CHAPTER 3
POISSON PROCESS APPROACH TO STATISTICAL MECHANICS

Contents
1. Introduction
2. Diffusion
   2.1. One dimensional diffusion
   2.2. Two dimensional diffusion and recovery of energy landscape
   2.3. Three dimensional diffusion and chemical reactions
3. Inter-conversion of states of a molecules
4. Photon statistics from lasers
5. Defects and deposition on a surface
6. Location of stars in the sky
7. Location of cosmic ray sources
8. Conclusion
9. Acknowledgment
Basic hypothesis:
Every random natural process can be approximated either as a single Poisson process (i.e., has constant probability of happening in space or time) leading to an exponential distribution of separation between consecutive events or as a chain of Poisson processes leading to a gamma distribution of separation between consecutive events with a possibility of underlying structure due to interactions.

Experimental consequence:
Distribution of times between events is an exponential for a single Poisson process and a gamma distribution for a chain of Poisson processes. However, distributions of events in a user defined blocks of time or space can have wide range of shapes including exponential, Poisson distribution, Gaussian distribution, geometric distribution, and so on. Underlying structures or clustering due to interactions lead to different Poisson processes in different regions of space or time.

1. Introduction
Random processes lead to fluctuations in the measured quantities and force us to treat a system of interest statistically. In a sense, statistical mechanics is an attempt to get a molecular understanding of the well-established empirical thermodynamic results. It is considered an attempt because we still have not resolved issues with the arrow of time arising from reversible mechanics, the ergodicity, and the inherent non-equilibrium nature of biology. In addition, transient violations of second law of thermodynamics for single particles have been experimentally confirmed, which is not surprising but the implications in the context of biology and electronics are not clear yet. There have been successful attempts to calculate equilibrium quantities from non-equilibrium experiments such as Jarzynski equality and various fluctuation theorems (for example, Crooks fluctuation theorem). The ensemble approach to statistical mechanics has not made much progress despite the necessity. One possible approach to treat non-equilibrium systems with no restriction on the number of molecules is to consider a system of interest as a sum of various Poisson processes. A random process is Poisson process if there is a constant probability of that process either in space or in time. Definite order or structure due to various interactions can then be imposed, if needed, to mimic the experimental observations.

Random processes are ubiquitous in nature. Almost every random process in nature can be considered as a single Poisson process or a chain of Poisson processes with a rate that can be
measured. For example, chocolate chips in cookies, location of stars in the sky, titration of two chemicals, radioactive decay, molecular diffusion, photons from a laser, defects in materials, chemical reactions, and so on can be simulated by considering them as a Poisson process. Therefore, a significant amount of statistical properties about a system can be derived from Poisson process approach and can be applied to single molecules, small systems comprising few molecules, and large systems comprising many molecules. This is related, but in contrast to the ensemble approach to statistical mechanics where one considers a large ensemble of molecules that are generally divided into three categories – microcanonical, canonical, and grand canonical, and applicable to equilibrium or very close to equilibrium. However, natural processes are mostly in non-equilibrium, in particular, biological processes are inherently far away from equilibrium.

To show the broad applicability of the Poisson process approach to statistical mechanics, we will consider various examples from different field of studies. We will study their statistical properties and show how they could be understood within the basic Poisson process framework.

2. Diffusion

Noisy diffusion hides the underlying interactions. Simulations are important to verify the proposed model of interactions and validate the analysis methods.

Molecules or a small system of molecules randomly collide with surrounding molecules all the time. If someone experimentally observes the resulting random walk or diffusion of such particles, one can gain insights into the underlying interaction landscape and into the functional roles of such a molecule. Diffusion is a Poisson process that can be easily explained for one dimension. At each time step, the diffusing particle in 1D has a constant probability (50%) of moving to the right or the left (i.e. binomial due to the two possibilities). The concepts can be extended to relative random motion of a part of the same molecule such as domain motion of proteins. Theoretically, diffusion has been studied more than 100 years and there are excellent sources to get ideas about theoretical treatment of diffusion (Perrin[1], Einstein[2], Smoluchowski[3], and Crank [4]). Experimentally, one needs to decide what system to work on (personal interest), how frequently (time resolution) and accurately (localization accuracy) the molecules of interest will be located, what are the potential sources of randomness other than the diffusion, how to analyze the data, and how to validate the analysis. For observation of diffusion with a time resolution slower than 1 ms, TIRF is a good choice because of high throughput of the data[5-7], for faster than 1 ms, fluorescence
correlation spectroscopy (FCS) [8, 9] is a suitable choice, and for even faster time scales nuclear magnetic resonance (NMR) can be used[10, 11]. While NMR provides the best localization accuracy, for tracking labeled biomolecules or nanoparticles fluorescence-based method such as FCS and TIRF can be used for localization accuracy of as low as 1 nm [12-14]. For normal diffusion, the mean squared displacement (MSD) is given by, \[ \langle [r(t) - r(0)]^2 \rangle = 2dDt \]; the probability distribution of positions at time \( t \) is given by \[ P(r,t) = \left( \frac{1}{\sqrt{2d\pi D t}} \right) \exp\left( -\frac{r^2}{2dDt} \right) \], where \( d \) is the dimension. In real life, however, normal diffusion becomes anomalous due to either hindrance (e.g. collagen degradation[15], binding sites, confined region[16, 17], and even due to data analysis [18]) or bias (e.g. molecular motors[15, 19, 20]). For anomalous subdiffusion, the mean squared displacement (MSD) is given by, \[ \langle [r(t) - r(0)]^2 \rangle = 2dDt^\gamma \]; this type of behavior has been well documented in cell membranes and has resulted in improved models of intracellular transport [21]. Careful analysis of diffusion under these conditions will show both simple diffusion and subdiffusive behavior on various scales determined by the spacing of binding or hindrance. Another biologically important type of diffusion is corralled diffusion which occurs when the diffusing particles are confined to a region[22]. For corralled diffusion, the MSD is given by, \[ \langle [r(t) - r(0)]^2 \rangle \equiv \left( r_c^2 \right) \left( 1 - A_1 \exp\left( -2dA_2 D t \right) / \langle r_c^2 \rangle \right) \], where \( A_1 \) and \( A_2 \) are determined by the geometry of the corral and \( \langle r_c^2 \rangle \) is the corral area. For diffusion in a flow, the MSD is given by, \[ \langle [r(t) - r(0)]^2 \rangle = 2dDt + (V t)^2 \].

Simulation of diffusion
At each time step \( \Delta t \), the step size \( \delta \) can be determined from a Gaussian distribution, \[ P(\delta) = \frac{1}{\sqrt{4\pi D \Delta t}} \exp\left( -\delta^2 / 4D\Delta t \right) \]. Locations of the diffusive particles can be determined, \((t_1, x_1) = (0, 0), (t_2, x_2 + \Delta t, x_2 + \delta_1), \ldots, (t_n, x_n + n\Delta t, x_n + \delta_1 + \delta_2 + \ldots \delta_n) \). Experimental uncertainty \( \sigma \), assuming it is Gaussian distributed noise, can be added to each simulated locations using \( \text{normrnd}(0, \sigma) \). The MSD can be calculated using the successive positions of the diffusing particle, \((t_1, x_1), (t_2, x_2), (t_3, x_3), \ldots, (t_n, x_n) \), using the recursive relation:
\[ MSD(m\Delta t) = \frac{1}{n-m} \sum_{k=1}^{n-m} (x_{k+m} - x_k)^2 \]

After one time step, \( \Delta t \), the MSD is:
\[ MSD(\Delta t) = \frac{1}{n-1} \left[ (x_2 - x_1)^2 + (x_3 - x_2)^2 + (x_4 - x_3)^2 + \ldots + (x_n - x_{n-1})^2 \right] \]

After two time steps, \( 2\Delta t \), the MSD is:
\[ MSD(2\Delta t) = \frac{1}{n-2} \left[ (x_3 - x_1)^2 + (x_4 - x_2)^2 + \ldots + (x_n - x_{n-2})^2 \right] \]

and so on.

MSD as a function of \( (m\Delta t) \) is given by,
\[ MSD(m\Delta t) = 2D \cdot m\Delta t + 2\sigma^2 \]

Note that the y-intercept of the MSD is twice the variance, \( 2\sigma^2 \), of the noise present in the data.

**Figure 1** shows simulated normal, biased, and hindered diffusion with a common underlying diffusion constant, \( D = 5 \mu m^2 / s \). Significance of simulation and validation of analysis can be exemplified by comparing the normal and the hindered diffusion. Hindrance was simulated by considering that the diffusing particle binds the substrate at specific places with defined rates. This particular type of hindrance is not apparent from the MSD for hindered diffusion in **Figure 1** that can be fitted well with a linear model for normal diffusion. In other words, both the normal and the hindered diffusion have the same underlying \( D = 5 \mu m^2 / s \), but MSD calculation or fluorescence correlation spectroscopy (FCS) measurements cannot distinguish the two cases. However, it is possible to distinguish these two cases by further analysis.
2.2. Two dimensional (2D) diffusion and recovery of energy landscape

Two dimensional diffusion is important in many scientific problems such as protein diffusion on membranes, DNA/RNA, and matrix metalloprotease diffusion on collagen fibrils. As an example, 2D diffusion has been simulated with \( D = 7 \mu m^2 / s \) and \( \Delta t = 0.1 ms \). Diffusion itself is noisy in character and therefore, enough data should be collected experimentally and simulated to verify and validate scientific conclusion. The simulation for a single particle was 0.3 s long and was repeated for 15000 particles. Diffusing particles can interact and bind at specific locations, a grid of \( 21 \times 21 \) binding sites with 50 nm spacing (Figure 2A). If the diffusing particle reaches within a diameter of 5 nm around a specific binding site, it binds with 100% probability. The particles can come off the site, a Poisson process with rate \( 100 s^{-1} \), and resume diffusion, another Poisson process.

![Figure 2](image)

**Figure 2. Simulation and analysis of 2D diffusion on an interaction landscape.** A) Two dimensional interaction landscape with specific sites every 50 nm. B) Simulated position histogram of diffusion of 15000 particles on the 2D landscape in Figure 2A. C) Recovered landscape after analysis of diffusion in Figure 2B. This project was assigned to John Czerksi.

**Figure 2B** shows the 2D histogram of locations of diffusing particles in the underlying landscape defined by **Figure 2A**. No underlying landscape is obvious. However, a simple analysis involving running average of diffusion trajectories can show the underlying landscape (**Figure 2C**). **Figure 2C** also motivates further work on better analysis methods to extract the underlying landscape with higher resolution.
2.3. Three dimensional diffusion and chemical reactions

The kinetics of chemical reactions have been well established since the kinetic theory of gases was developed at the beginning of the twentieth century. In the earliest models, gas molecules were assumed to be hard spheres, and a reaction occurred when two molecules collided with enough energy to overcome the energy of the transition state. Therefore, knowledge of the concentrations of the reacting species as a function of time provided sufficient information to determine the rate at which a chemical reaction occurred. There are primarily two types of reactions for which the kinetic model applies. A unimolecular reaction occurs as the result of a process in which a single molecule rearranges to form one or more product(s) and is not considered here. In contrast, a bimolecular reaction occurs as a result of the collision between two molecules, \( A + B \rightarrow C \). For this reaction to occur, it is considered that a molecule of A comes within a certain distance (the binding radius) of a molecule of B.

![Figure 3. Simulation and analysis of bimolecular chemical reactions. A) Simulated kinetics of binding as a function of time when two chemicals A and B are mixed. \([A]=0.1 \text{ nM (fixed)} \text{ and } [B]=2.5 \text{ nM (cyan), 3.0 nM (orange), 4.0 nM (yellow), 5.0 (purple), 9.0 (green), 35 nM (magenta), and 85 nM (black). Kinetics for 35 nM and 85 nM are overlapping. Diffusion constant, } D = 50 \mu \text{m}^2 / s; \text{ time resolution, } \Delta t = 1 \text{ms, } k_{\text{OFF}} = 2 s^{-1} \text{ Curves at different } [B]\text{'s fit to the expression, } y = a\left(1 - \exp\left(-x / b\right)\right) , \text{ where } a \equiv k_{\text{ON}} * k_{\text{OFF}} \text{ and } b \equiv k_{\text{OFF}}. \text{ B) } k_{\text{ON}} = a / b \text{ (s}^{-1}) \text{ (blue), the diffusion limited rate of binding, } k_{\text{ON}} \text{ (s}^{-1}) \text{ (black), binding counts tracked directly from simulation, and } a \text{ (red) as a function of } [B]. \text{ C) } k_{\text{OFF}} \text{ (s}^{-1}), \text{ the measure of true binding affinity between A and B, as a function of concentration of molecule B. This project was assigned to Warren Colomb, Joe Maestas, and Kevin-Scott Rozmiarek.}]

It has been established that the above rate laws are adequate to describe most large systems (i.e. test tube chemistry) [23]. However, there are some drawbacks to this model. For smaller systems, such as those that occur in many biological systems, the above equations fail to accurately model the behavior of the system. Further, most chemical systems are not mechanically isolated, and
therefore subject to random disturbances. The deterministic rate laws given above fail to make the distinction between the actual system and the random perturbations that thermally equilibrate it [24].

An equally appropriate approach to describing the kinetics of a bimolecular reaction has been proposed [24, 25]. As stated previously, reaction rates are based on the concentrations of reactants and products, and for the deterministic model this approximation is appropriate because concentration and reaction rates are observables that develop only when time averaged measurements of the system are made. In a chemical system, the number of reacting molecules is so large (\(\sim 10^{23}\) particles) that continuous approximations, such as those developed in the deterministic model, yield relatively accurate results [25]. At the molecular level however, collisions between molecules are discrete processes and combination/dissociation events occur with some probability that is independent of molecules that are far away from the reacting particles [26]. That is, for a system in thermal equilibrium, the collision of two molecules occurs in a random manner. At this scale, small changes in the behavior of the system that do not affect the deterministic rate reactions can have a large effect on the probability density function [27]. It is therefore appropriate to model a bimolecular reaction using a probability density function that relies on the randomness of the interactions that occur, rather than the time evolution of the reactant concentration. As such, it is possible to model these collisions as Poisson processes. As stated previously, a mixture of two particles will interact if a molecule of A passes within the volume of a sphere defined by the binding radius, \(r_{\text{bind}}\), of a molecule of B. At the molecular scale, the probability of one binding event occurring is independent of any other binding events. That is, for a given time interval, it is highly unlikely that a given molecule will influence another [28]. Further, for a given event, there is no record of any previous event (i.e. there are no interactions between products and reactants, only interactions between reactants). A molecule of AB will persist for some time, after which it will undergo a unimolecular reaction and decompose into a molecule of A and a molecule of B. Through these processes, the reactants and products will reach steady-state concentrations. Due to the random instances of binding events, the resulting distribution of concentrations can be used to recover the observed deterministic rate constant.

In order to develop a fundamental understanding of Poisson processes, computer simulations were used to effectively demonstrate the stochastic nature of bimolecular chemical reactions. Using Monte Carlo (MC) simulations, product (AB) concentration was tracked as a function of time and
the rate constant for the forward reaction was recovered through comprehensive analysis of the results. All simulations and analyses were performed using MATLAB. MC simulations were used to recover the forward reaction rate. All trials were simulated for 2000 steps, with a constant step size, $\Delta t = 1 \text{ms}$. Six trials were performed with the initial concentration of species A set to 1.00 nM, while the initial concentration of species B was varied (0.01, 0.10, 1.00, 10.00, 100.00, 200.00 nM). Particles of both reactant species were distributed randomly within the simulation box and allowed to diffuse in three dimensions, where both species had diffusion constants, $D = 50 \mu m^2 / s$. Boundary conditions were set such that the particles would reflect back into the simulation box upon collision with the wall of the box. Diffusion steps were chosen from a normal distribution with a mean of 0.00 $\mu m$ and a standard deviation, $\sigma = \sqrt{6D\Delta t}$. Particles were randomly moved, with steps chosen from the normal distribution described above. When a particle of A and B were within 0.001 $\mu m$ of each other after a move, a product particle was formed and a binding event was recorded. The numbers of bound particles per time step was recorded that were fitted to exponential distributions (Figure 3A) to recover the simulated rates (Figure 3B and 3C).

3. Inter-conversion of states of a molecules

Molecules often change their states to perform their functions. For example, a protein may undergo conformational dynamics as they perform their functions. An experimentalist might label two different domains with two different dyes and detect the Forster Resonance Energy Transfer (FRET) to study the conformational dynamics. Experimental results may look like a noisy signal going up and down between a minimum and a maximum value. As an example, inter-conversion of a molecule between three states has been simulated and analyzed with the following scheme:

$$
\begin{align*}
A & \xrightleftharpoons[20 s^{-1}]{5 s^{-1}} B \\
 & \xrightleftharpoons[25 s^{-1}]{10 s^{-1}} C \\
 & \xrightleftharpoons[30 s^{-1}]{15 s^{-1}} A
\end{align*}
$$
Figure 4. Simulation and analysis of interconversion of states of a molecule. A) Simulated data (Δt = 0.001s) with three states A = 0.1, B = 0.5, and C = 0.8 with Gaussian noise

\[ y = \left( \frac{1}{\sigma \sqrt{2\pi}} \right) \exp \left( -\frac{(x - \mu)^2}{2\sigma^2} \right), \]

with \( \mu_{\text{input}} = 0 \) and \( \sigma_{\text{input}} = 0.05 \). Inter-conversion rates are \( k_{A\rightarrow B} = 5, k_{B\rightarrow A} = 20, k_{B\rightarrow C} = 10, k_{C\rightarrow B} = 25, k_{C\rightarrow A} = 30, \) and \( k_{A\rightarrow C} = 15 \). B) Area-normalized distribution of pairwise differences between consecutive signal values (blue bars) in Figure 4A fits to a Gaussian (red line),

\[ y = a \exp \left( -\frac{(x - b)^2}{c^2} \right), \]

with \( c = 0.1001 \pm 0.002 = 2\sigma_{\text{input}} \). C) Area-normalized histogram of the signal (blue bars) fits to three distinguishable Gaussians (red lines),

\[ y = a \exp \left( -\frac{(x - b)^2}{c^2} \right), \]

with \( b = 0.0999 \pm 0.0008, 0.4998 \pm 0.0002, \) and \( 0.8001 \pm 0.0002 \); \( c = 0.0708 \pm 0.0002, 0.0708 \pm 0.0002, \) and \( 0.0710 \pm 0.0002 \approx \sqrt{2}\sigma_{\text{input}} \) respectively. D) Simulated data with higher noise, \( \sigma_{\text{input}} = 0.5 \). All other simulation parameters are same as in Figure 4A. E) Area-normalized distribution of pairwise differences between consecutive signal values (blue bars) in Figure 4D fits to a Gaussian (red line),

\[ y = a \exp \left( -\frac{(x - b)^2}{c^2} \right), \]

with \( c = 1.004 \pm 0.002 = 2\sigma_{\text{input}} \). F) Area-normalized histogram of the signal (blue bars) Figure
4D fits to a Gaussian (red line), \( y = a \exp\left( -\left( x - b \right)^2 / c^2 \right) \), with \( c = 0.5 \approx \sqrt{2} \sigma_{\text{input}} \). This project was assigned to Derek Wright.

**Figure 4A** shows a typical example of a molecule inter-converting between three states with relatively low noise. One way to categorize and noise present in the data is to calculate the area-normalized distribution of pairwise differences between the consecutive signal values as shown in **Figure 4B**. Noise in the simulated data (blue bar graph) can be fitted to a Gaussian (solid red line),

\[
y = a \exp\left( -\left( x - b \right)^2 / c^2 \right)
\]

; the underlying noise, the amount of noise used in simulation, is equal to \( c / 2 \). For low noise, individual states can be easily determined from the area normalized histogram of the signal (**Figure 4C**). Interestingly, each peak can be fitted to a Gaussian (solid red line),

\[
y = a \exp\left( -\left( x - b \right)^2 / c^2 \right)
\]

; the underlying noise, the amount of noise used in simulation, is equal to \( c / \sqrt{2} \). For higher noise, the individual states cannot be visually seen (Figure 4D) which could be the case for many experimental situations. However, the underlying noise can still be characterized and measured as shown in **Figure 4E** and **4F**.

**4. Photon statistics from lasers**

Emission of photons from a laser can be modeled as a Poisson process. Hence, one of the experimental consequences is that the time between consecutive photon detections will follow an exponential distribution. A simple experimental schematic is shown in **Figure 5A**. The light from a red or blue laser is collected by an objective (Zeiss A-Plan 40X 0.65NA infinity-corrected) and
is converted into current by a PMT (Hamamatsu AA0296 H7422P-40). The current output of the PMT is converted into voltage by a transimpedance amplifier ($P_{\text{Amp}}$) and compared with a threshold voltage ($V_{\text{th}}$) to discriminate between the signal and noise. 

**B)** An example of time series of photon arrival times from a laser at 635 nm set at 0.010 µW. Each peak denotes the detection of photon by the PMT with integration time 0.04 s and $V_{\text{th}}=0.03$ V.

**C)** Area normalized distribution of times between consecutive photon arrivals (solid black circles) fits to an exponential function (red line), $y = a \exp(-kt)$, with $a = 0.25 \pm 0.01$ and $k = 5.20 \pm 0.14$ s$^{-1}$. 

**D)** Photon detection rate, $k$, as a function of laser power at 635 nm (open red circles) and 488 nm (open blue circles). Rates fit to a line (635 nm: red line, 488 nm: blue line), $y = a + mx$, with $a = 0.09 \pm 0.01$ and $m = 16.06 \pm 1.49$ (635 nm); $a = -0.02$ and $m = 15.89$ (488 nm). This project was assigned to Nathan Worts.

**Figure 5. Arrival times of photons from a laser as a Poisson process.**

A) The schematics of photon statistics measurement setup. Photons collected by the objective are detected by a photomultiplier tube (PMT). The current output from the PMT is converted into voltage using a transimpedance amplifier ($P_{\text{Amp}}$) and compared with a threshold voltage ($V_{\text{th}}$) to discriminate between the signal and noise.

B) An example of time series of photon arrival times from a laser at 635 nm set at 0.010 µW. Each peak denotes the detection of photon by the PMT with integration time 0.04 s and $V_{\text{th}}=0.03$ V.

C) Area normalized distribution of times between consecutive photon arrivals (solid black circles) fits to an exponential function (red line), $y = a \exp(-kt)$, with $a = 0.25 \pm 0.01$ and $k = 5.20 \pm 0.14$ s$^{-1}$. 

D) Photon detection rate, $k$, as a function of laser power at 635 nm (open red circles) and 488 nm (open blue circles). Rates fit to a line (635 nm: red line, 488 nm: blue line), $y = a + mx$, with $a = 0.09 \pm 0.01$ and $m = 16.06 \pm 1.49$ (635 nm); $a = -0.02$ and $m = 15.89$ (488 nm). This project was assigned to Nathan Worts.
234 300MHz 2Gs/s). **Figure 5B** shows a typical time stamped train of photon detection events. Some peaks could be due to noise and a threshold voltage needs to be set to compare above which the peaks are considered photon detection events. The probability density of time between consecutive photons is shown in **Figure 5C** (black circles) and can be fitted to an exponential (red line), $y = a \cdot \exp(-kt)$, with a rate of photon detection, $k \approx 5 \text{s}^{-1}$. Setting the threshold voltage is very important and therefore, should be set objectively by doing the experiments at different threshold voltage, calculate the probability distribution as shown in **Figure 5C**, determine the rates by fitting exponentials, plot the rates as a function of threshold voltages, and set the threshold voltage at a value where the rate vs the threshold voltage is flat. Of course, the rates measured depend on the laser power because higher the power, more the number of photons detected by the PMT. **Figure 5D** shows the rates as a function of the laser power which shows a linear dependence for two different lasers. Intercepts for two lasers are different and related to the physical processes involved in the lasers.

### 5. Statistics of fluorescent centers on a surface

Defects and grains in materials can be considered as a result of a Poisson process. In other words, there is constant probability of finding a defect or a grain in all three dimensions. Of course, there is a distinct possibility of underlying structure due to various interactions that can lead to different regions with different rates of Poisson process. Such heterogeneities that can happen both in space and time are termed as clustering, and can simulated to validate the experimental data analysis. To convey the message, a commonly used dye named tetracene has been deposited on quartz and imaged. Tetracene is an organic semiconductor material commonly used in organic light emitting diodes as well as organic field effect transistors. It is important to understand how tetracene grows on a substrate in order to create more efficient devices. For example, if aggregates or clusters of tetracene are formed, it would have different optical properties than if it were randomly distributed. In order to model clustering we must first study how tetracene initially deposits in a thin film and simulate the result as a spatial Poisson process. Going further one could model cluster and grain formation in order to elucidate how tetracene grows on a substrate.

To test if the nucleation of tetracene can be modeled as a Poisson process, a thin film of tetracene on a quartz slide was grown. The film was roughly 30 nm thick and was thermally evaporated onto the quartz slide. The sample was then imaged using a light sheet microscope which uses a 1.2 W,
532 nm, CW laser to illuminate the sample with a sheet of light. Fluorescence from tetracene was detected using an EMCCD camera (Andor IXON3). A false color version of such images is shown in Figure 6A. The raw image was converted into binary by applying a threshold using freely available software ImageJ. Locations of tetracene molecules (Figure 6B) were then determined using another freely available software R by applying a simple mean centroid measure. From the locations of tetracene centers, the radial distribution function (Figure 6C) as well as the area normalized histogram of all pairwise distances (Figure 6D) were calculated. The radial distribution function measures the probability of finding a molecule some radial distance from a reference molecule. The radial distribution function counts how many molecules lie within a shell of varying radii from a molecule. A simple Poisson processes in two dimension where each point has a constant probability of being the location of tetracene molecules was then simulated. The number of total tetracene centers are same for both the experimental and simulated data. The radial distribution function and pairwise distances were then calculated and matched with the experiments.

Figure 6. Grain distribution of tetracene on quartz A) False color image of tetracene deposited on quartz. B) Extracted centroids of tetracene grains in Figure 6A. C) Pair correlation function of locations of tetracene grains from Figure 6B (black) and simulated locations of tetracene assuming the deposition is a Poisson process (red). D) Distribution of distances between all possible pairs of tetracene grains. Experimental from Figure 6B (black open circles) and simulated assuming that the deposition is governed by a Poisson process (solid red line). This project was assigned to Andrew Proudian and Abigail Meyer.
While the distributions of pairwise distances overlap for the experimental and simulated data (Figure 6D), the pair correlation functions differ at small distance (Figure 6C). This observation has two implications. First, it is important to look at the same data using different analysis methods. Second, the difference indicated in the pair correlation functions suggests an underlying interaction, which is likely due to a van der Waals attraction between tetracene molecules. This would cause agglomeration of tetracene molecules at small distances, giving rise to the grains that we observe. In simulating the data as a Poisson process, it was assumed that tetracene molecules do not interact, which causes the difference in the pair correlation functions. This can be corrected by a two-step approach. First, a spatial point Poisson process is generated, and then, for points lying within some threshold radius are replaced with a single point at the centroid of the point cluster. Based upon the experimental data, this threshold distance is ~0.5 µm, or half the observed inhibition length.

6. Location of stars in the sky
Like chocolate chips in cookies or raisins in breads, stars in the sky can also be thought as the result of a Poisson process. Figure 7A shows an image of a part of the sky. The locations of stars were determined by converting the image into a binary image using a threshold value and calculating centroids in MATLAB. The area normalized probability distribution of distances between all possible pairs of stars (blue line) is shown in Figure 7B. To simulate the locations by considering a Poisson process, a three dimensional 100×100×100 matrix of random integers was created and then converted into a binary array based on a set threshold value to mimic the real image in Figure 7A. The resulting array was then projected onto the z-axis to create a 2 dimensional array by summing the values of the respective x- and y-coordinates. Any value >1 was set to 1 and counted as one star, as overlapping stars appear to be only one star and we cannot distinguish based on the image in Figure 7A. The area normalized distribution of locations of all possible pairs of 1s in the grid is shown as the red line in Figure 7B; which evidently does not match with the distribution for the real image (blue line). The reason is that a camera captures light at a solid angle and therefore, the simulated grid should be projected onto a two dimensional grid at a solid angle. To mimic solid angle projection, a three dimensional 100×100×25 matrix of random integers was created, converted into a binary array based on a set threshold value, and a solid angle projection was implemented by a specified “line of sight” that ensured that a slightly
larger field of view is projected onto the two dimensional plane for successive planes. The origin was redefined to be at the center of the plane so the compression was radially inwards towards the

**Figure 7. Statistics of locations of stars in the sky.**  

A) An image of the sky.  
B) Distribution of pairwise distances (normalized to the maximum distance) between stars in Figure 7A (blue), of simulated 3D grid if the stars are governed by Poisson process: 2D projection of a solid angle (green line), direct 2D projection (red line).  
C) Comparison of different threshold percentages (normalized to the maximum distance) on distribution of stars for actual image 30% (blue), 40% (green), 50% (red), 60% (purple), and 70% (black).  
D) Distribution of angles between stars on the circumference of a circle (normalized to the maximum angle between stars) between an actual image, (blue) a solid angle compression (green) and a 2D projection (red).  
E) Distribution of distances between stars on a straight line (normalized to the maximum distance) between an actual image (blue), a solid angle compression (green), and a 2D projection (red).  
F) Plotting various probabilities for the solid angle compression simulation with .2% (red), .45% (green), .85% (purple), 1% (yellow), and 1.5% (black). This project was assigned to Matthew Lovely.
center of the viewing plane. Then each successive xy plane was compressed a slightly larger amount (as defined by the viewing angle) starting with the z=25 (back) plane, until the first plane z=1 was compressed. These are then overlaid, summing any points with two stars. The planes are no longer the same size so the edges had to be trimmed. After compression, the size of the trimmed plane equals the size of the first plane. For example, the compression ratio is 50% and the end plane becomes a 50×50 grid to mimic a solid angle projection of 45°. For a solid angle projection at 51.5°, the resulting distribution (green line) in Figure 7B matches the distribution for the real image. To study the effect of sensitivity and resolution of the camera, the distributions were calculated for different threshold values as shown in Figure 7C. Even though different thresholds give different numbers of stars, they all lead to similar distributions. One of the defining signatures of a Poisson process is that the distribution of intervals between consecutive events is an exponential. For two and three dimensions, this signature can tested by drawing a line or a circle on the image. If circles are drawn on the image in Figure 7A and the distribution of angles between consecutive stars is plotted, the distributions are indeed exponentials both for the real image and simulated image as shown in Figure 7D. Figure 7E shows the similar exponential distributions for lines drawn on the image in Figure 7A. To simulate expansion of the universe or a well baked cake, we can change the constant probability at each point in the three dimensional matrix. Figure 7F shows the distributions after taking two dimensional projection at a solid angle for different probabilities in the simulation. It is interesting that the distributions are similar. One conclusion is that the statistical nature of a poorly baked chocolate chip cookie is similar to that of a well baked chocolate chip cookie.

7. Quantifying the heterogeneity of locations of cosmic ray sources

If the locations of stars can be described by a random Poisson process, it is logical to extend the argument for the cosmic ray sources as well. The Pierre Auger Observatory is a dedicated facility for cosmic ray detection consisting of 24 air fluorescence telescopes and 1600 water Cherenkov tanks. The 3000 sq. km detector array of Pierre Auger is dedicated to the study of high-energy cosmic rays. A schematic of the setup is shown in Figure 8A. It is located in Argentina and therefore, the field of view of the observatory is constrained to the southern hemisphere. The direction of individual cosmic ray is calculated by reconstructing the observed cosmic ray showers in the atmosphere. The Pierre Auger collaboration has made a subset of the data publicly available.
This data comprises 28.5k points in an energy range of 0.1 and 49.7 EeV (10^{18} eV) collected between 2004 and 2013.

**Figure 8. Quantifying heterogeneity of locations of cosmic ray sources.**

A) Galactic Coordinate Diagram. The reference coordinates for Pierre Auger cosmic ray sources are galactic latitude (b) and longitude (l). The diagram shows the location of the origin of the coordinate system: 0 degree longitude points to the center of the galaxy while latitude is measured with respect to the galactic plane. The location of Pierre Auger on Earth is represented with its approx. 120 degree field of view. The diagram helps understand the field of view of the detector in terms of galactic coordinate. Latitude ranges from -90 to 90 degrees and longitude spans 360 degrees.

B) Data heat map of the Poisson mean of distances between two cosmic ray sources. The data is binned in latitude and longitude to segment the locations of cosmic ray sources in the sky. Within individual bins, the angular distance between individual sources is fitted with a Poisson distribution and the mean is assigned to that bin in the heat map. Longitude is plotted versus latitude. A darker region is observable between -60 and -120 degrees longitude, while a brighter region is present between -60 and 60 degrees in longitude. The brightest regions of the heat map, present on the upper and lower edges, are due to bins filled with few data points close to the Earth’s geographic poles.

C) Simulated heat map of the Poisson mean of consecutive distance between two cosmic ray sources. The simulated data is plotted in the same visual representation as in the previous figure. There is a visible pattern similar to the one present in the data.

D) The
probability distribution of the angular separations of events within an example bin of galactic latitude and longitude: data (blue bars) and the Poisson fit (red circles) that gives the Poisson mean \( \lambda \). E) The distribution of Poisson means: data (red line) and the simulation (black circles). Error bars in Figures 8D and 8E are given by the square root of the count in the respective bins. This project was assigned to Jeff Johnsen and Kevin-Druis Merenda.

For simulation, it was assumed that the arrival directions on the earth are isotropic and correction was made for the exposure of the Pierre Auger Observatory. The details about the generation and propagation of the cosmic rays were ignored and only the measurement characteristics of Auger was accounted for. First, the galactic latitude/longitude unit sphere was divided into pixels using 500,000 points spaced with equal solid angle. Each pixel was assigned a constant probability such that the total probability equals 1. Next, each pixel probability was weighted with the relative exposure of Auger with a maximum zenith angle of 60 degrees, lowering the total probability for all pixels to about 1/3. This total probability was the constant probability at each step of the simulation that a cosmic ray arrival was detected. The simulation was run for 100,000 time steps. During each time step, each pixel was treated as a Poisson process with a constant probability of a cosmic ray arrival. A random number was generated between 0 and 1 and if the number was less than the Poisson probability of the current pixel, a cosmic ray arrival was simulated. A list of all simulated events was kept with the arrival time, galactic latitude, and galactic longitude of each event. The last step of the simulation was to account for the arrival direction measurement uncertainty of Auger. Although the true uncertainty varies with the energy of the detected cosmic ray, the simulation assumes a constant uncertainty of 1° for all events. The arrival direction of each simulated event was ‘smeared’ to a new arrival direction using a two dimensional Gaussian distribution with a standard deviation of 1° in all directions. 32,966 events in total were simulated, but subsequently truncated to 28,492 events to directly match the number of events in the real data set. Using the same latitude and longitude binning, we compare the Poisson mean distributions computed for the real (Figure 8B) and the simulated data (Figure 8C).

The cosmic ray events was divided into sub regions (bins) of galactic latitude (b) and longitude (l) with equal solid angle. In each bin, the angular distance between each pair of events was measured. The area normalized histograms of the angular distances were fitted by a Poisson probability distribution and the Poisson mean, \( \lambda \), for each bin of latitude and longitude with at least 4 cosmic ray events was calculated. Figure 8D shows the results for a bin. If the underlying process is Poisson in nature and the characteristic rate is not significantly different from one part of the bin
to another, the Poisson fit should be a good fit. To characterize goodness of fit, a reduced $\chi^2$ was calculated for each fit. The number of longitude segments was set as twice the number of latitude segments. The number of bins was varied and the distributions of Poisson means, $\lambda$s, were calculated (Figure 8E) with 2.23 (mean) ± 0.91 (standard deviation) for experimental data and 2.23 (mean) ± 0.83 (standard deviation) for simulated data. Larger bins tended to include more events in bins with enough events to fit the Poisson distributions, and smaller bins tended to show a better average goodness of fit. We chose 30 latitude segments by 60 longitude segments as a good balance between including the most events in bins with 4 or more events and getting a good fit of the Poisson distribution for individual bins.

8. Conclusion
Different processes can be simulated as Poisson processes. Simulated data can be used to validate methods used for experimental data analysis [7]. The self-consistent circular approach of combining experiments, analyses, and simulations can help design better experiments. This chapter focuses on the simplicity of the Poisson process approach to statistical mechanics and applies it to problems from diverse fields of research.

9. Acknowledgment
This chapter is the result of research-integrated teaching by Susanta K. Sarkar. Students learnt the concept of Poisson process by working on problems that are directly related to their own research interests. Students usually used Matlab to simulate their models and to validate their data analysis methods. This work was supported by the TechFee fund and the professional development fund to Susanta K. Sarkar.
References


1. Simulation of a Poisson process

%Clear out the Matlab workspace
clear; % clear removes all variables from the current workspace
clc; % clc clears all input and output from the command window display
rng('Shuffle'); % rng('Shuffle') seeds the random number generator based on the current % time so that rand produces a different sequence of numbers each time.

%Define the parameters
rate = 5; % rate of Poisson events per second
dt = 0.001; % time step (s)
prob = rate*dt; % the constant probability of detecting a Poisson event at each time step dt.
% the rate multiplied by the time step is a good approximation of the probability for small dt.
NPoints = 100000; % number of data points to be simulated

%Simulate data for the Poisson process defined above
data = zeros([NPoints,1]); % creates NPoints (row) x 1 (column) matrix filled with 0s.

% the for loop below generates a random number between 0 and 1 using rand and compares with % the prob above. If the rand is less than the prob, the corresponding cell value data(i,1) is % changed from 0 to 1. If not, the value is unchanged and remains 0.
for i=1:NPoints
    if rand<prob
        data(i,1)=1;
    end
end

%Save data with the first column “time” and the second column “detection outcome”
time = (dt:dt:dt*NPoints); % time points
datatobesaved = [time, data];
[file,path] = uiputfile('PoissonSimData.txt', 'Save File Name');
pause on
save([path,file], 'datatobesaved', '-ascii');
pause off
2. Analysis of a Poisson process

%Clear out the Matlab workspace
clear; % clear removes all variables from the current workspace
clc; % clc clears all input and output from the command window display

%Load Data
data = load('PoissonSimData.txt');

%Get the distribution of waiting times between consecutive Poisson events
dt = data(2,1)-data(1,1); % time resolution
timeconsecutive1s = dt*diff(find(data(:,2) == 1)); % finds time between consecutive 1s
% find(data(:,2) == 1) returns the linear indices of all the cells with 1s in the second column
% diff(find(data(:,2) == 1)) calculates differences between adjacent indices of cells with 1s
meanwaitingtime = mean(timeconsecutive1s) % calculates the mean of waiting times
% the mean waiting time should be (1/rate)

[counts, binpositions] = hist(timeconsecutive1s, 100); % creates the histogram with 100 bins
area = trapz(binpositions, counts); % calculates the area under the histogram
areanormalizedhistogram = (1/area)*counts; % calculates area normalized probability
areacheck = trapz(binpositions, areanormalizedhistogram) % this should be 1

%Fit the histogram
expfit = fit(binpositions', areanormalizedhistogram', 'exp1') % fits with a*exp(b*x)
% b should be equal to the simulated rate of the Poisson process

%Plot the histogram and the fit
figure()
plot(expfit)
hold on
bar(binpositions, areanormalizedhistogram)
hold off
3. Simulation and analysis of one dimensional normal diffusion

% Clear out the Matlab workspace
clear; % clear removes all variables from the current workspace
clc; % clc clears all input and output from the command window display

% Define the parameters
D = 5; % diffusion constant um^2/s
dt = 0.001; % time step (s)
stepSize = sqrt(2*D*dt); % average distance a diffusing particle moves in 1d
trackLength = 1000; % number of data points for each diffusing particle
tracks = 1000; % number of diffusing particles
time = 0:dt:dt*(trackLength-1); % time points
sigmaNoise = 3*stepSize; % positional noise at each time point

% Simulate diffusion with noise
normalDiff = cumsum(normrnd(0,stepSize,[trackLength,tracks]));
normalDiff_Noise = normrnd(normalDiff, sigmaNoise);

% Calculate mean squared displacement (MSD)
MSD_normal = zeros(trackLength,tracks);

for m = 1:(trackLength - 1)
    tempN=zeros(1,tracks);
    for k = 1:trackLength-m
        tempN = tempN + ( normalDiff(k+m,:) - normalDiff(k,:)).^2;
    end
    MSD_normal(m+1,:) = (1/(trackLength -m))*tempN;
end

% Fit mean MSD
ft = fittype( 'a + b*x', 'independent', 'x', 'dependent', 'y' );

% Assign error bars in MSD
MSD_Norm10 = zeros(tracks,10);
for i = 1:10
    MSD_Norm10(:,i) = mean(MSD_normal(:,1+(i-1)*10: i*10),2);
end

[fit_normal, gof_normal] = fit(time(1:200)',mean(MSD_Norm10(1:200,:),2),ft,'Startpoint', [0 0]);
fit_normal
err_Norm = std(MSD_Norm10, 0, 2);
% Plot MSD with error bars
resamp = 10;
figure(); hold on
errorbar(time(1:10:200),mean(MSD_Norm10(1:10:200,:),2),err_Norm(1:10:200))
plot(fit_normal, 'k')
xlim([0 0.2])
xlabel('Time (s)')
ylabel('MSD (um^2)')
hold off

% Calculate and plot the distribution of pairwise distances
pwd_N = diff(normalDiff);
pwd_N = reshape(pwd_N,[],1);
pwd_N_Noise = diff(normalDiff_Noise);
pwd_N_Noise = reshape(pwd_N_Noise,[],1);

nBins = 20;
figure()
pd_N = Area_Norm_Hist_Fit(pwd_N,nBins,'Pairwise difference (um)','Probability
density','Normal Diffusion - No Noise');

figure()
pd_N_Noise = Area_Norm_Hist_Fit(pwd_N_Noise,nBins,'Pairwise difference
(um)','Probability density','Normal Diffusion - Noise');

%Plot an example of diffusion track with and without noise
figure()
plot(time, [normalDiff_Noise(:,1), normalDiff(:,1)])
xlabel('Time (s)')
ylabel('Position (um)')
title('Normal Diffusion')
legend('Noise', 'No Noise')
4. Simulation of a molecule undergoing transitions between three states

% Clear out the Matlab workspace
clear; % clear removes all variables from the current workspace
clc; % clc clears all input and output from the command window display

Filename1 = 'C:\Users\ssarkar\Desktop\WithoutNoise\';
Filename2 = 'C:\Users\ssarkar\Desktop\WithNoise\';

% Define state values and the kinetic rates between the states
E1 = 0.1; % state 1
E2 = 0.5; % state 2
E3 = 0.9; % state 3
K12 = 0.01; % the decay rate from state 1 to state 2
K32 = 0.02; % the decay rate from state 3 to state 2
K23 = 0.03; % the decay rate from state 2 to state 3
K21 = 0.04; % the decay rate from state 2 to state 1
K13 = 0.05; % the decay rate from state 1 to state 3
K31 = 0.06; % the decay rate from state 3 to state 1

% Calculate the probabilities for transitioning from one state to the another
P23 = K23 / (K23 + K21);
P21 = K21 / (K23 + K21);
P32 = K32 / (K32 + K31);
P31 = K31 / (K32 + K31);
P12 = K12 / (K12 + K13);
P13 = K13 / (K12 + K13);

% Define the size of simulation
Nframe = 100000; % number of time steps per particle
Nparticles = 50; % number of particles

% Define the noise
mu = 0;
gnoise = 0.05;

% Simulate the transitions
for particleNum = 1 : Nparticles

% a signal array is created to simulate the time series data for each particle and a random starting point is chosen
signal = [];
state = randi(3);
% this loop appends to the signal until the length of the signal is equal to Nframe
while length(signal) < Nframe

    if state == 1
        prob = rand;
        if prob <= P12
            t = exprnd( 1 / K12 );
            signal = [signal; E1*ones(ceil(t),1)];
            state = 2;
        else
            t = exprnd( 1 / K13 );
            signal = [signal; E1*ones(ceil(t),1)];
            state = 3;
        end

    if length(signal) > Nframe
        signal = signal(1:Nframe);
        break
    end

    elseif state == 2
        prob = rand;
        if prob <= P21
            t = exprnd( 1 / K21 );
            signal = [signal; E2*ones(ceil(t),1)];
            state = 1;
        else
            t = exprnd( 1 / K23 );
            signal = [signal; E2*ones(ceil(t),1)];
            state = 3;
        end

    if length(signal) > Nframe
        signal = signal(1:Nframe);
        break
    end

    elseif state == 3
        prob = rand;
        if prob <= P31
            t = exprnd( 1 / K31 );
            signal = [signal; E3*ones(ceil(t),1)];
            state = 1;
        else
            t = exprnd( 1 / K32 );
            signal = [signal; E3*ones(ceil(t),1)];

```
state = 2;
end

if length(signal) > Nframe
    signal = signal(1:Nframe);
    break
end
end

doisySignal = normrnd(signal, gnoise);

str = sprintf('%04d', particleNum);

save(strcat(Filename1, 'WithoutNoise', str, '.txt'), 'signal', '-ascii');
save(strcat(Filename2, 'WithNoise', str, '.txt'), 'noisySignal', '-ascii');
end