An Efficient Basis Update for Asymptotic Linear Programming*

Bernard F. Lamond
Faculté des Sciences de l'Administration
Université Laval
Québec, Canada G1K 7P4

Submitted by Uriel G. Rothblum

ABSTRACT

For a linear program in which the constraint coefficients vary linearly with the time parameter, we showed in a previous paper that a basic feasible solution can be evaluated using $O((k+1)m^3)$ arithmetic operations, where m is the number of constraints and k is the index of the basis matrix pair. Here we show, in the special case when $k = 1$ for all basis matrix pairs, and when one of the matrices in each pair has nearly full rank, how the (possibly singular) matrix factorization can be updated with only $O(m^2)$ operations, using rank-one update techniques. This makes the arithmetic complexity of updating the basis in asymptotic linear programming comparable to that of updating the inverse in ordinary linear programming, in this case. Moreover, we show that the result holds, in particular, when computing a Blackwell optimal policy for Markov decision chains in the unichain case or when all policies have only a small number of recurrent subchains.

1. INTRODUCTION

It was shown in [12] that the *asymptotic linear programming* problem could be solved by the simplex method, by storing all tableau entries as *rational functions* of time and performing all arithmetic operations over the

*This research was supported in part by the National Science Foundation, under grant ECS-8803061, while the author was at the University of Arizona, and by the Natural Science and Engineering Research Council of Canada, under grant OPG-0105560.

655 Avenue of the Americas, New York, NY 10010 0024-3795/93/$6.00

field of rational functions (using polynomial arithmetic and comparing the high-degree coefficients of the polynomials). When the constraint coefficients vary linearly with the time parameter, this method requires $O(m^4 \log m)$ *real* arithmetic operations (on a digital computer these are called floating-point operations, or *flops* for short) in order to invert a typical basis matrix, where m is the number of constraints.

This complexity bound was lowered in [14], where a Neumann expansion of the basis matrix inverse was used, based on a matrix factorization of [19], giving $O((k + 1)m^3)$ flops, where k is the index of the basis matrix pair. It is always true that $0 \leqslant k < m$, and there are classes of problems for which k is a fixed number independent of m. A similar series expansion was obtained independently in Anstreicher's unpublished dissertation [2], based on the *shuffle* algorithm of [16], but in a different context. More recently, yet another factorization was presented in [11] which obtains a similar series expansion in $O(m^3)$ flops, independent of the index k.

In this paper, we consider the special case when all the basis matrix pairs have the same index $k = 1$ (and hence are singular), and we assume that their corresponding rank is close to m. For example, the asymptotic linear program used in [6] and [10] for finding a Blackwell optimal policy [3] in Markov decision chains satisfies these conditions in the unichain case, or when the number of recurrent subchains is bounded by a fixed constant for all stationary policies. Our main result is that the factorization of [14], which takes $O(m^3)$ in this case, can be updated in $O(m^2)$ flops, compared to $O(m^3 \log m)$ in [10] and [12]. The update algorithm we present uses the well-known Sherman-Morrison formula for rank-one modification of nonsingular matrices, and its generalization to higher rank, the Woodbury formula (see, e.g., [9]). Because the modified matrices can be singular in our problem, our algorithm must first find a suitable (i.e., nonsingular) partition. We show that such a partition can always be found.

The main purpose of this research is to investigate whether the computational complexity of an iteration of asymptotic linear programming is comparable to that of an iteration of ordinary linear programming, especially for computing a Blackwell optimal policy in Markov decision chains. For this reason, this paper contains no results about the numerical stability of the update algorithm. Moreover, the factorization (and its update) is presented completely in terms of explicit matrix inverses. This allows for a simpler presentation than otherwise. We cannot claim that this update algorithm has optimal complexity; however, we observe that it has a lower complexity bound than other algorithms available in the literature.

The paper is organized as follows. The asymptotic linear programming problem is briefly defined in Section 2. The factorization of the basis and the corresponding series expansions are described in Section 3. The asymptotic linear-programming formulation for Markov decision chains is presented in Section 4, as a family of problems with index 1 and nearly full rank, thus justifying our restrictions. The algorithm for updating the basis is derived in Section 5 in the special case when the submatrix is nonsingular. The method is extended in Section 6 to handle singular submatrices. The main results of this paper are Lemmas 5.1 and 5.2. and Theorem 6.1.

2. DEFINITION OF ASYMPTOTIC LINEAR PROGRAMMING

As defined in [12], the asymptotic linear programming problem is formulated as follows:

$$\text{maximize } c(t)x, \qquad \text{subject to } A(t)x = b(t) \text{ and } x \geqslant 0, \tag{1}$$

where the entries of $c(t)$, $A(t)$, and $b(t)$ are rational functions of the time parameter t. It is required to find a basis that is feasible and optimal for *all* sufficiently large values of t. If we wanted only to find an optimal solution for a fixed value of t, this would be an ordinary linear program. Here instead, the parameter t is an indeterminate which represents an unknown, arbitrarily large real number. We will consider only the special case in which the constraint coefficients vary linearly with time, i.e., $A(t) = G + tH$ for some fixed matrices G and H. For simplicity, we will also assume that the vectors $c(t) = c$ and $b(t) = b$ are independent of time.

It was shown in [12] that for an arbitrary basic feasible solution, if $B(t)$ is the matrix of basic columns of $A(t)$, then the entries of $B(t)^{-1}$ are rational functions of the form $p(t)/q(t)$ where $p(t)$ and $q(t)$ are polynomials of degree at most m, the number of constraints. From this, selecting a variable to enter the basis and selecting a variable to leave the basis can be performed as in the usual simplex method, except that polynomial arithmetic must be used and all comparisons are based on the high-degree coefficients of the polynomials. That is, if

$$p(t) = p_0 + p_1 t + \cdots + p_k t^k,$$

and

$$q(t) = q_0 + q_1 t + \cdots + q_l t^l,$$

with $p_k \neq 0$ and $q_l \neq 0$, then we say that $p(t)/q(t) > 0$ if and only if $p_k q_l > 0$.

3. FACTORIZATION OF THE BASIS

Let $B(t)$ be the (square) matrix of basic columns of $A(t)$ corresponding to an arbitrary basic feasible solution of (1). We write $B(t) = A + tB$ (with A and B having the basic columns of G and H, respectively), and we assume that there is at least one point $t = d$ (on the real line) for which the real matrix $B(d)$ is nonsingular, although both A and B may be singular. This implies that there is a time $T > 0$ such that $B(t)$ is nonsingular for all $t \geqslant T$. In this sense, the matrix $B(t)$, where the parameter t is indeterminate, is defined to be nonsingular. Such a matrix is also known as a *regular pencil* in the literature (see, e.g., [7]).

Let C be an arbitrary square matrix with real (or complex) entries. The *index* of C, denoted $\operatorname{ind} C$, is the smallest nonnegative integer k such that $\operatorname{rank} C^{k+1} = \operatorname{rank} C^k$ (see, e.g., [4]). Following the approach of [5], we define the *index of a matrix pair*, denoted $\operatorname{ind}(A, B)$, as follows. Let d be any value of the time parameter for which the matrix $A + dB$ is nonsingular. Then $\operatorname{ind}(A, B) = \operatorname{ind}[(A + dB)^{-1}B]$. The index is uniquely determined by the matrices A and B and is independent of d.

We are interested in the special case when $\operatorname{ind}(A, B) = 1$. Then $\operatorname{rank} B = r < m$ and, according to [14], there are two nonsingular $m \times m$ matrices U and V such that

$$UAV = \begin{pmatrix} A_{11} & 0 \\ 0 & A_{22} \end{pmatrix} \quad \text{and} \quad UBV = \begin{pmatrix} B_{11} & 0 \\ 0 & 0 \end{pmatrix}, \tag{2}$$

where A_{11} and B_{11} are $r \times r$ matrices, A_{22} is an $(m - r) \times (m - r)$ matrix, and A_{22} and B_{11} are nonsingular. Consequently, for any fixed, large enough t we have

$$\begin{aligned} V^{-1}(A + tB)^{-1}U^{-1} = \begin{pmatrix} 0 & 0 \\ 0 & A_{22}^{-1} \end{pmatrix} + t^{-1}\begin{pmatrix} B_{11}^{-1} & 0 \\ 0 & 0 \end{pmatrix} \\ - t^{-2}\begin{pmatrix} D_{11}B_{11}^{-1} & 0 \\ 0 & 0 \end{pmatrix} + t^{-3}\begin{pmatrix} D_{11}^2 B_{11}^{-1} & 0 \\ 0 & 0 \end{pmatrix} - \cdots, \end{aligned} \tag{3}$$

where $D_{11} = B_{11}^{-1}A_{11}$.

Equation (3) can be used to express the reduced prices as Laurent series in powers of $1/t$. Let c_B be the vector of objective function coefficients for the basic variables, and define

$$w^{(0)} = c_B V \begin{pmatrix} 0 & 0 \\ 0 & A_{22}^{-1} \end{pmatrix}, \qquad w^{(1)} = c_B V \begin{pmatrix} B_{11}^{-1} & 0 \\ 0 & 0 \end{pmatrix}, \tag{4}$$

and, for $i = 2, 3, \dots,$

$$w^{(i)} = -w^{(i-1)} \begin{pmatrix} A_{11} B_{11}^{-1} & 0 \\ 0 & 0 \end{pmatrix}. \tag{5}$$

Suppose $G_j + tH_j$ is the column vector of constraint coefficients corresponding to the nonbasic variable x_j. Then the reduced price $\bar{c}_j$ is given by the series

$$\bar{c}_j = -t\left[w^{(0)}h_j\right] + \left[c_j - w^{(0)}g_j - w^{(1)}h_j\right] - \sum_{i=1}^{\infty} t^{-i}\left[w^{(i)}g_j + w^{(i+1)}h_j\right], \tag{6}$$

where $g_j = UG_j$ and $h_j = UH_j$.

Using Equations (4) and (5), each term of the series (6) can be computed with $O(m^2)$ flops. To determine whether the nonbasic variable x_j is eligible to enter the basis, we need to test whether the first nonzero term in (6) is positive. In a way similar to Theorem 5.1 of [14], it is easy to show that if the first $m + 2$ terms of (6) are equal to zero, then $\bar{c}_j = 0$. Hence in the worst case, it will take $O(m^3)$ flops to test whether a nonbasic variable is eligible to enter the basis. By comparison, this takes $O(m^3 \log m)$ flops when rational functions are used as in [10] and [12]. However, in the case when only a fixed, small number of terms need to be evaluated, testing an entering variable takes only $O(m^2)$ flops, which is comparable to ordinary linear programming. This is the case for Markov decision chains with the sensitive discount optimality criteria of [18].

A similar approach can be used to show that choosing a variable to leave the basis can also be performed with $O(m^3)$ flops in the worst case, and with $O(m^2)$ flops when only a fixed number of terms are required. Moreover, in the special case of Markov decision chains, the variable that leaves the basis can be selected without any computations at all, once the entering variable is chosen. This is because, as noted in the next section, there is exactly one basic variable for each state.

A typical iteration of the simplex method requires one more step: updating the factorization of the basis matrix. This takes $O(m^2)$ flops in ordinary linear programming and $O(m^3 \log m)$ flops in asymptotic linear programming using rational functions. We will show, in Section 5, how the factorization of Equation (2) can be updated in $O(m^2)$ flops when $m - r \leqslant K$, where K is a fixed constant such that $K \ll m$. A family of such problems is discussed next.

4. SPECIAL CASE: MARKOV DECISION CHAINS

In a *Markov decision chain*, the state i of a system is observed periodically and in each period an action a is selected. There is a finite set E of states, and for every state $i \in E$ there is a finite set $A(i)$ of possible actions. When action a is taken in state i, a reward of expected value r_{ia} is received instantly. The conditional probability p_{iaj} that state j is observed, given that the system was in state i at the previous period and that action a was selected, is given and invariant with time.

Future rewards are discounted with a one-period discount factor $\beta = 1/(1+\rho)$, where $\rho > 0$ is a (small) interest rate. The objective is to find a decision rule γ, such that action $a = \gamma(i)$ is selected whenever the system is in state i, in order to maximize the present value of all rewards received over an infinite planning horizon. A decision rule (or policy) that is optimal for *all* sufficiently small values of the interest rate ρ is said to be *Blackwell optimal* following [3].

The problem is formulated as an asymptotic linear program in Equation (3.2.2) of [10], as follows:

$$
\begin{aligned}
\max \quad & \sum_{i \in E} \sum_{a \in A(i)} (1+\rho) r_{ia} x_{ia} \\
\text{s.t.} \quad & \sum_{i \in E} \sum_{a \in A(i)} \left[(1+\rho)\delta_{ij} - p_{iaj}\right] x_{ia} = 1, \quad && j \in E \\
& \text{and} \quad x_{ia} \geqslant 0, && a \in A(i),\ i \in E,
\end{aligned}
$$

where δ_{ij}, as usual, denotes Kronecker's indicator function. Reparametrizing with $t = 1/\rho$, and rescaling the objective coefficients and the right-hand sides, we get an equivalent formulation with t large:

$$\begin{aligned} \max \quad & \sum_{i \in E} \sum_{a \in A(i)} r_{ia} x_{ia} \\ \text{s.t.} \quad & \sum_{i \in E} \sum_{a \in A(i)} \left[\delta_{ij} + t(\delta_{ij} - p_{iaj}) \right] x_{ia} = 1, \quad j \in E \\ & \text{and} \quad x_{ia} \geqslant 0, \qquad a \in A(i),\ i \in E, \end{aligned} \tag{7}$$

Each basic feasible solution of (7) has $x_{ia} > 0$ for exactly one $a \in A(i)$, for every $i \in E$. That is, each basic feasible solution determines precisely one decision rule γ.

Let x_B^γ be the vector of basic variables and P^γ the matrix of transition probabilities under policy γ. Then the basic values are obtained by solving the system of linear equations

$$B(t) x_B^\gamma = e, \tag{8}$$

where e is a column vector with all entries equal to 1 and

$$B(t)' = I + t(I - P^\gamma),$$

with I the identity matrix and $B(t)'$ the transpose of $B(t)$.

It is well known (see, e.g, [4]) that $\operatorname{ind}(I - P^\gamma) = 1$, and hence that $\operatorname{ind}(I, I - P^\gamma) = 1$ for all policies and hence for all feasible bases of (7). Further, under the frequently used unichain assumption that the Markov chain associated with each policy has exactly one recurrent subchain, we also have that $r = \operatorname{rank}(I - P^\gamma) = m - 1$, so that $m - r = 1 \ll m$ for all feasible bases. See [13] and [15] for the factorization and series expansion of (8) and related equations.

Using the algorithm of [10], it would take $O(m^2)$ *rational* operations to update the matrix $B(t)$ (whose elements are rational functions) at each pivot step. As implemented in [10], each rational operation can be performed in $O(m^2)$ *real* arithmetic operations, giving $O(m^4)$ flops for the complexity of a single pivot step. Using the more sophisticated algorithms of [1] to perform the rational arithmetic in $O(m \log m)$ real operations, the update still requires $O(m^3 \log m)$ flops. The factorization of the basis from scratch, as discussed in [14], would take $O(m^3)$ flops. The algorithm we present in the next sections reduces this complexity bound to $O(m^2)$ flops. Our method is not restricted to the unichain case. In fact, combining the operations counts for the two major steps described in Section 5, we obtain $[15 + 2(m - r)]m^2 + 4(m - r)m$ flops. This is $O(m^2)$, in particular, if $m - r \leqslant K$ for some constant K. For example, a Markov decision chain for which all policies have no more than, say $K = 10$ recurrent classes would satisfy our assumptions.

A different approach for computing a Blackwell optimal policy is the policy iteration method of [17]. Unlike the simplex method, each policy iteration step changes the action of several states, causing the matrix $B(t)$ to be modified in more than one column. The policy evaluation step then requires $O(m^3)$ flops to factorize the corresponding matrix $I - P^{\gamma}$. The correspondence between the simplex method and the policy iteration method is well known in the case of a *fixed* interest rate ρ. The matrix factorizations presented in [14] and [15] and the efficient basis update presented in this paper illustrate a similar connection between asymptotic linear programming and the policy iteration methods of [17] for the Blackwell optimality criterion and of [18] for the sensitive discount optimality criteria.

5. RANK-ONE UPDATE OF THE FACTORIZATION

In this section, we describe an algorithm for updating the factorization of Equation (2) in $O(m^2)$ flops. We assume that each of the matrices A and B is changed by a rank-one modification. The basis change of linear programming is then a special case in which precisely one column is modified. As in the previous sections, we assume that A and B are $m \times m$ matrices such that $\text{ind}(A, B) = 1$. Moreover, we assume also that $m - r \ll m$, where $r = \text{rank}\, B$.

Let α and β be column vectors and w a row vector. The modified matrices are then

$$\hat{A} = A + \alpha w \quad \text{and} \quad \hat{B} = B + \beta w. \tag{9}$$

In the context of asymptotic linear programming, w would be the unit vector pointing to the variable that leaves the basis. We assume that $\text{ind}(\hat{A}, \hat{B}) = 1$, so that a factorization of the same form as Equation (2) is valid for $\hat{A}$ and $\hat{B}$. This is consistent with our earlier assumption that all basis matrix pairs (encountered by the simplex method) have index 1 in the asymptotic linear program. Let $\hat{r} = \text{rank}\, \hat{B}$. Then three possible cases are readily identified: $\hat{r} = r - 1$, $\hat{r} = r$, and $\hat{r} = r + 1$, and hence $m - \hat{r} \ll m$ holds.

As we are concerned only with *updating* the factorization of Equation (2), we assume that the following matrices are available as *input* to our algorithm: the nonsingular $m \times m$ matrices U and V, the (possibly singular) $r \times r$ matrix A_{11}, the nonsingular $(m - r) \times (m - r)$ matrix A_{22}, and the $r \times r$ matrix B_{11}^{-1}. We are assuming that $m - r \leqslant K$, where K is a small enough constant so that the computational effort required to invert A_{22} is no greater than $O(m^2)$, i.e., $(m - r)^3 \leqslant K^3 \leqslant m^2$. The *output* of our algorithm then consists of the $m \times m$ matrices $\hat{U}$ and $\hat{V}$, the $\hat{r} \times \hat{r}$ matrices $\hat{A}_{11}$ and $\hat{B}_{11}^{-1}$, and the $(m - \hat{r}) \times (m - \hat{r})$ matrix $\hat{A}_{22}$, all of them nonsingular except possibly $\hat{A}_{11}$, such that

$$\hat{U}\hat{A}\hat{V} = \begin{pmatrix} \hat{A}_{11} & 0 \\ 0 & \hat{A}_{22} \end{pmatrix} \quad \text{and} \quad \hat{U}\hat{B}\hat{V} = \begin{pmatrix} \hat{B}_{11} & 0 \\ 0 & 0 \end{pmatrix}. \tag{10}$$

The update algorithm consists of two major steps, step I and step II. In the first, the matrix $\hat{B}_{11}^{-1}$ is computed. Then the factorization of $\hat{A}$ is obtained in step II. We first describe step I, which consists of four substeps, labeled step 1 to step 4.

Step 1 is simply the conversion of the vectors β and w. Using the same partitions as in Equation (2), let us write

$$U\beta = \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} \quad \text{and} \quad wV = (w_1 \quad w_2). \tag{11}$$

Then we apply the same transformation to $\hat{B}$, getting

$$U\hat{B}V = \begin{pmatrix} B_{11} + \beta_1 w_1 & \beta_1 w_2 \\ \beta_2 w_1 & \beta_2 w_2 \end{pmatrix}. \tag{12}$$

Although Equation (12) holds, the matrix $U\hat{B}V$ is not computed at this point. Hence step 1 takes $2m^2$ flops to transform the vectors β and w. (Here, the term *flop* is used as in [8] to denote a pair of arithmetic operations: one addition and one multiplication.)

At step 2, we eliminate the bottom and rightmost parts of the matrix $U\hat{B}V$, working directly on the subvectors β_2 and w_2. Let u denote a unit vector of appropriate dimension, so that $u' = (1, 0, \dots, 0)$. Denote by β_{2i} (w_{2i}) the ith component of the subvector β_2 (w_2). Let $k = \arg\max\{|\beta_{2i}| : i = 1, \dots, m - r\}$ and $l = \arg\max\{|w_{2i}| : i = 1, \dots, m - r\}$, and define $\beta^* = \beta_{2k}$ and $w^* = w_{2l}$. Then there are two nonsingular $(m - r) \times (m -$

r) matrices $X^{(1)}$ and $Y^{(1)}$ such that $X^{(1)}\beta_2 = \beta^* u$ and $w_2 Y^{(1)} = u' w^*$. For example, with $m - r = 4$ and $k = 1$ a typical form is given by

$$X^{(1)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\beta_{22}/\beta_{21} & 1 & 0 & 0 \\ -\beta_{23}/\beta_{21} & 0 & 1 & 0 \\ -\beta_{24}/\beta_{21} & 0 & 0 & 1 \end{pmatrix},$$

with an obvious permutation when $k \neq 1$. The matrix $Y^{(1)}$ could have the same structure, transposed. Define

$$U^{(1)} = \begin{pmatrix} I & 0 \\ 0 & X^{(1)} \end{pmatrix} \quad \text{and} \quad V^{(1)} = \begin{pmatrix} I & 0 \\ 0 & Y^{(1)} \end{pmatrix}. \tag{13}$$

We now have that

$$U^{(1)} U \hat{B} V V^{(1)} = \begin{pmatrix} B_{11} + \beta_1 w_1 & \beta_1 w^* & 0 \\ \beta^* w_1 & \beta^* w^* & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{14}$$

where the matrices of (13) and (14) are not computed explicitly. The matrices $U^{(1)}U$ and $VV^{(1)}$ can each be computed with $2m(m - r)$ flops.

Step 3 is the heart of the algorithm with the inversion of $\overline{B}_{11} = B_{11} + \beta_1 w_1$, which, for now, is assumed to be nonsingular. To simplify the presentation, the singular case will be postponed until Section 6. Let $\gamma = w_1 B_{11}^{-1} \beta_1$. Then $\gamma \neq -1$ and the Sherman-Morrison formula (see, e.g., [9]) gives

$$\overline{B}_{11}^{-1} = (B_{11} + \beta_1 w_1)^{-1} = B_{11}^{-1} - \frac{(B_{11}^{-1}\beta_1)(w_1 B_{11}^{-1})}{1 + \gamma}. \tag{15}$$

This takes $3r^2$ flops.

At step 4 we compute $\hat{B}_{11}^{-1}$. Using the same partitioning as in Equation (14), define

$$U^{(2)} = \begin{pmatrix} I & 0 & 0 \\ -\beta^* w_1 \overline{B}_{11}^{-1} & 1 & 0 \\ 0 & 0 & I \end{pmatrix} \quad \text{and} \quad V^{(2)} = \begin{pmatrix} I & -\overline{B}_{11}^{-1}\beta_1 w^* & 0 \\ 0 & 1 & 0 \\ 0 & 0 & I \end{pmatrix}. \tag{16}$$

Step 1. Transform the vectors β and w of Equation (9), and partition them as in Equation (11).
Step 2. Compute the matrices $U^{(1)}U$ and $VV^{(1)}$ with $U^{(1)}$ and $V^{(1)}$ as in Equation (13).
Step 3. Let $\gamma = w_1 B_{11}^{-1}\beta_1$. If $\gamma = -1$ then see Section 6, else compute $(B_{11} + \beta_1 w_1)^{-1}$ as in Equation (15).
Step 4. Compute the matrices $U^{(2)}U^{(1)}U$ and $VV^{(1)}V^{(2)}$ with $U^{(2)}$ and $V^{(2)}$ defined in Equation (16), and obtain $\hat{B}_{11}^{-1}$ and $\hat{m}$ from Equation (17).

FIG. 1. Major step I.

Then

$$U^{(2)}U^{(1)}U\hat{B}VV^{(1)}V^{(2)} = \begin{pmatrix} \overline{B}_{11} & 0 & 0 \\ 0 & \delta & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{17}$$

where $\delta = \beta^* w^*(1 - w_1 \overline{B}_{11}^{-1}\beta_1)$. If $\delta = 0$ then $\hat{B}_{11}^{-1} = \overline{B}_{11}^{-1}$ and $\hat{r} = r$, else

$$\hat{B}_{11}^{-1} = \begin{pmatrix} \overline{B}_{11}^{-1} & 0 \\ 0 & 1/\delta \end{pmatrix} \quad \text{and} \quad \hat{r} = r + 1.$$

It takes $2r^2 + 2rm$ flops to compute the matrices $U^{(2)}U^{(1)}U$ and $VV^{(1)}V^{(2)}$. Major step I is now completed, after performing approximately $9m^2 + 4m(m - r)$ flops. It is summarized in Figure 1.

We now examine the validity of the nonsingularity assumption for $\overline{B}_{11}$. There are applications in which $\overline{B}_{11}$ is in fact singular. This situation can occur, for instance, with Markov decision chains. For example, with

$$P = \begin{pmatrix} 0.5 & 0.5 \\ 0 & 1 \end{pmatrix}, \qquad U = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \quad \text{and} \quad V = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix},$$

we have $B_{11} = (0.5)$, where $B = I - P'$. Changing the first row of P to $(1, 0)$, we have indeed two vectors β and w such that $\hat{B} = B + \beta w$ and hence $B_{11} + \beta_1 w_1 = (0)$, which is singular. The above transition matrix P has a transient state, and $\hat{P}$ has two recurrent classes.

In the special case when both transition matrices P and $\hat{P}$ are irreducible, we now show how the factorization can be performed so that the submatrix $B_{11} + \beta_1 w_1$ is always nonsingular and the algorithm of Figure 1 is valid. Because the matrix A is an identity matrix, we need that $V = U^{-1}$ in order for Equation (2) to be satisfied.

LEMMA 5.1. *Suppose P and $\hat{P}$ are irreducible, stochastic $m \times m$ matrices, and let $B = I - P'$ and $\hat{B} = I - \hat{P}'$. Then there is a nonsingular matrix U such that*

$$UBU^{-1} = \begin{pmatrix} B_{11} & 0 \\ 0 & 0 \end{pmatrix} \quad \textit{and} \quad U\hat{B}U^{-1} = \begin{pmatrix} \hat{B}_{11} & \hat{B}_{12} \\ 0 & 0 \end{pmatrix},$$

where B_{11} and $\hat{B}_{11}$ are nonsingular.

Proof. By irreducibility, we have $r = \hat{r} = m - 1$. Hence we can write

$$P = \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & p_{22} \end{pmatrix} \quad \text{and} \quad \hat{P} = \begin{pmatrix} \hat{P}_{11} & \hat{P}_{12} \\ \hat{P}_{21} & \hat{p}_{22} \end{pmatrix},$$

where p_{22} and $\hat{p}_{22}$ are scalars. Let $e' = (1, \dots, 1)$, and construct a square, nonsingular matrix X as follows:

$$X = \begin{pmatrix} I & 0 \\ e' & 1 \end{pmatrix}.$$

Then we have

$$XBX^{-1} = \begin{pmatrix} B_{11} & -P_{21}' \\ 0 & 0 \end{pmatrix} \quad \text{and} \quad X\hat{B}X^{-1} = \begin{pmatrix} \hat{B}_{11} & -\hat{P}_{21}' \\ 0 & 0 \end{pmatrix},$$

where $B_{11} = I - P_{11}' + P_{21}'e'$, $\hat{B}_{11} = I - \hat{P}_{11}' + \hat{P}_{21}'e'$. The nonsingularity of B_{11} and $\hat{B}_{11}$ follows from Lemma 2.1 of [13], because ind B = ind $\hat{B} = 1$ and $r = \hat{r} = m - 1$. Then the result is obtained by taking $U = Y^{-1}X$, where

$$Y = \begin{pmatrix} I & B_{11}^{-1}P_{21}' \\ 0 & 1 \end{pmatrix}.$$

■

At major step II, we are to obtain the matrices $\hat{A}_{11}$ and $\hat{A}_{22}$. First we compute the vector $U\alpha$, and then, using the already computed vector wV, we get $\bar{A} = U\hat{A}V$ with

$$\bar{A} = \begin{pmatrix} A_{11} & 0 \\ 0 & A_{22} \end{pmatrix} + (U\alpha)(wV), \tag{18}$$

which we compute explicitly. Next we apply the transformations of major step I to the matrix $\bar{A}$, giving

$$U^{(2)}U^{(1)}U\hat{A}VV^{(1)}V^{(2)} = \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{pmatrix}, \tag{19}$$

where $\bar{A}_{11}$ is an $\hat{r} \times \hat{r}$ matrix and $\bar{A}_{22}$ is an $(m - \hat{r}) \times (m - \hat{r})$ matrix.

Lemma 5.2 below establishes that $\bar{A}_{22}$ is nonsingular. Let us define the matrices

$$u^{(3)} = \begin{pmatrix} I & -\bar{A}_{12}\bar{A}_{22}^{-1} \\ 0 & I \end{pmatrix} \quad \text{and} \quad V^{(3)} = \begin{pmatrix} I & 0 \\ -\bar{A}_{22}^{-1}\bar{A}_{21} & I \end{pmatrix}. \tag{20}$$

The factorization of Equation (10) is now completely obtained, with $\hat{U} = U^{(3)}U^{(2)}U^{(1)}U$, $\hat{V} = VV^{(1)}V^{(2)}V^{(3)}$,

$$\hat{A}_{11} = \bar{A}_{11} - \bar{A}_{12}\bar{A}_{22}^{-1}\bar{A}_{21}, \quad \text{and} \quad \hat{A}_{22} = \bar{A}_{22}. \tag{21}$$

Major step II is now completed, after performing approximately $[6 + 2(m - r)]m^2$ flops. It is summarized in Figure 2. The error message at step 3 diagnoses that one of the hypotheses of Lemma 5.2 is violated. This situation could occur, for instance, if the variable to leave the basis were not selected properly.

LEMMA 5.2. *Suppose there exists a real number d for which the matrix $\hat{A} + d\hat{B}$ is nonsingular, and also that* $\operatorname{ind}(\hat{A}, \hat{B}) = 1$. *Then the matrix $\bar{A}_{22}$ of Equation* (19) *is nonsingular.*

Proof. Let $\bar{U} = U^{(2)}U^{(1)}U$ and $\bar{V} = VV^{(1)}V^{(2)}$. Then both $\bar{U}$ and $\bar{V}$ are nonsingular by construction. By (17) and (19), we have

$$\bar{U}\hat{A}\bar{V} = \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{pmatrix} \quad \text{and} \quad \bar{U}\hat{B}\bar{V} = \begin{pmatrix} \hat{B}_{11} & 0 \\ 0 & 0 \end{pmatrix},$$

Step 1. Compute $U\alpha$, $\bar{A}$ from Equation (18) and $U^{(2)}U^{(1)}\bar{A}V^{(1)}V^{(2)}$.

Step 2. If $\bar{A}_{22}$ is nonsingular, then compute $\hat{U} = U^{(3)}U^{(2)}U^{(1)}U$ and $\hat{V} = VV^{(1)}V^{(2)}V^{(3)}$ with $U^{(3)}$ and $V^{(3)}$ defined in Equation (20), and $\hat{A}_{11}$ and $\hat{A}_{22}$ as in Equation (21), else

Step 3. "Error: the matrix $\hat{A} + t\hat{B}$ is singular for all t or $\operatorname{ind}(\hat{A}, \hat{B}) \neq 1$."

FIG. 2. Major step II.

where $\hat{B}_{11}$ is nonsingular. Now let $Y = [\bar{U}(\hat{A} + d\hat{B})\bar{V}]^{-1}$. Then

$$Y = \begin{pmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{pmatrix} = \begin{pmatrix} \bar{A}_{11} + d\hat{B}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{pmatrix}^{-1},$$

and hence, using inversion by partitioning, we have that $\bar{A}_{22}$ is nonsingular if and only if Y_{11} is nonsingular. The result will follow after we prove that Y_{11} is nonsingular. Let $X = \bar{V}^{-1}(\hat{A} + d\hat{B})^{-1}\hat{B}\bar{V}$. Then ind $X = \mathrm{ind}(\hat{A}, \hat{B}) = 1$, by hypothesis. But

$$X = \begin{pmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{pmatrix}\begin{pmatrix} \hat{B}_{11} & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} Y_{11}\hat{B}_{11} & 0 \\ Y_{21}\hat{B}_{11} & 0 \end{pmatrix},$$

and hence, by Lemma 2.1 of [13], $Y_{11}\hat{B}_{11}$ is nonsingular. ■

6. WHEN THE MODIFIED SUBMATRIX IS SINGULAR

In this section, the algorithm of Section 5 is extended to the case when the submatrix $B_{11} + \beta_1 w_1$ of Equation (12) is singular. Only steps 3 and 4 of major step I need to be modified. Major step II is unchanged. Let $C_{11} = B_{11}^{-1}$, $C_{i\cdot}$ be the ith row of C_{11}, and $C_{\cdot j}$ be the jth column of C_{11}. Because $B_{11} + \beta_1 w_1$ is singular, we have that

$$\gamma = w_1 C_{11}\beta_1 = \sum_{i=1}^{r} w_{1i}(C_{i\cdot}\beta_1) = \sum_{j=1}^{r}(w_1 C_{\cdot j})\beta_{1j} = -1, \tag{22}$$

and hence there is at least one row k such that $C_{k\cdot}\beta_1 \neq 0$ and at least one column l such that $w_1 C_{\cdot l} \neq 0$. Without loss of generality, let us assume that $k = r$ and $l = r$ (otherwise, we simply interchange the corresponding rows and columns).

Suppose also that $r \geqslant 2$ (the case $r = 1$ is trivial). Now let subscript 0 denote the range $1, \ldots, r-1$, and partition the matrices B_{11} and C_{11} as follows:

$$B_{11} = \begin{pmatrix} B_{00} & B_{0r} \\ B_{r0} & b_{rr} \end{pmatrix} \quad \text{and} \quad C_{11} = \begin{pmatrix} C_{00} & C_{0r} \\ C_{r0} & c_{rr} \end{pmatrix}.$$

The original matrix B is now partitioned as follows:

$$B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} = \begin{pmatrix} \begin{pmatrix} B_{00} & B_{0r} \\ B_{r0} & b_{rr} \end{pmatrix} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}.$$

Let also w_0 and w_r (β_0 and β_r) denote the corresponding subvectors of w_1 (β_1). The well-known identities resulting from $B_{11}C_{11} = I$ and $C_{11}B_{11} = I$ will be used throughout.

THEOREM 6.1. *Suppose* $w_1 C_{\cdot r} \neq 0$ *and* $C_{r\cdot}\beta_1 \neq 0$. *Then* $B_{00} + \beta_0 w_0$ *is nonsingular.*

Proof. There are two cases. Case 1 is when $c_{rr} \neq 0$, and case 2 is when $c_{rr} = 0$. In case 1, $c_{rr} \neq 0$ implies that the submatrix B_{00} is nonsingular and, through inversion by partitioning, we have

$$B_{00}^{-1} = C_{00} - \frac{C_{0r}C_{r0}}{c_{rr}}. \tag{23}$$

Moreover, let $\epsilon = w_0 B_{00}^{-1}\beta_0$. Then it is straightforward to show that

$$\epsilon = \gamma - \frac{(w_1 C_{\cdot r})(C_{r\cdot}\beta_1)}{c_{rr}}, \tag{24}$$

where $\gamma = -1$ by Equation (22) because the matrix $B_{11} + \beta_1 w_1$ is singular. By hypothesis, the second term on the right of Equation (24) is nonzero. Hence $\epsilon \neq -1$ and the result follows, with

$$(B_{00} + \beta_0 w_0)^{-1} = B_{00}^{-1} - \frac{(B_{00}^{-1}\beta_0)(w_0 B_{00}^{-1})}{1+\epsilon} \tag{25}$$

being obtained by applying the Sherman-Morrison formula.

We now turn to case 2, in which $c_{rr} = 0$, and hence the submatrix B_{00} is singular. First, we assume that $b_{rr} \neq 0$, so that C_{00} is nonsingular. Using this, we derive a formula for $(B_{00} + \beta_0 w_0)^{-1}$. We will show later that the same formula is also valid when $b_{rr} = 0$. Using inversion by partitioning, we write

$$C_{00}^{-1} = B_{00} - \frac{B_{0r} B_{r0}}{b_{rr}},$$

so that

$$B_{00} + \beta_0 w_0 = C_{00}^{-1} + MN, \tag{26}$$

where

$$M = \begin{pmatrix} \dfrac{1}{b_{rr}} B_{0r} & \beta_0 \end{pmatrix} \quad \text{and} \quad N = \begin{pmatrix} B_{r0} \\ w_0 \end{pmatrix}.$$

Then M is an $(r-1) \times 2$ matrix and N is $2 \times (r-1)$. The Woodbury formula (see, e.g., [9]) can be used to invert (26) if the capacitance matrix $I + NC_{00}M$ is nonsingular. Using the identities $B_{r0}C_{00} = -b_{rr}C_{r0}$, $C_{00}B_{0r} = -C_{0r}b_{rr}$, and $C_{r0}B_{0r} = 1$, we get

$$I + NC_{00}M = \begin{pmatrix} b_{rr} & 0 \\ 0 & 1 \end{pmatrix} S,$$

where

$$S = \begin{pmatrix} 0 & -C_{r0}\beta_0 \\ -w_0 C_{0r} & 1 + \nu \end{pmatrix}$$

and $\nu = w_0 C_{00} \beta_0$. Indeed, the matrix S is nonsingular, because $c_{rr} = 0$ implies that $w_0 C_{0r} = w_1 C_{\cdot r}$ and $C_{r0}\beta_0 = C_{r\cdot}\beta_1$, both being nonzero by hypothesis. Further, its inverse is given by

$$S^{-1} = \begin{pmatrix} \dfrac{-(1+\nu)}{(w_0 C_{0r})(C_{r0}\beta_0)} & \dfrac{-1}{w_0 C_{0r}} \\ \dfrac{-1}{C_{r0}\beta_0} & 0 \end{pmatrix}.$$

Then the Woodbury formula gives

$$\begin{aligned}(C_{00}^{-1} + MN)^{-1} &= C_{00} - (C_{00}M)(I + NC_{00}M)^{-1}(NC_{00}) \\ &= C_{00} - (-C_{0r} \quad C_{00}\beta_0)S^{-1}\begin{pmatrix} -C_{r0} \\ w_0 C_{00} \end{pmatrix} \\ &= Z_{00},\end{aligned}$$

where

$$Z_{00} = C_{00} + \frac{(1+\nu)C_{0r}C_{r0}}{(w_0 C_{0r})(C_{r0}\beta_0)} - \frac{C_{0r}(w_0 C_{00})}{w_0 C_{0r}} - \frac{(C_{00}\beta_0)C_{r0}}{C_{r0}\beta_0}. \tag{27}$$

By (26), we have that Z_{00} is the inverse of $B_{00} + \beta_0 w_0$, provided $b_{rr} \neq 0$. Observe now that the formula (27) does not use the quantity b_{rr}. In fact, it uses only the given, well-defined submatrices of C_{11}. Moreover, all its denominators are nonzero, independent of b_{rr}. Let us now verify that (27) gives the inverse of $B_{00} + \beta_0 w_0$. Using the identities $B_{00}C_{0r} = 0$ and $B_{00}C_{00} + B_{0r}C_{r0} = I$, we have indeed

$$\begin{aligned}(B_{00} + \beta_0 w_0)Z_{00} &= B_{00}C_{00} - B_{00}C_{00}\frac{\beta_0 C_{r0}}{C_{r0}\beta_0} + \frac{\beta_0 C_{r0}}{C_{r0}\beta_0} \\ &\quad + B_{00}C_{0r}\left[\frac{(1+\nu)C_{r0}}{(w_0 C_{0r})(C_{r0}\beta_0)} - \frac{w_0 C_{00}}{w_0 C_{0r}}\right] \\ &= I - B_{0r}C_{r0} - \frac{\beta_0 C_{r0}}{C_{r0}\beta_0} + B_{0r}C_{r0} + \frac{\beta_0 C_{r0}}{C_{r0}\beta_0} = I.\end{aligned}$$

The verification was carried out entirely without using the value of b_{rr}. Hence the result follows. ∎

Step 3 of major step I can now be described as follows. Find a row k of C_{11} such that $C_{k\cdot}\beta_1 \neq 0$, and interchange *columns* k and r of V, A_{11}, and w_1. Find a column l of C_{11} such that $wC_{\cdot l} \neq 0$, and interchange *rows* l and r of U, A_{11}, and β_1. The conditions of Theorem 6.1 are now satisfied. If $c_{rr} \neq 0$ then compute $(B_{00} + \beta_0 w_0)^{-1}$ using (23) and (25), else use Equation

(27). In either case, only the entries of C_{11}, w_1, and β_1 are used. The submatrix B_{11} itself is never used.

At the beginning of step 4, we have

$$U^{(1)}U\hat{B}VV^{(1)} = \begin{pmatrix} B_{00} + \beta_0 w_0 & B_{0r} + \beta_0 w_r & \beta_0 w^* & 0 \\ B_{r0} + \beta_r w_0 & b_{rr} + \beta_r w_r & \beta_r w^* & 0 \\ \beta^* w_0 & \beta^* w_r & \beta^* w^* & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad (28)$$

which replaces Equation (14). We need to eliminate the rows r and $r+1$ and columns r and $r+1$ of (28). To do this, we compute the vectors

$$X^{(2)} = -(B_{r0} + \beta_r w_0)(B_{00} + \beta_0 w_0)^{-1},$$

$$X^{(3)} = -\beta^* w_0 (B_{00} + \beta_0 w_0)^{-1},$$

$$Y^{(2)} = -(B_{00} + \beta_0 w_0)^{-1}(B_{0r} + \beta_0 w_r),$$

$$Y^{(3)} = -(B_{00} + \beta_0 w_0)^{-1}\beta_0 w^*,$$

and define the matrices

$$U^{(2)} = \begin{pmatrix} I & 0 & 0 & 0 \\ X^{(2)} & 1 & 0 & 0 \\ X^{(3)} & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad V^{(2)} = \begin{pmatrix} I & Y^{(2)} & Y^{(3)} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Then

$$U^{(2)}U^{(1)}U\hat{B}VV^{(1)}V^{(2)} = \begin{pmatrix} B_{00} + \beta_0 w_0 & 0 & 0 & 0 \\ 0 & 0 & \mu_{r,r+1} & 0 \\ 0 & \mu_{r+1,r} & \mu_{r+1,r+1} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad (29)$$

where

$$\mu_{r,r+1} = \beta_r w^* + X^{(2)}\beta_0 w^*,$$

$$\mu_{r+1,r} = \beta^* w_r + \beta^* w_0 Y^{(2)},$$

$$\mu_{r+1,r+1} = \beta^* w - \beta^* w_0 (B_{00} + \beta_0 w_0)^{-1} \beta_0 w^*.$$

Now let

$$E = \begin{pmatrix} 0 & \mu_{r,r+1,} \\ \mu_{r+1,r} & \mu_{r+1,r+1} \end{pmatrix}.$$

Then $\hat{r} = r - 1 + \text{rank } E$, and hence we can have $\hat{r} = r - 1$, $\hat{r} = r$, and $\hat{r} = r + 1$. The matrix $\hat{B}_{11}^{-1}$ can be constructed by modifying (29) after the matrix E is reduced, if rank $E < 2$.

It is worth mentioning here that the vectors $X^{(2)}$ and $Y^{(2)}$ can be computed without using any of the entries of B_{11}. This is because both formulas (23) and (27) are entirely written in terms of C_{00}, C_{0r}, and C_{r0}. After simplification, we get

$$(B_{00} + \beta_0 w_0)^{-1} B_{0r} = \begin{cases} \dfrac{(B_{00}^{-1}\beta_0)(w_0 C_{0r})}{(1+\epsilon)c_{rr}} - \dfrac{C_{0r}}{c_{rr}} & \text{if} \quad c_{rr} \neq 0, \\[2ex] \dfrac{(1+\nu)C_{0r}}{(w_0 C_{0r})(C_{r0}\beta_0)} - \dfrac{C_{00}\beta_0}{C_{r0}\beta_0} & \text{else.} \end{cases}$$

A similar formula holds for $B_{0r}(B_{00} + \beta_0 w_0)^{-1}$.

The author would like to thank the anonymous referees for their helpful comments on earlier versions of the manuscript.

REFERENCES

1 A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, Mass., 1974.

2 K. M. Anstreicher, Generation of Feasible Descent Directions in Continuous Time Linear Programming, Ph.D. Dissertation, Stanford Univ., Stanford, Calif. 1983.

3 D. Blackwell, Discrete dynamic programming, *Ann. Math. Statist.* 33:719–726 (1962).
4 S. L. Campbell and C. D. Meyer, Jr., *Generalized Inverses of Linear Transformations*, Pitman, London, 1979.
5 S. L. Campbell, C. D. Meyer, and N. J. Rose, Applications of the Drazin inverse to linear systems of differential equations with singular constant coefficients, *SIAM J. Appl. Math.* 31:411–425 (1976).
6 R. Dekker, Denumerable Markov Decision Chains: Optimal Policies for Small Interest Rates, Ph.D. Dissertation, Univ. of Leiden, The Netherlands, 1985.
7 F. R. Gantmacher, *The Theory of Matrices*, Vols. I, and II, Chelsea, New York, 1959.
8 G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins U.P., Baltimore, 1983.
9 W. W. Hager, Updating the inverse of a matrix, *SIAM Rev.* 31:221–239 (1989).
10 A. Hordijk, R. Dekker, and L. C. M. Kallenberg, Sensitivity analysis in discounted Markovian decision problems, *OR Spektrum* 7:143–151 (1985).
11 Y. Huang, A Canonical Form for Pencils of Matrices with Applications in Asymptotic Linear Programs, Working Paper, Stanford Univ., 1991.
12 R. G. Jeroslow, Asymptotic linear programming, *Oper. Res.* 21:1128–1141 (1973).
13 B. F. Lamond, An efficient factorization for the group inverse, *SIAM J. Algebraic Discrete Methods* 8:797–808 (1987).
14 B. F. Lamond, A generalized inverse method for asymptotic linear programming, *Math. Programming* 43:71–86 (1989).
15 B. F. Lamond and M. L. Puterman, Generalized inverses in discrete time Markov decision processes, *SIAM J. Matrix Anal. Appl.* 10:118–134 (1989).
16 D. G. Luenberger, Time-invariant descriptor systems, *Automatica* 14:473–480 (1978).
17 B. L. Miller and A. F. Veinott, Jr., Discrete dynamic programming with a small interest rate, *Ann. Math. Statist.* 40:366–370 (1969).
18 A. F. Veinott, Jr., Discrete dynamic programming with sensitive discount optimality criteria, *Ann. Math. Statist.* 40:1635–1660 (1969).
19 J. H. Wilkinson, Note on the practical significance of the Drazin inverse, in *Recent Applications of Generalized Inverses* (S. L. Campbell, Ed.), Pitman, Boston, 1982, pp. 82–99.

Received 23 April 1990; final manuscript accepted 21 April 1992