

A Progress Report on Numerical Solutions of Least Squares Adjustment in GNU Project Gama

A. Čepeck, J. Pytel

GNU project Gama for adjustment of geodetic networks is presented. Numerical solution of Least Squares Adjustment in the project is based on Singular Value Decomposition (SVD) and General Orthogonalization Algorithm (GSO). Both algorithms enable solution of singular systems resulting from adjustment of free geodetic networks.

Keywords: geodesy, least square adjustment, local geodetic network.

1 Introduction

Project Gama for adjustment of geodetic networks was started at the department of mapping and cartography, Faculty of Civil Engineering, Czech TU Prague, in 1998. At first it was planned to be only a local project with the main goal of demonstrating to students the power of object programming and at the same time being a free independent tool for comparing of adjustment results from various sources. The Gama project received the official status of GNU software in 2001, and now consists of a C++ library (including small C++ matrix/vector template library `gmatvec`) and two programs `gama-local` and `gama-g3`, which correspond to two development branches of the project.

The stable branch of the Gama project consists of the command line program `gama-local` for adjustment of three-dimensional geodetic networks in a local coordinates system (platform independent Qt based GUI `roci-local` is also available). The new development branch of the project (`gama-g3`) aims to adjust the geodetic networks in a global geocentric system. The stable branch (`gama-local`) enables common adjustment of possibly correlated horizontal directions and distances, horizontal angles, slope distances and zenith angles, height differences, observed coordinates (used in sequential adjustment, etc.) and observed coordinate differences (vectors). Although such an adjustment model has now been made obsolete by global positioning systems, it can still serve as an educational tool for demonstrating adjustment procedures to students and as a starting platform for the developing a new branch of the project (`gama-g3`).

Numerical solution of least squares adjustment in geodesy is most commonly based on the solution of normal equations. As the Gama project was also meant to be a comparison tool, it was desirable to use a different method, and Singular Value Decomposition (SVD) was implemented as the main numerical algorithm. As an testing alternative, Gama implements another algorithm from the family of orthogonal decompositions based on Gram-Schmidt orthogonalization (GSO). Practical experience with both algorithms is discussed. In the Gama project, the geodetic input data are described in Extensible Markup Language (XML). The primary motivation for using XML was to define structured input data for adjustment of a local geodetic network. The most important feature of XML is probably the ease of defining a grammar for user data

(a class of XML documents) that consequently can be validated even independently of our applications. One of the goals of the Gama project is to build a collection of model geodetic networks described in XML. The lack of reliable testing data was one of major obstacles when testing the implementation of the numerical solution of the geodetic network adjustment.

2 Adjustment and analysis of observations

Geodesy as a scientific discipline studies the geometry of the Earth or, from the practical point of view, the positioning of objects located on the Earth's surface or in zones relatively close to it. The input information consists of geodetic *observations*.

The spectrum of observation types dealt with by geodesy is very wide, and ranges from classical astro-geodetic observations (astronomical longitude and latitude, variations and position of the Earth's pole), measurements of geophysical quantities (gravity acceleration and its local anomalies), through traditional geometric observables like directions, angles and distances to photogrammetric measurements of historical monuments. However, the main importance in geodesy today is given to satellite global positioning systems (first of all NAVSTAR GPS and other complementary systems like DORIS or GLONASS).

The key role in processing geodetic data belongs to the sphere of applied statistics in geodesy, traditionally called *adjustment of observations*. The processing of geodetic observations is determined by the choice of an appropriate mathematical model, which can be symbolically expressed as

$$f(\mathbf{c}, \mathbf{x}, \mathbf{l}) = \mathbf{0}, \quad (1)$$

where f is a vector of functions describing the relations between constants \mathbf{c} , unknown parameters \mathbf{x} and observed quantities \mathbf{l} . Corresponding to the three components of this model are three mathematical spaces: parameter, observation and model space [1].

The three basic components of the mathematical model (1) are depicted in Fig. 1, where \mathbf{A} , \mathbf{B} , \mathbf{G} and \mathbf{H} are the matrices of corresponding linearized relations (values of constants \mathbf{c} are not estimated in geodesy and we can consider them to be a part of model space). Models can be direct, indirect or

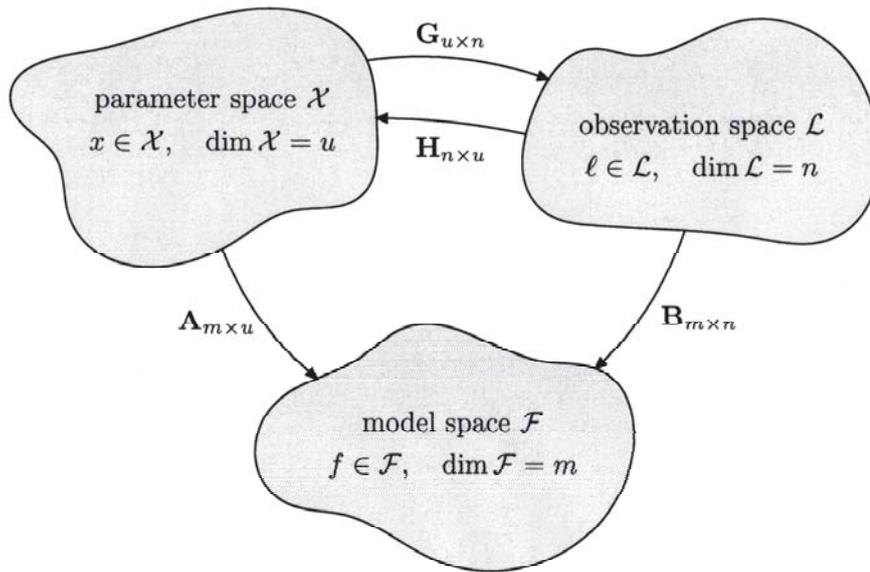


Fig. 1: Linear relations between parameter, observation and model spaces

implicit; linear or nonlinear; can occur individually or in combinations

- model explicit in x : $x = g(l), \quad x = Gl + v,$
- model explicit in l : $l = h(x), \quad l = Hx + v,$
- implicit model: $f(x, l) = 0, \quad Ax + Gl + v = 0.$

3 Least Squares and singular systems

When adjusting geodetic observations we are relatively often faced with models leading to singular sets of linear equations. Typically these are models without fixed points, i.e., no points with fixed coordinates are given, or the number of fixed points is not sufficient (*free networks*, see [6] for more information).

Let us take as an example the local network with observed directions and distances from Fig. 2. The relationship between the unknown adjusted coordinates and observations can be expressed after linearization as the *project equations*

$$Ax - l = v, \tag{2}$$

where A is design matrix, x vector of unknowns, l vector of reduced observations and v vector of residuals (misclosure vector).

In geodesy, the number of observations is always higher than the number of unknowns. Project equations (2) thus represent an overdetermined system, and matrix A has more rows than columns. Least Squares is the basic method used in geodesy for observation adjustment. It gives us the unique solution x of system (2) that minimizes the Euclidean norm of the residual vector

$$\min \sqrt{v'v}. \tag{3}$$

A method commonly used for solving projects equations (2) (model explicit in observations) is based on *normal equations*

$$\begin{aligned} N &= A'A, & n &= A'l, \\ x &= N^{-1}n. \end{aligned} \tag{4}$$

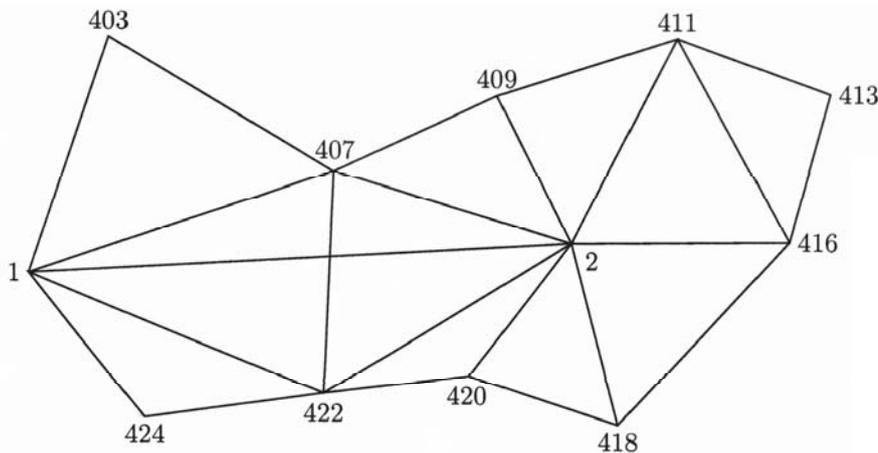


Fig. 2: Example of a local geodetic free network

Apart from unknown vector \mathbf{x} (and residuals \mathbf{v}) we are always interested in geodesy in estimates of the precision of the adjusted quantities, in geodetic practice represented by the variance-covariance matrix of adjusted unknowns \mathbf{C}_{xx} and adjusted observations \mathbf{C}_{ll}

$$\mathbf{C}_{xx} = m_0' \mathbf{N}^{-1}, \quad (5)$$

$$\mathbf{C}_{ll} = \mathbf{A} \mathbf{C}_{xx} \mathbf{A}'. \quad (6)$$

The geometric shape of our adjusted network is defined by observed directions (or angles) and directions. If we fixed the coordinates of two or more points the network shape would necessarily be distorted. Normal equations would lead to an adjustment solution in which the residuals would be dependent on the coordinates of the fixed points. This way we would degrade our observations in cases when the coordinates of the network points are either unknown or known with lower precision.

On the other hand if we consider the coordinates of all points to be free, the corresponding matrix \mathbf{N} is inevitably singular; the columns of matrix \mathbf{A} are linearly dependent (the network can float freely in the coordinate system) and normal equation matrix \mathbf{N} is positive-semidefinite

$$\mathbf{p}' \mathbf{N} \mathbf{p} \geq 0, \quad \mathbf{p} \neq \mathbf{0}.$$

To get a unique solution we have to define additional constraints regularizing the system, preferably without deforming of the network shape. In geodetic practice we most often meet the following approaches:

- A singular system is regularized by introducing pseudo-observations, typically with huge weights, that play a similar role as a set of constraint equations.
- An explicit system of constraint equations is defined to make the given system regular

$$\mathbf{C} \mathbf{x} = \mathbf{C}. \quad (7)$$

Normal equations then become

$$\begin{pmatrix} \mathbf{N} & \mathbf{C}' \\ \mathbf{C} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{\Lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{l} \\ \mathbf{c} \end{pmatrix}, \quad (8)$$

where $\mathbf{\Lambda}$ is the vector of Lagrange multipliers. In this case matrix \mathbf{C} is problem dependent and needs to be known explicitly in advance.

- The Euclidean norm of a certain subset of unknown parameters vector \mathbf{x} is minimized

$$\min_{x_i} \sqrt{\sum x_i^2}, \quad i \in \mathcal{O}. \quad (9)$$

The set of indices \mathcal{O} can contain all elements, but more often only selected elements of \mathbf{x} .

In the case of a plane geodetic free network we can geometrically interpret the last constraint (9) as follows. By minimizing of the Euclidean norm of the residual vector (3) the shape and scale (if at least one distance is available) of the adjusted network together with the covariances of the adjusted observations are uniquely defined. The second additional constraint (9) then defines the localization of the network in the coordinate system. Apart from the adjusted network shape we simultaneously define its shift and rotation in the coordinate system.

Another equivalent interpretation is that constraint (9) defines the particular solution of (2) in which the trace of

the variance-covariance submatrix corresponding to indices $i \in \mathcal{O}$ is minimal.

4 Normal equations and numerical stability

The numerical solution of the adjustment of observed quantities based on normal equations can be numerically unstable and in certain cases we should prefer other numerical algorithms that directly solve the project equations (2). A possible source of troubles are the normal equations themselves, or more precisely the condition number of normal equations. Let us restrict our discussion here to the simple case when matrix \mathbf{A} does not contain linearly dependent columns and matrix \mathbf{N} is positive-definite.

The condition number of matrix \mathbf{A} is defined as

$$\kappa(\mathbf{A}) = \sqrt{\frac{\lambda(\mathbf{A}'\mathbf{A})_{\max}}{\lambda(\mathbf{A}'\mathbf{A})_{\min}}}, \quad (10)$$

where $\lambda(\mathbf{A}'\mathbf{A})_{*}$ denotes the maximal and minimal eigenvalue of matrix $\mathbf{A}'\mathbf{A}$. If we solve a linear set of equations then its condition number represents the minimal upper estimate of ratio of relative error of \mathbf{x} and the relative error of the right hand side \mathbf{l} .

From equation (10) it directly follows that the condition number of normal equation matrix \mathbf{N} is the square of the condition number of the project equation matrix \mathbf{A}

$$\kappa(\mathbf{N}) = (\kappa(\mathbf{A}))^2. \quad (11)$$

We can say that when solving poorly conditioned normal equations we lose twice as many of correct decimal digits in a solution \mathbf{x} as in any direct solution of project equations.

Probably the most important class of algorithms for direct solution of project equations (2) is the family of orthogonal decomposition algorithms. Apart from other goals, GNU project Gama has been planned to be a kind of *benchmark*, i.e., a tool for checking adjustment results from other software products. For this reason it was desirable to have the adjustment based on a different numerical method other than the traditional solution of normal equations, and Singular Value Decomposition (SVD) was implemented as the main numerical algorithm. As an alternative, another orthogonal decomposition adjustment algorithm GSO (based on Gram-Schmidt orthogonalization) is also available. We give a brief description of both algorithms in the following section.

5 Gram-Schmidt orthogonalization

The Gram-Schmidt orthogonal decomposition is an algorithm for computing factorization

$$\mathbf{A} = \mathbf{Q}\mathbf{R}, \quad \mathbf{Q}'\mathbf{Q} = \mathbf{1}, \quad (12)$$

where \mathbf{Q} is the orthogonal matrix and \mathbf{R} is the upper triangular matrix. Matrix \mathbf{R} here is identical to the upper triangular matrix of the Cholesky decomposition of normal equations

$$\mathbf{N} = \mathbf{A}'\mathbf{A} = \mathbf{R}'\mathbf{Q}'\mathbf{Q}\mathbf{R} = \mathbf{R}'\mathbf{R}. \quad (13)$$

Gram-Schmidt orthogonalization is a very straightforward and relatively simple algorithm that can be implemented in several variants differing in the order in which the vectors are orthogonalized. The following three algorithms are adopted from [2, p. 300–301].

Algorithm 1.1

[Modified Gram–Schmidt (MGS) row version]

```

for k = 1, 2, ..., n
    q̂k := ak(k); rkk := (q̂kT q̂k)1/2;
    qk := q̂k/rkk;
    for i = k + 1, ..., n
        rki := qkT ai(k); ai(k+1) := ai(k) - rki qk;
    end
end
    
```

Algorithm 1.2

[Modified Gram–Schmidt (MGS) column version]

```

for k = 1, 2, ..., n
    for i = 1, 2, ..., k - 1
        rik := qiT ak(i); ak(i+1) := ak(i) - rik qi;
    end
    q̂k := ak(k); rkk := (q̂kT q̂k)1/2;
    qk := q̂k/rkk;
end
    
```

Algorithms 1.3

[Classical Gram–Schmidt (CGS)]

```

for k = 1, 2, ..., n
    for i = 1, 2, ..., k - 1
        rik := qiT ak;
    end
    q̂k := ak - ∑i=1k-1 rik qi
    rkk := (q̂kT q̂k)1/2; qk := q̂k/rkk;
end
    
```

It should not be forgotten that the variant known as *Classical Gram–Schmidt* has very poor numerical properties in that there is typically a severe loss of orthogonality among the computed q_i . A rearrangement of the calculation, known as *Modified Gram–Schmidt*, yields a much sounder computational procedure [3, p. 230–232].

5.1 Generalized orthogonalization algorithm (GSO)

The generalized orthogonalization algorithm (GSO), a method based on Gram–Schmidt orthogonalization, for numerical solution of various adjustment models in geodesy was elaborated by František Charamza [4, 5]. GSO was implemented in GNU Gama to conserve this rarely used but interesting method, and to offer an alternative numerical algorithm to SVD (which we expected to give better numerical results for numerically unstable systems).

Algorithm GSO operates on a block matrix structure

$$\begin{pmatrix} M_1 & M_2 \\ M_3 & M_4 \end{pmatrix} \rightarrow \begin{pmatrix} Q_1 & Q_2 \\ Q_3 & Q_4 \end{pmatrix}, \tag{14}$$

where the transition from M to Q is defined by the equations

$$Q_1^T Q_1 = \mathbf{1}, \tag{15}$$

$$M_1 = Q_1 R, \tag{16}$$

$$Q_1 = M_1 R^{-1}, Q_2 = M_2 - Q_1 Q_1^T M_2, \tag{17}$$

$$Q_3 = M_3 R^{-1}, Q_4 = M_4 - Q_3 Q_1^T M_2, \tag{18}$$

and R is the upper triangular matrix.

Algorithm MGS is applied to block matrix M so that the column dot products are computed only for submatrices (M_1, M_2) , the projections $r_{ki} q_k$ are computed for full columns of M and the whole process is terminated after of all columns of submatrix $(M_1, M_3)'$ have been processed. This step is called the *first orthogonalization* in algorithm GSO.

Let us take as an example the linear system of project equations (2)

$$Ax - l = v$$

and apply algorithm GSO to the block matrix

$$\begin{pmatrix} A & -l \\ \mathbf{1} & \mathbf{0} \end{pmatrix} \rightarrow \begin{pmatrix} Q & v \\ R^{-1} & x \end{pmatrix}.$$

The result is directly the vector of unknown parameters x and the vector of residuals v . The cofactors (weight coefficients) of the adjusted parameters $q_{x_i x_j}$ are available as the dot products of rows i and j of submatrix R^{-1} , the cofactors of the adjusted observations $q_{l_m l_n}$ are computed as the dot products of rows m and n of submatrix Q and the mixed cofactors $q_{x_i l_n}$ similarly as the dot products of the i -th row of R^{-1} and the n -th row of matrix Q .

5.2 Algorithm GSO and singular systems

Let us suppose now that project equations matrix A contains r linearly independent columns and the remaining d linearly dependent columns. Without loss of generality we can assume that the linearly dependent columns are located in the right part of matrix A . We denote linearly independent columns A_1 , linearly dependent columns A_2 and the matrix of their linear combinations α

$$A = (A_1, A_2), \quad A_2 = A_1 \alpha, \quad x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}. \tag{19}$$

Now we can rewrite the project equations as

$$v = A_1 x_1 + A_2 x_2 - l = A_1 (x_1 + \alpha x_2) - l = A_1 \tilde{x} - l. \tag{20}$$

As matrix A_1 does not contain linearly dependent columns, a unique solution \tilde{x} of (20) exists that minimizes the Euclidean norm of v .

If we know matrix α and vector \tilde{x} then any solution x of

$$\tilde{x} = x_1 + \alpha x_2 = (\mathbf{1}, \alpha) x \tag{21}$$

is at the same time the Least Square solution of (20) with the same vector of residuals v .

If we apply algorithm GSO to the matrix

$$M^I = \left(\begin{array}{c|c} M_1^I & M_2^I \\ \hline M_3^I & M_4^I \end{array} \right) = \left(\begin{array}{c|cc} A_1 & A_2 & -l \\ \hline \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{1} & \mathbf{0} \end{array} \right) \tag{22}$$

we receive a block matrix

$$Q^I = \left(\begin{array}{c|cc} Q & \mathbf{0} & v \\ \hline R_1^{-1} & -\alpha & \tilde{x} \\ \hline \mathbf{0} & \mathbf{1} & \mathbf{0} \end{array} \right). \tag{23}$$

In the case of singular systems in GSO we have the *first orthogonalization*, which defines a particular solution in which the unknown parameters corresponding to the linearly dependent columns of A are set to zero. From CGS it emerges directly that matrix α is the matrix of the linear combinations from (19). The cofactors are computed the same way as in the case of regular systems.

When computing GSO numerically we naturally do not obtain exactly zero vectors on the positions of the (almost) linearly dependent columns. We declare to be linearly dependent those columns of A whose norms drop below a given tolerance. During the first orthogonalization we set to zero corresponding subvectors in the area of A_2 . These values can be considered as *random noise* that adds no information to the whole solution.

The result of the first orthogonalization are first of all the vector of the residuals and cofactors of the adjusted observations. It now remains to determine the vector of the unknown parameters \mathbf{x} that satisfies condition (9) and its cofactors (weight coefficients). This step of GSO is called *second orthogonalization*.

By solving the system of linear equations

$$\begin{pmatrix} -\alpha \\ \mathbf{1} \end{pmatrix} \mathbf{x}_2 = \begin{pmatrix} \tilde{\mathbf{x}} \\ \mathbf{0} \end{pmatrix} \quad (24)$$

we get, according to (21), a vector \mathbf{x} with the minimal norm. If we select from (24) only certain rows, we obtain the solution minimizing the corresponding subvector. This system can naturally be solved using GSO.

If we need the cofactors of the adjusted unknowns, as is the standard case with geodetic applications, we have to process during the second orthogonalization the whole lower submatrix that resulted from the first orthogonalization step.

$$M^{\text{II}} = (M_1^{\text{II}} \mid M_1^{\text{II}}) = \begin{pmatrix} -\alpha & R_1^{-1} & \tilde{\mathbf{x}} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} \end{pmatrix}. \quad (25)$$

During the first orthogonalization, the linearly dependent columns in M_1 are identified and are explicitly zeroed. The result of the first orthogonalization is a particular solution in which the unknowns corresponding to the linearly dependent columns are all set to zero. Naturally their cofactors are zero as well.

During the second orthogonalization step only the submatrix (Q_3, Q_4) is influenced and the orthogonalization process is carried out as follows

- Gram-Schmidt orthogonalization runs only through columns corresponding to the linearly dependent columns of M_1 as if they were numbered 1, 2, ..., d , where d is the nullity of M_1
- dot products are computed only for indices $i \in \mathcal{O}$ from the regularization condition

$$\min_{x_i} \sqrt{\sum x_i^2}, \quad i \in \mathcal{O}.$$

Linearly dependent columns are zeroed during the second orthogonalization even in the region of submatrix (Q_3, Q_4) . The cofactors after the second orthogonalization are computed in the same way as in the case of regular systems.

6 Singular Value Decomposition (SVD)

For any real $m \times n$ matrix A , $m \geq n$, there exists the singular value decomposition

$$A = U W V' \quad (26)$$

$$U'U = \mathbf{1}, \quad VV' = V'V = \mathbf{1},$$

where U is an $m \times n$ matrix with orthogonal columns, W is a diagonal matrix $n \times n$ with nonnegative elements and V is a square orthogonal matrix $n \times n$, (this variant is referred to as the *thin SVD* [3]).

Matrix W is uniquely determined up to the permutation of its diagonal elements. The diagonal elements w_i are called singular values of matrix A . Their squares are eigenvalues of $n \times n$ matrix $A'A$. Thus, the condition number of matrix A can be computed as the ratio of the maximal and minimal singular value.

$$\kappa(A) = \frac{w_{\max}}{w_{\min}}. \quad (27)$$

With singular decomposition we can directly express the vector of unknown parameters \mathbf{x} from the project equations

$$A\mathbf{x} = \mathbf{l}, \quad \mathbf{x} = V W^{-1} U' \mathbf{l}, \quad W^{-1} = \text{diag}(1/w_i). \quad (28)$$

If matrix A has more rows than columns (overdetermined system), then the Euclidean norm of the residual vector

$$\mathbf{v} = A\mathbf{x} - \mathbf{l}$$

is minimal and vector \mathbf{x} is the Least Squares solution to the project equations (2).

For a matrix A with linearly dependent columns d singular values are zero (d is the dimension of null space of A). Singular value decomposition explicitly constructs the orthonormal vector basis of the null space and the range of A . The columns of matrix U corresponding to nonzero singular values w_i form the orthonormal base of the range of A . Similarly, the columns of matrix V corresponding to nonzero singular values form the orthonormal basis of the null space of A .

$$\mathcal{N}_A = \{ \mathbf{x} \mid A\mathbf{x} = \mathbf{0}, \quad \mathbf{x} \in \mathcal{R}^n \}$$

$$\mathcal{R}_A = \{ \mathbf{y} \mid \mathbf{y} = A\mathbf{x}, \quad \mathbf{x} \in \mathcal{R}^n \}.$$

In the case of rank deficient systems, we set into the diagonal of inverse matrix W^{-1} zeros instead of reciprocals for elements corresponding to linearly dependent columns A

$$W^{-1} = \text{diag} \begin{cases} 1/w_i & \text{for } w_i > 0 \\ 0 & \text{for } w_i = 0. \end{cases} \quad (29)$$

The resulting particular solution \mathbf{x} minimizes both the Euclidean norm of the residuals and at the same time the norm of unknown parameters \mathbf{x} .

The rather surprising replacement of reciprocal $1/0 \equiv \infty$ by zero can be explained as follows. The solution vector \mathbf{x} of the overdetermined system

$$A\mathbf{x} = \mathbf{l}$$

can be expressed as the linear combination of the columns of matrix V

$$\mathbf{x} = \sum_{i=1}^n \left(\frac{1}{w_i} U_{(i)} \mathbf{l} \right) V_{(i)}. \quad (30)$$

The coefficients in parentheses are dot products of columns U and right hand side l multiplied by the reciprocal value of the singular value. The zero singular values correspond to the linearly dependent columns of matrix A that add no other information to the given system. Setting corresponding diagonal elements of matrix W^{-1} to zeros is equivalent to eliminating the linearly dependent columns from matrix A .

With matrix W^{-1} defined according to (29), the cofactors computed the same way for regular and singular systems

$$Q_{xx} = N^{-1} = (A'A)^{-1} = (VWU'UWV')^{-1} = VW^{-1}W^{-T}V', \quad (31)$$

$$Q_{ll} = AQ_{xx}A' = (UWV')(VW^{-1}W^{-T}V')(VWU') = UU', \quad (32)$$

$$Q_{lx} = AQ_{xx} = UWV'VW^{-1}W^{-T}V' = UW^{-1}V'. \quad (33)$$

The cofactors (weight coefficients) for the adjusted parameters, observations and mixed cofactors are computed, similarly as in the case of GSO, as the dot products of the rows of matrices U and V ; multiplied by the diagonal elements of W^{-1} in the case of cofactors of x .

6.1 Algorithm SVD and singular systems

What now remains is to show how to compute the particular solution that minimizes only a given subset of subvector x according to the second regularization condition (9). We compose an overdetermined system of linear equations

$$\psi c + x = \hat{x}, \quad (34)$$

where the columns of matrix ψ are vectors of null space basis

$$\psi = (V_{(i_1)}, V_{(i_2)}, \dots, V_{(i_n)}), \quad w_{i_n} = 0,$$

and c is the vector of the coefficients of the linear combination of null space basis vectors that, when added to vector x , minimizes the selected subvector of unknown parameters \hat{x} (here they act as residuals).

From a comparison of (34) with equations (24) and (25) it is obvious that for computing \hat{x} we can use the second orthogonalization of algorithm GSO. If the GSO second orthogonalization is applied to matrix V from the singular decomposition

$$M^{\text{II}} = (M_1^{\text{II}} | M_2^{\text{II}}) = (\psi | \Psi), \quad (35)$$

we obtain matrix \hat{V} . If we now replace the singular value decomposition matrix V by matrix \hat{V} , we can compute vector \hat{x} and all cofactors according to the same formulas (30–33) as in the case of standard SVD solution x .

7 Network adjustment in GNU Gama

Gama was started in 1998 as a local educational project, mainly to demonstrate to our students the power and capability of object programming (the project is written in C++), and at the same time to show some alternatives to traditional approaches to numerical solutions of Least Squares adjustments based on normal equations. Project Gama was released under the terms of the GNU General Public license, and in 2001 received the official status of GNU software.

Numerical solution of geodetic network adjustment in Gama is based on an abstract C++ class and currently two derived classes are available implementing algorithms SVD and GSO. SVD is the primary algorithm used in Gama (one of our long term goals is to add more numerical solutions,

namely solutions exploiting the sparse structure of project equations). From this perspective, algorithm GSO was implemented in Gama only as an testing alternative, both for comparing numerical results and for testing the hierarchy of the adjustment classes in practice.

It is generally agreed that a bad implementation of GSO can produce disastrous results. For example, during the first orthogonalization step of GSO we set to zeros the unknown parameters corresponding to linearly dependent columns. In the case of a free geodetic network adjustment these are the coordinates of certain points – the whole network is *pinned* on these points and clearly, if close points are selected the regularization is unstable. The order of the columns in the orthogonalization is important.

From practical experience we know that vector norms in the GSO orthogonalization process generally tend to decrease. As GSO is just an alternative algorithm in Gama and its performance is not a crucial point, we implemented it with full pivoting, i.e., in each orthogonalization cycle the vector with the maximal norm is selected as a pivot (with this modification GSO is about twice as slow as SVD for large networks).

Singular Value Decomposition is a very robust method for dealing with systems that are either singular or numerically close to singular. Even with *full pivoting* we had expected GSO to prove to be inferior to SVD, at least in cases with ill-conditioned matrices. Surprisingly, with all the real geodetic networks that we have available this was not the case. Apart from real data, we used series of randomly generated three-dimensional networks for testing.

Our implementation of SVD is based on a classical algorithm published by Golub and Reinsch [7] (the ALGOL procedure SVD). The decomposition is constructed in two phases. It starts with the Householder reduction to bidiagonal form, followed by diagonalization. Contrary to our expectations, SVD as used in Gama has not proved to give numerically better results and in some cases it has even lost convergence in the diagonalization phase.

A simple and tempting explanation that comes first to mind would be that the SVD implementation in Gama is somehow wrong. After all testing and revisions this does not seem to be the point. A possible explanation might be given by the following quotation from [3]

... Finally, we mention Jacobi's method ... for the SVD. This transformation method repeatedly multiplies A on the right by elementary orthogonal matrices (Jacobi rotations) until A converges to $U\Sigma$; the product of the Jacobi rotations is V . Jacobi is slower than any of the above transformation methods (it can be made to run within about twice the time of QR ...) but has the useful property that for certain A it can deliver the tiny singular values, and their singular vectors, much more accurately than any of the above methods provided that it is properly implemented...

Surely to have more numerical methods implemented in Gama would be helpful, for example the above mentioned Jacobi method for SVD.

A practical problem during testing of the adjustment methods in Gama was the relative shortage of reliable observation data and their adjustment results for testing. To enable easy comparison with other softwares we made a description of geodetic networks in XML (we use DTD for the definition

of the formal syntax of our structured data). Conversion from a well defined data format into XML is a relatively simple task, but processing of XML is not a trivial task and cannot be done without an XML parser. In the GNU Gama project we use the XML parser `expat` by James Clark, see <http://expat.sourceforge.net/>. We believe that XML is the best data format for description and exchange of structured data in the Gama project. One of the goals of our project is to compile a free collection of geodetic networks described in XML.

References

- [1] Vaníček, P., Krakiwsky, E. J.: “Geodesy: The Concepts”, North-Holland, 2nd. ed. (1986), ISBN 0-444-87775-4.
- [2] Björck, : “Numerics of Gram-Schmidt Orthogonalization”. *Linear Algebra and Its Applications* Vol. **197**, **198**, (1994), p. 297–316.
- [3] Golub, G. H., Van Loan, C. F.: “Matrix Computations”, 3rd edition, The John Hopkins University Press 1996, ISBN 0-8018-5413-X.
- [4] Charamza, F.: “GSO – An Algorithm for Solving Least-Squares Problems with Possibly Rank Deficient Matrices”. *Optimization of Design and Computation of Control Networks*, Budapest, Akadémiai Kiadó, 1979.
- [5] Charamza, F.: “An Algorithm for the Minimum-Length Least-Squares Solution of a Set of Observation Equations”. *Studia geoph. et geod.*, Vol **22**, (1978), p. 129–139.
- [6] Koch, K. R.: “Parameter Estimation and Hypothesis Testing in Linear Models”. 2nd ed. Springer-Verlag (1999), ISBN 3-540-65257-4.
- [7] Golub, G. H., Reinsch, C.: “Singular Value Decomposition and Least Squares Solutions”. *Numer. Math.* 14, (1970), p. 403–420, *Handbook for Auto. Comp.*, Vol. II – Linear Algebra, (1971), p. 134–151.
- [8] Demmel, J. W.: “Singular Value Decomposition”. In Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, editors. *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*. SIAM, Philadelphia, 2000, <http://www.cs.utk.edu/~dongarra/etemplates/book.html>

Prof. Ing. Aleš Čepěk, CSc.
phone: +420 224 354 647
e-mail: cepek@fsv.cvut.cz

Department of Mapping and Cartography

Ing. Jan Pytel
phone: +420 224 354 644
e-mail: pytel@gama.fsv.cvut.cz

Czech Technical University in Prague
Faculty of Civil Engineering
Thákurova 7
166 29 Prague, Czech Republic