NASA Contractor Report 3286

Wheat Forecast Economics Effect Study

R. K. Mehra, R. Rouhani,
S. Jones, and I. Schick

CONTRACT NAS5-25463
MAY 1980

NASA

NASA Contractor Report 3286

Wheat Forecast Economics Effect Study

R. K. Mehra, R. Rouhani,
S. Jones, and I. Schick
Scientific Systems, Inc. ($S^2 I$)
Cambridge, Massachusetts

Prepared for
Goddard Space Flight Center
under Contract NAS5-25463

NASA
National Aeronautics
and Space Administration

Scientific and Technical
Information Office

1980

ACKNOWLEDGEMENTS

The technical monitors for this contract were Dr. David B. Wood and Mr. Ahmed Meer. Dr. Alex Levis of Systems Control, Inc. and Prof. Pravin Varaiya of U. C. Berkeley acted as consultants. Their contributions are gratefully acknowledged.

TABLE OF CONTENTS

		page
1.	INTRODUCTION	1
2.	THEORETICAL FORMULATION	3
	2.1 Introduction	3
	2.2 ECON Model	3
	2.3 Generalized Model	9
	2.4 Stochastic Asepcts of the Model	12
	2.5 Approximation by Discrete States	15
3.	ALGORITHMS	19
	3.1 LQG (Linear-Quadratic-Gaussian) Method	19
	3.2 Dynamic Programming	20
	3.3 Introduction to Markov Programming	21
	3.4 Structure of the Program	25
4.	A SIMPLIFIED MODEL: ONE COUNTRY - TWO PERIODS	28
	4.1 The Model	28
	4.2 Quality of Information	30
	4.3 Incremental Value Function	31
	4.4 Discretization - Probability Matrix P(y)	35
	4.5 Data	38
	4.6 Numerical Results	39
	4.7 Remarks	44
5.	CONCLUSION	46
REFERENCES		48

Appendix A: Documentation for the Crop Information Value Program

Appendix B: Optimal and Suboptimal Stationary Controls for Markov Chains

Appendix C: A Differential Theory of Markov Control

1. INTRODUCTION

The objective of this study is to find an algorithm which accurately and efficiently estimates production and consumption patterns in the wheat market when various information conditions are available to the participants. Such an algorithm could be used to estimate, given a suitable definition of overall "welfare," the net value to society of an improved wheat forecasting program such as the NASA LANDSAT system. Of course, the accuracy of the value estimates depends on both the information supplied to the program, viz.: the economic model, and the definition of "welfare," as well as the operation of the algorithm itself. An economic model of the wheat sector which is suitable for this problem has been developed by ECON [1], and our work has been limited to 1) precise formulation of the model within a stochastic control framework, 2) choice and development of a suitable algorithm to solve the model.

The theoretical foundation on which our work is based is the results of stochastic control theory (Witsenhausen [2]) and Markov Decision Theory (Howard [3], Schweitzer [4], Odoni [5], and Varaiya [6]), including some new theoretical results on discounted Markov Processes which will be presented here (and also in Jones [7]). The major theoretical steps in our work have been:

1) Generalization of the ECON model
2) Definition of stochastic control problem

3) Definition of infinite-horizon discounted rewards

4) Definition of information states

5) Approximation as finite-state Markov chain,

and these results will be presented in Section 2. The choice of algorithm will be discussed in Section 3. We have applied the algorithm to a simplified two-period, one country model, both for debugging purposes, and also to get a feel for the behavior of the algorithm in a more tractable problem than the many-thousand state Markov chain model based on larger ECON model. For a fixed quantization of the state-space, convergence is relatively fast and monotonic, and we have every reason to believe that this rate of convergence will be nearly achieved by the larger model. The memory requirement of the algorithm is a small multiple of the total number of states. Then the memory requirement increases with the number of discrete states. Our program is written for a generalized ECON model which includes multiple crops, countries, and harvest times, so no reprogramming will be necessary for extended models.

In Section 4, a one-country, two period example is treated in detail and problems concerning the use and convergence of algorithms, and a comparison with the ECON results for the same example are considered. A number of interesting differences between the results are also presented.

Three appendices are included at the end of the report to provide further details on the Markov programming algorithm.

2. THEORETICAL FORMULATION

2.1 Introduction

The first step in finding a suitable algorithm is posing the problem in the proper theoretical framework, so that convergence and uniqueness of the solution can be assured. Ultimately, we will formulate the model as a finite-state Markov chain, where the states represent the information available to consumers and producers about stocks and crops, namely their estimated values. But before we can logically get to this stage, we must start at the foundation, the real-world variables and model. We will then show how this is explicitly simplified to the information-state model, and then to the finite-state model, which lends itself to computer solution.

We begin by reexamining the ECON model in terms of real-world quantities and relationships.

2.2 ECON Model

In the ECON model there are two types of grain: grain which is growing on a farm and has not yet been harvested, and grain which has been harvested, but not yet consumed; e.g., in transit, in storage reserves, etc. Let us call these grain types 1 and 2, respectively.

Grain can exist, within the ECON model's discrimination, in one of two places: in the US, or in the rest of the world (ROW). The ECON model, then, is concerned with four real-world quantities, and although the state-space is not the quantities themselves but only estimates of them, let us momentarily take the state variables to be the real-world quantities: They are:

Type 1 grain in US, x_2 (unharvested, in the ground)

Type 2 grain in US, x_1 (harvested, unconsumed)

Type 1 grain in ROW, x_4 (unharvested, in the ground)

Type 2 grain in ROW, x_3 (harvested, unconsumed)

Now suppose that the world wheat market is in some state, which is a value $x(t) \in E_+^4$ of $(x_1\ x_2\ x_3\ x_4)$*. Consumers, producers and exporters have access to certain limited imperfect public information about x(t), which we can call I(t), on which they base their consumptions (y_1 in US, y_4 in ROW), plantings (y_3, y_5), and exports from US to ROW (y_2). The state of the wheat market at t+1 is then a function of x(t), y(t), and some random disturbances v(t):

$$x(t+1) = f(x(t),y(t),v(t),t).$$

f is, for any t, a linear function in x, y and v, due to the simple additive and subtractive nature of consumption and production. Specifically, the equations are:

$$\text{(US)} \qquad x_1(t+1) = \begin{cases} x_1(t) - y_1(t) - y_2(t) + v_1(t) & \text{non-harvest period} \\ x_1(t) - y_1(t) - y_2(t) + v_1(t) + x_2(t) & \text{harvest period} \end{cases}$$

$$\text{(ROW)} \qquad x_3(t+1) = \begin{cases} x_3(t) - y_4(t) + y_2(t) + v_3(t) & \text{non-harvest period} \\ x_3(t) - y_4(t) + y_2(t) + v_3(t) + x_4(t) & \text{harvest period} \end{cases}$$

$$\text{(US)} \qquad x_2(t+1) = \begin{cases} 0 & \text{post harvest and pre-planting} \\ x_2(t) + v_2(t) & \text{non-planting, pre harvest} \\ x_2(t) + y_3(t) + v_2(t) & \text{planting, pre harvest} \end{cases}$$

*E_+^4 denotes the positive quadrant of the 4 dimensional Euclidean space.

(ROW) $$x_4(t+1) = \begin{cases} 0 & \text{post harvest, pre-planting} \\ x_4 + v_4(t) & \text{non-planting, pre harvest} \\ x_4(t) + y_5(t) + v_4(t) & \text{planting, pre harvest} \end{cases}$$

or, for suitable choice of matrices M and N (which will have either 0, 1, -1 as elements):

$$x(t+1) = M(t)x(t) + N(t)y(t) + v(t)$$

The following inequalities restrict the choice of y:

1. $y \geq 0$
2. $y_3 = 0$ during nonplanting periods in US
3. $y_5 = 0$ during nonplanting periods in ROW
4. $y_1 + y_2 \leq x_1$
5. $y_4 \leq x_3$

Inequality (5) takes into account the transportation lag for exports of about one month. It should be noted from the above equations that the matrix N could be eliminated by defining $u_1 = -y_1 - y_2$, $u_3 = -y_4 + y_2$, $u_2 = y_3$, and $u_4 = y_5$ so that

$$x(t+1) = M(t)x(t) + u(t) + v(t)$$

Let us now determine the constraints on u. From the constraints on y_3, y_5 we evidently have:

1'. $$u_2 \in \begin{cases} [0,\infty] & \text{during planting period in US} \\ [0] & \text{during nonplanting period in US} \end{cases}$$

2'. $$u_4 \in \begin{cases} [0,\infty] & \text{during planting periods in ROW} \\ [0] & \text{during nonplanting periods in ROW} \end{cases}$$

The constraints in u_1 can be found by rewriting inequality (4) with the substitution $y_1 = -u_1 - y_2$:

$$y_1 + y_2 = -u_1 - y_2 + y_2 = -u_1 \le x_1$$

so $u_1 \ge -x_1$. An upper bound on u_1 is obtained from the inequalities $y_1 \ge 0$ and $y_2 \ge 0$:

$$y_1 = -u_1 - y_2 \ge 0 \rightarrow u_1 \le -y_2$$

$$y_2 \ge 0 \rightarrow -y_2 \le 0 \rightarrow u_1 \le 0$$

so

3'. $\quad u_1 \in [-x_1, 0]$

For the constraint on u_3 consider the following four inequalities:

$$y_1 \ge 0$$

$$y_4 \ge 0$$

$$y_2 \ge 0$$

$$y_4 \le x_3$$

Rewrite them in terms of u and y_2:

$$-u_1 - y_2 \ge 0$$

$$y_2 - u_3 \ge 0$$

$$y_2 \ge 0$$

$$y_2 - u_3 \le x_3$$

Since $u_1 \geq -x_1$, we conclude from the first inequality that $y_2 \leq x_1$, and from the other three we then have

4'. $$y_2 \in [\max(0,u_3), \min(x_1,x_3+u_3)]$$

It is easily seen that for the interval in 4' to be nonempty, u_3 must be between $-x_3$ and x_1:

5'. $$u_3 \in [-x_3,x_1]$$

The constraints 1' - 5' in u and y_2 are equivalent to the constraints 1 - 5 in y, but the dimension of the control space has been reduced by one, and the state transition equation has been simplified.

The inequality constraints present a major hurdle in solving the problem; two other difficult areas are defining the information $I_t(x)$ available to the market, and the statistics of v(t). No reliable estimates of the statistics of v(t) are available, and the principal ingenuity of the ECON formulation, although not entirely successful, was to circumvent the need for such statistics. We will see, upon careful derivation of the ECON approach, that there is actually no way around this problem. Reasonable assumptions must be made, and stated explicitly for scrutiny.

$I_t(x)$ presents problems because the information actually available to the market, a history of controls and state observations, has arbitrarily large dimension. From separation [2] we know that this information can be reduced without loss of optimality to a probability distribution $p_x(x(t)|I_t)$, but we still must choose a consumption law which is a function of a probability function. We will follow ECON in defining $I_t(x)$ to be the "best" estimate of $x(t)$ given past observation. Since $x(t)$ is governed by time-varying linear equations, we know that the "best" estimate is a Kalman filter estimate which we shall denote $\hat{x}(t)$. Because certainty equivalence does not hold (due to the inequality constraints), the market <u>cannot</u> act optimally given only $\hat{x}(t)$. But it is a reasonable assumption if we must keep the closed-loop state dimension to a minimum. In fact, ECON took only $\hat{x}(t)$ to be the state, and this is technically not a "state", since the statistics of $\hat{x}(t+1)$ cannot be determined from $\hat{x}(t)$ alone. Actually the state-space must be extended to $(\hat{x},P)$ for the state quality to remain, where P is the covariance matrix of $\hat{x}$. We will discuss this issue in more detail a little later.

To actually turn the ECON Model into a living and breathing economic organism, we must postulate the mechanism by which $y(t)$ is chosen, i.e., how the consumer, producer and exporter actually behave. It is at this point where economics per se enters, and we must hope that the economic assumptions are strong enough to withstand the additional battering of approximations in solving the stochastic control problem.

Let us summarize the economic assumptions which directly affect the problem formulation. It is assumed that $y(t)$ is a function of $\hat{x}(t)$ and t which optimizes some properly defined overall welfare measure. That is, consumers, producers and exporters are "optimal controllers" of overall welfare.

Let $F(t, x(t), y(t))$ be a measure of overall welfare at time t. Then participants behave to maximize their overall future welfare, discounted by a factor ρ each period of time. That is, $y(\hat{x},t)$ is chosen to maximize:

$$W^{\rho}(x(t)) = E[\sum_{t'=t}^{\infty} \rho^{(t'-t)} F(t', x(t'), y(t'))]$$

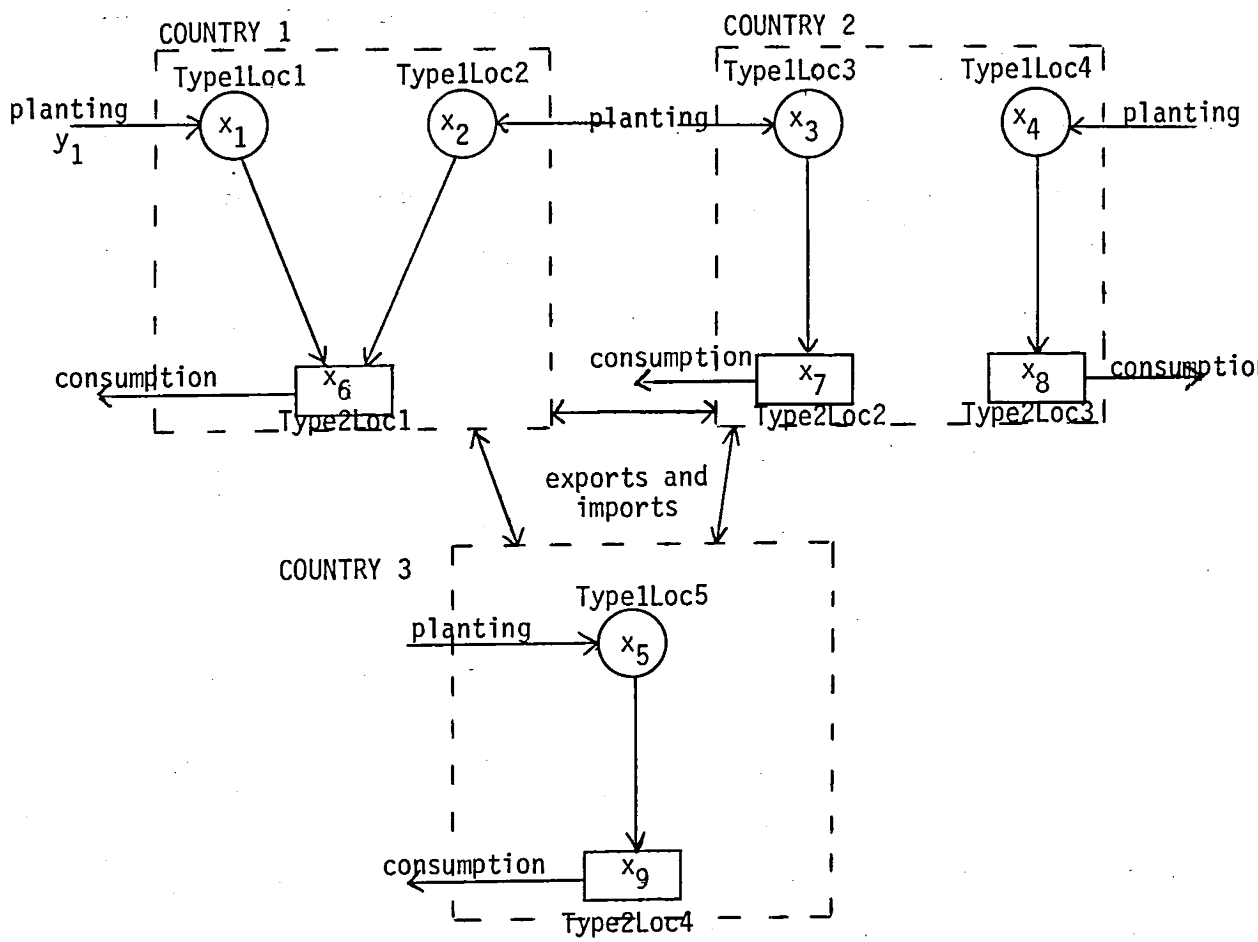
which we call the "discounted future welfare." This quantity depends on the starting state $x(t)$.

We will defer questions of uniqueness of existence of a solution of the above problem to the finite-state formulation, where the results are quite clear. First let us define our generalized model, and reexamine the state-space representation problem.

2.3 Generalized Model

In the interest of future research, we have generalized the ECON model to an arbitrary number of countries, with arbitrary planting times, harvest times, and fractional harvests. The general program has not been substantially harder to write, but it has enabled us to observe the behavior of the algorithm on smaller and more tractable models. Also, it should allow future researchers to examine larger models without any reprogramming.

A schematic of the physical model is shown below.



We follow the same convention that Type 1 grain is growing but not yet harvested, Type 2 grain is harvested but not yet consumed. Notice that the definition of x_1, x_2 is switched around from the ECON model. Circles in the above diagram are locations of Type 1 grain; we call them <u>aggregated crops</u>. The number of aggregated crops is arbitrary. In the ECON model there are only two aggregated crops: in the US and in ROW. Squares in the above diagram are locations of Type 2 grain; we call them aggregated bins. In this model, there is a state variable x_i for each circle and square, representing the amount of Type 1 or Type 2 grain at that location. Organization into countries is arbitrary.

The dynamics are analogous to the ECON model, except for harvesting, which is more general here. Planting takes place during restricted periods of the year, and the amount of planted grain u_i is simply added to the amount already existing x_i, with random variation $v_i(t)$ due to weather and other uncertainties:

$$x_i(t+1) = x_i(t) + u_i(t) + v_i(t) \qquad \begin{array}{l} i \in \text{aggregated crop} \\ t \text{ a planting season of } i \end{array}$$

At the end of the harvest season we simply put:

$$x_i(t+1) = 0 \qquad \begin{array}{l} i \in \text{aggregated crop} \\ t \text{ post-harvest season} \end{array}$$

Grain is harvested over a sequence of periods, and the fraction of the total crop harvested at period t is defined as hfr(t). Let j be an aggregated bin, i be an aggregated crop feeding j, and u_j be the net result during period t of all consumption, imports and exports, analogously to our previous definition. Then

$$x_j(t+1) = x_j(t) + u_j(t) + hfr_i(t) \times x_i(t)$$

During non-harvest season, $hfr_i = 0$. The constraints on the controls u are somewhat complicated and depend on the import-export assumptions, but the inequality constraints will have similar form to constraints 1' - 5' in Section 2.1

The state dynamic equation is, with suitable choice of M(t),

$$x(t+1) = M(t)x(t) + u(t) + v(t)$$

2.4 Stochastic Aspects of the Model

Up to this point we have avoided precise formulation of the probabilistic aspect of the model, because there are many problems and it is best to bring the discussion of them together in one section. It is now time to put the stochastic problem on a firm footing.

Controls, representing the behavior of consumer, producers, and exporters, are chosen according to a noisy state observation. We will assume that the information set $I(t) = (z(0), u(0), z(1),\ldots, u(t-1), z(t))$ is available where

$$z(t) = x(t) + w(t)$$

and $w(t)$ are zero-mean independent Gaussian random variables. If $v(t)$ is Gaussian also, then the optimal estimate of $x(t)$ given $I(t)$ is just the Kalman estimate of $x(t)$, which we call $\hat{x}(t|t)$.

We make the additional assumption that

$$u(t) = g(t, \hat{x}(t|t))$$

Although $\hat{x}(t|t)$ is an optimal estimate, $\hat{x}(t|t)$ may not contain enough information about the probability distribution of $x(t)$ given $I(t)$ to make an optimal choice of control. Nevertheless the problem quickly becomes intractable if we allow higher moments.

We can now write the stochastic equations of the economic system:

1. $x(t+1) = M(t)x(t) + u(t) + v(t)$

2. $z(t) = x(t) + w(t)$

3. $I(t) = (z(0), u(0), z(1), \ldots, u(t-1), z(t))$

4. $\hat{x}(t|t) = E(x(t)|I(t))$

5. $u(t) = g(t, \hat{x}(t|t))$

6. $E\{v(t)\ v^T(t)\} = Q$

7. $E\{w(t)\ w^T(t)\} = R$

Under the assumption that v and w are independent and Gaussian (3) and (4) can be replaced with the appropriate Kalman filter equations:

3'. $\nu(t+1) = z(t+1) - \hat{x}(t+1|t) = z(t+1) - M(t)\hat{x}(t|t) + u(t)$

4'. $\hat{x}(t+1|t+1) = M(t)\hat{x}(t|t) + u(t) + K(t)\nu(t)$

where K(t) is the Kalman filter gain which depends on Q, R and t, but not on the observations z(t) or controls u(t). Let us write K(Q,R,t) to be more explicit. It turns out that ν is independent from $\hat{x}(t|t)$ and is Gaussian with mean 0 and covariance $K_{\nu\nu}(Q,R,t)$. Thus equation 4' can be written

4". $\hat{x}(t+1|t+1) = M(t)\hat{x}(t|t) + u(t) + \phi(t)$

where $\phi(t) = K(t)\nu(t)$, and 4" has the state property. In other words, it is not necessary to know the true state x(t) in addition to $\hat{x}(t|t)$ to determine the statistics of $\hat{x}(t+1|t+1)$; $\hat{x}(t|t)$ alone will do.

The problem now remains to determine the statistics of the random variable $\phi(t)$. $E\{\phi(t)\} = 0$ since $E\{\nu(t)\} = 0$, and $E(\phi(t_1)\phi'(t_2)) = 0$ if $t_1 \neq t_2$. Let $K_{\phi\phi}(t) = E(\phi(t)\phi'(t))$. As we mentioned earlier, $K_{\phi\phi}(t)$ depends only on R and Q, but we can obviate the need for Q if the state error covariances are known:

$$P(t) \triangleq \mathrm{cov}\{x(t) - \hat{x}(t|t-1)\}$$

The standard Kalman filter equations then give:

$$K_{\nu\nu}(t) = P(t) + R(t)$$
$$K(t) = P(t)[P(t+1) + R]^{-1}$$

and thus

$$K_{\phi\phi}(t) = P(t)[P(t) + R]^{-1}P(t)$$

Thus the covariance of ϕ, $K_{\phi\phi}(t)$, can be determined either from the pair (Q,R) or from (R,P(t)), but <u>not from P(t) alone</u>, unless some additional assumptions are made. For example, if the wheat growing process is assumed to be of "random walk" nature, that is, the uncertainties are represented as a sequence of random weather influences represented by zero means random variable e(T-i) as follows:

$$\hat{x}(T|T-i+1) = \hat{x}(T|T-i) + e_{T-i+1}, \quad i = 1,\ldots,12$$

where $\hat{x}(T|T-i+1)$ is the estimate of x(T) at time T-i+1, e_{T-i+1} is the additional information available at time T-i+1 on the final yield x(T). e_{T-i+1} can be seen as innovation sequence.

In such a case $K_{\phi\phi}(t)$ can be inferred from the knowledge of P(t) and the statistics of e_{T-i}.

We now see that the Kalman filter evolution can be logically separated from the real system since the noise term $\phi(t)$ is white and independent of

$\hat{x}$. Thus we can restrict our attention to the state equations (for simplicity $\hat{x}(t)$ denotes $\hat{x}(t|t)$):

4". $\hat{x}(t+1) = M(t)\hat{x}(t) + u(t) + \phi(t)$

5. $u(t) = g(t,\hat{x}(t))$

The control $u(t) = g(t,\hat{x}(t))$ is chosen to maximize the discounted future welfare

$$W^{\rho}(x(t)) = E \sum_{t'=t}^{\infty} \rho^{(t'-t)} F(t', x(t'), y(t'))$$

Since $x(t)$ is not known to the economy, it is logical to replace $x(t)$ with $\hat{x}(t)$:

8. $$W^{\rho}(x(t)) = E \sum_{t'=t}^{\infty} \rho^{(t'-t)} F(t', \hat{x}(t'), y(t'))$$

Equations 4", 5 and 8 constitute a stochastic control model. In Sections 2.5 and 3 we will discuss several mathematical methods for solving these problems and finding optimal controls.

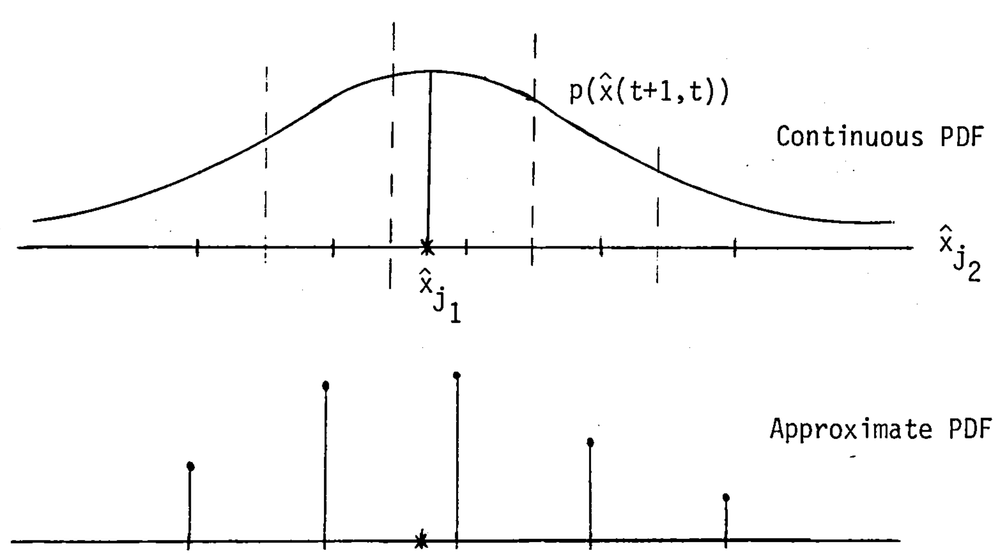
2.5 Approximation by Discrete States

For solution on the computer, it is necessary to approximate the model by discrete states. The method by which this approximation is made is very important and affects the results obtained, but discrete models are very well understood, good algorithms exist for solving them, and the questions of existence and uniqueness of their solutions answered easily.

Suppose that the set of possible states $\hat{x}(t)$ is finite. For the agricultural model, M(t) is periodic, and g is also periodic in t, so it is logical to choose the discrete sets $\{\hat{x}\}_t$ to be periodic also, since the steady state solution will be periodic.

Let s be the total number of discrete states through one entire cycle and let a state be uniquely specified by an index j, $1 \le j \le s$. We must choose a matrix $\{P_{j_1,j_2}\}$ which accurately reflects the probability that $\hat{x}(t+1) = \hat{x}_{j_2}$ if $\hat{x}(t) = \hat{x}_{j_1}$.

The continuous probability distribution of $\hat{x}(t+1)$ given $\hat{x}(t) = \hat{x}_{j_1}$ is the same as that of $M(t)\hat{x}_{j_1} + \phi(t) + g(t,\hat{x}_{j_1})$ where $M(t)\hat{x}_{j_1} + g(t,\hat{x}_{j_1})$ is an additive constant and $\phi(t)$ is a zero-mean Gaussian random variable with covariance matrix $K_{\phi\phi}(t)$. The problem is to divide the area under the p.d. curve for $\hat{x}(t+1)$ between the possible $\hat{x}_{j_2}$. Those states $\hat{x}_j$ which are defined at a different time period in the cycle receive no probability. For those defined at t, we might use an integration between midpoints as illustrated below. We will discuss methods in more detail in the section on Algorithms.



Whatever the approximation scheme, the resulting probability transition matrix P depends on the feedback control g.

For any choice of g, P(g) will be a <u>cyclic transition matrix</u>. For these type of matrices there exist for any $\hat{x}_j$ a limiting average state probability $\pi_j(g)$, defined as

$$\pi_j = \lim \frac{1}{T+1} \sum_{t=0}^{T} \Pr(\hat{x}(t) = x_j)$$

In [7], a more general concept is defined which is directly applicable to <u>discounted</u> Markov programming problems:

$$\Pi^p_{j_1 j_2} = \sum_{t=0}^{\infty} \rho^t \Pr(\hat{x}(t) = x_{j_2} | \hat{x}(0) = x_{j_1})$$

for if we define a discrete version of W^p to be

$$W^p_{j_1} = \sum_{t=0}^{\infty} \rho^t \sum_{j_2=1}^{s} \Pr(\hat{x}(t) = x_{j_2} | \hat{x}(0) = x_{j_1}) \times F(t, \hat{x}_{j_2}, g(t, \hat{x}_{j_1}))$$

then (in matrix notation):

$$W^p = \Pi^p k$$

where $k = (F(1, \hat{x}_1, g(1, \hat{x}_1)), F(1, \hat{x}_2, G(1, \hat{x}_2)), \ldots, F(nper, \hat{x}_s, g(nper, \hat{x}_s))'$

Thus the problem is to choose g to maximize simultaneously all of the elements of the vector:

$$\Pi^p(g)k(g) = W^p(g)$$

It can be shown that if $\{P(g)\}$ is compact, which will be assumed in this study, then such an optimum exists. Thus by formulating the problem as a discounted Markov program, the question of existence works out automatically. Powerful Markov programming techniques can be applied, and that is the subject of the next section.

3. ALGORITHMS

Before we explain the details of the computer algorithm implemented, let us review briefly the constraints of the problem and their relation to different algorithms in the control literature.

3.1 LQG (Linear-Quadratic-Gaussian) Method

To apply the classical results of LQG theory, a cost function must be defined. From our previous discussion, this cost function would have the following form:

$$\sum_{t=0}^{\infty} \rho^t \,[y(t)'\, A(t)\, y(t) + y(t)'\, B(t)].$$

The above cost function does not depend on x(t); therefore if the LQG were applied directly to the problem, ignoring the inequality constraints, then the solution would trivially be

$$\max_{y(t)} y(t)'\, A(t)\, y(t) + y(t)'\, B(t);$$

the inequality constraints are the essence of the problem.

Penalty and barrier methods can be applied to handle inequality constraints in general programs, but with the LQG method we are constrained to use quadratic costs, which cannot approximate inequality constraints. Another method to handle the inequality constraints is to normalize the controls about a nominal trajectory; then the question is how to find a "nominal" trajectory.

3.2 Dynamic Programming

In a dynamic programming algorithm, the basic idea is to find an optimal value function $V^t(x)$ for each state x, for then the optimal controls satisfy

$$y(x,t) \text{ maximizes } F(t,y) + E[\rho V^{t+1}\{M(t)x(t) + N(t)y(t) + \Phi(t)\}].$$

The approach taken by ECON was to choose Monte-Carlo outcome of $\Phi(t)$, thus eliminating the expectation operator and perform a deterministic maximization. This is completely invalid, as it is equivalent to assuming that consumers and producers can make their estimates based on a forecast not yet known, one time step ahead. Alternatively the operations of maximization and expectation are commuted, which, in general, is invalid. A more valid way to remove the expectation operator would be to move it inside of V, thus replacing $\Phi(t)$ with its expected value, 0:

$$F(t,y) + \rho V^{t+1}[M(t)x(t) + N(t)y(t)].$$

The other drawback associated with the dynamic programming approach is the predetermined quadratic assumption on the optimum welfare $V^t(x)$. The choice of quadratic, rather than other types of nonlinear functions, for $V^t(x)$ is dictated by the necessity of keeping the problem computationally tractable. However, in many cases such a predetermined behavior leads to wrong result. For instance, if the incremental value function F(t,y) is independent of the state (which means that no storage cost is taken into account), then the optimal policy y^* will not depend, in general, on the value of the state (the value of the state may influence y through constraint, if constraints on y depend on the state). Similarly the cost

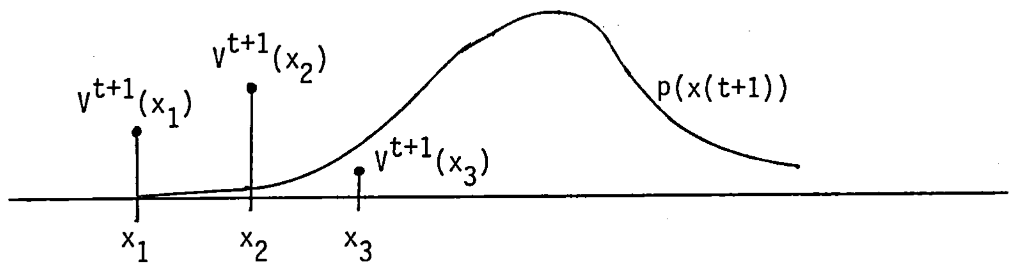
$\lim_{t\to\infty} V^t(x)$ will become independent from the state variable x. This case will be further discussed in Section 4.

3.3 Introduction to Markov Programming*

The biggest drawback to Markov programming is that the function V must be stored; hence its argument (the state space) must be discretized (cf. Section 2.5), and for a problem of the dimensionality of the wheat model, this can lead to serious storage problems for even coarse discretization (Bellman's "curse of dimensionality").

A coarse discretization leads to two fairly serious problems. The first is knowing where to choose the discrete states; a bad initial choice will give a meaningless answer. In fact, it may even give an answer that would lead one to requantize the state space in the wrong direction. Thus one must be careful in choosing the initial scheme. The second problem is dealing with the endpoints. Consider approximating the value of x(t+1) where x(t+1) has Gaussian distribution shown, and $V^t(x)$ is known at the discrete points. On the basis of the known values of $V^{t+1}(x)$, it is not possible to accurately estimate $E[V^{t+1}(x(t+1))]$. One would like to rule out such possibilities, but doing so is in effect adding new inequality constraints, In fact, just about any way one would care to define an expected value for the above problem will lead to strange effects near the endpoints of the approximation scheme. We have found, in our initial tests, that these effects can be so strong as to force the controls y(t) to be chosen to always place the distribution on an endpoint! The solution of this problem is to carefully choose the discretization scheme so that the optimal solution will not be close to the endpoints.

*For details of Markov programming, see Appendices A and B.



Despite these problems, there are many advantages to Markov programming. First, very speedy and efficient methods exist for solving them. Moreover, any distribution, rather than strictly Gaussian, may be used for the noise variable. Besides, monotonic bounds on the optimum values are available at each iteration and it is easy to prove that a solution to a Markov programming is indeed the current solution. With these advantages and drawbacks clearly in mind, let us now give a more formal description of the Markov programming method.

Recall our definition in Section 2.5 of P(g) the probability transition matrix between discrete states, dependent on the feedback control g, and k(g) the incremental value function. P(g) is cyclic of the form of $\left[\begin{array}{c|c} 0 & P_1(g) \\ \hline P_2(g) & 0 \end{array}\right]$. Define C(g) and $J^\rho(g)$ to be the unique vector solutions of the following matrix equation:

$$\rho[P(g) - I]\, C(g) + k(g) = J^\rho \qquad (1)$$

Existence and uniqueness are guaranteed for $\rho < 1$ [7], and C(g) represents the <u>discounted objective</u> value vector, within a constant of $V^t(x)$, and C(g) is the discounted analogy of Varaiya's [6] "dual variable." It also turns out that $J^\rho = W^\rho(g)$. The basic idea behind Markov programming is that the optimal feedback control g* must maximize the following expression:

$$g^* = \arg\max\, k(g) + \rho P(g) C(g) \qquad (2)$$

One simply begins with a "naive" C(g) (any initial given value will guarantee convergence), call it C_0, finds the g_1 which maximizes equation (2), and then updates C_0 to $C_1 \simeq C(g_1)$ in the following way. Rewrite equation (1) like this:

$$[\rho P(g) - I]C(g) + k(g) = J^{\rho} + (1- \ )C(g)$$

One of the fundamental results in [7] is that $J^{\rho} + (1-\rho)C(g)$ is a vector with equal entries, call it $J^1 = \alpha\mathbf{1}$ where $\mathbf{1} = (11...1)'$. Thus

$$\rho P(g)C(g) + k(g) = \alpha\mathbf{1} + C(g).$$

Since the additive constant $\alpha\mathbf{1}$ is irrelevant, set $C_1 = k(g_1) + \rho P(g_1)C_0 \simeq \alpha\mathbf{1} + C(g_1)$. Notice that the left side of equation (3) is the very expression that was maximized in equation (2), therefore simplifying the computation further.

Varaiya's "dual method" is very close to this and is easily explained with this background. Instead of setting $C_1 = k(g) + \rho P(g)C_0$, he sets $C_1 = \beta[k+\rho PC_0] + (1-\beta)C_0$. This guarantees convergence under slightly more general conditions, which are needed for our problem, involving a cyclic transition matrix. It is easy to see, however, that the convergence of $C_0, C_1, C_2, \ldots, C^*$ is slower for Varaiya's algorithm.

Nevertheless we can use Variaya's idea to actually speed up the standard Markov programming algorithm. Recall that C_1 is only an <u>approximation</u> to $C(g_1)$. In fact if a better estimate of $C(g_1)$ were available, the following maximization of g_2 would be closer to the optimal g^*. Since a simple computation of $k(g_1) + \rho P(g_1)C$ takes little time compared to the time required to find a maximization, it might be wise to compute, between optimizations, a finite sequence

$$C_1$$

$$C_2 = (k)g_1) + \rho P(g_1)C_1)\beta + (1-\beta)C_1$$

$$C_3 = (k(g_1) + \rho P(g_1)C_2)\beta + (1-\beta)C_2$$

$$\vdots$$

$$C_n = (k(g_1) + \rho P(g_1)C_{n-1})\beta + (1-\beta)C_{n-1}$$

Varaiya proved that $C_n \to C(g_1)$; thus for a little effort here we can get the most benefit out of the next maximization operation. Exactly what the tradeoff is we are not sure, that is, how large n should be. But our results show that more accurately calculating $C(g_1)$ speeds up convergence considerably, especially in the final stages of convergence.

Finally, and most importantly, the sequence $W^\rho(g_1)$, $W^\rho(g_2)$,... will converge to the optimal welfare; g_1, g_2,... "converge"* to an optimal control, and furthermore

$$W^\rho(g^*) \equiv \lim_{i\to\infty} (C_{i+1} - C_i) + (1-\rho)C_i = \lim_{i\to\infty} (C_{i+1} - \rho C_i).$$

This constitutes the theoretical basis for Markov programming.

If the optimum control g is not unique, oscillation is possible. In practice, computer algorithms usually "prefer" one optimum over another, so the sequence converges.

3.4 Structure of the Program

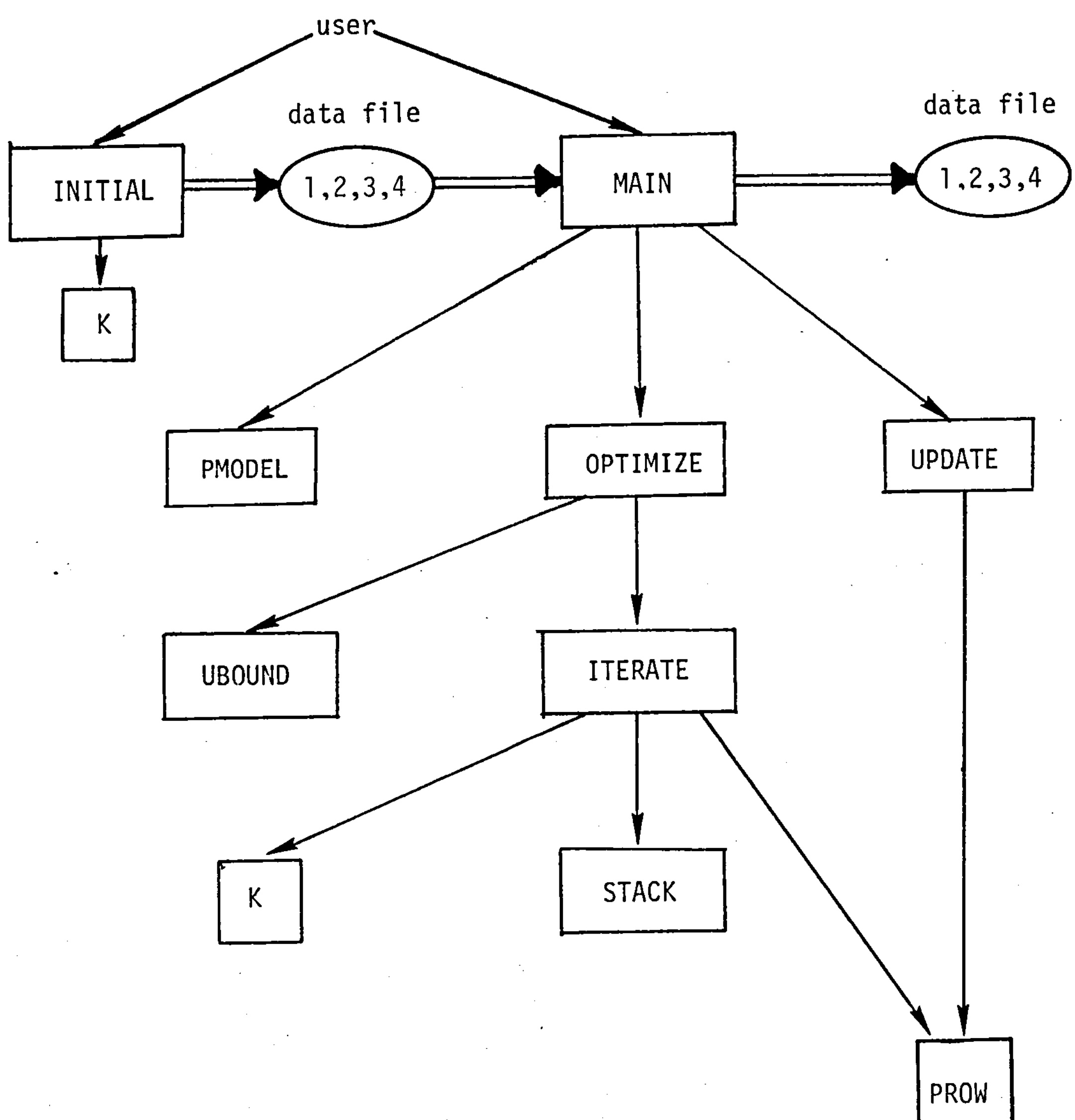
In our program the successive transition probability matrices $P(g_1)$, $P(g_2)$,... are not stored as they are prohibitively large, bur rows are computed as needed. What is stored is:

1. Details of the Economic model (number of countries, aggregated crops, planting times, etc.).
2. Details of the discretization scheme (how many discrete levels for each variable and what the levels are).
3. Statistics on means, variances, etc. of state variables at each time period.
4. Vectors c_i, the approximate value function, a dual vairable, g_i, the suboptimal control, σ_i, an approximation probability distribution (used to calculate 3) and two vectors the size of c_i and σ_i which are used as work space.
5. Algebraic workspace, and the program itself.

The program works as follows. An initialization program sets up a file with data 1, 2, 3 and 4, although only 1 and 2 affect the subsequent operation of the main program. Changes in Type 1 data require some redimensioning of matrices in the main program, but otherwise no changes in the main program are necessary. The main program takes the file with data 1, 2, 3 and 4, iterates, and when it has converged, writes the new values at 1, 2 3 and 4 onto another file. A third program may be written to examine the output file, which contains the optimal solution, more closely.

In the iteration Type 5 data changes most rapidly, followed by 4, 3 and 2. Type 1 does not change. After the approximate controls and probability distribution σ_i have converged (data Type 4), then 3 is updated. If the statistics are at too great variance from the discretization scheme (data Type 2), then data 2 is updated. This is the basic sequence of events.

The names of the various routines are as follows. INITIAL is the initializing program; it calls only one subroutine, K, which computes the incremental value function $F^t(\hat{x}; g(t,\hat{x}))$ for any discrete state $\hat{x}_j$, time t.



MAIN is the main program and calls three subroutines PMODEL, OPTIMIZE, UPDATE. The data file is read into a blank common area shared by all subroutines, except that scratch space is in a common area called SCR. PMODEL prints information about the economic model. OPTIMIZE computes

$$g_{i+1} = \max_{g} \arg\,[k(g_i) + \rho P(g_i)c_i]$$

by first computing bounds on the admissible control (UBOUND) and then searching through the controls (ITERATE) with first a coarse approximate search and then a detailed accurate search. The highest values are stacked (STACK) for later inspection for multiple peaks. Then UPDATE is called to update c_i to $c(g_{i+1}) = c_{i+1}$ by successive approximations, as we discussed above. MAIN then iterates OPTIMIZE and UPDATE until the solution converges and this is fairly fast. Then as we said, if the statistics are off from the discretization scheme by a significant amount, MAIN will call a program ORIENT to redefine the discrete states appropriately. Then MAIN writes out the answer to a disk file (see illustration).

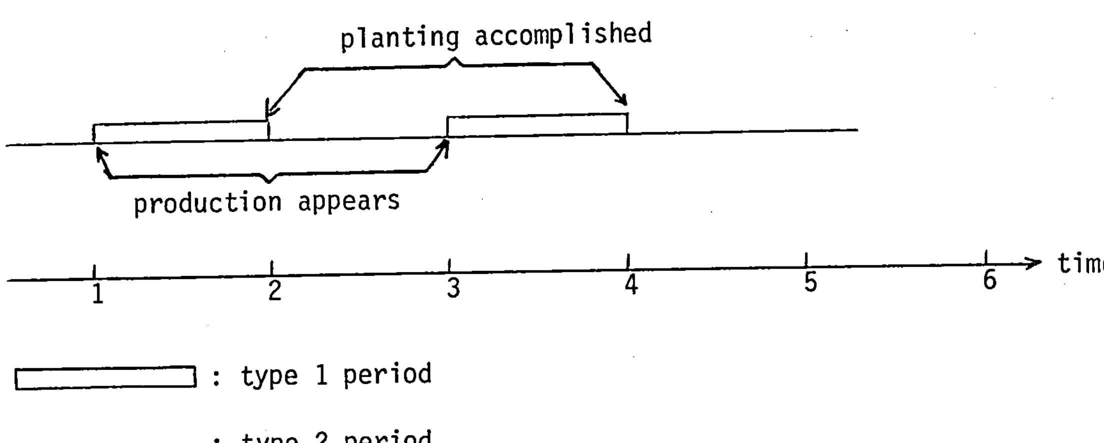
The subroutine PROW acts as a "virtual matrix" P(g) and computes, given a g and t, a specified row of the matrix P.

4. A SIMPLIFIED MODEL: ONE COUNTRY - TWO PERIODS

In order to get some insight into the Markov programming approach and in particular into the computational problems which are involved, we have decided to consider the case of one country - two period model. The model that we are going to study is the one considered by ECON in "Economic Benefits of Improved Information on Worldwide Crop Production and Distribution with Application to Wheat, Corn and Soybeans" (contract No. NASW-2558 - p. 27-58) [8]. The data which will be used are essentially the same as in the above ECON model.

4.1 The Model [8]

Based on [8], the model for one country - two periods is the following: The year is divided into two types of period, type 1 and 2, as depicted in the following diagram.



The state variables $x_1(t)$ and $x_2(t)$ are defined as:

$x_1(t) \triangleq$ Mean value of total stock at time t (inventory at time t)

$x_2(t) \triangleq$ Mean value of total quantity of growing crop (planted, but unharvested)

Now at times $t = 1,3,5,\ldots$, that is, at the beginning of type 1 periods, we have:

$$\begin{cases} x_1(t+1) = x_1(t) - y_1(t) + v_1(t) \\ x_2(t+1) = y_2(t) + v_2(t) \end{cases}$$

where $y_1(t)$ is the consumption

$y_2(t)$ is the planted crop

The second equation states simply that the unharvested period at period 2 equates to the planting done at period 1.

$v_1(t)$ and $v_2(t)$ are stochastic terms translating uncertainties on inventories and production yields. They are assumed to be zero mean Gaussian.

At time $t = 2,4,6,\ldots$, that is, at the beginning of type 2 periods, the state equation is:

$$\begin{cases} x_1(t+1) = x_1(t) + x_2(t) - y_1(t) + v_1(t) \\ x_2(t+1) = v_2(t) \end{cases}$$

The second equation states simply that at the beginning of type 1 period (time t+1), there are no crops in the ground beside a noise term $v_2(t)$.

It is possible, and desirable for computational reasons, to reduce the order of the system by retaining the mean value of the inventory $x_1(t)$, as the sole state variable. Then:

$$t = 1,3,5,\ldots$$

$$x(t+1) = x(t) - y_1(t) + \phi(t)$$

At time $t = 2,4,6,\ldots$:

$$x(t+1) = x(t) - y_1(t) + y_2(t) + \phi(t)$$

The choice of decision variable $y_1(t)$ and $y_2(t)$ is constrained to:

$$\begin{cases} 0 \le y_1(t) \le x(t) & \text{(consumption} \le \text{available stock)} \\ 0 \le y_1(t) & \text{(positive production)} \end{cases}$$

4.2 Quality of Information

In the context of the above model, the quality of information is directly related to the statistic of $\phi(t)$ (or equivalently $v_1(t)$, $v_2(t)$). $\phi(t)$ is assumed to be zero mean Gaussian; however, for the Markov programming approach the Gaussian assumption can be relaxed. In fact, any other distribution for $\phi(t)$ can be considered.

There are various information gathering schemes both on the level of inventories and on the future production. The issue is to evaluate the gain of the community (in terms of its welfare function) vis-a-vis an improvement in the information gathering scheme. In case of the Gaussian assumption, an improvement of information translates into a decrease in the noise variance.

In the case of the present example, the Gaussian random variable $\phi(t)$ represents the inventory uncertainty at time $t = 1,3,5$ (period of type 1) ($\phi(t) = v_1(t)$) with:

$$E[\phi^2(t)] = \sigma_1^2 \qquad (t = 1,3,5,\ldots)$$

But at time $t = 2,4,6,\ldots$, $\phi(t)$ represents the sum of uncertainties on the inventory level and on the production. These two latter uncertainties being independent:

$$\phi(t) = v_1(t) + v_2(t) \qquad t = 2,4,\ldots$$

$$E[\phi^2(t)] = \sigma_1^2 + \sigma_2^2 \qquad t = 2,4,\ldots$$

Hence the variance at period of type 2 is always greater than the variance at period of type 1.

4.3 Incremental Value Function

The incremental value function is given by:

$$\begin{cases} F(y_1,y_2) = a_1{y_1}^2 + b_1y_1 & \text{at } t = 1,3,5,\ldots \\ F(y_1,y_2) = a_1{y_1}^2 + b_1y_1 + a_2{y_2}^2 + b_2y_2 & t = 2,4,6,\ldots \end{cases}$$

where the expression $a_1{y_1}^2 + b_1y_1$ is referred to as consumer welfare and $\{-(a_2{y_2}^2 + b_2y_2)\}$ is the production cost.

The purpose of the optimal stationary control is to maximize:

$$W = \lim_{T\to\infty} \frac{1}{T+1} \sum_{i=0}^{T} E\{\rho^t F(y_1(x),y_2(x))\}$$

where ρ is a given discount rate.

Note that the incremental value function is independent from the state $x(t)$ of the system, and hence from the noise term $\phi(t)$, since this latter enters the system through the state $x(t)$. If there were no state dependent constraints on the input $y(t)$, one would clearly deduce that the optimum stationary control is independent from both the state and the noise characteristics. However, since the constraints on the control $y(t)$ are state dependent $(y(t) \leq x(t))$, theoretically the state of the system _may_ affect the optimum control $y(t)$ and the optimum welfare through the constraints. But since we are considering _stationary_ optimum control, for the state to affect the optimum welfare function, it is necessary for the stationary optimum control $y(t)$ to hit the constraints; that is, for stationary consumption to equate the available stock in at least one of the two periods. But in general this is not the case, or at least not a desirable case. If the consumption equates the available stock, one has to alter the incremental welfare function so that the corresponding optimum consumption/production policy would not deplete the available stock at any period. And in this latter case, the optimum control and welfare will be independent from the state and the noise.

Notwithstanding the above discussion, it is apparent that a state independent incremental value function, such as the one used by ECON in the one country - two period example [8], is not appropriate to measure the benefit of improved information on the noise statistic, because of its general lack of sensitivity. To overcome this problem one has to make the incremental value function state dependent. A natural way of doing so is to add a term representing the cost of storage, and hence depending on the state $x(t)$. There are several possible choices for such a cost function. We have chosen to represent the term referring to the storage by:

$$+ a_3(x - .8M_i)(x - 1.2M_i), \qquad i = 1,2$$

where M_i is the desired mean at periods of type 1 and of type 2 and a_3 is a positive constant. The intuitive meaning of the above term is that the community would favor an inventory within 20% of a mean M_i for which the storage capacities are prepared. Any inventory values outside the desired 20% range is disfavored. To be more precise, the incremental value function F is defined as:

$$t = 1,3,5,\ldots$$

$$F[y_1(t),\ x(t)] = a_1y_1^2(t) + b_1y_1(y) + E\{a_3[x(t+1) - .8M_2][x(t+1) - 1.2M_2]\}$$

Note that the last term refers to the estimate of the storage cost at time t+1, that is at period of type 2. M_2 is the desired mean at period of type 2. (Note that M_2 may be state dependent, that is, M_2 at time t+1 may depend on the value of the state $x(t)$.)

Carrying on the expectation operation, we deduce:

$$\begin{aligned} F(y_1(t),\ x(t)) = a_1y_1^2(t) + b_1y_1(t) &+ a_3[x(t) - y_1(t) - .8M_2] \\ &\cdot[x(t) - y_1(t) - 1.2M_2] + a_3E[\phi^2(t)] \\ &\qquad t = 1,3,5,\ldots \end{aligned}$$

Similarly for type 2 period we have:

$$F(y_1(t),\ y_2(t),\ x(t)) = a_1y_1^2(t) + b_1y_1(t) + a_2y_2^2(t) + b_2y_2(t)$$
$$+ a_3[x(t) + y_2(t)][x(t) + y_2(t) - y_1(t)] + a_3E[\phi^2(t)]$$
$$t = 2,4,6,\ldots$$

Before ending this section let us note that the numerical evaluation of the optimum policy y^* and the corresponding welfare value under the assumption that F has no state dependency, i.e., $a_3 = 0$, confirms the previous theoretical claim, that is, both the optimum policy and the optimum welfare remain insensitive to changes of noise variance. But, surprisingly, the numerical results of the ECON treatment of this example [8] infer that both the optimum policy y^* and the optimum welfare change when the noise variances vary. We can explain this apparent paradox in the following way. In the dynamic programming used by ECON, at each iteration one maximizes:

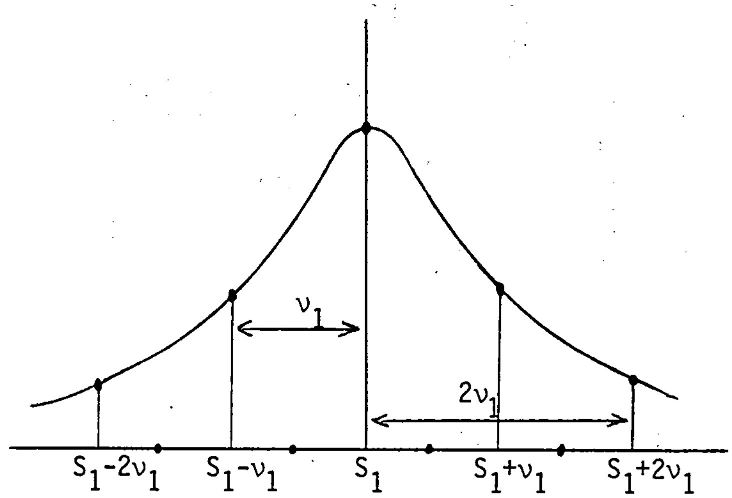
$$F(y_1(t),\ y_2(t)) + \rho E\{V^{t+1}(x(t+1))\}$$

where $V^{t+1}(x(t+1))$ is the optimum value of the welfare function from time (t+1) to infinity. F is independent from the state x(t), but for computational tractability $E\{V^{t+1}[x(t+1)]\}$ is made to be a quadratic functional in x(t) (see Section 3). This computational necessity changes the nature of the optimization problem and forces the optimal control y^* and the optimal welfare to depend on the state x and hence on the process noise variance. Hence the numerical results of ECON concerning the present example do not correspond to any benefit, in terms of the specific welfare function involved. In fact, there is simply no benefit in information improvement if there is no dependency between the incremental value function and the state.

4.4 Discretization - Probability Matrix P(y)

We have used 5 to 13 discrete values to represent the state at period of type 1 and type 2. For the sake of representation, assume that we have only 5 discrete values for each type of period.

Let us call S_1 the mean value of the total inventory at period 1 and ν_1 its variance. S_1 and ν_1 are provided by past statistics (either by Kalman filtering or other time series analysis).



There are various criteria for discretizing the state value around S_1. One may use "equal area" criterion, that is, the area under the p.d curve between two consecutive discrete values is the same. Or we can use a simpler "midpoint" scheme which consists of taking the sequence of

$$S_1-2\nu_1,\ S_1-\frac{3\nu_1}{2},\ S_1-\nu_1,\ S_1-\frac{\nu_1}{2},\ S_1,\ldots,\ S_1+2\nu_1$$

as discrete value. For the present example this last discretization scheme has been retained. Note that there is no restriction as to the type of discretization to be used. Similarly for the second type of period, the state is discretized around S_2 with variance ν_2.

Let us call S_{1i} the discrete value of the state at period of type 1 and S_{2i} the discrete value of period of type 2. Then the probability distribution matrix, corresponding to the case of $i = 5$, is cyclic and given by:

$$P(y_1, y_2) = \left[\begin{array}{c|c} \bigcirc & \begin{matrix} P_{16} & P_{17} & \cdots & P_{110} \\ \vdots & & & \\ P_{56} & & \cdots & P_{510} \end{matrix} \\ \hline \begin{matrix} P_{61} & & P_{65} \\ & & \\ P_{101} & & P_{105} \end{matrix} & \bigcirc \end{array}\right]$$

with

$$\begin{cases} P_{ij} = p[x(t) = S_{2j} | x(t-1) = S_{1i}, y_1, y_2]; & i \le 7,\ j \ge 8 \\ P_{ij} = p[x(t) = S_{1j} | x(t-1) = S_{2i}, y_1, y_2]; & i \ge 8,\ j \le 7 \end{cases}$$

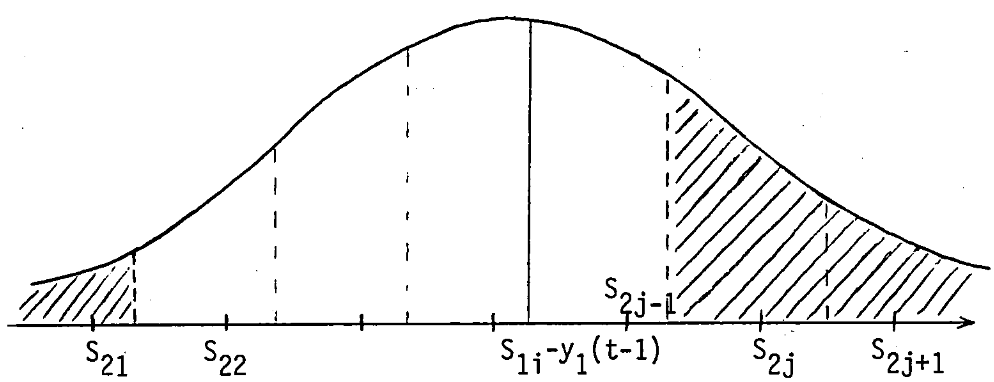
Note that for $i \le 7$, corresponding to the first period, the production y_2 equals zero, hence:

$$P_{ij} = p[x(t) - S_{2j} | x(t-1) = S_{1i}, y_1] \quad \text{for } i \le 7$$

To evaluate P_{ij}, assume that at time t-1, the state is in S_{1i}, the consumption being $y_1(t-1)$, the probability distribution for the state x(t) is given by:

$$f\{x(t)|x(t-1) - S_{1i},\ y_1(t-1)\} = \frac{1}{\sqrt{2\pi' E(\phi^2(t-1)}} \exp - \left\{\frac{[x(t)-(S_{1i}-y_1)]^2}{2E[\phi^2(t-1)]}\right\}$$

where $E[\phi(t-1)]$ = variance of the first period. This probability distribution is shown in Fig.



The probability $p[x(t) = S_{2j}|x(t-1) = S_{1i}, y_1(t-1)]$ is then computed as the area between $\frac{S_{2j-1}+S_{2j}}{2}$ and $\frac{S_{2j}+S_{2j+1}}{2}$ under the probability distribution $f(x(t)|x(t-1) = S_{1i}, y_1(t-1))$. That is,

$$P_{ij}(y_1(t-1) = \frac{1}{\sqrt{2\pi E(\phi^2(t-1))}} \int_{\frac{S_{2j}-S_{2j-1}}{2}}^{\frac{S_{2j}+S_{2j+1}}{2}} \exp - \frac{(\tau-(S_{1j}-y_1(t-1)))^2}{2(E\phi^2(t-1))}\, d\tau$$

$$i \leq 7$$

The probability corresponding to the endpoints, say to S_{21} and S_{25}, are computed as the area between $-\infty$ and $\frac{S_{21}+S_{22}}{2}$ for P_{i1} and between $\frac{S_{25}+S_{24}}{2}$ and $+\infty$ for P_{i5}.

Similarly the probability P_{ij} for $i \geq 8$ is computed; the only difference is that the mean is replaced by $(S_{2i}-y_1(t-1)+y_2(t-1))$ and the variance corresponds to the second type period noise.

At this point the essential elements of the Markov programming algorithm of Section 3, that is, the value function F and the transition matrix $P(y_1,y_2)$ have been determined.

4.5 Data

Incremental value function:

$a_1 = -2.$

$b_1 = 840.$

$a_2 = -.4$

$b_2 = 140.$

$a_3 = .5$

$\rho = .971$

Mean and variance of inventories at period 1 and 2, used for discretization:

$M_1 = 391.3$ (millions of metric tons) $\quad \nu_1 = 38.8$

$M_2 = 217.1 \quad \nu_2 = 43.$

4.6 Numerical Results

Tables 1a - 1c contain optimal production/consumption policy under three different information schemes for the case of 5 discrete values per period.

Period 1, $E[\phi^2(t)] = 784$			Period 2, $E[\phi^2(t)] = 1764$		
State	Consumption	Production	State	Consumption	Production
313.7	152.7	0	131.	131.	372.1
352.5	163.2	0	174.1	174.1	372.1
391.3	175.7	0	217.1	176.	345.2
430.1	186.7	0	260.2	181.2	319.
468.9	198.5	0	303.2	186.6	292.1

Table 1a

Period 1, $E[\phi^2(t)] = 196$			Period 2, $E[\phi^2(t)] = 1764$		
State	Consumption	Production	State	Consumption	Production
313.7	148.2	0	131.	131.	372.1
352.5	160.	0	174.1	174.1	372.1
391.3	179.9	0	217.1	176.1	345.2
430.1	183.8	0	260.2	181.2	319.
468.9	197.4	0	303.2	186.7	292.1

Table 1b

Period 1, $E[\phi^2(t)] = 441$			Period 2, $E[\phi^2(t)] = 784$		
State	Consumption	Production	State	Consumption	Production
313.7	148.4	0	131.	131.	378.1
352.5	163.2	0	174.1	170.	375.2
391.3	176.1	0	217.1	175.4	348.
430.1	186.	0	260.2	180.9	320.6
468.9	197.5	0	303.2	186.4	293.2

Table 1c

The transition probability matrices corresponding to 1a through 1c are:

$$\left.\begin{array}{lllll}
0.3805E+00 & 0.5108E+00 & 0.1059E+00 & 0.2788E-02 & 0.8225E-05 \\
0.9464E-01 & 0.4943E+00 & 0.3721E+00 & 0.3853E-01 & 0.4838E-03 \\
0.1211E-01 & 0.2248E+00 & 0.5573E+00 & 0.1966E+00 & 0.9167E-02 \\
0.5877E-03 & 0.4330E-01 & 0.3887E+00 & 0.4817E+00 & 0.8570E-01 \\
0.1278E-04 & 0.3755E-02 & 0.1245E+00 & 0.5282E+00 & 0.3436E+00
\end{array}\right\} p_{ij},\ \begin{array}{r} i\le 7 \\ j\ge 8 \end{array}$$

$$\left.\begin{array}{lllll}
0.1766E+00 & 0.3215E+00 & 0.3229E+00 & 0.1464E+00 & 0.3267E-01 \\
0.1766E+00 & 0.3215E+00 & 0.3229E+00 & 0.1464E+00 & 0.3267E-01 \\
0.1024E+00 & 0.2630E+00 & 0.3536E+00 & 0.2147E+00 & 0.6636E-01 \\
0.6135E-01 & 0.2064E+00 & 0.3518E+00 & 0.2708E+00 & 0.1097E+00 \\
0.3588E-01 & 0.1544E+00 & 0.3284E+00 & 0.3154E+00 & 0.1658E+00
\end{array}\right\} p_{ij},\ \begin{array}{r} i\ge 8 \\ j\le 7 \end{array}$$

1a

$$\left.\begin{array}{lllll}
0.1769E+00 & 0.8072E+00 & 0.1586E-01 & 0.8811E-07 & 0.5551E-16 \\
0.2134E-02 & 0.5839E+00 & 0.4134E+00 & 0.4966E-03 & 0.9612E-10 \\
0.1305E-04 & 0.1292E+00 & 0.8449E+00 & 0.2589E-01 & 0.2585E-06 \\
0.1058E-10 & 0.1457E-03 & 0.2918E+00 & 0.7023E+00 & 0.5748E-02 \\
0.9613E-17 & 0.2924E-07 & 0.9428E-02 & 0.7568E+00 & 0.2337E+00
\end{array}\right\} p_{ij},\ \begin{array}{r} i\le 7 \\ j\ge 8 \end{array}$$

$$\left.\begin{array}{lllll}
0.1815E+00 & 0.3241E+00 & 0.3202E+00 & 0.1428E+00 & 0.3133E-01 \\
0.1815E+00 & 0.3241E+00 & 0.3202E+00 & 0.1428E+00 & 0.3133E-01 \\
0.1049E+00 & 0.2656E+00 & 0.3530E+00 & 0.2119E+00 & 0.6426E-01 \\
0.6290E-01 & 0.2090E+00 & 0.3524E+00 & 0.2683E+00 & 0.1074E+00 \\
0.3695E-01 & 0.1570E+00 & 0.3301E+00 & 0.3134E+00 & 0.1625E+00
\end{array}\right\} p_{ij},\ \begin{array}{r} i\ge 8 \\ j\le 7 \end{array}$$

1b

0.2715E+00	0.6538E+00	0.7443E-01	0.2398E-03	0.1497E-07	p_{ij}, $i \leq 7$, $j \geq 8$
0.3993E-01	0.5774E+00	0.3723E+00	0.9420E-02	0.5450E-05	
0.1420E-02	0.1736E+00	0.6927E+00	0.1316E+00	0.7744E-03	
0.6470E-05	0.1041E-01	0.3866E+00	0.5662E+00	0.3682E-01	
0.7435E-08	0.1513E-03	0.5887E-01	0.6278E+00	0.3131E+00	
0.5385E-01	0.3579E+00	0.4657E+00	0.1171E+00	0.5410E-02	p_{ij}, $i \geq 8$, $j \leq 7$
0.4951E-01	0.3464E+00	0.4731E+00	0.1249E+00	0.6079E-02	
0.2171E-01	0.2413E+00	0.5108E+00	0.2098E+00	0.1628E-01	
0.8536E-02	0.1503E+00	0.4916E+00	0.3114E+00	0.3820E-01	
0.2982E-02	0.8329E-01	0.4224E+00	0.4117E+00	0.7966E-01	

1c

Finally, if we take the case 1a as the base, the gain of welfare due to the improvement of information (decrease in noise variances) corresponding to cases 1b and 1c are:

1b 65 ($ million)

1c 104 ($ million)

Note that the improvement in case 1b pertains only to the variance of the inventory (decrease of noise variance in the first period from 784 to 196). In case 1c both variances of period 1 and period 2 decrease.

The above figures are obtained under a rather coarse discretization (5 discrete values per period). For more accurate values one must consider a finer grid and also rediscretize the state space around the optimal value obtained with the coarse discretization. In this example, taking 9 discrete values per period leads to the following optimal consumption/production scheme:

Period 1, $E[\phi^2(t)] = 784$			Period 2, $E[\phi^2(t)] = 1764$		
State	Consumption	Production	State	Consumption	Production
313.7	154.5	0	131.	131.	370.3
333.1	158.5	0	152.5	152.5	370.3
352.5	164.2	0	174.1	174.1	370.3
371.9	170.2	0	195.6	173.7	356.6
391.3	176.	0	217.1	176.2	344.1
410.7	181.5	0	238.6	178.8	331.2
430.1	187.	0	260.2	181.4	318.1
449.5	192.7	0	281.7	184.	304.8
468.9	198.9	0	303.2	186.7	291.4

1a

Period 1, $E[\phi^2(t)] = 441$			Period 2, $E[\phi^2(t)] = 784$		
State	Consumption	Production	State	Consumption	Production
313.7	150.7	0	131.	131.	377.2
333.1	157.2	0	152.5	152.5	377.2
352.5	164.	0	174.1	170.1	374.4
371.9	170.2	0	195.6	172.8	361.
391.3	175.8	0	217.1	175.5	347.5
410.7	181.1	0	238.6	178.2	333.9
430.1	186.4	0	260.2	180.9	320.3
449.5	191.8	0	281.7	183.7	306.6
468.9	197.8	0	303.2	186.4	293.

1c

In this latter case the gain of the information improvement amounts to \$91 million instead of the \$105 million computed previously. Since we are using a finer grid (9 discrete values instead of 5), the \$91 million figure is more accurate than the \$105 million one. Moreover, taking a grid of 11 and 13 discrete values for each period results in a gain corresponding to an improvement of information from 1a to 1c equal in both cases to \$91 million. Therefore, the grid of 9 discrete values is a sufficient approximation. In the following tables the optimal consumption/production policies corresponding to a grid of 11 discrete values are shown.

Period 1, $E[\phi^2(t)] = 784$			Period 2, $E[\phi^2(t)] = 1764$		
State	Consumption	Production	State	Consumption	Production
313.7	154.7	0	131.	131.	370.1
329.2	157.6	0	148.2	148.2	370.1
344.7	162.	0	165.4	165.4	370.1
360.3	166.7	0	182.7	172.3	363.7
375.8	171.5	0	199.9	174.2	354.0
391.3	176.	0	217.1	176.2	343.9
406.8	180.4	0	234.3	178.3	333.7
422.3	184.8	0	251.5	180.4	323.2
437.9	189.2	0	268.8	182.5	312.7
453.4	193.9	0	206.0	184.6	302.0
468.9	199.0	0	303.2	186.7	291.3

1a

Period 1, $E[\phi^2(t)] = 441$			Period 2, $E[\phi^2(t)] = 784$		
State	Consumption	Production	State	Consumption	Production
313.7	151.	0	131.	131.	377.1
329.2	156.	0	148.2	148.2	377.1
344.7	161.5	0	165.4	165.4	377.1
360.3	166.7	0	182.7	171.2	369.
375.8	171.4	0	199.9	173.4	358.2
391.3	175.8	0	217.2	175.5	347.4
406.8	180.1	0	234.3	177.7	336.6
422.3	184.3	0	251.5	179.9	325.7
437.9	188.6	0	268.8	182.0	314.8
453.4	193.	0	286.0	184.2	303.9
468.9	197.8	0	303.2	186.4	292.9

1c

4.7 Remarks

Some experience has been gained in solving the example of one country - two period models. We feel that some of the practical problems encountered in this simple case will be present in the more complex setting of multi-country - multi-period problems. These are the following:

a) In applying the Markov programming algorithm of Section 3, at each step one has to find the optimum y^* such that:

$$y^* = \arg\max_{y} \{F(y,x) + \rho P(y)C\}$$

It is important that the subroutine finding the maximand at each period be as accurate as possible (Extended Precision). This will speed the convergence and allow more flexibility for the choice of initial "dual vector" C.

b) Theoretically, any initial choice C_0 of the vector C insures the convergence. But from a practical point of view a bad initial choice will cause the convergence to be very slow. Notwithstanding the order of the vector and matrices involved, which increase rapidly with finer discretization grid, it is important to keep the number of iterations small. Hence there are advantages to choosing the initial vector C_0 close to the optimum C.

c) As noted in Section 3, to speed up the convergence we calculate, as intermediate values, a sequence of n vector S, C^i as:

$$C^{i+1} = \beta[F(y,x) + \rho P(y)C^i] + (1-\beta)C^i$$

The right choice of the parameter β and the number of iterations can speed up the convergence significantly. Experience shows that a choice of β in the range [.5,.9] and an n between 5 and 20 speeds the convergence sufficiently.

d) The convergence is relatively fast. The number of iterations varies between 10 and 40; it rarely increases above 50. However, as noted before, the initial choice of C_0 together with the parameter β and the intermediate number of iterations n can strongly influence the speed of convergence. With appropriate choice of C_0, β and n the convergence is attained with less than 10 iterations.

5. CONCLUSION

The problem of "Wheat Forecast Economic Effect" has been formulated and solved within the framework of stochastic control and Markov programming. A general approach has been developed for the multi-country - multi-period model and the simple case of one country - two period model has been solved in detail.

It has been shown that:

(i) The states of the ECON model are state estimates rather than true states.

(ii) The number of states in the ECON model may be effectively reduced by one-half.

(iii) The Markov programming approach avoids simulation and is applicable to nonquadratic welfare functions and non-Gaussian errors.

(iv) Upper and lower bounds are obtained in the Markov programming approach so that if iterations are stopped prior to convergence, an estimate of the nearness to the optimal solution is known.

(v) In general, there is no value of information in the infinite horizon stationary case if the incremental value function does not depend upon a state. The ECON algorithm produces a value of information in such cases by forcing the dynamic programming value function to be quadratically dependent on the state and by considering finite horizons in simulations.

(vi) The main advantages of the dual variable Markov programming applied to "Wheat Forecast Economic Effects" can be summarized as follows.

First, speedy and efficient algorithms are available for solving Markov programming problems. Second, no solution to a large set of simultaneous equations is required. Third, unlike the dynamic programming approach, there is no need for an ad hoc assumption on the functional dependency of the welfare with respect to the state x.

The main drawbacks of the above Markov programming is the memory requirement, since the value of the welfare function at each iteration must be stored. However, it was found in the one country - two period model that a grid size of 9 is adequate and even with a grid size of 5, the results are fairly close. Therefore, it appears feasible to solve multi-country - multi-period problems with this approach.

Based on the results of this report, it is recommended that the Markov programming approach be applied to the complete ECON model and to other value of information problems faced by NASA researchers.

REFERENCES

1. ECON, Inc. (1977), "ECON's Optimal Decision Model of Wheat Production and Distribution--Documentation," NASA Contract NASW-3047.

2. Witsenhausen, H. S. (1971), "Separation of Estimation and Control for Discrete Time Systems," Proc. IEEE, Vol. 59, pp. 1557-1566.

3. Howard, R. A. (1960), Dynamic Programming and Markov Processes, New York: Wiley.

4. Schweitzer, P. J. (1965), "Perturbation Theory and Markov Decision Processes," PhD dissertation, MIT Operations Research Center Report 1S.

5. Odoni, A. R. (1969), "On Finding the Maximal Gain for Markov Decision Processes," Operations Research, Vol. 17, pp. 857-860.

6.* Varaiya, P. 1978), "Optimal and Suboptimal Chains," IEEE Trans. Auto. Cont., Vol. AC-23.

7.* Jones, S. N. (1979), "A Differential Theory of Markov Control," unpublished working paper.

8. ECON, Inc. (1977), "An Optimal Decision Model of Production and Distribution with Application to Wheat, Corn and Soybeans," NASA Contract NASW-2588.

*These papers appear as appendices at the end of this report.

APPENDIX A

DOCUMENTATION FOR THE CROP INFORMATION VALUE PROGRAM

DOCUMENTATION FOR THE CROP INFORMATION VALUE PROGRAM

0. Introduction

This document provides the necessary background to understand, use or modify the Crop Information Value Program (CIVP) developed at S^2I. The following topics will be covered:

1. Mathematical Preliminaries
2. Notation for Input-Output
3. Using the Programs
4. Detailed Study of the Program

The object of 1 - 3 is to provide the user with quickest access to the purpose and use of the program, from a basic mathematical sketch of the problem, to notation for the basic parameters necessary to run the program, and provide detailed instructions for actually running CIVP. Examples of runs are provided in the computer output (pp.

Section 4, on the other hand, is aimed at the user who needs a more detailed understanding of the program subroutines, internal variables, approximation and search methods, for the purpose of program verification or modification.

1. Mathematical Preliminaries

In this section the details of the ECON model and formulation of the finite-state model will be assumed familiar from our Progress Report. We will review briefly the results which are essential to an overall understanding of CIVP.

1.1 Discrete Markov Chains

In CIVP, the economy is approximated by a finite-state Markov chain with state space $X = \{1,2,...,ns\}$, ns being the number of states. Recall that a state $x_i \in X$ is a multidimensional estimate of the states of crops, grain in storage and transit. Due to the independence of the innovation sequence of the Kalman Filter from the state estimates x_i themselves (see

Progress Report), the statistics, or probability distribution of $x(t+1)$ depends only on the present state estimate $x(t) = x_j$ and the behavior of consumers, producers, and exporters in the time interval $(t,t+1]$. The behavior results in an overall control $v(t)$. Symbolically,

$$\text{Prob}\,(x(t+1) = x_j | x(t) = x_i) = P_{ij}(v(t))$$

The matrix $\{P_{ij}\}$ is called the transition probability matrix at t. If we assume $v(t) = v_i$ when $x(t) = x_i$, then the matrix P depends only on the vector $(v_1,\ldots,v_{ns}) = v$; we write $P(v)$.

A discrete (controllable) Markov Chain is then defined by the set V of possible controls, X, the state space, and the set $\{P(v)\}, v \in V$ of possible transition probability matrices.

1.2 Welfare in a Discounted Economy

If $k(t, v_i, x_i)$ is a measure of the overall welfare of the economy between t and t+1 when in state x_i and exercising control v_i, then it is reasonable to assume that the economy acts so as to maximize long-term "discounted" welfare defined by:

$$W_i^\rho = \sum_{t=0}^{\infty} \sum_{x_j \in X} \rho^t \,\text{Prob}\{x(t) = x_j | x(0) = x_i\} \cdot k(t, v_j, x_j)$$

where ρ is the per-period discount factor and i is the starting state. In CIVP we define

$$J^\rho = (1-\rho)W^\rho$$

so that as $\rho \to 1$, J^ρ will reach a definite limit (called J^1). J^1 is a constant vector, i.e., all elements are equal, and represent the average welfare per step. When the economy maximizes J^ρ, i.e., the welfare discounted into the future, instead of J^1, then the overall average welfare per step (J^1) will not be as high as possible. This is what is lost by some short-sightedness on the economy's part.

1.3 Markov Programming

The purpose of the program is to find a $v^* \in V$ so that $J^\rho(v') \le J^\rho(v^*)$ for all $v' \in V$. To do this it simultaneously finds c^*, a relative state value vector, and v^*, by the method of successive approximations. c^* has the properties

1. $v^* = \operatorname*{argmax}_{v \in V} \rho P(v)c^* + k(v)$

2. $c^* = \max_{v \in V} \rho P(v)c^* + k(v) + J^{1*}$.

(It is implicit in this notation that some $v^* \in V$ will maximize all elements of the vectors simultaneously. This is in fact the case.) By using the equations, we can generate successive approximations $\gamma_0, \gamma_1, \ldots$ to c^*, and $v_1, v_2, \ldots$ to v^*. The need to know J^{1*} is eliminated by defining

$$[c] = c - \frac{\mathbb{1}'c\mathbb{1}}{ns}$$

where $\mathbb{1}' = (1\ 1\ 1\ ..\ 1)$. Then since $J^1 = \alpha\mathbb{1}$ for some α, $[J^{1*}] = 0$ and we can write

2'. $[c^*] = [\max_{v \in V} \rho P(v)[c^*] + k(v)]$

also, since $P\mathbb{1} = \mathbb{1}$,

1'. $v^* = \operatorname*{argmax}_{v \in V} \rho P(v)[c^*] + k(v)$

The method of Markov Programming is then as follows: set v_0, γ_0 to arbitrary values. Then simply take

1". $v_{n+1} = \operatorname*{argmax}_{v \in V} \rho P(v)[\gamma_n] + k(v)$

2". $\gamma_{n+1} = [\max_{v \in V} \rho P(v)[\gamma_n] + k(v)]$

It is guaranteed that $\gamma_n \to c^*$ and $v_n \to v^*$ for most conditions. Unfortunately, convergence is not guaranteed for the case of cyclic transition matrices. However, this problem can easily be fixed by Varaiya's "Dual Method" as explained in the next section.

1.4 Incorporation of Varaiya's "Dual Method"

The essential difference between Varaiya's "Dual Method" and the Markov Programming method as described above is that Varaiya takes

$$\frac{d\gamma_t}{dt} = [\max_{v \in V} \rho P(v)[\gamma_t] + k(v) - [\gamma_t]]$$

instead of

$$\gamma_{n+1} = [\max_{v \in V} \rho P(v)[\gamma_n] + k(v)]$$

In practical terms this amounts to taking

$$\gamma_{n+1} = [\gamma_n] + \varepsilon[\max_{v \in V} \rho P(v)[\gamma_n] + k(v) - [\gamma_n]]$$

Letting v_{n+1} be as defined by 1", and assuming $[\gamma_n] = \gamma_n$,

$$2'''. \quad \gamma_{n+1} = \gamma_n + \varepsilon\rho P(v_{n+1})\gamma_n + \varepsilon k(v_{n+1}) - \varepsilon\gamma_n$$

$$= \varepsilon[\rho P(v_{n+1})\gamma_n + k(v_{n+1})] + (1-\varepsilon)\gamma_n$$

For small enough ε, Varaiya's theorem guarantees that $\gamma_n \to c^*$ for cyclical transition matrices (as well as others). Notice that if $\varepsilon = 1$, 2''' is equivalent to 2"; Varaiya's method is the same as Markov Programming. Our tests on small models indicated that convergence was fastest when $\varepsilon = \frac{1}{2}$, and this value of ε was used in CIVP.

The actual algorithm implemented is now just a synthesis of the Varaiya Algorithm, and Markov Programming. The reason for not using Varaiya's method directly is that the time required to compute 1 is much, much greater than the time to compute 2. Notice that if V_0 were to consist of only a single possible control, let us say $\{v_0\} = V_0$, then Varaiya's equation would yield

1. $v_{m+1} = v_0$

2. $\gamma_{m+1} = \varepsilon(\rho P(v_0)\gamma_m + k(v_0)) + (1-\varepsilon)\gamma_m$

and γ_m will converge to a $c(v_0)$, the <u>dual variable at v_0</u>, at very little expense, since only 2 is involved. Let us now take our arbitrary v_0 to be v_{n+1} in Equation 2'''. Then the more closely γ approximates the dual variable $c(v_{n+1})$ (by successive application of 1,2 above) then the closer will $v_{n+2} = \underset{v \in V}{\operatorname{argmax}}\ \rho P(v)\gamma + k(v)$ be to v^* in 1',2'''. In effect, we can speed up convergence to v^* and c^* at very little additional computational expense, by more accurately calculating the dual variable $c(v_n)$ between maximizations.

The equations are as follows (v_0, γ_0 are arbitrary):

1". $v_{n+1} = \underset{v \in V}{\operatorname{argmax}}\ \rho P(v)[\gamma_n] + k(v)$

2.1 $\gamma_{n,0} = [\rho P(v_{n+1})[\gamma_n] + k(v_{n+1})]$

2.2 $\gamma_{n,m+1} = \frac{1}{2}[\rho P(v_{n+1})\gamma_{n,m} + k(v_{n+1})] + \frac{1}{2}\gamma_{n,m}$

2.3 $\gamma_{n+1} = \gamma_{n,m}$

We have inserted the [ ] at somewhat arbitrary places to assure $1'\gamma_{n,m}$ and $1'\gamma_n$ are zero. The actual implementation may be somewhat different for efficiency.

1.5 <u>Tracking Convergence - Monotonic Bounds</u>

From equation 2 (Section 1.3) we can write

$$J^{1*} = \rho P(v^*)c^* + k(v^*) - c^*$$

Suppose that v_{n+1} and c_n are suboptimal but

3. $$v_{n+1} = \underset{v' \in V}{\text{argmax}}\ \rho P(v')c_n + k(v')$$

Then

4. $$J^{1*} \le \text{maximum element of } (\rho P(v_{n+1})c_n + k(v) - c)$$

and

4.5 $$\text{minimum element of } (\rho P(v_{n+1})c_n + k(v_{n+1}) - c_n) \le J^{1*}$$

Since, after 1" (Section 1.4), 3 will hold, 4 and 4.5 provide simple bounds on J^{1*} at each step. In fact these bounds should be monotonic as n increases. These bounds are printed out after each optimization.

Another bound computed by the program are maximum and minimum average return on the present control v_n. This can be done, once again, by considering $V_0 = \{v_n\}$. In this case, evidently

$$v_n = \underset{v' \in V_0}{\text{argmax}}\ \rho P(v') + c_n + k(v')$$

so 4 and 4.5 will hold with J^{1*} being replaced with $J^1(v_n)$. These bounds are computed at the intermediate steps $\gamma_{n,m}$.

2. Input-Output Conventions

In this section we describe the conventions and data structure used in specifying an economic model to CIVP, and the names of the quantities computed and displayed by CIVP. We begin with the input.

2.1 Scalar Constants

In parentheses we show the values used in the present CIVP.

NPER - Number of periods in year (2)

NAGG - Number of aggregated crops (1)

NBIN - Number of aggregated storage locations (1)

DIM = NAGG + NBIN

NS - Number of discrete states (for all periods) (14)

2.2 State Vector Convention

DIM is obviously the state dimension. The convention used for numbering the state variables (different from ECON) is: $x_1 \ldots x_{NAGG}$ are the aggregated crop estimates and $x_{NAGG+1} \ldots x_{DIM}$ are the storage estimates (there are NBIN of them).

2.3 Matrix Data

Dimensions of each matrix are in brackets [ ]; number of rows is first, number of columns is second.

2.3.1 PLN [NAGG, NPER]

This matrix contains entries indicating the planting periods, growing periods, and non-growing periods for each crop. Specifically let $i \in \{1,\ldots,NAGG\}$ be an aggregated crop. Then in row i, a 1 will appear in column t where t is the period in which the first planting is done (see illustration). If there is a second planting period, a 2 appears, etc. We now divide

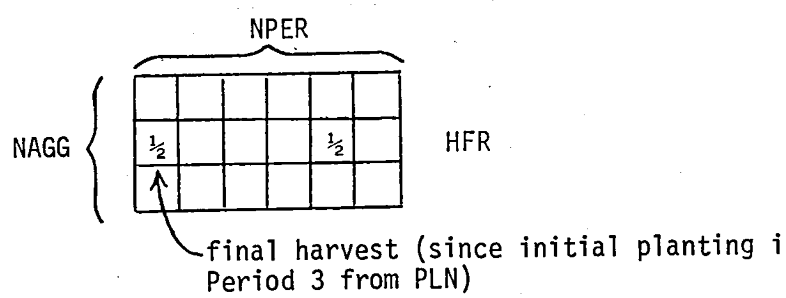
$$\text{NAGG}\left\{\overbrace{\begin{array}{|c|c|c|c|c|c|} \hline & & & & & \\ \hline -1 & -1 & 1 & 0 & 0 & 2 \\ \hline & & & & & \\ \hline \end{array}}^{\text{NPER}}\right. \qquad \text{PLN}$$

the remaining periods into two groups: A) periods in which the crop is growing <u>and</u> the final harvesting will <u>not</u> begin during that period; B) periods in which the crop's final harvesting is taking place, <u>or</u> periods after the final harvesting has taken place and before the first planting occurs. We define the <u>final harvest</u> to be the period in which all remaining crop is added to storage, so that the crop is 100% harvested by the next period. In A periods, 0 (zero) is to appear in PLN; in B periods, a -1.

One slight restriction of this convention is that the final harvesting cannot take place in the same period as the last planting: it must occur at least one period later. In the illustration, for example, last planting occurs in Period 6, final harvesting in Period 1. It is not possible for final harvesting to occur in Period 6.

2.3.2 HFR [NAGG, NPER]

For each aggregated crop i, time period t, HFR(i,t) is the fraction of crop harvested during that period. The final harvest period is therefore the first period since the initial planting for which the total fraction harvested is one.



2.3.3 NFC [DIM, NAGG]

This matrix shows which aggregated crops i feed which aggregated storage location j. Specifically NFC(j,i) = 1 if crop i's harvest goes to j; otherwise NFC(j,i) = 0. By the state variable convention (Section 2.2), j > NAGG; hence the first NAGG rows of NFC are unused.

2.4 Data for the Discretization Scheme

2.4.1 N [DIM, NPER]

N(i,t) is the number of discrete levels for state variable x_i at period t. Restrictions:

1) Totally harvested crops. Let us call a crop i totally harvested if it is 100% harvested and the first planting is not completed. Crop i will be totally harvested at t if and only if PLN(i,t-1) = -1. In the example given (Section 2.3.1), crop 2 is totally harvested in Periods 2 and 3. We require N(i,t) = 1 if i is a crop, and i is totally harvested during t.

2) $N(i,t)$ is odd

3) $3 \leq N(i,t) \leq 9$.

2.4.2 IND1 [NPER+1]

IND1 is a time-saving look-up array. Definition:

$$IND1(t) := \sum_{T=1}^{t-1} \prod_{\ell=1}^{DIM} N(i,t); \quad IND1(1) = 0$$

Note NS = IND1(NPER+1).

2.4.3 Discrete Levels

Discrete levels are defined by two matrices:

M : [DIM, NPER]--the center point of the discrete levels for each state variable x_i and period t is M(i,t).

STDE : [DIM, NPER]--an archaic name and convention as well. STDE(i,t) is the distance between discrete levels of x_i at t divided by 2.4.

Example: x_i at t has levels 0,10,20. Then M(i,t) = 10, STDE(i,t) = 10/2.4.

2.4.4 Information Model

The information variance, $var(\phi_i(t))$ is carried in a matrix called STD1.

STD1 : [DIM, NPER]--STD1(i,t) := $\sqrt{var(\phi_i(t))}$ where ϕ is defined in the Progress Report.

2.5 Numbering of Discrete States

Let $x_{i,j}(t)$ denote the j^{th} level $(1 \leq j \leq N(i,t))$ of state x_i at t. Then a typical state vector at period t will be some $(x_{1,j_1}(t), x_{2,j_2}(t),\ldots, x_{DIM,j_{DIM}}(t))$. There are NS of these discrete state vectors altogether, and they are ordered as follows: first the set of discrete vectors for t = 1, then t = 2, etc. In the sample runs there are 9 discrete vectors for t = 1, 5 for t = 2. The set of 9 discrete vectors for t = 1 is ordered by increasing levels with the most frequent changes in level occurring in the last state variable. Thus the ordering of the discrete states in the sample run is:

$(x_{1,1}(1),\ x_{2,1}(1))$

$(x_{1,1}(1),\ x_{2,2}(1))$

$(x_{1,1}(1),\ x_{2,3}(1))$

$(x_{1,2}(1),\ x_{2,1}(1))$

$(x_{1,2}(1),\ x_{2,2}(1))$

$(x_{1,2}(1),\ x_{2,3}(1))$

$(x_{1,3}(1),\ x_{2,1}(1))$

$(x_{1,3}(1),\ x_{2,2}(1))$

$(x_{1,3}(1),\ x_{2,3}(1))$

$(x_{1,1}(2),\ x_{2,1}(2))$

$(x_{1,1}(2),\ x_{2,2}(2))$

$(x_{1,1}(2),\ x_{2,3}(2))$

$(x_{1,1}(2),\ x_{2,4}(2))$

$(x_{1,1}(2),\ x_{2,5}(2))$

2.6 Output Names

A typical run of CIVP will print out values for various items. The naming conventions are:

XINT : [DIM, NPER]--a time-saving look-up table defined below.

FACT : [DIM, NPER]--for all x_i,t, the j^{th} level $(1 \le j \le N(i,t))$ of the quantization for x_i at t is

XINT(i,t) + j*FACT(i,t)

KM : [NS]--a vector of incremental welfare values for each of the discrete states, arrayed in the order described in 2.5. (Same as vector k in 1.2.) KM reflects the incremental welfare for the present suboptimal control v_m.

V : [NS, DIM]--each column of V displays the control vector applied for a particular discrete state vector. That is, if $x(t) = (x_{1,j_1}(t),\ldots, x_{DIM,j_{DIM}}(t))$, then

$$x(t+1) = x(t) + v(t) + \phi(t)$$

The sequence of suboptimal controls computed are the v_n of Section 1.3.

GAMMA : [NS]--same as γ in 1.3.

J1--same as 1.3.

SIGMA : [NS]--an approximation to the steady state probability distribution π vector which satisfies

$$\pi(v)P(v) = \pi(v)$$

as v_n converges, so does $\pi(v)$.

3. Running the Programs

Two steps are required to run the program. The first is to create a data file with the necessary economic data. The second is to apply CIVP to the data file.

3.1 Running INITIAL

A sample run of INITIAL appears of Page 10 of the computer output. INITIAL will ask for various scalars, vectors and arrays; all of these are defined in Section 2 except for STD, which is not used by CIVP and should be set to zero. The initial control matrix V should also be initialized to zero as this initial value is not actually used by the program, but instead a new value is recomputed immediately. Hence the entry for V is irrelevant to CIVP.

After the data entry, INITIAL will make some computations and write the results out into a file with logical name INIT (the actual name is specified previous to running the program via ASSIGN statement).

CAVEAT: Arrays must be dimensioned properly before running INITIAL. See Section 3.3.

3.2 Running CIVP (MAIN)

CIVP assumes two files exist: INIT and COMP are the logical names. INIT must have been initialized as described above. COMP should be a carbon copy of INIT. In the present program, no results are actually written out to COMP.

After creating COMP, INIT, and making the logical assignments, CIVP can be run. (See page 11.) CIVP will ask for the following:

Annual Discount factor

nd - determines the search time. The number of divisions of each dimension of the control space when searching for the optimal control. A subsequent fine search will then redivide the optimal segment into nd parts. The resulting search takes time approximately 2^*nd^{DIM} with an effective grid of $nd^{2 \cdot DIM}$ points.

nt - the M in Equation 2.3, Section 1.4.

Then CIVP will display the model data if the user so requests. Then the optimization routine begins. (Page 12.)

Each $$OPTIMIZE$$ represents a large iteration, an application of Equation 1", Section 1.4. After application of 2.1, the results (γ_n, v_{n+1}, and $k(v_{n+1})$) are displayed. Next, after 2.2 - 2.3 have been applied, the bounds on J^{1*} are shown, followed by bounds on $J^1(V_{n+1})$, and then $\gamma_{n,M}$ and σ (approximation to $\pi(v_{n+1})$). The program will then request to begin the next iteration.

For the examples solved, we achieved convergence in 4-8 iterations. Two examples are shown on pp. 12-15. The second example represents an "improved information" model over the first, and as a result the optimal average return came up from -2.136 to -1.930. The incremental welfare function was defined simply as $(v_2 + 5)^2$ (square deviation from consumption of 5 units).

3.3 Program Preparation

Some aspects of INITIAL and MAIN are model dependent, and must be modified for each model. Specifically:

3.3.1 Matrix Dimension

The common block matrix dimensions must agree with those given in Section 2 with the following exceptions:

a) mean and std are not used by the program.

b) any matrix with dimension NS (e.g., V, KM, Gamma, Sigma) in Section 2 must have at least NS for its dimension in the program. For example, if NS = 14, KM may be dimensional 14, 100, or 1000, but not 10. The inequality constraint enables the user to change the discretization scheme (and thus NS) without redimensioning any matrices.

In INITIAL:

vi : [DIM]

jp : [DIM]

In MAIN:

oldgam, p1, p2, p3 : [at least NS]

In OPTIMIZE:

vr, vlow, vhigh, vinc, j ; [DIM]

In UPDATE:

sigg, pi, gamg : [at least NS]

v1, j : [DIM]

In UBOUND:

vr, vlow, vhigh, vinc, j : [DIM]

In ITERATE:

Pi, gamma : [at least NS]

vlow, vhigh, vinc, j, vt : [DIM]

In PROW:

pi : [at least NS]

v, j, jp : [DIM]

3.3.2 Output Formats

Output formats must be modified suitably in MAIN, PMODEL.

3.3.3 Loops

Some loops in the program are nested DIM deep, so they must be changed when DIM changes. Specifically, in the routines INITIAL, lines 4500:4900, OPTIMIZE, lines 2100:2400, UPDATE, lines 2800:3500, ITERATE, lines 1900:2100, 3100:3500, PROW, lines 3400:3700, a loop similar to this appears:

```
    do - j1 = 1, n(1,j) }
    j(1) = j1           }  1
                           :
    do - j2 = 1, n(2,j) }  :
    j(2) = j2           }  DIM
         :
  - continue
```

This pattern should be repeated DIM times.

3.4 Program Compilation

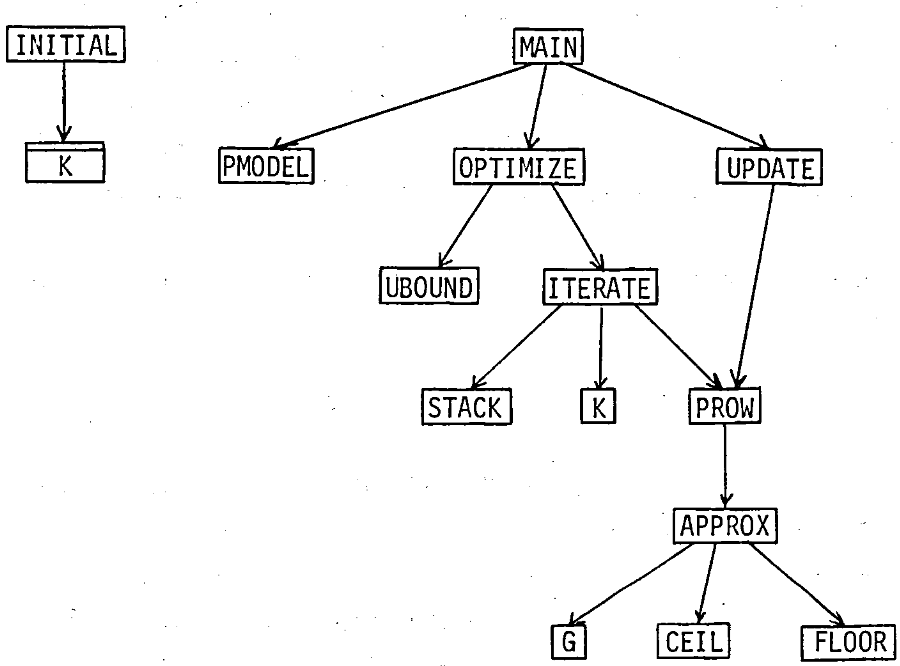
To create an executing file for INITIAL you must link an object module for INITIAL and an object module for K. (See file directory, p. 11.)

To create an executing file for MAIN (CIVP), you must link object modules for MAIN, PMODEL, OPTIMIZE, UPDATE, UBOUND, ITERATE, PROW, APPROX, STACK, and K.

4. Program Details

4.1 Program Subroutines

The structure of the program is illustrated in the following diagram, where each box represents a subroutine, and arrows represent possible calls:



Briefly, the purpose of each subroutine is as follows. INITIAL asks for user input for the economic model, initial values, etc., and sets up a disk file containing a common block with this information for use by the main program. K is the subroutine used to calculate rewards associated with the discrete states.

MAIN reads in the common block from the disk file and then calls other routines. The first, PMODEL, prints out the economic model data. MAIN then calls OPTIMIZE and UPDATE in alternation to find the optimal control vector, and associated optimal values.

OPTIMIZE is passed the approximate value vector γ_n and finds (see Section 1.4):

$$1''. \quad v_{n+1} = \underset{v \in V}{\operatorname{argmax}}\ \rho P(v)[\gamma_n] + k(v)$$

$$2.1 \quad \gamma'_{n,0} = [\rho P(v_{n+1})[\gamma_n] + k(v_{n+1})]$$

by searching the control space V, first using a coarse grid, then a fine grid centered on the best cause value. The bounds on the possible control values are found by UBOUND, and ITERATE is called first for the coarse search, then the fine search. The five highest values found are stacked by STACK, and then the values of v_{n+1} and γ' and k(v) are passed back to MAIN.

UPDATE improves the approximation to the state value vector by iteration of the following equation (see Section 1.4):

$$2.2 \quad \gamma_{n,m+1} = \tfrac{1}{2}[\rho P(v_{n+1})\gamma_{n,m} + k(v_{n+1})] + \tfrac{1}{2}\gamma_{n,m}$$

for M iterations, passes $\gamma_{n,M}$ back to MAIN, and MAIN $\gamma_{n+1} = \gamma_{n,M}$ so that the OPTIMIZE cycle can begin again.

In the above equations it is clear that the value of the transition matrix P must be recalculated with each new control v_{n+1}. In fact, if the state space is so large, it is not practical to store more than one row of P at a time. The subroutine which calculates a row of P for given control u is PROW. PROW uses APPROX to approximate the Gaussian distribution by a finite number of points. G is the Gaussian probability distribution function; CEIL is the ceiling function; FLOOR is the floor function.

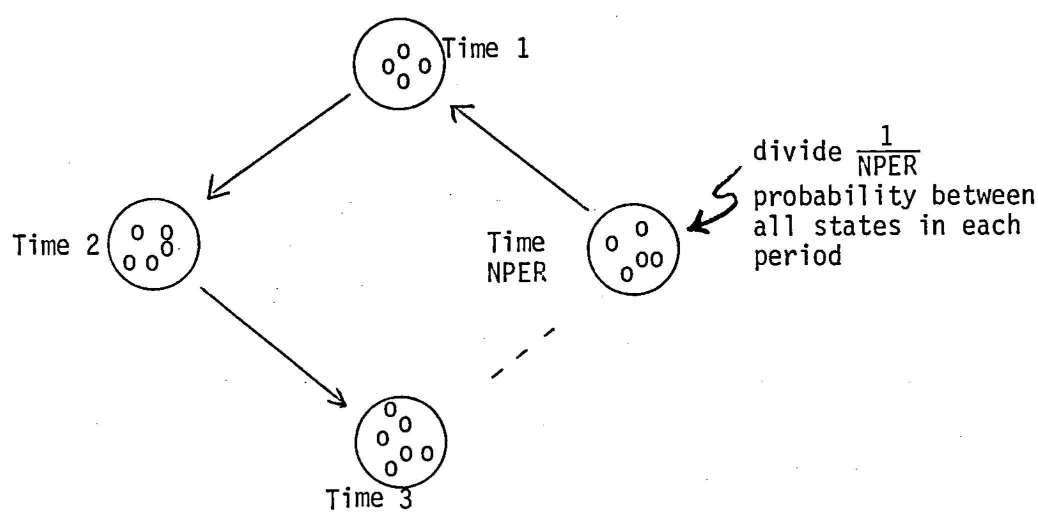
Let us now take a closer look at each subroutine.

4.2 INITIAL

Inputs: Accepts model data at teletype.

Purpose: To initialize a disk file (named 'INIT') with the economic and initial data.

Operation: The common block defined in lines 100-800 is equivalanced with an array called 'data' in lines 1500-1600, so that the entire common block can be written on the disk as a unit in line 6000. The disk fill is opened in lines 2000-2100. Then, several parameters and matrices are read in (see Section 2 for a description and definition of these arrays) with lines 2200-4200.



From the data that has been read in, several additional dependent arrays can now be initialized. In lines 4400-5500 the arrays GAMMA, SIGMA, and KM are initialized. Recall inducing scheme for the discrete states described in 2.5. The outermost index, line 4500, is the time period, j. The next most significant index is j_1 and represents the amount growing in the first aggregated crop (in this case there is only one aggregated crop). The least significant index, j_2, is the level of grain in the aggregated bin. The time period j, and levels j_1 and j_2 completely determine the discrete state. If there were more than one crop in a bin, then there would need to be more indices to specify a state.

The variable i is simply used to count the discrete states, for information about the i^{th} discrete state is stored in the i^{th} location of each array. Thus the incremental welfare of state i, is stored in km(i) (line 5200). It is calculated by calling the incremental welfare value function K (see details of K, Section 4.3). The initial probability of state i is stored in sigma(i) (line 5300). It is taken to be 1/(# of discrete states in period j * number of periods) so that $\sum_{i=1}^{ns} \text{sigma}(i) = 1$. Finally, gamma(i) (the initial discounted welfare estimate γ_0) is set to zero.

Then XINT and FACT are initialized in lines 5600-5900 by the formula given in Section 2.6. All the data is written out onto disk and this concludes the operation of INITIAL.

4.3 K (Page 9 of the computer output)

Inputs: A control $v = (v_1, \ldots, v_{DIM})$ which is usually some row of the feedback control matrix. A time period index t.

Result: A real number, the incremental welfare when control v is applied to state x at time t, k(v,x,t) of Section 1.2. Since this does not actually depend on x, x is not passed.

Operation: For the simplified model we simply took k = -(target consumption - actual consumption)^2. Since consumption = -v(2), this is accomplished by line 400.

4.4 MAIN

In MAIN and in subsequent subroutines, arrays and type declaration will appear in the following order: first common block definition, or whatever part is needed; second, the scratch common block SCR, if needed; and finally, local variables to the subroutine. In MAIN, all three types of declaration appear.

Input: Reals in disk file

Output: See Section 3.2

Operation: (Refer also to 3.2): First the disk file is read to initialize the common block (line 2500). The fill 'comp' is not used in this version of the program. The program then asks for discount factor, number of divisions nd for the coarse and fine search, and the number of update time nt, (Min Equation 2.3, Section 1.4). The annual discount factor is recomputed into a per-period discount factor ρ in line 3300. In line 3700 the economic data is printed out by a subroutine PMODEL. Then OPTIMIZE is called to find v_{n+1} as in Equation 1", Section 1.4. The results are printed out with lines 3900-4100. The next step is to update the value approximation and probability approximation γ and σ per Equations 2.1 - 2.3, Section 1.4, and also find bounds on the optimal return J^1 as described in Section 1.5. All of these functions are performed by the routine UPDATE. The updated values are returned to MAIN and printed out.

The calling conventions and detailed workings of PMODEL, OPTIMIZE, and UPDATE follow.

4.5 PMODEL

Arguments: None.

Returns: None.

Operation: Data is acquired via the common block, and the data is printed out in lines 1500-2900 if the user so desires.

4.6 OPTIMIZE

Arguments: ND - number of divisions in coarse and fine search

RHO - per-period discount factor

GAMMA - the γ_n as described previously (vector).

Returns: GAMMA - γ_{n+1}

OLDGAM - the original value of GAMMA, namely γ_n

V - the best control matrix found by searching the control space; the v_{n+1} as previously described.

KM - incremented welfare vector for the discrete states at this control v_{n+1}.

Operation:

1. Lines 1300-1400. GAMMA → OLDGAM
2. Lines 1700-4800. Find v_{n+1}, γ', km such that

(1) $$v_{n+1} = \underset{v \in V}{\text{argmax}}\ P(v)\gamma_n + k(v)$$

(2) $$\gamma' = \max_{v \in V} P(v)\gamma_n + k(v)$$

(3) $$km = k(v_{n+1})$$

Set V to v_{n+1}, GAMMA to γ', KM to km and return. Task 2 can be broken up as follows:

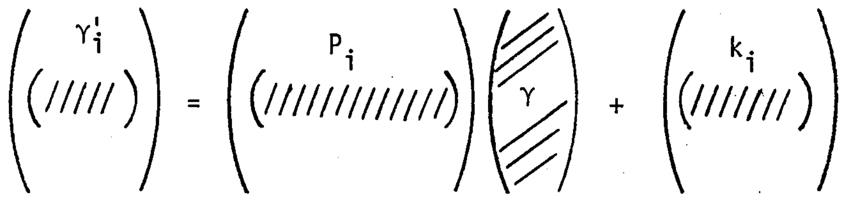
2a. Index through the discrete states. i is the state number, t is the time period, j1 is the first index, j2 is the second index (see Section 2.5 for more details of the state indexing scheme). Either i <u>or</u> (t, j_1, j_2) completely specify the discrete state. i is used for some cases, namely looking up positions in km, gamma, etc., whereas (t, j_1, j_2) is used when the acutal levels associated with the state are needed, i.e., the values of $x_1,\ldots,x_{DIM}$ are needed. Thus we will refer to a discrete state as either x_i or x_{t,j_1,j_2}.

Within lines 2600-4700 we are now concerned only with a single state x_i or x_{t,j_1,j_2}. Letting P_i be the i^{th} row of P(v), we are thus only concerned with maximizing the i^{th} component of γ' and v:

$$\gamma'(i) = \max_{v_i \in V_i} P_i\gamma + k_i$$

$$v(i) = \underset{v_i \in V_i}{\text{argmax}}\ P_i\gamma + k_i$$

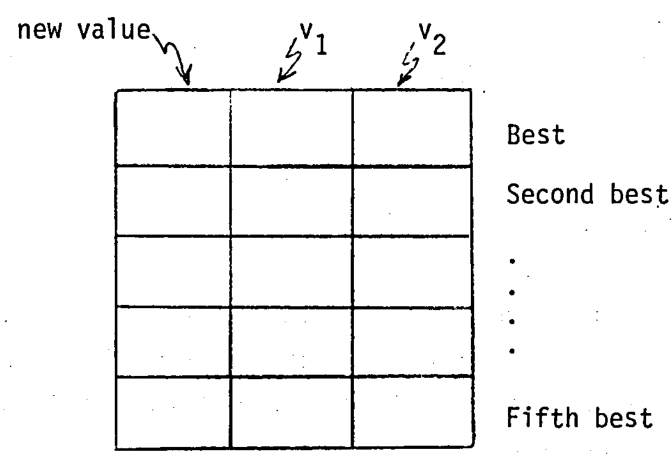
where V_i is the set of possible controls which can be applied to state i. Diagrammatically,



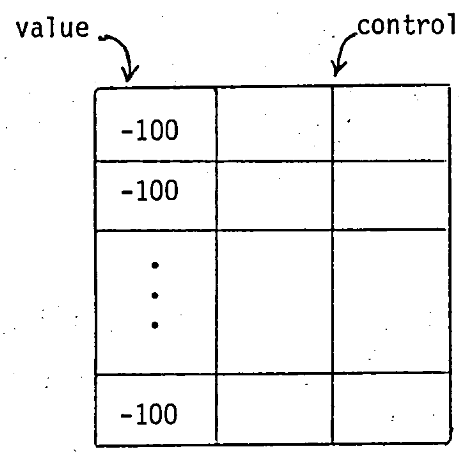
$$\gamma' = \max P_i \; \gamma + k_i$$

The i^{th} element of γ is the old value of state i; the i^{th} element of γ' is the new value is state i; the i^{th} row is v is the control applied to state i. We now proceed to task 2b.

2b. Initializing the stack. USTACK is a stack of the five best controls and the associated values which are found during the search. They are arranged as follows:



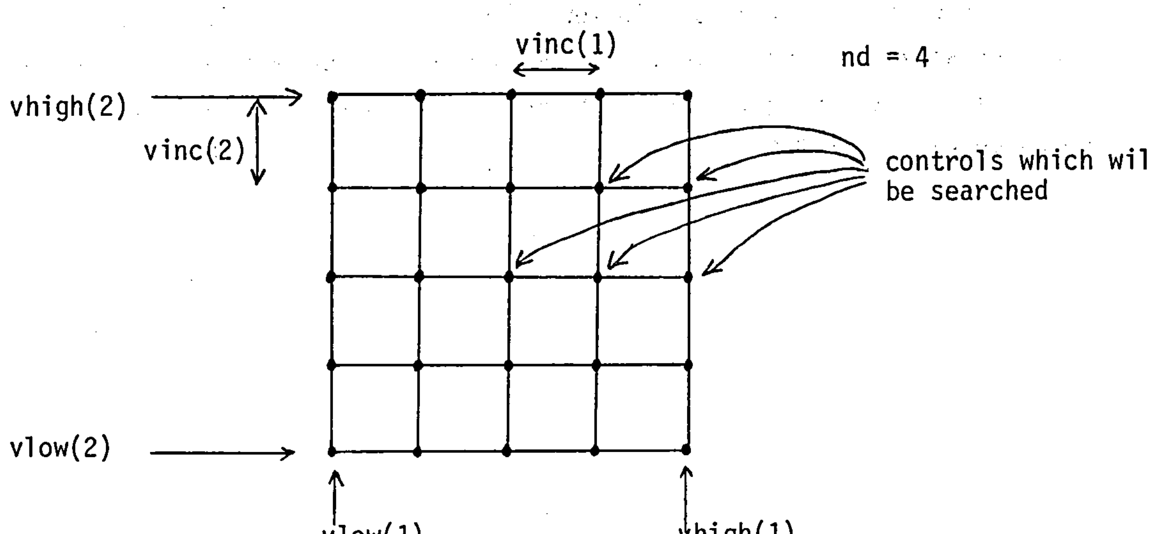
In each row is the new value, followed by the control applied (a 2 vector). The highest values are on top. But before the search begins, is is necessary to initialize the values to a low number, so that the real values later computed will fill the stock.



2c. <u>Calculating the bounds on admissible control</u>. In line 2800, UBOUND is called to calculate the bounds on admissible controls for state j,t. These bounds are returned in the vector vlow, vhigh, so that

$$\begin{array}{ccccc} \text{vlow}(1) & \le & v_1 & \le & \text{vhigh}(1) \\ \vdots & & \vdots & & \vdots \\ \text{vlow}(\text{dim}) & \le & v_{\text{dim}} & \le & \text{vhigh}(\text{dim}) \end{array}$$

nd is the number of divisions that will be searched in each dimension, and vinc is a vector of the increments that should be made in each dimension as the search proceeds. For dim = 2, say, UBOUND would return information defining the following grid:

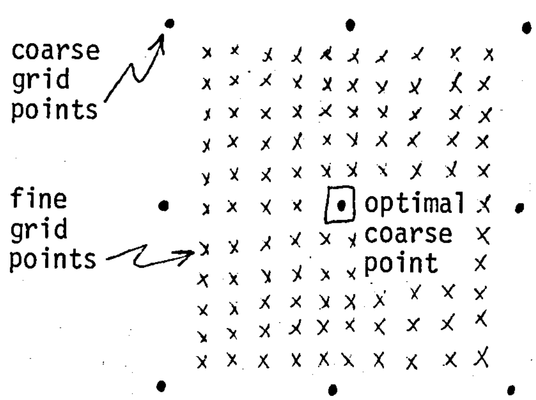


2d. Searching the coarse grid. Now having the bounds on the admissible controls for the state under question, we search through the possible controls to find a highest value. Namely, we find the v_i which maximizes

$$\gamma_i' = \max_{v_i \in V_i} P_i(v_i)\gamma + k_i$$

The actual search for the highest value is done by the subroutine ITERATE (line 3000), and the highest values and associated controls are returned on VSTACK as described above.

2e. Defining the fine grid. Around the optimal coarse point we now define a much finer grid. This is done by again calling UBOUND and specifying ur, the optimal coarse control, as the center of the fine search,



and width of the fine search to be twice the distance between the coarse points. Then in lines 3400-3500 ur is read off the top of the stack. In lines 3600-3700 the stack is again set to low values. Then in line 3800 UBOUND is called to find vlow, vhigh, and vinc for the fine grid.

2f. Searching the fine grid. Once the bounds on the fine grid are known, we call ITERATE once more to search the fine grid (line 4000). After this call, the maximum control and value are known and we can set γ_i' to be the highest value on the stack (line 4200), set the i^{th} row of v (the control matrix) to the best control of this state i (lines 4400-4600), and set km(i) to the incremental welfare for this control (line 4700). This concludes the subroutine OPTIMIZE, as we now have solved Equations (1), (2), (3) given above.

4.7 UBOUND (Page 5 of computer printout)

Note: For the ECON model, UBOUND must be reprogrammed. We give a description here of the simplified version.

Arguments: j,t - indices of a discrete state.

t1 - the time period subsequent to t

ur - a control vector, the center of the grid

prop - proportion of available control space which is to be searched. If prop = 0, all available control space is searched and ur is ignored.

nd - number of divisions in each dimension

Returns: vlow, vhigh, vinc - vectors of the low, high and increments in each control dimension.

Operation: 1. We look successively at each dimension i (line 1100); that is, the discrete state $x_{j,t}$ given as an argument is a vector

$$x_{j,t} = (x_{j,t,1},\ldots,x_{j,t,DIM})$$

where $x_{j,t,1}$ is the amount of grain growing in the first aggregated crop, $x_{j,t,2}$ in the second aggregated crop, and so forth, with $x_{j,t,DIM}$ being the amount of grain stored in the last aggregated bin. We consider each dimension separately and first calculate

$$m1 = x_{j,t,1}$$

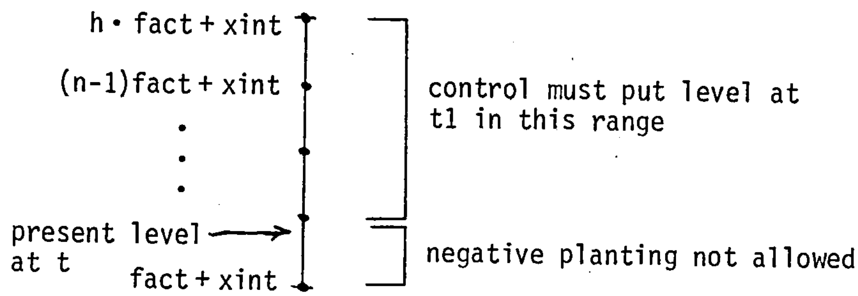
(line 1200), the amount of grain in that state variable. Then according to whether i is an aggregated crop or bin, t is a planting time or not, we branch to different parts of the program.

2. i is an aggregated crop. (True at line 20).

Case 2a. Preplanting or nonplanting season. (True at line 40). The only possible control is zero; obviously the amount planted must be zero in a nonplanting period.

Case 2b. Planting season. (True at line 50). We restrict the maximum sowing to the highest level representable in period t1. The highest level at t1 is n(i,t1) • fact(i,t1) + xint(i,t1); hence the amount sown must be less than vhigh as given in line 2000.

Diagram for planting season:



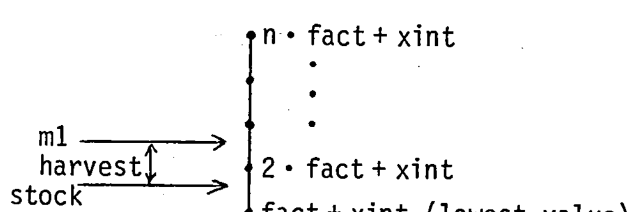
discrete values of variable i at t1

For a lower bound on the control, notice that the amount sown must cause a level at least as large as the lowest grid point, which is xint + fact. Thus the control must be at least xint + fact - ml. Also, the control must be positive; hence line 1900.

3. i an aggregated bin (True at line 30).

3a. First we calculate the amount of grain in storage assuming that no consumption, exports, or imports are made. This is done by summing the amount harvested from each feeding crop ij into the aggregated bin (lines 2300-2500). This amount available at t1 with no consumption, imports or exports is called ml and is different from "stock" which is the amount of stock at time t.

3b. The lowest stock level representable at t1 is xint + fact, so the lowest control is that which leaves us at that level at time t1, namely fact + xint - ml (line 2700).



discrete values available at t1

3c. The highest possible control is somewhat hard to calculate. In the ECON model, too, this calculation will be done somewhat differently. Certainly if

$$m1 > n \cdot fact + xint$$

then

$$v \le n \cdot fact + xint - m1$$

However,

$$v + stock \ge 0$$

so that

$$v \ge -stock$$

Hence,

$$v \le \min(-stock,\ n \cdot fact + xint - m1)$$

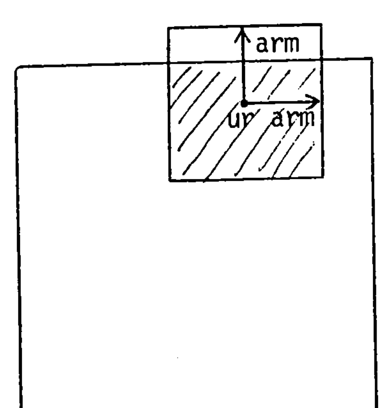
If it so happens, however, that this minimum is less than vlow, we must therefore take

$$vhigh = \max(vlow,\ \min(-stock,\ n \cdot fact + xint - m1))$$

(lines 2800-2900).

4. If prop = 0, we are done, so calculate vinc and return (line 3700).

5. If prop > 0, then we must recalculate the control bounds around the central control ur as shown below. This is carried out in lines 3300-3500.



4.8 ITERATE

Arguments: vlow, vhigh, vinc - lowest, highest and increments for control space

gamma - γ_n as described above.

j,t - indices of the state whose control is to be optimized

t1 - time period subsequent to t.

rho - per-period discount factor.

Returns: vstack - a stack of the five controls with highest values.

Operation: pi is used to store a row of the probability transition matrix $P_i(v)$. Lines 1900-2300 and 3100-3500 are "written out" loops for speed. These loop through the possible controls. (The $i=1$ statement at $i=1$ is superfluous and should be removed.)

The inner lines, 2400-2800, find the value of each control by the equation

$$\text{value} = \sum P_i(vt,ii)\cdot\gamma_n(ii) + k_i(vt,t)$$

where vt is the control under examination, γ_n is the old value vector, and $P_i(vt,\cdot)$ is the row of the probability transition matrix for state i under control vt. Once the value has been calculated, STACK is called in line 2900 to stack the value of the control and control itself should the value be one of the five best discovered so far in the search.

4.9 UPDATE

Arguments: rho - discount factor

v - feedback control matrix

km - imcremental welfare vector

gamma - $\gamma_{n,m}$ as described in Section 1.4.

oldgam - $\gamma_{n,m-1}$

sigma - approximation to probability distribution

Returns: gamma - $\gamma_{n,m+1}$ as described in Section 1.4.

oldgam - $\gamma_{n,m}$

sigma - updated approximation to probability distribution b1, b2 - lower and upper bounds on welfare

(The function of UPDATE is also overviewed in Sections 4.1 and 4.4.) Note: when UPDATE is past gamma $= \gamma_{n,0}$, then b_1 and b_2 will bound the optimal welfare, but when UPDATE is called thereafter with gamma $= \gamma_{n,m}$, $m > 0$, b_1 and b_2 will bound only the welfare of this particular feedback control matrix v. Thus the first call to update in MAIN is separated from the remaining calls.

Operation: Following Equation 2.2, Section 1.4, UPDATE calculates:

$$\gamma_{n,m+1} = \tfrac{1}{2}[\rho P(v_{n+1})\gamma_{n,m} + k(v_{n+1})] + \tfrac{1}{2}\gamma_{n,m}$$

and also

$$\sigma_{m+1} = \sigma_m P(v_{n+1})$$

What would thus be a straightforward matrix multiplication and addition is complicated by the fact that P is too large to be stored; one row is calculated at a time.

Indices: Let us begin by sorting out the indices. The row of P which we are calculating, and then the element of γ which can be calculated (notice, though, that an element of σ requires all the rows of P) is the index i.

Of course, a row of P corresponds to the outward transition probability from some discrete state, and this discrete state is indexed by t, j_1, j_2, etc., as described in Section 2.5.

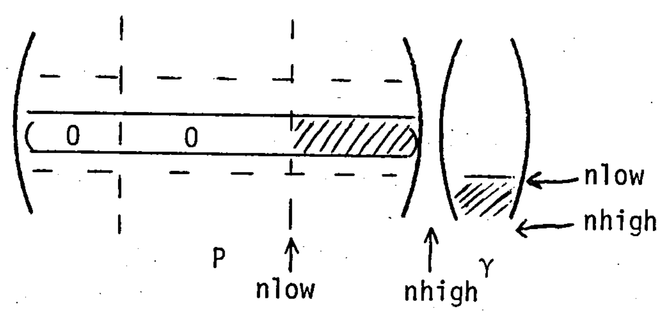
t1 is the time period after t.

nlow and nhigh are bounds on the indices (that is, the positions in γ or σ) of discrete states at time period t1. These are the states for which there is nonzero probability of going to it in the state i. Thus, these are the first and last elements in the i^{th} row of P which must be multiplied by γ (see diagram):

Partitioning of P:

$$\begin{array}{l|ccc} \text{Per 1} & 0 & \text{nonzero} & 0 \\ \text{Per 2} & 0 & 0 & \text{nonzero} \\ \text{Per NPER} & \text{nonzero} & 0 & 0 \\ \hline & \text{Per 1} & \text{Per 2} & \text{Per NPER} \end{array}$$

Thus a typical row of P, P_i, can be ignored except for nlow through nhigh:



In other words:

$$P_i\gamma = \sum_{ii=nlow}^{nhigh} P_i(ii)\gamma(ii)$$

Step 1. Initialize sigg, the updated sigma, to zero. Later sigg $\rightarrow$ sigma as required. (Lines 1600-1700, line 5600).

Step 2. Index a row of P, call it i or (j,t) (lines 2200-3500). Let v1 be the control applied to state x_i, the i^{th} row of v (line 3700). Let pi be the i^{th} row of P(v) (line 3800).

Step 3. Calculate $P_i\gamma_{n,m}$ = sum and $P_i\gamma_{n,m-1}$ = sum2 by above equation (lines 3900-4400). Also calculate $\sigma_m P(v_{n+1})$ by

$$\sigma_{m+1} = \sigma_m P(v_{n+1}) = \sum_{i=1}^{ns} \sigma_m(i)P_i(v_{n+1})$$

(line 4400).

Step 4. Set

$$\text{gamg} = \tfrac{1}{2}\rho P_i(v_{n+1})\gamma_{n,m} + k_i(v_{n+1}) + \tfrac{1}{2}\gamma_{n,m}(i)$$

(Notice a dcval still remains in this expression.) Calculate dcvalue in line 4600, and calculate bound (cf. Section 1.5) in lines 4700-4900. Here δvm2 is a calculated element of $(\rho P(v_{n+1})c_n + k(v_{n+1}) - c_n)$.

Step 5. (lines 5300-5600). Return the computed values in the proper arrays.

4.10 <u>PROW</u>

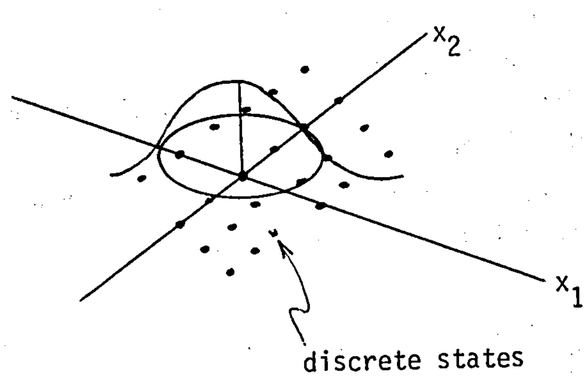
Function: Calculates a row of P(v).

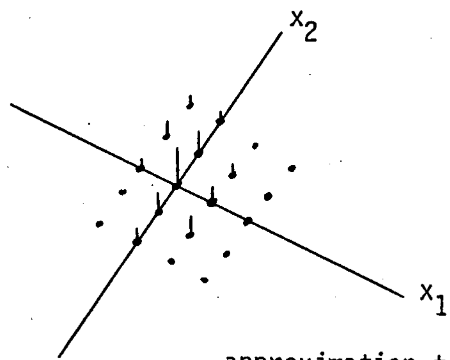
Arguments: j,t - discrete state index.
v - control (to be applied to this particular state).

Returns: pi - row of P(v) corresponding to probabilities leaving state j,t.

Note: Since the probability transition matrix is cyclic, certain elements of pi are constrained to be zero, but these elements are not actually zeroed by PROW. This must be remembered when using pi.

Operation: 1) Consider the discrete states at t1, arranged in a dim-dimensional grid. (See illustration, dim = 2.) We must somehow approximate the continuous probability distribution on this space, by a discrete probability distribution.



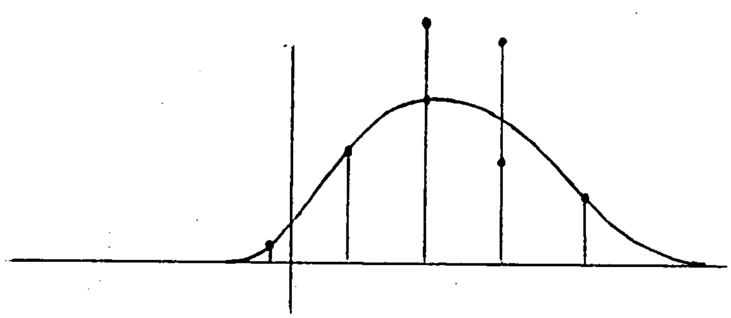


approximation to probability distribution

To simplify the problem a bit, we can assume that our discrete probability distribution is the product of dim independent probability distributions:

$$P_{ij} = P_i P_j \qquad 1 \le i \le h(1,t) \quad 1 \le j \le h(2,t)$$

and thus reduce our problem to finding an approximation to a one-dimensional continuous distribution:



A matrix P, dimensioned dim × 9, holds each of the dim independent probability distributions in successive rows. Lines 1400-2200 compute the discrete probabilities for the first NAGG state variables, lines 2400-3100 compute the probability distributions for the remaining NBIN state variables, and then lines 3300-4500 multiply the independent distributions together appropriately to calculate the discrete probability at each point on the dim-dimensional grid.

Step 1. For each crop, calculate the mean of the expected amount planted at the next time period under control v. This is done in lines 1500-1800. If t1 is a pre-planting period, then there is only one level for the discretization (namely 0), hence the probability of going to that state is one (line 2100). Otherwise APPROX is called to the PDF and stored in the i^{th} line of P (line 1900).

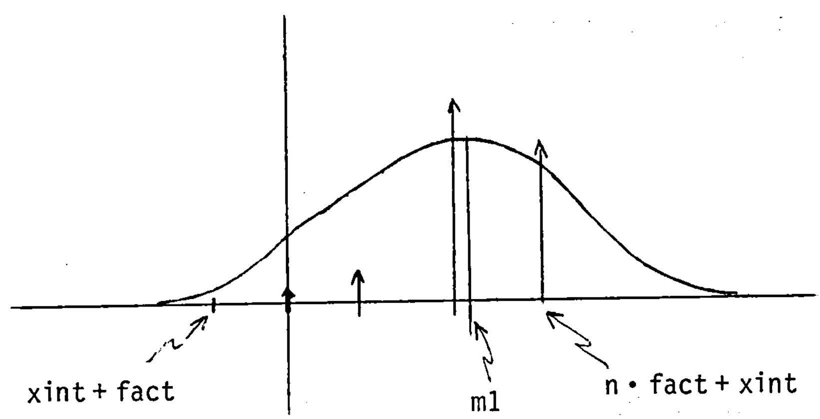
Step 2. For each bin, calculate the expected amount of grain stored at the next step. This is simply the present amount plus all harvest (lines 2600-2900). Again, APPROX calculates the PDF and stores it in the i^{th} line of P.

Step 3. Recall that a discrete state is ordered by t, j_1, j_2, etc. Line 4400 calculates

$$pi(i) = \prod_{ii=1}^{DIM} p(\text{line } ii, \text{ point } j_{ii})$$

the product distribution as above.

4.11 APPROX



Arguments: m1 - mean of probability distribution

sig1 - standard deviation of probability distribution

xint, fact - define the first discrete level (xint + fact) and distance between discrete levels (fact)

n - number of discrete levels

line - line of matrix a in which to place probability approximations

Returns: a - matrix for discrete probability distribution (only one row will be affected, namely "line")

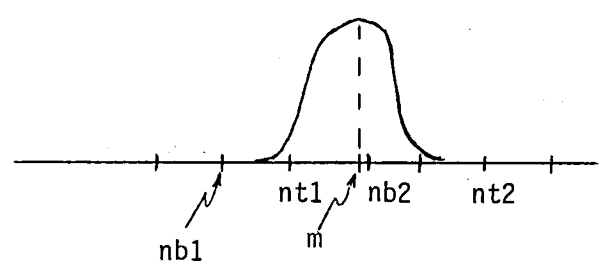
Operation: 1) The subroutine first checks to see that m1 is between xint + fact (the lowest discrete point) and n • fact + xint (the highest discrete point). If m1 is outside this range (it shouldn't be if UBOUND works properly), then APPROX prints an error message and aborts. Error detected is lines 1000-1500.

2) Just to make sure, set m1 in range if it is outside range (line 1900).

3) Zero out probability distribution a(line,•) (lines 2000-2100).

4) Each probability point will be computed by numerical integration of the continuous curve. The number of points for each point in the discrete approximation is nd and is computed in line 2200.

5) nb1 - the first index for which there is any significant probability (see figure)



nt2 - the last index for which there is any significant probability.

nt1 - the index just to the left of the mean.

nb2 - the index just to the right of the mean.

Note: nt1 = nb2 if mean of distribution lies exactly on a discrete point. This special case is taken into account in the program.

arm - 2.4 + sigl, i.e., the distance to which there is any significant probability, or the distance to the first or last discrete point from the mean, whichever is smallest.

6) Numerical integration (line 2900). This is a first approximation to a discrete PDF.

7) Probability refinement (lines 3200-6100) coaxes the mean of the discrete PDF to be the same as the mean of the continuous PDF. Let $\hat{m}$ be

the mean of the crude discrete PDF from numerical integration. Then

$$\hat{m} = \sum_{i=nb1}^{nt2} (i \cdot fact + xint) \cdot pi$$

To coax $\hat{m}$ to m, we define α_1 and α_2 such that

$$m = \sum_{i=nb1}^{nt1} (i \cdot fact + xint) \cdot \alpha_1 \cdot pi + \sum_{i=nb2}^{nt2} (i \cdot fact + xint) \cdot \alpha_2 \cdot pi$$

and also

$$1 = \sum_{i=nb1}^{nt1} Q_i \cdot \alpha_1 + \sum_{i=nb2}^{nt2} pi \cdot \alpha_2$$

letting

$$p1 \triangleq \sum_{i=nb1}^{nt1} pi \qquad p2 \triangleq \sum_{i=nb2}^{nt2} pi$$

$$s1 \triangleq \sum_{i=nb1}^{nt1} i*pi \qquad s2 \triangleq \sum_{i=nb2}^{nt2} i*pi$$

we then have

$$1 = \alpha_1 p_1 + \alpha_2 p_2$$

$$m = \alpha_1 \cdot fact \cdot s1 + \alpha_2 \cdot fact \cdot s2 + xint$$

p1, s1, p2, s2 are calculated in lines 3200-4100, with lines 4300-4700 taking care of the exceptional case nt1 = nb2.

We now rewrite the equations for α_1, α_2 as follows:

$$1 = \alpha_1 p_1 + \alpha_2 p_2$$

$$m = \alpha_1(fact \cdot s1 + xint \cdot p_1) + \alpha_2(fact \cdot s2 + xint \cdot p_1)$$

Redefine $s_1 = \text{fact}\cdot\text{si} + \text{xint}\cdot p_1$, $s_2 = \text{fact}\cdot\text{s2} + \text{xint}\cdot p_1$, (lines 4800, 4900), so that

$$\begin{pmatrix} p_1 & p_2 \\ s_1 & s_2 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = \begin{pmatrix} 1 \\ m \end{pmatrix}$$

Lines 5100-5300 solve this equation for α_1, α_2. Lines 5400-6100 then multiply p_{nb1} through p_{nt1} by α_1, and p_{nb2} through p_{nt2} by α_2, adjusting if necessary for nt1 = nb2.

APPENDIX B

OPTIMAL AND SUBOPTIMAL STATIONARY CONTROLS FOR MARKOV CHAINS

by

Pravin Varaiya

in IEEE Transactions on Automatic Control, Vol. AC-23, No. 3, pp. 388-394, 1978.

APPENDIX C

A DIFFERENTIAL THEORY OF MARKOV CONTROL

Steven N. Jones

Abstract

We consider the problem of controlling a Markov Process so as to minimize the long-run discounted (or undiscounted) cost. A new approach is taken, based on a matrix M which represents the difference in future state occupation caused by different starting states. Simple expressions result for the derivatives of the limiting state probability vector $\pi(u)$ and cost $J(u)$ with respect to changes in the applied control u. Using these derivatives, explicit formulas are derived for $\pi(u')$ and $J(u')$ where u' differs from u in the control of a single state, and for this case it is shown that the sign of $J(u') - J(u)$ depends on a very simply calculated discriminant. This leads to several new necessary and sufficient conditions for an optimal u*, which hold for both the discounted and undiscounted cases: optimality, first-order necessary conditions on the derivative are shown to be sufficient, Varaiya's necessary and sufficient condition for an optimal dual variable is extended to the discounted case, as is his bound $B(u) \geqq J(u) - J^*$, and several previous results are reproven from the differential perspective.

A DIFFERENTIAL THEORY OF MARKOV CONTROL

Steven N. Jones
Scientific Systems, Inc.
Cambridge, MA 02138

CONTENTS

1. Introduction
2. The Markov Control Problem
3. Problem Formulation
4. A Differential Theory
5. Optimality Conditions and Bounds
6. References

1. Introduction

The Markov control problem is defined in Section 2, previous work on the problem is reviewed briefly and the approach to be taken in this paper is outlined and the major results summarized. Section 3 gives a more mathematical formulation of the problem, and derives a succinct algebraic formulation of the problem which is proved equivalent in Section 4.

Section 4 comprises the differential theory: the differential state occupation matrix M is defined, derivatives are defined of $\pi(u)$ and $J(u)$, and simple expressions are derived for them in terms of M. An explicit formula for changes in π and J under statewise control policy changes is found, and the Monotonicity Theorem is proved. This theorem states that if u' differs from u in the control of a single state, then the sign of $J(u') - J(u)$ depends on the sign of the discriminant $\delta_i + \Delta_i c(u)$, where

$c(u)$ is a dual variable at u, $e_i'\delta_i = k(u') - k(u)$ (k is the vector of incremental costs associated with each state, e_i is the i^{th} row of the identity matrix) and $e_i'\Delta_i = P(u') - P(u)$. In fact, $J(a(u'-u)+u)$ is monotonic in a.

New optimality conditions are given in Section 5, whose proofs rely on the Monotonicity Theorem. It is proved that state-wise optimality is equivalent to global optimality, that global optimality is guaranteed by non-negative derivative, and the necessity and sufficiency of a dual variable is proven for the discounted case. Varaiya's bound $B(u) \geqq J(u) - J^*$ is also extended to the discounted case, and some previous results are reproven from the differential perspective.

2. The Markov Control Problem

Consider a perfectly observable Markov process x_t, $t=0,1,\ldots$, with finite state space $X=\{1,\ldots,s\}$, and a set of available controls $U(i)$ for each $i \in X$. We assume that by choosing the controls u_t according to a stationary control policy $u=(u(1)\ \ldots\ u(s)) \in U(1)\times\ldots\times U(s) = U$, so that $u_t = u(x_t) \in U(x_t)$, then the Markov process wil have a stationary state transition matrix $P(u)$. At each time t there is an incremental cost q_t, or reward $-q_t$, from the Markov process which depends on the state x_t and on the applied control $u_t = u(x_t)$: $q_t = k_i(u(i))$ under control policy u if $x_t = i$. Thus the statistics of q_t, $t=0,1,\ldots$, depend on the control policy not only through the functions $k_i(u)$, $i \in X$, but also through the statistics of x_t, $t=0,1,\ldots$ which are determined by x_0 and $P(u)$.

The problem considered here, which we call the Markov Control Problem, is to find stationary controls which minimize the long-run discounted or average cost. The long-run average cost starting in state j under control policy $u \in U$ is defined as:

$$J_j^1(u) = \lim_{T\to\infty} E\left\{\frac{1}{T+1}\sum_{t=0}^{T} k_{x_t}(u(x_t))\,|\,x_0 = j\right\} \qquad (j \in X) \tag{2.1}$$

which we call the "undiscounted" cost. It is well known that this limit converges for stationary control policies (Doob, [1]), and to guarantee that the long-run average cost is independent from the starting state, we make the following assumption:

<u>Strict Ergodicity Assumption</u>. For any $u \in U$, there is a $\pi(u)$ such that

$$\lim_{t\to\infty} P(u)^t = \underline{1}\pi(u) \tag{2.2}$$

where $\underline{1} = (1 \ldots 1)'$. ∎

Although most all of our results hold under the more general "single ergodic class" assumption (Varaiya [2]), which also guarantees that J_j^1 will not depend on j, we will restrict ourselves to the above condition for simplicity in the presentation.

Another type of cost frequently encountered is "discounted" cost; for a discount factor $0 \leq \rho < 1$:

$$J_j^\rho(u) = E\left\{\sum_{t=0}^{\infty} \rho^t k_{x_t}(u(x_t))\,|\,x_0 = j\right\} \qquad (j \in X) \tag{2.3}$$

$$J^\rho(u) = (J_1^\rho(u) \ldots J_S^\rho(u))' \tag{2.3.5}$$

In general $J_i^\rho \neq J_j^\rho$ even under the strict ergodic assumption, but there is a close relationship between the discounted and undiscounted costs: there exists a "dual variable" $c(u)$, which is defined in Section 4 for any $u \in U$, such that $\pi(u)c(u) = 0$ and

$$J^\rho(u) = \frac{1}{1-\rho} J^1(u)\, \underline{1} + c(u) \tag{2.4}$$

Thus $\pi(u)J^\rho(u) = J^1(u)/(1-\rho)$.

The <u>Markov Control Problem</u>, as considered in this paper, is to minimize $J^\rho(u)$ subject to $u \in U$ for a particular $0 \leqq \rho \leqq 1$. Since J^ρ is in general a vector, it is not immediately clear that all elements of J^ρ can be minimized simultaneously. However, by assuming perfect state observation (i.e., assuming that the controls u_t may depend on x_t), and by assuming that U is compact and P,k are continuously dependent on u, it can be proved that the elements of J^ρ can be minimized simultaneously and an optimum stationary control $u^* \in U$ exists which achieves this global minimum (Kushner, [3]). In fact, u^* will be optimal over <u>all</u> feedback controls (Kushner, [3]).

Many researchers have addressed the Markov control problem, finding necessary and sufficient conditions for the optimality of u^*, methods for finding bounds on $J^\rho(u^*)$, and algorithms for computing successive strategies whose cost approach the minimum. Most of this work has centered around some form of the following equation:

$$c^* = \min_{u \in U} \left(\rho P(u)c^* + k(u) - J^1(u)\underline{1}\right) \tag{2.5}$$

Any solution c* to this equation is called an optimal dual variable, and there is no confusion in notation as the c(u) mentioned above equals c* when u is an optimal control. It is known for $\rho = 1$ (Varaiya, [2]), and has been assumed for general ρ (and will be proved in this paper), that if c* is an optimal dual variable, then the minimizer of the right side is an optimal control policy. Furthermore, if u* is optimal, then there exists a c* (which we will show equals c(u*)) which satisfies:

$$(2.5.5) \qquad c^* = \rho P(u^*)c^* + k(u^*) - J^1(u^*)\underline{1}$$

Equations (2.5) and (2.5.5) thus constitute a necessary and sufficient condition for u* to be optimal, and it is interesting to review the previous results on the Markov control problem from this viewpoint.

Howard Algorithm. Consider (2.5) for $\rho = 1$ in the following form:

$$(2.6) \qquad c^* + J^1(u^*)\underline{1} = \min_{u \in U} (P(u)c^* + k(u))$$

Any solution c* of the above equation is an optimal dual variable (since it satisfies Eq. (2.5)), and it can be checked that $c^* + J^1(u^*)\underline{1}$ is also an optimal dual variable. Howard [4] and Schweitzer [5] showed under certain conditions that optimal dual variables are in a sense "stable fixed points" of the above equation, so that for the sequence beginning with an arbitrary v_0, and

$$(2.7) \qquad v_{i+1} = \min_{u \in U} (P(u)v_i + k(u)), \qquad i = 0,1,\ldots$$

the v_i's approach optimal dual variables, $v_{i+1} - v_i \to J^1(u^*)\underline{1}$, as we would expect from (2.6), and thus the minimizing u_i's approach minimum cost. Odoni then showed [6] that for each $i = 0,1,\ldots$, the largest element of $v_{i+1} - v_i$ upper bounds $J(u^*)$, and the smallest element lower bounds $J(u^*)$.

One slight problem with this algorithm is that the conditions for convergence are not fully general, but in most cases it is the most practical algorithm to use. An analogous algorithm for the discounted case is:

$$(2.8) \qquad v_{i+1} = \min_{u \in U} (\rho P(u) v_i + k(u))$$

and v_i is assured to converge to a definite dual variable since $\rho < 1$; however, to the author's knowledge, no analogy to the Odoni bound has been formulated.

Varaiya Algorithm ($\rho = 1$ only). For the undiscounted case, Varaiya [2] defined $Q(u) = P(u) - I$, and a "Hamiltonian" $H(u,c) = Q(u)c + k(u)$. Then Eq. (2.5) can be written as

$$(2.9) \qquad J^1(u^*)\underline{1} = \min_{u \in U} (Q(u)c^* + k(u)) = \min_{u \in U} H(u,c^*)$$

Varaiya proved the necessary and sufficient properties of (2.5) in this form. He then showed that for any c, $\min\limits_{u \in U} H(u,c) = H(u',c)$:

$$(2.10) \qquad \min_{i \in X} \min_{u \in U} H_i(u,c) \leqq J^1(u^*) \leqq J^1(u') \leqq \max_{i \in X} \min_{u \in U} H_i(u,c)$$

and gives a scheme for modifying c to bring the left and right sides of (2.3) closer together, so that $c \to c^*$, and $H \to H(u^*,c^*)$. This algorithm converges under the most general conditions (a single chain with transient states) but is formulated as a differential equation for c, rather than an algorithm which recursively computes a discrete series of c's.

Earlier work, and most of the results we have not reviewed here, have also relied on some form of Eq. (2.5), and we refer the reader to Ross [1], Kushner [3], Howard [4], or Bertsekas [8].

The Differential Approach. Our work takes a different approach to the problem, based on the concept of differential state occupation. We define $m_{ij}(u)$ to be the difference in total expected future occupation of state j in units of time if x_0 has probability one of being i rather than probability distribution $\pi(u)$. Take e_j as the j^{th} row of the $s \times s$ identity matrix. Then

$$(2.11) \qquad m_{ij} = (e_i - \pi)e_j' + (e_i - \pi)Pe_j' + (e_i - \pi)P^2e_j' + \ldots$$

Under the strict ergodicity assumption, this sum will be shown to converge for all i,j, and be continuous in u. The m_{ij} can be arranged into a differential state occupation M and

$$(2.12) \qquad M(u) = \sum_{t=0}^{\infty} (I - \underline{1}\pi(u))P^t(u)$$

In this paper we will show that M is a useful theoretical tool in Markov control. It is, for example, related to derivatives of $\pi(u)$ and J(u), has important algebraic properties useful in Eq. (2.5), leads to new and stronger optimality conditions, and relates the discounted to the undiscounted costs, for $c(u) = M(u)k(u)$. Let us review these applications in more depth.

Take $u, u' \in U$ and let $\Delta = P(u') - P(u)$. We can define a derivative of $\pi(u)$ in the direction of Δ as:

$$\text{(2.13)} \qquad \frac{d\pi}{d\Delta} = \lim_{\varepsilon \to 0} \frac{(P+\varepsilon\Delta) - \pi(P)}{\varepsilon}$$

We will show that this limit always exists and that

$$\text{(2.14)} \qquad \frac{d\pi}{d\Delta} = \pi\Delta M$$

In addition, since M is a function of u, we can define a derivative for M and

$$\text{(2.15)} \qquad \frac{dM}{d\Delta} = \lim_{\varepsilon \to 0} \frac{M(P+\varepsilon\Delta) - M(P)}{\varepsilon} = M\Delta M$$

Eq. (2.14) and (2.15) can be considered differential equations for $(\pi(P+\Delta),\ M(P+\Delta))$ in the independent variable Δ. When Δ consists of only one nonzero row, these equations can be solved analytically for $(\pi(P+\Delta),\ M(P+\Delta))$ in terms of $(\pi(P),\ M(P))$, and if Δ_i is the nonzero row of Δ, M_i is the i^{th} column of M, then

$$\text{(2.16)} \qquad \begin{pmatrix} \pi(u') \\ M(u') \end{pmatrix} = \begin{pmatrix} \pi(P+\Delta) \\ M(P+\Delta) \end{pmatrix} = \begin{pmatrix} \pi(u) \\ M(u) \end{pmatrix} + \frac{1}{1-\Delta_i M_i} \begin{pmatrix} \underline{1}\pi_i(u) \\ M_i(u)\Delta_i M(u) \end{pmatrix}$$

Notice for an arbitrary Δ, the sequence $(\pi,M)(P)$, $(\pi,M)(P+e_1\Delta_1)$, $(\pi,M)(P+e_1\Delta_1+e_2\Delta_2)$, leading to $(\pi,M)(P+\Delta)$ can be recursively computed from (2.16). Another application of Eq. (2.16) is our

Monotonicity Theorem. For $P(u') - P(u)$ having a single nonzero row i,

$$(2.17) \qquad \Delta_i M(u)k(u) + \delta \begin{pmatrix} \geq \\ = \\ < \end{pmatrix} 0 \qquad \text{iff} \qquad J^1(u') \begin{pmatrix} \geq \\ = \\ < \end{pmatrix} J^1(u)$$

where $e_i'\delta = k(u') - k(u)$, $e_i'\Delta_i = P(u') - P(u)$. □

All of these results extend to the discounted case, as a discounted M^ρ is defined as:

$$(2.18) \qquad M^\rho(u) = \sum_{t=0}^{\infty} \rho^t (I - \underline{1}\pi) P^t(u)$$

the differential equations (2.14) and (2.15) are supplemented by one for M^ρ, adding an additional row for $M^\rho(u')$ in Eq. (2.16), and the Monotonicity Theorem holds with slight modification.

Consider next the algebraic properties of M. Let $Q = \rho P - I$. Then for $0 \leqq \rho \leqq 1$,

$$(2.19) \qquad Q^\rho M^\rho = M^\rho Q^\rho = \underline{1}\pi - I$$

so by taking $c(u) = M^\rho(u)k(u)$, we see that $c(u)$ solves (2.5.5):

$$(2.20) \qquad \rho Pc + k - J^1\underline{1} = Qc + c + k - \underline{1}\pi k = (\underline{1}\pi - I)k + c + k - \underline{1}\pi k = c$$

and thus the optimal dual variable c^* in (2.5) can be taken to be

$$(2.21) \qquad c^* = M(u^*)k(u^*)$$

The proofs of optimality conditions in Section 5 will be facilitated by the following algebraic properties, which are interesting in their own right:

(2.22) $\quad \pi M^{\rho} = M^{\rho}\underline{1} = 0 \qquad 0 \leqq \rho \leqq 1$

(2.23) $\quad Q^{-1} = -(M^{\rho} + \frac{1}{1-\rho}\underline{1}\pi) \qquad 0 \leqq \rho < 1$

(2.24) $\quad J^{\rho}\cdot(1-\rho) = J^{1} + (1-\rho)\cdot M^{\rho}k \qquad 0 \leqq \rho \leqq 1$

Eq. (2.24) exhibits the relationship between discounted and undiscounted cost, and we see that by "normalizing" the discounted cost by a factor of $(1-\rho)$, $J^{\rho} \to J^{1}$ as $\rho \to 1$. This normalization will be assumed hereafter in the paper.

The differential theory described above leads to new and strengthened necessary and sufficient conditions on u*. First, it is proved that if a control u is optimal with respect to changes in the controls of single states, (i.e., u' differs from u in only one u(i)), then u is (globally) optimum. Obviously the converse holds so this is a necessary and sufficient condition.

Second, let $u \in U$, and

(2.25) $\quad \Phi(u) = \{P(u') - P(u),\ k(u') - k(u))\,|\,u \in U\}$

Then it is shown that

(2.26) $\quad \frac{dJ}{d\Delta,k} \geqq 0 \qquad \text{for all } (\Delta,k) \in \Phi(u)$

is a necessary and sufficient condition for u to be optimal.

A third result is an extension of Varaiya's necessary and sufficient condition:

$$(2.27) \qquad J^1(u)\underline{1} = \min_{u' \in U} \left(Q^1(u')c + k(u')\right) = \min_{u' \in U} H(c,u')$$

to the discounted case. In addition, Varaiya's bound $B(u) \geqq J^* - J(u)$ is extended to the discounted case.

We are presently working on improved algorithms based on this result and the new optimality conditions.

3. Problem Formulation

The state space of the Markov process is $X = \{1,\ldots,s\}$, the stationary control space U is a compact cartesian product $U = U(1)\times\ldots\times U(s)$, $k_i : U(i) \to R$ are continuous functions for each $i \in X$, and $P : U \to R^{s\times s}$ is continuous with the strict ergodic property (Eq. (2.2)) holding for each $P(u)$, $u \in U$. Take $J^1 : U \to R$ according to Eq. (2.1), and for $0 \leqq \rho < 1$ take $J^\rho : U \to R$ according to (2.3.5) times a normalization factor $(1-\rho)$. As we said earlier, this normalization factor insures that:

$$(3.1) \qquad \lim_{\rho \to 1} J^\rho(u) = J^1(u)\underline{1}$$

which we shall prove in Section 4. For any $0 \leqq \rho \leqq 1$, $u^{\rho *}$ is called <u>ρ-optimal</u> iff every element of $J^\rho(u^*) \leqq J^\rho(u)$ for all $u \in U$.

To arrive at a more succinct mathematical statement of the Markov control problem, consider an alternate expression for J^ρ, $0 \leqq \rho < 1$. Let $\Pi^\rho_{ij}(u)$ be the expected total discounted future occupation of state j under policy u (in units of time), given that $x_0 = i$. That is, including a normalization factor $(1-\rho)$ for consistency:

$$(3.2) \qquad \Pi^{\rho}_{ij}(u) = (1-\rho) \sum_{t=0}^{\infty} \rho^t e_i P^t(u) e_j'$$

where e_i is the i^{th} row of the $s \times s$ identity matrix. By arranging the $\Pi^{\rho}_{ij}(u)$ into an $s \times s$ matrix $\Pi^{\rho}(u)$ we have:

$$(3.3) \qquad \Pi^{\rho}(u) = (1-\rho) \sum_{t=0}^{\infty} \rho^t P^t = (1-\rho)(I - \rho P)^{-1}$$

and it follows that

$$(3.4) \qquad J^{\rho}(u) = \Pi^{\rho}(u) k(u)$$

We will show that $\lim_{\rho \to 1} \Pi^{\rho}(u) = \underline{1}\pi(u) = \lim_{t \to \infty} P^t(u)$, so that (3.4) holds for $\rho = 1$ also if $\Pi^1(u)$ is defined to be $\underline{1}\pi(u)$.

We can now formulate the Markov control problem more succinctly. For $\rho < 1$, Π^{ρ} is uniquely specified by the equation:

$$(3.5) \qquad Q^{\rho}\Pi^{\rho} = -(1-\rho)I$$

where $Q^{\rho} = \rho P - I$. Although (3.5) holds for $\rho = 1$, one additional constraint is needed to uniquely specify Π^1. In Section 4 we will see that $\Pi^{\rho}\underline{1} = \underline{1}$ for all $0 \leq \rho \leq 1$, so this constraint with (3.5) uniquely specifies Π^{ρ} and the Markov control problem can then be written:

$$(3.6) \qquad J^{\rho}(u^{\rho *}) = \min_{u \epsilon U} \{\Pi^{\rho}(u)k(u) \mid Q^{\rho}\Pi^{\rho} = -(1-\rho)I,\ \Pi^{\rho}\underline{1} = \underline{1}\}$$

We take Eq. (3.6) as the formal definition of the problem, $0 \leqq \rho \leqq 1$.

4. A Differential Theory

Let $(u(1)\ u(2)\ \dots\ u(i)\ \dots\ u(s)) = u \in U$ be given and suppose $u' = (u(1)\ u(2)\ \dots\ u'(i)\ \dots\ u(s))$ differs from u only in the control applied to i. The major results of this section are explicit expressions for $\pi(u')$, $J^{\rho}(u')$, and $M(u')$ in terms of $\pi(u)$ and $M(u)$, and the Monotonicity Theorem (Eq. (2.17)). These results lead to new optimality conditions in Section 5.

The development begins with discussion of the new matrix M and its properties. The derivatives of $\pi(P)$, $J(P)$ and $M(P)$ with respect to changes in P can then be expressed in terms of M. We will write a differential equation in $P(u) + a(P(u') - P(u))$ where a is the scalar parameter to be varied between 0 and 1, and by noting that $P(u')$ differs from $P(u)$ in only one row, we can solve the differential equation for $\pi(a)$, $M(a)$ and $J^{\rho}(a)$. Letting $a = 1$ we have the result mentioned above. Note that this result applies to both the discounted and undiscounted cases.

For any $u \in U$, $0 \leqq \rho \leqq 1$, define $A(u) = I - \underline{1}\ (u)$. Notice that $A^n = A$, and that $AP = PA$. Define:

$$(4.1) \qquad M^{\rho}(u) = \sum_{t=0}^{\infty} \rho^t A P^t(u) \qquad \text{for} \qquad 0 < \rho \leqq 1$$

and $M^0(u) = A$.

Since the magnitudes of all eigenvalues of P are no greater than one, this sum must converge for $\rho < 1$ (and uniformly). To see that it must converge for $\rho = 1$ also, recall that $\lim_{n\to\infty} P^n = \underline{1}\pi$ by the Strong Ergodicity Assumption, and since $(P - \underline{1}\pi)^n = (AP)^n = AP^n = P^n - \underline{1}\pi$, $\lim_{n\to\infty} (P - \underline{1}\pi)^n = 0$. Thus the geometric series:

$$(4.2) \qquad \sum_{n=0}^{\infty} (P - \underline{1}\pi)^n = \sum_{n=0}^{\infty} AP^n = M^1(u)$$

must converge (and uniformly also). In fact, since π is continuous in u (Varaiya, [2]), and P is continuous in u by definition, M^ρ must also be continuous in u. Also M^ρ can be defined in closed form by evaluating the left side of Eq. (4.2) and

$$(4.3) \qquad M^\rho = (I - \rho(P - \underline{1}\pi))^{-1} \qquad \text{for } 0 \leqq \rho \leqq 1 .$$

M has many interesting properties, some of which are given in the following theorem. Recall that $Q(u) = \rho P(u) - I$.

<u>Theorem 1</u>. a. $Q\underline{1} = -(1-\rho)\underline{1}$

b. $\pi Q = -(1-\rho)$

c. $\pi M = M\underline{1} = 0$

d. $QA = AQ = A$ if $\rho = 1$

e. $AM = MA = -A$

<u>Proof</u>: (a) through (c) follow easily from the definitions. (d) is due to $Q\underline{1} = \pi Q = 0$ when $\rho = 1$. For (e), we see that

$$(4.4) \qquad \rho MP = \sum_{t=0}^{\infty} \rho^{t+1} AP^{t+1} = \sum_{t=1}^{\infty} \rho^t AP^t = M - A$$

so $MQ = -A$. Also

$$(4.5) \qquad \rho PM = \sum_{t=0}^{\infty} \rho^{t+1} PAP^t = \sum_{t=0}^{\infty} \rho^{t+1} AP^{t+1} = M - A$$

since $PA = AP$, thus $QM = -A$. ∎

For $\rho < 1$, Q is invertible and it can be checked from the above relations that

$$(4.6) \qquad Q^{-1} = -M - \frac{1}{1-\rho} \underline{1}\pi$$

It is this "inverse" property which makes M particularly useful in derivations.

M^ρ relates Π^ρ to Π^1 and J^ρ to J^1: for $\rho < 1$,

$$(4.7) \qquad \Pi^\rho = (1-\rho) \sum_{t=0}^{\infty} \rho^t P^t = (1-\rho) \sum_{t=0}^{\infty} (\rho^t AP^t + \rho^t \underline{1}\pi P^t)$$

$$= (1-\rho)M^\rho + \underline{1}\pi = (1-\rho)M^\rho + \Pi^1$$

Since M is continuous in u, we see that $\lim_{\rho\to 1} \Pi^\rho = \Pi^1$, so equality of the first and last matrices in Eq. (4.7) holds for $\rho = 1$ also.

Using (4.7), we can express J^ρ in terms of J^1 and M^ρ:

$$(4.8) \qquad J^\rho = \Pi^\rho k = (1-\rho)Mk + J^1(u)$$

The quantity $M(u)k(u)$ appears so often we define it to be $c(u)$, and we will later see that this vector represents the relative costs of the states under k, or in effect is a "dual variable" of u.

We now turn to the application of M in the calculation of derivatives. The notion of a feasible direction is a preliminary concept.

Definition. Let $L = \{(P(u),\ k(u)) \mid u \in U\}$ and for any $u \in U$ call $\Phi(u) = L - (P(u),\ k(u))$ the set of all feasible directions from u. The convex hull of L is denoted $\overline{L}$. ∎

Lemma. $\overline{L}$ satisfies the Strict Ergodicity Assumption iff L does. ∎

If $(\Delta,\delta) \in \Phi(u)$, define the one-sided derivatives

$$(4.9)\qquad \frac{d\Pi^{\rho}}{d\Delta} = \lim_{\varepsilon\to 0^{+}} \frac{\Pi^{\rho}(P+\varepsilon\Delta) - \Pi^{\rho}(P)}{\varepsilon}$$

$$(4.10)\qquad \frac{dJ^{\rho}}{d\Delta,\delta} = \lim_{\varepsilon\to 0^{+}} \frac{J^{\rho}(P+\varepsilon\Delta,\ k+\delta) - J^{\rho}(P,k)}{\varepsilon}$$

if the limits exist.

Theorem 2. Let $u \in U$ be given and let $(\Delta,\delta) \in \Phi(u)$. Then $\frac{d\Pi^{\rho}}{d\Delta}$, $\frac{dJ^{\rho}}{d\Delta,\delta}$, and $\frac{dM^{\rho}}{d\Delta}$ exist for all $0 \leqq \rho \leqq 1$ and

$$(4.11)\qquad \frac{d\Pi^{\rho}}{d\Delta} = \rho\Pi^{\rho}\Delta M^{\rho}$$

$$(4.12)\qquad \frac{dJ^{\rho}}{d\Delta,\delta} = \Pi^{\rho}(\delta + \rho\Delta Mk)$$

$$(4.13)\qquad \frac{dM^{\rho}}{d\Delta} = \begin{cases} M^{1}\Delta M^{1} & \rho = 1 \\ \rho M^{\rho}\Delta M^{\rho} + \frac{1}{1-\rho}\,\underline{1}\pi\Delta(\rho M^{\rho} - M^{1}) & 0 \leqq \rho < 1 \end{cases}$$

If $\varepsilon(\Delta,\delta)$ satisfies the Strong Ergodicity Property for small enough ε, then the above derivatives are two-sided.

Before turning to the proof, we will show that Eq. (4.11) can be derived from simple intuitive reasoning. Consider first a change $\Delta = \varepsilon e_i'(e_{j_1} - e_{j_2})$, a small perturbation in P which adds probability ε to p_{ij_1} and subtracts probability ε from p_{ij_2}. For small ε, what is $\pi_k(P + \varepsilon e_i'(e_{j_1} - e_{j_2})) - \pi_k(P)$? In changing only one row in P we expect the Markov process to run, intuitively speaking, as it normally would except when leaving state i. When the process is in state i, however, there is an added probability of ε of going to j_1 and ε less probability of going to j_2. We can explore the change in overall behavior by analyzing the effect of each occupation of i.

Recall that m_{j_1k} is the difference in occupation of state k by starting in j_1 rather than π, and m_{j_2k} is the difference in occupation of k by starting in j_2 rather than π. Thus

$$m_{j_1k} - m_{j_2k} \tag{4.14}$$

represents the difference in future occupation of state k when starting in j_1 rather than j_2, and ε times (4.14) must be the difference in total occupation of k each time the Markov process is in state i. Since state i occurs with frequency π_i, we expect an <u>average</u> difference in occupation of state k to be:

$$\pi_i \varepsilon (m_{j_1k} - m_{j_2k}) \tag{4.15}$$

Now any $\varepsilon\Delta$ can be expressed as:

$$\sum_{i,j_1} \varepsilon e_i' e_{j_1} \Delta_{ij_1} = \sum_{i,j_1} \varepsilon e_i' \Delta_{ij_1} (e_{j_1} - e_{j_2}) \tag{4.16}$$

since $\sum_{j_1}^{\infty} \Delta_{ij_1} e_{j_2} = 0$, and so

$$(4.17) \qquad \pi_k(P) - \pi_k(P+\varepsilon\Delta) \simeq \sum_{i,j_1} \pi_i \varepsilon \Delta_{ij_1}(m_{j_1 k} - m_{j_2 k})$$

$$= \sum_{i,j_1} \varepsilon\pi_i \Delta_{ij_1}(m_{j_1 k}) = \varepsilon\pi\Delta M_k$$

and therefore:

$$(4.18) \qquad \pi(P+\Delta) - \pi(P) \simeq \varepsilon\pi\Delta M$$

Let us now prove Theorem 2 formally.

Proof: For $\varepsilon \geqq 0$ let $\Pi_\varepsilon^\rho = \Pi^\rho(P+\varepsilon\Delta)$, $Q_\varepsilon^\rho = \rho P + \rho\varepsilon\Delta - I$, and for $\varepsilon > 0$ take $D_\varepsilon = (\Pi_\varepsilon^\rho - \Pi_0^\rho)/\varepsilon$. Since

$$(4.19) \qquad \Pi_\varepsilon^\rho Q_0 = -(1-\rho)I - \rho\varepsilon\Pi_\varepsilon^\rho\Delta$$

$$(4.20) \qquad \Pi_0^\rho Q_0 = -(1-\rho)I$$

we can get an equation for D_ε by subtracting (4.20) from (4.19) and dividing by ε:

$$(4.21) \qquad D_\varepsilon Q_0^\rho = -\rho\Pi_\varepsilon^\rho\Delta$$

For $\rho < 1$, $\varepsilon > 0$, Eq. (4.21) has a unique solution for D_ε since Q^ρ is invertible. For $\rho = 1$, however, (4.21) yields only s-1 linearly independent equations, but the one additional independent condition

$$(4.22) \qquad D_\varepsilon \underline{1} = 0$$

specifies D_ε uniquely for all ρ. It can be checked that $D_\varepsilon = \rho\Pi^\rho_\varepsilon \Delta M^\rho$ is a solution to the above equations:

$$(4.23) \qquad (\rho\Pi^\rho_\varepsilon \Delta M^\rho)Q_0 = -\rho\Pi^\rho_\varepsilon \Delta A = -\rho\Pi_\varepsilon \Delta$$

and Eq. (4.22) is satisfied since $M^\rho \underline{1} = 0$. Thus

$$(4.24) \qquad \frac{d\Pi^\rho}{d\Delta} = \lim_{\varepsilon\to 0^+} D_\varepsilon = \lim_{\varepsilon\to 0^+} \rho\Pi^\rho_\varepsilon \Delta M^\rho = \rho\Pi^\rho \Delta M^\rho$$

Eq. (4.12) follows from (4.24) and the chain rule.

Eq. (4.13) can be derived for $\rho < 1$ using (4.7) rewritten in this form:

$$(4.25) \qquad M^\rho = \frac{1}{1-\rho}(\Pi^\rho - \Pi^1) \qquad (0 \leqq \rho < 1)$$

For the case $\rho = 1$, $\frac{dM^1}{d\Delta}$ can be calculated by defining a M_ε in complete analogy to Π_ε. ∎

Consider again the situation $u = (u(1) \ldots u(i) \ldots u(s))$ and $u' = (u(1) \ldots u'(i) \ldots u(s))$. The above derivatives can be used to solve for $\Pi^\rho(u')$, $J^\rho(u')$, and $M^\rho(u')$ in terms of $\Pi^\rho(u)$ and $M^\rho(u)$. Let $\Delta = P(u') - P(u)$; Δ has only a single non-zero row and there exists a Δ_i such that $\Delta = e_i'\Delta_i$. Also $\delta = k(u') - k(u) = e_i'\delta_i$ for some scalar δ_i. Let a be a scalar parameter, and define

$$(4.26) \qquad (P(a),k(a)) = (P(u) + a(P(u')-P(u)),k(u) + a(k(u')-k(u)))$$

Then for any value of a between zero and one inclusive,

$$(4.27) \qquad (P(a),k(a)) \in \overline{\Phi}(u)$$

where the overbar indicates the convex hull. The derivatives of all quantities exist and are two-sided for $0 < a < 1$, and are one-sided for $a = 0$ and $a = 1$, by Eq. (4.27). We will find $\Pi^\rho(u')$ and $M^\rho(u')$ by writing a differential equation for $\Pi(a)$ and $M^\rho(a)$, solving the differential equation, and then taking $a = 1$. To begin, let M_j^ρ be the j^{th} column of M^ρ, $v_j^\rho(a) = \Delta_i M_j^\rho(a)$ for $j = 1,\ldots,s$. Then

$$(4.28) \qquad \frac{dv_j^\rho(a)}{da} = \Delta_i \frac{dM_j^\rho(a)}{da} = \Delta_i M^\rho \Delta M_j^\rho$$

but this expression is a function of v_i^ρ and v_j^ρ since

$$(4.29) \qquad \Delta_i M^\rho \Delta M^\rho = \rho(v_1^\rho \ \ldots \ v_s^\rho)e_i^! \Delta_i M_j^\rho = \rho v_i^\rho v_j^\rho$$

Thus we can solve first for $v_i^\rho(a)$ and then all of the other v_j^ρ and get:

$$(4.30) \qquad v_j^\rho(a) = \frac{v_j^\rho(0)}{1 - \rho a v_j^\rho(0)}$$

Since $M_j^\rho(a)$ exists and is finite for $a = 1$, and $v_j^\rho(a)$ is continuous, we must have also:

$$(4.31) \qquad 1 - \rho v_j^\rho(0) = 1 - \rho\Delta_i M_j \geqq 0 \qquad (0 \leqq \rho \leqq 1,\ i,j \in \{1,\ldots,s\})$$

an ancillary fact we will use in proving the Monotonicity Theorem.

To continue, we can next solve for $\Pi^\rho(a)$, since the derivative of the j^{th} column of Π_j^ρ,

$$(4.32) \qquad \frac{d\Pi_j^\rho(a)}{da} = \rho\Pi^\rho e_i' \Delta_i M_j^\rho = \rho\Pi_i^\rho v_j^\rho$$

Again solving first for Π_i^ρ, and then Π_j^ρ, we get

$$(4.33) \qquad \Pi_j^\rho(a) = \Pi_j^\rho(0) + \Pi_i^\rho(0)\ \frac{av_j^\rho(0)}{1-\rho av_i^\rho(0)}$$

Letting $a = 1$, and writing (4.33) in matrix form, we arrive at an expression for $\Pi^\rho(u')$:

$$(4.34) \qquad \Pi^\rho(u') = \Pi^\rho(u) + \frac{1}{1-\rho\Delta_i M_i}\ \Pi_i^\rho \Delta_i M^\rho$$

where M_i^ρ is the i^{th} column of M^ρ, and Π_i^ρ is the i^{th} column of Π_i^ρ.

Once we have the v_j^ρ's, it is easy to solve for M^1, since

$$(4.35) \qquad \frac{dM_j^1(a)}{da} = M_i^1 v_j^1$$

and (4.35) must have a solution in the same form as the solution to Eq. (4.32):

$$(4.36) \qquad M^1(u') = M^1(u) + \frac{1}{1-\Delta_i M_i^1}\ M_i^1(u)\Delta_i M^1$$

To get M^ρ for $\rho < 1$, use Eq. (4.7):

$$(4.37) \qquad M^\rho(u') = \frac{1}{1-\rho}\ (\Pi^\rho(u') - \Pi^1(u'))$$

We now have analytical expressions for $\Pi^\rho(u')$ and $M^\rho(u')$ in terms of

$\Pi^\rho(u)$, $M^\rho(u)$, $\Pi^1(u)$ and $M^1(u)$. Since Π^ρ is a function of M^ρ and Π^1, we see that the triple $(\Pi^1(u'), M^1(u'), M^\rho(u'))$ can be explicitly calculated from the triple $(\Pi^1(u), M^1(u), M^\rho(u))$, when u' is different from u in the control of a single state.

Let us now turn to a calculation of $J^\rho(u')$ in terms of the last triple; this is certainly possible since

$$(4.38) \qquad J^\rho(u') = \Pi^\rho(u')k(u')$$

We spare the reader the necessary algebra which reduces Eq. (4.38) to the following:

$$(4.39) \qquad J^\rho(u') = J^\rho(u) + \frac{\delta_i + \rho\Delta_i M^\rho k}{1 - \rho\Delta_i M_i} \Pi_i^\rho(u)$$

where all of the quantities on the right side are taken at u.

With this expression we can now prove the following theorem.

<u>Theorem 3</u>. (Monotonicity Theorem). Let u' differ from u in the control of a single state, $e_i'\Delta_i = P(u') - P(u)$, $e_i'\delta_i = k(u') - k(u)$. Then

a. $$J^\rho(u') - J^\rho(u) \begin{pmatrix} \geqq \\ = \\ \leqq \end{pmatrix} 0 \qquad \text{if} \qquad \delta_i + \rho\Delta_i c^\rho(u) \begin{pmatrix} > \\ = \\ < \end{pmatrix} 0$$

where $c^\rho(u) = M^\rho(u)k(u)$. The inequality must be strict for at least one element of $J^\rho(u') - J^\rho(u)$; hence the "if" is an "if and only if" for $\rho = 1$.

b. $J(P + ae_i'\Delta_i,\ k + ae_i'\Delta_i)$ is monotonic in a for $0 \leqq \rho \leqq 1$.

Proof: (a) Recall from Eq. (4.31) that $1-\rho\Delta_i M_i$ must be greater than zero. Since the elements of $\Pi_i^\rho(u)$ are non-negative, and $\Pi_i^\rho(u)$ has at least one strictly positive element (namely $\Pi_{ii}^\rho(u)$), (a) follows.

(b) Follows from Lemma, Eqs. (4.39) and (4.31). ∎

5. Optimality Conditions and Bounds

In this section several new necessary and sufficient conditions for a global optimum u* are proved. In addition, several previous results are reproved or extended, as the differential viewpoint offers a new perspective, and in most cases, a simpler proof.

Consider the following conditions for a fixed $u \in U$:

C1. $J^\rho(u') \geqq J^\rho(u)$ all $u' \in U$

C2. $J^1(u) = \min\limits_{u' \in U} (Q^\rho(u')c^\rho + k(u'))$ where $c^\rho = M^\rho(u)k(u)$

C3. $J^\rho(u) = \min\limits_{u' \in U} (\rho Q^0(u')c^\rho + k(u'))$ where $c = M(u)k(u)$

C4. $\delta + \rho\Delta M(u)k(u) \geqq 0$ all $(\Delta,\delta) \in \Phi(u)$

C5. $J^\rho(u') \geqq J^\rho(u)$ for all $u' \in U$ s.t. for some $i \in X$, $u'(j) = u(j)$ unless $j = i$ all $j \in X$

C6. $\dfrac{dJ^\rho(P(u),k(u))}{d\Delta,\delta} \geqq 0$ all $(\Delta,\delta) \in \Phi(u)$

C1 is of course a statement of global optimality. C5 and C4 are new;

C2 is known to be equivalent to C1 when $\rho = 1$ (Varaiya, [2]); C3 and C6 are new.

Theorem 4. All of the above conditions are equivalent. Thus, any condition implies u is a global optimum and the solution to the Markov control problem.

We will need this preliminary lemma:

Lemma 2. Let $u \in U$, $0 \leqq \rho \leqq 1$, $c^\rho(u) = M^\rho(u)k(u)$. Then

$$(5.1) \qquad J^1(u)\underline{1} = Q^\rho(u)c^\rho(u) + k(u)$$

Proof: $Q^\rho(u)c^\rho(u) = Q^\rho(u)M^\rho(u)k(u) = -A(u)k(u) = (\underline{1}\pi - I)k(u) = J^1(u)\underline{1} - k(u)$ ∎

Proof of Theorem 4: (C1↔C2). Obviously C1→C2. For any u', $J_1(u') = Q^\rho(u')c^\rho(u') + k(u')$ by Lemma 2. If C2 holds, then $J_1(u) \leqq Q^\rho(u')c^\rho(u) + k(u')$. Since $\Pi^\rho(u)$ has only non-negative elements

$$\begin{aligned}
(5.2) \qquad \Pi^\rho(u')J_1(u) &= \Pi^\rho(u')\underline{1}\pi = J_1(u) \\
&\leqq \Pi^\rho(u')Q^\rho(u')c^\rho + \Pi^\rho(u')k(u') \\
&= (\underline{1}\pi(u') + (1-\rho)M^\rho(u')Q^\rho(u')c^\rho + J^\rho(u') \\
&= -(1-\rho)c + J^\rho(u')
\end{aligned}$$

Thus $J^\rho(u) = J_1(u) + (1-\rho)c \leqq J^\rho(u')$ which is condition C1. (This proof is in direct analogy to Varaiya [2]).

(C2↔C3). These two equations differ only by a constant factor $(1-\rho)c^\rho$.

(C2↔C4). By Lemma 2, C2 is equivalent to the statement:

(5.3) $\quad (\rho(P+\Delta)-I)c^{\rho}+k+\delta \geqq (\rho P-I)c^{\rho}+k$

which is equivalent to

(5.4) $\quad \rho\Delta M^{\rho}k + \delta_i \geqq 0$

which is C4.

(C4↔C5). Follows directly from the Monotonicity Theorem.

(C5↔C6). C6→C5 by the Monotonicity Theorem. Suppose then that condition -C6 holds, so that for some Δ,δ,

(5.5) $$\frac{dJ^{\rho}(P,k)}{d\Delta,\delta} = \Pi^{\rho}(\delta+\rho\Delta M^{\rho}k) \ngeq 0$$

Since all of the elements of Π^{ρ} are non-negative, there must be an $i \in X$ such that

(5.6) $\quad \delta_i + \rho\Delta_i M^{\rho}k < 0$

Again, by the monotonicity theorem, $J_i^{\rho}(P+e_i'\Delta_i,\ k+e_i'\delta_i) < J_i(P,k)$ which is -C2. ∎

We now extend Varaiya's bound $B(u) \leqq J^1(u) - J^1(u^*)$ to the discounted case. Recall the Hamiltonian

(5.7) $\quad H(u,\gamma) = Q(u)\gamma + k(u)$

Extending this to the discounted case, so that $H^{\rho}(u,\gamma) = Q^{\rho}(u)\gamma + k(u)$, we get an analogous result.

<u>Theorem 5</u>. Let γ be an arbitrary column vector, let $0 \leqq \rho \leqq 1$, and choose $u \in U$ such that

$$(5.8) \qquad H^{\rho}(u,\gamma) = \min_{u' \in U} H^{\rho}(u',\gamma)$$

Let $\underline{h} = \min_{i \in X} H_i^{\rho}(u,\gamma)$, $\overline{h} = \max_{i \in X} H_i^{\rho}(u,\gamma)$. Then

$$(5.9) \qquad \underline{h}\underline{1} + (1-\rho)\gamma \leqq J^{\rho *} \leqq J^{\rho}(u) \leqq \overline{h}\underline{1} + (1-\rho)\gamma$$

Furthermore, if $\underline{h} = \overline{h}$, then $\gamma = c^{\rho *} + a\underline{1}$ where a is a scalar, $J^{\rho}(u) = J^{\rho *}$, and $H^{\rho}(u,\gamma) = J^{1}(u^*)$. If also $\pi\gamma = 0$, then $\gamma = c^{\rho *}$.

<u>Proof</u>: Recall that $\Pi^{\rho}Q^{\rho} = -(1-\rho)I$. Thus

$$(5.10) \qquad \Pi^{\rho}(u)H^{\rho}(u) = \Pi^{\rho}Q^{\rho}\gamma + \Pi^{\rho}k = J^{\rho}(u) - (1-\rho)\gamma$$

and

$$(5.11) \qquad \Pi^{\rho *}(u^*)H^{\rho}(u^*) = J^{\rho *} - (1-\rho)$$

Since all of the elements of Π^{ρ} are non-negative and $\Pi^{\rho}\underline{1} = \underline{1}$, Eq. (5.9) follows.

Now suppose $\underline{h} = \overline{h}$. Since then $\underline{h}\underline{1} = \overline{h}\underline{1} = H$, $J^{\rho *} = J^{\rho}(u)$, $u = u^{\rho *}$, and

$$(5.12) \qquad H + (1-\rho)\gamma = J^{\rho}(u) = J^{1}(u) - (1-\rho)M^{\rho}c^{\rho}$$

so

$$(5.13) \qquad \gamma = \frac{1}{1-\rho}(J^{1} - H) + c^{\gamma}$$

and if $0 = \pi\gamma$ then $0 = \pi\gamma = \frac{1}{1-\rho}(J^1 - \underline{h})$ and $H = J^1$. ∎

6. References

1. Doob, J. L. (1953), Stochastic Processes, New York: Wiley.

2. Varaiya, P. P. (1978), "Optimal and Suboptimal Stationary Controls for Markov Chains," IEEE Trans. Auto. Control, Vol. AC-23.

3. Kushner, H. (1971), Introduction to Stochastic Control, New York: Holt, Rinehart and Winston, Inc.

4. Howard, R. A. (1960), Dynamic Programming and Markov Processes, Cambridge, Mass.: MIT Press.

5. Schweitzer, P. J. and A. Federgruen (1977), "The Asymptotic Behavior of Undiscounted Value Iteration in Markov Decision Problems," Math. of Operations Research, Vol. 2.

6. Odoni, A. (1969), "On Finding the Maximal Gain for Markov Decision Processes," Operations Research, Vol. 17, pp. 857-860.

7. Ross, S. M. (1970), Applied Probability Models with Optimization Applications, San Francisco: Holden Day.

8. Bertsekas, O. P. (1976), Dynamic Programming and Stochastic Control, New York: Academic Press.

1. Report No. NASA CR-3286	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Wheat Forecast Economics Effect Study		5. Report Date May 1980
		6. Performing Organization Code 903
7. Author(s) R.K. Mehra, R. Rouhani, S. Jones, and I. Schick		8. Performing Organization Report No.
9. Performing Organization Name and Address Scientific Systems, Inc. (S^2I) Suite 309-310 186 Alewife Brook Parkway Cambridge, MA 02138		10. Work Unit No.
		11. Contract or Grant No. NAS5-25463
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		13. Type of Report and Period Covered Contractor Report
		14. Sponsoring Agency Code

15. Supplementary Notes

Goddard Technical Monitors: David B. Wood and Ahmed Meer
Final Report

16. Abstract

This report describes the work performed by Scientific Systems, Inc. on the "Wheat Forecast Economic Effect" problem, that is, the evaluation of the value of improved information regarding the inventories, productions, exports and imports of crop on a worldwide basis.

The original model proposed by ECON is interpreted in a stochastic control setting and the underlying assumptions of the model are revealed. It is shown that for solving the stochastic optimization problem, the Markov programming approach is much more powerful and exact as compared to the dynamic programming-simulation approach of ECON. The convergence of a dual variable Markov programming algorithm is shown to be fast and efficient. A computer program for the general model of multi-country - multi-period is developed. As an example, the case of one country - two periods has been treated and the results are presented in detail. A comparison with ECON results reveals certain interesting aspects of the algorithms and the dependence of the value of information on the incremental cost function.

17. Key Words (Selected by Author(s))	18. Distribution Statement
Wheat Production Forecasts, Value of Information, economic models, stochastic optimization, Markov programming, control theory	Unclassified - Unlimited Subject Category 83

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
UNCLASSIFIED	UNCLASSIFIED	116	$6.50

*For sale by the National Technical Information Service, Springfield, Virginia 22161

End of Document