Exercise #1

Part A.

f=inline('2\*y','t','y');

t=linspace(0,.5,100); y=3\*exp(2\*t);

[t5,y5]=euler(f,[0,.5],3, 5); %solves the ODE using Euler with 5 steps

approx. = y5(end)

exact=y(end)

e5=abs(approx-exact)

approx =

7.4650

exact =

8.1548

e5 =

0.6899

[t50,y50]=euler(f,[0,.5],3, 50); % solves the ODE using Euler with 50 steps

approx = y50(end)

exact=y(end)

e50=abs(approx-exact)

ratio=e5/e50

approx =

8.0748

exact =

8.1548

[t500,y500]=euler(f,[0,.5],3, 500); % solves the ODE using Euler with 500 steps

approx = y500(end)

exact=y(end)

e500=abs(approx-exact)

ratio=e50/e500

approx =

8.1467

exact =

8.1548

e500 =

0.0081

ratio =

9.8381

e50 =

0.0801

ratio =

8.6148

|  |  |  |  |
| --- | --- | --- | --- |
| N | approximation | error | ratio |
| 5 | 7.4650 | 0.6899 | n/a |
| 50 | 8.0748 | 0.0801 | 8.6148 |
| 500 | 8.1467 | 0.0081 | 9.8889 |
| 5000 | 8.1540 | 8.1534e-004 | 9.9835 |

Part B.

As the number of steps increase we get a closer approximation to the real number and the error decreases. Therefore, the error is decreased by the ratio which is roughly 10.

Part C.

Because Euler’s method follows the tangent line, which will be concave down always, therefore Euler’s method underestimates the actual.

%Exercise 2

Part A

t = 0:.45:10; y = -30:6:42;

[T,Y]=meshgrid(t,y); % creates 2d matrices of points in the ty-plane

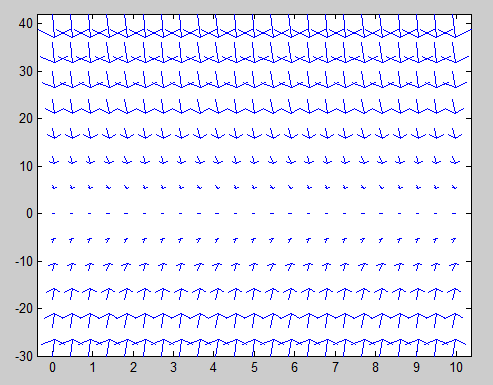
dT = ones(size(T)); % dt=1 for all points

dY = -2\*Y; % dy = -2\*y; this is the ODE

quiver(T,Y,dT,dY) % draw arrows (t,y)->(t+dt, t+dy)

axistight% adjust look

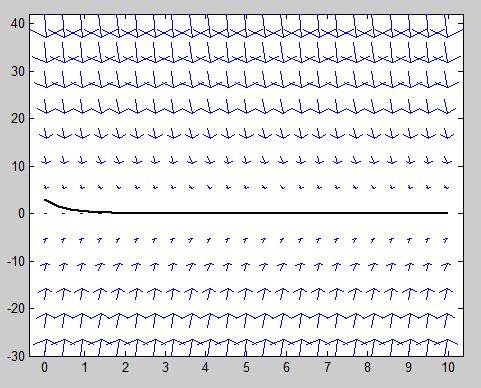
holdon



Part B.

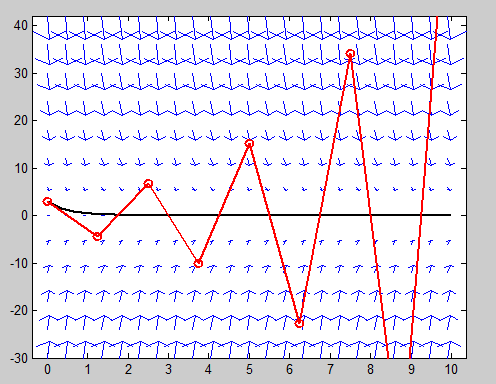
f=inline('-2\*y','t','y');  
t=linspace(0,10,200);y=3\*exp(-2\*t);

plot(t,y,'k-','linewidth',2)



Part C

[t8,y8]=euler(f,[0,10],3,8);  
plot(t8,y8,'ro-','linewidth',2)



The approximations are inaccurate because the number of steps that is utilized is low, which makes our errors larger.

Part D

t = 0:.4:10; y = -1:0.4:3;

[T,Y]=meshgrid(t,y);

dT = ones(size(T));

dY = -2\*Y;

quiver(T,Y,dT,dY)

axis tight

hold on

f=inline('-2\*y','t','y');

t=linspace(0,10,200);y=3\*exp(-2\*t);

plot(t,y,'k-','linewidth',2)

[t16,y16]=euler(f,[0,10],3,16);

plot(t16,y16,'ro-','linewidth',2)



Euler’s method is closer to the actual solution than it was in the previous part c because we used a larger number of steps, which makes the solutions accurate.

% Exercise 3

Function name: impeuler.m

function [tout,yout] = impeuler(f,tspan,y0,N )

h = (tspan(2)-tspan(1))/N;

t = tspan(1); tout = t;

y = y0(:); yout = y.';

for n = 1:N

f1=f(t,y);

f2=f(t+h, y+h\*f1);

y = y+h\*(f1+f2)/2; t = t+h

yout = [yout; y.']; tout = [tout; t];

end

end

[t5,y5] = improvedeuler(f,[0,.5],3,5);

[t5,y5]

ans =

0 3.0000

0.1000 3.6600

0.2000 4.4652

0.3000 5.4475

0.4000 6.6460

0.5000 8.1081

% Exercise 4

disp('Computing Solution for N=5')

[t5,y5]=impeuler(f,[0,.5],3, 5); % solves the ODE using Euler with 5 steps

approx = y5(end)

exact = y(end)

e5 = abs(approx - exact)

disp('Computing Solution for N=50')

[t50,y50]=impeuler(f,[0,.5],3, 50); % solves the ODE using Euler with 50 steps

approx = y50(end)

exact = y(end)

e50 = abs(approx - exact)

ratio=e5/e50

disp('Computing Solution for N=500')

[t500,y500]=impeuler(f,[0,.5],3, 500); % solves the ODE using Euler with 500 steps

approx = y500(end)

exact = y(end)

e500 = abs(approx - exact)

ratio=e50/e500

disp('Computing Solution for N=5000')

[t5000,y5000]=impeuler(f,[0,.5],3, 500); % solves the ODE using Euler with 5000 steps

approx = y5000(end)

exact = y(end)

e5000 = abs(approx - exact)

ratio=e500/e5000

|  |  |  |  |
| --- | --- | --- | --- |
| N | approximation | error | ratio |
| 5 | 8.1081 | 0.0467 | n/a |
| 50 | 8.1543 | 5.3555e-004 | 87.2394 |
| 500 | 8.1548 | 5.4284e-006 | 98.6567 |
| 5000 | 8.1548 | 5.4357e-008 | 99.8650 |

Part B

Because we are decreasing the stepsize by a factor of 10, our error is being decreased by a factor of approximately 102, or 100. Therefore, it makes sense that the ratios are very close to 100, since that is the factor that the errors have been decreasing

% Exercise 5

t = 0:.45:10; y = -30:6:42 ;

[T,Y]=meshgrid(t,y);

dT = ones(size(T));

dY = -2\*Y;

quiver(T,Y,dT,dY)

axis tight

hold on

f=inline('-2\*y','t','y');

t=linspace(0,10,200);y=3\*exp(-2\*t);

plot(t,y,'k-','linewidth',2)

[t8,y8]=improvedeuler(f,[0,10],3,8);

plot(t8,y8,'ro-','linewidth',2)



t = 0:.4:10; y = -1:0.4:3;

[T,Y]=meshgrid(t,y);

dT = ones(size(T));

dY = -2\*Y;

quiver(T,Y,dT,dY)

axis tight

hold on

f=inline('-2\*y','t','y');

t=linspace(0,10,200);y=3\*exp(-2\*t);

plot(t,y,'k-','linewidth',2)

[t16,y16]=improvedeuler(f,[0,10],3,16);

plot(t16,y16,'ro-','linewidth',2)



The reason why is you need a certain amount of steps, or else the approximations will be very off.