Discussion Paper No. 967

A Decentralized Linear Process for Finding Equlibria

by

Stanley Reiter
Northwestern University

November 1991

A Decentralized Linear Process for Finding Equilibria

by

Stanley Reiter *

November 1991

Section 1.

This paper presents a decentralized dynamic process designed to apply to the case of linear equilibrium equations. The computations required by the process are a few simple arithmetic calculations per step. Memory and time requirements increase roughly as the third power of the number of agents (equations). The process presented in this paper can also be viewed as a distributed algorithm for solving a distributed system of linear equations.

This paper may be view as a sequel to [3]. That paper describes a class of decentralized dynamic processes designed to converge to equilibrium when the equilibrium equations are linear. Those processes can also be viewed as distributed algorithms for solving systems of linear equations, or as learning algorithms. The class includes processes that use a message space larger by one binary digit (per agent or equation) than the space in which the equilibrium or solution resides. The computations involved in those processes are quite simple, the only computational steps being to take the average of two points of the space, and to evaluate the sign of a linear function at a point. However, memory and time requirements increase exponentially with the number of agents (equations). The main purpose of this paper is to present a decentralized process based on the same ideas as those in [3], but one whose computational

*Department of Economics, Northwestern University, Evanston, IL 60208

This research was supported by the National Science Foundation. I am indebted to Michael Chwe, Scott Page, Carl Simon and Steven Williams for helpful discussions and comments.

steps do not grow exponentially. This gain is achieved at a cost: each agent must communicate the value of a linear function at a point, instead of just its sign.

The background and motivation for investigating decentralized processes for finding equilibrium is discussed in [3], and references to the literature are given there.[1] In that literature dynamic processes are modeled by differential or difference equations that satisfy a condition expressing a requirement of decentralization. While these models differ in a variety of ways, the import of the results is that a decentralized process capable of stabilizing the equilibria of a static mechanism (one that realizes some goal function, such as giving Pareto optimal, Walrasian, or fair allocations) requires more information, (a message space larger in dimension) than the minimum needed for static realization or verification of equilibrium. While these results hold more generally, they do apply to the case of linear equilibrium conditions. Indeed, Scarf's example, which shows that it is not true that the competitive price adjustment process is stable for all classical environments, has linear excess demand functions [4]. Other results, cited in [3], show that a similar conclusion holds for a more general class of message exchange processes, namely that if the class of environments is rich enough, for any decentralized process given by differential or difference equations, there is an open set of environments for which the equilibria are not locally stable. In order to guarantee local stablility for the full class of environments, it is necessary to enlarge the message space of the dynamic process, more information must be exchanged among the agents to ensure local stability, and a fortiori, global convergence. Smale's globalized Newton's Method, [6] which applies to price dynamics, involves the use of the derivatives of the excess demand function in addition to the variables (prices and quantities) needed to characterize equilibrium.

The processes in [3] and in this paper are not given by differential equations whose trajectories lie in the space of the equilibrium, but are algorithmic and they are decentralized.[2] Moreover, the processes presented in [3] and in this paper converge globally to the equilibrium or solution on the full class of environments, which in the

[1] A recent paper not cited in [3] is [2].

[2] Other processes for finding equilibrium based on algorithms have been studied: Scarf's fixed point algorithm is a notable example [5]. However, not all of these are decentralized or distributed processes.

linear case means for all values of the parameters of the equations characterizing equilibrium. While these processes are formally not covered by the theorems about the informational requirements of local stability referred to above, they do require more information than is required to verify static equilibrium. It is not known in general just how much additional information must be communicated among the agents in order to guarantee convergence. There are evidently tradeoffs among a number of relevant performance characteristics. These include:
(a) The set of environments, or parameter values, for which the process is guaranteed to converge;

(b) the information to be exchanged;

(c) the computations to be performed;

(d) the speed of convergence.

The process presented in this paper does not require that the system of equations to be solved have full rank (that equilibrium be unique). Thus, with respect to the first performance characteristic noted above, the process applies to any system consisting of a finite number of linear equations.[3]

The process, like those described in [3], uses a special agent called the coordinator. It also involves N private agents. Each private agent knows his/her own equilibrium equation. In the linear case this amounts to knowing the (N+1)-vector of coefficients, including the constant term, of that equation. The coordinator knows

[3]After completing this paper I learned, from Alvin Bayliss, of an algorithm due to Kaczmarz [1] and later discussed by Tanabe [7]. Starting from an arbitrary point of the space that algorithm proceeds by projecting that point orthogonally onto the hyperplane defined by one equation, say the first equation. It continues by projecting the resulting point onto the second hyperplane, and so on iteratively. The iteration converges in the limit; it is stopped when the difference between successive approximations becomes small enough. Although the algorithm presented in this paper does not involve orthogonal projection, and requires only a fixed number of steps, depending on the number of equations, the underlying geometric idea seems similar to that of Kaczmarz. A systematic comparison of the performance of the algorithm presented here with Kaczmarz's and others awaits computational testing.

only that the space is R^N, and does not know any of the equations or coefficients of them. The coordinator communicates points of R^N to the private agents, chosen according to the rules of the process described in the next section; each private agent responds with a real number. Thus, the message exchange requirements of this process are larger than those of the processes described in [3] in requiring an additional real variable to be transmitted, where the processes in [3] required a binary variable.

The computational requirements of this process are as follows. Each agent must compute the value of a linear function, the one associated with his equilibrium manifold. The coordinator must make two calculations; one involves the difference of two numbers and the ratio of a third number to that difference; the other involves the multiplication of an N-vector by a scalar, performed twice, and the sum of the two resulting vectors. The computations that involve vectors lend themselves to vectorization.

This process arrives at the equilibrium in N steps when the solution is unique, and finds the solution manifold in fewer than N steps when the system is singular. Computational requirements grow with N as N^3; the number of computations per step is of order to N^2.

Sections 2.1 and 2.2 present the process in an informal intuitive way. The nonsingular case is presented in 2.1; the singular case in 2.2. Computational and communication requirements are discussed briefly in Section 2.3. Section 2.4 contains a more formal treatment, together with proofs of propositions that 2.1 and 2.2 rely on.

Section 2. The Process

There are N agents, $1, \ldots, N$. Each agent i is characterized by a vector of parameters $\theta^i \in \mathbb{R}^{k_i}$. There is a message space M, an open subset of $\mathbb{R}^N$. In what follows we assume for simplicity that $M = \mathbb{R}^N$. We assume that there are functions g^i whose zero set defines the equilibrium we are seeking. In a familiar example, the functions g^i are excess demand functions. Setting them equal to zero gives the equilibrium condition.

In this paper, as in [3], we assume the g^is are linear.

$$g^i(x, \theta^i) \equiv a^i \cdot x - c^i \qquad i = 1, \ldots, N$$

where α^i is a function of agent i^s parameters θ^i.

I.e., $\alpha^i = (\alpha^i_1(\theta^i), \alpha^i_2(\theta^i), \ldots, \alpha^i_N(\theta^i))$.

For the present purpose it is not necessary to distinguish the underlying parameters θ^i from the coefficients α^i.

Thus,

$$\alpha^i(\theta^i) \cdot x - c^i = \alpha^i \cdot x - c^i \quad i = 1, \ldots, N.$$

The system (2.1) of N equations

$$\alpha^i \cdot x - c^i = 0 \quad i = 1, \ldots, N \tag{2.1}$$

defines the equilibrium.

There is also a coordinator, who knows only that there are N agents and that the space in which equilibrium resides is $I\!R^N$.

2.1 The Nonsingular Case

It is assumed for now that (2.1) has a unique solution, i.e. that the $N \times N$ matrix $A = ((\alpha^i))$ has full rank. The case where A has less than full rank is discussed in Section 2.2.

Another interpretation of (2.1) is that each equation is "known" by (or in the memory of) a single one of N processors and not by any other. The problem is to find the solution by a "good" distributed method. A good method is one that finds the solution, but does not require too much communication among the processors nor too much computation by them. These are at present intuitive notions, not formalized in this paper.

In the present (nonsingular) case, the process takes N steps.

Step 1

Initially the coordinator chooses the standard basis of $I\!R^N$, denoted

$$v^0_i = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} i^{th}\text{position} \qquad \text{i} = 1, \ldots, N.$$

and the origin, denoted

$$v_0^0 = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Let

$$V^0 = \begin{pmatrix} v_{01} & v_{11} & \cdots & v_{N1}^0 \\ \vdots & & & \\ v_{0N}^0 & v_{1N}^0 & \cdots & v_{NN}^0 \end{pmatrix}.$$

The coordinator uses these points to find N points that satisfy one of the equations (2.1), say, the first equation. This is done as follows.

The coordinator announces the $N+1$ points $v_0^0, \ldots, v_N^0$ to agent 1.

Agent 1 replies with the value of the linear function characterizing 1's equilibrium manifold. Let $\sigma^1(x)$ denote 1's response to x. Then

$$\sigma^1(v_i^0) = g^1(v_i, \theta^1) = a^1 \cdot v_i^0 - c^1 \quad i = 0, 1, \ldots, N.$$

Generally [4]

$$\sigma^1(v_0^0) \neq \sigma^1(v_i^0) \quad for\ i = 1, \ldots, N.$$

The coordinator uses the N lines

$$\begin{aligned} L(v_0^0, v_i^0) &= \{x \in I\!R^N \mid x = (1 - \lambda_i^0)\, v_0^0 + \lambda_i^0 v_i^0 \quad \lambda_i \in I\!R\} \\ &= \{x \in I\!R^N \mid x = v_0^0 + \lambda_i^0\, (v_i^0 - v_0^0) \quad \lambda_i \in I\!R\}. \end{aligned}$$

[4] If for some i

$$\sigma^1(v_0^0) = \sigma^1(v_i^0),$$

then the coordinator perturbs v_i^0 slightly, say to

$$v_i' = (0, \ldots, 1, \epsilon, 0, \ldots, 0),$$

preserving linear independence of the resulting set. The coordinator finds a perturbation v_i' such that $\sigma^1(v_i') \neq \sigma^1(v_o^0)$ ensuring that $L(v_o, v_i')$ is not parallel to the linear manifold

$$H^1(a^1, c^1) = \{x \epsilon I\!R^N | g^1(x, \theta^1) = 0\} = \{x \epsilon R^N | a^1 \cdot x = c^1\}.$$

to find N points that satisfy the equation

$$a^1 \cdot x - c^1 = 0. \tag{2.2}$$

Let these points be $v_1^1, \ldots, v_N^1$. They may be written

$$\begin{aligned} v_i^1 &= (1 - \hat{\lambda}_i^0)\, v_0^0 + \hat{\lambda}_i^0\, v_i^0 \quad i = 1, \ldots, N \\ &= v_0^0 + \hat{\lambda}_i\, (v_i^0 - v_0^0) \end{aligned}$$

where

$$\hat{\lambda}_i^0 = \frac{-\sigma^1(v_0^0)}{\sigma^1(v_i^0) - \sigma^1(v_0^0)} \qquad \text{for i} = 1, \ldots, N, \text{ provided that } \sigma^1(v_0^0) \neq 0.^5$$

Since v_0^0 is the origin, the formula for v_i^1 reduces to

$$v_i^1 = \hat{\lambda}_i^0\, v_i^0 \qquad \text{for i} = 1, \ldots, N.$$

We show in Section 2.4 that these N points satisfy the first equation, i.e., that

$$a^1 \cdot v_i^1 - c^1 = 0 \quad \text{for i} = 1, \ldots, N. \tag{2.3}$$

This is the end of Step 1.

Step 2

The coordinator uses the N points v_i^1, $i = 1, \ldots, N$ to generate $N - 1$ lines

$$\begin{aligned} L(v_1^1, v_i^1) &= \{x \in I\!R^N \mid x = (1 - \lambda_i^1)\, v_1^1 + \lambda_i^1 v_i \quad \lambda_i^1 \in I\!R\} \\ &= \{x \in I\!R^N \mid x = v_1^1 + \lambda_i^1\, (v_i^1 - v_1^1) \quad \lambda_i^1 \in I\!R\}, \quad i = 2, \ldots, N \end{aligned} \tag{2.4}$$

[5] Note that the vectors $v_i^0 - v_0^0 = v_i^0$ $i = 1, \ldots, N$ are linearly independent.
If $\sigma^1(v_0^0) = a^1 x - c^1 = 0$, then, since $v_0^0 = 0$, it follows that $c^1 = 0$. Thus equation 2.2 defines an $N-1$ dimensional hyperplane through the origin. Since the vectors $v_1^0, \ldots, v_N^0$ are linearly independent, there must be at least one value of j, say $j\prime$, such that

$$a^1 v_{j\prime}^0 \neq 0.$$

It follows that $\sigma^1(v_{j\prime}^0) \neq 0$. We then define the lines

$$L(v_{j\prime}^0, v_j^0) = \{x \epsilon I\!R^n \mid x = (1 - \lambda_j^0) v_{j\prime}^0 + \lambda_j^0 v_j^0 \quad j \in \{0, \ldots, N\} \backslash \{j\prime\} \quad \text{and} \quad \lambda_j^0 \in \mathbf{R}\}$$

and correspondingly

$$\lambda_j^0 = \frac{-\sigma^1(v_{j\prime}^0)}{\sigma^1(v_j^0) - \sigma^1(v_{j\prime}^0)}\ .$$

Because of the linear independence of the initial vectors $v_i^0 = (v_i^0 - v_0^0)$ $i = 1, \ldots, N$, the lines $L(v_1^1, v_i^1)$ can be used to find $N-1$ points that satisfy the second equation of (2.1).

These points are

$$v_i^2 = (1 - \hat{\lambda}_i^1)v_1^1 + \hat{\lambda}_i^1 v_i^1 = v_1^1 \quad + \quad \hat{\lambda}_i^1(v_i^1 - v_1^1)$$
$$i \quad = 1, \ldots, N.$$

Where

$$\hat{\lambda}_i = \frac{-\sigma^2(v_1^1)}{\sigma^2(v_i^1) - \sigma^2(v_1^1)} \ .$$

Thus, $v_i^2, \quad i = 2, \ldots, N$ all satisfy the second equation of (2.1), i.e.,

$$a^2 \cdot v_i^2 - c^2 = 0 \quad \text{for i} = 2, \ldots, N. \tag{2.5}$$

But these points also satisfy (2.2), because the lines $L(v_1^1, v_i^1)$ are contained in the $N-1$ dimensional linear manifold generated by the N points $v_i^1, \ i = 1, \ldots, N$. Hence every point on such a line, including the points $v_i^2, \ i = 2, \ldots, N$, also belong to that manifold. Therefore, at the end of Step 2, the coordinator has $N-2$ points that satisfy the first 2 equations of (2.1).

Step t. $(1 < t \leq N)$.

The coordinator begins with $N-t+1$ points that satisfy the first $t-1$ equations of (2.1). Let these points be $v_{t-1}^{t-1}, v_t^{t-1}, \ldots, v_N^{t-1}$. The coordinator uses the $N-t$ lines

$$L(v_{t-1}^{t-1}, v_i^{t-1}), \quad i = t, \ldots, N.$$

to find $N-t$ points

$$v_t^t, \ldots, v_N^t.$$

where

$$v_j^t = (1 - \hat{\lambda}_j^{t-1})v_{t-1}^{t-1} + \hat{\lambda}_j^t \ v_j^{t-1},$$

and

$$\hat{\lambda}_j^t = \frac{-\sigma^t(v_{t-1}^{t-1})}{\sigma^t(v_j^{t-1}) - \sigma^t(v_{t-1}^{t-1})} \quad j = t+1, \ldots, N \quad \text{provided that}$$

(i) $\sigma^t(v_{t-1}^{t-1}) \neq 0$, and

(ii) it is not the case that for all $j \in \{t, \ldots, N\}$ $\sigma^t(v_j^{t-1}) = 0$. [6]

These $N - t$ points satisfy the first t equations of (2.1). I.e.,

$$\alpha^i \cdot v_j^t - c^i = 0 \quad \text{for} \quad \text{i} = 1, \ldots, \text{t} \quad \text{and} \quad \text{j} = \text{t}, \ldots, N. \tag{2.6}$$

At the N^{th} step, the process generates one point that satisfies all N of the equation system (2.1).[7]

The various assertions needed to establish the process works as described are proved in Section 2.4.

2.2 The singular case

If the matrix of the system of equilibrium equations has rank r, and r is less than N, then there is a linear manifold of solutions whose dimension is $N - r$. In these cases, the process finds the full set of solutions. For convenience suppose the agents are labelled to match the order in which they are addressed by the coordinator, so that agent s is addressed in step s, for $s = 1, \ldots, N$. Suppose further that the equation of agent s is the first equation (in the order $1, 2, \ldots, s$) with the property that equation s is a linear combination of the preceding ones.
If $A = ((\alpha_j^i))$ $i = 1, \ldots, N$, $j = 1, \ldots, N$, has rank $r < N$, there must be at least one such equation.

Then at step s the procedure has generated a set of $N + 1 - s$ (affinely) independent points that satisfy each of the first s equations.

[6] If $\sigma^t(v_{t-1}^{t-1}) = 0$, the coordinator can choose another point among the v_j^{t-1}, $j = \{t, \ldots, N\}$ as the common point of the lines $L(v_j^{t-1}, v_k^{t-1})$, as discussed in footnote 4. Condition (ii) does not arise if the equations 2.1 have full rank. The case where (2.1) has less than full rank is discussed in Sections 2.2 and 2.4.

[7] If equations (2.1) have less than full rank, the process will stop after fewer than N steps, and will give the full linear manifold of solutions of (2.1). This is discussed Section 2.2 and in Section 2.4.

Thus, after $s-1$ steps we have $N+1-(s-1) = N-s+2$ points that satisfy equations $1,\ldots,s-1$, and therefore lie in the linear manifold $H^1 \cap H^2 \cap \ldots \cap H^{s-1}$, where,

$$H^t = H^t(\alpha^t, c^t) = \{\ x \in I\!R^N \mid x = v_t^t + \lambda(v_j^t - v_t^t) \quad j = t+1,\ldots,N,\ \lambda \in I\!R\} \qquad t = 1,\ldots,N$$

But if equation s is a linear combination of equations $1,\ldots,s-1$, then these $N-s+2$ points also satisfy equation s, and so lie in

$$H^1 \cap \ldots \cap H^{s-1} \cap H^s.$$

This will be revealed to the coordinator, because at step s the responses from agent s will be $\sigma^s(v_j^{s-1}) = 0$ for each of the $N-s+2$ points $v_{s-1}^{s-1},\ldots,v_N^{s-1}$. Consequently $\hat{\lambda}_j^s = 0$ for all $N-x+2$ points.

In that case the procedure continues with the same lines $L(v_{s-1}^{s-1}, v_j^{s-1})$ for j = s − 1,...,N in step $s+1$, as in step $s-1$. Thus, the points generated by intersecting these lines with the hyperplane H^{s+1} given by the equation $\alpha^{s+1} \cdot x = c^{s+1}$ will span a linear manifold of dimension $N-s+1$ in H^{s+1}, instead of one of dimension $N-s$, as would be the case if equation s were not dependent.[8]

Thus, for every such dependent equation the dimension of the intersection up to that step is increased by 1. After the N^{th} step the process has $N-r+1$ (affinely) independent points, spanning a linear manifold of dimension $N-r$. The manifold is the solution manifold of (2.1). (In case $r = N$, then $N-r = 0$, and the solution manifold consists of a single point).

To illustrate the process suppose $N = 3$, and r = 2.
In this case equations (2.1) have the form:

$$\alpha^1 \cdot x = c^1 \qquad (2.2.1)$$
$$\alpha^2 \cdot x = c^2 \qquad (2.2.2)$$
$$\alpha^3 \cdot x = c^3 \qquad (2.2.3)$$

Since $r = 2$ there exists scalars $\beta_1, \beta_2, \beta_3$, not all 0, such that

$$\beta_1(\alpha^1, c^1) + \beta_2(\alpha^2, c^2) + \beta^3(\alpha^3, c^3) = 0 \qquad (2.7)$$

[8] If equation $s+1$ of (2.1) is also dependent on the preceding $(s-1)$ equations, then the same lines are used to generate points in H^{s+2}.

Consider 2 cases.

Case 1. $\mathcal{B}_3 = 0,\ \mathcal{B}_1 \neq 0 \neq \mathcal{B}_2.$

In that case, equation (2.1.2) is proportional to equation (2.2.1), i.e.,

$(a^2, c^2) = \beta(a^1, c^1)$, where $\beta = \mathcal{B}_2/\mathcal{B}_1$.

Case 2. $\mathcal{B}_2 \neq 0,\ \mathcal{B}_3 \neq 0.$

In that case equation (2.2.2) is a linear combination of equations (2.2.1) and (2.2.3).

In Case 1, starting from v_0, v_1, v_2, v_3 in $I\!R^3$, after Step 1 we have

$$v_1^1, v_2^1, v_3^1$$

each satisfying equation (2.1.1).

Step 2 reveals that these same points also satisfy equation(2.1.2).

Hence $v_1^2 = v_2^1, v_2^2 = v_2^1\ v_3^2 = v_3^1$.

Hence, at Step 3 the lines $L(v_1^2, v_2^2)$ and $L(v_1^2, v_3^2)$ (the same as $L(v_1^1, v_2^1)$ and L(v_1^1, v_3^1) generate 2 distinct points satisfying equation (2.2.3).

Since this is the last step, the process ends with the

$N - r = 3 - 2 = 1$-dimensional linear manifold satisfying all three equations.

In Case 2 the process starting from v_0, v_1, v_2, v_3, generates v_1^1, v_2^1, v_3^1 after Step 1, and the points v_2^2, v_3^2 after Step 2. But, since (2.7) implies that the third equation is a linear combination of the first two, it follows that v_2^2, v_3^2 satisfy the third equation and hence $(v_3^2 - v_2^2)$ spans the 1 -dimensional set of solutions of (2.2).

2.3 Complexity and Communication Requirements

2.3.1 Complexity

The number of elementary computations required by this process may be estimated. Counting addition, subtraction, multiplication and division of real numbers as elementary operations, we have the following.

At the t^{th} step the coordinator computes $t-1$ values $\hat{\lambda}_j^t$ $j = t+1, \ldots, N$. Each of these requires one subtraction and one division, two elementary operations. The coordinator must then perform two vector additions (actually a subtraction in one case) and one multiplication of a vector by a scalar. This amounts to $3N + 3(N-t)$ elementary operations at step t.

Adding these, for a fixed N, we get

$$3N^2 + \frac{3N(N+1)}{2} = \frac{9N^2 + 3N}{2}$$

At step t, Agent t must compute σ^t at each of the points v_j^t $j = t, \ldots, N$. Each such evaluation involves taking the inner product of 2 vectors and subtracting a number from the result. This is $N+1$ elementary operations per evaluation.

The first agent does N of these; thus the first agent performs $N(N+1)$ elementary operations.

The t^{th} agent performs

$$(N-t)(N+1)$$

such operations, the last agent performing just

$$(N+1)$$

operations.

The total of these over agents and steps is

$$N(N+1)^2 = \frac{N^3 + 2N^2 + N}{2}$$

The total of what the coordinator and the agents do is

$$\frac{1}{2}(N^3 + 11N^2 + 4N),$$

though no one agent, or processor, performs more than N^2 operations.

2.3.2 Communication

In this process communication takes place between the coordinator and each agent separately. When there are N agents, the coordinator sends $(N+1)$ points of $I\!R^N$ to the first agent, N to the second, $N-1$, to the third, and in general, $N-t+2$ to the t^{th} agent. He receives from each of them one real number per point – thus, $N-t+2$ from the t^{th} agent.

It appears to be the case that this algorithm minimizes the amount of communication needed. At each step only the minimum information needed to identify the intersection of hyperplanes up to that step is transmitted. However, this is as yet a conjecture, since no formal analysis of this point has so far been done.

Note that if $N \geq 3$ it is not possible for all the agents to transmit their equations to the coordinator via these messages. It is always possible for the first agent to use the $N+1$ responses allowed to encode the coefficients of the first equation. (There are N coefficients after normalization.) The second agent can also encode the N coefficients of the second equation. The third agent has N normalized coefficients, but only $N-1$ real numbers, and so cannot transmit his equation. When $N \geq 3$ the system is solved without knowledge of the whole system accumulating anywhere.

2.4 Propositions

To begin with recall some well-known facts about linear manifolds and subspaces in $I\!R^N$.

These are summarized in:

Proposition 1. The following statements are equivalent:

1) $p+1$ points, $p \leq N$, $w_0, w_1, \ldots, w_p$ determine a linear manifold of dimension p:

2) the rank of the matrix

$$W = \begin{pmatrix} w_{01} & w_{11} & \cdots & w_{p1} \\ \vdots & & & \vdots \\ w_{0N} & w_{1N} & & w_{pN} \end{pmatrix}$$

is p:

3) the p vectors $(w_1 - w_0)$, $(w_2 - w_0)$, ..., $(w_p - w_0)$ are linearly independent.

(The $p+1$ points $w^0, w_1, \ldots, w_p$ are called *affinely independent* if the vectors $(w_i - w_j)$ $j\prime \in \{0, \ldots, p\}$ are linearly independent).

Proposition 2 If $(w_1 - w_0)$, ..., $(w_p - w_0)$ is a linearly independent set of vectors in $\mathbb{R}^N$, then for any $i, j \in \{1, \ldots, p\}$ with i $\neq$ j, the lines $L(w_0, w_i)$ and $L(w_0, w_j)$ have exactly the point w_0 in common, i.e.

$$L(w_0, w_i) \cap L(w_0, w_j) = \{w_0\}$$

Proof. If $x \in L(w_0, w_i) \cap L(w_0, w_j)$ then,
for some λ_i and λ_j,

$$x = w_0 + \lambda_i(w_i - w_0) = w_0 + \lambda_j(w_j = w_0).$$

Hence

$$\lambda_i(w_i - w_0) - \lambda_j(w_j - w_0) = 0.$$

Since $(w_i - w_0)$ and $(w_j - w_0)$ are linearly independent, it follows that

$$\lambda_i = \lambda_j = 0.$$

Therefore

$$x = w_0 + 0(w_i - w_0) = w_0$$

Next we describe the process more formally.

Consider the first step of the process described in Section 2.1. This step can be represented by the matrix equation

$$V^1 = V^0 \Lambda^0. \tag{2.8}$$

where

$$V^0 = \begin{pmatrix} c_{01} & c_{11} & \dots & c^0_{N1} \\ \vdots & & & \\ c^0_{0N} & c^0_{1N} & \dots & c^0_{NN} \end{pmatrix}.$$

$$\Lambda^0 = \begin{bmatrix} (1-\hat{\lambda}^0_1) & \dots & & (1-\hat{\lambda}^0_N) \\ \hat{\lambda}^0_1 & & & \\ & \hat{\lambda}^0_2 & & 0 \\ & & \ddots & \\ 0 & & & \ddots \quad \hat{\lambda}^0_N \end{bmatrix}.$$

and

$$V^1 = \begin{pmatrix} c^1_{11} & \dots & c^1_{N1} \\ \vdots & & \\ c^1_{1N} & & c^1_{NN} \end{pmatrix} = \begin{pmatrix} (1-\hat{\lambda}^0_1)c^0_{01} + \hat{\lambda}^0_1 c^0_{11} & \dots & (1-\hat{\lambda}^0_N)c^0_{01} + \hat{\lambda}^0_N c^0_{N1} \\ \vdots & & \\ (1-\hat{\lambda}^0_1)c^0_{0N} + \hat{\lambda}^0_1 c^0_{1N} & \dots & (1-\hat{\lambda}^0_N)c^0_{0N} + \hat{\lambda}^0_N c^0_{NN} \end{pmatrix}$$

$$= ((c^1_1, \dots, c^1_N)).$$

where

$$\hat{\lambda}^0_j = \frac{-\sigma^1(c^0_0)}{\sigma^1(c^0_j) - \sigma^1(c^0_0)} \quad \text{for j} = 1, \dots, N.$$

First, we establish:

Proposition 3. The points $c^1_1, \dots, c^1_N$ satisfy

$$a^1 \cdot c^1_j - c^1 = 0 \quad \text{for j} = 1, \dots, N \tag{2.9}$$

Proof. Suppose (2.9) is satisfied, and solve for $\hat{\lambda}_j$. Thus,

$$\begin{aligned} 0 &= a^1 \cdot ((1-\lambda_j)\, v_0^0 + \lambda_j v_1^0) - c^1 \\ &= (1-\lambda_j)\, a^1 \cdot v_0^0 + \lambda_j\, a^1 \cdot v_1^0 - c^1 \\ &= a^1 \cdot v_0^0 + \lambda_j (a^1 \cdot v_j^0 - a^1 \cdot v_0^0) - c^1 \end{aligned}$$

Thus, $$\frac{c^1 - a^1 \cdot v_1^0}{a^1 \cdot v_1^0 - a^1 \cdot v_0^0} = \hat{\lambda}_j.$$

But $$c^1 - a^1 \cdot v_0^0 = -\sigma^1(v_0^0).$$

and $$a^1 \cdot v_1^0 - a^1 \cdot v_0^0 = a^1 \cdot v_1^0 - c^1 - a^1 \cdot v_0^0 + c = \sigma^1(v_1^0) - \sigma^1(v_0^0).$$

Hence $$\frac{-\sigma^1(v_0^0)}{\sigma^1(v_j^0) - \sigma^1(v_0^0)} = \hat{\lambda}_j.$$

Verifying (2.9) directly.

$$a^1 \cdot v_j^1 - c^1 = \left[1 - \frac{-(a^1 v_0^0 - c^1)}{a^1 v_1^0 - a^1 v_0^0}\right] a^1 v_0^0 + \frac{-(a^1 v_0^0 - c^1)}{a^1 v_1^0 - a^1 v_0^0}\, a^1 \cdot v_1^0 - c^1$$

$$= \frac{[a^1 v_1^0 - c^1] a^1 v_0^0 - (a^1 v_0^1 - c^1)\, a^1 v_1^0}{a^1 v_1^0 - a^1 v_0^0} - c^1$$

$$= \frac{1}{a^1 v_1^0 - a^1 v_0^0} \left[(a^1 v_1^0 - c^1)\, a^1 v_0^0 - (a^1 v_0^0 - c^1) a^1 v_1^0\right] - c^1$$

$$= \frac{1}{a^1 v_1^0 - a^1 v_0^0} \left[-c^1 a^1 v_0^0 + c^1 a^1 v_1^0\right] - c^1$$

$$= \frac{c^1 (a \cdot v_1^0 - a^1 v_0^0)}{a^1 v_1^0 - a^1 v_1^0} - c^1 = 0.$$

Next, Proposition 4 is immediate.

Proposition 4.

If x and y satisfy any equation of (2.1), then, so does

$$z = (1-\lambda)x + \lambda y \qquad \lambda \in I\!R.$$

Proof. Suppose

$$a^i \cdot x - c^i = 0 = a^i \cdot y - c^i \quad \text{for some } i \in \{1, \ldots, N\}.$$

$$\begin{aligned} \text{Then} \quad a^i \cdot z &= (1-\lambda) a^i \cdot x + \lambda a^i \cdot y \\ &= a^i \cdot x + \lambda (a^i \cdot y - a^i x) \\ &= c^i + \lambda (c^i - c^i) = c^i \end{aligned}$$

So $a^i \cdot z - c^i = 0$

It remains to show

Proposition 5. The $N-1$ vectors

$$(c_j^1 - c_1^1) \quad j = 2, \ldots, N$$

form a linearly independent set.

Proof. According to Proposition 1 an equivalent statement is that

$$R(V^{-1}) = N,$$

where $R(N)$ denotes the rank of the matrix N.
Now,

$$V^{-1} = V^{-0} \Lambda^0 \tag{2.10}$$

where

$$R(V^{-0}) = N$$

and

$$R(\Lambda^0) = R \begin{pmatrix} \hat{\lambda}_1^0 & & 0 \\ & \ddots & \\ 0 & & \hat{\lambda}_N^0 \end{pmatrix}$$

The latter matrix has rank N, provided that none of the scalars $\hat{\lambda}_j \quad j = 1, \ldots, N$ is 0.[9]

---

[9]Suppose that for some $j\prime \in \{1, \ldots, N\}$, $\lambda^0_{j\prime} = 0$

$$\text{then} \quad v^1_{j\prime} = v^0_0(1 - \lambda^0_{j\prime}) + \lambda^0_{j\prime} v^0_{j\prime} = v^0_0.$$

But, since for all $j = 1, \ldots, N$,

$$\lambda^0_j = \frac{-\sigma^1(v^0_0)}{\sigma^1(v^0_j) - \sigma^1(v^0_0)},$$

it follows that $\lambda^0_j = 0$ for all $j = 1, \ldots, N$, since $\sigma^1(v^0_0) = \alpha^1 \cdot v^0_0 - c^1$, *and* $\sigma^1(v^0_0) = 0$ imply $\alpha^1 \cdot v^0_0 = c^1$.

Further, since $v^0_0 = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$, it follows that $\alpha^1 \cdot v^0_0 = 0 = c^1$. Geometrically, the case $\lambda^0_{j\prime} = 0$ arises if and only if $c^1 = 0$, which means that the hyperplane defined by the first equation

$$\alpha^1 \cdot x = c^1$$

contains the origin. If that case should occur, the coordinator would recognize it by the fact that

$$\sigma^1(v^0_0) = 0.$$

(The reply from agent 1 to v^0_0 is 0.)

He can then replace the point $v^0_0 = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$ by one the N points $v^0_1, \ldots, v^0_N$, say v^0_k, for which $a^1 \cdot v^0_k \neq 0$. There must be at least one such point because the vectors $v^0_1, \ldots, v^0_N$ span an N-dimensional subspace, while $\alpha^1 \cdot x = 0$ defines an $N - 1$ dimensional subspace.

Alternatively, v^0_0 can be replaced by one of the N- points.

$$v^0_i(\epsilon) = \begin{pmatrix} 0 \\ \vdots \\ \epsilon \\ 0 \\ \vdots \\ 0 \end{pmatrix} i^{th} \text{coordinate}$$

which has ϵ, $0 < |\epsilon| < 1$, in the i^{th} coordinate and $0's$ elsewhere.
For some value of i, $\sigma^1(v^0_i(\epsilon) \neq 0$. *(cont. on next page)*

Since the rank of V^0 is also N, it follows from (2.10) that

$$R(V^1) = N.$$

By Proposition 1, the N points $v_1^1, \ldots, v_N^1$ determine the $N-1$ dimensional linear manifold that contains them. By Proposition 3 this is the hyperplane defined by equation (2.9).

Now

$$V^2 = V^1 \Lambda^1,$$

where

$$\Lambda^1 = \begin{pmatrix} (1-\hat{\lambda}_2^1 & \cdots & & (1-\hat{\lambda}_N^1 \\ \hat{\lambda}_2^1 & & & \\ & \hat{\lambda}_3^1 & & 0 \\ & 0 & \ddots & \\ & & & \hat{\lambda}_N^1 \end{pmatrix}$$

which generally has rank $N-1$, except in the case where $\sigma^2(v_1^1) = 0$.[10]
The following Proposition shows that linear independence of the vectors

$$(v_j^t - v_t^t)$$

is preserved at each step $t = 2, \ldots, N$.

Proposition 6. If the vectors

$$(v_j^{t-1} - v_{t-1}^{t-1}) \quad \text{j} = \text{t}, \ldots, N,$$

---

9 (cont) The lines

$$L(v_{jt}^0(\epsilon), v_j^0) = \{x \in I\!R^N \mid x = (1-\lambda_j)\, v_t^0(\epsilon) + \lambda_j v_j^0 \quad \lambda_j \in I\!R\} \qquad j = 1, \ldots, N$$

intersect uniquely at $v_j^0(\epsilon)$. The coordinator can proceed with the process, which is again in the case $\lambda j(\epsilon) \neq 0$ for all j.

[10] See footnote 9.

are linearly independent, then so are the vectors

$$v_i^t - v_t^t \quad i = t+1, \ldots, N.$$

Proof. By Proposition 1 it suffices to show that if the matrix V^{t-1} has rank $N-t+1$, then the matrix V^t has rank $N-t$.

But

$$V^t = V^{t-1} \Lambda^{t-1}$$

$$\text{where } \Lambda^{t-1} = \begin{pmatrix} (1-\hat{\lambda}_t^{t-1} & & \cdots & (1-\hat{\lambda}_N^{t-1} \\ \hat{\lambda}_t^{t-1} & & & \\ & \hat{\lambda}_{t+1}^{t-1} & & 0 \\ & 0 & \ddots & \\ & & & \hat{\lambda}_N^{t-1} \end{pmatrix}.$$

Now, except in the case where $\hat{\lambda}_t^{t-1} = 0$, i.e., the exceptional case where v_{t-1}^{t-1} lies in the hyperplane defined by equation

$$\alpha^1 \cdot x = c^t,$$

the matrix Λ^{t-1} has rank $N-t$. [11]

[11] In the exceptional case, the coordinator chooses another of the points v_j^{t-1} $j = t, \ldots, N$ to serve as the common point of the lines, or perturbs v_{t-1}^{t-1}, for example, by

$$v_{t-1}^{t-1}(\varepsilon, j) = (1-\gamma)v_{t-1}^{t-1} + \gamma v_j^{t-1} \quad 0 < \gamma < 1, \quad j = 5, \ldots, N.$$

If $\sigma^t(v_j^{t-1}) = 0$ for all $j \in \{t-1, t, \ldots, N\}$ the λ_j^{t-1} are not defined. In that case the points v_j^{t-1} satisfy

$$\alpha^t v_j^{t-1} - c^t = 0 \text{ for j} = t-1, t, \ldots, N. \tag{2.11}$$

By construction, the points v_j^{t-1} satisfy the equations $\alpha^i \cdot x - c^i = 0$ for $i = 1, \ldots, t-1$. Thus,

$$\alpha^i v_j^{t-1} - c^i = 0 \quad i = 1, \ldots, t-1, \text{ and}$$
$$j = t-1, \ldots, N. \tag{2.12}$$

In more geometric language, the $N-t+1$ points v_j^{t-1} determine a unique $(N-t)$-dimensional linear manifold. Since these points lie in $H^1 \cap H^2 \ldots \cap H^{t-1}$, and that intersection is $(N-t)$ dimensional, it is the linear manifold defined by equation (2.12). If α^t were independent of $\alpha^1, \ldots, \alpha^{t-1}$, then the $N-t+1$ points v_j^{t-1} would determine an $(N-t)$-dimensional linear *(cont. on next page)*

Hence the dimension of the linear manifold they span at step t is $N-t$. When $t=N$, the matrix V^N has only one entry, the unique point in the solution manifold of (1), which $N-N=0$ dimensional.

11 (cont) manifold in the hyperplane H^t given by (2.11) and therefore because of (2.12) in the the intersection

$$H^1 \cap \ldots \ldots \cap H^{t-1} \cap H^t.$$

This intersection is an $N-t-1$ dimensional manifold. This contradicts the affine independence of $v_{t-1}^{t-1}, \ldots, v_N^{t-1}$.

References

[1] KACZMARZ, S., Angenäherte Auflösung von Systemen linearer Gleichungen, *Acad. Polon. Sciences et Lettres*, **A**, 355-357 (1937).

[2] KIEFER, N. AND V. BALA,"Obtaining stability from chaos," Working Paper No. 91-05 (1991), Center for Analytic Economics, Cornell University, Ithaca, New York.

[3] S. REITER AND C. SIMON, Decentralized dynamic processes for finding equilibrium, *Journal of Economic Theory* (forthcoming).

[4] H.E. SCARF, Some examples of global instability of the competitive equilibrium, *International Economic Review* **1** (1960), 157-172.

[5] H.E. SCARF, The approximation of fixed points of a continuous mapping, *Journal of Applied Mathematics* **15, No. 5** (1967).

[6] S. SMALE, A convergent process of price adjustment and global Newton methods, *Journal of Mathematical Economics* **3** (1976), 107-120.

[7] K. TANABE, Projection method for solving a singular system of linear equations and its applications, *Numerical Mathematics* **17**, 203-214 (1971).