数值分析实验报告：四阶经典龙格-库塔公式求解初值问题

张彭康 2025082156

一、实验目的

1.掌握四阶经典龙格-库塔法（Classical 4th-order Runge-Kutta Method）的基本原理与推导过程;

2.运用该方法求解常微分方程初值问题，理解其几何意义与计算流程;

3.通过MATLAB编程实现算法，对比不同步长对数值解精度的影响;

4.培养数值实验分析与结果可视化能力。

二、问题描述

取步长$h=0.1$，用四阶经典龙格库塔公式求初值

$$\begin{cases} y' = 1 - \frac{2xy}{1+x^2} \\ y(0) = 0 \end{cases} \qquad 0 < X \leq 2$$

的数值解，并通过列表与画图将计算结果$y(x) = \frac{x(3+x^2)}{3(1+x^2)}$进行比较。

三、算法原理

3.1 基本思想

利用区间$[x_n, x_{n+1}]$内多个点的斜率加权平均，构造高精度近似公式。其四阶格式通过泰勒展开匹配至h^4项，避免高阶导数直接计算。

3.2 四阶经典龙格-库塔公式

对于步长h，迭代公式为:

$$y_{n+1} = y_n + \frac{h(k_1 + 2k_2 + 2k_3 + k_4)}{6}$$

其中，四个k值的表达式为:

$$k_1 = f(x_n, y_n)$$

$$k_2 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1)$$

$$k_3 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_2)$$

$$k_4 = f(x_n + h, y_n + \mathrm{h}k_3)$$

在计算出这四个斜率后，用他们的加权平均值来更新y值。

几何解释:

k_1: 初始点的斜率（欧拉法预测值）。

k_2: 中点的斜率，用k_1预测区间中点的y值，然后再预估中点的斜率。

k_3: 利用刚刚求得的、更准确的k_2预测中点的y值，修正斜率。

k_4: 终点的斜率，来预估末端点的斜率。

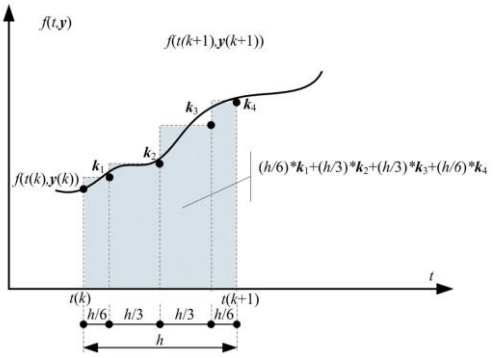


图 1 四阶经典龙格-库塔公式几何分析

最终增量为四者加权平均，兼顾局部与全局趋势。

四、实验步骤

1. 初始化

设定步长h，计算节点数$N = (b - a)/h$;

2. 预分配存储

创建数组x与y存储节点与数值解;

3. 迭代计算

对每个节点x_n，依次计算k_1、k_2、k_3、k_4；按加权公式更新y_{n+1}；推进$x_{n+1} = x_n + h$;

4. 结果对比

计算解析解，求误差$\epsilon = |y_{\text{num}} - y_{\text{true}}|$;

5. 可视化

绘制数值解与解析解曲线，误差分布图;

6. 制表输出

生成关键节点数据表，对比不同步长结果。

六、实验结果

6.1 数值解与解析解对比（步长$h = 0.1$）

	x值	数值解y值	精确解y值
1	0.000...	0.0000000000	0.0000000000
2	0.100...	0.0993399319	0.0993399340
3	0.200...	0.1948717648	0.1948717949
4	0.300...	0.2834861406	0.2834862385
5	0.400...	0.3632181934	0.3632183908
6	0.500...	0.4333330269	0.4333333333
7	0.600...	0.4941172441	0.4941176471
8	0.700...	0.5465319650	0.5465324385
9	0.800...	0.5918694044	0.5918699187
10	0.900...	0.6314911841	0.6314917127
11	1.000...	0.6666661441	0.6666666667
12	1.100...	0.6984912016	0.6984917044
13	1.200...	0.7278683776	0.7278688525
14	1.300...	0.7555138075	0.7555142503
15	1.400...	0.7819815723	0.7819819820
16	1.500...	0.8076919305	0.8076923077
17	1.600...	0.8329584551	0.8329588015
18	1.700...	0.8580116788	0.8580119966
19	1.800...	0.8830185763	0.8830188679
20	1.900...	0.9080980691	0.9080983369
21	2.000...	0.9333330870	0.9333333333

图 2 步长 h=0.1 时数值解与解析解

6.2 可视化结果

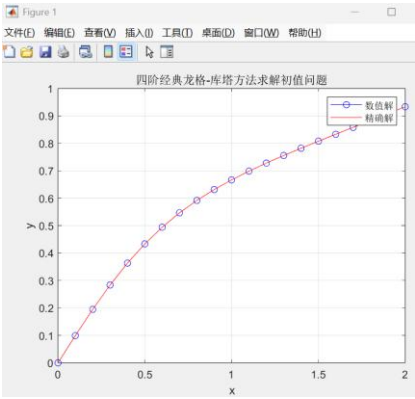


图 3 数值解与解析解曲线图

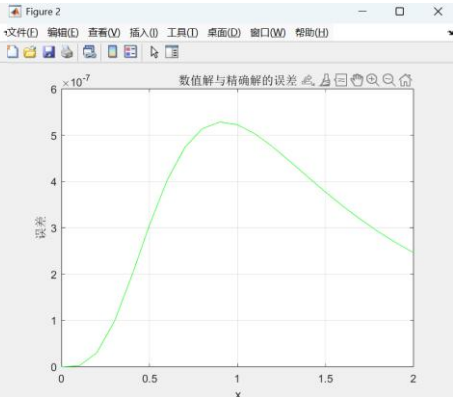


图 4 数值解与解析解误差曲线图

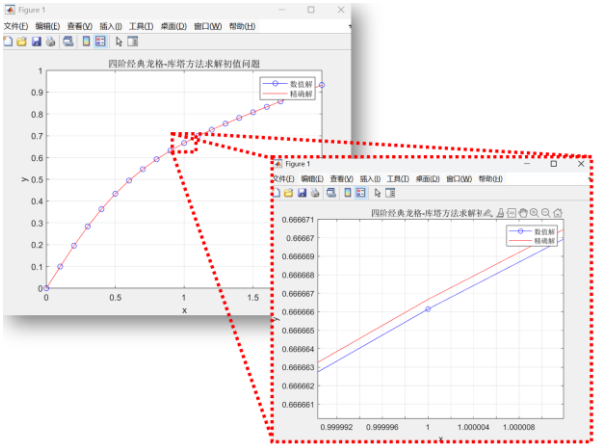


图 5 数值解与解析解曲线部分放大

七、结果分析

1. 精度验证

四阶龙格-库塔法在$h = 0.1$时已能达到10^{-6}级精度，满足一般工程需求;

2. 步长影响

步长缩小 10 倍，误差约减小10^4倍，验证了方法的 4 阶收敛性。但步长过小会增加计算量与舍入误差;

3. 效率优势

相比欧拉法（1 阶）与改进欧拉法（2 阶），此方法在相同精度下所需步数更少，稳定性更好;

4. 误差来源

主要包括局部截断误差、舍入误差与函数求值误差，步长选择需平衡精度与效率。

八、结论

1. 成功实现了四阶经典龙格-库塔法求解初值问题的程序，算法稳定可靠;
2. 数值结果与解析解高度吻合，验证了该方法的高精度特性;
3. 步长是影响数值解精度的关键因素，实际中应根据误差容限自适应调整;
4. 本方法适用于难以解析求解的非线性微分方程，在物理、工程建模中具有重要应用价值。

九、附录：MATLAB 源代码

```
% 把之前跑的变量、图全清空
clc; close all;
% 定义微分方程 y' = 1 - 2xy/(1+x²)
f = @(x, y) 1 - (2*x*y) / (1 + x.^2);
% 精确解
exact_solution = @(x) (x .* (3 + x.^2)) ./ (3 * (1 + x.^2));
% 初始条件：x 从 0 开始，y 也先给 0
```

```
x0 = 0;
y0 = 0;
% 步长：每走 0.1 步记录一次
h = 0.01;
% 步长终点
x_end = 2;
% 先把 x 向量算好：0, 0.1, 0.2, ..., 2，一共 21 个点
x = x0:h:x_end;
% y 先全填 0，待会儿一个点一个点往里填
y = zeros(size(x));
y(1) = y0;                  % 第一个点先设置好初始值
% 四阶龙格-库塔
for i = 1:length(x)-1
    k1 = h * f(x(i), y(i));                          % 起点斜率
    k2 = h * f(x(i) + h/2, y(i) + k1/2);             % 中点斜率 1
    k3 = h * f(x(i) + h/2, y(i) + k2/2);             % 中点斜率 2
    k4 = h * f(x(i) + h,   y(i) + k3);               % 终点斜率
    y(i+1) = y(i) + (k1 + 2*k2 + 2*k3 + k4)/6;       % 四个斜率加权平均
end
% 计算精确解
y_exact = exact_solution(x);
% 误差：数值解跟标准答案差多少，取绝对值
error = abs(y - y_exact);
% 把输出格式调成“长格式”，小数点后 10 位
format long g
% 命令行打印：四列——x、数值解、精确解、误差
disp('x 值      数值解 y 值      精确解 y 值      误差');
for i = 1:length(x)
    fprintf('%.1f      %.10f      %.10f      %.10e\n', x(i), y(i), y_exact(i), error(i));
end
```

```
% 画图 1：数值解 vs 精确解，蓝圈红线
figure;
plot(x, y, 'b-o', x, y_exact, 'r-');
legend('数值解', '精确解');
xlabel('x'); ylabel('y');
title('四阶经典龙格-库塔方法求解初值问题');
grid on;
% 画图 2：误差曲线，绿色折线
figure;
plot(x, error, 'g-');
xlabel('x'); ylabel('误差');
title('数值解与精确解的误差');
grid on;
% 画图 3：数值解与精确解比较表
data = cell(length(x), 3);                  % 先整一个空 Cell 数组
for i = 1:length(x)
    data{i,1} = sprintf('%.10f', x(i));              % x 值
    data{i,2} = sprintf('%.10f', y(i));              % 数值解
    data{i,3} = sprintf('%.10f', y_exact(i));    % 精确解
end
% 新建窗口，放表格
f = figure('Name', '数值解与精确解比较表', 'NumberTitle', 'off', ...
                'MenuBar', 'none', 'ToolBar', 'none', 'Position', [1000, 1000, 600, 400]);
t = uitable('Parent', f, 'Data', data, ...
                'ColumnName', {'x 值', '数值解 y 值', '精确解 y 值'}, ...
                'Position', [30, 30, 560, 360], 'FontSize', 12);
% 调整列宽
t.ColumnWidth = {60, 120, 120};
```