

Board of Governors of the Federal Reserve System

International Finance Discussion Papers

Number 351

May 1989

AN ALGORITHM TO SOLVE DYNAMIC MODELS

Wilbur John Coleman II

Note: International Finance Discussion Papers are preliminary materials circulated to stimulate discussion and critical comment. References in publications to International Finance Discussion Papers (other than an acknowledgment that the writer has had access to unpublished material) should be cleared with the author or authors.

ABSTRACT

This paper presents an algorithm to solve recursive systems, formulated in discrete or continuous time, which have an endogenous state variable. The basis of the algorithm is a fixed point equation in the function from the state variables to the control variables.

An Algorithm to Solve Dynamic Models

Wilbur John Coleman II^{*}

Consider a system, evolving through time (or space), whose state at time t can be completely described by a finite dimensional vector s_t . In general, this state consists of variables which evolve exogenously, and those which evolve endogenously, that is, their evolution is determined such that certain equilibrium conditions are satisfied. In addition to this state vector, the system consists of endogenous variables which are not part of the state of the system, but which nevertheless influence the evolution of the endogenous state variables (e.g., control variables). The following pages present an algorithm which may solve this type of system. Three contexts will be developed to describe the algorithm: the stochastic discrete time system, the deterministic continuous time system, and the stochastic continuous time system. Each section will also contain a specific example, and Table 1, which is towards the end of this paper, summarizes the results of applying this algorithm to these examples.

There certainly exist many other methods which have been employed to solve these systems, some of which have clearly been quite successful for particular problems. Value function iteration for discrete time problems, for example, has proven to be quite robust, although its handling of equilibrium conditions (i.e., properties which the control functions must satisfy, but which are not directly part of the maximization problem) is less than satisfactory. For example, how does one go about solving for an equilibrium price function

^{*}The author is a staff economist in the International Finance Division of the Federal Reserve Board. This paper represents the views of the author and should not be interpreted as reflecting the views of the Board of Governors of the Federal Reserve System or other members of its staff.

(in a dynamic non–Pareto Optimal economy), as there is no natural iterative method which may be used to solve for these functions? The algorithm proposed here essentially integrates the equilibrium conditions with the optimality conditions on the original control functions, thus obtaining a new set of "controls" which has a likewise recursive specification, and thus suggests a natural iterative method for its solution. A benefit of this approach is that often the new set of endogenous functions is defined on a much lower dimensional state space (e.g., in a representative agent economy, individual and aggregate variables are equated in equilibrium). Even without the problem of equilibrium conditions, though, the algorithm seems to compare favorably with other methods (e.g., see the discussion in Taylor and Uhlig, 1989).

The approach in discrete time has apparently been successful for a wide variety of problems (e.g., the multisector optimal growth and monetary growth models, and various other models of asset prices and trade flows), so it seemed natural to attempt to duplicate this success for continuous time models. For these problems the finite difference methods for solving differential equations (e.g., Runge–Kutta) has proven to be quite robust, and in fact it is not clear, for these continuous time setups, if the algorithm proposed here is any better. It is nevertheless a general method to construct control functions (and not simply a particular path), and since it is a much different approach, which works for the specific examples described here, it may at least be useful when these other methods fail.

The Discrete Time Formulation

Let $z_t \in Z \subset \mathbb{R}^n$ denote the exogenous state variables. If z_t is Markovian, then its evolution can be written as

$$z_{t+1} = f(z_t, \epsilon_{t+1}),$$

where $\epsilon_{t+1} \in E$ consists of iid random variables drawn from a fixed cdf $\mu: E \rightarrow [0,1]$, and

$f: Z \times E \rightarrow Z$. Let $x_t \in X \subset \mathbb{R}^m$ denote the endogenous state variables and $u_t \in U \subset \mathbb{R}^p$ denote the "control" variables, which evolve according to

$$x_{t+1} = g(x_t, z_t, u_t), \quad g: X \times Z \times U \rightarrow X,$$

$$u_t = h(x_t, z_t), \quad h: X \times Z \rightarrow U.$$

The task at hand is to find an $h \in H$ which satisfies a set of equilibrium conditions.

For many problems these equilibrium conditions take one of two forms. Temporal conditions can often be written as

$$B(x_t, z_t, u_t) = 0, \quad B: X \times Z \times U \rightarrow \mathbb{R}^{p-q},$$

whereas intertemporal conditions can be written as

$$N(x_t, z_t, u_t) = \int M(x_{t+1}, z_{t+1}, u_{t+1}) d\mu(\epsilon_{t+1}), \quad N, M: X \times Z \times U \rightarrow \mathbb{R}^q.$$

Using these equations, and the laws of motion defined above, define the function $A(h)$, $A: H \rightarrow A(H)$, such that

$$(1) \quad B[x, z, (Ah)(x, z)] = 0,$$

$$(2) \quad N[x, z, (Ah)(x, z)] = \int M\{g[x, z, (Ah)(x, z)], f(z, \epsilon), h[g(x, z, (Ah)(x, z)), f(z, \epsilon)]\} d\mu(\epsilon).$$

The equilibrium control function is the fixed point $h = A(h)$. Note that $A(h)$ is an argument of h , that $A(h)$ is defined pointwise, and that A 's existence depends upon a

unique Brouwer fixed point. If A is continuous, isotone, strictly concave, and if it maps a nonempty, convex, and compact subset H into itself, then a unique fixed point $h^* \in H$ exists to which the sequence $A^n(h)$ converges uniformly. The existence of a fixed point follows from Schauder's fixed point theorem, and the uniqueness and convergence results can be found in Krasnoselskiĭ and Zabreĭko (1984).

There exists an intuitive reason for why this algorithm should work. Suppose that the time horizon for the control problem is T , and that the equilibrium conditions above are required to hold for each period up to $T-1$. Suppose, also, that the control function is fixed at h at T (i.e., $u_T = h(x_T, z_T)$). The "optimal" control at $T-1$ is then $A(h)$, and, given *this* control at $T-1$, the optimal control at $T-2$ is $A^2(h)$, and so on. The further back you go, the less dependent should be the control on h , and thus $A^n(h)$ should converge to the optimal control.¹ This argument is not meant to substitute for a rigorous proof, especially since many algorithms which should work simply don't. At this point, though, I think proofs are most usefully carried out for particular examples (e.g., see Coleman (1987) for an application to the optimal growth and monetary growth models).

A Program for a Digital Computer

To implement this algorithm on a digital computer, clearly some modifications need to be made. First of all, H must be approximated by a finite dimensional set. This can be achieved by imposing a grid on (x, z) (heavily concentrated, say, around an estimate of the mean of the steady state distribution) and only requiring that equations (1)–(2) hold at these grid points. Since, though, h must be evaluated at arbitrary points of its domain (see eq. 2), an interpolation routine must be used. I have had success with multi-linear interpolation (in logs), but for small dimensional problems a more sophisticated interpolater (e.g., splines) might be more efficient. With multi-linear interpolation, H is

thus approximated by a finite dimensional set of piecewise linear functions, and if H is compact, this approximation is uniform.²

Another problem is the integration in equation (2). In general, an adaptive quadrature rule will compute the integral to within some prespecified tolerance, but this often proves to be too slow. A much faster route is to rely on a fixed-point quadrature rule based on the weight function $d\mu$. If $d\mu$ takes on a standard form then existing routines may compute the points and weights (such as Hermite–Gauss quadrature for the Normal distribution, and clearly any distribution can be transformed into the Normal). In effect, this imposes a discrete probability model on ϵ (but not on z), and thus the integration is approximated by a fixed summation.

Some extensions may be possible to speed up the convergence of A^n . The generalization of Newton's method as developed by Kantorovitch (1948),

$$h_{n+1} = h_n - [A'(h_n)]^{-1}A(h_n),$$

which uses the Fréchet derivative A' , may speed things up, provided that this derivative (and its inverse) can be easily constructed. There may also exist a satisfactory extension of the multi-dimensional secant method (a discussion of the secant method can be found in Ortega and Rheinboldt, 1970) which can be applied to this problem.

Example 1

Consider the stochastic optimal growth model,³ where a control function h is sought which solves (some of the notation is switched)

$$\max_h \mathbb{E} \sum_{t=0}^{\infty} \beta^t u(c_t), \quad c_t = h(x_t, z_t),$$

subject to

$$x_{t+1} = f(x_t, z_t) - c_t$$

where the exogenous state variable evolves according to

$$z_{t+1} = \rho z_t + \epsilon_{t+1}, \quad \epsilon_{t+1} \text{ is iid Normal}(0, \sigma^2).$$

The state vector is then $s = (x, z)$, and eq. (2) becomes (eq. (1) does not apply)⁴

$$u'[(Ah)(x, z)] = \beta \int u' \{h[f(x, z) - (Ah)(x, z), \rho z + \epsilon]\} f' [f(x, z) - (Ah)(x, z), \rho z + \epsilon] d\mu(\epsilon).$$

The Continuous Time Formulation: The Deterministic Case

Suppose now that the control and state variables satisfy the ordinary differential equation system

$$\dot{x}(t) = g[x(t), u(t)], \quad g: X \times U \rightarrow \mathbb{R}^m,$$

$$\dot{u}(t) = f[x(t), u(t)], \quad f: X \times U \rightarrow \mathbb{R}^p.$$

Embedded in f are the equilibrium conditions which define the problem. Write the equilibrium control as a function of the state variables,

$$u(t) = h[x(t)], \quad h: X \rightarrow U,$$

where

$$\dot{u}(t) = h'[x(t)] \dot{x}(t) = h'[x(t)] g\{x(t), h[x(t)]\}.$$

Substitute in for $\dot{u}(t)$, and drop the time argument, to obtain

$$f[x, h(x)] = h'(x)g[x, h(x)],$$

which is a first order quasi-linear partial differential equation in h .

To obtain an approximate solution, note that

$$h'(x)g[x, h(x)] \cong \frac{h\{x + g[x, h(x)]d(x)\} - h(x)}{d(x)},$$

where $d(x)$ is some small increment step, say some fixed Δ . Define, then, the fixed point equation as^{5,6}

$$f[x, (Ah)(x)] = \frac{h\{x + g[x, (Ah)(x)]d(x)\} - (Ah)(x)}{d(x)}.$$

Boundary conditions can easily be imposed by requiring that the initial h satisfy these conditions, and never updating h at the boundary.

Unfortunately, one shortcoming of the above algorithm is that $A(h) \rightarrow h$ as $d \rightarrow 0$ whether or not h is a fixed point. To see this, write

$$f[x, (Ah)(x)] = \frac{h\{x + g[x, (Ah)(x)]d(x)\} - h(x)}{d(x)} + \frac{h(x) - (Ah)(x)}{d(x)}.$$

Clearly, to prevent the last term from blowing up, $(Ah)(x)$ must tend to h . (It is necessary that $(Ah)(x)$ determines the derivative directly by entering the last term on the right hand side, otherwise an incorrect guess of the derivative for h has too much

influence on the iterations.)⁷ To construct a sequence which rapidly converges to the fixed point, define a sequence of functions $d_n \rightarrow 0$ and corresponding fixed points $A(h_n) = h_n$, where each set of iterations to find h_n begins from the previous solution h_{n-1} .

Example 2

Consider the deterministic continuous time optimal growth model,⁸ where a control c is sought which solves

$$\max_c \int_0^{\infty} e^{-\rho t} u\{c(t)\} dt,$$

subject to

$$\dot{x}(t) = f[x(t)] - c(t).$$

The solution for the control can easily be determined to satisfy

$$\dot{c}(t) = - \frac{u' [c(t)]}{u'' [c(t)]} \{f' [x(t)] - \rho\}.$$

Define the control function as $c(t) = h[x(t)]$, and derive the ordinary differential equation

$$- \frac{u' [h(x)]}{u'' [h(x)]} [f'(x) - \rho] = h'(x)[f(x) - h(x)].$$

The functional equation in h then becomes⁹

$$- \frac{u' [(Ah)(x)]}{u'' [(Ah)(x)]} [f'(x) - \rho] = \frac{h\{x + [f(x) - (Ah)(x)]d(x)\} - (Ah)(x)}{d(x)},$$

where the boundary condition is $h(0) = 0$.

The Continuous Time Formulation: The Stochastic Case

Suppose now that the state evolves according to the stochastic differential equation

$$dx(t) = b[x(t), u(t)]dt + \sigma[x(t), u(t)]dz(t), \quad b: X \times U \rightarrow \mathbb{R}^m, \quad \sigma: X \times U \rightarrow \mathbb{R}^w,$$

where we wish to find an optimal control function

$$u(t) = h[x(t)], \quad h: X \rightarrow U,$$

such that certain optimality conditions are obtained. For many problems, the partial differential equation of dynamic programming can, after some effort, be reduced to the partial differential equation in the control

$$P\{x, h(x), h'(x), h''(x), \dots, h^{(r)}(x)\} = 0, \quad P: X \times U \times \mathbb{R}^{m(m+1)/2} \times \dots \rightarrow \mathbb{R}^p,$$

$(h^{(r)}(x))$, e.g, denotes all derivatives of order r) where boundary conditions are imposed on h . At this point, as expected, there is no clear cut way to proceed. If P corresponds to an analytic non-characteristic Cauchy initial value problem, then the nonlinear r^{th} order differential equation can, by differentiation, be reduced to a first order quasi-linear system which can be solved via the method developed above. Often (if not always), though, the problem at hand suggests a natural way to proceed in estimating the directional derivatives, and with these estimates the fixed point equation can be constructed analogously to the one constructed above. This is the route which I will follow in the example below.

Example 3

Consider the continuous time stochastic optimal growth model, where a control function h is sought which solves

$$\max_h \mathbb{E} \int_0^{\infty} e^{-\rho t} u[c(t)] dt, \quad c(t) = h[x(t), \theta(t)],$$

subject to

$$dx(t) = \{f[x(t), \theta(t)] - c(t)\} dt,$$

$$d\theta(t) = -\lambda \theta(t) dt + \sigma dz(t).$$

The differential equation then becomes (the derivation is in Appendix A)

$$\begin{aligned} & -\frac{u'[h(x, \theta)]}{u''[h(x, \theta)]} [f'(x, \theta) - \rho] = [f(x, \theta) - h(x, \theta)] h_x(x, \theta) \\ & - \lambda \theta h_\theta(x, \theta) + \frac{1}{2} \sigma^2 \frac{u'''[h(x, \theta)]}{u''[h(x, \theta)]} h_\theta^2(x, \theta) + \frac{1}{2} \sigma^2 h_{\theta\theta}(x, \theta), \end{aligned}$$

with the boundary conditions $h(0, \theta) = h_\theta(0, \theta) = 0$. This equation can be represented as the system (this does not correspond to the quasi-linear system obtained by differentiating P)

$$g(x, \theta) = \frac{1}{\sqrt{2}} \sigma h_\theta(x, \theta),$$

$$-\frac{u'[h(x, \theta)]}{u''[h(x, \theta)]} [f'(x, \theta) - \rho] + \lambda \theta h_\theta(x, \theta) - \frac{u'''[h(x, \theta)]}{u''[h(x, \theta)]} g^2(x, \theta) =$$

$$[f(x, \theta) - h(x, \theta)] h_x(x, \theta) + \frac{1}{\sqrt{2}} \sigma g_\theta(x, \theta).$$

The following approximations can then be used:

$$[f(x, \theta) - h(x, \theta)]h_x(x, \theta) = \frac{h\{x + [f(x, \theta) - h(x, \theta)]d(x), \theta\} - h(x, \theta)}{d(x)},$$

$$h_\theta(x, \theta) = \frac{h\{x, \theta + d(\theta)\} - h(x, \theta)}{d(\theta)},$$

and similarly for $g_\theta(x, \theta)$. Construction of the fixed point equation $A(h, g)$ is then straightforward. Note that, given the solution h above, Ito's Lemma provides us with

$$dc(t) = -\frac{u'[c(t)]}{u''[c(t)]}[f'[x(t), \theta(t)] - \rho]dt - \frac{1}{2}\sigma^2 \frac{u'''[c(t)]}{u''[c(t)]}h_\theta^2[x(t), \theta(t)]dt + h_\theta[x(t), \theta(t)]\sigma dz(t).$$

Some Demonstrations of the Algorithm's Performance

Table 1 demonstrates the algorithm's convergence for specific parameterizations of the above three examples. Appendix B contains the program to solve Example 1, the discrete time stochastic growth model. The entries in the table correspond to distances between successive iterates of the control function. As can be seen, the successive distances monotonically decrease, although the rate of decrease is much faster for the discrete time algorithm.

Concluding Remarks

I have had considerable success in applying the discrete time version of the algorithm described here to a wide diversity of problems. Given the prevalent use of discrete time models in economics and engineering, as well as in other disciplines, this algorithm could be useful in solving a wide variety of problems. The continuous time version, since it involves the additional approximation of a derivative, is probably less robust, but I have included a discussion of it here since I think it offers some new insight into solving these types of systems.

Table 1

$$\sup_s |\log h_{n+1}(s) - \log h_n(s)|$$

iter. no.	Ex. 1 ^a		Ex. 2 ^b		<i>h</i>	Ex. 3 ^c	<i>g</i>
1	0.7254	<i>dx</i> =1.0	0.4899	<i>dx</i> =.20	0.6135	<i>dθ</i> =.10	0.2106
2	0.3227		0.1937		0.2373		0.0857
3	0.1938		0.0964		0.1737		0.0331
4	0.1205		0.0508		0.1303		0.0245
5	0.0756		0.0303		0.1057		0.0210
6	0.0471		0.0191		0.0890		0.0197
7	0.0292		0.0118		0.0777		0.0160
8	0.0178		0.0075		0.0685		0.0144
9	0.0109		0.0048		0.0623		0.0121
10	0.0066		.		.		.
11	0.0039	<i>dx</i> =0.10	0.0052	<i>dx</i> =.10	0.0013	<i>dθ</i> =.08	0.0014
12	0.0024		0.0043		0.0012		0.0004
13	0.0014		0.0040		0.0011		0.0003
14	0.0008		0.0038		0.0011		0.0002
15	0.0005		0.0034		0.0011		0.0002
16	0.0003		0.0030		0.0010		0.0002
17	0.0002		.		.		.
18	0.0001	<i>dx</i> =0.05	0.0000	<i>dx</i> =.05	0.0007	<i>dθ</i> =.05	0.0002
19	0.0000		0.0000		0.0007		0.0001
20	0.0000		0.0000		0.0006		0.0001

a. $\beta = .95$, $u(c) = -1/c$, $f(x,z) = e^z \sqrt{x}$, $\rho = .7$, $\sigma = .1$, initial $h(x,z) = f(x,z)$.

b. $\rho = .05$, $u(c) = -1/c$, $f(x) = \sqrt{x} + x$, initial $h(x) = f(x) - x$. The .’s symbolize the remaining iterations for the indicated values of dx .

c. $\rho = .05$, $u(c) = -1/c$, $f(x,\theta) = e^\theta \sqrt{x} + x$, $\lambda = .3$, $\sigma = .1$, initial $h(x,\theta) = f(x,\theta) - x$, $g(x,\theta) = \frac{\sigma}{\sqrt{2}} h(x,\theta)$. The distances under column g correspond to levels, not logs.

In terms of economic applications the solution methodology presented here may enhance our ability to study sophisticated intertemporal general equilibrium models. Indeed, I think this technology can truly operationalize general equilibrium rational expectations models and put them on the same footing as the econometric models which preceded them, and which are still in wide use today.

APPENDIX A

This appendix derives the solution to the continuous time stochastic growth model. Let $v(t, x, \theta)$ denote the maximum value of the objective function starting at time t with $x(t)$ and $\theta(t)$. The solution solves the partial differential equation of dynamic programming

$$(A1) \quad -v_t = \max_h \{e^{-\rho t} u(h) + v_x(f-h) - v_\theta \lambda \theta + \frac{1}{2} \sigma^2 v_{\theta\theta}\}.$$

The first order condition for h is then

$$(A2) \quad 0 = e^{-\rho t} u'(h) - v_x.$$

Differentiate (A1) with respect to x to obtain

$$(A3) \quad -v_{xt} = e^{-\rho t} u'(h) h_x + v_{xx}(f-h) + v_x(f_x - h_x) - v_{x\theta} \lambda \theta + \frac{1}{2} \sigma^2 v_{x\theta\theta}$$

Using (A2), we can derive the following expressions for v_{xx} , $v_{x\theta}$, $v_{x\theta\theta}$, and v_{xt} :

$$0 = e^{-\rho t} u''(h) h_x - v_{xx}$$

$$0 = e^{-\rho t} u''(h) h_\theta - v_{x\theta}$$

$$0 = e^{-\rho t} u'''(h) h_\theta^2 + e^{-\rho t} u''(h) h_{\theta\theta} - v_{x\theta\theta}$$

$$0 = -\rho e^{-\rho t} u'(h) - v_{xt}$$

Substituting these derivatives of v into (A3) yields the desired differential equation in the control h .

To obtain the equation for $dc(t)$, use Ito's Lemma to obtain

$$dc(t) = [h_x(f-h) - h_\theta \lambda \theta + \frac{1}{2} \sigma^2 h_{\theta\theta}] dt + h_\theta \sigma dz(t).$$

Use, then, the differential equation in the control to rewrite the term in brackets as

$$h_x(f-h) - h_\theta \lambda \theta + \frac{1}{2} \sigma^2 h_{\theta\theta} = -\frac{u'(h)}{u''(h)}(f_x - \rho) - \frac{1}{2} \sigma^2 \frac{2u'''(h)}{u''(h)} h_\theta^2.$$

APPENDIX B

```

C
C*****
C PROGRAM TO SOLVE THE STOCHASTIC OPTIMAL GROWTH MODEL,
C WRITTEN BY WILBUR JOHN COLEMAN II (APRIL, 1989).
C*****
C
C THE PROBLEM IS TO FIND THE FIXED POINT OF A, WHERE HP = A(H) IS DEFINED BY
C UP[HP(S)] = BETA*E{UP[H(SP)]*FP(SP)}
C SP(1) = LOG[F(S) - HP(S)]
C SP(2) = RHO*S(1) + QX, QX DISTRIBUTED NORMAL(0,SIMGA**2),
C LET NH = # OF FUNCTIONS THAT ARE BEING SOUGHT; HERE NH = 1. LET (NX,NZ) =
C # OF (ENDOGENOUS,EXOGENOUS) STATE VARIABLES, AND NS = NX + NZ; HERE NX = NZ
C = 1. LET {NSD(I),I=1,NS} = # OF GRID POINTS FOR CORRESPONDING STATE
C VARIABLES {S(I),I=1,NS}, SO THERE ARE NSP = NSD(1)*NSD(2)* * *NSD(NS)
C POSSIBLE STATES INDEXED BY {N=1,NSP}. LET {SMIN(I),SMAX(I)} CORRESPOND TO
C THE (MINIMUM,MAXIMUM) VALUES OF THE GRID POINTS FOR S(I). PLACE EQUIDISTANT
C GRID POINTS FOR S(1) INTO {SM(I),I=1,NSD(1)}, THOSE FOR S(2) INTO
C {SM(I),I=NSD(1)+1,NSD(1)+NSD(2)}, AND SO ON. FOR EACH INDEX N THERE
C CORRESPONDS A POINTER K INTO SM SUCH THAT S(1) = SM(K(1)), S(2) = SM(NSD(1)
C + K(2)), AND SO ON, AND THERE CORRESPONDS A VALUE OF H AT EACH SUCH S WHICH
C IS STORED IN {HM(N,J),N=1,NSP} FOR EACH FUNCTION J=1,NH, AND SIMILARLY FOR
C HPM. DEFINE H AT AN ARBITRARY STATE BY MULTILINEAR INTERPOLATION BASED ON
C HM. THE ALGORITHM THUS COMPUTES HPM FOR A GIVEN H, COMPUTES THE DISTANCE
C BETWEEN HPM AND HM, PLACES HPM INTO HM, AND ITERATES UNTIL CONVERGENCE IS
C OBTAINED.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION SM(1000),HM(10000,1),HPM(10000,1),EPS(2),NSD(10)
      DIMENSION SUPH(10)
      COMMON /COMN/SM,HM,HPM,EPS,BETA,NX,NZ,NS,NSD,NSP,NH,MAXIT
      WRITE(*,'(A35)') ' STOCHASTIC OPTIMAL GROWTH MODEL'
      CALL INIT
      DO 300 NUMIT = 1, MAXIT
        CALL A
        DO 100 I = 1, NH
          SUPH(I) = SUP(HM(1,I),HPM(1,I),NSP)
          WRITE(6,'(A10,I5,A10,10F10.4)')
          * 'A ITER. #',NUMIT,'NORM = ',(SUPH(I),I=1,NH)
          DO 200 I = 1, NH
            IF (SUPH(I) .LE. EPS(1)) GOTO 400
          200 CONTINUE
          WRITE(*,'(A40)') ' MAXIMUM NO. OF ITERATIONS EXCEEDED'
          400 CONTINUE
          DO 500 I = 1, NH
            WRITE(11) (HM(N,I),N=1,NSP)
          500 STOP
        END
C
C*****
C SUBROUTINE A
C INPUT: FUNCTION H, ETC. OUTPUT: HPM
C*****
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION SM(1000),HM(10000,1),HPM(10000,1),EPS(2),NSD(10)
      DIMENSION K(10),S(10),HGUESS(10),HROOT(10),NSMPRD(10),NSMSUM(10)
      COMMON /COMN/SM,HM,HPM,EPS,BETA,NX,NZ,NS,NSD,NSP,NH,MAXIT
      COMMON /CNM/NSMPRD,NSMSUM

```

```

COMMON /CZ/S
EXTERNAL SC
DO 100 I = 1, NS
  K(I) = 1
100  S(I) = SM(1 + NSMSUM(I))
  DO 200 N = 1, NSP
    HGUESS(1) = DLOG(HM(N,1)/(F(S) - HM(N,1)))
    CALL DNEQNF(SC, EPS(2), NH, MAXIT, HGUESS, HROOT, ZNORM)
C    DNEQNF IS AN IMSL ROUTINE TO SOLVE NH NONLINEAR EQS. IN NH UNKNOWNNS.
C    INPUT: SUBROUTINE SC(HGUESS, ZVEC, NH) WHICH EVALUATES ZVEC AT HGUESS,
C    CONV. CRITERION EPS(2), MAX. # OF ITER. MAXIT, GUESS OF ROOTS HGUESS.
C    OUTPUT: SOLUTION IN HROOT SUCH THAT ZVEC = 0, SUM OF SQUARES IN ZNORM.
    HPM(N,1) = F(S)/(1.0D0 + DEXP(-HROOT(1)))
    CALL UPDS(K, S)
200  CONTINUE
  RETURN
END

C
C*****
SUBROUTINE SC(HGUESS, ZVEC, NH)
C  INPUT: GUESS OF TRANSFORMED HP(S) AS HGUESS, DIMENSION NH, STATE S.
C  OUPUT: ZVEC = RHS - LHS OF EACH FOC-EQUIL. COND.
C*****
IMPLICIT REAL*8 (A-H, O-Z)
DIMENSION HGUESS(10), ZVEC(10), S(10), HP(10)
COMMON /CZ/S
HP(1) = F(S)/(1.0D0 + DEXP(-HGUESS(1)))
CALL Z(S, HP, ZVEC)
RETURN
END

C
C*****
SUBROUTINE Z(S, HP, ZVEC)
C  INPUT: STATE S, GUESS HP. OUTPUT: ZVEC.
C*****
IMPLICIT REAL*8 (A-H, O-Z)
DIMENSION SM(1000), HM(10000, 1), HPM(10000, 1), EPS(2), NSD(10)
DIMENSION S(10), SP(10), HP(10), ZVEC(10), QX(100), QW(100)
COMMON /COMN/SM, HM, HPM, EPS, BETA, NX, NZ, NS, NSD, NSP, NH, MAXIT
COMMON /CPDF/SIGMA, RHO, QX, QW, NGAUSS
SP(1) = DLOG(F(S) - HP(1)) ! SP IS NEXT PERIOD'S STATE
W = 0.0D0
DO 100 I = 1, NGAUSS ! INTEGRATE TO COMPUTE EXPECTED VALUE
  SP(2) = RHO*S(2) + QX(I)
  W = W + UP(H(SP, 1))*FP(SP)*QW(I)
100  CONTINUE
ZVEC(1) = BETA*W - UP(HP(1)) ! RHS - LHS
RETURN
END

C
C*****
DOUBLE PRECISION FUNCTION UP(C)
C  INPUT: C. OUTPUT: MARGINAL UTILITY.
C*****
IMPLICIT REAL*8 (A-H, O-Z)
COMMON /UTIL/TAU
UP = 1.0D0/C**TAU
RETURN
END

```

```

C
C*****
DOUBLE PRECISION FUNCTION F(S)
C INPUT STATE S. OUTPUT: PRODUCTION.
C*****
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION S(10)
COMMON /PROD/ALPHA,DELTA
F = DEXP(S(2))*DEXP(S(1))*ALPHA + DELTA*DEXP(S(1))
RETURN
END

C
C*****
DOUBLE PRECISION FUNCTION FP(S)
C INPUT: STATE S. OUTPUT: MARGINAL PRODUCTIVITY.
C*****
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION S(10)
COMMON /PROD/ALPHA,DELTA
FP = ALPHA*DEXP(S(2))*DEXP(S(1))*(ALPHA - 1.0D0) + DELTA
RETURN
END

C
C*****
DOUBLE PRECISION FUNCTION H(S,I)
C INPUT: STATE S, FUNCTION # I. OUTPUT: INTERPOLATED FUNCTION VALUE.
C*****
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION SM(1000),HM(10000,1),HPM(10000,1),EPS(2),NSD(10)
DIMENSION S(1)
COMMON /COMN/SM,HM,HPM,EPS,BETA,NX,NZ,NS,NSD,NSP,NH,MAXIT
H = GRID(S,SM,HM(1,I),NS,NSD)
RETURN
END

C
C*****
DOUBLE PRECISION FUNCTION SUP(F0,F1,NSP)
C INPUT: VECTORS F0, F1. OUTPUT: SUP DISTANCE, AND F1 => F0.
C*****
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION F0(1),F1(1)
SUP = 0.0D0
DO 100 N = 1, NSP
SUP = DMAX1(SUP, DABS(DLOG(F1(N)) - DLOG(F0(N))))
100 F0(N) = F1(N)
RETURN
END

C
C*****
SUBROUTINE UPDS(K,S)
C INPUT: POINTER K. OUTPUT: NEXT K, CORRESPONDING STATE S.
C*****
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION SM(1000),HM(10000,1),HPM(10000,1),EPS(2),NSD(10)
DIMENSION K(10),S(10),NSMPRD(10),NSMSUM(10)
COMMON /COMN/SM,HM,HPM,EPS,BETA,NX,NZ,NS,NSD,NSP,NH,MAXIT
COMMON /CNSM/NSMPRD,NSMSUM
K(1) = K(1) + 1
DO 100 I = 1, NS - 1

```

```

      IF (K(I) .LE. NSD(I)) GOTO 200
      K(I) = 1
      K(I+1) = K(I+1) + 1
100   CONTINUE
200   CONTINUE
      DO 300 I = 1, NS
300   S(I) = SM(K(I) + NSMSUM(I))
      RETURN
      END
C
C*****
      SUBROUTINE INIT
C      INITIALIZE PARAMETERS, ETC.
C*****
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION SM(1000),HM(10000,1),HPM(10000,1),EPS(2),NSD(10)
      DIMENSION QX(100),QW(100),NSMPRD(10),NSMSUM(10)
      DIMENSION K(10),S(10),SMIN(10),SMAX(10)
      COMMON /COMN/SM,HM,HPM,EPS,BETA,NX,NZ,NS,NSD,NSP,NH,MAXIT
      COMMON /UTIL/TAU
      COMMON /PROD/ALPHA,DELTA
      COMMON /CPDF/SIGMA,RHO,QX,QW,NGAUSS
      COMMON /CNSM/NSMPRD,NSMSUM
C
      READ(10,'(10X,I10)') NH,NX,NZ
      NS = NX + NZ
      READ(10,'(10X,F10.4)') BETA,TAU,RHO,SIGMA,ALPHA,DELTA,EPS,
* (SMIN(I),SMAX(I),I=1,NS)
      READ(10,'(10X,I10)') (NSD(I),I=1,NS),NGAUSS,MAXIT
C
      NSMPRD(1) = 1
      NSMSUM(1) = 0
      DO 100 I = 2, NS
      NSMPRD(I) = NSMPRD(I - 1)*NSD(I - 1)
      NSMSUM(I) = NSMSUM(I - 1) + NSD(I - 1)
100   CONTINUE
      NSP = NSMPRD(NS)*NSD(NS)
C
      SCALE = 1.0D0/DSQRT(DCONST('PI'))
      CALL DGQRUL(NGAUSS,4,A1,B1,0,QXFIX,QX,QW)
C      DGQRUL IS AN IMSL ROUTINE TO COMPUTE HERMITE-GAUSS QUADRATURE POINTS
C      AND WEIGHTS FOR A NORMAL(0,1) DISTRIBUTION.
C      INPUT: # OF QUAD. POINTS NGAUSS. OUTPUT: POINTS QX, UNSCALED WEIGHTS QW.
      DO 200 I = 1, NGAUSS
      QX(I) = SIGMA*QX(I)
200   QW(I) = SCALE*QW(I)
C
      N = 0
      DO 400 J = 1, NS
      N = N + 1
      SM(N) = SMIN(J)
      DO 300 I = 2, NSD(J)
      N = N + 1
      SM(N) = SMIN(J)
*      + (DFLOAT(I - 1)/DFLOAT(NSD(J) - 1))*(SMAX(J) - SMIN(J))
300   CONTINUE
400   CONTINUE
C
      DAPC = 1.0D0 - ALPHA*BETA/(1.0D0 - (1.0D0 - ALPHA)*DELTA*BETA)
! INITIALIZE HM

```

```

DO 500 I = 1, NS
  K(I) = 1
500  S(I) = SM(1 + NSMSUM(I))
  DO 600 N = 1, NSP
    HM(N,1) = DAPC*F(S)
    CALL UPDS(K,S)
600  CONTINUE
  RETURN
  END

C
C*****
DOUBLE PRECISION FUNCTION GRID(S,SM,GM,NS,NSD)
C  INPUT: STATE S, STATE GRID SM, FUNCTION VALUES GM, DIMENSIONS NS, NSD.
C  OUTPUT: MULTILINEAR INTERPOLATION OF FUNCTION AT S BASED ON GM AT SM.
C*****
  IMPLICIT REAL*8 (A-H,O-Z)
  DIMENSION S(1),SM(1),GM(1),NSD(1),KS(10),KS2(10),T(10),IP(10)
  DIMENSION NSMPRD(10),NSMSUM(10)
  COMMON /CNSM/NSMPRD,NSMSUM

C
DO 100 I = 1, NS                                ! FIND SURROUNDING GRID POINTS
  J = 1 + NSMSUM(I)
  CALL LOCATE(SM(J),NSD(I),S(I),KS(I))
  IF (NSD(I) .EQ. 1) THEN
    T(I) = 1.000
  ELSE                                           ! COMPUTE POSITION OF STATE RELATIVE TO GRID POINTS
    T(I) = (SM(J+KS(I)) - S(I))/(SM(J+KS(I)) - SM(J+KS(I)-1))
  ENDIF
  IP(I) = 0
100  CONTINUE
C
GRID = 0.000
DO 400 N = 1, 2**NS                                ! LOOP OVER ALL CORNERS OF NS DIMENSIONAL CUBE
  TP = 1.000
  DO 200 J = 1, NS                                ! FIND WEIGHT FOR A CORNER
    IF (IP(J) .EQ. 0) THEN
      TP = TP*T(J)
    ELSE
      TP = TP*(1.000 - T(J))
    ENDIF
    IF (NSD(J) .EQ. 1) THEN                       ! PLACE CORNER INTO KS2
      KS2(J) = KS(J)
    ELSE
      KS2(J) = KS(J) + IP(J)
    ENDIF
200  CONTINUE                                     ! ADD WEIGHT TIMES VALUE OF CORNER
  GRID = GRID + TP*DLOG(GM(KMAP(KS2,NS)))

C
  IP(1) = IP(1) + 1                                ! UPDATE CORNER
  DO 300 J = 1, NS - 1
    IF (IP(J) .LE. 1) GOTO 400
    IP(J) = 0
300  IP(J + 1) = IP(J + 1) + 1
400  CONTINUE
  GRID = DEXP(GRID)
  RETURN
  END

C
C*****

```

```

FUNCTION KMAP(K,NS)
C   INPUT: POINTER K, DIMENSION NS.  OUTPUT: INDEX N AS KMAP.
C*****
  IMPLICIT REAL*8 (A-H,O-Z)
  DIMENSION K(1),NSMPRD(10),NSMSUM(10)
  COMMON /CNSM/NSMPRD,NSMSUM
  KMAP = K(1)
  DO 100 I = 2, NS
100  KMAP = KMAP + NSMPRD(I)*(K(I) - 1)
  RETURN
  END

C
C*****
  SUBROUTINE LOCATE(XV,N,X,J)
C   INPUT: INCREASING VECTOR XV, DIMENSION N, VALUE X.
C   OUTPUT: LARGEST J SUCH THAT XV(J) < X, WHERE 0 < J < N.
C*****
  IMPLICIT REAL*8 (A-H,O-Z)
  DIMENSION XV(1)
  JL = 1
  JU = N
  100 CONTINUE
  IF (JU - JL .GT. 1) THEN
    JM = (JU + JL)/2
    IF (X .GT. XV(JM)) THEN
      JL = JM
    ELSE
      JU = JM
    ENDIF
  GOTO 100
  ENDIF
  J = JL
  RETURN
  END

```

SAMPLE DATA FILE

```

NH          1
NX          1
NZ          1
BETA       0.95
TAU        2.00
RHO        0.70
SIGMA      0.10
ALPHA      0.50
DELTA      0.00
EPS1       1.0D-4
EPS2       1.0D-5
XMIN       -5.00
XMAX        0.00
ZMIN       -0.30
ZMAX        0.30
NSD1        20
NSD2        10
NGAUSS      5
MAXIT       1000

```

ENDNOTES

1. Robert Lucas suggested something like this to me a few years ago, and subsequent to this discussion I realized that a version of this story works for the finite horizon optimal growth setup. Marianne Baxter's paper then clarified the discussion for me (also in the context of the optimal growth setup), which led to the generalization of the story presented here.
2. That is, within the set of piecewise linear functions a sequence can be constructed which converges uniformly to any given $h \in H$.
3. This problem (with iid shocks) is described in Brock and Mirman (1972).
4. In Coleman (1987) I construct a set H which is convex and compact, and a function $A:H \rightarrow H$ which is continuous, thus guaranteeing the existence of a fixed point. In the deterministic setting, and with a specific assumption on the utility function, I essentially prove A 's monotonicity and strict concavity, thus proving the uniform convergence of $A^n(h)$ to the unique fixed point.
5. The intuitive argument for convergence given above also has a loose analogue here. In this case, h fixes the control for tomorrow, and $A(h)$ is the optimal control which takes into account both the effect on the state today and the direction of the state into tomorrow.
6. Approximating $h'(x)g[x, h(x)]$ via partial derivatives results in an unnecessary interaction between $A(h)$ in the derivative and $A(h)$ in g (e.g., if g is linear then a quadratic in $A(h)$ appears).
7. In the following example, if $f(1) = 0$ and if $h'(1)$ does not equal the true value exactly, then $(Ah)(1) = 0$ if $A(h)$ does not enter the derivative as above. The correct

solution is $h(1) > 0$, and the above algorithm results in $(Ah)(1) > 0$.

8. This problem was solved by Cass (1965) and Koopmans (1965).

9. Suppose $-u'(c)/u''(c)$ is increasing in c , then, if h is increasing, and if Δ is small, $A(h)$ is uniquely defined since

$$z(y, x) = \frac{-u'(y)}{u''(y)} [f'(x) - \rho] - \frac{h\{x + [f(x) - y]\Delta\} - y}{\Delta}$$

is increasing in y (these statements need only be true for some large upper bound on x , say \bar{x} such that $f(\bar{x}) = 0$). Provided, again, that Δ is small, then $A(h)$ will be increasing since $z(y, x)$ is decreasing in x . These results, though, do not guarantee convergence or existence, but only that $A(h)$ is well defined at each step of the iteration.

REFERENCES

- Baxter, Marianne, 1988, Approximating Suboptimal Dynamic Equilibria: An Euler Equation Approach, Rochester Center for Economic Research Working Paper 139.
- Brock, William A. and Leonard J. Mirman, 1972, Optimal Economic Growth and Uncertainty: The Discounted Case, *Journal of Economic Theory* 4, 497–513.
- Cass, D., 1965, Optimum Growth in an Aggregative Model of Capital Accumulation, *Review of Economic Studies* 32, .
- Coleman, Wilbur John II, 1987, Money, Interest, and Capital, unpublished Ph.D. Dissertation, The University of Chicago.
- Courant, R. and D. Hilbert, 1962, *Methods of Mathematical Physics, Vol. II* (John Wiley and Sons, New York).
- Fleming, Wendell H. and Raymond W. Rishel, 1975, *Deterministic and Stochastic Optimal Control* (Springer–Verlag, Berlin).
- John, Fritz, 1971, *Partial Differential Equations* (Springer–Verlag, New York).
- Kantorovitch, L. V., 1948, Functional Analysis and Applied Mathematics, *Uspekhi Matematicheskikh Nauk* 3, no. 6, 89–185, translated from Russian by: Curtis D. Benster, 1952, National Bureau of Standards Report 1509.
- Koopmans, T. C., 1965, On the Concept of Optimal Economic Growth, in: *The Econometric Approach to Development Planning*, Pontificiae Academiae Scientiarum Scriptum Varia (North–Holland, Amsterdam).
- Krasnosel'skiĭ, M. A. and P. P. Zabreiko, 1984, *Geometrical Methods of Nonlinear Analysis* (Springer–Verlag, Berlin).
- Ortega, J. M. and W. C. Rheinboldt, 1970, *Iterative Solution of Nonlinear Equations in Several Variables* (Academic Press, New York).
- Taylor, John B. and Harald Uhlig, 1989, Solving Nonlinear Stochastic Growth Models: A Comparison of Alternative Solution Methods, unpublished working paper, The University of Minnesota and The Institute for Empirical Macroeconomics.

International Finance Discussion Papers

<u>IFDP NUMBER</u>	<u>TITLES</u> <u>1989</u>	<u>AUTHOR(s)</u>
351	An Algorithm to Solve Dynamic Models	Wilbur John Coleman II
350	Implications of the U.S. Current Account Deficit	David H. Howard
349	Financial Integration in the European Community	Sydney J. Key
348	Exact and Approximate Multi-Period Mean-Square Forecast Errors for Dynamic Econometric Models	Neil R. Ericsson Jaime R. Marquez
347	Macroeconomic Policies, Competitiveness, and U.S. External Adjustment	Peter Hooper
346	Exchange Rates and U.S. External Adjustment in the Short Run and the Long Run	Peter Hooper
345	U.S. External Adjustment: Progress and Prospects	William L. Helkie Peter Hooper
344	Domestic and Cross-Border Consequences of U.S. Macroeconomic Policies	Ralph C. Bryant John Helliwell Peter Hooper
343	The Profitability of U.S. Intervention	Michael P. Leahy
342	Approaches to Managing External Equilibria: Where We Are, Where We Might Be Headed, and How We Might Get There	Edwin M. Truman
341	A Note on "Transfers"	David B. Gordon Ross Levine
340	A New Interpretation of the Coordination Problem and its Empirical Significance	Matthew B. Canzoneri Hali J. Edison
339	A Long-Run View of the European Monetary System	Hali J. Edison Eric Fisher
<u>1988</u>		
338	The Forward Exchange Rate Bias: A New Explanation	Ross Levine
337	Adequacy of International Transactions and Position Data for Policy Coordination	Lois Stekler

Please address requests for copies to International Finance Discussion Papers, Division of International Finance, Stop 24, Board of Governors of the Federal Reserve System, Washington, D.C. 20551.

International Finance Discussion Papers

<u>IFDP NUMBER</u>	<u>TITLES</u>	<u>AUTHOR(s)</u>
336	Nominal Interest Rate Pegging Under Alternative Expectations Hypotheses	Joseph E. Gagnon Dale W. Henderson
335	The Dynamics of Uncertainty or The Uncertainty of Dynamics: Stochastic J-Curves	Jaime Marquez
334	Devaluation, Exchange Controls, and Black Markets for Foreign Exchange in Developing Countries	Steven B. Kamin
333	International Banking Facilities	Sydney J. Key Henry S. Terrell
332	Panic, Liquidity and the Lender of Last Resort: A Strategic Analysis	R. Glen Donaldson
331	Real Interest Rates During the Disinflation Process in Developing Countries	Steven B. Kamin David F. Spigelman
330	International Comparisons of Labor Costs in Manufacturing	Peter Hooper Kathryn A. Larin
329	Interactions Between Domestic and Foreign Investment	Guy V.G. Stevens Robert E. Lipsey
328	The Timing of Consumer Arrivals in Edgeworth's Duopoly Model	Marc Dudey
327	Competition by Choice	Marc Dudey