ESTIMATING THE CONSUMPTION MATRIX FROM INEXACT DATA IN THE LEONTIEF MODEL

STEFANIA RAGNI*, CARMELA MARANGI†, FASMA DIELE‡, AND MOODY T. CHU§

Abstract. The Leontief model, originally developed for describing an economic system in terms of mutually interrelated and structurally conditioned simultaneous flows of commodities and services, has important applications to wide ranging disciplines. A basic model assumes the linear form $\mathbf{x} = T\mathbf{x} + \mathbf{d}$, where $\mathbf{x}$ represents the total output vector and $\mathbf{d}$ represents the final demand vector. The consumption matrix T plays the critical role of characterizing the entire input-output dynamics. Normally, T is determined by massive and arduous data gathering means which inadvertently bring in measurement noises. This paper considers the inverse problem of reconstructing the consumption matrix in the open Leontief model based on a sequence of inexact output vectors and demand vectors. Such a formulation might have two advantages: one is that no internal consumption measurements are required and the other is that inherent errors could be reduced by total least squares techniques. Several numerical methods are suggested. A comparison of performance and an application to real-world data are demonstrated.

Key words. Leontief model, input-output analysis, consumption matrix, M- matrix, inverse problem, inexact data, constrained total least squares, projected gradient

AMS subject classifications. 91B02, 15A48, 65F20, 90C30, 65K05

1. Preliminaries. The input-output model first envisioned, developed, and stimulated by Leontief has tremendous impact in wide ranging applications. Subjects that could be studied by this model include the effects of disarmament or military base closures on local economy, the costs of pollution, the depletion of mineral resources, the impact of automation on workers, and the projection of world economy [9, 11]. An input-output analysis of the comprehensive source data collected by the U.S. Department of Commerce (DOC) about the national economy, for example, can provide detailed information on the flows of the goods and services that comprise the production process of industries which, in turn, influences critical decisions made by the government, private business, households, and individuals. More specifically, one of the principal establishments by the DOC is the Benchmark Input-Output Accounts based primarily on data from the quinquennial (and now annual since 1998) economic censuses conducted by the Bureau of the Census and supplemented by data from other sources [1]. The matrix tables in these accounts, mostly measured in the money value of all goods and services delivered by each industry to each other industry in a census year, show how industries interact, that is, how industries provide input to, and use output from, each other to produce Gross Domestic Product.

The literature for the input-out analysis is voluminous. Without repeating too many details, we briefly outline some basic background information below for introducing later ideas. In a Leontief system, it is assumed that

- An economy is disaggregated into n sectors $s_1, \ldots, s_n$ of industries.
- Each of the n sectors produces a single kind of commodity.
- The production in each sector requires the transformation of goods from several others sectors in a linear way into the good in the current sector.

Suppose t_{ij} represent the units of good produced by sector s_i that is needed to produce one unit of good by sector s_j. Let x_i denote the production level of sector s_i. Let p_i denote the unit price of commodity by sector s_i. Then the summation $\sum_{j=1}^{n} t_{ij} x_j$ stands for the number of units produced by sector s_i that has been consumed by the economy, whereas the summation $\sum_{i=1}^{n} t_{ij} p_i$ stands for the unit cost of the commodity produced by sector s_j. Assume further that an economy has to satisfy an outside demand d_i

of the good produced from sector s_i. Then it is desired that for each sector s_i, $i = 1, \ldots, n$, the equation

$$x_i = \sum_{j=1}^{n} t_{ij} x_j + d_i \qquad (1.1)$$

must be satisfied. In matrix form, the *open Leontief model* can be expressed as

$$\mathbf{x} = T\mathbf{x} + \mathbf{d} \qquad (1.2)$$

where $\mathbf{x} = [x_1, \ldots, x_n]^\top$, $T = [t_{ij}]$, and $\mathbf{d} = [d_1, \ldots, d_n]^\top$. Likewise, entries in the row vector

$$\mathbf{v}^\top := \mathbf{p}^\top - \mathbf{p}^\top T, \qquad (1.3)$$

where $\mathbf{p} = [p_1, \ldots, p_n]^\top$, represent the net revenues per unit commodity produced by the sectors $s_1, \ldots, s_n$, respectively. In a fair market, it is reasonable to expect that the supply and demand should satisfy the equation

$$\underbrace{\mathbf{v}^\top \mathbf{x}}_{\text{total net revenue}} = \underbrace{\mathbf{p}^\top \mathbf{d}}_{\text{market price}} . \qquad (1.4)$$

In the literature, the matrix T has many names — consumption matrix, technology matrix, direct requirement matrix, input-output matrix, and so on. In this paper, we shall call T the *consumption matrix*. Typically, the entries of T are taken as the ratios

$$t_{ij} = \frac{u_{ij}}{x_j}, \qquad (1.5)$$

where u_{ij} represents the amount of the product from sector s_i used by sector s_j as intermediate input to produce the amount x_j. It is this quantity u_{ij} which is usually collected, for example, by the Bureau of the Census through industrial surveys. The matrix $U = [u_{ij}]$ is referred to as the *Use Table* in the official documents by the Bureau of Economic Analysis (BEA) [1]. The size of the matrix U can be relatively large. The Standard Industrial Classification (SIC) system (which has been replaced by the new North American Industry Classification System (NAICS) since 1997) classifies economic sectors into industries on the basis of their primary activities. In 1992, for example, the economy was separated into 97 I/O industry groups from 498 industry sectors. In this case, a detailed consumption can be of size 498 by 498.

One general application of the input-output analysis is to determine the production level $\mathbf{x}$ from the overall industries in order to meet a given demand vector $\mathbf{d}$. That is, solve the linear equation (1.2) for $\mathbf{x}$ from a given $\mathbf{d}$. From a practical point of view, we are only interested in a nonnegative solution $\mathbf{x}$ of (1.2) and a nonnegative $\mathbf{p}$ of (1.3). In these cases, we say the Leontief model is *feasible* and *profitable*, respectively. The following result is well established in the literature. See, for example, [2, Theorem 9.3.9].

THEOREM 1.1. *In an open Leontief model with consumption matrix T, the following statements are equivalent:*

1. *The model is feasible.*
2. *The model is profitable.*
3. *The matrix $A := I - T$ is a nonsingular M-matrix.*

The notion of M-matrix is another well studied subject in the literature. A total of 50 equivalent conditions, for example, are listed in [2, Theorem 6.2.3] for the statement "A is a nonsingular M-matrix". On the other hand, the matrix $(I - T)^{-1}$ is commonly referred to as the *Leontief inverse*. The (i, j) entry in the Leontief inverse has a special meaning — it is the amount by which sector s_i must change it production level to satisfy an increase of one unit in the final demand from sector s_j. The Leontief inverse thus plays an essential role in input-output modelling. Efforts addressing the concern over the estimation of the Leontief

inverse when errors are presented in the consumption matrix T can be found in [4, 8]. In contrast, the main goal of this paper is to estimate T using consecutive measurements of inexact total outputs and final demands.

The Leontief model "describes the economic system in terms of mutually interrelated and structurally conditioned, simultaneous flows of commodities and services [10]." The system (1.2) is also known as a *static input-output model* because it represents the "flow" of commodities only at a single interval of time. It is sometimes desirable to reflect changes in time and to take into account model components that are constantly changing as a result of previous actions or future expectations [12]. For example, in the model (1.2) the final demand vector **d** comprises not only consumption goods, but also investment goods including additions to the stock of fixed capitals such as buildings, machinery, tools and so on. It is perhaps more reasonable to assume that the investment should not be taken as given from outside, but rather be explained within the model. An illustration of such a scenario is that one reason to justify an investment is the expectation that the expansion of production capacity, say, buying machines, should match in some way with the expansion of the output level demanded. Let $\mathbf{x}_k$ denote the vector of total outputs at the k-th time period. In contrast to (1.2), a simple *dynamic input-output model* reflecting the above-mentioned investment policy is of the form

$$\mathbf{x}_k = T\mathbf{x}_k + \underbrace{B(\mathbf{x}_{k+1} - \mathbf{x}_k) + \mathbf{y}_k}_{\mathbf{d}_k}, \tag{1.6}$$

where $\mathbf{y}_k$ is the demand vector excluding fixed capital investment, $B = [b_{ij}]$ is the *transit matrix* with b_{ij} defining the investment required of sector s_i in order to increase one unit of output capacity of sector s_j. Again, from the investment or management point of view, it will be interesting to estimate either T or B based on a sequence of observed vectors $\{\mathbf{x}_k\}$ and $\{\mathbf{y}_k\}$.

2. Inverse Problem. In either the static or the dynamic model described above, the usual task is the forward problem of determining, by assuming that the consumption matrix T and the transit matrix B are known and fixed, the production level from given demand vector. It is conceivable that, in a complicated and interwoven economic system, the coefficients of dependence t_{ij} and b_{ij} between any two given sectors s_i and s_j are never known precisely. Though the economy is actively engaging and inputs and outputs are revolving as we see them in the market, often we do not quite comprehend the "invisible hand" behind that drives the economy. That is, even after we have assumed the practicality of the Leontief model for the economy, often we do not know the exact consumption matrix T nor the transit matrix B. By the *inverse problem* of the Leontief model, we mean the estimation of either T or B based on a sequence of observed output vectors $\mathbf{x}_k$ and demand vectors $\mathbf{d}_k$ (or $\mathbf{y}_k$ in (1.6)). For simplicity, we will have to assume, as did the BEA before 1998 since the benchmark were made quinquennially, that during the period of observation the matrices T and B remain constant. Obviously, the fact that the resulting model must be feasible imposes a constraint on the estimation. In what follows we study only the inverse problem for estimating T. The general inverse problem is still an ongoing research project.

2.1. Formulation. At its first glance, the inverse problem is straightforward to solve. Assuming the static Leontief model (1.2), if there are n corrected observed output vectors $\tilde{X} = [\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_n]$ and demand vectors $\tilde{D} = [\tilde{\mathbf{d}}_1, \ldots, \tilde{\mathbf{d}}_n]$, then

$$T = I - \tilde{D}\tilde{X}^{-1}. \tag{2.1}$$

However, questions arise under any of the following scenarios:

S1. Either $\tilde{\mathbf{x}}_k$ or $\tilde{\mathbf{d}}_k$ (and often both) cannot be measured precisely.

S2. There are not enough data collected, that is, an estimation is to be made with less than n pairs $(\tilde{\mathbf{x}}_k, \tilde{\mathbf{d}}_k)$ of observations.

S3. There are too many observations and consistency is not necessarily maintained.

In the case of S1, the notion of total least squares should be brought in [13]. In the case of S2, we shall have an underdetermined system and multiple solutions are expected. In the case of S3, we encounter an overdetermined system. In all situations, the constraint is that $I - T$ must be a nonsingular M-matrix. We think both the theory of solvability and the practice of computation deserve further consideration.

Let $\mathbb{R}_+^{n \times m}$ denote the cone of nonnegative matrices of size $n \times m$. Assume that there are p observations. Given inexact data $\tilde{X}$, $\tilde{D} \in \mathbb{R}_+^{n \times p}$, one possible formulation of reconstructing the consumption matrix in the Leontief model is to consider the *constrained total least squares problem* stated as follows:

$$\min_{\substack{X, D \in \mathbb{R}_+^{n \times p} \\ \text{The equation } (I - T)X = D \text{ is solvable for } T \in \mathbb{R}_+^{n \times n} \\ \text{The matrix } I - T \text{ is a nonsingular } M\text{-matrix}}} \frac{1}{2} \| \left[\tilde{X} - X | \tilde{D} - D \right] \|_F. \tag{2.2}$$

Because $T \in \mathbb{R}^{n \times n}$ is the parameter to be found, it might be more convenient to rewrite (2.2) in the transpose of its matrices. Without causing any ambiguity, we shall use henceforth the same notation to represent the transpose of the problem, that is,

$$\min_{\substack{X \in \mathbb{R}_+^{p \times n}, \\ T \in \mathbb{R}_+^{n \times n} \\ I - T \text{ is a nonsingular } M\text{-matrix.}}} \frac{1}{2} \| \left[\tilde{X} - X | \tilde{D} - X(I - T) \right] \|_F^2. \tag{2.3}$$

By using the Neumann lemma [2, Lemma 2.1], the M-matrix constraint can be expressed in terms of a spectral constraint. We rewrite (2.3) as follows:

$$\min_{\substack{X \in \mathbb{R}_+^{p \times n}, \\ T \in \mathbb{R}_+^{n \times n}, \\ \rho(T) < 1.}} \frac{1}{2} \| \left[\tilde{X} - X | \tilde{D} - X(I - T) \right] \|_F^2. \tag{2.4}$$

Likewise, by exploiting condition I_{28} of [2, Theorem 2.3], we can formulate (2.3) as an algebraically constrained optimization problem:

$$\min_{\substack{X \in \mathbb{R}_+^{p \times n}, \\ T \in \mathbb{R}_+^{n \times n}, \\ X(I - T) \in \mathbb{R}_+^{p \times n}.}} \frac{1}{2} \| \left[\tilde{X} - X | \tilde{D} - X(I - T) \right] \|_F^2. \tag{2.5}$$

One of our main contributions in this paper is the description of numerical methods for solving either (2.4) or (2.5). In particular, we demonstrate numerically that the projected gradient approach is perhaps the most robust and efficient. More details are given in Section 3.

2.2. Solvability. For convenience, denote the objective function by

$$g(X, T) := \frac{1}{2} \left\{ \langle \tilde{X} - X, \tilde{X} - X \rangle + \langle \tilde{D} - X(I - T), \tilde{D} - X(I - T) \rangle \right\}, \tag{2.6}$$

where $\langle A, B \rangle = \sum_{i,j} a_{ij} b_{ij}$ denotes the Frobenius inner product of two matrices A and B. It is easy to see that the gradient $\nabla g(X, T)$ with respect to the Frobenius inner product is given by the pair of matrices

$$\nabla g(X, T) = - \left(\tilde{X} - X + (\tilde{D} - X(I - T))(I - T)^\top, -X^\top (\tilde{D} - X(I - T)) \right). \tag{2.7}$$

Because neither the objective function nor the constraints in problem (2.4) is convex, a brief remark on the solvability might be worth mentioning.

Consider first the simplest case when $n = p = 1$. It is easy to see that the objective function $g(X, T)$ has only two critical points in $\mathbb{R}^2$. These are

$$(0, 1 + \frac{\tilde{X}}{\tilde{D}}) \quad \text{and} \quad (\tilde{X}, 1 - \frac{\tilde{D}}{\tilde{X}}).$$

The first critical point is infeasible, if we assume that both $\tilde{X}$ and $\tilde{D}$ are positive scalars to begin with. It is interesting to observe that if $\tilde{D} > \tilde{X}$, that is, if the observed demand is higher than the observed output, then the second critical point is also infeasible. Even in the feasible case where $\tilde{D} \leq \tilde{X}$, the minimum occurs at $(\frac{\tilde{X}+\tilde{D}}{2}, 0)$ which would still be "undesirable" because the consumption matrix(scalar) T is identically zero. This simple consideration is an indicator of a subtle question that the consumption matrix T estimated from the total least squares formulation needs to be properly interpreted. We point out at this early stage that a similar issue of interpretation arises in Example 3 of Section 4 when our numerical methods are applied to some real EBA data.

For the general multidimensional case, we observe that the function $g(X, T)$ is coercive, that is, $g(X, T) \to \infty$ as $\|(X, T)\| \to \infty$ (with respect to any norm). Therefore, the minimum of $g(X, T)$ must occur within a bounded subset, say $\mathcal{B}$, of $\mathbb{R}_+^{p \times n} \times \mathbb{R}_+^{n \times n}$. For arbitrary but fixed positive scalar $\epsilon < 1$, $g(X, T)$ must attain a global minimal value $g(X_\epsilon, T_\epsilon)$ within the compact subset $\{(X, T) \in \overline{\mathcal{B}} \mid \rho(T) \leq \epsilon < 1\}$. What is not clear about, but would be interesting to investigate in a separate study, is the limiting behavior of (X_ϵ, T_ϵ) as ϵ approaches 1. The limit point may or may not exist. For computational purpose, we usually implement the constraint $\rho(T) \leq 1$ with the hope that the spectral radius at the (local) minimizer is strictly less than one. Through our extensive numerical experiments, we are inclined to think that it is often the case in practice.

The issue of solvability can also be addressed in terms of the optimality conditions. We briefly outline the idea below. Assume, without loss of generality, the generic condition that T is irreducible. By the Perron-Frobenius theorem, the spectral radius $\rho(T)$ is a simple eigenvalue with positive eigenvector. It is not difficult to argue that $\rho(T)$ is differentiable with respect to T. Indeed, let $\mathbf{u}$ and $\mathbf{v}$ denote, respectively, the left and right eigenvectors of T corresponding to $\rho(T)$. Assume further that $\mathbf{u}$ and $\mathbf{v}$ are normalized to $\mathbf{u}^\top \mathbf{v} = 1$. Then it can be shown that

$$\frac{\partial \rho(T)}{\partial T} = \mathbf{u} \mathbf{v}^\top. \tag{2.8}$$

This piece of derivative information can be used to facilitate numerical computation in various ways which will be exploited in Section 4.

The positivity of dominant eigenvectors implies that the gradient $\frac{\partial \rho(T)}{\partial T} > 0$. It follows that the *constraint qualification* [6, Lemma 9.2.2] is satisfied by the constraints in problem (2.4). Therefore, the KKT conditions for problem (2.4) can be stated as follows.

THEOREM 2.1. *Suppose (X, T) is an optimal solution to (2.4). Let $\mathbf{u}$ and $\mathbf{v}$ be the normalized left and right eigenvectors of T corresponding to $\rho(T)$. Then in addition to satisfying the feasibility conditions $X \geq 0$, $T \geq 0$, and $\rho(T) \leq 1$, it is necessary that*

$$\Phi = (\tilde{X} - X) + (\tilde{D} - X(I - T))(I - T)^\top \leq 0, \tag{2.9}$$
$$\Psi = X^\top(\tilde{D} - X(I - T)) + \lambda \mathbf{u} \mathbf{v}^\top \geq 0, \tag{2.10}$$
$$\lambda = \lambda \rho(T) \geq 0, \tag{2.11}$$
$$\Phi. * X = 0, \tag{2.12}$$
$$\Psi. * T = 0. \tag{2.13}$$

where .* denote the Hadamand product of two matrices.

5

In the above, we have used the notation $M \geq 0$ to indicate a nonnegative matrix M. Note that both matrices $-\Phi$ and Ψ and the scalar λ serve as the Lagrange multipliers. In a similar manner, the KKT conditions for problem (2.5) can be stated in the following way.

THEOREM 2.2. *Let (X, T) be an optimal solution to (2.5). Then the feasibility conditions $X \geq 0$, $T \geq 0$, $X(I - T) \geq 0$ have to be satisfied and, in addition, there must exists a matrix $M \in \mathbb{R}^{p \times n}$, $M \geq 0$, such that*

$$\Phi = (\tilde{X} - X) + (\tilde{D} - X(I - T))(I - T)^\top + M(I - T)^\top \leq 0, \tag{2.14}$$
$$\Psi = X^\top (\tilde{D} - X(I - T)) + X^\top M \geq 0, \tag{2.15}$$
$$M. * (X(I - T)) = 0, \tag{2.16}$$
$$\Phi. * X = 0, \tag{2.17}$$
$$\Psi. * T = 0. \tag{2.18}$$

3. Numerical Methods. To our knowledge, the constrained total least squares formulation for the inverse Leontief problem has not been studied in the literature. The nonlinear or spectral constraint also makes the classical total least squares techniques impractical. In this section, we propose three approaches for solving the constrained total least squares problem (2.3) numerically. The two formulations (2.4) and (2.5) are equivalent, but have different characteristics on their constraints. Thus for each approach outlined below, we describe its variants applied to (2.4) and (2.5) separately. A preliminary comparison of their performance will be reported in Section 4.

3.1. Classical Optimization Approach. By considering X and T as the unknown parameters, it is possible to employ classical optimization techniques, such as the trust region method or the sequential quadratic programming method [6], to tackle the constrained problems (2.4) or (2.5). Without specifying which techniques are being employed, we shall refer to any types of these classical schemes for problem (2.4) and (2.5) as Algorithms C1 and C2, respectively. Plenty of existing software is readily available for use in this application. Thus we shall make no effort to review any programming details in this presentation. Be it sufficient to say that built in these algorithms are various tactics to satisfy the KKT conditions described earlier. For later comparison, we use in this paper the MATLAB routine FMINCON as the "classical" scheme. Obviously, many other routines could be used in its place and the performance could be very different.

It is important to point out that there are $n(n + p)$ unknowns in the underlying optimization problem. Thus the problem can easily be large-scaled. Unless the inherent structure is further exploited, this approach in general is more expensive.

3.2. Projected Gradient Approach. Though it is known to converge only linearly, the method of line search in the descent direction remains to be one of the cheapest and easiest ways to tackle unconstrained optimization problems. For linearly constrained problems, the steepest descent idea can be modified with the application of projected gradient. In our case, we have a mixture of nonlinear inequality constraints. However, we find it possible to enforce these inequality constraints by simple projections or scalings.

Given a matrix S, let $\mathcal{P}(S)$ denote its projection to the cone of nonnegative matrices. That is, define the (i, j) entry of $\mathcal{P}(S)$ by

$$(\mathcal{P}(S))_{ij} := \max\{s_{ij}, 0\}. \tag{3.1}$$

We propose the following hybrid scheme for solving the constrained problem (2.4). The basic idea is to employ the steepest descent method so long as the constraints are inactive. A correction is made only when a constraint is violated. In that case, the corrected iterate is put back to the "boundary" of the feasible set. For later reference, this method is referred to as Algorithm G1.

ALGORITHM G1. Given initial value $(X^{(0)}, T^{(0)})$ that is strictly feasible, do the following for $k = 0, 1, 2, \ldots$ until convergence:

1. Select a step size $\mu > 0$ and compute

$$\hat{X} = X^{(k)} + \mu \left(\tilde{X} - X^{(k)} + \left(\tilde{D} - X^{(k)}(I - T^{(k)}) \right) (I - T^{(k)})^\top \right),$$

$$\hat{T} = T^{(k)} - \mu (X^{(k)})^\top \left(\tilde{D} - X^{(k)}(I - T^{(k)}) \right);$$

2. Update

$$X^{(k+1)} = \mathcal{P}(\hat{X}),$$

$$T^{(k+1)} = \begin{cases} \mathcal{P}(\hat{T}), & \text{if } \rho(\mathcal{P}(\hat{T})) \leq 1, \\ \mathcal{P}(\hat{T})/\rho(\mathcal{P}(\hat{T})), & \text{if } \rho(\mathcal{P}(\hat{T})) > 1. \end{cases}$$

Likewise, the following hybrid scheme for solving the constrained problem (2.5) is referred to as Algorithm G2. The basic idea in Algorithm G2 is that the implicitly defined matrix $D = X(I - T)$ should be nonnegative itself. We thus employ the steepest descent method with the correction of all three matrices D, X, and T to the nearest nonnegative matrices.

ALGORITHM G2. Given initial value $(X^{(0)}, T^{(0)})$ that is strictly feasible, do the following for $k = 0, 1, 2, \ldots$ until convergence:

1. Compute $D^{(k)} = \mathcal{P}(X^{(k)}(I - T^{(k)}))$.
2. Select a step size $\mu > 0$ and compute

$$\hat{X} = X^{(k)} + \mu \left(\tilde{X} - X^{(k)} + (\tilde{D} - D^{(k)})(I - T^{(k)})^\top \right),$$

$$\hat{T} = T^{(k)} - \mu (X^{(k)})^\top (\tilde{D} - D^{(k)});$$

3. Update

$$X^{(k+1)} = \mathcal{P}(\hat{X}),$$

$$T^{(k+1)} = \mathcal{P}(\hat{T}).$$

With properly chosen μ, these iterations can be made into a descent method.

3.3. Penalty Approach. Thus far, we have been tackling the inverse problem head on. In this section, we explore the idea of approximately solving the inverse problem by solving its approximate problem. We describe two ways to formulate the approximate problem – the penalty function and the log barrier function.

The penalty function is to approximate the optimization problem (2.2) by the new problem

$$\min_{\substack{X, D \in \mathbb{R}_+^{p \times n}, \\ T \in \mathbb{R}_+^{n \times n}.}} \frac{1}{2} \left\{ \left\| \left[\tilde{X} - X | \tilde{D} - D \right] \right\|_F^2 + \alpha \| D - X(I - T) \|_F^2 \right\}, \tag{3.2}$$

where $\alpha \in \mathbb{R}$ represents the weight of the penalty. The idea is that, instead of enforcing the relationship $D = X(I - T)$ directly, we bring down the gap between D and $X(I - T)$ through the penalty term. The Leontief model (1.2) is only approximately satisfied. However, we gain the advantage that the constraints involve only simple bounds. The problem (3.2) can be solved by many existing nonlinear optimization techniques. Such an approach of employing existing software to the approximate problem (3.2) will be

referred to as Algorithm P1 for problem (2.5). Note that the size of the problem grows as $n(n + 2p)$. Any Algorithm P1 involving second order derivatives will become expensive when n is large.

On the other hand, it might be much easier to apply the steepest descent method with projection to problem (3.2). The following Algorithm P2 is a projected gradient method for the approximate problem.

ALGORITHM P2. Given initial value $(X^{(0)}, T^{(0)}, D^{(0)})$ that is strictly feasible, do the following for $k = 0, 1, 2, \dots$ until convergence:
1. Select a step size $\mu > 0$ and compute

$$\hat{X} = X^{(k)} + \mu(\tilde{X} - X^{(k)} + \alpha(D^{(k)} - X^{(k)}(I - T^{(k)}))(I - T^{(k)})^\top),$$
$$\hat{T} = T^{(k)} - \mu\alpha(X^{(k)})^\top(D^{(k)} - X^{(k)}(I - T^{(k)})),$$
$$\hat{D} = D^{(k)} + \mu(D - D^{(k)});$$

2. Update

$$X^{(k+1)} = \mathcal{P}(\hat{X}),$$
$$T^{(k+1)} = \mathcal{P}(\hat{T}),$$
$$D^{(k+1)} = \mathcal{P}(\hat{D}).$$

In a similar spirit, we could also approximate problem (2.4) by the log barrier function [3, 5]. The logarithmic barrier function can be formulated as

$$f(X, T; \mu) = \frac{1}{2}\left(\|\tilde{X} - X\|_F^2 + \|\tilde{D} - X(I - T)\|_F^2\right) - \gamma\left(\ln(1 - \rho(T))\right) \tag{3.3}$$

where $\gamma > 0$ is the barrier parameter. The basic idea in the log barrier algorithm is to apply the Newton procedure with a fixed value of μ to find a minimizer $(X(\gamma), T(\gamma))$ of (3.3) where $X \in \mathbb{R}_+^{p \times n}$ and $T \in \mathbb{R}_+^{n \times n}$. The approximate minimizer of $f(X, T; \gamma)$ for one value of γ is then used as a starting point for the Newton procedure at the next reduced value of γ until γ converges to zero. To accomplish that goal, we point out that the gradient ∇f for each fixed γ can be calculated as follows:

$$\frac{\partial f}{\partial X} = -\left((\tilde{X} - X) + (\tilde{D} - X(I - T))(I - T)^\top\right), \tag{3.4}$$

$$\frac{\partial f}{\partial T} = X^\top(\tilde{D} - X(I - T)) + \frac{\gamma}{1 - \rho(T)}\mathbf{u}\mathbf{v}^\top. \tag{3.5}$$

Such an approach is referred to as Algorithm P3.

4. Numerical Experiment. We have outlined seven numerical methods classified into three categories for the inverse Leontief problem. In this section, we carry out some numerical experiments on the various techniques and report our test results. Because these methods have different features and wide-ranging degrees of complexities, they perform differently. It is not easy to make a fair comparison of their performance. In all these experiments, we use the CPU time and the attainable objective values on a singly devoted machine to evaluate the performance.

For Algorithms G1, G2 and P2, the step size is set at $\mu = 10^{-3}$ and we assume to have reached convergence when $\|[X^{(k+1)}|T^{(k+1)}] - [X^{(k)}|T^{(k)}]\|_F \leq 10^{-8}$. The routine FMINCON in the Matlab environment is used as the main driver in Algorithms C1, C2, P1 and P3, where the tolerance parameters are set at $TolX = 10^{-10}$, $TolFun = 10^{-10}$ and the maximal number of iterations are set at $MaxFunEvals = 10^5$ and $MaxIter = 10^4$. For Algorithms P1 and P2, we set $\alpha = 1$; For Algorithm P3, the value of γ is initially set at 1 and is decreased linearly with a slope of 1/50 until it reaches the minimum value $\frac{10^{-8}}{n(n+p+1)}$.

We carry out the experiments on three sets of test data for different purposes. The first two examples are based on artificially generated data. The purpose of these two experiments is to compare the performance

of the algorithms. The third example is based on real-world data and it shows an optimal consumption matrix which is surprisingly different from what has been observed in the market.

Example 1. We generate the first set of test data according to the following rules:

(a) Randomly generate an irreducible nonnegative matrix $T_* \in \mathbb{R}^{n \times n}$. Choose an arbitrary positive scalar $0 < \beta < 1$ and define $T := (\beta/\rho(T_*))T_*$. The matrix T, with a controlled spectral radius $\rho(T) = \beta$, is used as the "correct" consumption matrix in the Leontief model. In our experiments, we take $\beta = 0.8$.

(b) Randomly generate a nonnegative matrix $\tilde{D}_{obs} \in \mathbb{R}_+^{n \times p}$. This matrix is regarded as the observed demand matrix.

(c) Define $\tilde{X}_{obs} = (I - T)^{-1}\tilde{D}_{obs}$. This matrix is regarded as the observed output matrix.

We perform tests on problems of sizes $n = 3\ell$ and $p = 5\ell$ with $\ell = 1, 2, 3$. For each given and fixed size, the same starting points $(X^{(0)}, T^{(0)})$ generated randomly are used for all methods.

The goal of the first experiment is to test how well the algorithms handle the inverse problem when no errors are introduced in the measurements. Let the observed data $\tilde{D}_1$ and $\tilde{T}_1$ be the "exact" data, that is,

$$\tilde{D}_1 := \tilde{D}_{obs},$$
$$\tilde{X}_1 := \tilde{X}_{obs}.$$

We expect to see that if $p \geq n$, then T should be uniquely and completely recovered. The attainable objective values and the CPU overhead are reported in Table 4.1 and Figure 4.1, respectively.

Alg.	$n^2 + np$		
	24	96	216
C1	1.0212e-11	2.1455e-10	1.0581e-7
G1	1.5482e-9	2.1868e-9	7.4370e-10
P1	2.3285e-11	N/A	7.2634e-9
C2	2.2318e-12	2.6071e-8	5.3670e-10
G2	1.5482e-9	2.1868e-9	7.4548e-10
P2	1.5384e-9	2.2027e-9	8.0719e-10
P3	5.4504e-9	5.3516e-9	3.1549e-10

TABLE 4.1
Objective values attained by the different algorithms in Example 1.

Note that the Matlab routine FMINCON may fail to converge within the maximum number of function evaluations or iterations. It is worth mentioning that overhead of the three algorithms G1, G2, and P2 seems to be incentive to the size of the problem.

Example 2. In this experiment, we subject both the demand matrix $\tilde{D}_1$ and the output matrix $\tilde{X}_1$ to perturbations. Our goal is to examine how much fidelity of the original consumption matrix T can be recovered. There are two ways to introduce perturbations — errors that are stochastic entry-wise or errors that are bounded within a certain rectangle. For demonstration purpose, we consider only the following two cases.

Case 2.1. Let $\texttt{randn}(n, p)$ stand for an $n \times p$ random matrix whose entries are chosen from *iid* standard Gaussian distribution. Define the observed demand matrix and the output matrix by

$$\tilde{D}_2 := \tilde{D}_1 + \sigma_1 \texttt{randn}(n, p),$$
$$\tilde{X}_2 := \tilde{X}_1 + \sigma_2 \texttt{randn}(n, p),$$

respectively, where σ_k, $k = 1, 2$, represent the variance of the perturbations.
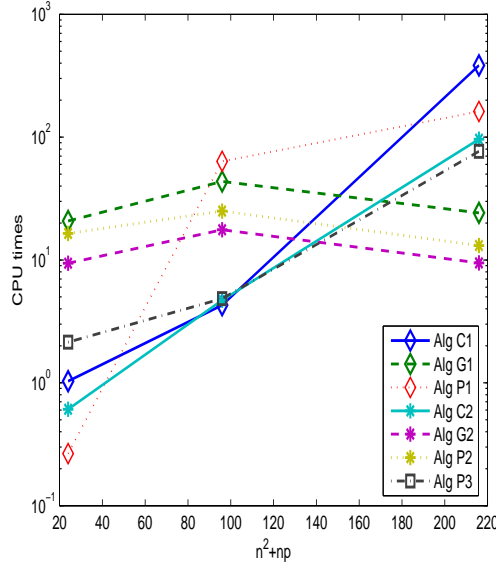
Fig. 4.1. *CPU time required by the different algorithms in Example 1.*

Case 2.2. Let E stand for an arbitrary but fix matrix with unit Frobenius norm. Define the observed demand matrix and the output matrix by

$$\tilde{D}_3 := \tilde{D}_1 + \epsilon_1 E,$$
$$\tilde{X}_3 := \tilde{X}_1 + \epsilon_2 E,$$

respectively, where ϵ_k, $k = 1, 2$, represent the absolute magnitude of the perturbations.

Objective values corresponding to the choice $\epsilon_1 = \epsilon_2 = 0.05$ and $\sigma_1 = \sigma_2 = 0.05$ are reported in Table 4.2. The corresponding CPU overhead is reported in Figure 4.2.

All of these test results seem to suggest that among the seven proposed methods, Algorithms G1, G2 and P2 are most robust and insensitive to the size of the problem. The classical techniques C1 and C2 are effective for low dimensional problems, but are at least one order more expensive when the size becomes modestly large.

Alg.	$n^2 + np$ (Case 2.1)			$n^2 + np$ (Case 2.2)		
	24	96	216	24	96	216
C1	5.2807e-3	2.3707e-2	N/A	1.5167e-4	N/A	N/A
G1	5.2807e-3	2.3707e-2	7.4340e-2	1.5167e-4	6.3775e-5	3.3688e-5
P1	5.2777e-3	2.3312e-2	7.3879e-2	1.4145e-4	6.3139e-5	3.3701e-5
C2	5.2807e-3	2.3707e-2	7.4463e-2	1.5167e-4	6.3775e-5	3.3688e-5
G2	5.2807e-3	2.3707e-2	7.4694e-2	1.5167e-4	6.3775e-5	3.3688e-5
P2	5.2807e-3	2.3312e-2	7.3885e-2	1.5167e-4	6.3775e-5	3.3688e-5
P3	5.2807e-3	2.3707e-2	7.4340e-2	1.5167e-4	6.3780e-5	3.3688e-5

TABLE 4.2
Objective values obtained by the different algorithms in Example 2.

Example 3. The BEA began to collect annual industry accounts in 1998 based on a comprehensive revision NAICS of the classification system. The database they have established contains massively detailed
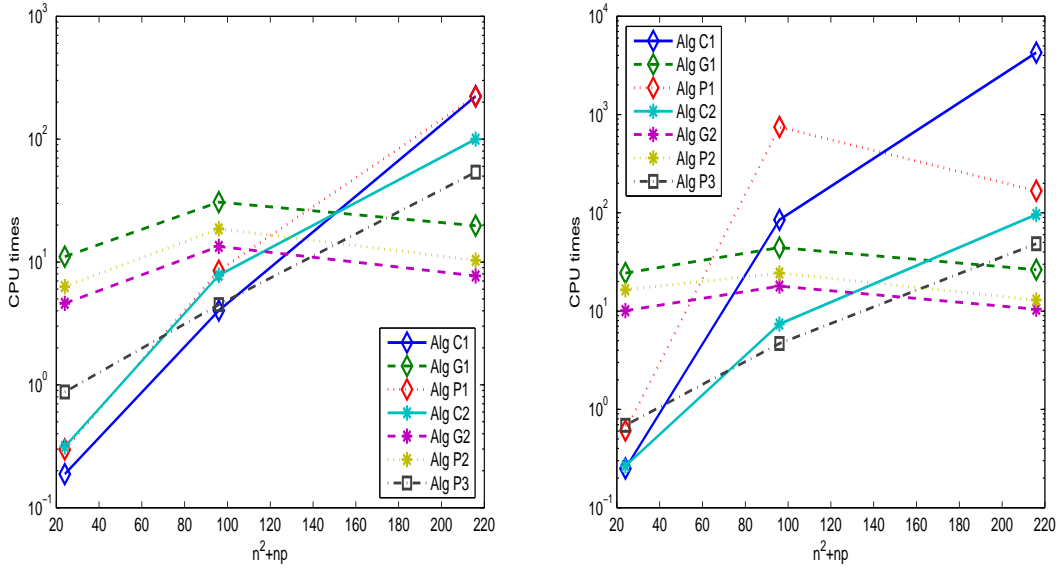
FIG. 4.2. *CPU time required by the different algorithms in Case 2.1 (left) and Case 2.2 (right).*

information. A full-scope input-output table will be too large to fit in this paper. It is nice that the BEA offers an interactive access to the input-output accounts data [1] by which users can retrieve information from a partial list of the sectors. For demonstrate purpose, we have selected five industries to study the interrelationship, these are, (01) Agriculture, forestry, fishing, and hunting; (04) Construction; (05) Manufacturing; (08) Transportation and warehousing; and (10) Finance, insurance, real estate, rental and leasing. (Numbers in the parentheses are the IO codes used in the tables.)

To avoid possible incompatibility in the definition of classification, we examine the BEA data of the six years from 1998 to 2003 only when the newer NAICS is used. Listed in Table 4.3 are the Use Tables, that is, the consumption matrices, for the five selected industries identified by their IO codes. For comparison, the 6×1 column in each cell represents the values t_{ij} of the corresponding sectors i and j over the six years. The points we want to make in this table are twofold: one is that these values represent only estimates of the true yet unknown input-output flows and the other is that the estimates in each cell over the span of six years are approximately the same. Could it be possible that the slight variations in each cell are due to input-output measurement errors whereas the consumption matrix should have been constant (like what has been assumed quinquennially before 1998), or are these variations representing true economic growth? In either case, *it would be interesting to see how the notion of our inverse problem would suggest about the consumption matrix.*

To conduct our experiment, we gather the total industry output from the six years' Use tables. The output vectors of the five selected industries for each year is listed as a row in Table 4.4. Because we are not referring to the entire spectrum of industries, the demand vector of each industry in our context is defined to be the sum of "Intermediate Not Selected" and "Sum of Value Added Not Selected". This information is listed in Table 4.5. These data are of large scale. To improve the computational efficiency, we work on the modified matrices $\tilde{X}_N$ and $\tilde{D}_N$ obtained by scaling data in Table 4.4 and Table 4.5 by the mean value of the entire data.

We use $(\tilde{X}_N, T^{(2002)})$, where $T^{(2002)}$ denotes the consumption matrix associated with year 2002, as the starting point for our iteration. For each of the seven methods, we summarize the computed total least squares output matrix X in Table 4.6 and demand matrix D in Table 4.7. The results by any of the

11

Commodities	Industries				
	01	04	05	08	10
01	0.223	0.001	0.039	0	0
	0.237	0.001	0.036	0	0
	0.230	0.001	0.035	0	0
	0.231	0.001	0.037	0	0
	0.226	0.001	0.036	0	0
	0.225	0.001	0.038	0	0.001
04	0.003	0.001	0.002	0.003	0.009
	0.003	0.001	0.002	0.003	0.009
	0.004	0.001	0.002	0.002	0.008
	0.004	0.001	0.002	0.002	0.009
	0.004	0.001	0.002	0.003	0.009
	0.003	0.001	0.002	0.003	0.010
05	0.180	0.268	0.360	0.099	0.016
	0.186	0.263	0.357	0.103	0.016
	0.183	0.258	0.347	0.122	0.016
	0.183	0.256	0.338	0.112	0.014
	0.185	0.247	0.330	0.105	0.014
	0.173	0.249	0.321	0.106	0.014
08	0.028	0.019	0.033	0.120	0.009
	0.029	0.018	0.031	0.120	0.009
	0.029	0.018	0.030	0.118	0.010
	0.029	0.018	0.030	0.121	0.008
	0.031	0.018	0.031	0.122	0.007
	0.028	0.018	0.030	0.116	0.007
10	0.073	0.031	0.025	0.057	0.186
	0.065	0.030	0.026	0.054	0.194
	0.064	0.032	0.026	0.051	0.200
	0.066	0.033	0.028	0.053	0.185
	0.055	0.032	0.026	0.053	0.178
	0.058	0.035	0.026	0.053	0.182

TABLE 4.3

"Use Table" of five selected industries from BEA input-output account data for years 1998 to 2003.

year	Industries				
	01	04	05	08	10
1998	257843.2	824368.1	3801739.6	530616.8	2558430.0
1999	250549.4	890481.4	3943198.0	559076.7	2765987.6
2000	254252.9	957442.8	4118977.1	593706.4	3037986.9
2001	258307.2	1000890.8	3852737.6	571210.7	3102808.3
2002	249564.5	1009983.4	3797031.8	571512.9	3208738.0
2003	275057.9	1063284.8	3890415.9	592871.8	3394219.2

TABLE 4.4

Total output vectors of five selected industries for year 1998 to 2003.

year	Industries				
	01	04	05	08	10
1998	50557.3	790038.4	20.69769.7	295677.8	1911555.0
1999	47004.6	855243.9	2155880.6	322589.2	2054120.6
2000	48892.8	920437.8	2273986.1	347418.3	2246536.3
2001	52263.5	962474.6	2138755.1	336973.9	2341720.7
2002	53164.2	969264.0	2142228.8	337839.9	2463951.6
2003	62553.7	1019074.4	2216503.6	357648.2	2590789.8

TABLE 4.5

Total demand vectors of five selected industries for year 1998 to 2003.
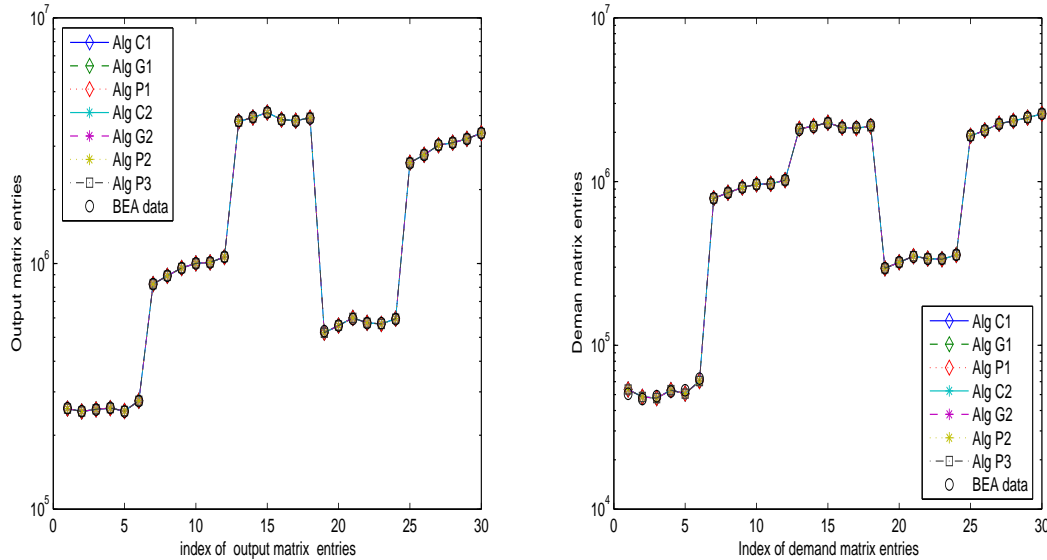

FIG. 4.3. *Comparison of the computed output matrices (left) and demand matrices (right) by the different algorithms.*

seven methods described in this paper point to almost the same optimal solution expect perhaps the last two digits which are within the stopping criteria after scaling. The logarithmic plots of these matrices in Figure 4.3 clearly demonstrate this closeness of total least squares solutions to the original data.

What is interesting and deserves further attention is the corresponding computed consumption matrices T recorded in Table 4.6. Note that these computed consumption matrices by the seven methods are almost identical, confirming that our calculation should be correct. However, these matrices are remarkably different from those consumption matrices in the User Table 4.3 which are obtained through arduous undertakings. An explanation of this discrepancy from an economics point of view is yet to be done. The meaning of this optimal consumption matrix is yet to be understood. We point out that the objective value has been reduced from the initial order of approximately 10^{10} to the order of 10^9 at convergence, which we think is quite significant.

5. Conclusion. The inverse problem of estimating the consumption matrix from output vectors and demand vectors has been considered in this paper. Assuming that the measurements usually are inexact, we employ the notion of total least squares to remove the noise while performing the estimation. A total of seven numerical schemes has been proposed as means to solve the inverse problem. Numerical experiments

seem to suggest that the projected gradient approach is the most robust and problem-size insensitive.

As a mathematical result when our methods are applied to some real-world EBA data, we discover a consumption matrix that is very different from the User Table collected in the EBA database. An economical interpretation of this result is beyond our expertise. It would be interesting to see if the estimated consumption matrix by our inverse formulation has any useful meaning to the economics community.

Alg.	X					T				
C1	256373.6	824042.8	3784175.5	522545.0	2567582.2	0.545	0.027	0	0	0
	249885.2	890653.9	3928805.5	558948.3	2767065.4	0	0	0	0	0
	254809.5	958159.2	4113402.9	599945.2	3028375.3	0.017	0	0.445	0.050	0
	257808.5	1001315.2	3852490.2	572977.8	3099403.3	0	0	0	0.070	0.865
	250664.1	1009756.5	3811192.9	567412.2	3215452.0	0	0.010	0	0	0.083
	275978.6	1062575.7	3914156.6	595990.1	3391741.2					
G1	256363.6	824029.4	3784176.4	522574.9	2567581.7	0.543	0.026	0	0	0
	249889.0	890648.5	3928805.3	558964.6	2767060.6	0	0	0	0	0
	254821.3	958160.0	4113401.7	599947.0	3028369.9	0.017	0	0.445	0.051	0
	257807.4	1001313.8	3852490.9	572967.5	3099398.1	0	0	0	0.068	0.864
	250667.0	1009768.8	3811191.6	567404.7	3215461.6	0	0.010	0	0	0.083
	275969.8	1062581.4	3914157.1	595967.4	3391744.7					
P1	256370.3	824051.5	3784177.4	522546.0	2567580.1	0.544	0.027	0	0	0
	249886.2	890656.8	3928805.6	558948.3	2767065.9	0	0	0	0	0
	254812.2	958160.6	4113401.7	599944.8	3028375.3	0.017	0	0.445	0.050	0
	257811.1	1001312.1	3852489.4	572978.6	3099399.9	0	0	0	0.07	0.860
	250665.3	1009757.8	3811191.6	567411.9	3215452.9	0	0.01	0	0	0.080
	275977.3	1062575.0	3914156.6	595989.6	3391742.3					
C2	256371.1	824050.6	3784177.2	522545.2	2567580.0	0.544	0.027	0	0	0
	249886.6	890657.8	3928806.5	558946.4	2767063.3	0	0	0	0	0
	254813.0	958160.6	4113401.4	599945.3	3028378.3	0.017	0	0.445	0.050	0
	257810.8	1001309.1	3852490.2	572976.4	3099399.7	0	0	0	0.070	0.865
	250666.2	1009757.1	3811191.9	567411.2	3215454.5	0	0.010	0	0	0.083
	275977.1	1062576.5	3914156.0	595987.1	3391742.3					
G2	256363.6	824029.4	3784176.4	522574.9	2567581.7	0.543	0.026	0	0	0
	249889.0	890648.5	3928805.3	558964.6	2767060.6	0	0	0	0	0
	254821.3	958160.0	4113401.7	599947.0	3028369.9	0.017	0	0.445	0.051	0
	257807.4	1001313.8	3852490.9	572967.5	3099398.1	0	0	0	0.068	0.864
	250667.0	1009768.8	3811191.6	567404.7	3215461.6	0	0.010	0	0	0.083
	275969.8	1062581.4	3914157.1	595967.4	3391744.7					
P2	256363.6	824029.4	3784176.4	522574.9	2567581.7	0.543	0.026	0	0	0
	249889.0	890648.5	3928805.3	558964.6	2767060.6	0	0	0	0	0
	254821.3	958160.0	4113401.7	599947.0	3028369.9	0.017	0	0.445	0.051	0
	257807.4	1001313.8	3852490.9	572967.5	3099398.1	0	0	0	0.068	0.864
	250667.0	1009768.8	3811191.6	567404.7	3215461.6	0	0.010	0	0	0.083
	275969.8	1062581.4	3914157.1	595967.4	3391744.7					
P3	256365.2	824007.4	3784176.8	522563.3	2567582.7	0.544	0.025	0	0	0
	249888.3	890640.4	3928805.7	558958.0	2767063.3	0	0	0	0	0
	254819.4	958161.3	4113401.8	599946.8	3028373.4	0.017	0	0.445	0.051	0
	257808.5	1001317.4	3852490.6	572971.6	3099398.3	0	0	0	0.069	0.865
	250666.1	1009781.3	3811191.6	567408.5	3215457.5	0	0.011	0	0	0.083
	275971.2	1062588.5	3914156.6	595976.4	3391742.2					

Table 4.6

Total least squares solutions of output matrix X and consumption T based on BEA data for year 1998 to 2003.

Alg.	D=X(I-T)				
	53809.6	790358.6	2101437.9	295069.8	1901577.4
	48451.6	855069.9	2181754.2	321630.4	2052944.6
C1	47624.8	919721.8	2284265.3	350446.9	2257010.7
	53327.3	962057.2	2139374.5	338530.2	2345439.3
	50761.2	969484.8	2116441.2	335437.4	2456625.4
	60574.1	1019781.8	2173619.3	356822.2	2593500.3
	53816.3	790377.1	2101437.5	295050.1	1901573.5
	48440.9	855076.8	2181753.2	321621.7	2052947.9
G1	47607	919720.6	2284263.7	350446.3	2257021.1
	53333.4	962051.6	2139374.1	338535.5	2345436.7
	50762.7	969478.6	2116439.7	335446.3	2456618
	60597.7	1019777.8	2173618.7	356837.6	2593497.7
	53811.0	790359.5	2101436.6	295071.2	1901571.1
	48451.2	855069.3	2181751.9	321630.7	2052941.3
P1	47623.5	919723.0	2284262.1	350446.5	2257007.3
	53330.6	962054.4	2139371.8	338530.6	2345433.1
	50762.9	969489.6	2116438.2	335436.9	2456625.0
	60579.1	1019782.7	2173616.9	356821.1	2593500.2
	53811.0	790358.4	2101438.2	295072.8	1901571.6
	48449.0	855070.3	2181754.1	321630.8	2052940.5
C2	47620.5	919723.2	2284263.8	350448.6	2257009.4
	53329.7	962051.8	2139373.9	338529.7	2345434.3
	50762.0	969489.3	2116440.1	335437.4	2456626.3
	60580.0	1019784.6	2173618.3	356819.7	2593501.6
	53816.3	790377.1	2101437.5	295050.1	1901573.5
	48440.9	855076.8	2181753.2	321621.7	2052947.9
G2	47607.0	919720.6	2284263.7	350446.3	2257021.1
	53333.4	962051.6	2139374.1	338535.5	2345436.7
	50762.7	969478.6	2116439.7	335446.3	2456618.0
	60597.7	1019777.8	2173618.7	356837.6	2593497.7
	53816.3	790377.1	2101437.5	295050.1	1901573.5
	48440.9	855076.8	2181753.2	321621.7	2052947.9
P2	47607.0	919720.6	2284263.7	350446.3	2257021.1
	53333.4	962051.6	2139374.1	338535.5	2345436.7
	50762.7	969478.6	2116439.7	335446.3	2456618.0
	60597.7	1019777.8	2173618.7	356837.6	2593497.7
	53815.1	790398.1	2101437.7	295057.6	1901572.7
	48443.7	855085.3	2181753.4	321624.8	2052944.5
P3	47611.6	919719.0	2284263.7	350446.9	2257015.6
	53333.1	962048.8	2139373.9	338533.4	2345436.9
	50762.8	969465.7	2116439.7	335443.5	2456623.0
	60593.1	1019771.1	2173618.4	356831.6	2593501.2

TABLE 4.7

Total least squares solutions of demand matrix D based on BEA data for year 1998 to 2003.

REFERENCES

[1] Bureau of Economic Analysis, Department of Commerce, Industry Economic Accounts Information Guide, http://www.bea.gov/bea/dn2/iedguide.htm.

[2] A. Berman and R. J. Plemmons, Nonnegative Matrices in the Mathematical Sciences, SIAM Publications, Philadelphia, 1994.

[3] S. Boyd and L. Vandenberghe, Convex Optimization, Cambridge University Press, Cambridge, 2004.

[4] E. Dietzenbacher, Estimation of the Leontief inverse from the practitioner's point of view, Math. Social Sci., 16(1988), 181-187.

[5] A. Fiacco and G. McCormick, Nonlinear Programming, SIAM, Philadephia, PA, 1968.

[6] R. Fletcher, Practical Methods of Optimization, 2nd ed., Wiley, Chichester, 1987.

[7] C. Johnson and R. Smith, The completion problem for M-matrices and inverse M-matrices, Linear Algebra Appl., 241/243(1996), 655-667.

[8] S. Lahiri and S. Satchell, Underestimation and overestimation of the Leontief inverse revisited, Economics Letters, 18(1985), 181-186.

[9] W. Leontief, Input-output economics, Scientific American, October, 1951, 15-21.

[10] W. Leontief, Dynamic Analysis: In Studies in the Structure of the American Economy. Eds. W. Leontief, et al. New York: Oxford University Press, 1953, 53-90.

[11] W. Leontief, Input-Output Economics, Oxford University Press, New York, 1966.

[12] M. Ruth and B. Hannon, Modeling Dynamic Economic Systems, Springer-Verlag, New York, 1997.

[13] S. Van Huffel and J. Vandewalle, The Total Least Squares Problem: Computational Aspects and Analysis, SIAM Publications, Philadelphia, 1991.

[14] S. Wright, On the convergence of the Newton/log-barrier method, Math. Program., 90(2001), 71-100.