SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 1, pp. 209–224

RECURSIVE ALGORITHM FOR THE FUNDAMENTAL/GROUP INVERSE MATRIX OF A MARKOV CHAIN FROM AN EXPLICIT FORMULA*

ISAAC SONIN† AND JOHN THORNTON†

Abstract. We present a new accurate algorithm (REFUND) for computing the fundamental matrix (or closely related group inverse matrix) of a finite regular Markov chain. This algorithm is developed within the framework of the state reduction approach exemplified by the GTH (Grassmann, Taksar, Heyman)/S (Sheskin) algorithm for recursively finding invariant measure. The first (reduction) stage of the GTH/S algorithm is shared by REFUND, as well as by an earlier algorithm FUND developed for the fundamental matrix by Heyman in 1995, and by a modified version of Heyman and O'Leary in 1998. Unlike FUND, REFUND is recursive, being based on an explicit formula relating the group inverse matrix of an initial Markov chain and the group inverse matrix of a Markov chain with one state removed. Operation counts are approximately the same: $\Theta(\frac{7}{3}n^3)$ for REFUND versus $\Theta(\frac{8}{3}n^3)$ for FUND. Numerical tests indicate that REFUND is accurate. The structure of REFUND makes it easily combined with the other algorithms based on the state reduction approach. We also discuss the general properties of this approach, as well as connections to the optimal stopping problem and to tree decompositions of graphs related to Markov chains.

Key words. Markov chain, fundamental/group inverse matrix, recursive algorithm, GTH/S algorithm

AMS subject classifications. Primary, 60J10; Secondary, 65U05

PII. S0895479899351234

1. Introduction.

Let $P = [p(i,j)]$, $i,j = 1,2,\dots,n$, be a stochastic (transition) matrix. The calculation of various characteristics of a Markov chain specified by P is an important part of applied probability theory and computational algebra. These characteristics include the distribution of a Markov chain at the moment of the first visit to a subset of its state space, the mean time spent at given states until such visit, the invariant distribution, the fundamental matrices for both transient and regular Markov chains, the covariance matrix, and many others. Mainly we will discuss two of the most important ones, the invariant distribution and the fundamental matrix for a regular Markov chain.

The *invariant* (steady state) *distribution* $\boldsymbol{\pi}$ is the solution of the system of linear equations

$$\boldsymbol{\pi}^T = \boldsymbol{\pi}^T P, \tag{1.1}$$

where ^{T} denotes transposition and all vectors are assumed to be column vectors. In the regular (ergodic) case, i.e., when there is a k for which all elements of P^k are strictly positive, π is the limiting distribution for any initial point. The matrix

$$A = \lim_n P^n = \boldsymbol{e}\boldsymbol{\pi}^T \tag{1.2}$$

has all rows equal to the vector $\boldsymbol{\pi}^T$; and $\mathbf{e}$ is a vector all of whose entries are ones.

*Received by the editors February 5, 1999; accepted for publication (in revised form) by C. Meyer November 30, 2000; published electronically July 2, 2001.
http://www.siam.org/journals/simax/23-1/35123.html
†Department of Mathematics, University of North Carolina at Charlotte, Charlotte, NC 28223 (imsonin@email.uncc.edu, thornton@uncc.edu).

The *fundamental matrix* Z, for the regular case, is given (see [10]) by

$$Z = (I - (P - A))^{-1} = I + \sum_{n=1}^{\infty}(P - A)^n = I + \sum_{n=1}^{\infty}(P^n - A). \tag{1.3}$$

Instead of calculating Z, we will calculate the *group (generalized) inverse matrix* V,

$$V = Z - A = \sum_{n=0}^{\infty}(P^n - A). \tag{1.4}$$

This matrix has a simple relationship to Z, but it has its own important role. Its significance was explained in a pioneering paper [15], which also discusses the relationship between the group inverse and other generalized matrix inverses. For applications of the group inverse to Markov decision processes, see [14] (and references therein) and the comprehensive monograph [18] (especially Appendix A.5). The elements $v(x, y)$ of V have the following probabilistic interpretation (see [10]):

$$v(x, y) = \lim_n E_x[\eta^n(y) - n\pi(y)], \tag{1.5}$$

where $\eta^n(y)$ is the number of visits to y during the first n moments; and E_x denotes mathematical expectation, given that the Markov chain starts from the initial point x. Thus the $v(x, y)$ measure the expected deviation in the number of visits to state y due to starting in state x instead of starting randomly according to the invariant distribution $\boldsymbol{\pi}$.

The classical formulas (closed form solutions) for V and $\boldsymbol{\pi}$, as well as many other related probabilistic quantities, are well known (see, e.g., [10]) and involve matrix inversion or the solution of a system of linear equations.

There is a vast literature on methods for computing various characteristics of Markov chains. We refer the reader to [16], [31], and the proceedings in which [29] appears, which give a thorough description of the current situation in this field and describe both traditional and some more recent methods to calculate characteristics of Markov chains.

The development of a new class of algorithms was initiated in 1985 by two pioneering papers in which Sheskin [19], and Grassmann, Taksar, and Heyman [2] independently proposed practically the same algorithm to calculate invariant distribution. Later it became known as the GTH algorithm. Taking into account the short but very precise paper of Sheskin [19], we refer to it as the GTH/S algorithm.

The algorithm constructs a sequence of stochastic matrices, each having dimension one less than the previous, and has a simple and transparent probabilistic interpretation (see section 2). Numerous papers (see more references in section 2 and in volumes [16] and [31]) have studied the computational properties of this algorithm, different generalizations, and particular cases. It has been shown, among other things, that the GTH/S algorithm has significant advantages over traditional methods to calculate $\boldsymbol{\pi}$.

In 1995 on the basis of this algorithm, Heyman proposed an algorithm FUND [5] for the sequential computation of the fundamental matrix of a regular Markov chain. In 1998 it was improved and modified by Heyman and O'Leary in [8]. This algorithm uses the idea, outlined by Grassmann in [3], of a triangular factorization of the matrix $(I - P)$ that is produced by the first stage of the GTH/S algorithm.

The main goal of this paper is to present a new algorithm, REFUND, to calculate the fundamental/group inverse matrix. This algorithm, like Heyman's FUND, begins

with the sequence of stochastic matrices constructed by the GTH/S algorithm. The primary distinction is that we forego the triangular factorization, basing this algorithm instead on an explicit formula that relates the group inverse matrices of two stochastic matrices that are adjacent in the sequence of matrices produced by the GTH/S algorithm. A very similar formula can be written for the fundamental matrix. The repeated application of this formula recursively produces the sequence of associated group inverse matrices.

This provides us with the opportunity to begin calculation with any submodel for which the fundamental matrix or group inverse matrix is known, and to bring probabilistic (in addition to numerical) techniques to bear in analyzing where accuracy is lost; and it aids us in taking corrective steps. Like FUND, REFUND requires $O(n^3)$ arithmetic operations to complete, where P is $n \times n$, although REFUND, with a leading constant of $\frac{7}{3}$ versus $\frac{8}{3}$ for FUND, is slightly faster. Like FUND, REFUND can also be applied to calculate the fundamental matrix for a continuous time Markov chain.

The GTH/S algorithm, Heyman's FUND algorithm, REFUND, Sheskin's algorithm [22] to compute the fundamental matrix of a transient Markov chain, the algorithm of optimal stopping of Markov chains proposed in [25] and some others can be viewed as examples of the application of a more general approach, which can be called the state reduction (SR) approach. The elements of the SR approach can be found in the works of many authors, so we do not claim authorship. But we have found no attempts, other than [26], to analyze these algorithms together in a general framework. Since the approach itself has become important enough, the brief presentation of an overview is another goal of our paper. We begin this in section 2. Although the reading of that section is not strictly necessary to a purely formal understanding of the REFUND algorithm, it does furnish a general framework in which all SR algorithms can be compared. (Part of this description and related results were presented in [26].) Section 2 also contains brief descriptions of the GTH/S and FUND algorithms. In section 3 we present Theorem 3.1, which provides an auxiliary characterization of the group inverse matrix, and our main result, Theorem 3.2, which provides the exact formula(s) on which our algorithm is based. Section 4 specifies the REFUND algorithm, gives operation counts, summarizes the results of numerical testing, interprets these results, and compares REFUND to FUND. A detailed study of the computational properties of REFUND and a comparison to FUND was presented in [29].

2. The GTH/S and FUND algorithms, the SR approach, and related problems. In our subsequent presentation an important role is played by the transformations of state spaces and transition matrices. So in what follows, instead of the term "Markov chain," we prefer to use the term "Markov model." A *Markov model* M is a pair (X, P), where X is a finite or countable state space, and P is a stochastic matrix, indexed by elements of X.

In his original paper, Sheskin [19] states that the GTH/S algorithm is motivated by a result from Kemeny and Snell [10], while Grassmann [3] and Heyman [5] describe GTH/S as a variant of Gaussian elimination. Though it is difficult to object to either statement, at the same time (in our opinion) it can be said that the SR approach is based on the following simple probabilistic idea that appeared in the pioneering works of Kolmogorov and Döeblin more than sixty years ago. This idea, described in Proposition 2.1 below, has been used since that time in probability theory in several contexts on numerous occasions.

Let us assume that a finite Markov model $M_1 = (X_1, P_1)$ is given and let (Z_n), $n = 1, 2, \dots$, be a Markov chain specified by the model M_1. Let $X_2 \subset X_1$ and let $\tau_1, \tau_2, \dots, \tau_n, \dots$, be the sequence of Markov times of first, second, and so on visits of (Z_n) to the set X_2, so that $\tau_1 = \min\{k > 0 : Z_k \in X_2\}$, $\tau_{n+1} = \min\{k : \tau_n < k, Z_k \in X_2\}$, $0 < \tau_1 < \tau_2 < \dots$. Let $u_1^{X_2}(x, \cdot)$ be the distribution of Markov chain (Z_n) for the initial model M_1 at the moment τ_1 of first visit to set X_2 (first exit from $X_1 \setminus X_2$) starting at x, $x \in X_1 \setminus X_2$. Let us consider the random sequence $Y_n = Z_{\tau_n}, n = 1, 2, \dots$.

PROPOSITION 2.1. (a) *The random sequence* (Y_n) *is a Markov chain in a model* $M_2 = (X_2, P_2)$, *where* (b) *the transition matrix* $P_2 = \{p_2(i, j)\}$ *is given by the formula*

$$p_2(i,j) = p_1(i,j) + \sum_{x \in X_1 \setminus X_2} p_1(i,x) u_1^{X_2}(x,j), \quad i, j \in X_2. \tag{2.1}$$

Part (a) is immediately implied by the strong Markov property for (Z_n), while the proof of part (b) is straightforward.

Formula (2.1) can be represented in matrix form. This representation is proved, for example, in [10, pp. 114–116]. For the sake of brevity, we will call M_2 the (X_2-) *reduced* model of M_1. (Proposition 2.1 is also true for countable X, with minor modifications.)

An important case is when the set $X_1 \setminus X_2$ consists of one point z. In this case formula (2.1) obviously takes the form

$$p_2(i,j) = p_1(i,j) + \frac{p_1(i,z)p_1(z,j)}{(1 - p_1(z,z))}, \quad (i, j \in X_2). \tag{2.2}$$

According to this formula, each row-vector of the new stochastic matrix P_2 is a linear combination of two rows of P_1 (with the z-column deleted). For a given row of P_2, these two rows are the corresponding row of P_1 and the zth row of P_1. This transformation corresponds formally to one step of the Gaussian elimination method.

It is easy to understand that although the initial and reduced Markov models are different, some of their characteristics will either coincide or be related in a simple way.

The theoretical basis for the GTH/S algorithm is provided by Proposition 2.2, which we formulate here for the case where the set $X_1 \setminus X_2$ consists of one point z. It shows the relation between the invariant distribution in the initial and the reduced models.

PROPOSITION 2.2. *Let* $M_1 = (X_1, P_1)$ *be a Markov model,* $X_1 = X_2 \cup \{z\}$, *and let* $M_2 = (X_2, P_2)$ *be the corresponding* X_2*-reduced Markov model with* $p_2(i, j)$ *defined according to* (2.2). *Let set* X_2 *and state* z *communicate in the model* M_1; *i.e., there are states* i, $j \in X_2$, *such that* $p_1(i, z) > 0$, $p_1(z, j) > 0$. *Then*

(a) *if the invariant distribution* $\pi_2(\cdot)$ *exists in model* M_2, *the invariant distribution* $\pi_1(\cdot)$ *also exists (in* M_1*) and can be calculated by the formulas* $\left(\sum_{y \in X_2} = \sum_y\right)$

$$\pi_1(j) = \alpha_1 \pi_2(j), \quad j \in X_2, \tag{2.3}$$

$$\pi_1(z) = \alpha_1 \sum_i \pi_2(i) p_1(i,z)/s_1 \equiv 1 - \alpha_1, \tag{2.4}$$

where

$$\alpha_1 = 1/\left(1 + \sum_i \pi_2(i) p_1(i,z)/s_1\right), \qquad s_1 = 1 - p_1(z,z). \tag{2.5}$$

(b) *If the invariant distribution* $\pi_1(\cdot)$ *exists in model* M_1*, then the invariant distribution* $\pi_2(\cdot)$ *also exists (in* M_2*) and is given by formula* (2.3)*, with* $\alpha_1 = 1 - \pi_1(z)$.

Relations (2.3) and (2.4) have a transparent probabilistic meaning. The invariant distribution is the long-term proportion of time spent at a state. Therefore, the invariant distributions must be proportional on X_2, i.e., equality (2.3) holds. Formula (2.4) can be easily received from (2.3) and a balance equation for distribution $\boldsymbol{\pi}_1$ at the point z. The formulations of Propositions 2.1 and 2.2, as well as a formal proof of the latter, omitted here, were given in [26].

2.1. The GTH/S algorithm. We now describe briefly the GTH/S algorithm as given in [2] and [19]. In contrast to those papers, we index the states to be eliminated in the order that is customary for Gaussian elimination without pivoting, beginning with state number one.

2.1.1. GTH/S reduction stage (generic for SR algorithms). Let an initial Markov model $(X, P) = M \equiv M_1 = (X_1, P_1)$ be given. A sequence of *stochastic* matrices (P_k), $k = 2, \ldots, n$, is calculated recursively on the basis of formula (2.2), in which the subscripts "1" and "2" are replaced by "k" and "k+1," respectively. Each matrix P_k corresponds to a model $M_k = (X_k, P_k)$, $X_k = \{k, k+1, \ldots, n\}$, and has dimension $(n-k+1) \times (n-k+1)$, and P_n is an identity matrix of dimension 1. A Markov chain in a model M_k is specified by a corresponding Markov chain in the initial model at the moments of its visits to the reduced state space X_k. For the subsequent recovery of $\boldsymbol{\pi}$, only the first (scaled) columns of each of the matrices P_k are used.

An important role in maintaining accuracy is played by the sequence $s_1, s_2, \ldots, s_{n-1}$, where each s_k (see (2.2) and (2.5), where the subscript "1" is again replaced by "k") is calculated as the sum

$$s_k = \sum_{j \neq z} p_k(z, j) = 1 - p_k(z, z)$$

rather than the mathematically equivalent difference. This choice avoids subtractive cancellation without adding significantly to computational effort. The kth step (for $k = 1$) of the reduction phase of the GTH/S algorithm can be represented as

$$P_1 = \begin{bmatrix} a & \mathbf{p}^T \\ s\mathbf{q} & Q \end{bmatrix}, \quad P_2 = Q + \mathbf{q}\mathbf{p}^T, \quad \overline{P}_2 = \begin{bmatrix} a & \mathbf{p}^T \\ \mathbf{q} & P_2 \end{bmatrix}, \tag{2.6}$$

where

$$s \equiv s_1 = \mathbf{p}^T \mathbf{e} = 1 - a,$$

and $\overline{P}_2$ is the matrix stored after the first step of computation. Thus $\mathbf{p}^T$ is the first (zth) row, and $\mathbf{q}$ is the first column (scaled by $s = s_1$) of the matrix P_1, both without the first element $p_1(z, z)$.

2.1.2. GTH/S second (recovery) stage. Three normalizations and tree decomposition. Proposition 2.2 provides the possibility to compute the invariant distribution $\boldsymbol{\pi}_k$ for each of the models M_k on the basis of $\boldsymbol{\pi}_{k+1}$ in the model M_{k+1}, beginning from the trivial invariant distribution $\boldsymbol{\pi}_n = \{1\}$ of the model M_n. This can be done in three different ways. The first way of normalizing is to use formulas

(2.3) and (2.4), i.e., to receive each time vector $\boldsymbol{\pi}_k$. From (2.3) and (2.4), this can be represented as follows:

$$\boldsymbol{\pi}_1 = \alpha_1 \begin{bmatrix} \boldsymbol{\pi}_2^T \mathbf{q} \\ \boldsymbol{\pi}_2 \end{bmatrix} \equiv \begin{bmatrix} 1-\alpha_1 \\ \alpha_1 \boldsymbol{\pi}_2 \end{bmatrix}. \tag{2.7}$$

Notice that this way provides an extra opportunity to increase the accuracy of calculations because the sum of the elements of the obtained vector must equal one. This is the formula used later by REFUND.

Because the goal of the GTH/S algorithm is to produce only the invariant distribution $\boldsymbol{\pi}_1$ in the initial model M_1, that algorithm uses a second method of normalizing. The first equality of (2.7) is used with α_k being replaced by 1, i.e., each new vector is calculated by appending a single element to its predecessor. Only the last vector in the sequence is normalized to produce $\boldsymbol{\pi}_1$.

A third way to normalize is to use the first equality of (2.7) again with α_k now being replaced by s_k. This gives valuable information about a tree decomposition of a Markov chain as follows. In [26] the relationship between GTH/S and the interesting formula discussed below was considered. In their 1979 book on large deviations [1], Freidlin and Wentzell used the following interesting approach to calculate $\boldsymbol{\pi}$ on the basis of a tree decomposition. Their book uses the formula $\pi(x) = q(x)/\sum_{y\in X} q(y)$, where $q(y)$ is defined as follows. Let X be a finite set and P be a stochastic matrix. Let T be a *spanning tree directed* to y. This means that T is a connected graph without cycles (tree), that it contains all the vertices of X (spanning), and that a vertex y is designated as a *root*. In any rooted tree with a root y there is a unique path, directed to y, between any vertex v and y; this direction makes the tree a tree directed to y. Let $G(y) = \{\text{spanning trees on } X \text{ directed to } y\}$. Then $q(y) = \sum_{T\in G(y)} r(T)$, where $r(T) = \prod_{(u,v)\in T} p(u,v)$.

Theorem 1 of [26] establishes that $q(x)$ can be computed by normalizing the recovery steps of the GTH/S algorithm by the replacements $\alpha_k = s_k$, instead of $\alpha_k = 1$ as in GTH/S. This opens the way to use results from Markov chain theory to obtain some results in graph theory. Note also that in [26] the relationship between the SR approach and graph-based computational methods in electrical engineering was noted. In particular the formula mentioned above, which relates $\boldsymbol{\pi}$ to $\mathbf{q}$, is well known in electrical engineering as the star mesh transformation, though the interpretation is quite different.

2.2. The two FUND algorithms. Based on the triangular factorization ($LU = I-P$) provided by the first stage of the GTH/S algorithm, Heyman proposed in (1995) [5] the algorithm FUND to calculate the fundamental matrix Z ($= V+A$) of a regular Markov chain. This led to a fast ($O(n^3)$) algorithm that performed accurately on test problems. A paper of Heyman and O'Leary (1998) [8] gives examples for which the factor U is badly ill-conditioned even though $(I-P)$ is not, and presents a new version of FUND which avoids this instability by modifying U and by introducing pivoting. Both versions of the algorithm are based on the equation

$$Z = A + (I - A)X, \tag{2.8}$$

where X is any solution to

$$(I - P)X = (I - A). \tag{2.9}$$

The solution to the latter equation is not unique, as $(I - P)$ is rank-deficient by 1. In both versions, the triangular factorization produced by the GTH/S algorithm

is used (but in somewhat different ways) to find an X which satisfies (2.9), and then Z is found by substituting X into (2.8).

Notice also that the specific form taken by the triangular factorization depends on the order in which states are eliminated: when states are eliminated beginning with the largest indices (as in the usual presentation of the GTH/S algorithm), the triangular factorization has the form $(I - P) = UL$ (upper followed by lower triangular factors); however, when states are eliminated in the order $1, 2, \ldots$, then the factorization takes the familiar LU form.

2.3. The SR approach. We have cited examples of algorithms (the GTH/S algorithm [2], [19], the two FUND algorithms [5], [8], and the elimination algorithm for optimal stopping [25], [26], [27]) that share a common feature: they are based on a sequence of models in which each model (except the first) is constructed from its predecessor by removing states and recalculating transition probabilities according to Proposition 2.1. We will refer to such algorithms as SR algorithms, and to the general approach to their development as the state reduction (SR) approach. Additional SR algorithms include the algorithms to compute mean first passage times and absorption probabilities in Markov and semi-Markov chains that are discussed by Kohlas in [12] and by Sheskin in [21] and [23], the algorithm of Sheskin [22] for calculating the fundamental matrix for a reducible Markov chain, and the algorithms that Lal and Bhat discuss in [13]. (Sheskin also gives algorithms for matrix inversion [20] and for solving linear systems [24], whose structures are similar to those of the above SR algorithms, but no stochastic interpretation is given for them.) Although Proposition 2.1 does provide for the elimination of several, or even infinitely many, states in a single reduction step, the majority of given examples eliminate one state at each reduction step, and we will confine our discussion to those. Such algorithms must begin, up to minor variation, with the reduction stage of the GTH/S algorithm, whose appearance has stimulated an outpouring of works in recent years.

The algorithms under consideration differ only in their portions that follow the standard reduction stage. In this regard, all of the given examples except one, FUND, are recursive in the sense that the necessary reduction stage is followed by a stage of backward iteration during which some characteristic of, or quantity related to, the smallest model is deduced or calculated, and then the analogous characteristic or quantity in each larger model is inferred or calculated from its counterpart in the adjacent smaller model. (Whenever we use the term "recursive" in what follows, we shall mean it in this sense.) The GTH/S algorithm is a good and well-known example in which backward iteration (up to normalization) retraces all reduction steps.

Another example, also based on Proposition 2.1, is an algorithm for optimal stopping proposed by Sonin in 1995 [25] (see also [26] and [27]). Briefly, it can be described as the construction of a sequence of models where each time a set (often, but not always of size one) of states, which have been shown not to belong to the stopping set, is eliminated, and new transition probabilities are computed on the basis of (2.1) or (2.2). The stopping sets in both models coincide, and this offers the possibility of recursive calculation of the stopping set. In contrast to other state reduction algorithms, in this algorithm the number of steps required is not known in advance. In some SR algorithms (e.g., membership in an optimal stopping set, mean time to reach a designated subset of states) the characteristic to be calculated is preserved by reduction (i.e., coincides on the shared portions of the domains of the initial and reduced models). In these cases part (e.g., mean times) or all (e.g., membership in an optimal stopping set) of the backward stage is trivial, but nearly

all of the SR algorithms are recursive. In the sole exception, FUND, the quantity to be calculated is obtained directly by solving a linear system using a triangular factorization received as a byproduct of the GTH/S reduction stage. The algorithm REFUND introduced in this paper is another example of a two-stage algorithm, with the backward stage being nontrivial and based on an explicit formula.

3. Sequential calculation of the fundamental matrix. Let P be a regular (i.e., irreducible, aperiodic with no transients states) finite stochastic matrix. Equivalently, there is some $k > 0$ for which the matrix P^k has all positive elements. We have already defined $\boldsymbol{\pi}$, A, Z, and V in (1.1), (1.2), (1.3), and (1.4), respectively. We also have

$$PA = AP = A, \quad A^n = A \quad \text{for any } n = 1, 2, \dots. \tag{3.1}$$

The following theorem provides a useful characterization of the matrix V. We will substantially use this theorem in the construction of our main result. All elements of this theorem are well known, but we fail to find such a formulation. (Formula (3.2) below is equivalent to Theorem 2.3 in [15]. Compare this also with Theorem 1 in [5] which is very similar but without the uniqueness, or compare it with results in Appendix A.5 in [18].)

THEOREM 3.1. *Let $M = (X, P)$ be a regular Markov model, π be its invariant measure, $A = \lim_{n\to\infty} P^n$, Z be the fundamental matrix, and $V = Z - A$. Then V is the unique solution of the system of equations*

$$V = (I - A) + PV = (I - A) + VP, \tag{3.2}$$

$$\boldsymbol{\pi}^T V = 0, (AV = 0) \tag{3.3}$$

and also satisfies

$$V\mathbf{e} = 0, (VE = VA = 0). \tag{3.4}$$

Note that the equations in (3.2) are of Bellman type in forward and backward time, and that explains why the group inverse (fundamental) matrix plays a role in the theory of Markov decision processes with average criterium. The formulas (3.3) and (3.4) just say that a scalar product of the invariant vector and any column of V is equal to zero, and the sum of every row of V is also equal to zero. Both relationships have a simple probabilistic meaning according to (1.5).

Proof. Let V be the group inverse. Then from (1.4) and (3.1) we have $PV = P(I - A + \sum_{n=1}(P^n - A)) = (P - A) + (P^2 - A) + \dots = V + A - I$, i.e., the first of the formulas (3.2). The second formula in (3.2) is derived similarly. Formulas (3.3) and (3.4) follow immediately from (1.5) since $E_\pi \eta^n(y) = n\pi(y)$ for all n, and $\sum_y E_x \eta^n(y) = n = \sum_y n\pi(y)$. (Formally, from (1.4) and (3.2) we have $AV = A(I - A) + A\sum_{n=1}(P^n - A) = 0$, i.e., (3.3).) Similarly $VA = 0$, i.e., (3.4). To prove the uniqueness of V, let V and V' be two solutions of (3.2). Then $V - V' = P(V - V')$. According to the well-known statement that for a regular matrix P any solution of equation $\mathbf{x} = P\mathbf{x}$ has the form $c\mathbf{e}$, where c is a constant (see [10, Th. 4.1.7]), thus each column of $V - V'$ is $c\mathbf{e}$ for some constant c. By (3.3) $\pi^T(V - V') = 0$, i.e., all these constants are equal to zero. ∎

Let $M_i = (X_i, P_i), i = 1, 2$ be two models with $X_1 = X_2 \cup \{z\}$; M_2 is an X_2-reduced model of M_1, i.e., P_2 is calculated by formula (2.2). Then according to Proposition 2.2 the relation between invariant measures in these models is given by

the formulas (2.3) and (2.4) with the constants defined by (2.5). Without loss of generality $X_1 = \{1, 2, \ldots, n\}$, $X_2 = \{2, 3, \ldots, n\}$, i.e., $z = 1$, but we will continue to use the letter z. Denote $\sum_j \equiv \sum_{j \in X_2} \equiv \sum_{j \geq 2}$. Consider $x, y \in X_2 \subset X_1$. If in model M_1 state y can be reached from x in k steps, then obviously in M_2 state y can be reached from x in k or fewer steps. Therefore, if P_1 is a regular matrix, then P_2 will also be a regular matrix.

Our aim is to express the group inverse V_1 through the group inverse V_2. We will denote the row-vectors of matrix V_2 as $\mathbf{v}_i$ and the columns as $\mathbf{v}^j$, i.e., $\mathbf{v}_i = (v(i, \cdot))$, $\mathbf{v}^j = (v(\cdot, j))$. Let us denote vector $\boldsymbol{\pi}$=$\{\pi_2(2), \pi_2(3), \ldots, \pi_2(n)\}$.

For the model M_1, let the constants $s = s_1$ and $\alpha = \alpha_1$ be given by (2.5); also let vectors $\mathbf{p}$ and $\mathbf{q}$ be given by (2.6). The scalar product of vectors $\mathbf{x}$ and $\mathbf{y}$ will be denoted by $\mathbf{x}^T\mathbf{y}$.

We define the (column) vectors $\mathbf{r}, \mathbf{t}, \mathbf{c}$ and the constant c by

$$\mathbf{r} = \alpha V_2 \mathbf{q}, \tag{3.5}$$

$$\mathbf{t}^T = \frac{(1-\alpha)}{s} \mathbf{p}^T V_2, \tag{3.6}$$

$$c = \frac{(1-\alpha)}{s}(\alpha + \mathbf{p}^T\mathbf{r}), \qquad \text{and} \qquad \mathbf{c} = c\boldsymbol{\pi}_2 - \mathbf{t}. \tag{3.7}$$

It is clear from (3.4) and (3.6) that $\mathbf{t}^T\mathbf{e} = 0$; hence (3.7) implies that $\mathbf{c}^T\mathbf{e} = c$.

Our main result is the following.

THEOREM 3.2. *Let $M_2 = (X_1 \backslash \{z\}, P_2)$ be the reduced model of $M_1 = (X_1, P_1)$,as defined in section 2, with P_2 related to P_1 by (2.2). Denote their associated group inverse matrices by V_1, V_2, and their invariant vectors by $\boldsymbol{\pi}_1$,$\boldsymbol{\pi}_2$. If states are indexed so that $z = 1$, vectors $\mathbf{r}$ and $\mathbf{c}$ and constant c are as defined in (3.5) through (3.7), $\alpha = \alpha_1$ is given by (2.5), and $\mathbf{q}$ is given by (2.6), then the group inverse matrix V_1 can be described in terms of four matrix blocks as follows:*

$$V_1 = \begin{bmatrix} v_{11} & \mathbf{v}_{12}^T \\ \mathbf{v}_{21} & W_1 \end{bmatrix} = \begin{bmatrix} \frac{\alpha}{1-\alpha} c & \frac{-\alpha}{1-\alpha}\mathbf{c}^T \\ \mathbf{r} - c\mathbf{e} & V_2 + U \end{bmatrix}, \quad \text{where } U = -\mathbf{r}\boldsymbol{\pi}_2^T + \mathbf{e}\mathbf{c}^T. \tag{3.8}$$

Proof. It is possible to prove the result by checking that the matrix V_1 given in (3.8) satisfies Theorem 3.1. Instead, our proof will show explicitly how we arrive at each block of (3.8). To simplify our notation we will omit the index "2" in all references to the matrices P_2, $A_2 = \mathbf{e}\boldsymbol{\pi}_2^T, V_2$, identity matrix $I_2 \equiv I$, and to the invariant measure $\boldsymbol{\pi}_2$=$\boldsymbol{\pi}$.

The first step is to express $v_1(i, j)$ for $i, j \neq z$ in terms of the elements of matrix $V_2 \equiv V$. Formula (3.2) (the first of two equalities), applied to V_1, implies (using the Kronecker symbol $\delta(i, j)$)

$$v_1(i, j) = \delta(i, j) - \pi_1(j) + p_1(i, z)v_1(z, j) + \sum_{k \neq z} p_1(i, k)v_1(k, j). \tag{3.9}$$

Recall that $z \equiv 1$. When $i = z$, $j \neq z$, formula (3.9), using $1 - p_1(z, z) = s$ becomes

$$sv_1(z, j) = -\pi_1(j) + \sum_{k \neq z} p_1(z, k)v_1(k, j). \tag{3.10}$$

Substituting the expression for $v_1(z, j)$ from (3.10) into (3.9) we obtain for $i \geq 2$, $j \geq 2$,

$$v_1(i, j) = \delta(i, j) - [1 + p_1(i, z)/s]\pi_1(j) \tag{3.11}$$

$$+\sum_{k\neq z}[p_1(i,k)+p_1(i,z)p_1(z,k)/s]v_1(k,j).$$

Now, replacing $\pi_1(j)$ by $\alpha\pi(j)$ (formula (2.3)) and the expression in the brackets in the sum by $p(i,k)$ (formula (2.2) with $p_2\equiv p$), we can rewrite formula (3.11) for $i\geq 2$, $j\geq 2$ in matrix form (the restriction of V_1 for i, $j\neq z$ is denoted by W_1)

$$W_1 = I - TA + PW_1, \tag{3.12}$$

where T is a diagonal matrix with elements equal to $\alpha(1+p_1(i,z)/s)$, $i\geq 2$.

Recall (see (3.8)) that $U=W_1-V_2\equiv W_1-V$. Then subtracting $V=(I-A)+PV$ (formula (3.2) for $V=V_2$) from (3.12) we obtain the equation for U

$$U = GA + PU, \tag{3.13}$$

where $G=I-T$ is a diagonal matrix with elements (using $\alpha=1-\pi_1(z)$)

$$1-\alpha(1+p_1(i,z)/s) = \pi_1(z)-\alpha p_1(i,z)/s, \quad i\geq 2. \tag{3.14}$$

LEMMA 3.3. *Any solution of* (3.13) *has the form*

$$U = VGA + C, \tag{3.15}$$

where the jth column of C is $c(j)\mathbf{e}$, $c(j)$ a constant.

Proof of Lemma 1. As we mentioned earlier, it is well known that any matrix solution of $X=PX$ for a regular P is a matrix C with constant columns. Hence any solution of (3.13) is a particular solution of this equation plus such a matrix C. Therefore we need to show only that the matrix VGA is a solution of (3.13). By formula (3.2) for model M_2, ($P\equiv P_2$, $A\equiv A_2$), we have $VGA=GA-AGA+PVGA$. Let us show that $AGA=0$. It is easy to see this is equivalent to $\boldsymbol{\pi}^T\mathbf{g}=0$, where $\mathbf{g}$ is a vector of diagonal elements of G, i.e., given by (3.14). Using the equality $\sum_i\pi(i)=1$ and formula (2.4), we obtain

$$\boldsymbol{\pi}^{\boldsymbol{T}}\mathbf{g} = \sum_i \pi(i)(\pi_1(z)-\alpha p_1(i,z)/s) = \pi_1(z)-\alpha\sum_i\pi(i)p_1(i,z)/s = 0,$$

which establishes Lemma 1.

Thus we have calculated $W_1=V+U$ (the restriction of V_1 for i, $j\neq z$) up to unknown constants $c(j)$. To finish the calculation of V_1 we need to show that unknown constants $c(j)$ (matrix C in (3.15)) coincide with the components of vector $\mathbf{c}$ defined in (3.7), and to provide formulas for the first row and the first column of V_1.

First, we can simplify VGA further, noticing that by (3.14) $G=\pi_1(z)I+D$, where $d(i,z)=-\alpha p_1(i,z)/s, i\geq 2$, and $VIA=0$ by formula (3.4). Therefore $U=VDA+C$, and we have $W_1=V(I+DA)+C$. Thus, using $d(i,z)$ and the definition of vector $\mathbf{r}$ in (3.5), we obtain for k, $j\geq 2$

$$v_1(k,j) = v(k,j)-\alpha\pi(j)\sum_i v(k,i)p_1(i,z)/s + c(j) \tag{3.16}$$

$$= v(k,j)-\pi(j)r(k)+c(j), \tag{3.17}$$

which verifies that $W_1=V+U$, as claimed in (3.8).

Using (3.16), we can rewrite the jth column of (3.3) for $V = V_1$ as

$$\pi_1(z)v_1(z,j) + \sum_k \pi_1(k)[(k,j) - \pi(j)r(k) + c(j)] = 0, \quad j \geq 2.$$

Using the equalities $\pi_1(k) = \alpha\pi(k)$, $\sum_k \pi(k)v(k,j) = 0$ (i.e., $\boldsymbol{\pi}^T V = 0$; see (3.3) for $V = V_2$), $\sum_k \pi_1(k) = 1 - \pi_1(z) = \alpha$, and the equality $\boldsymbol{\pi}^T\mathbf{r} = \alpha\ \boldsymbol{\pi}^T(V\mathbf{q}) = \alpha\ (\boldsymbol{\pi}^T V)\mathbf{q} = 0$, we obtain

$$(1-\alpha)v_1(z,j) + \alpha c(j) = 0, \quad j \geq 2, \tag{3.18}$$

which is equivalent to $\mathbf{v}_{12}^T = -\mathbf{c}^T \frac{\alpha}{1-\alpha}$ given in (3.8).

It remains to be verified that $c(j)$ satisfy (3.7). Substituting $v_1(k,j)$ from (3.16) into (3.10) and using the equalities $\sum_k p_1(z,k) = 1 - p_1(z,z) = s$, $\pi_1(j) = \alpha\pi(j)$, the definition of vector $\mathbf{t}$ (3.6), and the definition of constant c in (3.7), we can rewrite (3.10) as

$$v_1(z,j) - c(j) = (-c\pi(j) + t(j))/(1-\alpha), \quad j \geq 2. \tag{3.19}$$

Using (3.18) to replace $v_1(z,j)$ in (3.19), we obtain (3.7).

Now the entries of $\mathbf{v}_{21}$ in (3.8) can be found using equality (3.4) as follows. For $i \geq 2$, using $W_1 = V - \mathbf{r}\boldsymbol{\pi}_2^T + \mathbf{e}\mathbf{c}^T$ from (3.8) we obtain

$$v_1(i,z) = -\sum_j v_1(i,j) = -\sum_j v(i,j) + r(i)\sum_j \pi(j) - \sum_j c(j).$$

The first sum of the rightmost expression is equal to zero by (3.4) for $V = V_2$. Using $\sum \pi_j = 1$, $\sum c(j) = c$, we obtain $\mathbf{v}_{21} = \mathbf{r} - c\mathbf{e}$. The 1×1 block v_{11} is obtained similarly, using (the now-established representation of $\mathbf{v}_{21}$ in) (3.8) and (3.4) for $V = V_1$. ∎

Remark. Note that only the first two expressions in (3.2) have been used in this section and that an additional opportunity to check (increase) the accuracy of computations of $V = V_1$ is provided by considering the rightmost expression in that equation.

4. The REFUND algorithm and numerical tests.

4.1. The REFUND algorithm. The results of the previous section lead to a recursive algorithm, REFUND, to calculate the group inverse matrix $V = V_1$ of $n \times n$ stochastic matrix $P = P_1$. The reduction stage implicitly produces a finite sequence $M = M_1, M_2, \ldots, M_n$ of models, where in each model $M_k = (X_k, P_k)$, X_k is a state space, and P_k is a stochastic matrix. A single step of reduction, in which P_2 (and $\bar{P}_2$) are calculated from P_1, was depicted in (2.6), which also describes (with obvious changes of index) any such step yielding P_{k+1} from P_k. During computation, the array in which P was originally stored is altered by repeated application of (2.6): at each step P_k is replaced by $\bar{P}_{k+1}$. Thus the reduction stage duplicates (up to minor variation in the order in which states are eliminated) that of the GTH/S algorithm and is shared by all of the SR algorithms described in section 2. The output of the reduction stage is the matrix/array $\bar{P}_n = \bar{P}$, containing (for $k = 1, \ldots, n-1$) the vectors $\mathbf{p}_k^T$ and $\mathbf{q}_k$ given (without subscripts) by (2.6).

REFUND's recovery stage is initialized with information from the smallest (one state) model M_n: the stochastic matrix $P_n = [1]$, invariant distribution $\boldsymbol{\pi}_n = [1]$, and group inverse matrix $V_n = [0]$. Each recovery step ($k = n-1, \ldots, 1$) begins with

the pair ($\boldsymbol{\pi}_{k+1}$, V_{k+1}) and calculates ($\boldsymbol{\pi}_k$,V_k). The vector $\boldsymbol{\pi}_k$ is obtained (with the appropriate change of index) from (2.7); then the matrix V_k is calculated (with similar reindexing) from (3.8), using information from (3.5) through (3.7).

Remark. As was mentioned above, to avoid unnecessary subtractive cancellation, the GTH/S algorithm calculates s_k ($= s$) (see the formula immediately above (2.6)) as the sum $\sum_{j\neq z} p_k(z,j)$. Similarly, REFUND calculates the scalars $\frac{\alpha}{1-\alpha}$ (in (3.8)) and $\frac{1-\alpha}{s}$ (in (3.5) and (3.7)) as $\frac{1}{\boldsymbol{\pi}_2^T\mathbf{q}}$ and $\frac{\boldsymbol{\pi}_2^T\mathbf{q}}{s\left(1+\boldsymbol{\pi}_2^T\mathbf{q}\right)}$, respectively. Thus in all division operations, divisors are calculated without subtraction from the output of the GTH/S algorithm, which contains no subtraction at all. Since the group inverse matrix and fundamental matrix generally contain both positive and negative elements, any algorithm calculating either must contain subtraction. Whether or not some of these subtractions involve numbers that are "nearly equal" (thus reducing the number of significant digits in some element) depends on the structure of the particular matrix given as input. Note that the explicit formula on which REFUND is based provides an opportunity to analyze this question directly. We are going to address the application of REFUND to the NCD case in a separate paper.

Operation counts. The number of arithmetic operations encountered by the REFUND algorithm is an $O(n^3)$ function f, where the stochastic matrix $P = P_1$ is $n \times n$. Taken together, the calculations of the reduction stage, and those entailed in the recovery of $\boldsymbol{\pi}$, duplicate the GTH/S algorithm, which is $\Theta(\frac{2}{3}n^3)$. The additional operations required to recover V contribute $\Theta(\frac{7}{3}n^3)$ to the dominant term of f, which is affected only by the products $V_2\mathbf{q}$ (in (3.5)), $\mathbf{p}^TV$ (in (3.6)), $\mathbf{r}\boldsymbol{\pi}_2^T$ (and not $\mathbf{e}\mathbf{c}^T$) in the lower right block of (3.8), and by the matrix addition and subtraction in that block.

4.1.1. Example. We include the calculations for the well-known "Land of Oz" example from Kemeny and Snell [10].

Reduction. Initialize: $\bar{P}_0 = P\ (= P_1)$.

$$\bar{P}_0 = \begin{bmatrix} 1/2 & 1/4 & 1/4 \\ 1/2 & 0 & 1/2 \\ 1/4 & 1/4 & 1/2 \end{bmatrix} \rightarrow \bar{P}_1 = \begin{bmatrix} 1/2 & 1/4 & 1/4 \\ 1 & 1/4 & 3/4 \\ 1/2 & 3/8 & 5/8 \end{bmatrix} \rightarrow \bar{P}_2 = \begin{bmatrix} 1/2 & 1/4 & 1/4 \\ 1 & 1/4 & 3/4 \\ 1/2 & 1/2 & 1 \end{bmatrix}.$$

Recovery. Initialize: $\boldsymbol{\pi}_3 = \begin{bmatrix} 1 \end{bmatrix}, V_3 = \begin{bmatrix} 0 \end{bmatrix}$.

Step 1:

$\boldsymbol{\pi}_3^T\mathbf{q}_2 = \begin{bmatrix} 1 \end{bmatrix}\begin{bmatrix} 1/2 \end{bmatrix} = 1/2$, $\alpha_2 = \frac{1}{1+\boldsymbol{\pi}_3^T\mathbf{q}_2} = 2/3$, $\boldsymbol{\pi}_2 = \alpha_2 \begin{bmatrix} \boldsymbol{\pi}_3^T\mathbf{q}_2 \\ \boldsymbol{\pi}_3 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 1 \\ 2 \end{bmatrix}$;

$\mathbf{r}_2 = \alpha_2 V_3\mathbf{q}_2 = \begin{bmatrix} 0 \end{bmatrix}$; $\mathbf{t}_2 = \left(\frac{\alpha_2\boldsymbol{\pi}_3^T\mathbf{q}_2}{s_2}\right)\mathbf{p}_2^TV_3 = \begin{bmatrix} 0 \end{bmatrix}$,

$c_2 = \left(\frac{\alpha_2\boldsymbol{\pi}_3^T\mathbf{q}_2}{s_2}\right)(\alpha_2 + \mathbf{p}_2^T\mathbf{r}_2) = 8/27$, $\mathbf{c}_2 = c_2\boldsymbol{\pi}_3 - \mathbf{t}_2^T = \frac{1}{27}\begin{bmatrix} 8 \end{bmatrix}$,

$$V_2 = \begin{bmatrix} \frac{\alpha_2}{1-\alpha_2}c_2 & \frac{-\alpha_2}{1-\alpha_2}\mathbf{c}_2^T \\ \mathbf{r}_2 - c_2\mathbf{e} & V_3 - \mathbf{r}_2\boldsymbol{\pi}_3^T + \mathbf{e}\mathbf{c}_2^T \end{bmatrix} = \frac{1}{27}\begin{bmatrix} 16 & -16 \\ -8 & 8 \end{bmatrix}.$$

Step 2:

$$\boldsymbol{\pi}_2^T\mathbf{q}_1 = 2/3,\ \alpha_1 = \tfrac{1}{1+\boldsymbol{\pi}_2^T\mathbf{q}_1} = 3/5,\ \boldsymbol{\pi}_1 = \alpha_1 \begin{bmatrix} \boldsymbol{\pi}_2^T\mathbf{q}_1 \\ \boldsymbol{\pi}_2 \end{bmatrix} = \tfrac{1}{5}\begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix};$$

$$\mathbf{r}_1 = \alpha_1 V_2\mathbf{q}_1 = \tfrac{1}{45}\begin{bmatrix} 8 \\ -4 \end{bmatrix},\ \mathbf{t}_1 = \left(\tfrac{\alpha_1\boldsymbol{\pi}_2^T\mathbf{q}_1}{s_1}\right)\mathbf{p}_1^T V_2 = \tfrac{1}{135}\begin{bmatrix} 8 & -8 \end{bmatrix},$$

$$c_1 = \left(\tfrac{\alpha_1\boldsymbol{\pi}_2^T\mathbf{q}_1}{s_1}\right)\left(\alpha_1 + \mathbf{p}_1^T\mathbf{r}_1\right) = 112/225,\ \mathbf{c}_1 = c_1\boldsymbol{\pi}_2 - \mathbf{t}_1^T = \tfrac{1}{225}\begin{bmatrix} 24 \\ 88 \end{bmatrix},$$

$$V_1 = \begin{bmatrix} \frac{\alpha_1}{1-\alpha_1}c_1 & \frac{-\alpha_1}{1-\alpha_1}\mathbf{c}_1^T \\ \mathbf{r}_1 - c_1\mathbf{e} & V_2 - \mathbf{r}_1\boldsymbol{\pi}_2^T + \mathbf{e}\mathbf{c}_1^T \end{bmatrix} = \tfrac{1}{75}\begin{bmatrix} 56 & -12 & -44 \\ -24 & 48 & -24 \\ -44 & -12 & 56 \end{bmatrix}.$$

4.2. Numerical tests.

4.2.1. Implementation. The REFUND algorithm was coded and run in MATLAB, using IEEE arithmetic with 16 decimal digit working precision. Pivoting is easily incorporated, but tabulated results are for tests runs with pivoting disabled.

4.2.2. Measures of accuracy. We define measures of residual error for all conditions required in Theorem 3.1. So that these measures will continue to be appropriate during our later comparison of REFUND to FUND, we first let $D = \mathrm{diag}(P\mathbf{e})$ be the diagonal matrix whose nonzero entries are the rowsums of P; thus (as now) when P is stochastic, $D = I$. Define (block) matrices $\Phi = \begin{bmatrix} (D-P)^T & \boldsymbol{\pi} \end{bmatrix}^T$ and $S = S(\beta) = \begin{bmatrix} (I-A)^T & \beta\boldsymbol{\pi} \end{bmatrix}^T$. When $\beta = 0$, V is the unique solution to

$$\Phi X = S(\beta), \tag{4.1}$$

which combines the first equality of (3.2) with (3.3). Now let $\tilde{V}$ denote the calculated value for matrix V; let $H = \Phi\tilde{V} - S(0)$; and (for each $j = 1, \ldots, n$) $\mathbf{h}_j$ will denote the jth column of H. Let $W = \tilde{V}P - P\tilde{V}$, with jth column $\mathbf{w}_j$; and let $\mathbf{e}$ denote a vector of ones. We define measures $\delta_1, \delta_2, \delta_3$ by

$$\delta_1 = \max_j\{||\mathbf{h}_j||_2\}, \qquad \delta_2 = ||\tilde{V}\mathbf{e}||_\infty, \qquad \delta_3 = \max_j\{||\mathbf{w}_j||_2\}. \tag{4.2}$$

Condition number. The $n\times n$ matrix $(I-P)$ in (3.2) has rank $n-1$, so its matrix condition number is undefined. To compare relative error to problem condition, we use

$$\kappa = \sigma_{\max}/\sigma_{\min}, \tag{4.3}$$

where $\sigma_{\max}$ and $\sigma_{\min}$ are, respectively, the largest and smallest nonzero singular values of $(I-P)$. Sonin and Thornton [29] remark that κ, rather than the analogous (and larger) ratio of singular values of Φ in (4.1), is a true relative condition number for the calculation of V. The ratio κ was used earlier by Heyman [5] and by Heyman and O'Leary [8] for the problem of calculating the fundamental matrix.

4.2.3. Test problems. Two sets of test problems were used. The first problem set consists of seven problems and comes from Harrod and Plemmons [4]. Although the stochastic matrices in this set are not large, the set contains problems that are numerically difficult and has provided test problems used in Heyman and Reeves [7], Heyman and O'Leary [6], two examples in Heyman [5], and Sonin and Thornton [29]. Three of these problems have condition numbers of more than 10^5, and one exceeds

10^7. The last four problems involve NCD chains. Such chains have subsets of states between which transitions occur only rarely and are known to be ill-conditioned. They are discussed in several works of Stewart, for example, in [30] and [32]. The test matrices are available in any of the foregoing sources, and we omit them here.

The problems in the second set were used by Heyman and O'Leary [8] to test the stabilized version of FUND and concern continuous time Markov chains. The matrix entries represent transition rates, not probabilities, so these matrices, as given in [8], are not stochastic. State spaces for these chains have the form $\{0, 1, 2, \ldots, n\}$, so the resulting matrices have dimension $(n+1)\times(n+1)$. All transition rates are zero except $p_{i,i+1} = \lambda$ (for $0 \le i < n$) and $p_{i,i-1} = i$ (for $0 < i \le n$). As Heyman and O'Leary do in [8], we solve these problems for $n = 5, 10, 15, \ldots, 50$, and choose $\lambda = n$ in each problem.

4.2.4. Test results and interpretation. For each test problem of the first set Sonin and Thornton [29] tabulate δ_1, δ_2, δ_3 (given by (4.2)), and scaled measures $\delta_1/(\kappa\varepsilon)$, $\delta_2/(\kappa\varepsilon)$, $\delta_3/(\kappa\varepsilon)$, where κ is the condition number given by (4.3), and $\varepsilon = 2.22 \times 10^{-16}$ (reported by MATLAB) is the smallest positive floating point number γ such that the floating point result $1 + \gamma \ne 1$. For all problems of the first set, $\delta_1/(\kappa\varepsilon) \le 0.48$, $\delta_2/(\kappa\varepsilon) \le 0.65$, $\delta_3/(\kappa\varepsilon) \le 0.74$, indicating that all requirements of Theorem 3.1 were satisfied as well as can be expected, given problem conditioning. Now δ_1 corresponds to conditions which were used in calculation. That both $\delta_2/(\kappa\varepsilon)$ and $\delta_3/(\kappa\varepsilon)$ were consistently small is significant, since REFUND does not make explicit use of either the rowsum conditions (3.4) or of the commutativity conditions (second equality of (3.2)).

Similar results are also tabulated in [29] for a less well-known set of ten problems. For all of these $\delta_1/(\kappa\varepsilon) \le 0.95$, $\delta_2/(\kappa\varepsilon) \le 2.88$, $\delta_3/(\kappa\varepsilon) \le 1.52$, still indicating close agreement with all requirements.

4.3. Comparison with the FUND algorithms. The FUND algorithms described in section 2.2 calculate the fundamental matrix Z defined in (1.3). Since both algorithms require that π be calculated by the GTH/S algorithm, either V or Z can be obtained accurately from the other by (1.4) at a cost that does not affect the dominant term of the workload.

Speed. In addition to the $\Theta(\frac{2}{3}n^3)$ operations required by reduction, REFUND's $\Theta(\frac{7}{3}n^3)$ operation count is slightly smaller than the comparable figure for FUND, $\Theta(\frac{8}{3}n^3)$.

Much more substantial savings in time are possible in some cases: because REFUND is recursive while FUND is not, REFUND can reduce model (X, P) to any submodel (X_k, P_k) for which $\boldsymbol{\pi}_k$ and V_k are available and begin recovery immediately.

Accuracy. To make a direct comparison between REFUND's accuracy and that of the stabilized (1998) FUND, Sonin and Thornton [29] tested REFUND using the second problem set, which was used by Heyman and O'Leary in [8]. The measures of accuracy that were defined in (4.2) remain appropriate, since V remains the unique solution to (4.1) for $\beta = 0$. But for $\beta = 1$, (4.1) subsumes (2.8) and (2.9), and Z becomes the unique solution. We let $\tilde{Z}$ denote the matrix computed by the stabilized FUND. For each test problem, [29] tabulates δ_1, measured for REFUND, beside a comparable measure for FUND obtained from [8]. Their measure $r_{imp} = r_{improved}$ is the norm (2-norm assumed) of the residual error in the last column $\tilde{\mathbf{z}}_n$ of Z relative to the first equality of (3.2) only: $r_{imp} = \|\mathbf{e}_n - \pi_n\mathbf{e} - (D - P)\tilde{\mathbf{z}}_n\|$.

Since δ_1 measures residual errors in all columns and includes both conditions (3.2) and (3.3), it is a (slightly) more sensitive measure of error than r_{imp} is: If for the

same matrix P, $\tilde{V}$ is calculated by REFUND and δ_1 is calculated from $\tilde{V}$ as discussed, and similarly $\tilde{Z}$ is calculated by FUND and r_{imp} calculated from $\tilde{Z}$, and the residual errors are identical (i.e., $\Phi\tilde{V} - S(0) = \Phi\tilde{Z} - S(1)$), then $\delta_1 \geq r_{imp}$. But for every test problem, $\delta_1 < r_{imp}$; thus (at least for this problem set) REFUND appears to be the more accurate algorithm. Also, the results tabulated for REFUND were obtained without pivoting, while those tabulated for FUND come from the stabilized version, which pivots in order to achieve stability.

Structure. Because REFUND is a recursive algorithm like most other SR algorithms, which also share the reduction stage of the GTH/S algorithm, it can be readily implemented along with other SR algorithms in computer code that produces a variety of information in one run. When calculating several characteristics of a system together in such a simultaneous recursion, it becomes possible to exploit any known relationships among them, either to save time, to improve accuracy, or to derive further information about the system under study. Of course REFUND produces results for submodels and allows a user to reduce to and restart from any solved submodel. Also, the explicit formula on which REFUND is based provides a new means by which to analyze and compare various cases, e.g., sparse matrices, NCD chains, or decompositions required by parallel implementations.

REFERENCES

[1] M. I. Freidlin and A. D. Wentzell, *Perturbations of Stochastic Dynamic Systems*, Springer-Verlag, New York, 1984.

[2] W. K. Grassmann, M. I. Taksar, and D. P. Heyman, *Regenerative analysis and steady-state distributions for Markov chains*, Oper. Res., 33 (1985), pp. 1107–1116.

[3] W. K. Grassmann, *Means and variances in Markov reward systems*, in Linear Algebra, Markov Chains, and Queueing Models, C. D. Meyer and R. J. Plemmons, eds., IMA Vol. Math. Appl. 48, Springer-Verlag, New York, 1993, pp. 193–204.

[4] W. J. Harrod and R. J. Plemmons, *Comparison of some direct methods for computing stationary distributions of Markov chains*, SIAM J. Sci. Statist. Comput., 5 (1984), pp. 453–469.

[5] D. P. Heyman, *Accurate computation of the fundamental matrix of a Markov chain*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 954–963.

[6] D. P. Heyman and D. P. O'Leary, *What is fundamental for Markov chains: First passage times, fundamental matrices, and group generalized inverses*, in Computations with Markov Chains, W. J. Stewart, ed., Kluwer, Boston, 1995, pp. 151–159.

[7] D. P. Heyman and A. Reeves, *Numerical solution of linear equations arising in Markov chain models*, ORSA J. Comput., 1 (1989), pp. 52–60.

[8] D. P. Heyman and D. P. O'Leary, *Overcoming instability in computing the fundamental matrix for a Markov chain*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 534–540.

[9] D. L. Isaacson and R. W. Madsen, *Markov Chains: Theory and Applications*, Wiley, New York, 1976.

[10] J. G. Kemeny and J. L. Snell, *Finite Markov Chains*, Van Nostrand Reinhold, Princeton, 1960.

[11] J. G. Kemeny, J. L. Snell, and A. V. Knapp, *Denumerable Markov Chains*, Springer-Verlag, New York, 1976.

[12] J. Kohlas, *Numerical computation of mean passage times and absorption probabilities in Markov and semi-Markov models*, Z. Oper. Res., 30 (1986), pp. A197–A207.

[13] R. Lal and U. N. Bhat, *Reduced systems algorithms for Markov chains*, Management Sci., 34 (1988), pp. 1202–1220.

[14] B. F. Lamond and M. L. Puterman, *Generalized inverses in discrete time Markov decision processes,* SIAM J. Matrix Anal. Appl., 10 (1989), pp. 118–134.

[15] C. D. Meyer, Jr., *The role of the group generalized inverse in the theory of finite Markov chains,* SIAM Rev., 17 (1975), pp. 443–464.

[16] C. D. Meyer and R. J. Plemmons, eds., *Linear Algebra, Markov Chains, and Queueing Models*, IMA Vol. Math. Appl. 48, Springer, New York, 1993.

[17] C. O. O'Cinneide, *Entrywise perturbation theory and error analysis for Markov chains*, Numer. Math., 65 (1993), pp. 109–120.
[18] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, Wiley, New York, 1994.
[19] T. J. Sheskin, *A Markov partitioning algorithm for computing steady-state probabilities*, Oper. Res., 33 (1985), pp. 228–235.
[20] T. J. Sheskin, *Matrix inversion by augmentation and reduction*, Internat. J. Math. Ed. Sci. Tech., 22 (1991), pp. 103–110.
[21] T. J. Sheskin, *Computing absorption probabilities for a Markov chain*, Internat. J. Math. Ed. Sci. Tech., 22 (1991), pp. 799–805.
[22] T. J. Sheskin, *Computing the fundamental matrix for a reducible Markov chain*, Math. Mag., 68 (1995), pp. 393–398.
[23] T. J. Sheskin, *Computing mean first passage times for a Markov chain*, Internat. J. Math. Ed. Sci. Tech., 26 (1995), pp. 729–735.
[24] T. J. Sheskin, *A partitioning algorithm for solving systems of linear equations*, Internat. J. Math. Ed. Sci. Tech., 27 (1996), pp. 641–648.
[25] I. M. Sonin, *Two simple theorems in the problems of optimal stopping*, in Proceedings of the Eighth INFORMS Applied Probability Conference, Atlanta, 1995, INFORMS, Linthicum, MD, 1995.
[26] I. M. Sonin, *The state reduction and related algorithms and their applications to the study of Markov chains, graph theory and the optimal stopping problem*, Adv. Math., 145 (1999), pp. 159–188.
[27] I. M. Sonin, *The Elimination Algorithm for the Problem of Optimal Stopping,* Math. Methods Oper. Res, 49 (1999), pp. 111–123.
[28] I. M. Sonin and J. R. Thornton, *The elimination algorithm and its applications*, in Proceedings of the Ninth INFORMS Applied Probability Conference, Boston, 1997, INFORMS, Linthicum, MD, 1997.
[29] I. M. Sonin and J. R. Thornton, *Computational properties of algorithm REFUND for the fundamental/group inverse matrix of a Markov chain,* in Numerical Solution of Markov Chains, B. Plateau, W. J. Stewart, and M. Silva, eds., Proceedings of the Third International Workshop, Prensas Universitarias de Zaragoza, Zaragoza, Spain, 1999, pp. 131–148.
[30] J. R. Koury, D. F. McAllister, and W. J. Stewart, *Iterative methods for computing stationary distributions of nearly completely decomposable Markov chains*, SIAM J. Algebraic Discrete Methods, 5 (1984), pp. 164–186.
[31] W. J. Stewart, ed., *Computations with Markov Chains*, Kluwer, Boston, 1995.
[32] W. J. Stewart, *Numerical methods for computing stationary distributions of finite irreducible Markov Chains*, in Advances in Computational Probability, W. Grassmann, ed., Kluwer, Boston, to appear.