Iterative Method For Linear System - Report

NTU CSIE & MATH Upstair Looser

2014/11/24

1 Homework Description

In this homework, we are going to use four iterative methods to solve two kind of sparse linear system with different scales. Then compare the performance of each methods. Hoping to get deeper knowledge into the extraordinary iterative linear system method.

The four iterative methods are:

- Gauss-Seidel (GS)
- PCG with identity matrix as preconditioner
- PCG with incomplete Cholesky matrix as preconditioner (PCG-IC)
- Symmetric Gauss-Seidel (Sym-GS)

There are two type of input linear system in this homework. The first problem is that the matrix is regular and the other is refined at a hole. For each method, we tried three cases with type 1 and 2 cases with type 2. On the other hand, we focus on the following features in the result:

- Number of iterations
- Elapsed time
- Storage

I'm using Matlab to do the whole calculation in this report. You can find my code and the result .mat file on Google drive. The links are in the reference at the last page of this report[1].

2 Result

We compare three important features between the four methods we use. The features are: number of iterations, elapsed time, and storage. The result of the first category dataset is as follow:

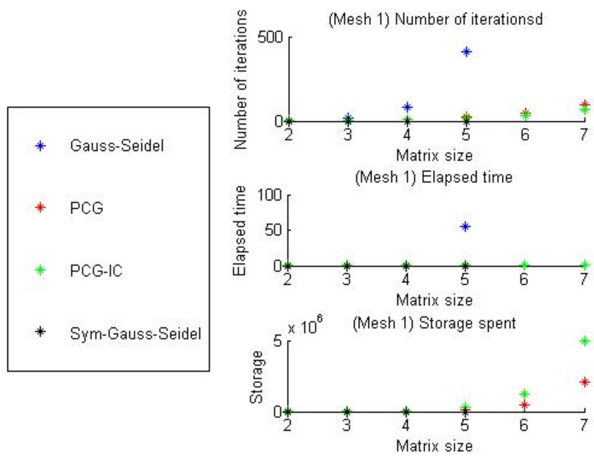

Figure 1: Result using mesh 1 dataset

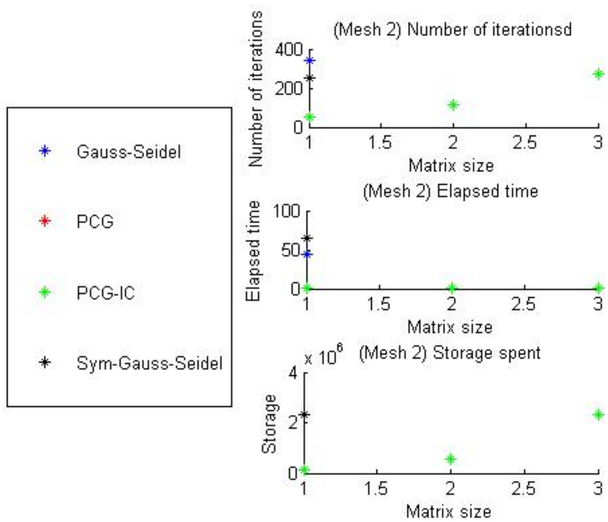

Figure 2: Result using mesh 2 dataset

Remark. *The matrices size of the first dataset(mesh1) are*

2	3	4	5	6	7
Size = 9	Size = 49	Size = 225	Size 961	Size = 3969	Size = 16129

Table 1: The matrices size of mesh1

The matrices size of the second dataset(mesh2) are

1	2	3
Size = 915	Size = 3652	Size = 14586

Table 2: The matrices size of mesh2

2.1 Number of iterations

First, let's take a look at the number of iterations using four different iterative linear system solving method:

Method	Size = 9	Size = 49	Size = 225	Size = 961	Size = 3969	Size = 16129
GS	3	17	85	411		
PCG	2	4	10	23	47	95
PCG-IC	1	2	5	12	26	57
Sym-GS	3	14	67	321		

Table 3: Number of iterations using four iterative methods

We can easily see that no matter using Gauss-Seidel or symmetric Gauss-Seidel method to solve iterative linear system, the number of iterations will quickly increases. If we look more deep inside the algorithm of Gauss-Seidel method, we know that the iteration rule of it is as follow:

$$x^{(k+1)} = (D+L)^{-1}Ux^{(k)} + (D+L)^{-1}b$$

Recall that the convergence rate of stationary iterative method will be same as the convergence rate of $\|B\|$ to zero, where B is the iteration matrix. Also, the convergence rate of $\|B\|$ to zero is equal to the convergence rate to zero of B 's top eigenvalue λ_{max}. Using $Matlab$ to calculate the absolute value of the top eigenvalue of the iteration matrices of each size.

On the other hand, we know the bond on the error after k iterations of the PCG algorithm:[2, pp.155-156]

$$\|x^{(k)} - x^*\|_A \le \frac{2c^k}{1+c^{2k}}\|x^{(0)} - x^*\|_A$$

	Size = 9	Size = 49	Size = 225
λ_{max}	-0.509587	-0.854224	-0.961987

Table 4: top eigenvalue of the iteration matrices

where $c = \frac{\sqrt{\kappa(\hat{G})}-1}{\sqrt{\kappa(\hat{G})}+1}$, $\hat{G} = I - G$, G is the iteration matrix

This is a tighter upper bound for iteration error than the upper bound given in the homework description.

$$\|x^{(k)} - x^*\|_A \leq 2(\frac{\sqrt{\kappa(\hat{G})} - 1}{\sqrt{\kappa(\hat{G})} + 1})^k \|x^{(0)} - x^*\|_A$$

There are some papers claiming that the rate of convergence of PCG with some specific algorithm in certain cases are quadratic. However, I'm not really understand the proof. A result I found in a paper[3] seems more reasonable, although it is also a loose upper bound.

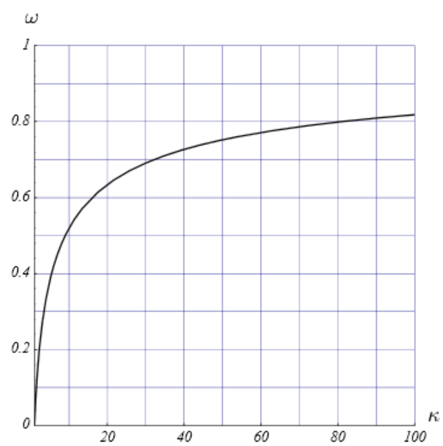

Figure 3: Convergence of Conjugate Gradients (per iteration) as a function of condition number

The estimation is based on condition number. So let's calculate the condition number of iteration matrices of different size. The results are as follow:

Consider the iteration matrix with size = 225. Using extrapolation to estimate the convergence rate per iteration is about 0.83. From Table 2, the convergence rate of same system with Gauss-Seidel method is 0.96. Thus we can calculate

Method	Size = 9	Size = 49	Size = 225	Size 961
$\kappa(A)$	9.000	37.265	150.417	603.052

Table 5: Condition Number of iteration matrices with different size

the ratio of the number of iterations between the two methods at the same error level. The result is $\frac{\log 0.96}{\log 0.83} = 0.2191$, which is closed to the ratio of real number of iterations: $\frac{10}{85} = 0.1176$

After comparing the the convergence rate of Gauss-Seidel and PCG, we can learn that the convergence rate of PCG is far more faster than Gauss-Seidel when the iteration will converge.

In the end of the discussion about number if iterations, let's consider only using PCG method with preconditioner to be identity matrix and incomplete Cholesky matrix. Since these two methods are relatively fast, I will increase the size of the linear system and observe how the number of iterations change.

Method	9	49	225	961	3969	16129
PCG	2	4	10	23	47	95
PCG-IC	2	3	9	17	32	66

Table 6: Number of iterations using PCG and PCG-IC iterative methods. The first row is the size of the linear system.

It's clearly that the number of iterations are smaller when using incomplete Cholesky preconditioner in PCG method. The reason is that the condition number of iteration matrix of PCG-IC is significantly smaller than the usual PCG method. Note that we should consider the condition number of $\hat{G} = I - G$. The condition number of each $\hat{G}$ with different size are listed as follow:

Method	Size = 9	Size = 49	Size = 225	Size 961	Size = 3969
$\kappa(A)$	9.000	37.265	150.417	603.052	2413.599
$\kappa(P^{-1}A)$	1.741	6.458	24.846	100.681	403.525

Table 7: Compare the condition number of A and preconditioned $P^{-1}A$

2.2 Elapsed Time

In the beginning, let's take a look at the elapsed time using four different iterative linear system solving method:

Method	Size = 9	Size = 49	Size = 225	Size = 961	Size = 3969	Size = 16129
GS	0.0001	0.0027	0.0795	5.9445		
PCG	0.0002	0.0003	0.0009	0.0029	0.0158	0.1079
PCG-IC	0.0004	0.0002	0.0004	0.0019	0.0132	0.1248
Sym-GS	0.0002	0.0036	0.1277	8.8022		

Table 8: Elapsed time using four iterative methods

The elapsed time is proportional to the number of iterations multiply an ordered of the iteration matrix size plus some initialization fixed time.

$$t_{elapsed} = n_{iteration} * t_{iteration} + t_{initialization}$$

where, $t_{iteration} = f(n_{matrixSize})$

First, let's consider the difference between the four methods. If we divide the elapse time by the number of iteration, we will get the result as follow:

Method	Size = 9	Size = 49	Size = 225	Size = 961	Size = 3969	Size = 16129
GS	0.0000	0.0002	0.0009	0.0145		
PCG	0.0001	0.0001	0.0001	0.0001	0.0003	0.0011
PCG-IC	0.0004	0.0001	0.0001	0.0002	0.0005	0.0022
Sym-GS	0.0001	0.0003	0.0019	0.0274		

Table 9: Elapsed time divided by number of iteration of iterative methods

We can clearly see that the mean elapsed time per iteration is significantly smaller when using Conjugate methods than using Gauss-Seidel methods. Let's dig into each algorithm and see why PCG is faster.

The following analysis will separated in three parts. The first part is the analysis in Gauss-Seidel and symmetric Gauss-Seidel. The second analysis is about PCG methods with different preconditioners. The last part is the comparison of Gauss-Seidel and PCG methods.

2.2.1 Gauss-Seidel

Consider a $n * n$ linear system. In Gauss-Seidel method, there are n vector multiplication for each iteration. Moreover, for symmetric Gauss-Seidel method, there are $2n$ vector multiplication for each iteration. As a result, we can see that the elapsed time of symmetric Gauss-Seidel is about two times to the elapsed time of Gauss-Seidel.

Since each vector multiplication takes n operations, we can conclude that the complexity of each iteration is $O(n^2)$.

2.2.2 PCG

On the other hands, let's consider PCG methods. The whole process involves a matrix inverse operations, and several matrix vector multiplications. We discuss the two main operations respectively.

The matrix inverse operation is done in preprocessed part of the algorithm which will be only calculated once. For the PCG using identity matrix as preconditioners, inverse operation takes no additional calculation. When using incomplete Cholesky preconditioners, the preprocessing part has two operations: incomplete Cholesky factorization and an inverse operation on a triangular matrix. The complexity of Cholesky factorization is $n^3/3$. Also, the complexity of triangular matrix inverse operation is about $n^3/3$. As a result, the complexity of the preprocessing part of the PCG using incomplete Cholesky preconditioners is $O(n^3)$.

As to the matrix vector multiplications part, the complexity is only $O(n^2)$

2.2.3 Gauss-Seidel and PCG

From the above analysis, we can see that both Gauss-Seidel and PCG algorithm has $O(n^2)$ complexity in each iteration. PCG with incomplete Cholesky preconditioners are expensive during preprocessing since it has to factorize a matrix in $O(n^2)$ and inverse a triangular matrix.

Let's study the result, it shows that the cost of each iteration is more expensive when using Gauss-Seidel algorithm. However, the analysis tells us that the complexity of both algorith are in the same order, $O(n^2)$. When using Gauss-Seidel algorithm, it performs n vector multiplications, while in PCG algorithm, it performs a product of matrix and vector. Both n vector multiplications and product of matrix and vector take about $O(n^2)$ operations. Why is that they have the same complexity order but result in very different result?

The reason is that the n vector multiplications performed in Gauss-Seidel is not parallel, it's sequential! As a result, in each iteration, it needs to complete the $O(n^2)$ operations sequentially, while the product of matrix and vector in PCG can be calculated in parallel. We compare the elapsed time used in a product of matrix and vector with two different methods: the first method sequentially calculates the vector multiplications for n times, and the other is to calculate the result using the matrix vector multiplication operation directly in Matlab. The results are as follow:

Method	Size = 9	Size = 49	Size = 225	Size = 961	Size = 3969	Size = 16129
Sequential	0.0000	0.0002	0.0009	0.0134	0.2693	7.0105
Direct	0.0000	0.0000	0.0001	0.0005	0.0081	0.1793

Table 10: Elapsed time calculating product of matrix and vector with different approach

We can easily see that when calculating the product of matrix and vector sequentially through every row takes more time than simply using the direct matrix vector multiplication function in Matlab.Especially when the system size grows up, the difference is much more bigger.

Note that the matrices we used in this experiment is the linear system matrices in the homework problem. So they are all sparse matrices. For a non-sparse matrix, the result will be more dramatically because of more arithmetic operations.

2.3 Storage

The storages used in different size of linear system with each iterative method are listed as follow:

Method	Size = 9	Size = 49	Size = 225	Size = 961	Size = 3969	Size = 16129
GS	752	4656	22448	97968		
GS(full)	792	19992	408600	7403544		
PCG	968	5832	27848	121032	504008	2056392
PCG-IC	2104	13016	62840	286392	1204152	4928824
Sym-GS	792	4656	22448	97968		
Sym-GS(full)	792	19992	408600	7403544		

Table 11: Storage using four iterative methods

From the above table, we can see that the storage used in PCG is a little greater than Gauss-Seidel algorithm using sparse data structure. However, when Gauss-Seidel algorithms choose to use full matrix data structure in order to accelerate elapsed time, the storage be dramatically increase and surpass the storage used in PCG algorithm.

For PCG algorithm using different preconditioners. When using incomplete Cholesky preconditioners, there are additional storage for inverse matrces, while using identity matrices as preconditioners don't have to allocate additional space.

3 Reordering

4 Remarks

During the whole experiment, the results had been modified various of time because of the design and implementation of each algorithms. In this section, we list some problem we had encountered and the improvement we have done.

4.1 Convergence

The number of iterations is very different in Gauss-Seidel algorithm and PCG algorithms. This is the result of different rate of convergence. We can easily understand the reason is that in PCG methods, each iteration gives the best guess in current state. In other words, PCG methods minimize the energy norm of the residual in current *Krylov subspace*. However, when we started to analyse the rate of convergence of PCG algorithms, we found it difficult to give an accurate estimation. The rate of convergence seems to lie between quadratic and linear convergence. Moreover, the upper bound given by the textbook or some other references are too loose.

Finally, we gave up seeking for a tight bound for the convergence rate of PCG methods. Instead, we compare the condition number of different iteration matrices. And showed that the smaller the condition number of the iteration matrix is, the less iterations of convergence are needed. Also, we found a paper discussed PCG method's rate of convergence.[3]

4.2 Sparse of Full

In the first version of our algorithms implementation, the result of Gauss-Seidel algorithm is very bad. The elapsed time is much greater than the time used in PCG methods.

After a discussion with prof.Wang, we found out that the vector multiplications in in each iteration do not utilize the benefit of sparse structure. In contrast, the sparse structure makes the vectorized multiplications more expensive because of the row extraction operations. As a result, sparse structure doesn't improve the performance, and even makes the result worse.

Then we change the data structure of linear system matrix from sparse structure to full structure. Although the amount of storage significantly increased, the elapsed time was improved. This is a trade off between time and space.

4.3 Vector Multiplication is Expensive

After changing the matrix data structure, there's still exists a large gap between the elapsed time per iteration on Gauss-Seidel method and PCG method. Just as the reason mentioned above, the cost of n vector multiplications are much greater than the cost of product between matrix and vector.

4.4 Inverse Operations

In the classroom log on 11/26, TA said that it's abandoned to use Matlab functions to calculate inverse matrix or solve linear system problem in PCG methods. When using incomplete Cholesky preconditioners, the inverse process involves two backward substitutions. However, the linear system in this homework is sparse. As a result, direct backward substitution will have poor performance. To solve this problem, the algorithm needs to utilize sparse structure. Because we are not familiar with sparse algorithm, so we still use Matlab built-in function to solve the inverse problem.

Reference

[1] Jerry Chou. Matlab code for this homework.

[2] Sacco R. Quarteroni A. and Saleri F. *Numerical Mathematics*. Springer-Verlag, Berlin, 2007.

[3] Jonathan Richard Shewchuk. An introduction to the conjugate method without agonizing pain. August 1994.