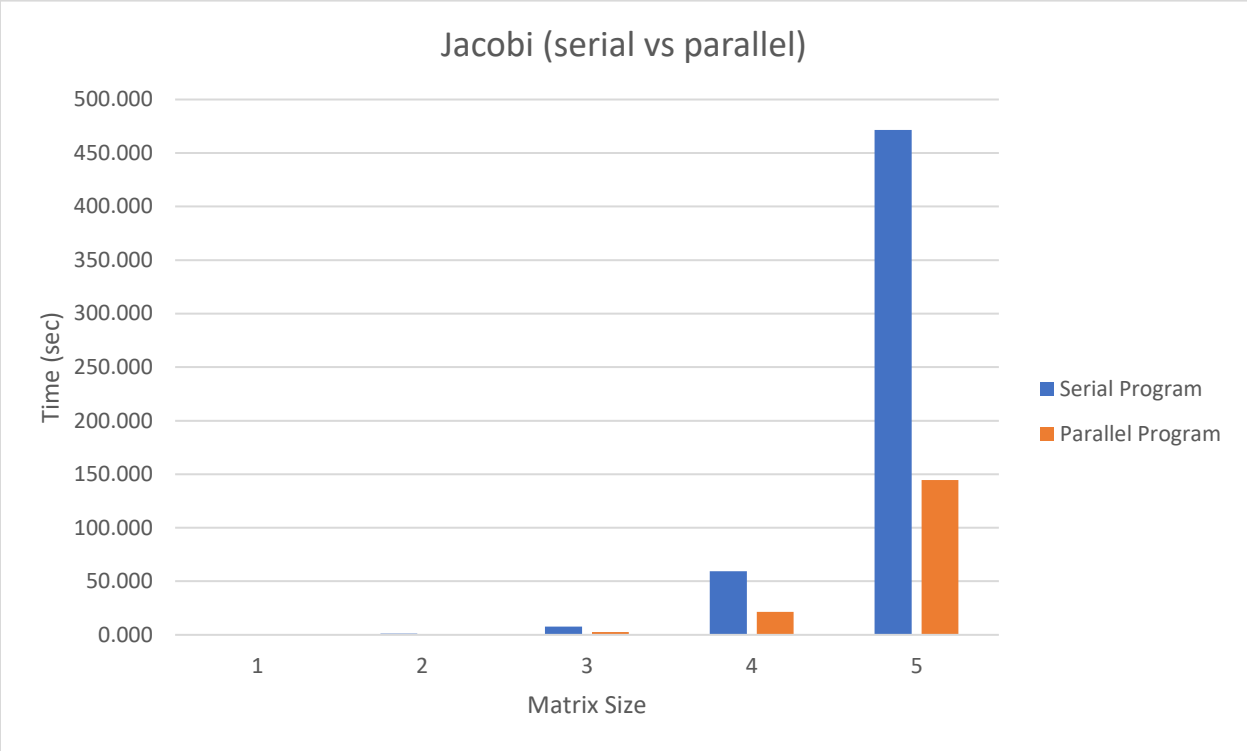
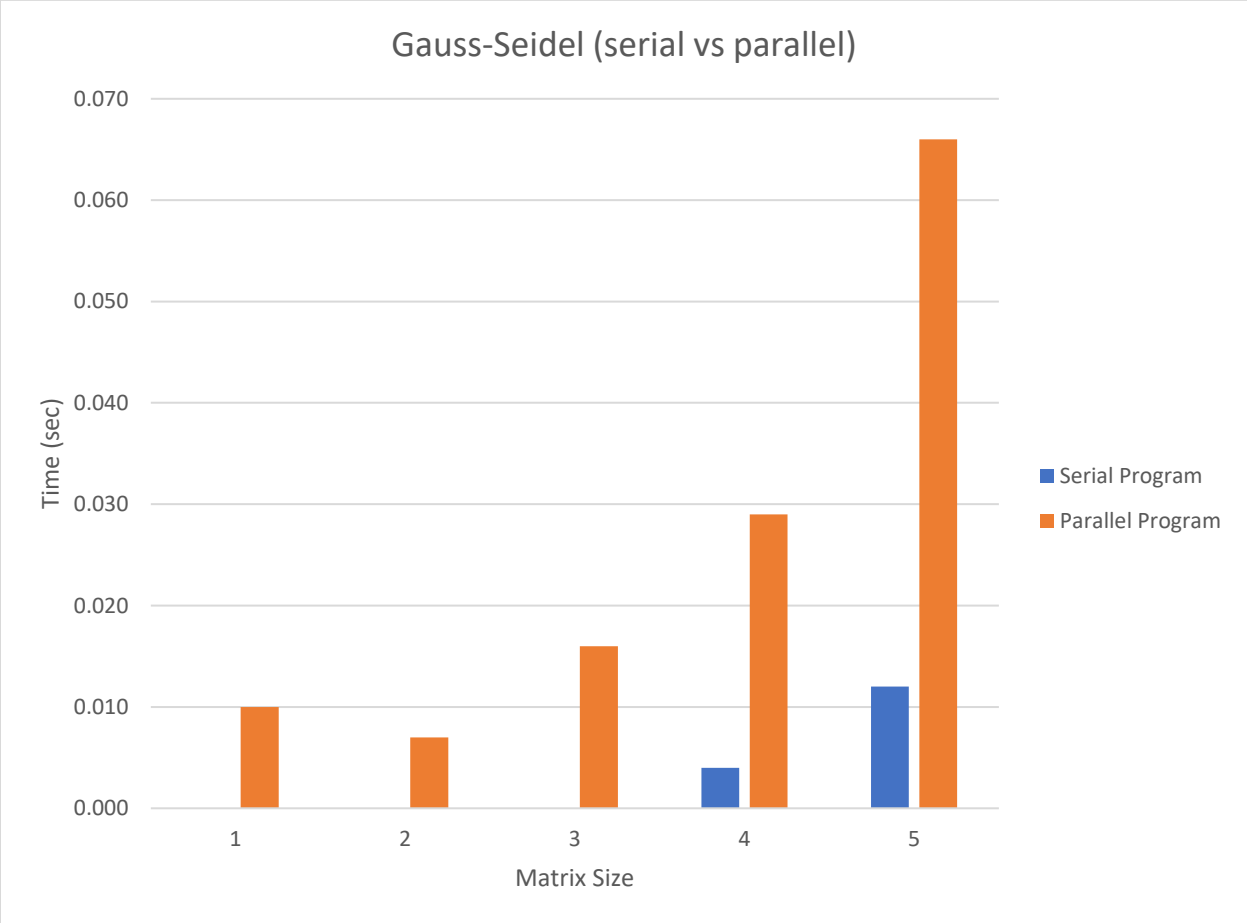
Figure 1 shows a graph and associated data table for comparing the time difference between a serial programed version of Jacobi iteration and a parallel programed version of Jacobi iteration. Along the x axis are the vales one through five that line up with the different size of matrices in the data table. The trend appears to be a considerably smaller amount of time to compute larger matrices using parallel over serial programing.



Jacobi				
Chart Item Number	**matrix size**	**time serial**	**time parallel**	**iterations**
1	40x40	0.141	0.126	18672
2	80x80	1.004	0.511	38953
3	160x160	7.593	2.571	77359
4	320x320	59.520	21.440	157509
5	640x640	471.500	144.700	314719

Figure 1: Time comparison for Jacobi iteration programed in serial vs parallel.

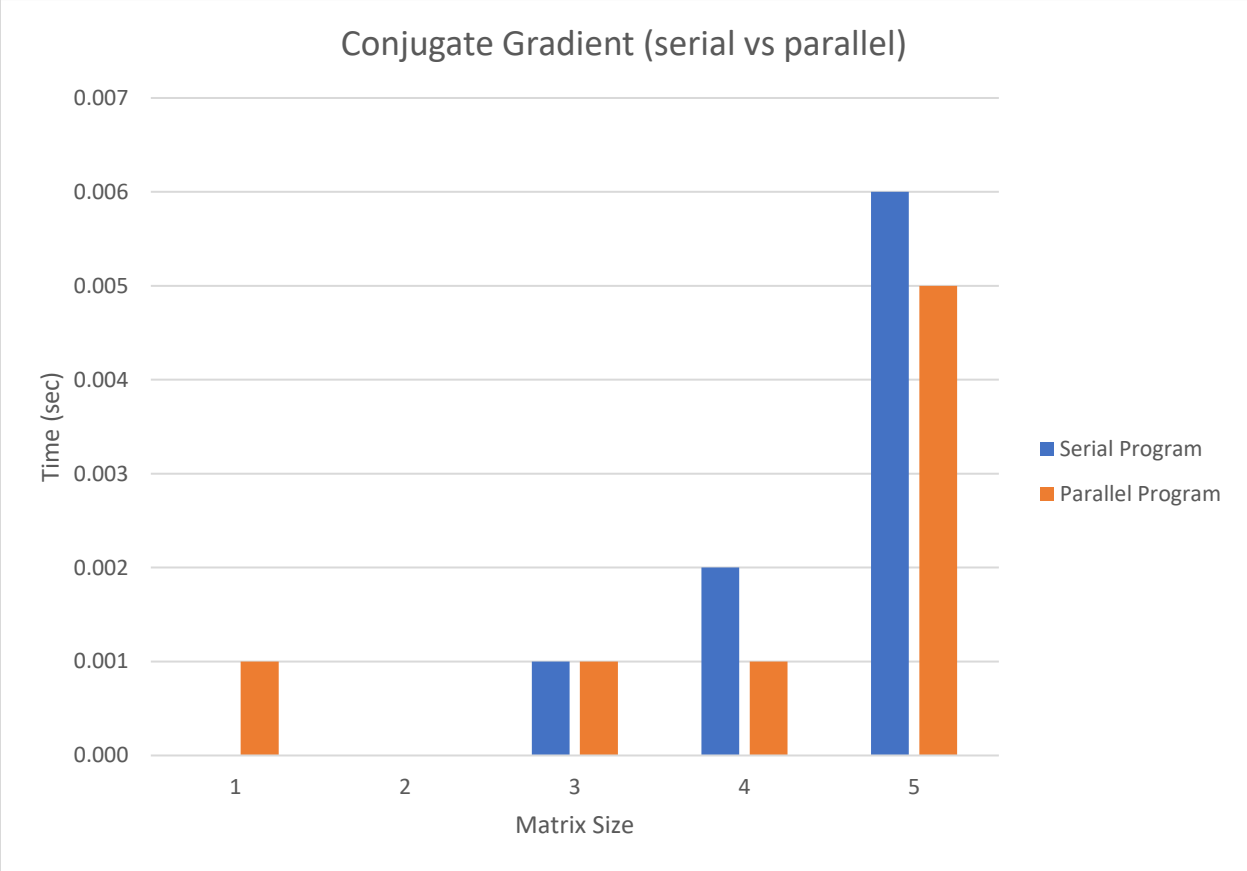
Figure 2 shows a graph and associated data table for comparing the time difference between a serial programed version of Gauss-Seidel iteration and a parallel programed version of Gauss-Seidel iteration. Along the x axis are the vales one through five that line up with the different size of matrices in the data table. The trend appears to be a considerably larger amount of time to compute all size matrices using parallel over serial programing. Due to the nature of Gauss-Seidel iteration I was only able to parallelize small portions of each iteration. The result is a high cost in time for overhead setup to run parallel threads even in the larger matrices.



Gauss-Seidel				
Chart Item Number	**matrix size**	**time serial**	**time parallel**	**iterations**
1	40x40	0.000	0.010	8
2	80x80	0.000	0.007	8
3	160x160	0.000	0.016	8
4	320x320	0.004	0.029	8
5	640x640	0.012	0.066	8

Figure 2: Time comparison for Gauss-Seidel iteration programed in serial vs parallel.

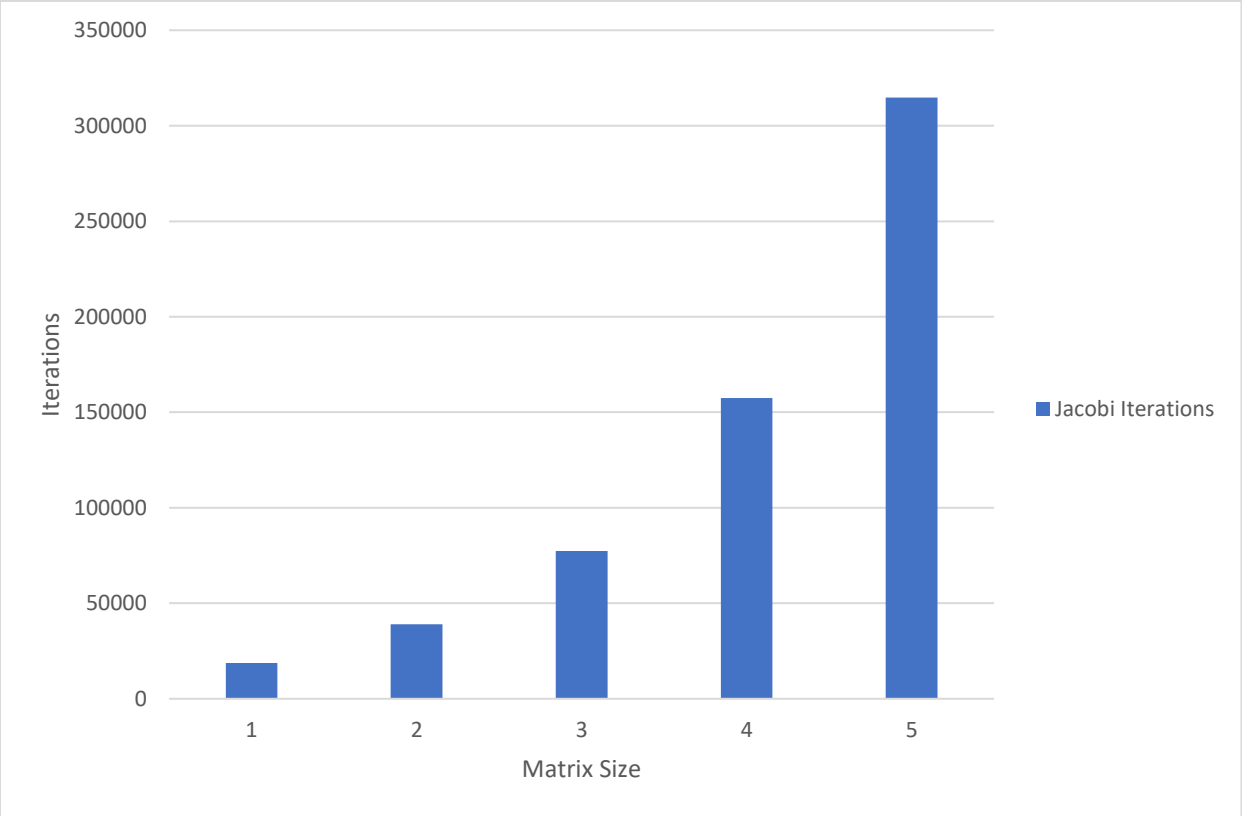
Figure 3 shows a graph and associated data table for comparing the time difference between a serial programed version of Conjugate Gradient iteration and a parallel programed version of Conjugate Gradient iteration. Along the x axis are the vales one through five that line up with the different size of matrices in the data table. The trend appears to be a smaller amount of time to compute larger matrices using parallel over serial programing.



Conjugate Gradient				
Chart Item Number	**matrix size**	**time serial**	**time parallel**	**iterations**
1	40x40	0.000	0.001	4
2	80x80	0.000	0.000	4
3	160x160	0.001	0.001	4
4	320x320	0.002	0.001	3
5	640x640	0.006	0.005	3

Figure 3: Time comparison for Conjugate Gradient iteration programed in serial vs parallel.

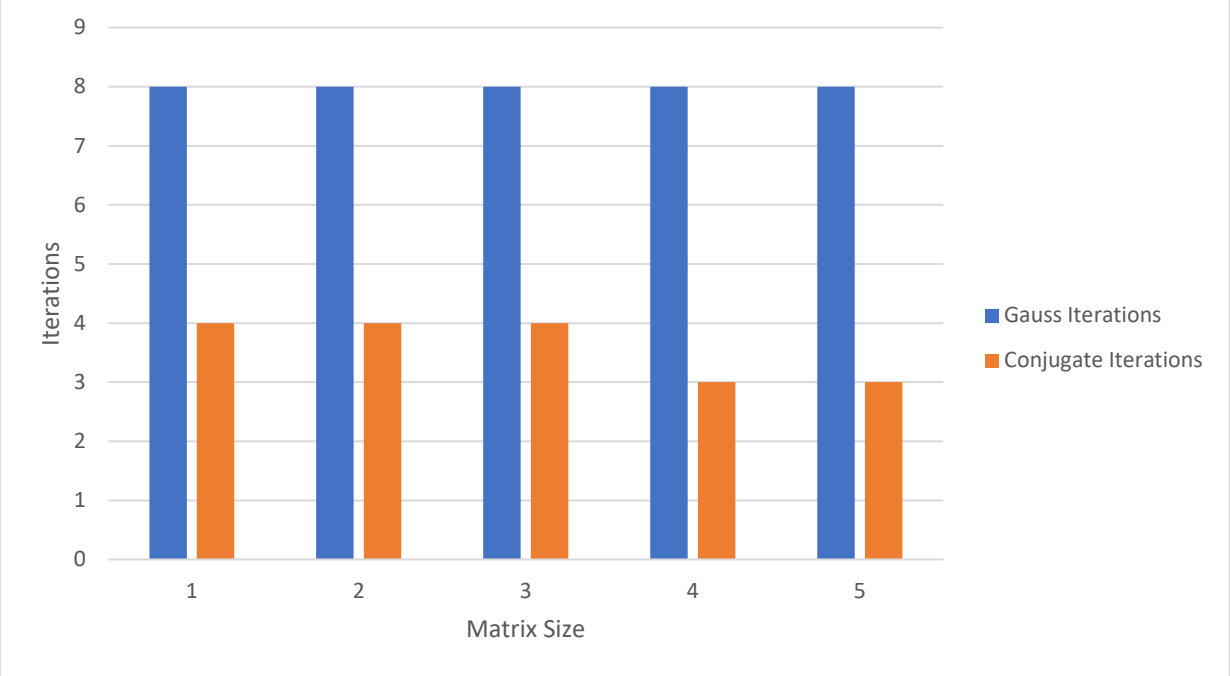
Figure 4 shows a graph and associated data table for comparing the number of iterations to the size of a matrix for the Jacobi iteration method. Along the x axis are the vales one through five that line up with the different size of matrices in the data table. The trend appears to be a considerably larger amount of iterations as the size of the matrix grows larger.



Jacobi				
Chart Item Number	**matrix size**	**time serial**	**time parallel**	**iterations**
1	40x40	0.141	0.126	18672
2	80x80	1.004	0.511	38953
3	160x160	7.593	2.571	77359
4	320x320	59.520	21.440	157509
5	640x640	471.500	144.700	314719

Figure 4: Jacobi iteration comparison for different size matrices.

Figure 5 shows a graph and associated data table for comparing the number of iterations to the size of a matrix for the Gauss-Seidel and Conjugate Gradient iteration methods. Along the x axis are the vales one through five that line up with the different size of matrices in the data table. The number of iterations for these two methods are considerably smaller than Jacobi. The trend for these two methods appears to be a relatively small change or no change in the number of iterations as the matrix size grows larger.

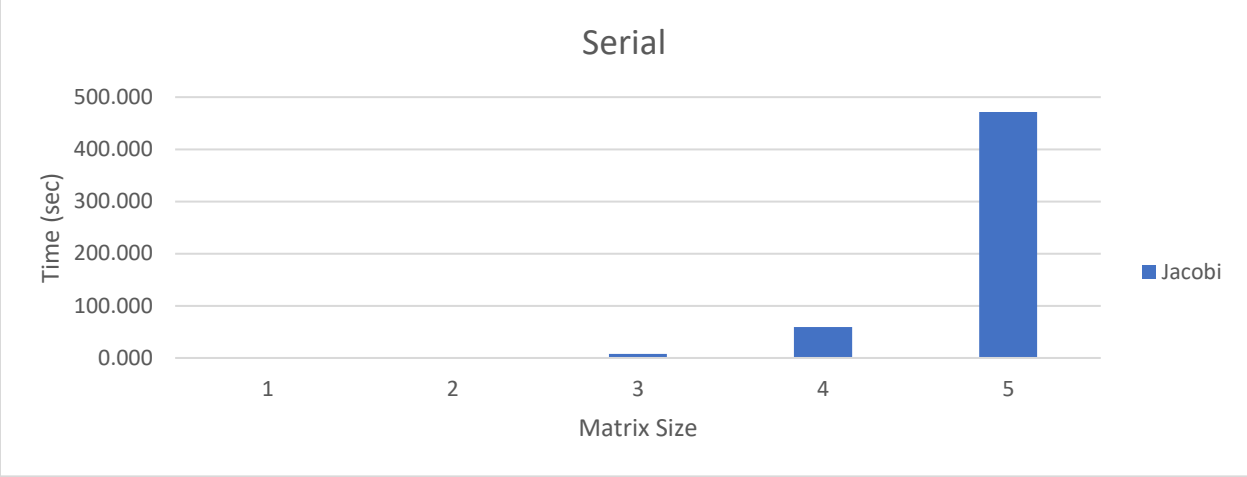


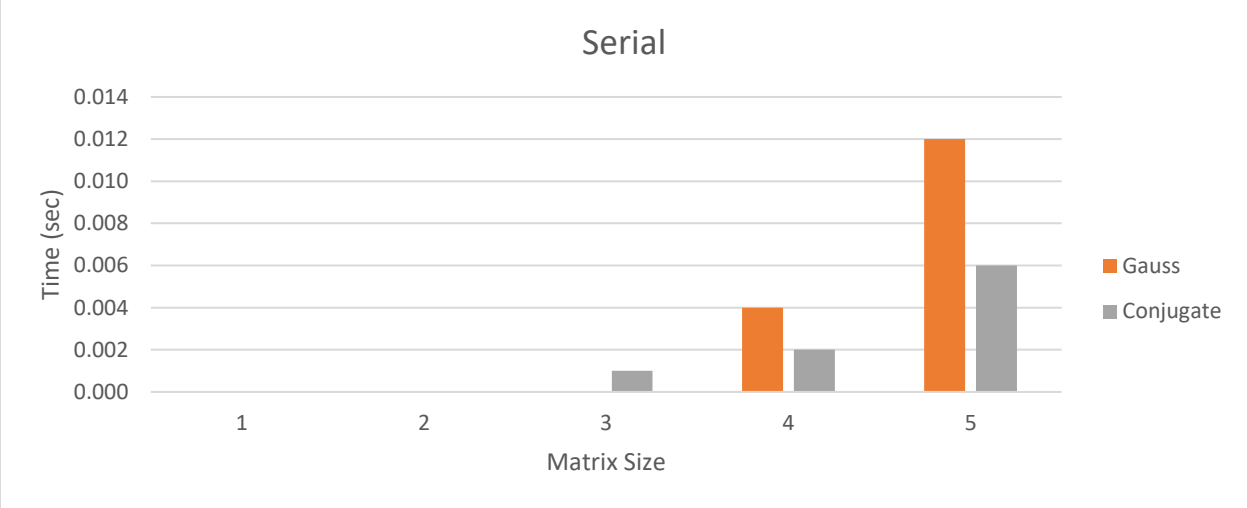
Gauss-Seidel				
Chart Item Number	**matrix size**	**time serial**	**time parallel**	**iterations**
1	40x40	0.000	0.010	8
2	80x80	0.000	0.007	8
3	160x160	0.000	0.016	8
4	320x320	0.004	0.029	8
5	640x640	0.012	0.066	8

Conjugate Gradient				
Chart Item Number	**matrix size**	**time serial**	**time parallel**	**iterations**
1	40x40	0.000	0.001	4
2	80x80	0.000	0.000	4
3	160x160	0.001	0.001	4
4	320x320	0.002	0.001	3
5	640x640	0.006	0.005	3

Figure 5: Gauss-Seidel and Conjugate Gradient iteration comparison for different size matrices.

Figure 6 shows a graph and associated data table for comparing the time difference vs matrix size. In this case, execution of the Jacobi, Gauss-Seidel, and Conjugate Gradient iteration methods were done using a serial programed version. The trend shows Jacobi requires more time in all cases of matrix size. Gauss-Seidel and Conjugate Gradient showed now difference until the matrix size grew to 160x160. As the matrices grow larger, Gauss-Seidel becomes the second slowest method and Conjugate Gradient becomes the fastest of the three methods.

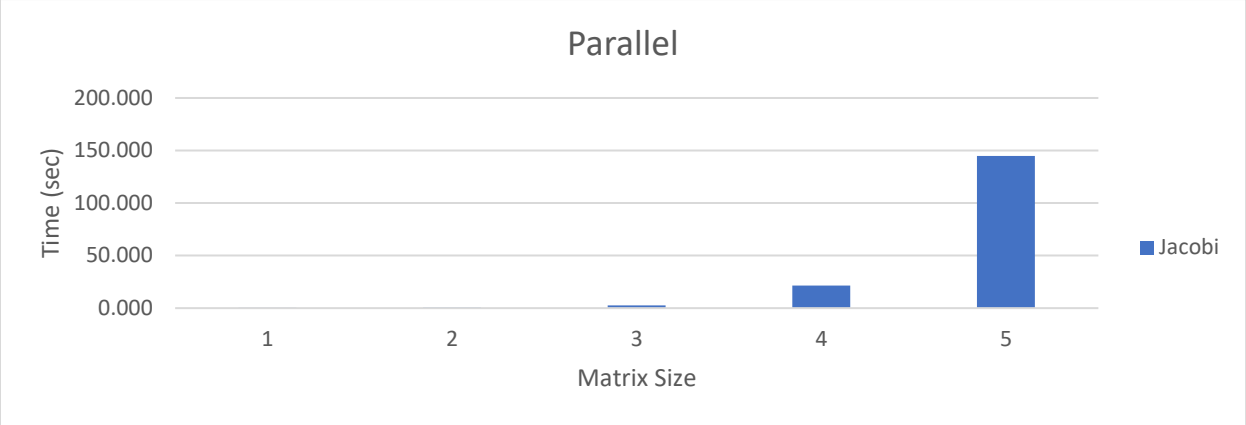


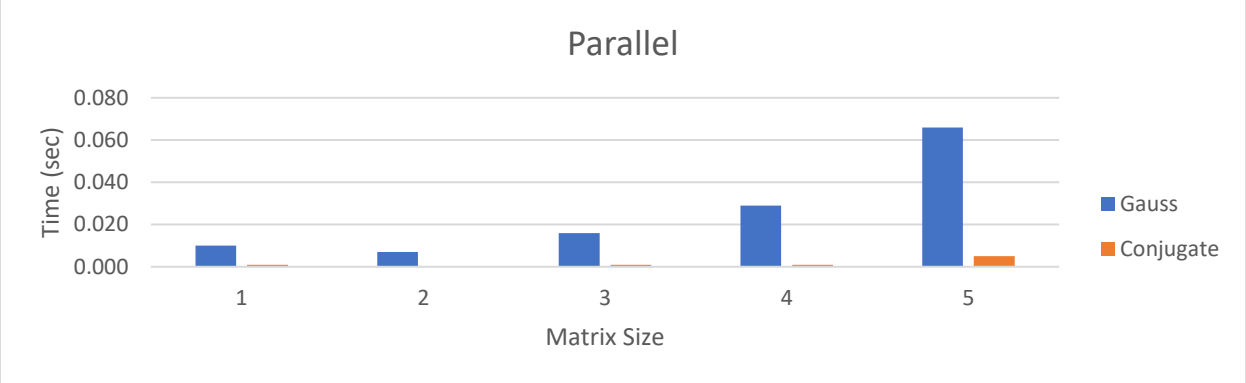


Serial (time in seconds)				
Chart Item Number	**matrix size**	**Jacobi**	**Gauss-Seidel**	**Conjugate Gradient**
1	40x40	0.141	0.000	0.000
2	80x80	1.004	0.000	0.000
3	160x160	7.593	0.000	0.001
4	320x320	59.520	0.004	0.002
5	640x640	471.500	0.012	0.006

Figure 6: Serial program time comparison for the three different methods.

Figure 7 shows a graph and associated data table for comparing the time difference vs matrix size. In this case, execution of the Jacobi, Gauss-Seidel, and Conjugate Gradient iteration methods were done using a parallel programed version. The trend shows Jacobi requires the most time as the matrix size grows. Gauss-Seidel is second in line and Conjugate Gradient is the faster method of the three.





Parallel (time in seconds)				
Chart Item Number	**matrix size**	**Jacobi**	**Gauss-Seidel**	**Conjugate Gradient**
1	40x40	0.126	0.010	0.001
2	80x80	0.511	0.007	0.000
3	160x160	2.571	0.016	0.001
4	320x320	21.440	0.029	0.001
5	640x640	144.700	0.066	0.005

Figure 7: Parallel program time comparison for the three different methods.