Chmielewska Urszula

300167

Advisor: dr inż. Andrzej Miękina

Numerical Methods

Solving nonlinear equations

Assignment C nr. #8

I declare that this piece of work, which is the basis for recognition of achieving learning outcomes in the Numerical Methods course, was completes on my own.

Urszula Chmielewska

300167

Theoretical introduction

There are several methods of solving equations of a given function. The aim of this project was to first obtain results by means of build in MatLab function *ode45* and later by using two other methods. One explicit and another implicit. Explicit and implicit methods are used to obtain numerical approximations to the solutions od time-dependent and partial differential equations.

In this projects calculations were performed for:

- the explicit Adams-Bashforth method
- the implicit Adams-Moulton method

The **explicit Adams-Bashforth** method is defined by the formula:

$$v_n = v_{n-1} + h \sum_{k=1}^{K} \beta_k f(t_{n-k}, v_{n-k})$$

The **implicit Adams-Moulton** method is defined by the formula:

$$v_n = v_{n-1} + h \sum_{k=0}^{K} \beta_k f(t_{n-k}, v_{n-k})$$

The difference between explicit and implicit methods, is that the implicit methods require and extra computation. It means that we need to solve also such equation at each step, to find $Y(t + \Delta t)$, using:

$$G\big(Y(t), Y(t + \Delta t)\big) = 0$$

Estimations of errors were be determined using following indicators of uncertainty:

$$\delta_2(h) = \frac{\|\hat{\mathrm{y}}(t;h) - \dot{\mathrm{y}}(t)\|_2}{\|\,\dot{\mathrm{y}}(t)\|_2}$$

$$\delta_\infty(h) = \frac{\|\hat{\mathrm{y}}(t;h) - \dot{\mathrm{y}}(t)\|_\infty}{\|\,\dot{\mathrm{y}}(t)\|_\infty}$$

Task 1

I was supposed to implement in MatLab given system of differential-algebraic equations, which were given in the form:

$$\begin{cases} \frac{d\boldsymbol{v}(t)}{dt} = \boldsymbol{A}\boldsymbol{v}(t) + \boldsymbol{b}x(t \\ y(T) = \boldsymbol{c}^T\boldsymbol{v}(t) \end{cases}$$

where A = $\begin{matrix} 0 & 1 & -1 \\ 0 & 1 & -2 \\ 300 & -21 & -38 \end{matrix}$, b = $\begin{matrix} 0 \\ 0 \\ -1 \end{matrix}$ and c = $\begin{matrix} 320 \\ -89 \\ -8 \end{matrix}$

I decided to implement functions as an anonymous functions. Declared values and formulas were necessary to implement, to be able to solve other tasks.

Task 2

In this task we were supposed to solve, implemented previously, system of equations with zero conditions for:

$x(t)$ = $\begin{cases} 0 & \text{for t} \leq 0 \\ 1 & \text{for t} > 0 \end{cases}$ $for\ t \in (0{,}5)$

and for:

$x(t)$ = $\begin{cases} 0 & \text{for t} \leq 0 \\ \exp(-t) & \text{for t} > 0 \end{cases}$ $for\ t \in (0{,}5)$

To solve such problem I was supposed to use *ode45* procedure, which is a build in function in MatLab. To be able to perform this operation, I had to set values of ***'AbsTol'*** and ***'RelTol'.***

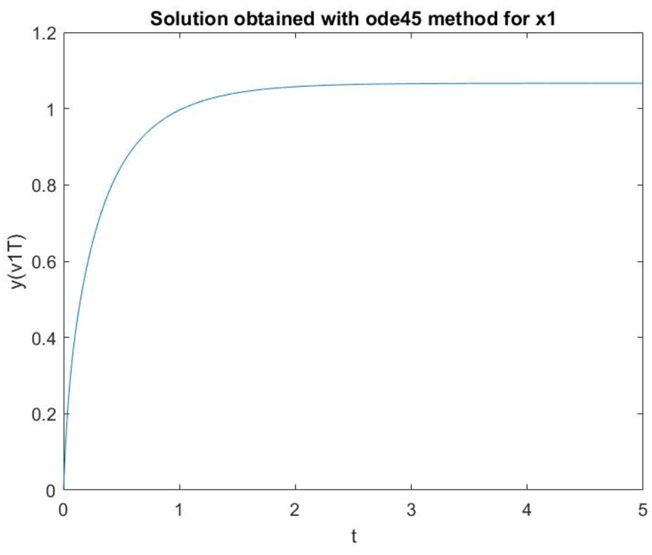
AbsTol = eps

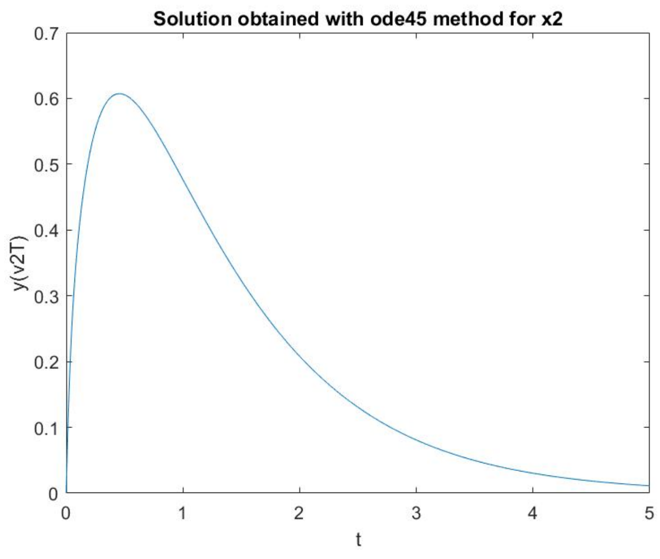
*RelTol = 2.22045 * 10^{-14}*

I decided to set value *eps*, because it means a precision of the computer. With such correctness, my result is as precise, as the computer is able to compute the result. For *RelTol* I set this value, because it is the smallest possible value, which can be set in MatLab. I wasn't able to declare smaller number. The program changes it automatically to *2.22045e-14*.

After performance of this method I obtained the values of the functions y_1 and y_2 with respect to x_1 and x_2. These solutions will be treater later as exact solutions.

Below, I present graphs visualizing the solution obtained by means of *ode45* method for two x values, given in the task.





Task 3

Finally, in the third task I was supposed to solve given system of equations by means of explicit Adams-Bashforth method and implicit Adams-Moulton method and test it for various values of the integration step h ϵ [h_{min},h_{max}].

In MatLab solution I use similar approach for both methods solutions. I created *while* loop which performs operations until $h<h_{max.}$. Step of *h* is set to increase by 0,001. Adams-Bashforth method required from me to almost manually implement calculations of values of v_n, because for each K, all β_k values changes. Similar in Adams-Moultons, however here I had K=3 given in the task. Inside these loops I implemented also *for* loop, working in the length of time and another while loop, which calculates the final sum, as a result. I decided to store my results in the same way as I did it in the previous project. I store solutions in arrays, using counter, to know on which position put the next obtained value, for each indicator *h* value.

When I was coding these functions I was looking for h value, for which I will notice significant changes. I was trying to find the proper indicator *h* value by means of trial and error method. The value of indicators started rapidly increasing for such parameters:

Explicit:

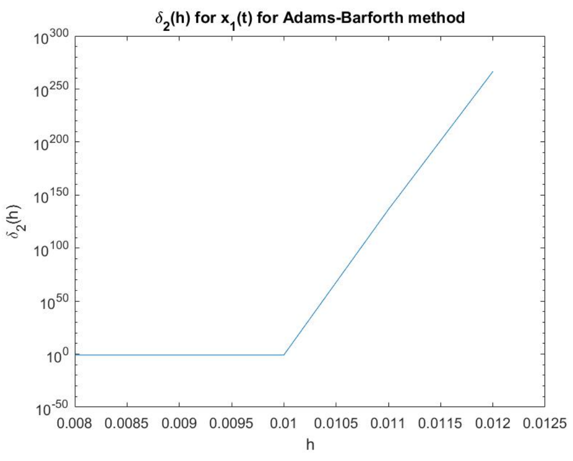
hmin= 0,008

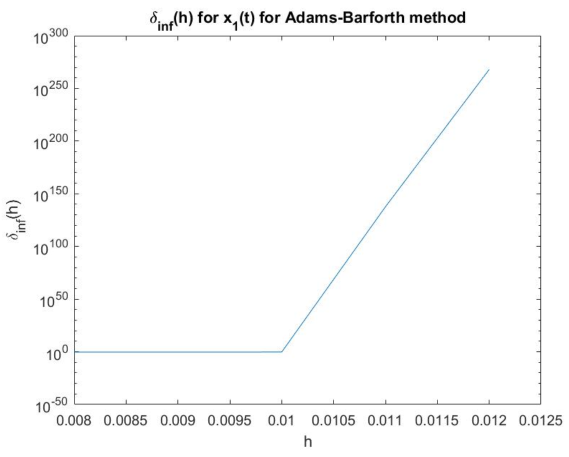
hmax= 0,02

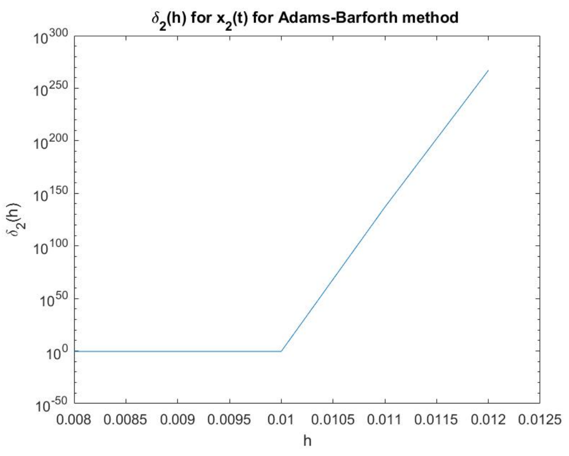
Implicit:

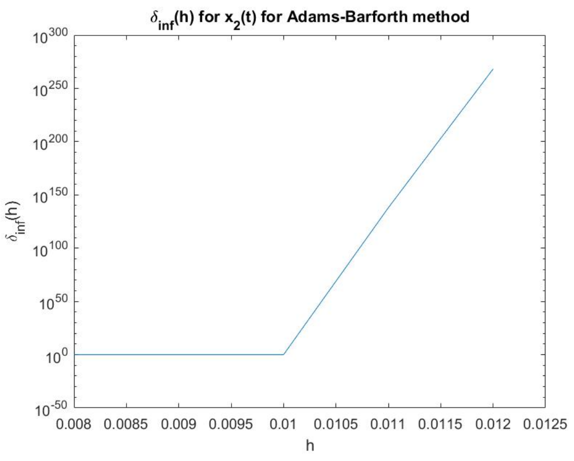
hmin= 0,16

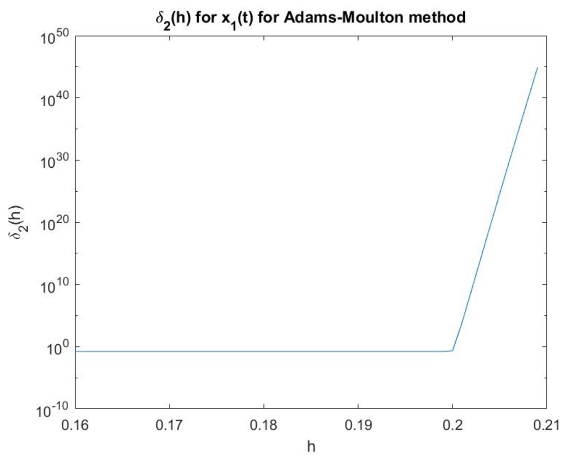
hmax= 0,21

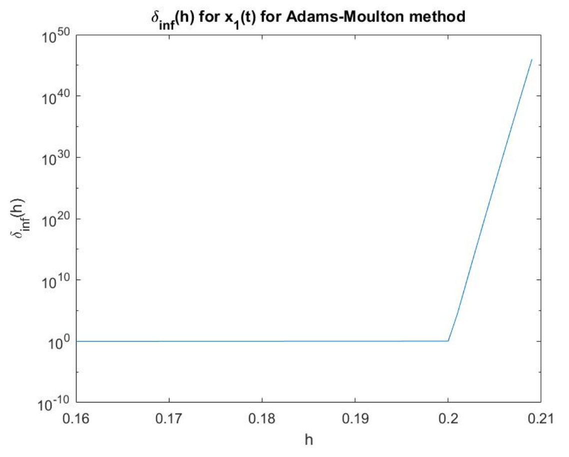

δ_2(h) for x_1(t) for Adams-Barforth method
δ_2(h)
h
10^{300}
10^{250}
10^{200}
10^{150}
10^{100}
10^{50}
10^{0}
10^{-50}
0.008
0.0085
0.009
0.0095
0.01
0.0105
0.011
0.0115
0.012
0.0125

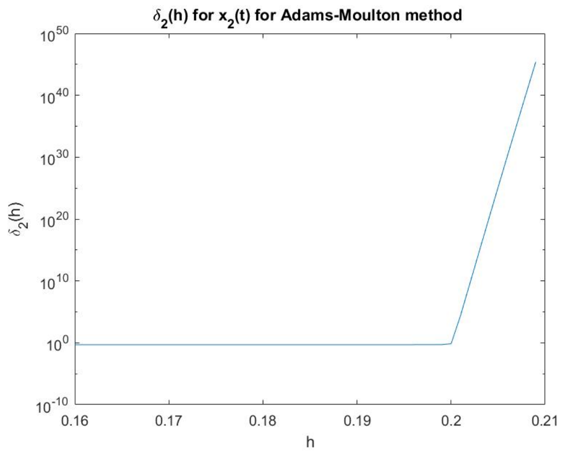

δ_{inf}(h) for x_1(t) for Adams-Barforth method
δ_{inf}(h)
h
10^{300}
10^{250}
10^{200}
10^{150}
10^{100}
10^{50}
10^{0}
10^{-50}
0.008
0.0085
0.009
0.0095
0.01
0.0105
0.011
0.0115
0.012
0.0125

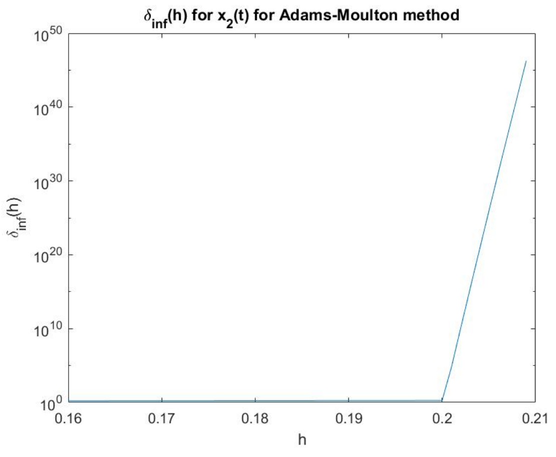

δ_2(h) for x_2(t) for Adams-Barforth method
δ_2(h)
h
10^{300}
10^{250}
10^{200}
10^{150}
10^{100}
10^{50}
10^{0}
10^{-50}
0.008
0.0085
0.009
0.0095
0.01
0.0105
0.011
0.0115
0.012
0.0125


δ_{inf}(h) for x_2(t) for Adams-Barforth method
δ_{inf}(h)
h
10^{300}
10^{250}
10^{200}
10^{150}
10^{100}
10^{50}
10^{0}
10^{-50}
0.008
0.0085
0.009
0.0095
0.01
0.0105
0.011
0.0115
0.012
0.0125


δ_2(h) for x_1(t) for Adams-Moulton method
δ_2(h)
h
10^{50}
10^{40}
10^{30}
10^{20}
10^{10}
10^{0}
10^{-10}
0.16
0.17
0.18
0.19
0.2
0.21


δ_{inf}(h) for x_1(t) for Adams-Moulton method
δ_{inf}(h)
h
10^{50}
10^{40}
10^{30}
10^{20}
10^{10}
10^{0}
10^{-10}
0.16
0.17
0.18
0.19
0.2
0.21





Final conclusions

For values smaller than h_{max}, the indicator of uncertainty are on the similar, low level. These values increase rapidly when the value 0,01 (in case of Adams-Bashforth method) or the value 0,2 (in case od Adams-Moulton method) are crossed. From these observations I concluded that for such high values of the indicator h, the above formulas are unstable.

APPENDIX – SOURCE CODE

```
close all
clear
clc
%===========================TASK 1===========================
A=[0,1,-1;0,1,-2;300,-21,-38];
b=[0;0;-1];
c=[320;-89;-8];

h=0.001;
t=0:h:5;
v0=[0;0;0];

x1=@(t) 1*(t>0);
x2=@(t) exp(-t)*(t>0);
y=@(v) c'*v;

dvdt1=@(t,v) A*v + b*x1(t);
dvdt2=@(t,v) A*v + b*x2(t);

%===========================TASK 2===========================

parameters = odeset('AbsTol', eps, 'RelTol', 2.22045e-14);
[t1, v1] = ode45(dvdt1, t, v0, parameters);
[t2, v2] = ode45(dvdt2, t, v0, parameters);
yexact1=c'*v1';
yexact2=c'*v2';

figure(1)
plot(t1,y(v1'));
%plot(t1, v1(:,1));
%plot(t1, v1(:,2));
%plot(t1, v1(:,3));
hold on;
title('Solution obtained with ode45 method for x1');
xlabel('t');
ylabel('y(v1T)');
hold off;

figure(2)
plot(t2,y(v2'));
%plot(t2, v2(:,1));
%plot(t2, v2(:,2));
%plot(t2, v2(:,3));
hold on;
title('Solution obtained with ode45 method for x2');
xlabel('t');
ylabel('y(v2T)');
hold off;

%===========================TASK 3===========================
%Adams-Bashforth method

h=0.008;
hmax=0.02;
beta4=[55/24 -59/24 37/24 -9/24];
counter=1;

while h<hmax
    vb1(:,1)=v0;
    vb2(:,1)=v0;

    vb1(:,2)=vb1(:,1)+h*1*(A*vb1(:,1) + b*x1(t(1)));
    vb2(:,2)=vb2(:,1)+h*(A*vb2(:,1) + b*x2(t(1)));

    vb1(:,3)=vb1(:,2)+h*(3/2)*(A*vb1(:,2) + b*x1(t(1)))+h*(-1/2)*(A*vb1(:,1) +
b*x1(t(1)));
```

```matlab
    vb2(:,3)=vb2(:,2)+h*(3/2)*(A*vb2(:,2) + b*x2(t(1)))+h*(-1/2)*(A*vb2(:,1) +
b*x2(t(1)));

    vb1(:,4)=vb1(:,3)+h*(23/12)*(A*vb1(:,3) + b*x1(t(1)))+h*(-16/12)*(A*vb1(:,2) +
b*x1(t(1)))+h*(5/12)*(A*vb1(:,1) + b*x1(t(1)));
    vb2(:,4)=vb2(:,3)+h*(23/12)*(A*vb2(:,3) + b*x2(t(1)))+h*(-16/12)*(A*vb2(:,2) +
b*x2(t(1)))+h*(5/12)*(A*vb2(:,1) + b*x2(t(1)));

    for i=5:length(t)
        k=1;
        sum1=0;
        sum2=0;
        while k<=4
            sum1=sum1+beta4(k)*(A*vb1(:,i-k)+b*x1(t(i-k)));
            sum2=sum2+beta4(k)*(A*vb2(:,i-k)+b*x2(t(i-k)));
            k=k+1;
        end
        vb1(:,i)=vb1(:,i-1)+h*sum1;
        vb2(:,i)=vb2(:,i-1)+h*sum2;
    end
    yb1=c'*vb1;
    yb2=c'*vb2;

    deltab1(counter)=(norm(yb1-yexact1)/norm(yexact1));
    deltab2(counter)=(norm(yb2-yexact2)/norm(yexact2));
    deltab1inf(counter)=(norm(yb1-yexact1,"inf")/norm(yexact1,"inf"));
    deltab2inf(counter)=(norm(yb2-yexact2,"inf")/norm(yexact2,"inf"));
    hball(counter)=h;
    h=h+0.001;
    counter=counter+1;
end


%Adams-Moulton method
h=0.16;
hmax=0.21;
beta3 = [5/12 8/12 -1/12];
counter=1;

while h<hmax
    vm1(:,1)=v0;
    vm2(:,1)=v0;

    vm1(:,2)=(eye(3)-h*(1/2)*A)\(vm1(:,1)+h*(1/2)*(2*b*x1(t(1))+A*vm1(:,1)));
    vm2(:,2)=(eye(3)-h*(1/2)*A)\(vm2(:,1)+h*(1/2)*(2*b*x2(t(1))+A*vm2(:,1)));

    for i=3:length(t)
        vm1(:,i)=(eye(3)-h*beta3(1)*A)\(vm1(:,i-
1)+h*beta3(1)*b*x1(t(i))+h*beta3(2)*(A*vm1(:,i-1)+b*x1(t(i-
1)))+h*beta3(3)*(A*vm1(:,i-2)+b*x1(t(i-2))));
        vm2(:,i)=(eye(3)-h*beta3(1)*A)\(vm2(:,i-
1)+h*beta3(1)*b*x2(t(i))+h*beta3(2)*(A*vm2(:,i-1)+b*x2(t(i-
1)))+h*beta3(3)*(A*vm2(:,i-2)+b*x2(t(i-2))));
    end
    ym1=c'*vm1;
    ym2=c'*vm2;

    deltam1(counter)=(norm(ym1-yexact1)/norm(yexact1));
    deltam2(counter)=(norm(ym2-yexact2)/norm(yexact2));
    deltam1inf(counter)=(norm(ym1-yexact1,"inf")/norm(yexact1,"inf"));
    deltam2inf(counter)=(norm(ym2-yexact2,"inf")/norm(yexact2,"inf"));
    hmall(counter)=h;
    h=h+0.001;
    counter=counter+1;
end

%===========================PRINT GRAPHS==========================
figure(3)
semilogy(hball,deltab1,'-');
title("\delta_{2}(h) for x_{1}(t) for Adams-Barforth method");
ylabel("\delta_{2}(h)");
xlabel("h");
figure(4)
semilogy(hball,deltab2,'-');
title("\delta_{2}(h) for x_{2}(t) for Adams-Barforth method ");
ylabel("\delta_{2}(h)");
xlabel("h");
figure(5)
semilogy(hball,deltab1inf,'-');
title("\delta_{inf}(h) for x_{1}(t) for Adams-Barforth method");
ylabel("\delta_{inf}(h)");
xlabel("h");
figure(6)
semilogy(hball,deltab2inf,'-');
title("\delta_{inf}(h) for x_{2}(t) for Adams-Barforth method");
ylabel("\delta_{inf}(h)");
xlabel("h");


figure(7)
semilogy(hmall,deltam1,'-');
title("\delta_{2}(h) for x_{1}(t) for Adams-Moulton method");
ylabel("\delta_{2}(h)");
xlabel("h");
figure(8)
semilogy(hmall,deltam2,'-');
title("\delta_{2}(h) for x_{2}(t) for Adams-Moulton method ");
ylabel("\delta_{2}(h)");
xlabel("h");
figure(9)
semilogy(hmall,deltam1inf,'-');
title("\delta_{inf}(h) for x_{1}(t) for Adams-Moulton method");
ylabel("\delta_{inf}(h)");
xlabel("h");
figure(10)
semilogy(hmall,deltam2inf,'-');
title("\delta_{inf}(h) for x_{2}(t) for Adams-Moulton method");
ylabel("\delta_{inf}(h)");
xlabel("h");
```