JOURNAL OF CHEMICAL EDUCATION

Open Access

pubs.acs.org/jchemeduc

Activity

Interactive Learning of Diffusion and the Diffusion Equation with Mathematical Software

Victor Leborán and Francisco Rivadulla*

Cite This: *J. Chem. Educ.* 2023, 100, 4525−4529

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: The process of diffusion is intimately related to random molecular displacement and is central to many problems in chemistry. Here, we provide an intuitive derivation of the diffusion equation based on the analysis of the random trajectories of particles generated and plotted by simple MATLAB and Python scripts. The codes are very simple and profusely commented on, so they can be modified by students and instructors with little experience with these software packages. The codes plot the trajectories in 1D, 2D, and 3D of a number of particles during a number of steps, both specified by the operator. Advanced versions of the codes with animated evolution of the diffusion process are also provided. The interactive generation and analysis of the trajectories of the particles in different scenarios allows a faster and more complete understanding of the effect of the number of particles, diffusion time, or dimensionality of the system, on the basic aspects of chemical diffusion, like mean square displacement, through the statistical nature of the process. A series of activities for the instructor and students was also suggested.

KEYWORDS: *Upper-Division Undergraduate, Computer-Based Learning, Chemical Diffusion*

INTRODUCTION

The general nonsteady state diffusion equation predicts how diffusion causes the concentration to change with time and position, and therefore, it is of the highest importance to many areas of chemistry. This, so-called second Fick's law, can be expressed as

$$\frac{\partial c}{\partial t} = D\frac{\partial^2 c}{\partial x^2} \tag{1}$$

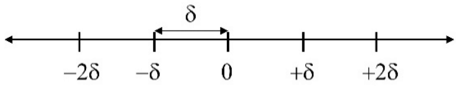
where c, D, x, and t refer to the molecular concentration, diffusion coefficient, position, and time, respectively.

However, obtaining the solutions of this parabolic partial differential equation is not easy, and most chemistry-physics textbooks do not make a formal derivation, offering only the general solutions of this equation, which are[1−3]

$$\begin{cases} c(x,\ t) = \dfrac{n_0}{2(\pi Dt)^{1/2}} e^{-x^2/4Dt} \\ c(r,\ t) = \dfrac{n_0}{8(\pi Dt)^{3/2}} e^{-r^2/4Dt} \end{cases} \tag{2}$$

in 1D and 3D, respectively (starting with n_0 moles of molecules concentrated at a point at the initial state, $t = 0$; note that r is the radial distance from the origin). [In the 2D solution of the diffusion equation, the denominator in the preexponential term is $4\pi Dt$; the term r^2 in this case refers to the radial distance from the origin for a process of isotropic diffusion in a plane.] As a result, many students miss the step-by-step connection between eqs 1 and 2, get lost in the mathematical derivation, and more importantly, do not properly understand the physical meaning of concepts like root-mean-square displacement (RMSD), which is key for a proper understanding of diffusion.

Some general physical-chemistry textbooks approach the concept of diffusion based on the Random Walk (RW) model,[2,3] which in its simple 1D case considers the displacement of independent particles moving along one dimension, to the left or to the right, with equal probability and in steps of size δ (τ time per step):

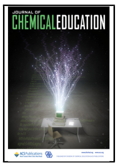


After a given time, the particle moves a distance equal to the net difference between steps taken to the right (n_R) and to the left (n_L), so that the net probability of finding a particle at a distance from the origin is given by

$$P = \frac{1}{2^n}\frac{n!}{n_R!n_L!}$$

This result is the binomial coefficient $\binom{n}{n_R}$ divided by the total number of configurations after n steps, 2^n. For a large enough

Received: April 14, 2023
Revised: September 1, 2023
Published: September 21, 2023



https://doi.org/10.1021/acs.jchemed.3c00309

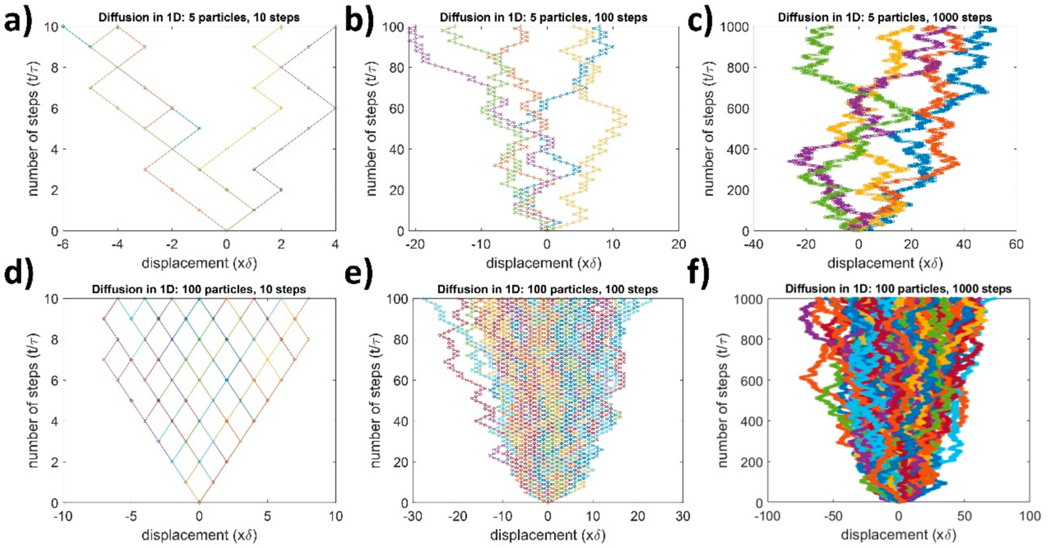


Figure 1. 1D random trajectories of $N = 5$ particles after (a) $S = 10$ steps, (b) $S = 100$ steps, and (c) $S = 1000$ steps. The case for $N = 100$ particles after 10, 100, and 1000 steps is presented in (d−f), respectively. The paths were generated by the MATLAB code provided as Supporting Information.

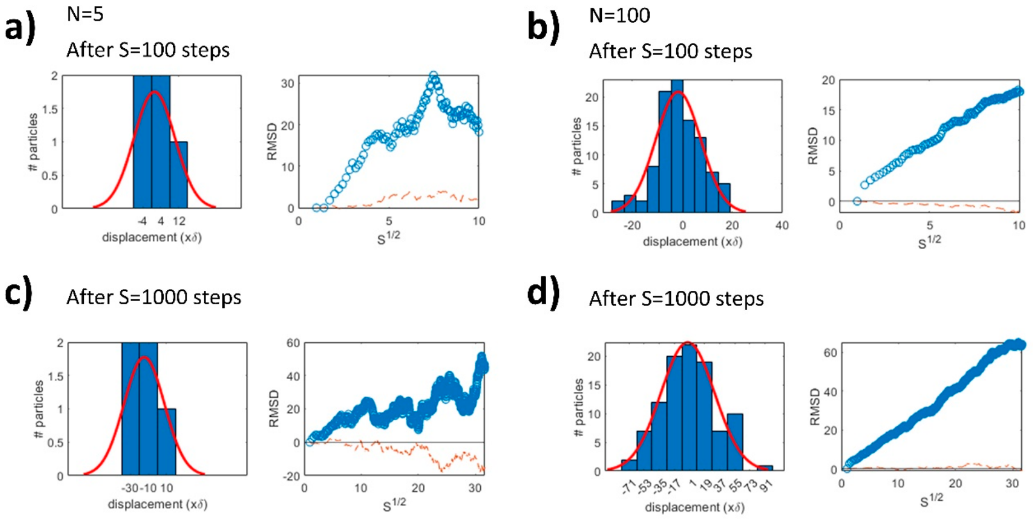


Figure 2. Histograms of the distribution of $N = 5$ particles (a, c) and $N = 100$ particles (b, d) after different numbers of steps and their corresponding RMSDs. As time increases, the distribution tends to a Gaussian function centered at zero for a large enough number of particles. The RMSD (open blue dots) and the MSD (mean square displacement, dotted orange line) are presented as a function of the square root of the number of steps. Increasing the number of particles makes the MSD ≈ 0 and the RMSD $\propto S^{1/2}$. Plots were obtained with the MATLAB code in the Supporting Information.

number of steps, the factorials can be approximated by the Stirling formula to give:

$$\ln P = \ln\left(\frac{2\tau}{\pi t}\right)^{1/2} - \frac{x^2\tau}{2t\delta^2}$$

and therefore, we arrive at the probability distribution:

$$P(x,\, t) = \left(\frac{2\tau}{\pi t}\right)^{1/2} e^{-x^2\tau/2t\delta^2} \tag{3}$$

This equation is similar to that of 2, assuming $Dt = n\delta^2/2$.

Next, we show that an analysis of the random trajectories of a set of particles with the aid of mathematical software provides an easy and intuitive approach to the process of diffusion and the probabilistic meaning of the diffusion equation. The MATLAB live scripts used to generate the plots of random trajectories in 1D, 2D, and 3D appearing in this paper are provided as Supporting Information (an advanced animated version in 2D and 3D is also provided). Working with MATLAB scripts is a clear improvement over previous reports of computer-aided solutions of stochastic differential equations to simulate the 1D random walk.[4] The use of MATLAB and other mathematical software for teaching complex concepts in chemistry is increasingly common.[5−12] In addition to their pedagogical value in chemistry, the tasks described in some of these papers were designed to initiate students into program environments that will surely be very useful in their professional development.[13−19]

The scripts accompanying this paper can be easily modified by instructors and students for an interactive understanding of the process of diffusion and could serve as a basis for students to build small programs for other tasks. A Python version of the MATLAB code is also provided in the Supporting Information.

This activity is intended as a 1 or 2 h classroom seminar (depending on previous experience using MATLAB or Python) plus another 2 h of student work on the data analysis. It is

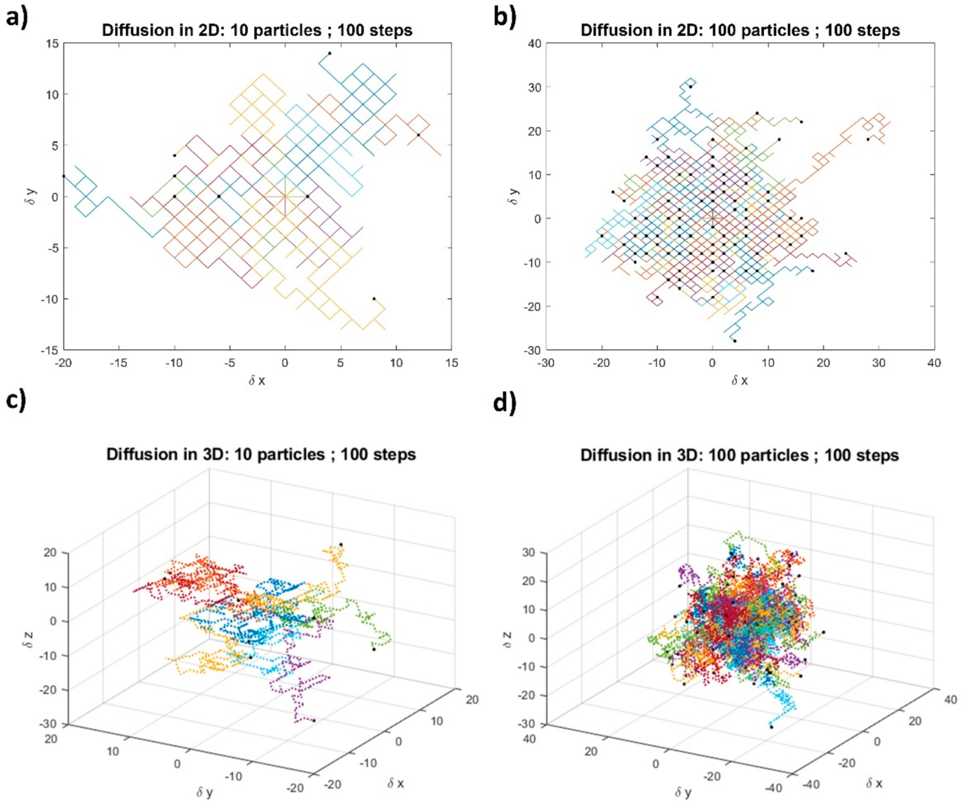


Figure 3. Random trajectories in 2D (a, b) and 3D (c, d) for different numbers of particles and steps. The black dots at the end of each trajectory mark the position of the particles after 100 steps.

primarily aimed for students of introductory courses to statistical thermodynamics and transport phenomena. However, the representation of the diffusion trajectories in several dimensions can also be used to introduce the phenomenon of chemical diffusion to students in earlier courses for a degree in chemistry, where Fick's second law is explained for the first time.

A series of student activities are suggested at the end of the paper.

■ DISCUSSION

The trajectory of an RW can be visualized through a simple computer program which generates a set of random numbers: every positive/negative number adds/subtracts δ to the previous position. In Figure 1, we show the random trajectories in 1D generated for an increasing number of particles, N, and steps, S (see MATLAB live script in the Supporting Information). Note that all the steps take the same time, τ, so that the total number of steps after time t is $n = t/\tau$.

Some important observations can be derived from these plots:

(1) For a small number of particles ($N = 5$), the trajectories look very random, even after a large number of steps.

(2) Increasing the number of particles ($N = 100$) results in quite symmetric distributions of trajectories around $x = 0$: the average displacement of the particle of the system is zero.

(3) The particles reach larger distances from the origin ($\pm x\delta$) as the number of steps increases: the distribution of trajectories widens.

The width of the distribution of trajectories can be measured by representing the number of particles at a given distance from the origin (at a given $\pm$xδ) after a given number of steps. These correspond to the horizontal sections of Figure 1, which can also be obtained from the MATLAB code. Some representative distributions are shown in Figure 2.

As the number of particles and steps increase, the histograms of Figure 2 fit to a Gaussian function centered at zero:[20]

$$G(x) = \frac{1}{(\sigma^2 2\pi)^{1/2}} e^{-x^2/2\sigma^2} \tag{4}$$

$G(x)$ in eq 4 is a probability density function and represents the probability of finding a particle between x and x + dx. This function has a shape similar to that of eq 2 in 1D. Looking at the histograms in Figure 2, the width of the distribution increases as the number of steps does. Therefore, the variance of the Gaussian distribution, σ^2, is proportional to the average or mean displacement squared: $\sigma^2 = \langle(x\delta)^2\rangle$.

In Figure 2, we plot the mean displacement $\langle x\delta\rangle$ and the square root of the mean displacement squared, RMSD = $\langle(x\delta)^2\rangle^{1/2}$ for a different number of particles and steps. As N increases, it becomes evident that the average displacement is zero, and the RMSD increases linearly with the square root of the number of steps: RMSD $\propto S^{1/2}$. Therefore, after S steps of size δ, the RMSD is as follows

$$\text{RMSD} = S^{1/2}\delta \tag{5}$$

This is a key concept of diffusion, which can be verified with the MATLAB code by systematically changing the number of particles and steps.

Finally, given $S = t/\tau$, the RMSD can be related to the time during which the system evolves:

$$\mathrm{RMSD} = S^{1/2}\delta = \left(\frac{\delta^2}{\tau}\right)^{1/2} t^{1/2} \tag{6}$$

The magnitude (δ^2/τ) has the units of m^2/s and represents the proportionality coefficient between the RMSD of a particle and the elapsed time, i.e., a diffusion coefficient, D. This is the result of the Einstein−Smoluchowski equation:[2] $D = \lambda^2/2\tau$; where λ is the step length in the diffusion process.

Putting this result back into eq 4,

$$G(x) = \frac{1}{(2\pi Dt)^{1/2}} e^{-x^2/2\sigma^2} \tag{7}$$

which differs from eq 2 only by multiplying the diffusion coefficient by a factor of 2.

It is very easy to generalize these codes to observe the process of diffusion in 2D and 3D. Figure 3 includes some plots obtained with the corresponding code in MATLAB accompanying this article.

Students may compare the trajectories generated by the codes with experimental results of the diffusion paths of cells (in both 2D and 3D) reported by Wu et al.[21] and the 3D tracking of single mRNA macromolecules reported by Thompson et al.[22]

Also, in a recent paper published in this *Journal*,[23] students can see how the structure of a polymer in solution (its effective radius) can be simulated through a random walk process. The authors provide the structures generated by a random walk code to avoid the computational load to the students. However, the MATLAB and Python scripts provided here can be used by the students to generate random structures that reproduce the structure of polymers in 2D and 3D, helping them to understand and comprehend the phenomena.

In summary, we have presented a simple analysis of the process of diffusion using a MATLAB code that calculates the random trajectories of a collection of particles moving in 1D, 2D, and 3D for a given time. The analysis of these trajectories allows an intuitive derivation of the diffusion equation as a probability distribution function. The concept of RMSD and its dependence on the square root of time, a key concept characteristic of a diffusive process, are also easily derived from the analysis of the trajectories.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available at https://pubs.acs.org/doi/10.1021/acs.jchemed.3c00309.

MATLAB livescript for the generation of random trajectories of a collection of particles in 1D, 2D, and 3D; calculation of the histograms of the distribution of particles (ZIP)

Jupyter Notebook with the equivalent Python code; a MATLAB and Python code to follow the dynamic evolution of the particles with time in 2D and 3D (ZIP)

Instructor handbook with the activities suggested, as well as detailed instruction on how to write these codes (PDF; DOCX)

Student handout (PDF; DOCX)

■ AUTHOR INFORMATION

Corresponding Author

Francisco Rivadulla − *CiQUS, Centro Singular de Investigación en Química Biolóxica e Materiais Moleculares, Universidade de Santiago de Compostela, 15782 Santiago de Compostela, Spain; Departamento de Química-Física, Universidade de Santiago de Compostela, 15782 Santiago de Compostela, Spain;* orcid.org/0000-0003-3099-0159; Email: f.rivadulla@usc.es

Author

Victor Leborán − *CiQUS, Centro Singular de Investigación en Química Biolóxica e Materiais Moleculares, Universidade de Santiago de Compostela, 15782 Santiago de Compostela, Spain*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jchemed.3c00309

Notes

The authors declare no competing financial interest.

■ REFERENCES

(1) Crank, J. *The Mathematics of Diffusion*; Oxford University Press, 1975.

(2) Atkins, P.; de Paula, J. *Physical Chemistry*, 8th ed.; W. H. Freeman, 2006.

(3) Engel, T.; Reid, P. *Physical Chemistry*, 3rd ed.; Prentice Hall, 2012.

(4) Muranaka, K. Simulation of One-Dimensional Brownian Motion by Stochastic Differential Equations. *J. Chem. Educ.* **1999**, *76* (7), 994.

(5) Jameson, G.; Brüschweiler, R. Active Learning Approach for an Intuitive Understanding of the Boltzmann Distribution by Basic Computer Simulations. *J. Chem. Educ.* **2020**, *97* (10), 3910−3913.

(6) Menke, E. J. Series of Jupyter Notebooks Using Python for an Analytical Chemistry Course. *J. Chem. Educ.* **2020**, *97* (10), 3899−3903.

(7) van Staveren, M. Integrating Python into a Physical Chemistry Lab. *J. Chem. Educ.* **2022**, *99* (7), 2604−2609.

(8) Lozano-Parada, J. H.; Burnham, H.; Machuca Martinez, F. Pedagogical Approach to the Modeling and Simulation of Oscillating Chemical Systems with Modern Software: The Brusselator Model. *J. Chem. Educ.* **2018**, *95* (5), 758−766.

(9) Joss, L.; Müller, E. A. Machine Learning for Fluid Property Correlations: Classroom Examples with MATLAB. *J. Chem. Educ.* **2019**, *96* (4), 697−703.

(10) Weiss, C. J. A Creative Commons Textbook for Teaching Scientific Computing to Chemistry Students with Python and Jupyter Notebooks. *J. Chem. Educ.* **2021**, *98* (2), 489−494.

(11) Bravenec, A. D.; Ward, K. D. Interactive Python Notebooks for Physical Chemistry. *J. Chem. Educ.* **2023**, *100* (2), 933−940.

(12) Mazza, D.; Canuto, E. *Fundamental Chemistry with Matlab*; Elsevier, 2022.

(13) Arrabal-Campos, F. M.; Cortés-Villena, A.; Fernández, I. Building "My First NMRviewer": A Project Incorporating Coding and Programming Tasks in the Undergraduate Chemistry Curricula. *J. Chem. Educ.* **2017**, *94* (9), 1372−1376.

(14) Srnec, M. N.; Upadhyay, S.; Madura, J. D. Teaching Reciprocal Space to Undergraduates via Theory and Code Components of an IPython Notebook. *J. Chem. Educ.* **2016**, *93* (12), 2106−2109.

(15) Tanemura, K. A.; Sierra-Costa, D.; Merz, K. M. *Python for Chemists* **2022**, DOI: 10.1021/acsinfocus.7e5030.

(16) Fisher, A. A. E. Developing the Chemist's Inner Coder: A MATLAB Tutorial on the Stochastic Simulation of a Pseudo-First-Order Reaction. *J. Chem. Educ.* **2020**, *97* (5), 1476−1480.

(17) Weiss, C. J. Introduction to Stochastic Simulations for Chemical and Physical Processes: Principles and Applications. *J. Chem. Educ.* **2017**, *94* (12), 1904−1910.

(18) Konkol, J. A.; Tsilomelekis, G. Porchlight: An Accessible and Interactive Aid in Preprocessing of Spectral Data. *J. Chem. Educ.* **2023**, *100* (3), 1326−1332.

(19) Lafuente, D.; Cohen, B.; Fiorini, G.; García, A. A.; Bringas, M.; Morzan, E.; Onna, D. A Gentle Introduction to Machine Learning for

Chemists: An Undergraduate Workshop Using Python Notebooks for Visualization, Data Processing, Analysis, and Modeling. *J. Chem. Educ.* **2021**, *98* (9), 2892−2898.

(20) Steiner, E. *The Chemistry Maths Book*; Oxford University Press, 1996.

(21) Wu, P.-H.; Giri, A.; Sun, S. X.; Wirtz, D. Three-Dimensional Cell Migration Does Not Follow a Random Walk. *Proc. Natl. Acad. Sci. U. S. A.* **2014**, *111* (11), 3949−3954.

(22) Thompson, M. A.; Casolari, J. M.; Badieirostami, M.; Brown, P. O.; Moerner, W. E. Three-Dimensional Tracking of Single MRNA Particles in *Saccharomyces Cerevisiae* Using a Double-Helix Point Spread Function. *Proc. Natl. Acad. Sci. U. S. A.* **2010**, *107* (42), 17864−17871.

(23) Everest, M. A.; Toves, A. J.; Reeves, M. S. What Is the Structure of a Polymer in Solution? *J. Chem. Educ.* **2023**, *100* (7), 2719−2723.