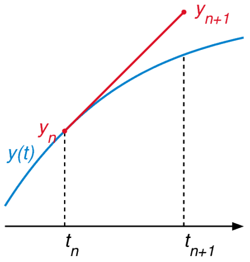
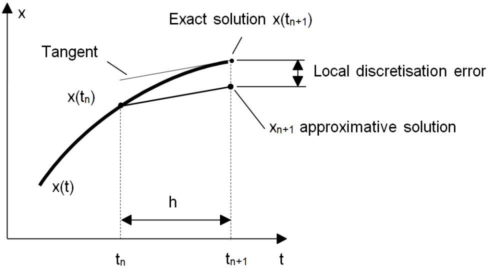
Backward Euler Method

The backward Euler is method is similar to forward Euler, but one uses the slope at the future time step t_{i+1} rather than time t_i. Below: forward vs backward Euler.





See notes.

Backward Euler Method

Definition

The **backward Euler method** approximately solves and IVP by $y_0 = y(t_0)$ and

$$y_{i+1} = y_i + hf(t_{i+1}, y_{i+1}), \quad i = 0, 1, 2, \ldots, N$$

with $h = (T - t_0)/N$ and $t_i = t_0 + hi$.

Note that the y_{i+1} is a function of y_{1+1}, and so it must be found with a root-finding algorithm.

Yes it is slower, but it has a benefit ... (good for a family of problems called "stiff" equations.)

(See notes for derivation.)

Fixed Point Iteration

There are many ways to find a value x such that $x = g(x)$. A common one is fixed point iteration.

Definition

Let r denote a fixed point of $g(x)$ so that $g(x^*) = x^*$. **Fixed point iteration** constructs a converging sequence to x^* given an initial guess x_0 via the iteration

$$x(t+1) = g(x(t)).$$

Theorem

The iterations x_t converge to x^* iff $|\frac{dg}{dx}| < 1$ at $x = x^*$.

(See code.)

Backward Euler Method

Example. Solve $f(t, y) = 2y$ and $y(0) = 1$ with backward Euler.

(See notes and code.)

Backward Euler Errors

The error is similar to that of forward Euler

Theorem

The **local truncation error** e_i for the backward Euler method is second order, $e_i = \mathcal{O}(h^2)$, and if $|y''(t)| \leq M$ for some constant M, then $e_i \leq ch^2$ for some c

Theorem

Considering an IVP with a Lipschitz function with Lipschitz constant L, the **global error** E_i for the backward Euler method is first order, $E_i = \mathcal{O}(h)$, then

$$E_i \leq \frac{ch}{L}(e^{Lhi} - 1)$$

(See notes for proof.)

Backward Euler for Higher-order ODEs

Example: Revisit the example

$$y'''(t) = \frac{d^3y}{dt^3}(t) = y(t) + 2y'(t) + y''(t)$$

with initial conditions $y(0) = 1$, $y'(0) = 1$ and $y''(0) = 0$

(See notes.)

Implicit vs Explicit Methods

Definition

An IVP solver is called an **explicit method** if the next iteration only depends on past iterations

$$y_{i+1} = g(t, y_i, y_{i-1}, y_{i-2}, \dots)$$

Definition

An IVP solver is called an **implicit method** if the next iteration is a function of itself as well as past iterations

$$\underline{y_{i+1}} = g(t, \underline{y_{i+1}}, y_i, y_{i-1}, y_{i-2}, \dots)$$

See examples.

Implicit vs Explicit Methods

Definition

An IVP is called a **stiff** if the ODE has multiple timescales.

Implicit methods are better at solving "stiff" equations. Explicit methods can solve a stiff problem only if the time step *h* is very, very small.