9 Numerical Solutions of Ordinary Differential Equations

Solve $\mathbf{y}' = \mathbf{F}(x, \mathbf{y}), \mathbf{y}(a) = \boldsymbol{\alpha}$

Solve $y'' = f(x, y, y'), \quad y(a) = \alpha, \quad y(b) = \beta$

9.1 Classification of ODEs (Recall)

- An *ordinary differential equation* (ODE) is a differential equation where the dependent variable or variables depend on only one independent variable (usually *time* or *space*).

- Order of an ODE refers to the highest derivative or equivalently, to the number of simultaneous equations.

- ODEs can be classifed by the order of the equation as well as whether the system is *linear* or *nonlinear*.

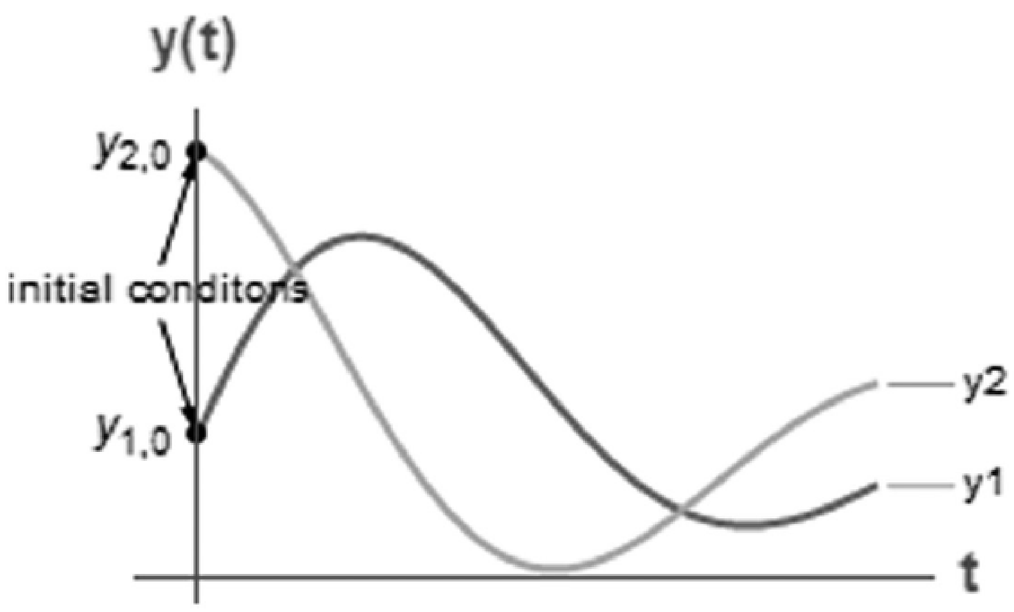
	Linear	**Nonlinear**
First-order ODE	$\frac{d\theta}{dt} = a(t)\theta + b(t)$	$\frac{d\theta}{dt} = f(t, \theta)$
Second-order ODE	$\frac{d^2\theta}{dt^2} + c_1\frac{d\theta}{dt} + c_2\theta = F(t)$ or $\frac{d\theta_1}{dt} = a_{11}\theta_1 + a_{12}\theta_2 + b_1$ $\frac{d\theta_2}{dt} = a_{21}\theta_1 + a_{22}\theta_2 + b_2$	$\frac{d^2\theta}{dt^2} = f\left(t, \theta, \frac{d\theta}{dt}\right)$ or $\frac{d\theta_1}{dt} = f_1(t, \theta_1, \theta_2)$ $\frac{d\theta_2}{dt} = f_2(t, \theta_1, \theta_2)$
n^{th}-order ODE	$\frac{d\theta_1}{dt} = a_{11}\theta_1 + a_{12}\theta_2 + \cdots + a_{1n}\theta_n + b_1$ $\frac{d\theta_2}{dt} = a_{21}\theta_1 + a_{22}\theta_2 + \cdots + a_{2n}\theta_n + b_2$ $\vdots$ $\frac{d\theta_n}{dt} = a_{n1}\theta_1 + a_{n2}\theta_2 + \cdots + a_{nn}\theta_n + b_n$	$\frac{d\theta_1}{dt} = f_1(t, \theta_1, \theta_2, \cdots, \theta_n)$ $\frac{d\theta_2}{dt} = f_2(t, \theta_1, \theta_2, \cdots, \theta_n)$ $\vdots$ $\frac{d\theta_n}{dt} = f_n(t, \theta_1, \theta_2, \cdots, \theta_n)$

- An equation of the form $\frac{d\theta}{dt}$=f(t, θ) is non-autonomous, while $\frac{d\theta}{dt}$=f(θ) is autonomous.

9.1 Initial Value and Boundary Value Problems (Recall)

Solve $\mathbf{y}' = \mathbf{F}(x, \mathbf{y}), \mathbf{y}(a) = \boldsymbol{\alpha}$

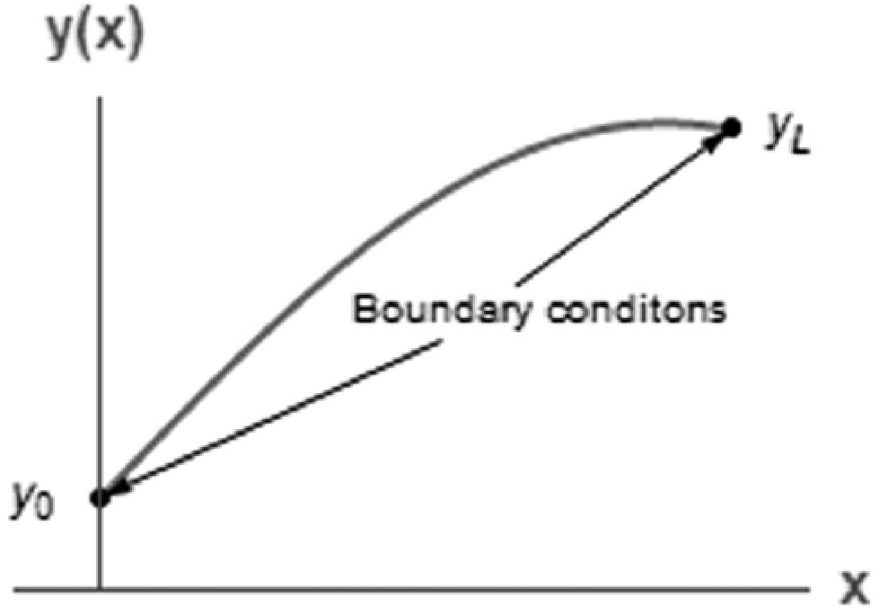
Initial Value Problem



If all the conditions are specifed at the same value of the independent variable, then we have an initial value problem.

Solve $y'' = f(x, y, y'), \quad y(a) = \alpha, \quad y(b) = \beta$

Boundary Value Problem



If conditions are known at different locations of the independent variable, then we have a boundary value problem

9.2 Taylor's Series Method

- Determine $y(0.2)$ with the fourth-order Taylor series method

$$y' + 4y = x^2 \qquad y(0) = 1$$

Also compute the estimated error and compare it with the actual error. The analytical solution of the differential equation is

$$y = \frac{31}{32}e^{-4x} + \frac{1}{4}x^2 - \frac{1}{8}x + \frac{1}{32}$$

```
%Exact Solution
syms y(x)
eqn = diff(y,x) == x^2-4*y;
cond = y(0) == 1;
S = dsolve(eqn,cond)
```

```
S =

(31*exp(-4*x))/32 - x/8 + x^2/4 + 1/32
```

9.2 Taylor's Series Method

$$y' + 4y = x^2 \qquad y(0) = 1$$

The Taylor series up to and including the term with h^4 is

$$y(h) = y(0) + y'(0)h + \frac{1}{2!}y''(0)h^2 + \frac{1}{3!}y'''(0)h^3 + \frac{1}{4!}y^{(4)}(0)h^4$$

$$\begin{aligned}
y' &= -4y + x^2 \\
y'' &= -4y' + 2x = 16y - 4x^2 + 2x \\
y''' &= 16y' - 8x + 2 = -64y + 16x^2 - 8x + 2 \\
y^{(4)} &= -64y' + 32x - 8 = 256y - 64x^2 + 32x - 8
\end{aligned}
\quad \longrightarrow \quad
\begin{aligned}
y'(0) &= -4(1) = -4 \\
y''(0) &= 16(1) = 16 \\
y'''(0) &= -64(1) + 2 = -62 \\
y^{(4)}(0) &= 256(1) - 8 = 248
\end{aligned}$$

$$\begin{aligned}
y(0.2) &= 1 + (-4)(0.2) + \frac{1}{2!}(16)(0.2)^2 + \frac{1}{3!}(-62)(0.2)^3 + \frac{1}{4!}(248)(0.2)^4 \\
&= 0.4539
\end{aligned}$$

9.2 Taylor's Series Method

$$y' + 4y = x^2 \qquad y(0) = 1$$

The analytical solution yields

$$y(0.2) = \frac{31}{32}e^{-4(0.2)} + \frac{1}{4}(0.2)^2 - \frac{1}{8}(0.2) + \frac{1}{32} = 0.4515$$

so that the actual error is $0.4515 - 0.4539 = -0.0024$.

```
% Exact Solution
syms y(x) a b
eqn = diff(y,x) == -4*y+x^2;
cond = [y(0)==a];
ySol(x) = dsolve(eqn,cond)
vpa(subs(ySol(x),[a,x],[1,0.2]),4)
```

9.3 Runge-Kutta Methods (First-Order: Euler's Method)

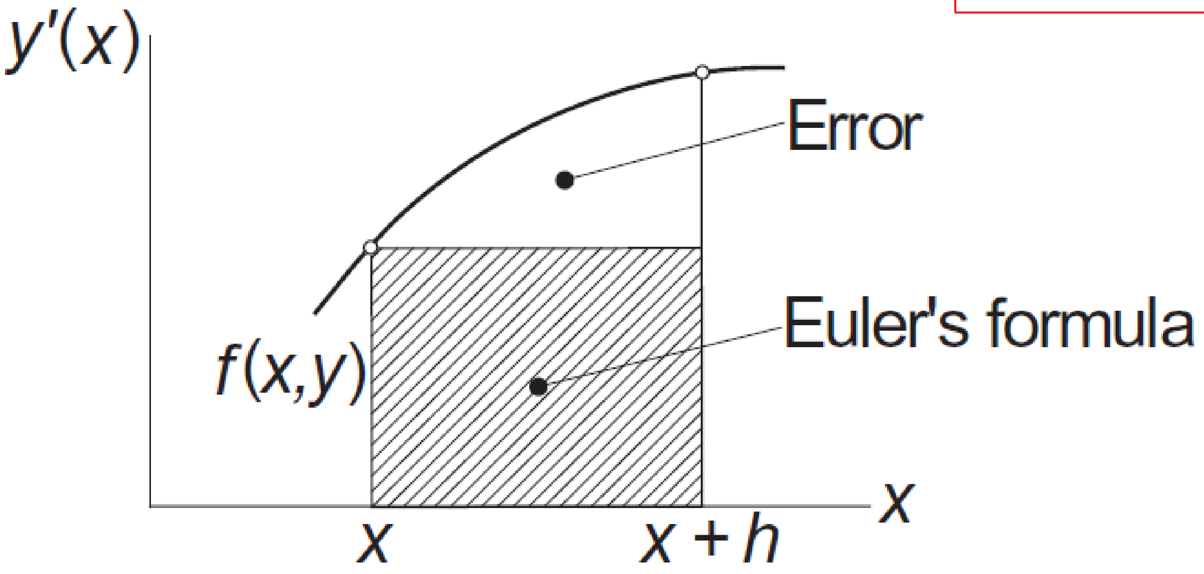
- The aim of Runge–Kutta methods is to eliminate the need for repeated differentiation of the differential equations.

- Since no such differentiation is involved in the first-order Taylor series integration formula

$$\mathbf{y}(x+h) = \mathbf{y}(x) + \mathbf{y}'(x)h = \mathbf{y}(x) + \mathbf{F}(x, \mathbf{y})h$$

it can be considered as the first-order Runge–Kutta method; it is also called *Euler's method*.

$$\boxed{\begin{aligned} k_1 &= f\left(t_i, y_i\right) \\ y_{i+1} &= y_i + h \cdot k_1 \end{aligned}}$$



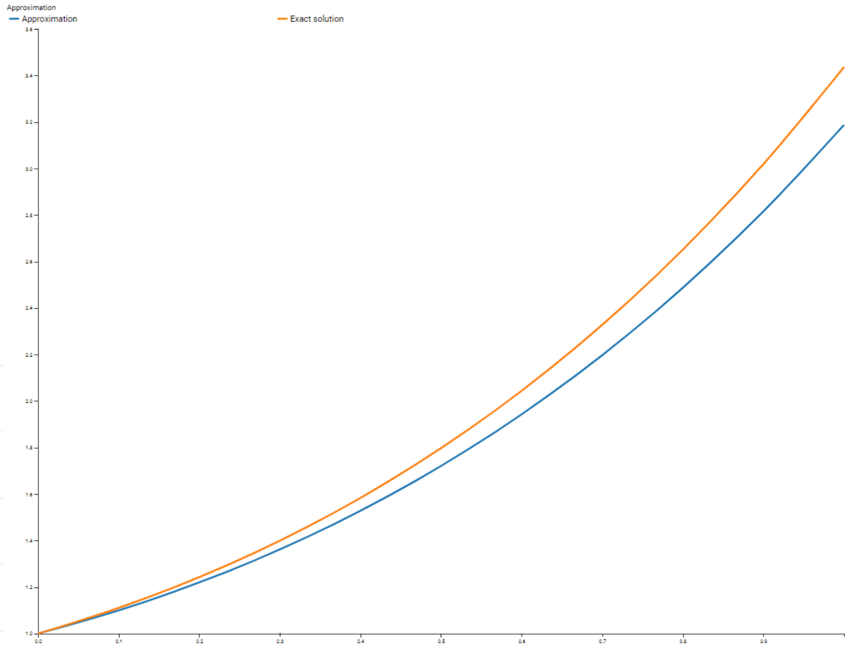
$$y(x+h) - y(h) = \int_x^{x+h} y'\, dx = \int_x^{x+h} f(x, y) dx$$

9.3 Runge-Kutta Methods (First-Order: Euler's Method)

$$\frac{dy}{dx} = x + y$$

If y(0)=1, Find y(1)

https://planetcalc.com/8389/?dydx=x%2By&x0=0&y0=1&h=0.1&x=1&yexact=2*e%5Ex-x-1



n	x(n)	Approximation	f(x,y)	dy	y(n+1)	Exact solution	Absolute error
0	0	1	1	0.1	1.1	1	0
1	0.1	1.1	1.20	0.12	1.22	1.11	0.0103
2	0.2	1.22	1.42	0.14	1.36	1.24	0.0228
3	0.30	1.36	1.66	0.17	1.53	1.40	0.0377
4	0.4	1.53	1.93	0.19	1.72	1.58	0.0554
5	0.5	1.72	2.22	0.22	1.94	1.80	0.0764
6	0.6	1.94	2.54	0.25	2.20	2.04	0.1011
7	0.7	2.20	2.90	0.29	2.49	2.33	0.1301
8	0.80	2.49	3.29	0.33	2.82	2.65	0.1639
9	0.90	2.82	3.72	0.37	3.19	3.02	0.2033
10	1.00	3.19				3.44	0.2491

9.3 Runge-Kutta Methods (First-Order: Euler's Method)

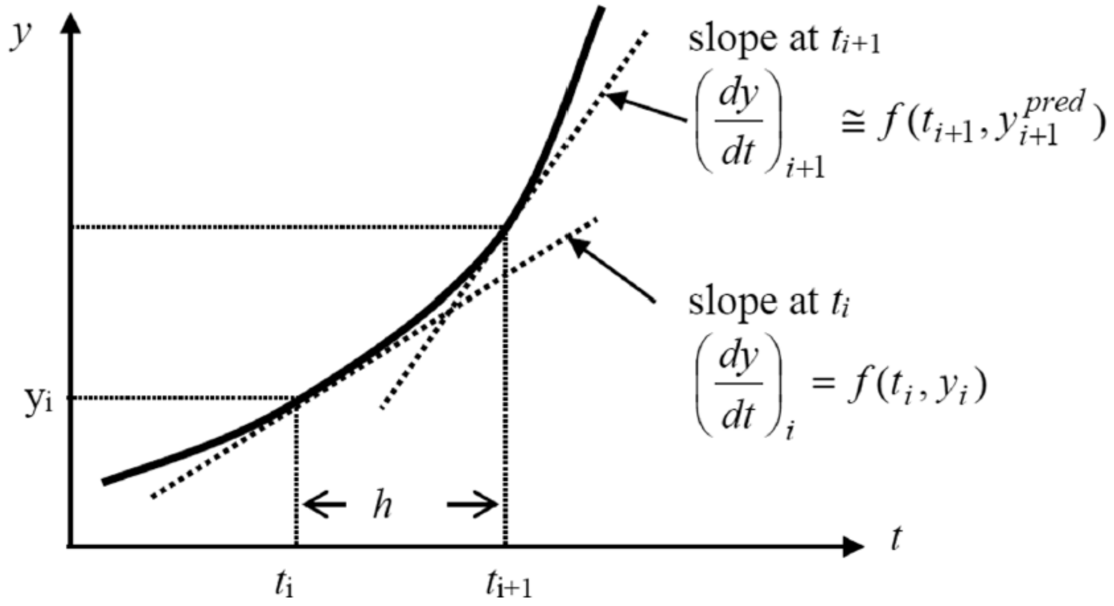
$$\frac{dy}{dx} = x + y$$

If y(0)=1, Find y(1)

```
%Housekeeping
clear; clc; close all;

%Exact Solution
syms y(x)
eqn = diff(y,x) == x+y;
cond = y(0) == 1;
S = dsolve(eqn,cond)

%Eulers Method (First Order)
a=0;b=1;ya=1;N=10;
euler_ode(a,b,ya,N)
```

```
% Define Function
function dydx=func(x,y)
dydx=x+y;
end

function E=euler_ode(a,b,ya,N)
h=(b-a)/N;
x(1)=a;
y(1)=ya;
for i=1:N
x(i+1)=x(i)+h;
y(i+1)=y(i)+h*func(x(i),y(i));
end
E=y(i+1);
end
```

9.3 Runge-Kutta Methods (Second-Order): Heun's Predictor-Corrector Method



$$k_1 = f\left(t_i, y_i\right)$$

$$k_2 = f\left(t_i + h, y_i + hk_1\right)$$

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

9.3 Runge-Kutta Methods (Second-Order): Heun's Predictor-Corrector Method

$$\frac{dy}{dx} = x + y$$

If y(0)=1, Find y(1)

```
%Housekeeping
clear; clc; close all;

%Exact Solution
syms y(x)
eqn = diff(y,x) == x+y;
cond = y(0) == 1;
S = dsolve(eqn,cond)

%Heun's Method (Second Order: Predictor Corrector)
a=0;b=1;ya=1;N=10;
heun_ode(a,b,ya,N)
```

```
% Define Function
function dydx=func(x,y)
dydx=x+y;
end

function E=heun_ode(a,b,ya,N)
h=(b-a)/N;
x(1)=a;
y(1)=ya;
for i=1:N
k1=func(x(i),y(i));
k2=func(x(i)+h,y(i)+h*k1);
x(i+1)=x(i)+h;
y(i+1)=y(i)+0.5*h*(k1+k2);
end
E=y(i+1);
end
```

9.3 Runge-Kutta Methods (2nd Order: Midpoint method)

Method	***Discretizion of*** $\boldsymbol{\frac{dy}{dt} = f(t, y)}$	$\boldsymbol{k_i}$**'s**
2nd Order: Midpoint Method	$y_{i+1} = y_i + h \cdot k_2$	$k_1 = f(t_i, y_i)$ $k_2 = f(t_i + \frac{1}{2}h, y_i + \frac{1}{2}h \cdot k_1)$

$$\frac{dy}{dx} = x + y$$

If y(0)=1, Find y(1)

9.3 Runge-Kutta Methods (2^{nd} Order: Ralston's method)

Method	Discretizion of $\frac{dy}{dt} = f(t,y)$	k_i's
2^{nd} Order: Ralston's Method	$y_{i+1} = y_i + h\left(\frac{1}{3}k_1 + \frac{2}{3}k_2\right)$	$k_1 = f(t_i, y_i)$ $k_2 = f(t_i + \frac{1}{2}h, y_i + \frac{1}{2}h \cdot k_1)$

$$\frac{dy}{dx} = x + y$$

If y(0)=1, Find y(1)

9.3 Runge-Kutta Methods (3rd Order)

Method	$\boldsymbol{Discretizion\ of\ \frac{dy}{dt} = f(t, y)}$	$\boldsymbol{k_i}$'s
3rd Order	$y_{i+1} = y_i + \frac{h}{6}(k_1 + 4k_2 + k_3)$	$k_1 = f(t_i, y_i)$ $k_2 = f(t_i + \frac{1}{2}h, y_i + \frac{1}{2}h \cdot k_1)$ $k_3 = f(t_i + h, y_i - h \cdot k_1 + 2h \cdot k_2)$

$$\frac{dy}{dx} = x + y$$

If y(0)=1, Find y(1)

9.3 Runge-Kutta Methods (4^{th} Order)

Method	$Discretizion\ of\ \frac{dy}{dt} = f(t, y)$	k_i's
4^{th} Order	$y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$	$k_1 = f(t_i, y_i)$ $k_2 = f(t_i + \frac{1}{2}h, y_i + \frac{1}{2}h \cdot k_1)$ $k_3 = f(t_i + \frac{1}{2}h, y_i + \frac{1}{2}h \cdot k_2)$ $k_4 = f(t_i + h, y_i + h \cdot k_3)$

$$\frac{dy}{dx} = x + y$$

If y(0)=1, Find y(1)

9.3 Runge-Kutta Methods (4^{th} Order)

```
function [xSol,ySol] = runKut4(dEqs,x,y,xStop,h)
% 4th-order Runge--Kutta integration.
% USAGE: [xSol,ySol] = runKut4(dEqs,x,y,xStop,h)
% INPUT:
% dEqs = handle of function that specifies the
% 1st-order differential equations
% F(x,y) = [dy1/dx dy2/dx dy2/dx ...].
% x,y = initial values; y must be row vector.
% xStop = terminal value of x.
% h = increment of x used in integration.
% OUTPUT:
% xSol = x-values at which solution is computed.
% ySol = values of y corresponding to the x-values.
```

```
% House Keeping
clear; clc; close all;
[x,y] = runKut4(@fn,0,1,1,0.05);
printSol(x,y,0)

function F = fn(x,y)
F=x+y;
end
```

```
if size(y,1) > 1 ; y = y'; end % y must be row vector
xSol = zeros(2,1); ySol = zeros(2,length(y));
xSol(1) = x; ySol(1,:) = y;
i = 1;
while x < xStop
i = i + 1;
h = min(h,xStop - x);
K1 = h*feval(dEqs,x,y);
K2 = h*feval(dEqs,x + h/2,y + K1/2);
K3 = h*feval(dEqs,x + h/2,y + K2/2);
K4 = h*feval(dEqs,x+h,y + K3);
y = y + (K1 + 2*K2 + 2*K3 + K4)/6;
x = x + h;
xSol(i) = x; ySol(i,:) = y; % Store current soln.
end
```

x	y1
0.0000	1.0000
1.0000	3.4366

9.3 Runge-Kutta Methods (MATLAB built-in)

$\frac{dy}{dx} = x + y$, *y(0)=1, y(1)=3.4366 (exact)*

```
% House Keeping
clear; clc; close all;
[x, y] = ode45(@fn,[0:0.2:1],1)

function F = fn(x,y)
F=x+y;
end
```

```
x =

        0
   0.2000
   0.4000
   0.6000
   0.8000
   1.0000


y =

   1.0000
   1.2428
   1.5836
   2.0442
   2.6511
   3.4366
```

9.3 Runge-Kutta Methods (Higher-Order)

$\frac{dy}{dx} = x + y$, *y(0)=1, y(1)=3.4366 (exact)*

Method	Discretizion of $\frac{dy}{dt} = f(t, y)$	k_i's
1^{st} Order: Euler Method	$y_{i+1} = y_i + h \cdot k_1$	$k_1 = f(t_i, y_i)$
2^{nd} Order: Heun Method	$y_{i+1} = y_i + h\frac{k_1 + k_2}{2}$	$k_1 = f(t_i, y_i)$ $k_2 = f(t_i + h, y_i + h \cdot k_1)$
2^{nd} Order: Midpoint Method	$y_{i+1} = y_i + h \cdot k_2$	$k_1 = f(t_i, y_i)$ $k_2 = f(t_i + \frac{1}{2}h, y_i + \frac{1}{2}h \cdot k_1)$
2^{nd} Order: Ralston's Method	$y_{i+1} = y_i + h\left(\frac{1}{3}k_1 + \frac{2}{3}k_2\right)$	$k_1 = f(t_i, y_i)$ $k_2 = f(t_i + \frac{1}{2}h, y_i + \frac{1}{2}h \cdot k_1)$
3^{rd} Order	$y_{i+1} = y_i + \frac{h}{6}(k_1 + 4k_2 + k_3)$	$k_1 = f(t_i, y_i)$ $k_2 = f(t_i + \frac{1}{2}h, y_i + \frac{1}{2}h \cdot k_1)$ $k_3 = f(t_i + h, y_i - h \cdot k_1 + 2h \cdot k_2)$
4^{th} Order	$y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$	$k_1 = f(t_i, y_i)$ $k_2 = f(t_i + \frac{1}{2}h, y_i + \frac{1}{2}h \cdot k_1)$ $k_3 = f(t_i + \frac{1}{2}h, y_i + \frac{1}{2}h \cdot k_2)$ $k_4 = f(t_i + h, y_i + h \cdot k_3)$

Method	y(1)
Euler	3.1875
Heun	3.4282
Midpoint	3.4282
Ralson	3.3468
3^{rd} Order	3.4364
4^{th} Order	3.4366
ode45	3.4366

9.4 State Space Form (Higher Order DEs)

$$\frac{d^n y(t)}{dt^n} + a_{n-1}\frac{d^{n-1}y(t)}{dt^{n-1}} + \cdots + a_1\frac{dy(t)}{dt} + a_0 y(t) = u(t)$$

$$x_1(t) = y(t)$$

$$x_2(t) = \frac{dy(t)}{dt}$$

$$x_3(t) = \frac{d^2y(t)}{dt^2}$$

$$\vdots$$

$$x_n(t) = \frac{d^{n-1}y(t)}{dt^{n-1}}$$

$$\frac{dx_1(t)}{dt} = \dot{x}_1 = \frac{dy(t)}{dt} = x_2(t)$$

$$\frac{dx_2(t)}{dt} = \dot{x}_2 = \frac{d^2y(t)}{dt^2} = x_3(t)$$

$$\frac{dx_3(t)}{dt} = \dot{x}_3 = \frac{d^3y(t)}{dt^3} = x_4(t)$$

$$\vdots$$

$$\frac{dx_n(t)}{dt} = \dot{x}_n = \frac{d^n y(t)}{dt^n}$$

$$= -a_0 y(t) - a_1\frac{dy(t)}{dt} - a_2\frac{d^2y(t)}{dt^2} - \cdots - a_{n-1}\frac{d^{n-1}y(t)}{dt^{n-1}} + u(t)$$

$$= -a_0 x_1(t) - a_1 x_2(t) - \cdots - a_2 x_3(t) - \cdots - a_{n-1}x_n(t) + u(t)$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \cdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & \cdots & 0 & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & \cdots & -a_{n-1} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ \vdots \\ x_{n-1}(t) \\ x_n(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ 1 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_{n-1}(t) \\ x_n(t) \end{bmatrix}$$

Matrix

Vector　　Vector　　Vector or Matrix

$$\mathbf{x}'(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$$

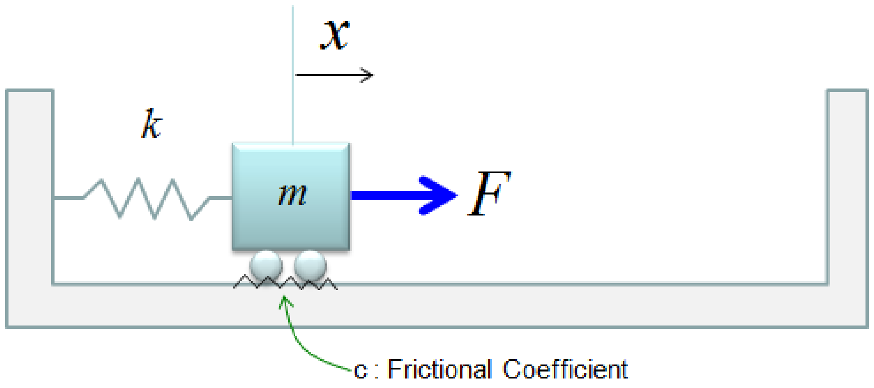
Vector

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t)$$

Vector

Vector or Matrix　　Vector

9.4 State Space Form (Higher Order DEs)



$$m.\ddot{y} + c.\dot{y} + k.y = F$$

$$x_1 = y$$
$$x_2 = \dot{y}$$

$$\dot{x_1} = x_2$$
$$\dot{x_2} = \frac{1}{m}(F - c.x_2 - k.x_1)$$

$$\begin{Bmatrix} \dot{x_1} \\ \dot{x_2} \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix} \cdot \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} + \begin{Bmatrix} 0 \\ \frac{F}{m} \end{Bmatrix}$$

$$\{x_1\} = [1 \quad 0] \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} + (0)F$$

Matrix

Vector

Vector

Vector or Matrix

$$\mathbf{x}'(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$$

Vector

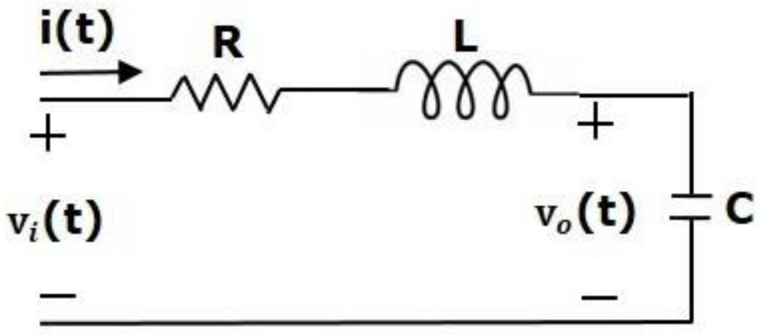
$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t)$$

Vector

Vector or Matrix

Vector

9.4 State Space Form (Higher Order DEs)



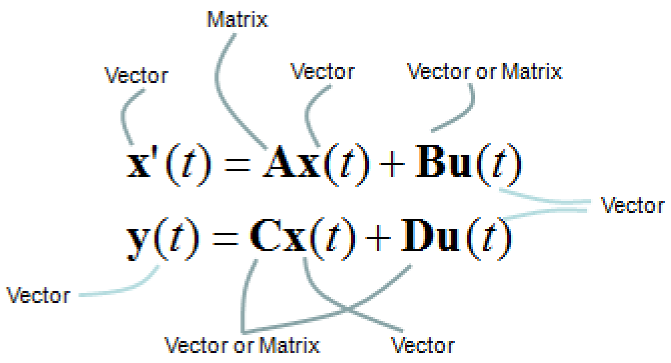
$$\ddot{i} + \frac{R}{L}.\dot{i} + \frac{1}{L.C}.i = \frac{1}{L.C}.u$$

$$x_1 = i$$
$$x_2 = \dot{i}$$

$$\dot{x_1} = x_2$$
$$\dot{x_2} = \left(-\frac{R}{L}.x_2 - \frac{1}{L.C}.x_1 + \frac{1}{L.C}.u\right)$$

$$\begin{Bmatrix} \dot{x_1} \\ \dot{x_2} \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{1}{L.C} & -\frac{R}{L} \end{bmatrix} . \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} + \begin{Bmatrix} 0 \\ \frac{1}{L.C} \end{Bmatrix} .u$$

$$\{x_1\} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} + (0).u$$



9.4 Higher order ODES: Runge-Kutta Methods (4th Order)

Method	Discretizion of $\frac{dy}{dt} = f(t, y)$	k_i's
4th Order	$y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$	$k_1 = f(t_i, y_i)$ $k_2 = f(t_i + \frac{1}{2}h, y_i + \frac{1}{2}h \cdot k_1)$ $k_3 = f(t_i + \frac{1}{2}h, y_i + \frac{1}{2}h \cdot k_2)$ $k_4 = f(t_i + h, y_i + h \cdot k_3)$

Solve:

$y'' = -0.1y' - x$

y(0)=1 & y'(0)=1

from $x = 0$ to 2 in increments of $h = 0.25$ with the fourth-order Runge–Kutta method

```
% House Keeping
clear; clc; close all;
[x,y]=runKut4(@fn,0,[1 1],2,0.25);
printSol(x,y,1)

function F = fn(x,y)
F = zeros(1,2);
F(1) = y(2); F(2) = -0.1*y(2) - x;
end
```

x	y1	y2
0.0000	1.0000	1.0000
0.2500	1.2443	0.9443
0.5000	1.4671	0.8283
0.7500	1.6536	0.6534
1.0000	1.7890	0.4211
1.2500	1.8594	0.1328
1.5000	1.8509	-0.2101
1.7500	1.7500	-0.6062
2.0000	1.5434	-1.0543

9.4 Higher order ODES: MATLAB built-in

Method	Discretizion of $\frac{dy}{dt} = f(t,y)$	k_i's
4^{th} Order	$y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$	$k_1 = f(t_i, y_i)$ $k_2 = f(t_i + \frac{1}{2}h, y_i + \frac{1}{2}h \cdot k_1)$ $k_3 = f(t_i + \frac{1}{2}h, y_i + \frac{1}{2}h \cdot k_2)$ $k_4 = f(t_i + h, y_i + h \cdot k_3)$

Solve:

$y'' = -0.1y' - x$

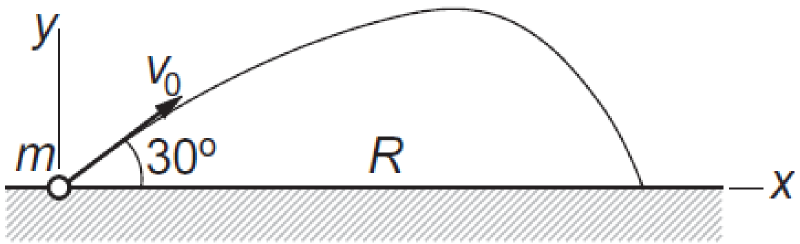
y(0)=1 & y'(0)=1

from $x = 0$ to 2 using ode45 built-in MATLAB function

```
% House Keeping
clear; clc; close all;
[x, y] = ode45(@fn,[0:0.25:2],[1; 1])

function F = fn(x,y)
F = zeros(2,1);
F(1) = y(2); F(2) = -0.1*y(2) - x;
end
```

```
y =

    1.0000    1.0000
    1.2443    0.9443
    1.4671    0.8283
    1.6536    0.6534
    1.7890    0.4211
    1.8594    0.1328
    1.8509   -0.2101
    1.7500   -0.6062
    1.5434   -1.0543
```

9.4 Example



A ball of mass $m = 0.25$ kg is launched with the velocity $v_0 = 50$ m/s in the direction shown. Assuming that the aerodynamic drag force acting on the ball is $F_D = C_D v^{3/2}$, the differential equations describing the motion are

$$\ddot{x} = -\frac{C_D}{m}\dot{x}v^{1/2} \qquad \ddot{y} = -\frac{C_D}{m}\dot{y}v^{1/2} - g$$

where $v = \sqrt{\dot{x}^2 + \dot{y}^2}$. Determine the time of flight and the range R. Use $C_D = 0.03$ kg/(m·s)$^{1/2}$ and $g = 9.80665$ m/s^2.

Letting

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix}$$

the first-order differential equations are

$$\mathbf{F} = \begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \dot{y}_3 \\ \dot{y}_4 \end{bmatrix} = \begin{bmatrix} y_2 \\ -(C_D/m)y_2v^{1/2} \\ y_4 \\ -(C_D/m)y_4v^{1/2} - g \end{bmatrix}$$

```
% House Keeping
clear; clc; close all;

% Define known quantities
g=9.80665;
CD=0.03;
m=0.25;

% Specify the initial conditions and time span
y0=[0;50*cosd(30);0;50*sind(30)];
tspan=[0:0.2:8];

% Call ode45
[t,y] = ode45(@(t,y) fn(t,y,g,CD,m), tspan, y0);

% Find the flight time and range using spline interpolation and root
% solving
f1=spline(t,y(:,3));
flight_time = unique(fnzeros(f1))
f2=spline(y(:,1),y(:,3));
range = max(unique(fnzeros(f2)))

% Define function
function F = fn(t,x,g,CD,m)
F = zeros(4,1);
F(1) = x(2);
F(2) = -(CD/m)*x(2)*(x(2)^2+x(4)^2)^0.25;
F(3) = x(4);
F(4) = -(CD/m)*x(4)*(x(2)^2+x(4)^2)^0.25-g;
end
```

9.5 Implicit versus Explicit Schemes

- The Euler scheme and other versions of the Runge–Kutta method are plagued by stability problems—that is, for a time step that is too large, nonphysical oscillations occur in the solutions. The implicit method is often used to avoid these problems.

$$\frac{dy}{dt} = -a.y$$

- Euler's scheme:

$$y_{i+1} = y_i + h.f(t_i, y_i)$$

$$\left(\frac{dy}{dt}\right)_i = f(t_i, y_i)$$

$$y_{i+1} = y_i + h.f(t_i, y_i)$$

$$y_{i+1} = (1 - a.h)y_i$$

- Implicit scheme:

$$y_{i+1} = y_i + h.f(t_{i+1}, y_{i+1})$$

$$\left(\frac{dy}{dt}\right)_{i+1} = f(t_{i+1}, y_{i+1})$$

$$y_{i+1} = y_i + h.f(t_{i+1}, y_{i+1})$$

$$y_{i+1} = y_i - a.h.y_{i+1}$$

$$y_{i+1} = \frac{y_i}{1+a.h}$$

References

- *Applied Engineering Mathematics*, Brian Vick, CRC Press, 2020
- *Numerical Methods in Engineering with MATLAB*, Jaan Klusalaas, Cambridge University Press, 2012