Journal of Computational and Applied Mathematics 33 (1990) 261–268
North-Holland

An optimal linear estimation approach to solve systems of linear algebraic equations

Ricardo L.U. DE FREITAS PINTO and Atair RIOS NETO
Instituto de Pesquisas Espaciais, Ministério de Ciências e Tecnologia, 12201 São José dos Campos, SP, Brazil

Received 31 August 1989
Revised 9 June 1990

Abstract: A method for the solution of systems of linear algebraic equations is presented. It is derived using an approach based on linear estimation theory and is directly related to a generalization of Huang's method (1975) proposed by Abaffy and Spedicato (1984). Exploring the approach adopted, the paper presents properties complementary to those found in the literature. Among these is included a Potter's factorized form of the method. In terms of applications, the method is analyzed as an alternative tool to get pseudoinverses, the solution of a class of quadratic programming problems and of ill-conditioned linear systems where iterative schemes are necessary.

Keywords: Algebraic linear systems, matrix pseudoinverses, quadratic programming, ill-conditioned linear systems.

1. Introduction

Huang [6] presented in 1975 a direct method to solve linear systems of the form

$$Hx = z, \tag{1}$$

where $x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$, $m \leqslant n$, $H' \triangleq [h_1;\ h_2; \dots;\ h_m]$ with the h_i linearly independent. This method is related to an algorithm used by Rosen [9] to find feasible points when dealing with linear equality and inequality constraints.

Abaffy and Spedicato [1] introduced in 1984 a generalization of Huang's method, using properties of deflection matrices, which can be viewed as a generalization of the concept of projection matrix.

In this paper a different approach is adopted to get results similar to those of Huang [6] and Abaffy and Spedicato [1]. Based on results of linear estimation theory, more specifically the Kalman filtering algorithm (see, for example, [7]), a method is presented which is closely related to the symmetric update version of Abbaffy and Spedicato's method.

Properties complementary to those shown in [1,6] are also presented. Among these is a property coming from results available in estimation theory and leading to an equivalent factorized Potter's form of the method [2]. This form is expected to improve the numerical stability of the method, leading to an algorithm less sensitive to the effects of round-off errors.

To make some particular features of the method explicit, it is considered as an alternative tool to solve pseudoinverses, a class of quadratic programming problems, and ill-conditioned linear systems where an iterative scheme is necessary.

2. The deterministic sequential filtering algorithm

The Kalman filtering sequential algorithm was developed to get the solution of linear estimation problems, or, according to an alternative interpretation (see, for example, [2,7]), to get the solution of the following optimization problem:

$$\text{minimize} \quad f(x) = \tfrac{1}{2}\left[(x-\bar{x})'A^{-1}(x-\bar{x}) + (z-Hx)'R^{-1}(z-Hx)\right], \tag{2}$$

where x, $\bar{x} \in \mathbb{R}^n$; $z \in \mathbb{R}^m$, $H \in \mathbb{R}^m \times \mathbb{R}^n$ as defined in (1); $R \triangleq \text{diag}[r_1, r_2, \ldots, r_m] \in \mathbb{R}^m \times \mathbb{R}^m$, positive definite; and $A \in \mathbb{R}^n \times \mathbb{R}^n$ is symmetric and positive definite. The meaning attached to $\bar{x}$, A and R in estimation theory is not of interest here.

To solve the problem of (2), the Kalman filtering sequential algorithm can be given as follows:

Step 1. Take $x_0 = \bar{x}$, $P_0 = A$.
Step 2. For $i = 1, \ldots, m$, compute

$$\text{(i)} \quad x_i = x_{i-1} + (z_i - h_i' x_{i-1}) p_i, \tag{3}$$

$$p_i = \beta_i P_{i-1} h_i, \quad \beta_i = (r_i + h_i' P_{i-1} h_i)^{-1}, \tag{4}$$

$$\text{(ii)} \quad P_i = P_{i-1} - p_i h_i' P_{i-1}. \tag{5}$$

In the limit case where $r_i = 0$, $i = 1, 2, \ldots, m$, the problem of (2) is reduced to the following constrained optimization problem:

$$\text{minimize} \quad f(x) = \tfrac{1}{2}(x-\bar{x})'A^{-1}(x-\bar{x}), \tag{6}$$

$$\text{subject to} \quad Hx = z, \tag{7}$$

which can also be solved by the algorithm of (3)–(5), with β_i particularized to:

$$\beta_i = (h_i' P_{i-1} h_i)^{-1}. \tag{8}$$

By comparison with the results of [1], it is easily seen that the Kalman filtering algorithm particularized to this limit case coincides with the symmetric update version of Abaffy and Spedicato, in the situation where row pivoting is not done and when P_0 is chosen symmetric.

For the situation where $\text{rank}(H) = m$ and β_i in (8) is well defined, then x_m as given by (3) satisfies (1). These results are demonstrated in [1], including the case when the hypothesis of symmetry of A is dropped. The proof that x_m as given by (3) is, in fact, the solution to the problem of (6) and (7) is given in Theorem 2 of the next section.

3. Complementary properties

To further explore the implications of the procedure, the following generalized projection matrix is defined:

$$P_i^A \triangleq A - AH_i'(H_i A H_i')^{-1} H_i A, \tag{9}$$

where, for any $i \leqslant m$,

$$H_i' \triangleq [h_1;\ h_2; \ldots;\ h_i]. \tag{10}$$

The matrix A can be any $n \times n$ matrix such that the inverse in (9) exists, but here it is taken as real symmetric and positive definite. By application of this definition, it is easily verified that:

$$P_i^A H_i' = 0 \quad (n \times i, \text{ zero matrix}), \tag{11}$$

$$H_i P_i^A = 0 \quad (i \times n, \text{ zero matrix}), \tag{12}$$

and, as a consequence, for any $v \in \mathbb{R}^n$

$$H_i(P_i^A v) = 0, \tag{13}$$

and thus P_i^A projects v in the subspace orthogonal to that generated by the row vectors of H_i. Considering the previous results, it results that:

(i) $$P_i^A A^{-1} P_j = P_i^A, \quad \text{for } j \leqslant i, \tag{14}$$

where P_j is as defined in (5) in the limit case when $r_j = 0$, $1 \leqslant j \leqslant m$, and for any $i \leqslant m$ the property follows by induction in j, applying (11);

(ii) $$P_i^A = P_i, \quad \text{for } i = 1, \ldots, m, \tag{15}$$

which again can be proved by induction, noticing it is true for $i = 1$ (see (5)) and that by assuming it to be true for $1 < i < m$, then from (14):

$$P_{i+1}^A = P_{i+1}^A A^{-1} P_{i+1} = \left(A - AH_{i+1}'(H_{i+1} A H_{i+1}')^{-1} H_{i+1} A\right) A^{-1} P_{i+1} = P_{i+1}$$

as a consequence of

$$H_{i+1} P_{i+1} = H_{i+1}\left(P_i^A - P_i^A h_{i+1}\left(h_{i+1}' P_i^A h_{i+1}\right)^{-1} h_{i+1}' P_i^A\right) = 0,$$

by applying (12) and considering that $H_{i+1}' \triangleq [H_i' : h_{i+1}]$.

A batch version of the algorithm of (3)–(5) can now be obtained.

Theorem 1. *The solution x_m of system* (1), *as given by the algorithm of Section* 2, *can be equivalently obtained as*:

$$x_m = x_0 + AH'(H'AH)^{-1}(z - Hx_0). \tag{16}$$

Proof. To prove (16) it is necessary to notice that for $0 \leqslant i \leqslant m-1$

$$\left(I + AH'(HAH')^{-1} H\right) x_{i+1} = \left(I + AH'(HAH')^{-1} H\right) x_i, \tag{17}$$

a result that can be proved by taking x_{i+1} from (3) and (4) in the limit case, with z_i given by (7) and using properties (14) and (12). Equation (17) by induction leads to

$$\left(I + AH'(HAH')^{-1} H\right) x_m = \left(I + AH'(HAH')^{-1} H\right) x_0 \tag{18}$$

and, subsequently, to

$$x_m + AH'(HAH')^{-1}(z - Hx_m) = x_0 + AH'(HAH')^{-1}(z - Hx_0). \tag{19}$$

But, since

$$z - Hx_m = 0 \quad (\text{as proved in } [1]),$$

(19) is equivalent to (16), the result sought. □

Another way of looking at the result x_m given by the sequential algorithm of Section 2 is as the solution of an optimization problem, as in the following theorem.

Theorem 2. *If $A = A'$, positive definite, then x_m is also a solution of the optimization problem:* [1]

$$\text{minimize} \quad J = \tfrac{1}{2}(x - x_0)' A^{-1}(x - x_0), \tag{20}$$

$$\text{subject to} \quad Hx = z. \tag{21}$$

Proof. For x_m to be a solution of the optimization problem formulated, it is necessary and sufficient that (see, e.g., [8]):

$$z - Hx_m = 0, \tag{22}$$

$$P\nabla J(x_m) = 0, \tag{23}$$

where P and $\nabla J(x_m)$ are, respectively, the projection matrix associated to H and the gradient of J evaluated in x_m:

$$P \triangleq I - H'(HH')^{-1}H, \qquad \nabla J(x_m) \triangleq \frac{\partial J}{\partial x}(x_m).$$

Condition (22) is certainly satisfied, since x_m solves (1). By considering (16) and since

$$P\nabla J(x_m) = PA^{-1}(x_m - x_0),$$

there results:

$$P\nabla J(x_m) = PA^{-1}\big(AH'(HAH')^{-1}(z - Hx_0)\big) = 0,$$

since $PH' = 0$, and thus (23) is verified. □

The approach of using results of linear estimation theory to get the method of Section 2 can be further explored, adopting a factorized version of the algorithm [2]. The purpose is to have the algorithm in a form that, though being mathematically equivalent, it has a better numerical performance.

For the type of problem at hand, the Potter's square root factorization (see, for example, [2]) seems adequate, and leads to the following alternative algorithm:

Step 1. Take $x_0 = \bar{x}$, $S_0 S_0' = P_0 = A$.
Step 2. For $i = 1, 2, \ldots, m$, compute

$$\text{(i)} \quad v_i' = h_i' S_{i-1}, \qquad \beta_i = (v_i' v_i)^{-1}, \tag{24}$$

$$p_i = \beta_i S_{i-1} v_i, \tag{25}$$

$$x_i = x_{i-1} + (z_i - h_i' x_{i-1}) p_i, \tag{26}$$

$$\text{(ii)} \quad S_i = S_{i-1} - p_i v_i'. \tag{27}$$

Note that, once given $P_{i-1} = S_{i-1} S_{i-1}'$, there results from (27)

$$S_i S_i' = P_{i-1} - p_i h_i' P_{i-1} = P_i$$

as in (5), and the proof of convergence follows straightforward.

[1] In fact, it can be proved that the algorithm of (3)–(5) holds also for A definite negative. If A is definite negative, Theorem 2 is still valid, but the problem of (20) is changed to one of maximization.

4. Applications and numerical examples

4.1. Solution of simultaneous equations and determination of pseudoinverses

Given a system of simultaneous equations

$$HX = Z, \tag{28}$$

with $X \triangleq (x^{(1)}; x^{(2)}; \ldots; x^{(q)}) \in \mathbb{R}^n \times \mathbb{R}^q$ and $Z \triangleq (z^{(1)}; z^{(2)}; \ldots; z^{(q)}) \in \mathbb{R}^m \times \mathbb{R}^q$, the algorithm of Section 2 can be applied to successively (or simultaneously) solve

$$Hx^{(j)} = z^{(j)}, \quad j = 1, 2, \ldots, q,$$

without the need of recalculating the vector p_i's for each $j \geqslant 2$. Notice that when $q = m$ and $Z = I$ ($m \times m$ identity matrix), then the problem is one of pseudoinverse determination.

4.2. Redundant equations

Based upon the result of Theorem 1, the application of the algorithm of this paper can be extended to problems with redundant equations (see [9]). This can be done by including in the algorithm the following test.

Test: If $h_i' P_{i-1} h_i \neq 0$ is not verified [2], then:

(i) if $z_i - h_i' x_{i-1} = 0$, then the equation correspondent to h_i is redundant and must be ignored,

(ii) if $z_i - h_i' x_{i-1} \neq 0$, then the problem has no solution. The equations are not consistent.

4.3. Quadratic programming

Consider the problem of

$$\text{minimize (maximize)} \quad J = \tfrac{1}{2} x' M x - b' x, \quad M = M', \tag{29}$$

$$\text{subject to} \quad Hx = z, \tag{30}$$

where the m rows of H are linearly independent and M is positive definite (negative definite). The conditions for solving this problem are similar to (22), (23), where now

$$\nabla J(x^*) = Mx^* - b. \tag{31}$$

In compact form it can be written as

$$H^* x = z^*, \tag{32}$$

where

$$H^* \triangleq \begin{bmatrix} H \\ \cdots \\ PM \end{bmatrix} \quad \text{and} \quad z^* \triangleq \begin{bmatrix} z \\ \cdots \\ Pb \end{bmatrix}. \tag{33}$$

Relation (32) represents a system of equations with redundancy and can be solved in the following manner:

(i) determine one solution for (30) with $A = I$ ($n \times n$ identity matrix);

[2] For considering round-off errors in the test, see [6].

(ii) find the remaining equations of (32) by calculating PM and Pb, where P is as defined in Theorem 2 and $P = P_m$ since $A = I$ (see (15));

(iii) continue the solution of (32) using the algorithm of Section 2, including the test described in Section 4.2 for redundancy identification.

Notice that, because of the sequential nature of the algorithm, the remaining equations of (32) are obtained after the part corresponding to (30) has been solved.

4.4. Iterative solution of linear systems

Apart from its stability properties (see [3,6]) the deterministic sequential filtering algorithm can be coupled with an iterative scheme to improve the result obtained in ill-conditioned problems. Two ways of doing this are described.

Residue treatment

If x_a is an approximation to an exact solution x of (1) and if one takes

$$r_a \triangleq z - Hx_a, \tag{34}$$

the residue associated to x_a, and

$$\Delta x \triangleq x - x_a, \tag{35}$$

then

$$H\Delta x = r_a. \tag{36}$$

The residue treatment technique consists (see, e.g. [4]) in solving (36) and taking

$$x = x_a + \Delta x \tag{37}$$

as an improved value for the solution and in repeating the process as necessary with x_a substituted for the last value obtained in (37).

Notice that in the reiterations it is not necessary to recalculate the vector p_i and, thus, the number of calculations per reiteration is low.

4.5. Numerical example

An example is taken to illustrate the influence of round-off errors. (See [5, p.141].) Let

$$H = \begin{bmatrix} 6 & 13 & -17 \\ 13 & 29 & -38 \\ -17 & -38 & 50 \end{bmatrix}, \qquad z = \begin{bmatrix} 1 \\ 2 \\ -3 \end{bmatrix},$$

where H is ill-conditioned (condition number = 1441) and the exact solution is:

$$x' = (1, -3, -2).$$

This problem was solved in a Burroughs 6800, initializing the procedure with

$$P_0 = I(3 \times 3), \qquad x_0' = (0, 0, 0).$$

Basic algorithm

Using the algorithm of (3)–(5), one gets, after initial iteration,

$$x_a^{(1)} = \begin{bmatrix} 0.99999999933 \\ -3.00000058924 \\ -2.00000044805 \end{bmatrix}, \qquad r_a^{(1)} = \begin{bmatrix} 4.72646206616 \cdot 10^{-8} \\ 6.98491930960 \cdot 10^{-8} \\ 9.31322574616 \cdot 10^{-10} \end{bmatrix},$$

where the upper index is to indicate the number of iterations.

Reiterating twice with the residue treatment, there results:

$$x_a^{(3)} = \begin{bmatrix} 0.99999999980 \\ -2.99999999971 \\ -1.99999999985 \end{bmatrix}, \qquad r_a^{(3)} = \begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \end{bmatrix}.$$

Here the zeros are to be understood as the computer numerical zeros.

Whenever it is possible to get the residue (r_a) with a better accuracy, a significant improvement occurs in the solution by residue treatment. As an illustration for the previous example, taking

$$r_a^{(1)} = \begin{bmatrix} 4.72100509796 \cdot 10^{-8} \\ 7.05895217834 \cdot 10^{-8} \\ 2.27373675443 \cdot 10^{-10} \end{bmatrix},$$

calculated with double precision operations, there results in just one reiteration

$$x_a^{(2)} = \begin{bmatrix} 1.00000000000 \\ -3.00000000000 \\ -2.00000000000 \end{bmatrix},$$

which is the correct value under the number of considered digits (twelve).

Notice that only the residue $r_a^{(1)}$ has to be calculated in double precision.

Factorized Potter's algorithm

Using the algorithm of (24)–(27) one gets, after one iteration,

$$x_a^{(2)} = \begin{bmatrix} 1.00000000083 \\ -3.00000000432 \\ -2.00000000300 \end{bmatrix}, \qquad r_a^{(1)} = \begin{bmatrix} 2.32830643654 \cdot 10^{-10} \\ 9.31322574616 \cdot 10^{-10} \\ 0.0 \end{bmatrix}.$$

It is seen that these results have a better accuracy than those obtained with the basic algorithm. Reiterating once with the residue treatment, there results:

$$x_a^{(2)} = \begin{bmatrix} 0.99999999850 \\ -2.99999999501 \\ -1.99999999671 \end{bmatrix}, \qquad r_a^{(2)} = \begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \end{bmatrix}.$$

Where the zeros must be understood again as the computer numerical zeros.

If the residue $r_a^{(1)}$ is calculated with double precision, the exact value is obtained in just one reiteration.

5. Conclusions

A deterministic sequential filtering algorithm for the solution of systems of linear algebraic equations was presented. The method was obtained using results available in linear estimation theory and was seen to coincide with the symmetric update version of Abaffy and Spedicato.

The method seems to be a good alternative in the solution of pseudoinversion and quadratic programming problems. Associated to an iterative scheme, it provides a tool for the solution of ill-conditioned problems.

The numerical performance of the factorized Potter's version of the method indicates that this form of expressing the numerical algorithm has a potential worth of being further explored. It seems to be a recommended form to be used to attenuate numerical accuracy deterioration due to computer round-off.

References

[1] J. Abaffy and E. Spedicato, A generalization of Huang's method for solving systems of linear algebraic equations, *Boll. Un. Mat. Ital.* **6** (1984) 517–519.

[2] G.J. Bierman, *Factorization Methods for Discrete Sequential Estimation* (Academic Press, New York, 1977).

[3] C.G. Broyden, On the numerical stability of Huang's and related methods, *J. Optim. Theory Appl.* **47** (4) (1985).

[4] L. Fox, *An Introduction to Numerical Linear Algebra* (Oxford Univ. Press, New York, 1965).

[5] M. Hestenes, *Conjugate Direction Methods in Optimization* (Springer, New York, 1980).

[6] H.Y. Huang, A direct method for the general solution of a system of equations, *J. Optim. Theory Appl.* **16** (5/6) (1975).

[7] A.H. Jazwinski, *Stochastic Processes and Filtering Theory* (Academic Press, New York, 1970).

[8] D.G. Luenberger, *Introduction to Linear and Nonlinear Programming* (Addison-Wesley, Reading, MA, 1973).

[9] J. Rosen, The gradient projection method for nonlinear programming I, linear constraints, *SIAM J. Appl. Math.* **8** (1960) 181–217.