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.

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 \( \sigma_i Q_j \). 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

\[
G_k = \begin{pmatrix}
1 \\
\vdots \\
1
\end{pmatrix} + \begin{pmatrix}
-\frac{q_{i1}}{q_{i0}} & \cdots & \cdots & -\frac{q_{in}}{q_{in}} & 1 \\
\vdots & \ddots & \ddots & \vdots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & 1
\end{pmatrix}
\]

if \( q_{i0} \leq q_{i0} \); or
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 of the variance-covariance matrix \( Q_i \), and then \( P_i \) is decomposed as \( P_i = B B_i^T \), where \( B \) is a certain matrix of full rank. Here \( B \) is reached by using Cholesky decomposition. By letting \( b_1, b_2, ..., b_i \) be row vectors of \( B \), which are linearly independent, the integer Gram-Schmidt orthogonalization process then works as following:

\[
b_i' = b_i - \sum_{j=1}^{i-1} u_{ij} b_j'
\]

\[
u_{ij} = \frac{<b_i', b_j'>}{<b_j', b_j'>}
\]

where \(<,>\) denotes the ordinary inner product, \( b_1', b_2', ..., b_i' \) 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

\[
L = \begin{bmatrix}
1 & u_{i1} & u_{i2} & u_{i3} & \cdots & 1 \\
u_{11} & 1 & u_{12} & u_{13} & \cdots & \\
\vdots & \vdots & \vdots & \vdots & \ddots & \\
u_{i1} & u_{i2} & u_{i3} & \cdots & u_{ii} & 1 \\
\end{bmatrix}
\]

and \( B = LB' \). Therefore,

\[
\|\hat{x} - x_i\|_B^2 = \|B'(\hat{x} - x_i)\|_B^2 = \|B^T L'(\hat{x} - x_i)\|_B^2 = \min
\]

i.e.

\[
(\hat{x} - x_i)^T B B^T L' (\hat{x} - x_i) = \min
\]

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 \( Q_i \) be decomposed as follows:

\[
Q_i = LA L_i^T
\]

where \( \Lambda \) is the diagonal matrix with positive elements \( \lambda_1, \lambda_2, ..., \lambda_n \), for more commonly, the values of \( \lambda_i \) are also stochastic, not sorted as \( \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n \). \( L \) 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 \( Q_i \) 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, \( Z = 1 \) 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

\[ Q_1 = \begin{bmatrix} 53.4 & 38.4 \\ 38.4 & 28.0 \end{bmatrix} \]

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 \( QZ_1 \) are

\[ Z_1 = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}, \quad QZ_1 = \begin{bmatrix} 53.4 & -15.0 \\ -15.0 & 4.6 \end{bmatrix}. \]

Then the iteration approach is applied, the resulting transformation matrix \( Z_2 \) is a unit one with the condition number of \( QZ_1 \) being 160.979. If we sort the matrix with ascending order before the Cholesky decomposition is carried out, we obtain the following result:

\[ Z_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad QZ_2 = \begin{bmatrix} 28.0 & 38.4 \\ 38.4 & 10.4 \end{bmatrix}. \]

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

\[ Z_3 = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}, \quad Z = Z_3 Z_1 = \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix}, \quad QZ_3 = \begin{bmatrix} 28.0 & 10.4 \\ 10.4 & 4.6 \end{bmatrix}. \]

Thus through iteration approach, the resulting transformation matrix \( Z_3 \) presenting a unit matrix, the condition number of \( QZ_2 \) 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

\[ Z_3 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Z = Z_3 Z_1 = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 2 & -1 \end{bmatrix}, \quad QZ_4 = \begin{bmatrix} 4.6 & 1.2 \\ 1.2 & 4.8 \end{bmatrix}. \]

The condition number of \( QZ_4 \) 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.

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

\[ Q_1 = \begin{bmatrix} 1.000 & 0.517 & 0.534 & 0.020 & 0.148 & 0.485 \\ 0.517 & 1.267 & 0.277 & 0.773 & 0.350 & 0.757 \\ 0.534 & 0.277 & 1.285 & 0.685 & 0.335 & 0.399 \\ 0.020 & 0.773 & 0.685 & 2.036 & 1.315 & 1.268 \\ 0.148 & 0.350 & 0.335 & 1.315 & 2.029 & 1.212 \\ 0.485 & 0.757 & 0.399 & 1.268 & 1.212 & 1001.174 \end{bmatrix}. \]

While applying the LLL algorithm, the inverse matrix \( P_i \) of \( Q_i \) should be first computed, and then is decomposed as \( P_i = BB^T \), thus the corresponding matrices \( P_i \) and \( B \) are

\[ P_i = \begin{bmatrix} 2.399 & -1.319 & -1.235 & 1.251 & -0.554 & -0.001 \\ -1.319 & 1.772 & 0.676 & -1.169 & 0.437 & 0.000 \\ -1.235 & 0.676 & 1.592 & -1.020 & 0.372 & 0.000 \\ 1.251 & -1.169 & -1.020 & 1.868 & -0.931 & -0.001 \\ -0.554 & 0.437 & 0.372 & -0.931 & 1.000 & 0.000 \\ -0.001 & 0.000 & 0.000 & -0.001 & 0.000 & 0.001 \end{bmatrix}. \]

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

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

\[ b_1^* = b_1 \Rightarrow \]

\[ u_{11} = [-1.319/2.399] = -1 \Rightarrow b_1^* = [0.697 1.023 0 0 0 0] \Rightarrow \]

\[ u_{11} = [1.080/2.399] = 0 \]

Now that \( u_{11} = 0 \), we stop the iteration of computing \( b_i^* \) and transfer to \( b_i^* \) calculation to obtain

\[ u_{11} = [-1.235/2.399] = -1 \Rightarrow \]

\[ u_{12} = [-0.559/1.534] = 0 \Rightarrow \]

\[ b_1^* = [0.752 -0.003 0.978 0 0 0] \Rightarrow \]

\[ u_{12} = [1.164/2.399] = 0 \]

\[ u_{12} = [0.521/1.534] = 0 \]

Again, since \( u_{11}^* = 0, u_{12}^* = 0 \), the iteration of computing \( b^*_1 \) is ended and \( b^*_1 \) is further calculated.
Now it is found that are not all zero, and \( b_i = [0.708, 0.549, 0.592, 0.919, 0, 0] \), which means that the iteration for \( b_i \) is divergent, leading to the unavailability of other vectors \( b_j \) and \( b_k \).

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 \( N_{iter} \) increase by one automatically. The iteration process will be stopped when \( N_{iter} \) 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 \( N_{iter} = 0 \) and replace \( C_0, Z_0, Q_0 \) with \( C_1, Z, Q \), 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 \( N_{iter} \) denotes the times of iteration. Certainly, in order to obtain the smallest condition number, we sort the matrix \( Q \) 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 \( b_j \) 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
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.

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 $b_j$ without iteration, and then follow the similar way as processing $Q_2$ 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 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.

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 \textit{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 \textit{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 \textit{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 \textit{LLL algorithm} and the integer Gauss decorrelation, respec-

![Fig. 7: Logarithmic condition numbers of 200 examples. Solid line – CHOL; Dotted line – original condition numbers](image1)

![Fig. 8: Logarithmic condition numbers of 200 examples. Solid line – LLL; Dotted line – original condition numbers](image2)

![Fig. 9: Logarithmic condition numbers of 200 examples. Solid line – GAUS; Dotted line – original condition numbers](image3)

![Fig. 10: Logarithmic condition numbers of 200 examples after decorrelation. Solid line – CHOL; Dotted line – GAUS](image4)

![Fig. 11: Logarithmic condition numbers of 200 examples after decorrelation. Solid line – CHOL; Dotted line – LLL.](image5)

![Fig. 12: Logarithmic condition numbers of 200 examples after decorrelation. Solid line – LLL; Dotted line – GAUS](image6)
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

<table>
<thead>
<tr>
<th>Method</th>
<th>Max</th>
<th>Mean</th>
<th>Min</th>
<th>Stdev</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAUS</td>
<td>+0.043</td>
<td>-0.9502</td>
<td>-3.103</td>
<td>0.4319</td>
</tr>
<tr>
<td>CHOL</td>
<td>-0.011</td>
<td>-1.3713</td>
<td>-3.503</td>
<td>0.5916</td>
</tr>
<tr>
<td>LLL</td>
<td>-0.011</td>
<td>-1.3823</td>
<td>-3.778</td>
<td>0.6016</td>
</tr>
</tbody>
</table>

Stdev: Standard deviation

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

$$(\hat{x} - x)^T Q_0^{-1} (\hat{x} - x) = \min,$$  \hspace{1cm} (8)

where $\hat{x}$ is a floating-valued vector solved by least squares, $x$ is the corresponding ambiguities and should be solved, $Q_0$ is the variance-covariance matrix of $\hat{x}$ and is symmetric positive definite. Through the decorrelation of $Q_0$, the corresponding transformation matrix $Z$ is obtained. Thus we calculate:

$$Q_z = Z Q_0 Z^T.$$  \hspace{1cm} (9)

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

$$(z - z_0)^T Q_z^{-1} (z - z_0) = \min,$$  \hspace{1cm} (10)

where $z = Zx$, $z_0 = Z \hat{x}$. With $z = [z_0]^T$, 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,

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Max</th>
<th>Mean</th>
<th>Min</th>
<th>Stdev</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAUS</td>
<td>3.421</td>
<td>0.9084</td>
<td>-1.272</td>
<td>0.6946</td>
</tr>
<tr>
<td>CHOL</td>
<td>1.685</td>
<td>0.6747</td>
<td>-1.272</td>
<td>0.4629</td>
</tr>
<tr>
<td>LLL</td>
<td>1.918</td>
<td>0.6942</td>
<td>-1.272</td>
<td>0.4878</td>
</tr>
</tbody>
</table>

Stdev: Standard deviation
From Figure 16 and Table 2, the values of 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


Authors’ address

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. Grafae
Department of Geodesy and Geoinformatics, Stuttgart University
Geschwister-Scholl-Str. 24 D
70174 Stuttgart
Germany
grafae@gis.uni-stuttgart.de

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