Available online www.jocpr.com

Journal of Chemical and Pharmaceutical Research, 2014, 6(3):149-157

Research Article

ISSN : 0975-7384
CODEN(USA) : JCPRC5

The design of parallel least squares model based on the surface fitting problem

Aimin Yang[1,2*], Yingna Zhao[1], Tielei Tian[1] and Weixing Liu[1]

[1]Hebei United University, Tangshan, Hebei, China
[2]Yanshan University, College of Mechanical Engineering, Qinhuangdao, Hebei, China

---

ABSTRACT

With the field of science research is more and more widespread, to solve large-scale over determined system of linear equations has become a key problem, and the use of parallel processing technology also become the trend of the research. Firstly, in this paper we introduce the basic principle of the Serial least squares algorithm of the curve fitting, and based on the least square principle, we found an parallel least squares curve and surface fitting method; Secondly, combined with QR decomposition, we further improve the parallel least squares fitting algorithm. We also solve the problem of over determined system of linear equations and the curved surface fitting question in three-dimensional space. Finally, through the specific examples, comparing the scatter diagram to the fitting chart and considering the serial and parallel computing time, we can show the advantages and effectiveness of this method.

Keywords: Surface fitting Method; Least squares Model

INTRODUCTION

For a long time, the rapid development of parallel computer in order to improve the scientific compute speed, which makes the large science and engineering can be calculated and make scientific computing as a scientific research of three kinds of scientific method in scientific theory and scientific experiments. And with the computer hardware, software and the progress of the algorithm produced, this trend become strengthen[1]. With the increasing popularity of the high performance parallel machine, especially low price performance ratio of computer application and parallel programming standards, the high performance parallel computing has become a key supporting technology of China's science and engineering who want to apply the numerical simulation, and it is engaged in the science and engineering calculation of professional researchers or a foundation course that students want to study[2].

Least squares method is a kind of mathematical optimization technique. The basic idea is seeking for the best function matching data by minimize the sum of squares error. The fitting function method called the least squares fitting. Using least square method can easily get unknown data, and make the solution has the minimum sums of squares error of obtained data and real data[3]. The best matching function is called the least squares fitting function of the known data. Least squares fitting can be divided into linear least squares fitting and nonlinear least squares fitting.

In recent years, with the development of science and technology, in the circuit analysis, network analysis etc puts forward the problem which need to solve large-scale sparse linear equations.The development of the computer

science computing is an important field, and it is the key of numerical calculation to solve the linear system of equations, so solve the core of large-scale system of linear equations with the speed in scientific computing has very important significance. With the scientific research field become more and more widely, solving the large overdetermined system of linear equations has become possible. Using the parallel processing technology become the trend in the future. Solving large-scale overdetermined linear equations has many methods, but serial algorithm tend to be complicated, waste of time and not easy to promote[4]. Zhengjiu Kang gives the least square parameter estimation of parallel algorithm from parameter estimation, Zhixia Yang presents a new parallel algorithm in linear problem, Deren Wang puts forward some solutions to large linear problem parallel method according to matrix splitting technology, Zhong Chen analyzed the convergence of least squares parallel algorithm[5] , but the basic is not involved in the curve fitting least squares acceleration ratio and efficiency analysis ,there is no design surface fitting least squares method. Based on the partition strategy and master-slave mode,with the parallel QR decomposition method, this paper studies the curved surface and curve fitting least squares algorithm, and finally gives the surface fitting example,this study provides a new method solution of overdetermined linear equations.

PARALLEL LEAST SQUARES ALGORITHM

We mainly introduced the curve fitting least squares parallel algorithm design process, surface fitting problem can be finished after expanding.

Assume there is n dimension polynomial $y = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$, the error is defined as:

$$I = Y_i - y_i = Y_i - a_0 - a_1 x - a_2\ x^2 - \cdots - a_n x^n$$

Among them, Y_i is corresponding to the observation or experimental data, which has free error, in order to solve the value in sum of squares of the minimum:

$$S = \sum_{i=1}^{\mathrm{N}} I^2 = \sum_{i=1}^{N} (Y_i - a_0 - a_1 x_i\ - a_2 x_i^2 - \cdots - a_n x_i^n)^2$$

In the minimum point, the partial derivative,
$\partial S / \partial a_0$, $\partial S / \partial a_1$,$\cdots$, $\partial S / \partial a_n$ are zero. So we get $n+1$ equations:

$$\begin{cases} \dfrac{\partial S}{\partial a_0} = 0 = \sum_{i=1}^{N} 2\left(Y_i - a_0 - a_1 x_i - \cdots - a_i x_i^2\right)(-1) \\ \dfrac{\partial S}{\partial a_1} = 0 = \sum_{i=1}^{N} 2\left(Y_i - a_0 - a_1 x_i - \cdots - a_i x_i^2\right)(-x_i) \\ \qquad\qquad \vdots \\ \dfrac{\partial S}{\partial a_n} = 0 = \sum_{i=1}^{N} 2\left(Y_i - a_0 - a_1 x_i - \cdots - a_i x_i^2\right)\left(-x_i^n\right) \end{cases}$$

Each equation both sides divided by -2 then we arranged to $n+1$ equations:

$$\begin{cases} a_0 N + a_1 \sum x_i + a_2 \sum x_i^2 + \cdots + a_n \sum x_i^n = \sum Y_i \\ a_0 \sum x_i + a_1 \sum x_i^2 + a_2 \sum x_i^3 + \cdots + a_n \sum x_i^{n+1} = \sum x_i Y_i \\ a_0 \sum x_i^2 + a_1 \sum x_i^3 + a_2 \sum x_i^4 + \cdots + a_n \sum x_i^{n+2} = \sum x_i^2 Y_i \\ \qquad\qquad \vdots \\ a_0 \sum x_i^n + a_1 \sum x_i^{n+1} + a_2 \sum x_i^{n+2} + \cdots + a_n \sum x_i^{2n} = \sum x_i^n Y_i \end{cases}$$

If these equations are written into matrix form, the all sum variables are from 1 to N .

$$\begin{pmatrix} N & \sum x_i & \sum x_i^2 & \sum x_i^3 & \cdots & \sum x_i^n \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \cdots & \sum x_i^{n+1} \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \sum x_i^5 & \cdots & \sum x_i^{n+2} \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ \sum x_i^n & \sum x_i^{n+1} & \sum x_i^{n+2} & \sum x_i^{n+3} & \cdots & \sum x_i^{2n} \end{pmatrix} a = \begin{pmatrix} \sum Y_i \\ \sum x_i Y_i \\ \sum x_i^2 Y_i \\ \vdots \\ \sum x_i^n Y_i \end{pmatrix}$$

Coefficient matrix B is called formal matrix of the least squares problem. And the coefficient matrix B have more troublesome to calculating with $\sum$, in order to reduce computation cost and complexity, we can use matrix parallel multiplication to elimination $\sum$ then we can get simplified coefficient matrix B^* [6].

We can find a matrix which is corresponding to coefficient matrix *B* and it is called design matrix [9]. It is form

$$A = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ x_1 & x_2 & x_3 & \cdots & x_N \\ x_1^2 & x_2^2 & x_3^2 & \cdots & x_N^2 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ x_1^n & x_2^n & x_3^n & \cdots & x_N^n \end{pmatrix} \quad A^{\mathrm{T}} = \begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ 1 & x_3 & x_3^2 & \cdots & x_3^n \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & x_N & x_N^2 & \cdots & x_N^n \end{pmatrix}$$

It is easy to be proved that $AA^{\mathrm{T}} = B$, and Ay is the right end vector; here y is column vector which is consisting of Y_i . It can be written as the following form:

$AA^{\mathrm{T}}a = Ay$.

We get evaluation B^* parallel to B and it is worthy for discussion in later. First we suppose there are p processors, according to block thought, the matrix A can divided into p pieces according to the line, then each piece contains $(n+1)/p$ lines; The matrix A^{T} is divided into p pieces according to the column, then each piece contains $(n+1)/p$ columns.

When $(n+1)/p$ is integer, Matrix parallel multiplication steps are as follows：

First step: The first row of the matrix *A* multiplied with the matrix A^{T} in each block column, according to the matrix multiplication, we can get numerical Solution B^* of the first row can be obtained as $B_{1,i}^* \ (i = 1,2,3,\cdots,n+1)$, Then each of the blocks in a row but the first row except of the matrix A respectively multiply with the first column of the matrix A^{T} then the numerical solution B^* of the first column as $B_{i,1}^* \ (i = 2,3,4,\cdots,n+1)$ 。

The second step: The second row of the matrix *A* are multiplied by the matrix A^{T} in each block column except the first column, according to the matrix multiplication we can get B^* of the rest of the second line numerical solution of $B_{2,i}^* \ (i = 2,3,4,\cdots,n+1)$, Then the matrix A of rows in each block(except the first two lines)multiply with the second column of the matrix A^{T} , according to the matrix multiplication we get the B^* of rest of the second column of the numerical solution $B_{i,2}^* \ (i = 3,4,5,\cdots,n+1)$.

Following the steps of sequentially used the matrix *A* of the $3,4,5,\cdots,n+1$ line multiplied with the matrix A^{T} in each block column Then followed by the matrix A rows in each block respectively multiply with the $3,4,5,\cdots,n+1$ column of the matrix A^{T} multiplication, according to the matrix multiplication we finally obtained a simplified new coefficient matrix as B^*

$$B^* = \begin{pmatrix} b^*_{1,1} & b^*_{1,2} & b^*_{1,3} & \cdots & b^*_{1,n+1} \\ b^*_{2,1} & b^*_{2,2} & b^*_{2,3} & \cdots & b^*_{2,n+1} \\ b^*_{3,1} & b^*_{3,2} & b^*_{3,3} & \cdots & b^*_{3,n+1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ b^*_{n+1,1} & b^*_{n+1,2} & b^*_{n+1,3} & \cdots & b^*_{n+1,n+1} \end{pmatrix}$$

Among them, $b^*_{i,j}$ as specific numerical values（$i = 1,2,3,\cdots,n+1$； $j = 1,2,3,\cdots,n+1$）

When $(n+1)/p$ is not integer， $(n+1)/(p-1) = k$ mod q $(q < p)$， and satisfy with the following conditions $(p-1)k + q = n+1$, then the matrix A (and matrix A^{T}) Divided into p blocks by row (or column)， the $1,2,3,\cdots,p-1$ blocks has k row (or column)， and the p blocks has q row (or column). Accordance with calculation method when $(n+1)/p$ is integer, Sequentially used the matrix A of the $1,2,3,\cdots,n+1$ row respectively multiplied with the matrix A^{T} in each block column, The matrix A of rows in each block respectively multiplied with the matrix A^{T} of the $1,2,3,\cdots,n+1$ columns, then we can also be obtained to simplify the coefficient matrix B^* [7].

Throughout matrix multiplication parallel computing we can get the coefficient matrix B^*.

$AA^{\mathrm{T}}a = Ay \Leftrightarrow B^*a = Ay$

The QR parallel processing on B^*, then the B^* will decomposed into orthogonal matrix and an upper triangular matrix plot. $B^* = QR$, So make the general equation solving into triangular equations solving.

Matrix QR decomposition can be used Givens transform to achieve [6]. Standard Givens transform converting by selecting the appropriate plane of rotation the product Q^{T} of the sequence to achieve, each transform eliminate an element under the main diagonal of B^* without destroying the zero elements of the previously formed, So that $Q^{\mathrm{T}}B^* = R$. In order to eliminate B^* of (p,q) non-zero elements, simply make the two lines $p-1$， p of B^* as the following conversion：

$$\begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} b_{p-1,1} & b_{p-1,2} & \cdots & b_{p-1,q} & \cdots & b_{p-1,n} \\ b_{p,1} & b_{p,2} & \cdots & b_{p,q} & \cdots & b_{p,n} \end{pmatrix}$$

Among them, $s = \dfrac{b_{p,q}}{\sqrt{b^2_{p,q} + b^2_{p-1,q}}}$， $c = \dfrac{b_{p-1,q}}{\sqrt{b^2_{p,q} + b^2_{p-1,q}}}$ and this permutation matrix is denoted as $G_{p,q}$， so that the transformation B^* of the role $p-1$， p and eliminate of (p,q) elements.

According to a column-by-column the Givens calculated as following Assume the matrix order number is n， if $r = 1,2,3,\cdots,n-1$， so

$$Q_r = \prod_{i=0}^{n-r-1} G_{n-i,r} \qquad B^*_r = \left(\prod_{i=0}^{r-1} Q_{r-i} \right) B^*$$

Among them, Givens transform $G_{p,q}(1 \le q < p \le n)$ act on $\left(\prod_{i=1}^{p-q-1} G_{p-i,q} \right) B^*_{q-1}$ two lines $p-1$ and p， erasing (p,q) elements. Obviously we having $(n-1)+(n-2)+\cdots+1 = n(n-1)/2$ step-by-step transformation, we can get $R = B^*_{n-1}$， $Q^{\mathrm{T}} = Q_{n-1}\cdots Q_1$ and this was Givens reduction process of matrix B^*.

Actually, the number of (p_k, q_k) , $k = 1,2,\cdots,n(n-1)/2$ in serial Givens reduction set $S = \{(p,q) | 1 \le q < p \le n\}$ order by：

$(n,1),(n-1,1),\cdots,(2,1);(n,2),(n-1,2)$
$,\cdots,(3,2);\cdots;(n,n-1)$

Among them (p_k, q_k) expressed Givens transform $G(p_k, q_k)$ role in $G(p_{k-1}, q_{k-1})\cdots G(p_1, q_1)$. B^* first two lines of p_{k-1} and p_k eliminate the (p_k, q_k) element.

From the above-described serial algorithm can be seen the Givens reduction between zero suppression processes have strong constraints, however, some elements can still parallel zero suppression. Based on the correlation of $G_{p,q}$ between each Givens transform this paper give the QR orthogonal transformation of the fine-grained parallel algorithm. The rearrange the number in set, and divided them into different subsets S_r , $r = 1,2,3,\cdots,l$. Make them meet with $\bigcup S_r = S$, And each of the subset S_r in the rotation transformation do not intersect with each other, but can execute in parallel. Their specific segmentation method as described as follow[8].

Parallel algorithm for elimination the element of (p,q), but not necessarily need to transform the role in the $p-1$, p line, Can act on any of the k line and p line $(k < p)$, only the k line of $(k, j)(1 \le j < p)$ elements is 0. So take advantage of this feature the number (p_k, q_k) , $k = 1,2,3,\cdots,n(n-1)/2$ on the set $S = \{(p,q) | 1 \le q < p \le n\}$ can decomposed as follow ：

First step: In the n , $\lfloor n/2 \rfloor$ lines； $n-1$, $\lfloor n/2 \rfloor - 1$ lines； ⋯； $n - \lfloor n/2 \rfloor + 1$, 1 givens transformation, to eliminate several elements at the following position at the same time：

$(n,1)$, $(n-1,1)$, ⋯ , $(n - \lfloor n/2 \rfloor + 1, 1)$
Obviously, above $\lfloor n/2 \rfloor$ Givens transforms do not intersect with each other.

The second step: In the $\lceil n/2 \rceil$, $\lfloor n/4 \rfloor$ lines;⋯ ;
$\lceil n/2 \rceil - k$, $\lfloor n/4 \rfloor - k$ lines ； ⋯； $\lceil n/2 \rceil - \lfloor n/4 \rfloor + 1$, 1 lines ； n , $\lceil n/2 \rceil + \lfloor n/4 \rfloor$ lines; ⋯ ;$n-k$, $\lceil n/2 \rceil + \lfloor n/4 \rfloor - k$; ⋯； $n - \lfloor n/4 \rfloor + 1$, $\lceil n/2 \rceil + 1$ Givens transformation, to eliminate several elements at the following position at the same time：

$(\lceil n/2 \rceil, 1)(\lceil n/2 \rceil - 1, 1)\cdots$,
$(\lceil n/2 \rceil - \lfloor n/4 \rfloor + 1, 1)(n,2)$, $(n-1,2)$, ⋯ , $(n - \lfloor n/4 \rfloor + 1, 2)$

Other steps empathy, calculated column Givens transform in turn. Take $n=8$ for example; give the process of the Parallel Givens greedy algorithm

```
*
3  *
2  5  *
2  4  7  *
1  3  6  8  *
1  3  5  7  9  *
1  2  4  6  8  10  *
1  2  3  5  7  9  11  *
```

In processing resources unlimited case, above fine-grained parallel algorithms, each processor in each parallel step up to compute one Givens about and $G_{p,q}Q_r$ only acting on the two rows of Q_r , Other elements remain unchanged Two lines $G_{p,q}B^*{}_r$ corresponding to the participation $B^*{}_r$ of the zero suppression） , So every time the host processor is required to pass the current B^* and Q^{T} of two rows of data to sub-processor, each sub-processor is also required to pass the same result of the amount of data to the host processor[9]. So the total number of data transfer $n(n-1)$ times, the amount of data transfer $n(n-1)\cdot 4n$. Obviously, these fine-grained parallel algorithms for data transfer a large amount of $O(n^3)$, unsuitable for communication delay large cluster parallel computing systems, we should appropriate to improve the parallel algorithm the matrix QR decomposition of coarse-grained parallel algorithm suitable for cluster systems.

Network parallel computing environments to achieve parallel algorithm requires minimizing the communication overhead, and using the medium-grained task parallelism. To this end, according to the inherent parallelism of matrix QR factorization, we redraw its task. Sub-blocks of the original matrix B^*： Its rows and columns are divided into *K* equal parts, and these dividing lines intersect, the elements under the diagonal are divided into triangular sub-blocks or rectangular sub-blocks, and then one of the rectangular sub-blocks in a diagonal line from left to right is divided into two triangular sub-block and the diagonal elements are contained in the in the lower right corner of the rectangular sub-blocks the sub-block. So we obtained K^2 triangular sub-blocks, The matrix B^* of the QR decomposition process actually make the respective triangular sub-blocks of the elements into zero suppression, therefore each of which a triangular sub-block corresponding to the zero suppression task may be seen as a sub-task[10].

Obviously, we obtained K^2 triangular sub-blocks for B^* . Hutchison subtasks i from left to right in the j row block triangular corresponding to sub block $T^{i,j}$ $(1\le i\le K, 1\le j\le 2i-1)$.

According to the definition of Givens about, in the QR transformation process of the matrix B^* Peer-element elimination process should column sequentially $1,2,\cdots,n-1$ and also asked if the elimination of the (p,q) element, then asked participate in transformation of the first $q-1$ element is 0 but q element was not. Therefore, the above task division obtained sub-tasks, The Givens transformations between the various sub-tasks and its internal elements should meet the following requirements：

sub-tasks $T^{i,j}$ $(j \bmod 2=1)$， when $T^{i,j-1}$ has been completed, it can only performed And if its corresponding sub-block where the rectangular sub-block is viewed as the QR transform matrix, Givens about of its internal elements is equivalent to the matrix, simply put the transformation matrix of the corresponding sub-blocks where the entire row block.

sub-tasks $T^{i,j}\left(j \bmod 2 = 0\right)$， when $T^{i,j-1}$ has been completed, it can only performed And its elements of Givens transformation requires the use of the *k* line block data, this need the line block to meet the subtasks $T^{k,j-1}$ have been executed, that its first triangular sub-blocks have been zero suppression.

Obviously, put matrix B^* obtained into the division of sub-tasks, in each parallel steps, while satisfying with 1), 2) and has a plurality of parallel execution. This is the block parallel cluster system Givens reduction The matrix of rows and columns respectively *K* equal parts, Matrix diagonal Givens about elements are divided into K^2 triangular sub-blocks, Each triangular sub-block corresponds to a sub-task mark in accordance with the method described above In each of the parallel step, find all satisfy the conditions 1), 2) sub-tasks and assigned to each sub-processor parallel zero suppression.

According to the above-mentioned tasks, We can obtain the precursors of task graph, Scheduling algorithm is applied to the task pool can be obtained by the network parallel computing the matrix QR factorization environment sub block parallel algorithms. Actually, the scheduling procedure is a simple, The Intuitive approach is these subtasks to be seen as an element in the matrix, and applied the greedy algorithm on it little improvement and obtained each sub-task scheduling order.

The paper give the least squares curve fitting problem parallelization, If the known data group is expanded to a three-dimensional data set put $l_i = (x_i, y_i)^T$ replace x_i， then we will get the least squares surface fitting problem of parallel computing.

LEAST-SQUARES ALGORITHM IS APPLIED PARALLEL SURFACE FITTING

Based on Least squares parallel algorithm we solve the design of the surface fitting image surface fitting. To solve the image surface fitting problem based on least squares parallel algorithm design surface fitting. This method makes full use of the parallel features of the method of least squares, and it is able to maintain the image surface fitting results in the higher fitting accuracy premise so this fitting results facilitate has the efficient calculation of the surface characteristics of the depth image. By use the real depth image numerical experiments, the results illustrate is the effectiveness of the method.

There are two types of methods to calculate the curvature of the image surface, One is the use of differential geometry formulas calculated directly, since the direct calculation of the error is large, that has been rarely used. The other is Surface Fitting but this method within the local region on the image surface fitting, this method for some depth image complex area can not be well fitted in a short period of time.

To 1000Mbps Ethernet cluster simulation algorithm, the program uses the master-slave mode structure use the MPI + FORTRAN language and its implementation. 4-node cluster system design parallels least squares surface fitting algorithm simulation, assuming solving the QR decomposition is divided into four sub-tasks.[11]

To illustrate the effectiveness of the algorithm, experimental use of an image in the depth image library of the University of South Florida, as shown in Fig.1. In calculation process, all known data points either as a data point can be used as computing nodes, however, in the actual calculation process, we not used all the data points as compute nodes , but the depth of the image data, to trap a take one or the interval of two to take a as compute nodes. Such as compute nodes spaced points, the shape function through the compute nodes, and reuse all the data points of the depth image obtained after fitting the image surface. This not only can reduce the amount of computation, but can effectively reduce the random noise of the depth of the image itself. Compare via serial (see Fig.2) and parallel (Fig.3) algorithm fitting results visible, the fitting surfaces results are consistent But the use of the design of parallel algorithms for the 9.62s, far lower than that of the serial algorithm 27.68s so it has higher speedup and efficiency.

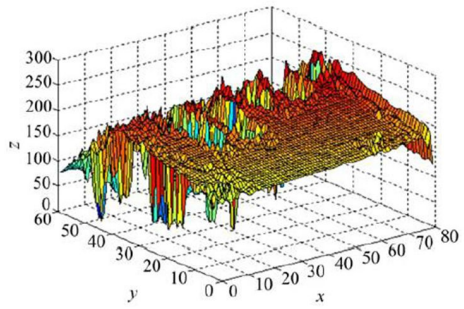

Figure 1 Scatterplot

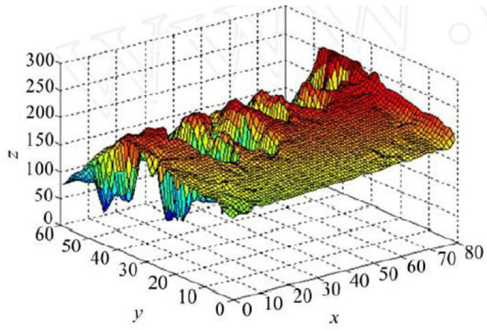

Figure 2 Serial fitting graphics (computing time 16.57s)

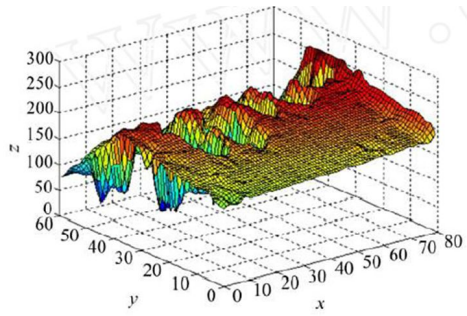

Figure 3 Parallel fitting graphics (computing time 7.36s)

Acknowledgements

This work was supported by National scientific and technological support projects (No.2012BAE09B00), the National Natural Science Foundation of China (No.51274270) and the National Natural Science Foundation of Hebei Province (No. E2013209215).

REFERENCES

[1]Curtis F. Gerald etc, "Numerical Analysis (version 7of the original book)", *beijing: Machinery Industry Press,* **2006**,pp.6～7.

[2]Qinsheng Liu, "The calculation method of the least squares", *Beijing: Beijing University of Technology Press*, **1989**,pp.12～16.

[3]Jiala Duo etc, "Summary of parallel computing", *Beijing: Electronic Industry Press*, **2004**,pp.9～15.

[4]Guoliang Chen, "Parallel Computing - structure algorithm programming(revised version)",*Beijing: Higher Education Press*, **2003**,pp.3～6.

[5]Xavier (USA) Garr, "Introduction to Algorithms of Parallel", *Beijing: Machinery Industry Press,***2004**,pp.4～5.

[6]Shixin Sun etc, "Parallel algorithm and its application", *Beijing: Machinery Industry Press,* **2005**,pp.61～66.

[7]John H,MathewsKurtis D.Fink,"Numerical Methods Using MATLAB Fourth Edition", *Beijing: Electronic Industry Press*, **2005**, pp.10～15.
[8G.H.Golub, D. F. Van Loan, "Matrix calculation", *DaLian: Dalian University of Technology Press*. **1988**,pp.12～18.
[9]P. G. Ciarlet, "Matrix numerical analysis and optimization", *Beijing: Higher Education Press*, **1990**, pp.18～21.
[10]Goodrich, " Algorithm Analysis and Design", *Beijing: People's Posts and Telecommunications Press,*2006,pp.23～24.
[11]Li Xie, Peng Wang, "Packaging Engineering", *Beijing: Fifty-nine Institute of China Ordnance Industry*,**2009**,pp.13～15.