

ACCELERATED PROCEDURES FOR THE SOLUTION OF DISCRETE MARKOV
CONTROL PROBLEMS

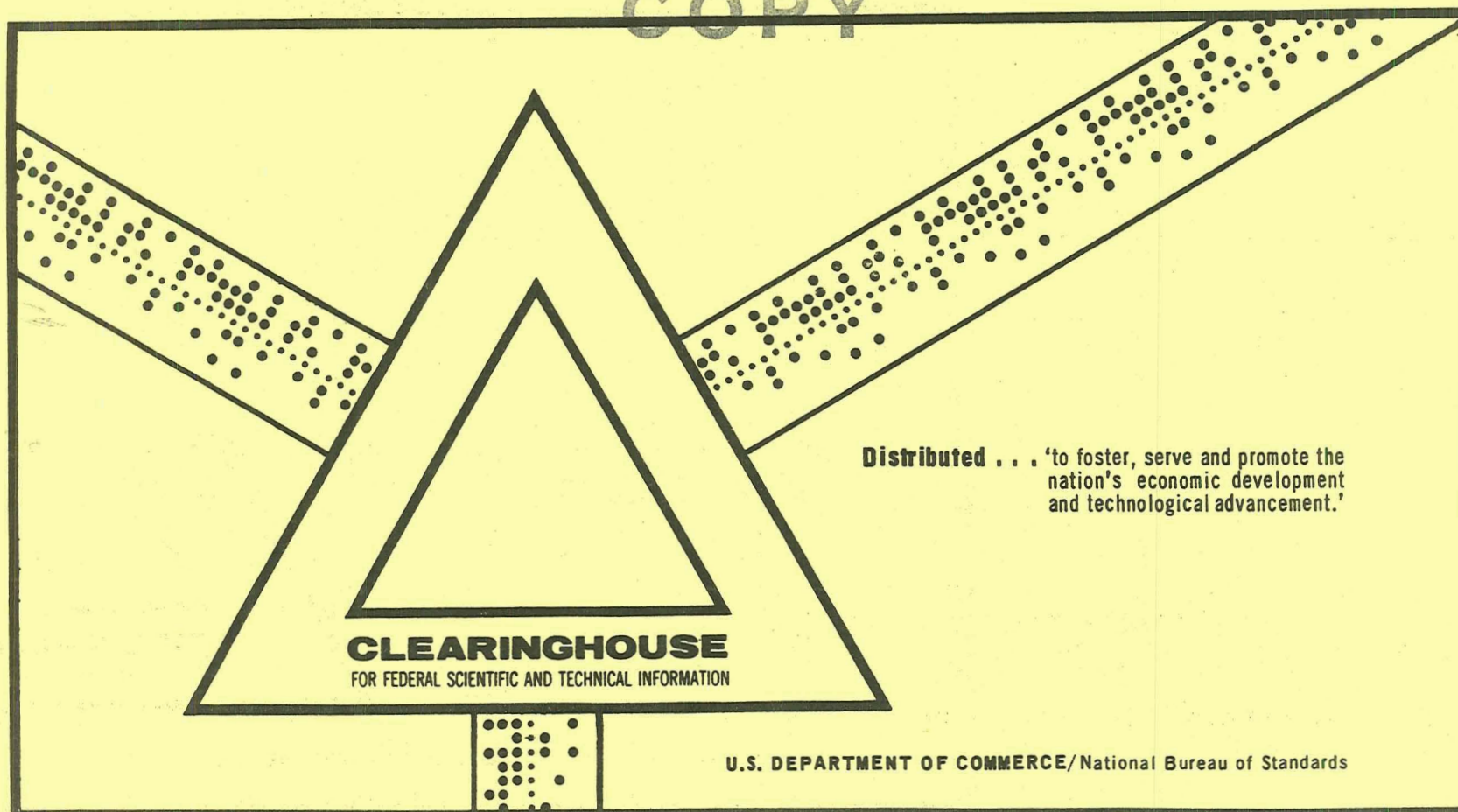
H. J. Kushner, et al

Brown University
Providence, Rhode Island

March 1970

N70-29658

CASE FILE
COPY



Distributed . . . 'to foster, serve and promote the
nation's economic development
and technological advancement.'

CLEARINGHOUSE
FOR FEDERAL SCIENTIFIC AND TECHNICAL INFORMATION

U.S. DEPARTMENT OF COMMERCE/National Bureau of Standards

AFOSR 70-1485 TR

AD 706896

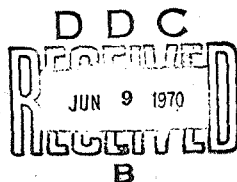
Accelerated Procedures for the Solution of
Discrete Markov Control Problems

by

H. J. Kushner*

and

A. J. Kleinman**



Reproduced by the
CLEARINGHOUSE
for Federal Scientific & Technical
Information Springfield Va. 22151

1. This document has been approved for public
release and sale; its distribution is unlimited.

* This research was supported in part by the National Science Foundation under Grant No. GK 2788, in part by the National Aeronautics and Space Administration under Grant No. NGL 40-002-019 and in part by the Air Force Office of Scientific Research under Grant No. AF-AFOSR 67-0692A., Brown Univ., Div. of Applied Math. Providence, Rhode Island 02912

** This research was supported by the National Science Foundation under Grant No. GK 2788.

25

Abstract

Accelerated procedures for computing optimal controls for a Markov chain model are discussed, and numerical results are presented. For the example, one of the methods described gave a 10 fold decrease in computation time over a more usual procedure of dynamic programming.

1. Introduction. Let $\{X_n, n = 0, 1, \dots\}$ be a controlled Markov chain on the state space $S = \{0, 1, \dots, N\}$, with transition probabilities $\{p_{ij}(u_i)\}$, where the control u_i takes values in a set U_i . State 0 is a target state, and it can be assumed to be absorbing ($p_{00}(u_0) \equiv 1$), since, for the problem of the paper, control terminates when the target set is reached. Let $u = (u_1, \dots, u_N)$ denote a control vector; i.e., if u is used and $X_n = i$, then $u(X_n) = u_i$. Define $k(0, u_0) \equiv 0$. The cost (1) of attaining the target is to be minimized over u ,

$$(1) \quad V(u; i) = E_i^u \sum_{n=0}^{\infty} k(X_n, u(X_n)),$$

where E_i^u is the expectation given that $X_0 = i$ and that u is always used.

Control problems on Markov chains arise in numerous ways in control and operations research. They also arise, when one seeks to solve by finite difference methods the non-linear partial differential cost equation which arises in the optimal control of continuous time diffusion models (see [1] for the construction). In the latter case, N is usually large, perhaps many thousand. This paper discusses accelerated and convenient computational methods for computing the optimal control $\bar{u} = (\bar{u}_1, \dots, \bar{u}_N)$ and minimal cost $V_i = \inf_u V(u; i)$.

Define the matrix $R(u) = \{p_{ij}(u_i); i, j = 1, \dots, N\}$, the transition matrix with state 0 deleted. Define the column vectors $V(u) = (V(u; 1), \dots, V(u; N))$, $V = (V_1, \dots, V_N)$, $K(u) = (k(1, u_1), \dots, k(N, u_N))$. Under

(C1) and (C2) below, the least cost and optimal control satisfy

$$(2) \quad V_i = \min_{u_i} \left[\sum_{j=1}^N p_{ij}(u_i) V_j + k(i, u_i) \right] \equiv \sum_{j=1}^N p_{ij}(\bar{u}_i) V_j + k(i, \bar{u}_i)$$

or, in vector form

$$(3) \quad V = \min_u [R(u)V + K(u)] = R(\bar{u})V + K(\bar{u}).$$

Also, for any fixed control vector u ,

$$(4) \quad V(u) = R(u)V(u) + K(u).$$

(C1) $k(i, \alpha)$ and $p_{ij}(\alpha)$ are continuous for each i, j and α in the compact set U_i .

(C2) There is an $\epsilon > 0$ so that the N step transition probabilities satisfy $p_{i0}^{(N)}(u) \geq \epsilon > 0$ for all $u \in U_1 \times \dots \times U_N$, and all i .

(C2) can be weakened, but it is quite often satisfied by problems, such as the numerical example, which are derived from diffusion models. It states that for any fixed control policy, the target can eventually be reached, since if $p_{i0}^{(n)}(u) > 0$ for some n , then (C2) holds for $n = N$.

In addition to the methods discussed here, there is the 'iteration (or approximation) in policy space' method. However, at least for the type of Markov problems which are derived as discretizations of continuous time problems, it seems to be inferior to the methods compared in this paper.

2. Iterative Methods for the Solution of (2)-(3).

The backward iterative procedure of dynamic programming (henceforth referred to as the Jacobi procedure, owing to the analogy to the Jacobi procedure for solving linear equations [2]) is

$$(5) \quad \begin{aligned} v^{n+1} &= \min_u [R(u)v^n + K(u)] \equiv R(u^n)v^n + K(u^n) \\ v_i^{n+1} &= \min_{u_i} \left[\sum_{j=1}^N p_{ij}(u_i) v_j^n + k(i, u_i) \right] \end{aligned}$$

(where v^n is the column vector $(v_1^n, \dots, v_N^n)$), and converges to V for any v^0 . In [1], it was shown that the Gauss-Seidel or successive substitution procedure

$$(6) \quad v_i^{n+1} = \min_u \left[\sum_{j=1}^{i-1} p_{ij}(u_i) v_j^{n+1} + \sum_{j=i}^N p_{ij}(u_i) v_j^n + k(i, u_i) \right]$$

converges at least as fast as $^+(5)$. Experience indicates that (6) is generally far superior to (5).

According to [1] (to which the reader is referred for more details),

(6) can be written as

$$(6a) \quad v_i^{n+1} = \min_{u_i} \sum_{j=1}^N q_{ij}(u) v_j^n + \tilde{k}(i, u)$$

or, in vector form,

$$(6b) \quad v^{n+1} = \min_u [Q(u)v^n + \tilde{K}(u)] \equiv Q(u^n)v^n + \tilde{K}(u^n)$$

$^+u^n$ is used to denote the minimizing control at the n^{th} iteration, for all procedures. The values may, of course, be different for different procedures, and different initial values v^0 .

where

$$\begin{aligned}
 q_{ij}(u) &= p_{ij}(u_i) + \sum_{k=1}^{i-1} p_{ik}(u_i) q_{kj}(u) & j \geq i \geq 1 \\
 &= \sum_{k=1}^{i-1} p_{ik}(u_i) q_{kj}(u) & 1 \leq j < i \\
 &= p_{ij}(u_i) & \text{for } i = 1 \\
 \tilde{k}(i, u) &= \sum_{j=1}^{i-1} p_{ij}(u_i) \tilde{k}(j, u) + k(i, u_i).
 \end{aligned}
 \tag{6c}$$

$Q(u)$ is the transition matrix for a Markov chain $\{Y_n\}$ (add $q_{00}(u) = 1$, $q_{i0}(u) = 1 - \sum_{j=1}^N q_{ij}(u)$). Indeed, $\{Y_n\}$ is absorbed at 0 faster than $\{X_n\}$, for each fixed control; this is the clue to the preferability of (6) over (5). In (6), the ordering of the states is important, in that a judicious ordering can greatly improve the convergence. See [1], [3], for a discussion of preferable orderings.

Most useful state orderings have the property (see Theorem 2 [1])

$$(C3) \quad \sum_{j=1}^N p_{ij}(u_i) \leq 1 - \epsilon' < 1$$

$$p_{ij}(u) > 0 \text{ for some } 1 \leq j(i, u) < i-1 \text{ (for } i > 1)$$

$$p_{ij}(u) > 0 \text{ for some } j(i, u) > 1.$$

(C3) guarantees (C2), and states that state 1 is always connected to state 0.

state i is always connected to some state below i , and to some state other than the target. (C3) is used ([1]) to show that $\{Y_n\}$ is absorbed at 0 'faster' than $\{X_n\}$. Note, for future reference, that under (C3), for any u , [1]

$$1 = q_{i0}(u) = \sum_{j=1}^N q_{ij}(u) < \sum_{j=1}^N p_{ij}(u_i) = 1 - p_{i0}(u_i), \quad i > 1$$

(7)

$$q_{i0}(u_i) = p_{i0}(u_i) > 0.$$

2. Accelerated Procedures

Next, the accelerated procedures are defined. Let Ω denote a diagonal matrix whose i^{th} diagonal entry is ω_i , $\omega_i \geq 1$. The accelerated Jacobi (AJ) procedure is

$$\begin{aligned}
 (8) \quad v^{n+1} &= \Omega \min_u [R(u)v^n + K(u)] + (I - \Omega)v^n \\
 &= \min_u [\tilde{R}(u)v^n + \Omega K(u)] \\
 \tilde{R}(u) &= \Omega R(u) + I - \Omega.
 \end{aligned}$$

The semi-accelerated Gauss-Seidel (SAGS) procedure is

$$\begin{aligned}
 (9) \quad \tilde{v}_i^n &= \min_{u_i} \left[\sum_{j=1}^{i-1} p_{ij}(u_i) \tilde{v}_j^n + \sum_{j=i}^N p_{ij}(u_i) v_j^n + k(i, u_i) \right] \\
 v_i^{n+1} &= \omega_i \tilde{v}_i^n + (1 - \omega_i) v_i^n.
 \end{aligned}$$

The accelerated Gauss-Seidel procedure (AGS) is

$$(10) \quad v_i^{n+1} = \omega_i \min_{u_i} \left[\sum_{j=1}^{i-1} p_{ij}(u_i) v_j^{n+1} + \sum_{j=i}^N p_{ij}(u_i) v_j^n + k(i, u_i) \right] + (1-\omega_i) v_i^n.$$

(9) is 'accelerated' after a complete cycle, while (10) is 'accelerated' at each step with a cycle. The term 'accelerated' is taken from analogous procedures for accelerating the convergence of iterative methods for linear equations [2], [4].

For judicious choices of parameters (see numerical results where $\Omega = \omega I$, for a scalar ω), (10) is preferable to (9) which is preferable to (5), (6) and (8). Some mild experimentation yields useful values of Ω . Usually, in practice the same problem is solved many times with varying costs $k(i, u_i)$. For a fixed Ω , the convergence seems independent of the costs; hence useful values of Ω on one run should be useful for other runs. Using (6c), (9) can be written as

$$(9a) \quad v^{n+1} = \Omega \min_u [Q(u) v^n + \tilde{K}(u)] + (I - \Omega) v^n \\ = \min_u [\Omega Q(u) + (I - \Omega)) v^n + \tilde{K}(u)].$$

Similarly, (10) can be written as

$$(10a) \quad v^{n+1} = \min_u [Q_\Omega(u) v^n + K_\Omega(u)] \\ = Q_\Omega(u^n) v^n + K_\Omega(u^n),$$

for some $Q_\Omega(u)$ and $K_\Omega(u)$; $Q_\Omega(\cdot)$ is not usually a transition matrix or even part of one.

Upon setting $u^n = \bar{u}$, (5), (6), (9), (10) hold for $v^{n+1} = v^n = v$, the optimal cost.

5. Acceleration in the Linear Problem. As useful background, consider a linear problem. Suppose that the spectral radius $\rho(A)$ of A , is less than one, and we seek to solve the vector system $X = AX + b$. The Jacobi iteration is $X^{n+1} = AX^n + b$, the Gauss-Seidel is

$$x_i^{n+1} = \sum_{j=1}^{i-1} a_{ij} x_j^{n+1} + \sum_{j=i}^N a_{ij} x_j^n + b_i,$$

or equivalently $X^{n+1} = \hat{A} X^n + \hat{b}$, where $\hat{A}, \hat{b}$ are calculated by the rules (6c). Then $\rho(\hat{A}) \leq \rho(A)$. The accelerated Jacobi procedure is

$$X^{n+1} = (\Omega A + (I - \Omega)) X^n + \Omega b = \tilde{A} X^n + \Omega b.$$

Let $\Omega = \omega I$. Clearly, the eigenvalues of $\tilde{A}$ are $\omega \lambda + (1 - \omega)$, where λ are the eigenvalues of A . For suitable ω , $\rho(\tilde{A}) < \rho(A)$ and similarly for the Gauss-Seidel procedure (see [2], [4]).

Unfortunately, the concept of spectral radius does not apply to the non-linear iterations, and the analysis of the non-linear procedures for the control problem uses a max norm, rather than a norm akin to the spectral radius, and the ω or Ω estimates are very conservative.

4. Analysis of the Nonlinear Iterations.

Theorem 1. Write (AJ, SAGS, AGS, resp.)

$$(11a) \quad V = [\Omega R(\bar{u}) + (I - \Omega)]V + \Omega K(\bar{u})$$

$$(11b) \quad V = [\Omega Q(\bar{u}) + (I - \Omega)]V + \Omega \tilde{K}(\bar{u})$$

$$(11c) \quad V = Q_{\Omega}(\bar{u})V + K_{\Omega}(\bar{u}).$$

If the spectral radius of a matrix multiplying V on a right side of (11) are less than unity, then

$$\limsup_n V_i^n \leq V_i$$

for that procedure.

Proof. Only (11b) will be treated. Write $\tilde{Q}(u) = \Omega Q(u) + (I - \Omega)$, and

$$V = \tilde{Q}(\bar{u})V + \Omega K(\bar{u}) \leq \tilde{Q}(u^n)V + \Omega K(u^n)$$

$$V^{n+1} = \tilde{Q}(u^n)V^n + \Omega K(u^n) \leq \tilde{Q}(\bar{u})V^n + \Omega K(\bar{u}).$$

Then

$$(12) \quad D \equiv \tilde{Q}(u^n) \dots \tilde{Q}(u^0)(V^0 - V) = \tilde{Q}(u^n)(V^n - V) \leq V^{n+1} - V \leq$$

$$\tilde{Q}(\bar{u})(V^n - V) \leq [\tilde{Q}(\bar{u})]^{n+1}(V^0 - V)$$

$$D = (D_1, \dots, D_n)$$

The theorem follows, since $\rho(\tilde{Q}(\bar{u})) < 1$ implies that $[\tilde{Q}(\bar{u})]^n \rightarrow 0$. Q.E.D.

Next convergence of V^n will be discussed via a method that is analogous to that used for a general non-linear problem in [4], Chapter 3.3. To analyze the lower bound in (12), we are unfortunately forced to use a max norm, and seek conditions under which $\max_j |D_j| \leq (1-\delta) \max_j |V_j^n - V_j|$ for some $\delta > 0$. Under (C1), (C2) or (C1), (C3), for $\Omega = I$, $\tilde{Q}(u^n) = Q(u^n)$ and $Q(u^n) \dots Q(u^0) \rightarrow 0$ as $n \rightarrow \infty$, and similarly for the Jacobi procedure.

Proceeding (see (9a)); and noting that the sums of the i^{th} row elements of $\Omega Q(u^n) + (I - \Omega)$ are less than $|1 - \omega_i(1 - q_{ii}(u^n))| + \omega_i \sum_{j \neq i} q_{ij}(u^n)$,

$$|D_j| \leq \{|1 - \omega_i(1 - q_{ii}(u^n))| + \omega_i \sum_{j \neq i} q_{ij}(u^n)\}_{\max_k} |V_k^n - V_k|.$$

Thus, there is convergence for any ω_i for which

$$(13) \quad 1 \leq \omega_i < \frac{2}{1 - q_{ii}(u^n) + \sum_{j \neq i} q_{ij}(u^n)}$$

for all u^n . But, by (C1), (C3) (see (7)), the suprema of the denominator of (13) is less than 2. Thus there is an interval of convergence which contains more than the point $\omega_i = 1$. Numerical experience suggests that this interval is very conservative. Thus, trial and error is required to determine the useful ω_i .

Similarly, for the AJ procedure, if

$$(14) \quad \sup_{u_i} [1 - p_{ii}(u_i) + \sum_{j \neq i} p_{ij}(u_i)] < 2.$$

The interval of convergence contains more than $\omega_i = 1$. We have not succeeded in treating AGS (except under the restrictive condition (14)), which, in

fact, computational experience indicates is the best of all the accelerated procedures. Theorem 1 suggests that the accelerated procedures may well converge for all Ω for which the spectral radii of the matrices in (11) are less than unity. Numerical experience also suggests this, since, for such Ω , no subsequence of the V_i^n can tend to $+\infty$. We have not been successful in exploiting this fact.

5. Numerical Comparisons.

A Continuous Time Problem. Consider the following continuous time problem.

Write

$$(15) \quad \dot{y} = u + \sigma \xi$$

where u is a control, $|u| \leq 1$ and ξ is 'white Gaussian noise'. A state variabilization of (15), in Itô stochastic differential equation form, is

$$(16) \quad \begin{aligned} dx_1 &= x_2 dt \\ dx_2 &= u dt + \sigma dz, \end{aligned}$$

where z_t is a Wiener process. The object of the control is to drive the state x_t to the internal line in the box in Figure 1, in minimum average time. The process is confined to the box, and, upon reaching the boundary, it is either reflected in, or moves along the boundary, each with a probability which is consistent with the dynamics internal to the box; there is also a cost (1.5 · time) associated with movement on the boundary. The boundary properties are not too important here, since we are interested here only in a comparison between different computational techniques.

An approximating problem on a Markov chain. Using the method of [1] for this problem, gives an approximating problem on a Markov chain on the nodes

of Figure 1. (See [1] for details). The target nodes correspond to state 0. The general form for the cost equations (for the approximating chain) are, of course,

$$V_i = \min_{|\alpha| \leq 1} \left[\sum_{j=1}^N p_{ij}(\alpha) V_j + k(i) \right],$$

for a suitable $p_{ij}(\cdot)$ and $k(\cdot)$, where i ranges over the nodes (other than the target) in Figure 1.

More particularly, let $x = (x_1, x_2)$ be a nodal point, and let e_i denote the unit vector in the i^{th} coordinate direction. Thus $x \pm e_i h$ is also a nodal point, for x internal in the box. For x internal in the box, and $\sigma^2 > h$ (write V_x for V_i , where x is the cartesian coordinate of node i in Fig. 1

$$(17) \quad \begin{aligned} V_x = \min_{|\alpha| \leq 1} & \left[V_{x+e_1 h} \begin{bmatrix} h|x_2|/Q \\ 0 \end{bmatrix} + V_{x-e_1 h} \begin{bmatrix} 0 \\ h|x_2|/Q \end{bmatrix} \right. \\ & \left. + V_{x+e_2 h} \left(\frac{\sigma^2}{2Q} + \frac{h\alpha}{2Q} \right) + V_{x-e_2 h} \left(\frac{\sigma^2}{2Q} - \frac{h\alpha}{2Q} \right) + \frac{h^2}{Q} \right] \end{aligned}$$

where $Q = \sigma^2 + h|x_2|$. The upper entries in the brackets pertain when $x_2 \geq 0$, the lower otherwise. A similar equation can be written for x on the outer boundary. The chain satisfies (C1), (C2) and, for a suitable ordering (C3). In particular the ordering in Figure 2 was used. Also, we set $\Omega = \alpha I$.

The numerical results are described by Figures 3 and 4, where the error

$$E = \sum_i |V_i^n - V_i|$$

is plotted vs. the iteration number n - on a semi-log scale. Figure 3

plots the results for AGS, GS, and J. AJ diverged for all ω which we tried. In other, simpler problems AJ did not diverge for some values of ω , but was not as good as the forms of the GS. For the AGS, the performance deteriorated rapidly as ω increased beyond 1.4, this being really the best value. The I/D notations give the number of iterations required per significant digit improvement in precision. Figure 4 plots the SAGS which, while better than J and G.S., is not as good as the best AGS. Note that the best ω for the SAGS is larger than the best ω for the AGS.

The AGS procedure is clearly superior. This was somewhat expected at the terminal stages, when, assuming that the control converged, the linear system behavior prevails. Even there it is not really obvious, since the ω for which the control converged, could be poor ω for the asymptotic (or linear) part of the iteration. Note, however, that the runs on Fig. 3 which are better asymptotically are also better initially - right from the beginning.

The best AGS gave a 10 fold reduction in computation over the J (which is the usual backward iteration procedure of dynamic programming) and a four fold improvement over the ordinary GS, the procedure of [1].

It is also apparent that the range of ω for which the result of Section (4) guarantees convergence of SAGS is grossly underestimated.

Let $\bar{u}$ denote the optimal control. Some interesting properties of the accelerated procedures can be seen by comparing their errors to the errors for the linear procedures (18) (AGS) and (19) (SAGS) for computing the least cost when $\bar{u}$ is given

$$(18) \quad V_i^{n+1} = \omega \left[\sum_{j=1}^{i-1} p_{ij}(\bar{u}_i) V_j^{n+1} + \sum_{j=i}^N p_{ij}(\bar{u}_i) V_j^n + k(i, \bar{u}_i) \right] + (1-\omega) V_i^n$$

$$(19) \quad \hat{V}_i^n = \left[\sum_{j=1}^{i-1} p_{ij}(\bar{u}_i) \hat{V}_j^n + \sum_{j=i}^N p_{ij}(\bar{u}_i) V_j^n + k(i, \bar{u}_i) \right]$$

$$V_i^{n+1} = \omega \hat{V}_i^n + (1-\omega) V_i^n.$$

Theorem 1 proved that for all ω for which (18) (or (19)) converges, the iterates (10) (or (9)) are bounded from above (for $\omega_1 = \omega$). A comparison of Figures 3 and 5 suggests that (10) converges for all ω for which (18) converges, and that the ω which are preferable for (18) are also preferable for (10) (for $\omega = \omega_1$). The asymptotic rates (slopes) in Figures 3 and 5 are the same, an expected result given that convergence occurs, for ultimately $u_1^n = \bar{u}_1$. However, the asymptotically preferable ω are also preferable for all iterations - even before the sequence u_1^n has converged.

Next, compare the SAGS iteration (19) in Figure 6, with (9) (Figure 4, $\omega_1 = \omega$). Up to $\omega = 1.6$, Figures 4 and 6 are similar. For $\omega = 1.8, 1.9$ the linear procedure (19) oscillates substantially before settling down. Observe, in Figure 7, the oscillation in the spectral radius in this region. This is typical for linear problems. See [2] for more information. This is also true for the non-linear problem (9). It can still be said that, for all runs taken, (9) converges for all ω ($\omega_1 = \omega$) for which (18) converged. For $\omega = 1.8, 1.9$, the iteration (9) took much longer for the control to converge, than for smaller values of ω . But the initial behavior of (19) for these ω is not as smooth as for smaller ω , and it is still possible that the ω which are best for (19) for moderate values of n , are best for (9) for moderate values of n , but a precise analysis has eluded us.

A graphic illustration of the advantage of acceleration is given in Figure 7, where the spectral radii of the linear operators in (18), (19) are plotted vs. ω .

Conclusions

Accelerated procedures for computing optimal controls for a Markov chain model are discussed, and numerical results are presented. For the example, one of the methods described gave a 10 fold decrease in computation time over a more usual procedure of dynamic programming.

References

1. H. Kushner, A. Kleinman, 'Numerical Methods for the Solution of the generate Nonlinear Elliptic Equations Arising in Optimal Stochastic trol Theory', IEEE Trans. on Automatic Control, AC-13, 1968, p. 344.
2. R. Varga, Matrix Iterative Analysis, Prentice-Hall, 1962.
3. H. Kushner, Introduction to Stochastic Control Theory, Holt, Rinehart and Winston. In press.
4. E. Isaacson, H. Keller, Analysis of Numerical Methods, Wiley, 1966.

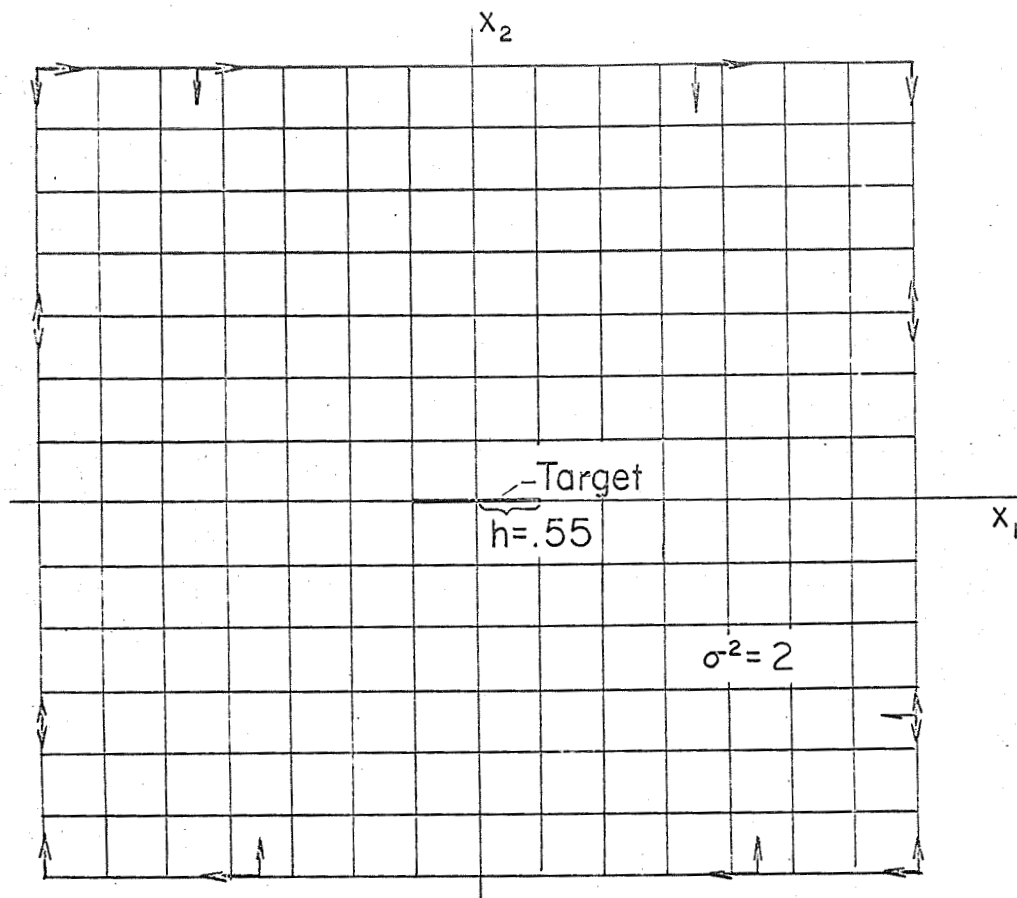


FIG. 1

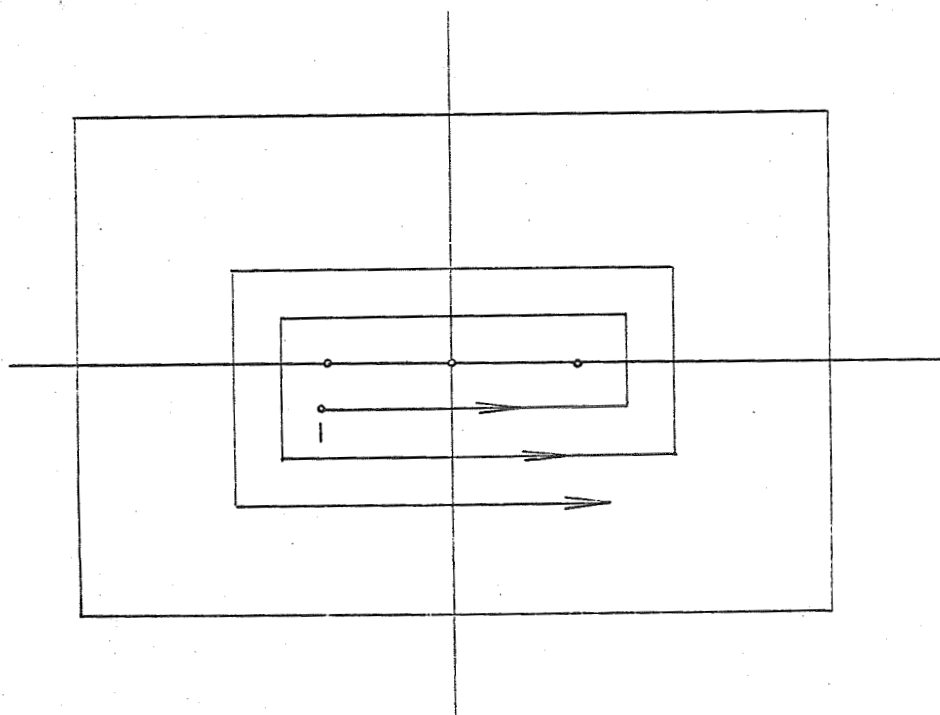


FIG. 2. THE ORDERING OF THE STATES

FIG. 3. COMPUTATION FOR THE
OPTIMAL CONTROL PROBLEM.

AGS and J

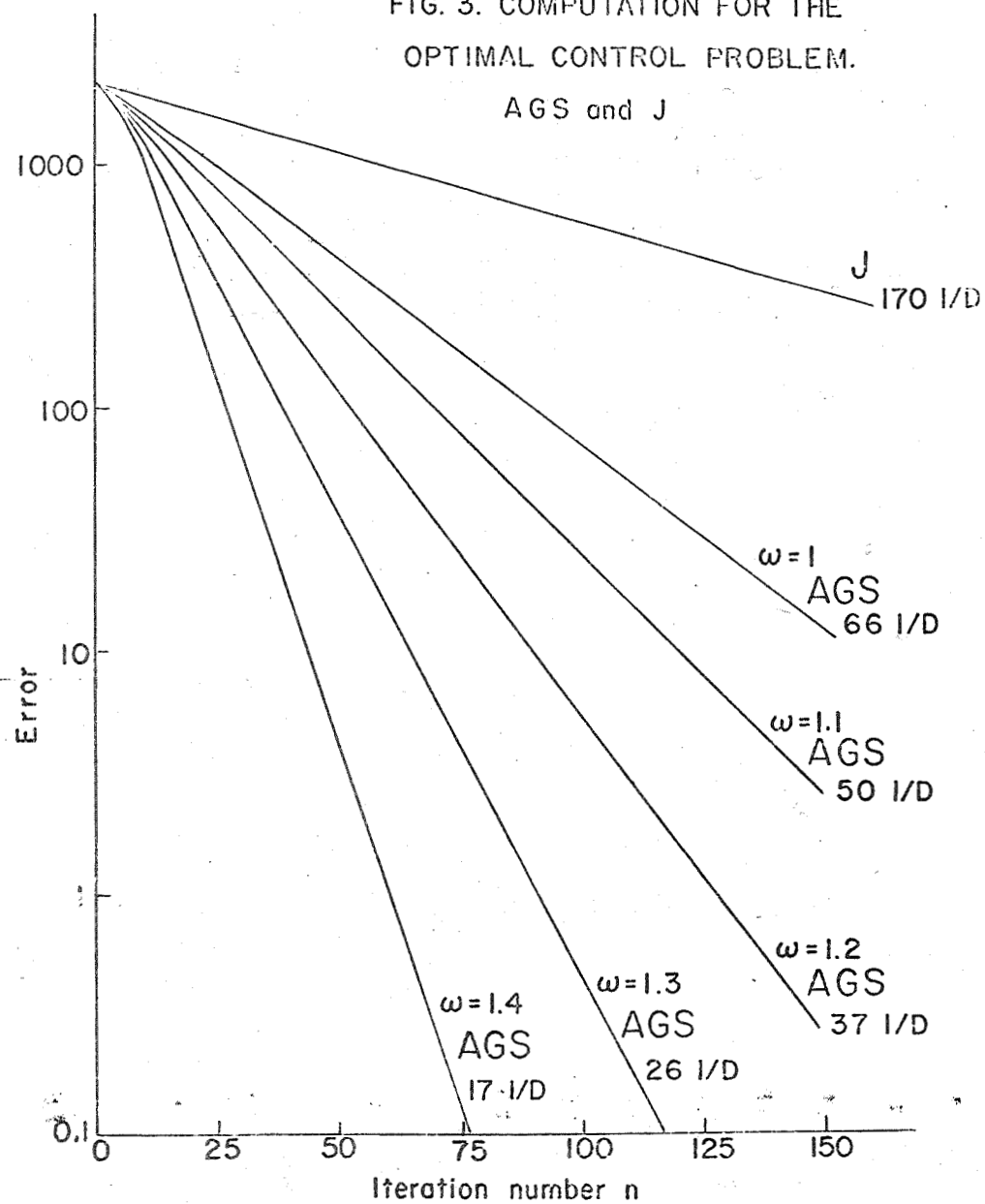


FIG. 4. COMPUTATION FOR THE
CONTROL PROBLEM. SAGS

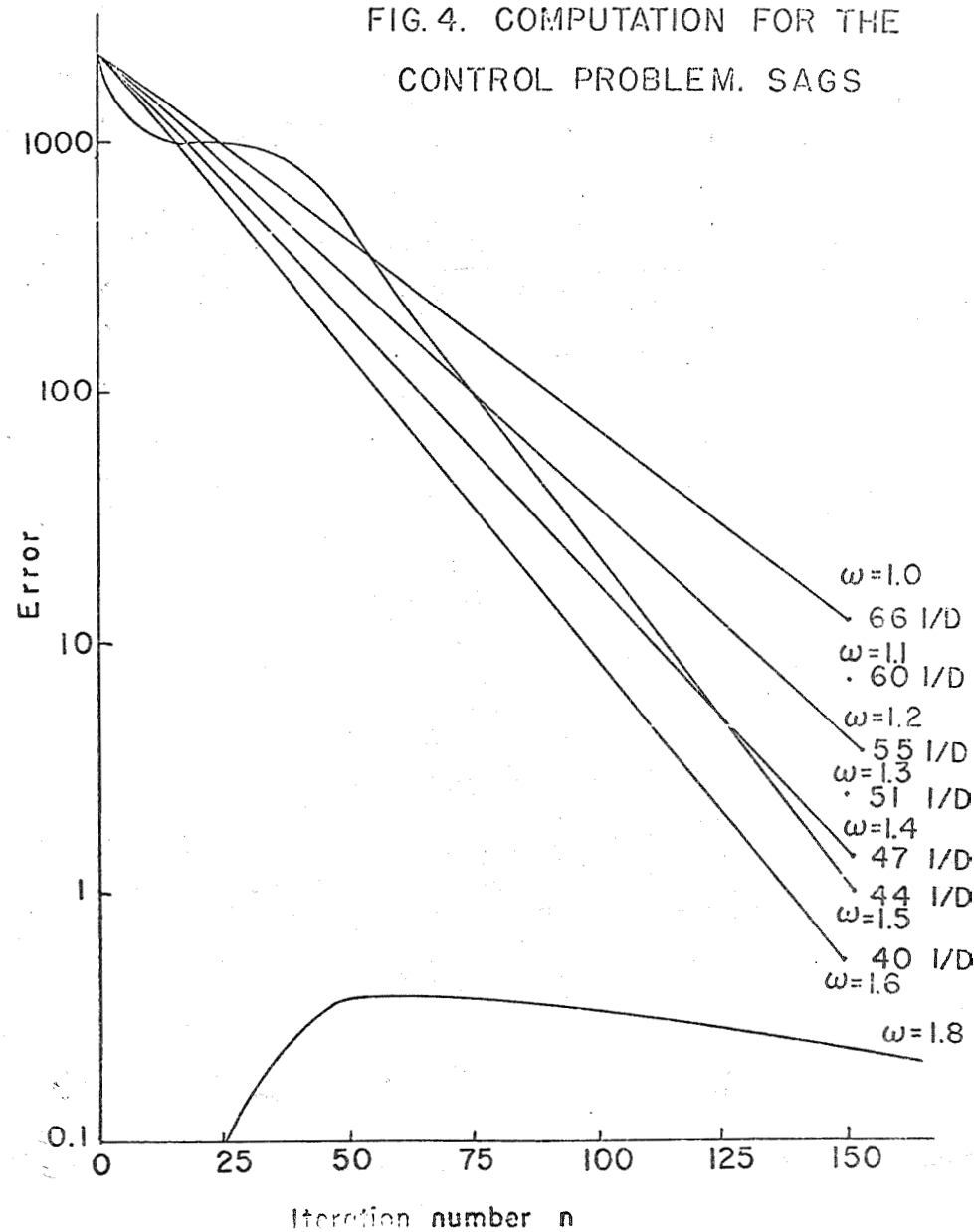


FIG. 5. COMPUTATION FOR
FIXED $R(\bar{u})$, $\bar{u}$ = OPTIMAL
CONTROL. AGS

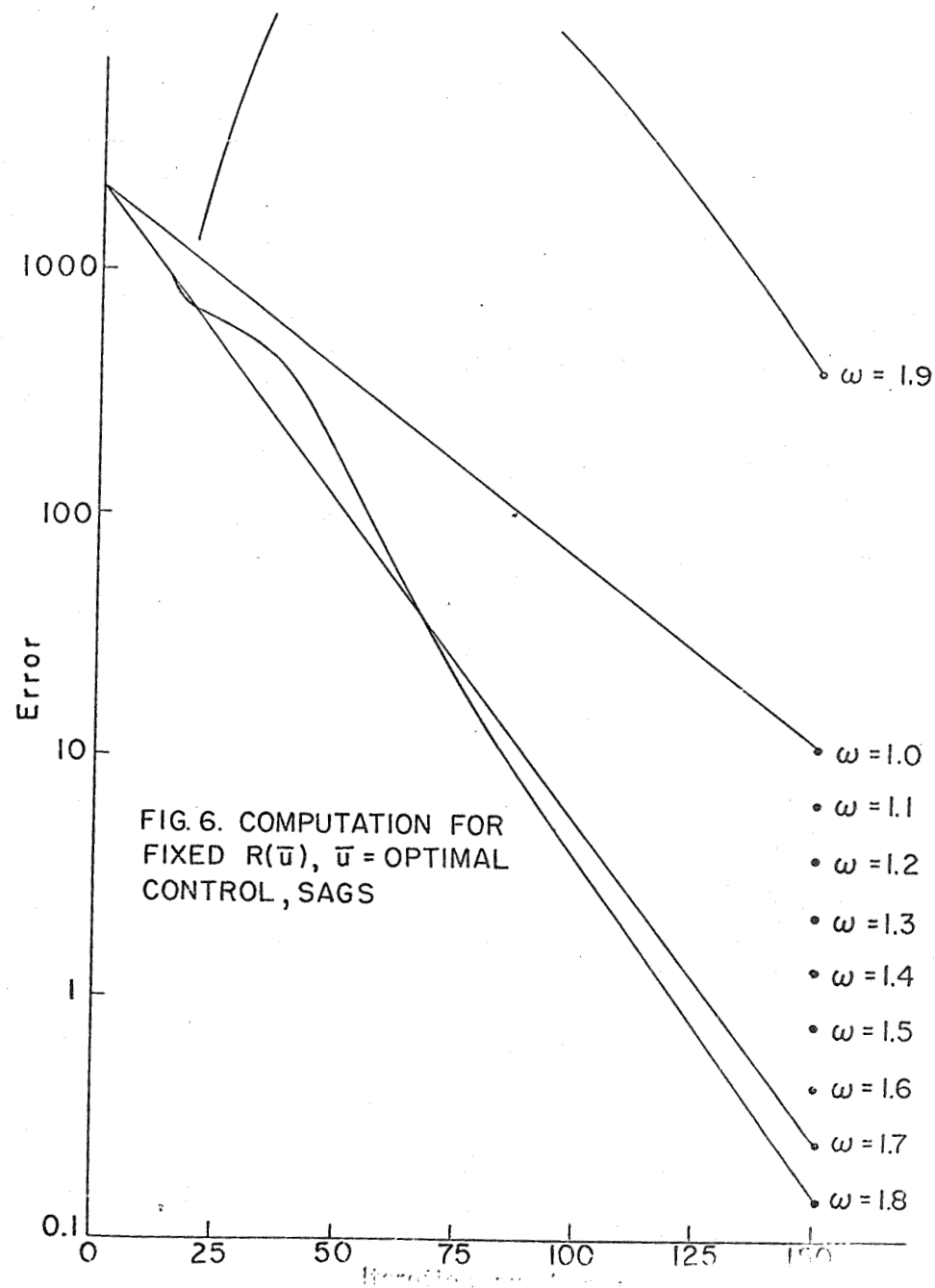
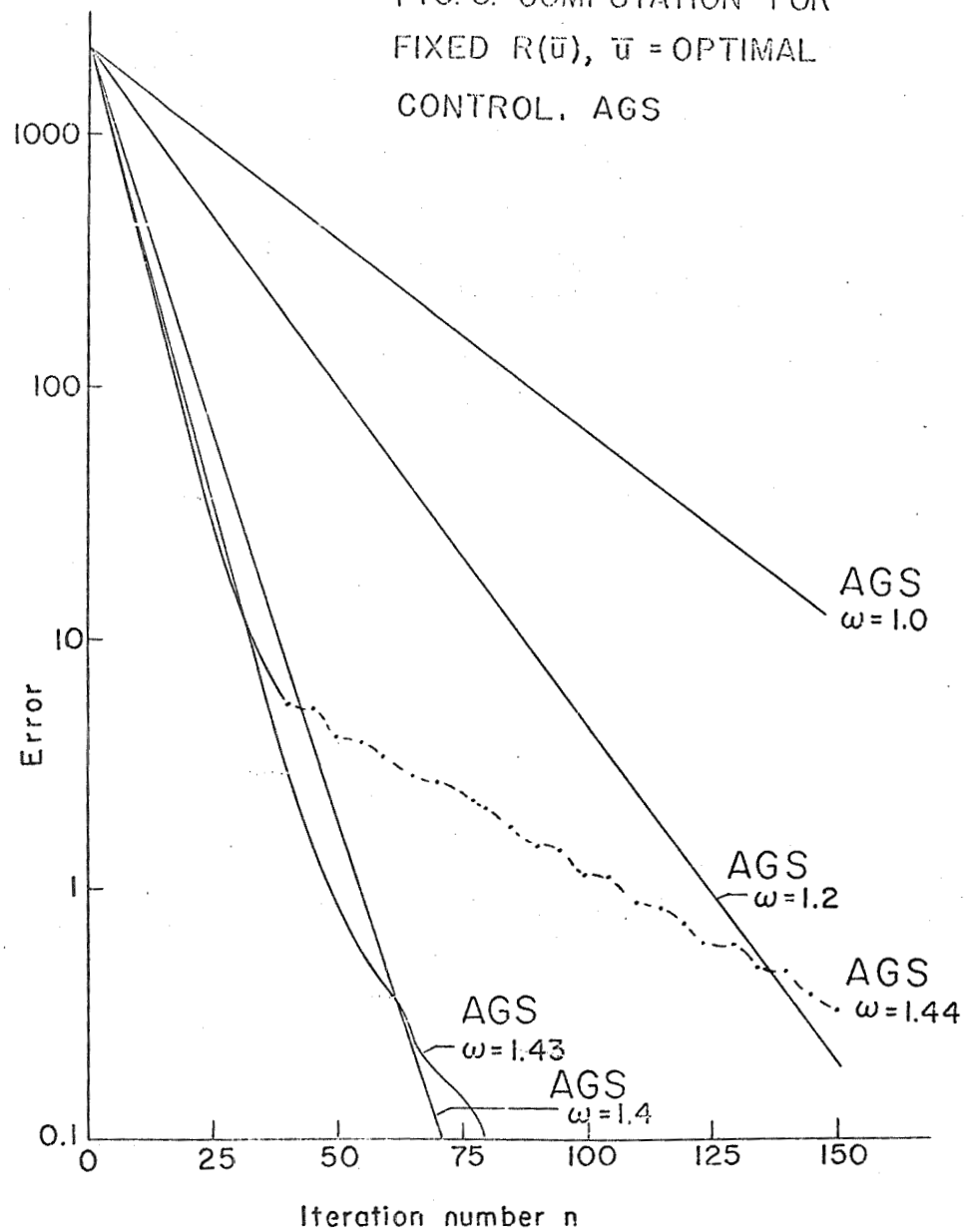
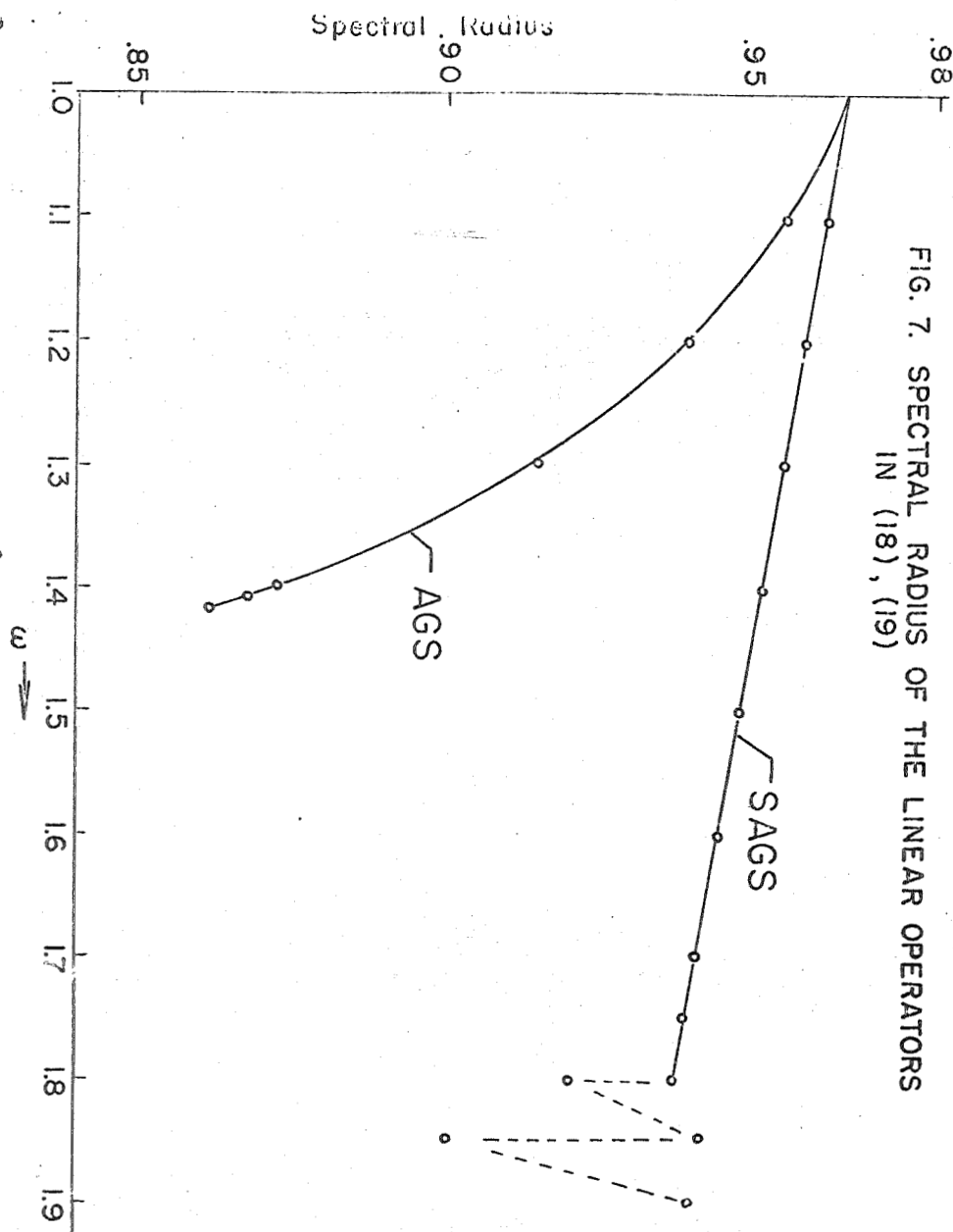


FIG. 6. COMPUTATION FOR
FIXED $R(\bar{u})$, $\bar{u}$ = OPTIMAL
CONTROL, SAGS



UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)

Brown University
Providence, Rhode Island 02912
Division of Applied Mathematics

2a. REPORT SECURITY CLASSIFICATION

UNCLASSIFIED

2b. GROUP

3. REPORT TITLE

ACCELERATED PROCEDURES FOR THE SOLUTION OF DISCRETE MARKOV CONTROL PROBLEMS

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Scientific Interim

5. AUTHOR(S) (Last name, first name, initial)

H. J. Kushner
A. J. Kleinman

6. REPORT DATE

May 1970

7a. TOTAL NO. OF PAGES

22

7b. NO. OF REFS

4

8a. CONTRACT OR GRANT NO. AF-AFOSR-693-67A

9a. ORIGINATOR'S REPORT NUMBER(S)

b. PROJECT NO. 9749-01

c. 61102F

d. 681304

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned
file report)

AFOSR 70-1485 TR

10. AVAILABILITY/LIMITATION NOTICES

1. This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES

TECH, OTHER

12. SPONSORING MILITARY ACTIVITY

Air Force Office of Scientific Research (SP)
1400 Wilson Boulevard
Arlington, Virginia 22209

13. ABSTRACT

Accelerated procedures for computing optimal controls for a Markov chain model are discussed, and numerical results are presented. For the example, one of the methods described gave a 10 fold decrease in computation time over a more usual procedure of dynamic programming.