Schrage方法下的16807随机数产生器

许传奇 PB16021546

1 题目

用Schrage方法编写随机数子程序，用连续两个随机数作为点的坐标值绘出若干点的平面分布图。再用$< x^k >$测试均匀性（取不同量级的N值，讨论偏差与N的关系）、C(l)测试其2维独立性（总点数$N > 10^7$）。

2 原理与算法

2.1 原理

2.1.1 Lehmer线性同余法

Lehmer线性同余法是最简单的均匀随机数的产生法，具体步骤如下：

$$I_{n+1} = (aI_n + b)\ mod\ m \tag{1}$$

$$x_n = I_n / m \tag{2}$$

其中，整数$I_1, I_2, ...$的取值是0至$m-1$中的一个值，故实数x_n是严格小于1的，但偶尔会为0。
a是乘子，b是增量，m是模数，如何选择它们决定了产生器的质量。

2.1.2 16807产生器

16807是上面提到的参数选择方案之一，被认为是“最低标准”的产生器，取值如下：

$$a = 7^5 = 16807, b = 0, m = 2^{31} - 1 = 2147483647 \tag{3}$$

即16807随机数产生器为：

$$I_{n+1} = 16807\ I_n\ mod\ 2147483647 \tag{4}$$

$$x_n = I_n / m \tag{5}$$

16807产生器通过了许多理论测试和大量使用的考验，被认为是32位机上最有效的产生器。

2.1.3 Schrage方法

由于32位机的整数区间是$[-2^{31}, 2^{31}-1]$，aI_n在取模之前就可能超过最大整数值，因此需要设计一种取模方法来解决这个问题。本实验中用到的就是Schrage方法，该方法如下：

$$I_{n+1} = \begin{cases} a(I_n \ mod \ q) - r[I_n/q], & if \geqslant 0 \\ a(I_n \ mod \ q) - r[I_n/q] + m, & otherwise \end{cases} \tag{6}$$

其中，a、m分别为随机数产生器的乘子和模数，q、r的取值满足如下关系：

$$m = aq + r, \ q = [m/a], \ r = m \ mod \ a \tag{7}$$

通过Schrage方法，就可以保证随机数产生器取模运算时不会大于最大整数值。

2.1.4 $\langle x^k \rangle$均匀性检验

均匀分布的随机变量的k阶距$\langle x^k \rangle$在理论上满足：

$$\langle x^k \rangle = \int_0^1 x^k dx = \frac{1}{k+1} \tag{8}$$

而16807随机数产生器近似认为是在[0,1]上产生均匀的随机分布。因此，对实际通过16807随机数产生器产生的n个随机数求k阶距，有：

$$\langle x^k \rangle_{16807} = \frac{1}{n} \sum_{i=1}^{n} x_i^k \tag{9}$$

通过比较$\langle x^k \rangle$与$\langle x^k \rangle_{16807}$之间的大小，我们可以检验其独立性。

可以简单地认为$|\langle x^k \rangle - \langle x^k \rangle_{16807}|$越小，均匀性越好。

2.1.5 C(l)独立性检验

讨论随机数序列独立性的一个方法是顺序相关法，它用相邻的两个随机数的自相关函数（或相关系数）来标识随机数序列的独立性情况。

间距为l的自相关函数是：

$$C(l) = \frac{\langle x_n x_{n+l} \rangle - \langle x_n \rangle^2}{\langle x_n^2 \rangle - \langle x_n \rangle^2} \tag{10}$$

一般认为，相关系数越小，独立性越好。

但要注意到，相关函数很小只能说明x_n与x_{n+1}之间的线性关系很弱，不能保证它们之间没有其他的函数关系。

例如，当两个随机数序列x_n与x_{n+1}不相关时，$\langle x_n x_{n+l} \rangle = \langle x_{n+l} \rangle \langle x_n \rangle = \langle x_n \rangle^2$，故相关系数为0。

2.2 算法

本实验的算法较简单，直接按照原理中的具体公式即可。

2.2.1 以连续两个随机数作为点的坐标绘出若干点的平面分布图

设置一个足够长的一维数组用来保存迭代过程中的随机数。在循环中，利用16807随机数产生器的迭代公式与Schrage方法来保存每次迭代的随机数，即：

$$I_{n+1}=\begin{cases}16807\times(I_n\ mod\ 127773)-2836\times[I_n/127773], & if\geqslant 0\\ 16807\times(I_n\ mod\ 127773)-2836\times[I_n/127773]+2147483647, & otherwise\end{cases} \tag{11}$$

$$x_n=I_n/2147483647 \tag{12}$$

一维数组的下标对应第n次生成的随机数，起始下标为0。

将点的横、纵坐标输出并画图。

2.2.2 $\langle x^k\rangle$均匀性检验

第一层循环用来循环随机数num，即循环的每步生成不同个数的随机数；

第二层第一个循环用来生成随机数并保存在一维数组中；第二层第二个循环用来循环k，计算生成的随机数不同的k阶距。最后将打印出不同num与k下的k阶距与理论k阶距的差别，并将数据输出。

2.2.3 C(l)独立性检验

第一层循环用来循环随机数num，即循环的每步生成不同个数的随机数；

第二层第一个循环用来生成随机数并保存在一维数组中；第二层第二个循环用来循环l，计算生成的随机数间距为l的自相关函数C(l)。最后打印出不同num与l下的自相关函数C(l)，并将数据输出。

3 源文件使用说明

编译并运行04Schrage.cpp文件，弹出控制台，如下图所示：

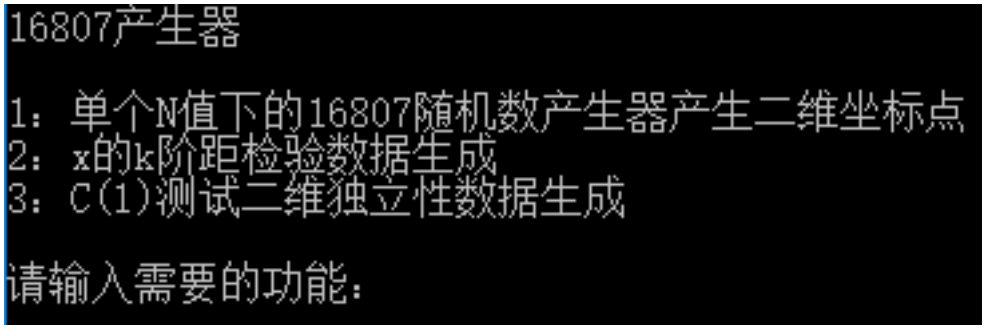


图 1: 程序界面

选择需要的功能：

1. **单个N值下的16807随机数产生器产生二维坐标点**

   该步下，输入需要的点的个数与起始点的数值，程序会自动运行，并将得到的二维坐标点输出到“num=输入的点的个数,start=输入的起始点的数值.txt”文件中。

2. **x的k阶距检验数据生成**

   该步下，输入起始点的数值，程序会从100个随机数开始，到超过210000000个随机数结束生成随机数，每次循环点数是上次的点数的2倍。然后会计算$k = 1, 2, 3, 4, 5$下的理论k阶距与实际k阶距的差值，并输出到控制台和文件“xk test,start=输入的起始点的数值.txt”文件中。

3. **C(l)测试二维独立性数据生成**

   该步下，输入起始点的数值，程序会从100个随机数开始，到超过210000000个随机数结束生成随机数，每次循环点数是上次的点数的2倍。然后会计算$l = 1, 2, 3, ..., 9, 10$下的C(l)，并输出到控制台和文件“C(l)test,start=输入的起始点的数值.txt”文件中。

在源程序中可以修改相应的参数，例如起始、终止随机数的个数等。

打开文件“plot.py”，选择相应的数据文件，可以绘制出二维坐标点的图。

打开文件“xk test.py”，选择相应的数据文件，可以绘制出不同随机数个数与k值对应的$\langle x^k \rangle$折线图。

打开文件“C(l)test.py”，选择相应的数据文件，可以绘制出不同随机数个数与l值对应的C(l)折线图。

4 计算结果及具体分析

以下的讨论中若不做特别说明，都取1作为随机数生成的起始值。

4.1 单个N值下的16807随机数产生器产生二维坐标点

首先设置点的个数num为100000，如图所示，可以看到，生成的点基本上铺满了正方形区域，仅有稍许白色缺陷。

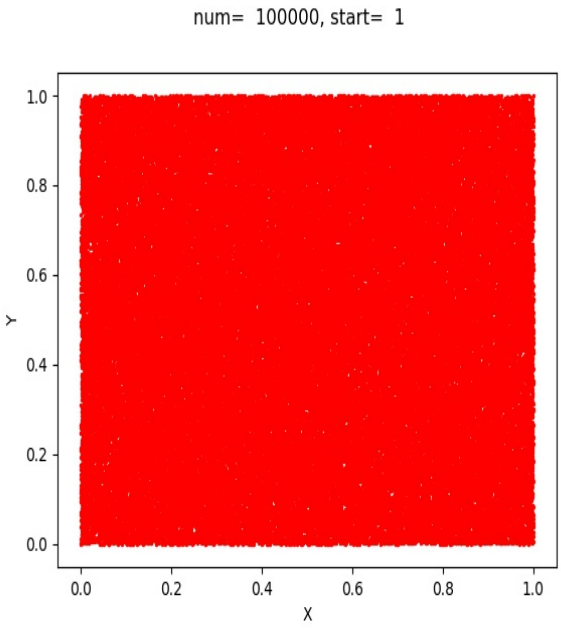


图 2: num=100000,start=1

继续设置点的个数num为1000000，如图所示，与上图相比,显得更加均匀，整个正方形区域没有明显的白色缺陷，仅在正方形边缘上有毛边。

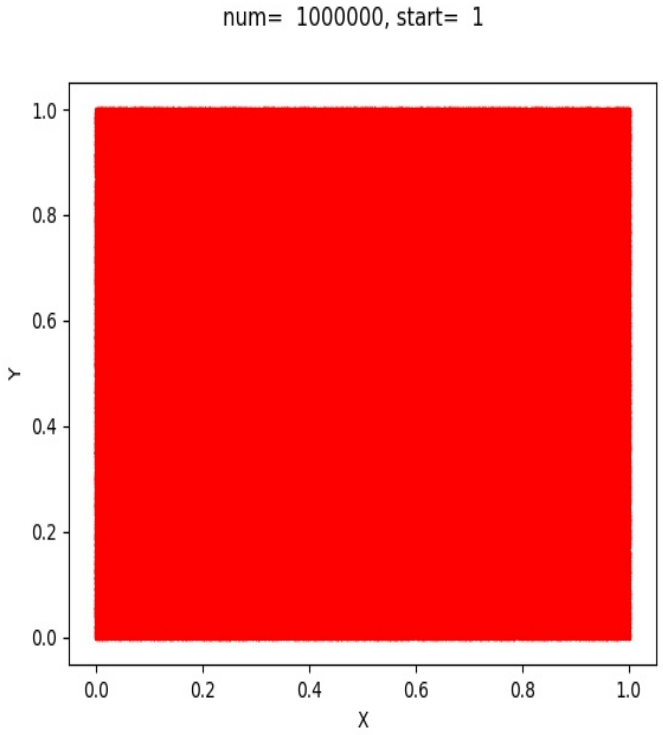
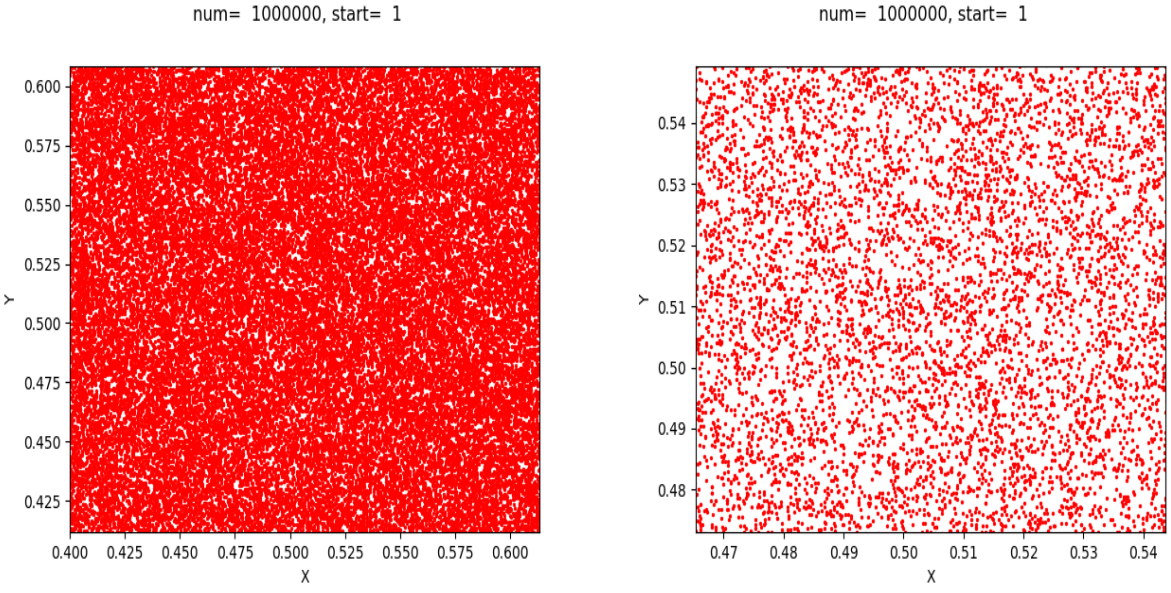


图 3: num=1000000,start=1

我们将上图放大，得到下面两张图：



(a) 放大图1

(b) 放大图2

图 4: num=1000000,start=1放大图

基本上看不出明显的相关性，分布还是比较均匀的。因此编写代码中的16807随机数生成器生成随机数比较成功。

4.2 $\langle x^k \rangle$均匀性检验

运行程序，进行$\langle x^k \rangle$均匀性检验，得到以下数据：

请输入起始值：1

Num	k=1	k=2	k=3	k=4	k=5
100	-0.014271	-0.019721	-0.018908	-0.017227	-0.015669
200	0.002975	0.002152	0.001635	0.001076	0.000526
400	-0.004730	-0.001428	0.001428	0.003112	0.003960
800	0.007755	0.011438	0.013358	0.013992	0.013861
1600	0.002086	0.004240	0.005248	0.005564	0.005514
3200	0.005908	0.007202	0.007239	0.006810	0.006225
6400	-0.001732	-0.000899	-0.000482	-0.000350	-0.000361
12800	-0.000333	-0.000970	-0.001135	-0.001133	-0.001094
25600	0.000432	0.000498	0.000437	0.000344	0.000239
51200	-0.000287	0.000077	0.000274	0.000338	0.000330
102400	-0.000277	-0.000127	0.000038	0.000153	0.000219
204800	-0.000300	-0.000250	-0.000174	-0.000113	-0.000075
409600	0.000344	0.000306	0.000282	0.000269	0.000257
819200	-0.000015	0.000040	0.000083	0.000110	0.000124
1638400	-0.000194	-0.000137	-0.000080	-0.000043	-0.000020
3276800	-0.000294	-0.000219	-0.000154	-0.000113	-0.000087
6553600	-0.000128	-0.000120	-0.000112	-0.000104	-0.000095
13107200	-0.000059	-0.000047	-0.000035	-0.000025	-0.000017
26214400	0.000044	0.000055	0.000060	0.000062	0.000062
52428800	-0.000014	-0.000007	-0.000000	0.000005	0.000009
104857600	-0.000036	-0.000031	-0.000025	-0.000020	-0.000017
209715200	-0.000036	-0.000032	-0.000027	-0.000024	-0.000020

图 5: $\langle x^k \rangle$均匀性检验数值,start=1

从图中数据可以看出，同一个k值下，随着N的增大，虽然$\langle x^k \rangle$理论与实际之差有时变大有时变小，但在总体上呈逐渐变小的趋势。

对于同一个N值，不同的k之间相差的并不是很大，也好像没有明确的规律。

将对应点的坐标绘制成折线图，得到更直观地表现，如下图所示。

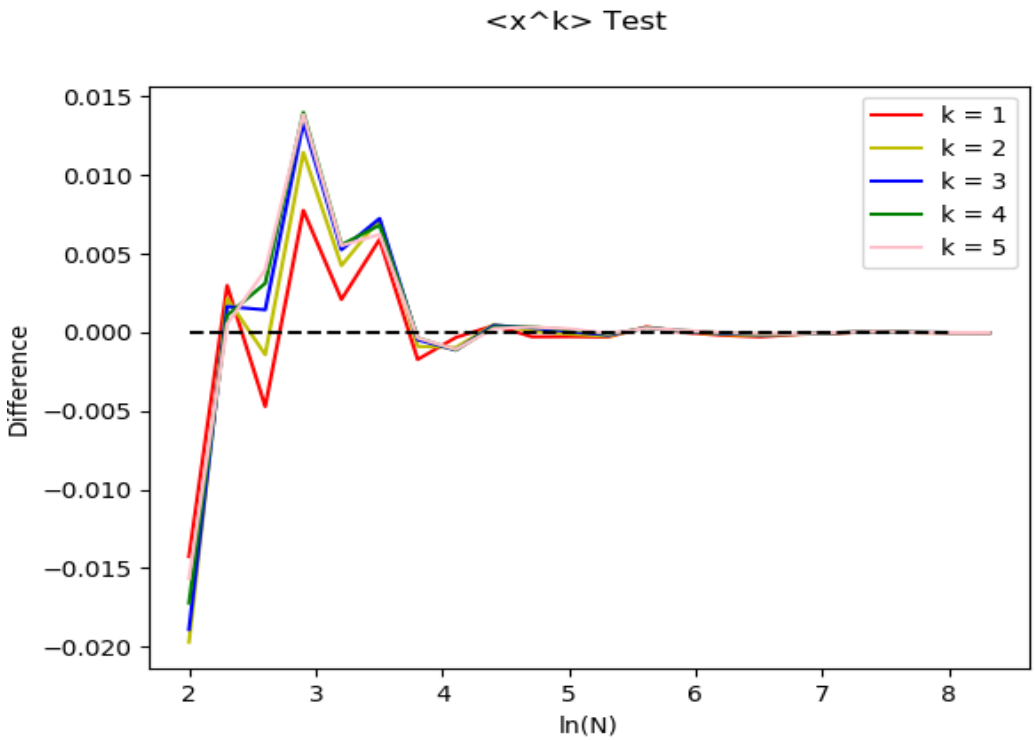


图 6: $\langle x^k \rangle$均匀性检验数据图像,start=1

可以看到，随着N的增大，震荡的幅度逐渐变小，为了验证这个观点，我们将后端放大继续看一下。

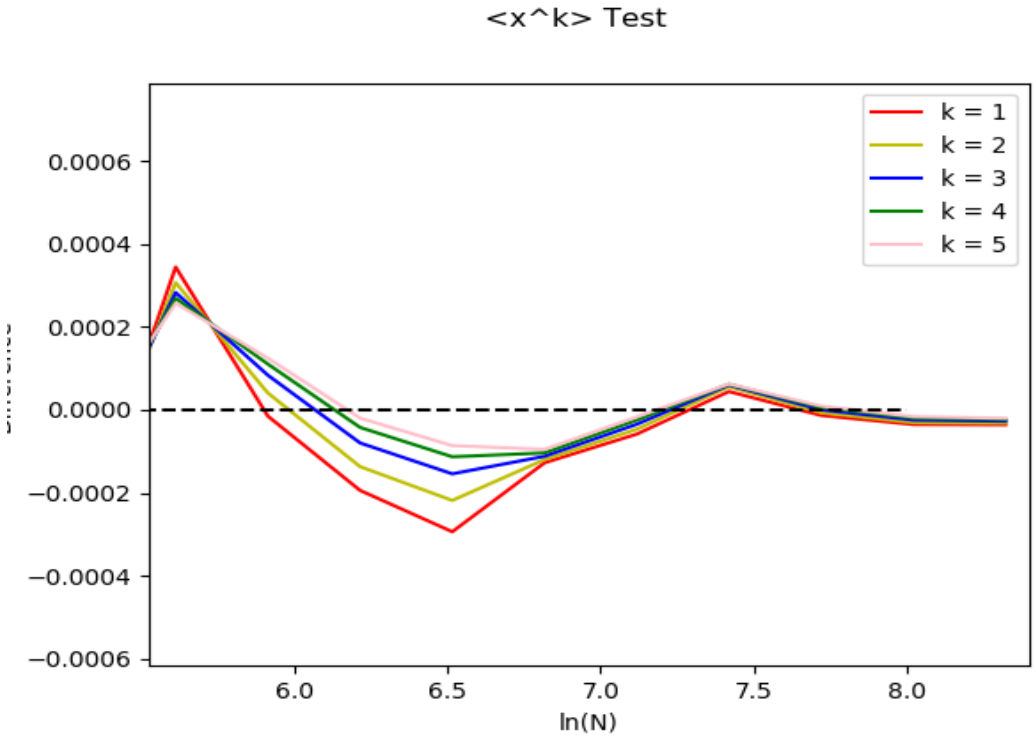


图 7: $< x^k >$均匀性检验数据图像（放大版）,start=1

随着N增大，震荡逐渐减小，最后逐渐趋于0，比较符合我们的论断。

另外我们也可以比较直观地看出，每条线的变化趋势很类似。用直觉分析也可以想到，如果一个k值对应的Difference变大的话，其他的k值应该也很大几率上是变大，因此不同的k值对应的曲线变化趋势会很类似。当然这只是直觉上的分析，因为我们也可以看到图中有N值变大但不同的k值对应的Difference既有增大也有减小的情况。

我们再观察一下不同的起始值对$\langle x^k \rangle$均匀性检验数据的影响。

设置起始值为1000，我们得到以下数值：

请输入起始值：1000

Num	k=1	k=2	k=3	k=4	k=5
100	-0.000746	0.003622	0.006371	0.006896	0.006075
200	0.010180	0.012605	0.012237	0.010872	0.009245
400	-0.000039	-0.001754	-0.001852	-0.001688	-0.001577
800	0.002838	0.004907	0.006171	0.006651	0.006645
1600	0.010234	0.011540	0.011304	0.010524	0.009587
3200	0.003186	0.003241	0.003384	0.003549	0.003672
6400	-0.000369	-0.001239	-0.001268	-0.001040	-0.000787
12800	-0.000534	-0.000757	-0.000626	-0.000430	-0.000256
25600	-0.000017	0.000211	0.000377	0.000500	0.000588
51200	-0.001085	-0.001444	-0.001402	-0.001241	-0.001065
102400	-0.001396	-0.001646	-0.001522	-0.001304	-0.001085
204800	-0.000483	-0.000830	-0.000880	-0.000829	-0.000754
409600	-0.000384	-0.000595	-0.000637	-0.000626	-0.000601
819200	0.000005	-0.000155	-0.000244	-0.000289	-0.000310
1638400	0.000005	0.000009	0.000013	0.000018	0.000023
3276800	0.000130	0.000115	0.000091	0.000073	0.000060
6553600	0.000129	0.000129	0.000104	0.000081	0.000063
13107200	0.000069	0.000059	0.000043	0.000028	0.000018
26214400	-0.000018	-0.000008	-0.000004	-0.000004	-0.000005
52428800	0.000031	0.000033	0.000031	0.000029	0.000028
104857600	-0.000017	-0.000011	-0.000007	-0.000004	-0.000003
209715200	-0.000029	-0.000025	-0.000021	-0.000018	-0.000015

图 8: $\langle x^k \rangle$均匀性检验数据,start=1000

直接分析数据得到的结论与起始值为1的时候一样，我们不再赘述。

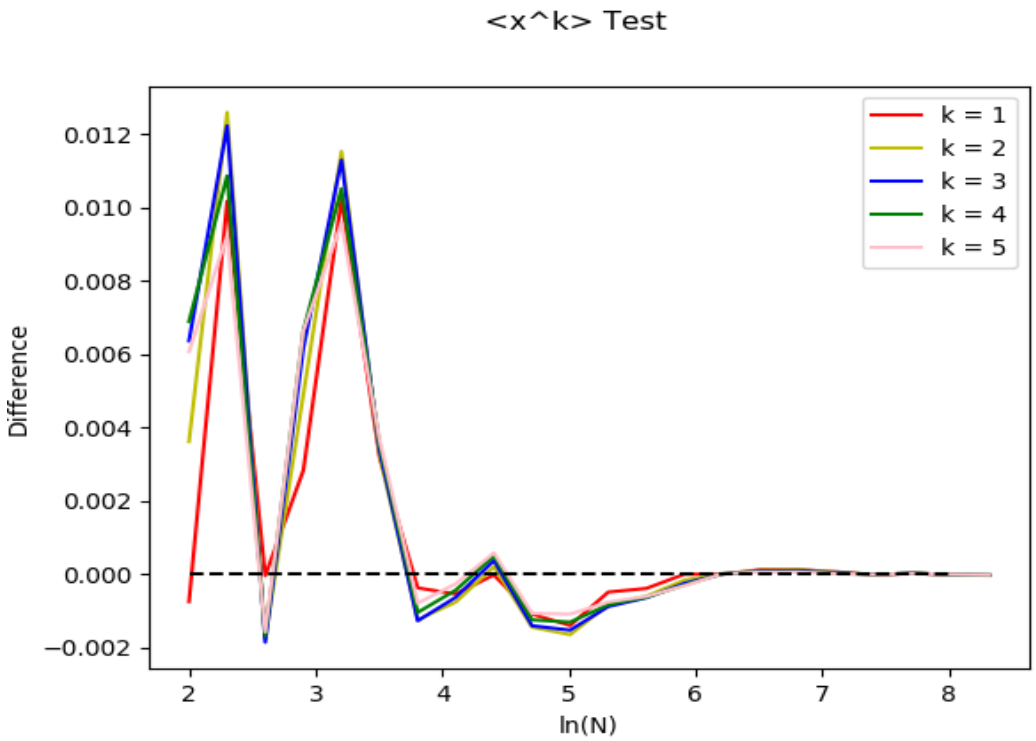


图 9: $\langle x^k \rangle$均匀性检验数据图像,start=1000

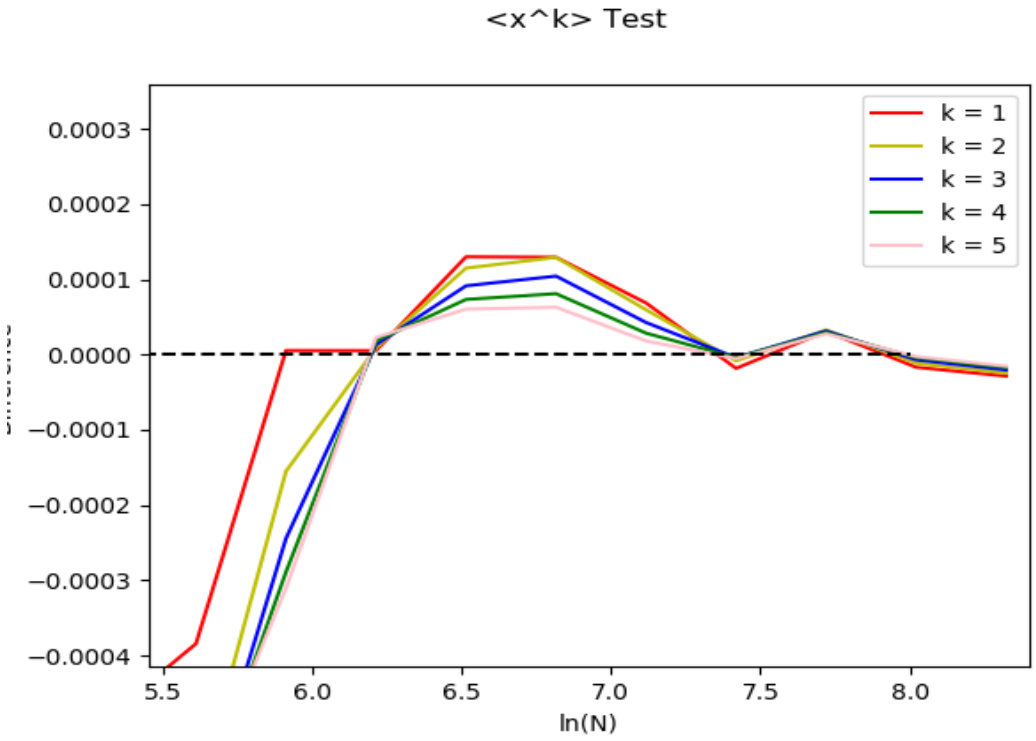


图 10: $\langle x^k \rangle$均匀性检验数据图像（放大版）,start=1000

这张图与起始值为1的时候就明显不同，前面的震荡很大，但最后依然是震荡减小并且逐渐趋于0。

因此我们可以发现，对于不同的起始值，$< x^k >$均匀性检验数据随机数很少的时候会有较大区别，但在随机数很多的时候都与理论值非常符合。

4.3 C(l)独立性检验

运行程序，进行C(l)独立性检验，得到以下数据：

请输入起始值：1

Num	l=1	l=2	l=3	l=4	l=5	l=6	l=7	l=8	l=9	l=10
100	0.0899	-0.0617	0.0152	0.0550	-0.0143	0.0686	0.1512	0.1786	-0.0048	0.0499
200	0.0489	-0.0505	0.1000	0.0462	-0.0101	-0.0317	0.0172	0.0449	-0.1207	-0.0052
400	0.0313	-0.0111	0.0664	0.0003	0.0244	-0.0150	0.0530	-0.0019	0.0127	-0.0035
800	0.0403	-0.0248	0.0379	0.0330	-0.0167	-0.0454	0.0351	-0.0060	-0.0198	-0.0393
1600	0.0225	-0.0431	-0.0147	0.0329	-0.0130	-0.0108	-0.0092	-0.0074	0.0392	-0.0383
3200	0.0111	-0.0133	-0.0221	0.0113	-0.0036	-0.0084	-0.0059	-0.0003	0.0187	-0.0438
6400	0.0055	-0.0136	-0.0101	-0.0059	-0.0157	-0.0077	-0.0184	0.0033	0.0217	-0.0223
12800	-0.0022	-0.0124	0.0005	-0.0029	-0.0058	-0.0122	-0.0029	-0.0013	0.0103	0.0100
25600	0.0020	-0.0066	0.0030	-0.0036	-0.0035	-0.0109	0.0019	-0.0006	0.0042	-0.0004
51200	0.0027	-0.0029	0.0030	0.0010	-0.0033	-0.0065	-0.0067	-0.0031	0.0058	-0.0003
102400	0.0019	-0.0026	0.0030	0.0037	-0.0013	-0.0013	-0.0057	-0.0042	0.0040	-0.0018
204800	-0.0004	-0.0025	-0.0015	0.0008	0.0005	0.0000	-0.0010	-0.0027	0.0006	0.0021
409600	-0.0002	-0.0023	-0.0005	-0.0014	-0.0017	-0.0014	-0.0003	-0.0012	0.0009	-0.0008
819200	-0.0001	-0.0017	0.0005	-0.0008	0.0003	0.0003	-0.0004	0.0004	0.0006	0.0007
1638400	-0.0003	-0.0011	0.0002	-0.0003	-0.0002	-0.0004	0.0001	-0.0006	-0.0001	0.0001
3276800	-0.0003	-0.0004	0.0009	0.0000	-0.0006	0.0001	0.0001	0.0001	0.0001	-0.0001
6553600	-0.0001	-0.0004	0.0009	-0.0002	0.0000	0.0004	0.0003	0.0005	0.0000	-0.0004
13107200	0.0002	-0.0002	0.0005	0.0001	-0.0002	0.0001	0.0001	0.0003	-0.0001	-0.0004
26214400	0.0001	-0.0001	0.0002	0.0001	0.0001	0.0000	-0.0001	0.0000	0.0000	-0.0003
52428800	0.0001	-0.0003	0.0001	0.0000	0.0001	-0.0001	0.0001	0.0001	-0.0000	-0.0002
104857600	0.0000	-0.0001	0.0001	0.0001	-0.0000	-0.0000	0.0001	0.0001	-0.0001	-0.0001
209715200	0.0001	-0.0000	0.0001	0.0000	-0.0001	-0.0000	0.0001	0.0000	-0.0000	-0.0001

图 11: C(l)独立性检验数据,start=1

可以发现，l相同时，随着N逐渐变大，C(l)逐渐减小，最后基本上趋于0。

对于同一个N值，不同的k之间相差的并不是很大，也好像没有明确的规律。

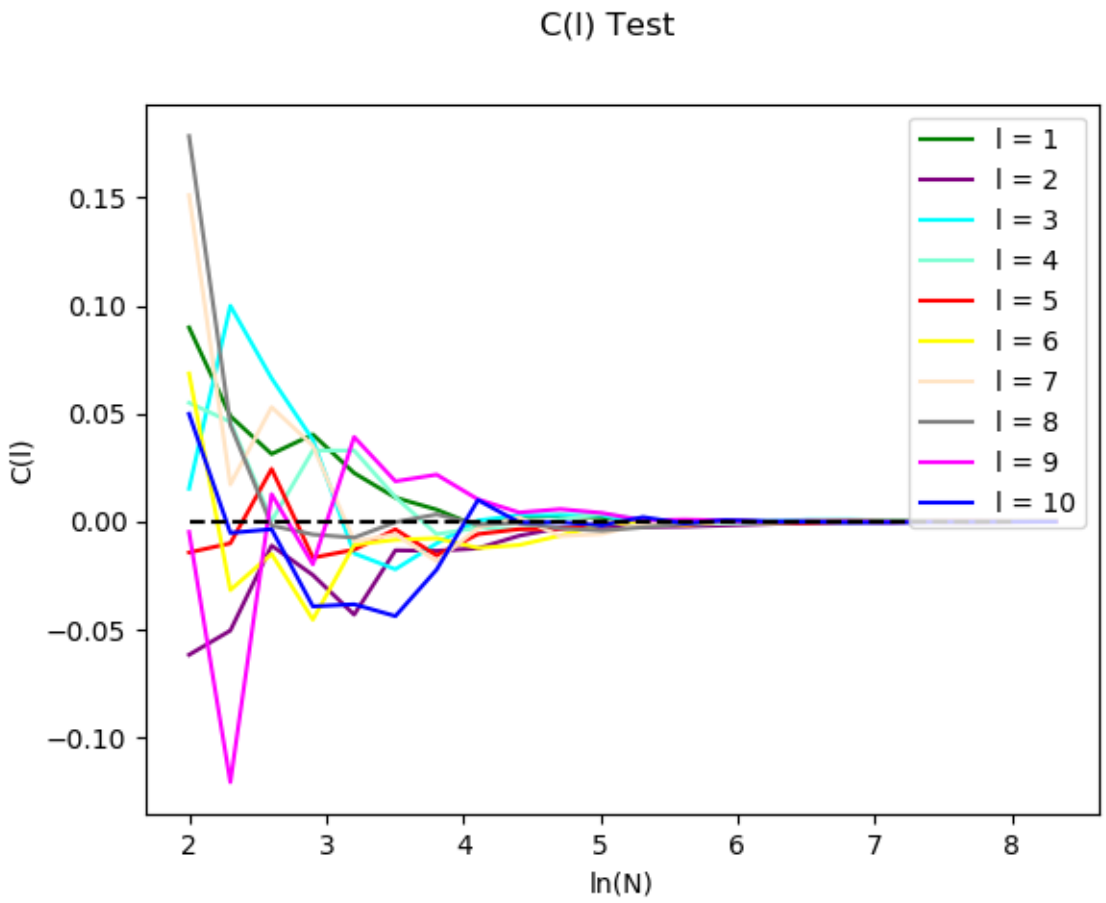


图 12: C(l)独立性检验数据图像,start=1

与$\langle x^k \rangle$均匀性检验数据图像类似，N值小的时候震荡很大，随着N值增大，数据的震荡逐渐减小，而且逐渐趋于0。与$\langle x^k \rangle$均匀性检验数据图像的不同点在于，不同的l没有类似的变化趋势，可以从N值较小的区域看出，像是一堆乱麻。

我们将后端放大，来继续看下我们的观点是否正确。

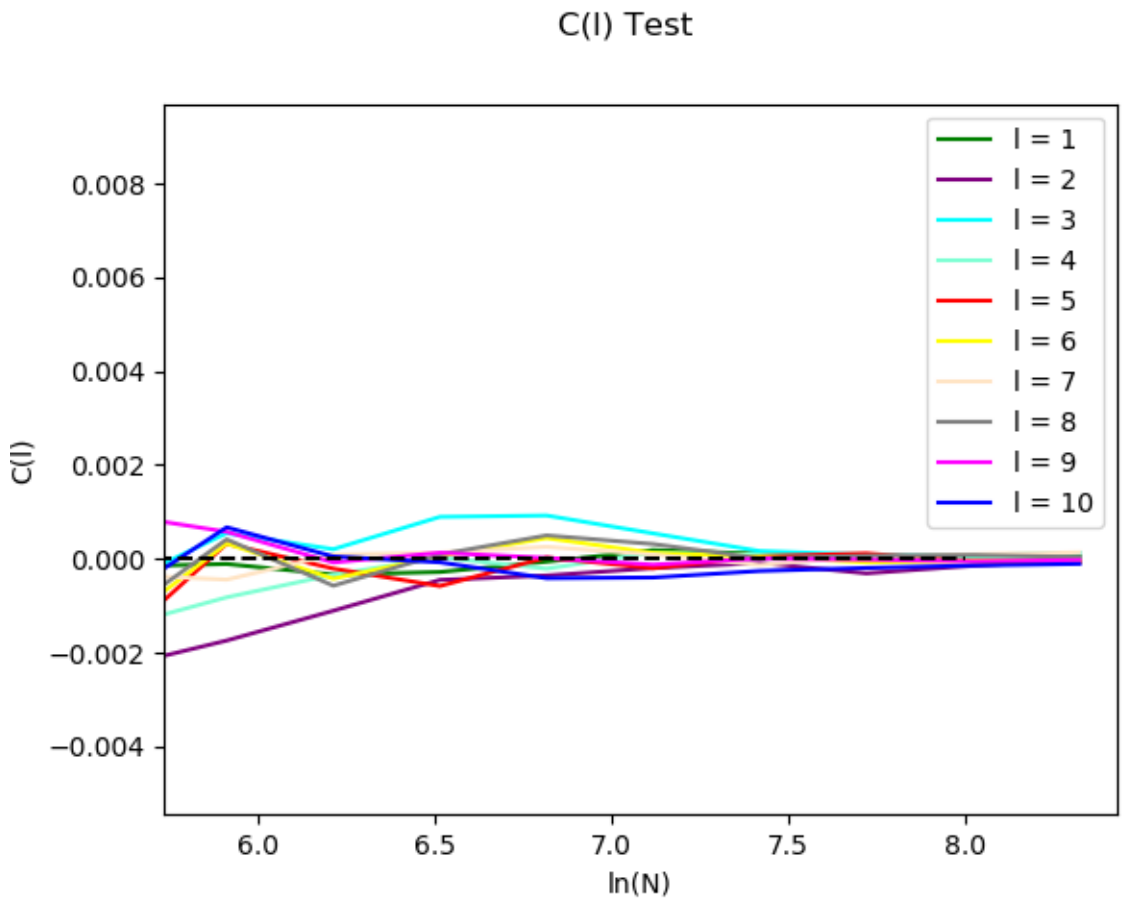


图 13: C(l)独立性检验数据图像（放大版）,start=1

震荡逐渐减小，最后逐渐趋于0，比较符合我们的论断。
与$\langle x^k \rangle$均匀性检验类似，我们设置不同的起始点，看下对数据的影响。
设置起始值为1000，我们得到以下数值：

请输入起始值：1000

Num	1=1	1=2	1=3	1=4	1=5	1=6	1=7	1=8	1=9	1=10
100	0.0598	0.1743	0.0509	0.0795	0.1511	0.1973	0.0805	0.2093	0.1763	0.0177
200	0.0597	0.2180	0.0390	0.1471	0.0664	0.0808	0.0445	0.1410	0.0682	-0.0136
400	-0.0669	0.0255	-0.0172	0.1251	0.0498	0.0283	-0.0296	0.0812	0.0509	-0.0250
800	-0.0511	0.0395	0.0065	0.0517	0.0184	0.0099	-0.0028	0.0375	0.0212	-0.0218
1600	-0.0361	0.0266	0.0179	0.0339	-0.0191	-0.0207	-0.0073	0.0339	0.0069	-0.0011
3200	-0.0282	0.0395	0.0050	0.0324	-0.0129	-0.0082	-0.0022	0.0094	0.0097	-0.0080
6400	-0.0253	0.0164	0.0139	0.0210	-0.0139	-0.0059	0.0028	0.0059	0.0016	-0.0019
12800	-0.0098	0.0054	0.0176	0.0187	-0.0053	-0.0037	0.0032	0.0047	0.0056	0.0011
25600	-0.0043	0.0028	-0.0026	0.0134	-0.0051	-0.0093	-0.0048	-0.0001	0.0016	-0.0017
51200	-0.0030	0.0057	-0.0004	0.0108	-0.0015	0.0009	-0.0042	-0.0030	0.0034	0.0004
102400	0.0025	0.0078	-0.0018	0.0054	0.0001	0.0014	-0.0039	-0.0027	-0.0031	0.0011
204800	0.0014	0.0041	0.0021	0.0010	-0.0002	0.0023	-0.0026	0.0021	-0.0017	0.0013
409600	-0.0003	0.0037	-0.0011	0.0016	-0.0006	-0.0002	-0.0008	-0.0005	-0.0007	-0.0013
819200	-0.0005	0.0022	-0.0013	0.0007	0.0009	-0.0003	-0.0007	-0.0009	0.0011	-0.0005
1638400	-0.0003	0.0003	-0.0003	-0.0000	0.0015	-0.0008	-0.0001	-0.0000	-0.0000	-0.0012
3276800	0.0001	0.0004	-0.0004	0.0005	0.0004	-0.0006	0.0001	-0.0000	0.0004	-0.0001
6553600	0.0006	0.0000	-0.0000	0.0004	0.0001	-0.0002	0.0001	0.0003	0.0001	0.0002
13107200	0.0004	0.0001	-0.0001	0.0007	0.0001	-0.0000	-0.0001	0.0002	0.0000	0.0004
26214400	0.0000	-0.0002	-0.0000	0.0002	-0.0000	0.0000	-0.0001	0.0002	-0.0000	0.0001
52428800	0.0001	-0.0001	0.0001	0.0001	0.0001	-0.0000	-0.0002	0.0001	0.0000	-0.0000
104857600	0.0001	-0.0002	0.0000	0.0001	0.0000	-0.0001	0.0000	0.0002	-0.0001	-0.0001
209715200	0.0001	-0.0001	0.0001	0.0001	-0.0000	-0.0001	0.0001	0.0001	-0.0000	-0.0001

图 14: C(l)独立性检验数据,start=1000

直接分析数据得到的结论与起始值为1的时候一样，我们不再赘述。

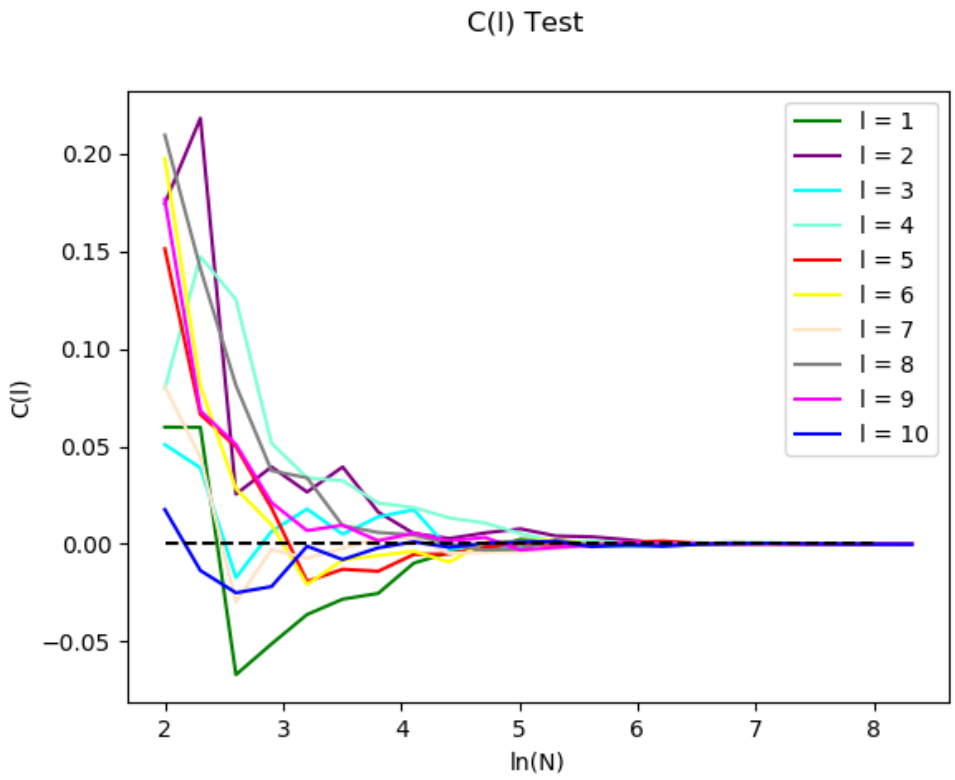


图 15: C(l)独立性检验数据图像,start=1000

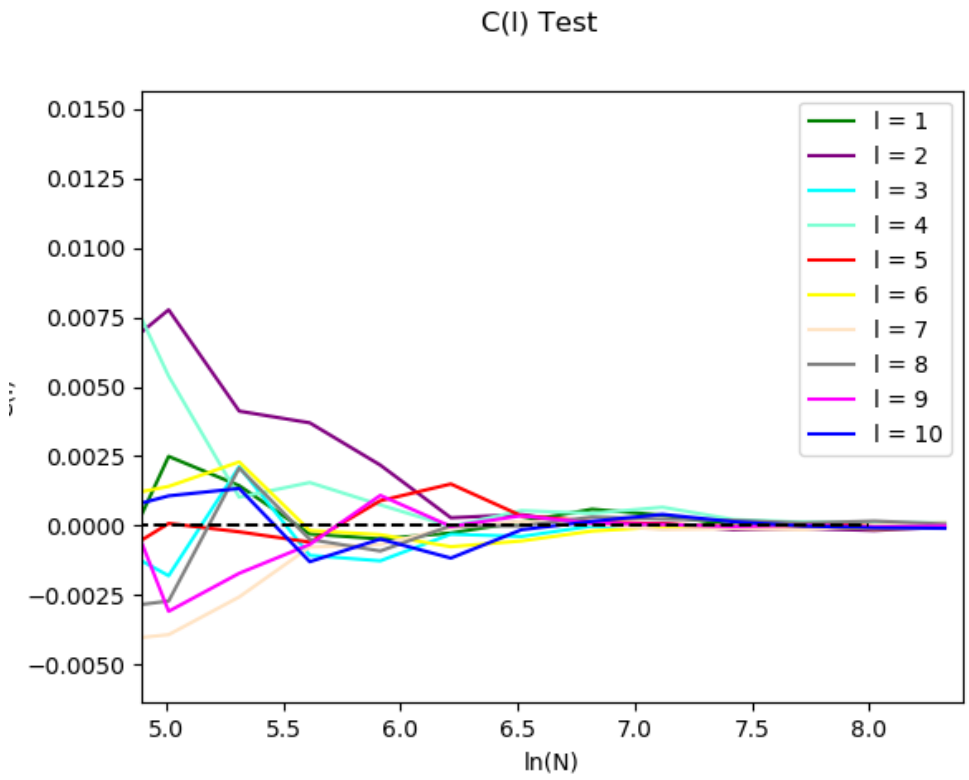


图 16: C(l)独立性检验数据图像（放大版）,start=1000

可见在N值较小的时候也很混乱，不同的起始值数据不同，但最后都会随着N值变大，震荡减小，逐渐趋于0。

5 讨论

5.1 原理的讨论

本次实验原理较简单，主要是16807随机数产生程序的正确编写与Schrage方法防止数据超过范围限制。$\langle x^k \rangle$均匀性检验与C(l)独立性检验中只需按照对应的公式求解k阶距与C(l)，与理论结果进行比较即可。

5.2 算法的改进

由于本人的计算机知识并不是很丰富，下面的讨论可能会欠妥，源代码的编写也肯定会有很多需要改进的地方，希望以后学习中能够不断完善。

1. 编程易错点：

   程序中大部分有double型变量参加的运算都要先将其中的int型变量进行强制类型转换变成double型变量。例如求$x_n = I_n/m$时，需要将其中的一个变量转换成double型变量，否则会因为int型变量的除法相当于进行除法并取整，最后得到的随机数都是0。对于$\langle x^k \rangle$的标准值$\frac{1}{k+1}$同理。

   设置一维数组时也需要设置在main函数外面，否则会因为函数内部的栈的内存的限制对数组长度有很大的限制。

   switch中的case语句后面必须要加花括号，否则会因为文件读写问题报错，与一般情况下的case语句不同。

2. 数据结构的选取：

   可以看到我写的程序中，在以连续两个随机数作为点的坐标绘出若干点的平面分布图的程序中，我并没有运用到一维数组，直接用一个double型变量就可以得到点的坐标。这样可以有效地减小程序的空间复杂度，也没有多出更高量级的时间复杂度，而且理论上可以求的点的数量非常大，主要取决于能输出的文件的存储空间大小。如果直接用一维数组来保存，一维数组的长度上限远比直接用double型变量得到的小，而且空间占用会很大。

   但在后续的实验中，我选择使用一维数组进行数据的保存，主要是因为C(l)检验中由于需要很方便地求取对应下标的数据，而用一个变量保存则很麻烦。

3. 结果的改进：

   可以对结果进行一些改进。对于单个N值下产生16807随机数产生器产生二维坐标点的算法并不需要修改太多，已经是十分简便高效的算法了。但对于均匀性与独立性检验的算法中，我们可以设置更加窄的N的步长来获取更精确的图像，代价是时间复杂度会提升很多，特别对于$N > 10^7$的点来说，生成随机数所需的时间已经非常的长了。如果要做更精确的分析，可以考虑减小步长，本实验中没有设置那么小的步长。

5.3 结果的讨论

1. 可以看到在N值较小的情况下，$\langle x^k \rangle$均匀性检验与C(l)独立性检验给出16807随机数生成器的均匀性与独立性并不好，这主要来自于偶然误差与统计所需大数据的要求。首先，N小的情况下，数据对起始值的依赖将更大，数据少时个体对统计结果影响很大。其次，统计结果的准确性很大程度上依赖于极大的数据量，例如概率的公理性定义就要求测试的数量是无穷大。一般而言，随着数据量的增大，实际情况将更接近于理论分析下的统计结果，这也跟我们实验中得到的一致。

2. 对于$\langle x^k \rangle$均匀性检验与C(l)独立性检验实验中，都可以看出在N小时与理论值偏差较大，但随着N的增加，两个检验都会与理论值非常接近，说明16807随机数产生器确实在均匀性与独立性方面在大多数情况下都能满足随机数生成的要求。

3. 在$\langle x^k \rangle$均匀性检验中，我们也可以直观地看出，每条线的变化趋势很类似。简单地说，与我们在计算结果中分析的一样，是因为$\langle x^i \rangle$较大时，$\langle x^j \rangle$也较大，这个关系在很大几率上的成立的，仅在些许情况下不成立，我们在图中也可以很容易地找到这种例外的点。但在C(l)独立性检验中，不同l值的C(l)则没有这样明显的关联。

4. 在计算结果的讨论里，可以看出，不同起始值会实际影响到数据的大小结果。但当点数量非常多时，在误差允许的范围内可以认为符合理论。因此大多数情况下，我们运用16807随机数生成器产生随机数时，随便设置一个种子值即可。但由于还是伪随机数的生成，因此需要不同的随机数列的时候，要尽量保证随机数种子值的不同。