MEASURING ERRORS IN COMPUTER SIMULATIONS

Katrina A. Magalotti and Marvin Bishop
Department of Mathematics/Computer Science
Manhattan College

Abstract

We have studied the errors involved in computer simulations by investigating the results of a two dimensional Brownian dynamics simulation of ideal linear polymers. The auto-correlation function of the mean-square radius of gyration is calculated using the random access capabilities of the C language. This function is used to determine the spacing needed between sampling to insure that the samples are statistically independent. The mean-square radius of gyration and its error are computed for different numbers of samples. This type of project is suitable for junior/senior majors in engineering, mathematics or science.

Introduction

Computer simulation is a standard tool [1] for studying molecular systems. Properties of these systems are determined by averaging over a set of generated configurations. However, computation of the average value of a property is not sufficient. An error estimate must be provided so that the accuracy of this average value can be assessed. Also it has been known from the earliest work in simulation that “it takes some time for a simulation model to 'warm up' “ [2]. Average values should only be computed after this equilibration period. The approach to equilibrium is observed by studying the time variation of the property of interest. In dynamic simulations the elementary time step, Δt, is the time step used to integrate the Newtonian differential equations of motion. Often, data for a subsequent analysis are saved at intervals of $i\Delta t$, where i is an integer.

Recently in this journal [3], Waldron and Bishop have reported on their Brownian dynamics simulations of ideal two dimensional linear polymers. In that article they focused on properties such as the mean-square radius of gyration, $\langle S^2 \rangle$, which measures the size of a polymer. They presented a time series for the square radius of gyration of a 284 unit linear polymer. There were two distinct time behaviors in their data. At first the square radius of gyration displayed transient effects as the polymer relaxed from an initial square configuration to a more typical random situation. Then the square radius of gyration displayed fluctuating values around an overall average. They found that this particular property needed about 250,000 time steps to ensure that subsequent configurations were in the equilibrium state.

After obtaining an additional 1000 equilibrium configurations, the mean-square value of the radius of gyration was computed by

$$\langle S^2 \rangle = \frac{1}{1000} \sum_{j=1}^{1000} S_j^2 \qquad (1)$$

where S_j^2 represents the square radius of gyration for the configuration at time $j(i\Delta t)$. The error in $\langle S^2 \rangle$ was computed by employing the usual [4] sample variance of the mean, σ^2.

$$\sigma^2 = \frac{1}{1000 * 999} \sum_{j=1}^{1000} [\, S_j^2 - \langle S^2 \rangle \,]^2 \qquad (2)$$

However, Equation 2 only applies if the data are uncorrelated and that is often not true in

computer simulations. In an attempt to avoid the correlation effects between adjacent samples Waldron and Bishop [3] saved data every i = 25,000 steps in their calculations of the mean-square radius of gyration for chains with 64, 128, 195 and 284 units. In this article their study is extended to examine the impact of the sample size on the various averages and standard deviations of the mean. Moreover, the effects of sample correlation are investigated by computing the time auto-correlation function of the data.

Data will be uncorrelated if the time auto-correlation function becomes essentially zero within the sampling spacing. The time auto-correlation function of any time-dependent property A(t) is defined [1] as <A(t)A(0)>. Here A(0) means that the quantity A is sampled at the time origin and A(t) means that it is sampled after a delay time t. However, one usually calculates the normalized auto-correlation function, $\psi(t)$, given by

$$\psi(t) = \frac{\langle A(t)A(0)\rangle - \langle A(t)\rangle^2}{\langle A^2(t)\rangle - \langle A(t)\rangle^2} \qquad (3)$$

Equation 3 is normalized since as $t \rightarrow 0$, $\langle A(t)A(0)\rangle \rightarrow \langle A^2(0)\rangle$ and the ratio will become equal to one. This means that before a delay happens, A(0) is already completely correlated with itself. However, as $t \rightarrow \infty$, $\langle A(t)A(0)\rangle \rightarrow \langle A(\infty)\rangle > \langle A(0)\rangle$ since the values become uncorrelated for long delay times. The average value of A becomes independent of time as long as the same number of samples are used. Then the numerator becomes zero. Thus, $\psi(t)$ decays from a maximum value of one at t = 0 to a value of zero at long times. It can also become negative, indicating anti-correlation. Often $\psi(t)$ has an exponential form, $\psi(t) = \exp(-t / \tau)$, where τ is a constant.

Equation 3 can be recast into three discrete terms suitable for programming on a computer. The first term in the numerator becomes

$$\langle A(t)A(0)\rangle = \langle A(j)A(0)\rangle = \frac{1}{NORIG}\sum_{k=1}^{NORIG} A(j+k)A(k) \qquad (4)$$

Here, NORIG is the number of time origins the data is averaged over. This value is typically about 2000. The other two terms are determined as standard averages given by Equation 1.

Method

The C program developed by Waldron and Bishop [3] to perform their simulations was employed to generate the different data samples, .i.e. the values of S^2 used as input to an auto-correlation function program. The auto-correlation function program was also written in C. It uses the random access capabilities of the C language. The data, of type double, are placed into a random access file with the fwrite command:

```
fwrite(&s2,sizeof(double),1,dumpFile)        (5)
```

Here &s2 is the address of an S^2 data item, the sizeof operator returns the number of bytes in an item of type double for the computer hardware in use, 1 is the number of items being stored in the data file with this write operation and dumpFile is a pointer to the data file which will contain the item. Similarly, data items are fetched from a random access file by employing the fread command:

```
fread(&s2,sizeof(double),1,dumpFile)         (6)
```

Evaluation of Eq. 4 requires two different positionings of the file pointer in order to fetch the needed data with a fread command.. The C language provides the fseek command to move a file pointer to different locations:

```
fseek(dumpFile, j * sizeof(double), SEEK_SET)
                                             (7)
```

Here, the dumpFile pointer is moved forward j times the number of bytes in a double, starting from the beginning of the file. The macro SEEK_SET indicates this starting address. Thus, the file pointer is moved by the byte count. The two separately fetched data items are then multiplied and their product added to a running sum.

Results

Figure 1 presents the auto-correlation function for the radius of gyration of a 284 unit chain. The data were sampled every 2,500 time steps after discarding the first 250,000 time steps. NORIG was set at 1800 for the averaging. It is clear from this figure that the radius of gyration auto-correlation function needs about 150 steps (in these units) to decay to zero; e.g. a total of 375,000 Δt steps. The study of Waldron and Bishop [3] employed a spacing of 25,000 steps which is too small for the sample data to be fully uncorrelated.

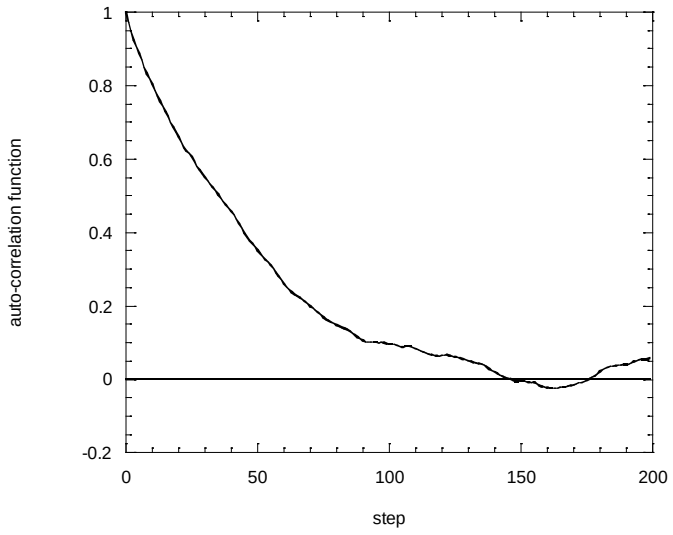

Figure 1: The auto-correlation function.

The auto-correlation function is of exponential form because its natural logarithm is linear in the number of steps as shown in Figure 2. The dotted line is the linear fit to this function, $\ln(\psi) = -0.019077 * \text{step} - 0.024912$.

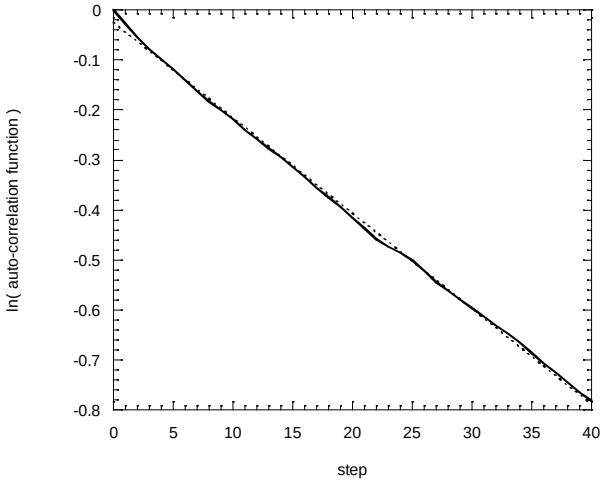

Figure 2: The natural logarithm of the function.

We have repeated Waldron and Bishops simulations for 64, 128, 195 and 284 unit chains but used a spacing of 375,000 steps and varied the number of data samples generated from 100 to 400 to 1600. Table I presents the simulation results for all the systems studied. The number in parenthesis denotes one standard deviation in the last displayed digits. The 95% confidence interval is about twice this value.

The first column contains the original findings of Waldron and Bishop [3] using 1000 samples with a spacing of 25,000 whereas data in the other three columns employed a spacing of

Table I: The Simulation Data.

N	$<S^2>$ 1000	$<S^2>$ 100	$<S^2>$ 400	$<S^2>$ 1600
64	10.92(22)	9.91(56)	10.30(28)	10.56(16)
128	23.58(45)	24.54(1.46)	22.81(72)	22.62(35)
195	31.80(61)	30.04(1.63)	32.72(1.11)	32.85(53)
284	47.64(97)	46.07(2.93)	47.61(1.53)	47.41(76)

375,000. It is clear from this data that Waldron and Bishops 1000 samples at a spacing of 25,000 give statistically the same results, within the 95% confidence interval, as our runs with 1600 samples and a spacing of 375,000. Hence, the correlation effects are not statistically significant. Note also that the standard deviation of the mean decreases approximately as the square root of the number of samples; e.g. when N = 284 the error decreases from 1.53 for 400 samples to 0.76 for 1600 samples. This finding is expected for independent samples.

Conclusion

Brownian dynamics has been used to generate two dimensional linear polymer configurations and to study the errors entailed in computing the mean-square radius of gyration. Correlation effects have been examined by computing the auto-correlation function. It is found that there needs to be a large spacing in data sampling in order for the data to be completely statistically independent. However, 1000 samples are sufficient to accurately determine the mean-square radius of gyration. These types of simulations provide interesting projects in which students can get experience in computational science. This will be very useful in their future careers.

Appendix: The Manhattan College Undergraduate Research Program

Manhattan College has a long tradition of involving undergraduates in research and was one of the original members of the Oberlin 50. This is a group of undergraduate institutions whose students have produced many Ph.D.'s in engineering and science. At Manhattan College, students can elect to take an independent study course for 3 credits during the academic year. In addition, the College provides grant support to the students for 10 weeks of work during the summer. I have personally recruited the students from my junior level course in Systems Programming. Previously published articles in this journal by Manhattan College student co-authors are a very effective recruitment tool. The students have also presented their results at a variety of undergraduate research conferences including the Hudson River Undergraduate Mathematics Conference and the Spuyten Duyvil Undergraduate Mathematics Conference.

Acknowledgements

We wish to thank Professor Paula A. Whitlock for helpful comments on the manuscript.

References

1. M.P. Allen and D.J. Tildesley, "Computer Simulation of Liquids", (Clarendon Press, Oxford, 1987).

2. R.W. Conway, Manage. Sci., 10, 47 (1963).

3. M. Waldron and M. Bishop, "Modeling and Simulation of Two Dimensional Ideal Linear and Ring Polymers with Brownian Dynamics", Comp. Educ . J., 3, no. 3, 2 (2012).

4. P.R. Bevington, "Data Reduction and Error Analysis for the Physical Sciences" (McGraw-Hill, New York, 1969).

Biographical Information

Katrina A. Magalotti is an undergraduate student in the computer engineering program at Manhattan College. She will complete a B.S. in computer engineering at Manhattan College in 2013.

Marvin Bishop is a Professor in the Department of Mathematics and Computer Science at Manhattan College. He received his Ph.D. from Columbia University, his M.S. from New York University and his B.S. from the City College of New York. His research interests include simulation and parallel processing.