Computational Physics Homework 2

Nicolas Loizeau

October 11, 2019

11/10

1 Newman 6.10: relaxation

We solve the equation $x = 1 - e^{-cx}$ using the relaxation method and a precision of 1E-6.

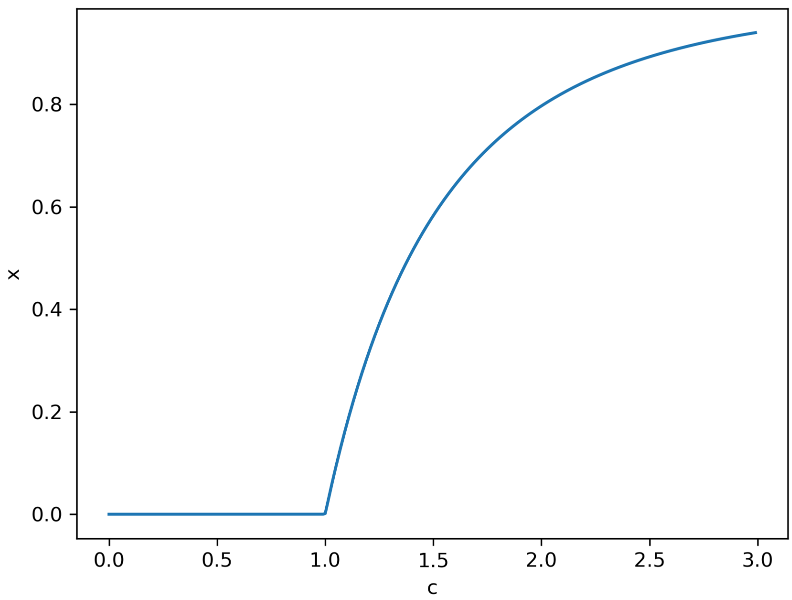


Figure 1: solution of $x = 1 - e^{-cx}$ versus parameter c.

2 Newman 6.11: overrelaxation

In each step of the overrelaxation method, the new x value is updated to $x' = (1+\omega)f(x) - \omega x$

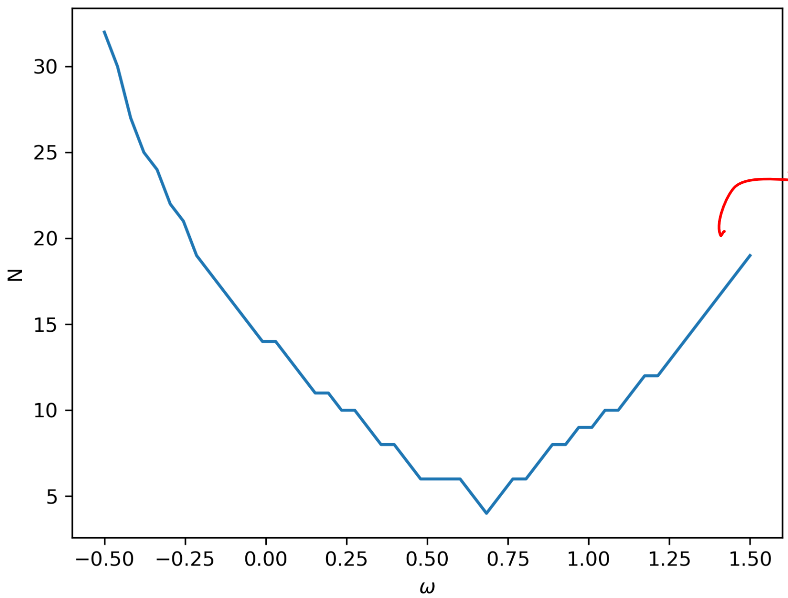
We solve the equation $x = 1 - e^{-2x}$ using the overrelaxation method and a precision of 1E-6. The relaxation method reaches the precision of 1E-6 in 14 steps whereas the overrelaxation method does it in 5 steps with $\omega = 0.7$. We do a first order development of an algorithm step:

$$\begin{aligned} x' &= (1+\omega)f(x) - \omega x \\ &= (1+\omega)\left[f(x^*) + (x - x^*)f'(x^*) + ...\right] - \omega x \\ &= (1+\omega)x^* + (1+\omega)(x-x^*)f'(x^*) - \omega x \\ x - x' &= \omega x^* + (1+\omega)(x-x^*)f'(x^*) - \omega x \\ &= \omega(x^* - x) + (1+\omega)(x-x^*)f'(x^*) \\ &= (x^* - x)\left[\omega - (1+\omega)f'(x^*)\right]. \end{aligned} \tag{1}$$

Now, using the same reasoning as for eq. 6.83 of Newman, we get

$$\epsilon \approx \frac{x - x'}{1 - 1/[(1+\omega)f'(x) - \omega]}. \tag{2}$$

✓



nice presentations

Figure 2: Number of steps needed to reach an error of 1E-6 with the overrelaxation versus ω

3 Newman 6.13: binary search

We want to maximize

$$I(\lambda) = \frac{2\pi hc^2\lambda^{-5}}{e^{hc/\lambda k_B T} - 1} = \frac{2\pi hc^2\lambda^{-5}}{e^{\lambda_0/\lambda} - 1}. \tag{3}$$

looking for a root of

$$
\begin{aligned}
I'(\lambda) &= 0 \Leftrightarrow && (4)\\
\frac{\lambda^{-5}}{e^{\lambda_0/\lambda}-1} &= 0 \Leftrightarrow && (5)\\
\frac{-5\lambda^{-6}(e^{\lambda_0/\lambda}+1)+\lambda^{-2}e^{\lambda_0/\lambda}\lambda^{-5}}{(e^{\lambda_0/\lambda}-1)^2} &= 0 \Leftrightarrow\\
5e^{-\lambda_0/\lambda}+\lambda_0/\lambda-5 &= 0
\end{aligned}
$$

which is equivalent to finding a root of $5e^{-x}+x-5=0$ with $\lambda=\lambda_0/x$. Using a binary search initialized with $x_1=4.5$ and $x_2=5.5$ we find T=5772K.

4 Schechter function

4.1 Gradient descent

First we implement a gradient descent method. Partial derivatives are computed using midpoint method. We test this method on a simple example (Figure 3).

nice plot, but better to center on (2,2) to see gradient on all sides

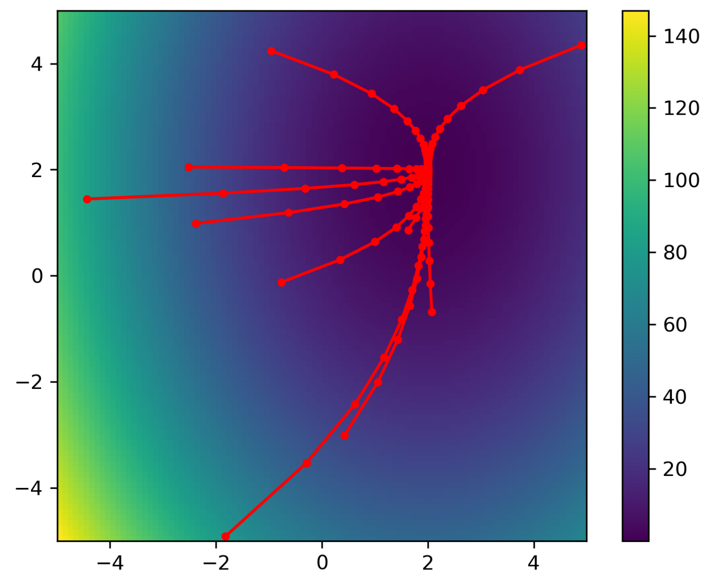


Figure 3: Steps of the gradient method optimizing $f(x,y)=2(x-2)^2+(y-2)^2$ starting from 10 random points.

4.2 Data fitting

We want to fit the data $n(M_{gal})$ with the Schechter function. First the dex data are converted in masses. The fits are performed in mass space and in units of 1E11 solar mass in order to ensure the numbers are close to 1. We solve the minimization problem

$$\min_{\phi, M_*, \alpha} \left[\chi^2(\phi, M_*, \alpha) = \frac{(n_{data} - n_{model}(\phi, M_*, \alpha)^2}{\sigma_n^2} \right]. \tag{6}$$

We use a constant step gradient computation and a vector γ parameter for the gradient descent algorithm.

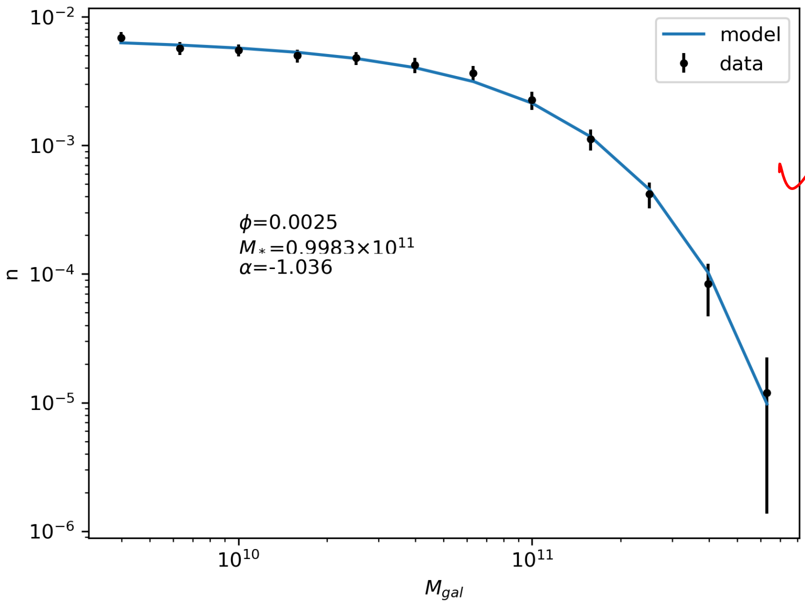


Figure 4: Best fit of the Schechter function.

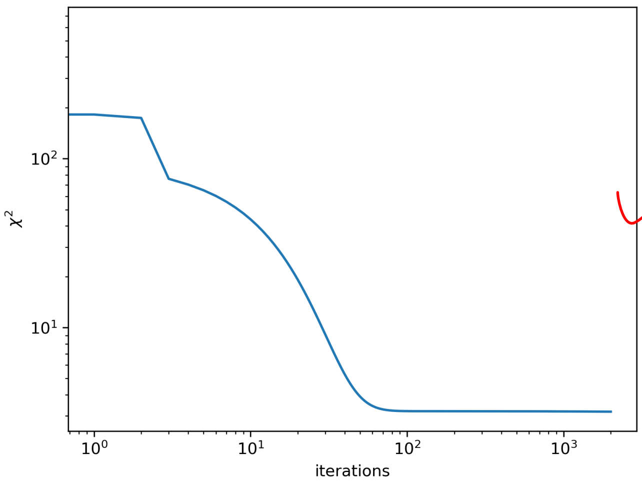
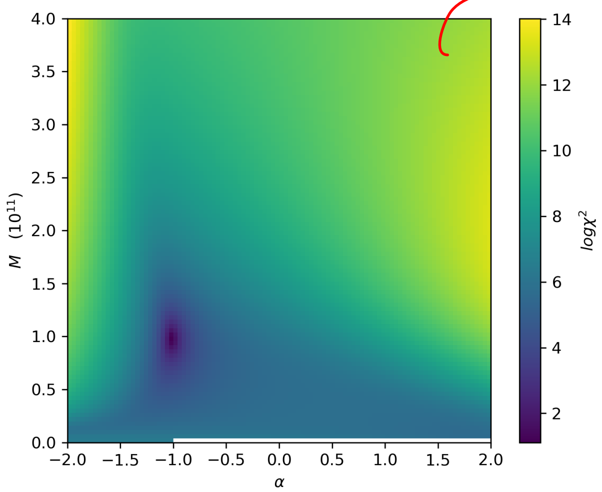


Figure 5: Convergence of the χ^2

excellent



+1 for extra effort

Figure 6: χ^2 versus M and α at fixed ϕ. We clearly see the minimum, but the χ^2 function is non convex and returns NaN for negative masses. Hence the convergence of the method is very dependent on the initial parameters.