1. Code: srctargint.m

b Numerical rank of interaction matrix, given by kernel \( \frac{1}{2\pi} \log \frac{1}{||x-y||} \)

Numerical rank of interaction matrix, with 50 logarithmically spaced \( n \) values between 1 and 1000.

For \( n = 1 \) to 15, matrix is full rank. After that, the rank still increases with \( n \), but more gradually. When \( n \approx 100 \). The numerical rank appears to maximize at 21, with some random matrices generating slightly larger ranks.

c Rank for \( n = 100 \) with tolerance of \( 10^{-10} \) is 17. Leaving out constants of order 1, the number of multipole coefficients \( P \) needed depends on \((R/b)^n\), where \( R \) is the radius of the circle that contains all the sources, and \( b \) is the radius of the circle that excludes all the targets, and \( n \) is the number of coefficients. In this case \( R = \sqrt{2}/2 \), and \( b = 3/2 \) so \( R/b \approx 0.47 \). Thus, to have a total error of \( 10^{-10} \), we need \( \approx 31 \) \( p \) coefficients, which is about 50% greater than the rank of the matrix. However, we know that the rank of the matrix is a lower bound on the number of \( p \) coefficients needed, so it is expected for \( n \) to be larger than the rank, however, it is on the same order of magnitude.

2. Code: multipole.m

\( l^2 \) norm of the real part of the vector potentials is \( 9.721396404 \times 10^4 \).

Setting up timers on the multipole method and 6 points done by the brute-force matrix multiplication method, we find that the multipole method takes \( \approx 7.14 \) seconds to complete, whereas the brute-force method would take an estimated \( 5.5 \times 10^4 \) seconds (\( \approx 15 \) hours) to complete. Using this estimation, the multipole method is about 13000 times faster than the brute-force method.