Modeling Mass Diffusion in Materials Science Applications

2024 MITE Summer Program
Hosted by the Wang Materials Group

The University of Texas at Austin
McKetta Department of Chemical Engineering
Cockrell School of Engineering

TEXAS MATERIALS INSTITUTE
Materials Science & Engineering

Today's plan

- What is Materials Science and Engineering?
  - Examples
  - Materials Science Tetrahedron
- What is diffusion?
  - In-person activity
- Where is diffusion important?
  - Examples from everyday
  - Examples from materials science
- What is computation and simulation?
  - Random Walk Diffusion
- Using simulations to model diffusion

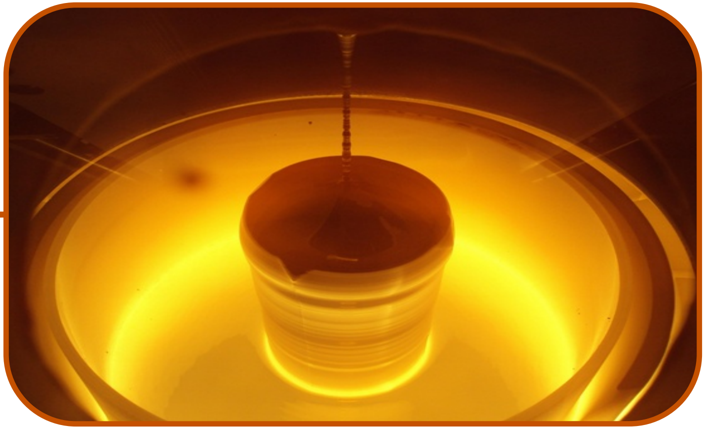
What is materials science and engineering?

Discovering, understanding, and designing materials (often times solids) through the study and application of processing-structure-property-performance relationships

Processing → Structure → Properties → Performance

What is materials science and engineering?

How is the material actually made in a lab/factory or adapted from nature?

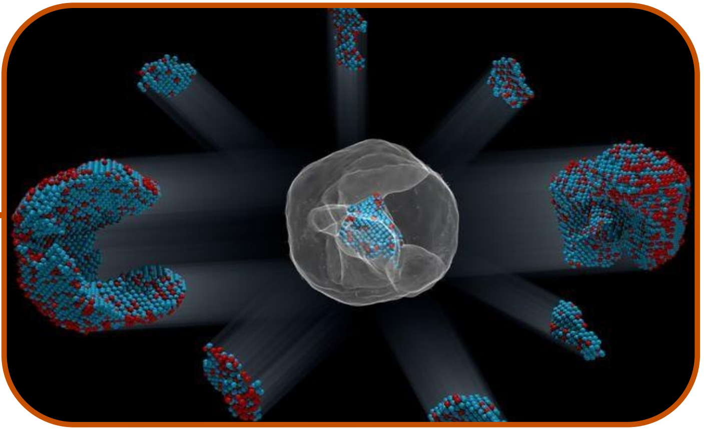


Processing → Structure → Properties → Performance

Adapted from Modern Materials Technologies, Image from Omnidyne

What is materials science and engineering?

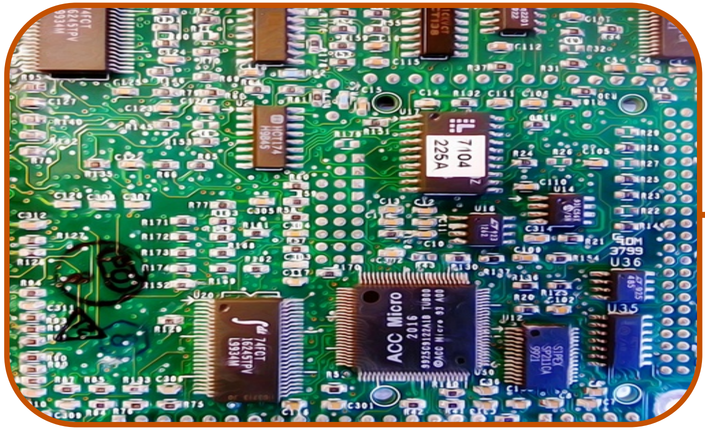
How are the internal components arranged?



Processing → Structure → Properties → Performance

Adapted from Modern Materials Technologies, Image from Chong, L. “Bringing atomic mapping to the mainstream” Phys Org (2017)

What is materials science and engineering?



How does it respond to an external stimulus?

Mechanical

Electrical

Thermal

Magnetic

Optical

Chemical

Processing → Structure → Properties → Performance

Adapted from Modern Materials TechnologiesModern Materials Technologies

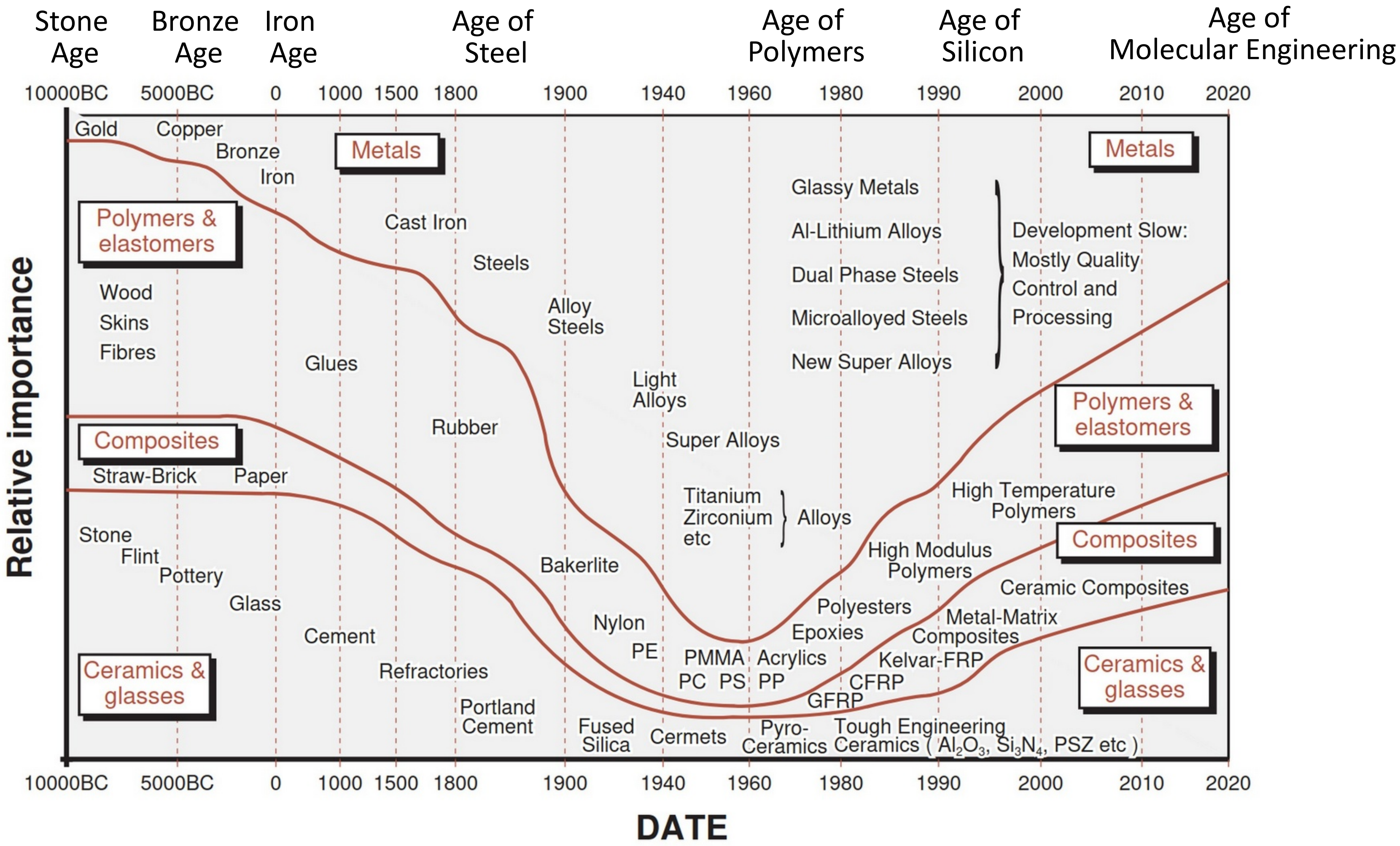
What is materials science and engineering?

How does the material help the user/world?

Processing → Structure → Properties → Performance

Image from Alset International

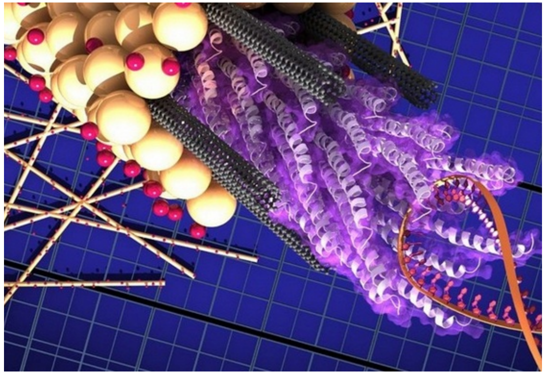
Why study materials? (Think-pair-share)



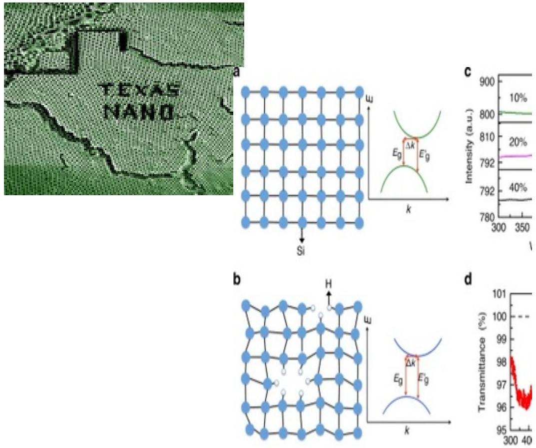
Adapted from Ashby M.F., 2005. Materials Selection in Mechanical Design, 3rd ed

Examples of Materials Science and/or Engineering

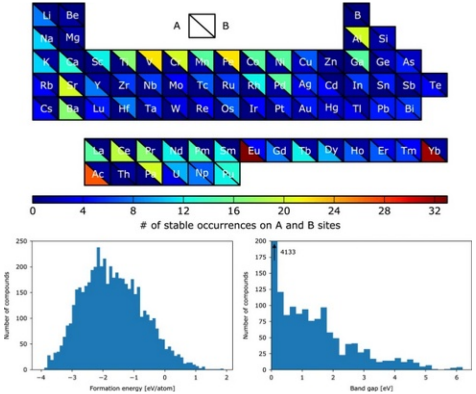
3D printing soft bio-polymers
Nelson Lab, University of Washington



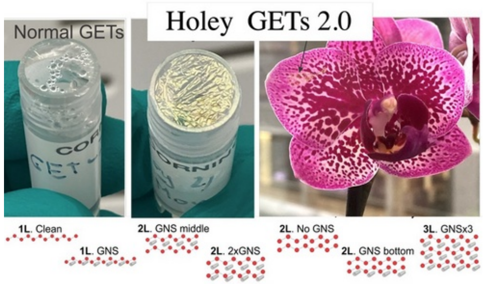
Using viruses as templates to make more efficient solar cells
Belcher Lab, MIT



Using defects in nanoparticles for high-efficiency, low-loss nanophotonics
Korgel & Zheng Research Groups, UT Austin



Using supercomputers to discover new and ideal oxides
Wolverton Research Group, Northwestern

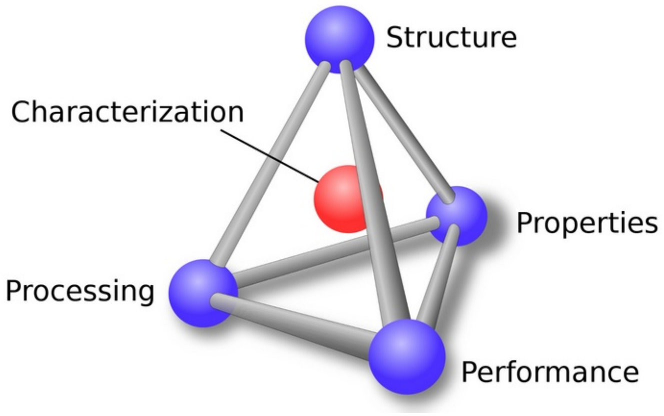


Breathable graphene tatoos for personalized healthcare
Akinwande Nano Research Group, UT Austin

What is materials science and engineering?

Discovering, understanding, and designing materials (often times solids) through the study and application of processing-structure-property-performance relationships

Processing → Structure → Properties → Performance



What is diffusion?

https://www.youtube.com/watch?v=UlnubVuJvZM

In-person activity: Human Diffusion

What happens to the distribution of the people in the room?

Step 0: Find a space at least twice as large as the space occupied by participants.

Step 1: Pack all participants to one side of the room as close as possible. Each participant faces a different direction.

Step 2: Each participant begins shuffling (slowly and carefully!) in the direction they are facing.

Step 3: If a participant runs into another object (e.g., wall or other participant), s/he moves in another random direction.

Step 4: Repeat for steps 2 and 3 many times.

What is diffusion? The effect of temperature

What happens to the distribution of the ink in the flask?

*not strictly mass diffusion

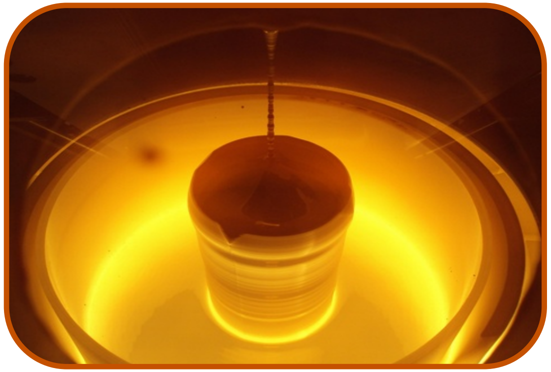
Everyday examples of diffusion

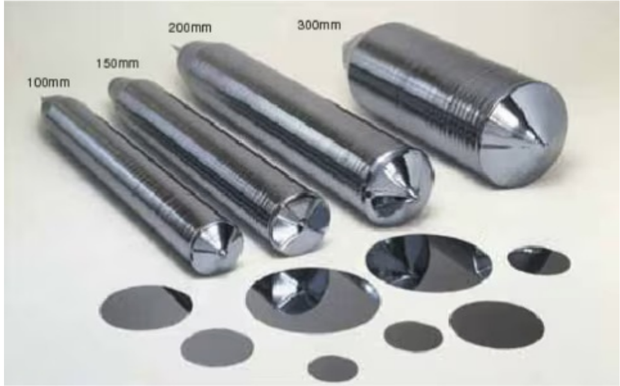
CO_2 (gas) dissolved in flavored liquid

Air pollutants

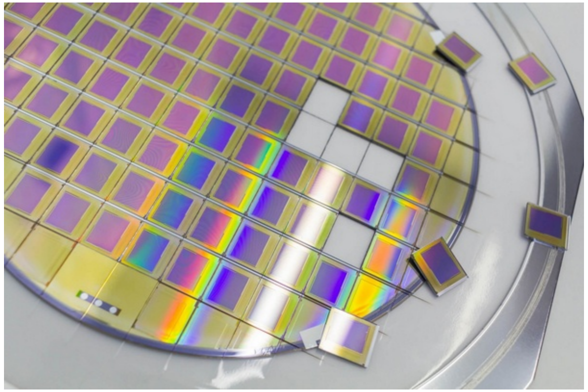
Aerosols, air fresheners, perfumes

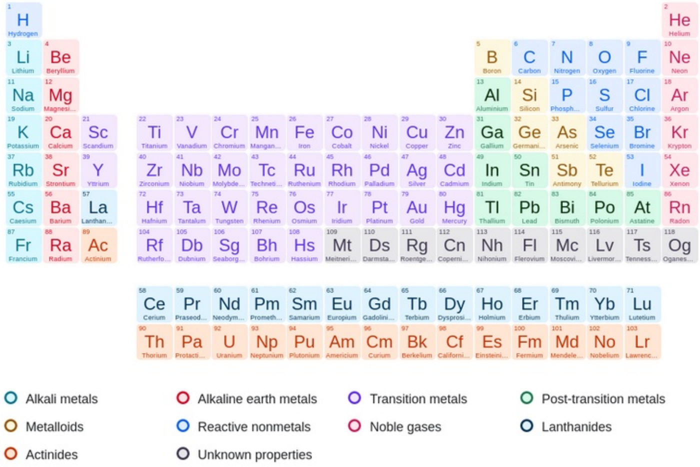
Silicon Wafer Processing: Example of Diffusion in Materials Science





99.999999999% pure Silicon

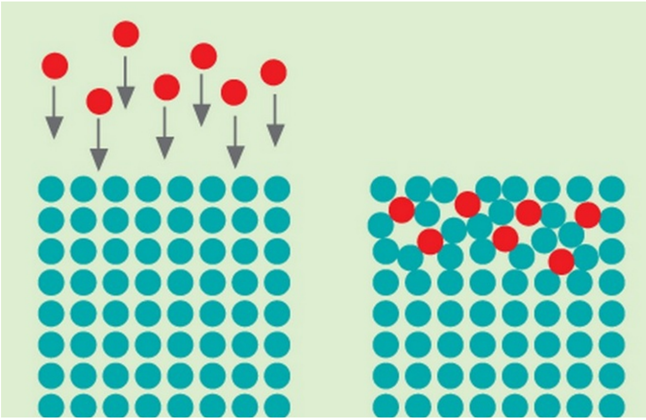




Silicon Wafer Processing: Example of Diffusion in Materials Science

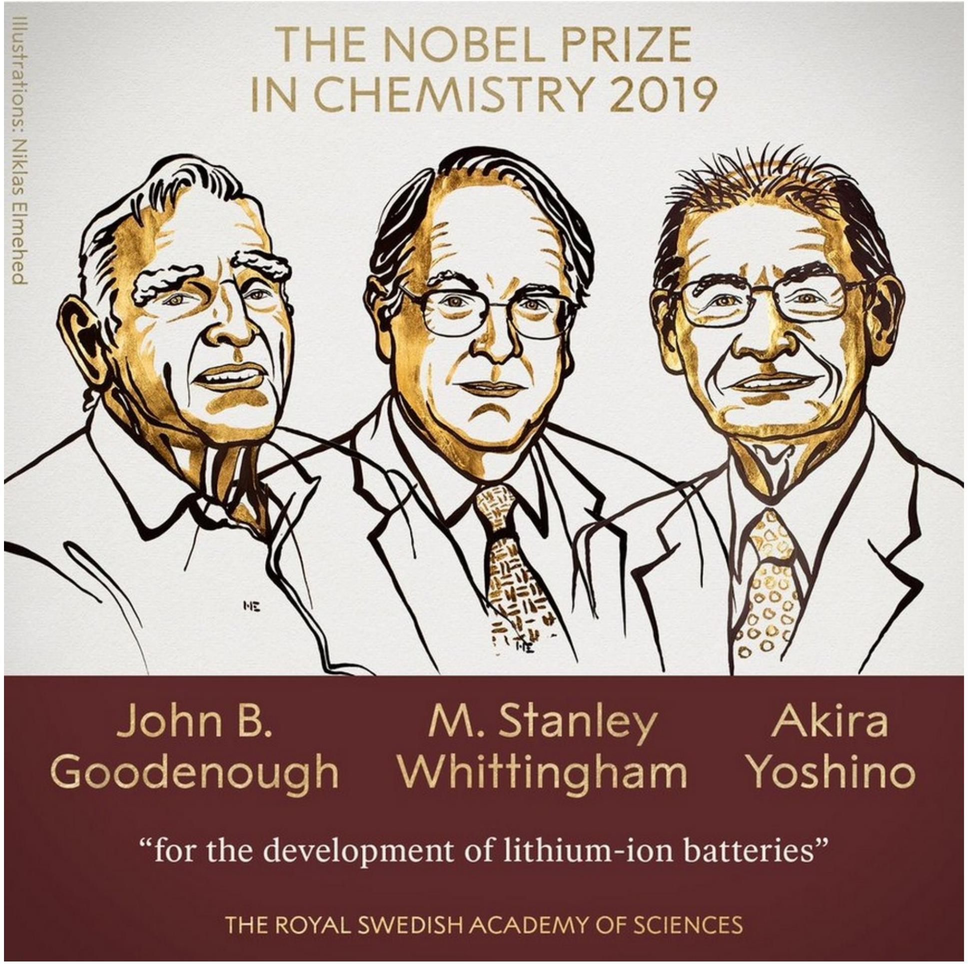
Intentional incorporation of impurities (e.g., boron, phosphorous)

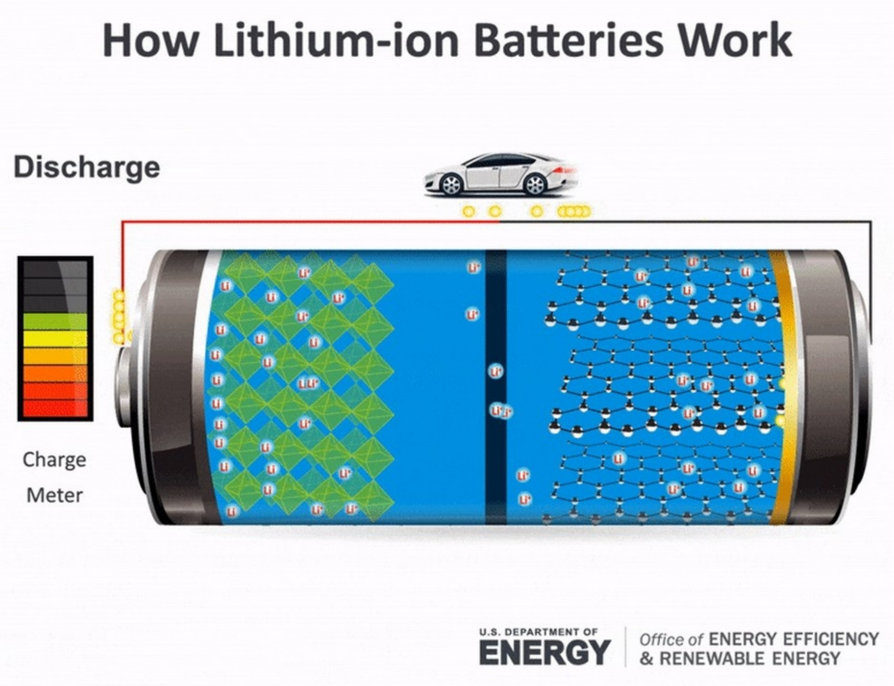
Step 1: Steady-state gas diffusion
or ion implantation



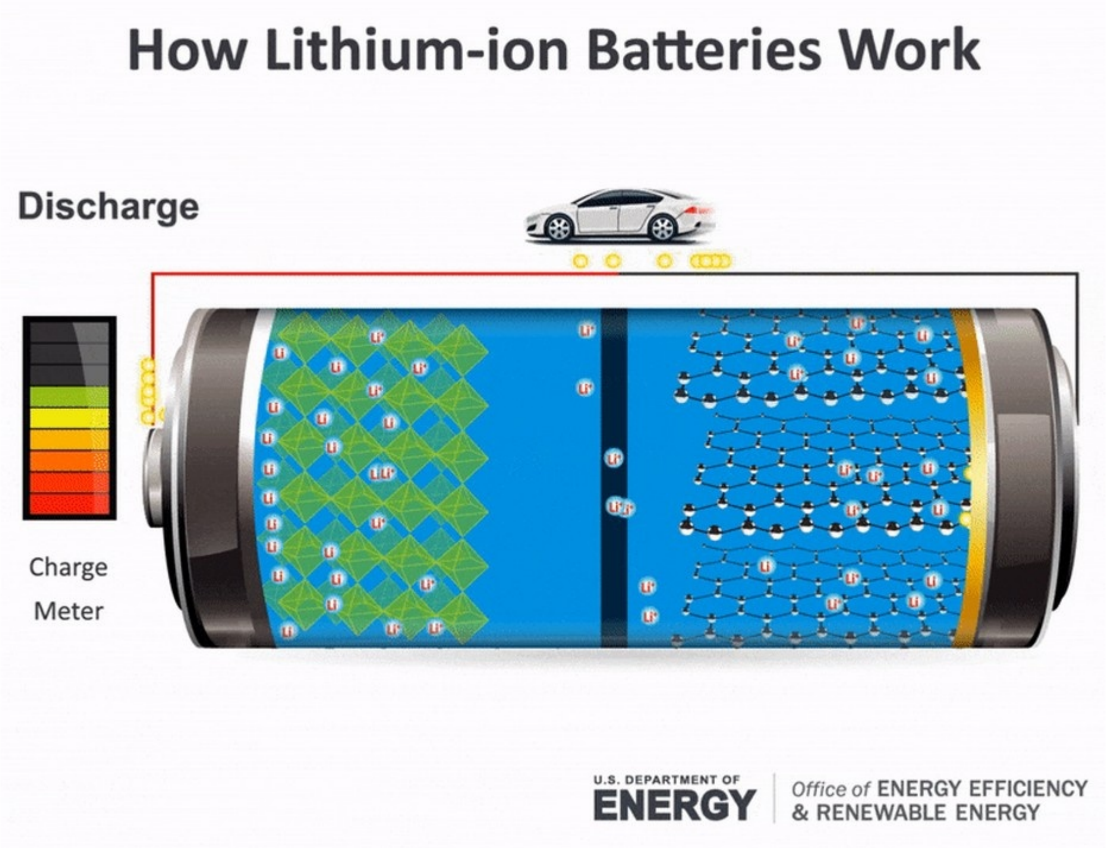
Step 2: Drive-in process (higher temperature)
→ uniform distribution of impurities

What is the materials science behind the Li-ion battery winning the 2019 Nobel prize in Chemistry?



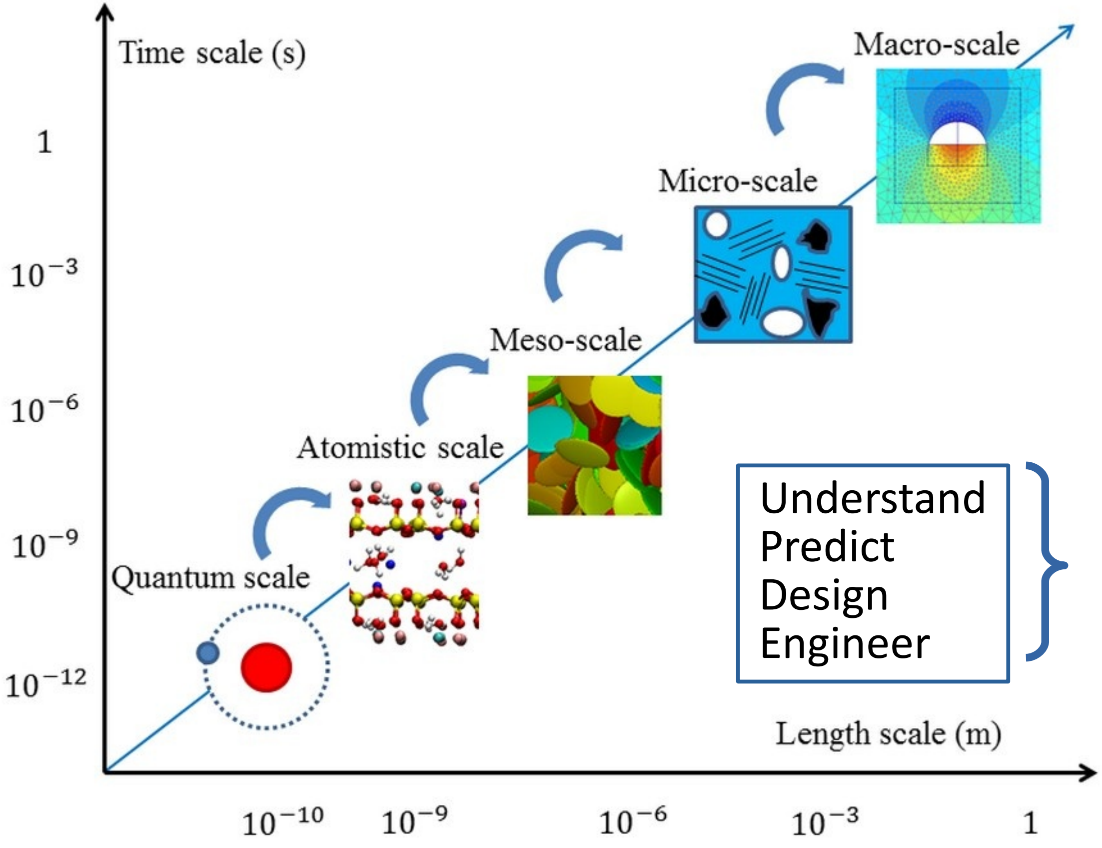


Diffusion of Lithium ions in a battery



https://www.energy.gov/science/doe-explainsbatteries

What is Modeling and Simulation in Materials Science?



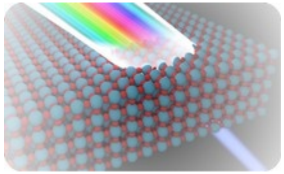
Understand
Predict
Design
Engineer

Material properties
New materials

TACC
TEXAS ADVANCED COMPUTING CENTER

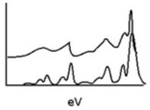
Research Overview and Philosophy

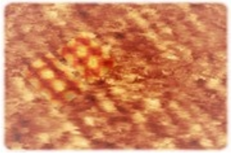
First-principles calculations



Enabled by HPC resources

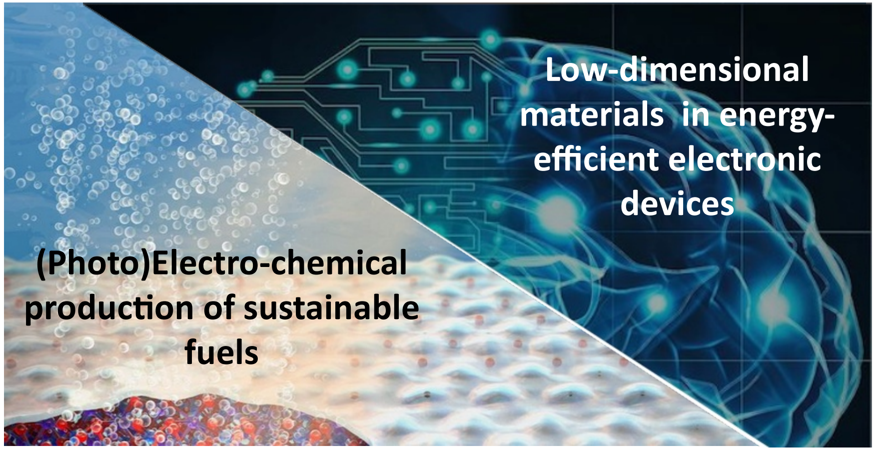
Experimental collaboration



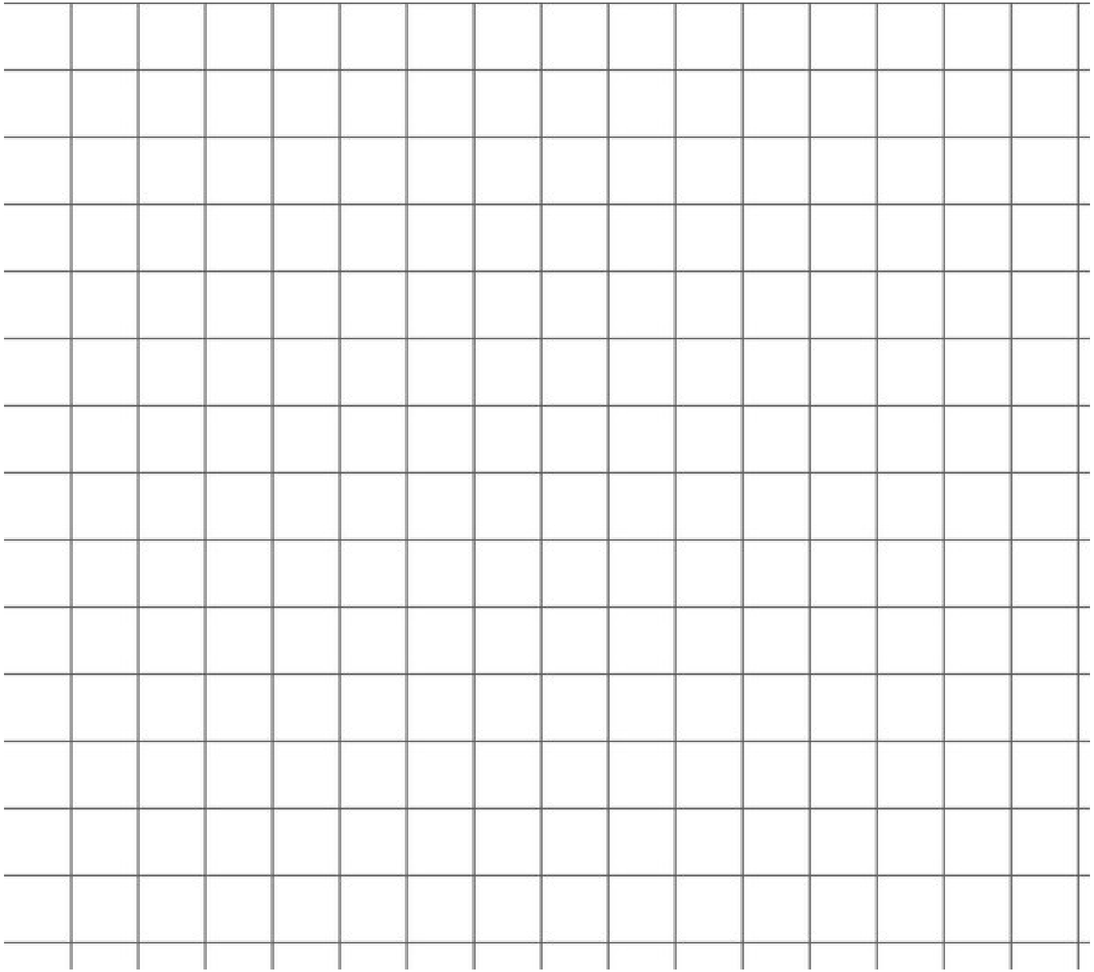


Spectroscopies, microscopies, ...

Harnessing Materials Imperfections for Energy Sustainability

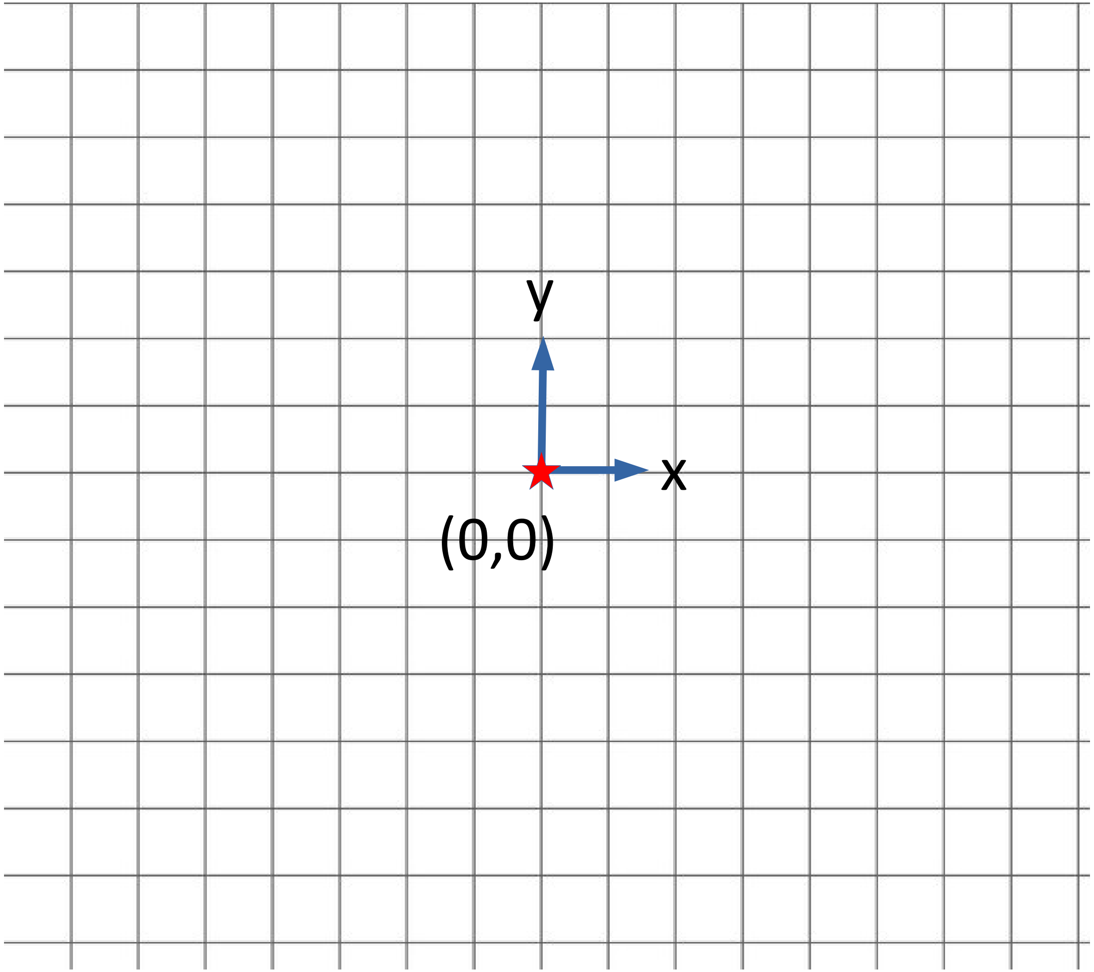


Random walk diffusion: an atomic model for diffusion



Rules for the random walker:

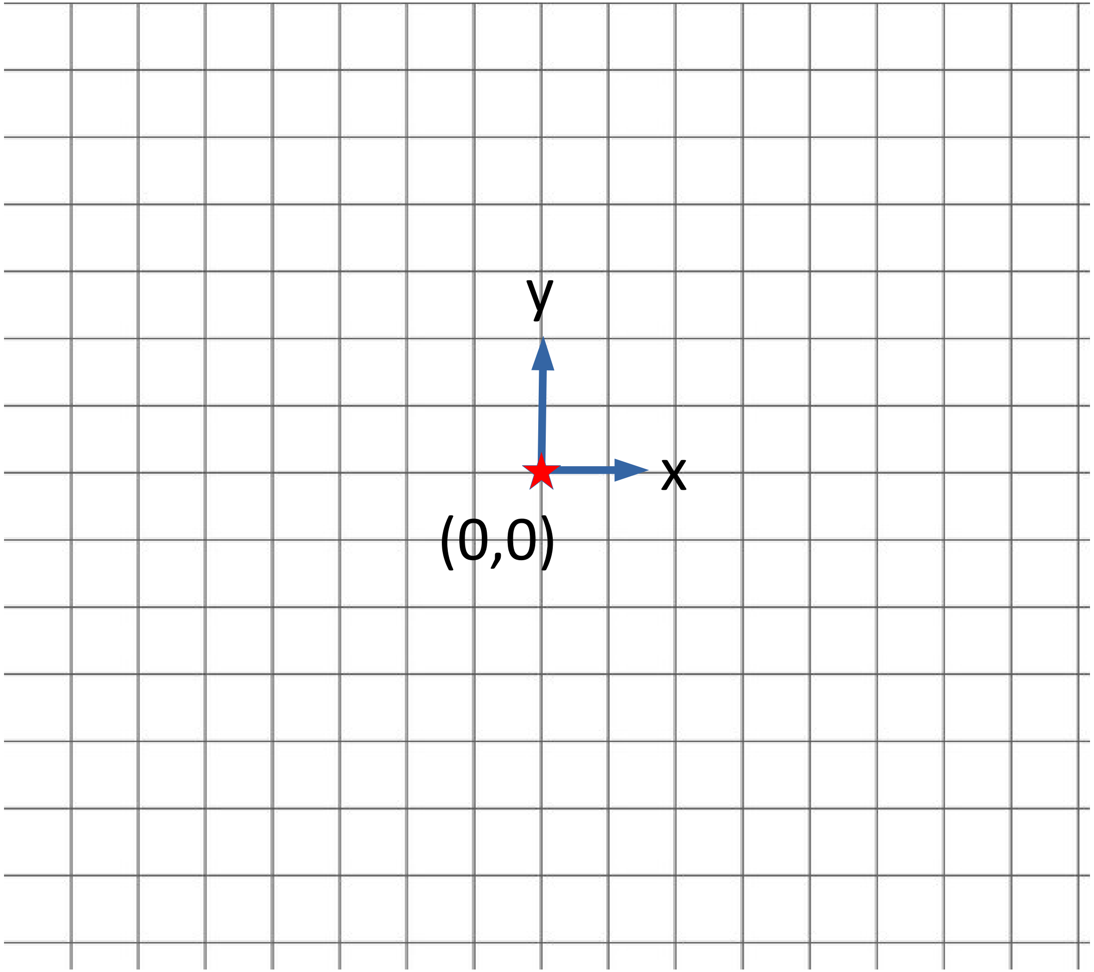
Random walk diffusion: an atomic model for diffusion



Rules for the random walker:

- divide time into *nt* discrete steps spaced by Δt time, where *nt* is an integer and Δt is a number
- can only move 1 space at each time step
- equal and random probability of moving up, down, left, right

Random walk diffusion: an atomic model for diffusion

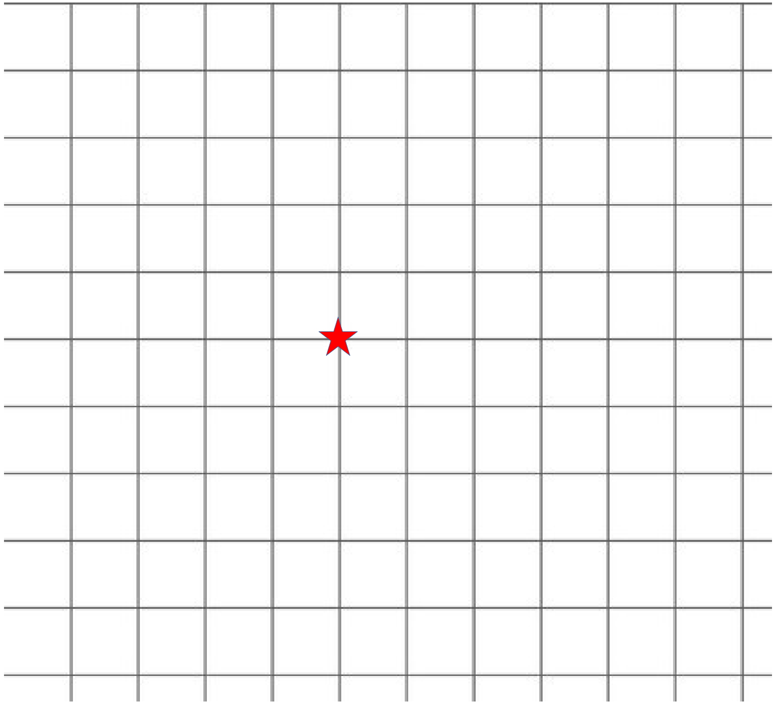


Rules for the random walker:

- divide time into *nt* discrete steps spaced by Δt time, where *nt* is an integer and Δt is a number
- can only move 1 space at each time step
- equal and random probability of moving up, down, left, right

If the random walker starts at the red star, where will it be after many (n >> 1) steps?

An implementation of the 2D Random Walk Diffusion



- Example: nt = 10 time steps, Δt = 1 s
  Total time is $t = nt * \Delta t$ = 10 s

Start at origin $(x,y) = (0,0)$ at time $t = 0$

Use a random number generator and encode the following:

1 = move up
2 = move down
3 = move left
4 = move right

Random walk diffusion: a small simulation

https://rwd2d-mercury.runmercury.com/

A brief intro to Python

Python is a high-level, general-purpose programming language.

Its design philosophy emphasizes code readability.

This makes Python a great language to learn basic programming concepts, which can be used to learn other programming languages.

A brief intro to Python

Python is a high-level, general-purpose programming language.

Its design philosophy emphasizes code readability.

This makes Python a great language to learn basic programming concepts,
used to learn other programming languages.

Just like any other language, there are rules on how to write code.
We call these rules the ***syntax***.

For example, consider the role of the comma in these two sentences:

Let’s eat grandma vs Let’s eat, grandma

Computers need *exact* instructions, so poor syntax (e.g., typos) will lead to code errors,
Let’s learn some programming rules.

Let’s look at a Python code

vim lattice_2D_EXERCISE.py

```
# Random walk on a 2D square lattice

# UT MITE 2024 Summer activity

#----------------------------------------

# import relevant libraries,
#  contains useful functions already implemented
import numpy as np
import matplotlib.pyplot as plt

# we define a function that encapsulates all the code related to
#  generating the path for a random walker
def random_walk(nt):
    """
    Random walk on a 2D square lattice
    Inputs:
        nt [integer] = number of desired jumps (i.e., time steps)
    Outputs:
        x   [array] = x-coordinate at each time step
        y   [array] = y-coordinate at each time step
    """
    # array for x- and y-coordinates along hopping path
    x = np.zeros(nt+1)
    y = np.zeros(nt+1)
    rs2 = np.zeros(nt+1)
"lattice_2D_EXERCISE.py" 71L, 1986B                    1,1           Top
```

To make code easier to read for humans, we often put in descriptive ***comments***

Two types of comments: line comments (marked with #) and block comments (marked with “””)

Let’s look at a Python code

vim lattice_2D_EXERCISE.py

```python
# Random walk on a 2D square lattice

# UT MITE 2024 Summer activity

#-------------------------------------------------

# import relevant libraries,
#  contains useful functions already implemented
import numpy as np
import matplotlib.pyplot as plt

# we define a function that encapsulates all the code related to
#  generating the path for a random walker
def random_walk(nt):
    """
    Random walk on a 2D square lattice
    Inputs:
        nt [integer] = number of desired jumps (i.e., time steps)
    Outputs:
        x   [array] = x-coordinate at each time step
        y   [array] = y-coordinate at each time step
    """
    # array for x- and y-coordinates along hopping path
    x = np.zeros(nt+1)
    y = np.zeros(nt+1)
    rs2 = np.zeros(nt+1)
"lattice_2D_EXERCISE.py" 71L, 1986B                    1,1           Top
```

Python is ***modular***, so you can pick and choose to add different packages, which we will ***import***

numpy let’s us work more easily with numerical data

matplotlib is for generating plots

Let’s look at a Python code (advanced)

vim lattice_2D_EXERCISE.py

```
# Random walk on a 2D square lattice

# UT MITE 2024 Summer activity

#-----------------------------------------------

# import relevant libraries,
#  contains useful functions already implemented
import numpy as np
import matplotlib.pyplot as plt

# we define a function that encapsulates all the code related to
#  generating the path for a random walker
def random_walk(nt):
    """
    Random walk on a 2D square lattice
    Inputs:
        nt [integer] = number of desired jumps (i.e., time steps)
    Outputs:
        x   [array] = x-coordinate at each time step
        y   [array] = y-coordinate at each time step
    """
    # array for x- and y-coordinates along hopping path
    x = np.zeros(nt+1)
    y = np.zeros(nt+1)
    rs2 = np.zeros(nt+1)
"lattice_2D_EXERCISE.py" 71L, 1986B                        1,1           Top
```

If a piece of code gets used many times, you can put it in a ***function***

Here, we are defining a function called ***random_walk*** that takes in an input called ***nt*** to be specified later.

Let's look at a Python code

vim lattice_2D_EXERCISE.py

```
# Random walk on a 2D square lattice

# UT MITE 2024 Summer activity

#-------------------------------------------------

# import relevant libraries,
#  contains useful functions already implemented
import numpy as np
import matplotlib.pyplot as plt

# we define a function that encapsulates all the code related to
#  generating the path for a random walker
def random_walk(nt):
    """
    Random walk on a 2D square lattice
    Inputs:
        nt [integer] = number of desired jumps (i.e., time steps)
    Outputs:
        x   [array] = x-coordinate at each time step
        y   [array] = y-coordinate at each time step
    """
    # array for x- and y-coordinates along hopping path
    x = np.zeros(nt+1)
    y = np.zeros(nt+1)
    rs2 = np.zeros(nt+1)
"lattice_2D_EXERCISE.py" 71L, 1986B                    1,1           Top
```

We will use different types of variables to store different types of numerical data

Let’s look at a Python code

vim lattice_2D_EXERCISE.py

```python
# Random walk on a 2D square lattice

# UT MITE 2024 Summer activity

#------------------------------------------------

# import relevant libraries,
#  contains useful functions already implemented
import numpy as np
import matplotlib.pyplot as plt

# we define a function that encapsulates all the code related to
#  generating the path for a random walker
def random_walk(nt):
    """
    Random walk on a 2D square lattice
    Inputs:
        nt [integer] = number of desired jumps (i.e., time steps)
    Outputs:
        x   [array] = x-coordinate at each time step
        y   [array] = y-coordinate at each time step
    """
    # array for x- and y-coordinates along hopping path
    x = np.zeros(nt+1)
    y = np.zeros(nt+1)
    rs2 = np.zeros(nt+1)
```

"lattice_2D_EXERCISE.py" 71L, 1986B   1,1   Top

We use the ***numpy*** package to help generate our first variable

An implementation of the 2D Random Walk Diffusion

Objective	Concept	Python Representation
User chooses *nt* = number of time steps	Variable, integer	`nt`

An implementation of the 2D Random Walk Diffusion

Objective	Concept	Python Representation
User chooses *nt* = number of time steps	Variable, integer	`nt`

Keep track of position of random walker at each time step.
Let's assume it starts at the origin.
	Array (list of items)
	e.g., [3, 4.5, 8, -1]
	
	2D → *x* and *y* coordinate for each position
		Use the library numpy, shorthand is np:
		
		`x = np.zeros(nt+1)`
		`y = np.zeros(nt+1)`

An implementation of the 2D Random Walk Diffusion

Objective	Concept	Python Representation
User chooses *nt* = number of time steps	Variable, integer	`nt`

Keep track of position of random walker at each time step.
Let's assume it starts at the origin.
	Array (list of items)
	e.g., [3, 4.5, 8, -1]
	
	2D → *x* and *y* coordinate for each position
		Use the library numpy, shorthand is np:
		
		`x = np.zeros(nt+1)`
		`y = np.zeros(nt+1)`

Specify how the position changes at each time step.		`delx = np.array([?,?,?,?])` `dely = np.array([?,?,?,?])`

An implementation of the 2D Random Walk Diffusion

Objective	Concept	Python Representation
User chooses *nt* = number of time steps	Variable, integer	`nt`

Keep track of position of random walker at each time step.
Let's assume it starts at the origin.
	Array (list of items)
	e.g., [3, 4.5, 8, -1]
	
	2D → *x* and *y* coordinate for each position
		Use the library numpy, shorthand is np:
		
		`x = np.zeros(nt+1)`
		`y = np.zeros(nt+1)`

Specify how the position changes at each time step.		`delx = np.array([?,?,?,?])` `dely = np.array([?,?,?,?])`

Save each new position of the diffusion path	Index the array
	i.e., access a specific element
	"zero index"
		`x = [1,2,3]`
		`x[0] = 1`
		`x[1] = 2`

An implementation of the 2D Random Walk Diffusion

Objective

Repeat for *nt* times

Encode the random number to a change in position of the random walker

Concept

for loop
range function

Generate a (pseudo)-random number

Python Representation

Input:

```
for i in range(3):
    print(i)
```

Output:

```
0
1
2
```

```
np.floor(4* np.random.rand(nt))
```

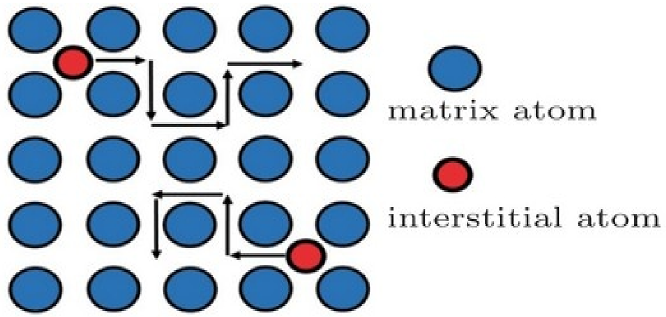
Generate random number b/t 0 and 1

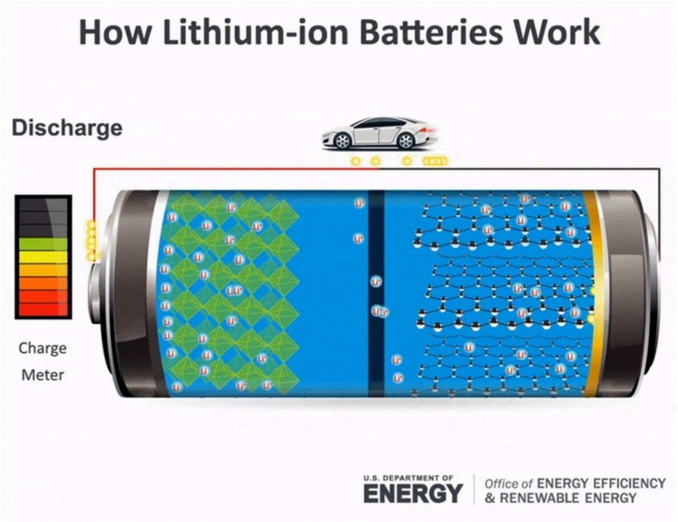
Random number b/t 0 and 4

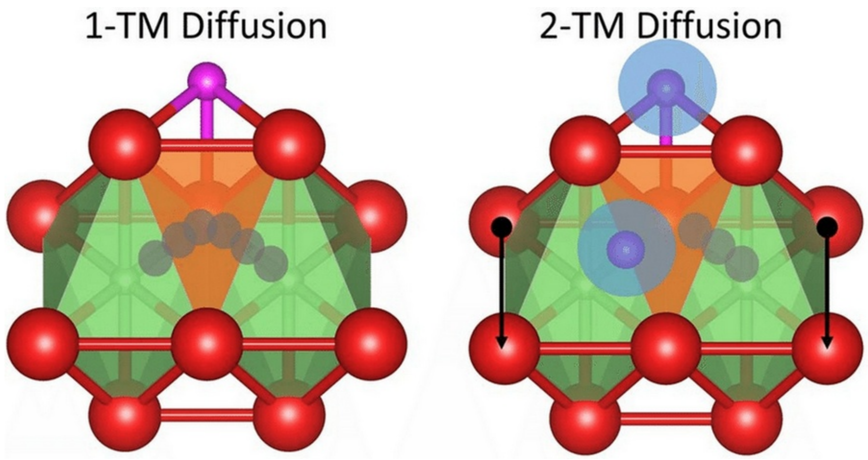
Random integer: 0, 1, 2, 3

Let’s try a little coding

Does the Random Walk Diffusion model have any correspondence with a real material?







Florian Schipper et al 2017 J. Electrochem. Soc. 164 A6220

Things learned today

Materials Science and Engineering and the Materials Science Tetrahedron

Diffusion for mass transport

Examples of diffusion

- Soda, air fresheners
- Impurities in semiconductors, Li-ion battery

Computation and simulation of diffusion

- Random Walk Diffusion

More questions? Reach out at https://wangmaterialsgroup.com