Review (and Worksheet) of Basic Objects and Algorithms in Dynamical Models in Biology

By Dr. Z. (Doron Zeilberger)

Preliminaries

Mathematics consists of **objects**: numbers, functions, limits, equations, differential equations, difference equations, matrices, eigenfunctions, eigenvalues, etc. etc. .

It also consists of **algorithms** to handle, manipulate, and **solve** them.

DO NOT CONFUSE THE ALGORITHM WITH THE OBJECT!

I was disappointed that even some of the best students, when I asked them to check that a certain vector $\mathbf{v}$ is an eigenvector of a certain matrix A, instead of doing it directly, applying $A\mathbf{v}$ and checking whether it is a scalar multiple of $\mathbf{v}$, automatically performed the **algorithm** for finding all the eigenvalues and the corresponding eigenvectors. Technically it is a correct way (and I should have specified, 'without doing it from scratch') but it was still disappointing).

It is much more important to understand the **concepts**, then being able to perform the algorithms, that computers can do much faster and more reliably.

Analogy: Check that $x = 1$ is a solution of the equation $x^2 - 9x + 8 = 0$.

Correct but STUPID WAY: Using the formula for the quadratic equation we get

$$\frac{-(-9) \pm \sqrt{(-9)^2 - 4 \cdot 1 \cdot 8}}{2 \cdot 1} = \frac{9 \pm \sqrt{81 - 32}}{2 \cdot 1} = \frac{9 \pm \sqrt{49}}{2} = \frac{9 \pm 7}{2} = 8, 1 \quad .$$

Hence the roots of the equation are $x = 1$ and $x = 8$. Since $x = 1$ is one them, we solved this problem.

Correct and Smart way: The whole point of this problem was to make sure that you understand the **concept** of what it means for a **number** to be a **root** (or solution) of a given equation. Just plug-it-in!

$$1^2 - 9 \cdot 1 + 8 = 1 - 9 + 8 = 0 \quad .$$

Yea! $x = 1$ is indeed a solution of the equation $x^2 - 9x + 8 = 0$.

Basic Notion 1: NUMBER. The most fundamental numbers are integers, then rational numbers, (ratios of integers) but the numbers that show up in science are **real numbers**. **Complex numbers** are also useful, but they are really (usually) *theoretical* devices to help us say something about the real world. For example the criteria for stability of an equilibrium (discrete or continuous) dynamical system use complex numbers (the eigenvalues are often complex), but these are only theoretical devices. Even negative numbers are not realizable as quantities of species (of course if you talk about money, you can have a negative amount, in other words you can owe money).

Numbers can be added, multiplied, raised to powers etc. This is **numerics** that computers are very good at.

Basic Notion 2: (Algebraic) EQUATION (not to be confused with differential equation). An (algebraic, or if it involves exponential and trig functions, it is called transcendental) , is a puzzle.

'I am a number, I satisfy this equation. who can I be?'

As I said above, to solve an equation is often very hard, and even more often, **impossible** to solve exactly (but with computers we can get very good approximations, good enough for science and technology).

To check whether a candidate solution is indeed a solution, is easy! PLUG-IT-IN.

Do right now:

P1: Check whether $z = 2$ is a solution of the equation $z^3 + 3z^2 - 11z + 2 = 0$. Is $z = 3$ a solution?

P2: Check whether $z = \pi$ is a solution of the equation $\sin z = 0$, is $z = \frac{\pi}{2}$ a solution?

P3: Check whether $z = \frac{\pi}{3}$ is a solution of the equation $\sin^2 z + \cos^2 z = 1$, is $z = \frac{\pi}{5}$ a solution?

P4: Find the set of **all solutions** of the trig equation $\sin^2 z + \cos^2 z = 1$.

Basic notion 3: FUNCTION. The notion of function is very general.

$$f : A \to B$$

A is called the **domain** and B is called the **range**. If A is a finite set, it can be given in **table form**.

The functions that come up in dynamical systems always have the domain 'time'. If it is **continuous time**, the domain is the real axis (if you also care about the past) or the half-line $t \geq 0$ (if you only care about the future). The functions are usually denoted by $x(t)$, or if you have several variables, $x(t), y(t)$; $x(t), y(t), z(t)$; or, in general, $x_1(t), \ldots, x_k(t)$. Of course you can use any symbols to denote the dynamical functions.

If we have a **discrete time** dynamical system, the domain is the set of integers (if you also care about the past, but usually you don't), or, or more often, the set of non-negative integers. Such discrete functions are called **sequences**. In this case we can't talk about 'rate of change', but in a discrete-time dynamical system you talk about the evolution, how to get from the value today to the value tomorrow (see below).

Note that $x(t)$ and $x(n)$ are functions that are **solutions** of dynamical systems, and the domain is always 'time'. Either continuous time t, or discrete time n. In addition there is a completely different kind of functions, called **underlying** functions, whose domain is also R (but not meaning time), but, in this case 'the number (or concentration)' of some species (or of infected people, or of nutrients etc.). Don't confuse these two kinds of functions. When you have two species, you have functions from R^2 to R^2, and if you have k species, the underlying function is from R^k to R^k. When $k \geq 2$, it is called the **underlying transformation**.

Do right now:

P5: For the function $x(t) = t^4$, find its rate of change, and the rate of change of rate of change, when the time is $t = 2$.

Basic notion 4: Given a function (or transformation) with the same domain and range

$$f : A \rightarrow A \quad ,$$

a **fixed point** is a member of $a \in A$ such that $f(a) = a$.

Warning: the notion of 'fixed point' only makes sense if the domain and range are the same.

Do right now:

P6: Check whether $x = 1$, $x = 2$, $x = 3$, $x = -1$, are fixed points of the function (from R to R) $f(x) = (x - 1)\,(x - 2)\,(x - 3) + x$.

P7: Check whether the point $(x, y) = (0, -1)$ in R^2 is a fixed point of the transformation (from R^2 to R^2)

$$f(x, y) = (x + y + 1, x - y - 2) \quad .$$

Is $(x, y) = (1, 1)$ also a fixed point?

Basic notion 5: The **Orbit** (alias **trajectory**) of a function $f : A \rightarrow A$, starting at a_0.

It is

$$a_0, f(a_0), f(f(a_0)), f(f(f(a_0))), \ldots \quad .$$

If A is an infinite set (usually the case in applications) the orbit is infinite, so we humans can only find the first few terms (Maple can easily find the first 10000 terms). In the Maple package

`https://sites.math.rutgers.edu/~zeilberg/Bio21/DMB.txt` ,

it is given by the **very important** function `Orb(F,x,x0,K1,K2)`, that inputs `F` (the transformation), the list of variables , `x`, the *initial point* `x0` and integers `K1,K2` and outputs the segment of the orbit from $n = K1$ to $n = K2$.

Another way of thinking about the orbit (alias trajectory) is as the terms $n = K1$ through $n = K2$ of the first-order difference equation

$$x(n) = f(x(n-1)) \quad , \quad x(0) = x0 \quad .$$

Warning: If things take place in R the format is `Orb([f],[x],[x0],K1,K2)`.

Example: For the function $f(x) = 2\,x\,(1-x)$

(i) By hand (w/o computer) find the first three terms of the orbit starting at $x(0) = 0.4$.

(ii) Write down the Maple line to get the same answer.

(iii) Using Maple, write the `Orb` command to find the 1000-th term of the orbit, i.e., $x(1000)$. What is it?

Sol.

(i)

$x(0) = 0.4 \quad , \quad x(1) = 2{\cdot}0.4{\cdot}(1-0.4) = 2{\cdot}0.4{\cdot}0.6 = 0.48 \quad , \quad x(2) = 2{\cdot}0.48{\cdot}(1-0.48) = 2{\cdot}0.48{\cdot}0.52 = 0.4992 \quad ,$

(ii) `Orb([2*x*(1-x)],[x],[0.4],0,2);`

(ii) `Orb([2*x*(1-x)],[x],[0.4],1000,1000)[1];` , you get [0.5000000000].

Do right now

P8: For the function $f(x) = 1/(x+1)$

(i) By hand (w/o computer) find the first three terms of the orbit starting at $x(0) = 0.5$.

(ii) Write down the Maple line to get the same answer.

(iii) Using Maple, write the `Orb` command to find the 1000-th term of the orbit, i.e., $x(1000)$. What is it?

Another Example: For the transformation $f(x,y) = (\,\frac{x}{1+y}\,,\,\frac{y}{1+x})$

(i) By hand (w/o computer) find the first three terms of the orbit starting at $[1.0, 1.0]$.

(ii) Write down the Maple line to get the same answer.

(iii) Using Maple, write the `Orb` command to find the 1000-th term of the orbit. What is it?

Answer:

(i)

$$(1.0,1.0) \quad , \quad ; f(1.0,1.0) = (0.5,0.5) \quad , \quad f(0.5,0.5) = (0.3333\ldots, 0.3333\ldots) \quad .$$

Hence the first three terms, starting at $n = 0$ are (in Maple notation)

$$[[1.0,1.0],[0.5,0.5],[0.333\ldots,0.333\ldots]$$

(ii) `Orb([x/(1+y),y/(1+x)],[x,y],[1.0,1.0],0,2);` (you get the above output)

(iii) `Orb([x/(1+y),y/(1+x)],[x,y],[1.0,1.0],1000,1000)[1];`

You get `[0.0009990010090, 0.0009990010090]` .

P9 For the transformation $f(x,y,z) = (x/(1+y+z), y/(1+x+z), z/(1+x+y))$

(i) By hand (w/o computer) find the first three terms of the orbit starting at $[1.0,1.0,1.0]$.

(ii) Write down the Maple line to get the same answer.

(iii) Using Maple, write the `Orb` command to find the 1000-th term of the orbit. What is it?

Basic notion 6: a **first-order difference equation** aka **discrete-time first-order dynamical system** with **one quantity**.

A first-order difference equation has the **format**

$$x(n) = f(x(n-1)) \quad ,$$

where $f(x)$ is the **underlying function**. Often you are also given an **initial condition**, $x(0) = x_0$.

A famous example is the **discrete logistic equation** (studied by Robert May, Mitchell Feigenbaum, and others).

$$x(n) \; = \; k\, x(n-1)\,(1 - x(n-1)) \quad ,$$

where k is the **reproduction parameter**. In general there is no formula for $x(n)$, but you can easily get a *numerical solution*, starting with a given initial value $x(0) = x_0$. Say the first 10000 terms, using `Orb` or `OrbF`, that in application is enough to see what is going on.

Basic notion 7: An **equilibrium solution** of the first-order difference equation $x(n) = f(x(n-1))$ is a solution such that $x(n)$ is always the same. In other words it has the **format**

$$x(n) = c \quad ,$$

for $n = 0, 1, 2, \ldots$. Note that here we do not prescribe an initial value $x(0)$. To find all the equilibrium solutions, you solve the algebraic equation

$$c = f(c) \quad .$$

(In an equilibrium solution of the discrete dynamical system, $x(n) = c$ and $x(n-1) = c$, so plugging in x for $x(n)$ and $x(n-1)$ you get that equation.)

In other words 'equilibrium solution' of a first-order discrete dynamical system with one quantity is the same as **fixed-point of the underlying function**.

Sample Problem: Find all the equilibrium solutions of the discrete-time first-order dynamical system

$$x(n) \; = \; 2\,x(n-1)\,(1 - x(n-1)) \quad .$$

Sol.: The **underlying function** is

$$f(x) = 2x(1-x) \quad .$$

To get the equilibrium (i.e. constant) solutions, we solve

$$x = 2x(1-x) \quad ,$$

which is the same as

$$x - 2x(1-x) = x(1 - 2 + 2x) = x(-1 + 2x) = 0 \quad .$$

Getting two equilibrium solutions

$$x(n) = 0 \quad ,$$

$$x(n) = \frac{1}{2} \quad ,$$

(for all n).

Ans. to the problem: There are two equilibrium solutions, $x(n) = 0$ and $x(n) = \frac{1}{2}$.

Do right now

P11: Find all equilibrium solutions of the first-order discrete time dynamical system

$$x(n) = x(n-1)^2 - 2x(n-1) + 2$$

P12: Find all equilibrium solutions of the first-order discrete time dynamical system

$$x(n) = \frac{5}{2}x(n-1)(1-x(n-1)) \quad ,$$

P13:: More generally, find all equilibrium solutions of the first-order discrete time dynamical system with parameter k

$$x(n) = k\, x(n-1)(1-x(n-1)) \quad ,$$

(your answers (may) depend on k, of course)

Obviously for an equilibrium solution $x(n) = c$, the solution of the initial-value problem

$$x(n) \;=\; f(x(n-1)) \quad , \quad x(0) = c \quad ,$$

is itself, namely $x(n) = c$. In other words 'Once at $x = c$ always at $x = c$.

Basic notion 8: A **stable equilibrium solution**, $x(n) = c$, of the first-order difference equation $x(n) = f(x(n-1))$ is an equilibrium solution (see above) with the additional property that for some neighborhood of $x = c$, $c - \delta < x < c + \delta$) (often pretty large), the long-time behavior of the solution of the modified initial value problem

$$x(n) = f(x(n-1)) \quad , \quad x(0) = d \quad ,$$

for d in the interval $c - \delta < d < c + \delta$, in the long-run, goes back to c. Mathematically

$$\lim_{n \to \infty} x(n) \;=\; c \quad .$$

How to find whether an equilibrium solution is stable or not (using numerics)?

We use procedure `Orb`.

Sample problem': Using procedure `Orb` (or `OrbF`), to decide, for each of the equilibrium solutions of the discrete-time dynamical system

$$x(n) \;=\; 2\, x(n-1)\,(1 - x(n-1)) \quad ,$$

found above, namely $x(n) = 0$ and $x(n) = 0.5$, decide whether it stable or unstable.

Sol. to sample problem:

```
Orb([2*x*(1-x)],[x],[0],1000,1010);
```

gives

```
[[0], [0], [0], [0], [0], [0], [0], [0], [0], [0], [0]]
```

confirming that indeed $x(n) = 0$ is an equilibrium solution (alias a fixed point of the underlying function). Now try

```
Orb([2*x*(1-x)],[x],[0.01],1000,1010);
```

and you get

```
[[0.5000000000], [0.5000000000], [0.5000000000], [0.5000000000], [0.5000000000], [0.5000000000
[0.5000000000], [0.5000000000], [0.5000000000], [0.5000000000], [0.5000000000]]
```

so the limit does not go to 0 but to the other equilibrium point.

Even starting ever so close to $x = 0$, for example $x(0) = 0.001$, gives the same thing. This indicates that the equilibrium solution $x(n) = 0$ is **not** stable. It is an **unstable** equilibrium solution.

Now let's investigate the other equilibrium solution $x(n) = 0.5$.

```
Orb([2*x*(1-x)],[x],[0.5],1000,1010);
```

gives you

```
[[0.5000000000], [0.5000000000], [0.5000000000], [0.5000000000], [0.5000000000], [0.5000000000
[0.5000000000], [0.5000000000], [0.5000000000], [0.5000000000], [0.5000000000]]   ,
```

as it should, confirming that $x(n) = 0.5$ is an equilibrium solution. But now set $x(0) = 0.6$ and type

```
Orb([2*x*(1-x)],[x],[0.6],1000,1010);   ,
```

and you get the same thing.

Also

```
Orb([2*x*(1-x)],[x],[0.4],1000,1010);   ,
```

gives you the same thing, so **in the long-run** (even if you go pretty far from $x = 0.5$ in this case) you go back to $x = 0.5$, *sooner or later* (in this particular case, pretty soon!)

Comment: To be honest, you never actually make it **exactly** back to $x = \frac{1}{2}$, but you get *ever-so-close*, so Maple can't distinguish it with its decimal approximations.

Hence, using *numerics* we found out that $x = \frac{1}{2}$ is a **stable fixed point** of our discrete-time first-order dynamical system, $x(n) = 2\,x(n-1)\,(1 - x(n-1))$.

Do right now

P11': Using **numerics** Find all **stable** equilibrium solutions of the first-order discrete-time dynamical system

$$x(n) = x(n-1)^2 - 2x(n-1) + 2$$

(You should use what you found in P11 above, as the starting points (don't do it again)), but decide for each of them whether or not it is stable.

P12"' Using **numerics**, find all **stable** equilibrium solutions of the first-order discrete time dynamical system

$$x(n) = \frac{5}{2}x(n-1)(1-x(n-1)) \quad ,$$

(You should use what you found in P12 above, as the starting points (don't do it again)), but decide for each of them whether or not it is stable.

How to find whether an equilibrium solution is stable or not (using calculus)

Comment: As I said above, this is an **algorithm** to decide **conclusively** whether an equilibrium solution of a first-order, one-quantity, discrete dynamical system is stable or not. It is not to be confused with the **concept**, that the numeric way illustrates so well. Unlike the numerics that sometimes can fool you (in some cases the stable fixed point has a very small 'basin of attraction', so you may miss it), this is completely safe. The drawback is that you need to differentiate (but Maple does not mind).

• Decide on the **underlying function**, let's call it $f(x)$.

• compute $f'(x)$, getting (usually) an expression in x (you may get a constant sometimes, i.e. that the derivative is a constant function)

• plug-in the candidate equilibrium solution (alias fixed point, i.e. the number c that you already know is an equilibrium from the algebraic step), into $f'(x)$, getting the **number** $f'(c)$.

If $|f'(c)| < 1$ then $x = c$ is a **stable** equilibrium solution.

If $|f'(c)| \geq 1$ then $x = c$ is an **unstable** equilibrium solution.

Comment: If $|f'(c)| = 1$, then it is called **semi-stable**.

Another comment: This rule follows from 'linearization'. If $f(c) = c$ then $f(x) - f(c) = f(x) - c$ is 'close' to $f'(c)(x - c)$.

Sample problem": Using calculus, decide **conclusively** whether the equilibrium solutions

$$x(n) = 2\,x(n-1)\,(1 - x(n-1)) \quad ,$$

found above, namely $x(n) = 0$ and $x(n) = 0.5$ are stable or not.

Sol. The underlying function is

$$f(x) = 2x(1-x) = 2x - 2x^2 \quad .$$

Differentiate

$$f'(x) = 2 - 4x \quad .$$

Regarding $x = 0$, we get

$$f'(0) = 2 \quad ,$$

Since $|2| = 2$ is **larger** than 1, $x = 0$ is **unstable**.

Regarding $x = \frac{1}{2}$, we get

$$f'(\frac{1}{2}) = 2 - 4 \cdot \frac{1}{2} = 0 \quad ,$$

Since $|0| = 0$ is **smaller** than 1, $x = \frac{1}{2}$ is **stable**.

Do right now

P11" Using **calculus** Find all **stable** equilibrium solutions of the first-order discrete time dynamical system

$$x(n) = x(n-1)^2 - 2x(n-1) + 2$$

(You should use what you found in P11 above, as the starting points (don't do it again)), but decide for each of them which is stable and which ones is not)

P12" Using **calculus**, find all **stable** equilibrium solutions of the first-order discrete time dynamical system

$$x(n) = \frac{5}{2}x(n-1)(1 - x(n-1)) \quad ,$$

(You should use what you found in P12 above, as the starting points (don't do it again)), but decide for each of them which is stable and which ones is not)

Basic notion 9: a **first-order differential equation** aka **continuous-time first-order dynamical system** with **one quantity**.

A first-order differential equation has the **format**

$$x'(t) = f(x(t)) \quad ,$$

where $f(x)$ is the **underlying function**. Often you are also given an **initial condition**, $x(0) = x_0$.

A (biologically boring) example is the **continuous logistic equation**

$$x'(t) \;=\; k\,x(t)\,(1-x(t)) \quad ,$$

where k is the **reproduction parameter**.

In some cases, it is possible to find an **explicit** solution to an initial value problem

$$x'(t) = f(x(t)) \quad , \quad x(0) = x_0 \quad ,$$

either by hand, or using Maple's `dsolve`. But often not even Maple can do it. Nevertheless we can investigate the long-term behavior.

Basic notion 10: An **equilibrium solution** of the first-order differential equation

$$x'(t) = f(x(t)) \quad ,$$

is a solution such that $x(t)$ is always the same. In other words, it is a **constant function**. it has the **format**

$$x(t) = c \quad ,$$

for all real numbers t, for *some* number c. Note that here we do not prescribe an initial value $x(0)$. To find all the equilibrium solutions, you solve the algebraic equation

$$0 = f(c) \quad .$$

(Since an equilibrium solution has the form $x(t) = c$, for *some* constant, c, $x'(t)$ is **automatically** 0, *no matter* what c happens to be.)

Warning: Do not confuse with the way to determine equilibrium solutions to discrete-time dynamical systems! (where you solve $c = f(c)$)!

Using **discretization** we can approximate a continuous-time dynamical system by a discrete one, and then use `Orb` or `OrbF`. This is implemented in procedure `TimeSeries` in `DMB.txt`.

Sample Problem: Find all the equilibrium solutions of the continuous-time first-order dynamical system

$$x'(t) \;=\; 2\,x(t)\,(1-x(t)) \quad .$$

Sol. The **underlying function** is $f(x) \;=\; 2\,x\,(1-x)$. Setting it equal to 0 gives the algebraic equation

$$2x(1-x) = 0 \quad .$$

Solving it, gives $x = 0$ and $x = 1$. Hence there are two equilibrium solutions to this continuous-time dynamical system, $x(t) = 0$ and $x(t) = 1$.

Ans.: The two equilibrium solutions to the continuous-time dynamical system $x'(t) = 2\,x(t)\,(1 - x(t))$ are $x(t) = 0$ and $x(t) = 1$.

The conceptual definition of what it means for an equilibrium solution to be stable (NOT to be confused with the algorithm for deciding it, recalled below)

Given an equilibrium solution (found as above), $x(t) = c$, to the dynamical system $x'(t) = f(x(t))$, it means that the (unique!, it is always unique if the underlying function $f(x)$ is 'nice') solution of the initial value problem

$$x'(t) = f(x(t)) \quad , \quad x(0) = c \quad ,$$

is the constant function $x(t) = c$. It is a **stable** equilibrium solution if there is *some* interval around $x = c$,

$$c - \delta < x < c + \delta \quad ,$$

such for each number d in that neighborhood of $x = c$, i.e. for each number such that $c-\delta < d < c+\delta$ the unique solution of the modified initial value problem

$$x'(t) = f(x(t)) \quad , \quad x(0) = d \quad ,$$

has the property that in the **long-run** it gets ever-so-close to the horizontal line $x = c$. Formally

$$\lim_{t\to\infty} x(t) \ = \ c \quad .$$

Graphically, if you plot it, there is a **horizontal asymptote** at height c above the t-axis.

How to investigate whether a given equilibrium solution is stable using Maple's dsolve?

Let's illustrate it with the above continuous-time dynamical system $x'(t) = 2\,x(t)\,(1 - x(t))$ where we already know that $x(t) = 0$ and $x(t) = 1$ are equilibrium solutions.

Let's investigate $x(t) = 0$.

```
dsolve(diff(x(t),t)=2*x(t)*(1-x(t)),x(0)=0,x(t));
```

gives you, not surprisingly, $x(t) = 0$. Now let's 'tweak' the initial condition, and instead of $x(0) = 0$ let's take $x(0) = 0.1$. Typing, in a Maple worksheet,

```
dsolve(diff(x(t),t)=2*x(t)*(1-x(t)),x(0)=0.1,x(t));
```

gives

$$x\,(t) = \frac{1}{1 + 9\,e^{-2t}} \quad ,$$

whose limit, as t goes to infinity is 1 **not** 0 (note that it goes to the other equilibrium solution). This indicates that $x(t) = 0$ is **not** stable!

Now let's investigate the other equilibrium solution, $x(t) = 1$.

```
dsolve(diff(x(t),t)=2*x(t)*(1-x(t)),x(0)=1,x(t));
```

gives you, not surprisingly, $x(t) = 1$. Now let's 'tweak' the initial condition, and instead of $x(0) = 0$ let's take $x(0) = 1.1$. Typing, in a Maple worksheet,

```
dsolve(diff(x(t),t)=2*x(t)*(1-x(t)),x(0)=1.1,x(t));
```

gives

$$x(t) = \frac{11}{11 - e^{-2t}}$$

whose limit, as t goes to infinity is 1. How about the initial condition $x(0) = 0.9$ (slightly below 1)? (you do it!) you should also get that the limit is 1. Hence we deduce that $x(t) = 1$ is a stable equilibrium solution.

How to investigate whether a given equilibrium solution is stable using TimeSeries in DMB.txt?

`dsolve` can't solve many (more complicated) differential equation, but to get approximate plots for the above scenarios you type:

```
TimeSeries([2*x*(1-x)],[x],[1.1],0.01,10,1);
```

(for the initial value $x(0) = 1.1$ and

```
TimeSeries([2*x*(1-x)],[x],[0.9],0.01,10,1);
```

(for the initial value $x(0) = 0.9$ and in both cases you get a horizontal asymptote $x = 1$. (Do it!)

How to conclusively decide whether an equilibrium solution is stable or not?

• Once you found the **underlying function**, $f(x)$, take its derivative, $f'(x)$, getting some expression in x.

• For each equilibrium solution $x(t) = c$, that you found before by solving the algebraic equation $f(c) = 0$, you **plug-in** $x = c$, and look at the **number** $f'(c)$.

• If $f'(c) < 0$ (i.e. if it is **negative**) then $x(t) = c$ is **stable**.

• If $f'(c) \geq 0$ then it is **unstable**.

In the border-line case of $f'(c) = 0$ it is called **semi-stable**.

Comment: Later on when we do systems with more than one quantity, we have to consider the Jacobian matrix at the candidate equilibrium solution and then the condition is (see below) 'all eigenvalues must have negative real part'. Right now the 'Jacobian matrix' is the 1×1 matrix $(f'(x))$, and at the candidate equilibrium solution $x(t) = c$, the **numerical** 1×1 matrix $(f'(c))$

whose 'eigenvalue' is $f'(c)$, and if the system came from the real world, this is never complex, so in this case 'negative real part' is the same as 'negative'.

Sample Example: For each of the two equilibrium solutions of the one-quantity continuous-time dynamical system, $x'(t) = 2\,x(t)(1-x(t))$, namely $x(t)=0$ and $x(t)=1$, decide whether or not they are stable.

Sol. The underlying function is

$$f(x) = 2x(1-x) = 2x - 2x^2 \quad .$$

Taking derivative with respect to x gives

$$f'(x) = 2 - 4x \quad .$$

Regarding $x(t)=0$, we have

$$f'(0) = 2 - 4 \cdot 0 = 2 \quad ,$$

since this **not** negative $x(t)=0$ is an **unstable** equilibrium solution.

Regarding $x(t)=1$, we have

$$f'(1) = 2 - 4 \cdot 1 = -2 \quad ,$$

since this **is** negative $x(t)=1$ is a **stable** equilibrium solution.

Do right now

P14:

(i) Find all the equilibrium solutions of the continuous-time dynamical system (with one quantity, $x(t)$)

$$x'(t) \;=\; 2\,x(t)\,(1-x(t))\,(2-x(t))\,(3-x(t)) \quad .$$

(ii) Using `TimeSeries` with $h=0.01$, investigate numerically, for **each** of the equilibrium solutions that you found in (i), whether it seems to be stable or not (look at the horizontal asymptotes, it they exist)

(iii) Using the algorithm to decided stability, conclusively decide, for each of the equilibrium solutions that you found in (i), whether it is stable or not. Explain!

Basic notion 12: A **discrete-time dynamical system** with **two** (changing) quantities.

Often, in applications, there are two or more species. For the sake of clarity, in these review notes, we will only talk about **two** quantities, and indeed, many dynamical systems in practice do have only two quantities (e.g. 'predators and prey', 'infected and susceptible' etc.). The theory and concepts are valid for *any* (finite) number of different 'species'.

Let's call these two quantities $x(n)$ and $y(n)$. This means that $x(n)$, and $y(n)$ are the value of the 'species' at generation n. A first-order discrete-time system has the **format**

$$x(n) = f(x(n-1), y(n-1)) \quad ,$$
$$y(n) = g(x(n-1), y(n-1)) \quad .$$

This means that the values of x and y at any given generation depends on **both** their values in the previous generation. The **underlying transformation** from R^2 to R^2 is

$$(x, y) \to (f(x, y), g(x, y)) \quad .$$

The **orbit** staring at $[x_0, y_0]$ is defined as above.

Example: By hand-calculations, find the first three terms of the orbit, starting at $n = 0$ of the following discrete-time dynamical system, if $x(0) = 1, y(0) = 2$. Confirm it with the output of `Orb`.

$$x(n) = x(n-1)^2 + y(n-1) \quad , \quad y(n) = x(n-1) + y(n-1)^2 \quad .$$

Sol.

$$x(1) = x(0)^2 + y(0) = 3 \quad , \quad y(1) = y(0)^2 + x(0) = 5 \quad ,$$

$$x(2) = x(1)^2 + y(1) = 9 + 15 = 14 \quad , \quad y(2) = y(1)^2 + x(1) = 25 + 3 = 28 \quad ,$$

Ans. $[[1, 2], [3, 5], [14, 28]]$. Now in `DMB.txt`, type:

```
Orb([x**2+y,x+y**2],[x,y],[1,2],0,2);
```

getting the same thing.

Do right now

P15: By hand-calculations, find the first four terms of the orbit, starting at $n = 0$ of the discrete-time dynamical system, if $x(0) = 1, y(0) = 3$. Confirm it with the output of `Orb`.

$$x(n) = x(n-1)^3 + 2y(n-1) \quad , \quad y(n) = x(n-1)^2 + 5y(n-1)^2 \quad ,$$

Basic notion 13: Equilibrium solution of **discrete-time dynamical system** with **two** (changing) quantities. These are solutions where **both** $x(n)$ and $y(n)$ **never change**, i.e. they are two numbers c and d such that **for every** n,

$$x(n) = c \quad , \quad y(n) = d \quad .$$

How to use algebra to find the Equilibrium solution of a discrete-time dynamical system with two species

Recall that such a system has the format

$$x(n) = f(x(n-1), y(n-1)) \quad ,$$
$$y(n) = g(x(n-1), y(n-1)) \quad .$$

Plugging-in, $x(n) = c$, $x(n-1) = c$; $y(n) = d$, $y(n-1) = d$ (recall that the solutions that we are interested in **never** change, so the values at generation $n-1$ is the same as the respective values in generation n)

$$c = f(c, d) \quad ,$$
$$d = g(c, d) \quad .$$

In other words, the point, in R^2, (c, d) is a **fixed point** of the **underlying transformation** $(x, y) \to (f(x, y), g(x, y))$. Usually this is too hard to solve by hand, but procedure `FP` in `DMB.txt` can do it for you.

Basic notion 14: What does it mean for an Equilibrium solution of **discrete-time dynamical system** to be **stable**? (the concept!, not 'how to decide it').

Obviously, if $x(0) = c$ and $y(0) = d$, and you find the orbit, say up to 10000, it is **always** the same, by definition! But now 'tweak' the initial conditions, say make them $x(0) = c + 0.1$ and $y(0) = d - 0.1$, and run the orbit to 10000. If it is stable, then *in the long run*, $x(n)$ will get ever-so-close to c, and $y(n)$ will get ever so close to d.

Example: Consider the discrete-time system with two quantities

$$x(n) = \frac{3 + x(n-1) + 2y(n-1)}{1 + 2x(n-1) + y(n-1)} \quad ,$$
$$y(n) = \frac{1 + 3x(n-1) + 2y(n-1)}{2 + 2x(n-1) + 3y(n-1)} \quad .$$

The underlying transformation is

$$(x, y) \to \left(\frac{3 + x + 2y}{1 + 2x + y}, \frac{1 + 3x + 2y}{2 + 2x + 3y} \right) \quad .$$

Using `SFP` we type:

```
F:=[(3+x+2*y)/(1+2*x+y), (1+3*x+2*y)/(2+2*x+3*y)]   ,
```

and then

```
SFP(F,[x,y]);
```

we get

$\{[1.342780106, 0.9222435790]\}$

Now let's check this using `Orb`, with initial conditions close, but not the same, how about $[1.0, 1.0]$. Typing

```
Orb(F,[x,y],[1.0,1.0],1000,1002);
```

we get

```
[[1.342780106, 0.9222435789], [1.342780106, 0.9222435789], [1.342780106, 0.9222435789]]
```

so indeed, it tends to the above values!

Do Right Now

P16: Using `SFP` find the stable equilibrium solutions of the discrete-time dynamical system with two quantities

$$x(n) = \frac{2 + x(n-1) + y(n-1)}{2 + 2x(n-1) + 2y(n-1)} \quad ,$$

$$y(n) = \frac{2 + x(n-1) + y(n-1)}{1 + 2x(n-1) + 2y(n-1)} \quad .$$

Make sure that it is $[0.6953496364, 0.8641637014]$, then confirm it numerically, by using `Orb` with $x(0) = 0.5$ and $y(0) = 0.4$.

How to conclusively decide whether an equilibrium solution of a discrete-time dynamical system is stable or not

Suppose that the underlying transformation is

$$(x, y) \to (f(x, y), g(x, y)) \quad .$$

and you found out that for some **numbers** c and d, the constant functions $x(t) = c$ and $y(t) = d$ is a fixed point, i.e. $f(c, d) = c, g(c, d) = d$.

• Find the Jacobian matrix, in general

$$J(x, y) = \begin{pmatrix} f_x(x, y) & f_y(x, y) \\ g_x(x, y) & g_y(x, y) \end{pmatrix}$$

• plug-in the numbers $x = c$, $y = d$, getting a **numerical** matrix, $J(c, d)$.

• Find the eigenvalues.

If all the eigenvalues have absolute value less than 1, $x(t) = c, y(t) = d$ is a stable equilibrium. Otherwise is is unstable.

Basic notion 15: A **continuous-time dynamical system** with **two** (changing) quantities.

The format is

$$x'(t) = f(x(t), y(t)) \quad ,$$
$$y'(t) = g(x(t), y(t)) \quad .$$

where $f(x, y)$ and $g(x, y)$ are functions of two variables. If you are given **initial conditions** $x(0) = x_0$, $y(0) = y_0$ there there is a unique solution

$(x(t), y(t))$ that satisfies the system.

The **underlying transformation** is

$$(x, y) \to (f(x, y), g(x, y)) \quad .$$

To get the graph of $x(t)$ using `TimeSeries` from $t = 0$ to $t = A$ in `DMB.txt`, type:

```
TimeSeries([f(x,y),g(x,y)],[x,y],[x0,y0], 0.01, A ,1);
```

To get the graph of $y(t)$ using `TimeSeries` from $t = 0$ to $t = A$ in `DMB.txt`, type:

```
TimeSeries([f(x,y),g(x,y)],[x,y],[x0,y0], 0.01, A ,2);
```

How to get the equilibrium solution of a continuous-time dynamical system with two quantities?

Assuming that there exist **constants** c and d such that the pair $x(t) = c$ and $y(t) = d$ is a solution, since for such constant functions, $x'(t) = 0$, and $y'(t) = 0$, the system becomes

$$0 = f(c, d) \quad ,$$
$$0 = g(c, d) \quad .$$

In other words, you have to solve the system of two algebraic equations with two unknowns $\{c, d\}$:

$$\{f(c, d) = 0 \quad , \quad g(c, d) = 0\} \quad .$$

Basic notion 16: what does it mean for an equilibrium solution to be stable?

Suppose that you did the previous step, and found out that $x(t) = c, y(t) = d$, is an equilibrium solution (found using `EquP` in `DMB.txt`). This means that if the initial conditions are **exactly** $x(0) = c, y(0) = d$ then the solution is **exactly** $x(t) = c$ and $y(t) = d$ for **ever after**.

If there is a $\delta > 0$ such that whenever $c - \delta < c' < c + \delta$, $d - \delta < d' < d + \delta$ we have the property that any solution of the continuous-time dynamical system

$$x'(t) = f(x(t), y(t)) \quad ,$$

$$y'(t) = g(x(t), y(t)) \quad ,$$

with the 'tweaked' initial conditions $x(0) = c'$, $y(0) = d'$, have the property that

$$\lim_{t\to\infty} x(t) \; = \; c \quad ,$$

$$\lim_{t\to\infty} y(t) \; = \; d \quad .$$

In words: Even you don't start exactly at $x = c, y = d$ at $t = 0$, but at a close location, in the long-run you will wind-up as close to (c, d) as you wish.

If that happens the examined equilibrium solution $x(t) = c, y(t) = d$ is a **stable equilibrium**. If $x(t)$ and/or $y(t)$ 'blow up', or oscillate, or end to other values than c and d respectively, when the initial conditions are close (but not exactly the same) as (c, d), then the examined equilibrium solution (or 'point' in the 'phase diagram') is **unstable**.

How to check numerically for stability using `TimeSeries`?

Let's illustrate it with an example. Consider the continuous-time dynamical system

$$x'(t) = (1 - 2x(t) - 3y(t))\,(2 - 2x(t) - 3y(t)) \quad ,$$

$$y'(t) = (3 - x(t) - 2y(t))\,(1 - x(t) - 2y(t)) \quad .$$

The **underlying transformation** is

$$[(1 - 2x - 3y)\,(2 - 2x - 3y)\,, (3 - x - 2y)\,(1 - x - 2y)]$$

Typing

```
EquP([(1-2*x-3*y)*(2-2*x-3*y), (3-x-2*y)*(1-x-2*y)],[x,y]);    ,
```

tells you that the equilibrium solutions are

$$[-7,5],[-5,4],[-1,1],[1,0] \quad .$$

Let's examine $[-7,5]$ meaning the equilibrium solution $x(t)=-7$, $y(t)=5$.

```
TimeSeries([(1-2*x-3*y)*(2-2*x-3*y), (3-x-2*y)*(1-x-2*y)],[x,y],[-7,5],0.01,10,1);
```

gives you the horizontal line $x=-7$, (i.e. the constant function $x(t)=7$), and typing

TimeSeries([(1-2*x-3*y)*(2-2*x-3*y), (3-x-2*y)*(1-x-2*y)],[x,y],[-7,5],0.01,10,2);

gives you the horizontal line $y=5$, (i.e. the constant function $y(t)=5$). Now try

```
TimeSeries([(1-2*x-3*y)*(2-2*x-3*y), (3-x-2*y)*(1-x-2*y)],[x,y],[-7.1,5.1],0.01,10,1);
```

(i.e. just a little away from $[-7,5]$ and it blows up! So we found numerically that $x(t)=-7, y(t)=5$ is an **unstable** equilibrium solution,

Let's examine $[-1,1]$ meaning the equilibrium solution $x(t)=-1$, $y(t)=1$.

```
TimeSeries([(1-2*x-3*y)*(2-2*x-3*y), (3-x-2*y)*(1-x-2*y)],[x,y],[-1,1],0.01,10,1);
```

gives you the horizontal line $x=-1$, (i.e. the constant function $x(t)=-1$, and typing

TimeSeries([(1-2*x-3*y)*(2-2*x-3*y), (3-x-2*y)*(1-x-2*y)],[x,y],[-1,1],0.01,10,2);

gives you the horizontal line $y=1$, (i.e. the constant function $y(t)=1$. As it should! Now try

```
TimeSeries([(1-2*x-3*y)*(2-2*x-3*y), (3-x-2*y)*(1-x-2*y)],[x,y],[-1.1,1.1],0.01,20,1);
```

and you get a curve that clearly has a horizontal asymptote at $x=-1$, i.e. in the long-run it goes to -1.

Similarly,

```
TimeSeries([(1-2*x-3*y)*(2-2*x-3*y), (3-x-2*y)*(1-x-2*y)],[x,y],[-1.1,1.1],0.01,20,2);
```

has a horizontal asymptote at $y=1$. This is a **numerical indication** that $x(t)=-1, y(t)=1$ is a **stable** equilibrium solution.

Do right now

P17: Use `DMB.txt` to convince yourself, numerically, that the other two equilibrium solutions $[-5,4]$ and $[1,0]$ are unstable.

How to determine conclusively whether an equilibrium solution of a continuous-time dynamical system is stable?

Suppose that the underlying transformation is

$$(x, y) \rightarrow (f(x, y), g(x, y)) \quad .$$

and you found out that for some **numbers** c and d, the constant functions $x(t) = c$ and $y(t) = d$ is a solution to the system (recall that such constant solutions are called 'equilibrium solutions'), in other words

$$f(c, d) = 0 \quad , \quad g(c, d) = 0 \quad .$$

• Find the Jacobian matrix, in general

$$J(x, y) = \begin{pmatrix} f_x(x, y) & f_y(x, y) \\ g_x(x, y) & g_y(x, y) \end{pmatrix} \quad .$$

• Plug-in the numbers $x = c$, $y = d$, getting a **numerical** matrix, $J(c, d)$.

• Find the eigenvalues.

If all the eigenvalues have **negative real part**, then the examined equilibrium solution is stable, otherwise not.

Comment: This is implemented in `SEquP` in the Maple package `DMB.txt`.