IMPLEMENTATION OF SPARSE MATRIX IN CHOLESKY DECOMPOSITION TO SOLVE NORMAL EQUATION

Halim Setan and Muhammad Asyran

Survey Engineering Research Group (SERG), Department of Geomatic Engineering
Faculty of Geoinformation Science and Engineering, Universiti Teknologi Malaysia
E-Mail: halim@fksg.utm.my, asyran_utm@yahoo.com

Abstract

Practical measurement schemes require redundant observations for quality control and errors checking. This led to inconsistent solution where every subset (minimum required data) gives different results. Least Square Estimation (LSE) is a method to provide a unique solution (of the normal equation) from redundant observations by minimizing the sum of squares of the residuals. Analysis of LSE also provide estimate quality of parameters, observations and residuals, assessment of network's reliability and precision, detection of gross errors etc. Many methods can be applied to solve normal equation, e.g. Gauss-Doolittle, Gauss-Jordan Elimination, Singular Value Decomposition, Iterative Jacoby etc. Cholesky Decomposition is an efficient method to solve normal equation with positive definite and symmetric coefficient matrix. It is also capable of detecting weak condition[1] of the system. Solving large normal equation will require a lot of times and computer memory. Implementation of sparse matrix in Cholesky Decomposition will speed up the execution times and minimize the memory usage by exploiting the zeros and symmetrical of coefficient matrix. This paper discusses the procedures and benefits of implementing sparse matrix in Cholesky Decomposition. Some preliminary results are also included.

1.0 INTRODUCTION

Least Square Estimation (LSE) is required in many survey and engineering applications to provide a unique solution from redundant data, estimate the quality of parameters, observations and residuals, assessment of network's reliability and precision, detection of gross errors etc. LSE can be implemented either using observation, condition or combine equations. According to Abdul Wahid & Halim (2001), observation and condition equation are specific cases of combine equation. The mathematical model for observation equation can be defined as observation is a function of parameters, condition equation form the mathematical model with hypothesis that function of adjusted observations will satisfy certain condition (usually zero), while mathematical model for combine equation is a function of adjusted observations and parameters will satisfied certain condition (usually zeros). A complete discussion of these techniques can be found in Wolf and Ghilani (1997). This paper focused on the implementation of LSE using observation equation.

According to Caspary (1987), 3 method of performing LSE are minimum constraint, minimum trace and partial minimum trace. LSE by minimum constraint require smallest possible number of coordinates to be datum definition (equal to number of datum defect[2]), minimum trace use all network stations as datum definition and partial minimum trace require number of datum definition to be more than datum defect. General equations for LSE using observation equation are shown in Equations (1) - (5).

$$\overline{X} = N^{-1}U \tag{1}$$

$$Q_{\overline{X}} = \hat{\sigma}^2 \cdot N^{-1} \tag{2}$$

Where:-

[1] Small change in coefficient matrix will cause drastic change of the inverse matrix
[2] Datum defect cause the impossibility of inversing matrix *N*

$$N = A^T PA \tag{3}$$

$$U = A^T PL \tag{4}$$

$$\hat{\sigma}^2 = \frac{V^T PV}{n-u} \tag{5}$$

Numerical methods for solving normal equations [Equation (1)] fall into 2 general classes, iterative and direct. A typical iterative method involves the initial selection of an approximation $x^{(1)}$ to x, and the determination of a sequence $x^{(2)}$, $x^{(3)}, \ldots$ such that $\lim_{i \to \infty} x^{(i)} = x$ (Geoge and Liu, 1981). Theoretically an iterative method will require performing an infinite number of arithmetic operations in order to obtain x, but in practice the iteration will already be stopped after the current approximation is acceptably close to x. In the absence of rounding errors, direct methods provide the solution after a finite number of arithmetic operations have been performed. This paper focused on the solution of normal equation using direct method (Cholesky Decomposition).

2.0 SOLUTION OF NORMAL EQUATION

Solution of normal equation in LSE involve large and sparse matrix. Many direct methods can be applied to solve normal equation, e.g. Gauss-Doolittle, Gauss-Jordan Elimination, Singular Value Decomposition etc. Cholesky Decomposition is an efficient method to solve normal equation with symmetric positive definite of coefficient matrix, *N* (Press et al., 1986). Implementations of sparse matrix in Cholesky Decomposition are necessary to speed up the computation, memory management and for numerical stability. This is done based on the facts that *N* matrix is symmetric, large but sparse[3]. Implementation of Cholesky Decomposition involves 2 main steps, factorization and solving triangular systems.

2.1 Factorization

According to George & Liu (1981), a symmetric positive definite matrix, *N* with dimension of $n \times n$ will have a unique triangular factorization LL^T, where *L* is a lower triangular matrix with positive diagonal entries. The purpose of factorization is to obtain the *L* matrix from *N* as represented in Equation (6). There are 2 ways in computing *L* matrix either using outer product form or bordering method.

$$N = L \cdot L^T \tag{6}$$

$$\underset{\boldsymbol{N}}{\begin{bmatrix} N_{11} & & & & & \\ N_{21} & N_{22} & & & & \\ N_{31} & N_{32} & N_{33} & & & \\ \vdots & & & \ddots & & \\ \vdots & & & & \ddots & \\ N_{n1} & N_{n2} & N_{n3} & \cdots & \cdots & N_{nn} \end{bmatrix}} = \underset{\boldsymbol{L}}{\begin{bmatrix} L_{11} & 0 & \cdots & \cdots & \cdots & 0 \\ L_{21} & L_{22} & \cdots & \cdots & \cdots & 0 \\ L_{31} & L_{32} & L_{33} & \cdots & \cdots & 0 \\ \vdots & & & & & \\ \vdots & & & & & \\ L_{n1} & L_{n2} & L_{n3} & \cdots & \cdots & L_{nn} \end{bmatrix}} \times \underset{\boldsymbol{L^T}}{\begin{bmatrix} L_{11} & L_{21} & L_{31} & \cdots & \cdots & L_{n1} \\ 0 & L_{22} & L_{32} & \cdots & \cdots & L_{n2} \\ \vdots & & L_{33} & \cdots & \cdots & L_{n3} \\ \vdots & & & & & \\ \vdots & & & & & \\ 0 & 0 & 0 & \cdots & \cdots & L_{nn} \end{bmatrix}}$$

[3] Contains a lot of zero values

2.1.1 Outer Product Form

Computation of lower triangular matrix, L using outer product form can be derived as follows (George & Liu, 1981):-

$$
\begin{aligned}
N &= \begin{pmatrix} d & v^T \\ v & \overline{H} \end{pmatrix} \\
&= \begin{pmatrix} \sqrt{d} & 0 \\ \frac{v}{\sqrt{d}} & I_{N-1} \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & H \end{pmatrix} \cdot \begin{pmatrix} \sqrt{d} & \frac{v^T}{\sqrt{d}} \\ 0 & I_{N-1} \end{pmatrix} \\
&= \begin{pmatrix} \sqrt{d} & 0 \\ \frac{v}{\sqrt{d}} & I_{N-1} \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & L_H \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & L_H^T \end{pmatrix} \cdot \begin{pmatrix} \sqrt{d} & \frac{v^T}{\sqrt{d}} \\ 0 & I_{N-1} \end{pmatrix} \\
&= \begin{pmatrix} \sqrt{d} & 0 \\ \frac{v}{\sqrt{d}} & L_H \end{pmatrix} \cdot \begin{pmatrix} \sqrt{d} & \frac{v^T}{\sqrt{d}} \\ 0 & L_H^T \end{pmatrix} \\
&= L \cdot L^T
\end{aligned} \tag{7}
$$

Where:-

$N =$ Symmetric positive definite matrix of order n

$d =$ Positive scalar

$v =$ An $(n-1)\times 1$ sub matrix

$\overline{H} =$ An $(n-1)\times(n-1)$ sub matrix

$H = \overline{H} - \frac{vv^T}{d}$

The constructive proof of Equation (7) suggests a computational scheme to determine the factor L. It is so-called outer product form of the algorithm. The scheme can be described step by step in matrix as shown in Equations (8) – (11):-

$$
\begin{aligned}
N = N_0 = H_0 &= \begin{pmatrix} d_1 & v_1^T \\ v_1 & \overline{H}_1 \end{pmatrix} \\
&= \begin{pmatrix} \sqrt{d_1} & 0 \\ \frac{v_1}{\sqrt{d_1}} & I_{N-1} \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & \overline{H}_1 - \frac{v_1 v_1^T}{d_1} \end{pmatrix} \cdot \begin{pmatrix} \sqrt{d_1} & \frac{v_1^T}{\sqrt{d_1}} \\ 0 & I_{N-1} \end{pmatrix} \\
&= L_1 \cdot \begin{pmatrix} 1 & 0 \\ 0 & H_1 \end{pmatrix} \cdot L_1^T \\
&= L_1 N_1 L_1^T
\end{aligned} \tag{8}
$$

$$
\begin{aligned}
N_1 &= \begin{pmatrix} 1 & 0 \\ 0 & H_1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & d_2 & v_2^T \\ 0 & v_2 & \overline{H}_2 \end{pmatrix} \\
&= \begin{pmatrix} 1 & & 0 \\ 0 & \sqrt{d_2} & \\ 0 & \dfrac{v_2}{\sqrt{d_2}} & I_{N-2} \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \overline{H}_2 - \dfrac{v_2 v_2^T}{d_2} \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ & \sqrt{d_2} & \dfrac{v_2^T}{\sqrt{d_2}} \\ 0 & & I_{N-2} \end{pmatrix} \\
&= L_2 N_2 L_2^T \\
&\quad \vdots \\
N_{n-1} &= L_n I_n L_n^T
\end{aligned} \tag{9}
$$

After n steps of algorithm, the symmetric positive definite matrix, *N* will be completely decomposed into lower triangular matrix, *L*.

$$
N = L_1 L_2 \cdots L_n L_n^T \cdots L_2^T L_1^T = LL^T \tag{10}
$$

Where it can be shown that:-

$$
L = L_1 + L_2 + \cdots + L_n - (n-1) I_n \tag{11}
$$

In this scheme, the columns of *L* are computed one by one. Each step will involve the modification of the sub matrix $\overline{H}_i$ by the outer product $v_i v_i^T / d_i$ to give H_i, which is simply the sub matrix remaining to be factored. The access to the components of A during the factorization is depicted as in Figure 1.

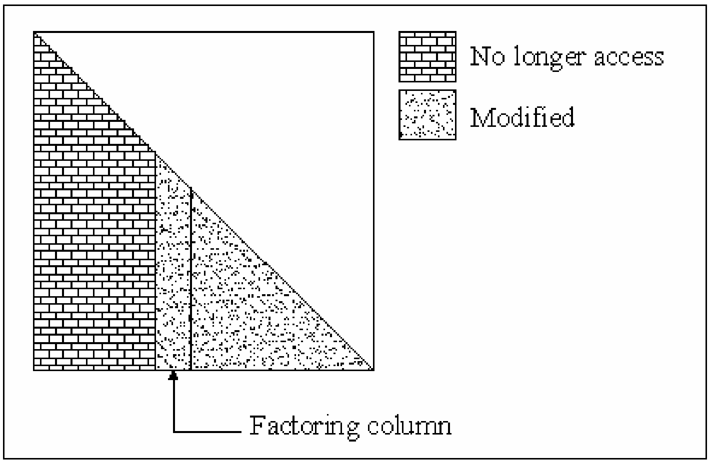

Figure 1: Factorization Using Outer Product Form

2.1.2 Bordering Method

An alternative formulation of the factorization is the bordering method. The algorithm can be derived directly by equating the elements of *N* to the corresponding elements of the product LL^T (Wolf & Ghilani, 1997).

$$L_{jj} = \left(N_{jj} - \sum_{k=1}^{j-1} L_{jk} \cdot L_{jk} \right)^{\frac{1}{2}} \tag{12}$$

Equation (12) show that the rows of *L* are computed one at the time. The part of the matrix remaining to be factored is not accessed until the corresponding part of L is to be computed. The sequence of computations can be depicted as follows.

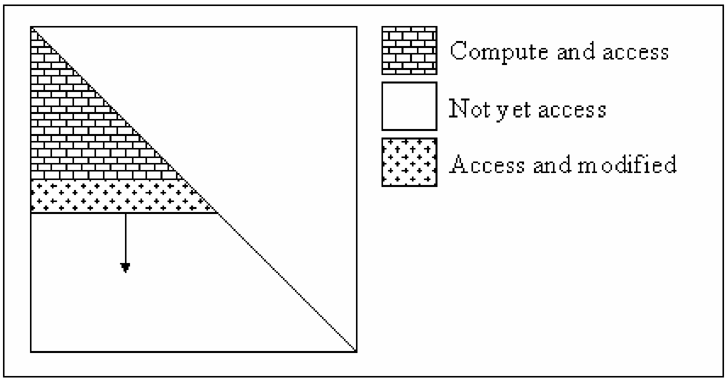

Figure 2: Factorization Using Bordering Method
(Computing Diagonal Elements)

The final scheme for computing the components of *L* is the inner product form of the algorithm. The columns of *L* are computed one by one like the outer product version of algorithm, but the part of the matrix remaining to be factored is not accessed during the scheme.

For $j = 1, 2, \ldots, n$

Compute $L_{jj} = \left(N_{jj} - \sum_{k=1}^{j-1} L_{jk} \cdot L_{jk} \right)^{\frac{1}{2}}$

For $i = j+1,\ j+2, \ldots, n$ (13)

Compute $L_{ij} = \left(N_{jj} - \sum_{k=1}^{j-1} L_{ik} \cdot L_{jk} \right) / L_{jj}$

The access to the components of *L* for both equations can be represented as in Figure 3.

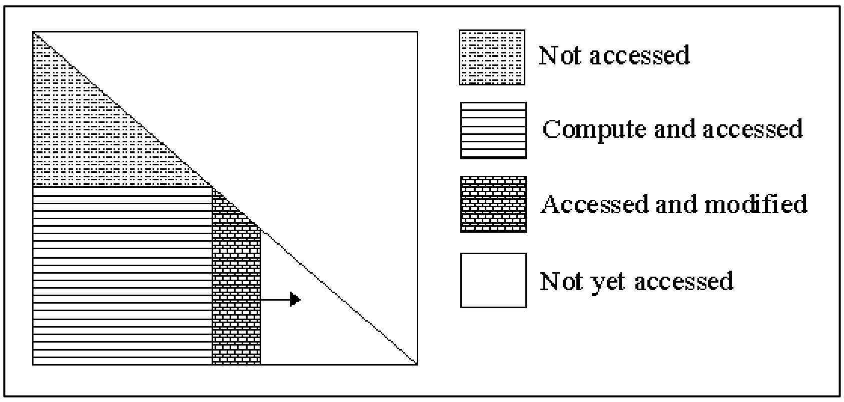

Figure 3: Factorization Using Bordering Method (Combine Equations)

2.2 Solving Triangular Systems

In Cholesky Decomposition, solving triangular system involves forward and back solution. During forward solution, the vector *F* is calculated as shown in Equations (14) – (16):-

$$N \cdot X = U \tag{14}$$
$$LL^T \cdot X = U \tag{15}$$
$$L \cdot F = U \tag{16}$$

$$\begin{array}{ccccc} \mathbf{L} & & \mathbf{F} & & \mathbf{U} \\ \begin{bmatrix} L_{11} & 0 & \cdots & \cdots & \cdots & 0 \\ L_{21} & L_{22} & \cdots & \cdots & \cdots & 0 \\ L_{31} & L_{32} & L_{33} & \cdots & \cdots & 0 \\ \vdots & & & & & \\ \vdots & & & & & \\ L_{n1} & L_{n2} & L_{n3} & \cdots & \cdots & L_{nn} \end{bmatrix} & \times & \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ \vdots \\ \vdots \\ F_n \end{bmatrix} & = & \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ \vdots \\ \vdots \\ U_n \end{bmatrix} \end{array}$$

There are 2 common ways of solving the systems, which differ in the order in which the operations are performed. The first one involves the use of inner-product and the defining equations are given by:-

For $i = 1, 2, \ldots, n$

$$F_i = \left(U_i - \sum_{k=1}^{i-1} L_{i,k} F_k \right) / L_{i,i} \tag{17}$$

The sequence of computation is depicted by the following diagram:-

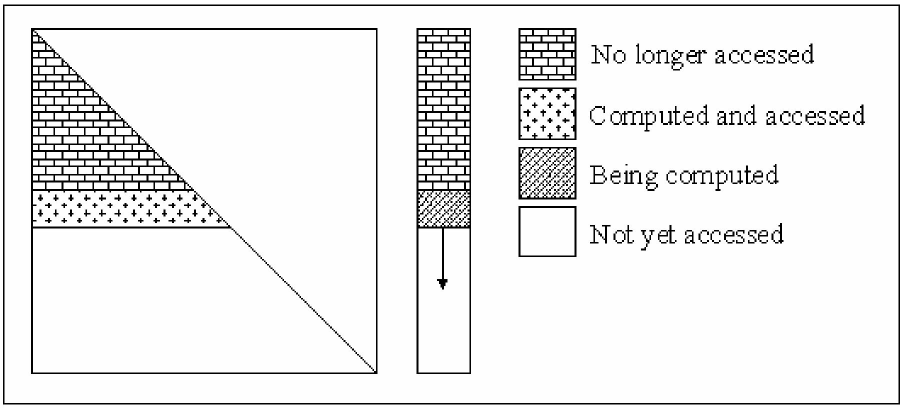

Figure 4: Solving Triangular Systems (Involve Inner-Product)

The second method uses the matrix components of *L* in the same way as the outer-product version of the factorization. The defining equations are as follows:-

$$\begin{aligned}&\text{For } i = 1, 2, \ldots, n \\ &\quad F_i = U_i / L_{i,i} \\ &\begin{pmatrix} U_{i+1} \\ \vdots \\ U_n \end{pmatrix} \leftarrow \begin{pmatrix} U_{i+1} \\ \vdots \\ U_n \end{pmatrix} \leftarrow F_i \begin{pmatrix} L_{i+1,i} \\ \vdots \\ L_{n,i} \end{pmatrix}\end{aligned} \tag{18}$$

This scheme lends itself to exploiting sparsity in the solution *F*. If U_i turns out to be zero at the beginning of the *i*-th step, F_i is zero and the entire step can be skipped. The accessed to the components of the system is shown as follows.

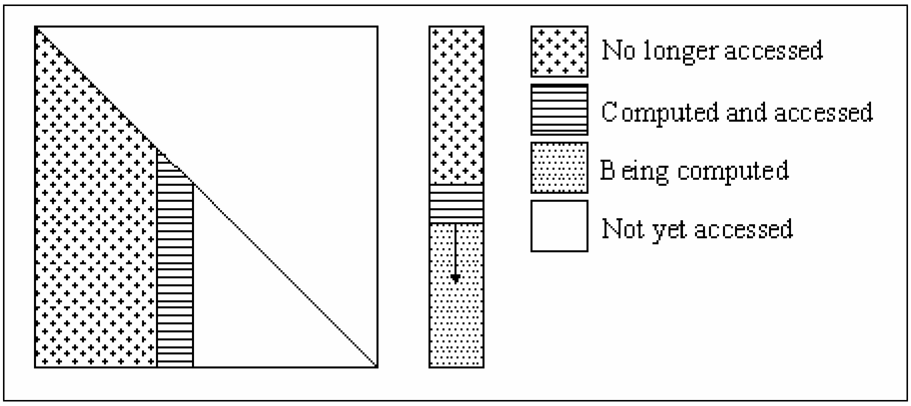

Figure 5: Solving Triangular Systems (Exploiting Sparsity)
Back solution is the final step to get solution of normal equation base on *F* values.
The entire rocesses for computing forward solution are repeated with upper triangular elements.

$$L^T X = F \tag{20}$$

$$\begin{array}{ccccc} \boldsymbol{L} & & \boldsymbol{X} & & \boldsymbol{F} \\ \begin{bmatrix} L_{11} & L_{21} & L_{31} & \cdots & \cdots & L_{n1} \\ 0 & L_{22} & L_{32} & \cdots & \cdots & L_{n2} \\ \vdots & & L_{33} & \cdots & \cdots & L_{n3} \\ \vdots & & & & & \\ \vdots & & & & & \\ 0 & 0 & 0 & \cdots & \cdots & L_{nn} \end{bmatrix} & \times & \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ \vdots \\ \vdots \\ X_n \end{bmatrix} & = & \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ \vdots \\ \vdots \\ F_n \end{bmatrix} \end{array}$$

3.0 COMPUTING THE INVERSE MATRIX

Computation of inverse matrix is required to calculate the cofactor matrix[4] [Equation (2)]. Using Cholesky Decomposition, the inverse of *N* matrix can be shown as follows.

[4] Represent the quality of results

$$\begin{aligned} N^{-1} &= \left(L^T L\right)^{-1} \\ &= L^{-1}\left(L^T\right)^{-1} \\ &= L^{-1}\left(L^{-1}\right)^T \end{aligned} \tag{21}$$

Where L^{-1} is also a lower triangular matrix.

$$L^{-1} = \begin{bmatrix} d_{11} & 0 & 0 \\ d_{21} & d_{22} & 0 \\ d_{31} & d_{32} & d_{33} \end{bmatrix} \tag{22}$$

The next computation are:-

$$LL^{-1} = I \tag{23}$$

$$\begin{bmatrix} l_{11} & & \\ l_{21} & l_{22} & \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \cdot \begin{bmatrix} d_{11} & 0 & 0 \\ d_{21} & d_{22} & 0 \\ d_{31} & d_{32} & d_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Elements for L^{-1} can be calculated using Equation (24).

$$d_{ij} = \frac{1}{l_{ij}} \sum_{k=i+1}^{j} - l_{ik} \cdot d_{kj} \tag{24}$$

Based on Equation (24), solution of N^{-1} can be calculated as below.

$$\begin{bmatrix} {n_{11}}^{-1} & {n_{12}}^{-1} & {n_{13}}^{-1} \\ & {n_{22}}^{-1} & {n_{23}}^{-1} \\ & & {n_{33}}^{-1} \end{bmatrix} = \begin{bmatrix} d_{11} & 0 & 0 \\ d_{21} & d_{22} & 0 \\ d_{31} & d_{32} & d_{33} \end{bmatrix} \cdot \begin{bmatrix} d_{11} & d_{12} & d_{13} \\ 0 & d_{22} & d_{23} \\ 0 & 0 & d_{33} \end{bmatrix}$$

$${N_{ij}}^{-1} = \sum_{k=1}^{n} d_{ik} \cdot d_{kj} \tag{25}$$

4.0 ADVANTAGES OF USING CHOLESKY DECOMPOSITION

Solution of normal equation and inversing using Cholesky Decomposition has advantages as follows (Press et al., 1986; Wolf & Ghilani, 1997):-

i. Faster computation times can be obtained due to solution of equation with only triangular elements.
ii. Capable of detecting ill condition or singular matrix by checking the diagonal elements of lower triangular matrix, *L*.-

$$L_{ii} = \sqrt{n_{ii} - \left({L_{11}}^2 + {L_{11}}^2 + \ldots {L_{i-1,i}}^2\right)} \tag{26}$$

If the matrix is singular or ill condition occur, diagonal elements for L_{ii} will be zeros or near to zeros.

iii. Cholesky Decomposition also capable on detecting lost of efficiency by calculating the Googe number.

$$g_i = \frac{L_{ii}^{\ 2}}{n_{ii}} \tag{27}$$

Lost of efficiency is claim to be happen if the Googe number is not within range 0 and 1.

5.0 PRELIMINARY RESULTS

An LSE program is currently being developed at SERG based on the theory discussed. The module comes with the packages of deformation detection and graphic visualization. The program allows users to do adjustment of terrestrial (1D, 2D and 3D) and GPS data, link to STARNET (commercial LSE program from USA), S-Transformation to get results from other datum definition, feasibility of detecting gross errors and pre-analysis for network planning. The program will have extra features compared to the previous version (GPSAD2000) as listed in Table 2 (Bong, 2000; Halim et.al, 2004). Figure 6 shows interface of LSE program that is currently being developed.

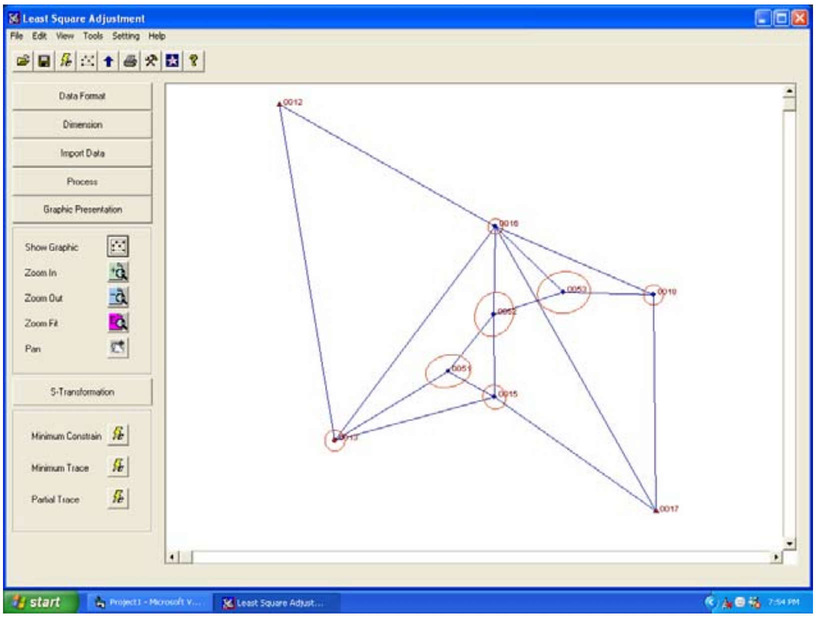

Figure 6: Interface of LSE Program

Table 1: Comparison of LSE Program vs GPSAD2000 (Previous Version)

FUNCTIONS	GPSAD2000	NEW SOFTWARE
- Dimensions	3D	1D, 2D and 3D
- Computer Optimizations	Normal approach	Using single array and optimize with Cholesky Decomposition, sparse matrix, Gauss-Doolittle etc
- Data Format	Local (in-house software)	Local (in-house software)
- S-Transformation	Adjustment in minimum constraint, minimum trace and partial minimum trace	S-Transformation to minimum constraint, minimum trace and partial minimum trace
- Linking to StarNet	None	Allow users to choose either LSE in local software or StarNet
- Configurations menu	Normal configuration	More features
- Graphical Presentation	Show in graphic menu	Graphic in LSE module
- Help files	None	Guide users to use the program
- Expot drawing	Format **.dxf*, **.bmp* dan **.jpg*	Format **.dxf*, **.bmp* dan **.jpg*

The program gives four outputs: result of computation file (**.txt*), deformation file (**.def*), AutoCAD script file (**.scr*) and graphic view. Users can decide either to save graphic in raster (**.gif*) or vector (**.dxf*) format. The program also provide some configurations e.g. unit to use, required results and etc as shown in Figure 8.

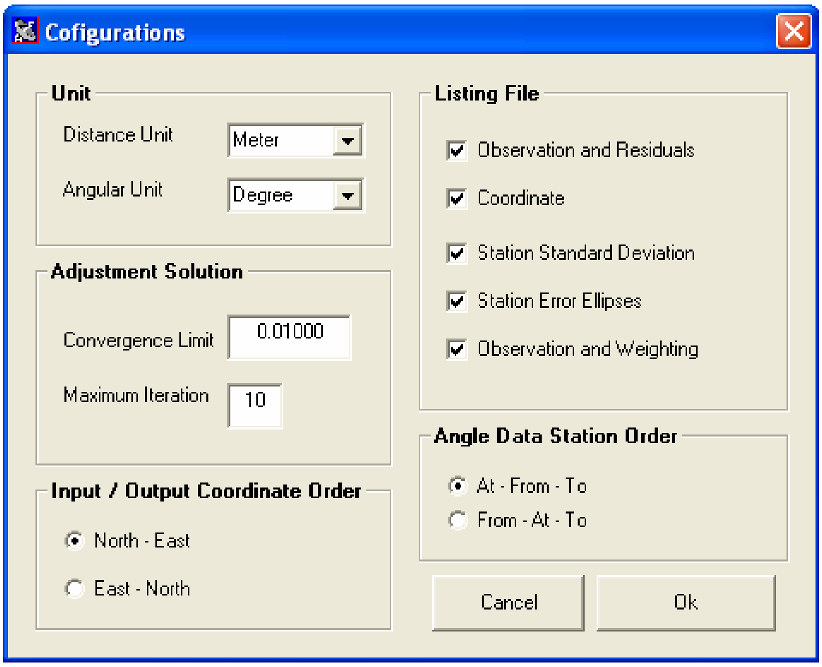

Figure 7: Configurations Menu

6.0 CONCLUSION

Least Square Estimation (LSE) has been gaining popularity rapidly as the method used for analyzing and adjusting surveying data. LSE is the most rigorous adjustment procedure available recently that based on the mathematical theory of probability. Computing normal equation in LSE involves large and sparse matrix. Several methods can be implemented (iterative and direct) to solve normal equation. Cholesky Decomposition is an option and efficient to solve symmetric positive definite of coefficient matrix, *N*. Implementation of sparse matrix in Cholesky Decomposition will reduce the requirement of computer memory, processing times and capable on detecting weak condition of the systems.

REFERENCES

1. Abdul Wahid Idris & Halim Setan (2001). Pelarasan Ukur. Kuala Lumpur: Dewan Bahasa dan Pustaka

2. Bong, Chin Nyet (2000). GPSAD2000 – Sistem Perisian Untuk Pelarasan Vektor GPS, Pengesanan Deformasi dan Analisis Visualisasi. Tesis Sarjana Sains (Ukur Tanah), Universiti Teknologi Malaysia.

3. Cham, Kok Yeong (1997). Pembangunan Perisian Untuk Persembahan Grafik Ukur Deformasi. Tesis Sarjana Muda Ukur Tanah, Universiti Teknologi Malaysia.

4. George, A. & J.W.-H. Liu (1981). Computer Solution of Large Sparse Positive Definite Systems. Prentice Hall, Upper Saddle River, New Jersey.

5. Halim Setan (1995). Functional and Stochastic Models for Geometrical Detection of Deformation In Engineering: A Practical Approach. City University: Ph.D Thesis.

6. Halim Setan, Muhammad Asyran, Ernyza Endot & Mohd Sharuddin Ibrahim (2004). Pembaharuan Perisian Gpsad2000 Untuk Pakej Lengkap Analisis Deformasi. Survey Engineering Research Group, Jabatan Kejuruteraan Geomatik, Fakulti Kejuruteraan Dan Sains Geoinformasi, Universiti Teknologi Malaysia.

7. Heng, Aik Chuan (1997). Prosidur Pelarasan Data GPS Untuk Ukur Deformasi. Tesis Sarjana Muda Ukur Tanah, Universiti Teknologi Malaysia.

8. L. Volgyesi (1974). Matrix-Orthogonalization Method In Adjustment. Department of Geodesy, Institude for Geodesy, Surveying and Photogrammetry, Technical University, Budapest.

9. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P. (1986). Numerical Recipes in Fotran 77: The Art of Scientific Computing. Cambridge University Press.

10. Ranjit Singh s/o Dharam Singh (1997). Analisis Deformasi 2-D Dengan Kaedah Ujian Kongruensi. Tesis Sarjana Muda Ukur Tanah, Universiti Teknologi Malaysia.

11. Ranjit Singh s/o Dharam Singh (1999). Pelarasan dan Analisis Jaringan Pengawasan Untuk Pengesanan Deformasi Secara Geometri. Tesis Sarjana Ukur Tanah, Universiti Teknologi Malaysia.

12. Wolf, P.R & Ghilani, C.D. (1997). Adjustment Computations, Statistics and Least Squares in Surveying and GIS. Department of Civil and Environmental Engineering, University of Wisconsin-Madison. Surveying Program, Pennsylvania State University.