

# 计算方法实验（实验 3）

2022 年 4 月 19 日

数据显示结果已保留 4 位小数。

## 1 实验题目 3：四阶龙格——库塔方法

### 1.1 问题分析

准确描述并总结出实验题目（摘要），并准确分析原题的目的和意义。

给定常微分方程初值问题：

$$\begin{cases} \frac{dy}{dx} = f(x, y), & a \leq x \leq b \\ y(a) = \alpha, & h = \frac{b-a}{N} \end{cases}$$

求其数值解  $y_n, n = 1, 2, \dots, N$ 。

#### 1.1.1 实验目的

输入：  $a, b, \alpha, N$

输出：初值问题的数值解  $x_n, y_n, n = 0, 1, 2, \dots, N$

### 1.2 数学原理

数学原理表达清晰且书写准确。

记  $x_n = a + n \times h, n = 0, 1, \dots, N$ ，利用四阶龙格——库塔方法：

$$\begin{aligned}
K_1 &= hf(x_n, y_n) \\
K_2 &= hf(x_n + \frac{h}{2}, y_n + \frac{K_1}{2}) \\
K_3 &= hf(x_n + \frac{h}{2}, y_n + \frac{K_2}{2}) \\
K_4 &= hf(x_n + h, y_n + K_3) \\
y_{n+1} &= y_n + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4) \\
n &= 0, 1, \dots, N-1
\end{aligned}$$

即可逐次求出微分方程初值问题的数值解  $x_n, y_n, n = 0, 1, 2, \dots, N$ 。

### 1.3 程序设计流程

编译通过，根据输入能得到正确输出。

[126]: # 引入需要的包

```
from typing import *
import numpy as np
from pandas import DataFrame
```

[127]: # 四阶龙格——库塔方法

```
def runge_kutta(
    f: Callable[[float, float], float],
    a: float, b: float, alpha: float, N: int):
    x_list, y_list = [], []
    h = (b - a) / N
    x, y = a, alpha
    x_list.append(x)
    y_list.append(y)
    for _ in range(N):
        k_1 = h*f(x, y)
        k_2 = h*f(x+h/2, y+k_1/2)
        k_3 = h*f(x+h/2, y+k_2/2)
        k_4 = h*f(x+h, y+k_3)
        x = x + h
        y = y + (k_1 + 2*k_2 + 2*k_3 + k_4) / 6
```

```

        x_list.append(x)
        y_list.append(y)
    return x_list, y_list

```

[128]: # 运行测试参数

```

global_args = [
    [lambda x, y: x + y, 0, 1, -1, [5, 10, 20], lambda x: -x-1, "问题 1 (1)"],
    [lambda x, y: -y**2, 0, 1, 1, [5, 10, 20],
     lambda x: 1 / (x + 1), "问题 1 (2)"],
    [lambda x, y: 2 * y / x + x**2 +
     np.exp(x), 1, 3, 0, [5, 10, 20], lambda x: x**2 * (np.exp(x) - np.e),
     ↪ "问题 2 (1)"],
    [lambda x, y: (y + y**2) / x, 1, 3, -2, [5, 10, 20],
     lambda x: 2 * x / (1 - 2 * x), "问题 2 (2)"],
    [lambda x, y: -20 * (y-x**2) + 2 * x, 0, 1, 1.0 / 3,
     [5, 10, 20], lambda x: x**2 + np.exp(-20*x)/3, "问题 3 (1)"],
    [lambda x, y: -20 * y + 20 *
     np.sin(x) + np.cos(x), 0, 1, -1, [5, 10, 20], lambda x: np.exp(-20*x) +
     ↪ np.sin(x), "问题 3 (2)"],
    [lambda x, y: -20*(y-np.exp(x)*np.sin(x)) + np.exp(x)*(np.sin(x) + np.
     ↪ cos(x)),
     0, 1, 0, [5, 10, 20], lambda x: np.exp(x)*np.sin(x), "问题 3 (3)"]
]

```

[129]: # 求数据的均方误差

```

def get_error(f: Callable[[float], float], data):
    x, y = data
    standard = np.array([f(x_i) for x_i in x])
    return sum((y - standard) ** 2) / len(x)

```

[130]: # 运行一次

```

def run(index: int):
    res = []
    for n in global_args[index][-3]:
        data = runge_kutta(*[
            *global_args[index][: -3], n

```

```

    ])
    error = get_error(global_args[index][-2], data)
    res.append({
        "N": n,
        "标号": global_args[index][-1],
        "均方误差": error,
        "x": data[0],
        "y": data[1],
    })
    return res

```

[131]: # 运行所有并且返回结果表格

```

def run_all():
    all_data = [run(i) for i in range(len(global_args))]
    all = []
    for d in all_data:
        all.extend(d)
    return DataFrame(all)

run_all()

```

```

[131]:      N  ...                                     y
0     5  ...  [-1, -1.2, -1.4, -1.5999999999999999, -1.79999...
1    10  ...  [-1, -1.1, -1.20000000000000002, -1.300000000000...
2    20  ...  [-1, -1.05, -1.1, -1.15000000000000001, -1.2000...
3     5  ...  [1, 0.8333390356230387, 0.7142921304635431, 0...
4    10  ...  [1, 0.9090911863322196, 0.8333337288430721, 0...
5    20  ...  [1, 0.9523809630269818, 0.9090909268125394, 0...
6     5  ...  [0, 2.6076891538492872, 8.124196625549118, 17...
7    10  ...  [0, 1.0055508321940254, 2.613659182614748, 4.9...
8    20  ...  [0, 0.434963271851454, 1.0058077554774965, 1.7...
9     5  ...  [-2, -1.5539889980952382, -1.3836172899114931,...
10    10  ...  [-2, -1.7142451804511538, -1.5555228848496192,...
11    20  ...  [-2, -1.8333328294259301, -1.7142851698413297,...
12     5  ...  [0.3333333333333333, 2.506666666666667, 11.69...
13    10  ...  [0.3333333333333333, 0.2511111111111111, 0.363...

```

```

14 20 ... [0.3333333333333333, 0.16437499999999997, 0.16...
15 5 ... [-1, -4.802661893779973, -24.623829295619263, ...
16 10 ... [-1, -0.2335276701694724, 0.0874382457825747, ...
17 20 ... [-1, -0.32502148139805487, -0.0407937778694009...
18 5 ... [0, 0.2986462127501341, 0.927219870027348, 2.8...
19 10 ... [0, 0.11205510913037421, 0.2451165144244346, 0...
20 20 ... [0, 0.05259503995574239, 0.11040898628183947, ...

```

[21 rows x 5 columns]

为防止输出 PDF 时表格格式被破坏，在此放入上方表格的图片。

	N	标号	均方误差	x	y
0	5	问题 1 (1)	2.465190e-32	[0, 0.2, 0.4, 0.6000000000000001, 0.8, 1.0]	[-1, -1.2, -1.4, -1.5999999999999999, -1.79999...
1	10	问题 1 (1)	3.182337e-31	[0, 0.1, 0.2, 0.30000000000000004, 0.4, 0.5, 0...	[-1, -1.1, -1.2000000000000002, -1.30000000000...
2	20	问题 1 (1)	2.206932e-31	[0, 0.05, 0.1, 0.15000000000000002, 0.2, 0.25,...	[-1, -1.05, -1.1, -1.1500000000000001, -1.2000...
3	5	问题 1 (2)	2.569560e-11	[0, 0.2, 0.4, 0.6000000000000001, 0.8, 1.0]	[1, 0.8333390356230387, 0.7142921304635431, 0....
4	10	问题 1 (2)	1.282857e-13	[0, 0.1, 0.2, 0.30000000000000004, 0.4, 0.5, 0...	[1, 0.9090911863322196, 0.8333337288430721, 0....
5	20	问题 1 (2)	5.366889e-16	[0, 0.05, 0.1, 0.15000000000000002, 0.2, 0.25,...	[1, 0.9523809630269818, 0.9090909268125394, 0....
6	5	问题 2 (1)	2.023870e+03	[1, 1.4, 1.7999999999999998, 2.199999999999999...	[0, 2.6076891538492872, 8.124196625549118, 17....
7	10	问题 2 (1)	1.541237e+03	[1, 1.2, 1.4, 1.5999999999999999, 1.7999999999...	[0, 1.0055508321940254, 2.613659182614748, 4.9...
8	20	问题 2 (1)	1.315425e+03	[1, 1.1, 1.2000000000000002, 1.300000000000000...	[0, 0.434963271851454, 1.0058077554774965, 1.7...
9	5	问题 2 (2)	7.459298e-07	[1, 1.4, 1.7999999999999998, 2.199999999999999...	[-2, -1.5539889980952382, -1.3836172899114931,...
10	10	问题 2 (2)	4.401675e-10	[1, 1.2, 1.4, 1.5999999999999999, 1.7999999999...	[-2, -1.7142451804511538, -1.5555228848496192,...
11	20	问题 2 (2)	8.946298e-14	[1, 1.1, 1.2000000000000002, 1.300000000000000...	[-2, -1.8333328294259301, -1.7142851698413297,...
12	5	问题 3 (1)	3.263086e+05	[0, 0.2, 0.4, 0.6000000000000001, 0.8, 1.0]	[0.3333333333333333, 2.5066666666666677, 11.69...
13	10	问题 3 (1)	4.889478e-01	[0, 0.1, 0.2, 0.30000000000000004, 0.4, 0.5, 0...	[0.3333333333333333, 0.2511111111111111, 0.363...
14	20	问题 3 (1)	4.815643e-01	[0, 0.05, 0.1, 0.15000000000000002, 0.2, 0.25,...	[0.3333333333333333, 0.16437499999999997, 0.16...
15	5	问题 3 (2)	1.697642e+06	[0, 0.2, 0.4, 0.6000000000000001, 0.8, 1.0]	[-1, -4.802661893779973, -24.623829295619263, ...
16	10	问题 3 (2)	3.852943e-01	[0, 0.1, 0.2, 0.30000000000000004, 0.4, 0.5, 0...	[-1, -0.2335276701694724, 0.0874382457825747, ...
17	20	问题 3 (2)	2.209629e-01	[0, 0.05, 0.1, 0.15000000000000002, 0.2, 0.25,...	[-1, -0.32502148139805487, -0.0407937778694009...
18	5	问题 3 (3)	3.617925e+02	[0, 0.2, 0.4, 0.6000000000000001, 0.8, 1.0]	[0, 0.2986462127501341, 0.927219870027348, 2.8...
19	10	问题 3 (3)	9.913683e-06	[0, 0.1, 0.2, 0.30000000000000004, 0.4, 0.5, 0...	[0, 0.11205510913037421, 0.2451165144244346, 0...
20	20	问题 3 (3)	1.212561e-08	[0, 0.05, 0.1, 0.15000000000000002, 0.2, 0.25,...	[0, 0.05259503995574239, 0.11040898628183947, ...

## 1.4 实验结果

准确规范地给出各个实验题目的结果，并对相应的思考题给出正确合理的回答与说明。

实验数据结果如上表所示。

思考题：

1. 对实验 1，数值解和解析解相同吗？为什么？试加以说明。

在误差范围内基本可以认为相同。由上表可知，对问题 1，当  $N = 20$  时，其结果和标准值的

均方误差均小于  $10^{-15}$ ，都是非常小的，所以在误差范围内可以认为数值解和解析解相同。

2. 对实验 2， $N$  越大越精确吗？试加以说明。

在实验 2 的数据中，随着  $N$  的增大，其均方误差越来越小，所以对实验二， $N$  越大越精确。

3. 对实验 3， $N$  较小会出现什么现象？试加以说明。

在实验 3 的数据中，当  $N$  较小时，其均方误差非常大，达到  $10^2$  甚至  $10^6$ 。