of the χ^2 function, we use the gradient descent method. In our case, the Schechter function has three free parameters, so we need to implement the gradient descent method in three dimensions, using Eq. (31). First of all, we consider the initial guesses for the parameters

	$\log_{10}(\phi^*)$	$\log_{10}(M^*)$	α
Initial guess	-2.5	10	-1.5

Table 3: Initial guesses for the free parameters of the Schechter function.

and we take as learning rate $\gamma = 10^{-4}$ while $h = 10^{-2}$ for the first step. By applying the gradient descent method, we find that the parameters which minimize the χ^2 function, with an accuracy of 10^{-6}, are as follows

	$\log_{10}(\phi^*)$	$\log_{10}(M^*)$	α
Best-fit Parameters	-2.568126	10.976169	-1.009303

Table 4: Values of the free parameters that minimize the χ^2 function.

Indeed, by plotting the Schechter function from Eq. (38) using these parameters and comparing it with the data, we obtain

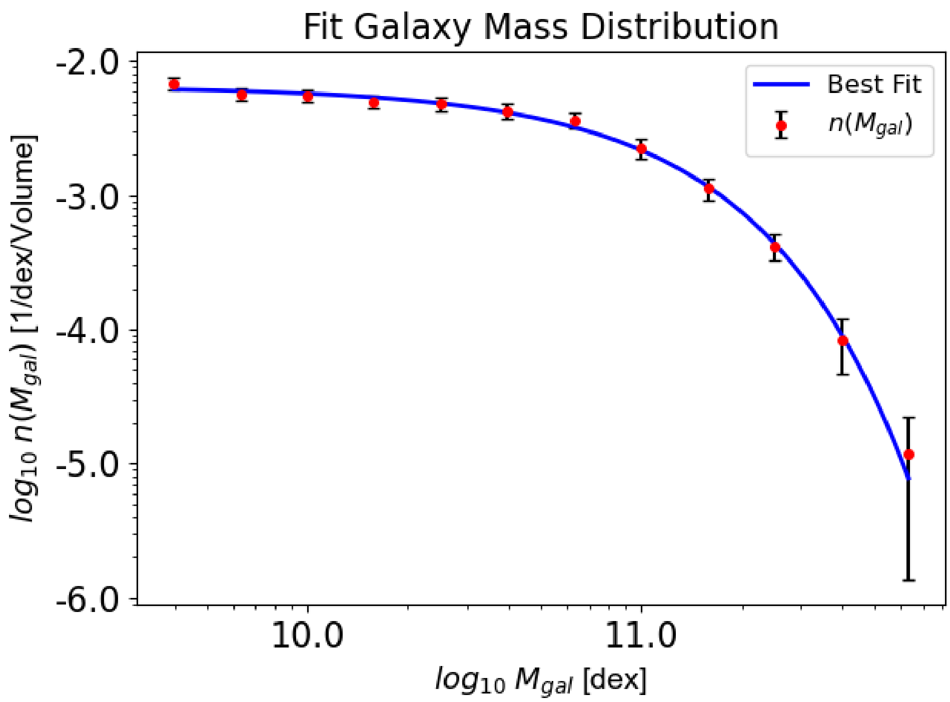

Figure 3: Plot of the data contained in the file and the fit obtained minimizing the χ^2 function.

We immediately see that the Schechter function, using the parameters obtained from minimizing the χ^2, provides an excellent fit to the data. By contrast, plotting the Schechter function with the initial guesses yields a much poorer agreement, as shown below

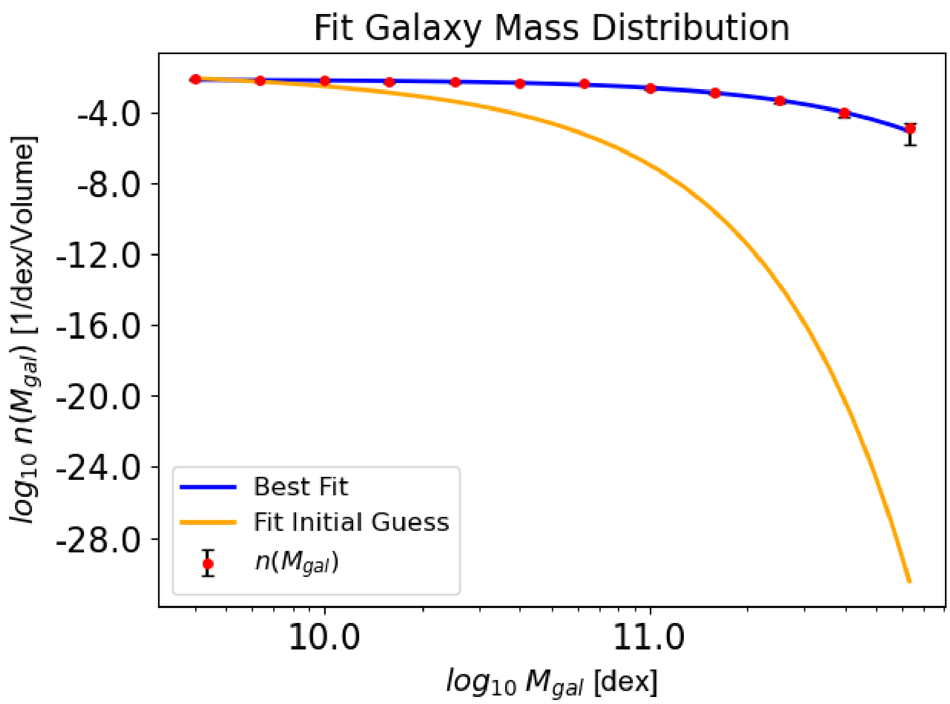

Figure 4: Plot of the data from the file, together with the best–fit Schechter function obtained by minimizing the χ^2, and the Schechter function evaluated using the initial parameter guesses.

Now, in order to see the convergence of the gradient descent method, or in other word the fact that our function χ^2 achieve the minima, we can plot the behavior of the χ^2 in function of the step. We do this for different initial guess, in particular for the following starting points

	$\log_{10}(\phi^*)$	$\log_{10}(M^*)$	α
Guess 1	-4.0	10.0	-1.5
Guess 2	-3.5	9	-0.8
Guess 3	-2.5	10.0	-1.5

Table 5: Initial values of the free parameters.

and what we get is the following

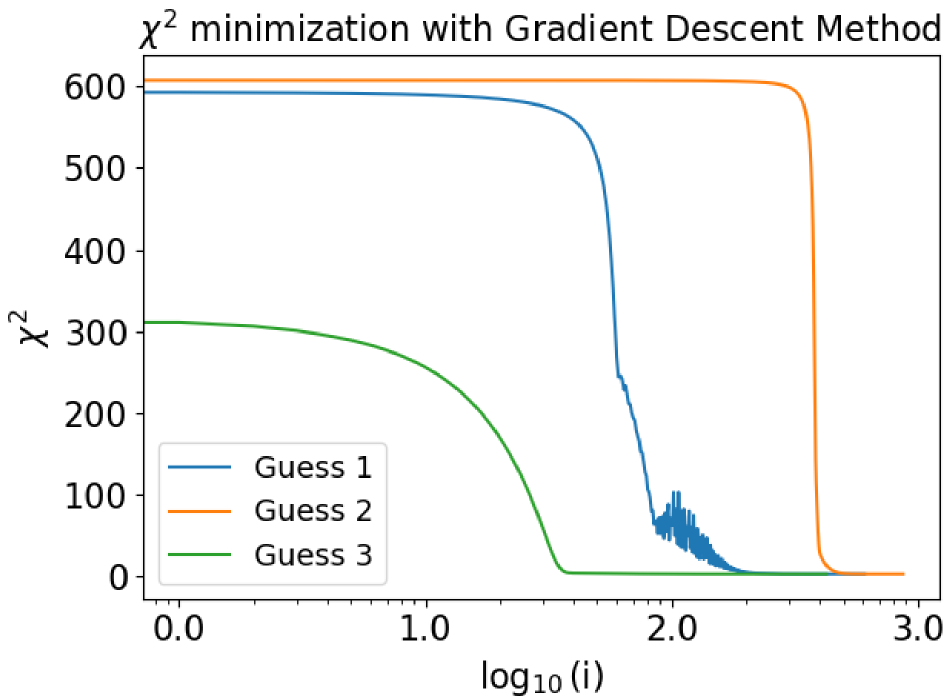

Figure 5: Behavior of the χ^2 function as a function of the iteration step i. A logarithmic scale is used on the x-axis to better highlight the convergence behavior.

We see immediately that, for all three initial guesses, the gradient descent method converges and indeed finds a minimum of the χ^2 function (which is more or less 3). Notice also that, when the initial guess is closer to the true parameter values (as in the case of Guess 3), the method converges faster, as expected; otherwise, more iterations are required. We also recall that all these results were obtained with an accuracy of 10^{-6}, our chosen threshold. Requesting a higher accuracy would, of course, lead to slightly different results. Finally, for all three initial guess cases, we obtain essentially the same fit (not shown here for brevity), which is displayed in Fig. 4.2, as expected.

5 Conclusion

In this homework, we have explored how to solve non-linear equations using the relaxation and overrelaxation methods, comparing their relative efficiency. We also learned how to find the root of a function using the well-known binary method, and applied this to a physical problem: determining the Wien displacement constant. Finally, we implemented another important numerical technique for minimizing a function, applying it to fit the galaxy stellar mass function data by minimizing the χ^2 function, and studied its convergence behavior as a function of the number of iterations.

6 Code

The code containing all the results can be found at the following Repository GitHub.