Quantification of the dependence of the results on several network adjustment applications

Stéphane Durand, Michael Lösler, Mark Jones, Paul-Henri Cattin, Sébastien Guillaume, Laurent Morel

5th Joint International Symposium on Deformation Monitoring (JISDM), 20-22 June 2022, Valencia, Spain
DOI: http://doi.org/10.4995/JISDM2022.2022.13671

Quantification of the dependence of the results on several network adjustment applications

Stéphane Durand[1], Michael Lösler[2], Mark Jones[3], Paul-Henri Cattin[4], Sébastien Guillaume[4], Laurent Morel[1]

[1] Geomatics and Land Law Lab (Cnam/GeF - EA 4630), 1 boulevard Pythagore, F-72000, Le Mans, France, (stephane.durand@lecnam.net; laurent.morel@lecnam.net)
[2] Laboratory for Industrial Metrology, Frankfurt University of Applied Sciences, Nibelungenplatz 1, D-60318 Frankfurt am Main, Germany, (michael.loesler@fb1.fra-uas.de)
[3] Geodetic Metrology, BE Dept., European Organization for Nuclear Research (CERN), 1211 Genève, Switzerland, (mark.jones@cern.ch)
[4] School of Management and Engineering Vaud (HEIG-VD), Institut Géomatique, insit – Institut d'ingénierie du territoire, Route de Cheseaux 1, 1400 Yverdon-les-Bains, Switzerland, (paul-henri.cattin@heig-vd.ch; sebastien.guillaume@heig-vd.ch)

Key words: *network adjustment; functional model discrepancies; generated measurements*

ABSTRACT

The dependency of the results from a network adjustment on the application used is investigated. For that purpose, the results obtained by each tested application on several sets of simulated measurements are compared. In each simulation, only one parameter varies. We first present our comparison methodology and the method that was used to add Gaussian-like errors to theoretical measurements. We then apply it to study the impact of the side length of the network and of the ellipsoidal height difference among points in the network for several network adjustment applications: Columbus, CoMeT, Geolab, JAG3D, LGC, Move3, Star*Net and Trinet+.

I. INTRODUCTION

Adjusting geodetic networks is one of the most important tasks in the daily business of modern geodesists. In applied geodesy, such networks are the basis for planning landscape reorganization in, for example, urban development, operating and extending mechanical engineering facilities like particle accelerators, or the monitoring of structural facilities such as dams, bridges, or radio telescopes.

Many network adjustment applications are available today. Each user is free to use the application that best suits their own needs and preferences. In case of software replacement, for economic or practical reasons, of sending raw data to someone else, who is using different software, or for control purposes, the question of the dependency of the results on the software used (so called software effect) arises.

In the last decade, several studies have investigated this problem, using field measurements from given networks. The results from different adjustment applications are then compared, in terms of values and standard deviations for all estimated coordinates and relevant quality parameters (Lösler and Bähr, 2010; Schwieger *et al.*, 2010; Herrmann *et al.*, 2015).

In Durand *et al.* (2020) a comparison methodology to study the impact of the software on the results for applications using the Gauss-Markov model was proposed, based on the comparison of the results obtained by each application, on the same simulated measurements. The main advantage of using generated measurements is that it allows the controlled alteration of one parameter at a time, between sets of measurements: shape or size of the network, refraction coefficient value, a priori accuracy, and so on. Thus, the generated measurements allow us to investigate the impact of each parameter.

In Durand *et al.* (2020), the comparison methodology was used to investigate the impact of the size of the network on the compatibility among the network adjustment applications developed in the authors institutions, *i.e.*, CoMeT, LGC, and Trinet+.

The main objectives of this contribution are to include more network adjustment applications in the comparisons, and to show that our methodology is relevant to better understand the impact of a particular parameter on the discrepancies between the applications.

In the first part of this contribution, we present our comparison methodology, based on the adjustment of a set of networks with simulated measurements. In the second part, we focus on the method that was used to add Gaussian-like errors to theoretical measurements fulfilling the following properties:

1) Adjusted coordinates are equal to the theoretical values and the computation is achieved in only one iteration step.

2) The estimated variance factor of unit weight is equal to the empirical one.
3) For a selected significance level, standardized residuals are not rejected in Baarda's test.

In the last part, we apply our comparison methodology to study the impact of two parameters on the results of the adjustment process: the side length of the network and the ellipsoidal height of the unknown point in the network.

We present and discuss the results obtained after processing the same simulated measurements with the following network adjustment applications. These are all using three-dimensional mathematical functional models, with an ellipsoidal approach, except for the LGC application where, for practical reasons, a spherical approach has been used in this instance:

1. *Columbus*, from Best-Fit computing, version 4.6.2.41 (http://bestfit.com/;
2. *CoMeT* (Compensation de Mesures Topographiques), from the Cnam/GeF, version 2017.02.14 (http://comet.esgt.cnam.fr);
3. *Geolab*, from Bitwise Ideas Inc., version 2021.1.1 (https://www.geolabsolutions.com);
4. *JAG3D* (Java Applied Geodesy 3D), using the ellipsoidal approach, version 20211123 (https://software.applied-geodesy.org);
5. *LGC* (Logiciel Général de Compensation), from the CERN, version 2.03.01 (https://move3software.com);
6. *Move3*, from Sweco, version 4.5.1 (https://move3software.com);
7. *Star**Net, from MicroSurvey, version 10.0.15.974 (https://www.microsurvey.com/products/starnet);
8. *Trinet+*, from the Heig-Vd, version 7.2.

II. Test Methodology

A. *Using simulated test networks*

As in Durand *et al.* (2020), our test methodology is based on simply shaped networks, all with only one variable parameter. Each network is made up of two points (S1 and S2) with known coordinates and one point M with coordinates to be estimated. In order to build these networks, we defined a point P0 as the barycentre of all our networks, with ETRS89 Cartesian and ellipsoidal coordinates as defined in Table 1, and with no geoidal undulation with respect to the IAG GRS80 ellipsoid. To ensure compatibility with the LGC application, this P0 point has the same definition as the CERN P0 point used in the LGC application and linked to the CERN Coordinate System (CCS).

In the local astronomical system of point P0, given a side length d, the horizontal coordinates for the points of each network are computed to form an equilateral triangle, with point P0 as barycentre, point M being located north of point P0, and point S1 and S2 being located east and west of point P0, respectively, as illustrated in Figure 1.

Table 1. ETRS89 Cartesian and ellipsoidal coordinates of point P0, the barycentre of all the test networks

	ETRS89 Cartesian Coordinates	Ellipsoidal coordinates (w.r.t. IAG GRS80)
X [m]	4395400.3638	--
Y [m]	465785.0567	--
Z [m]	4583458.2260	--
Latitude [grad]	--	51.3692
Longitude [grad]	--	6.72124
Ellipsoidal height [m]	--	433.65921

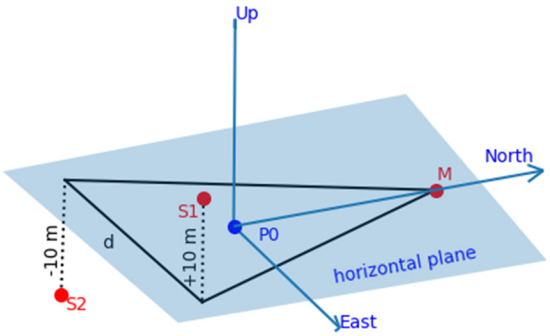

Figure 1. Common design of the test networks with side length d, in the local astronomical system of P0.

Such a simple network without peculiarities is sufficient to investigate the software effect. An application that yields different results for such a simple network design will also give different results in complex networks. However, the effect will be easier to be detected in a simple network design than in a complex one.

The local vertical coordinates of points S1, S2, and M were fixed to be respectively +10 m, −10 m, and 0 m. These local coordinates were used to compute ETRS89 geodetic Cartesian coordinates for all points in the test networks (with no deflection of the vertical for point P0 with respect to the IAG GRS80 ellipsoid).

These ETRS89 Cartesian coordinates were used to compute simulated observations between points in the networks, as the sum of a theoretical value and an error value. The theoretical value of the observation was computed using the three-dimensional mathematical functional models of the CoMeT application, with an ellipsoidal datum (IAG GRS80 ellipsoid, deflection of the vertical of zero for all points). The error value was computed in a particular way explained in part III. As illustrated in Table 2, conventional measurements, *i.e.*, horizontal direction, zenith angle, and slope distance, are made from points S1 and S2 to point M and only horizontal direction observations are included from S1 to S2 and from S2 to S1.

Table 2. Type of observations between points in the test networks

	From S1 to:	From S2 to:
Horizontal direction	M, S2	M, S1
Zenith angle	M	M
Slope Distance	M	M

Therefore, the adjustment of each network entails the estimation, using least squares, of the coordinates of point M as well as the orientation unknowns for the instrument stations at S1 and S2.

B. *Functional and stochastic parameters*

Even if all the applications involved in this study make use of three-dimensional mathematical functional models, with an ellipsoidal datum, some differences may exist both in the functional and stochastic implementations. Table 3 summarizes the main parameters used in this work to both simulate and process the measurements, in order to avoid any unwanted discrepancies in the results. We used a convergence criterion of 0.1 mm, as this is the smallest selectable value in some applications.

Table 3. Relevant functional and stochastic parameters for both simulation and processing

Parameter	Value
Refraction coefficient	None
Standard deviation (angle measurements)	0.3 mgrad
Standard deviation (slope distance)	5 mm
Convergence criterion	0.1 mm

The distance dependent part of the a priori uncertainty for slope distances is left out because their processing differs in each application. Let a, in meters, and b, in parts per million (ppm), denote the constant and distance-dependant parts respectively of the standard deviation for a slope distance D between the station and the target. As indicated in Shih (2013), there exists two interpretations of electronic distance meter accuracy specifications, based on different assumptions: the "additive" method, in which the standard deviation σ is computed from Equation 1, and the "propagated" one, which uses Equation 2.

$$\sigma = a + b \times 10^{-6} \times D \quad (1)$$

$$\sigma = \sqrt{a^2 + (b \times 10^{-6} \times D)^2} \quad (2)$$

Depending on the applied model and especially for long distances, different results are obtained. This effect must be taken into account when comparing adjustment applications. Table 4 indicates for each application involved in this study which approaches are implemented and used by default.

The refraction correction to zenith angle measurements is also not applied since the method used to compute this correction differs for each program or does not exist (Durand *et al.*, 2020). Let V_{mes} denote the zenith angle observation, V_{cor} the corrected value from the refraction effect, and k the coefficient of refraction.

In the LGC application, no refraction corrections are applied to zenith angle measurements - the application was developed primarily for networks located in underground tunnels, or buildings.

Table 4. Implemented stochastic models to obtain the a priori uncertainty of a slope distance

Software	Additive approach (Equation 1)	Propagated approach (Equation 2)
Columbus	Yes	No
CoMeT	Yes	Yes (default)
Geolab	No	Yes
JAG3D	No	Yes
LGC	Yes	No
Move3	Yes	No
Star*Net	Yes (default)	Yes
Trinet+	Yes	No

As indicated in their user manuals, the Columbus and CoMeT applications apply the following formula (Eq. 3):

$$V_{cor} = V_{mes} + \frac{k \times D}{2 \times R} \quad (3)$$

In the Columbus application, R corresponds to the sum of the radius of the Earth and the ellipsoidal height at the station. In the CoMeT application, R corresponds to the semi-major axis of the selected ellipsoid.

In the Trinet+ and JAG3D applications, the correction is computed using a slightly different formula (Eq. 4):

$$V_{cor} = V_{mes} + \frac{k \times D_h}{2 \times R} \quad (4)$$

where D_h denotes the horizontal distance between points. In the JAG3D application, the Earth radius is approximated using the latitude of the barycentre P0. In the Trinet+ application, R can be selected in the range $[6000, 7000]$ km.

The documentation of the Geolab and Move3 applications do not clearly indicate which formula is used to compute the refraction correction. Nevertheless, in these applications it is possible to output a measurement value including the refraction correction, and potentially deduce the refraction correction formula. From tests that we have conducted, it seems that the Geolab and Move3 applications use a formula very close to the one in Equation 3.

The user manual of the Star*Net application indicates that the refraction error is computed following (Bomford, 1971). In practice, using the same notations as above, this corresponds to the formula (Eq. 5):

$$V_{cor} = V_{mes} + \frac{k \times D_h}{R} \quad (5)$$

In this representation, the largest difference is the factor 2, which means that in the Star*Net application, the refraction coefficient, k, may not be directly comparable with those used in Equations 3 and 4. This

effect of the refraction handling must be taken into account when comparing adjustment applications.

For all applications involved in this work, known and initial coordinates for all points are entered directly as ETRS89 Cartesian or ellipsoidal (w.r.t. IAG GRS80 ellipsoid) coordinates. There are only a few exceptions. The JAG3D application does not process global Earth-fixed coordinates but supports local tangent plane coordinates. For that reason, the ETRS89 Cartesian geodetic coordinates are converted into local coordinates in the local tangent plane system of point P0 w.r.t. the IAG GRS80 ellipsoid. In the Trinet+ application, observation equations are expressed in terms of topocentric coordinates. The topocentre as well as the points in the network have to be defined using CH1903+ Cartesian coordinates and the values of the deflection of the vertical with respect to the Bessel 1841 reference ellipsoid. Thus, the ETRS89 Cartesian coordinates were converted to CH1903+ Cartesian coordinates. Starting from zero value deflection of the vertical on each point relative to the IAG GRS80 reference ellipsoid, the deflection of the vertical relative to the Bessel 1841 reference ellipsoid was computed. In the case of geodetic calculations in the LGC application, coordinates have to be entered in a CERN 2D+1 system derived from the CERN Coordinate System (CCS). Thus, the ETRS89 Cartesian coordinates of each point in each network were transformed into the Cartesian CCS system. Subsequently, the Z-coordinates were converted to ellipsoidal heights using the CERN Spherical vertical datum (Durand *et al.*, 2020) - to avoid the application of a geoid model.

III. Generation of random errors

Simulated measurements are the sum of theoretical measurements and error values, which could be equal to zero, or randomly selected from a Gaussian distribution. In this work, we have developed a particular method for computing the error values, hereafter detailed.

Suppose an unknown vector $X = (x_1, \dots, x_m)^T$ is to be estimated from observation vector $L = (l_1, \dots, l_n)^T$, where $n \geq m$. In classical least squares estimation, they are related by (Eq. 6):

$$L = f(X) + V, V \sim \mathcal{N}(0, s_0^2 Q) \quad (6)$$

where f denotes the functional relation between measurements and parameters, V the error vector, s_0^2 the a priori variance factor of unit weight and Q the cofactor matrix of the observations. Equation 6 assumes the absence of systematic and gross errors. Only random errors are considered.

As indicated for example in (Caspary and Rüeger, 2000), in an iterative least squares method, Equation 6 is linearized by approximation to a first-order Taylor series expansion (Eq. 7):

$$B = L - f(X_0) = A(X - X_0) + V, V \sim \mathcal{N}(0, s_0^2 Q) \quad (7)$$

where $B = L - f(X_0)$ is the difference between the measurements and their estimated values using X_0, the initial values of the parameters. Solution of Equation 6 is computed using an iterative process on the least squares solution of Equation 7, until the adjusted estimate of Equation 7 is sufficiently close to the initial value X_0 used in the last iteration step. Using matrix Q to define the norm, the least squares solution $\hat{X}$ to Equation 7 is expressed as (Eq. 8):

$$\hat{X} - X_0 = (A^T Q^{-1} A)^{-1} A^T Q^{-1} B \quad (8)$$

At the end of the iterative process, we obtain the estimated least squares solution of Equation 7. It is thus possible to compute the vector of residuals (Eq. 9):

$$\hat{V} = B - A(\hat{X} - X_0) \quad (9)$$

And the estimated variance factor of unit weight (Eq. 10):

$$\hat{s}_0^2 = \frac{\hat{V}^T Q^{-1} \hat{V}}{n - m} \quad (10)$$

In the process of simulating measurements for a particular network, the theoretical coordinates of all points, the different measurements between the points introduced and the a priori standard deviations of the measurements are known. Let X_{th} denote the vector containing the theoretical coordinates of all points in a network, as well as initial or randomly selected values for the orientation unknowns. Let V_s denote a simulated error vector. It is possible to directly use Equation 6 to compute simulated values for the measurements (Eq. 11):

$$L_s = f(X_{th}) + V_s \quad (11)$$

The simulated error vector V_s could be obtained using several methods: a null vector (no error), randomly selected values, randomly selected values with respect to a Gaussian distribution with the a priori standard deviation of each measurement, to cite but a few.

In this study, we used a particular method to compute the error vector V_s. We wanted to compute an error vector which fulfilled the following conditions:

(i) Vector V_s is not null, nor unique;
(ii) If the initial coordinates for all points are the theoretical ones (which means $X_O = X_{th}$), then the adjusted coordinates are equal to the theoretical ones (which means $\hat{X} = X_{th}$) and the iterative process is achieved in only one iteration step;
(iii) The global model test used as a first step in outlier detection as described in (Caspary and

Rüeger, 2000) is not rejected regardless of the selected significance level;

(iv) For a selected significant level α, standardized residuals are not rejected in Baarda's test.

Now let V_s denote an error vector fulfilling these four conditions. From Equation 11, the measurements vector to process is $L_s = f(X_{th}) + V_s$. As the theoretical coordinates are used as initial values in the adjustment process, we have, for the first iteration step, as indicated in Equation 7, (Eq. 12):

$$B = L_s - f(X_{th}) = V_s \tag{12}$$

From Equation 8, the least squares solution for the first iteration step yields (Eq. 13):

$$\hat{X} - X_{th} = (A^T Q^{-1} A)^{-1} A^T Q^{-1} V_s \tag{13}$$

Condition (ii) means that $\hat{X} - X_{th} = 0$. The first consequence is that the vector of residuals corresponds to the simulated error vector (Eq. 14):

$$\hat{V} = (L_s - f(X_{th})) - A\left(\hat{X} - X_{th}\right) = V_s \tag{14}$$

The second consequence is that vector V_s lies in the kernel of the linear map with corresponding matrix (Eq. 15):

$$H = (A^T Q^{-1} A)^{-1} A^T Q^{-1} \tag{15}$$

As matrix Q is symmetric positive definite, it is possible to find an upper triangular matrix R such that $Q^{-1} = R^T R$, and to define matrix K as (Eq. 16):

$$K = (A^T Q^{-1} A)^{-1} A^T R^T \tag{16}$$

Thus, if $V_s \in \ker(H)$ then $y = RV_s \in \ker(K)$. As K is a matrix of size $n \times m$ and of rank m, $\ker(K)$ is of dimension $n - m$. Let β be an orthogonal basis of $\ker(K)$. By definition, for each vector $z \in \mathbb{R}^{n-m}$, $y = \beta z$ lies in $\ker(K)$. This gives us a practical way to compute an error vector that fulfils conditions (i) and (ii). We just have to randomly select a non-zero vector $z \in \mathbb{R}^{n-m}$, to compute an orthogonal basis β of $\ker(K)$ (from the singular value decomposition (SVD) of matrix K for example), and to compute (Eq. 17):

$$V_s = R^{-1} \beta z \tag{17}$$

The third condition is related to the global model test commonly used as a first step in outlier detection. As explained, for example in (Caspary and Rüeger, 2000), its aim is to verify that the error vector is compatible with a statistical distribution of the errors according to the $\mathcal{N}(0, s_0^2 Q)$ normal law. The statistic used for this test is, according to the definitions in Equations 9 and 10, (Eq. 18):

$$T = \frac{\hat{V}^T Q^{-1} \hat{V}}{s_0^2} = (n-m)\frac{\hat{s}_0^2}{s_0^2} \sim \chi^2_{n-m} \tag{18}$$

After selecting a significance level α for the test, and according to the χ^2_{n-m} distribution, one has to verify that the observed test statistic T is located in the acceptance region. As the central value of the χ^2 distribution is equal to $(n-m)$, if the observed test statistic is equal to $(n-m)$, it is in the acceptance region, whatever the selected significant level.

Thus, we have to compute an error vector V_s in such a way that the corresponding observed test statistic yields $T = n - m$.

As vector V_s fulfils conditions (i) and (ii), it is of the form $V_s = R^{-1} \beta z$ with $z \in \mathbb{R}^{n-m}$ and corresponds to the vector of residuals. The observed test statistic is (Eq. 19):

$$T = \frac{V_s^T Q^{-1} V_s}{s_0^2} = \frac{z^T z}{s_0^2} \tag{19}$$

To fulfil condition (iii), we just have to compute a vector $z \in \mathbb{R}^{n-m}$ in such a way that (Eq. 20):

$$z^T z = (n-m) s_0^2 \tag{20}$$

The fourth condition is related to the test strategy proposed by Baarda, which is commonly used to detect outliers when the residuals are normally distributed (Baarda, 1968; Caspary and Rüeger, 2000). This test assumes that the variance factor s_0^2 is known and uses the standardized residual as the test statistic (Eq. 21):

$$\forall i = 1, \dots, n, \quad \widehat{w}_i = \frac{\hat{v}_i}{s_0 \sqrt{q_{v_i v_i}}} \tag{21}$$

In Equation 21, $\hat{v}_i$ denotes the i-th element of the vector of residuals $\hat{V}$ and $q_{v_i v_i}$ is the corresponding element (i,i) of the cofactor matrix $Q_{\hat{V}}$ of $\hat{V}$ defined by (Eq. 22):

$$Q_{\hat{V}} = Q - A(A^T Q^{-1} A)^{-1} A^T \tag{22}$$

Selecting a significance level α, it is possible to compute the critical value γ of the test statistic using the normal distribution $\mathcal{N}(0,1)$. No gross error is detected if the absolute value of the standardized residual $\widehat{w}_i$ is less than γ.

Let G denote the vector of length n in which each entry G_i corresponds to (Eq. 23):

$$\forall i = 1, \dots, n, \quad G_i = s_0 \sqrt{q_{v_i v_i}} \tag{23}$$

The acceptance region of Baarda's test could be written in the following matrix formulation (Eq. 24):

$$U = \{u \in \mathbb{R}^n / -G\gamma \leq u \leq G\gamma\} \tag{24}$$

To fulfil condition (iv), the simulated error vector V_s must be included in U. As it also fulfils conditions (i) and (ii), it is computed as $V_s = R^{-1}\beta z$, where $z \in \mathbb{R}^{n-m}$, (see Equation 17). Thus, the acceptance region of Baarda's test could be written as (Eq. 25):

$$U = \{u \in \mathbb{R}^{n-m} / -G\gamma \leq R^{-1}\beta u \leq G\gamma\} \quad (25)$$

In practice, finding a simulated error vector that fulfils conditions (i) to (iv) is related to the quadratic optimization problem (Eq. 26):

$$\begin{matrix} max_{u\in\mathbb{R}^{n-m}} & \varphi(u) = u^T u \\ s.t. & -G\gamma \leq R^{-1}\beta u \leq G\gamma \end{matrix} \quad (26)$$

In the general case, finding a solution of Equation 26 is a non-deterministic polynomial-time hard problem (Pardalos and Vavasis, 1991) and called a nonconvex quadratic programming problem. It can be resolved using several global optimization methods for nonconvex QP problems (Pardalos, 1991).

In practice, we do not really have to find the solution of problem (Equation 26). We just need to find $\breve{u} \in \mathbb{R}^{n-m}$ fulfilling conditions (i), (ii) and (iv). It is thus more suitable to use vertex enumeration methods like the ones described in (Avis and Fukuda, 1992). For the polyhedron defined in Equation 25, a point $u \in U$ is a vertex of U if, and only if, it is the unique solution to a subset of $n-m$ inequalities, solved as equations. The enumeration process can be stopped when we get a vertex $\breve{u} \in \mathbb{R}^{n-m}$ such that $\varphi(\breve{u}) \geq (n-m)s_0^2$. It is also possible to continue the enumeration process in order to select another vertex with the same properties, to simulate a kind of random selection. If the enumeration process finishes, we get the solution to the QP problem in Equation 26.

Suppose now that we get a vertex $\breve{u} \in \mathbb{R}^{n-m}$ such that $\varphi(\breve{u}) \geq (n-m)s_0^2$. Let S denote the sphere defined by (Eq. 27):

$$S = \{u \in \mathbb{R}^{n-m} / u^T u = (n-m)s_0^2\} \quad (27)$$

We may then compute a vector $\hat{z}$ that fulfils conditions (i) to (iv) as the orthogonal projection of $\breve{u}$ onto the sphere defined in Equation 27. If for the solution $\hat{u}$ of problem (Equation 26) we have $\varphi(\hat{u}) < (n-m)s_0^2$, then it is not possible to find a point fulfilling the four conditions and $\hat{z}$ is the point which is closest to fulfilling condition (iv).

In our particular case, it is very easy to find the vertices of the polyhedron. As β is an orthogonal basis of $\ker(K)$, $\beta^T\beta$ corresponds to the identity matrix, and by setting $\kappa = \beta^T R G\gamma$, Equation 25 is simplified by (Eq. 28):

$$U = \{u \in \mathbb{R}^{n-m} / -\kappa \leq u \leq \kappa\} \quad (28)$$

Polyhedron U corresponds to a $(n-m)$-dimensional rectangle, whose 2^{n-m} vertices are easy to find. For each vertex $\breve{u}$, the quantity $\varphi(\breve{u})$ is the same and corresponds to $\varphi(\breve{u}) = \kappa^T\kappa$.

In contrast to randomly selected values, the benefit of the described procedure is to control the synthetically generated observations w.r.t. the effect under investigation.

IV. Results and discussion

A. *Criteria for comparisons*

For each network under consideration, measurements were simulated using the functional mathematical models of the CoMeT application, and were processed by each network adjustment application, using the same parameters as summarized in Table 3. We obtain adjusted ETRS89 coordinates for the unknown point M, directly or after transformation (JAG3D, Trinet+, LGC). Then the local coordinates of the estimated point M in the local astronomical system of the theoretical point M are computed, allowing the computation of coordinate deviations in the local horizontal plane and the local vertical.

We also obtain quality parameters, such as residuals and estimated variance factors. The estimated variance factor can be seen as a global metric of the agreement between all the measurements in a network and the three-dimensional mathematical models used in a particular application. The method used to generate the error vector associated with simulated measurements ensures that when processed with an application that uses similar functional models, the estimated variance factor is close to one. This facilitates the comparisons as the closer the estimated variance factor is to one, the greater the similarity of CoMeT's functional models to those of the tested application.

B. *Impact of the side length of the network*

This test is similar to the one proposed in Durand *et al.* (2020) and uses the same simulated measurements, which are available on a dedicated webpage[1]. All networks share the same shape (equilateral triangle in the horizontal plane) and the same barycentre P0, with point M located north of point P0 and side length d as the varying parameter, as shown in Figure 1. In this contribution, we focus on the results obtained by all the tested applications for side lengths varying from 30 m up to 10 km.

Figure 2 shows the estimated variance factors obtained by each adjustment application against the side length d of each network. Small deviations of the estimated variance of unit weight is usually tolerable and does not significantly affect the estimates. For all network side lengths, the estimated variance factors are coherent among all the tested applications. Small discrepancies, about 10 percent difference, are just

[1] http://comet.esgt.cnam.fr/index.php?page=0801

noticeable for the Move3 and Geolab applications, for the network with side length 30 m. All applications are in good agreement whatever the network side length. It is evident that for the Trinet+ application, for network side lengths above 2 km, the estimated variance factor tends to decrease with the network side length (*cf.* Durand *et al.*, 2020). This is coherent with the fact that the Trinet+ application was developed initially for application in an industrial context and not for large networks.

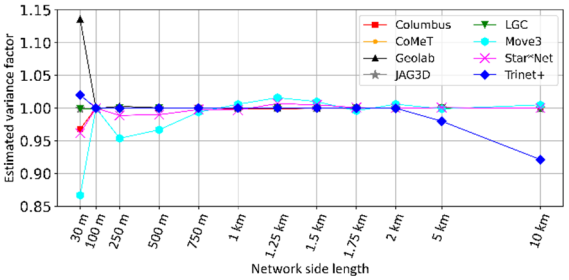

Figure 2. Estimated variance factor against network side length.

Figure 3 shows the differences, in the vertical component, between the adjusted and theoretical coordinates of point M, against network side length. In this study, a convergence criterion of 0.1 mm is used, indicated by the black dotted line, and absolute values of discrepancies less than this criterion could not be considered as significant. As shown in Figure 3, the discrepancies among the applications are negligible for network side lengths under 10 km. The only exception is the LGC application where non-negligible discrepancies exist: 2.3 mm at 5 km and 10 mm at 10 km. These discrepancies are coherent with the use of the spherical datum, the IUGG Sphere, and not the IAG GRS80 reference ellipsoid.

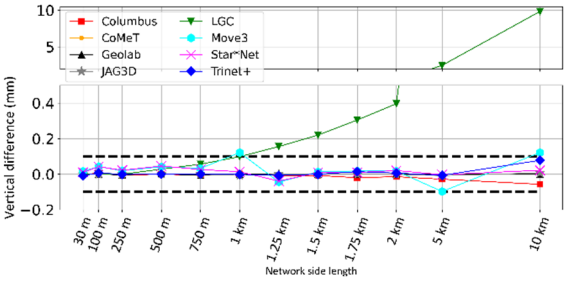

Figure 3. Differences, in millimetres, in the vertical component, between estimated and theoretical coordinates of point M, against network side length.

Figure 4 shows the horizontal distances between the adjusted and theoretical coordinates of point M against network side length. The black dotted line indicates the convergence criterion value for 2D coordinates. For the Columbus, CoMeT, JAG3D, LGC and Trinet+ applications, no discrepancies exist, showing that their three-dimensional mathematical functional models are consistent. For Geolab, Move3 and Star*Net, discrepancies above the convergence criterion exist, but remain very small. We observe mean discrepancies values of 0.15 mm and 0.12 mm for the Move3 and Star*Net applications respectively. For Geolab the maximum discrepancy value is 0.19 mm for the network with side length 250 m.

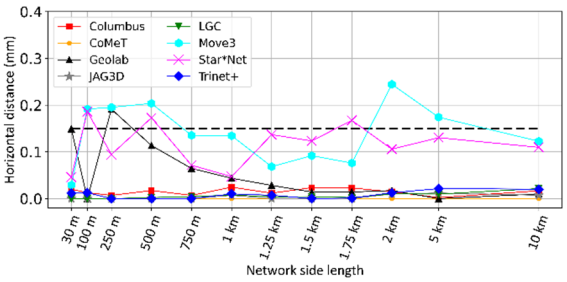

Figure 4. Horizontal distance, in millimetres, between estimated and theoretical coordinates of point M, against network side length.

C. *Impact of the ellipsoidal height of point M*

In this test, the main objective is to study the impact of an ellipsoidal height difference between the points on the results obtained by each software. The starting point is the network with a side length of 500 meters, with local vertical coordinates of 0 m, +10 m and -10 m for points M, S1 and S2 respectively, corresponding to ellipsoidal heights of 433.6658 m, 443.6658 m and 423.6658 m respectively with respect to the IAG GRS80 ellipsoid.

Simulated measurements were computed for networks in which the ellipsoidal height of point M is modified by an offset varying from 30 m to 2500 m. This generates a varying ellipsoidal height difference among the points in the networks. The coordinates of the simulated networks, as well as the measurements, are available on a dedicated webpage[2].

Figure 5 shows the estimated variance factors of unit weight, obtained from each adjustment application, against the offset on the ellipsoidal height of point M. Whilst the network length affects the estimated variance of unit weight slightly, a vertical variation of the network biases the variance significantly in some applications. For the Trinet+ application, the estimated variance factor increases with the offset value on the ellipsoidal height of point M, from 1.0 under 250 m, up to 1.12 at 2.5 km. For the Geolab application, we observe that for offset values above 500 m, non-negligible biases may exist, with a maximum value of 1.3 visible for an offset value of 1 km.

Figure 6 depicts the vertical difference between adjusted and theoretical coordinates of point M, against the offset on the ellipsoidal height of point M. We observe a very good agreement among all the applications, with negligible discrepancies. There are only two exceptions. For the Geolab application,

[2] http://comet.esgt.cnam.fr/index.php?page=0802

significant deviations exist, with a maximum value of 0.5 mm for the network with an offset value of 1.5 km, and a standard deviation of 0.32 mm on the discrepancy value. For the Trinet+ application, small discrepancies (less than 0.2 mm) are visible for offset values between 500 m and 1.5 km.

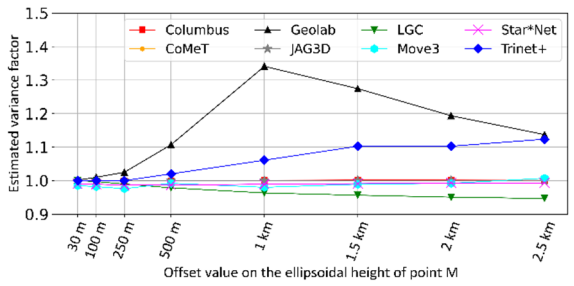

Figure 5. Estimated variance factor of unit weight against offset value on the ellipsoidal height of point M.

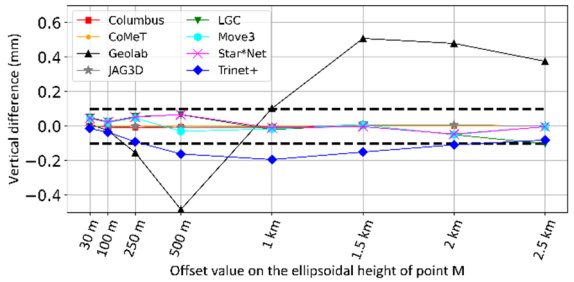

Figure 6. Differences, in millimetres, in the vertical component, between estimated and theoretical coordinates of point M, against offset value on its ellipsoidal height.

Figure 7 shows the horizontal distances, between adjusted and theoretical coordinates of point M, against the offset on the ellipsoidal height of point M. For the Columbus, CoMeT and JAG3D applications, no discrepancies exist. The Move3 and Star*Net applications show very similar results, with small discrepancies: mean value of 0.22 mm and 0.19 mm, respectively. For the LGC and Trinet+ applications, discrepancies increase with the offset value. These go from 0 mm at 100 m to 0.3 mm at 500 m for Trinet+; and for LGC from 0.15 mm at 1.5 km to 0.4 mm at 2.5 km. Geolab is the application most affected by the variation of the offset value, with a mean discrepancy value of 1.7 mm and the maximum value of 3.3 mm for an offset value of 1.5 km.

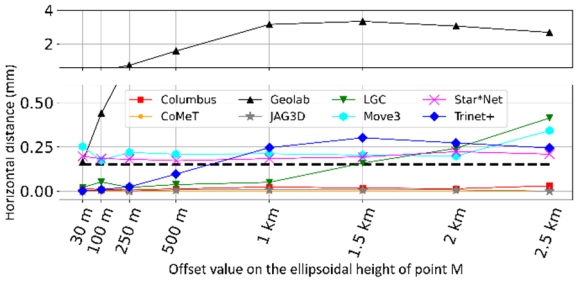

Figure 7. Horizontal distance, in millimetres, between estimated and theoretical coordinates of point M, against offset value on the ellipsoidal height of point M.

V. Conclusion

This paper has investigated the dependency of network results on the adjustment application. Simulated measurements were processed with all tested applications. The main advantage of the test methodology presented is that it allows the selection and control of a single changing parameter in comparable sets of measurements. It also allows random error values, added to the theoretical measurements, to be determined to facilitate the comparison process. Both the estimated variance factors and the differences between adjusted and theoretical coordinates of the unknown point are used as metrics of the agreement between the three-dimensional mathematical models used in each of the tested applications. We apply our test methodology on several network adjustment applications, to study the impact of the network side length and of a difference in the ellipsoidal heights of the points. We show that all applications agree to the first order, but there can be non-negligible discrepancies, that have a real impact on the adjusted coordinates of up to several millimeters.

Almost identical results are obtained from Columbus, CoMet, JAG3D and Trinet+ in the horizontal network. LGC approximates the Earth by a sphere instead of an ellipsoid. For small networks, the resulting vertical deviations are well below 0.1 mm and are negligible. However, the vertical deviation depends on the network extent and, thus, becomes significant in networks larger than 1 km.

In the vertical network, the results are more heterogeneous. Geolab achieved the largest deviations in the results. Moreover, for Move3 and Star*Net, an offset was detected in the horizontal distance, which needs further investigation. Almost identical results yield Columbus, CoMet and JAG3D.

References

Avis, D., and Fukuda, K. (1992). A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra, *Discrete Comput. Geom.*, 8(3), pp. 295–313.

Baarda, W. (1968). *A Testing Procedure for Use in Geodetic Networks*. Netherlands Geodetic Commission, Delft: Publications on Geodesy, 2(5).

Bomford, G. (1971). *Geodesy*. Clarendon Press.

Caspary, W., and Rüeger, J.M. (2000). *Concepts of network and deformation analysis*. 3rd (corrected) impression. Sydney, N.S.W. : School of Geomatic Engineering, University of New South Wales.

Durand, S., Touzé, T., Jones, M., Guillaume, S., Henri Cattin, P., and Morel, L. (2020). Evaluation of Compatibility among Network Adjustment Software: CoMeT, LGC, and Trinet+. *Journal of Surveying Engineering*, 146(2), 04020005.

Herrmann, C., Lösler, M., and Bähr, H. (2015). Comparison of SpatialAnalyzer and Different Adjustment Programs, in Kutterer, H. et al. (eds) *The 1st International Workshop on the Quality of Geodetic Observation and Monitoring Systems (QuGOMS'11)*. Cham: Springer, pp. 79–84.

Lösler, M., and Bähr, H. (2010). Vergleich der Ergebnisse verschiedener Netzausgleichungsprogramme, *Zippelt, K. (ed) Vernetzt und ausgeglichen.*, 3, pp. 205–214.

Pardalos, P.M. (1991). Global optimization algorithms for linearly constrained indefinite quadratic problems, *Comput. Math. Appl.*, 21(6), 87–97.

Pardalos, P.M., and Vavasis, S.A. (1991). Quadratic programming with one negative eigenvalue is NP-hard, *J. Glob. Optim.*, 1(1), pp. 15–22.

Schwieger, V., Foppe, K., and Neuner, H. (2010). Qualitative Aspekte zu Softwarepaketen der Ausgleichungsrechnung, in Kutterer, H. and Neuner, H. (eds) *Qualitätsmanagement geodätischer Mess- und Auswerteverfahren,* Augsburg: Wißner, 61, pp. 129–163.

Shih, P.T.-Y., (2013). On accuracy specifications of electronic distance meter. *Survey Review* 45, pp. 281–284.