Computational Modelling

Luscombe Lab Tech Meeting - 2022-06-29
Marc Jones

Law of Mass Action

“Rates of chemical reactions are directly proportional to the product of the concentrations of the reactants”

Law of Mass Action

“Rates of chemical reactions are directly proportional to the product of the concentrations of the reactants”

$$\varnothing \overset{k_{prod}}{\rightarrow} mRNA$$

$$mRNA \overset{k_{degr}}{\rightarrow} \varnothing$$

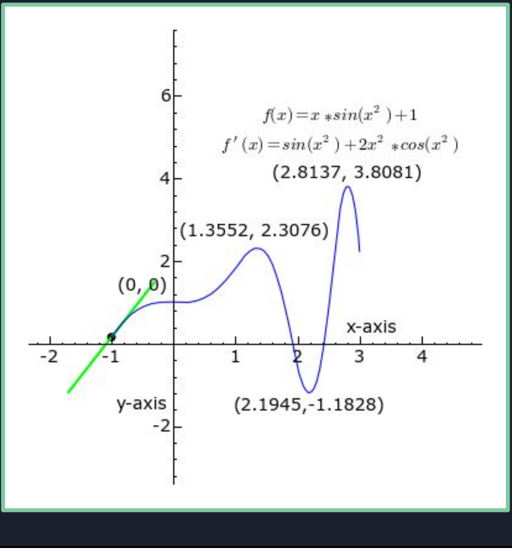
$$\frac{\mathrm{d}[mRNA]}{\mathrm{d}t} = k_{prod} - k_{degr}[mRNA]$$

Ordinary Differential Equations

“Ordinary” because they only depend on one independent variable, in this case, time (*t*)

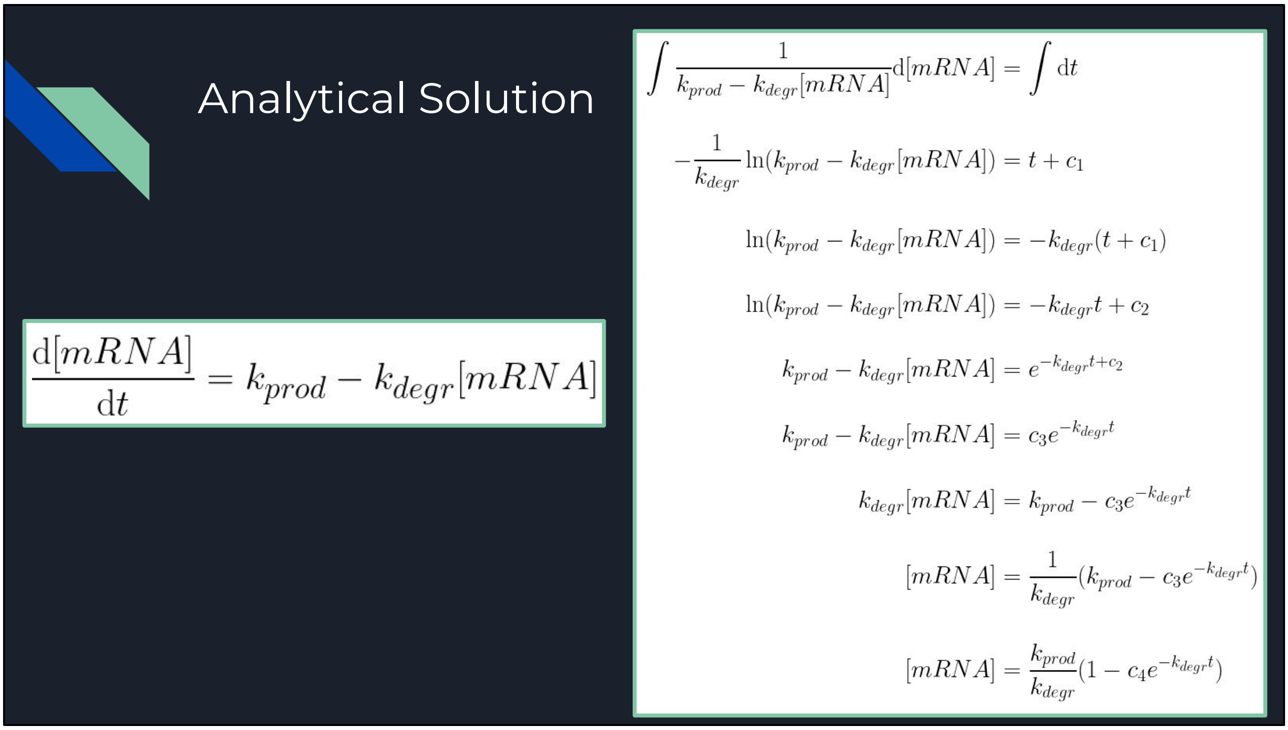
$$\frac{\mathrm{d}[mRNA]}{\mathrm{d}t} = k_{prod} - k_{degr}[mRNA]$$

Ordinary Differential Equations



Analytical Solution

$$\boxed{\frac{\mathrm{d}[mRNA]}{\mathrm{d}t} = k_{prod} - k_{degr}[mRNA]}$$



Analytical Solution

$$\frac{\mathrm{d}[mRNA]}{\mathrm{d}t} = k_{prod} - k_{degr}[mRNA]$$

$$\begin{aligned}
\int \frac{1}{k_{prod} - k_{degr}[mRNA]}\mathrm{d}[mRNA] &= \int \mathrm{d}t \\
-\frac{1}{k_{degr}}\ln(k_{prod} - k_{degr}[mRNA]) &= t + c_1 \\
\ln(k_{prod} - k_{degr}[mRNA]) &= -k_{degr}(t + c_1) \\
\ln(k_{prod} - k_{degr}[mRNA]) &= -k_{degr}t + c_2 \\
k_{prod} - k_{degr}[mRNA] &= e^{-k_{degr}t + c_2} \\
k_{prod} - k_{degr}[mRNA] &= c_3 e^{-k_{degr}t} \\
k_{degr}[mRNA] &= k_{prod} - c_3 e^{-k_{degr}t} \\
[mRNA] &= \frac{1}{k_{degr}}(k_{prod} - c_3 e^{-k_{degr}t}) \\
[mRNA] &= \frac{k_{prod}}{k_{degr}}(1 - c_4 e^{-k_{degr}t})
\end{aligned}$$

Analytical Solution

$$\frac{\mathrm{d}[mRNA]}{\mathrm{d}t} = k_{prod} - k_{degr}[mRNA]$$

$$[mRNA] = \frac{k_{prod}}{k_{degr}}(1 - c_4 e^{-k_{degr}t})$$

$$0 = \frac{k_{prod}}{k_{degr}}(1 - c_4 e^0)$$

$$0 = \frac{k_{prod}}{k_{degr}}(1 - c_4)$$

$$c_4 = 1$$

DEMO!

$$[mRNA] = \frac{k_{prod}}{k_{degr}}(1 - e^{-k_{degr}t})$$

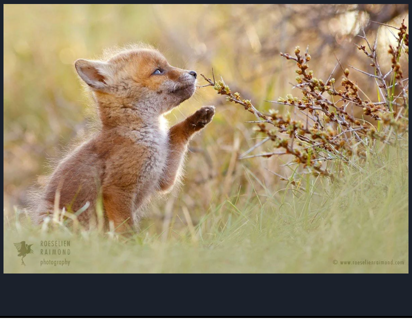
Lotka-Volterra Equations

x - Prey

$$\frac{dx}{dt} = \alpha x - \beta xy,$$

y - Predator

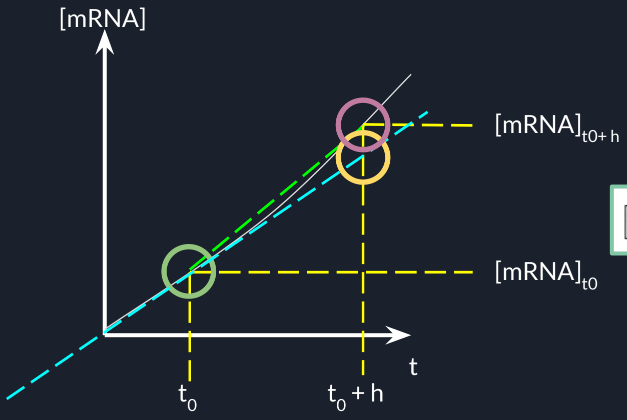
$$\frac{dy}{dt} = \delta xy - \gamma y,$$



Alpha - reproductive rate of prey
Beta - rate of death due to predator
Delta - reproductive rate of predator, based on feeding on prey
Gamma - death rate of predator

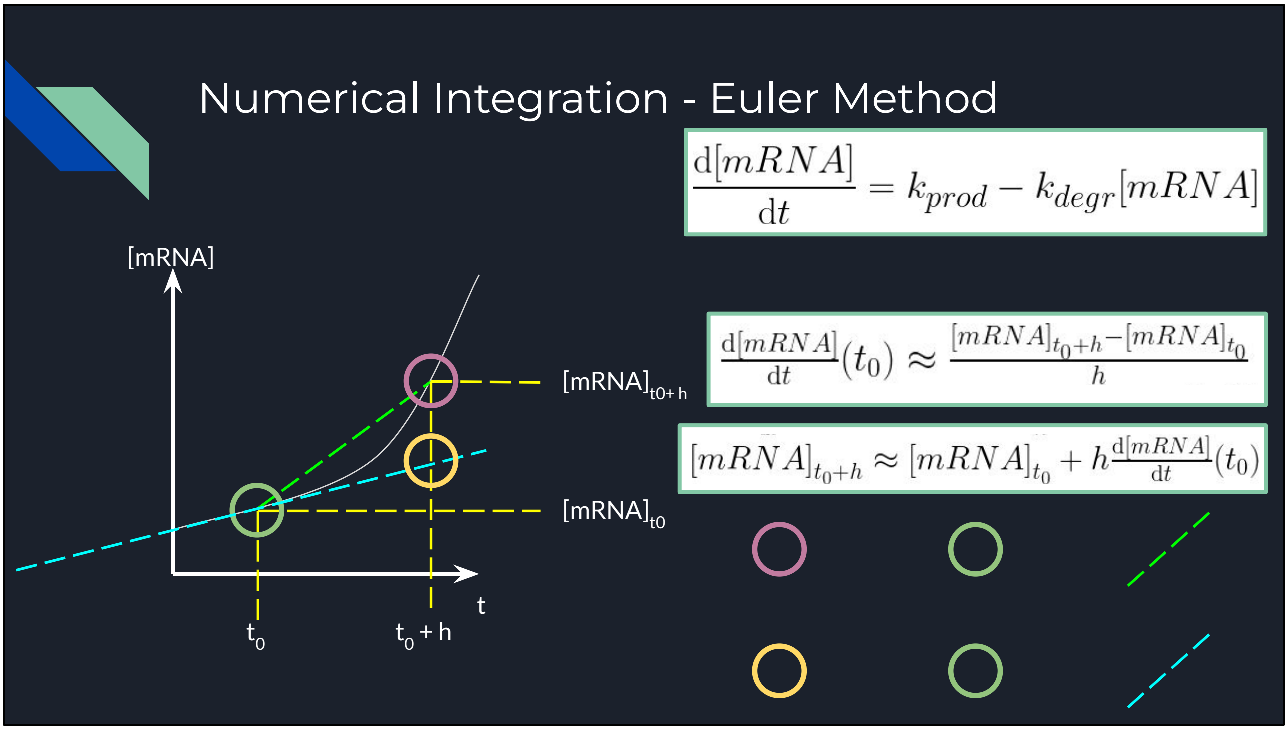
Numerical Integration - Euler Method

$$\frac{\mathrm{d}[mRNA]}{\mathrm{d}t} = k_{prod} - k_{degr}[mRNA]$$



$$\frac{\mathrm{d}[mRNA]}{\mathrm{d}t}(t_0) \approx \frac{[mRNA]_{t_0+h} - [mRNA]_{t_0}}{h}$$

$$[mRNA]_{t_0+h} \approx [mRNA]_{t_0} + h\frac{\mathrm{d}[mRNA]}{\mathrm{d}t}(t_0)$$



Numerical Integration - Euler Method

$$\frac{d[mRNA]}{dt} = k_{prod} - k_{degr}[mRNA]$$

$$\frac{d[mRNA]}{dt}(t_0) \approx \frac{[mRNA]_{t_0+h} - [mRNA]_{t_0}}{h}$$

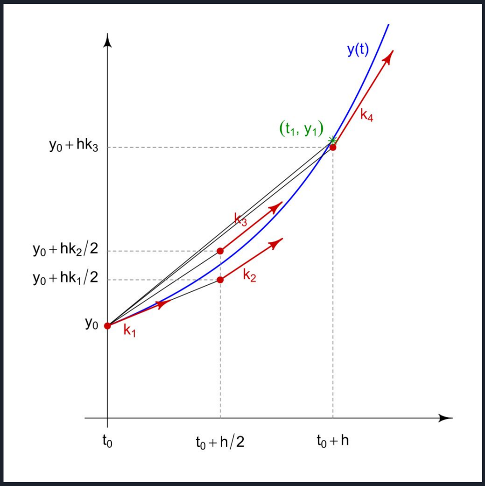
$$[mRNA]_{t_0+h} \approx [mRNA]_{t_0} + h\frac{d[mRNA]}{dt}(t_0)$$

If you’re calculating it near a point there the line changes a lot, then the approximation is less accurate

Numerical Integration - Runge-Kutta

SciPy default for numerical integration

RK45: Explicit Runge-Kutta method of order 5(4)



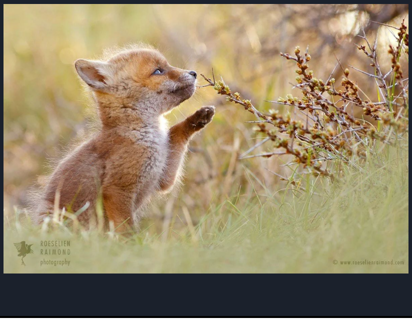
Lotka-Volterra Equations

x - Prey

$$\frac{dx}{dt} = \alpha x - \beta x y,$$

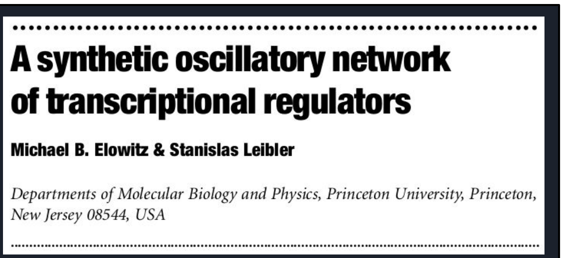
y - Predator

$$\frac{dy}{dt} = \delta x y - \gamma y,$$



DEMO!

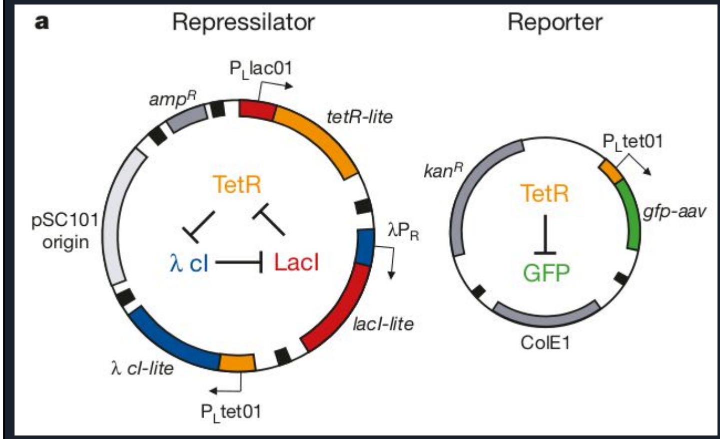
Repressilator

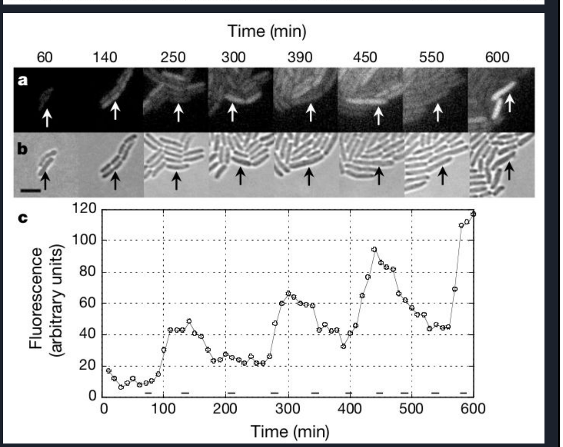


A synthetic oscillatory network of transcriptional regulators

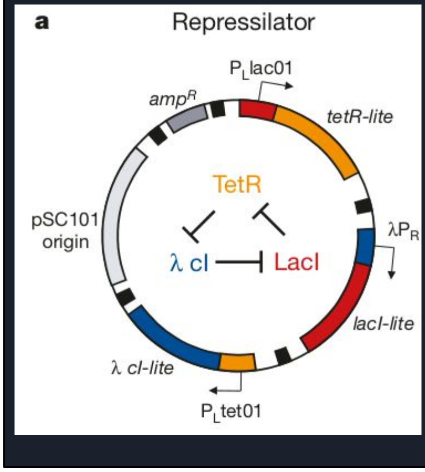
Michael B. Elowitz & Stanislas Leibler

Departments of Molecular Biology and Physics, Princeton University, Princeton, New Jersey 08544, USA





Repressilator



m - mRNA

$$\frac{dm_i}{dt} = -m_i + \frac{\alpha}{(1+p_j^n)} + \alpha_0$$

p - Protein

$$\frac{dp_i}{dt} = -\beta(p_i - m_i)$$

$$\begin{pmatrix} i = lacI, tetR, cI \\ j = cI, lacI, tetR \end{pmatrix}$$

DEMO!

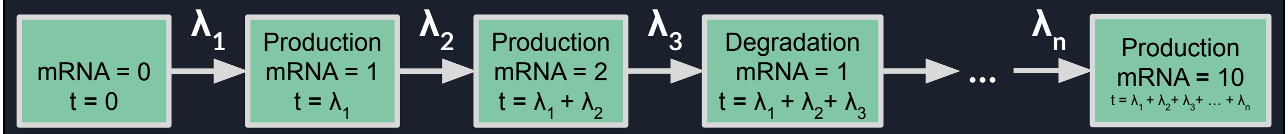
Downsides to ODE modelling

- Does not model biological noise
  - Assumes that we can accurately model based on “average behaviour”
- Continuous
  - “Atto-fox” problem (10^{-18})

Stochastic Models - Gillespie Algorithm

$$\varnothing \overset{k_{prod}}{\rightarrow} mRNA$$

$$mRNA \overset{k_{degr}}{\rightarrow} \varnothing$$



mRNA = 0
t = 0

λ_1

Production
mRNA = 1
$t = \lambda_1$

λ_2

Production
mRNA = 2
$t = \lambda_1 + \lambda_2$

λ_3

Degradation
mRNA = 1
$t = \lambda_1 + \lambda_2 + \lambda_3$

...

λ_n

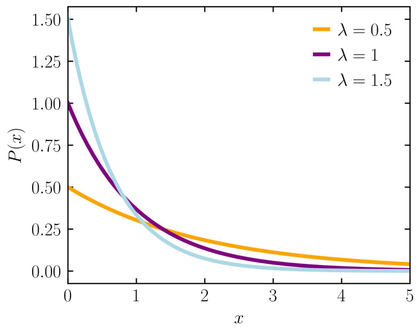
Production
mRNA = 10
$t = \lambda_1 + \lambda_2 + \lambda_3 + \ldots + \lambda_n$

At each step of the algorithm:

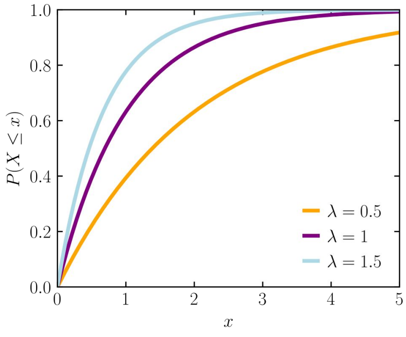
1) Determine the time since the last event
2) Determine what the event is and update the state

Exponential Distribution

Probability distribution of the time between events which occur continuously and independently at a constant average rate



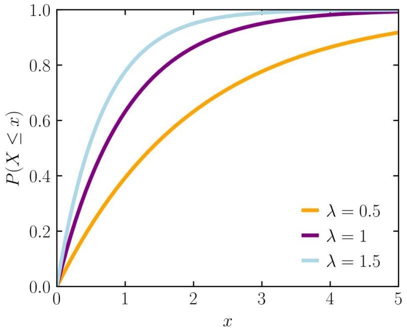
$$P(x) = \lambda e^{-\lambda x}$$



$$P(X \leq x) = 1 - e^{-\lambda x}$$

Sampling from an exponential distribution

1) Sample from a uniform distribution in the range [0, 1)
2) Use the quartile function (inverse of the cumulative distribution function)



$$P(X \leq x) = 1 - e^{-\lambda x}$$

$$p = 1 - e^{-\lambda x}$$

$$e^{-\lambda x} = 1 - p$$

$$-\lambda x = \ln(1 - p)$$

$$x = -\frac{\ln(1 - p)}{\lambda}$$

p is a uniform random number [0, 1)

Gillespie Algorithm - Worked Example

1) Determine the time since the last event
2) Determine what the event is and update the state

$$k_{prod} = 4$$

$$k_{degr} = 2$$

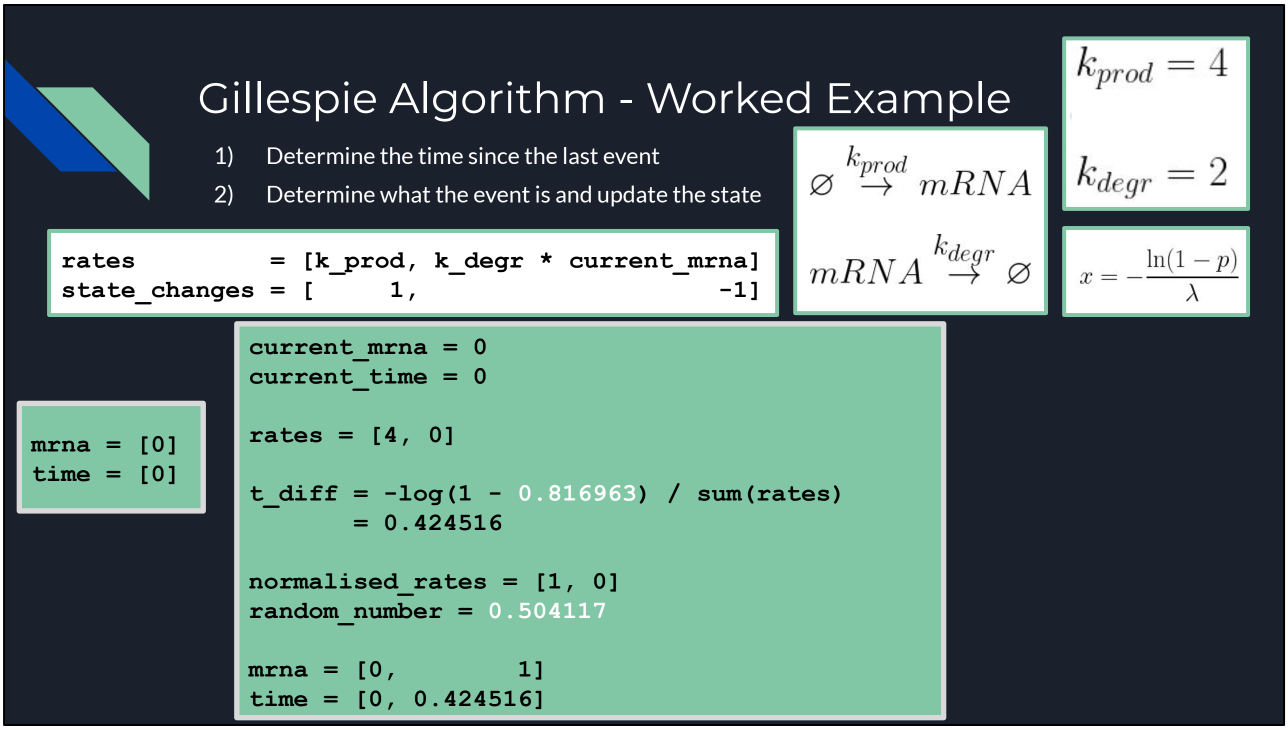
$$\varnothing \overset{k_{prod}}{\rightarrow} mRNA$$

$$mRNA \overset{k_{degr}}{\rightarrow} \varnothing$$

$$x = -\frac{\ln(1-p)}{\lambda}$$

```
rates         = [k_prod, k_degr * current_mrna]
state_changes = [     1,                     -1]
```

```
mrna = [0]
time = [0]
```



Gillespie Algorithm - Worked Example

1) Determine the time since the last event
2) Determine what the event is and update the state

$$\varnothing \overset{k_{prod}}{\rightarrow} mRNA$$

$$mRNA \overset{k_{degr}}{\rightarrow} \varnothing$$

$$k_{prod} = 4$$

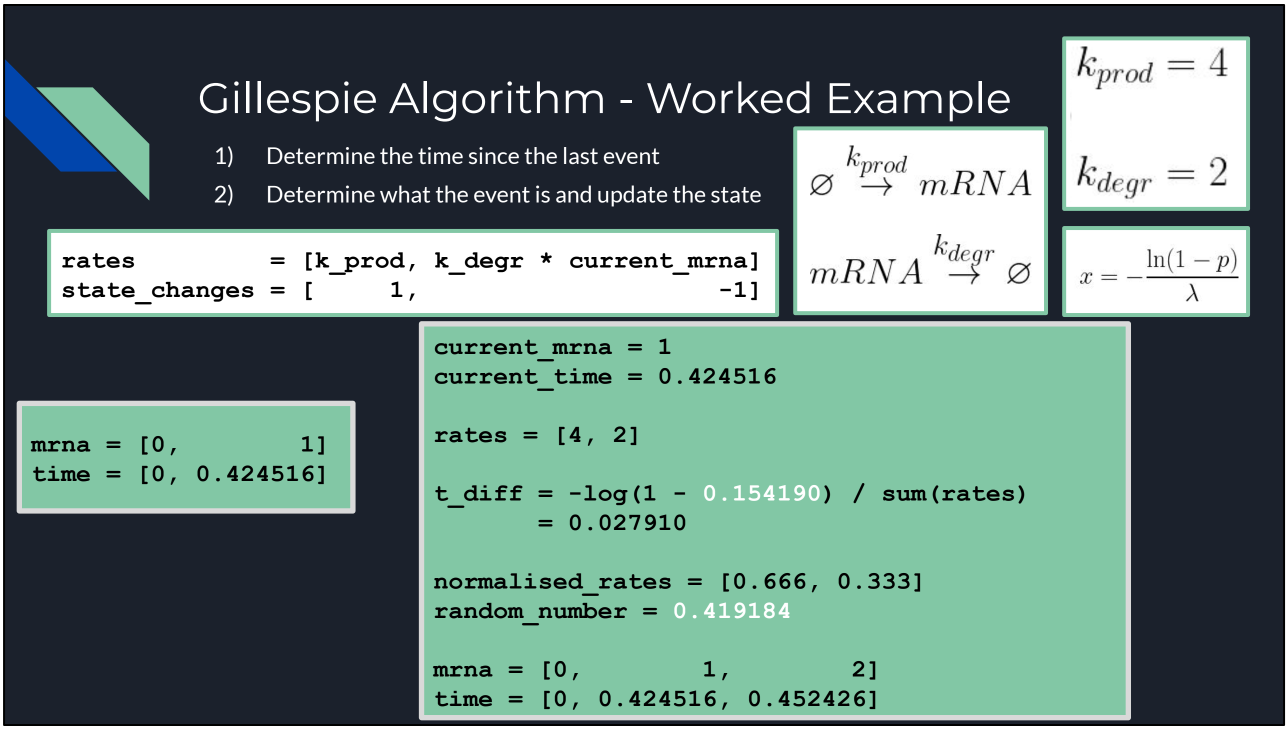
$$k_{degr} = 2$$

$$x = -\frac{\ln(1-p)}{\lambda}$$

```
rates         = [k_prod, k_degr * current_mrna]
state_changes = [     1,                     -1]
```

```
mrna = [0]
time = [0]
```

```
current_mrna = 0
current_time = 0

rates = [4, 0]

t_diff = -log(1 - 0.816963) / sum(rates)
       = 0.424516

normalised_rates = [1, 0]
random_number = 0.504117

mrna = [0,        1]
time = [0, 0.424516]
```



Gillespie Algorithm - Worked Example

1) Determine the time since the last event
2) Determine what the event is and update the state

$$k_{prod} = 4$$

$$k_{degr} = 2$$

$$\varnothing \overset{k_{prod}}{\rightarrow} mRNA$$

$$mRNA \overset{k_{degr}}{\rightarrow} \varnothing$$

$$x = -\frac{\ln(1-p)}{\lambda}$$

```
rates         = [k_prod, k_degr * current_mrna]
state_changes = [     1,                     -1]
```

```
mrna = [0,        1]
time = [0, 0.424516]
```

```
current_mrna = 1
current_time = 0.424516

rates = [4, 2]

t_diff = -log(1 - 0.154190) / sum(rates)
       = 0.027910

normalised_rates = [0.666, 0.333]
random_number = 0.419184

mrna = [0,        1,        2]
time = [0, 0.424516, 0.452426]
```

Gillespie Algorithm - Worked Example

1) Determine the time since the last event
2) Determine what the event is and update the state

$$\varnothing \overset{k_{prod}}{\rightarrow} mRNA$$

$$mRNA \overset{k_{degr}}{\rightarrow} \varnothing$$

$$k_{prod} = 4$$

$$k_{degr} = 2$$

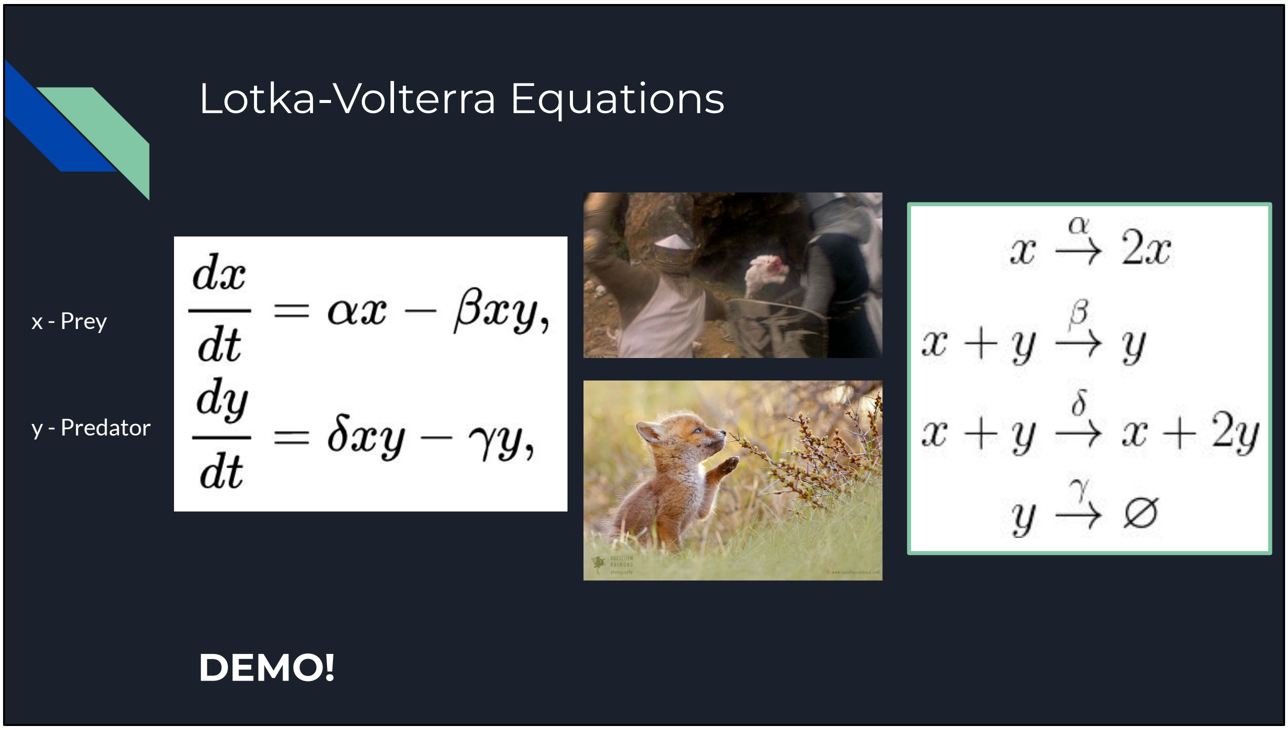
$$x = -\frac{\ln(1-p)}{\lambda}$$

```
rates         = [k_prod, k_degr * current_mrna]
state_changes = [     1,                    -1]
```

```
mrna = [0,        1,        2]
time = [0, 0.424516, 0.452426]
```

```
current_mrna = 2
current_time = 0.452426

rates = [4, 4]

t_diff = -log(1 - 0.166574) / sum(rates)
       = 0.022776

normalised_rates = [0.5, 0.5]
random_number = 0.766496

mrna = [0,        1,        2,        1]
time = [0, 0.424516, 0.452426, 0.475202]
```

DEMO!



Lotka-Volterra Equations

x - Prey

y - Predator

$$\frac{dx}{dt} = \alpha x - \beta x y,$$
$$\frac{dy}{dt} = \delta x y - \gamma y,$$

$$x \xrightarrow{\alpha} 2x$$
$$x + y \xrightarrow{\beta} y$$
$$x + y \xrightarrow{\delta} x + 2y$$
$$y \xrightarrow{\gamma} \varnothing$$

DEMO!

Now, there are two unstable states - extinction of prey, leading to extinction of predators, or extinction of predators, leading to prey population explosion
Potentially interesting that a prey spike often leads to complete extinction

Repressilator

m - mRNA

p - Protein

$$\frac{dm_i}{dt} = -m_i + \frac{\alpha}{(1+p_j^n)} + \alpha_0$$

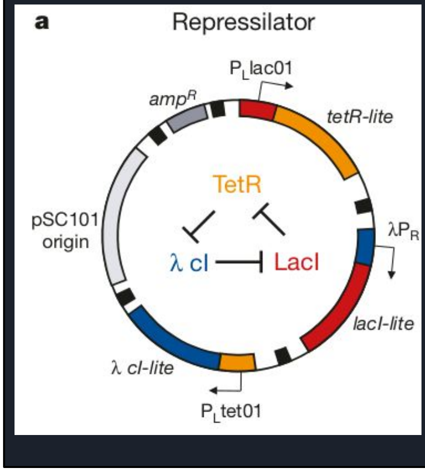
$$\frac{dp_i}{dt} = -\beta(p_i - m_i) \qquad \begin{pmatrix} i = lacI, tetR, cI \\ j = cI, lacI, tetR \end{pmatrix}$$

$$\varnothing \xrightarrow{\alpha_0 + \frac{\alpha}{1+p_j{}^n}} m_i$$

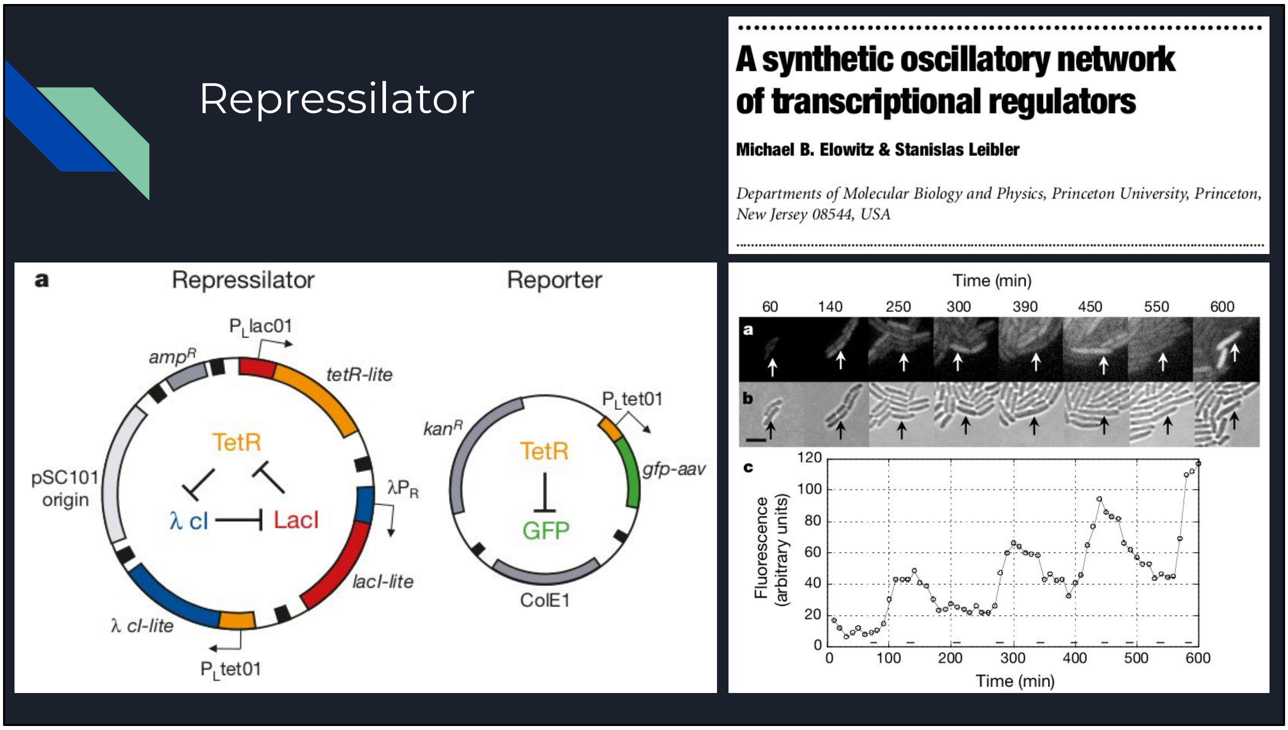
$$m_i \rightarrow \varnothing$$

$$m_i \xrightarrow{\beta} m_i + p_i$$

$$p_i \xrightarrow{\beta} \varnothing$$



DEMO!



Repressilator

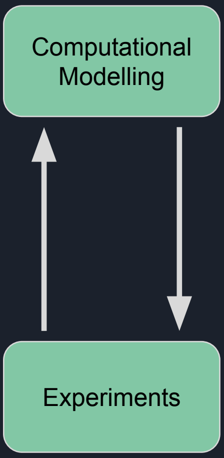
A synthetic oscillatory network of transcriptional regulators

Michael B. Elowitz & Stanislas Leibler

Departments of Molecular Biology and Physics, Princeton University, Princeton, New Jersey 08544, USA

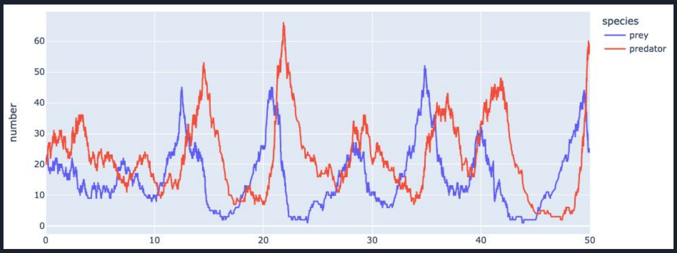
The stochastic models are better able to model the observed fluorescence

Computational modelling should be iterative



Computational Modelling

Experiments



Let’s revisit the Lotka-Volterra equations…

Lotka-Volterra Equations

x - Prey

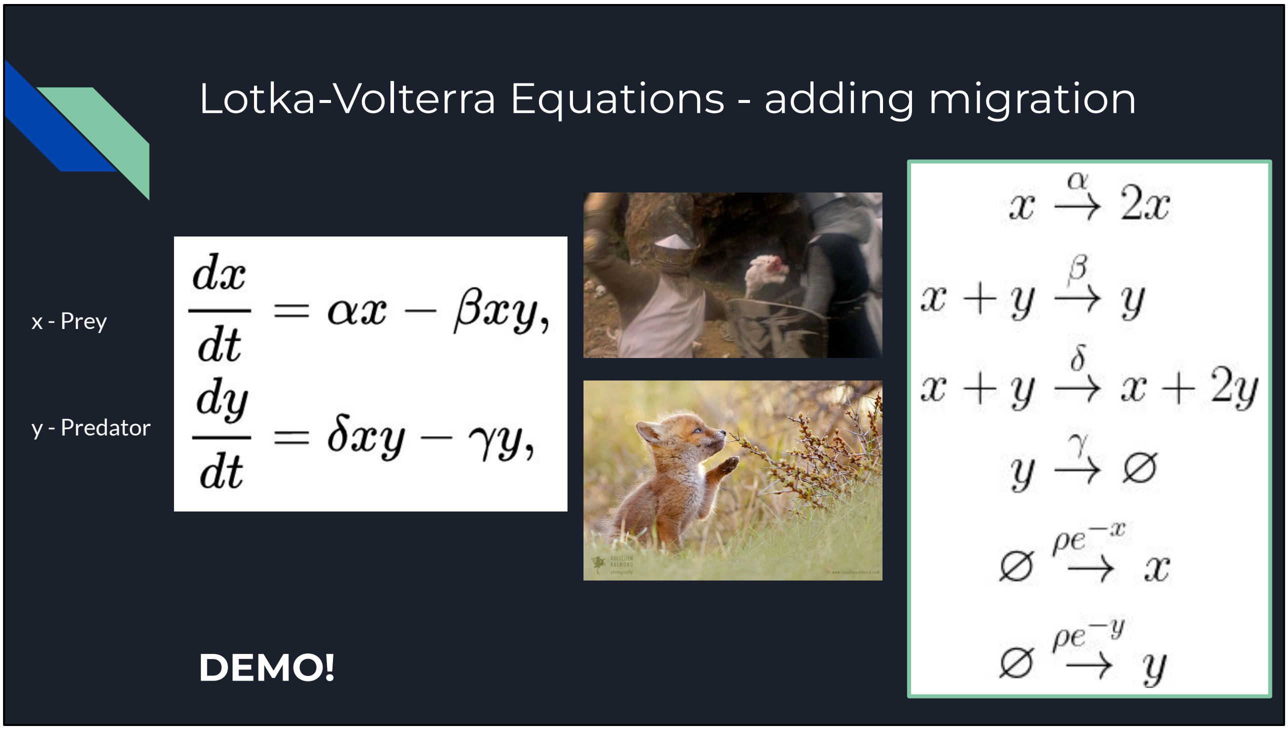
y - Predator

$$\frac{dx}{dt} = \alpha x - \beta xy,$$
$$\frac{dy}{dt} = \delta xy - \gamma y,$$

$$x \xrightarrow{\alpha} 2x$$
$$x + y \xrightarrow{\beta} y$$
$$x + y \xrightarrow{\delta} x + 2y$$
$$y \xrightarrow{\gamma} \varnothing$$

DEMO!

How would we try and fix the problems with this model, namely, really common extinction events



Lotka-Volterra Equations - adding migration

x - Prey

y - Predator

$$\frac{dx}{dt} = \alpha x - \beta x y,$$
$$\frac{dy}{dt} = \delta x y - \gamma y,$$

$$x \xrightarrow{\alpha} 2x$$
$$x + y \xrightarrow{\beta} y$$
$$x + y \xrightarrow{\delta} x + 2y$$
$$y \xrightarrow{\gamma} \varnothing$$
$$\varnothing \xrightarrow{\rho e^{-x}} x$$
$$\varnothing \xrightarrow{\rho e^{-y}} y$$

DEMO!

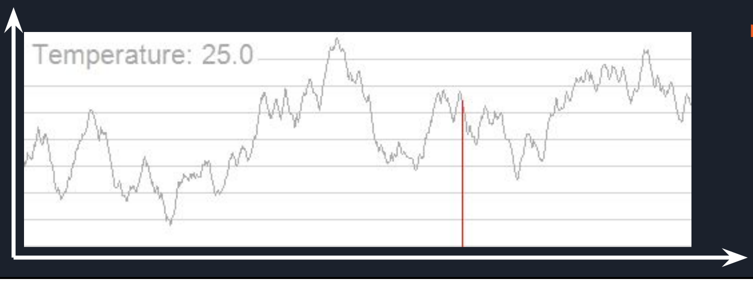
Let’s add terms that represent migration
The e^{-x} term ensures that when x is 0, migration is highest, and when x is large, migration is low

Downsides to stochastic modelling

- Computationally intensive
- Above a certain number of “molecules” you’re better off with continuous modelling approaches

Model parameters

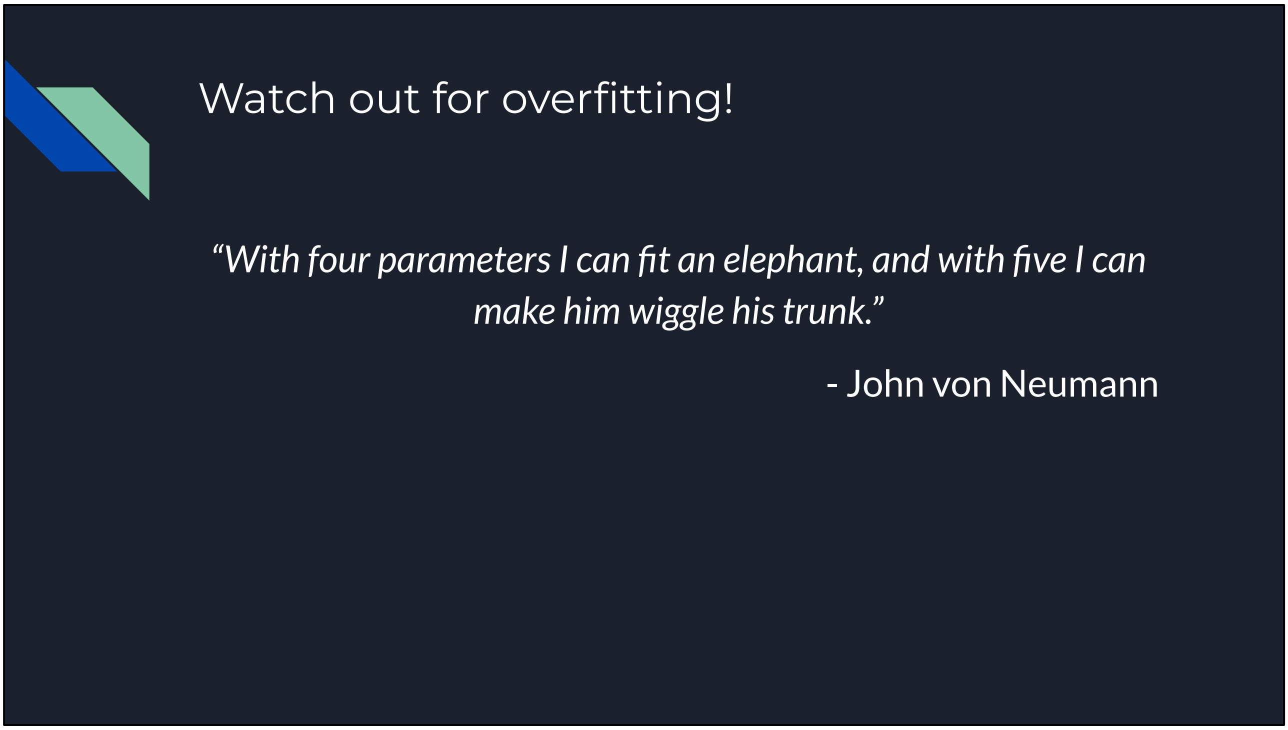
- Measure
  - e.g. block transcription and measure the decay rate of mRNA
- Look up to see if someone else has measured it!
- Fit models to experimental data
  - e.g. Simulated annealing



Model fit to experimental data

Nearly always multidimensional!

Parameter space

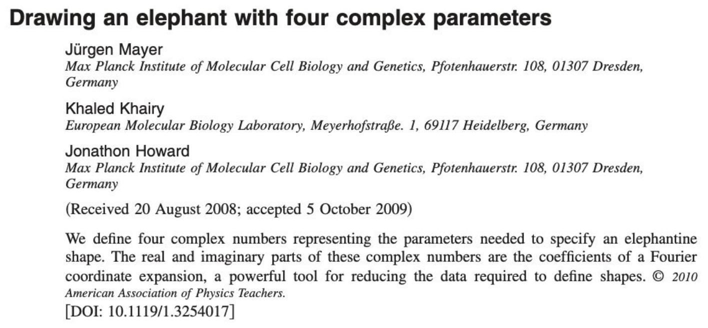


Watch out for overfitting!

> *“With four parameters I can fit an elephant, and with five I can make him wiggle his trunk.”*
>
> - John von Neumann

John von Neumann was a polymath engineer, computer scientist, mathematician and physicist.
What’s meant by this quote is that the more complex the model, the more it will be able to fit the data.

Watch out for overfitting!



Drawing an elephant with four complex parameters

Jürgen Mayer
Max Planck Institute of Molecular Cell Biology and Genetics, Pfotenhauerstr. 108, 01307 Dresden, Germany

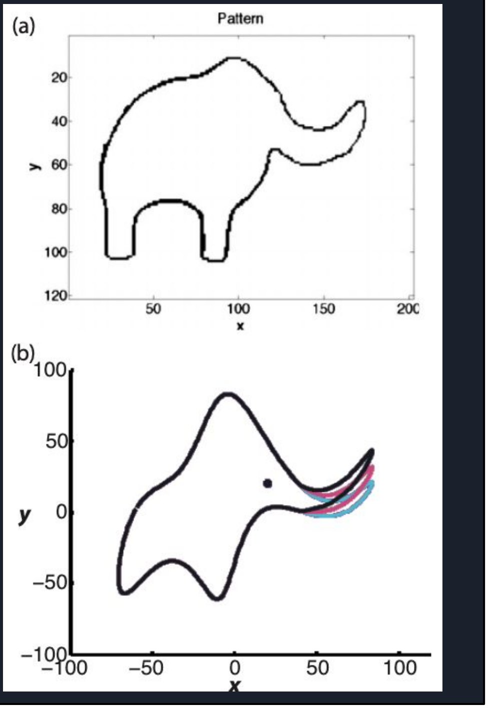
Khaled Khairy
European Molecular Biology Laboratory, Meyerhofstraße. 1, 69117 Heidelberg, Germany

Jonathon Howard
Max Planck Institute of Molecular Cell Biology and Genetics, Pfotenhauerstr. 108, 01307 Dresden, Germany

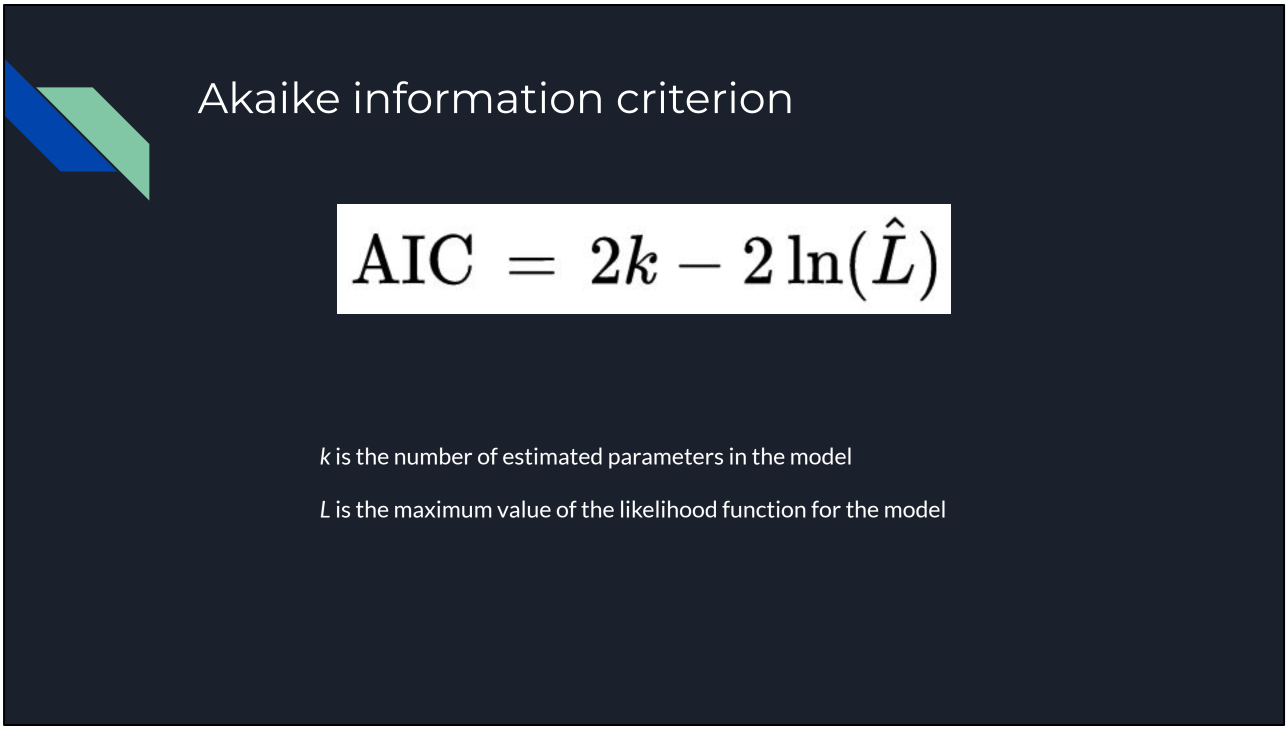
(Received 20 August 2008; accepted 5 October 2009)

We define four complex numbers representing the parameters needed to specify an elephantine shape. The real and imaginary parts of these complex numbers are the coefficients of a Fourier coordinate expansion, a powerful tool for reducing the data required to define shapes.

[DOI: 10.1119/1.3254017]



And of course, someone *has* drawn an elephant with 4 parameters and made it wiggle its trunk with a 5th



Akaike information criterion

$$\mathrm{AIC} = 2k - 2\ln(\hat{L})$$

k is the number of estimated parameters in the model

L is the maximum value of the likelihood function for the model

The AIC, and related measures, can be used to score models taking into account the number of parameters

Summary

- Ordinary differential equation (ODE) modelling
  - Analytical solutions are rare, so numerical integration is needed
  - Deterministic - doesn't take into account biological noise
  - Continuous
- Stochastic models
  - Gillespie algorithm can be used to simulate
  - Computationally intensive
  - Best used where biological noise is important
- Computational modelling is best done in an iterative cycle
- Model parameters can be measured directly, or estimated based on fitting
- Be careful not to overfit models to data!
- Akaike information criterion (among others) can be used to measure whether the complexity of a model is "worth it"
- https://github.com/marc-jones/modelling-lecture