Residual statics analysis by LSQR

Yan Yan, Zhengsheng Yao, Gary F. Margrave and R. James Brown

ABSTRACT

Residual statics corrections can be formulated as a linear inverse problem. Usually, solving this problem involves a large ill-posed matrix inverse calculation. Therefore, a fast algorithm that can handle the ill-posed problem is needed in practice. In this paper, the least-squares QR factorization (LSQR), based on Lanczos and QR factorization, is presented. This method works on rectangular matrix and is a rapidly converging and stable algorithm for the ill-posed problem.

INTRODUCTION

Residual static corrections are uniform time shifts that are applied to trace to compensate for time delays in the highly variable near-surface weathering zone. These delays occur near both sources and receivers (Wiggins et al., 1976). See Figure1.

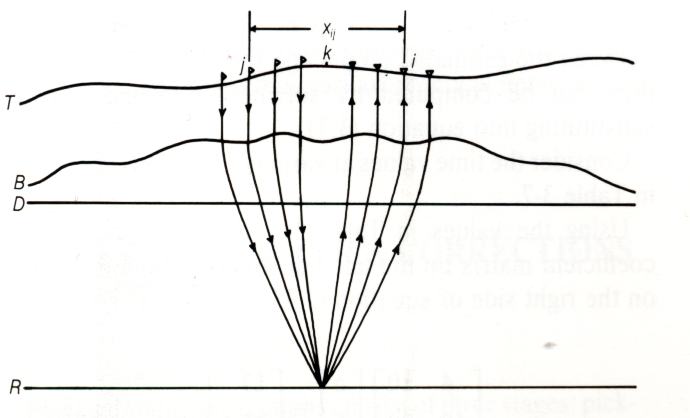

Figure 1. Surface-consistent statics model (after Yilmaz, 1987,p.201).

If we assume that the statics corrections are surface- and subsurface-consistent, we can sort out the overlapping contributions to the total traveltime for a particular reflector. The residual shot static is the traveltime from a shot through the near-surface to some datum level. By surface consistency we mean that the same shot static applies to the shot at a particular location, independent of the various receiver locations. Similarly, the receiver statics are assumed to be the same for all shots received at a given location. The structural term pertains to a location midway between the source and receiver, and is the two-way vertical-path traveltime between the datum and a deep reflector. It is assumed to be subsurface-consistent, i.e., independent of the source-receiver offset, and the final term is the RNMO that exists after initial inexact NMO compensation has been applied. It is also assumed to be subsurface-consistent (Wiggins et al., 1976). So, with the assumption of surface and subsurface consistency, the total traveltime for a given reflection event on a particular trace can be written as the summation of four terms:

$$S_i + R_j + G_k + M_k X_{ij}^2 = T_{ij}, \tag{1}$$

where Si is the traveltime from the source to the datum plane at the ith source position, Rj is the traveltime from the receiver to the datum plane at the jth receiver position, Gk is the normal-incidence two-way traveltime from the datum plane to a subsurface reflector at the kth CDP position; (structural term), Mk is the time-averaged RNMO coefficient at the kth CDP position, Xij is the distance between the jth receiver and the ith shot, Tij is the total traveltime for the ij trace, j is the receiver position index, i is the shot position index, k = i+j-1 is the CDP-position index.

The measured reflection time of a given event on each seismic trace in a profile provides an equation of the above type: one equation for each source-receiver combination along the line. The parameters S_i, R_j, G_k, and M_k are unknowns which, when determined, form the solution of the statics problem. Therefore, the residual statics problem associated with observations in CDP data can be cast in the form of a matrix equation that relates the source and receiver statics, residual normal moveout, and structure to the observed time deviation among the traces.

In order to apply inversion theory, we first rewrite the statics equations in a matrix form:

$$\mathbf{Ap} = \mathbf{t}, \tag{2}$$

where **t** is a vector of reflection time observations, **p** is a vector of unknown parameters, including source-statics term, receiver-statics term, residual normal movement moveout term, and CDP structure term. **A** is a *m*-by-*n* matrix that its elements correspond to the left hand of equation (1). The problem of solving the residual statics becomes to solving the matrix equation (2). Usually, *m* is much larger than *n* and this equation can be solved by Least-Squares (Wiggin et al, 1976).

There are many methods available for computing the least-squares solution to the above equation, i.e. the Gauss-Newton method, the Marquardt-Levenberg method, the singular value decomposition method, and the conjugate gradient method. The Gauss-Newton method can solve this equation by transforming it into a normal equation. However, the Marquardt-Levenberg method can bring about a constrained least-squares solution and produces significant improvements in computational precision, the singular values decomposition (SVD) can produce a solution equivalent to the Marquardt-Levenberg method when applied to the same system of normal equations, but both of them are time-consuming. Conjugate gradient (CG) methods are not time-consuming, but some CG methods require that the matrix **A** be square.

In this paper, the least-squares QR factorization (LSQR), based on Lanczos and QR factorization is presented, and numerical tests are described comparing LSQR with several other algorithms for solving the residual statics problem.

DESCRIPTION OF METHODS

Based on the Lanczos decomposition, the matrix **A** can be decomposed by bidiagonalization (Paige and Saunders, 1982). The method can be described as follows:

first let

$$\beta_1 \mathbf{u_1} = \mathbf{t}, \alpha_1 \mathbf{v_1} = \mathbf{A}^T \mathbf{u_1}, \tag{3}$$

and for $i = 1, 2, \ldots,$ we compute

$$\beta_{i+1} \mathbf{u_{i+1}} = \mathbf{A v_i} - \alpha_i \mathbf{u_i}, \alpha_{i+1} \mathbf{v_{i+1}} = \mathbf{A}^T \mathbf{u_{i+1}} - \beta_{i+1} \mathbf{v_i}, \tag{4}$$

where $\{\alpha_i\}$ and $\{\beta_i\}$, $i = 1, 2, \ldots$ are positive numbers chosen so that $\| \mathbf{u_i} \| = \| \mathbf{v_i} \|$. If we define

$$\mathbf{U_k} = (\mathbf{u_1}, \mathbf{u_2} \ldots \mathbf{u_k}), \mathbf{V_k} = (\mathbf{v_1}, \mathbf{v_2} \ldots \mathbf{v_k}) \tag{5}$$

and

$$\mathbf{B_k} = \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & & & \\ & \cdot & & \\ & & \cdot & \\ & & & \alpha_k \\ & & & \beta_k \end{bmatrix}, \tag{6}$$

we can obtain the recursive relation:

$$\begin{aligned} &\mathbf{A V_k} = \mathbf{U_{k+1} B_k}, \\ &\mathbf{A}^T \mathbf{U_{k+1}} = \mathbf{V_k B_k}^T + \alpha_{k+1} \mathbf{V_{k+1} e_{k+1}}^T, \\ &\mathbf{U_{k+1}} (\boldsymbol{\beta}_1 \mathbf{e_1}) = \mathbf{t}, \end{aligned} \tag{7}$$

where $\mathbf{e_i}$ is the i-th column of a unit matrix of appropriate dimension. If the exact arithmetic were used we would have $\mathbf{U_{k+1}^T U_{k+1}} = \mathbf{I}$ and $\mathbf{V_k^T V_k} = \mathbf{I}$. Seeking an approximate solution $\mathbf{x_k} \in \Re^{\mathbf{k}}$ = span($\mathbf{V_k}$) in k-dimensional Krylov space, we write $\mathbf{x_k} = \mathbf{V_k f_k}$ (Yao et al., 1999). It then follows that

$$\| \mathbf{t} - \mathbf{A p_k} \| = \| \beta_1 \mathbf{e_1} - \mathbf{B_k f_k} \|. \tag{8}$$

and $\| \mathbf{t} - \mathbf{A p_k} \|$ is minimized over all $\mathbf{x_k}$ in span($\mathbf{V_k}$) by taking $\mathbf{f_k}$ to be the solution to the least-squares problem, i.e. min $\| \beta_1 \mathbf{e_1} - \mathbf{B_k f_k} \|$ (Yao et al., 1999).

The relations above are the basis for the LSQR algorithm. One of the advantages of this algorithm is that only the most recent columns of matrix $\mathbf{U_k}$ and $\mathbf{V_k}$ need to be stored, which leads to a small computer memory space requirement when a very large matrix calculation is involved. Moreover, during the procedure of the calculations, the matrix **A** is preserved. Therefore, if the matrix is very sparse, the cost of the computation is economical.

To summarize: an application of the Lanczos process to the problem min $\| \mathbf{t} - \mathbf{Ap_k} \|$ leads to a series of sub-problems min$\| \beta_1 \mathbf{e_1} - \mathbf{B_k} \mathbf{f_k} \|$ which can effectively be dealt with using a conventional QR factorization of $\mathbf{B_k}$. This observation forms the basis for algorithm LSQR (Paige and Saunders, 1982).

NUMERICAL EXAMPLE

To test the performance of the LSQR algorithm and compare it with several other residual-statics algorithms (i.e. the Gauss-Newton method, the Marquardt-Levenberg method, the singular value decomposition method, and the conjugate gradient method), we use a seismic line shot as a split-spread. This line consists of 20 shots, with 120 receivers for each shot. The shot-point spacing is 60 m, and the group interval is 20 m. Instead of using a variable CDP structure, we use a simple linear structure, with just two CDP points for the CDP structure term.

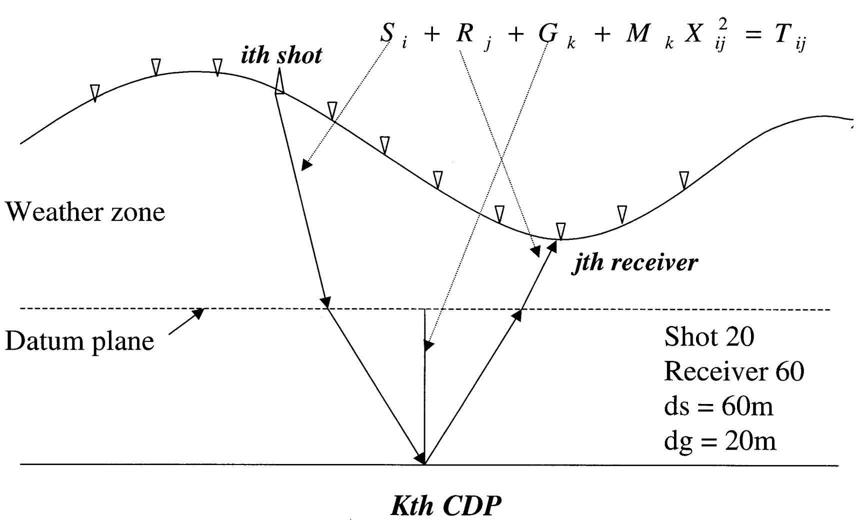

Figure 2. Residual statics analysis model.

SOLUTIONS

Here we compare LSQR numerically with the other four methods: The Gauss-Newton method, the Marquardt-Levenberg method, the SVD method (CREWES tool box), and the CG method (CREWES tool box). We use the program restat.m (CREWES tool box) to generate the matrix **A** and the observation time vector **t**.

Illustrations show the Gauss-Newton solutions (Figure 3), the Marquardt-Levenberg (Figure 4), the Singular Value Decomposition (SVD) solutions (Figure 5) and the CG solutions (Figure 6) and the LSQR solutions (Figure 7). At the same time, we calculate the root-mean-square (RMS) errors of receivers, shots, and CDP structure terms for the five methods, respectively.

Comparing the solutions of all five methods and their RMS error values, it turns out that the RMS error of receivers, shots, and CDP structure terms for the Gauss-Newton method are 0.0316, 0.0141, 0.0566, respectively. For the other four methods, the RMS error of receivers, shots, and CDP structure terms are very close, about 0.0044, 0.0015, 0.0067, respectively. So, we can say that the Marquardt-Levenberg method, the singular value decomposition (SVD) method, the CG method, and the LSQR method have the same efficiency in getting receiver and shot statics solutions, and they all have significant improvement in computational precision compared with the Gauss-Newton method.

The computer times for the five methods are 4.1663 s, 3.9663 s, 43.0387 s, 2.2938 s, and 1.6823 s, respectively. The cumulative number of flops for the five methods are 97101567, 142419708, 992382802, 48261269, 7488958, respectively. So, we can say that the SVD method is the most time-consuming method; the CG method and the LSQR method are faster than other methods. In addition, the LSQR method is the fastest method.

For further comparing the CG method and the LSQR method, we plot the root mean square error of the observation time for both CG and LSQR methods as a function of the iterative number. This function is shown in Figure 8. We can see clearly that the RMS error of the observation time for CG method is bigger than the RMS error of the observation time for the LSQR method, especially, in the beginning of iteration and we can say that the LSQR method converges faster and it is more stable than the CG method.

We can also notice that the RMS error of the CDP structure term in Figure 6 and Figure 7 is relatively larger than others. This may depend on the fact that only two CDP points are used for the inversion. In order to improve the solution of the CDP term, a weighted least-squares is tried. Figure 9 and Figure 10 show the solutions of both CG and LSQR methods after a weighting factor = 100 is applied to the CDP term. The RMS error of CDP term for the CG method is reduced from 0.0070 (before weighting) to 0.0058 (after weighting), while the RMS error of CDP term for LSQR method drops from 0.0068 (before weighting) to 0.0052 (after weighting). The solutions of the CDP term for both methods have been improved. However, unlike LSQR, the RMS error of the receiver for the CG method increases a lot. The reason may come from the fact that in CG method, the original equation has to be transformed into the normal equation. This may also show that LSQR is more robust.

After weighting, we see again the RMS error of observation time for both CG and LSQR methods as a function of the iterative number (Figure 11). The RMS of the observation time for LSQR is still less than RMS of the observation time for the CG method, and it shows again that LSQR method converges faster and is more stable than the CG method. But also we can notice that the RMS errors of the observation time for the CG and LSQR methods are bigger than the values before weighting. This may be because the objective function to be minimized differs from the original one.

CONCLUSIONS

We have shown that the residual-statics problem associated with observations in CDP data can be cast in the form of a matrix equation that relates the source and receiver statics, residual normal moveout, and structure to the observed time deviation among the traces; and the problem of solving the statics can become the problem of solving the linear equation.

There are many methods that can be used to solve the linear equation, e.g. the Gauss-Newton method, the Marquardt-Levenberg method, the singular value decomposition method, and the conjugate gradient method. In this paper, another kind of CG method, Least-Squares QR factorization, based on Lanczos and QR factorization is presented. This method works in general on a rectangular matrix, has a fast convergence and is a stable algorithm to a matrix problem that is ill-posed and sparse. The numerical example shows that this method can be directly used for solving the residual-statics problem with high precision and high speed. In addition, relatively little computer memory space is required. Evidently, the LSQR method is by far the most robust algorithm.

ACKNOWLEDGEMENTS

We thank the sponsors of the CREWES Project for their financial support.

REFERENCES

Claerbout, J., 1986, Imaging the Earth's interior, Blackwell.

Lines, L.R. and Treitel, S., 1984, A review of least-squares inversion and its application to geophysical problems: Geophysical Prospecting **32**, 159-186.

Paige, C.C. and Saunders, M.A., 1982. LSQR: an algorithm for sparse linear equations and sparse least squares: ACM Trans, Math. Soft., **8**, 195-209.

Wiggins R.A., Larner K.L., and Wisecup R.D., 1976, Residual statics analysis as a linear inverse problem: Geophysics, **41**, 922-938.

Yao Z.S., Roberts R., and Tryggvason A., 1999, Calculating resolution and covariance matrices for seismic tomography with the LSQR method: Geophysical Journal International, **138**, 886-894.

Yilmaz, O., 1987, Seismic Data Processing: Society of Exploration Geophysicists.

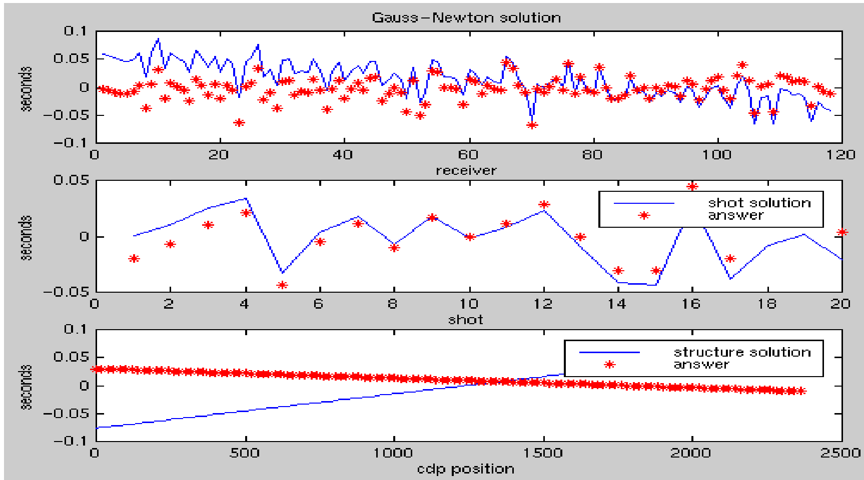

Figure 3. The statics solutions for the Gauss-Newton method.

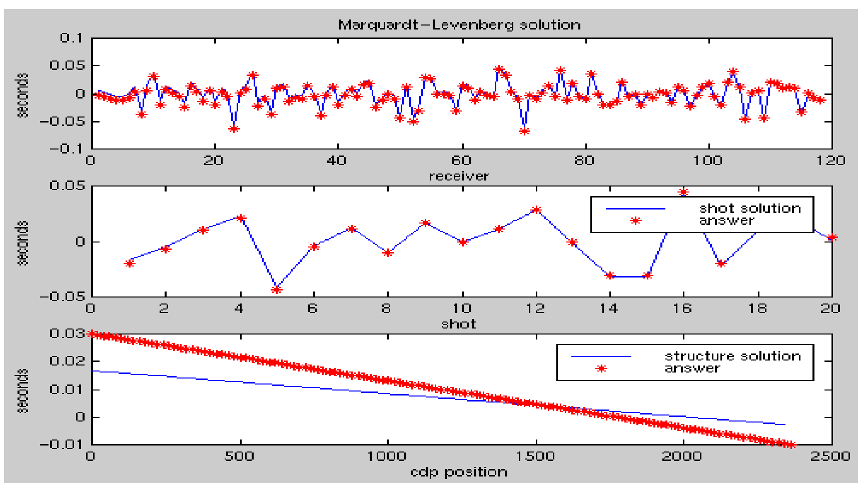

Figure 4. The statics solutions for the Marquardt-Levenberg method.

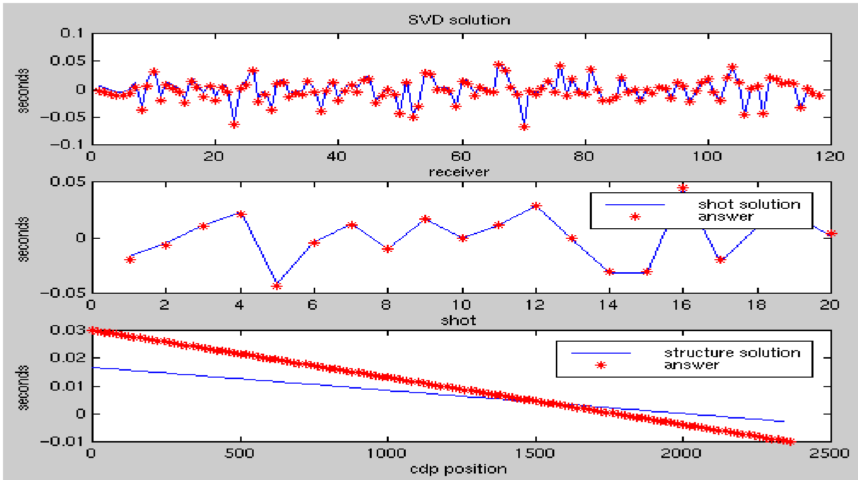

Figure 5. The statics solutions for the SVD method.

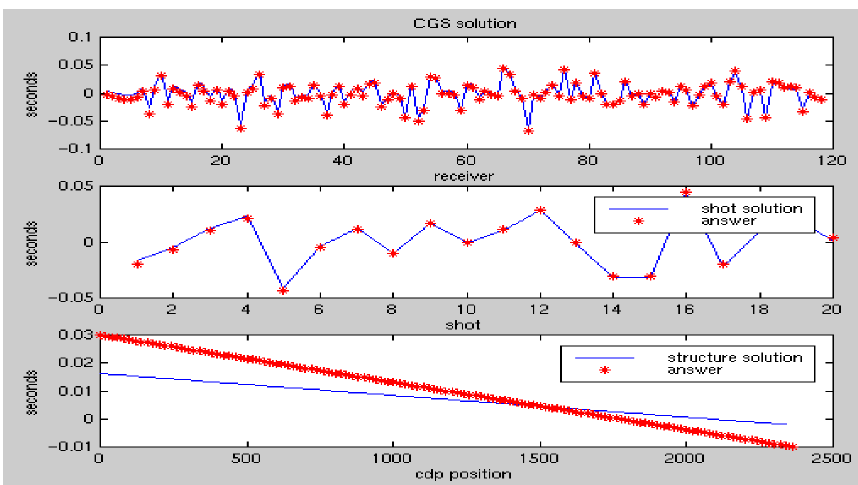

Figure 6. The statics solutions for the CG method.

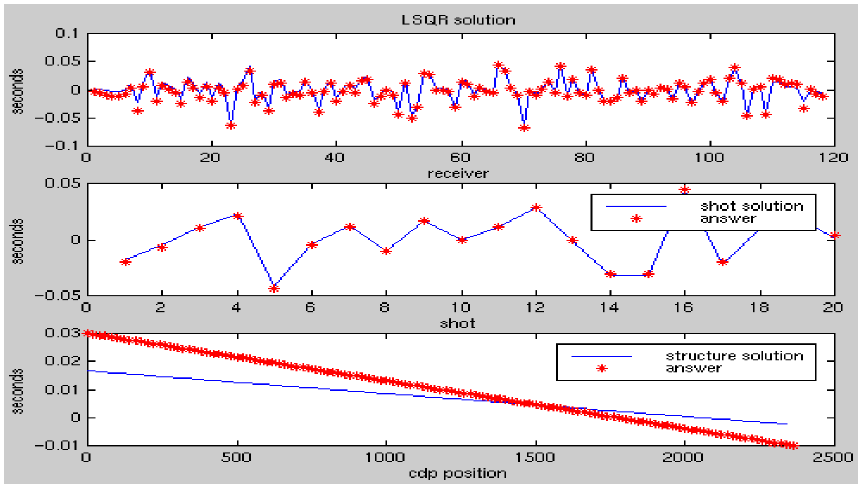

Figure 7. The statics solutions for the LSQR method.

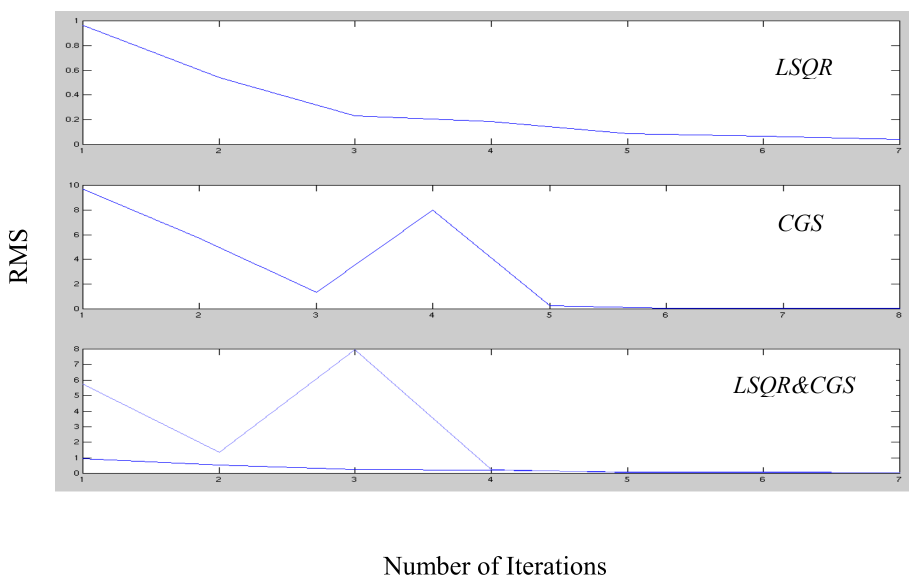

Figure 8. The RMS error of the observation time as a function of number of iterations.

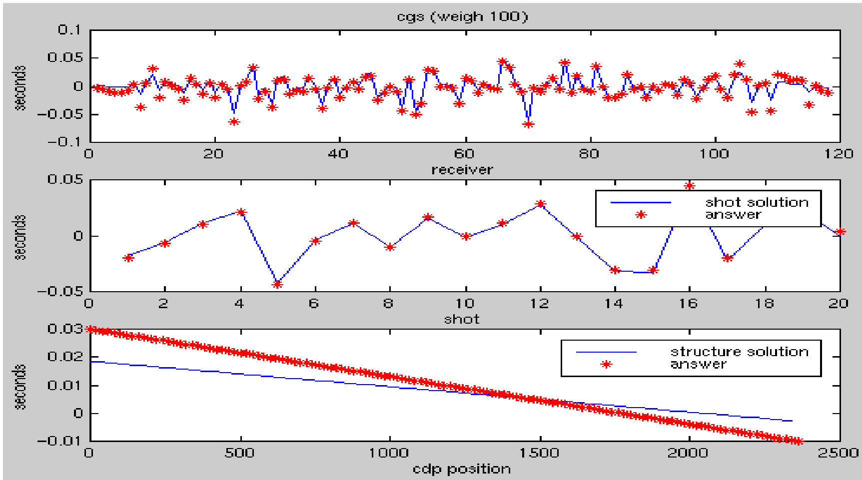

Figure 9. The statics solutions for the CG method (weighting factor =100).

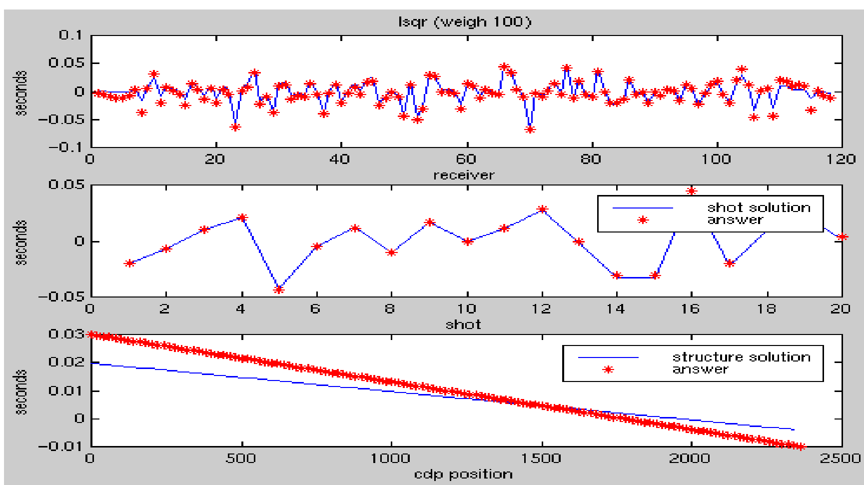

Figure 10. The statics solutions for the LSQR method (weighting factor = 100).

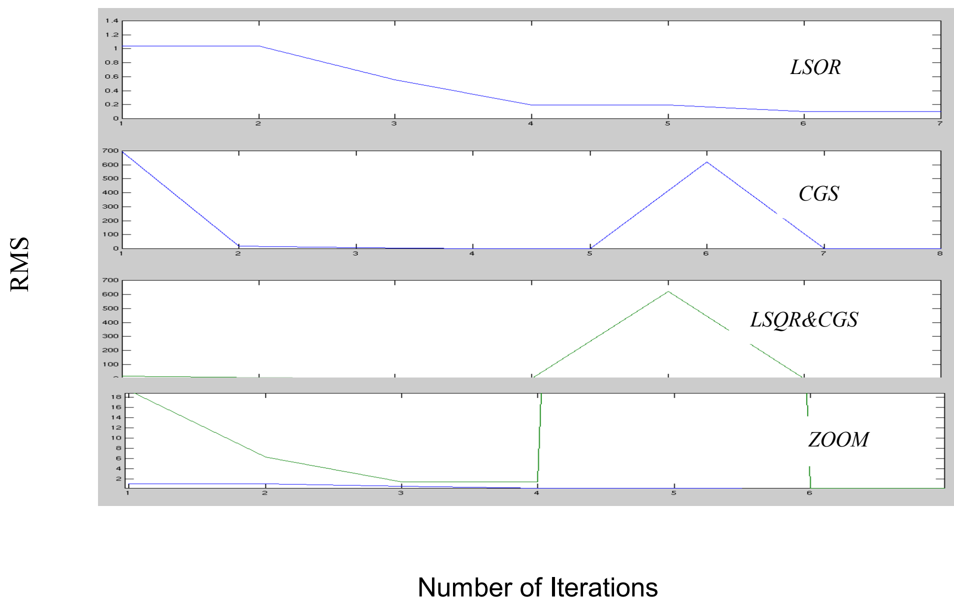

Figure 11. The RMS error of the observation time as a function of number of iterations (weighting factor = 100).