NUMERICAL SOLUTION OF A BLOCK ADJUSTMENT BY STABLE QR FACTORIZATION

Fabio Crosilla, Ivano Iob
Istituto di Urbanistica e Pianificazione
Università di Udine
Via Larga 42, 33100 Udine, Italia

ABSTRACT
The solution of a great, sparse, over-determined linear or linearized system of equations needs euristics such as 'minimum degree' and 'dissection' to control fill-in so to reduce costs and resources. Euristics applied up to now are general and have been studied to be coupled with the method of solution of a normal system of equations. The purpose of this paper is double: to present a new euristic for a sub-class of problems including adjustment procedures and solve the matrix block structure so obtained by a modified version of the classic QR factorization. In this way, fill-in results completely confined into blocks that, according to their location within the matrix, allow to obtain a method of solution that is stable, less expensive and efficient for parallel implementations.

KEY WORDS: fill-in, euristic, QR factorization.

Introduction

Sparse systems of equations are often solved with indirect methods which don't cause fill-in (i.e. elements previously equal to zero become not null). The sequence of points computed during the interactive process can diverge for a general initial point which is frequently difficult or impossible to find in order to ensure the convergence to the solution. On the contrary, direct methods permit to compute the solution in one step but they produce fill-in which has unpleasant effects on the efficiency of the algorithms. Since the problem of minimum fill-in is NP-complete, several euristics have been studied and proposed up to now to reduce it. These euristics alter the position of not null coefficients in a way suitable with the particular method chosen to compute the solution. In practice, they are algorithms for the rows and columns permutation of the linear or linearized system of equations; but, since permutations done with matrix products can cost enought in terms of operations ($O(n^3)$), they are implemented by an opportune graph connected with the matrix structure. Given the linear system $\mathbf{Ax=b}$, the graph G=(V,E) is defined as follows:
- $V=\{n_1,\dots,n_n\}$ each variable x_i is represented within the graph by the node n_i;
- $E=\{(n_i,n_j)\mid a_{i,j}\neq 0,\ a_{i,j}\in A\}$, every not null coefficient produces one edge.

The biunivocal correspondence between the node n_i and the variable x_i permits to number nodes from 1 to n and to transfer the order to the variables so that these one are arranged without an explicit matrix product.
In this paper the authors propose a new euristic with the double aim of improving the efficiency compared to classical procedures and of resuming the QR factorization method to solve over-determined linear systems of equations. The first purpose is pursued considering a sub-class of problems for which a new type of graph will be defined in order to take advantage of the restrictions imposed. Relating to the second purpose, it is well known that an over-determined linear system of equations doesn' t admit exact solutions and, alternately, it is generally computed the vector $\boldsymbol{x}$ which minimizes $\|\mathbf{Ax-b}\|_2$ (least squares problem). The literature suggests different direct methods of solution such as:
- normal equations : $A^t\mathbf{Ax} = A^t\mathbf{b}$;
- QR decomposition : $\min\|\mathbf{Ax-b}\|_2 = \min\|\mathbf{Rx}-Q^t\mathbf{b}\|_2$ with $A=QR$;
- Single Value Decomposition (S.V.D.).

Neglecting the last one, used for very bad conditioned problems, the QR factorization works well in our case. In practice, it is rarely utilized because it requires more floating point operations (flops) than normal equations method. That's true for general matrices, but a good management of the sparsity can significantly reduce the number of flops giving, at the same time, all advantages derived from the stable process which transforms matrix A in the triangular matrix R.

1 MATRIX STRUCTURING

1.1 The sub-class problems

Let' s consider the sub-class of problems whose sets of the variables X and the equations verify the following hypothesis:

Hypothesis 1.1. Let X be the set of variables, $X_1,X_2,\dots X_n$ subsets of X and suppose that:
1) the set of variables X is partitionable into $X_1, X_2, \dots ,X_n$.
2) all equations are: $f(X_i,X_j)=0$, for $i,j=1 \dots n$.

The first request is simply to understand; it imposes that the set of variables X can be divided in 'n' sets $X_1, X_2, \dots ,X_n$ so that :

- $\bigcup_{i=1..n} X_i = X$
- $X_i \neq \varnothing$
- X_i intersects $X_j \neq \varnothing$.

The second request needs more specifications because $f(X_i,X_j)=0$ represents a generic equation where the not null coefficients belong only to the sets X_i and X_j. In other words, the partition of the set X is made so that every equation concerns only variables of two or one (index 'i' can correspond to 'j') sets of the partition. The meaning of this assumption will be more manifest in the next paragraph; now, let' s define the graph G=(V,E) for structuring the matrix.

<u>Definition 1.1</u>. Under the assumption of the hypothesis 1.1, let G=(V,E) be the graph defined in the following way:
- $V = \{n_1, n_2, \ldots , n_n\}$;
- $E = \{(n_i, n_j) | \exists\ X_i,\ X_j \text{ and } f(X_i, X_j)=0\}$.

Every set X_i is represented in the graph by the node n_i and all equations considering the variables of X_i and X_j by the edge (n_i, n_j).
Once the graph is built, the next step is to number its nodes and edges and, using this order, to rearrange variables and equations respectively. Formally this operation corresponds to define two functions:

$$\mathcal{V} : V \dashrightarrow \{1,2,\ldots,n\} \qquad \mathcal{V}(n_i)=j$$
$$\mathcal{E} : E \dashrightarrow \{1,2,\ldots,m\} \qquad \mathcal{E}((n_i,n_j))=k$$

where 'n' is the number of nodes and 'm' the number of edges.
The classic approach also structures matrices in a similar way, but it associates one node to every variable and an edge to every not null coefficient of the matrix. The proposed definition 1.1 of the graph implies two advantages. The first one is the reduction of the dimension of the sets V and E; infact the classic approach creates a biunivocal correspondence between one node and one variable, so that $|V|=|X|$, while definition 1.1 associates one node to one set X_i of the variables. In this case $|V|$ is usually less then $|X|$ and only in the worst case $|V|=|E|$. The difference becomes more evident if we evaluate the dimension of the set E because one edge includes all the equations between two sets of the partition (see the table of the next paragraph).
The second advantage regards the linearized systems of equations. In this case the solution is obtained at the end of an interactive process in which many different linear equation systems are solved. Chosen an initial point, the first computed solution is considered a better approximation of the not linear system and so it's used for another linearization that produces a new linear system to solve. In the classic approach the graph must be rebuilt and the nodes renumbered at every step because set E changes. On the contrary using the graph defined at 1.1, functions $\mathcal{V}()$ and $\mathcal{E}()$ have to be fixed just one time, at the beginning. This is due to the fact that the graph is defined without any assumption on the linear system; not only, hypothesis 1.1 doesn't require any information on the type of the equation (i.e. integral, differential, trigonometric,..); the only important thing is which variables are related by the equations; consequently, the graph is not related to the linear system but, more properly, to the nature of the problem.

1.2 <u>Block adjustment application</u>

Let us consider the block adjustment relationships, by independent models, for the coordinates of the same tie point relating to two different models.

$$\lambda_i R_i \begin{bmatrix} x_{h,i} \\ y_{h,i} \\ z_{h,i} \end{bmatrix} + \begin{bmatrix} T_{x,i} \\ T_{y,i} \\ T_{z,i} \end{bmatrix} - \lambda_j R_j \begin{bmatrix} x_{h,j} \\ y_{h,j} \\ z_{h,j} \end{bmatrix} - \begin{bmatrix} T_{x,j} \\ T_{y,j} \\ T_{z,j} \end{bmatrix} = 0$$

where the vector $(x_{h,i}, y_{h,i}, z_{h,i})^t$ contains the model coordinates of the h^{th} tie point in model 'i'; the vector $(T_{x,i}, T_{y,i}, T_{z,i})$ contains the three translation parameters of model 'i'; R_i is the rotation matrix of model 'i' and, finally, 'λ_i' is the scale factor. The same parameters can be defined for model 'j'. The relationship contains fourteen variables:
- $\lambda_i, \Omega_i, \Phi_i, K_i, T_{x,i}, T_{y,i}, T_{z,i}$
- $\lambda_j, \Omega_j, \Phi_j, K_j, T_{x,j}, T_{y,j}, T_{z,j}$

representing the seven unknown parameters associated to each model.
For control points the following equations can be defined:

$$\begin{bmatrix} X_h \\ Y_h \\ Z_h \end{bmatrix} = \lambda_i\ R_i \begin{bmatrix} x_{h,i} \\ y_{h,i} \\ z_{h,i} \end{bmatrix} + \begin{bmatrix} T_{x,i} \\ T_{y,i} \\ T_{z,i} \end{bmatrix}$$

where the vector $(X_h, Y_h, Z_h)^t$ contains the ground coordinates of the h^{th} control point. In this case, the variables are:
- seven $(\lambda_i, \Omega_i, \Phi_i, K_i, T_{x,i}, T_{y,i}, T_{z,i})$ if the control point is known both in altimetry and planimetry.
- eight $(\lambda_i, \Omega_i, \Phi_i, K_i, T_{x,i}, T_{y,i}, T_{z,i}, Z_h)$ if the control point is known only in planimetry.
- nine $(\lambda_i, \Omega_i, \Phi_i, K_i, T_{x,i}, T_{y,i}, T_{z,i}, X_h, Y_h)$ if the control point is known only in altimetry.

Now, suppose to index all the control points in the following way;
- from 1 to p all points known in altimetry;
- from p+1 to q those one known in planimetry;
- from q+1 to r those one known in planimetry and altimetry.

If
- $M_j = \{\lambda_i, \Omega_i, \Phi_i, K_i, T_{x,i}, T_{y,i}, T_{z,i}\}$ i=1..n
- $N_j = \{X_j, Y_j\}$ i=1...p,
- $N_j = \{Z_j\}$ i=p+1...q,

then, the set of variables X can be written in the following way:

$$X=(\bigcup_{i=1..n} M_i)\bigcup(\bigcup_{i=1..p} N_i)\bigcup(\bigcup_{i=p+1..q} N_i)$$

Note that if all points are known both in altimetry and planimetry, the previous expression reduces to:

$$X = \bigcup_{i=1..n} M_i$$

that is to say, the set of variables corresponds only to the orientation parameters of the models. Known points in planimetry or in altimetry add to the linear system one or two new variables.
It' s manifest that:
a) $(M_1, \ldots, M_n, N_1, \ldots, N_q)$ is a partition of X;

b) all equations are of the type:
- $f(M_i,M_j) = 0$ $i,j=1..n$,
- $f(M_i,N_j) = 0$ $i=1..n$, $j=1..q$.

So, block adjustment by independent models belongs to the sub-class of problems satisfying the definition 1.1.
Since every model is rapresented by the seven orientation parameters M_i and every control point known in altimetry or planimetry by the added variables N_j every node has a physical meaning, as underlined later too.
Let' s now define :

$$V=\{n_1,..,n_n,n_{n+1},..n_{n+q}\}$$

and, if:

$E_1 = \{(n_i,n_j)|i,j=1...n, \exists$ one point connecting model 'i' and 'j'}

$E_2 = \{(n_i,n_j)|j=n+1...n+q, i=1...n$ if the control point known in planimetry or altimetry has the image coordinates in the model 'i'}

then, the set of the edges is:

$$E = E_1 \cup E_2$$

The above definitions of the sets V and E summarize an immediate way to build the graph, underlining at the same time that the graph is properly related to the nature of the problem instead of the matrix of the linear or linearized system. Although this, the reordering of the variables done to preserve the order of numbered nodes determines the location of not null coefficients and, so, the structure of the matrix. Infact, the generic equation $f(x_1,x_2,x_3,x_4)=0$, after linearization, becomes:

$$a_1x_1 + a_2x_2 + a_3x_3 + a_4x_4 = b$$

where a_1,a_2,a_3,a_4 and b are the coefficients whose position within the matrix is determined by the location of the variables and not by the value of the coefficients which depends on the type of the equation.

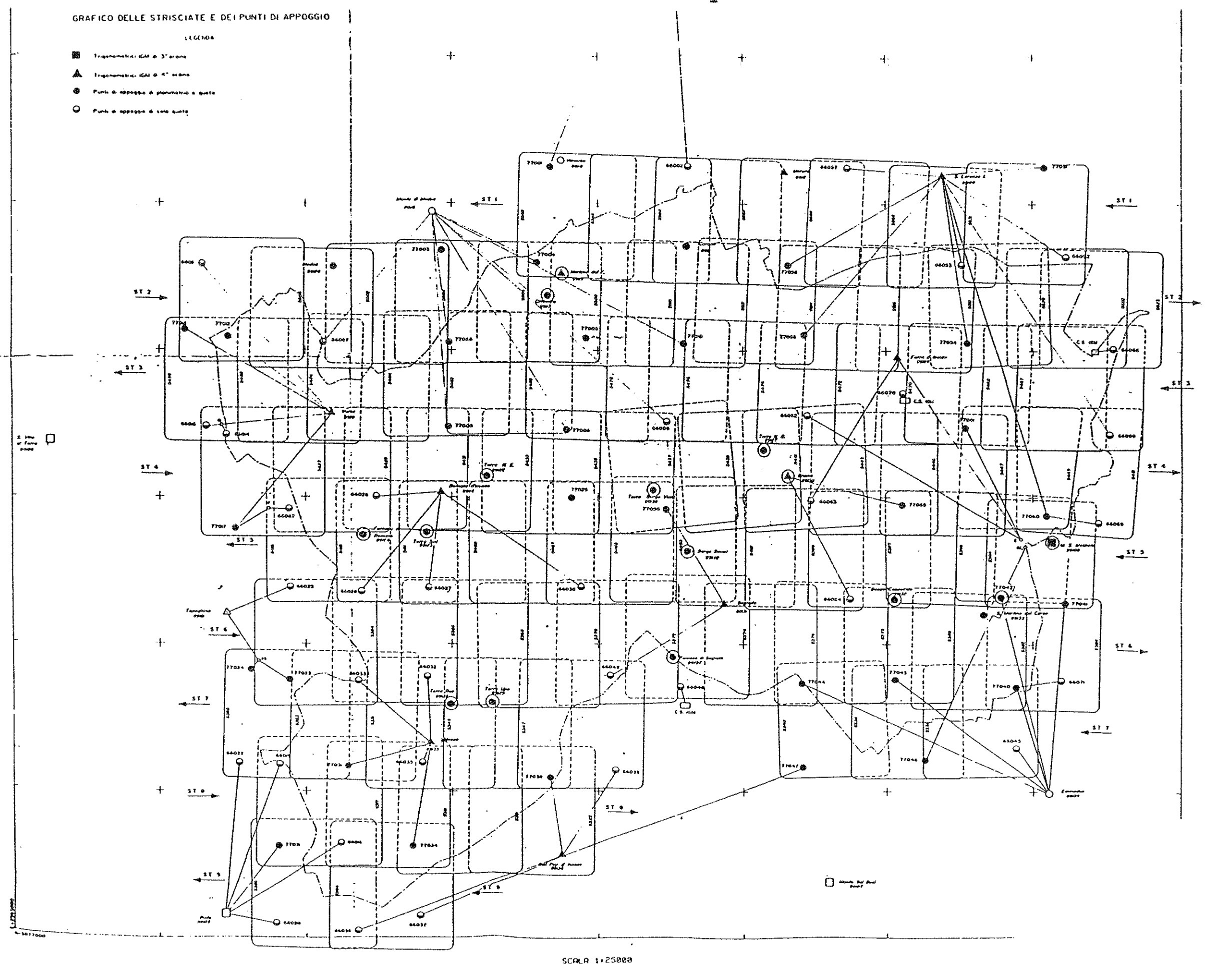


Figure 1. Scheme of photos ralating to the experimental example of block adjustment by independent models for the realization of numerical cartography in the Friuli-V.G. region.

Figure 1 reports the scheme of the photos and the control points for an experimental example of block adjustment relating to the realization of numerical cartography in the Friuli-V.G. region; figure 2 shows the graph relative to the example reported in figure 1. Note how the location of the models is repeated within the distribution of the nodes, how the edges describe precisely which models get together the connecting points and in which model the control points lie.

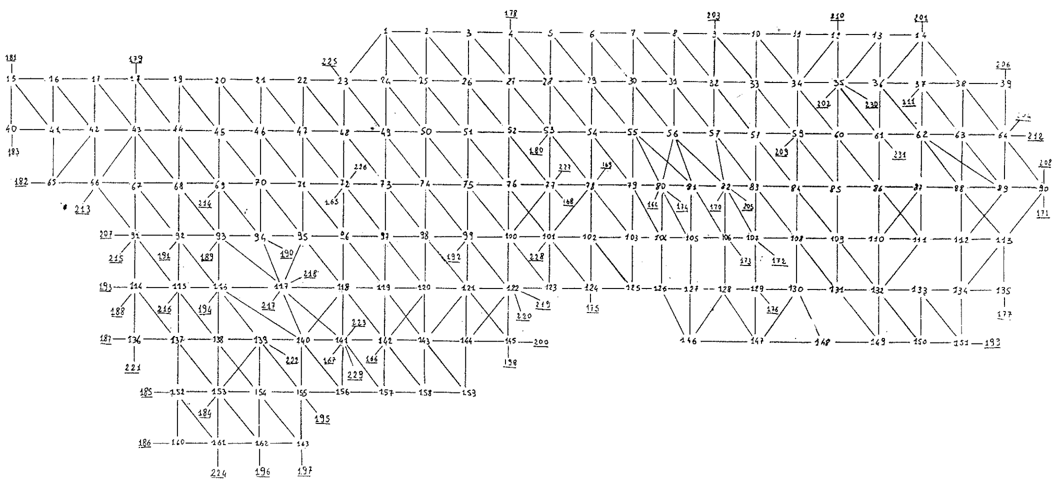


Figure 2. The graph obtained applying the definition 1.1

The following table summarizes the differences, in terms of graph dimensions, between the classical and the proposed graph.

	classical	proposed
$\|V\|$	1141	231
$\|E\|$	12500	458

The value of $|E|$ of the classic method is approximated; however, comparing the reported values, the difference of the costs for a procedure that handles graph with 458 edges instead of 12500 and 231 nodes instead of 1141 seems evident.

1.3 Numbering nodes and edges

The way for numbering edges is synthetized in the following procedure which assumes that nodes were previously ordered in the same way.

```
procedure   E
   num=1
  for   i=1   to    n
        let $n_k$ be the node so that $V(n_k)=i$
        let n_b be the number
        of brothers of node $n_k$
        for   j=1    to    n_b
             let $n_j$ be $j^{th}$
             brother of $n_k$
             if( not   used( $(n_i,n_j)$ ) )
             then
                 $E( (n_i,n_j) ) = num$
                 $(n_i,n_j)$ = used
                 num = num+1
             endif
        endfor
  endfor
endprocedure
```

The edges are numbered using a visit of the graph based on nodes order and labeling all edges not yet numbered; this operation is repeated for all the edges of the node n_i and for all nodes of the set V. As every edge (n_i,n_j) is considered twice, during the visit of the node n_i and of the node n_j, it is necessary to label the edge as 'used' by the function 'used()' when it is examined the first time. The effect of the procedure E is to create in the matrix A a low outline under which all elements are zero. This property is due to two facts: during the visit of the node 'i' all no labeled edges connect n_i with the brothers n_j whose indices 'j' are greater then 'i', and, futhermore, the no null coefficients of n_i (i.e. the coefficients of the variables associated to n_i) are always at the right side of those of n_{i-1} and at the left side of those of n_{i+1} (because the variables reflect the order of the nodes). In terms of matrix, the equations associated with the edge (n_i,n_j) put the not null coefficients of n_j always at the right side of those of n_i.
Once the numbering procedure for the edges is fixed, the problem is transferred to the nodes. This will be solved in two parts consisting of:
1) building a partition of the set V of nodes by three steps which determine:
- a partition of V into $(K_1,K_2,...,K_r,I)$
- a partition of I into $(KC_{11}...KC_{1\log(r)})$
- an arrangement of the last partition.
2) Numbering the nodes following the criterion synthetized by the next procedure based on the partition of the first step.

```
procedure    V
   num = 1
   for   i=1    to    r
      for   j=1    to   $|K_i|$
      let $n_j$ be the $j^{th}$ node of $K_i$
          $V( n_j ) = num$
      num = num +1
    endfor
   endfor
   for   j=1   to   $\log_2 r$
      for   i=1    to   $r/2^j$
          for   h=1    to   $|KC_{i,j}|$
           let $n_h$ be the $h^{th}$
           node of $KC_{i,j}$
              $V( n_h ) = num$
           num = num +1
         endfor
      endfor
   endfor
endprocedure
```

The nodes are numbered following the next order of the sets of the partition

$$K_1,\dots,K_r,KC_{1,1},\dots,KC_{r/2,1},\dots,KC_{1,\log(r)}$$

where log(r) is $\log_2$.

First partition.
During this step, a partition of V into $(K_1,K_2, \dots ,K_n,I)$ is determined such that an edge connecting two nodes belonging to two different K_h , K_k, (k,h=1..r and $k \neq h$) doesn' t exist.

Definition 1.2. Given a graph G=(V,E), we define "first partition" of V the sets $(K_1,K_2, \dots ,K_n,I)$so that :
1) $(K_1,K_2, \dots ,K_n,I)$ is a partition of V
2) not $\exists$ $(n_i,n_j)\varepsilon E$ and $k,h=1..r$, $k\neq h$, so that: $n_i\varepsilon K_h$, $n_j\varepsilon K_k$

The effects in terms of the structure of the matrix A, are visible in figure 3.

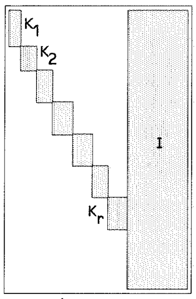


Figure 3

The restriction produced by the definition 1.2 is obvious: the request that there isn't an edge that connects two nodes belonging to two different sets K_h and K_k produces a diagonal of blocks; in fact, there aren' t equations between corresponding variables and so, over and under each K_i diagonal block all matrix coefficients are zero. Block I, on the right side, gathers the equations related to those edges which connect one node of I and one of K_i (the part away all diagonals blocks), or two nodes of I (the part below K_r).

Second partition.
During the second step, the matrix block I is also structured in some way. The idea is to locate a partition into the set I. A good way, particularly favourable for the QR decomposition method, is to transform this block into diagonals of blocks placed side by side. We don' t explain the reasons of this conjecture because it would be necessary to known how QR factorization transforms the structured matrix A in the triangular form R.

Definition 1.3 Given the first partition $(K_1,K_2, \dots ,K_n,I)$ of V, we define "second partition" the sets:
$K_1, \dots ,K_n,KC_{1,1},\dots,KC_{r/2,1},\dots,KC_{1,\log(r)}$
so that:
- $\forall$ $n_i\varepsilon KC_{h,k}$ $k=1..\log_2(r)$, $h=1..2^{(k-1)}$
- $\forall$ $n_j\varepsilon K_p$ $p=1..n$
the edge (n_i,n_j) belongs to E and

$$p = \text{int}((p-1)/2^k) + 1$$

where the function int() returns the value truncated to the integer part of the division. In terms of matrix structure, the consequence of this definition is visible in the figure 4. On the right side of the first diagonal of blocks, obtained at the previous step, there are $\log_2 r$ new diagonals in which the number of blocks halves proceeding to the right side.

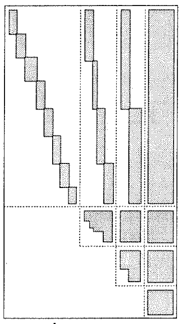


Figure 4

Note that the lower part of the matrix is a block triangular; actually, this definition is misleaded. Now, having a look at figure 11, where the structure of the matrix relative to the experiment discussed at the end of this paper, is reported. The lower part of the matrix is similar, although not equal, to the upper one; to discriminate there are the few blocks (bad blocks) not present in the right matrix of the figure 11. During QR factorization these blocks produce very bad effects increasing the costs (flops and memory); if possible, the best thing to do is to eliminate them ensuring a repetition of the upper structure into the lower part too (compatibility of blocks location).

Arrangement.
Bad blocks can be eliminated simply shifting on the right side, to a greater level, some nodes. This operation can be made only when the second partition is determined; in the same way, the second partition can be done when the first one is completed. It' s sufficient to scan all the nodes belonging to $KC_{i,j}$ and to verify if they produce a 'bad' block; in this case, the node must be moved to an appropriate block of major level. In order to make this process working correctly, it' s important for the scan operation to proceed from the first to the last level; so, transfering nodes to an upper level doesn' t damage previous block diagonal structure. The only consequence is that the blocks of the last level become larger and larger. As this blocks are sparse and have got great dimension, this effect is unpleasant. If a lot of nodes should be moved, some blocks can disappear and the level structure may become more deficient. At the end, the coefficient matrix, so structured, assumes the form reported in the figure 5.

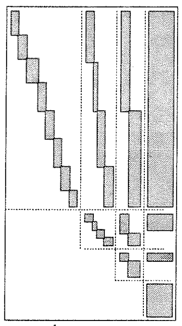


Figure 5

1.4 Some remarks

The two partitions and the arrangement of the third step can be implemented with simple procedures and efficient data structures (Iob, 1991), so that the cost is:

$$C = k\ |E| \qquad 5 \le k \le 10$$

with the constant k depending on the particular procedures that can be added to improve the structure. Unitary cost is assigned to visit a node. The cost is not only polinomial but linear with low coefficient. As regards data, it is possible to implement simple data type, (data structures and operations), that, in terms of required memory for storing graph and all sets, needs about

$$M = (8n + 2m)*\mathrm{sizeof(integer)}$$

where the function sizeof(integer) returns the number of words necessary to store an integer.

2 COMPUTING SOLUTION

2.1 The QR decomposition

Once the matrix of the linear or linearized system of equations is structured as in figure 5, the next step is the computation of the solution. As mentioned before, the stable QR decomposition was chosen, adapting it to the structure of the matrix obtained previously, instead of the well studied and used normal equations method, which is less robust against gross errors. The main purpose is to show that the euristic of the previous paragraph is good enough, because it permits to implement efficient QR decomposition with low costs and few memories. The matrix A is transformed in the triangular matrix R by the 'l' meta-steps, where 'l' is the number of level, of the following procedure:

```
**procedure** QR
  **for**   k=1   **to**   l
      $A_{k+1} = Q_k A_k$
      $A_{k+1} = P_{k+1} A_{k+1}$
  **end**
**endprocedure**
```

where: $Q_k=\mathrm{diag}(I_k,Q_1^{(k)},\dots,Q_b^{(k)},J_k)$, $b=r/2^{(k-1)}$, I_k and J_k are identity matrices, $Q_i^{(k)}$, i=1..b, are orthogonal and they decompose the corresponding first diagonal blocks $A_i^{(k)}$ into the products $R_i^{(k)}=Q_i^{(k)}*A_i^{(k)}$ $(i=1..r/2^{(k-1)})$, with $R_i^{(k)}$ triangular matrices. The following figure synthetizes how the first meta-step of the QR procedure modifies the structured matrix $A=A_1$ of the linear system.

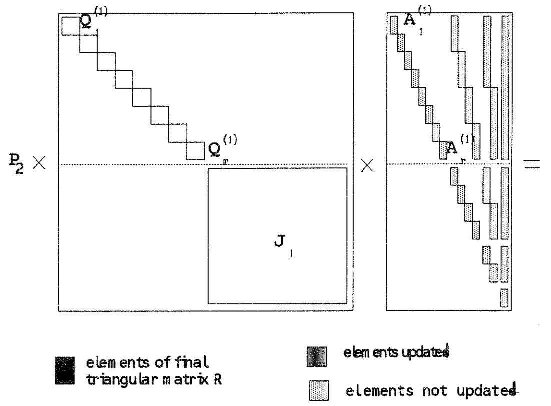


Figure 6

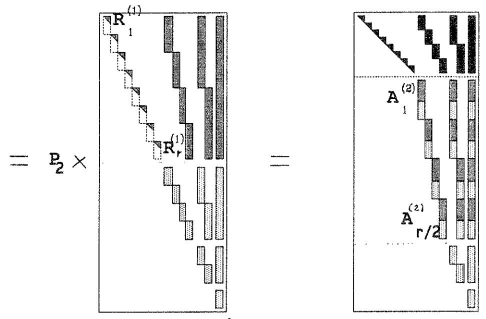


Figure 7

The first meta-step uses $Q_1=\mathrm{diag}(Q_1^{(1)},\dots,Q_r^{(1)}J_1)$ to transform the first 'r' blocks $A_1^{(1)},..,A_r^{(1)}$ of the first upper diagonal into the triangular form $R_1^{(1)},..,R_r^{(1)}$; the blocks on the right side of $A_i^{(k)}$ are updated and the fill-in process take place within them; but, as of Q_1 structure, it' s confined into the blocks themself. It' s not important to know which kind of the orthogonal matrices to prefer: the projection matrices of Givens or the reflection matrices of Householder; the choice can be done analyzing blocks and studying the fill-in process. J_1 is the identity matrix dimensioned in such a way that the lower part of the matrix, below the dot line, isn' t modified. Figure 7 describes the effects of the permutation matrix P_2; P_2 divides blocks, just factorized or updated, in two parts; it gathers all the upper pieces and shifts them upon the others creating the first part of the final matrix R. The lower parts of the blocks are joined together with the blocks of the next part of the matrix (under the dot line) and a new piece of the matrix A_2 is ready to be submited to the same process. The following two figures summarize the second step.

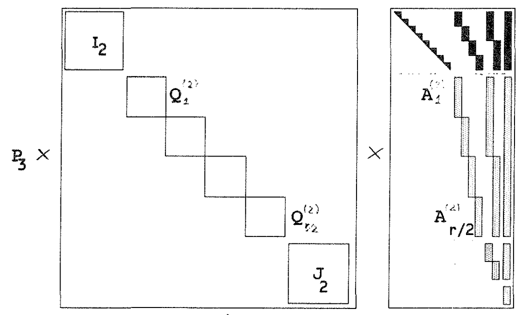


Figure 8

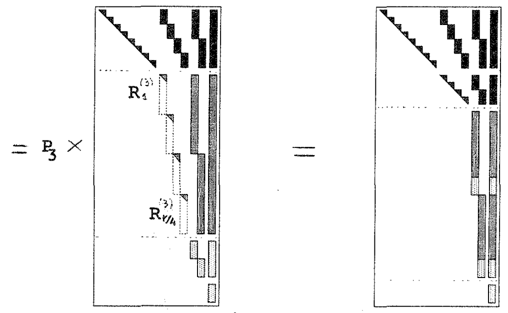


Figure 9

There are two identity matrices until the last step, in order to ensure that during each meta-step only the contiguous set of rows of the blocks (between the two dot lines) are interested by the transformation of the QR procedure.
It is important to note that every matrix $Q_i^{(k)}$, for $i=1..r/2^{k-1}$, can be computed independently and all the products $Q_i^{(k)}A_i^{(k)}$ can be done at the same time. This thing has direct consequence on the parallel implementation.
It' s simple to demonstrate that the transformation process of the QR procedure

$$A_{l+1} = P_{l+1}\ Q_l\ P_l\ Q_{l-1}\ \cdots\ P_2\ Q_1\ A_1$$

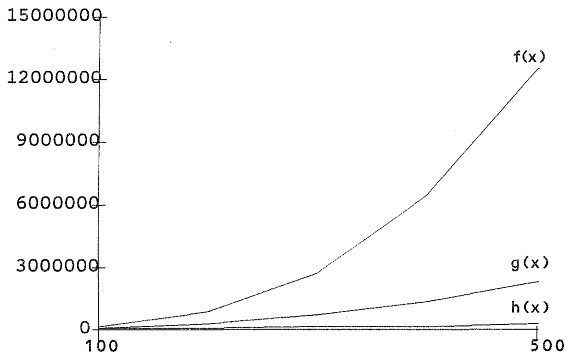
is truly orthogonal because the permutation matrices P_{i+1} and Q_i, $i=1..l$, are orthogonal.

2.2 Cost

To conclude this paragraph, we examine the benefits of the euristic on the QR decomposition solving method. We analyze the cost defined as the number of flops evaluated when $Q_i^{(k)}$ are the reflexive matrices of Householder. The matrix product $A_{k+1}=Q_k\ A_k$ is reduced to the block products shown in figures 6 and 8, althought it could be further translated into singular element product for the blocks of the last levels. So, assuming that the first partition create 'r' sets K_i and all the first diagonal blocks have the same dimension, i.e., if structured matrix A is m*n, each block is m/ra*n/ra, where a=(2r - 1)/r, then cost is:

$$C=(n/r)^3\log^2(r)(q-1)\text{ flops}\qquad m=qn$$

In general, every method to solve a generic over-determined system costs $O(n^3)$ but, for block adjustment by independent models the parameter 'r' is a function of 'n'; so the cost reduces to $O((n/r)^3\log^2(r))$. If $r=\sqrt{n}$ then $C=O(n\sqrt{n}\ \log^2(\sqrt{n}))$; as $\log^2(\sqrt{n})$ assume low values, matrices structured as in the figure 5 have low cost of factorization. It' s not necessary to remark the difference between n^3 and $n\sqrt{n}$; the second function is enormeously less than the first one; see the following graphic where the case $r=\sqrt[3]{n}$ is also reported.



$f(x)=n^3$; $g(x)=n^2\log^2(\sqrt[3]{n})$;
$h(x)=n\sqrt{n}\log^2(\sqrt{n})$
Figure 10

If $r=\sqrt{n}$, the cost is h(x); for n varing from 100 to 500, the difference of increase between n^3 and $n\sqrt{n}\ \log^2(\sqrt{n})$ is so great that h(x) is very near the 'x' axes.

2.3 Computing variance and covariance

It' s also important to compute the variances and covariances associated with the estimated vector of unknowns. This is done computing the matrix $(A^tA)^{-1}$. But, as the decomposition A=QR implies that:

$$A^tA = R^tQ^tQR = R^tR$$

with R triangular, then

$$(A^tA)^{-1} = (R^tR)^{-1}$$

The same procedures can be used as for the method of solution of the normal equations, where $A^tA = T^tT$, because T is a triangular matrix like R.

3 EXPERIMENTAL RESULTS

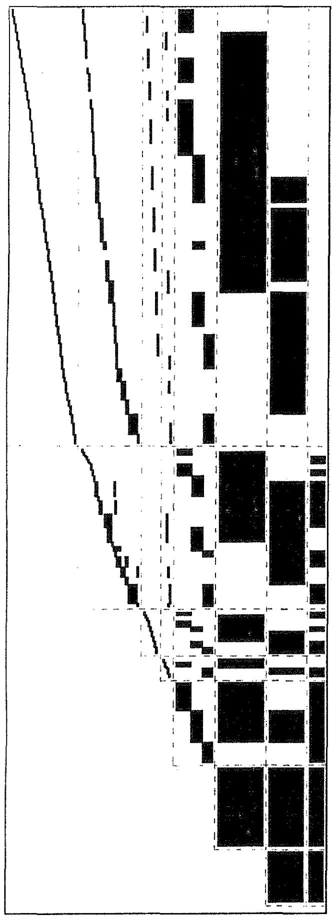
We report some results about an experiment of block adjustment performed by independent models according to the scheme in figure 1.

- models = 163
- tie points = 627
- control points of planimetry and altimetry = 51
- control points of planimetry = 5
- control points of altimetry = 63
- equations due to tie points = 3399
- equations due to control points known in planimetry and altimetry = 234
- equations due to control points known in planimetry = 15
- equations due to control points known in altimetry = 189

The system of equations has:
variables = 1272 equations = 3837

The graph has 231 nodes and 458 edges (figure 2).

The following two matrices refer, respectively, to the end of the second (on the left) and third step (on the right) of the procedures described in the first paragraph to structure the matrix of the linearized system of equations; the number of the sets K_i is forty-two (r=42). Reconstruction is in scale and so figure 11 represents the real dimensions of the blocks within the matrix. Below the first dot line, in the second level, blad blocks are visible. They are few and can be eliminated shifting on the right some nodes to ensure the compatibility of the block location. The effect of this process is the increment of the block dimension of the last level, also visible in the figure 11.
The assumption of equi-dimension of blocks used to compute the cost is not completely true, expecially for the last levels. Since these blocks are very sparse a compact techinique for storing not null elements can enormeously reduce the cost of their update (up to now the cost is evaluated assuming that the block products are implemented without taking advantage of the sparsity). Furthermore, as there exists algorithms (Iob, 1991) which produce r=42 blocks and $\sqrt{n}=\sqrt{1141}\approx 32$, one assumption is verified in defect and the other in excess; consequently, the conjecture $r=\sqrt{n}$ is acceptable. Then, in the graphic of the figure 10, the cost function corresponds to the lowest one.



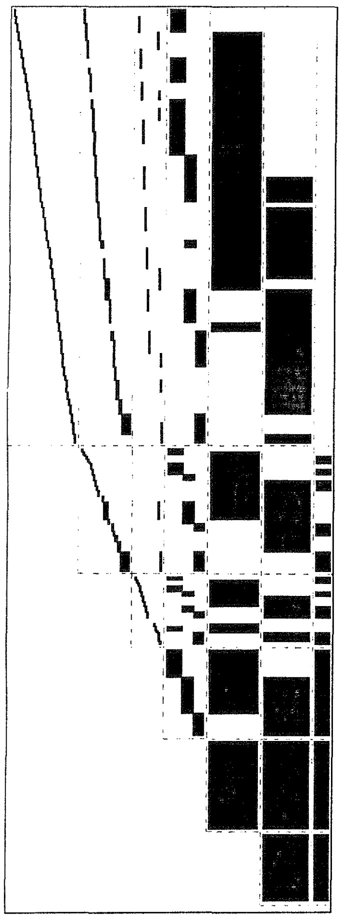


Figure 11

Bibliography

- Benciolini B., Mussio L. 1984. "Algoritmi di riordino delle incognite nelle compensazioni ai minimi quadrati", Ricerche di Geodesia e Topografia e Fotogrammetria, CLUP Milano.
- Bunch J.R., Rose D.J. 1976. Sparse Matrix Computation, Academic Press Inc, New York, San Francisco, London.
- Forlani G., Mussio L. 1994. "Il calcolo di una compensazione minimi quadrati", Ricerche di Geodesia e Topografia e Fotogrammetria, CLUP Milano.
- George A., Liu J.W. 1981. Computer Solution of Large Sparse Positive Definite System, Prentice-Hall Inc, Enflewood Cliff, New Jersey.
- Golub G.H., Plemmons R.J. 1981. "Large-scale geodetic least-squares adjustment by dissection and orthogonal decomposition", Large Scale Matrix Problems, North Holland, New York. Oxford.
- Iob I. 1991, "Un' alternativa al metodo della 'dissection' per la soluzione di grandi sistemi lineari sparsi e sovradeterminati", graduation thesis in Informatic Science, University of Udine.