Worksheet 4: Error Analysis and Langevin Thermostat

Olaf Lenz

December 14, 2012
Institute for Computational Physics, University of Stuttgart

Contents

Merry Christmas and a happy new year!

1 General Remarks

- Deadline for the report is **Thursday, 10th January 2013, 10:00**.
- On this worksheet, you can achieve a maximum of 20 points.
- The report should be written as though it would be read by a fellow student who attends to the lecture, but does not do the tutorials.
- To hand in your report, send it to your tutor via email
  - Olaf (`olenz@icp.uni-stuttgart.de`; Thursday, 14:00 – 15:30)
  - Jens (`smiatek@icp.uni-stuttgart.de`; Friday, 8:00 – 9:30)
- Please attach the report to the email. For the report itself, please use the PDF format (we will *not* accept MS Word DOC files!). Include graphs and images into the report.
- If the task is to write a program, please attach the source code of the program, so that we can test it ourselves.
- The report should be 5–10 pages long. We recommend to use LaTeX. A good template for a report is available.
- The worksheets are to be solved in groups of two or three people.

On this worksheet, you will finally complete the MD simulations of the Lennard-Jones System and do error analysis on the produced data. In the first part, you will develop functions to perform error analysis of time series data, and you will apply them to artificial datasets with a defined error. In the second part, you will first implement the Langevin thermostat for the LJ simulation, then perform some simulation runs and finally analyze the results using the error analysis functions from the first part.

All files that are required for this tutorial can be found in the archive `templates.tar.gz` that can be downloaded from the lecture's homepage.

2 Error Analysis

From the lecture's home page, you can download the file `janke02.pdf` of an article by Wolfhard Janke. This article was what the lecture on error analysis based on, and all that needs to be done in this part of the worksheet can be learned from the article.

The data file `series.dat` contains a pickled (100000×5)-NumPy-Array with 5 time series of 10000 values each. Each time series has a different mean value and correlation time. Write a Python program that reads the data via pickle and plots the first 1000 points of the series to get a feeling for the data.

2.1 Autocorrelation Analysis

Task (3 points)

- Implement a Python function that computes the autocorrelaton function of a given time series.
- Compute the autocorrelation function of the data sets.
- Plot the autocorrelation function of the datasets.
- Plot the integrated autocorrelation function. The function should converge against the autocorrelation time.
- Can you guess the autocorrelation times from the plots?
- Why is it not useful to integrate the function over the whole interval where it can be computed to detetmine the autocorrelation time?

Hint For those of you that have heard the lecture "Physik auf dem Computer", you can use the Python function to compute the autocorrelation function via the Fast Fourier Transform for the tasks. Note that you still have to implement the function to compute the autocorrelation function directly in the previous task!

Task (3 points)

- Implement a Python function that performs automatic error analysis via autocorrelation analysis of a given time series of an observable.
- The function should compute an estimate for the autocorrelation time via the running autocorrelation time estimator (eq. 33 in ref), where you cut off the integration when $k_{\max} \geq 6\hat{\tau}_{\mathcal{O},\text{int}}(k_{\max})$.
- The function should return:
  - the mean value of the observable $\bar{\mathcal{O}}$
  - the error of the mean value of the observable $\epsilon_{\bar{\mathcal{O}}}$
  - the estimated autocorrelation time $\tau_{\mathcal{O},\text{int}}$
  - the error in the autocorrelation time $\epsilon_{\tau_{\mathcal{O},\text{int}}}$
  - the effective statistics N_{eff}
- Apply the error analysis function to the sample datasets. Can you guess the autocorrelation times of the different datasets?

2.2 Binning Analysis

While the autocorrelation analysis of the previous task can do its work automatically, binning analysis can only work semiautomatically, as one has to visually determine the autocorrelation time from the binning.

Task (2 points)

- Implement a Python function that computes the block variance σ_B^2 for a given block size k.
- From the block variance, one can compute an estimate for the autocorrelation time $\tau_{\mathcal{O},\text{int}} = \frac{1}{2}k\sigma_B^2/\sigma_{\mathcal{O}_i}^2$ (a.k.a. "blocking τ").
- Furthermore, one can use the block variance to estimate the error of the mean value $\epsilon_{\bar{\mathcal{O}}}^2 = \sigma_B^2/N_B$.
- Plot the blocking τ and the error estimate for $1 < k < 2000$ for the different datasets.
- Can you guess the autocorrelation time and the error of the mean value of the datasets from the plots?
- Do they match the values obtained via autocorrelation analysis?

2.3 Jackknife Analysis

Jackknife analysis for this kind of data produces identical results to the binning analysis.

Task (2 points)

- Implement a Python function that computes the Jackknife error $\epsilon_{\bar{\mathcal{O}}}^2$ of a time series for a given block size k.
- Plot the Jackkinfe error estimate for $1 < k < 2000$ for the different datasets.
- Can you guess the error of the mean value of the datasets from the plots?
- Do they match the values obtained via the above methods?

3 Combining Everything: Error Analysis of Real Simulation Data

3.1 Langevin Thermostat

Task (4 points)

- Extend the Lennard-Jones simulation program that you created in the last worksheet and add the Langevin thermostat. If you are not confident with your own simulation program or it wasn't complete, you can find the sample solution of the last worksheet in the archive file in the subdirectory `ljsim/`.
- Did you implement the function in the Python- or in the C-part of the program? Explain why you did it in the one and not the other part!
- Run simulations of $N = 1000$ LJ particles at density $\rho = 0.316$ and temperatures $T \in \{0.3, 1.0, 2.0\}$ using the Langevin thermostat ($\gamma = 0.5$).
- Determine the equilibrium mean values of the pressure P and the potential energy per particle E_{pot}.

3.2 Error Analysis of the Simple Observables

Task (4 points)

- Extend the program `ljanalyze.py` to include the error analysis functions.
- Analyze the errors of the equilibrium values of the observables that you have determined in the previous task.
- Extend the simulation runs such that the error of the observables is be less than 5%. Use any of the error estimates to determine the error.
- How many simulation steps do you need to achieve this error? What is the effective statistics of the sample?

3.3 Error Analysis of the RDF

Task (2 points)

- Analyze the errors of the equilibrium radial distribution function of the simulation runs in the previous task.
- Create a plot of the RDFs with corresponding error bars at the different temperatures.