实验三 常微分方程的差分方法实验

一. 实验目的

（1）深入理解常微分方程的差分方法的原理，学会用差分方法解决某些实际的常微分方程问题，比较这些方法解题的不同之处。

（2）熟悉 Matlab 编程环境，利用 Matlab 实现具体的常微分方程。

二. 实验要求

用 Matlab 软件实现欧拉方法、改进的欧拉方法、龙格-库塔方法和亚当姆斯方法，并用实例在计算机上计算。

三. 实验内容

1. 实验题目

3-1： 用欧拉方法、改进的欧拉方法、四阶龙格-库塔方法求解初值问题：

$$\begin{cases} y' = \dfrac{2x}{3y^2} \\ y(0) = 1 \end{cases}$$

在区间[0,1]上取 h=0.1 时的数值解，并与精确解 $y = \sqrt[3]{1 + x^2}$ 进行比较。

3-2： 分别用四阶亚当姆斯方法、改进的四阶亚当姆斯预估校正系统求解题 3-1 中的初值问题。提示：可用四阶龙格-库塔方法求出开头三步的值。

2. 设计思想

要求针对上述题目，详细阐述每种算法的设计思想，并对每种方法的计算结果进行比较、分析。

参考教材内容以及书后代码，编写相应的欧拉方法、改进的欧拉方法、四阶龙格-库塔方法、四阶亚当姆斯方法、改进的四阶亚当姆斯方法计算其值。

3. 对应程序

列出每种算法的程序。

```
% MendEuler.m
function E = MendEuler(f,a,b,N,ya)
h = (b-a)/N;
y=zeros(1,N+1);
x=zeros(1,N+1);
y(1)=ya;
x=a:h:b;
for i=1:N
    y1=y(i)+h*feval(f,x(i),y(i));
    y2=y(i)+h*feval(f,x(i+1),y1);
    y(i+1)=(y1+y2)/2;
```

```
end
E=[x',y'];

% Rungkuta4.m
function R=Rungkuta4(f,a,b,N,ya)
h=(b-a)/N;
y=zeros(1,N+1);
x=a:h:b;
y(1)=ya;
for i=1:N
    k1=feval(f,x(i),y(i));
    k2=feval(f,x(i)+h/2,y(i)+(h/2)*k1);
    k3=feval(f,x(i)+h/2,y(i)+(h/2)*k2);
    k4=feval(f,x(i)+h,y(i)+h*k3);
    y(i+1)=y(i)+(h/6)*(k1+2*k2+2*k3+k4);
end
R=[x',y'];

% Adams4PC.m
function A = Adams4PC(f,a,b,N,ya)
if N<4
    return;
end
h=(b-a)/N;
y=zeros(1,N+1);
x=a:h:b;
y(1)=ya;
for i =1:N
    if i<4
        k1=feval(f,x(i),y(i));
        k2=feval(f,x(i)+h/2,y(i)+(h/2)*k1);
        k3=feval(f,x(i)+h/2,y(i)+(h/2)*k2);
        k4=feval(f,x(i)+h,y(i)+h*k3);
        y(i+1)=y(i)+(h/6)*(k1+2*k2+2*k3+k4);
    else
        F=feval(f,x(i-3:i),y(i-3:i));
        py=y(i)+(h/24)*(F*[-9,37,-59,55]');
        p=feval(f,x(i+1),py);
        F=[F(2) F(3) F(4) p];
        y(i+1)=y(i)+(h/24)*(F*[1,-5,19,9]');
    end
end
A=[x',y'];
```

```
% CAdams4PC.m
function A = CAdams4PC(f,a,b,N,ya)
if N<4
    return;
end
h=(b-a)/N;
y=zeros(1,N+1);
x=a:h:b;
y(1)=ya;
for i =1:N
    if i<4
         k1=feval(f,x(i),y(i));
        k2=feval(f,x(i)+h/2,y(i)+(h/2)*k1);
        k3=feval(f,x(i)+h/2,y(i)+(h/2)*k2);
        k4=feval(f,x(i)+h,y(i)+h*k3);
        y(i+1)=y(i)+(h/6)*(k1+2*k2+2*k3+k4);
    elseif i==4
        F=feval(f,x(i-3:i),y(i-3:i));
        py=y(i)+(h/24)*(F*[-9,37,-59,55]');
        p=feval(f,x(i+1),py);
        F=[F(2) F(3) F(4) p];
        y(i+1)=y(i)+(h/24)*(F*[1,-5,19,9]');
        p=py;c=y(i+1);
    else
        F=feval(f,x(i-3:i),y(i-3:i));
        py=y(i)+(h/24)*(F*[-9,37,-59,55]');
        my=py-251*(p-c)/270;
        m=feval(f,x(i+1),my);
        F=[F(2) F(3) F(4) m];
        cy=y(i)+(h/24)*(F*[1,-5,19,9]');
        y(i+1)=cy+19*(py-cy)/270;
        p=py;c=cy;
    end
end
A=[x',y'];

f0 = @(x,y) 2.*x ./(3.*y.*y);
disp('欧拉方法:')
disp(1+0.1*f0(0,1))
disp('改进的欧拉方法:')
disp(MendEuler(f0,0,1,10,1))
disp('四阶龙格-库塔方法:')
disp(Rungkuta4(f0,0,1,10,1))
disp('四阶亚当姆斯预估校正系统:')
```

```
disp(Adams4PC(f0,0,1,10,1))
disp('改进的四阶亚当姆斯预估校正系统:')
disp(CAdams4PC(f0,0,1,10,1))
```

4. 实验结果

```
f0 = @(x,y) 2.*x ./(3.*y.*y);
disp('欧拉方法:')
disp(1+0.1*f0(0,1))
disp('改进的欧拉方法:')
disp(MendEuler(f0,0,1,10,1))
disp('四阶龙格-库塔方法:')
disp(Rungkuta4(f0,0,1,10,1))
disp('四阶亚当姆斯预估校正系统:')
disp(Adams4PC(f0,0,1,10,1))
disp('改进的四阶亚当姆斯预估校正系统:')
disp(CAdams4PC(f0,0,1,10,1))
20002271 王麒淞
欧拉方法:
     1

改进的欧拉方法:
                   0   1.000000000000000
   0.100000000000000   1.003333333333333
   0.200000000000000   1.013180434398852
   0.300000000000000   1.029171244550309
   0.400000000000000   1.050751079998023
   0.500000000000000   1.077252310612832
   0.600000000000000   1.107965053358377
   0.700000000000000   1.142194135689444
   0.800000000000000   1.179297284217600
   0.900000000000000   1.218705575564524
   1.000000000000000   1.259930265862033
```

四阶龙格-库塔方法:

0	1.000000000000000
0.100000000000000	1.003322292719565
0.200000000000000	1.013159438200695
0.300000000000000	1.029142535439115
0.400000000000000	1.050717679021904
0.500000000000000	1.077217479999272
0.600000000000000	1.107931808368870
0.700000000000000	1.142164929384162
0.800000000000000	1.179273883780467
0.900000000000000	1.218689083410464
1.000000000000000	1.259921221582087

四阶亚当姆斯预估校正系统:

0	1.000000000000000
0.100000000000000	1.003322292719565
0.200000000000000	1.013159438200695
0.300000000000000	1.029142535439115
0.400000000000000	1.050720005701320
0.500000000000000	1.077222006226289
0.600000000000000	1.107937969725546
0.700000000000000	1.142171994240436
0.800000000000000	1.179281183806477
0.900000000000000	1.218696130520936
1.000000000000000	1.259927725124502

改进的四阶亚当姆斯预估校正系统:

0	1.000000000000000
0.100000000000000	1.003322292719565
0.200000000000000	1.013159438200695
0.300000000000000	1.029142535439115
0.400000000000000	1.050720005701320
0.500000000000000	1.077219577281977
0.600000000000000	1.107933404741466
0.700000000000000	1.142165924289030
0.800000000000000	1.179274343902799
0.900000000000000	1.218689154161893
1.000000000000000	1.259921053872529

四. 实验体会

对实验过程进行分析总结，对比求解常微分方程的不同方法，指出每种算法的设计要点及应注意的事项，以及自己通过实验所获得的对常微分方程的差分方法的理解。

通过常微分方程的差分方法实验，进一步对常微分方程求解的方法有了理解与感悟。可以更加熟练的针对不同的要求应用和设计出不同的算法来计算，并且对于应用 matlab 求解常微分方程有了认识，对于 matlab 的操作使用更加的准确纯熟。

对于常微分方程的各种算法的精度在此进行详细的分析。显式的 Euler 格式虽然很结构简单、计算量小，但是它的精度很低；改进的 Euler 格式，相对于 Euler 格式，明显的改善了精度，并且计算量也是可取的。四阶 Runge-Kutta 格式具有更高的精度，但是计算量比较大。四阶 Adams 预报校正系统是在 Runge-Kutta 的基础上进行修改，改善了精度以及计算量。改进的四阶 Adams 预报矫正系统效果最好。

（注：不要改变实验报告的结构，写清页码和题号，源程序以自己的姓名命名，如 3-1 题可命名为“Zhangsan_3-1.m”，运行截图中应出现自己的姓名和题号）