

One-Dimensional Weighted-Ensemble Brownian Dynamics Simulation in Perl

Amanda J. McCoy

Harvard University

Mentor: Daniel M. Zuckerman, Ph.D.

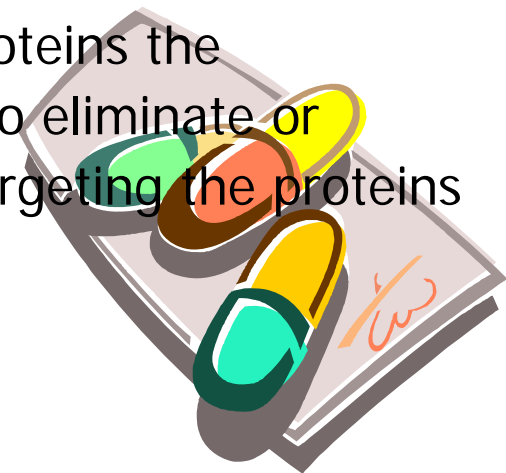
University of Pittsburgh School of Medicine
Department of Computational Biology





Introduction

- Pharmaceuticals that cure illness, treat disease, and improve quality of life are the byproducts of our greater understanding of how the human body functions.
- In the last century, our knowledge of human anatomy and physiology has penetrated beyond the level of organ systems, organs, differentiated tissue cells, organelles, and nucleic acids to the very molecules of functionality themselves, proteins.
- It is through our better understanding of proteins the pharmaceutical companies have been able to eliminate or facilitate specific reactions in the body by targeting the proteins that act as their catalysts.

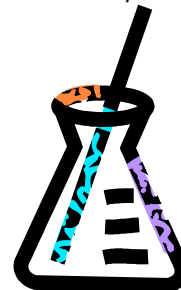


Background

- It is necessary to know a protein's active state to develop a drug that will act as a powerful inhibitor of protein function
 - It is often difficult to ascertain the structure of a protein in its active form. This is because proteins remain in their high energy, active conformations for too fleeting of time to obtain laboratory data through X-Ray crystallography or NMR
- An alternative for obtaining the structures of high-energy protein conformations is through dynamics simulations
 - Unfortunately, many dynamics algorithms are comprised of too many degrees of freedom to run a simulation long enough to observe a change in protein conformation within a reasonable amount of computing time
- The Weighted-Ensemble Brownian (WEB) Dynamics method proposed by Huber and Kim surpasses other algorithms in its ability to calculate the same length of simulation as other methods, but at only a fraction of the time
- To fully understand the applicability of the WEB method, one must try it out for himself → My project



versus



Current Research

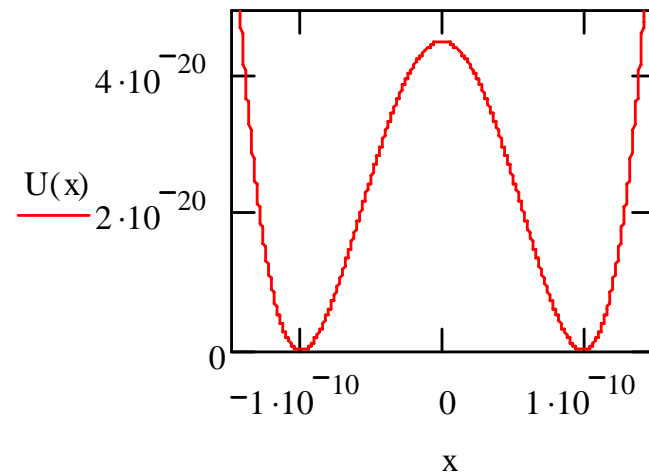
- My research goal was to program a small-scale WEB Dynamics simulation in Perl and analyze the results
 - Quite the feat because I have never programmed before in my life
- The System
 - One-Dimensional
 - One Energy Barrier

$$U(x) = E_b \left[\left(\frac{x}{d} \right)^2 - 1 \right]^2$$

where

$$d = 1 \text{ \AA}$$

$$E_b = 11 k_B T$$



Methods

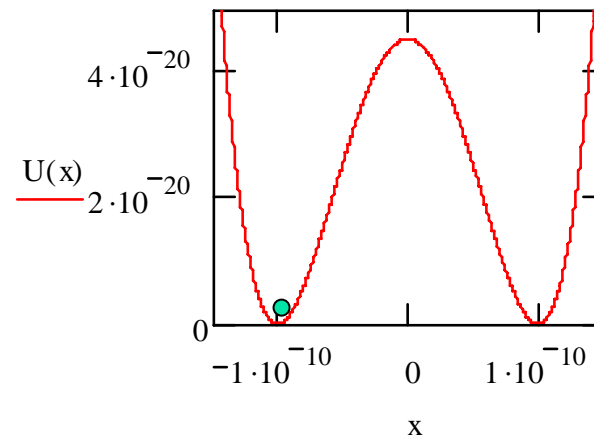
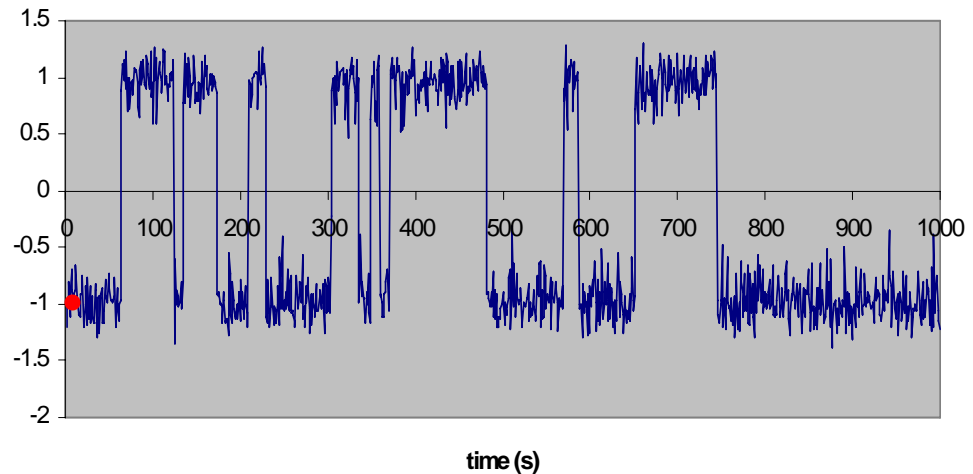
Standard Brownian Algorithm

$$x_{i+1} = x_i + F_i \times (m\gamma)^{-1} \times \delta t + X_i(\delta t)$$

where:

$$m = \frac{1k_B T s^2}{\text{\AA}^2} \quad \delta t = .001s \quad \gamma = 1s^{-1}$$

$$F_i = \left. \frac{-dU}{dx} \right|_{x_i} \quad X_i(\delta t) = \pm \sqrt{\frac{2k_B T \delta t}{m\gamma}}$$





Methods

- Run each particle through 100 iterations of a standard Brownian algorithm
 - Sort particles into proper bins
 - Modify (split/combine) particles so that each bin with any particles in it (active bin) will have 10 evenly-weighted particles in it after modification
 - Modify the particles' weights in each active bin to ensure that the total weight of particles in the bin before modification is the same as the total weight in the bin after modification.
- Repeat this process 1000 times

$$\frac{100 \text{ iterations}}{1 \text{ modification event}} \times 1,000 \text{ modification events} = 100,000 \text{ iterations}$$

$$100,000 \text{ iterations} \times \frac{.001 \text{ second}}{1 \text{ iterations}} = 100 \text{ second simulation}$$

Methods

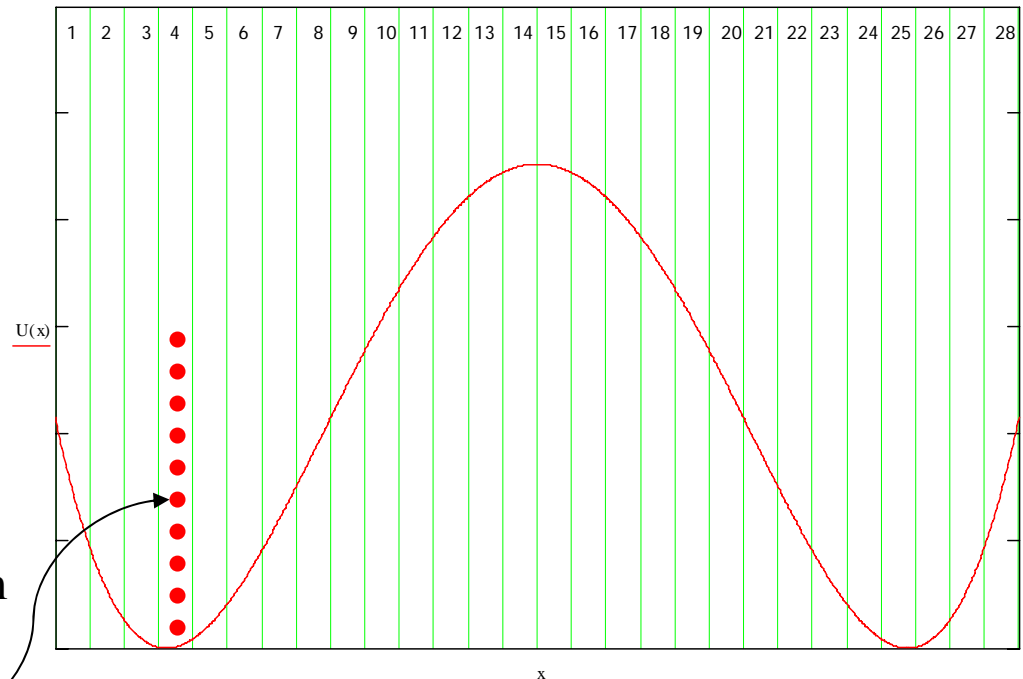
Initial System:

- 28 bins
- 10 evenly-weighted particles in bin 4
- 0 particles in other bins
- Bin division should follow particle movement

$$\sum_{i=1}^N w_i = 1$$

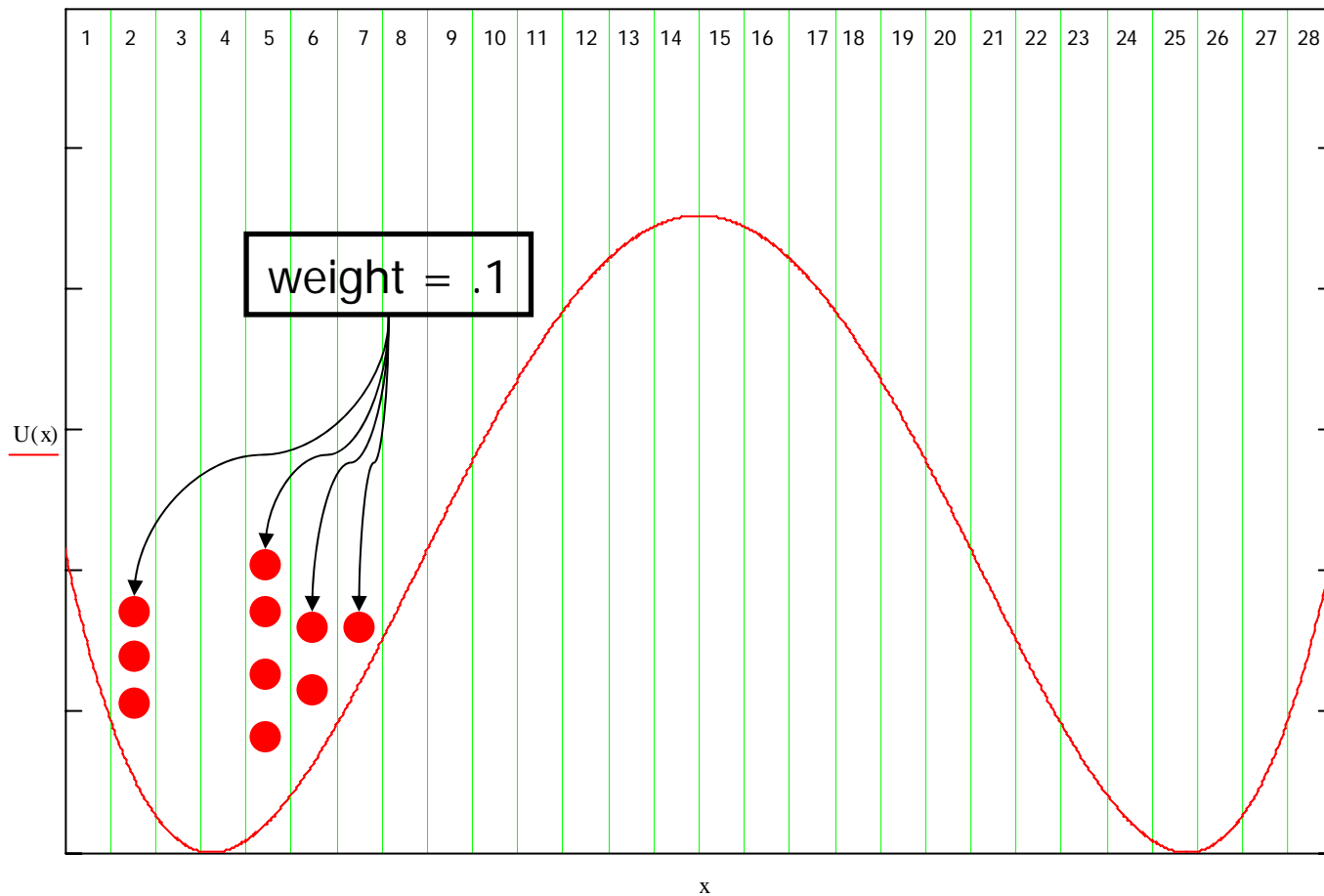
N = total number of particles in the system

position = -1
weight = .1



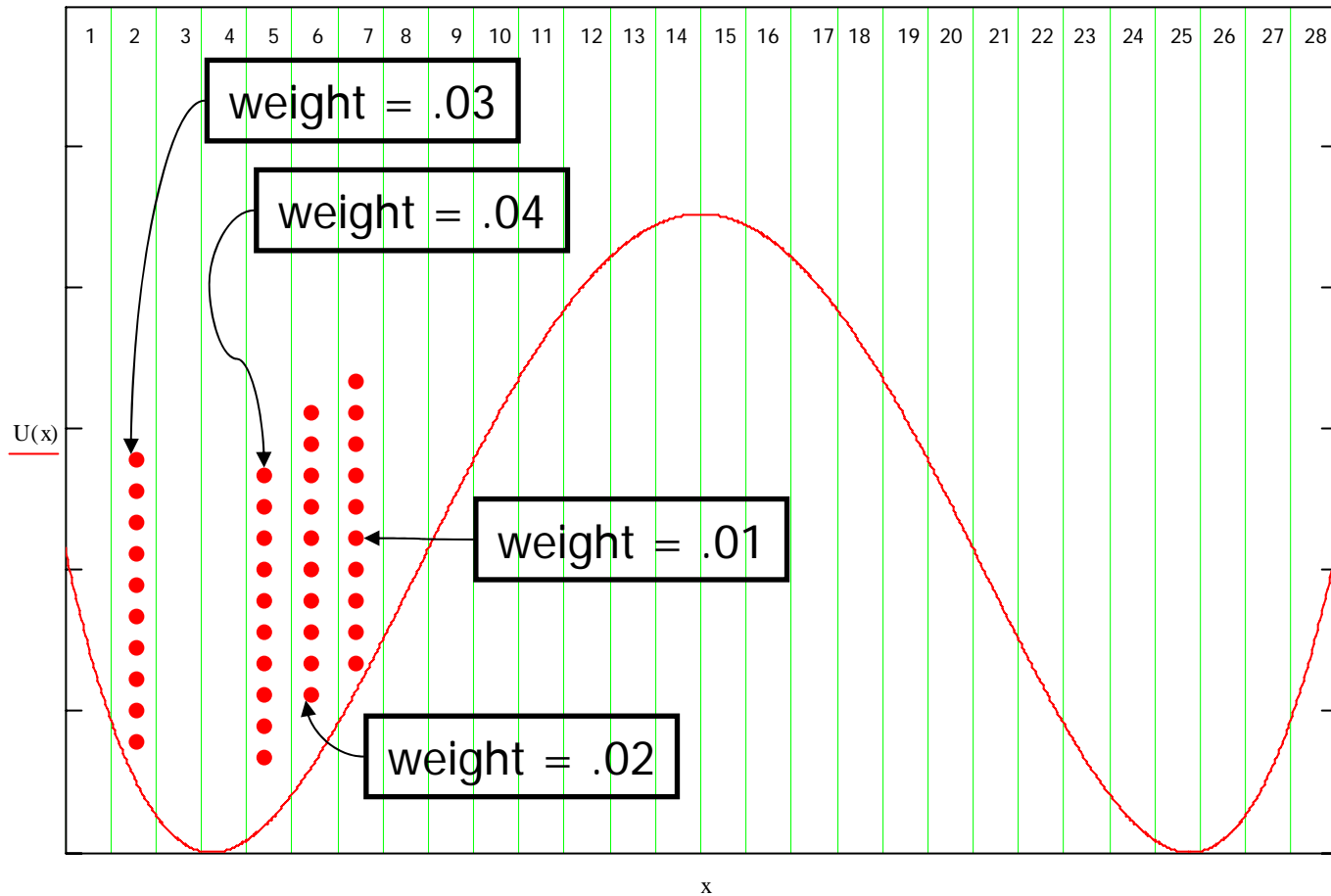
Methods

First Modification Event (100 iterations {.1 seconds})



Methods

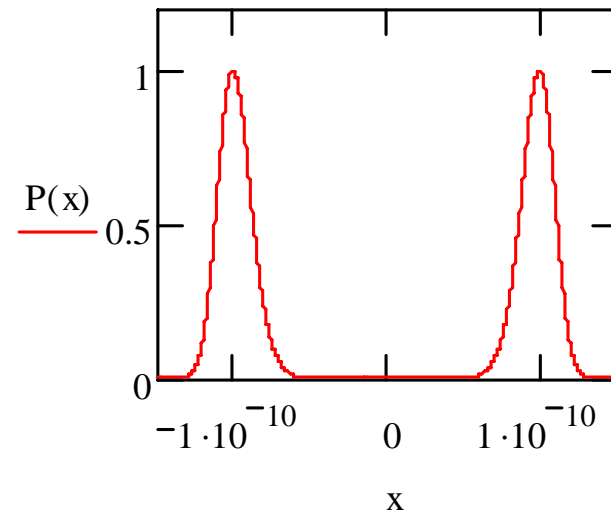
After Modification



Expected Results

- The Distribution of bin weight should follow the Boltzmann Distribution
 - Remember that the total weight of all the particles at all times is 1, which represents a probability distribution by bin

$$P(x) = e^{\frac{-U(x)}{k_B T}}$$



To normalize:

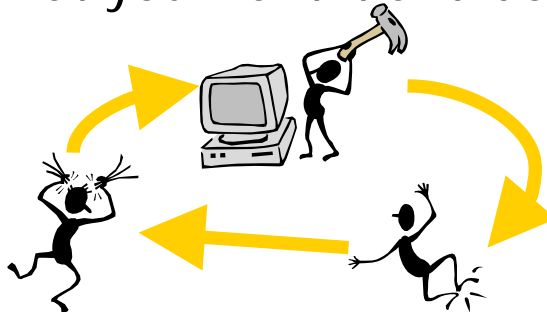
$$P(x) = \frac{e^{\frac{-U(x)}{k_B T}}}{Z}$$

Partition Function:

$$Z = \int_{-\infty}^{+\infty} e^{\frac{-U(x)}{k_B T}}$$

Hurdles Along the Way

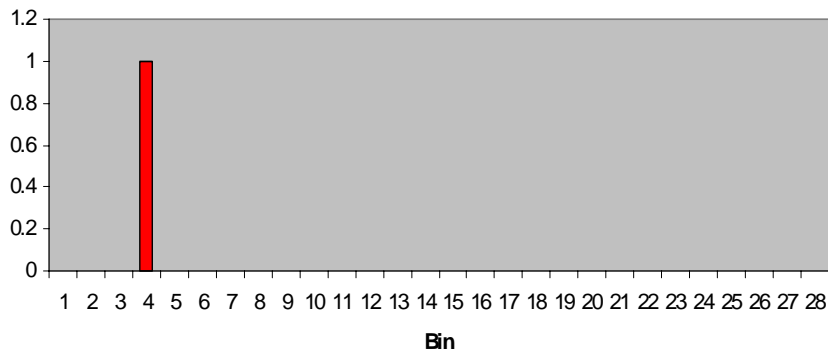
- I first tried a two-dimensional array, but had problem reassigning new values to the arrays representing the particles
 - I then revised my approach to use two one dimensional arrays to define my particles
- Sorting the particles was surprisingly easier than I expected; the only difficulty I had was indexing the particles so that the weight and position of each particle maintained correlation
- I found splitting particle to ensure each bin that had any particles in it had 10 particles in much easier than combining particles while conserving probability
- I learned that you cannot expect Perl to do anything, you have to tell it exactly what you want it and don't want it to do



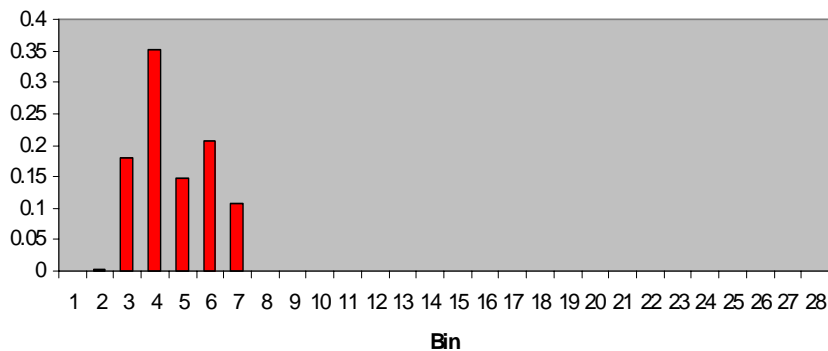
Results

WEB

Bin Weights at $t = 0s$

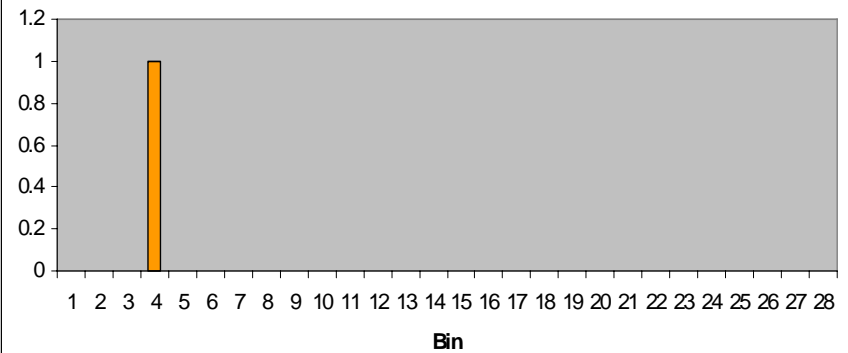


Bin Weights at $t = 5s$

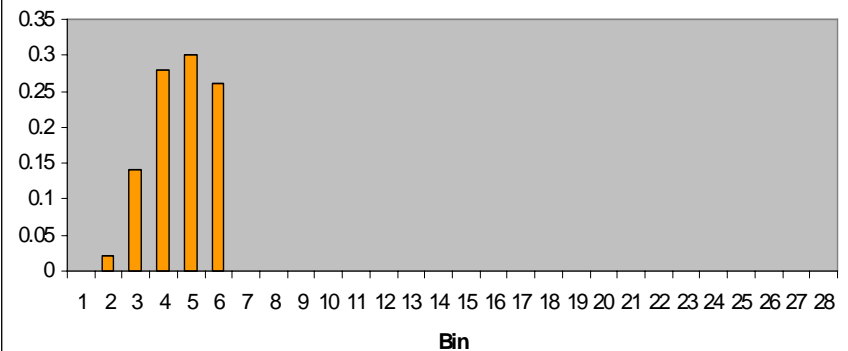


Standard Brownian

Bin Weights at $t = 0s$



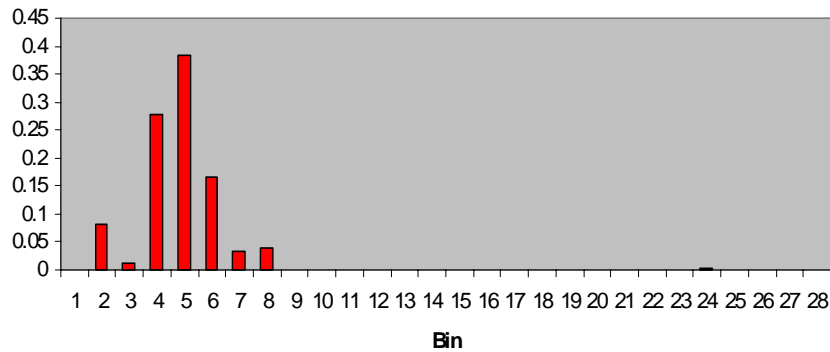
Bin Weights at $t = 5s$



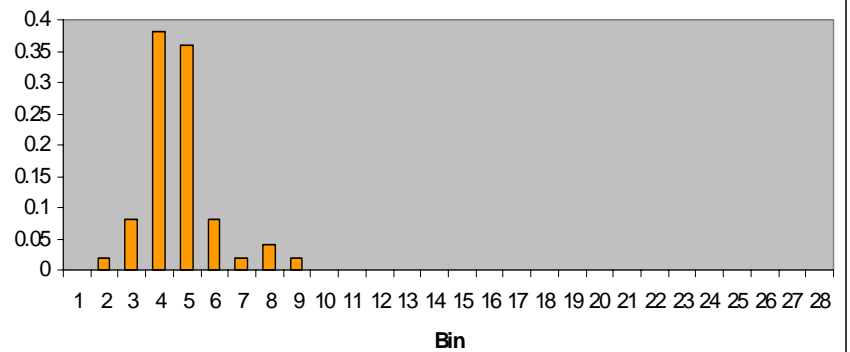
Results

WEB

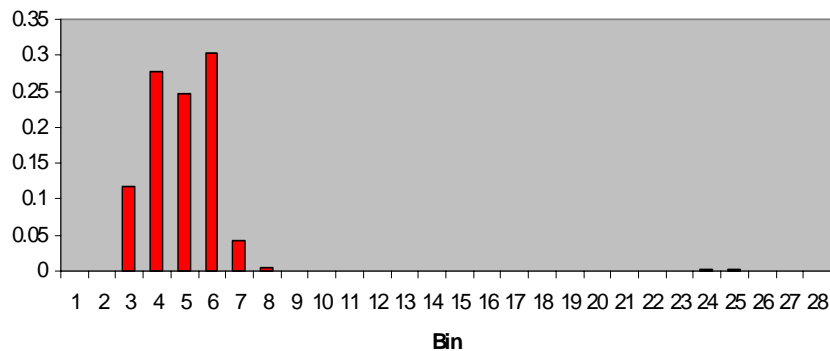
Bin Weights at $t = 10s$



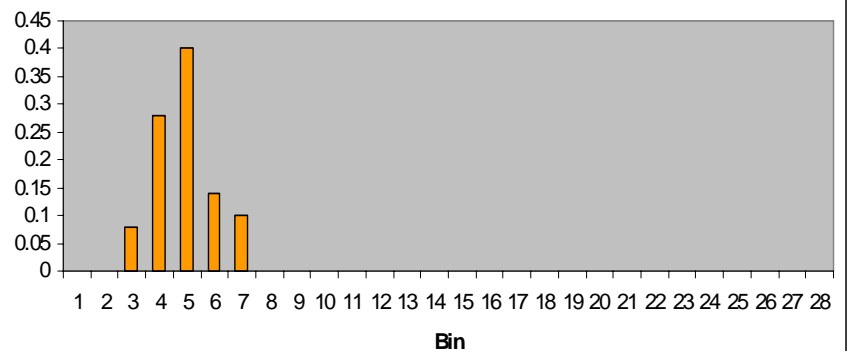
Bin Weights at $t = 10s$



Bin Weights at $t = 15s$



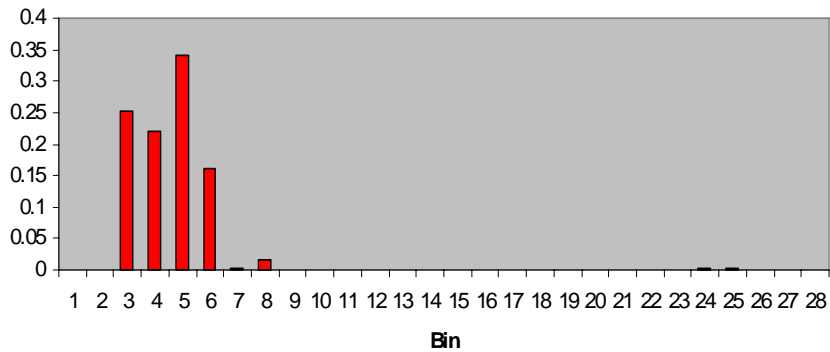
Bin Weights at $t = 15s$



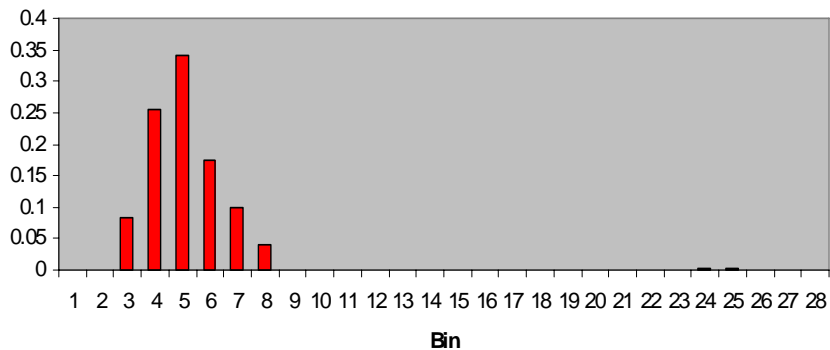
Results

WEB

Bin Weights at $t = 20s$

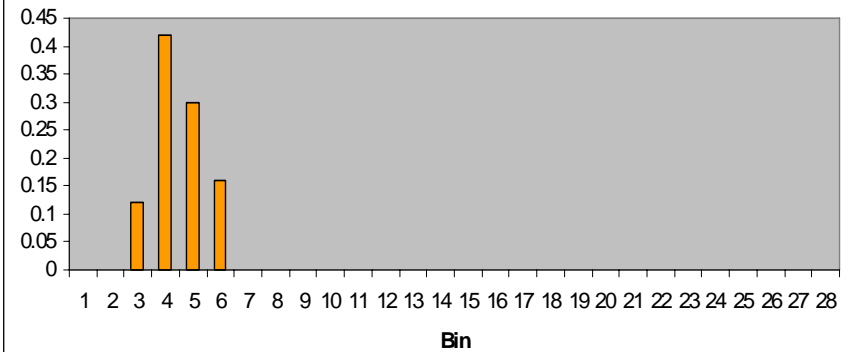


Bin Weights at $t = 40s$

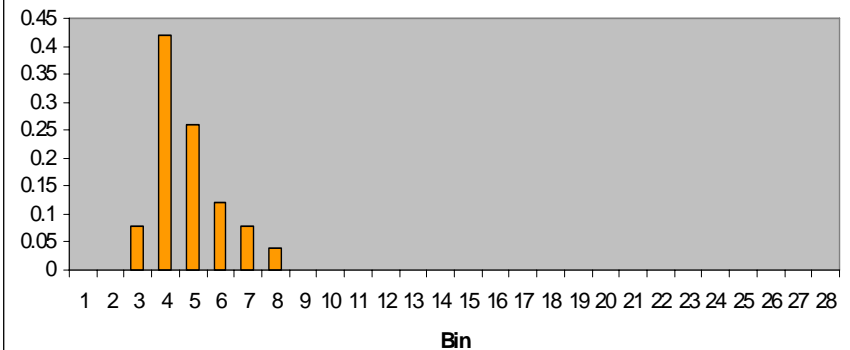


Standard Brownian

Bin Weights at $t = 20s$



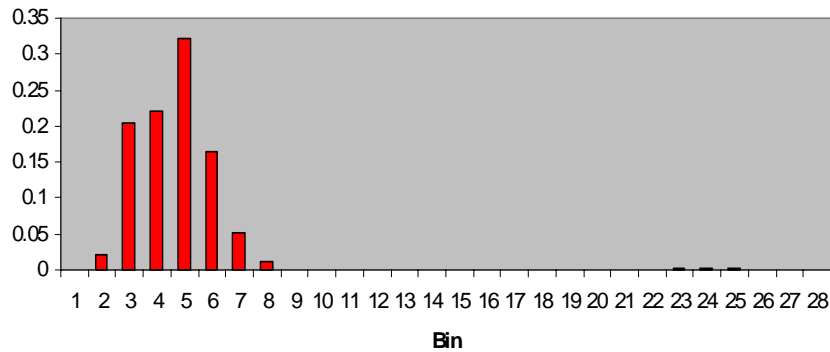
Bin Weights at $t = 40s$



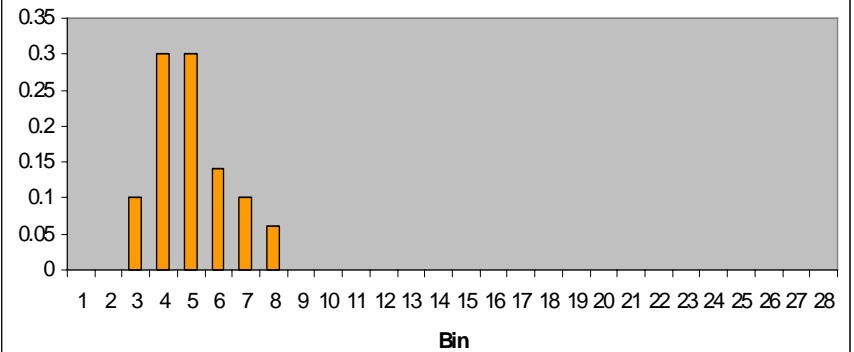
Results

WEB

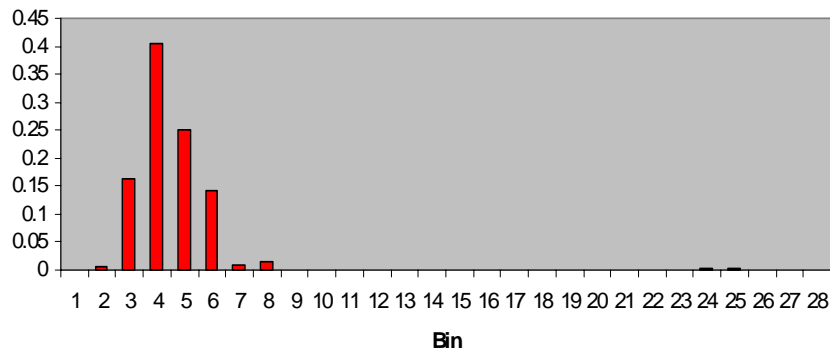
Bin Weights at $t = 60s$



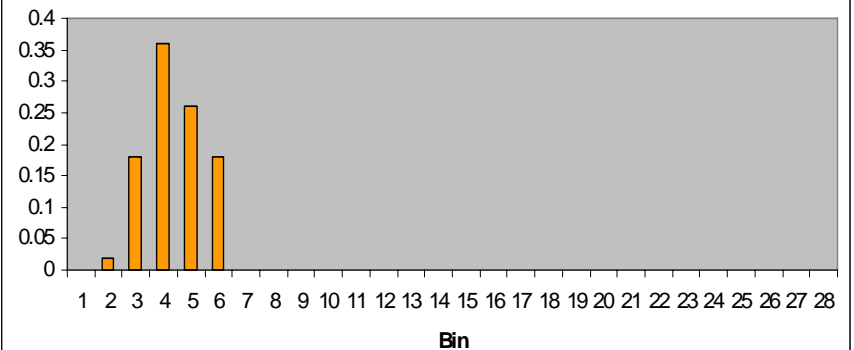
Bin Weights at $t = 60s$



Bin Weights at $t = 80s$



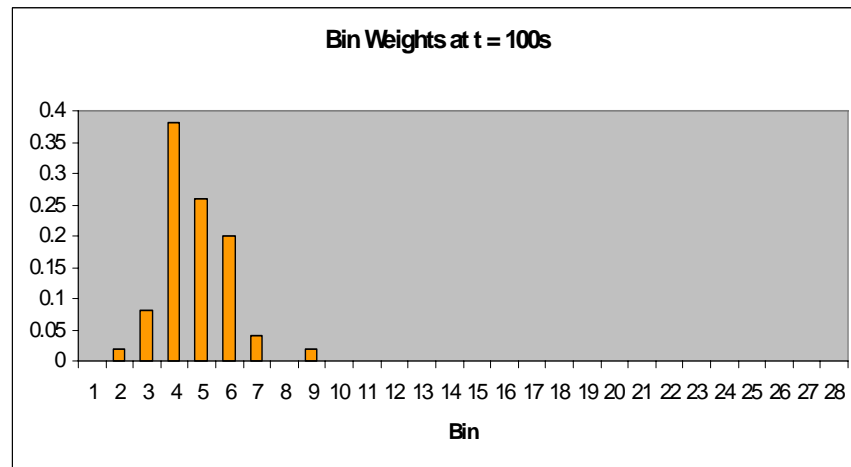
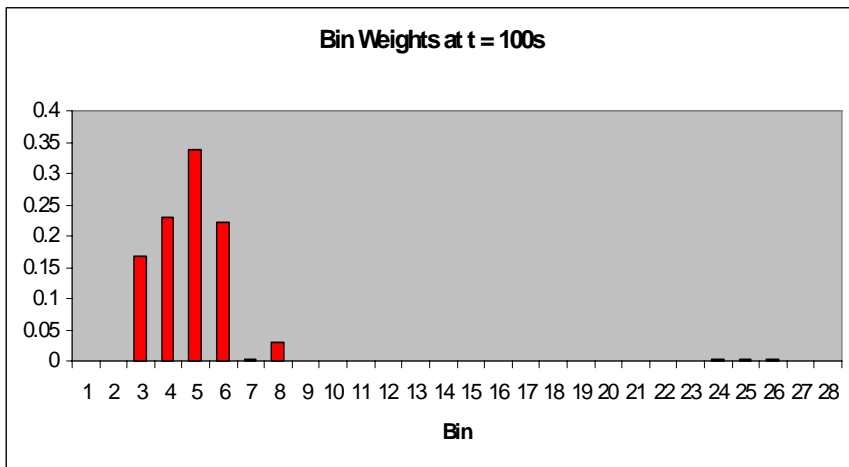
Bin Weights at $t = 80s$



Results

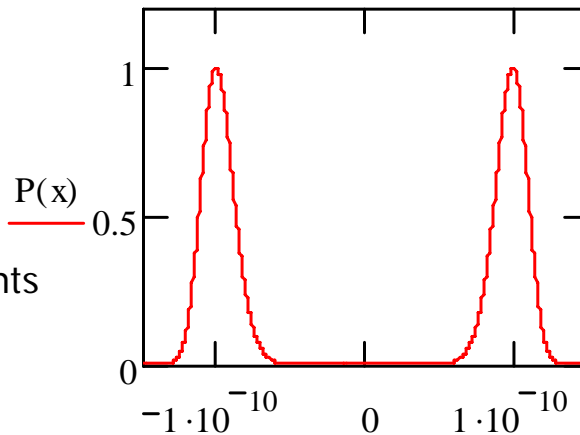
WEB

Standard Brownian



As $t \rightarrow \infty$

9 Energy Barrier Crossing Events



As $t \rightarrow \infty$

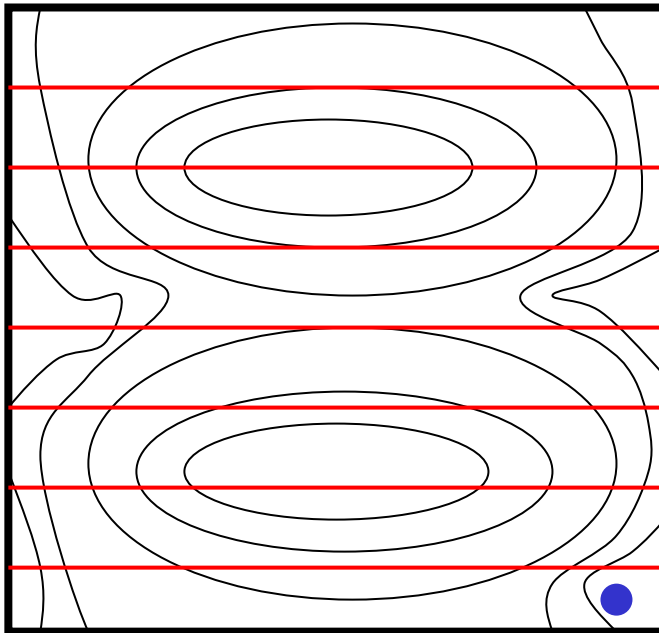
0 Energy Barrier Crossing Events



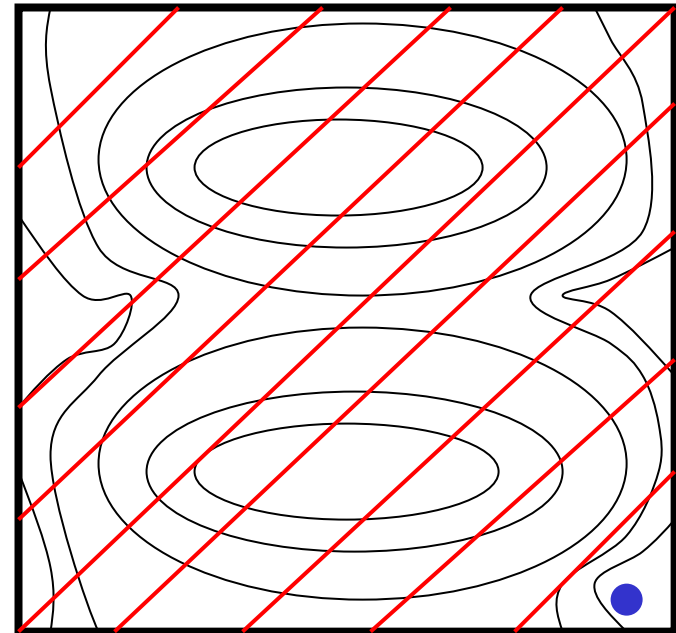
Continuing Work

The WEB method can be expanded far beyond 1-D systems

2-D Twin Mountain Peak Energy Coordinate



Bottle Necking



Bins follow particle movement more directly



Conclusions

- The WEB method yields the same results as the standard Brownian method if permitted to run indefinitely → approaching Boltzmann Distribution
- The WEB method allows more particles to cross the energy barrier earlier on in the simulation → less computing time to determine a change in equilibrium
- Possible to use this method when we don't know the energy coordinate
 - If we know the forces acting on a given protein, we can still run a WEB dynamics simulation
 - From the resulting probability distribution, we can determine the likelihood of a protein to remain in any given energy state
 - Much more cost effective and efficient than using NMR and X-Ray crystallography to isolate the numerous equilibrium structure of any given protein





References

Huber, Gary A. and Sangtae Kim. 1996. "Weighted-Ensemble Brownian Dynamics Simulations for Protein Association Reactions." *Biophysical Journal*. Vol 70: 97-110.

Leach, Andrew R. *Molecular Modelling Principles and Applications*. 2nd ed. New York: Prentice Hall, 2001.

Northrup, S. H. and H. P. Erickson. 1992. "Kinetics of a Protein-Protein Association Explained by Brownian Dynamics Computer Simulation." *Proc. Nat. Acad. Sci. USA*. Vol 89: 3338-3342.

Northrup, S.H. and J. A. McCammon. 1980. "Efficient Trajectory Simulation Methods for Diffusional Barrier Crossing Processes." *Journal of Chemical Physics*. Vol 72: 4569-4578.

Reif, F. *Fundamentals of Statistical and Thermal Physics*. San Francisco: McGraw-Hill. 560-573.



Acknowledgements

I would like to thank the following people for answering my questions regarding physics, computer programming, and just information in general. They also deserve a round of applause for their patience with me while answering all these questions.

- Dan Zuckerman, Ph.D.
- Ed Lyman, Ph.D.
- Bin Zhang
- Marty Ytreberg, Ph.D.
- Jason Boles, MSEE
- Judy Wieber, Ph.D.