A user-friendly software for geodetic network adjustment

Kamal A. Atia, M. Nabil Aly Shoukry and Sherif Ezzat Loutfi
Transportation Engg. Dept., Faculty of Engg., Alexandria University, Alexandria 21544, Egypt

The purpose of this research paper is to develop a software project to perform geodetic network adjustment. The project is presented in a user-friendly, yet professional looking, graphical interface for both input and output data. Three considerations are highly regarded in designing the program: Firstly, ability of the program to perform the adjustment of the angular observations in a general way so that it can handle any number of fixed stations, any number of new stations and any number of angular observations. Computation procedures are based on the "Variation of Coordinates" method. Secondly, it is aimed to achieve the highest possible accuracy in the resulted data. Thirdly, to make a good use of the advances of the visual programming represented in control tools and efficient GUI (Graphical User Interface) to provide an easy and fast way for the user to enter the data and obtain the results. A general description of the program is presented, supported by a numerical example. The results of the numerical example are verified through a comparison between the results given by the program and the results given in scientific literature.

الغرض من هذا البحث هو تقديم مشروع برمجى يقوم بإجراء عملية ضبط الشبكات الجيوديسية ، و قد تم تصميم البرنامج بحيث يقدم واجهة بيانية ذات مظهر احترافى و فى نفس الوقت سهلة الإستخدام ، و روعى فى تصميم المشروع تحقيق ثلاثة اعتبارات : الأول هو قابلية البرنامج للتطبيق بشكل عام على أى شبكة ذات أى عدد من النقط الثابتة و أى عدد من النقط الجديدة و كذلك أى عدد من الأرصاد الزاوية و لذلك تم تصميم البرنامج على أساس إجراء عملية الضبط بطريقة التغير فى الإحداثيات ، و الثانى هو الحصول على أعلى درجة ممكنة من الدقة فى النتائج ، و أما الاعتبار الثالث فقد كان الاستفادة من مزايا البرمجة المرئية المتمثلة فى مجموعة أدوات التحكم و البيئة البرمجية البيانية لإتاحة طريقة سهلة و سريعة للمستخدم فى عملية إدخال البيانات و الحصول على النتائج . و يعرض البحث وصف عام للبرنامج المقدم مدعم بتطبيق على مثال عددى لضبط شبكة مثلثات مع تحقيق للنتائج من خلال مقارنتها بالنتائج الواردة بمراجع علمية.

Keywords: Geodetic networks, Triangulation, Programming, Visual basic, GUI

1. Introduction

The employment of computers in surveying and geodetic applications has obviously increased during last few decades due to the achieved progress in computer technology and developing software and programming facilities. Saving time and effort, evasion of probable human blunders and achieving high accuracy are the main aspects that are realized by using the computers in performing geodetic applications such as the networks adjustment.

Adjustment of a network is the process performed to eliminate the contradiction resulted by the redundancy of observations, that provide more than one possible solution for the model, to find out only one solution which is expected to be the most accurate one. This is done by determining the most probable correction for every observation, which was added to the observations, must remove any contradictions existing between the observations, and the adjusted observations must satisfy all condition equations existing in the model.

The best and most accurate techniques of adjustment are those based on the least squares theorem, which depends on the assumption that the best solution is obtained when the sum of the corrections' squares is minimum. Adjustment by least squares can be done by either of condition equations or variation of coordinates. Since both methods are based on least squares technique, applying either of them on the same problem yield essentially the same unique solution. Variation of coordinates is the method chosen in designing the program presented in this study, due to its capability for programming unlike the method of "Condition Equations", which mostly requires the human judgment and is difficult to support a general program to fit complicated geodetic networks [3].

"Visual Basic" is an event-driven programming language. It is one of the modern computer programming languages, which has come into view in the early nineties of the last century. It was developed by "Microsoft" as a visual version of the programming language "BASIC", which was one of the first computer languages invented. It was developed by John G. Kemeny and Thomas E. Kurtz, two mathematics professors at Dartmouth College in 1964, with the aim of providing their students with an easily learned language that could still handle complicated programming projects [Maureen and Mike]. The codes look a bit like simple English Language. Different software companies produced different versions of "BASIC", such as Microsoft "QBASIC", "QUICKBASIC", "GWBASIC", "IBM BASICA" and so on [6 - 7].

The word "VISUAL" refers to the method used to create the Graphical User Interface (GUI), which distinguishes it from the old "BASIC" programming language. Unlike "BASIC" language where programming is done only in a text-only environment and the program is executed sequentially, in "VISUAL BASIC" programming is done in a graphical environment, that enables the Rapid Application Development (RAD) of Graphical User Interface (GUI), which includes software tools to automatically create the detailed programming required by Windows. These software tools not only create Windows programs, they also take full advantage of the graphical way that Windows works by letting programmers "draw" their systems with a mouse on the computer. The user can simply drag and drop these tools or pre-built objects into place on the screen rather than writing numerous lines of code that describe the appearance and location of interface elements. [6]. This complete set of control tools provided by "VISUAL BASIC" is able to simplify the process of programming for Surveying and Geodesy applications.

2. Mathematical model

"Variation of Coordinates" is one of the accurate methods of adjustment that are based on the least squares theorem. Its idea is to assume approximate coordinates to the network stations that can be obtained for instance from existing maps, approximate calculations or scale drawing, and to form through differentiation a function between the variation of coordinates and the resulted variation of angles values due to these variations of coordinate. This function is called "Observation Equation" and has to be done to every observation. Every Observation Equation includes a constant term which is the difference between the observed value of the angle and the calculated one from the provisional coordinates. From the Observation Equations and according to the least squares theorem, the Normal Equations are formed, which when solved simultaneously, give the values of the corrections needed to be added to the provisional coordinates to present the adjusted coordinates, from which the adjusted values of observations are derived. Since there is no direct function between coordinates and angles, but angles are calculated from lines' bearings, and bearings are calculated from coordinates, an Observation Equation for bearing had to be found out first, and the Observation Equation for angles is developed from the observation equation for bearings. Hereinafter an explanation to the observation equations as introduced in [2 and 3].

2.1. Observation equation for bearings

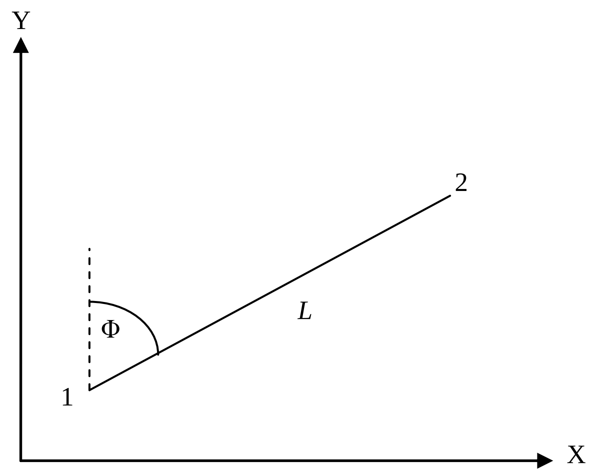


Fig. 1. Observation equation for bearings.

Referring to fig. 1, and regarding the line 1– 2:

$$L = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}\ , \qquad (1)$$

$$\tan\ \Phi = \frac{X_2 - X_1}{Y_2 - Y_1}\ , \qquad (2)$$

by differentiating:

$(Y_2 - Y_1)\ \sec^2 \Phi\ \partial\Phi + \tan \Phi\ (\partial Y_2 - \partial Y_1) = \partial X_2 - \partial X_1$

since

$$(Y_2 - Y_1)\ \sec^2 \Phi\ \ \partial\Phi + \tan\ \Phi\ (\partial Y_2 - \partial Y_1)\ =\ \partial X_2 - \partial X_1$$

$$\text{since}\ \ L = \frac{Y_2 - Y_1}{\cos \Phi}\ . \qquad (3)$$

$$\begin{aligned}\partial\Phi &= -\frac{\sin \Phi}{L}(\partial Y_2 - \partial Y_1) + \frac{\cos \Phi}{L}(\partial X_2 - \partial X_1) \\ &= \frac{\sin \Phi}{L}(\partial Y_1 - \partial Y_2) - \frac{\cos \Phi}{L}(\partial X_1 - \partial X_2)\ , \qquad (4)\end{aligned}$$

where $\partial\Phi$ is the difference in the bearing of the line 1-2 due to the variation of coordinates of the points 1 , 2 as (∂x_1, ∂y_1) and (∂x_2, ∂y_2) in radial scale, while the value of $\partial\Phi$ in seconds can by calculated as:

$$\partial\Phi'' = \frac{\sin \phi}{L}\frac{1}{\sin 1''}(\partial Y_1 - \partial Y_2) - \frac{\cos \Phi}{L}\ \frac{1}{\sin 1''}(\partial X_1 - \partial X_2)\ , \qquad (5)$$

but $\partial\Phi''\ = \Phi_m - \Phi_c$,

where:
Φ_m is the measured bearing, and
Φ_c is the calculated bearing from the provisional coordinates.

So, the general form of observation equation for bearings can be rewritten as:

$$\Phi_m - \Phi_c = \frac{\sin \Phi}{L}\frac{1}{\sin 1''}(\partial Y_1 - \partial Y_2) - \frac{\cos \Phi}{L}\ \frac{1}{\sin 1''}(\partial X_1 - \partial X_2)\ . \qquad (6)$$

2.2. Observation equation for angles

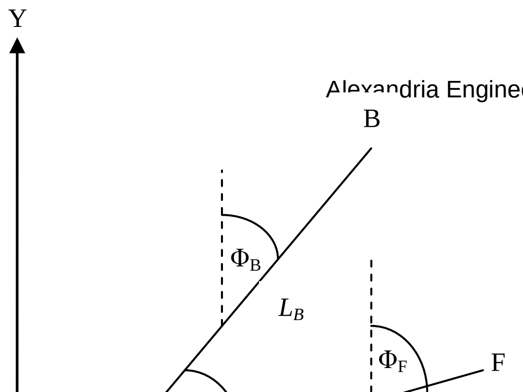


Fig. 2. Observation equation for angles.

Referring to fig. 2, and regarding the angle Θ where:
O is the Observation station,
B is the Back station, and
F is the Fore station.

$$\Theta = \Phi_F - \Phi_B, \qquad (7)$$

and the observation equation for the angle Θ can be obtained by subtracting the observation equation of Φ_B from the observation equation of Φ_F. if Θ_m is the measured angle, and Θ_c is the calculated angle from the provisional coordinates, then:

$$\theta_m - \theta_c = \left[\frac{\sin \Phi_F}{L_F \sin 1''}(\partial y_o - \partial y_F) - \frac{\sin \Phi_F}{L_F \sin 1''}(\partial x_o - \partial x_F)\right] -$$

$$\left[\frac{\sin \Phi_B}{L_B \sin 1''}(\partial y_o - \partial y_B) - \frac{\sin \Phi_B}{L_B \sin 1''}(\partial x_o - \partial x_B)\right]. \qquad (8)$$

This equation can be rewritten as:

$$\begin{aligned}\partial\theta &= (B_B - B_F)\ \partial x_O - (B_B)\ \partial x_B + (B_F)\ \partial x_F \\ &+ (A_F - A_B)\ \partial y_O + (A_B)\ \partial y_B - (A_F)\ \partial y_F + (\theta_c - \theta_m), \qquad (9)\end{aligned}$$

Where,

$\partial\theta$ is the variation of the angle value due to the variation in the coordinates of the three stations.

$$A = \frac{\sin\Phi}{L\sin 1''}, \tag{10}$$

$$B = \frac{\cos\Phi}{L\sin 1''}. \tag{11}$$

Forcing $\partial\theta$ to zero due to least squares theorem, the value (θ_c – θ_m) is the constant term of the equation

$$(B_B - B_F)\,\partial x_O - (B_B)\,\partial x_B + (B_F)\,\partial x_F + (A_F - A_B)\,\partial y_O + (A_B)\,\partial y_B - (A_F)\,\partial y_F + (\theta_c - \theta_m) = 0. \tag{12}$$

3. Description of the program

3.1. Overview

The program was designed to perform the adjustment of angular observation for any network up to 99 stations and contains up to 99 observations. It is considered in designing the program to make optimal use of the outstanding facilities of "VISUAL BASIC" represented in its complete set of visual control tools and Graphical User Interface (GUI) to provide an easy way for the user to enter the input data and to read the results. Hence, the program is constructed in multi forms of various layouts to fit every kind of data given and obtained as well as every step of working with program by the user. In addition to displaying the results in an easy to read form on the graphical interfaces, a printable form of the results in a text file is optionally created and presented by the program. One more text file containing the observation equations and normal equations that are formed by the program within the computation procedures, can be optionally created and presented.

3.2. Project configuration

The program project consists of total 32 files, distributed as 26 forms, 4 binary files and 1 module file in addition to the project file. The total size of the project under Visual Basic environment is 2.05 MB, while the executable file (The genuine program) is compressed into 884 KB. The program contains a total of 38 independent variables, 35 arrays of different sizes and 2 constants [1].

The 26 forms of the program are distributed as the title form, the program info (About) form, the Network configuration data input form, 3 forms for stations data input, 5 forms for observations data input, the computations and output data types form, 2 forms for coordinates corrections output, 2 forms for adjusted coordinates output, 5 forms for the adjusted observations output and finally 5 forms for the observations residuals output [1].

Variables and arrays, whose values represent mathematical items such as equations coefficients and results, are declared as "Double" to get highest possible accuracy, while variables that don't require accuracy, such as the number of network stations and number of observations and those who work as counters for the loops in the program procedures, and arrays representing stations names, are declared as "Byte". The two mathematical constants declared in the grogram are the circular constant "Π", its value is declared as "Pi = 3.14159265358979", and "S1" defining the value of Sine ($1''$) declared as "S1= 4.84813681107636E-06" [1].

3.3. Data input

The data required to run the program are the number of stations in the network, number of angular observations, provisional coordinates of the stations in meters, the observations in degrees, minutes and seconds, and the weights of observations.

Base lines stations or fixed points have to be marked for the program as "Fixed", so it is not subject to corrections. The observations are named by the numbers of "Back stations", "Observation Station" and "Fore Station". Weights can be given in either the form of standard deviation (σ) or direct weight form (W).

The "Network Configuration" Interface, fig. 3 is dedicated to enter the total number of stations, the number of observations and to select whether the observations are of equal or unequal weights and the form of available weight data in case of unequal weights.

Next to the "Network Configuration" Interface, the "Stations Data" interface, fig. 4 is dedicated to enter the Provisional Coordinates of the stations and to mark the Base Lines stations as "Fixed". Since the "Stations Data" interface absorbs data for up to 45 stations, it may be repeated one or more times if the given number of stations exceeds 45. The program stores the Provisional coordinates in two arrays PX(1 to NS), PY(1 to NS) where NS is the total number of stations.

Next to "Stations Data" interface comes the "Observations Data" interface, fig. 5. Every observation is given by 7 values. the first 3 values are the names (numbers) of the Back station (*B*), Observation station (*O*) and the Fore station (*F*), which specify the angle and are stored by the program in 3 arrays BS(1 to NO), OS (1 to NO), FS(1 to NO) where NO is the total number of observations. The next 3 values are the value of observation in degrees, minutes and seconds. The program stores the angles values after transforming them to decimal form in array OA(1 to NO). The last required value for every observation is the weight only if the option "Unequal weights" in the "Network Configuration" Interface was selected. Since the "Stations Data" interface takes up to 20 observations only, it may be repeated one or more times according to the given number of observations.

The weights array W(1 to NO) is created anyway. If the option "Equal weights" is selected, all values in this array are set to one. If the options "Unequal Weights" with data form "Deviation (σ)" are selected, every given value will be inversed and squared before being stored in the array, but if the data form "Weight (W)", is selected, the given values are handled without change [1].

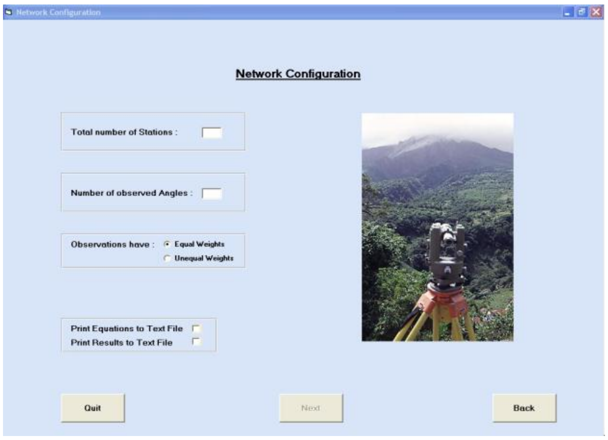


Fig. 3. Network configuration interface.

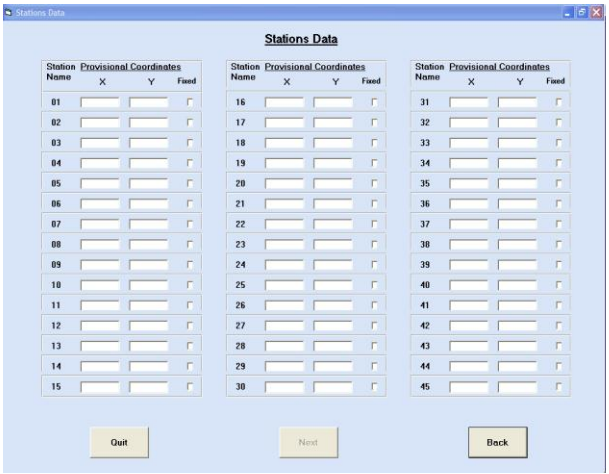


Fig. 4. Stations data interface.

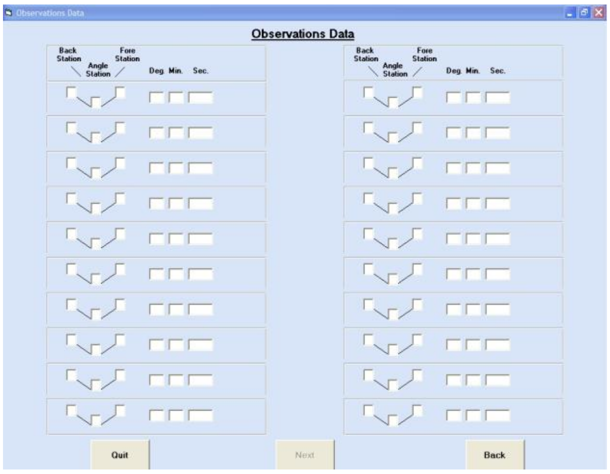


Fig. 5. Observations input interface - equal weights.

3.4. Computation algorithm

3.4.1. Forming observation equations

For every observation the program calculates the bearings (Φ_F), (Φ_B) from the Provisional Coordinates of the three stations: (B), (O) and (F) using eq. (2). The calculated angle (Θ_c) is then calculated using eq. (7). The lengths of lines (B – O) and (O – F) are calculated using eq. (1).

The factors A_B, A_F, B_B, B_F are calculated using the eqs. (10 and 11) one time for station (B) and another time for station (F). Thus, all coefficients of the Observation Equation as shown in eq. (12) are determined. A loop in the program lets it repeat this process for every observation. The coefficients of the observation equations are ordered into a two-dimensional matrix $M_{(m+n+1)}$,

Where,

n is the number of observations, and

m is the total number of stations × 2.

Every pair of columns in this matrix refers to the coefficients of ∂x, ∂y of one station and every row expresses one observation equation, table 1. Thus, every row of the array must contain 6 values related to ∂x, ∂y for each of the 3 stations B, O, F in addition to the constant term in the last column, while the rest items in the row are zero. Stations marked as "Fixed", their ∂x, ∂y coefficients are considered zero, their columns are eliminated from matrix, and the matrix gets resized by reducing its columns by the number of eliminated columns.

3.4.2. Forming normal equations

If "A" represents the matrix of coefficients given in table 1, then the observation equations can be given in matrix form as:

$$A\,X + L = V, \tag{13}$$

Where,

L is the vector of absolute terms (length: m),

V is the vector of residuals of observations (length: m),

n is the number of observations, and

m is the number of unknown parameters.

The normal equations matrix can be given by [3]:

$$(A^T P\, A)\, X + A^T P\, L = 0, \tag{14}$$

where P is the diagonal matrix of weights. Eq. (14) is based on the theorem of least squares as expressed by:

$V^T P\, V$ = minimum.

3.4.3. Solving the normal equations

A sub-routine code based on "Gauss-Jordan" elimination algorithm for solving the normal equations was quoted from ref. [4]. The original code, which was written in "Fortran" programming language, was subjected to some alteration by the researcher to match "Visual Basic" before being merged into this program. "Gauss-Jordan" elimination is based on two essential rules concerning a matrix expressing a set of linear eq. [5]:

a. All elements in any row can be multiplied or divided by any non-zero value without changing the results.

b. Any row can be added to or subtracted from any other row without changing the result.

These two rules can get repeatedly applied as needed until the values of the unknowns are achieved and presented as the values of the last column in the matrix.

Table 1
Observation equations matrix

∂x_1	∂y_1	∂x_2	∂y_2	∂x_3	∂y_3						constant terms
$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$	$a_{1,5}$	$a_{1,6}$	.	.	.	.	.	$a_{1,m+1}$
$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$a_{2,4}$	$a_{2,5}$	$a_{2,6}$	.	.	.	.	.	$a_{2,m+1}$
$a_{3,1}$	$a_{3,2}$	$a_{3,3}$	$a_{3,4}$	$a_{3,5}$	$a_{3,6}$	.	.	.	.	.	$a_{3,m+1}$
.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.
$a_{n,1}$	$a_{n,2}$	$a_{n,3}$	$a_{n,4}$	$a_{n,5}$	$a_{n,6}$	.	.	.	.	.	$a_{n,m+1}$

3.5. Results arrangement

Values of unknowns resulted from solving the normal equations are ordered in a separate one-dimension array, as every following pair of its elements represents the corrections ∂x and ∂y needed to be added to the provisional coordinates. This array is then resized to (m) elements (m = Total number of stations × 2) to incorporate ∂x and ∂y as zeros for the stations marked as "Fixed" in their original order. Afterwards, this array is divided into 2 arrays DX(1 to NS) and DY(1 to NS), so the array DX()contains the ∂x values and the array DY() contains the ∂y values.

A new pair of arrays is launched, AX (1 to NS) and AY (1 to NS) to contain and represent the required adjusted coordinates as the sum of provisional coordinates plus corrections

$$AX(i) = PX(i) + DX(i)$$
$$AY(i) = PY(i) + DY(i)$$

3.6. Solution Iteration

If the Provisional Coordinates are relatively far from the true coordinates, the adjusted coordinates resulted from first round of solution might not be accurate enough. Therefore, the program is designed to test the results and iterate the solution as many times as needed to achieve the maximum level of accuracy [1]. The iteration is performed by replacing the Provisional Coordinates by the newly resulted adjusted coordinates and repeating the steps of adjustment. When the values of ∂x and ∂y for all stations reach less than 0.0001, the program stops the iteration, recalls the last resulted coordinates and considers it the final adjusted coordinated. The final values of corrections are calculated by subtracting the original provisional coordinates from the final adjusted coordinates.

$$DX(i) = AX(i) - PX(i)$$
$$DY(i) = AY(i) - PY(i)$$

From the adjusted coordinates, the adjusted bearings of lines are calculated, from which the adjusted angles are derived and stored into array *AA*(1 to NO) where "NO" is the number of observations. Finally one more array is created *DA*(1 to NO) containing the observations residuals.

$$DA(i) = AA(i) - OA(i),$$

where,
$AA(i)$ is the adjusted angle, and
$OA(i)$ is the observed angle.

Both of adjusted angles and residuals values get transformed from decimal form to be displayed into the form of degrees, minutes and seconds. It's essential that the resulted corrected observations must satisfy all the angular conditions in the network.

3.7. Results display

Like the data input interfaces, other four models of interfaces were designed to display the four types of results:

- Coordinates corrections data output interface fig. 9
- Adjusted coordinates data output interface fig. 10
- Adjusted observations data output interface fig. 11
- Observations residuals data output interface fig. 12

The output data type selection interface, fig. 8, enables the user selecting the type of results to be displayed. In addition to the interfaces, the program can optionally save the results into a printable text file.

Fig. 6. Observations input Interface - weights (*W*).

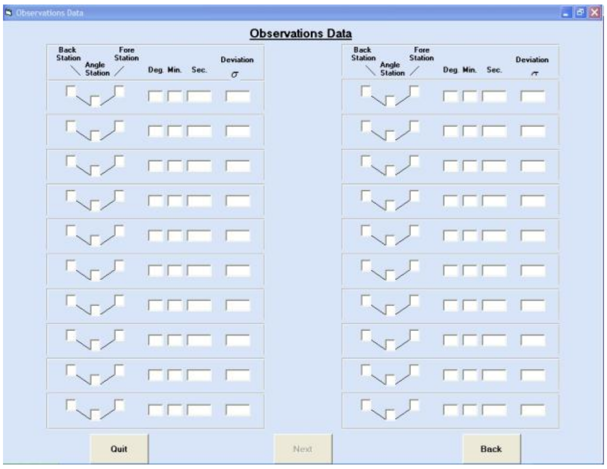


Fig. 7. Observations input Interface - deviations (σ).

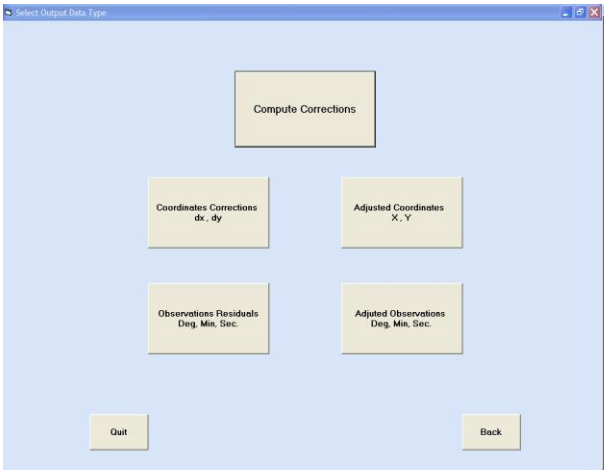


Fig. 8. Output data type selection interface.

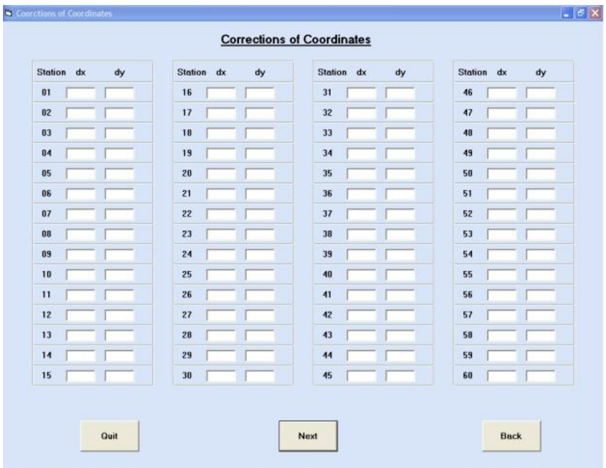


Fig. 9. Coordinates corrections interface.

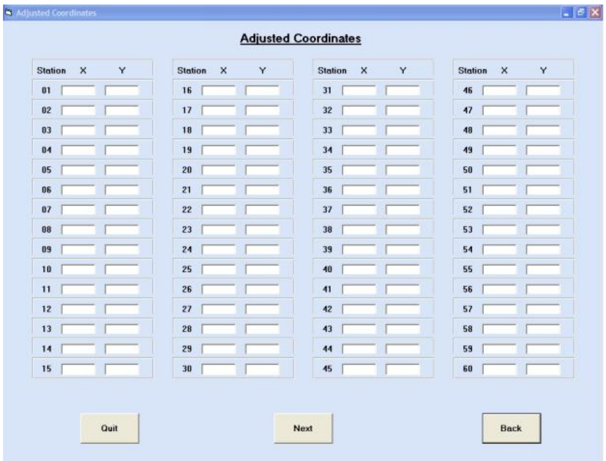


Fig. 10. Adjusted coordinates interface.

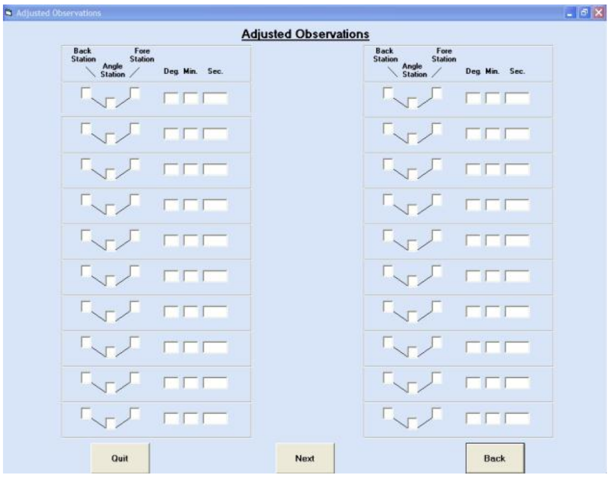


Fig. 11. Adjusted observations interface.

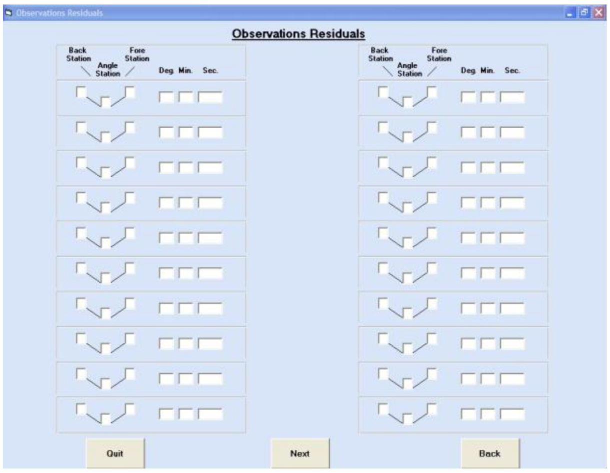
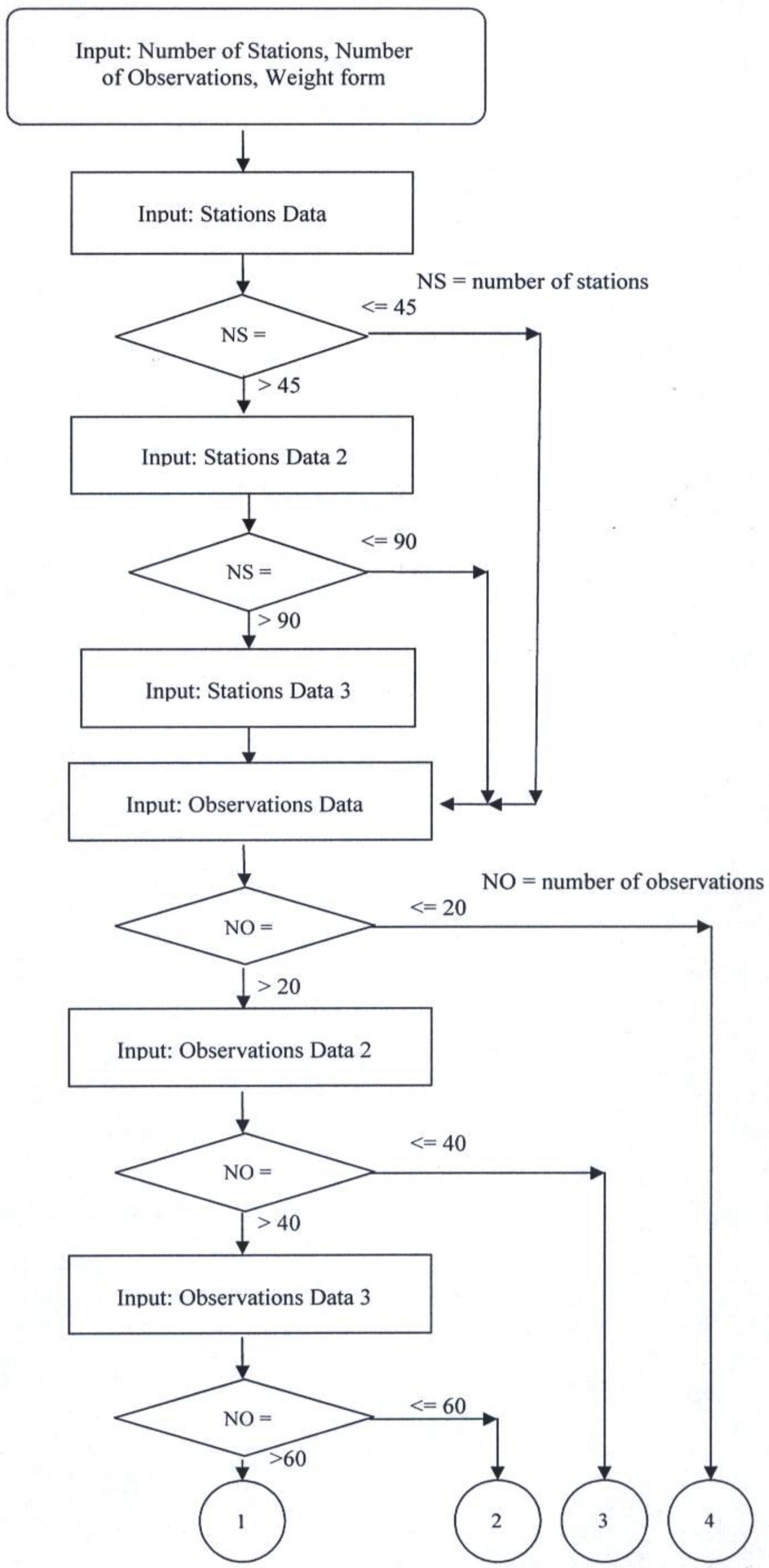
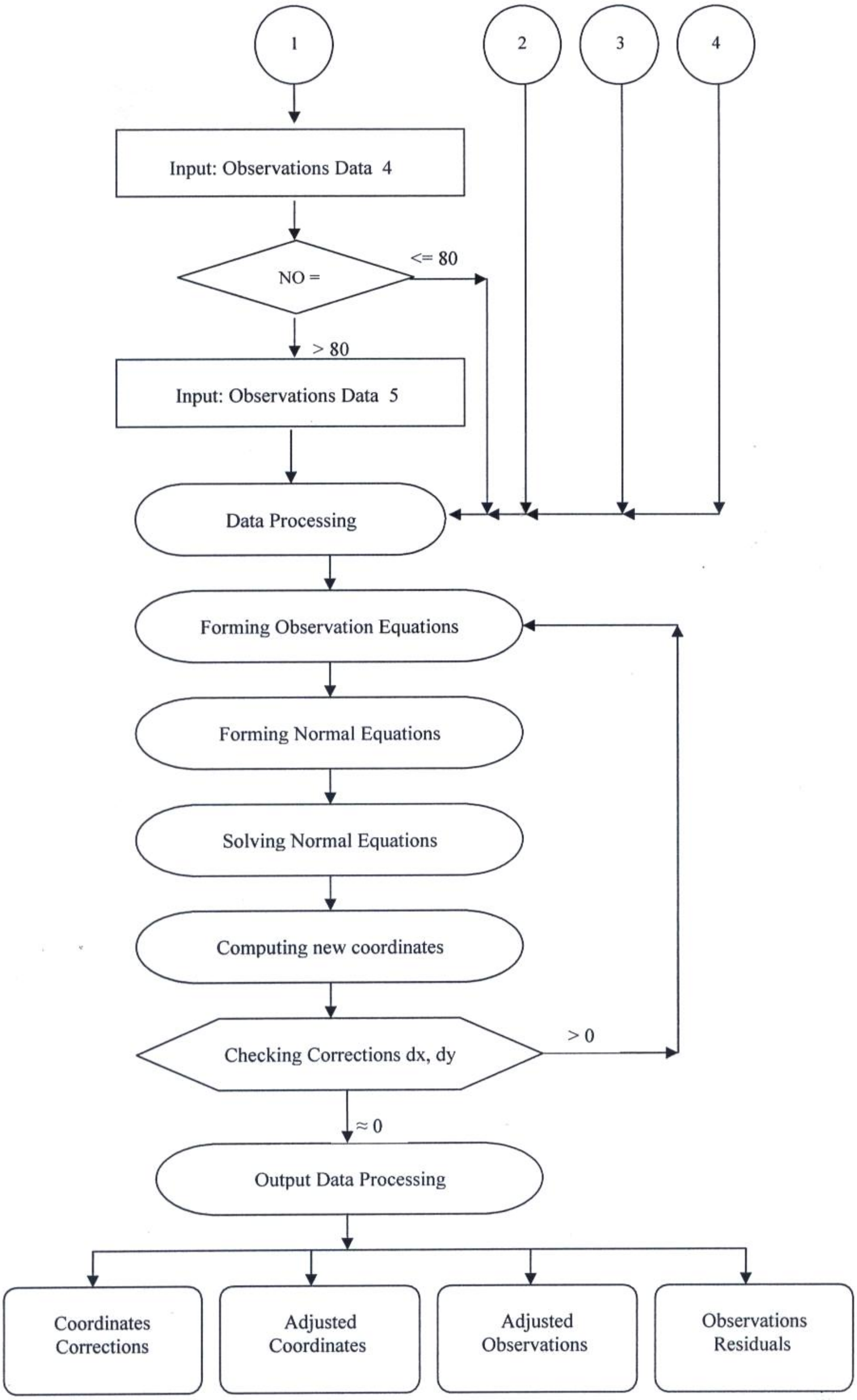


Fig. 12. Observations residuals interface.

4. Program flow chart




1
2
3
4
Input: Observations Data 4
NO =
<= 80
> 80
Input: Observations Data 5
Data Processing
Forming Observation Equations
Forming Normal Equations
Solving Normal Equations
Computing new coordinates
Checking Corrections dx, dy
> 0
≈ 0
Output Data Processing
Coordinates Corrections
Adjusted Coordinates
Adjusted Observations
Observations Residuals

5. Conclusions

The software introduced in this paper can be summarized into the following characteristics and remarks:

1. A software sample was designed to perform the Adjustment of Triangular observation for Geodetic Networks using the "Variation of Coordinates" method, which is one of the least squares techniques. The programming language used is "MICROSOFT VISUAL BASIC 6.0" for 32-bit Windows Development, Professional Edition.
2. The introduced program is able to adjust geodetic networks of any figure with any number of stations and any number of observations up to 99 for each.
3. The program was based on the "Variation of Coordinates" method because it is more suitable for programming as compared to the method of condition equations. Actually, the former can fit any geodetic network unlike the latter method, which would require entering the condition equations of network as a part of the input data, which is a complicated and much time-consuming process.
4. To achieve highest possible accuracy, it was considered to establish the computation procedures on the most accurate and convenient mathematical algorithm and equations referring to the "Variation of Coordinates" method. A brief theoretical derivation of the used formula has been introduced. All variables and matrices representing, in a direct or indirect way, equation coefficients were declared as "Double", which supports values with significant figures of 14 digits.
5. Taking advantage of the outstanding facilities of "VISUAL BASIC" represented in its complete set of visual control tools and Graphical User Interface (GUI) was highly regarded in constructing the program to provide a simple and time-saving way for the user to enter the input data and read the results. Moreover, a printable text file containing the results is optionally created and presented by the program.
6. The results produced and presented by the program are classified into the following four types of output data:

- Coordinates corrections (*dx*, *dy*) for each station.
- Adjusted coordinates (*X*, *Y*) of each station.
- Adjusted angular value (Deg, Min, Sec) of each observation.
- Residual value (Deg, Min, Sec) of each observation

6. Recommendations for future works

1. Since this program was designed to adjust Triangulation Networks, it can be developed to perform also the adjustment of Trilateration and Hybrid Geodetic Networks.
2. A similar program can be developed to perform the adjustment of elevations for level nets based on least squares technique.
3. Developing a software for the adjustment of 3-D Networks and combining it with differential Global Positioning System (GPS) software to adjust the networks of vectors between stations occupied by GPS receivers.

References

[1] S. Loutfi, "The Use of Visual Basic in Geodetic Network Adjustment", Dissertation, M.Eng., Faculty of Engineering; Alexandria University, p. 100 (2007).
[2] A. Shoukry, M. Hosny and M. Rashad, "Geodetic Surveying", Monshaat El Maaref, Alexandria, Faculty of Engg., Alexandria University ISBN 977-103-485-5 (1993).
[3] K.A. Atia and M. Farid "Modern Geodetic Survey", Lecture Notes, Faculty of Engg., Alexandria University, p. 365 (1998).
[4] K.A. Atia "Comparison Between Different Methods Used in Fixing Geodetic Points and Programming", M.sc. Thesis; Faculty of Engg.; Alexandria University, (1970).
[5] Don Lancaster and Synergetics, "Tech Musings", http://www.tinaja.com/glib/muse142.pdf (1999).
[6] Liew Voon Kiong, "Visual Basic Tutorial"; http://www.vbtutor.net/vbtutor.html (2007).
[7] Wikipedia, "Visual Basic", Wikimedia Foundation, Inc http://en.wikipedia.org/wiki/Visual_Basic (2007).

Received May 4, 2008
Accepted May 17, 2008