On computing the determinant more efficiently...

In last weeks tutorial, you programmed a function for computing the determinant of a matrix. Your function was stable and accurate but not efficient. This week, you will take a different approach.

1. Using some small example matrices, convince yourself of the following facts:
   - The determinant satisfies $\det(AB) = \det(A)\det(B)$.
   - If an $n \times n$ matrix A is upper or lower triangular, then $\det(A) = \prod_{i=1}^{n} A_{ii}$.
   - If P is a permutation matrix, then $\det P = \pm 1$.
2. Deduce from the above that, for any $n \times n$ matrix A
$$\det(A) = \pm\det(PA) = \pm\det(LU) = \pm\det(L)\det(U) = \pm\prod_{i=1}^{n} U_{ii}$$
3. Write a function that computes the determinant of a general $n \times n$ matrix A *up to the sign*. Use the LUP decomposition code from the course repository. Compare this function to the one you wrote last week. How do they compare in terms of stability, accuracy and efficiency?
4. In order to compute the determinant with the correct sign, you must find the determinant of P. The rule is simple: the determinant of the identity matrix is 1. Every time you swap two rows, the determinant is multiplied by -1. Add an output variable to the LUP decomposition function that is 1 for an even number of row swaps and -1 for an odd number. Use this new output variable to compute the determinant with the correct sign.

Discussion: What is the order of complexity of the new method of computing the determinant? In class, we saw that the order of complexity of the recursive method is $\mathcal{O}(n!)$. How do these orders compare?