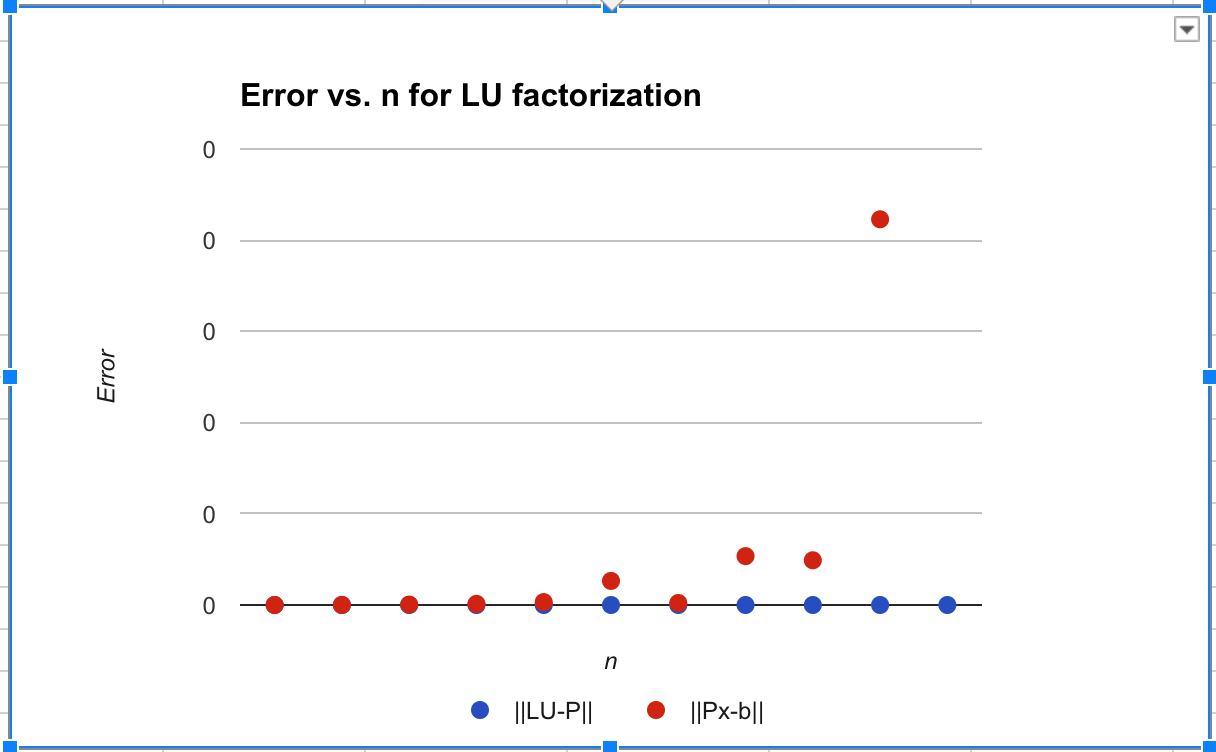
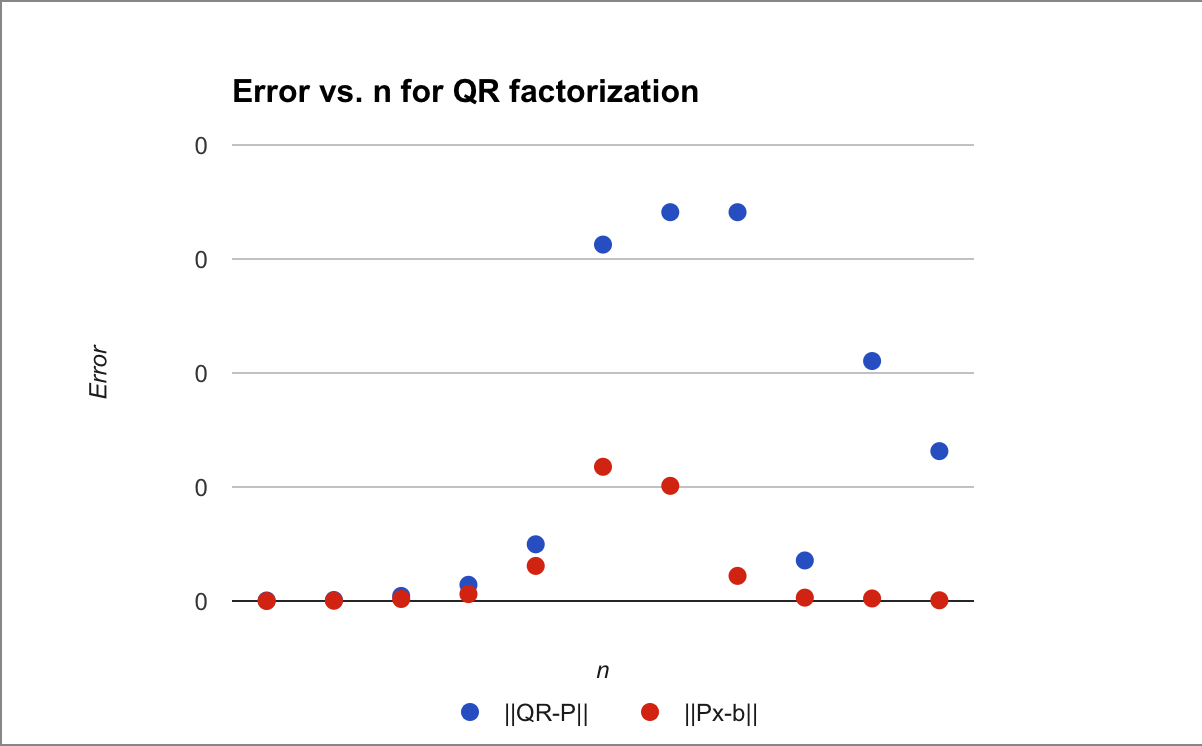
**Part A:**



(The error of ||LU-P|| and ||Px-b|| is so close that both lines overlap with each other.)



(The error of ||QR-P|| and ||Px-b|| is so close that both lines overlap with each other.)

(i) Why is it justified to use the LU or QR−factorizations as opposed of calculating an inverse matrix?

Using LU or QR factorizations is better than calculating the inverse of matrix because matrix inversion is numerically unstable. Solving the system using matrix inversion, the error in x is bigger than solving the system using LU or QR factorization. This makes it much more conducive when it is scaled to higher iterations, as it will retain more of its accuracy for a small price in runtime. In addition, it is computationally more complex to calculate the inverses of matrices with larger dimensions, as opposed to LU and QR, which has a more linear increase in runtime when a larger dimension is added.

(ii) What is the benefit of using LU or QR−factorizations in this way? (Your answer should consider the benefit in terms of conditioning error.)

Using QR factorization is better than using LU factorization. The reason is because when solving the system, QR factorization splits the system Px = b into two parts: Qy = b and Rx = y, where Q is an orthogonal matrix and R is an upper triangular matrix. Since the condition number for any orthogonal matrix is 1, the first step Qy = b has no error amplification at all. Then cond(QR) would be simply equal to cond(R), which minimizes the error of solving the entire system. This can be shown in the graphs, as the error for an LU factorization is always lower than the error for P.

**Part B:**

In this chart we see the error and iterations needed for two iterative methods to converge upon an accurate x(n) for the solution of Ax(n)=b. It is immediately apparent that Gauss Seidel’s method is far faster than Jacobi’s method. Jacobi’s method, at some points reaches ridiculous inaccuracy and takes a longer time to converge to the desired answer. There was one point for Gauss-Seidel that was an obvious outlier, but even without it the overall trend for convergence is much slower. This is because the spectral radius of Gauss-Seidel is smaller than the spectral radius for Jacobi.

**Part C:**

According to the data and graphs, there seems to be a general trend for the determinants of a 2x2 matrix with values between -2 and 2 to be around 0, and -6 and 6 for the inverse of the matrix. Generally, the fewer the iterations, the closer the determinant was to zero. However, the lower the number of iterations, the more varied the trace. In fact, the biggest outliers on both of these graphs occurred when the eigenvalue was calculated with three iterations. This is probably due to the accuracy generated by multiple iterations, or the lack thereof when few are done. About 40% of the data had iterations of 100, which was the maximum allowed. Overall, there were very few deviations to the determinant, as most did resolve to 0.

The inverse graph shows an interesting trend, since the points seem to be perfectly spread out on just the axes of the graph. Unlike the graph of A, the graph of A-1 has no noticeable points in any of the quadrants. Generally, the higher the number of iterations, the closer the trace was to 0. In addition, the lower the number of iterations, the closer the determinant was to 0. Again, around 40% of the 1000 data points had iterations of 100. This is opposite of what occurred in the graph of A, which makes sense because the inverse is being calculated. This is also due to the fact that the trace of A-1 is equal to the trace of A, divided by the determinant of A. This is because the trace is the addition of the diagonal of the matrix, and when calculating the inverse of a 2x2 matrix, you divide the matrix by the determinant and flip the top left and bottom right values. This will naturally modify the trace to be just divided by the determinant, since the values themselves only move positions along the diagonal.