**PROJECT ASSIGNMENT C: SOLVING ORDINARY DIFFERENTIAL EQUATIONS**

**Author: Minh Hieu Do (ID: 288414)**

**Course: Numerical Methods (ENUME)**

**Advisor: Andrzej Miekina, Ph.D., Assistant Professor**

**Warsaw, 12/06/2018**

**Contents**

[1. The concise description of numerical algorithms 3](#_Toc516502874)

[a. Butcher’s tableau of coefficients 3](#_Toc516502875)

[b. Implicit Runge-Kutta formula using Butcher’s tableau of coefficients 3](#_Toc516502876)

[c. Explicit Euler method 3](#_Toc516502877)

[d. Accuracy indictors 3](#_Toc516502878)

[2. The methodology for testing numerical algorithms 4](#_Toc516502879)

[3. The results of testing numerical algorithms 5](#_Toc516502880)

[a. Problem 1, 3 5](#_Toc516502881)

[b. Problem 2, 3 9](#_Toc516502882)

[4. Conclusion 10](#_Toc516502883)

[List of references 11](#_Toc516502884)

[MATLAB Code 12](#_Toc516502885)

[Figure 1. Numerical solution obtained for **h = 0.0001** 5](file:///E:\PW\Sem%204\ENUME%20-%20Numerical%20Methods\Project%203\MinhHieuDO.docx#_Toc516502865)

[Figure 2. Numerical solution obtained for **h = 0.001** 5](file:///E:\PW\Sem%204\ENUME%20-%20Numerical%20Methods\Project%203\MinhHieuDO.docx#_Toc516502866)

[Figure 3. Numerical solution obtained for **h = 0.01** 6](file:///E:\PW\Sem%204\ENUME%20-%20Numerical%20Methods\Project%203\MinhHieuDO.docx#_Toc516502867)

[Figure 4. Numerical solution obtained for **h = 0.1** 6](file:///E:\PW\Sem%204\ENUME%20-%20Numerical%20Methods\Project%203\MinhHieuDO.docx#_Toc516502868)

[Figure 5. Numerical solution obtained for **h = 1** 7](file:///E:\PW\Sem%204\ENUME%20-%20Numerical%20Methods\Project%203\MinhHieuDO.docx#_Toc516502869)

[Figure 6. Numerical solution obtained for **h = 2** 7](file:///E:\PW\Sem%204\ENUME%20-%20Numerical%20Methods\Project%203\MinhHieuDO.docx#_Toc516502870)

[Figure 7. Numerical solution obtained for **h = 3** 8](file:///E:\PW\Sem%204\ENUME%20-%20Numerical%20Methods\Project%203\MinhHieuDO.docx#_Toc516502871)

[Figure 8. The dependence of on h 9](file:///E:\PW\Sem%204\ENUME%20-%20Numerical%20Methods\Project%203\MinhHieuDO.docx#_Toc516502872)

[Figure 9. The dependence of on h 9](file:///E:\PW\Sem%204\ENUME%20-%20Numerical%20Methods\Project%203\MinhHieuDO.docx#_Toc516502873)

# The concise description of numerical algorithms

## Butcher’s tableau of coefficients

## Implicit Runge-Kutta formula using Butcher’s tableau of coefficients

ODE systems:

may be solve by using the following algorithm:

Where:

## Explicit Euler method

## Accuracy indictors

with is the most accurate solution and is the numerical solution obtained of ODE equation for the integration step h.

# The methodology for testing numerical algorithms

* Design a MATLAB procedure to solve the equation

by applying the implicit method Lobatto IIID of order 2.

* Make the graphs of the obtained results of the above equation using method Lobatto IIID, explicit Euler method and MATLAB function ***ode45*** for some integration step *h*.
* For the MATLAB function ***ode45***, repeat several times to obtain the values of its parameters *AbsTol and RelTol* to maximize the exactness of the solution, which are the smallest values they can be.
* Perform the dependence of the accuracy indicators and of Lobatto IIID method and explicit Euler method on the integration step *h*.
* Plot graphs of the dependence of and of both methods.

# The results of testing numerical algorithms

## A screenshot of a cell phone Description generated with high confidenceProblem 1, 3

Figure 1. Numerical solution obtained for **h = 0.0001**

Figure 2. Numerical solution obtained for **h = 0.001**

A close up of a map

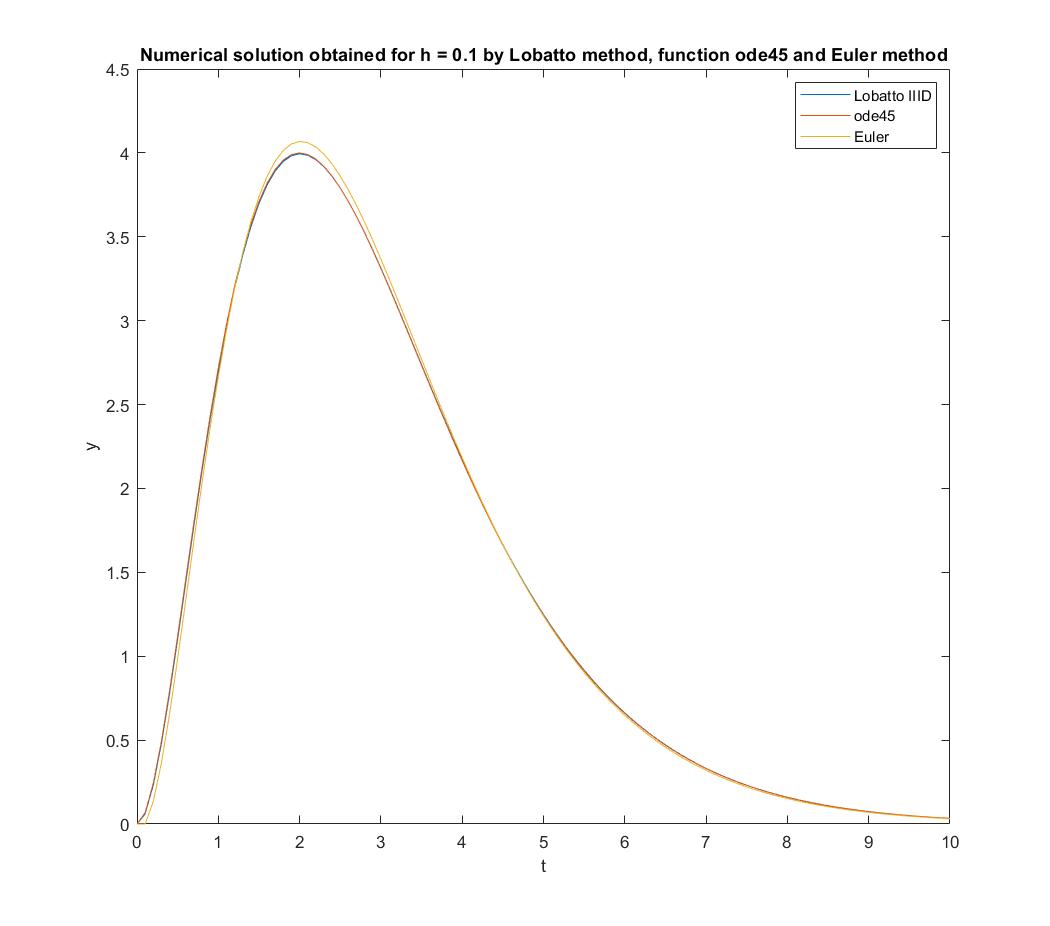
Description generated with high confidence

Figure 4. Numerical solution obtained for **h = 0.1**

Figure 3. Numerical solution obtained for **h = 0.01**

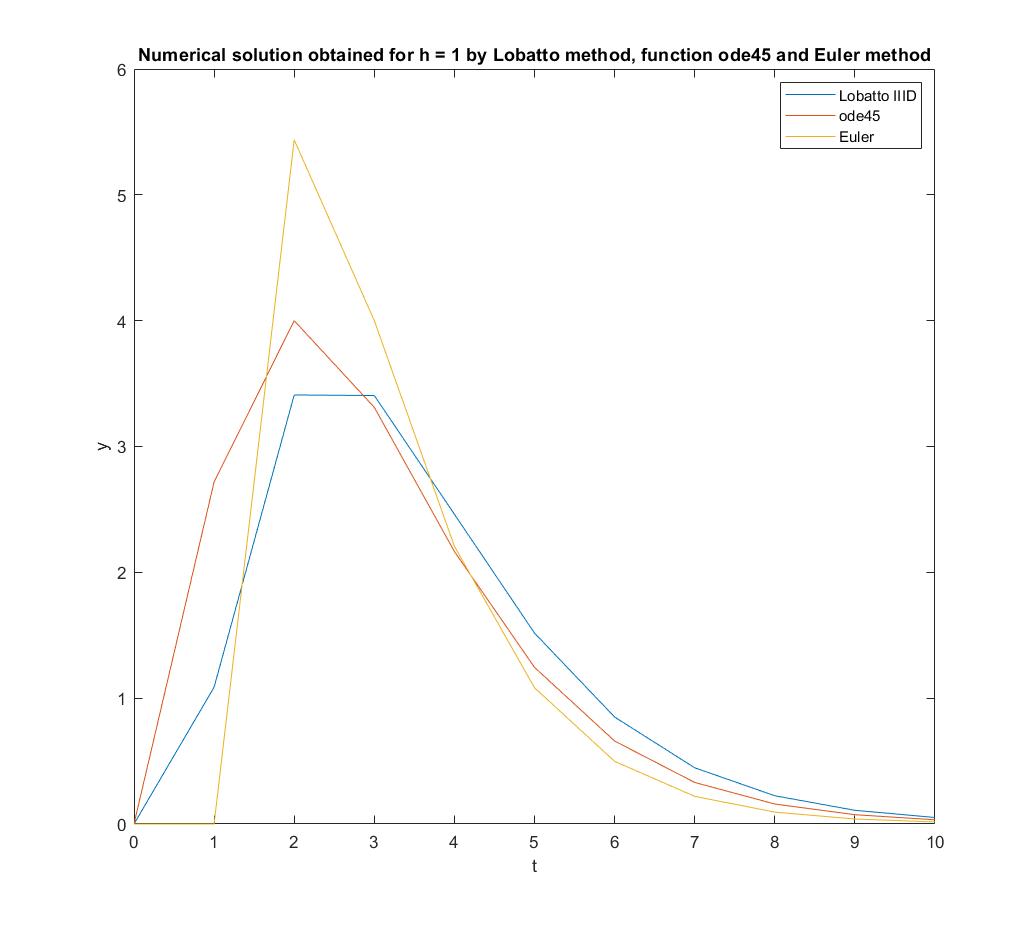
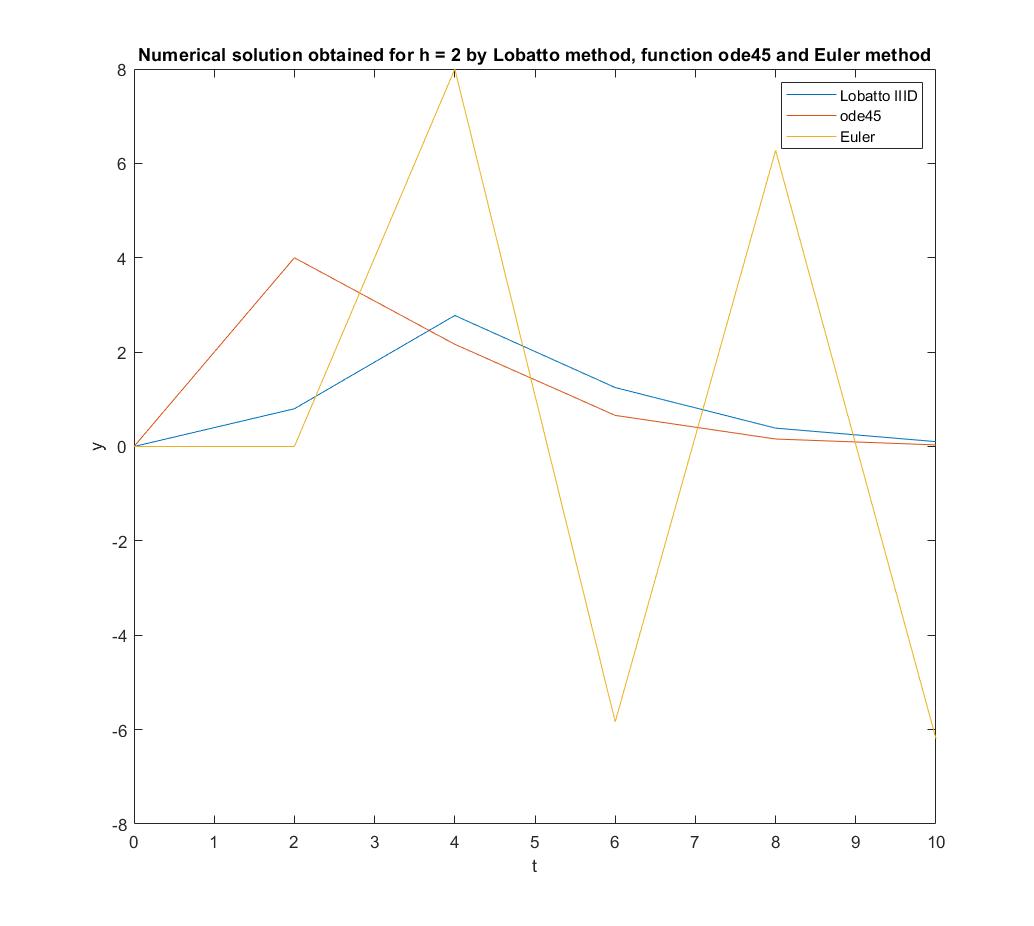
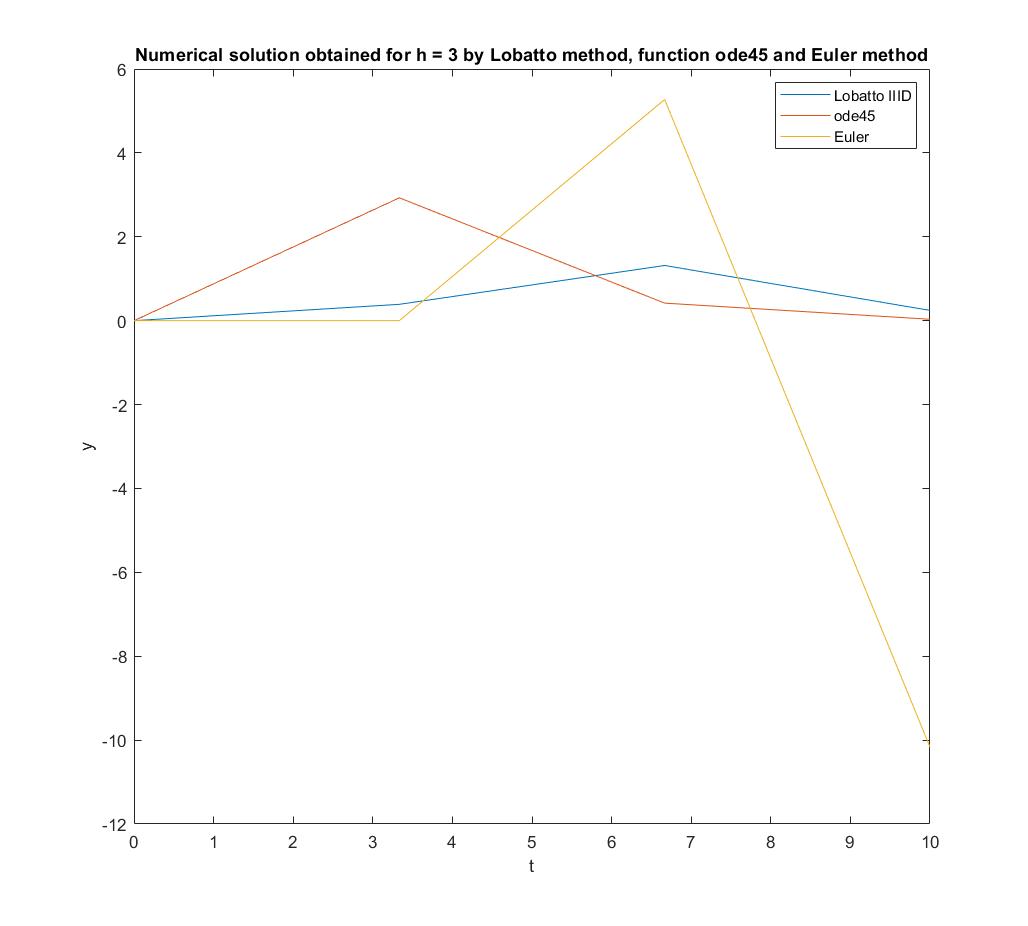


Figure 3. Numerical solution obtained for **h = 1**

Figure 4. Numerical solution obtained for **h = 2**

Figure 5. Numerical solution obtained for **h = 3**



## **A close up of a map Description generated with very high confidenceA close up of a map Description generated with very high confidence**Problem 2, 3

Figure 6. The dependence of on h

Figure 7. The dependence of on h

# Conclusion

* After using the implicit method Lobatto IIID of order 2, the explicit Euler method and the MATLAB function ***ode45*** to solve ordinary differential equation and analyzing the graphs of the dependence of and on , it can be seen that:
  + The obtained results are very close to each other for the small integration step .
  + The smaller the integration step is, the more accurate the obtained results of both Lobato IIID and Euler method are.
  + The obtained solutions of the implicit method Lobatto IIID of order 2 are more precise than the one of the explicit Euler method.
* To maximize the accuracy of the solution of the MATLAB function ***ode45***, and were chosen as the values of its parameters *RelTol* and *AbsTol* respectively.

# List of references

R. Z. Morawski, Lecture notes for ENUME students

A. Miękina, ENUME MatLab Intro 2018

MathWorks, MATLAB Documentation, <https://www.mathworks.com/help/index.html>

# MATLAB Code

clear all

close all

%%Problem 1

H = [0.0001, 0.001, 0.01, 0.1, 1, 2, 3];

hl = length(H);

for i = 1 : hl

[y, yy, T] = Lobatto(H(i));

yE = Euler(H(i));

figure(i)

plot (T, y, T, yy, T, yE)

xlabel('t')

ylabel('y')

legend('Lobatto IIID', 'ode45', 'Euler')

title(['Numerical solution obtained for h = ',num2str(H(i)),' by Lobatto method, function ode45 and Euler method']);

end

%Problem 2

H = logspace(-5, 0, 20);

hl = length(H);

accuracy = zeros(size(H));

accuracyInf = zeros(size(H));

accuracyEuler = zeros(size(H));

accuracyInfEuler = zeros(size(H));

for i = 1 : hl

h = H(i);

[y, yy, T] = Lobatto(h);

accuracy(i) = norm(y - yy) / norm(yy);

accuracyInf(i) = norm(y - yy, Inf) / norm(yy, Inf);

yEuler = Euler(h);

accuracyEuler(i) = norm(yEuler - yy) / norm(yy);

accuracyInfEuler(i) = norm(yEuler - yy, Inf) / norm(yy, Inf);

end

figure(8)

loglog(H, accuracy, H, accuracyEuler);

xlabel('h');

ylabel('\bf \delta\_{2}');

legend('\bf \delta\_{2 Lobatto}', '\bf \delta\_{2 Euler}');

title(['The dependence of \delta\_{2}(h) on h']);

figure(9)

loglog(H, accuracyInf, H, accuracyInfEuler);

xlabel('h');

ylabel('\bf \delta\_{\infty}');

legend('\bf \delta\_{\infty Lobatto}', '\bf \delta\_{\infty Euler}');

title(['The dependence of \delta\_{\infty}(h) on h']);

%Solving using the implicit method Lobatto IID of order 2

function [y, yy, T] = Lobatto(h)

T = linspace (0, 10, 10/h + 1);

y = zeros(size(T));

a = 1/2\*h;

A = [1 + a, a;

-a, 1 + a];

B = zeros(2, 1);

F = zeros(2, 1);

N = length(y);

for n = 2 : N

B = [-y(n-1) + 2 \* T(n-1) \* exp(-T(n-1) + 2);

-y(n-1) + 2 \* (T(n-1) + h) \* exp(-T(n-1) - h + 2)];

F = A\B;

y(n) = y(n-1) + h \* (1/2\*F(1) + 1/2\*F(2));

end

y = y';

opts = odeset('RelTol',2.22045e-14,'AbsTol',1e-20);

yy0 = 0;

[~, yy] = ode45(@(a, x) -x + 2\*a\*exp(-a+2), T, yy0, opts);

end

%Solving using the explicit Euler method

function [y] = Euler(h)

T = linspace (0, 10, 10/h + 1);

y = zeros(size(T));

N = length(y);

for n = 2 : N

y(n) = y(n-1) + h \* (-y(n-1) + 2 \* T(n-1) \* exp(-T(n-1) + 2));

end

y = y';

end