LOCKHEED MARTIN ENERGY RESEARCH LIBRARIES
3 4456 0510966 4

ORNL-5754

ornl

OAK
RIDGE
NATIONAL
LABORATORY

UNION
CARBIDE

# Mathematical Foundations of LEAP

E. M. Oblow

OAK RIDGE NATIONAL LABORATORY
CENTRAL RESEARCH LIBRARY
CIRCULATION SECTION
4500N ROOM 175
LIBRARY LOAN COPY
DO NOT TRANSFER TO ANOTHER PERSON
If you wish someone else to see this report, send in name with report and the library will arrange a loan.
UCN-7969 3 9-77

OPERATED BY
UNION CARBIDE CORPORATION
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
NTIS price codes—Printed Copy: A04; Microfiche A01

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.

ORNL-5754
Special

Contract No W-7405-eng-26

Engineering Physics Division

# Mathematical Foundations of LEAP

E M Oblow

Date Published - April 1981

OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37830
operated by
UNION CARBIDE CORPORATION
for the
DEPARTMENT OF ENERGY

LOCKHEED MARTIN ENERGY RESEARCH LIBRARIES
3 4456 0510966 4

# CONTENTS

ABSTRACT

I INTRODUCTION

II BASIC LEAP ALGORITHM

III OVERALL SCHEME

A ITERATIVE APPROACH

B UP/DOWN ALGORITHM

IV CONVERGENCE TESTS

V RELAXATION AND ACCELERATION

VI PRACTICAL EXPERIENCE WITH LEAP ALGORITHMS

VII CONCLUSIONS BASED ON MODEL 82 RESULTS

VIII ALTERNATIVE LEAP RELAXATION ALGORITHM - NEWTON'S METHOD

IX SUMMARY

REFERENCES

# ABSTRACT

As part of the ORNL energy model validation program an anlysis was made of the mathematical structure of the LEAP energy modeling code Emphasis was placed on understanding the equations which the code attempts to solve and the iterative numerical algorithm used to achieve a converged solution The equations were found to be highly nonlinear, similar in form to those found in equilibrium boundary value problems A combination of a standard nonlinear Gauss-Seidel approach and a modification of Newton's method was used as the basis of an iterative algorithm to find local solutions to the equilibrium equations Several weaknesses in this approach are discussed, mainly those dealing with the computation of relaxation coefficients in the Newton's method segment of the algorithm The chief failing of the method as employed in LEAP is the use of a diagonal Jacobian matrix for Newton's method with an ad hoc scheme for calculating the diagonal matrix elements Most problems encountered had large off-diagonal elements leading to a characteristic oscillatory breakdown in convergence It was concluded that the LEAP algorithm cannot be used to achieve convergence in an automated fashion for any general problem In practice hand intervention is needed in most cases Recommendations are made to approximate certain key elements of the Jacobian matrix to improve the general usefulness of the algorithm Analytic expressions for the approximate elements are derived so as to facilitate implementation in LEAP Other aspects of the numerical procedures are discussed, and recommendations for thier improvement are also made

# I INTRODUCTION

The Energy Information Administration (EIA) of the Department of Energy is sponsoring a program at Oak Ridge National Laboratory to develop and demonstrate a methodology for evaluating energy-economy modeling codes The recent increased use of such codes in the energy policy decision-making process makes the evaluation task a crucial one if these codes are to address important energy-related issues It is essential to know the scope and validity of model predictions before their usefulness in the policy arena can be justified

In conjunction with EIA's interest in long-range energy-use forecasting and their need to evaluate the impact of new technologies on the energy economy, the LEAP modeling[1,2] system was chosen for the initial evaluation effort at ORNL As part of this overall effort a small subtask was designated to explore the mathematical foundations of this code in order to better understand the nature of the solutions the code gives to various modeling problems This report focuses on the numerical aspects of the solution of the equations which constitute the LEAP system Primary interest is attached to the iteration schemes used and their convergence properties A future report[3] will explore the uniqueness and existence of LEAP solutions and describe a more in-depth analytic analysis of the iteration algorithm

Much of the work in this report is understood best only in the context of the rest of the ORNL model evaluation program Information relevant to this and all other tasks in the program can be found in the iterim progress report[4] and several topical reports comprising the project's

final report [5,6] Descriptions of several important models upon which this present study is based are dealt with in more detail in these other reports

## II BASIC LEAP ALGORITHM

LEAP[1,2] comprises a coupled set of nonlinear, time-dependent equilibrium equations in prices and quantities, which in general terms can be written as

$$\overline{N}(\overline{p},\overline{q}) = 0 \ , \tag{1}$$

where $\overline{N}$ is a nonlinear algebraic vector function and $\overline{p}$ and $\overline{q}$ are vectors representing prices and quantities at discrete time points, respectively, for all processes in the model-energy economy The basic code is modular in structure, with generic processes (i e , energy conversion, allocation, etc ) used to construct a network of interrelated energy-economy activities These processes are linked to each other in the economy through the time-dependent prices and quantities of energy related products The vector $\overline{N}$ has dimensions equal to two times the number of time intervals (typically ten), times the number of process links in the network (typically several hundred), for a total dimensionality of the order of several thousand in a modeling system such as LEAP Model 22C [6,7]

Due to the nature of the basic conversion process in LEAP, the time structure of Eq (1) is such that there are time intervals before and after the time period defining the solution space The prices and quantities in these intervals are specified exogenously or are modelled very simply and therefore act as boundary values for the problem Equation (1) can also therefore be characterized as a nonlinear boundary value problem

Becuase of the nonlinear nature of $\overline{N}$ and its large dimension in LEAP (i e , many processes are used to construct a typical network), the choice of solution method for such a problem is limited The authors of the code judged that a conventional iterative numerical scheme, the Gauss-Seidel/ Successive Overrelaxation (SOR) method[8] was the most practical alternative under the circumstances Due to other considerations, however, the method actually implemented in the code is not strictly a Gauss-Seidel/SOR algorithm, but a combination of closely related standard iterative procedures This algorithm requires a starting guess for $\overline{p}$ and $\overline{q}$ and a means for calculating the relaxation coefficients needed to update the solutions obtained during the iteration procedure

From these brief comments, it should be clear that the LEAP algorithm can be classified as a "locally' convergent scheme That is, it is only designed to find a solution, if one exists, near the initial vectors $\overline{p}$ and $\overline{q}$ used to start the iteration procedure The details of this approach are discussed in more detail in later sections It is only important here to recognize that local convergence is being sought and that the local nature of the initial values of $\overline{p}$ and $\overline{q}$ are crucial to the practical success of the algorithm In addition although some difinitive work has been done to explore questions of existence and uniqueness of LEAP-type solutions,[3,9,10] the answers to these questions can still only be given in a general sense When faced with a large energy model like Model 22C, it is clearly more difficult to characterize the solutions which arise from the LEAP iteration algorithm For the purposes of this report the assumption by the authors of the code and its user, that the solutions

which arise are unique and those being sought by the algorithm exist, is taken as a starting point for this analysis The fact that conditions exist in LEAP which lead to nonexistence or non-uniqueness of these solutions are dealt with in more detail in a later report [3]

## III ITERATIVE APPROACH

### A OVERALL SCHEME

To meet the generality and complexity of any proposed LEAP energy model the solution algorithm was chosen by the authors of the code to closely resemble some physical attribute of the problems modeled by LEAP That is, the LEAP algorithm attempts to model the equilibration of supply and demand driven by consumers and producers in a Walrasian-type energy economy [11] This procedure gives rise to the familiar cobweb iteration scheme,[11] in which prices are offered by producers, consumers respond with an appropriate level of demand, which in turn causes producers to change their prices accordingly This iterative procedure was chosen after the basic network and modular process structure of the code were developed The fact that each modular process was chosen to be either a basic supply or demand unit of the nergy economy model lead naturally to the choice of the cobweb-type iteration scheme

As employed in LEAP this cobweb scheme is referred to as the up/down algorithm, in reference to the fact that on the up pass of the algorithm producers offer prices and on the down pass consumers respond with demand As pointed out by Falk et al ,[9] LEAP is not formulated as an explicit optimization problem in economics, although other documentation associated with LEAP[11] does contain considerable discussion about the connection between supply-demand equilibration and optimal solutions to economics

problems Numerical procedures suitable for optimization problems were therefore not used in the code The up/down algorithm however was only one of many approaches initially tested in the code before its formal release[12] and therefore represents a practical compromise to the LEAP solution problem

B UP/DOWN ALGORITHM

The specific characteristics that lead to the use of the up/down algorithm are the fact that each modular generic process used to construct a LEAP network is a natural supply-demand unit linked with other process units in a supply and demand relationship This structure as implemented in LEAP also has the desirable property that it can be described mathematically by separable supply and demand transformation functions in each generic process That is, on the supply side a generic process can be written as,

$$\overline{p}_k = \overline{G}_k(\overline{q}_k, \overline{p}_{k'}, \overline{q}_{k'}) \qquad k' \neq k \tag{2}$$

and likewise on the demand side,

$$\overline{q}_k = \overline{H}_k(\overline{p}_k, \overline{q}_{k'}, \overline{p}_{k'}) \qquad k' \neq k \tag{3}$$

where $\overline{p}_k$ and $\overline{q}_k$ are the time-dependent components of $\overline{p}$ and $\overline{q}$, respectively, in the $k^{th}$ generic process, $G_k$ and $H_k$ are the $k^{th}$ process supply and demand transformation functions, respectively, and $k'$ represents other processes whose quantities $q_{k'}$ are used in the $k^{th}$ one In the basic conversion process this structure is somewhat more complex in that $k'$ and $k$ also represent different time-dependent components internal to the $k^{th}$ process,[13] but again because of certain approximations, $k' \neq k$

This overall structure of separability and supply-demand unit linking defines an up/down procedure in which all process prices are solved for in an up pass, given the transformation functions $\overline{G}_k$, and all process quantities are solved for in the down pass, given the transformation functions $\overline{H}_k$ It also allows the entire up pass to be written more compactly as

$$\overline{p} = \overline{G}(\overline{q}) \tag{4}$$

and the entire down pass as

$$\overline{q} = \overline{H}(\overline{p}) \tag{5}$$

where $\overline{G}$ and $\overline{H}$ are the product of the successive operations of supply and demand process transformation functions, respectively

The above results allow the equations solved by the entire code to be represented in terms of either prices alone or quantities alone In terms of quantities then the code can be written as

$$\overline{q} = \overline{F}(\overline{q}) = \overline{H}[\overline{G}(\overline{q})]$$

Note that in this format, denoted as a fixed point form,[8] if the input quantity vector to the up pass of the $n^{th}$ iteration is given by $\overline{q}_n(I)$, then Eq (6) can be written in iterative form to calculate an output quantity vector $\overline{q}_n(0)$ on the down pass as follows

$$\overline{q}_n(0) = \overline{F}[\overline{q}_n(I)] \tag{7}$$

This symbolic form of the LEAP equations will be useful in analyzing the convergence properties of the iteration algorithm

The actual calculation scheme as described in Eqs (2) and (3) is such that in executing an up (or down) pass in the network, the code steps from process module to process module calculating the prices (or quantities) in each basic process module for all time steps Where the process module equations are non-linear in time, or the k components of $\overline{p}$ or $\overline{q}$ are not yet available, the initial guesses for $\overline{p}$ (or $\overline{q}$) are used to begin the calculation in the first iteration

The specific LEAP algorithm chosen by the authors uses the results of each process module calculation immediately in the next adjacent process module calculation This immediate update procedure in the midst of an iteration pass wthout modification of the calculated value is what characterizes the algorithm as a standard Gauss-Seidel iterative approach to nonlinear equations [14,15] The procedure is locally convergent with a linear rate of convergence which is guaranteed only if the Jacobian of $\overline{F}$ is negative dominant diagonal in the vicinity of the solution

The Gauss-Seidel scheme can be modified slightly by the inclusion of a relaxation parameter to alter the output price (or quantity) of each module calculation for use in the next module This addition of relaxation for the calculated results is the classic nonlinear successive overrelaxation (SOR) method,[15] an iterative procedure with characteristics similar to the Gauss-Seidel method As will be seen later, the SOR approach is also used in LEAP but is limited to only certain process modules in any given LEAP network, leaving the Gauss-Seidel method as the major constituent of the algorithm

To complete the general iterative procedure, a visitation sequence is defined for the various processes that make up the network structure so that in any pass (up or down), the algorithm passes through each process

once and only once In principle, the visitation sequence used in conjunction with this iteration scheme can be arbitrary, albeit at the expense of the process network being modeled and the form of Eq (6) In practice, therefore, it is useful to order this sequence to follow the flow of supply to demand on an up pass and the flow of demand to supply on a down pass so as to preserve the supply-demand network structure Thus, the conventional LEAP procedure is to start the iteration process at the bottom (supply side) of the network and proceed to the top (demand side) Because this procedure approximates the supply-demand equilibration process in a classical Walrasian economic sense it has the potential to increase the speed of convergence of the Gauss-Seidel iteration scheme in those cases where the algorithm is guaranteed to converge

Note that, in general, if the unknown price and quantity vectors and their associated equations (i e , Eq (7)) are ordered properly by the visitation sequence, the matrix operator $\bar{F}$ will block upper-triangular for prices on the up pass of the iteration sequence and block lower-triangular for quantities on the down pass Each block represents a process module, with time-dependent prices or quantities coupled to each other inside the module block This block format makes the procedure for calculating all prices in a given process dependent only on processes appearing below it in the visitation sequences on the up pass and on processes appearing above it on the down pass The Gauss-Seidel algorithm is well suited for such a decomposed system of equations

## IV CONVERGENCE TESTS

The criteria used to test convergence of this algorithm should be noted at this point Two different criteria for evaluating the convergence of the results need to be discussed here, (1) the one described in the DFI documentation of LEAP[1] and (2) the one in used in the algorithm at EIA[2] to generate Model 22C results The two versions use different norms for testing convergence The following comments describe both the DFI documented version of LEAP and the EIA version Attention will be drawn to the specific differences in the norms used for convergence testing as needed

The state of convergence of the solution during the iteration procedure must be checked to determine a stopping point for the algorithm The LEAP convergence test is performed *only* in allocation processes and *only* on the down pass of the algorithm when quantities are being calculated In each allocation process, a pointwise convergence check of the components of the quantity vector $\overline{q}_k$ in the $k^{th}$ allocation process is therefore made Again, it should be emphasized that these are the *only* quantities checked for convergence in the whole algorithm

The rationale for this scheme of testing convergence only in allocation processes is unclear from both physical and theoretical viewpoints Basically, it is a heuristic approach derived from practical experience with the code [12] The rationale appears to be that most quantities in a general model network link directly to some allocation process and these allocation processes are sufficiently dispersed in various sectors of the network so that they are decoupled Allocation quantity convergence would therefore generally represent a good test of overall convergence of the

other quantities and prices in the model This test can lead to complications in problems where prices are very weakly coupled to allocation quantities, yielding poorly converged price results in conjunction with well converged quantities The success of this approach clearly depends on the particular network structure under investigation

The test used in the DFI documented algorithm is of the form,

$$\left| \frac{q_{n,i}^{(0)} - q_{n,i}^{(I)}}{q_{n,i}^{(0)}} \right| < \varepsilon \quad \text{for all } q_i \in \overline{q}_k \tag{8}$$

where the subscripts n and i refer the $n^{th}$ iterate of the $i^{th}$ quantity component $q_i$ of the vector $\overline{q}_k$, the superscripts I and 0 refer to input and output of the $n^{th}$ iteration, and $\varepsilon$ is the exogeneously supplied convergence criteria In practice, the convergence test represented by Eq (8) is only used when the absolute value of $q_{n,i}^{(0)}$ is greater than 0 001 quad Below this value, <u>no check</u> is made of convergence

In the EIA version of LEAP that produced the Model 22C results, the convergence check is of the following form,

$$\left| \frac{q_{n+1,i}^{(I)} - q_{n,i}^{(0)}}{q_{n,i}^{(0)}} \right| < \varepsilon , \quad \text{for all } q_i \in \overline{q}_k \tag{9}$$

where again $\varepsilon$ is the exogenously supplied convergence criteria and $q_{n+1,i}^{(I)}$ is the input quantity to the $n+1^{st}$ iteration

Theoretically, the choice of either of these forms of allocation quantity convergence testing leads to the second major characteristics of the iteration algorithm - it is an incomplete pointwise convergence

testing scheme The procedure is a pointwise scheme because it checks the convergence of the components of the solution price and quantity vectors It is an incomplete procedure because it only tests the convergence some of the components of $\overline{q}$ (i e , the time components, $q_k$, in each allocation process), and none of the other prices and quantities in the model It is true that allocation quantity convergence implies some global convergence of all other quantities and prices in the code through the transformation functions $\overline{F}$, $\overline{G}$, and $\overline{H}$ Convergence in allocation quantities, however does not imply equal or better convergence of all the other solution prices and quantities in the code A careful examination of the state of convergence of all other prices and quantity results for Model 22C reveal that this problem exists although not on a scale that is significant in this problem (i e , only a handful of quantities are not fully converged in Model 22C)

Despite the fact that relaxation has not been discussed as yet, it is necessary here to point out that the specific norm discussed in the DFI documentation for checking convergence (i e , Eq (8)) is a poor norm for convergence testing given the specific relaxation algorithm employed in LEAP allocation processes Results discussed at length later will show that use of Eq (9) leads to convergence of some quantities in Model 22C to within $\varepsilon/(1 - \alpha_i)$, where $\alpha_i$ is the relaxation parameter used for the $i^{th}$ allocation quantity component to get $q^{(I)}_{n+1,i}$ from the values $q^{(I)}_{n,i}$ and $q^{(0)}_{n,i}$ Because $\alpha_i$ was about 0 5 for most of these quantities, the reported converged results for these quantities actually satisfy an $\sim 2\varepsilon$ convergence criteria (i e , about 6% in Model 22C) During the analysis of Model 22C results, this test was changed at ORNL from Eq (9) to Eq (8) and reruns of Model 22C with true convergence to $\varepsilon$ showed a change of less than a

few percent in all major results Only a few quantities with very small magnitudes seem to have converged to only $2\varepsilon$ as predicted by this analysis Therefore, in this instance no major problem resulted from using Eq (9) Other problems run with this version of the code, however, should be checked because $\alpha_1$ can potentially be close to unity, and differences in convergence as large as a factor of ten are possible (i e , a 3% convergence criteria could actually be in error by as much as 30% in some quantities) The change from Eq (9) to Eq (8) has now also been made in the current version of LEAP used at EIA and will therefore be the test used in all future reported results from EIA

## V RELAXATION AND ACCELERATION

During the iteration procedure, a decision is made to improve the results of any calculated price or quantity The general mathematical relationship used for this update procedure is of the form

$$\bar{q}_{n+1}^{(I)} = \bar{q}_n^{(I)} + A^{-1}[\bar{q}_n^{(0)} - q_n^{(I)}], \tag{10}$$

where A is, in general, a matrix of relaxation parameters, and the I and 0 superscripts refer again to the input and output of the $n^{th}$ iteration Depending on how the elements of the full A matrix are calculated (to be discussed in the next section), this general form describes a class of Newton or Generalized Secant methods [15] They are, again, locally convergent schemes, whose rate of convergence can be linear to quadratic[15] in principle, depending on the choice of A

The two special forms of Eq (10) used in LEAP are the Gauss-Seidel and Successive Overrelaxation cases For the SOR case, the matrix A is

assumed to be diagonal so that the components of $\bar{q}_n$(i e , $q_{n,i}$) are relaxed individually by a relaxation coefficient, $\alpha_i$ as follows,

$$q^{(I)}_{n+1,i} = q^{(I)}_{n,i} + \alpha_i(q^{(0)}_{n,i} - q^{(I)}_{n,i}) \tag{10a}$$

In the Gauss-Seidel version, Eq (10a) is used, but modified further so that $\alpha_i \equiv 1$ and $q^{(I)}_{n+1,i} = q^{(0)}_{n,i}$ This updated component is also used immediately in all subsequent calculations internal to the $n^{th}$ iteration The vector $\bar{q}_n^{(I)}$ used in Eq (7) at the stage in the $n^{th}$ iteration when the $i^{th}$ component of $\bar{q}_n^{(0)}$ is being calculated, therefore has the following form in the Gauss-Seidel scheme

$$\bar{q}_n^{(I)} = \{q_{n,J}\} \quad \text{with} \quad q_{n,J} = \begin{cases} q^{(I)}_{n+1,J} = q^{(0)}_{n,J} & J < i \\ q^{(I)}_{n,J} & J \geq i \end{cases} \tag{10c}$$

The choice by DFI of the particular theoretical form of Eq (10) to be implemented in the LEAP code is varied, depending on which particular process module and which component of the price and quantity vector is being calculated In general, for all prices and quantities in modules other than allocation, the Gauss-Seidel version of Eq (10) is used In all allocation processes, the Gauss-Seidel scheme is used for all prices and a combination of the Newton and SOR schemes (i e the more general Eq (10) and the specific Eq (10a)) is used for allocation quantities In total, then the LEAP relaxation algorithm is a peculiar mixture of basic theoretical schemes, which, as will be seen later, makes it somewhat difficult to analyze

Looking at this algorithm as it is actually implemented in the LEAP code,[2] it becomes clear that further ad hoc modifications of the above schemes have been made In one particularly important case, each allocation process only deals with the allocation of the fraction of the total quantity demanded (or supplied), and relaxation is applied to these fractions, called market shares, rather than to the actual quantities themselves This apparently minor modification of the relaxation algorithm, however, complicates the analysis of relaxation considerable because market shares are coupled by behavioral lag coefficients in time These coefficients cause market share changes from one year to the next to be damped, thereby allowing the market to respond only incompletely to time changes in demand (or supply) The net effect of such damping is that, when relaxation is applied in each time period (i e , to each component of the allocation quantity vector), a time lag couples the earlier time period relaxation coefficient to the present time period coefficient This characteristic makes the relaxation parameters a function of the behavioral lag coefficients, complicating the algorithm considerably

In addition to this difference, the actual LEAP algorithm[2] also relaxes a particular variable which in internal to certain modules in the code That is, LEAP relaxes new capacity additions in all conversion processes This choice was based on observations in several practical problems of wide fluctuations in new capacity additions between iterations [12] The modelers chose to damp these oscillations and hopefully improve the convergence properties of the algorithm The particular relaxation method chosen for this task was the simplest ad hoc scheme available That is, the new capacity addition vector $N_W$ in each

conversion process is relaxed by a constant, $\alpha$ For each of the time component $N_j$ of the new capacity addition vector $\overline{N}_W$ in each conversion processes module, a relaxed component is computed from,

$$N^{(I)}_{n+1,J} = \alpha N^{(O)}_{n,J} + (1 - \alpha)\, N^{(I)}_{n,J}$$

$$= N^{(I)}_{n,J} + \alpha\left[N^{(O)}_{n,J} - N^{(I)}_{n,J}\right] , \qquad (11)$$

where I and O again stand for the input and output of the $n^{th}$ iteration In the EIA version of LEAP used for Model 22C,[2] $\alpha$ = 0 7 for all components of $\overline{N}_W$ Note that although $\overline{N}_W$ is relaxed it is never tested for convergence since it was considered to be a dependent variable by the modelers

At this point, again it must be noted that the DFI LEAP documentation[3] and the code that generated Model 22C results[2] differ significantly insofar as implementing relaxation in the allocation process modules of LEAP The DFI LEAP documentation describes a Newton-Secant method for updating the quantity vector in allocation processes in which it is implied that the full A matrix can be evaluated by using the partial derivatives of $\overline{F}$ with respect to $\overline{q}$ (i e , $\partial\overline{F}/\partial\overline{q}$ the Jacobian of $\overline{F}$) Due to the difficulties in obtaining these derivatives analytically for a code as complex as LEAP, numerical approximations are used instead Since the only quantity vectors available in allocation processes are $\overline{q}^{(I)}$ and $\overline{q}^{(O)}$ for the current previous iterations, a discrete-secant approximation to the partial derivatives of $\overline{F}$ is employed The Newton approach in its discrete-secant form (i e , the Newton-Secant method) is defined by

$$A \equiv I - \left(\frac{\partial \overline{F}}{\partial \overline{q}}\right) , \tag{12}$$

where I is the identity matrix, the $i,j^{th}$ component of the Jacobian of $\overline{F}$ is given by

$$\left(\frac{\partial \overline{F}}{\partial \overline{q}}\right)_{i,j} \cong \left(\frac{\Delta \overline{F}_n}{\Delta \overline{q}_n^{(I)}}\right)_{i,j} = \frac{\Delta q_{n,i}^{(0)}}{\Delta q_{n,j}^{(I)}} \tag{13}$$

and $\Delta q_{n,i} = q_{n\ i} - q_{n-1\ i}$

Since all the input components of $\overline{q}_n^{(I)}$ change from iteration to iteration, the change in $\Delta \overline{q}_n^{(o)}$ caused by the $q_{n,j}^{(I)}$'th element of $\Delta \overline{q}_n^{(I)}$ cannot be practically evaluated in LEAP The modelers therefore approximated the full matrix by assuming that each $q_{n,j}^{(I)}$ component acts independently on only $q_{n,j}^{(0)}$ This approximation reduces A to a diagonal matrix, giving rise to the following relationship for the relaxation coefficient $\alpha_i$ in terms of the diagonal elements $a_{ii}$ of the A matrix

$$\alpha_i^{-1} \equiv a_{ii} \cong \left[1 - \left(\frac{q_{n\ i}^{(0)} - q_{n-1,i}^{(0)}}{q_{n,i}^{(I)} - q_{n-1,i}^{(I)}}\right)\right] \tag{14}$$

The DFI documentation, therefore, describes a nonlinear SOR scheme for relaxation of allocation process quantities [i e , Eq (10a) is used] with a relaxation coefficient [i e , Eq (14)] derived from the diagonal elements of the Jacobian of $\overline{F}$ used in a Newton-Secant formalism In this case the success of this portion of the algorithm will depend critically on the behavior of the off-diagonal elements of the matrix A

Since allocated quantities are always directly coupled to one another in any given year, this hybrid Newton-Secant/SOR method can be expected to encounter some difficulties in converging any general LEAP modeling problem This problem can be ameliorated to some degree by imposing limits on the maximum absolute value of $\alpha_i$ as is done in the norm reducing version of Newton's method [15] In this modified approach, explored by Falk et al [9] on a LEAP-type model, it is possible from a theoretical point of view to achieve convergence to an accuracy $\varepsilon$, which is a function of the maximum absolute value of $\alpha_i$ That is, convergence achieved on the $n^{th}$ iteration, $\varepsilon_n$, is a function of the form $\varepsilon_n = \varepsilon_n(\alpha_u)$, where $\alpha_u$ is the limit imposed on the maximum value of the $\alpha_i$'s In practice, restricting the upper limits to very small values, however, usually leads to an inordinate number of iterations to achieve convergence to a specific $\varepsilon$ This is a result of $\bar{q}_{n+1}^{(I)}$ being only slightly different from $\bar{q}_n^{(I)}$ from one iteration to the next, making it possible, for example, to take several thousand iterations to achieve 1/ convergence with $\alpha_u$ = 0 01 As a compromise then, the EIA version of LEAP uses a maximum upper limit of $\alpha$ = 0 9

A further major modification of the approach described above was also used in the EIA version of LEAP[3] used to generate the results for Model 22C As in the DFI documented version the EIA version uses a diagonal A matrix for updating $\bar{q}_n^{(I)}$ in each allocation process The EIA algorithm, however uses the following formula to calculate $\alpha_i$,

$$\alpha_i^{-1} \equiv \left[ 1 - \left( \frac{q_{n-1,i}^{(I)} - q_{n,i}^{(0)}}{q_{n-2,i}^{(I)} - q_{n,i}^{(I)}} \right) \right] , \tag{15}$$

with a condition that $0.1 \leq \alpha_1 \leq 0.9$. If $\alpha_1$ is outside this range, a fixed upper or lower limit (whichever the case may be) rather than the computed $\alpha_1$ is used. Thus, $\alpha_1$ is always positive and less than unity.

The theoretical origin of this approach is unknown, and EIA personnel have not given a rationale for its choice. The approach appears to be a scheme for finding a sequence of $q_{n,1}^{(I)}$ values as a function of n that can be used for extrapolation purposes to determine $q_{n+1,1}^{(I)}$ in a three-point Newton-Secant scheme.[15] Even if such a method was being used, the form of Eq. (15) does not fully conform to any known theoretical technique. This approach appears to be heuristic and one that happens to work to a limited extent on a limited class of problems.

Further analysis here is not warranted because of the ad hoc nature of the scheme. A description of actual experience with this algorithm in some specific LEAP problems is more fruitful and will be discussed in the next section. It is only worth conjecturing here that the use of a diagonal A matrix with constraints on the $\alpha_1$ values and their calculation using Newton-Secant type formulas are inconsistent. As will be discussed later in Section VIII, the nature of the equations in the allocation process do not lend themselves to the constraints imposed or the nonlinear SOR method in general and these inconsistencies are the potential failings of the algorithm.

As a last point it should be noted that an algorithm using Eq. (15) as its basis must use a different scheme for calculating $\alpha_1$ on the first three iterations (i.e., the first iterations of a "cold start" which only has $\bar{p}$ and $\bar{q}$ guesses available to begin the iterative procedure). Since no previous iteration results are available on the first iteration to compute relaxation coefficients, the EIA version of LEAP sets all of

these allocation quantity relaxation coefficients equal to a constant $\alpha_1 = 1\ 0$ This constant, can be fortuitously correct or completely wrong for the problem at hand leading to an initial divergence of the scheme because of the particular behavior of $\overline{F}$ in Eq (7) near the solution point

## VI PRACTICAL EXPERIENCE WITH LEAP ALGORITHMS

When the project began, little was known about LEAP and its numerical algorithm except that Model 22C results appeared to have converged to within 3/, and the output looked reasonable Since that time, an analysis of the results of a smaller test system, Model 82 [4] has produced more insights into the code's iterative algorithm This smaller model has all the essential processes contained in Model 22C but is simple enough to study in some detail through actual runs These runs can be used indirectly to comment on the expected impact on Model 22C of the algorithms described in Section V

Regarding actual degree of convergence, the use of the EIA convergence criteria [i e , Eq (9)] in Model 22C runs lead, in principle, to an overstatement of the level of accuracy (i e , $\varepsilon$) achieved in the reported results Therefore, as a first step in checking Model 22C results, the convergence test was changed to that given in Eq (8) and the code was rerun with $\varepsilon$ = 3/, beginning with the "converged" results reported in the EIA 1978 *Annual Report to Congress*[7] and all derivative information After eight additional iterations the new modified convergence test was satisfied, and the results were printed Generally, the new results differed by less than a percent from the *Annual Report* numbers Thus, no major discrepancy exists between a solution truly converged to 3/ and the results published in the *Annual Report*

The explanation of this result is that the quantities with the largest magnitudes in Model 22C were already converged to well below 3% in the original EIA run and only quantities with small magnitudes were near the 3/ criteria Changing the convergence algorithm simply resulted in these latter quantities converging to the true 3% criteria instead of the effective 6/ criteria resulting from the use of Eq (9) with $\alpha_1 \cong 0\ 5$

Note at this point that the mode in which LEAP runs are ended (i e , stopping the iterative process at convergence and then making one more iteration from a "cold start" to get printed results in the form of a "data report") can adversely alter the convergence achieved by the code Restarting a problem from an already converged solution, does not continue the iteration process from its termination point Instead, it begins the procedure over again, using only the solution values themselves as a starting guess (i e , all previous iteration information is lost in a restart) As mentioned before, a relaxation coefficient of $\alpha_1 = 1$ for all allocation quantities can be theoretically incorrect depending on the behavior of $\overline{F}$ near the solution point In effect the additional data report iteration could diverge the converged results already achieved This is a poor procedure and one that should be changed in future versions of the code

The above comments are the only ones that can be made directly about the LEAP algorithm concerning Model 22C Running Model 82, however, generated a better understanding of this algorithm In Model 82, the runs are usually only a few minutes of CPU time versus an hour or more in Model 22C and changes in the algorithm can readily be explored

It was, in fact, the attempt to run Model 82 to better than 5/ convergence (i e , $\varepsilon = 5 \times 10^{-2}$) for screening work[4] that provided the first true understanding of the characteristics of the algorithm for this type of problem Many small perturbations had to be made in the model parameters for screening, and reruns had to be made to get the perturbed results The algorithm in the code was never able to converge these problems to better than about 5%, dispite the fact that a converged unperturbed case was used to start all these cases The characteristic breakdown in convergence was always of an oscillatory nature as opposed to complete divergence

In order to determine the cause of the oscillatory breakdown in convergence in Mdoel 82, several numerical experiments were performed in which the assumptions inherent in the conversion and allocation process portions of the relaxation algorithm were changed All tests were conducted using starting guesses for the solutions of Model 82 which were close to being converged (i e , within the convergence of 5/ already achieved by the oscillating algorithm)

The first change that was made was to alter the algorithm to relax quantities as opposed to market shares This change was made to prevent the "relaxation" of the relaxation coefficients by the behavioral lag parameter in allocation quantities in future years The net effect of this change was still an oscillatory breakdown at about the 5/ level, but a shift in the year in which the maximum error occurred to earlier model years as opposed to those near the model horizon

The next change made in addition to the one above was the replacement of the EIA formula for calculating the re axation coefficient

[i e , Eq (15)] with the DFI formula based on the Newton-Secant method [i e , Eq (14)] This change was made to strengthen the theoretical basis for analyzing the algorithm At the same time the fixed upper and lower limits on the values of the relaxation coefficients were systematically varied to include negative lower limits ($\alpha_1 \leqslant -50$) and large positive values ($\alpha_1 \leqslant 50$) In no case did the results with these modifications effectively alter the oscillatory breakdown of the scheme in the few percent range

Several additional runs were then made using different constant relaxation coefficients (from $1 \leqslant \alpha \leqslant 9$) in place of the $\alpha = 7$ value in the conversion process portion of the algorithm [i e , Eq (11)] The same oscillatory results persisted in these cases also

At this point in the analysis it was conjectured that the Model 82 results might not be converging because multiple solutions might exist for this specific modeling problem with solutions separated in price and quantity values by a few percent An analytic examination of the allocation quantity and price equations was made to determine the conditions needed for uniqueness and it was shown that multiple solutions can indeed exist for allocation process prices and quantities [3] The conditions for multiple solution existence (the rare combination of at least one nearly elastic and one nearly inelastic supply curve to be allocated and a nearly elastic demand curve into allocation), however, were not present in Model 82, so this possibility was ruled out

A second theoretical explanation was derived from the work of Falk et al [9] on LEAP type equations In their work they observed an oscillatory breakdown in convergence exhibiting an asymptotic oscillation between two

limit points in the solution The oscillations exhibited a correlation between the maximum absolute value of the relaxation coefficient, $|\alpha|_{max}$, and the separation of the limit points A decrease in the allowed $|\alpha|_{max}$ reduced the limit point separation, with a resulting convergence in the limit of $|\alpha|_{max} \to 0$, to a single limit point solution

Since no prescription was given for reducing $|\alpha|_{max}$ and the fact that a small $|\alpha|_{max}$ causes an excessively large number of iterations to achieve convergence, an automated scheme for reducing $|\alpha|_{max}$ as a function of the level of convergence achieved in the $n^{th}$ iteration, $\varepsilon_n$, was devised to test this theory of Model 82 Results for this series of test runs as well as the formula for $|\alpha|_{max}$ as a function of $\varepsilon_n$ are summarized in Table I These results show again that the level of convergence can only be minimally improved by staging the maximum value of $\alpha_1$ down systematically as a function of $\varepsilon_n$ and then only at the expense of the number of iterations required The solution in each case still oscillated after the best converged values were achieved, and the improved best error values were smaller than the non-staged algorithm by, at most, only a factor of two

Table I Best Maximum Error Achieved in Less than 100 Iterations with Various $\alpha$ Staging Algorithms

| | $\lvert\alpha\rvert$ max = | $\sqrt{\varepsilon_n}$ | $\varepsilon_n$ | $\sqrt{\varepsilon}/4$ |
|---|---|---|---|---|
| Best maximum error | $\varepsilon_N$ | 0 05 | 0 05 | 0 02 |
| Number of iterations | N | 48 | 53 | 100 |

As a last resort, a complete analysis of the local partial derivatives for each allocation quantity was made using the calculated $\bar{q}$ results of previous and present iterations near the 5/ converged solution achieved As a result of this analysis it was concluded that a small positive $\alpha$ value should produce convergence for most of the supply-demand curve slopes encountered in this modeling problem A series of constant values of $\alpha$ for all quantities in all time steps was therefore tried on this problem in order to achieve convergence Surprisingly, a value of $\alpha$ = 0 2 yielded convergence to arbitrary accuracy in all perturbed cases of Model 82 Typically, $\varepsilon = 5 \times 10^{-4}$ was achieved in about 50 iterations for all starting guesses, which were small perturbations of the initial base-case quantities Other $\alpha$ values appeared to be converging the results well below 5% but at the expense of the number of iterations, which in most cases was several hundred or more to get below $\varepsilon$ = 0 005

These results were clearly unexpected and seem to go counter to the theoretical analysis previously done on this problem A constant $\alpha$ reduces the allocation portion of the LEAP algorithm to a pure nonlinear SOR scheme and it appears that the rather restrictive conditions under which this scheme is convergent exist in the Model 82 network In addition to this, a constant value of $\alpha$ also has the desirable property of not changing the relationship between total quantities (i e , the sum of the $q_{n,i}$'s) and individual components thereby retaining the property that supply equals demand before and after relaxation

It should be noted here that most of the $\alpha_i$'s for the largest allocated quantities in the EIA algorithm were found to be about 0 5, and this

fortuitous result seems to produce most of the convergence down to the level of $\varepsilon \cong 10^{-2}$ observed in the original Model 82 results The use of nonuniform $\alpha_1$ values seems to result in small quantities being relaxed so as to cause divergence in those quantities with large magnitudes after these latter quantities have by and large converged quickly to well below $\varepsilon \sim 10^{-2}$

## VIII CONCLUSIONS BASED ON MODEL 82 RESULTS

The conclusion that should be drawn from the results presented above is that the EIA-LEAP algorithm[2] is a nonconvergent scheme, in that it cannot be made to reduce $\varepsilon_n$ as a function of iteration number n when started with price and quantity vectors which are close to the solution point (i e , within 5/) That is to say, an algorithm which cannot be made to reduce the error norm [i e , Eq (8)] when started with a solution guess that is in the vicinity of a known solution point is almost by definition non-convergent From private communications with EIA personnel this conclusion has been substantiated in that most Model 22C results are not achieved in an automated fashion, and even the 5/ level of convergence is reached in most cases only after considerable hand intervention

It appears then, that a diagonal A matrix seems to be just adequate to quickly converge the largest magnitude quantities and therefore reach the level of a few percent convergence in Model 82 or Model 22C The algorithm fails at this point in trying to converge the smaller magnitude quantities This procedure eventually results in oscillation as the smaller values begin to affect the large well-converged values through the functional relationship between allocation quantities The algorithm is reasonable then for answers of a few percent but will not generally converge

allocation quantities further, even though the results are already close to a solution Although staging the maximum value of $\alpha_1$ down as a function of $\varepsilon_n$ is a reasonable approach from a theoretical viewpoint, it is also not adequate to converge the solution Only hand-choosing a constant $\alpha$ and the inherent use of a nonlinear SOR scheme seems to yield converged solutions with this algorithm To further complicate this picture, there are indications from DFI[12] that the convergence of most large models (e g , Model 22C) is dominated by relaxation of new capacity additions in basic conversion processes Here, the choice of a constant $\alpha$ relaxation in new capacity additions may be the fortuitous reason that convergence is achieved in many of these problems

Since the only workable scheme found was the constant $\alpha$ SOR approach, it seems imperative to try to analyze this algorithm more completely in the general context of Eq (10) The fact that most of the test runs with Model 82 were started with $\bar{p}$ and $\bar{q}$ guesses which were close to converged, says that the classical analysis of this algorithm using a rigorous Newton's method is essential to understanding the failings of all the diagonal schemes except constant $\alpha$ SOR In the case where the starting values are in the vicinity of the solution, Newton's method is guaranteed to converge with a quadratic rate of convergence [16] This fact indicates that the off-diagonal elements of the relaxation coefficient matrix A are crucial for success of the LEAP algorithm and only a fortuitous, non-automated procedure for finding a constant $\alpha$ seems to remedy the problem

The oscillatory nature of the algorithm, characterized by the quick convergence of large magnitude allocation quantities followed by perturbations induced by small magnitude allocation quantity non-convergence, leads

strongly to the conclusion that the coupling of allocation quantities in any single year is the dominant off diagonal functional behavior in the A matrix This then is the working assumption for the next section in which a more nearly rigorous Newton-Secant method for allocation quantity relaxation is presented In this approach the relaxation coefficients are calculated using Eq (12) and a full Jacobian matrix of the allocation transformation function $\overline{F}$ An analysis of the contribution of off-diagonal elements to convergence in the previous LEAP schemes will be made in an attempt to explain the current oscillatory breakdown in LEAP results

## VIII ALTERNATIVE LEAP RELAXATION ALGORITHM — NEWTON'S METHOD

In order to overcome the theoretical problems of analyzing the current LEAP relaxation algorithm relaxation coefficients for a full A matrix in each allocation process can be estimated using the Jacobian of the allocation process transformation function Because of its relative simplicity the Jacobian matrix can be derived analytically from the given allocation process transformation functions for use in the LEAP algorithm In what now follows, a Jacobian matrix for all quantities in an allocation process is derived and a Newton's method for relaxation using these results is sketched out

The scheme to be discussed below deals directly with quantities rather than market shares, to eliminate the complications introduced by behavioral lag coefficients in the latter approach The working assumption for these derivations is the time independence of all allocation quantities as far as the allocation transformation function is concerned Although this assumption is not theoretically true, it is felt that most of the iterative difficulties in a time dependent framework

can be removed by performing the analysis in terms of quantities as opposed to market shares

The price and quantity equations in the $k^{th}$ allocation process which are needed to begin the analysis are respectively as follows

$$\bar{p}_k = G(\bar{q}_k, \bar{p}_{k'}, \bar{q}_{k'}) \qquad k' \neq k \tag{16}$$

$$\bar{q}_k = H(\bar{p}_k, \bar{q}_{k'}, \bar{p}_{k'}) \tag{17}$$

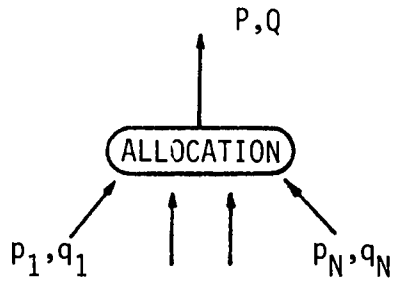
where the vectors $\bar{p}_{k'}$ and $\bar{q}_{k'}$ denote all the prices and quantities linked directly to an allocation process

In this notational form the components of the vectors $\bar{p}_k$ and $\bar{q}_k$ can be defined as follows

$$\bar{p}_k = (p_1, \quad , p_N, P) \tag{18}$$

$$\bar{q}_k = (q_I, \quad , q_N, Q) \tag{19}$$

where each vector has N + 1 components, the first N being the prices or quantities entering an allocator and the last (P or Q) being the price or quantity leaving the allocator These relationships can be represented schematically as



Assume now for simplicity that $p(q_1)$, $p_N(q_N)$ and $Q(P)$ are supply and demand transformation functions, respectively, which are given analytically by the transformation functions of the process modules linked to the $k^{th}$ allocation process These functions can also be derived analytically, if needed, from an analysis of the LEAP process module equations, but as will be seen later, this is not necessary for this Newton's method analysis The specific transformation functions for $P(\overline{q})$ and $q_1(\overline{p})$, $,q_N(\overline{p})$ defined in the allocation process are given as follows

$$P = \sum_{i=1}^{N} \frac{q_i}{Q} p_i \tag{20}$$

$$\frac{q_i}{Q} = \left(\frac{1}{p_i}\right)^{\gamma} \frac{1}{\sum_{j=1}^{N} \left(\frac{1}{p_j}\right)^{\gamma}} \tag{21}$$

$$Q = \sum_{i=1}^{N} q_i \tag{22}$$

where $\gamma$ is the price sensitivity coefficient

For notational convenience define

$$a_{i,\gamma} \equiv \left(\frac{1}{p_i}\right)^{\gamma} \tag{23}$$

$$H_{\gamma}^{-1} = \sum_{i=1}^{N} \left(\frac{1}{p_i}\right)^{\gamma} = \sum_{i=1}^{N} a_{i,\gamma} \tag{24}$$

then using the above expressions, the total transformation function for allocation quantities in the $k^{th}$ allocator can be written concisely as

$$\bar{q} = \bar{F}(\bar{q}) = \begin{pmatrix} a_{1,\gamma}[p_1(q_1)]\ H_\gamma[\bar{p}(\bar{q})]\ Q[P(\bar{q})] \\ \\ a_{N,\gamma}H_\gamma Q \end{pmatrix} \tag{25}$$

Note here that the functional dependence of $a_{1,\gamma}$, $H_\gamma$ and Q on $\bar{p}$ and $\bar{q}$ is given in parentheses only for the first element of the $\bar{F}$ operator to simplify the notation

In order to derive Newton's method for relaxation of the quantities given in Eq (25), the Jacobian matrix, $\partial\bar{F}/\partial\bar{q}$, needs to be evaluated The Jacobian for $\bar{F}$ is defined as follows

$$\frac{\partial\bar{F}}{\partial\bar{q}} \equiv \begin{pmatrix} \frac{\partial F_1}{\partial q_1} & & \frac{\partial F_1}{\partial q_N} \\ & \frac{\partial F_1}{\partial q_J} & \\ \frac{\partial F_N}{\partial q_1} & & \frac{\partial F_N}{\partial q_N} \end{pmatrix} \equiv \begin{pmatrix} f_{1,1} & & f_{1,N} \\ & f_{1,J} & \\ f_{N,1} & & f_{N,N} \end{pmatrix} \tag{26}$$

In terms of the specific functional form of $\bar{F}$ given in Eq (25), the elements of the Jacobian matrix are derivable from the following relationship,

$$\frac{\partial F_1}{\partial q_J} \equiv f_{1,J} = \frac{\partial}{\partial q_J}[a_{1,\gamma}H_\gamma Q]$$

$$= H_\gamma Q \frac{\partial a_{1,\gamma}}{\partial q_J} + a_{1\ \gamma}Q \frac{\partial H_\gamma}{\partial q_J} + a_{1,\gamma}H_\gamma \frac{\partial Q}{\partial q_J} \tag{27}$$

The three partial derivative terms in this equation can be evaluated using the definitions of $a_{i,\gamma}$ and $H_\gamma$ and also the relationships given in Eqs (20) and (21) These results on a term-by-term basis can be derived as follows For the first term on the right-hand side of Eq (27) the derivative is given by,

$$H_\gamma Q \frac{\partial a_{i,\gamma}}{\partial q_j} = -\gamma Q \frac{H_\gamma}{p_i^{\gamma+1}} \frac{dp_i}{dq_i} \delta_{ij} = -\gamma q_i \frac{p_i'}{p_i} \delta_{ij} \tag{28}$$

where $p_i' \equiv dp_i(q_i)/dq_i$ is the derivative of the $i^{th}$ supply curve feeding the allocation process and $\delta_{ij}$ is the Kroneker delta function In a similar fashion the second term in Eq (27) is

$$a_{i,\gamma} Q \frac{\partial H_\gamma}{\partial q_j} = \frac{\gamma H_\gamma^2}{p_j^{\gamma+1}} \frac{dp_j}{dq_j} = \gamma \frac{q_j}{Q} \frac{p_j'}{p_j} q_i \tag{29}$$

The third term is slightly more complicated but can be evaluated in several steps starting with the following expression,

$$a_{i\ \gamma} H_\gamma \frac{\partial Q}{\partial q_j} = a_{i\ \gamma} H_\gamma \frac{\partial Q}{\partial P} \frac{\partial P}{\partial q_j} = a_{i\ \gamma} H_\gamma Q' \frac{\partial P}{\partial q_j} \tag{30}$$

where $Q' = dQ(P)/dP$ is the demand curve slope of the final allocated quantity Since P can be written as

$$P = \frac{H_\gamma}{H_{\gamma-1}} \tag{31}$$

the derivative of $dP/dq_J$ can be evaluated using the results in Eq (29) for the derivative $\partial H_\gamma/\partial q_J$ as follows

$$\frac{\partial P}{\partial q_J} = P\left[\frac{\partial H_\gamma/\partial q_J}{H_\gamma} - \frac{\partial H_{\gamma-1}/\partial q_J}{H_{\gamma-1}}\right]$$

$$= P\left[\gamma \frac{H_\gamma}{p_J^{\gamma+1}} \; p_J' - (\gamma - 1) \frac{H_{\gamma-1}}{p_J^{\gamma}} \; p_J'\right] \tag{32}$$

The final form for the third term is therefore

$$a_{1,\gamma} H_\gamma \frac{dQ}{\partial q_J} = \gamma q_J \frac{p_J'}{p_J} \frac{q_1}{Q} \frac{Q_P'}{Q} \left[1 - \left(\frac{\gamma-1}{\gamma}\right) \frac{p_J}{P}\right] \tag{33}$$

If we now define the relative slopes (i e , the elasticities) of the supply and demand functions respectively as follows,

$$\rho_1 \equiv \frac{dp_1/p_1}{dq_1/q_1} \tag{34}$$

$$\beta = \frac{dQ/Q}{dP/P} \tag{35}$$

the final form for the Jacobian matrix element $f_{1,J}$ is,

$$f_{1,J} \equiv \frac{\partial F_1}{\partial q_J} = -\gamma\rho_1\delta_{1J} + \gamma\rho_J \frac{q_1}{Q}\left[1 + \beta\left\{1 - \left(\frac{\gamma-1}{\gamma}\right)\frac{p_J}{P}\right\}\right] \tag{36}$$

From the expression above, it is clear that the value of any element of the Jacobian matrix is a linear function of the elasticities of the supply and demand curves Since these elasticities can in general take on arbitrary values (usually, however, all the $\rho_i$ are positive and $\beta$ is negative), the element $f_{i,j}$ can also assume any value Thus, the Jacobian matrix is not in principle diagonal dominant and off-diagonal elements can play a key role in convergence It should also be noted that the supply curve with the largest elasticity and/or the largest market share will produce the largest off-diagonal element $f_{i,j}$ as given in Eq (36) This element will therefore have the most noticeable effect on convergence

At this point a simple analysis of the weaknesses of a diagonal A matrix approximation can be made In principle there is an equivalent diagonal A that will yield the same results as an exact Newton's approach For this case the following relationship must hold

$$\bar{q}_{n+1}^{(I)} = \bar{q}_n^{(I)} + \left[ I - \frac{\partial \bar{F}}{\partial \bar{q}} \right]^{-1} (\bar{q}_n^{(0)} - \bar{q}_n^{(I)})$$

$$= \bar{q}_n^{(I)} + (\bar{i} - \bar{a}) \; (\bar{q}_n^{(0)} - \bar{q}_n^{(I)}) \tag{37}$$

where $\bar{i}$ is a vector with elements of unity [i e , $\bar{i} = (1, \quad ,1)$] and $\bar{a}$ is a vector of inverse relaxation coefficients (i e $\bar{a} = \frac{1}{\alpha_1}, \quad \frac{1}{\alpha_N}$) Under these conditions the equivalent diagonal relaxation coefficients $\alpha_i$ are defined as

$$\alpha_i^{-1} = \frac{\sum_j a_{ij}^{-1} \; (q_{n,j}^{(0)} - q_{n,j}^{(I)})}{(q_{n,i}^{(0)} - q_{n,i}^{(I)} \;)} \tag{38}$$

where $a_{i,j}^{-1}$ are the coefficients of the inverse of the A matrix

Since both the numerator and the denominator of Eq (38) are in general of arbitrary magnitude and sign, it should be clear that a single $\alpha_1$ value can also be arbitrarily large and either positive or negative This is in direct contrast to the size and sign constraints placed on the relaxation coefficients in the current LEAP algorithm These constraints in general can cause oscillation and/or nonconvergence of the algorithm under very weak conditions on the structure of any LEAP network

To continue the analysis, Eq (36) can now be used to define a Newton's method approach to accelerate convergence of allocation process quantities Since a full Jacobian matrix is available analytically, Eq (10) can be used if $A^{-1}$ can be evaluated In practice it is easier to solve the following equivalent set of equations instead of inverting $A^{-1}$, thus Eq (10) can be rewritten as

$$\bar{q}_{n+1}^{(I)} = \bar{q}_n^{(I)} + \bar{\nu}_n \tag{39}$$

where $\bar{\nu}_n$ is the solution to the following linear set of equations

$$A\bar{\nu}_n = \bar{\delta}_n \tag{40}$$

Here the approximate error vector on the $n^{th}$ iteration, $\bar{\delta}_n$, is defined as

$$\bar{\delta}_n \equiv \bar{q}_n^{(0)} - \bar{q}_n^{(I)} \tag{41}$$

and as before

$$A = (I - \frac{\partial \bar{F}}{\partial \bar{q}}) \tag{42}$$

The above equation makes it possible to now impliment a full Newton's scheme in LEAP allocation processes This procedure should in theory alleviate the oscillation problem observed in results using the previous algorithm Its use in practice proceeds in the following manner Once the convergence test in allocation is made on the $n^{th}$ iteration, $\delta_n$ can be calculated This quantity together with the analytic expression for $f_{i,j}$ [i e , Eq (36)] can be used in Eq (40) to solve for $\bar{\nu}_n$, the increment to be made in $\bar{q}_n^{(I)}$ for the next iteration Solving this small set of linear equations can be done very rapidly for the new quantities in each time interval of a LEAP allocator The time spent in this procedure should easily be justified by the time saved in accelerating the solution procedure (or for that matter, just finding any converged solution)

The only remaining item needed to implement this algorithm is a procedure for evaluating $\rho_i$ and $\beta$ Since these elasticities are not generally available analytically for each generic process in LEAP (although expressions might be derived for them with some effort), it is more practical to use a simple secant approximation for them based on the results of the present and previous iterations

Thus, $\rho_i$ and $\beta$ can be evaluated approximately as follows

$$\rho_i = \frac{q_{i,n}}{p_{i,n}} \left[ \frac{p_{i,n} - p_{i,n-1}}{q_{i,n} - q_{i\ n-1}} \right] \tag{41}$$

$$\beta = \frac{P_n}{Q_n} \left[ \frac{Q_n - Q_{n-1}}{P_n - P_{n-1}} \right] \tag{42}$$

In both of these expressions the quantities and prices should be the latest available (i e , they should refer to output prices on the up pass and output quantities on the down pass of the current iteration) Both expressions are easily evaluated in LEAP since both previous and present iteration prices and quantities are available for all process modules linked to each allocation process

In summary, the analysis of the LEAP allocation process equations strongly suggests that the approximation to Newton's method outlined in this report can be used to represent the strong coupling of allocation quantities in LEAP and therefore to accelerate and/or guarantee some level of convergence of the current algorithm Although this latter algorithm is based on several assumptions (i e , time independence and process module transformation function separability), it should still offer considerable advantages over the present diagonal matrix scheme Only further practical experience will determine whether additional functional relationships must be included in this analysis to guarantee and accelerate convergence of the LEAP algorithm

## IX SUMMARY

The following eight points summarize the findings of this study

1) The LEAP code was found to be comprised of a set of coupled non-linear algebraic equations in prices and quantities The equations are modularized for each process in the energy economy and have the mathematical form of a non-linear equillibrium boundary value problem

2) An iterative up/down algorithm is used to solve the equations which closely resembles the tatonnement process of equilibrating supply and demand in a classical Walrasian economy

3) In mathematical terms, the iterative algorithm is of the non-linear Gauss-Seidel scheme for most process modules with the major exceptions being the use of a discrete Newton/SOR scheme for allocation quantities and a constant relaxation SOR scheme for new capacity additions in conversion processes

4) The EIA-LEAP algorithm was found to be non-convergent because of two major theoretical problems The first being the ad hoc scheme for calculating the relaxation coefficients in the application of the diagonal/SOR scheme and the second being the off-diagonal dominance of the Jacobian matrix in the allocation process equations where the diagonal Newton/SOR approach is used

5) The convergence test in the EIA-LEAP algorithm was also found to be weak in two respects First, the norm used generally over estimates the level of convergence achieved and second the convergence test is only used in allocation processes quantities, so that the degree of convergence of all other prices and quantities in the code is unknown and untested

6) In attempting to achieve convergence an ad hoc constant relaxation SOR scheme was found to be the only fully convergent scheme which worked on the models tested

7) Based on a significant number of numerical experiments with the algorithm it was concluded that the scheme could be improved

considerably (possibly even made convergent) if a full discretized Newton scheme were adopted for allocation process quantities Analytic expressions for the Jacobian matrix needed for the implementation of such a scheme are thus derived and recommendations for their use in LEAP are made

8) Potential problems with non-uniqueness and non-existence of the solution of the LEAP equations were also identified in the study

## REFERENCES


1 R J Adler et al , "The DFI Energy-Economy Modeling System," Decision Focus, Inc , Palo Alto, California (December 1979)

2 R Diedrich, C Cornett, and J Pearson, "LEAP Model Network and Data Documentation," Office of Energy Information Validation, U S Department of Energy (September 1979)

3 E M Oblow, "Mathematics and Taxonomy of Equilibrium Modeling," Oak Ridge National Laboratory (to be published)

4 R G Alsmiller et al "Interim Report on Model Evaluation Methodology and Evaluation of LEAP," ORNL/TM-7245 (April 1980)

5 M Goldstein et al , "Specifications of the Equation for LEAP Model 22C," ORNL-57-7 (to be published)

6 R W Peelle C R Weisbin, R G Alsmiller, Jr , "An Evaluation of the Long-Term Energy Analysis Program Used for the 1978 EIA Administrator's Report to Congress," ORNL-5741 (to be published)

7 Energy Information Administration, *Annual Report to Congress*, Vol 3, DOE/EIA-0173/13, U S Department of Energy (1979)

8 J M Ortega and W C Rheinholt, "Iterative Solutions of Nonlinear Equations in Several Variables," Academic Press, New York (1970)

9 J E Falk et al , *Methodology of the LEAP Model*, Final Report to Brookhaven National Laboratory from George Washington University, Institute for Management Science and Engineering, Washington, D C (1979)

10 P Werbos, Energy Information Agency, Washington, D C *private communication*

11 H W Brock and D M Nesbitt, Large Scale Energy Planning Models A Methodological Approach " Decision Analysis Group Stanford Rsearch Institute NSF Contract No C-915 (1977)

12 D M Nesbitt, Decision Focus, Inc , Palo Alto, CA, *private communication* (1979)

13 ibid Ref 5, Eqs II 25 - II 27

14 ibid Ref 8, p 219

15 ibid Ref 8, Sect 7 1 - 7 2

16 ibid Ref 8, Sect 10

ORNL-5754
Special

INTERNAL DISTRIBUTION

1-2 L S Abbott
3 R G Alsmiller, Jr
4 F S Alsmiller
5 S J Auerbach
6 J Barhen
7 J Barish
8 D E Bartine
9 M A Bjerke
10 D Bjornstad
11 D G Cacuci
12 S Cantor
13 G de Saussure
14 D J Downing
15-39 EPIC
40 G F Flanagan
41 T B Fowler
42 W E Ford, III
43 W Fulkerson
44 D A Gardiner
45 D Gilai
46 H Goldstein (Consultant)
47 M Goldstein
48 D Greene
49 P M Haas
50 E Halbert
51 M C G Hall
52 L Hill
53 E Hirst
54 J Horwedel
55 D T Ingersoll
56 J M Kallfelz
57-66 A S Loebl
67 J L Lucius
68 B J Jacobs
69 F C Maienschein
70 J J Manning
71 J H Marable
72 B F Maskewitz
73 B D Murphy
74 D Nguyen
75 E M Oblow
76 J V Pace
77 S C Parikh
78 C V Parks
79-88 R W Peelle
89 F G Perey
90 R B Perez
91 R Pevey
92 Y Ronen
93 M W Rosenthal
94 R W Roussin
95 R T Santoro
96 D L Selby
97 R Sharma
98 L Stewart
99 C Thomas
100 B C Toney
101 J Trimble
102 D K Trubey
103 D R Vondy
104 E Wacholder
105 J J Wagschal
106 C F Weber
107-116 C R Weisbin
117 L W Weston
118 J R White
119 G E Whitesides
120 T J Wilbanks
121 M L Williams
122 R Q Wright
123 A Zucker
124 P Greebler (Consultant)
125 H J C Kouts (Consultant)
126 W B Loewenstein (Consultant)
127 R Wilson (Consultant)
128-129 Central Research Library
130 ORNL Y-12 Technical Library - Document Reference Section
131 Laboratory Records
132 ORNL Patent Office
133 Laboratory Records - RC

EXTERNAL DISTRIBUTION

134 Office of Assistant Manager for Energy Research and Development, DOE-ORO, Oak Ridge, TN 37830

135 R L Basmann, Department of Economics, Texas A&M University, College Station, Texas 77843

136 M Becker, Dept of Nuclear Engineering, Rensselaer Polytechnic Institute, Tibbets Avenue, Troy, NY 12181

137 D J Beecy Acting Director, Division of Policy & Strategic Planning, Office of Fossil Energy, MS C-125, U S Department of Energy, Washington, DC 20545

138 W G Bentley, Florida Power and Light Company, P O Box 529100, Miami, FL 33152

139 T E Browne, Energy Supply Studies Program, Electric Power Research Institute, P O Box 10412, Palo Alto, CA 94303

140 D Cato, Office of Energy Information, U S Dept of Energy, Federal Building, 12th and Pennsylvania Avenue, NW, Washington, DC 20461

141 E Cazalet, Decision Focus, Incorporated 5 Palo Alto Square, Suite 410, Palo Alto, CA 94303

142 R Gene Clark Office of Energy Information, U S Department of Energy, Federal Building, 12th and Pennsylvania Avenue, NW Washington, DC 20461

143 S Cohen, Energy Information Administration, U S Dept of Energy, 12th and Pennsylvania Ave NW Washington, DC 20461

144 Charles Cook, Office of Energy Information, U S Department of Energy, Federal Building, 12th and Pennsylvania Avenue, NW, Washington, DC 20461

145 C Cornett, Energy Information Administration, U S Department of Energy, 12th and Pennsylvania Ave , NW, Washington, DC 20461

146 C W Craven, Science Applications, Inc , 800 Oak Ridge Turnpike, Building C, Suite 100, Oak Ridge, TN 37830

147 Robert Crockett, Energy Information Administration, U S Department of Energy, 12th and Pennsylvania Ave , NW, Washington, DC 20461

148 Ned Dearborn, Energy Information Administration, U S Department of Energy, 12th and Pennsylvania Ave , NW, Washington, DC 20461

149 L E DeSimone, San Diego Gas & Electric Company, P O Box 1831, San Diego, CA 92112

150 R C Diehl, Aerojet Energy Conversion Co , P O Box 13222, Sacramento, CA, 95813

151 Mitchell Durst, Office of Energy Information, U S Department of Energy Federal Building, 12th and Pennsylvania Avenue, NW, Washington, DC 20461

152 Energy Systems Library, Brookhaven National Laboratory, Building 475, Upton, Long Island, NY 11973

153 R T Eynon, Office of Energy Information, U S Department of Energy, Federal Building, 12th and Pennsylvania Avenue, NW, Washington, DC 20461

154 J Eyster, Office of Energy Information Validation, U S Department of Energy, Federal Building, 12th and Pennsylvania Avenue, NW, Washington, DC 20461

155 R C Fair, Department of Economics, Yale University, 206 Elm Street New Haven Connecticut 06520

156 J Finucane, Office of Energy Information Validation, U S Department of Energy, 12th and Pennsylvania Avenue, NW, Washington, DC 20461

157 A Ford, Los Alamos Scientific Laboratory, P O Box 1663, Los Alamos, NM, 87545

158 D Freedman, Professor of Statistics, University of California, Berkeley CA 94708

159 S I Gass, College of Business and Management, University of Maryland College Park, Maryland 20742

160 Donald Gantzer, U S Department of Energy, Federal Building, Room 4447, 12th and Pennsylvania Avenue, Washington, DC 20461

161 O Gildersleeve, Electric Power Research Institute, 3412 Hillview Ave , Palo Alto, CA 94303

162 C R Glassey, University of California Berkeley, 1163 Keeler Avenue, Berkeley, CA 94708

163 Neil Goldman, Energy Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

164 L Gordon, Office of Energy Information Validation, Federal Building, 12th and Pennsylvania Avenue, NW, 20461

165 H J Greenberg, U S Department of Energy, 12th and Pennsylvania Avenue, NW, Washington, DC 20461

166 M Greenberger, Stanford University Graduate School of Business, Room 383, Stanford, CA 94305

167 E Greenspan, University of Illinois, Nuclear Engineering Program, 214 Nuclear Engineering Laboratory, Urbana, IL 61801

168 James Gruhl, Energy Model Analysis Program, Model Validation Group, MIT Energy Laboratory, Cambridge, Massachusetts 02138

169 N Gruhl, 142 Plain Road, Wayland, Massachusetts 01778

170 D E Hale, Office of Energy Information Validation, U S Department of Energy, 12th and Pennsylvania, NW, Washington, DC 20461

171 John Hansen, Charles River Associates, 200 Clarendon Street Boston, MASS 02116

172 C Harris, Center for Management & Policy Research, 1625 "I" Street, NW Washington, DC 20006

173 J Hausman, Economics Department, E-52, Massachusetts Institute of Technology, Cambridge, MASS 02139

174 W W Havens, Columbia University, 520 West 120th Street, New York, NY 10027

175 W W Hogan, Harvard University 79 Boylston Street, Cambridge MASS 02138

176 Mary Hutzler, Energy Information Administration, 12th and Pennsylvania Avenue NW, Washington, DC 20461

177 R H F Jackson, Center for Applied Mathematics, U S Department of Commerce, National Bureau of Standards, Washington, DC 20234

178 C W Kilgore, Energy Information Administration, U S Department of Energy, 12th and Pennsylvania Avenue, NW Washington, DC 20461

179 D Kresge, 53 Church Street, Cambridge, MASS 02138

180 G M Lady, Energy Information Administration, U S Department of Energy, 12th and Pennsylvania Avenue, NW, Washington, DC 20461

181 Joel Lambert, National Bureau of Standards, Building 101, Washington, DC 20234

182 W Langley, FJSRL, U S Air Force Academy, Colorado Springs, CO 80840

183 C J Lee, Department of Economics, Knox College, Galesburg, IL 61401

184 A Linden, Jr , Energy Information Administration, U S Department of Energy, 2000 M Street, Washington, DC 20545

185 T V Long, II, University of Chicago, The Committee on Public Policy Studies, 1050 E 59th Street, Chicago, IL 60637

186 H McDaniel, Office of Energy Information Validation, U S Department of Energy, 12th and Pennsylvania Avenue, NW, Washington, DC 20461

187 M D McKay, Los Alamos Scientific Laboratory, P O Box 1663, Los Alamos, NM 87545

188 D L McNichol, Energy Information Administration, U S Department of Energy, 12th and Pennsylvania Avenue, NW Washington, DC 20461

189 R H Males, Director, Energy Systems, Environment and Conservation Division, Electric Power Research Institute, 3412 Hillview Avenue, P O Box 10412, Palo Alto, CA 94303

190 R A Marshalla, Decision Focus, Inc , 5 Palo Alto Square, Suite 410, Palo Alto, CA 94303

191 J S Maybee, University of Colorado, Mathematics Department, Boulder, CO 90309

192 John Mitrisin, Energy Information Administration, U S Department of Energy, 12th and Pennsylvania Avenue, NW, Washington, DC 20461

193 Thomas Mooney Energy Information Administration U S Department of Energy 12th and Pennsylvania Avenue, NW, Washington, DC 20461

194 G H Moore, Center for Energy Studies, University of Texas, Austin, TX 78712

195 T H Morlan Energy Information Administration, U S Department of Energy 12th and Pennsylvania Avenue NW, Washington, DC 20461

196 F H Murphy, Energy Information Administration, U S Deparment of Energy, 12th and Pennsylvania Avenue, NW, Washington, DC 20461

197 E L Nelson, Energy Information Administration, U S Department of Energy, 12th and Pennsylvania Avenue, NW, Washington, DC 20461

198 D M Nesbitt, Decision Focus, Inc , 5 Palo Alto Square, Suite 410, Palo Alto, CA 94304

199 David Nissen, Chase Manhattan Bank, 1 Chase Manhattan Plaza, New York, NY 10015

200 E Oatman, Electric Power Research Institute, 3412 Hillview Avenue, P O Box 10412, Palo Alto, CA 94303

201 Richard P O'Neill, Energy Information Administration, U S Department of Energy, 12th and Pennsylvania Avenue, NW, Washington, DC 20461

202 Jerry Peabody, Energy Information Administration, U S Department of Energy, 12th and Pennsylvania Avenues, NW, Washington, DC 20461

203 John Pearson Energy Information Administration, U S Department of Energy, 12th and Pennsylvania Avenue, NW, Washington, DC 20461

204 R Phillips, Decision Focus, Inc , 5 Palo Alto Square, Suite 410, Palo Alto, CA 94303

205 B Poole, Energy Information Administration, U S Department of Energy, 12th and Pennsylvania Avenue, NW, Washington, DC 20461

206 J Rambo, Lawrence Livermore Laboratory, University of California, Livermore, CA 94550

207 Lillian Regelson, Energy Information Administration, U S Department of Energy, 12th and Pennsylvania Avenue, NW, Washington, DC 20461

208 Richard Richels Electric Power Research Institute 3412 Hillview Avenue P O Box 10412 Palo Alto, CA 94303

209 Fred S Roberts, Rutgers University, Department of Mathematics, New Brunswick, NJ 08903

210 M E Rodekohr, Energy Information Administration, U S Department of Energy, 12th and Pennsylvania Avenue, NW, Washington, DC 20461

211 D J Rose, Dept of Nuclear Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139

212 L Rubin, Electric Power Research Institute, 3412 Hillview Avenue, P O Box 10412, Palo Alto, CA 94034

213 Roy Shanker, Resource Planning Associates, 1901 "L' Street, NW, Washington DC 20036

214 John B Shewmaker, Energy Information Administration, U S Department of Energy, 12th and Pennsylvania Avenue, NW, Washington, DC 20461

215 Scott Sitzer, Energy Information Administration, U S Department of Energy, 12th and Pennsylvania Avenue, NW, Washington, DC 20461

216 C Smith, Energy Information Validation, U S Department of Energy, 12th and Pennsylvania Avenues, NW, Washington, DC 20461

217 James Sweeney, Energy Modeling Forum Stanford University, Stanford, CA 94305

218 Takashi Takayama, Energy Information Administration, U S Department of Energy, 12th and Pennsylvania Avenue, NW, Washington, DC 20461

219 J E Todd, Energy Information Administration, U S Department of Energy, 12th and Pennsylvania Avenue, NW, Washington, DC 20461

220 Noel Uri, Energy Information Administration, U S Department of Energy, 12th and Pennsylvania Avenue, NW, Washington, DC 20461

221 J Weatherly, Institute of Gas Technology, 3424 S State Street, Chicago, IL 60601

222 A M Weinberg, Institute for Energy Analysis, Oak Ridge Associated Universitities, Oak Ridge, TN 37830

223 P Werbos, Office of Energy Information Validation, U S Department of Energy, 12th and Pennsylvania Avenue, NW, Washington, DC 20461

224 D O Wood, Energy Laboratory #-38, 418, Massachusetts Institute of Technology, Cambridge, MA 02139

225-228 L Zutovsky, Energy Information Administration, U S Department of Energy 12th and Pennsylvania Avenues NW Washington DC 20461

229-255 Technical Information Center (TIC)