On computing the determinant...

In this tutorial you will compare two methods for computing the determinant of a matrix. You have probably learned what the determinant is following this algorithm:
Let $A \in GL(n, n)$ $(n > 1)$ be a square matrix.

1. If $n = 2$ then $\det(A) = A_{11}A_{22} - A_{12}A_{21}$.
2. If $n > 2$ then construct the k^{th} submatrix $A^{(k)}$ by deleting the first column and k^{th} row of A. Then compute the determinant as

$$\det(A) = \sum_{k=1}^{n} (-1)^{k+1} A_{k1} \det(A^{(k)})$$

Of course, you can pick any row or column of A to construct submatrices.
This is a recursive definition: only the determinant of a 2×2 matrix is defined explicitly, in all other cases the determinant is computed as a sum of determinants of smaller matrices.
An alternative way to compute a determinant is offered by the LUP-decomposition:

$$\det(A) = \det(P^t LU) = \det(P) \det(L) \det(U) = \sigma \prod_{i=1}^{n} U_{ii}$$

since both L and U are triangular and L has diagonal entries equal to 1. The sign is $\sigma = 1$ if P has an even number of row interchanges and $\sigma = -1$ if it has an odd number.

Exercise A

Write a function that computes the determinant of a matrix recursively, following the algorithm above. Check its output for a random matrix against the scipy.linalg.det command.
Next, write a script that

1. loops over matrix sizes $n = 2, \ldots, 9$;
2. for each matrix size generates a random matrix;
3. computes the determinant of that matrix;
4. uses the time function from the time module to compute the wall time taken;
5. plots the wall time versus the matrix size on a semilogarithmic scale.

Exercise B

Use the LUP-decomposition function we discussed in lecture 6 and use it in a function to compute the determinant of a $n \times n$ matrix.
Next, write a script that

1. loops over matrix sizes $n = 2, \ldots, 9$;
2. for each matrix size generates a random matrix;
3. computes the determinant of that matrix;
4. uses the time function from the time module to compute the wall time taken;
5. plots the wall time versus the matrix size on a semilogarithmic scale.

Discussion

Compare the plots of wall time versus matrix size. Which method is faster for larger matrices? How fast does the wall time increase with n?