...RSITY
...SITEIT
BRABANT
POSTBOX 90153
5000 LE TILBURG
THE NETHERLANDS

[redacted]

DEPARTMENT OF ECONOMICS

RESEARCH MEMORANDUM

AN SLSPP-ALGORITHM TO COMPUTE AN EQUILIBRIUM IN AN ECONOMY WITH LINEAR PRODUCTION TECHNOLOGIES

Hans Kremers and Dolf Talman

FEW 496

An *SLSPP*-algorithm to compute an equilibrium in an economy with linear production technologies

Hans KREMERS and Dolf TALMAN

Department of Econometrics
Tilburg University
P.O. Box 90153
5000 LE Tilburg
The Netherlands

June 12, 1991

Abstract

The problem of computing an equilibrium in an economy with linear production technologies is to find prices and activity levels such that no activity makes positive profits and the market for each commodity is cleared. If one normalizes prices on the unit simplex then this problem is equivalent to the so-called stationary point problem of the excess demand function in this economy on a particular subset of the unit-simplex. This problem can be solved by approximating this stationary point problem by a sequence of linear stationary point problems (*SLSPP*). In this way one obtains an iterative algorithm. In each iteration of this *SLSPP* the linear stationary point problem is solved by an algorithm based on ideas of Kamiya and Talman.

1 Introduction

This paper considers the computation of an equilibrium in an economy with linear production technologies. In Mathiesen [6] it is shown that this equilibrium problem is equivalent to a nonlinear complementarity problem (*NLCP*). Mathiesen approximates this *NLCP* by a sequence of linear complementarity problems (*SLCP*). Each linear complementarity problem (*LCP*) is a linearization of the *NLCP* in the solution either obtained from the previous *LCP* in the sequence or, if no previous *LCP* exists, in an arbitrarily chosen point. Each *LCP* in the sequence is then solved by the Lemke-Howson algorithm, see [5]. In case the Lemke-Howson algorithm solves each *LCP* in the sequence a sequence of approximating solutions is generated, possibly converging to a solution to the *NLCP*.

As shown in [7] the Lemke-Howson algorithm may also diverge in an *LCP* obtained in the sequence thereby failing to give an approximating solution to start up the next iterate of the *SLCP*. In order to get rid of this possible divergence of the Lemke-Howson algorithm in each iterate as well as to improve the possibilities for a better understanding of the theoretical properties of an *SLCP*-algorithm an alternative to Mathiesen [6] was introduced in Eaves [2]. Eaves shows that the equilibrium problem in an economy with linear production technologies is equivalent to a stationary point problem (*SPP*). Similar to Mathiesen [6] the *SPP* is approximated by a sequence of linear complementarity problems and each *LCP* in the sequence is again solved by the Lemke-Howson algorithm [5]. Furthermore Eaves [2] shows that the Lemke-Howson algorithm finds a solution in each *LCP* obtained in the sequence of his *SLCP*. Hence the *SLCP*-algorithm in Eaves [1] always generates a sequence of approximating solutions, possibly converging to a solution to the *SPP*.

In this paper we introduce a sequence of linear stationary point problems (*SLSPP*) to solve the *SPP*. In each iterate of the sequence a linear stationary point problem has to be solved. This problem is of a much smaller dimension than the *LCP*s obtained in the *SLCP* of Eaves. The linear stationary point problem obtained in each iterate of the sequence is solved by an algorithm which is based on an algorithm introduced by Kamiya and Talman [3] to compute a solution to linear stationary point problems on a polytope. Within a finite number of steps every *LSPP* in the sequence will be solved. Moreover, to initiate the algorithm at every iterate no perturbation of the starting point is needed as in [2].

In Section 4 of this paper we describe the steps of the *SLSPP*-algorithm while in Section 3 we prove that the equilibrium problem in an economy with linear production technologies is equivalent to a stationary point problem on a polytope. Section 5 discusses convergency issues concerning our algorithm and shows that our method always succeeds to compute a solution to the linear stationary point problem obtained in each iteration of the sequence. In Section 6 we compare our *SLSPP*-algorithm with the *SLCP*-methods of Mathiesen [6] and Eaves [2] but first we describe an economy with linear production technologies.

2 The economy with linear production technologies

Consider an economy with a finite number of consumers, commodities, and production activities. Each consumer in the economy is assumed to have an initial endowment in each of the, say $n+1$, commodities. Let ω denote the $(n+1)$-vector of total initial endowments of each commodity in the economy where ω_j is the endowment in commodity j, $j \in \{1, \ldots, n+1\}$. Given a price vector $p \in \Re^{n+1}_{+} \setminus \{0\}$ with p_j denoting the price of commodity j, let $d(p)$ denote the total demand of the consumers where $d_j(p)$ is the demand for commodity $j \in \{1, \ldots, n+1\}$.

Assumption 2.1 *The demand function d has the following properties.*

i) d is continuous in $p > 0$.

ii) d is homogeneous of degree zero.

iii) $p^\top d(p) = p^\top \omega$ *for every* $p > 0$.

At price vector $p > 0$ market excess demand is denoted by $z(p)$, i.e. $z_j(p)$ equals $d_j(p) - \omega_j$. The properties of the market excess demand function follow immediately from Assumption 2.1.

Property 2.1 *The market excess demand function* z *has the following properties.*

i) z *is continuous in* $p > 0$.

ii) z *is homogeneous of degree zero.*

iii) (Walras' law) $p^\top z(p) = 0$ *for every* $p > 0$.

Commodities in the economy can be produced by a finite number of, say l, activities. Activity j, $j \in \{1, \ldots, l\}$, is represented by a vector $a^j \in \Re^{n+1}$. The column vector a^j has components a^j_s such that $a^j_s \geq 0$ implies that activity j has an output of a^j_s units of commodity s while $a^j_s \leq 0$ implies that activity j uses $-a^j_s$ units of commodity s as an input when the activity level equals one. The level of activity j is denoted by the nonnegative number y_j. If one puts all activity levels y_j into an $(n+1)$-vector y and all columns a^j into an $(n+1) * l$-matrix A then the vector Ay denotes aggregate net input-output for all commodities in the economy given the activity level vector $y \in \Re^l_+$. So, given $y \geq 0$, $(Ay)_s > 0$ implies that commodity s serves as a net output to the economy as a whole and $(Ay)_s < 0$ implies that commodity s serves as a net input to the economy as a whole. With respect to the activity matrix A we assume that in case of production there exists at least one commodity serving as an input to the economy. Formally, we have the following assumption.

Assumption 2.2 *If* $Ay \geq 0$ *and* $y \geq 0$ *then* $y = 0$.

We assume that the producers running the activities in the economy choose the activity levels in such a way that the profits made by each activity are maximized. If the profit $p^\top a^j$ for some activity j is negative then the producer running activity

j sets the activity level y_j equal to zero. Raising y_j from zero would bring losses to the producer. If $p^\top a^j > 0$ then y_j is raised towards infinity by the producer running activity j while if $p^\top a^j = 0$ then any activity level y_j can be chosen such that $y_j \geq 0$.

The economy described above is said to be in equilibrium if the prices of the commodities in the economy and the activity levels are such that for every commodity demand is met by the initial endowment of the economy in this commodity and total net input-output of the activities. This condition implies that in equilibrium no activity makes positive profit. Therefore the following definition constitutes an equilibrium in an economy with linear production technologies.

Definition 2.1 *An equilibrium in an economy with linear production technologies is a price vector $p^* > 0$ and a vector with activity levels $y^* \geq 0$ such that*

i) $z(p^*) - Ay^* \leq 0$ *: market-clearance.*

ii) $p^{*\top} A \leq 0$ *: no profits.*

The properties of the market excess demand function together with the equilibrium conditions on p and y imply the following relations between the equilibrium prices and activity levels in the economy.

Property 2.2 *If p^* is an equilibrium price vector and y^* a vector of equilibrium activity levels in an economy with linear production technologies then*

$$p^{*\top} Ay^* = 0.$$

Proof:

From the market-clearance condition in Definition 2.1 it follows that $p^{*\top}(z(p^*) - Ay^*) = p^{*\top} z(p^*) - p^{*\top} Ay^* = -p^{*\top} Ay^* \leq 0$, applying Walras' law. Furthermore, multiplying $p^{*\top} A \leq 0$ by $y^* \geq 0$ gives $p^{*\top} Ay^* \leq 0$. Together with $-p^{*\top} Ay^* \leq 0$ this implies that $p^{*\top} Ay^* = 0$. □

Property 2.2 says that in equilibrium an activity showing a deficit $(p^{*\top}a^j < 0)$ is not producing $(y_j^* = 0)$ while an activity in operation $(y_j^* > 0)$ runs at balance $(p^{*\top}a^j = 0)$. This also follows from the presumed profit maximizing behaviour of the producers running the activities.

Property 2.3 *If p^* is an equilibrium price vector and y^* a vector of equilibrium activity levels in an economy with linear production technologies then*

$$p^{*\top}(z(p^*) - Ay^*) = 0.$$

Proof:
$p^{*\top}(z(p^*) - Ay^*) = p^{*\top}z(p^*) - p^{*\top}Ay^* = 0$ by applying Walras' law and Property 2.2.

□

Property 2.3 says that a commodity in excess supply $(z_i(p^*) - (Ay^*)_i < 0)$ has a price equal to zero $(p_i^* = 0)$ while a commodity with positive price $(p_i^* > 0)$ implies market clearance $(z_i(p^*) - (Ay^*)_i = 0)$.

3 The sequence of linear stationary point problems

An equilibrium in an economy with linear production technologies as defined in Definition 2.1 cannot be computed directly. First we have to take notice of the fact that because of homogeneity of degree zero of the excess demand function in the prices (Property 2.2ii) the existence of an equilibrium price vector p^* implies the existence of a ray of equilibrium price vectors $\{\lambda p^* \mid \lambda > 0\}$. To overcome this problem we normalize the prices on the unit-simplex S^n defined as $S^n = \{p \in \Re_+^{n+1} \mid \sum_{i=1}^{n+1} p_i = 1\}$.

Secondly, to compute an equilibrium as defined in Definition 2.1 we only should take into account prices for which $p^\top A \leq 0$. The prices $p \in S^n$ such that $p^\top A \leq 0$ have the shape of a polytope. Let S_A^n be this subset of S^n, i.e. $S_A^n = \{p \in \Re^{n+1} \mid$

$p^\top A \leq 0,\ p^\top e = 1,\ p \geq 0\}$ where e denotes the $(n+1)$-vector with all components equal to one. Hence to compute an equilibrium in an economy with linear production technologies is equivalent to finding a price vector $p^* \in S^n_A$ and a vector of activity levels $y^* \geq 0$ such that $z(p^*) - Ay^* \leq 0$. This problem is equivalent to finding a stationary point of the function f on the set S^n_A.

Definition 3.1 *Let C be a nonempty subset in $\Re^k$ and let $f : C \to \Re^k$ be a function. A point $\overline{x}$ in C is a* stationary point *of f on C if $\overline{x}^\top f(\overline{x}) \geq x^\top f(\overline{x})$ for all x in C. The stationary point problem (SPP) of f on C is to find a stationary point of f on C.*

We now prove that the problem to compute an equilibrium in an economy with linear production activities is equivalent to the *SPP* of the excess demand function z on S^n_A.

Theorem 3.1 *A price vector $p^* \in S^n_A$ is an equilibrium price vector in an economy with linear production technologies if and only if p^* is a stationary point of the excess demand function z on S^n_A.*

Proof:

Let $p^* \in S^n_A$ and $y^* \geq 0$ be an equilibrium in an economy with linear production technologies. Then

$$p^\top z(p^*) \leq p^\top z(p^*) - p^\top Ay^* = p^\top(z(p^*) - Ay^*) \leq 0 = p^{*\top} z(p^*)$$

for all $p \in S^n_A$. Therefore $p^* \in S^n_A$ is a stationary point of z on S^n_A.

Let $p^* \in S^n_A$ be a stationary point of z on S^n_A. Then for all $p \in S^n_A$ it holds that $p^\top z(p^*) \leq p^{*\top} z(p^*) = 0$. This is equivalent to p^* solving the optimization problem given by

$$\max_p \ p^\top z(p^*)$$

$$\text{s.t.} \quad p^\top A \leq 0 \tag{3.1}$$
$$e^\top p = 1$$
$$p \geq 0.$$

Take the vector y as the vector with the dual variables to the constraints in $p^\top A \leq 0$, β as the dual variable to the constraint $e^\top p = 1$, and $\mu \geq 0$ as the vector with dual variables to the constraints in $p \geq 0$. Then p^* being a solution to the optimization problem above implies that p^* fulfils the following first-order conditions:

$$\begin{aligned} z(p^*) &= Ay + \beta e - \mu \\ A^\top p^* &\leq 0 \\ e^\top p^* &= 1 \\ \mu^\top p^* &= 0, \ p^{*\top} Ay = 0 \\ \mu &\geq 0, \ y \geq 0, \ \beta \in \Re. \end{aligned} \tag{3.2}$$

Premultiplying $z(p^*) = Ay + \beta e - \mu$ by p^* results in $\beta = 0$. Thus, with $\mu \geq 0$ it follows that $z(p^*) - Ay = -\mu \leq 0$. Therefore p^* is an equilibrium price vector in an economy with linear production technologies. □

Theorem 3.1 implies that to compute an equilibrium in an economy with linear production technologies we have to compute a stationary point of z on S^n_A. In order to solve this *SPP* of z on S^n_A we propose to approximate this nonlinear problem by a sequence of linear stationary point problems. This *SLSPP* consists of a sequence of iterates where in each iterate the nonlinear *SPP* is linearized in a price vector either obtained as a solution to the previous iterate in the sequence or, if no previous iterate exists, chosen arbitrarily from S^n_A.[1]

Consider iteration k of the *SLSPP*. In this iteration the algorithm linearizes the *SPP* by replacing the excess demand function z with its first-order Taylor expansion

[1] $S^n_A \neq \emptyset$ as follows from Assumption 2.2 and Minkowsky's Separating Hyperplane Theorem: $Ay \geq 0$, $y \geq 0$ does not have a solution $y > 0$ implies that there exists a vector $p > 0$ such that $p^\top A \leq 0$.

in the price vector obtained as a solution to the previous iterate in case $k \geq 1$ and chosen arbitrarily from S_A^n in case $k = 0$. Denote this price vector by p^k. We allow p^k to lie on the boundary of S_A^n and we assume that the excess demand function z is differentiable in each point $p \in S^n$. Then the first order Taylor expansion of the excess demand function z in p^k, denoted z^k, exists and is equal to

$$z^k(p) = z(p^k) + \nabla z(p^k)p \tag{3.3}$$

for all $p \in S_A^n$ where $\nabla z(p^k)$ denotes the Jacobian matrix of first order derivatives of z to p in p^k. Notice that $\nabla z(p^k)p^k = 0$.

In iteration k the algorithm solves the linear stationary point problem of z^k on S_A^n, denoted by $LSPP_k$. A solution to $LSPP_k$ is not necessarily an equilibrium price vector in the original economy with linear production technologies.

If we are able to solve $LSPP_k$ for each $k = 0, 1, 2, \ldots$, then the *SLSPP*-algorithm generates a sequence of prices $\{p^k\}_{k=0}^{\infty}$ possibly converging to a solution to the *SPP* of z on S_A^n and therefore to an equilibrium price vector. In Section 4 we introduce an algorithm to solve $LSPP_k$ for all $k = 0, 1, 2, \ldots$.

4 The algorithm to solve $LSPP_k$

A face of S_A^n is determined by

$$\begin{aligned} \mathcal{F}(\mathcal{G}^1, \mathcal{G}^2) = \{p \in S_A^n \mid \ p^\top a^j &= 0 \text{ for all } j \in \mathcal{G}^1 \text{ and} \\ p_j &= 0 \text{ for all } j \in \mathcal{G}^2\} \end{aligned}$$

for certain sets $\mathcal{G}^1 \subseteq \{1, \ldots, l\}$ and $\mathcal{G}^2 \subset \{1, \ldots, n+1\}$. We assume without loss of generality that S_A^n is simple, i.e. if $\mathcal{F}(\mathcal{G}^1, \mathcal{G}^2) \neq \emptyset$ then its dimension is equal to $n - \mid \mathcal{G}^1 \mid - \mid \mathcal{G}^2 \mid$. The normal cone of $\mathcal{F}(\mathcal{G}^1, \mathcal{G}^2)$, denoted $\mathcal{N}(\mathcal{G}^1, \mathcal{G}^2)$, is then defined by

$$\begin{aligned} \mathcal{N}(\mathcal{G}^1, \mathcal{G}^2) = \{ & \textstyle\sum_{j \in \mathcal{G}^1} y_j a^j - \sum_{j \in \mathcal{G}^2} \mu_j e(j) + \beta e \mid \\ & y_j \geq 0 \ (j \in \mathcal{G}^1), \ \mu_j \geq 0 \ (j \in \mathcal{G}^2), \text{ and } \beta \in \Re\}. \end{aligned}$$

Theorem 4.1 provides for the relation between $\mathcal{F}(\mathcal{G}^1, \mathcal{G}^2)$ and $\mathcal{N}(\mathcal{G}^1, \mathcal{G}^2)$, and a stationary point $\overline{p}$ of z^k on S^n_A.

Theorem 4.1 *A point $\overline{p} \in S^n_A$ is a stationary point of z^k on S^n_A if and only if there exist sets $\mathcal{G}^1 \subseteq \{1, \ldots, l\}$ and $\mathcal{G}^2 \subset \{1, \ldots, n+1\}$ such that $\overline{p} \in \mathcal{F}(\mathcal{G}^1, \mathcal{G}^2)$ and $z^k(\overline{p}) \in \mathcal{N}(\mathcal{G}^1, \mathcal{G}^2)$.*

Proof:

Suppose $\overline{p} \in S^n_A$ is a stationary point of z^k on S^n_A. Then $\overline{p}$ solves the linear programming problem

$$\max_p \; p^\top z^k(\overline{p}) \text{ s.t. } A^\top p \leq 0, \; e^\top p = 1, \; p \geq 0.$$

The Duality Theorem of Linear Programming implies that this maximization problem is equivalent to solving the minimization problem

$$\min_\beta \; \beta$$

$$\text{s.t.} \quad z^k(\overline{p}) = Ay + \beta e - \mu$$
$$\mu \geq 0, \; y \geq 0, \; \beta \in \Re.$$

As $\overline{p}$ is a solution to the maximization problem there exists a solution $\overline{y}$, $\overline{\beta}$, and $\overline{\mu}$ to the maximization problem. Choose $\mathcal{G}^1$ to be the set $\{j \mid \overline{y}_j > 0\}$ and $\mathcal{G}^2$ to be the set $\{j \mid \overline{\mu}_j > 0\}$. Then the minimization problem implies that $z^k(\overline{p}) \in \mathcal{N}(\mathcal{G}^1, \mathcal{G}^2)$ and the complementary slackness condition in linear programming gives $\overline{p} \in \mathcal{F}(\mathcal{G}^1, \mathcal{G}^2)$.

□

The algorithm we propose to solve $LSPP_k$ in iteration k is a piecewise linear path following algorithm starting in $p^k \in S^n_A$ and ending up with a stationary point of z^k on S^n_A. Each point p on the path can be seen as a stationary point of z^k on $S^n_A(\lambda) := (1 - \lambda)\{p^k\} + \lambda S^n_A$ for some λ between zero and one. The parameter λ is a homotopy parameter running from zero to one. If $\lambda = 0$ then $S^n_A(0) = \{p^k\}$. So

p^k is a stationary point of z^k on $S^n_A(0)$ and an end point of the path followed by the algorithm. When, while following the path, λ becomes one in a point $\overline{p} \in S^n_A$ then $\overline{p}$ is a stationary point of z^k on $S^n_A = S^n_A(1)$ and $p^{k+1} = \overline{p}$ is taken as the starting point for the next iteration.

Let p be an arbitrary point in $S^n_A(\lambda)$ for some given λ between zero and one. Then, by definition of $S^n_A(\lambda)$, $p = (1-\lambda)p^k + \lambda q$ for some $q \in S^n_A$. If q is such that it solves $\max\{\hat{p}^\top z^k(p) \mid \hat{p} \in S^n_A\}$ then Lemma 4.2 shows that p is a stationary point of z^k on $S^n_A(\lambda)$.

Lemma 4.1 *If $p = (1-\lambda)p^k + \lambda q$ for some λ, $0 \leq \lambda \leq 1$, where $q = \arg\max\{\hat{p}^\top z^k(p) \mid \hat{p} \in S^n_A\}$ then p is a stationary point of z^k on $S^n_A(\lambda)$.*

Proof:
Let $p = (1-\lambda)p^k + \lambda q$ for some λ, $0 \leq \lambda \leq 1$, where $q = \arg\max\{\hat{p}^\top z^k(p) \mid \hat{p} \in S^n_A\}$. For every $\hat{p} \in S^n_A(\lambda)$ there exists a $\hat{q} \in S^n_A$ such that $\hat{p} = (1-\lambda)p^k + \lambda\hat{q}$ by definition of $S^n_A(\lambda)$. Consequently,

$$\begin{aligned}
\hat{p}^\top z^k(p) &= (1-\lambda)(p^k)^\top z^k(p) + \lambda\hat{q}^\top z^k(p) \\
&\leq (1-\lambda)(p^k)^\top z^k(p) + \lambda q^\top z^k(p) \\
&= ((1-\lambda)p^k + \lambda q)^\top z^k(p) \\
&= p^\top z^k(p).
\end{aligned}$$

Hence p is a stationary point of z^k on $S^n_A(\lambda)$. □

Choose $\mathcal{G}^1 = \{j \mid q^\top a^j = 0\}$ and $\mathcal{G}^2 = \{j \mid q_j = 0\}$. Then $q \in \mathcal{F}(\mathcal{G}^1, \mathcal{G}^2)$ and, by construction of q, $z^k(p) \in \mathcal{N}(\mathcal{G}^1, \mathcal{G}^2)$. Therefore the algorithm we propose follows a path of points in S^n_A such that for every point p there exist subsets $\mathcal{G}^1 \subseteq \{1, \ldots, l\}$ and $\mathcal{G}^2 \subset \{1, \ldots, n+1\}$ satisfying

1) $p = (1-\lambda)p^k + \lambda q$ for some λ, $0 \leq \lambda \leq 1$, and some $q \in \mathcal{F}(\mathcal{G}^1, \mathcal{G}^2)$

2) $z^k(p) \in \mathcal{N}(\mathcal{G}^1, \mathcal{G}^2)$.

Clearly p solves $LSPP_k$ when $\lambda = 1$ or $p^k \in \mathcal{F}(\mathcal{G}^1, \mathcal{G}^2)$, since in these cases also $p \in \mathcal{F}(\mathcal{G}^1, \mathcal{G}^2)$.

Suppose $p^k \notin \mathcal{F}(\mathcal{G}^1, \mathcal{G}^2)$ and $\dim \mathcal{F}(\mathcal{G}^1, \mathcal{G}^2) = m$. We can represent each point $q \in \mathcal{F}(\mathcal{G}^1, \mathcal{G}^2)$ as an affine combination of $m + 1$ affinely independent points in the affine hull of $\mathcal{F}(\mathcal{G}^1, \mathcal{G}^2)$, denoted $\text{aff}\mathcal{F}(\mathcal{G}^1, \mathcal{G}^2)$. Let $w^0, \ldots, w^m$ represent these affinely independent points in $\text{aff}\mathcal{F}(\mathcal{G}^1, \mathcal{G}^2)$. Then, for every $q \in \mathcal{F}(\mathcal{G}^1, \mathcal{G}^2)$, there exist numbers λ_j, $j = 0, 1, \ldots, m$, such that $q = \sum_{j=0}^m \lambda_j w^j$, $\lambda_j \in \Re$, and $\sum_{j=0}^m \lambda_j = 1$. From $\sum_{j=0}^m \lambda_j = 1$ it follows that $\lambda_0 = 1 - \sum_{j=1}^m \lambda_j$. Therefore we can represent each $q - w^0$, $q \in \mathcal{F}(\mathcal{G}^1, \mathcal{G}^2)$ as a linear combination of linearly independent vectors $w^j - w^0$, $j = 1, \ldots, m$, or $q = w^0 + \sum_{j=1}^m \lambda_j (w^j - w^0)$, $\lambda_j \in \Re$. Hence, the algorithm follows a path of points in S_A^n such that for every point p there exist subsets $\mathcal{G}^1 \subseteq \{1, \ldots, l\}$ and $\mathcal{G}^2 \subset \{1, \ldots, n+1\}$ satisfying

$$\begin{aligned} &1)\ p = (1 - \lambda)p^k + \lambda w^0 + \textstyle\sum_{j=1}^m \lambda_j (w^j - w^0), \text{ for some } \lambda \in [0, 1] \\ &\quad \text{and } \lambda_j \in \Re,\ j = 1, \ldots, m \end{aligned} \tag{4.1}$$

2) $z^k(p) \in \mathcal{N}(\mathcal{G}^1, \mathcal{G}^2)$.

All the points $p \in S_A^n$ satisfying (4.1) are stationary points of z^k on $S_A^n(\lambda)$. If one combines (4.1) together with $z^k(p) = z(p^k) + \nabla z(p^k) p$, then p is a stationary point of z^k on $S_A^n(\lambda)$ if and only if for some $\mathcal{G}^1 \subseteq \{1, \ldots, l\}$ and $\mathcal{G}^2 \subset \{1, \ldots, n+1\}$ the system of linear equations

$$\lambda \nabla z(p^k) w^0 + \sum_{j=1}^m \lambda_j \nabla z(p^k)(w^j - w^0) - \sum_{j \in \mathcal{G}^1} y_j a^j + \sum_{j \in \mathcal{G}^2} \mu_j e(j) - \beta e = -z(p^k) \tag{4.2}$$

has a solution $0 \le \lambda \le 1$, $\lambda_j \in \Re$, $y_j \ge 0$ $(j \in \mathcal{G}^1)$, $\mu_j \ge 0$ $(j \in \mathcal{G}^2)$, $\beta \in \Re$, such that $p = (1 - \lambda)p^k + \lambda q$ with $q^\top a^j \le 0$ for $j \notin \mathcal{G}^1$ and $q_j \ge 0$ for $j \notin \mathcal{G}^2$, where $q = w^0 + \sum_{j=1}^m \lambda_j \lambda^{-1}(w^j - w^0)$. This system contains $n+1$ equations with $n+2$ unknowns leaving us with one degree of freedom. Therefore, assuming nondegeneracy, system (4.2) represents a line segment of solutions to the *LSPP* of z^k on $S_A^n(\lambda)$.

As will be shown in Section 5 and assuming nondegeneracy the line segment of solutions obtained from (4.2) has two end points. This line segment will be followed by making a linear programming pivot step in (4.2) in one end point with one of the variables λ, λ_m, y_j for some $j \in \mathcal{G}^1$, or μ_j for some $j \in \mathcal{G}^2$. The other end point of this line segment is a point $\overline{p}$ in S^n_A where $\lambda = 1$, one of the restricted variables in (4.2) is equal to zero, or one of the constraints on q is binding.

Case 1: λ becomes equal to 1. Then $\overline{p} = w^0 + \sum_{j=1}^m \lambda_j(w^j - w^0) \in \mathcal{F}(\mathcal{G}^1, \mathcal{G}^2)$ while $z^k(\overline{p}) \in \mathcal{N}(\mathcal{G}^1, \mathcal{G}^2)$. Theorem 4.1 implies that $\overline{p}$ is a stationary point of z^k on S^n_A and $p^{k+1} = \overline{p}$ is taken as the starting point to the next iteration.

Case 2: y_t becomes zero for some $t \in \mathcal{G}^1$. Then, at $\overline{p}$, it holds that

$$z^k(\overline{p}) = \sum_{j \in \mathcal{G}^1 \setminus \{t\}} y_j a^j - \sum_{j \in \mathcal{G}^2} \mu_j e(j) + \beta e.$$

Hence $z^k(\overline{p}) \in \mathcal{N}(\mathcal{G}^1 \setminus \{t\}, \mathcal{G}^2)$. If $p^k \in \mathcal{F}(\mathcal{G}^1 \setminus \{t\}, \mathcal{G}^2)$, i.e. $(p^k)^\top a^j = 0$ for all $j \in \mathcal{G}^1 \setminus \{t\}$ and $p^k_j = 0$ for all $j \in \mathcal{G}^2$, then $\overline{p} \in \mathcal{F}(\mathcal{G}^1 \setminus \{t\}, \mathcal{G}^2)$ and Theorem 4.1 implies that $\overline{p}$ is a stationary point of z^k on S^n_A. Subsequently $p^{k+1} = \overline{p}$ is taken as the starting point to the next iteration. Otherwise, if $p^k \notin \mathcal{F}(\mathcal{G}^1 \setminus \{t\}, \mathcal{G}^2)$, then the algorithm maintains the validity of the conditions in (4.1) by generating prices $p = (1-\lambda)p^k + \lambda q$ such that $0 \le \lambda \le 1$, $z^k(p) \in \mathcal{N}(\mathcal{G}^1 \setminus \{t\}, \mathcal{G}^2)$, and the vector q an affine combination in $\mathcal{F}(\mathcal{G}^1 \setminus \{t\}, \mathcal{G}^2)$ of $m+2$ affinely independent points $\hat{w}^0, \ldots, \hat{w}^{m+1}$ in $\mathrm{aff}\mathcal{F}(\mathcal{G}^1 \setminus \{t\}, \mathcal{G}^2)$. Since the points $w^0, \ldots, w^m$ are already such that $\mathrm{aff}\mathcal{F}(\mathcal{G}^1, \mathcal{G}^2) = \mathrm{aff}\{w^0, \ldots, w^m\}$ one can take $\hat{w}^j = w^j$ for $j = 0, 1, \ldots, m$. Then $\hat{w}^{m+1}$ must be such that $\hat{w}^{m+1} \in \mathrm{aff}\mathcal{F}(\mathcal{G}^1 \setminus \{t\}, \mathcal{G}^2)$ and $\hat{w}^{m+1} \notin \mathrm{aff}\mathcal{F}(\mathcal{G}^1, \mathcal{G}^2)$. By definition this implies that $\hat{w}^{m+1}$ has to fulfil the conditions

$$\begin{aligned} (a^j)^\top \hat{w}^{m+1} &= 0 \text{ for all } j \in \mathcal{G}^1 \setminus \{t\} \\ (a^t)^\top \hat{w}^{m+1} &\neq 0 \\ e^\top \hat{w}^{m+1} &= 1 \\ \hat{w}^{m+1}_j &= 0 \text{ for all } j \in \mathcal{G}^2. \end{aligned}$$

For example we can take $(a^t)^\top \hat{w}^{m+1} = -1$.

Let C be the matrix containing the basic column vectors $-e$, $-a^j$ ($j \in \mathcal{G}^1$), and $e(j)$ ($j \in \mathcal{G}^2$) of system (4.2). Then the conditions on $\hat{w}^{m+1}$ reduce to

$$C^\top \hat{w}^{m+1} = \hat{e}(h_1) - \hat{e}(h_2),$$

where h_2 denotes the index of the column $-e$ in C, h_1 denotes the index of the column $-a^t$ in C, and $\hat{e}(j)$ is the j-th unit vector having the same length as the number of columns in the matrix C. Let D denote the matrix containing the remaining basic vectors of (4.2) as its columns. Then $\hat{w}^{m+1}$ can be determined from

$$\begin{pmatrix} C^\top \\ D^\top \end{pmatrix} \hat{w}^{m+1} = \begin{pmatrix} \hat{e}(h_1) - \hat{e}(h_2) \\ d \end{pmatrix} \tag{4.3}$$

for some arbitrarily chosen vector d of appropriate length. Without loss of generality we take $d = 0$. Then we take $\hat{w}^{m+1}$ equal to

$$\hat{w}^{m+1} = (B^\top)^{-1} e(h_1) - (B^\top)^{-1} e(h_2) \tag{4.4}$$

where $B = (C\ D)$, $e(h_1) = (\hat{e}(h_1)^\top,\ 0)^\top$ and $e(h_2) = (\hat{e}(h_2)^\top,\ 0)^\top$. Notice that the inverse of the matrix $B^\top$ is the transpose of the basis inverse obtained from the pivoting tableau corresponding to system (4.2). To obtain $\hat{w}^{m+1}$ we must therefore subtract the column of the transpose of the basis inverse corresponding to a^t from the column of this matrix corresponding to e. The algorithm proceeds by pivoting the column $\nabla z(p^k)(\hat{w}^{m+1} - \hat{w}^0)$ into the appropriate pivot system thereby raising the variable λ_{m+1} from zero.

Case 3: μ_t becomes zero for some $t \in \mathcal{G}^2$ in a point $\overline{p} \in S^n_A$. This case is similar to the previous one. In $\overline{p}$ it holds that

$$z^k(\overline{p}) = \sum_{j \in \mathcal{G}^1} y_j a^j - \sum_{j \in \mathcal{G}^2 \setminus \{t\}} \mu_j e(j) + \beta e.$$

Hence $z^k(\overline{p}) \in \mathcal{N}(\mathcal{G}^1, \mathcal{G}^2 \setminus \{t\})$. If $p^k \in \mathcal{F}(\mathcal{G}^1, \mathcal{G}^2 \setminus \{t\})$, i.e. $(p^k)^\top a^j = 0$ for all $j \in \mathcal{G}^1$ and $p^k_j = 0$ for all $j \in \mathcal{G}^2 \setminus \{t\}$, then $\overline{p} \in \mathcal{F}(\mathcal{G}^1, \mathcal{G}^2 \setminus \{t\})$ and Theorem 4.1 implies that $\overline{p}$ is a stationary point of z^k on S^n_A. Subsequently $p^{k+1} = \overline{p}$ is taken as the starting point to the next iteration. Otherwise, if $p^k \notin \mathcal{F}(\mathcal{G}^1, \mathcal{G}^2 \setminus \{t\})$, then the algorithm maintains the validity of the conditions in (4.1) by generating prices $p = (1-\lambda)p^k + \lambda q$ such that $0 \le \lambda \le 1$, $z^k(p) \in \mathcal{N}(\mathcal{G}^1, \mathcal{G}^2 \setminus \{t\})$, and the vector q is an affine combination in $\mathcal{F}(\mathcal{G}^1, \mathcal{G}^2 \setminus \{t\})$ of $m+2$ affinely independent given points $\hat{w}^0, \ldots, \hat{w}^{m+1}$ in aff$\mathcal{F}(\mathcal{G}^1, \mathcal{G}^2 \setminus \{t\})$. Similar to *Case 2* we take $\hat{w}^j = w^j$, $j = 0, 1, \ldots, m$, and $\hat{w}^{m+1} = (B^\top)^{-1}e(h_1) - (B^\top)^{-1}e(h_2)$ where $(B^\top)^{-1}$ and h_1 are defined as in *Case 2* while h_2 corresponds to the index of the column $e(t)$. The algorithm proceeds by pivoting the column $\nabla z(p^k)(\hat{w}^{m+1} - \hat{w}^0)$ into the appropriate pivot system thereby raising the variable λ_{m+1} from zero.

Case 4: For some $t \notin \mathcal{G}^1$ it holds that

$$\lambda(a^t)^\top w^0 = \sum_{j=1}^{m} \lambda_j (a^j)^\top (w^0 - w^j).$$

Then $(a^t)^\top q^1 = 0$ where $q^1 = w^0 + \sum_{j=1}^m (\lambda_j/\lambda)(w^j - w^0)$ is such that $\overline{p} = (1-\lambda)p^k + \lambda q^1$. Hence $q^1 \in \mathcal{F}(\mathcal{G}^1 \cup \{t\}, \mathcal{G}^2) \subset bd\mathcal{F}(\mathcal{G}^1, \mathcal{G}^2)$. The algorithm maintains the validity of (4.1) by generating prices $p = (1-\lambda)p^k + \lambda q$ such that $0 \leq \lambda \leq 1$, $q \in \mathcal{F}(\mathcal{G}^1 \cup \{t\}, \mathcal{G}^2)$, and $z^k(p) \in \mathcal{N}(\mathcal{G}^1 \cup \{t\}, \mathcal{G}^2)$. Since $\dim \mathcal{F}(\mathcal{G}^1 \cup \{t\}, \mathcal{G}^2) = m - 1$ we should determine m affinely independent points aff$\mathcal{F}(\mathcal{G}^1 \cup \{t\}, \mathcal{G}^2)$. These points, say $\hat{w}^0, \ldots, \hat{w}^{m-1}$, can be obtained by a parallel movement of the points $w^0, \ldots, w^m$ onto aff$\mathcal{F}(\mathcal{G}^1 \cup \{t\}, \mathcal{G}^2)$ and deleting one of them in such a way that the remaining m points are affinely independent as follows.

Let $r = (1-\lambda^0)p^k + \lambda^0 q^0$ be the previous end point obtained by the algorithm where $q^0 \in \mathcal{F}(\mathcal{G}^1, \mathcal{G}^2)$ is such that $q^0 = w^0 + \sum_{j=1}^m (\lambda_j^0/\lambda^0)(w^j - w^0)$ with $0 < \lambda^0 \leq 1$ and $\lambda_j^0 \in \Re$ obtained from the solution to (4.2) in r. The algorithm moves all points w^j, $j = 0, 1, \ldots, m$, parallel to $q^1 - q^0$ onto aff$\mathcal{F}(\mathcal{G}^1 \cup \{t\}, \mathcal{G}^2)$. This parallel movement of w^j, $j = 0, 1, \ldots, m$, results in $m+1$ points $w^j + \delta^j(q^1 - q^0) \in$ aff$\mathcal{F}(\mathcal{G}^1 \cup \{t\}, \mathcal{G}^2)$ where $\delta^j = (a^t)^\top w^j/(a^t)^\top q^0$, $j = 0, 1, \ldots, m$. The appendix to this paper shows that if we delete a point with index $g \in \{0, \ldots, m\}$ for which $\lambda_g \neq \lambda_g^0$ then the remaining points constitute m affinely independent points in aff$\mathcal{F}(\mathcal{G}^1 \cup \{t\}, \mathcal{G}^2)$

Take $\hat{w}^j = w^j + \delta^j(q^1 - q^0)$, $j = 0, 1, \ldots, g-1$, and $\hat{w}^{j-1} = w^j + \delta^j(q^1 - q^0)$, $j = g+1, \ldots, m$. Then, in the current basis inverse we replace $\nabla z(p^k)w^0$ by the column $\nabla z(p^k)\hat{w}^0$, $\nabla z(p^k)(w^j - w^0)$ by $\nabla z(p^k)(\hat{w}^j - \hat{w}^0)$, $j = 1, \ldots, g-1$, $\nabla z(p^k)(w^j - w^0)$

by $\nabla z(p^k)(\hat{w}^{j-1} - \hat{w}^0)$ for $j = g+1, \ldots, m$, and $\nabla z(p^k)(w^g - w^0)$ by the vector with which the pivoting step was made. Notice that if the pivoting step was made with the vector $\nabla z(p^k)(w^m - w^0)$ then $\hat{w}^0, \ldots, \hat{w}^{m-1}$ must be affinely independent since $\lambda_m > 0 = \lambda_m^0$. In that case we can take $g = m$ and $\nabla z(p^k)(w^m - w^0)$ does not enter the basis inverse. The algorithm proceeds by pivoting the column $-a^t$ into the new system thereby raising the variable y_t from zero .

Case 5: For some $t \notin \mathcal{G}^2$ it holds that

$$\lambda w_t^0 = \sum_{j=1}^{m} \lambda_j (w_t^0 - w_t^j).$$

This case is similar to the previous one. Then $q_t^1 = 0$ where $q^1 = w^0 + \sum_{j=1}^{m} (\lambda_j/\lambda)(w^j - w^0)$ is such that $\bar{p} = (1-\lambda)p^k + \lambda q^1$ and $q^1 \in \mathcal{F}(\mathcal{G}^1, \mathcal{G}^2 \cup \{t\}) \subset bd\mathcal{F}(\mathcal{G}^1, \mathcal{G}^2)$. The algorithm maintains the validity of (4.1) by generating prices $p = (1-\lambda)p^k + \lambda q$ such that $0 \leq \lambda \leq 1$, $q \in \mathcal{F}(\mathcal{G}^1, \mathcal{G}^2 \cup \{t\})$ and $z^k(p) \in \mathcal{N}(\mathcal{G}^1, \mathcal{G}^2 \cup \{t\})$. Similar to *Case 4* we determine m affinely independent points in aff$\mathcal{F}(\mathcal{G}^1, \mathcal{G}^2 \cup \{t\})$ from $w^j + \delta^j(q^1 - q^0)$ for $j = 0, 1, \ldots, m$ where $\delta^j = w_t^j / q_t^0$ and q^0 as defined in *Case 4*, and adapt the current basis inverse. The algorithm proceeds by pivoting the column $e(t)$ into the new pivot system thereby raising the variable μ_t from zero.

The cases 1 to 5 describe the performance of the algorithm at the end points of all possible line segments generated by the algorithm except at p^k where the algorithm is initiated. To show that p^k is an end point of a line segment generated by the algorithm we have to find sets $\mathcal{G}^1 \subseteq \{1, \ldots, l\}$ and $\mathcal{G}^2 \subset \{1, \ldots, n+1\}$ such that (4.1) is satisfied in p^k. This means that according to Lemma 4.2 this line segment contains points $p \in S_A^n$ such that $p = (1-\lambda)p^k + \lambda q$ for some λ, $0 \leq \lambda \leq 1$, and some $q \in \mathcal{F}(\mathcal{G}^1, \mathcal{G}^2)$ maximizing $p^\top z^k(p^k)$ subject to $p \in S_A^n$. Clearly q follows from solving either

the Primal	or	*the Dual*
$\max\ p^\top z(p^k)$		$\min\ \beta$
s.t. $A^\top p \leq 0$		s.t. $Ay - \mu + \beta e = z(p^k)$
$-p \leq 0$		$y \geq 0,\ \mu \geq 0.$
$e^\top p = 1$		

Let $\mathcal{G}_0^1 = \{j \mid y_j > 0\}$ and $\mathcal{G}_0^2 = \{j \mid \mu_j > 0\}$ after solving the Dual. Then assuming nondegeneracy the sets $\mathcal{G}_0^1$ and $\mathcal{G}_0^2$ define a face $\mathcal{F}(\mathcal{G}_0^1, \mathcal{G}_0^2)$, being a vertex of S_A^n, say w^0. In case $w^0 = p^k$ then p^k is an equilibrium price vector for the original economy. Otherwise p^k is the end point of a line segment of points $p \in S_A^n$ such that

1) $p = (1-\lambda)p^k + \lambda w^0$,

2) $z^k(p) \in \mathcal{N}(\mathcal{G}_0^1, \mathcal{G}_0^2)$.

Hence p^k fulfils the conditions in (4.1).

Notice that combining these conditions leads to the system

$$\lambda \nabla z(p^k) w^0 - \sum_{j \in \mathcal{G}_0^1} y_j a^j + \sum_{j \in \mathcal{G}_0^2} \mu_j e(j) - \beta e = -z(p^k) \tag{4.7}$$

having solutions $0 \leq \lambda \leq 1$, $y_j \geq 0$ $(j \in \mathcal{G}_0^1)$, $\mu_j \geq 0$ $(j \in \mathcal{G}_0^2)$, and $\beta \in \Re$. In $p = p^k$, system (4.7) has as solution $\lambda = 0$ and $y_j > 0$ $(j \in \mathcal{G}_0^1)$, $\mu_j > 0$ $(j \in \mathcal{G}_0^2)$, $\beta > 0$ obtained from solving the Dual in (4.5).

The line segment of solutions to (4.7) is followed from $p = p^k$ by making a linear programming pivot step with the column vector $\nabla z(p^k) w^0$ in (4.7) thereby raising λ from zero. While raising λ from zero the algorithm will encounter another end point of the line segment when either y_j becomes zero for some $j \in \mathcal{G}_0^1$, μ_j becomes zero for some $j \in \mathcal{G}_0^2$, or λ becomes one. The performance of the algorithm in such an end point is described in the cases 1 to 3 above.

5 Convergency issues

Starting in the point p^k obtained from the previous iteration of the *SLSPP* or, in case no previous iteration exists, chosen arbitrarily from S^n_A, the algorithm follows in iteration k a path of points either ending up with a solution to the $LSPP_k$ or possibly ending up in a secondary ray. In the previous section we described the performance of the algorithm in all possible end points of the line segments of the path followed by the algorithm. During the description we encountered cases in which the algorithm ends up with a solution to the $LSPP_k$. These cases are summarized in Lemma 5.1.

Lemma 5.1 *Let $\overline{p} = (1-\lambda)p^k + \lambda\overline{q}$ be an end point of a line segment of points generated by the algorithm with $\overline{q} \in \mathcal{F}(\mathcal{G}^1, \mathcal{G}^2)$ and $z^k(\overline{p}) \in \mathcal{N}(\mathcal{G}^1, \mathcal{G}^2)$ for some $\mathcal{G}^1 \subseteq \{1, \ldots, l\}$ and $\mathcal{G}^2 \subset \{1, \ldots, n+1\}$. Then $\overline{p}$ is a solution to the $LSPP_k$ if one of the following cases holds:*

i) $\lambda = 1$.

ii) $y_t = 0$ *for some* $t \in \mathcal{G}^1$, $(p^k)^\top a^j = 0$ *for all* $j \in \mathcal{G}^1 \setminus \{t\}$, *and* $p^k_j = 0$ *for* $j \in \mathcal{G}^2$.

iii) $\mu_t = 0$ *for some* $t \in \mathcal{G}^2$, $(p^k)^\top a^j = 0$ *for all* $j \in \mathcal{G}^1$, *and* $p^k_j = 0$ *for* $j \in \mathcal{G}^2 \setminus \{t\}$.

The algorithm does not find a solution to the $LSPP_k$ if it ends up in a so-called secondary ray in the pivot system (4.2). This means that the pivot variable can be raised towards infinity without violating any of the constraints on the variables in (4.2). Theorem 5.1 shows that the algorithm cannot end up in a secondary ray.

Theorem 5.1 *For all possible sets $\mathcal{G}^1 \subseteq \{1, \ldots, l\}$ and $\mathcal{G}^2 \subset \{1, \ldots, n+1\}$ and all $\hat{p} \in S^n_A$ the system of equations*

$$\lambda \nabla z(\hat{p}) w^0 + \sum_{j=1}^{m} \lambda_j \nabla z(\hat{p})(w^j - w^0) - \sum_{j \in \mathcal{G}^1} y_j a^j + \sum_{j \in \mathcal{G}^2} \mu_j e(j) - \beta e = -z(\hat{p}) \tag{5.1}$$

does not contain a ray of solutions satisfying $0 < \lambda \leq 1$, $\lambda_j \in \Re$, $y_j \geq 0$ $(j \in \mathcal{G}^1)$, $\mu_j \geq 0$ $(j \in \mathcal{G}^2)$, $\beta \in \Re$, such that $p = (1-\lambda)p^k + \lambda q$ where $q = w^0 + \sum_{i=1}^{m} \lambda_j \lambda^{-1}(w^j - w^0)$ such that $q^\top a^j \leq 0$ for $j \notin \mathcal{G}^1$ and $q_j \geq 0$ for $j \notin \mathcal{G}^2$.

Proof:

Suppose (5.1) contains a ray of admissable solutions $\lambda^0+\alpha\lambda^1$, $\lambda_j^0+\alpha\lambda_j^1$, $y_j^0+\alpha y_j^1$ ($j \in \mathcal{G}^1$), $\mu_j^0+\alpha\mu_j^1$ ($j \in \mathcal{G}^2$), $\beta^0+\alpha\beta^1$ for $\alpha \geq 0$. Then λ^0, λ_j^0, y_j^0 ($j \in \mathcal{G}^1$), μ_j^0 ($j \in \mathcal{G}^2$), β^0 is an admissable solution to (5.1) and λ^1, λ_j^1, y_j^1 ($j \in \mathcal{G}^1$), μ_j^1 ($j \in \mathcal{G}^2$), β^1 is a solution to the homogeneous system of equations

$$\lambda\nabla z(\hat{p})w^0 + \sum_{j=1}^{m}\lambda_j\nabla z(\hat{p})(w^j - w^0) - \sum_{j\in\mathcal{G}^1} y_j a^j + \sum_{j\in\mathcal{G}^2}\mu_j e(j) - \beta e = 0. \tag{5.2}$$

It is obvious that $\lambda^1 = 0$ as $\lambda^1 \neq 0$ would imply that there exists some $\alpha > 0$ such that $\lambda^0 + \alpha\lambda^1 > 1$. Suppose $\lambda_i^1 \neq 0$ for some $i \in \{1,\ldots,m\}$. For $\alpha > 0$, let $p(\alpha) = \hat{p} + \lambda^0(w^0 - \hat{p}) + \sum_{j=1}^{m}(\lambda_j^0 + \alpha\lambda_j^1)(w^j - w^0)$. Since $w^0,\ldots,w^m$ are affinely independent we must have that $\sum_{j=1}^{m}\lambda_j^1(w^j - w^0) \neq 0$. Hence for some $h \in \{1,\ldots,m\}$ we have that $p_h(\alpha) \neq p_h(0)$ when $\alpha > 0$. Since $p(\alpha)$ is linear in α this implies that for large enough α either $p_h(\alpha) < 0$ or $p_h(\alpha) > 1$. This contradicts the fact that $p(\alpha)$ lies in S_A^n for all $\alpha > 0$. Hence $\lambda_i^1 = 0$ for all $i \in \{1,\ldots,m\}$. Premultiplying (5.2) with $q^0 = w^0 + \sum_{j=1}^{m}\lambda_j^0(\lambda^0)^{-1}(w^j - w^0)$ such that $(q^0)^\top a^j \leq 0$ for $j \notin \mathcal{G}^1$ and $q_j^0 \geq 0$ for $j \notin \mathcal{G}^2$ gives $\beta^1 = 0$. Then it follows that

$$\sum_{j\in\mathcal{G}^2}\mu_j^1 e(j) = \sum_{j\in\mathcal{G}^1} y_j^1 a^j. \tag{5.3}$$

Notice that $y_j^1 \geq 0$ ($j \in \mathcal{G}^1$) as well as $\mu_j^1 \geq 0$ ($j \in \mathcal{G}^2$). This follows from $y_j^0+\alpha y_j^1$ ($j \in \mathcal{G}^1$) and $\mu_j^0 + \alpha\mu_j^1$ ($j \in \mathcal{G}^2$) being a solution to (5.1) for all $\alpha \geq 0$. Indeed, as $y_h^1 < 0$ for some $h \in \mathcal{G}^1$ or $\mu_j^1 < 0$ for some $j \in \mathcal{G}^2$ would imply the existence of an $\alpha > 0$ such that $y_h^0 + \alpha y_h^1 < 0$ or $\mu_j^0 + \alpha\mu_j^1 < 0$ thereby violating the constraints on y_h and μ_j in (5.1). Then $\sum_{j\in\mathcal{G}^2}\mu_j^1 e(j) \geq 0$ and with (5.3) it follows that $\sum_{j\in\mathcal{G}^1} y_j^1 a^j \geq 0$ and $y_j^1 \geq 0$ for all $j \in \mathcal{G}^1$. Hence Assumption 2.2 implies that $y_j^1 = 0$ ($j \in \mathcal{G}^1$). Therefore, with (5.3), it follows that $\mu_j^1 = 0$ ($j \in \mathcal{G}^2$).

But now we are left with the result that $\lambda^1 = 0$, $\lambda_j^1 = 0$, $y_j^1 = 0$ ($j \in \mathcal{G}^1$), $\mu_j^1 = 0$ ($j \in \mathcal{G}^2$), and $\beta^1 = 0$. Hence no secondary ray can occur which is in contradiction with the assumption underlying this proof. □

Theorem 5.1 guarantees that each line segment on the path contains exactly two end points. Assuming nondegeneracy at an end point of a line segment just one of the five cases described in Section 4 can occur. Therefore the starting point is the end point of a unique line segment whereas each other end point of a line segment is either an end point of a uniquely determined other line segment or a solution to $LSPP_k$.

Every line segment on the path in S^n_A is determined by the line segment of solutions to (4.2) for some unique $\mathcal{G}^1 \subseteq \{1, \ldots, l\}$ and $\mathcal{G}^2 \subset \{1, \ldots, n+1\}$. As $\mathcal{G}^1$ and $\mathcal{G}^2$ are both subsets of finite sets the total number of line segments is finite. Hence starting in $p = p^k$ the algorithm generates a sequence of different line segments. This sequence must be finite. Therefore the algorithm terminates within a finite number of steps with a solution to $LSPP_k$.

6 Conclusions

In this paper we have introduced an *SLSPP*-algorithm to compute an equilibrium in an economy with linear production technologies. An *SLSPP*-algorithm consists of iterations in which the stationary point problem determining an equilibrium in an economy with linear production technologies is linearized by taking the first-order Taylor expansion of the excess demand function z in the price vector either obtained from the previous iteration or, if no previous iteration exists, in an arbitrarily chosen price vector from S^n_A. In each iteration this results in a linear stationary point problem which we solve by the algorithm introduced in Section 3. In Theorem 5.1 we proved that this algorithm finds a solution for all possible starting points in S^n_A. Therefore the *SLSPP*-algorithm we introduced in this paper certainly generates a sequence of prices $\{p^k\}_{k=0}^{\infty}$, possibly converging towards an equilibrium in an economy with linear production technologies.

Contrary to the existence of convergence to an approximating price vector p^{k+1}

in each iteration k of the algorithm we are not able to show global convergence. In Mathiesen [6] some empirical results are given suggesting global convergence for his *SLCP*-algorithm. However, Mathiesen [6] also gave some examples, namely Scarf's unstable equilibria (see Scarf [7] for details), where his *SLCP*-algorithm failed to converge. Mathiesen's *SLCP*-algorithm does not even need to converge in each iteration thereby possibly failing to generate a sequence of prices $\{p^k\}_{k=0}^{\infty}$. Furthermore Mathiesen also has to choose among different *LCPs* to solve (dependent on the numeraire choice) in each iteration.

Our algorithm can also be seen as an improvement of the algorithm introduced in Eaves [2]. Like we do Eaves also normalizes prices on the unit-simplex in each iteration instead of taking some commodity as a numeraire. Then Eaves introduces an algorithm which generates a path in S_A^n by incorporating the "no-profit"-conditions in the pivot system. This results in a pivot system consisting of $n + l + 2$ equations in each iteration contrary to our pivot system which only consists of the "market-clearance"-conditions and hence results in a system of only $n+1$ equations. Therefore our algorithm processes the information much more efficiently than Eaves' algorithm. To give an idea of this difference in efficiency suppose one intends to compute an equilibrium in an economy consisting of 3 commodities and 1000 activities. Then our algorithm generates the path using a pivot system consisting of 3 equations. Eaves' *SLCP* however employs a pivot system consisting of 1004 equations in each iteration!

In iteration k the linearized excess demand function was denoted by z^k. Premultiplying z^k by p^{k+1} results in $(p^{k+1})^\top z^k(p^{k+1}) = \beta$. This β isn't necessarily equal to zero. In some way β says something about the inaccuracy of the obtained approximation in p^{k+1}. In Mathiesen [6] the inaccuracy in iteration k follows from $p_s^{k+1} z_s^k(p^{k+1}) - p_s^{k+1} z_s(p^{k+1})$ with commodity s being the numeraire commodity. In iteration k the market clearance condition with respect to the numeraire commodity s is not taken into account. Therefore $z_s^k(p^{k+1})$ can take any value at the end of iteration k, while β can take any value at the end of iteration k in our algorithm.

The possibility that the algorithm has to start at the beginning of some iteration on the boundary with the price of some commodities equal to zero may cause problems in Mathiesen's *SLCP*, Eaves' *SLCP* as well as our *SLSPP* because in general $z_j(p) \rightarrow \infty$ when p_j converges to zero. Mathiesen and Eaves suggested to perturbe p^k by taking the modified price vector $p^{k-1} + t(p^k - p^{k-1})$, $0 < t < 1$, instead of p^k as a starting point for iteration k. This perturbation of the starting point can cause the algorithm to zigzag along the boundary as some of Mathiesen's numerical examples showed.

As Eaves algorithm as well as our algorithm generate only prices in the subset S^n_A of S^n it may also occur that the starting point of the algorithm in some iteration lies in the boundary of S^n_A with only binding "no-profit"-conditions. This causes no problems for our algorithm. The performance remains exactly the same as described in Section 4. In Eaves' algorithm the starting point has again to be perturbed. Because this case may often occur when applying the algorithms this feature of our algorithm can be regarded as an improvement of Eaves' *SLCP*. In Mathiesen [6] the Lemke algorithm [4] was applied to solve the *LCP* in each of the iterations of the *SLCP*. The main drawback of applying Lemke's algorithm is that it lacks the possibility to start in an arbitrary starting point, causing a loss of information when proceeding from one iteration of the *SLCP* to the next one. Furthermore Mathiesen [6] had to restate the original equilibrium problem to a form suitable to apply the Lemke algorithm. This last feature as well as the fixed starting point imposed by applying the Lemke algorithm contributed to a large extent in the problems encountered when using the *SLCP* of Mathiesen.

In order to get rid of the problems encountered with Mathiesen [6] and Eaves [2] we showed that the original equilibrium problem in an economy with linear production technologies is equivalent to the stationary point problem of z on the subset S^n_A of S^n instead of rewriting this problem to a form fit for applying an already existing algorithm. This *SPP* was approximated by a sequence of linear stationary point problems. To solve this linear stationary point problem in each iteration of the sequence we introduced a new algorithm based on the ideas of Kamiya and Talman [3]. They improved already existing algorithms to solve a stationary point problem on a polytope like Yamamoto [9]. Applying this algorithm to our problem would imply that in cases 2 and 3 of our *SLSPP*-algorithm described in Section 4 we would be obliged to calculate a sequence of vertices of S^n_A. This means solving a sequence of linear programming problems similar to the case when we had to show that the starting point was an end point of the path followed by the algorithm. The fact that Kamiya and Talman [3] presented an alternative avoiding the calculation of vertices of S^n_A but using affinely independent points instead made application of their ideas to our equilibrium problem very worthwile.

Appendix

Lemma 6.1 *Let $w^0, w^1, \ldots, w^m$ be affinely independent points in $\Re^{n+1}$. Given $\delta_i \in \Re$, define the points $\hat{w}^0, \hat{w}^1, \ldots, \hat{w}^m$ as*

$$\hat{w}^i = w^i + \delta^i(q^1 - q^0),$$

for $i = 0, 1, \ldots, m$ with $q^1 \neq q^0$ such that

$$q^0 = w^0 + \sum_{h=1}^{m} \lambda^0_h (w^h - w^0)$$

and

$$q^1 = w^0 + \sum_{h=1}^{m} \lambda^1_h (w^h - w^0).$$

Let $g \in \{0, 1, \ldots, m\}$ be such that $\lambda^0_g \neq \lambda^1_g$. Then the points $\hat{w}^i$, $i = 0, 1, \ldots, g-1, g+1, \ldots, m$, are affinely independent.

Proof:

Let β_i, $i = 0, 1, \ldots, g-1, g+1, \ldots, m$, be such that

$$\sum_{i=0,\ i\neq g}^{m} \beta_i \hat{w}^i, \qquad \sum_{i=0,\ i\neq g}^{m} \beta_i = 0.$$

Then we have to prove that $\beta_i = 0$ for all $i \neq g$. Substituting $\hat{w}^i$ into this expression gives

$$\sum_{i=0,\ i\neq g}^{m} \beta^i w^i + \left(\sum_{i=0,\ i\neq g}^{m} \beta_i \delta^i \right) \left(\sum_{h=1}^{m} (\lambda_h^1 - \lambda_h^0)(w^h - w^0) \right) = 0.$$

Defining δ as

$$\delta = \sum_{i=0,\ i\neq g}^{m} \beta_i \delta^i,$$

this expression reduces to

$$\sum_{i=0,\ i\neq g}^{m} \beta_i w^i + \delta \sum_{h=1}^{m} (\lambda_h^1 - \lambda_h^0)(w^h - w^0) = 0.$$

Rewriting this result in a suitable way gives

$$\sum_{i=1,\ i\neq g}^{m} (\beta_i + \delta(\lambda_i^1 - \lambda_i^0))w^i + (\beta_0 - \delta \sum_{h=1}^{m} (\lambda_h^1 - \lambda_h^0))w^0 + \delta(\lambda_g^1 - \lambda_g^0)w^g = 0.$$

Let

$$\begin{aligned} \gamma_i &= \beta_i + \delta(\lambda_i^1 - \lambda_i^0) \text{ for all } i \neq 0, g, \\ \gamma_0 &= \beta_0 - \delta \sum_{h=1}^{m} (\lambda_h^1 - \lambda_h^0), \\ \gamma_g &= \delta(\lambda_g^1 - \lambda_g^0). \end{aligned}$$

Then

$$\sum_{i=0}^{m} \gamma_i = \sum_{i=0,\ i\neq g}^{m} \beta_i = 0$$

and

$$\sum_{i=0}^{m} \gamma_i w^i = 0.$$

By affine independence of $w^0, w^1, \ldots, w^m$ it follows that $\gamma_i = 0,\ i = 0, 1, \ldots, m$. Therefore

$$\beta_i + \delta(\lambda_i^1 - \lambda_i^0) = 0 \text{ for all } i \neq 0, g,$$
$$\beta_0 - \delta \sum_{h=1}^{m} (\lambda_h^1 - \lambda_h^0) = 0,$$
$$\delta(\lambda_g^1 - \lambda_g^0) = 0.$$

We took $g \in \{0, 1, \ldots, m\}$ such that $\lambda_g^1 - \lambda_g^0 \neq 0$. Therefore $\delta = 0$. But then $\beta_i = 0$ for all $i \neq g$.

□

References

[1] G. Debreu, *Theory of Value* (Yale University Press, New Haven, 1959).

[2] B.C. Eaves, "Thoughts on computing market equilibrium with *SLCP*," in: A. Talman and G. van der Laan, eds., *The Computation and Modelling of Economic Equilibria* (North-Holland, Amsterdam, 1987).

[3] K. Kamiya and A.J.J. Talman, "Linear Stationary Point Problems," CentER Discussion Paper 9022, (Center for Economic Research, Tilburg University Tilburg, The Netherlands, 1990), to appear in Mathematical Programming.

[4] C.E. Lemke, "Bimatrix equilibrium points and mathematical programming," *Management Science* 11 (1965) 681-689.

[5] C.E. Lemke and J.T. Howson Jr., "Equilibrium points of bimatrix games," *SIAM Journal of Applied Mathematics* 12 (1964) 413-423.

[6] L. Mathiesen, "Computation of economic equilibria by a sequence of linear complementarity problems," *Mathematical Programming Study* 23 (1985) 144-162.

[7] H. Scarf, "Some examples of global instability of the competitive equilibrium," *International Economic Review* 1 No.3 (1960) 157-172.

[8] A.J.J. Talman and Y. Yamamoto, "A simplicial algorithm for stationary point problems on polytopes," *Mathematics of Operations Research* 14 (1989) 383-399.

[9] Y. Yamamoto, "A path following algorithm for stationary point problems," *Journal of the OR Society of Japan* 30 (1987) 181-198.

IN 1990 REEDS VERSCHENEN

419 Bertrand Melenberg, Rob Alessie
A method to construct moments in the multi-good life cycle consumption model

420 J. Kriens
On the differentiability of the set of efficient (μ,σ^2) combinations in the Markowitz portfolio selection method

421 Steffen Jørgensen, Peter M. Kort
Optimal dynamic investment policies under concave-convex adjustment costs

422 J.P.C. Blanc
Cyclic polling systems: limited service versus Bernoulli schedules

423 M.H.C. Paardekooper
Parallel normreducing transformations for the algebraic eigenvalue problem

424 Hans Gremmen
On the political (ir)relevance of classical customs union theory

425 Ed Nijssen
Marketingstrategie in Machtsperspectief

426 Jack P.C. Kleijnen
Regression Metamodels for Simulation with Common Random Numbers: Comparison of Techniques

427 Harry H. Tigelaar
The correlation structure of stationary bilinear processes

428 Drs. C.H. Veld en Drs. A.H.F. Verboven
De waardering van aandelenwarrants en langlopende call-opties

429 Theo van de Klundert en Anton B. van Schaik
Liquidity Constraints and the Keynesian Corridor

430 Gert Nieuwenhuis
Central limit theorems for sequences with m(n)-dependent main part

431 Hans J. Gremmen
Macro-Economic Implications of Profit Optimizing Investment Behaviour

432 J.M. Schumacher
System-Theoretic Trends in Econometrics

433 Peter M. Kort, Paul M.J.J. van Loon, Mikulás Luptacik
Optimal Dynamic Environmental Policies of a Profit Maximizing Firm

434 Raymond Gradus
Optimal Dynamic Profit Taxation: The Derivation of Feedback Stackelberg Equilibria

435 Jack P.C. Kleijnen
Statistics and Deterministic Simulation Models: Why Not?

436 M.J.G. van Eijs, R.J.M. Heuts, J.P.C. Kleijnen
Analysis and comparison of two strategies for multi-item inventory systems with joint replenishment costs

437 Jan A. Weststrate
Waiting times in a two-queue model with exhaustive and Bernoulli service

438 Alfons Daems
Typologie van non-profit organisaties

439 Drs. C.H. Veld en Drs. J. Grazell
Motieven voor de uitgifte van converteerbare obligatieleningen en warrantobligatieleningen

440 Jack P.C. Kleijnen
Sensitivity analysis of simulation experiments: regression analysis and statistical design

441 C.H. Veld en A.H.F. Verboven
De waardering van conversierechten van Nederlandse converteerbare obligaties

442 Drs. C.H. Veld en Drs. P.J.W. Duffhues
Verslaggevingsaspecten van aandelenwarrants

443 Jack P.C. Kleijnen and Ben Annink
Vector computers, Monte Carlo simulation, and regression analysis: an introduction

444 Alfons Daems
"Non-market failures": Imperfecties in de budgetsector

445 J.P.C. Blanc
The power-series algorithm applied to cyclic polling systems

446 L.W.G. Strijbosch and R.M.J. Heuts
Modelling (s,Q) inventory systems: parametric versus non-parametric approximations for the lead time demand distribution

447 Jack P.C. Kleijnen
Supercomputers for Monte Carlo simulation: cross-validation versus Rao's test in multivariate regression

448 Jack P.C. Kleijnen, Greet van Ham and Jan Rotmans
Techniques for sensitivity analysis of simulation models: a case study of the CO_2 greenhouse effect

449 Harrie A.A. Verbon and Marijn J.M. Verhoeven
Decision-making on pension schemes: expectation-formation under demographic change

IN 1991 REEDS VERSCHENEN

466 Prof.Dr. Th.C.M.J. van de Klundert - Prof.Dr. A.B.T.M. van Schaik
Economische groei in Nederland in een internationaal perspectief

467 Dr. Sylvester C.W. Eijffinger
The convergence of monetary policy - Germany and France as an example

468 E. Nijssen
Strategisch gedrag, planning en prestatie. Een inductieve studie binnen de computerbranche

469 Anne van den Nouweland, Peter Borm, Guillermo Owen and Stef Tijs
Cost allocation and communication

470 Drs. J. Grazell en Drs. C.H. Veld
Motieven voor de uitgifte van converteerbare obligatieleningen en warrant-obligatieleningen: een agency-theoretische benadering

471 P.C. van Batenburg, J. Kriens, W.M. Lammerts van Bueren and R.H. Veenstra
Audit Assurance Model and Bayesian Discovery Sampling

472 Marcel Kerkhofs
Identification and Estimation of Household Production Models

473 Robert P. Gilles, Guillermo Owen, René van den Brink
Games with Permission Structures: The Conjunctive Approach

474 Jack P.C. Kleijnen
Sensitivity Analysis of Simulation Experiments: Tutorial on Regression Analysis and Statistical Design

475 An O($nlogn$) algorithm for the two-machine flow shop problem with controllable machine speeds
C.P.M. van Hoesel

476 Stephan G. Vanneste
A Markov Model for Opportunity Maintenance

477 F.A. van der Duyn Schouten, M.J.G. van Eijs, R.M.J. Heuts
Coordinated replenishment systems with discount opportunities

478 A. van den Nouweland, J. Potters, S. Tijs and J. Zarzuelo
Cores and related solution concepts for multi-choice games

479 Drs. C.H. Veld
Warrant pricing: a review of theoretical and empirical research

480 E. Nijssen
De Miles and Snow-typologie: Een exploratieve studie in de meubelbranche

481 Harry G. Barkema
Are managers indeed motivated by their bonuses?

482 Jacob C. Engwerda, André C.M. Ran, Arie L. Rijkeboer
Necessary and sufficient conditions for the existence of a positive definite solution of the matrix equation $X + A^T X^{-1} A = I$

483 Peter M. Kort
A dynamic model of the firm with uncertain earnings and adjustment costs

484 Raymond H.J.M. Gradus, Peter M. Kort
Optimal taxation on profit and pollution within a macroeconomic framework

485 René van den Brink, Robert P. Gilles
Axiomatizations of the Conjunctive Permission Value for Games with Permission Structures

486 A.E. Brouwer & W.H. Haemers
The Gewirtz graph - an exercise in the theory of graph spectra

487 Pim Adang, Bertrand Melenberg
Intratemporal uncertainty in the multi-good life cycle consumption model: motivation and application

488 J.H.J. Roemen
The long term elasticity of the milk supply with respect to the milk price in the Netherlands in the period 1969-1984

489 Herbert Hamers
The Shapley-Entrance Game

490 Rezaul Kabir and Theo Vermaelen
Insider trading restrictions and the stock market

491 Piet A. Verheyen
The economic explanation of the jump of the co-state variable

492 Drs. F.L.J.W. Manders en Dr. J.A.C. de Haan
De organisatorische aspecten bij systeemontwikkeling
een beschouwing op besturing en verandering

493 Paul C. van Batenburg and J. Kriens
Applications of statistical methods and techniques to auditing and accounting

494 Ruud T. Frambach
The diffusion of innovations: the influence of supply-side factors

495 J.H.J. Roemen
A decision rule for the (des)investments in the dairy cow stock