2016

Is the precision of computed solutions more closely related with componentwise condition number than normwise condition number?

Don Bing Dong Tan
Hong Kong Baptist University

Is The Precision Of Computed Solutions More Closely Related With Componentwise Condition Number Than Normwise Condition Number?

TAN Bing Dong, Don

A thesis submitted in partial fulfillment of the requirements

for the degree of

Master of Philosophy

Principal Supervisor: Dr. Dennis Cheung

Hong Kong Baptist University

May 2015

Declaration

I declare that this thesis has been composed by myself under the guidance of my principal supervisor Dr. Dennis Cheung. The thesis has not previously included in any thesis, dissertation or report submitted to any institution for a degree, diploma or other qualification. All sources of information have been acknowledged by means of references to the relevant publications.

Signature:_____

May 2015

Abstract

We have a conjecture that "the precision of computed solutions for systems of linear equations is more closely related with componentwise condition number $c(A)$ than normwise condition number $\kappa(A)$". We conducted simulation experiments to verify this conjecture. A statistical tool, Hotelling-Williams T-Test is employed to check if difference between correlations is significant. Simulation results suggest that our conjecture is true for most of the well-known methods and matrix sizes.

Keywords: condition numbers, simulation, correlation coefficients, Hotelling-Williams T-Test

Acknowledgements

I would like to express my gratitude to all those who helped me during the writing of this thesis. I gratefully acknowledge the help of my supervisor, Dr. Dennis Cheung, who has given me valuable suggestions in these two years of studies. In the preparation of the thesis, he has spent much time to read and check each draft. Without his patient instruction and expert guidance, the completion of this thesis would not be possible.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Condition numbers are **defined to measure the sensitivity** of the problem outputs to small perturbations of the problem inputs. For the problem of solving system of linear equations ($Ax = b$), condition number is defined as follows, where $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$ are matrix and vector respectively.

$$\kappa(A) = \lim_{\delta\to 0} \sup \left\{ \frac{\|\tilde{x} - x\|}{\|x\|\,\delta} : \frac{\|\tilde{A} - A\|}{\|A\|} \leq \delta,\ \frac{\|\tilde{b} - b\|}{\|b\|} \leq \delta,\ \tilde{A}\tilde{x} = \tilde{b},\ Ax = b \right\},$$

where $\|A\|$ denotes the operator norm (i.e. the largest singular value) of A and $\|x\| = \sqrt{\sum_{i=1}^{n} x_i^2}$ denotes the two norm of x. We will keep using these definitions of norm throughout the rest of the paper.

In additional to measuring sensitivity, condition numbers are also **considered as an indicator of the difficulty** to compute accurate solutions when there are round-off errors. Conjugate gradient method (CGM) is one of the many methods solving systems of linear equations. (CGM) is an iterative method and the rate of convergence depends on $\kappa(A)$ (see [4]).

Preconditioning is a process and also a research topic about reducing condition numbers. Researchers work on this topic because of the following **belief**.

The smaller is the condition number,

the easier it is to compute an accurate solution.

For another problem **Linear Program (LP)**, there are similar results. In [1], there is an algorithm solving (LP) with a complexity upper bound which is written in terms of a condition number when round-off errors are considered.

However, so far, we can not find any complexity lower bound which is in terms of condition number. Instead, in [3] it has been found that, for random triangular matrices A, condition number is usually huge. That is, if all entries below the diagonal is equal to zero and other entries in the matrix A are i.i.d. random variable following standard normal distribution $N(0,1)$,

$$\mathbb{E}(\ln(\kappa(A))) \geq \Omega(n).$$

However, in practice, computed solutions are usually very accurate (see [5]). These theoretical results together the practical experience **contradicts** the **belief** that the smaller is the condition number, the easier it should be to compute an accurate solution.

To explain the above contradiction, researchers work on the **componentwise condition number** $c(A)$. Below is the definition of $c(A)$ for the problem of solving system of linear equations ($Ax = b$).

$$c(A) = \lim_{\delta \to 0} \sup \left\{ \frac{|\tilde{x}_j - x_j|}{|x_j|\,\delta} : \max_{1 \leq i,j \leq n} \frac{|\tilde{a}_{ij} - a_{ij}|}{|a_{ij}|} \leq \delta,\ \frac{|\tilde{b}_i - b_i|}{|b_i|} \leq \delta, \tilde{A}\tilde{x} = \tilde{b},\ Ax = b \right\}.$$

$c(A)$ is called componentwise since the errors $\frac{|\tilde{a}_{ij} - a_{ij}|}{|a_{ij}|}$ appearing in the above definition is componentwise. We will call $\kappa(A)$ the normwise condition number. In [2], it has been proved that, for random triangular matrices A, $c(A)$ is usually small and

$$\mathbb{E}(\ln(c(A))) \leq O(\log(n)).$$

1.1 Conjecture, goal, suggestion and value of this paper

Because of the above theoretical results and the practical experience, we have the following **conjecture**.

Comparing with the normwise condition number $\kappa(A)$,
the componentwise condition $c(A)$ has a closer relationship
with the accuracy of computed solutions.

The main **goal** of this paper is to verify the above conjecture by simulation experiments. Our simulation results suggest that our conjecture is true for most of the well-known methods and matrix sizes.

Our finding should be **valuable** for researchers who work on the topic of preconditioning. To the best of our knowledge, all existing research works on preconditioning target at reducing k(A). Our simulation results **suggest** that reducing $c(A)$ may be a better approach than reducing $\kappa(A)$ if one targets at computing solutions with high accuracy.

1.2 Organization of this paper

We will organize this paper as follows. This chapter 1 has already covered the motivation, origin, goal and suggestion and values of this paper. In chapter 2, we will describe all details about our simulation method, including Probability Model, the methods to compute condition numbers and precision of computed solutions, etc. We will also restate our conjecture in a more formal way and introduce statistical method for correlation comparison method in chapter 2. In chapter 3, we will show the correlation coefficients between the losses of precision and the precision of solution computed by different methods. In chapter 4, Hotelling-Williams T-test will be employed to examine our conjectures with the simulation

results.

Chapter 2

Simulation methods

Our simulation program is written based on algorithm 1 below.

2.1 Probability model

We will generate independently and identically distributed (i.i.d.) entries of A and x with the standard normal distribution, i.e. $N(0,1)$ and $b = Ax$. We generate x instead of b as we need to know the precise solution x to compute the precision of computed solutions.

A more commonly used model is as follows. Entries of A and b are generated i.i.d. with the standard normal distribution, i.e. $N(0,1)$ and $x = A^{-1}b$. We claim that our simulation results should hold for both models because of the following reasons.

- In both models, the direction of b, i.e. $\dfrac{b}{\|b\|}$, follows the same probability distribution (uniform distribution on the unit sphere in $\mathbb{R}^n$).

- The relative error of the computed solution $\tilde{x}$, i.e. $\dfrac{\|\tilde{x} - x\|}{\|x\|}$ is independent of scaling of the vector b.

Data: Matrix size n

Result: The 10 correlation coefficients $r_{L_C,P_{\mathrm{MLD}}}$, $r_{L_N,P_{\mathrm{MLD}}}$, ... $r_{L_N,P_{\mathrm{CGM}}}$ mentioned in section 2.8

$k := 0$ (k denotes the sample size);

Terminate := 0;

while *Terminate = 0* **do**

  Generate A, x and b as described in section 2.1;

  Compute L_C and L_N as described in section 2.3;

  Compute P_C^{MLD}, P_N^{MLD}, ... as described in section 2.5;

  if $k \geq 2^{11}$ *and is a power of* 2 **then**

    Compute the 10 correlation coefficients $r^{\mathrm{FH}}_{L_C,P_{\mathrm{MLD}}}$, $r^{\mathrm{FH}}_{L_N,P_{\mathrm{MLD}}}$, ... $r^{\mathrm{FH}}_{L_N,P_{\mathrm{CGM}}}$ as described in section 2.9;

    Compute the 10 correlation coefficients $r_{L_C,P_{\mathrm{MLD}}}$, $r_{L_N,P_{\mathrm{MLD}}}$, ... $r_{L_N,P_{\mathrm{CGM}}}$ as described in section 2.9;

    if Error < 0.002 *as described in section 2.9* **then**

      Terminate := 1;

    end

  end

  $k := k + 1$;

end

Algorithm 1: Pseudocode for our simulation program

2.2 Computing condition numbers

To compute the condition number $\kappa(A)$, we will use the following formula .

$$\kappa(A) \;=\; \|A\|\,\|A^{-1}\|. \tag{2.2.1}$$

To the best of our knowledge, there is no explicit formula to compute $c(A)$. So, instead of $c(A)$, we will compute the following componentwise condition number $c^{\det}(A)$.

$$c^{\det}(A) \;=\; \sum_{1\le i,j\le n} \left|a_{ij}\, A^{-1}_{ji}\right|, \tag{2.2.2}$$

where a_{ij} and A^{-1}_{ij} are respectively entries of A and A^{-1} on ith row and jth column. The following two results (2.2.3) and (2.2.4) can be found in [2]. They tell us $c^{\det}(A)$ is a close relative of $c(A)$ and actually a condition number for the problem of computing determinant.

$$c^{\det}(A) \;\le\; c(A) \;\le\; 2\,c^{\det}(A). \tag{2.2.3}$$

$$c^{\det}(A) \;=\; \lim_{\delta\to 0}\sup\left\{\frac{|\det(\tilde{A})-\det(A)|}{|\det(A)|\,\delta} \;:\; \max_{1\le i,j\le n}\frac{|\tilde{a}_{ij}-a_{ij}|}{|a_{ij}|}\le\delta\right\}. \tag{2.2.4}$$

2.3 Computing Losses of Precision

Instead of $\mathbb{E}(\kappa(A))$, most researchers work on $\mathbb{E}(\ln(\kappa(A)))$. So do we. It is because $\mathbb{E}(\kappa(A))=\infty$ is undefined. $\ln(\kappa(A))$ is also called the Loss of Precision. Roughly speaking, it is equal to the difference between the precision (the number of accurate significant figures/digits) of the problem input (A and b) and the precision of the problem output (x). **Denote** by the normwise and componentwise Losses of Precision by

$$L_N = \ln(\kappa(A)) \qquad \text{and} \qquad L_C = \ln(c^{\det}(A)).$$

2.4 Methods for solving systems of linear equations

Since there are too many methods for solving systems of linear equations, it is impossible for us to cover them all. In this paper, we have chosen the following five methods. We believe they are the most commonly used and also the most well known methods for solving systems of linear equations.

1. (MLD) MATLAB command - mldivide ($x = A\backslash b$).
2. (GEP) Gaussian Elimination Method with partial pivoting.
3. (GEM) Gaussian Elimination Method without partial pivoting.
4. (QRD) QR Decomposition.
5. (CGM) Conjugate Gradient Method.

Denote the solutions computed by the methods (MLD), (GEP), (GEM), (QRD) and (CGM) respectively by

$$x^{\mathrm{MLD}},\ x^{\mathrm{GEM}},\ x^{\mathrm{GEP}},\ x^{\mathrm{QRD}} \text{ and } x^{\mathrm{CGM}}.$$

2.4.1 (MLD) MATLAB command - mldivide and (GEP) Gaussian Elimination Method with partial pivoting

"$x = A\backslash b$" is the most commonly used and also the most standard command for solving system of linear equations in MATLAB. That is why we include this command in our simulation experiments.

According to the web-site of "MATHWORKS", the developer of MATLAB, the (MLD) command "$x = mldivide(A, b)$" is the same as the command "$x = A\backslash b$".

In this paper, since the generated matrix A is a square, non-triangular and non-symmetric matrix, this command (MLD) will apply Gaussian Elimination Method with partial pivoting (GEP).

To be safe, we have also written the MATLAB code for the (GEP) method by ourselves. Based on the Algorithm 2, one part of our MATLAB simulation program is written to compute x_{GEP}.

Data: Matrix size n, matrix A and vector b

Result: Solution x_{GEP}

for $k = 1$, ..., n **do**

  $i_{\max} := \arg\max_{i \geq k} |a_{ik}|$;

  Swap the kth and the $i_{\max}$th rows;

  for $j = k + 1$, ..., **do**

    $a_{ij} := a_{ij} - \dfrac{a_{kj}\, a_{ik}}{a_{kk}}$;

  end

end

Compute x_{GEP} by backward substitution;

Algorithm 2: (GEP) Gaussian Elimination Method with partial pivoting

We expected that the command (MLD) and the method (GEP) will find the same solution. Surprisingly, our simulation results show that solutions found by them are not completely the same, although they are very close to each other. We suspect that the command (MLD) applies some advanced techniques without mentioning in the help function and the web-site of "MATHWORKS".

2.4.2 (GEM) Gaussian Elimination Method without partial pivoting

Although (GEM) Gaussian Elimination Method without partial pivoting is not commonly used, it is the most well-known and also the most elementary method for solving system of linear equations. That is why we include this method in our simulation experiments.

It is a little bit simpler than the (GEP) method. Based on Algorithm 3, one part of our MATLAB simulation program is written to compute x_{GEM}. As you see, Algorithm 3 is two lines shorter than algorithm 2.

Data: Matrix size n, matrix A and vector b

Result: Solution x_{GEM}

for $k = 1, ..., n$ **do**

  for $j = k + 1, ...,$ **do**

    $a_{ij} := a_{ij} - \dfrac{a_{kj}\, a_{ik}}{a_{kk}}$;

  end

end

Compute x_{GEP} by backward substitution;

Algorithm 3: (GEM) Gaussian Elimination Method without partial pivoting

2.4.3 (QRD) QR Decomposition

(QRD) method is also included in our simulation experiments since it is also well known. According to the website of "MATHWORKS", the command (MLD) applies this (QRD) method when A is not a square matrix. Algorithm 4 below is its Pseudocode for our MATLAB simulation program.

Data: Matrix size n, matrix A and vector b

Result: Solution x_{QRD}

Compute the QR decomposition by command $[Q, R] = qr(A)$;

Compute x_{QRD} by MATLAB command "$x_{\text{QRD}} = R\backslash(Q' * b)$";

Algorithm 4: (QRD) QR Decomposition

2.4.4 (CGM) Conjugate Gradient Method

(CGM) Conjugate Gradient Method is also a well known method for solving system of linear equations, especially for sparse matrices. There is another reason why we include this method in our simulation experiments. Research works show that the rate of convergency of the Conjugate Gradient Method (CGM) depends on the condition number $\kappa(A)$.

Based on Algorithm 5, a part of our MATLAB simulation program is written to compute x_{CGM}. Note that the starting point is $x_0 = 0$ and the number of iteration is $2n$. In theory (but not in practice), with arbitrary starting point x_0, (CGM) output the exact solution after n iterations provided that round-off error is absent.

In practice, the number of iterations is usually not fixed. Instead the program will keep running until the residual (error) is small enough. We chose to fixed the number of iterations (instead of fixed the error) as we want to our simulation experiments to be consistent for different methods.

Data: Matrix size n, matrix A and vector b

Result: Solution x_{CGM}

$b := A^T b$ and $A := A^T A$;

$x_0 := 0$, $r_0 := b$ and $p_0 = r_0$;

for $k = 1, ..., 2n$ **do**

$\alpha_k := \frac{r_k^T r_k}{p_k^T A p_k}$;

$x_{k+1} := x_k + \alpha\, p_k$;

$r_{k+1} := r_k - \alpha\, A\, p_k$;

$\beta_k := \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$;

$p_{k+1} := r_{k+1} + \beta_k\, p_k$;

k:=k+1;

end

$x_{\mathrm{CGM}} = x_{2n}$;

Algorithm 5: (CGM) Conjugate Gradient Method

2.5 Computing the precision of computed solutions

Define the precision of computed solutions as follows.

$$P_{\mathrm{MLD}} = \ln\left(\frac{\|x\|}{\|x_{\mathrm{MLD}} - x\|}\right) \quad \text{and} \quad P_{\mathrm{GEP}} = \ln\left(\frac{\|x\|}{\|x_{\mathrm{GEP}} - x\|}\right).$$

P_{GEM}, P_{QRD} and P_{CGM} are defined similarly. P_{CGM} is roughly equal to the number of accurate significant figures (digits) of the solution x_{CGM} computed by (CGM) Method.

2.6 Population correlation coefficients

For any two random variables X and Y, the **population correlation coefficient** is defined as follows.

$$\text{corr}(X,Y) \quad = \quad \frac{\mathbb{E}\left((X-\mathbb{E}(X))(Y-\mathbb{E}(Y)\right)}{\sigma_X \sigma_Y}, \qquad \text{where}$$

σ_x and σ_y denote the population standard derivation of x and y respectively. Correlation coefficient is always a real number between $+1$ and -1. If $\text{corr}(X,Y) \approx 1$ (or -1), there is strong increasing (or decreasing) correlation between the two random variables X and Y. If $\text{corr}(X,Y) \approx 0$, it is usually interpreted or considered as no relationship.

When both random variables X and Y have average values equal to zero, $\text{corr}(X,Y)$ has a nice and clear geometrical meaning as follows. Suppose $(x,y) \in \mathbb{R}^{k\times 2}$ is a matrix and (X,Y) is random vector $\in \mathbb{R}^2$, s.t. for $i = 1, 2, ..., k$,

$$\textbf{Prob}(X = x_i \text{ and } Y = y_i) = \frac{1}{k}.$$

Then, $\text{corr}(X,Y) = \dfrac{x^T y}{\|x\|\|y\|}$ is equal to the cosine of angle between x and y,

2.7 Sample correlation coefficients

For any two sets of sample data x and $y \in \mathbb{R}^k$, the **sample correlation coefficient** is defined as follows.

$$r_{x,y} \quad = \quad \frac{\sum_{i=1}^{k}(x_i-\bar{x})(y_i-\bar{y})}{(n-1)\, s_x\, s_y}, \qquad \text{where}$$

s_x and s_y denote the sample standard derivation of x and y respectively.

Note that $r_{x,y}$ is a random variable and $\text{corr}(X,Y)$ is a fixed number. In particle, when the sample size is big enough, after Fisher Transformation, $r_{x,y}$ is considered to be approximately normal distributed with the following mean and

variance.

$$\begin{aligned} \mathbb{E}\left(F\left(r_{x,y}\right)\right) &\approx F\left(\operatorname{corr}(X,Y)\right) \text{ and} \\ \mathbf{Var}\left(F\left(r_{x,y}\right)\right) &\approx \frac{1}{n-3}, \text{ where} \\ F(r) &= \frac{1}{2}\ln\frac{1+r}{1-r} = \tanh^{-1}(r). \end{aligned}$$

2.8 Restate our conjectures

With the above notations and definitions, we are now ready to restate our conjectures in a more formal way as follows.

$$\left|\operatorname{corr}\left(L_C, P_{\mathrm{MLD}}\right)\right| > \left|\operatorname{corr}\left(L_N, P_{\mathrm{MLD}}\right)\right|. \tag{2.8.5}$$

$$\left|\operatorname{corr}\left(L_C, P_{\mathrm{GEP}}\right)\right| > \left|\operatorname{corr}\left(L_N, P_{\mathrm{GEP}}\right)\right|. \tag{2.8.6}$$

$$\left|\operatorname{corr}\left(L_C, P_{\mathrm{GEM}}\right)\right| > \left|\operatorname{corr}\left(L_N, P_{\mathrm{GEM}}\right)\right|. \tag{2.8.7}$$

$$\left|\operatorname{corr}\left(L_C, P_{\mathrm{QRD}}\right)\right| > \left|\operatorname{corr}\left(L_N, P_{\mathrm{QRD}}\right)\right|. \tag{2.8.8}$$

$$\left|\operatorname{corr}\left(L_C, P_{\mathrm{CGM}}\right)\right| > \left|\operatorname{corr}\left(L_N, P_{\mathrm{CGM}}\right)\right|. \tag{2.8.9}$$

2.9 Accuracy of computed correlations and sample size

As a well known fact, the simulation results are random and vary from time to time. But as the sample size increase, the variation should become smaller and smaller. We wish all computed correlation coefficients are likely to be **accurate to at least** 2 **decimal places.** To do so, we will take the following approach.

We will first generate $2^{11} = 2048$ samples of L_C, L_N, P_{MLD}, P_{GEP}, ...P_{CGM}. By using the **first half** (i.e. $2^{10} = 1024$) of the generated samples, we will compute

the following 10 correlation coefficients.

$$r^{\mathrm{FH}}_{L_C,P_{\mathrm{MLD}}},\ r^{\mathrm{FH}}_{L_N,P_{\mathrm{MLD}}},\ r^{\mathrm{FH}}_{L_C,P_{\mathrm{GEP}}},\ r^{\mathrm{FH}}_{L_N,P_{\mathrm{GEP}}},\ r^{\mathrm{FH}}_{L_C,P_{\mathrm{GEM}}},\ r^{\mathrm{FH}}_{L_N,P_{\mathrm{GEM}}},$$
$$r^{\mathrm{FH}}_{L_C,P_{\mathrm{QRD}}},\ r^{\mathrm{FH}}_{L_N,P_{\mathrm{QRD}}},\ r^{\mathrm{FH}}_{L_C,P_{\mathrm{CGM}}} \text{ and } r^{\mathrm{FH}}_{L_N,P_{\mathrm{CGM}}}.$$

Similarly, by using **the whole set** of the generated samples, we will compute the following 10 correlation coefficients.

$$r_{L_C,P_{\mathrm{MLD}}},\ r_{L_N,P_{\mathrm{MLD}}},\ r_{L_C,P_{\mathrm{GEP}}},\ r_{L_N,P_{\mathrm{GEP}}},\ r_{L_C,P_{\mathrm{GEM}}},\ r_{L_N,P_{\mathrm{GEM}}},$$
$$r_{L_C,P_{\mathrm{QRD}}},\ r_{L_N,P_{\mathrm{QRD}}},\ r_{L_C,P_{\mathrm{CGM}}} \text{ and } r_{L_N,P_{\mathrm{CGM}}}.$$

Define

$$\textbf{Error} = \max\left\{\begin{array}{c} \left|r^{\mathrm{FH}}_{L_C,P_{\mathrm{MLD}}} - r_{L_C,P_{\mathrm{MLD}}}\right|, \\ \left|r^{\mathrm{FH}}_{L_N,P_{\mathrm{MLD}}} - r_{L_N,P_{\mathrm{MLD}}}\right|, \\ \vdots, \\ \left|r^{\mathrm{FH}}_{L_N,P_{\mathrm{CGM}}} - r_{L_N,P_{\mathrm{CGM}}}\right| \end{array}\right\}.$$

If **Error** < 0.002, we will accept and return the above correlation coefficients $r_{L_C,P_{\mathrm{MLD}}}$, $r_{L_N,P_{\mathrm{MLD}}}$, ..., $r_{L_N,P_{\mathrm{CGM}}}$. Otherwise, we will double the sample size k, generate more samples and check if **Error** < 0.002 again. We will keep doing so until **Error** < 0.002.

Chapter 3

Simulation results

Algorithm 1 has been implemented and executed with MATLAB for the matrix size $n = 8, 16, ..., 128$. We have stopped increasing the matrix size at $n = 128$ for two reasons. Firstly, our results can never cover all matrix sizes $n \in \mathbb{N}$ anyway. Instead, we will use line graphs to show the **trend** which should be more important than result for any particular value of matrix size n.

Secondly, we have tried the simulation program for $n = 256$. After running for a whole day, the program is still running. Table 3.1 shows sample size for different values of n. The numbers of generated samples are at least $2^{14} = 16\,384$.

Matrix size (n)	8	16	32	64	128
Sample size (k)	2^{16}	2^{18}	2^{18}	2^{17}	2^{14}

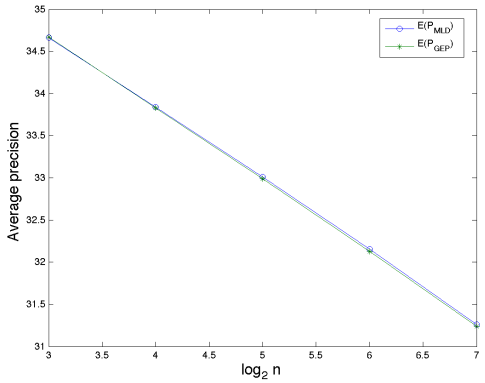
Table 3.1: Sample size of the simulation program

Matrix size n	8	16	32	64	128
$\mathbb{E}(P_{\mathrm{MLD}})$	34.6626	33.8394	33.0122	32.1538	31.2589
$\mathbb{E}(P_{\mathrm{GEP}})$	34.6708	33.8302	32.9880	32.1255	31.2371
$\mathbb{E}(P_{\mathrm{GEM}})$	33.1106	31.4461	29.8612	28.3478	26.8969
$\mathbb{E}(P_{\mathrm{QRD}})$	34.1242	33.4027	32.6718	31.9181	31.0949
$\mathbb{E}(P_{\mathrm{CGM}})$	31.8404	30.4969	29.1479	27.7222	26.3446

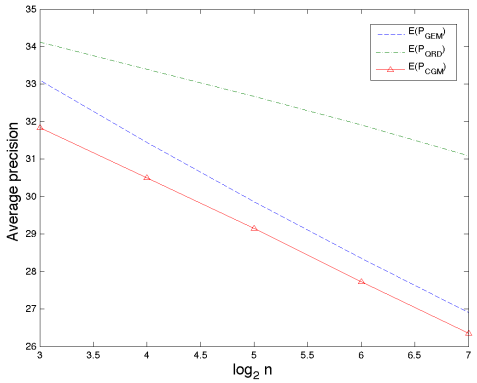
Table 3.2: Comparing average precisions of solutions computed by different methods

Figure 3.1: Numbers from table 3.2

(a) First two rows of table 3.2



(b) Remaining rows of table 3.2



Matrix size n	8	16	32	64	128
$r_{L_C,P_{\mathrm{MLD}}}$	-0.7822	-0.8024	-0.8090	-0.8137	-0.8204
$r_{L_N,P_{\mathrm{MLD}}}$	-0.7778	-0.7960	-0.8014	-0.8062	-0.8119
$r_{L_C,P_{\mathrm{MLD}}} - r_{L_N,P_{\mathrm{MLD}}}$	-0.0044	-0.0064	-0.0075	-0.0076	-0.0085

Table 3.3: Correlation coefficients between the losses of precision and the precision of solution computed by the command (MLD)

3.1 Average precisions of solutions computed by different methods

Tables 3.2 and figure 3.1 show that (MLD) is the most accurate method and (CGM) is the least accurate. According to the precision of computed solutions, we can rearrange the five methods as follows.

$$\begin{array}{ccccccc} (\mathrm{MLD}) & > & (\mathrm{GEP}) & > & (\mathrm{QRD}) & > & (\mathrm{GEM}) > (\mathrm{CGM}) \\ \text{(the most accurate)} & & & & & & \text{(the least accurate)} \end{array}$$

As mentioned, according to the website of "MATHWORKS", the command (MLD) applies the (GEP) method. However, from the first two rows of table 3.2, we can see that solutions found by (MLD) and (GEP) are not completely the same although they are very close to each other. We suspect that (MLD) may apply some advanced techniques without mentioning in the website of "MATHWORKS".

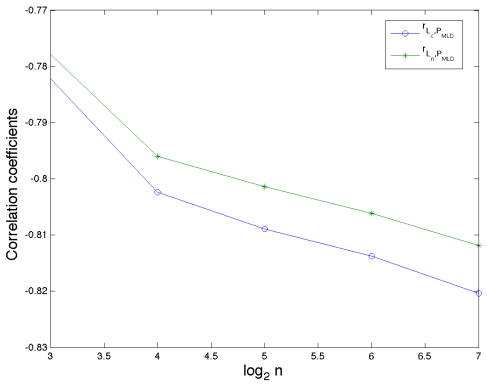
3.2 MATLAB command - mldivide (MLD)

Be reminded conjecture (2.8.5) is as follows.

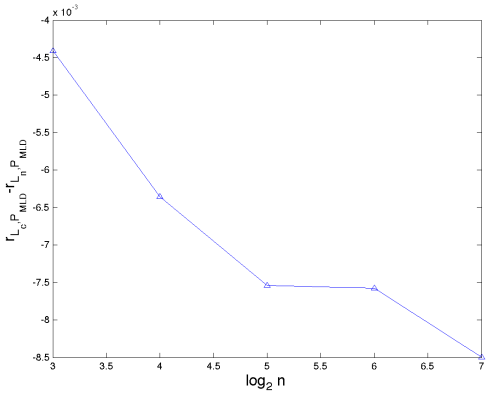
$$|\mathrm{corr}\,(L_C, P_{\mathrm{MLD}})| > |\mathrm{corr}\,(L_N, P_{\mathrm{MLD}})|\,.$$

Figure 3.2: Numbers from table 3.3

(a) First two rows of table 3.3



(b) Last row of table 3.3



The trend in the figure 3.2a suggests that, for all $n \in \mathbb{N}$,

$$\text{corr}\,(L_N, P_{\text{MLD}}) \;<\; 0. \tag{3.2.1}$$

The trend in the figure 3.2b suggests that, for all $n \in \mathbb{N}$,

$$\text{corr}\,(L_C, P_{\text{MLD}}) \;<\; \text{corr}\,(L_N, P_{\text{MLD}})\,. \tag{3.2.2}$$

Combining equations (3.2.1) and (3.2.2), our simulation results suggest that conjecture (2.8.5) is true for all $n \in \mathbb{N}$.

3.3 Gaussian Elimination Method with partial pivoting (GEP)

Be reminded conjecture (2.8.6) is as follows.

$$|\text{corr}\,(L_C, P_{\text{GEP}})| > |\text{corr}\,(L_N, P_{\text{GEP}})|\,.$$

The trend in the figure 3.3a suggests that, for all $n \in \mathbb{N}$,

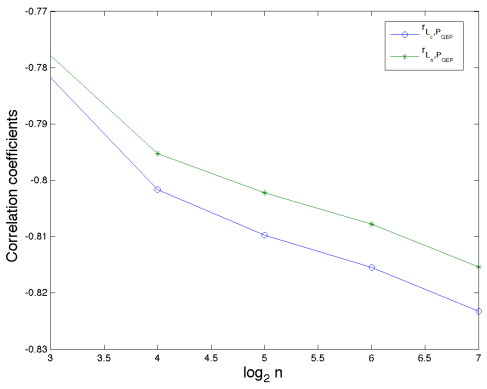
$$\text{corr}\,(L_N, P_{\text{GEP}}) \;<\; 0. \tag{3.3.3}$$

Matrix size n	8	16	32	64	128
$r_{L_C,P_{\text{GEP}}}$	-0.7818	-0.8017	-0.8097	-0.8154	-0.8233
$r_{L_N,P_{\text{GEP}}}$	-0.7778	-0.7953	-0.8022	-0.8078	-0.8154
$r_{L_C,P_{\text{GEP}}} - r_{L_N,P_{\text{GEP}}}$	-0.0039	-0.0064	-0.0076	-0.0077	-0.0079

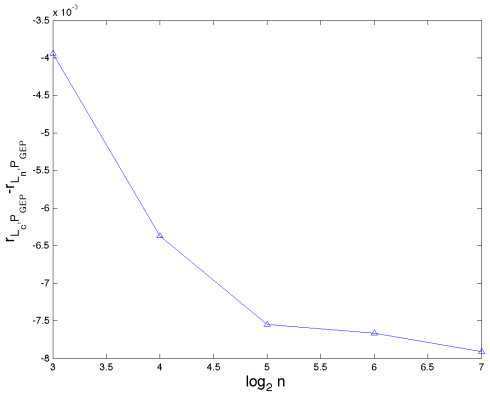
Table 3.4: Correlation coefficients between the losses of precision and the precision of solution computed by the method (GEP)

Figure 3.3: Numbers from table 3.4

(a) First two rows of table 3.4



(b) Last row of table 3.4



The trend in the figure 3.3b suggests that, for all $n \in \mathbb{N}$,

$$\text{corr}\,(L_C, P_{\text{GEP}}) \quad < \quad \text{corr}\,(L_N, P_{\text{GEP}})\,. \tag{3.3.4}$$

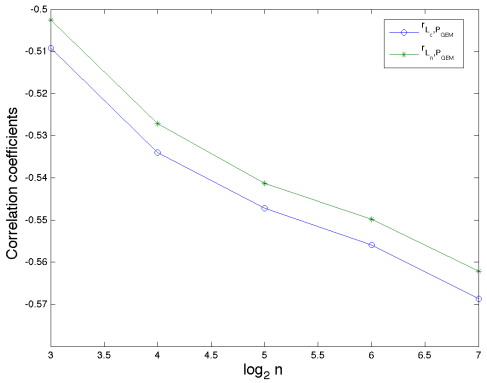
Combining equations (3.3.3) and (3.3.4), our simulation results suggest that conjecture (2.8.6) is true for all $n \in \mathbb{N}$.

Matrix size n	8	16	32	64	128
$r_{L_C,P_{\mathrm{GEM}}}$	-0.5092	-0.5340	-0.5473	-0.5559	-0.5687
$r_{L_N,P_{\mathrm{GEM}}}$	-0.5025	-0.5271	-0.5413	-0.5498	-0.5621
$r_{L_C,P_{\mathrm{GEM}}} - r_{L_N,P_{\mathrm{GEM}}}$	-0.0067	-0.0069	-0.0059	-0.0061	-0.0066

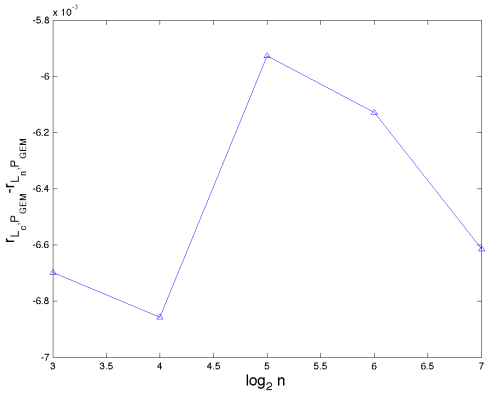
Table 3.5: Correlation coefficients between the losses of precision and the precision of solution computed by the method (GEM)

Figure 3.4: Numbers from table 3.5

(a) First two rows of table 3.5



(b) Last row of table 3.5



3.4 Gaussian Elimination Method without partial pivoting (GEM)

Be reminded conjecture (2.8.7) is as follows.

$$|\mathrm{corr}\,(L_C, P_{\mathrm{GEM}})| > |\mathrm{corr}\,(L_N, P_{\mathrm{GEM}})|\,.$$

The trend in the figure 3.4a suggests that, for all $n \in \mathbb{N}$,

$$\mathrm{corr}\,(L_N, P_{\mathrm{GEM}}) \;<\; 0. \tag{3.4.5}$$

The trend in the figure 3.4b is unclear for $n > 128$ but it still suggests that, for all $n \leq 128$,

$$\mathrm{corr}\,(L_C, P_{\mathrm{GEM}}) \;<\; \mathrm{corr}\,(L_N, P_{\mathrm{GEM}})\,. \tag{3.4.6}$$

Combining equations (3.4.5) and (3.4.6), our simulation results suggest that conjecture (2.8.7) is true for all $n \leq 128$ but it is unclear if (2.8.7) is true for $n > 128$.

3.5 QR Decomposition (QRD)

Be reminded conjecture (2.8.8) is as follows.

$$|\mathrm{corr}\,(L_C, P_{\mathrm{QRD}})| > |\mathrm{corr}\,(L_N, P_{\mathrm{QRD}})|\,.$$

The trend in the figure 3.5a suggests that, for all $n \in \mathbb{N}$,

$$\mathrm{corr}\,(L_N, P_{\mathrm{QRD}}) \;<\; 0. \tag{3.5.7}$$

The trend in the figure 3.5b suggests that, for all $n \in \mathbb{N}$,

$$\mathrm{corr}\,(L_C, P_{\mathrm{QRD}}) \;<\; \mathrm{corr}\,(L_N, P_{\mathrm{QRD}})\,. \tag{3.5.8}$$

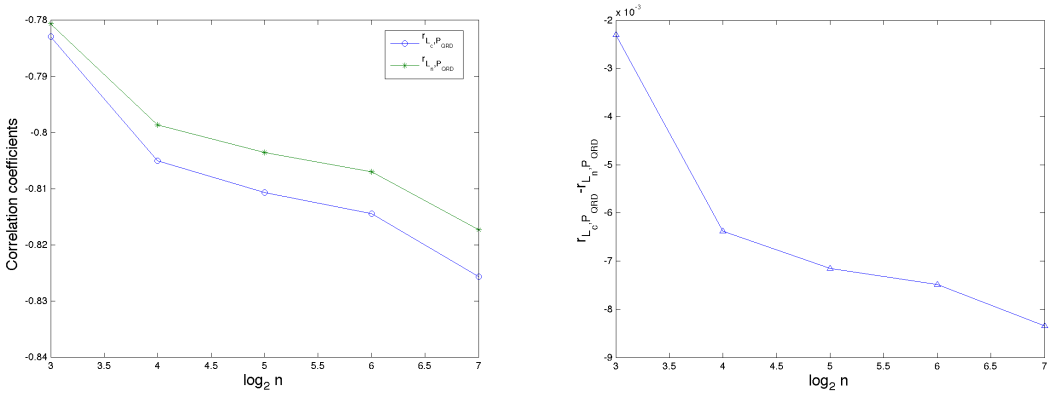
Combining equations (3.5.7) and (3.5.8), our simulation results suggest that conjecture (2.8.8) is true for all $n \in \mathbb{N}$.

Matrix size n	8	16	32	64	128
$r_{L_C,P_{\mathrm{QRD}}}$	-0.7829	-0.8050	-0.8106	-0.8144	-0.8256
$r_{L_N,P_{\mathrm{QRD}}}$	-0.7806	-0.7986	-0.8035	-0.8069	-0.8173
$r_{L_C,P_{\mathrm{QRD}}} - r_{L_N,P_{\mathrm{QRD}}}$	-0.0023	-0.0064	-0.0072	-0.0075	-0.0083

Table 3.6: Correlation coefficients between the losses of precision and the precision of solution computed by the method (QRD)

Figure 3.5: Numbers from table 3.6

(a) First two rows of table 3.6 (b) Last row of table 3.6

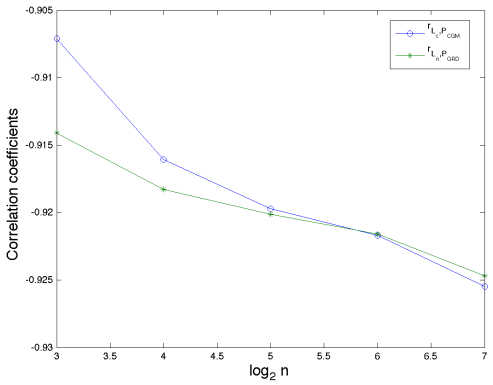


Matrix size n	8	16	32	64	128
$r_{L_C,P_{\mathrm{CGM}}}$	-0.9071	-0.9161	-0.9197	-0.9217	-0.9255
$r_{L_N,P_{\mathrm{CGM}}}$	-0.9141	-0.9183	-0.9201	-0.9216	-0.9247
$r_{L_C,P_{\mathrm{CGM}}} - r_{L_N,P_{\mathrm{CGM}}}$	0.0070	0.0022	0.0004	-0.0001	-0.0008

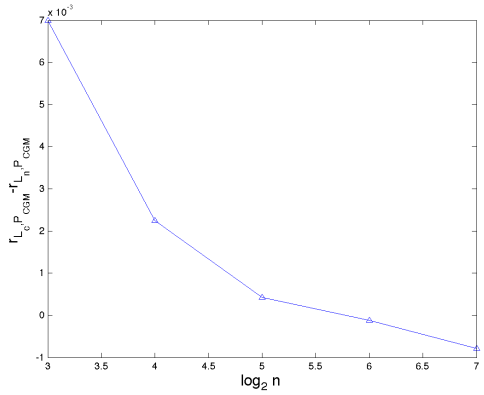
Table 3.7: Correlation coefficients between the losses of precision and the precision of solution computed by the method (CGM)

Figure 3.6: Numbers from table 3.7

(a) First two rows of table 3.7



(b) Last row of table 3.7



3.6 Conjugate Gradient Method (CGM)

Be reminded conjecture (2.8.9) is as follows.

$$|\text{corr}\,(L_C, P_{\text{CGM}})| > |\text{corr}\,(L_N, P_{\text{CGM}})|\,.$$

The trend in the figure 3.6a suggests that, for all $n \in \mathbb{N}$,

$$\text{corr}\,(L_N, P_{\text{CGM}}) \;<\; 0. \tag{3.6.9}$$

The trend in the figure 3.6b suggests that

$$\text{corr}\,(L_C, P_{\text{CGM}}) \;<\; \text{corr}\,(L_N, P_{\text{CGM}}) \qquad \text{when } n > 64. \tag{3.6.10}$$

$$\text{corr}\,(L_C, P_{\text{CGM}}) \;>\; \text{corr}\,(L_N, P_{\text{CGM}}) \qquad \text{when } n \leq 32. \tag{3.6.11}$$

Combining equations (3.6.9), (3.6.10) and (3.6.11), our simulation results suggest that conjecture (2.8.9) is true for $n > 64$ but wrong for $n \leq 32$.

Method	simulation result suggestions
(MLD)	Conjecture seems to be true for all $n \in \mathbb{N}$.
(GEP)	Conjecture seems to be true for all $n \in \mathbb{N}$.
(GEM)	Conjecture seems to be true for $n \leq 128$. But it is unclear if it is also true for $n > 128$.
(QRD)	Conjecture seems to be true for all $n \in \mathbb{N}$.
(CGM)	Conjecture seems to be true for $n > 64$. But it seems to be wrong for $n \leq 32$.

Table 3.8: Summary of our simulation result suggestions

3.7 Summarizing our simulation result suggestions

Table 3.8 summarizes our simulation suggestions. For the (MLD), (GEP) and (QRD) methods, it seems our conjecture is true, i.e. componentwise condition number $c(A)$ has a closer relation with the precision of computed solutions than normwise condition number $\kappa(A)$.

For (CGM) method, it seems our conjecture is true for $n > 64$. However, it seems our conjecture is wrong when $n \leq 32$.

For (GEM) method, it seems our conjecture is true for $n \leq 128$. However, the trend is unclear for $n > 128$. We are not so sure if our conjecture is also true when the matrix size $n > 128$. Perhaps, one can get a better result by increasing the sample size.

Chapter 4

Explanation for the small difference between the two correlation coefficients

From table 3.7, we can see that the difference between $r_{L_C,P_{\mathrm{CGM}}}$ and $r_{L_N,P_{\mathrm{CGM}}}$ is very small. Similar observation can also be seen in tables 3.3, 3.4, 3.5 and 3.6. It is because L_c and L_N are strongly correlated. From table 4.1 and figure 4.1, we can see that $\mathrm{corr}(L_C, L_N)$ is very closed to 1.

Indeed, for any random variables X, Y and Z, if $\mathrm{corr}(X, Y) = 1$, then

$$\mathrm{corr}(X, Z) = \mathrm{corr}(Y, Z).$$

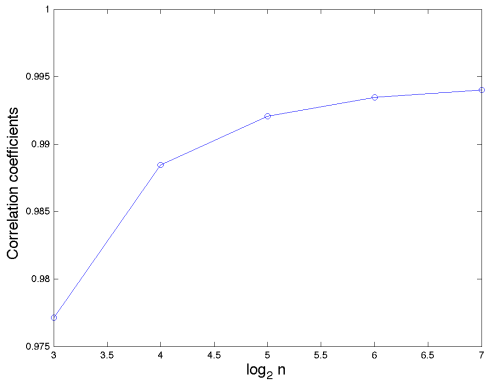
From table 4.1 and graph 4.1b, we can see that the correlation of L_c and L_N is strong correlated. So, it is difficult to compare the performance of component-wise condition number and normwise condition number. In this chapter, we will use Hotelling-Williams T-Test to provide a statistical verification, based on our simulation result.

Matrix size n	8	16	32	64	128
r_{L_C,L_N}	0.9771	0.9885	0.9921	0.9935	0.9940

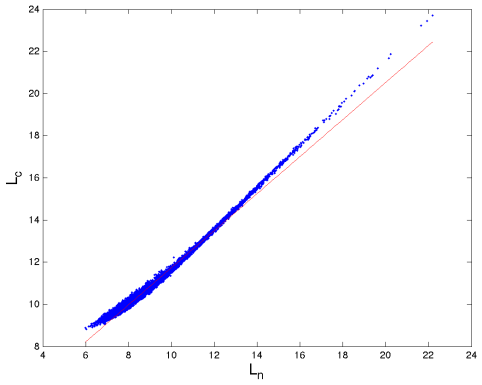
Table 4.1: Correlation coefficients between the normwise and the componentwise losses of precision

Figure 4.1: Numbers from table 4.1 and Scatter diagram example

(a) Numbers from table 4.1



(b) Scatter diagram for $n = 128$



4.1 Statistical Method For Correlation Coefficients Comparison

For simulation experiments, sample correlation coefficient will approach to population correlation coefficient as sample size getting large. As correlation coefficients of L_C and L_N towards the losses of precision are closed to each other, it is hard to check whether the differences are significant. Also, for the phenomena we observed, it is extremely difficult to conduct a mathematical proof. Fortunately, we still get statistical methods which can help. From Table 3.3, we can see that condition number is usually big when the precision is low. In other words, the accuracy is usually high when condition number is small. As an usual practice, we rewrite our conjuctures as statistical hypothesis below,

$$H_0 : \rho_{L_C,P_{MLD}} \geq \rho_{L_N,P_{MLD}} \quad \text{V.S.} \quad H_a : \rho_{L_C,P_{MLD}} < \rho_{L_N,P_{MLD}} \tag{4.1.1}$$

$$H_0 : \rho_{L_C,P_{GEP}} \geq \rho_{L_N,P_{GEP}} \quad \text{V.S.} \quad H_a : \rho_{L_C,P_{GEP}} < \rho_{L_N,P_{GEP}} \tag{4.1.2}$$

$$H_0 : \rho_{L_C,P_{GEM}} \geq \rho_{L_N,P_{GEM}} \quad \text{V.S.} \quad H_a : \rho_{L_C,P_{GEM}} < \rho_{L_N,P_{GEM}} \tag{4.1.3}$$

$$H_0 : \rho_{L_C,P_{QRD}} \geq \rho_{L_N,P_{QRD}} \quad \text{V.S.} \quad H_a : \rho_{L_C,P_{QRD}} < \rho_{L_N,P_{QRD}} \tag{4.1.4}$$

$$H_0 : \rho_{L_C,P_{CGM}} \geq \rho_{L_N,P_{CGM}} \quad \text{V.S.} \quad H_a : \rho_{L_C,P_{CGM}} < \rho_{L_N,P_{CGM}} \tag{4.1.5}$$

whereas, $\rho_{L_C,P_{MLD}}$ is the population correlation coefficient between L_C and P_{MLD} and the rests are similar. Suppose we have variables h; j; k. According to Steiger[6], "Hotelling-Williams T-test" is the best choice for correlation coefficients comparison when the null hypothesis is of this form $\rho_{jk} = \rho_{jh}$. That is similar to our case.

The definition of "Hotelling-Williams T test" is as follows:

$$t_{(n-3)} = (r_{j,k} - r_{j,h})\sqrt{\frac{(n-1)(r_{k,h})}{2(n-1)|R|/(n-3) + \bar{r}^2(1-r_{k,h})^2}}.$$

- n is sample size.

- $|R| = 1 - r_{j,k}^2 - r_{j,h}^2 - r_{k,h}^2 + 2(r_{j,k})(r_{j,h})(r_{k,h})$.
- $\bar{r} = (r_{j,k} + r_{j,h})/2$.
- degree of freedom is $n - 3$.

p-value is a conditional probability obtained form the t-statistic. p-value is suggesting the probability of observing the current phenomena when the null hypothesis is true. In Hypothesis Testing, the term "significant" means "the observed phenomena is not likely due to chance". Also, p-value is usually interpreted as follows:

- p value $> .10 \Leftrightarrow$ "not significant".
- p value $\leq .10 \Leftrightarrow$ "marginally significant".
- p value $\leq .05 \Leftrightarrow$ "significant".
- p value $\leq .01 \Leftrightarrow$ "highly significant".

4.2 Hypothesis Testing of MLD

Recall that the hypothesis (4.1.1) we stated in section 4.1 is as follows.

$$H_0 : \rho_{L_C,P_{MLD}} \geq \rho_{L_N,P_{MLD}} \text{ V.S. } H_a : \rho_{L_C,P_{MLD}} < \rho_{L_N,P_{MLD}}$$

As our concern is checking whether conjecture (2.8.5) is true, one-tail test will be applied. According to the p-values (all of them are smaller than 0.01) in Table (4.2), for each matrix size, we reject the null hypothesis and accepted the alternative hypothesis, $\rho_{L_C,P_{MLD}} < \rho_{L_N,P_{MLD}}$. That is, our observation is "highly significant". Conjecture (2.8.5) is very likely to be true for all $n \in \mathbb{N}$. Now we can conclude that componentwise condition number is more closely related with precision of computed solutions than normwise condition number in the MLD case.

Matrix size n	8	16	32	64	128
H.W T-statistic	-8.5081	-35.9131	-52.1382	-41.2776	-17.4151
p-value	$9.02 * 10^{-18}$	$4.64 * 10^{-282}$	0	0	$1.28 * 10^{-67}$

Table 4.2: H.W T-Statistic of MLD

Matrix size n	8	16	32	64	128
H.W. T-statistic	-7.6042	-35.9548	-52.3030	-41.9310	-16.3279
p-value	$1.45 * 10^{-14}$	$1.05 * 10^{-282}$	0	0	$9.21 * 10^{-60}$

Table 4.3: H.W T-Statistic of GEP

4.3 Hypothesis Testing of GEP

The hypothesis (4.1.2) in section 4.1 is as follows.

$$H_0 : \rho_{L_C,P_{GEP}} \geq \rho_{L_N,P_{GEP}} \text{ V.S. } H_a : \rho_{L_C,P_{GEP}} < \rho_{L_N,P_{GEP}}$$

As we mentioned before, MLD in matlab employs GEP method(in Table(4.3)). Similar to the MLD case, we reject the null hypothesis and have a strong confidence to accept the alternative hypothesis. We conclude that conjecture (2.8.6) is very likely to be true for all $n \in \mathbb{N}$.

4.4 Hypothesis Testing of GEM

Be reminded the hypothesis (4.1.3) of the GEM case is as follows.

$$H_0 : \rho_{L_C,P_{GEM}} \geq \rho_{L_N,P_{GEM}} \text{ V.S. } H_a : \rho_{L_C,P_{GEM}} < \rho_{L_N,P_{GEM}}$$

Matrix size n	8	16	32	64	128
H.W. T-statistic	-9.3153	-27.3370	-28.7861	-23.3402	-9.4149
p-value	$6.17 * 10^{-21}$	$1.31 * 10^{-164}$	$3.08 * 10^{-182}$	$1.52 * 10^{-120}$	$2.67 * 10^{-21}$

Table 4.4: H.W T-Statistic of GEM

Matrix size n	8	16	32	64	128
H.W. T-statistic	-4.4535	-36.2462	-49.6525	-40.8548	-17.3386
p-value	$4.23 * 10^{-6}$	$2.95 * 10^{-287}$	0	0	$4.74 * 10^{-67}$

Table 4.5: H.W T-Statistic of QRD

From Table(4.4), we can see p-values are smaller than 0.01 for all matric size. So, we reject the null hypothesis and have a strong confidence to state that $\rho_{L_C,P_{GEM}} < \rho_{L_N,P_{GEM}}$. We conclude that conjecture (2.8.7) is very likely to be true.

4.5 Hypothesis Testing of QRD

The results for QRD methods are similar to the methods listed before (see Table(4.5)).

Recall that the hypothesis (4.1.4) is as follows.

$$H_0 : \rho_{L_C,P_{QRD}} \geq \rho_{L_N,P_{QRD}} \text{ V.S. } H_a : \rho_{L_C,P_{QRD}} < \rho_{L_N,P_{QRD}}$$

As all the p-values are smaller than 0.01, we reject the null hypothesis and have a strong faith to state that $\rho_{L_C,P_{QRD}} < \rho_{L_N,P_{QRD}}$. The conjecture (2.8.8) is very likely to be true.

Matrix size n	8	16	32	64	128
H.W. T-statistic	20.9055	19.2905	4.4043	-1.0315	-2.4357
p-value	1	1	1	0.1512	0.0074

Table 4.6: H.W T-Statistic of CGM

4.6 Hypothesis Testing of CGM

For the CGM case, the results are very different from before. The results are reasonable because the principle of CGM is different from others'. CGM is usually implemented as an iterative method. The classic methods, like QRD, are implemented as direct methods.

Recall that the hypothesis (4.1.5) is as follows.

$$H_0 : \rho_{L_C,P_{CGM}} \geq \rho_{L_N,P_{CGM}} \text{ V.S. } H_a : \rho_{L_C,P_{CGM}} < \rho_{L_N,P_{CGM}}$$

When matrix size is 8, 16, 32 and 64, we could not reject the null hypothesis for p-values are larger than 0.1. When matrix size is 128, we reject the null hypothesis as p-value is less than 0.01. We have strong faith to state that $\rho_{L_C,P_{CGM}} < \rho_{L_N,P_{CGM}}$ when the matrix size is 128. We conclude that conjecture (2.8.9) is very likely to be true when $n \geq 128$.

Chapter 5

Conclusion

We have presented a statistical evidence to show that component wise condition number has a stronger correlation with precision of computed solutions rather than norm wise condition number. In chapter 1, we addressed our conjectures. In chapter 2, we briefly introduced the details of the simulation experiments and some concepts of precision and well known algorithms. We presented the simulation results in chapter 3. In section 3.7, we summarised the results. Since we could not find a way to conduct a mathematical proof, we verified the conjectures by employing statistical method H.W. T-statistic. In Chapter 4, we further examined the result we got in chapter 3 by using H.W. T-statistics. For classic methods, the simulation results are positive and suggest our conjuctures are true. However, the results of CGM case are different. When the matrix size is 8, 16, 32 and 64, the results are negative and suggest our conjecture are not true. The difference between the two correlations seems to be more and more siginificant when $n \geq 128$. Based on our simulation results, we could conclude that our conjecture is true for all the algorithms and all matrix sizes except CGM when $n < 128$. As we said in section 1.1, the simulation results suggest that a better approach of pre-conditioning is reducing $c(A)$, rather than reducing $\kappa(A)$.

Bibliography

[1] D. Cheung, F. Cucker and J. Pena, *Solving linear programs with finite precision: II. Algorithm*, Journal of complexity, 22 (2006), 305 - 335.

[2] D. Cheung and F. Cucker, *Component-wise condition numbers for random sparse matrices*, SIAM. J. Matrix Anal. & Appl. Volume 31, Issue 2 (2009), pp. 721-731

[3] D. Viswanath and L.N. Trefethen, *Condition numbers of random triangular matrices*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 564-581.

[4] van der Vorst, H. A.. *Iterative Krylov Methods for Large Linear systems*. Cambridge University Press, Cambridge, 2003.

[5] J. H. Wilkinson, *Rounding Errors in Algebraic Processes*, Prentice-Hall, Englewood Cliffs, NJ, 1963.

[6] Steiger, J.H. *Tests for comparing elements of a correlation matrix* Psychological Bulletin, 87, 245-251.

Curriculum Vitae

Personal Data

- Name: TAN BingDong
- Date of Birth: Nov 05,1989
- Nationality: Chinese

Education

- March. 2013 - now: Math Department, Baptist University. Master candidate.
- Sept. 2008 - July. 2012: S&T Department, UIC. Bachelor of Statistics.