A ~~PENGUIN~~ PRACTICAL INTRODUCTION TO DIFFERENTIAL EQUATIONS

Cristoffer Cordes and Lars Ole Schwen

MEVIS Academy, 2015-05-22

A ~~PENGUIN~~ PRACTICAL INTRODUCTION TO DIFFERENTIAL EQUATIONS

1. Motivation
2. Freezing Fish: A Very Simple ODE
3. Lonely Penguins on an Ice Floe: Step Size Adaption
4. Penguins and Fish: A Predator–Prey Equation
5. Swimming Baby Penguin: A Simple ODE
6. Travelling Penguins: Understanding a Given ODE
7. Summary

Disclaimer

This talk **will** be about

- ✔ modeling physical processes by ordinary differential equations (ODEs)
- ✔ implementing and numerically solving ODEs in python
- ✔ understanding a given ODE

Disclaimer

This talk **will** be about

- ✔ modeling physical processes by ordinary differential equations (ODEs)
- ✔ implementing and numerically solving ODEs in python
- ✔ understanding a given ODE

This talk will **not** be about

- ✘ theory and mathematical details

Disclaimer

This talk **will** be about

✔ modeling physical processes by ordinary differential equations (ODEs)

✔ implementing and numerically solving ODEs in python

✔ understanding a given ODE

This talk will **not** be about

✘ theory and mathematical details

✔ no animals were harmed in the making of this presentation

✘ this talk may contain biological nonsense

✘ this talk discriminates against fish and is not BfE approved

1. Motivation

Ordinary Differential Equations

- time-dependent processes
- no space dependency

1. Motivation

Ordinary Differential Equations

- time-dependent processes
- no space dependency

Mathematical Description

$$y$$

- some unknown function y

1. Motivation

Ordinary Differential Equations

- time-dependent processes
- no space dependency

Mathematical Description

$$\frac{\mathrm{d}}{\mathrm{d}\,t}y = \mathrm{d}_t y = \dot{y}$$

- some unknown function y
- varies over time

1. Motivation

Ordinary Differential Equations

- time-dependent processes
- no space dependency

Mathematical Description

$$\frac{\mathrm{d}}{\mathrm{d}\,t}y = \mathrm{d}_t y = \dot{y} = \text{some right hand side } f$$

- some unknown function y
- varies over time
- depending on something

1. Motivation

Solving ODEs

Analytically

- pen and paper
- computer algebra system

1. Motivation
Solving ODEs

Analytically

- pen and paper
- computer algebra system

Numerically

- various iterative schemes
- own implementation
- existing solvers
  - scipy
  - sundials (C++, contains cvode)
  - odeint (part of boost)
  - gsl (C++; GPL!)

2. Freezing Fish: A Very Simple ODE

Problem

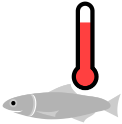
- Humboldt penguins *(Eudyptes chrysocome)* have caught fish at 5°C

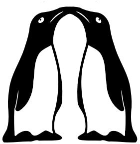
https://upload.wikimedia.org/wikipedia/commons/d/d8/Penguincotswoldwildlifepark.jpg

Drcwp1, public domain

2. Freezing Fish: A Very Simple ODE

Problem



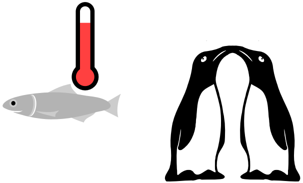


- Humboldt penguins *(Eudyptes chrysocome)* have caught fish at 5°C
- are currently busy

https://upload.wikimedia.org/wikipedia/commons/d/d8/Penguincotswoldwildlifepark.jpg
Drcwp1, public domain

2. Freezing Fish: A Very Simple ODE

Problem



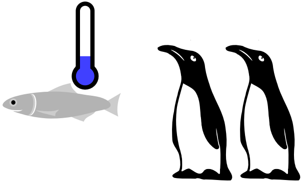
- Humboldt penguins *(Eudyptes chrysocome)* have caught fish at 5°C
- are currently busy
- ambient temperature 0°C

https://upload.wikimedia.org/wikipedia/commons/d/d8/Penguincotswoldwildlifepark.jpg

Drcwp1, public domain

2. Freezing Fish: A Very Simple ODE

Problem



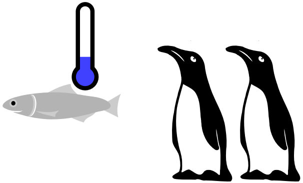
- Humboldt penguins *(Eudyptes chrysocome)* have caught fish at 5°C
- are currently busy
- ambient temperature 0°C
- want to describe cooling

https://upload.wikimedia.org/wikipedia/commons/d/d8/Penguincotswoldwildlifepark.jpg

Drcwp1, public domain

2. Freezing Fish: A Very Simple ODE

Problem



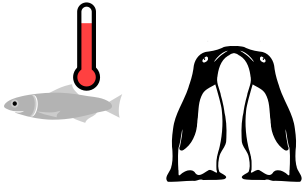
- Humboldt penguins *(Eudyptes chrysocome)* have caught fish at 5°C
- are currently busy
- ambient temperature 0°C
- want to describe cooling
- simplification: fish is just a point

https://upload.wikimedia.org/wikipedia/commons/d/d8/Penguincotswoldwildlifepark.jpg
Drcwp1, public domain

2. Freezing Fish: A Very Simple ODE

Model

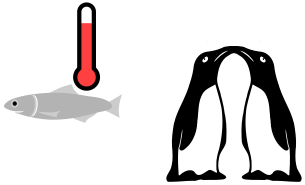
- [temp. decrease] $\sim$ [fish temp.] – [ambient temp.]



2. Freezing Fish: A Very Simple ODE

Model

- [temp. decrease] $\sim$ [fish temp.] $-$ [ambient temp.]
- proportionality constant $a = 0.01$ Kelvin per second

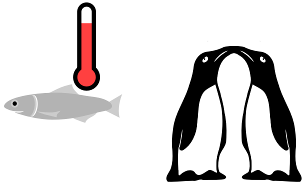


2. Freezing Fish: A Very Simple ODE

Model

- [temp. decrease] $\sim$ [fish temp.] $-$ [ambient temp.]
- proportionality constant $a = 0.01$ Kelvin per second

$$\dot{y} = -a\,(y - 0°\mathrm{C}) \qquad (1)$$



2. Freezing Fish: A Very Simple ODE

Implementation I

Setup

```python
import numpy as np
from scipy.integrate import ode
```

2. Freezing Fish: A Very Simple ODE

Implementation I

Setup

```
import numpy as np
from scipy.integrate import ode
```

Define derivative

```
a = -0.1

def rhs(t, y):
  return y*(a-0.0)
```

2. Freezing Fish: A Very Simple ODE

Implementation II

Solve

```
integrator = ode(rhs)
integrator.set_initial_value(y=y0, t=t0)
temperature = []
while integrator.t < tend:
    integrator.integrate(t=tend, step=True)
    temperature.append([integrator.t, integrator.y])
temperature = np.array(temperature)
```

2. Freezing Fish: A Very Simple ODE

Demo

2. Freezing Fish: A Very Simple ODE

Demo

Lessons learned

- ✔ ordinary differential equations don't bite ☺
- ✔ scipy offers easy use

2. Freezing Fish: A Very Simple ODE

Demo

Lessons learned

✔ ordinary differential equations don't bite ☺

✔ scipy offers easy use

✘ `step` unclear at this point

2. Freezing Fish: A Very Simple ODE

Demo

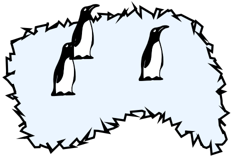
Lessons learned

- ✔ ordinary differential equations don't bite ☺
- ✔ scipy offers easy use
- ✘ `step` unclear at this point

- ✔ ipython notebook nice tool

3. Lonely Penguins on an Ice Floe: Step Size Adaption

Problem

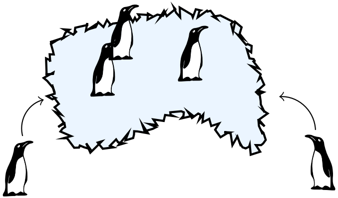


- gentoo penguin
  (Pygoscelis papua)

https://upload.wikimedia.org/wikipedia/commons/0/04/Pygoscelis_papua_-Jougla_Point,_Wiencke_Island,_Palmer_Archipelago_-adults_and_chicks-8.jpg

3. Lonely Penguins on an Ice Floe: Step Size Adaption

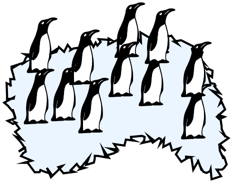
Problem



- gentoo penguin
  (Pygoscelis papua)
- cries for company if it is cold

3. Lonely Penguins on an Ice Floe: Step Size Adaption

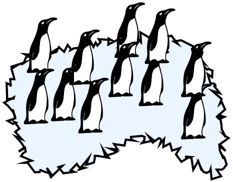
Problem



- gentoo penguin
  (Pygoscelis papua)
- cries for company if it is cold
- more penguins feel less cold

3. Lonely Penguins on an Ice Floe: Step Size Adaption

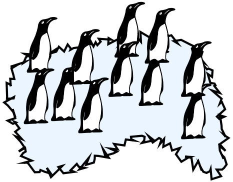
Problem



- gentoo penguin
  (Pygoscelis papua)
- cries for company if it is cold
- more penguins feel less cold
- crying becomes more quiet

3. Lonely Penguins on an Ice Floe: Step Size Adaption

Problem



- gentoo penguin
  (Pygoscelis papua)
- cries for company if it is cold
- more penguins feel less cold
- crying becomes more quiet
- attracts less additional penguins

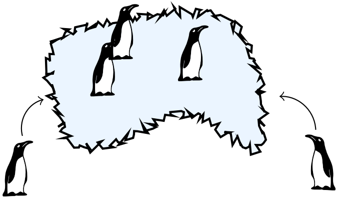
https://upload.wikimedia.org/wikipedia/commons/0/04/Pygoscelis_papua_-Jougla_Point,_Wiencke_Island,_Palmer_Archipelago_-adults_and_chicks-8.jpg

3. Lonely Penguins on an Ice Floe: Step Size Adaption

Model

$$\dot{y} = \alpha\,(y - 2)$$

- preferred penguin density 2
- speed of attraction $\alpha = -4$
- initially lower density $y(0) = 1$



3. Lonely Penguins on an Ice Floe: Step Size Adaption

Forward Euler

In addition to scipy's ode solver, we implement explicit (forward) Euler

https://upload.wikimedia.org/wikipedia/commons/b/b9/Euler-10_Swiss_Franc_banknote_(front).jpg (adapted)

no copyright

3. Lonely Penguins on an Ice Floe: Step Size Adaption

Forward Euler

In addition to scipy's ode solver, we implement explicit (forward) Euler

$$f(y) = \dot{y}(t) \approx \frac{y(t+\Delta t) - y(t)}{\Delta t}$$

https://upload.wikimedia.org/wikipedia/commons/b/b9/Euler-10_Swiss_Franc_banknote_(front).jpg (adapted)

no copyright

3. Lonely Penguins on an Ice Floe: Step Size Adaption

Forward Euler

In addition to scipy's ode solver, we implement explicit (forward) Euler

$$f(y) = \dot{y}(t) \approx \frac{y(t+\Delta t) - y(t)}{\Delta t}$$

$$\rightsquigarrow \qquad f(Y^k) = \frac{Y^{k+1} - Y^k}{\Delta t}$$

https://upload.wikimedia.org/wikipedia/commons/b/b9/Euler-10_Swiss_Franc_banknote_(front).jpg (adapted)

no copyright

3. Lonely Penguins on an Ice Floe: Step Size Adaption

Forward Euler

In addition to scipy's ode solver, we implement explicit (forward) Euler

$$
\begin{aligned}
& & f(y) = \dot{y}(t) \approx \frac{y(t+\Delta t) - y(t)}{\Delta t} \\
&\rightsquigarrow & f(Y^k) = \frac{Y^{k+1} - Y^k}{\Delta t} \\
&\Rightarrow & Y^{k+1} = Y^k + \Delta t \cdot f(Y^k)
\end{aligned}
$$

https://upload.wikimedia.org/wikipedia/commons/b/b9/Euler-10_Swiss_Franc_banknote_(front).jpg (adapted)

no copyright

3. Lonely Penguins on an Ice Floe: Step Size Adaption

Implementation

```python
def explicitEuler(Y0,T0,Dt,Tend):
    y = Y0
    t = T0
    temperature = []
    while t < Tend:
        y += Dt * rhs(t,y)
        t += Dt
        temperature.append([t, y])
    return(np.array(temperature))
```

3. Lonely Penguins on an Ice Floe: Step Size Adaption

Demo

3. Lonely Penguins on an Ice Floe: Step Size Adaption

Demo

Lessons learned

✔ simple schemes easy to implement

3. Lonely Penguins on an Ice Floe: Step Size Adaption

Demo

Lessons learned

✔ simple schemes easy to implement

✘ but may fail for incorrect parameters

3. Lonely Penguins on an Ice Floe: Step Size Adaption

Demo

Lessons learned

- ✔ simple schemes easy to implement
- ✘ but may fail for incorrect parameters
- more complex schemes more difficult to implement

3. Lonely Penguins on an Ice Floe: Step Size Adaption

More Advanced Methods

- backwards Euler $Y^{k+1} = Y^k + \Delta t \cdot f(Y^{k+1})$ (implicit)

3. Lonely Penguins on an Ice Floe: Step Size Adaption

More Advanced Methods

- backwards Euler $Y^{k+1} = Y^k + \Delta t \cdot f(Y^{k+1})$ (implicit)
- Runge–Kutta methods (explicit multi-stage; implicit)

3. Lonely Penguins on an Ice Floe: Step Size Adaption

More Advanced Methods

- backwards Euler $Y^{k+1} = Y^k + \Delta t \cdot f(Y^{k+1})$ (implicit)
- Runge–Kutta methods (explicit multi-stage; implicit)
- e.g., Runge–Kutta–Fehlberg 4th/5th order adaptive

3. Lonely Penguins on an Ice Floe: Step Size Adaption

More Advanced Methods

- backwards Euler $Y^{k+1} = Y^k + \Delta t \cdot f(Y^{k+1})$ (implicit)
- Runge–Kutta methods (explicit multi-stage; implicit)
- e.g., Runge–Kutta–Fehlberg 4th/5th order adaptive
- Rosenbrock methods

3. Lonely Penguins on an Ice Floe: Step Size Adaption

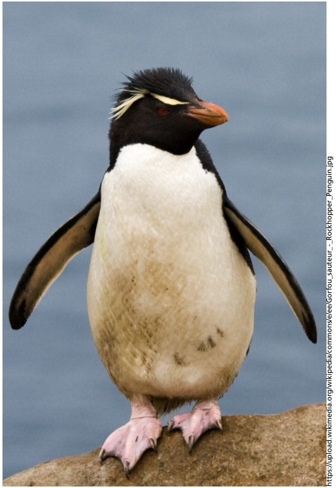
More Advanced Methods

- backwards Euler $Y^{k+1} = Y^k + \Delta t \cdot f(Y^{k+1})$ (implicit)
- Runge–Kutta methods (explicit multi-stage; implicit)
- e.g., Runge–Kutta–Fehlberg 4th/5th order adaptive
- Rosenbrock methods
- backwards differentiation formulas (BDF) methods for “stiff” problems

3. Lonely Penguins on an Ice Floe: Step Size Adaption

More Advanced Methods

- backwards Euler $Y^{k+1} = Y^k + \Delta t \cdot f(Y^{k+1})$ (implicit)
- Runge–Kutta methods (explicit multi-stage; implicit)
- e.g., Runge–Kutta–Fehlberg 4th/5th order adaptive
- Rosenbrock methods
- backwards differentiation formulas (BDF) methods for “stiff” problems



https://upload.wikimedia.org/wikipedia/commons/e/ee/Gorfou_sauteur_-_Rockhopper_Penguin.jpg

Samuel Blanc, CC BY-SA 3.0

4. Penguins and Fish: A Predator–Prey Equation

Problem

- little penguin
  (Eudyptula minor)
  needs fish to be happy

https://upload.wikimedia.org/wikipedia/commons/f/f5/Little_Penguin_Feb09.jpg
Fir0002/Flagstaffotos, CC BY-NC 3.0

4. Penguins and Fish: A Predator–Prey Equation

Problem

- little penguin
  (Eudyptula minor)
  needs fish to be happy
- the more fish, the more penguins are attracted

4. Penguins and Fish: A Predator–Prey Equation

Problem

- little penguin
  (Eudyptula minor)
  needs fish to be happy
- the more fish, the more penguins are attracted
- the more penguins, the more fish is eaten
  (i.e., less survives)

https://upload.wikimedia.org/wikipedia/commons/f/f5/Little_Penguin_Feb09.jpg
Fir0002/Flagstaffotos, CC BY-NC 3.0

4. Penguins and Fish: A Predator–Prey Equation

Model interpretation

Volterra–Lotka model

$$\begin{aligned}\dot{F}(P,F) &= \alpha F - \beta PF\\ \dot{P}(P,F) &= -\gamma P + \delta PF\end{aligned}$$

4. Penguins and Fish: A Predator–Prey Equation

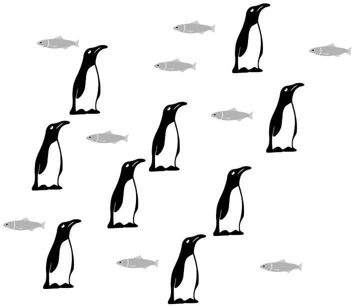
Model interpretation

Volterra–Lotka model

$$\begin{aligned}\dot{F}(P,F) &= \quad \alpha F - \beta PF \\ \dot{P}(P,F) &= -\gamma P + \delta PF\end{aligned}$$

What do βPF and δPF mean?

- the populations are *well-stirred*, PF is proportional to the *interaction* of the populations



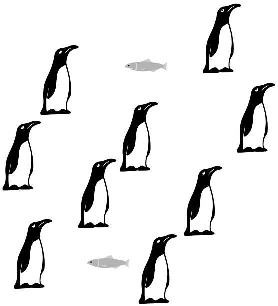
Model interpretation

Volterra–Lotka model

$$\dot{F}(P,F) = \alpha F - \beta PF$$
$$\dot{P}(P,F) = -\gamma P + \delta PF$$

What do βPF and δPF mean?

- the populations are *well-stirred*, PF is proportional to the *interaction* of the populations
- less fish $\Rightarrow$ less interaction



4. Penguins and Fish: A Predator–Prey Equation

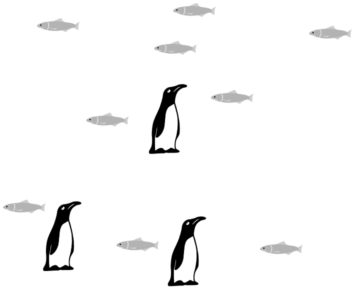
Model interpretation

Volterra–Lotka model

$$\begin{aligned}\dot{F}(P,F) &= \quad \alpha F - \beta PF \\ \dot{P}(P,F) &= -\gamma P + \delta PF\end{aligned}$$

What do βPF and δPF mean?

- the populations are *well-stirred*, PF is proportional to the *interaction* of the populations
- less fish $\Rightarrow$ less interaction
- less penguins $\Rightarrow$ less interaction



4. Penguins and Fish: A Predator–Prey Equation

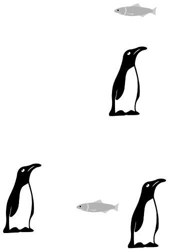
Model interpretation

Volterra–Lotka model

$$\begin{aligned}\dot{F}(P,F) &= \quad \alpha F - \beta PF \\ \dot{P}(P,F) &= -\gamma P + \delta PF\end{aligned}$$

What do βPF and δPF mean?

- the populations are *well-stirred*, PF is proportional to the *interaction* of the populations
- less fish $\Rightarrow$ less interaction
- less penguins $\Rightarrow$ less interaction
- less penguins and fish $\Rightarrow$ even less interaction

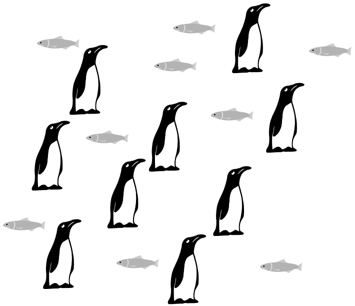


4. Penguins and Fish: A Predator–Prey Equation

Model interpretation

Volterra–Lotka model

$$\dot{F}(P,F) = \quad \alpha F - \beta PF$$
$$\dot{P}(P,F) = -\gamma P + \delta PF$$



What do βPF and δPF mean?

- the populations are *well-stirred*, PF is proportional to the *interaction* of the populations
- less fish $\Rightarrow$ less interaction
- less penguins $\Rightarrow$ less interaction
- less penguins and fish $\Rightarrow$ even less interaction
- β and δ determine how populations change proportional to the interactions

4. Penguins and Fish: A Predator–Prey Equation

Model interpretation

Volterra–Lotka model

$$\begin{aligned}\dot{F}(P,F) &= \alpha F - \beta PF\\ \dot{P}(P,F) &= -\gamma P + \delta PF\end{aligned}$$

What do βPF and δPF mean?

- the populations are *well-stirred*, PF is proportional to the *interaction* of the populations
- less fish $\Rightarrow$ less interaction
- less penguins $\Rightarrow$ less interaction
- less penguins and fish $\Rightarrow$ even less interaction
- β and δ determine how populations change proportional to the interactions

4. Penguins and Fish: A Predator–Prey Equation

Model interpretation

Volterra–Lotka model

$$\dot{F}(P,F) = \quad \alpha F - \beta PF$$
$$\dot{P}(P,F) = -\gamma P + \delta PF$$

What do βPF and δPF mean?

- the populations are *well-stirred*, PF is proportional to the *interaction* of the populations
- less fish $\Rightarrow$ less interaction
- less penguins $\Rightarrow$ less interaction
- less penguins and fish $\Rightarrow$ even less interaction
- β and δ determine how populations change proportional to the interactions

What do αF and γP mean?

- exponential population growth/decay
- independent of interaction

4. Penguins and Fish: A Predator–Prey Equation

Model

Volterra–Lotka model

$$\begin{aligned}\dot{F}(P,F) &= \alpha F - \beta PF \\ \dot{P}(P,F) &= -\gamma P + \delta PF\end{aligned}$$

4. Penguins and Fish: A Predator–Prey Equation

Model

Volterra–Lotka model

$$\dot{F}(P, F) = \alpha F - \beta P F$$
$$\dot{P}(P, F) = -\gamma P + \delta P F$$

with example parameters

$$\alpha = 1$$
$$\beta = 1$$
$$\gamma = 3$$
$$\delta = 1$$

4. Penguins and Fish: A Predator–Prey Equation

Model

Volterra–Lotka model

$$\begin{aligned}\dot{F}(P,F) &= \alpha F - \beta PF\\ \dot{P}(P,F) &= -\gamma P + \delta PF\end{aligned}$$

with example parameters

$$\begin{aligned}\alpha &= 1\\ \beta &= 1\\ \gamma &= 3\\ \delta &= 1\end{aligned}$$

and initial fish and penguin numbers

$$\begin{aligned}F_0 &= 5\\ P_0 &= 5\end{aligned}$$

4. Penguins and Fish: A Predator–Prey Equation

Model

Volterra–Lotka model

$$\dot{F}(P,F) = \alpha F - \beta PF$$
$$\dot{P}(P,F) = -\gamma P + \delta PF$$

with example parameters

$$\alpha = 1$$
$$\beta = 1$$
$$\gamma = 3$$
$$\delta = 1$$

and initial fish and penguin numbers

$$F_0 = 5$$
$$P_0 = 5$$

Units

$$[F] = \text{fish}$$
$$[P] = \text{penguins}$$

4. Penguins and Fish: A Predator–Prey Equation

Model

Volterra–Lotka model

$$\dot{F}(P,F) = \quad \alpha F - \beta PF$$
$$\dot{P}(P,F) = -\gamma P + \delta PF$$

with example parameters

$$\alpha = 1$$
$$\beta = 1$$
$$\gamma = 3$$
$$\delta = 1$$

and initial fish and penguin numbers

$$F_0 = 5$$
$$P_0 = 5$$

Units

$$[F] = \text{fish}$$
$$[P] = \text{penguins}$$
$$\dot{F} = \frac{\text{fish}}{\text{time}}$$
$$\dot{P} = \frac{\text{penguins}}{\text{time}}$$

4. Penguins and Fish: A Predator–Prey Equation

Model

Volterra–Lotka model

$$\dot{F}(P,F) = \alpha F - \beta PF$$
$$\dot{P}(P,F) = -\gamma P + \delta PF$$

with example parameters

$$\alpha = 1$$
$$\beta = 1$$
$$\gamma = 3$$
$$\delta = 1$$

and initial fish and penguin numbers

$$F_0 = 5$$
$$P_0 = 5$$

Units

$$[F] = \text{fish}$$
$$[P] = \text{penguins}$$
$$\dot{F} = \frac{\text{fish}}{\text{time}}$$
$$\dot{P} = \frac{\text{penguins}}{\text{time}}$$
$$[\alpha] = \text{Hz}$$

4. Penguins and Fish: A Predator–Prey Equation

Model

Volterra–Lotka model

$$\dot{F}(P,F) = \alpha F - \beta PF$$
$$\dot{P}(P,F) = -\gamma P + \delta PF$$

with example parameters

$$\alpha = 1$$
$$\beta = 1$$
$$\gamma = 3$$
$$\delta = 1$$

and initial fish and penguin numbers

$$F_0 = 5$$
$$P_0 = 5$$

Units

$$[F] = \text{fish}$$
$$[P] = \text{penguins}$$
$$\dot{F} = \frac{\text{fish}}{\text{time}}$$
$$\dot{P} = \frac{\text{penguins}}{\text{time}}$$
$$[\alpha] = \text{Hz} = \frac{\text{fish}}{\text{fish}\cdot\text{time}} \text{ (reproduction)}$$

4. Penguins and Fish: A Predator–Prey Equation

Model

Volterra–Lotka model

$$\dot{F}(P,F) = \quad \alpha F - \beta PF$$
$$\dot{P}(P,F) = -\gamma P + \delta PF$$

with example parameters

$$\alpha = 1$$
$$\beta = 1$$
$$\gamma = 3$$
$$\delta = 1$$

and initial fish and penguin numbers

$$F_0 = 5$$
$$P_0 = 5$$

Units

$$[F] = \text{fish}$$
$$[P] = \text{penguins}$$
$$\dot{F} = \frac{\text{fish}}{\text{time}}$$
$$\dot{P} = \frac{\text{penguins}}{\text{time}}$$
$$[\alpha] = \text{Hz} = \frac{\text{fish}}{\text{fish}\cdot\text{time}} \text{ (reproduction)}$$
$$[\beta] = \frac{1}{\text{penguin}\cdot\text{time}} \text{ (predator-dependent decay)}$$

4. Penguins and Fish: A Predator–Prey Equation

Model

Volterra–Lotka model

$$\dot{F}(P,F) = \quad \alpha F - \beta PF$$
$$\dot{P}(P,F) = -\gamma P + \delta PF$$

with example parameters

$$\alpha = 1$$
$$\beta = 1$$
$$\gamma = 3$$
$$\delta = 1$$

and initial fish and penguin numbers

$$F_0 = 5$$
$$P_0 = 5$$

Units

$$[F] = \text{fish}$$
$$[P] = \text{penguins}$$
$$\dot{F} = \frac{\text{fish}}{\text{time}}$$
$$\dot{P} = \frac{\text{penguins}}{\text{time}}$$
$$[\alpha] = \text{Hz} = \frac{\text{fish}}{\text{fish}\cdot\text{time}} \text{ (reproduction)}$$
$$[\beta] = \frac{1}{\text{penguin}\cdot\text{time}} \text{ (predator-dependent decay)}$$
$$[\gamma] = \text{Hz} = \frac{\text{penguins}}{\text{penguins}\cdot\text{time}} \text{ (emigration)}$$

4. Penguins and Fish: A Predator–Prey Equation

Model

Volterra–Lotka model

$$\dot{F}(P,F) = \quad \alpha F - \beta PF$$
$$\dot{P}(P,F) = -\gamma P + \delta PF$$

with example parameters

$$\alpha = 1$$
$$\beta = 1$$
$$\gamma = 3$$
$$\delta = 1$$

and initial fish and penguin numbers

$$F_0 = 5$$
$$P_0 = 5$$

Units

$$[F] = \text{fish}$$
$$[P] = \text{penguins}$$
$$\dot{F} = \frac{\text{fish}}{\text{time}}$$
$$\dot{P} = \frac{\text{penguins}}{\text{time}}$$
$$[\alpha] = \text{Hz} = \frac{\text{fish}}{\text{fish}\cdot\text{time}} \text{ (reproduction)}$$
$$[\beta] = \frac{1}{\text{penguin}\cdot\text{time}} \text{ (predator-dependent decay)}$$
$$[\gamma] = \text{Hz} = \frac{\text{penguins}}{\text{penguins}\cdot\text{time}} \text{ (emigration)}$$
$$[\delta] = \frac{1}{\text{fish}\cdot\text{time}} \text{ (prey-dependent growth)}$$

4. Penguins and Fish: A Predator–Prey Equation

Demo

4. Penguins and Fish: A Predator–Prey Equation

Demo

Lessons learned

- modeling including parameters and units

4. Penguins and Fish: A Predator–Prey Equation

Demo

Lessons learned

- modeling including parameters and units
- two coupled processes

$$\begin{aligned} \dot{F}(P,F) &= \quad \alpha F - \beta PF \\ \dot{P}(P,F) &= -\gamma P + \delta PF \end{aligned}$$

4. Penguins and Fish: A Predator–Prey Equation

Demo

Lessons learned

- modeling including parameters and units
- two coupled processes

$$\begin{aligned}\dot{F}(P,F) &= \quad \alpha F - \beta PF\\ \dot{P}(P,F) &= -\gamma P + \delta PF,\end{aligned}$$

i.e., two unknowns

4. Penguins and Fish: A Predator–Prey Equation

Demo

Lessons learned

- modeling including parameters and units
- two coupled processes

$$
\begin{aligned}
\dot{F}(P,F) &= \quad \alpha F - \beta PF \\
\dot{P}(P,F) &= -\gamma P + \delta PF\,,
\end{aligned}
$$

  i.e., two unknowns
- chemical rate equations work similarly

5. Swimming Baby Penguin: A Simple ODE

Problem

- baby emperor penguin
  (Aptenodytes forsteri)
  has been separated from its parent

https://upload.wikimedia.org/wikipedia/commons/2/2d/Aptenodytes_forsteri_-Snow_Hill_Island,_Antarctica_-juvenile_with_people-8.jpg

Ian Duffy, CC BY-SA 2.0

https://upload.wikimedia.org/wikipedia/commons/0/07/Emperor_Penguin_Manchot_empereur.jpg

Samuel Blanc, CC BY-SA 3.0

5. Swimming Baby Penguin: A Simple ODE

Problem

- baby emperor penguin *(Aptenodytes forsteri)* has been separated from its parent
- force attracting to parent

https://upload.wikimedia.org/wikipedia/commons/2/2d/Aptenodytes_forsteri_-Snow_Hill_Island,_Antarctica_-juvenile_with_people-8.jpg

Ian Duffy, CC BY-SA 2.0

https://upload.wikimedia.org/wikipedia/commons/0/07/Emperor_Penguin_Manchot_empereur.jpg

Samuel Blanc, CC BY-SA 3.0

5. Swimming Baby Penguin: A Simple ODE

Problem

- baby emperor penguin *(Aptenodytes forsteri)* has been separated from its parent
- force attracting to parent
- flow due to circumpolar current

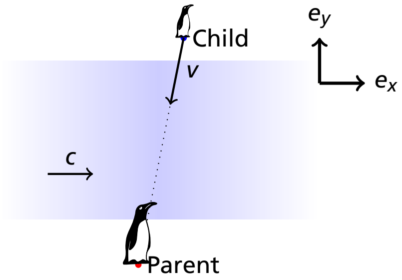
https://upload.wikimedia.org/wikipedia/commons/2/2d/Aptenodytes_forsteri_-Snow_Hill_Island,_Antarctica_-juvenile_with_people-8.jpg
Ian Duffy, CC BY-SA 2.0

https://upload.wikimedia.org/wikipedia/commons/0/07/Emperor_Penguin_Manchot_empereur.jpg
Samuel Blanc, CC BY-SA 3.0

5. Swimming Baby Penguin: A Simple ODE

Model

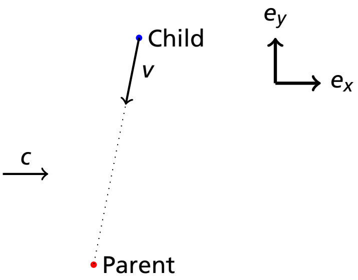
- child penguin at $\begin{pmatrix} x \\ y \end{pmatrix}$
- parent penguin at origin
- swimming velocity v
- drift velocity (current) c



Model

- child penguin at $\begin{pmatrix} x \\ y \end{pmatrix}$
- parent penguin at origin
- swimming velocity v
- drift velocity (current) c

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} \left(t, \begin{pmatrix} x \\ y \end{pmatrix}\right) = \frac{-1}{\sqrt{x^2(t) + y^2(t)}} \begin{pmatrix} x \\ y \end{pmatrix} v + \begin{pmatrix} c \\ 0 \end{pmatrix}$$



5. Swimming Baby Penguin: A Simple ODE

Implementation

- again use scipy
- time to simulate unknown, thus need stopping criterion

```
def termination_condition(y):
    return numpy.linalg.norm(y)<1e-3
```

```
while not termination_condition(integrator.y):
    integrator.integrate(t=t_max, step=True)
    trajectory.append([integrator.t, integrator.y[0], integrator.y[1]])
```

5. Swimming Baby Penguin: A Simple ODE

Demo

5. Swimming Baby Penguin: A Simple ODE

Demo

Lessons learned

- modeling based on force diagram
- stopping criterion may be necessary
- parameters influence behavior of solution

Problem

- Magellanic penguins *(Spheniscus magellanicus)* like to travel

6. Travelling Penguins: Understanding a Given ODE

Problem

- Magellanic penguins
  (Spheniscus magellanicus)
  like to travel
- swim in water, climb ice floe, enter igloo

6. Travelling Penguins: Understanding a Given ODE

Problem

- Magellanic penguins
  (Spheniscus magellanicus)
  like to travel
- swim in water, climb ice floe, enter igloo
- preference depending on “fishyness”

6. Travelling Penguins: Understanding a Given ODE

Problem

- Magellanic penguins
  (Spheniscus magellanicus)
  like to travel
- swim in water, climb ice floe, enter igloo
- preference depending on “fishyness”
- effort to climb ice floe, narrow entrance to igloo

6. Travelling Penguins: Understanding a Given ODE

Problem

- Magellanic penguins
  (Spheniscus magellanicus)
  like to travel
- swim in water, climb ice floe, enter igloo
- preference depending on "fishyness"
- effort to climb ice floe, narrow entrance to igloo
- penguins breed inside igloo

6. Travelling Penguins: Understanding a Given ODE

Model

- u_p penguin density in water (**p**olar sea)
- u_i penguin density on **i**ce floe
- u_c penguin density inside igloo (**c**abin)

6. Travelling Penguins: Understanding a Given ODE

Model

- u_p penguin density in water (**p**olar sea)
- u_i penguin density on **i**ce floe
- u_c penguin density inside igloo (**c**abin)

$$\mathrm{d}_t \begin{pmatrix} u_p \\ u_i \\ u_c \end{pmatrix} = \begin{pmatrix} \frac{1}{v_p} P_{pi} \left(\kappa_p u_p - \kappa_i u_i\right) \\ \frac{1}{v_i} \left[P_{pi} \left(\kappa_i u_i - \kappa_p u_p\right) + P_{ic} \left(\kappa_i u_i - \kappa_c u_c\right)\right] \\ \frac{1}{v_c} P_{ic} \left(\kappa_c u_c - \kappa_i u_i\right) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ P_f u_c \end{pmatrix} \tag{2}$$

6. Travelling Penguins: Understanding a Given ODE

Model

- u_p penguin density in water (**p**olar sea)
- u_i penguin density on **i**ce floe
- u_c penguin density inside igloo (**c**abin)

$$\mathrm{d}_t \begin{pmatrix} u_p \\ u_i \\ u_c \end{pmatrix} = \begin{pmatrix} \\ \\ \\ \end{pmatrix} \tag{2}$$

Penguin density changes when

6. Travelling Penguins: Understanding a Given ODE

Model

- u_p penguin density in water (polar sea)
- u_i penguin density on ice floe
- u_c penguin density inside igloo (cabin)

$$\mathrm{d}_t \begin{pmatrix} u_p \\ u_i \\ u_c \end{pmatrix} = \begin{pmatrix} \frac{1}{v_p} P_{pi} \left(\kappa_p u_p - \kappa_i u_i \right) \\ \frac{1}{v_i} \quad P_{pi} \left(\kappa_i u_i - \kappa_p u_p \right) \\ \\ \end{pmatrix} \tag{2}$$

Penguin density changes when

- climbing ice floes

6. Travelling Penguins: Understanding a Given ODE

Model

- u_p penguin density in water (**p**olar sea)
- u_i penguin density on **i**ce floe
- u_c penguin density inside igloo (**c**abin)

$$\mathrm{d}_t \begin{pmatrix} u_p \\ u_i \\ u_c \end{pmatrix} = \begin{pmatrix} \frac{1}{v_p} P_{pi} \left(\kappa_p u_p - \kappa_i u_i \right) \\ \frac{1}{v_i} \left[P_{pi} \left(\kappa_i u_i - \kappa_p u_p \right) + P_{ic} \left(\kappa_i u_i - \kappa_c u_c \right) \right] \\ \frac{1}{v_c} P_{ic} \left(\kappa_c u_c - \kappa_i u_i \right) \end{pmatrix} \tag{2}$$

Penguin density changes when

- climbing ice floes
- entering igloo

6. Travelling Penguins: Understanding a Given ODE

Model

- u_p penguin density in water (**p**olar sea)
- u_i penguin density on **i**ce floe
- u_c penguin density inside igloo (**c**abin)

$$d_t \begin{pmatrix} u_p \\ u_i \\ u_c \end{pmatrix} = \begin{pmatrix} \frac{1}{v_p} P_{pi} \left(\kappa_p u_p - \kappa_i u_i \right) \\ \frac{1}{v_i} \left[P_{pi} \left(\kappa_i u_i - \kappa_p u_p \right) + P_{ic} \left(\kappa_i u_i - \kappa_c u_c \right) \right] \\ \frac{1}{v_c} P_{ic} \left(\kappa_c u_c - \kappa_i u_i \right) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ P_f u_c \end{pmatrix} \tag{2}$$

Penguin density changes when

- climbing ice floes
- entering igloo
- hatching

6. Travelling Penguins: Understanding a Given ODE

Interpretation

$$\mathrm{d}_t \begin{pmatrix} u_p \\ u_i \\ u_c \end{pmatrix} = \begin{pmatrix} \frac{1}{v_p} P_{pi} \left(\kappa_p u_p - \kappa_i u_i \right) \\ \frac{1}{v_i} \left[P_{pi} \left(\kappa_i u_i - \kappa_p u_p \right) + P_{ic} \left(\kappa_i u_i - \kappa_c u_c \right) \right] \\ \frac{1}{v_c} P_{ic} \left(\kappa_c u_c - \kappa_i u_i \right) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ P_f u_c \end{pmatrix} \tag{3}$$

6. Travelling Penguins: Understanding a Given ODE

Interpretation

$$\mathrm{d}_t \begin{pmatrix} u_p \\ u_i \\ u_c \end{pmatrix} = \begin{pmatrix} \frac{1}{v_p} P_{pi} \left(\kappa_p u_p - \kappa_i u_i \right) \\ \frac{1}{v_i} \left[P_{pi} \left(\kappa_i u_i - \kappa_p u_p \right) + P_{ic} \left(\kappa_i u_i - \kappa_c u_c \right) \right] \\ \frac{1}{v_c} P_{ic} \left(\kappa_c u_c - \kappa_i u_i \right) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ P_f u_c \end{pmatrix} \tag{3}$$

Additionally: circumpolar current sends penguins to other holiday resorts
⇝ another, larger ODE (stiff)

6. Travelling Penguins: Understanding a Given ODE

Interpretation

$$\mathrm{d}_t \begin{pmatrix} u_p \\ u_i \\ u_c \end{pmatrix} = \begin{pmatrix} \frac{1}{v_p} P_{pi} \left(\kappa_p u_p - \kappa_i u_i\right) \\ \frac{1}{v_i} \left[P_{pi} \left(\kappa_i u_i - \kappa_p u_p\right) + P_{ic} \left(\kappa_i u_i - \kappa_c u_c\right)\right] \\ \frac{1}{v_c} P_{ic} \left(\kappa_c u_c - \kappa_i u_i\right) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ P_f u_c \end{pmatrix} \tag{3}$$

- volume fractions $\Leftarrow$ penguin conservation $\rightsquigarrow$ density

Additionally: circumpolar current sends penguins to other holiday resorts
$\rightsquigarrow$ another, larger ODE (stiff)

6. Travelling Penguins: Understanding a Given ODE

Interpretation

$$
\mathrm{d}_t \begin{pmatrix} u_p \\ u_i \\ u_c \end{pmatrix} = \begin{pmatrix} \frac{1}{v_p} P_{pi} \left(\kappa_p u_p - \kappa_i u_i \right) \\ \frac{1}{v_i} \left[P_{pi} \left(\kappa_i u_i - \kappa_p u_p \right) + P_{ic} \left(\kappa_i u_i - \kappa_c u_c \right) \right] \\ \frac{1}{v_c} P_{ic} \left(\kappa_c u_c - \kappa_i u_i \right) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ P_f u_c \end{pmatrix} \tag{3}
$$

- volume fractions $\Leftarrow$ penguin conservation $\rightsquigarrow$ density
- permeability $\Leftarrow$ effort for climbing ice floe / squeezing through igloo opening

Additionally: circumpolar current sends penguins to other holiday resorts
$\rightsquigarrow$ another, larger ODE (stiff)

6. Travelling Penguins: Understanding a Given ODE

Interpretation

$$\mathrm{d}_t \begin{pmatrix} u_p \\ u_i \\ u_c \end{pmatrix} = \begin{pmatrix} \frac{1}{v_p} P_{pi} \left(\kappa_p u_p - \kappa_i u_i \right) \\ \frac{1}{v_i} \left[P_{pi} \left(\kappa_i u_i - \kappa_p u_p \right) + P_{ic} \left(\kappa_i u_i - \kappa_c u_c \right) \right] \\ \frac{1}{v_c} P_{ic} \left(\kappa_c u_c - \kappa_i u_i \right) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ P_f u_c \end{pmatrix} \tag{3}$$

- volume fractions $\Leftarrow$ penguin conservation $\rightsquigarrow$ density
- permeability $\Leftarrow$ effort for climbing ice floe / squeezing through igloo opening
- partition coefficient $\Leftarrow$ preference for staying at location, “when is equilibrium achieved?”

Additionally: circumpolar current sends penguins to other holiday resorts
$\rightsquigarrow$ another, larger ODE (stiff)

6. Travelling Penguins: Understanding a Given ODE

Interpretation

$$
\mathrm{d}_t \begin{pmatrix} u_p \\ u_i \\ u_c \end{pmatrix} = \begin{pmatrix} \frac{1}{v_p} P_{pi} \left(\kappa_p u_p - \kappa_i u_i \right) \\ \frac{1}{v_i} \left[P_{pi} \left(\kappa_i u_i - \kappa_p u_p \right) + P_{ic} \left(\kappa_i u_i - \kappa_c u_c \right) \right] \\ \frac{1}{v_c} P_{ic} \left(\kappa_c u_c - \kappa_i u_i \right) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ P_f u_c \end{pmatrix} \tag{3}
$$

- volume fractions $\Leftarrow$ penguin conservation $\rightsquigarrow$ density
- permeability $\Leftarrow$ effort for climbing ice floe / squeezing through igloo opening
- partition coefficient $\Leftarrow$ preference for staying at location, “when is equilibrium achieved?”
- fertility

Additionally: circumpolar current sends penguins to other holiday resorts
$\rightsquigarrow$ another, larger ODE (stiff)

6. Travelling Penguins: Understanding a Given ODE

Implementation

- Runge–Kutta–Fehlberg 4th/5th order scheme
  with adaptive step size control
- backwards differentiation formula (BDF) scheme
  as external solver from CVODE/SUNDIALS library
  adaptive step size and suitable for stiff problems

6. Travelling Penguins: Understanding a Given ODE

Simulation Results

Sorry about the incorrect geometry ☺

7. Summary

- penguins are cool
- ODEs are useful to describe many processes
- there are various techniques for solving them numerically

Contact:

Cristoffer Cordes ‹cristoffer.cordes@mevis.fraunhofer.de›

Ole Schwen ‹ole.schwen@mevis.fraunhofer.de›

Multifunktionspinguin ‹mufupi@wp1162144.server-he.de›