长春理工大学
Changchun University of Science and Technology

实　验　报　告

实验地点		学生姓名	WJT
实验日期	2021 年 11 月 28 日　第 7、8 节	学院	数学与统计学院
实验课程	数值逼近	学号	
实验项目	数值积分	成绩	

一、实验目的或要求

编写梯形公式、Simposon 公式、复化梯形公式、复化 Simposon 公式

创新点

二、实验过程记录

（零）实验所用积分

被积函数：$f(x) = \frac{1}{1+x^2}$

求积区间：$[-4, 4]$

```
def f(x):
    y = 1/(1+x**2)
    return y
```

（一）编写梯形公式

1、实验代码

1 梯形公式

$$I_1(f) = \frac{1}{2}(b-a)(f(a)+f(b))$$

```
def trapezoid(fun,a,b):
    return 1/2*(b-a)*(fun(a)+fun(b))
```

2、实验测试

```
#test
out = trapezoid(f,-4,4)
print("梯形公式所计算出来的值为：%.6f"%out)
```

梯形公式所计算出来的值为：0.470588

即利用梯形公式在$[-4,4]$对函数 $f(x)=\frac{1}{1+x^2}$ 的数值积分值为：　0.470588

（二）编写 Simposon 公式

1、实验代码

2 Simpson公式

$$I_2(f) = \frac{1}{3}h(f(a) + 4f(b) + f(c))$$

```
def simposon(fun,a,b):
    h = (b-a)/2
    c = (a+b)/2
    return 1/3*h*(fun(a)+4*fun(c)+fun(b))
```

2、实验测试

```
#test
out = simposon(f,-4,4)
print("Simposon公式所计算出来的值为: %.6f"%out)
```

```
Simposon公式所计算出来的值为: 5.490196
```

即利用 Simposon 公式在 $[-4,4]$ 对函数 $f(x)=\frac{1}{1+x^2}$ 的数值积分值为： 5.490196

（三）编写 Newton-Cotes 公式

此公式包含了上述两个公式，输入插值函数的次数 n 即可在被积区间了对函数进行 n 次 lagrange 插值，进而对插值函数多项式积分，求出积分值，即为 n 次 Newton-Cotes 数值积分值。

1、实验代码

①构建 Lagrange 函数来生成插值多项式

```
# Lagrange插值
def lagrange(X,Y):
    """
    输入: 插值点
    输出: 插值函数表达式
    """
    x = sy.symbols('x')
    if len(X) != len(Y):
        raise ValueError("输入的插值节点X变量与Y变量长度不对应! ")
    if type(x)==int:
        x = [x]
    Y = np.array(Y)
    n = len(X)
    # 定义l保存l_1,l_2,...l_n
    y=0
    for i in range(n):
        l=1
        for ii in range(n):
            if i != ii:
                l = l*(x-X[ii])/(X[i]-X[ii])
        y = y+Y[i]*l
        y = sy.simplify(y)
    return y
```

②构造 Newton-Cotes 公式

```python
def NC(fun,a,b,n):
    """
    Newton-Cotes公式
    fun：被积函数
    a,b：上下限
    n：插值函数的次数

    return：被积函数值
    """
    X = np.linspace(a,b,n+1)
    Y = fun(X)
    pn = lagrange(X,Y)
    pn_int = sy.Integral(pn,x)
    out = pn_int.subs(x,b)-pn_int.subs(x,a)
    out = sy.simplify(out)
    out = float(out)
    return out
```

2、代码测试

①2 次 Newton-Cotes 公式即为 Simposon 公式：

```python
## test
nc2 = NC(f,-4,4,2)
sim = simposon(f,-4,4)
```

```python
print("2次Newton-Cotes公式:{}".format(nc2))
print("Simpson公式:          {}".format(sim))
```

```
2次Newton-Cotes公式:5.490196078431373
Simpson公式:          5.490196078431372
```

②对积分函数利用 n 次 Newton-Cotes 计算积分值，n=1,2,…,10

```python
## test2
import scipy.integrate as si
real,_ = si.quad(f,-4,4)
n=10
print("积分函数的真实值为：{}".format(real))
for i in range(1,n+1):
    nc = NC(f,-4,4,i)
    print("{}次Newton-Cotes公式积分值:{}".format(i,nc))
```

```
积分函数的真实值为：2.6516353273360647
1次Newton-Cotes公式积分值:0.47058823529411764
2次Newton-Cotes公式积分值:5.490196078431373
3次Newton-Cotes公式积分值:2.277647058823529
4次Newton-Cotes公式积分值:2.277647058823529
5次Newton-Cotes公式积分值:2.372229249615856
6次Newton-Cotes公式积分值:3.328798127470156
7次Newton-Cotes公式积分值:2.7997007824976405
8次Newton-Cotes公式积分值:1.9410943043884252
9次Newton-Cotes公式积分值:2.430841156646758
10次Newton-Cotes公式积分值:3.595560400191437
```

可以发现，随着次数的增加，Newton-Cotes 公式并没有向着精确值收敛，分析原因是由

于等距节点下的插值函数在次数较高时出现龙格现象。解决的方案有两个，一个是用正交函数系根节点，如切比雪夫节点代替等距节点进行插值。二是利用分段插值解决。

（四）基于切比雪夫节点的 Newton-Cotes 公式

1、实验代码

①切比雪夫节点产生器

3*为解决龙格现象，利用切比雪夫节点的n阶Newton-Cotes公式

```
def chebP(a,b,N):
    """
    切比雪夫节点产生器
    a,b:取值范围
    N: 个数
    """
    t =  np.linspace(0,np.pi,N)
    x = (b-a)/2*(np.cos(t)+1)+a
    return x
```

②基于切比雪夫节点的 Newton-Cotes 公式

```
def NC_cheb(fun,a,b,n):
    """
    Newton-Cotes公式
    fun: 被积函数
    a,b: 上下限
    n: 插值函数的次数

    return: 被积函数值
    """
    X = chebP(a,b,n+1)
    Y = fun(X)
    pn = lagrange(X,Y)
    pn_int = sy.Integral(pn,x)
    out = pn_int.subs(x,b)-pn_int.subs(x,a)
    out = sy.simplify(out)
    out = float(out)
    return out
```

2、代码测试

```
## test
import scipy.integrate as si
real,_ = si.quad(f,-4,4)
n=10
print("积分函数的真实值为: {}".format(real))
for i in range(1,n+1):
    nc = NC_cheb(f,-4,4,i)
    print("{}次切比雪夫节点的Newton-Cotes公式积分值:{}".format(i,nc))
```

```
积分函数的真实值为: 2.6516353273360647
1次切比雪夫节点的Newton-Cotes公式积分值:0.47058823529411764
2次切比雪夫节点的Newton-Cotes公式积分值:5.490196078431373
3次切比雪夫节点的Newton-Cotes公式积分值:1.4745098039215692
4次切比雪夫节点的Newton-Cotes公式积分值:3.705446623093682
5次切比雪夫节点的Newton-Cotes公式积分值:2.166869506423258
6次切比雪夫节点的Newton-Cotes公式积分值:2.9837017884076733
7次切比雪夫节点的Newton-Cotes公式积分值:2.4602070662519226
8次切比雪夫节点的Newton-Cotes公式积分值:2.7757812119556835
9次切比雪夫节点的Newton-Cotes公式积分值:2.579664747157829
10次切比雪夫节点的Newton-Cotes公式积分值:2.6958773072327338
```

可以看到，由于解决了龙格现象，随着公式次数的增加，积分值向着真实值收敛。
（五）复化梯形公式
使数值积分更加精确的第二种方式是利用分段插值。
1、实验代码

4 复化梯形公式

$$I_n(f) = h(\frac{f_0}{2} + f_1 + f_2 + \cdots + f_{n-1} + \frac{f_n}{2})$$

```
def re_trapezoid(fun,a,b,n):
    h = (b-a)/n
    x = np.linspace(a,b,n)
    y = fun(x)
    return h*(sum(y)-1/2*y[0]-1/2*y[-1])
```

2、测试结果

```
re_trapezoid(f,-4,4,1000)
```

```
2.648983396442303
```

对区间进行 1000 次分段，得到结果与真实值差距不大。
（五）复化 Simposon 公式
1、实验代码

5 复化Simposon公式

$$I_n(f) = \sum_{j=1}^{m} \frac{h}{3}(f_{2j-2} + 4f_{2j-1} + f_{2j})$$

```
def re_simposon(fun,a,b,n):
    if n%2 !=0:
        print("n必须为偶数")
        return 0
    h = (b-a)/n
    m = int(n/2)
    x = np.linspace(a,b,n+1)
    y = fun(x)
    I = 0
    for j in range(m):
        j = j+1
        I=I+1/3*h*(y[2*j-2]+4*y[2*j-1]+y[2*j])
    return I
```

2、测试结果

```
re_simposon(f,-4,4,1000)
```

```
2.6516353273352773
```

对区间进行 1000 次分段，得到结果与真实值的误差小于 1e-11。

四、实验结果报告及总结 1、随着次数的增加，Newton-Cotes 公式并没有向着精确值收敛，分析原因是由于等距节点下的插值函数在次数较高时出现龙格现象。解决的方案有两个，一个是用正交函数系根节点，如切比雪夫节点代替等距节点进行插值。二是利用分段插值解决。 2、利用正交函数系根节点插值随着次数的增加，时间复杂度迅速提升，计算缓慢且精度不高 3、利用分段插值，对于复化 Simposon 公式，对区间进行 1000 分段，以误差为 1e-11 逼近真实值，且运算迅速，时间复杂度低。
实验结果反思及讨论：
教师对报告的最终评价和意见: 年　月　日