Journal of Global Positioning Systems (2004)
Vol. 3, No. 1-2: 290-295

Journal of Global Positioning Systems
2004
© CPGPS

Convergence of Block Decorrelation Method for the Integer Ambiguity Fix

Samsung Lim* & Binh Quoc Tran**

*School of Surveying & Spatial Information Systems, UNSW, Sydney NSW 2052, Australia
Email: s.lim@unsw.edu.au Tel: +61-2-9385 4505 Fax: +61-2-9313 7493

**Department of Geography, Hanoi University of Sciences, VNU, 332 Nguyen Trai, Thanh Xuan, Hanoi, Vietnam
Email: quocbinh@mail.ru

Received: 15 Nov 2004 / Accepted: 3 Feb 2005

Abstract. Because of the integer-valued nature of carrier phase ambiguities, it is essential to fix the float estimates into integer values in order for high precision DGPS positioning. A decorrelation process is necessary to solve the problem since double-differenced ambiguities are highly correlated in general. In this paper, Block Decorrelation Method (BDM) is presented and tested for its convergence. BDM divides the variance-covariance matrix into four blocks and decorrelates them simultaneously. A number of randomly selected examples show that BDM is comparable to the existing decorrelation algorithm, however its speed of convergence is relatively faster due to the computations performed on small blocks.

Key words: Carrier Phase, Decorrelation, Double-Differenced, Integer Ambiguity, Variance-Covariance

1 Introduction

A float estimate for an initial ambiguity of GPS carrier phase measurements can be obtained by the ordinary least squares technique. However, it is essential to fix the integer value in order to achieve high precision positioning. The problem of integer ambiguities is equivalent to the minimization of the following objective function (Teunissen, 1998):

$$\min S(a) = (\hat{a} - a)^T Q_{\hat{a}}^{-1} (\hat{a} - a) \text{ with } a \in Z^n \tag{1}$$

where $\hat{a}$ is a vector of n float values of double-differenced ambiguities, which is obtained by the least squares estimation with respect to the corresponding variance-covariance matrix $Q_{\hat{a}}$, a is a vector of n unknown integer values of ambiguities, and Z^n is the n-dimensional integer space.

Searching for the minimum for $S(a)$ is difficult because it involves the discrete parameter a. In practice, Equation (1) is usually solved by a discrete search strategy. The idea is that the search space Z^n can be replaced by a small subset or *ambiguity search space* bounded by hyper-ellipsoid:

$$(\hat{a} - a)^T Q_{\hat{a}}^{-1} (\hat{a} - a) \leq \chi^2 \tag{2}$$

where χ^2 is a suitably chosen constant which ensures that the ellipsoid contains at least one integer vector a. The variance-covariance matrix $Q_{\hat{a}}$ affects overall the geometry of elongation and rotation of the search ellipsoid and χ^2 determines its size. Consequently, $Q_{\hat{a}}$ directly affects the effectiveness of the search process. In an ideal case, $Q_{\hat{a}}$ is diagonal and hence a can be obtained simply by rounding the float solution $\hat{a}$ to nearest integer values.

Double-differenced ambiguities are highly correlated, and consequently, $Q_{\hat{a}}$ is far from diagonal. It means that the search ellipsoid is greatly elongated and its principal axes are misaligned with the grids axes (Teunissen 1998, De Jonge 1996). To speed up the search process, $Q_{\hat{a}}$ needs to be "decorrelated". This can be done by using the linear transformation $z = Z^T a$ (Teunissen 1998). Equation (1) is now equivalent to:

$$\min S(z) = (\hat{z} - z)^T Q_{\hat{z}}^{-1} (\hat{z} - z)$$
$$\text{with } \hat{z} = Z^T \hat{a},\ z = Z^T a,\ Q_{\hat{z}} = Z^T Q_{\hat{a}} Z \tag{3}$$

In order to preserve the nature of *a* in *z*, the transformation matrix *Z* must satisfy two conditions:

- C1: Elements of *Z* must be integers;
- C2: Elements of Z^{-1} must be integers too. It is equivalent to $\det[Z] = \pm 1$.

Any permutation matrix or triangular integer matrix with ± 1 in its diagonal meets both conditions C1 and C2. The transformation matrix Z^T will be chosen so that $Q_{\hat{z}}$ become near-diagonal and its condition number *c* become near 1. In general, a diagonal matrix $Q_{\hat{z}}$ with condition number *c*=1 is impossible because of the two conditions C1 and C2 above.

There may exist several methods of devising Z^T. A group of methods based on Gauss transformation algorithm (Strang 1997, Teunissen 1998, Xu 2000) is at hand. To decorrelate the element $\left(Q_{\hat{a}}\right)_{ij}$, Z_k^T can be constructed as an identity matrix except one nonzero element at row *i* and column *j*:

$$\begin{aligned} \left(Z_k^T\right)_{ij} &= -\left[\left(Q_{z,k}\right)_{ij} / \left(Q_{zk}\right)_{jj}\right]^{\text{int}} \\ Q_{z,k} &= Z_{k-1}^T Q_{z,k-1} Z_{k-1},\ \ Q_{z,0} = Q_{\hat{a}},\ \ Z^T = Z_h^T Z_{h-1}^T \ldots . Z_1^T \end{aligned} \tag{4}$$

where the operator $[\cdot]^{\text{int}}$ denotes rounding to the nearest integer. The procedure consists of many steps until the last matrix Z_k^T becomes an identity matrix. The transformation matrix Z^T is a product of matrices Z_k^T, $(k = 1,\ldots,h)$. Although Gauss transformation algorithm usually decorrelates $Q_{\hat{z}}$ well, its convergence is slow because each element of $Q_{\hat{a}}$ should be decorrelated separately.

Another group of methods are based on the factorization of the original matrix $Q_{\hat{a}}$:

$$Q_{\hat{a}} = K^T D K \tag{5}$$

where *D* is a diagonal or a "near diagonal" matrix. Processing float-valued K^T properly, one can obtain integer-valued transformation matrix Z^T satisfying C1 and C2 so that

$$Q_{\hat{z}} = Z^T Q_{\hat{a}} Z \tag{6}$$

where $Q_{\hat{z}}$ is an almost diagonal matrix. The most popular method for factorizing $Q_{\hat{a}}$ is Cholesky's factorization (De Jonge 1996, Xu 2000):

$$Q_{\hat{a}} = LDL^T \quad \text{or} \quad Q_{\hat{a}} = U^T D U \tag{7}$$

where *L* and *U* are lower- and upper-triangular matrices with diagonal elements 1, respectively, and *D* is a diagonal matrix. Note that *D* is an ideal form of $Q_{\hat{a}}$.

The simplest way to construct Z^T is to round each element of *L* or U^T to nearest integers, that is,

$$Z^T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ [L_{21}]^{\text{int}} & 1 & 0 & 0 \\ \ldots. & \ldots. & \ldots. & \ldots \\ [L_{n1}]^{\text{int}} & \ldots & [L_{n,n-1}]^{\text{int}} & 1 \end{bmatrix}^{-1} \quad \text{or}$$

$$Z^T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ [U_{21}]^{\text{int}} & 1 & 0 & 0 \\ \ldots. & \ldots. & \ldots. & \ldots \\ [U_{n1}]^{\text{int}} & \ldots & [U_{n,n-1}]^{\text{int}} & 1 \end{bmatrix}^{-T} \tag{8}$$

For the factorization of $Q_{\hat{a}}$, one can also use Gram-Schmidt orthogonalization process (Grapharend 2000, Xu 2000):

$$Q_{\hat{a}} = V^T V = Z^{-T} O^T O Z^{-1} = Z^{-T} Q_{\hat{Z}} Z^{-1} \tag{9}$$

where O is an almost orthogonal matrix, and consequently, $Q_{\hat{z}}$ is almost diagonal.

Methods based on factorizations are usually faster than methods based on the Gauss transformation. However, due to the fact that they deal with the original matrix $Q_{\hat{a}}$ indirectly via K^T, their results may be relatively worse. Also, some of the methods still experience difficulty with convergence of iteration process (Xu, 2000).

In this paper, a new method for integer decorrelation of variance-covariance matrix $Q_{\hat{a}}$, which is faster than the method based on the Gauss transformation, will be presented. This method deals with the original matrix $Q_{\hat{a}}$ directly, but unlike the Gauss transformation, it divides $Q_{\hat{a}}$ into 4 small blocks and decorrelates elements in each block simultaneously. Therefore, the new method will be named as "Block Decorrelation Method" (BDM) hereafter.

2 Block Decorrelation Method

Consider the following matrix multiplication:

$$Z_k^T Q_{\hat a} Z_k = \begin{bmatrix} I^k & 0 & 0 \\ x_k^T & 1 & 0 \\ 0 & 0 & I^{n-k-1} \end{bmatrix}$$

$$* \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{12}^T & q_{22} & q_{23} \\ q_{13}^T & q_{23}^T & q_{33} \end{bmatrix} * \begin{bmatrix} I^k & x_k & 0 \\ 0 & 1 & 0 \\ 0 & 0 & I^{n-k-1} \end{bmatrix}$$

$$= \begin{bmatrix} q_{11} & s_k & q_{13} \\ s_k^T & p_k & t_k^T \\ q_{13}^T & t_k & q_{33} \end{bmatrix} \quad (10)$$

$s_k = q_{11} x_k + q_{12}$,

$t_k = q_{13}^T x + q_{23}^T$,

$p_k = (x_k^T q_{11} + q_{12}^T) x_k + x_k^T q_{12} + q_{22} = s_k^T x_k + x_k^T q_{12} + q_{22}$

where I^m is an identity matrix of size m; the symmetric positive-definite matrix $Q_{\hat a}$ of size $n*n$ is divided into 3 by 3 blocks because of the pre-multiplication by Z_k^T and post-multiplication by Z_k. Note that p_k is a scalar, but s_k, t_k are vectors. In Equation (10), it is clear that:

If the elements of x_k are integers, then Z_k^T is an admissible transformation matrix that satisfies C1 and C2.

The upper-left q_{11} and lower-right q_{33} are intact after the multiplications. The blocks q_{13} and q_{13}^T do not change too.

Since $Q_{\hat a}$ is symmetric positive-definite, q_{11} is invertible. Consequently the off-diagonal blocks s_k and s_k^T will be zero if x_k satisfies the following.

$$s_k = q_{11} \hat{x}_k + q_{12} = 0 \text{ or } \hat{x}_k = -q_{11}^{-1} q_{12} \quad (11)$$

Due to the condition C1, x_k cannot be a float solution of (11) and s_k, s_k^T cannot be set to zero. But one can expect that s_k, s_k^T will be "nearly zero" if x_k is rounded to the nearest integers:

$$x_k = [\hat{x}_k]^{\text{int}} \quad (12)$$

Assume that index k increases monotonically from *1* to *(m-1)* with a step size 1. By using Equations (10) to (12), the elements of s_k and s_k^T can become close to zero. After the recursive process up to *(m-1)* step, the (*m*m*) upper-left block of the last matrix in Equation (10) will have "decorrelated" elements all over the off-diagonal area.

If another form Z_h^T of transformation matrix is taken, Equations (10) to (12) will become Equation (13) to (15), respectively:

$$Z_h^T Q_{\hat a} Z_h = \begin{bmatrix} I^h & 0 & 0 \\ 0 & 1 & x_h^T \\ 0 & 0 & I^{n-h-1} \end{bmatrix}$$

$$* \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{12}^T & q_{22} & q_{23} \\ q_{13}^T & q_{23}^T & q_{33} \end{bmatrix} * \begin{bmatrix} I^h & 0 & 0 \\ 0 & 1 & 0 \\ 0 & x_h & I^{n-h-1} \end{bmatrix}$$

$$= \begin{bmatrix} q_{11} & t_h & q_{13} \\ t_h^T & p_h & s_h^T \\ q_{13}^T & s_h & q_{33} \end{bmatrix} \quad (13)$$

$s_h = q_{33} x_h + q_{23}^T$, $t_h = q_{13} x_h + q_{12}$,

$p_h = (x_h^T q_{33} + q_{23}) x_h + x_h^T q_{23}^T + q_{22} = s_h^T x_h + x_h^T q_{23}^T + q_{22}$

$$s_h = q_{33} \hat{x}_h + q_{23}^T = 0 \text{ or } \hat{x}_h = -q_{33}^{-1} q_{23}^T \quad (14)$$

$$x_h = [\hat{x}_h]^{\text{int}} \quad (15)$$

Comparing with Equation (10), positions of s and t are exchanged and the role of q_{11} is now given to q_{33} in Equation (13). If the index h decreases from *(n-2)* to m with the step size -1, the block q_{33} will augment from (1*1) to *(n-m-1)*(n-m-1)* matrix in the recursive process of Equations (13), (14) and (15). In addition to that, there is another useful block matrix multiplication:

$$Z_g^T Q_{\hat a} Z_g = \begin{bmatrix} I^g & 0 \\ y_g^T & I^{n-g} \end{bmatrix} * \begin{bmatrix} q_{11} & q_{12} \\ q_{12}^T & q_{22} \end{bmatrix} * \begin{bmatrix} I^g & y_g \\ 0 & I^{n-g} \end{bmatrix}$$

$$= \begin{bmatrix} q_{11} & s_g \\ s_g^T & p_g \end{bmatrix} \quad (16)$$

$s_g = q_{11} y_g + q_{12}$,

$p_g = (y_g^T q_{11} + q_{12}^T) y_g + y_g^T q_{12} + q_{22} = s_g^T y_g + y_g^T q_{12} + q_{22}$

Again, the upper-left block q_{11} does not change after transformation. The off-diagonal blocks s_g, s_g^T will be zero if $\hat{y}_g$ is a solution of:

$$q_{11} \hat{y}_g + q_{12} = 0 \text{ or } \hat{y}_g = -q_{11}^{-1} q_{12} \quad (17)$$

Because of the conditions C1, C2 imposed to Z_g^T, it is only possible to set s_g, s_g^T to near zero by rounding the solution of (17) to nearest integers:

$$y_g = \left[\hat{y}\right]^{\text{int}} \tag{18}$$

Equations (10) to (18) provide an idea of decorrelating $Q_{\hat{a}}$. Dividing $Q_{\hat{a}}$ into four blocks of nearly equal size (see Figure 1), the upper-left block **A** can be decorrelated using Equations (10) to (12). Equations (16) to (18) can be used to make lower-left $\mathbf{B}^T$ and upper-right **B** near zero, and then the lower right block **C** will be decorrelated by Equations (13) to (15). Instead of full-size matrix $Q_{\hat{a}}$, this process deals with smaller (or half-size) blocks **A**, **B**, and **C.** Thus it will speed up the decorrelation process of $Q_{\hat{a}}$.

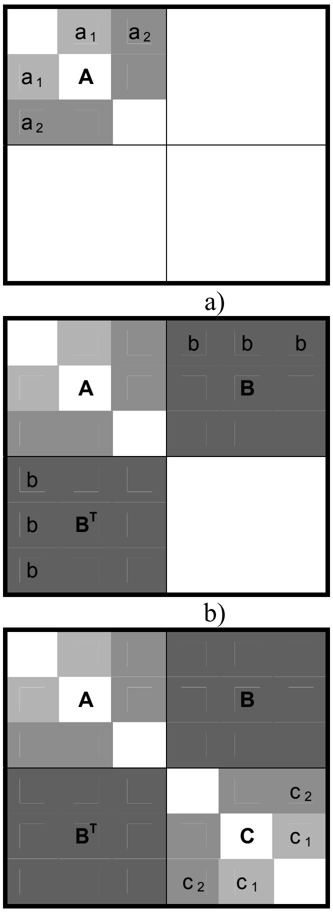

Figure 1. Graphical visualization of block decorrelation method for a 6*6 variance-covariance matrix: (a) - matrix Q_A; (b) - Q_{AB}; (c) - $Q_Z = Q_{ABC}$; a$_1$, a$_2$, c$_1$, c$_2$ show the order of substeps in the corresponding step; elements with gray color have near zero values.

Assume that there are n ambiguities in Equation (1), $m=n/2$ if n is even, and $m=(n+1)/2$ if n is odd number. BDM suggests decorrelating $Q_{\hat{a}}$ within a few numbers of iteration; each consists of the following steps:

Step 1: Permute matrix $Q_{\hat{a}}$ to obtain $\vec{Q}_{\hat{a}}$ so that its first m diagonal elements are minimal and stay in increasing order, i.e.,

$$[\vec{Q}_{\hat{a}}]_{11} \le [\vec{Q}_{\hat{a}}]_{22} \le \ldots \le [\vec{Q}_{\hat{a}}]_{mm} \le [\vec{Q}_{\hat{a}}]_{jj} \ (m < j \le n)$$

This step is necessary to achieve a better decorrelation of block **A** (Figure 1a). Assume that the minimal diagonal element of $Q_{\hat{a}}$ currently stays at row r. To make it the first diagonal element, one can use the permutation:

$$\vec{Q}_{\hat{a}} = H_{A1}^T Q_{\hat{a}} H_{A1} \tag{19}$$

where H_{A1}^T is a permutation matrix, obtained from the identity matrix by exchanging rows 1 and r. Repeating the procedure above for the second, third, … and m^{th}-diagonal elements, it yields to:

$$\begin{aligned} \vec{Q}_{\hat{a}} &= H_{Am}^T H_{A(m-1)}^T \ldots H_{A1}^T Q_{\hat{a}} H_{A1} \ldots H_{A(m-1)} \ldots H_{Am} \\ &= H_A^T Q_{\hat{a}} H_A \end{aligned} \tag{20}$$

Note that H_{Aj}^T and H_A^T satisfy conditions C1 and C2 and are admissible transformation matrices.

Step 2: Apply the decorrelation process to the upper-left block **A** of $\vec{Q}_{\hat{a}}$ using Equations (10), (11) and (12) with index k increasing from 1 to *(m-1)*. The upper-left diagonal block q_{11} in Equation (10) will be gradually augmented by s_k, s_k^T and p_k to fill up **A** as shown in Figure 1a. To speed up the process, the following formula can be used for inverting the augmented matrix q_{11}^k, based on the inverted matrix $[q_{11}^{k-1}]^{-1}$ in previous substep *k-1*:

$$\begin{aligned} \left[q_{11}^k\right]^{-1} &= \begin{bmatrix} q_{11}^{k-1} & s_{k-1} \\ s_{k-1}^T & p_{k-1} \end{bmatrix}^{-1} = \begin{bmatrix} G_{11} & G_{12} \\ G_{12}^T & G_{22} \end{bmatrix} \\ R &= s_{k-1}^T \left[q_{11}^{k-1}\right]^{-1} \\ G_{22} &= (p_{k-1} - R s_{k-1})^{-1}, \\ G_{12} &= -R^T G_{22}, \\ G_{11} &= \left[q_{11}^{k-1}\right]^{-1} - G_{12} R \end{aligned} \tag{21}$$

Thus the process of Step 2 is recursive. It produces a partially decorrelated matrix Q_A:

$$\begin{aligned} Q_A &= Z_A^T \vec{Q}_{\hat{a}} Z_A = Z_A^T H_A^T Q_{\hat{a}} H_A Z_A \\ &= [Z_{A(m-1)}^T \ldots Z_{A2}^T Z_{A1}^T H_A^T] Q_{\hat{a}} [H_A Z_{A1} Z_{A2} \ldots Z_{A(m-1)}] \end{aligned} \tag{22}$$

Step 3: Decorrelate blocks **B** and $\mathbf{B}^T$ of Q_A (Figure 1b) using Equations (16) to (18). The size of the block q_{11} in Equation (16) is *m*m*. To get its inverse, Equations (21) and $[q_{11}^{m-1}]^{-1}$ from Step 2 can be used. As a result, Q_{AB} is obtained by:

$$Q_{AB} = Z_B^T Q_A Z_B \tag{23}$$

Step 4: Permute Q_{AB} to yield $\overleftarrow{Q}_{AB}$ so that the diagonal elements of its lower-right block **C** (Figure 1c) stays in decreasing order, i.e.,

$$[\overleftarrow{Q}_{AB}]_{nn} \le [\overleftarrow{Q}_{AB}]_{(n-1)(n-1)} \le \ldots \le [\overleftarrow{Q}_{AB}]_{(m+1)(m+1)} .$$

Analogous to the Step 1, this procedure can be done by the matrix H_C^T:

$$\overleftarrow{Q}_{AB} = H_C^T Q_{AB} H_C \tag{24}$$

The upper-left block **A** remains unchanged in this step.

Step 5: Decorrelate the last block **C** in $\overleftarrow{Q}_{AB}$ using Equations (13) to (15) with index *h* decreasing from *(n-2)* to *m*. The lower-right block q_{33} in Equation (13) will be gradually augmented by s_h, s_h^T and p_h to fill up **C** as shown in Figure 1. The formula for inverting the augmented matrix is similar to Equation (21):

$$\begin{gathered}
\left[q_{33}^{h}\right]^{-1} = \begin{bmatrix} p_{h-1} & s_{h-1}^T \\ s_{h-1} & q_{33}^{h-1} \end{bmatrix}^{-1} = \begin{bmatrix} G_{11} & G_{12} \\ G_{12}^T & G_{22} \end{bmatrix} \\
R = \left[q_{33}^{h-1}\right]^{-1} s_{h-1} \\
G_{11} = (p_{h-1} - s_{h-1}^T R)^{-1} , \\
G_{12} = -G_{11} R^T , \\
G_{22} = \left[q_{33}^{h-1}\right]^{-1} - R G_{12}
\end{gathered} \tag{25}$$

The first m elements of the vector q_{12} and the first m rows of q_{13} in Equation (13) are near zero because the block **B** is already decorrelated in Step 3. Thus the first m elements of the vector $t_h = q_{13} x_h + q_{12}$ in Equation (13) also become near zero. Hence, elements of $\mathbf{B}^T$ and **B** remain near-zero, even though they can change at Step 5. It provides a decorrelated matrix $Q_{\hat{z}i}$:

$$Q_{\hat{z}i} = Q_{ABC} = Z_C^T \overleftarrow{Q}_{AB} Z_C \tag{26}$$

Combining Equations (20), (22), (23), (24) and (26) yields:

$$\begin{aligned}
Q_{\hat{z}i} &= Z_C^T H_C^T Z_B^T Z_A^T H_A^T Q_{\hat{a}} H_A Z_A Z_B H_C Z_C \\
&= [Z_{Cm}^T \ldots Z_{C(n-2)}^T H_C^T Z_B^T Z_{A(m-1)}^T \ldots Z_{A1}^T H_A^T] Q_{\hat{a}} \\
&\quad * [H_A Z_{A1} \ldots Z_{A(m-1)} Z_B H_C Z_{C(n-2)} \ldots Z_{Cm}]
\end{aligned} \tag{27}$$

The transformation matrix Z_i^T in current i^{th}-iteration is obtained by Equation (27):

$$Z_i^T = Z_{Cm}^T \ldots Z_{C(n-2)}^T H_C^T Z_B^T Z_{A(m-1)}^T \ldots Z_{A1}^T H_A^T \tag{28}$$

Since Z_{Ch}^T, Z_B^T, Z_{Ak}^T $(h = n-2,\ldots,m;\ k = 0,\ldots,m-1)$ and H_A^T, H_C^T satisfy conditions C1 and C2, it is readily seen that Z_i^T satisfies these conditions too.

The procedure described in steps 1 to 5 can be repeated until Z_i^T becomes an identity matrix i.e., until no further decorrelation of $Q_{\hat{a}}$ can be obtained. The final transformation matrix Z^T is then computed by:

$$Z^T = Z_M^T Z_{M-1}^T \ldots Z_1^T = \prod_{i=M}^{1} Z_i^T \tag{29}$$

where *M* denotes the number of iteration. An estimation shows that each iteration without explicit calculation of Z_i^T requires about *$7n^3/8$* multiplication. Therefore, the decorrelation process requires about *$7Mn^3/8$* multiplication.

3 Numerical Example

In this section, BDM is used for decorrelating two sets of ambiguities. Ambiguities of these numerical examples are highly correlated. To quantify the decorrelation of $Q_{\hat{a}}$, two kinds of measures are used:

- The correlation coefficients;
- The condition number ***c*** which is the ratio of the largest and the smallest singular value of variance-covariance matrix.

If *e* denotes the elongation of the ellipsoid in Equation (2) then:

$$c = e^2 = R_{\max}^2 / R_{\min}^2 \tag{30}$$

where $R_{\max}$ and $R_{\min}$ are the largest and smallest axes of the ellipsoid, respectively.

Table 1 shows the main characteristics of the decorrelation process: the condition numbers ca and cz, the smallest $\rho_{\min}$ and the largest $\rho_{\max}$ correlation coefficients of original matrix $Q_{\hat{a}}$ and decorrelated matrix $Q_{\hat{z}}$; the number of iteration circles M. To compare with the existing methods, Table 1 also shows the results obtained by United Decorrelation Method (Liu, 1999) and Gauss transformation.

NN	Description	$Q_{\hat{a}}$		$Q_{\hat{z}}$		# of iter.
		c	ρ_{min}	c	ρ_{max}	
1	6 ambiguities, BDM	$2.2*10^7$	0.8442	12.2	0.3990	6
2	6 ambiguities, United decorrelation method	$2.2*10^7$	0.8442	12.2	0.3990	6
3	6 ambiguities, Gauss transformation	$2.2*10^7$	0.8442	11.7	0.4275	6
4	12 ambiguities, BDM	$2.1*10^5$	0.9448	24.8	0.5056	9
5	12 ambiguities, Gauss transformation	$2.1*10^5$	0.9448	54.5	0.4778	10

Table 1. Parameters of the correlation processes: c, ρ_{min}, ρ_{max} denotes condition numbers, minimal and maximal absolute values of correlation coefficients.

The test results show that highly correlated ambiguities are significantly decorrelated: the condition number and the corresponding elongation of the search ellipsoid drastically reduced from 10^5-10^7 to less than 100, and the average correlation coefficients diminished more than 2 times. In a relatively small number of ambiguities, all three methods give almost identical results, but for a larger number of ambiguities, BDM produces a better result. The MatLab implementation of the algorithm proves that BDM is 70-120% faster than Gauss transformation depending on the original matrix $Q_{\hat{a}}$ in the cases of 12 ambiguities.

4 Conclusions

The decorrelation process plays an important role in resolving integer ambiguities of GPS carrier phase measurements. In this paper, a new method for the decorrelation is presented. The method is based on dividing the variance-covariance matrix into 4 small blocks and decorrelating them simultaneously. The decorrelation of each block is processed recursively so that the result of the previous step is not affected by the next step. This algorithm reduces the dimension of the original variance-covariance matrix and therefore increases the speed of the decorrelation process. The proposed algorithm provides comparable or better result than that of the existing algorithm.

References

De Jonge, P.J. and Tiberius, C.C.J.M. (1996), ***The LAMBDA method for integer ambiguity estimation: implementation aspects,*** Delft Geodetic Computing Center LGR series, No. 12.

Grapharend, E.W. (2000), ***Mixed integer-real valued adjustment (IRA) problem: GPS initial cycle ambiguity resolution by means of the LLL algorithm,*** GPS Solution, No. 4, pp. 31-44.

Liu, L. T. et al. (1999), ***A new approach to GPS ambiguity decorrelation,*** Journal of Geodesy, No 73, pp. 478-490.

Strang, G. (1997), ***Linear algebra, geodesy and GPS,*** Wellesley-Cambridge Press.

Teusnissen, P.J.G. and Kleusberg, A. (1998), ***GPS for Geodesy,*** Springer.

Xu, P. (2000), ***Random simulation and GPS decorrelation,*** Journal of Geodesy, No 75, pp. 408-423.