A FORTRAN BLOCK ADJUSTMENT PROGRAM BY INDEPENDENT MODELS AND LEAST SQUARES SUITABLE FOR ARBITRARY BLOCKS AND SMALL COMPUTERS.-

JOSE EDUARDO JULIA

The treatment of this program is that of the Anblock with iterations between planimetry and altimetry and with direct solution of the reduced normal equations, which are solved by the Gauss elimination technique with an optimisation consisting not in minimizing the band width but in the skipping of all zeros.

The program can adjust any type of block regardless of shape, overlap and points distribution. It has only 900 Fortran instructions which require almost 30 K bytes in an IBM 370 computer. The working area requirements depend on the block size (50 models need 10 K bytes and 200 models require 40 K bytes). Thus for relatively large blocks and small computers, there is not even need of program segmentation or special techniques of dynamic storage for the working area. This aids however will be necessary either with very large blocks or very small computers. The data handling is extremely easy; the output of the plotting instrument can be used directly as input of the program. Some adjusted blocks of real work are included.-

JULIA JOSE EDUARDO

Avenida INDEPENDENCIA 1800 4.000 S.M. de TUCUMAN

ARGENTINA

Instituto de GEODESIA- Facultad de CIENCIAS EXACTAS-Universidad Nacional de TUCUMAN

December 29th, 1979

1.Introduction

Our Institute could develop in 1977 a block adjustment program by least squares and independent models(3),which was able to handle relatively irregular blocks .

Later,in 1978,a second program(4)based on Anblock formulation and Gauss Seidel procedure,suitable for blocks of arbitrary shape and overlap was completed. The main feature of this program was that the solved system was that of the Reduced Normal Equations and not that of the Normal Equations as it is the case in the solutions given by Créhange (Anblock) and Masson d'Autum (Bundles) described in (1) and (6). The convergence was good:Starting with good approximate values (previous strip formation),less than 5 iterations were enough;starting with approximate values sensibly worse than those obtained with strip formation,10 iterations sufficed;starting with arbitrary approximate values (all of them equal to zero,for instance),40 iterations were needed. With too scarce ground control the number of iterations was somewhat greater .-

Finally in july 1979 a program suitable for large arbitrary blocks and small computers based on a direct solution of the reduced normal equations could be completed. This program,whose name is COBLO (for Compensación en Bloque),is based on the Anblock solution with iterations between planimetry and altimetry and it will be described in this paper.

2.Employed Algorithm

The mathematical treatment for the planimetry is that of Anblock as presented in (1). For the altimetry similar formulas were developed by the author.

2.1. Formation of the Reduced Normal Equations

The block of Fig.1 will give rise to a normal equations system(Fig. 2). Starting from this system one can obtain the so called "reduced normal equations system "; Fig.3 shows this system for the case in which the vector X_R containing the unknown point coordinates has been eliminated and the models have been numbered according to flight direction(the vector X_t in normal equations will be called simply X in the reduced normal equations system).

In Fig.3,the submatrixes N will be of 4 by 4 in planimetry and of 3 by 3 in altimetry,and the subvectors X_1, X_2, etc., which contain the orientatation parameters of models,will be of 4 by 1 in planimetry and 3 by 1 in altimetry . The independent terms vector R is also subdivided into subvectors R1,.....,R5, in the same way.

The formation of the reduced normal equations is done in the following way :

The program takes a point,establishes the models to which it belongs and computes the contributions to the corresponding submatrixes.

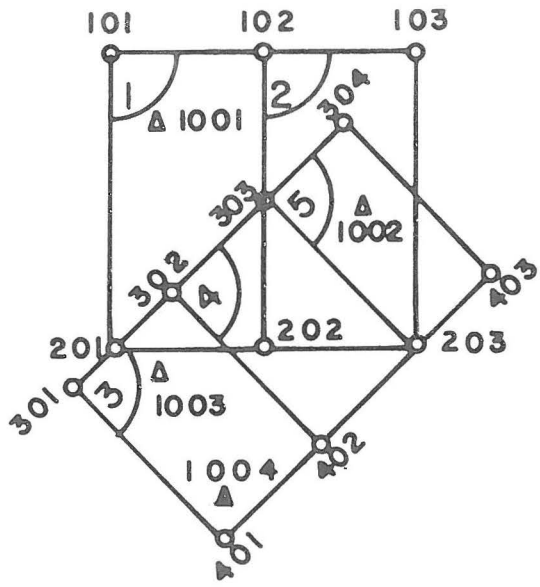
101
102
103
1
2
Δ 1001
304
303
5
Δ 1002
302
4
403
201
202
203
301
3
Δ 1003
1004 Δ
402
401

Fig.1

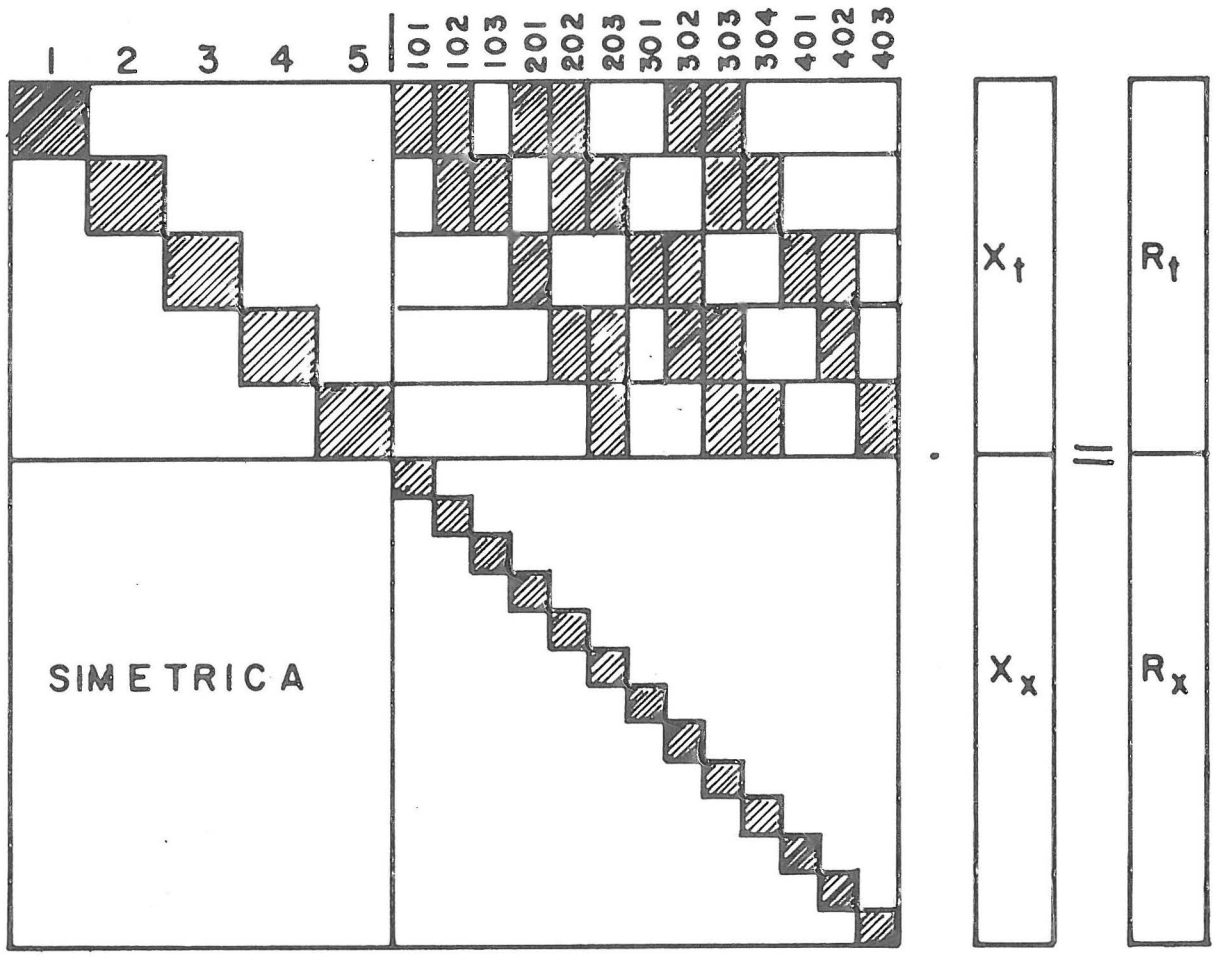
1 2 3 4 5
101 102 103 201 202 203 301 302 303 304 401 402 403
SIMETRICA
X1
XX
R1
RX

Fig.2

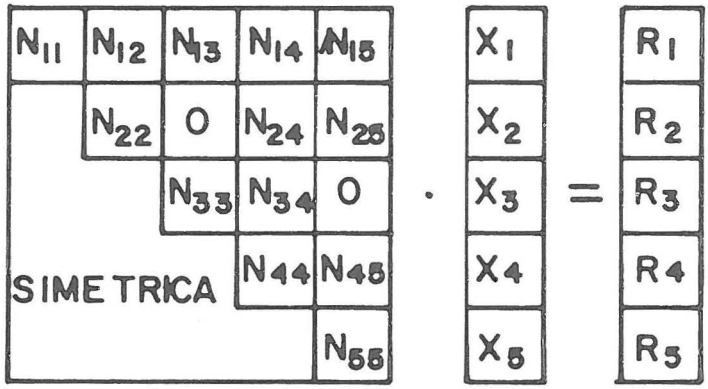

Fig.3

The point 303, for instance, contributes to form submatrixes N11, N12, N14, N15, N22, N24, N25, N44, N45, N55. Only one of these submatrixes is in core memory. Once the contribution is computed, it is transfered to external storage. The same process is repeated for all the unknown points participating in the adjustment and their contributions are added to the already stored ones.

2.2. Solution of the Reduced Normal Equations

We shall consider the block of Fig.4, which gives rise to the reduced normal equations system represented in Fig.5 .

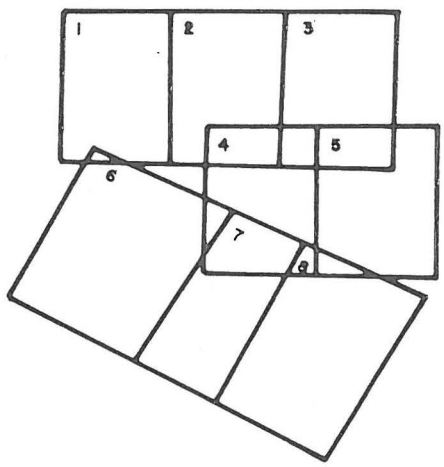

Fig.4

N_{11}	N_{12}				N_{16}		
N_{12}^T	N_{22}	N_{23}	N_{24}				
	N_{23}^T	N_{33}	N_{34}	N_{35}			
	N_{24}^T	N_{34}^T	N_{44}	N_{45}	N_{46}	N_{47}	N_{48}
		N_{35}^T	N_{45}^T	N_{55}			N_{58}
N_{16}^T			N_{46}^T		N_{66}	N_{67}	
			N_{47}^T		N_{67}^T	N_{77}	N_{78}
			N_{48}^T	N_{58}^T		N_{78}^T	N_{88}

Fig.5

Unlike what it is done in other methods (2), no attempt is made here to minimize the band width, and the solution is basically the Gauss elimination and back sustitution method, employing submatrixes as computation units.

In spite of the large bandwidths,which could be present, the procedure is efficient because:

1) One works only with half part of the matrix because of its symmetry.
2) In computing the succesive reductions,all matrix products,having a zero matrix as one of its factors,are omitted. Thus, in the first reduction of the system of Fig.5,in substracting from the second "row"the first "row" premultiplied by $N_{12}^T N_{11}^{-1}$, only the following computations will be performed:

$$N_{22} - N_{12}^T N_{11}^{-1} N_{12} \; ; \; 0 - N_{12}^T N_{11}^{-1} N_{16}$$

The submatrixes N_{23} and N_{24} will remain unchanged,and the computations:

$$N_{23} - N_{12}^T N_{11}^{-1} \cdot 0 \; ; \; N_{24} - N_{12}^T N_{11}^{-1} \cdot 0$$

are skipped.
In the same way the substraction,third "row"minus first "row" premultiplied by $N_{13}^T N_{11}^{-1}$ is also skipped because $N_{13}^T = 0$, and the third "row" remains thus unchanged.
In this way much computer time is saved and a good efficiency is attained.

3. Type of blocks to be admitted by COBLO

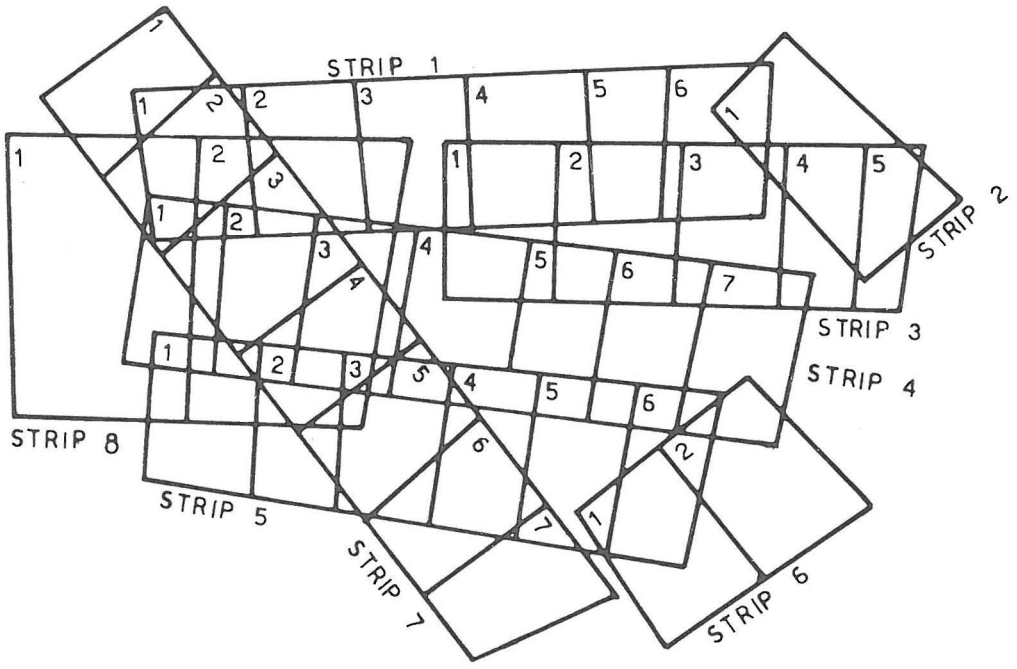

Fig.6

There are no restrictions regarding the shape and overlap of the block . The strips can have completely different scales and positions . Thus,it is possible to adjust simultaneously blocks formed by two or more independent flights ,(Fig.6). The minimun block size is one model.

The number of unknown points per model participating in the adjustment is arbitrary and can be rather large because they are stored externally. Each point has an identification number, by means of which the program establishes the ties. The numbering of tie and terrain points is arbitrary. The perspective centres act as additional tie points in altimetry.

COBLO admits ground control points in arbitrary positions and their number per model can be also large.

The ground control points can be planialtimetric,altimetric or planimetric.

No initial approximate values for the unknowns are required.

Some minor changes will be made in the future to introduce the use of points along shorelines .

4. Data handling

The data handling is very easy because the plotting instrument output can be used directly as input of COBLO. Each model card deck is formed by the cards of all model points (one card per point) grouped arbitrarily. Preceding each model deck there is one card containing the model coordinates of the two perspective centres and information about the quantity of tie and ground control points.

The model decks thus formed are grouped sequentially (The same order of measurement at the plotting instrument) to form the strip deck.

All strip decks are then grouped in an arbitrary way to form the final data deck. Figure 7 shows one way to form the data deck of block of Fig.6. Before each strip deck a card should be placed indicating how many models the strip has.-

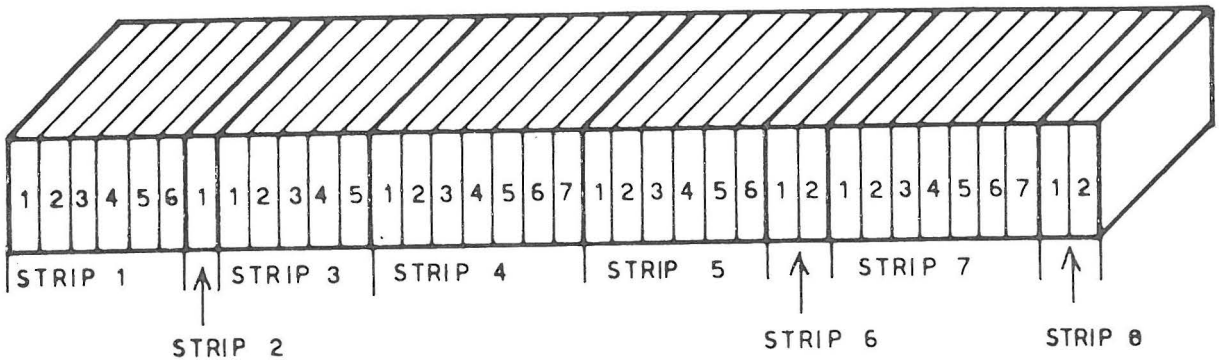

Fig.7

Figure 8 shows a fictitious block of 6 models: 3 strips of 2 models each. Supposing that all models were measured with the same base, the scale of the third strip will differ strongly from the other two (the factor is almost 3).

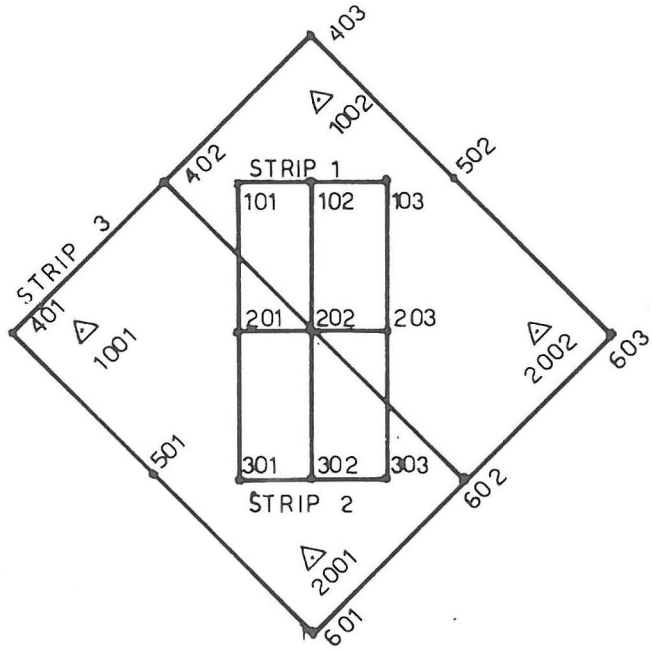

Fig.8

The simulated terrain is somewhat montaneous and all models were taken out from their correct positions to prepare the fictitious input data, by applying 3 translations and 3 rotations (orthogonal transformations). As no simulated errors were introduced, the final results of the adjustment were, naturally, exact.

Figure 9 is the input data of block of Fig.8. The first card contains information about the number of iterations (3 for montaneous terrain; 2 for flat terrain) and the weight to be assigned to ground control. The second card contains the block size (number of models) and the number of strips. After these two specification cards, there follows the above described data deck.

Figure 10 is the output of this example (the results of two iterations are shown). Each line contains (from left to right): the identification number of the point, its three adjusted coordinates and the discrepancies between these last coordinates and the transformed coordinates of the same point in all models in which it appears. If the point happens to be in more than 4 models, a second line will be necessary to print all discrpancies (this is the case with point 202 which appears in 6 models).

The treated example was included to show the data handling, but it also shows how COBLO can process blocks with cross strips.

```
3   1
  6  3
 2
 4  0          205.311   388.674   506.734   405.272   390.510   503.244
       101    158.209   596.482   209.119
       102    364.312   598.110   213.626
       201    201.734   196.592   200.947
       202    401.625   198.527   193.109
 4  0          206.260   403.434   496.655   406.185   398.229   496.413
       102    213.933   598.121   196.234
       103    413.817   593.005   202.078
       202    203.547   198.142   193.217
       203    403.496   192.884   191.963
 2
 4  0          205.928   397.099   499.905   405.892   398.576   496.414
       201    199.260   599.477   200.747
       202    399.155   601.137   193.218
       301    202.245   199.607   208.214
       302    402.242   201.008   207.724
 4  0          184.980   397.035   506.231   389.935   401.214   506.929
       202    186.747   601.252   204.055
       203    386.732   605.473   201.753
       302    195.085   201.126   216.478
       303    395.034   205.277   219.166
 2
10  2          183.066   415.368   493.918   382.981   420.654   495.361
       401    183.247   606.443    62.334
       402    382.189   611.751    65.837
       501    188.476   406.628    70.057
       202    388.482   411.768    65.431
       201    337.176   460.448    65.187
       301    239.786   357.941    69.948
       302    291.060   309.313    72.646
       303    342.338   260.677    75.031
       601    193.756   206.667    70.711
       602    393.662   211.968    73.861
      1001    234.597   557.756    62.225  1200.000  1800.000  1000.000
      2001    242.448   257.924    65.507  1800.000  1200.000  1000.000
10  2          201.835   394.857   503.338   401.773   398.408   499.848
       402    190.773   602.107    80.492
       403    390.738   605.634    74.422
       101    241.561   553.077    74.845
       102    292.468   503.931    75.947
       103    343.357   454.804    75.969
       202    194.182   402.247    72.412
       502    394.258   405.664    76.698
       203    245.020   353.168    69.607
       602    197.747   202.235    73.143
       603    397.704   205.768    70.734
      1002    341.549   554.834    74.180  1800.000  2400.000  1000.000
      2002    346.784   254.925    68.944  2400.000  1800.000  1000.000
```

fig.9 Input data

Nº	X	Y	Z	Discrepancies											
101	1600.20	2200.02	996.97	-.23	-.03	-.02	.23	.03	.02						
102	1800.25	2200.04	1005.01	-.16	.03	-.08	-.05	-.03	.09	.22	0.00	-.01			
201	1600.16	1799.92	998.98	-.02	-.18	0.00	-.10	.15	0.00	.12	.03	0.00			
202	1800.16	1799.97	994.99	.10	-.14	-.02	-.23	.18	.05	-.15	-.06	-.05	-.20	.01	.03
				.17	0.00	-.01	.31	.01	0.00						
103	2000.31	2199.96	1010.01	-.19	-.08	0.00	.19	.08	0.00						
203	2000.06	1799.98	991.98	-.21	.04	-.03	-.23	-.05	.01	.44	0.00	-.01			
301	1600.02	1400.02	1010.01	-.28	.15	.01	.28	-.15	-.01						
302	1799.96	1400.06	1012.99	-.27	-.03	0.00	-.17	.20	0.00	.39	-.14	0.00			
303	2000.05	1400.03	1014.97	-.34	.14	0.00	.34	-.14	.01						
401	1000.15	1799.93	1005.00	0.00	0.00	0.00									
402	1400.32	2200.05	1007.99	-.09	0.00	0.00	.09	0.00	0.00						
501	1400.26	1399.85	1015.00	0.00	0.00	0.00									
601	1800.38	999.78	1004.99	0.00	0.00	0.00									
602	2200.50	1399.96	1006.99	-.04	-.01	-.01	.04	.01	.01						
403	1800.47	2600.12	1012.01	0.00	0.00	0.00									
502	2200.54	2200.05	1017.01	0.00	0.00	0.00									
603	2600.61	1799.98	1010.01	0.00	0.00	0.00									

1st iteration

101	1599.99	2200.00	996.99	0.00	0.00	0.00	0.00	0.00	0.00						
102	1799.99	2200.00	1004.99	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00			
201	1599.99	1800.00	998.99	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00			
202	1800.00	1799.99	994.99	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
				0.00	0.00	0.00	0.00	0.00	0.00						
103	2000.00	2200.00	1009.99	0.00	0.00	0.00	0.00	0.00	0.00						
203	2000.00	1799.99	991.99	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00			
301	1599.99	1399.99	1010.00	0.00	0.00	0.00	0.00	0.00	0.00						
302	1799.99	1399.99	1013.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00			
303	2000.00	1399.99	1015.00	0.00	0.00	0.00	0.00	0.00	0.00						
401	999.98	1800.00	1004.99	0.00	0.00	0.00									
402	1399.99	2200.00	1007.99	0.00	0.00	0.00	0.00	0.00	0.00						
501	1399.99	1399.99	1014.99	0.00	0.00	0.00									
601	1800.00	999.98	1004.99	0.00	0.00	0.00									
602	2200.00	1399.99	1007.00	0.00	0.00	0.00	0.00	0.00	0.00						
403	1799.99	2600.01	1011.99	0.00	0.00	0.00									
502	2200.00	2200.00	1016.99	0.00	0.00	0.00									
603	2600.01	1799.99	1010.00	0.00	0.00	0.00									

2nd iteration

fig.10 Output

5. Gross error detection

As central memory requirements are not high, it could be inexpensive to use COBLO for previous strip formation and gross errors detection. This stage is almost unavoidable when working with large blocks and it turns out to be imperative in Bundles Adjustment for finding good initial values for the unknowns (5). The printing of the discrepancies mentioned in the last paragraph and shown in Fig.10 will be of great help at this stage.

6. Central Memory requirements and CPU times

Regarding these two items it is difficult to compare our obsolete IBM 1620 computer with modern ones.

A week spent by the author in Buenos Aires last december was hardly enough to verify the FORTRAN IV version and to process some few blocks (though translating FORTRAN II into FORTRAN IV seems to be simple, it is an awkward task, specially if direct access instructions are present). Nevertheless, some interesting conclusions could be drawn :

1) The program has only 900 FORTRAN instructions which require almost 30 K bytes in an IBM 370/125 computer.
2) The working area requirements depend, for the time being, on the block size: 50 models need 10 K bytes, 100 models need 20 K bytes and 200 models need 40 K bytes . With very large blocks or very small computers special dynamic storage techniques will be necessary. The program has been recently restudied and it seems not to be difficult to store externally all integer variable arrays which are still in central memory. Thus, the central memory requirements will be independent of the block size and will not surpass 5 K bytes.
3) A proof carried out with a block of 20 models, with 3 plan height iterations resulted in a CPU time of 18 seconds per model. According to information obtained from IBM people in Tucumán, an IBM 370/125 is about 8 times slower than an IBM 370/145. Thus, the above mentioned proof would take a CPU time of only 2 seconds per model in an IBM 370/145, which is considered a good value if compared with CPU times given in (7) and (8).

7. Conclusion

The last version of our block adjustment program COBLO seems to be promising because it can handle, in small computers, any type of block regardless of shape, overlap and point distribution.

Several blocks of fictitious and real data were already processed with this program and a good performance was observed.

As most of the proofs were carried out at our old IBM 1620 computer, only small blocks were adjusted because the machine times in this computer are extremely large for blocks of more than 20 models (it should be born in mind that CPU time in an IBM 1620 can be more than 400 times

greater than CPU time in an IBM 370/145).

Fortunately within few months and after many years of hard work, our old IBM 1620 will give way to a very modern and powerful computer recently acquired by our University. Then, we shall be able to carry out proofs with large blocks and to perform some research on subjects like, gross error detection, control saving by using cross strips, blocks with strips differing strongly in scale, etc.

REFERENCES

(1) Jordan/Eggert/Kneissl: "Hanbuch der Vermessungskunde". Band III a/3. Stuttgart 1972.

(2) Ackermann F., Ebner H. und Klein H.:" Ein Programm-Paket für die Aerotriangulation mit Unabhängigen Modellen". Bildmessung und Luftbildwesen. Heft 4. 1970 .

(3) Juliá J.E. :" Compensación en Bloque Rigurosa por modelos independientes ". Revista Cartográfica IPGH N° 33. Junio 1968.

(4) Juliá J.E. :" Un Programa Fortran de compensación en bloque por modelos independientes apto para bloques completamente arbitrarios " . Cátedra de Fotogrametría. Instituto de Geodesia. Universidad Nacional de Tucumán.

(5) Tewinkel G.C. : " A Bundles Method of block Aerotriangulation ". Revista Cartográfica. IPGH N°26 Junio 1976.

(6) Kubik J. : " Survey of Methods in analytical block triangulation. ITC Publication, A 39.

(7) Schwidefsky-Ackermann: " Photogrammetrie " . B.G. Teubner, Stuttgart, 1976.

(8) Klein H. : " Block Adjustment programms for minicomputers " Nachrichten aus dem Karten-und Vermessungswesen. Institut für Angewandte Geodäsie. Frankfurt, 1978.