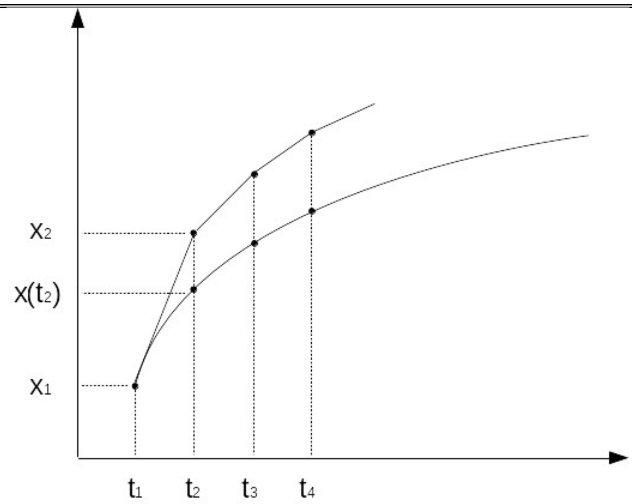
<table>
<tr><th colspan="3">TOPIC PLAN</th></tr>
<tr><td>Partner organization</td><td colspan="2">Politehnica University of Timisoara</td></tr>
<tr><td>Topic</td><td colspan="2">Ordinary Differential Equations</td></tr>
<tr><td>Lesson title</td><td colspan="2">Numerical Solutions for First-order Ordinary Differential Equations</td></tr>
<tr><td>Learning objectives</td><td>Explanation of the notion of a numerical/approximate solution for a Cauchy problem associated to a first-order ODE.
Basic idea of a numerical method for ODE
Presentation of the Euler's method: the simplest numerical method of this type – understanding of its geometrical interpretation
Possible ways to improve the precision of the approximation: the Runge-Kutta method
Implementation in Matlab of several numerical methods for ODE</td><td>Methodology
☐Modeling
☑Collaborative learning
☑Project based learning
☑Problem based learning

Strategies/Activities
☐Graphic Organizer
☐Think/Pair/Share
☑Discussion questions</td></tr>
<tr><td>Aim of the lecture / Description of the practical problem</td><td>The problem: We have a Cauchy problem associated to a first-order ODE but since the equation can not be solve by any of the exact methods studied so far we can not find its exact / analytical solution. Can we find an approximation of the solution by means of a numerical method?</td><td>Assessment for learning
☑Observations
☑Conversations
☑Work sample
☐Conference
☐Check list
☐Diagnostics</td></tr>
<tr><td>Previous knowledge assumed:</td><td>First-order ordinary differential equations of the usual types: equations with separable variables, linear equations etc.
Basic usage of the Matlab software.</td><td>Assessment as learning
☑Self-assessment
☐Peer-assessment</td></tr>
</table>

Introduction / Theoretical basics	*Question for the students:* ➢ What is an ordinary differential equation (ODE)? ➢ Which are the types od ODEs studied so far and how do we compute the exact solution in each case? *Presentation/discussion:* • If necessary a short review of the usual types of first-order ordinary differential equations (such as equations with separable variables, linear equations, homogeneous equations etc.) is presented, including a quick remainder of the methods available to find the exact solution in each case. • Next the problem given in section "Description of the practical problem" is presented and the notion of a numerical (approximate) solution for a Cauchy problem associated to a first-order ODE is introduced as a sequence of values approximating the cooresponding sequence of exact values of the solution (or, from a geometric ponit of view, as a sequence of points following the plot of the exact solution). • We consider the Cauchy problem consisting of the first-order ordinary differential equation $\frac{dx}{dt}=f(t,x)$ and of the inital condition $x(t_1)=x_1$, whose exact solution (unknown) is the function $x=x(t)$. The numerical condition wil be computed on the interval $[a,b]$ where $a=t_1$. On this interval we construct the division $\Delta: a=t_1<t_2<\ldots t_{n+1}=b$. • A numerical soution on $[a,b]$ consists of	☑Presentation ☐Graphic Organizer ☑Homework **Assessment of learning** ☐Test ☐Quiz ☑Presentation ☑Project ☐Published work

	the set of values x_i computed such as $x_i \simeq x(t_i)$. • From a geometric point of view this means that the set of points of coordinates (t_i, x_i) follow the curve which is the plot of the exact solution $x=x(t)$.	
Action	*Presentation/discussion:* • Euler's method may be developed by starting with the following observation: from the geometrical point of view, the differential equation specifies the slope of the tangent line corresponding to the plot of the solution $x=x(t)$ at the current point of coordinates (t, x). By computing the slope $m_1=f(t_1, x_1)$of the tangent at the starting point (t_1, x_1) (given by the initial condition), we can trace (plot) the tangent in (t_1, x_1). The next values (t_2, x_2) of the numerical solution are chosen such that $t_2=t_1+h$ and the point of coordinates (t_1, x_1) lies on the plot of the tangent (of slope m_1). We repeat the whole process, obtaining a sequence of polygonal lines which follows the plot of the exact solution. The process is illustrated in the following figure.	



Presentation/discussion:

- Euler's method can be easily derived as follows: we consider the equidistant division $\Delta: a=t_1<t_2<\ldots t_{n+1}=b$ where $t_{i+1}-t_i=h$ and h denotes the step of the division, $h=\frac{b-a}{n}$. Expanding in Taylor series the function $x(t_i+h)$ and keeping only the first two terms of the expansion and also taking into account the fact that $x'(t)=\frac{dx}{dt}=f(t,x)$ (the ODE itself), we obtain:

  $x(t_{i+1})=x(t_i+h)\simeq x(t_i)+\frac{x'(t_i)\cdot h^1}{1!}=$
  $=x(t_i)+f(t_i,x_i)\cdot h.$

  Denoting the approximate value of the solution $x(t_k)$ by x_k we obtain the so-called Euler's formula:

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

Question for the students:

- ➢ What are the advantages/disadvantages of a numerical solution of an ODE?

Presentation/discussion:

- Euler's method can be written as an algorithm (to be implemented later in Matlab):

Algorithm for Euler's method:

Input data: The starting point (t_1, x_1) (given by the initial condition), the interval of integration $[a, b]$ (where $a = t_1$), the number of subdivisions *n*.

Output data: The approximate values x_i of the unknown function *x* in the corresponding nodes t_i of the division.

Start

$$h = \frac{b-a}{n}$$

For i from 1 to n

$$t_{i+1} = t_i + h$$

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

Stop

Example. *We consider the Cauchy problem consisting of the first-order ordinary differential equation* $x' = -2 \cdot t \cdot x^2$ *and of the inital condition* $x(0) = 1$. *Find the exact solution of the problem, compute a few steps of a numerical solution on the [0,1] interval and compare the two solutions.*

The full solution of the example can be found in the file „Numerical Solutions for Differential Equations – Lecture.pdf".

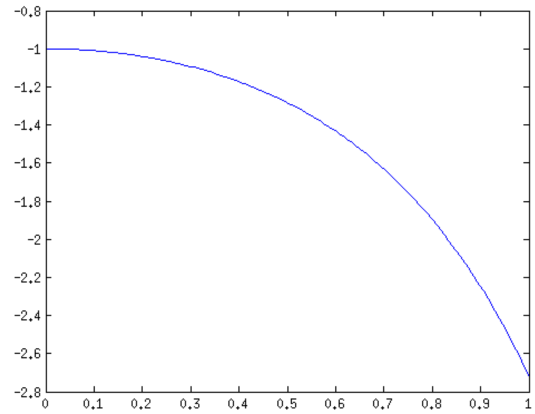
Remarks:

- Due to the nature of Euler's method, which involves a truncation of the Taylor series expansion, the value of the approximation x_i presents an inherent error. In the case of Euler's method the errors are rather large and actually increase with *i*, as clearly shown in the previous example.

	• Euler's method is a single-step method, meaning that x_{i+1} is computed as a function of x_i only, in contrast to multistep methods which compute x_{i+1} as a function of not only x_i but also x_{i-1}, x_{i-2}..., thus obtaining a more precise approximation. *Question for the students:* ➢ How can we modify Euler's formula in order to obtain a more precise approximation? *Presentation/discussion:* • Another way to obtain a better approximation is to take into account more terms in the Taylor series expansion, such as the widely-used Runge-Kutta method. The Runge-Kutta methods of order *r* are a family of numerical methods based on the folowing formula: $x_{i+1}=x_i+\sum_{j=1}^{r} c_j \cdot k_j$, where $k_1=h \cdot f(t_i, x_i)$ $k_j=h \cdot f\left(t_i+\alpha_j \cdot h, x_i+\sum_{s=1}^{j-1} \beta_{js} \cdot k_s\right)$ The constants $c_j, \alpha_j, \beta_{js}$ are determined by imposing the followingcondition: *The coefficients of the powers of* h *from the Taylor series expansion of* x_{i+1} *given by the previous formulas must coincide with the corresponding coefficients from the Taylor series expansion of* $x(t_i+h)$. The full computation of the coefficients for particular cases (such as *r=2, r=4* ...) is rather complicated and as such it is optional. For the case *r=2* it can be found in the file „Numerical	

	Solutions for Differential Equations – Lecture.pdf“. In the **Laboratory part** of this lesson, presented in detail in the file „Numerical Solutions for Differential Equations – Lecture.pdf“, we will implement in Matlab the Euler Method and we will also introduce a predefined Matlab command to find numerical solutions for ODEs (the „ode45“ command). By means of a simple example we will compare the exact solution of the problem to the numerical ones given by Euler’s Methos and by the Matlab command „ode45“. We mention the fact that all the Matlab files employied in the folowing computations can be found in the folder „Numerical Solutions for Differential Equations - Matlab Functions“. **Example.** *We consider the Cauchy problem consisting of the first-order ordinary differential equation* $x'=2\cdot t\cdot x$ *and of the inital condition* $x(0)=-1$. *Compute numerical solutions on the [0,1] interval using Euler’s method and Matlab commands. Compare the numerical solutions with the exact solution by means of graphical representations.* Solving this simple separable ODE in the usual way („pen-and-paper“) we find that the exact solution of this problem is $x(t)=-e^{t^2}$. The same solution can be found also in Matlab using the commands: `>> syms t x(t)` `>> dsolve(diff(x,t)==2*t*x,x(0)==-1)` We define this solution as a Matlab function in the file „sol.m“: `function x=sol(t)` `x=-exp(t∧2);` `end`	

The plot of this function (obtained by using the Matlab command „`fplot(@sol,[0,1])`“) is the plot of the exact solution:



In order co compute numerical solutions we define first in Matlab the function *f* from the equation:

```
function y=f(t,x)
y=2*t*x;
end
```

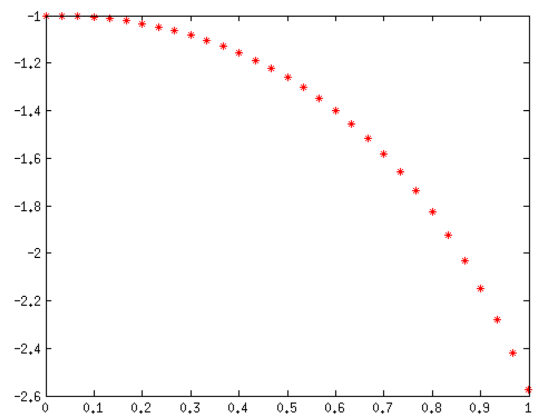
The Matlab program corresponding to Euler's Method is:

```
function [t,x]=Euler(a,b,x1,n)
h=(b-a)/n;t(1)=a;x(1)=x1;
for i=1:n
     t(i+1)=t(i)+h;
     x(i+1)=x(i)+h*f(t(i),x(i));
end
end
```

We apply this method for the case of the example:

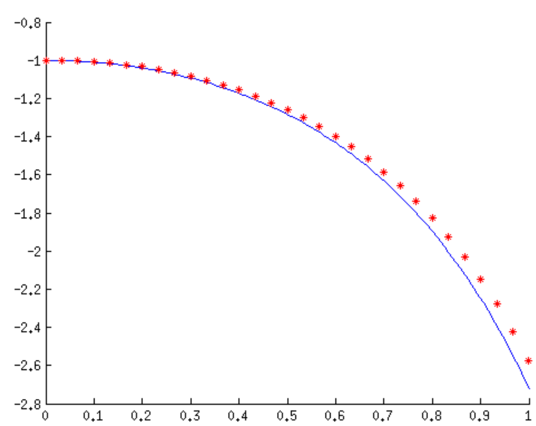
```
>> [t,x]=Euler(0,1,-1,30);
>> plot(t,x,'*')
```

We obtain the plot of the numerical solution:



We can compare this numerical solution with the exact solution by placing the previous two plots on the same figure. This is done in the next figure by using the pair of ”hold on” and ”hold off” commands:

```
>>hold on; fplot(@sol,[0,1]);
plot(t,x,'*');  hold off
```



We remark that the precision of Euler’s method is rather low, the numerical solution is relatively far from the exact solution.

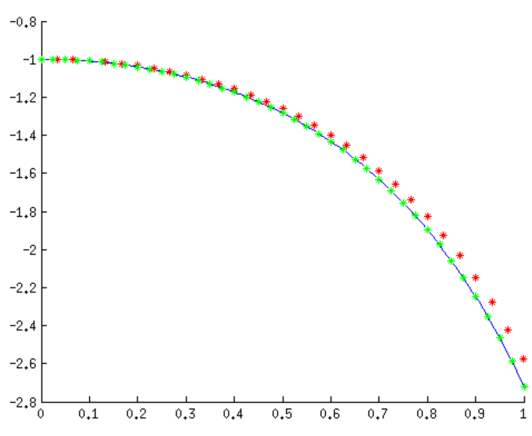
Matlab includes several commands which compute numerical solutions for differ-ential equations. The one presented here is the ”ode45” command, which has three arguments: thefunction *f* from the differential equation $x'=f(t,x)$, the interval on which the computation is performed (starting with the value of *t* from the initial condition) and the value of *x* from the initial condition. For the problem in Example 1 we write:

```
>> [tm,xm]=ode45('f',[0,1],-1);
```

In order to perform a graphical comparison of all the methods we employed so far we can use the following commands:

```
>> hold on; fplot(@sol,[0,1]);
 plot(t,x,'*r',tm,xm,'*g'); hold off
```

The following figure presents the exact solution (solid blue line), the numerical solution given by Euler’s method (dotted red line) and the numerical solution given by ”ode45” (dotted green line):



Materials / equipment / digital tools / software	*The materials for learning:* the references at the end of the document; *Equipment*: classroom, blackboard/whiteboard, different colours of chalk/markers; laboratory room with computers running the Matlab software. *Digital tools*: laptop, projector, smart board; *Software*: Mathworks Matlab	
Consolidation	• Use of materials, equipment, digital tools, software by teachers and students; • The teacher's discussion with the students through appropriate questions; • Independent solving of simple tasks by the students under the supervision of the teacher; • Given of examples by the teacher for introducing a new concept in a cooperation and a discussion with the students; • Assignment of homework by the teacher with a time limit until the next class.	

Reflections and next steps

Activities that worked	**Parts to be revisited**
After the class, the teacher, according to his personal perceptions regarding the success of the class, fills in this part.	Based on the homework done by the students, and on the questions and discussion at the beginning of the next class, the teacher concludes which parts of this class should be revised.

References

[1] P.Naslau, R. Negrea, L.Cadariu, B.Caruntu et.al - Computer Aided Mathematics (in romanian), Editura Politehnica, Timisoara 2007
[2] R.Negrea, B.Caruntu, C.Hedrea – Advanced Calculus in Engineering, Editura Politehnica, Timisoara 2009
[3] https://ocw.mit.edu/courses/mechanical-engineering/2-087-engineering-math-differential-equations-and-linear-algebra-fall-2014/index.htm