Hw11

Bharddwaj Vemulapalli

(1) Write a function rk4(f, a, b, alpha, h) to implement Runge-Kutta of order 4 to solve an IVP. Show the output for Exercise 16d from Section 5.4 as a tabular comparison with the actual solution.

function []= rk4(f, a, b, alpha, h)

%Runge kutta method order 4

% output: approximation w to y at the (A + 1) values of t.

%h = (b-a)/n;

n = (b-a)/h;

K = zeros(4,1);

w = zeros(n+1,1);

t = zeros(n+1,1);

t(1) = a;

w(1) = alpha;

disp("(" + t(1) + ", " + w(1) + ")");

for i=1:n

K(1) = h\*f(t(i),w(i));

K(2) = h\*f(t(i) + h/2,w(i) + K(1)/2);

K(3) = h\*f(t(i) + h/2,w(i) + K(2)/2);

K(4) = h\*f(t(i) + h,w(i) + K(3));

w(i+1) = w(i) + (K(1) + 2\*K(2) + 2\*K(3) + K(4))/6;

t(i+1) = a + i\*h;

% disp("(" + t(i+1) + ", " + w(i+1) + ")");

fprintf("(");

fprintf('%.5f',t(i+1));

fprintf(", ");

fprintf('%.15f',w(i+1));

fprintf(")");

fprintf("\n");

end

end

(0, 1)

(0.10000, 1.014815867002839)

(0.20000, 1.057182198104166)

(0.30000, 1.121699889717769)

(0.40000, 1.201488103624477)

(0.50000, 1.289807148756180)

(0.60000, 1.380932546425368)

(0.70000, 1.470415759334389)

(0.80000, 1.555031097138222)

(0.90000, 1.632611866406378)

(1.00000, 1.701867708542123)

Actual:

(0, 1)

(0.1, 1.014815499858224)

(0.2, 1.057181007464205)

(0.3, 1.121698018267981)

(0.4, 1.201486010364401)

(0.5, 1.289805276305608)

(0.6, 1.380931215443733)

(0.7, 1.470415185397224)

(0.8, 1.555031423435184)

(0.9, 1.632613181782867)

(1, 1.701870052761277)

(2) Implement Runge-Kutta-Fehlberg Methods (Algorithm 5.3) on page 296-297.  Use this implementation to show the output of 3d as a tabular comparison with the actual solution.

function [t,w,h] = rkfm(f,a,b,alpha,TOL,hmax,hmin)

%Runge-Kutta-Fehlberg Methods (Algorithm 5.3)

%Output : t,w,h,where w approximates y(t) and the step size h was used or a message that the minimum step size was exceeded.

K = zeros(6,1);

t = a;

w = alpha;

h = hmax;

FLAG = 1;

disp("(" + t + ", " + w + ")");

format long;

while FLAG == 1

K(1) = h\*f(t,w);

K(2) = h\*f(t + (1/4)\*h, w + K(1)\*(1/4));

K(3) = h\*f(t + (3/8)\*h , w + (3/32)\*K(1)+ (9/32)\*K(2));

K(4) = h\*f(t + (12/13)\*h, w + (1932/2197)\*K(1) - (7200/2197)\*K(2) +(7296/2197)\*K(3));

K(5) = h\*f(t + h, w + (439/216)\*K(1) - 8\*K(2) + (3680/513)\*K(3) - (845/4104)\*K(4));

K(6) = h\*f(t + .5\*h, w - (8/27)\*K(1) + 2\*K(2) - (3544/2565)\*K(3) + (1859/4104)\*K(4) - (11/40)\*K(5));

%disp(K);

R = (1/h) \* abs( (1/360)\*K(1) - (128/4275)\*K(3) - (2197/75240)\*K(4) + (1/50)\*K(5) + (2/55)\*K(6) );

if R <= TOL

t = t + h;

w = w + (25/216)\*K(1) + (1408/2565)\*K(3) + (2197/4104)\*K(4) - (1/5)\*K(5);

% disp("(" + t + ", " + w + ", " + h + ")");

fprintf("(");

fprintf('%.15f',t);

fprintf(", ");

fprintf('%.15f',w);

fprintf(", ");

fprintf('%.15f',h);

fprintf(")");

fprintf("\n");

end

gamma = 0.84\*(TOL/R)^.25;

if gamma <= .1

h = 0.1\*h;

elseif gamma >= 4

h = 4\*h;

else

h = gamma\*h;

end

if h > hmax

h = hmax;

end

if t >= b

FLAG = 0;

elseif t+h > b

h = b-t;

elseif h < hmin

FLAG = 0;

disp("Minimum h exceeded.")

end

end

end

(0, 0.33333)

%Of the form (t,w,h)

(0.398605138699089, 0.310820109039545, 0.398605138699089)

(0.683725949501497, 0.272057207565061, 0.285120810802409)

(0.970396983879385, 0.222118944797817, 0.286671034377887)

(1.263081852154853, 0.167197904562855, 0.292684868275468)

(1.567290542637737, 0.113308496371436, 0.304208690482885)

(1.733307135729408, 0.087689897033825, 0.166016593091671)

(1.909769804813315, 0.064482283034826, 0.176462669083907)

(2.000000000000000, 0.054345435085901, 0.090230195186685)

Actual

(0, 0.33333)

(0.398605138699089, 0.310819945642301)

(0.683725949501497, 0.272056954652640)

(0.970396983879385, 0.222118556645298)

(1.263081852154853, 0.167197339968042)

(1.567290542637737, 0.113308155854171)

(1.733307135729408, 0.087689701506332)

(1.909769804813315, 0.064482355518636)

(2.000000000000000, 0.054345506612664)

(3) Write a function adam\_bashforth5(f, a, b, alpha, h) that implements Adam-Bashforth methods of order 5. Show the output of problem 2(a) in section 5.6  as a tabular comparison with the actual solution.

function [] = adam\_bashforth5(f, a, b, alpha, h)

%adam bashforth five step explicit method

n = (b-a)/h;

K = zeros(4,1);

t = zeros(n+1,1);

w = zeros(n+1,1);

t(1) = a;

w(1) = alpha;

disp("(" + t(1) + ", " + w(1) + ")");

for i=2:5

K(1) = h\*f(t(i-1),w(i-1));

K(2) = h\*f(t(i-1) + h/2,w(i-1) + K(1)/2);

K(3) = h\*f(t(i-1) + h/2,w(i-1) + K(2)/2);

K(4) = h\*f(t(i-1) + h,w(i-1) + K(3));

w(i) = w(i-1) + (K(1) + 2\*K(2) + 2\*K(3) + K(4))/6;

t(i) = a + (i-1)\*h;

% disp("(" + t(i) + ", " + w(i) + ")");

fprintf("(");

fprintf('%.5f',t(i));

fprintf(", ");

fprintf('%.15f',w(i));

fprintf(")");

fprintf("\n");

end

% disp(n);

for i = 6:n+1 % i = 5:n

%w(i+1) = w(i) + (h/720)\*( 1901\*f(t(i),w(i)) - 2774\*f(t(i-1), w(i-1)) + 2616\*f(t(i-2), w(i-2)) -1274\*f(t(i-3),w(i-3)) + 251\*f(t(i-4), w(i-4)) );

w(i) = w(i-1) + h\*( 1901\*f(t(i-1),w(i-1)) - 2774\*f(t(i-2), w(i-2)) + 2616\*f(t(i-3), w(i-3)) -1274\*f(t(i-4),w(i-4)) + 251\*f(t(i-5), w(i-5)) )/720;

% disp("(" + t(i) + ", " + w(i) + ")");

t(i) = a + (i-1)\*h;

fprintf("(");

fprintf('%.5f',t(i));

fprintf(", ");

fprintf('%.15f',w(i));

fprintf(")");

fprintf("\n");

end

end

(1, 0)

(1.10000, 0.105159959170369)

(1.20000, 0.221242988451047)

(1.30000, 0.349121365736641)

(1.40000, 0.489681868508386)

(1.50000, 0.643882719831293)

Actual:

(1, 0)

(1.1, 0.105159815777454)

(1.2, 0.221242772757631)

(1.3, 0.349121132275639)

(1.4, 0.489681663750943)

(1.5, 0.643875331894374)

4) Implement the predictor-corrector method of order 5 that we discussed in class. You can modify Algorithm 5.4 to use the constants for the fifth-order method.  The function from the problem (3) could be called here but be mindful that doing so increases the number of function evaluations unnecessarily. Show the output of problem 3(d) in 5.6 as a tabular comparison with the actual solution.

function [] = predictorCorrectorOrder5(f, a, b, alpha, h)

%Adams 5th Order Predictor-Corrector

format long;

n = (b-a)/h;

K = zeros(4,1);

t = zeros(n+2,1);

w = zeros(n+2,1);

t(1) = a;

w(1) = alpha;

disp("(" + t(1) + ", " + w(1) + ")");

for i=2:5

K(1) = h\*f(t(i-1),w(i-1));

K(2) = h\*f(t(i-1) + h/2,w(i-1) + K(1)/2);

K(3) = h\*f(t(i-1) + h/2,w(i-1) + K(2)/2);

K(4) = h\*f(t(i-1) + h,w(i-1) + K(3));

w(i) = w(i-1) + (K(1) + 2\*K(2) + 2\*K(3) + K(4))/6;

t(i) = a + (i-1)\*h;

% disp("(" + t(i) + ", " + w(i) + ")");

fprintf("(");

fprintf('%.5f',t(i));

fprintf(", ");

fprintf('%.15f',w(i));

fprintf(")");

fprintf("\n");

end

for i = 6:n+1

t(i) = a + (i-1)\*h; %because matlab starts at 1 as opposed to 0 we are 1 index off

w(i) = w(i-1) + h\*( 1901\*f(t(i-1),w(i-1)) - 2774\*f(t(i-2), w(i-2)) + 2616\*f(t(i-3), w(i-3)) -1274\*f(t(i-4),w(i-4)) + 251\*f(t(i-5), w(i-5)) )/720; %prediction wi

w(i) = w(i-1) + h\*( 251\*f(t(i),w(i)) + 646\*f(t(i-1),w(i-1)) - 264\*f(t(i-2),w(i-2)) + 106\*f(t(i-3),w(i-3)) - 19\*f(t(i-4),w(i-4)) )/720;

% disp("(" + t(i) + ", " + w(i) + ")");

fprintf("(");

fprintf('%.5f',t(i));

fprintf(", ");

fprintf('%.15f',w(i));

fprintf(")");

fprintf("\n");

end

end

(0, 0.33333)

(0.10000, 0.212282986111111)

(0.20000, 0.162765457718461)

(0.30000, 0.164516540751045)

(0.40000, 0.205240505195296)

(0.50000, 0.277549231725274)

(0.60000, 0.376775425190256)

(0.70000, 0.500203293418676)

(0.80000, 0.646217947142655)

(0.90000, 0.813786776976697)

(1.00000, 1.002302621480558)

Actual:

(0, 0.33333)

(0.1, 0.212176886570878)

(0.2, 0.162626480390481)

(0.3, 0.164376720049477)

(0.4, 0.205111761078871)

(0.5, 0.277361666207966)

(0.6, 0.376595689455955)

(0.7, 0.500065794474106)

(0.8, 0.646105212962912)

(0.9, 0.813702998846081)

(1, 1.002245982333029)