Exercise #1

(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
>> approx5 = y5(end)

approx5 =

    7.4650

>> exact5 = y(end)

exact5 =

    8.1548

>> e5=abs(approx5-exact5)

e5 =

    0.6899

>> [t50,y50]=euler(f,[0,.5],3, 50); % solves the ODE using Euler with 50 steps
>> approx50 = y50(end)

approx50 =

    8.0748

>> exact50 = y(end)

exact50 =

    8.1548

>> e50=abs(approx50-exact50)

e50 =

    0.0801
```

```
>> ratio1 = e5/e50

ratio1 =

   8.6148

>> [t500,y500]=euler(f,[0,.5],3, 500); % solves the ODE using Euler with 500 steps
>> approx500 = y500(end)

approx500 =

   8.1467

>> exact500 = y(end)

exact500 =

   8.1548

>> e500=abs(approx500-exact500)

e500 =

   0.0081

>> ratio2=e50/e500

ratio2 =

   9.8381

>> [t5000,y5000]=euler(f,[0,.5],3, 5000); % solves the ODE using Euler with 5000 steps
>> approx5000 = y5000(end)

approx5000 =

   8.1540

>> exact5000 = y(end)

exact5000 =

   8.1548
```

>> e5000=abs(approx5000-exact5000)

e5000 =

 8.1534e-004

>> ratio3=e500/e5000

ratio3 =

 9.9835

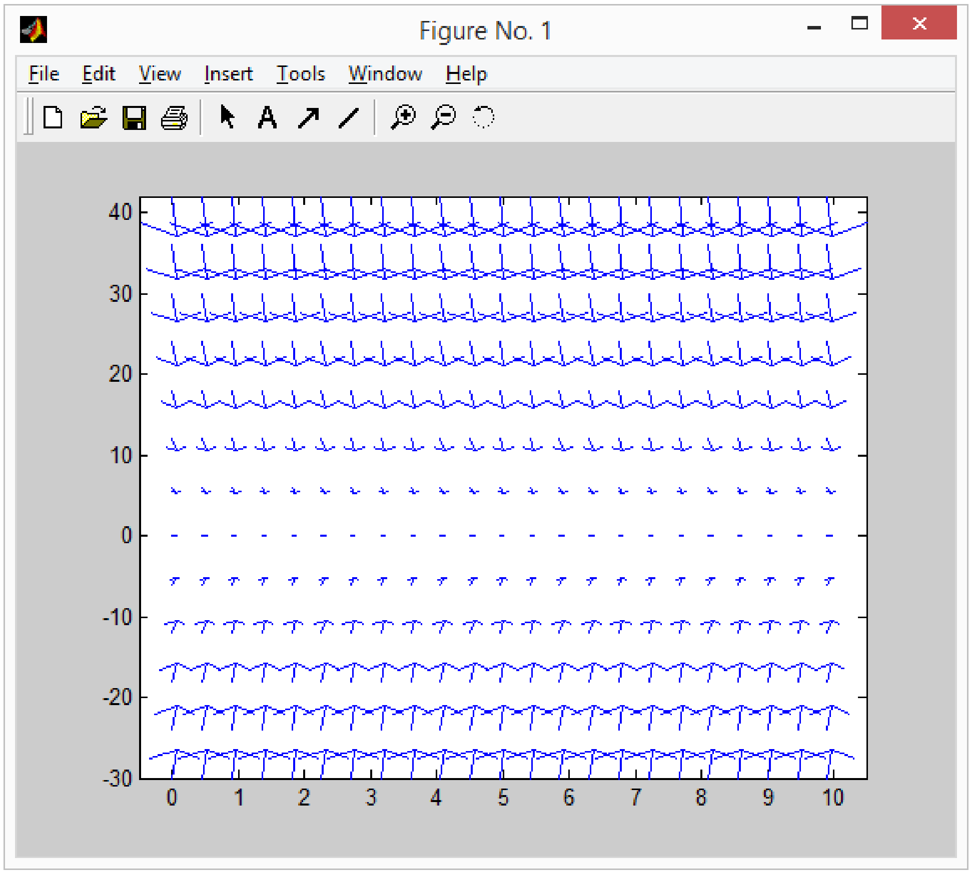
N	approximation	error	ratio
5	7.4650	0.6899	None
50	8.0748	0.0801	8.6148
500	8.1467	0.0081	9.8381
5000	8.1540	8.1534e-004	9.9835

(B).
As the number of steps increased by 10, the approximation got closer to the real number and the error decreased by the ratio of roughly 10.

(C).
The reason is that Euler's method follows the tangent line, which will always be concave down, therefore Euler's method always underestimates actual value.

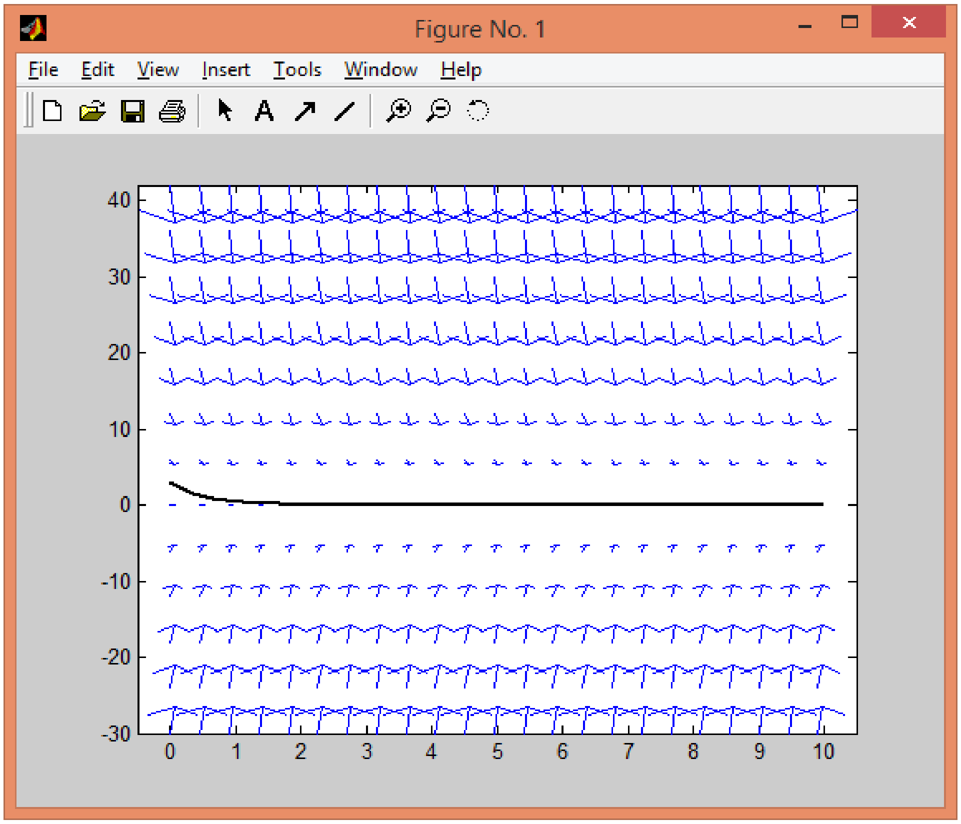
Exercise 2

(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)
axis tight % adjust look
hold on
```



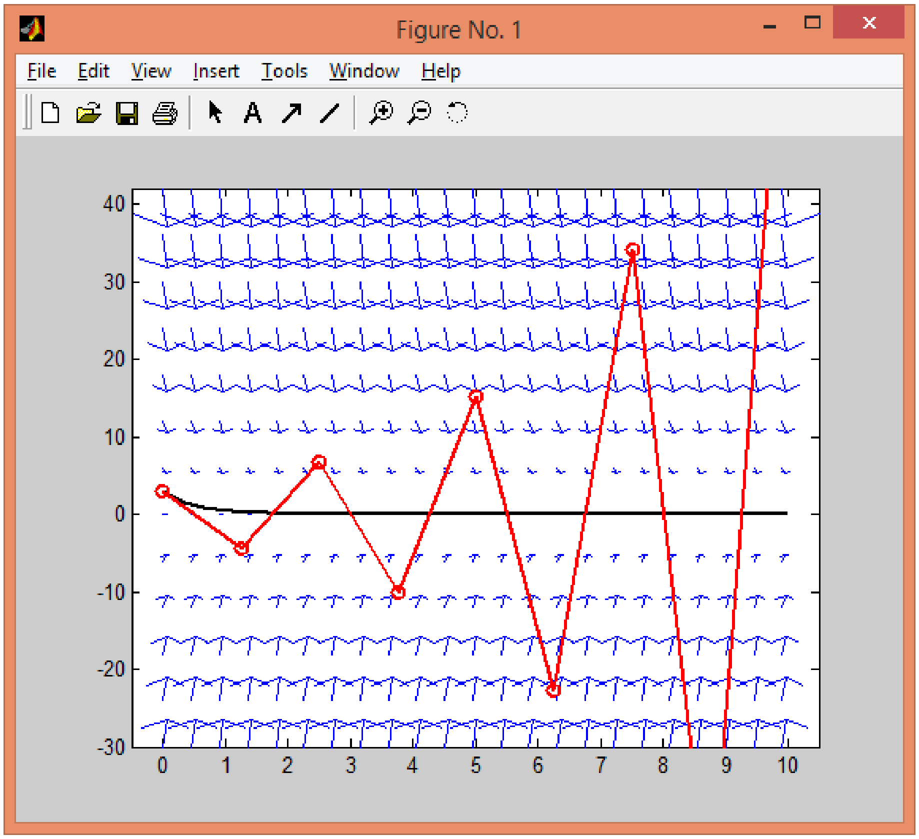
(B).

```
f=inline('-2*y','t','y');
t=linspace(0,10,200);y=3*exp(-2*t);
plot(t,y,'k-','linewidth',2)
```



Part C

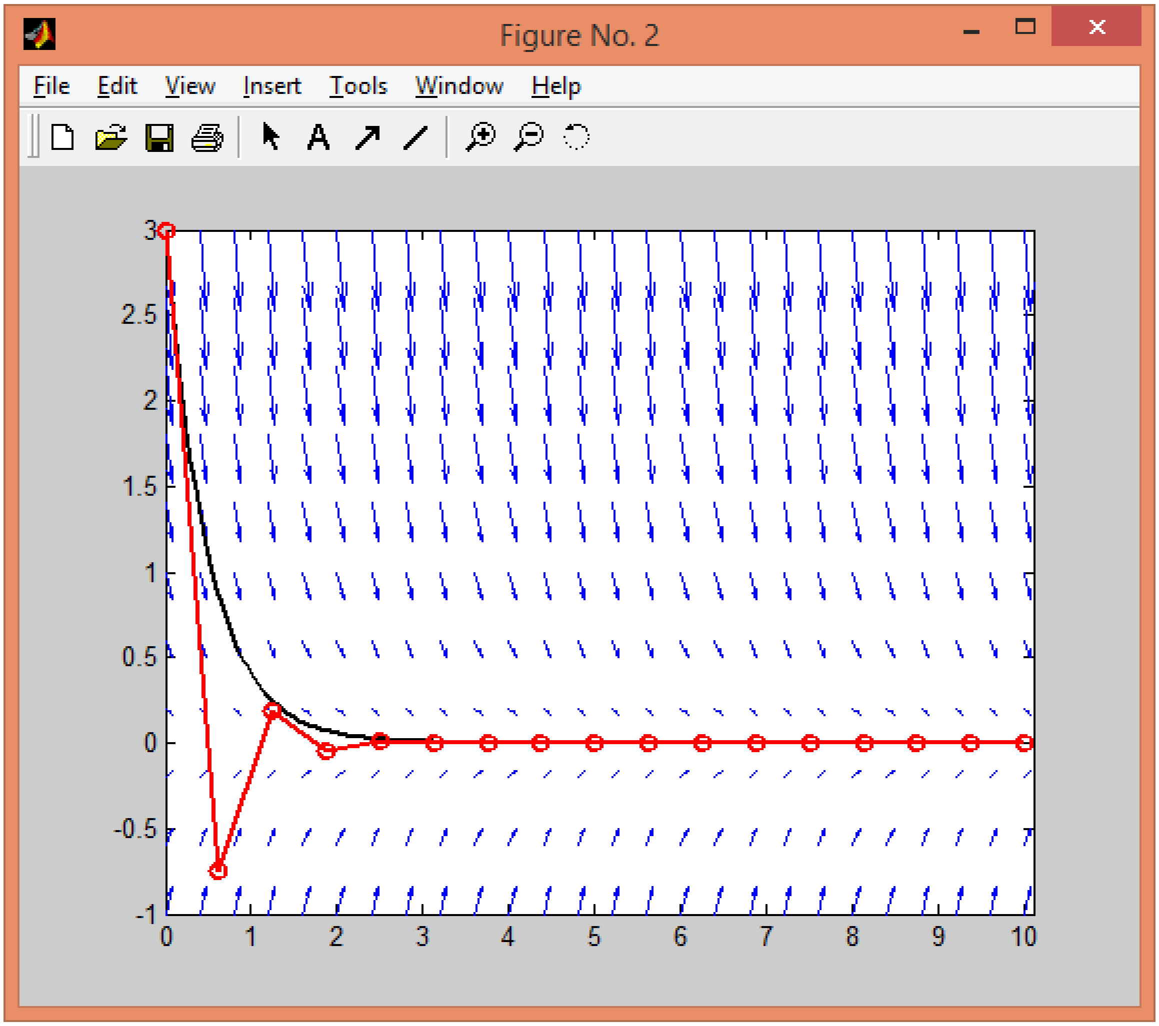
```
[t8,y8]=euler(f,[0,10],3,8);
plot(t8,y8,'ro-','linewidth',2)
```



The reason is that Euler method becomes closer to actual value with larger number of step size, as the error is inversely proportional to the number of step size.

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)
```



Observation: The Euler's method gets closer to the actual solution than it was in part c because of a larger number of steps, which makes the solutions more accurate.

Exercise 3

```
Function name: impeuler.m
function [tmod,ymod] = impeuler(f,tspan,y0,N )
h = (tspan(2)-tspan(1))/N;
t = tspan(1); tmod = t;
y = y0(:); ymod = 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
    ymod = [ymod; y.']; tmod = [tmod; t];
end

[t5,y5] = impeuler(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('Eulers method with N = 5')
Eulers method with N = 5
>> [t5,y5]=impeuler(f,[0,.5],3, 5); % solves the ODE using Euler with 5 steps
>> approx5 = y5(end)

approx5 =

    8.1081

>> exact5 = y(end)

exact5 =

    8.1548

>> e5=abs(approx5-exact5)

e5 =

    0.0467

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

approx50 = y50(end)

approx50 =

    8.1543

>> exact50 = y(end)

exact50 =

    8.1548

>> e50 = abs(approx50 - exact50)
```

```
e50 =

  5.3555e-004

>> ratio1=e5/e50

ratio1 =

   87.2394

>> disp('Eulers method with N = 500')
Eulers method with N = 500
>> [t500,y500]=impeuler(f,[0,.5],3, 500); % solves the ODE using Euler with
500 steps
>> approx500 = y500(end)

approx500 =

    8.1548

>> exact500 = y(end)

exact500 =

    8.1548

>> e500 = abs(approx500 - exact500)

e500 =

  5.4284e-006

>> ratio2 = e50/e500

ratio2 =

   98.6567

>> disp('Eulers method with N = 5000')
Eulers method with N = 5000
>> [t5000,y5000]=impeuler(f,[0,.5],3, 5000);
approx5000 = y5000(end)

approx5000 =

    8.1548

>>
>> exact5000 = y(end)

exact5000 =

    8.1548

>> e5000 = abs(approx5000 - exact5000)
```

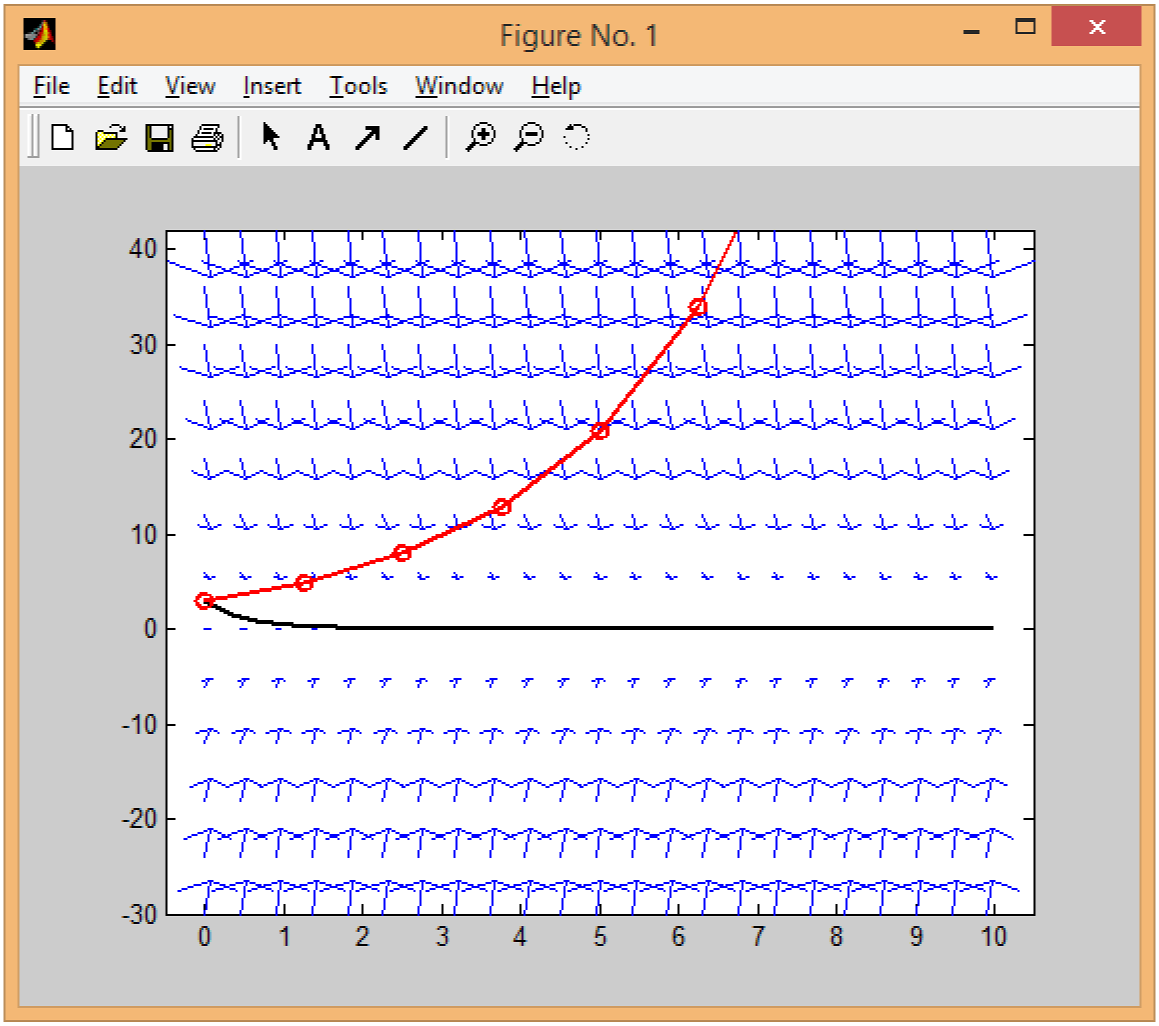
```
e5000 =

  5.4357e-008

>> ratio3 = e500/e5000

ratio3 =

   99.8650
```

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

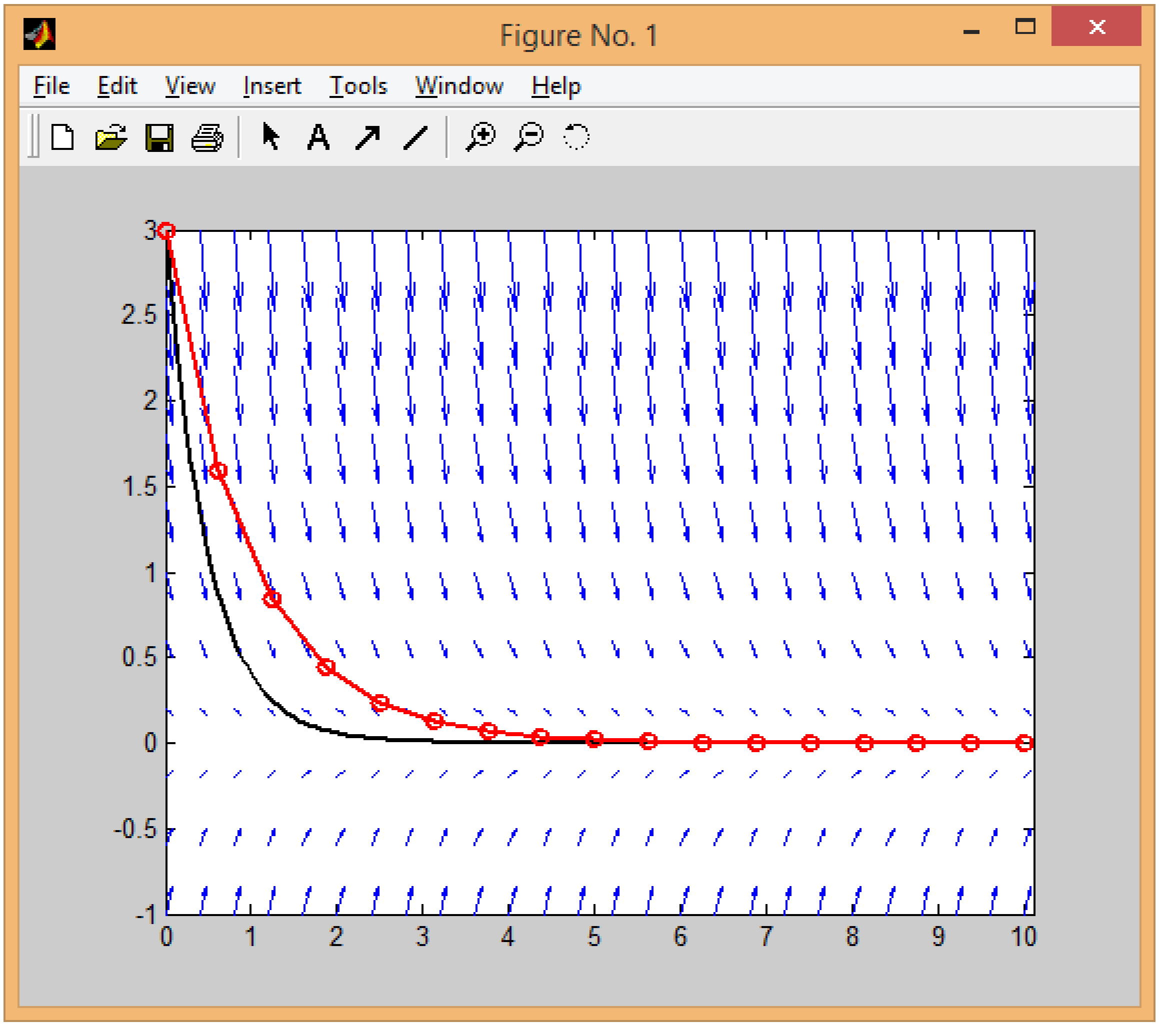
Part B
The reason is that the step size is increased by a factor of 10, coupling with the improved Euler's method which is of order h^2. Therefore, the error is decreased by a factor of approximately 10^2, or 100.

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]=impeuler(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]=impeuler(f,[0,10],3,16);
plot(t16,y16,'ro-','linewidth',2)
```




```
The reason is that in Euler's method, the more step size is the closer the
approximation.
```