NBER TECHNICAL WORKING PAPER SERIES

A NOTE ON THE TIME-ELIMINATION METHOD FOR SOLVING RECURSIVE DYNAMIC ECONOMIC MODELS

Casey B. Mulligan

Xavier Sala-i-Martin

Technical Working Paper No. 116

NATIONAL BUREAU OF ECONOMIC RESEARCH
1050 Massachusetts Avenue
Cambridge, MA 02138
November 1991

The September 15, 1991 5:00 PM version of this paper is based on the September 15, 1991 4:59 PM version of Mulligan (1991). The examination of growth models in the final section of the Mulligan paper had been greatly improved after many discussions with Professor Sala-i-Martin. He also appreciates several stimulating discussions with Serge Marquis. Casey's parents enthusiastically invested in computer hardware and software. This paper is part of NBER's research program in Growth. Any opinions expressed are those of the authors and not those of the National Bureau of Economic Research.

NBER Technical Working Paper #116
November 1991

A NOTE ON THE TIME-ELIMINATION METHOD
FOR SOLVING RECURSIVE DYNAMIC ECONOMIC MODELS

ABSTRACT

The Time-Elimination Method for solving recursive dynamic economic models is described. By defining control-like and state-like variables, one can transform the equations of motion describing the economy's evolution through time into a system of differential equations that are independent of time. Unlike the transversality conditions, the boundary conditions for the system in the state-like variable are not asymptotic boundary conditions. In theory, this reformulation of the problem greatly facilitates numerical analysis. In practice, problems which were impossible to solve with a popular algorithm - shooting - can be solved in short order.

The reader of this paper need not have any knowledge of numerical mathematics or dynamic programming or be able to draw high dimensional phase diagrams. only a familiarity with the first order conditions of the "Hamiltonian" method for solving dynamic optimization problems is required.

The most natural application of Time-Elimination is to growth models. The method is applied here to three growth models: the Ramsey/Cass/Koopmans one sector model, Jones & Manuelli's (1990) variant of the Ramsey model, and a two sector growth model in the spirit of Lucas (1988). A very simple - but complete - computer program for numerically solving the Ramsey model is provided.

Casey B. Mulligan
5528 S. Hyde Park Boulevard #805
Chicago, IL 60637

Xavier Sala-i-Martin
Yale University
Economics Department
28 Hill House
New Haven, CT 06520-1972
and NBER

At least since Ramsey's 1928 analysis of optimal capital accumulation, economists have recognized the advantages of thinking about the macroeconomy in terms of dynamic optimization problems. In more modern times it has even been suggested that young macroeconomists study dynamic optimization before they learn about Keynes' *General Theory*. In light of such developments, economists should know how to solve dynamic optimization problems. This paper describes a very simple, but not commonly employed, method for numerically solving a certain class of such problems.[2] The method is applied to three growth models.

The Time-Elimination Method is a very powerful and efficient algorithm for numerically solving dynamic optimization problems. When applicable, it vastly improves upon a common alternative algorithm, shooting. Time-Elimination can be thought of as an alternative interpretation of Ken Judd's (1990) work on numerical techniques for solving dynamic models. Judd's paper successfully achieves at least two objectives. First, he provides some very general algorithms for numerically solving dynamic models. Second, his algorithms economize on computer time.

The objective here is to economize on all of the inputs for the process of solving models. Among these are computing time, computing power, the "raw labor" of the researcher and the human capital of the researcher. I therefore concentrate on only a few of Judd's techniques, expressing them in terms of the Maximum Principle. The cost of my approach is that I narrow the class of models which can be examined. However, because relatively little knowledge of the mathematics of dynamic optimization and no familiarity with numerical methods is required to make use of the very practical and efficient algorithms in this paper, my approach requires less human capital and is therefore accessible by a wider audience.[3] The exposition here assumes that the reader:

(i) is familiar with the first order conditions of the "Hamiltonian" method for

[2] Skiba (1978) eliminates time from his optimal growth problem in order to gain some analytical insights.

[3] Judd's approach is both elegant and widely applicable. However, his readers should be comfortable with continuous time dynamic programming, Ito's calculus, and the idea of numerically searching for solutions to functional equations. The emphasis of Judd's paper is a detailed examination of the sixth of my six step algorithm.

solving dynamic optimization problems

(ii) has a vague understanding of the transversality conditions

(iii) has access to a computer package that can solve - for initial value problems - first order ordinary differential equations (eg., MATLAB)

Here I apply Time-Elimination to recursive, deterministic continuous-time dynamic optimization problems. Recursiveness is crucial because it is assumed that optimal controls can be described by a *policy function*. In other words, there must be a single functional relationship - independent of time - between optimal controls and the state of the economy at all dates. The key advantages of Time-Elimination are more simply communicated by limiting the discussion to deterministic problems. A theme of this paper is that the continuous-time formulation of economic problems may not only be more realistic than a discrete time one, but may offer significant computational advantages.

To solve a dynamic economic model, I follow a six step algorithm:

(i) Write down the model in terms of a continuous-time dynamic optimization problem

(ii) Use the Maximum Principle to describe the dynamics of the optimal controls

(iii) Define *state-like* and *control-like* variables

(iv) Argue that the transversality conditions (TVC's) require the state-like and control-like variables to approach constant steady state values

(v) Use the dynamics of the state-like and control-like variables to derive a system of differential equations describing *policy functions*

(vi) Numerically solve the system (from (v)), subject to steady state values

If one is specifically interested in the time paths of optimal controls, rather than policy functions, three steps can be added:

(vii) "Substitute" the policy functions into the system (written down in step (i)) which describes the dynamics of the state variables

(viii) Numerically solve this system, subject to initial state variables (also specified in step (i)) and subject to initial controls (use policy functions here).

(ix) Use the policy functions and the results of (viii) to find the time paths for optimal controls

The first three sections of this note will discuss these steps in more detail. First, a general dynamic optimal control problem is outlined and frequently encountered computational difficulties are noted. Following steps (iii) & (iv), the second section proposes a change in variables; the TVC's can then be conceptualized by thinking about a steady state. Section III stresses the importance of thinking about policy functions instead of time paths for optimal controls. The algorithm yields a system of differential equations - one for which the (numerical) application of boundary conditions is quite simple. The system describing policy functions and its "new" boundary conditions offer huge computational improvements over solution procedures which rely on time dynamics - solutions can be found in a few seconds on a computer as small as an IBM XT. A fourth section provides an intuitive explanation of numerical solution procedures for this system. A fifth section notes that time dynamics can be easily derived from the policy functions.

As an example, the final section considers three growth models: that of Ramsey/Cass/Koopmans, Jones & Manuelli's (1990) variant of the Ramsey model, and a two sector growth model in the spirit of Lucas (1988). The analysis of this final section deliberately follows the structure of the first five so that the reader may look ahead for concrete examples when reading the early sections. Since the algebra behind the solution of these interesting models is trivial and the computer program provided (for the Ramsey model) is shockingly brief, the reader will agree that the Time-Elimination Method also economizes on labor input.

I. A Dynamic Optimization Problem and the "Shooting" Method of Solution

Consider the following dynamic optimization problem:

$$\max_{u(t)} \; V_0 \equiv \int_0^{\infty} f\big(x(t),u(t),t\big)dt \qquad \text{s.t.}$$

$$\dot{x}(t) = g\big(x(t),u(t),t\big)$$

$$x(0) \text{ given}$$

Where x(t) is a vector of n "state" variables and u(t) is a vector of n "control" variables.

The Maximum Principle yields a set of 2n first order conditions and a set of n TVC's.[4] For the class of problems that we consider, one can eliminate the shadow prices to obtain a set of differential equations (in time) for the optimal controls. Together with the assumed dynamics for the state variables, these Euler equations form a system of 2n differential equations (in time) to describe optimal evolution of the economy - often called the equations of motion:

Optimal u(t) and x(t) are solutions to a *boundary value* type system of ordinary differential

[4] It will be assumed that if a candidate solution satisfies the first order conditions and the TVC's, then it is optimal.

$$\dot{u}(t) = h(x(t),u(t),t)$$
$$\dot{x}(t) = g(x(t),u(t),t)$$

equations in time. The system is the equations of motion and the boundary conditions are the TVC's. Since closed form solutions for u(t) and x(t) do not always exist, numerical solution procedures may be necessary. A popular approach for solving boundary value problems, "shooting", guesses initial controls u(0) (for a given x(0)). The subsequent dynamics of the economy (as dictated by the equations of motion) are then examined to see if the TVC's are violated. If so, the initial guess is revised and the process is repeated.[5]

There are several problems with shooting. First, the computer programming required may be lengthy and tedious. Second, when n is greater than 1, it is quite difficult to determine whether TVC's will be satisfied by the time path in question. Finally, the procedure will take a significant amount of computing time (when compared to the methods which follow). The Time-Elimination Method does not share these problems with shooting.

Numerically, boundary value problems are much more difficult - both conceptually and computationally - to solve than are *initial value* problems.[6] One key advantage of the Time-Elimination Method is that it transforms the boundary value problem described by the equations of motion and the TVC's into an initial value problem. Sections II and III will explain how one can make this transformation.

II. TVC's & Steady States

It is common to think about a "steady state" describing the behavior of the system at $t = \infty$. We will be looking for "state-like" variables and "control-like" variables. First, ***state-like and control-like variables will be constant in the steady state.*** Second, control-like variables will be a function of the states and the controls. State-like variables will be functions of the states only.

---

[5] Section VI includes a discussion of the shooting method for the Ramsey problem.

[6] One a more pragmatic level, computer math packages are much more likely to include routines that solve initial value problems than to include routines that solve boundary value problems. I use MATLAB's ODE23 routine to solve initial value problems. I can therefore worry about economics rather than numerical mathematics (I believe that I have a comparative advantage in the former). See Press, et al (1990) for a comparison of initial value and boundary value problems.

$$\text{(vector of) state-like variables: } y(t) = y(x(t))$$
$$\text{(vector of) control-like variables: } v(t) = v(u(t), x(t))$$

If the state-like and control-like variables were skillfully chosen, the dynamics for u & x can be used to find dynamics for v & y:

$$\dot{v}(t) = q(y(t), v(t), t)$$
$$\dot{y}(t) = r(y(t), v(t), t)$$

We will assume that the TVC's are equivalent to requiring that the state-like and control-like variables approach their constant steady state values.[7]

$$\lim_{t\to\infty} v(t) = v_{\infty} < \infty \quad \leftrightarrow \quad \lim_{t\to\infty} \dot{v}(t) = 0$$
$$\lim_{t\to\infty} y(t) = y_{\infty} < \infty \quad \leftrightarrow \quad \lim_{t\to\infty} \dot{y}(t) = 0$$

Numerically, this steady state relationship will be quite advantageous.

III. We can't wait forever, or
Use Policy Functions to Transform Transversality Conditions into "Initial Conditions"

Instead of time paths, we will look for *policy functions*, which express optimal choices for the control-like variables as a function of the state-like variables, instead of as functions of time:

[7] Consider the Ramsey/Cass/Koopmans growth model (this will be addressed in detail later). In this model, we commonly assume that the TVC's require that consumption and capital approach their steady state values. Capital would both be a state variable and a state-like variable. Consumption would both be a control variable and a control-like variable.

$$\text{policy functions: } v^* = p(y)$$

$$\text{time paths: } v^* = v(t)$$

In special cases, the dynamic system of section I (and its TVC's) can be transformed into a system of ordinary differential equations *for the policy functions*. In such special cases, the TVC's will be equivalent to requiring that all optimal time paths approach their steady state values. Second, the functions q and r will not depend explicitly on time. Third, the vector y will be only one dimensional; each policy function will depend on only one state-like variable:[8]

$$v_i(t) = p_i(y(t)) \qquad y \text{ is a } 1\times1 \text{ vector} \qquad i = 1...n$$

To transform the dynamics of section II into a system of ordinary differential equations for the policy functions, note that the ratio of the time derivative of a control-like variable to that of its corresponding state-like variable is equal to the first derivative of the policy function:

$$p_i'(y) = \frac{d\,p_i(y)}{d\,y} = \frac{\dot{v}_i(t)}{\dot{y}(t)} = \frac{q_i(y,p)}{r(y,p)} \qquad i = 1...n$$

The boundary conditions for this system are the TVC's. Conveniently, since time has been eliminated, the TVC's are no longer asymptotic boundary conditions. Denote this system of n differential equations as:

$$p'(y) = F\big(y,p(y)\big) \text{ s.t.}$$
$$p(y_{ss}) = v_{ss}$$

[8] If this third requirement is not fulfilled, then we would end up with a system of partial differential equations instead of a system of ordinary differential equations. In principle, this is not a problem, but I am guessing that it is more difficult to access a computer package which solves partial differential equations. Appendix II discusses how one may find policy functions that depend on more than one variable without thinking about partial differential equations. However, the Time-Elimination Method looses some of its advantage in such cases.

As presented here, this third requirement is purely a mathematical restriction. However, I have not found it to be economically restrictive. See Mulligan & Sala-i-Martin (1991) for some very interesting examples and some economic intuition behind the "one state-like variable" restriction. Appendix III relates the restriction to the existence of a balanced growth path in an n-sector growth model.

Before numerically solving this system, we must address two technical details. First, as defined above, $p_i'(y)$ is 0/0 in the steady state - we need to find the slope of the policy functions in the steady state. One way to do this is to apply L'Hopital's rule.[9]

A second problem is that application of L'Hopital's rule will yield n+1 "solutions" for the slope of the policy function at the steady state. This is because both stable and unstable arms satisfy the system of differential equations and both pass through the steady state (only the stable arm satisfies the TVC's). Often economic intuition will tell us which "solution" corresponds to the stable arm. When this is not the case, I randomly choose a solution, assume that it is stable and continue with the algorithm. Numerical implementation is so easy and efficient that is not inconvenient to come back to this step when I realize that I have chosen an unstable arm.

IV. Numerical Solutions

Now we know a point on each policy function - the steady state - and the slope of the policy functions as a function of p and y. Conceptually, it is obvious that a numerical solution would take the following steps (let n = 1 for clarity):

(i) calculate the slope of the policy function at the steady state
(ii) in the v,y plane, go a "small" distance in the direction of the derivative calculated in (i)
(iii) recalculate the slope at this new point
(iv) repeat steps (ii) & (iii) until a sufficient portion of the policy function has been "traced out"

It is common for computer math packages to come equipped with a subroutine that can do the four steps above.[10]

Note that these four steps will "trace out" the policy function for $y > y_{ss}$. Of course, the process can easily be run "backwards" to obtain the "other half" of the policy function.

How is it that "time-elimination" is a crucial step in this algorithm? In theory, the elimination of time only reduces the dimension of the problem by one. In theory, one could start

[9] Alternatively, linearization around the steady state will give the exact slope at the steady state and will distinguish stable and unstable arms.

[10] In MATLAB, the subroutines ODE23 and ODE45 perform this function. Strictly speaking, the method described in the text is the "Euler method." It captures the intuition behind fancier methods, including the (automatic step size) Runge-Kutta method, which is used by MATLAB. See Press, et al. (1990), chapter 15 for a discussion of some fancy methods.

the economy very near to the steady state and apply the shooting algorithm backwards.[11] Such a procedure - let's call it *shooting backwards* - could enjoy most of the computational savings of the time-elimination method, since shooting backwards is also an initial value problem. Where things become quite difficult in practice is starting the economy "very near to the steady state." Even for simple problems, some human capital and a lot of raw labor is required to write a computer routine that performs this step. With the Time-Elimination method, that first point "very near to the steady state" is expertly chosen by MATLAB's ODE23 routine.

V. Policy Functions?
But I want time paths!

After numerically finding policy functions, it is quite easy to derive time paths for the state-like and control-like variables. Since x(0) was given in the problem formulation, we know y(0). Using the policy function, we can therefore find v(0). We therefore have a system of differential equations (the vector function r) and know initial conditions:

$$\dot{y}(t) = r\big(y(t), p(y(t))\big) \equiv G\big(y(t)\big)$$
$$\text{s.t.} \quad y(0) \equiv y(x(0))$$

The system G can be solved exactly as was the system F in the previous section (the solution is a time path for the vector of state-like variables y).[12]

Use the policy functions and the solution to the system G to find time paths for the control-like variables:

$$v(t) = p\big(y(t)\big)$$

[11] It is this procedure that Appendix II suggests for problems with more than one state-like variable.

[12] In section IV, we did not find policy *functions* in the usual sense of the word - we found a (finite element) set of ordered pairs that lie on the true policy function. In practice, I treat these ordered pairs as a function by interpolating between them. Conveniently, MATLAB provides subroutines to perform this interpolation.

VI. Three Growth Models[13]

This final section applies the Time-Elimination Method to three growth models. The simplest is the Ramsey/Cass/Koopmans one sector growth model. Here there is no steady state growth, so we do not have to define new state-like and control-like variables. Appendix I provides the two very brief MATLAB files that I use to find policy functions for the Ramsey model. The second part of this section considers a one sector growth model for which there is endogenous growth. I therefore define a state-like variable, the potential output to capital ratio, that is constant in the steady state. Finally, this section concludes with a two sector growth model. Although there are two state variables, there is still only one state-like variable - the potential output to capital ratio. Even though shooting is a nightmare for the model, Time-Elimination is applied just as easily as it was for the one sector models.

As one works with these three growth models, two methodological rules of thumb emerge. First, the potential output to capital ratio is an appropriate state-like variable for all of these prototypical growth models. Second, dynamic economic problems may be more easily analyzed in a continuous-time framework. This is because continuous-time models tend to have, at least for particular (but fairly interesting) parameterizations, elegant closed form solutions. These closed form solutions can be easily used to think about empirical implications of the model. For more general parameterizations - those for which closed form solutions are not available, the Time-Elimination Method can be used to check the qualitative robustness of conclusions based on a particular closed form formula.

A. Ramsey/Cass/Koopmans

The characteristic assumptions of this one sector growth model are the infinite horizon representative agent, a production function which satisfies the Inada conditions and the linearity of c in the capital accumulation equation. Optimal growth can be described as the solution to a dynamic optimization problem:

[13] See Sala-i-Martin (1990) and Mulligan & Sala-i-Martin (1991) for economic interpretations and empirical implications of these (and other) growth models.

$$\max_{c(t)} \int_0^\infty e^{-\rho t} U(c(t))dt$$

$$\text{s.t.} \quad \dot{k}(t) = f(k(t)) - c(t) - \delta k(t)$$
$$k(0) \text{ given}$$

Let production be Cobb-Douglas and utility exhibit CIES. Write down the Hamiltonian for this problem:

$$H(c,k,\lambda,t) = e^{-\rho t}\cdot\left(\frac{c^{1-\theta}-1}{1-\theta}\right) + \lambda\cdot\left[k^{\alpha} - c - \delta\cdot k\right]$$

The usual first order conditions describe the time dynamics for optimally chosen consumption:

$$\dot{c}(t) = \frac{c(t)}{\theta}\cdot\left[\alpha\cdot k(t)^{\alpha-1} - \rho - \delta\right]$$

For this problem consumption will serve as both the control variable and the control-like variable. k is both a state and a state-like variable. This is because, in this problem, the state and control variables do themselves approach constant steady state values:

$$\lim_{t\to\infty} c(t) = c_\infty < \infty \quad \leftrightarrow \quad \lim_{t\to\infty} \dot{c}(t) = 0$$

$$\lim_{t\to\infty} k(t) = k_\infty < \infty \quad \leftrightarrow \quad \lim_{t\to\infty} \dot{k}(t) = 0$$

One finds the steady state values:

$$k_\infty = \left(\frac{\alpha}{\rho+\delta}\right)^{\frac{1}{1-\alpha}}, \quad c_\infty = k_\infty^{\alpha} - \delta\cdot k_\infty$$

Shooting

Figure 1 below is a phase diagram for the Ramsey problem. Three curves are emerging from the origin. The top curve is the gross production function $f(k)$. The middle curve is the net production function, $f(k) - \delta\cdot k$. It is also the locus of points (k,c) for which (net) capital accumulation is zero. The lowest curve is the stable arm. The steady state (k_{ss}, c_{ss}) is the intersection of the $\dot{k} = 0$ and $\dot{c} = 0$ schedules.

The four arrowed curves are paths which satisfy the equations of motion. We have argued that the optimal solution - the stable arm - must necessarily obey the equations of motion, but also approach (k_{ss}, c_{ss}). In order to numerically find the stable arm, the shooting method (for some k(0)) guesses initial consumption. One then finds the time path through (k(0), c(0)) that obeys the equations of motion. If this path does not reach the steady state (you're really lucky if it does), as is the case for three paths in Figure 1, then the guess for initial consumption c(0) is adjusted "appropriately" and the process is repeated until a time path is found that gets sufficiently close to the steady state.

Clearly it will take some time for you to code (without bugs) this algorithm irto your computer. For more complicated problems, it is not obvious how to "appropriately" adjust your guess for the optimal initial control.[14]

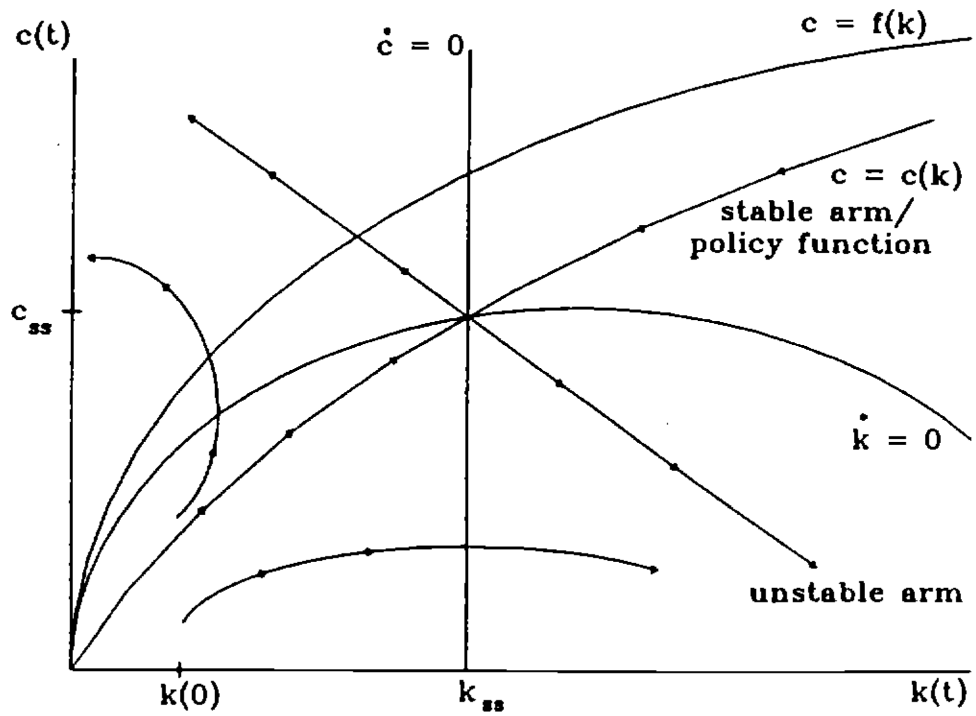

Figure 1 Phase Diagram for the Ramsey Problem

[14] I invite (dare) the reader to draw a phase diagram for the Lucas model in section VIC and for the more general two sector models in Mulligan & Sala-i-Martin (1991).

Policy Functions

The first order condition and the steady state relationships can used to describe a *policy function* c(k) for consumption. In particular, the policy function satisfies the following differential equation (in k):

$$c'(k) = \frac{\dot{c}(t)}{\dot{k}(t)} = \frac{c(k)}{\theta} \cdot \frac{\alpha \cdot k^{\alpha-1} - (\rho+\delta)}{k^{\alpha} - c(k) - \delta \cdot k}$$

$$\text{s.t.} \quad c_{ss} = c(k_{ss})$$

The above differential equation[15] specifies the slope of the policy function everywhere except at the steady state. Figure 1 shows that the differential equation (in k) and its boundary condition is solved by both the stable and unstable arms - since both satisfy the equations of motion and both get arbitrarily close to the steady state. Once the slope at the steady state is specified, the differential equation and its boundary condition will be uniquely solved by the stable arm. One can use L'Hopital's Rule on the above differential equation (and the economic intuition that the slope of the policy function is positive) to find the slope of the policy function at the steady state:

$$c'(k_{ss}) = \frac{\rho + \sqrt{\rho^2 + \frac{4}{\theta} \cdot c_{ss} \cdot \alpha \cdot (1-\alpha) \cdot k_{ss}^{\alpha-2}}}{2}$$

Time Paths

For some applications, one may want to express an economic variable, say consumption, as a function of time. To do this, notice that - with the policy function - the rate of change of k with respect to time is a function of k only. This is a differential equation (in time)[16] which can be numerically solved in the same manner as was the differential equation for the policy function:

$$\dot{k}(t) = k(t)^{\alpha} - c(k(t)) - \delta \cdot k(t)$$

$$\text{s.t.} \quad k(0) = \text{prespecified initial capital stock}$$

With some interpolation, variables which could be expressed as functions of k can now be expressed

---

[15] This system corresponds to the system F in section III.

[16] This system corresponds to the system G in section V.

as functions of t.

A Program

The (only) two MATLAB files that I use to find policy functions for the Ramsey model are included in Appendix I. For simplicity, this program does not find time paths. Note that ODE23 is the MATLAB subroutine for solving systems of ordinary differential equations.

On my 386SX 16MHz (with a math co-processor), I can use these files to find the policy function (conditional on a set of parameters) in a mere 4 seconds! Some experimentation reveals that, on a 33Mhz 386, one can find policy functions for one hundred sets of parameters in one minute. By contrast, shooting would take about 15 *minutes* for each set of parameters!

Potential Output per Unit of Capital: State-like Variables Again

The above solution procedure chose c & k as state-like and control-like variables. This was done because these are the variables that growth theorists like to think about in the Ramsey model. However, the solution of two "endogenous" growth models which follow suggest a different choice, *potential output per unit of capital*:[17]

$$\text{control-like variable:} \quad a(t) = \frac{c(t)}{k(t)}$$

$$\text{state-like variable:} \quad z(t) = \frac{f(k(t))}{k(t)}$$

My conjecture is that whenever c enters linearly in the capital accumulation equation, z(t) (or a simple transformation) can be used as a state-like variable. This is true in the three models examined here. Using the Time-Elimination Method, one can quickly analyze these models for various sets of parameters. One quickly notices that the policy function a(z) is very nearly linear in these models:

$$\begin{aligned} a(z) &\doteq A + B\cdot z \\ \text{i.e.,} \quad c(k) &\doteq A\cdot k + B\cdot f(k) \end{aligned}$$

"Closed Form Solutions"

Thus, to a close approximation, we have a closed form solutions for these models. Look at the "closed form solution" for the Ramsey model (production is still Cobb-Douglas):

[17] The need for the qualifier "potential" will become apparent with the Lucas model.

$$c(k) \approx \left(\frac{\rho+\delta}{\theta\cdot\alpha} - \delta\right)\cdot k + \left(\frac{\theta-1}{\theta}\right) f(k) \equiv \hat{c}(k)$$

This closed form solution $\hat{c}(k)$ shares the following properties with the true solution c(k):

(i) <u>concavity</u> for large θ

(ii) <u>constant savings</u> for the same sets of parameters

i.e., when $\theta = \frac{\rho+\delta}{\delta\cdot\alpha}$, the solution is exact:

$$c(k) = \hat{c}(k) = \frac{\theta-1}{\theta}\cdot f(k)$$

(iii) For some parameterizations, consumption is a linear function of capital

This occurs for the true policy function when $\theta = \alpha$ and for the approximation when $\theta = 1$

(iv) $c_{ss} = c(k_{ss}) = \hat{c}(k_{ss})$

(v) $0 = c(0) = \hat{c}(0)$

(vi) <u>No investment</u> when consumers are unwilling to substitute intertemporally:

$$\lim_{\theta\to\infty} c(k) = \lim_{\theta\to\infty} \hat{c}(k) = f(k) - \delta\cdot k$$

B. Jones & Manuelli (1990)

Consider modifying the Ramsey model so that production no longer satisfies the Inada conditions:

$$\max_{c(t)} \int_0^\infty e^{-\rho t} \cdot \frac{c(t)^{1-\theta}}{1-\theta} dt$$

$$\text{s.t.} \quad \dot{k}(t) = A \cdot k(t) + B \cdot k(t)^\alpha - c(t) - \delta \cdot k(t)$$

$$k(0) \text{ given}$$

First order conditions describe the time dynamics for optimally chosen consumption:

$$\dot{c}(t) = \frac{c(t)}{\theta} \cdot \left[A + B \cdot \alpha \cdot k(t)^{\alpha-1} - \rho - \delta\right]$$

Control-like & State-like Variables

In this model, consumption and capital do not approach finite steady state values - there is endogenous growth. Possible choices for control-like & state-like variables are:

$$\text{control-like variable: } a(t) \equiv \frac{c(t)}{k(t)}$$

$$\text{state-like variable: } z(t) \equiv \frac{y(k(t))}{k(t)} = \frac{A \cdot k(t) + B \cdot k(t)^\alpha}{k(t)}$$

Using these definitions, one can derive time dynamics for the state-like & control-like variables:

$$\dot{a}(t) = a(t) \cdot \left[\frac{A \cdot (1-\alpha) - (\rho + \delta)}{\theta} + \frac{\alpha - \theta}{\theta} \cdot z(t) + a(t) + \delta\right]$$

$$\dot{z}(t) = (\alpha - 1) \cdot [z(t) - A] \cdot [z(t) - a(t) - \delta]$$

These state-like & control-like variables do approach constant steady state-values:

$$\lim_{t\to\infty} a(t) = a_\infty < \infty \quad \leftrightarrow \quad \lim_{t\to\infty} \dot{a}(t) = 0$$

$$\lim_{t\to\infty} z(t) = z_\infty < \infty \quad \leftrightarrow \quad \lim_{t\to\infty} \dot{z}(t) = 0$$

Policy Functions

Use the dynamics of the state-like and control-like variables to describe a policy function a(z):[18]

$$a'(z) = \frac{\dot{a}(t)}{\dot{z}(t)} = \frac{a(z)}{(\alpha-1)\cdot(z-A)\cdot\theta} \cdot \frac{A\cdot(1-\alpha) - (\rho+\delta) + (\alpha-\theta)\cdot z + a(z)\cdot\theta + \theta\cdot\delta}{z - a(z) - \delta}$$

$$\text{s.t.} \quad a_\infty = a(z_\infty)$$

Time Paths

These are derived as with the Ramsey model. When the policy function is known, the dynamics for the state-like variable (z) is a differential equation in time:

$$\dot{z}(t) = (\alpha-1)\cdot[z(t) - A]\cdot[z(t) - a(z(t)) - \delta]$$

[18] This system corresponds to the system F in section III.

C. Lucas (1988): Two Sector Growth[19]

Growth in a two sector economy can be described as the solution to the following dynamic optimization problem (this is a special case of Lucas (1988)):

$$\max_{c(t),\,u(t)} \; V_0 = \int_0^{\infty} e^{-\rho t}\cdot\left(\frac{c(t)^{1-\theta}}{1-\theta}\right) dt \quad \text{s.t.}$$

$$\dot{k}(t) = u(t)^{1-\alpha}\cdot k(t)^{\alpha}\cdot h(t)^{1-\alpha} - c(t) - \delta\cdot k(t)$$

$$\dot{h}(t) = \phi\cdot(1-u(t))\cdot h(t)$$

The usual first order conditions can be reduced to equations for the (time) dynamics for optimal controls:

$$\dot{c}(t) = \frac{c(t)}{\theta}\cdot\left[\alpha\cdot u(t)^{1-\alpha}\cdot k(t)^{\alpha-1}\cdot h(t)^{1-\alpha} - \rho - \delta\right]$$

$$\dot{u}(t) = u(t)\cdot\left[\frac{\phi + \delta - \alpha\cdot u(t)^{1-\alpha}\cdot k(t)^{\alpha-1}\cdot h(t)^{1-\alpha}}{\alpha} + \gamma_k - \gamma_h\right]$$

with definition:

$$\gamma_x \equiv \frac{\dot{x}(t)}{x(t)}$$

Define the steady state:

$$\gamma_c = \gamma_k = \gamma_h \equiv \gamma$$
$$\gamma_u = 0$$

From this definition, the first order conditions and the dynamics for h & k, the following *steady state* relationships can be derived:

[19] Interpretations, out of steady-state behavior, and other interesting features of some more general two sector growth models are discussed in Mulligan & Sala-i-Martin (1991).

$$\gamma_{\infty} = \frac{1}{\theta}(\phi - \rho)$$

$$u_{\infty} = 1 - \frac{\phi - \rho}{\theta \cdot \phi}$$

$$\left(\frac{k}{h}\right)_{\infty} = \left(\frac{\delta + \phi}{\alpha}\right)^{\frac{1}{\alpha - 1}} \cdot u_{\infty}$$

$$\left(\frac{c}{k}\right)_{\infty} = \frac{\delta + \phi}{\alpha} - \frac{\phi - \rho}{\theta} - \delta$$

Describing the Policy Functions

Define control-like variables a & b and state-like variable z:[20]

$$a(t) \equiv \frac{c(t)}{k(t)} \qquad b(t) \equiv u(t) \qquad z(t) \equiv \frac{k(t)}{h(t)}$$

From these definitions, find dynamics for the state-like and control-like variables. Use these dynamics to describe the policy functions by a system of two first order differential equations:

$$\gamma_a = \gamma_c - \gamma_k = \frac{1}{\theta}\left[\alpha \cdot \left(\frac{b(t)}{z(t)}\right)^{1-\alpha} - \rho - \delta\right] - \left(\frac{b(t)}{z(t)}\right)^{1-\alpha} + a(t) + \delta$$

$$\gamma_z = \gamma_k - \gamma_h = \left(\frac{b(t)}{z(t)}\right)^{1-\alpha} - a(t) - \delta - \phi \cdot (1 - b(t))$$

$$a'(z) = \frac{\dot{a}(t)}{\dot{z}(t)} = \left[\frac{a(z)}{z} \cdot \frac{\frac{\alpha - \theta}{\theta} \cdot \left(\frac{b(z)}{z}\right)^{1-\alpha} - \frac{\rho}{\theta} + \frac{\theta - 1}{\theta} \cdot \delta + a(z)}{\gamma_z}\right] \equiv f(a(z), b(z), z)$$

$$b'(z) = \frac{\dot{b}(t)}{\dot{z}(t)} = \left[\frac{b(z)}{z} \cdot \frac{\frac{\delta + \phi}{\alpha} - \left(\frac{b(z)}{z}\right)^{1-\alpha} + \gamma_z}{\gamma_z}\right] \equiv g(a(z), b(z), z)$$

Define the vector v:

$$v(z) \equiv \begin{bmatrix} a(z) \\ b(z) \end{bmatrix}$$

---

[20] z is a transformation of potential output per unit of capital. Potential output is that level of output that would be obtained if u = 1.

$$v'(z) = \begin{bmatrix} a'(z) \\ b'(z) \end{bmatrix} = \begin{bmatrix} f(a(z),b(z),z) \\ g(a(z),b(z),z) \end{bmatrix}$$

The following system describes the policy functions and fits exactly into the framework of section III:

$$v'(z) = F(v(z),z)$$

$$\text{s.t.} \quad v(z_\infty) = \begin{bmatrix} a_\infty \\ b_\infty \end{bmatrix}$$

Time Paths

To find time paths, substitute the policy functions in the equation describing the dynamics of z:

$$\dot{z}(t) = \left[\left(\frac{b(z(t))}{z(t)} \right)^{1-\alpha} - a(z(t)) - \delta - \phi \cdot [1 - b(z(t))] \right] \cdot z(t)$$

$$\text{s.t.} \quad z(0) = \frac{k(0)}{h(0)}$$

VII. Concluding Remarks

As a numerical technique, Time-Elimination has two main practical advantages. First, a boundary value problem is transformed into an initial value problem. The "initial" condition is the steady state. As is standard for initial value problems, a computer algorithm will calculate the solution at points that are progressively further from the steady state. Therefore, the first point chosen outside of the steady state is crucial. The second advantage of Time-Elimination is that, unlike shooting backwards in time, this step is expertly performed by a black box (i.e., MATLAB) about which the economist need not worry.

The Time-Elimination Method has proven to be a very simple yet powerful tool for solving dynamic optimization problems. As such a simple tool, it should be in every dynamic macroeconomist's toolbox. Mulligan & Sala-i-Martin (1991) demonstrate that, when Time-Elimination is used together with other tools, interesting economic questions that would ordinarily be considered quite complicated can be answered in short order. At the very least, Time-Elimination's exact solutions can allow us to determine just how approximate our analytical approximations are. Applications to some growth models suggest that the potential output to capital ratio may be an important concept in growth theory. Finally, the amazing speed of the algorithm allows the economist to draw lots of pretty pictures with very little time investment.

References

Mulligan, Casey B. "A Note on the Time-Elimination Method for Solving Recursive Dynamic Economic Models." Manuscript, University of Chicago, September 15, 1991 4:59 PM.

The References for this paper are listed below.

References (for Mulligan (1991))

Jones, L & R. Manuelli. "A Convex Model of Optimal Equilibrium Growth." *Journal of Political Economy*. 98(5) October 1990.

Judd, K. "Minimum Weighted Residual Methods for Solving Dynamic Economic Models." Working Paper, Hoover Institution, July 1990.

Lucas, R.E. "On the Mechanics of Economic Development." *Journal of Monetary Economics*. vol 22, June 1988.

Mulligan, Casey B. & Xavier Sala-i-Martin. "Transitional Dynamics in Two Capital Goods Models of Endogenous Growth." Working Paper, Yale University, September 1991.

Press, William H., et al. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge: Cambridge University Press, 1990.

Ramsey, Frank P. "A Mathematical Theory of Saving." *Economic Journal*. 38 December 1928.

Sala-i-Martin, Xavier. "Lecture Notes on Economic Growth (I & II)." NBER working papers #3563 & #3564, December 1990.

Skiba, A. K. "Optimal Growth with a Convex-Concave Production Function." *Econometrica*. 46(3) May 1978.

Appendix I

MATLAB files that find the policy function for the Ramsey model

```
% RAMSEY.M
% Finds the policy function for the Ramsey model for k < steady state
% c = per capita consumption
% k = per capita capital
% ks = steady state capital

% Initialize parameters
n = .014;           % population growth rate
d = .10;            % depreciation rate of aggregate capital
delta = n + d;      % depreciation rate of k
p = .065;           % rate of time preference
theta = 2;          % (inverse of) intertemporal elas. of substitution
a = .5;             % capital share (production per capita is k^a)
rho = p - n;        % exponent in dynastic utility function

% Calculate steady state values
ks = (a/(rho+delta))^(1/(1-a));
cs = (ks^a - delta*ks);

% Find the policy function
k0 = ks/80; kf = ks;
% MATLAB needs initial conditions, so run backwards from steady-state
% define ssd = ks - k
ssd0 = ks - k0;
ssdf = ks - kf;
global theta rho delta a ks cs
[ssd,c] = ode23('cprime',ssdf,ssd0,cs,.00001);
k = ks - ssd;

% Calculate some interesting functions of k
y = k.^a;                        % Output
s = (y-c)./y;                    % saving rate
ys = ks.^a;                      % steady state output
ss = (ys - cs)/ys;               % steady state saving rate
kdotis0 = y - delta*k;           % those (c,k) for which kdot = 0
gk = k.^(a-1) - c./k - delta;    % capital growth rate
plot(k,c,k,kdotis0)              % Plots policy function c(k) & kdot = 0 schedule
```

```
function cdot = cprime(ssd,c);

k = ks - ssd;
if k == ks
     cdot = -.5*(rho + sqrt(rho*rho+4*cs*(1-a)*a*(ks^(a-2))/theta));
else
     cdot = c.*(rho + delta - a*(k.^(a-1))) / (theta*((k.^a) - c - delta*k));
end      % if k == ks
```

Appendix II

Numerical Solutions with Multiple State-Like Variables

Not all dynamic optimization problems can be studied with only one state-like variable. However, with some additional algebra and coding time, one can find numerical solutions with multiple state-like variables and still enjoy some of the advantages of the Time-Elimination Method. In particular, one will still be dealing with initial value type systems of *ordinary* differential equations.[21] Most of the numerical mathematics will be left to your "black box" (eg, MATLAB).

In section IV, I noted that a "shooting backwards" algorithm was in many ways equivalent to Time-Elimination. The only hangup was the choice of that first crucial point "near to the steady state." Here, I will show how one can obtain that first point from your computer package. One then only needs to shoot backwards from that point. To clarify the exposition, I will assume that there are two state-like variables - y_1 & y_2. The system of differential equations in time is therefore:

$$\begin{aligned}\dot{v}(t) &= q(y_1(t),y_2(t),v(t))\\ \dot{y}_1(t) &= r_1(y_1(t),y_2(t),v(t))\\ \dot{y}_2(t) &= r_2(y_1(t),y_2(t),v(t))\end{aligned}$$

With two state-like variables, a policy function is a *surface*, rather than a curve. Numerically, we must look for a set of sets of triplets (y_1, y_2, v), rather than a single set of ordered pairs (y, v). I will think of each set of triplets as representing a curve that lies on the policy surface. A set of sets of triplets therefore represents a set of curves that lie on the policy surface. Here, each curve will be a possible path that the economy could follow from some initial values of the state-like variables to the steady state. In other words, any given economy must remain on the same curve.[22] As in the one state-like variable case, each curve will be calculated by beginning with a point near to the steady state and then finding those points that are successively farther.

First one must pick a curve to calculate. A curve will be identified by its point that is nearest to (but not equal to) the steady state - let's call it the *near point*. To pick a near point - and therefore a curve - I will pick a function $y_2 = y_2(y_1)$.[23] The values of the two state-like variables for the near point will satisfy this functional relationship. Now, form the system of ordinary

---

[21] There is a tradeoff between human capital, raw labor and computing time. Here I propose to do some additional algebra and spend some extra programming time (ie, use more "raw labor" and a little more computer time) in order to cling to the Maximum Principle and systems of ordinary differential equations (ie, economize on human capital). An alternative approach could take a dynamic programming approach and attempt to numerically solve the resulting system of partial differential equations using Judd's Minimum Weighted Residual techniques.

[22] Note that in the one state-like variable case, there were "two" curves - one for economies for which $y(0) < y_\infty$, the other for economies for which $y(0) > y_\infty$.

[23] It is probably easiest to let these functions be rays originating at the steady state.

differential equations (in y_1):

$$p_i'(y_1) = \frac{d\,p_i(y_1,y_2(y_1))}{d\,y_1} = \frac{\dot{v}_i(t)}{\dot{y}_1(t)} = \frac{q_i(y_1,y_2(y_1),p)}{r_1(y_1,y_2(y_1),p)} \qquad i = 1...n$$

$$\text{s.t.}\quad p_i((y_1)_-) = (v_i)_- \qquad i = 1...n$$

Numerically solve this system and choose the near point of the curve to be estimated to be the near point of the solution of this system.[24] Finally, shoot backwards from the near point to find the rest of the curve. Remember that shooting backwards is the numerical solution to the following system of differential equations in time - an initial value problem:

$$\begin{aligned} \dot{v}(t) &= -\,q(y_1(t),y_2(t),v(t)) \\ \dot{y}_1(t) &= -\,r_1(y_1(t),y_2(t),v(t)) \\ \dot{y}_2(t) &= -\,r_2(y_1(t),y_2(t),v(t)) \end{aligned}$$

$$\text{s.t.}\quad (y_1(T),y_2(T),v(T)) = \text{the near point}$$

The solution to the above system will be a curve that lies in the policy surface.

Repeat this procedure until there are enough curves to appropriately represent the policy surface. If one is interested in the evolution of the economy through time from a particular initial condition rather than the policy function itself, then this repetition can be thought of as a shooting solution to a boundary value problem where the boundary conditions are y = y(0). I still prefer this procedure to shooting from y = y(0) towards the steady state because the steady state is only reached at t = ∞. Also, the shooting is done in two dimensions (the number of state-like variables) rather than (n+2) dimensions (the number of state-like and control-like variables).

[24] Note that the solution here does *not* lie in the policy surface because I have restricted $y_2 = y_2(y_1)$. I am only interested in the near point of the solution and hope that it is sufficiently close to the true policy surface.

Appendix III

N-Sector Balanced Growth and the One State-Like Variable Restriction

Here I related a "balanced growth" restriction to restrictions on the number of state-like variables in an n-sector Cobb-Douglas economy. For this economy, the various capital goods evolve according to:

$$\dot{k}_1(t) = g_1(k_1(t),k(t),u(t)) - c(t)$$
$$\dot{k}(t) = g(k_1(t),k(t),u(t))$$

u(t) is a vector of control variables. k(t) and $g(k_1(t), k(t), u(t))$ are (n-1)x1 vectors:

$$k(t) \equiv \begin{bmatrix} k_2(t) \\ \vdots \\ k_n(t) \end{bmatrix}, \quad g(k_1(t),k(t),u(t)) \equiv \begin{bmatrix} g_2(k_1(t),k(t),u(t)) \\ \vdots \\ g_n(k_1(t),k(t),u(t)) \end{bmatrix}$$

Gross production in each sector is Cobb-Douglas with constant depreciation:

$$g_i(k_1(t),k(t),u(t)) \equiv h_i(u(t)) \cdot \prod_{j=1}^{n} k_j(t)^{\alpha_{ij}} - \delta_i \cdot k_i(t)$$

There is balanced growth when c and k_1 are growing at the same constant rate, all other capital stocks are growing at constant (although possibly different) rates and u is constant. Algebraically, balanced growth requires that a determinant be zero:

$$\begin{vmatrix} (\alpha_{11}-1) & \alpha_{12} & \cdots & \alpha_{1n} \\ \alpha_{21} & (\alpha_{22}-1) & \cdots & \alpha_{2n} \\ \vdots & & & \vdots \\ \alpha_{n1} & \alpha_{n2} & \cdots & (\alpha_{nn}-1) \end{vmatrix} = 0$$

In order for this determinant to be zero, the matrix must have rank less than n. Economically, *the potential output to capital ratio cannot be independent in every sector*. At least one sector's potential output to capital ratio (g_i/k_i when the u's are 1) must be a transformation of the others. Therefore, *there are at most n-1 state-like variables* - the n-1 independent potential output to capital ratios.

As one would expect from the above argument, Mulligan & Sala-i-Martin (1991) find that for two sector models, there is only one state-like variable if there exists a balanced growth path. We can extend this result and say that for (Cobb-Douglas) economies with two nonlinear sectors and n linear sectors, there will be only one state-like variable if there is a balanced growth path -

for any n, however large.[25]

[25] For a linear sector production is $g_i = h_i(u)xk_i$.