EE4070 Numerical Analysis

Homework 6. Conjugate Gradient Method

110071021 吳睿芸

1. Introduction

This is a report about Conjugate Gradient Method. In this report, we are going to use Conjugate Gradient Method to do linear solution to limit error in a certain range given. We define iteration error as infinite error. Implement conjugate gradient method to solve the resistor network problem in HW04 with 50 resistors per side, and find the tolerance such that the solution accuracy of 10^{-7} volts, as compared to the solution found using a direct method, is reached.

Then we use the same tolerance to solve the resistor network problem with N_R resistors per side, for N_R = 2, 4, 10, 20, 40, 50, 60, 80, 100. And for each N_R we find Vw, Vne, Ve and Req and compare these results from different size of resistor networks.

At last, compare the CPU time and the convergence behavior we got by using conjugate gradient method to the result we got by implement LU decomposition to the resistor network to see if there is any different , and write down the observation and analyze it.

2. Approach - Conjugate Gradient Method

The method we use in this report is called stationary linear iterative methods of the first order that have the following form. Given a linear system Ax = b with a symmetric positive definite matrix A and an initial guess $x^{(0)}$, let $p^{(0)} = r^{(0)} = b - A x^{(0)}$, for k = 0, 1, . . ., we have the algorithm below:

$\alpha_k = (p^k)^T r^k / (p^k)^T A p^k$,

$x^{k+1} = x^k + \alpha_k p^k$,

$r^{k+1} = r^k - \alpha_k A p^k$,

$\beta_k = (p^k)^T A r^{k+1} / (p^k)^T A p^k$,

$p^{k+1} = r^{k+1} - \beta_k p^k$

Then we iterate the process until the iteration error, where we define it as infinite error, which means $error^{(k+1)} = |x^{(k+1)} - x^{(k)}|_\infty$ is smaller than the tolerance given.

3. Result and Observation

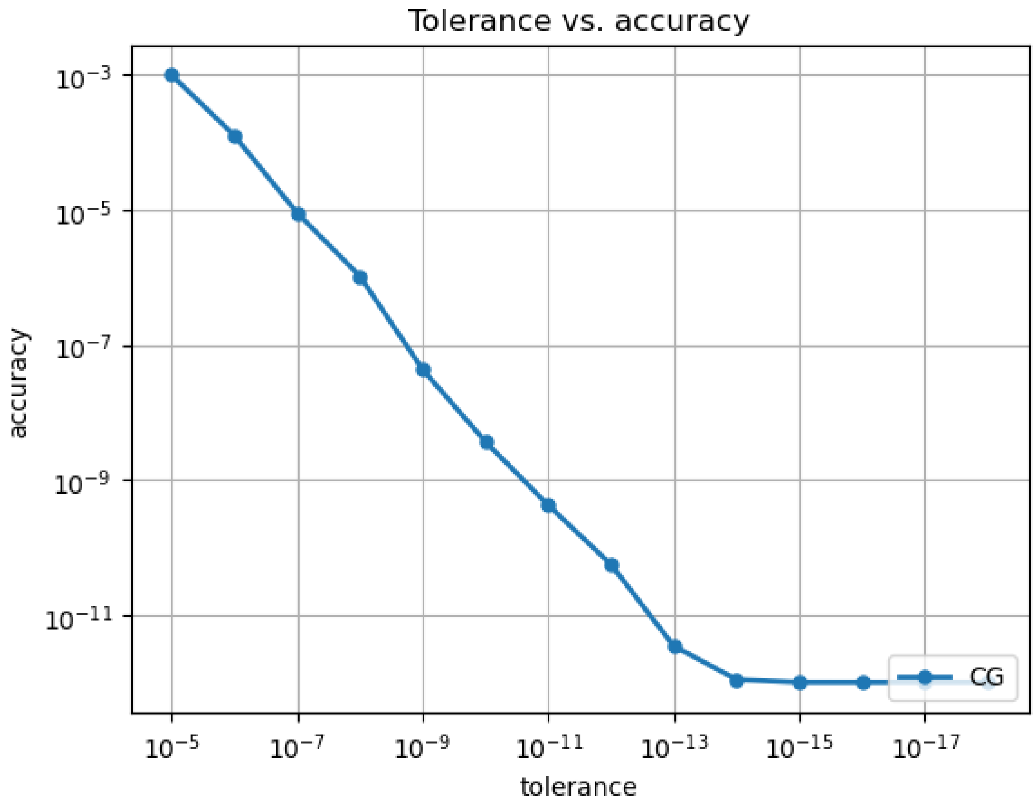
We record the timestamp before and after creating simple resistor network system matrix and doing LU decomposition and conjugate gradient method, and subtract these timestamps to get the CPU operating time for solving the resistor network problems.

We use the conjugate gradient method to solve the resistor network problem in HW04 with 50 resistors per side, and find the tolerance(tol) such that the solution accuracy of 10−7 volts, as compared to the solution found using a direct method, is reached.

tolerance	accuracy
1e-5	0.0009962315784981942
1e-6	0.00012695756748303438
1e-7	9.07340403442233e-06
1e-8	1.0497464786487477e-06
1e-9	4.3634942483766015e-08
1e-10	3.710211245819366e-09
1e-11	4.2671373127075724e-10
1e-12	5.642714704745278e-11
1e-13	3.4915374214093697e-12
1e-14	1.114336478104337e-12
1e-15	1.0086720675361793e-12
1e-16	1.0082713937453716e-12
1e-17	1.0082713937453716e-12

1e-18	1.0082713937453716e-12

Graph 1. Solution accuracy of using CG vs LU method with different tolerance



Graph 2. accuracy compared to with different tolerance

Therefore, we find the relevant accuracy with tolerance 1e-q, and q is an integer from -5 to -18. From the result we could find that when the tolerance getting smaller, the accuracy getting smaller too, and the accuracy converge to around 10^{-12} after the tolerance is smaller than 10^{-14}.

Also, we find the tolerance such that the solution accuracy of 10^{-7} volts, as compared to the solution found using a direct method, is around 10^{-9}.

Then, we use the same tolerance 10^{-9} to solve the resistor network problem with N_R resistors per side, for N_R = 2, 4, 10, 20, 40, 50, 60, 80, 100. And for each N_R we find Vw, Vne, Ve and Req and compare these nodes.

N	Vw (V)	Vne (V)	Ve (V)	Req (Ohms)
2	0.551724	0.482759	0.344828	1208.33
4	0.501826	0.44108	0.341498	850.53
10	0.452439	0.407838	0. 338617	491.2
20	0. 428654	0. 392824	0. 337465	307.39
40	0. 412276	0. 382576	0. 336732	185.65
50	0. 408082	0. 379957	0. 336548	156.85
60	0. 404961	0. 378008	0. 336412	136.41
80	0.400517	0.375235	0. 33622	109.08
100	0. 397422	0. 373304	0. 336086	91.48

Graph 3. Vw, Vne, Ve and Req for different size resistor networks

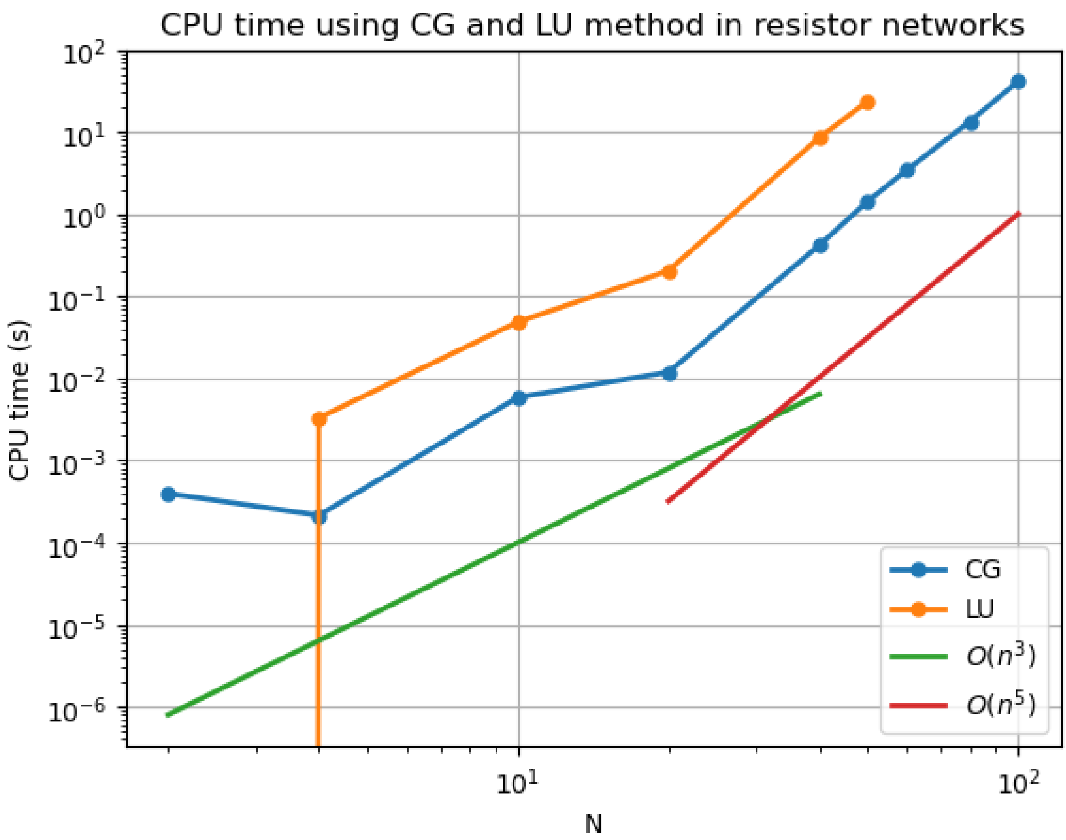
In this table, N is number of resistors per side, Vw, Vne, Ve are the node in different direction in resistor network. Req stands for the equivalent resistance. We could find out Req decrease when the number of resistors per side increase. That’s because the resistors are arrange in parallel. When adding more and more resistors in system for the same voltage, the equivalent resistance will decrease. Also, we could find out that the result of Vw, Vne and Ve in each resistor network are similar.

Then we try to compare the CPU time of CG method and LU decomposition.

N	dimension	CG CPU time (s)	LU CPU time (s)
2	9*9	0.000395775	0
4	25*25	0. 000214338	0.00330544
10	121*121	0. 00585794	0.0484321
20	441*441	0. 0118828	0.204211
40	1681*1681	0. 414074	8.47988
50	2601*2601	1. 42698	23.5516
60	3721*3721	3.45647	X
80	6561*6561	13.3685	X
100	10201*10201	41.2944	X

Graph 4. CPU time (s) for using CG and LU method in resistor networks

From graph 4, N is number of resistors per side, we use both methods to solve resistor network. The conjugate gradient method under the tolerance equals to 10^{-9} compare to LU decomposition. We could discover that the conjugate gradient method is extremely efficient compare to LU decomposition.



Graph 5. CPU time (s) for using CG and LU method in resistor networks and bigO

From graph 5, We know when N is below 20 (For N=20, the dimension of matrix is 441*441), CPU time of LU and CG method are both O(n^3). When N is larger than 20, CPU time became O(n^5). But CG method still run faster than LU decomposition for about 10^1 times faster. Therefore, CG method is still a better way to solve matrix, especially extreme large matrix.

4. Conclusion

Accuracy of conjugate gradient method converge to around 10^{-12} after the tolerance is smaller than 10^{-14}, and the tolerance such that the solution accuracy of 10^{-7} volts, as compared to the solution found using a direct method, is around 10^{-9}.

Then, use the same tolerance 10^{-9} to solve the resistor network problem with N_R resistors per side, for N_R = 2, 4, 10, 20, 40, 50, 60, 80, 100. And for each N_R we find Vw, Vne, Ve and Req. We could find out Req decrease when the number of resistors per side increase. Also, we could find out that the result of Vw, Vne and Ve in each resistor network are similar.

Then we try to compare the CPU time of CG method and LU decomposition. We could discover that the conjugate gradient method is extremely efficient compare to LU decomposition. When number of resistors per side is below 20 (dimension of matrix is 441*441), CPU time of LU and CG method are both $O(n^3)$. When N is larger than 20, CPU time became $O(n^5)$. But CG method still run faster than LU decomposition for about 10^1 times faster. Therefore, CG method is still a better way to solve matrix, especially extreme large one.