



《计算机模拟》

第1讲 - 简介

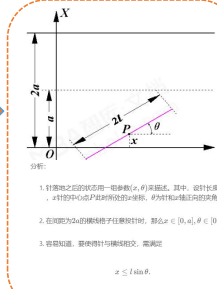
胡贤良
浙江大学数学科学学院

关于模拟

- 模拟**：利用物理的、数学的模型来类比、模仿现实系统及其演变过程，以寻求过程规律的一种方法。
- 基本思想**：建立一个试验的模型，这个模型包含所研究系统的主要特点。通过对这个实验模型的运行，获得所要研究系统的必要信息。
- 基本方法**：数学模拟、物理模拟



π 的近似计算：Buffon投针



分析：

- 针落地之后落在任意一条横线 $[x, x+a]$ 上，其中， x 为针左端点， $x+a$ 为针右端点。针落在 $[x, x+a]$ 上的概率为 $\frac{a}{a} = 1$ 。
- 在长度为 a 的区间内任意取一点 p ，则 $p \in [0, a]$ ， $p \in [0, a]$ 。
- 针与横线相交，即针与横线相交，即 $p \leq l \sin \theta$ 。

上式表明 $p \leq l \sin \theta$ ，当针与横线相交时一定成立！

曲线 $y = l \sin \theta$ 与 $y = a$ 下面积比

$$\frac{M}{N} = \frac{l}{a\pi} \int_0^{\pi} \sin \theta d\theta = \frac{l}{a\pi} \cdot \frac{2}{\pi} = \frac{2}{\pi}$$

为了尽量增大这个比值（最后取 $l = a$ ）

投针模拟结果

实验者	时间	针长	投针次数	相交次数	π 近似值
Wofl	1850	0.80	5000	2532	3.25956
Smith	1855	0.60	3204	1218	3.15665
Fox	1884	0.75	1030	489	3.15951
Lazzarini	1901	0.83	3408	1808	3.14159292

估算依据

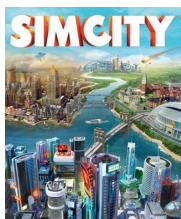
$$\frac{M}{N} = \frac{2}{\pi} \rightarrow \pi = 2 * N / M$$

- 原则上，实验次数越多，所得到的结果才越精确



数学(计算机)模拟

- 数学模拟**：在一定的假设条件下，运用数学运算模拟系统的运行。
- 计算机模拟**：现代的数学模拟都是在计算机上进行的，可以反复进行，改变系统的结构和系数都比较容易。实际问题需要考虑一些复杂的随机因素，理论分析须作模型简化和假设，导致的结果与实际情况相差较远。此时，计算机模拟几乎是唯一选择。
- 蒙特卡洛 (Monte Carlo) 方法**：一种应用随机数来进行计算机模拟的方法。对研究的系统进行随机抽样，通过对样本值的观察统计，计算感兴趣的某些系统参数。



π 的近似计算 - 计算机模拟

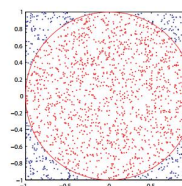
记

圆的半径为 r ，则正方形边长为 $2r$

总共撒点 n 个，其中有 p 个（红点）落在圆内。可计算得 π 的近似值：

$$\frac{p}{n} = \frac{\pi r^2}{(2r)^2} = \frac{\text{area of circle}}{\text{area of square}} \approx \frac{\pi}{4}$$

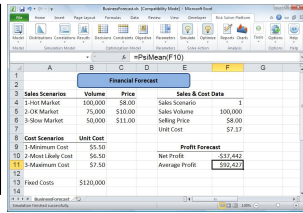
当随机点取得越多（但即使取 10 的 9 次方个随机点时，其结果也仅在前 4 位与圆周率吻合），其结果越接近于圆周率。



关于编程语言

1. 近似计算Pi的.m和.py
2. Python及其运算环境
3. Javascript、excel也可以用来做模拟！
4. 结论：`random()` + `plot()` 以及相关的数学运算函数库即可构成所需函数。

S.A.Peng



Screenshot : Do Simulation with Excel

Octave :开源(可替代Matlab)

- https://wiki.octave.org/Octave_for_Microsoft_Windows

1. **Extract** the 7z/zip archives to a directory on the harddrive (such as **C:\Octave**). Spaces or non-ASCII characters in the path are discouraged and may cause program errors.
2. **Create a shortcut** to the **octave-launch.exe** file in the main installation directory. If a command-line only instance of Octave is desired, add **--no-gui** to the end of the Target field.
3. **IMPORTANT**: Run the **post-install.bat** file before running Octave the first time to reduce plot delays due to the Windows font cache and make the pre-installed packages visible to the system.

GNU Octave

Microsoft Windows

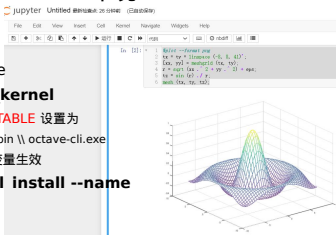
Note: All installers below bundle several Octave packages. Read more.

- Windows-64 (recommended)
 - octave-7.2.0-w64-installer.exe (~380 MB) (signature)
 - octave-7.2.0-w64-7z (~375 MB) (signature)
 - octave-7.2.0-w64-zip (~660 MB) (signature)
- Windows-32 (old computers)
 - octave-7.2.0-w32-installer.exe (~380 MB) (signature)
 - octave-7.2.0-w32-7z (~375 MB) (signature)
 - octave-7.2.0-w32-zip (~660 MB) (signature)
- Windows-64 (64-bit linear algebra for large data)
 - octave-7.2.0-w64-64-installer.exe (~380 MB) (signature)
 - octave-7.2.0-w64-64-7z (~375 MB) (signature)
 - octave-7.2.0-w64-64-zip (~660 MB) (signature)

Unless your computer has more than ~2GB of memory and you need to use the recommended Windows-64 version above.

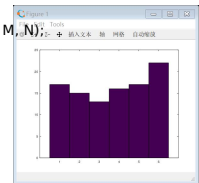
Jupyter 下的 Octave 环境

- **virtualenv** octave
- octave\Scripts\activate
- **pip3 install octave_kernel**
- 将环境变量 **OCTAVE_EXECUTABLE** 设置为
D:\Octave\Octave-7.2.0\bin\octave-cli.exe
重新启动Windows后该环境变量生效
- **python -m ipykernel install --name Octave-7.2.0 --user**
- ✓ Conda环境, 请参考:
https://blog.csdn.net/qq_44992157/article/details/123729375



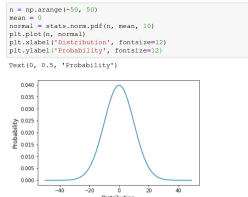
均匀分布随机数

- 离散情形:
 >> `K = unidrnd(6, 3, 10);` % (NMAX, M, N)
 >> `hist(K, [1,2,3,4,5,6]);` % 1:6
- 连续情形:
 >> `unifrnd()`
 >> `rand()`
`rng('shuffle')`: 基于时钟时间来设置随机数发生器的种子数。

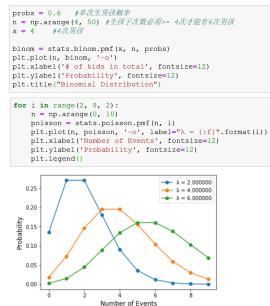


其它编程工具

随机数发生器 + 可视化



- Excel/Python 产生随机数的示例:
<https://open.baidu.com/article/4511a61269977a6a125d1.html>
<https://www.cnblogs.com/steven1024/p/5000000.html>



模拟案例 1: 电池问题

- 系统状态变量 $S \in \{0, 1, 2\}$: 可用的电池数
- 当 $S = 0$ 出现时, 系统停机
- 电池耗尽时长: 随机变量 $r \in \{1, 2, 3, 4, 5, 6\}$ 小时
- 电池充电时间: 2.5 小时
- 问: 系统可持续工作的时长?



初始: $x_0 = 0, S_0 = 2$ (t_0 代表第 1 个事件发生的时间) 和 t_0 内系统状态

$t_1 = 2.5$ (充电)

$t_2 = 5, S_1 = 1$

$t_3 = 7.5, S_2 = 2$ (电池耗尽就充电了, 否则就 $t < 2.5$, 直接充电)

$t_4 = 8, S_3 = 1$ (一个充电, 一个耗尽)

$t_5 = 10.5, S_4 = 2$

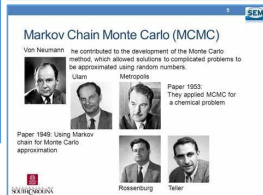
② ① ①

Monte Carlo Methods 发展历史

- 30年代：E. Formi (费米, 1901-1954) 发明“FERMIAC”装置模拟中子在核裂变过程中的随机扩散运动
- 原子弹时期：由 S. Ulam (1909-1984, 意大利)、J. von Neumann (1903-1957, 匈牙利) 和 N. Metropolis (1915-1999, 希腊) 正式提出，在“ENIAC”上实现了中子在原子弹内扩散和增殖的模拟（模拟一个中子需3分钟）
- 氢弹时期 (1948-1953)：在“MANIAC”上实现 **物质状态方程的蒙特卡罗模拟**（计算一个数据点需5小时）

基本想法：通过大量随机样本，去了解一个系统，进而得到所要计算的量。

诞生于“曼哈顿计划”：John von Neumann, Stan Ulam, and Nick Metropolis, all at the Los Alamos Scientific Laboratory, cook up the Metropolis algorithm, also known as the Monte Carlo method. 1946.



20世纪十大算法之首

- “最后的方法” - 可用于解决其他数值方法不能解决的问题，如：
 - 科学计算：粒子输运模拟、稀薄气体动力学、大科学实验模拟。
 - 人文学科：金融经济学、数理统计学
 - 数据分析领域：自然语言处理中的 LDA 模型、Hinton 较早提出的深度学习模型 DBN
- 没有计算机就没有蒙特卡罗模拟！
- 1950年代末：
 - 徐钟济教授留美回国，在中国科学院计算技术研究所讲授
 - 裴鹿成研究员，在中国科学院原子能研究所开展研究

- 一、1946 蒙特卡洛方法
- 二、1947 单纯形法
- 三、1950 Krylov 子空间迭代法
- 四、1951 矩阵计算的分解方法
- 五、1957 优化的 Fortran 编译器
- 六、1959 求矩阵特征值 QR 算法
- 七、1962 快速排序算法
- 八、1965 快速傅立叶变换
- 九、1977 整数关系探测算法
- 十、1987 快速多极算法

Monte Carlo Methods 概况

- 普遍认为 1946 年是该方法的“开创年”。
- 以 1980 年为分界线的前、后两个发展时期：
 - 1980 前：随机数发生慢 (2 个/秒, 1978)；伪随机数理论不严格；Metropolis 算法还未集成 Markov 链；实用方面尚不能求解最优化问题
 - 1980 后：随机数产生和检验方法“焕然一新”，真随机数 50 万个/秒，伪随机数种类丰富周期长；抽样效率大大提高；MCMC、拟蒙特卡洛方法、序贯蒙特卡洛方法以及并行化等
- 蒙特卡罗理论包括：
 - ① 随机数产生和检验方法
 - ② 概率分布抽样方法
 - ③ 降低方差提高效率方法

小结：HowTO MCM

1. 抽样 (Sampling)
 2. 估计 (Estimation)
 3. 优化 (Optimization)
- see www.montecarlohandbook.org for code and more information

课程教学安排 - 2023 秋冬

▶ 新学期	▶ 新学期
▼ 第1讲：课程简介	▼ 第1讲：课程简介
▼ 第2讲：概率基础	▼ 第2讲：蒙特卡罗
▶ 第3讲：随机数生成与检验	▶ 第10讲：优化算法
▼ 第4讲：生成模型与变分自编码器	▼ 第11讲：人工神经网络
▶ 第5讲：离散事件模拟	▼ 第12讲：生成式模型
▼ 第6讲：基本蒙特卡罗方法	▼ 第13讲：蒙特卡罗
▶ 第7讲：马尔科夫链蒙特卡罗方法	▶ 第14讲：变分自编码器



其他参考材料



Homework 01

- 上机课要求到指定机房，默认可使用系统已安装的 Matlab 环境
 - 允许自带电脑，鼓励安装 Ubuntu Linux 开发环境 (Jupyter-lab)
1. 课本第一章课后习题：2, 5, 6
 2. 请参考本讲稿的方法，计算 π 的近似值。并探索其他模拟方法（非数值计算方法），给出一类方法、实现和数值结果。

注意：

- 1或2配置完毕在前两周的上机课向我展示
- 3和4的主要过程和主要代码，写入作业本（偶数周上机课交作业）