y-axis) of the "age" of each species. In other words, plot a heatmap where at a given location you plot the time since a species was last replaced at that location. You should hopefully find that as time goes on, species tend to stay around longer and longer.
Hint: only plot every 100th time step along the y-axis to speed up your code.

(e) Based on the above plot, make a comment about *mass extinctions*.

5.2 Chemical Reactions

CO_2 is soluble in water, which it may further react with to form carbonic acid:

$$CO_2 + H_2O \underset{k_2}{\overset{k_1}{\rightleftharpoons}} H_2CO_3. \tag{5.1}$$

We will consider a really small system where we have just $N_{CO_2} = 10000$ molecules of CO_2, but "infinitely" many H_2O molecules. We start with no carbonic acid. The rate of the forward reaction is

$$R_{\rightarrow} = k_1 \, N_{CO_2} \tag{5.2}$$

and backward reaction

$$R_{\leftarrow} = k_2 \, N_{H_2CO_3}, \tag{5.3}$$

where $k_1 = 10^{-3}$ and $k_2 = 1.0$.

If we had an infinite number of molecules, the concentration c would follow the ODEs

$$\frac{\mathrm{d}c_{\mathrm{CO}_2}(t)}{\mathrm{d}t} = -k_1 c_{\mathrm{CO}_2}(t) + k_2 c_{\mathrm{H}_2\mathrm{CO}_3}(t), \tag{5.4}$$

$$\frac{\mathrm{d}c_{\mathrm{H}_2\mathrm{CO}_3}(t)}{\mathrm{d}t} = k_1 c_{\mathrm{CO}_2}(t) - k_2 c_{\mathrm{H}_2\mathrm{CO}_3}(t). \tag{5.5}$$

(a) Solve and plot the solution to the ODEs with $c_{\mathrm{CO}_2}(0) = 1$ and $c_{\mathrm{H}_2\mathrm{CO}_3}(0) = 0$ for $t \in [0, 20]$ using any method you wish and plot $c_{\mathrm{H}_2\mathrm{CO}_3}(t)$.

For smaller systems, such as $N = 10000$ there will be a lot of noise. To simulate this we will use the Gillespie method.

(b) Simulate the system five times using the Gillespie algorithm and plot $N_{\mathrm{H}_2\mathrm{CO}_3}(t)/10000$ in the same plot as $c_{\mathrm{H}_2\mathrm{CO}_3}(t)$. Use `plt.step(..., where='post')` to make the plot.

(c) Make the same the plots for $N = 1000$, $N = 10000$, $N = 100000$, and $N = 1000000$.

(d) *(optional)* As N is increased, the simulation becomes slower and slower. For large N, however, we can use the approximate τ-stepping method. Implement this and plot the solution for $N = 1000$, $N = 10000$, $N = 100000$, and $N = 1000000$ using $\Delta t = 0.1$.

5.3 Local Voter model

Independent exercise

Consider an American neighbourhood of $N = 50$ residents. Each resident identifies as democratic (D) or republican (R).

The following four events can happen (with rates[1]):

$$\begin{cases} \text{A democratic randomly becomes republican,} & \text{rate} = 0.1D \\ \text{A republican randomly becomes democratic,} & \text{rate} = 0.1R \\ \text{A republican convinces a democratic to become republican,} & \text{rate} = 0.01DR \\ \text{A democratic convinces a republican to become democratic.} & \text{rate} = 0.01RD \end{cases}$$

(a) Explain what the rates mean. Why are the ‘convincing‘ rates proportional to $D \cdot R$?

(b) Initialize a system with $R = 25$ and $D = 25$ and simulate the above rate system using the Gillespie method for 500,000 steps.

(c) Plot R and D as a function of time and discuss the result.

We now introduce a third type of resident: undecided (U). The rules are then updated such that when someone is convinced to leave their party, they become undecided, and thus needs to be convinced twice to be

[1] In reality, these rates are probably near-zero.

converted to the other party. The random rates are thus

$$\begin{cases} \text{A democratic randomly becomes undecided,} & \text{rate} = 0.1D \\ \text{A republican randomly becomes undecided,} & \text{rate} = 0.1R \\ \text{An undecided randomly becomes democratic,} & \text{rate} = 0.05U \\ \text{An undecided randomly becomes republican,} & \text{rate} = 0.05U \end{cases}$$

and the convincing rates

$$\begin{cases} \text{A republican convinces a democratic to become undecided,} & \text{rate} = 0.01DR \\ \text{A republican convinces an undecided to become republican,} & \text{rate} = 0.01UR \\ \text{A democratic convinces a republican to become undecided,} & \text{rate} = 0.01RD \\ \text{A democratic convinces an undecided to become democratic.} & \text{rate} = 0.01UD \end{cases}$$

(d) Initialize a system with $R = 0$, $U = 50$ and $D = 0$ and simulate the above rate system using the Gillespie method for 500,000 steps.

(e) Plot R and D as a function of time and discuss the result.

(f) (*optional*) How would a mean-field ODE description [e.g. similar to that of Eq. (5.4)] for this system behave?

5.4 Damped Harmonic Oscillator with stochastic noise

Consider the system of equations:

$$\Delta X = Y\Delta t$$
$$\Delta Y = (-\omega_0^2 X - \delta Y)\Delta t + \sigma\Delta W(t)$$

Where ω_0, δ and σ are parameters.

(a) Choose $\omega_0 = 1$, $\delta = .1$ and $\sigma = 0$. Simulate the system for

$t \in [0; 100]$ using the Euler-Muryama method with $\Delta t = 0.01$. Compare this to the same system but with $\sigma = 1$. What does the stochasticity do?

(b) Now choose $\delta = 0, \sigma = 1$ and simulate for $t \in [0; 1000]$. What happens to the system? Why is this?

(c) Fix $\delta = .1$ and vary $\sigma \in [0.1; 10]$. How does the amplitude of oscillations grow?

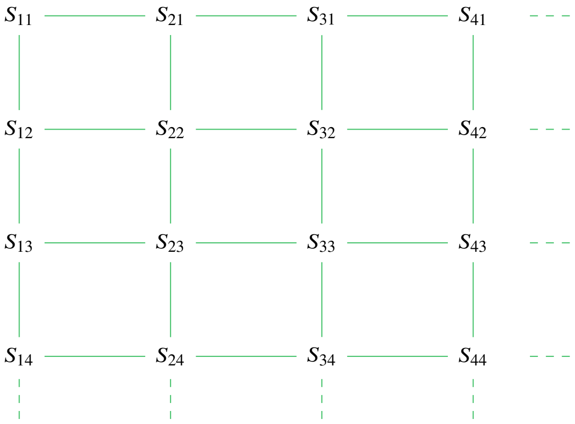
5.5 Ising Model

The Ising model is a simple model for ferromagnetism. Atoms with spin $S \in \{-1, 1\}$ are given fixed locations and their spin interactions are limited to nearest neighbours. The total energy (Hamiltonian) of the system is therefore

$$\mathcal{H} = -\sum_{\langle ij\rangle} S_i S_j, \tag{5.6}$$

where $\langle ij \rangle$ denotes neighbouring sites such that each interaction is only counted once.

In two dimensions, the simplest Ising model looks like



where, for instance, the neighbours of S_{22} are S_{21}, S_{12}, S_{32}, S_{23}.

(a) Make a function that takes an $N \times N$ matrix $\mathbf{S}$ and calculates $\mathcal{H}$ assuming a periodic system.

In the lowest possible energy state all spins are aligned, i.e. they all equal 1 or -1, which corresponds to full magnetisation. However, fluctuations at finite temperatures will drive the system away from this minimum. The likelihood of a given spin configuration is given by the Boltzmann distribution

$$p(\mathbf{S}) = \frac{e^{-\mathcal{H}/T}}{Z}, \tag{5.7}$$

where T is the temperature and Z is the partition function. The magnetisation of a specific spin configuration is simply

$$m(\mathbf{S}) = \langle S_i \rangle. \tag{5.8}$$

We shall be interested in the time-averaged magnetisation as a function temperature T. We will use the Metropolis–Hastings algorithm to simulate the time evolution of the system.

(b) Make a function that calculates the change in energy ΔE from a spin flip at a given location i.
Speed hint: this calculation can be done without evaluating the full Hamiltonian twice, as the change in energy only depends on the local spins.

(c) Make a function that randomly accepts or rejects a spin flip based on ΔE with probability $\alpha = \min(1, e^{-\Delta E/T})$.

To take a Metropolis–Hastings step, a random random spin is chosen and flipped according to the above rule.

(d) Start with a random initialisation of **S** with $N \geq 5$ depending on the speed of your code ($N = 1000$ should be possible with `numba`). Let the system run for $1000\,N^2$ time steps to equilibrate the system. Plot the spin configuration (`plt.imshow` with `interpolation='nearest'`) at different times during the simulation. Use $T = 0.5$. Comment on the result.

(e) After initialisation let the system evolve another $1000\,N^2$ time steps, and store at each 100th time step the energy and the magnetisation. Do this for temperatures between 0.1 and 5.0 (e.g. `np.linspace(0.1, 5.0, 100)`) and plot the average absolute magnetisation and energy as a function of temperature.

You should find that the system spontaneously magnetises at low temperature. In particular, there is a *phase transition* at a critical temperature T_c below which the material is ferromagnetic.

(f) Determine T_c as best you can. What ways could you improve your estimate?

(g) *(optional)* We are only updating one spin per Monte Carlo step. Do you have ideas for updating many spins per step?

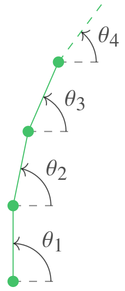
(h) *(optional)* A real magnetic has many, many spins ($\sim$ Avogadro's number). This basically corresponds to $N \to \infty$ (the thermodynamic limit). Is your estimate of T_c a good one for this? How would you find out?

Incredibly, this model has actually been solved analytically in the thermodynamic limit. The critical temperature is exactly $T_c = 2/\log(1+\sqrt{2})$.

5.6 Finite Temperature Euler Buckling

Note: For this exercise you will need to use `numba.njit` to make it run reasonably fast.

If you put enough weight on top of a spring at some point it will *buckle* meaning it will collapse under the weight. Here we will consider a simple model of a spring: a chain of $N = 15$ links that are connected by torsional springs. We will denote the angle that link i makes with the x-axis as θ_i as illustrated in the following figure:


θ_4
θ_3
θ_2
θ_1