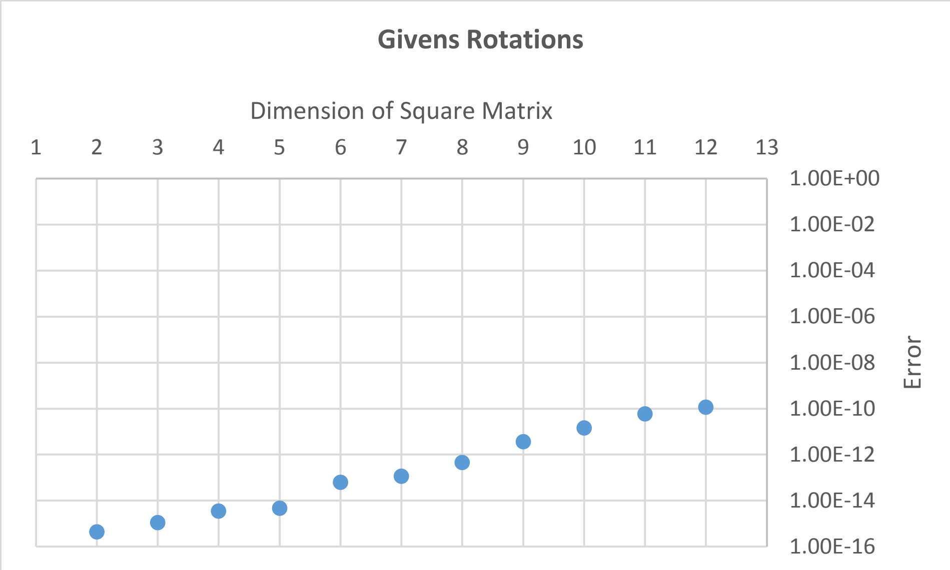
Part 1. (e & f)

The use of LU or QR-factorization will introduce less error into the solution of a linear system of equations than the use of an inverse matrix.

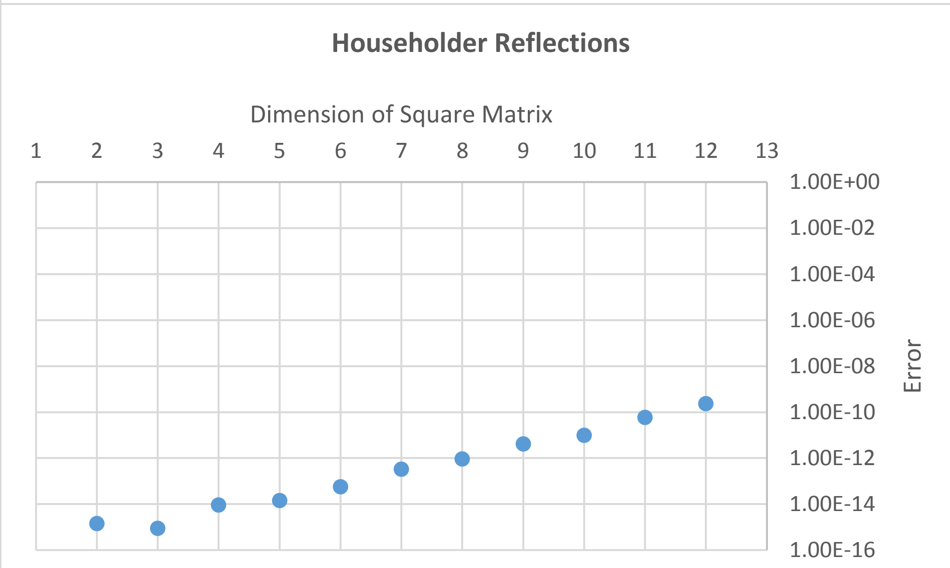
For QR-factorization, the norm of Q equals the norm of Q transpose which equals 1. Therefore, the condition number of A is equal to the condition number of R, meaning that no error is introduced that wasn't already inherent to the system. Specifically, the use of Householder Reflections and Givens Rotations as orthogonal transformations Q_i are stable methods for solving the system because the condition number of orthogonal transformation Q_i is 1.

For LU-factorization, partial Gaussian Elimination is used to find the matrix L. From the row operations used in this calculation, a matrix U can then also be constructed. From here, the equations Ly=b and Ux=y can be solved to calculate a final solution vector x by using forward and backward substitution methods. For this reason, the LU-factorization method uses less floating point operations than the full Gaussian Elimination needed for finding an inverse matrix or solving the system straight up. Using less floating point operations results in lower error in the arrived solution because of the computational limits of floating point processing.

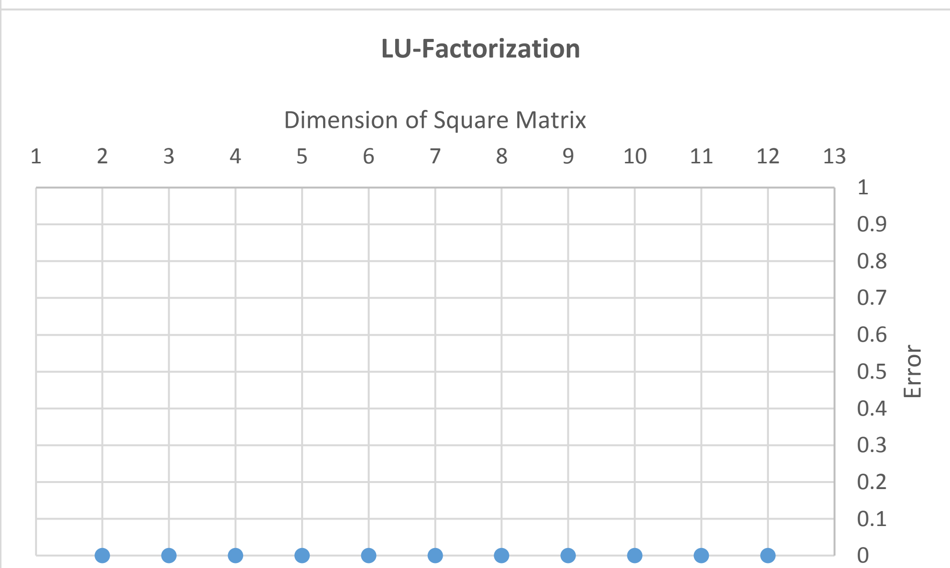
Our implementation of LU-factorization resulted in an identical Pascal matrix P, when L and U were multiplied together, for all n = 2 to 12. For this reason, the result of ||LU-P|| was always a zero matrix, and the norm of which was always 0. (Graphs are on page 2)



Givens Rotations
Dimension of Square Matrix
1
2
3
4
5
6
7
8
9
10
11
12
13
1.00E+00
1.00E-02
1.00E-04
1.00E-06
1.00E-08
1.00E-10
1.00E-12
1.00E-14
1.00E-16
Error



Householder Reflections
Dimension of Square Matrix
1
2
3
4
5
6
7
8
9
10
11
12
13
1.00E+00
1.00E-02
1.00E-04
1.00E-06
1.00E-08
1.00E-10
1.00E-12
1.00E-14
1.00E-16
Error



LU-Factorization
Dimension of Square Matrix
1
2
3
4
5
6
7
8
9
10
11
12
13
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Error