The Mathematics of Modern Growth Theory

Stephen Kinsella

Introduction:

Abstract. These notes provide an overview of modern growth theory as it is taught in graduate schools around the world. I provide a Mathematica implementation of the workhorse models of modern growth theory as a pedagogical aid. Further work will focus on transforming these Mathematica 5.0 implementations into Mathematica 6.0 Demonstrations.

Initialisations (Press \[Ctrl+N] \[Enter])

\[n[7]= \text{Off[General::spell, General::spell1]}\]
\[\text{Off[InverseFunction::ifun]}\]
\[\text{Off[Solve::ifun]}\]

Introduction: defining terms

Theories of economic growth are central to most mainstream textbooks in macroeconomics. At one time or another, they have occupied the greatest minds in the discipline. The stakes being played for in the game to get economic growth right are enormous. "Once you start thinking about it, its very hard to think about anything else", to paraphrase Nobel Laureate Robert Lucas. While Lucas was wrong about a great many things, he was correct on this point. If real Gross Domestic Product (GDP) is a measure of the output of an open economy at a certain point in time, then economic growth can be defined as the year on year increase of GDP. Through compounding, a year on year increase in GDP of 5% corresponds to a doubling of living standards every 25 years.

DEFINITION (GDP) Gross Domestic Product is the total market value of the goods and services produced by a nation's economy during a specific period of time.

Some qualifications are in order though. First, the GDP we will talk about is real GDP—it is GDP corrected for the changes in prices over the time period we are measuring. Second, the GDP we will measure is gross, in that this measure includes the replacement of worn out and obsolete equipment and structures in the economy, as well as completely new (or autonomous) investment. We don't use net measures because the information needed to create them is unreliable. Third, we will normally talk about GDP per worker (sometimes called per capita), which is real GDP divided by the number of workers in the economy. This is the most commonly used index of development or growth in the economy, and is part of the Lingua Franca of modern economics, so you'd better know something about it.

Why should a student in Ireland care about economic growth? Well, the fact that you are taking this course is proof-positive that economic growth can affect the wellbeing of a population. Irish economic growth over the past 15 years has been nothing short of miraculous. For details on the Irish economic experience, consult O'Hagan and Newman, (2005), or Honohan, (2002). If you are reading this, you are likely to be twice as well off in terms of living standards than your parents. So, if your parents could afford one annual holiday, you can afford two, on average.

"Wellbeing" can be defined in many ways, and GDP per worker does have its critics, for example Joseph Stiglitz (http://www.argmax.com/mt_blog/archive/000671.php). Stiglitz's main arguments are GDP does not take account of (1) degradation of the environment and use of natural resources, (2) depreciation, and (3) payments to foreigners, and he is not alone.

So GDP per worker is far from a perfect measure, but it is the standard measure of economic wellbeing, and so we must
use it. The table below shows an index of GDP per capita for each of the EU 15 countries and some composites.

Table 1. GDP per capita in Purchasing Power Standards, 2001-2003

<table>
<thead>
<tr>
<th>Country</th>
<th>2001</th>
<th>2002</th>
<th>2003</th>
</tr>
</thead>
<tbody>
<tr>
<td>Luxembourg</td>
<td>213.3</td>
<td>212.6</td>
<td>214.7</td>
</tr>
<tr>
<td>Ireland (GDP)</td>
<td>129.5</td>
<td>132.6</td>
<td>132.5</td>
</tr>
<tr>
<td>Denmark</td>
<td>126.3</td>
<td>122.5</td>
<td>122.6</td>
</tr>
<tr>
<td>Austria</td>
<td>124.4</td>
<td>122.7</td>
<td>122.2</td>
</tr>
<tr>
<td>Netherlands</td>
<td>124.2</td>
<td>122.0</td>
<td>121.0</td>
</tr>
<tr>
<td>United King</td>
<td>151.1</td>
<td>117.8</td>
<td>118.5</td>
</tr>
<tr>
<td>Belgium</td>
<td>117.3</td>
<td>116.7</td>
<td>117.8</td>
</tr>
<tr>
<td>Sweden</td>
<td>116.4</td>
<td>114.8</td>
<td>115.2</td>
</tr>
<tr>
<td>Finland</td>
<td>114.1</td>
<td>113.7</td>
<td>113.7</td>
</tr>
<tr>
<td>Ireland (GNI)</td>
<td>109.8</td>
<td>109.7</td>
<td>111.0</td>
</tr>
<tr>
<td>France</td>
<td>114.8</td>
<td>112.9</td>
<td>111.0</td>
</tr>
<tr>
<td>Germany</td>
<td>110.1</td>
<td>108.7</td>
<td>108.1</td>
</tr>
<tr>
<td>Italy</td>
<td>109.6</td>
<td>109.0</td>
<td>106.9</td>
</tr>
<tr>
<td>EU 25</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
</tr>
<tr>
<td>Spain</td>
<td>92.3</td>
<td>94.6</td>
<td>97.8</td>
</tr>
<tr>
<td>Cyprus</td>
<td>88.8</td>
<td>82.9</td>
<td>81.3</td>
</tr>
<tr>
<td>Greece</td>
<td>73.8</td>
<td>77.7</td>
<td>80.9</td>
</tr>
<tr>
<td>Slovenia</td>
<td>74.8</td>
<td>75.3</td>
<td>76.8</td>
</tr>
<tr>
<td>Portugal</td>
<td>77.2</td>
<td>76.7</td>
<td>74.7</td>
</tr>
<tr>
<td>Malta</td>
<td>74.6</td>
<td>73.8</td>
<td>73.8</td>
</tr>
<tr>
<td>Czech Repub</td>
<td>66.1</td>
<td>67.6</td>
<td>68.8</td>
</tr>
<tr>
<td>Hungary</td>
<td>56.4</td>
<td>58.6</td>
<td>60.5</td>
</tr>
<tr>
<td>Slovakia</td>
<td>48.9</td>
<td>51.3</td>
<td>52.1</td>
</tr>
<tr>
<td>Estonia</td>
<td>44.8</td>
<td>46.6</td>
<td>48.5</td>
</tr>
<tr>
<td>Poland</td>
<td>45.9</td>
<td>45.6</td>
<td>46.0</td>
</tr>
<tr>
<td>Lithuania</td>
<td>40.8</td>
<td>42.4</td>
<td>45.8</td>
</tr>
<tr>
<td>Latvia</td>
<td>37.4</td>
<td>38.9</td>
<td>41.0</td>
</tr>
<tr>
<td>Norway</td>
<td>158.2</td>
<td>149.5</td>
<td>147.7</td>
</tr>
<tr>
<td>Iceland</td>
<td>125.3</td>
<td>119.8</td>
<td>118.7</td>
</tr>
<tr>
<td>Bulgaria</td>
<td>28.6</td>
<td>28.8</td>
<td>29.7</td>
</tr>
<tr>
<td>Romania</td>
<td>28.7</td>
<td>28.6</td>
<td>28.6</td>
</tr>
</tbody>
</table>


And the figure below shows Irish real GDP per capita. This is not a growth rate, however. You're going to figure that out for yourselves in exercise 1.

```mathematica
gdpdata = Import["irlgdp.csv", "csv"]; ListPlot[gdpdata, AxesLabel -> {"Year", "Real GDP Per Capita"}]
```


**A Word about Growth Rates**

We will be talking about rates of growth of measured economic quantities like GDP and the capital stock, so its important you understand what I mean by this term. The rate of growth of GDP is expressed as an index normally, that is, we talk about the difference between the calculated GDP of two years, divided by the 'base' or reference year, and normally expressed as a percentage. If we take two years, 2004 and 2005, and 2004 is the base year, then \( g \), the growth rate of GDP, is given by equation 1 below.
\[ g = \frac{\text{GDP}_{2005} - \text{GDP}_{2004}}{\text{GDP}_{2004}} \] 

Exercise 1

Your first exercise is to calculate the growth rates in the excel sheet (NUIG_MACRO_EX_1.nb) downloadable from http://www.stephenkinsella.net. Answer the following questions.

1. Calculate the rate of growth of the Irish economy, using 1995 as the base year.
2. Graph the change in growth rates against time from 1995 -2005 with time on the x-axis.
3. Is growth accelerating? Based on the evidence you have just created and nothing else, what are your predictions for growth 3 years into the future, from 2005 to 2008?

Mathematical Terms

When I say these words in the lectures, these are the meanings I'm ascribing to them. I won't dwell on these too long, but you need to know them to do well in the course. You'll see references at the end of each definition: in the online version of this lecture note, you can investigate each property further. There are lots of links here to explore; just use the ones you need to get up to speed quickly.

DEFINITION (Vector) Vector means several things to us in this course. A quantity, such as velocity, completely specified by a magnitude and a direction. 2. A one-dimensional array. 3. An element of a vector space. (ref here)

DEFINITION (Matrix) A matrix (plural matrices) is a rectangular table of numbers or, more generally, a table consisting of abstract quantities that can be added and multiplied. Matrices are used to describe linear equations, keep track of the coefficients of linear transformations and to record data that depend on two parameters (ref here).

DEFINITION (Space) A topological space is a set X together with a collection T of subsets of X satisfying the following axioms:

1. The empty set and X are in T.
2. The union of any collection of sets in T is also in T.
3. The intersection of any pair of sets in T is also in T.

The collection T is a topology on X, and the elements of X are called points. The sets in T are the open sets, and their complements in X are the closed sets. The requirement that the union of any collection of open sets be open is more stringent than simply requiring that all pairwise unions be open, as the former includes unions of infinite collections of sets. It follows that a closed set must satisfy the following:

1. The empty set and X are closed (as well as being open).
2. The intersection of any collection of closed sets is also closed.
3. The union of any pair of closed sets is also closed.

By induction, the intersection of any finite collection of open sets is open. (ref here)

DEFINITION (Metric Space) A set where a notion of distance between elements of the set is defined. The metric space which most closely corresponds to our intuitive understanding of space is the 3-dimensional Euclidean space. A metric space induces topological properties like open and closed sets which leads to the study of even more abstract topological spaces (ref here).

DEFINITION (Open Sets) In topology and related fields of mathematics, a set U is called open if, intuitively speaking, you can wiggle or change any point x in U by a small amount in any direction and still be inside U. In other words, if x is surrounded only by elements of U; it can't be on the edge of U.

As a typical example, consider the open interval (0,1) consisting of all real numbers x with 0 < x < 1. Here, the topology is the usual topology on the real line. If you wiggle such an x a little bit (but not too much), then the wiggled version will still be a number between 0 and 1. Therefore, the interval (0,1) is open. However, the interval [0,1] consisting of all numbers x with 0 ≤ x ≤ 1 is not open; if you take x = 1 and move even the tiniest bit in the positive direction, you will be outside of (0,1). (ref here)
DEFINITION (Closed Sets) A closed set contains its own boundary. In other words, if you are "outside" a closed set and you "wiggle" a little bit, you will stay outside the set.(ref here)

Any intersection of arbitrarily many closed sets is closed, and any union of finitely many closed sets is closed. In particular, the empty set and the whole space are closed. In fact, given a set X and a collection F of subsets of X that has these properties, then F will be the collection of closed sets for a unique topology on X. The intersection property also allows one to define the closure of a set A in a space X, which is defined as the smallest closed subset of X that is a superset of A. Specifically, the closure of A can be constructed as the intersection of all of these closed supersets.

DEFINITION (Bounded Sets in Metric Spaces) A set S of real numbers is called bounded above if there is a real number k such that k ≥ s for all s in S. The number k is called an upper bound of S. The terms bounded below and lower bound are similarly defined. A subset S of a metric space (M, d) is bounded if it is contained in a ball of finite radius, i.e. if there exists x in M and r > 0 such that for all s in S, we have d(x, s) < r. M is a bounded metric space (or d is a bounded metric) if M is bounded as a subset of itself. Properties which are similar to boundedness but stronger, that is they imply boundedness, are total boundedness and compactness.

A set S is bounded if it is bounded both above and below. Therefore, a set of real numbers is bounded if it is contained in a finite interval (ref here).

DEFINITION (Banach Fixed Point/Contraction-Mapping Theorem) The Banach fixed point theorem (also known as the contraction mapping theorem or contraction mapping principle) is an important tool in the theory of metric spaces; it guarantees the existence and uniqueness of fixed points of certain self maps of metric spaces, and provides a semi-constructive method to find those fixed points. The theorem is named after Stefan Banach (1892-1945), and was first stated by Banach in 1922. (ref to Full version of proof)

Calculus and Growth Rates

Obviously we will want to use calculus to derive analytical results about models of growth. Calculus should have you thinking about derivatives. One of the assumptions we need to take a derivative of a quantity with respect to another is very small (or infinitesimal) changes. The assumption is made here (and everywhere) that we can do this. So, you will see me trying to find out if the capital stock, K, is growing over time, t. We will need to find out if dK/dt (or K≥0). The way we do this trick of moving from a yearly data point to an infinitesimally small one is by taking a limit, or imagining a decrease in the size of the gap between the two observations (let’s let this gap be given by Δ). Then equation 2 gives us

\[
\lim_{\Delta t \to 0} \frac{K_t - K_{t-\Delta}}{\Delta t} = \frac{\partial K}{\partial t} = \dot{K}.
\]

Having got our instantaneous change, we need to look how that changed relative to what was already there, so our growth rate would be given by \( \dot{K}/K \). We will be using this formalism a lot, so get used to it. We will also be using lots of logarithm transformations, which I detail in the Lagrangian section below. Also, Jones (1998, pp. 167-169) has a good exposition of these rules.

Having defined my terms somewhat, let me take you through three mathematical tools you will need to understand before we touch on models of growth theory in Romer. First, we’ll look at differential equations, then linear programming, then dynamic programming.

Exercise 2

If \( Y = K^{(\alpha)} AL^{(1-\alpha)} \), where Y is output, K is capital employed and AL is technology-augmented (Harrod-neutral) labour, with \( \alpha \) being the share of capital employed in producing output, and assuming constant returns to scale, use this equation to find

1. \( \frac{dY}{dK} \) and \( \frac{d^2 Y}{dK^2} \).
2. \( \frac{dY}{dt} \) and \( \frac{d^2 Y}{dt^2} \).
3. Assume these functions are time dependent, so \( Y[t] = K[t]^{(\alpha)} AL[t]^{(1-\alpha)} \). What is \( dY/dt \)? Is it greater than 0?
Solving Differential Equations for dummies

Now for a little unpleasantness. Dynamical systems use terms like stability in a very specific way, so to get through the solution algorithm, we need to define more terms, and very precisely this time.

We want to investigate and visualize the dynamics of the first-order difference equation

$$ x_{n+1} = f(x_n) \quad (3) $$

where f is given function. We will often refer to such a difference equation as a map f or one-dimensional dynamical system.

**DEFINITION (Orbit).** A positive orbit of $x_0$ is the set of points $\{x_0, f(x_0), f(f(x_0)), \ldots\}$, and is denoted by $O(x_0)$.

**DEFINITION (Equilibrium (Fixed) Point).** The point q is called an equilibrium (fixed) point for f if $f(q) = q$.

**DEFINITION (Periodic Point).** A point p is called a periodic point of minimal period n if $\mathcal{F}^n(p) = f(f(\ldots f(p) \ldots)) = p$ and n is the least such positive integer. The set of all iterates of a periodic point is called a periodic orbit. In other words, a periodic point p of minimal period n is a fixed point of the map $\mathcal{F}^n(x) = f(\ldots f(x) \ldots)$.

The geometric method for visualization of the solutions of our one-dimensional difference equation (3) is called the *stair-step diagram*. You proceed by plotting the pairs $(x_n, x_{n+1})$, where $n = 0, 1, \ldots$ in the standard rectangular coordinate system and connecting the pair of points $(x_n, x_{n+1})$ and $(x_{n+1}, x_{n+2})$ with the segments of the straight line, just like you did for your Junior Cert when drawing graphs. The line segments mentioned above build an impression of stairs. Also, it is important to observe that the fixed points of Eq(3) are the points of intersection of the graph of f with the diagonal $y = x$.

In this economics, we're especially interested in stability of fixed points. It has been shown that, under certain conditions, the stability type of the fixed point q of Eq(3) is the same as the stability type of the fixed point of the corresponding linearized equation:

$$ y_{n+1} = f'(q) y_n \quad (4) $$

The stability of Eq(4) is evident from the stair-case diagrams. This suggests the following linearized stability theorem about a fixed point.

**THEOREM (Linearized Stability).** Let f be continuously differentiable function defined on $\mathbb{R}$. A fixed point q of f is asymptotically stable if $|f'(q)| < 1$, and it is unstable if $|f'(q)| > 1$.

We introduce the following notion for fixed points of Eq(1):

**DEFINITION (Hyperbolic Fixed Point).** A fixed point q of Eq(1) is said to be hyperbolic if $|f'(q)|$ is not equal to 1. From linearized stability theorem, if a fixed point q of Eq(1) is hyperbolic, then it must be either asymptotically stable or unstable and the stability type is determined from $f'(q)$.

As we have seen above, a periodic point p of minimal period n is a fixed point of the map $\mathcal{F}^n$. Consequently, the notion of stability of p follows that of a fixed point and the linearized stability result can be applied to $\mathcal{F}^n$ to determine the stability type of p.

**DEFINITION (Stability and Asymptotic Stability of Periodic Point).** A periodic point p of minimal period n is said to be stable, asymptotically stable, or unstable if p is, respectively, a stable, an asymptotically stable, or an unstable fixed point of $\mathcal{F}^n$. In particular, 2-periodic solution $(p, f(p))$ of Eq(1) is stable if $|f'(p)| < 1$, and unstable if $|f'(p)| > 1$.

The number $|f'(p)|$ is called a *multiplier* of the orbit. Likewise, the multiplier of a periodic orbit of any period n can be defined.

Now that we've defined our terms, let's set about solving a simple, linear differential equation like equation 3 and analysing it.
Basic Idea of Solving Differential Equations

DEFINITION (Differential Equation) A description of how something continuously changes over time. Some differential equations can have an analytical solution such that all future states can be known without simulation of the time evolution of the system. However, most can have a numerical answer, with only limited accuracy.

The basic idea when looking for a solution is to find a function which describes a sequence of values which, when pumped through the differential equation describes all of its behaviors in a predictable way. The general solution is the sequence of values that describes this behavior totally by including some constant, C. There's still work to do because you need to find an initial condition to specify the value of C before you completely understand things. A particular solution will be when you've found (or specified) an initial condition.

Don't worry if none of that made any sense. You'll see what I mean when we do a few examples.

Example 1

\[ x_{n+1} - 2x_n = 0, \quad n = 0, 1, \ldots \]

has the general solution

\[ x_n = C2^n. \]

Where C is our arbitrary constant.

Example 2 (Hirsh, Smale and Devaney, 2004, pp.1-4)

Assume we have some function where \( \frac{dx}{dt} = ax. \)

\( x = x(t) \) is an unknown real-valued function of a real variable, and \( x'(t) \) is its derivative. For us, \( t \) will always mean the function is a function of time. We need to assume that

\[ x'(t) = ax(t) \]

is true.

The solution to this equation is obtained by calculus, the idea being that if \( C \) is an real number, the function \( x(t) = Ce^{ax} \) is a solution, because

\[ x'(t) = aCe^{at} = ax(t), \]

by integration.

And then, to find the exact solution, specify the initial value of the problem

We want to find the constant \( C \) capable of satisfying an initial value problem, so basically we're searching for some \( x'(t) = ax \), such that \( x(0) = 0 \). We'll go through two more examples of this in class.

But for now, here's an economic interpretation of what we are up to.

Linear Example

Setting up the function

The example below is \( ydot + ay = b \), exactly what we did in class.

```
Clear[y, t, a, b]

sol = DSolve[y'[t] + a y[t] == b, y, t];

Dimensions[sol]
{1, 1}
```
The timepath of our little equation

The stability condition here has to be: \( a > 0 \). The parameters are \( A \) (the constant of integration or initial value), \( a \), and \( b \).

\[
\text{Clear}[ydot] \\
ydot[y_, b_, a_] := b - a y
\]

And its phase diagram

Below you see one way to plot the phase. The function and it’s derivative with respect to time.

\[
\{Y[t, 2, 2, 2], D[Y[t, 2, 2, 2], t]\}
\]

Easier, however, is the following: Define \( ydot \) as a function of \( y \).

\[
\text{Clear}[ydot] \\
ydot[y_, b_, a_] := b - a y
\]

Note that the steady state is the location where \( ydot = 0 \).

Stability condition: Derivative less than 0 at the steady state:
Exercise 3

Solve the differential equation

\[ \frac{dy}{dx} + 2xy = 5x^2 - 3. \]

1. What type of differential equation is this?
2. What is the general form of this type of equation?
3. What is the general solution?

Balanced Growth Paths—Solow Style

Now we're up to speed on differential equations, here's the traditional Solow Model. For the development of the basic Solow model, refer to Romer (2004, pages 7--13). Now I want to show you the difference equation version of the Solow model, the starting point for all analyses of economic growth.

```
Clear[k, t, s, d];
eqnl = Simplify[Dsolve[k'[t] == s (k[t])^(1/2) - d k[t], k[t], t]]

\[ \{ \text{k[t] -> } e^{-d t} \left( e^{\frac{d t}{2}} c[1] + e^{\frac{d t}{2}} s \right)^2 \} \]
```

Replace the complicated expression \( e^{\frac{d t}{2}} c[1] \) with a simple variable placeholder, \( c \), as follows:

\[ k1[t_, c_, s_, d_] = k[t] / . \text{First[eqnl] / . } e^{\frac{d t}{2}} c[1] \rightarrow c \]

\[ e^{-d t} \left( c + e^{\frac{d t}{2}} s \right)^2 / d^2 \]

And, we compute \( c \) from the Initial Condition \( k(0)=k0 \).

```
Solve[k1[0, c, s, d] == k0, c]

\[ \{ \text{c -> -d \sqrt{k0 - s}, c -> d \sqrt{k0 - s}} \} \]
```

Then, we replace with \( c \rightarrow d \sqrt{k0 - s} \)

\[ k2[t_, k0_, s_, d_] = k1[t, d \sqrt{k0 - s}, s, d] \]

\[ e^{-d t} \left( d \sqrt{k0 - s} + e^{\frac{d t}{2}} s \right)^2 / d^2 \]

Next we plot several Solution Curves for different values of \( k0 \), and for \( s = 0.3, d = 0.1 \):
Now we fix \(k_0=2\), \(s=0.3\), \(d=0.1\) and define \(k(t)\):

\[
\text{Clear}[k3,t];
\]

\[
k3[t_] = k2[t,2,0.3,0.1]
\]

\[
100. e^{-0.1 t} \left(-0.158579 + 0.3 e^{0.05 t}\right)^2
\]

Then we compute

\[
N[k3[10]]
\]

4.15414

\[
N[k3[100]]
\]

8.936

\[
N[k3[200]]
\]

8.99957

And take the limit of \(k(t)\) as \(t\) goes to infinity to obtain the Steady-State Level:

\[
\text{Limit}[k3[t], t \to \text{Infinity}] = 9.
\]

I can use \texttt{NDSolve} to solve another Solow DE numerically:

\[
\text{Clear}[k, t];
\]

\[
eqn2 = \text{NDSolve}\{\text{k'[t] == 0.3 (k[t])}^{(1/3)} - 0.1 k[t], \\
\quad k[0] == 4, k, \{t, 0, 80\}\}
\]

\[
\{\{k \to \text{InterpolatingFunction}[[\{0., 80.\}, <>]]\}\}
\]

Now we define a function \(k4(t)\) from the last output

\[
k4[t_] := k[t] /. \text{First[eqn2]}
\]

And we may compute, say \(k4[10]\).
Von Neumann, Turnpike Theorems and Linear Programming

John von Neumann (1937) introduced the first linear programming long-run economic growth model into economics at the same time as showing us the concept of a point-to-set mapping, which showed most economists the Brouwer Fixed Point Theorem for the first time. An of this very powerful (if in practice unusable) theorem is the turnpike theorem.

Consider a closed economy where inputs at some point in time are given as outputs at the end of that time, with full employment (the happy suds) and no overproduction. Then as we watch the economy evolving at discrete points, we'll see it grow on an efficient path with constant relative prices, and a constant rate of growth. This is the von Neumann path. von Neumann showed in his 1937 paper that such a path will always exist.

But say you didn't start at the correct initial endowments, which is a very likely starting point. Then the economy must correct itself, starting from endowments $E_0$ to get to $E_\infty$ at the end of $N$ periods. The turnpike theorem says that if $N$ is sufficiently large, so we run the economy for quite a long time, and we can get from $E_0$, $E_1$, $\ldots$, $E_{N-1}$, then $E_\infty$ is an efficient path from the start to the finish, and will lie pretty close to what is now called the von Neumann path. The turnpike theorem shows us an optimal growth path arising from feasible ones.

Neoclassical economists typically represent market equilibrium by drawing an upward-sloping supply curve and a downward-sloping demand curve that intersect at a positive price and quantity, or by drawing a production possibilities curve convex to the origin and indifference curves concave to the origin. For many neoclassical economists the essence of the neoclassical vision was that marginal changes in output and consumption patterns could achieve market equilibrium. The hallmark of this way of thinking is the identification of optima with the equalization of marginal benefit and marginal cost, or of marginal rates of transformation and marginal rates of substitution.

Already before the Second World War some economists had begun to work with linear programs. In this set-up the constraints are linear, not smooth convex functions, and the indifference curves are also linear. As a result marginal rates of transformation and substitution are often not well-defined and optima often occur at extreme points in the decision space, with some variables forced to their zero constraint. Typically this type of model arises when someone wants to operationalize the idea of maximization or optimization in a concrete situation (for example minimizing the cost of a nutritionally adequate diet at given market prices). Using bold letters to denote matrices and vectors rather than scalars, a generic linear program is:
Max \( \nu^T x \) \( \text{subject to} \) \( Ax \leq b. \) 

where \( A \) is an \( n \times m \) matrix of operated processes, \( x \) is an \( m \times 1 \) vector of levels at which the various processes might be carried out, \( \nu^T \) is a (transposed, hence the superscript) \( 1 \times m \) vector of values assigned to the processes, and \( b \) is an \( n \times 1 \) vector of resource availabilities, or constraints. The solution to this problem gives a vector of Lagrange multipliers, \( \lambda \), which solve the following problem:

\[
L(x, \lambda) = \nu^T x - \lambda (Ax - b) = \lambda b - (\lambda A - \nu^T)x.
\]

So you see we have connected the objective function we want to maximise to the resource constraints we must satisfy because they are binding. The idea is that we choose a series of \( \lambda \)'s to get \( \nu^T x \) (which is the value of the processes running in the economy we are looking for) up as high as we can, given the binding constraints represented by \( Ax - b \). The \( \lambda \)'s are called Shadow Prices because they tell us which resource constraints must be satisfied, and also they show us that the resources which are not used at the maximum have a zero shadow price. The first order conditions of equation 4 are

\[
\frac{\partial L}{\partial x} = \nu^T - \lambda A \leq 0
\]

\[
\frac{\partial L}{\partial \lambda} = -(Ax - b) \geq 0
\]

It is important to note that equation 5 is in matrix form, so depending on the size of the matrix, you can be writing out a lot of first order conditions. Also, I'm leaving out what are called complementary slackness conditions for 5 and 6, because for this course you won't need them. All you need to know is that there is a way to set the first order conditions to zero to solve for the Lagrange Multipliers.

For more information on complementary slackness and linear programming in general, see the classic by Samuelson, Solow and Dorfman "Linear Programming and Economic Analysis", Dover Publications, available here, or see the Mathematical Programming Glossary.

Don't worry if this is all Greek to you. We'll be doing a simple Lagrangian in a minute to get the idea across in a more concrete manner, step by step.

The linear programming problem is a special case of the general programming problem. But when there are a large number of resources and a large number of processes, the number of combinations of possible scarce resources and operated processes becomes very large. Thus the linear programming problem emphasizes the essentially combinatorial nature of optimization. In principle if we try all the possibilities the first-order conditions will tell us which one is the optimum. But in a large problem it may take even a very powerful computer too long to try out all the possibilities. For more on this, see Velupillai, (2000), chapters 5 and 8.

### Lagrange Multipliers, step by step

An entire branch of neoclassical economics after World War II was based on Linear Programming, where the planner has some quantity to maximise, let's say social welfare or

#### Problem

Solve the constrained Maximisation Problem

\[
\max y = x_1^{0.25} x_2^{0.75} \text{ subject to } 100 - 2x_1 - 4x_2 = 0
\]
Step One

Create a new variable, L, defined as

$$L = f(x, y) + \lambda g(x, y)$$

Notice that L is obtained by adding the constraint to the objective function and multiplying the constant by a new variable, \( \lambda \), that we've just produced out of thin air. \( \lambda \) is called the Lagrange multiplier, and this equation L is called the Lagrangian expression.

Step Two

We find the unconstrained maximum or minimum of L. To do this, we

1. take all the partial derivatives of the function,
2. set them all equal to zero, and
3. solve them as simultaneous equations.

So, to implement step 2 and its sub steps, L has three partial derivatives because it has 3 variables---x, y, and \( \lambda \). So we find \( \frac{\partial L}{\partial x}, \frac{\partial L}{\partial y}, \) and \( \frac{\partial L}{\partial \lambda} \).

First we evaluate \( \frac{\partial L}{\partial x} \) (using the normal rules of differentiation). we get

$$\frac{\partial L}{\partial x} = f(x) + \lambda g(x) = 0$$

How did I get this? To find \( \frac{\partial L}{\partial x} \), you have to go through the right hand side of the first equation, and

1. differentiate each variable with respect to x,
2. treating the other two variables, y and \( \lambda \), as constants.

So, first, we get \( f(x) \), the partial derivative of \( f(x, y) \) and then we get the multiplicative constant, \( \lambda \). This multiplies the partial derivative of \( g(x, y) \), which we denote by \( g(x) \) here.

Repeat Step 2 for \( \frac{\partial L}{\partial y} \) and \( \frac{\partial L}{\partial \lambda} \). You should now have 3 equations in 3 unknowns, all set to zero.

If you are unsure how to differentiate, read step 3. If you are sure, skip to step four.

Step Three

There are three main rules you need to know to solve the Lagrangian, especially using Cobb-Douglas Utility Functions:

1. Power Rule \( \frac{\partial (a^n)}{\partial a} = n(a^{n-1}) \), e.g. \( \frac{\partial (a^4)}{\partial a} = 5a^3 \)
2. Product Rule

The derivative of the product of two functions is the derivative of the first times the second plus the first times the derivative of the second.

3. Employ Log Rules

   1. \( \log(ab) = \log(a) + \log(b) \)
   2. \( \log\left(\frac{a}{b}\right) = \log(a) - \log(b) \)
   3. \( \log(a^n) = n \cdot \log(a) \)

4. Differentiating Logarithms
derivative of the product of two functions is the derivative of the first times the second plus the first times the derivative of the second.

Step Four
Solve for 3 equations in 3 unknowns, find the maximum or minimum values of x and y, constrained by g(x, y).

Exercise 4
A constrained Maximisation Problem
(I've set it up for you)

\[ \mathcal{L} = x^2 + y^2 + \lambda (10-x-y) \]

for \( \lambda, x \) and \( y \).

Do it now, in class.

Visual Representations of Lagrange Multipliers in action

Lagrange multipliers in action
This section draws heavily on Barry McQuarrie's LMCode.nb notebook, available here. Here I am defining the function, and the constraint condition.

```
Clear[x, y, z];
Off[General::obspkg];
Off[General::newpkg];
<< RealOnly`
On[General::obspkg];
On[General::newpkg];

f[x_, y_, z_] = x^2 + y^3 - z^4;
g[x_, y_, z_] = x^2 + y^2 + z^2;
```

Now it is time to use the Lagrange Multiplier technique to find the extrema. LM is my Lagrange Multiplier.
Solve[{D[f[x, y, z], x] = LMD[g[x, y, z], x],
      D[f[x, y, z], y] = LMD[g[x, y, z], y],
      D[f[x, y, z], z] = LMD[g[x, y, z], z], g[x, y, z] = 1}, {x, y, z, LM}]

Nonreal::warning: Nonreal number encountered.

\[
\begin{aligned}
\{(LM \to -2, x \to 0, y \to 0, z \to -1), &\ (LM \to -2, x \to 0, y \to 0, z \to 1), \\
\{(LM \to -\frac{3}{2}, x \to 0, y \to -1, z \to 0), &\ (LM \to 1, x \to -1, y \to 0, z \to 0), \\
\{LM \to 1, x \to 1, y \to 0, z \to 0\}, &\ \{LM \to 1, x \to -\frac{3}{2}, y \to 0, z \to \text{Nonreal}\}, \\
\{LM \to 1, x \to -\sqrt{\frac{3}{2}}, y \to 0, z \to \text{Nonreal}\}, &\ \{LM \to 1, x \to \sqrt{\frac{3}{2}}, y \to 0, z \to \text{Nonreal}\}, \\
\{LM \to 1, x \to \sqrt{\frac{3}{2}}, y \to 0, z \to \text{Nonreal}\}, &\ \{LM \to 1, x \to -\frac{\sqrt{5}}{3}, y \to -\frac{2}{3}, z \to 0\}, \\
\{LM \to 1, x \to \frac{\sqrt{5}}{3}, y \to \frac{2}{3}, z \to 0\}, &\ \{LM \to 1, x \to -\frac{19}{3}, y \to \frac{2}{3}, z \to \text{Nonreal}\}, \\
\{LM \to 1, x \to -\frac{19}{3}, y \to \frac{2}{3}, z \to \text{Nonreal}\}, &\ \\
\{LM \to 1, x \to \frac{19}{3}, y \to \frac{2}{3}, z \to \text{Nonreal}\}, &\ \{LM \to \frac{3}{2}, x \to 0, y \to 1, z \to 0\}, \\
\{LM \to \frac{3}{2}, x \to 0, y \to 1, z \to 0\}, &\ \{LM \to \frac{3}{16} (3 - \sqrt{73}), x \to 0, y \to \frac{1}{8} (3 - \sqrt{73}), z \to -\frac{9}{32} + \frac{3 \sqrt{73}}{32}\}, \\
\{LM \to \frac{3}{16} (3 - \sqrt{73}), x \to 0, y \to \frac{1}{8} (3 - \sqrt{73}), z \to \frac{9}{32} + \frac{3 \sqrt{73}}{32}\}, \\
\{LM \to \text{Nonreal}, x \to 0, y \to \text{Nonreal}, z \to \text{Nonreal}\}, &\ \\
\{LM \to \text{Nonreal}, x \to 0, y \to \text{Nonreal}, z \to \text{Nonreal}\}
\end{aligned}
\]

Lots of solutions present themselves, even for this simple problem. I used the RealOnly package since I am not interested in any complex valued solutions. Now I have to evaluate the function at all the points that I found above.
The minimum values in the above list are the minimum of the function subject to the constraint. So, we have a minimum of \(-1\) which occurs at the points \((0, 0, -1), (0, 0, 1),\) and \((0, -1, 0)\). We also have a maximum of \(+1\) which occurs at the points \((-1, 0, 0), (1, 0, 0),\) and \((0, 1, 0)\).

Let's generate some graphs to see what this means in terms of level surfaces. First, we need our constraint surface, which is just a sphere of radius one centered at the origin.
constraint = ContourPlot3D[g[x, y, z], {x, -1, 1}, {y, -1, 1},
{z, -1, 1}, Contours -> {1},
Axes -> True, Lighting -> Automatic, ContourStyle -> {RGBColor[0, 1, 0]}]

Now, we need the level surfaces of the function. First, let's generate the level surface for the maximum, which occurs when \( f(x, y, z) = +1 \).
We know that graphically, what must be happening is the tangent plane to the constraint surface must be the same as the tangent plane to the level surface of the function at the points where they touch. From the above diagram, we can
We know that graphically, what must be happening is the tangent plane to the constraint surface must be the same as the tangent plane to the level surface of the function at the points where they touch. From the above diagram, we can see that this occurs for the points \((-1,0,0), (1,0,0), \) and \((0,1,0),\) as expected. What would happen if we decreased the level surface of the function slightly? We would expect the constraint surface to "poke through", and they would not have the same tangent planes where they touched. Let's try it.

```
levelsurface1 = ContourPlot3D[f[x, y, z], {x, -2, 2}, {y, -2, 2}, {z, -2, 2}, Contours -> {0.8}, Axes -> True, Lighting -> Automatic, ContourStyle -> {RGBColor[1, 0, 0]}]
Show[levelsurface1, constraint, ViewPoint -> {5, -5, 2}, AxesLabel -> {x, y, z}];
```

The constraint curve is poking through the side. Although we can't see it, we know that it is also poking through in the back and on the other side as well. Now at any point of contact between the two surfaces the tangent planes are not the same. If we had increased the level surface of the function to \(f(x, y, z) = k, \) \(k > 1,\) then the surfaces would not touch.

Let's look at the minimum level surface now, where \(f(x, y, z) = -1.\)
levelsurfacemin = ContourPlot3D[f[x, y, z], {x, -5, 5}, {y, -5, 5}, {z, -5, 5}, Contours -> {-1}, Axes -> True, Lighting -> Automatic, ContourStyle -> {RGBColor[1, 0, 0]}]
Here the level surface of the function is encompassing the surface for the constraint. We can see that the two surfaces touch at the points \((0, 0, -1), (0, 0, 1),\) and \((0, -1, 0),\) and also have the same tangent plane at those points. If we increased the level surface of the function, we would again find that the constraint surface would poke through the surface. If we decreased the level curve, then the two surfaces would not touch.

As for the extraneous values, let's generate a plot of the level curve for one of them. The 0.851 is really 23/27, so we need a level surface for the function at this value.
We see that the constraint surface and the level surface do indeed meet at a point that has a common tangent plane—however, the constraint surface is poking through elsewhere. This tells us that although the conditions of the Lagrange
We see that the constraint surface and the level surface do indeed meet at a point that has a common tangent plane—however, the constraint surface is poking through elsewhere. This tells us that although the conditions of the Lagrange multiplier method have been satisfied, we have not identified a maximum or minimum. The extrema occur when the surfaces touch, have common tangent planes where they touch, and do not intersect anywhere else.

We can do a similar plot to see this happening for the level curve when \( f(x, y) = -0.609 \).

```math
levelsurface2 = ContourPlot3D[f[x, y, z], {x, -2, 2}, {y, -2, 2}, {z, -2, 2}, Contours -> {-0.602954}, Axes -> True, Lighting -> Automatic, ContourStyle -> {RGBColor[1, 0, 0]}]

Show[levelsurface2, constraint, ViewPoint -> \{-\frac{1}{2}, -1, \frac{1}{3}\}, AxesLabel -> \{x, y, z\}]
```
Just because I'm a geek, let's get the cool rotational version of this last sketch. Right click and drag the image to rotate it.
levelsurface2 = ContourPlot3D[f[x, y, z], {x, -2, 2}, {y, -2, 2}, {z, -2, 2}, Contours -> {-0.602954}]
Dynamic Programming

It also seems reasonable to postulate an interdependence between the variables entering an economic system in the case concerning the determination of the conditions for correctly anticipated processes. These conditions are that the individuals have such expectations of the future that they act in ways which are necessary for their expectations to be fulfilled. It follows that the interdependence between present and future magnitudes is conditioned in this case by the fact that the latter, via correct anticipations, influence the former. If we also choose to describe such developments as equilibrium processes, this implies that we widen the concept of equilibrium to include also economic systems describing changes over time where the changes that take place from period to period do not cause any interruption in, but, the contrary, are an expression of the continual adjustment of the variables to each other.

Lindahl (1954), p.27.

The above quote is the best example I can find of an intuitive explanation for why dynamic programming is needed in neoclassical economics. We can define dynamic programming as "a method of solving multi-stage problems in which the decisions at one stage become the conditions governing the succeeding stages" (www.indiainfoline.com/bisc/jama/jmnld.html), but what we need to see is the economic rationale for the technique. We use dynamic programming because the economic problem is to maximise the value of the processes run in the economy through time. So, we need a method that performs the Lagrangian analysis I've just shown you at various intervals throughout time. The solution should be expressed as a function of time.

The economic problem faced by neoclassical economists is: what processes should be run now and later in order to maximise the value of resources used throughout the lifetime of the economy. The way to do this is to produce a functional that can iterate each shadow price over time to get the optimal solution. The planner interested in optimal growth must thus form the Hamiltonian.

The Hamiltonian technique performs exactly the same role of converting constrained extrema problems to unconstrained extrema ones as the Lagrange multiplier shown above, but it does it dynamically, in the sense that it captures interactions of past values of the constraint (now called the control) variable(s) with the present and future values of the objective (now called state) variable(s) via the use of an operator, exactly like the familiar $\lambda$, but here called the costate variable(s). The Lagrangian $\lambda$ finds the maximum value of the objective function which satisfies the constraint in each single period, while the costate variable attaches a value to the next period's production/consumption/investment, which must be zero at the end of time. To set up a Hamiltonian problem, therefore, we require a description of the state of the system at some initial point, called the initial condition, and a description of the end of the system, called the transversality or boundary condition, which I will say more about in the lecture. Then, we need a function which assigns values to the state variable at each moment, given the value of the control. So, for the (Ramsey-esque) case of maximizing utility subject to a consumption and labour force constraint, the set up of the problem will go something like this.

**Hamiltonians, a Step by Step Guide**

**Step One. Form the Hamiltonian**

$$H = \text{Max} \int_0^\infty e^{-\rho t} L(t) \frac{u[C(t)]}{n(t)} \, dt,$$

With $L(t)$ being the labour force, $C(t)$ its consumption, $n(t)$ the growth rate of the population, and $\rho$ a measure of the discount rate. What this does is to set the pure discount rate, $\epsilon$, proportional to an increasing weight of the level of a future population.

Intuitively, this means that when the population is bigger, utility will increase by the number of people existing in that later period. This is quite a Benthamite argument, in my view. Let's allow our initial labour force to be one, and cap population growth at $n(t) = \bar{n}$. Things aren't that rosy, though, as we have to consider the level of (per capita) capital accumulation at any time, which is given by

$$H = \int_0^\infty e^{-\rho t} n(t) \, dt,$$

**s.t. $k < f(k(t)) - C(t) - nk(t).$**

(Think of this as a budget constraint for the State variable )
We have our initial (or boundary) condition,

\[ k(0) = k_0, \]

And our transversality condition,

\[ \mu(t) = 0. \]

**What does the transversality condition mean?** Imagine you wish to optimally save over some finite horizon, say the time of your life. Will you save in the last period? Most likely not. The transversality condition supplies another boundary, where the costate variable returns a value to the previous period of zero. The rocket, if the control variable is fuel fuelling a rocket, and the state variable is the velocity and position of the rocket should run out of fuel just as it hits its target and goes boom.

### Step Two

We partially differentiate \( H(\bullet) \) with respect to the costate and state variables. The punchline before the joke is told is: in order to get the integrand to converge at all, the profit rate must be greater than the level of population growth. This has enormous implications. This analysis uses Pontryagin's maximum and minimum principles, which ensure convergence when \( \eta = \epsilon \cdot \tau, \ \eta > 0 \). Forming the Hamiltonian in the same way as the Lagrangian, we get

\[ H = e^{-\eta} u(C) + \mu(f(k(t) - C(t)) - n k(t)). \]

### Step Three

Take the partial derivatives of this equation to find that

\[
\frac{\partial H}{\partial C} = 0 \Rightarrow u' - \eta \mu, \\
\dot{u} - \frac{\partial H}{\partial k} = -u(f' - n)
\]

We get a differential equation for the costate variable \( \mu \), which is paired with

\[ \dot{k} = (f(k(t)) - C(t)) - n k(t). \]

These are Euler (pronounced 'Oy-Ler') equations which give us a 2*2 system of differential equations, solvable in the usual way. We'll look at the 'usual way' in the next section.

Points to note are:
1. equation (6) is running backward in time, because of the minus sign.
2. equation (7) is running forward in time.

So, Hamilton has given us a system running both backwards and forwards in time, a 2-point boundary value problem, expressed in (usually finite) time. The \( \mu \) values are transmitting information backward in time about what the resource situation is there. It turns out that the system is most efficient computationally when both are allowed to move forward and backward in time.

### Steps Four and Five

To 'close' the model, all we need to do is specify the shape of the utility function (usually Constant Elasticity of Substitution, CES) as we do below in several examples, and step five, we assess the stability of the model, meaning we try to determine in what range and over what domain the function varies, how it varies, and what we might expect it to do given some arbitrary input of values--converge to a point, cycle endlessly, or explode, or a combination of these. Setting the value of our utility function to equal something like

\[ u(C(t)) = \begin{cases} \frac{C^\rho}{1-\rho}, & \rho \neq 1 \\ \text{Log}(C), & \rho = 1 \end{cases}. \]

Then if Log[C] pertains, we have a Bernoulli utility function, where \( 1/\rho \) is the elasticity of substitution between two points in time. Intuitively, if \( \rho \approx 0 \), then we don't care about the present (we are immortal, anyway). You can also show
that

\[ \epsilon + \rho \hat{C} = \hat{f}, \]

which says (dropping the t-subscript for clarity) that the pure discount rate and the growth rate of consumption per head \( \hat{C} \), all weighted by \( \rho \) is equal to \( \hat{f} \), the marginal product of capital. This is the Keynes – Ramsey rule, which means that the economy can account for and solve the problem of different wants and provisions, as Bohm-Bawerk put it, by seeing how fast utility per person is changing over time, and adjusting the speed of production to accommodate this. The importance of such a finding does not need much comment, as it should be obvious.

There are problems, however. Under realistic descriptions of utility and uncertainty—stochastic income and habit formation—these intertemporal problems are very difficult to solve. As Velupillai (2000, ch 8) shows, these techniques are based upon mathematical machinery called Pontryagin Maxima, which have serious flaws in their constructive interpretations. All of this is a long winded way of saying these models are hard to compute except under very restrictive assumptions.

Optimizing agents must build up precautionary savings to buffer bad income realizations, and must anticipate the negative "internality" of current consumption on future utility, through habits. Here we will look at an unrealistic, 1960's version of the system of optimal growth studied by Cass and Koopmans, the exposition of which is in Barro & Sala-i-Martin, 2004, chapter 2, as well as Romer, 2004, chapter 2.

It is historically significant that the first model of this type was studied by the Cambridge philosopher F.P. Ramsey (1929), using a technique called the calculus of variations. Ramsey's idea was that as societies advanced through technological progress, there would be such a build up of the capital stock, causing a falling rate of profit, that capital accumulation as an activity would just go away as the profit motive could not be satisfied at such low rates of profit growth for such high levels of capital input. Ramsey's steady state is thus not a steady state as we understand it today, but rather a 'state of bliss', wherein the cultural context for capital accumulation is destroyed by the existence of overweening wealth. We would say these lucky souls have their utility functions saturated, thus consumption growth flattines. This idea is formally equivalent with Keynes' 'Euthanasia of the Rentiers' notion, discussed in his general theory.

"you can't just draw a demand and supply curve for capital and apply the usual apparatus of saver's rent etc.; e.g. the treatment of saving as a use of income with its own elasticity of demand (as in Pigou Public Finance p. 138) is not really right. I did a very elaborate treatment of taxation and savings which was cut out by Maynard; rightly as it was too involved in comparison with the conclusions which were feeble.

I first started thinking about saving through being dissatisfied with Pigou's treating it in this way; but now I think what he says is good enough perhaps, as anything better would be so complicated and fruitless." —Letter from Frank Ramsey to Roy Harrod, (1929)

Below, in the optimal Cass-Koopmans growth model we will study, the utility does not saturate at all, but rather increases without bound. As we shall see, another element 'closes' the system for the lucky Neoclassical modeler, willfully (or blissfully: it depends) ignorant of computability considerations.

Exercise 5

There is a very large number of households with perfect foresight, ordered on the [0,1] interval. A given household, of size \( L \), will grow at a rate \( n \). Each person is endowed with one unit of labour per period of time, and the household income of the model is made up of labour income (obviously dependent on the wage rate, \( w \)) plus the net real rate of return \( (r-\delta) \) on the capital they employ per period, \( K \). Each household wants to maximise the value of an infinite utility stream, \( \int_0^{\infty} u (c) \), with each generation taking into account the welfare of its descendants (via the transversality conditions we derived in class. Form the Maximisation problem.

Examples of Dynamic Programming: the Ramsey-Cass Koopmans Model

Assume a representative agent household. A representative agent is one whose preferences are copied throughout the economy, thus we need only look at one household in the aggregate. Our 'macro economy' thus consists of a simple addition of the outputs of a continuum of identical households of size \( L \), which we will allow grow at a rate \( n \). Each person is endowed with one unit of labour per period of time, and the household income of the model is made up of labour income (obviously dependent on the wage rate, \( w \)) plus the net real rate of return \( (r-\delta) \) on the capital they employ per period, \( K \). Each household wants to maximise the value of an infinite utility stream, with each generation taking into account the welfare of its descendants via transversality conditions in exactly the way Ramsey (1929) described to Harrod as being incorrect (see quote above). We can express this intertemporal maximisation as:
The utility function being maximised is usually of the Constant Elasticity of Substitution (CES) form, modified with intertemporal assumptions (which is the cause for Ramsey's misgivings), so it looks like:

\[ u(c) = \frac{c^{1-\sigma} - 1}{1-\sigma}, \text{ with } c, \sigma > 0 \]

Table of Variables and their meanings

<table>
<thead>
<tr>
<th>Variable</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>Total Utility of household</td>
</tr>
<tr>
<td>c</td>
<td>level of consumption</td>
</tr>
<tr>
<td>K</td>
<td>capital stock per period</td>
</tr>
<tr>
<td>r</td>
<td>return on capital (rental price)</td>
</tr>
<tr>
<td>δ</td>
<td>capital depreciation</td>
</tr>
<tr>
<td>ρ</td>
<td>elasticity of demand wrt consumption</td>
</tr>
<tr>
<td>-σ</td>
<td>elasticity of utility wrt consumption</td>
</tr>
</tbody>
</table>

What does this look like? Here I plot three utility functions with different values of \( \sigma \) for comparison.

\[
\sigma 1 = 0; \sigma 2 = 1.1; \sigma 3 = 3;
\]

\[
\text{Plot}\left[\left\{\frac{c^{1-\sigma_1} - 1}{1-\sigma_1}, \frac{c^{1-\sigma_2} - 1}{1-\sigma_2}, \frac{c^{1-\sigma_3} - 1}{1-\sigma_3}\right\}, \{c, 0, 5\}, \right.
\]

\[
\text{PlotStyle} \rightarrow \{(\text{Thickness}[0.001], \text{RGBColor}[1, 0, 0]),
\]

\[
\{\text{Thickness}[0.001], \text{RGBColor}[0, 0, 1]\}, \text{Thickness}[0.001],
\]

\[
\text{RGBColor}[1, 0, 0]\}, \text{PlotRange} \rightarrow \{-2, 2\}, \text{AxesLabel} \rightarrow \{"c", "u(c)"\}\]

The marginal utility of consumption is \( u'(c) = c^{-\sigma} \). The elasticity of the marginal utility relative the amount consumed is \( -\sigma \). To get our maximisation on, though, we need to set up a current value Hamiltonian (Here we are doing this according to Shell's article on Hamiltonians in the New Palgrave, referenced below):

\[
H(c, K, \lambda) = \frac{c^{1-\sigma_1} - 1}{1-\sigma_1} + \lambda [(r - \delta)K + WL - cL]
\]

(11)

Derive the first order conditions of the system in the usual way to obtain:

\[
H_c = c^{-\sigma} - \lambda L = 0 \iff C^{-\sigma} = \lambda L^{1-\sigma}
\]

(12)
\[
\dot{\lambda} = -H_K + \rho \lambda = -\lambda(r - \delta) + \rho \lambda \iff \frac{\dot{\lambda}}{\lambda} = \rho - r + \delta
\]

\[
\dot{K} = H_K = (r - \delta)K + wL - C
\]

\[
\lim_{t \to \infty} e^{r t} \lambda(t) K(t) = 0 \iff -\rho + \lim_{t \to \infty} \dot{\lambda} + \lim_{t \to \infty} \dot{K} < 0
\]

Now just differentiate $C^{-\sigma} = \lambda L^{1-\sigma}$ (called the necessary condition) with respect to time and sub in the fact that $\frac{\dot{\lambda}}{\lambda} = \rho - r + \delta$, (called the costate equation) and we have the Ramsey-Cass-Koopmans rule of optimal consumption:

\[
\dot{C} = \frac{C}{\sigma} [r - \delta - \rho - (1 - \sigma) n]
\]

### So What?

So what? Who cares about this measure? What does it mean? Firstly, is says that the household's borrowing solution is simple: just borrow up until $u'(c) = 0$, which is a restatement of the Keynes-Ramsey Rule derived and described above. Second, When one uses the Hamiltonian method to consider optimal production as well, there is a significant policy proposition: the Hamiltonian method implies that policy over the (infinitely) long run is inefficient, because it would take away resources that might have been used to jump the economy onto its balanced growth path.

The stationary solution of the system is defined by $C = K = 0$. This stationary solution gives the level of the maximal balanced level of utility over time, constrained by $w$, $L$ and $r$.

```mathematica
variables = {C, K};
equations = \{(r - \delta) * K * w * L - C = 0, \frac{C}{\sigma} * (r - \delta - \rho - (1 - \sigma) * n) = 0\};
sol = Solve[equations, {C, K}] // FullSimplify;
sol1 = sol[[1]];
statpoint = {csol, ksol} = variables /. sol

\{0, \frac{L w}{r + \delta}\}
```

### Stability Analysis of the Pure Ramsey Model

As you can see from reading Romer (2006), the Ramsey model of economic growth is a workhorse of contemporary macro-economics. It is the starting point not only for growth theory but also for real business cycle theory. Ask an economist how the economy will react to an increase in government purchases or to a change in the tax rate on capital and the first model he or she will reach for in a search for answers is the Ramsey model.

The model requires that we solve two paired differential equations with a boundary condition. The solution methods used, the shooting and time elimination method, are of wide applicability and may be of interest to researchers in fields other than economics.

Since the Ramsey model is well known I will focus on how Mathematica can be used to solve the model. Introductions to the Ramsey model can be found in the texts of Romer (1996), Barro and Sala-i-Martin (1995), and Blanchard and Fischer (1989) and of course the original articles by Ramsey (1928), Cass (1965), and Koopmans (1965). My notation again follows Romer's.

Assume a large number of identical firms. Each firm uses capital, $K$, and labour, $L$ to produce output $Y$, according to the production function $Y = F(K, A \times L)$. The parameter $A$ measures the state of "technology." Technology allows $L$ workers to produce as if there were actually $A \times L$ workers. We will assume that $A$ grows exogenously at the rate $g$, and simply designate the parametrized labour variable by $AL$.

Profit-maximizing firms hire capital and labour in competitive markets which they use to produce output. The
The price of output is normalized to 1 which implies that a firm's profit function can be written as below where \( w \) is the wage rate of effective labour, \( r \) is the interest rate and \( \delta \) the depreciation rate on capital. To be competitive with other assets capital must pay its owners a return of \( r + \delta \) which is therefore the rental rate or interest rate on capital.

\[
\text{profit} = F[K, AL] - w AL - (r + \delta) K;
\]

To maximize profit the firm hires capital and labour until the first derivatives of the profit function with respect to capital and labour respectively are zero.

\[
\begin{align*}
D[\text{profit}, K] &= 0 \\
-r - \delta + F^{(1,0)} [K, AL] &= 0
\end{align*}
\]

\[
\begin{align*}
D[\text{profit}, AL] &= 0 \\
-w + F^{(0,1)} [K, AL] &= 0
\end{align*}
\]

We will also assume that the production function \( F \) is linearly homogeneous or in economic terms that it exhibits constant returns to scale (CRS). The CRS assumption means that if inputs increase by a factor of \( c \), output will also increase by a factor of \( c \), \( F[cK, cAL] = cF[K, AL] \). Euler's theorem says that if \( F[K, AL] \) is linearly homogeneous, then \( Y = F_K K + F_{AL} AL \). Using the above first-order conditions, Euler's theorem implies that factor payments exhaust the product, \( Y = (r + \delta) K + w AL \). Setting \( c = 1/AL \) allows us to write the production function in terms of one variable, capital per unit of effective labour \( k = K/AL \). Thus, output per unit of effective labour can be written \( Y/AL = y = f[k] \). Rewriting the first-order condition for capital we have, as before

\[
r + \delta = f'[k]
\]

and using (1) and Euler's theorem

\[
w = f[k] - kf'[k].
\]

Household behavior is more complicated. There are a large number, \( H \), of identical households. Each member of the household supplies 1 unit of labour at every point in time. Households own the capital stock, which they rent to firms. Each household begins with capital holdings of \( K[0]/H \) where \( K[t] \) is the amount of capital at time \( t \). At each moment in time the household chooses how to divide its income - from wages and the renting of capital - between consumption and savings in order to maximize lifetime utility. The household's lifetime utility is written:

\[
U = \int_0^\infty e^{-\rho t} u(C[t]) \frac{L[t]}{H} dt
\]

where \( C[t] \) is the consumption of each household member at time \( t \). \( u[C[t]] \) is the instantaneous utility created by the consumption of \( C[t] \) at time \( t \). \( L[t] \) is the total population of the economy and \( L[t]/H \) the number of members of the household at time \( t \). \( \rho > 0 \) is the household's discount rate (assumed the same for all households). The higher is \( \rho \) the more the household discounts future consumption relative to current consumption. It is common to assume that instantaneous utility is of the form \( u(C[t]) = \frac{C[t]^{1-\theta}}{1-\theta} \), with \( \theta > 0 \) and \( \rho - n - (1-\theta)g > 0 \). \( \theta \) governs how willing individuals are to substitute consumption from one time period into another. A lower \( \theta \) implies a greater willingness to substitute consumption tomorrow for consumption today. The condition \( \rho - n - (1-\theta)g > 0 \) is a technical condition needed to ensure that utility is bounded below infinity. It's convenient to write the budget constraint in terms of consumption per unit of effective labour. After some algebraic manipulations and taking into account the fact that \( L \) and \( A \) are growing we have (see Romer (1996) for the derivation).

\[
U = B \int_0^\infty e^{-\rho t} \frac{c[t]^{1-\theta}}{1-\theta} dt
\]
where $B = A[0]^{1-\theta}L[0]/H$ and 
\[ \beta = \rho - n - (1 - \theta) g. \]

B is an unimportant constant and will be normalized to 1 in what follows.

The household faces two constraints when maximizing utility.

\[ k(t) = r(t) k(t) + e^{(\rho - g) t}(w(t) - c(t)) \]

\[ \lim_{t \to \infty} e^{-R(t)} e^{(\rho - g) t} k(t) \geq 0, \]

where $R(t) = \int_{0}^{t} r(\tau) d\tau$.

The first constraint is closer to an accounting identity than a constraint it that says that the growth in the household's capital stock (per unit of effective labour) is equal to the interest earnings on the current capital stock plus the difference between the household's wages in period $t$ and its consumption in period $t$ (the exponential term adjusts for growth in effective labour). Notice that 1) does not forbid the household from consuming more than its wages by borrowing (creating a negative capital stock). It follows that the optimal solution to the household's problem is to borrow an infinite amount and live it up! Since the latter strategy is unrealistic, we need constraint 2, which says that the household's capital stock (adjusted per unit of effective labour) cannot be negative in the limit. In other words, the household must live within its means, if not on any given day then in the limit.

The household's problem has been set up as a standard problem in optimal control theory that can be solved using the method of Hamiltonians (see the references section or Kamien and Schwartz, 1981 for an introduction to optimal control theory). Write the Hamiltonian as:

\[ H = e^{-\beta t} \frac{c(t)^{1-\theta}}{1-\theta} + \lambda(t) \left( r(t) k(t) + e^{(\rho - g) t} (w(t) - c(t)) \right) \]

Optimal control theory tells us that the solution to our maximization problem must satisfy $H_c = 0$, $H_k = -\frac{\partial H}{\partial \lambda}$ and the transversality condition $\lim_{t \to \infty} \lambda(t) k(t) \to 0$. (It can be shown that the transversality condition implies that the no infinite debt constraint of the household's problem will be satisfied in equilibrium, see Barro and Salai-i-Martin, 1995).

\[ H_c = D[H, c(t)] = 0 \]

\[ e^{-\beta t} c(t)^{1-\theta} - e^{(\rho - g) t} \lambda(t) = 0 \]

\[ H_k = D[H, k(t)] = -D[\lambda(t), t] \]

\[ r(t) \lambda(t) = -\lambda'(t) \]

We now rearrange the first-order conditions to find a particularly convenient representation. First solve $H_c$ for $\lambda(t)$ then differentiate with respect to $t$.

\[ \text{soll} = \text{Solve}[H_c, \lambda(t)] // \text{Simplify} \]

\[ \{ \lambda(t) \to e^{-t (g - n - \beta)} c(t)^{-\theta} \} \]

\[ \text{sol2} = D[\text{soll}, t] \]

\[ \{ \lambda'(t) \to e^{-t (g - n - \beta)} \} \]

\[ (-g - n - \beta) c(t)^{-\theta} - e^{-t (g - n - \beta)} \theta c(t)^{-1-\theta} c'(t) \]
Now divide sol2 by sol1 and simplify.

\[
\text{tmp} = \frac{\text{sol2}[[1, 1, 1]]}{\text{sol1}[[1, 1, 1]]} = \frac{\text{sol2}[[1, 1, 2]]}{\text{sol1}[[1, 1, 2]]} // \text{Simplify}
\]
\[
g + n + \beta + \frac{\lambda'[t]}{\lambda[t]} = 0
\]

From the first-order condition for \( H_t \) we can make the substitution \( \frac{\lambda'(t)}{\lambda(t)} \rightarrow r[t] \) and from the definition of \( \beta \) we can make the substitution \( \beta \rightarrow \rho - n - (1 - \theta)g \).

\[
\text{tmp} = \text{tmp} /. \frac{\lambda'(t)}{\lambda(t)} \rightarrow -r[t]
\]
\[
g + n + \beta - r[t] + \frac{\theta c'[t]}{c[t]} = 0
\]

\[
\text{Solve}[\text{tmp}, c'[t]]
\]
\[
\{(c'[t] \rightarrow -\frac{c[t] (g + n + \beta - r[t])}{\theta})\}
\]

\[
\% /./ \{\beta \rightarrow \rho - n - (1 - \theta) g \} // \text{Simplify}
\]
\[
\{(c'[t] \rightarrow -\frac{c[t] (g \theta + \rho - r[t])}{\theta})\}
\]

Rearranging we have:

\[
\text{tmp} = \frac{c'[t]}{c[t]} = \frac{(r(t) - \rho - g \theta)}{\theta}
\]

\[
\frac{c'[t]}{c[t]} = \frac{-g \theta + \rho + r[t]}{\theta}
\]

Using the fact that in equilibrium, \( r[t] = f'[k[t]] - \delta \) we have:

\[
\text{tmp} = \text{tmp} /. r[t] \rightarrow f'[k[t]] - \delta
\]

\[
\frac{c'[t]}{c[t]} = \frac{-\delta - g \theta - \rho + f'[k[t]]}{\theta}
\]

The dynamics of \( k \) are determined from the accounting identity that changes in the capital stock equal production minus consumption minus depreciation (with suitable adjustments for changes in the stock of effective labour).

\[
k'[t] = f[k[t]] - c[t] - (n + g + \delta) k[t]
\]
\[
k'[t] = -c[t] + f[k[t]] - (g + n + \delta) k[t]
\]

The two key equations of the Ramsey model are thus:

1. \( c'(t) = c(t) \left( \frac{f'(k(t)) - \delta - \rho - g \phi}{\phi} \right) \) and
2. \( k'(t) = f(k(t)) - c(t) - (g + n + \delta) k(t) \)
Numerically Solving the Model

We now turn to numerically solving the model for steady states and transition paths. To do so we specify that the production function is of the Cobb-Douglas form, \( f(k) = k^\alpha \). Now define CPrime and KPrime as follows:

\[
\begin{align*}
\text{In}[20] &:= \text{Clear}[\alpha, \delta, g, \rho, \theta, x, y, a, b, k, t] \\
\text{In}[21] &:= \text{CPrime}[\alpha_, \delta_, g_, \rho_, \theta_] := c[t] (\alpha k[t]^{\alpha - 1} - \delta - \rho - g \theta) / \theta \\
\text{In}[22] &:= \text{KPrime}[\alpha_, \delta_, g_, n_] := k[t] - (g + n + \delta) k[t] \\
\end{align*}
\]

We initially set \( \alpha=1/3, \ \delta=.05, \ g=.02, \ n=0.01, \ \rho=.02, \text{ and } \ \theta=1.75 \). We now solve for the steady state, the levels of \( c \) and \( k \) such that \( c'[t]=k'[t]=0 \).

\[
\begin{align*}
\text{In}[23] &:= \text{kbar1} = \\
&= k[t] /. \text{Flatten@\text{Solve}[\text{CPrime}[1/3, 0.05, 0.02, 0.02, 1.75] == 0.0, k[t]]} \\
\text{Out}[23] &:= 5.65632 \\
\text{In}[24] &:= \text{cfunct} = c[t] /. \text{Flatten@\text{Solve}[\text{KPrime}[1/3, 0.05, 0.02, 0.01] == 0.0, c[t]]} \\
\text{Out}[24] &:= -1.0 \left( 0.01 - 1.0 \cdot k[t]^{1/3} + 0.08 \cdot k[t] \right) \\
\text{In}[25] &:= \text{cbar1} = \text{cfunct} /. k[t] \to \text{kbar1} \\
\text{General::spell1} &:= \text{New symbol name "cbar1" is similar to existing symbol "kbar1" and may be misspelled.} \Rightarrow \\
\text{Out}[25] &:= 1.32924 \\
\text{In}[26] &:= \text{pl = \text{Plot}[\text{cfunct} /. k[t] \to k, \{k, 0.01, 10\}, \text{DisplayFunction} \to \text{Identity}];} \\
\text{Out}[26] &:= \\
\text{The next figure shows the steady state levels of capital and consumption.}
\]

\[
\begin{align*}
\text{In}[27] &:= \text{SS = \text{Show}[\text{pl, \text{Graphics[}}\{\text{\text{Line[}}\{\{\text{kbar1, 0}, \{\text{kbar1, 1.7}\}\}\}\}\}\}, \\
&\text{AxesLabel} \to \{"K Stock", "Consumption"}, \\
&\text{DisplayFunction} \to \$\text{DisplayFunction}] \\
\text{Out}[27] &:=
\end{align*}
\]

Transition Dynamics

Suppose that we start off with a capital stock less than the steady state capital stock. How does the economy evolve through time? A Phase Plot gives us the answer.

The "fish field" tells us both the direction and strength of the flow. Notice that the flow is stronger the farther the system is from the \( k'[t]=0 \) or \( c'[t]=0 \) lines and in particular that the flow is slowest nearest the equilibrium point. The field also tells us something interesting about the solution to the consumer's maximization problem. Suppose that the
capital stock starts at 1. If consumption is too low then according to the dynamics, households begin to add to their capital stock. They keep adding to the capital stock even as consumption begins to fall. Eventually consumption approaches zero and the capital stock approaches a large constant (the \( k \) where \( \dot{k} = 0 \) intersects the \( x \) axis). Intuitively, this path cannot be optimal. Households could attain higher utility by consuming some of their capital horde! (More technically, it can be shown that capital hoarding violates the transversality condition). If consumption starts too high, however, the dynamics indicate increasing consumption financed with a falling capital stock. Eventually the capital stock is fully consumed and consumption crashes to zero. But this too cannot be optimal. Even if it were optimal for the households to consume all of their capital stock they would never do it in a way which requires a consumption crash—they would seek to smooth consumption instead. We can rule out any solution, therefore, that does not converge to the steady state. To understand what happens when the capital stock starts away from the steady state we must solve for a level of consumption such that the dynamics of the model lead exactly to the steady state. We demonstrate two methods for solving this problem. First, the trial and error or shooting method and then the more elegant time elimination method due to Mulligan and Salai-i-Martin (1991).

Suppose \( k[0] = 1 \), the shooting method picks an arbitrary initial consumption level, \( c[0] = a \) and asks whether given \( k[0] = 1 \) and \( c[0] = a \) the solution path converges to the steady state. Below we solve the model given two possible initial levels of consumption \( c[0] = 0.6 \) and \( c[0] = 0.7 \).

\[
\text{NDSolve\text{::nds}z:}  \\
\text{At } t \approx 13.017352193778757^*, \text{ step size is effectively zero; singularity or stiff system suspected.} \Rightarrow
\]

\[
\text{NDSolve\text{::nds}z:}  \\
\text{At } t \approx 13.017352193778757^*, \text{ step size is effectively zero; singularity or stiff system guessed.} \Rightarrow
\]

Neither candidate solution converges to the steady state. The first implies capital hoarding and eventually zero consumption. The second implies that the capital stock goes to zero and that consumption crashes. From the phase plot, however, it appears that a solution exists somewhere in between \( c[0] = 0.55 \) and \( c[0] = 0.65 \) which will converge exactly (or arbitrarily closely) to the steady state. The following code uses \texttt{FindRoot} to arrive at an approximate solution.

\[
\text{system[a_]} :=  \\
\text{NDSolve\{c'[t] == CPrime[1/3, 0.05, 0.02, 0.02, 1.75],}  \\
\text{k'[t] == KPrime[1/3, 0.05, 0.02, 0.01], c[0] == a, k[0] == 1,}  \\
\text{\{c[t], k[t]\}, \{t, 0, 150\};}
\]
In[50]:= gun := c[system[H1][[1, 1, 2, 1, 1, 2]]] /. system[H1][[1, 1]] &

In[51]:= init = a /. FindRoot[gun[a] == cbar1, {a, 0.55, 0.65}, AccuracyGoal -> 4, MaxIterations -> 35];

NDSolve::ndinnt : Initial condition a is not a number or a rectangular array of numbers. ⇒

Part::partd : Part specification NDSolve[<<1>>][1, 1, 2, 1, 1, 2] is longer than depth of object. ⇒

NDSolve::ndinnt : Initial condition a is not a number or a rectangular array of numbers. ⇒

ReplaceAll::reps :

\[
\{c'[t] = 0.571429 c[t] \left( -0.105 + \frac{1}{3 k[t]^{2/3}} \right) \}
\]

is neither a list of replacement rules nor a valid dispatch table, and so cannot be used for replacing. ⇒

ReplaceAll::reps :

\[
\{c'[t] = 0.571429 c[t] \left( -0.105 + \frac{1}{3 k[t]^{2/3}} \right) \}
\]

is neither a list of replacement rules nor a valid dispatch table, and so cannot be used for replacing. ⇒

ReplaceAll::reps :

\[
\{c'[t] = 0.571429 c[t] \left( -0.105 + \frac{0.333333}{k[t]^{2/3}} \right) \}
\]

is neither a list of replacement rules nor a valid dispatch table, and so cannot be used for replacing. ⇒

General::stop : Further output of ReplaceAll::reps will be suppressed during this calculation. ⇒

FindRoot::nlnum :

The function value \(-1.32924 + (c[100.]. c'[t] = 0.571429 c[t] (\sim 0.105 + 0.333333 \text{Power}[\sim 1.]))\)
is not a list of numbers with dimensions {1} at \{a\} = \{0.55\}.

NDSolve::ndinnt : Initial condition a is not a number or a rectangular array of numbers. ⇒

General::stop : Further output of NDSolve::ndinnt will be suppressed during this calculation. ⇒

Part::partd : Part specification NDSolve[<<1>>][1, 1, 2, 1, 1, 2] is longer than depth of object. ⇒

FindRoot::nlnum :

The function value \(-1.32924 + (c[100.]. c'[t] = 0.571429 c[t] (\sim 0.105 + 0.333333 \text{Power}[\sim 1.]))\)
is not a list of numbers with dimensions {1} at \{a\} = \{0.55\}.
The function value \(-1.32924 + (c[100.] \cdot c[t] = 0.571429 c[t] (-0.105 + 0.333333 k[t]^2/3)\)

is not a list of numbers with dimensions \(1\) at \(a = 0.55\).

NDSolve::ndint : Initial condition \(a\) is not a number or a rectangular array of numbers. \(\Rightarrow\)

FindRoot::nlnum :

The function value \((-1.32924 + (c[100.] \cdot c[t] = 0.571429 c[t] (-0.105 + 0.333333 Power[\langle\rangle ])\))

is not a list of numbers with dimensions \(1\) at \(a = 0.55\).

NDSolve::ndint : Initial condition \(a\) is not a number or a rectangular array of numbers. \(\Rightarrow\)

FindRoot::nlnum :

The function value \((-1.32924 + (c[100.] \cdot c[t] = 0.571429 c[t] (-0.105 + 0.333333 Power[\langle\rangle ])\))

is not a list of numbers with dimensions \(1\) at \(a = 0.55\).

NDSolve::ndint : Initial condition \(a\) is not a number or a rectangular array of numbers. \(\Rightarrow\)

FindRoot::nlnum :

The function value \((-1.32924 + (c[100.] \cdot c[t] = 0.571429 c[t] (-0.105 + 0.333333 Power[\langle\rangle ])\))

is not a list of numbers with dimensions \(1\) at \(a = 0.55\).

General::stop : Further output of \texttt{FindRoot::nlnum} will be suppressed during this calculation. \(\Rightarrow\)

NDSolve::nlnum : The function value \(-1.32924 + (c[100.] \cdot c[t] = 0.571429 c[t] (-0.105 + 0.333333 k[t]^2/3)\)

is neither a list of replacement rules nor a valid

dispatch table, and so cannot be used for replacing. \(\Rightarrow\)

General::stop : Further output of \texttt{NDSolve::ndint} will be suppressed during this calculation. \(\Rightarrow\)

Part::partd : Part specification \texttt{NDSolve[\langle\rangle][1, 1, 2, 1, 2]} is longer than depth of object. \(\Rightarrow\)

NDSolve::ndint : Initial condition \(a\) is not a number or a rectangular array of numbers. \(\Rightarrow\)

General::stop : Further output of \texttt{NDSolve::ndint} will be suppressed during this calculation. \(\Rightarrow\)

Part::partd : Part specification \texttt{NDSolve[\langle\rangle][1, 1, 2, 1, 2]} is longer than depth of object. \(\Rightarrow\)

General::stop : Further output of \texttt{Part::partd} will be suppressed during this calculation. \(\Rightarrow\)

FindRoot::nlnum :

The function value \(-1.32924 + (c[100.] \cdot c[t] = 0.571429 c[t] (-0.105 + 0.333333 k[t]^2/3)\)

is neither a list of replacement rules nor a valid

dispatch table, and so cannot be used for replacing. \(\Rightarrow\)

General::stop : Further output of \texttt{FindRoot::nlnum} will be suppressed during this calculation. \(\Rightarrow\)

NDSolve::dsvar : \(0.0010204081632653062 \cdot \text{cannot be used as a variable.} \Rightarrow\)

NDSolve::dsvar : \(0.0010204081632653062 \cdot \text{cannot be used as a variable.} \Rightarrow\)

NDSolve::dsvar : \(0.0010204081632653062 \cdot \text{cannot be used as a variable.} \Rightarrow\)

General::stop : Further output of \texttt{NDSolve::dsvar} will be suppressed during this calculation. \(\Rightarrow\)
The Time Elimination Method

A more elegant way to solve these types of problems is the time elimination method (Mulligan and Sala-i-Martin, 1991). With this method we solve for the \( c[k] \) function directly. Notice that \( \frac{dc}{dt} = \frac{dk}{dt} = \frac{dc}{dk} \) and recall that we already know one point on the \( c[k] \) function, the steady state point, \( \bar{c} = c[\bar{k}] \). With these two pieces of information we can use \texttt{NDSolve} to find the entire \( c[k] \) function. First we define the derivative \( \frac{dc}{dk} \).

\begin{verbatim}
In[93]:= deriv[k_] :=
   CPrime[1/3, 0.05, .02, .02, 1.75] / KPrime[1/3, 0.05, .02, .01] /.
   {c[t] -> c[k], k[t] -> k}
\end{verbatim}

At the steady state, the derivative is undefined, since by definition \( \frac{dc}{dt} = \frac{dk}{dt} = 0 \). To get around this problem we give \texttt{NDSolve} an initial condition which is just slightly different from the true steady state. [See Endnote 2.]

\begin{verbatim}
In[94]:= sol1 = NDSolve[{c'[k] == deriv[k], c[kbar1] == cbar1 - .00001, c,
   {k, 0.01, kbar1}};
\end{verbatim}

\begin{verbatim}
In[95]:= PolicyF1 = c /. sol1[[1]]; 
\end{verbatim}

\begin{verbatim}
In[96]:= p1 = Plot[PolicyF1[k], {k, 0.01, kbar1}, PlotStyle -> Hue[0],
   DisplayFunction -> Identity]; 
\end{verbatim}

If we are interested in the finding the function \( c[k] \) when \( k \) starts out above the initial state we solve:

\begin{verbatim}
In[97]:= Show[SS, p1] 
\end{verbatim}
In[88]:= sol2 = NDSolve[
{c'[k] == deriv[k], c[kbar1] == cbar1 + .001}, c, 
{k, kbar1, 10}];

In[89]:= PolicyF2 = c /. sol2[[1]]; 

In[100]:= p2 = Plot[PolicyF2[k], {k, kbar1, 10}, PlotStyle -> Hue[0], 
DisplayFunction -> Identity]; 

In[101]:= Show[SS, p1, p2]

Out[101]= 2 4 6 8 10 K Stock

0.8 0.9 1.0 1.1 1.2 Consumption

It will be convenient to collect both parts of c[k] into one function.

(*Clear[PolicyFunction]*)

**Time Paths**

Often we are more interested in time paths than the "policy function" c[k]. We can solve for time paths using the time elimination method as an intermediate step. Consider the function \( \frac{dk}{dt} = -c(t) - 0.08 k(t) + \sqrt{k(t)} \), although we do not know the function c[t] we just solved for c[k[t]]. If we substitute c[k[t]] into \( \frac{dk}{dt} \) then with an arbitrary initial condition c[x] == y we can numerically solve for the function k[t].

First define f = c[k[t]] using the solution found earlier.

In[102]:= f = c /. sol1[[1]]

Out[102]= InterpolatingFunction[{{0.01, 5.65632}}, <>]

Then in \( \frac{dk}{dt} \) substitute f[k[t]] for c[t] and solve using the initial condition k[0] = 1.

In[103]:= sol3 = NDSolve[{k'[t] == KPrime[1/3, 0.05, .02, .01] /. c[t] -> f[k[t]], 
k[0] == 1}, k, {t, 0, 65}]

Out[103]= {{k -> InterpolatingFunction[{{0.01, 5.65632}}, <>]}}

In[104]:= KPath = k /. sol3[[1]]

Out[104]= InterpolatingFunction[{{0.01, 5.65632}}, <>]

In[105]:= KPlot = Plot[KPath[t], {t, 0, 65}, DisplayFunction -> Identity]; 

The figure below shows how the capital stock approaches its steady state value as a function of time.
Suppose that a war cuts a nation's capital stock in half. Figures such as the previous one can be used to answer questions like "How long will it take a nation to recover from the war" and "How fast will the economy grow along the transition path?" (See for example King and Rebelo (1993).)

Since $Y/L = Ay=Af[k]$ we can calculate the behavior of output per unit of effective labour along the transition path.

The economy's growth rate along the transition path is:

Notice from above that the economy initially grows very quickly but then the growth rate slows as the economy approaches the steady state. We can calculate half-life's and related data by using Mathematica's findroot function. (To get the functions in appropriate form for FindRoot we define an intermediate function z).
Notice from above that the economy initially grows very quickly but then the growth rate slows as the economy approaches the steady state. We can calculate half-life's and related data by using Mathematica's findroot function. (To get the functions in appropriate form for FindRoot we define an intermediate function \( z \)).

\[
\text{Out}[116]= \text{z}[x_] = \text{yPath} /. \{t \to x\}
\]

\[
\text{Out}[116]= \text{InterpolatingFunction}[\{\{0., 65.\}\}, \langle\rangle][x]^{1/3}
\]

Taking \( z[65] \) as the steady state \( y \) and noting that \( y \) starts out at level 1 we find that it takes 6.1 years for output to recover 1/2 of the war time loss.

\[
\text{In}[117]= \text{FindRoot}[z[t] == 1 + (z[65] - 1)/2, \{t, 2\}]
\]

\[
\text{Out}[117]= \{t \to 6.14073\}
\]

To solve for the time path of consumption, recall that \( \frac{dc}{dt} = 0.571429 c(t) \left( \frac{1}{3k(t)^2} - 0.105 \right) \). We can, therefore, substitute \( k(t) \) using the function we have just found and then solve with an initial point. We choose the initial point for consumption to be \( c[k[1]] \) so our consumption path will correspond to the capital path derived above.

\[
\text{In}[118]= \text{sol4} = \text{NDSolve}[\{c'[t] == \text{CPrime}[1/3, 0.05, .02, .02, 1.75] /. k[t] \to \text{KPath}[t],
\]

\[
c[0] == \text{PolicyFunction}[1], c, \{t, 0, 65\}\}
\]

\[
\text{NDSolve}::\text{ndinnt} :
\]

Initial condition PolicyFunction[1] is not a number or a rectangular array of numbers. ✗

\[
\text{Out}[118]= \text{NDSolve}[
\]

\[
\{c'[t] =
0.571429 c[t] \left( -0.105 + \frac{1}{3 \text{InterpolatingFunction}[\{\{0., 65.\}\}, \langle\rangle][t]^{2/3}} \right),
\]

\[
c[0] == \text{PolicyFunction}[1]\}, c, \{t, 0, 65\}\}
\]

\[
\text{In}[91]= \text{CPath} = c /. \text{sol4}[[1]]
\]

\[
\text{ReplaceAll}::\text{reps} :
\]

\[
\{c'[t] = 0.571429 c[t] \left( -0.105 + \frac{1}{3 \text{InterpolatingFunction}[\{\{0., 65.\}\}, \langle\rangle][t]^{2/3}} \right), c[0] == \text{PolicyFunction}[1]\} \text{ is neither a list of replacement rules nor a valid dispatch table, and so cannot be used for replacing. ✗}
\]

\[
\text{Out}[91]= c /. \{c'[t] =
0.571429 c[t] \left( -0.105 + \frac{1}{3 \text{InterpolatingFunction}[\{\{0., 65.\}\}, \langle\rangle][t]^{2/3}} \right),
\]

\[
c[0] == \text{PolicyFunction}[1]\}
\]
If we have proceeded correctly then in the steady state we should find that the difference between output and consumption is just enough to keep the effective capital stock constant. Recall that to keep the effective capital stock constant it must be increased by 8% every period. Using the values at t=65 as an approximation to the steady state we can check this as follows:

\begin{verbatim}
In[114]:= (yPath /. \{t -> 65\}) - CPath[65]

ReplaceAll::reps : \{c[0.571429 c[t - 0.105 + \frac{1}{3 \times 1.77952} \times c[t] = PolicyFunction[1]}] \text{ is neither a list of replacement rules nor a valid dispatch table, and so cannot be used for replacing.}

Out[114]= 1.77952 - \{c /. \{c'[t] =

0.571429 c[t] - 0.105 + \frac{1}{3 \text{InterpolatingFunction}[\text{\{\{0., 65.\}\}, <\text{\[\to\]}][t^{2/3}]],

\text{c[0]} = \text{PolicyFunction[1]}\}\}[65]
\end{verbatim}

\begin{verbatim}
In[115]:= KPath[65] * 0.08

Out[115]= 0.450815
\end{verbatim}

Thus we see that in the steady state savings are just equal to what is necessary to keep the effective capital stock constant.
References (with hyperlinks)


Alan Heston, Robert Summers and Bettina Aten, Penn World Table Version 6.2, Center for International Comparisons of Production, Income and Prices at the University of Pennsylvania, September 2006., available here.


D. K. Foley, Lecture Notes on Non-Linear Dynamics. Available here

General::newpkg : Graphics `PlotField` is now available as the Vector Field Plot Package. See the Compatibility Guide for updating information. ✧

General::obspkg :

Graphics `Arrow` is now obsolete. The legacy version being loaded may conflict with current Mathematica functionality. See the Compatibility Guide for updating information. ✧

General::obspkg :

Statistics `DataSmoothing` is now obsolete. The legacy version being loaded may conflict with current Mathematica functionality. See the Compatibility Guide for updating information. ✧

Printed by Mathematica for Students
General::newpkg: Graphics`Polyhedra` is now available as the Polyhedron Operations Package. See the Compatibility Guide for updating information. »

Triangle::shdw:
Symbol Triangle appears in multiple contexts {Geometry`Polytopes`, Graphics`MultipleListPlot`}; definitions in context Geometry`Polytopes` may shadow or be shadowed by other definitions. »

General::newpkg: Graphics`PlotField3D` is now available as the Vector Field Plot Package. See the Compatibility Guide for updating information. »

General::stop: Further output of General::newpkg will be suppressed during this calculation. »

General::obspkg:
Graphics`Graphics3D` is now obsolete. The legacy version being loaded may conflict with current Mathematica functionality. See the Compatibility Guide for updating information. »

General::stop: Further output of General::obspkg will be suppressed during this calculation. »

KurtosisExcess::shdw: Symbol KurtosisExcess appears in multiple contexts {Statistics`MultiDescriptiveStatistics`, Statistics`DescriptiveStatistics`}; definitions in context Statistics`MultiDescriptiveStatistics` may shadow or be shadowed by other definitions. »

CovarianceMatrix::shdw:
Symbol CovarianceMatrix appears in multiple contexts {Statistics`MultiDescriptiveStatistics`, Statistics`Common`MultivariateCommon`}; definitions in context Statistics`MultiDescriptiveStatistics` may shadow or be shadowed by other definitions. »

CorrelationMatrix::shdw:
Symbol CorrelationMatrix appears in multiple contexts {Statistics`MultiDescriptiveStatistics`, Statistics`Common`MultivariateCommon`}; definitions in context Statistics`MultiDescriptiveStatistics` may shadow or be shadowed by other definitions. »