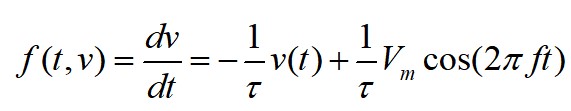
**Exercise: 1st order ODE-IVP**

**What you need to submit**

* Submit the report+source files as a zip file online (LMS)
* **Report:** including pseudocode, output results, and source codes as instructed
* **Src Code:** (1) **Assignment\_ode1\_Name\_ID.cpp, (2) myNP.h, (3) myNP.cpp**
* All the functions you have created should be updated in myNP.h and myNP.cpp **Refer to MATLAB tutorial code:** [**download source file here**](https://github.com/ykkimhgu/NumericalProg-student/tree/main/tutorial)

# Problem

Solve for the response of an RC circuit with a sinusoidal input



tau=1; T=1/tau; f=10; Vm=1; w=2\*pi\*f; a=0; b=0.1; h=0.001; v(0)=0

## Procedure

* Write down a pseudocode for the function of (1) Euler (2) Euler's modified method and (3) Runge-Kutta 2nd order method
* Complete MATLAB tutorial: [TU\_ODE\_Par1\_student.mlx](https://github.com/ykkimhgu/NumericalProg-student/tree/main/tutorial)
* Use MATLAB’s function command “ ode45()” to solve for the answer and plot the result.
* Create your own C/C++ function.
* You can plot the results in MATLAB and compare with MATLAB’s ode45().

Example:

void odeEU (double myfunc(const double t, const double y), double y[ ], double t0, double tf, double h)

void odeEM(double myfunc(const double t, const double y), double y[ ], double t0, double tf, double h) void odeRK2(double myfunc(const double t, const double y), double y[], double t0, double tf, double h, double y0);

Parameter o y: 1-D array for output y(t). The length should be predefined and fixed.

o myfunc is the user defined function that returns f(y,t)=dy/dt o t0,tf, h: start time, end time and time intervals, respectively.

* Also, create a function that calls different ODE method

void ode(double myfunc(const double t, const double y), double y[ ], double t0, double tf, double h, int method)

Parameter o method: 'EU', ‘RK2’ 'RK3' o Use preprocessor definitions such as

#define EU 0

#define RK2 1

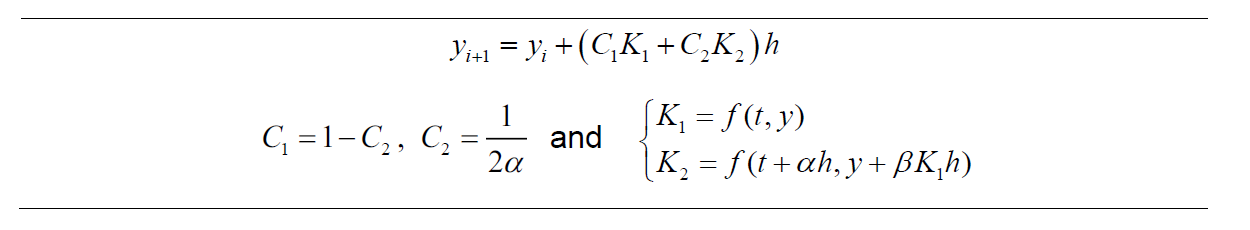
#define RK3 2

Example:

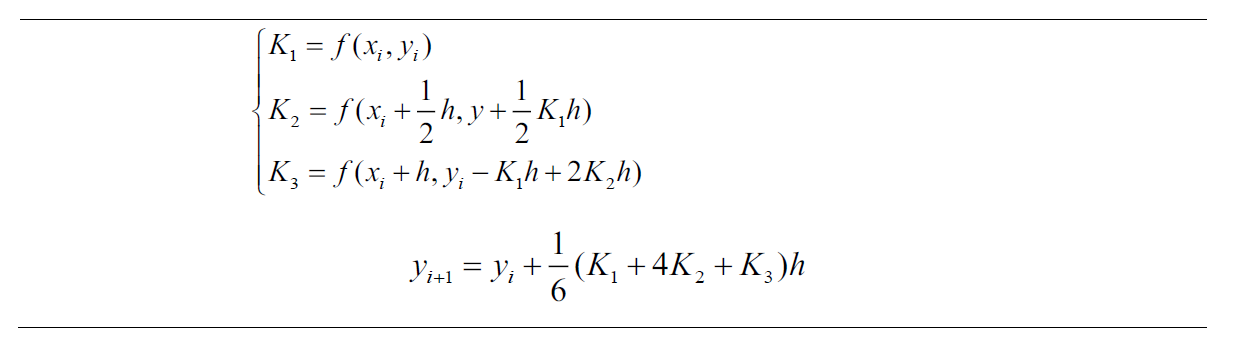
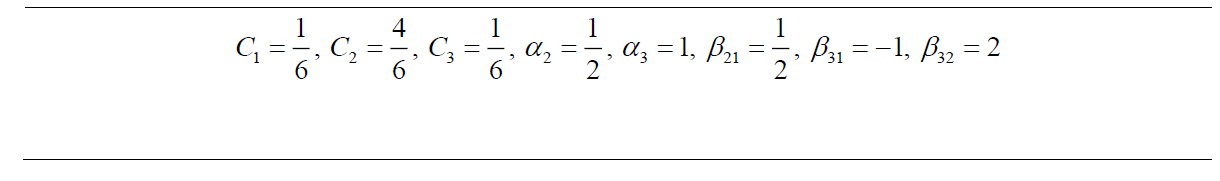
ode(myfunc(t[i], y[i]), double y[ ], t0, tf, h, RK3)

## For RK

For RK2, default value is alpha=1



For RK3, use classical third-order Runge-Kutta



**PSEUDOCODE AND SOURCE CODE**

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

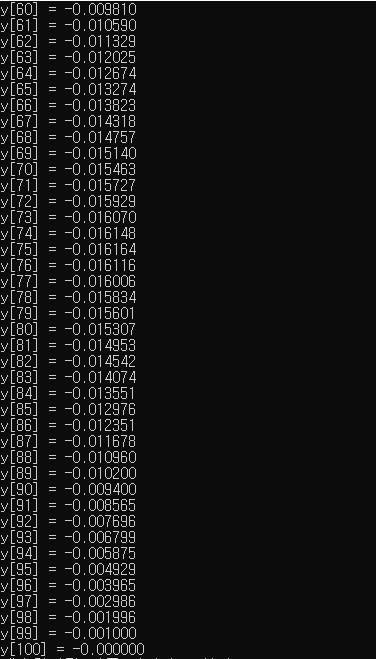
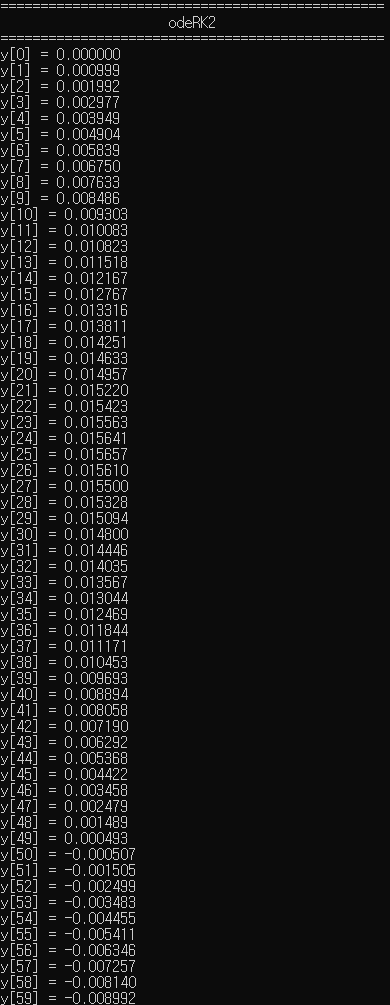
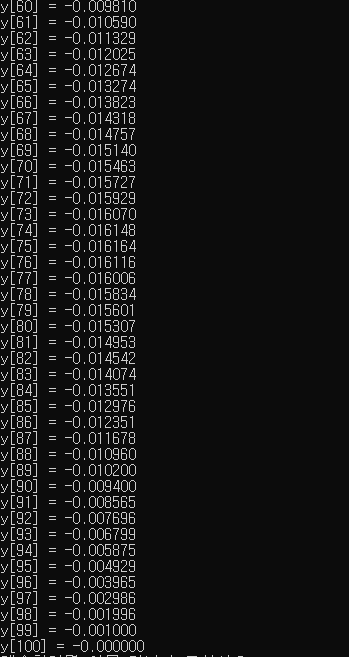
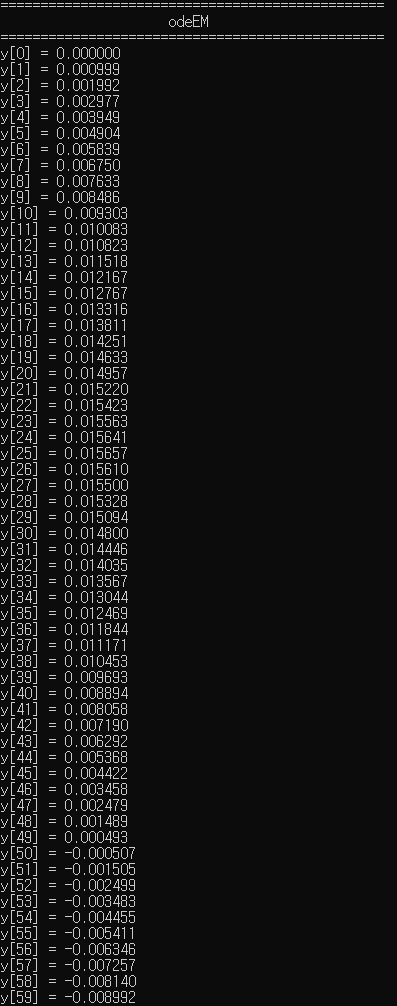
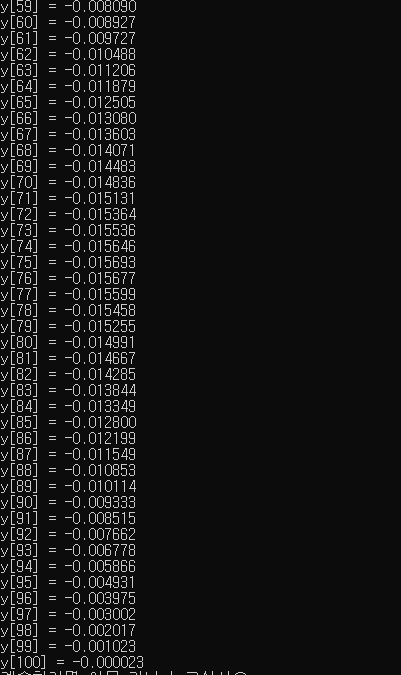
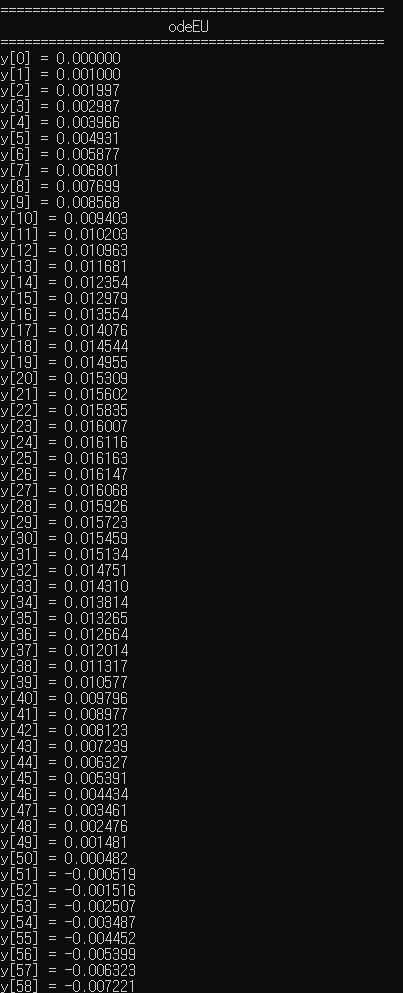
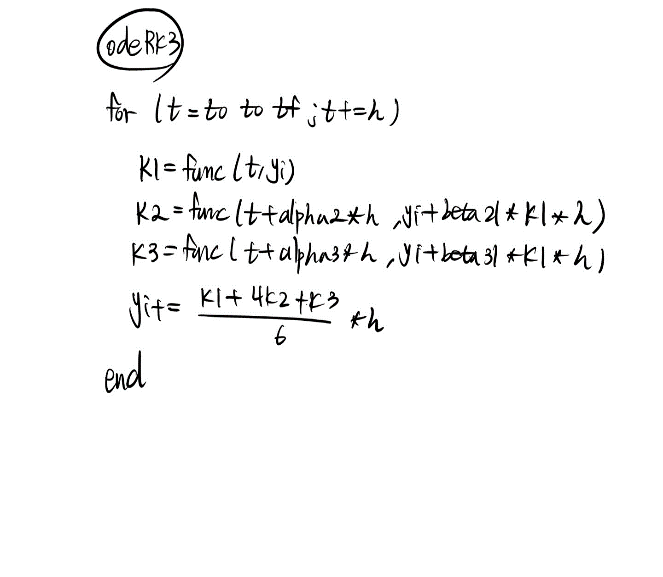
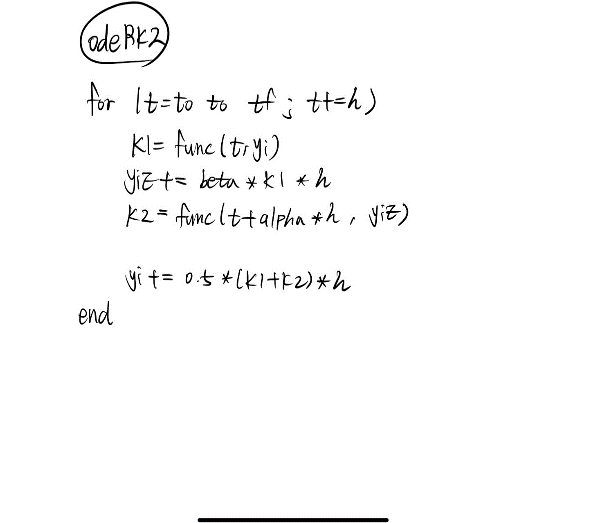
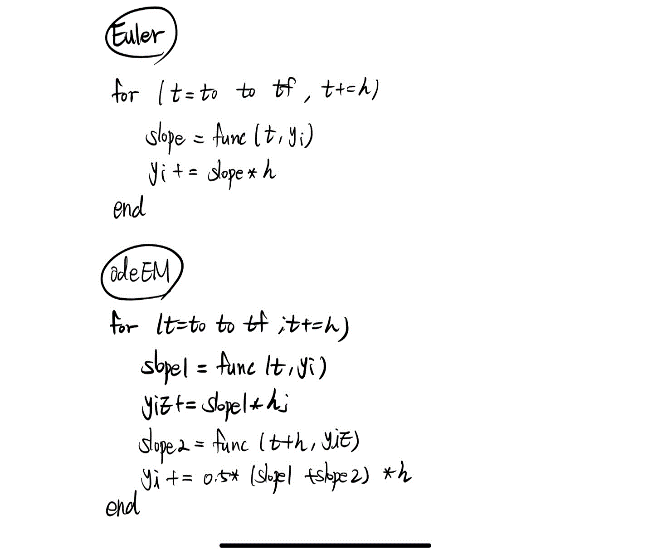
자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

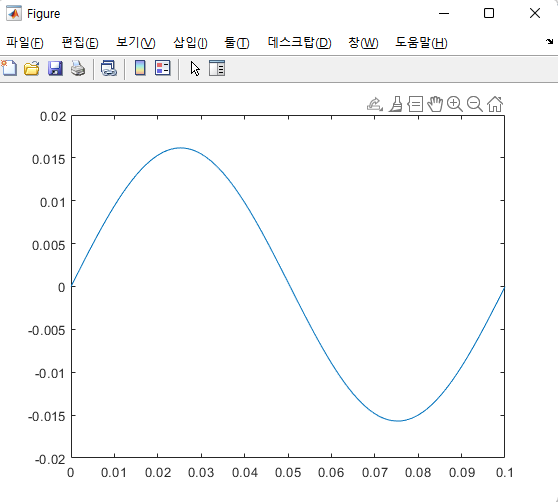
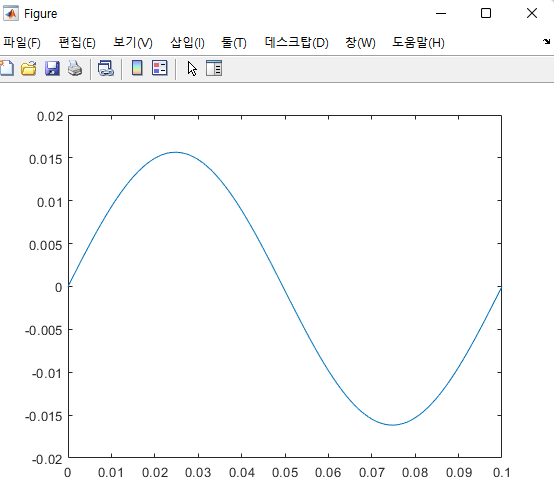
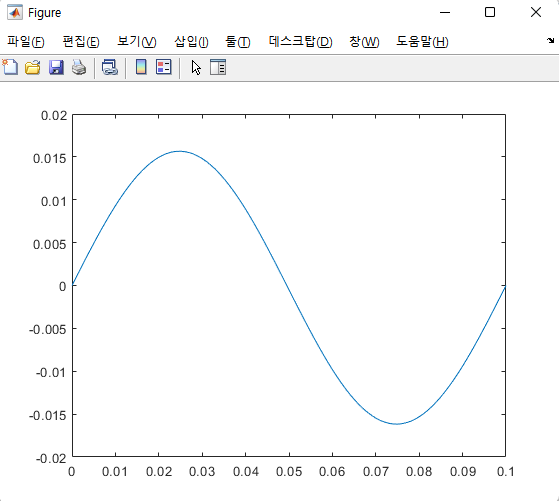
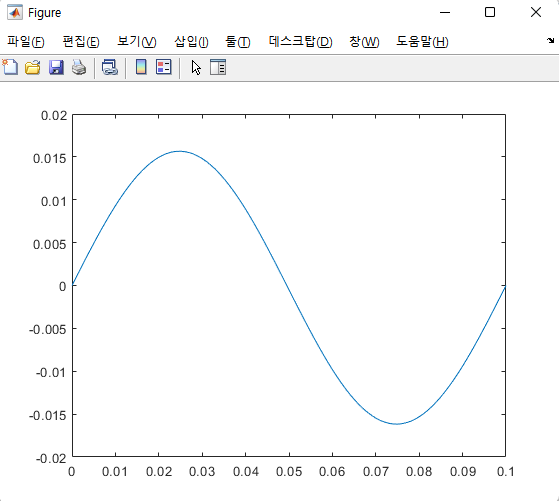
자동 생성된 설명

**텍스트이(가) 표시된 사진

자동 생성된 설명**



테이블이(가) 표시된 사진

자동 생성된 설명

< odeRK3 >

< odeRK2 >

< odeEM >

< odeEU >

