COWLES FOUNDATION FOR RESEARCH IN ECONOMICS

AT YALE UNIVERSITY

Box 2125, Yale Station
New Haven, Connecticut 06520

COWLES FOUNDATION DISCUSSION PAPER NO. 837

Note: Cowles Foundation Discussion Papers are preliminary materials circulated to stimulate discussion and critical comment. Requests for single copies of a Paper will be filled by the Cowles Foundation within the limits of the supply. References in publications to Discussion Papers (other than acknowledgment that a writer had access to such unpublished material) should be cleared with the author to protect the tentative character of these papers.

IMPLEMENTATIONAL ISSUES AND COMPUTATIONAL PERFORMANCE SOLVING APPLIED GENERAL EQUILIBRIUM MODELS WITH SLCP

by

Thomas F. Rutherford

May 1987

IMPLEMENTATIONAL ISSUES AND COMPUTATIONAL PERFORMANCE SOLVING APPLIED GENERAL EQUILIBRIUM MODELS WITH SLCP

Abstract

This paper reports on an implementation of Mathiesen's sequential method for solving applied general equilibrium models. In this approach, the underlying nonlinear complementarity problem is solved by successive linearization. The paper discusses model formulation, implementation and performance. Several test problems and empirical models are used to evaluate efficiency and robustness.

Thomas F. Rutherford

Cowles Foundation for Research in Economics
Box 2125 Yale Station
New Haven, Ct. 06520

For presentation at TIMS/ORSA Joint National Meeting, May 4-6, 1987.

1. INTRODUCTION

This paper describes the implementation and performance of an algorithm for solving applied general equilibrium (AGE) models. These models are typically employed to study systems involving more than one economic agent, each of which may have a separate objective function. Models of international trade and single-country models which focus on public finance issues are commonly formulated in an AGE format.

Scarf (with Hansen) [1973] demonstrated the feasibility and potential of numerical modeling in the Arrow-Debreu general equilibrium framework. In Scarf's work constructive proofs of existence provided the building blocks for solution methods. Subsequent research has produced new approaches to model formulation and methods of solution. Following along the lines of Robinson [1975], Hogan [1977], Eaves [1978], and Josephy [1979], Mathiesen [1985a] proposed a modeling format and sequential method for solving market equilibrium models. The method is named SLCP. Preckel [1983] compared SLCP with other methods and reported a considerable advantage in terms of efficiency. This paper discusses implementational issues and reports on computational experience with several test problems and empirical models.

The software described in this paper is named MPS/GE (a "mathematical programming system for general equilibrium models"). MPS/GE is a microcomputer-based system designed to reduce the technical expertise required for formulating and analyzing AGE models. The system is written in Fortran-77 with some assembly support for interactive modules.

Solution codes for AGE models[1] typically require that the model structure be provided in the form of a computer subroutine. With this arrangement, AGE modeling requires close familiarity with the solution algorithm. In contrast, MPS/GE separates the tasks of model formulation and model solution. This frees model builders from the tedious task of writing model-specific function evaluation subroutines. All features of a particular model are communicated to MPS/GE either interactively or through an input data file.

To simplify coding, utility and production functions are restricted to the "nested" constant elasticity of substitution (CES) family. Two special cases: Leontief (fixed coefficient) and Cobb-Douglas are included. These nested CES functions are characterized by different trade-off possibilities within each aggregate as well as between aggregates. Because functions are entered in a data file, revisions of model structure are simplified.

In contrast to the nonlinear programming language GAMS[2],

---

[1] See, for example, Broadie [1983a,b], Todd [1980], Kimball and Harrison [1985], Merrill [1972], and Mathiesen [1985a].

[2] See Meeraus [1983]. For a limited class of market structures, an extension of GAMS (HERCULES) can solve economywide models. See Drud, Kendrick and Meeraus [1986].

Pearson [1986] has also developed a higher-level language for economic modeling in the Johansen framework.

MPS/GE does not altogether eliminate the need for programming. In large-scale projects, model-specific programs may be used in order to avoid the tedium of entering function coefficients one element at a time. When "model-generator programs" are required, they be written in whatever language is most familiar. Furthermore, these programs operate "stand-alone", thereby simplifying development and debugging.

The MPS/GE system consists of three programs: a solver, a model editor and a case generator. This paper focuses exclusively on the solution code. Details on the other components can be found in Rutherford [1987].

2. FORMULATION

Mathiesen [1985a] observed that applied general equilibrium models can be formulated in nonlinear complementarity (NLCP) format. The general NLCP is stated:

Given $F: \mathbb{R}^N \to \mathbb{R}^N$

Find $z \in \mathbb{R}^N$, $z \geq 0$ such that $F(z) \geq 0$ $\quad z' F(z) = 0$

As formulated in MPS/GE, the unknown vector z represents activity levels (y), prices (π), auxiliary variables (μ) and income levels (Y). That is, z is partitioned into (y, π, μ, Y) where $y \in \mathbb{R}^m$, $\pi \in \mathbb{R}^n$, $\mu \in \mathbb{R}^L$, $Y \in \mathbb{R}^p$ and $N = n + m + L + p$. The nonlinear function F represents unit profit-, market excess demand-, auxiliary constraint-, and income-functions.

This formulation includes income variables in order to simplify the treatment of ad-valorem taxes on producers which enter into consumer incomes. When there are endogenous income flows, the addition of income equations can reduce the number of nonzeros in the linearization. Furthermore, as observed by Shoven and Whalley [1979], the income variables are useful in avoiding simultaneity which otherwise arises in the computation of consumer demands.

For concreteness, consider a model with an activity-analysis description of production, Cobb-Douglas utility functions, and no auxiliary constraints. The vector function F is defined as:

$$F(y,\pi,Y) = \begin{bmatrix} & -A'\pi & \\ A\,y & & -\xi(\pi,Y) \\ & b\,\pi - Y & \end{bmatrix}$$

where $\xi(\pi,Y):\mathbb{R}^{n\times p}\rightarrow\mathbb{R}^{n}$ is the aggregate excess demand function, derived from utility maximization subject to income constraints:

$$\xi_i(\pi,Y) = \sum_k \left[\frac{\alpha_{ik} \quad Y_k}{\pi_i} - b_{ki} \right] ;$$

$A\in\mathbb{R}^{n\times m}$ is the activity analysis production matrix, $\alpha\in\mathbb{R}^{p\times n}$ is the matrix of elasticities of utility with respect to consumption, and $b\in\mathbb{R}^{p\times n}$ is a matrix of commodity endowments.

To solve the nonlinear complementarity problem, SLCP repeatedly solves linear complementarity problems (LCPs). These are defined as follows:

Given $q\in\mathbb{R}^{N}$, $M\in\mathbb{R}^{N\times N}$

Find $z \in \mathbb{R}^N$, $z \geq 0$ such that

$$q + M z \geq 0 \quad z'(q + Mz) = 0$$

The LCP is a fundamental problem of mathematical programming. It contains the linear- and quadratic programs as special cases. (See Cottle and Dantzig [1968].)

The data for SLCP subproblems are determined by taking a first-order Taylor expansion of the nonlinear function, F. Linearized at point $\bar{z}$, they are:

$$q(\bar{z}) = \nabla F(\bar{z})\, \bar{z} - F(\bar{z}) \qquad \text{and} \qquad M(\bar{z}) = \nabla F(\bar{z})$$

In the Cobb-Douglas - activity analysis example, these data are given by:

$$q(\bar{z}) = \begin{bmatrix} 0 \\ -\xi(\bar{\pi}, \bar{Y}) \\ 0 \end{bmatrix}, \qquad M(\bar{z}) = \begin{bmatrix} 0 & -A' & 0 \\ A & -D(\bar{\pi}, \bar{Y}) & e(\bar{\pi}) \\ 0 & b & -I \end{bmatrix}$$

where $D \in \mathbb{R}^{n \times n}$ is a diagonal matrix[3] of own- and cross-price effects:

$$D_{ii} = -\sum_k \frac{\alpha_{ik}\, Y_k}{\pi_i^2} ;$$

$e \in \mathbb{R}^{n \times p}$ is a nonnegative matrix of income effects:

$$e_{ik} = \frac{\alpha_{ik}}{\pi_i} ;$$

[3] If more general CES utility or production functions are employed, D is not diagonal but remains postitive semi-definite.

and I is the pxp identity matrix.

MPS/GE, because it includes income variables explicitly, solves subproblems which differ slightly from those described by Mathiesen [1985b]. In his formulation, matrices e, b, and I do not appear, and income and price effects enter together in the Jacobian matrix, D.

3. ALGORITHMIC ISSUES

The SLCP algorithm is an extension of Newton's classical method for solving simultaneous nonlinear *equations* which accomodates both linear and nonlinear *inequalities*. Applied to general equilibrium models, SLCP involves the following steps:[4]

1. Select a starting point, z_0 and set the iteration counter k to zero.
2. Stipulate a numeraire commodity, index h.
3. Increment the iteration count ($k \leftarrow k + 1$).
4. Construct $q_h(z_{k-1})$ and $M_h(z_{k-1})$.
5. Apply Lemke's algorithm to solve the subproblem. If no solution exists or the problem fails to solve, select another numeraire index and repeat step 4.
6. Given the LCP solution, z, conduct a backtracking line search to determine the step length α_k and the next iterate:

$$z_k = \alpha_k z + (1-\alpha_k) z_{k-1},$$

In the process of conducting the search, determine the current deviation from equilibrium.

---

[4]. The same algorithm may be used for partial equilibrium models. In those cases, supply and demand functions are expressed in nominal terms and Walras' law does not apply.

7. If the deviation exceeds the convergence tolerance ε, repeat steps 3-7.

Several details must be considered in implementing the algorithm. Among these are the following.

Function and gradient evaluation

The need to evaluate both F(z) and the n x n matrix $\nabla F(z)$ is a drawback of the SLCP algorithm. While in theory, ∇F is obtained simply by applying the chain rule, it can be a tedious and error-prone calculation in practice. An appealing aspect of standard packages, such as MPS/GE and GAMS, is that they provide gradients automatically. GAMS' ability to parse and linearize arbitrary algebraic systems makes it a flexible modeling tool, whereas MPS/GE offers only a limited range of functional forms. A GAMS front-end to SLCP could be quite useful for the computation of both partial and general equilibrium models.

An advantage of MPS/GE's specialized functional forms is that the Jacobian can be generated inexpensively. *Simple* arithmetic operations such as addition and multiplication are less costly than *transcendental* operations such as exponentiation. In the case of nested CES utility and production functions, exponentiations are required to compute F, but (through judicious use of intermediate variables) ∇F can then be obtained using only simple operations. Ignoring programming costs, ∇F is therefore inexpensive to compute.

Choice of numeraire

Production and utility functions are homogeneous of degree zero, so the basis associated with the set of positive prices, activity levels and income variables becomes singular as the equilibrium point is approached. This reflects the fact that equilibrium conditions determine relative and not absolute prices.

MPS/GE overcomes this problem by fixing a price and dropping the corresponding market clearance constraint. Closure of income flows assures that the omitted constraint is satisfied at equilibrium. This technique explains the difference between (q,M) and (q_h, M_h). The latter LCP is of dimension N-1, derived from (q,M) by fixing π_h at unity and dropping the constraint for market h. The omitted constraint will, through Walras' law, automatically be satsified at equilibrium. (In subproblem solutions away from the equilibrium this constraint will not be satisfied.)

The iterative process is influenced by the choice of numeraire. A subproblem solution might be denoted $z(\bar{z},h)$, indicating that it depends both on the linearization point, $\bar{z}$, and the numeraire index, h. For computational efficiency, some prices are more effective numeraires than others. Foremost, the numeraire's price must be nonzero at equilibrium. In addition, experience suggests that the best numeraires are "important commodities", i.e. goods which command a large share of factor payments or final demand expenditures.

The freedom to specify a numeraire provides a recovery strategy when a particular subproblem fails to solve. When this happens, the algorithm *switches numeraires*, evaluates at the same point and continues. It is likely that improvements could be made in this procedure. An inexpensive method for ex-ante selection of a "good" numeraire would be particularly useful. Presently, a poor numeraire is only detected when Lemke's algorithm fails to compute a subproblem solution.

Stipulation of a numeraire is but one of several methods for dealing with linear dependence. One might, for example, perform the substitution: $\pi_h = 1 - \sum_i \pi_i$, but this would result in an unacceptable increase in density. Other approaches are proposed by Stone [1985].

Measuring deviation

The deviation associated with a point z is taken to be:

$$\delta(z) = \max_i \; (-z_i, \; - F(z)_i, \; | \, z_i \, F(z)_i \, | \;)$$

This measure is affected by scaling of prices and activity levels. This is but one of several norms which could be employed. It would be useful if a norm could be designed which would guarantee a decrease in each iteration for a sufficiently short step length.

Subproblem solution

Lemke's algorithm[5] is used to solve subproblems. It performs two functions: it identifies a feasible, complementary basic set, and it solves the corresponding system of equations.

---

[5] See Lemke [1965] and Cottle and Dantzig [1968].

Observe that the generic linear complementarity problem (q,M) may be stated as follows:

Given $q \in \Re^N$, $M \in \Re^{N \times N}$

Find $z \in \Re^N$, $w \in \Re^N$ such that

$$w = q + M\,z \qquad z'\,w = 0 \qquad z \geq 0 \qquad w \geq 0$$

In this formulation, w may be regarded as a vector of slack variables.

Lemke's algorithm maintains a complementary basis throughout. That is, at most one of each pair of complements (w_i, z_i) is positive. Typically the algorithm is initiated "at the origin", with the complementary basis $w = q$. If this solution is infeasible, an artificial variable and column are added, producing a tableau of the following form:

w	z	z_0		RHS
I	$-M$	a	$=$	q

The *artificial column*, $a \in \Re^n$, contains elements a_i which are -1 when q_i is negative and zero otherwise. The *artificial variable*, z_0, enters the basis on the first pivot, initiating a sequence of pivots. A combinatorial rule determines variables entering the basis. In each pivot, the *complement* of the variable which last left the basis is introduced. For example, if z_0 replaces w_k in the initial pivot, the next pivot introduces z_k. Analogous to the simplex method for linear programs, a min-ratio test is used to determine outgoing variables.

To accelerate Lemke's algorithm, our implementation initiates from a basis defined by the previous subproblem's solution. That is, rather than solve (q,M), Lemke's algorithm is applied to ($B^{-1}q$, $B^{-1}N$) where (I -M) is partitioned into (N B) according to an initial basis. By construction, the basis B is complementary. It may contain both slacks and structural columns. If B is a "nearly feasible", this technique often reduces the cost of solution. It provides significant savings in later iterations where the optimal basis stabilizes and a single factorization provides an LCP solution. In these iterations, SLCP iterates are equivalent to Newton steps.

In the initial iteration, a user-supplied starting point defines the set of prices and activities which are employed in the initial basis. In cases in which this basis is singular, Lemke's algorithm is initiated with a slack basis.

Lemke's algorithm concludes in one of two ways. If, at any point, z_0 is selected as the outgoing variable, the subproblem is solved. Alternatively, the algorithm can terminate *on a secondary ray*. This describes a situation in which there are no feasible pivot candidates in an incoming column. This can and does occasionally happen with SLCP.

When a ray is encountered, the LCP is not abandoned. Instead, a recovery procedure is invoked which constructs a new artificial column with respect to the best basis encountered to that point. (The code maintains a copy of basis indices and updates these pointers whenever the artificial variable begins to increase from the minimum value which has yet been observed. In the event of a restart, the artificial column is discarded from this basis and one of the noncomplementary pair is inserted. This re-parameterizes Lemke's algorithm, presumably closer to a new solution than at the secondary ray.) In experiments reported below, an LCP is abandoned and a new numerire is installed only after the recovery routine has been invoked four times.

Data handling

Early implementations of SLCP represented all or part of ∇F in dense format. This simplified implementation, but it imposed limitations on the size of problems which could be processed. Typically, these matrices have desities on the order of 6% to 10%.

In MPS/GE, sparse data structures are used throughout. Elements of ∇F are computed and initially stored in a linked list with three parallel arrays, A(), IA() and JA(). These first two contain nonzeroes and row indices. The index array JA() sets up a linked list for storing data during computation of the Jacobian. After all coefficients have been loaded, they are sorted into the column-dominant order. The workspace occupied by JA() is then allocated for matrix factorization.

Basis factorization and maintenance

The factorization routine is a fundamental building block for direct numerical methods. The MPS/GE code benefits from a efficient and robust piece of software for computing and maintaining sparse basis factors. LUSOL (See Gill *et al.* [1986]) is used to factorize bases, solve systems of equations, and update the factorization through column replacement operations.

<u>Convergence</u>

There are two obstacles to convergence of SLCP. First, it can and does happen that a subproblem fails to solve. Second, and less often, the sequence of iterates might fail to converge. Rutherford [1982] and Mathiesen and Rutherford [1983] tested the SLCP algorithm with small examples and identified instances in which a subproblem $(q_h(\bar{z}),\ M_h(\bar{z}))$ failed to solve for some linearization point, $\bar{z}$, or numeraire, h. It seemed exceedingly rare that $(q_h(\bar{z}),\ M_h(\bar{z}))$ failed to solve for all h for a fixed $\bar{z}$.[6]

Failure for one numeraire slows but does not interrupt the iterative process. It is possible to try different numeraire indices until a solution is obtained. In large problems, however, this can be time consuming and it is not always successful.

Eaves [1978], Josephy [1979], Hogan and Ahn [1982] and Pang and Chan [1983] provide various conditions under which iterative algorithms such as SLCP will converge, either globally or when initiated in a neighborhood of the equilibrium. These convergence results do not appear to apply to the complete class of models representable with MPS/GE. For more on this topic, see

---

[6] Scarf's exchange model (Scarf [1960]) is a model in which, for <u>some</u> values of $\bar{z}$, $(q_h(\bar{z}), M_h(\bar{z}))$ fails *for all h*.

Mathiesen [1987].

4. NUMERICAL TESTS

A number of models have been implemented using MPS/GE. This section reports on convergence tests conducted with a few of these. Our objective is to evaluate the speed with which MPS/GE can solve AGE models containing various economic features. When formulating a new model, solution cost and feasibility are important considerations. They can govern the size of a model which can be undertaken and/or dictate the machine which must be used to solve it. From the standpoint of computational cost, the salient characteristics of MPS/GE models seem to be the following:

- Dimensions: the number of commodities, activities, consumers and institutional constraints.
- Density: the complexity of supply and demand functions, as evidenced by the number of nonzeros in the linearized system.
- Structure: determined by the types of constraints and the pattern of nonzeros in the linearization. For example, with linear programs, staircase structures are typically more difficult than diagonal or band matrices.

Before describing the test problems individually, we provide a broad overview of numerical results. Figures 1 and 2 indicate the solution times for all test problems included in this study. In the first figure, the horizontal axis relates problem dimension, measured as the number of production sectors, commodity markets, institutional constraints and consumers ($N = m + n + L + p$). The vertical axis of Figure 1 measures solution time in minutes elapsed, excluding input and output. All problems are solved to a tolerance of 10^{-5}. Indices appearing in Figure 1 correspond to problem numbers in Table 1.

Figure 2 measures problem size by the number of nonzeros in a subproblem linearization. This accounts for the density of supply and demand functions.

Figure 1: Dimension versus Compute Time

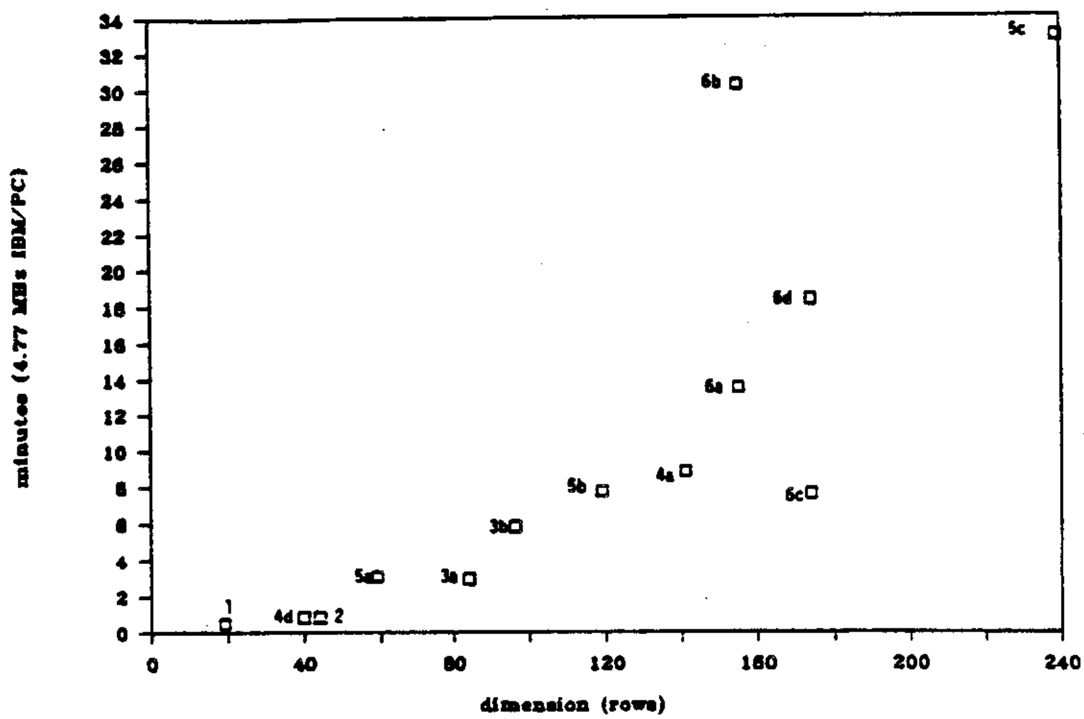


Figure 2: Nonzeros versus Compute Time

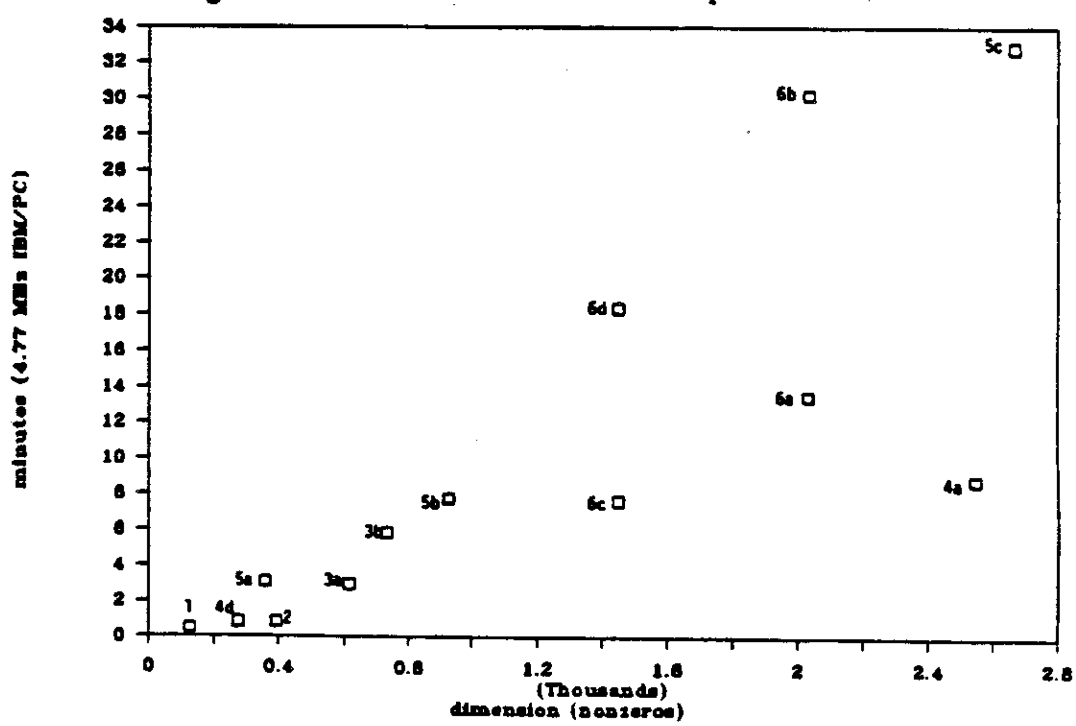


Table 1: Solution Statistics

Model	S	M	C	A	N	Itrs.	Time LCP	Time Total
	----- Dimensions ----					- Solution Stats -		
1. SCARF	8	6	5	0	126	3	:11	:28
2. HANSEN	26	14	4	0	393	3	:20	:49
3. GEN.V.T.								
a) 15 region	60	21	3	0	612	8	2:08	2:57
b) 18 region	72	24	3	0	729	6	5:03	5:48
4. GEMTAP								
a) 10%	62	65	14	0	2546	3	6:01	8:49
b) 20%	62	65	14	0	2546	4	8:01	10:56
c) 40%	62	65	14	0	2546	8	16:02	23:11
5. RAMSEY								
a) 10 period	19	39	1	0	356	11	1:44	3:04
b) 20 period	39	79	1	0	926	10	5:49	7:45
c) 40 period	79	159	1	0	2666	9	29:56	32:52
6. LTM								
a) 3RT-warm	63	83	3	6	2032	11	10:35	13:30
b) 3RT-cold	63	83	3	6	2032	13	23:26	30:14
c) 4RT-warm	72	98	4	0	1449	6	5:44	7:37
d) 4RT-cold	72	98	4	0	1449	12	15:38	18:23
7. CAMEROON	49	73	4	0	1026	5		6:49

Key:

S	Production sectors.	N	Nonzeros in $\nabla F(z)$.
M	Commodity markets.	Iters.	Iterations.
C	Consumers.	Time	minutes : seconds
A	Auxiliary constraints.		

All computations were conducted on a 4.77 MHz IBM/XT microcomputer, with 8087 coprocessor. Times exclude input and output. The convergence tolerance was .00001 for all cases.

4.1. Test Problems

<u>Scarf and Hansen</u>

These models have linear activity analysis description of production and single-level CES utility functions. Both are described in Scarf [1973].

<u>Generalized von Thünen</u>

This highly stylized spatial equilibrium model is originally due to von Thünen, a 19th century mathematical economist. His model was generalized in a partial equilibrium format by MacKinnon [1976]. Rowse [1981] and Mathiesen [1985] also report numerical results with the partial equilibrium model. Although it closely resembles MacKinnon's test problem, the model used here has *noncompensated* demand functions.

The model describes a closed agricultural economy in which there are c commodities grown on M tracts of land. Agricultural production requires inputs of land, labor and transportation. Discrete tracts of land, oriented in concentric rings, are centered on a town which serves as the point location of final demand. Cobb-Douglas production functions describe labor and land inputs to production. These functions vary between goods but are constant over regions. Transport requirements increase linearly with distance from the town and may vary between goods.

There are three consumers, distinguished by their endowments. *Workers* provide labor for production at all locations, *land owners* collect rents from each of the production rings, and *porters* provide a commodity entitled "transport".

Associated with each consumer is a Cobb-Douglas utility function including one or more of the agricultural goods. Workers value leisure, so the labor supply is elastic.

Algebraic parameters for this model include:

M	Number of regions.
c	Number of produced commodities.
L	Labor endowment. Labor supply equals L minus leisure demand. Leisure's budget share in workers' final demand equals $(1 - \sum_i \alpha_{i1})$.
T	Transport endowment.
d_j	Distance from the jth ring to town: $d_j = 5\ (2\ j - 1)$
a_j	Area of the jth production ring: $a_j = 2 \times 3.1415 \times d_j$
t_i	Transport requirements of the ith good, per unit distance transported. Unit transport inputs for the ith good from the jth region are then t_i times d_j.
A_i, β_i	Cobb-Douglas production function coefficients for the ith good. For labor inputs of x and land inputs of y, output equals: $f(x,y) = A_i\ x^{\beta_i}\ y^{1-\beta_i}$
α_{ik}	Budget share of the ith commodity by the kth consumer, k=1,2,3 representing workers, owners and porters.

In total, the model contains 3 consumers, M $\times$ n nested (Leontief:Cobb-Douglas) production functions, and n+M+2 commodities. Numerical values for these parameters are as follows:

Table 2: Numerical Parameters for von Thünen Model

dimensions: $c = 4$ $M = 15, 18$

endowments: $L = 30.$ $T = 20.$

technology:

	1	2	3	4
A_i	1.	2.	3.	4.
β_i	.9	.7	.5	.3
t_i^1	.015	.006	.004	.01

preferences:

	1	2	3	4
α (workers)	.2	.3	.1	.3
α (owners)	.3	.3	.2	.2
α (porters)	.6	.2	.1	.1

This model is a particularly challenging test case involving inequalities that are sometimes binding and sometimes not. For M sufficiently large, equilibrium rental rates on outer rings equal zero. Unit factor demands for land in the fallow tracts are infinite. The associated activities, however, are not operated because the farmgate price does not cover the cost of transport. For purposes of computation, prices are perturbed in order to evaluate producer responses.[7] These features can lead to numerical difficulties. For example, the magnitudes of matrix coefficients vary from 10^{-5} to 10^{6}. It is possible that scaling could overcome these problems. A first step might be to adopt

[7] That is, when evaluating functions, any factor price less than 10^{-7} is replaced by 10^{-7}. At equilibrium, no sector in which factor prices are perturbed is operated at postive intensity (cf. equilibrium activity levels in rings 17 and 18, Table 3).

the Harwell scaling routine used in LCPL. (See Tomlin [1976].)

SLCP does not process this model from an arbitrary starting point. The following heuristic has been used successfully in determining a good starting point:

a) Specify initial estimates of commodity prices (π_i), the wage rate (w), the price on transport (τ) and the price of land in region one (r_1). In the cases reported here, unity is specified for all prices.

b) Specify an estimate of γ for recursively computing the price of land in regions 2 to M, using the relation: $r_j = \gamma\, r_{j-1}$. In both cases reported here, a value of .5 is used for γ.

c) Compute the unit rate of profit for production of each good in each region:

$$\Pi_{ij} = \pi_i - \tau\, d_j\, t_i - C_i(w, r_j)$$

where $C_i(w, r_j)$ is the unit cost function for labor and land inputs:

$$C_i(w, r_j) = \frac{1}{A_i} \left(\frac{w}{\beta_i} \right)^{\beta_i} \left(\frac{r_j}{1 - \beta_i} \right)^{1-\beta_i}$$

d) Assign estimates of production levels based on the following rules:

$$Y_{ij} = \begin{cases} a_j \,/\, a_{Kij} & \text{if } \Pi_{ij} > 0 \text{ and } \Pi_{ij} \geq \Pi_{kj} \ \forall\ k \\ 0 & \text{otherwise,} \end{cases}$$

where a_{Kij} is the profit-minimizing input coefficient for land inputs:

$$a_{Kij} = \frac{\partial\ C_i(w, r_j)}{\partial\ r_j}$$

Equilibrium prices and activity levels for the 18 region version are displayed in Table 3. (The 15 region model has very nearly the same solution.)

MPS/GE fails to solve this model when the number of regions is increased from 18 to 20. In fact, the 18 region version very nearly does not solve: the second LCP encounters and recovers from three secondary rays before a solution is obtained. The difficulties in solving the first two LCPs explain the difference in run times (2:48 versus 5:57 minutes) between the 15 and the 18 region models.

Table 3: Equilibrium Values for von Thünen Model

π = (1., .580730, .330647, .413635) w = .611334 τ = .610758

Ring	r	Y_1	Y_2	Y_3	Y_4
1	2.032540	6.69174	0.	0.	0.
2	.740740	6.88784	0.	0.	.46128
3	.548514	0.	0.	0.	4.71679
4	.374762	0.	0.	0.	5.89040
5	.222589	0.	1.48949	0.	4.56510
6	.163663	0.	4.97181	0.	0.
7	.116631	0.	4.63521	1.09237	0.
8	.079986	0.	3.22987	4.83489	0.
9	.055672	0.	0.	4.33032	0.
10	.035752	0.	0.	3.59977	0.
11	.020224	0.	0.	1.78738	0.
12	.009090	0.	.44500	0.	0.
13	.003803	0.	.81172	0.	0.
14	.001165	0.	.38300	0.	0.
15	.000183	0.	.11262	0.	0.
16	.000002	0.	.00508	0.	0.
17	0.	0.	0.	0.	0.
18	0.	0.	0.	0.	0.

GEMTAP

This model of the U.S. economy and tax system has been used in a number of policy studies. (See Shoven and Whalley [1972] and Ballard *et al.* [1985].) The model contains 19 production sectors and associated sectoral commodities. There are 14 consumers and 16 consumption goods. The factors of production include labor and capital. There are numerous detailed tax distortions: ad valorem taxes on production and consumption, affine income tax schedules, and taxes on factor payments.

The model structure permits dimensionality reduction to be exploited. This was essential given the solution tools available at that time the model was conceived. Functional forms are specified so that three variables (two factor prices and government income) are sufficient to evaluate excess demands in all markets.

The MPS/GE implementation of GEMTAP does not take advantage of dimensionality reduction. The model is implemented with all sectoral and consumption goods explicitly represented. This leads to a larger system, but this simplified implementation and accomodates revisions in structure.

Using the factor price revision rule (Kimball and Harrison [1985]) the computational cost for this model is roughly 15 minutes. The computational cost under SLCP is typically faster, but it depends to a certain degree on the quality of the initial guess. In the runs reported here, we perturb prices from the benchmark levels. With a p% deviation, p% is added and subtracted from alternate initial values. As p increases, the starting point moves further from the equilibrium.

Ramsey

This growth model is a traditional test problem for nonlinear optimization. It concerns allocation of output between consumption and investment over a finite planning horizon in order to maximize an intertemporal utility function. The utility function is Cobb-Douglas:

$$U(C) = \sum_{t=1}^{T-1} \alpha^t \ln(C_t) + \sum_{t=T}^{\infty} \alpha^t \ln [C_T (1 + \gamma)^{t-T}]$$

Output in each period is determined by inputs of capital (K_t) and labor (L_t):

$$Y_t = A L_t^{\alpha} K_t^{1-\alpha} ,$$

and it must satisfy investment and consumption demands:

$$Y_t \geq I_t + C_t$$

To complete the formulation, there are equations which govern investment and capital accumulation:

$$K_t = K_{t-1} + I_{t-1} ;$$

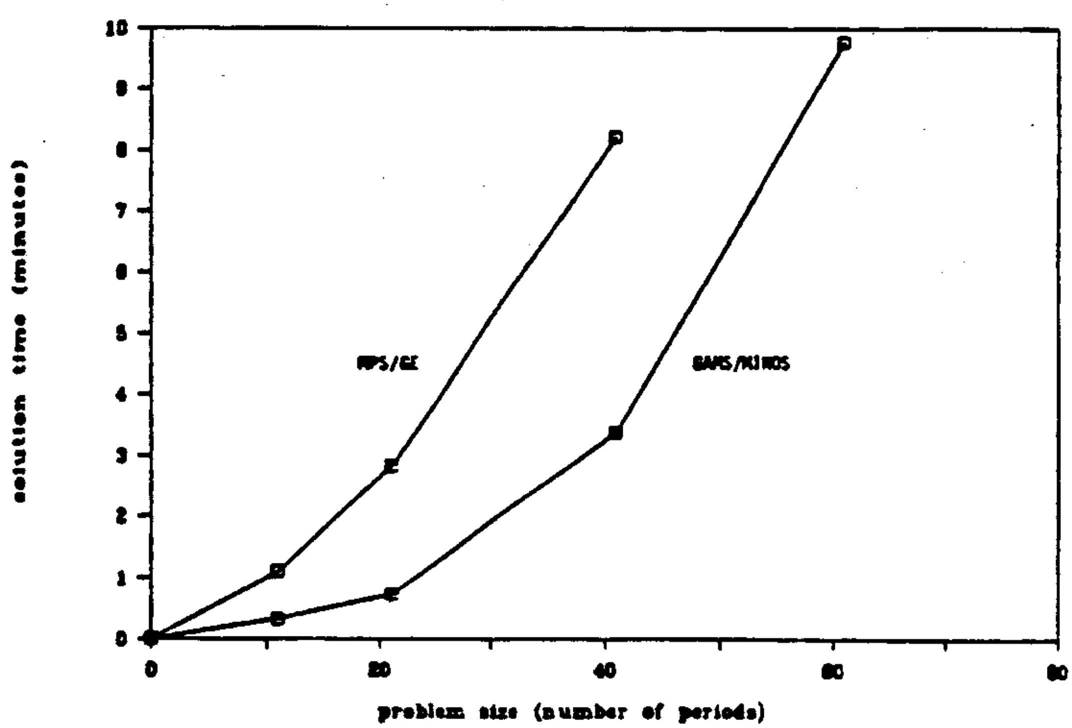
K_1 fixed (initial condition);

$I_T = \theta K_T$ (terminal condition);

and $I_t \leq B_t$.

In AGE models with a single consumer, demand functions are integrable and equilibrium conditions are equivalent to first order optimality conditions for a particular nonlinear optimization problem, namely that which maximizes the sum of producer and consumer surplus. (See Samuelson [1947]). It is for precisely this reason that Ramsey is a interesting test problem. It permits an efficiency comparison of SLCP with an optimization code. Figure 3 illustrates these results, comparing MPS/GE with GAMS/MINOS[8].

Figure 3: Ramsey Model Solution Times



It can be seen that the GAMS/MINOS times are faster

---

[8] For a general description of GAMS, see Meeraus [1983]. Manne [1986] describes this particular model. The runs in Figure 3 were conducted on an 8 MHz IBM/AT microcomputer.

than can be achieved with the MPS/GE code on this problem. Just the same, the comparison is encouraging: the optimization method works with a system containing half as many rows and roughly one-third as many coefficients as does SLCP. It should also be noted that the simple upper bounds on investment activities are handled implicitly in MINOS but are specified explicitly in SLCP. Implicit upper bounding could be implemented in Lemke's algorithm, and it would improve the SLCP run times for this and related models.[9]

Cameroon

This is a static open economy model which has been formulated as a square system of equations by Dahl and Devarajan [1987]. They solve the model with GAMS/MINOS, using the nonlinear optimizer simply to obtain a feasible solution. The solution time using MINOS is roughly five times longer than SLCP. The lesson provided by this example is that not every model which can be represented in optimization format should be solved as such. In particular, the projected Lagrangian algorithm implemented in MINOS 5.0 performs well when nonlinearities appear only in the objective function and the constraints are linear or nearly-linear functions. (See Murtaugh and Saunders [1982].)

LTM

This model is described in Manne and Rutherford [1984] and Rutherford [1986]. Two data sets are considered here. The three

---

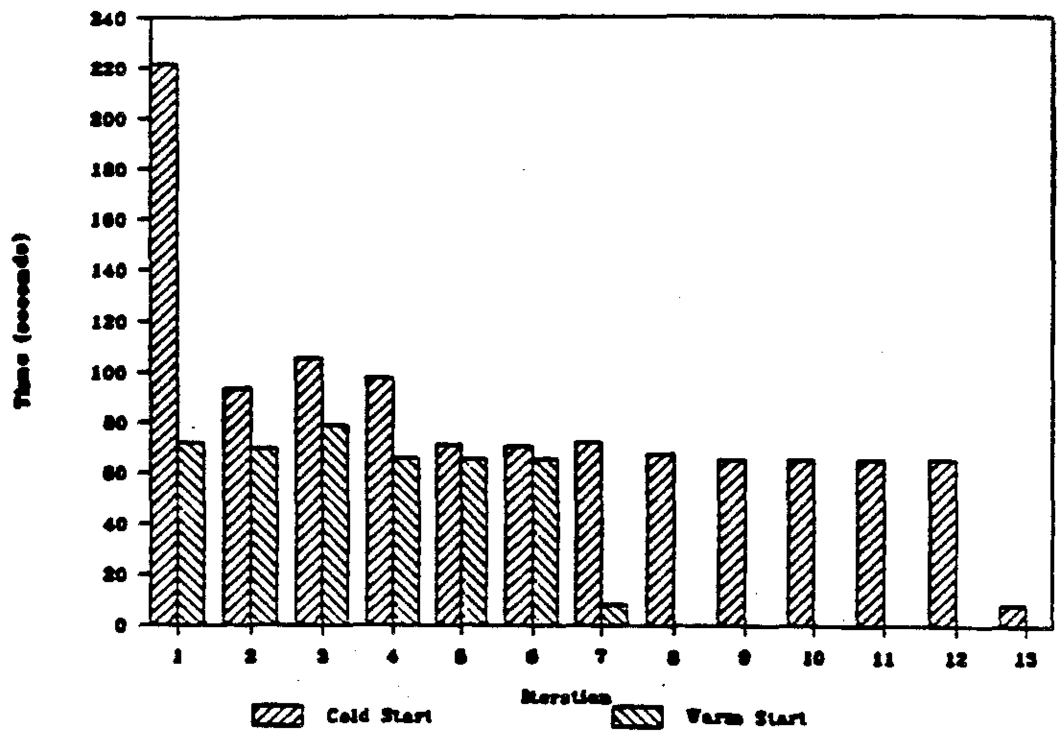
9 Bounds could be incorporated into the min-ratio test in a manner directly analogous to the treatment of bounds in the simplex method. On theoretical implications, see Kaneko [1978].

region model, although smaller in terms of variables, has more nonzeros and is considerably more difficult to solve. This is because it includes bilinear balance of payment constraints on capital flows of one region and these constraints are particularly dense.

"Cold-start" runs for both models are initiated from prices and activity levels from a steady-state equilibrium. In the base year, capital stocks are poorly adjusted, so this is not a particularly good starting point. The "warm-start" runs are initiated from the solutions of related cases. In the four region model, the initial deviations are 31.00 and 0.41 in the cold and warm starts, respectively.

The high cost of starting from scratch is motivation to save solutions for initiating related cases. Moving further from the solution increases both the number and cost of iterations. Figure 4 compares the cost per iteration for warm- and cold-starts, using the four region test case. In the cold start case, the initial basis is singular, so the first LCP is initiated from the origin. As a result, this LCP takes far more time than later subproblems. From iteration 5 onwards, the algorithm takes few if any Lemke pivots, and from iteration 8 it is equivalent to a Newton process on a system of equations.

Figure 4: Elapsed Time by Iteration



5. CONCLUSION

This paper has described an implementation of Mathiesen's SLCP algorithm for solving applied general equilibrium models. We have explored issues which do not arise in a theoretical analysis but which are unavoidable and important in designing reliable and efficient software. The paper reviewed system performance with test problems and empirical models. We substantiated Mathiesen's finding that SLCP is considerably faster than fixed-point and other methods for solving general equilibrium models. We find that when demand functions are integrable, optimization methods can be faster, but this is not necessarily true when there are highly nonlinear constraints.

In the course of the paper, several directions for further development and research work have been identified. These include

(1) using inexact subproblem solutions to avoid costly computation of exact LCP solutions when far from an equilibrium;

(2) implementing (implicit) simple bounds;

(3) introducing a scaling routine to improve robustness and overcome numerical problems;

(4) finding a merit norm for which a decrease in each iteration is assured; and

(5) determining ex-ante rules for the assignment of numeraire index.

Items (1), (2) and (3) are ideas which can be implemented and tested empirically, wheras (4) and (5) require new results in convergence theory.

In light of the efficiency of the SLCP algorithm applied to test problems and empirical models, it seems worthwhile to explore means of improving robustness and accessibility of the method. For the second of these objectives, extending GAMS to accomodate nonlinear complementarity problems would be a logical approach.

References

Ahn, B. and W. Hogan (1982), "On convergence of the PIES algorithm for computing equilibria," *Operations Research* 30 2:281-300.

Ballard, C., D. Fullerton, J.B. Shoven and J. Whalley (1985), *A General Equilibrium Model for Tax Policy Evaluation*, National Bureau of Economic Research Monograph, The University of Chicago Press.

Broadie, M. N. (1983a), "OCTASOLV user's guide," Systems Optimization Laboratory Technical Resport No. 83-9, Department of Operations Research, Stanford University.

Broadie, M. N. (1983b), "Subdivisions and antiprisms for PL homotopy algorithms," Ph.D. thesis, Department of Operations Research, Stanford University.

Cottle, R.W. and G.B. Dantzig (1968), "Complementary pivot theory of mathematical programming," *Linear Algebra and its Applications*, 1:103-125.

Dahl, H. and S. Devarajan, (1986) "Implementing a computable general equilibrium model on GAMS: The Cameroon Model", World Bank Working Paper (draft).

Drud, A., D. Kendrick and A. Meeraus (1986), "Hercules: A system for development of multisectoral economywide models," mimeo, World Bank Development Research Department.

Eaves, B.C. (1978), "A locally quadratically convergent algorithm for computing stationary points," Technical Report SOL 78-13, Department of Operations Research, Stanford University.

Gill, P., W. Murray, M. Saunders and M. Wright (1986), "Maintaining LU factors of a general sparse matrix," Technical Report SOL 86-8, Department of Operations Research, Stanford University.

Haaland, J., V. Norman, T. Rutherford and T. Wergeland (1986), "VEMOD: A Ricardo-Heckscher-Ohlin-Jones Model of World Trade", Scandanavian Journal of Economics (forthcoming).

Hogan, W. W. (1975), "Energy policy models for project independence," *Computation and Operations Research*, 2:251-271.

Josephy, N. (1979), "Newton's method for generalized equations," Technical summary report #1965, University of Wisconsin Mathematics Research Center.

Kaneko, I. (1978), "A linear complementarity problem with an n by

2n "P"-matrix," Mathematical Programming Study 7: 120-141.

Kimbell, L. and G. Harrison, (1986) "On the solution of general equilibrium models," *Economic Modeling*: 197-212.

Lemke, L. (1965), "Bimatrix equilibrium points and mathematical programming," *Management Science*, 11:681-689.

MacKinnon, J. (1976), "A technique for the solution of spatial equilibrium models," *Journal of Regional Science*, (16) 3:293-307.

Manne, A. and T. Rutherford, "LTM: A Long-Term Model of Mexico's Growth and Balance-of-Payment Constraints," Department of Operations Research, Stanford University.

Mathiesen, L. (1987), "An algorithm based on a sequence of linear complementarity problems applied to a Walrasian equilibrium model: an example," *Mathematical Programming* 37:1-18.

Mathiesen, L. (1985a), "Computational experience in solving equilibrium models by a sequence of linear complementarity problems," *Operations Research*, (33) 6:1225-1250.

Mathiesen, L. (1985b), "Computation of economic equilibria by a sequence of linear complementarity problems," *Mathematical Programming Study*, 23, North-Holland.

Mathiesen, L., and T. Rutherford (1983), "Testing the robustness of an iterative LCP algorithm for solving Walrasian equilibrium models," Discussion paper 0883, Norwegian School of Economics and Business Administration, Bergen, Norway.

Meeraus, A. (1983), "An algebraic approach to modeling," *Journal of Economic Dynamics and Control*, Volume 5, No. 1, February.

Merrill, O. H. (1972), "Applications and extensions of an algorithm that computes fixed points to certain upper semi-continuous point-to-set mappings, Ph.D. dissertation, Department of Industrial Engineering, University of Michigan.

Murtaugh, B. and M. Saunders (1984), "A projected lagrangian algorithm and its implementation for sparse nonlinear constraints", *Mathematical Programming Study* 16: 84-117.

Pang, J. (1986), "Inexact Newton methods for the nonlinear complementarity problem," *Mathematical Programming* 36:54-71.

Pang, J. and D. Chan (1982), "Iterative methods for variational and complementarity problems," *Mathematical Programming*, 24 : 284-313.

Pearson, K. R. (1986), "Automating the computation of solutions of large economic models," IMPACT Research Centre, Preliminary Working Paper No. IP-27, Melbourne, Australia; March.

Preckel, P. (1983), "Intertemporal Equilibrium Models: Developments and Results," Ph.D. dissertation, Department of Operations Research, Stanford University.

Robinson, S. M. (1975), "A quadratically-convergent algorithm for general nonlinear programming problems," *Mathematical Programming Study*, 3, North-Holland.

Rowse, J. (1982), "On the Solution of Spatial Equilibrium Models," *Decision Science* 13:619-637.

Rutherford, T. (1987), "A modeling system for applied general equilibrium analysis," Cowles Foundation Discussion Paper, Economics Department, Yale University.

Rutherford, T. (1986), "Applied General Equilibrium Modeling", Ph.D. dissertation, Department of Operations Research, Stanford University.

Rutherford, T. (1982), "Computing general equilibria in a complementarity format. NORGE: a general equilibrium model of the Norwegian economy," Engineer's Degree Thesis, Department of Operations Research, Stanford University.

Samuelson, P. (1947), *Foundations of Economic Analysis*, Harvard University Press.

Scarf, H. (1960), "Some examples of the global instability of the competitive equilibria," *International Economic Review* 1:157-171.

Scarf, H. with collaboration of T. Hansen, (1973) *The Computation of Economic Equilibria*, Yale University Press.

Shoven, J. and J. Whalley (1972), "A general equilibrium calculation of the effects of differential taxation of income from capital in the U.S.," *Journal of Public Economics* 1:281-322.

Shoven, J. and J. Whalley (1973), "General equilibrium with taxes: a computational procedure and existence proof", *Review of Economic Studies* 40: 475-490.

Stone, J. (1983), "Sequential optimization and complementarity techniques for computing economic equilibria," *Mathematical Programming Study* 23: 173-191.

Todd, M.J. (1980), "PLAGO: a FORTRAN implementation of a piecewise-linear homotopy algorithm for solving systems of nonlinear equations," Technical Report No. 454, School of

Operations Research and Industrial Engineering, Cornell University, Ithaca, New York (revised 1983, 1985).

Tomlin, J. (1976), "Programmer's guide to LCPL," Technical Report SOL 76-25, Department of Operations Research, Stanford University.

van der Laan, G. and A. J. Talman, (1979), "A restart algorithm for computing fixed points without an extra dimension," *Mathematical Programming Study*, 17, North-Holland.