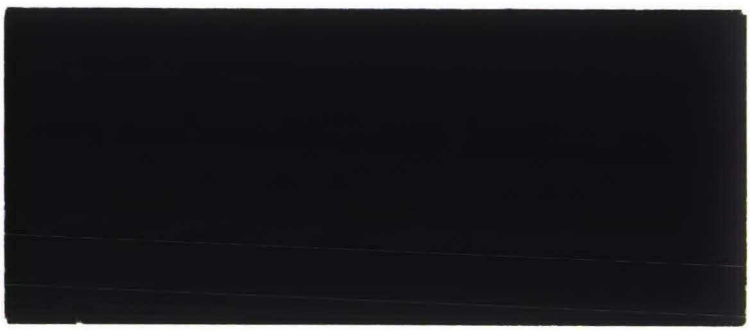
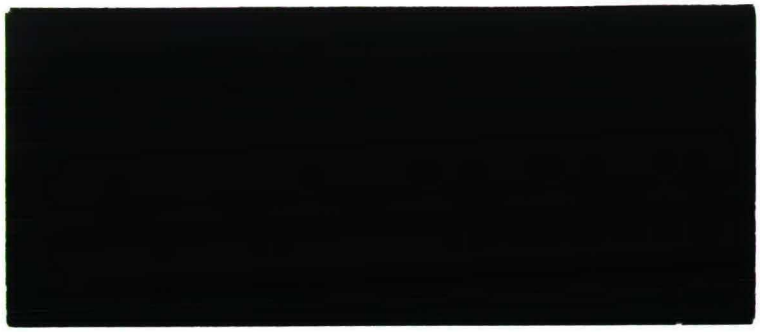
Computing economic equilibria by variable dimension algorithms

van der Laan, G.; Talman, A.J.J.

Publication date:
1987

Document Version
Publisher's PDF, also known as Version of record





COMPUTING ECONOMIC EQUILIBRIA BY VARIABLE DIMENSION ALGORITHMS: STATE OF THE ART

G. van der Laan and A.J.J. Talman

FEW 270

COMPUTING ECONOMIC EQUILIBRIA BY VARIABLE DIMENSION ALGORITHMS: STATE OF THE ART*)

by

G. van der Laan[1] and A.J.J. Talman[2]

August 1987.

*) This research is part of the VF-program "Equilibrium and disequilibrium in demand and supply", which has been approved by the Netherlands Ministry of Education and Sciences. This paper was presented at the Workshop on Mathematical Economics organized by the European University Institute in San Miniato, 8-19 September 1986.

1) Department of Economics and Econometrics, Free University, de Boelelaan 1105, 1081 HV Amsterdam, The Netherlands.

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

COMPUTING ECONOMIC EQUILIBRIA BY VARIABLE DIMENSION ALGORITHMS: STATE OF THE ART.

1. Introduction.

Since the originating work of **Scarf** [46,47] and **Kuhn** [19,20], many algorithms have been developed to compute a Walras' equilibrium in a general equilibrium model. In this paper we intend to give an exposition on simplicial algorithms for finding an equilibrium price system and their interpretation as a price adjustment mechanism. In this exposition we are mainly concerned with variable dimension restart algorithms. Such an algorithm was first introduced by the authors in 1979.

In the fifties **Debreu** [2], **Gale** [14], **McKenzie** [37,38], and others developed a theory from basic axioms on the existence of a market clearing price system in an economy where the agents act as price takers. This theory answered the first question of Walras' research programme, see **Ingrao** and **Israel** [15], but did not say anything about the computation of such a market clearing price system. A further question in Walras' program was the existence of an effective price mechanism, that is a globally and universally converging price adjustment mechanism. A mechanism is globally convergent if it converges to some price equilibrium from any starting point. A mechanism is universally if it converges for any economy which is described in terms of standard quasi-concave utility functions. We know that the classical Walras' tatonnement process may fail to converge to a vector of equilibrium prices, even when the set of initial price systems is restricted. In fact, neither global nor local convergence can be guaranteed. So, the mechanism is not effective (see **Saari** and **Simon** [42]). Counterexamples where the prices can spiral forever have been constructed by **Scarf** [45]. **Sonnenschein** [52] proved that any continuous function satisfying Walras' law can be realized as the excess demand function for some pure exchange economy. So, we need a process that converges universally and globally, i.e., a process that converges for any continuous function satisfying Walras' law and from any initial starting price vector.

While the classical Walras' tatonnement process does not converge universally, for Smale's global Newton process (see **Smale** [50]) convergency may not hold for an arbitrarily chosen starting point. In [42] it is shown that an effective mechanism needs information about both the excess demand at any price and the value of the gradients of all except one of its component functions. **Saari** [41] further concludes that there does not exist an iterative, globally and universally convergent mechanism which depends

only upon a finite amount of information about the excess demand function and its derivatives. So, other information is required to design a global and universal iterative price mechanism. Saari suggests that a mechanism which depends on values of prices could be a possible option. This brings us to Scarf-type algorithms. In such an algorithm an approximate equilibrium price system is computed by generating a sequence of adjacent simplices in a simplicial subdivision of the unit price simplex. Each new simplex is obtained from the previous simplex by replacing one of its vertices by a new vertex. At each new vertex the excess demand function is evaluated. This evaluation determines the next vertex to be replaced. This vertex and the location of the simplex uniquly determine the next simplex to be considered by the algorithm.

In this paper we expose Scarf-type algorithms to find an equilibrium price vector in an exchange economy. We are mainly concerned with algorithms on the unit price simplex. However, we should be aware about the generalizations of these algorithms to other spaces, namely the Euclidean space and the product space of several unit price simplices. These generalizations allow us to handle with more complex general equilibrium models or to utilize some specific structure in pure exchange models.

The original Scarf algorithm to find an approximate equilibrium price vector starts at a corner of the unit price simplex. Also the two algorithms of Kuhn start at the boundary of the unit simplex. The accuracy of the approximate solution generally increases if the mesh of the underlying simplicial subdivision decreases. If a given approximate solution is found to be of insufficient accuracy, the subdivision needs to be refined. Since the algorithms of Scarf and Kuhn have to start on the boundary of the unit simplex the computational results are of slow speed. To overcome this inefficiency **Eaves** [9] presented a simplicial algorithm which continuously refines the subdivision by embedding the unit simplex in a one higher dimensional space. This has given a vast improvement in computational speed.

Another approach to computing approximate solutions of increasing accuracy is the use of a restart algorithm. A restart algorithm is an algorithm which can be initiated at an arbitrary grid point. Successive restarts with subdivisions having decreasing mesh sizes yield increasingly more accurate solutions. Every restart is initiated at or close to the previous found approximate solution. These methods are used now in virtually all practical applications.

Merrill [39] first introduced a restart algorithm for solving systems of nonlinear equations. **Kuhn** and **MacKinnon** [21] proposed a similar algorithm for fixed point problems on the unit simplex. The restart possibility is obtained by introducing an additional dimension and embedding the unit simplex in the product of itself and the

interval [0,1]. In a subdivision of this product space in simplices the algorithm traces a sequence of simplices. It starts with a simplex having a facet on the zero-level containing the unique, known solution of a well-defined artificial problem, and the sequence ends with a simplex having a facet on the one-level containing an approximate solution to the real problem.

Whereas the algorithms of Scarf and Kuhn generate a sequence of adjacent n-dimensional simplices in a subdivision of the n-dimensional unit simplex, the algorithms of Eaves and of Kuhn and MacKinnon trace a path of (n+1)-dimensional simplices. The restart algorithm of **van der Laan** and **Talman** [27] bypasses the introduction of an artificial dimension and traces a path of simplices of varying dimension in the subdivision of the unit simplex. This path starts at an arbitrary grid point representing a zero-dimensional simplex, and terminates at a full-dimensional simplex containing an approximate solution point. The attractiveness of this restart method lies in the fact that movements with simplices of varying dimension in the n-dimensional unit simplex are typically faster than movements with full-dimensional simplices in an (n+1)-dimensional set.

In the algorithm of van der Laan and Talman the function value at the starting point determines a unique ray out of n+1 possible rays along which the starting point is left. The directions in which the rays point are induced by the underlying simplicial subdivision of the unit simplex. Simplicial subdivisions introduced in **van der Laan** and **Talman** [29] and **Doup** and **Talman** [5] yield different directions. This has resulted in both an improvement of the computational efficiency and a more reasonable interpretation of the path followed by the algorithm as a price adjustment mechanism. A further development can be found in **Doup, van der Laan** and **Talman** [7] in which an algorithm is introduced having $2^{n+1}-2$ rays to leave the starting point.

This work has been extended into several directions. In **van der Laan** and **Talman** [28] and in **Todd** [55] an (n+1)-ray variable dimension simplicial algorithm for solving systems of n-dimensional nonlinear equations has been given. Further results in designing simplicial algorithms for solving this problem can be found in **van der Laan** and **Talman** [31, 32], **Reiser** [40], **Todd** [57], **Wright** [59], **Kojima** and **Yamamoto** [17,18], **Saigal** [44], **Yamamoto** [60], **van der Laan** and **Seelen** [26], and **Broadie** [1]. Variable dimension simplicial algorithms also have been developed for solving the nonlinear complementarity problem on the product space of several unit simplices. Therefore we refer to **van der Laan** and **Talman** [33], **Doup** and **Talman** [5], **van der Laan, Talman** and **Van der Heyden** [35], **Freund** [13], and **Doup, van den Elzen** and **Talman** [6].

The work of **van der Laan** and **Talman** [34], **van den Elzen, van der Laan** and **Talman** [11], and **van den Elzen** and **van der Laan** [10] deals with the interpretation of the various algorithms as price adjustment mechanisms. These adjustment processes are governed by relating the value of the excess demand to the location of the corresponding price vector with respect to the initial price vector. We will see that these processes are both effective and economicly meaningfull as an alternative for the classical Walras' tatonnement process.

In this paper we survey the simplicial algorithms literature mentioned above. The paper is organized as follows. In the next section we formulate the problem of finding equilibrium prices on the unit price simplex for a general pure exchange economy model. The basic idea of simplicial algorithms on the unit simplex is given in Section 3 by exposing the algorithm of Scarf. Moreover, in that section we handle briefly with the restart method of Kuhn and MacKinnon and the continuous deformation method of Eaves. The basic idea of a variable dimension restart algorithm on the unit simplex is given in Section 4. Section 5 deals with vector labelling algorithms. Section 6 considers the paths followed by the algorithms as price adjustment mechanisms. Finally, in Section 7 we discuss the generalization for solving problems on the product space of several unit simplices.

2. General pure exchange economy model.

In this paper we deal with excess demand functions on the n-dimensional unit simplex

$$S^n = \{x \in R^{n+1} \mid \Sigma_j x_j = 1,\ x_j \geq 0,\ j = 1,\ldots,n+1\}.$$

In case of a competitive exchange economy with n+1 commodities, S^n is the price simplex with the sum of the prices normalized to one. Suppose we have an economy with m consumers and for each consumer i = 1,...,m holds

a) the consumption set X^i is a compact, convex subset of R_+^{n+1}, containing the set

$$\{x \in R^{n+1} \mid 0 \leq x_j \leq w_j,\ j = 1,\ldots,n+1\},$$

where w^i is the (n+1)-vector of initial endowments of consumer i and $w_j = \Sigma_i w^i{}_j$

b) $w^i{}_j > 0$ for all i,j

c) the preferences of the consumers are continuous, monotonic and strictly convex.

Let $x^i(p)$ be the demand of consumer i given price $p \in S^n$, i.e., $x^i(p)$ is preferred by i to all other consumptions x^i subject to $x \in X^i$ and $p^\top x \leq p^\top w^i$. Then the (total) excess demand function z defined by $z(p)=\Sigma_i(x^i(p)-w^i)$ belongs to the class of continuously differentiable functions from S^n to R^{n+1} satisfying

i) for all $p \in S^n$, $p^\top z(p)=0$ (Walras' law)

ii) $z_j(p) \geq 0$ if $p_j=0$ (nonnegative excess demand at zero price).

A price vector p^* is called an equilibrium price vector if $z(p^*)=0$, i.e., if p^* is a zero point of z.

From **Sonnenschein** [52] and **Debreu** [3] we know that any continuously differentiable function satifying i) and ii) can be obtained as an excess demand function of some pure exchange economy. So, we want to have computational procedures and price adjustment mechanisms that are effective for this whole class of functions. In this paper we will even allow for a more general class of functions. Clearly the next definition contains the class of functions given above.

Definition 2.1. A continuous function $z: S^n \to R^{n+1}$ is an excess demand function if there exists a nonnegative function $y: S^n \to R^{n+1}$ such that

i) for all $p \in S^n$, $y^\top(p)z(p)=0$

ii) $y_i(p)>0$ if $p_i>0$

iii) $z_j(p) \geq 0$ if $p_j=0$.

Example. (Price rigidities). Suppose that for an exchange economy with m consumers and n+1 commodities the conditions i)-iii) above hold. Further assume that the set of admissible prices is given by

$$P = \{p \in R_+^{n+1} \mid 0 < p^l_j \leq p_j \leq p^u_j \text{ for all } j\}.$$

Clearly, P does not necessarily contain a vector p^* such that $z(p^*)=0$. **Drèze** [8] defined an equilibrium concept with quantity constraints on the excess supplies and the excess demands. The existence of an equilibrium with quantity constraints on the supplies only has been proven by **van der Laan** [24,25] and **Kurz** [22]. Such a supply-constrained equilibrium is an allocation x^i, i=1,...,m, a price vector $p \in P$, and a rationing scheme $r \leq 0$ such that

S1) for all i, x^i is a maximal element with respect to the preferences of i in the set

$$B^i(p,r) = \{x \in X^i \mid p^\top x \leq p^\top w^i,\ x - w^i \geq r\}$$

S2) $\Sigma_i\, x^i = \Sigma_i\, w^i$

S3) $r_j = -\infty$ if $p_j > p^l_j$ $\qquad j=1,\ldots,n+1$

S4) $r_j = -\infty$ for at least one j.

To show the existence of such an equilibrium we construct an excess demand function for which a zero point yields a supply-constrained equilibrium. For $q \in S^n$, let p(q) and r(q) be defined by

$$p_j(q) = \max\,[p^l_j,\ q_j p^u_j / \max_h q_h] \qquad j=1,\ldots,n+1$$

$$r_j(q) = -\min\,[1,\ q_j p^u_j / \max_h q_h] w_j \qquad j=1,\ldots,n+1.$$

Let $x^i(q)$ be the optimal consumption in the budget set

$$B^i(q) = \{x \in X^i \mid p^\top(q)x \leq p^\top(q)w^i,\ x - w^i \geq r(q)\}$$

and let $z(q) = \Sigma_i x^i(q) - w$. From the conditions i)-iii) it follows that x^i is a continuous function of q and satisfies $p^\top(q)x^i(q) = p^\top(q)w^i$. Hence z is a continuous function from S^n into R^{n+1} satisfying $y^\top(q)z(q) = 0$ for all $q \in S^n$, with $y(q) = p(q) > 0$. Furthermore, $q_j = 0$ implies $r_j(q) = 0$ and hence $z_j(q) \geq 0$. So, z is an excess demand function. Clearly, $x^i(q^*)$, $i=1,\ldots,m$, $p(q^*)$ and $r(q^*)$ is a supply-constrained equilibrium iff $z(q^*) = 0$.

The example shows that Definition 2.1 covers excess demand functions z which may arise both from an economy with flexible prices (Walrasian) as well as from an economy with bounded prices.

The existence of a zero point of a given excess demand function on S^n follows from the theory of fixed points. Any continuous function which maps the unit simplex into itself has according to Brouwer's theorem a fixed point. It is well-known how to construct such a function for which x^* is a fixed point if and only if x^* is a zero point of z. In fact, the existence of a zero point of an excess demand function is equivalent to the existence of a fixed point of a mapping from the unit simplex into itself (see e.g. **Uzawa** [58] or **Scarf** [48]). The fixed point theorem is also equivalent with the intersection theorem of **Knaster, Kuratowski** and **Mazurkiewicz** [16] on the unit simplex. A dual analogue of this theorem can be found **in Scarf** [47] and can be stated as follows.

Theorem 2.2. (Scarf's intersection theorem). Let $C_1,...,C_{n+1}$ be n+1 closed subsets of S^n, with the properties:

a) S^n is covered by the union of all sets C_i, i=1,...,n+1

b) for each p in S^n, $p_i=0$ implies $p\in C_i$, i.e., the set of points for which the i-th component is equal to zero is contained in C_i, for i=1,...,n+1.

Then the intersection of all sets C_i is not empty.

A constructive proof of the theorem will be given in the next section. We use the intersection theorem to prove that any excess demand has a zero point.

Theorem 2.3. Let z be an excess demand function. Then there exists a p^* in S^n such that $z(p^*)=0$.

Proof. For i=1,...,n+1, let C_i be defined by

$$C_i = \{p\in S^n | p_i=0 \text{ or } z_i(p) = \max_h z_h(p)\}. \qquad (2.1)$$

Clearly, these sets satisfy the conditions of Theorem 2.2 and hence there exists an intersection point p^*. From the definition of the sets C_i it follows that for each i, $z_i(p^*)=\max_h z_h(p^*)$ if p^*_i is positive. Suppose that $p^*_i>0$ for all i. Then $y^\top(p^*)z(p^*)=\Sigma_i y_i(p^*)\max_h z_h(p^*)=0$. Since $y_i(p)>0$ if $p_i>0$ we obtain that $z_i(p^*)=\max_h z_h(p^*)=0$. If, for some i, $p^*_i=0$, then $\max_h z_h(p^*)\geq 0$ because $z_i(p^*)\geq 0$, so that $z_j(p^*)\geq 0$ for all j. Since for at least one j, $p^*_j>0$ and hence also $y_j(p^*)>0$, it follows from $y^\top(p^*)z(p^*)=0$ and $yi(p^*)\geq 0$ for all i, that $\max_h z_h(p^*)=0$. Consequently $z(p^*)=0$, which proves the theorem.

The proof of Theorem 2.3 shows that an equilibrium price vector is an intersection point of the sets C_i defined in (2.1). In the next sections we will see that the basic idea of a simplicial algorithm is to find such an intersection point approximately, i.e., to approximate a zero point of z by a point lying close to all C_i's.

We conclude this section with a more general existence theorem.

Definition 2.4. A continuous function z: $S^n\to R^{n+1}$ is a generalized excess demand function if there exists a nonnegative function y: $S^n\to R^{n+1}$ such that

i) for all $p\in S^n$, $y^\top(p)z(p)=0$

ii) $y_i(p)>0$ if $p_i>0$

iii) if $p_j=0$, then $z_j(p)\geq 0$ or $y_j(p)=0$.

Theorem 2.5. Let z be a general excess demand function. Then there exists a p^* in S^n such that $z(p^*)\leq 0$.

Proof. Again the proof follows from the intersection theorem by taking the sets C_i as defined in (2.1). If p^* is positive, then again $z_i(p^*)=\max_h z_h(p^*)=0$ for all i. If, for some i, $p^*_i=0$ and $z_i(p^*)\geq 0$, then $\max_h z_h(p^*)\geq 0$, and with condition iii) we have that $y_j(p^*)z_j(p^*)\geq 0$ for all j. From condition i) it follows that $\max_h z_h(p^*)=0$ and hence that $z(p^*)\leq 0$. Finally, if $y_i(p^*)=0$ for all i with $p^*_i=0$, it follows from $y^T(p^*)z(p^*)=0$ and condition ii) that $\max_h z_h(p^*)=0$.

We notice that for a solution point p^*, $z_i(p^*)$ can be less than zero only if $p^*_i=0$. Therefore we call this existence problem the NonLinear Complementarity Problem (NLCP) on S^n. By taking $C_i=\{p\in S^n | z_i(p) = \max_h z_h(p)\}$, i=1,...,n+1, a point p^* is a solution to the NLCP with respect to z iff for all i, $p^*_i=0$ or $p^*\in C_i$. Clearly, these sets C_i do not satisfy condition b) of Theorem 2.2. However, from the intersection theorem the next corollary follows immediately (see also **Freund** [13] or **van der Laan, Talman** and **Van der Heyden** [35]).

Corollary 2.6. (Generalized intersection theorem). Let $C_1,...,C_{n+1}$ be n+1 closed, possibly empty, subsets of S^n covering S^n. Then there exists a point p^* with for all i, $p^*_i=0$ or $p^*\in C_i$.

3. Simplicial algorithms; the basic idea.

In this section we expose a simplicial algorithm for finding an intersection point that is closely related to the original algorithm of Scarf, see **Scarf** [46,47]. First some notation is introduced.

The vertices of S^n are denoted by e^i, the i-th unit vector in R^{n+1}, i=1,...,n+1. A t-dimensional simplex or t-simplex in R^{n+1} is the convex hull of t+1 linearly independent points in R^{n+1}, called the vertices of the simplex. A t-simplex σ with vertices $v^1,...,v^{t+1}$ is denoted by $\sigma(v^1,...,v^{t+1})$. A k-face of a t-simplex σ, $k\leq t$, is the convex hull of k+1 vertices of σ. A k-face of σ is called a facet of σ if k=t-1. The facet $\tau(v^1,...,v^{i-1},v^{i+1},...,v^{t+1})$ of σ is denoted by $\tau(v^{-i})$. The facet of S^n with $x_k=0$ is denoted by S^n_k, k=1,...,n+1.

Definition 3.1. A finite collection of n-simplices with vertices in S^n is a simplicial subdivision or triangulation of S^n if

i) S^n is the union of the simplices

ii) the intersection of any two simplices is either empty or a common face of both of them.

By the restrictions i) and ii) a simplicial subdivision has the property that any two simplices can not have interior points in common. Moreover, any facet of a simplex is either a facet of just one other simplex or lies in the boundary of S^n and is not a facet of any other simplex.

Together with the concept of a simplicial subdivision the concept of a labelling plays a central role in the theory of simplicial algorithms. Let I_{n+1} denote the set of integers $\{1,...,n+1\}$.

Definition 3.2. A labelling function l on S^n assigns to each element $x \in S^n$ a label $l(x) \in I_{n+1}$.

Definition 3.3. An n-simplex $\sigma(v^1,...,v^{n+1})$ is completely labelled if $I_{n+1} = \{l(v^1),...,l(v^{n+1})\}$.

Definition 3.4. A facet $\tau(v^{-i})$ of an n-simplex $\sigma(v^1,...,v^{n+1})$ is almost-complete if $\{l(v^1),...,l(v^{i-1}),l(v^{i+1}),...,l(v^{n+1})\}=\{2,...,n+1\}$.

Definition 3.5. An n-simplex is almost-complete if it has at least one almost-complete facet.

Since an almost-complete n-simplex bears the labels 2,...,n+1 on its n+1 vertices, at most one of these labels occurs twice. So, the next lemma follows immediately.

Lemma 3.6. An almost complete simplex is either completely labelled or has just two almost-complete facets.

In applying a simplicial algorithm for finding an intersection point of n+1 sets satisfying the conditions of Theorem 2.2, an appropiate labelling function is defined such that a completely labelled simplex yields an approximate intersection point. A simplicial algorithm searches for a completely labelled simplex. The existence of a completely labelled simplex is guaranteed by asserting a properness condition for the labelling on the boundary of S^n.

Lemma 3.7. (Sperner's lemma [51]) Let l be a labelling on S^n satisfying $l(x) \neq i$ if $x \in S^n_i$, $i=1,...,n+1$. Then a simplicial subdivision of S^n contains a completely labelled simplex.

The properness condition $l(x)\neq i$ if $x_i=0$ implies that $l(e^i)=i$, $i=1,\ldots,n+1$. The Sperner-properness condition is enough for the existence of a completely labelled simplex. In the algorithm of Scarf a dual properness condition is utilized.

Dual Properness Condition: A labelling is said to be dual proper or "Scarf proper" when $l(x)\in\{i\in I^{n+1} | x_i=0\}$ if x is on the boundary of S^n.

This condition does not guarantee that each label appears at least once on the set of vertices. The subdivision must be sufficiently fine in order to guarantee a completely labelled simplex. For instance, if there is only one simplex in the subdivision, the simplex S^n itself, then we can easily assign a label to each e^i, such that S^n is not completely labelled.

Lemma 3.8. (Dual Sperner's lemma). Let l be a dual proper labelling and assume that for a simplicial subdivision no simplex has a non-empty intersection with every facet $S^n{}_k$. Then there exists at least one completely labelled simplex.

The lemma can be proved by embedding S^n in a larger simplex (see **Scarf [48]**). Here we prove the lemma for a specific triangulation and labelling. When applying Scarf's algorithm for this triangulation and labelling, a path of simplices is generated, which terminates with a completely labelled simplex.

Let G be a triangulation of S^n having, for some $0<\alpha<1$, the convex hull σ^0 of $w^1=e^1$ and $w^i=\alpha e^1+(1-\alpha)e^i$, $i=2,\ldots,n+1$ as a simplex (see figure 1). Moreover, let l^* be a dual proper labelling such that for x on the boundary of S^n, $l^*(x)=i-1$ with i the least index for which $x_{i-1}=0$ and $x_i>0$ where $i-1=n+1$ if $i=1$. Now, we have the following properties:

i) σ^0 is the only simplex of G having e^1 as a vertex
ii) $l^*(w^1)=l^*(w^2)=n+1$, $l^*(w^i)=i-1$, $i=3,\ldots,n+1$, and hence $\tau(w^{-1})$ and $\tau(w^{-2})$ are the two almost-complete facets of σ^0
iii) the facet $\tau(w^{-2})$ of σ^0 is the only almost-complete facet in the boundary of S^n.

We are now ready to describe the algorithm which finds for a labelling l^* a completely labelled simplex in a simplicial subdivision G having σ^0 as a simplex. The algorithm starts with σ^0, leaves this simplex through the almost-complete facet $\tau(w^{-1})$, and finds the unique simplex σ^1 sharing this facet as a common facet with σ^0. If the vertex, say v, of σ^1 opposite $\tau(w^{-1})$ has label 1, then σ^1 is completely labelled and the

algorithm terminates. If $l(v)=j$, $j\neq 1$, then the facet of σ^1 opposite the vertex w^i ($i>1$) with label j is also almost-complete and the algorithm proceeds with the unique simplex σ^2 having the almost-complete facet opposite w^i in common with σ^1. The algorithm continues by determining in each new simplex σ the label of the vertex of σ opposite the facet shared with the previous simplex and going to the unique simplex having the facet opposite the vertex with the same label in common with σ, until a vertex carries label 1. In such a way a sequence of adjacent simplices is generated from σ^0 in which each pair of adjacent simplices has an almost-complete facet in common. The door-in door-out principle which first appeared in **Lemke** and **Howson** [36] proves that the sequence terminates with a completely labelled simplex within a finite number of steps. The finiteness argument is based on the fact that no simplex can be visited more than once. Since the number of simplices is finite the algorithm must then terminate, either with a completely labelled simplex or with a simplex having the almost-complete facet opposite the vertex to be replaced in the boundary of S^n. However, according to property iii) only the facet $\tau(w^{-2})$ of σ^0 lies in the boundary of S^n. So, in this case the algorithm must have been returned in the starting simplex. This simplex can only be entered through the almost-complete facet in common with σ^1 and hence the algorithm must also have been returned in σ^1. Now, suppose that in the sequence of generated simplices σ^j, $j=1,2,...$, for some h, $h\geq 2$, all simplices σ^j, $j=1,...,h-1$, are different and that $\sigma^h = \sigma^i$ for some i, $1\leq i<h$. Since σ^i can only be entered through the two almost-complete facets in common with the adjacent simplices σ^{i-1} or σ^{i+1} we must have that either $\sigma^{h-1} = \sigma^{i-1}$ or $\sigma^{h-1} = \sigma^{i+1}$. Unless $h-1=i+1$, this contradicts the fact that all simplices up to σ^h are different. On the other hand, σ^{i+1} has been entered and left through two different almost-complete facets which excludes that $\sigma^{i+2} = \sigma^i$ and hence each simplex can nly be visited once. So, seeing almost-complete simplices as rooms and the almost-complete facets as doors, the door-in door-out principle shows that the algorithm must terminate with a room having one door, being a completely labelled simplex. This shows the existence of a completely labelled simplex for a dual labelling l^* and a triangulation G having σ^0 as a simplex. In Figure 1 the algorithm is illustrated for n=2.

The existence of a completely labelled simplex immediately proves the existence of an intersection point for sets C_i satisfying the conditions of Theorem 2.2. Observe that a labelling l^* and condition b) of Theorem 2.2 imply that $l^*(x)\in\{i|x\in C_i\}$ for x on the boundary of S^n. Now, for interior points x we choose $l^*(x)$ also such that $l^*(x)\in\{i|x\in C_i\}$. Let $\sigma(v^1,...,v^{n+1})$ be a completely labelled simplex for the labelling l^* and let the vertices be indexed such that v^i has label i, $i=1,...,n+1$. By definition, $v^i\in C_i$, $i=1,...,n+1$, and hence any point x in σ lies close to C_i, $i=1,...,n+1$. Let G^k, $k=1,2,...$, be a

sequence of triangulations with mesh size tending to zero if k goes to infinity, let σ^k be a completely labelled simplex in G^k, and let x^k be any point in σ^k. Then it follows from the compactness of S^n and the closedness of the sets C_i that there is a subsequence $x^{k(j)}$, j=1,2,..., with limit point x in the intersection of all sets C_i.

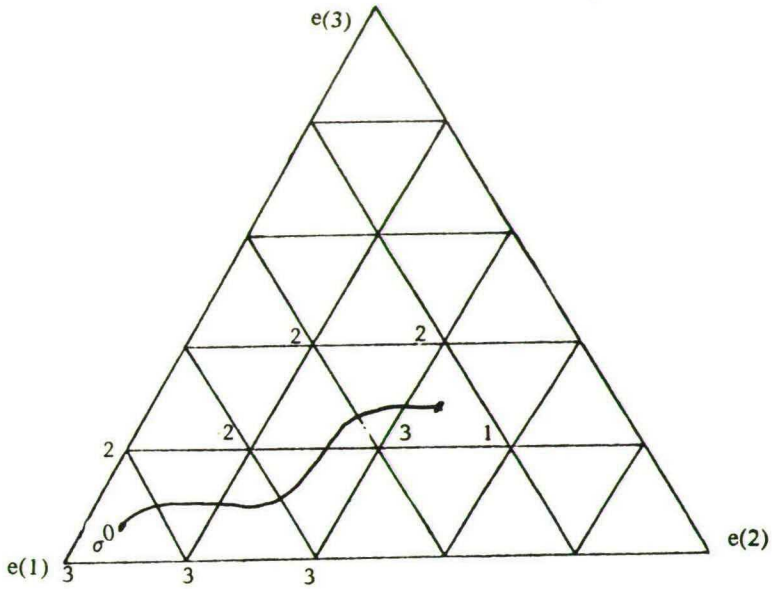


Figure 1. Scarf's algorithm, n=2

The reasoning above gives two results. Firstly, it shows the existence of an intersection point as stated in Theorem 2.2. Secondly, we have that for a sufficiently fine subdivision, each point x^* in a completely labelled simplex for a labelling l^* satisfying $l^*(x) \in \{i | x \in C_i\}$ is an approximate intersection point, in the sense that x^* lies close to each set C_i, where the distance depends on the mesh size of the subdivision and goes to zero if the mesh size tends to zero. So, with the sets C_i as defined in (2.1), an approximate zero point can be found for any excess demand function z by applying a simplicial algorithm in an appropriately labelled triangulation.

Kuhn [19,20] introduced two similar algorithms for Sperner-proper labellings. For an appropriately chosen proper labelling function, again a completely labelled simplex yields an approximate zero point of z. However, all these originating or "first-generation" simplicial algorithms have a major drawback. They suffer from inefficiency. They all have to start outside the region of interest, namely in a vertex or on the boundary of S^n. If an approximate zero point has been found, whose accuracy is not satisfactory, then we have to start again with a finer grid in a vertex or on the boundary of S^n, and all

information about the location of the solution obtained from the previous calculations becomes worthless. So, as could be expected, soon after these algorithms "second-generation" algorithms were introduced. These algorithms, of **Eaves** [9] and of **Kuhn** and **MacKinnon** [21] permit to utilize information obtained at an approximate zero for finding an approximate zero in a finer simplicial subdivision.

The Sandwich method of Kuhn and MacKinnon, which is essentially identical to a method of **Merrill** [39] to solve zero point problems on R^n, permits to start at an arbitrary point of the simplex by embedding S^n in the set $S^n x[0,1]$. The latter set is subdivided into (n+1)-dimensional simplices, such that for each vertex (w,α) of the triangulation, $\alpha \in \{0,1\}$. Now, let v be an initial guess of the solution point, for instance obtained from a previous application of the algorithm with a larger mesh size. To find a new approximation, all vertices in $S^n x\{1\}$ are labelled by l^* as above. However, the labelling on $S^n x\{0\}$ is completely determined by v. Assume that (v,0) lies in the interior of an n-simplex, say τ, in $S^n x\{0\}$, being a facet of only one (n+1)-simplex, say σ, of the subdivision. If not, v is slightly perturbed. Now, on $S^n x\{0\}$ we take a proper labelling l^* such that τ is the only simplex with label set equal to I_{n+1}. For instance, $l^*(x)=i-1$ with i the least index for which $x_{i-1}<v_{i-1}$ and $x_i \geq v_i$ (i-1=n+1 if i=1). Since v is in the interior of τ and hence in the interior of S^n, this labelling is dual proper on the boundary of $S^n x\{0\}$. Since τ is in $S^n x\{0\}$, the vertex of σ opposite τ lies in $S^n x\{1\}$. Suppose this vertex has label j. Then the facet of σ opposite the vertex of τ with label j is also completely labelled. In other words, the (n+1)-simplex σ has just two completely labelled facets, namely τ and the facet opposite the vertex of τ with label j. Now, again by the door-in door-out principle, the Sandwich method generates a sequence of different adjacent (n+1)-simplices having completely labelled common facets. Since the number of simplices is finite, within a finite number of steps we must find an (n+1)-simplex having a completely labelled facet on the boundary of $S^n x[0,1]$. Since each simplex is visited at most once, this simplex can not be σ and hence the facet can not be τ. Moreover, there are no other completely labelled n-simplices in $S^n x\{0\}$, while the dual properness of the labelling guarantees that there is no completely labelled facet in $bd(S^n)x[0,1]$. So, the algorithm must terminate with an (n+1)-simplex having a completely labelled facet in $S^n x\{1\}$. Since the labelling of the vertices in $S^n x\{1\}$ have been derived from the original problem this n-simplex yields an approximate solution point. The algorithm is illustrated in Figure 2 for n=1.

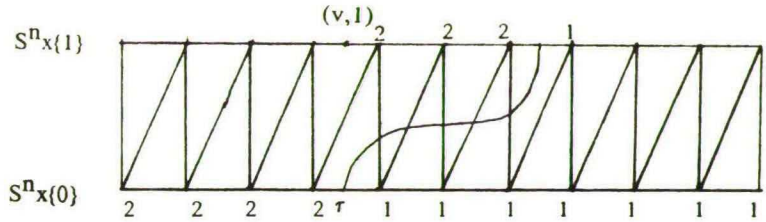


Figure 2. Sandwich method, n=1

While the Sandwich method allows to start at an arbitrary point of the simplex, Eaves' method is based on a continuous refinement of the grid. More precisely, S^n is embedded in $S^n x[1,\infty)$ and this set is subdivided into (n+1)-simplices, such that for each vertex (w,α), $\alpha=2^k$, for some k, k=0,1,2,.... Moreover, for all k=0,1,2,..., the subdivision induces a subdivision of $S^n x[2^k,2^{k+1}]$ into a finite number of (n+1)-simplices and a subdivision G^k of $S^n x\{2^k\}$ in n-simplices, such that each G^{k+1} is a refinement of G^k. Finally, G^0 consists of only one simplex, namely $S^n x\{1\}$. Each vertex of the triangulation is labelled according to l^*. This labelling assures that $S^n x\{1\}$ is the only completely labelled facet in the boundary of $S^n x[1,\infty)$. Starting with the unique (n+1)-simplex having $S^n x\{1\}$ as a facet, a sequence of adjacent simplices having completely labelled common facets is generated. Since, for any k, the number of simplices in $S^n x[1,2^k]$ is finite, the algorithm must find within a finite number of steps (a simplex in $S^n x[2^{k-1},2^k]$ having) a completely labelled facet in $S^n x\{2^k\}$ (in common with a subsequent simplex in $S^n x[2^k,2^{k+1}]$). For k large enough, such a facet yields an approximate solution point with acceptable accuracy. It should be remarked that although each level k will be reached within a finite number of steps, the algorithm can return to previous levels.

4. A variable dimension algorithm.

The second-generation algorithms of Kuhn and MacKinnon and of Eaves gave a substantial improvement of the efficiency of simplicial algorithms. In both methods a path of (n+1)-dimensional simplices having completely labelled common facets is generated. To do this the n-dimensional unit simplex S^n is embedded in an (n+1)-dimensional space. The algorithm of Eaves is rather complicated to work with, because

of the construction of a simplicial decomposition of the product space with decreasing mesh size. On the other hand, the Sandwich method is rather easy to implement. This is one reason to be in favor of the Sandwich method. Another reason is its greater flexibility. In Eaves' method the mesh size is halved between each two levels. In many cases, it may be much more efficient to decrease the mesh size with a larger factor. Some work in this direction has been done, see e.g. **Saigal** [43]. Saigal showed that under some conditions the algorithm may jump from a certain level to a higher level without bypassing the levels between. Of course, in this case the algorithm can not return to the lower level. Later on there also appeared subdivisions allowing for a larger grid refinement between two levels, see **van der Laan** and **Talman** [30] and **Shamir** [49].

The Sandwich method allows for any factor of refinement at a restart. Together with its simplicity this resulted in a lot of attention for the Sandwich method. However, there is one drawback. In each run about half of the generated vertices lies in the artificially labelled level $S^n x\{0\}$, implying that about half of the effort is spent without obtaining any new information. In the integer labelling version of the algorithm considered so far this does not matter too much. However, in general it is much more efficient to use vector labelling. Under vector labelling each vertex receives an (n+1)-vector as label instead of an integer. Then the algorithm operates by making alternately a replacement step in the simplicial subdivision and a linear programming pivot step with the label of the new vertex in a corresponding system $A\mu=b$ of n+1 linear equations with the labels of the current vertices being the columns of the matrix A. The linear programming pivot step eliminates one of the columns and determines in this way the next vertex to be removed. An approximate solution is found as soon as all columns of A correspond to vertices on the real level. Before reaching such a 'solution system' about half of the linear programming steps will be made with labels corresponding to artificially labelled vertices. To minimize the work on the artificially labelled level **Todd** [56] utilized the linearity of the artificial function. This linearity allows for combining n-simplices on the artificial level into polyhedra. This reduces the number of vertices and therefore the amount of work to be done with artificial labels.

At the end of the seventies **van der Laan and Talman** [27] introduced a new restart algorithm. This algorithm can be described by embedding S^n into $S^n x[0,1]$ with a subdivision having S^n itself as the only n-simplex on the zero-level, see e.g. **Todd** [55] and **van der Laan** [23]. So, in this case the number of vertices on the zero-level achieves its absolute minimum. However, it is much more attractive to describe the algorithm as generating in a subdivision of S^n a sequence of adjacent simplices of variable dimension. The sequence starts at an arbitrary grid point, and can therefore start anywhere, and

terminates at a full-dimensional simplex containing an approximate solution point. The attractiveness of this method lies in the fact that movements with simplices of varying dimension in S^n are typically faster than movements with (n+1)-dimensional simplices in $S^n x[0,1]$.

We now consider the variable dimension algorithm in more detail. The original algorithm utilizes the well-known Q-triangulation of S^n (see e.g. [23] or [54]). Given some mesh size m^{-1}, this triangulation subdivides S^n into simplices $\sigma(v^1,...,v^{n+1})$ with $v^1=(k_1/m,...,k_{n+1}/m)^\top$ for some nonnegative integers $k_1,...,k_{n+1}$ summing up to m, and with $v^{t+1}=v^t+(e(\pi_t)-e(\pi_t-1))/m$, t=1,...,n, where $(\pi_1,...,\pi_n)$ is a permutation of the elements of the set of integers I_n. Given m, this triangulation is fixed and the starting point v must be an element of the vertex set $\{w\in S^n | w=(k_1/m,...,k_{n+1}/m)^\top$ for nonnegative integers $k_1,...,k_{n+1}$ summing up to m$\}$. However, in a later version of the algorithm the starting point v can be any arbitrarily chosen point of S^n and the algorithm operates in the V-triangulation induced by the starting point v. This version of the algorithm is not only more efficient but also more attractive from a didactical viewpoint.

To describe the algorithm, let v be an arbitrarily chosen point in, for simplicity, the interior of S^n. Then, for any proper subset T of I_{n+1}, the t-dimensional subset A(T) of S^n, where t=|T|, is defined as the convex hull of the point v and the vertices e(i), $i\in T$, of S^n, i.e.,

$$A(T) = \{x \in S^n \mid x = v + \Sigma_i \lambda_i(e(i) - v), \text{ with } \lambda_i \geq 0,\ i \in T\}.$$

For n=2 the sets A(T) are illustrated in Figure 3. Observe that the collection of sets A(T), T a proper subset of I_{n+1}, induces a simplicial subdivision of S^n, such that for each pair S,T with $S\subset T$, A(S) is a face of A(T), while the intersection of two sets A(T) and A(S) is the common face $A(S\cap T)$. In particular, for i=1,...,n+1 the set A({i}), in the sequel to be denoted by A(i), is a one-dimensional face being the line segment connecting v and the vertex e(i) of S^n, i.e., A(i) is a ray pointing from v to e(i). Further, $A(\emptyset) = \{v\}$, i.e., $A(\emptyset)$ is the zero-dimensional simplex $\sigma(v)$. Finally, observe that for $T\neq\emptyset$, the face Conv{e(i)|$i\in T$} of S^n, where Conv means convex hull, and the sets $A(T\setminus\{h\})$, $h\in T$, are the t+1 facets of A(T). The V-triangulation of S^n, illustrated in Figure 4 for n=2, subdivides each t-dimensional set A(T) into t-dimensional simplices. For a formal description of this triangulation we refer to **Doup** and **Talman** [5] and **Doup** [4].

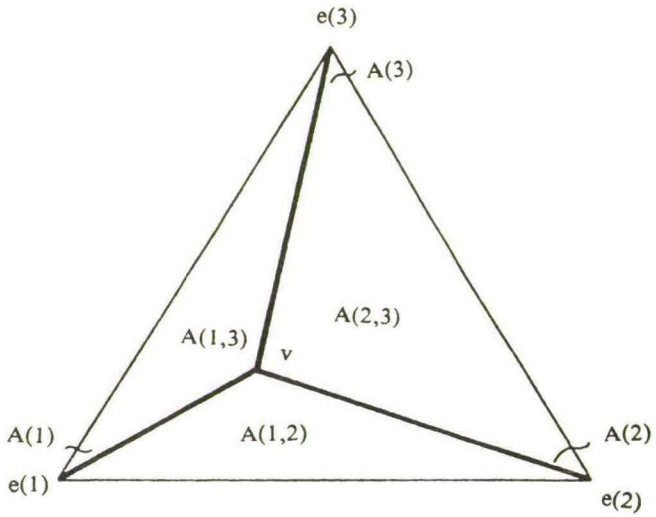


Figure 3. The sets A(T), n=2

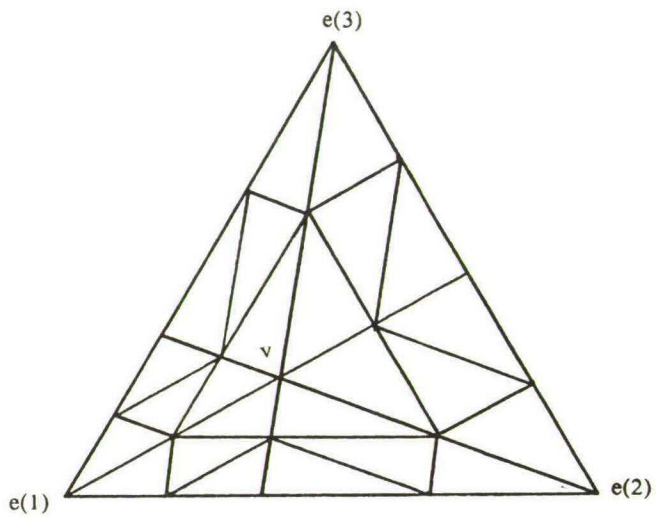


Figure 4. The V-triangulation of S^n, n=2

It is now very easy to expose the variable dimension algorithm. In fact, it is described by the next device:

> *starting with the zero-dimensional simplex $\sigma(v)$, generate for various T and corresponding $t=|T|$, a sequence of adjacent t-simplices in $A(T)$, such that the common $(t-1)$-facet of two adjacent simplices in $A(T)$ has labelset T, i.e., the vertices of such a facet jointly bear all the labels in T.*

This device is extremely simple and operates as follows. First the starting point v is evaluated and suppose that this vertex has label i. For the moment we assume that we have a dual proper labelling l^*. The zero-dimensional simplex $\sigma(v)$ is a facet of a unique one-simplex $\sigma'(v,w)$ in A(i). Starting with this simplex in A(i), the algorithm continues by generating one-dimensional simplices in A(i) having common vertices labelled i, until a new label is found, i.e., a simplex is generated for which the new vertex has some label $j\neq i$. Because of the proper labelling we know that $l^*(e(i))\neq i$ and therefore such a simplex, say $\sigma(w^1,w^2)$, must exist. This simplex is a facet of a unique 2-dimensional simplex in $A(\{i,j\})$. Then the algorithm continues by generating adjacent two-dimensional simplices in $A(\{i,j\})$ having common facets with vertices bearing the labels i and j. Until now, only an increasement of the dimension has been possible. However, we now come to the general case. For some set T, let the algorithm generate a sequence of different adjacent t-simplices in A(T) having the t labels in the set T on the vertices of the common facets. Such a facet is called T-complete. If a new label j is found and $T\cup\{j\} \neq I_{n+1}$, then the current t-simplex in A(T) is a $(T\cup\{j\})$-complete facet of a unique (t+1)-simplex $\sigma(w^1,\ldots,w^{t+2})$ in $A(T\cup\{j\})$ and, starting with this simplex σ, the algorithm continues by generating adjacent (t+1)-simplices in $A(T\cup\{j\})$ having $(T\cup\{j\})$-complete common facets. If $T\cup\{j\} = I_{n+1}$ the current simplex is completely labelled and the algorithm terminates. On the other hand, operating in A(T), a simplex σ can be generated having a T-complete facet, say $\tau(w^1,\ldots,w^t)$, in the boundary of A(T). Because of the proper labelling τ can not lie in the face $\mathrm{Conv}\{e(i)|i\in T\}$ of S^n. Hence, for some $h\in T$, the facet lies in $A(T\setminus\{h\})$. Guided by the device, now the vertex of σ not in τ is eliminated, label h is deleted from the current label set T, and the unique vertex of τ having label h is replaced, i.e., starting with τ the algorithm continues by generating adjacent (t-1)-simplices in $A(T\setminus\{h\}$ having $(T\setminus\{h\})$-complete common facets.

Again all steps are uniquely determined. Since, for each T, $T\subset I_{n+1}$, the total number of t-simplices is finite, the door-in door-out principle again guarantees that the sequence of generated simplices ends with a completely labelled simplex within a finite number of steps. If the labelling has been induced by an excess demand function any

point in the completely labelled simplex yieds an approximate zero point. If the accuracy of an approximate solution x^* is not satisfactory, a restart can be made with v equal to x^* and a finer subdivision of the new regions A(T). Successive restarts with simplicial subdivisions having decreasing mesh sizes yield increasingly more accurate approximate solutions.

For n=2 the algorithm is illustrated in Figure 5. Before finding a completely labelled simplex, the algorithm operates in A(1), A(1,2), A(2) and A(2,3) succesively.

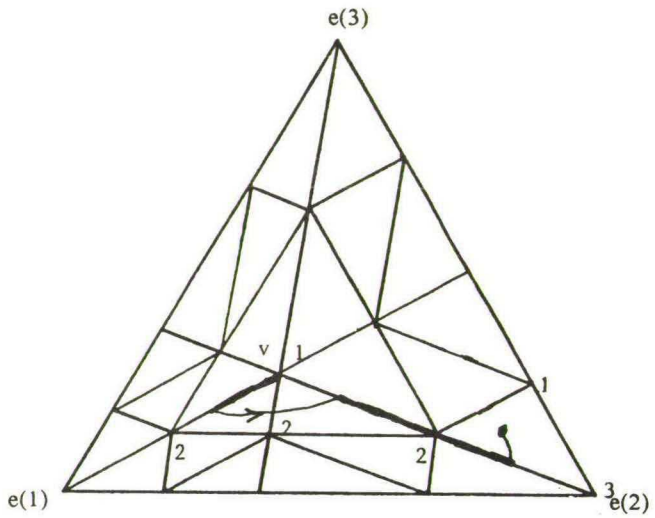


Figure 5. The variable dimension algorithm, n=2

We conclude this section with some remarks. First, recall that when the algorithm operates in A(T), a sequence of adjacent t-simplices in A(T) is generated having common facets with labelset T. Of course, except for the sequences generated by the algorithm also other such sequences may exist. By the door-in door-out principle each sequence is either a loop or a chain having two terminal simplices. As described above, for given T a terminal simplex is either a t-simplex having, for some $h \in T$, a facet with labelset T in a facet $A(T\setminus\{h\})$ of A(T), or is a t-simplex with labelset $T\cup\{k\}$ for some $k \notin T$. Of course a simplex could present both features, in which case the chain reduces to this single simplex. In the former case the boundary facet with labelset T is a terminal simplex of a chain of (t-1)-simplices in $A(T\setminus\{h\})$, in the latter case the t-simplex is either a facet of a terminal (t+1)-simplex of a chain in $A(T\cup\{k\})$ or, if $T\cup\{j\}=I_{n+1}$, the simplex is completely labelled. So, except for the terminal simplex $\sigma(v)$ in $A(\emptyset)$, each

terminal simplex of a chain in A(T) for some fixed T is either completely labelled or uniquely determines a terminal simplex of another chain. Chains can thus be linked yielding loops or paths with two terminal simplices. Except for the terminal simplex $\sigma(v)$, each terminal simplex of a path is a completely labelled n-simplex. Thus there is a unique path which connects $\sigma(v)$ with a completely labelled simplex. The algorithm follows this path from v to the other terminal simplex which will be completely labelled. All other paths connect two completely labelled simplices. On the other hand for each completely labelled simplex σ there is a unique j, such that σ lies in $A(I_{n+1}\backslash\{j\})$ and is therefore a terminal simplex of a chain in $A(I_{n+1}\backslash\{j\})$. Hence each completely labelled simplex is a terminal simplex of a path. This shows the well-known result that the number of completely labelled simplices is odd.

The second remark concerns the labelling. In the description of the algorithm we used the properness properties of the labelling l^* to show that a terminal simplex of a chain can not have a facet τ with labelset T on the face $\mathrm{Conv}\{e(i)|i\in T\}$ of S^n. If we allow a general labelling, i.e., a labelling l which does not have to satisfy any properness condition on the boundary, a vertex on the boundary of S^n can posses any label in I_{n+1}. Of course, in this case a chain in some region A(T) can have a terminal simplex having a facet τ with labelset T on the face $\mathrm{Conv}\{e(i)|i\in T\}$ of S^n. Now, consider such a facet τ with vertices, say $w^1,\ldots,w^t$. Then $\{l(w^1),\ldots,l(w^t)\}=T$, while on the other hand for all x in τ, $x_i=0$ for $i\notin T$. Hence we have that

$$\{l(w^1),\ldots,l(w^t)\} \cup \{i|x_i=0 \text{ for all } x \text{ in } \tau\} = I_{n+1}. \qquad (4.1)$$

Such a (t-1) simplex, $t\leq n$, is called complete. Observe that the sets on the left hand side of (4.1) partition I_{n+1}. For a completely labelled simplex we have that (4.1) holds with t=n+1 and such a simplex is therefore also said to be complete. Linking all chains we obtain by the same reasoning as above that there is a unique path having $\sigma(v)$ as one of its terminal simplices and a complete simplex as its other terminal simplex, whereas all other paths have two complete simplices as terminal simplices. So, in case of a general labelling there is an odd number of complete simplices (including the completely labelled simplices). See also **Freund** [12,13] and **van der Laan, Talman** and **Van der Heyden** [35].

If the general labelling l is induced by a generalized excess demand function z (i.e., z is not required to satisfy $z_i(p)\geq 0$ if $p_i=0$), for example

$$l(p) \in \{i \mid z_i(p) = \max_h z_h(p)\},$$

then it is not difficult to see that a complete simplex τ yields an approximate solution point to the NLCP with respect to z. Clearly, for all p in τ we have that $p_i=0$ for $i\notin T$, while the labelling assures that for all $i\in T$ the corresponding excess demands are close to each other and hence close to zero. This result also proves Corollary 2.6.

5. Vector labelling algorithms.

The variable dimension algorithm described in the previous section generates a sequence of adjacent simplices of varying dimension starting with a zero-dimensional simplex and ending with a complete simplex (in case of an arbitrary labelling). In case the labelling has been induced by a generalized excess demand function z the terminating simplex yields an approximate solution to the NLCP with respect to z. Now, for some T, let p be any point in a T-complete simplex σ. Then T is a subset of the labelset of σ, i.e., for each i in T there is a vertex of σ carrying label i. So, any point p in σ is close to $C_i=\{x\in S^n | z_i(x)=\max_h z_h(x)\}$ for all i in T. Hence, for all i in T, there is a positive ϵ such that

$$\max_h z_h(p) - \epsilon < z_i(p) < \max_h z_h(p) + \epsilon, \qquad (5.1)$$

whereas for k not in T

$$z_k(p) < \max_h z_h(p) + \epsilon, \qquad (5.2)$$

where ϵ depends on the mesh size of the underlying triangulation. If the mesh size tends to zero, then ϵ can be taken arbitrarily small so that approximately (5.1) and (5.2) reduce to $z_i(p)=\max_h z_h(p)$ and $z_k(p)<\max_h z_h(p)$ respectively. In case of vector labelling a path of points is followed by the algorithm along which for various T these properties exactly hold for a piecewise linear approximation of z.

The piecewise linear approximation of z with respect to an underlying triangulation of S^n is the function Z obtained by the linearization of z on each simplex of the triangulation, given the function values on the vertices of the simplex. So, for a point x in a (t-1)-simplex $\sigma(y^1,\ldots,y^t)$,

$$Z(x) = \Sigma_i \lambda_i z(y^i),$$

with the nonnegative weights λ_i, i=1,...,t, summing up to 1 given by the unique convex combination of the vertices yielding x, i.e.,

$$x = \Sigma_i \lambda_i y^i.$$

The vector labelling algorithm then follows starting in p equal to v a path of points p satisfying for some set T, $T \subset I_{n+1}$, both $p \in A(T)$ and

$$Z_i(p) = \beta \qquad \text{for } i \in T$$

and

$$Z_k(p) = \beta - \mu_k \qquad \text{for } k \notin T$$

for some β with $\beta = \max_h Z_h(p)$ and positive numbers μ_k, $k \notin T$. Let $\sigma(y^1,\ldots,y^{t+1})$ be a t-simplex in A(T) containing p. Then there are unique nonnegative numbers λ_i, i=1,...,t+1, summing up to one, such that $p = \Sigma_i \lambda_i y^i$ and $Z(p) = \Sigma_i \lambda_i z(y^i)$. Consequently, the number β and the nonnegative numbers λ_i, i=1,...,t+1, and μ_k, $k \notin T$ form a solution, to be denoted by (λ,μ,β), to the system of n+2 linear equations

$$\sum_{i=1}^{t+1} \lambda_i \begin{pmatrix} z(y^i) \\ 1 \end{pmatrix} + \sum_{k \notin T} \mu_k \begin{pmatrix} e(k) \\ 0 \end{pmatrix} - \beta \begin{pmatrix} e \\ 0 \end{pmatrix} = \begin{pmatrix} \underline{0} \\ 1 \end{pmatrix} \qquad (5.3)$$

where e(k) is the k-th unit vector in R^{n+1}, e is an (n+1)-vector of ones, and $\underline{0}$ is an (n+1)-vector of zeroes.

Definition 5.1. Given a continuous function $z: S^n \to R^{n+1}$, a t-simplex $\sigma(y^1,\ldots,y^{t+1})$ for $t=|T| \leq n$ is T-complete if the system (5.3) has a solution (λ,μ,β), with $\lambda_i \geq 0$, i=1,..,t+1, and $\mu_k \geq 0$, $k \notin T$.

We assume that at a feasible solution at most one variable of (λ,μ) is equal to zero (nondegeneracy assumption). If, for some i=1,...,t, $\lambda_i = 0$, then the facet $\tau(y^{-i})$ of σ is also called T-complete. The nondegeneracy assumption implies that in this case all variables λ_h, $h \neq i$, and μ_k, $k \notin T$, are positive. If, on the other hand, $\mu k = 0$ for some $k \notin T$, then σ is also $T \cup \{k\}$-complete. In this case all variables λi, i=1,...,t+1, and μh, $h \notin T \setminus \{k\}$, are positive. A solution with one of the variables equal to zero is called a basic solution. The nondegeneracy assumption guarantees that the system (5.3) has a line segment of solutions (λ,μ,β). This line segment connects two basic solutions and can be followed by making a linear programming pivot step in (5.3) with one of the variables that are zero at a basic solution.

Definition 5.2. A T-complete simplex $\sigma(y^1,...,y^{h+1})$, h=|T| or |T|-1, is complete if it has a feasible solution (λ,μ,β) such that for all $k\notin T$,

$$\mu_k = 0$$

or

$$x_k = 0 \text{ for all } x \text{ in } \sigma.$$

Theorem 5.3. For some T, let σ be a complete T-complete simplex in A(T). Then, either h=|T|=n and $\mu_k=0$ for k the unique element of I_{n+1} not in T, or h=|T|-1 and $x_k=0$ for all k not in T.

Proof. First, suppose h=|T|. Because of the definition of A(T), a |T|-dimensional simplex in A(T) can not lie in the boundary of S^n and hence $x_k=0$ can not hold for all x in σ. Thus σ is complete if and only if $\mu_k=0$ for all $k\notin T$. The nondegeneracy assumption implies that only one variable can be equal to zero and hence |T|=n.
Second, suppose that h=|T|-1. Then h<n since |T|≤n. The nondegeneracy sssumption implies that not all μ_k, $k\notin T$, can be zero and hence $x_k=0$ for all x in σ.

We now show that a complete simplex yields an approximate solution in case z is a generalized excess demand function. First, for some T with |T|=n, let $\sigma(y^1,...,y^{n+1})$ be a complete T-complete n-simplex in A(T). Then the corresponding system (5.3) has a solution $\lambda^*_i>0$, i=1,...,n+1, $\mu^*_k=0$, k the unique element of I_{n+1} not in T, and β^*. With $x^* = \Sigma_i \lambda^*_i y^i$ it follows from (5.3) that

$$Z_k(x^*) = \Sigma_i \lambda^*_i z_k(y^i) = \beta^* \quad k=1,...,n+1.$$

So, all components of the piecewise linear approximation Z to z are equal to each other at x^* and hence the components of z at x^* are close to each other because of the continuity of z. The accuracy again depends on the mesh size. The (generalized) Walras' condition assures that all components are close to zero and hence that x^* is an approximate solution point. Secondly, for some T, let $\sigma(y^1,...,y^{h+1})$ be a complete T-complete h-simplex in A(T) with h=|T|-1 and $x_k=0$ for all $k\notin T$ and all x in σ, and with solution $\lambda^*_i>0$, i=1,...,h+1, $\mu^*_k>0$, $k \notin T$ and β^*. With $x^* = \Sigma_i \lambda^*_i y^i$ it follows from (5.3) that

$$Z_k(x^*) = \Sigma_i \lambda^*_i z_k(y^i) = \beta^* \qquad k \in T$$
$$Z_k(x^*) = \Sigma_i \lambda^*_i z_k(y^i) = \beta^* - \mu^*_k < \beta^* \qquad k \notin T.$$

By a similar reasoning as above it follows that for all k, either $x^*_k>0$ and $z_k(x^*)$ is close to zero or $x^*_k=0$ and $z_k(x^*)$ is negative or close to negative. So, again x^* is an approximate solution point.

Analogously to the device of Section 4 the vector labelling algorithm generates for varying T, a path of uniquely determined adjacent simplices with T-complete common facets in the region A(T) of the V-triangulation of S^n, leading from the arbitrarily chosen starting point v to a complete simplex yielding an approximate solution. Let $\beta'=z_h(v)=\max_k z_k(v)$. Then $\sigma(v)$ is a zero-dimensional {h}-complete simplex with solution $\lambda_1=1$ and $\mu_k=\beta'-z_k(v)$ for $k\neq h$. The nondegeneracy assumption assures that h is the unique index for which the maximum of the components of z is obtained at v. The simplex $\sigma(v)$ is a facet of just one one-dimensional simplex $\sigma'(y^1,y^2)$ with $y^1=v$ in A(h). Analogously to the case of integer labelling the vector labelling algorithm initially generates a sequence of adjacent {h}-complete 1-simplices in A(h) with common {h}-complete facets until a simplex σ is generated having a {h}-complete simplex in common with the previously generated simplex, whereas at the other basic solution one of the variables μ_j, $j\neq h$, say μ_k, equals zero. Then σ is also {h,k}-complete and is a facet of a uniquely determined 2-simplex in A(h,k). In general, generating for some T a sequence of adjacent T-complete t-simplices in A(T), a piecewise linear path of points is traced corresponding to the solutions of (5.3). This path is traced by alternating replacement steps in the triangulation and going from basic solution to basic solution through linear programming steps in the system (5.3). Let $\sigma(y^1,\ldots,y^{t+1})$ be a simplex generated by the algorithm and suppose that λ_i equals zero for some $i=1,\ldots,t+1$ at the basic solution not in common with the previously generated simplex. Then $\tau(y^{-i})$ is T-complete and lies either in the boundary of A(T) or not. In the latter case y^i is replaced by the vertex y opposite $\tau(y^{-i})$ of the unique simplex in A(T) having $\tau(y^{-i})$ in common with σ and a linear programming step is made in (5.3) with the vector $(z(y)^\top,1)^\top$. Doing so, we get a new basic solution. If τ lies in the boundary of A(T) then either τ lies in the boundary of S^n or in $A(T\backslash\{k\})$ for some $k\in T$. The definition of A(T) implies that in the first case $x_k=0$ for all $k\notin T$ and for all x in τ, and hence τ is complete and yields an approximate solution. In the latter case τ is an $(T\backslash\{k\})$-complete simplex in $A(T\backslash\{k\})$ having $\mu_k=0$ at one of its basic solutions and the algorithm continues in $A(T\backslash\{k\})$ by making a linear programming step with $(e^\top(k),0)^\top$ in (5.3). Finally, suppose that a simplex σ is generated with $\mu_h=0$ for some $h\in T$ at the basic solution not in common with the previously

generated simplex. Then either h is the unique element of T and hence σ is complete, or σ is also (T∪{h})-complete. In the latter case σ is a facet of a unique (t+1)-simplex σ' in A(T∪{h}) and the algorithm continues in A(T∪{h}) by making a linear programming step in (5.3) with the vector corresponding to the vertex of σ' opposite σ. By the finiteness arguments the algorithm finds a complete simplex in a finite number of steps.

For a detailed description of the algorithm and computational results we refer to **Doup** and **Talman** [5], see also **Doup** [4]. The technique of vector labelling gives more possibilities than integer labelling. In this respect we notice that both in the integer labelling algorithm of Section 4 and the vector labelling algorithm described above the starting poiont v can be left along one out of n+1 rays, namely the one-dimensional sets A(j), j=1,...,n+1. Therefore these algorithms are called (n+1)-ray algorithms. In case of integer labelling the starting point v is left along A(j) iff v carries label j. Because there are n+1 variables the number of n+1 labels is natural. However, in case of vector labelling there is no need to restrict the number of rays for leaving the starting point to be equal to n+1. An algorithm with 2^{n+1}-2 rays has been described in **Doup, van der Laan** and **Talman** [7]. To motivate the $(2^{n+1}-2)$-ray algorithm we recall that a point p in A(T) for some $T \subset I_{n+1}$ on the path followed by the n+1-ray algorithm satisfies the complementarity property that for all j, $x_j \geq b(x)v_j$ if $Z_j(x)=\beta=\max_h Z_h(x)$ and $x_j=b(x)v_j$ if $Z_j(x)\leq\beta=\max_h Z_h(x)$, with $0\leq b(x)=1-\Sigma_{i\in T}\gamma_i(x)$, the nonnegative $\gamma_i(x)$'s being uniquely determined by $x=v+\Sigma_{i\in T}\gamma_i(x)(e(i)-v)$. In a similar way complementarity between the variables x_j and the values $Z_j(x)$ is utilized in the $(2^{n+1}-2)$-ray algorithm. In this algorithm the starting point v is left along a ray in S^n on which the components of v with positive z-value are proportionally increased and those with negative z-value are proportionally decreased. In the specific case that only one component of z(v) is positive, say $z_i(v)$, the ray leads from v to the vertex e(i) of S^n. In fact, there is a ray from v to each face of S^n. Since a face of S^n is the convex hull of a proper subset of the n+1 vertices of S^n, there are 2^{n+1}-2 faces in S^n and therefore there are 2^{n+1}-2 rays. The sign pattern of z(v) determines along which ray the algorithm leaves v. The algorithm moves along this ray until a point y is reached where $Z_h(y)$ is equal to zero for some h, $1\leq h\leq n+1$. Then the algorithm continues from y along a piecewise linear path of points x for which $Z_h(x)$ is kept equal to zero while the components x_k, $k\neq h$, are further proportionally increased (decreased) if $Z_k(x)$ is still positive (negative). In general the algorithm traces a piecewise linear path of points x in S^n satisfying for all j

$$\begin{aligned} x_j &= av_j && \text{if } Z_j(x) > 0 \\ bv_j \le x_j \le av_j & && \text{if } Z_j(x) = 0 \\ bv_j &= x_j && \text{if } Z_j(x) < 0, \end{aligned} \qquad (5.4)$$

with $0<b\le 1\le a$. When comparing a point x on the path with the starting point v we have that all components x_j for which the piecewise linear excess demand $Z_j(x)$ is positive (negative) are the same factor a (b) larger (smaller) than v_j, while the components x_j with zero excess demand lie between bv_j and av_j. So, for each index j either $Z_j(x)=0$ or x_j is equal to one of the two relative bounds bv_j or av_j depending on whether $Z_j(x)$ is negative or positive. The algorithm terminates as soon as a point x^* is reached for which either $Z(x^*)\le 0$ or $Z_j(x^*)\ge 0$ for all the j's with $x^*_j>0$. Because of Walras' law such a point x^* is an approximate solution to the NLCP.

The piecewise linear path of points from v which satisfies (5.4) is followed by the algorithm through alternating replacement steps in the V-triangulation of S^n and pivot steps in a linear system of equations. To describe the algorithm we first subdivide S^n into subsets A(s) for sign vectors s in R^{n+1} with components $s_j \in \{-1,0,+1\}$. For a sign vector s, let

$$\begin{aligned} I^-(s) &= \{i \in I_{n+1} |\ s_i = -1\} \\ I^0(s) &= \{i \in I_{n+1} |\ s_i = 0\} \\ I^+(s) &= \{i \in I_{n+1} |\ s_i = +1\}. \end{aligned}$$

In the sequel we assume that both $|I^+(s)|$ and $|I^-(s)|$ are at least equal to one, so that at least one component of s is equal to +1 and at least one component of s is equal to -1. Observe that there are $2^{n+1}-2$ of such sign vectors containing no zeroes at all. Each sign vector s induces a t-dimensional subset A(s) of S^n with $t=|I^0(s)|+1$. Notice that t lies between 1 and n and is equal to one for the $2^{n+1}-2$ sign vectors containing no zeroes at all. We assume that v lies in the interior of S^n.

Definition 5.4. Let s be a sign vector with $|I^+(s)|$ and $|I^-(s)|$ positive. Then

$$\begin{aligned} A(s) = \{x \in S^n |\ & x_i = av_i \text{ if } i \in I^+(s),\ x_i = bv_i \text{ if } i \in I^-(s), \text{ and} \\ & bv_i \le x_i \le av_i \text{ if } i \in I^0(s), \text{ with } 0 \le b \le 1 \le a\}. \end{aligned}$$

The boundary of a t-dimensional A(s) consists of the (t-1)-dimensional sets A(s') with $s'_i=\pm 1$ for exactly one i in $I^0(s)$ and $s'_h=s_h$, $h\ne i$, and of the intersection of A(s) with the

boundary face $S^n(I^-(s))=\{x\in S^n | x_k=0,\ k\in I^-(s)\}$ of S^n. The V-triangulation of S^n triangulates each A(s).

The algorithm traces for the piecewise linear approximation Z to z with respect to the underlying V-triangulation the path of points x from v satisfying (5.4), i.e., for some sign vector s, x lies in A(s) and s=sgn Z(x). A t-simplex containing such a point is called s-complete.

Definition 5.5. A t-simplex $\sigma(y^1,\ldots,y^{t+1})$ is s-complete if the system of n+2 linear equations

$$\sum_{i=1}^{t+1} \lambda_i \begin{pmatrix} z(y^i) \\ 1 \end{pmatrix} + \sum_{h\notin I^0(s)} -\mu_h s_h \begin{pmatrix} e(h) \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \qquad (5.5)$$

has a nonnegative solution λ^*_i, i = 1,...,t+1, and μ^*_h, $h \notin I^0(s)$.

Again we assume that for each solution to the system (5.5) at most one of the variables λ_i's and μ_h's is equal to zero. Under this nondegeneracy assumption the system (5.5) has a line segment of solutions (λ^*,μ^*), if any. An end point of such a line segment is called a basic solution and has exactly one of the variables equal to zero. The line segment of solutions (λ,μ) induces a line segment of points $x=\Sigma_i \lambda_i y^i$ in σ for which according to (5.4) sgn $Z(x)=\text{sgn}(\Sigma_i \lambda_i z(y^i))=s$. The line segment of such points or solutions to (5.5) can be followed by making a linear programming step in (5.5) with one of the variables which are equal to zero at an end point. At an end point x' either $\mu^*_h=0$ and hence $Z_h(x')=0$ for some $h\notin I^0(s)$, or for some i, $\lambda^*_i=0$ and hence x' lies in the facet of σ opposite the vertex y^i. At the starting point v, let $s^0=\text{sgn } z(v)$ and let $\sigma^0(y^1,y^2)$, where $y^1=v$, be the unique one-simplex in $A(s^0)$ having v as a vertex. Then σ^0 is s^0-complete with $\lambda_2=0$ at one of its basic solutions. The algorithm starts with this solution by making a linear programming pivot step with $(z^T(y^2),1)^T$ in the corresponding system of linear equations (5.5). In genral, if at a basic solution $\mu_k=0$ for some $k\in I^0(s)$, then the corresponding point x' is an approximate zero of z if $s_k=+1$ and $|I^+(s)|=1$ or if $s_k=-1$ and $|I^-(s)|=1$. In the first case $Z(x')\le 0$ and in the latter case $Z(x')\ge 0$. Otherwise, σ is also s'-complete and a facet of just one s'-complete (t+1)-simplex σ' in A(s') with $s'_k=0$ and $s'_h=s_h$, $h\neq k$. Then the algorithm continues with making an l.p. pivot step with $(z^T(y),1)^T$ with y the new vertex of σ'. If by an l.p. pivot step with respect to σ, λ_j becomes zero for some j, $1\le j\le t+1$, then the facet τ of σ opposite to vertex y^j is also s-complete. If the (t-1)-simplex τ lies in the boundary bd A(s) of A(s) then either τ lies in the boundary face $S^n(I^-(s))$ of S^n or τ lies in a (t-1)-dimensional set A(s') with $s'_i=\pm 1$ for exactly one i in $I^0(s)$ and $s'_h=s_h$ for all $h\neq i$. In the first case, at the point x' corresponding to the

solution, $x'_j=0$ for all j with $Z_j(x')<0$ and hence x' is an approximate solution. In the other case τ is the s'-complete (t-1)-simplex σ' in A(s') and the algorithm continues by making an l.p. pivot step in (5.5) with $(-s_i e^T(i),0)^T$. Finally, if the s-complete facet τ of σ does not lie in bd A(s), then the algorithm continues by making an l.p. pivot step in (5.5) with $(z^T(y), 1)^T$, where y is the vertex of the unique t-simplex adjacent to σ sharing τ.

In this way the algorithm generates a unique sequence of adjacent simplices of varying dimension. Under the nondegeneracy assumption no simplex can be generated more than once. Since the number of simplices of the underlying triangulation is finite, the algorithm must terminate within a finite number of steps with an approximate solution point x^* for which either $Z(x^*)\leq 0$ or $Z_j(x^*)\geq 0$ for all j with $x^*_j>0$. If the accuracy of this approximate solution is not satisfactory the algorithm can be restarted in x^* with a finer triangulation. For computational results we refer to [7].

6. Path-following as price adjustment.

As noticed in the introduction the classical Walras' tatonnement process may fail to converge to a vector of equilibrium prices, even when the set of initial price systems is restricted. In fact, neither global nor local convergence can be guaranteed. So, the mechanism is not effective in the sense that from any initial price system in any given standard pure exchange economy the process always yields a path which converges to a system of equilibrium prices. In **Saari** and **Simon** [42] it is shown that an effective mechanism needs information about both the excess demand at any price and the value of the gradients of all except one of its component functions. **Saari** [41] showed that an effective iterative mechanism which depends upon information obtained solely from the excess demand function and its derivatives does not exist.

In the algorithms discussed in the Sections 3, 4 and 5 a path of points is followed leading from an (arbitrarily chosen) starting point to an approximate solution. In case the underlying function z is an excess demand function the algorithms can be seen as a price adjustment processes leading from an arbitrarily chosen initial price system to a market clearing price system. The process converges globally and universally and is governed by comparing the value of the excess demand at a price vector with the location of that price vector with respect to the initial price vector. In this section we discuss the economic justification of such a process. We mainly restrict ourselves to the process described in the last part of the previous section.

Taking z instead of the piecewise linear approximation Z, the process corresponding to the $(2^{n+1}-2)$-algorithm traces for varying sign vectors s a path of prices p in $AC(s)=A(s)\cap C(s)$, with A(s) as defined in the previous section and

$$C(s) = \{p \in S^n \mid \operatorname{sgn} z(p) = s\}.$$

So,

$$\begin{aligned} AC(s) = \{p \in S^n \mid & p_j/v_j = \min_h p_h/v_h \text{ and } z_j(p) \le 0 \text{ if } s_j=-1, \\ & \min_h p_h/v_h \le p_j/v_j \le \max_h p_h/v_h \text{ and } z_j(p) = 0 \text{ if } s_j=0, \\ & p_j/v_j = \max_h p_h/v_h \text{ and } z_j(p) \ge 0 \text{ if } s_j=+1\}. \end{aligned}$$

The process starting in v traces such a path of prices in AC(s) until the path reaches either the boundary of A(s) or the boundary of C(s). In the first case either the boundary of S^n is reached, i.e., $p_j=0$ for all j with $z_j(p)<0$ and a solution point has been found, or we get an equality in the price ratio for some i in the set of indices $\{j \mid s_j=0\}$, i.e, for a commodity with zero excess demand either p_i/v_i becomes equal to $\max_h p_h/v_h$ or p_i/v_i becomes equal to $\min_h p_h/v_h$. Then the process continues in A(s') with $s'_j=s_j$ for all j except i and s'_i equal to +1 and -1 respectively. In case the boundary of C(s) is reached, i.e., $z_i(p)$ becomes equal to zero for some i with $s_i \in \{-1,+1\}$, the process continues in A(s') with $s'_i=0$ and $s'_j=s_j$ for all other components of s. In this way the sets AC(s) can be linked together and the union $AC = \cup_s AC(s)$ over all sign vectors s contains under some regularity and nondegeneracy conditions a curve leading from the initial price system v to an equilibrium price system p^*. This is illustrated in the Figures 6 and 7. In these figures the curves along which $z_i = 0$ are drawn for i = 1, 2, 3. In Figure 6, AC contains just one curve. In Figure 7, AC is a collection of three one-dimensional manifolds, namely a curve from v to an equilibrium price system p^*, a curve connecting two equilibrium price vectors and a loop along which $z_1=0$ in A(0,-1,+1). The formal proof that AC contains a curve leading from v to an equilibrium price vector p^* is given in **van der Laan and Talman** [34].

The process described above converges for any v and for any continuously differentiable excess demand function z, i.e., the process is globally and universally. This is the most appealing feature of the adjustment mechanism. It also shows the attractiveness of the process, above both the classical tatonnement process and Smale's global Newton method. The process is also economically meaningfull. To define the path of the process we introduced primal sets A(s) and dual sets C(s). The primal sets contain information about the location of the price vectors with respect to v. The dual sets contain information about the value of the excess demands. So, the path can be traced

by an auctioneer by keeping in mind the starting price vector v and by observing the reaction of the people in the market as reflected by the excess demand.

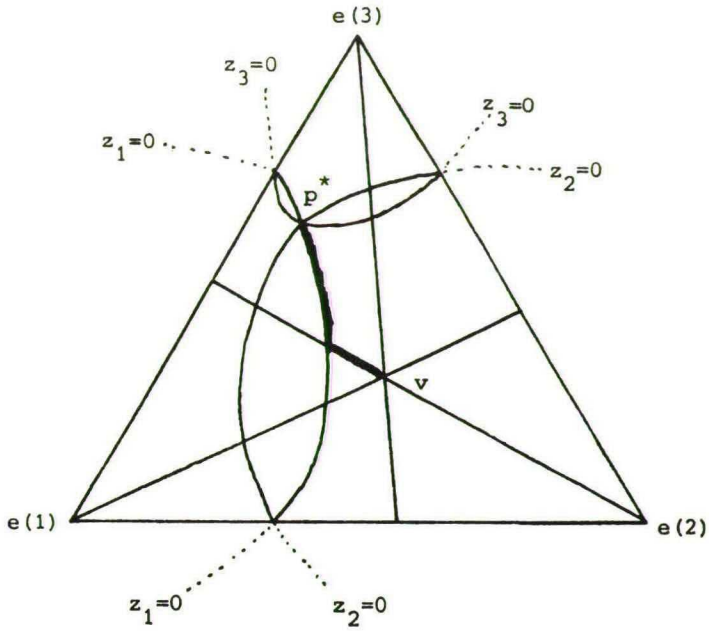


Figure 6. AC consists of the curve from v to p^*

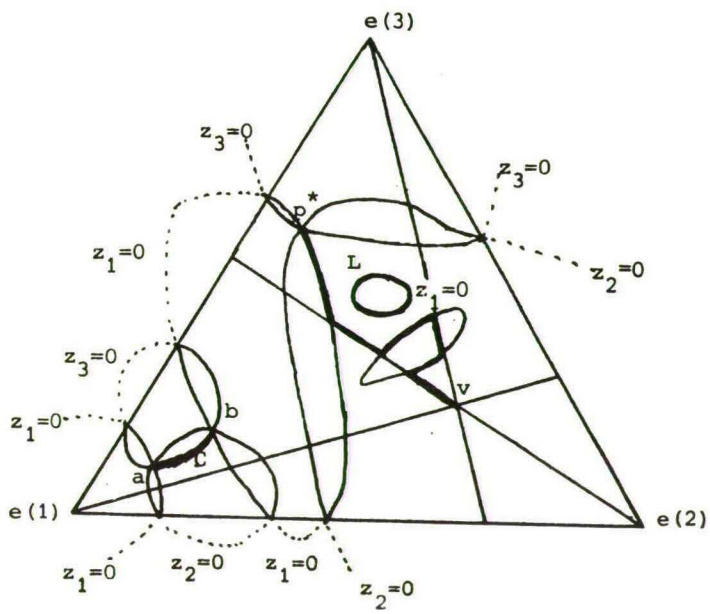


Figure 7. AC consists of the curve P from v to p^*, the curve C and the loop L

The behaviour of the auctioneer is governed by the total excess demand expressed by the individual agents. Initially, the auctioneer decreases all prices of the commodities with negative excess demand and increases the prices of all commodities with positive excess demand in such a way that the ratio between any two prices with either positive or negative excess demand is kept constant. Prices are adapted in this way until one of the markets becomes in equilibrium. Then the auctioneer adjusts the prices in order to keep the excess demand of this commodity equal to zero. In general, the auctioneer keeps with respect to their initial values at v the relative prices of the commodities with positive (negative) excess demand maximal (minimal) and allows the relative prices of the commodities with zero excess demand to vary between these two bounds. As soon as one of the markets with positive (negative) excess demand becomes in equilibrium the corresponding price is decreased (increased) away from the relative upper (lower) bound and the auctioneer adjusts this price simultaneously with the prices of the other commodities with zero excess demand in order to keep these markets in equilibrium. On the other hand, if one of the prices of the commodities with zero excess demand reaches the relative upper (lower) bound, then this market is not longer kept in equilibrium and the price is kept equal to the relative upper (lower) bound. In this way the auctioneer traces a path of prices leading to an equilibrium price system.

We want to conclude this section by considering the necessary information to follow the path of prices. Suppose that for some s, a path of prices is traced in A(s). Along this path we have that $z_j(p)=0$ for all j with $s_j=0$. So, the prices p_j with j in the set $I=I^0(s)$ solve the differential equation

$$d\, z^I(p)/dt = - \mu z^I(p),$$

with $z^I(p)$ the (t-1)-dimensional vector containing the elements of z(p) in I, under the restriction that p belongs to A(s). For the adjustment mechanism induced by this process the auctioneer needs information about $z^I(p)$ and the corresponding gradients. Moreover the auctioneer has to keep in mind the starting price vector. Following the path approximately by the steps of the simplicial algorithm described in Section 5, the corresponding piecewise linear path of prices in A(s) is traced by generating a sequence of adjacent t-dimensional simplices. In each step the excess demand at a vertex is needed. A new vertex is completely determined by the vertices of the current simplex, their corresponding excess demands, and by the initial point v. This confirms the observation of **Saari** [41] that a convergent adjustment procedure should depend on the values of the prices. This paper shows that not only the values of the current prices are

needed, but that also the value of the initial price system might play a very important role.

Of course also the other processes can be described in terms of price adjusment mechanisms. For the (n+1)-ray algorithm the process follows a path in $\cup_T$ AC(T) for various T with A(T) as defined in Section 4 and $C(T)=\{p\in S^n | z_i(p)=\max z_h(p)$ for all i in T$\}$.

7. Algorithms on the simplotope.

Thusfar we have considered algorithms on the unit (price) simplex. However, simplicial algorithms have also been utilized on other sets. For algorithms solving the problem of finding a zero point of a continuous function f: $R^n \rightarrow R^n$ we refer to [1], [17], [18], [26], [28], [31], [32], [40], [57], [59], and [60]. The vector labelling algorithms on S^n and R^n can also be applied to find zero or fixed points of upper semi-continuous point-to-set mappings, see e.g [23] and [54]. In this section we want to consider the generalization of the algorithms on the unit simplex to algorithms on the simplotope, i.e., the Carthesian product of several unit simplices. For more detailed descriptions of the algorithms to be discussed in this section we refer to **van der Laan** and **Talman** [33], **Talman** [53], **Freund** [13], **Doup** and **Talman** [5], **van der Laan, Talman** and **Van der Heyden** [35], **Doup, van den Elzen** and **Talman** [6], and **Doup** [4].

Let the simplotope $S = S^{n_1} \times \ldots \times S^{n_N}$ denote the Carthesian product of N unit simplices S^{n_j}, j=1,...,N. An element $x\in S$ will be denoted $(x_1,\ldots x_n)$ with $x_j\in S^{n_j}$, j=1,...,N. The k-th component of x_j will be denoted by x_{jk}. The set of indices (j,1),..., (j,n_j+1) is denoted by I(j), and I denotes $\cup_j I(j)$. Let T_0 a subset of I, such that $|T_0\cap I(j)|=1$ for all j, say $T_0\cap I(j)=(j,k_j)$, then $e(T_0)$ denotes the vertex of S, such that $e_j(T_0)$ is the k_j-th unit vector in R^{n_j+1}. Furthermore, for any (proper) subset T of I, such that $|T\cap I(j)|\geq 1$ for all j, the boundary face S(T) of S is defined by

$$S(T) = \{x \in S |\ x_{jk} = 0 \text{ for all } (j,k) \notin T\}.$$

Finally, let z be a continuous function from S to $R^{n_1+1} \times \ldots \times R^{n_N+1}$, satisfying $x_j^T z_j(x)=0$ for all $x\in S$ and $j\in I_N$, where $z(x)=(z_1(x),\ldots,z_N(x))$ with $z_j(x)\in R^{n_j+1}$. Given such a function the NLCP on S consists in finding a point x in S such that $z(x)\leq 0$.

A well-known example of an NLCP on S arises in game theory when computing a Nash equilibrium for a noncooperative N-person game. Given an element x in S, x_j is the mixed strategy of player j in his strategy space S^{n_j}, and $z_{jk}(x)$ is the excess profit to player j when he plays his k-th pure strategy and the others play x. A Nash equilibrium

strategy vector is a solution to the NLCP with respect to z. An other application concerns an international trade model in which each country has a group of domestic goods traded only within that country, while a group of common goods is traded among all countries. Of course, the problem of computing an equilibrium price vector can be formulated on the full-dimensional price simplex. However, by exploiting the block-diagonal structure of the demand function, the problem can be converted into an NLCP problem on the simplotope S with N-1 the number of countries, n_j the number of domestic commodities in country j, and n_N+1 the number of common goods. This approach gives a substantial improvement in the computational efficiency.

In this section we will discuss two algorithms to find a solution to the NLCP on S. The first one is a generalization of the (n+1)-ray algorithm on S^n and the second of the $(2^{n+1}-2)$-ray algorithm. The triangulation underlying both algorithms is the V-triangulation of S, introduced in [5]. This triangulation is a direct generalization of the V-triangulation of S^n. Furthermore, let Z be again the piecewise linear approximation to z with respect to the underlying triangulation.

In the same way that the (n+1)-ray algorithm has a ray to each of the n+1 vertices of the unit simplex, the generalization of this algorithm to S has a ray from the arbitrary starting point v to each vertex of the simplotope S. Because the number of vertices of S is equal to $\Pi_j\ (n_j+1)$, this algorithm is called the product-ray algorithm or the $\Pi_j\ (n_j+1)$-ray algorithm. From the (interior) point v, the algorithm makes initially a search along the ray pointing to the vertex $e(T_0)$ of S where T_0 is the set of indices (j,k_j), $j\in I_N$, such that the k_j-th component of $z_j(v)$ is equal to $\max_h z_{jh}(v)$. Going along this ray to $e(T_0)$, for each j the k_j-th component of x_j is increased, while all other components of x are proportionally decreased, until either the point $e(T_0)$ is found as an exact solution of the NLCP, or a point x is found, such that for some (j,k) in I, $k\neq k_j$, also $Z_{jk}(x)=\max_h Z_{jh}(x)$. In the latter case the algorithm continues in the convex hull of v and $S(T_0\cup\{(j,k)\})$ keeping $Z_{jk}(x)$ also maximal. In general, for varying T satisfying $|T\cup I(j)|\geq 1$ for all j, the algorithm traces a piecewise linear path of points x in the convex hull A(T) of v and S(T), such that for all j, $Z_{jk}(x)=\max_h Z_{jh}(x)$ for all (j,k) in T. If a point x^* is reached in the boundary of A(T), then either x^* lies in S(T) or x^* lies in $A(T\setminus\{(j,k')\})$ for some (j,k') in T. In the first case for all (j,k) not in T and hence for all (j,k) with $Z_{jk}(x^*)<\max_h Z_{jh}(x^*)$, $x^*_{jk}=0$ and hence x^* is an approximate solution. In the other case the algorithm continues in $A(T\setminus\{(j,k')\})$ by relaxing the condition $Z_{jk'}(x)=\max_h Z_{jh}(x)$. On the other hand, a point x^* in A(T) can be reached where for some (j,k') not in T, $Z_{jk'}(x^*)=\max_h Z_{jh}(x^*)$. Then, either the algorithm continues in $A(T\cup\{(j,k')\})$, or $T\cup\{(j,k')\}=I$ and x^* is an approximate solution satisfying $Z_{jk}(x^*)=\max_h$

$Z_{jh}(x^*)$ for all $(j,k)\in I(j)$, $j\in I_N$. The algorithm traces this piecewise linear path from the point v to an approximate solution x^* by generating for varying T a sequence of adjacent T-complete simplices in A(T) with T-complete common facets. Generalizing Definition 5.1, a t-simplex $\sigma(y^1,...,y^{t+1})$ is T-complete, $t=|T|$, if the system of $\Sigma_j (n_j+1)+1$ of linear equations

$$\sum_{i=1}^{t+1} \lambda_i \begin{pmatrix} z(y^i) \\ 1 \end{pmatrix} + \sum_{(j,k)\notin T} \mu_{jk} \begin{pmatrix} e(j,k) \\ 0 \end{pmatrix} - \sum_{j=1}^{N} \beta_j \begin{pmatrix} e'(j) \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \qquad (7.1)$$

where e(j,k) is the (j,k)-th unit vector in $\Pi_j R^{n_j+1}$, $e'(j)=\Sigma_h e(j,h)$, and 0 is the $\Sigma_j (n_j+1)$ vector of zeroes, has a solution (λ,μ,β) with $\lambda_i\geq 0$, i=1,...,t+1 and $\mu_{jk}\geq 0$, $(j,k)\notin T$. Again we assume nondegeneracy, i.e., at a basic solution (λ,μ,β) of (7.1) at most one of the variables (λ,μ) is equal to zero. If at a basic solution $\lambda_i=0$ for some i, then the facet opposite y^i is also called T-complete. As soon as a T-complete simplex is generated having a T-complete facet in S(T), or having a solution with $\mu_{jk}=0$ while $T=I\setminus\{(j,k)\}$, the algorithm terminates with the approximate solution $x^*=\Sigma_i\lambda_i y^i$. If the accuracy at the approximate solution is not sufficient, the algorithm can be restarted at x^* with a finer mesh size of the triangulation. We remark that the algorithm also allows for starting on the boundary of S. For further details and computational results we refer to **Doup** and **Talman** [5] and **Doup** [4].

The second algorithm on S we want to discuss shortly is the generalization of the $(2^{n+1}-2)$-ray algorithm on S^n. Recall that the path traced by this algorithm is governed by the sign pattern of Z. This also holds for the generalized algorithm on S. Because the total number of different sign patterns of z at an interior point v is equal to $\Pi_j (2^{n_j+1}-2)$, this algorithm on S has this number of rays to leave the starting point v and is called the $\Pi_j (2^{n_j+1}-2)$-ray or exponent-ray algorithm. From x=v, initially the algorithm decreases proportionally all components (i,h) of x with negative $z_{ih}(v)$ and increases for each j proportionally all components (j,k) of x_j with positive $z_{jk}(v)$. As soon as for some (j,h) in I, $Z_{jh}(x)$ becomes equal to zero, the algorithm adapts x_{jh}, keeping $Z_{jh}(x)$ equal to zero. In general the algorithm traces for varying sign vectors s a piecewise linear path of points x in S satisfying s=sgn Z(x) and lying in the subset A(s) of S defined by

(A) $x_{jk}/v_{jk} = \min_{(i,h)} x_{ih}/v_{ih}$ if $s_{jk}<0$ and $s_{jh}>0$ for some h

(B) $x_{jk}/v_{jk} = \min_h x_{jh}/v_{jh}$ if $s_{jk}<0$ and $s_j\leq 0$

(C) $\min_{(i,h)} x_{ih}/v_{ih} \leq x_{jk}/v_{jk} \leq \max_h x_{jh}/v_{jh}$ if $s_{jk}=0$ and $s_{jh}>0$ for some h

(D) $\min_h x_{jh}/v_{jh} \leq x_{jk}/v_{jk} \leq \max_h x_{jh}/v_{jh}$ if $s_{jk}=0$ and $s_j\leq 0$

(E) $x_{jk}/v_{jk} = \max_h x_{jh}/v_{jh}$ if $s_{jk}>0$.

Remark that the piecewise linear approximation Z to z only satisfies the Walras' condition approximately. Therefore, the case that $s_j=\text{sgn } Z_j(x) \leq 0$ (≥ 0) can not be excluded. The path is followed by generating for varying s a sequence of adjacent s-complete simplices in A(s) with s-complete common facets. Generalizing Definition 5.5, a t-simplex $\sigma(y^1,...,y^{t+1})$ is s-complete, $t=|I^0(s)|+1$, if the system of Σ_j $(n_j+1)+1$ linear equations

$$\sum_{i=1}^{t+1} \lambda_i \begin{pmatrix} z(y^i) \\ 1 \end{pmatrix} + \sum_{(j,h)\notin I^0(s)} -\mu_{jh} s_{jh} \begin{pmatrix} e(j,h) \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \qquad (7.2)$$

has a nonnegative solution λ^*_i, i = 1,...,t+1, and μ^*_{jh}, $(j,h)\notin I^0(s)$. Again we assume nondegeneracy, i.e., at a solution (λ,μ) of (7.2) at most one of the variables (λ,μ) is equal to zero. If at a solution $\lambda_i=0$ for some i, then the facet opposite y^i is also called s-complete. As soon as an s-complete t-simplex is generated in A(s) having an s-complete facet in the boundary of S or having a solution with for all j either $s_{jk}\mu_{jk}\geq 0$ for all $(j,k)\in I(j)$ or $s_{jk}\mu_{jk}\leq 0$ for all $(j,k)\in I(j)$, the algorithm terminates with the approximate solution $x^*=\Sigma_i\lambda_i y^i$. Notice that $Z_{jk}(x^*)<0$ if $x^*_{jk}=0$ and that for all j either $Z_{jk}(x^*)\geq 0$ for all $(j,k)\in I(j)$ with $x^*_{jk}>0$ or $Z_{jk}(x^*)\leq 0$ for all $(j,k)\in I(j)$ with $x^*_{jk}>0$. If the accuracy at the approximate solution is not sufficient, the algorithm can be restarted at x^* with a finer mesh size of the triangulation. We remark that also this algorithm can be started on the boundary of S. For further details and computational results we refer to **Doup, van den Elzen** and **Talman** [6] and **Doup** [4].

REFERENCES

[1] M.N. Broadie, "Computational experience with the octahedral algorithm and related results", Department of Operations Research, Stanford University, Stanford, California, 1983.

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

[3] G. Debreu, "Excess demand functions", *Journal of Mathematical Economics* 1 (1974) 15-23.

[4] T.M. Doup, "Simplicial algorithms on the simplotope", Ph.D. Dissertation (Tilburg University, Tilburg, The Netherlands, 1987).

[5] T.M. Doup and A.J.J. Talman, "A new variable dimension simplicial algorithm to find equilibria on the product space of unit simplices", *Mathematical Programming* 37 (1987) 319-355.

[6] T.M. Doup, A.H. van den Elzen and A.J.J. Talman, "Simplicial algorithms for solving the nonlinear complementarity problem on the simplotope", in: A.J.J. Talman and G. van der Laan (eds.), *The Computation and Modelling of Economic Equilibria* (North-Holland, Amsterdam, 1987, pp. 125-154).

[7] T.M. Doup, G. van der Laan and A.J.J. Talman, "The $(2^{n+1}-2)$-ray algorithm: a new simplicial algorithm to compute economic equilibria", *Mathematical Programming*, to appear.

[8] J. Drèze, "Existence of an exchange equilibrium under price rigidities" *International Economic Review* 16 (1975) 310-320.

[9] B.C. Eaves, "Homotopies for computation of fixed points", *Mathematical Programming* 3 (1972) 1-22.

[10] A.H. van den Elzen and G. van der Laan, "A price adjustment process for an economy with a block-diagonal pattern", Research Report 166, Free University, Amsterdam, The Netherlands, 1985.

[11] A.H. van den Elzen, G. van der Laan and A.J.J. Talman, "Adjustment processes for finding equilibria on the simplotope, Research Memorandum 196, Tilburg University, Tilburg, The Netherlands, 1985.

[12] R.M. Freund, "Variable dimension complexes with applications", Ph.D. Thesis (Stanford University, Stanford, California, 1980)

[13] R.M. Freund, "Combinatorial theorems on the simplotope that generalize results on the simplex and cube", *Mathematics of Operations Research* 11 (1986) 169-179.

[14] D. Gale, "The law of supply and demand", *Mathematica Scandinavica* 3 (1955) 155-169.

[15] B. Ingrao and G. Israel, "General economic equilibrium theory, a history of ineffectual paradigmatic shifts," *Fundamenta Scientiae* 6 (1985) 1-45 (part I) and 89-125 (part II).

[16] B. Knaster, C. Kuratowski and S. Mazurkiewicz "Ein Beweis des Fixpunktsatzes fur n-dimensionale Simplexe", *Fund. Math.* 14 (1929) 132-137.

[17] M. Kojima and Y. Yamamoto, "Variable dimension algorithms: Basic theory, interpretations and extensions of some existing methods", *Mathematical Programming* 24 (1982) 177-215.

[18] M. Kojima and Y. Yamamoto, "A unified approach to the implementation of several restart fixed point algorithms and a new variable dimension algorithm", *Mathematical Programming* 28 (1984) 288-328.

[19] H.W. Kuhn, "Simplicial approximation of fixed points", *Proc. Nat. Acad. Sci. U.S.A.* 61 (1968) 1238-1242.

[20] H.W. Kuhn, "Approximate search for fixed points", *Computing Methods in Optimization Problems* 2 (1969) 199-211.

[21] H.W. Kuhn and J.G. MacKinnon, "The Sandwich method for finding fixed points", *Journal of Optimization Theory and Applications* 17 (1975) 189-204.

[22] M. Kurz, "Unemployment equilibria in an economy with linked prices", *Journal of Economic Theory* 26 (1982) 100-123.
[23] G. van der Laan, *Simplicial Fixed Point Algorithms* (Mathematical Centre Tracts 129, Amsterdam, The Netherlands, 1980).
[24] G. van der Laan, "Equilibrium under rigid prices with compensation for the consumers", *International Economic Review* 21 (1980) 53-73.
[25] G. van der Laan, "Simplicial approximation of unemployment Equilibria", *Journal of Mathematical Economics* 9 (1982) 83-97.
[26] G. van der Laan and L.P. Seelen, "Efficiency and implementation of simplicial zero algorithms", *Mathematical Programming* 30 (1984) 196-217.
[27] G. van der Laan and A.J.J. Talman, "A restart algorithm for computing fixed points without an extra dimension", *Mathematical Programming* 17 (1979) 74-84.
[28] G. van der Laan and A.J.J. Talman, "A restart algorithm without an artificial level for computing fixed points on unbounded regions", in: H.-O. Peitgen and H.-O. Walther (eds.0, *Functional Differential Equations and Apporximation of Fixed Points* (Springer, Berlin, 1979, pp. 247-256)
[29] G. van der Laan and A.J.J. Talman, "An improvement of fixed point algorithms by using a good triangulation", *Mathematical Programming* 18 (1980) 274-285.
[30] G. van der Laan and A.J.J. Talman, "A new subdivision for computing fixed points with a homotopy algorithm", *Mathematical Programming* 19 (1980) 78-91.
[31] G. van der Laan and A.J.J. Talman, "Convergence and properties of recent variable dimension algorithms", in: W. Forster (ed.), *Numerical Solution of Highly Nonlinear Problems* (North-Holland, Amsterdam, 1980, pp. 3-36).
[32] G. van der Laan and A.J.J. Talman, "A class of simplicial restart fixed point algorithms without an extra dimension", *Mathematical Programming* 20 (1981) 33-48.
[33] G. van der Laan and A.J.J. Talman, "On the computation of fixed points on the product space of unit simplices and an application to noncooperative N-person games", *Mathematics of Operations Research* 7 (1982) 1-13.
[34] G. van der Laan and A.J.J. Talman, "Adjustment processes for finding economic equilibria", in: A.J.J. Talman and G. van der Laan (eds.), *The Computation and Modelling of Economic Equilibria* (North-Holland, Amsterdam, 1987, pp. 85-124).
[35] G. van der Laan, A.J.J. Talman and L. Van der Heyden, "Simplicial variable dimension algorithms for solving the nonlinear complementarity problem on a product of unit simplices using a general labelling", *Mathematics of Operations Research*, to appear.
[36] C.E. Lemke and J.T. Howson, "Equilibrium points of bi-matrix games", *SIAM Journal of Applied Mathematics* 12 (1964) 788-780.
[37] L.W. McKenzie, "On equilibrium in Graham's model of world trade and other competitive systems", *Econometrica* 22 (1954) 147-161.
[38] L.W. McKenzie, "On the existence of general equilibrium for a competitive market", *Econometrica* 27 (1959) 54-71.
[39] O.H. Merrill, "Applications and extension of an algorithm that computes fixed points of certain upper semi-continuous point-to-set mappings", Ph.D. Thesis, (University of Michigan, Ann Arbor, Michigan 1972).
[40] P.M. Reiser, "A modified integer labelling for complementarity algorithms", *Mathematics of Operations Research* 6 (1981) 129-139.
[41] D.G. Saari, "Iterative Price Mechanisms", *Econometrica* 53 (1985) 1117-1131.
[42] D.G. Saari, and C. Simon, "Effective Price Mechanisms", *Econometrica* 46 (1978) 1097-1125.
[43] R. Saigal, "On the convergence rate of algorithms for solving equations that are based on complementarity pivoting", *Mathematics of Operations Research* 2 (1977) 108-124.
[44] R. Saigal, "A homotopy for solving large, sparse and structured fixed point problems", *Mathematics of Operations Research* 8 (1983) 557-578.

[45] H. Scarf, "Some examples of global instability of the competitive equilibrium", *International Economic Review* 1 (1960) 157-172.
[46] H. Scarf, "The approximation of fixed points of a continuous mapping", *SIAM Journal of Applied Mathematics* 15 (1967) 1328-1343.
[47] H. Scarf, *The computation of economic equilibria* (Yale University Press, 1973).
[48] H.E. Scarf, "The computation of equilibrium prices: an exposition," in: K.J. Arrow and M.D. Intriligator (eds.), *Handbook of Mathematical Economics, Vol II* (North-Holland, Amsterdam, 1982, pp.1007-1061).
[49] S. Shamir, "Two new triangulations for homotopy fixed point algorithms with an arbitrary grid refinement", in: S.M. Robinson (ed.), *Analysis and Computation of Fixed Points* (Academic Press, New York, 1980, pp. 25-56).
[50] S. Smale, "A convergent process of price adjustment and global Newton methods", *Journal of Mathematical Economics* 3 (1976) 107-120.
[51] E. Sperner, "Neuer Beweis fur die Invarianz der Dimensionszahl und des Gebietes", *Abh. Math. Sem. Univ. Hamburg* 6 (1928) 265-272).
[52] H. Sonnenschein, "Market excess demand functions", *Econometrica* 40 (1972) 549-563.
[53] A.J.J. Talman, *Variable Dimension Fixed Point Algorithms and Triangulations* (Mathematical Centre Tracts 128, Mathematisch Centrum, Amsterdam, 1980).
[54] M.J. Todd, *The Computation of Fixed points and Applications* (Lecture Notes in Economics and Mathematical Systems 124, Springer, Berlin, 1976).
[55] M.J. Todd, "Fixed-point algorithms that allow restarting without an extra dimension", Technical Report 379, SORIE, Cornell University, Ithaca, New York 1978.
[56] M.J. Todd, "Traversing large pieces of linearity in algorithms that solve equations by following piecewise-linear paths", *Mathematics of Operations Research* 5 (1980) 242-257.
[57] M.J. Todd, "Global and local convergence and monotonicity results for a recent variable dimension simplicial algorithm", in: W. Forster (ed.), *Numerical Solution of Highly Nonlinear Problems* (North-Holland, Amsterdam, 1980 pp. 43-69).
[58] H. Uzawa, "Walras' existence theorem and Brouwer's fixed point theorem", *Economic Studies Quarterly* 13 (1962) 1.
[59] A.H. Wright, "The octahedral algorithm, a new simplicial fixed point algorithm", *Mathematical Programming* 21 (1981) 47-69.
[60] Y. Yamamoto, "A new variable dimension algorithm for the fixed point problem", *Mathematical Programming* 25 (1983) 329-342.

IN 1986 REEDS VERSCHENEN

202 J.H.F. Schilderinck
Interregional Structure of the European Community. Part III

203 Antoon van den Elzen and Dolf Talman
A new strategy-adjustment process for computing a Nash equilibrium in a noncooperative more-person game

204 Jan Vingerhoets
Fabrication of copper and copper semis in developing countries. A review of evidence and opportunities

205 R. Heuts, J. van Lieshout, K. Baken
An inventory model: what is the influence of the shape of the lead time demand distribution?

206 A. van Soest, P. Kooreman
A Microeconometric Analysis of Vacation Behavior

207 F. Boekema, A. Nagelkerke
Labour Relations, Networks, Job-creation and Regional Development. A view to the consequences of technological change

208 R. Alessie, A. Kapteyn
Habit Formation and Interdependent Preferences in the Almost Ideal Demand System

209 T. Wansbeek, A. Kapteyn
Estimation of the error components model with incomplete panels

210 A.L. Hempenius
The relation between dividends and profits

211 J. Kriens, J.Th. van Lieshout
A generalisation and some properties of Markowitz' portfolio selection method

212 Jack P.C. Kleijnen and Charles R. Standridge
Experimental design and regression analysis in simulation: an FMS case study

213 T.M. Doup, A.H. van den Elzen and A.J.J. Talman
Simplicial algorithms for solving the non-linear complementarity problem on the simplotope

214 A.J.W. van de Gevel
The theory of wage differentials: a correction

215 J.P.C. Kleijnen, W. van Groenendaal
Regression analysis of factorial designs with sequential replication

216 T.E. Nijman and F.C. Palm
Consistent estimation of rational expectations models

217 P.M. Kort
The firm's investment policy under a concave adjustment cost function

218 J.P.C. Kleijnen
Decision Support Systems (DSS), en de kleren van de keizer ...

219 T.M. Doup and A.J.J. Talman
A continuous deformation algorithm on the product space of unit simplices

220 T.M. Doup and A.J.J. Talman
The 2-ray algorithm for solving equilibrium problems on the unit simplex

221 Th. van de Klundert, P. Peters
Price Inertia in a Macroeconomic Model of Monopolistic Competition

222 Christian Mulder
Testing Korteweg's rational expectations model for a small open economy

223 A.C. Meijdam, J.E.J. Plasmans
Maximum Likelihood Estimation of Econometric Models with Rational Expectations of Current Endogenous Variables

224 Arie Kapteyn, Peter Kooreman, Arthur van Soest
Non-convex budget sets, institutional constraints and imposition of concavity in a flexible household labor supply model

225 R.J. de Groof
Internationale coördinatie van economische politiek in een twee-regio-twee-sectoren model

226 Arthur van Soest, Peter Kooreman
Comment on 'Microeconometric Demand Systems with Binding Non-Negativity Constraints: The Dual Approach'

227 A.J.J. Talman and Y. Yamamoto
A globally convergent simplicial algorithm for stationary point problems on polytopes

228 Jack P.C. Kleijnen, Peter C.A. Karremans, Wim K. Oortwijn, Willem J.H. van Groenendaal
Jackknifing estimated weighted least squares

229 A.H. van den Elzen and G. van der Laan
A price adjustment for an economy with a block-diagonal pattern

230 M.H.C. Paardekooper
Jacobi-type algorithms for eigenvalues on vector- and parallel computer

231 J.P.C. Kleijnen
Analyzing simulation experiments with common random numbers

232 A.B.T.M. van Schaik, R.J. Mulder
On Superimposed Recurrent Cycles

233 M.H.C. Paardekooper
Sameh's parallel eigenvalue algorithm revisited

234 Pieter H.M. Ruys and Ton J.A. Storcken
Preferences revealed by the choice of friends

235 C.J.J. Huys en E.N. Kertzman
Effectieve belastingtarieven en kapitaalkosten

236 A.M.H. Gerards
An extension of König's theorem to graphs with no odd-K_4

237 A.M.H. Gerards and A. Schrijver
Signed Graphs - Regular Matroids - Grafts

238 Rob J.M. Alessie and Arie Kapteyn
Consumption, Savings and Demography

239 A.J. van Reeken
Begrippen rondom "kwaliteit"

240 Th.E. Nijman and F.C. Palmer
Efficiency gains due to using missing data. Procedures in regression models

241 S.C.W. Eijffinger
The determinants of the currencies within the European Monetary System

IN 1987 REEDS VERSCHENEN

242 Gerard van den Berg
Nonstationarity in job search theory

243 Annie Cuyt, Brigitte Verdonk
Block-tridiagonal linear systems and branched continued fractions

244 J.C. de Vos, W. Vervaat
Local Times of Bernoulli Walk

245 Arie Kapteyn, Peter Kooreman, Rob Willemse
Some methodological issues in the implementation of subjective poverty definitions

246 J.P.C. Kleijnen, J. Kriens, M.C.H.M. Lafleur, J.H.F. Pardoel
Sampling for Quality Inspection and Correction: AOQL Performance Criteria

247 D.B.J. Schouten
Algemene theorie van de internationale conjuncturele en strukturele afhankelijkheden

248 F.C. Bussemaker, W.H. Haemers, J.J. Seidel, E. Spence
On (v,k,λ) graphs and designs with trivial automorphism group

249 Peter M. Kort
The Influence of a Stochastic Environment on the Firm's Optimal Dynamic Investment Policy

250 R.H.J.M. Gradus
Preliminary version
The reaction of the firm on governmental policy: a game-theoretical approach

251 J.G. de Gooijer, R.M.J. Heuts
Higher order moments of bilinear time series processes with symmetrically distributed errors

252 P.H. Stevers, P.A.M. Versteijne
Evaluatie van marketing-activiteiten

253 H.P.A. Mulders, A.J. van Reeken
DATAAL - een hulpmiddel voor onderhoud van gegevensverzamelingen

254 P. Kooreman, A. Kapteyn
On the identifiability of household production functions with joint products: A comment

255 B. van Riel
Was er een profit-squeeze in de Nederlandse industrie?

256 R.P. Gilles
Economies with coalitional structures and core-like equilibrium concepts

257 P.H.M. Ruys, G. van der Laan
Computation of an industrial equilibrium

258 W.H. Haemers, A.E. Brouwer
Association schemes

259 G.J.M. van den Boom
Some modifications and applications of Rubinstein's perfect equilibrium model of bargaining

260 A.W.A. Boot, A.V. Thakor, G.F. Udell
Competition, Risk Neutrality and Loan Commitments

261 A.W.A. Boot, A.V. Thakor, G.F. Udell
Collateral and Borrower Risk

262 A. Kapteyn, I. Woittiez
Preference Interdependence and Habit Formation in Family Labor Supply

263 B. Bettonvil
A formal description of discrete event dynamic systems including perturbation analysis

264 Dr. Sylvester C.W. Eijffinger
A monthly model for the monetary policy in the Netherlands

265 F. van der Ploeg, A.J. de Zeeuw
Conflict over arms accumulation in market and command economies

266 F. van der Ploeg, A.J. de Zeeuw
Perfect equilibrium in a model of competitive arms accumulation

267 Aart de Zeeuw
Inflation and reputation: comment

268 A.J. de Zeeuw, F. van der Ploeg
Difference games and policy evaluation: a conceptual framework

269 Frederick van der Ploeg
Rationing in open economy and dynamic macroeconomics: a survey