$$t > 1535.06 \; minutes \tag{29}$$

$$t > 25.58 \; hours \tag{30}$$

This implies that the time taken for concentration to fall below safe level is t must greater than 25.58 hours in this example.

Hence, the time taken for concentration to fall below safe level is $t > \frac{2000}{2000k-9}\ln(\frac{2\times10^6}{M_o})$.

4.2 Result for Numerical Solution

The analytical and numerical solutions are plotted on the same graph (Figure 4.3) to compare whether there is an error.

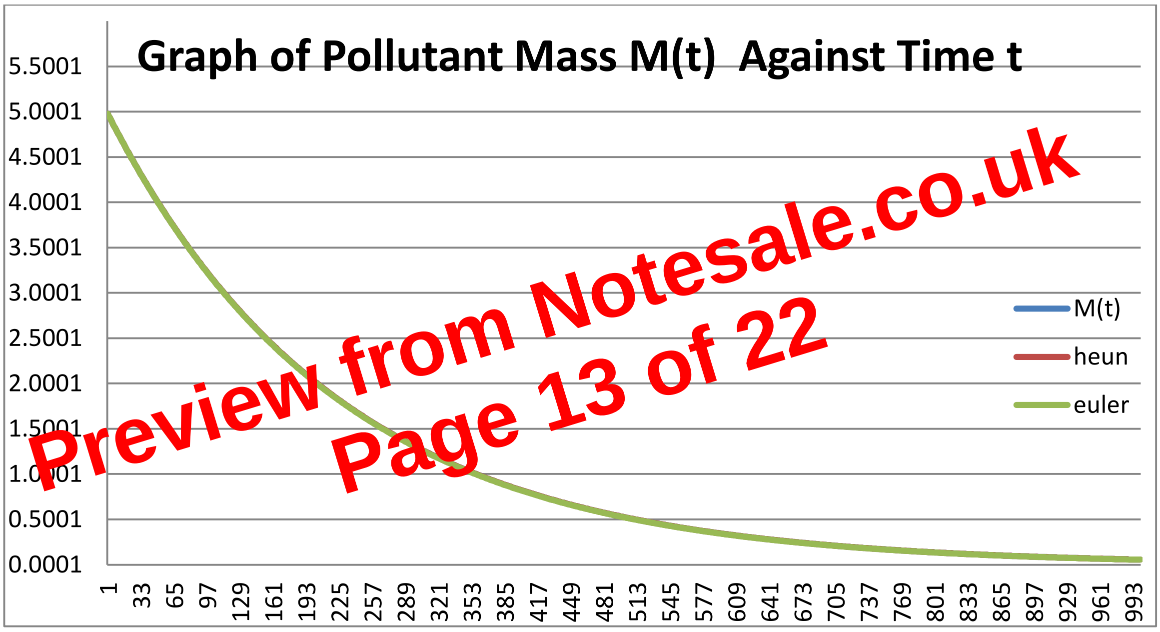

Figure 4.3 Graph of pollutant mass, *M(t)* against time, *t* for analytical and numerical solution.

From the graph, both the analytical and numerical solutions are almost the same, with a very small error. The range of error is $0 < error < 0.0045$.

Now, Euler and Heun method are compared. A graph of errors against time is plotted for both methods as shown in Figure 4.4.

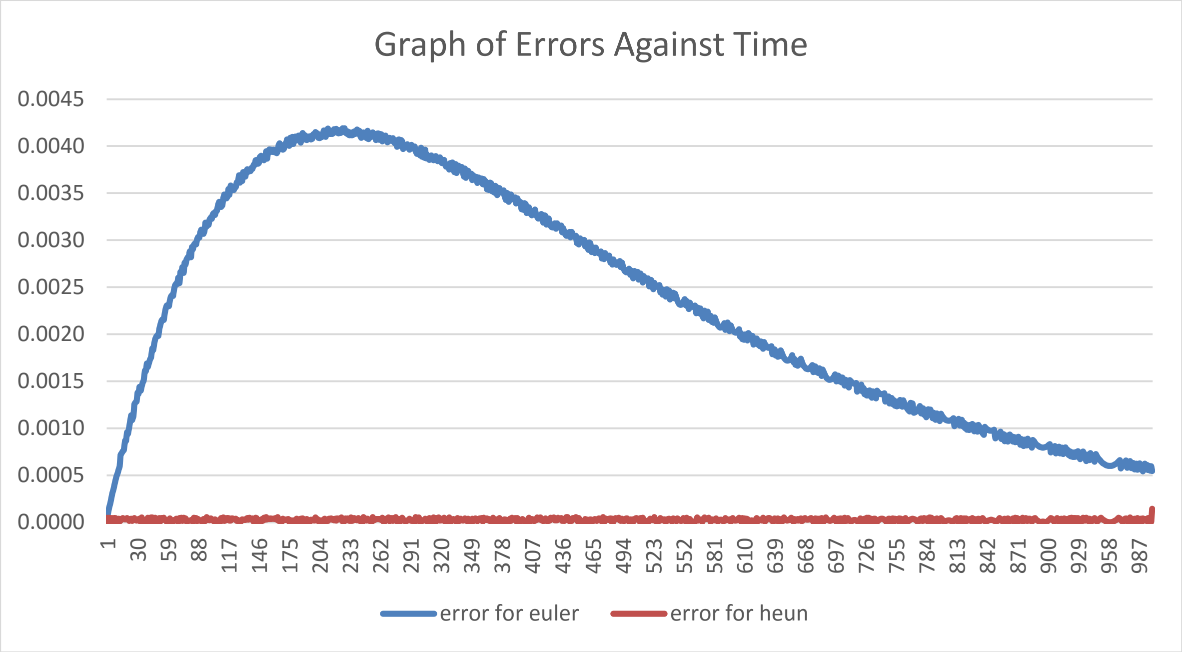

Figure 4.4 Graph of errors against time, *t* for Euler and Heun method.

From the graph, it is found that the error for Heun method is smaller than Euler method. Therefore, Heun method is a better approximation for the problem.

```
t=t0:dt:tf;

%setting initial y value
y(1)=y0;

%loop using euler's method
for i=1:length(t)-1
    y(i+1)=y(i)+dt*(feval(func,t(i),y(i)));
end

%print column vectors of t and y
t=t'
y=y'

plot(t,y)
xlabel('Time')
ylabel('y')
```

Code 1.3(for *Matlab*): Function (funct.m)

```
function z=funct(t,y)
z=(0.004-9/2000)*y;

%your f(t,y) function
```


Code 1.4(for *Matlab*): Ordinary Heun method (odeheun.m)

```
function [t,y]=odeheun(f,a,b,alpha,h)
%input
%f(x,y) right hand side of ODE
%a      initial value of x
%b      final value of x
%alpha  initial condition y(a)=alpha
%h      step size
%output
%t     vector of grid points
%y     vector of approximation value
N=length(a:h:b) %number of grid points
t=zeros(1,N);   %zero vector of grid points
y=zeros(1,N);   %zero vector of approximation values
t(1)=a; t(N)=b;
y(1)=alpha;
disp('      t(i)      y(i)       k1         k2          ')
disp(' -------------------------------------------------------- ')
for i=1:N-1
    t(i)=a+(i-1)*h;
    k1=h*f(t(i),y(i));
    k2=h*f(t(i)+h,y(i)+k1);
    disp([t(i),y(i),k1,k2]);
    y(i+1)=y(i)+(k1+k2)/2;
end;


%print column vectors of t and y
t=t'
y=y'
```