Data space contraction in overdetermined inverse problems

Peter Mora

INTRODUCTION

Discretized inverse problems with more known data points than than unknown model points are overdetermined. The usual way to solve such inverse problems is by using optimization techniques that minimize a cost functional defined by some given norm. Equivalently, such optimization may be thought of as maximization of posterior probability given the prior probabilities. The most popular norm is the L^2 norm, which assumes Gaussian prior probability distributions. The methods of optimization are usually techniques of linear algebra that enable the minimum cost (maximum posterior probability) solution to be obtained. These techniques are computationally intensive and are frequently formulated as matrix equations though operator concepts may also be useful (Tarantola, 1983). Henceforth, the term inversion in this paper will refer to use of an optimization technique, in a discretized overdetermined problem, to solve for the unknown model vector.

A method of contracting the data dimensionality without destroying information useful in the inversion is presented in this paper. Hence, an inversion in contracted data space will be more efficient than an inversion in uncontracted data space in terms of both memory requirements and execution time. The saving in memory and computation time is greater for more overdetermined problems. These savings are such that the rapidly converging but computationally intensive Newton algorithm (of least squares) may become worthwhile.

The cost of the data dimension contraction technique is an increasing loss of accuracy as the data dimension is contracted more and more. In most practical situations, where some noise must be added to stabilize the inversion, this loss of accuracy should not be a major problem. The theoretical limit to the contraction is reached when the data dimension is the same as the model dimension. However, it has been found that such an extreme data contraction leads to a significant loss of accuracy while only a slight loss of accuracy is observed when the data dimension is not contracted below approximately twice the dimension of the model vector. Hence, the method is effective in reducing the data dimension of overdetermined problems to about twice the model dimension.

LEAST SQUARES INVERSION AND DATA CONTRACTION

A linear equation in matrix form is

$$\mathbf{Ax} = \mathbf{b} \tag{1}$$

Matrix **A** represents the equations describing the physical system. It has n_d rows n_m columns where n_d is the number of data points or observations and n_m is the number of unknown model points. The column vector $\mathbf{x}$ is the discretized physical model with dimension n_m and column vector $\mathbf{b}$ is the data vector with dimension n_d. For overdetermined problems, there are more data points than model points so $n_\mathrm{d} > n_\mathrm{m}$. The square error functional is

$$\mathbf{E}^*\mathbf{E} = (\mathbf{Ax} - \mathbf{b})^*(\mathbf{Ax} - \mathbf{b}) \tag{2}$$

where the $*$ indicates the conjugate transpose. The $\mathbf{x}$ derivative of the error functional is zero where the error is at a minimum. Taking the $\mathbf{x}$ derivative of equation (2) and rearranging the terms to solve for $\mathbf{x}$ yields the classic least squares formula (Newton algorithm for linear systems)

$$\mathbf{x} = (\mathbf{A}^*\mathbf{A})^{-1}\mathbf{A}^*\mathbf{b} \tag{3}$$

Consider a new square error functional rather than the one in equation (2)

$$\mathbf{E}'^*\mathbf{E}' = (\mathbf{A}'\mathbf{x} - \mathbf{b}')^*(\mathbf{A}'\mathbf{x} - \mathbf{b}') \tag{4}$$

where the prime indicates a contracted data space with $n_\mathrm{d}' < n_\mathrm{d}$. The condition that the new least squares solution be the same as that of the old unprimed problem is realized if the error functionals in equations (2) and (4) are identical functions of $\mathbf{x}$. This will be true if the result of dot products in both the primed and unprimed data space is the same.

The following is a heuristic development of the method of data space contraction. Consider equal dimensioned but otherwise arbitrary matrices **C** and **D** in the uncontracted data space. We want the dot product $\mathbf{C}^*\mathbf{D}$ to equal the dot product $\mathbf{C}'^*\mathbf{D}'$ in the contracted space. Let **C** and **D** be partitioned into submatrices,

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{bmatrix} \tag{5a}$$

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}_1 \\ \mathbf{D}_2 \end{bmatrix} \tag{5b}$$

Then we have the dot product

$$\mathbf{C}^*\mathbf{D} = \mathbf{C}_1^*\mathbf{D}_1 + \mathbf{C}_2^*\mathbf{D}_2 \tag{6}$$

Now consider the equations

$$\mathbf{C}' = \mathbf{C}_1 + \mathbf{C}_2 \ , \ \mathbf{D}' = \mathbf{D}_1 + \mathbf{D}_2 \tag{7a,b}$$

Hence,

$$\mathbf{C}'^*\mathbf{D}' = \mathbf{C}_1^*\mathbf{D}_1 + \mathbf{C}_2^*\mathbf{D}_2 + \mathbf{C}_1^*\mathbf{D}_2 + \mathbf{C}_2^*\mathbf{D}_1 \tag{8}$$

Observe that equation (8) looks the same as (6) except for the cross terms $\mathbf{C}_1^*\mathbf{D}_2 + \mathbf{C}_2^*\mathbf{D}_1$. It was this observation which provided the clue to the data space contraction method for if the cross terms could cleverly be made to vanish while the other terms were left uneffected, equations (6) and (8) would be identical. Then we would have a powerful method of reducing the data dimensionality in overdetermined problems without eliminating any useful information.

Now, instead of equation (7), we assume that there is a more general mapping matrix **M** that maps from the unprimed to primed data space. Let the form of this mapping be

$$\mathbf{C}' = \mathbf{M}\mathbf{C} = \mathbf{M}\begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{M}_1 & \mathbf{M}_2 \end{bmatrix}\begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{bmatrix} = \mathbf{M}_1\mathbf{C}_1 + \mathbf{M}_2\mathbf{C}_2 \tag{9}$$

Note that the mapping **M** has been divided into two parts, $\mathbf{M}_1$ and $\mathbf{M}_2$, which act independently upon $\mathbf{C}_1$ and $\mathbf{C}_2$. Furthermore, let the $\mathbf{M}_i$ be square matrices. Similarly, matrix **D** is mapped into the primed data space so the equivalent of equation (8) in primed data space is

$$\mathbf{C}'^*\mathbf{D}' = \mathbf{C}_1^*\mathbf{M}_1^*\mathbf{M}_1\mathbf{C}_1 + \mathbf{C}_2^*\mathbf{M}_2^*\mathbf{M}_2\mathbf{D}_2 + \mathbf{C}_1^*\mathbf{M}_1^*\mathbf{M}_2\mathbf{D}_2 + \mathbf{C}_2^*\mathbf{M}_2^*\mathbf{M}_1\mathbf{D}_1 \tag{10}$$

In order for the dot product in the unprimed space given in equation (6) to equal the dot product in the primed space given in equation (10) we require

$$\mathbf{M}_1^*\mathbf{M}_1 = \mathbf{I} \text{ and } \mathbf{M}_2^*\mathbf{M}_2 = \mathbf{I} \tag{11a}$$

$$\mathbf{C}_1^*\mathbf{M}_1^*\mathbf{M}_2\mathbf{D}_2 + \mathbf{C}_2^*\mathbf{M}_2^*\mathbf{M}_1\mathbf{D}_1 = 0 \tag{11b}$$

Equations (11a) indicate that the mapping matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ must be unitary (rotation) matrices. Equations (11b) are satisfied if we have the two conditions

$$\mathbf{C}_1{}^*\mathbf{M}_1{}^*\mathbf{M}_2\mathbf{D}_2 = 0 \text{ and } \mathbf{C}_2{}^*\mathbf{M}_2{}^*\mathbf{M}_1\mathbf{D}_1 = 0 \tag{11c}$$

Therefore, for equality of primed and unprimed dot products, it is sufficient for the rotation matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ to have the property (11c), which states that the dot product between any two differently mapped matrices is zero. In other words differently mapped matrices, such as $\mathbf{C}_1$ mapped with $\mathbf{M}_1$ and $\mathbf{C}_2$ mapped with $\mathbf{M}_2$, are uncorrelated.

The problem of verifying the data space contraction technique is now reduced to finding matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ with the described properties. Because these matrices are data dependent it should be possible to actually solve a system of equations for the mapping matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ given a particular data set (i.e. matrices matrices $\mathbf{C}$ and $\mathbf{D}$). However, this would defeat the purpose of using the data space contraction method, which is to increase the speed of an inversion algorithm and decrease memory requirements. Therefore, I will propose a particular set of mapping matrices $\mathbf{M}_i$ that do a reasonable job and give fairly accurate results as illustrated in the example.

The mapping matrices must destroy correlations between the different data matrices $\mathbf{C}_i$, so I propose what I term random mixing matrices. Such matrices move data from each location in the original unprimed space to an arbitrary random location in the primed space. Hence, each column of a mixing matrix zero except for one unitary element, which is located randomly but with the constraint that no two columns have the non-zero element in the same location. I have chosen this very specific form of the mixing matrices because multiplication by such a mixing matrix can be achieved efficiently. This is important in the context of this paper, which seeks to provide a more efficient method for solving overdetermined inverse problems. Therefore, multiplication of a sub-matrix $\mathbf{C}_i$ by a random mixing matrix $\mathbf{M}_i$ is achieved by

$$\mathbf{C}'_i(k,m) = \mathbf{C}_i(k'_i,m) \;,\; where \;\; k'_i = k'_i(k) \tag{12a}$$

Thus, the mixing is a random reshuffling of row indices. It is argued but not proven, that such a mixing matrix should largely destroy correlations that may have existed between the different $\mathbf{C}_i$ prior to mixing. Therefore, the conditions for identical dot products in both uncontracted and contracted data spaces, equations (11c), should be satisfied. It is expected that for large $n_d{}'$, the suggested random mix mapping will be good. This expectation is due to the belief that mixing somehow destroys correlations. Therefore, the sums over $n_d{}'$ in dot products between differently mixed matrixes $\mathbf{C}'_i$ are sums of random numbers of unknown probability distributions. By the central limit theorem such sums tend toward Gaussian distributions with width proportional to

$1/\sqrt{n_d'}$ so the error tends to decrease as n_d' increases.

The full mapping that corresponds to premultiplication by **M**, is achieved by adding all of the mixed submatrices

$$\mathbf{C}' = \sum_i \mathbf{C}'_i \tag{12b}$$

This sum is expected to introduce some inaccuracy by compounding the problem of computer precision.

It is noteworthy that the mapping **M** does not have an exact inverse because it is not square so it is not possible to return to uncontracted data space after a contraction mapping. In other words, the mapping matrix **M** does destroy information content although mapped matrices retain the same dot products (hence the usefulness of **M**).

EXAMPLE OF DATA SPACE CONTRACTION

A simple synthetic example demonstrates the data contraction method in the context of least squares inversion (figures 1 and 2). The example illustrates the method of data contraction and its accuracy for different degrees of contraction.

A random matrix **A** of dimension 256 times 16 was generated along with a random model vector **x** of dimension 16. The data vector **b** was then generated by multiplying **x** by **A**. The least squares solution was computed using equation (3). Subsequently, least squares solutions were computed in the contracted data space for a number of different degrees of contraction. The least squares model vectors are denoted $\mathbf{x}^{(c)}$ where the superscript $^{(c)}$ is the contraction factor. For example, a superscript of $^{(1/2)}$ indicates that the data space was contracted to one half of its original size. The result of the least squares inversion without data contraction is denoted $\mathbf{x}^{(1)}$. The model dimension in this example was 1/16 of the data dimension and least squares solutions $\mathbf{x}^{(1)}$, $\mathbf{x}^{(1/2)}$, $\mathbf{x}^{(1/4)}$, $\mathbf{x}^{(1/8)}$ and $\mathbf{x}^{(1/16)}$ were computed. These results are graphed in figure 1. All solutions overplot one another on this scale of plot so it is apparent that all degrees of data contraction yield fairly accurate least squares solutions. To highlight the errors, the difference between each least squares solution $\mathbf{x}^{(c)}$ and the uncontracted solution $\mathbf{x}^{(1)}$ was graphed in figure 2. From figure 2 it is obvious that the contracted results are quite accurate except for $\mathbf{x}^{(1/16)}$ where the maximum contraction was applied so the contracted data dimension equaled the model dimension.

Note that this synthetic example is an ideal situation in which there is essentially no correlation between any groupings of rows of **A** because **A** is random. For this reason, it is expected that the inversion result would have been just as good had the mixing

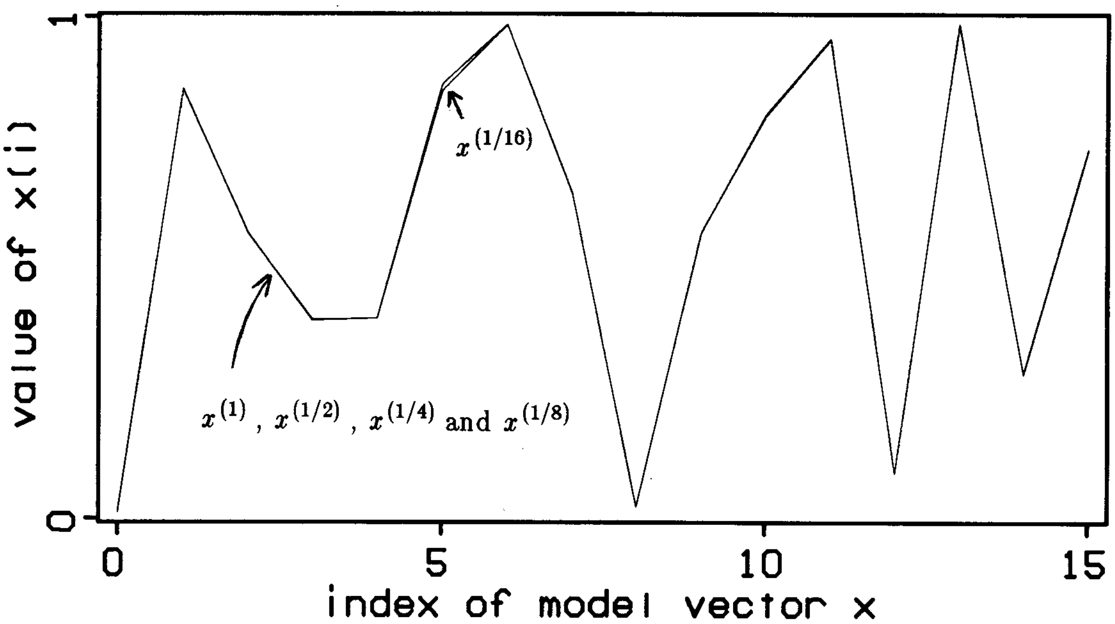


FIG. 1. Least squares solutions $\mathbf{x}^{(c)}$ computed with various degrees of data contraction. The superscript indicates the size of the contracted data dimension relative to the uncontracted data dimension.

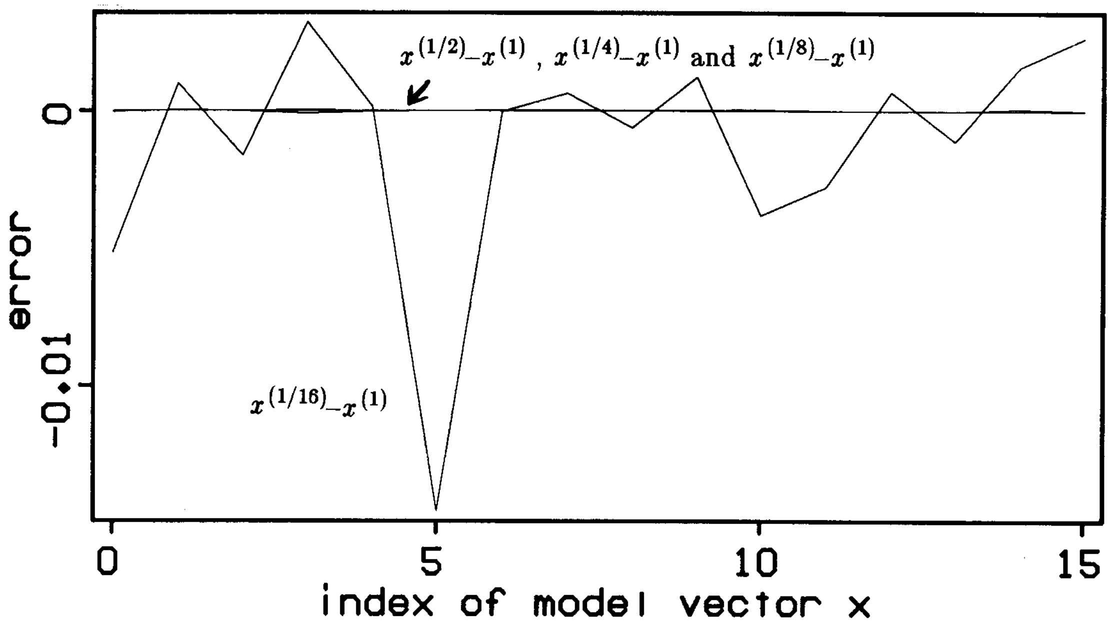


FIG. 2. Differences between the true least squares solutions and solutions where the data space has undergone various degrees of contraction. Differences are denoted $\Delta\mathbf{x}^{(c)} = \mathbf{x}^{(c)} - \mathbf{x}^{(1)}$. The superscript indicates the size of the contracted data dimension relative to the uncontracted data dimension.

part of the contraction (12a) been ignored and only the contraction sum (12b) been applied. For a more realistic example of contraction mapping results see Mora (SEP-41) where the technique has been utilized in an elastic inverse problem.

DATA SPACE CONTRACTION WITH COVARIANCES

Data space contraction may be applied in general by simply replacing unprimed values with primed values. The following section illustrates the method applied to the more general least squares formula with data and model covariance matrices. Consider the posterior probability density function

$$P = \mathit{constant}\ \exp-\frac{1}{2}\left[\Delta\mathbf{d}^{*}\mathbf{C}_{\mathrm{d}}^{-1}\Delta\mathbf{d} + \Delta\mathbf{m}^{*}\mathbf{C}_{\mathrm{m}}^{-1}\Delta\mathbf{m}\right] \tag{13}$$

where $\Delta\mathbf{d}$ is the data error, $\Delta\mathbf{m}$ is the model error and $\mathbf{C}_{\mathrm{d}}^{-1}$ and $\mathbf{C}_{\mathrm{m}}^{-1}$ are the inverse covariance matrices for the data and model spaces respectively. The maximum posterior probability solution is obtained by maximizing P and hence minimizing the error functional (Tarantola, 1982)

$$\mathbf{E}^{*}\mathbf{E} = \Delta\mathbf{d}^{*}\mathbf{C}_{\mathrm{d}}^{-1}\Delta\mathbf{d} + \Delta\mathbf{m}^{*}\mathbf{C}_{\mathrm{m}}^{-1}\Delta\mathbf{m} \tag{14}$$

Assume the following linear theoretical relationship exists between data and model vectors:

$$\mathbf{d}(\mathbf{m}) = \mathbf{f}_0 + \mathbf{F}\Delta\mathbf{m} \tag{15}$$

Therefore, the data error is

$$\Delta\mathbf{d} = \mathbf{d}_0 - \mathbf{d}(\mathbf{m}) = \mathbf{d}_0 - \left(\mathbf{f}_0 + \mathbf{F}\Delta\mathbf{m}\right) = \mathbf{f} - \mathbf{F}\Delta\mathbf{m} \tag{16}$$

where $\mathbf{d}_0$ is the observed data. The solution for $\Delta\mathbf{m} = \mathbf{m} - \mathbf{m}_0$, found by setting the $\Delta\mathbf{m}$ derivative of equation (14) to zero, is

$$\Delta\mathbf{m} = \left(\mathbf{F}^{*}\mathbf{C}_{\mathrm{d}}^{-1}\mathbf{F} + \mathbf{C}_{\mathrm{m}}^{-1}\right)^{-1}\mathbf{F}^{*}\mathbf{C}_{\mathrm{d}}^{-1}\mathbf{f} \tag{17}$$

Equation (17) can easily be put into the contracted form by replacing $\mathbf{f}$, $\mathbf{F}$ and $\mathbf{C}_{\mathrm{d}}^{-1}$ with $\mathbf{f}'$, $\mathbf{F}'$ and $\mathbf{C'}_{\mathrm{d}}^{-1}$. The primes indicate the contracted data space and so imply a pre-multiplication of the original matrices by the mapping matrix $\mathbf{M}$. This pre-multiplication by $\mathbf{M}$ is done efficiently using equations (12a) and (12b). The contracted form of equation (17) is therefore

$$\Delta\mathbf{m} = \left(\mathbf{F}'^{*}\mathbf{C'}_{\mathrm{d}}^{-1}\mathbf{F}' + \mathbf{C}_{\mathrm{m}}^{-1}\right)^{-1}\mathbf{F}'^{*}\mathbf{C'}_{\mathrm{d}}^{-1}\mathbf{f}' \tag{18}$$

where the mapped quantities are given by

$$\mathbf{f'} = \mathbf{Mf} \tag{19a}$$

$$\mathbf{F'} = \mathbf{MF} \tag{19b}$$

$$\mathbf{C'}_{\mathbf{d}}^{-1} = \mathbf{C}_{\mathbf{d}}^{-1/2*}\mathbf{M}^*\mathbf{M}\mathbf{C}_{\mathbf{d}}^{-1/2} \tag{20}$$

with the defining equation

$$\mathbf{C}_{\mathbf{d}}^{-1} = \mathbf{C}_{\mathbf{d}}^{-1/2*}\mathbf{C}_{\mathbf{d}}^{-1/2} \tag{21}$$

CONCLUSIONS

A method for contracting matrix dimensions while retaining dot products has been described. The method presents a more efficient method for solving overdetermined problems by transforming them into equivalent but less overdetermined problems. The transformation is achieved with a random mix and sum operation over the data dimension to a new primed data space of lower dimensionality. This transformation is summarized by the dot product

$$\mathbf{D'} = \mathbf{MD} = \begin{bmatrix} \mathbf{M}_1\mathbf{M}_2 \cdots \mathbf{M}_i \cdots \mathbf{M}_n \end{bmatrix} \begin{bmatrix} \mathbf{D}_1 \\ \mathbf{D}_2 \\ \cdot \\ \cdot \\ \mathbf{D}_i \\ \cdot \\ \cdot \\ \mathbf{D}_n \end{bmatrix} \tag{22}$$

where matrix multiplications by $\mathbf{M}_i$ are efficiently computed by a random reshuffle of row indices as described by equation (12a). A program listing which implements equation (22) is given in figure 3.

From the example it is apparent that no significant loss of accuracy results, as long as the primed data dimension is greater than or equal to about twice the model dimension. (See Mora, SEP-41, for a more realistic example.) In practical situations, where there is usually some noise in the data, the loss of accuracy introduced by the contraction mapping should be insignificant.

ACKNOWLEDGMENT

Thanks to Jeff Thorson for the many enlightening discussions.

REFERENCES

Bracewell, R., 1978, The Fourier Transform and its Applications, McGraw Hill.

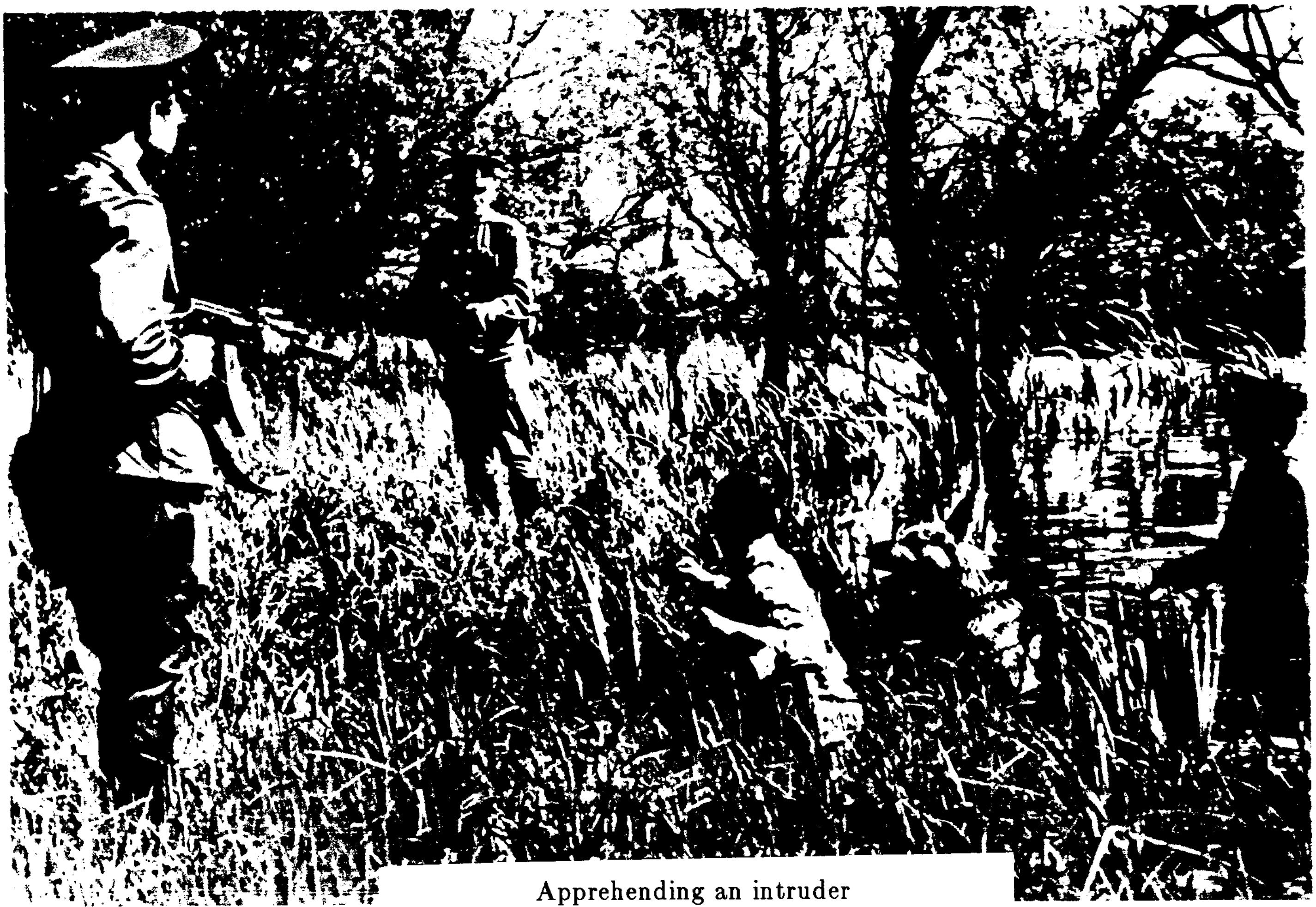
Mora, P., 1984, Elastic inversion using ray theory: SEP-41.

Tarantola, A., 1982, Generalized non linear inverse problems solved using the least-squares criterion. Reviews of Geophysics and Space Physics, v. 20, no. 2, p. 219-232.

Tarantola, A., 1983, Principles of seismic imagery, preprint.

```
# program to contract data matrix a with dimensions nd times nm
# to a contracted data matrix ai with dimensions ndi=nm*2 times nm.

define maxnd 1000                      # define max data dimension
define maxnm 50                        # define max model dimension
define maxndi maxnm*2                  # define max contracted data dimension
integer map(maxnd),time()
real a(maxnd,maxnm),ai(maxndi,maxnm)
nd=maxnd ; ndi=maxndi ; iseed=time        # initialization of data dimensions
do im=1,nm { do id=1,nd {a(id,im)=rand(iseed)}}# initialize data
do im=1,nm { do idi=1,ndi { ai(idi,im)=0. } }  # initialize contracted data
call getmap(iseed,map,nd,ndi)             # get a map
do im=1,nm {
  do id=1,nd { ai(map(id),im)=ai(map(id),im)+a(id,im) } # sum the sub-matrices
}
end

subroutine getmap(iseed,map,nd,ndi)
integer i,j,nd,ndi,map(nd),iseed
do j=1,nd { map(j)=0 }                 # initialize map
j=rand(iseed)*nd+1                     # initialize random index

#   get a random nd to nd map

do i=1,nd-1 { while(map(j)!=0) { j=rand(iseed)*nd+1 } ; map(j)=i }
do i=1,nd { if(map(i)==0) { map(i)=nd ; goto 1 } }  # last point

#   convert the nd to nd mapping to an nd to ndi mapping

1 do i=1,nd { map(i)=map(i)-(map(i)-1)/ndi*ndi }

end
```

FIG. 3. Program to contract a matrix over the data dimension



Apprehending an intruder
(from the Soviet book *Guardians of the Border*)