Assignment 5

Note: All matrix calculations in this assignment (multiplications, additions, eigenvaluee and singular value decomposition) should use the appropriate `numpy` functions.

An interesting matrix. This $L\times L$ matrix depends on two positive parameters r_u and r_d. The matrix is tri-diagonal with r_d on the sub-diagonal, r_u on the super-diagonal, diagonal entries that make the row sums equal to zero, and all other entries equal to zero.

$$R = \begin{pmatrix} -r_u & r_u & 0 & \cdots & 0 \\ r_d & -(r_u + r_d) & r_u & \ddots & \vdots \\ 0 & r_d & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & -(r_u + r_d) & r_u \\ 0 & \cdots & 0 & r_d & -r_d \end{pmatrix} . \tag{1}$$

The matrix describes a random "hopping" process in which a "walker" makes random "hops" from one "site" to a neighboring site. There are L sites, labeled $1, 2, \cdots, L$. At time t, the walker is at site[1] $X(t)$. The probability that $X(t) = k$ is

$$p_k(t) = \Pr(\, X(t) = k \,) \ .$$

The random process involves $k \to k+1$ and $k \to k-1$ transitions. The $k = tok+1$ transitions are "up hops" and the $k \to k-1$ ones are "down hops". Up hops from $k = L$ are not allowed, nor are down hops from $k = 1$. Up hops happen at random times with rate parameter r_u. This means that the probability of an up hop in a small interval of time of length dt is equal to $r_u\, dt$. The probability of a down hop is $r_d\, dt$. The probability of some hop (either up or down) is $(r_u + r_d)\, dt$. In this derivation, we assume that these probabilities are so small that we may neglect the possibility that there is more than one hop in the dt time interval.

The *occupation probabilities* $p_k(t)$ satisfy a system of differential equations derived using reasoning involving conditional probability. We denote conditional probability using the symbol "|", so $\Pr(A|B)$ is the probability of A conditional on B. This is the probability that A happens if you already know that B happened. Conditional probabilities satisfy *Bayes' rule*, which is

$$\Pr(\text{both } A \text{ and } B) = \Pr(A|B)\Pr(B) \ .$$

[1] It is common to denote random quantities with capitol letters and possible values of a random variable with lower case letters. If $k \in \{1, 2, \cdots, L\}$ is one of the sites, then is is possible that $X(t) = k$.

You can think of this as intuitive, but in theoretical probability is taken as the definition of conditional probability.

We also need the *law of total probability*. Suppose B_1, B_2, and B_3 are three "disjoint" events (only one of them can occur) that "cover" all possibilities (one of them must happen), then if A happens, it happens with one of the events B_1, B_2, or B_3. The total probability of A is the sum of the three "partial" probabilities

$$\begin{aligned}\Pr(A) = &\ \Pr(A \text{ and } B_1) \ + \ \Pr(A \text{ and } B_2) \ + \ \Pr(A \text{ and } B_3) \\ &= \Pr(A|B_1)\Pr(B_1) + \Pr(A|B_1)\Pr(B_1) + \Pr(A|B_1)\Pr(B_1)\ .\end{aligned}$$

For the evolution of the occupation probabilities $p_k(t)$, we use the events

$$\begin{aligned}A &= \{\, X(t+dt) = k\} \\ B_1 &= \{\, X(t) = k-1\} \\ B_2 &= \{\, X(t) = k\} \\ B_3 &= \{\, X(t) = k+1\}\end{aligned}$$

The three B events cover all possibilities for A because only one hop can happen in time dt, and because hops only go to neighbors. The law of total probability in this case becomes

$$\begin{aligned}\Pr(X(t+dt)=k) = &\ \Pr(X(t+dt)=k \mid X(t)=k-1)\cdot\Pr(X(t)=k-1) \\ &+ \Pr(X(t+dt)=k \mid X(t)=k+1)\cdot\Pr(X(t)=k+1) \\ &+ \Pr(X(t+dt)=k \mid X(t)=k) \qquad \cdot\Pr(X(t)=k)\end{aligned}$$

The hopping rate model is expressed technically as formulas for the conditional probabilities on the right. For example, a $k-1 \to k$ transition is an up hop and has probability $r_u\,dt$:

$$\begin{aligned}\Pr(X(t+dt)=k \mid X(t)=k-1) &= \qquad r_u\,dt \\ \Pr(X(t+dt)=k \mid X(t)=k+1) &= \qquad r_d\,dt \\ \Pr(X(t+dt)=k \mid X(t)=k) \qquad &= 1-(r_u+r_d)\,dt\end{aligned}$$

The last probability above is the probability not to hop, which is one minus the probability to hop, and the hop may be either up or down. These probability formulas have to be modified a little if $k=1$ (no down hops) or $k=L$ (no up hops). With some algebra, this leads to

$$\frac{p_k(t+dt)-p_k(t)}{dt} = r_u\,p_{k-1}(t) - (r_u+r_d)\,p_k(t) + r_d\,p_{k+1}(t)\ . \tag{2}$$

The left side of this is $\frac{d}{dt}p_k(t)$.

The occupation probabilities may be organized into a row vector

$$p(t) = \begin{pmatrix} p_1(t) & p_2(t) & \cdots & p_L(t)\end{pmatrix}\ .$$

You can check that the system of differential equations (2) may be formulated as a vector/matrix differential equation involving the transition rate matrix R from (1):

$$\frac{d}{dt}p(t) = p(t)R \; . \tag{3}$$

This rate matrix (or any continuous time Markov transition rate matrix) is singular because the row sums are equal to zero. In matrix/vector form, this may be expressed using the vector of all ones:

$$R\,\mathbf{1} = R \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} = 0 \; .$$

This corresponds to conservation of total probability (using associativity of matrix/vector multiplication)

$$\frac{d}{dt}\sum_{k=1}^{L} p_k(t) = \frac{d}{dt}p(t)\,\mathbf{1} = (pR)\,\mathbf{1} = p(R\mathbf{1}) = 0 \; .$$

You can get a non-singular rate matrix by allowing down hops from $k = 1$. This corresponds to the model in which a walker can hop down from $k = 1$, but if it does then it is lost from the system. The rate matrix for this is S, which is equal to R except that the (1,1) entry is $-(r_u + r_d)$, which allows for down hops to oblivion. The sum of the probabilities then can decrease in time, as

$$\frac{d}{dt}\sum_{k=1}^{L} p_k(t) = -r_d\, p_1(t) \; .$$

1. **Condition number for inversion.**

   Recall that for functions with a single input and output, the condition number is the relative change in the output divided by the relative change in the input, with the understanding that the change in the input is very small. In notation, let x be an input number, f a function, and $\dot{x}$ a small change in x (which was called Δx in earlier classes). The condition number is

   $$\kappa(x) = \frac{\frac{\dot{f}(x)}{f(x)}}{\frac{\dot{x}}{x}} \; . \tag{4}$$

   A first derivative approximation is

   $$\dot{f}(x) = f'(x)\,\dot{x} \; .$$

   This puts the condition number formula in the possibly more familiar form

   $$\kappa(x) = f'(x)\,\frac{x}{f(x)} \; . \tag{5}$$

You can verify that the scalar condition number κ of (5) is dimensionless, i.e., a "pure number".

It is common (but not always correct) to define the condition number of matrix and vector operations using norms and worst case relative sensitivities. If A is a matrix and $f(A)$ is a matrix function of A (such as $f(A) = A^2$ or $f(A) = A^{-1}$, or $f(A) = e^A$), the condition number is defined as (4) but with the worst case perturbation $\dot{A}$. The perturbation $\dot{f}(A)$ is given by matrix perturbation theory, which depends on which function f is being used.

$$\kappa_f(A) = \max_{\dot{A} \neq 0} \frac{\frac{\left\|\dot{f}(A)\right\|}{\|f(A)\|}}{\frac{\left\|\dot{A}\right\|}{\|A\|}} \tag{6}$$

The perturbation formula is

$$\dot{(A^{-1})} = -A^{-1}\dot{A}\,A^{-1} \ .$$

Show that, for any matrix norm (writing κ for the condition number of the inverse matrix computation),

$$\kappa(A) \leq \left\|A^{-1}\right\| \ \|A\| \ . \tag{7}$$

An inequality is called *sharp* (a somewhat vague term) if it the best possible inequality of a certain form. For example, "sharp" might mean that an inequality $u(x) \leq v(x)$ for all x has some x with $u(x) = v(x)$. Show that the condition number inequality (7) is sharp in the matrix 2 norm. *Hint.* One step might be to show that the right side of (7) is $\sigma_1(A)/\sigma_n(A)$ (the largest and smallest singular values of A).

2. **Singular value perturbation theory.**

Work out first order perturbation theory for singular values for the case where singular values are distinct. If $\sigma_1(A) > \sigma)2(A) > \cdots$ are the singular values of A, and v_j and u_j are corresponding normalized right and left singular vectors, find a formula

$$\dot{\sigma}_k = ??(\text{something involving } \dot{A}, \text{ and the singular vectors})?? \ .$$

Hint. The formula looks like first order perturbation formulas for eigenvalues.

Do computations to show that (using your formula for $\dot{\sigma_j}$ and the computed SVD of A)

$$\sigma_j(A + s\dot{A}) = \sigma_j(A) + s\dot{\sigma}_j + cs^2 + O(s^3) \quad \text{as } s \to 0 \ .$$

The code can be something like the order of accuracy checks from an earlier assignment using a sequence s, $\frac{1}{2}s$, $\frac{1}{4}s$, $\cdots$, and showing that $\Delta\sigma_j - s\dot{\sigma}_j$

is order s^2 as $s \to 0$. The code should be set up to take as input matrices A and $\dot{A}$ of whatever size and shape. One test case might be

$$A = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \quad , \quad \dot{A} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

Try another non-trivial example also.

3. **Matrix exponential.**

   If A is a square matrix, the *matrix exponential* is defined by the power series of matrices

   $$e^A = \sum_{n=0}^{\infty} \frac{1}{n!} A^n \ . \tag{8}$$

   This is a matrix analogue of the Taylor series for the ordinary exponential

   $$e^a = \sum_{n=0}^{\infty} \frac{a^n}{n!} \ .$$

   The ordinary exponential has the property that $e^a e^b = e^{a+b}$. The corresponding matrix exponential formula is not true if matrices A and B do not commute

   $$e^A \, e^B \neq e^{(A+B)} \ , \quad \text{probably, if } AB \neq BA \ .$$

   The following restricted version is useful in many applications:

   $$e^{tA} \, e^{sA} = e^{(t+s)A} \ . \tag{9}$$

   The matrices being exponentiated, which are tA and sA, do commute.

   (a) Write a functions that computes and returns the sum (8) using a fixed number of terms (maybe 100, but not "hard wired"). This function will be slow, but it is a source of "ground truth" to test the fancier algorithms that follow.

   (b) Show that

   $$e^{\Delta t A} = I + \Delta t A + \frac{\Delta t^2}{2} A^2 + \cdots + \frac{\Delta t^p}{p!} A^p + O(\Delta t^{p+1}) \ .$$

   For part (d), we denote the order p approximation by

   $$E_p = I + \cdots + \frac{\Delta t^p}{p!} A^p \ .$$

   Hint. You can factor Δt^{p+1} out of the rest of the terms in the sum.

(c) Suppose n is a power of 2, which means $n = 2^k$ for some integer k. Find an algorithm to compute B^n (B being a square matrix) using $\log_2(n)$ matrix multiplications. *Extra credit, time permitting.* Find an algorithm to compute B^n using at most $2\log_2(n)$ matrix multiplies. *Hint.* Express n in in base 2, which means $n = d_0 + 2d_1 + 2^2 d_2 + \cdots$, where the "digits" have $d_j = 0$ or $d_j = 1$. Use this to express B^n as a product of matrices B^{2^j}.

(d) Choose n large and $\Delta t = 1/n$ small. We know $A = \Delta t A + \cdots + \Delta t A$ (n terms) so

$$e^A = \left(e^{\Delta t A}\right)^n \ .$$

Demonstrate numerically in some of the matrices (1) that the approximation of part (b) gives an order p approximation to the exponential.

$$e^A = E_p^n + O(\Delta t^p) \ .$$

Hints. Use the slow but "exact" (in exact arithmetic, with large enough n) method of part (a) for the "ground truth" on the left. Use the fast algorithm of part (c) to evaluate the right side. It suffices to take n as a power of 2. A simple and good way to measure the size of a matrix, without computing the SVD or doing some other optimization, is to use the *Frobenius norm*

$$\|B\|_F = \left(\sum_{jk} B_{jk}^2 \right)^{\frac{1}{2}} \ . \tag{10}$$

This in not the matrix norm derived from the vector 2-norm, but you can think of it as the vector 2-norm applied to the matrix B. The matrix norm that comes from a vector norm is

$$\|B\| = \max_{\|v\|=1} \|Bv\| \ .$$

Any norm like that has $\|I\| = 1$. But the Frobenius norm has $\|I\|_F = \sqrt{d}$ for the $d \times d$ identity matrix. You can code the Frobenius formula (10) in Python/numpy using a vectorized square operation applied to the elements of B and the `np.sum` function (it really is simpler than other norms). *Explanation.* We saw in panel method integration that local truncation error of order $p+1$ leads to global error of order p. The global error was one order larger because it is the accumulated result of many local errors. This is true here too, but the analysis is more complicated because the global error is not just the sum of the local errors. We will come back to this point when we discuss simulation methods for differential equations.

(e) Show that the matrix exponential is a way to express the solution matrix/vector differential equations, which may involve a time de-

pendent column vector or row vector or matrix

$$\frac{d}{dt}\,x(t) = A\,x(t) \tag{11}$$

$$\frac{d}{dt}\,p(t) = p(t\,)A \tag{12}$$

$$\frac{d}{dt}\,S(t) = A\,S(t) \tag{13}$$

$$\frac{d}{dt}\,S(t) = S(t)\,A\;. \tag{14}$$

To do this, show that if the *fundamental solution* is given by the formula

$$S(t) = e^{tA}$$

then S satisfies $S(0) = I$ and the differential equations (13) and (14). Conclude that the solutions of (11) and (12) are given by $x(t) = S(t)x(0)$ and $p(t) = S(t)p(0)$ respectively. *Hint.* Put tA into (8) and differentiate term by term with respect to t.

(f) *Optional, not to hand in even if you do it.* The sum formula for ordinary exponentials may be understood (among other ways) using Taylor series, written out and multiplied

$$\begin{aligned} e^a e^b &= \left(\,1 + a + \frac{1}{2}a^2 + \frac{1}{6}a^3 + \cdots\right)\left(\,1 + b + \frac{1}{2}b^2 + \frac{1}{6}b^3 + \cdots\right) \\ &= 1 + a + b + \frac{1}{2}a^2 + ab + \frac{1}{2}b^2 + \frac{1}{6}a^3 + \frac{1}{2}a^2 b + \frac{1}{2}ab^2 + \frac{1}{6}b^3 + \cdots \\ &= 1 + (\,a + b\,) + \frac{1}{2}\,(\,a + b\,)^2 + \frac{1}{6}\,(\,a + b\,)^3 + \cdots \\ &= e^{a+b}\;. \end{aligned}$$

Show that this calculation does not work for $e^A e^B$ if $AB \neq BA$ but does work for $e^{tA}e^{sA}$.

4. **Ill conditioned eigenvalue problem.** Let $P = R^{-1}\Lambda R$ be the eigenvalue/eigenvector decomposition of the matrix P from (1). For this exercise, we call that matrix A and write the eigenvalue/eigenvector decomposition as $A = R^{-1}\Lambda R$. The matrix A is symmetric when $r_u = r_d$ and the eigenvalue problem is well conditioned. When $r_u \neq r_d$ the matrix is not symmetric and the eigenvalue/eigenvector decomposition can be ill conditioned. It is a theorem that the (exact) eigenvalues and eigenvectors of A are real and distinct for any choice of r_u and r_d, but the computed eigenvalues and eigenvectors may have significant imaginary parts.

(a) Made scatterplots (visualization of eigenvalues as points in the complex plane) of the computed eigenvalues of A for $r_u = 2r_d$ and various values of L. They should be very close to real for small L but not

for larger L. Choose two or three scatterplots to upload that show this. Large imaginary parts show that the computed eigenvalues are far from the true ones.

(b) The condition number of R is large when L is large. Demonstrate this by plotting the condition number

$$\kappa(R) = \sigma_{\max}(R)/\sigma_{\min}(R)$$

as a function of L. If you make a semi-log plot, you should see (in an important L range) a linear graph that shows that κ grows exponentially with L. This is approximate for large L so small L values may deviate from the line. Large L values may "saturate" in floating point because it is hard for a computed condition number to be much larger than $1/\epsilon_{\text{mach}}$.

5. **An unstable algorithm.** A computational algorithm is *unstable* if it gives inaccurate results for a well conditioned problem. If the problem itself is ill conditioned, no algorithm should be accurate. The problem of finding the matrix exponential seems well conditioned, judging from the fact that the two algorithms of Exercise 3 give quite similar results. This can be confirmed by theory (omitted). An algorithm may be unstable because it uses the solution of an ill-conditioned sub-problem.

(a) Show that if $A = R^{-1}\Lambda R$, then

$$e^A = R^{-1}e^{\Lambda}R\ .$$

Show that e^{Λ} (as given by the power series (8)) is equal to the diagonal matrix with e^{λ_j} on the diagonal. The exponential of a diagonal matrix is diagonal. The exponentials of similar matrices are similar by the same transformation.

(b) Write a function that uses the eigenvalue/eigenvector decomposition of the matrix (1) to compute its exponential. Compute the difference (in the Frobenius norm) between this result and results from the algorithms of Exercise 3. The results should be good if $r_u \approx r_d$ or if L is not large. The result should be bad otherwise.