Adding simulated Annealing to Monte Carlo code

# Changing the input

ini\_temp 0.1  
final\_temp 0.1  
n\_step 100000  
x\_ini -1.5,0.5,1  
x\_delta 0.5  
seed 1637  
foutname ex1.csv

Last week, your code read an input file as in the text in the box, but we did not use one of the values. Now when you read the text, make sure you store final\_temp. You will need to substitute it into eq. (3) below.

# Converting the code from Monte Carlo to simulated annealing

Assume you have a functioning Monte Carlo code. All you have to do is add a few lines so the temperature is gradually reduced at each step. We will implement exponential cooling. To get the temperature at step , one says

|  |  |
| --- | --- |
|  | (1) |

where is a decay constant. If you know the value for , you could use eq (1) directly, but this would mean calling an exponential function at each step. It is simpler to multiply the temperature by some value at each step so, . Then

|  |  |
| --- | --- |
|  | (2) |

To get the value of , we rearrange (1)

|  |  |
| --- | --- |
|  | (3) |

and substitute the known values. We set

* to the number of steps in the simulation
* to our final temperature
* to our initial temperature

To summarise from the point of view of programming:

* From your input you have initial and final temperatures and a number of steps
* Before you start your main loop, calculate and calculate and store it in a variable like c\_mult
* In your main loop you add one line  
  temp = temp \* c\_mult

Checking

There are at least two things you can do to check for mistakes.

1. If you have values like and and you have then the variable c\_mult will have a value that is positive and a bit smaller than 1.
2. Plot the temperature at each step and make sure it looks like a smooth exponential decay going from the correct starting temperature and finishing at the correct final temperature. Do not worry if there is small rounding error.