MATH 3610 – 02
Single population growth models
Introduction to modelling

Julien Arino
University of Manitoba
julien.arino@umanitoba.ca

Fall 2024

The University of Manitoba campuses are located on original lands of Anishinaabeg, Ininew, Anisininew, Dakota and Dene peoples, and on the National Homeland of the Red River Métis.
We respect the Treaties that were made on these territories, we acknowledge the harms and mistakes of the past, and we dedicate ourselves to move forward in partnership with Indigenous communities in a spirit of Reconciliation and collaboration.

Outline

The data – US census

Fitting a curve to the data

Least squares problems

Population growth – The logistic equation & friends

The continuous-time Markov chain logistic "equation"

The delayed logistic equation

The logistic map

Objective of this part

In this set of slides, we introduce some of the basic concepts of population growth models

We introduce some of the basic concepts of mathematical modelling and some of the questions that will be considered during the course

- **The data – US census**
- Fitting a curve to the data
- Least squares problems
- Population growth – The logistic equation & friends
- The continuous-time Markov chain logistic “equation”
- The delayed logistic equation
- The logistic map

Objective

We are given a table with the population census at different time intervals between a date *a* and a date *b*, and want to get an expression for the population. This allows us to:

- compute a value for the population at any time between the date *a* and the date *b* (**interpolation**)
- predict a value for the population at a date before *a* or after *b* (**extrapolation**)

PROCEEDINGS

OF THE

NATIONAL ACADEMY OF SCIENCES

Volume 6 JUNE 15, 1920 Number 6

ON THE RATE OF GROWTH OF THE POPULATION OF THE UNITED STATES SINCE 1790 AND ITS MATHEMATICAL REPRESENTATION[1]

By Raymond Pearl and Lowell J. Reed

Department of Biometry and Vital Statistics, Johns Hopkins University

Read before the Academy, April 26, 1920

Showing the Dates of the Taking of the Census and the Recorded Populations from 1790 to 1910

Date of Census		Recorded Population (Revised Figures from Statistical Abst., 1918)
Year	Month and Day	
1790	First Monday in August	3,929,214
1800	First Monday in August	5,308,483
1810	First Monday in August	7,239,881
1820	First Monday in August	9,638,453
1830	June 1	12,866,020
1840	June 1	17,069,453
1850	June 1	23,191,876
1860	June 1	31,443,321
1870	June 1	38,558,371
1880	June 1	50,155,783
1890	June 1	62,947,714
1900	June 1	75,994,575
1910	April 15	91,972,266

USA census from 1790 to 1910

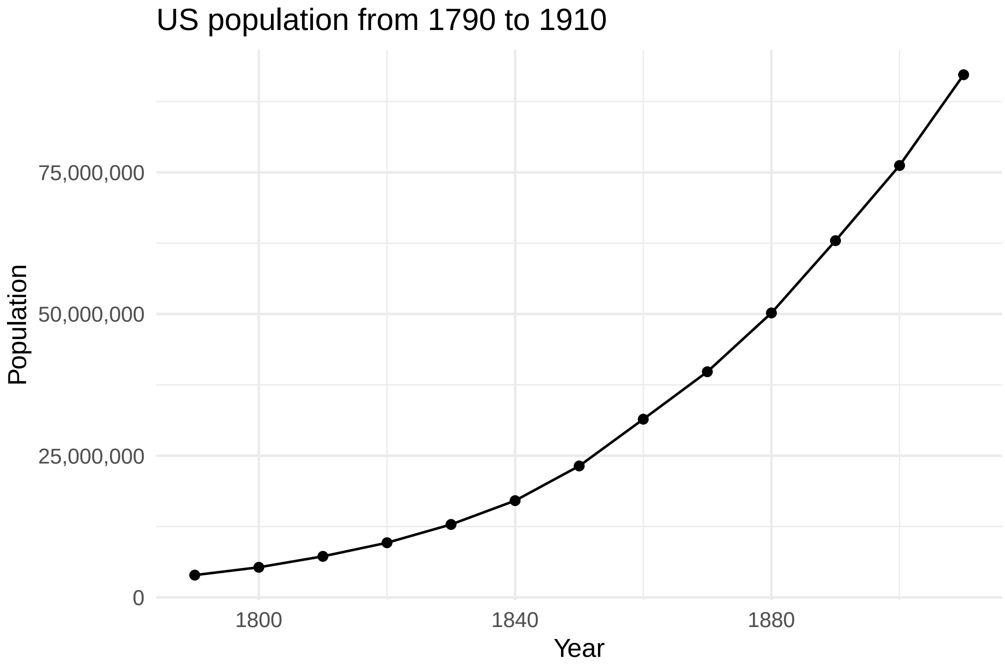
Although we have data up to 2020, we use the data up to 1910 like Pearl & Reed (note that there were some corrections to the census since the paper of Pearl & Reed)

Year	Population	Year	Population	Year	Population
1790	3,929,326	1840	17,069,458	1890	62,947,714
1800	5,308,483	1850	23,191,876	1900	76,212,168
1810	7,239,881	1860	31,443,321	1910	92,228,496
1820	9,638,453	1870	39,818,449		
1830	12,866,020	1880	50,189,209		

Plot the data !!!

It is always a good idea to plot the data before trying to do anything with it

```
plot_USA_census_to_1910 =
  ggplot(USA_census_to_1910, aes(x=Year, y=Population)) +
  geom_line() +
  geom_point() +
  labs(title="US population from 1790 to 1910",
       x="Year",
       y="Population") +
  theme_minimal()
print(plot_USA_census_to_1910)
```


US population from 1790 to 1910
Population
75,000,000
50,000,000
25,000,000
0
1800
1840
1880
Year

The data – US census

Fitting a curve to the data

Least squares problems

Population growth – The logistic equation & friends

The continuous-time Markov chain logistic “equation”

The delayed logistic equation

The logistic map

Fitting a curve to the data

First idea – This looks quadratic!

The curve looks like a piece of a parabola. So let us "fit" a curve of the form

$$P(t) = a + bt + ct^2$$

This means we want to find coefficients a, b, c such that the curve $P(t)$ is as close as possible to the data points

The data points

Year	Population	Year	Population	Year	Population
1790	3,929,326	1840	17,069,458	1890	62,947,714
1800	5,308,483	1850	23,191,876	1900	76,212,168
1810	7,239,881	1860	31,443,321	1910	92,228,496
1820	9,638,453	1870	39,818,449		
1830	12,866,020	1880	50,189,209		

We have 13 data points (t_k, P_k), $k = 1, \ldots, 13$, e.g., $(t_1, P_1) = (1790, 3929214)$, $(t_2, P_2) = (1800, 5308483)$, etc.

Some of you are familiar with this problem

If you have taken MATH 2740 (Math of Data Science), you have seen this before!

See the notes on the course website for a refresher on this problem here and the corresponding videos here, here, here, here and here

(Sorry about the number of videos, I need to reorganize them!)

To do this, we want to minimize

$$S = \sum_{k=1}^{13} (P(t_k) - P_k)^2$$

where t_k are the known dates, P_k are the known populations, and $P(t_k) = a + bt_k + ct_k^2$

The t_k and P_k are known, a, b, c are to be found, so we write S as a function of a, b, c: $S(a, b, c)$

Recall your multivariable calculus:

$$S = S(a, b, c) = \sum_{k=1}^{13} \left(a + bt_k + ct_k^2 - P_k\right)^2$$

is maximal if (necessary condition) $\partial S/\partial a = \partial S/\partial b = \partial S/\partial c = 0$

We have

$$\frac{\partial S}{\partial a} = 2\sum_{k=1}^{13}(a + bt_k + ct_k^2 - P_k)$$

$$\frac{\partial S}{\partial b} = 2\sum_{k=1}^{13}(a + bt_k + ct_k^2 - P_k)t_k$$

$$\frac{\partial S}{\partial c} = 2\sum_{k=1}^{13}(a + bt_k + ct_k^2 - P_k)t_k^2$$

Thus, we want

$$\frac{\partial S}{\partial a} = 0 \iff 2\sum_{k=1}^{13}(a + bt_k + ct_k^2 - P_k) = 0$$
$$\frac{\partial S}{\partial b} = 0 \iff 2\sum_{k=1}^{13}(a + bt_k + ct_k^2 - P_k)t_k = 0$$
$$\frac{\partial S}{\partial c} = 0 \iff 2\sum_{k=1}^{13}(a + bt_k + ct_k^2 - P_k)t_k^2 = 0$$

that is

$$\sum_{k=1}^{13}(a + bt_k + ct_k^2 - P_k) = 0$$
$$\sum_{k=1}^{13}(a + bt_k + ct_k^2 - P_k)t_k = 0$$
$$\sum_{k=1}^{13}(a + bt_k + ct_k^2 - P_k)t_k^2 = 0$$

Rearranging the system

$$\sum_{k=1}^{13}(a + bt_k + ct_k^2 - P_k) = 0$$
$$\sum_{k=1}^{13}(a + bt_k + ct_k^2 - P_k)t_k = 0$$
$$\sum_{k=1}^{13}(a + bt_k + ct_k^2 - P_k)t_k^2 = 0$$

we get

$$\sum_{k=1}^{13}(a + bt_k + ct_k^2) = \sum_{k=1}^{13}P_k$$
$$\sum_{k=1}^{13}(at_k + bt_k^2 + ct_k^3) = \sum_{k=1}^{13}P_k t_k$$
$$\sum_{k=1}^{13}(at_k^2 + bt_k^3 + ct_k^4) = \sum_{k=1}^{13}P_k t_k^2$$

$$\sum_{k=1}^{13}(a + bt_k + ct_k^2) = \sum_{k=1}^{13} P_k$$
$$\sum_{k=1}^{13}(at_k + bt_k^2 + ct_k^3) = \sum_{k=1}^{13} P_k t_k$$
$$\sum_{k=1}^{13}(at_k^2 + bt_k^3 + ct_k^4) = \sum_{k=1}^{13} P_k t_k^2$$

after a bit of tidying up, takes the form

$$\left(\sum_{k=1}^{13} 1\right) a + \left(\sum_{k=1}^{13} t_k\right) b + \left(\sum_{k=1}^{13} t_k^2\right) c = \sum_{k=1}^{13} P_k$$
$$\left(\sum_{k=1}^{13} t_k\right) a + \left(\sum_{k=1}^{13} t_k^2\right) b + \left(\sum_{k=1}^{13} t_k^3\right) c = \sum_{k=1}^{13} P_k t_k$$
$$\left(\sum_{k=1}^{13} t_k^2\right) a + \left(\sum_{k=1}^{13} t_k^3\right) b + \left(\sum_{k=1}^{13} t_k^4\right) c = \sum_{k=1}^{13} P_k t_k^2$$

So the aim is to solve the linear system

$$\begin{pmatrix} 13 & \sum_{k=1}^{13} t_k & \sum_{k=1}^{13} t_k^2 \\ \sum_{k=1}^{13} t_k & \sum_{k=1}^{13} t_k^2 & \sum_{k=1}^{13} t_k^3 \\ \sum_{k=1}^{13} t_k^2 & \sum_{k=1}^{13} t_k^3 & \sum_{k=1}^{13} t_k^4 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} \sum_{k=1}^{13} P_k \\ \sum_{k=1}^{13} P_k t_k \\ \sum_{k=1}^{13} P_k t_k^2 \end{pmatrix}$$

With R, this is easy to solve..?

```
> t = as.double(USA_census_to_1910$Year)
> pop = as.double(USA_census_to_1910$Population)
> A = matrix(c(13, sum(t), sum(t^2),
+              sum(t), sum(t^2), sum(t^3),
+              sum(t^2), sum(t^3), sum(t^4)),
+            nrow=3,byrow=TRUE)
> b = c(sum(pop),
+       sum(pop * t),
+       sum(pop * t^2))
> sol = try(solve(A,b))
> writeLines(sol)
```

```
Error in solve.default(A, b) :
  system is computationally singular: reciprocal condition number = 1.11839e-20
```

So we need to do some "time shifting": the problem is that some of the entries are too large

```
> t = t - 1790
> A = matrix(c(13, sum(t), sum(t^2),
+              sum(t), sum(t^2), sum(t^3),
+              sum(t^2), sum(t^3), sum(t^4)),
+            nrow=3,byrow=TRUE)
> b = c(sum(pop),
+       sum(pop * t),
+       sum(pop * t^2))
> sol = try(solve(A,b))
> print(sol)
[1] 5544964.000 -109242.513    6849.346
```

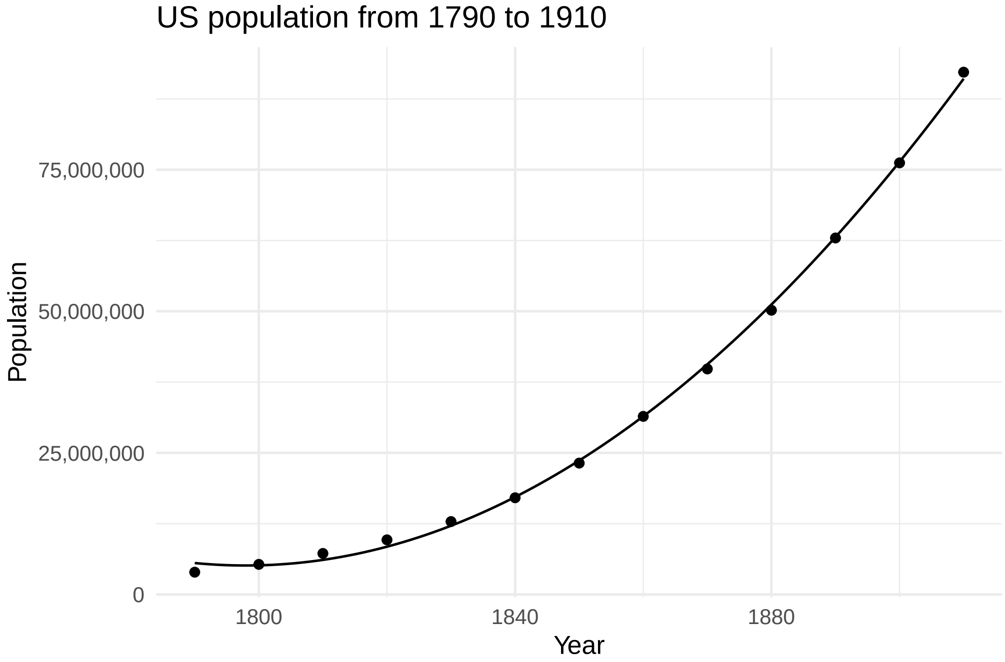
Thus

$$P(t) = 5544964 - 109243t + 6849t^2$$

(keeping in mind that time is here shifted and starts at 0)

So we define the function

```
> sol_plot = function(t, sol) {
+   t = t - 1790
+   return(sol[1] + sol[2]*t + sol[3]*t^2)
+ }
```


US population from 1790 to 1910
Population
75,000,000
50,000,000
25,000,000
0
1800
1840
1880
Year

Form the vector of errors, and compute sum of errors squared:

```
> t = USA_census_to_1910$Year
> P = USA_census_to_1910$Population
> E = sum((P - sol_plot(t, sol))^2)
```

Quite a large error (9,256,979,482,173), which is normal since we have used actual numbers, not thousands or millions of individuals, and we are taking the square of the error

Now for the big question...

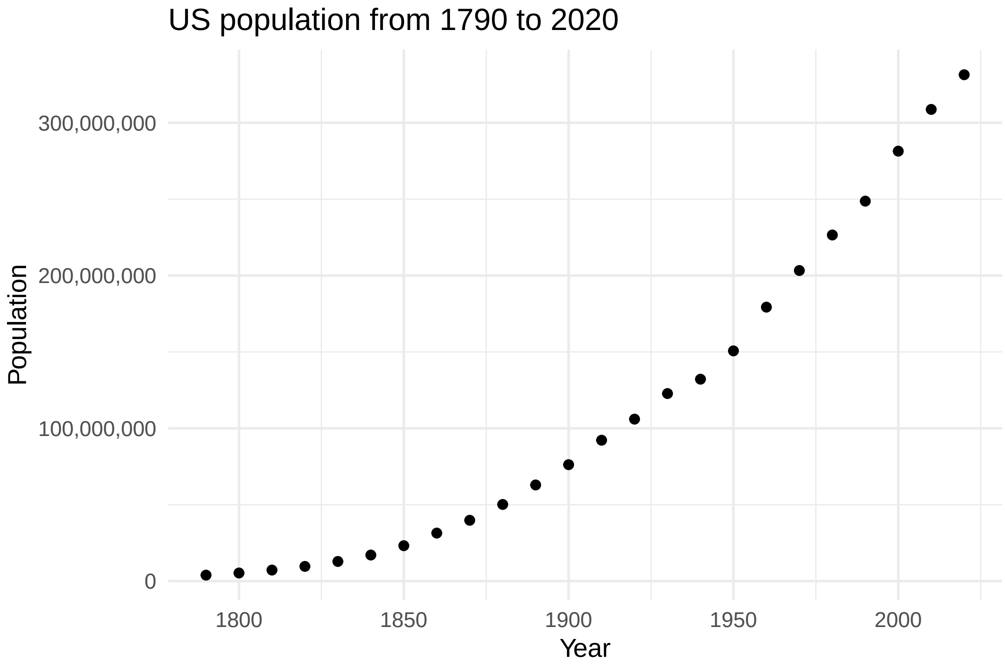
How does our formula do for present times?

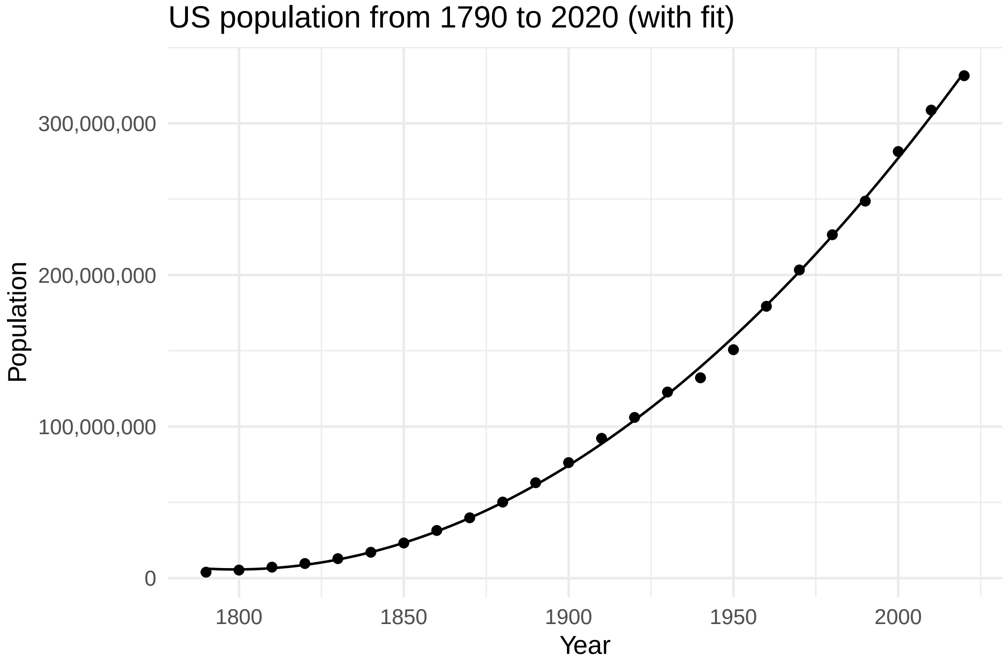
```
> format(sol_plot(2024, sol), big.mark = ",")
[1] "355,024,999"
```

Actually, quite well: 355,024,999, compared to the 345,786,196 September 2024 estimate, overestimates the population by 9,238,803, a relative error of 2.67%

The US population from 1790 to 2020 (revised numbers)

Year	Population	Year	Population	Year	Population
1790	3,929,326	1890	62,947,714	1990	248,709,873
1800	5,308,483	1900	76,212,168	2000	281,421,906
1810	7,239,881	1910	92,228,496	2010	308,745,538
1820	9,638,453	1920	106,021,537	2020	331,449,281
1830	12,866,020	1930	122,775,046		
1840	17,069,458	1940	132,164,569		
1850	23,191,876	1950	150,697,361		
1860	31,443,321	1960	179,323,175		
1870	39,818,449	1970	203,302,031		
1880	50,189,209	1980	226,545,805		


US population from 1790 to 2020
300,000,000
200,000,000
100,000,000
0
Population
1800
1850
1900
1950
2000
Year


US population from 1790 to 2020 (with fit)
300,000,000
200,000,000
100,000,000
0
Population
1800
1850
1900
1950
2000
Year

How does our formula do for present times?

```
> format(sol_plot(2024, sol_full), big.mark = ",")
[1] "345,749,152"
```

Actually, quite well: 345,749,152, compared to the 345,786,196 September 2024 estimate, underestimates the population by -37,043.51, a relative error of -0.01%

Fitting a curve to the data

Fitting a quadratic curve to the data

Some similar curves

Population curves – Gompertz

Other similar approaches

Pritchett, 1891:

$$P = a + bt + ct^2 + dt^3$$

(we have done this one, and found it to be quite good too)

Pearl, 1907:

$$P(t) = a + bt + ct^2 + d\ln t$$

Finds

$$P(t) = 9,064,900 - 6,281,430t + 842,377t^2 + 19,829,500\ln t.$$

Showing (*a*) the Actual Population[1] on Census Dates, (*b*) Estimated Population from Pritchett's Third-Order Parabola, (*c*) Estimated Population from Logarithmic Parabola, and (*d*) (*e*) Root-Mean Square Errors of Both Methods

Census Year	(*a*) Observed Population	(*b*) Pritchett Estimate	(*c*) Logarithmic Parabola Estimate	(*d*) Error of (*b*)	(*e*) Error of (*c*)
1790	3,929,000	4,012,000	3,693,000	+ 83,000	− 236,000
1800	5,308,000	5,267,000	5,865,000	− 41,000	+ 557,000
1810	7,240,000	7,059,000	7,293,000	− 181,000	+ 53,000
1820	9,638,000	9,571,000	9,404,000	− 67,000	− 234,000
1830	12,866,000	12,985,000	12,577,000	+ 119,000	− 289,000
1840	17,069,000	17,484,000	17,132,000	+ 415,000	+ 63,000
1850	23,192,000	23,250,000	23,129,000	+ 58,000	− 63,000
1860	31,443,000	30,465,000	30,633,000	− 978,000	− 810,000
1870	38,558,000	39,313,000	39,687,000	+ 755,000	+1,129,000
1880	50,156,000	49,975,000	50,318,000	− 181,000	+ 162,000
1890	62,948,000	62,634,000	62,547,000	− 314,000	− 401,000
1900	75,995,000	77,472,000	76,389,000	+1,477,000	+ 394,000
1910	91,972,000	94,673,000	91,647,000	+2,701,000	− 325,000
				935,000[2]	472,000[2]
1920		114,416,000	108,214,000		

[1] To the nearest thousand.

[2] Root-mean square error.

The logistic curve

Pearl and Reed try

$$P(t) = \frac{be^{at}}{1 + ce^{at}}$$

or

$$P(t) = \frac{b}{e^{-at} + c}$$

What is wrong with the logistic equation here?

- The carrying capacity is constant
- The model does not take immigration into account (for the US, this is an important component)

PROCEEDINGS

OF THE

NATIONAL ACADEMY OF SCIENCES

Volume 18 January 15, 1932 Number 1

THE GOMPERTZ CURVE AS A GROWTH CURVE

By Charles P. Winsor

Department of Biology, School of Hygiene and Public Health, Johns Hopkins University

Communicated December 2, 1931

PROPERTY	GOMPERTZ	LOGISTIC
Equation	$y = ke^{-e^{a-bx}}$	$y = \dfrac{k}{1 + e^{a-bx}}$
Number of constants	3	3
Asymptotes	$\begin{cases} y = 0 \\ y = k \end{cases}$	$\begin{cases} y = 0 \\ y = k \end{cases}$
Inflection	$\begin{cases} x = \dfrac{a}{b} \\ y = \dfrac{k}{e} \end{cases}$	$\begin{cases} x = \dfrac{a}{b} \\ y = \dfrac{k}{2} \end{cases}$
Straight line form of equation	$\log \log \dfrac{k}{y} = a - bx$	$\log \dfrac{k - y}{y} = a - bx$
Symmetry	Assymetrical	Symmetrical about inflection
Growth rate	$\dfrac{dy}{dx} = bye^{a-bx} = by \log \dfrac{k}{y}$	$\dfrac{dy}{dx} = \dfrac{b}{k} y(k - y)$
Maximum growth rate	$\dfrac{bk}{e}$	$\dfrac{bk}{4}$
Relative growth rate as function of time	$\dfrac{1}{y} \dfrac{dy}{dx} = be^{a-bx}$	$\dfrac{1}{y} \dfrac{dy}{dx} = \dfrac{b}{1 + e^{-a+bx}}$
Relative growth rate as function of size	$\dfrac{1}{y} \dfrac{dy}{dx} = b (\log k - \log y)$	$\dfrac{1}{y} \dfrac{dy}{dx} = \dfrac{b}{k} (k - y)$

The data – US census

Fitting a curve to the data

Least squares problems

Population growth – The logistic equation & friends

The continuous-time Markov chain logistic “equation”

The delayed logistic equation

The logistic map

A.k.a. if the Math Dept was less #$%&, you'd know this

The following are a brief extract from MATH 2740 slides...

The least squares problem (simplest version)

Definition 1

Given a collection of points $(x_1, y_1), \ldots, (x_n, y_n)$, find the coefficients a, b of the line $y = a + bx$ such that

$$\|\mathrm{e}\| = \sqrt{\varepsilon_1^2 + \cdots + \varepsilon_n^2} = \sqrt{(y_1 - \tilde{y}_1)^2 + \cdots + (y_n - \tilde{y}_n)^2}$$

is minimal, where $\tilde{y}_i = a + bx_i$ for $i = 1, \ldots, n$

We just saw how to solve this by brute force using a genetic algorith to minimise $\|e\|$, let us now see how to solve this problem "properly"

For a data point $i = 1, \ldots, n$

$$\varepsilon_i = y_i - \tilde{y}_i = y_i - (a + bx_i)$$

So if we write this for all data points,

$$\begin{aligned} \varepsilon_1 &= y_1 - (a + bx_1) \\ &\vdots \\ \varepsilon_n &= y_n - (a + bx_n) \end{aligned}$$

In matrix form

$$\boldsymbol{e} = \boldsymbol{b} - A\boldsymbol{x}$$

with

$$\boldsymbol{e} = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix}, A = \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}, \boldsymbol{x} = \begin{pmatrix} a \\ b \end{pmatrix} \text{ and } \boldsymbol{b} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

The least squares problem (reformulated)

Definition 2 (Least squares solutions)

Consider a collection of points $(x_1, y_1), \dots, (x_n, y_n)$, a matrix $A \in \mathcal{M}_{mn}$, $\boldsymbol{b} \in \mathbb{R}^m$. A **least squares solution** of $A\boldsymbol{x} = \boldsymbol{b}$ is a vector $\tilde{\boldsymbol{x}} \in \mathbb{R}^n$ s.t.

$$\forall \boldsymbol{x} \in \mathbb{R}^n, \quad \|\boldsymbol{b} - A\tilde{\boldsymbol{x}}\| \leq \|\boldsymbol{b} - A\boldsymbol{x}\|$$

Needed to solve the problem

Definition 3 (Best approximation)

Let V be a vector space, $W \subset V$ and $\mathrm{v} \in V$. The **best approximation** to v in W is $\tilde{\mathrm{v}} \in W$ s.t.

$$\forall \mathrm{w} \in W, \mathrm{w} \neq \tilde{\mathrm{v}}, \quad \|\mathrm{v} - \tilde{\mathrm{v}}\| < \|\mathrm{v} - \mathrm{w}\|$$

Theorem 4 (Best approximation theorem)

Let V be a vector space with an inner product, $W \subset V$ and $\mathrm{v} \in V$. Then $\mathrm{proj}_W(\mathrm{v})$ is the best approximation to v in W

Let us find the least squares solution

$\forall \boldsymbol{x} \mathbb{R}^n$, $A\boldsymbol{x}$ is a vector in the **column space** of A (the space spanned by the vectors making up the columns of A)

Since $\boldsymbol{x} \in \mathbb{R}^n$, $A\boldsymbol{x} \in \text{col}(A)$

$\implies$ least squares solution of $A\boldsymbol{x} = \boldsymbol{b}$ is a vector $\tilde{\boldsymbol{y}} \in \text{col}(A)$ s.t.

$$\forall \boldsymbol{y} \in \text{col}(A), \quad \|\boldsymbol{b} - \tilde{\boldsymbol{y}}\| \leq \|\boldsymbol{b} - \boldsymbol{y}\|$$

This looks very much like Best approximation and Best approximation theorem

Putting things together

We just stated: The least squares solution of $A\boldsymbol{x} = \boldsymbol{b}$ is a vector $\tilde{\boldsymbol{y}} \in \text{col}(A)$ s.t.

$$\forall \boldsymbol{y} \in \text{col}(A), \quad \|\boldsymbol{b} - \tilde{\boldsymbol{y}}\| \leq \|\boldsymbol{b} - \boldsymbol{y}\|$$

We know (reformulating a tad):

Theorem 5 (Best approximation theorem)

Let V be a vector space with an inner product, $W \subset V$ and $\mathrm{v} \in V$. Then $\text{proj}_W(\mathrm{v}) \in W$ is the best approximation to v in W, i.e.,

$$\forall \mathrm{w} \in W, \mathrm{w} \neq \text{proj}_W(\mathrm{v}), \quad \|\mathrm{v} - \text{proj}_W(\mathrm{v})\| < \|\mathrm{v} - \mathrm{w}\|$$

$\implies W = \text{col}(A)$, $\boldsymbol{v} = \boldsymbol{b}$ and $\tilde{\boldsymbol{y}} = \text{proj}_{\text{col}(A)}(\mathrm{b})$

So if $\tilde{\boldsymbol{x}}$ is a least squares solution of $A\boldsymbol{x} = \boldsymbol{b}$, then

$$\tilde{\boldsymbol{y}} = A\tilde{\boldsymbol{x}} = \text{proj}_{\text{col}(A)}(\text{b})$$

We have

$$\boldsymbol{b} - A\tilde{\boldsymbol{x}} = \boldsymbol{b} - \text{proj}_{\text{col}(A)}(\text{b}) = \text{perp}_{\text{col}(A)}(\text{b})$$

and it is easy to show that

$$\text{perp}_{\text{col}(A)}(\text{b}) \perp \text{col}(A)$$

So for all columns $\boldsymbol{a}_i$ of A

$$\boldsymbol{a}_i \cdot (\boldsymbol{b} - A\tilde{\boldsymbol{x}}) = 0$$

which we can also write as $\boldsymbol{a}_i^T(\boldsymbol{b} - A\tilde{\boldsymbol{x}}) = 0$

For all columns $\boldsymbol{a}_i$ of A,

$$\boldsymbol{a}_i^T(\boldsymbol{b} - A\tilde{\boldsymbol{x}}) = 0$$

This is equivalent to saying that

$$A^T(\boldsymbol{b} - A\tilde{\boldsymbol{x}}) = 0$$

We have

$$\begin{aligned} A^T(\boldsymbol{b} - A\tilde{\boldsymbol{x}}) = 0 &\iff A^T\boldsymbol{b} - A^TA\tilde{\boldsymbol{x}} = 0 \\ &\iff A^T\boldsymbol{b} = A^TA\tilde{\boldsymbol{x}} \\ &\iff A^TA\tilde{\boldsymbol{x}} = A^T\boldsymbol{b} \end{aligned}$$

The latter system constitutes the normal equations for $\tilde{\boldsymbol{x}}$

Least squares theorem

Theorem 6 (Least squares theorem)

$A \in \mathcal{M}_{mn}$, $\boldsymbol{b} \in \mathbb{R}^m$. Then

1. *$A\boldsymbol{x} = \boldsymbol{b}$ always has at least one least squares solution $\tilde{\boldsymbol{x}}$*
2. *$\tilde{\boldsymbol{x}}$ least squares solution to $A\boldsymbol{x} = \boldsymbol{b} \iff \tilde{\boldsymbol{x}}$ is a solution to the normal equations $A^T A\tilde{\boldsymbol{x}} = A^T\boldsymbol{b}$*
3. *A has linearly independent columns $\iff A^T A$ invertible.*
   In this case, the least squares solution is unique and

$$\tilde{\boldsymbol{x}} = \left(A^T A\right)^{-1} A^T\boldsymbol{b}$$

We have seen 1 and 2, we will not show 3 (it is not hard)

Suppose we want to fit something a bit more complicated..

For instance, instead of the affine function

$$y = a + bx$$

suppose we want to do the quadratic

$$y = a_0 + a_1 x + a_2 x^2$$

or even

$$y = k_0 e^{k_1 x}$$

How do we proceed?

Fitting the quadratic

We have the data points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ and want to fit

$$y = a_0 + a_1 x + a_2 x^2$$

At (x_1, y_1),

$$\tilde{y}_1 = a_0 + a_1 x_1 + a_2 x_1^2$$

$\vdots$

At (x_n, y_n),

$$\tilde{y}_n = a_0 + a_1 x_n + a_2 x_n^2$$

In terms of the error

$$\varepsilon_1 = y_1 - \tilde{y}_1 = y_1 - (a_0 + a_1 x_1 + a_2 x_1^2)$$
$$\vdots$$
$$\varepsilon_n = y_n - \tilde{y}_n = y_n - (a_0 + a_1 x_n + a_2 x_n^2)$$

i.e.,

$$\boldsymbol{e} = \boldsymbol{b} - A\boldsymbol{x}$$

where

$$\boldsymbol{e} = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix}, A = \begin{pmatrix} 1 & x_1 & x_1^2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 \end{pmatrix}, \boldsymbol{x} = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} \text{ and } \boldsymbol{b} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

Theorem 6 applies, with here $A \in \mathcal{M}_{n3}$ and $\boldsymbol{b} \in \mathbb{R}^n$

Fitting the exponential

Things are a bit more complicated here

If we proceed as before, we get the system

$$y_1 = k_0 e^{k_1 x_1}$$
$$\vdots$$
$$y_n = k_0 e^{k_1 x_n}$$

$e^{k_1 x_i}$ is a nonlinear term, it cannot be put in a matrix

However: take the ln of both sides of the equation

$$\ln(y_i) = \ln(k_0 e^{k_1 x_i}) = \ln(k_0) + \ln(e^{k_1 x_i}) = \ln(k_0) + k_1 x_i$$

If $y_i, k_0 > 0$, then their ln are defined and we're in business..

$$\ln(y_i) = \ln(k_0) + k_1 x_i$$

So the system is

$$\boldsymbol{y} = A\boldsymbol{x} + \boldsymbol{b}$$

with

$$A = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \boldsymbol{x} = \begin{pmatrix} k_1 \end{pmatrix}, \boldsymbol{b} = \begin{pmatrix} \ln(k_0) \end{pmatrix} \text{ and } \boldsymbol{y} = \begin{pmatrix} \ln(y_1) \\ \vdots \\ \ln(y_n) \end{pmatrix}$$

Population growth – The logistic equation & friends

- Ordinary differential equations 101
- Formulating the logistic equation
- From birth-death to the logistic
- Qualitative analysis of the logistic equation
- Including an Allee effect
- Gompertz growth model
- Tumor growth
- Harvest model
- Computational analysis of the logistic equation

Population growth – The logistic equation & friends

Ordinary differential equations

Let t be an independent variable (often time) and $x(t) \in \mathbb{R}^n$ a dependent variable

An ordinary differential equation (ODE) is an equation of the form

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

where $f : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$ is a given function that describes how $x(t)$ changes with t

Some equations involve higher order derivatives, but we will not consider any here so we do not present them

Initial value problems

An **initial value problem** (IVP) is an ODE with an **initial condition** (IC)

$$\frac{d}{dt}x(t) = f(t, x(t)) \tag{1a}$$

$$x(t_0) = x_0 \tag{1b}$$

where $t_0 \in \mathbb{R}$ is the initial time and $x_0 \in \mathbb{R}^n$ is the initial value

In practice, solutions to (1a) form families of curves and the initial condition (1b) allows to pick one of these curves

Classification of ODEs

If $f(t, x(t)) = f(x(t))$, i.e., the function f does not depend on t, the ODE is **autonomous**

Most of what we consider here will be autonomous, so assume that for now

If $f(x(t))$ is linear in $x(t)$, the ODE is **linear**

Simplifying notation

We often drop the time dependence of x on t and write $'$ for d/dt. Further taking into account that f is autonomous, we write (1) as

$$x' = f(x) \tag{2a}$$

$$x(t_0) = x_0 \tag{2b}$$

Existence and uniqueness of solutions

There is a theory that studies the existence and uniqueness of solutions to (2); see, e.g., MATH 3440 (Ordinary Differential Equations)

Here, we take a shortcut and use one of the easiest forms

Theorem 7 (Existence and uniqueness of solutions to IVPs)

If f is C^1 (has continuous first-order partials) in x, then there exists a unique solution to (2) in some interval $[t_0 - \varepsilon, t_0 + \varepsilon]$ for some $\varepsilon > 0$

(Also works for nonautonomous equations)

Studying ODE/IVP

In short, there are three non-mutually exclusive ways to tackle an IVP such as (1) or (2)

1. Analytically: find a formula for $x(t)$

2. Numerically: approximate $x(t)$

3. Qualitatively: study the behaviour of $x(t)$ without finding $x(t)$

Don't expect to be able to do 1 more than a few times in this course (or anywhere for that matter!); so we mention it but focus on 2 and 3

Solutions to scalar autonomous ODE are monotone

Theorem 8

Consider an IVP of the form (2) where $f : \mathbb{R} \to \mathbb{R}$, i.e., where the ODE is scalar autonomous.

Then solutions to (2) are monotone, in the sense that if we denote $\phi_1(t)$ and $\phi_2(t)$ the solutions through $(t_0, x(t_0) = x_1)$ and $(t_0, x(t_0) = x_2)$, respectively, with $x_1 \leq x_2$, then

$$\phi_1(t) \leq \phi_2(t), \quad t \geq t_0$$

This is useful in the present slide set, since the ODEs we consider are scalar autonomous

Population growth – The logistic equation & friends

Our goal: the logistic ODE

The logistic *curve* mentioned by Pearl and Reed is the solution to the **logistic** ODE

$$\frac{d}{dt}N(t) = rN(t)\left(1 - \frac{N(t)}{K}\right)$$

often denoted without the time dependence and using $'$ for d/dt

$$N' = rN\left(1 - \frac{N}{K}\right) \tag{3}$$

r is the **intrinsic growth rate**, K is the **carrying capacity**

This equation was introduced by Pierre-François Verhulst (1804-1849) in 1844

Deriving the logistic equation

Our aim here is to derive (3) from first principles

This illustrates the mathematical modelling process when using a differential equations approach

The birth-death process

The simplest model for population growth is the **birth-death** process

Let $N(t)$ (N for short if unambiguous) be the total number of individuals in an isolated population at time t

Neglecting all other sources of change, N is

- positively influenced (increases) because of births into the population
- negatively influenced (decreases) because of deaths in the population

Where ODEs come in

We said N increases/decreases because of births/deaths...

This sounds a lot like what derivatives do. Remember first-year calculus: if $f(x)$ is a function $\mathbb{R} \to \mathbb{R}$, then f is increasing at x if $f'(x) > 0$ and decreasing at x if $f'(x) < 0$

Returning to $N(t)$, we thus should have $N'(t) > 0$ because of births and $N'(t) < 0$ because of deaths

Intuitively, if births happen more often than deaths, the population should increase, while if deaths are more frequent than births, it should decrease

The birth-death equation

Assume that

- birth happens at the **per capita** rate b
- death happens at the *per capita* rate d

Rates constant *per capita* versus "just" constant

Birth increases the population, so

$$N' = b(N)$$

with $b(N) : \mathbb{R}_+ \to \mathbb{R}_+$ the birth rate function

Two easiest choices, for $b \in \mathbb{R}_+ \setminus \{0\}$:

1. $b(N) = b$, a constant
2. $b(N) = bN$, a linear function of N (a *per capita* rate, for which N'/N is constant)

An example (2022-06-01 to 2023-06-01)

Suppose time units are years. Consider data from regional health authorities and assume that the number of kids under 1 year old is a good proxy for the number of births

	Portage la Prairie	Winnipeg
Population	16,389	832,214
Births (raw number)	176	7,574
Births (*per capita*)	0.0107	0.0091

If $b(N) = b$, we should use $b = 176$ for PIP and $b = 7,574$ for Winnipeg. On the other hand, if $b(N) = bN$, we should use $b = 0.0107$ for PIP and $b = 0.0091$ for Winnipeg.. so could get by with $b = 0.01$ for both

Per capita is typically more realistic

It is more realistic to assume that the birth rate is proportional to the population size, because that means that the parameter b is a property of the species more so than of where they are located

Note that there are important differences, though: birth rates are generally consistent across a country (and even that varies), but differ from country to country

We will see later in the course that using a constant birth rate can be feasible as well, once we understand its role on the model behaviour

Modelling assumptions

Modelling is all about making assumptions and understanding the consequences of these assumptions

Use the *per capita* rate bN for births and the *per capita* rate dN for deaths

This gives the birth-death model

$$N' = bN - dN$$

This model must be "equiped" with an initial condition, e.g. $N(t_0) = N_0 \geq 0$

The birth-death model

All things considered, the model for the population with only birth and death is therefore

$$N' = bN - dN \tag{4a}$$

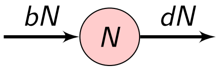
$$N(t_0) = N_0 \tag{4b}$$

Compartmental models and flow diagrams

The birth-death equation is a **compartmental model** using a single compartment $N(t)$

A **compartment** is defined as a kinetically homogeneous group of individuals

A **flow diagram** is a graphical representation of a compartmental model



What can we say about model (4)?

(4a) is a scalar first-order linear autonomous ODE $\Rightarrow$ we easily find the explicit solution

$$N(t) = N(t_0)e^{(b-d)(t-t_0)}, \qquad t \geq t_0 \tag{5}$$

To check that (5) is a solution to (4), we differentiate it and check it satisfies (4a); we also check it satisfies (4b):

$$\begin{aligned}
N'(t) &= \frac{d}{dt}\left(N(t_0)e^{(b-d)(t-t_0)}\right) \\
&= N(t_0)\frac{d}{dt}\left(e^{(b-d)(t-t_0)}\right) \\
&= N(t_0)(b-d)e^{(b-d)(t-t_0)} \\
&= (b-d)N(t_0)e^{(b-d)(t-t_0)} \\
&= (b-d)\left(N(t_0)e^{(b-d)(t-t_0)}\right) \\
&= (b-d)N(t) \\
N(t_0) &= N(t_0)e^{(b-d)(t_0-t_0)} = N(t_0)e^0 = N_0
\end{aligned}$$

So the behaviour is easy to understand

$$N(t) = N(t_0)e^{(b-d)(t-t_0)}, \qquad t \geq t_0 \tag{5}$$

For $t > t_0$, $t - t_0 > 0$, so the behaviour of (5) is determined by the sign of $b - d$

- If $b > d$, then $N(t)$ grows exponentially
- If $b = d$, then $N(t)$ remains constant (and equal to $N(t_0)$)
- If $b < d$, then $N(t)$ decays exponentially to 0

This analysis is possible because we have the explicit solution (5). That is almost never the case, so what can we do?

Qualitative analysis

As remarked earlier, (4a) is a scalar first-order linear autonomous ODE, so the qualitative analysis is also super easy, but the ideas that follow are general and are used later in more complex models

- Check that the model "makes sense" (we sometimes say that the model is *well-posed*)
- Seek an **equilibrium solution** (an **equilibrium** for short), i.e., a constant solution $N(t) = N^*$ such that $N' = 0$ when $N(t) = N^*$
- Consider the sign of N' for $N < N^*$ and $N > N^*$ to determine the *stability* of the equilibrium solution

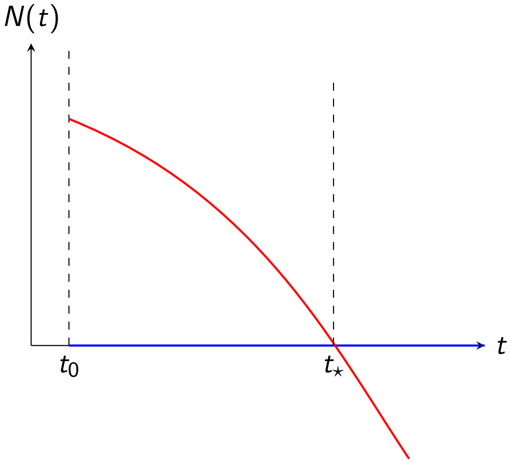
First, a remark – The model makes sense

The model (4) makes sense only if its solutions exist uniquely and $N(t) \geq 0$ for all $t \geq t_0$

The existence and uniqueness of solutions is guaranteed by Theorem 7, since the function $f(N) = bN - dN$ is C^1 (it is actually C^∞)

Now remark that if $N(t_0) = 0$, then $N(t) = 0$ for all $t \geq t_0$ because $N' = bN - dN = 0$ for $N = 0$

Now suppose that $N(t_0) > 0$. Then there does not exist $t_\star > t_0$ such that $N(t_\star) = 0$. If there were, then at $t_\star$ we would have two solutions such that $N(t_\star) = 0$: the one that is identically 0 and the one that starts with $N(t_0) > 0$, contradicting uniqueness of solutions


$N(t)$
t
t_0
$t_\star$

Equilibrium

An equilibrium solution N^* satisfies $N^{*\prime} = 0$, i.e.,

$$bN^* - dN^* = 0$$

This gives $N^* = 0$ or $b = d$

If $b = d$, then (4a) reduces to $N' = 0$, which has the general solution $N(t) = C$ (C the integration constant), so, with the initial condition (4b), we have $N(t) = N_0$ for all $t \geq t_0$

So let us now consider the case $b \neq d$ and $N^* = 0$

Stability of the equilibrium $N^* = 0$

Write the ODE (4a) as

$$N' = f(N)$$

to emphasise that $f(N) = bN - dN$ is the right-hand side of the ODE

We have seen that if we start with $N(t_0) = 0$, then $N(t) = 0$ for all $t \geq t_0$ so assume $N(t_0) > 0$. This means that $N(t) > 0$ for all $t \geq t_0$

As a consequence, the sign of $f(N)$ is the same as the sign of $b - d$

- If $b > d$, then $f(N) > 0$
- If $b = d$, then $f(N) = 0$
- If $b < d$, then $f(N) < 0$

The situation

- If $b > d$, then $f(N) > 0 \Rightarrow N(t)$ is increasing for all $t \geq t_0 \Rightarrow N(t) \to \infty$ as $t \to \infty$
- If $b = d$, then $f(N) = 0 \Rightarrow N(t) = N_0$ for all $t \geq t_0$
- If $b < d$, then $f(N) < 0 \Rightarrow N(t)$ is decreasing for all $t \geq t_0 \Rightarrow N(t) \to 0$ as $t \to \infty$, since we have also seen that $N(t) > 0$ for all $t \geq t_0$ when $N(t_0) > 0$

Population growth – The logistic equation & friends

Deriving the logistic equation

The birth-death model does not account for the fact that (competitive) interactions between individuals reduces their ability to survive, potentially resulting in death

For instance, resources are limited

This gives

$$N' = bN - dN - \text{competition}$$

Accounting for competition

Competition describes the mortality that occurs when two individuals meet

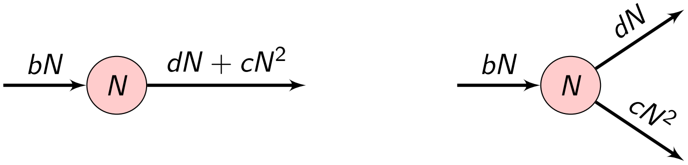
- In chemistry, if there is a concentration X of one product and Y of another product, then XY, called **mass action**, describes the number of interactions of molecules of the two products
- Here, we assume that X and Y are of the same type (individuals). So there are N^2 contacts
- These N^2 contacts lead to death of one of the individuals at the rate c

Therefore, the **logistic** equation is

$$N' = bN - dN - cN^2 \tag{6}$$

Flow diagram(s) of the logistic equation

$$N' = bN - dN - cN^2 \tag{6}$$



(Left is more compact. Right emphasises that there are two different ways to leave the compartment)

Reinterpreting the logistic equation

The equation

$$N' = bN - dN - cN^2 \tag{6}$$

is rewritten as

$$N' = (b - d)N - cN^2$$

- $b - d$ represents the rate at which the population increases (or decreases) in the absence of competition. It is called the **intrinsic growth rate** of the population
- c is the rate of **intraspecific** competition. The prefix **intra** refers to the fact that the competition is occurring between members of the same species, that is, within the species

Yet aother interpretation of the logistic equation

We have

$$N' = (b - d)N - cN^2$$

Factor out an N:

$$N' = \big((b - d) - cN\big)N$$

This gives us another interpretation of the logistic equation. Writing

$$\frac{N'}{N} = (b - d) - cN$$

we have the *per capita* growth rate N'/N given by a constant $b - d$ minus a *density dependent inhibition* factor cN

Obtaining the well-known form (3)

$$
\begin{aligned}
N' &= (b-d)N - cN^2 \\
&= \big((b-d) - cN\big)N \\
&= \left(r - \frac{r}{r}cN\right)N \qquad \text{with } r = b-d \text{ the growth rate} \\
&= rN\left(1 - \frac{c}{r}N\right) \\
&= rN\left(1 - \frac{N}{K}\right)
\end{aligned}
$$

with $c/r = 1/K$

$K = r/c$ is the **carrying capacity**

Another interpretation of the logistic equation (3)

Think in terms of the *per capita* growth rate and rewrite slightly the equation

$$N' = rN\left(1 - \frac{N}{K}\right)$$

as

$$\frac{N'}{N} = r\,\frac{K-N}{K}$$

This gives a more intuitive understanding of the role of K in regulating the population: when N is smaller than K, the population grows, when N is larger than K, the population decreases

The parameter r sets how fast this regulation occurs

Units

It is important, especially later when we look at numerics, to understand the units of the parameters

On the left, $dN(t)/dt$ has units of N/t (or $\#/t$), i.e., population per unit time

Consider three potential right hand sides, e.g., for death

$$N' = -d, \qquad N' = -dN \quad \& \quad N' = -dN^2$$

First case: d has units of $\#/t$. Second case, d has units of $1/t$ (since dN has units $\#/t$ with N having units $\#$). Third case, d has units of $1/(t\#)$ (since N^2 has units of $\#\#$)

Several ways to tackle this equation

1. The equation is separable [explicit method]

2. The equation is a Bernoulli equation [explicit method]

3. Use qualitative analysis

4. Use numerical methods

Population growth – The logistic equation & friends

- Ordinary differential equations 101
- Formulating the logistic equation
- From birth-death to the logistic
- Qualitative analysis of the logistic equation
- Including an Allee effect
- Gompertz growth model
- Tumor growth
- Harvest model
- Computational analysis of the logistic equation

Studying the logistic equation qualitatively

We study

$$N' = rN\left(1 - \frac{N}{K}\right) \tag{3}$$

For this, write

$$f(N) = rN\left(1 - \frac{N}{K}\right)$$

Consider the initial value problem consisting of (3) and an initial condition

$$N' = f(N), \quad N(0) = N_0 > 0 \tag{7}$$

- f is C^1 so solutions to (7) exist and are unique

Equilibria of the logistic equation

Equilibria of (3) are points such that $f(N) = 0$. So we solve $f(N) = 0$ for N:

$$rN\left(1 - \frac{N}{K}\right) = 0 \iff rN = 0 \text{ or } 1 - \frac{N}{K} = 0$$

So we find two points:

- $N = 0$
- $N = K$

By uniqueness of solutions to (7), solutions cannot cross the lines (solutions) $N(t) = 0$ and $N(t) = K$

Several cases

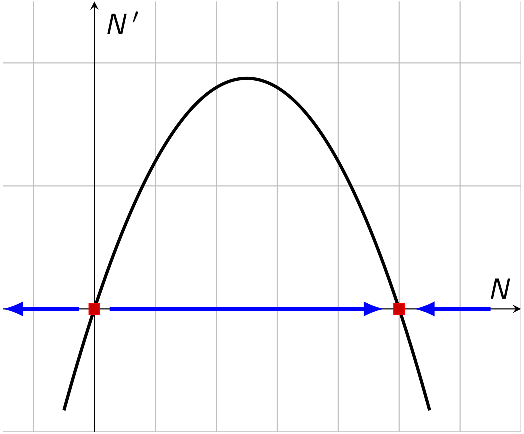
- $N = 0$ for some t, then $N(t) = 0$ for all $t \geq 0$, by uniqueness of solutions

- $N \in (0, K)$, then $rN > 0$ and $N/K < 1$ so $1 - N/K > 0$, which implies that $f(N) > 0$. As a consequence, $N(t)$ increases if $N \in (0, K)$

- $N = K$, then $rN > 0$ but $N/K = 1$ so $1 - N/K = 0$, which implies that $f(N) = 0$. As a consequence, $N(t) = K$ for all $t \geq 0$, by uniqueness of solutions

- $N > K$, the $rN > 0$ and $N/K > 1$, implying that $1 - N/K < 0$ and in turn, $f(N) < 0$. As a consequence, $N(t)$ decreases if $N \in (K, +\infty)$

Stability of the equilibria



Theorem 9

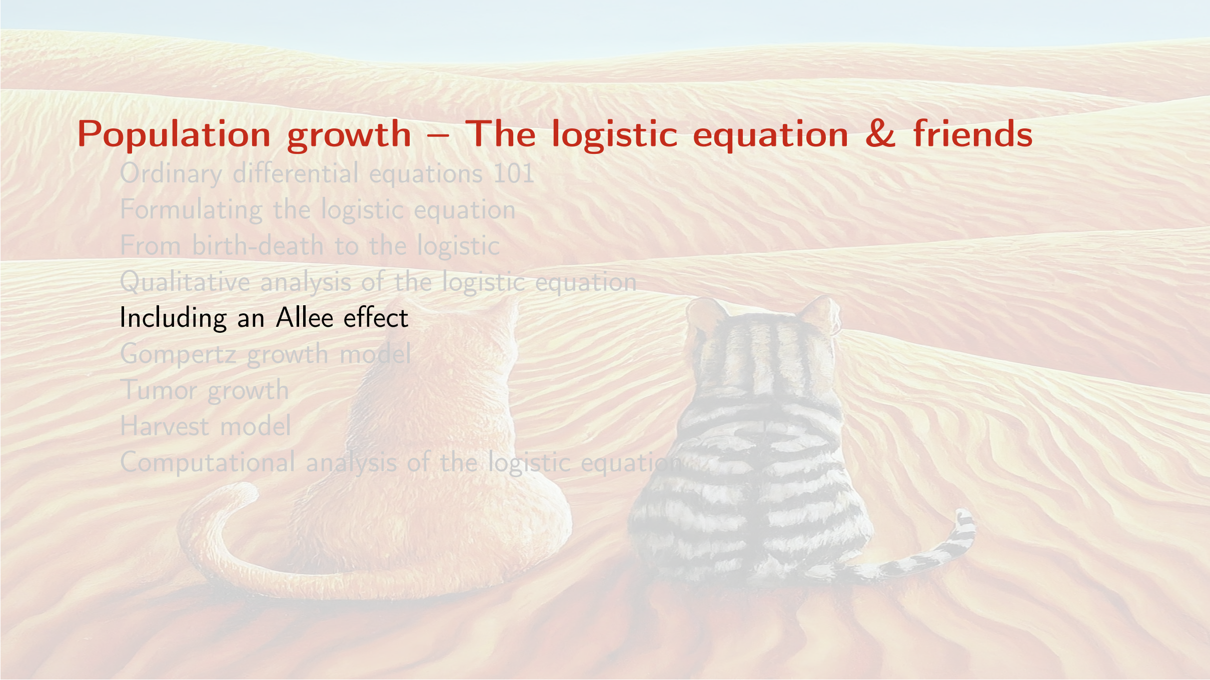
Consider the initial value problem (7). Then

- *If $N_0 = 0$, then $N(t) = 0$ for all $t \geq t_0$*
- *If $N_0 > 0$, then $N(t)$ is such that*

$$\lim_{t \to \infty} N(t) = K$$

*so that K is the number of individuals that the environment can support, the **carrying capacity** of the environment*

We could be more precise and break up the second case depending on whether $N_0 < K$, $N_0 = K$ or $N_0 > K$, but the end result is the same



Population growth – The logistic equation & friends

Including an Allee effect

Logistic equation assumes that when the population is less than the carrying capacity, the population grows. But what if the population is too small?

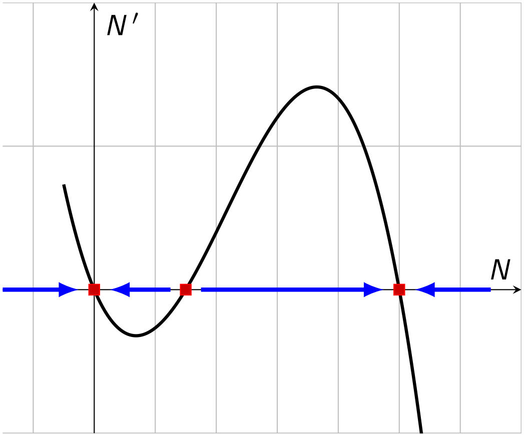
There may not be enough individuals to find mates, or to find food, or to protect themselves from predators. This is an **Allee effect**

Logistic equation with an Allee effect

$$N' = rN\left(\frac{N}{A} - 1\right)\left(1 - \frac{N}{K}\right) \tag{8}$$

- $r > 0$ growth rate
- $K > 0$ carrying capacity
- $A > 0$ Allee threshold

Stability of the equilibria



Population growth – The logistic equation & friends

Deriving the Gompertz growth model – Assumptions

Let us think for instance about the growth of a tumour

Solid tunours grow by cell division, and the rate of cell division is proportional to the number of cells present

Solid tumours do not grow exponentially with time

As the tumour becomes larger, the doubling time of total tumour volume increases

The Gompertz growth model – V1.0

Let $x(t)$ be the volume of dividing cells at time t, with $\alpha, k > 0$ constants

$$x' = ke^{-\alpha t}x \tag{9}$$

This is a *nonautonomous* ODE, harder to study than autonomous ODEs

Eliminating time-dependence

Remember what you have probably heard about General Relativity: time is just another dimension. Any nonautonomous ODE can be transformed into an autonomous one, just with one more dimension

Let τ be the "new time", then (9) becomes

$$\frac{d}{d\tau}x = ke^{-\alpha t}x$$
$$\frac{d}{d\tau}t = 1$$

Another method

Let $a(t) = ke^{-\alpha t}$, then (9) becomes

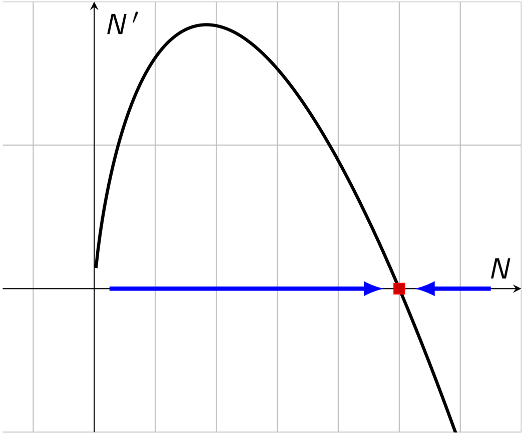
$$\frac{d}{dt}x = a(t)x \tag{10a}$$

$$\frac{d}{dt}a = -\alpha a(t) \tag{10b}$$

Gompertz growth model

$$N' = rN \ln\left(\frac{K}{N}\right) \tag{11}$$

Stability of the equilibria



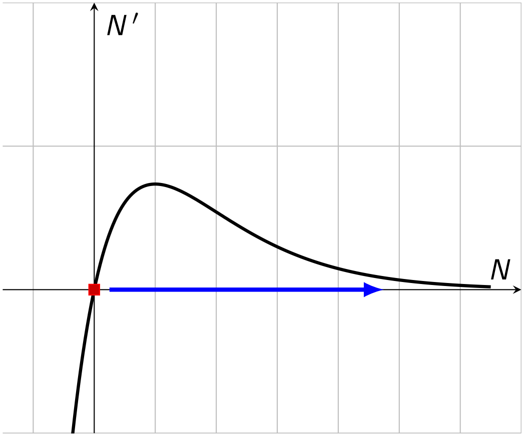
Population growth – The logistic equation & friends

Tumor growth

$$N' = rNe^{-bN} \quad (12)$$

- $r > 0$ growth rate
- > 0

Stability of the equilibria



Solutions are always increasing, so $N(t) \to \infty$ as $t \to \infty$

However, the rate of growth decreases as N increases

Population growth – The logistic equation & friends

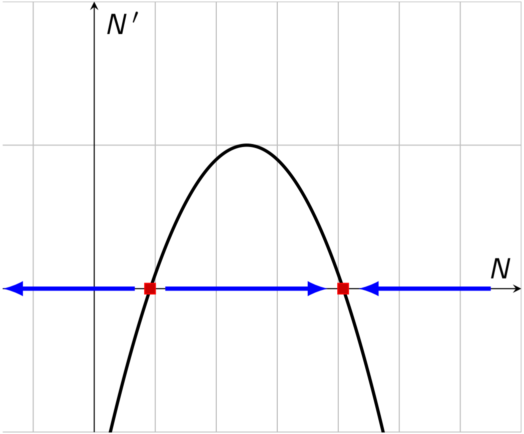
- Ordinary differential equations 101
- Formulating the logistic equation
- From birth-death to the logistic
- Qualitative analysis of the logistic equation
- Including an Allee effect
- Gompertz growth model
- Tumor growth
- **Harvest model**
- Computational analysis of the logistic equation

Logistic model with harvesting

$$N' = rN\left(1 - \frac{N}{K}\right) - H \tag{13}$$

- $r > 0$ growth rate
- $K > 0$ carrying capacity
- $H > 0$ harvest rate

Stability of the equilibria



Solutions can become negative, so to make the model realistic, we want to ensure that the lower equilibrium (left red point in the plot) is nonpositive

Solving for equilibria, we get

$$N^\star = \frac{Kr \pm \sqrt{rK(rK - 4H)}}{2r}$$

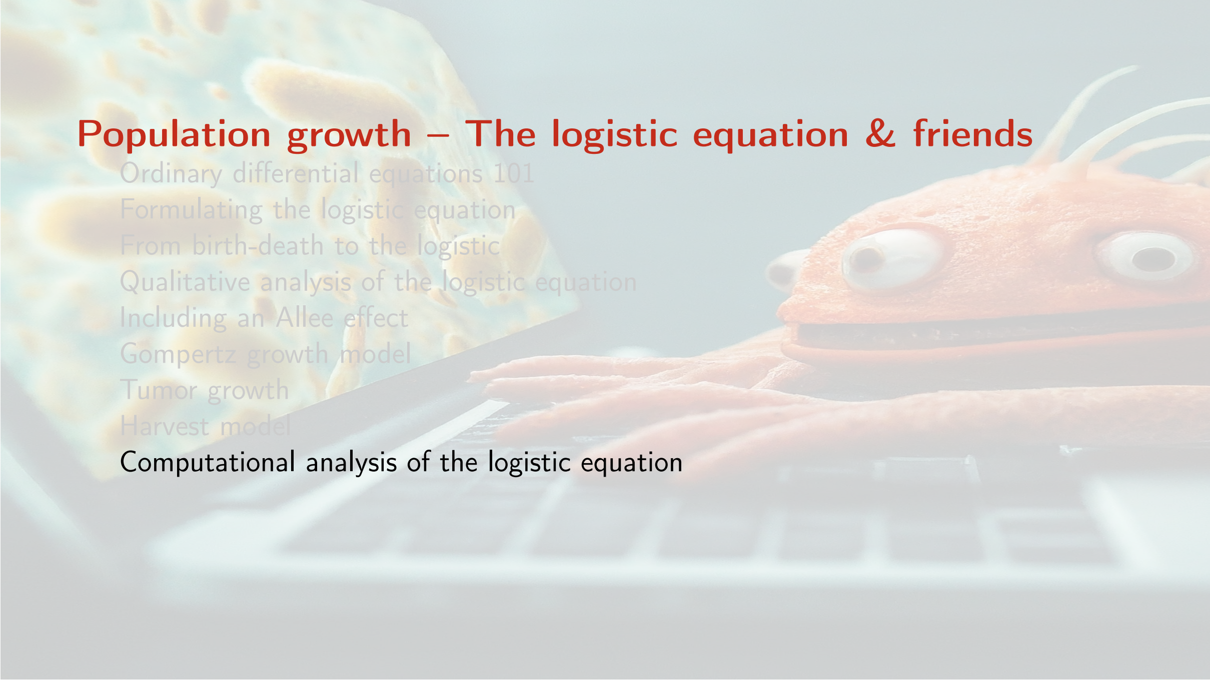
So we want $rK - 4H \geq 0$ and $rK - \sqrt{rK(rK - 4H)} \leq 0$

However

$$
\begin{aligned}
rK - \sqrt{rK(rK-4H)} \leq 0 &\iff rK \leq \sqrt{rK(rK-4H)} \\
&\iff r^2K^2 \leq rK(rK-4H) \\
&\iff rK \leq rK - 4H
\end{aligned}
$$

which is only true for $H = 0$ when $H \geq 0$

So the model always has some solutions that go to $-\infty$ as $t \to \infty$, even with positive initial conditions, when $H > 0$ and $N(0) < N^\star_{\min}$ (the smallest of the two equilibria)



Population growth – The logistic equation & friends

Motivation of the problem

Qualitative methods provide us with an understanding of the behaviour of an ODE model

Typically, we obtain information about what happens as $t \to \infty$

What happens during *transients*? Much (much much!) harder mathematical problem. Even for such simple objects as ODEs, we don't know much

$\Rightarrow$ It is useful to be able to approximate solutions numerically

Guiding principle

Let $x(t)$ be a solution to the IVP

$$\frac{d}{dt}x(t) = f(t, x(t)) \tag{1a}$$

$$x(t_0) = x_0 \tag{1b}$$

By construction, at each $(t, x) \in \mathbb{R} \times \mathbb{R}^n$, $x(t)$ is tangent to $f(t, x)$, since $x' = f(t, x)$

Use this to construct an approximation to the solution

Euler's method

Suppose $x(t) \in \mathbb{R}$ and $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, i.e., we have a scalar ODE. Then the derivative of $x(t)$ takes the form

$$x'(t) = \frac{d}{dt}x(t) = \lim_{h\to 0} \frac{x(t+h) - x(t)}{h}$$

This equals the right hand side, i.e.,

$$\lim_{h\to 0} \frac{x(t+h) - x(t)}{h} = f(t, x(t))$$

Instead of letting $h \to 0$, consider a set value of $h > 0$

$$\frac{x(t+h) - x(t)}{h} = f(t, x(t)) \iff x(t+h) = x(t) + h\, f(t, x(t))$$

Example for birth-death model

Model (4) is autonomous, so $f(t,x) \equiv f(x)$ for all t

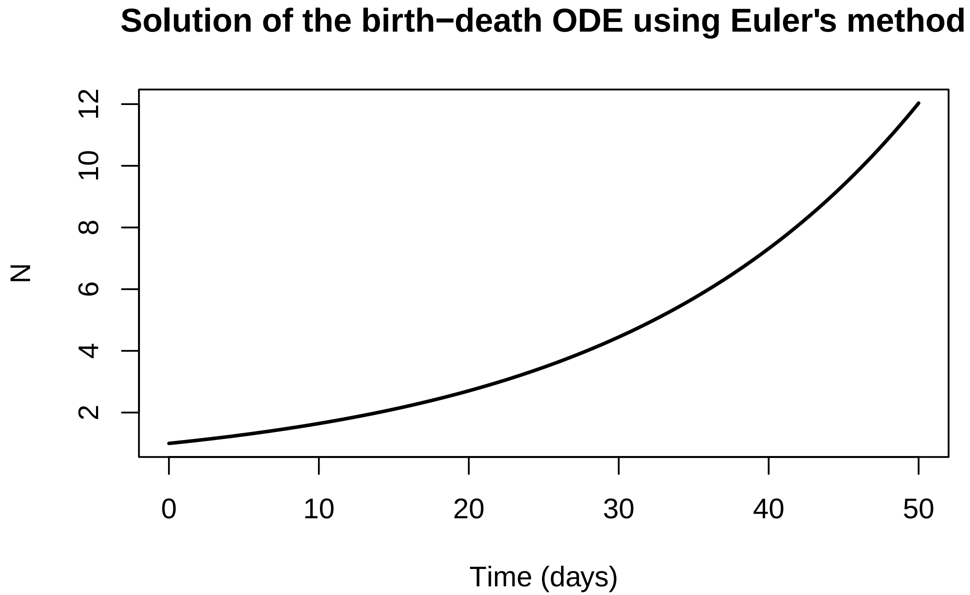
For simplicity, let $t_0 = 0$. Consider an IC $N(0) = N_0$. Fix $h > 0$

Solution is then a sequence of times $t_0, t_1 = t_0 + h$, $t_2 = t_1 + h = t_0 + 2h$, ..., and, for $i = 0, 1, \ldots$, of states

$$
\begin{aligned}
N_{1+1} &= N_i + h\ f(N_i) \\
&= N_i + h(bN_i - dN_i) \\
&= (1 + hb - hd)N_i
\end{aligned}
$$

Example for birth-death model

```
t = c(0)
N = c(1)
b = 0.1
d = 0.05
h = 0.2
t_f = 50
while (t[length(t)] <= t_f) {
  N = c(N, (1+h*b-h*d)*N[length(N)])
  t = c(t, t[length(t)]+h)
}
plot(t,N,
     type = "l", lwd = 2,
     main = "Solution␣of␣the␣birth-death␣ODE␣using␣Euler's␣method",
     xlab = "Time␣(days)")
```


Solution of the birth−death ODE using Euler's method
N
2
4
6
8
10
12
0
10
20
30
40
50
Time (days)

The step-size is important

Clearly, we have the "true" solution if we let $h \to 0$. To show the influence of the step size, let us make a function that allows us to easily change all parameters involved in the numerical solution

We will then call it with different values of h

Example for birth-death model

```
Euler_approximation_Malthus = function(p,
                                        t_0 = 0, N_0 = 1,
                                        h = 0.5, t_f = 50) {
  t = c(t_0)
  N = c(N_0)
  while (t[length(t)] <= t_f) {
    N = c(N, (1+h*p$b-h*p$d)*N[length(N)])
    t = c(t, t[length(t)]+h)
  }
  OUT = data.frame(t = t, N = N)
  return(OUT)
}
```

Default values of function arguments in R

Note what we did when defining the function

```
Euler_approximation_Malthus = function(p,
                                       t_0 = 0, N_0 = 1,
                                       h = 0.5, t_f = 50) {
  ...
}
```

This means that we have indicated *default values* for some of the parameters. If such a default value is provided, then unless a value is specified, the default value is used

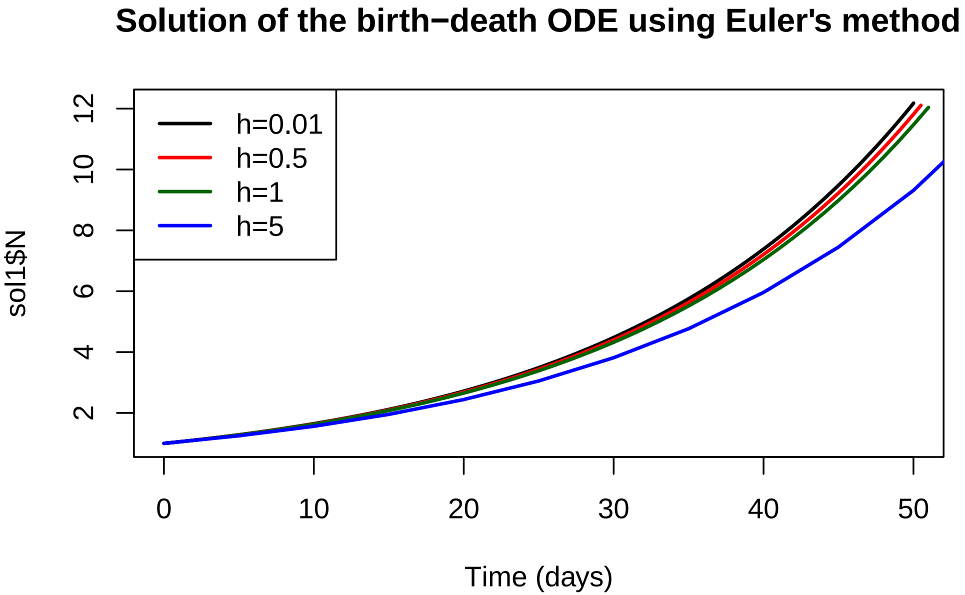
Using lists in R

At this point, it not super important to do that, but when things become more complicated later in the course, it will be useful to pass parameters to the differential equation solvers using R *lists*

```
p = list()
p$b = 0.1
p$d = 0.05
```

Plotting a few cases

```
sol1 = Euler_approximation_Malthus(p, h = 0.01)
sol2 = Euler_approximation_Malthus(p)
sol3 = Euler_approximation_Malthus(p, h = 1)
sol4 = Euler_approximation_Malthus(p, h = 5)
plot(sol1$t, sol1$N,
     type = "l", lwd = 2,
     main = "Solution␣of␣the␣birth-death␣ODE␣using␣Euler's␣method",
     xlab = "Time␣(days)")
lines(sol2$t, sol2$N, type = "l", lwd = 2, col = "red")
lines(sol3$t, sol3$N, type = "l", lwd = 2, col = "darkgreen")
lines(sol4$t, sol4$N, type = "l", lwd = 2, col = "blue")
legend("topleft",
       legend = c("h=0.01", "h=0.5", "h=1", "h=5"),
       lty = 1, lwd = 2,
       col = c("black", "red", "darkgreen", "blue"))
```


Solution of the birth−death ODE using Euler's method
h=0.01
h=0.5
h=1
h=5
sol1$N
2
4
6
8
10
12
0
10
20
30
40
50
Time (days)

There are more complex algorithms

Here, we have considered one of the simplest possible ODEs: a scalar linear autonomous one

If there is more than one equation or if the equations are nonlinear, then the situation is more complex

Euler often performs poorly in this case (see Logistic map later in these slides)

There are many other algorithms that are more efficient and accurate

The "philosophy" of the algorithms is the same, though: approximate the solution by taking small steps using the vector field

Numerical ODE "solvers" in R

There are many packages in R that can solve ODEs

The most popular one is `deSolve`. Among other reasons:

> *Functions that solve initial value problems of a system of first-order ordinary differential equations ('ODE'), of partial differential equations ('PDE'), of differential algebraic equations ('DAE'), and of delay differential equations. The functions provide an interface to the FORTRAN functions 'lsoda', 'lsodar', 'lsode', 'lsodes' of the 'ODEPACK' collection, to the FORTRAN functions 'dvode', 'zvode' and 'daspk' and a C-implementation of solvers of the 'Runge-Kutta' family with fixed or variable time steps.*

This means you are using very efficient and well-tested algorithms. E.g., ODEPACK is LLNL's library for ODEs from ... 1982!

Side note – Old $\neq$ bad (for numerics at least)

The fact that the algorithms are old does not mean they are bad. In fact, the opposite is true

In days of yore, computers were much slower, had less memory, and were less reliable. This means that the algorithms had to be very efficient and robust

Apollo Guidance Computer (Lunar lander, 1966–1975): ran on a 2.048 MHz computer with 2,048 16-bits words RAM & 36,864 16-bits words ROM

Voyager 1 & 2 (1977): now in interstellar space, still running on a computer with 250,000 instructions per second, 8 KB of RAM, and 64 KB of ROM

Using a numerical ODE solver

Process is almost always the same

1. Write a function that returns the value of $f(t, x, p)$ (where p are the parameters of the model) at a given point (t, x, p)
2. Set initial conditions and interval of time on which the solution (approximation) is desired
3. Call solver with reference to the function, time interval and parameters

Additionally, some solvers work better or faster if you also provide the Jacobian of your system (see later in the course)

Some considerations that will make more sense later

Type of time step –

- Fixed time step
- Variable (adaptive) time step

If you come from the MatLab world, you are used to adaptive time step: time output of `ode45` and other methods is usually the vector at which the solution was computed $\neq$ `deSolve` in R, where time (and resulting solution) is interpolated to desired output time points

Nonstiff versus stiff methods – Some problems are *stiff* (roughly: they don't handle numerical imprecision well) and require specific (and more computationally costly) methods. `lsoda`, the default method in `deSolve`, automatically switches between nonstiff and stiff algorithms

The birth-death model

```r
if (!require(deSolve)) {
  install.packages("deSolve")
  library(deSolve)
}
rhs_Malthus <- function(t, N, p) {
  with(as.list(p), {
    dN <- (b-d) * N
    return(list(dN))
  })
}
p = list()
p$b = 0.1
p$d = 0.05
IC = c(N = 1)
tspan = 0:100
sol = ode(y = IC, times = tspan, func = rhs_Malthus, parms = p)
```

Explanation (1)

```r
if (!require(deSolve)) {
  install.packages("deSolve")
  library(deSolve)
}
```

This loads the `deSolve` package. Another command could be `library(deSolve)`

`require` is a little bit more flexible: it returns a logical value indicating whether the package was successfully loaded or not. This is useful in scripts, as used here: if the package is not installed, it will be installed and loaded

Explanation (2)

```r
rhs_Malthus <- function(t, N, p) {
  with(as.list(p), {
    dN <- (b-d) * N
    return(list(dN))
  })
}
```

The right hand side function. For a given point (t, N, p), returns $(b-d)N$, i.e., N'

Note the `as.list(p)` command. We could instead use `p$b` and `p$d`, but this "unpacks" the variable `p` into its components that can then be used without the $ sign

Note also that in this case, the `return` command must occur within the `with` command

Explanation (3)

```
p = list()
p$b = 0.1
p$d = 0.05
IC = c(N = 1)
tspan = 0:100
```

Here, we setup the list of parameters, set initial conditions and the time span over which we want to solve the ODE

Note that the initial condition is a *named vector*, even if it is a single value. This is useful: if passed a named vector as IC, the solver will return a named matrix with the solution

Explanation (4)

```
sol = ode(y = IC, times = tspan, func = rhs_Malthus, parms = p)
```

Finally, a call to the solver itself. There are many options to the call, but the most basic are those here: initial conditions, the time span, the function and the parameters (the latter is optional)

```
> head(sol, 5)
     time        N
[1,]    0 1.000000
[2,]    1 1.051273
[3,]    2 1.105172
[4,]    3 1.161835
[5,]    4 1.221404
```

RHS – logistic equation(s)

```
rhs_logistic <- function(t, N, p) {
  with(as.list(p), {
    dN <- r * N * (1 - N/K)
    return(list(dN))
  })
}
rhs_logistic_Allee <- function(t, N, p) {
  with(as.list(p), {
    dN <- r * N * (N/A - 1) * (1 - N/K)
    return(list(dN))
  })
}
rhs_logistic_harvesting <- function(t, N, p) {
  with(as.list(p), {
    dN <- r * N * (1 - N/K) - H
    return(list(dN))
  })
}
```

RHS – the other two

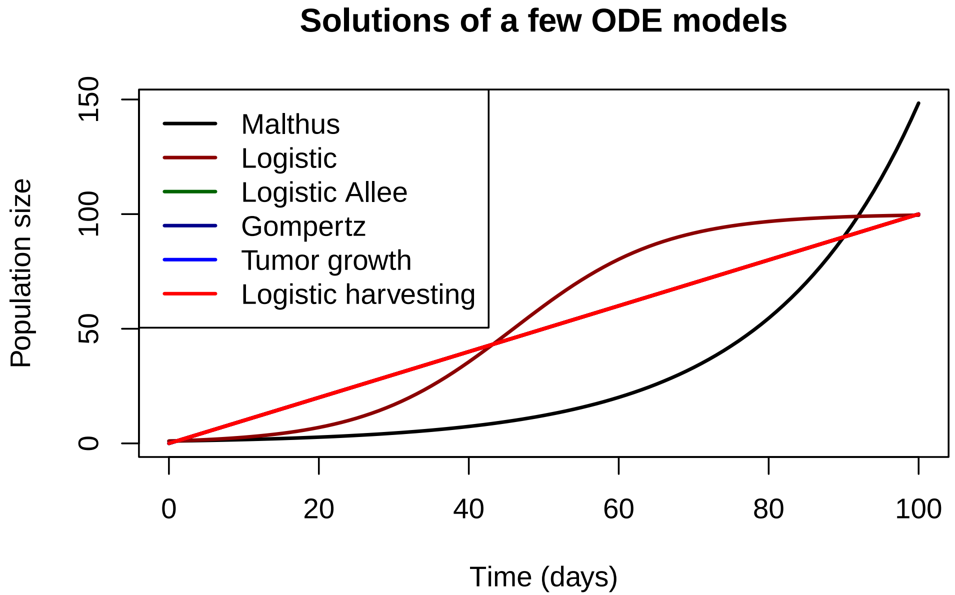
```r
rhs_Gompertz <- function(t, N, p) {
  with(as.list(p), {
    dN <- r * N * log(K/N)
    return(list(dN))
  })
}
rhs_tumor_growth <- function(t, N, p) {
  with(as.list(p), {
    dN <- r * N * exp(-b * N)
    return(list(dN))
  })
}
```

Parameters

```
p = list(
  r = 0.1,
  K = 100,
  A = 50,
  b = 0.1,
  H = 0.05
)
IC = c(N = 1)
tspan = 0:100
sol = ode(y = IC, times = tspan, func = rhs_Malthus, parms = p)
```

Run a few models

```
sol_Malthus =
  ode(y = IC, times = tspan, func = rhs_Malthus, parms = p)
sol_logistic =
  ode(y = IC, times = tspan, func = rhs_logistic, parms = p)
sol_logistic_Allee =
  ode(y = IC, times = tspan, func = rhs_logistic_Allee, parms = p)
sol_Gompertz =
  ode(y = IC, times = tspan, func = rhs_Gompertz, parms = p)
sol_tumor_growth =
  ode(y = IC, times = tspan, func = rhs_tumor_growth, parms = p)
sol_logistic_harvesting =
  ode(y = IC, times = tspan, func = rhs_logistic_harvesting, parms = p)
max_y = max(c(
  max(sol_Malthus[,"N"]),
  max(sol_logistic[,"N"]),
  max(sol_logistic_Allee[,"N"]),
  max(sol_Gompertz[,"N"]),
  max(sol_tumor_growth[,"N"]),
  max(sol_logistic_harvesting[,"N"])))
```



Solutions of a few ODE models
Malthus
Logistic
Logistic Allee
Gompertz
Tumor growth
Logistic harvesting
Population size
0
50
100
150
0
20
40
60
80
100
Time (days)

Investigating the logistic with harvesting

Let us dig a little deeper into the logistic model with harvesting..

First of all, note that with the parameters we chose, we find the equilibria 0.5 and 99.5. That's why in the previous plot, where the initial condition was 1, there were no problems

Let us raise the value of H a little, see what happens

```
> p$H = 0.15
> print(EPs_logistic_harvesting(p))
$low
[1] 1.523201

$high
[1] 98.4768
```

With this value of $N^\star_{\text{min}}$, we can expect things to go bad when we use the initial condition $N(0) = 1$

```
> sol_logistic_harvesting_bad = ode(y = IC, times = tspan,
+       func = rhs_logistic_harvesting, parms = p)
DLSODA-  Warning..Internal T (=R1) and H (=R2) are
      such that in the machine, T + H = T on the next step
     (H = step size). Solver will continue anyway.
In above message, R1 = 53.9164, R2 = 3.13622e-15

DLSODA-  Warning..Internal T (=R1) and H (=R2) are
      such that in the machine, T + H = T on the next step
     (H = step size). Solver will continue anyway.
In above message, R1 = 53.9164, R2 = 3.13622e-15

DLSODA-  Warning..Internal T (=R1) and H (=R2) are
      such that in the machine, T + H = T on the next step
     (H = step size). Solver will continue anyway.
In above message, R1 = 53.9164, R2 = 3.13622e-15
```

The data – US census

Fitting a curve to the data

Least squares problems

Population growth – The logistic equation & friends

The continuous-time Markov chain logistic “equation”

The delayed logistic equation

The logistic map

Continuous-time Markov chains

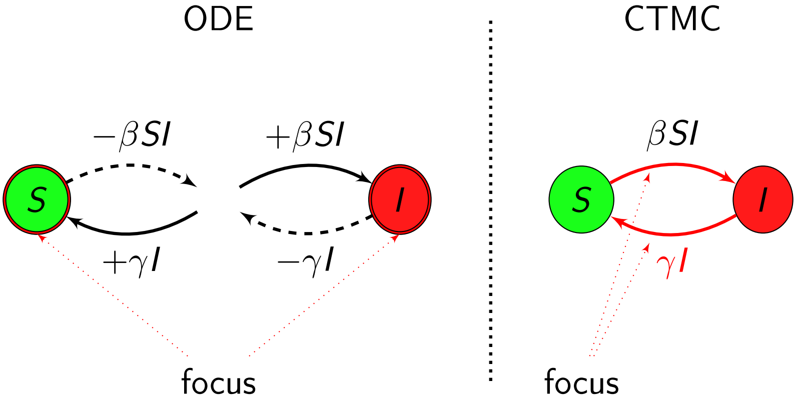
CTMC are roughly equivalent to ODE

Converting your compartmental ODE model to CTMC

Easy as π :)

- Compartmental ODE model focuses on flows into and out of compartments
- ODE model has as many equations as there are compartments
- Compartmental CTMC model focuses on transitions
- CTMC model has as many transitions as there are arrows between (or into or out of) compartments

ODE to CTMC : focus on different components



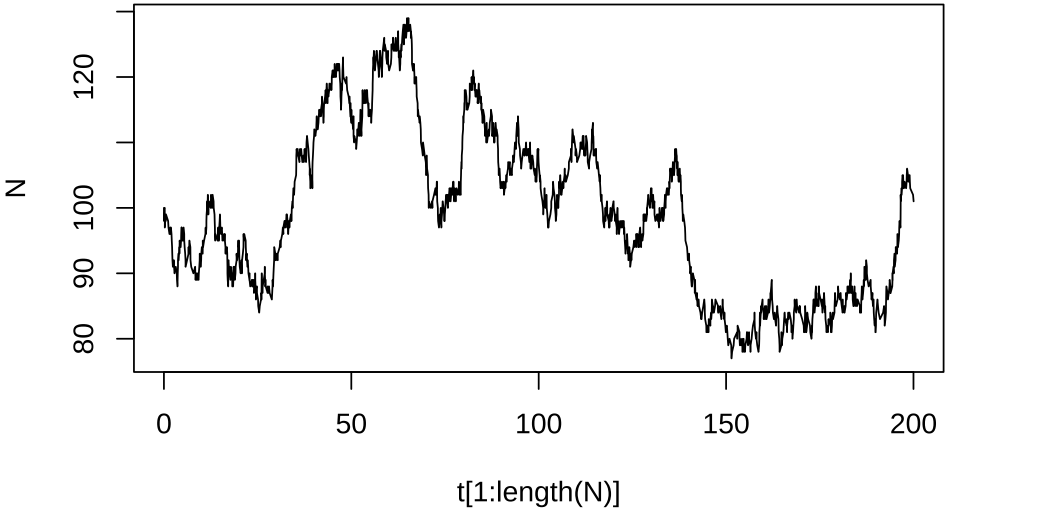
Gillespie algorithm for Malthus – Setup

```
t0 = 0
N0 = 100
tf = 200
b = 0.05
d = 0.05
```

Gillespie algorithm for Malthus – The main loop

```r
t = c(t0)
N = c(N0)

while(t[length(t)] <= tf) {
  xi_t = N[length(N)] * (b+d)
  if (xi_t<1e-12) {
    break
  }
  tau_t = rexp(1, rate = xi_t)
  t = c(t, t[length(t)] + tau_t)
  pi_t = runif(1)
  if(pi_t <= b/(b+d)) {
    N = c(N, N[length(N)] + 1)
  } else {
    N = c(N, N[length(N)] - 1)
  }
}
```



120
100
90
80
N
0
50
100
150
200
t[1:length(N)]

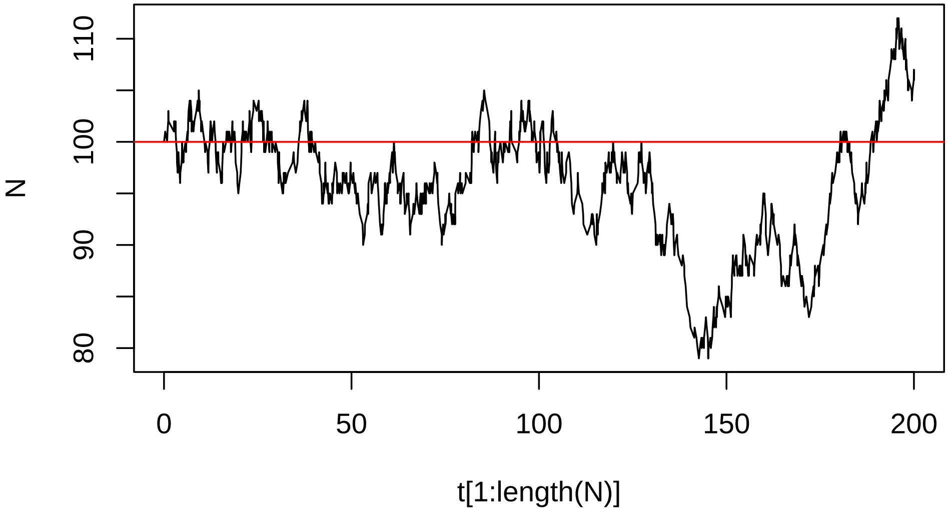
Gillespie algorithm for the logistic – Setup

```
t0 = 0
N0 = 100
tf = 200
b = 0.05
d = 0.02
r = b-d
K = 100
```

Gillespie algorithm for the logistic – The main loop

```
t = c(t0)
N = c(N0)

while(t[length(t)] <= tf) {
  xi_t = r*N[length(N)] + r*N[length(N)]^2/K
  if (xi_t<1e-12) {
    break
  }
  tau_t = rexp(1, rate = xi_t)
  t = c(t, t[length(t)] + tau_t)
  pi_t = runif(1)
  if(pi_t <= 1/(1+N[length(N)]/K)) {
    N = c(N, N[length(N)] + 1)
  } else {
    N = c(N, N[length(N)] - 1)
  }
}
```



110
100
90
80
N
0
50
100
150
200
t[1:length(N)]

The data – US census

Fitting a curve to the data

Least squares problems

Population growth – The logistic equation & friends

The continuous-time Markov chain logistic “equation”

The delayed logistic equation

The logistic map

The delayed logistic equation

Consider the equation as

$$\frac{N'}{N} = (b - d) - cN$$

that is, the per capita rate of growth of the population depends on the net growth rate $b - d$, and some density dependent inhibition cN (resulting of competition)

Suppose that instead of instantaneous inhibition, there is some delay τ between the time the inhibiting event takes place and the moment when it affects the growth rate

For example, two individuals fight for food, and one later dies of the injuries sustained during this fight

The delayed logistic equation

In the case of a time τ between inhibiting event and inhibition, the equation would be written as

$$\frac{N'}{N} = (b - d) - cN(t - \tau)$$

Using the change of variables introduced earlier, this is written

$$N'(t) = rN(t)\left(1 - \frac{N(t-\tau)}{K}\right) \tag{15}$$

Such an equation is called a **delay differential equation** (DDE). It is much more complicated to study than (3). In fact, some things remain unknown about (15)

Delayed initial value problem

The IVP takes the form

$$\begin{aligned} N'(t) &= rN(t)\left(1-\frac{N(t-\tau)}{K}\right) \\ N(t) &= \phi(t) \text{ for } t\in[-\tau,0] \end{aligned} \tag{16}$$

where $\phi(t)$ is some continuous function

Hence, initial conditions (called **initial data** in this case) must be specified on an interval, instead of being specified at a point, to guarantee existence and uniqueness of solutions

We will not learn how to study this type of equation (this is graduate level mathematics). I will give a few results

To find equilibria, remark that delay should not play a role, since N should be constant. Thus, equilibria are found by considering the equation with no delay, which is (3)

Theorem 10

Suppose that $r\tau < \pi/2$. Then solutions of (16) with positive initial data $\phi(t)$ starting close enough to K tend to K. If $r\tau < 37/24$, then all solutions of (16) with positive initial data $\phi(t)$ tend to K. If $r\tau > \pi/2$, then K is an unstable equilibrium and all solutions of (16) with positive initial data $\phi(t)$ on $[-\tau, 0]$ are oscillatory

There is a gray zone between $37/24$ ($\simeq 1.5417$) and $\pi/2$ ($\simeq 1.5708$). The global aspect was proved for $r\tau < 37/24$ in 1945 by Wright. Although there is very strong numerical evidence that this is in fact true up to $\pi/2$, nobody has yet managed to prove it [Edit: now done!]

The data – US census

Fitting a curve to the data

Least squares problems

Population growth – The logistic equation & friends

The continuous-time Markov chain logistic “equation”

The delayed logistic equation

The logistic map

Discrete-time systems

So far, we have seen continuous-time models, where $t \in \mathbb{R}_+$. Another way to model natural phenomena is by using a discrete-time formalism, that is, to consider equations of the form

$$x_{t+1} = f\left(x_t\right)$$

where $t \in \mathbb{N}$ or $\mathbb{Z}$, that is, t takes values in a discrete valued (countable) set

Time could for example be days, years, etc.

The logistic map

The logistic **map** is, for $t \geq 0$,

$$N_{t+1} = rN_t \left(1 - \frac{N_t}{K}\right) \tag{17}$$

To transform this into an initial value problem, we need to provide an initial condition $N_0 \geq 0$ for $t = 0$

Some mathematical analysis

Suppose we have a system in the form

$$x_{t+1} = f\,(x_t)$$

with initial condition given for $t = 0$ by x_0. Then,

$$\begin{aligned} x_1 &= f\,(x_0) \\ x_2 &= f\,(x_1) = f\,(f(x_0)) \stackrel{\Delta}{=} f^{\,2}(x_0) \\ &\vdots \\ x_k &= f^{\,k}(x_0) \end{aligned}$$

The $f^{\,k} = \underbrace{f \circ f \circ \cdots \circ f}_{k \text{ times}}$ are the **iterates** of f

Fixed points

Definition 11 (Fixed point)

Let f be a function. A point p such that $f(p) = p$ is called a **fixed point** of f

Theorem 12

Consider the closed interval $I = [a, b]$. If $f : I \to I$ is continuous, then f has a fixed point in I

Theorem 13

Let I be a closed interval and $f : I \to \mathbb{R}$ be a continuous function. If $f(I) \supset I$, then f has a fixed point in I.

Periodic points

Definition 14 (Periodic point)

Let f be a function. If there exists a point p and an integer n such that

$$f^n(p) = p, \quad \text{but} \quad f^k(p) \neq p \text{ for } k < n,$$

then p is a periodic point of f with (least) period n (or a n-periodic point of f).

Thus, p is a n-periodic point of f iff p is a 1-periodic point of f^n.

Stability of fixed points, of periodic points

Theorem 15

Let f be a continuously differentiable function (that is, differentiable with continuous derivative, or C^1), and p be a fixed point of f.

1. *If $|f'(p)| < 1$, then there is an open interval $\mathcal{I} \ni p$ such that $\lim_{k\to\infty} f^k(x) = p$ for all $x \in \mathcal{I}$.*
2. *If $|f'(p)| > 1$, then there is an open interval $\mathcal{I} \ni p$ such that if $x \in \mathcal{I}$, $x \neq p$, then there exists k such that $f^k(x) \notin \mathcal{I}$.*

Definition 16

Suppose that p is a n-periodic point of f, with $f \in C^1$.

- If $|(f^n)'(p)| < 1$, then p is an **attracting** periodic point of f.
- If $|(f^n)'(p)| > 1$, then p is an **repelling** periodic point of f.

Parametrized families of functions

Consider the equation (17), which for convenience we rewrite as

$$N_{t+1} = rN_t(1 - N_t) \tag{18}$$

where r is a parameter in $\mathbb{R}_+$, and N will typically be taken in $[0, 1]$. Let

$$f_r(x) = rx(1 - x).$$

The function f_r is called a **parametrized family** of functions.

Bifurcations

Definition 17 (Bifurcation)

Let f_μ be a parametrized family of functions. Then there is a **bifurcation** at $\mu = \mu_0$ (or μ_0 is a bifurcation point) if there exists $\varepsilon > 0$ such that, if $\mu_0 - \varepsilon < a < \mu_0$ and $\mu_0 < b < \mu_0 + \varepsilon$, then the dynamics of $f_a(x)$ are "different" from the dynamics of $f_b(x)$.

An example of "different" would be that f_a has a fixed point (that is, a 1-periodic point) and f_b has a 2-periodic point.

Back to the logistic map

Consider the simplified version (18),

$$N_{t+1} = rN_t(1 - N_t) \triangleq f_r(N_t).$$

Are solutions well defined?
Suppose $N_0 \in [0, 1]$, do we stay in $[0, 1]$? f_r is continuous on $[0, 1]$, so it has a extrema on $[0, 1]$. We have

$$f_r'(x) = r - 2rx = r(1 - 2x),$$

which implies that f_r increases for $x < 1/2$ and decreases for $x > 1/2$, reaching a maximum at $x = 1/2$.

$f_r(0) = f_r(1) = 0$ are the minimum values, and $f(1/2) = r/4$ is the maximum. Thus, if we want $N_{t+1} \in [0, 1]$ for $N_t \in [0, 1]$, we need to consider $r \leq 4$.

- Note that if $N_0 = 0$, then $N_t = 0$ for all $t \geq 1$.
- Similarly, if $N_0 = 1$, then $N_1 = 0$, and thus $N_t = 0$ for all $t \geq 1$.
- This is true for all t: if there exists t_k such that $N_{t_k} = 1$, then $N_t = 0$ for all $t \geq t_k$.
- This last case might occur if $r = 4$, as we have seen.
- Also, if $r = 0$ then $N_t = 0$ for all t.

For these reasons, we generally consider

$$N \in (0, 1)$$

and

$$r \in (0, 4).$$

Fixed points: existence

Fixed points of (18) satisfy $N = rN(1-N)$, giving:

- $N = 0$;
- $1 = r(1-N)$, that is, $p \triangleq \dfrac{r-1}{r}$.

Note that $\lim_{r\to 0^+} p = 1 - \lim_{r\to 0^+} 1/r = -\infty$, $\frac{\partial}{\partial r} p = 1/r^2 > 0$ (so p is an increasing function of r), $p = 0 \Leftrightarrow r = 1$ and $\lim_{r\to\infty} p = 1$. So we come to this first conclusion:

- 0 always is a fixed point of f_r.
- If $0 < r < 1$, then p takes negative values so is not relevant.
- If $1 < r < 4$, then p exists.

Stability of the fixed points

Stability of the fixed points is determined by the (absolute) value f_r' at these fixed points. We have

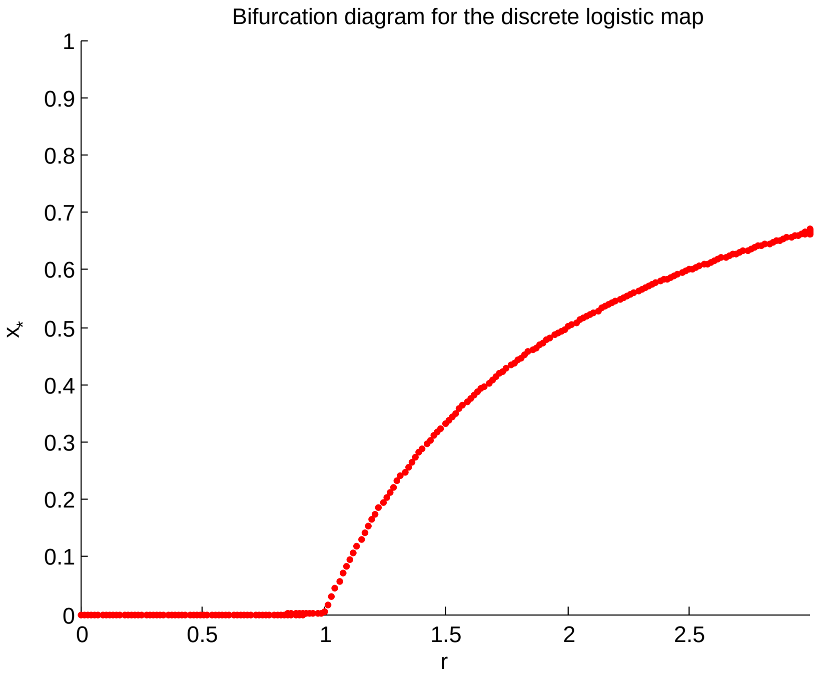
$$|f_r'(0)| = r,$$

and

$$\begin{aligned} |f_r'(p)| &= \left| r - 2r\frac{r-1}{r} \right| \\ &= |r - 2(r-1)| \\ &= |2 - r| \end{aligned}$$

Therefore, we have

- if $0 < r < 1$, then the fixed point $N = p$ does not exist and $N = 0$ is attracting,
- if $1 < r < 3$, then $N = 0$ is repelling, and $N = p$ is attracting,
- if $r > 3$, then $N = 0$ and $N = p$ are repelling.


Bifurcation diagram for the discrete logistic map
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
x_*
0
0.5
1
1.5
2
2.5
r

Another bifurcation

Thus the points $r = 1$ and $r = 3$ are bifurcation points. To see what happens when $r > 3$, we need to look for period 2 points.

$$\begin{aligned} f_r^2(x) &= f_r(f_r(x)) \\ &= r f_r(x)(1 - f_r(x)) \\ &= r^2 x(1-x)(1 - rx(1-x)). \end{aligned} \tag{19}$$

0 and p are points of period 2, since a fixed point x^* of f satisfies $f(x^*) = x^*$, and so, $f^2(x^*) = f(f(x^*)) = f(x^*) = x^*$.
This helps localizing the other periodic points. Writing the fixed point equation as

$$Q(x) \triangleq f_r^2(x) - x = 0,$$

we see that, since 0 and p are fixed points of f_μ^2, they are roots of $Q(x)$. Therefore, Q can be factorized as

$$Q(x) = x(x-p)(-r^3x^2 + Bx + C),$$

Substitute the value $(r-1)/r$ for p in Q, develop Q and (19) and equate coefficients of like powers gives

$$Q(x) = x\left(x - \frac{r-1}{r}\right)\left(-r^3x^2 + r^2(r+1)x - r(r+1)\right). \tag{20}$$

We already know that $x = 0$ and $x = p$ are roots of (20). So we search for roots of

$$R(x) := -r^3x^2 + r^2(r+1)x - r(r+1).$$

Discriminant is

$$\begin{aligned}\Delta &= r^4(r+1)^2 - 4r^4(r+1)\\ &= r^4(r+1)(r+1-4)\\ &= r^4(r+1)(r-3).\end{aligned}$$

Therefore, R has distinct real roots if $r > 3$. Remark that for $r = 3$, the (double) root is $p = 2/3$. For $r > 3$ but very close to 3, it follows from the continuity of R that the roots are close to $2/3$.

Descartes' rule of signs

Theorem 18 (Descartes' rule of signs)

Let $p(x) = \sum_{i=0}^{m} a_i x^i$ *be a polynomial with real coefficients such that* $a_m \neq 0$. *Define* v *to be the number of variations in sign of the sequence of coefficients* $a_m, \ldots, a_0$. *By 'variations in sign' we mean the number of values of* n *such that the sign of* a_n *differs from the sign of* a_{n-1}, *as* n *ranges from* m *down to 1. Then*

- *the number of positive real roots of* $p(x)$ *is* $v - 2N$ *for some integer* N *satisfying* $0 \leq N \leq \dfrac{v}{2}$,
- *the number of negative roots of* $p(x)$ *may be obtained by the same method by applying the rule of signs to* $p(-x)$.

Example of use of Descartes' rule

Example 19

Let

$$p(x) = x^3 + 3x^2 - x - 3.$$

Coefficients have signs $++--$, i.e., 1 sign change. Thus $v = 1$. Since $0 \leq N \leq 1/2$, we must have $N = 0$. Thus $v - 2N = 1$ and there is exactly one positive real root of $p(x)$.

To find the negative roots, we examine $p(-x) = -x^3 + 3x^2 + x - 3$. Coefficients have signs $-++-$, i.e., 2 sign changes. Thus $v = 2$ and $0 \leq N \leq 2/2 = 1$. Thus, there are two possible solutions, $N = 0$ and $N = 1$, and two possible values of $v - 2N$. Therefore, there are either two or no negative real roots. Furthermore, note that $p(-1) = (-1)^3 + 3 \cdot (-1)^2 - (-1) - 3 = 0$, hence there is at least one negative root. Therefore there must be exactly two.

Back to the logistic map and the polynomial R..

We use Descartes' rule of signs.

- R has signed coefficients $- + -$, so 2 sign changes imlying 0 or 2 positive real roots.
- $R(-x)$ has signed coefficients $- - -$, so no negative real roots.
- Since $\Delta > 0$, the roots are real, and thus it follows that both roots are positive.

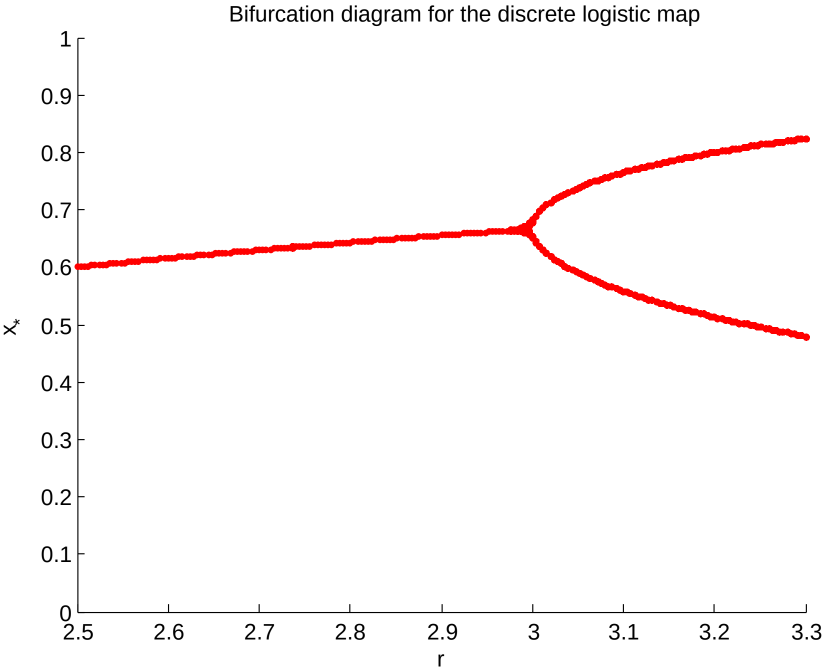
To show that the roots are also smaller than 1, consider the change of variables $z = x - 1$. The polynomial R is transformed into

$$\begin{aligned} R_2(z) &= -r^3(z+1)^2 + r^2(r+1)(z+1) - r(r+1) \\ &= -r^3 z^2 + r^2(1-r)z - r. \end{aligned}$$

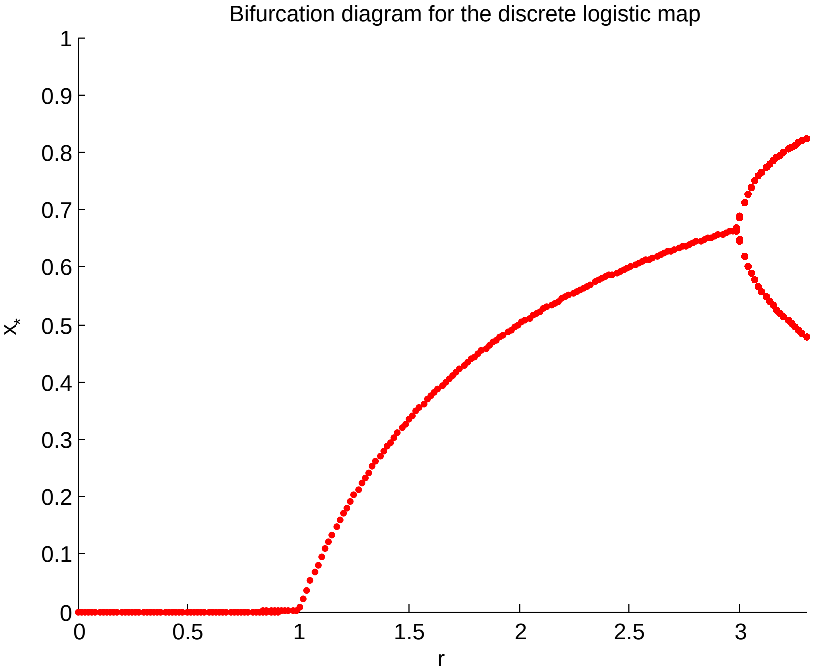
For $r > 1$, the signed coefficients are $- - -$, so R_2 has no root $z > 0$, implying in turn that R has no root $x > 1$.

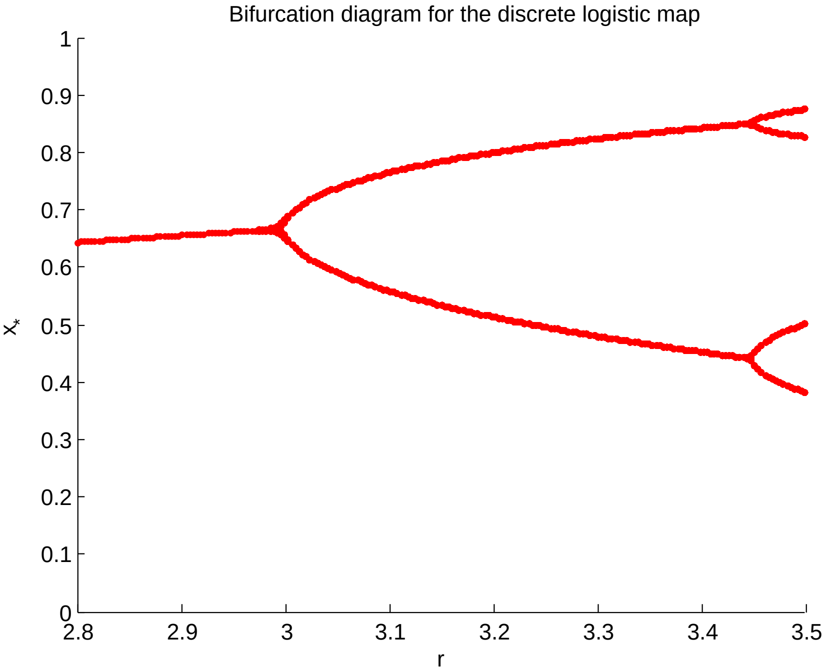
Summing up

- If $0 < r < 1$, then $N = 0$ is attracting, p does not exist and there are no period 2 points.
- At $r = 1$, there is a bifurcation (called a **transcritical** bifurcation).
- If $1 < r < 3$, then $N = 0$ is repelling, $N = p$ is attracting, and there are no period 2 points.
- At $r = 3$, there is another bifurcation (called a **period-doubling** bifurcation).
- For $r > 3$, both $N = 0$ and $N = p$ are repelling, and there is a period 2 point.



Bifurcation diagram for the discrete logistic map
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
x_*
2.5
2.6
2.7
2.8
2.9
3
3.1
3.2
3.3
r


Bifurcation diagram for the discrete logistic map
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
x_*
0
0.5
1
1.5
2
2.5
3
r


Bifurcation diagram for the discrete logistic map
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
x*
2.8
2.9
3
3.1
3.2
3.3
3.4
3.5
r

The period-doubling cascade to chaos

The logistic map undergoes a sequence of period doubling bifurcations, called the **period-doubling cascade**, as r increases from 3 to 4.

- Every successive bifurcation leads to a doubling of the period.
- The bifurcation points form a sequence, $\{r_n\}$, that has the property that

$$\lim n \to \infty \frac{r_n - r_{n-1}}{r_{n+1} - r_n}$$

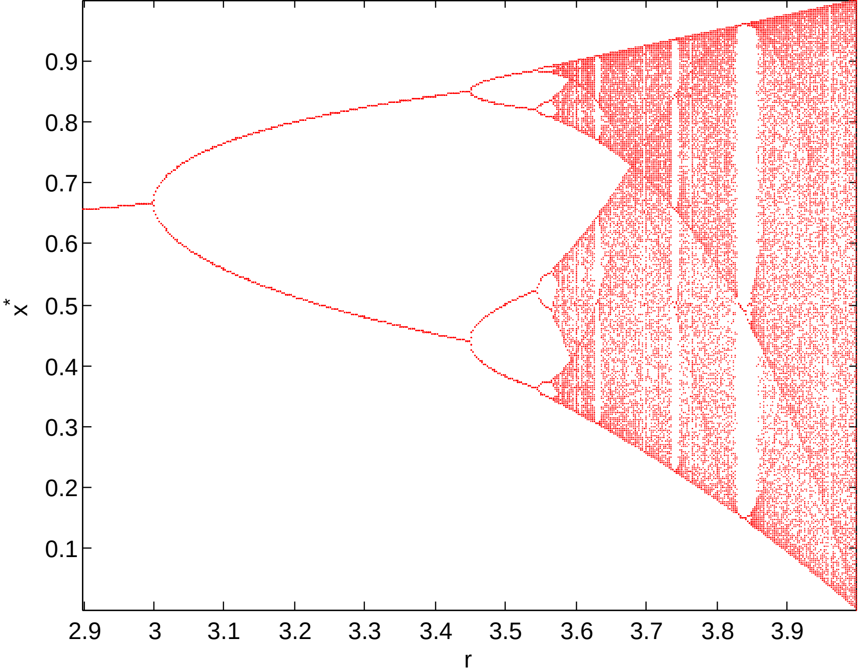
  exists and is a constant, called the Feigenbaum constant, equal to 4.669202...
- This constant has been shown to exist in many of the maps that undergo the same type of cascade of period doubling bifurcations.

Chaos

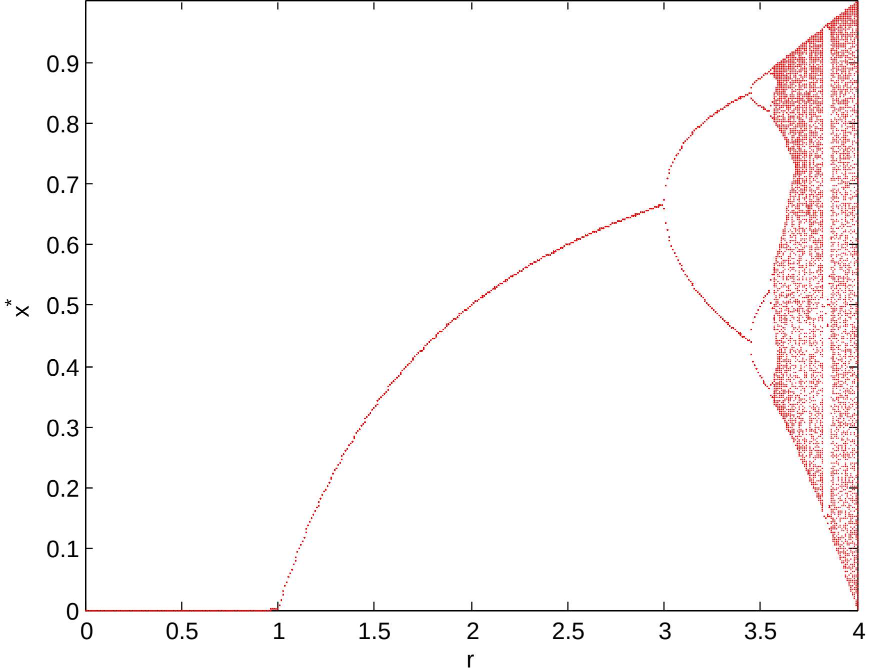
After a certain value of r, there are periodic points with all periods. In particular, there are periodic points of period 3.

By a theorem (called **Sarkovskii's theorem**), the presence of period 3 points implies the presence of points of all periods.

At this point, the system is said to be in a chaotic regime, or chaotic.



0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
x*
2.9
3
3.1
3.2
3.3
3.4
3.5
3.6
3.7
3.8
3.9
r



0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
x*
0
0.5
1
1.5
2
2.5
3
3.5
4
r

Conclusion – A word of caution

We have used four different modelling paradigms to describe the growth of a population in a **logistic** framework:

- The ODE version has monotone solutions converging to the carrying capacity K
- The CTMC version behaves like the ODE version, but with a stochastic component
- The DDE version has oscillatory solutions, either converging to K or, if the delay is too large, periodic about K
- The discrete time version has all sorts of behaviours, including chaotic

The **choice of modelling method** is almost **as important** in the outcome of the model as the precise formulation/hypotheses of the **model**