On Some Methods for Entropy Maximization and Matrix Scaling

Tommy Elfving
National Defence Research Institute
Box 1165
S-581 11 Linköping, Sweden

Submitted by Robert J. Plemmons

ABSTRACT

We describe and survey in this paper iterative algorithms for solving the discrete maximum entropy problem with linear equality constraints. This problem has applications e.g. in image reconstruction from projections, transportation planning, and matrix scaling. In particular we study local convergence and asymptotic rate of convergence as a function of the iteration parameter. For the trip distribution problem in transportation planning and the equivalent problem of scaling a positive matrix to achieve *a priori* given row and column sums, it is shown how the iteration parameters can be chosen in an optimal way. We also consider the related problem of finding a matrix X, diagonally similar to a given matrix, such that corresponding row and column norms in X are all equal. Reports of some numerical tests are given.

1. INTRODUCTION

In this paper we will study the behavior of particular methods for solving the following maximum entropy problem with linear equality constraints:

$$\min_x \sum_{j=1}^{n} x_j \ln\frac{x_j}{u_j} \qquad \text{subject to} \quad Ax=b, \quad x\geqslant 0, \tag{1.1}$$

where u is a given nonnegative vector and $A=\{a_{ij}\}$ an $m\times n$ matrix, with all rows $a_i^T\neq 0$. We state for clarity the following lemma, which is easily verified.

LEMMA 1. *Assume that the set $F=\{x\,|\,Ax=b\cap x\geqslant 0\}$ is nonempty and that $u>0$. Then* (1.1) *has a unique solution, which is positive if F contains a positive element.*

52 Vanderbilt Ave., New York, NY 10017 0024-3795/80/060321+19$01.75

REMARK. If some components of u equal zero, then we will always put the corresponding elements of x to zero. We then consider a reduced problem with a matrix $\bar{A}$ obtained from A by deleting all columns corresponding to $u_j = x_j = 0$. Similarily it is always possible, at least in theory, to reduce the problem so that the nonempty set F has a positive element. We will assume throughout this paper that this reduction has taken place. This assumption will allow us to apply the usual Kuhn-Tucker theorem to (1.1).

The need to solve the problem (1.1) may arise when a linear model is described in a statistical environment (usually then $\Sigma x_j = T$, with T an *a priori* given constant) and a solution which is most objective with respect to the missing information is sought; cf. [22], where the problem is derived from a minimum information principle. We further refer to the work by Frieden [9] for a derivation of (1.1) for use in image restoration and image reconstruction from projections.

In Sec. 2 iterative methods for solving the problem (1.1) are described. The algorithms utilize the special structure of the Kuhn-Tucker conditions for the problem. We consider some specific schemes and also relate these with methods proposed in the literature. In the following section local convergence and asymptotic rate of convergence are studied. It is shown that the methods exhibit the same rate of convergence as the linear SOR and Jacobi methods [17] applied to a certain linear system. In particular we derive conditions on the iteration parameters which will insure local convergence.

In some applications, e.g. in the unweighted model used in picture reconstruction [10] and in the trip distribution problem in traffic planning [20], the matrix elements a_{ij} equal 0 or 1. We show in Sec. 4 how the algorithms simplify and unify in this case. In Sec. 5 the trip distribution problem is presented. This problem is identical with scaling a positive matrix so that the row and column sums take on prescribed values [21]. We give for this case the optimal values of the iteration parameters. The side conditions in this application have one redundant row. We shortly discuss how the rate of convergence is affected by deleting this redundant information.

In Sec. 6 we consider the related problem of finding a matrix X, diagonally similar to a given matrix, such that X is balanced, i.e., corresponding row and column norms in X are all equal. We show that this problem can be written in the form (1.1) for appropriate choices of A and u. In the last section we present some numerical results.

2. THE ALGORITHMS

The Kuhn-Tucker conditions for the problem $\min g(x)$ s.t. $Ax = b$ are given by (i) $\nabla g(x) = A^T\beta$ and (ii) $b = Ax$. For the problem (1.1) they become

$$x_j = u_j \exp\left[(A^T\beta)_j - 1 \right], \quad (A^T\beta)_j = \sum_{i=1}^{m} a_{ij}\beta_i, \qquad j = 1,2,\dots,n. \tag{2.1a}$$

$$b = Ax. \tag{2.1b}$$

Note that the positivity constraint on x is automatically satisfied. We remark that (2.1) provide both necessary and sufficient conditions for $x^* > 0$ to be the unique solution of (1.1). However if $\text{rank}(A) < m$, then the corresponding dual vector is not unique. In fact, as seen from (2.1a), if β is a solution, then $\beta + z$, where z is any vector in the nullspace of A^T, is also a solution. By combining (2.1a) and (2.1b) it follows that

$$h_i(\beta) = \sum_{j=1}^{n} a_{ij} u_j \exp\left[(A^T\beta)_j - 1 \right] - b_i = 0, \qquad i = 1,2,\dots,m. \tag{2.2}$$

The methods considered in this paper are based on solving (2.2) for $a\beta$ and retrieving (the unique) x from (2.1a). We will consider algorithms of the following form:

$$\beta^{k+1} = \beta^k + \omega\lambda^k, \qquad k = 0,1,2,\dots, \tag{2.3}$$

where ω is an iteration parameter. By defining [cf. (2.1a)]

$$x_j^k = u_j \exp\left[(A^T\beta^k)_j - 1 \right], \tag{2.4}$$

the corresponding primal iteration scheme becomes

$$x_j^{k+1} = x_j^k \exp\left[\omega (A^T\lambda^k)_j \right], \qquad j = 1,2,\dots,n, \quad k = 0,1,2,\dots. \tag{2.5}$$

The starting vector x^0 is taken as any vector that can be written in the form (2.1a), where β^0 is an arbitrarily vector of length m.

We now first consider a class of methods, which will be referred to as Gauss-Seidel type schemes, where only one component in β^k is changed in every step k:

$$\beta_i^{k+1} = \begin{cases} \beta_i^k + \omega\lambda_i^k, & i = (k \bmod m) + 1, \\ \beta_i^k, & i \neq (k \bmod m) + 1. \end{cases} \tag{2.6}$$

The primal iteration then becomes

$$x_j^{k+1} = x_j^k \exp\left(\omega a_{ij} \lambda_i^k \right), \qquad j = 1,2,\dots,n, \quad i = (k \bmod m) + 1. \tag{2.7}$$

Let a_i^T be the ith row of the matrix A. We first consider the two choices

$$\lambda_i^k = \ln \frac{b_i}{a_i^T x^k} \quad \text{and} \quad \lambda_i^k = b_i - a_i^T x^k, \qquad i = (k \bmod m) + 1. \quad (2.8\text{a,b})$$

Here the first choice is restricted to matrices with nonnegative elements $(a_{ij} \geqslant 0)$. We next describe the nonlinear SOR method [17] applied to the equations (2.2). Then λ_i^k is determined such that $h_i(\beta^k + \lambda^k) = 0$. From (2.2), (2.4), and (2.6) this equation can be written

$$h_i(\beta^k + \lambda^k) = f_i(x^k, \lambda_i^k) = \sum_{j=1}^{n} a_{ij} x_j^k \exp(a_{ij}\lambda_i^k) - b_i = 0. \qquad (2.9)$$

Note that λ_i^k is uniquely determined $(\partial f/\partial\lambda > 0)$ from this equation. If Newton's method is used for solving (2.9), the combined scheme is called the SOR-Newton method.

We now consider a second class of iterative schemes, which will be called Jacobi type methods. Here, in general, all components are changed in the dual iterate β^k in every step. Hence the Jacobi iteration is described by (2.3) and (2.5) respectively. All methods presented so far have their corresponding Jacobi variant. As an example we consider the Jacobi version of (2.8a),

$$\lambda_i^k = \ln \frac{b_i}{a_i^T x^k}, \qquad i = 1, 2, \ldots, m. \qquad (2.10)$$

To clarify the difference between Gauss-Seidel and Jacobi type methods we write below the resulting primal schemes corresponding to (2.8a) and (2.10):

$$x_j^{k+1} = x_j^k \left(\frac{b_i}{a_i^T x^k} \right)^{\omega a_{ij}}, \qquad j = 1, 2, \ldots, n, \quad i = (k \bmod m) + 1$$

$$\text{(Gauss-Seidel)}, \quad (2.11)$$

$$x_j^{k+1} = x_j^k \exp\left[\omega (A^T \lambda^k)_j\right] = x_j^k \prod_{1}^{m} \left(\frac{b_i}{a_i^T x^k} \right)^{\omega a_{ij}}, \qquad j = 1, 2, \ldots, n$$

$$\text{(Jacobi)}, \quad (2.12)$$

Let $X^k = \operatorname{diag}(x_1^k, x_2^k, \ldots, x_n^k)$. Then Newton's method applied to (2.2) can be written

$$(A X^k A^T)\lambda^k = b - A x^k. \qquad (2.13)$$

This method also belongs to the Jacobi type schemes, and the primal iteration is given by (2.5). If the linear SOR method is used for solving the semidefinite system (2.13), the combined algorithm is called the Newton-SOR method. Note the lack of commutativity in terminology: Newton-SOR and SOR-Newton denote two different schemes.

The arithmetical work in the primal iteration for performing a whole cycle (going through all rows of A) is, in general, quite different between Jacobi and Gauss-Seidel methods. By comparing (2.5) and (2.7) it follows that while the Jacobi schemes require n evaluations of the exponential function, the corresponding number for Gauss-Seidel is at least equal to the number of nonzero elements in A. (Note that, from this point of view, the first formula in (2.12) rather than the second should be preferred.) We remark also that, for the case $a_{ij} \in \{-1,0,1\}$, as in the applications discussed later in this paper, no evaluations of the exponential are required (using $\omega = 1$).

We end this section by giving some background to the algorithms just described. The method (2.11) was suggested by Gordon, Bender, and Herman [10] for reconstructing an image from its projections. They called the method MART and conjectured that it converged towards the solution of (1.1). This was later proved by Lent [12] for the case $0 \leqslant a_{ij} \leqslant 1$ and using $\omega = 1$. The main idea in Lent's proof is to consider the related dual problem of (1.1) and then show that the dual objective function is monotonically increasing in β^k. Andersson in [1, Proposition 2.3] uses a similar idea (applying a general theorem by Zangwill [25, p. 91]) to establish convergence conditions for entropy maximization methods. We further remark that the nonlinear SOR method (2.9) [we will in the following—unless the method is explicitly specified, as e.g. (2.11)—refer to each method by its choice of λ^k] equals the convex programming method due to Bregman [3], applied to problem (1.1). Bregman's method is also treated by Censor and Lent [4]. Lamond and Stewart [11] describe the method for use in traffic planning. Newton's method was treated in a paper by Erlander [7]. Eriksson [6] has implemented the method using the conjugate gradient method for solving the linear system (2.13).

3. RATE OF CONVERGENCE

We start this section by giving some basic results on local convergence and rate of convergence for one-step iterations of the form

$$x^{k+1} = G(x^k), \qquad k = 0,1,\ldots, \tag{3.1}$$

where $G: D \subset R^n \to R^n$. This presentation follows closely the one given by

Ortega and Rheinboldt [17, Chapters 9, 10]. By local convergence of (3.1) to a point x^* we mean that the iterates of (3.1) will converge to x^* whenever x^0 is sufficiently close to x^*. This is made more precise by the following concept. Suppose there is an open neighborhood S of x^* such that $S \subset D$ and for any $x^0 \in S$, the iterates $\{x^k\}$ defined by (3.1) all lie in D and converge to x^*. Then x^* is a *point of attraction* of the iteration (3.1). We have the following theorem, [17, Chapter 10]:

Ostrowski Linear Convergence Theorem. *Assume that* $G: D \subset R^n \to R^n$ *has a fixed point* $x^* \in \operatorname{int}(D)$ *and is Frechet-differentiable at* x^*. *If the spectral radius of* $G'(x^*)$ *satisfies* $\rho(G'(x^*)) = \sigma < 1$, *then* x^* *is a point of attraction of the iteration* (3.1). *Further, if the convergence factor* σ *is nonzero, the iteration converges linearly.*

Remark 1. The convergence factor σ is a lower bound for any possible constant γ in the estimate $\|x^{k+1} - x^*\| \leqslant \gamma \|x^k - x^*\|$, $\forall k > k_0$.

Remark 2. The size of the open neighborhood S of x^* for which the starting points allow convergence depends, in general, on the nonlinearity of G in a neighborhood of x^*. In the extreme case of an affine operator, we may conclude convergence from any starting point.

We will now study local convergence and rate of convergence for the methods described in Sec. 2. We start by writing the primal iteration (2.5) in the following form:

$$x^{k+1} = D_{z^k} x^k = g(x^k), \qquad \text{where} \quad z_j^k = \exp\left[\omega (A^T \lambda^k)_j\right], \tag{3.2}$$

where D_z is a diagonal matrix such that the jth diagonal element equals z_j. Let x^* be a fixed point of (3.2), and let z^* be the corresponding value of the vector z. Then obviously $z_j^* = 1$, $j = 1, 2, \ldots, n$. We proceed (suppressing the iteration index k in the formulation and proof to simplify the notation) with

Lemma 2. $g'(x^*) = I + \omega D_{x^*} A^T \Lambda(x^*)$, *with*

$$\Lambda_{ij}(x^*) = \left[\frac{\partial \lambda_i}{\partial x_j}\right]_{x = x^*}.$$

Proof. From (3.2), $g'(x) = D_z + D_x z'$. Further

$$\frac{\partial z_j}{\partial x_r} = \sum_{t=1}^{m} \frac{\partial z_j}{\partial \lambda_t} \frac{\partial \lambda_t}{\partial x_r} = \omega \cdot z_j \left(A^T \Lambda_{\odot r}\right)_j.$$

Here

$$\Lambda_{\odot r} = \left(\frac{\partial \lambda_1}{\partial x_r}, \frac{\partial \lambda_2}{\partial x_r}, \ldots, \frac{\partial \lambda_m}{\partial x_r} \right)^T$$

is the rth column in $\Lambda(x)$. It follows $z' = \omega D_z A^T \Lambda(x)$ and thus $z'_{x=x^*} = \omega A^T \Lambda(x^*)$, which completes the proof. ■

Denote by $P_{R(Z)}$ the orthogonal projection matrix onto the range of a matrix Z. We remind the reader of the relation $P_{R(a_i)} = (a_i^T a_i)^{-1} a_i a_i^T$ and consider now specially the Gauss-Seidel schemes (2.7).

Lemma 3. $g'(x^*) = D_{x^*}^{1/2}(I - \omega_i P_{R(\hat{a}_i)}) D_{x^*}^{-1/2}$, $\hat{a}_i = D_{x^*}^{1/2} a_i$. *Put* $w_i = \Sigma a_{ij}^2 x_j^*$. *Then* $\omega_i = \omega w_i / b_i$ *for* (2.8a), $\omega_i = \omega w_i$ *for* (2.8b), *and* $\omega_i = \omega$ *for* (2.9).

Proof. Note by (2.6) that λ_j^k, $j \neq i$, can be specified arbitrarily. We define $\lambda_j^k = 0$, $j \neq i$. It follows $A^T \Lambda(x^*) = a_i \Lambda_i^T$ where is the ith, and only nonzero, row of $\Lambda(x^*)$. Consider now the nonlinear SOR method and let $\lambda(x) = \lambda_i^k(x^k)$ be the solution of (2.9). Taking derivatives of this equation, we get

$$\frac{\partial f_i}{\partial \lambda} \frac{\partial \lambda}{\partial x_j} = -\frac{\partial f_i}{\partial x_j}, \qquad j = 1, 2, \ldots, n,$$

where

$$\left[\frac{\partial f_i}{\partial \lambda} \right]_{x=x^*} = -w_i \neq 0.$$

Hence by the implicit function theorem

$$\left[\frac{\partial \lambda}{\partial x_j} \right]_{x=x^*} = -\frac{a_{ij}}{w_i}.$$

It follows $\Lambda_i^T = -a_i^T / w_i$, which implies $g'(x^*) = I - \omega D_{x^*} a_i a_i^T / w_i = D_{x^*}^{1/2}(I - \omega P_{R(\hat{a}_i)}) D_{x^*}^{-1/2}$. The proofs for methods (2.8) are simpler and are left to the reader. ■

Remark 3. Consider one full iterative step in (2.7), sweeping through all rows in cyclic order, starting with the first one. Then the resulting iteration operator is $G_{\mathrm{GS}}(x^{(k+1)m}) = g(g(\ldots(g(x^{k \cdot m})\ldots))$, $k = 0, 1, \ldots,$ and it follows easily by Lemma 3 that

$$G'_{\mathrm{GS}}(x^*) = D_{x^*}^{1/2} Q_{\mathrm{GS}}(\omega_i) D_{x^*}^{-1/2}, \qquad Q_{\mathrm{GS}}(\omega_i) = \prod_{m}^{1} \left(I - \omega_i P_{R(\hat{a}_i)}\right).$$

For the Jacobi schemes (for the moment disregarding Newton's method) it is straightforward, using Lemma 2, to verify the following expression for $g'(x^*)$:

$$G'_{\mathrm{J}}(x^*) = g'(x^*) = D_{x^*}^{1/2} Q_{\mathrm{J}}(\omega_i) D_{x^*}^{-1/2}, \qquad Q_{\mathrm{J}}(\omega_i) = I - \sum_{1}^{m} \omega_i P_{R(\hat{a}_i)}. \tag{3.3}$$

Here ω_i takes the same value as for the corresponding Gauss-Seidel method (see Lemma 3).

The matrices $Q_{\mathrm{GS}}(\omega_i)$ and $Q_{\mathrm{J}}(\omega_i)$, appearing above, are the resulting iteration matrices when the linear SOR and Jacobi method respectively are applied to the following minimum norm problem (see [2], [5]):

$$\min \|x\|_2 \qquad \text{s.t.} \quad \hat{A}x = b, \quad \hat{A} = A D_{x^*}^{1/2}. \tag{3.4}$$

By applying the Kuhn-Tucker conditions on (3.4) the following system arises:

$$\hat{A}\hat{A}^T y = b, \qquad x = \hat{A}^T y. \tag{3.5}$$

We remark that the linear SOR method applied to (3.5) equals Kacmarz's projection method and is a special case of a class of projection methods studied by Bauer and Householder; for references and further details see [2].

Let Q denote any of the iteration matrices $Q_{\mathrm{GS}}(\omega_i)$ and $Q_{\mathrm{J}}(\omega_i)$. Similarly we denote by G' any of the matrices $G'_{\mathrm{GS}}(x^*)$ and $G'_{\mathrm{J}}(x^*)$. In [5] conditions on ω_i are given which insure that

$$\rho\left(Q P_{R(\hat{A}^T)}\right) < 1, \text{ and } Qu = u \quad \Leftrightarrow \quad u \in N(\hat{A}). \tag{3.6a,b}$$

Here $N(A)$ is the nullspace of a matrix A. We note that with $v = D_{x^*}u$ and using (3.6), $G'v = v \Leftrightarrow (D_{x^*}^{1/2} Q D_{x^*}^{-1/2})v = v \Leftrightarrow Q D_{x^*}^{1/2} u = D_{x^*}^{1/2} u$. Hence by (3.6b), $A D_{x^*} u = 0 \Leftrightarrow v = D_{x^*} u \in N(A)$. It follows that

$$G'v = v \quad \Leftrightarrow \quad v \in N(A). \tag{3.7}$$

From (3.6a),(3.7), and the fact that G' and Q are similarly equivalent it follows that

$$\sigma = \rho_e(Q) = \rho_e(G') = \rho\left(G' P_{R(A^T)}\right) < 1, \tag{3.8}$$

where $\rho_e(Q)$ is the largest eigenvalue in absolute value, excluding $+1$, of the matrix Q. Note by (2.4) that x^k has the form (2.1a) $\forall k$. Hence the primal iteration (2.5), or equivalently (3.2), can be viewed as a scheme for obtaining a solution of the form (2.1a) to the system $b = Ax$. Consider the corresponding iteration in the residual, $r^k = b - Ax^k = \Psi_1(x^k)$, and denote by A^+ the pseudoinverse of a matrix A. It follows $x^k = A^+(b - r^k) + z = \Psi_2(r^k)$, where $z \in N(A)$. Hence $r^{k+1} = \Psi_1(x^{k+1}) = \Psi_1(G(\Psi_2(r^k))) = R(r^k)$. We then obtain $R'(r^*) = AG'(x^*)A^+$, and thus the condition (3.8) insures that the vector $r^* = b - Ax^*$ is a point of attraction of $r^{k+1} = R(r^k)$. But $r^* = 0$, and hence (3.8) is equivalent to x^* being a point of attraction of the iterative schemes. For the Gauss-Seidel methods we have the following result:

$$0 < \omega_i < 2 \quad \Rightarrow \quad \sigma = \rho\left(G'_{GS}(x^*) P_{R(A^T)}\right) < 1. \tag{3.9}$$

Remark 4. If $\omega_i = \omega$, as for the nonlinear SOR method, then $0 < \omega < 2$ is also a necessary condition for convergence. Lent [12] proves convergence of MART, (2.11), for the case $0 \leqslant a_{ij} \leqslant 1$, $\omega = 1$. We note that $\omega_i = \omega \Sigma a_{ij}^2 x_j^* / b_i \leqslant \omega$ and hence that, provided $0 \leqslant a_{ij} \leqslant 1$, $0 < \omega < 2$ is a sufficient condition for x^* to be a point of attraction of MART. Assuming $a_{ij} \leqslant 1$, it follows similarly that $0 < \omega < 2/(\max b_i)$ is a sufficient point of attraction condition for the method (2.8b).

Remark 5. Assume that the matrix $H = AD_{x^*}A^T$ appearing in (3.5) is two-cyclic, i.e. of the form

$$H = \begin{bmatrix} D_1 & H_1 \\ H_1^T & D_2 \end{bmatrix},$$

where D_1 and D_2 are diagonal matrices. Then from the SOR theory [24], the convergence factor for the method (2.9) is minimized by $\omega_{\text{opt}} = 2/(1 + \sqrt{1 - \sigma_1}\,)$, where σ_1 is the convergence factor using $\omega = 1$. Further, the minimum value is given by $\sigma_{\text{opt}} = \omega_{\text{opt}} - 1$. Note also, as remarked in [2], that this factor is invariant under row scalings of $(A|b)$ and permutation of the unknowns. However, it does depend on the ordering of the equations in $Ax = b$.

The restrictions on ω_i for the Jacobi methods are more complex. We remark however that for the method (2.12), $0<\omega<2/m$ provides a sufficient, but rather restrictive, point of attraction condition [5].

We finally comment on the two combined schemes, the SOR-Newton and the Newton-SOR method, described in Sec. 2. We first define a dual iteration step by $\beta^{k+1}=H(\beta^k)$, and assume that β^* is a fixed point of this iteration formula. The relation (2.4) will be written in vector form as $x^k=F(\beta^k)$, and it follows that (i) $F'(\beta^*)=D_{x^*}A^T$. Let A^+ be the pseudoinverse of A. From (2.4) it follows that $\beta^k=(A^T)^+v+t=E(x^k)$, where $v_j=\ln(x_j/u_j)+1$ and $t\in N(A^T)$ (t does not depend on x^k). Hence, (ii) $E'(x^*)=(A^T)^+D_{x^*}^{-1}$. Now $x^{k+1}=F(\beta^{k+1})=F(H(\beta^k))=F(H(E(x^k)))=G(x^k)$. It follows that $G'(x^*)=F'(\beta^*)H'(\beta^*)E'(x^*)$, or by using the expressions (i) and (ii) above,

$$G'(x^*)=\left(D_{x^*}A^T\right)H'(\beta^*)\left((A^T)^+D_{x^*}^{-1}\right). \tag{3.10}$$

We conclude from (3.10) that if the two methods have the same dual rate of convergence, they also have the same primal rate of convergence. We can now apply the results given in [17, p. 325 ff.] for the two combined dual schemes. The first result is that the nonlinear SOR method has the same convergence factor σ as the SOR-Newton method. Hence it does not enhance the asymptotic rate of convergence to take more than one step with Newton's method when solving (2.9) for λ_i^k. On the other hand, by combining Newton's method (2.13) with, say, $t1$ steps of the linear SOR method [using the same value on ω as in (2.9)], the convergence factor becomes $\sigma\uparrow t1$.

4. SIDE CONDITIONS WITH ZERO-ONE STRUCTURE

We consider here a matrix A with the properties

$$a_{ij}\in\{0,1\}, \tag{4.1a}$$

and

$$A^T=(A_1,A_2,\ldots,A_r),\quad r\leqslant m,\qquad \text{s.t.}\quad A_j^TA_j=D_j\quad \forall j, \tag{4.1b}$$

where D_j is diagonal. It follows easily from the property (4.1a) that MART, (2.11), and the nonlinear SOR method for the same starting vector and the same value of ω produce the same sequence of iterates. Note, with $\omega=1$, that no evaluations of the exponential function are needed in (2.11).

It follows quite easily from the property (4.1b) that the methods (2.11) and (2.12) behave like block-SOR and block-Jacobi methods respectively. The expressions for the iteration matrices Q (note that $\omega_i=\omega$ in this case) become [5]

$$Q_{\mathrm{GS}}(\omega)=\prod_{r}^{1}\left(I-\omega P_{R(\hat{A}_j)}\right) \tag{4.2a}$$

and

$$Q_{\mathrm{J}}(\omega)=I-\omega\sum_{1}^{r}P_{R(\hat{A}_j)},\qquad \hat{A}_j=D_{x*}^{1/2}A_j. \tag{4.2b}$$

Examples of such a block structure include the unweighted picture reconstruction model, [10, 5]. Here the number of blocks, r, equals the number of projection angles. Further applications will be studied in Sec. 5.

We also note the possibility, using the relation (2.4) and (4.1a), of devising dual algorithms corresponding to (2.11) and (2.12), iterating in $\exp(\beta_i^k)$, such that no evaluations of the exponential function are needed. It is interesting to note that the continuous approach ($\min\int f\ln f$), studied in [14] in connection with inverting the Radon transform of f, gives rise to a method which behaves like a dual scheme of this type. We remark that even in the continuous case only a finite number of projection data are used, and hence that the dual vector is of finite length.

5. APPLICATIONS WITH TRANSPORTATION CONSTRAINTS

Assume that a given urban area has been partitioned into zones. In trip distribution the problem is to generate a table of demand for travel from each zone to every other. Let $S=\{k_1,k_2,\dots,k_{i_0}\}$ be the set of all zones whence a trip may depart, and $T=\{m_1,m_2,\dots,m_{j_0}\}$ the set of all zones where a trip may end. Denote by x_{ij} the number of trips from zone k_i to zone m_j. If it is desired to avoid predicting intrazonal trips, any unknown x_{ij} such that $k_i=m_j$ should be excluded from the model. We will assume that from some trip generation process, estimates $b'^T=(b_1',b_2',\dots,b_{i_0}')$ and $b''^T=(b_1'',b_2'',\dots,b_{j_0}'')$ are available of the future number of trips departing and arriving in each zone. The transportation constraints are then defined by $\Sigma_j x_{ij}=b_i'$ and $\Sigma_i x_{ij}=b_j''$. By arranging the matrix x_{ij} rowwise into a vector of length $n=i_0j_0$, these constraints take the form

$$Ax=b,\quad A^T=(A_1,A_2)\quad\text{and}\quad b=(b'^T,b''^T),\quad \sum b_i'=\sum b_j''. \tag{5.1}$$

The following example, where $i_0=j_0=3$, illustrates the nonzero structure of the matrices A_1^T $(i_0\times n)$ and A_2^T $(j_0\times n)$:

$$A_1^T=\begin{bmatrix} 1 & 1 & 1 & & & & & & \\ & & & 1 & 1 & 1 & & & \\ & & & & & & 1 & 1 & 1 \end{bmatrix},$$

$$A_2^T=\begin{bmatrix} 1 & & & 1 & & & 1 & & \\ & 1 & & & 1 & & & 1 & \\ & & 1 & & & 1 & & & 1 \end{bmatrix}.$$

Note that these matrices have disjoint rows:

$$A_1^TA_1=j_0I \quad\text{and}\quad A_2^TA_2=i_0I. \tag{5.2}$$

In a commonly used approach, called the gravity model, it is required that the solution of (5.1) have the multiplicative form $x_{ij}=r_iu_{ij}s_j$, where the positive numbers, r_i and s_j, are unknowns and $\{u_{ij}\}$ is a given estimation of the trip destination table. The name gravity model comes from the possible choice $u_{ij}=1/d_{ij}$, where d_{ij} is the squared distance between zone k_i and zone m_j. It easily follows from (2.1a) and the structure of the transportation matrix (5.1) that the gravity model solution also equals the solution of (1.1), where u_{ij} is arranged rowwise into a vector u, and A is the matrix in (5.1).

The computation of r_i and s_j can also be viewed as the problem of scaling a positive matrix $\{u_{ij}\}$ to make the row and column sums of the scaled matrix $\{x_{ij}]$ equal certain values given by b' and b'' respectively [21, 13]. The scaled matrix is of interest, e.g., when estimating the transition matrix of a Markov chain, which is known to be doubly stochastic (i.e. $b_i'=b_j''=1$ $\forall i,j$).

For $\omega=1$ the method (2.11) is well known when applied to the problem (5.1). The algorithm is usually referred to as the balancing method (see [11] for a recent survey). A mathematically equivalent, but computationally superior version was given by Murchland in [16]. Stewart in [23] provides a roundoff error analysis. Also for this problem Robillard and Stewart [20] proposed the Newton-SOR method, noting that the matrix is two-cyclic [cf. (5.2) and Remark 5 in Sec. 3]. Andersson [1] analyzes the convergence and the rate of convergence of Newton's method, solving (2.13) by Choleski factorization, and compares it with the balancing method. He further discusses simplified Newton iterations, in the spirit of Erlander [8], for solving (5.1).

We will now study the behavior of the methods (2.11) and (2.12) applied to the transportation problem. Recall from (3.8), using (4.2), that $\sigma_{\mathrm{GS}}=\rho_e(Q_{\mathrm{GS}}(\omega))$ and $\sigma_{\mathrm{J}}=\rho_e(Q_{\mathrm{J}}(\omega))$. We first derive useful equivalent expressions for the convergence factors σ_{GS} and σ_{J}, for the case $\omega=1$.

LEMMA 4. *Let $\omega=1$. Then $\sigma_{GS}=\rho_e(Q_1)$ and $\sigma_J=\rho_e(Q_2)$, where*

$$Q_1=\hat{A}_2^+\hat{A}_1\hat{A}_1^+\hat{A}_2 \quad \text{and} \quad Q_2=\begin{bmatrix} 0 & \hat{A}_1^+\hat{A}_2 \\ \hat{A}_2^+\hat{A}_1 & 0 \end{bmatrix}.$$

Proof. Consider first the method (2.12). Put $R^T=((\hat{A}_1^+)^T, (\hat{A}_2^+)^T)$. By direct computation and using $\hat{A}_j^+\hat{A}_j=I$ it follows that $Q_2=I-R\hat{A}^T$, where $\hat{A}^T=(\hat{A}_1, \hat{A}_2)$. On the other hand, from (4.2b) and using $P_{R(\hat{A}_j)}=\hat{A}_j\hat{A}_j^+$, it follows that $Q_J(1)=I-\hat{A}^TR$. Observe that $Q_2=I-R\hat{A}^T$ and $Q_J(1)=I-\hat{A}^TR$ have the same set of eigenvalues, excluding $+1$, for any matrices R $(m\times n)$ and $\hat{A}^T$ $(n\times m)$. Thus $\sigma_J=\rho_e(Q_J(1))=\rho_e(Q_2)$, which completes the proof for (2.12). With $R^T=((\hat{A}_1^+)^T, [\hat{A}_2^+(I-P_{R(\hat{A}_1)})]^T)$ a similar argument gives the proof for the method (2.11). ■

The optimal values of ω and σ for the transportation problem can now be derived.

LEMMA 5. $\omega_{\text{opt}}=2/(1+\sin\Psi)$, $\sigma_{\text{opt}}=\omega_{\text{opt}}-1$ *for* (2.11); $\omega_{\text{opt}}=1$, $\sigma_{\text{opt}}=\cos\Psi$ *for* (2.12). *Further, if $\omega=1$ in* (2.11), *then $\sigma=\cos^2\Psi$.*

Proof. Consider the eigenvalue problem $Q_2x=\lambda x$, $x^T=(x_1^T, x_2^T)$. By eliminating x_1 it follows easily that $Q_1x_2=\lambda^2x_2$. Hence by defining $\rho_e(Q_2)=\cos\Psi$ the results for $\omega=1$ are proved. The optimal value for ω (in the sense that $\sigma_{\text{opt}}=\min_\omega\sigma$) in (2.12) are given by $\omega_{\text{opt}}=2/(a+b)$, where a is the smallest nonzero eigenvalue of $\Sigma P_{R(\hat{A}_j)}$ and b the largest [5]. We further note that $-\cos\Psi$ is also an eigenvalue of the matrix Q_2 [with eigenvector $(-x_1^T, x_2^T)$]. It follows that $a=1-\cos\Psi$ and $b=1+\cos\Psi$, and hence $\omega_{\text{opt}}=1$. The expression for ω_{opt} in (2.11) follows by (5.2) and Remark 5 in sec. 3. ■

REMARK. The angle Ψ can be identified as the smallest nonzero angle between the two subspaces $R(\hat{A}_1)$ and $R(\hat{A}_2)$ [5]. Note, by Lemma 5, that the method (2.11) requires asymptotically a factor of $\log(\omega_{\text{opt}}-1)/\log(\cos^2\Psi)\approx 2/\sin\Psi$ fewer iterations using $\omega=\omega_{\text{opt}}$ than using $\omega=1$ to achieve a certain accuracy.

One of the rows in (5.1) is redundant (since the sum of all rows in A_1^T equals the sum of all rows in A_2^T). In fact, any row can be deleted without changing the solution of the problem. It is our computational experience, however (cf. Table 2) that the rate of convergence of the balancing method decreases by deleting a redundant row. We will briefly discuss this issue.

Let $C=A_1^TD_{x*}A_2$. It is straightforward to verify that the i, jth element of C equals x_{ij}^*. Further it can be verified that $A_1^TD_{x*}A_1=D_{b'}$. It follows, using the expression $A_j^+ = (A_j^T A_j)^{-1} A_j^T$, that the matrix Q_1 appearing in Lemma 4 can be written $Q_1 = D_{b''}^{-1} C^T D_{b'}^{-1} C$. Assume that the kth row is deleted in A_1^T. This corresponds to deleting the kth row, α_k^T, in C and the kth row in $D_{b'}$. Denote by Q_{NS} the resulting expression for Q_1. Let e_k be the kth unit vector, and put $E = C - e_k \alpha_k^T$. We then get

$$Q_{\mathrm{NS}} = D_{b''}^{-1} E^T D_{b'}^{-1} E = Q_1 - \frac{1}{b'_k} D_{b''}^{-1} \alpha_k \alpha_k^T. \tag{5.3}$$

Put $R = C D_{b''}^{-1} C^T D_{b'}^{-1}$ (note that Q_1 and R have the same set of nonzero eigenvalues) and denote by R_{NS} the corresponding matrix when the kth row in A_2^T is deleted. Let β_k be the kth column in C. It follows that

$$R_{\mathrm{NS}} = R - \frac{1}{b''_k} D_{b'}^{-1} \beta_k \beta_k^T. \tag{5.4}$$

In general it seems hard to carry this analysis further. We observe however the following simple case, which is easily verified using the given expressions for Q_1, Q_{NS}, and R_{NS} respectively.

LEMMA 6. *Let $\omega = 1$, and suppose $x_{ij}^* = 1$ $\forall i, j$. Then the convergence factor for* (2.11) *becomes $\sigma = 0$ if all rows in A are kept, $\sigma = 1 - 1/i_0$ if one row in A_1^T is deleted, and $\sigma = 1 - 1/j_0$ if one row in A_2^T is deleted.*

6. MATRIX BALANCING

Let $V = \{v_{ij}\}$ be a given irreducible, nonsymmetric matrix of order $m \times m$ with no null rows or null columns. Matrix balancing is the problem of finding D, a nonsingular diagonal matrix, such that $X = D^{-1} V D$ is balanced, i.e., corresponding row and column norms are all equal. This routine is recommended to precede the computation of eigenvalues of V. An algorithm for balancing in l_2-norm has been given by Osborne [18]. Parlett and Reinsch [19] describe this method for any l_p-norm, restricting all scalings to numbers that are integer powers of the machine base.

We will show that the balancing problem can be written on the form (1.1). We start by defining the balancing conditions,

$$\sum_{j \neq 1i} x_{ij}^p = \sum_{j \neq i} x_{ji}^p, \quad i = 1, 2, \ldots, n \qquad (x_{ij} > 0). \tag{6.1}$$

These equations can also be written

$$ED_x^{p-1}x=0,$$

with

$$E=E_1-E_2 \quad \text{and} \quad x=(x_{12},x_{13},\dots,x_{21},x_{23},\dots,x_{m,m-1})^T.$$

The matrices E_1 and E_2 have the same structure as A_1 and A_2 [with $i_0=j_0=m$ and $n=m(m-1)$] defined in Sec. 5, except that columns corresponding to the unknowns x_{ii}, $i=1,2,\dots,m$, have been deleted.

LEMMA 7. *Let $x^*=(x_{12}^*,x_{13}^*,\dots,x_{21}^*,x_{23}^*,\dots,x_{m,m-1}^*)^T$ be the solution of the problem* (1.1) *with $A=E$ and $u_j=|v_j|^pe$ $(e=2.71\dots)$, $j=1,2,\dots,n$. Denote by β^* the corresponding value of the dual vector, and define $d_i=\exp(-\beta_i^*/p)$, $i=1,2,\dots,m$. Then the matrix $|X|=\{|x_{ij}|\}$, defined from $x_{ij}=(x_{ij}^*)^{1/p}\operatorname{sign}(v_{ij})$, is a solution of* (6.1) *of the form $x_{ij}=d_i^{-1}v_{ij}d_j$.*

Proof. Obviously $\{|x_{ij}|\}$ satisfies (6.1). From the dual relation (2.1a) and the structure of the matrix E (note that E has two nonzero elements, $+1$ and -1, per column), it easily follows that $x_{ij}^*=\exp(\beta_i^*)(u_{ij}/e)\exp(-\beta_j^*)$. Hence $x_{ij}=d_i^{-1}|v_{ij}|\operatorname{sign}(v_{ij})d_j=d_i^{-1}v_{ij}d_j$. ■

We now consider the method (2.9), the nonlinear SOR method, for solving the matrix balancing problem. Due to the simple structure of the matrix E, $e_{ij}\in\{-1,0,+1\}$, it is possible to compute explicitly the number λ_i^k occurring in (2.9). Thus from (2.9) and (6.1), with $p=1$, it follows that $\lambda_i^k=0.5\ln t$ with $t=\Sigma_{j\neq i}x_{ij}^k/\Sigma_{j\neq i}x_{ji}^k$. By Remark 4 in Sec. 3, x^* is a point of attraction if and only if $0<\omega<2$. However, and in contrast to the case with transportation constraints, the matrix $ED_{x^*}E^T$ is not, in general, two-cyclic, and thus the SOR theory is not applicable here. We also remark that (2.9), with $\omega=1$, is identical to Osborne's method, a fact that is readily verified. This fact was also recently pointed out by Lamond; see [11] for further references.

7. NUMERICAL RESULTS

We first present some results which illustrate how the choice of the iteration parameter ω in (2.11) affects the rate of convergence. The computations were made on a DEC-10 computer with a relative precision of 8

TABLE 1
BEHAVIOR OF CURRENT RESIDUAL WHEN SCALING MATRIX C USING $\omega=1$ IN (2.11)

k	$\|s^k\|_2\times 100$	σ[a]	ω
30	0.207		
		0.9951	1.87
60	0.179		
		0.9955	1.87
90	0.156		
		0.9958	1.88
120	0.137		
		0.9961	1.88
150	0.122		
		0.9963	1.89
180	0.109		

[a] The convergence factor is estimated by $\sigma\approx 10\uparrow\exp$, where $\exp=(\Delta\log\|s^k\|_2)/\Delta k$ and $\omega=2/(1+\sqrt{1-\sigma})$.

decimal places. The iterations were terminated when $\|\text{res}\|_2/\|b\|_2<10^{-6}$. Here $s_i=(\text{res})_i=b_i-a_i^Tx^k$, $i=(k \bmod m)+1$. We will refer to the vector s $(m\times 1)$ as the current residual. In all tests the starting vector x^0 was taken equal to u.

Consider first the problem of scaling the following matrices, taken from [13], so that they become doubly stochastic [i.e. $(b')^T=(b'')^T=(1,1,1)$]:

$$A=\begin{bmatrix}10^4 & 10^2 & 10^2\\ 10^2 & 1 & 1\\ 10^2 & 1 & 1\end{bmatrix},\qquad C=\begin{bmatrix}10^2 & 10^2 & 0\\ 10^2 & 10^4 & 1\\ 0 & 1 & 10^2\end{bmatrix}.$$

Table 1 illustrates the computation of ω_{opt} (cf. Remark 5 in Sec. 3). In Table 2 we summarize the number of iterations and the CPU times required when

TABLE 2
NUMBER OF ITERATIONS, MEASURED CPU TIME, AND VALUE OF ω_{opt} WHEN SOLVING THE SCALING PROBLEMS WITH (2.11)

	No. of iterations (CPU time [sec])		
Problem	$\omega=1$	$\omega=\omega_{\text{opt}}$	ω_{opt}
A	2 (0.16)	2 (0.16)	1
A_{ns}	33 (0.12)	17 (0.22)	1.27
C	2584 (2.07)	138 (0.38)	1.90
C_{ns}	7833 (6.50)	294 (0.55)	1.94

TABLE 3
NUMBERS OF ITERATIONS AND MEASURED CPU TIME WHEN SOLVING TWO RECONSTRUCTION PROBLEMS

	MART ($\omega = 1$)		Newton		
Problem	No. of iterations	CPU time (sec)	No. of iterations	Total no. of inner iterations	CPU time (sec)
b1	7	1.2	5	19	3.3
b2	10	1.2	16	85	6.3

scaling A and C. The corresponding problems with the last row in A_1^T deleted are also included. The results in the third column were obtained by using ω_{opt} during the whole computation.

Finally we report on a numerical comparison between MART, (2.11), and Newton's method. We have used the implementation of Newton's method given by Eriksson [6], where the conjugate gradient algorithm is applied when solving (2.13). For Newton's method we used $\text{res} = b - Ax^k$ in the stopping criterion. The test example is taken from [15] and results from a modification of the unweighted picture reconstruction model, which compensates for the lack of weights in the case with three projection angles. The corresponding linear system has the structure (4.1) with $r = 3$ and u a vector with all components equal to $+1$ (there is no *a priori* information available about the unknown picture). We have solved this problem using $m = 63$ and $n = 333$ for two different right hand sides, namely (i) $x_i = v_i$, $i = 1, 2, \dots, n$, where v_i is a uniformly distributed random number between zero and one, and $\text{b1} = Ax$, and (ii) x equals the test picture given in [15, Fig. 8a] and $\text{b2} = Ax$. Table 3 shows the necessary number of iterations as well as the corresponding CPU times when solving these two systems.

The matrix in this application has two redundant rows. By deleting two rows, giving a matrix with full row rank, the rate of convergence in MART was drastically changed. For example, in case (i) the spectral radius was increased from $\rho = 0.07$ (singular case) to $\rho = 0.99$ (nonsingular case).

REFERENCES

1 P-Å. Andersson, On the convergence of iterative methods for the distribution balancing problem, Report LiTH-MAT-R-79-14, University of Linköping, Sweden, 1979.

2 Å. Björck and T. Elfving, Accelerated projection methods for computing pseudoinverse solutions of systems of linear equations, *Nordisk Tidskr. Informationsbehandling (BIT)* 19:145–163 (1979).

3 L. M. Bregman, The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming, *USSR Computational Math. and Math. Phys.* 7:200–217 (1967).

4 Y. Censor and A. Lent, An iterative row-generation method for interval convex programming, Technical Report MIPG11, Medical Image Processing Group, Dept. of Computer Science, SUNY at Buffalo, Amherst, N.Y., 1978.

5 T. Elfving, Block-iterative methods for consistent and inconsistent linear equations, *Numer. Math.* 35:1–12(1980).

6 J. Eriksson, Solution of large sparse maximum entropy problems with linear equality constraints, *Math. Programming* 18:146–154(1980).

7 S. Erlander, Entropy in linear programs—an approach to planning, Report LiTH-MAT-R-1977-3, University of Linköping, Sweden, 1977.

8 S. Erlander, A note on balancing procedures for trip distribution estimation, *Transportation Res.* 12:335–336 (1978).

9 B.R. Frieden, Image enhancement and restoration, in *Picture Processing and Digital Filtering* (T. Huang, Ed.), Springer, Berlin, 1975.

10 R. Gordon, R. Bender, and G. T. Herman, Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and x-ray photography, *J. Theoret. Biol.* 29:471–481 (1970).

11 B. Lamond and N. F. Stewart, Bregman's balancing method, Publ. 354, Dépt. d'IRO, Université de Montréal, Montréal, Canada, 1980.

12 A. Lent, Maximum entropy and multiplicative ART, in *Image Analysis and Evaluation, SPSE Conference Proceedings*, (R. Shaw, Ed.), Toronto, Canada, 1976, pp. 249–257.

13 A. W. Marshall and I. Olkin, Scaling of matrices to achieve specified row and column sums, *Numer. Math.* 12:83–90 (1968).

14 G. N. Minerbo, MENT: A maximum entropy algorithm for reconstructing a source from projection data, *Comput. Graphics and Image Processing* 10:48–68 (1979).

15 G. N. Minerbo and J. G. Sanderson, Reconstruction of a source from a few (2 or 3) projections, Report LA-6747-MS, Los Alamos Scientific Laboratory, Los Alamos, N. Mex., 1977.

16 J. D. Murchland, *Biproportional Problem*, Lecture notes in Traffic Studies 1976:1:8, Transport Studies Group, University College, London, 1976.

17 J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic, New York, 1970.

18 E. E. Osborne, On pre-conditioning of matrices, *J. Assoc. Comput. Mach.* 7:338–345 (1960).

19 B. N. Parlett and C. Reinsch, Balancing a matrix for calculation of eigenvalues and eigenvectors, *Numer. Math.* 13:293–304 (1969).

20 P. Robillard and N. F. Stewart, Iterative numerical methods for trip distribution problems, *Transportation Res.* 8:575–582 (1974).

21 R. Sinkhorn, Diagonal equivalence to matrices with prescribed row and column sums, *Amer. Math. Monthly* 74:402–405 (1967).

22 F. Snickars and J. W. Weibull, A minimum information principle—Theory and practice, *Regional Science and Urban Economics* 7:137–168 (1977).

23 N. F. Stewart, The effect of round-off error on computed solutions of trip distribution problems, *Transportation Res.* 13b:227–228 (1979).

24 D. M. Young, *Iterative Solutions of Large Linear Systems*, Academic, New York, 1971.

25 W. Zangwill, *Nonlinear Programming: a Unified Approach*, Prentice-Hall, Englewood Cliffs, N.J., 1969.

Received 23 July 1979; revised 27 May 1980