Durham University
Department of Computer Science

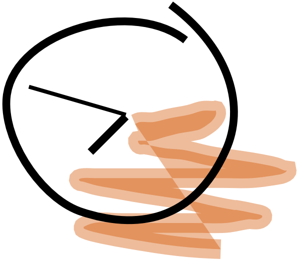
Numerical Algorithms (COMP 3371)

Session VI: ODEs and first glimpses of Euler

Dr. Weinzierl

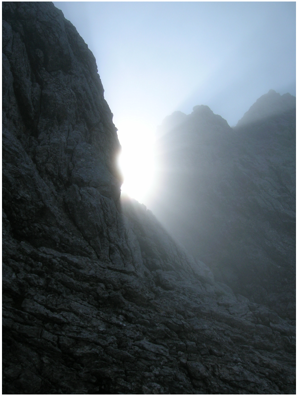
Michaelmas term 2019

Recapitulation



1. What is an *ill-posed* problem?
2. Why is an *ill-conditioned* problem? Why are they problematic to solve?
3. Can an algorithm for an ill-conditioned problem be forward stable?
4. Write down the Taylor series.

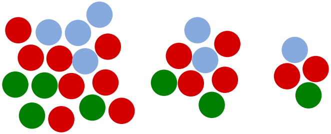
Outline



Ordinary Differential Equations
Two model problems
Taylor and explicit Euler

Example: Radioactive decay

Durham University
Department of Computer Science



- Many problems in engineering, physics, social sciences are formulated in terms of functions that satisfy an equation with one or more derivatives.
- Example: Radioactive decay

$$
\begin{aligned}
\frac{\partial r(t)}{\partial t} &= -k \cdot r(t) \\
\partial_t r(t) &= -k \cdot r(t) \\
r'(t) &= -k \cdot r(t) \\
\dot{r}(t) &= -k \cdot r(t)
\end{aligned}
$$

  with constant k and r being the amount of radioactive substance
- The four variants are different notations for the same thing
- Can you describe (informally) what the ODE means?

Ordinary Differential Equations (ODE)

Durham University
Department of Computer Science

ODE: Let the function $f(t)$ depend on the time t. An ordinary differential equation (ODE) of order n is

$$F(t, f(t), f'(t), f''(t), f'''(t), \dots, f^{(n-1)}(t)) = f^{(n)}(t).$$

- Function $f(t)$ depends on its own derivatives.
- n above is the *order* of the ODE.
- $\underbrace{1 - t}_{F(t)} = f(t)$
  Degenerated ODE (not really one)
- $\underbrace{1 - f(t)}_{F(t,f)} = f'(t)$
  First order ODE

$\Rightarrow$ Most physical phenomena (first order) are described this way

$\Rightarrow$ Only one derivative (∂_t or ∂_x). More than one derivative would construct partial differential equations (PDEs)

Some ODE terminology

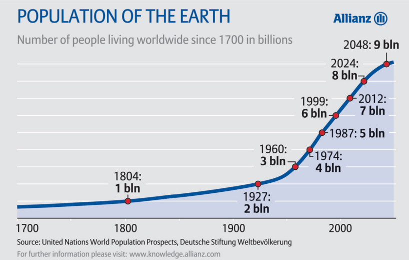
Linear: An ODE is linear iff the individual derivatives of f are isolated:

$$
\begin{aligned}
f'(t) &= f^2(t) && \text{non-linear} \\
f''(t) &= f'(t)f(t) + t && \text{non-linear} \\
f'(t) &= f(t) \cdot t && \text{linear}
\end{aligned}
$$

Autonomous: An ODE is autonomous if F does not directly depend on t (but only indirectly through $f(t)$):

$$
\begin{aligned}
f''(t) &= f'(t)f(t) + t && \text{not autonomous} \\
f'(t) &= f(t) \cdot t && \text{not autonomous} \\
f'(t) &= f(t) + 2 && \text{autonomous}
\end{aligned}
$$

Example: Model of Malthus



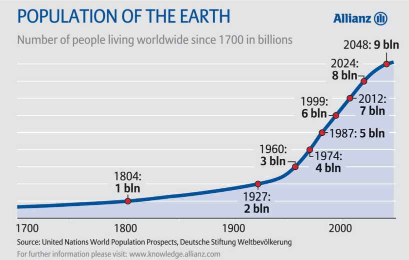
- Thomas Robert Malthus: *An Essay on the Principle of Population* (1798)
- Study one type of species (rabbits in Australia, e.g.)
- Malthus” assumptions:
  1. The ratio of rabbits reproducing is constant: γ
  2. The ratio of rabbits dying is constant: δ

$$\partial_t p(t) = (\gamma - \delta) p(t)$$

- We know the solution to this equation
- Is it a realistic model (same model was used also in economics sometimes)?

Example: Model of Malthus

Durham University
Department of Computer Science



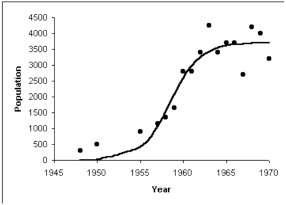
- Thomas Robert Malthus: *An Essay on the Principle of Population* (1798)
- Study one type of species (rabbits in Australia, e.g.)
- Malthus" assumptions:
  1. The ratio of rabbits reproducing is constant: γ
  2. The ratio of rabbits dying is constant: δ

$$\partial_t p(t) = (\gamma - \delta) p(t)$$

- We know the solution to this equation
- Is it a realistic model (same model was used also in economics sometimes)?
- Model was realistic for population growth between 1700–1960!

Example: Model of Verhulst

Durham University
Department of Computer Science



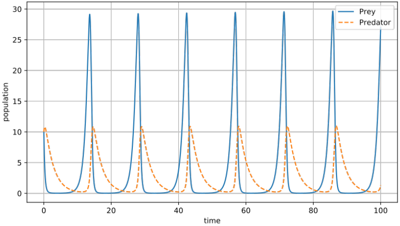
Barnacle Goose population (from Armson, Cockroft and Stone, 2000)

- Established by Verhulst in 19th century
- Extension: due to limited food
  1. the bigger the population the higher the death rates
  2. the bigger the population the lower the birth rates

$$
\begin{aligned}
\partial_t p(t) &= \gamma(t) - \delta(t) \\
\gamma(t) &= \gamma_0 - \gamma_1 p(t) \\
\delta(t) &= \delta_0 + \delta_1 p(t) \qquad \text{or simply} \\
\partial_t p(t) &= b - ap(t)
\end{aligned}
$$

Vertical-valued ODEs

Durham University
Department of Computer Science



From Wikipedia.

- Our f (or p or whatever used) can be vector, i.e. $(f_1, f_2)(t)$
- Predator-prey model (Lotka-Volterra):

$$\begin{aligned}\partial_t p_1(t) &= \alpha p_1(t) - \beta p_1(t) p_2(t) \\ \partial_t p_2(t) &= \delta p_1(t) p_2(t) - \gamma p_2(t)\end{aligned}$$

- p_1 is prey (rabbit)
  - The more rabbits, the faster the rabbit population growth (α)
  - The more rabbits and foxes, the more rabbits die (β)
- p_2 is predator (foxes) feeding on rabits
  - The more rabbits and foxes, the higher the reproduction rate (δ)
  - The more foxes, the more foxed die (γ)

First oder ODEs

Durham University
Department of Computer Science

$$
\begin{aligned}
f''(t) + f(t) &= f'(t) \\
x_1(t) := f(t) & \qquad x_2(t) := f'(t)
\end{aligned}
$$

We obtain:

$$
x_2'(t) + x_1(t) = x_2(t) \quad \Rightarrow \quad
\begin{aligned}
x_2'(t) &= x_2(t) - x_1(t) \\
x_1'(t) &= x_2(t)
\end{aligned}
$$

- For ODEs, we can mechanically replace higher-order terms by new (helper) functions
- We reduce the order of the ODE system but increase the entries of the vector

$\Rightarrow$ We don't have to study higher-order ODEs

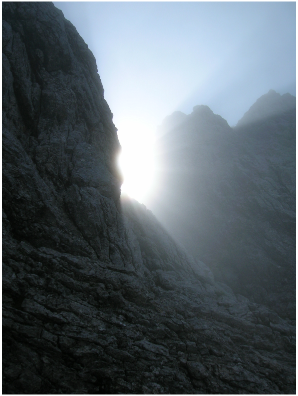
$\Rightarrow$ This does not work (that way) for PDEs

Concept of building block

- Content
  - Introduce term ODE at hands of radioactive decay
  - Define terminology around ODEs
  - Give a few examples of ODEs from population modelling
  - Discuss vector-valued ODEs
- Expected Learning Outcomes
  - Student knows fundamental ODE terminology
  - Student can explain population models (not to be memorised)
- Further reading
  - KHANAcademy: `https://www.khanacademy.org/math/differential-equations/first-order-differential-equations/modeling-with-differential-equations/v/modeling-population-with-simple-differential-equation`

Outline

Durham University
Department of Computer Science



Ordinary Differential Equations
Two model problems
Taylor and explicit Euler

Newton's law of gravity

Durham University
Department of Computer Science

- Gravity is a constant (9.8 here)
- Acceleration is the derivative of velocity with respect to time:

$$v'(t) = -9.8$$

- Position of particle is changed by velocity:

$$x'(t) = v(t)$$

- We obtain the second-order ODE:

$$x''(t) = -9.8$$

- First-order formulation:

$$\partial_t \begin{pmatrix} x \\ v \end{pmatrix}(t) = \begin{pmatrix} v \\ 0 \end{pmatrix}(t) + \begin{pmatrix} 0 \\ -9.8 \end{pmatrix}$$

$\Rightarrow$ This is something we can solve with A-level calculus

Newton's Second Law

$$F = ma$$

- A particle is given by its position x, mass m and its velocity v
- x and v are vectors and they depend on the time t
- Definitions of x and v plus Newton's second law:

$$\begin{array}{rcl} x'(t) & = & v(t) \\ v'(t) & = & F/m \\ F & = & \sum_i \frac{m \cdot m_i}{r^2} \left(\frac{x - x_i}{r} \right) \qquad r \text{ depends on all the } x \end{array}$$

- This is a system of two ordinary differential equations (ODEs), i.e. functions of type

$$\begin{pmatrix} x \\ v \end{pmatrix}'(t) = \begin{pmatrix} v(t) \\ \frac{1}{m} \sum_i \frac{m \cdot m_i}{r^2} \left(\frac{x(t) - x_i(t)}{r} \right) \end{pmatrix}$$

$\Rightarrow$ Not that easy to solve anymore

$\Rightarrow$ For N particles, we end up with $3 \cdot 2 \cdot N$ array entries (doubles)

Initial value problems (IVP)

Durham University
Department of Computer Science

- We know know where our coursework's equations come from, but we have to solve them
- High school approach (symbolic):
  - Search for fundamental solution (integrate $\Rightarrow$ obtain some magic constants)
  - Plug given values into solution
  - Closed expression represents *all* solutions
- Our approach (numerical):
  - Write code that approximates solution $f(t)$
  - Take given initial values and plug them into code
  - Run the code
  - Numerical result represents *one* solution

$\Rightarrow$ To "solve" an ODE, we always need initial value (*Initial Value Problem* (IVP))

Without proper/valid initial values, problem would be ill-posed

IVP in action

Radioactive decay—an analytical solution

$$
\begin{aligned}
\partial_t r(t) &= -k \cdot r(t) \\
\text{ansatz:} \quad r(t) &= ae^{bt}
\end{aligned}
$$

Our ansatz here stems from experience/intuition. In general, it is not that trivial (maybe even impossible). Then, we need numerical approximation.

IVP in action

Radioactive decay—an analytical solution

Durham University
Department of Computer Science

$$
\begin{array}{rrcl}
 & \partial_t r(t) & = & -k \cdot r(t) \\
\text{ansatz:} & r(t) & = & ae^{bt}
\end{array}
$$

Our ansatz here stems from experience/intuition. In general, it is not that trivial (maybe even impossible). Then, we need numerical approximation.

$$
\begin{array}{rrcl}
\text{ansatz:} & r(t) & = & ae^{bt} \\
 & \Rightarrow \partial_t r(t) & = & abe^{bt} \\
 & & = & b \cdot r(t) \\
 & \Rightarrow b & = & -k \\
 & \Rightarrow r(t) & = & ae^{-kt}
\end{array}
$$

So there's an infinite number of solutions for all the a?

IVP in action

Radioactive decay—an analytical solution

Durham University
Department of Computer Science

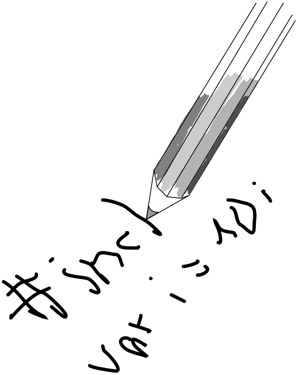
$$
\begin{aligned}
\partial_t r(t) &= -k \cdot r(t) \\
\text{ansatz:} \quad r(t) &= ae^{bt}
\end{aligned}
$$

Our ansatz here stems from experience/intuition. In general, it is not that trivial (maybe even impossible). Then, we need numerical approximation.

$$
\begin{aligned}
\text{ansatz:} \quad r(t) &= ae^{bt} \\
\Rightarrow \partial_t r(t) &= abe^{bt} \\
&= b \cdot r(t) \\
\Rightarrow b &= -k \\
\Rightarrow r(t) &= ae^{-kt}
\end{aligned}
$$

So there's an infinite number of solutions for all the a?
No, because we have an initial amount of substance $r(t)$ that fixes a.

Dahlquist's test equation



- Radioactive decay had been motivating example, but ...
  $\Rightarrow$ abstract from application

$$y'(t) = \lambda y(t) \quad \text{with} \quad y(0) = y_0$$

- Perfect test equation
  - Simple
  - We know analytical solution

$$y(t) = y_0 \; e^{\lambda t}$$

    $\Rightarrow$ easy to compare/measure errors
    $\Rightarrow$ well-posed problem
  - Interesting behaviour (as we'll see)
- Called Dahlquist's test equation
- Condition number: $\kappa_{y_0} = |\frac{y_0 \cdot e^{\lambda t}}{y_0 \; e^{\lambda t}}|$
  $\Rightarrow$ without analytic solution, computing κ would be nasty
  $\Rightarrow$ well-conditioned problem

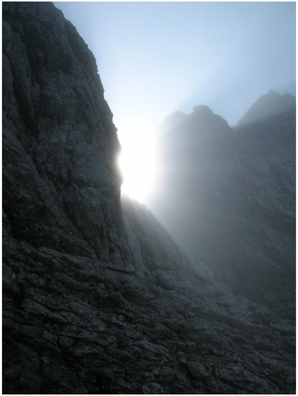
Concept of building block

Durham University
Department of Computer Science

- Content
  - Derive model problem of coursework from school physics
  - Introduce Dahlquist test equation
- Expected Learning Outcomes
  - Student knows Dahlquist test equation
  - Student can explain where coursework ODE comes from
  - Student can show properties of test equation (well-posedness, conditioning)

Outline

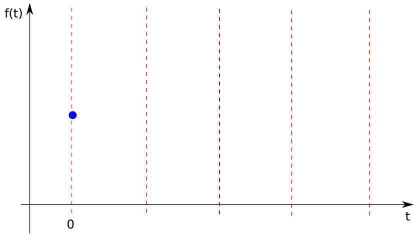
Durham University
Department of Computer Science



Ordinary Differential Equations
Two model problems
Taylor and explicit Euler

Time stepping for ODEs

Durham University
Department of Computer Science



$$f'(f(t),t) = F(f,t) \qquad \text{with } f(0) = f_0$$

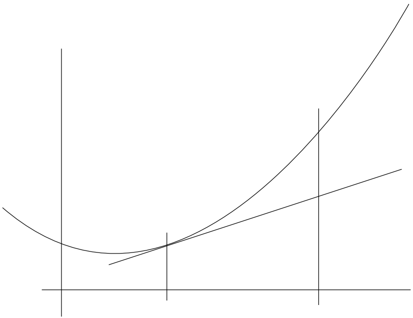
- Problem with ODEs: $f(t)$ is implicitly given while we would like to have it explicitly
- Time slicing: We cut the solution into snapshots (with fixed time step sizes Δt for the time being)

$\Rightarrow$ We are interested in the numerical solution f_h at fixed time steps

- We know $f(0) = f_h(0)$
- We make certain assumptions/simplifications about the solution in-between two snapshots

Taylor

Durham University
Department of Computer Science



- Assume $f_h(t)$ known
- Use Taylor:

$$f_h(t+\Delta t) \quad = \quad f(t) + \frac{\Delta t^1}{1!} f'(t) + \frac{\Delta t^2}{2!} f''(t) + \frac{\Delta t^3}{3!} f'''(t) + \dots$$

- Cut after second term
- Insert ODE: $f'(t) = F(f(t), t)$
- Walk for Δt along derivative

Explicit Euler

Durham University
Department of Computer Science

$$f(t+\Delta t) = f(t) + \Delta t \cdot f'(t)$$

- We cut the Taylor series after the first term and end up with *explicit Euler*
- The source code realisation of this scheme is a simple for loop
- Explicit = we can step through time without running into linear equation systems
- It is a *single step method*, as a new solution only needs the previous one (Δt is often also denoted as h or dt)
- It is a *finite difference* method, as we use the difference quotient instead of the derivative:

$$f(t+\Delta t) = f(t) + \Delta t \cdot f'(t) \Leftrightarrow f'(t) = \frac{f(t+\Delta t) - f(t)}{\Delta t}$$

```
double t    =   0.0;
double f    =   4.0;
double dt   = 0.001;

while (t<2.0) {                       // simulate until t=2.0
  t = t + dt;
  f = f + dt * F(f,t);                // you have to implement F
  std::cout << "t=" << t << ",f=" << f << std::endl;
}
```

Concept of building block

Durham University
Department of Computer Science

- Content
  - Introduce idea of explicit Euler at hands of Taylor expansion
  - Motivate Euler with “walk along derivative” sketch
  - Show pseudo code
- Expected Learning Outcomes
  - Student *knows* the definition of explicit Euler
  - Student can *sketch* idea how to derive explicit Euler

Summary, outlook & homework

Durham University
Department of Computer Science

Concepts discussed:

- We have defined condition number
- We have found a formal derivation of the explicit Euler
- We have studied some experiments we'd done before

Next:

- We study Euler's error in detail
- We find that there are further notions of unstable

Preparation:

- Revise the section on radioactive decay as ODE example and reiterate how to derive the analytic solution
- Solve the problem **Explicit Euler for some test ODEs** from the worksheet