GPS integer ambiguity resolution by various decorrelation methods

Lizhi Lou and Erik W. Grafarend

Summary

In order to obtain the highest possible accuracy from GPS measurements, in particular needed for deformation analysis, carrier phase observations are used. Within the mode of double difference carrier phase, the fundamental mixed integer-real valued adjustment problem is met. Through standard parameter estimation like weighted least squares or alternative robust objective functions only the »floating solution« – including the vector of ambiguity – can be given. In order to speed up the searching process for the integer values of ambiguity, the method of decorrelation is applied which works »in practice« sufficiently well. Here we evaluate three proposals for the decorrelation of float solutions for ambiguity resolution, namely (i) the inverse integer Cholesky decorrelation (CHOL) as proposed by P. Xu (2001) (ii) the integer Gauss decorrelation (GAUSS) initiated by P. Teunissen (1997) and (iii) the A.K. Lenstra, H.W. Lenstra, and L. Lovacs (LLL) algorithm as proposed by A. Hassibi, S. Boyd (1998), and E. Grafarend (2000). The analysis of different decorrelation methods is made as realistic and as statistically meaningful as possible: a random simulation approach has been implemented which guarantees a symmetric, positive definite variance covariance matrix of the ambiguity vector (»float solution«) derived from the double difference observation variance covariance matrix. The spectral condition number is used as a criterion for the performance of the three decorrelation methods. Three sets of simulated data are finally comparatively evaluated.

Zusammenfassung

Die höchste Genauigkeit, die für die Analyse von geodätischen Deformationsnetzen Voraussetzung ist, wird nur durch die Auswertung von GPS Trägerphasen erreicht. Im Modus »doppelte Differenzen« tritt das Problem auf, dass sowohl reellwertige unbekannte Parameter als auch ganzzahlige Parameter in einem gemischt reell-ganzzahligen Ausgleichungsproblem geschätzt werden müssen. Die ganzzahligen Parameter, auch »Ambiguitäten« genannt, werden in der »float« Lösung in einem ersten Schritt reellwertig geschätzt, um danach in einem zweiten Schritt mittels der Methode der Dekorrelation als ganzzahlig identifiziert zu werden. Eine derartige Vorgehensweise hat sich insbesondere in der GPS Praxis bewährt. Hier untersuchen wir drei Methoden der Dekorrelation: (i) die inverse Cholesky Dekorrelation (CHOL) empfohlen von P. Xu (2001), (ii) die ganzzahlige Gauss Dekorrelation (GAUSS) initiiert von P. Teunissen (1997) und (iii) den A.K. Lenstra, H.W. Lenstra und L. Lovacs (LLL) Algorithmus, eingeführt von A. Hassibi, S. Boyd (1998) und E. Grafarend (2000). Um die unterschiedlichen Verfahren der Dekorrelation so realistisch und so statistisch bedeutsam wie möglich zu gestalten, wurde ein Zufallssimulator implementiert, welcher dem Vektor der »Ambiguity« eine symmetrisch positive definite Varianz Kovarianz Matrix zuweist, zugeordnet der Varianz Kovarianz Matrix der beobachteten Trägerphasendifferenzen. Die spektrale Konditionszahl wurde als Kriterium für die Qualität der untersuchten drei Dekorrelationsmethoden herangezogen. Drei umfangreiche simulierte Datensätze erlaubten eine objektive Wertung der Verfahren: CHOL und LLL ergeben nahezu identische hochwertige Resultate, etwas schwächer erweist sich GAUSS als Dekorrelationsverfahren.

1 Introduction

By means of random simulation we obtain the GPS double difference (DD) least squares solution (LESS), in particular the float solution of the ambiguity vector $\hat{x}$ and the corresponding one component variance covariance matrix $\sigma^2 Q_{\hat{x}}$. In order to speed up the searching process for the integer ambiguity resolution, the concept of decorrelation has been developed and was reported to work practically well (Teunissen P.J.G. et al. 1997; Liu L.T. et al. 1999). Three methods of GPS decorrelation are analyzed here:

(i) *Integer Gauss decorrelation;*
(ii) *Inverse integer Cholesky decorrelation;*
(iii) *Lenstra-Lenstra-Lovacs decorrelation.*

Given a positive definite matrix with off-diagonal elements, the target of decorrelation is to find a unimodular matrix in order to completely eliminate or to reduce the sizes of the correlation coefficients. But we can not tell which is the best. Firstly, we outline the three different decorrelation methods.

1.1 Integer Gauss decorrelation

The decorrelation by *integer Gauss decorrelation* is to construct the unimodular (admissible) matrix G_k, (k = 1, 2, ..., n), which is written as

$$G_k = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & \vdots & \ddots & & \\ & & -[q_{ij}/q_{ii}] & \cdots & 1 & \\ & & & & \ddots & \\ & & & & & 1 \end{bmatrix} \tag{1a}$$

if $q_{ii} \le q_{ij}$; or

$$G_k = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & \cdots & -[q_{ij}/q_{jj}] & & \\ & & & \ddots & \vdots & & \\ & & & & 1 & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix} \tag{1b}$$

if $q_{ii} > q_{jj}$ (Xu P.L. 2001).

Here $[x]$ is the operation to round the floating number x to the nearest integer, until we have

$$Q_Z = G_n \cdots G_1 Q_{\hat{x}} G_1^T \cdots G_n^T \tag{2}$$

where $|q_{ij(n)}| \le ½ \min(q_{ii(n)}, q_{jj(n)})$ for all $i \neq j$.

1.2 Inverse integer Cholesky decorrelation

The decorrelation *by inverse integer Cholesky decorrelation* is also to construct the unimodular (admissible) matrix Z_k, (k = 1, 2, ..., n), but it applies Cholesky decomposition to the symmetric positive definite matrix $Q_{\hat{x}}$ instead of repeating the above method of constructing the matrix G. By using Cholesky decomposition, $Q_{\hat{x}} = L_1 D L_1^T$ is obtained, where D is a diagonal matrix with positive elements and L_1 is a lower triangular matrix with diagonal elements equal to 1. And then $Z_1 = [L_1^{-1}]$, $Q_1 = Z_1 Q_{\hat{x}} Z_1^T$ are acquired. In general, $Z_1 \neq L_1^{-1}$, namely, $Q_1 \neq D$, or Q_1 is not a diagonal matrix. Thus is the iteration implemented in our inverse integer Cholesky decorrelation. That is to say, one should calculate $Q_1 = L_2 D L_2^T$, $Z_2 = [L_2^{-1}]$, ..., $Q_Z = Z_n \cdots Z_1 \;\; Q_{\hat{x}} Z_1^T \cdots Z_n^T$, until Z_{n+1} is a unit matrix.

1.3 Lenstra-Lenstra-Lovacs decorrelation

The basic idea of the *LLL algorithm* is to use the integer Gram-Schmidt orthogonalization process to generate an almost orthogonal basis (Grafarend E.W. 2000). The first step is to obtain the inverse matrix $P_{\hat{x}}$ of the variance-covariance matrix $Q_{\hat{x}}$, and then $P_{\hat{x}}$ is decomposed as $P_{\hat{x}} = BB^T$, where B is a certain matrix of full rank. Here B is reached by using Cholesky decomposition. By letting $b_1, b_2, ..., b_t$ be row vectors of B, which are linearly independent, the integer Gram-Schmidt orthogonalization process then works as following:

$$b_i^* = b_i - \sum_{j=1}^{i-1} u_{ij} b_j^*$$

$$u_{ij} = \left[< b_i, b_j^* > / < b_j^*, b_j^* > \right] \tag{3}$$

where <,> denotes the ordinary inner product, $\boldsymbol{b}_1^*, \boldsymbol{b}_2^*, \cdots, \boldsymbol{b}_t^*$ form the almost orthogonal matrix B^*. In order to improve the orthogonality of the new basis, here iteration is also applied till $u_{ij} = 0$ for all $j < i$. Thus, a unimodular matrix which is also a lower triangular matrix is given, namely

$$L = \begin{bmatrix} 1 & & & & \\ u_{21} & 1 & & & \\ u_{31} & u_{32} & 1 & & \\ \vdots & \vdots & \vdots & \ddots & \\ u_{t1} & u_{t2} & u_{t3} & \cdots & 1 \end{bmatrix} \tag{4}$$

and $B = LB^*$. Therefore,

$$\|\hat{x} - x_z\|_{P_{\hat{x}}}^2 = \left\|B^T(\hat{x} - x_z)\right\|^2 = \left\|B^{*T} L^T (\hat{x} - x_z)\right\|^2 = min \tag{5}$$

i.e.

$$\begin{aligned} &(\hat{x} - x_z)^T L B^* B^{*T} L^T (\hat{x} - x_z) \\ &= (\hat{x} - x_z)^T L (D + E) L^T (\hat{x} - x_z) = min. \end{aligned} \tag{6}$$

Here $B^* B^{*T} = D + E$ is not diagonal but almost diagonal (Grafarend E.W. 2000).

Secondly, we outline the method of random simulation. Although we can use a particular positive definite matrix to compare different decorrelation methods, such a comparison is of limited practical value (Xu P.L. 2001). Let $Q_{\hat{x}}$ be decomposed as follows:

$$Q_{\hat{x}} = L \Lambda L^T \tag{7}$$

where Λ is the diagonal matrix with positive elements $\lambda_1, \lambda_2, ..., \lambda_t$, for more commonly, the values of λ_i are also stochastic, not sorted as $\lambda_1 \le \lambda_2 \le ... \le \lambda_t$. L is a lower triangular matrix with diagonal elements equal to 1, the other elements are simulated by random function and are not integers. And thus is a symmetric positive definite matrix $Q_{\hat{x}}$ with random dimensions obtained.

2 Modification of some methods

As we have mentioned above, iterations are needed in the *inverse integer Cholesky decorrelation* and the *LLL algorithm*. In the *inverse integer Cholesky decorrelation* the criterion for stopping iteration is given, namely, Z_{n+1} is a unit matrix. It should also be mentioned that a larger condition number might be encountered if the matrix is not sorted by ascending order.

Example 1 (Cholesky decorrelation)
Assume that the real-valued ambiguity vectors and their variance-covariance matrix

$$Q_{\hat{x}} = \begin{bmatrix} 53.4 & 38.4 \\ 38.4 & 28.0 \end{bmatrix}$$

are given by means of the LS (least squares). When the inverse integer Cholesky decomposition is used directly, the transformation matrix Z_1 and corresponding Q_Z are

$$Z_1 = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}, \quad Q_z = \begin{bmatrix} 53.4 & -15.0 \\ -15.0 & 4.6 \end{bmatrix}.$$

Then the iteration approach is applied, the resulting transformation matrix Z_2 is a unit one with the condition number of Q_Z being 160.979. If we sort the matrix with ascending order before the Cholesky decomposition is carried out, we obtain the following result:

$$Z_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Q_{Z_1} = \begin{bmatrix} 28.0 & 38.4 \\ 38.4 & 53.4 \end{bmatrix}.$$

After the inverse integer Cholesky decomposition is applied, a different result is found:

$$Z_2 = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}, \; Z = Z_2 Z_1 = \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix}, \; Q_Z = \begin{bmatrix} 28.0 & 10.4 \\ 10.4 & 4.6 \end{bmatrix}.$$

Thus through iteration approach, the resulting transformation matrix Z_3 presenting a unit matrix, the condition number of Q_Z now reads 49.470, which is much smaller than the one acquired without matrix sorting. But this is still not the smallest condition number. If we resort the matrix with ascending order before iterating each time, the new results will be

$$Z_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Z_2 = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}, Z_3 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Z_4 = \begin{bmatrix} 1 & 0 \\ -2 & 1 \end{bmatrix},$$

$$Z = Z_4 Z_3 Z_2 Z_1 = \begin{bmatrix} 1 & -1 \\ -2 & 3 \end{bmatrix}, \; Q_Z = \begin{bmatrix} 4.6 & 1.2 \\ 1.2 & 4.8 \end{bmatrix}.$$

The condition number of Q_Z is now 1.689. In this case we may encounter the divergence of the iteration process. In the following flow chart (Figure 1) the processing method will be described.

As for the *LLL algorithm*, we also give an iteration-stopping criterion that for each vector b_i^* all the corresponding $u_{ij} = 0$. However, this is only valid for most cases. For some cases, divergence of the iteration is still encountered.

Example 2 (LLL algorithm)
Assume that by means of the LS (least squares) we have got the real-valued ambiguity vectors and their variance-covariance matrix

$$Q_{\hat{x}} = \begin{bmatrix} 1.000 & 0.517 & 0.534 & 0.020 & 0.148 & 0.485 \\ 0.517 & 1.267 & 0.277 & 0.773 & 0.350 & 0.757 \\ 0.534 & 0.277 & 1.285 & 0.685 & 0.335 & 0.399 \\ 0.020 & 0.773 & 0.685 & 2.036 & 1.315 & 1.268 \\ 0.148 & 0.350 & 0.335 & 1.315 & 2.029 & 1.212 \\ 0.485 & 0.757 & 0.399 & 1.268 & 1.212 & 1001.174 \end{bmatrix}.$$

While applying the *LLL algorithm*, the inverse matrix $P_{\hat{x}}$ of $Q_{\hat{x}}$ should be first computed, and then $P_{\hat{x}}$ is decomposed as $P_{\hat{x}} = BB^T$, thus the corresponding matrices $P_{\hat{x}}$ and B are

$$P_{\hat{x}} = \begin{bmatrix} 2.399 & -1.319 & -1.235 & 1.251 & -0.554 & -0.001 \\ -1.319 & 1.772 & 0.676 & -1.169 & 0.437 & 0.000 \\ -1.235 & 0.676 & 1.592 & -1.020 & 0.372 & 0.000 \\ 1.251 & -1.169 & -1.020 & 1.868 & -0.931 & -0.001 \\ -0.554 & 0.437 & 0.372 & -0.931 & 1.000 & 0.000 \\ -0.001 & 0.000 & 0.000 & -0.001 & 0.000 & 0.001 \end{bmatrix}$$

$$B = \begin{bmatrix} 1.549 & 0 & 0 & 0 & 0 & 0 \\ -0.851 & 1.023 & 0 & 0 & 0 & 0 \\ -0.797 & -0.003 & 0.978 & 0 & 0 & 0 \\ 0.808 & -0.471 & -0.386 & 0.919 & 0 & 0 \\ -0.358 & 0.129 & 0.089 & -0.595 & 0.702 & 0 \\ 0.000 & 0.000 & 0.000 & 0.000 & -0.001 & 0.032 \end{bmatrix}.$$

Applying the integer Gram-Schmidt orthogonalization process to the row vectors of B, which are linearly independent, we obtain

$$\begin{aligned} & b_1^* = b_1 \Rightarrow \\ & u_{21} = [-1.319/2.399] = -1 \Rightarrow \\ & b_2^* = [0.697 \quad 1.023 \quad 0 \quad 0 \quad 0 \quad 0] \Rightarrow \\ & u_{21}^* = [1.080/2.399] = 0 \end{aligned} \quad .$$

Now that $u_{21}^* = 0$, we stop the iteration of computing b_2^* and transfer to b_3^* calculation to obtain

$$\left.\begin{aligned} & u_{31} = [-1.235/2.399] = -1 \\ & u_{32} = [-0.559/1.534] = 0 \end{aligned}\right\} \Rightarrow$$

$$b_3^* = [0.752 \quad -0.003 \quad 0.978 \quad 0 \quad 0 \quad 0] \Rightarrow$$

$$\left.\begin{aligned} & u_{31}^* = [1.164/2.399] = 0 \\ & u_{32}^* = [0.521/1.534] = 0 \end{aligned}\right\} .$$

Again, since $u_{31}^* = 0, u_{32}^* = 0$, the iteration of computing b_3^* is ended and b_4^* is further calculated

$$\left.\begin{array}{l} u_{41} = [1.251/2.399] = 1 \\ u_{42} = [0.082/1.534] = 0 \\ u_{43} = [0.231/1.522] = 0 \end{array}\right\} \Rightarrow$$

$$b_4^* = [-0.741 \quad -0.471 \quad -0.386 \quad 0.919 \quad 0 \quad 0] \Rightarrow$$

$$\left.\begin{array}{l} u_{41}^* = [-1.148/2.399] = 0 \\ u_{42}^* = [-0.999/1.534] = -1 \\ u_{43}^* = [-0.934/1.522] = -1 \end{array}\right\} \Rightarrow$$

$$b_4^{**} = [0.708 \quad 0.549 \quad 0.592 \quad 0.919 \quad 0 \quad 0] \Rightarrow$$

$$\left.\begin{array}{l} u_{41}^{**} = [1.097/2.399] = 0 \\ u_{42}^{**} = [1.056/1.534] = 1 \\ u_{43}^{**} = [1.109/1.522] = 1 \end{array}\right\} \Rightarrow$$

$$b_4^{***} = [-0.741 \quad -0.471 \quad -0.386 \quad 0.919 \quad 0 \quad 0] = b_4^* \,.$$

Now it is found that $u_{41}^*, u_{42}^*, u_{43}^*$ are not all zero, and $b_4^{***} = b_4^*$, which means that the iteration for b_4^* is divergent, leading to the unavailability of other vectors b_5^* and b_6^*.

Because of the divergence of the iteration process, a criterion should be given to control the iteration process and avoid possible dead circulation.

Employing the method of the *inverse integer Cholesky decorrelation*, we can acquire the condition number in each iteration. The condition number can be regarded as an indicator. In this process, the minimum condition number C_0 and the corresponding Z_0, Q_0 will be saved. Compare the obtained condition number C_1 in each iteration, (i) if the condition number C_1 continuously grows up, the times of iteration *Niter* increase by one automatically. The iteration process will be stopped when *Niter* exceeds a given number N (See the flow chart in Figure 1); (ii) if C_1 is smaller than the value of saved minimum condition number C_0, we set *Niter* equal to zero and replace C_0, Z_0, Q_0 with C_1, Z, Q_Z, respectively, continue the process of iteration; (iii) if the condition number C_1 changes no longer or equal to the saved minimum condition number C_0, the iteration process will be stopped too. The flow chart is described in Fig. 1, where *Niter* denotes the times of iteration. Certainly, in order to obtain the smallest condition number, we sort the matrix $Q(Q_Z)$ in each iteration with ascending order, the corresponding transformation matrix is obviously also an integer matrix.

In order to see whether iterations could improve the conditioning of Q matrix, in the same time, without loss of generality, we sample 200 non-informative random matrices. The logarithmic condition numbers and the dimensions of these examples are shown in Figure 2.

Figure 3 shows that the logarithmic condition numbers of matrices after applying the *inverse integer Cholesky decorrelation* are much smaller by iteration. It clearly demonstrates that the iteration results are much better than that uniterated after applying the *inverse integer Cholesky decorrelation*.

In the *LLL algorithm*, at first we can obtain the inverse matrix of Q as weight matrix P. In the former calculation procedure, from each row vector b_i of the matrix B, we can get b_i^* through iteration, the criterion is $u_{ij} = 0$ for all $j < i$. As shown in Example 2, when one of the vectors

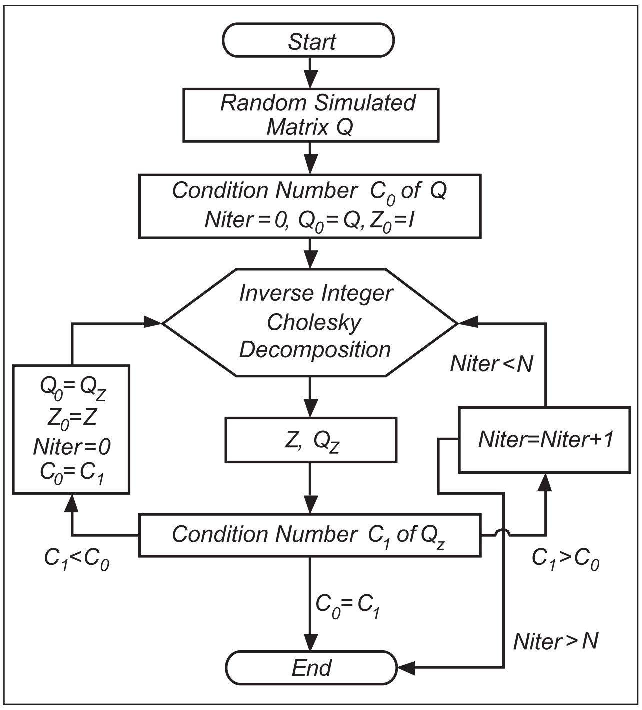

Fig. 1: Flow chart of the inverse integer Cholesky decorrelation

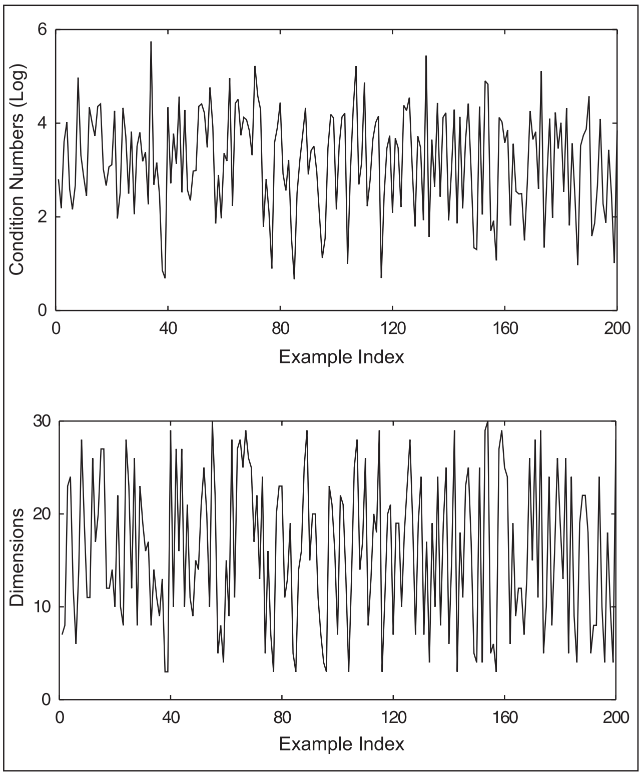

Fig. 2: Logarithmic condition numbers and dimensions of 200 examples

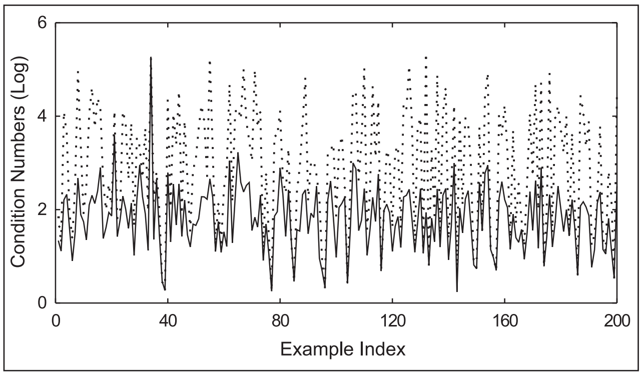

Fig. 3: Logarithmic condition numbers of matrices after applying the inverse integer Cholesky decorrelation. Solid line – iterated, dotted line – uniterated.

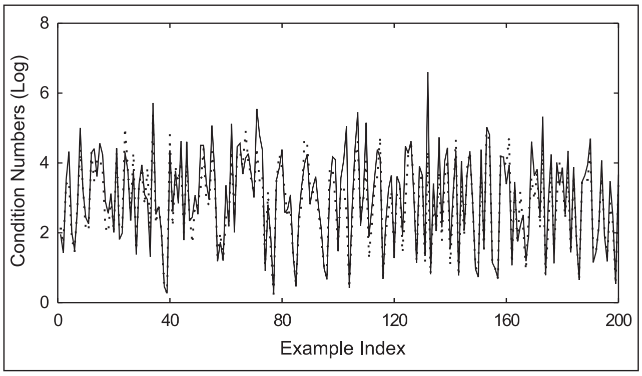

Fig. 5: Logarithmic condition numbers of matrices after applying the LLL algorithm. Solid line – iterated LLL algorithm, dotted line – uniterated LLL algorithm.

failed to converge after iteration, the other following vectors will lose the foundation for computation. Since the *LLL algorithm* is an orthogonalization process, we may calculate all the ordinary inner products between this processing vector and the different vectors, which have been processed, and work out their quadratic sum, which acts as a new criterion. The better the degree of orthogonalization, the closer to zero is the quadratic sum. Thus we obtain the new flow chart of the *LLL algorithm* (See Figure 4). The corresponding results of applying the *LLL algorithm* are shown in Figure 5.

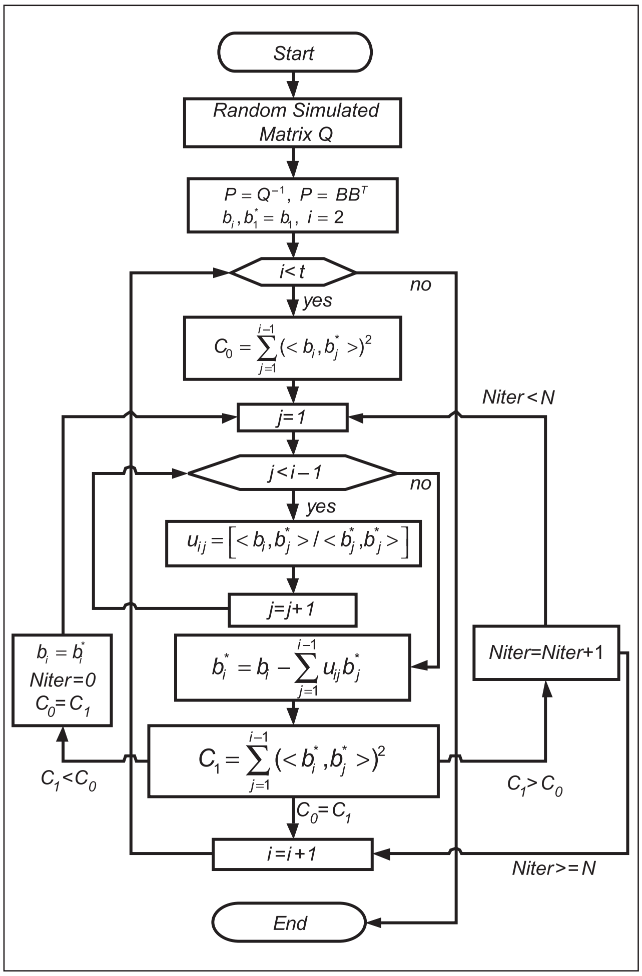

Fig. 4: Flow chart of the modified LLL algorithm.

It is observed from Fig. 5 that the logarithmic condition numbers of matrices after applying the *LLL algorithm* decrease not as significantly as expected. We may take another criterion into account. Therefore, with the same integer Gram-Schmidt orthogonalization process, we can just simply calculate and get all the vectors b_i^* without iteration, and then follow the similar way as processing Q_Z in the *inverse integer Cholesky decorrelation*, only change into integer Gram-Schmidt orthogonalization process and replace Q with P^*. In this way, not each vector of B^*, but the matrix $P^* = B^* B^{*T}$ as a whole is dealt with. So the condition number can also be taken into consideration, and the iteration process will be carried out. We call this the *modified LLL algorithm*. However, whether iterations could improve the conditioning of P^* can be tested by experiments, the results are shown in Fig. 6.

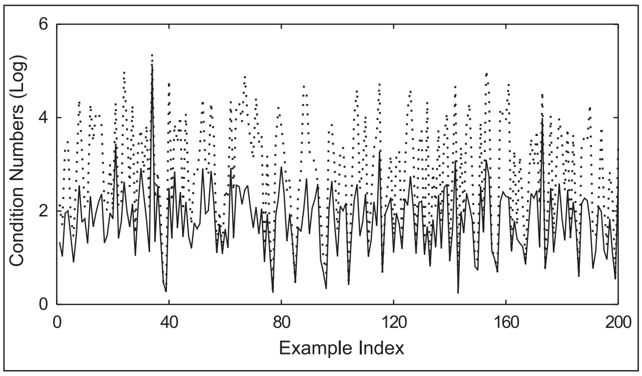

Fig. 6: Logarithmic condition numbers of matrices after applying the LLL algorithm. Solid line – modified LLL algorithm, dotted line – unmodified LLL algorithm.

It clearly shows that the results of the examples are much better after applying the *modified LLL algorithm*.

3 Comparison of decorrelation methods

Here we will compare the performances of decorrelation of the *integer Gauss decorrelation*, the *inverse integer*

Cholesky decorrelation and the *LLL algorithm* with the 200 examples. For simplicity, we will denote these three methods by GAUS, CHOL and LLL in the following comparisons. A number of factors have been shown in the previous sections to significantly influence the results of the *inverse integer Cholesky decorrelation* and the *LLL algorithm*, we will concentrate on the comparison with the other method – the *integer Gauss decorrelation*.

We computed the condition numbers of the 200 examples by GAUS, CHOL and LLL, and compared them with the simulated ones in order to understand the performances of decorrelation by the three methods. The results are shown in Figures 7, 8 and 9. The three methods have generally performed well for the decorrelation of the 200 examples.

We now compared the three methods. Figure 10 shows the logarithmic condition numbers of the 200 examples after applying the *inverse integer Cholesky decorrelation* and the *integer Gauss decorrelation*, respectively. We can find that CHOL is better than GAUS. Figure 11 shows the logarithmic condition numbers of the 200 examples after applying the *inverse integer Cholesky decorrelation* and the *LLL algorithm*, respectively. It means that CHOL is similar to LLL. Figure 12 shows the logarithmic condition numbers of the 200 examples after applying the *LLL algorithm* and the *integer Gauss decorrelation*, respec-

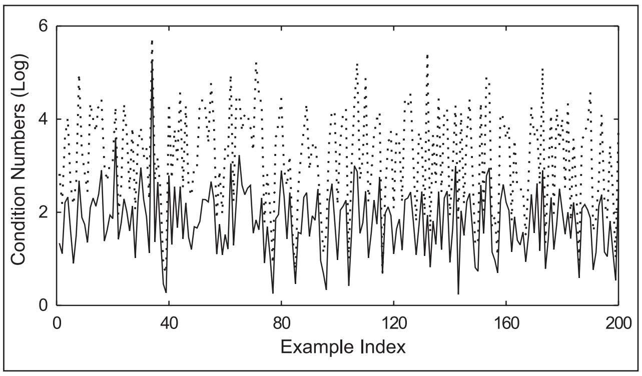

Fig. 7: Logarithmic condition numbers of 200 examples. Solid line – CHOL; Dotted line – original condition numbers

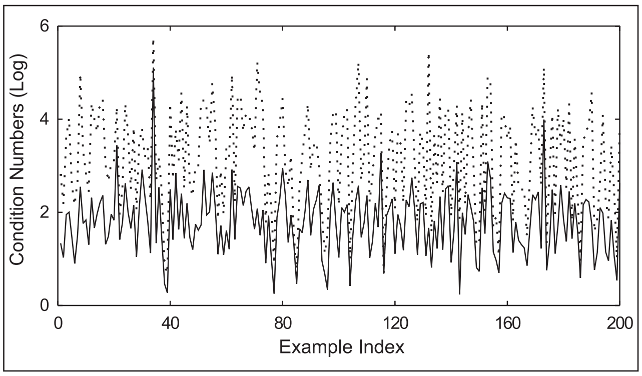

Fig. 8: Logarithmic condition numbers of 200 examples. Solid line – LLL; Dotted line – original condition numbers

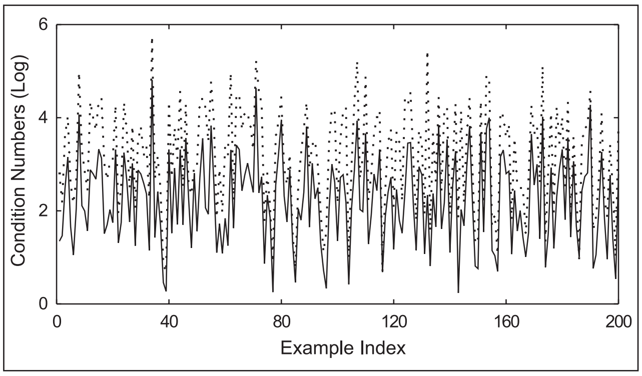

Fig. 9: Logarithmic condition numbers of 200 examples. Solid line – GAUS; Dotted line – original condition numbers

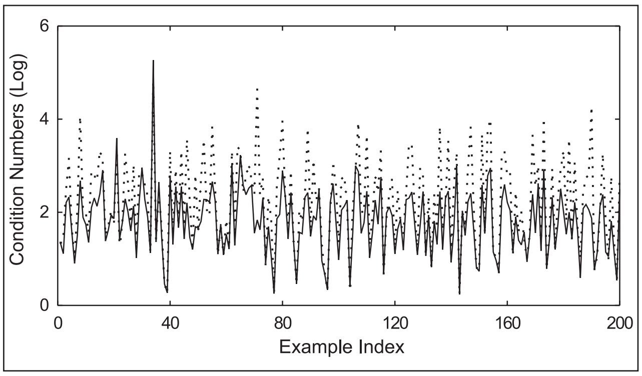

Fig. 10: Logarithmic condition numbers of 200 examples after decorrelation. Solid line – CHOL; Dotted line – GAUS

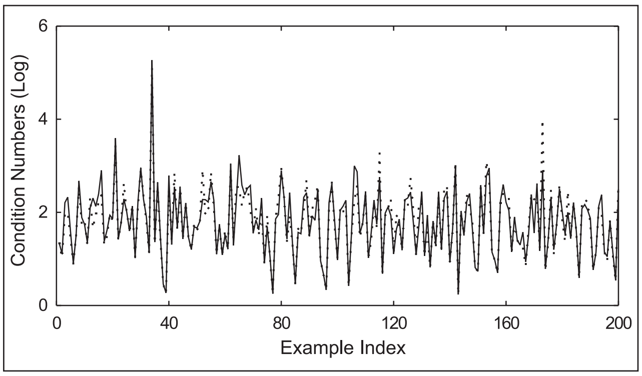

Fig. 11: Logarithmic condition numbers of 200 examples after decorrelation. Solid line – CHOL; Dotted line – LLL.

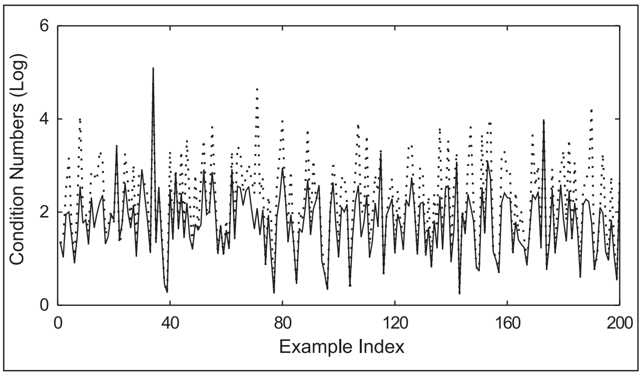

Fig. 12: Logarithmic condition numbers of 200 examples after decorrelation. Solid line – LLL; Dotted line – GAUS

tively. It clearly demonstrates that LLL is better than GAUS. Thus, the *integer Gauss decorrelation* is the worst one of the three methods, the *inverse integer Cholesky decorrelation* and the *LLL algorithm* are better than it.

Figures 13, 14 and 15 show the differences of the condition numbers by subtracting the simulated condition numbers from those obtained by GAUS, CHOL and LLL. Negative ratios imply an improvement of conditioning of Q (or P), namely, the decrease of the condition number. However, the *LLL algorithm* is the best method with a mean value of –1.3823, the *inverse integer Cholesky decorrelation* and the *integer Gauss decorrelation* are slightly worse with mean values of –1.3713 and –0.9502, respectively. Table 1 gives more details of the statistics of the 200 examples, which further indicate that the *LLL algorithm* and the *inverse integer Cholesky decorrelation* perform better compared to the *integer Gauss decorrelation*. Although the *LLL algorithm* is only slightly better than the *inverse integer Cholesky decorrelation* on average, it has the larger standard deviation.

Table 1: Statistics of a comparison of the results by GAUS, CHOL and LLL with the condition numbers of the simulated examples

Method	Max	Mean	Min	Stdev
GAUS	+0.043	–0.9502	–3.103	0.4319
CHOL	–0.011	–1.3713	–3.503	0.5916
LLL	–0.011	–1.3823	–3.778	0.6016

Stdev: Standard deviation

The aim of GPS ambiguity searching is to find an integer vector x that satisfies the following minimization (Teunissen P.J.G. et al. 1998; Hofmann-Wellenhof B. et al. 1997):

$$(\hat{x}-x)^T Q_{\hat{x}}^{-1}(\hat{x}-x) = min\ , \tag{8}$$

where $\hat{x}$ is a floating-valued vector solved by least squares, x is the corresponding ambiguities and should be solved, $Q_{\hat{x}}$ is the variance-covariance matrix of $\hat{x}$ and is symmetric positive definite. Through the decorrelation of $Q_{\hat{x}}$ the corresponding transformation matrix Z is obtained. Thus we calculate:

$$Q_Z = ZQ_{\hat{x}}Z^T\ . \tag{9}$$

As a consequence, the integer LS (least squares) problem (Eq. 8) turns into:

$$(z-z_0)^T Q_Z^{-1}(z-z_0) = min\ , \tag{10}$$

where $z = Zx$, $z_0 = Z\hat{x}$. With $z = [z_0]$, the different methods can lead to different values of $(z-z_0)^T Q_Z^{-1}(z-z_0)$. This value may also be used as a criterion to check which method is better. The results are shown in Figure 16,

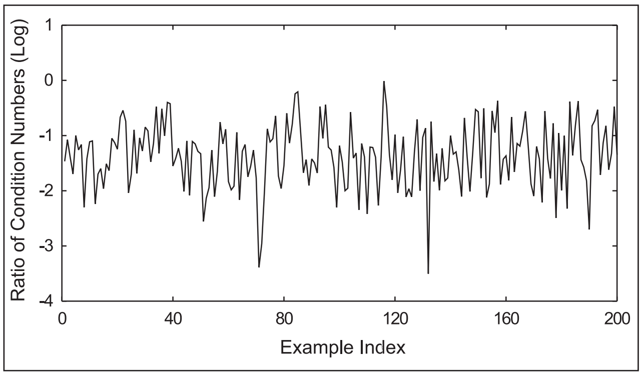

Fig. 13: Logarithmic Ratios of condition numbers of matrices after applying CHOL

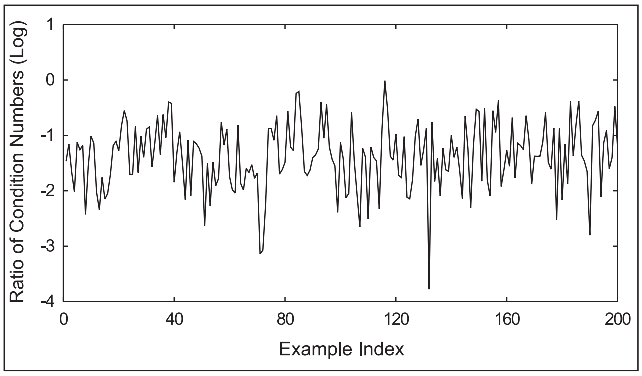

Fig. 14: Logarithmic Ratios of condition numbers of matrices after applying LLL

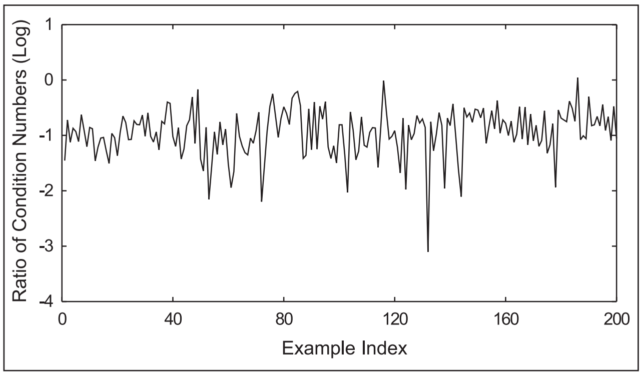

Fig. 15: Logarithmic Ratios of condition numbers of matrices after applying GAUS

where PVV presents the value of $(z-z_0)^T Q_Z^{-1}(z-z_0)$, and the statistical results are listed in Table 2.

Table 2: Statistics of a comparison of the results by GAUS, CHOL and LLL with the $(z-z_0)^T Q_Z^{-1}(z-z_0)$ of the simulated examples

Method	Max	Mean	Min	Stdev
GAUS	3.421	0.9084	–1.272	0.6946
CHOL	1.685	0.6747	–1.272	0.4629
LLL	1.918	0.6942	–1.272	0.4878

Stdev: Standard deviation

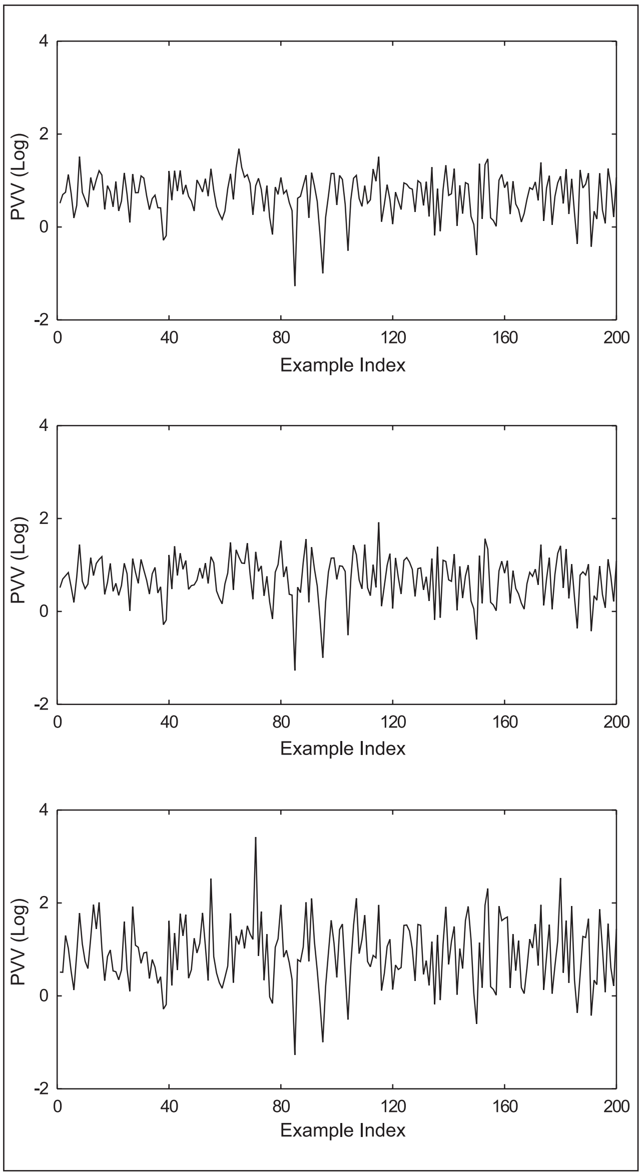

Fig. 16: Logarithmic value of $log\{(z-z_0)^T Q_Z^{-1}(z-z_0)\}$ after applying decorrelation: Upper plot – CHOL; Center plot – LLL; Lower plot – GAUS

From Figure 16 and Table 2, the values of $(z-z_0)^T Q_Z^{-1}(z-z_0)$ show that GAUS is the worst method, and CHOL is slightly better than LLL.

4 Conclusions

The most accurate and precise applications of GPS depend on a successful ambiguity resolution. In order to speed up the ambiguity resolution process, the concept of GPS decorrelation had been initiated by Teunissen (1997). Here we investigated three methods of GPS decorrelation, namely GAUSS, CHOL, and LLL. In order to investigate the performance of different decorrelation methods, we have constructed the matrix Q by using a random simulation method, which ensures the statistical fairness of comparing different methods numerically.

The simulation results have shown that the three methods can be used for decorrelation. But the *inverse integer Cholesky decorrelation* performs almost in the same way as the *LLL algorithm*. It has also been clearly demonstrated that the *LLL algorithm* and the *inverse integer Cholesky decorrelation* perform slightly better than the *integer Gauss decorrelation*. At the same time we found that none of the decorrelation methods can guarantee a significant reduction of the condition number.

Acknowledgments

This work was supported by a grant to Lizhi Lou from the state of Baden-Württemberg. L. T. Liu and C. W. Hu gave some useful comments, B. Kuang improved an early English version of the manuscript. This support is gratefully acknowledged.

References

Hassibi, A. and Boyd, S.: Integer parameter estimation in linear models with application to GPS. IEEE Transactions on Signal Processing 46, pp. 2938–52, 1998.

Hofmann-Wellenhof, B., Lichtenegger, H. and Collins, J.: Global positioning system: theory and practice. Springer-Verlag Wien New York, 1997.

Grafarend, E. W.: Mixed integer/real valued adjustment (IRA) problems: GPS initial cycle ambiguity resolution by means of the LLL algorithm. GPS Solutions 4, pp. 31–44, 2000.

Liu, L. T., Hsu, H. T., Zhu, Y. Z. and Ou, J. K.: A new approach to GPS ambiguity decorrelation. J. Geod. 73, pp. 478–490, 1999.

Teunissen, P. J. G. and Kleusberg, A.: GPS for geodesy. Springer-Verlag Berlin Heildelberg New York, 1998.

Teunissen, P. J. G., de Jonge, P. J. and Tiberius, C. C.: The least squares ambiguity decorrelation adjustment: its performance on short GPS baselines and short observation spans. J. Geod. 71, pp. 589–602, 1997.

Xu, P. L.: Random simulation and GPS decorrelation. J. Geod. 75, pp. 408–423, 2001.

Authors' adress
Dipl. -Ing. (M. Sc.) Lizhi Lou
Department of Surveying and GeoInformatics, Tongji University
Siping Road 1239
200092, Shanghai
PR. China
llz@mail.tongji.edu.cn

Professor Dr.-Ing. Dr.-Ing. E. h. mult. E. W. Grafarend
Department of Geodesy and GeoInformatics, Stuttgart University
Geschwister-Scholl-Str. 24 D
70174 Stuttgart
Germany
grafarend@gis.uni-stuttgart.de