```julia
using Plots
```

```julia
begin

    using PlutoUI
    using LaTeXStrings
    using Roots
end
```

Nonlinear dynamics: stability and bifurcations

In this module on climate modelling we will be studying various **differential equations** that model the evolution of aspects of the climate over time.

The simplest such models are **ordinary differential equations** (ODEs), where one or a few continuous variables evolve continuously in time, with a model that specifies their instantaneous rate of change as a function of their current values.

The general form of an ODE (that doesn't depend explicitly on time, i.e. that is **autonomous**), is

$$\frac{dx(t)}{dt} = f(x(t)),$$

or

$$\dot{x}(t) = f(x(t)),$$

with an initial condition $x(t = 0) = x_0$.

Here, $\dot{x}(t)$ denotes the derivative of the function $t \mapsto x(t)$ at time t.

The simplest numerical method to solve such an equation is the **(forward) Euler method**, in which we convert this equation into an explicit time-stepping routine: we take a small time step of length h and approximate the derivative as

$$\frac{dx(t)}{dt} \simeq \frac{x(t+h) - x(t)}{h},$$

giving the approximation

$$x(t+h) \simeq x(t) + h\, f(x(t))$$

for the approximation at the next time step.

Solving the ODE: Euler method

Let's use this to simulate a simple nonlinear ODE that describes the dynamics of a population of bacteria. The bacteria will grow by reproduction at a rate λ provided there is sufficient food, in which case we would have $\dot{x} = \lambda x$. But the available food will actually always limit the sustainable population to a value K. A simple model for this is as follows:

$$\dot{x} = \lambda \, x \, (K - x).$$

When x is close to 0, the growth rate is λ, but that rate decreases as x increases.

This is sometimes called the **logistic differential equation** (although the name does not seem particularly helpful).

Our goal is to use computational thinking, but we will actually not be interested so much in the exact dynamics in time, but rather in the **qualitative** features of the behaviour of the system. For example, at long times (formally $t \to \infty$) does the population get arbitrarily large? Or does it, for example, oscillate around a particular value? Or does it converge to a particular size? This forms the subject of **nonlinear dynamics** or **dynamical systems** theory.

Let's simulate the system using the Euler method to try to guess the answer to this question. Note that there are many much more sophisticated methods for solving ODEs collected in the DifferentialEquations.jl package. We should never use the Euler method in practice, but should use a tried and tested library instead, and algorithms that provide much better accuracy in the solutions, if we are interested in faithful numerical results.

We'll rescale the variables to the simplest form:

$$\dot{x} = x \, (1 - x).$$

```
logistic (generic function with 1 method)
```

```julia
logistic(x) = x * (1 - x)
```

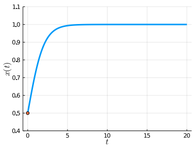
Let's simulate this with Euler and plot the trajectory $x(t)$ as a function of time t:

```
results =
  (ts = Float64[0.0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0
```

```julia
results = euler(logistic, 0.5, 0.01, 20.0)
```

We see that for this particular initial condition, the solution seems to settle down to a fixed value after some time, and then remains at that value thereafter.

Such a value is called a **fixed point** or a **stationary point** of the ODE.

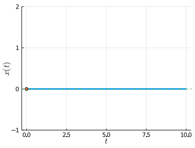


Qualitative behaviour: Fixed points and their stability

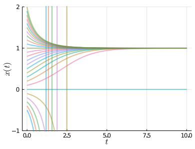
Let's see if what happens for other initial conditions:

0.0

```
@bind x0 Slider(-1.0:0.001:20, default=0.0, show_value=true)
```



To get an overview of the behaviour we can also draw all the results on a single graph:



We see that all the curves starting near to $x_0 = 1.0$ seem to converge to 1 at long times. If the system starts *exactly* at 0 then it stays there forever. However, if it starts close to 0, on either side, then it moves *away* from 0 (on that same side of 0) – starting from a negative value x becomes ever more negative. (Even though negative populations have no meaning in the original interpretation as the dynamics of a population, we can still ask study the dynamics of the equation with negative initial conditions, since it may model other systems too.)

The special values $x_1^* = 1$ and $x_2^* = 0$ are called **stationary points** or **fixed points** of the differential equation. If we start at x_i^*, then the derivative there is $f'(x_i^*) = 0$, and hence we cannot move away from x_i^*! The fixed points can be found as zeros or **roots** of the function f, i.e. values x^* such that $f(x^*) = 0$.

We see, though, that the two types of fixed points are **qualitatively different**: trajectories that start close to $x_1^* = 1$ move *towards* x_1^*, whereas trajectories that start close to $x_2^* = 0$ move *away* from it. We say that x_1^* is a **stable fixed point** and x_2^* is an **unstable fixed point**.

In general it is not possible to find analytical formulas for the position and stability of fixed points; instead, we can use numerical **root-finding algorithms**, for example the Newton method: see e.g. course 18.330 next semester. There are various Julia packages for root finding; we will use `Roots.jl`.

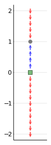
State space: Vector field and phase portrait

If we want to find the whole trajectory for a given initial condition then we need to solve the equations, either numerically or analytically.

However, we may want less information about the system, for example the **long-time** or **asymptotic** dynamics. It turns out that we can obtain some information about that *without* explicitly solving the ODE! This is the **qualitative approach** to studying nonlinear systems.

Instead of drawing trajectories $x(t)$ as a function of time t, as we did above, let's use a different graphical representation, where we draw **state space** or **phase space**: This is the set ("space") of all possible values of the dependent variables ("states"). For the above ODE there is only a single dependent variable, x, so the state space is the real line, $\mathbb{R}$.

At each possible value of x, the ODE gives us information about the rate of change of $x(t)$ at that point. Let's draw an **arrow** at that point, pointing in the direction that a particle placed at that point would move: to the right if $\dot{x} > 0$ and to the left if $\dot{x} < 0$.



```
vector_field(logistic)
```

This vector field indeed gives us a *qualitative* picture of the dynamics. It does not tell us how fast the dynamics will occur in each region, but it indicates what the *tendency* is. We have coded the fixed points according to their stability; this may be calculated using the derivative evaluated at the fixed point, $f'(x^*)$, since this derivative controls the behaviour of nearby initial conditions $x^* + \delta x$. The unstable fixed point is shown as a green square, and the stable fixed point as a grey circle; we will use this convention throughout the notebook.

Bifurcations

Now suppose that there is a **parameter** μ in the system that can be varied. For each value of μ we have a *different* ODE

$$\dot{x} = f_\mu(x).$$

For example, $\dot{x} = \mu + x^2$.

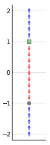
Let's draw the state space for each different value of μ:

```
g (generic function with 1 method)
```

```
g(μ, x) = μ + x^2
```

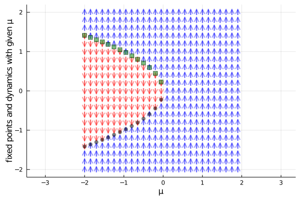
-1.0

```
@bind λ Slider(-1.0:0.05:1, show_value=true)
```



```
vector_field(x -> g(λ, x))
```

Now let's collect all the vector fields into a single plot. We *rotate* the vector field to now be vertical, thinking of the dynamics of x as occurring along the vertical direction. The horizontal axis now represents the different possible values of the parameter μ:



```
bifurcation_diagram(g)
```

We see that at the **critical value** $\mu_c = 0$ there is a **qualitative change in behaviour** in the system: for $\mu_c < 0$ there are two fixed points, whereas for $\mu_c > 0$ there are no fixed points at all. In this particular ODE the two fixed points collide in a **saddle–node** or **fold** bifurcation.

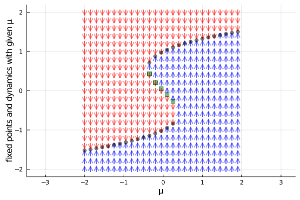
Bistability and hysteresis

Now let's look at the dynamics of the following system:

$$\dot{x} = \mu + x - x^3.$$

```
h (generic function with 1 method)
```

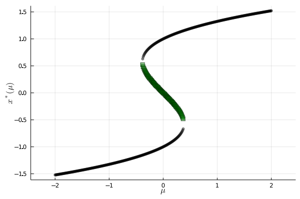
```
h(μ, x) = μ + x - x^3
```



```
bifurcation_diagram(h)
```

We see that there is a range of values of μ for which there are *three coexisting fixed points*, two stable and one unstable. Since there are two stable fixed points in which the system can remain, we say that the system is **bistable**.

Now that we understand what the plots mean and the dynamics, let's plot just the fixed points $x^*(\mu)$ as a function of μ. Such a plot is called a **bifurcation diagram**:



```
fixed_points(h)
```

The pieces of curve are called **branches**.

Hysteresis

Suppose we now think about slowly varying the parameter μ. If we change the parameter μ by a little, the system is no longer at a fixed point, since the position of the fixed point moves when μ changes. However, the system will theb **relax** by following the dynamics at the new value of μ, and will rapidly converge to the new fixed point nearby.

For example, starting at $\mu = -2$, the system will stay on the lower black (stable) **branch** until $\mu = 0.4$ or so. At that point, two fixed points collide and annihilate each other! After that there is no longer a fixed point nearby. However, there is another fixed point much further up that will now attract all trajectories, so the system rapidly transitions to that fixed point.

Now suppose we decrease the parameter again. The system will now track the *upper* branch until $\mu = -0.4$ or so, when again it will jump back down.

For each parameter value μ in the interval $[-0.4, 0.4]$ there is **bistability**, i.e. **coexistence** of *two* fixed points with the same value of μ (together with a third, unstable fixed point that is not observable).

The fact that the system tracks different stable branches depending on where we started, i.e. on the history, is known as **hysteresis**.

Hysteretic behaviour like this is found in many scientific and engineering contexts, including switches in biology, for example genetic switches, and in the historical dynamics of the earth's climate.

Slow–fast systems

What are we actually doing when we let the parameter μ vary? Effectively we now have a system with *two* equations, for example

$$\dot{x} = \mu + x - x^3;$$

$$\dot{\mu} = \epsilon,$$

where μ varies at some slow speed ϵ. On a time scale much shorter than $1/\epsilon$, the dynamics of x "does not know" that μ is changing, so it will converge to a fixed point $x^*(\mu)$ for the current value of μ. [An associated term is **adiabatic approximation**.] However, μ does gradually change, so the value of x will effectively "slide along" the curve $x(t) \simeq x^*(\mu(t))$, tracking the curve of fixed points as μ changes.

Once μ reaches a critical value μ_c, however, there is no longer a nearby fixed point, and the dynamics will rapidly transition to the far away alternative fixed point.

If we now reverse the dynamics of μ, we slide back along the upper branch.

Function library

```
euler_step (generic function with 1 method)

euler (generic function with 1 method)

derivative (generic function with 2 methods)

Main.workspace3.horiz_vector_field

Main.workspace3.vector_field!

vector_field (generic function with 1 method)

bifurcation_diagram (generic function with 1 method)

fixed_points (generic function with 1 method)
```