ECON INCORPORATED

N78-16412

ECON'S OPTIMAL DECISION MODEL OF WHEAT PRODUCTION AND DISTRIBUTION--DOCUMENTATION

Report No. 77-264-1

ECON INCORPORATED

NINE HUNDRED STATE ROAD
PRINCETON, NEW JERSEY 08540
609 924-8778

ECON'S OPTIMAL DECISION MODEL OF WHEAT PRODUCTION AND DISTRIBUTION--DOCUMENTATION

Prepared for

The National Aeronautics and Space Administration
Office of Applications
Washington, D.C.

Contract No. NASW-3047

Final

15 December 1977

NOTE OF TRANSMITTAL

This report is prepared for the National Aeronautics and Space Administration, Office of Applications, under Contract NASW-3047.

The programs documented in this report were written for use in estimating benefits to the United States of improved information on agricultural production, and ECON believes them to be valid for that purpose. However, the subject is immensely complex, and the best mathematical and numerical methods produce only estimates, which may be improved in the future with the availability of better data or better analytic methods and techniques. The programs should be used only with the understanding of these limitations.

ECON acknowledges the work of Francis Sand and John Andrews in preparing this report, and the helpful cooperation of Dave Wood of Goddard Space Flight Center.

Submitted by:

John Andrews

John Andrews
Principal Investigator

Approved by:

Klaus P. Heiss

Klaus Heiss
Project Director

TABLE OF CONTENTS

LIST OF FIGURES

INTRODUCTION

In ECON Report 76-243-1, we presented the mathematical formulation and the basic results of ECON's optimal decision model of wheat production and distribution. The present report will serve to document the computer programs written to implement the model.

The programs were written in APL, an extremely compact and powerful language particularly well suited to this model, which makes extensive use of matrix manipulations.

Chapter 1 of this report presents the algorithms used and gives listings of and descriptive information on the APL programs used. Chapter 2 gives an account of possible changes in input data.

This report should be used in conjunction with ECON Report 76-243-1, which gives a complete mathematical description of the model.

1. ALGORITHMS AND NOTATION

1.1 Overview and Stages of Calculations

Essentially, the algorithms used calculate value of information as output in response to two kinds of inputs: economic parameters such as elasticities, interest rates and typical production levels; and numerical descriptions of performance of production information systems. The stages involved in these calculations are shown in Figure 1.1.

For each of two production information systems, called "current system" and "improved system," calculations are performed in three stages, dynamic programming, simulation, and function evaluation.

The first stage is dynamic programming, of which the primary output consists of the coefficients of the value functions. The dynamic programming stage requires two kinds of inputs: the economic parameters; and the statistical parameters on inventories of stored and growing crops. These statistical parameters are mathematically determined by the functional equations solved in the dynamic programming stage. Ideally, then, they would be only internal variables to this stage, rather than input variables. However, we have found no satisfactory means of solving for these variables within the dynamic programming calculations, so an iterative procedure is used, in which the decision rules determined as a by-product in the dynamic programming calculations are used to simulate the system (the second stage), and then the statistics collected from the simulation are used as input for a new pass through the dynamic programming calculations. This alternation is continued until convergence is achieved. The third stage of the calculations (function evaluation) produces

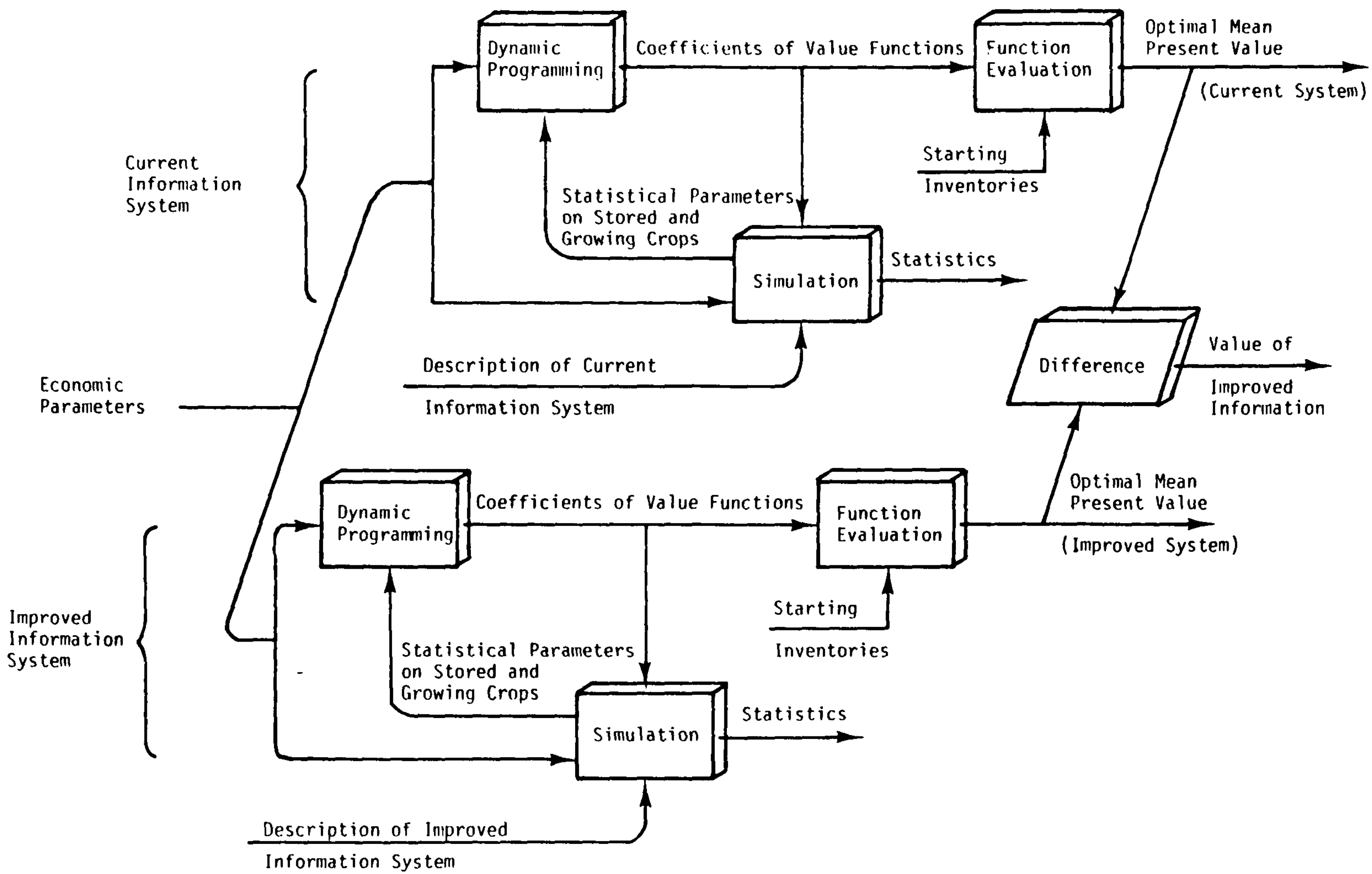

Figure 1.1 Stages of Value of Information Calculation

the optimal mean present value to the United States, and the whole world, of the produced and distributed wheat, given a specified starting level of storage and growing inventories.

After finding the optimal mean present value for both the current information system and the improved information system, we find the value of the information improvement by taking the difference of these values.

The following (18) steps are involved in performing a complete value of information calculation with the model:

1. Initialize economic parameters.
2. Initialize parameters specific to current information system.
3. Dynamic programming--find decision rules and quadratic and linear coefficients of world value functions.
4. Simulate to find statistical parameters (current information system).
5. Check for convergence of statistical parameters. If not converged, update and go to (3).
6. Find quadratic and linear coefficients of U.S. value functions.
7. Find constant terms of value functions.
8. Simulate to collect numerous statistics.
9. Evaluate world and U.S. value functions for selected starting conditions.
10. Initialize parameters specific to improved information system.
11. Dynamic programming--find decision rules and quadratic and linear coefficients of world value functions.
12. Simulate to find statistical parameters (improved information system).
13. Check for convergence of statistical parameters. If not converged, update and go to (11).

14. Find quadratic and linear coefficients of U.S. value functions.
15. Find constant terms of value functions.
16. Simulate to collect numerous statistics.
17. Evaluate world and U.S. value functions for selected starting conditions.
18. Value of information improvement--form difference of results of (9) and (17).

1.2 Details on Stages

1.2.1 Initialization

The initialization of the economic parameters consists of the following steps.

Step 1 Input

(a) m = number of periods in year.

(b) E_d = annual price elasticity of demand for United States and rest of world. This is a 2-vector.

(c) E_c - cost elasticity of production for United States and rest of world by period of year. This is a matrix of shape m × 2.

(d) P = average price in United States and rest of world. This is a 2-vector.

(3) $\mathcal{C}$ = average annual consumption in United States and rest of world. This is a 2-vector.

(f) Π = average planting in United States and rest of world by period of year. This is a matrix of shape m × 2.

(g) n = number of grid points in each dimension of the state vector for purposes of quadratic approximation of value function.

(h) r = annual discount rate.

(i) $\mathcal{E}$ = average annual exports from United States to rest of world.

(j) τ = quadratic coefficient in cost of transportation function.

(k) ω = linear coefficient in cost of transportation function.

(ℓ) D = dimension of state vector by period. This is a vector or length m.

Step 2 Compute periodic demand and production function parameters, and discount factor.

price = $\beta + 2\,\alpha \times$ period's consumption.

cost = $\delta + 2\,\gamma \times$ period's production.

(a) $\alpha = m P \,/\, (2\,\mathcal{C}\, E_d)$, a 2-vector.

(b) $\beta = P\,(1 - \frac{1}{E_d})$, a 2-vector.

(c) $\gamma_{ij} = P_i \,/\, (2\,\Pi_{ij}\, E_{c,ij})$, $i = 1, \ldots, m;\ j = 1, 2$.

γ is a matrix of shape $m \times 2$.

(d) $\delta_{ij} = P_i\,(1 - \frac{1}{E_{c,ij}})$, $i = 1, \ldots, m;\ j = 1, 2$.

(e) $\rho = (\frac{1}{1+r})^{\frac{1}{m}}$, discount factor for a single period.

Step 3 Definition of period-independent part of constant arrays.

$$A1 = \begin{pmatrix} \alpha_1 & 0 & 0 \\ 0 & -\tau & 0 \\ 0 & 0 & \alpha_2 \end{pmatrix} \qquad B1 = \begin{pmatrix} \beta_1 \\ -\omega \\ \beta_2 \end{pmatrix}$$

$$A2 = \begin{pmatrix} \alpha_1 & \alpha_1 & 0 \\ \alpha_1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \qquad B2 = \begin{pmatrix} \beta_1 \\ \beta_1 \\ 0 \end{pmatrix}$$

<u>Step 4</u> Initial estimates of value function coefficients and statistical parameters. All are chosen by rough heuristic procedure. Values are refined during calculations.

(a) $Q_{ijk} \leftarrow -.1$; $i=1, \ldots, 6$; $j=1, \ldots, 4$; $k=1, \ldots, 4$.
Quadratic coefficients of world value functions

(b) $Q^u_{ijk} \leftarrow Q_{ijk}$, Quadratic coefficients of United States value functions

(c) $L_{ij} \leftarrow 0$; $i=1, \ldots, 6$; $j=1, \ldots, 4$. Linear coefficients of world value functions

(d) $L^u_{ij} \leftarrow 0$. Linear coefficients of United States value functions.

(e)

$$U_x \leftarrow \begin{pmatrix} 55 & 350 & 0 & 0 \\ 45 & 300 & 0 & 0 \\ 35 & 40 & 250 & 250 \\ 25 & 40 & 200 & 250 \\ 15 & 40 & 150 & 250 \\ 5 & 50 & 100 & 250 \end{pmatrix},$$

mean values of grid point components in state space.

(f)

$$\Sigma_x \leftarrow \begin{pmatrix} 10 & 30 & 0 & 0 \\ 10 & 30 & 0 & 0 \\ 5 & 1 & 25 & 5 \\ 4 & 1 & 25 & 5 \\ 3 & 1 & 25 & 5 \\ 2 & 1 & 25 & 5 \end{pmatrix},$$

standard deviations of grid point components in state space

1.2.2 Current System Initialization

Step 1 Define 2 x 10 array, Σ^2, specifying the performance of the current information system.

Step 2 If better approximations of Q, Q^u, L, L^u, U_x, and Σ_x are available than those of Section 1.2.1, initialize them accordingly.

1.2.3 Dynamic Programming

Step 1 i = 6, period counter.

Step 2 Define constant arrays specifying state transformation and incremental value functions.

(a) New rows and columns are inserted, depending on the value of i, in the matrices A1, A2, to create the three dimensional array A. The insertions are the period-dependent elements. The first component of A is used to label the two value functions (whole world and United States). For each fixed value j of the first component, A_j is a symmetric matrix as follows. If i = 2 or i = 5, then

$$A_1 = \begin{pmatrix} \alpha & 0 & 0 & 0 & 0 \\ 0 & -\tau & 0 & 0 & 0 \\ 0 & 0 & -\gamma_{i1} & 0 & 0 \\ 0 & 0 & 0 & \alpha_2 & 0 \\ 0 & 0 & 0 & 0 & -\gamma_{i2} \end{pmatrix},$$

$$A_2 = \begin{pmatrix} \alpha_1 & \alpha_1 & 0 & 0 & 0 \\ \alpha_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\gamma_{i1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

If $i = 1$, $i = 3$, or $i = 4$, then

$$A_1 = A1,\ A_2 = A2.$$

If $i = 6$, then A_1 and A_2 are just as for the case of $i = 2$, except the third row and third column are omitted.

(b) New elements are inserted, depending on the value of i, before and after element 3 of B1 and B2, to create the matrix B. The first component of B (row index) is used to label the two value functions.

If $i = 2$ or $i = 5$, then

$$B = \begin{pmatrix} \beta_1 & -\omega & -\delta_{i1} & \beta_2 & \delta_{i2} \\ \beta_2 & \beta_2 & \delta_{i2} & 0 & 0 \end{pmatrix} .$$

If $i = 1$, $i = 3$, or $i = 4$, then B is the 2 x 3 matrix given by

$$B = \begin{pmatrix} B1 \\ B2 \end{pmatrix} .$$

If $i = 6$, then B is just as for the case of $i = 2$, except that the third column is omitted.

(c) The matrix C is defined, depending on the value of i, for use in building the quadratic programming tableau.

If $i = 1$, $i = 3$, or $i = 4$, then

$$C = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} .$$

If $i = 2$, or $i = 5$, then

$$C = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} .$$

If $i = 6$, then

$$C = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} .$$

(d) State transformation matrices M and N are defined, depending on the value of i, as follows.

If i = 1, then

$$M = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad N = \begin{pmatrix} -1 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} .$$

If i = 2, then

$$M = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad N = \begin{pmatrix} -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} .$$

If i = 3 or i = 4, then

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad N = \begin{pmatrix} -1 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{pmatrix} .$$

If i = 5, then

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad N = \begin{pmatrix} -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} .$$

If i = 6, then

$$M = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}, \quad N = \begin{pmatrix} -1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 1 \end{pmatrix} .$$

Step 3 Build a matrix X whose columns are the grid points for value function approximation. X has D_i rows and N^{D_i} columns. In each coordinate j of the state space (of dimension D_i), a set $\mathcal{S}_j$ of n equally spaced points in formed with mean $U_{x,j}$ and standard deviation $\Sigma_{x,j}$. Each column of X is formed from one of the N^{D_i} combinations of elements, the first element from $\mathcal{S}_1$, the second from $\mathcal{S}_2$, ..., the D_i th from $\mathcal{S}_{D_i}$.

Step 4 $j = 1$, counter for state points.

Step 5 Quadratic Programming

(a) S = column j of X, the state point.

(b) $E = A_1 + \rho N' Q_{i+1} N$.

(c) $F = B_1 + 2\rho Q_{i+1} MSN + \rho L_{i+1} N$.

(d) $G = \rho(S'M' Q_{i+1} MS + L_{i+1} MS)$.

(e) $$D = \begin{Bmatrix} S \text{ if } D_i = 2 \\ (s_1, s_2) \text{ if } D_i = 4 \end{Bmatrix} .$$

(Note on notation. When $i = 6$, we take $i + 1$ reduced mod 6. That is, for this case, $i + 1 = 1$. When $D_{i+1} = 2$, the matrix Q_{i+1} is 4 x 4, but only the upper left block of size 2 x 2 is significant--the other elements are never used nor changed during the algorithm. For this case, we write "Q_{i+1}" to mean only this 2 x 2 matrix. Similarly, "L_{i+1}" means only the first D_{i+1} elements of the 4-vector.)

(f) Find 1-stage decision vector Y to maximize $Y'E\,Y + Y'F$ subject to the constraints

$Y \geq 0$, $CY \leq D$.

Step 6 Evaluate the cumulative value function for period i at point S.

$$V_i = Y'E_{i+1} Y + Y'F_{i+1} + G_{i+1}$$

Step 7 End of loop on state points.

(a) $j \leftarrow j + 1$

(b) Go to Step 5 if $j \leq n^{D_i}$, Step 8 otherwise.

Step 8 Find period-i coefficients of cumulative value function by least squares approximation.

(a) Form the matrix MAT which, when postmultiplied by the values of a function at the points which are the columns of X, produces the coefficients of the least squares fit to the function at those points.

(b) COEFF = (MAT)V

(c) Select Q and L from COEFF

Step 9 End of loop on periods.

(a) $i \leftarrow i - 1$

(b) Go to Step 2 if $i \geq 1$, Step 10 otherwise.

Step 10 End of loop for convergence.

Go to Step 1 if Q and L are unchanged from last iteration; otherwise stop.

1.2.4 Simulation

The simulation algorithm is an adaptation of the dynamic programming algorithm. It is run after the value function coefficients Q and L have been calculated, and uses many of the same variables.

Step 1 Initialization

(a) Input Σ^2 array, a 2×10 matrix giving the means of the squares of the stochastic term Φ_i of the state transformation.

(b) Input n, the number of years of simulation to be performed.

(c) Create standard deviation array SD for use in sampling values of Φ_i by reshaping the Σ^2 array and taking square root. In its original shape, the first row refers to the Untied States, and the second to the rest of the world. Each row contains terms referring to the growing crop before the beginning of the crop year, followed by terms referring to the same crop during the crop year, considered to begin June 1. The reshaped array has shape 4 × m; the growing crop and current crop at a given time of year are represented by different rows in the same column. Formally, the process is as follows.

$$SD_{ij} = \left(\Sigma^2{}_{T_{ij}, U_{ij}} \right)^{\frac{1}{2}}$$

where

$$T = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 1 & 1 & 2 \\ 0 & 2 & 2 & 2 & 2 & 0 \\ 0 & 2 & 2 & 2 & 2 & 0 \end{pmatrix} ,$$

$$U = \begin{pmatrix} 6 & 7 & 8 & 9 & 10 & 5 \\ 6 & 1 & 2 & 3 & 4 & 5 \\ 0 & 7 & 8 & 9 & 10 & 0 \\ 0 & 1 & 2 & 3 & 4 & 0 \end{pmatrix} .$$

and $\Sigma^2_{0,0}$ is taken to be 0.

(d) Select vector Φ with 4nm elements by random sampling. For each of the mn periods of the simulation, four successive elements of Φ are obtained from discrete uniform distributions with mean 0 and standard deviations SD, i, where i is the index (1 to 6) of the current period.

(e) Find standard deviations SD' over 2n years of the blocks of 4m adjacent elements of the vector $(\Phi, -\Phi)$.

(f) $SD \leftarrow SD^2 / SD'$

(g) Repeat step (d). At this point, the sample points comprising $(\Phi, -\Phi)$ have mean 0 and the standard deviations of the original SD array.

(h) S = first two components of first row of U_x, the mean inventory level vector at time 1.

(i)

$$U_x = \Sigma_x = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \text{ initialize}$$

mean and standard deviation of state vector.

<u>Step 2</u> $\ell \leftarrow 0$, counter for Φ vector.

<u>Step 3</u> $i \leftarrow 6$, period counter.

<u>Step 4</u> Increment grid parameter accumulators.

(a) $U_{x,i} \leftarrow U_{x,i} + \frac{S}{2n}$ (If $d_i = 2$, add zero components to S).

(b) $\Sigma_{x,i} \leftarrow \Sigma_{x,i} + \frac{S^2}{2n}$ (If $d_i = 2$, add two zero components to S).

Step 5 Define constant arrays specifying state transformation and incremental value functions. This step is identical to Step 2 of Section 1.2.3.

Step 6 Quadratic Programming. This step is identical to Step 5 of Section 1.2.3, except that 5a is omitted, since S has already been selected.

Step 7 Apply state transformation to get next period's state vector.

(a) $S \leftarrow MS + NY + (\phi_{\ell+1}, \phi_{\ell+2}, \phi_{\ell+3}, \phi_{\ell+4})$

(b) $\ell \leftarrow \ell + 4$

Step 8 End of loop on periods.

(a) $i \leftarrow i + 1$

(b) Go to Step 4 if $i \leq 6$, Step 9 otherwise.

Step 9 End of loop on years.

(a) $k \leftarrow k + 1$

(b) Go to Step 3 if $k \leq n$, Step 10 otherwise.

Step 10 End of loop on antithetic variates.

(a) $\phi \leftarrow - \phi$

(b) Go to Step 2 if $\ell < 8$ mm, Step 11 otherwise.

Step 11 Complete statistical calculations.

$$\Sigma_x \leftarrow \left[\frac{n(\Sigma_x - (U_x)^2)}{n - 0.5}\right]^{\frac{1}{2}}$$

Step 12 Stop.

1.2.5 United States Value Functions

The algorithm of this section is an extension of the dynamic programming algorithm (Section 1.2.3) to calculate the United States value function, as well as the world value function which is maximized.

Steps 1 through 5 Identical to Steps 1 through 5 of Section 1.2.3.

Step 6 Evaluate the cumulative value functions (world and United States) for period i at point S.

$$V_i = Y'E_{i+1}Y + Y'F_{i+1} + G_{i+1} .$$

$$E^u = A_2 + \rho N'Q^u_{i+1}N .$$

$$F^u = B_2 + 2\rho Q^u_{i+1}MSN + \rho L^u_{i+1}N .$$

$$G^u = \rho(S'M'Q^u_{i+1}MS + L^u_{i+1}MS) .$$

$$V^u_i = Y'E^u_{i+1}Y + Y'F^u_{i+1} + G^u_{i+1} .$$

Step 7 End of loops on state points.

(a) $j \leftarrow j + 1$

(b) Go to Step 5 if $j \leq n^{D_i}$, Step 8 otherwise.

Step 8 Find period-i coefficients of cumulative value function (world and United States) by least squares approximation.

(a) New rows and columns are inserted, depending on the value of i, in the matrices A1, A2, to create the three dimensional array A. The insertions are the period-dependent elements. The first component of A is used to label the two value functions (whole world and United States). For each fixed value j of the first component, A_j is a symetric matrix as follows. If i = 2 or i = 5, then

$$A_1 = \begin{pmatrix} \alpha & 0 & 0 & 0 & 0 \\ 0 & -\tau & 0 & 0 & 0 \\ 0 & 0 & -\gamma_{i1} & 0 & 0 \\ 0 & 0 & 0 & \alpha_2 & 0 \\ 0 & 0 & 0 & 0 & -\gamma_{i2} \end{pmatrix},$$

$$A_2 = \begin{pmatrix} \alpha_1 & \alpha_1 & 0 & 0 & 0 \\ \alpha_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\gamma_{i1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

If $i = 1$, $i = 3$, or $i = 4$, then

$A_1 = A1$, $A_2 = A2$.

If $i = 6$, then A_1 and A_2 are just as for the case of $i = 2$, except the third row and third column are omitted.

(b) COEFF $\leftarrow$ (MAT)V

(c) COEFFU $\leftarrow$ (MAT)V^u

(d) Select Q, L, and KEF from COEFF; Q^u, L^u, and KEFU from COEFFU.

<u>Step 9</u> End of loop on periods.

(a) $i \leftarrow i - 1$

(b) Go to Step 2 if $i \geq 1$, Step 10 otherwise.

<u>Step 10</u> End of loop for convergence. Go to Step 1 if any of Q, L, Q^u, L^u are different from last iteration; otherwise stop.

1.2.6 Constant Terms

The constant terms of the cumulative value functions (world and U.S.) are most efficiently calculated in this separate algorithm rather than in the dynamic programming algorithm since convergence is slow and the results are not needed for the determination of the decision rules, but only for evaluation of the cumulative value functions.

Step 1 Initialization. Identical to Step 1 of Section 1.2.4., except (b) is omitted.

Step 2 i = 6, period counter.

Step 3 Update estimate of period-i constant terms of cumulative value functions (world and U.S.).

$$\text{(a)} \quad K_i \leftarrow KEF_i + \rho\left[K_{i+1} + \sum_{j=1}^{D_i} (SD_{ij})^2\, Q_{i+1,j,j}\right],$$

$$\text{(b)} \quad K_i^u \leftarrow KEFU_i + \rho\left[K_{i+1}^u + \sum_{j=1}^{D_i} (SD_{ij})^2\, Q_{i+1,j,j}^u\right].$$

(Note on notation. When $i = 6$, we take $i + 1$ reduced mod 6, as explained in the note after Step 6 of Section 1.2.3.).

Step 4 End of loop on periods

(a) $i \leftarrow i - 1$

(b) Go to Step 3 of $i \geq 1$, Step 5 otherwise.

Step 5 End of loops for convergence. If K or K^u has changed since last iteration, go to Step 2; otherwise Step 6.

Step 6 Stop.

1.2.7 Statistics

The statistical summaries of the operation of the wheat markets are obtained by simulation; this alogrithm is thus essentially the same as that of Section 1.2.4, but additional quantities are tracked.

Step 1 Initialization

(a) Input Σ^2 array, a 2×10 matrix giving the means of the squares of the stochastic terms Φ_i of the state transformation.

(b) Input n, the number of years of simulation to be performed.

(c) Create standard deviation array SD for use in sampling values of Φ_i by reshaping the Σ^2 array and taking square root. In its original shape, the first row refers to the United States, and the second to the rest of the world. Each row contains terms referring to the growing crop before the beginning of the crop year, followed by terms referring to the same crop during the crop year, considered to begin June 1. The reshaped array has shape $4 \times m$; the growing crop and current crop at a given time of year are represented by different rows in the same column. Formally, the process is as follows.

$$SD_{ij} = \left(\Sigma^2{}_{T_{ij}, U_{ij}} \right)^{\frac{1}{2}}$$

where

$$T = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 1 & 1 & 2 \\ 0 & 2 & 2 & 2 & 2 & 0 \\ 0 & 2 & 2 & 2 & 2 & 0 \end{pmatrix},$$

$$U = \begin{pmatrix} 6 & 7 & 8 & 9 & 10 & 5 \\ 6 & 1 & 2 & 3 & 4 & 5 \\ 0 & 7 & 8 & 9 & 10 & 0 \\ 0 & 1 & 2 & 3 & 4 & 0 \end{pmatrix}.$$

and $\Sigma^2_{0,0}$ is taken to be 0.

$$MULT \leftarrow \left(\begin{array}{ccccccccccccc} \multicolumn{4}{c}{\overbrace{\hphantom{\tfrac{1}{6}\ \tfrac{1}{6}\ \cdots\ \tfrac{1}{6}}}^{\text{4 Columns}}} & \multicolumn{4}{c}{\overbrace{\hphantom{1\ 1\ \cdots\ 1}}^{\text{9 Columns}}} & & & \multicolumn{4}{c}{\overbrace{\hphantom{\rho\ \rho\ \cdots\ \rho}}^{\text{12 Columns}}} \\ \frac{1}{6} & \frac{1}{6} & & \frac{1}{6} & 1 & 1 & & 1 & \frac{1}{6} & \frac{1}{6} & \rho & \rho & \rho \\ \frac{1}{6} & \frac{1}{6} & & \frac{1}{6} & 1 & 1 & & 1 & \frac{1}{6} & \frac{1}{6} & \rho^2 & \rho^2 & \rho^2 \\ \frac{1}{6} & \frac{1}{6} & \cdots & \frac{1}{6} & 1 & 1 & \cdots & 1 & \frac{1}{6} & \frac{1}{6} & \rho^3 & \rho^3 \cdots & \rho^3 \\ \frac{1}{6} & \frac{1}{6} & & \frac{1}{6} & 1 & 1 & & 1 & \frac{1}{6} & \frac{1}{6} & \rho^4 & \rho^4 & \rho^4 \\ \frac{1}{6} & \frac{1}{6} & & \frac{1}{6} & 1 & 1 & & 1 & \frac{1}{6} & \frac{1}{6} & \rho^5 & \rho^5 & \rho^5 \\ \frac{1}{6} & \frac{1}{6} & & \frac{1}{6} & 1 & 1 & & 1 & \frac{1}{6} & \frac{1}{6} & \rho^6 & \rho^6 & \rho^6 \end{array}\right),$$

multiplier to weight tracked quantities for annual mean, total, or present value calculations.

(e) Select vector Φ with 4nm elements by random sampling. For each of the mn periods of the simulation, four successive elements of Φ are obtained from discrete uniform distributions with mean 0 and standard deviations $SD_{,i}$, where i is the index (1 to 6) of the current period.

(f) Find standard deviations SD′ over 2n years of the blocks of 4m adjacent elements of the vector $(\Phi, -\Phi)$.

(g) $SD \leftarrow SD^2/SD'$

(h) Repeat Step (d). At this point, the sample points comprising $(\Phi, -\Phi)$ have mean 0 and the standard deviations of the original SD array.

(i) S = first two components of first row of U_x, the mean inventory level vector at time 1.

(j)

$$\mu = VAR = \begin{pmatrix} 0 & 0 & & & & & 0 \\ 0 & 0 & & & & & 0 \\ 0 & 0 & & & & & 0 \\ 0 & 0 & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & & & & & 0 \\ 0 & 0 & & & & & 0 \end{pmatrix} \text{(27 columns)},$$

initialize mean and variance by period of 27 quantities to be tracked for statistics. These are:

(1) U.S. Stock Estimate

(2) U.S. Growing Crop Estimate

(3) R.O.W. Stock Estimate

(4) R.O.W. Growing Crop Estimate

(5) U.S. Consumption

(6) U.S. Exports

(7) U.S. Planting

(8) R.O.W. Consumption

(9) R.O.W. Planting

(10) Random Change in U.S. Stock Estimate

(11) Random Change in U.S. Growing Crop Estimate

(12) Random Change in R.O.W. Stock Estimate

(13) Random Change in R.O.W. Growing Crop Estimate

(14) U.S. Price

(15) R.O.W. Price

(16) U.S. Export Revenue

(17) U.S. Production Cost

(18) R.O.W. Production Cost

(19) R.O.W. Transportation Cost

(20) U.S. Gross Welfare

(21) R.O.W. Gross Welfare

(22) World Net Welfare (all agents)

(23) U.S. Net Welfare (all agents)

(24) U.S. Consumers' Net Welfare

(25) U.S. Producers' Net Welfare

(26) R.O.W. Consumers' Net Welfare

(27) R.O.W. Producers' Net Welfare

(k) $\mu_A \leftarrow AVAR \leftarrow (0, 0, \ldots, 0)$, initialize annual means and variances of above 27 quantities.

(l) Input S, the state vector at the start of the simulation, time 1.

<u>Step 2</u> $\ell \leftarrow 0$, Counter for Φ vector.

<u>Step 3</u> $i \leftarrow 6$, Period counter.

<u>Step 4</u> Define constant arrays specifying state transformation and

incremental value functions. This step is identical to Step 2 of Section 1.2.3.

Step 5 Quadratic Programming. This step is identical to Step 5 of Section 1.2.3, except that 5a is omitted, since S has already been selected.

Step 6 Build vector $STAT_i$ of 27 quantities at time i for statistics.

(a) $STAT_{i1}, \ldots, STAT_{i4}$ are taken from S.

(b) $STAT_{i5}, \ldots, STAT_{i9}$ are taken from Y.

(c) $(STAT_{i10}, \ldots, STAT_{i13}) \leftarrow (\phi_{\ell+1}, \ldots, \phi_{\ell+4})$.

(d) $(STAT_{i14}, STAT_{i15}) \leftarrow (\beta_1 + 2\alpha_1\ STAT_{i5}, \beta_2 + 2\alpha_2\ STAT_{i8})$, prices.

(e) $STAT_{i16} \leftarrow STAT_{i14} \times STAT_{i6}$ (export revenue).

(f) $(STAT_{i17}, STAT_{i18}) \leftarrow (STAT_{i7}, STAT_{i9}) \times (\delta_i + (STAT, STAT_{i9})\gamma_i)$, production cost

(g) $STAT_{i19} \leftarrow STAT_{i6} \times (\omega + STAT_{i6} \times \tau)$, transportation cost.

(h) $(STAT_{i20}, STAT_{i21}) \leftarrow (STAT_{i5} \times (\beta_1 + \alpha_1 \times STAT_{i5}), STAT_{i8} \times (\beta_2 + \alpha_2\ STAT_{i8}))$, gross welfare

(i) $(STAT_{i22}, \ldots, STAT_{i27}) \leftarrow Y'AY + Y'B$, net welfare.

Step 7 Increment Statistical Accumulators.

$$\mu_i \leftarrow \mu_i + \frac{STAT_i}{2_n} ,$$

$$VAR_i \leftarrow VAR_i + \frac{(STAT_i)^2}{2_n} .$$

Step 8 Apply state transformation to get next period's state vector.

(a) $S \leftarrow MS + NY + (\phi_{\ell+1}, \ldots, \phi_{\ell+4})$.

(b) $\ell \leftarrow \ell + 4$.

<u>Step 9</u> End of loop on periods.

(a) $i \leftarrow i + 1$

(b) Go to Step 4 if $i \leq 6$, Step 10 otherwise.

<u>Step 10</u>

(a) $$\mu_A \leftarrow \mu_A + \frac{\sum_{i=1}^{6} \mathrm{STAT}_i \times \mathrm{MULT}_i}{{}^{2}n} ,$$

(b) $$\mathrm{AVAR} \leftarrow \mathrm{AVAR} + \frac{(\sum_{i=1}^{6} \mathrm{STAT}_i \times \mathrm{MULT}_i)^2}{{}^{2}n} ,$$

calculate annual means and mean squares of tracked quantities.

<u>Step 11</u> End of loop on years.

(a) $k \leftarrow k + 1$

(b) Go to Step 3 if $k \leq n$, Step 12 otherwise.

<u>Step 12</u> End of loop on antithetic variates.

(a) $\Phi \leftarrow - \Phi$.

(b) Go to Step 2 if $\ell < 8mn$, Step 13 otherwise.

<u>Step 13</u>

(a) $$\mathrm{VAR} \leftarrow \frac{(\mathrm{VAR} - \mu^2)n}{n - \frac{1}{2}} ,$$

(b) $$\mathrm{AVAR} \leftarrow \frac{(\mathrm{AVAR} = \mu_A^2)n}{n - \frac{1}{2}} ,$$

complete calculation of periodic and annual variances of tracked quantities.

<u>Step 14</u> Print μ, μ_A, VAR, and AVAR arrays.

<u>Step 15</u> Stop.

1.2.8 Evaluations

This algorithm evaluates the quadratic functions giving optimal mean net present value to the whole world and to the United States, for chosen arguments (starting inventory estimates). As an option, the algorithm calculates the mean value of these value functions over the various possible starting inventory arguments.

Step 1 Input argument $S = (s_1, s_2)$

Step 2 If $S = 0$, go to Step 5.

Step 3

(a) Print $r[K_1 + S'(L_1 + Q_1 S)]$, whole world value.

(b) Print $r[K_1^u + S'(L_1^u + Q_1^u S)]$, United States value.

Step 4 Go to Step 6.

Step 5

(a) Print $r[Q_{111}\Sigma_{x,11}^2 + Q_{122}\Sigma_{x,12}^2 + K_1 + S'(L_1 + Q_1 S)]$, whole world mean value.

(b) Print $r[Q_{111}^u\Sigma_{x,11}^2 + Q_{122}^u\Sigma_{x,12}^2 + K_1^u + S'(L_1^u + Q_1^u S)]$, United States mean value.

Step 6 Stop.

1.2.9 Value of Information

The annual value of the improvement in information in the steady state is obtained as follows.

Step 1 Do the evaluation of Step 1.2.8 (mean value case) with K, L, Q, K^u, L^u, and Q^u corresponding to the current information system and S obtained from a simulation corresponding to the current information system (S is defined in Section 1.2.4, Step 1h). Denote the results (W_c, V_c).

Step 2 Do the evaluations of Step 1.2.8 with K, L, Q, K^u, L^u, and Q^u corresponding to the improved information system and S obtained from a simulation corresponding to the improved information system. Denote the results (W_I, V_I).

Step 3 The value of the improvement to the world is $W_I - W_C$, and to the United States in $V_I - V_C$,

Step 4 Stop.

1.3 The APL Programs

The dynamic programming algorithm, which is the heart of the model, is programmed in the APL function VFS (standing for value functions). The initializations required are handled by the function INITIAL. The simulation to find the statistical parameters on the state variables is done by SIMGRID, and the alternation between VFS and SIMGRID required to obtain convergence of the statistical parameters is performed by ITERATE. Thus, the user begins a calculation by running INITIAL and ITERATE, with VFS and SIMGRID called automatically from ITERATE. The quadratic and linear coefficients of the United States value functions are obtained after convergence by USVFS, and the constant terms of the value functions are calculated by CONTERMS. The statistics are obtained by simulation in SIMSTATS, and finally, the evaluation of the quadratic value functions is done by EVAL.

Besides the functions mentioned above, there are various subordinate functions which they call. The following sections contain a glossary of the main global variables, and a listing and discussion of each APL function.

1.3.1 Global Variables

With each APL variable, we give the shape, algebraic name used in this report, a brief description, and the step in the algorithm where the variable is defined.

APL Name and Shape	Algebraic Name and Description	
A, 2 4 4	A	Quadratic coefficients of incremental value functions, Section 1.2.3, Step 2a.
ADR, 1	r	Annual discount rate, Section 1.2.1, Step 1h.
A, 2 4 4	A	Quadratic coefficients of incremental value functions, Section 1.2.3, Step 2a.
AFLAG, 1	--	Flag used in function NOTESHIFT.
AIN, 2 3 3	A1, A2	Period-independent part of A, Section 1.2.1, Step 3.
ALPHA, 2	α	$-\frac{1}{2}$ Slope of demand function, Section 1.2.1, Step 2a.
APRD, 2	--	Average annual production.
B, 2 4	B	Linear coefficients of incremental value functions, Section 1.2.3, Step 2b.
BETA, 2	β	Intercept of demand function, Section 1.2.1, Step 2b.
BIN, 2 3	B1, B2	Period-independent part of B, Section 1.2.1, Step 3.
C, 2 4	C	Matrix of left hand side coefficients in quadratic programming problem, Section 1.2.3, Step 2c.
CNSMPTN, 2	$\mathcal{C}$	Average annual consumption, Section 1.2.1, Step 1e.
CUR, 2 10	Σ^2	Description of current information system, Section 1.2.2, Step 1.

APL Name and Shape	Algebraic Name and Description	
DELTA, 6 2	δ	Intercept of production cost function, Section 1.2.1, Step 2d
DIMX, 6	D	Dimension of state vector by period, Section 1.2.1, Step 1ℓ.
E, 4 4	E	Quadratic coefficients in expansion of value functions in powers of Y. Section 1.2.3, Step 6b.
EXPRTS, 1	$\mathcal{E}$	Average annual exports from United States to rest of world. Section 1.2.1, Step 1e.
F, 4	F	Linear coefficients in expansion of value functions in powers of Y. Section 1.2.3, Step 6c.
Gamma, 6 2	γ	½ Slope of cost function for production. Section 1.2.1, Step 2c.
GE, 2 10	Σ^2	Description of improved information system.
IMAX, 1	--	Number of state points at which value functions are evaluated, n^{D_i}. Section 1.2.2, Step 8b.
KEF, 6	KEF	Part of the constant terms of the world cumulative value functions, Section 1.2.5, Step 9c.
KEFU, 6	KEFU	Part of the constant terms of the U.S. cumulative value functions, Section 1.2.5, Step 9c.
KON, 6	K	Constant terms of the world cumulative value functions, Section 1.2.6, Step 3a.
KONU, 6	K^u	Constant terms of the U.S. cumulative value functions, Section 1.2.6, Step 3b.
L, 6 4	L	Linear coefficients of world cumulative value functions. Section 1.2.3, Step 9b, and Section 1.2.1, Step 4c.
LU, 6 4	L^u	Linear coefficients of U.S. cumulative value functions. Section 1.2.5, Step 9c, and Section 1.2.1, Step 4d.

APL Name and Shape	Algebraic Name	Description
M, 2 2 4 2 4 4 2 4	M	State transformation matrix. Section 1.2.3, Step 2d.
MEANX, 6 4	U_x	Mean values of grid point components in state space. Section 1.2.1, Step 4e.
N, 2 4 2 3 4 5 4 3 4 6	N	State transformation matrix. Section 1.2.3, Step 2d.
NGRID, 1	n	Number of gird points in each dimension of the state vector. Section 1.2.1, Step 1g.
NPERIODS, 1	m	Number of periods in year.
OMEGA, 1	ω	Linear coefficient in cost of transportation function. Section 1.2.1, Step 1k.
PRDCTN, 6 2	Π	Average planting in United States and rest of world by period of year. Section 1.2.1, Step 1f.
Q, 6 4 4	Q	Quadratic coefficients of world cumulative value functions. Section 1.2.3, Step 9b., and Section 1.2.1, Step 4a.
QU, 6 4 4	Q^u	Quadratic coefficients of U.S. cumulative value functions. Section 1.2.5, Step 9c, and Section 1.2.1, Step 4b.
QPREV, 6 4 4	--	Stored value of Q for convergence check. Section 1.2.3, Step 11.
QPREVU, 6 4 4	--	Stored value of Q^u for convergence check. Section 1.2.5, Step 11.
RHO, 1	ρ	Discount factor for one period. Section 1.2.1. Step 2e.
STDX, 6 4	Σ_x	Standard deviations of grid point components in state space. Section 1.2.1, Step 4f.

APL Name and Shape	Algebraic Name	Description
TAU, 1	τ	Quadratic coefficient in cost of transportation function. Section 1.2.1, Step 1j.
TOL, 1	--	Tolerance factor for convergence of value function coefficients. Defined from keyboard.
X, 2 9 4 81	X	Matrix of grid point coordinates. Section 1.2.3, Step 3.
Y, 3 4 5	Y	1 stage decision vector. Section 1.2.3, Step 6f.

1.3.2 Function INITIAL

Function INITIAL is listed in Figure 1.2. If a new calculation is desired in which demand or cost parameters are changed, or the discount rate, or the degree of resolution (NGRID), the appropriate changes should be made in lines [2] through [13] of INITIAL. Then INITIAL should be run, followed by VFS or ITERATE.

INITIAL is quite straightforward, but some explanation of line [6] is in order. Planting of wheat is assumed to take place in the United States in the second and fifth periods, and in the rest of the world in the second, fifth, and sixth periods. Thus, the production cost function parameters, γ and δ, are used only in these periods (by function STRUCTURE). The elements of the array PRDCTN corresponding to the non-planting times, therefore, have no effect on the calculation, but if they are zero, numerical problems are encountered in line [16]. To avoid this, we have put 0.1 in these positions.

1.3.3 Function ITERATE

This function, listed in Figure 1.3 performs the iterations of dynamic programming and simulation leading to convergence of the statistical

```
      ∇INITIAL[⎕]∇
    ∇ INITIAL;EDMND;ECST;PRICE;HI;LO;T
[1]   AFLAG←1
[2]   EDMND← ¯0.48 ¯0.16
[3]   NGRID←3
[4]   ECST← 6 2 ⍴ 0.5 0.5
[5]   PRICE← 132 140
[6]   PRDCTN← 6 2 ⍴ 0.1 0.1 42.5 249 0.1 0.1 0.1 0.1 7.5 27 0.1 24
[7]   EXPRTS←30
[8]   CNSMPTN←(APRD←+/[1] PRDCTN)-EXPRTS,-EXPRTS
[9]   NPERIODS←6
[10]  ADR←0.1
[11]  RHO←(÷1+ADR)*÷NPERIODS
[12]  TAU←0.05
[13]  OMEGA←8
[14]  ALPHA←NPERIODS×PRICE÷2×CNSMPTN×EDMND
[15]  BETA←PRICE×1-÷EDMND
[16]  GAMMA←(6 2 ⍴PRICE)÷2×PRDCTN×ECST
[17]  DELTA←(6 2 ⍴PRICE)×1-÷ECST
[18]  DIMX← 2 2 4 4 4 4
[19]  T← 0 0 0
[20]  AIN← 6 3 3 ⍴0
[21]  BIN← 6 3 ⍴0
[22]  AIN[1;;]← 3 3 ⍴ALPHA[1],T,(-TAU),T,ALPHA[2]
[23]  BIN[1;]←BETA[1],(-OMEGA),BETA[2]
[24]  AIN[2;;]← 3 3 ⍴ALPHA[1],ALPHA[1],0,ALPHA[1], 0 0 ,T
[25]  BIN[2;]←BETA[1],BETA[1],0
[26]  AIN[3;;]← 3 3 ⍴(-ALPHA[1]),T,T, 0 0
[27]  BIN[3;]←T
[28]  AIN[4;;]← 3 3 ⍴0
[29]  BIN[4;]←T
[30]  AIN[5;;]← 3 3 ⍴T,T, 0 0 ,-ALPHA[2]
[31]  BIN[5;]←T
[32]  AIN[6;;]← 3 3 ⍴0
[33]  BIN[6;]←T
    ∇
```

Figure 1.2 Listing of APL Function INITIAL

```
      ∇ITERATE[⎕]∇
      ∇ ARG ITERATE NTIME
[1]   OLDMEAN←MEANX
[2]   OLDSTD←STDX
[3]   ARG SIMGRID NTIME
[4]   'GRID SHIFT'
[5]   ⌈/,|(MEANX,STDX)-(OLDMEAN,OLDSTD)
[6]   NEWGRID STEP
[7]   VFS NUM
[8]   →1
      ∇
```

```
      ∇NEWGRID[⎕]∇
      ∇ NEWGRID T
[1]   MEANX←(OLDMEAN×1-T)+MEANX×T
[2]   STDX←(OLDSTD×1-T)+STDX×T
      ∇
```

Figure 1.3 Listing of APL Functions ITERATE and NEWGRID parameters. Its left argument is the array Σ^2 (Section 1.2.4, Step 1a) describing the performance of the information system. Its right argument is 1/2 the number of years to run the simulations. After running SIMGRID to determine the simulated statistical parameters, ITERATE calls NEWGRID to form the new parameters as a linear combination of the old ones and the simulated ones. The relaxation coefficient, STEP, is a global variable defined from the keyboard. Values of STEP between .05 and .5 have been found suitable. The variable NUM is also defined from the keyboard and is used as the argument of VFS to limit the number of dynamic programming cycles before SIMGRID is rerun. NUM = 1 has been found most often suitable for speedy convergence.

1.3.4 Function VFS

The listing of this function is in Figure 1.4. The following is a glossary of important variables used in VALUEFUNCTIONS and not discussed in Section 1.3.1.

APL Name and Shape	Algebraic Name and Description	
CLOOP, 1	--	Label for start of loop for convergence of value function coefficients.
COEFF (Vector, length varies)	COEFF	Receives results of least squares approximation of value function coefficients. Defined in Function LS.
D, 2 4	D	Right hand side of constraint inequality on decision vector Y in quadratic programming problem.
DIM, 1	D_{i+1}	Dimension of state variable in next period.
G, 1	G	Constant term in expansion of value functions in powers of Y.
I, 1	j	Counter for state points.

```
      ∇VFS[⎕]∇
      ∇ VFS ITERS
[1]    FILLX← 6 4 ⍴ 1 0 1 0 1 0 1 0 ,16⍴1
[2]    K←1
[3]   CLOOP:
[4]    IPER←6
[5]   PERLOOP:
[6]    STRUCTURE IPER
[7]    IPER BUILDX NGRID
[8]    I←1
[9]    INEXT←1+NPERIODS⊤IPER
[10]   IMAX←(⍴X)[2]
[11]   DIM←DIMX[INEXT]
[12]   QLTD←(DIM,DIM)↑Q[INEXT;;]
[13]   LLTD←DIM↑L[INEXT;]
[14]   E←A[1;;]+RHO×(⍉N)+.×QLTD+.×N
[15]   NCY←(⍴C)[2]
[16]   NRY←(⍴C)[1]
[17]   TABLEAU
[18]   NR←(⍴TAB)[1]
[19]   NC←1+(⍴TAB)[2]
[20]   R←NCY+⍳NR
[21]   VAL←IMAX⍴0
[22]  STATELOOP:
[23]   S←DIMX[IPER]↑X[;I]
[24]   F←B[1;]+RHO×((2×QLTD+.×(MS←M+.×S))+LLTD)+.×N
[25]   G←RHO×(LLTD+QLTD+.×MS)+.×MS
[26]   XB←FILLX[IPER;]\S
[27]   D←XB[ 1 3 ]
[28]   D←0.00001⌈D
[29]   MAXIMIZE
[30]   VAL[I]←G+(F+E+.×Y)+.×Y
[31]   →STATELOOP×⍳IMAX≥I←I+1
[32]   DIM←DIMX[IPER]
[33]   LS
[34]   U←(,V∘.≤V←⍳DIM)
[35]   T←(DIM,DIM)⍴U\(-DIM+1)↓COEFF
[36]   Q[IPER;(⍳(⍴T)[1]);(⍳(⍴T)[2])]←0.5×T+⍉T
[37]   L[IPER;⍳⍴T]←T←DIM↑(-DIM+1)↑COEFF
[38]   →PERLOOP×⍳1≤IPER←IPER-1
[39]   →((CONVERGE K)=1)/0
[40]   →CLOOP×⍳ITERS≥K←K+1
      ∇
```

Figure 1.4 Listing of APL Function VFS

APL Name and Shape	Algebraic Name and Description	
INEXT, 1	i+1	Index of next period of the year.
IPER, 1	i	Counter for period of the year.
ITERS, 1	--	Maximum number of iterations in seeking convergence.
K, 1	--	Counter for iterations in seeking convergence.
LLTD, 4 2	L	Row of L referring to next period
NC, 1	--	Number of columns in quadratic programming tableau.
NCY, 1	--	Dimension of decision vector, number of columns in constraint matrix C.
NR, 1	--	Number of rows in quadratic programming tableau.
NRY, 1	--	Number of rows in constraint matrix C.
PERLOOP, 1	--	Label for start of loop over periods of year.
QLTD, 4 4 2 2	Q	Submatrix of Q referring to next period.
R (vector,size varies)	--	Indices of basic columns in quadratic programming algorithm.
S, 2 4	S	State point.
STATELOOP, 1	--	Label for start of loop on state points.
VAL, 9 81	V	Vector of values of cumulative value function at state points.

The function VFS is a direct implementation of the dynamic programming algorithm, as presented in Section 1.2.3. It calls the subordinate functions DRAW, STRUCTURE, BUILDX, TABLEAU, MAXIMIZE, LS, and CONVERGE.

DRAW is called at the beginning of execution to select the random sample of values of Φ_i .

STRUCTURE is called at the beginning of the loop on periods of the year to set up the arrays A, B, C, M and N. Thus, it covers Step 2 of Section 1.2.3.

BUILDX covers Step 3, the formation of the array of grid points.

TABLEAU prepares the initial tableau for the quadratic programming calculations, which are actually carried out in MAXIMIZE.

LS performs the least squares fit to obtain new Q and L.

CONVERGE compares the new Q array with the one stored in the last iteration, and takes the value 1 when convergence has been achieved.

1.3.5 Function SIMGRID

Figure 1.5 gives a listing of this function. The following is a glossary of important variables used in SIMGRID and not discussed above.

APL Name and Shape	Algebraic Name	Description
ANTLOOP, 1	--	Label for start of loop on antithetic variates.
FILLP, 6 4	--	Array of flags to expand Φ to maximum dimension of state space.
FILLX, 6 4	--	Array of flags to expand S to maximum dimension of state space.
KNT, 1	--	Pointer for array of random terms. Locates starting point for this year and period.
ND, 1	--	Dimension of state space next period, number of rows of M.
PH (vector, length depends on ITERS)	--	Random terms selected in DRAW.
PHI, 2 4	Φ	Random term in state transformation.

```
      ∇SIMGRID[⎕]∇
      ∇ SIGSQ SIMGRID ITERS
[1]    FILLX← 6 4 ⍴ 1 0 1 0 1 0 1 0 ,16⍴1
[2]    FILLP← 6 4 ⍴( 1 0 1 0 ),(16⍴1), 1 0 1 0
[3]    →L1×⍳∧/(ITERS,,SIGSQ)=(SITERS,,SSIGSQ)
[4]    DRAW
[5]   L1:
[6]    STSTATE←2↑MEANX[1;]
[7]    MEANX←STDX← 6 4 ⍴0
[8]    SGN←1
[9]   ANTLOOP:KNT←0
[10]   SGN←-SGN
[11]   S←STSTATE
[12]   K←1
[13]  YEARLOOP:
[14]   IPER←1
[15]  PERLOOP:
[16]   MEANX[IPER;]←MEANX[IPER;]+(4↑S)÷ITERS×2
[17]   STDX[IPER;]←STDX[IPER;]+(4↑S*2)÷ITERS×2
[18]   STRUCTURE IPER
[19]   INEXT←1+NPERIODS⊤IPER
[20]   DIM←DIMX[INEXT]
[21]   QLTD←(DIM,DIM)↑Q[INEXT;;]
[22]   LLTD←DIM↑L[INEXT;]
[23]   E←A[1;;]+RHO×(⍉N)+.×QLTD+.×N
[24]   NCY←(⍴C)[2]
[25]   NRY←(⍴C)[1]
[26]   TABLEAU
[27]   NR←(⍴TAB)[1]
[28]   NC←1+(⍴TAB)[2]
[29]   R←NCY+⍳NR
[30]   F←B[1;]+RHO×((2×QLTD+.×(MS←M+.×S))+LLTD)+.×N
[31]   G←RHO×(LLTD+QLTD+.×MS)+.×MS
[32]   ND←(⍴M)[1]
[33]   PHI←PH[KNT+⍳ND]×SGN
[34]   KNT←KNT+ND
[35]   XB←FILLX[IPER;]\S
[36]   D←XB[ 1 3 ]
[37]   D←0.00001⌈D
[38]   MAXIMIZE
[39]   S←(M+.×S)+(N+.×T)+PHI
[40]   →PERLOOP×⍳6≥IPER←IPER+1
[41]   →YEARLOOP×⍳ITERS≥K←K+1
[42]   →ANTLOOP×⍳SGN=¯1
[43]   STDX←((STDX-MEANX*2)×ITERS÷ITERS-0.5)*0.5
      ∇
```

Figure 1.5 Listing of APL Function SIMGRID

SGN, 1	--	Multiplier (± 1) to fix sign of random terms to achieve antithetic variates.
SIGSQ, 2 10	Σ^2	Input array describing performance of information system. See Section 1.3, Step 1a.
XB, 4	X	State variable expanded to maximum dimension.

The function SIMGRID is a direct implementation of the simulation algorithm presented in Section 1.2.4. It uses some of the same subordinate functions as VFS, namely STRUCTURE, TABLEAU, and MAXIMIZE. It uses the subordinate function DRAW to select the random terms.

1.3.6 Function USVFS

The listing of this function is in Figure 1.6. It does all the calculations of VFS, and simultaneously calculates QU, LU, KEF, and KEFU, as discussed in Section 1.2.5 For the least squares fit, USVFS calls the function LSU, and for the convergence check, it calls CONVERGEU; otherwise, it uses the same subordinate functions as VFS.

1.3.7 Function CONTERMS

This function is listed in Figure 1.7. It is a straightforward implementation of the algorithm described in Section 1.2.6.

1.3.8 Function EVAL

This function is listed in Figure 1.7. It is a straightforward implementation of the algorithm described in Section 1.2.8.

1.3.9 Function SIMSTATS

Figure 1.8 gives a listing of this function. The following is a glossary of important variables used in SIMSTATS and not discussed above.

```
      ∇USVFS[⎕]∇
      ∇ SIGSQ USVFS ITERS
[1]     FILLX← 6 4 ⍴ 1 0 1 0 1 0 1 0 ,16⍴1
[2]     FILLP← 6 4 ⍴ 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0
[3]     K←1
[4]   CLOOP:
[5]     IPER←6
[6]   PERLOOP:
[7]     STRUCTURE IPER
[8]     IPER BUILDX NGRID
[9]     I←1
[10]    INEXT←1+NPERIODS⊤IPER
[11]    IMAX←(⍴X)[2]
[12]    DIM←DIMX[INEXT]
[13]    QLTD←(DIM,DIM)↑Q[INEXT;;]
[14]    LLTD←DIM↑L[INEXT;]
[15]    E←A[1;;]+RHO×(⍉N)+.×QLTD+.×N
[16]    QLTDU←(DIM,DIM)↑QU[INEXT;;]
[17]    LLTDU←DIM↑LU[INEXT;]
[18]    EU←A[2;;]+RHO×(⍉N)+.×QLTDU+.×N
[19]    NCT←(⍴C)[2]
[20]    NRT←(⍴C)[1]
[21]    TABLEAU
[22]    NR←(⍴TAB)[1]
[23]    NC←1+(⍴TAB)[2]
[24]    R←NCT+⍳NR
[25]    VAL←IMAX⍴0
[26]    VALU←VAL
[27]  STATELOOP:
[28]    S←DIMX[IPER]↑X[;I]
[29]    F←B[1;]+RHO×((2×QLTD+.×(MS←M+.×S))+LLTD)+.×N
[30]    G←RHO×(LLTD+QLTD+.×MS)+.×MS
[31]    FU←B[2;]+RHO×((2×QLTDU+.×MS)+LLTDU)+.×N
[32]    GU←RHO×(LLTDU+QLTDU+.×MS)+.×MS
[33]    XB←FILLX[IPER;]\S
[34]    D←XB[1 3]
[35]    D←0.00001⌈D
[36]    MAXIMIZE
[37]    VAL[I]←G+(F+E+.×T)+.×T
[38]    VALU[I]←GU+(FU+EU+.×T)+.×T
[39]    →STATELOOP×⍳IMAX≥I←I+1
[40]    DIM←DIMX[IPER]
[41]    LSUS
[42]    U←(,V∘.≤V←⍳DIM)
[43]    T←(DIM,DIM)⍴U\(-DIM+1)↓COEFF
[44]    Q[IPER;(⍳(⍴T)[1]);(⍳(⍴T)[2])]←0.5×T+⍉T
[45]    L[IPER;⍳⍴T]←T←DIM↑(-DIM+1)↑COEFF
[46]    KEF[IPER]←¯1↑COEFF
[47]    TU←(DIM,DIM)⍴U\(-DIM+1)↓COEFFU
[48]    QU[IPER;(⍳(⍴TU)[1]);(⍳(⍴TU)[2])]←0.5×TU+⍉TU
[49]    LU[IPER;⍳⍴TU]←TU←DIM↑(-DIM+1)↑COEFFU
[50]    KEFU[IPER]←¯1↑COEFFU
[51]    →PERLOOP×⍳1≤IPER←IPER-1
[52]    →((CONVERGEU K)=1)/0
[53]    →CLOOP×⍳ITERS≥K←K+1
      ∇
```

Figure 1.6 Listing of APL Function USVFS

```
        ∇CONTERMS[⎕]∇
        ∇ SIGSQ CONTERMS ITERS
[1]     SD2←LOSSSHAPE RESHAPE SIGSQ
[2]     K←1
[3]    CLOOP:
[4]     IPER←6
[5]    PERLOOP:
[6]     DIM←DIMX[INEXT←1+6⊤IPER]
[7]     DSD←DIM↑SD2[;IPER]
[8]     U←,(⍳DIM)∘.=⍳DIM
[9]     QLTD←(DIM,DIM)↑Q[INEXT;;]
[10]    QLTDU←(DIM,DIM)↑QU[INEXT;;]
[11]    KON[IPER]←KEF[IPER]+RHO×KON[INEXT]+DSD+.×U/,QLTD
[12]    KONU[IPER]←KEFU[IPER]+RHO×KONU[INEXT]+DSD+.×U/,QLTDU
[13]    →PERLOOP×⍳1≤IPER←IPER−1
[14]    →CLOOP×⍳ITERS≥K←K+1
[15]    12 2 ⍴ADR×KON
[16]    12 2 ⍴ADR×KONU
        ∇
        ⋄

        ∇EVAL[⎕]∇
        ∇ EVAL ARG
[1]     ADR×KON[1]+ARG+.×L[1;]+ARG+.×Q[1;;]
[2]     ADR×KONU[1]+ARG+.×LU[1;]+ARG+.×QU[1;;]
        ∇
        ⋄
```

Figure 1.7 Listings of APL Functions CONTERMS and EVAL

```
      ∇SIMSTATS[⎕]∇
      ∇ SIGSQ SIMSTATS ITERS
[1]   MULT←⍉(4 6 ⍴AVG),[1](9 6 ⍴1),[1](2 6 ⍴AVG←1÷6),[1] 12 6 ⍴RHO*⍳6
[2]   FILLX← 6 4 ⍴ 1 0 1 0 1 0 1 0 ,16⍴1
[3]   FILLT← 6 5 ⍴ 1 1 0 1 0 1 1 1 1 1 1 1 0 1 0 1 1 0 1 0 1 1 1 1 1 1 1 0 1 1
[4]   FILLP← 6 4 ⍴ 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0
[5]   HD←' XCONS EXPTS XPLNT ICONS IPLNT RNDXS RNDXG RNDIS RNDIG XPRCE IPRCE REVNU XPRDC IPRDC'
[6]   HD←' XSTOK XGROW ISTOK IGROW',HD
[7]   HD← 27 6 ⍴HD,' ITRNC XGROS IGROS TTNET XTNET XCNET XPNET ICNET IPNET'
[8]   →L1×⍳∧/(ITERS,,SIGSQ)=(SITERS,,SSIGSQ)
[9]   DRAW
[10] L1:
[11]  MN←TP←VAR←(NPERIODS,27)⍴0
[12]  AMN←AVAR←27⍴0
[13]  'STARTING STATE'
[14]  SGN←1
[15]  STSTATE←⎕
[16] ANTLOOP:
[17]  KNT←0
[18]  SGN←-SGN
[19]  S←STSTATE
[20]  K←1
[21] YEARLOOP:
[22]  IPER←1
[23] PERLOOP:
[24]  STRUCTURE IPER
[25]  INEXT←1+NPERIODS⊤IPER
[26]  DIM←DIMX[INEXT]
[27]  QLTD←(DIM,DIM)↑Q[INEXT;;]
[28]  LLTD←DIM↑L[INEXT;]
[29]  E←A[1;;]+RHO×(⍉N)+.×QLTD+.×N
[30]  NCT←(⍴C)[2]
[31]  NRT←(⍴C)[1]
[32]  TABLEAU
[33]  NR←(⍴TAB)[1]
```

Figure 1.8 Listings of APL Function SIMSTATS

```
NC←1+(⍴TAB)[2]
R←NCT+⍳NR
F←B[1;]+RHO×((2×QLTD+.×(MS←M+.×S))+LLTD)+.×N
G←RHO×(LLTD+QLTD+.×MS)+.×MS
ND←(⍴M)[1]
PHI←PH[KNT+⍳ND]×SGN
KNT←KNT+ND
XB←FILLX[IPER;]\S
D←XB[1 3]
D←0.00001⌈D
MAXIMIZE
YB←FILLY[IPER;]\Y
PHB←FILLP[IPER;]\PHI
PRC←BETA+2×ALPHA×YB[1 4]
REV←PRC[1]×YB[2]
PCST←YB[3 5]×DELTA[IPER;]+YB[3 5]×GAMMA[IPER;]
TCST←YB[2]×OMEGA+YB[2]×TAU
GRS←YB[1 4]×BETA+YB[1 4]×ALPHA
NET←(B+A+.×Y)+.×Y
MN[IPER;]←MN[IPER;]+(TP[IPER;]←XB,YB,PHB,PRC,REV,PCST,TCST,GRS,NET)÷ITERS×2
VAR[IPER;]←VAR[IPER;]+(÷ITERS×2)×TP[IPER;]*2
S←(M+.×S)+(N+.×Y)+PHI
→PERLOOP×⍳6≥IPER←IPER+1
AMN←AMN+(TERM←+/[1] MULT×TP)÷ITERS×2
AVAR←AVAR+(TERM*2)÷ITERS×2
→YEARLOOP×⍳ITERS≥K←K+1
→ANTLOOP×⍳SGN=¯1
VAR←((VAR)-MN*2)×ITERS÷ITERS-0.5
AVAR←(AVAR-AMN*2)×ITERS÷ITERS-0.5
(32⍴' '),'MEANS',(48⍴' '),'STANDARD DEVIATIONS'
⍳0
'PERIOD',(8 0 ⍕⍳6),'   ANNUAL ',(8 0 ⍕⍳6),'  ANNUAL'
⍳0
HD,(8 1 ⍕⍉MN),(9 1 ⍕ 27 1 ⍴AMN),(27 1 ⍴' '), 8 2 ⍕⍉(0⌈VAR,[1] AVAR)*0.5
∇
```

Figure 1.8 Listings of APL Function SIMSTATS (continued)

APL Name and Shape	Algebraic Name	Description
AMN, 10	μ_A	Annual means of 27 quantities listed in Section 1.2.7, Step 1j.
AVAR, 10	AVAR	Annual variance of 10 quantities listed in Section 1.2.7, Step 1j.
FILLP, 6 4	--	Array of flags to expand Φ to maximum dimension of state space.
FILLX, 6 4	--	Array of flags to expand S to maximum dimension of state space.
FILLY, 6 5	--	Array of flags to expand Y to maximum dimension of decision space.
GRS, 2	STAT(i20,i21)	Gross welfare this period to United States and rest of world
HD, 27 6	--	Heading for printout.
MN, 6 16	μ_i	Accumulator for mean values by period of 27 quantities listed in Section 1.2.7, Step 1j.
MULT, 6 27	MULT	Multiplier defined in Section 1.2.7, Step 1d.
ND, 1	--	Dimension of state space next period, number of rows of M.
NET, 5	STAT(i22,…,i27)	Net welfare each of six categories this period.
PCST, 2	STAT(i17,i18)	Cost to producers of production planted this period in United States and rest of world.
PHI, 2 4	Φ	Random term in state transformation.
PHB, 4	--	Expansion of Φ to maximum dimension of state space.
PRC, 2	STAT(i14,i15)	Price of wheat this period in United States and rest of world.
REV, 1	STAT(i16)	Revenue to United States for exports shipped this period.

APL Name and Shape	Algebraic Name and Description	
SIGSQ, 2 10	Σ^2	Input array describing performance of information system.
TCST, 1	STAT(i19)	Transportation cost this period.
VAR, 6 16	VAR	Accumulator for variances by period of 27 quantities listed in Section 1.2.7, Step 1j.
XB, 4	STAT(i1,…,i4)	State variable expanded to maximum dimension.
YB, 5	STAT(i5,…,i9)	Decision variable expanded to maximum dimension.

The function SIMSTATS is direct implementation of the statistics algorithm presented in Section 1.2.7. It does not change the arrays MEANX and STDX as SIMGRID does, but simply prints out means and standard deviations of the 27 tracked quantities.

1.3.10 Subordinate Functions

The remaining functions are listed in Figure 1.9 through Figure 1.15.

BUILDX builds the grid of points as described in Section 1.2.3, Step 3. In case the grid so built includes any points with negative coordinates, it is shifted to avoid this condition, and NOTESHIFT is called to print a warning. This does not happen in the normal workings of the algorithm, but when the statistical parameters MEANX and STDX are very far from their final values, it may occur. Unless it persists in iterations close to convergence, there is no problem.

The function PIVOT is called from the quadratic programming algorithm MAXIMIZE, to do the pivot operation for each iteration. The left argument is the pivot row index, and the right argument is the pivot column index.

```
      ∇BUILDX[⎕]∇
      ∇ I BUILDX N;F;T;K;ROW;DIM;BMINX;SHIFT;TSHIFT
[1]     F←(12÷¯1+N×N)*0.5
[2]     BMINX←0.01
[3]     SHIFT←0
[4]     →((TSHIFT←((BMINX+((N−1)÷2)×F×STDX[I;1])−MEANX[I;1]))<0)/SKIP1
[5]     SHIFT←TSHIFT
[6]     NOTESHIFT
[7]   SKIP1:X←(1,N)⍴SHIFT+(MEANX[I;1])+(STDX[I;1])×F×(⍳N)−0.5×N+1
[8]     K←N
[9]     DIM←2
[10]  LOOP:SHIFT←0
[11]    →((TSHIFT←((BMINX+((N−1)÷2)×F×STDX[I;DIM])−MEANX[I;DIM]))<0)/SKIP2
[12]    SHIFT←TSHIFT
[13]    NOTESHIFT
[14]  SKIP2:ROW←SHIFT+MEANX[I;DIM]+STDX[I;DIM]×F×(⍳N)−0.5×N+1
[15]    T←( 3 1 2 )⍉(N,⍴X)⍴X
[16]    T←(1,(DIM−1),K←N×K)⍴T
[17]    T←(⍴T)[ 2 3 ]⍴T
[18]    X←T,[1](1,K)⍴ROW
[19]    →LOOP×DIMX[I]≥DIM←DIM+1
      ∇

      ∇CONVERGE[⎕]∇
      ∇ Z←CONVERGE K
[1]     'AT ITERATION' , 3 1 ⍕K
[2]     Z←(⎕←(⌈/|,Q−QPREV))<TOL×⌈/|,Q,TOL
[3]     QPREV←Q
[4]     →0
      ∇
```

Figure 1.9 Listings of APL Functions BUILDX and CONVERGE

```
∇CONVERGEU[⎕]∇
∇ Z←CONVERGEU K
'AT ITERATION' , 4 0 ⍕K
Z←(⎕←(⌈/|,(Q,QU)−(QPREV,QPREVU)))<TOL×⌈/|,Q,TOL
QPREV←Q
QPREVU←QU
→0
∇
⋄
∇NOTESHIFT[⎕]∇
∇ NOTESHIFT
→AFLAG×NEXT
NEXT: 'GRID SHIFTED DURING PERIOD' , 4 0 ⍕IPER
AFLAG←0
∇
⋄

∇PIVOT[⎕]∇
∇ I PIVOT J;IR;T
R[I]←J
MATRIX[I;]←T←MATRIX[I;]÷MATRIX[I;J]
MATRIX←MATRIX−(MATRIX[;J]×I≠⍳NR)∘.×T
∇
⋄
```

Figure 1.10 Listing of APL Functions CONVERGEU, NOTESHIFT and PIVOT

```
    ∇MAXIMIZE[⎕]∇
    ∇ MAXIMIZE;PC;PR;T;T1;T4;X;T6;T7;RPR;MATRIX;S;NEGY;NFS
[1]   NEGY←0
[2]   MATRIX←TAB,((⍴TAB)[1],1)⍴D,(−F),NRY⍴0
[3]   MATRIX←(⌹MATRIX[;R])+.×MATRIX
[4]  YCHEK:→(∧/T←1−(R≤NR−NRY)∧(MATRIX[;NC]<0))/YFEAS
[5]   T1←T⍳0
[6]   NEGY←R[T1]
[7]   PC←NEGY+NR
[8]   S←2×((MATRIX[T1;PC]÷MATRIX[T1;NC])≥0)−0.5
[9]   →PIVOTROW
[10] YFEAS:S←1−NEGY←0
[11] LCHECK:T←0>MATRIX[;NC]
[12]  T6←NR<R
[13]  T7←T∧T6
[14]  →(0=+/T7)/END
[15]  T1←T7⍳1
[16]  PC←R[T1]−NR
[17] PIVOTROW:X←MATRIX[;PC]÷(MATRIX[;NC]+MATRIX[;NC]=0)
[18]  T4←R≤NR−NRY
[19]  T4[T1]←1
[20]  PR←(T4×X)⍳(NFS←⌈/S×T4/X)
[21]  →NOFEAS×⍳NFS=0
[22]  RPR←R[PR]
[23]  PR PIVOT PC
[24]  →YCHEK×⍳RPR=NEGY
[25]  →LCHECK×⍳RPR>NR−NRY
[26]  PC←RPR+NR
[27]  →PIVOTROW
[28] NOFEAS:'NO FEASIBLE SOLUTION'
[29]  5÷NFS
[30] END:Y←(NC−1)⍴0
[31]  Y[R]←MATRIX[;NC]
[32]  Y←NCY↑Y
    ∇
```

Figure 1.11 Listing of APL Function MAXIMIZE

```
      ∇RESHAPE[⎕]∇
      ∇ C←B RESHAPE A;NEWR;NEWC;B1;B2
[1]     NEWR←(⍴B)[2]
[2]     NEWC←(⍴B)[3]
[3]     B1←(NEWR,NEWC)⍴(1,NEWR,NEWC)↑B
[4]     B2←(NEWR,NEWC)⍴(¯1,NEWR,NEWC)↑B
[5]     C←(0,(,A))[(⍴B1)⍴1+B2+(B1-(⍴B1)⍴1)×(⍴A)[2]]
      ∇
      ⋄

      ∇LOSSSHAPE[⎕]∇
      ∇ L←LOSSSHAPE
[1]     L← 1 1 1 1 1 1 2 1 1 1 1 2 1 2 2 2 2 1 1 2 2 2 2 1
[2]     L← 2 4 6 ⍴L, 6 7 8 9 10 5 6 1 2 3 4 5 0 7 8 9 10 0 0 1 2 3 4 0
      ∇
      ⋄
```

Figure 1.12 Listings of APL Functions RESHAPE and LOSSSHAPE

```
      ∇STRUCTURE[⎕]∇
      ∇ STRUCTURE I;M44;U
      M44←(⍳4)∘.=⍳4
      →(I-1)⌽L1,L2,L3,L3,L5,L6
     L1:M← 2 2 ⍴ 1 0 0 1
      N← 2 3 ⍴ ¯1 ¯1 0 0 1 ¯1
     L7:A←AIN
      B←BIN
      C← 2 3 ⍴ 1 1 0 0 0 1
      →0
     L2:M← 4 2 ⍴ 1 0 0 0 0 1 0 0
     L8:N← 4 5 ⍴ ¯1 ¯1 ,(5⍴0), 1 0 0 0 1 0 ¯1 ,(5⍴0),1
      U← 1 1 0 1 0
      A←U\U\[2]AIN
      B←U\BIN
      A[1;3;3]←A[2;3;3]←-GAMMA[I;1]
      A[1;5;5]←-GAMMA[I;2]
      B[1;3]←B[2;3]←-DELTA[I;1]
      B[1;5]←-DELTA[I;2]
      A[4;1;3]←A[4;3;1]←GAMMA[I;1]
      A[6;5;5]←GAMMA[I;2]
      C← 2 5 ⍴ 1 1 0 0 0 0 0 0 1 0
      →0
     L3:M←M44
      N← 4 3 ⍴ ¯1 ¯1 0 0 0 0 0 1 ¯1 0 0 0
      →L7
     L5:M←M44
      →L8
     L6:M← 2 4 ⍴ 1 1 0 0 0 0 1 1
      N← 2 4 ⍴ ¯1 ¯1 0 0 0 1 ¯1 1
      U← 1 1 1 0
      A←U\U\[2]AIN
      B←U\BIN
      A[1;4;4]←-GAMMA[I;2]
      B[1;4]←-DELTA[I;2]
      A[6;4;4]←GAMMA[I;2]
      C← 2 4 ⍴ 1 1 0 0 0 0 1 0
      →0
      ∇

      ∇TABLEAU[⎕]∇
      ∇ TABLEAU;ZN;ZNM;IN;INM
      IN←(⍳NRT)∘.=⍳NRT
      INM←(⍳NRT+NCT)∘.=⍳NRT+NCT
      TAB←C,[1](2×E),[1]ZNM←(NRT,NCT)⍴0
      TAB←TAB,IN,[1](⍉ZNM),[1]ZN←(NRT,NRT)⍴0
      TAB←TAB,ZN,[1](⍉C),[1]IN
      TAB←TAB,(ZNM,ZN),[1]INM
      →0
      ∇
```

Figure 1.13 Listings of APL Functions STRUCTURE and TABLEAU

```
      ∇DRAW[⎕]∇
      ∇ DRAW
[1]   ⎕RL←123456789
[2]   SITERS←ITERS
[3]   SSIGSQ←SIGSQ
[4]   SD←(LOSSSHAPE RESHAPE SIGSQ)*0.5
[5]   K←1
[6]   SD2← 4 6 ⍴0
[7]  YRLP:IPER←1
[8]  PRLP:DIM←DIMX[1+6⊤IPER]
[9]   PHI←(DIM↑SD[;IPER])×(1.5*0.5)×(¯2+?DIM⍴3)
[10]  SD2[⍳DIM;IPER]←SD2[⍳DIM;IPER]+PHI*2
[11]  →PRLP×⍳6≥IPER←IPER+1
[12]  →YRLP×⍳ITERS≥K←K+1
[13]  SD2←(SD2÷ITERS-0.5)*0.5
[14]  PH←⍳0
[15]  ⎕RL←123456789
[16]  SD←SD×SD÷SD2
[17]  K←1
[18] YLP:IPER←1
[19] PLP:DIM←DIMX[1+6⊤IPER]
[20]  PH←PH,(DIM↑SD[;IPER])×(1.5*0.5)×(¯2+?DIM⍴3)
[21]  →PLP×⍳6≥IPER←IPER+1
[22]  →YLP×⍳ITERS≥K←K+1
      ∇
      •
```

Figure 1.14 Listing of APL Function DRAW

```
      ∇LS[⎕]∇
      ∇ LS;T;V;K;I;Y;TE;E
[1]     K←2!2+DIM
[2]     T←(K,IMAX)⍴I←1
[3]     V←,(⍳DIM)∘.≤⍳DIM
[4]     E←(+/[2]X)÷IMAX
[5]   LOOP:Y←X[;I]
[6]     Y←Y÷E
[7]     T[;I]←(V/,Y∘.×Y),Y,1
[8]     →LOOP×⍳IMAX≥I←I+1
[9]     TE←(V/,E∘.×E),E,1
[10]    COEFF←(⌹T+.×⍉T)+.×T+.×VAL
[11]    COEFF←COEFF÷TE
      ∇

      ∇LSUS[⎕]∇
      ∇ LSUS;T;V;K;I;Y;TE;E
[1]     K←2!2+DIM
[2]     T←(K,IMAX)⍴I←1
[3]     V←,(⍳DIM)∘.≤⍳DIM
[4]     E←(+/[2]X)÷IMAX
[5]   LOOP:Y←X[;I]
[6]     Y←Y÷E
[7]     T[;I]←(V/,Y∘.×Y),Y,1
[8]     →LOOP×⍳IMAX≥I←I+1
[9]     TE←(V/,E∘.×E),E,1
[10]    COEFF←(MAT←(⌹T+.×⍉T)+.×T)+.×VAL
[11]    COEFFU←MAT+.×VALU
[12]    COEFF←COEFF÷TE
[13]    COEFFU←COEFFU÷TE
      ∇
```

Figure 1.15 Listings of APL Functions LS and LSUS

Function DRAW is called from SIMGRID and SIMSTATS to select the vector of random terms as described in Section 1.2.4 d.

Functions CONVERGE and CONVERGEU simply determine whether convergence has been achieved in VFS and USVFS respectively.

Function LS calculates the least squares fit described in Step 8 (a and b) of Section 1.2.3. Function LSUS does the same calculation for the United States value functions as described in Step 8 (a and b) of Section 1.2.5.

Function NEWGRID calculates the new statistical parameters after a run of SIMGRID. It is called from ITERATE.

2. VARIATIONS IN INPUT DATA

As shown in the diagram of Figure 1.1, there are two categories of input data used for the value of information calculations. There are the economic parameters (discount rate, elasticities, timing assumptions) used in both the dynamic programming calculations and the simulations. Then there are the data describing the information systems used directly in the simulations, but also used indirectly in the dynamic programming calculations, since these require the statistical parameters (MEANX, STDX) calculated by the simulations.

Changes in the economic parameters are easily made by replacing the appropriate lines of INITIAL. To correctly make changes in the description of either the current or the improved information system, one must understand how these descriptions are related to the assumed production estimate accuracies.

This relationship is portrayed in Figure 3.1. The Σ^2 array describing an information system is formed by a difference operation from the mean squared errors in production estimates by time of year, together with the mean squared error of the a priori production estimate. Let $i = 1$ for the United States, $i = 2$ for the rest of the world. Let ε_{ij}^2, $j = 1, 2, \ldots, 12$, be the mean squared errors in production estimates for the United States and the rest of the world at two month intervals from June before planting $(j = 1)$ to April after harvest $(j = 12)$. We assume $\varepsilon_{i12}^2 = 0$, since regardless of what is published, the "market" must discover the truth as the supply is exhausted.

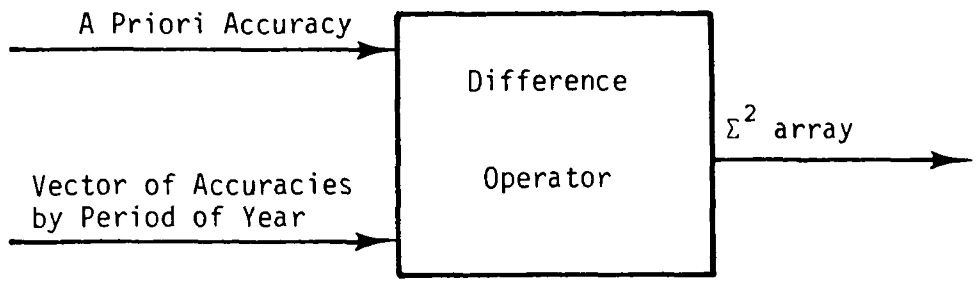

Figure 2.1 Calculation of Σ^2 array—Either Current or Improved System

In the case of the current system, we have obtained estimates of the ε_{ij}^2's from a study of published statistics. But these estimates are not based on the assumption that the true annual production is ever known with perfect accuracy. Built into our ε_{ij}^2's is the assumption that a certain residual mean squared error r_i^2, $i = 1, 2$ remains in the final published estimates. If we were to replace this assumption with the assumption of another value $(r_i')^2$ for the residual error, then the ε_{ij}^2's would be replaced with

$$\varepsilon_{ij}^2 - r_i^2 + (r_i')^2, \quad j = 1, \ldots, 11; \quad i = 1, 2,$$

with ε_{i12}^2 remaining 0.

For the first few periods, ε_{ij}^2 gives the mean squared error of the <u>a priori</u> estimate, since there is no information available specific to the crop about to be planted. For our wheat calculations, this applies up to $j = 6$. Now we form

$$\sigma_{ij}^2 , \quad i = 1 , 2 \; ; \; j = 1 , 10 , \text{ by}$$

$$\sigma_{ij}^2 = \varepsilon_{i,j+1}^2 - \varepsilon_{i,j+2}^2 .$$

Since $\varepsilon_{ij}^2 = \varepsilon_{i1}^2$ for $j = 1, 2, \ldots, 6$, the first few σ_{ij}^2's are 0, namely for $j = 1, 2, 3, 4$.

Notice that if the residual error assumption is changed from r_i to r_i', each σ_{ij}^2 remains the same except for σ_{i10}, $i = 1 , 2$, which are simply increased by $(r_i')^2 - r_i^2$, $i = 1 , 2$.

In the case of an improved information system, the mean squared error estimates ε_{ij}^2 are not based on published statistics, but on an analysis of the sampling and measurement methods used. Thus, the above concept of "residual error" does not apply, except as it affects the <u>a priori</u> mean squared errors ε_{ij}^2 for the first few j. In advance of the system's measurement of the growing crop, for $j < J$ (some appropriate J), we have the same ε_{ij}^2 as for the current system. Therefore, if we change the residual error assumption of the current system from r_i to r_i', σ_{iJ}^2 changes $(i = 1,2)$, being increased by $(r_i')^2 - r_i^2$. The other σ_{ij}^2's are unchanged.

ECON *Corporate Headquarters:*

Princeton, New Jersey
Telephone 609-924-8778

Western Office:

San Jose, California
Telephone 408-249-6364