Assignment report 1 Jakub Suszwedyk

* What happens to the error if you halve the step size for both methods? Why?

For Adams–Bashforth method the error changes from 0.0733 to 0.0437 with halved step size

For Adams–Moulton method the error changes from 0.0376 to 0.0335 with halved step size

The error gets smaller with a smaller step size as for Adams–Bashforth method the local error is equal to O(h3), and for Adams–Moulton method its O(h4)where h is step size.

So the smaller is h the smaller is h3 and h4.

* How does the error of AB2 compare to the error of AM2?

The error of AM2 is smaller. As mentioned above, for Adams–Bashforth method the local error is equal to O(h3), and for Adams–Moulton method it’s O(h4)where h is step size. With values smaller than 1, the higher the power the smaller the number, so if h<1 then h4 < h3.

* What method did you choose to solve the implicit equation in AM2? Why?

I choose the secant method as It’s simple and efficient. I also had the best understanding of it and didn't see the advantage of using other methods.

* Would you prefer a fully-implicit AM2 or a predictor-corrector method using AB2 and AM2? Why? Give a detailed answer.

For that example I would prefer the predictor-corrector method using AB2 as here it would give us the same result ( I tried it and will attach the code for it below in the report so as not to mistake it with my actual answer ) and is significantly simpler to implement. I believe that for more complex problems the implicit method would be better but that’s not the case here.

Code output:

>> i6310933\_SuszwedykJakub\_assignment1

AM2\_error =

0.0375

AB\_error =

0.0733

>>

THE PREDICTOR-CORRECTOR METHOD IMPLEMENTATION OF AM2:

% Initialize Variables

dydt = @(t, y) sin(2\*t) + y - y^4;

h = 0.1;

t0 = 0;

tf = 1;

t = t0:h:tf;

y0 = 0;

y = zeros(1,length(t));

exact = 0.9426297327;

% First Step - Runge-Kutta 3rd order Method for both w1 and w2

%w1

k10 = h \* dydt(t0,y0);

k11 = h \* dydt(t0+0.5\*h,y0+0.5\*k10);

k12 = h \* dydt(t0+h,y0-k10+2\*k11);

y(2) = y0 + 1/6\*(k10+4\*k11+k12);

%w2

k20 = h \* dydt(h,y(2));

k21 = h \* dydt(h +0.5 \* h,y(2)+0.5\*k20);

k22 = h \* dydt(h +h,y(2)-k20+2\*k21);

y(3) = y(2) + 1/6\*(k20+4\*k21+k22);

% 2-stage Adams-Moulton method

for index = 3 : (tf/h+1)

tmp = y(index) + (h/12) \* (23\*dydt(index\*h,y(index)) - 16\*dydt((index-1)\*h, y(index-1)) + 5\*dydt((index-2)\*h, y(index-2)))

y(index+1) = y(index) + (h/12)\*( 5\*dydt((index+1)\*h,tmp) + 8\*dydt(index\*h,y(index)) - dydt((index-1)\*h,y(index-1)))

end

display('This is a Adams-Multon methods error:')

display(abs(y(tf/h+1)-exact))