Assignment 3

Corrections: [none yet]

1. Suppose $A(s)$ is a differentiable family of positive definite symmetric matrices. Suppose that A has a Cholesky factorization $A(s) = L(s)L^t(s)$. Here, L is lower triangular. Find a formula for

$$\dot{L} = \frac{d}{ds}L(s)$$

as a function of L, L^{-1} and $\dot{A}$. It is a useful fact that the inverse of a lower triangular matrix is lower triangular.

2. Suppose A is an $n \times n$ matrix. Show that the power series

$$e^{tA} = \sum_0^\infty \frac{1}{n!} t^n A^n \ .$$

converges in any matrix norm. Hint: Never mind the first N terms. What matters is the "tails", the terms $M_n = \frac{1}{n!}t^n A^n$ for $n \geq N$. Choose N so that $N \geq 2\,|t|\,\|A\|$. Then

$$M_n = \frac{1}{n} t A M_{n-1} \ .$$

Therefore, if $n \geq N$,

$$\|M_n\| \leq \frac{1}{N}\,|t|\,\|A\|\,\|M_{n-1}\| \leq \frac{1}{2}\,\|M_{n-1}\| \ .$$

Therefore, if $n > N$, we have $\|M_n\| \leq \frac{1}{2^{n-N}}\,\|M_N\|$. The tail sum satisfies

$$\left\| \sum_{n=N}^\infty M_n \right\| \leq \left(\sum_{n \geq N} \frac{1}{2^{n-N}} \right) \|M_N\| \leq 2\,\|M_N\| < \infty \ .$$

The tail converges absolutely so the whole sum does too.

3. (Properties of the matrix exponential)

   (a) Show that the matrix $S(t) = e^{tA}$ satisfies a matrix differential equation and commutes with A:

$$\frac{d}{dt}S(t) = AS(t) = S(t)A \ .$$

(b) Show that $S(t) \to I$ as $t \to 0$.

(c) Show that $S(t_1)S(t_2) = S(t_1 + t_2)$. Hint: You can do this using part (d) below, or you can do it directly. For the direct argument, define $p = \frac{t_1}{t_1+t_2}$ and $q = \frac{t_2}{t_1+t_2}$ and write the formula $(p+q)^n = 1$ as

$$\sum_{j+k=n} \binom{n}{k} p^j q^k = \sum_{j+k=n} \frac{n!}{j!k!} p^j q^k = 1 \; .$$

Then write

$$S(t_1)S(t_2) = \sum_j \sum_k \frac{1}{j!} \frac{1}{k!} t_1^j t_2^k A^{j+k} = \sum_n \left(\sum_{j+k=n} \frac{n!}{j!k!} p^j q^k \right) t^n A^n \; .$$

(d) Suppose that $A = R\Lambda L$, where $RL = I$ and Λ is the diagonal matrix of eigenvalues. Allow the possibility that R, L, and Λ are complex. Show that $S(t) = Re^{t\Lambda}L$. Show that $e^{t\Lambda}$ is diagonal and find formulas for the numbers on the diagonal.

(e) The Python package `linalg` has routines for computing the eigenvalues and eigenvectors of a non-symmetric matrix. Use this to write a Python module that evaluates $S(t)$ given A and t. Test your module by calling it for the case $A = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$. In this case $S(t) = \begin{pmatrix} \cos(t) & \sin(t) \\ -\sin(t) & \cos(t) \end{pmatrix}$. Warning: because of roundoff in complex arithmetic, the computed result might not be real. The imaginary part of the computed $S(t)$ is all error.

4. Another way to compute the matrix exponential is to compute $S(\Delta t)$ using a few terms of the Taylor series, then take powers to apply the formula $S(n\Delta t) = S(\Delta t)^n$. Experiment with k terms of the Taylor series. Then choose n depending on t, and then $\Delta t = t/n$ so that k terms are enough to get $S(\Delta t)$ ten significant digits.

   (a) Compute the Taylor series numerically using *Horner's rule* which is

$$\begin{aligned} & x_0 I + x_1 A + \cdots + x_{k-1}A^{k-1} + x_k A^k \\ &= x_0 I + x_1 A + \cdots + x_{k-1}A^{k-1} \left(I + \frac{x_k}{x_{k-1}} A \right) \\ &= x_0 I + x_1 A + \cdots + x_{k-2}A^{k-2} \left(I + \frac{x_{k-1}}{x_{k-2}} A \left(I + \frac{x_k}{x_{k-1}} A \right) \right) \\ &= \cdots \\ &= x_0 \left(I + \frac{x_1}{x_0} A \left(I + \frac{x_2}{x_1} A \left(\cdots \right) \right) \right) \end{aligned}$$

   This gives a way to calculate the k term sum using k matrix multiplications. Doing it directly takes $O(k^2)$ multiplications.

(b) Compute a power B^n in the following way. First compute $B_1 = B^2$, then $B_2 = B_1^2 = B^4$, then $B_3 = B_2^2 = B^{2^3}$, etc. Then write $n = n_0 + 2n_1 + 4n_2 + \cdots$, where $n_j = 0$ of $n_j = 1$ (expand n base 2) and compute

$$B^n = \prod_{n_j=1} B_j \ .$$

Show that this computes B^n in $O(\log(n))$ matrix multiplications.

(c) Test your program on the A from problem 3.

5. Consider the differential equation system

$$\begin{aligned}
\frac{dx_k}{dt} &= p(x_{k+1} - x_k) + q(x_{k-1} - x_k) \ \text{ if } 1 < k < n \\
\frac{dx_1}{dt} &= p(x_2 - x_1) \\
\frac{dx_n}{dt} &= q(x_{n-1} - x_n) \ .
\end{aligned}$$

Here, x_k could represent the number of particles at spot k, and p represents the rate at which particles move to the left (from $k+1$ to k or from k to $k-1$) and q represents the rate at which they move to the right. Particles cannot go from $k = 1$ to $k = 0$ or from $k = n$ to $k = n+1$. Try your codes from Problem 4 and Problem 5 on this system. If you explore $p > q$ and n relatively large, you should find that the eigenvalue method breaks down because the eigenvalue/eigenvector calculation is too ill conditioned. You will need code to create the matrix A. Note (check) that $M = \sum_k x_k(t)$ is a constant. Start with $x_1(n) = 1$, and $x_k(0) = 0$ for $k < n$. The solution has $x(t) \to x_*$ as $t \to \infty$. You can find x_* by setting $Ax_* = 0$ and $\sum_k x_k = 1$. Here are some things to explore as time and interest permits

- Find values of n, $p > 0$, $q > 0$, and t so that the eigenvalue method breaks down.
- The eigenvalues of A are all real. Monitor the imaginary parts of the computed eigenvalues.
- The problem of finding e^{tA} for this problem is well conditioned. The eigenvalue method is an unstable algorithm.

6. Write Python code to compute the matrix product AB when A and B are $n \times n$ matrices. You can initialize the matrices how you want because the values are unimportant. Do it in three ways:

- Scalar loop, as

```
for i in range(n):
    for j in range(n):
        c[i,j] = 0.
        for k in range(n):
            c[i,j] += a[i,k]*b[k,j]
```

- Hand coded vector loops, as (names and syntax wrong, please fix)

  ```
  c[i,j] = np.sum( a[i,:]*b[:,j] )
  ```

- The numpy matrix multiply routine, as

  ```
  c = np.linalg.matrixMultiply(a, b)
  ```

Learn how to time sections of Python code. Print the times for the three methods as a function of n for an interesting range of n. Comment on the differences.