Computational Physics Homework II

Link to GitHub repository

10/10

1 Overrelaxation

We aim to numerically find a solution to the equation

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

using the method of overrelaxation. I start by deriving an estimate for the error. In the case of overrelaxation, the current estimate of the solution x and next estimate x' are related by

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

where the parameter ω is some value, generally greater than zero, that is found heuristically. If we define the true solution of our equation to be x^*, we can write the error of our current estimate as

$$\epsilon = x^* - x\,, \tag{3}$$

and that of the next step as

$$\epsilon' = x^* - x'\,. \tag{4}$$

Taylor-expansion of Equation 2 close to the real solution x^* gives

$$x' \approx (1+\omega)f(x^*) - wx^* + (x - x')\left((1+\omega)f(x^*) - \omega\right)\,, \tag{5}$$

and identifying $f(x^*) = x^*$ per definition, we arrive at

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

and thus,

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

Solving for the error ϵ' we arrive at the estimate

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

I implement both the relaxation and the overrelaxation method in my code and use them to to iteratively solve the equation

$$x = 1 - e^{2x} \,. \tag{9}$$

/ wow.

With a starting point of $x_0 = 0.5$ it takes 16 iterations to solve the equation to a precision of 10^{-6} using relaxation. With overrelaxation, I only need 3 iterations to reach the same precision, with the same starting point and a parameter $\omega = 0.7$ (found by trial and error). In some cases, a negative ω might be beneficial, if ... ... yes?

2 Wien's displacement constant

We start out with Planck's radiation law for the intensity of radiation per unit area per unit wavelength λ for a black body at temperature T:

$$I(\lambda) = \frac{2\pi hc^2\lambda^{-5}}{e^{hc/\lambda k_B T}} \tag{10}$$

We want to find the wavelength of maximal radiation, so we differentiate:

$$\frac{dI}{d\lambda} = -5\,\frac{2\pi hc^2\lambda^{-6}}{e^{hc/\lambda k_B T}} - \frac{2\pi hc^2\lambda^{-5}}{(e^{hc/\lambda k_B T})^2}(hc/\lambda^2 k_B T) \overset{!}{=} 0 \tag{11}$$

and with some basic manipulations we arrive at the equation

$$\frac{hc}{\lambda k_B T} + 5\,e^{-hc/\lambda k_B T} - 5 = 0 \,. \tag{12}$$

With the substitution $x = \frac{hc}{\lambda k_B T}$, this is equivalent to

$$x + 5\,e^{-x} - 5 = 0 \,, \tag{13}$$

and we can write the wavelength of maximal radiation as

$$\lambda = \frac{hc}{k_B T x} \equiv \frac{b}{T}, \tag{14}$$

where we have defined Wien's displacement constant $b = \frac{hc}{k_B x}$. I implement a binary search method to find the solution of Equation 13. I obtain $x = 4.965\,11\,\mathrm{m}$ within 22 iterations. This corresponds to a displacement constant of $b = 2.8982 \times 10^{-3}$ mK (meters times Kelvin).

For a centre wavelength of $\lambda = 502\,\mathrm{nm}$ of the sun's radiation spectrum, we obtain a temperature of the sun of

$$T_{\mathrm{sun}} = \frac{b}{\lambda} \approx 5770\,\mathrm{K} \tag{15}$$

from Equation 14.

3 Multi-dimensional gradient descent

I start out by implementing a gradient descent algorithm and testing it on the simple function

$$f(x,y) = (x-2)^2 + (y-2)^2\,. \tag{16}$$

Figure 1 shows the x value versus the numbers of iteration for different starting values. The y starting values are chosen to be the same as the x-values.

Since I need to (different) values in the gradient descend step of the loop, in order to not divide by zero, I give a second initial value produced by adding my estimate of the inverse second derivative, γ to the first one (where *gamma* just serves as a small number that is larger than the convergence limit). I then cycle through the values in each loop, and stop when the maximum of the current distances between values is below a preset threshold or when a maximum number of steps is reached.

I then apply the code to fitting a function of the given form

$$n(M_{gal}) = \Phi^* \left(\frac{M_{gal}}{M_*}\right)^{\alpha+1} \exp\left(-\frac{M_{gal}}{M_*}\right) \ln(10) \tag{17}$$

to the provided data. Here the function that is minimised is the χ^2 of the fit. I guessed parameters of $\Phi = 0.002$, $M_* = 10^{10.8}$, and $\alpha = -1$ from plotting the data, but for my best fit I used the results of previous fits as starting parameters.

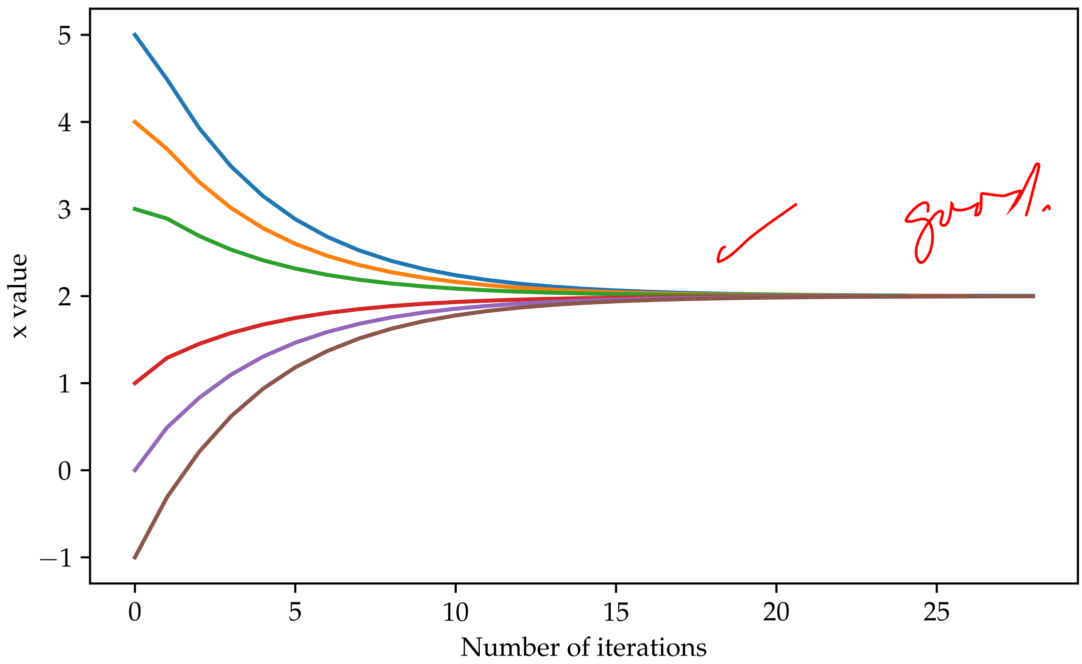
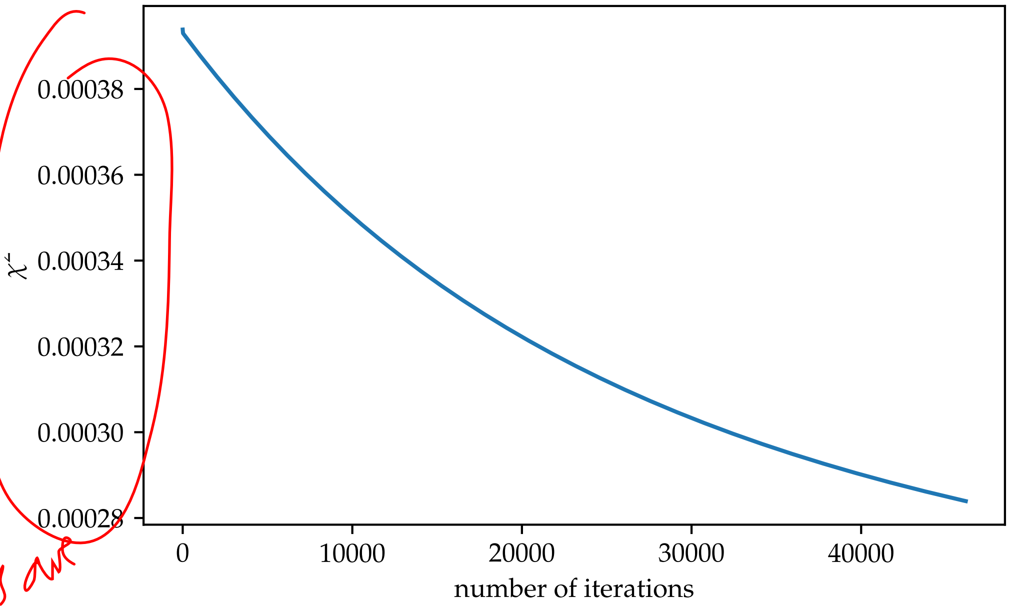


Figure 1: Convergence of gradient descent algorithm when finding the minimum of Equation 2 for different starting values.

I had some problems with the convergence of M_*, since the numbers are so large. Consequently I fitted $\log(M_*)$ instead.

With $\gamma = 10^{-4}$ and sufficient number of steps, I can get the results to converge for different starting values, however the fit does not always look great for large values of M_{gal}.



These numbers are weird. did you use the χ^2 defn from class?

Figure 2: Convergence of χ^2 when minimising using the gradient decent algorithm.

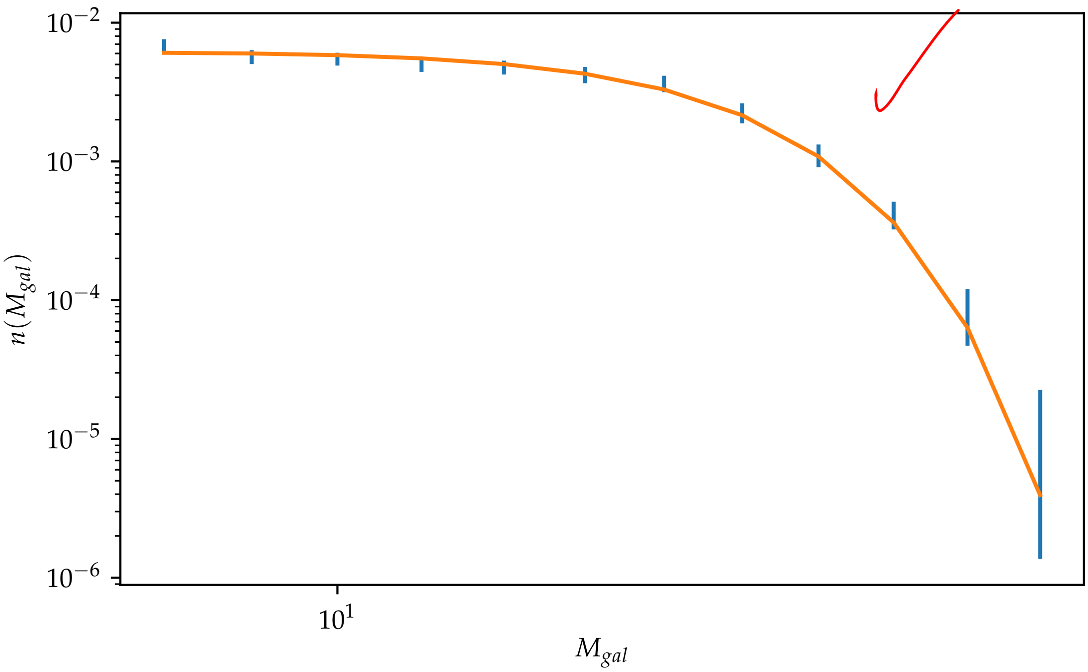


Figure 3: Given initial data and best fit produced by my code, with a convergence to 3×10^{-7} in 46174 iterations. The resulting parameters are $\Phi = 0.0030859$, $M_* = 10^{10.9208646}$, and $\alpha = -0.9637289$.