1.

Leibniz定理：

如果已经求出

接下来求

所以。

所以。

所以

当和前k位相同时，则为的前k位。

但求和，当n有多少位，结果大概前面有多少个0，如n=10^10,和约为5\*10^11，即结果只有前11位(大概)是有效的，意味着求得位数精确的Pi，需要很大的n。如求100位，至少需要n>10^100，计算机难以在短时间内实现。

2.

wallis公式：

对于分子2,2,4,4,6,6,……,2n,2n：每个数除以2，就变为

1,1,2,2,3,3,……,n,n。求1~n关于某个质数的系数。

对于分母：1,3,3,5,5,7,……,2n-1,2n-1,2n：补上2,2,4,4,6,6,……,2n,2n，就变成1,2,2,3,3,4,4,……,2n,2n,2n+1。求1~2n+1关于某个质数的系数。

而1~m关于某个质数k的系数是，其中[x]代表不大于x的最大的整数。

所以，

即，

最后求得 为质数，为该质数的系数，若为大于0，则相乘，若小于0，则相除。因为没有上界，所以难以判断结果前几位正确。

3.

蒙特卡洛法：设置一个r\*r的矩形和与矩形内切的半径为r/2的圆。把大量点随机放入r\*r矩形内，记录点的总数D1和在圆内的点的总数D2（在圆内代表点到中心(矩形,圆)的距离小于等于r/2）。

因为，所以。

x=rand()/(double)RAND\_MAX;//横坐标

y=rand()/(double)RAND\_MAX;//纵坐标

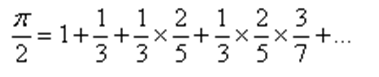
在0~1之间取一个数，每个点在32768种固定的可能中选择一种可能。事实上，(x,y) x：0~32767/RAND\_MAX y：0~32767/RAND\_MAX，共32768\*32768= 1073741824种可能，经过计算，有843296058种可能使x^2+y^2>=1，在此情况下，pi’=3.141523。而(x,y)每种可能性出现的概率相同，所以求得的数值在pi’=3.141523附近。但经测试数据不稳定，波动较大。

程序中：

可以把一个0~1的数的小数部分当成是32768进制，每一位的数为0~32767,而最后一位32767进制(之后没有位了)。

u代表小数点后1位，v代表小数点后两位，r代表小数点后三位……

无疑，考虑后面的位数越多，更接近真实的pi。但由于用double存储，double只保留15~16的精度，当位数大于3（32768^4大于16位）时，最后一位的部分没有影响，而当位数大于等于3时，1/32768^3已经在10位之后，对pi前几位影响甚微。

4. 

对于任意t，有，即后一个数小于前一个数的一半。

若前k项相加，每个数求小数点后m位，每个数误差小于 (忽略小数点第n+1位及后面的位数)，总误差小于。

对于n的逐渐递增，逐渐减小，直到n=t，结果小于。而总的值大于，小于。