2024 UT Austin MITE Name:____________________

Modeling Mass Diffusion in Materials Science Applications

I. Warm up: Think-pair-share

Write down a few reasons why it is important to study materials

-
-
-

II. In-person activity: Human diffusion

1. Move to the space according to the instructor directions
2. Find a spot on the side of the room specified by the instructor. Choose a direction to face (forward, backward, left, or right).
3. When instructed, shuffle (slowly and carefully) along the direction you are facing.
4. If you are about to run into a wall or another person, change directions and shuffle in the new direction.
5. Repeat steps 3 and 4 many times.

What has happened to the distribution of people in the room?

III. Demo (optional): The effect of temperature

Materials:

- 2 containers of equal volume with an open mouth (e.g., 250 mL)
- 2 colors of food coloring
- 2 pipettes (optional)
- equal volumes of hot water and cold water (could be tap, e.g., 250 mL)
- timer or clock

1. Heat up the water.
2. Pour the hot water in one of the containers (be careful!) and the cold water in the other container.
3. Drop ~10 drops food coloring slowly near the surface of the cold water container. Note the time when you put in the food coloring. Be sure not to bump the containers!
4. Repeat step 3 with the hot water container using the other color of food coloring.
5. Wait until food coloring has spread throughout the container(s).

2024 UT Austin MITE Name:____________________

Which container (hot or cold water) had the faster spread of food coloring?

Bonus: why is temperature important in how fast the food coloring spreads?

For the second half of this activity:
The materials for this portion of the activity are located:

On github:
https://github.com/wangmatgroup/outreach/tree/main/MITE/Su24-diffusion/random-walk-mercury

On Google Drive:
https://shorturl.at/uebUF

The contents are:

- random-walk-mercury (folder): contains the Python code and mercury demo file
- instructions-modify-code.pdf: instructions for modifying the code (activity IV and VI)
- instructions-run-mercury-locally.pdf: instructions for installing the mercury package for running the random walk simulator GUI (activity V).
- Su24-MIT-activity-diffusion.pdf: the presentation
- Su24-MITE-diffusion-worksheet.pdf: this document

IV. Some familiarity with Python.

We are going to use Python to implement the Random Walk Diffusion Model. Python is known for its easy syntax and readability and is a great language to first learn some programming.

Below are some exercises to get familiar with Python. Type the following in the Spyder editor and run the code. What are the outputs of the following commands?

- `print(“Hello World!”)`
- `print(nt)`
- `nt = 10; print(nt)`
- `import numpy as np; x = np.zeros(nt); print(x)`
- `nt = 5; x = np.zeros(nt); print(x)`

Try the following set of programming lines, hit ‘Enter’ for each new line. What is the output?

-
  ```
  x = [1,2,3]
  x[0]
  x[0] = 4
  ```

2024 UT Austin MITE Name:____________________

V. Simulating Diffusion with the Random Walk Diffusion Model

Go to https://rwd2d-mercury.runmercury.com/ or scan the QR code →

Note: A reasonable number steps for this simulation demonstration is a few 100s of steps. A higher number of steps would require a faster code and a faster computer!

Note: There are two display types. “Static (fast)” produces a static plot of the entire random walk path whereas “interactive (slow)” produces an animation of the random walk path.

Play around with the simulation and the parameters provided with the following guiding questions:

1. If the random walker starts at the origin (x, y) = (0,0), where does the walker end up after $n = 100$ steps? After $n = 500$ steps? Repeat the simulation by changing the number of steps and hitting the ‘Enter’ key.

2. There are two types of 2D lattices to choose from: a square lattice and a triangular lattice. Switch between the square and triangular lattice and rerun the simulation. What do you notice about the path?

VI. Modeling Diffusion with the Random Walk Diffusion Model

Now we are going to look a little under the hood into the code and do a little coding ourselves! We have chosen the Python language, which is known for its ease of use, programming syntax, and popularity as a scientific coding language.

Fill in the missing part of the code:

Open the file `lattice_2D_EXERCISE.py.`

1. Run the code, what happens?

2. Fill in the missing part of the code:
```
# ==== FILL ME IN ====== #
# next two lines define the jumps on the square lattice:
#   right, up, left, down
delx = np.array([1,0,?,?])
dely = np.array([0,1,?,?])
```

3. Save and run the code.

Modifying the code:

As given, the code generates two trajectories with `nt = 100` and `nt = 1000`.

1. Modify the code to give you three different trajectories with `nt = 100`.
2. Modify the code to give a trajectory with `nt = 10000`.