UNIVERSITY OF HELSINKI
DEPARTMENT OF PHYSICS

BASICS OF MONTE CARLO SIMULATIONS

Exercise 4

Student: Caike Crepaldi

Professor: Flyura Djurabekova

February 2016

Note. In order to run the exercise's source codes, use the command *make all* (see the Makefile for more compilation commands and details). The graphs and plots are available in the ./Figures/ folder.

Problem 1

Please compile and run the **ex4p1** program. The output is the RMS for 10^6 points. If needed, the user can declare the number of points as an optional command line argument.

The comparison of the expected distribution and the one obtained by the Markov Chain Monte Carlo can be seen in figure 1. The chosen optimal value for ΔE_{MAX} was 0.1, see the plot in figure 2.

See the shell script in order to see the commands used to create the data ploted in the figure 3. We can notice that the RMS values follow what it seems to be a distribution that resembles a function like $f(x) = x^{-1}$. In the graph we have a basic fitting, in this case I tried $f(x) = ax^{-3} + b$, with 2 adjusted parameters, a and b.

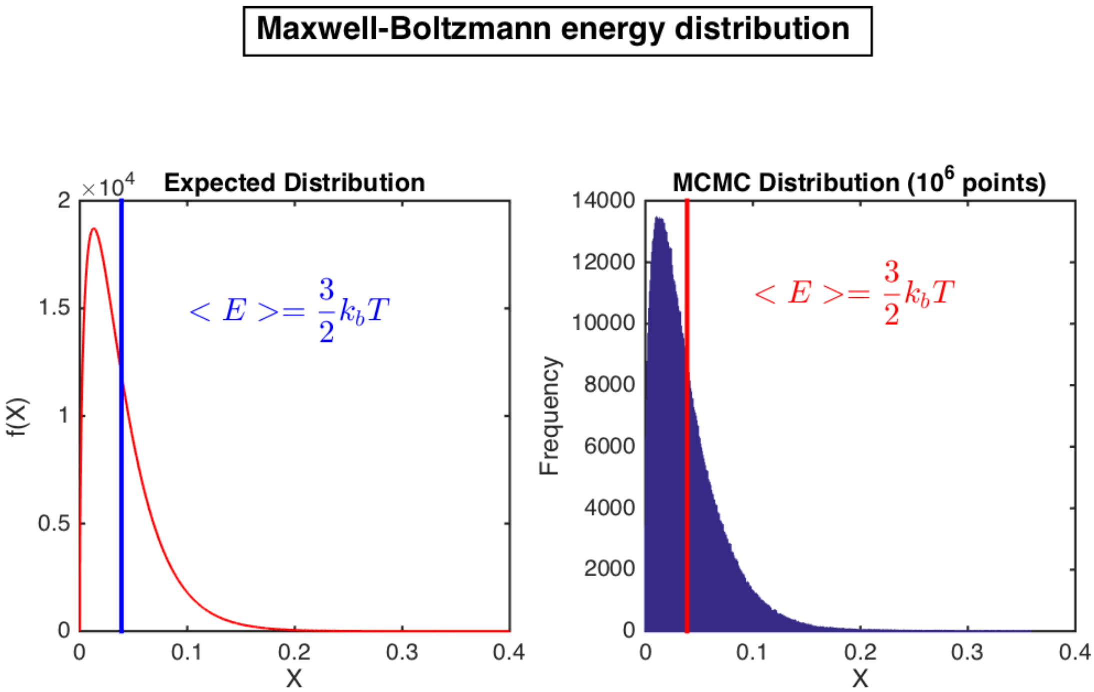

Figure 1: Maxwell-Boltzmann energy distribution for particles: The expected distribution and the one obtained by the MCMC method for 10^6 points.

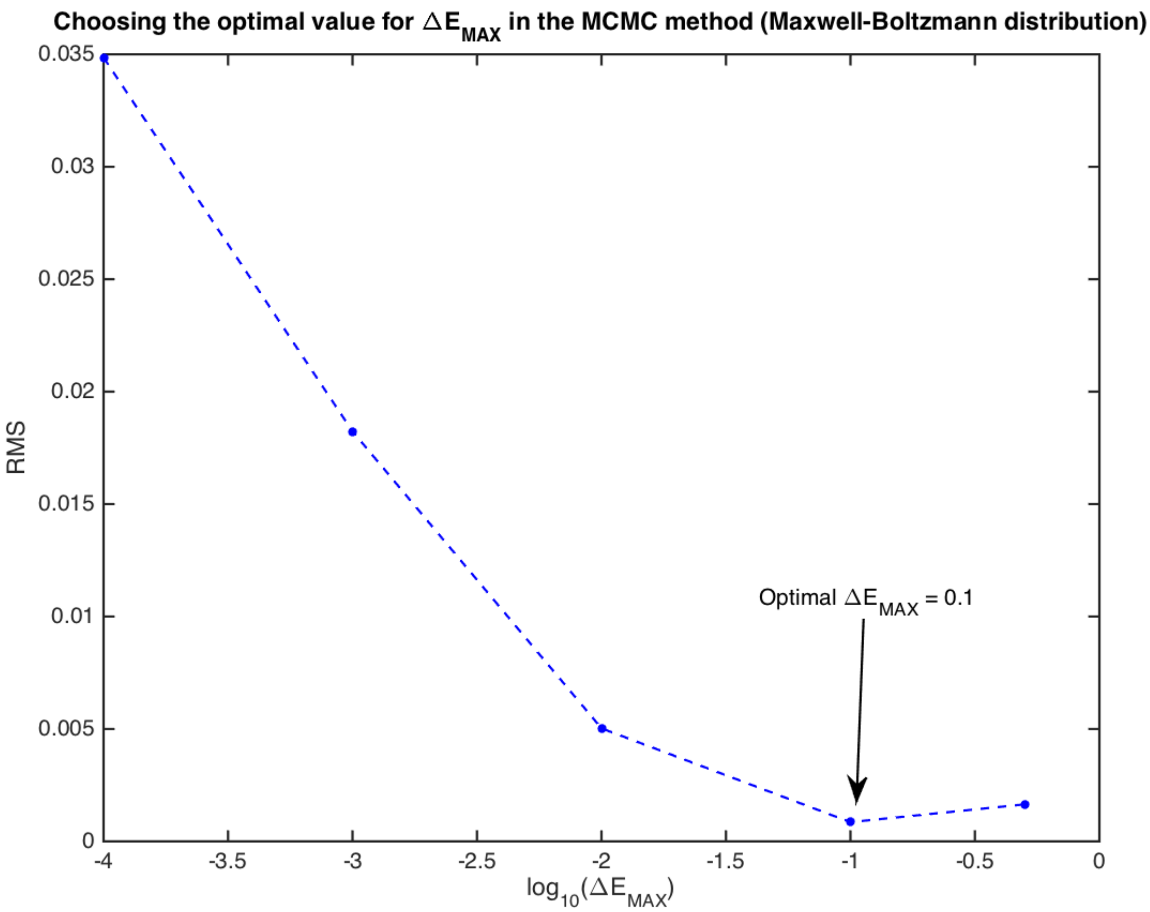

Figure 2: Mean Square (RMS) Error from ΔE_{MAX}.

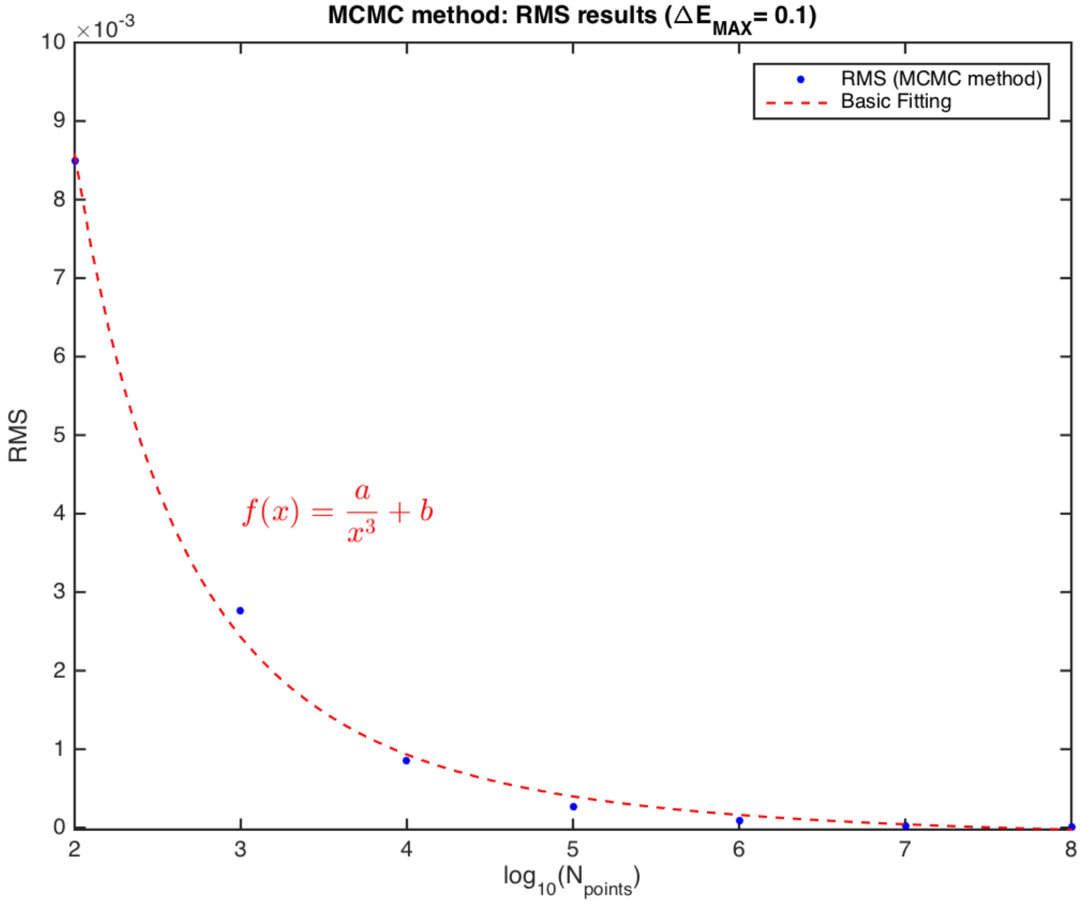

Figure 3: Mean Square (RMS) Error from several runs.

Problem 2

To see the results for the first, experimental, distribution, please compile and run the **ex4p2** program. To see the results for the second, theoretical, distribution, please compile and run the **ex4p2b** program.

Both programs uses 1000 distributions with 100 counts in each distribution. The value of N_h and $\Delta\bar{x}$ is chosen accordingly with the mean distribution in each case. See the output below. Notice that the gaussian error (standard deviation of the mean) is far from the error obtained by the MC based analysis of data in the first distribution, but quite close from it in the second distribution.

A small script in awk, python or MATLAB, can be used to generate the distribution seen in the assignment's figure and used in the Fortran routine. In my case, I used a awk routine inside a shell script, see file poisson.sh.

COMMAND 1

```
$ ./ex4p2
 True mean of the distribution:    559.71147726469133
 Lower bound uncertainty:    34.511477264691280
 Upper bound uncertainty:    34.588522735308743
 Gaussian distr., Std. deviation of the mean:    2.0245572881479927
$ ./ex4p2b
 True mean of the distribution:    2.9975687962643307
 Lower bound uncertainty:  0.16756879626433063
 Upper bound uncertainty:  0.18243120373566946
 Gaussian distr., Std. deviation of the mean:  0.16996460129057836
```