CFD Homework 2 - BRYAN ACOSTA

Contents

- Solving ODES

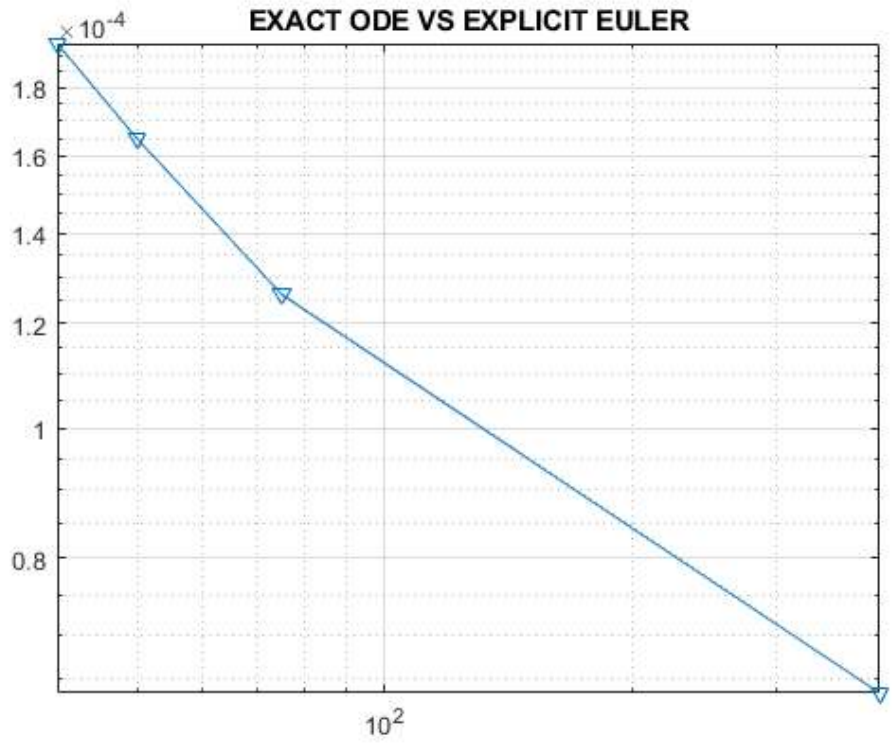
Solving ODES

```
clear
clc
close all
```

```
clear
clc
close all

x1 = linspace(0,1,30);
x2 = linspace(0,1,40);
x3 = linspace(0,1,50);
x4 = linspace(0,1,75);
x5 = linspace(0,1,150);
x6 = linspace(0,1,400);

diffequation=@(x,y) -50*(y - cos(x));
```

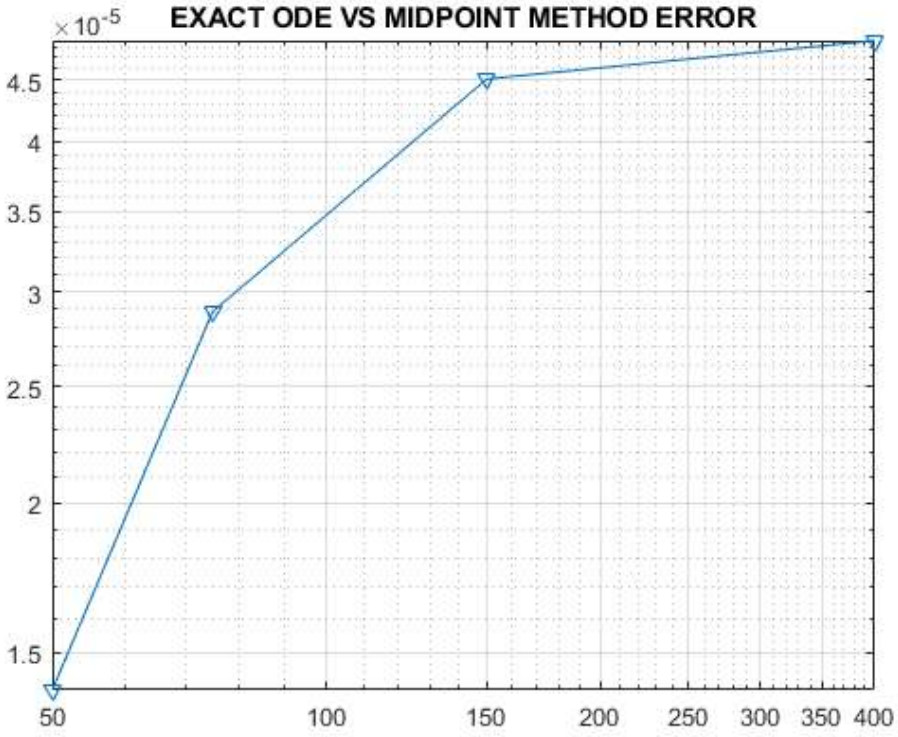
EXACT ODE VS EXPLICIT EULER

```
figure(1)
title('EXACT ODE VS EXPLICIT EULER')
exactval = exact_solution(diffequation,x2);
estval = Explicit_Euler(diffequation,x2);
input1 = abs(exactval(end) - estval(end));

exactval = exact_solution(diffequation,x3);
estval = Explicit_Euler(diffequation,x3);
input2 = abs(exactval(end) - estval(end));

exactval = exact_solution(diffequation,x4);
estval = Explicit_Euler(diffequation,x4);
input3 = abs(exactval(end) - estval(end));

exactval = exact_solution(diffequation,x6);
estval = Explicit_Euler(diffequation,x6);
input4 = abs(exactval(end) - estval(end));

loglog([length(x2),length(x3),length(x4),length(x6)],[input1,input2,input3,input4],'v-')
grid on
title('EXACT ODE VS EXPLICIT EULER')
```



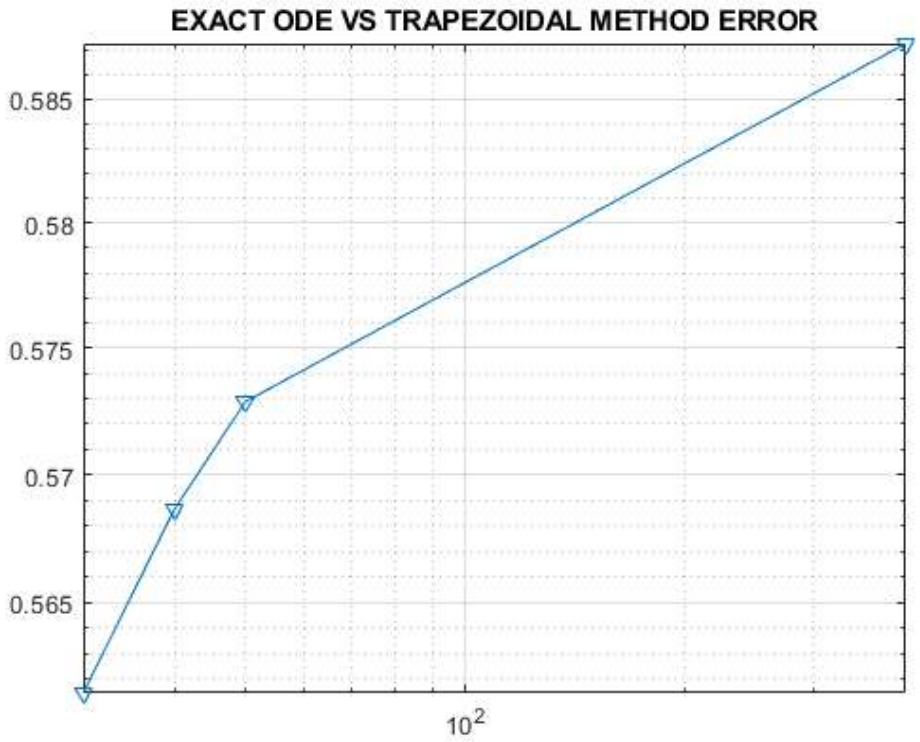

```
% EXACT ODE VS IMPLICIT EULER
figure(2)
exactval = exact_solution(diffequation,x2);
estval = Implicit_Euler(diffequation,x2);
input1 = abs(exactval(end) - estval(end));

exactval = exact_solution(diffequation,x3);
estval = Implicit_Euler(diffequation,x3);
input2 = abs(exactval(end) - estval(end));

exactval = exact_solution(diffequation,x4);
estval = Implicit_Euler(diffequation,x4);
input3 = abs(exactval(end) - estval(end));

exactval = exact_solution(diffequation,x6);
estval = Implicit_Euler(diffequation,x6);
input4 = abs(exactval(end) - estval(end));

loglog([length(x2),length(x3),length(x4),length(x6)],[input1,input2,input3,input4],'v-')
grid on
title('EXACT ODE VS IMPLICIT EULER ERROR')
```


```
% EXACT ODE VS MIDPOINT METHOD
figure(3)
exactval = exact_solution(diffequation,x3);
estval = Midpoint(diffequation,x3);
input1 = abs(exactval(end) - estval(end));

exactval = exact_solution(diffequation,x4);
estval = Midpoint(diffequation,x4);
input2 = abs(exactval(end) - estval(end));

exactval = exact_solution(diffequation,x5);
estval = Midpoint(diffequation,x5);
input3 = abs(exactval(end) - estval(end));

exactval = exact_solution(diffequation,x6);
estval = Midpoint(diffequation,x6);
input4 = abs(exactval(end) - estval(end));

loglog([length(x3),length(x4),length(x5),length(x6)],[input1,input2,input3,input4],'v-')
grid on
title('EXACT ODE VS MIDPOINT METHOD ERROR')
```



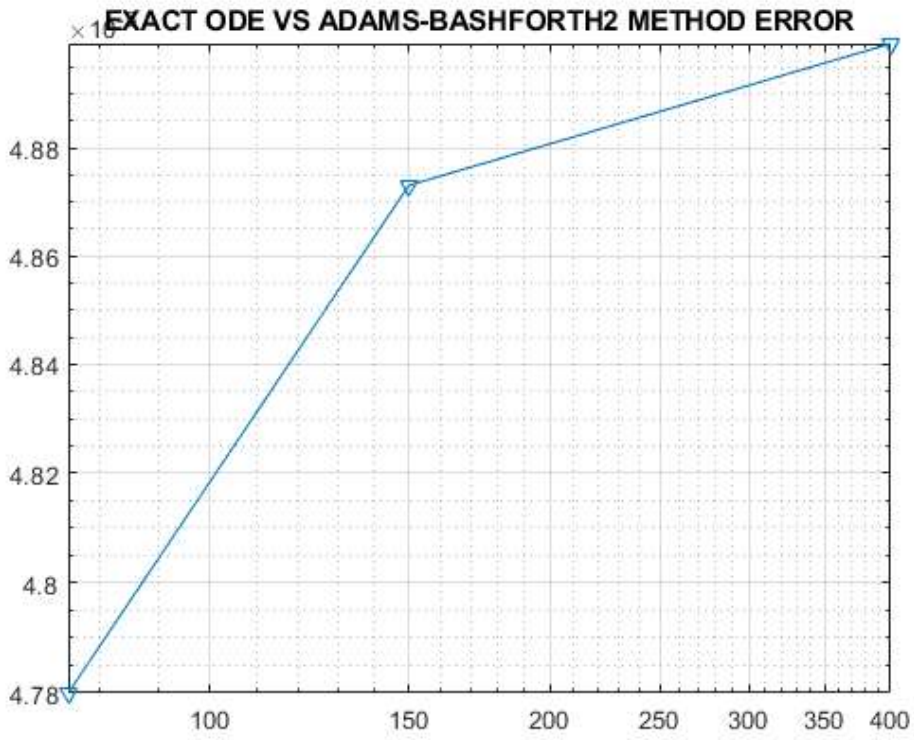
EXACT ODE VS TRAPEZOIDAL METHOD

```
figure(4)
exactval = exact_solution(diffequation,x1);
estval = trapezoidal(diffequation,x1);
input1 = abs(exactval(end) - estval(end));

exactval = exact_solution(diffequation,x2);
estval = trapezoidal(diffequation,x2);
input2 = abs(exactval(end) - estval(end));

exactval = exact_solution(diffequation,x3);
estval = trapezoidal(diffequation,x3);
input3 = abs(exactval(end) - estval(end));

exactval = exact_solution(diffequation,x6);
estval = trapezoidal(diffequation,x6);
input4 = abs(exactval(end) - estval(end));

loglog([length(x1),length(x2),length(x3),length(x6)],[input1,input2,input3,input4],'v-')
grid on
title('EXACT ODE VS TRAPEZOIDAL METHOD ERROR')
```



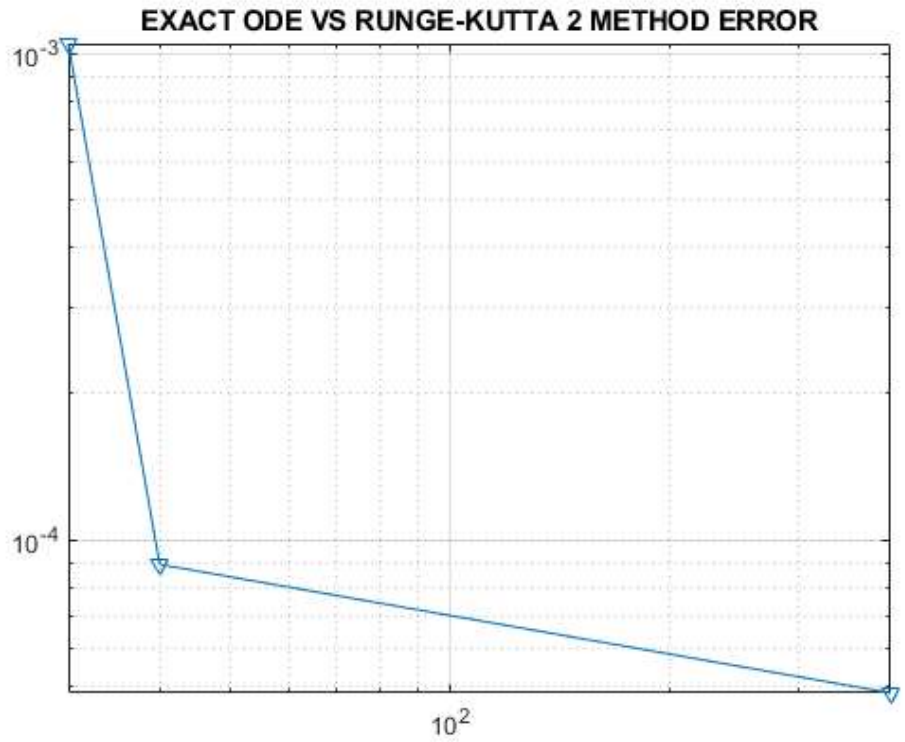
EXACT ODE VS ADAMS-BASHFORTH2 METHOD

```
figure(5)

exactval = exact_solution(diffequation,x4);
estval = AdamsB2(diffequation,x4);
input1 = abs(exactval(end) - estval(end));

exactval = exact_solution(diffequation,x5);
estval = AdamsB2(diffequation,x5);
input3 = abs(exactval(end) - estval(end));

exactval = exact_solution(diffequation,x6);
estval = AdamsB2(diffequation,x6);
input4 = abs(exactval(end) - estval(end));

loglog([length(x4),length(x5),length(x6)],[input1,input3,input4],'v-')
grid on

title('EXACT ODE VS ADAMS-BASHFORTH2 METHOD ERROR')
```



EXACT ODE VS RUNGE-KUTTA 2 METHOD

```
figure(6)

exactval = exact_solution(diffequation,x1);
estval = RK2(diffequation,x1);
input1 = abs(exactval(end) - estval(end));

exactval = exact_solution(diffequation,x2);
estval = RK2(diffequation,x2);
input3 = abs(exactval(end) - estval(end));

exactval = exact_solution(diffequation,x6);
estval = RK2(diffequation,x6);
input4 = abs(exactval(end) - estval(end));

loglog([length(x1),length(x2),length(x6)],[input1,input3,input4],'v-')
grid on
title('EXACT ODE VS RUNGE-KUTTA 2 METHOD ERROR')
```



EXACT ODE VS RUNGE-KUTTA 4 METHOD

```
figure(7)

exactval = exact_solution(diffequation,x1);
estval = RK4(diffequation,x1);
input1 = abs(exactval(end) - estval(end));

exactval = exact_solution(diffequation,x2);
estval = RK4(diffequation,x2);
input2 = abs(exactval(end) - estval(end));

exactval = exact_solution(diffequation,x3);
estval = RK4(diffequation,x3);
input3 = abs(exactval(end) - estval(end));

exactval = exact_solution(diffequation,x6);
estval = RK4(diffequation,x6);
input4 = abs(exactval(end) - estval(end));

loglog([length(x1),length(x2),length(x6)],[input1,input3,input4],'v-')
grid on
title('EXACT ODE VS RUNGE-KUTTA 4 METHOD ERROR')
```

```
function [y] = exact_solution(diffeq, xspan)
    [~, y] = ode45(diffeq,xspan, 0);

end
function [output] = Explicit_Euler(diffeq, xspan)
    jump = xspan(2);
    output = xspan.*0;
    for i = 1: (length(xspan)-1)
           output(i+1) = output(i) + jump*diffeq(xspan(i), output(i));
    end
end
function [output] =Implicit_Euler(diffeq, xspan)
    jump = xspan(2);
    misc = xspan.*0;
    output = xspan.*0;
    for i = 1: (length(xspan)-1)
           misc(i+1) = misc(i) + diffeq(xspan(i), misc(i))*jump;
           output(i+1) = output(i)+ jump* diffeq(xspan(i+1), misc(i+1));
    end
end
function [output] =Midpoint(diffeq, xspan)
    jump = xspan(2);
    halfjump = jump/2;
    output = xspan.*0;
    for i = 1: (length(xspan)-1)
           output(i+1) = output(i) + jump*diffeq(xspan(i)+halfjump,output(i)+halfjump*diffeq(xspan(i),output(i)));
    end
end
function [output] =trapezoidal(diffeq, xspan)
    jump = xspan(2);
    halfjump = jump/2;
    output = xspan.*0;
    for i = 1: (length(xspan)-1)
           output(i+1) = output(i)+ halfjump*(diffeq(xspan(i), output(i))+ diffeq(xspan(i+1), output(i+1)));
    end
end
```

```
function [output] =AdamsB2(diffeq, xspan)
    jump = xspan(2);
    output = xspan.*0;
    for i = 1: (length(xspan)-2)
           output(i+2) = output(i+1) + 0.5*jump*(3*diffeq(xspan(i+1), output(i+1))-diffeq(xspan(i),output(i)));
    end
end
function [output] = RK2(diffeq, xspan)
    jump = xspan(2);
    output = xspan.*0;
    for i = 1: (length(xspan)-1)
           input1 = xspan(i)+ 0.5*jump;
           input2 = output(i)+(0.5*jump*diffeq(xspan(i),output(i)));
           output(i+1) = output(i) + jump*diffeq(input1,input2);
    end
end
function [output] = RK4(diffeq, xspan)
    jump = xspan(2);
    halfjump = jump/2;
    output = xspan.*0;
    for i = 1: (length(xspan)-1)
           input1 = diffeq(xspan(i),output(i));
           input2 = diffeq(xspan(i)+ halfjump,output(i)+halfjump*input1);
           input3 = diffeq(xspan(i)+ halfjump,output(i)+halfjump*input2);
           input4 = diffeq(xspan(i)+ jump,output(i)+jump*input3);
           output(i+1) = output(i) + (1/6)*jump*(input1+input2+input3+input4);
    end
end
function [output] = grabsize(xspan)
    output = xspan.*1;
    jump = 1/length(xspan);
    for i = 1 : length(xspan)
        output(i) = jump*i;

    end
end
```

Published with MATLAB® R2021a