HW10

白博臣、何骐多、夏营

图 1: 白博臣

图 2: 何骐多

图 3: 夏营

1 Problem1

1.1 a

直接使用 Romberg 积分，第一个积分被积函数 $\sqrt{1-x^2}$，上下限 -1, 1。用这个作为参数，调用 `IntegrationPrecise()`。

第二个积分被积函数 $\sin^2 x$，上下限 0，π。用这个作为参数，调用 `IntegrationPrecise()`。

1.2 b

在对第一个积分进行数值积分时，$\theta = \theta_0$时，分母会变为0，为避免出现 `Infinity`，将分母添加一个小量，避免除以0，再进行积分。

1.3 c

对原式进行如下变形

$$
\begin{aligned}
I &= \int_0^{\infty} \frac{dx}{(1+x)\sqrt{x}} \\
&= \int_0^1 \frac{dx}{(1+x)\sqrt{x}} + \int_1^{\infty} \frac{dx}{(1+x)\sqrt{x}} \\
&= \int_0^1 \frac{dx}{(1+x)\sqrt{x}} + \int_1^0 \frac{d\left(\frac{1}{x}\right)}{\left(1+\frac{1}{x}\right)\sqrt{\frac{1}{x}}} \\
&= 2\int_0^1 \frac{dx}{(1+x)\sqrt{x}}
\end{aligned}
$$

使用 `Java` 代码实现，见附件 `IntegrationDiscreteCal1.java`，运行结果如下

```
Integration 1: 1.5707489780150121
Integration 2: 1.5707963289188798
```

图 4: a 题结果

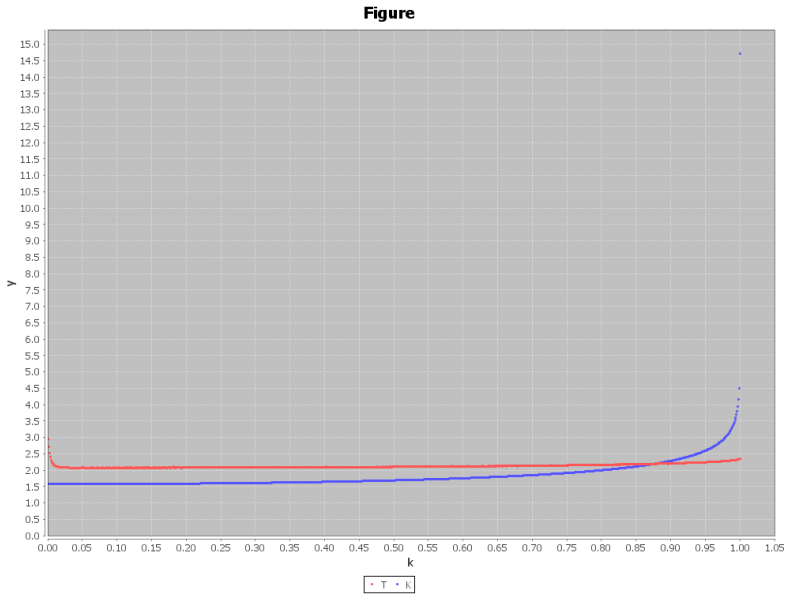

图 5: b 题结果

```
Integration = 3.1376927506055474
```

图 6: c 题结果

2 Problem2

对于此问题，所求积分为无穷积分，为使用数值积分的方法对其进行处理，这里需要采取变量代换的方式。为方便代换，不妨考虑将高斯积分置于极坐标系下进行变换，即对积分式：

$$I^2 = \int_{-\infty}^{\infty} e^{-x^2} dx \int_{-\infty}^{\infty}$$

进行变换。

有极坐标变换：

$$r^2 = x^2 + y^2, \quad dxdy = rdrd\theta$$

进而有：

$$I^2 = \int_0^{2\pi} d\theta \int_0^{\infty} re^{-r^2} dr$$

从而可以将 I^2 分解成两个定积分的乘积：

$$I^2 = I_1 \cdots I_2, \quad I_1 = \int_0^{2\pi} d\theta, \quad I_2 = \int_0^{\infty} r r^{-r^2} dr$$

可以发现对于 I_2 ，其仍是一个无穷积分，因此需要进行进一步的变量代换，令：

$$t = e^{-r^2}, \quad dt = de^{-r^2} = -2re^{-r^2} dr$$

得到：

$$I_2 = \int_0^1 \frac{1}{2} dt$$

由此 I_1 与 I_2 均化作了可以使用数值积分计算的形式。
这里使用上次作业所实现的 Simpson’s Rule ，选取 49 个点进行计算。
计算结果如下：

```
计算所得积分值为： 1.7724538509055159
误差为： 0.0
```

图 7: 运行结果

3 Problem3

使用 Java 代码实现，见附件 `IntegrationDiscreteCal2.java`，运行结果如下

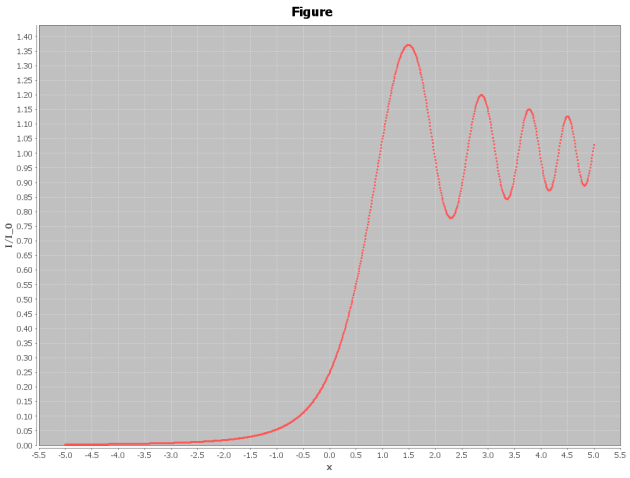

图 8: 运行结果

4 Problem4

4.1 a

由题干可知，$q(u)$ 为光的传输函数，其对应的是光在离中心轴的距离为 u 的地方的透过率，而唯有当光正好通过光栅狭缝时，$q(u)$ 有最大值。因此光栅狭缝之间的间隔，即为 $q(u)$ 最大值之间的间隔。

$$q(u) = \sin^2 \alpha u$$

所以有间隔：

$$T = \frac{\pi}{\alpha}$$

进而有：

$$d = \frac{\pi}{\alpha}$$

4.2 b

Python 代码如下：

```
1 import numpy as np
2 def q(u):
3 d=0.00002
4 alpha=np.pi/d
5 return np.sin(alpha*u)**2
6
```

4.3 c

对于此题，先将题干中的所有数据列出如下：

光栅狭缝间隙：$d = 0.00002m$

光栅总长度：$w = d \times 10 = 0.0002m$

屏幕宽度：$l = 0.1m$

焦距：$f = 1m$

入射光波长：$\lambda = 0.0000005m$

利用上一问中的 $q(u)$ 的代码，再依次将数据代入被积函数中，使用 Simpson’s Rule 进行积分即可得出结果。

其中，由光栅方程：

$$d \sin \theta = k\lambda$$

其中k为主级大的阶数，可以给出一阶主级大所在位置为：

$$x = \pm 0.025m$$

为此，可以以此为判断根据来选择Simpson’s Rule所用的样本点数。在进行了多次试探以后，我们发现当样本点数为49个时，即可得到十分相近的光强分布图，所以我们可以认为此时的积分结果是准确的。仿真结果如下：

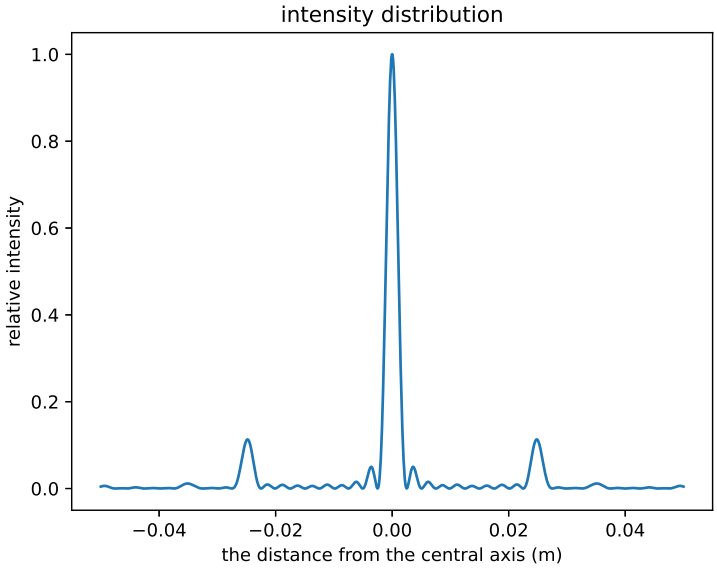

图 9: 仿真结果

4.4 d

为实现可视化，我们使用灰度图的方式来模拟屏幕上条纹的分布，其中用到了 `python matplotlib.pyplot` 库中的 `imshow` 函数。

实现结果如下

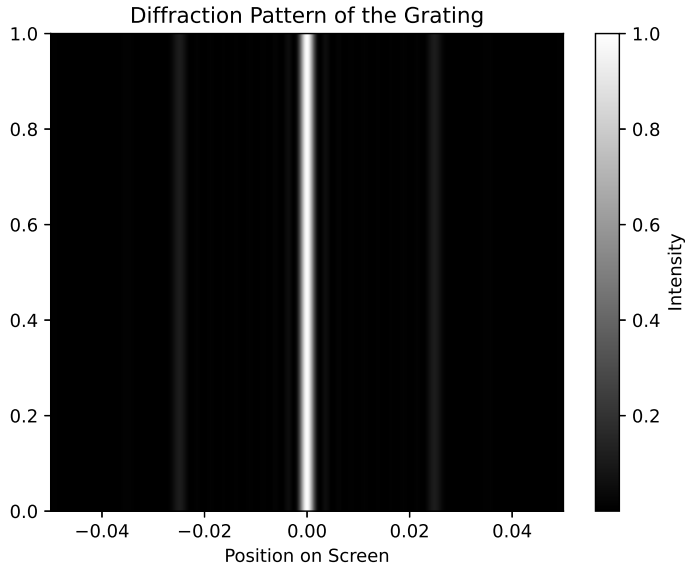

图 10: 仿真结果

4.5 e

4.5.1 i

对前面的仿真代码中修改 $q(u)$ 函数即可，即此时有：

$$q(u) = \sin^2 \alpha u \sin^2 \beta u$$

其中，$\beta = \frac{\alpha}{2}$。

实现结果如下;

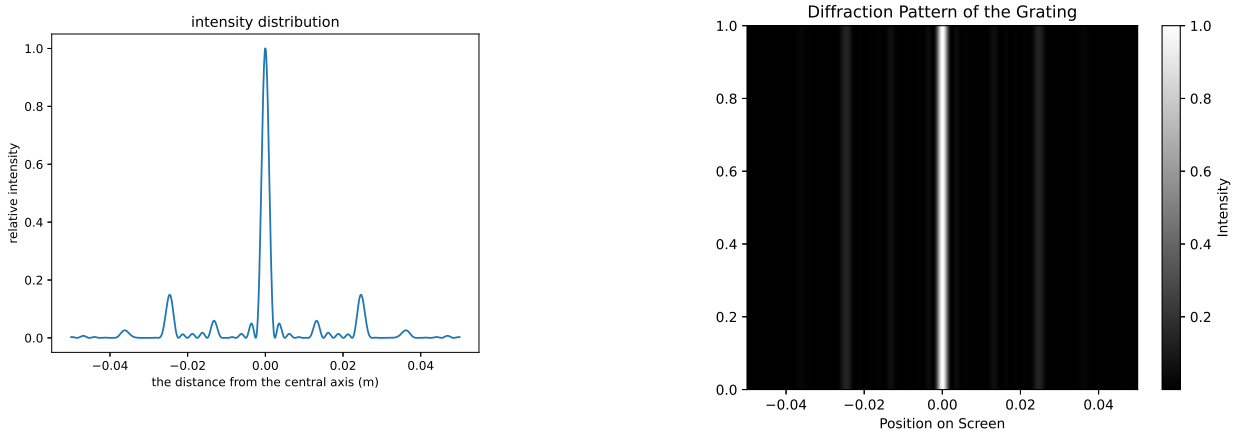

图 11: 仿真结果

4.5.2 ii

此时的 $q(u)$ 为分段函数，即：

$$q(u) = \begin{cases} 1, & u \leq -0.000025 \, or \, u \geq 0.000035 \\ 0, & others \end{cases}$$

同时,光栅的总宽度也发生了变化，即：

$$w = 0.00009m$$

实现结果如下：

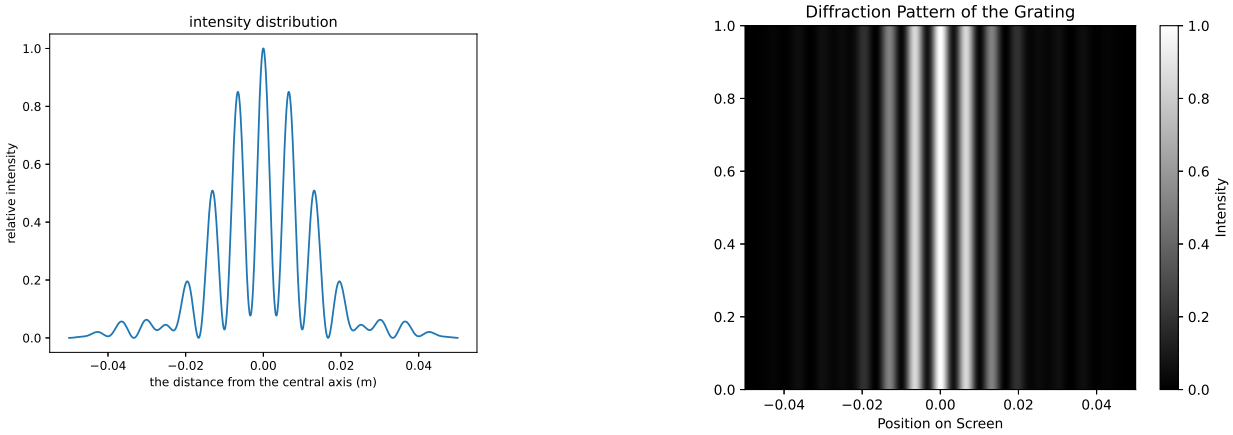

图 12: 仿真结果

5 Problem 5

5.1 问题回顾

利用传统的积分方法计算多维球的体积，并将计算结果作图分析。

5.2 问题解答

我们参考了以下n维球体积计算公式：

$$V_n = \frac{\pi^{\frac{n}{2}} R^n}{\Gamma(\frac{n}{2}+1)} = C_n R^n$$

$$C_n = \frac{\pi^k}{k!}(ForEvenn = 2k)$$

$$C_n = C_{2k+1} = \frac{2^{2k+1} k! \pi^k}{(2k+1)!}(Foroddn = 2k+1)$$

5.2.1 坐标与数组

x维坐标数组，$x\,[k]$表示k维坐标，A为一维的权重数组，对应计算积分时取得权重值。

5.2.2 宏定义和全局变量

使用 **#define** 定义了圆周率 π。**n** 表示每个维度中用于积分的点数。**N** 是多维空间的维度数。**sum** 是用于累积积分结果的变量。**step** 表示每个维度的步长。**start** 和 **end** 是用来测量程序执行时间的变量。

5.2.3 阶乘和双阶乘函数

factorial 函数计算给定正整数m的阶乘。**double_factorial** 函数计算给定正整数 m的双阶乘，这是m!!的定义，只计算 $m, m-2, m-4, \ldots$ 直到1。

5.2.4 计算多维球体积的真实值函数

n_dimensional 函数根据维度数的奇偶性选择不同的数学公式来计算多维球体积的精确值。不同的维度数需要不同的数学公式来计算其体积。

5.2.5 函数 fx

fx 函数根据传入的点 x判断该点是否位于单位球内部。如果在单位球内部，返回1；否则返回0。这个函数用于积分计算过程中的判断条件。

5.2.6 递归计算积分值的函数 Circulation

Circulation 函数通过递归计算多维空间中球体积的近似值。它接受当前维度的索引k，当前积分点的数组 X，当前权重数组 A，和当前的步长 a。递归地计算每个维度的积分点，累加权重乘积来逼近球体积的积分值。

其核心为递归法，根据积分公式有：

$$\int^{(N)} f(x_1, x_2, \ldots, x_N) dx_1 dx_2 \ldots dx_N = \sum_i A_i \int^{(N-1)} f(x1, \ldots, x_{N-1}, x_{N_i}) dx_1 dx_2 \ldots dx_{N-1}$$

根据这一关系，可以得到Circulation(k,x,A,a)函数递归关系如下：

$$Circulation(k, x, A, a) = \sum_i * Circulation(k-1, x, A, a)(x_k = x_{k_i})$$

$$= \sum_i Circulation(k-1, x, A, A_i * a)(x_k = x_{k_i})$$

在$k < 0$时$sum = sum + a * fx$，最终可以计算出结果sum.

5.2.7 函数 findn

findn 函数根据给定的维度数 N，选择合适的点数 n，即每个维度的分段数。这个选择影响了积分的精度和计算效率。

利用gnuplot作出计算体积图以及误差图：

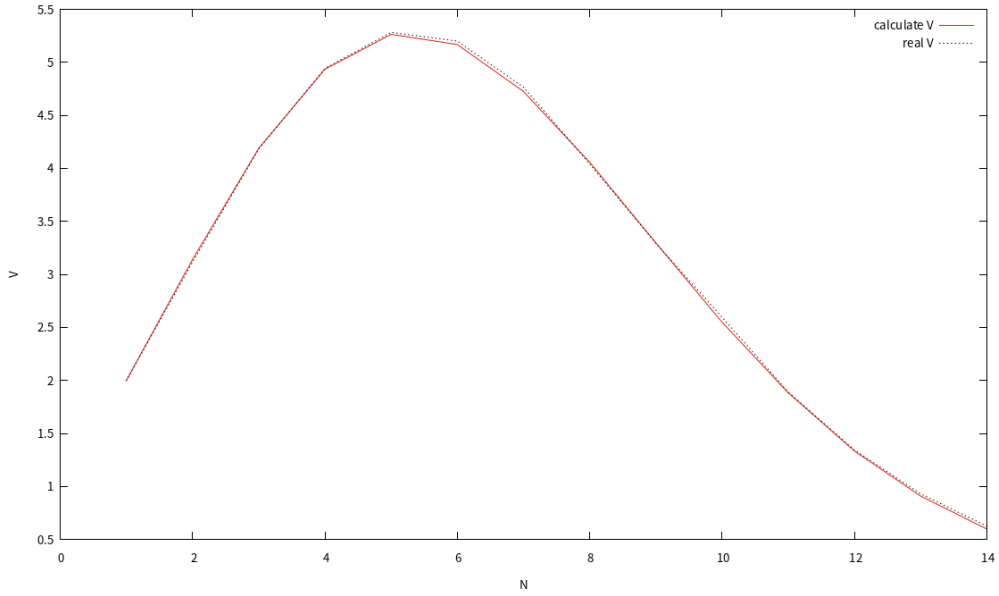

图 13: 计算体积与计算维数关系图

可以看出，误差基本都在5%以内，可见计算结果还是比较准确。为加速计算，在高维减少了取点数，控制误差在5%以内。高维虽然取点较少，但实际上参与计算的积分的坐标点较多，也就时参与计算的体积元较多，所以对结果影响不是非常大，最后在用积分公式计算时就表现出较小的误差。

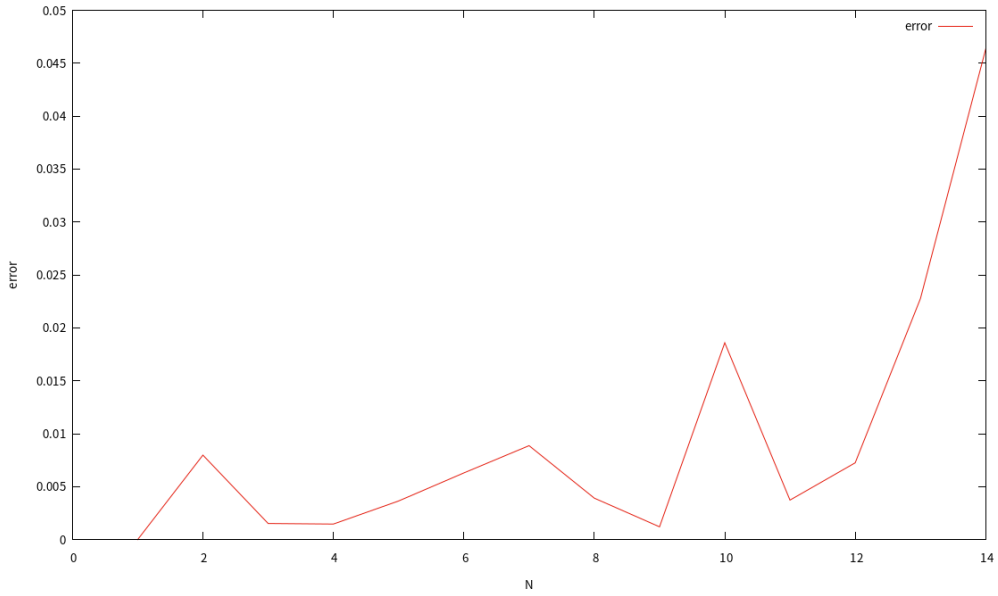

图 14: 计算误差与计算维数关系图