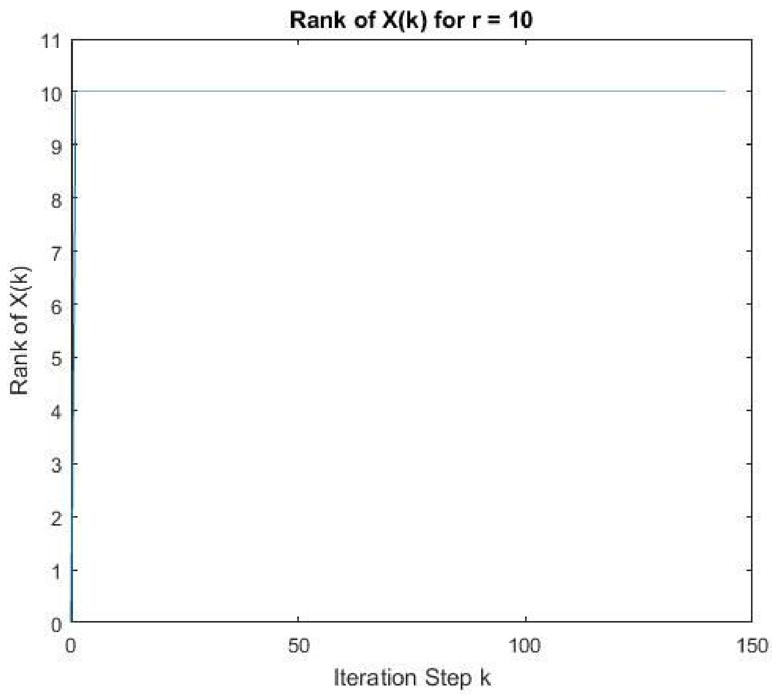
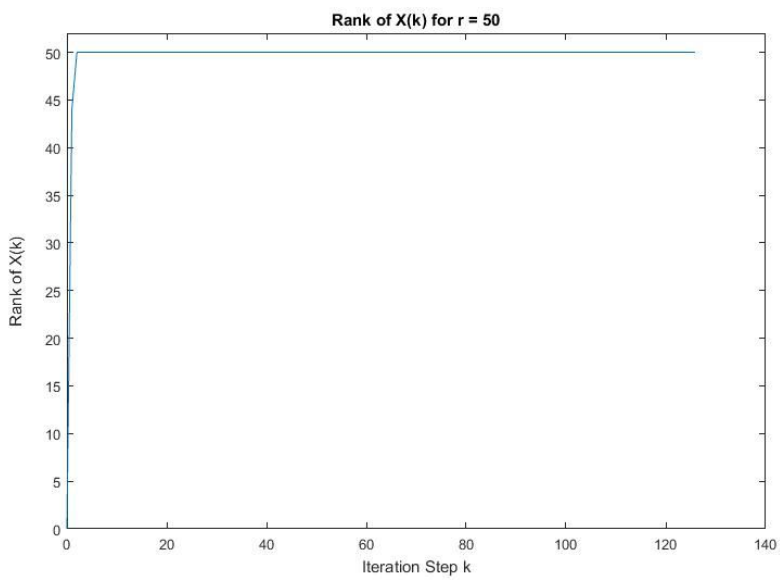
Bri Miskovitz
5/13/17
Numerical Linear Project

Unknown M				Computational Results		
Size (nxn)	Rank (r)	m/d_r	m/n^2	Time (s)	# Iters	Relative Error
1000x1000	10	6	0.12	23.0178	115	1.6203e-04
	50	4	0.39	89.1026	112	1.6471e-04
	100	3	0.57	266.0153	128	1.6926e-04
5000x5000	10	6	0.024	118.1498	124	1.7143e-04
	50	5	0.10	705.2660	108	1.5658e-04
	100	4	0.158	2904	122	1.7519e-04
10000x10000	10	6	0.012	425.1485	124	1.7290e-04
	50	5	0.050	2325	110	9.5188e-05
	100	4	0.080	6228	103	1.6005e-04
20000x20000	10	6	0.006	Out of Memory	Out of Memory	Out of Memory
	50	5	0.025	Out of Memory	Out of Memory	Out of Memory
30000x30000	10	6	0.004	Out of Memory	Out of Memory	Out of Memory

Table 5.1 The table represents the average data for various M matrices varying in size, rank, step size, and the number of sampled entries and important data for its convergence for five runs in the SVT algorithm. Matrices with sizes 20000 and 30000 ran out of memory for all ranks. The larger matrices with larger ranks took the longest to converge with the matrix with size 10000 and rank 100 taking almost 2 hours. The matrices took about the same time to converge, and the relative error is only to about 1e01 at most.


Rank of X(k) for r = 10
Rank of X(k)
Iteration Step k
0
1
2
3
4
5
6
7
8
9
10
11
0
50
100
150


Rank of X(k) for r = 50
Rank of X(k)
Iteration Step k
0
5
10
15
20
25
30
35
40
45
50
0
20
40
60
80
100
120
140

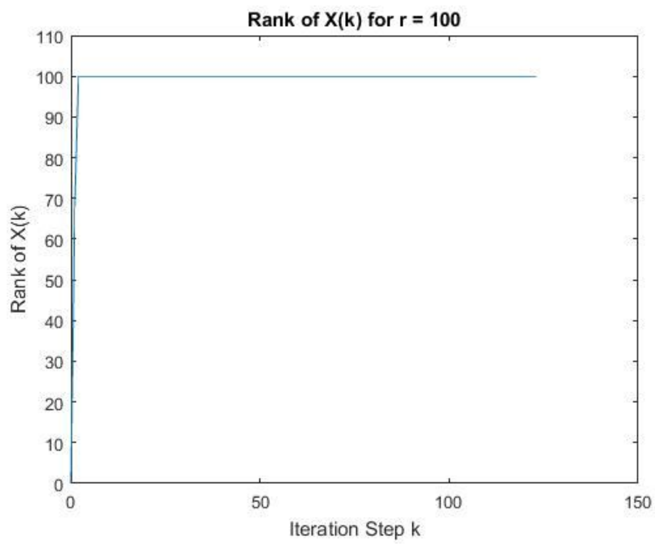


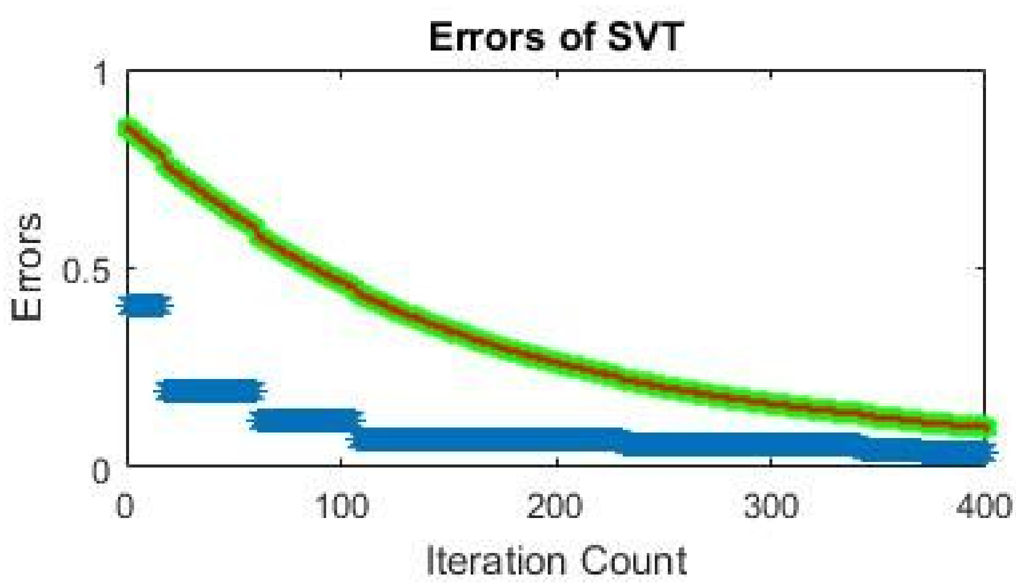
Figure 5.1 : The graphs above represent the rank of the X matrix that is reconstructed over k iterations steps. The initial matrix M has a size of 5000, and a rank of 10, 50, and 100, for M, are observed. The graphs match the graphs in the paper, which shows the algorithm is working correctly. The rank of X increases and converse rapidly very fast and in similar times.

	$\delta = 0.8p^{-1}$			$\delta = 1.2p^{-1}$			$\delta = 1.6p^{-1}$		
tau	# of Iters		Rank	# of Iters		Rank	# of Iters		Rank
	mean	std	mean	mean	std	mean	mean	std	mean
2n	352	142	16.4	165	81.3	13.9	DNC	0	DNC
3n	121	1.5	10.0	76	1.7	10.0	410	450	10.0
4n	149	3.3	10.0	97	3.1	10.0	100	2.3	10.0
5n	181	3.8	10.0	119	2.2	10.0	92	1.3	10.0
6n	201	5.7	10.0	138	2.5	10.0	110	1.97	10.0

Table 5.2 This table shows important average data for the convergence in the SVT algorithm for five runs various step sizes and threshold parameters. Larger step size seems to cause the algorithm to have an issue with converging; it takes too much work and too many steps. Increasing the threshold doesn't seem to have an effect until the step size is too big.

Algorithm	rank	k_i	time	$\frac{\|M - M\_i\|_F}{\|M\|_F}$	$\frac{\|M - X^{k_i}\|_F}{\|M\|_F}$
SVT	1	1	5.692	0.189455	0.770118
	2	62	18.36	0.115872	0.583208
	3	108	32.56	0.070615	0.439332

Table 5.5   This table shows the relative error and the best error for a M 312x312 matrix that contains distances between cities. For this M to converge, it will take a long time because M is a full rank, so the number of iterations, time, and error to reach rank 1, 2, and 3 is observed. The rate between the ranks seems relatively linear, but the errors are very high compared to other the earlier tables.



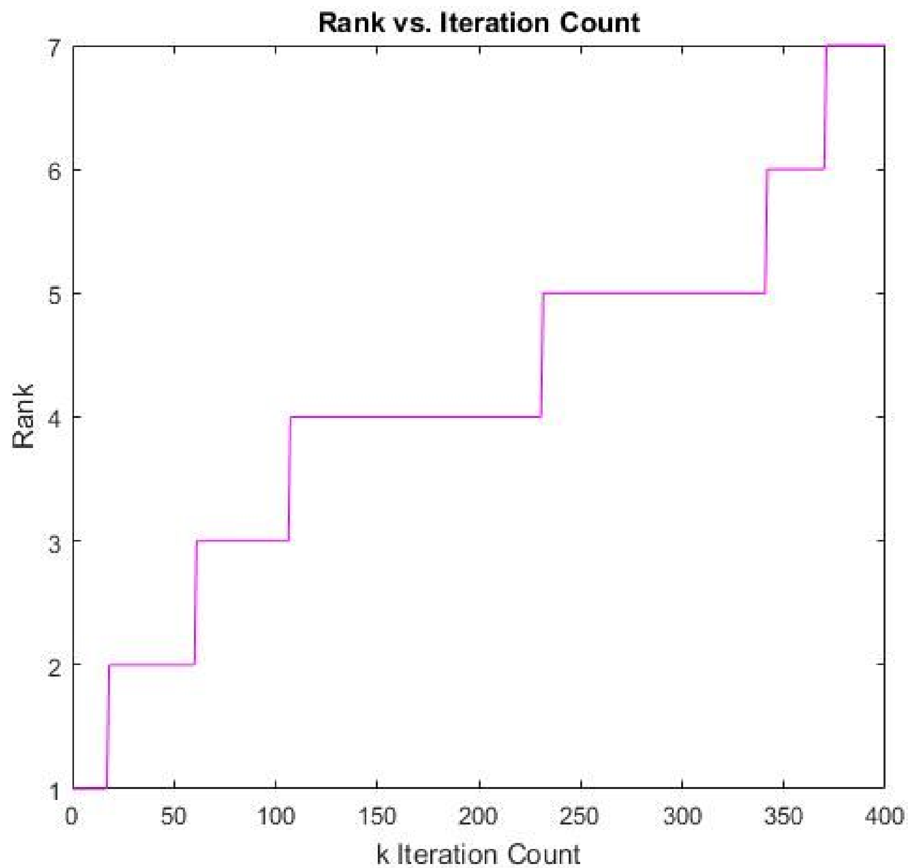


Figure 5.2   The graphs represent relations between the iteration count and important data that helps describe the converge flow produced from the SVT algorithm. The first graph compares three types of error: relative error, residual relative error, and the error from shrinking the SVD of M based on the rank of M. The errors follow roughly relation, but the differences are pretty significant. The blue line represents the relative error, the red line represents the residual relative error, and the black represents the best error. The second graph follows the same pattern in the pattern, and as iterations increase, a step graph is produced to represent the increasing rank of X. From rank = i to rank = i + 1, it will take an increasing amount of iterations.