

实验报告一 拉格朗日 (Lagrange) 插值

第一部分：问题分析 (描述并总结出实验题目)

已知区间 $[a, b]$ 上的实值函数 $f(x)$ 在 $n + 1$ 个不同的数据点 x_k 处的值是 $y_k = f(x_k)$, $k = 0, 1, \dots, n$, 要求利用拉格朗日插值多项式 $P_n(x)$ 计算 $f(x)$ 在 $[a, b]$ 中某点 x 处的近似值.

输入: $n + 1$ 个数据点 $(x_k, f(x_k))$, $k = 0, 1, \dots, n$; 插值点 x

输出: $f(x)$ 在插值点 x 的近似值 $P_n(x)$

- 问题 1 拉格朗日插值多项式的次数 n 越大越好吗?
- 问题 2 插值区间越小越好吗?
- 问题 4 考虑拉格朗日插值问题, 内插比外推更可靠吗?

目的: 利用拉格朗日插值多项式 $P_n(x)$ 计算 $f(x)$ 的近似值.

意义:

1. 学习根据实际问题建立数学模型、针对数学模型的特点确定适当的计算方法、编制出计算机能够执行的计算程序、输入计算机、进行调试、完成运算等数值计算的过程。

2. 不能只会套用教科书中的标准程序进行数值计算, 要独立地将学过的数值算法编制成计算机程序, 灵活应用已经掌握的算法求解综合性较大的课题。

3. 加深对数值计算程序结构化思想的印象, 提高编程能力, 加深对“计算方法”课程内容的理解和掌握, 进一步奠定从事数值计算工作的基础。

4. 可以利用所掌握的“高级语言”顺利地编制出计算机程序,

上机实习，完成实验环节的教学要求。不简单地套用现成的标准程序完成实验题目，把重点放在对算法的理解、程序的优化设计、上机调试和计算结果分析上，达到实验课的目的。

第二部分：数学原理

给定平面上 $n + 1$ 个不同的数据点 $(x_k, f(x_k))$, $k = 0, 1, \dots, n$, $x_i \neq x_j, i \neq j$; 则满足条件:

$$P_n(x_k) = f(x_k), \quad k = 0, 1, \dots, n$$

的 n 次拉格朗日插值多项式

$$P_n(x) = \sum_{k=0}^n f(x_k) l_k(x)$$

是存在且唯一的，其中：

$$l_k(x) = \frac{(x - x_0) \dots (x - x_{k-1})(x - x_{k+1}) \dots (x - x_n)}{(x_k - x_0) \dots (x_k - x_{k-1})(x_k - x_{k+1}) \dots (x_k - x_n)},$$
$$k = 0, 1, \dots, n$$

称为拉格朗日插值基底.

若 $x_k \in [a, b]$, $k = 0, 1, \dots, n$, 且函数 $f(x)$ 充分光滑，则当 $x \in [a, b]$ 时，有误差估计式

$$f(x) - P_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} (x - x_0)(x - x_1) \dots (x - x_n), \quad \xi \in [a, b]$$

第三部分：程序设计流程

程序流程：

1. 置 $y = 0.0$; $k = 0$
2. 当 $k \leq n$ 时，执行 2.1—2.4
 - 2.1 置 $l = 1.0$;
 - 2.2 对 $j = 0, 1, \dots, k-1, k+1, \dots, n$,
$$\text{置 } l = l \cdot (x - x_j) / (x_k - x_j)$$
 - 2.3 置 $y = y + l \cdot f(x_k)$
 - 2.4 置 $k = k + 1$
3. 输出 x, y
4. 停机

代码实现：

```
1  function result = Lagrange(X, Y, x)
2  % 利用拉格朗日插值多项式P_n(x)求f(x)的近似值
3  % 输入：n+1个数据点(x_k, f(x_k))的横坐标向量X，纵坐标向量Y，k=0,1,...,n；插值点x
4  % 输出：f(x)在插值点x的近似值P_n(x)
5  y = 0.0;
6  k = 1;
7  n = length(X);
8  while k <= n
9      l = 1.0;
10     for j = 1:n
11         if j == k
12             continue;
13         end
14         l = l * (x - X(j)) / (X(k) - X(j));
15     end
16     y = y + l * Y(k);
17     k = k + 1;
18 end
19 result = [x, y];
20 end
```

第四部分：实验结果、结论与讨论

问题 1 拉格朗日插值多项式的次数 n 越大越好吗？

考虑下面两个拉格朗日插值问题：

(1) 设 $f(x) = \frac{1}{1+x^2}$, $x \in [-5, 5]$, 考虑等距节点的拉格朗日插值多项式 $P_n(x)$, 即将区间 $[-5, 5]$ 进行 n 等分, 记 $h = \frac{10.0}{n}$, $x_k = -5.0 + k \cdot h$, $k = 0, 1, \dots, n$, 构造 $P_n(x)$, 利用拉格朗日插值多项式 $P_n(x)$ 作为 $f(x)$ 的近似值。分别取 $n = 5$, $n = 10$, $n = 20$, 同时计算 $P_n(x)$ 在 $x = 0.75$, $x = 1.75$, $x = 2.75$, $x = 3.75$, $x = 4.75$ 处的函数值。

计算结果如下：(Command Window 截图及整理)

```
>> LagrangeCalc
Lagrange 1.(1)
n = 5
    0.7500    0.5290
    1.7500    0.3733
    2.7500    0.1537
    3.7500   -0.0260
    4.7500   -0.0157
n = 10
    0.7500    0.6790
    1.7500    0.1906
    2.7500    0.2156
    3.7500   -0.2315
    4.7500    1.9236
n = 20
    0.7500    0.6368
    1.7500    0.2384
    2.7500    0.0807
    3.7500   -0.4471
    4.7500  -39.9524
```

$n \setminus x$	0.7500	1.7500	2.7500	3.7500	4.7500
5	0.5290	0.3733	0.1537	-0.0260	-0.0157
10	0.6790	0.1906	0.2156	-0.2315	1.9236
20	0.6368	0.2384	0.0807	-0.4471	-39.9524
actual	0.6400	0.2462	0.1168	0.0664	0.0424

(2) 设 $f(x) = e^x$, $x \in [-1, 1]$, 考虑等距节点的拉格朗日插值多项式 $P_n(x)$, 即将区间 $[-1, 1]$ 进行 n 等分, 记 $h = \frac{10.0}{n}$, $x_k = -5.0 + k \cdot h$, $k = 0, 1, \dots, n$, 构造 $P_n(x)$, 利用拉格朗日插值多项式 $P_n(x)$ 作为 $f(x)$ 的近似值。分别取 $n = 5$, $n = 10$, $n = 20$, 同时计算 $P_n(x)$ 在 $x = -0.95$, $x = -0.05$, $x = 0.05$, $x = 0.95$ 处的函数值。

计算结果如下: (Command Window 截图及整理)

```
Lagrange 1.(2)
n = 5
-0.9500    0.3868
-0.0500    0.9512
0.0500     1.0513
0.9500     2.5858

n = 10
-0.9500    0.3867
-0.0500    0.9512
0.0500     1.0513
0.9500     2.5857

n = 20
-0.9500    0.3867
-0.0500    0.9512
0.0500     1.0513
0.9500     2.5857
```

$n \setminus x$	-0.95	-0.05	0.05	0.95
5	0.3868	0.9512	1.0513	2.5858
10	0.3867	0.9512	1.0513	2.5857
20	0.3867	0.9512	1.0513	2.5857
actual	0.3867	0.9512	1.0513	2.5857

根据（1）和（2）的计算结果可以得出，拉格朗日插值多项式的次数 n 不一定越大越好。

问题（1）中的函数 $f(x) = \frac{1}{1+x^2}$ 的拉格朗日插值多项式 $P_n(x)$ 在增大次数 n 时，在插值点 0.75, 1.75, 2.75 的近似值逐渐接近对应的函数值，但是在插值点 3.75, 4.75 的近似值却偏离对应函数值。而问题（2）中的函数 $f(x) = e^x$ 的拉格朗日插值多项式 $P_n(x)$ 在增大次数 n 时，没有出现上述情况，近似值越来越精确。

查询资料得知函数 $f(x) = \frac{1}{1+x^2}$ 的拉格朗日插值多项式 $P_n(x)$ 插值次数 n 越高，插值结果越偏离原函数的现象称为龙格现象，即多项式插值在区间边缘振荡。

问题 2 插值区间越小越好吗？

考虑下面两个拉格朗日插值问题：

（1）设 $f(x) = \frac{1}{1+x^2}$ ， $x \in [-1, 1]$ ，考虑等距节点的拉格朗日插值多项式 $P_n(x)$ ，即将区间 $[-1, 1]$ 进行 n 等分，记 $h = \frac{2.0}{n}$ ， $x_k = -1.0 + k \cdot h$ ， $k = 0, 1, \dots, n$ ，构造 $P_n(x)$ ，利用拉格朗日插值多项式 $P_n(x)$ 作为 $f(x)$ 的近似值。分别取 $n = 5$ ， $n = 10$ ， $n = 20$ ，同时计算 $P_n(x)$ 在 $x = -0.95$ ， $x = -0.05$ ， $x = 0.05$ ， $x = 0.95$ 处的函数值。

计算结果如下：

Lagrange 2.(1)

n = 5

-0.9500 0.5171

-0.0500 0.9928

0.0500 0.9928

0.9500 0.5171

n = 10

-0.9500 0.5264

-0.0500 0.9975

0.0500 0.9975

0.9500 0.5264

n = 20

-0.9500 0.5256

-0.0500 0.9975

0.0500 0.9975

0.9500 0.5256

$n \setminus x$	-0.95	-0.05	0.05	0.95
5	0.5171	0.9928	0.9928	0.5171
10	0.5264	0.9975	0.9975	0.5264
20	0.5256	0.9975	0.9975	0.5256
actual	0.5256	0.9975	0.9975	0.5256

(2) 设 $f(x) = e^x$, $x \in [-5, 5]$, 考虑等距节点的拉格朗日插值多项式 $P_n(x)$, 即将区间 $[-5, 5]$ 进行 n 等分, 记 $h = \frac{10.0}{n}$, $x_k = -5.0 + k \cdot h$, $k = 0, 1, \dots, n$, 构造 $P_n(x)$, 利用拉格朗日插值多项式 $P_n(x)$ 作为 $f(x)$ 的近似值。分别取 $n = 5$, $n = 10$, $n = 20$, 同时计算 $P_n(x)$ 在 $x = -4.75$, $x = -0.25$, $x = 0.25$, $x = 4.75$ 处的函数值。

计算结果如下：

Lagrange 2.(2)

n = 5

-4.7500 1.1470

-0.2500 1.3022

0.2500 1.8412

4.7500 119.6210

n = 10

-4.7500 -0.0020

-0.2500 0.7787

0.2500 1.2841

4.7500 115.6074

n = 20

-4.7500 0.0087

-0.2500 0.7788

0.2500 1.2840

4.7500 115.5843

$n \setminus x$	-4.75	-0.25	0.25	4.75
5	1.1470	1.3022	1.8412	119.6210
10	-0.0020	0.7787	1.2841	115.6074
20	0.0087	0.7788	1.2840	115.5843
<i>actual</i>	0.0087	0.7788	1.2840	115.5843

插值区间越小越好。对比在等分次数相同时，不同插值区间得到的计算结果的误差可以得出：在等分次数 n 相同的情况下，插值区间越大，误差越大。所以插值区间越小越好。

问题 4 考虑拉格朗日插值问题，内插比外推更可靠吗？

考虑下面四个拉格朗日插值问题：

(1) 设 $f(x) = \sqrt{x}$ ，关于以 $x_0 = 1$ ， $x_1 = 4$ ， $x_2 = 9$ 为节点的拉格朗日插值多项式 $P_2(x)$ ，利用拉格朗日插值多项式 $P_2(x)$ 作为 $f(x)$ 的近似值。同时计算 $P_2(x)$ 在 $x = 5$ ， $x = 50$ ， $x = 115$ ， $x = 185$ 处的函数值。

计算结果如下：

```
Lagrange 4.(1)
5.0000    2.2667

50.0000   -20.2333

115.0000  -171.9000

185.0000  -492.7333
```

x	5	50	115	185
$P_2(x)$	2.2667	-20.2333	-171.9000	-492.7333
<i>actual</i>	2.2361	7.0711	10.7238	13.6015

(2) 设 $f(x) = \sqrt{x}$ ，关于以 $x_0 = 36$ ， $x_1 = 49$ ， $x_2 = 64$ 为节点的拉格朗日插值多项式 $P_2(x)$ ，利用拉格朗日插值多项式 $P_2(x)$ 作为 $f(x)$ 的近似值。同时计算 $P_2(x)$ 在 $x = 5$ ， $x = 50$ ， $x = 115$ ， $x = 185$ 处的函数值。

计算结果如下：

```
Lagrange 4.(2)
5.0000    3.1158

50.0000    7.0718

115.0000   10.1670

185.0000   10.0388
```

x	5	50	115	185
$P_2(x)$	3.1158	7.0718	10.1670	10.0388
<i>actual</i>	2.2361	7.0711	10.7238	13.6015

(3) 设 $f(x) = \sqrt{x}$ ，关于以 $x_0 = 100$ ， $x_1 = 121$ ， $x_2 = 144$ 为节点的拉格朗日插值多项式 $P_2(x)$ ，利用拉格朗日插值多项式 $P_2(x)$ 作为 $f(x)$ 的近似值。同时计算 $P_2(x)$ 在 $x = 5$ ， $x = 50$ ， $x = 115$ ， $x = 185$ 处的函数值。

计算结果如下：

```

Lagrange 4.(3)
5.0000    4.4391

50.0000   7.2850

115.0000  10.7228

185.0000  13.5357

```

x	5	50	115	185
$P_2(x)$	4.4391	7.2850	10.7228	13.5357
<i>actual</i>	2.2361	7.0711	10.7238	13.6015

(4) 设 $f(x) = \sqrt{x}$ ，关于以 $x_0 = 169$ ， $x_1 = 196$ ， $x_2 = 225$ 为节点的拉格朗日插值多项式 $P_2(x)$ ，利用拉格朗日插值多项式 $P_2(x)$ 作为 $f(x)$ 的近似值。同时计算 $P_2(x)$ 在 $x = 5$ ， $x = 50$ ， $x = 115$ ， $x = 185$ 处的函数值。

计算结果如下：

```

Lagrange 4.(4)
5.0000    5.4972

50.0000   7.8001

115.0000  10.8005

185.0000  13.6006

```

x	5	50	115	185
$P_2(x)$	5.4972	7.8001	10.8005	13.6006
<i>actual</i>	2.2361	7.0711	10.7238	13.6015

根据上述四个拉格朗日插值问题的计算结果与真实函数值的误差，内插时插值与真实函数值误差小，超出内插范围的插值与真实函数值误差大，并且距离插值区间越远误差越大。因此，内插比外推更可靠。

思考题：

1. 根据问题 2，可以使用缩小插值区间的方法来解决问题 1 中函数 $f(x) = \frac{1}{1+x^2}$ 的拉格朗日插值多项式 $P_n(x)$ 随插值次数 n 的增大在区间边缘振荡剧烈的现象。此外，既然问题是插值点处于区间边缘，我们可以尝试改变插值区间来使原本处于区间边缘的插值点位于区间恰当位置而不会产生振荡。

2. 在较大的插值区间里，相较于较小的插值区间，每个等分数据点之间的变化更大。因此插值区间越小，函数摆动越小，误差越小。

4. 内插是在现有观测值之间进行估计，外推是对超出已知观测范围进行预测。在插值问题中，内插就是对在给出的数据点之间的插值点进行函数值近似，而外推就是对不在给出的数据点之间的插值点进行函数值近似。

实验报告二 牛顿 (Newton) 迭代法

第一部分：问题分析 *(描述并总结出实验题目)*

利用牛顿迭代法求非线性方程 $f(x) = 0$ 的根.

输入：初值 α ，精度 $\varepsilon_1, \varepsilon_2$ ，最大迭代次数 N

输出：方程 $f(x) = 0$ 的根 x^* 的近似值或计算失败标志

目的：利用牛顿迭代法求非线性方程 $f(x) = 0$ 的根.

意义：

1. 学习根据实际问题建立数学模型、针对数学模型的特点确定适当的计算方法、编制出计算机能够执行的计算程序、输入计算机、进行调试、完成运算等数值计算的过程。

2. 不能只会套用教科书中的标准程序进行数值计算，要独立地将学过的数值算法编制成计算机程序，灵活应用已经掌握的算法求解综合性较大的课题。

3. 加深对数值计算程序结构化思想的印象，提高编程能力，加深对“计算方法”课程内容的理解和掌握，进一步奠定从事数值计算工作的基础。

4. 可以利用所掌握的“高级语言”顺利地编制出计算机程序，上机实习，完成实验环节的教学要求。不简单地套用现成的标准程序完成实验题目，把重点放在对算法的理解、程序的优化设计、上机调试和计算结果分析上，达到实验课的目的。

第二部分：数学原理

迭代法的定义：

将非线性方程 $f(x) = 0$ 转化成等价方程 $x = \varphi(x)$ ；选定一个根的初始近似值 x_0 ，利用递推关系式

$$x_{k+1} = \varphi(x_k), \quad k = 0, 1, \dots$$

产生序列 $\{x_k\}$ 。如果序列收敛于 α ，则 α 满足方程 $x = \varphi(x)$ ，即为方程 $f(x) = 0$ 的根（也称为不动点）。

牛顿迭代法：

求非线性方程 $f(x) = 0$ 的根 x^* ，由牛顿迭代法计算公式

$$x_0 = \alpha$$
$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, \dots$$

牛顿迭代法局部收敛的条件及收敛阶：

一般地，牛顿迭代法具有局部收敛性，为保证迭代收敛，要求对充分小的 $\delta > 0$ ， $\alpha \in O(x^*, \delta)$ 。

如果 $f(x) \in C^2[a, b]$ ， $f(x^*) = 0$ ， $f'(x^*) \neq 0$ ，那么，对充分小的 $\delta > 0$ ，当 $\alpha \in O(x^*, \delta)$ 时，由牛顿迭代法计算出的 $\{x_k\}$ 收敛于 x^* ，且收敛速度是 2 阶的；

如果 $f(x) \in C^m[a, b]$ ， $f(x^*) = f'(x^*) = \dots = f^{(m-1)}(x^*) = 0$ ， $f^{(m)}(x^*) \neq 0$ ($m > 1$)，那么，对充分小的 $\delta > 0$ ，当 $\alpha \in O(x^*, \delta)$ 时，由牛顿迭代法计算出的 $\{x_k\}$ 收敛于 x^* ，且收敛速度是 1 阶的。

第三部分：程序设计流程

程序流程：

1. 置 $n = 1$
2. 当 $n \leq N$ 时，做 2.1—2.4
 - 2.1 置 $F = f(x_0)$, $DF = f'(x_0)$
如果 $|F| < \varepsilon_1$ ，输出 x_0 ；停机
如果 $|DF| < \varepsilon_2$ ，输出失败标志；停机
 - 2.2 置 $x_1 = x_0 - F/DF$
 - 2.3 置 $Tol = |x_1 - x_0|$
如果 $|Tol| < \varepsilon_1$ ，输出 x_1 ；停机
 - 2.4 置 $n = n + 1$, $x_0 = x_1$
3. 输出失败标志
4. 停机

代码实现：

```
1  function result = Newton(x0, e1, e2, N, f)
2  % 利用牛顿迭代法求f(x)=0的根
3  % 输入：初值x0, 精度e1,e2, 最大迭代次数N
4  % 输出：方程f(x)=0根x*的近似值或计算失败标志
5  n = 1;
6  while n <= N
7      F = double(subs(f, symvar(f), x0));
8      DF = double(subs(diff(f), symvar(f), x0));
9      if abs(F) < e1
10         result = x0;
11         return;
12     end
13     if abs(DF) < e2
14         result = 'Iteration failed!';
15         return;
16     end
17     x1 = x0 - F / DF;
18     Tol = abs(x1 - x0);
19     if abs(Tol) < e1
20         result = x1;
21         return;
22     end
23     n = n + 1;
24     x0 = x1;
25 end
26 result = 'Iteration failed!';
27 end
```

第四部分：实验结果、结论与讨论

问题 1: (1) $\cos x - x = 0$, $\varepsilon_1 = 10^{-6}$, $\varepsilon_2 = 10^{-4}$, $N = 10$, $x_0 = \frac{\pi}{4} \approx 0.785398163$

(2) $e^{-x} - \sin x = 0$, $\varepsilon_1 = 10^{-6}$, $\varepsilon_2 = 10^{-4}$, $N = 10$, $x_0 = 0.6$

(1)

Newton 1.(1)

x* =
0.7391

(2)

Newton 1.(2)

x* =
0.5885

问题 2: (1) $x - e^{-x} = 0$, $\varepsilon_1 = 10^{-6}$, $\varepsilon_2 = 10^{-4}$, $N = 10$, $x_0 = 0.5$

(2) $x^2 - 2xe^{-x} + e^{-2x} = 0$, $\varepsilon_1 = 10^{-6}$, $\varepsilon_2 = 10^{-4}$, $N = 20$, $x_0 = 0.5$

(1)

Newton 2.(1)

x* =
0.5671

(2)

Newton 2.(2)

x* =
0.5666

思考题：

1. 能够使用牛顿迭代法求非线性方程 $f(x) = 0$ 的根的条件是迭代收敛. 一般地, 牛顿迭代法具有局部收敛性, 为保证迭代收敛, 要求对充分小的 $\delta > 0$, $\alpha \in O(x^*, \delta)$. 所以在实验 1 中确定初值的原则是尽可能接近方程的根.

在实际计算中, 只需在 $f(x) = 0$ 的根附近任取一点 x_0 作为初始值即可. 当初始值难以确定时, 可以通过如二分法等算法确定根的近似值, 再将这个近似值作为初始值, 以保证牛顿迭代法的收敛性.

2. 问题 2 的 (1) 中, $f(x) = x - e^{-x}$, $f'(x) = 1 + e^{-x}$, $f(x^*) = 0$, $f'(x^*) \neq 0$. 所以牛顿迭代法计算出的 $\{x_k\}$ 收敛于 x^* , 且收敛速度是 2 阶的; 但是, (2) 中 $f(x) = x^2 - 2xe^{-x} + e^{-2x}$, $f'(x) = 2x - 2e^{-x} + 2xe^{-x} - 2e^{-2x}$, $f(x^*) = 0$, $f'(x^*) = 0$. 所以牛顿迭代法计算出的 $\{x_k\}$ 收敛于 x^* , 且收敛速度是 1 阶的.

综上, (1) 中方程的根是单重根, 牛顿迭代法平方收敛, 迭代速度快; (2) 中方程有重根, 牛顿迭代法线性收敛, 迭代速度慢.

实验报告三 高斯 (Gauss) 列主元消元法

第一部分：问题分析 (描述并总结出实验题目)

利用高斯列主元消元法求给定的 n 阶线性方程组 $Ax = b$ 的解或者确定该线性方程组是奇异的.

输入: $n, a_{ij}, b_i, i, j = 1, 2, \dots, n$

输出: 线性方程组 $Ax = b$ 的近似解 $x_i, i = 1, 2, \dots, n$

目的: 利用高斯列主元消元法求解线性方程组 $Ax = b$.

意义:

1. 学习根据实际问题建立数学模型、针对数学模型的特点确定适当的计算方法、编制出计算机能够执行的计算程序、输入计算机、进行调试、完成运算等数值计算的过程。

2. 不能只会套用教科书中的标准程序进行数值计算，要独立地将学过的数值算法编制成计算机程序，灵活应用已经掌握的算法求解综合性较大的课题。

3. 加深对数值计算程序结构化思想的印象，提高编程能力，加深对“计算方法”课程内容的理解和掌握，进一步奠定从事数值计算工作的基础。

4. 可以利用所掌握的“高级语言”顺利地编制出计算机程序，上机实习，完成实验环节的教学要求。不简单地套用现成的标准程序完成实验题目，把重点放在对算法的理解、程序的优化设计、上机调试和计算结果分析上，达到实验课的目的。

第二部分：数学原理

高斯消元法过程：

解 n 阶线性方程组 $Ax = b$ ：

(1) 若 $a_{11} \neq 0$ ，第 i 个方程减去第一个方程乘以 $m_{i1} = \frac{a_{i1}}{a_{11}}$ ，得到同解方程组 ($i = 2, 3, \dots, n$):

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{22}^{(1)}x_2 + \cdots + a_{2n}^{(1)}x_n = b_2^{(1)} \\ \vdots \\ a_{n2}^{(1)}x_2 + \cdots + a_{nn}^{(1)}x_n = b_n^{(1)} \end{array} \right.$$
$$\begin{cases} a_{ij}^{(1)} = a_{ij} - m_{i1}a_{1j}, & i, j = 2, 3, \dots, n \\ b_i^{(1)} = b_i - m_{i1}b_1 \end{cases}$$

(2) 若 $a_{22}^{(1)} \neq 0$ ，则上一步方程组的第 i 个方程减去第二个方程乘以 $m_{i2} = \frac{a_{i2}^{(1)}}{a_{22}^{(1)}}$ ，得到同解方程组 ($i = 3, 4, \dots, n$):

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n = b_1 \\ a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 + \cdots + a_{2n}^{(1)}x_n = b_2^{(1)} \\ a_{33}^{(2)}x_3 + \cdots + a_{3n}^{(2)}x_n = b_3^{(2)} \\ \vdots \\ a_{n3}^{(2)}x_3 + \cdots + a_{nn}^{(2)}x_n = b_n^{(2)} \end{array} \right.$$
$$\begin{cases} a_{ij}^{(2)} = a_{ij}^{(1)} - m_{i2}a_{2j}^{(1)} \\ b_i^{(2)} = b_i^{(1)} - m_{i2}b_2^{(1)} \end{cases}, \quad i, j = 3, 4, \dots, n$$

(3) 依次下去执行这个过程，经过 $n - 1$ 次消元后，得到

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n = b_1 \\ a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 + \cdots + a_{2n}^{(1)}x_n = b_2^{(1)} \\ a_{33}^{(2)}x_3 + \cdots + a_{3n}^{(2)}x_n = b_3^{(2)} \\ \vdots \\ a_{nn}^{(n-1)}x_n = b_n^{(n-1)} \end{array} \right.$$

$$m_{ik} = \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}}, \quad k = 1, 2, \dots, n-1$$

$$\begin{cases} a_{ij}^{(k)} = a_{ij}^{(k-1)} - m_{ik}a_{kj}^{(k-1)} \\ b_i^{(k)} = b_i^{(k-1)} - m_{ik}b_k^{(k-1)} \end{cases}, \quad i, j = k+1, \dots, n$$

(4) 回代

$$\begin{cases} x_n = \frac{b_n^{(n-1)}}{a_{nn}^{(n-1)}} \\ x_i = \left[b_i^{(i-1)} - \sum_{j=i+1}^n a_{ij}^{(i-1)}x_j \right] / a_{ii}^{(i-1)} \end{cases} \quad i = n-1, n-2, \dots, 2, 1$$

高斯列主元素消元法:

在高斯消元法执行过程中, 始终假设 $a_{kk}^{(k-1)} \neq 0$, 称 $a_{kk}^{(k-1)}$ 为主元素. 若 $a_{kk}^{(k-1)} = 0$, 则消元过程无法进行. 实际上, 即使 $a_{kk}^{(k-1)} \neq 0$, 但 $|a_{kk}^{(k-1)}|$ 很小时, 用它作除数对计算结果也是很不利, 称这样的 $a_{kk}^{(k-1)}$ 为小主元素.

若 $|a_{r_k k}^{(k-1)}| = \max_{k \leq i \leq n} |a_{ik}^{(k-1)}|$, 称 $|a_{r_k k}^{(k-1)}|$ 为列主元素, r_k 行为主元素行, 这时可将第 r_k 行与第 k 行进行交换, 使 $|a_{r_k k}^{(k-1)}|$ 位于交换后等价方程组的 $a_{kk}^{(k-1)}$ 位置, 然后再实施消元法, 这种方法称为列主元消元法.

第三部分：程序设计流程

程序流程：

1. 对 $k = 1, 2, \dots, n - 1$ ，做 1.1—1.3，消元过程

1.1 寻找最小的正整数 p , $k \leq p \leq n$ 和 $|a_{pk}| = \max_{k \leq j \leq n} |a_{jk}|$. 如

果 $a_{pk} = 0$ ，输出奇异标志，停机；

1.2 如果 $p \neq k$ ，那么交换 p, k 两行；

1.3 对 $i = k + 1, \dots, n$ ，记 $m_{ik} = a_{ik}/a_{kk}$ ，计算

$$\begin{cases} a_{ij} = a_{ij} - a_{kj}m_{ik} \\ i = k + 1, \dots, n \\ j = k + 1, \dots, n \\ b_i = b_i - b_k m_{ik} \\ i = k + 1, \dots, n \end{cases}$$

2. 如果 $a_{nn} = 0$ 输出奇异标志，停机；

3. 置 $x_n = b_n/a_{nn}$ ，回代过程；

4. 对 $k = n - 1, \dots, 2, 1$ ，置 $x_k = (b_k - \sum_{j=k+1}^n a_{kj}x_j)/a_{kk}$.

代码实现：

```
1 function result = Gauss(n, A, b)
2 % 利用高斯列主元消元法求给定的n阶线性方程组Ax=b的解或者确定该线性方程组是奇异的.
3 % 输入: n, a_ij, b_i, i, j=1,2,...,n
4 % 输出: 线性方程组Ax=b的近似解x_i, i=1,2,...,n
5
6 for k = 1:n-1
7     max = abs(A(k, k));
8     p = k;
9     for j = k+1:n
10         if abs(A(j, k)) > max
11             max = abs(A(j, k));
12             p = j;
13         end
14     end
15     if A(p, k) == 0
16         result = 'Singular matrix!';
17         return ;
18     end
19     if p ~= k
20         A([k p], :) = A([p k], :);
21         b([k p], :) = b([p k], :);
22     end
23     for i = k+1:n
24         m_ik = A(i, k)/A(k, k);
25         for j = k+1:n
26             A(i, j) = A(i, j) - A(k, j) * m_ik;
27         end
28         b(i) = b(i) - b(k) * m_ik;
29     end
30 end
```

```

31 - if A(n, n) == 0
32 -     result = 'Singular matrix!';
33 -     return ;
34 - end
35 - result = zeros(n, 1);
36 - result(n, 1) = b(n)/A(n, n);
37 - for k = n-1:-1:1
38 -     sum = 0;
39 -     for j = k+1:n
40 -         sum = sum + A(k, j) * result(j, 1);
41 -     end
42 -     result(k, 1) = (b(k) - sum)/A(k, k);
43 - end
44 - end

```

第四部分：实验结果、结论与讨论

问题 1 实验题目：

$$(1) \begin{bmatrix} 0.4096 & 0.1234 & 0.3678 & 0.2943 \\ 0.2246 & 0.3872 & 0.4015 & 0.1129 \\ 0.3645 & 0.1920 & 0.3781 & 0.0643 \\ 0.1784 & 0.4002 & 0.2786 & 0.3927 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1.1951 \\ 1.1262 \\ 0.9989 \\ 1.2499 \end{bmatrix}$$

$$(2) \begin{bmatrix} 136.01 & 90.860 & 0 & 0 \\ 90.860 & 98.810 & -67.590 & 0 \\ 0 & -67.590 & 132.01 & 46.260 \\ 0 & 0 & 46.260 & 177.17 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 226.87 \\ 122.08 \\ 110.68 \\ 223.43 \end{bmatrix}$$

$$(3) \begin{bmatrix} 1 & 1/2 & 1/3 & 1/4 \\ 1/2 & 1/3 & 1/4 & 1/5 \\ 1/3 & 1/4 & 1/5 & 1/6 \\ 1/4 & 1/5 & 1/6 & 1/7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 25/12 \\ 77/60 \\ 57/60 \\ 319/420 \end{bmatrix}$$

$$(4) \begin{bmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 32 \\ 23 \\ 33 \\ 31 \end{bmatrix}$$

(1)

```

>> GaussCalc
Gauss 1.(1)
x =
    1.0000
    1.0000
    1.0000
    1.0000

```

(2)

```
>> GaussCalc
Gauss 1.(2)
x =
    1.0000
    1.0000
    1.0000
    1.0000
```

(3)

```
>> GaussCalc
Gauss 1.(3)
x =
    1.0000
    1.0000
    1.0000
    1.0000
```

(4)

```
>> GaussCalc
Gauss 1.(4)
x =
    1
    1
    1
    1
```

问题 2

$$(1) \begin{bmatrix} 197 & 305 & -206 & -804 \\ 46.8 & 71.3 & -47.4 & 52.0 \\ 88.6 & 76.4 & -10.8 & 802 \\ 1.45 & 5.90 & 6.13 & 36.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 136 \\ 11.7 \\ 25.1 \\ 6.60 \end{bmatrix}$$

$$(2) \begin{bmatrix} 0.5398 & 0.7161 & -0.5554 & -0.2982 \\ 0.5257 & 0.6924 & 0.3565 & -0.6255 \\ 0.6465 & -0.8187 & -0.1872 & 0.1291 \\ 0.5814 & 0.9400 & -0.7779 & -0.4042 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0.2058 \\ -0.0503 \\ 0.1070 \\ 0.1859 \end{bmatrix}$$

$$(3) \begin{bmatrix} 10 & 1 & 2 \\ 1 & 10 & 2 \\ 1 & 1 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 13 \\ 13 \\ 7 \end{bmatrix}$$

$$(4) \begin{bmatrix} 4 & -2 & -4 \\ -2 & 17 & 10 \\ -4 & 10 & 9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -2 \\ 25 \\ 15 \end{bmatrix}$$

(1)

```
>> GaussCalc
Gauss 2.(1)
x =
    0.9537
    0.3210
    1.0787
   -0.0901
```

(2)

```
>> GaussCalc
Gauss 2.(2)
x =
    0.5162
    0.4152
    0.1100
    1.0365
```

(3)

```
>> GaussCalc
Gauss 2.(3)
x =
    1.0000
    1.0000
    1.0000
```

(4)

```
>> GaussCalc
Gauss 2.(4)
x =
    1
    1
    1
```

实验报告四 龙贝格 (Romberg) 积分法

第一部分：问题分析 (描述并总结出实验题目)

利用龙贝格(Romberg)积分法计算积分 $\int_a^b f(x)dx$.

输入: a, b, ε, f

输出: 龙贝格 T -数表

- 二分次数和精度的关系如何?

目的: 利用龙贝格(Romberg)积分法计算积分 $\int_a^b f(x)dx$.

意义:

1. 学习根据实际问题建立数学模型、针对数学模型的特点确定适当的计算方法、编制出计算机能够执行的计算程序、输入计算机、进行调试、完成运算等数值计算的过程。

2. 不能只会套用教科书中的标准程序进行数值计算, 要独立地将学过的数值算法编制成计算机程序, 灵活应用已经掌握的算法求解综合性较大的课题。

3. 加深对数值计算程序结构化思想的印象, 提高编程能力, 加深对“计算方法”课程内容的理解和掌握, 进一步奠定从事数值计算工作的基础。

4. 可以利用所掌握的“高级语言”顺利地编制出计算机程序, 上机实习, 完成实验环节的教学要求。不简单地套用现成的标准程序完成实验题目, 把重点放在对算法的理解、程序的优化设计、上机调试和计算结果分析上, 达到实验课的目的。

第二部分：数学原理

利用复化梯形求积公式、复化辛普生求积公式、复化柯特斯求积公式的误差估计式计算积分 $\int_a^b f(x)dx$. 记 $h = \frac{b-a}{n}$, $x_k = a + k \cdot h$, $k = 0, 1, \dots, n$, 其计算公式:

$$\begin{aligned}T_n &= \frac{1}{2}h \sum_{k=1}^n [f(x_{k-1}) + f(x_k)] \\T_{2n} &= \frac{1}{2}T_n + \frac{1}{2}h \sum_{k=1}^n f(x_k - \frac{1}{2}h) \\S_n &= \frac{1}{3}(4T_{2n} - T_n) \\C_n &= \frac{1}{15}(16S_{2n} - S_n) \\R_n &= \frac{1}{63}(64C_{2n} - C_n)\end{aligned}$$

一般地, 利用龙贝格算法计算积分, 要输出所谓的 T -数表

$$\begin{array}{ccccccc}T_1 & & & & & & \\T_2 & S_1 & & & & & \\T_4 & S_2 & C_1 & & & & \\T_8 & S_4 & C_2 & R_1 & & & \\\vdots & \vdots & \vdots & \vdots & \ddots & & \end{array}$$

第三部分：程序设计流程

程序设计：

1. 置 $h = b - a, i = 1$

$$T_{0,0} = \frac{1}{2}h[f(a) + f(b)]$$

2. 输出 T_1

3. 做 3.1—3.3

3.1 置 $ii = 2^{i-1}$

$$\text{置 } T_{0,i} = \frac{1}{2}T_{0,i-1} + \frac{1}{2}h \sum_{k=1}^{ii} f(a + (k - \frac{1}{2})h)$$

计算出 $T_{0,i}$

3.2 置 $m = 1, 2, \dots, i, k = i - m, T_{m,k} = \frac{4^m T_{m-1,k+1} - T_{m-1,k}}{4^m - 1}$

计算出 $T_{m,k}$

3.3 根据 3.2 计算出 $T_{i,0}$ ，若 $|T_{i,0} - T_{i-1,0}| < \varepsilon$ ，则停机，把 $T_{i,0}$ 作为定积分的近似值，输出 T -数表；否则转去继续执行步骤 3.4

3.4 置 $h = \frac{h}{2}, i = i + 1$

4. 停机

代码实现：

```
1  function result = Romberg(a, b, e, f)
2  % 利用龙贝格(Romberg)积分法计算f(x)在区间[a,b]上的定积分.
3  % 输入: a, b, e, f
4  % 输出: 龙贝格T-数表
5  h = b - a;
6  i = 2;
7  T = zeros(20, 20);
8  T(1, 1) = h * (subs(f, symvar(f), a) + subs(f, symvar(f), b)) / 2;
9  fprintf('%f\n', T(1, 1));
```

```

10 - while i > 0
11 -     ii = 2^(i-2);
12 -     sum = 0;
13 -     for k = 1:ii
14 -         sum = sum + subs(f, symvar(f), a + (k - 0.5) * h);
15 -     end
16 -     T(1, i) = T(1, i-1) / 2 + h * sum / 2;
17 -     fprintf('%f ', T(1, i));
18 -     for m = 2:i
19 -         k = i - m + 1;
20 -         T(m, k) = (4^(m-1) * T(m-1, k+1) - T(m-1, k)) / (4^(m-1) - 1);
21 -         fprintf('%f ', T(m, k));
22 -     end
23 -     fprintf('\n');
24 -     if abs(T(i, 1) - T(i-1, 1)) < e
25 -         result = T(i, 1);
26 -         return;
27 -     end
28 -     h = h / 2;
29 -     i = i + 1;
30 - end

```

第四部分：实验结果、结论与讨论

问题 1： 利用龙贝格(Romberg)积分法计算积分

(1) $\int_0^1 x^2 e^x dx, \quad \varepsilon = 10^{-6}$

(2) $\int_1^3 e^x \sin x dx, \quad \varepsilon = 10^{-6}$

(3) $\int_0^1 \frac{4}{1+x^2} dx, \quad \varepsilon = 10^{-6}$

(4) $\int_0^1 \frac{1}{x+1} dx, \quad \varepsilon = 10^{-6}$

(1)

```

>> RombergCalc
Romberg 1.(1)
1.359141
0.885661 0.727834
0.760596 0.718908 0.718313
0.728890 0.718321 0.718282 0.718282
0.720936 0.718284 0.718282 0.718282 0.718282
I(f) = 0.718282

```

1.359141

0.885661	0.727834
----------	----------

0.760596	0.718908	0.718313
----------	----------	----------

0.728890	0.718321	0.718282	0.718282
----------	----------	----------	----------

0.720936	0.718284	0.718282	0.718282	0.718282
----------	----------	----------	----------	----------

$$\int_0^1 x^2 e^x dx \approx 0.718282$$

(2)

```
>> RombergCalc
Romberg 1.(2)
5.121826
9.279763 10.665742
10.520554 10.934151 10.952045
10.842043 10.949207 10.950210 10.950181
10.923094 10.950111 10.950171 10.950170 10.950170
10.943398 10.950167 10.950170 10.950170 10.950170 10.950170
I(f) = 10.950170
```

5.121826

9.279763	10.665742
----------	-----------

10.520554	10.934151	10.952045
-----------	-----------	-----------

10.842043	10.949207	10.950210	10.950181
-----------	-----------	-----------	-----------

10.923094	10.950111	10.950171	10.950170	10.950170
-----------	-----------	-----------	-----------	-----------

10.943398	10.950167	10.950170	10.950170	10.950170	10.950170
-----------	-----------	-----------	-----------	-----------	-----------

$$\int_1^3 e^x \sin x dx \approx 10.950170$$

(3)

```
>> RombergCalc
Romberg 1.(3)
3.000000
3.100000 3.133333
3.131176 3.141569 3.142118
3.138988 3.141593 3.141594 3.141586
3.140942 3.141593 3.141593 3.141593 3.141593
3.141430 3.141593 3.141593 3.141593 3.141593 3.141593
I(f) = 3.141593
```

3. 000000

3. 100000 3. 133333

3. 131176 3. 141569 3. 142118

3. 138988 3. 141593 3. 141594 3. 141586

3. 140942 3. 141593 3. 141593 3. 141593 3. 141593

3. 141430 3. 141593 3. 141593 3. 141593 3. 141593 3. 141593

$$\int_0^1 \frac{4}{1+x^2} dx \approx 3.141593$$

(4)

```
>> RombergCalc
Romberg 1.(4)
0.750000
0.708333 0.694444
0.697024 0.693254 0.693175
0.694122 0.693155 0.693148 0.693147
0.693391 0.693148 0.693147 0.693147 0.693147
I(f) = 0.693147
```

0. 750000

0. 708333 0. 694444

0. 697024 0. 693254 0. 693175

0. 694122 0. 693155 0. 693148 0. 693147

0. 693391 0. 693148 0. 693147 0. 693147 0. 693147

$$\int_0^1 \frac{1}{x+1} dx \approx 0.693147$$

思考题：

在实验 1 中二分次数和精度的关系如何？

二分次数越多，精度越高。

实验报告五 四阶龙格—库塔（Runge—Kutta）方法

第一部分：问题分析 （描述并总结出实验题目）

利用四阶龙格—库塔（Runge—Kutta）方法求解微分方程的初值问题.

输入： a, b, α, N

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

目的：利用四阶龙格—库塔（Runge—Kutta）方法求解微分方程的初值问题.

意义：

1. 学习根据实际问题建立数学模型、针对数学模型的特点确定适当的计算方法、编制出计算机能够执行的计算程序、输入计算机、进行调试、完成运算等数值计算的过程。

2. 不能只会套用教科书中的标准程序进行数值计算，要独立地将学过的数值算法编制成计算机程序，灵活应用已经掌握的算法求解综合性较大的课题。

3. 加深对数值计算程序结构化思想的印象，提高编程能力，加深对“计算方法”课程内容的理解和掌握，进一步奠定从事数值计算工作的基础。

4. 可以利用所掌握的“高级语言”顺利地编制出计算机程序，上机实习，完成实验环节的教学要求。不简单地套用现成的标准程序完成实验题目，把重点放在对算法的理解、程序的优化设计、上机调试和计算结果分析上，达到实验课的目的。

第二部分：数学原理

给定常微分方程初值问题

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

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

$$K_1 = hf(x_n, y_n)$$

$$K_2 = hf\left(x_n + \frac{h}{2}, y_n + \frac{K_1}{2}\right)$$

$$K_3 = hf\left(x_n + \frac{h}{2}, y_n + \frac{K_2}{2}\right)$$

$$K_4 = hf(x_n + h, y_n + K_3)$$

$$y_{n+1} = y_n + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4), \quad n = 0, 1, \dots, N-1$$

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

第三部分 程序设计流程

程序设计：

1. 置 $x_0 = a, y_0 = \alpha, h = \frac{b-a}{N}$

2. 对 $n = 1, 2, \dots, N$, 做 2.1—2.4

2.1 置

$$K_1 = hf(x_0, y_0)$$

$$K_2 = hf\left(x_0 + \frac{h}{2}, y_0 + \frac{K_1}{2}\right)$$

$$K_3 = hf(x_0 + \frac{h}{2}, y_0 + \frac{K_2}{2})$$

$$K_4 = hf(x_0 + h, y_0 + K_3)$$

2.2 置

$$x_1 = x_0 + h$$

$$y_1 = y_0 + \frac{1}{6}(K_1 + 2K_2 + 3K_3 + K_4)$$

2.3 输出 x_1, y_1

2.4 置 $x_0 = x_1, y_0 = y_1$

3. 停机

代码实现：

```

1  function result = Runge_Kutta(a, b, alpha, N, f)
2  % 利用四阶龙格-库塔 (Runge-Kutta) 方法求解微分方程的初值问题
3  % 输入：迭代区间[a,b]，初始值alpha，数值点个数N，函数f(x, y)
4  % 输出：初值问题的数值解x_n, y_n, n=0,1,2,...,N
5  x0 = a;
6  y0 = alpha;
7  h = (b - a) / N;
8  X = zeros(N, 1);
9  Y = zeros(N, 1);
10 for n = 1:N
11     K1 = h * f(x0, y0);
12     K2 = h * f(x0 + h/2, y0 + K1/2);
13     K3 = h * f(x0 + h/2, y0 + K2/2);
14     K4 = h * f(x0 + h, y0 + K3);
15     X(n) = x0 + h;
16     Y(n) = y0 + (K1 + 2 * K2 + 2 * K3 + K4) / 6;
17     x0 = X(n);
18     y0 = Y(n);
19 end
20 result = [X, Y];
21 end

```

第四部分：实验结果、结论与讨论

问题 1

(1)

$$\frac{dy}{dx} = x + y, 0 \leq x \leq 1, N = 5, 10, 20$$

$$y(0) = -1 \quad h = \frac{b-a}{N}$$


准确解

$$y = -x - 1$$

```
>> Runge_KuttaCalc
Runge Kutta 1.(1)
N = 5
    0.2000    -1.2000
    0.4000    -1.4000
    0.6000    -1.6000
    0.8000    -1.8000
    1.0000    -2.0000

N = 10
    0.1000    -1.1000
    0.2000    -1.2000
    0.3000    -1.3000
    0.4000    -1.4000
    0.5000    -1.5000
    0.6000    -1.6000
    0.7000    -1.7000
    0.8000    -1.8000
    0.9000    -1.9000
    1.0000    -2.0000

N = 20
    0.0500    -1.0500
    0.1000    -1.1000
    0.1500    -1.1500
    0.2000    -1.2000
```



0.2500	-1.2500
0.3000	-1.3000
0.3500	-1.3500
0.4000	-1.4000
0.4500	-1.4500
0.5000	-1.5000
0.5500	-1.5500
0.6000	-1.6000
0.6500	-1.6500
0.7000	-1.7000
0.7500	-1.7500
0.8000	-1.8000
0.8500	-1.8500
0.9000	-1.9000
0.9500	-1.9500
1.0000	-2.0000

N = 5

n	x_n	y_n	$y(x_n)$	E
1	0.2000	-1.2000	-1.2000	0.0000
2	0.4000	-1.4000	-1.4000	0.0000
3	0.6000	-1.6000	-1.6000	0.0000
4	0.8000	-1.8000	-1.8000	0.0000
5	1.0000	-2.0000	-2.0000	0.0000

N = 10

n	x_n	y_n	$y(x_n)$	E
1	0.1000	-1.1000	-1.1000	0.0000
2	0.2000	-1.2000	-1.2000	0.0000
3	0.3000	-1.3000	-1.3000	0.0000

<i>4</i>	0. 4000	−1. 4000	−1. 4000	0. 0000
<i>5</i>	0. 5000	−1. 5000	−1. 5000	0. 0000
<i>6</i>	0. 6000	−1. 6000	−1. 6000	0. 0000
<i>7</i>	0. 7000	−1. 7000	−1. 7000	0. 0000
<i>8</i>	0. 8000	−1. 8000	−1. 8000	0. 0000
<i>9</i>	0. 9000	−1. 9000	−1. 9000	0. 0000
<i>10</i>	1. 0000	−2. 0000	−2. 0000	0. 0000

N = 20

<i>n</i>	x_n	y_n	$y(x_n)$	E
<i>1</i>	0. 0500	−1. 0500	−1. 0500	0. 0000
<i>2</i>	0. 1000	−1. 1000	−1. 1000	0. 0000
<i>3</i>	0. 1500	−1. 1500	−1. 1500	0. 0000
<i>4</i>	0. 2000	−1. 2000	−1. 2000	0. 0000
<i>5</i>	0. 2500	−1. 2500	−1. 2500	0. 0000
<i>6</i>	0. 3000	−1. 3000	−1. 3000	0. 0000
<i>7</i>	0. 3500	−1. 3500	−1. 3500	0. 0000
<i>8</i>	0. 4000	−1. 4000	−1. 4000	0. 0000
<i>9</i>	0. 4500	−1. 4500	−1. 4500	0. 0000
<i>10</i>	0. 5000	−1. 5000	−1. 5000	0. 0000
<i>11</i>	0. 5500	−1. 5500	−1. 5500	0. 0000
<i>12</i>	0. 6000	−1. 6000	−1. 6000	0. 0000

13	0.6500	-1.6500	-1.6500	0.0000
14	0.7000	-1.7000	-1.7000	0.0000
15	0.7500	-1.7500	-1.7500	0.0000
16	0.8000	-1.8000	-1.8000	0.0000
17	0.8500	-1.8500	-1.8500	0.0000
18	0.9000	-1.9000	-1.9000	0.0000
19	0.9500	-1.9500	-1.9500	0.0000
20	1.0000	-2.0000	-2.0000	0.0000

(2)

$$\frac{dy}{dx} = -y^2, 0 \leq x \leq 1, N = 5, 10, 20$$

$$y(0) = 1 \quad h = \frac{b-a}{N}$$

准确解

$$y = \frac{1}{x+1}$$

```
>> Runge_KuttaCalc
Runge Kutta 1.(2)
N = 5
    0.2000    0.8333
    0.4000    0.7143
    0.6000    0.6250
    0.8000    0.5556
    1.0000    0.5000

N = 10
    0.1000    0.9091
    0.2000    0.8333
    0.3000    0.7692
```

0.4000	0.7143
0.5000	0.6667
0.6000	0.6250
0.7000	0.5882
0.8000	0.5556
0.9000	0.5263
1.0000	0.5000

N = 20

0.0500	0.9524
0.1000	0.9091
0.1500	0.8696
0.2000	0.8333
0.2500	0.8000
0.3000	0.7692
0.3500	0.7407
0.4000	0.7143
0.4500	0.6897
0.5000	0.6667
0.5500	0.6452
0.6000	0.6250
0.6500	0.6061
0.7000	0.5882
0.7500	0.5714
0.8000	0.5556
0.8500	0.5405
0.9000	0.5263
0.9500	0.5128
1.0000	0.5000

N = 5

n	x_n	y_n	$y(x_n)$	E
1	0.2000	0.8333	0.8333	0.0000
2	0.4000	0.7143	0.7143	0.0000
3	0.6000	0.6250	0.6250	0.0000
4	0.8000	0.5556	0.5556	0.0000
5	1.0000	0.5000	0.5000	0.0000

N = 10

n	x_n	y_n	$y(x_n)$	E
1	0.1000	0.9091	0.9091	0.0000
2	0.2000	0.8333	0.8333	0.0000
3	0.3000	0.7692	0.7692	0.0000
4	0.4000	0.7143	0.7143	0.0000
5	0.5000	0.6667	0.6667	0.0000
6	0.6000	0.6250	0.6250	0.0000
7	0.7000	0.5882	0.5882	0.0000
8	0.8000	0.5556	0.5556	0.0000
9	0.9000	0.5263	0.5263	0.0000
10	1.0000	0.5000	0.5000	0.0000

N = 20

n	x_n	y_n	$y(x_n)$	E
1	0.0500	0.9524	0.9524	0.0000
2	0.1000	0.9091	0.9091	0.0000
3	0.1500	0.8696	0.8696	0.0000
4	0.2000	0.8333	0.8333	0.0000
5	0.2500	0.8000	0.8000	0.0000
6	0.3000	0.7692	0.7692	0.0000
7	0.3500	0.7407	0.7407	0.0000

8	0.4000	0.7143	0.7143	0.0000
9	0.4500	0.6897	0.6897	0.0000
10	0.5000	0.6667	0.6667	0.0000
11	0.5500	0.6452	0.6452	0.0000
12	0.6000	0.6250	0.6250	0.0000
13	0.6500	0.6061	0.6061	0.0000
14	0.7000	0.5882	0.5882	0.0000
15	0.7500	0.5714	0.5714	0.0000
16	0.8000	0.5556	0.5556	0.0000
17	0.8500	0.5405	0.5405	0.0000
18	0.9000	0.5263	0.5263	0.0000
19	0.9500	0.5128	0.5128	0.0000
20	1.0000	0.5000	0.5000	0.0000

问题 2

(1)

$$\frac{dy}{dx} = \frac{2}{x}y + x^2 e^x, 1 \leq x \leq 3, N = 5, 10, 20$$

$$y(1) = 0 \quad h = \frac{b-a}{N}$$

准确解

$$y = x^2 (e^x - e)$$


```

>> Runge_KuttaCalc
Runge Kutta 2.(1)
N = 5
    1.4000    2.6139
    1.8000   10.7763
    2.2000   30.4917
    2.6000   72.5856
    3.0000  156.2252

N = 10
    1.2000    0.8664
    1.4000    2.6197
    1.6000    5.7199
    1.8000   10.7920
    2.0000   18.6809
    2.2000   30.5216
    2.4000   47.8324
    2.6000   72.6345
    2.8000  107.6089
    3.0000  156.2983

N = 20
    1.1000    0.3459
    1.2000    0.8666
    1.3000    1.6072
    1.4000    2.6203
    1.5000    3.9676
    1.6000    5.7209
    1.7000    7.9638
    1.8000   10.7935
    1.9000   14.3229
    2.0000   18.6829
    2.1000   24.0250
    2.2000   30.5244
    2.3000   38.3835
    2.4000   47.8359
    2.5000   59.1510
    2.6000   72.6389
    2.7000   88.6566
    2.8000  107.6143
    2.9000  129.9833
    3.0000  156.3048

```

N = 5

n	x_n	y_n	$y(x_n)$	E
1	1.4000	2.6139	2.6204	0.0064
2	1.8000	10.7763	10.7936	0.0173
3	2.2000	30.4917	30.5246	0.0329
4	2.6000	72.5856	72.6393	0.0537
5	3.0000	156.2252	156.3053	0.0801

N = 10

n	x_n	y_n	$y(x_n)$	E
1	1.2000	0.8664	0.8666	0.0003
2	1.4000	2.6197	2.6204	0.0006
3	1.6000	5.7199	5.7210	0.0011
4	1.8000	10.7920	10.7936	0.0016
5	2.0000	18.6809	18.6831	0.0022
6	2.2000	30.5216	30.5246	0.0030
7	2.4000	47.8324	47.8362	0.0038
8	2.6000	72.6345	72.6393	0.0048
9	2.8000	107.6089	107.6147	0.0058
10	3.0000	156.2983	156.3053	0.0070

N = 20

n	x_n	y_n	$y(x_n)$	E
1	1.1000	0.3459	0.3459	0.0000
2	1.2000	0.8666	0.8666	0.0000
3	1.3000	1.6072	1.6072	0.0000
4	1.4000	2.6203	2.6204	0.0000
5	1.5000	3.9676	3.9677	0.0001
6	1.6000	5.7209	5.7210	0.0001
7	1.7000	7.9638	7.9639	0.0001
8	1.8000	10.7935	10.7936	0.0001
9	1.9000	14.3229	14.3231	0.0001
10	2.0000	18.6829	18.6831	0.0002
11	2.1000	24.0250	24.0252	0.0002
12	2.2000	30.5244	30.5246	0.0002
13	2.3000	38.3835	38.3837	0.0003
14	2.4000	47.8359	47.8362	0.0003
15	2.5000	59.1510	59.1513	0.0003
16	2.6000	72.6389	72.6393	0.0004
17	2.7000	88.6566	88.6570	0.0004
18	2.8000	107.6143	107.6147	0.0004
19	2.9000	129.9833	129.9838	0.0005
20	3.0000	156.3048	156.3053	0.0005

(2)

$$\frac{dy}{dx} = \frac{1}{x}(y^2 + y), 1 \leq x \leq 3, N = 5, 10, 20$$

$$y(1) = -2 \quad h = \frac{b-a}{N}$$

准确解

$$y = \frac{2x}{1-2x}$$

```
>> Runge_KuttaCalc
```

```
Runge Kutta 2.(2)
```

```
N = 5
```

1.4000	-1.5540
1.8000	-1.3836
2.2000	-1.2934
2.6000	-1.2375
3.0000	-1.1995

```
N = 10
```

1.2000	-1.7142
1.4000	-1.5555
1.6000	-1.4545
1.8000	-1.3846
2.0000	-1.3333
2.2000	-1.2941
2.4000	-1.2631
2.6000	-1.2381
2.8000	-1.2174
3.0000	-1.2000

```
N = 20
```

1.1000	-1.8333
1.2000	-1.7143
1.3000	-1.6250
1.4000	-1.5556
1.5000	-1.5000
1.6000	-1.4545
1.7000	-1.4167
1.8000	-1.3846
1.9000	-1.3571

2.0000	-1.3333
2.1000	-1.3125
2.2000	-1.2941
2.3000	-1.2778
2.4000	-1.2632
2.5000	-1.2500
2.6000	-1.2381
2.7000	-1.2273
2.8000	-1.2174
2.9000	-1.2083
3.0000	-1.2000

N = 5

n	x_n	y_n	$y(x_n)$	E
1	1.4000	-1.5540	-1.5556	0.0016
2	1.8000	-1.3836	-1.3846	0.0010
3	2.2000	-1.2934	-1.2941	0.0007
4	2.6000	-1.2375	-1.2381	0.0006
5	3.0000	-1.1995	-1.2000	0.0005

N = 10

n	x_n	y_n	$y(x_n)$	E
1	1.2000	-1.7142	-1.7143	0.0000
2	1.4000	-1.5555	-1.5556	0.0000
3	1.6000	-1.4545	-1.4545	0.0000
4	1.8000	-1.3846	-1.3846	0.0000
5	2.0000	-1.3333	-1.3333	0.0000
6	2.2000	-1.2941	-1.2941	0.0000

7	2. 4000	-1. 2631	-1. 2632	0. 0000
8	2. 6000	-1. 2381	-1. 2381	0. 0000
9	2. 8000	-1. 2174	-1. 2174	0. 0000
10	3. 0000	-1. 2000	-1. 2000	0. 0000
N = 20				
n	x_n	y_n	$y(x_n)$	E
1	1. 1000	-1. 8333	-1. 8333	0. 0000
2	1. 2000	-1. 7143	-1. 7143	0. 0000
3	1. 3000	-1. 6250	-1. 6250	0. 0000
4	1. 4000	-1. 5556	-1. 5556	0. 0000
5	1. 5000	-1. 5000	-1. 5000	0. 0000
6	1. 6000	-1. 4545	-1. 4545	0. 0000
7	1. 7000	-1. 4167	-1. 4167	0. 0000
8	1. 8000	-1. 3846	-1. 3846	0. 0000
9	1. 9000	-1. 3571	-1. 3571	0. 0000
10	2. 0000	-1. 3333	-1. 3333	0. 0000
11	2. 1000	-1. 3125	-1. 3125	0. 0000
12	2. 2000	-1. 2941	-1. 2941	0. 0000
13	2. 3000	-1. 2778	-1. 2778	0. 0000
14	2. 4000	-1. 2632	-1. 2632	0. 0000
15	2. 5000	-1. 2500	-1. 2500	0. 0000

16	2.6000	-1.2381	-1.2381	0.0000
17	2.7000	-1.2273	-1.2273	0.0000
18	2.8000	-1.2174	-1.2174	0.0000
19	2.9000	-1.2083	-1.2083	0.0000
20	3.0000	-1.2000	-1.2000	0.0000

问题 3

(1)

$$\frac{dy}{dx} = -20(y - x^2) + 2x, 0 \leq x \leq 1, N = 5, 10, 20$$

$$y(0) = \frac{1}{3} \quad h = \frac{b-a}{N}$$

准确解

$$y = x^2 + \frac{1}{3}e^{-20x}$$

```
>> Runge_KuttaCalc
Runge Kutta 3.(1)
N = 5
1.0e+03 *

0.0002    0.0018
0.0004    0.0088
0.0006    0.0437
0.0008    0.2173
0.0010    1.0843

N = 10
0.1000    0.1228
0.2000    0.0793
0.3000    0.1048
0.4000    0.1666
0.5000    0.2539
```

0.6000	0.3630
0.7000	0.4927
0.8000	0.6426
0.9000	0.8125
1.0000	1.0025

N = 20

0.0500	0.1276
0.1000	0.0569
0.1500	0.0402
0.2000	0.0467
0.2500	0.0651
0.3000	0.0910
0.3500	0.1229
0.4000	0.1602
0.4500	0.2026
0.5000	0.2501
0.5500	0.3026
0.6000	0.3601
0.6500	0.4226
0.7000	0.4901
0.7500	0.5626
0.8000	0.6401
0.8500	0.7226
0.9000	0.8101
0.9500	0.9026
1.0000	1.0001

N = 5

n	x_n	y_n	$y(x_n)$	E
1	0.2000	1.7600	0.0461	1.7139
2	0.4000	8.8133	0.1601	8.6532
3	0.6000	43.6800	0.3600	43.3200
4	0.8000	217.2933	0.6400	216.6533
5	1	1084.3200	1.0000	1083.3200

N = 10

n	x_n	y_n	$y(x_n)$	E
1	0.1000	0.1228	0.0551	0.0677
2	0.2000	0.0793	0.0461	0.0332
3	0.3000	0.1048	0.0908	0.0139
4	0.4000	0.1666	0.1601	0.0065
5	0.5000	0.2539	0.2500	0.0038
6	0.6000	0.3630	0.3600	0.0030
7	0.7000	0.4927	0.4900	0.0027
8	0.8000	0.6426	0.6400	0.0026
9	0.9000	0.8125	0.8100	0.0025
10	1.0000	1.0025	1.0000	0.0025

N = 20

n	x_n	y_n	$y(x_n)$	E
1	0.0500	0.1276	0.1251	0.0024
2	0.1000	0.0569	0.0551	0.0018
3	0.1500	0.0402	0.0391	0.0011
4	0.2000	0.0467	0.0461	0.0006
5	0.2500	0.0651	0.0647	0.0003
6	0.3000	0.0910	0.0908	0.0002
7	0.3500	0.1229	0.1228	0.0001
8	0.4000	0.1602	0.1601	0.0001

9	0.4500	0.2026	0.2025	0.0001
10	0.5000	0.2501	0.2500	0.0001
11	0.5500	0.3026	0.3025	0.0001
12	0.6000	0.3601	0.3600	0.0001
13	0.6500	0.4226	0.4225	0.0001
14	0.7000	0.4901	0.4900	0.0001
15	0.7500	0.5626	0.5625	0.0001
16	0.8000	0.6401	0.6400	0.0001
17	0.8500	0.7226	0.7225	0.0001
18	0.9000	0.8101	0.8100	0.0001
19	0.9500	0.9026	0.9025	0.0001
20	1.0000	1.0001	1.0000	0.0001

(2)

$$\frac{dy}{dx} = -20y + 20\sin x + \cos x, \quad 0 \leq x \leq 1 \quad N=5, 10, 20$$

$$y(0)=1 \quad h = \frac{b-a}{N}$$

准确解

$$y = e^{-20x} + \sin x$$

```

>> Runge_KuttaCalc
Runge Kutta 3.(2)
N = 5
1.0e+03 *

0.0002    0.0052
0.0004    0.0254
0.0006    0.1255
0.0008    0.6253
0.0010    3.1238

N = 10
0.1000    0.4331
0.2000    0.3097
0.3000    0.3323
0.4000    0.4014
0.5000    0.4831
0.6000    0.5654
0.7000    0.6440
0.8000    0.7167
0.9000    0.7825
1.0000    0.8405

N = 20
0.0500    0.4250
0.1000    0.2405
0.1500    0.2022
0.2000    0.2184
0.2500    0.2548
0.3000    0.2983
0.3500    0.3439
0.4000    0.3898
0.4500    0.4351
0.5000    0.4795
0.5500    0.5227
0.6000    0.5646
0.6500    0.6052
0.7000    0.6442
0.7500    0.6816
0.8000    0.7173
0.8500    0.7513
0.9000    0.7833
0.9500    0.8134
1.0000    0.8414

```

N = 5

n	x_n	y_n	$y(x_n)$	E
1	0.2000	5.1973	0.2170	4.9804
2	0.4000	25.3762	0.3898	24.9864
3	0.6000	125.4868	0.5646	124.9222
4	0.8000	625.3121	0.7174	624.5947
5	1	3123.7952	0.8415	3122.9537

N = 10

n	x_n	y_n	$y(x_n)$	E
1	0.1000	0.4331	0.2352	0.1980
2	0.2000	0.3097	0.2170	0.0927
3	0.3000	0.3323	0.2980	0.0343
4	0.4000	0.4014	0.3898	0.0117
5	0.5000	0.4831	0.4795	0.0036
6	0.6000	0.5654	0.5646	0.0008
7	0.7000	0.6440	0.6442	0.0002
8	0.8000	0.7167	0.7174	0.0006
9	0.9000	0.7825	0.7833	0.0008
10	1.0000	0.8405	0.8415	0.0009

N = 20

n	x_n	y_n	$y(x_n)$	E
1	0.0500	0.4250	0.4179	0.0071
2	0.1000	0.2405	0.2352	0.0053
3	0.1500	0.2022	0.1992	0.0029
4	0.2000	0.2184	0.2170	0.0015
5	0.2500	0.2548	0.2541	0.0007
6	0.3000	0.2983	0.2980	0.0003
7	0.3500	0.3439	0.3438	0.0001
8	0.4000	0.3898	0.3898	0.0000
9	0.4500	0.4351	0.4351	0.0000
10	0.5000	0.4795	0.4795	0.0000
11	0.5500	0.5227	0.5227	0.0000
12	0.6000	0.5646	0.5646	0.0000
13	0.6500	0.6052	0.6052	0.0000
14	0.7000	0.6442	0.6442	0.0000
15	0.7500	0.6816	0.6816	0.0000
16	0.8000	0.7173	0.7174	0.0000
17	0.8500	0.7513	0.7513	0.0000
18	0.9000	0.7833	0.7833	0.0000
19	0.9500	0.8134	0.8134	0.0000
20	1.0000	0.8414	0.8415	0.0000

(3)

$$\frac{dy}{dx} = -2.0(y - e^x \sin x) + e^x(\sin x + \cos x), 0 \leq x \leq 1$$

$$y(0) = 0 \quad N = 5, 10, 20, h = \frac{b-a}{N}$$

准确解

$$y = e^x \sin x$$

```
>> Runge_KuttaCalc
```

```
Runge Kutta 3.(3)
```

```
N = 5
```

0.2000	0.2986
0.4000	0.9272
0.6000	2.8355
0.8000	10.7109
1.0000	47.9414

```
N = 10
```

0.1000	0.1121
0.2000	0.2451
0.3000	0.4018
0.4000	0.5841
0.5000	0.7938
0.6000	1.0324
0.7000	1.3010
0.8000	1.6003
0.9000	1.9305
1.0000	2.2912

```
N = 20
```

0.0500	0.0526
0.1000	0.1104
0.1500	0.1737
0.2000	0.2427
0.2500	0.3178
0.3000	0.3990
0.3500	0.4867
0.4000	0.5811

0.4500	0.6823
0.5000	0.7906
0.5500	0.9061
0.6000	1.0290
0.6500	1.1594
0.7000	1.2974
0.7500	1.4432
0.8000	1.5966
0.8500	1.7579
0.9000	1.9268
0.9500	2.1034
1.0000	2.2875

N = 5

n	x_n	y_n	$y(x_n)$	E
1	0.2000	0.2986	0.2427	0.0560
2	0.4000	0.9272	0.5809	0.3463
3	0.6000	2.8355	1.0288	1.8066
4	0.8000	10.7109	1.5965	9.1144
5	1	47.9414	2.2874	45.6541

N = 10

n	x_n	y_n	$y(x_n)$	E
1	0.1000	0.1121	0.1103	0.0017
2	0.2000	0.2451	0.2427	0.0025
3	0.3000	0.4018	0.3989	0.0029
4	0.4000	0.5841	0.5809	0.0032
5	0.5000	0.7938	0.7904	0.0034
6	0.6000	1.0324	1.0288	0.0036

<i>7</i>	0. 7000	1. 3010	1. 2973	0. 0037
<i>8</i>	0. 8000	1. 6003	1. 5965	0. 0038
<i>9</i>	0. 9000	1. 9305	1. 9267	0. 0038
<i>10</i>	1. 0000	2. 2912	2. 2874	0. 0038
N = 20				
<i>n</i>	<i>x_n</i>	<i>y_n</i>	<i>y(x_n)</i>	<i>E</i>
<i>1</i>	0. 0500	0. 0526	0. 0525	0. 0001
<i>2</i>	0. 1000	0. 1104	0. 1103	0. 0001
<i>3</i>	0. 1500	0. 1737	0. 1736	0. 0001
<i>4</i>	0. 2000	0. 2427	0. 2427	0. 0001
<i>5</i>	0. 2500	0. 3178	0. 3177	0. 0001
<i>6</i>	0. 3000	0. 3990	0. 3989	0. 0001
<i>7</i>	0. 3500	0. 4867	0. 4866	0. 0001
<i>8</i>	0. 4000	0. 5811	0. 5809	0. 0001
<i>9</i>	0. 4500	0. 6823	0. 6822	0. 0001
<i>10</i>	0. 5000	0. 7906	0. 7904	0. 0001
<i>11</i>	0. 5500	0. 9061	0. 9059	0. 0001
<i>12</i>	0. 6000	1. 0290	1. 0288	0. 0001
<i>13</i>	0. 6500	1. 1594	1. 1593	0. 0001
<i>14</i>	0. 7000	1. 2974	1. 2973	0. 0001
<i>15</i>	0. 7500	1. 4432	1. 4430	0. 0001

16	0.8000	1.5966	1.5965	0.0001
17	0.8500	1.7579	1.7577	0.0001
18	0.9000	1.9268	1.9267	0.0001
19	0.9500	2.1034	2.1033	0.0001
20	1.0000	2.2875	2.2874	0.0001

思考题:

1. 实验 1 中 (1) 数值解和解析解相同, (2) 数值解和解析解略有不同. 因为四阶龙格库塔方法是以小段的线性算法来近似获得微分方程的数值解, (1) 的准确解是 1 阶的, (2) 的准确解是无限阶的. 但是, 由于 (2) 的准确解的二阶导数比较小, 即准确解在迭代区间上变化不剧烈, 用小段线性算法求得的数值解非常接近解析解. 因此, 对于 (1) 数值解和解析解相同, 对于 (2) 数值解和解析解略有不同.

2. 实验 2 中 N 越大越精确. 因为四阶龙格库塔方法是用 x_n 加上步长 $h = \frac{b-a}{N}$ 乘以估算斜率来计算 x_{n+1} , 是一种小段的线性算法; 当 N 越大时, 步长越小, 计算区间 $[x_n, x_{n+1}]$ 越小, 而函数在足够小的区间 $[x_n, x_{n+1}]$ 内可以近似看作线性的, 所以用这种线性算法在 N 越大时结果越精确.

3. 实验 3 中 N 较小会出现误差很大的现象. 因为四阶龙格库塔方法本质上是小段的线性算法, 实验 3 中的准确解都是高阶的; 当

N 较小时，区间 $[x_n, x_{n+1}]$ 较大，而当准确解在区间 $[x_n, x_{n+1}]$ 内变化剧烈时，四阶龙格库塔方法使用线性近似无法得到精确的数值解，导致误差很大.