- Linear systems
- Non-linear systems
- Systems w/ derivatives

**ODEs (d. in one direction)**  
**PDEs (d. in multiple dir.)**

**IVP**  
**BVP**

### Applications:

<table>
<thead>
<tr>
<th>ODE/IVP</th>
<th>BVP</th>
</tr>
</thead>
</table>
| - Population dynamics  
  \[ y_1 = -\alpha y_2 \\text{ prey} \]  
  \[ y_2 = \beta y_2 \\text{ predator} \]  
  - Chemical reactions  
  - Equations of motion | - Bridge load  
  - Pollutant concentration  
  - Temperature  
  - Bed  
  - Radiator |
3. **Initial Value Problems**

\[ y : [0, T] \rightarrow \mathbb{R}^n \] desired solution

\[ f(t, y, y', y'', \ldots, y^{(k)}) = 0 \quad \text{(implicit)} \]

\[ y^{(k)}(x) = f(t, y, y', \ldots, y^{(k-1)}) \quad \text{(explicit)} \]

\[ \text{solvable as is?} \]

Consider simple example: \( y' = \alpha y \)

\[ \rightarrow \text{need initial values. how many?} \]

\[ y(0) = y_0 \]
\[ y'(0) = y_1 \]
\[ \vdots \]
\[ y^{(k-1)}(0) = y_{k-1} \]

\[ \text{could swap some derivatives for conditions at other end:} \]

\[ y(T) = \tilde{y}_0 \]

\[ \rightarrow \text{BVP} \]

\[ \text{ODE + Initial values = IVP} \]

\[ \text{Can always reduce to first order:} \]

\[ y''(t) = f(y) \quad \rightarrow \quad \begin{pmatrix} y_1' \\ y_2' \end{pmatrix} = \begin{pmatrix} y_2 \\ f(y, t) \end{pmatrix} \]

\[ \rightarrow y_1''(t) = (y_1'(t))' = y_2'(t) = f(y, t) \]
Properties

autonomous: \( f \) does not depend on \( t \)

\[ \text{can always make autonomous using extra variable: } y_0'(t) = 1, y_0(0) = 0 \]

\( \Rightarrow \) will omit explicit \( t \) dependence

linear: \( f(t) = A(t)x + b \)

linear homogeneous: \( f(t) = A(t)x \)

constant-coefficient: \( A(t) = A \)
9.1 Existence, Uniqueness, Conditioning

Consider:
\[
\begin{cases}
    y'(t) = f(y) \\
y(t_0) = y_0
\end{cases}
\]
\[
\begin{cases}
    \hat{y}'(t) = f(\hat{y}) \\
\hat{y}(t_0) = \hat{y}_0
\end{cases}
\]

Assume: \( f \) Lipshitz continuous ("bounded slope")

\[ \| f(y) - f(\hat{y}) \| \leq L \| y - \hat{y} \| \]

\( L \) Lipshitz constant

If so: ("Picard-Lindelöf theorem")

- there exists a solution of \( \mathcal{O} \) in a neighborhood of \( t_0 \)

- \( \| y(t) - \hat{y}(t) \| \leq e^{Lt_0} \| y_0 - \hat{y}_0 \| \)

What does this mean for uniqueness?
Conditioning! (in ODE-speak: "stability")

For us: "...of a method"

For them: used for both ODEs and methods

ODE stable iff solution continuously dependent on initial condition

In c-b speak:

\[
\text{For all } \epsilon > 0 \text{ there exists a } \delta > 0 \text{ such that } \\
\| y(t) - y(t_0) \| < \delta \Rightarrow \| y(t) - y(t_0) \| \leq \epsilon \text{ for all } t \geq t_0.
\]

ODE asymptotically stable iff \( \lim_{t \to \infty} \| y(t) - y(t_0) \| = 0 \)
Example I  \[ y'(t) = \lambda y \quad \lambda = a + ib \]
\[ y(0) = y_0 \]

Solution:  \[ y(t) = y_0 e^{\lambda t} = y_0 e^{at} e^{ib} \quad \Rightarrow \quad |y(t)| = |y_0 e^{at}| \]

Stable? 
\[ a = \text{Re} \lambda \geq 0 \]

\[ a = \text{Re} \lambda \leq 0 \]

Example II \[ \ddot{y}'(t) = A \dot{y}(t) \]
\[ \ddot{y}(0) = \ddot{y}_0 \]

Assume \( A \) is diagonalizable \( V'AV \)
\[ \dot{\hat{w}} = V'\ddot{y} \]
\[ \dot{\hat{w}}(t) = V'\ddot{y}(t) = V' A \dot{y}(t) = V' A V^{-1} \hat{y}(t) \]
\[ = D \hat{\alpha}(t) \]
\[ \hat{w}_0 = V'\ddot{y}_0 \]

\[ \Rightarrow \hat{y}(t) = V\hat{w}(t) \]

\[ \text{Stable what?} \]

WSZ9p1
lec32

- Exam on Wed.
- Finish WS19
5.2 Numerical Methods (pt. 1)

Discrete times: \( t_0, t_1, \ldots, t_i = t_{i-1} + h \) (for now)

Discrete function values: \( y_k = y(t_k) \)

IVP:

\[
\begin{align*}
y'(t) &= f(y(t)) \\
y(t_0) &= y_0
\end{align*}
\]

Integral equation:

\[
y(t) = y_0 + \int_{t_0}^{t} f(y(x)) \, dx
\]

Idea: Trapezoid rule or \( \square \) Which? (for solvability)

\[
y_{mn} = y_m + h f(t_n, y_n) \quad \text{(Euler's method)}
\]

Demo
9.3 Accuracy and Stability

Local error: $e_k = y_x - u_x(t_x)$ (where $u_x$ solves the ODE with IV $u(t_0) = y_x$)

Global error: $y(t_x) - y_x$

- Is global error $\sum$ (local errors)?
- Interest analogy.
- Local error much easier to estimate!

Time integrator of order $p \Rightarrow e_k = O(h^p)$

Local error per length-1 step: $\frac{1}{h} \cdot O(h^p)$ (assuming limited "accidental" of propagated error)
Stability of a method

Instability can be caused:

- by ODE
- by the method

In Forward Euler, for \( y'(t) = \lambda y(t) \):

\[
y_k = y_{k-1} + h \lambda y_{k-1}
\]

\[
= (1 + h \lambda) y_{k-1}
\]

growth factor

Stable if \( |1 + h \lambda| < 1 \)

\[
y_k = (1+h\lambda)^k y_0
\]

Stability region:
Stability in the nonlinear case

For Euler:

\[ y_{k+1} = y_k + hf(y_k) \]

For Taylor:

\[ y(t_{k+1}) = y(t_k) + hf(y(t_k)) + O(h^2) \]

\[ \Delta = e_{k+1} = \frac{y_{k+1} - y(t_{k+1})}{h} = \frac{(y_k - y(t_k)) + hf(y_k) - f(y(t_k)) + O(h^2)}{h} \]

\[ e_{k+1} = e_k 
\]

\[ e_{k+1} = e_k (1 + h^T f) + le_k \]

Errors do not grow if \( g(1 + h^T f) \leq 1 \).

Need intermediate result: a (not "the") mean value theorem

Assume \( f \) differentiable:

\[ \int_a^b (y_x + (1-x)y(t_k)) \, dx \]

\[- \int_a^b \frac{\partial}{\partial x} (f(x)) \, dx \]

\[ = [f(x)]_a^b = f(y) - f(t_k) \quad (\text{A bug in book}) \]

Then:

\[ f(y_{k+1}) - f(y(t_k)) = \int_a^b f(x y_{k+1} + (1-x) y(t_k)) \, dx \cdot (y_{k+1} - y(t_k)) \]
9.4 Numerical Methods (pt. II)

0 Obtained Euler method by throwing left rectangle rule at:

\[ y(t_0) = y_0 + \int_{t_0}^{t} f(y(t)) \, dt \]

\[ y_{n+1} = y_n + h \cdot f(y_n) \]

0 What if we had used right rectangle rule?

\[ y_{n+1} = y_n + h \cdot f(y_{n+1}) \]

\[ \text{Backward Euler method (earlier Euler method: Forward Euler by contrast)} \]

\[ \rightarrow \text{need to solve equation } \rightarrow \text{ "implicit method"} \]

(\text{not implicit: } " \text{explicit method}"")

Example: \[ y'(t) = \lambda y(t) \]

\[ y_{n+1} = y_n + h \cdot \lambda \cdot y_{n+1} \rightarrow (1 - h \cdot \lambda) y_{n+1} = y_n \]

\[ y_{n+1} = \frac{1}{1 - h \cdot \lambda} \cdot y_n \]

\[ \text{Amplification factor} \]

\[ \text{Stable here??} \]

(yes, but not accurate)

Generically (not just for implicit):

Two restrictions on time step:

- accuracy
- stability
55) Stiffness

- Demo

- Two time scales:
  - fast
  - slow

- $y'(t) = f(y(t))$ stiff if $J_f$ has eigenvalues of very different magnitude

- Why not just "small" or "large"?

- What's the problem with applying explicit methods to stiff problems?

- Express this as conflict between stability and accuracy

- Can an implicit method take arbitrarily large time steps?

- Example: acoustic vs. flow

WS 30pl
9.6 Numerical Methods (Part III)

Predictor-corrector methods

Idea: Obtain immediate result, improve it (with same or different method). E.g.:

- predict with 4th Euler \( \tilde{y}_{n+1} = y_n + hf(y_n) \)

- correct with trapezoidal \( y_{n+1} = y_n + \frac{h}{2}(f(y) + f(\tilde{y}_{n+1})) \)

\( \uparrow \) Heun's method (2nd order accurate)

More general than P-C: Methods that evaluate \( f \) multiple times per step:

- single-step
- multi-stage
  - Runge-kutta

\[ \text{Wikipedia: Runge-Kutta:} \]

- point out RK4
- equivalence to Simpson's rule (if no \( y \) dependency)

- point out "stages"

- implicit variants also exist
Systematic view of Runge-Kutta methods: Butcher tableau

\[
\begin{align*}
  r_i &= f(t_i + c_i h, y_i + a_{i1} r_1 + \cdots + a_{iN} r_N) \\
  &\vdots \\
  r_N &= f(t_N + c_N h, y_N + a_{N1} r_1 + \cdots + a_{NN} r_N)
\end{align*}
\]

\[
\begin{array}{c|cccc}
  c_i & a_{i1} & \cdots & a_{iN} \\
  \vdots & \ddots & \ddots & \vdots \\
  c_N & a_{N1} & \cdots & a_{NN} \\
  \hline
  b_1 & \cdots & b_N \\
\end{array}
\]

\( \sigma = \text{number of "stages"} \)

When is this explicit? implicit? "diagonally implicit?"

Embedded Runge-Kutta method

\[
\begin{align*}
  y_{n+1} &= y_n + b_1 r_1 + \cdots + b_\sigma r_\sigma \\
  \tilde{y}_{n+1} &= y_n + \tilde{b}_1 r_1 + \cdots + \tilde{b}_\sigma r_\sigma
\end{align*}
\]

estimates of different order, to be used for error estimation/adaptation of \( h \)

Recap: Heun's method

RK scheme:

\[
\begin{align*}
  \tilde{y}_{n+1} &= y_n + hf(y_n) \\
  r_1 &= f(t_n, y_n) \\
  r_2 &= f(t_n + h, y_n + hr_1) \\
  y_{n+1} &= y_n + \frac{1}{2}(f(y_n) + f(\tilde{y}_{n+1}))
\end{align*}
\]

Wikipedia
Another idea: Instead of computing stage values, use history

\[ y_{k+1} = \sum_{i=1}^{M} \alpha_i \cdot y_{k+1-i} + h \sum_{i=1}^{N} \beta_i \cdot f(y_{k+1-i}) \]

Known as:

- single-stage
- multi-step
- Adams-Bashforth (if \( M = 1 \))
- Backward Differentiating Formulas (BDF) (if \( N = 1 \))

- also exist in implicit variants
- What if there is no history?

Demo: Stability Regions

WS31