```
clear;
clc;
syms k x y;
f = -k*sqrt(y)/log(x+1);
f1 = subs(f,k,0.25);
f2 = subs(f,k,0.5);
method1 = @euler;
method2 = @predictor;

% useing euler
disp('Using euler method');
N_m1_f1 = grid(method1,f1,1,5,4,0.001);
N_m1_f2 = grid(method1,f2,1,5,4,0.001);
y1_euler = euler(f1,1,5,4,N_m1_f1)';
y2_euler = euler(f2,1,5,4,N_m1_f2)';
%%
fprintf(' euler %s        %s\n',char(f1), char(f2));
% fprintf('%8.8f    %8.8f',y1_euler, y2_euler);

sprintf('%30.8f \n',y1_euler)
fprintf(' euler %s        %s\n',char(f1), char(f2));
sprintf('%60.8f\n',y2_euler)
%%

% useing predictor
N_m2_f1 = grid(method2,f1,1,5,4,0.001);
N_m2_f2 = grid(method2,f2,1,5,4,0.001);

y1_pred = euler(f1,1,5,4,N_m2_f1)';

y2_pred = euler(f2,1,5,4,N_m2_f2)';
%%
disp('Using predictor correct method');
fprintf(' euler %s        %s\n',char(f1), char(f2));
sprintf('%30.8f \n',y1_pred)
fprintf(' euler %s        %s\n',char(f1), char(f2));
sprintf('%60.8f\n',y2_pred)
```

```
function y = euler(f,xa,xb,ya,N)
step = (xb - xa)/N;
x = linspace(xa,xb,N+1);
y = linspace(0,0,N+1);
y(1) = ya;
for i = 1 : N
    y(i+1) = y(i) + eval(subs(subs(f,'x',x(i)),'y',y(i))) * step;
    %subs(cos(a) + sin(b), [a, b], [sym('alpha'), 2])
end

end
```

```
function y = predictor(f,xa,xb,ya,N)
step = (xb - xa)/N;
x = linspace(xa,xb,N+1);
y = linspace(0,0,N+1);
y(1) = ya;
for i = 1 : N
    yp = y(i) + eval(subs(subs(f,'x',x(i)),'y',y(i)))*step;
    left = eval(subs(subs(f,'x',x(i)),'y',y(i)));
    right = eval(subs(subs(f,'x',x(i+1)),'y',yp));
    y(i+1) = y(i) + 1/2*(left+right)*step;
    %subs(cos(a) + sin(b), [a, b], [sym('alpha'), 2])
end

end
```

```
function N = grid(method,f,xa,xb,ya, delta)
N = 2;
while N < 100
    [yn2c, yn2b] = findy(method,f,xa,xb,ya,N+2);
    [yn1c, yn1b] = findy(method,f,xa,xb,ya,N);
    ec = abs(2*(yn2c-yn1c)/(yn2c+yn1c));
    eb = abs(2*(yn2b-yn1b)/(yn2b+yn1b));
    if ec < delta && eb < delta
        break;
    end
    N = N + 2;
end
end
```

```
function [yc, yb, y] = findy(method,f,xa,xb,ya,N)
% N must be even number
    y = feval(method,f,xa,xb,ya,N)'; %  find the y array
    yc = y(N/2+1); %  find the center point of y
    yb = y(N+1); %  find the last point of y
end
```

```
Using euler method
 euler -y^(1/2)/(4*log(x + 1))        -y^(1/2)/(2*log(x + 1))

ans =

                    4.00000000
                    3.88902346
                    3.79016653
                    3.70062579
                    3.61848812
                    3.54238875
                    3.47131912
                    3.40451223
                    3.34137075
                    3.28142001
                    3.22427632
                    3.16962489
                    3.11720408
                    3.06679394
                    3.01820767
                    2.97128523
                    2.92588832
                    2.88189659
                    2.83920463
                    2.79771949
                    2.75735881
                    2.71804921
                    2.67972500
                    2.64232713
                    2.60580232
                    2.57010226
                    2.53518306


 euler -y^(1/2)/(4*log(x + 1))        -y^(1/2)/(2*log(x + 1))

ans =

                                                    4.00000000
                                                    3.87977541
                                                    3.76795677
                                                    3.66335102
                                                    3.56500777
                                                    3.47215756
                                                    3.38416858
                                                    3.30051559
                                                    3.22075717
                                                    3.14451867
                                                    3.07147936
                                                    3.00136249
                                                    2.93392753
```

```
                                                    2.86896410
                                                    2.80628704
                                                    2.74573257
                                                    2.68715502
                                                    2.63042428
                                                    2.57542361
                                                    2.52204783
                                                    2.47020188
                                                    2.41979948
                                                    2.37076210
                                                    2.32301806
                                                    2.27650170
                                                    2.23115274
                                                    2.18691571
                                                    2.14373941
                                                    2.10157651
                                                    2.06038312
                                                    2.02011849
                                                    1.98074472
                                                    1.94222646
                                                    1.90453070
                                                    1.86762657
                                                    1.83148514
                                                    1.79607927
                                                    1.76138344
                                                    1.72737365
                                                    1.69402728
                                                    1.66132299
                                                    1.62924062
                                                    1.59776110
                                                    1.56686636
                                                    1.53653930
                                                    1.50676367
                                                    1.47752404
                                                    1.44880574
                                                    1.42059480


Using predictor correct method
 euler -y^(1/2)/(4*log(x + 1))        -y^(1/2)/(2*log(x + 1))

ans =

                    4.00000000
                    3.71146099
                    3.49140570
                    3.30992793
                    3.15351479
                    3.01488048
                    2.88962994
                    2.77489682
```

```
                    2.66870115
                    2.56961357
                    2.47656548


 euler -y^(1/2)/(4*log(x + 1))        -y^(1/2)/(2*log(x + 1))

ans =

                                                    4.00000000
                                                    3.51910165
                                                    3.15009980
                                                    2.84850941
                                                    2.59246628
                                                    2.36957729
                                                    2.17211669
                                                    1.99492846
                                                    1.83439019
                                                    1.68785281
                                                    1.55331579
                                                    1.42922793
                                                    1.31435971


>>
```