168
4-8-80
Total
1209

MASTER

DOE/EIA-451887

Research into the Methodology of the LEAP Model

December 1979

Prepared for
U.S. Department of Energy
Energy Information Administration
Assistant Administrator for Applied Analysis
Under Contract No. 451887-S

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

DOE/EIA-451887
Dist. Cat. UC-13

Research into the Methodology of the LEAP Model

December 1979

Prepared by
James E. Falk, Garth P. McCormick,
Richard M. Soland
The George Washington University
School of Engineering and Applied Science
Washington, DC

Prepared for
U.S. Department of Energy
Energy Information Administration
Assistant Administrator for Applied Analysis
Long-Term Analysis Division
Office of Integrative Analysis
Washington, DC 20461
Under Contract No. 451887-S

DISCLAIMER

This book was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Released for printing: March 11, 1980

ABSTRACT

This report summarizes the research carried out in the analysis of the LEAP model and its underlying methodology. We have divided the report into two parts, with the first part describing the LEAP model (as opposed to the algorithm used to solve the model), and placing it in context with the general economic theory. A basic result of the first part emphasizes that LEAP is mathematically a simulation model described by sets of equations governing the relations between quantities and prices of various commodities at various "processes" over time. It is not an optimization model, but can be conceptualized as an n-person sequential game.

The equations that constitute LEAP are structured in such a way that their solution (if one exists) can be conveniently handled with the Successive Over-Relaxation Method. The second part of this report treats various aspects and extensions of this method, as well as alternate methods commonly employed to solve simultaneous equations. Remarks addressing "decomposition" are also developed in this part.

Applied Analysis Preface

The Energy Information Administration regularly receives reports from contractors. On occasion, a report is of interest and worthy of distribution in its original form. Reports released in this manner have not been edited by EIA or circulated for review, and thus cannot be endorsed as EIA products. This contractor report has been released under this general policy without EIA endorsement.

This report was written under an original task supervised by the Brookhaven National Laboratory to investigate mathematical programming and related aspects of the LEAP methodology. Of specific interest were two questions:

- (i) whether the LEAP algorithm has a deterministic equivalent mathematical programming problem (MPP);
- (ii) were any related MPP algorithms or concepts of value in accelerating convergence or proving stability of the LEAP solution.

The results of the report show why LEAP has no deterministic equivalent MPP and that the procedure only analog is to the solution of simultaneous non-linear equations. Several results are established for the existence, uniqueness and stability of equilibria and finally a series of solution algorithms are proposed and evaluated.

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

TABLE OF CONTENTS

PART I: THE BASIC MODEL

A. DESCRIPTION	1
1. The Diamond Example	8
2. The Two Period Example	10
B. LEAP AND OTHER ECONOMIC MODELS	13
C. INTERPRETATION OF LEAP AS AN OPTIMIZATION MODEL	17
1. Equilibrium as Maximization	17
Computation of $p(q)$	19
Computation of $D(F)$	20
Assumption of relationship between cost of final goods, and cost of basic supplies.	20
Equivalent consumer maximization problems.	21
Equivalence of equilibrium and maximization.	22
2. Implications for a Maximizing Algorithm	23
LEAP's algorithm as a maximizer over prices.	25
3. The Total Surplus is Not Maximized by LEAP	32
D. EXISTENCE, UNIQUENESS, AND STABILITY OF EQUILIBRIA	37
E. MATHEMATICAL CHARACTERIZATION OF AN ALLOCATION PROCESS	43

PART II: ALGORITHMS FOR SOLVING THE EQUATIONS OF EQUILIBRIUM

A. THE SUCCESSIVE OVER-RELAXATION METHOD APPLIED IN LEAP	51
1. The Successive Over-relaxation Method (SOR)	51
2. Some Comments on the Convergence Properties of LEAP	52
Convergence of LEAP in an extreme case of the market share formula.	53

Lack of convergence of the LEAP algorithm for a simple example.	57
Lack of convergence of LEAP through market share.	59
3. The LEAP Algorithm Viewed as a Newton Method for Finding a Fixed Point	61
4. Computational Results in the Diamond Example	65
5. Oscillation--An Illustration	66
6. Computational Results on the Two Period Problem	71
Introduction.	71
Convergence criterion used with the algorithms.	71
Solutions for two typical cases.	72
Summary of numerical results.	72
Discussion.	76
B. VARIATIONS ON THE BASIC ALGORITHM	77
1. An Alternative to Averaging in Passing Prices Up	77
2. Passing Derivatives (Differences) Through Allocation Processes	81
Example.	82
Example.	86
C. NEW APPROACHES FOR SOLVING THE EQUILIBRIUM EQUATIONS	88
1. Newton's Method	89
2. Generalized Secant Method	95
3. Summary of Attempts to Accelerate Convergence to an Equilibrium Point	100
D. DECOMPOSITION OF THE ENERGY NETWORK	102
1. Decomposition and Variable Reduction in the Dynamic Equilibrium Case	108
2. An Interpretation of Decomposition	114
REFERENCES	119

LIST OF ILLUSTRATIONS

Figure

1.	The Diamond Example	2
2.	A Simple Allocation Process	3
3.	Market Share at an Allocation Node	4
4.	The Network Model	5
5.	The Diamond Example	8
6.	The Two Period Example	10
7.	Network Equilibrium	18
8.	Total Surplus Maximization	23
9.	Sequence of Points Generated by the Algorithm Used to Solve LEAP	24
10.	Maximization Over Price	25
11.	Problem	29
12.	Convergence Rate of the LEAP Algorithm with $\omega = .25$	29
13.	Path of Accelerated Newton Approach	31
14.	Example of Market Share	33
15.	Equilibrium Solution in Presence of Various Market Share Formulas	34
16.	Aggregation of Suppliers 1 and 2	35
17.	Aggregation of Networks	36
18.	An Allocation Process	43
19.	Aggregating Supply Curves	44
20.	Horizontal Addition to Devine R	47
21.	Γ	54
22.	Trajectories of q_1 and q_2 for the Example	56
23.	SOR Fails	57
24.	An Unstable Problem	60
25.	Convergence Characteristics	67
26.	Oscillation	69
27.	Oscillation if $\omega = \frac{1}{2}$	70
28.	An Allocation Node	77
29.	Supply/Demand Network	81
30.	An Allocation Process	83
31.	An Example	86
32.	Flow of Energy Network	90
33.	Network with Second Supplier Price Fixed	103
34.	Network with Demand Fixed	106
35.	Decomposition Example	115
36.	LEAP Estimates of the Implied Supply Curve	116

Exhibit

1.	Diamond Shaped Energy Network with Three Suppliers	91
2.	Encoding of Diamond Shaped Energy Model in Factorable Programming Language	92

Exhibit

3.	Solution of Diamond Shaped Example Using the Generalized Secant Method	97
4.	Rate of Convergence of LEAP on Diamond Shaped Equilibrium Problem	101

LIST OF TABLES

Table		
I	An Equilibrium Solution of the Diamond Example	9
II	An Equilibrium Solution of the Two Period Problem	12
III	Linear Rate of Convergence of LEAP's Algorithm	30
IV	(untitled)	55
V	(untitled)	55
VI	Number of Iterations to Converge for Different Relaxation Coefficients ω	66
VII	Convergence Characteristics	80
VIII	Comparison of Averaging Alternatives	79
IX	A Single Run on Example	86
X	Number of Iterations Required Until $\ (q_1, q_2) - q^*\ < 10^{-6}$	87
XI	Printout of Final Iteration of Newton's Method Applied to Diamond Shaped Energy Model ($a_{ij} = d[H_i(q) - q_i]/dq_j$) . . .	94
XII	Derivative Matrix for Equilibrium Equations ($a_{ij} =$ $d[H_i(q) - q_i]/dq_j$)	103
XIII	Derivative Matrix for Equilibrium Conditions ($a_{ij} =$ $d[H_i(q) - q_i]/dq_j$)	107
XIV	Matrix of Derivatives of Equilibrium Equations	109
XV	Derivatives of Equilibrium Equations ($a_{ij} = d[H_i(1) -$ $[Q_i(1) - Q_i(0)]]/dQ_j(2))$	113
XVI	Data for Figure 36	118

PART I: THE BASIC MODEL

A. DESCRIPTION

This section contains a concise description of the principal features of the "LEAP" model as described in the source documents [1] and [3]. The complete model contains a number of detailed features not reported herein.

Underlying the entire discussion is a directed, connected network whose nodes are partitioned into three nonempty, mutually exclusive sets. The first set consists of all source nodes, and these are associated with the "*supply* side" of the network. Such nodes are termed "Primary Resource Processes" in [3]. The second set of nodes in the network consists of intermediate nodes, associated with "Simple and Complex Conversion Processes," "Allocation Processes," and "Transportation Processes." The set of all sinks in the network constitutes the third or *demand* set, and these represent the "End-use Demand."

Figure 1 illustrates an example network with 12 nodes (processes) and 12 links. Nodes A, B, and C are supply nodes; nodes D, E, F, G, H, and I are intermediate nodes; and the demand set consists of nodes J, K, and L.

Each supply node i has associated with it a supply curve S_i which maps quantities of the commodities produced at i into (marginal) prices of these commodities. Throughout this report, we shall assume that each S_i is a monotone increasing, nonnegative function of a

Figure 1.--The diamond example.

single variable $q_i \geq 0$, although in general S_i is a monotonically increasing mapping, i.e.,

$$(S_i(q) - S_i(\bar{q}))(q - \bar{q}) \geq 0, \quad \text{for all } q, \bar{q} \geq 0,$$

and has the interpretation of being the subdifferential (see, e.g., [13]) of the optimal value function associated with a minimum cost problem faced by supplier i . See, e.g., Owen [10].

Each demand node j has associated with it a demand curve D_j which maps prices of the commodities demanded at j into quantities of these commodities. Throughout this report, we shall assume that each D_j is a monotone decreasing, nonnegative function of a single variable p_j , although in general D_j is a monotonically decreasing mapping, i.e.,

$$(D_j(p) - D_j(\bar{p}))(p - \bar{p}) \leq 0, \quad \text{for all } p, \bar{p} \geq 0,$$

and has the interpretation of being the subdifferential of a measure of "consumer satisfaction" that consumer j enjoys. See, e.g., Owen [10].

In this report, we focus attention on intermediate nodes associated with "Allocation Processes," of which there are two types, as illustrated in Figure 1. The simplest type of allocation node is represented by nodes E, G, and I of Figure 1, and is illustrated in Figure 2. At such a node,

Figure 2.--A simple allocation process.

conservation of flow must hold, and all prices are equal. Thus such a node represents a type of "distribution" center, which simply allocates q to its various customers, and does not distinguish between them in pricing.

The second, and more difficult type of allocation node is exemplified by nodes D, F, and H of Figure 1, and is illustrated in Figure 3.

Figure 3.--Market share at an allocation node.

Again, conservation of flow must hold. Traditionally, (e.g., [2], [11]) spatial models assume that a buyer confronted with n suppliers will purchase from the least expensive supplier. This implies that if two or more suppliers are required, the buyer will purchase in quantities so that the marginal price that he pays is identical over the active suppliers.

The LEAP model recognizes that this ideal situation is seldom realized in the real world, and thus assumes that the amount purchased is proportional to the inverse of the price, possibly raised to some power $\gamma \geq 1$ and possibly adjusted by a "price premium" k_i to account for differences not reflected in the price. Large values of γ produce simulated market behavior approximating the cheapest price decision rule. Figure 19 on page 44 illustrates an example of this approximation.

To exercise this (static) model, quantities q_i are set throughout the network. Prices $S_i(q_i)$ are then computed at the supply nodes, and sent *up* the network via the equations listed in Figures 2 and 3. With prices thus computed at the demand nodes, the functions D_j are then computed to estimate the quantities demanded there. These quantities are then passed *down* the network via the appropriate equations in Figures 2 and 3, with prices fixed as computed in the upward pass. In general, the (new) quantities computed on the downward pass will not equal the quantities placed on the network to initiate the upward pass.

Letting $F(S)$ represent the prices at the demand nodes as functions of the prices at the supply nodes, and letting $G(D)$ represent the quantities appearing at the supply nodes as functions of the quantities at the demand nodes, we can schematically represent the model as in Figure 4.

Figure 4.--The network model.

Letting $H(q) = G(D(F(S(q))))$, we say that the model is operating at an equilibrium point $\bar{q}$ if

$$\bar{q} = H(\bar{q}) .$$

In the dynamic case, there are n time periods so that the prices and quantities over the network become vectors, e.g., $p_j = (p_j(1), \dots, p_j(n))$, and $q_j(t)$ represents the flow on arc j in time period t . For each demand j , we assume given a vector-valued function $D_j(p_j) =$

$\{D_{j1}(p_j(1)), \dots, D_{jn}(p_j(n))\}$, where $D_{jt}(p_j(t))$ yields the quantity $q_j(t)$ demanded in time period t as a function of the price $p_j(t)$.

On the supply side, for each supplier i and each time period t , we assume given a function $m_i(Q_i(t))$ which yields the marginal cost of production in time period t as a function of the *cumulative* production up to and including time period t . Thus,

$$Q_i(t) = Q_i(t-1) + q_i(t) ,$$

where $Q_i(0)$ is a given constant. In order to determine the selling price $S_i(t)$, LEAP allows the producer to postpone production in anticipation of future prices. The relation becomes

$$S_i(n) = m_i(Q_i(n)) , \quad (1a)$$

and

$$S_i(t) = m_i(Q_i(t)) + \max_{\tau > t} \left\{ \frac{S_i(\tau) - m_i(Q_i(t))f(t, \tau)}{(1 + r_i)^{\tau-t}} \right\} , \quad (1b)$$

where $f(t, \tau)$ is a factor to account for inflation and technological change (in this report, we set $f(t, \tau) = (1 + \bar{r}_i)^{\tau-t}$ where $\bar{r}_i$ is the producer's estimate of the rate of inflation), and where r_i is a producer's discount factor for future prices in terms of present value.

Thus, the steps in computing the selling prices of the primary resources are:

Given $q_i(1), \dots, q_i(n)$, and $Q_i(0)$,

- (1) compute $Q_i(t) = Q_i(t-1) + q_i(t)$ in the order $t=1, 2, \dots, n$;
- (2) compute $m_i(Q_i(t))$ in any order, for $t=1, \dots, n$; and
- (3) compute $S_i(t)$ as above in the order $t=n, n-1, \dots, 1$.

In this way we define a vector $S_i(q_i)$ of the selling prices as a function of the quantities produced at i over the time horizon. Note that

the selling price $S_i(t)$ in time period t is a function of the selling prices in time periods $t+1, t+2, \dots$.

Contrast this last statement with the demand side, where $q_j(t)$ demanded in period t depends only on the price $p_j(t)$ in period t , and not on any anticipated future prices.

The function F defined previously for the static case to pass prices up can be carried over directly to the dynamic case by simply indexing the prices and quantities of Figures 2 and 3 by t . Likewise,

the fractions $\frac{\hat{q}_i}{q} = \left[\frac{(p_i + k_i)^{-\gamma}}{\sum_i (p_i + k_i)^{-\gamma}} \right]$ which are used by the

function $G(\cdot)$ to pass quantities down are also computed by replacing p_i by $p_i(t)$. In the dynamic case, however, the actual fraction $q_i(t)/q(t)$ used in time period t is an interpolation of the fraction $\hat{q}_i(t)/q_i(t)$ above, and the fraction $q_i(t-1)/q(t-1)$ used in the previous time period:

$$\frac{q_i(t)}{q(t)} = \beta_t \left(\frac{\hat{q}_i(t)}{q_i(t)} \right) + (1-\beta_t) \left(\frac{q_i(t-1)}{q(t-1)} \right), \quad (2)$$

where $\beta_t \in [0,1]$ is a "behavioral lag" coefficient. If $\beta_t = 0$, the fractional allocation $q_i(t)/q(t)$ remains unchanged from time period $t-1$. If $\beta_t = 1$, the market share formula in time period t alone is used to determine the fractional allocation. For $0 < \beta_t < 1$ the static response in time period t is tempered in the direction of the allocation determined in time period $t-1$, supposedly reflecting some reluctance to change.

Thus the steps in computing the quantities during the downward pass are:

Given $p_i(t)$ throughout the network,

- (1) compute the fraction $\hat{q}_i(t)/q(t)$ according to the static formula (see Figure 3) for each i , and t ;
- (2) compute the dynamic market shares from (1) above, in the order $1, 2, \dots, n$.

The following examples illustrate the model, and will be referred to later in this report.

1. The Diamond Example

In this example, we exhibit the static LEAP model for the network of Figure 1, here reproduced in Figure 5.

Figure 5.--The diamond example.

At nodes D, F, and H, we apply market share formulas of the form

$$q_i = \left[\frac{(p_i + i)^{-1}}{(p_i + i)^{-1} + (p_j + j)^{-1}} \right] q_k,$$

where links i and j merge to link k . An equilibrium solution is given in Table I.

TABLE I
AN EQUILIBRIUM SOLUTION
OF THE DIAMOND EXAMPLE

link i	q_i	p_i
1	0.7168	2.0478
2	1.1325	1.4526
3	0.9597	1.9210
4	0.4006	1.4526
5	0.7319	1.4526
6	1.1174	1.8344
7	1.6916	1.7183
8	0.5451	1.8344
9	0.5723	1.8344
10	1.0597	1.7810
11	0.4875	1.7183
12	1.2041	1.7183

It is easy to verify the following steps in the order indicated:

Given $q_1, \dots, q_{12}$, compute prices throughout the network on the "up-pass:"

With prices $p_1, \dots, p_{12}$ so determined, and using the market share formula at D, F, and H, compute new quantities throughout the network on the "down-pass:"

2. The Two Period Example

The network for this two-period LEAP model is shown in Figure 6. Starting at the bottom, the marginal cost functions are given by

$$m_1(Q_1(t)) = 1.1 + .55(Q_1(t))^2, \quad t=1,2,$$

$$m_2(Q_2(t)) = .88 + .44(Q_2(t))^{2.2}, \quad t=1,2,$$

with $Q_1(0) = Q_2(0) = 0$. Equations (1) become in this two-period case

$$p_i(2) = S_i(q,2) = m_i(Q_i(2)), \quad i=1,2,$$

$$p_i(1) = S_i(q,1) = m_i(Q_i(1)) + [S_i(2) - m_i(Q_i(1))(1+\bar{r}_i)] / (1+r_i), \quad i=1,2,$$

where $\bar{r}_i$ was chosen to be zero and r_i was set to $1/5.5 \approx .182$.

Figure 6.--The two-period example.

Then, since node C is an allocation node,

$$p_3(t) = F_3(q,t) = \frac{\sum_{i=1}^2 p_i(t)q_i(t)}{\sum_{i=1}^2 q_i(t)} ,$$

for $t=1,2$. This completes the set of equations used on the upward pass and serves to define the function $F(S)$.

In this example the demand functions are given by

$$q_3(1) = D_3(p,1) = 2.0 / \sqrt{p_3(1)} ,$$

$$q_3(2) = D_3(p,2) = 2.6 / \sqrt{p_3(2)} .$$

To continue the downward pass, the static market share allocation formula shown in Figure 3 is replaced by the alternate form

$$\hat{q}_i(t)/q(t) = f_i(p_i(t))^{-\gamma} / \sum_{i=1}^2 f_i(p_i(t))^{-\gamma} , \quad (3)$$

where $i=1,2$ here. The parameter γ has the same interpretation as before, and f_i is defined as the fractional market share at equal prices, so that $f_1 + f_2 = 1$; use of the f_i in Equation (3) replaces the earlier use of the price adjustments k_i . The dynamic market shares at node C are then given by Equation (2), and the downward pass is completed by calculating

$$q_i(t) = H_i(q,t) , \quad i=1,2 ; t=1,2 .$$

An equilibrium solution for the case $\gamma = 4$, $f_1 = .6$, and $\beta_1 = \beta_2 = .3$ is given in Table II.

TABLE II

AN EQUILIBRIUM SOLUTION OF THE TWO-PERIOD PROBLEM

link i	$q_i(1)$	$q_i(2)$	$p_i(1)$	$p_i(2)$
1	0.6703	0.8459	2.2079	2.3644
2	0.7133	0.8902	2.0291	2.1234
3	1.3836	1.7361	2.0895	2.2428

B. LEAP AND OTHER ECONOMIC MODELS

The LEAP model is a variant of the earlier "SRI-Gulf Energy Model" developed by Cazelet [3] and associates. It is one of a number of descriptive models designed to describe the flows of energy resources and their prices over the U.S. supply/demand network over time. Several of the alternative models, as well as the SRI-Gulf product, are described in a recent NSF report [2], which also contains some of the basic underlying economic theory.

In game theoretic terms, with I suppliers, J consumers, and n time periods, the model can be conceptualized as a sequential $(I+J)$ person noncooperative game. Postulating the existence of production cost functions c_i ($i=1, \dots, I$), and utility functions u_j ($j=1, \dots, J$), we could assume that each supplier desires to minimize his production costs, discounted over the time horizon, while each consumer seeks to maximize his utility function over each time period. Both suppliers and consumers must operate in accordance with the network constraints, including the market share and behavioral lag relations. Since the price functions S_i are only dependent on the basic resources q_i in LEAP, the game is noncooperative (otherwise, suppliers i and i' could enter into a coalition, which would be reflected in a price pair $(S_i(q_i, q_{i'}), S_{i'}(q_i, q_{i'}))$). The game is sequential in that the players make their decisions simultaneously at the beginning of each time period. Note that each supplier's

payoff function is computed in terms of future prices, while the consumers' functions are not. This appears to be a defect in the LEAP model, as a consumer's behavior surely depends, to some extent, on his expectation of future prices.

Cournot [5] was apparently the first to study the notion of an equilibrium point in a noncooperative situation. In 1838, he investigated the competition between two producers of the same commodity (the "duopoly" problem), each producer trying to maximize his payoff. Cournot proved, under mild assumptions on the payoff functions, that an equilibrium point *existed*, was *unique*, and was *stable*.

Cournot's work has been, and is currently being, extended to more realistic models. Friedman's recent book [7] contains a comprehensive and readable survey of economic models of oligopolies (several suppliers of differentiated goods, one consumer). Gabay and Moulin [8] investigate general noncooperative games, and establish existence, uniqueness, and stability under quite general assumptions. Unfortunately, even the mildest assumptions, which are standard in the economic literature, do not seem to apply to LEAP.

For example, consider an allocation node of the type depicted in Figure 3. Let $S_1(q_1) = \frac{1}{6} e^{3q_1}$ and $S_2(q_2) = 2e^{q_2/6}$. One might expect, that if $Q = q_1 + q_2$ were less than $\bar{Q} = \bar{q}_1 + \bar{q}_2$, the average price passed up by the model at Q would be less than the average price passed up at $\bar{Q}$, i.e.,

$$Q < \bar{Q} \implies F(Q) < F(\bar{Q}) .$$

However, let $q_1 = 2$, $q_2 = 1$ while $\bar{q}_1 = 1$ and $\bar{q}_2 = 3$. Then

$$F(Q) = 45.61 \quad \text{and} \quad F(\bar{Q}) = 3.31 .$$

To be sure, the pairs $q = (2,1)$ and $\bar{q} = (1,3)$ cannot be consistent with the market share formula imposed to pass quantities down for any $\gamma \geq 1$, but the above data are entirely in accord with the LEAP model assumptions. This particular example will be discussed further in Section E of this part.

The features which primarily distinguish LEAP from standard general economic equilibrium models are the use of the market share allocator and the imposition of the behavioral lag concept (see, e.g., [3]). Both of these concepts, designed to more accurately model observed market behavior in the aggregate, not only represent the primary departure of LEAP from the other models, but also introduce computational features (some undesirable) which are unique. This will be discussed in Part II of this report.

The imposition of the market share formula

$$q_i = \left[\frac{(p_i + k_i)^{-\gamma}}{\sum_i (p_i + k_i)^{-\gamma}} \right] q$$

at an allocation node of the type depicted in Figure 3 causes the demand q to be dispersed over the sources feeding the node. The constants k_i are "price premiums," and allow the buyer to differentiate products at the sources according to quality, etc. It should be noted that the k_i 's are actually dependent on the measure used to express price (a change of scale from dollars to, say, 10's of dollars would produce different results unless the k_i 's are changed accordingly). Product differentiation might better be handled by using *multiplicative* factors instead of additive, as in the two period example of Section A above.

Note that, with the above understanding of the k_i 's, the above relation is independent of the scale used to measure prices.

The factor γ is used to shape the market shares in the direction of the cheapest supplier rule, with higher γ values reflecting more sensitivity to prices. In the dynamic version, the behavioral lag coefficient β also tempers the market share according to time, by interpolating between the most recent market share and the one determined by the above formula. This particular linking causes difficulties in decomposition schemes (see Section D, Part II).

C. INTERPRETATION OF LEAP AS AN OPTIMIZATION MODEL

In this part an investigation is made into the question of whether or not the attaining of equilibrium in a general energy network is equivalent to solving a maximization problem. It is shown that if the consumer acts to maximize a cost equivalent utility function, and the supplier prices his supplies at the margin, then quantities at equilibrium also maximize "consumer surplus" plus supplier profit. The question of whether or not the algorithm used to solve LEAP can be interpreted as a maximization algorithm for this function is also taken up. It is shown that for a simple network, the algorithm can be interpreted as a descent method for minimizing (over prices) a function similar to that above. Finally, it is shown by example that because the LEAP model handles demand using market share formulas, attaining of equilibrium cannot be considered equivalent to maximization.

The implications for developing a faster algorithm from this theory are also considered.

1. Equilibrium as Maximization

The network discussed in this part is depicted in Figure 7. A vector of basic quantities q is produced by the supplier. There is a vector of supply prices $p = S(q)$ which results. These basic supplies are transformed by technological processes "up the network" into final goods which are demanded by the consumer. The prices for the final goods

Figure 7.--Network equilibrium.

are a function $F(p)$ of the basic supply prices. Based on a vector of final prices, the consumer buys, or demands D of the final products. Clearly, in the marketplace, quantities are produced, prices are set, and quantities are bought. A descriptive theory of what these quantities and prices are is that they are those for which the network is in equilibrium. Equilibrium is achieved when the consumer will buy the final goods supported by the basic supplies produced at the prices implied by the basic level of production. A final function $G(D)$ gives the basic supplies required to support the final demand D . Figure 7 gives a view of the network in equilibrium.

There is a considerable body of theory relating to the existence of an equilibrium for the network. A related matter is the development of an algorithm for finding the equilibrium prices and quantities. Another theoretical question is whether or not the equilibrium point can be considered to maximize some function. If it does, then algorithms for solving maximization problems might be applied to finding

an equilibrium. Furthermore, economic insight could be provided by examining the form of the function being maximized.

In order to develop a function being maximized it is necessary to make some assumptions about the way in which the supplier determines the price of his basic quantities produced, and how the consumer determines the final goods he will buy as a function of their prices.

Computation of $p(q)$.

Required to supply q , the supplier solves the problem

$$\min_z c(z) \quad \text{s.t.} \quad V(z) \geq q, \quad (4)$$

where z is the vector of activity levels and $V(z)$ the function transforming them into the basic supplies. The function $c(z)$ is the cost of utilization of the activity levels z . The function $c(z)$ includes nominal profit, but the true profit obtained is $p(q)^T q$ - (the minimum cost solution to the above problem). The vector of supply prices contains the dual variables (shadow prices) associated with the solution to Problem (4). If $z^*(q)$ is the solution to the problem with given q , then $c^*(q) = c[z^*(q)]$ is the optimal value function written as a function of the supplies required. It is well known that under conditions which usually hold, the dual variables are the derivatives of the optimal value function with respect to the right-hand side of the constraints of (4), i.e., with respect to q , therefore,

$$p(q) = \partial c^*(q) / \partial q. \quad (5)$$

Under assumptions which make intuitive economic sense, the optimal value function is a twice continuously differentiable convex function of q . Thus,

$$\partial p(q)/\partial q = \partial^2 c^*(q)/\partial q^2 \quad (6)$$

exists and is a positive (semi) definite matrix.

Computation of $D(F)$.

The consumer is presumed to have a utility function $\phi(D)$ associated with the owning of final goods. This is assumed to be comparable in units to the price he pays for owning them. The total value he associates with owning a vector D of goods is equal to his utility function minus the total cost for purchasing them. Thus,

$$\phi(D) - F^T D$$

is the value associated with purchasing the vector D of goods, given the prices F .

Being rational, the consumer is presumed to purchase so as to maximize this function. For given prices F , $D(F)$ solves

$$\underset{D}{\text{maximize}} \quad \phi(D) - F^T D \quad (7)$$

Since the solution is unconstrained, $D(F)$ satisfies

$$\partial \phi[D(F)]/\partial D - F = 0 \quad (8)$$

The utility function $\phi(D)$ is usually considered to be a twice continuously differentiable concave function of the final goods purchased.

Thus,

$$\partial^2 \phi[D(F)]/\partial D^2$$

exists and is negative (semi) definite.

Assumption of relationship between cost of final goods, and cost of basic supplies.

Clearly there is a relationship in an economic network between the cost of the basic supplies, and the cost of the final goods. Without

some explicit assumption about this relationship, it is not possible to analyze the network equilibrium in terms of a maximization problem.

As before, let D be the final goods purchased. Then $G(D)$ is the vector of basic supplies required to support D . Let p be the vector of basic supply prices, and $F(p)$ the vector of final good prices.

The total cost passed up from the supplier is $p^T G(D)$. It is now assumed that there is a constant M , a multiplicative factor and a constant K such that the final cost to the consumer is related to the basic supplier cost by

$$F(p)^T D = M[p^T G(D)] + K. \quad (9)$$

Equivalent consumer maximization problems.

One final assumption required is the existence of an inverse function for G . Thus, knowing q , the basic quantities supplied, it is possible to find $G^{-1}(q)$, the vector of final goods which can be produced by these supplies. This assumption is necessary in that the consumer must be made to view his consumption maximization from the point of view of the basic supplies (and their prices) required to deliver his final goods.

Under these assumptions the following three consumer optimization problems are equivalent: -

$$(A) \max_D \phi(D) - F(p)^T D$$

$$(B) \max_q \phi[G^{-1}(q)] - F(p)^T G^{-1}(q)$$

$$(C) \max_q \phi[G^{-1}(q)] - (p^T q)M - K.$$

Thus if $D[F(p)]$ solves (A), $G[D[F(p)]]$ solves (B) and (C). If $q^*(p)$ solves (C), $G^{-1}[q^*(p)]$ solves (A), and $q^*(p)$ solves (B). The corresponding conclusions hold for a solution of (B).

Equivalence of equilibrium and maximization.

Let $\bar{q}$ be a vector of basic supplies for which the network is in equilibrium. Then

$$\bar{q} = G[D[F[p(\bar{q})]]] .$$

Taking the inverse function,

$$G^{-1}(\bar{q}) = D[F[p(\bar{q})]] .$$

Since $D[F[p(\bar{q})]]$ solves (A), it must be equal to $G^{-1}[q^*[p(\bar{q})]]$, where this solves (C). Thus,

$$G^{-1}(\bar{q}) = G^{-1}[q^*[p(\bar{q})]] ,$$

and

$$\bar{q} = q^*[p(\bar{q})] .$$

Then, because $\bar{q}$ solves (C), it follows from the first order necessary conditions that

$$\partial\phi[G^{-1}(\bar{q})]/\partial q = p(\bar{q})M$$

or

$$\partial\phi[G^{-1}(\bar{q})]/\partial q - p(\bar{q})M = 0 .$$

Because of (5) it follows that $\bar{q}$ is a zero of the derivative of the function

$$\phi[G^{-1}(q)] - c^*(q)M . \quad (10)$$

Because of the arguments presented earlier, $\bar{q}$ is a maximizer of that function.

The economic interpretation of this function derives from the fact that (10) is equal to (using the assumption (9))

$$\underbrace{\phi[G^{-1}(q)] - F[p(q)]^T G^{-1}(q) + K}_{\text{consumer surplus}} + \underbrace{[p(q)^T q - c^*(q)]M}_{\text{supplier profit (weighted)}}. \quad (11)$$

Figure 8 shows a geometric interpretation of this result.

Figure 8.--Total surplus maximization.

For this simple classical case, the quantities supplied are those demanded (i.e., $G^{-1}(q) = q$, $M = 1$, $K = 0$, $F(p) = p$.) Maximization over q is to be done for $\int_0^q \{\phi(t)/\partial t - \partial c^*(t)/\partial t\} dt$. The quantity q at the equilibrium point $(\bar{q}, \bar{p})$ is the value for which the area is being maximized (this is the shaded area).

2. Implications for a Maximizing Algorithm

The minimum requirement for an algorithm based directly on the maximization of (11) is that the derivative of the function be computable. This means, given a value q_0 , $\partial\phi(q_0)/\partial q - \partial c^*(q_0)/\partial q$ must be available. The second term is the negative of the vector of shadow prices of the optimization problem, and these prices are available in most network models.

That is, it is standard for a model to be able to give supply prices as a function of quantity produced.

It is the *first* term of the derivative which is *not available*. That is, no known network model will produce the demand prices given the final goods purchased. The standard information available is to compute demand for final goods given the prices for final goods. Thus problem (7) is solved (or its solution is available). In principle, then, it seems fruitless to try to develop an algorithm for solving the network equilibrium problem which tries to directly maximize the function (7).

The algorithm used to solve LEAP proceeds in the only way it can, as illustrated in Figure 9.

Figure 9.--Sequence of points generated by the algorithm used to solve LEAP.

If one *could* proceed from point A to point D, then an algorithm for finding the maximum area (shaded in Figure 8) could be devised. However, LEAP's algorithm moves as indicated in the figure, and then computes the new starting point q_1 as a convex combination of q_0 and C. The possibility still exists that its algorithm can be considered a maximizing

procedure, but of the function whose maximum area is shaded in Figure 9.

Obviously this area is maximized over p at $\bar{p}$.

LEAP's algorithm as a maximizer over prices.

Consider Figure 10.

Figure 10.--Maximization over prices.

Supply and demand curves are now written as functions of price. Thus given prices p , $q_S(p)$ will be supplied, and $q_D(p)$ will be demanded. The shaded area is equal to (where C is a constant equal to the entire area under the consumer's utility function)

$$\psi(p) = C - \{ \phi[q_D(p)] - p^T q_D(p) + p^T q_S(p) - c^*[q_S(p)] \} .$$

The derivative of this function (with respect to p) is

$$\begin{aligned} \partial\psi(p)/\partial p &= [p - \partial\phi[q_D(p)]/\partial q][\partial q_D(p)/\partial p] + q_D(p) \\ &\quad + [\partial c^*[q_S(p)]/\partial q_S][\partial q_S(p)/\partial p] - q_S(p) . \end{aligned}$$

Using (5) and (8), this results in

$$\partial\psi(p)/\partial p = q_D(p) - q_S(p) .$$

It is easy to see then that a maximizing algorithm could be constructed if only $q_S(p)$ were known. The information used in the

algorithm is not available in this form. The way it operates is to give prices as a function of quantities supplied, i.e., $p_S(q)$. It turns out, however, that the algorithm does have an interpretation as an ascent method for going from a set of prices p_0 to an improved set of prices p_1 by moving in a direction which points in a direction of increase of the function $\psi(p)$.

The algorithm proceeds as follows. (Refer to Figure 10 for a one-dimensional interpretation.) It starts with quantities q_0 . This implies a vector of prices $p_S(q_0) = p_0$. The demand function gives $q_D[p_S(q_0)]$. A convex combination of q_0 and $q_D[p_S(q_0)]$ is computed to give q_1 . Then the supply function gives $p_1 = p_S(q_1)$. The obtaining of p_1 from p_0 by this process will now be shown to be an ascent algorithm for maximizing $\psi(p)$. [Note $q_0 = q_S(p_0)$.] Now [here ω is the parameter ($0 < \omega < 1$) determining the convex combination],

$$\begin{aligned} p_1 &= p_S(q_1) = p_S\{q_0(1-\omega) + q_D[p_S(q_0)]\omega\} \\ &= p_S\{q_0 + [q_D[p_S(q_0)] - q_0]\omega\} \\ &= p_0 + \left[\int_0^1 \partial p_S[q(t)] / \partial q dt \right] [q_D(p_0) - q_S(p_0)]\omega, \end{aligned} \quad (12)$$

where

$$q(t) = q_0(1-t) + \{q_D[p_S(q_0)] - q_0\}\omega t.$$

[This used the fact that $p_S(q_0) = p_0$ twice.]

From (6) it follows that $\partial p_S(q(t)) / \partial q$ is a symmetric positive definite matrix for every t . Using the fact that the integral of a

positive definite matrix is positive definite, it follows that p_1 is obtained from p_0 by moving along an ascent direction. It is an ascent direction since

$$\begin{aligned} & \left. \frac{d\psi[p_0 + A[q_D(p_0) - q_S(p_0)]t]}{dt} \right|_{t=0} \\ &= [q_D(p_0) - q_S(p_0)]^T A [q_D(p_0) - q_S(p_0)] > 0, \end{aligned}$$

where A is the integral matrix in (12) above. Of course, this assumes $p_0 \neq \bar{p}$. If equality holds, the function is already maximized and no movement is possible.

It is well known that any form of steepest ascent can be expected to exhibit a linear rate of convergence unless the metric mapping the gradient converges to the inverse Hessian matrix of the function being maximized. The mapping shown for the LEAP model does not have this property and the computer tests (see Section A, Part II of this report) indicate that the convergence is linear (indeed, if convergence occurs at all). Modern optimization algorithms for maximizing an unconstrained function (or solving a set of nonlinear equations) are highly developed for accelerating the rate of convergence of algorithms by using information from previous iterations to approximate the inverse Hessian of the function being maximized. Next is a proposal for speeding up the algorithm by modifying it using derivative information available implicitly to update the metric which maps the supply prices.

The basic idea is this. Any time, for some p and q , when the values where $p_S(q) = p$ and the vector $q_D[p_S(q)]$ are available, the vector

$$q_D[p_S(q)] - q$$

is the derivative of the function $\psi(p)$. When this information is available at two sets of prices, then enough information is known to approximate a portion of the inverse Hessian of $\psi(p)$. This gives rise to either some quasi-Newton algorithm (updating formulas for which are legion) or to the generalized secant method. Further investigations will be needed to see which of these methods would be more suitable. (Conjugate direction methods are also a possibility, but the necessity of using an optimal step-size procedure in conjunction with these seems to limit their applicability.)

For the one variable case (an example of which is analyzed in Figure 11) the algorithm is as follows. Start with some q_0 ; do as the usual procedure does now. That is, compute $F(q_0) = q_D[p_S(q_0)]$. Interpolate for the point $q_1 = q_0(1-\omega) + H(q_0)\omega$. Go up and down once more, i.e., compute $H(q_1)$. Call $p_S(q_k)$, p_k . Then the Newton approximation takes the form

$$p'_{EST} = p_1 - \hat{\psi}''(p_1)^{-1}[H(q_1)-q_1] ,$$

where $\hat{\psi}''(p_1)$, the approximation to the second derivative of $\psi(p)$, is given by

$$\hat{\psi}''(p_1) = \left[(H(q_1)-q_1) - (H(q_0)-q_0) \right] / (p_1-p_0) .$$

This algorithm will be tried on the following problem in one variable. Given the requirement that q be supplied, the supplier's price is

$$p_S(q) = q^2 .$$

For the price p , the consumer will demand

$$q_D(p) = e^{-p} .$$

Figure 11.--Problem.

This is plotted in Figure 11. Equilibrium is at $(\bar{q}, \bar{p}) = (.65292, .42630)$. The algorithm used to solve LEAP gives a rate of convergence which is linear for this problem as indicated in Figure 12 and Table III. A value of $\omega = .25$ was used.

Figure 12.--Convergence rate of the LEAP algorithm with $\omega = .25$.

TABLE III
 LINEAR RATE OF CONVERGENCE OF LEAP'S ALGORITHM
 (HERE $\omega = .25$)

k	q_k	$p_S(q_k)$	$q_D[p_S(q_k)]$	$q_k - \bar{q}$	$\frac{q_k - \bar{q}}{q_{k-1} - \bar{q}}$
0	.5	.25	.77880	-.152917	--
1	.56970	.32456	.72285	-.083217	.544
2	.607996	.36965	.69098	-.044921	.540
3	.62873	.39531	.67347	-.024187	.538
4	.63992	.40949	.66399	-.012997	.537
5	.64594	.41724	.65887	-.00698	.537

The new accelerated technique proceeds as indicated below and is pictorially described in Figure 13.

The starting point is $q_0 = .5$. Then

k =	q_k	$p_S(q_k)$	$q_D[p_S(q_k)]$	$H(q_k) - q_k$
0	.5	.25	.77880	.27880
	(interpolated)			
1	.56970	.32456	.72205	.15315

Therefore,

$$\hat{\psi}'(\bar{p}) \doteq (.27880 - .15315) / (.25 - .32456) = -1.6853.$$

Newton's method gives the first estimate of $\bar{p}$ as

$$p_{EST}^1 = .32456 - (-1.6853)^{-1}(.15315) = .41543.$$

Proceeding from this,

Figure 13. Path of accelerated Newton approach.

$$\begin{array}{rcccc} 2 & -- & .41543 & .66006 & \\ 3 & .66006 & .43568 & .64683 & -.01323 \end{array}$$

Using values from iteration 1, the new estimate of $\hat{\psi}'(\bar{p})$ is

$$-1.4973 = [(-.01323) - (.15315)] / (.43568 - .32456) .$$

Newton's method gives the next estimate of $\bar{p}$ as

$$p_{EST}^2 = .43568 - (-1.4973)^{-1}(-.01323) = .42684 ,$$

which is very close to the true value of $\bar{p} = .42630$.

3. The Total Surplus is Not Maximized by LEAP

It has been suggested that the "total surplus" is being maximized by LEAP. Consider the example depicted in Figure 14. The total surplus is given by

$$\begin{aligned} & \left[\int_0^{q_3} p_3(q_3) dq_3 - p_3(q_3) dq_3 \right] \\ & + \left[p_1(q_1)q_1 - \int_0^{q_1} p_1(q_1) dq_1 \right] + \left[p_2(q_2)q_2 - \int_0^{q_2} p_2(q_2) dq_2 \right] , \end{aligned}$$

and represents the sum of the producers (at nodes A and B) and the consumer's (at node C) surpluses.

If we maximize this subject to the conservation of flow constraint:

$$q_1 + q_2 = q_3$$

and the conservation of value constraint:

$$p_1(q_1)q_1 + p_2(q_2)q_2 = p_3(q_3)q_3 ,$$

we obtain, upon differentiating the Lagrangian:

$$(\pi-1)p'_i(q_i)q_i + \pi p_i(q_i) = \rho ,$$

where ρ and π are the multipliers associated with the above two constraints. Note that both of these constraints are imposed by the LEAP model.

Figure 14.--Example of market share.

Multiplying the last relation by q_i , and using the imposed constraints to simplify, we obtain

$$(\pi-1)(p'_3q_3^2 - p'_1q_1^2 - p'_2q_2^2) = 0 .$$

But $p'_3 < 0$ while $p'_1, p'_2 > 0$. It follows that $\pi = 1$. Therefore,

$$p_i(q_i) = \rho ,$$

i.e., the prices are all equal at optimality.

This latter result is not true in the presence of a market share allocation with finite exponent. Let $p_1 = 1+3q_1$, $p_2 = 2+2q_2$, and $p_3 = 3-q_3$ in the example under discussion. Figure 15 illustrates the set of all points (q_1, q_2) satisfying the two conservation equations.

Figure 15.--Equilibrium solution in presence of various market share formulas.

Also depicted in this figure are the equilibrium solutions obtained by the LEAP algorithm, for various values of the exponent γ in the equations

$$q_1 = q_3 \left(\frac{\left(\frac{1}{p_1}\right)^\gamma}{\left(\frac{1}{p_2}\right)^\gamma + \left(\frac{1}{p_3}\right)^\gamma} \right)$$

$$q_2 = q_3 - q_1 .$$

However, if suppliers 1 and 2 are lumped together as a *single* supplier 12, whose supply curve $p = p_{12}(q_3)$ can be generated as is shown in Figure 16 (for $\gamma=1$), we can now interpret the problem as

$$\begin{aligned} &\text{maximize} \quad (\text{Consumer surplus of consumer 3}) + \\ &\quad q_3 \quad (\text{Producer surplus of [implied] producer 12}) , \end{aligned}$$

as in the traditional interpretation of a single supplier, single consumer problem.

The implied supply curve $p_{12}(q)$ was generated point by point by setting

Figure 16.--Aggregation of suppliers 1 and 2.

$$p_{12}(q) = \frac{p_1(q_1)q_1 + p_2(q_2)q_2}{q},$$

where q_1 and q_2 satisfy the equations

$$q_1 = q \left(\frac{\left(\frac{1}{p_1}\right)}{\left(\frac{1}{p_1}\right) + \left(\frac{1}{p_2}\right)} \right)$$

$$q_2 = q \left(\frac{\left(\frac{1}{p_2}\right)}{\left(\frac{1}{p_1}\right) + \left(\frac{1}{p_2}\right)} \right),$$

with the p_i 's previously set as functions of q_1 and q_2 , and $q_1 + q_2 = q$.

Section D of this part contains a detailed characterization of the implied supply curve. Generally several suppliers can be aggregated into a single supplier serving a common distribution point. Likewise, several consumers can be aggregated into a single consumer if they are dealing with a common supplier. In this fashion, certain types of networks can be reduced in size by aggregation, as is indicated in Figure 17, and this is the essence of "decomposition," discussed in Section D of Part II.

Figure 17.--Aggregation of network.

D. EXISTENCE, UNIQUENESS, AND STABILITY OF EQUILIBRIA

In this section, we will discuss the existence, uniqueness, and stability of equilibria points of general economic models. We will limit attention to the static case, although the arguments generally apply to the dynamic case.

The assumptions made in this section are as mild as we could find in the general economic literature. While continuity is assumed herein, differentiability is not.

Theorem 1: Assume the composite map $F(S(q)) : R^I \rightarrow R^J$ is continuous and isotone increasing, and the map $G(D(p)) : R^J \rightarrow R^I$ is continuous and isotone decreasing.

Then there is a pair $(\bar{p}, \bar{q})$ such that

$$F(S(\bar{q})) = \bar{p}$$

$$G(D(\bar{p})) = \bar{q}$$

Proof: It suffices to show that the map

$$H(q) = G(D(F(S(q))))$$

has a fixed point. Now

$$\begin{aligned} 0 \leq q &\implies F(S(0)) \leq F(S(q)) \\ &\implies G(D(F(S(0)))) \geq G(D(F(S(q)))) \end{aligned}$$

so that H maps $\{q : q \geq 0\}$ into the hyperrectangle $\{q : 0 \leq q \leq H(0)\}$.

But H is continuous, so that H has a fixed point.

The isotonicity assumption on $F(S(\cdot))$ is simply a mathematical expression for the fact that prices will uniformly increase at the top of the network if the amounts supplied at the bottom uniformly increase. If prices at the top uniformly increase, one expects that amounts demanded of the supply side will uniformly decrease, hence $G(D(\cdot))$ is assumed isotone decreasing.

For the static case, an equivalent definition of an isotone increasing function is:

$$q_i \leq \bar{q}_i$$

$$F(S(\dots, q_{i-1}, q_i, q_{i+1}, \dots)) \leq F(S(\dots, q_{i-1}, \bar{q}_i, q_{i+1}, \dots))$$

for $i=1, \dots, I$. In other words, prices at the demand side cannot decrease if all quantities save one remain stationary, and the one quantity increases on the supply side. An isotone decreasing function has a similar equivalent characterization.

Here we wish to make two points:

- (1) The above isotonicity assumptions are mild when compared to standard assumptions made in economic theory. Indeed, Friedman [7] lists a number of standard common assumptions imposed on economic models, and isotonicity is perhaps the most basic of all.
- (2) Isotonicity is *not* satisfied by the LEAP model. This point will be verified by characterizing the sequence of points $q^i = H(q^{i-1})$, $q^0 = 0$ under our assumptions, and pointing out in Part II that our computational experience indicates that LEAP does not generate such sequences.

Theorem 2: If H is isotone decreasing, $q^0 = 0$, and if $q^i = H(q^{i-1})$, then

- (a) $\{q^{2k}\}_{k=0}^{\infty}$ is a monotonically nondecreasing sequence, converging to a point q^{ℓ} ;
- (b) $\{q^{2k+1}\}_0^{\infty}$ is a monotonically nonincreasing sequence, converging to a point q^u ; and
- (c) any fixed point of H lies in $[q^{\ell}, q^u]$.

Proof:

$$\begin{aligned} 0 &\leq q^1 \\ q^1 &\geq q^2 \\ q^2 &\leq q^3, \text{ etc.} \end{aligned}$$

so that $q^{2k} \leq q^{2k+1}$, for all $k=0,1,\dots$.

Also

$$\begin{aligned} 0 &\leq q^2 \\ q^1 &\geq q^3 \\ q^2 &\leq q^4, \text{ etc.} \end{aligned}$$

so that

$$q^{2k+1} \leq q^{2k-1}$$

and

$$q^{2k} \leq q^{2k+1}, \text{ for all } k.$$

Hence $\{q^{2k}\}$ converges monotonically up to a point q^{ℓ} , and $\{q^{2k+1}\}$ converges monotonically down to a point q^u . Clearly,

$$q^{\ell} \leq q^u.$$

To prove part (c), let $\bar{q}$ be any point which is greater than or equal to zero. Then

$$\begin{aligned} 0 &\leq \bar{q} \\ q^1 &\geq \bar{q}^1 \\ q^2 &\leq \bar{q}^2, \text{ etc.} \end{aligned}$$

so that

$$q^{2k} \leq \bar{q}^{2k}$$

and

$$\bar{q}^{2k+1} \leq q^{2k+1}.$$

Therefore, the limit points of $\{\bar{q}^k\}$ lie in $[q^l, q^u]$. In particular, if $\bar{q}$ is a fixed point, then $\bar{q} = \bar{q}^k$ for all k , so that

$$\bar{q} \in [q^l, q^u].$$

Corollary: H maps $[q^l, q^u]$ into itself, and $H(q^u) = q^l$, $H(q^l) = q^u$.

Proof: Fix q so that $q^l \leq q \leq q^u$. Then

$$H(q^u) \leq H(q) \leq H(q^l).$$

But

$$q^{2k} \rightarrow q^l,$$

so

$$H(q^{2k}) \rightarrow H(q^l),$$

i.e.,

$$q^{2k+1} \rightarrow H(q^l),$$

i.e.,

$$q^u = H(q^l).$$

Likewise,

$$q^l = H(q^u),$$

so, from above,

$$q^l \leq H(q) \leq q^u,$$

i.e.,

$$H(q) \in [q^l, q^u],$$

and we are done.

Somewhat stronger assumptions would guarantee uniqueness in the following manner for the static case.

Theorem 3: Assume $H(q)$ is a monotone decreasing mapping, i.e.,

$$(H(q) - H(\bar{q}))(q - \bar{q}) \leq 0$$

for $q, \bar{q} \geq 0$. Then there is a unique equilibrium point q^* .

Proof: If there were two equilibrium points q^1 and q^2 , we would have

$$(H(q^1) - H(q^2))(q^1 - q^2) \leq 0 ,$$

i.e., $(q^1 - q^2)(q^1 - q^2) \leq 0 ,$

i.e., $||q^1 - q^2||^2 \leq 0 .$

Monotone mappings are discussed in some detail in Rockafellar [13]. Note all monotone decreasing maps are isotonically decreasing, but the opposite is not true, e.g.,

$$H_1(q) = 1 - q_2^2$$

$$H_2(q) = 1 - q_1^2 .$$

While the assumption of monotonicity is more severe than isotonicity, it is, nevertheless, a common assumption in economic theory (e.g., compare assumptions A4 and A7 of page 294 in Friedman [7] to monotonicity). It is equivalent to the negative definiteness of ∇H if the latter exists.

Finally we remark that LEAP does not even exhibit isotonicity. Convergence characteristics of LEAP (as in Part II) do not exhibit the regular behavior predicted by Theorem 2 of this section.

The following example illustrates the possible instability of LEAP.

Example. See Figure 3, Section A, Part I.

$$S_1(q_1) = -2 + q_1$$

$$S_2(q_2) = -6 + 4q_2$$

$$D_3(p_3) = 31.7 - 9.1p_3$$

$$\gamma = 1 .$$

There is an equilibrium point at

$$q^* \approx (4.181118928, 2.435935767) .$$

Applying the H map to q^* will yield a point near q^* , but successive applications of H quickly diverge from q^* .

The instability becomes even more dramatic if the market share parameter γ is increased.

E. MATHEMATICAL CHARACTERIZATION OF
AN ALLOCATION PROCESS

In this section, we analyze an allocation process of the type depicted in Figure 18. We assume given supply curves $p_i(q_i)$ at nodes 1 through k . These curves may or may not correspond to primary resource processes, but here we will assume that each p_i is a strictly increasing function of the single variable q_i .

Following Brock and Nesbitt [2], in their preliminary description of the SRI-Gulf model, the supply curves can be aggregated at the node A by simple horizontal addition, when this results from the policy: buy *all* from the supplier offering the cheapest price. Figure 19 illustrates the curve so obtained in an example.

Figure 18.--An allocation process.

We now wish to extend their interpretation to the case where the allocation rule:

"buy all from the cheapest supplier"

Figure 19.--Aggregating supply curves.

is changed to:

"buy according to the market share formula

$$q_i = \left[\frac{(p_i(q_i) + k_i)^{-\gamma}}{\sum_i (p_i(q_i) + k_i)^{-\gamma}} \right] Q ."$$

For notational convenience, set

$$P_i(q_i) = (p_i(q_i) + k_i) .$$

Given a specific value of Q , we will first show that there is a unique value $P(Q)$ such that

$$(a) \quad P(Q) = \frac{\sum_{i=1}^k P_i(q_i) q_i}{Q}$$

$$(b) \quad q_i = Q \left[\frac{P_i(q_i)^{-\gamma}}{\sum_{i=1}^k P_i(q_i)^{-\gamma}} \right]$$

i.e., $P(Q)$ is the average of the prices of the quantities collected at the node, and the market share formula is satisfied.

Theorem 4: Given functions $p_i(q_i)$ strictly increasing over $\{q_i \geq 0\}$ ($i=1, \dots, k$), for any given $Q > 0$ there is a unique set of q_i 's for which (b) above will hold.

Proof: The existence of a set of q_i 's follows from the fact that (b) defines a continuous map of $\{q : \sum q_i = Q\}$ into itself. Assume there are two points $q^1 \neq q^2$ such that

$$\sum_{i=1}^k q_i^1 = \sum_{i=1}^k q_i^2 = Q ,$$

and which satisfy (b) above. Then

$$q_I^1 < q_I^2, \text{ for some } I, \text{ while}$$

$$q_J^1 > q_J^2, \text{ for some } J.$$

But both q^1 and q^2 satisfy

$$q_i P_i(q_i)^\gamma = \frac{Q}{\sum_{i=1}^k P_i(q_i)^{-\gamma}} \quad (13)$$

for any i (from (b)), so that

$$q_I P_I(q_I)^\gamma = q_J P_J(q_J)^\gamma.$$

Now both $P_I(\cdot)$ and $P_J(\cdot)$ are strictly increasing, so

$$q_I^1 P_I(q_I^1) > q_I^2 P_I(q_I^2),$$

i.e.,

$$q_J^1 P_J(q_J^1) < q_J^2 P_J(q_J^2).$$

But this last inequality cannot hold as $q_J^1 > q_J^2$.

Corollary. Under the above hypothesis, there is a unique value $P(Q)$ satisfying (a) and (b) above.

Indeed, for a given Q , one must solve the equations

$$q_i P_i(q_i)^\gamma = \frac{Q}{\sum_{i=1}^k P_i(q_i)^{-\gamma}}, \quad (i=1, \dots, k)$$

to obtain the proper q_i values, then use the average

$$P(Q) = \frac{\sum_{i=1}^k P_i(q_i) q_i}{Q}$$

to obtain P .

Geometrically, Figure 20 illustrates the steps involved in this operation. Let

$$R_i(q_i) \doteq q_i P_i(q_i)^Y .$$

Then each $R_i(\cdot)$ is strictly monotone increasing and the function $R(\cdot)$ obtained by adding the R_i 's horizontally is also strictly monotone increasing.

Figure 20.--Horizontal addition to define R .

By Equation (13) above, we must know $R_i(q_i) = R_j(q_j)$ for all i and j . For any given Q , one reads the value $R(Q)$ from the graph, thus establishing the common values of the R_i 's. The functions R_i are then inverted to obtain the q_i 's, which sum to q , and satisfy the market share formula.

Finally, $P(Q)$ is obtained by averaging the prices $p_i(q_i)$ obtained from the values of q_i now established.

One would expect that the implied supply curve is always an increasing function of Q . This is clearly the case when the cheapest supplier supplies all.

We now show that this is also true when the market share allocator with $\gamma = 1$ is imposed on differentiable, strictly increasing supply curves.

Theorem 5: Assume $p_i(q_i)$ is continuously differentiable over $q_i \geq 0$, and that $p'_i(q_i) > 0$. Then $P(Q)$ is monotone nondecreasing when $\gamma = 1$.

Proof: It suffices to show that $P'(Q) \geq 0$. Now

$$P'(Q) = \frac{Q \sum_i \left(\left(\frac{dp_i}{dq_i} \right) q_i + p_i \right) \frac{dq_i}{dr} \frac{dr}{dQ} - \sum_i p_i q_i}{Q^2}$$

where $(p_i(q_i) + k_i)q_i = (p_j(q_j) + k_j)q_j = r = \frac{Q}{\sum_i \left(\frac{1}{p_i(q_i)} \right)}$ for all i and j .

Thus we must show that

$$Q \sum_i \left(\left(\frac{dp_i}{dq_i} \right) q_i + p_i \right) \frac{dq_i}{dr} \frac{dr}{dQ} \geq \sum_i p_i q_i. \quad (14)$$

But $p_i(q_i)q_i = (p_i(q_i) + k_i)q_i = r$ implies that

$$\left(\left(\frac{dp_i}{dq_i} \right) q_i + p_i(q_i) \right) \frac{dq_i}{dr} = 1,$$

and $Q = r \sum_j \left(\frac{1}{p_j(q_j)} \right)$ so that (14) becomes

$$r \left[\sum_j \left(\frac{1}{p_j(q_j)} \right) \right] \left[\sum_i \frac{\left(\left(\frac{dp_i}{dq_i} \right) q_i + p_i \right) \frac{dq_i}{dr}}{\left(\left(\frac{dp_i}{dq_i} \right) q_i + p_i \right)} \frac{dr}{dQ} \right] \geq r \sum_i \left(\frac{p_i}{p_i} \right). \quad (15)$$

Finally, $\sum_i q_i(r) = Q$ implies that

$$\sum_i \frac{dq_i}{dr} = \frac{dQ}{dr},$$

i.e.,

$$\sum_i \left(\frac{1}{\left(\frac{dp_i}{dq_i} \right) q_i + p_i(q_i)} \right) = \frac{dQ}{dr} ,$$

so that (15) becomes

$$\left[\sum_j \left(\frac{1}{p_j} \right) \right] \frac{\sum_i \left(\frac{\left(\frac{dp_i}{dq_i} \right) q_i + p_i}{\left(\frac{dp_i}{dq_i} \right) q_i + p_i} \right)}{\sum_i \left(\frac{1}{\left(\frac{dp_i}{dq_i} \right) q_i + p_i} \right)} \geq \sum_i \left(\frac{p_i}{p_i} \right) ,$$

i.e., we must show

$$\left[\sum_j \left(\frac{1}{p_j} \right) \right] \left[\sum_i \frac{\left(\left(\frac{dp_i}{dq_i} \right) q_i + p_i \right)}{\left(\left(\frac{dp_i}{dq_i} \right) q_i + p_i \right)} \right] \geq \left[\sum_i \left(\frac{p_i}{p_i} \right) \right] \left[\sum_j \left(\frac{1}{\left(\frac{dp_j}{dq_j} \right) q_j + p_j} \right) \right] .$$

There are four terms in this expression, which we refer to counting from the left. The first term clearly exceeds the last, since $\frac{dp_i}{dq_j} q_j > 0$. We will now show that the second expression exceeds the third, by showing that each term in the second sum exceeds each term in the third sum. This is true if

$$\left(\frac{dp_i}{dq_i} q_i + p_i \right) p_i \geq \left(\frac{dp_i}{dq_i} q_i + p_i \right) p_i ,$$

i.e., if

$$\frac{dp_i}{dq_i} q_i p_i \geq \frac{dp_i}{dq_i} q_i p_i ,$$

which is true, because $p_i = p_i + k_i > p_i$, and $\frac{dp_i}{dq_i} q_i > 0$.

Thus we have established that $P(Q)$ is a monotone increasing function when $\gamma = 1$. Since it is also monotone in the extreme $\gamma = \infty$, it suggests that the property is true for general γ , but we have been unable to verify this. The above proof does not seem to carry through.

The previous example is used to illustrate the implied supply curves when $\gamma = 1$ and $\gamma = 5$ (see Figure 19).

PART II: ALGORITHMS FOR SOLVING THE EQUATIONS OF EQUILIBRIUM

A. THE SUCCESSIVE OVER-RELAXATION

METHOD APPLIED IN LEAP

There are six subsections of this section, each describing some aspect of the algorithm currently employed by LEAP.

1. The Successive Over-relaxation Method (SOR)

The Successive Over-relaxation Method (SOR) is a generalization of the nonlinear Gauss-Seidel method for solving systems of equations of the form

$$f_i(q_1, \dots, q_n) = 0, \quad i=1, \dots, n.$$

The basic reference for this subsection is Ortega and Rheinbolt [9].

Other supporting material can be found in Collatz [4] and Rheinbolt [12].

The algorithm proceeds as follows:

Given trial values $q_1^k, \dots, q_n^k$ of the variables $q_1, \dots, q_n$, suppose we have computed values $q_1^{k+1}, \dots, q_{i-1}^{k+1}$ for the $(k+1)$ st iteration. We then determine $\tilde{q}_i^{k+1}$ by solving the equation

$$f_i(q_1^{k+1}, \dots, q_{i-1}^{k+1}, q_i, q_{i+1}^k, \dots, q_n^k) = 0. \quad (16)$$

We then set

$$q_i^{k+1} = (1-\omega_k)q_i^k + \omega_k \tilde{q}_i^{k+1},$$

where ω_k is the "relaxation parameter." When $\omega_k = 1$, we have the nonlinear Gauss-Seidel algorithm.

The nature of the equations describing the LEAP model are such that the solution of (16) is trivial for each i . This is apparent from the basic structure of the model:

$$S(q^S) = p^S$$

$$F(p^S) = p^D$$

$$D(p^D) = q^D$$

$$G(q^D) = q^S,$$

where the superscripts S and D refer to the supply and demand sides of the network. Intermediate prices (going up) are determined sequentially in F from prices set lower in the network, and quantities set on the previous downward pass. Intermediate quantities (going down) are determined sequentially in G from quantities set higher in the network, and prices set on the previous upward pass.

The SOR technique has been used previously to determine equilibrium points (see, e.g., Gabay [8] and the references therein). Sufficient conditions which guarantee convergence are given in Ortega [9], and, while they apply in many economic models, they cannot be used in LEAP. In particular, while the matrix ∇H often exhibits row diagonal dominance, it cannot be generally established for LEAP.

Indeed, there is an example in this section for which convergence was not obtained for *any* value of the relaxation parameter ω .

2. Some Comments on the Convergence Properties of LEAP

In this subsection it is shown that while it is possible to produce convergence with LEAP even on some problems with discontinuous or extremely peaked market share functions, it is *also* possible to find simple problems

whose structures fit the basic model, yet which completely defy convergence of the algorithm. Three examples illustrating these two extremes are presented below.

*Convergence of LEAP in an extreme case
of the market share formula*

The following example (see Figure 21) illustrates convergence of the algorithm, with the relaxation coefficient ω tending toward zero, for a case in which the market share parameter γ is infinite so that the consumer buys all from the cheaper supplier. The algorithm forces equality of suppliers' prices, so that the consumer buys from both suppliers (but not in equal quantities, as would be the case for $\gamma < \infty$ and prices equal). Convergence can only be obtained if the relaxation coefficient ω tends toward zero because equality of prices is obtained only in the limit, and the market shares sent down are either 0 or 100% at each iteration (and flip back and forth).

One can admire the robustness of the algorithm for its convergence in this extreme case, and at the same time appreciate the importance of being able to dynamically alter the relaxation coefficients in portions of the network in which there is significant sensitivity to them. One can see somewhat more clearly why in other examples with large, but finite, values of the market share parameters γ oscillations occur with moderate values of ω while convergence is obtained with ω sufficiently small.

The problem is shown in the network below.

Figure 21.--T .

For $\gamma = +\infty$, the market share formula becomes

$$\begin{aligned} q_1 &= q_3, \text{ if } p_1 < p_2 \\ &= 0, \text{ if } p_1 > p_2 \\ q_2 &= 0, \text{ if } p_1 < p_2 \\ &= q_3, \text{ if } p_1 > p_2. \end{aligned}$$

We cannot require $q_1 = q_2$ if $p_1 = p_2$, for there is no solution in this case. (Anyway, since the consumer is minimizing cost in choosing among suppliers, he does not care if $q_1 \neq q_2$ when $p_1 = p_2$.)

For the case $\gamma = +\infty$ there is no equilibrium solution if $q_1 = 0$ or $q_2 = 0$. We thus need $p_1 = p_2$ at equilibrium and this leads to the equation $q_1^3 + q_1^2(q_1^2 - .5)^{1/2} = 1$, whose solution is $q_1 = 0.8604396 \rightarrow q_2 = .4902615$.

The LEAP algorithm was started at $q_1 = 1$, $q_2 = 0$ with a relaxation coefficient of $\omega = .05$; ω was changed somewhat heuristically, as shown in Table IV.

From Figure 22, which shows the trajectories of q_1 and q_2 , it seems clear that true convergence can be obtained if ω is decreased

TABLE IV

Iterations	ω
0 - 14	.05
15 - 20	.03
21 - 35	.02
36 - 39	.03
40 - 42	.02
43 - 51	.01
52 - 64	.02

to zero in an appropriate fashion. One can also see that unless ω is decreased to zero, true convergence will *not* be obtained and oscillations of at least some minimum size will continue indefinitely.

It may be of interest to examine the solution of this problem as a function of the market share parameter γ . This is shown in Table V. In this example the solution is relatively insensitive to γ , but, as we have noted, convergence of the LEAP algorithm should be considerably "trickier" for large versus small values of γ .

TABLE V

γ	q_1	q_2	q_3	Market Share of Supplier 1
0	.6894	.6894	1.3788	.500
0.81	.7851	.6000	1.3851	.567
2.4	.8236	.5500	1.3736	.600
9.7	.8492	.5100	1.3592	.625
20.9	.8550	.5000	1.3550	.631
44.3	.8578	.4950	1.3528	.634
95.3	.8592	.4925	1.3517	.636
∞	.8604	.4903	1.3507	.637

Lack of convergence of the LEAP algorithm for a simple example.

Following is a small example of two simultaneous linear equations for which LEAP fails to converge to the solution. This failure to converge is *independent* of the starting point *and* of the relaxation coefficients used.

The two equations are:

$$p = 2q - 1 = f_1(q), \quad (\text{supply equation}) \quad (17)$$

$$p = q = f_2(q), \quad (\text{demand equation}) \quad (18)$$

These are sketched in Figure 23.

Figure 23.--SOR fails.

It is clear that the solution is at $p = q = 1$. We choose to start with q_1 given and

$$p_1 = f_1(q_1) = 2q_1 - 1. \quad (19)$$

The successive values $\{q_i, p_i\}$ are then found from:

$$q_i^* = f_2^{-1}(p_{i-1}) = p_{i-1}, \quad i \geq 2 \quad (20)$$

$$q_i = \alpha_i q_i^* + (1 - \alpha_i) q_{i-1}, \quad i \geq 2, \quad \alpha_i \in (0, 1) \quad (21)$$

$$p_i^* = f_1(q_i) = 2q_i - 1, \quad i \geq 2 \quad (22)$$

$$p_i = \beta_i p_i^* + (1-\beta_i) p_{i-1}, \quad i \geq 2, \quad \beta_i \in (0,1) \quad (23)$$

We shall demonstrate lack of convergence to the solution of the above iteration scheme. Specifically, if $q_1 > 1$, then $p_1 > 1$ and the sequences $\{q_i\}$ and $\{p_i\}$ are strictly increasing (similarly, if $q_1 < 1$, then $p_1 < 1$ and the sequences $\{q_i\}$ and $\{p_i\}$ are strictly decreasing).

Demonstration. (Only for the case $q_1 > 1$.)

The above results follow from:

$$1 < q_{i-1} < q_i < q_i^*, \quad i \geq 2, \quad (24)$$

$$1 < p_{i-1} < p_i < p_i^*, \quad i \geq 2, \quad (25)$$

which we show by induction. For the general step, suppose that (24) and (25) are true for $i = n$; we show that they hold for $i = n+1$. First,

$$p_n^* = 2q_n - 1 = q_n + (q_n - 1) > q_n, \quad (26)$$

(from (22) and (24)). Then

$$q_{n+1}^* = p_n, \quad (\text{from (20)})$$

$$= \beta_n p_n^* + (1-\beta_n) p_{n-1}, \quad (\text{from (23)})$$

$$= \beta_n p_n^* + (1-\beta_n) q_n^*, \quad (\text{from (20)})$$

$$> \beta_n q_n + (1-\beta_n) q_n = q_n, \quad (\text{from (26) and (24)}).$$

Then $q_{n+1} \in (q_n, q_{n+1}^*)$ by (21), so that (24) holds for $i = n+1$. Next,

$$\begin{aligned}
p_{n+1}^* &= 2q_{n+1} - 1, & (\text{from (22)}) \\
&= 2[\alpha_{n+1} q_{n+1}^* + (1-\alpha_{n+1})q_n] - 1, & (\text{from (21)}) \\
&= 2\left[\alpha_{n+1} p_n + (1-\alpha_{n+1})\left(\frac{p_n^*+1}{2}\right)\right] - 1, & (\text{from (22)}).
\end{aligned}$$

Then

$$p_{n+1}^* > 2\alpha_{n+1} p_n + (1-\alpha_{n+1})(p_n+1) - 1, \quad (\text{from (25)})$$

or

$$p_{n+1}^* > p_n + \alpha_{n+1}(p_n-1) > p_n, \quad (\text{from (25)}).$$

Then $p_{n+1} \in (p_n, p_{n+1}^*)$ by (23), so that (25) holds for $i = n+1$.

Now $p_1 = 2q_1 - 1 > q_1 > 1$; and, with only a couple of slight modifications, the above procedure can be used to demonstrate the validity of (24) and (25) for $i = 2$. This completes the demonstration for $q_1 > 1$. For $q_1 < 1$, all the above inequalities are reversed and so the proof is obvious.

It is not difficult to see that if the roles of $f_1(q)$ and $f_2(q)$ are reversed, the method will converge to the solution with any reasonable choices for the α_i and β_i .

Lack of convergence of LEAP through market share.

The previous example displayed a lack of convergence because the fixed point was a "repulsive" as opposed to an "attractive" point (see Collatz [4], p. 202). This is due to the increasing nature of the demand function, a case unlikely to occur in the LEAP model.

The following example also exhibits a repulsive fixed point, but with functions satisfying the basic assumptions of the LEAP model.

$$\begin{array}{c}
 \begin{array}{l}
 | \\
 \diagup \quad \diagdown
 \end{array}
 \begin{array}{l}
 q_3 = 31.7 - 9.1p_3 \\
 q_i = \left(\frac{(1/p_i)}{(1/p_1) + (1/p_2)} \right) q_3 \\
 p_1 = q_1^{-2} \quad p_2 = 4q_2^{-6}
 \end{array}
 \end{array}$$

Figure 24.--An unstable problem.

There is an equilibrium solution at

$$(q_1^*, q_2^*) = (4.181118929, 2.435935767)$$

and this may be generated by starting from

$$(q_1^0, q_2^0) = (1, 1)$$

with a relaxation coefficient of $\omega = 0.1$.

However, we were unable to force convergence from the points (10,1), (1,10), or (5,5). Indeed, with the starting point (5,5), relaxation coefficients of $\omega = 0.1, 0.05, 0.01$ and 0.001 were used, and SOR was applied until divergence was noted. With $\omega = 0.001$, divergence was noted after some 100 iterations.

This example was constructed so that the map $F(S(q))$ is not monotone, and the poor convergence properties are probably a result of this. It has already been noted that the equilibrium point is not stable, in the sense that small perturbations from it produced images under H which tend to diverge from q^* , i.e., q^* is a repulsive fixed point.

3. The LEAP Algorithm Viewed as a Newton Method for Finding a Fixed Point

The following attempts to explain the LEAP algorithm as a heuristic, approximate Newton method for finding the fixed point of a vector transformation (or, equivalently, the zero of a vector function).

The problem addressed may be considered to be one of finding a solution, i.e., a set of values of quantities and prices, which satisfies a large system of simultaneous nonlinear equations. It is probably true that the total number of variables is equal to the total number of equations, so that there is at least a chance that a solution exists and that it is unique. The number of variables and equations is very large, however; probably between 2000 and 3000 for many models. This makes it computationally infeasible to use the usual *direct* methods of seeking a solution. The structure of the equations, however, is such that an alternative approach seems computationally attractive.

Due to the network structure of the model, most of the equations involve only a small number of variables, and, more particularly, the series-parallel nature of the network is reflected in the fact that most of the equations involve only variables associated with a particular process or pair of processes in the network. Thus, if some set of "sufficient" variables, say at the bottom of the network, are assigned particular values, it is possible to sequentially solve the equations for values of the remaining variables. This does not, however, require use of *all* the equations since the "sufficient" variables were assigned values. Because of the network structure, the remaining equations can be used to now solve for values of the "sufficient" variables. If these

values match the assigned ones, then clearly a solution for the entire system is at hand. If not, then it is in order to try a new set of values for the "sufficient" variables, and try the process over again. With intelligence (and, perhaps, luck) an iterative process converging to a true solution of the system may be obtained.

Denote by q the $n \times 1$ vector of "sufficient" variables, typically the "quantities" at the bottom of the network plus, perhaps, some other variables higher up in the network. It is the set of equations, and therefore the network structure, which dictates an appropriate choice of the variables which appear in q .

Assuming that the network, variables, and equations have been specified in a consistent way so that the process described above can be carried out to find a new and unique value of q from a given value, and that unique values of all the other variables are determined in the process, denote by H the vector function which produces the new q from the old one. Let q^t be the value of q after t iterations, $t=0,1,2,\dots$, and let q_i^t be the i th component of q^t , $i=1,\dots,n$. Thus $q^{t+1} = H(q^t)$, and we seek a fixed point q^* such that $H(q^*) = q^*$. The function H cannot be written down conveniently or dealt with analytically except in the simplest cases; we assume for the moment, however, that it has continuous first partial derivatives (and this is a valid assumption in many cases). Then, given q^t at iteration t , a Taylor expansion of H around the point q^t gives for $H(q^*)$,

$$H(q^*) = H(q^t) + [\nabla H(\hat{q})](q^* - q^t), \quad (27)$$

where $[\nabla H(\hat{q})]$ is the $n \times n$ matrix whose i th row is the gradient of

the i th component of H and $\hat{q}$ is some unknown point which is, however, a convex combination of q^t and q^* . But $H(q^*) = q^*$, so substitution and rearrangement yield

$$[I - \nabla H(\hat{q})]q^* = H(q^t) - [\nabla H(\hat{q})]q^t, \quad (28)$$

where I is the $n \times n$ identity matrix. If the inverse matrix $[I - \nabla H(\hat{q})]^{-1}$ exists, then we obtain

$$q^* = [I - \nabla H(\hat{q})]^{-1}H(q^t) - [I - \nabla H(\hat{q})]^{-1}[\nabla H(\hat{q})]q^t \quad (29)$$

or

$$q^* = A H(q^t) + B q^t, \quad (30)$$

where A and B are defined as indicated. Since $\hat{q}$ is unknown we can estimate it by q^t and thus obtain a first order Newton method which gives, instead of q^* , a new estimate of it which is then q^{t+1} . Thus

$$q^{t+1} = A(q^t)H(q^t) + B(q^t)q^t, \quad (31)$$

where we write $A(q^t)$ and $B(q^t)$ to indicate the dependence on q^t . The difficulty is that $H(\cdot)$ is unknown so that $H(q^t)$ is unknown, and thus $A(q^t)$ and $B(q^t)$ are unknown. And to compute numerical estimates by differencing would be a huge numerical task and might possibly yield instabilities or poor numerical results.

Let's return to Equation (30), write $q^{t+1,*}$ for $H(q^t)$, write $q_1^{t+1,*}$ for its i th component, and then write Equation (30) for the i th component of q^* as

$$q_1^* = \sum_{j=1}^n a_{1j} q_j^{t+1,*} + \sum_{j=1}^n b_{1j} q_j^t \quad (32)$$

or

$$q_1^* = \left(a_{11} q_1^{t+1,*} + b_{11} q_1^t \right) + \sum_{j \neq i} \left(a_{1j} q_j^{t+1,*} + b_{1j} q_j^t \right). \quad (33)$$

If $\{q^t\}$ converges to q^* , then $\{q^{t*}\}$ also converges to q^* , and since $(A+B) = I$ (the dependence of $(A+B)$ on t is not explicitly indicated), the second term of (33) approaches 0 as t approaches infinity. By neglecting it we thus obtain an estimate of q_i^* which we designate as q_i^{t+1} , i.e., we take

$$q_i^{t+1} = a_{ii} q_i^{t+1,*} + b_{ii} q_i^t. \quad (34)$$

Now only a_{ii} and b_{ii} are needed, but we still don't know them because $A = A(t) = A(\hat{q}(t))$ (and similarly for B) and $\hat{q}(t)$ is unknown. We do know, however, that $(a_{ii} + b_{ii}) = 1$ in Equation (34) for all t . With $\omega_i(t) = a_{ii}$ and $(1-\omega_i(t)) = b_{ii}$ we get from Equation (34)

$$q_i^{t+1} = \omega_i(t) q_i^{t+1,*} + (1-\omega_i(t)) q_i^t. \quad (35)$$

Now the question is: How should $\omega_i(t)$ be chosen or computed? One approach is as follows. Since we suppose that q^t and $q^{t+1,*}$ are quite close to q^* and therefore to each other for t sufficiently large, and in addition that $(A+B) = I$, we will neglect the dependence of $q_i^{t+1,*}$ on q_j^t for $j \neq i$ and treat $q_i^{t+1,*}$ as only a function of q_i^t . Denote this function by $\hat{H}_i(q_i^t)$. A one-dimensional version of Equation (29) then results:

$$\begin{aligned} q_i^* &= \frac{\hat{H}_i(q_i^t)}{[1 - \hat{H}'_i(\hat{q}_i)]} + \frac{(-\hat{H}'_i(\hat{q}_i))}{[1 - \hat{H}'_i(\hat{q}_i)]} q_i^t \\ &= \frac{1}{[1 - \hat{H}'_i(\hat{q}_i)]} q_i^{t+1,*} + \frac{(-\hat{H}'_i(\hat{q}_i))}{[1 - \hat{H}'_i(\hat{q}_i)]} q_i^t, \end{aligned} \quad (36)$$

where the prime denotes an ordinary derivative. Again, $\hat{q}_i$ is unknown, but one can estimate $\hat{H}'_i(\hat{q}_i)$ by the ratio

$$\frac{\hat{H}_i(q_i^t) - \hat{H}_i(q_i^{t-1})}{q_i^t - q_i^{t-1}} = \frac{q_i^{t+1,*} - q_i^{t,*}}{q_i^t - q_i^{t-1}}. \quad (37)$$

Then we replace q_i^* in Equation (36) by its estimate q_i^{t-1} , thus obtaining Equation (35) with

$$\omega_i(t) = \frac{q_i^t - q_i^{t-1}}{(q_i^t - q_i^{t-1}) - (q_i^{t+1,*} - q_i^{t,*})}. \quad (38)$$

While Equation (35), or one very closely related to it, appears to be used for each i in the LEAP algorithm, Equation (38) is *not* used to determine all $\omega_i(t)$. An alternative to using Equation (38) is simply to use a constant value ω_i ; another alternative is to calculate the $\omega_i(t)$ iteratively by some scheme other than that implied by Equation (38).

One could argue that the approximations behind Equations (35) and (38) make the efficacy of using (38) in (35) somewhat dubious. Nevertheless, it is mentioned in the D.F.I. report [3] that use of (38) produces faster convergence than the use of constant values ω_i (but at the obvious expense of increased data storage). But one important point is that any scheme that causes $\{q^t\}$ to converge to q^* is working, i.e., the proof is in the pudding.

4. Computational Results in the Diamond Example

Referring to Figure 5, page 8 of this report, we now describe some computational characteristics of the Diamond Example.

For the initial run, we set $\gamma = 1$ and $\omega = 0.5$. The algorithm converges quickly to the same point for any initial point which does not make the computer over- or under-flow. Next we tried higher values of

TABLE VI

NUMBER OF ITERATIONS TO CONVERGE FOR DIFFERENT RELAXATION COEFFICIENTS ω

	SP* $1-\omega$	.95	.90	.85	.80	.75	.70	.67	.66	.65	.64	.63	.62	.61	.60	.57	.55 $\rightarrow$
①.	$q_1 = 5.$ $q_2 = 2.$ $q_3 = 1.$	193	70	46	33	26	21	26	19	20	22	24	22	31	43	255	> 300
②.	$q_1 = 4.$ $q_2 = 16.$ $q_3 = .3$	200	99	65	48	37	31	27	27	26	25	25	25	24	35	224	> 300
③.	$q_1 = .5$ $q_2 = 12.$ $q_3 = .3$	197	97	63	38	37	30	27	26	25	24	24	23	29	38	51	> 300

*SP = Starting Point

γ , and different values of ω , to look at how the algorithm converges. This can yield an indication of the algorithm's rate of convergence.

For $\gamma = 10$ and ω above .65, the higher the value of ω , the more iterations are required for convergence. For values of $(1-\omega)$ in the range $.6 < (1-\omega) \leq .67$, we get the smallest number of iterations for convergence from different starting points. As $(1-\omega)$ decreases from .62, the number of iterations increases sharply, and for $(1-\omega) \leq .5$, the algorithm does not converge at all. No matter what the starting point is, however, it fluctuates between two or more points whose average is very close to the equilibrium point achieved in cases where it converges.

Table VI and Figure 25 show the number of iterations required for convergence for various values of ω .

5. Oscillation--An Illustration

It has been pointed out that LEAP can oscillate between distinct vectors, neither of which is an equilibrium point. This phenomenon is easily explained by referring to Figure 26.

In this example, we set $F(q) = q^4$ and $G(p) = \frac{1}{p}$.

If we start at $q^0 = 0.8363$, we obtain the sequence indicated in Figure 26.

In general, let $r = (1-\omega)q + \omega H(q)$. There will be points of oscillation if the system

$$r = (1-\omega)q + \omega H(q)$$

$$q = (1-\omega)r + \omega H(r)$$

has distinct solutions $r \neq q$ for some ω . See Figure 27.

Figure 26.--Oscillation.

Figure 27.--Oscillation if $\omega = \frac{1}{2}$.

6. Computational Results on the Two Period Problem

Introduction.

Referring to Figure 6 of this report, we now describe some computational characteristics of the two period problem. Both the LEAP algorithm and a variant of it were used to solve this problem. The variant passes prices up via the market share formulas instead of by averaging over the supply quantities and their prices, and is described in detail in the next section. The basic LEAP algorithm worked very well in all cases for which the relaxation coefficient ω was sufficiently small to produce convergence. The variant algorithm never produced faster convergence than the basic one, but in many cases it was equally fast, and it never required more than about 10% more iterations than the basic one required. In general, nothing unusual was observed in the operation of the algorithms or in the results obtained. Specific observations and discussion will be given after a summary of the numerical results obtained.

Convergence criterion used with the algorithms.

Let $q_j^i(t)$ be the value of $q_j(t)$ at the beginning of iteration i and let $q_j^{i*}(t)$ be the value of $q_j(t)$ returned at the end of the down pass. The algorithm was stopped for the first i for which

$$\sum_{t=1}^2 \sum_{j=1}^2 \left[q_j^{i*}(t) - q_j^i(t) \right]^2 < \epsilon ,$$

where $\epsilon > 0$ was a parameter to be specified. Unless otherwise indicated, ϵ was taken as 10^{-6} .

Solutions for two typical cases.

(A) Base Case: $\gamma = 4$, $f_1 = 0.6$, $f_2 = 0.4$, $\beta = 0.3$

Starting point: $q_j(t) = 0.8$, $j, t = 1, 2$.

Solution ($\omega = 0.3$, 16 iterations required):

$$q_1(1) = 0.6703 , \quad q_1(2) = 0.8459 ,$$

$$q_2(1) = 0.7133 , \quad q_2(2) = 0.8902 .$$

(B) Alternate Case: $\gamma = 4$, $f_1 = 0.1$, $f_2 = 0.9$, $\beta = 0.3$

Starting point: $q_j(t) = 0.8$, $j, t = 1, 2$.

Solution ($\omega = 0.3$, 17 iterations required):

$$q_1(1) = 0.4778 , \quad q_1(2) = 0.6192 ,$$

$$q_2(1) = 0.8643 , \quad q_2(2) = 1.0641 .$$

Summary of numerical results.

A total of about 60 versions of the problem were solved with variation in the starting point and/or the parameters ω , β , γ , f_1 , f_2 , and ϵ . While these 60 versions were certainly not chosen randomly, they did not constitute any sort of factorial design or statistically valid sample. They were chosen in groups in order to see the effects of the various parameters on (1) the solution obtained, (2) the speed of convergence (or lack thereof), and/or (3) the type of convergence. The following remarks are meant to generally summarize the results, without any real precision intended.

- (i) The starting point does not make much difference. Convergence was sometimes faster from a starting point very far away from the final solution as compared with a starting point quite close to the final solution.

- (ii) The larger the value of γ , the smaller must be the value of ω for convergence to occur. There is, roughly speaking, an optimal value of ω (to minimize the number of iterations required) for a given problem. For smaller values of ω , increasingly large numbers of iterations are required. For larger values of ω , this is also true, and for ω beyond a certain value there is no convergence. Actually, when convergence does not occur the algorithms oscillate between two points in the limit (more about this below), and if these two points happen to be close enough (compared to ϵ), then the algorithms naturally interpret this as convergence.
- (iii) When ω is sufficiently large that the algorithm does not converge but instead oscillates between two limit points, these limit points depend on ω , but not on the starting point. The larger the value of ω , the farther apart are the two limit points. We illustrate with a case with $\gamma = 6$, $f_1 = 0.6$, $\beta = 0.3$. The solution, obtained with $\omega = 0.2$, is:

$$\begin{aligned} q_1(1) &= 0.6583, & q_1(2) &= 0.8346, \\ q_2(1) &= 0.7254, & q_2(2) &= 0.9013. \end{aligned} \quad (\epsilon = 10^{-6})$$

With $\omega = 0.3$ one obtains the following two limit points:

$$\begin{aligned} q_1(1) &= 0.5823, & q_1(2) &= 0.7352, \\ q_2(1) &= 0.7921, & q_2(2) &= 0.9903, \end{aligned}$$

and

$$\begin{aligned} q_1(1) &= 0.7311, & q_1(2) &= 0.9245, \\ q_2(1) &= 0.6425, & q_2(2) &= 0.7998. \end{aligned}$$

With $\omega = 0.35$ one obtains the following two limit points:

$$q_1(1) = 0.5393, \quad q_1(2) = 0.6785,$$

$$q_2(1) = 0.8228, \quad q_2(2) = 1.0308,$$

and

$$q_1(1) = 0.7705, \quad q_1(2) = 0.9712,$$

$$q_2(1) = 0.5911, \quad q_2(2) = 0.7370.$$

Note that the two limit points are not, respectively, dominating and dominated by the true solution with respect to every component.

- (iv) Convergence to the two limit points mentioned in point (iii) may be rather rapid or else quite slow. In one case ($\gamma = 10$, $\beta = 0.3$, $\omega = 0.165$, $f_1 = 0.6$), more than 700 iterations were required before the two limit points did not change in the sixth decimal place. Many fewer iterations were required for such stabilization for both $\omega = 0.2$ and $\omega = 0.155$. The case with $\omega = 0.155$ is interesting because convergence does not actually occur although the two limit points are extremely close to each other; so close that convergence was signaled for ϵ as small as 10^{-8} . With $\epsilon = 10^{-10}$ the following two limit points were obtained:

$$q_1(1) = 0.645041, \quad q_1(2) = 0.826624,$$

$$q_2(1) = 0.737540, \quad q_2(2) = 0.909485,$$

and

$$q_1(1) = 0.645043, \quad q_1(2) = 0.826627,$$

$$q_2(1) = 0.737538, \quad q_2(2) = 0.909482.$$

- (v) The case of point (iv) above was run (for $\omega = 0.2$) with both the basic algorithm and the variant. Both versions settled on two limit points, but different ones in the two versions.
- (vi) Examination of one case ($\gamma = 2$, $\beta = 0.3$, $f_1 = 0.6$, $\epsilon = 10^{-9}$) for a number of different values of ω reveals the following behavior: The optimal value of ω is about 0.475 (15 iterations required for convergence). For values of ω less than or equal to 0.5, the convergence is monotone in the sense that each component value $q_j^{i*}(t)$ forms an essentially monotonic sequence for successive values of i (but for cases with starting points far from the solution there is overshoot). For ω greater than 0.55 the convergence is alternating in the sense that each component value $q_j^{i*}(t)$ forms an alternating sequence for successive values of i . For $\omega = 0.525$ the convergence starts monotone and then alternates.
- (vii) There appears to be a true rate of convergence. This is evident from examination of the base case given in the third subsection. The following results show an increase of three iterations for each reduction of ϵ by a factor of 10:

ϵ	No. Iterations Required
10^{-4}	10
10^{-6}	16
10^{-8}	22
10^{-10}	29
10^{-12}	35

Discussion.

Most of the points worth making were stated in the subsection just preceding, but even some of those bear repeating here. In general, the basic LEAP algorithm works very well with relaxation coefficients in an appropriate range (but of course this is a very small, simple, and well-behaved problem). The variant algorithm worked essentially as well (but not quite), but it must be kept in mind that the variant algorithm has the potential advantage that for certain types of networks it requires fewer input values specified by the user. It is worth mentioning that there is no difficulty in using the variant algorithm because of the multi-period nature of this problem.

For this small problem the character of the solution depends somewhat on the interaction between γ and f_1 ($f_2 = 1 - f_1$), but not much on β . The larger is the value of γ , the smaller is the effect of f_1 . Rate of convergence essentially depends only on γ and ω , and is better for smaller values of γ .

B. VARIATIONS ON THE BASIC ALGORITHM

This section is composed of two subsections. In the first subsection, we present an alternative method for passing prices up through an allocation node, by imposing the market share relations directly on the averaging relation. In the second subsection, we interpret the LEAP algorithm on a simple supply/demand network, and extend the results there to an allocation process, thereby obtaining relations useful to the passing up of derivatives.

Numerical results are given in both subsections.

For notational convenience, we will set $F(q) = F(S(q))$ and $G(p) = G(D(p))$.

1. An Alternative to Averaging in Passing Prices Up

Consider an allocation process of the type displayed in Figure 28.

Figure 28.--An allocation node.

LEAP passes up the average price:

Method 1: Given quantities $q_1, \dots, q_k$ and prices $p_1, \dots, p_k$, pass up price p where

$$p = \frac{\sum_{i=1}^k p_i q_i}{\sum_{i=1}^k q_i} . \quad (39)$$

There is, however, an alternative which does not require the q_i 's:

Method 2: Enforcing the market share formula on (39), we get

$$p = \frac{\sum_{i=1}^k p_i \left(\frac{\left(\frac{1}{p_i + k_i} \right)^\gamma}{\sum_{i=1}^k \left(\frac{1}{p_i + k_i} \right)^\gamma} \right) q}{q} ,$$

i.e.,

$$p = \frac{\sum_{i=1}^k p_i \left(\frac{1}{p_i + k_i} \right)^\gamma}{\sum_{i=1}^k \left(\frac{1}{p_i + k_i} \right)^\gamma} . \quad (40)$$

Note that this price passed up is independent of the quantities q_i .

Note also that the fractions

$$\frac{\left(\frac{1}{p_i + k_i} \right)^\gamma}{\sum_{i=1}^k \left(\frac{1}{p_i + k_i} \right)^\gamma}$$

which weight the p_i 's of (40), are also used in the downward pass, and hence must be calculated anyway.

Table VII exhibits convergence characteristics of the two schemes on the diamond example. Here, the stopping criterion used is

$$||x_{k+1} - x_k||^2 \leq \epsilon ,$$

where $\epsilon = 10^{-7}$. The rate of convergence used was

$$r = \left(\frac{\epsilon}{||x_1 - x^*||^2} \right)^{\frac{1}{2N}} .$$

For Method 1, the initial q_i 's were $q_1=5$, $q_2=1$, $q_3=1$, $q_4=1$, $q_5=7.5$, $q_9=5$, and $q_{11}=11$. The same values were used in Method 2, except only q_1 , q_2 , and q_3 were required.

Method 2 is clearly superior for this example.

However, other examples exhibit the opposite behavior. In particular, comparisons were made in the two-period example described in Section A of Part I. Table VIII illustrates the results for nine specific runs, with different starting points, different relaxation coefficients, etc.

TABLE VIII
COMPARISON OF AVERAGING ALTERNATIVES

LEAP (Method 1)	25	10	61	76	50	101	51	48	60
Variant (Method 2)	25	10	69	81	51	108	52	48	62

In each case, standard averaging did at least as well as the variant, although the variant never did significantly worse.

In addition to requiring slightly fewer computations, the variant method is useful in decomposition schemes (see Section D of this Part).

TABLE VII
CONVERGENCE CHARACTERISTICS

Relaxation Co- efficient $1-\omega$:	.95	.90	.85	.80	.75	.70	.65
Number of Iterations	187	73	47	35	27	22	18
Rate of Convergence	.9525	.8827	.8238	.7709	.7137	.6610	.6029
New Scheme							
Number of Iterations	127	62	40	29	23	18	15
Rate of Convergence	.9308	.8634	.7964	.7305	.6730	.6029	.5449

TABLE VII--continued

Relaxation Co- efficient $1-\omega$:	.60	.59	.58	.51	.50	.45	.40
Number of Iterations	16	14	19	123	-- ^a	-- ^a	-- ^a
Rate of Convergence	.5660	.5218	.6192	.9286	-- ^a	-- ^a	-- ^a
New Scheme							
Number of Iterations	14	15	15	19	27	155	-- ^b
Rate of Convergence	.5218	.5449	.5449	.6192	.7134	.9429	-- ^b

^a Does not converge in 300 for $\omega \geq .50$.

^b Does not converge.

2. Passing Derivatives (Differences) Through Allocation Processes

Consider the simplest supply/demand network illustrated in Figure

29. LEAP sets $q^1 = (1-\omega)q^0 + \omega H(q^0)$ where $H = G(F(\cdot))$. If $F(q) \approx F(q^0) + s(q-q^0)$ and $G(p) \approx G(F(q^0)) - t(p-F(q^0))$, then the point of intersection of the graphs of F and G is

$$q^* = \left(\frac{st}{1+st} \right) q^0 + \left(\frac{1}{1+st} \right) H(q^0) .$$

Figure 29.--Supply/demand network.

In particular, if $s = F'(q^0)$ and $t = -G'(p^0) = -G'(F(q^0))$, and noting that $H'(q^0) = G'(F(q^0)) \cdot F'(q^0)$, we have

$$q^* = \left(\frac{-H'(q^0)}{1-H'(q^0)} \right) q^0 + \left(\frac{1}{1-H'(q^0)} \right) H(q^0) ,$$

which is the formula one obtains by applying Newton's Method to the equation $0 = q - H(q)$. See also subsection 3 of Section A (pp. 61 ff.).

Two successive passes up the network, starting from points q^0 and q^1 , will suffice to determine an approximation s to the slope of F :

$$s = \frac{F(q^1) - F(q^0)}{q^1 - q^0} .$$

Coming down the network, starting from $p^0 (= F(q^0))$ and $p^1 (= F(q^1))$ will yield an approximation t to the negation of the slope of G :

$$t = - \left(\frac{G(p^0) - G(p^1)}{p^0 - p^1} \right) .$$

If $G(p^i) = G(F(q^i)) \doteq H(q^i)$ ($i=0,1$) , we get

$$q^* = \left[\frac{1}{1 - \left(\frac{q_1 - q_0}{H(q_1) - H(q_0)} \right)} \right] q^i + \left[\frac{1}{1 - \left(\frac{H(q_1) - H(q_0)}{q_1 - q_0} \right)} \right] H(q^i) .$$

Example.

In Section A of this part we presented an example illustrating oscillation with ω fixed at 0.5. There, $F(q) \doteq q^4$ and $G(p) \doteq \frac{1}{p}$.

Applying the formula

$$1-\omega = \frac{1}{1 - \left(\frac{q_1 - q_0}{H(q_1) - H(q_0)} \right)} ,$$

we generated the results below:

i	q_i	$1 - \omega_i$
0	.836 000	0.5 (arbitrarily set)
1	1.441 630	0.250 119
2	1.138 960	0.454 873
3	0.891 184	0.199 997
4	1.030 019	0.166 114
5	1.006 498	0.214 756
6	0.999 610	0.202 441
7	1.000 005	0.199 846
8	1.000 000	0.200 000

The actual value of $1-\omega$ at the equilibrium point is

$$1-\omega = \left(\frac{-H'(q^0)}{1 - H'(q^0)} \right)$$

$$= \frac{1}{5}.$$

We now wish to extend this analysis to obtain a formula which will apply at an allocation node. Here we will assume derivatives exist, although differences could be employed instead.

Referring to Figure 30, we will set

Figure 30.--An allocation process.

$$F(q) = \sum_{i=1}^k \frac{S_i(q_i)q_i}{q}$$

and

$$G(p) = \begin{pmatrix} f_1 \\ \vdots \\ f_k \end{pmatrix} D(p),$$

where

$$f_i = \frac{\left(\frac{1}{S_i(q_i) + k_i} \right)^{-\gamma}}{\sum_{i=1}^k \left(\frac{1}{S_i(q_i) + k_i} \right)^{-\gamma}}.$$

Note that G is actually a function of q and p , but for q fixed

on the upward pass, G depends only on the (real) variable p .

It is easy to verify that

$$\nabla F(q) = \frac{1}{\left(\sum_{i=1}^k q_i \right)} \left[(\dots, s'_1(q_1)q_1 + s_1(q_1), \dots) - F(q)(1, \dots, 1) \right]$$

and

$$G'(p) = \begin{pmatrix} f_1 \\ \vdots \\ f_k \end{pmatrix} D'(p).$$

Also, note that both ∇F and G' can be approximated by differences, which can be obtained by two successive passes through the network.

As before, we note

$$F(q) \approx F(q^0) + \nabla F(q^0)(q - q^0) \quad (41)$$

and

$$G(p) \approx H(q^0) + G'(p^0)(p - p^0) \quad (42)$$

where $H(q^0) = G(F(q^0))$. Setting $q = G(p)$, $p = F(q)$ and $p^0 = F(q^0)$, multiplying (42) on the left by $\nabla F(q^0)$, we obtain

$$\nabla F(q^0)q = \nabla F(q^0)H(q^0) + [\nabla F(q^0)G'(p^0)](p - p^0).$$

From (4), we see that

$$\nabla F(q^0)q = p - p^0 + \nabla F(q^0)q^0,$$

so that

$$p - p^0 = \left[\frac{\nabla F(q^0)(H(q^0) - q^0)}{1 - \nabla F(q^0)G'(p^0)} \right], \quad (43)$$

which determines the value of p where (41) and (42) are simultaneously satisfied.

If we substitute (43) into (42) to obtain the value of q for which (41) and (42) are simultaneously satisfied, we obtain

$$q = H(q^0) + G'(p^0) \left[\frac{\nabla F(q^0)(H(q^0) - q^0)}{1 - \nabla F(q^0)G'(p^0)} \right]$$

i.e.,

$$q = \left[\frac{1}{1 - \nabla F(q^0)G'(p^0)} \right] \left[-G'(p^0)\nabla F(q^0) \right] q^0 + \left[I + \left(\frac{1}{1 - \nabla F(q^0)G'(p^0)} \right) \left[G'(p^0)\nabla F(q^0) \right] \right] H(q^0),$$

which is of the SOR form

$$q = (I - \omega)q^0 + \omega H(q^0),$$

where ω is a *matrix*. The interpolated value of q does not generally lie on the line segment $[q^0, H(q^0)]$.

Not only is the above relation expensive to carry out on the computer, but its application to the example which follows exhibited rather slow convergence.

Instead, we note that

$$q = \left(1 - \frac{1}{1 - \nabla F(q^0)G'(p^0)} \right) q^0 + \left(\frac{1}{1 - \nabla F(q^0)G'(p^0)} \right) H(q^0) \quad (44)$$

does lie on the line segment $[q^0, H(q^0)]$ and also does yield the proper value of p as derived in (40).

We note that these formulas generalize the single commodity case discussed earlier, but fail to take into account the new feature implied by the market share requirement, i.e., that the function G is also now a function of q . To take this dependence into account would require differentiation of the fractional shares, resulting in matrix operations which would undoubtedly be beyond the spirit of the SOR algorithm.

Note also that the fraction

$$\omega = \frac{1}{1 - \nabla F(q^0)G'(p^0)}$$

should generally lie in the interval (0,1), since F generally behaves like a monotone increasing operator, and G a monotone decreasing operator (see Section D of Part I).

Example.

Table IX illustrates the convergence of the example displayed in Figure 31. Table X compares this convergence with fixed values of ω , and different starting points.

Figure 31.--An example.

TABLE IX

A SINGLE RUN ON EXAMPLE

ω_1	q_1	q_2
0.838	.286	.392
0.751	.333	.447
0.751	.332	.448
0.750	.332	.447
$\vdots$	$\vdots$	$\vdots$
0.750 202	.331 998	.447 602

TABLE X

NUMBER OF ITERATIONS REQUIRED UNTIL $||(q_1, q_2) - q^*|| < 10^{-6}$

Starting Point	Fixed Relation Coefficients ω								Variable Coefficient
	1.0	.9	.8	.7	.6	.5	.4	.3	ω_1
(1,1)	29	16	9	7	9	13	18	27	8
(5,5)	31	18	11	8	11	15	22	--	11
(1,5)	--	19	13	9	10	14	21	--	12
(5,1)	--	19	11	9	11	14	19	--	12

While the use of variable relation coefficients produces reasonably good convergence, it does not do as well as *some* of the fixed coefficients. There are two reasons for this. First, it has already been pointed out that (44) does not account for the dependence of G on q . Second, the variable coefficients computed at points far from the equilibrium point are poor approximations to the limiting value, thus producing slow convergence initially.

C. NEW APPROACHES FOR SOLVING THE EQUILIBRIUM EQUATIONS

Even if LEAP does converge to the equilibrium solution, the rate of convergence is at best linear, and this linear rate depends strongly on the selection of a parameter value. In this section, some traditional methods for solving equations are discussed and their merits (mostly through the ability to do post-equilibrium sensitivity analysis) are presented.

Newton's method, when it can be implemented, is far and away the fastest algorithm for solving equations. Use of it requires derivatives. For the small models used to experiment with, these were automatically computed by the factorable programming language developed at George Washington University. There might be some difficulty in obtaining derivatives for the large LEAP model because of the complicated FORTRAN subroutines used to establish the functional relationships. Some of these appear to be nonsmooth and discontinuous, which further complicates matters. The generalized secant method, discussed second, operates with function values only. These are available in the LEAP model, since they define the equilibrium equations. The main objection to the use of these equation-solving techniques is that viewed in the way the LEAP model builders do, there are approximately 100,000 equations and variables. Part of the effort in the later sections is to show how the attaining of equilibrium can be regarded as solving as many equations as there are

basic suppliers in just one time period. Part of the variable reduction occurs because of the techniques developed in Section B in passing prices without regard to intermediate quantities produced using the market share formulas. In Section D, a reformulation of the basic variables into cumulative production eliminates the dependence upon all but the cumulative production in the final time period. Other efforts to decompose the problem are investigated in Section D, but the results here are not as clear cut.

The main advantage in using either the Generalized Secant Method or Newton's method is that sensitivity analysis can be performed. As is the case with all large energy models, there is uncertainty as to the parameter values which specify the functional forms (e.g., values of γ having to do with market penetration, values of k_i that specify "price premiums," and beyond the horizon supply prices). The central question is which of the parameter values are most important in determining an equilibrium point. Knowing this points the way to allocating scarce data gathering resources to determine the proper values. Both Newton's method and the Generalized Secant Method generate enough information to perform sensitivity analysis.

1. Newton's Method

The following diagram illustrates the way quantities are produced, prices are set and transformed up the network, energy products are demanded, and demand is transferred down the network.

The network is in equilibrium when $H(q) = G[D[F[S(q)]]] = q$. Let ϵ be the vector of parameters which define the model. The equivalent problem

Figure 32.--Flow of energy network.

is to find q^* solving

$$F(q, \epsilon) = H(q, \epsilon) - q = 0.$$

In Exhibit 2 is the encoding of this process using the factorable programming language. Exhibit 1 is the figure describing the diamond-shaped network being analyzed. Note the parameter GAM which has a value of -10. for this example.

Newton's method for solving the above equation takes the form

$$q_{k+1} = q_k - F'(q_k, \epsilon)^{-1} F(q_k, \epsilon).$$

The values of the functions and the $n \times n$ derivative matrix are automatically calculated by the factorable language. The code implementing Newton's method was integrated with the factorable programming language. It has a step-size modification capability used to insure that the sum of the squares of the individual functions decreases on each iteration. The final information on the solution of this problem (final iteration) is given in Table XI. It took five iterations starting from the point $q_0 = (.5, 1., 1.)^T$.

Exhibit 1.--Diamond shaped energy network with three suppliers.


```

4FJHT
NJ/4-2338
4/4-1920
TRY SOLVING DIAMONDEXAMPLENETWORK USING NEWTON'S METHOD
3 43 0 3 1 0 1
V IQ1 5.
V O2 1.
V O3 1.
P GAM 10.
S P1 Q1 EXP
S P2 Q2 POW 3.
S P3 Q3 POW 2.
S P3 Q3 LIN 0. 1.
S C1 P1 LIN 1. 1.
S C4 P2 LIN 1. 4.
S MC1 C1 POW GAM
S MC4 C4 POW GAM
S CC MC1 IDN
S CC MC4 IDN
Q MSC1 MC1 IDN
Q MSC1 CC POW -1.
S MSC4 MSC1 LIN -1. 1.
Q P6 P1 IDN
Q P6 MSC1 IDN
Q P6 P2 IDN
Q P6 MSC4 IDN
S C3 P3 LIN 1. 3.
S C5 P2 LIN 1. 5.
S MB3 C3 POW GAM
S MB5 C5 POW GAM
S BB MB3 IDN
S BB MB5 IDN
Q MSB3 MB3 IDN
Q MSB3 BB POW -1.
S MSB5 MSB3 LIN -1. 1.
Q P7 P3 IDN
Q P7 MSB3 IDN
Q P7 P2 IDN
Q P7 MSB5 IDN
S C9 P6 LIN 1. 9.
S C11 P7 LIN 1. 11.
S MA9 C9 POW GAM
S MA11 C11 POW GAM
S AA MA11 IDN
S AA MA9 IDN
Q MSA9 MA9 IDN
Q MSA9 AA POW -1.
S MS11 MSA9 LIN -1. 1.
Q P10 P6 IDN
Q P10 MSA9 IDN
Q P10 P7 IDN
Q P10 MS11 IDN
S O8 P6 POW -1.
S O10 P10 POW - .5 1.4142135
S O12 P11 POW -.33333333 1.4422496
Q O9 Q10 IDN
Q O9 MSA9 IDN
Q Q11 Q10 IDN
Q Q11 MS11 IDN
S O6 O8 IDN
S O6 O9 IDN
Q O4 O6 IDN

```

Exhibit 2.--Encoding of
diamond shaped
energy model
in factorable
programming
language.

S	07	011	IDN		
S	07	012	IDN		
Q	05	07	IDN		
Q	05	MSB5	IDN		
Q	H1	06	IDN		
Q	H1	MSC1	IDN		
S	H1	01	LIN	-1.	
S	H2	04	IDN		
S	H2	05	IDN		
S	H2	02	LIN	-1.	
Q	H3	07	IDN		
Q	H3	MSB3	IDN		
S	H3	03	LIN	-1.	
S	OBJ	01	LIN	0.	7.77.
U	H1	H3		0.0	
S					

	.0001	.0001
--	-------	-------

1	4
---	---

//

Exhibit 2.--continued

TABLE XI

PRINTOUT OF FINAL ITERATION OF NEWTON'S METHOD APPLIED
TO DIAMOND SHAPED ENERGY MODEL ($a_{ij} = d[H_i(q) - q_i]/dq_j$)

K=	4		
STEP XX	0.974290D 00	0.705625D 00	0.101179D 01
FUNCTION VALUES	0.746368D-06	-0.350967D-05	0.189553D-05
I=	0	CNORM= 0.1646863D-10	
DERIVATIVE MATRIX			
	-0.210530D 01	0.135306D 00	-0.152389D-01
	0.103532D 01	-0.273849D 01	0.144634D 01
	0.152199D-01	0.593251D 00	-0.235013D 01

Sensitivity analysis takes the following form: let $q^*(\epsilon)$ denote the solution point for some vector of parameter values ϵ . The rate of change of the solution point with respect to changes in ϵ is gotten by differentiating the identity

$$F[q^*(\epsilon), \epsilon] \equiv 0$$

with respect to ϵ , yielding

$$F'[q^*(\epsilon), \epsilon] dq^*(\epsilon)/d\epsilon + \partial F[q^*(\epsilon), \epsilon]/\partial \epsilon = 0.$$

Solving,

$$dq^*(\epsilon)/d\epsilon = -F'[q^*(\epsilon), \epsilon]^{-1} \partial F[q^*(\epsilon), \epsilon]/\partial \epsilon.$$

The matrix inverse is available, having been used in applying Newton's method. The quantity on the right above is automatically computed by the factorable programming system.

The computer program calculated the sensitivity of the answer for the diamond shaped equilibrium problem as

$$dq^*(\epsilon)/d\epsilon = (-.00867, .0184456, .001977)^T.$$

To show how this information might be used, the solution q^* for $\gamma = -9$ can be estimated by

$$q^*(-9) \doteq q^*(-10) + dq^*(-10)/d\gamma \cdot (1) = (.96562, .724071, 1.013767)^T.$$

For comparison purposes the problem was resolved for $\gamma = -9$ yielding as an equilibrium point, $q^*(-9) = (.964926, .724322, 1.01337)^T$.

2. Generalized Secant Method

It was considered useful to try to solve the diamond shaped energy network using an algorithm whose requirements could be fulfilled by the large LEAP model. The generalized secant method requires only function values for its implementation, and these are of course available from the big model. The advantage of the generalized secant method over LEAP is its rapid rate of convergence which should be superlinear (faster than a linear rate). It uses information gained from iteration to iteration in accelerating the rate at which convergence to the equilibrium point is achieved. It can be considered as a finite difference approximation to Newton's method. This will now be done in the following brief description.

Assume $N+1$ points $q^1, \dots, q^{N+1}$ and their associated function values $F(q^1), \dots, F(q^{N+1})$ are available. The approximation to a point satisfying the equations is

$$q^{N+2} = [q^1, \dots, q^{N+1}] \left\{ \begin{matrix} F(q^1), \dots, F(q^{N+1}) \\ 1, \dots, 1 \end{matrix} \right\}^{-1} \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

where $\underline{0}$ is an $N \times 1$ vector of zeroes.

Since 1959 quasi-Newton methods have been most popular both in minimizing unconstrained functions, and solving nonlinear equations. It was our feeling that the Secant method had not been thoroughly investigated with respect to its rate of convergence and convergence properties, and that it should be at least as good, if not better, than the quasi-Newton methods. The thing not tried in the literature with the generalized secant method is the allowance of a variable step-size (instead of the value of one implicitly assumed above). A code was written and an algorithm developed which allowed for a variable step-size. Basically, a new point is estimated using the formulas above. Interpolation between that and a "good" base point is used to obtain a point whose function values "decrease" in some sense. (If the point given above does have this property, it is accepted.) The new point then replaces the "worst" point of the $N+1$ currently available, and the next iteration is performed assuming convergence has not been achieved.

The results of this applied to the diamond shaped energy example are given in Exhibit 3. The algorithm required 15 iterations to converge to a point where the function values differed by at most 10^{-10} .

Since the generalized secant method also gives an approximation to the inverse derivative matrix required for sensitivity analysis, it is next envisioned to code a sensitivity analysis routine to be used in conjunction with it. The obtaining of the partial derivative with respect to any parameter of interest of the large LEAP model can be done by finite differencing. This requires simply one up and down pass keeping the q values fixed and varying the parameter value slightly.

	X(1)	X(2)	X(3)	F(X)	F(1)	F(2)	F(3)
	0.1500D 01	0.4500D 01	0.5500D 01	F(X):	-0.6089D 00	-0.4500D 01	-0.5042D 01
	0.3500D 01	0.3000D 01	0.5500D 01	F(X):	-0.3446D 01	-0.2476D 01	-0.5306D 01
	0.3500D 01	0.4500D 01	0.4500D 01	F(X):	-0.3450D 01	-0.4500D 01	-0.3697D 01
	0.3500D 01	0.4500D 01	0.5500D 01	F(X):	-0.3343D 01	-0.4500D 01	-0.4920D 01
RMS:	0.1015D 02	0.1123D 02	0.1165D 02		0.1276D 02	4	1

1580
AR: 0.5061D 00 0.1165D 01 0.6203D 00

TYPE: 0
I= 0

				F(X)			
	0.5061D 00	0.1165D 01	0.6203D 00	F(X):	0.1008D 01	-0.1138D 01	0.8500D 00
S:	0.1015D 02	0.1123D 02	0.1165D 02		0.2996D 01	3	4
	0.1211D 01	0.3592D 00	0.1150D 01				

				F(X)		
S:	0.8586D 00	0.7619D 00	0.8850D 00	F(X):	0.2492D 00	-0.3568D 00
CTION:	-0.7050D 00	0.8054D 00	-0.5294D 00			0.3053D 00

TYPE: 1
I= 1

S:	0.1015D 02	0.1123D 02	0.9213D 00		0.2996D 01	2
	0.9111D 00	0.5993D 00	0.1031D 01			3

Exhibit 3.--
Solution of
diamond shaped
example using
the generalized
secant method.

TYPE: 0
I= 0

				F(X)			
	0.9111D 00	0.5993D 00	0.1031D 01	F(X):	0.1196D 00	0.2451D 00	-0.1013D 00
S:	0.1015D 02	0.4659D 00	0.9213D 00		0.2996D 01	1	2
	0.9736D 00	0.6285D 00	0.1031D 01				

TYPE: 0
I= 0

				F(X)			
	0.9736D 00	0.6285D 00	0.1031D 01	F(X):	-0.8515D 02	0.2389D 00	-0.8768D 01
S:	0.3351D 00	0.4659D 00	0.9213D 00		0.2996D 01	4	1
	0.9700D 00	0.3836D 00	0.1047D 01				

0.3351D 00 0.4659D 00 0.9213D 00 0.2449D 00
 0.9700D 00 0.3336D 00 0.1047D 01

0.9754D 00 0.7510D 00 0.023D 01 F(X): 0.4092D-02 -0.1085D 00 0.2322D-02

82
 CTION: 0.3629D-02 0.2449D 00 -0.1585D-01

TYPE: 2
 I= 1

 S: 0.3351D 00 0.4659D 00 0.9213D 00 0.1149D 00 3 4
 0.9746D 00 0.7097D 00 0.1012D 01

TYPE: 0
 I= 0

 S: 0.9746D 00 0.7097D 00 0.1012D 01 F(X): -0.7342D-04 -0.1045D-01 0.1717D-02 0.1000D 0
 0.3351D 00 0.4659D 00 0.1224D-01 0.1149D 00 2 3
 0.9743D 00 0.7056D 00 0.1012D 01

Exhibit 3.--
 continued

TYPE: 0
 I= 0

 S: 0.9743D 00 0.7056D 00 0.1012D 01 F(X): -0.1082D-04 -0.5983D-04 -0.1542D-05 0.1000D 0
 0.3351D 00 0.7219D-04 0.1224D-01 0.1149D 00 1 2
 0.9743D 00 0.7056D 00 0.1012D 01

TYPE: 0
 I= 0

 S: 0.9743D 00 0.7056D 00 0.1012D 01 F(X): 0.5310D-06 -0.2438D-05 0.1003D-05 0.1000D 0
 0.3972D-05 0.7219D-04 0.1224D-01 0.1149D 00 4 1
 0.9743D 00 0.7056D 00 0.1012D 01

TYPE: 0
 I= 0

 S: 0.9743D 00 0.7056D 00 0.1012D 01 F(X): -0.5691D-07 -0.1582D-06 -0.1308D-06 0.1000D 0
 0.3972D-05 0.7219D-04 0.1224D-01 0.3459D-06 3 4
 0.9743D 00 0.7056D 00 0.1012D 01

0.9743D 00 0.7056D 00 0.1012D 01 F(X): 0.5310D-06 -0.2438D-05 0.1003D-05 0.100
 RMS: 0.3972D-05 0.7219D-04 0.1224D-01 0.1149D-00 4 1
 XBAR: 0.9743D 00 0.7056D 00 0.1012D 01

TYPE: 0

I= 0

X: 0.9743D 00 0.7056D 00 0.1012D-01 F(X): -0.5691D-07 0.1582D-06 -0.1308D-06 -0.100
 RMS: 0.3972D-05 0.7219D-04 0.1224D-01 0.3459D-06 3 4
 XBAR: 0.9743D 00 0.7056D 00 0.1012D-01

RUP:

ZZFR

RE2-259

X 1*= 0.9743D 00 F(1)*= 0.2405D-10

X 2*= 0.7056D 00 F(2)*= -0.6930D-10

X 3*= 0.1012D 01 F(3)*= 0.6516D-10

TYPE: 0

I= 0

Exhibit 3.--
continued

3. Summary of Attempts to Accelerate Convergence to an Equilibrium Point

In Exhibit 4 are reproduced the best recent solutions of the diamond shaped example using LEAP. This was the best because the use of the market share formulas going *up* as well as down reduced the number of basic variables of the problem. As is evident, the number of iterations is strongly dependent upon the choice of the parameter ω . Newton's method applied to this same problem took five iterations and the generalized secant method took 15. In addition to their increase in speed, both the latter methods allow for automatic sensitivity computation after the equilibrium point is obtained. It is suggested that these techniques be applied to the larger model to see if computation time can be reduced.

Relaxation Co-efficient $1-\omega$:		.95	.90	.85	.80	.75	.70	.65
	Number of Iterations	187	73	47	35	27	22	18
	Rate of Convergence	.9525	.8827	.8238	.7709	.7137	.6610	.6029
New Scheme	Number of Iterations	127	62	40	29	23	18	15
	Rate of Convergence	.9308	.8634	.7964	.7305	.6730	.6029	.5449

Relaxation Co-efficient $1-\omega$:		.60	.59	.58	.51	.50	.45	.40
	Number of Iterations	16	14	19	123	-- ^a	-- ^a	-- ^a
	Rate of Convergence	.5660	.5218	.6192	.9286	-- ^a	-- ^a	-- ^a
New Scheme	Number of Iterations	14	15	15	19	27	155	-- ^b
	Rate of Convergence	.5218	.5449	.5449	.6192	.7134	.9429	-- ^b

^a Does not converge in 300 for $\omega \geq .50$.

^b Does not converge.

Exhibit 4.--Rate of convergence of LEAP on diamond shaped equilibrium problem.

D. DECOMPOSITION OF THE ENERGY NETWORK

Several experiments were conducted to see if it is possible to decompose the energy network in such a way that equilibrium could be achieved by iterating on pieces of the network separately. The experiments were conducted on small problems and for which the derivatives of the separate equations could be computed. This was done automatically using the factorable programming language developed at G.W.U. The advantage of using small analytic models is that insight can be gained into the coupling of the different parts of the network. Examination of the derivative matrix should show the relative importance of values of some of the supply levels on the equilibrium equations of another. If the derivative values are small, then this implies that the equilibrium attainment could possibly be achieved separately, i.e., through some decomposition procedure. With this goal in mind, the diamond example and the two supplier two-time period example of Section A, Part I, were examined.

The first experiment used the diamond shaped example. The price of the second supplier was frozen at $p_2 = .8$ (see Figure 33 below). Then were calculated the equilibrium quantities for the three suppliers. Finally, the question was asked how would the equilibrium quantities change as a function of a change in the constant price? Although this problem still has three variables, in effect it only has two since the quantity produced by the second supplier is a function of the first and third.

Figure 33.--Network with second supplier price fixed.

This is reflected by looking at the derivative matrix (at equilibrium) of the matrix of equilibrium equations (see Table XII). The second column has its only nonzero element as the second element. This reflects the fact that the constant price for the second supplier makes the iterations of the equilibrium equations for the first and third independent of the second supplier. The calculations for solving the reduced problem should be fewer than those required for the full problem. This is one way in which part of the network could be "shut down."

TABLE XII

DERIVATIVE MATRIX FOR EQUILIBRIUM EQUATIONS

$$(a_{ij} = d[H_i(q) - q_i]/dq_j)$$

$i \backslash j$	1	2	3
1	-2.0930	0.0000	.0481
2	.5034	-1.0000	1.0244
3	.1506	0.0000	-2.2459

After equilibrium is achieved for the reduced, or partially shut down, network, the question must be answered: how would the equilibrium quantities change if the price, held fixed, were changed? This is a question of the sensitivity of the equilibrium quantities with respect to the parameter, the second supplier's price in this case.

Without the use of derivatives it is not clear how to answer this question. Because of the analytic nature of this problem and the capability of the factorable programming language in differentiating the problem functions with respect to their parameters, calculation of a vector which answers this question can be done automatically. This yields $d[q_1(p_2), dq_2(p_2), dq_3(p_2)]/dp_2 = (.0400, -.4877, .1302)$, (at $p_2 = .8$). This information can be used in two ways. Analysis of the vector yields the intuitive conclusion that the second supplier's equilibrium quantity would be most affected by a change in the arbitrary price assigned to it. The *not a priori* conclusion that the first quantity would be least affected by a change in that price is also a result of this analysis.

The second use is to estimate the new quantities by giving a possible value to the change in p_2 and estimate the change in the quantities supplied. For purposes of this example, it is known that the equilibrium with p_2 computed as a function of q_2 occurs at $p_2 = .3513$. Using the change $\Delta p_2 = -.4487$, the equilibrium quantities are approximated by

$$\begin{aligned} q(p_2 + \Delta p_2) &\doteq q(p_2) + [dq(p_2)/dp_2] \Delta p_2, \\ \text{or} \\ q(.3513) &= (.9743, .7056, 1.0118) \\ &\doteq (.9754, .6301, 1.0204) \\ &= q(.8) + [dq(.8)/dp_2](-.4487), \end{aligned}$$

which is fairly close agreement in the first and third components, but not in the second.

A second reasonable way in which part of the energy network can be "shut down" is to freeze demand on certain sectors or quantities on the top of the new network. With part of the demand portion of the model frozen, the attaining of equilibrium can be concentrated on satisfying the bottom equations without making full iterations to the top and down. This presumably would require less effort, particularly in situations in which demand is fairly insensitive in certain areas to the prices passed up from the bottom, or in which the prices are insensitive to the quantity produced, but large changes in quantity produced result from small changes in price. (This happens when the market share formulas approximate the case of allocating demand to the cheapest supplier, giving a seesaw effect between iterations.)

To examine this, the diamond shaped example was changed to fix the demands q_6 and q_7 (see Figure 34 below). In Table XIII is the derivative matrix of the equations for this network. The (3,1) and (1,3) elements are now zero because of the fact that the attaining of equilibrium on the first and third suppliers can be done independently.

The attaining of equilibrium for this network occurs at quantities (.8933, .6348, .9719) . The second question must again be answered: What would the effect be on the equilibrium quantities if the demands q_6 and q_7 were altered? Again, this question can be answered using knowledge of the derivative matrix and the partial derivatives of the equilibrium equations with respect to the parameters q_6 and q_7 .

Figure 34.--Network with demand fixed.

The sensitivity (first order) approximation is again simply

$$q[(q_6, q_7) + \Delta q(q_6, q_7)] \doteq q[(q_6, q_7)] + [dq(q_6, q_7)/d(q_6, q_7)] \begin{pmatrix} \Delta q_6 \\ \Delta q_7 \end{pmatrix}.$$

The values of q_6 and q_7 for the network without shutdown are (1.1420, 1.5497). As a check, the full equilibrium values were estimated using the derivative quantities given by the factorable programming differentiation routine as

$$\begin{aligned} q(1.1420, 1.5497) &= (.9743, .7056, 1.0118) \\ &\doteq (.9789, .7040, 1.0088) \\ &= (.8933, .6348, .9719) + (.5755, .3178, .1067)(.1420) \\ &\quad + (.0785, .4844, .4371)(.0497) \\ &= q(1., 1.5) + \{dq(1., 1.5)/dq_6\}\Delta q_6 \\ &\quad + \{dq(1., 1.5)/dq_7\}\Delta q_7. \end{aligned}$$

In the context of the derivative matrix, the effect of "shutting down" part of the network should hopefully be to introduce mostly zeroes or small numbers in the off diagonal terms. This would indicate that the

TABLE XIII

DERIVATIVE MATRIX FOR EQUILIBRIUM CONDITIONS
 $(a_{ij} = d[H_i(q) - q_i]/dq_j)$

$\begin{matrix} j \\ i \end{matrix}$	1	2	3
1	-1.6796	.2720	0.0000
2	.6796	-2.0590	1.3450
3	0.0000	.7869	-2.3450

equilibrium equations for the individual quantities could be found by separately iterating. Thus if some of the equilibrium equations were already satisfied, only those which were not would need to be considered.

In the example experiments performed, although each introduced zeroes, the derivative matrices were not completely diagonal. If both the price p_2 , and the demand quantities q_6 and q_7 were kept constant, then the 3×3 matrix would have five nonzero elements. There seems no way to extrapolate from this to the effects on big structured networks.

The results of these experiments are inconclusive in that it still remains an open question whether or not shutting down a portion of the network and iterating on separate parts will ultimately shorten the time required to find the equilibrium quantities. A tentative conclusion is that if anything could be shut down and have less effect on the final answers, it is the demand portion of the network. Shutting down the bottom portion affects quantities produced, ultimate top demand, and the market share portion. This can create an equilibrium far from the

true one, and even with derivative information to estimate the true solution can create inaccurate results.

1. Decomposition and Variable Reduction in the Dynamic Equilibrium Case

The LEAP model as currently implemented can be viewed as the solution of as many equilibrium equations as there are basic quantities produced *times* the number of time periods. Let M be the number of basic quantities supplied, and N the number of time periods. The quantities are then denoted as $q_i(t)$, and the equilibrium equations are

$$H_i(q, t) - q_i(t) = 0, \quad \text{for } i=1, \dots, M; t=1, \dots, N.$$

The number of simultaneous nonlinear equations to be solved is therefore $M \cdot N$, a very large number in the LEAP model. It is the purpose of this section to propose a reformulation of the basic variables of the model so that the number of variables is considerably reduced (and therefore the number of equations). The idea is to basically uncouple the system by doing away with the time dependence. The model used to test and explain these ideas is the two supplier two time period model discussed in Section A of Part I.

The reason that there is no natural uncoupling of the system of equations for the dynamic case within the current model is as follows.

In each of the N time periods there are M basic quantities to be supplied. The price, $p_i(t)$ of the i th quantity in time period t depends upon the marginal production cost plus economic rent. The marginal cost depends upon the cumulative production through (including) the production of the i th item to time period t . That is, the marginal

cost is a function of $\sum_{j=0}^t q_i(t)$. The economic rent is a function of *future* prices which includes amounts of quantity *i* produced through the final time period. This means that the price for item *i* in time period *t* depends upon all quantities $\{q_i(t), t=1, \dots, N\}$. Demand in this model depends upon the computation of an average price which depends upon all $\{p_i(t), i=1, \dots, M\}$. The result is that the average price in each time period depends upon *all* quantities produced for *all* items in *all* time periods. This average price gets passed up and the demand on top is then passed down via the market share formulas to give the equilibrium equations at the bottom. This means each equation depends upon *every* quantity produced in *every* time period.

Consider the example of Section IA2. In the table below is the 4×4 derivative matrix at equilibrium of this problem. Any hope at decomposing this network is doomed to fail since the size of the derivatives show no small numbers. This means the production of any quantity in any period has a significant impact on the equilibrium values.

TABLE XIV
MATRIX OF DERIVATIVES OF EQUILIBRIUM EQUATIONS

$q_1(1)$	$q_2(1)$	$q_1(2)$	$q_2(2)$
-0.208631D 01	0.970590D 00	-0.992322D 00	0.894833D 00
0.811743D 00	-0.220776D 01	0.766416D 00	-0.114324D 01
-0.138037D 01	0.122911D 01	-0.239407D 01	0.124229D 01
0.106340D 01	-0.156557D 01	0.104926D 01	-0.255196D 01

To avoid this, a simple device is used. The variables of the model are changed to be the *cumulative* amounts of basic quantities produced through (inclusive) *the final time period*. These are denoted by $\{Q_i(N), i=1, \dots, M\}$.

The formulation of the evaluation of the equilibrium equations now becomes movement up and down the network working backwards in time. This operates as follows.

Given the problem variables $\{Q_i(N)\}$, the basic supply costs are computed from the marginal cost curves and assumptions about beyond the horizon prices (this is exactly as is done in the current model). Using the device developed in Section IIB of this report, the prices of intermediate quantities can be passed up the network, assuming that demand is a function of current prices (and perhaps beyond the horizon prices), demand quantities are computed and then passed down the network to the bottom. Using the same notation as before, the basic supply quantities at the bottom are $H_i(N)$. The equilibrium equations at the bottom of the network in the final time period can now be easily satisfied by the simple device of setting

$$q_i(N) = H_i(N), \quad \text{for } i=1, \dots, M.$$

Once this is done, the cumulative production of the *i*th basic supply in time period $N-1$ is *derived* as

$$Q_i(N-1) = Q_i(N) - q_i(N), \quad \text{for } i=1, \dots, M.$$

This procedure is generalized inductively. Suppose the cumulative amounts of the basic supplies are known for time period *t*, and the *future* prices for all the quantities are also known. Then the price charged for

the basic supplies can be computed as the sum of the marginal cost (which is a function of future prices). These supply prices are then passed up the network in time period t . Demand at the top is computed and passed down to the bottom. These bottom quantities are $\{H_i(t)\}$. The bottom equilibrium equations in all but the initial time period are satisfied by letting

$$q_i(t) = H_i(t) , \quad \text{for } i=1, \dots, M .$$

The cumulative production through period $t-1$ is then *derived* as

$$Q_i(t-1) = Q_i(t) - q_i(t) , \quad \text{for } i=1, \dots, M .$$

This procedure terminates in the first time period, and the equations

$$H_i(1) - [Q_i(1) - Q_i(0)] = 0 , \quad \text{for } i=1, \dots, M$$

result. (Here $Q_i(0)$ represents cumulative production before the first time period which is input to the model.)

In this formulation, the quantities $\{H_i(1), Q_i(1)\}$ are now functions of the variables $\{Q_i(N)\}$. Thus the number of variables and equations has been reduced from $M \cdot N$ to M , the number of basic supply quantities.

To illustrate this, consider the two variable two time period example. The same numbers are used as in that example, and the variables are now the final cumulative demand. The reformulation is given here for completeness.

Given $Q_i(2)$, for $i=1, 2$, $Q_i(0)$ is taken to be zero in this example. Compute $m_1(2)$ (the marginal production cost)

$$= 1. + 0.5[Q_1(2)]^{2.0} ,$$

and

$$m_2(2) = .8 + .4[Q_2(2)]^{2.2} .$$

Compute the selling price as

$$p_i(2) = (1.1)m_i(2), \quad \text{for } i=1,2.$$

Compute the market share as

$$S_i(2) = f_i[p_i(2)]^{-\gamma} / \sum_{k=1}^2 f_k[p_k(2)]^{-\gamma}, \quad \text{for } i=1,2.$$

Compute the average price as

$$p_3(2) = \sum_{i=1}^2 p_i(2) S_i(2).$$

The demand (using a value of $\beta = 0$) is then

$$q_3(2) = 2.6[p_3(2)]^{-0.5}.$$

This is passed down via the market share formulas to give

$$H_i(2) = S_i(2)q_3(2), \quad \text{for } i=1,2.$$

The cumulative demand in the first period is then given as

$$Q_i(1) = Q_i(2) - H_i(2), \quad \text{for } i=1,2.$$

The prices, etc. in the first time period are computed exactly as above with (2) being replaced by (1) wherever it occurs. The only exception is that the demand in the first time period has the form

$$q_3(1) = 2.0[p_3(1)]^{-0.5}.$$

Coming down the network in the first time period yields the two equations

$$H_i(1) - Q_i(1) = 0, \quad \text{for } i=1,2$$

(since the $Q_i(0)$ were assumed zero).

This new formulation was set up in the factorable programming language and solved using Newton's method. The 2×2 matrix of derivatives of the reduced set of equilibrium equations is given in Table XV. The

TABLE XV

DERIVATIVES OF EQUILIBRIUM EQUATIONS

$$(a_{ij} = d[H_i(1) - \{Q_i(1) - Q_i(0)\}] / dQ_j(2))$$

$\begin{matrix} j \\ i \end{matrix}$	1	2
1	-3.5774	2.3175
2	2.1321	-4.0684

matrix is negative definite and resembles the derivative matrices given by the original formulation. The implications of this are being studied.

Solving the reduced set of equilibrium equations is now reduced to a problem of much less magnitude. In the following are some ideas about how an algorithm imitative of the current LEAP algorithm can be devised.

Instead of going up and down, the natural approach to solving the new equations is from right to left and back. Assume values $Q_i^k(N)$, for $i=1, \dots, M$ for the terminal cumulative production of the basic supplies are available. The up and down from right to left computations, as described previously, is performed. The results in the final values

$$H_i^k(1) - [Q_i^k(1) - Q_i(0)], \quad \text{for } i=1, \dots, M.$$

If these are all zero, then the equilibrium equations are all satisfied.

If they are not satisfied, then the following way of getting a new guess can be programmed.

Set

$$C_i^k(1) = H_i^k(1) + Q_i(0), \quad \text{for } i=1, \dots, M.$$

Then for $j=1, \dots, N-1$ (and for all $i=1, \dots, M$), set

$$C_i^k(j+1) = C_i^k(j) + H_i^k(j) .$$

At the N th time period, $\{C_i^k(N)\}$ are new estimates of the final cumulative supplies required based on the assumption that the supplies $H_i^k(j)$ are the true amounts in the j th period since they were calculated on the right to left pass. Then the variable values at the beginning of the iteration could be computed similarly to the way they are now, with

$$Q_i^{k+1}(N) = Q_i^k(N)(1-\omega) + C_i^k(N)\omega , \quad \text{for some value of } 0 < \omega < 1 .$$

Of course this is just one way to try to solve the reduced equilibrium equations. For the small test problems Newton's method has worked very well. The secant method, which just requires function values, has also performed well. The point is that with the reduction in variables, these algorithms may now be competitive with the SOR method.

2. An Interpretation of Decomposition

In this subsection, we describe a geometric interpretation of decomposition by means of the example illustrated in Figure 35.

In Section E of Part I, we described the notion of an "implied supply curve" at node 3, giving prices P_3 as a function of $Q_3 = q_1 + q_2$. The implied supply curve was generated and is shown on Figure 36. The actual generation of $P_3(Q_3)$ was carried out point by point, *fixing a value of $\bar{Q}_3$* , guessing q_1 and q_2 such that $q_1 + q_2 = Q_3$, computing the fractions

Figure 35.--Decomposition example.

$$f_i = \frac{(p_i)^{-2}}{(p_1)^{-2} + (p_2)^{-2}} , \quad (i=1,2)$$

and passing down q_1^{new} and q_2^{new} . Interpolating (as in SOR) between the guessed values of q_1 and the new values, gave a fresh starting point and this scheme was repeated until values $\bar{q}_1$ and $\bar{q}_2$ were found which satisfied

$$\bar{q}_1 + \bar{q}_2 = \bar{q}_3$$

and

$$\bar{f}_i \bar{q}_3 = \bar{q}_i .$$

Figure 36.--LEAP estimates of the implied supply curve.

Finally, with $\bar{q}_1$ and $\bar{q}_2$ so determined, $P_3(\bar{Q}_3)$ was computed by averaging $p_1(\bar{q}_1)$ and $p_2(\bar{q}_2)$ as in LEAP.

An *implied demand curve* can also be generated (see Brock and Nesbitt [2] for a description) by simple horizontal addition of the curves q_5 and q_6 . Note here that points on the demand curve are generated by *specifying a price* $\bar{P}_4$, and computing a quantity $\bar{Q}_4$ to send down. The implied demand curve is also illustrated in Figure 36.

We now interpret LEAP in terms of this example. Suppose we start with $q_1^1 = q_2^1 = 1$. The averaged price p_3 is plotted on Figure 36, and lies *below* the implied supply curve at $Q_3^1 = 2$. The implied demand curve is generated so simply in this example that the quantity q_4 demanded from p_4 is such that (p_4, q_4) lies exactly on the graph of the implied demand curve.

In this illustration, we took $\omega = 0.5$, which resulted in a new interpolated value (q_1^2, q_2^2) where $q_1^2 + q_2^2 = Q_3^2 = 1.78$. Then $P_3^2 = 1.70$, so that (Q_3^2, P_3^2) lies *above* the implied supply curve.

Data for this example are given in Table XVI and plotted (for the first three iterations) on Figure 36. Notice that oscillation is beginning to occur, due to the poor choice of ω .

As is evident from the figure, LEAP is producing points both above and below the true implied supply curve. Decomposition *could* be used here by "shutting down" the demand side of the network and iterating on the supply side in order to more accurately approximate the implied supply curve.

Note, however, at LEAP iteration 1, this would have yielded a *worse* estimate of the equilibrium point, whereas at iterations 2 and 3, better estimates would be obtained.

TABLE XVI
DATA FOR FIGURE 36

q_1	q_2	Q_3	P_3	$D_3(P_3)$
1.	1.	2.	1.859	1.552
.5925	1.183	1.776	1.708	1.642
.6711	1.038	1.709	1.448	1.836
.5617	1.211	1.773	1.769	1.604
.6869	1.002	1.688	1.404	1.874

Of course, SOR need not be used to sharpen up estimates of the implied supply curve. The equations characterizing this curve were presented in Section E of Part I, and Newton's method could be used.

In general, portions of the network can be "shut down" so that the remaining network can be more accurately estimated. It must be pointed out, however, that certain variables must be fixed at some value (quantities at the supply side, or prices at the demand side), and there is no assurance that more accurate estimates of the behavior of the active portion of the network will yield useful estimates when joined with the fixed variables.

REFERENCES

- [1] ADLER, J. R., E. G. CAZALET, *et al.* "The DFI Energy-Economy Modeling System." Decision Focus Report (draft), Decision Focus, Incorporated, Palo Alto, California, May 1978.
- [2] BROCK, H. W. and D. M. NESBITT. "Large Scale Energy Planning Models: A Methodological Analysis." Technical Report, Decision Analysis Group, Stanford Research Center, Palo Alto, California, May 1977.
- [3] CAZALET, EDWARD G. "Generalized Equilibrium Modeling: The Methodology of the SRI-GULF Model." SRI Report, SRI International, Menlo Park, California, May 1977.
- [4] COLLATZ, LOTHAR. *Functional Analysis and Numerical Mathematics*. Translated by Hansjorg Oser. New York: Academic Press, 1966.
- [5] COURNOT, A. *Researches into the Mathematical Principle of the Theory of Wealth*. English translation. New York: Macmillan, 1897 (1838).
- [6] FEDERAL ENERGY ADMINISTRATION. "Project Independence Report." Washington, D.C.: U.S. Government Printing Office, 1976.
- [7] FRIEDMAN, JAMES. *Oligopoly and the Theory of Games*. Amsterdam: North Holland, 1977.
- [8] GABAY, D. and H. MOULIN. "On the Uniqueness and Stability of Nash's Equilibrium in Non-Cooperative Games." Report No. 7833, Centre de Recherche de Mathématiques de la Décision, Université de Paris - Dauphine, 1978.

- [9] ORTEGA, J. M. and W. W. RHEINBOLT. *Iterative Solution of Nonlinear Equations in Several Variables*. New York: Academic Press, 1970.
- [10] OWEN, GUILLERMO. "Computation of Equilibrium Price and Quantity Vectors." In *Convex Analysis and its Applications*, pp. 170-179. Edited by M. Beckmann and H. P. Künzi. New York: Springer-Verlag, 1977.
- [11] QUIRK, JAMES and RUBIN SAPOSNIK. *Introduction to General Equilibrium Theory and Welfare Economics*. New York: McGraw-Hill, 1968.
- [12] RHEINBOLT, W. C. "On M-functions and their Application to Nonlinear Gauss-Seidel Iterations and to Network Flows." *Journal of Mathematical Analysis and Applications* 32 (1970): 274-307.
- [13] ROCKAFELLAR, R. TYRRELL. *Convex Analysis*. Princeton, New Jersey: Princeton University Press, 1970.