Numerical Treatment of ODE

- remember population dynamics: one or some ODE
  - there as an initial value problem; starting point given
  - sometimes also as a boundary value problem; starting and final point given (think of a space shuttle's trajectory, e.g.)
- prototypes of an initial value problem (IVP):
  \[ \dot{y}(t) = f(t, y(t)), \quad y(a) = y_a, \quad t \geq a \]
  \[ \dot{y}_i(t) = f(t, y_1(t), ..., y_n(t)), \quad y_i(a) = y_{i,a}, \quad t \geq a, \quad i = 1, ..., n \]
- if \( f \) depends on \( t \) only: simple integration or quadrature
- standard approach for IVP: finite difference approximation (difference quotient instead of derivative)
  \[ y(a + \delta t) \approx y(a) + \delta t \cdot f(t, y(a)) \]
  \[ y_{k+1} = y_k + \delta t \cdot f(t_k, y_k), \quad t_k = a + k\delta t, \quad k = 0, 1, ... \]

---

Euler's Method, Discretization Error

- this is the simplest strategy and called *Euler's method*
- other derivation: truncated Taylor expansion of \( y(t) \)
  \[ y(t_{k+1}) = y(t_k) + (t_{k+1} - t_k) \dot{y}(t_k) + R = y(t_k) + (t_{k+1} - t_k) f(t_k, y_k) \]
- local discretization error: local influence of using differences instead of derivatives; here:
  \[ l(\delta t) = \max_{[a,b]} \left\{ \left| y(t + \delta t) - y(t) / \delta t - f(t, y(t)) \right| \right\} \]
- global discretization error: maximum error of all computed discrete approximations:
  \[ e(\delta t) = \max_{[a,b]} \left\{ \left| y_k - y(t_k) \right| \right\} \]
- consistency: \( l(\delta t) \to 0 \) for \( \delta t \to 0 \)
- convergence (stronger): \( e(\delta t) \to 0 \) for \( \delta t \to 0 \)
Order of Convergence

- Euler (some restrictions with respect to y and f):
  - consistent of first order: \( t(\delta t) = O(\delta t) \)
  - convergent of first order: \( e(\delta t) = O(\delta t) \)
- There are methods that are consistent but do not converge!
- Look for higher-order methods (faster convergence)
  - start from Taylor expansion: leads to complicated formulas (higher derivatives of f)
  - use additional evaluations of f: Runge-Kutta-type methods
- Simplest representative: method of Heun
  \[
  y_{k+1} = y_k + \frac{\delta t}{2} \left( f(t_k, y_k) + f(t_{k+1}, y_k + \delta t \cdot f(t_k, y_k)) \right)
  \]
- Both consistent and convergent of second order

Method of Runge and Kutta

- Most famous representative: Runge-Kutta method
  \[
  y_{k+1} = y_k + \frac{\delta t}{6} \left( T_1 + 2T_2 + 2T_3 + T_4 \right),
  \]
  \[
  T_1 = f(t_k, y_k),
  \]
  \[
  T_2 = f\left(t_k + \frac{\delta t}{2}, y_k + \frac{\delta t}{2} T_1\right),
  \]
  \[
  T_3 = f\left(t_k + \frac{\delta t}{2}, y_k + \frac{\delta t}{2} T_2\right),
  \]
  \[
  T_4 = f(t_{k+1}, y_k + \delta t T_3)
  \]
- Consistent and convergent of fourth order
- Euler/Heun/Runge-Kutta correspond to trapezoidal/ Simpson quadrature!
Alternative: Multistep Methods

- Runge-Kutta-type methods are expensive: many evaluations of \( f \) (sometimes not given in closed form)
- Different way to get higher order by profiting from history: Adams-Bashforth-type or multistep methods
  - Prominent representative: second-order method
    \[
    y_{k+1} = y_k + \frac{\delta t}{2} \left( 3 f(t_k, y_k) - f(t_{k-1}, y_{k-1}) \right)
    \]
  - General form: take polynomial \( P(t) \) interpolating \( f \) in the discrete points of time
    \[
    y_{k+1} = y_k + \int_{t_k}^{t_{k+1}} y(t) \, dt = y_k + \int_{t_k}^{t_{k+1}} f(t, y(t)) \, dt \pm \int_{t_k}^{t_{k+1}} P(t) \, dt
    \]
  - \( p=1 \): Euler; \( p=2 \): above method; generally: order \( p \)
  - Start: no/not enough predecessors available; hence modify!

Implicit Methods

- All schemes mentioned so far are explicit ones: The rule shows a direct way to do another time step.
- Now: use the new value \( y_{k+1} \) on the right-hand side, too
- This leads us to Adams-Moulton multistep schemes:
  - Use interpolation and previous values as with Adams-Bashforth
  - Second order variant:
    \[
    y_{k+1} = y_k + \frac{\delta t}{2} (f_k + f_{k+1})
    \]
  - Fourth order variant:
    \[
    y_{k+1} = y_k + \frac{\delta t}{24} (f_k - 5 f_{k-1} + 19 f_k + 9 f_{k+1})
    \]
- How to get \( y_{k+1} \) in the implicit case?
  - Straightforward way: solve the (generally nonlinear) equation
  - Easier (and widespread): predictor-corrector approach
Ill-Conditioned Problems

- small changes in input entail completely different results
- Numerical treatment of such problems is difficult!
- an example:
  - consider the ODE \( \dot{y}(t) - N \cdot y(t) - (N + 1) \cdot y(t) = 0 \)
  - initial conditions: \( y(0) = 1, \quad \dot{y}(0) = -1 \)
  - exact solution: \( y(t) = e^{-t} \)
- slight change in initial condition:
  - new value of \( y \) in \( t=0 \): \( y_e(0) = 1 + \varepsilon \)
  - resulting new solution: \( y_e(t) = \left(1 + \frac{N+1}{N+2} \varepsilon\right) e^{-t} + \frac{\varepsilon}{N+2} e^{(N+1)t} \)
  - arbitrarily small change leads to completely different result: \( t \to \infty \)
- risk: non-precise input, round-off errors, ...

\[ \text{Introduction to Scientific Computing} \]
\[ \text{Lesson 6: Discretizing ODE} \]
\[ \text{Slide 7} \]

Stability

- consider another IVP: \( \dot{y}(t) = -2y(t) + 1, \quad y(0) = 1 \)
- exact solution: \( y(t) = \left(e^{-2t} + 1\right)/2 \)
- well-conditioned: \( y_e(0) = 1 + \varepsilon \Rightarrow y_e(t) - y(t) = \varepsilon e^{-2t} \)
- use the midpoint rule:
  \( y_{k+1} = y_{k-1} + 2\delta t \cdot f_k \)
- 2-step rule: start with initial value and the exact \( y(\delta t) \)
  - time step \( \delta t = 1.0 \Rightarrow y_9 = -4945.5, \quad y_{10} = 20953.9 \)
  - time step \( \delta t = 0.1 \Rightarrow y_{70} = -1725.3, \quad y_{80} = 2105.7 \)
  - time step \( \delta t = 0.01 \Rightarrow y_{999} = -154.6, \quad y_{1000} = 158.7 \)
- midpoint rule is second-order consistent, but does not converge here: oscillations or unstable behaviour
- there are stability conditions; generally:
  consistency + stability = convergence

\[ \text{Introduction to Scientific Computing} \]
\[ \text{Lesson 6: Discretizing ODE} \]
\[ \text{Slide 8} \]
Stiffness

- for another phenomenon, consider the IVP
  \[ \dot{y}(t) = -1000y(t) + 1000, \quad y(0) = y_0 = 2 \]
- exact solution: \[ y(t) = e^{-1000t} + 1 \]
- problem well-conditioned
- explicit Euler: stable (as all explicit 1-step methods or all Adams-Bashforth or all s-step Adams-Moulton methods for s>1 are)
  \[ y_{k+1} = y_k + \delta t(-1000y_k + 1000) = (1-1000\delta t)y_k + 1000\delta t \]
  \[ = (1-1000\delta t)^y_{k+1} + 1 \]
- oscillations and divergence for \( \delta t > 0.002 \)
- Why that? Consistency and stability are asymptotic terms. Remedy: implicit methods (try implicit Euler)!

Boundary Value Problems: Outlook

- example: \( \ddot{y} = f(t, y, \dot{y}), \quad t_a \leq t \leq t_b, \quad y(t_a) = y_a, \quad y(t_b) = y_b \)
- special case: \( \ddot{y}(t) = a(t)\dot{y}(t) + b(t)y(t) + c(t), \) same b.c.
- a vanishing and b positive: BVP has unique solution
- discrete grid: \( \delta t = (t_b - t_a)/n, \quad t_0 = t_a, \quad t_n = t_b, \quad t_i = t_a + i\delta t \)
- finite difference approximation for second derivative:
  \[ \ddot{y}(t) \approx \frac{y(t+\delta t) - 2y(t) + y(t-\delta t)}{\delta t^2} \]
- discrete analogon to ODE in each grid point:
  \( \delta t^2 \cdot (y_{i+1} - 2y_i + y_{i-1}) - b_iy_i = c_i, \quad i = 1, \ldots, n-1 \)
- tridiagonal system of linear equations
- alternative: shooting methods (reduction to IVPs)