Report 3

Luo YiWen

April 9, 2024

0.1 MATLAB Comparison of Methods

Consider the differential equation and initial condition

$$\frac{dy}{dt} = 3y, y(0) = 1 \tag{1}$$

0.1.1 Method of ode45

We can solve this using the MATLAB standard ode45 function, also plot the exact solution y = exp(3t) on the interval [0,2]. The code is attached below:

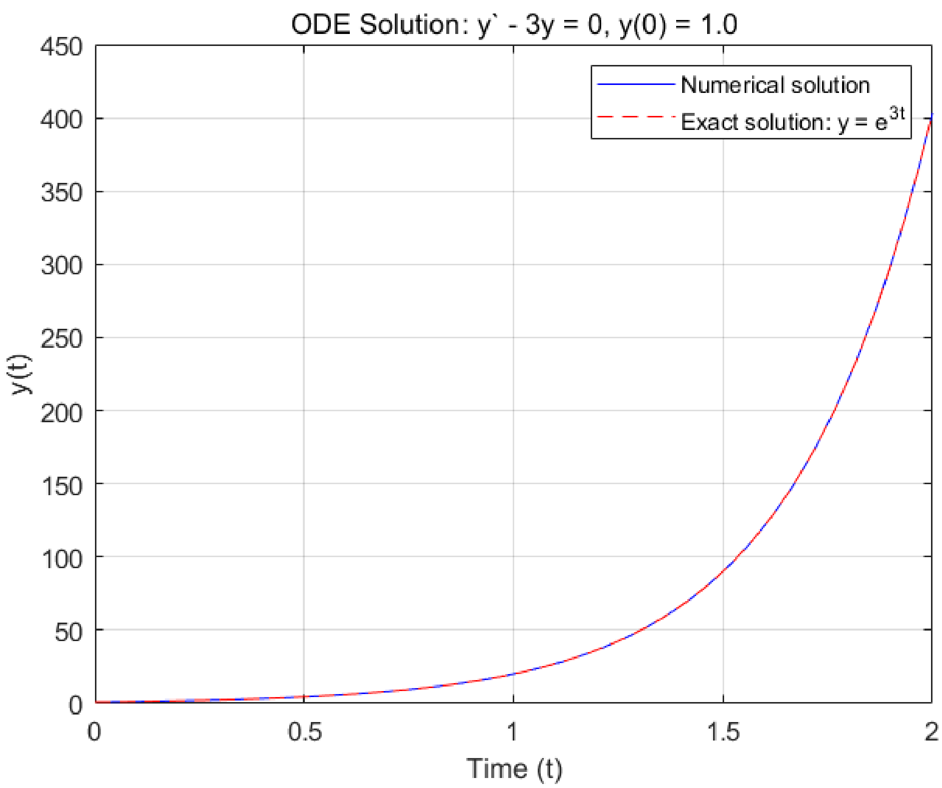

Figure 1: Solution

$f_y = @(t, y)3 * y;$
$tspan = [0, 2];$
y0 = 1.0;
$[t, y] = ode45(f_y, tspan, y0);$
$y_exact = exp(3 * t);$
$plot(t, y,' b', t, y_exact,' r --');$
title('ODE Solution: y' - 3y = 0, y(0) = 1.0');
$legend('Numericalsolution',' Exactsolution : y = exp(3t)');$
xlabel('Time (t)');
ylabel('y(t)');
grid on;

0.1.2 Euler's Method

The code in Listing 4.5 shows how to write a function to solve the differential equation on [0, 2], using Euler's method, with N = 20 equally spaced time-steps h = 0.1. Using this code, compare, on the same figure, the solutions graphically for h = 0.1, h = 0.05, h = 0.01 and the exact solution.

$tend = 1; trange = [0, tend];$
$Npts = 10;$
$y0 = 1;$
$[tsol, ysol] = odeEuler(@rhs, trange, y0, Npts);$
plot(tsol, ysol,'b'); hold on;
plot(tsol, exp(3*tsol),'g');
$functionydot = rhs(t, y)$
$ydot = 3 * y;$
$function[t, y] = odeEuler(fcn, trange, y0, Npts)$
$h = trange(end)/Npts;$
$t = zeros(1, Npts); y = zeros(1, Npts);$
$y(1) = y0; t(1) = trange(1);$
$fork = 1 : Npts$
$y(k + 1) = y(k) + h * fcn(t(k), y(k));$
$t(k + 1) = t(k) + h;$
end

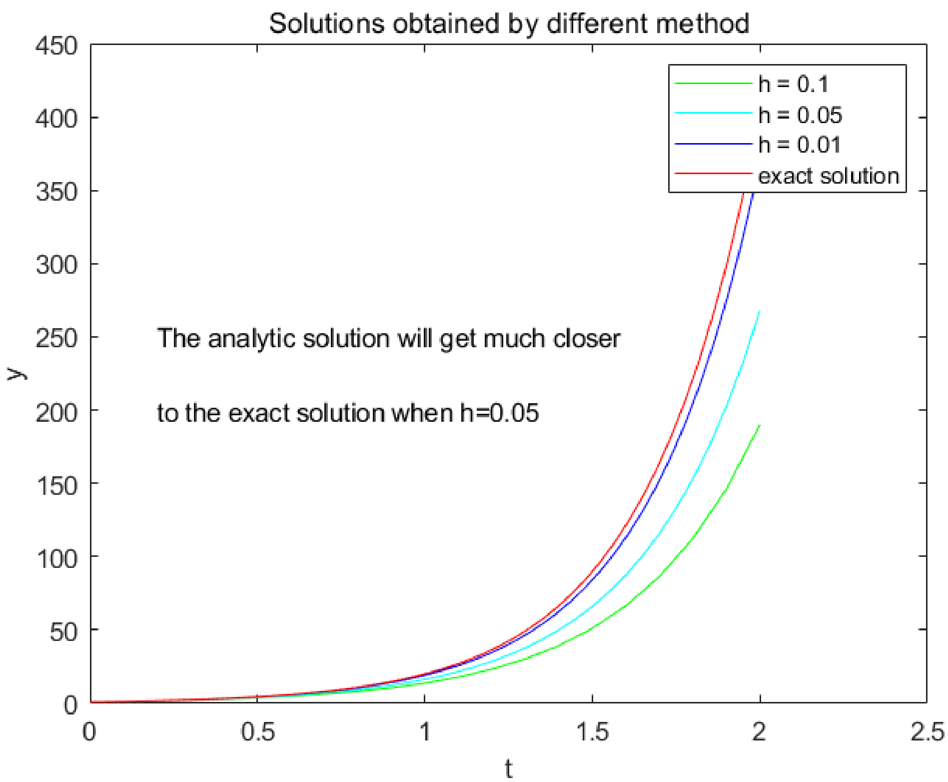

Figure 2: Comparing Solution

The code is written below:
tend = 2; trange = [0, tend];
Npts1 = 20; Npts2 = 40; Npts3 = 200;
y0 = 1;

```
[tsol1, ysol1] = odeEuler(@rhs, trange, y0, Npts1);
[tsol2, ysol2] = odeEuler(@rhs, trange, y0, Npts2);
[tsol3, ysol3] = odeEuler(@rhs, trange, y0, Npts3);
plot(tsol1,ysol1,'g'); hold on;
plot(tsol2,ysol2,'C'); hold on;
plot(tsol3,ysol3,'b'); hold on;
plot(tsol3,exp(tsol3*3),'r');
title('Solutions obtained by different method')
xlabel('t'); ylabel('y');
legend('h = 0.1','h = 0.05','h = 0.01','exact solution')
text(0.2,250,'The analytic solution will get much closer')
text(0.2,200,'to the exact solution when h=0.05')
function ydot = rhs(t, y)
ydot = 3*y;
end
function [t, y] = odeEuler(fcn, trange, y0, Npts)
h = trange(end)/Npts;
t = zeros(1,Npts); y = zeros(1,Npts);
y(1) = y0; t(1)=trange(1);
for k=1:Npts
y(k+1) = y(k) + h*fcn(t(k),y(k));
t(k+1) = t(k) + h;
end
end
```

0.1.3 Huen's Method

Modify the code to solve the equation using Huen's method and compare the solution with Euler's method and the exact solution

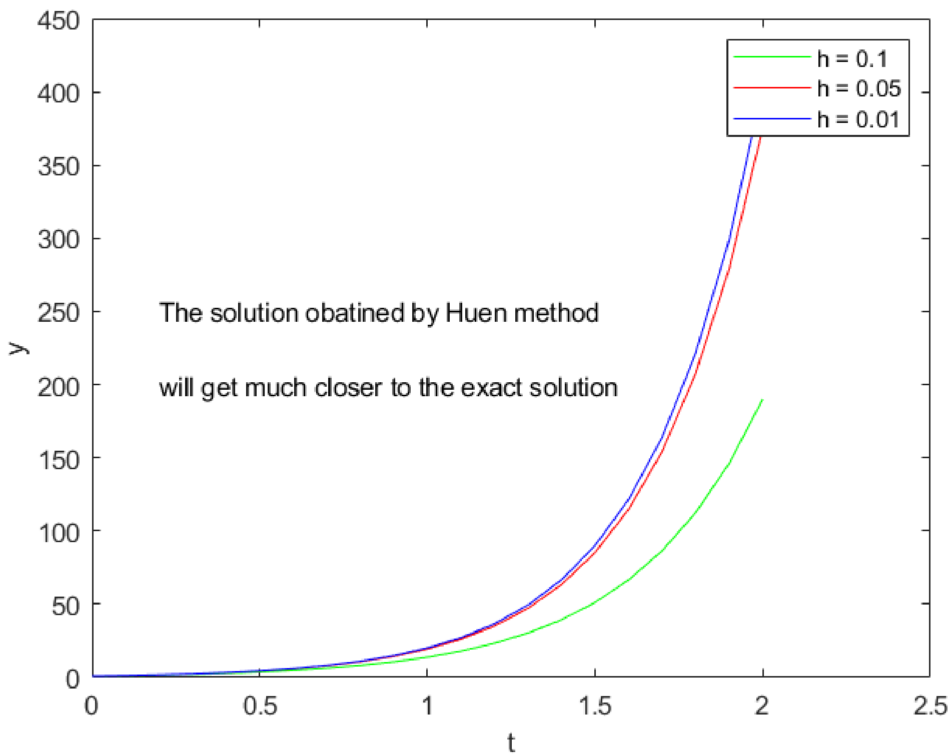

Figure 3: Comparing Solution

The code is written below:
for n = 20 points, h = 0.1.

```
tend = 2; trange = [0, tend];
Npts = 20;
y0 = 1;
[tsol, ysol1] = odeEuler(@rhs, trange, y0, Npts);
[tsol, ysol2] = odeHuen(@rhs, trange, y0, Npts);
plot(tsol,ysol1,'g'); hold on;
plot(tsol,ysol2,'r'); hold on;
plot(tsol,exp(tsol*3),'b');
legend('h = 0.1','h = 0.05','h = 0.01','exact solution')
text(0.2,250,'The solution obatined by Huen method ')
text(0.2,200,'will get much closer to the exact solution')
xlabel('t'); ylabel('y');
function ydot = rhs(t, y)
ydot = 3*y;
end
function [t, y] = odeEuler(fcn, trange, y0, Npts)
h = trange(end)/Npts;
t = zeros(1,Npts); y = zeros(1,Npts);
y(1) = y0; t(1)=trange(1);
```

```
for k=1:Npts
t(k+1) = t(k) + h;
y(k+1) = y(k) + h*fcn(t(k),y(k));
end
end
function [t, y] = odeHuen(fcn, trange, y0, Npts)
h = trange(end)/Npts;
t = zeros(1,Npts); y = zeros(1,Npts);
y(1) = y0; t(1)=trange(1);
for k=1:Npts
t(k+1) = t(k) + h;
y(k+1) = y(k) + h/2*(fcn(t(k),y(k))+fcn(t(k+1),y(k)+h*fcn(t(k),y(k))));
end
end
```