Hydroinformatics
Module 4: Numerical Methods I

Lecture 1 and 2: Models, ODE, Explicit and implicit methods

Remarks:

1. Schedule

2. Prerequisites
   - Basic mathematics,
   - Fluid dynamics and fluid mechanics,
   - Mathematical representation of fluid flow equations

3. Marking
   - Assignments (15%)
   - Written examination (30%)
Course

Main aim
- introduce numerical techniques which can be used on computers
- overview of
  - WHAT it can be done
  - HOW it can be done

At the end of lectures you will:
- apply standard methods to simple flow problems and determine the optimal values for time steps and grid size
- assess simulation results produced by different methods
- evaluate the relevance of their use
- identify and analyse the reason for failure of particular methods and, possibly, provide solutions
**Computational Fluid Dynamics**

- Computational Fluid Dynamics (CFD) is the science of predicting fluid flow, heat transfer, mass transfer, chemical reactions, and related phenomena by solving the mathematical equations which govern these processes using a numerical process (that is, on a computer).

**Solution of Equations**

Phenomena in nature - described in terms of properties that prevail at each point of space and time separately

- Formulating PDE's governing the phenomena
- SOLUTIONS
- HOW ??
Solution of Equations

Phenomena in nature - described in terms of properties that prevail at each point of space and time separately

Formulating PDE's governing the phenomena

SOLUTIONS

Analytical

Separation of variables
Integral Solution (Green)
Integral transforms (Fourier & Laplace)

Too complex equations?
What is a model?

Is this a model?
What is a model?

Is this a model?

— Naomi Campbell
What is a model?

Is this a model?
What is a model?

A simplification of reality, but with essential features -- an example:
What is a model?

- Different forms
- Engineering problem solving
  - A model is a simplified, schematic representation of the real world, a representation that retains enough aspects of the original system to make it useful to the modeller.
- In fluid dynamics
  - Physical models (scale models);
  - Mathematical models
    - Conceptual
    - Simulation models

Error and uncertainty in modelling

- Definitions of model:
  - "A system which will convert a given input (geometry, boundary conditions, force, etc.) into an output (flow rates, levels, pressures, etc.) to be used in civil engineering design and operation" (Novak)
  - "A simplified representation of the natural system" (Resfgaard)

- It is not reasonable to expect that a "simplified representation" convert input into exact output.
- Errors are inevitable.
- The actual values of the output variables are uncertain.
What’s the difference between a modeling program, and a model?

- Modelling program = software
  - Software with an array of equations that simulate river flow
- Model = software + data
  - A modelling program + data specific to a certain river

What is a model?

- Mathematical models
Solution of Equations

Phenomena in nature - described in terms of properties that prevail at each point of space and time separately

Formulating PDE's governing the phenomena

SOLUTIONS

Analytical
- Separation of variables
- Integral Solution (Green)

Numerical techniques
- Integral transforms (Fourier & Laplace)
- FDM
- FVM
- BEM
- FEM

Why numerical methods?

- Physical phenomenon
- Differential Equations + Boundary Conditions

Analytical Methods

Approximate Methods

Simple Differential Equations
- FDM
- FVM
- FEM
- BEM

- General differential equations are too complicated to be solved analytically
- FDM, FEM, FVM, BEM are approximate numerical approaches for solving differential equations
### Why Numerical Methods?

- **Analytical solutions**
  - derived only for few engineering problems
  - limited number of closed form solutions
  - useful to provide insight to the behaviour of an engineering system

- **Numerical solutions**
  - fill the gap between theory and experiment

---

### Why Numerical Methods?

- Easy input - preprocessor.
- Solves many types of problems
- Modular design
  - fluids, dynamics, heat, etc.
- Can run on PC’s now.
- Relatively low cost, even sometimes cheaper
Why Numerical Methods?

- Models are no longer restricted to closed solution problems
- Represent accurately realistically complex system such as:
  - non-linear systems
  - time dependent state variables
- Sometimes physical experiments are impossible to be simulated

Simple Mathematical Models

Dependent variable = \( f \) (independent variable, parameters, forcing function)

- **Dependent variable**
  - A characteristic that usually reflects the behavior or state of the system
- **Independent variable**
  - Are usually dimensions, such as time and space
- **Parameters**
  - Are reflective of system’s properties or compositions
- **Forcing functions**
  - External influences acting on the system
**Basic principle of Numerical Methods**

**Discretisation**

- An exact 'continuous' solution is represented by an approximate 'discrete' solution at a number of points.

**Diagram:**
- **Exact solution**
- **Discretised approximation**

**Approximation of the derivative**

\[
\frac{\Delta U}{\Delta t} = \frac{U(t^2) - U(t^1)}{t^2 - t^1} = \frac{U^2 - U^1}{t^2 - t^1}
\]
Example - simple mathematical model

- Newton Second Law

\[ F = ma \quad \text{or} \quad a = \frac{F}{m} \]

\[ v(t^{+}) = v(t) + \left[ g - \frac{c}{m}v(t) \right] (t^{+} - t) \]

New value = old value + slope x step size

Analytical solution - \[ v(t) = \frac{gm}{c} \left[ 1 - e^{-ct/m} \right] \]

---

Example - simple mathematical model

- Newton Second Law - Falling object

\( M = 68.1 \text{ kg} \)

\( c = 12.5 \text{ kg/s} \)

\( t \text{ step} = 2 \text{ sec} \)

<table>
<thead>
<tr>
<th>Time (s)</th>
<th>Computed velocity (m/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Numerically</td>
</tr>
<tr>
<td>0.0</td>
<td>0.000</td>
</tr>
<tr>
<td>2.0</td>
<td>19.600</td>
</tr>
<tr>
<td>4.0</td>
<td>32.005</td>
</tr>
<tr>
<td>6.0</td>
<td>39.856</td>
</tr>
<tr>
<td>8.0</td>
<td>44.824</td>
</tr>
<tr>
<td>10.0</td>
<td>47.969</td>
</tr>
<tr>
<td>12.0</td>
<td>49.959</td>
</tr>
<tr>
<td>:</td>
<td>53.390</td>
</tr>
</tbody>
</table>
Example - simple mathematical model

Newton Second Law - Falling object

M = 68.1 kg
\( c = 12.5 \text{ kg/s} \)
\( \text{t step} = 2 \text{ sec} \)

2- Differential equations

- Quick review -
Introduction and First Definitions

- Differential equations are equations that contain derivatives.
  - In an algebraic equation, the unknown is a number. In a differential equation, the unknown is a function.
  - Most often the unknown function we seek, expresses the behavior of the state variable or variables as a function of time. This function it is called state equation.

- The order of the differential equation is the order of the derivative of the unknown function involved in the equation.

Introduction and First Definitions

- A linear differential equation of order n is a differential written in the following form:
  \[ a_n(x) \frac{d^n y}{dx^n} + a_{n-1}(x) \frac{d^{n-1} y}{dx^{n-1}} + \ldots + a_1(x) \frac{dy}{dx} + a_0(x) y = f(x) \]
  where \( a_n(x) \) is not the zero function.

  Note: Some books may use the notation \( y' \), \( y'' \), \( y''' \), \( y^{(4)} \), ... for the derivatives.
Linear Differential Equations

- A linear equation obliges the unknown function $y(x)$ to have some restrictions. Accepted operation on $y$ are:
  - Differentiating $y$
  - Multiplying $y$ and its derivatives by a function of the variable $x$.
  - Adding what you obtained in previous step and let it be equal to a function of $x$.

Solving differential equations (DE)

- A major field of study
  - Existence: Does a differential equation have a solution?
  - Uniqueness: Does a differential equation have more than one solution? If yes, how can we find a solution which satisfies particular conditions?
- Differential equations require integration to be solved
  - If the values of the unknown function $y(x)$ and its derivatives at some point are known the solution to DE is called an initial value problem (in short IVP).
  - If no initial conditions are given, the description of all solutions to the DE is called the general solution
- Some kinds of D.E.’s can be solved exactly; many not
  - for those that cannot, there are various ways to approximate the solution
Differential Equations

Single Variable

We know:

\[ \frac{dy}{dt} = F(y) \]

We want to know:

\[ y = f(t) \]

Differential Equations

Several Variable

We know:

\[ \frac{dy}{dt} = F_1(y, x, \ldots, q) \]
\[ \frac{dx}{dt} = F_2(y, x, \ldots, q) \]
\[ \vdots \]
\[ \frac{dq}{dt} = F_n(y, x, \ldots, q) \]

We want to know:

\[ y = f_1(t) \]
\[ x = f_2(t) \]
\[ \vdots \]
\[ q = f_n(t) \]
Methods for solving DEs

- Solving a differential equation can be done in three major ways:
  - analytically,
  - qualitatively,
  - numerically.

Partial derivatives

- When a function changes with more than one variable at once – for example, both time and space – we use partial derivatives
- Notation:
  \[ \frac{\partial y}{\partial t} \] means the rate of change in \( y \) with \( t \)
  \[ \frac{\partial y}{\partial x} \] means the rate of change in \( y \) with \( x \)
## Total time derivative

- The *total time derivative* is used to describe the change in a variable in relation to a moving (as opposed to stationary) object.

- Example:

\[
\frac{Dy}{Dt} = \frac{\partial y}{\partial t} + u \frac{\partial y}{\partial x}
\]

*\(u\) is velocity*

---

## Time & space derivatives: review

### Notation:

- \(\frac{dy}{dt}\) = ordinary derivative: \(y\) changes only with \(t\)

- \(\frac{\partial y}{\partial t}\) = partial derivative: change in \(y\) with \(t\) at a fixed point

- \(\frac{Dy}{Dt}\) = total time derivative: change in \(y\) as one moves about (e.g., in a fluid)
Classification of DE (Classes)

- **dimension of unknown:**
  - ordinary differential equation (ODE) – unknown is a function of one variable, e.g. $y(t)$
  - partial differential equation (PDE) – unknown is a function of multiple variables, e.g. $u(t,x,y)$

- **number of equations:**
  - single differential equation, e.g. $y'=y$
  - system of differential equations (coupled), e.g. $y_1'=y_2, \ y_2'=-g$

- **order**
  - nth order DE has nth derivative, and no higher, e.g. $y''=-g$

- **linear & nonlinear:**
  - linear differential equation: all terms linear in unknown and its derivatives.
  - e.g. $x''+ax'+bx+c=0$, $x'=t^2x$ – linear; $x''=1/x$ – nonlinear

Classification of PDEs

\[ a \frac{\partial^2 u}{\partial x^2} + b \frac{\partial^2 u}{\partial x \partial y} + c \frac{\partial^2 u}{\partial y^2} = d \frac{\partial u}{\partial x} + e \frac{\partial u}{\partial y} + f u + g \]

\[
\Delta = b^2 - 4ac
\]

- **elliptic PDE** - $b^2 - 4ac < 0$
  - smooth solutions

- **hyperbolic PDE** - $b^2 - 4ac > 0$
  - solutions with discontinuities

- **parabolic PDE** - $b^2 - 4ac = 0$
**General features of PDEs**

- **Elliptic PDEs**
  - describe processes that have already reached steady state
  - time-independent (e.g. steady-state aquifer flow or heat diffusion).

- **Hyperbolic PDEs**
  - describe time-dependent, conservative physical processes, such as convection, that are not evolving towards steady state (e.g. free surface flow, pipe flow).

- **Parabolic PDEs**
  - describe time-dependent, dissipative physical processes, such as diffusion, that are evolving towards steady state. (e.g. transient aquifer flow).

**What do we need to solve DE?**

- **Boundary values**
- **Problem domain**
- **Initial values**
- **Boundary values**
Boundary and initial conditions

- **Initial value problems (IVP)**
  - derivative with respect to time (either \(\frac{dU}{dt}\), or \(\frac{d^2U}{dt^2}\), etc.) in the equation, it is necessary to know the *initial* value of \(U\) in order to compute its value in the future, within the model domain.
  - initial condition of \(U\) must be given.
  - these problems are also called *initial value problems*.

- **Boundary value problems (BVP)**
  - Initial conditions are not always sufficient to ensure the existence of a unique solutions. Conditions on the boundary of the domain of the problem must be given as well. These conditions are called *boundary conditions*.
Reminder - Basic formulas

What is a derivative?
- Rate of change of a function
- slope of the tangent line to the graph of a function
- best linear approximation to a function

\[ f'(x) = \lim_{\Delta x \to 0} \frac{f(x + \Delta x) - f(x)}{\Delta x} \]

Approx. derivatives

\[ f'(x) \approx \frac{f(x + \Delta x) - f(x)}{\Delta x} \]

\[ \Delta x \text{ is small} \]

Taylor series expansion

\[ f(x + \Delta x) = f(x) + \Delta x f'(x) + \frac{\Delta x^2}{2!} f''(x) + \frac{\Delta x^3}{3!} f'''(x) + \frac{\Delta x^4}{4!} f^{(4)}(x) + \ldots \]

\[ \frac{f(x + \Delta x) - f(x)}{\Delta x} = \frac{1}{\Delta x} \left[ \Delta x f'(x) + \frac{\Delta x^2}{2!} f''(x) + \frac{\Delta x^3}{3!} f'''(x) + \ldots \right] = f'(x) + O(\Delta x) \]

First order accurate in \( \Delta x \)

\[ f(x + \Delta x) - f(x - \Delta x) = \frac{1}{2\Delta x} \left[ 2 \Delta x f'(x) + \frac{2 \Delta x^3}{3!} f'''(x) + \ldots \right] = f'(x) + O(\Delta x^2) \]

Second order accurate in \( \Delta x \)
Reminder - Basic formulas

\[ f(x) - f(x - \Delta x) \]
\[ \Delta x \]
\[ f(x + \Delta x) - f(x) \]
\[ \Delta x \]
\[ f(x + \Delta x) - f(x - \Delta x) \]
\[ 2\Delta x \]

Why study Numerical Methods?

- NO numerical method is completely trouble free in all situations!
  - How should one choose/use an algorithm to get trouble free and accurate answers?
- No numerical method is error free!
  - What level of error/accuracy do one take in the way the problem is solved?
- No numerical method is optimal for all types/forms of an equation!

**IMPORTANT:** In order to solve a physical problem numerically, you must **understand** the behavior of the numerical methods used as well as **the physics of the problem**
Typical difficulties with NM

- Instability

Typical difficulties with NM

- Inconsistency
Typical difficulties with NM

- Inaccurate

Errors

- Errors due to computers:
  - Round-off error
  - Exponent underflow
  - Exponent overflow

- Errors due to numerical methods
  - Truncation error
  - Propagated error

- Measures for errors
  - absolute error
  - relative error
**Errors**

A MATHEMATICAL MODEL

- **Physical System**
  - Nature

- **ERROR 1: Formulation Error**
  - Governing Equations
    - Set of Mathematical Equations
  - Numerical Solution
    - Numbers

- **ERROR 2: Numerical Errors**
  - How various numerical algorithms are derived,
  - How the physics effects the numerics
  - How the numerics effects the physics
  - What are the accuracy/stability properties
  - What are the cost of a method for given level of accuracy (this varies substantially from method to method)

**Hydroinformatician**

- As a developer and user must understand how a numerical method performs for his/her given problem
- Understanding how various numerical algorithms are derived,
- How the physics effects the numerics
- How the numerics effects the physics
- What are the accuracy/stability properties
- What are the cost of a method for given level of accuracy (this varies substantially from method to method)
Properties of Numerical Methods

- **Convergence**
  - numerical scheme solution is *convergent* if it comes closer and closer to the analytical solution of the real ODE/PDE when the time step decreases;

- **LaxTheorem**: 2 conditions need it for convergence
  - Consistency; and
  - Stability

---

- **Consistency**
  - A scheme is *consistent* if it gives a correct approximation of the ODE/PDE as the time/space step is decreased
  - verified using Taylor Series expansion
Properties of Numerical Methods

- **Stability**
  - A scheme is *stable* if any initially finite perturbation remains bounded as time grows.

- **Stability verification**
  - Matrix method
  - Fourier method
  - Domains of dependence

---

Notations

- $U^n = U(t^n)$
- $U_j = U(x_j)$
- $U^n_j = U(t^n, x_j)$
4-Finite difference methods for ODE

The general Initial Value Problem\(^1\)

Given

\[
\frac{du}{dt} = f(u, t)
\]

and initial condition given

*If you know values at time \(n\), find the values at time \(n+1\)*
The general IVP

- All initial value problems are solved by Integrating forward in \( t \)
- There are two main types of integration procedure
  - One-step methods
  - Multi-step methods

General classification

- Numerical scheme
  - a particular discretization of a differential equation

Schemes:
  - Explicit (Euler method) - the value of the variable at time level \( n+1 \) can be computed directly (or explicitly) from the value at time level \( n \).
  - Implicit (Improved Euler) - involves values of the variable at time level \( n+1 \), there is an implicit relationship between the derivative and the variable which has to be computed.
  - Mixed (Crank-Nicholson) - schemes in which the explicit and implicit schemes are combined.
Explicit schemes/ Euler/ One Step Method

Forward difference approximation

\[
\frac{du}{dt} = f(u, t)
\]

\[
u^{n+1} \approx u^n + \Delta t \cdot f(u^n, t^n)
\]

\[
u_{\text{final}} = u_{\text{initial}} + \Delta t \sum_{n=0}^{N-1} f(u^n, t^n)
\]

This is the same as saying:

new value = old value + (slope) x (step size)

These schemes are also called one step methods or \textit{marching in time}, because new value is calculated as one step forward from the old value.

Implicit schemes

Backward difference approximation

\[
\frac{dU}{dt} = f(U, t)
\]

\[
U^{n+1} = f(U^{n+1}, t^{n+1}) \Delta t = U^n
\]

Iteration need it, usually
**Mixed schemes**

Averaging implicit and explicit schemes

- Use this “average” slope to predict $U^{n+1}$

\[
U^{n+1} = U^n + \frac{f(U^{n+1}, t^{n+1}) + f(U^n, t^n)}{2} \Delta t
\]

**Numerical schemes for ODE - Example**

Ground water reservoir

\[
\frac{dh}{dt} = -\alpha h
\]

- exact solution

\[
h = h_0 e^{-\alpha t}
\]
Multi step methods: Midpoint Methods

- A better estimate of the function $U$ would come from evaluating the derivative at the midpoint of the $\Delta t$ interval.

- The problem: we know $t$ at the midpoint but we don’t know the $u(t)$ at the midpoint (yet).

- The solution is to use Euler’s method to estimate $\Delta u$ and then re-estimate $\Delta u$ using the derivatives evaluated halfway along the line segment encompassing the original $\Delta u$.

- This method is called the second order Runge Kutta method, or the midpoint method.

Mixed schemes - Midpoint Methods

\[ u^{n+1} = u^n + (a_1 k_1 + a_2 k_2) \Delta t \]

Where for two point schemes

\[ k_1 = f(u^n, t^n) \]

\[ k_2 = f(u^n + q_{11} k_1 \Delta t, t^n + p_1 \Delta t) \]

$a_1, a_2$ - weighted coefficients

$a_1 = 1 - a_2$;

$a_2 p_1 = \frac{1}{2}$

$a_2 q_{11} = \frac{1}{2}$
Exercise

Apply the Euler and Mid-point methods to the following problem (with known solutions!):

\[
\frac{du}{dt} = f(u, t)
\]

for

1. \(f(u, t) = t\)
2. \(f(u, t) = t^2\)
3. \(f(u, t) = u\)
4. \(f(u, t) = -u\)

In each case, compare the errors in the two schemes after taking the same number (10, say) of identical time steps (start with ). Is the midpoint method noticeably superior to Euler?

Mixed schemes - Multi Step Methods

The midpoint method can be extended by considering other intermediate estimates. The most frequently used variation is the fourth-order Runge-Kutta method which considers one estimate at the initial point, two estimates at the midpoint, and one estimate at a trial endpoint.

This is popular for two reasons

- It is easy to program
- It is stable

However

- It requires more computing time
Runge-Kutta Method - 4th-order

\[ \begin{align*}
    f &= \frac{1}{6} (f_1 + 2f_2 + 2f_3 + f_4) \\
    f_1 &= f(t_i, u_i) \\
    f_2 &= f(t_i + \frac{h}{2}, u_1 + \frac{h}{2} f_1) \\
    f_3 &= f(t_i + \frac{h}{2}, u_2 + \frac{h}{2} f_2) \\
    f_4 &= f(t_i + h, u_3 + h f_3)
\end{align*} \]

Runge-Kutta - 4 point scheme

\[ u^{n+1} = u^n + (a_1 k_1 + a_2 k_2 + a_3 k_3 + a_4 k_4) \Delta t \]

Where

- \( f(u, t) \) - the derivative at \( (u, t) \)
- \( k_1 = f(u, t) \)
- \( k_2 = f(u + k_1 \Delta t / 2, t + \Delta t / 2) \)
- \( k_3 = f(u + k_2 \Delta t / 2, t + \Delta t / 2) \)
- \( k_4 = f(u + k_3 \Delta t, t + \Delta t) \)
Runge-Kutta 4th order - solution

\( f(u,t) \) – the derivative at \((u,t)\)

\[
k_1 = f(u,t)
\]

\[
k_2 = f(u + k_1 \Delta t / 2, t + \Delta t / 2)
\]

\[
k_3 = f(u + k_2 \Delta t / 2, t + \Delta t / 2)
\]

\[
k_4 = f(u + k_3 \Delta t, t + \Delta t)
\]

slope = \( k_1 \)

\( (t^n + k_1 \Delta t / 2, u^n + k_1 \Delta t / 2) \)

\( (k_3 \Delta t / 2) \)

\( u \)

\( \Delta t / 2 \)

\( \Delta t / 2 \)

Runge-Kutta versus Euler-example

Solve the equation below using both Euler and Runge Kutta methods

\[
\frac{du}{dt} = 6u - 0.007u^2
\]

Problem: estimate the slope to calculate \( \Delta u \)

\( \Delta t = 0.5 \)

\( \Delta u \)

\( t \)

\( u \)
Step 1: Evaluate slope at current value of state variable.

\[ u_0 = 10 \]
\[ k_1 = \frac{du}{dt} \text{ at } u_0 \]
\[ k_1 = 6 \times 10^{-10} - 0.007 \times (10)^2 \]
\[ \Delta u = k_1 \Delta t \]
\[ u_{est} = u_0 + \Delta u \]
Step 2: Calculate $u_1$ at $t + \Delta t/2$ using $k_1$.

Evaluate slope at $u_1$.

\[ u_1 = u_0 + k_1 \cdot \Delta t/2 \]
\[ u_1 = 24.82 \]

\[ k_2 = \text{du/dt at } u_1 \]
\[ k_2 = 6 \cdot 24.8 - 0.007 \cdot (24.8)^2 \]
\[ k_2 = 144.63 \]

\[ u_1 = u_0 + k_1 \cdot \Delta t/2 \]
\[ u_1 = 24.82 \]

Step 3: Calculate $u_2$ at $t + \Delta t/2$ using $k_2$.

Evaluate slope at $u_2$.

\[ u_2 = u_0 + k_2 \cdot \Delta t/2 \]
\[ u_2 = 46.2 \]

\[ k_3 = \text{du/dt at } u_2 \]
\[ k_3 = 6 \cdot 46.2 - 0.007 \cdot (46.2)^2 \]
\[ k_3 = 263.0 \]

\[ u_2 = u_0 + k_2 \cdot \Delta t/2 \]
\[ u_2 = 46.2 \]
Step 4: Calculate \( u_3 \) at \( t + \Delta t \) using \( k_3 \).
Evaluate slope at \( u_3 \).

\[
\begin{align*}
  u_3 &= u_0 + k_3 \cdot \Delta t \\
  u_3 &= 141.0 \\
  k_4 &= \text{du/dt at } u_2 \\
  k_4 &= 6 \cdot 141.0 - 0.007 \cdot (141.0)^2 \\
  k_4 &= 706.9
\end{align*}
\]

\[
\begin{align*}
  k_3 &= \text{slope 4} \\
  k_4 &= \text{slope 4}
\end{align*}
\]

\[
\begin{align*}
  k_3 &= \text{slope 4} \\
  k_3 &= \text{slope 4}
\end{align*}
\]

\[
\begin{align*}
  u_2 &= u_2 \\
  u_2 &= u_2 \\
  \Delta t &= 0.5
\end{align*}
\]

Step 5: Calculate weighted slope.
Use weighted slope to estimate \( u \) at \( t + \Delta t \)

\[
\text{weighted slope} = \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4)
\]

\[
\begin{align*}
  u_{n+1} &= u_n + \Delta t \cdot \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4)
\end{align*}
\]
Conclusions

- 4th order Runge-Kutta offers substantial improvement over Euler's.
- Both techniques provide estimates, not "true" values.
- The accuracy of the estimate depends on the size of the step used in the algorithm.

Numerical schemes for ODE - Example

Ground water reservoir

\[
\frac{dh}{dt} = -\alpha h
\]

- exact solution

\[ h = h_0 e^{-\alpha t} \]
Numerical schemes for ODE - Example
Solution methods : 1. Explicit (Euler)

The right hand side is determined at current time ($h^n$)

\[
\frac{h^{n+1} - h^n}{\Delta t} = -\alpha h^n
\]

\[
h^{n+1} = h^n (1 - \alpha \Delta t)
\]

\[
\frac{h^{n+1}}{h^n} = (1 - \alpha \Delta t)
\]

Numerical schemes for ODE - Example
Solution methods : 2. Implicit

The right hand side is determined at future time ($h^{n+1}$)

\[
\frac{h^{n+1} - h^n}{\Delta t} = -\alpha h^{n+1}
\]

\[
\frac{u^{n+1}}{u^n} = \frac{1}{(1 + \alpha \Delta t)}
\]
Numerical schemes for ODE - Example

\[ \frac{dh}{dt} = -\alpha h \]

Analytic solution: \( h = h_o e^{-\alpha t} \)

Euler scheme: \( h^{n+1} = (1 - \alpha \Delta t) h^n \)

Implicit scheme: \( h^{n+1} = \frac{h^n}{1 + \alpha \Delta t} \)

The BIG question

\[ \approx \neq \approx \]
The **BIG** question

- If equations are approximation then it should:
  - behave in the same way as the function to which it is an approximation, and
  - become a better approximation as the time-step is reduced.

### Explicit vs Implicit

**Exact solution**

\[
\frac{h^{n+1}}{h^n} = \frac{\exp(-\alpha(n+1)\Delta t)}{\exp(-\alpha n\Delta t)} = \exp(-\alpha \Delta t)
\]

**Comparison of different solution methods**

- Exact solution lies between explicit and implicit solutions.
Three different solution methods - comparison

Stability, Consistency, Convergence

Lax Theorem:
Consistency + stability $\iff$ convergence

*This is why you can trust your models!*

**Consistency**: The scheme approximates the ODE more correctly as the discretisation is refined.

**Stability**: Any initial perturbation remains bounded.

**Convergence**: The numerical solution becomes closer to the real one as the discretisation is refined.
**The BIG question**

- If equations are approximation then it should:
  - behave in the same way as the function to which it is an approximation, and
  - become a better approximation as the time-step is reduced.

**Stability, Consistency, Convergence**

Convergence - Explicit method

\[ h^n \rightarrow h^0 e^{-\alpha t} \]

\[
\begin{align*}
    h^1 &= h^0 (1 - \alpha \Delta t) \\
    h^2 &= h^1 (1 - \alpha \Delta t) = h^0 (1 - \alpha \Delta t)^2 \\
    \vdots \\
    h^n &= h^0 (1 - \alpha \Delta t)^n
\end{align*}
\]

\[
h^n = \left[ 1 + \left( -\frac{\alpha t}{n} \right) \right]^n h^0
\]
Stability, Consistency, **Convergence**

Convergence - Explicit method (cont)

\[
h^n = h^0 \left[ 1^n + 1^{n-1} \left( -\frac{\alpha \cdot t}{n} \right)^n + 1^{n-2} \left( -\frac{\alpha \cdot t}{n} \right)^2 \frac{n}{1} + \ldots \right]
\]

where

\[
\binom{n}{r} = \frac{n!}{(n-r)!r!}
\]

\[
\frac{n!}{(n-r)!r!} = n(n-1)\ldots(n-r) \to n^r
\]

Back to the equation

\[
h^n \to h^0 \left[ 1^n + \left( -\frac{\alpha \cdot t}{n} \right)^n \right]
\]

\[
\to h^0 \left[ 1^n + \left( -\frac{\alpha \cdot t}{1} \right)^1 + \left( -\frac{\alpha \cdot t}{1} \right)^2 \frac{1}{2!} + \ldots \right] \to h^0 e^{-\alpha t}
\]

Taylor series for \( e^{-\alpha t} \)
Stability, Consistency, Convergence

Stability condition

\[ \frac{|h^{n+1}|}{h^n} \leq 1 \implies -1 \leq \frac{h^{n+1}}{h^n} \leq 1 \]

Stability - Explicit schemes

\[-1 \leq 1 - \alpha \Delta t \leq 1 \implies \begin{cases} -\alpha \Delta t \leq 0 & \Rightarrow \Delta t_{\text{max}} = \frac{2}{\alpha} \\ \alpha \Delta t \leq 2 & \Rightarrow \Delta t_{\text{max}} = \frac{2}{\alpha} \end{cases} \]

Example

Take \( \alpha = 1 \) \( \Rightarrow \Delta t_{\text{max}} = \frac{2}{\alpha} = 2 \)

and \( h^0 = 1 \)

a) \( 0 < \Delta t < 1 \)
b) $1 \leq dt < 2$

Stability, Consistency, Convergence

$d_t = 1.25$

$d_t = 1.5$

$d_t = 1.5$

$\Rightarrow$ Stable But Oscillatory Approximations

---

c) $2 < dt$

Stability, Consistency, Convergence

$d_t = 4.5$

$d_t = 3$

$d_t = 2.5$

$\Rightarrow$ Unstable (i.e. Bad) Approximations
Stability, Consistency, Convergence

Stability - Explicit schemes

\[ 0 \leq 1 - \alpha \Delta t \leq 1 \quad \text{or} \quad 0 \leq \Delta t \leq \frac{1}{\alpha} \]
Stability, Consistency, Convergence

Consistency - explicit schemes

\[ \frac{dh}{dt} + \alpha h = -\frac{\Delta t}{2} \frac{d^2h}{dt^2} + O(\Delta t^2) \]

Truncation error - first order accurate

Implicit schemes
- convergence - FOR YOU TO TEST
- stability - unconditionally stable

Mixed schemes
- unconditionally stable
Amplification factor

$ U^{n+1} / U^n $ is called the numerical amplification factor.

Amplitude portrait

<table>
<thead>
<tr>
<th>Method</th>
<th>Euler</th>
<th>Improved Euler</th>
</tr>
</thead>
</table>

Region of instability

Region of instability
4- Finite difference methods for ODE

Quick summary

What you should remember

- ODEs can be solved numerically by replacing the d/dt operator by a difference of the considered function at two time levels.
- When the derivative is estimated using the values of the function at time step level n, the scheme is said to be explicit. When the derivative is estimated using the values at time n+1, the scheme is said to be implicit.
- The discretized ODE is consistent to the real one if it comes closer to the real ODE as dt (or dx) comes closer to zero.
What you should remember

- The difference between the discretised ODE and the real one is called the truncation error. It is obtained by substituting Taylor series expansions into the discretisation.

- The solution is said to be stable if initial perturbations in the numerical solution remain bounded.

- The stability of the solution is investigated by calculating the value of the amplification factor. If $|A^N|$ is smaller than unity, the solution is stable.

What you should remember

- In general, explicit schemes are not stable for all values of $\Delta t$ (or $\Delta x$).

- Implicit schemes are always stable.

- The numerical solution is said to converge to the real one if it comes closer to it as $\Delta t$ (or $\Delta x$) comes closer to 0.

- Consistency and stability of a scheme are necessary and sufficient conditions to the convergence of the solution.