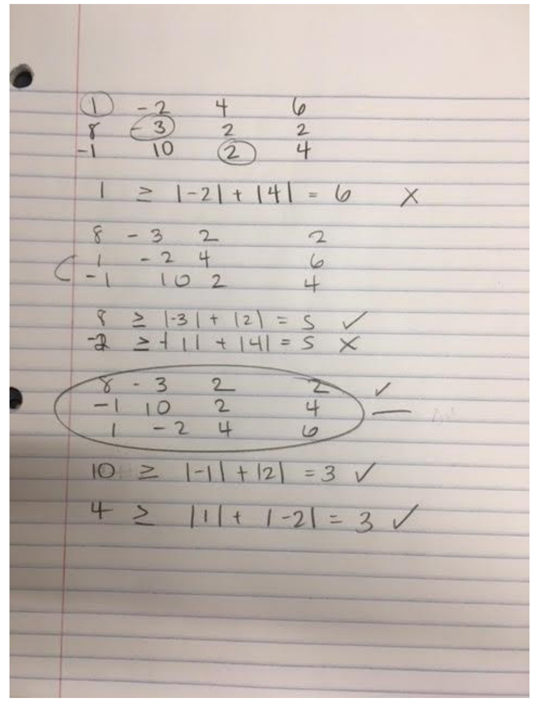
Gauss/Jacobi Iterative Method
Numerical Computation
Lilybeth Delgado
Project #7



The following image shows the work done to determine the diagonal dominant matrix. This could have been done on MATLAB but it was visually and algebraically easier by hand. Although in the future for an arbitrary matrix of length x it would be much more difficult to perform all permutations to find the dominant diagonal. The Gaussian and Jacobi follow similar approaches in the iterations but differ in that Jacobi takes the previous approximations while Gauss-Seidal takes on ones currently available.

Let Matrix A=

$$\text{a)} \begin{bmatrix} 3 & 1 & -4 & |7 \\ -2 & 3 & 1 & |-5 \\ 2 & 0 & 5 & |10 \end{bmatrix}$$

Let Matrix B=

$$\text{b)} \begin{bmatrix} 1 & -2 & 4 & |6 \\ 8 & -3 & 2 & |2 \\ -1 & 10 & 2 & |4 \end{bmatrix}$$

Results:

	Matrix A	Iterations	Stopping Criteria	Matrix B	Iterations
Gauss	3.2094 0.2343 0.7163	63	MAE	-0.1131 0.0756 1.5661	15
Gauss	3.2094 0.2343 0.7163	63	RMSE	-0.1132 0.0755 1.5660	18
Jacobi	3.2087 0.2351 0.7159	169	MAE	-0.1132 0.0755 1.5659	21
Jacobi	3.2094 0.2339 0.7165	171	RMSE	-0.1132 0.0755 1.5659	21

The following is table demonstrating all the roots and the iterations followed by which stopping criteria. Gaussian Iteration method does converge faster in both cases. And for some the roots differ be 0.001 or 0.0002 which questions the accuracy of our results, although very close to each other. On the bottom or some screen shots that display the iteration number to confirm my results on the table. The rest can be seen through the “testingAssign2.m” file.

18	-0.112891	0.075430	1.566016	0.012266
19	-0.113218	0.400000	1.500000	0.000327
20	-0.113218	0.075508	1.500000	0.000405
21	-0.113218	0.075508	1.565937	0.000483
	-0.1132	0.0755	1.5659	

1	-0.375000	0.400000	1.500000	70.140625
2	-0.375000	1.000000	1.500000	74.140625
3	-0.375000	1.000000	-1.000000	78.140625
4	0.875000	0.400000	1.500000	1.562500
5	0.875000	0.562500	1.500000	1.753906
6	0.875000	0.562500	2.093750	11.325195
7	-0.062500	0.400000	1.500000	0.878906
8	-0.062500	0.068750	1.500000	1.122695
9	-0.062500	0.068750	1.562500	1.404922
10	-0.114844	0.400000	1.500000	0.002740
11	-0.114844	0.081250	1.500000	0.002896
12	-0.114844	0.081250	1.550000	0.003052
13	-0.107031	0.400000	1.500000	0.000061
14	-0.107031	0.078516	1.500000	0.000069
15	-0.107031	0.078516	1.569336	0.000442
16	-0.112891	0.400000	1.500000	0.000034
17	-0.112891	0.075430	1.500000	0.000044
18	-0.112891	0.075430	1.566016	0.000055
19	-0.113218	0.400000	1.500000	0.000000
20	-0.113218	0.075508	1.500000	0.000000
21	-0.113218	0.075508	1.565937	0.000000
	-0.1132	0.0755	1.5659	

47	3.213178	0.236132	0.717960	0.000080
48	3.213178	0.236132	0.714729	0.000090
49	3.207595	0.236132	0.714729	0.000031
50	3.207595	0.233487	0.714729	0.000038
51	3.207595	0.233487	0.716962	0.000043
52	3.211454	0.233487	0.716962	0.000015
53	3.211454	0.235315	0.716962	0.000018
54	3.211454	0.235315	0.715418	0.000021
55	3.208786	0.235315	0.715418	0.000007
56	3.208786	0.234051	0.715418	0.000009
57	3.208786	0.234051	0.716486	0.000010
58	3.210630	0.234051	0.716486	0.000003
59	3.210630	0.234925	0.716486	0.000004
60	3.210630	0.234925	0.715748	0.000005
61	3.209355	0.234925	0.715748	0.000002
62	3.209355	0.234321	0.715748	0.000002
63	3.209355	0.234321	0.716258	0.000002
	3.2094	0.2343	0.7163	