Simulation – Ordinal Differential Equations

ROBERTO ALEJANDRO CÁRDENAS OVANDO

Outline

- Ordinal Differential Equations (ODE)
- One step methods
  - Euler method
  - Heun method
- Taylor method
- Modified Euler method
- Runge-Kutta methods
- Coupled-Systems numerical approximation

Ordinal Differential Equations (ODE)

- First order ODE have the form:

$$y' = \frac{dy}{dx} = f(x, y)$$

ODE – Example

- Numeric approximation to compute the distance:

new value = present value + slope · size

$$y_i = y_{i-1} + m \cdot \Delta x$$

ODE – Generalization

$$m = \frac{\Delta y}{\Delta x} \qquad h = \Delta x \qquad \Delta y = y_{i+1} - y_i$$

$$\therefore \quad y_{i+1} = y_i + m \cdot h$$

ODE – One step methods

- All the methods that use only the slope to extrapolate are called one step methods.
- The two most common one step methods are:
  - Euler method
  - Heun method

Only this part of the TS

$$\boxed{T_n(f)(x) = f(x_0)(x-x_0)^0 + f'(x_0)(x-x_0)^1} + \frac{f''(x_0)(x-x_0)^2}{2}$$

Euler Method

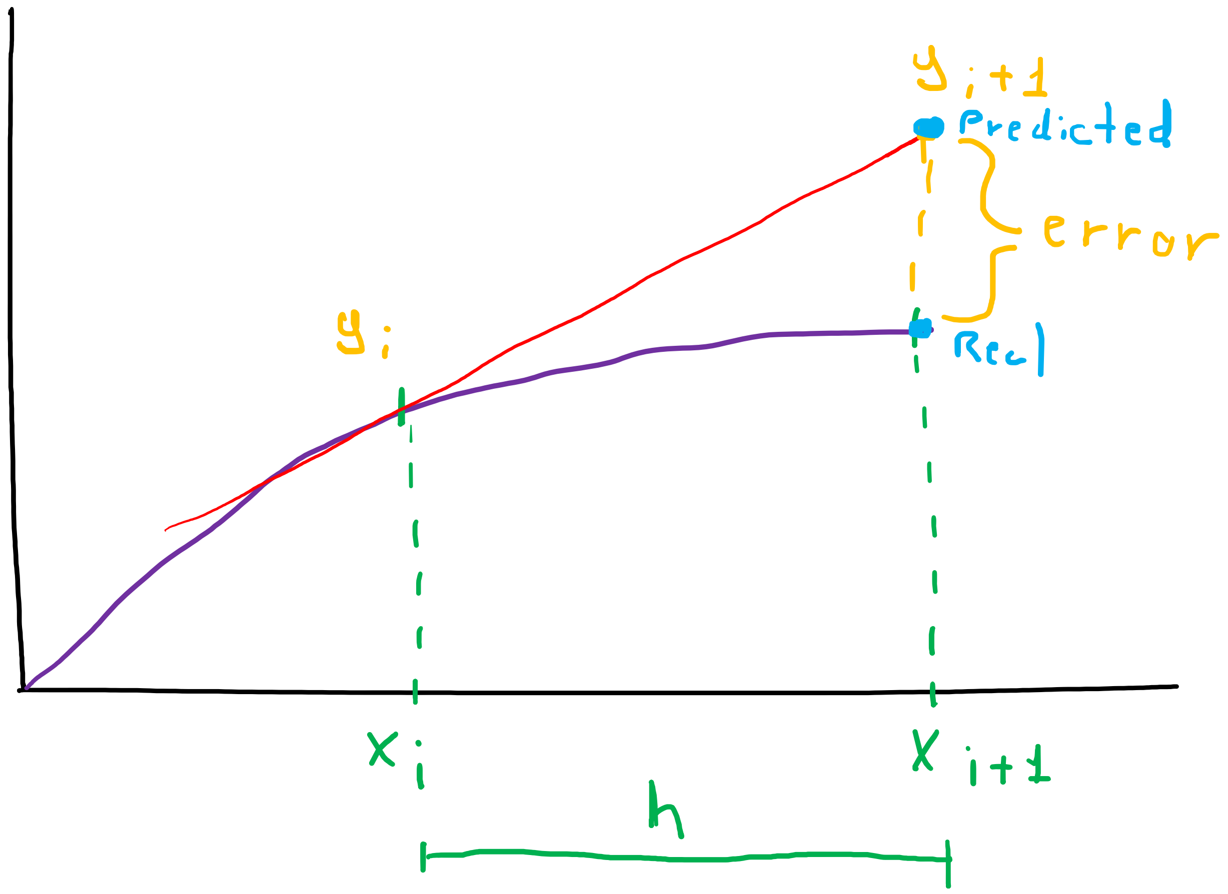
- Given the ODE:

$$\frac{dy}{dx} = f(x, y)$$

- We can find the solution as:

$$y_{i+1} = y_i + f(x, y) \cdot h$$

Euler Method



Euler Method - Example

- The ODE:

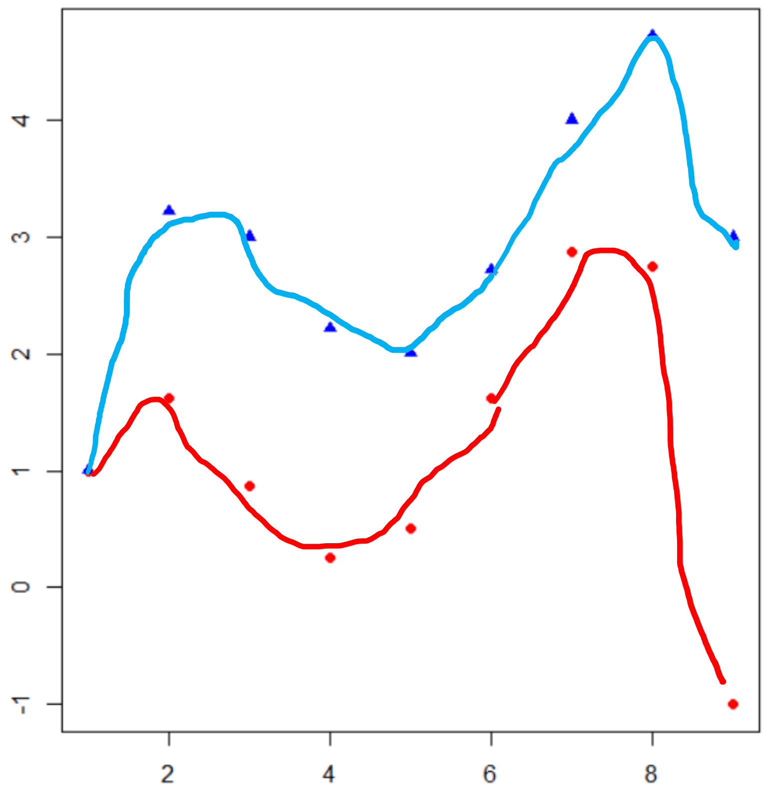
$$\frac{dy}{dx} = -2x^3 + 12x^2 - 20x + 8.5$$

- From 0 to 4 with steps of 0.5.
- Initial conditions: x = 0 and y = 1
- Compute the error knowing that the exact solution is:

$$y = -0.5x^4 + 4x^3 - 10x^2 + 8.5x + 1$$

Euler Method – Example – code

```
  x Approx    Real
0.0  1.000 1.00000
0.5  1.625 3.21875
1.0  0.875 3.00000
1.5  0.250 2.21875
2.0  0.500 2.00000
2.5  1.625 2.71875
3.0  2.875 4.00000
3.5  2.750 4.71875
4.0 -1.000 3.00000
```



Truncation error

- Truncation error is the error made by truncating an infinite sum and approximating it by a finite sum.

we take this

Residue

$$\underbrace{y_{i+1} = y_i + y_i' h}_{\text{we take this}} + \underbrace{\frac{y_i''}{2!} h^2 + \frac{y_i'''}{3!} h^3 + \cdots}_{\text{Residue}}$$

- Where $h = x_{i+1} - x_i$

Truncation error

- As:

$$y_i = f(x_i, y_i)$$

- Then:

$$y_{i+1} = y_i + f(x_i, y_i) \cdot h + \varepsilon$$

$$\varepsilon = \frac{f'(x_i, y_i) h^2}{2!} \rightarrow O(h^2)$$

Error Order

Heun method

- Given the ODE:

$$\frac{dy}{dx} = f(x_i, y_i)$$

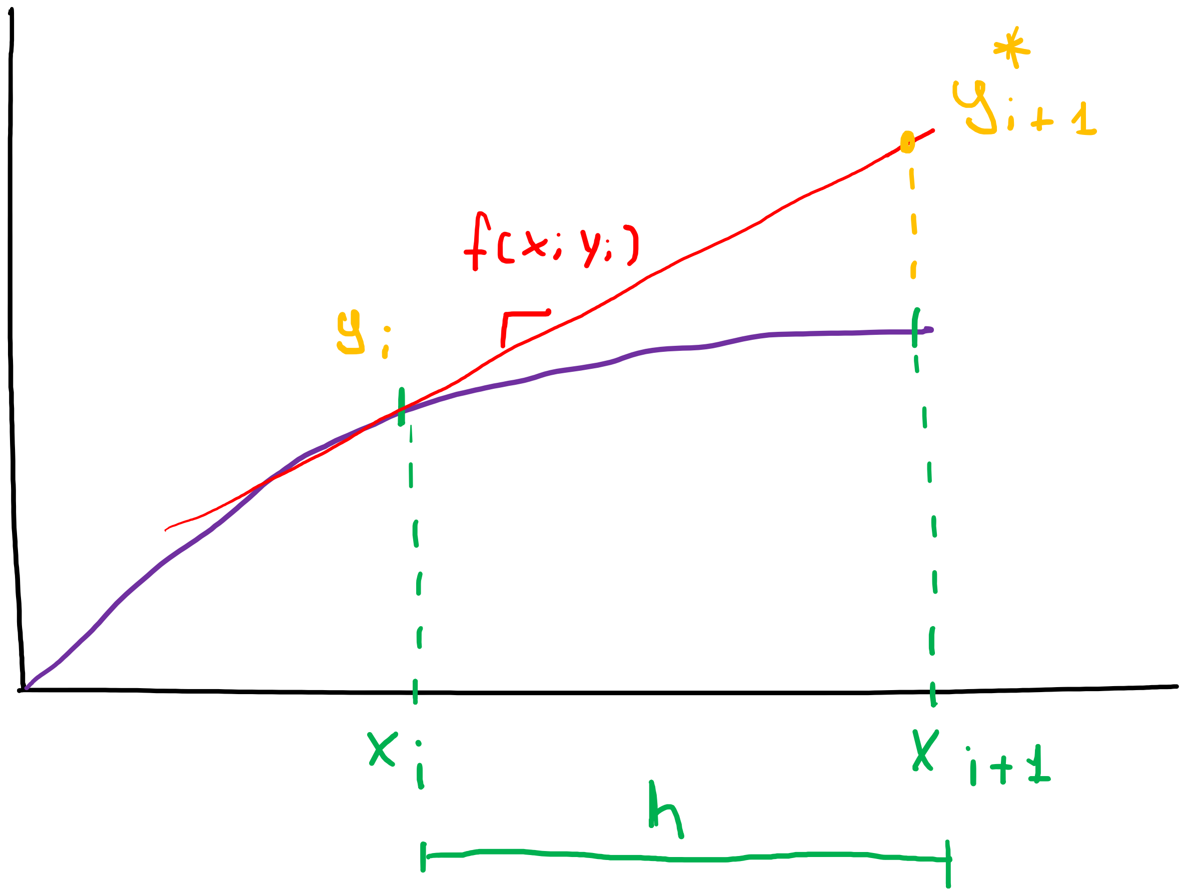
- We can find the solution as:

$$y^*_{i+1} = y_i + f(x_i, y_i)h \leftarrow \text{First step (Euler)}$$

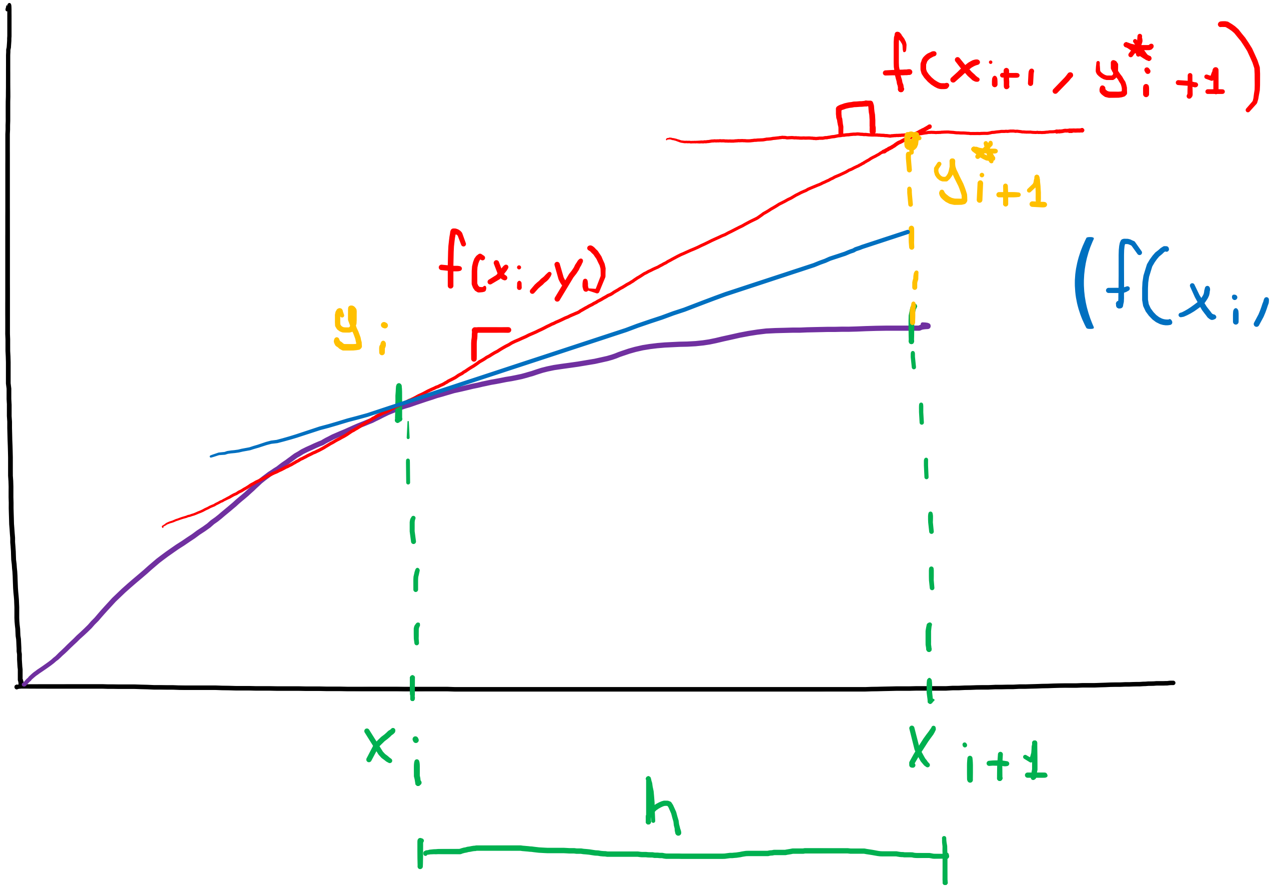
$$y_{i+1} = y_i + \left[f(x_i, y_i) + f(x_{i+1}, y^*_{i+1})\right] \cdot \frac{h}{2} \leftarrow 2^{nd} \text{ step}$$

Correction

Heun method – First step



Heun method – Slope correction



$$\left(f(x_i, y_i) + f(x_{i+1}, y^*_{i+1})\right)\frac{h}{2}$$

Heun method - Example

- The ODE:

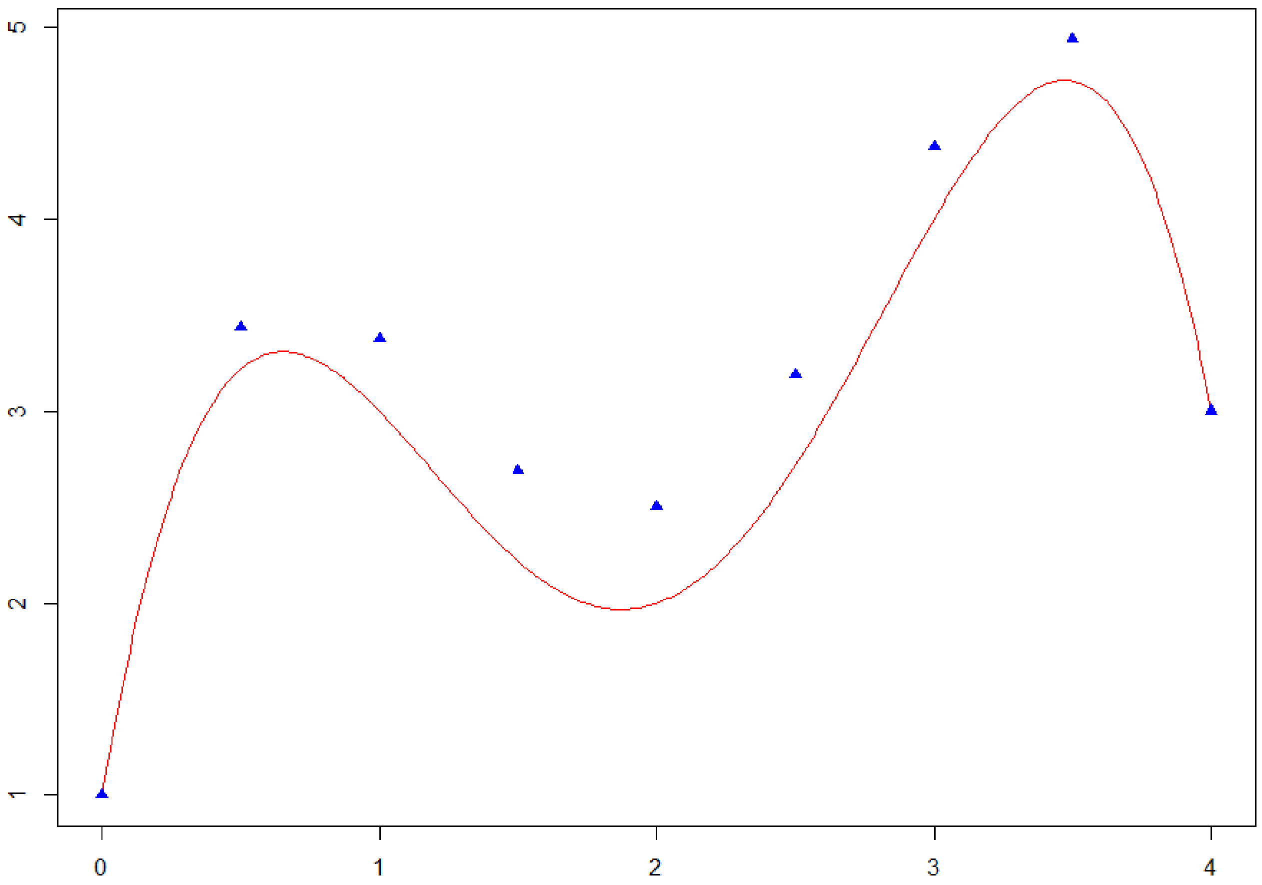
$$\frac{dy}{dx} = -2x^3 + 12x^2 - 20x + 8.5$$

- From 0 to 4 with steps of 0.5.
- Initial conditions: x = 0 and y = 1
- Compute the error knowing that the exact solution is:

$$y = -0.5x^4 + 4x^3 - 10x^2 + 8.5x + 1$$

Heun method – Example – code

x	Approx
0.0	1.0000
0.5	3.4375
1.0	3.3750
1.5	2.6875
2.0	2.5000
2.5	3.1875
3.0	4.3750
3.5	4.9375
4.0	3.0000



Example 2

$$\frac{dy}{dx} = f(x, y) = \ln(x)$$

$$x(0) = x_0 = 0.001$$

$$y(x_0) = 1$$

$$h = 0.25$$

Example – Euler method

$x_{i+1} = x_i + h$

$y_{i+1} = y_i + f(x,y) \cdot h$

First - step:

$y_1 = y_0 + f(x_0, y_0) \cdot h$

$= 1 + (-6.907) \cdot 0.25 \rightarrow$

$= -0.726$

$x_1 = x_0 + h$

$= 0.251$

Second - step

$Y_2 = -0.726 + (-1.382) \cdot 0.25$

$= -1.07$

$x_2 = 0.251 + 0.25$

$= 0.501$

Example – Heun method

$$y_{i+1}^* = y_i + f(x_i, y_i) \cdot h$$

$$y_{i+1} = y_i + \frac{h}{2}\left(f(x_i, y_i) + f(x_{i+1}, y_{i+1}^*)\right)$$

$$x_{i+1} = x_i + h$$

1st iter

$$y_1^* = 1 + (-6.907) \cdot 0.25$$

$$= -0.726$$

$$y_1 = 1 + \frac{0.25}{2}\left(-6.907 + (-1.382)\right)$$

$$= -0.0361$$

$$x_1 = 0.251$$

2nd iter

$$y_2^* = -0.0361 + (-1.382) \cdot 0.25$$

$$= -0.3816$$

$$y_2 = -0.0361 + \frac{0.25}{2}\left(-1.382 - 0.691\right)$$

$$= -0.295$$

$$x_2 = 0.501$$

Taylor method

- General case of Euler method
- It uses the Taylor series with a bigger order

$$\left.\begin{aligned} y' &= f(x,y) \\ y(x_0) &= y_0 \end{aligned}\right\}$$

unknown

$$y_{i+1} = y_i + y'(x_n)\cdot h + \frac{y''(x_n)\cdot h^2}{2!} + \frac{y'''(x_n)\cdot h^3}{3!} + \ldots$$

Taylor method – Pros and Cons

- Pros
  - Explicit
  - Can be higher order
- Cons
  - Needs the explicit form of derivatives of f(x,y)

Local Truncation Error

Euler method - LTE

$$LTE = \frac{y(x_i + h) - y(x_i)}{h} - f(x_i, y_i) \qquad \varepsilon = O(h)$$

$\therefore$

$$y_{i+1} = y_i + h \cdot f(x, y) + h \cdot LTE \rightarrow h \cdot O(h) \rightarrow O(h^2)$$

$$y_{i+1} = y_i + h \cdot f(x, y) + O(h^2)$$

Taylor method - LTE

$$y_{i+1} = y_i + h \cdot f(x,y) + \frac{h^2 f'(x,y)}{2!} + h \cdot LTE$$

$$F = f(x,y) + f'(x,y) \cdot \frac{h}{2}$$

$$LTE \sim o(h^2)$$

$$y_{i+1} = y_i + h \cdot F + h \cdot LTE \rightarrow h \cdot o(h^2) \rightarrow o(h^3)$$

$$y_{i+1} = y_i + h \cdot F + o(h^3)$$

Modified Euler Method – First step

- Improving the Euler method -> Midpoint

1st step → Euler method

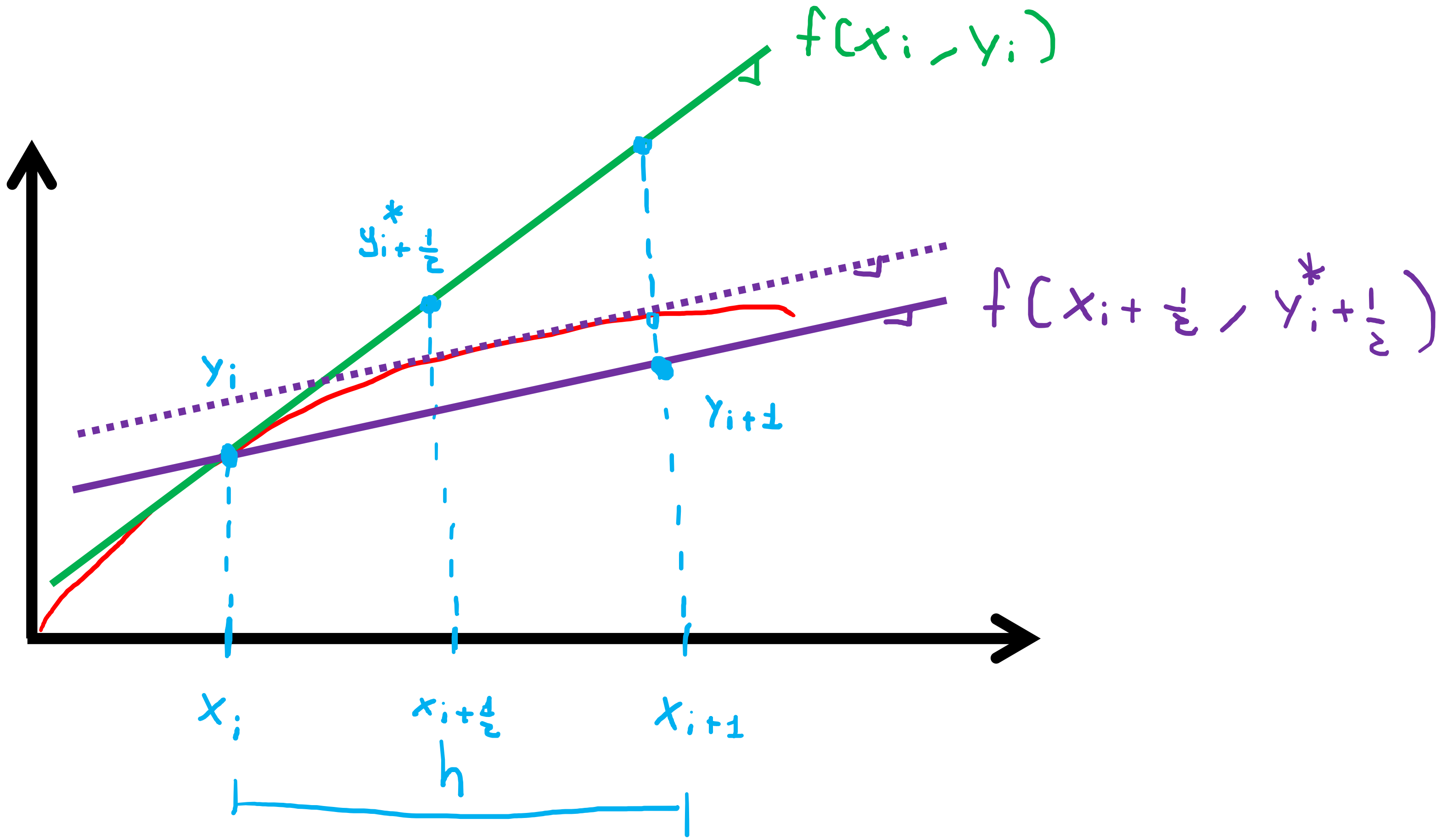
$$y_{i+1} = y_i + h \cdot f(x, y)$$

2nd step → Midpoint

$$y^*_{i+\frac{1}{2}} = y_i + \frac{h}{2} f(x, y)$$

$$y_{i+1} = y_i + h \cdot f\Big(\underbrace{x_{i+\frac{1}{2}}, \; y^*_{i+\frac{1}{2}}}_{\text{half-step}}\Big)$$

Modified Euler Method – First step



Modified Euler Method – Midpoint

$$y_{i+1} = y_i + h \cdot f\left(x_{i+\frac{1}{2}}, y^*_{i+\frac{1}{2}}\right) \qquad y^*_{i+1} = y_i + hf(x,y)$$

$$K_1 = hf(x,y)$$

$$K_2 = hf\left(x_{i+\frac{1}{2}}, y_{i+\frac{1}{2}}\right) \longrightarrow hf\left(x_{i+\frac{1}{2}}, y_i + \frac{K_1}{2}\right)$$

$$\therefore \quad \underline{y_{i+1} = y_i + hK_2}$$

Modified Euler Method – Second step

- Heun + Midpoint corrections

Mean Half-step

1^{st} step

$$y^*_{i+1} = y_i + hf(x_i, y_i) \qquad K_1 = hf(x_i, y_i)$$

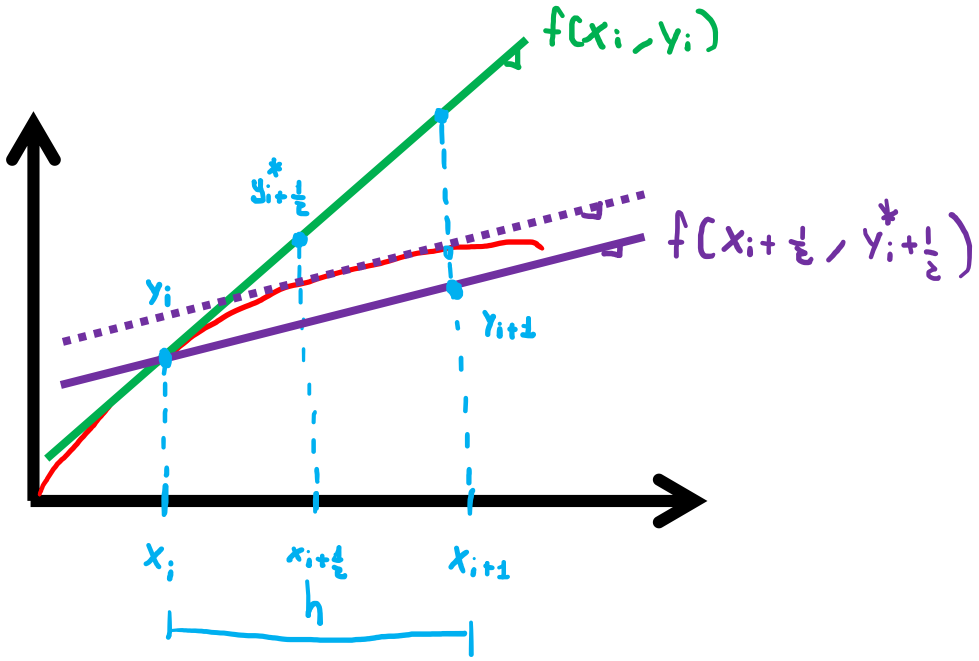
2^{nd} step → Midpoint

$$y^{**}_{i+1} = y_i + hf(x_{i+\frac{1}{2}}, y_{i+\frac{1}{2}}) \qquad K_2 = hf(x_{i+\frac{1}{2}}, y_i + \tfrac{1}{2}K_1)$$

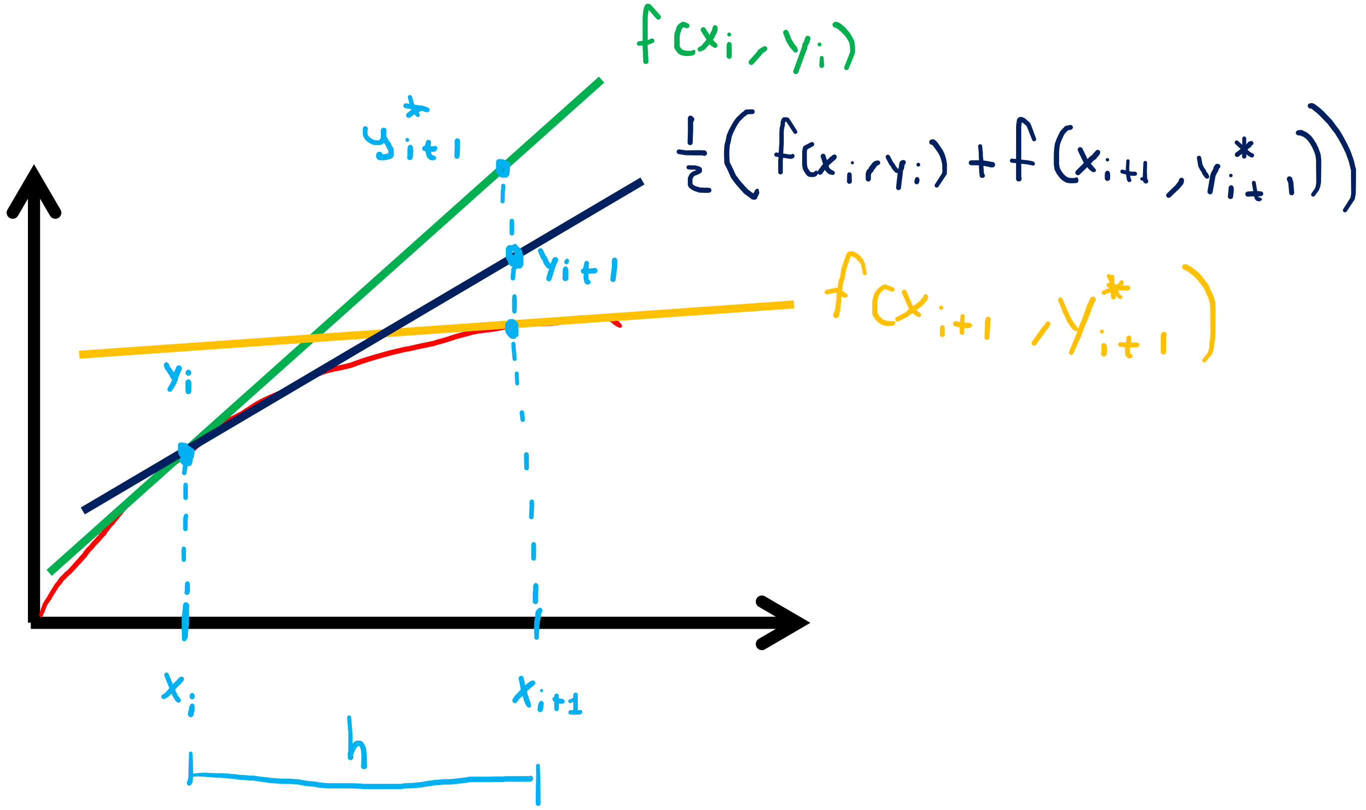
3^{rd} step → Heun

$$y_{i+1} = y_i + \frac{h}{2}\left(f(x_i, y_i) + f(x_{i+1}, y_{i+1})\right) = y_i + \frac{1}{2}(K_1 + K_2)$$

Modified Euler Method – Second step

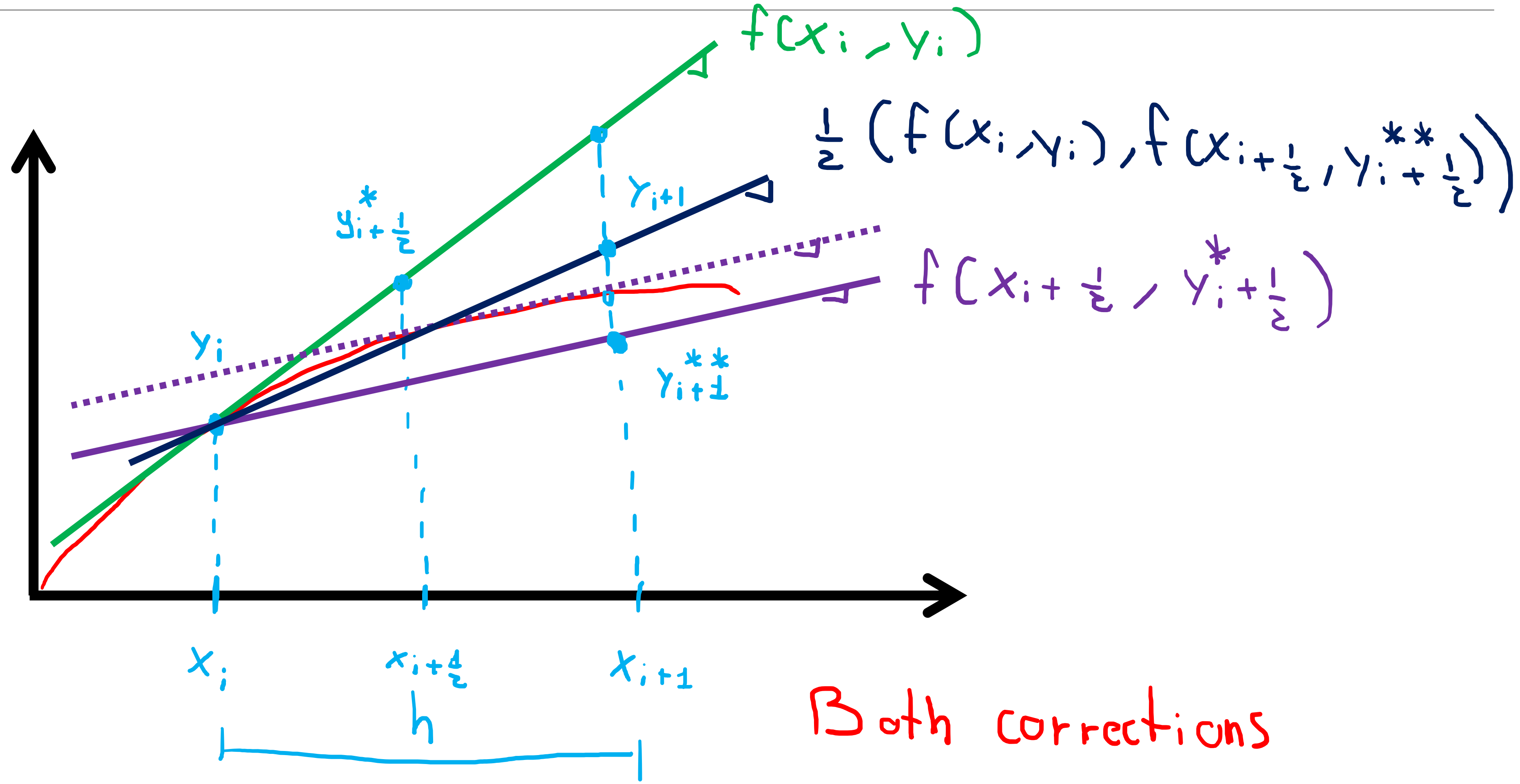


Midpoint correction



Heun correction

Modified Euler Method – Second step



Modified Euler Method - LTE

$$y_{i+1} = y_i + \frac{1}{2}(k_1 + k_2)$$

Taylor Series k_2

$$\frac{k_2}{h} = f(x_i, y_i) + h\frac{\partial}{\partial x} f(x_i, y_i) + k_1 \frac{\partial}{\partial y} f(x_i, y_i) + O(h^2, k_1^2)$$

Modified Euler Method - LTE

$K_2 = h \cdot o(h^2, k_1)$ $\quad k_1 \sim o(h^2)$

$\therefore h \cdot o(h^2) = \boxed{o(h^3)}$

Same error as Taylor Series to the 2nd derivative

LTE – Comparison

Euler $\sim O(h^2)$

Taylor 2nd derivative $\sim O(h^3)$

✓ Modified Euler $\sim O(h^3)$

Comparing methods

- Second-order Runge-Kutta method:

$$\begin{cases} K_1 = h(f(x_i, y_i)) \\ K_2 = h(f(x_i + \alpha h, y_i + \beta h)) \\ y_{i+1} = y_i + a_1 K_1 + a_2 K_2 \end{cases}$$

(1) $a_1 + a_2 = 1$

(2) $\alpha a_2 = \frac{1}{2}$

(3) $\beta a_2 = \frac{1}{2}$

$= \infty$ solutions

3 equations + 4-unknowns = under saturated

Comparing methods

- Modified Euler:

$a_1 = a_2 = \frac{1}{2}$ $\alpha = \beta = 1$

$k_2 = h(f(x_i + 1, y_i + k_1)$

$y_{i+1} = y_i + \frac{1}{2}k_1 + \frac{1}{2}k_2$

Comparing methods

- Midpoint method:

$a_1 = 0 \quad a_2 = 1 \qquad \alpha = \beta = \frac{1}{2}$

$k_2 = h f(x_i + \frac{1}{2}, y_i + \frac{1}{2} k_1)$

$y_{i+1} = y_i + k_2$

Comparing methods

- Heun-Ralston method

$$a_1 = \frac{1}{4} \quad a_2 = \frac{3}{4} \quad \alpha = \beta = \frac{2}{3}$$

$$k_2 = h f\left(x_{i+2/3}, y_i + \tfrac{2}{3} k_1\right)$$

$$y_{i+1} = y_i + \frac{1}{4}\left(k_1 + 3k_2\right)$$

ODE solution

- To solve the ODE in the interval [a,b] where b = a + Δx:
- Interval $[x_i, x_{i+1}]$ where $x_{i+1} = x_i + h$

$$\int_a^b \frac{dy}{dx}\,dx = \int_a^b f(x,y)\,dx$$

$$\therefore\ y_{i+1} = y_i + \int_{x_i}^{x_{i+1}} f(x,y)\,dx$$

Simpson´s rule

- Other numerical integration

$$\int_a^b f(x)\,dx \approx \int_a^b P(x)\,dx$$

Simpson´s rule

- Lagrange polynomial interpolation with 3 points
- Quadratic interpolation

$f(a),\ f(b),$

$f(m) \quad m = \frac{b-a}{2}$

$$P(x) = f(a)\frac{(x-m)(x-b)}{(a-m)(a-b)} +$$

$$f(m)\frac{(x-a)(x-b)}{(m-a)(m-b)} +$$

$$f(b)\frac{(x-a)(x-m)}{(b-a)(b-m)}$$

Simpson´s rule

- Integration equivalence

$$\int_a^b P(x)\,dx = \frac{\frac{b-a}{2}}{3}\left[f(a) + 4f(m) + f(b)\right]$$

$$\int_{x_i}^{x_{i+1}} P(x)\,dx = \frac{h}{6}\left[f(x_i) + 4f\left(\frac{x_{i+1}-x_i}{2}\right) + f(x_{i+1})\right]$$

Third order Range-Kutta method

- Mean -> Integral -> Simpson's rule

$$\mathbb{E}_x[f(x,y)] \equiv \int_a^b f(x,y)\,dx \equiv$$

$$\frac{h}{6}\left[f(x_i, y_i) + 4f\left(x_{i+\frac{1}{2}}, y_{i+\frac{1}{2}}\right) + f(x_{i+1}, y_{i+1})\right]$$

- Known: $x_i, y_i, x_{i+1/2}, x_{i+1}$
- Unknown: $y_{i+1/2}, y_{i+1}$

Third order Range-Kutta method

- First step: half-step

Euler $K1 = h f(x_i, y_i)$ $y^*_{i+\frac{1}{2}} = y_i + \frac{K1}{2}$

- Second step: Full-step

$$\begin{cases} \text{Midpoint } y^{**}_{i+1} = y_i + \overbrace{h f(x_{i+1/2}, y^*_{i+1/2})}^{K2} \\ \text{Euler } y^{**}_{i+1} = y_i + h \cdot K1 \end{cases}$$

3^{rd} Correction: [half-step] + [half-step] − [Full-step]

Third order Range-Kutta method

- RK 3th-order equations:

$$k1 = hf(x_i, y_i)$$

$$k2 = hf(x_{i+1/2}, y_i + 1/2\, k1)$$

$$k3 = hf(x_{i+1}, y_i + 2k2 - k1)$$

$$y_{i+1} = y_i + \frac{1}{6}(k1 + 4k2 + k3)$$

- LTE: $O(h^4)$

Fourth order Range-Kutta method

- Also based on Simpson's rule
- Equations:

$$K1 = h f(x_i, y_i)$$
$$K2 = h f(x_{i+\frac{1}{2}}, y_i + \frac{1}{2} K1)$$
$$K3 = h f(x_{i+\frac{1}{2}}, y_i + \frac{1}{2} K2)$$
$$K4 = h f(x_{i+1}, y_i + K3)$$

$$y_{i+1} = y_i + \frac{1}{6}(K1 + 2K2 + 2K3 + K4)$$

- LTE: $O(h^5)$

Range-Kutta method – Example

- Third order

$\frac{dy}{dx} = f(x, y) = \ln(x)$

$y_0 = 1$
$x_0 = 0.001$
$h = 0.25$

Solución analítica

$y(x) = 1 - x + x\ln(x)$

$y(x_0) = 1$

$y(0.251) = 0.40$

$y(0.501) = 0.15$

Solucion RK4

$\underline{1}$

0.3093

0.15

Range-Kutta method – Example

- Fourth order

$\frac{dy}{dx} = f(x, y) = \ln(x)$

$y_0 = 1$

$x_0 = 0.001$

$h = 0.25$

Solución analítica

$y(x) = 1 - x + x\ln(x)$

$y(x_0) = 1$

$y(0.251) = 0.40$

$y(0.501) = 0.15$

Solucion RK4

$\underline{1}$

0.3093

0.15

iff $\frac{dy}{dx} \perp y \rightarrow RK3 == RK4$

Systems numerical approximation

- Solve: Higher order, multivariate or coupled cases
- Two first order equations

$$\begin{cases} \dfrac{dy}{dt} = g(t, x, y) \\ \dfrac{dx}{dx} = f(t, x, y) \\ y(t_0) = y_0 \\ x(t_0) = x_0 \end{cases}$$

Higher order – Example

$$y'' - 0.05y' + 0.15y = 0 \qquad y'(t_0) = 0 \quad y(t_0) = 1$$

$$\begin{cases} y' = x \\ x' = 0.05x - 0.15y \end{cases} \qquad h = 0.5$$

$$\begin{cases} y(t_0) = 1 \\ x(t_0) = 0 \end{cases}$$

Higher order – Example

Euler method

$$y_{i+1} = y_i + h\,g(t_i, x_i, y_i)$$

$$y_1 = y_0 + 0.5 \cdot x$$

$$x_{i+1} = x_i + h \cdot f(t_i, x_i, y_i)$$

$$x_1 = x_0 + 0.5(0.05x - 0.15y)$$

Coupled system – Example

$$\begin{cases} \dfrac{dx}{dt} = -4x + 3y + 6 \\ \dfrac{dy}{dt} = -2x + y + 3 \\ x_0 = 0 \\ y_0 = 0 \end{cases}$$

Coupled system – Example

RK4

First: $K_{1x} = hf(t_i, x_i, y_i)$ $\quad K_{1y} = hg(t_i, x_i, y_i)$

$= h(-4x + 3y + 6)$ $\quad = h \cdot (-2x + y + 3)$

Second: $K_{2x} = hf(t_{i+\frac{1}{2}}, x^*_{i+\frac{1}{2}}, y^*_{i+1/2})$ $\quad K_{2y} = hg(t_{i+\frac{1}{2}}, x^*_{i+\frac{1}{2}}, y^*_{i+\frac{1}{2}})$

...

Stiffness

- Stability issues
- E.g. Exponential growth

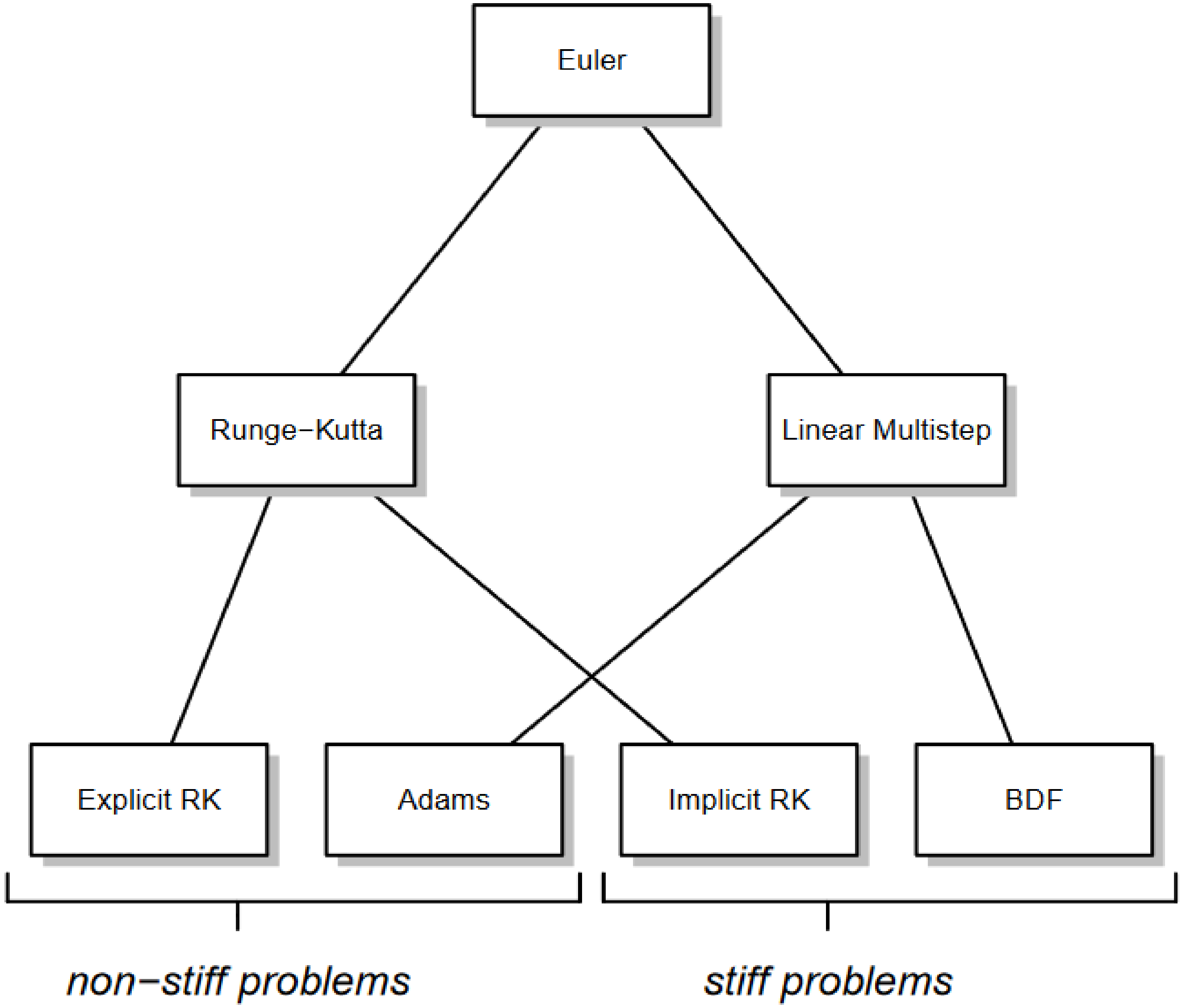
$$y'(x) = -\alpha y$$
$$y(0) = y_0$$
$$y(x) = y_0\, e^{\alpha x}$$

- If $\alpha x > 1$ unstable and the solution may oscillate

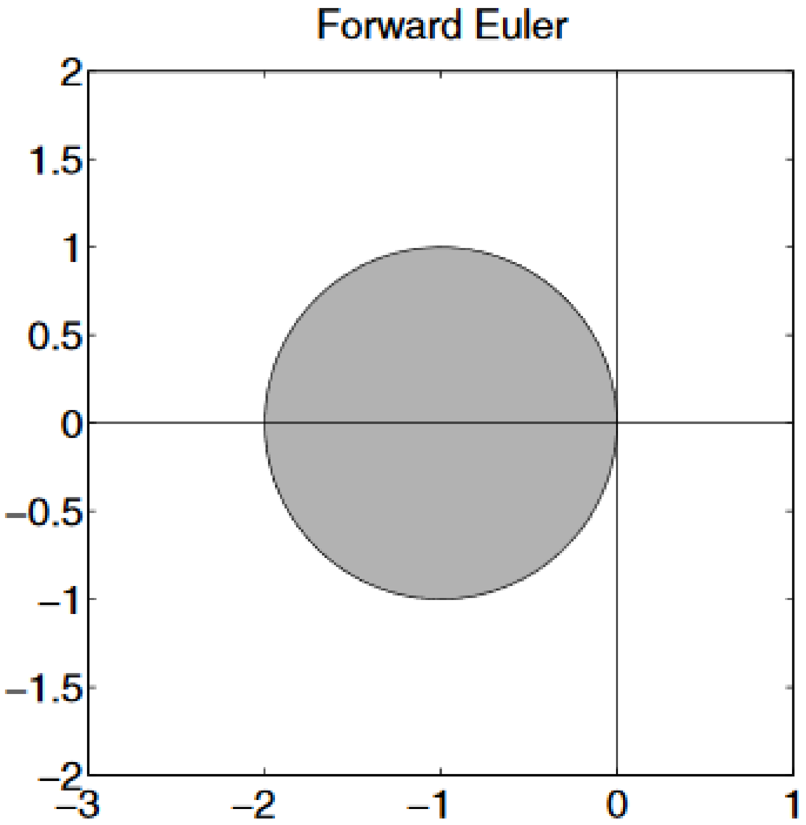
Stiffness

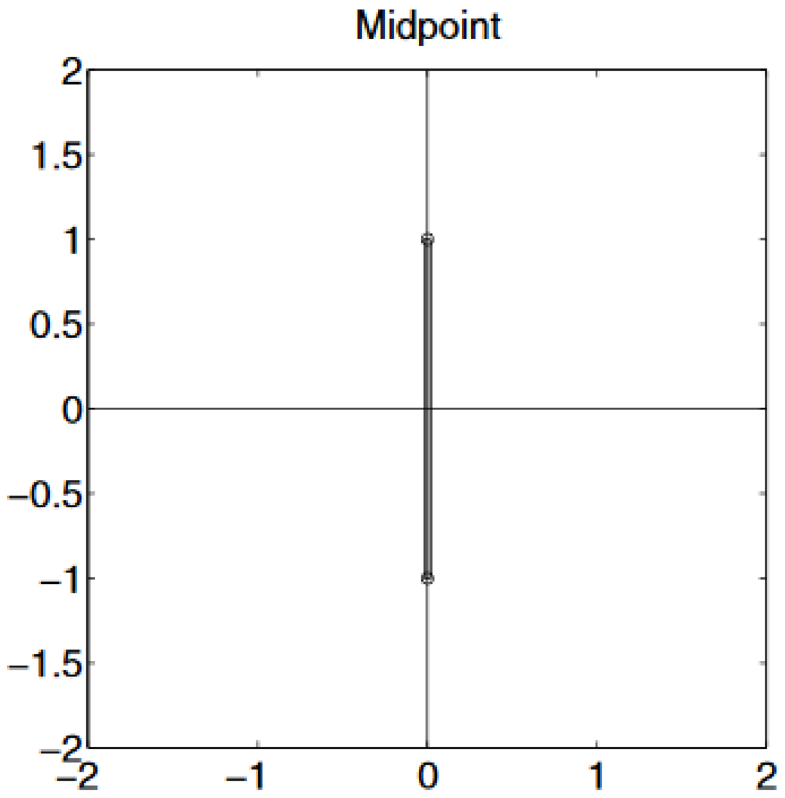
- A stiff equation is a differential equation for which certain numerical methods for solving the equation are numerically unstable, unless the step size is taken to be extremely small.

Stiffness

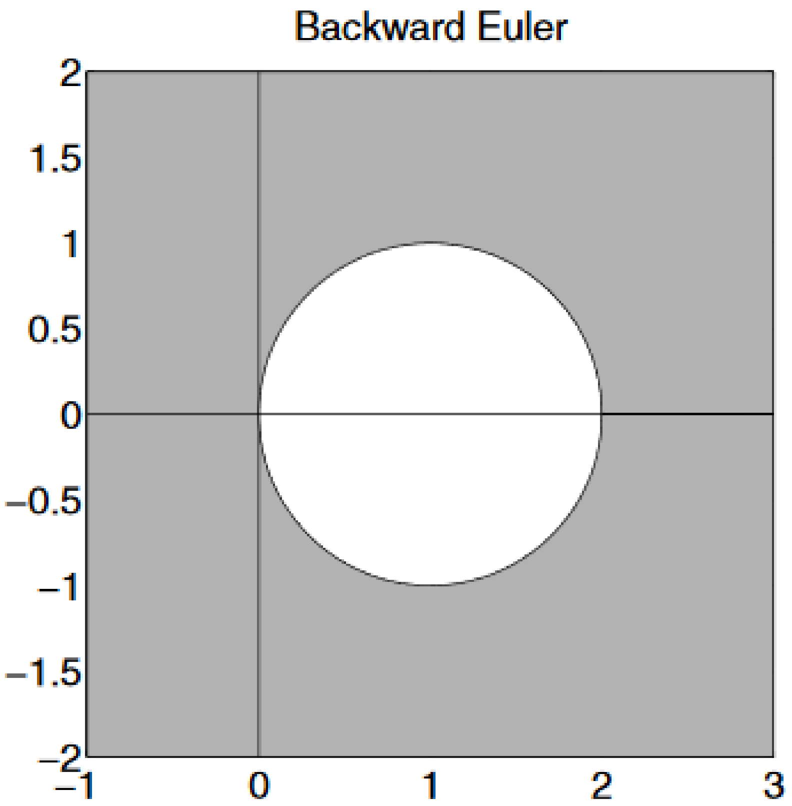

Euler
Runge–Kutta
Linear Multistep
Explicit RK
Adams
Implicit RK
BDF
non–stiff problems
stiff problems

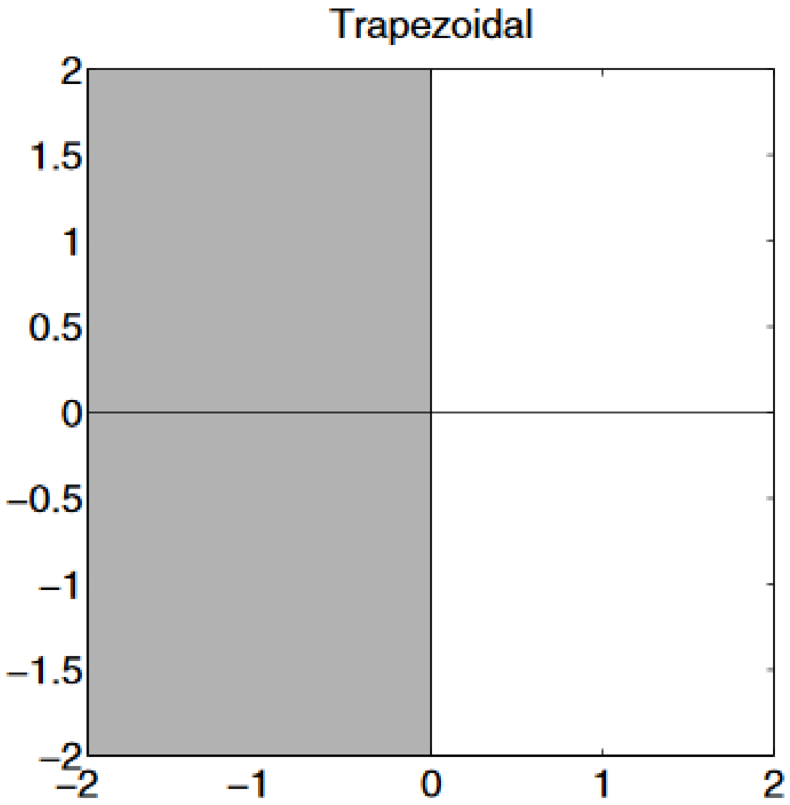
Absolute stability – Explicit





Absolute stability - Implicit





deSolve library

Function	Description
lsoda [9]	IVP ODEs, full or banded Jacobian, automatic choice for stiff or non-stiff method
lsodar [9]	same as lsoda; includes a root-solving procedure.
lsode [5], vode [2]	IVP ODEs, full or banded Jacobian, user specifies if stiff (bdf) or non-stiff (adams)
lsodes [5]	IVP ODEs; arbitrary sparse Jacobian, stiff
rk4, rk, euler	IVP ODEs; Runge-Kutta and Euler methods
radau [4]	IVP ODEs+DAEs; implicit Runge-Kutta method
daspk [1]	IVP ODEs+DAEs; bdf and adams method
zvode	IVP ODEs, like vode but for complex variables

adapted from [19].