# PhD Macroeconomics 1: <br> 1. Deterministic Dynamic Models 

Karl Whelan<br>School of Economics, UCD

Autumn 2023

## Part I

## Introduction

## Macroeconomic Dynamics

- Parts of macroeconomics relate to purely static relationships, e.g. national income accounting identities such as $Y=C+I+G+X-M$. But most of modern macroeconomics is focused on dynamics: How do macroeconomic variables change over time and how do the variables interact with each other?
- In some cases, macroeconomists design models intended to match data and be used for policy analysis. In other cases, they design "stylised" models designed to shed light on certain specific phenomenon or inter-relationships.
- Macroeconomic models also differ in other ways:
- Are they deterministic or stochastic?
- Do they use discrete time or continuous time?
- Do they use Lagrangian-based optimisation methods or more sophisticated techniques like dynamic programming and optimal control?
- Do they have a "'representative agent" or do they explicitly model and aggregate heterogenous agents?
- We will use "type of model" as our organising framework for presenting the material.


## An Applied Focus: Learning Macroeconomics via Matlab

- The point of this course is not to teach you a bunch of theories.
- The goal is to teach macroeconomics in an applied way. You will learn how to get models solved and running on a computer and how to adjust them and get them to answer questions.
- To do this we will use Matlab. Here are some reasons we will use Matlab rather than R or Python.
(1) It has higher functionality, built-in-help and better documentation and is available for free download for all UCD staff and students.
(2) It is the most commonly used software for modelling theoretical macroeconomics. Once you know how to use Matlab, you will have access to lots of online code relevant to macroeconomics, including replication code for many papers.
(3) In particular, Matlab allows you to use the Dynare package for solving and simulating macro models and also the Macro Model Database which has code for a huge number of models.
(9) Matlab code can also be used with the free software package GNU Octave.


## A Quick Taxonomy of Dynamic Model Types

- We will start by exploring different types of variables that feature in different macroeconomic models. See the taxonomy below for four different types of model.
- We will start in the top-left quadrant: Discrete-time deterministic models. We will not cover the bottom right quadrant but I can direct you to resources on this if you would like.

|  |  | Treatment of Time |  |
| :--- | :--- | :--- | :--- |
|  |  | Discrete | Continuous |
| Treatment of <br> Uncertainty | Deterministic | Difference <br> equations | Differential <br> equations |
|  | Stochastic | Stochastic <br> difference <br> equations | Wiener <br> processes |
|  |  | Time series <br> Discrete- <br> time <br> Markov <br> chains | Continuous- <br> time Markov <br> chains |

## Part II

## Single Variable Difference Equations

## The Simplest Difference Equation

- Difference equations describe discrete-time dynamics.
- Consider the simplest possible linear difference equation

$$
x_{t}=\lambda x_{t-1}
$$

- We assume this equation has always held, so that means

$$
x_{t}=\lambda x_{t-1}=\lambda^{2} x_{t-2}=\ldots=\lambda^{t} x_{0}
$$

- What kind of dynamics will this equation imply? It depends on the value of $\lambda$
- $\lambda>1$ : Explosive dynamics in which the series heads steadily for plus or minus infinity, depending on whether $x_{0}$ is positive or negative.
- $\lambda=1$ : Series remains unchanged.
- $0<\lambda<1$ : Smooth dynamics in which the series heads steadily towards zero from its starting point.
- $\lambda=0$ : Series set equal to zero after period zero and stays there.
- $-1<\lambda<0$ : Damped oscillations, settling down at zero.
- $\lambda<-1$ : Exploding oscillations


## $1^{s t}$-Order Difference Equation Paths With Different $\lambda$






## Setting the First Period Value to -1






## MATLAB Code for Calculations

- The previous graphs were generated using Matlab and saved as PNG files.
- Here's the code to do the calculations. Note how we didn't have to code this as $y 1(t, 1)=1$ ambda1*y1 ( $t-1,1$ )
- This is because (in a rare forgiving mood) Matlab lets you index vectors with only one row or one column with a single number.

```
T = 30;
% Initialise the four series.
% Matlab likes to know exactly what shape a matrix is
% So set up with zeros or ones or NaNs before filling it with what you
% really want.
y1 = ones(T,1);
y2 = ones (T,1);
y3 = ones(T,1);
y4 = ones(T,1);
lambdal = 1.2;
lambda2 = 0.8;
lambda3 = -0.8;
lambda4 = -1.2;
for t=2:30
    yl(t) = lambdal*yl(t-1);
    y2(t) = lambda2* y2 (t-1);
    y3(t) = lambda3*y3(t-1);
    y4}(t)=1\mp@subsup{\textrm{ambda4*}}{}{*}\mp@subsup{\textrm{y}}{}{4}(\textrm{t}-1)
end
```


## And Here's the Code for the Graphs

```
figure(1)
subplot (2,2,1)
plot(y1);
xlim([1 T])
title('$\lambda = 1.2$','Fontsize',16)
subplot (2,2,2)
plot(y2);
xlim([1 T])
title('$\lambda = 0.8$','Eontsize',16)
subplot (2,2,3)
plot(y3);
xlim([1 T])
title('$\lambda = -0.8$','Fontsize',16)
subplot (2,2,4)
plot(y4);
xlim([1 T])
title('$\lambda = -1.2 $','Fontsize',16)
figl = figure(1);
```


## Solving Difference Equations

- Consider the $n$-th order linear difference equation

$$
y_{t}+a_{1} y_{t-1}+a_{2} y_{t-2}+\ldots+a_{n} y_{t-n}=0
$$

- This is called a homogenous difference equation because the constant term on the right-hand-side is zero.
- Guess that the solution is of the form $y_{t}=A b^{t}$ and insert into the equation

$$
A b^{t}+a_{1} A b^{t-1}+a_{2} A b^{t-2}+\ldots .+a_{n} A b^{t-n}=0
$$

- Divide by $A b^{t-n}$ and this becomes

$$
b^{n}+a_{1} b^{n-1}+a_{2} b^{n-2}+\ldots .+a_{n}=0
$$

- This is an $n$-th order polynomial equation with up to $n$ distinct possible solutions. This is known as the characteristic equation of the original difference equation.


## Solving Difference Equations

- Assuming $n$ distinct roots, there are $n$ different values $b_{1}, b_{2}, \ldots, b_{n}$ such that $y_{t}=b_{i}^{t}$ solves the difference equation.
- Also $y_{t}=A_{i} b_{i}^{t}$ works as a solution for any $A_{i}$ as do sums of different solutions.
- This gives a general solution of the form

$$
y_{t}=A_{1} b_{1}^{t}+A_{2} b_{2}^{t}+\ldots+A_{n} b_{n}^{t}
$$

- We can be more specific about values of the $A_{i}$ if we are given $n$ "boundary conditions", e.g. the first or last $n$ values of the series $y_{t}$. This allows you to pin down a unique solution. (Intuitively, for most problems it makes sense to have initial conditions rather than terminal ones but it depends on the example.)
- What if we have a non-homogenous equation?

$$
y_{t}+a_{1} y_{t-1}+a_{2} y_{t-2}+\ldots+a_{n} y_{t-n}=c
$$

- The constant "particular solution" below works and can be added to the solution of the homogenous equation.

$$
y^{*}=\frac{c}{1+a_{1}+a_{2}+\ldots .+a_{n}}
$$

## Example: Second Order Difference Equations

- Consider the following second-order equation

$$
y_{t}+a_{1} y_{t-1}+a_{2} y_{t-2}=c
$$

- The constant solution is

$$
y^{*}=\frac{c}{1+a_{1}+a_{2}}
$$

- The characteristic equation is

$$
b^{2}+a_{1} b+a_{2}=0
$$

- The solutions are given by

$$
b_{1}, b_{2}=\frac{-a_{1} \pm \sqrt{a_{1}^{2}-4 a_{2}}}{2}
$$

## Example: Second Order Difference Equations

There are three types of solutions.

- Case 1: Real Distinct Roots: $a_{1}^{2}-4 a_{2}>0$ and the solution is of the form

$$
y_{t}=y^{*}+A_{1} b_{1}^{t}+A_{2} b_{2}^{t}
$$

- Case 2: Real Repeated Roots: $a_{1}^{2}-4 a_{2}=0$ so the solution of the characteristic equation is just $b=-\frac{a_{1}}{2}$ and the solution is of the form

$$
y_{t}=y^{*}+\left(A_{1}+A_{2} t\right)\left(\frac{a_{1}}{2}\right)^{t}
$$

The additional type of solution $A_{2} t\left(\frac{a_{1}}{2}\right)^{t}$ only works when there are two repeated roots.

- Case 3: Complex Roots: $a_{1}^{2}-4 a_{2}<0$ so the characteristic equation has two complex solutions $b_{1}, b_{2}=h \pm i v$ implying a solution to the difference equation of the form

$$
y_{t}=y^{*}+A_{1}(h+i v)^{t}+A_{2}(h-i v)^{t}
$$

where $h=-\frac{a_{1}}{2}$ and $v=\frac{\sqrt{4 a_{2}-a_{1}^{2}}}{2}$

## Second Order Difference Equations with Complex Roots

- What kind of behaviour do we get from the solution?

$$
y_{t}=y^{*}+A_{1}(h+i v)^{t}+A_{2}(h-i v)^{t}
$$

- A way to characterise this behaviour in terms of known mathematical functions is to express the complex numbers in polar form

$$
h \pm i v=R(\cos \theta \pm i \sin \theta)
$$

where

$$
\begin{aligned}
& R \quad=\quad \sqrt{h^{2}+v^{2}}=\sqrt{\left(\frac{a_{1}}{2}\right)^{2}+\left(\frac{\sqrt{4 a_{2}-a_{1}^{2}}}{2}\right)^{2}}=a_{2}^{\frac{1}{2}} \\
& \theta=\tan ^{-1}\left(\frac{v}{h}\right)
\end{aligned}
$$

- De Moivre's theorem can be used to characterise the solution as being of the form

$$
y_{t}=y^{*}+B_{1} \sin \theta t+B_{2} \cos \theta t
$$

## An Example

- Consider the second-order difference equation

$$
y_{t}-1.05 y_{t-1}+0.3 y_{t-2}=3
$$

- What kind of roots does this have? Matlab can tell us using its roots command. (Matlab has various ways of finding solutions to non-linear equations. This is a command specifically for polynomials.)

```
% Check that the characteristic equation has complex roots
al = -1.05;
a2 =0.3;
de_roots = roots([[1 al a2]);
disp('Roots of the Characteristic Equation');
disp(de_roots);
Roots of the Characteristic Equation
    0.5250 + 0.1561i
    0.5250 - 0.1561i
```

- So this characteristic equation has complex roots.
- The next page shows time paths for the variable described by this difference equation using two different sets of initial conditions.


## Time Paths with Two Different Sets of Initial Conditions



## Matlab Code for Previous Graph

```
* Set latex as default interpreter
set(groot, 'defaulttextinterpreter','latex'):
set(groot, 'defaultAxesTickLabelInterpreter','latex'):
set(groot, 'defaultLegendIntexpreter','latex'):
T = 15;
tine = linapace (1,T,T)
al - -1.05;
a.2 - 0.3;
Y1 = zeros(T,1);
Y2 = zeros(T,1);
yl(1) = 5;
yl (2) = 11;
y2(1) = 11;
y}2(2)=12
Gor t=3:T
    yl(t) = -al*yl(t-1) - a2*yl(t-2) + 3;
    y2(t) - -al*y2(t-1) - a2* y2 (t-2) + 3;
end
I1gure(1)
plot(time, y1, time, y2);
xlim([l T])
legend('$y_0=5, y_I=11$' , '$y_0=11, Y_l=12$','Fonts1ze',32,'Location','southeast')
figl = figure(1);
```


## Example: Initial Conditions Pinning Down the Solution

- Consider the following second-order equation

$$
y_{t}-7 y_{t-1}+10 y_{t-2}=5
$$

where $y_{0}=2$ and $y_{1}=3$.

- The characteristic equation is

$$
b^{2}-7 b+10=0
$$

with solutions $b_{1}=5$ and $b_{2}=2$.

- A particular solution $y^{*}$ satisfies

$$
y^{*}-7 y^{*}+10 y^{*}=5 \Longleftrightarrow y^{*}=\frac{5}{1-7+10}=\frac{5}{4}
$$

- Calculate coefficients from

$$
\begin{aligned}
& y_{0}=A_{1}+A_{2}+\frac{5}{4}=2 \\
& y_{1}=5 A_{1}+2 A_{2}+\frac{5}{4}=3
\end{aligned}
$$

which solve to give $A_{1}=\frac{1}{12}$ and $A_{2}=\frac{2}{3}$. Solution is $y_{t}=\frac{1}{12} 5^{t}+\frac{2}{3} 2^{t}+\frac{5}{4}$.

## Some Lessons

- It's pretty easy to use Matlab to simulate discrete-time models!
- Depending on the coefficients, linear discrete time models can generate behaviour that is explosive, zig-zag, smoothly convergent or oscillating.
- The particular paths that processes take will depend upon the initial conditions.
- Second-order difference equations can display more complex behaviour than first-order equations. This is an illustration of a general point. The higher the order of the difference equation-the larger the gap between the highest and lowest time index-the more complex the dynamics can potentially be.
- This latter point matters when specifying time series models: If your model has lots of lags in it, it might exhibit odd behaviour.


## Part III

## Systems of Difference Equations

## The Simplest System

- Models with just one variable don't feature much in economics.
- More relevant are models with more than one variable and interactions between these variables.
- The simplest possible system of difference equations features two variables and one lag:

$$
\begin{aligned}
& y_{1, t}=a_{11} y_{1, t-1}+a_{12} y_{2, t-1} \\
& y_{2, t}=a_{21} y_{1, t-1}+a_{22} y_{2, t-1}
\end{aligned}
$$

- A compact way to express this sytem is to use matrices. Defining the matrices

$$
\begin{aligned}
Y_{t} & =\binom{y_{1 t}}{y_{2 t}} \\
A & =\left(\begin{array}{ll}
a_{11} & a_{12} \\
a_{21} & a_{22}
\end{array}\right)
\end{aligned}
$$

- This system can be written as

$$
Y_{t}=A Y_{t-1}
$$

## Generality of the First-Order Matrix Formulation: I

- The model we've been looking at may seem like a small subset of all possible systems of difference equations because it doesn't have a constant term on the right-hand side and only has lagged values from one period ago.
- However, you can add a third variable here which takes the constant value 1 each period. The equation for the constant term will just state that it equals its own lagged values. So this formulation actually incorporates models with constant terms.
- What about systems where current variables depend on values from more than one period ago? Surely this makes things much more complicated?
- Not really. It turns out the first-order matrix formulation can represent systems with longer lags.
- Consider the two-lag system

$$
\begin{aligned}
& y_{1, t}=a_{11} y_{1, t-1}+a_{12} y_{1, t-2}+a_{13} y_{2, t-1}+a_{14} y_{2, t-2} \\
& y_{2, t}=a_{21} y_{1, t-1}+a_{22} y_{1, t-2}+a_{23} y_{2, t-1}+a_{24} y_{2, t-2}
\end{aligned}
$$

## Generality of the First-Order Matrix Formulation: II

- Now define the vector

$$
Z_{t}=\left(\begin{array}{c}
y_{1, t} \\
y_{1, t-1} \\
y_{2, t} \\
y_{2, t-1}
\end{array}\right)
$$

- This system can be represented in matrix form as

$$
Z_{t}=A Z_{t-1}
$$

where

$$
A=\left(\begin{array}{cccc}
a_{11} & a_{12} & a_{13} & a_{14} \\
1 & 0 & 0 & 0 \\
a_{21} & a_{22} & a_{23} & a_{24} \\
0 & 0 & 1 & 0
\end{array}\right)
$$

- This is sometimes called the "companion form" matrix formulation of a dynamic model.


## Stability Conditions for Systems

- We have seen how single-variable difference equations can display explosive behaviour or else converge to a long-run equilibrium.
- What about systems?
- What are the conditions must the $A$ matrix obey for these different outcomes to emerge?
- To answer this question, we need to discuss eigenvalues (sorry ...).


## Eigenvalues

- A value $\lambda_{i}$ is an eigenvalue of the matrix $A$ if there exists a vector $e_{i}$ (known as an eigenvector) such that

$$
A e_{i}=\lambda_{i} e_{i}
$$

- I'm going to assume here for simplicity that the $n \times n$ matrix $A$ has $n$ distinct eigenvalues (it could have multiple eigenvectors associated with one eigenvalue and then there would be fewer than $n$ eigenvalues - this case slightly complicates the situation and we'll leave it aside).
- Denote by $P$ the matrix that has as its columns $n$ eigenvectors corresponding to these eigenvalues. In this case,

$$
A P=P \Omega
$$

where

$$
\Omega=\left(\begin{array}{cccc}
\lambda_{1} & 0 & 0 & 0 \\
0 & \lambda_{2} & 0 & 0 \\
0 & 0 & . & 0 \\
0 & 0 & 0 & \lambda_{n}
\end{array}\right)
$$

is a diagonal matrix of eigenvalues.

## Stability Condition

- Note now that this equation implies that

$$
A=P \Omega P^{-1}
$$

- This tells us something about the relationship between eigenvalues and higher powers of $A$ because

$$
A^{n}=P \Omega^{n} P^{-1}=P\left(\begin{array}{cccc}
\lambda_{1}^{n} & 0 & 0 & 0 \\
0 & \lambda_{2}^{n} & 0 & 0 \\
0 & 0 & . & 0 \\
0 & 0 & 0 & \lambda_{n}^{n}
\end{array}\right) P^{-1}
$$

- So, the difference between lower and higher powers of $A$ is that the higher powers depend on the eigenvalues taken to the power of $n$. If all of the eigenvalues are inside the unit circle (i.e. less than one in absolute value) then all of the entries in $A^{n}$ will tend towards zero as $n \rightarrow \infty$.
- So, a condition that ensures that the variables in the system tend towards a stable long-run equilibrium is that the eigenvalues of $A$ are all inside the unit circle.
- If any eigenvalues are greater than one, the solutions will generally explode.


## Diagonalising the System

- If $A$ has $n$ distinct eigenvalues, then is a matrix of eigenvectors $P$ such that

$$
P^{-1} A P=\Omega
$$

where $\Omega$ is a diagonal matrix with the $n$ eigenvalues on the diagonal.

- Let's define a transformed set of variables

$$
Z_{t}=P^{-1} Y_{t}
$$

- Then we can write the dynamics of the transformed set of variables as

$$
Z_{t}=P^{-1} Y_{t}=P^{-1} A Y_{t-1}=P^{-1} A P Z_{t-1}=\Omega Z_{t-1}
$$

- This is a simple diagonal system consisting of $n$ first-order single-variable difference equations.

$$
\begin{aligned}
z_{1, t} & =\lambda_{1} z_{1, t-1} \\
z_{2, t} & =\lambda_{2} z_{2, t-1} \\
& \cdots \\
z_{n, t} & =\lambda_{n} z_{n, t-1}
\end{aligned}
$$

## Initial Conditions and Solutions

- The diagonalised system solves to give

$$
\begin{aligned}
z_{1, t} & =\lambda_{1}^{t} z_{1,0} \\
z_{2, t} & =\lambda_{2}^{t} z_{2,0} \\
& \cdots \\
z_{n, t} & =\lambda_{n}^{t} z_{n, 0}
\end{aligned}
$$

so the dynamics depend on the eigenvalues and the initial values of the $z_{i}$ s. Since the $z_{i}$ variables are just linear combinations of $y_{i, t}, \ldots, y_{n, t}$ these depend on the initial conditions for our original variables.

- Having solved a diagonalised system, we can recover the original variables via $Y_{t}=P Z_{t}$, so we get solutions of the form

$$
\begin{aligned}
y_{1, t} & =\theta_{11} z_{1, t}+\theta_{12} z_{2, t}+\ldots \theta_{1 n} z_{n, t} \\
y_{2, t} & =\theta_{21} z_{1, t}+\theta_{22} z_{2, t}+\ldots \theta_{2 n} z_{n, t} \\
& \ldots \\
y_{n, t} & =\theta_{n 1} z_{1, t}+\theta_{n 2} z_{2, t}+\ldots \theta_{n n} z_{n, t}
\end{aligned}
$$

where the $\theta \mathrm{s}$ are the coefficient of $P^{-1}$.

## On Stability of Economic Models

- If all eigenvalues of this system are less then one in absolute value, then it is clear the system will converge with all the variables heading towards specific long-run levels.
- If, however, at least one of the eigenvalues is greater than one in absolute value, then for any general set of initial conditions, all or some of the variables in the model will explode.
- One caveat to this: In some macroeconomic models, the agents in the model get to choose the initial conditions. We will provide an example later where the equations describing a model of optimal consumption have generally explosive solutions but optimising households choose an inital level of consumption such that the coefficients on the solutions associated with explosive eigenvalues are zero.
- This idea that a model is generally explosive but there is a particular sub-set of solutions that provide stable dynamics is common in macroeconomics. Sometimes we refer to the stable and convergent part of the solution as the "saddle path".


## Example: Calculating Eigenvalues for a $2 \times 2$ Matrix

- Consider, for example, a $2 \times 2$ matrix.

$$
A=\left(\begin{array}{ll}
a_{11} & a_{12} \\
a_{21} & a_{22}
\end{array}\right)
$$

- Suppose $A$ has two eigenvalues, $\lambda_{1}$ and $\lambda_{2}$ and define $\lambda$ as the vector

$$
\lambda=\binom{\lambda_{1}}{\lambda_{2}}
$$

- The fact that there is a matrix of distinct eigenvectors which when multiplied by $A-\lambda /$ equal a vector of zeros means that the determinant of the matrix

$$
A-\lambda I=\left(\begin{array}{cc}
a_{11}-\lambda_{1} & a_{12} \\
a_{21} & a_{22}-\lambda_{2}
\end{array}\right)
$$

equals zero.

- So we get the two eigenvalues of $A$ by solving the quadratic formula

$$
\left(a_{11}-\lambda_{1}\right)\left(a_{22}-\lambda_{2}\right)-a_{12} a_{21}=0
$$

- In practice, it is easy to get eigenvalues using the eig function in Matlab.


## Part IV

## First-Order Differential Equations

## Continuous Time Dynamics

- Up to now, we have been looking at discrete-time models: Time is indexed as a sequence of numbers.
- But for some issues, there are benefits to working in a continuous time format where the time index can be any positive real-numbered value.
- In these models, instead of difference equations we have differential equations. These are equations involving derivatives of a function and the equations are solved by finding out what the function is.
- A first-order differential equation is one of the form

$$
\frac{d y}{d t}=f(y(t), t)
$$

where, in economics applications, the $t$ is understood to be an index for time. It is termed first-order because it features the first derivative of a function but no higher derivatives.

- We solve the equation by figuring out what the function $f$ is.


## First-Order Linear Differential Equations

- The simplest kind of differential equation is a linear first-order differential equation of the form

$$
\frac{d y}{d t}+\alpha y=c
$$

where $c$ is a constant.

- If $c=0$, so the equation is

$$
\frac{d y}{d t}+\alpha y=0
$$

- The solution is a function whose derivative is a multiple of the function itself. This suggests we should guess that it involves the exponential function. And indeed a solution that works is

$$
y(t)=e^{-\alpha t}
$$

- To solve the non-homogenous version of the equation, we need to what we call a "particular solution" and then add this to general solution for the homogenous equation.


## Simple and More Complicated Particular Solutions

- For the simple model

$$
\frac{d y}{d t}+\alpha y=c
$$

where $c$ is a constant, the particular solution is a constant $y^{p}=\frac{c}{\alpha}$ so the full solution to the differential equation is

$$
y(t)=e^{-\alpha t}+\frac{c}{\alpha}
$$

- For more complicated equations of the form

$$
\frac{d y}{d t}+\alpha y=c(t)
$$

where the exogenous element is itself a function of time, the general principle is still the same. First, solve the homogenous equation, then try to figure out a particular solution and add the two together.

## Boundary Conditions

- We have described how $y(t)=e^{-\alpha t}$ is a solution of

$$
\frac{d y}{d t}+\alpha y=0
$$

- But note that any solution of the form

$$
y_{t}=\gamma e^{-\alpha t}
$$

also works.

- If we want to use the differential equation to pin down an actual time path for $y$ how do we figure out which of the possible solutions is the right one?
- As with difference equations, one convention is to specific an initial condition e.g. note that $y_{0}=-\gamma \alpha$ so if we specify for example that $y_{0}$ equals a specific number, then this pins down the value of $\gamma$.
- While people often discuss "initial conditions" or "boundary condition", technically you only need to specify the value of $y_{t}$ at one point in time.


## Part V

## Higher Order Differential Equations and Systems

## Higher Order Linear Differential Equations

- Solving the $N$-th order differential equation

$$
\frac{d^{n} y}{d t}+\alpha_{1} \frac{d^{n-1} y}{d t}+\alpha_{2} \frac{d^{n-2} y}{d t}+\ldots+\alpha_{n} y=c
$$

where $c$ is a constant. It follows a similar process to first-order equations and the solution method is analogous to the solutions for N -th order difference equations we covered previously.

- First, find a particular solution. In this case, with $c$ being constant, the particular solution is just $y^{p}=\frac{c}{\alpha_{n}}$
- Then find a solution to the homogenous equation

$$
\frac{d^{n} y}{d t}+\alpha_{1} \frac{d^{n-1} y}{d t}+\alpha_{2} \frac{d^{n-2} y}{d t}+\ldots+\alpha_{n} y=0
$$

- As before, start by guessing $y_{t}=A e^{r t}$ and use the fact that this means

$$
\frac{d^{n} y}{d t}=r^{n} e^{r t}
$$

## Finding the Right Values for $r$

- Taking all $n$ of the derivatives of our guess $y_{t}=A e^{r t}$, we get

$$
A r^{n} e^{r t}+\alpha_{1} A r^{n-1} e^{r t}+\alpha_{2} A r^{n-2} e^{r t}+\ldots+\alpha_{n} A e^{r t}=0
$$

- The $A e^{r t}$ terms can be cancelled out so we're left with

$$
r^{n}+\alpha_{1} r^{n-1}+\alpha_{2} r^{n-2}+\ldots+\alpha_{n}=0
$$

- This is $n$-order polynomial is known as the characteristic equation and it can have up to $n$ distinct (and possibly complex) solutions. (This may be feeling a bit familiar to you at this point ...)
- The general solution is thus of the form

$$
y(t)=A_{1} e^{r_{1} t}+A_{2} e^{r_{2} t}+\ldots . A_{n} e^{r_{n} t}+\frac{c}{\alpha_{n}}
$$

where $r_{1}, r_{2}, \ldots, r_{n}$ are the roots of the characteristic equation.

- This solution will solve the original differential equation for any combination of coefficients $A_{1}, A_{2}, \ldots, A_{n}$. To obtain a unique solution, we need to have $n$ "boundary conditions" of the form $y(0)=k_{0}, y\left(t_{1}\right)=k_{1}, \ldots y\left(t_{n}\right)=k_{n}$


## Second-Order Linear Differential Equations

- Let's work through a specific case. Second order linear differential equations take the form

$$
\frac{d^{2} y}{d t}+\alpha_{1} \frac{d y}{d t}+\alpha_{2} y=c
$$

- This gives a characteristic equation

$$
r^{2}+\alpha_{1} r+\alpha_{2}=0
$$

which has two solutions

$$
r_{1,2}=\frac{1}{2}\left(-\alpha_{1} \pm \sqrt{\alpha_{1}^{2}-4 \alpha_{2}}\right)
$$

- As with our analysis of second-order difference equations, there are three distinct cases.


## Again, Three Cases

- Case 1: Distinct Real Roots. In this case $\alpha_{1}^{2}>4 \alpha_{2}$ and the behaviour of the solution

$$
y(t)=A_{1} e^{r_{1} t}+A_{2} e^{r_{2} t}+\frac{c}{\alpha_{n}}
$$

depends on the signs of $r_{1}$ and $r_{2}$. Assuming non-zero values of $A_{1}$ and $A_{2}$, this solution will explode to plus or minus infinity if either $r_{1}$ or $r_{2}$ are positive. If both are negative, the solution will tend towards $\frac{c}{\alpha_{n}}$.

- Case 2: Identical Real Roots. In this case, $\alpha_{1}^{2}=4 \alpha_{2}$ and there is only one solution, $r$. You can show that the following works as a solution in this case

$$
y(t)=A_{1} e^{r t}+A_{2} t e^{r t}+\frac{c}{\alpha_{n}}
$$

Again, its behaviour will depend on whether $r$ is positive or negative.

## Case 3: Complex Roots

- In this case $\alpha_{1}^{2}<4 \alpha_{2}$ and we get complex roots

$$
r_{1,2}=h \pm i v
$$

- We can use the identities

$$
\begin{aligned}
e^{i x} & =\cos x+i \sin x \\
e^{-i x} & =\cos x-i \sin x
\end{aligned}
$$

to show the solution can be written as

$$
y(t)=e^{h t}\left(A_{3} \cos v t+A_{4} \sin v t\right)+\frac{c}{\alpha_{n}}
$$

where $A_{3}=A_{1}+A_{2}$ and $A_{4}=i\left(A_{1}-A_{2}\right)$.

- These solutions display oscillations. Whether they decline over time or explode depends on the sign of $h$. And since $h=-\frac{\alpha_{1}}{2}$, this depends on $\alpha_{1}$. If $\alpha_{1}$ is positive, the oscillations will decay over time.


## Systems of Linear Differential Equations

- Often when differential equations refer to changes over time, people use the shorthand

$$
\dot{y}(t)=\frac{d y}{d t}
$$

- Now consider the system of linear differential equations

$$
\begin{aligned}
& \dot{x_{1}}(t)=a_{11} x_{1}(t)+a_{12} x_{2}(t) \\
& \dot{x_{2}}(t)=a_{21} x_{1}(t)+a_{22} x_{2}(t)
\end{aligned}
$$

- Let's collect this together using vector and matrix notation.

$$
\binom{\dot{x_{1}}(t)}{\dot{x}_{2}(t)}=\left(\begin{array}{ll}
a_{11} & a_{12} \\
a_{21} & a_{22}
\end{array}\right)\binom{x_{1}(t)}{x_{2}(t)}
$$

Or, in shorthand

$$
\dot{x}=A x
$$

## Getting Solutions

- As before, in the case where $A$ has $n$ distinct eigenvalues, there is a matrix of eigenvectors $P$ such that

$$
P A P^{-1}=D
$$

where $D$ is a diagonal matrix with the $n$ eigenvalues on the diagonal.

- Let's define a transformed set of variables

$$
y=P x
$$

- Then we can write the dynamics of the transformed set of variables as

$$
\dot{y}=P \dot{x}=P A x=P A P^{-1} y=D y
$$

- This is a simple diagonal system that gives $n$ first-order differential equations in one variable. For instance, in the case of $n=2$, we get

$$
\begin{aligned}
\dot{y_{1}} & =\lambda_{1} y_{1} \\
\dot{y_{2}} & =\lambda_{2} y_{2} \\
& \cdots \\
\dot{y_{n}} & =\lambda_{n} y_{n}
\end{aligned}
$$

## Getting Solutions

- These equations can be solved separately to give

$$
\begin{aligned}
y_{1}(t) & =A_{1} e^{-\lambda_{1} t} \\
y_{2}(t) & =A_{2} e^{-\lambda_{2} t} \\
& \cdots \\
y_{n}(t) & =A_{n} e^{-\lambda_{n} t}
\end{aligned}
$$

- And given these solutions, the original $x$ variables can be obtained from

$$
x=B^{-1} y
$$

## Part VI

## Taylor Series and Nonlinear Differential Equations

## Taylor Series

- The Taylor series approximation for a function around point $a$ is the following:

$$
\begin{aligned}
f(x)= & \left.\left.f(a)+f^{\prime}(a)(x-a)+\frac{1}{2} f^{\prime \prime}(a)\right)(x-a)^{2}+\frac{1}{3!} f^{\prime \prime \prime}(a)\right)(x-a)^{3}+ \\
& \ldots \frac{1}{n!} f^{n}(a)(x-a)^{n}+\ldots
\end{aligned}
$$

where $n!=(1)(2)(3) \ldots(n-1)(n)$.

- Some Taylor series approximations characterise the behaviour of the function for all real but others only hold in a specific interval around the point a.
- Mathematicians sometime express these series as

$$
f(x)=f(a)+f^{\prime}(a)(x-a)+\mathcal{O}(2)
$$

In other words, $f$ is a function of the first two terms of the approximation plus terms that involve powers of 2 or higher. If $a$ is small, then these $\mathcal{O}(2)$ may be close to zero and we can approximate $f$ as a linear function:

$$
f(x) \approx f(a)+f^{\prime}(a)(x-a)
$$

- Taylor series are used a lot in macroeconomics.


## Taylor Series: Example

- Consider the function $f(x)=\frac{1}{1-x}$. Approximating this series around zero gives

$$
f(x)=\sum_{n=0}^{\infty} \frac{f^{n}(0)}{n!} x^{n}
$$

- You can show that

$$
f^{n}(x)=\frac{n!}{(1-x)^{n+1}} \Rightarrow \frac{f^{n}(0)}{n!}=1
$$

- This means the Taylor approximation around zero is

$$
\frac{1}{1-x}=1+x+x^{2}+x^{3}+\ldots . .+x^{N} \ldots
$$

which is, of course, the famous multiplier formula from Macro 101.

- This approximation works for $-1<x<1$ but not outside this range. (Why might the approximation not work some times?)

$$
\frac{1}{1-5} \neq 1+5+5^{2}+5^{3}+\ldots
$$

## The Exponential Function

- The number $e \approx 2.71828$ is a special number such that the function

$$
\frac{d e^{x}}{d x}=e^{x}
$$

- One way to see why the number is 2.718 is to use the Taylor series approximation for a function
$\left.\left.\left.f(x)=f(a)+f^{\prime}(x)(x-a)+\frac{1}{2} f^{\prime \prime}(x)\right)(x-a)^{2}+\frac{1}{3!} f^{\prime \prime \prime}(x)\right)(x-a)^{3}+\ldots \frac{1}{n!} f^{n}(x)\right)(x-a)^{\prime}$
where $n!=(1)(2)(3) \ldots(n-1)(n)$.
- If there is a number, $e$ that has the property that $e^{x}=f(x)=f^{\prime}(x)$, then that means that all derivatives also equal $e^{x}$. In this case, we have

$$
e^{x}=e^{a}+e^{a}(x-a)+\frac{1}{2} e^{a}(x-a)^{2}+\frac{1}{3!} e^{a}(x-a)^{3}+\ldots
$$

Setting $x=1, a=0$, this becomes

$$
e=1+\frac{1}{2}+\frac{1}{3!}+\frac{1}{4!}+\ldots \ldots
$$

This converges to 2.71828 .

## Multivariate Taylor Series

- You can also approximate multivariate functions using Taylor series. For example, a function of two variables $F\left(x_{t}, y_{t}\right)$ can be approximated as

$$
\begin{aligned}
F\left(x_{t}, y_{t}\right)= & F\left(x_{t}^{*}, y_{t}^{*}\right)+F_{x}\left(x_{t}^{*}, y_{t}^{*}\right)\left(x_{t}-x_{t}^{*}\right)+F_{y}\left(x_{t}^{*}, y_{t}^{*}\right)\left(y_{t}-y_{t}^{*}\right) \\
& +\frac{1}{2} F_{x x}\left(x_{t}^{*}, y_{t}^{*}\right)\left(x_{t}-x_{t}^{*}\right)^{2}+\frac{1}{2} F_{x y}\left(x_{t}^{*}, y_{t}^{*}\right)\left(x_{t}-x_{t}^{*}\right)\left(y_{t}-y_{t}^{*}\right) \\
& +\frac{1}{2} F_{y y}\left(x_{t}^{*}, y_{t}^{*}\right)\left(y_{t}-y_{t}^{*}\right)^{2}+\ldots
\end{aligned}
$$

- To give an example, for the function $z=y \log x$, the linearised Taylor approximation around $(a, b)$ is

$$
z \approx b \log a+\frac{b}{a}(x-a)+\log a(y-b)
$$

- The 3-d picture on the next page shows the size of the approximation error for values of $x$ and $y$ between zero and 10 , using $a=5, b=5$.
- As you can see, the approximation errors get very big as $x$ approaches zero. Ultimately, it is hard to approximate highly nonlinear functions with linear ones. The page after shows, however, that the approximation works very well in the window [4.5, 5.5].


## Approximation Error for $z=y \log x$ on $[0,10] \times[0,10]$



## Approximation Error for $z=y \log x$ on $[4.5,5.5] \times[4.5,5.5]$



## Matlab Code for Previous Two Graphs

```
clear all; clc; close all; set 10,'DefaultIegendAutoUpclate','off
tic;
% Set latex as default interpreter
set(groot, 'defaulttextinterpreter','latex');
set(groot, 'deIaultfxesIIckLabelInterpreter','latex"),
set(groot, 'defaultLegendInterpreter','latex'):
a=5;
b-5;
[X,Y] - meshgridl[4.5:0.1:5.5,4.5:0.1:5.5),
%[X,Y] = meshgrid(0:0.1:10,0:0.1:10);
z = b* log(a) + (b / a)*(X - a) + log(a)*(Y-b) -Y.* 酋 (X);
surf(X,Y,Z)
colorbar:
xlabel('$x$','Ionts1ze',20);
ylabel('$y$','fontsize',20):
zlabel('Error',''Iontslze',20);
```


## Nonlinear Difference Equations

- Linear difference equations represent only a small subset of the kinds of dynamics that can be generated by economic models. A more general model of nonlinear difference equations would take the form

$$
\begin{aligned}
\dot{y_{1}}(t) & =f_{1}\left(y_{1}, y_{2}, \ldots, y_{n}\right) \\
\dot{y_{2}}(t) & =f_{2}\left(y_{1}, y_{2}, \ldots, y_{n}\right) \\
. . & . . \\
\dot{y_{n}}(t) & =f_{n}\left(y_{1}, y_{2}, \ldots, y_{n}\right)
\end{aligned}
$$

- There is no general solution method for nonlinear differential equations. However, one way to understand their local behaviour around particular points is to use a multivariate Taylor approximation to linearise the functions around a particular point, a. This will give something of the format

$$
\dot{y}=f(a)+J(a) y
$$

where $J(a)$ is a so-called Jacobian matrix of partial derivatives of the $f$ functions evaluated at a. As a linear system, this can be solved to get a sense of how they model's dynamics behave in the region of $a$.

## Example of Local Approximation

- Consider the example

$$
\begin{aligned}
& \dot{x}_{1}(t)=F_{1}\left(x_{1}, x_{2}\right)=-\log x_{1}-\log x_{2} \\
& \dot{x}_{2}(t)=F_{2}\left(x_{1}, x_{2}\right)=-x_{2}^{2}-2 x_{2}+3
\end{aligned}
$$

- This system has an equilibrium at $\left(x_{1}, x_{2}\right)=(1,1)$. To find out whether this is a stable equilibrium, we linearise the $F$ functions around $(1,1)$ as follows

$$
\binom{\dot{x}_{1}}{\dot{x}_{2}}=\binom{F_{1}(1,1)}{F_{2}(1,1)}+\left(\begin{array}{ll}
\frac{\partial F_{1}(1,1)}{\partial x_{1}} & \frac{\partial F_{1}(1,1)}{\partial x_{2}} \\
\frac{\partial F_{2}(1,1)}{\partial x_{1}} & \frac{\partial F_{2}(1,1)}{\partial x_{2}}
\end{array}\right)\binom{x_{1}-1}{x_{2}-1}
$$

- We calculate these partial derivatives as follows

$$
\begin{aligned}
\frac{\partial F_{1}}{\partial x_{1}}=-\frac{1}{x_{1}} & \frac{\partial F_{2}}{\partial x_{2}}=-\frac{1}{x_{2}} \\
\frac{\partial F_{2}}{\partial x_{1}}=0 & \frac{\partial F_{2}}{\partial x_{2}}=-2 x_{2}-2
\end{aligned}
$$

## Local Dynamics Around the Equilibrium

- Evaluating these partial derivatives at $(1,1)$, our linearly approximated system becomes

$$
\binom{\dot{x}_{1}}{\dot{x}_{2}}=\binom{F_{1}(1,1)}{F_{2}(1,1)}+\left(\begin{array}{cc}
-1 & -1 \\
0 & -4
\end{array}\right)\binom{x_{1}-1}{x_{2}-1}
$$

- We calculate the eigenvalues by setting the determinant of the Jacobian matrix equal to zero.

$$
\left|\begin{array}{cc}
-1-\lambda & -1 \\
0 & -4-\lambda
\end{array}\right|=0
$$

- This implies

$$
(\lambda+1)(\lambda+4)=0
$$

so the system has two negative roots $\lambda_{1}=-1$ and $\lambda_{2}=-4$.

- This means the system is stable around the equilibrium $(1,1)$.


## Part VII

## Numerical Methods

## Finite Difference Methods

- It is nice to have a full analytical solution of our model. You can just evaluate the analytical formula at various points in time.
- But often this isn't possible or at least isn't possible without lots of analytical calculations.
- And in practice, often all we want is to figure out how our series behave over time. This can generally be done using numerical methods.
- One class of methods for solving systems of ordinary differential equations is finite-difference methods.
- The best known finite-difference method is the Euler method, which relies on the first-order Taylor series approximation:

$$
f(t+h) \approx f(t)+f^{\prime}(t) h
$$

and uses it to approximate derivatives as

$$
f^{\prime}(t) \approx \frac{f(t+h)-f(t)}{h}
$$

## Applying the Euler Method

- Consider a system of differential equations:

$$
\begin{aligned}
\dot{y_{1}}(t) & =f_{1}\left(y_{1}(t), y_{2}(t)\right) \\
\dot{y_{2}}(t) & =f_{2}\left(y_{1}(t), y_{2}(t)\right)
\end{aligned}
$$

- We can approximate this as

$$
\begin{aligned}
& \frac{y_{1}(t+h)-y_{1}(t)}{h}=f_{1}\left(y_{1}(t), y_{2}(t)\right) \\
& \frac{y_{2}(t+h)-y_{2}(t)}{h}=f_{2}\left(y_{1}(t), y_{2}(t)\right)
\end{aligned}
$$

- Which can be re-written as

$$
\begin{aligned}
y_{1}(t+h) & =y_{1}(t)+h f_{1}\left(y_{1}(t), y_{2}(t)\right) \\
y_{2}(t+h) & =y_{2}(t)+h f_{2}\left(y_{1}(t), y_{2}(t)\right)
\end{aligned}
$$

- This is a recursive system, meaning once we have calculated the values at time $t$, we can then move on to calculate the values at time $t+h$. We implement this by starting at some initial time $t_{0}$ and then calculating values for $t_{0}+h, t_{0}+2 h, t_{0}+3 h, \ldots . t_{0}+N h$.


## Example: Using Matlab to Simulate a System

- Consider our previous system

$$
\begin{aligned}
& \dot{x_{1}}(t)=F_{1}\left(x_{1}, x_{2}\right)=-\log x_{1}-\log x_{2} \\
& \dot{x}_{2}(t)=F_{2}\left(x_{1}, x_{2}\right)=-x_{2}^{2}-2 x_{2}+3
\end{aligned}
$$

- Here's a Matlab programme that simulates the system from arbitrary initial conditions. You can see how the model converges from each set of initial conditions towards its equilibrium point of $(1,1)$.

```
* Set latex as default interpreter
set(groot, 'defaulttextinterpreter','latex');
set(groot, 'defaultAxesTickLabelInterpreter','latex');
set(groot, 'defaultLegendInterpreter','latex');
T = 600; % Length of simulation
dt = 0.01; % Time step
time = dt*linspace (0,T,T);
xl = zeros(1,T);
x2 = zeros(1,T);
% Pick a random starting point
x1(1) = 2;;
x2(1) = 0.1;
for n=2:T
    xl(n) = xl(n-1) +dt* (-log(xl(n-1)) - log(x2(n-1)));
    x2(n)=x2(n-1)+dt** (-x2(n-1)^2 - 2*x2(n-1) + 3);
end
plot(time,xl,time,x2)
legend('$x_1$','$x_2$','Location','SouthEast','Fontsize',24)
```


## Simulation With $x_{1}(0)=2, x_{2}(0)=0.1$



## Simulation With $x_{1}(0)=0.1, x_{2}(0)=2$



## Applying the Euler Method to Higher-Order Derivatives

- The Euler method can be applied to differential equations involving higher-order derivatives by, for example, using the same approach to get an approximation to the the first derivative of the first derivative. For example, we can approximate the second derivative as

$$
\begin{align*}
f^{\prime \prime}(t+h) & \approx \frac{f^{\prime}(t+h)-f^{\prime}(t)}{h}  \tag{1}\\
& \approx \frac{f(t+2 h)-f(t+h)}{h^{2}}-\frac{f(t+h)-f(t)}{h^{2}}  \tag{2}\\
& \approx \frac{f(t+2 h)-2 f(t+h)+f(t)}{h^{2}} \tag{3}
\end{align*}
$$

- To implement this within a recursive structure, you will be implementing a second-order difference equation: To know $f(t+2 h)$, you need to know $f(t+h)$ and $f(t)$. This means you will need two initial conditions for each equation in your system.
- Applying the Euler method to calculate a third-order derivative you get an expression with three terms.

$$
f^{\prime \prime \prime}(t+h)=\frac{f(t+3 h)-3 f(t+2 h)+3 f(t+h)-f(t)}{h^{2}}
$$

## Other Finite Difference Methods

- Euler's method has the advantage of simplicity but it can be inaccurate in some cases.
- For example, our earlier application used $f_{1}\left(y_{1}(t), y_{2}(t)\right)$ to approximate changes between time $t$ and time $t+h$. If the function $f_{1}$ changes quickly, this might turn out to be an inaccurate assumption.
- There are many more sophisticated algorithms. One of them, the Runge-Kutta algorithm works like this when applied to our two variable example:

$$
\begin{aligned}
d y_{1} & =f_{1}\left(y_{1}(t), y_{2}(t)\right) \\
d y_{2} & =f_{2}\left(y_{1}(t), y_{2}(t)\right) \\
y_{1}(t+h) & =y_{1}(t)+0.5 * h\left[f_{1}\left(y_{1}(t)+d y_{1}, y_{2}(t)\right)+d y_{1}\right] \\
y_{2}(t+h) & =y_{2}(t)+0.5 * h\left[f_{2}\left(y_{1}(t), y_{2}(t)+d y_{2}\right)+h d y_{2}\right)
\end{aligned}
$$

- Applying this method to our recent example gives basically the same time path. But that is because the convergence paths are almost monotonic so the dynamics are fairly predictable. In other models with more complex dynamics, the Euler method may not work so well.

