

计算机随机模拟



基本内容：

- 1、计算机仿真的基本概念
- 2、随机数的产生
- 3、时间步长法
- 4、事件步长法
- 5、**Monte Carlo**方法



一、计算机仿真的基本概念

仿真：就是将所研究的对象用其它手段加以模仿的一种活动。

{ 实物仿真
非实物仿真

如曹冲称象、军事演习、
风洞试验等



计算机仿真：

是一种非实物仿真方法，通过建立数学模型、编制计算机程序实现对真实系统的模拟。从而了解系统随时间变化的行为或特性，以评价或预测一个系统的行为效果，为决策提供信息的一种方法。它是解决较复杂的实际问题的一条有效途径。

计算机仿真举例：

鼠疫的检测和预报，三峡的安全、生态，公交车的调度，航空管理，经营投资，道路的修建，通信网络服务，电梯系统服务等。



适用计算机仿真解决的问题:

- 难以用数学公式表示的系统，或者没有建立和求解数学模型的有效方法.
- 虽然可以用解析的方法解决问题，但数学的分析与计算过于复杂，这时计算机仿真可能提供简单可行的求解方法.
- 希望能在较短的时间内观察到系统发展的全过程，以估计某些参数对系统行为的影响.
- 难以在实际环境中进行实验和观察时，计算机仿真是唯一可行的方法，例如太空飞行的研究.
- 需要对系统或过程进行长期运行比较，从大量方案中寻找最优方案.



计算机仿真的优点:

- 便于重复进行试验，便于控制参数，时间短，代价小；
- 可以在真实系统建立起来之前，预测其行为效果，从而可以从不同结构或不同参数的模型的结果比较之中，选择最佳模型；
- 对于缺少解析表示的系统，或虽有解析表示但无法精确求解的系统，可以通过仿真获得系统运行的数值结果；
- 对于随机性系统，可以通过大量的重复试验，获得其平均意义上的特性指标。



仿真常用术语:

系统: 一些具有特定功能相互之间以一定的规律联系着的物体所组成的总体.

实体: 系统的对象、系统的组成元素都可以称为实体.

属性: 属性是对实体特征的描述, 可以是文字型、数字型或逻辑型.

活动: 实体在一段时间内持续进行的操作或活动.

例如: 银行系统, 销存系统



系统边界：为了限制所研究问题涉及的范围，用系统边界把所研究的系统与影响系统的环境区分开来。

事件：改变系统状态的瞬间变化的事情。

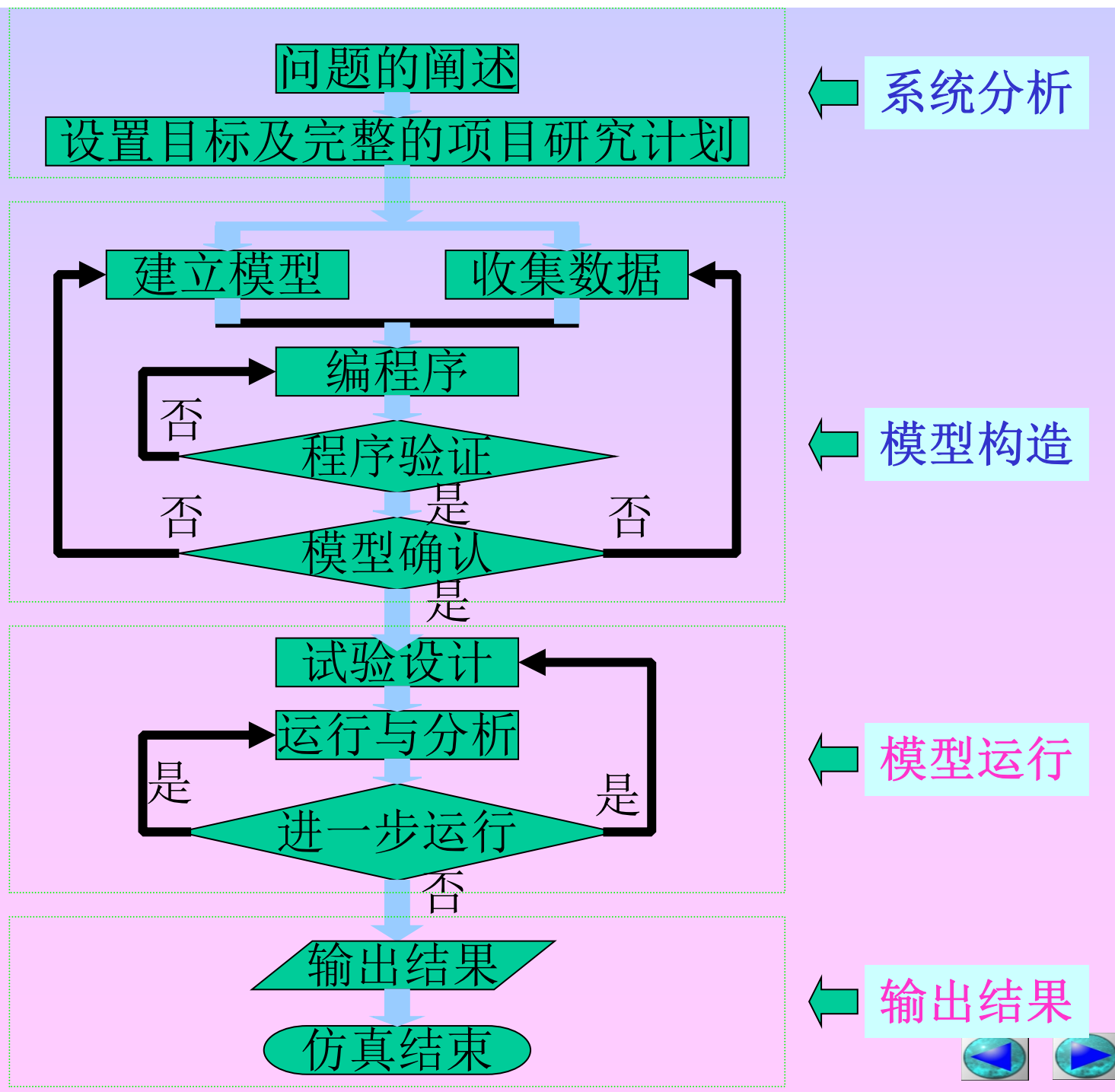
事件表：事件表一般是一个有序的记录列，每个记录包括事件发生时间、事件类型等一些内容。

状态：系统的状态是指在某一时刻实体及其属性值的集合。

仿真时钟：表示仿真时间的变量。



仿真研究步骤



系统分析

明确问题和提出总体方案：
把被仿真系统的内容表达清楚；
弄清仿真的目的、系统的边界；
确定问题的目标函数和可控变量；
找出系统的实体、属性和活动等。

模型构造

建立模型；
 选择合适的仿真方法（如时间步长法、事件表法等）；确定系统的初始状态；设计整个系统的仿真流程图。
收集数据；
编写程序、程序验证；
模型确认。



模型
的运行
与改进

运行：确定具体的运行方案, 如初始条件、参数、步长、重复次数等, 然后输入数据, 运行程序。

改进：将得出的仿真结果与实际系统比较, 进一步分析和改进模型, 直到符合实际系统的要求及精度为止。

设计
仿真
结果
输出
格式

设计出结构清晰的仿真结果输出。包括提供文件的清单, 记录重要的中间结果等。

输出格式要有利于用户了解整个仿真过程, 分析和使用仿真结果。



计算机仿真举例

例1、射击命中率

在我方某前沿防守地域，敌人以一个炮排（含两门火炮）为单位对我方进行干扰和破坏。为躲避我方打击，敌方对其阵地进行了伪装并经常变换射击地点。经过长期观察发现，我方指挥所对敌方目标的指示有50%是准确的，而我方火力单位，在指示正确时，有 $\frac{1}{3}$ 的射击效果能毁伤敌人一门火炮，有 $\frac{1}{6}$ 的射击效果能全部消灭敌人。

现在希望能用某种方式把我方将要对敌人实施的20次打击结果显现出来，确定有效射击的比率及毁伤敌方火炮的平均值。



分析：这是一个概率问题，可以通过理论计算得到相应的概率和期望值. 但这样只能给出作战行动的最终静态结果，而显示不出作战行动的动态过程.

为了能显示我方20次射击的过程，现采用模拟的方式。

1. 问题分析

需要模拟出以下两件事：

[1] 观察所对目标的指示正确与否

模拟试验有两种结果，每一种结果出现的概率都是 $1/2$.

因此，可用投掷一枚硬币的方式予以确定，当硬币出现正面时为指示正确，反之为不正确。



[2] 当指示正确时，我方火力单位的射击结果情况

模拟试验有三种结果：

毁伤一门火炮的可能性为 $1/3$ (即 $2/6$)，

毁伤两门的可能性为 $1/6$ ，

没能毁伤敌火炮的可能性为 $1/2$ (即 $3/6$)。

这时可用投掷骰子的方法来确定：

出现 1 、 2 、 3 点： 则认为没击中敌人；

出现 4 、 5 点： 则认为击毁敌人一门火炮；

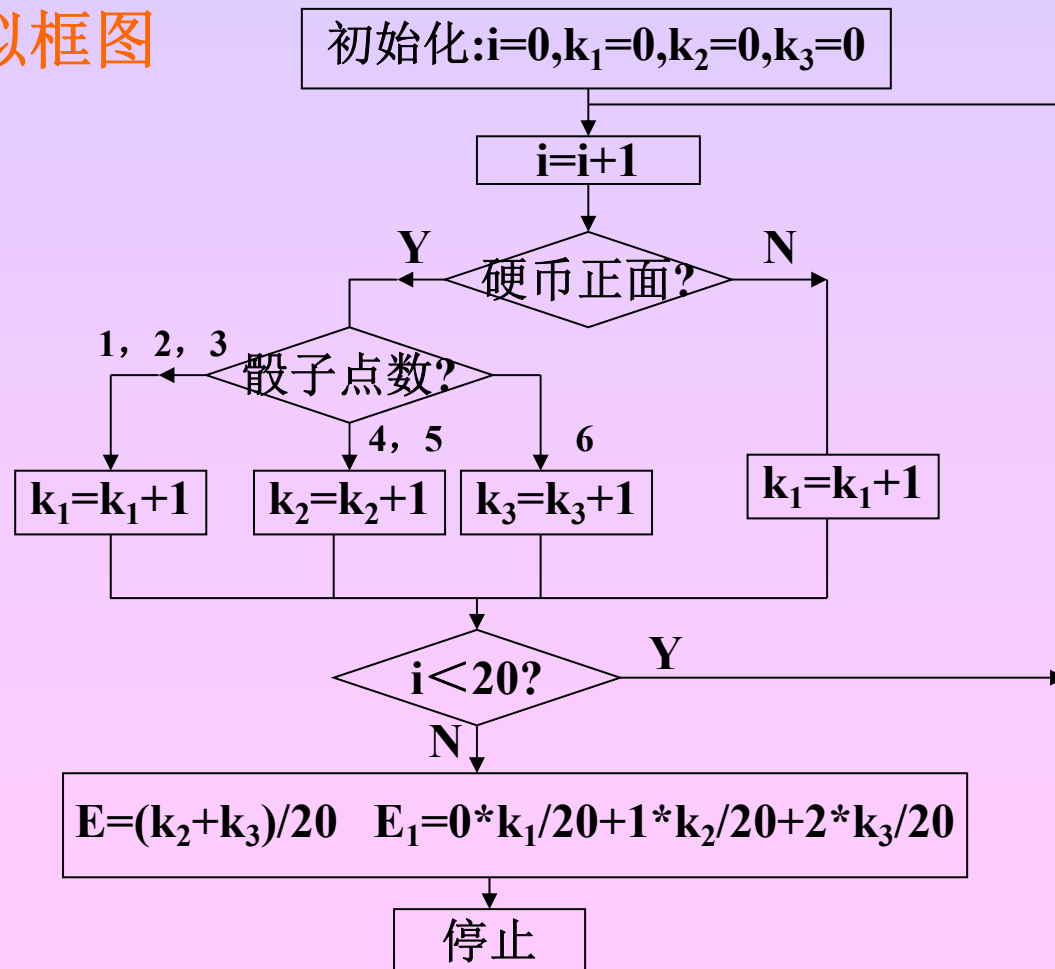
出现 6 点： 则认为击毁敌人两门火炮



2. 符号假设

i : 要模拟的打击次数; k_1 : 没击中敌人火炮的射击总数;
 k_2 : 击中敌人一门火炮的射击总数; k_3 : 击中敌人两门火炮的射击总数.
 E : 有效射击比率; E_1 : 20次射击平均每次毁伤敌人的火炮数.

3. 模拟框图



4. 模拟结果

试验 序号	投硬币 结 果	指示 正确	指 示 不正确	掷骰子 结 果	消灭敌人火炮数		
					0	1	2
1	正	√		4		√	
2	正	√		4		√	
3	反		√		√		
4	正	√		1	√		
5	正	√		2	√		
6	反		√		√		
7	正	√		3	√		
8	正	√		6			√
9	反		√		√		
10	反		√		√		



试验 序号	投硬币 结 果	指示 正确	指 示 不正确	掷骰子 结 果	消灭敌人火炮数		
					0	1	2
1 1	正	√		2	√		
1 2	反		√		√		
1 3	正	√		3	√		
1 4	反		√		√		
1 5	正	√		6			√
1 6	正	√		4		√	
1 7	正	√		2	√		
1 8	正	√		4		√	
1 9	反		√		√		
2 0	正	√		6			√

从以上模拟结果可计算出：

$$E=7/20=0.35 \quad E_1 = 0 \times \frac{13}{20} + 1 \times \frac{4}{20} + 2 \times \frac{3}{20} = 0.5$$



5. 理论计算

设: $j = \begin{cases} 0 & \text{观察所对目标指示不正确} \\ 1 & \text{观察所对目标指示正确} \end{cases}$

A_0 : 射中敌方火炮的事件; A_1 : 射中敌方一门火炮的事件;

A_2 : 射中敌方两门火炮的事件.

则由全概率公式:

$$E = P(A_0) = P(j=0)P(A_0 | j=0) + P(j=1)P(A_0 | j=1)$$

$$= \frac{1}{2} \times 0 + \frac{1}{2} \times \frac{1}{2} = 0.25$$

$$P(A_1) = P(j=0)P(A_1 | j=0) + P(j=1)P(A_1 | j=1)$$

$$= \frac{1}{2} \times 0 + \frac{1}{2} \times \frac{1}{3} = \frac{1}{6}$$

$$P(A_2) = P(j=0)P(A_2 | j=0) + P(j=1)P(A_2 | j=1)$$

$$= \frac{1}{2} \times 0 + \frac{1}{2} \times \frac{1}{6} = \frac{1}{12}$$

$$E_1 = 1 \times \frac{1}{6} + 2 \times \frac{1}{12} \approx 0.33$$



6. 结果比较

理论计算和模拟结果的比较

项目 \ 分类	分类		
	无效射击	有效射击	平均值
模 拟	0 . 6 5	0 . 3 5	0 . 5
理 论	0 . 7 5	0 . 2 5	0 . 3 3

模拟结果与理论计算虽然结果不完全一致，可这反映了事件发生的随机性。只要多次实验求平均值，模拟值就会很接近理论值。



二、随机数的产生

1. 均匀分布的随机数及其产生

对随机现象进行模拟，实质上是要给出随机变量的模拟。也就是说利用计算机随机地产生一系列数值，它们的出现服从一定的概率分布，则称这些数值为随机数。

最常用的是在 $(0, 1)$ 区间内均匀分布的随机数，也就是我们得到的这组数值可以看作是 $(0, 1)$ 区间内均匀分布的随机变量的一组独立的样本值。

其它分布的随机数可利用均匀分布的随机数产生。



产生模拟随机数的计算机命令

在Matlab软件中，可以直接产生满足各种分布的随机数，命令如下：

1、产生 $m*n$ 阶(0,1)均匀分布的随机数矩阵：**rand (m, n)**

产生一个(0,1)均匀分布的随机数：**rand**

2、产生 $m*n$ 阶 $[a, b]$ 均匀分布 $U(a, b)$ 的随机数矩阵：

unifrnd (a, b, m, n)

产生一个 $[a, b]$ 均匀分布的随机数：**unifrnd(a, b)**



3、正态分布

randn(m,n): $m*n$ 阶 $N(0,1)$ 标准正态分布随机数矩阵

normrnd(μ, σ):

均值为 μ ，标准差为 σ 的正态分布的随机数

normrnd(μ, σ, m, n):

$m*n$ 阶均值为 μ ，标准差为 σ 的正态分布的随机数矩阵:



4. 指数分布

exprnd(μ ,m,n):

m*n 阶期望值为 μ 的指数分布的随机数矩阵

- 若连续型随机变量 X 的概率密度函数为 $f(x) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases}$
其中 $\lambda > 0$ 为常数，则称 X 服从**参数**为 λ 的指数分布。
- 指数分布的期望值为 $\frac{1}{\lambda}$
- 排队服务系统中顾客到达率为常数时的到达间隔、故障率为常数时零件的寿命都服从指数分布。
- 指数分布在排队论、可靠性分析中有广泛应用。



注意：Matlab中，产生参数为 λ 的指数分布的命令为 $\text{exprnd}(\frac{1}{\lambda})$.

例 顾客到达某商店的间隔时间服从参数为0.1的指数分布

指数分布的均值为 $1/0.1=10$ 。

指两个顾客到达商店的平均间隔时间是10个单位时间. 即平均10个单位时间到达1个顾客. 顾客到达的间隔时间可用 $\text{exprnd}(10)$ 模拟。



5. 泊松分布

poissrnd (λ, m, n):

$m \times n$ 阶参数为 λ 的泊松分布的随机数矩阵

- 设离散型随机变量 X 的所有可能取值为 $0, 1, 2, \dots$, 且取各个值的概率为

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}, k = 0, 1, 2, \dots,$$

其中 $\lambda > 0$ 为常数, 则称 X 服从参数为 λ 的泊松分布。

- 泊松分布的期望值为 λ
- 泊松分布在排队系统、产品检验、天文、物理等领域有广泛应用。



指数分布与泊松分布的关系:

- 如相继两个事件出现的间隔时间服从参数为 λ 的指数分布, 则在单位时间间隔内事件出现的次数服从参数为 λ 的泊松分布. 即单位时间内该事件出现 k 次的概率为:

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}, k = 0, 1, 2, \dots,$$

反之亦然。

例 (1) 顾客到达某商店的间隔时间服从参数为0.1的指数分布 \Leftrightarrow (2) 该商店在单位时间内到达的顾客数服从参数为0.1的泊松分布

(1) 指两个顾客到达商店的平均间隔时间是10个单位时间. 即平均10个单位时间到达1个顾客.

(2) 指一个单位时间内平均到达0.1个顾客



例2 敌坦克分队对我方阵地实施突袭，其到达规律服从泊松分布，平均每分钟到达 4 辆. （1）模拟敌坦克在 3 分钟内到达目标区的数量，以及在第 1、2、3 分钟内各到达几辆坦克. （2）模拟在3分钟内每辆敌坦克的到达时刻。

（1）用**poissrnd(4)**进行模拟

（2）坦克到达的间隔时间应服从参数为4的指数分布，用**exprnd（1/4）**模拟。



2. 随机变量的模拟

利用均匀分布的随机数可以产生具有任意分布的随机变量的样本，从而可以对随机变量的取值情况进行模拟。

(1) 离散型随机变量的模拟

设随机变量 X 的分布律为 $P(X=x_i)=p_i, i=1, 2, \dots$, 令 $p^{(0)}=0, p^{(n)}=\sum p_i, n=1, 2, \dots$, 将 $p^{(n)}$ 作为分点，把区间 $(0, 1)$ 分为一系列小区间 $(p^{(n-1)}, p^{(n)})$. 对于均匀的随机变量 $R \sim U(0,1)$, 则有

$$P(p^{(n-1)} < R \leq p^{(n)}) = p^{(n)} - p^{(n-1)} = p_n, n=1, 2, \dots,$$



由此可知，事件 $(p^{(n-1)} < R \leq p^{(n)})$ 和事件 $(X = x_n)$ 有相同的发生的概率。因此我们可以用随机变量 R 落在小区间内的情况来模拟离散的随机变量 X 的取值情况。

具体执行的过程是：每产生一个 $(0,1)$ 上均匀分布的随机数 r ，若 $p^{(n-1)} < r \leq p^{(n)}$ 则理解为发生事件“ $X = x_n$ ”。



例3 随机变量 $x = \{0, 1, 2\}$ 表示每分钟到达超市收款台的人数，有分布列

$$x_k \quad 0 \quad 1 \quad 2$$

$$p_k \quad 0.4 \quad 0.3 \quad 0.3$$

模拟十分钟内顾客到达收款台的状况.



- 用MATLAB模拟随机事件
- **r=rand(1,10);**
- **for i=1:10;**
- **if r(i)<0.4**
- **n(i)=0;**
- **elseif 0.4<=r(i)&r(i)<0.7**
- **n(i)=1;**
- **else n(i)=2;**
- **end;**
- **end**



- $\gg r$
- $r = 0.5678 \quad 0.7942 \quad 0.0592 \quad 0.6029 \quad 0.0503$
- $0.4565 \quad 0.0185 \quad 0.8214 \quad 0.4447 \quad 0.6154$
- $\gg n$
- $n = 1 \quad 2 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 2 \quad 1 \quad 1$
- $\gg r$
- $r = 0.2311 \quad 0.6068 \quad 0.4860 \quad 0.8913 \quad 0.7621$
- $0.7919 \quad 0.9218 \quad 0.7382 \quad 0.1763 \quad 0.4057$
- $\gg n$
- $n = 0 \quad 1 \quad 1 \quad 2 \quad 2 \quad 2 \quad 2 \quad 2 \quad 0 \quad 1$



$$G(x) = \begin{cases} 0 & x < 0 \\ x & 0 < x \leq 1 \\ 1 & 1 < x \end{cases}$$

(2) 连续型随机变量的模拟

具有给定分布的连续型随机变量可以利用在区间(0, 1)上均匀分布的随机数来模拟. 最常用的方法是反函数法.

结论: 若随机变量Y有连续的分布函数F(y), 而X是区间(0, 1)上均匀分布的随机变量, 令 $Z = F^{-1}(X)$, 则Z与Y有相同的分布.

证明: $F_Z(z) = P\{Z < z\} = P\{F^{-1}(X) < z\} = P\{X < F(z)\}$
 $= G(F(z))$ (G是X的分布函数)
 $= F(z)$



由此，若已知Y的概率密度为 $f(y)$ ，由 $Y=F^{-1}(X)$ 可得

$$X = F(Y) = \int_{-\infty}^y f(y)dy$$

如果给定区间(0, 1)上均匀分布的随机数 r_i ，则具有给定分布的随机数 y_i 可由方程中解出.

$$r_i = \int_{-\infty}^{y_i} f(y)dy$$

例4 当需要模拟服从参数为 λ 的指数分布时, 由

$$r_i = \int_0^{y_i} \lambda e^{-\lambda x} dy = 1 - e^{-\lambda y_i}$$

可得 $y_i = -(1/\lambda) \ln(1-r_i)$

简化为 $y_i = -(\ln r_i)/\lambda$.



例5 产生服从几何分布的随机数

$$f(x = k) = pq^{k-1}, \quad p + q = 1, k = 1, 2, \dots, n$$

$$F(x) = \sum_{i=1}^x pq^{i-1} = 1 - q^x = r$$

$$x = \ln(1 - r) / \ln q; \quad x = \ln r / \ln q$$

其它方法:

舍选法、组合法、近似法等



三、时间步长法

时间步长法：按照时间流逝的顺序，一步一步地对系统的活动进行仿真。在整个仿真过程中，时间步长固定不变。

基本思想：

在进行系统仿真的过程中，可以把整个过程分成许多相等的时间间隔，时间步长的长度可以根据实际问题分别取作秒, 分, 小时, 天等。程序中按照这个步长前进的时钟就是仿真的时钟。

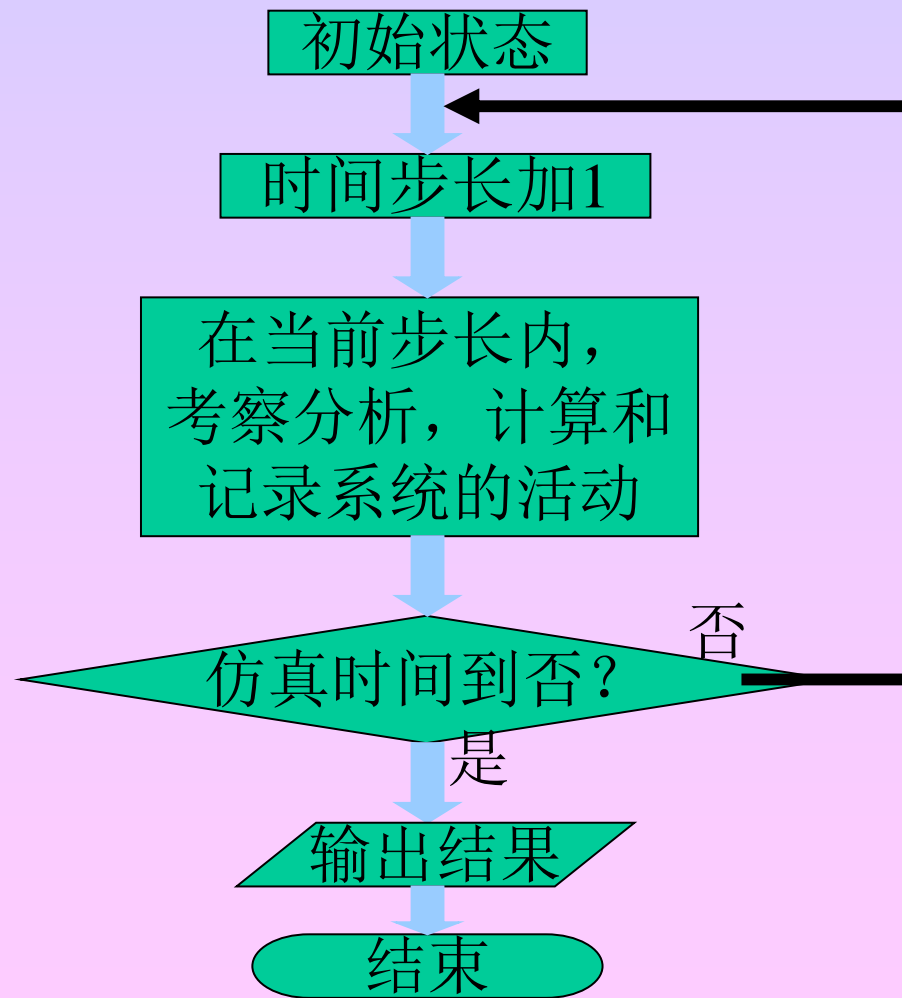


基本步骤:

首先选取对象系统的一个初始起点作为仿真时钟的零点，然后根据实际问题的需要，选定一个时间步长。于是从仿真时钟的零点开始，每推进一个时间步长就对系统的活动和状态按照预定的规则和目的进行考察、分析、计算和记录，直到预定仿真结束时刻为止。



时间步长法

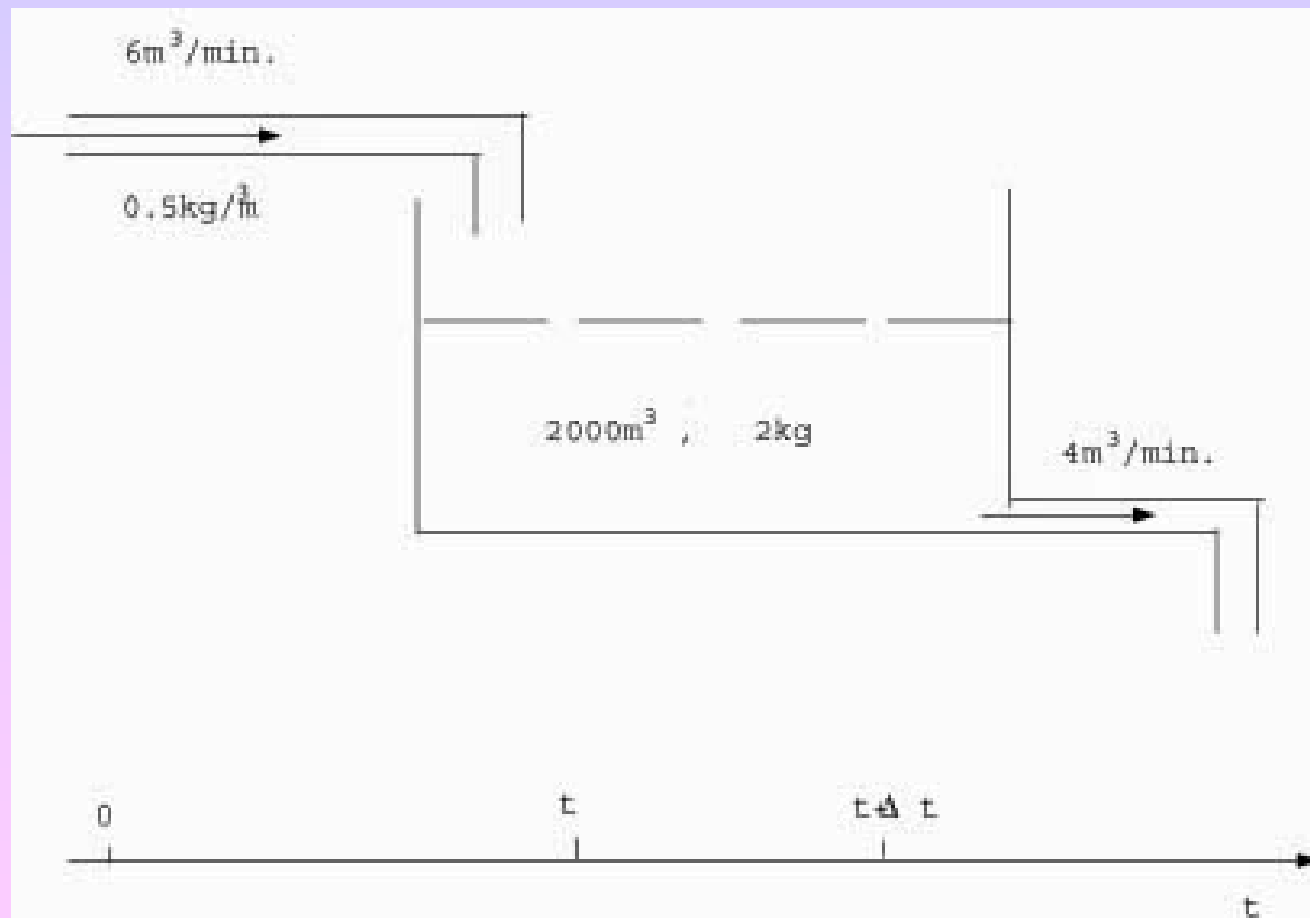


应用举例-池水含盐量问题

例6:池水含盐量问题

某水池有 2000m^3 水，其中含盐 2kg ，以每分钟 6m^3 的速率向水池内注入含盐率为 $0.5\text{kg}/\text{m}^3$ 的盐水，同时又以每分钟 4m^3 的速率从水池流出搅拌均匀的盐水。试用计算机仿真该水池内盐水的变化过程，并每隔 10min 计算水池中水的体积、含盐量和含盐率。欲使池中盐水的含盐率达到 $0.2\text{kg}/\text{m}^3$ ，需经过多少时间？





池水含盐量随时间变化的示意图



分析:系统中，实体是水，属性是水的体积、含盐量和含盐率，活动是水的注入与流出，由于注入和流出活动的作用，池中水的体积与含盐量、含盐率均随时间变化，初始时刻含盐率为 0.001kg/m^3 ，以后每分钟注入含盐率为 0.5 kg/m^3 的水 6m^3 ，流出混合均匀的盐水 4m^3 ，当池中水的含盐率达到 0.2kg/m^3 时，结束仿真过程。



为了能定量地考察系统在任一时刻的属性，引入下列记号：

注水速度： $WI=6 \text{ m}^3/\text{min}$

排水速度： $WO=4 \text{ m}^3/\text{min}$

注入水的含盐率： $SI=0.5 \text{ kg}/\text{m}^3$

最终含盐率： $SF=0.2 \text{ kg}/\text{m}^3$

T时刻水的体积： $VT\text{m}^3$

T时刻水的含盐量： $ST \text{ kg}$

T时刻水的含盐率： $SR=ST/VT \text{ kg}/\text{m}^3$



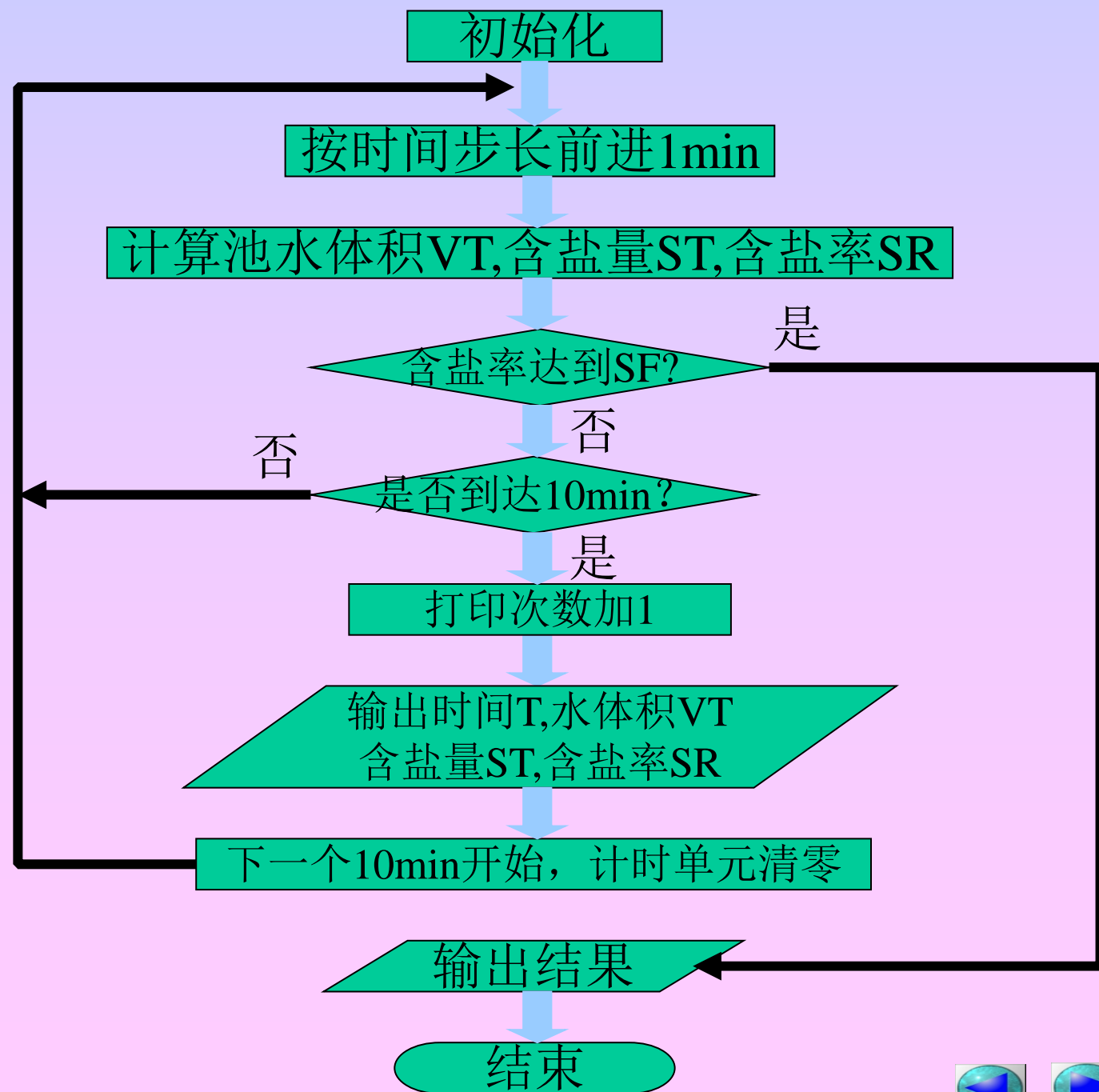
对于这样一个连续系统仿真时，必须在一系列离散时间点 $t_0 < t_1 < t_2 < \dots < t_n$ 上来进行考察，这些离散时间点之间的间隔 $T = t_i - t_{i-1} (i=1, 2, \dots, n)$ 就是时间步长．若取步长为1min时，就要每隔1min考察一次系统的状态特性，并作出相应的记录与分析．在本题中每隔1min池水的动态变化过程是：

每分钟水的体积增加	$6-4=2 \text{ m}^3$;
每分钟向池内注入盐	$6 \times 0.5 = 3 \text{ kg}$;
每分钟向池外流出盐	$4 \times SR \text{ kg}$;
每分钟池内增加盐	$3-4 \times SR \text{ kg}$.

取 $T=0$ 作为系统仿真的初始时刻，取 $T=0$ 时系统的属性为系统的初始状态．取池中盐水的含盐率达到 0.2kg/m^3 的时刻作为仿真的终止时刻．



池水含盐量仿真流程



池水含盐量仿真程序

%Matlab程序为

```
h=1;      s0=2;      w0=2000;      r0=s0/w0;
s(1)=s0+0.5*6*h-4*h*r0;
w(1)=w0+2*h;
r(1)=s(1)/w(1);      t(1)=h;
y(1)=(2000000+3000000*h+3000*h^2+h^3)/(1000+h)^2;
for i=2:200
    t(i)=i*h;
    s(i)=s(i-1)+0.5*6*h-4*h*r(i-1);
    w(i)=w(i-1)+2*h;
    r(i)=s(i)/w(i);
    y(i)=(2000000+3000000*t(i)+3000*t(i)^2+t(i)^3)/(1000+t(i))^2;
    m=floor(i/10);
    if i/10-m<0.1
        tm(m)=m;
        wm(m)= w(i),
        sm(m)=s(i);
        rm(m)=r(i);
    end
    if r(i)>0.2
        t02=i*h;
        r02=r(i);
        break
    end
end
[t02,r02]
[10*tm',sm',rm']
plot(t,s,'blue',t,y,'red')
```



池水含盐量仿真结果

时间T/min	池水体积VT/m ³	含盐量ST/kg	含盐率SR/kg/m ³
10	2020	31.69365	0.01569
20	2040	60.81008	0.02909
30	2060	89.37162	0.04338
40	2080	117.3996	0.05644
50	2100	144.9141	0.06901
60	2120	171.9346	0.08110
70	2140	198.4795	0.09275
80	2160	224.5663	0.10397
90	2180	250.2118	0.11478
100	2200	275.4319	0.12519
110	2220	300.242	0.13524
120	2240	324.6566	0.14494
130	2260	348.6897	0.15429
140	2280	372.3546	0.16331
150	2300	395.6642	0.17203
160	2320	418.6306	0.18044
170	2340	441.2655	0.18857
180	2360	463.5802	0.19643
185	2370	474.6208	0.20026



在这个问题中，池水的体积和含盐量、含盐率都是随时间连续变化的．故这是一个连续系统的仿真问题，我们是把这个连续过程离散化后用时间步长法进行仿真的．我们还可以用微分方程建立这个问题的数学模型，并求出它的解析解，这个解析解就是问题的精确解．有兴趣可以按照这一思路求出该问题的精确解，考察相应时刻精确解与仿真解的差异．也可进一步调整前面仿真过程的时间步长，通过与精确解的比较来研究时间步长的大小对仿真精度的影响．

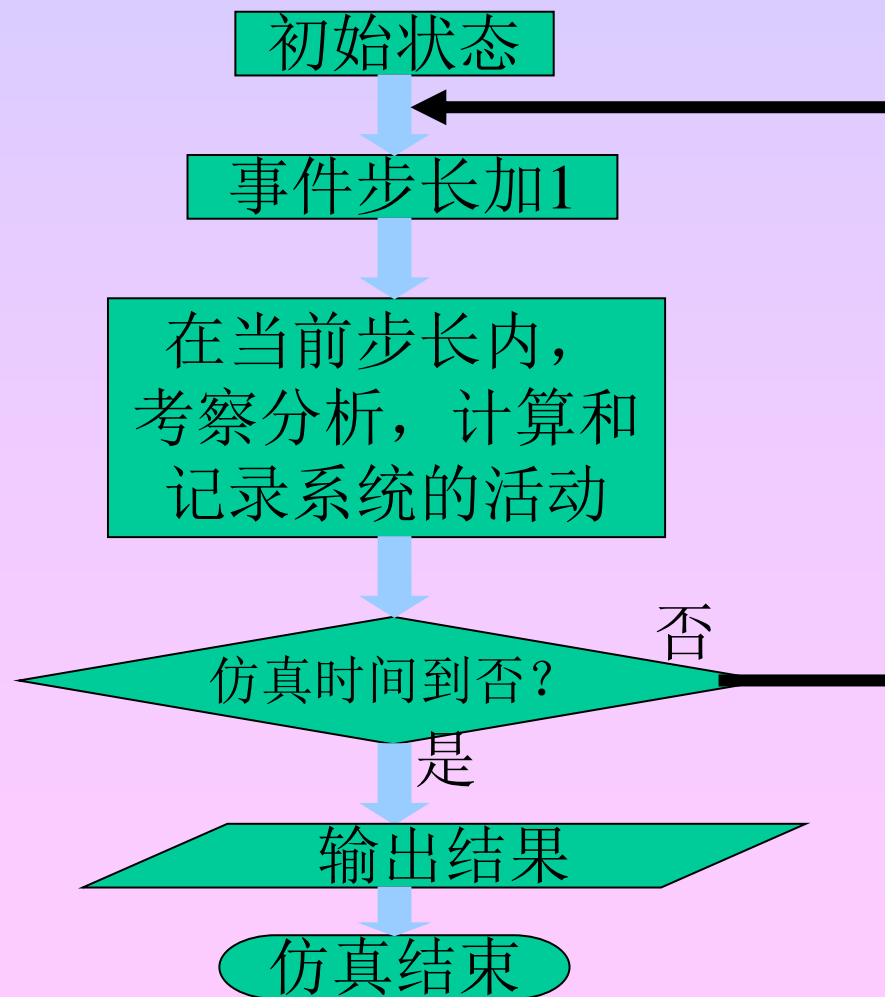


四、事件步长法

事件步长法： 是以事件发生的时间为增量，按照事件发生的时间顺序，一步一步地对系统的行为进行仿真，直到预定的时间结束为止。



事件步长法



事件步长法与时间步长法的主要区别是：

(1)事件步长法与时间步长法都是以时间为增量来考察系统状态的变化。但在时间步长法中，仿真时钟以等步长前进，而在事件步长法中，仿真时钟的步长取决于事件之间的间隔。

(2)时间步长法在一个步长内，认为系统所处的状态相同，因而所选步长的大小将影响仿真的精度。而在事件步长法中，每个事件的发生均有确切的时刻，不需要人为地选取步长，步长的大小对仿真精度影响较小。



(3) 时间步长法每前进一个步长就要对整个系统进行一次全面考察，即使状态没有发生变化时也要扫描，而事件步长法只是在某一事件点上判断和比较事件是否出现。

因此，一般地讲，当判断比较的数目较大时，用时间步长法可以节省用机时间，而当相继两个事件出现的平均间隔较长时，更适合于用事件步长法。



例7 收款台前的排队过程的仿真

假设：

- 1、顾客的到达收款台是随机的，平均时间间隔为0.5分钟，即间隔时间服从 $\lambda=2$ 的指数分布。
- 2、对不同的顾客收款和装袋的时间服从正态分布 $N(1, 1/3)$ 。

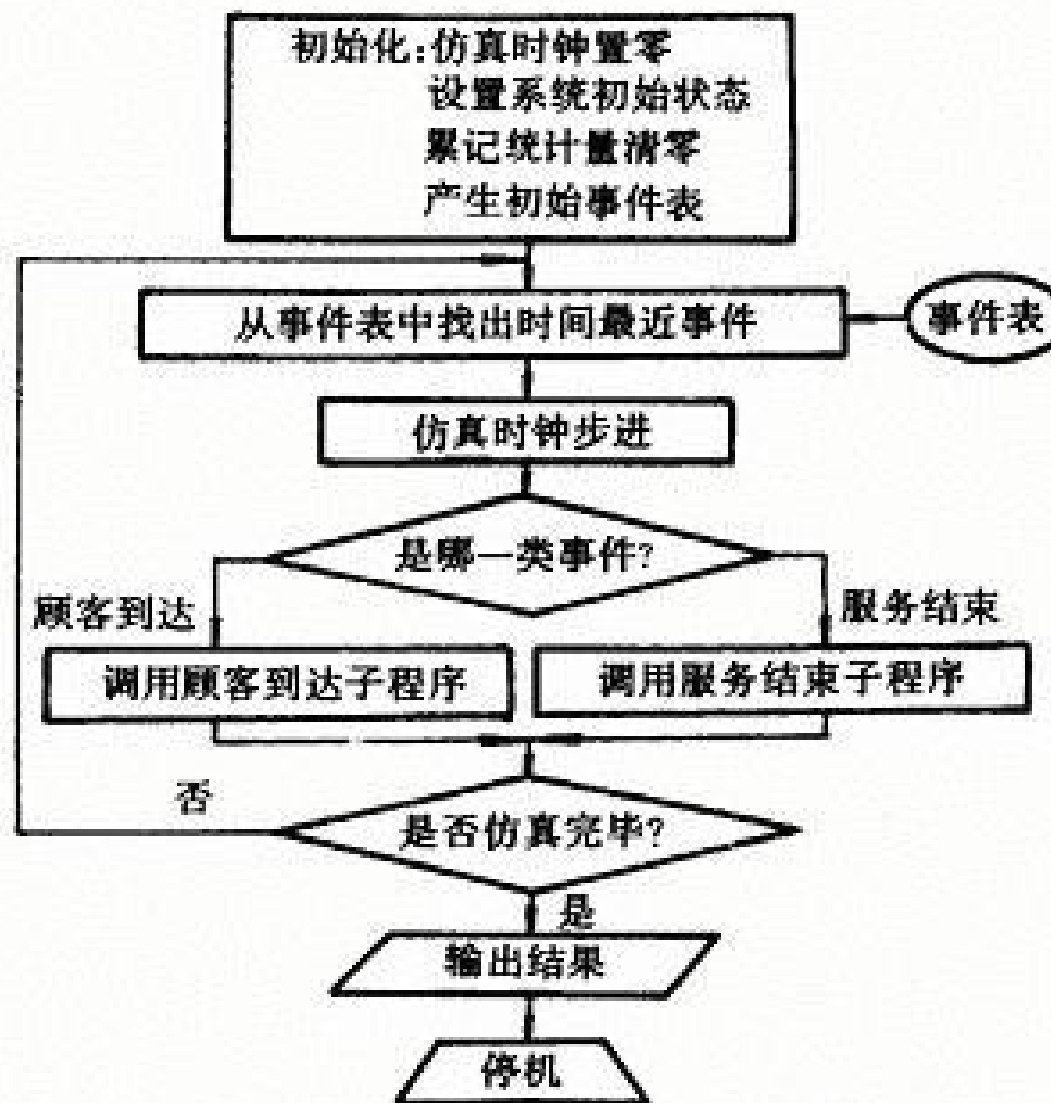
模拟20位顾客到收款台前的排队情况

关心问题：

每个顾客的平均等待时间，
队长，
服务员的工作效率。



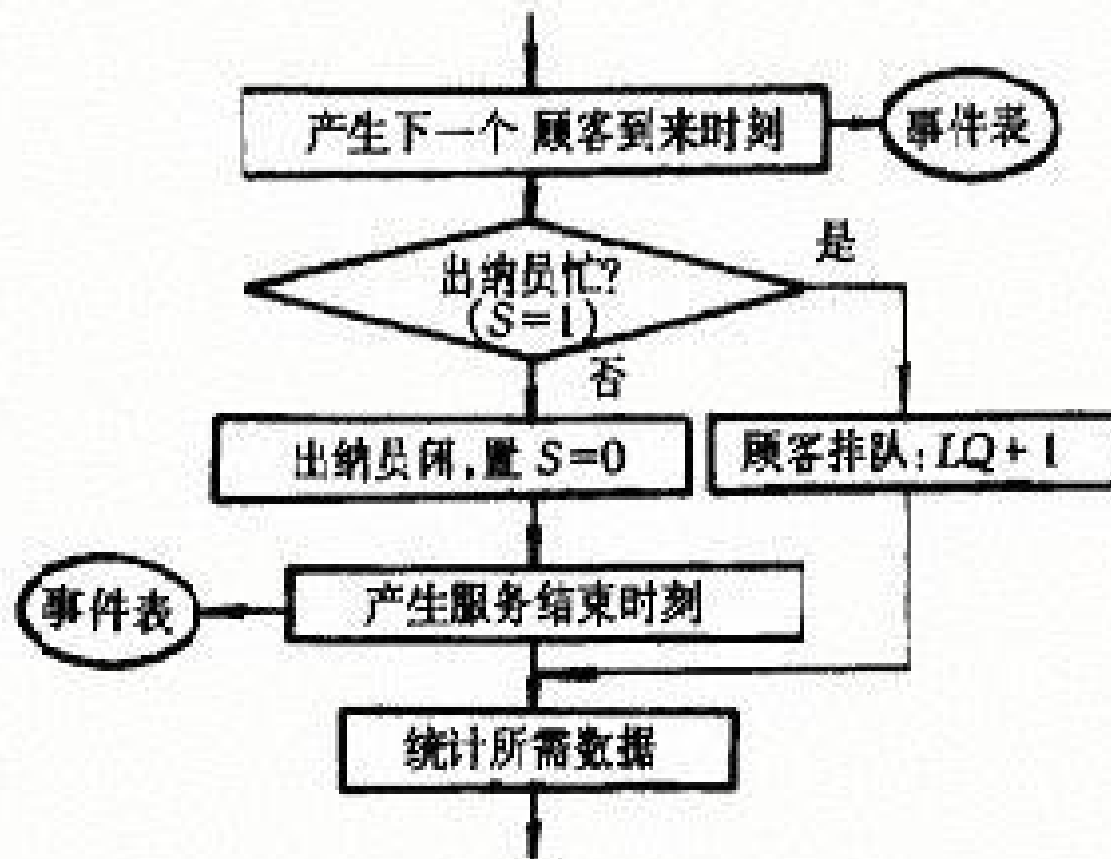
收款排队系统主控程序图



收款排队系统主控程序框图



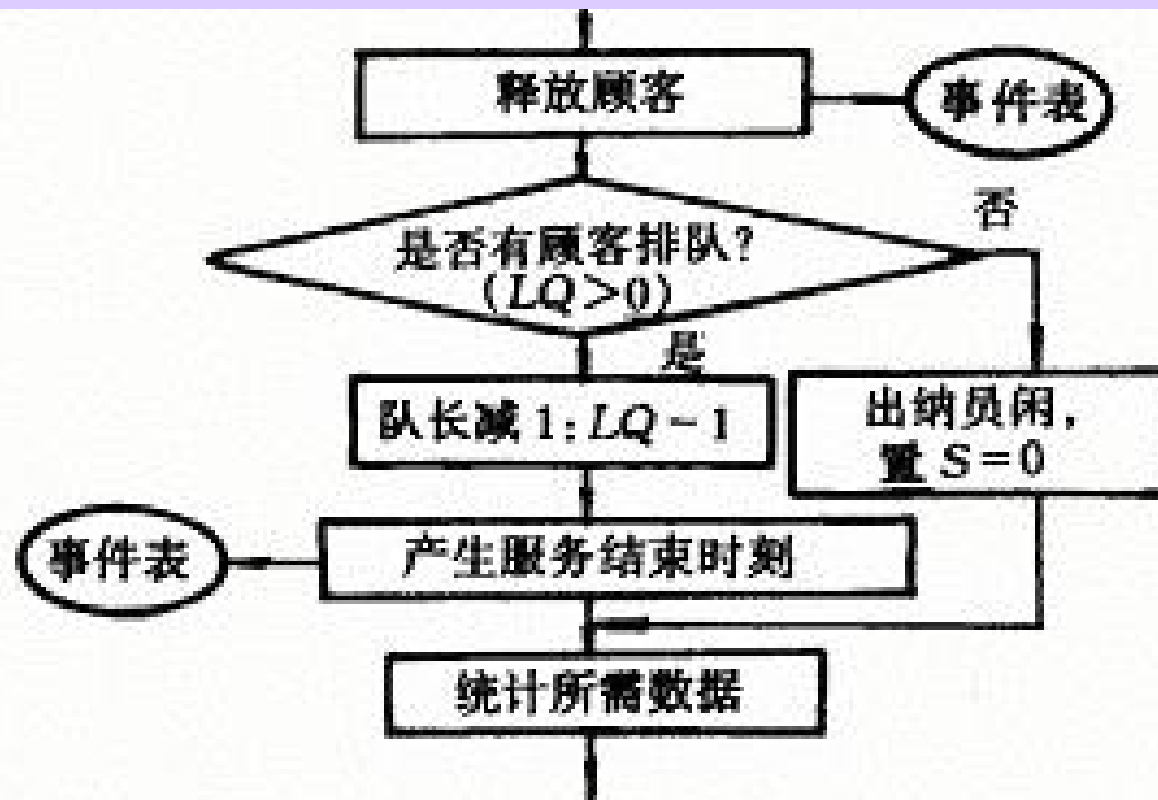
顾客到达子程序图



顾客到达子程序框图



服务结束子程序图



服务结束子程序框图



假设 $t(i)$: 第 i 位顾客到达时刻,

$t2(i)$:第 i 位顾客受到的服务时间(随机变量),

$T(i)$: 第 i 位顾客离去时刻.

将第 i 位顾客到达作为第 i 件事发生;

$$t(i+1) - t(i) = r(i) \text{ (随机变量)}$$

平衡关系:

当 $t(i+1) \geq T(i)$ 时, $T(i+1) = t(i+1) + t2(i+1)$;

否则, $T(i+1) = T(i) + t2(i+1)$



五、Monte Carlo方法

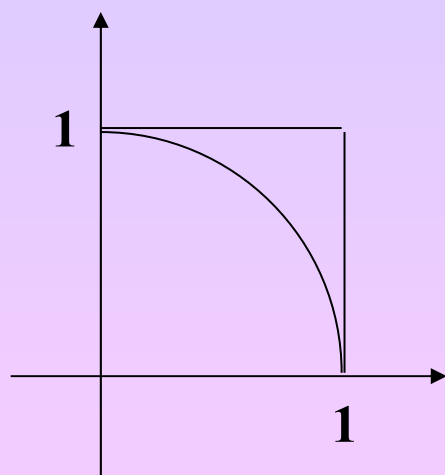
Monte Carlo方法亦称为随机模拟（**Random simulation**）方法，有时也称作随机抽样技术或统计试验方法。此方法对研究的系统进行随机观察抽样，通过对样本值的观察统计，求得所研究系统的某些参数。

基本思想：

当试验次数充分多时，某一事件出现的频率近似于该事件发生的概率。



例8：用蒙卡模拟求圆周率 π 的估计值。



设二维随机变量 (X,Y) 在正方形服从均匀分布，

则 (X,Y) 落在四分之一圆内的概率为 $P\{X^2+Y^2\leq 1\} = \pi/4$

现产生 n 对二维随机点 (x_i, y_i) , x_i, y_i 是区间 $[0,1]$ 上均匀分布的随机数，若其中有 k 对满足 $x_i^2 + y_i^2 \leq 1$ ，即相当于做 n 次投点试验，其中有 k 点落在 $1/4$ 圆内。计算落入 $1/4$ 圆内的频率为 k/n ，由大数定律，事件发生的频率依概率收敛于发生的概率，可得圆周率 π 的估计值为 $4k/n$ 。次数越多，精度越大。



%例8 用蒙特卡罗方法模拟求圆周率 π 的近似值。

clear

k=0; %k用于随机点落在1/4圆内的计数

for j=1:1000000 %样本个数取为N=100000

a=rand(1,2); %生成区间(0,1)上的均匀分布随机数作取样值

if a(1)^2+a(2)^2<=1 %检查随机数是否满足:

k=k+1;

end

end

PI=4*k/j %计算近似值



应用举例-报童的策略

例9 报童每天清晨从报社购进报纸零售, 晚上将没有卖掉的报纸退回. 每份报纸的购进价为1.3元, 零售价为2元, 退回价为0.2元. 报童售出一份报纸赚0.7元, 退回一份报纸赔1.1元. 报童每天如果购进的报纸太少, 不够卖时会少赚钱, 如果购得太多卖不完时要赔钱. 试为报童筹划每天应如何确定购进的报纸数使得收益最大. 报纸每捆10张, 只能整捆购买, 报纸可以分为3种类型的新闻日: 好、一般、差, 它们的概率分别为0.35, 0.45和0.2, 在这些新闻日中每天对报纸的需求分布的统计结果如下图:



需求量	好新闻的需求概率	一般新闻的需求概率	差新闻的需求概率
40	0.03	0.10	0.44
50	0.05	0.18	0.22
60	0.15	0.40	0.16
70	0.20	0.20	0.12
80	0.35	0.08	0.06
90	0.15	0.04	0.00
100	0.07	0.00	0.00

试确定每天报童应该订购的报纸数量



解：我们通过计算机仿真来解决此问题。最优策略应该是每天的利润最大。

利润=销售收入-报纸成本+残值

这是一个随机现象的计算机仿真问题，

故先确定各种情况的随机数的对应关系。

新闻日和需求量对应的随机数分别如下面两个表格所示

新闻种类	出现概率	对应的随机数区间
好新闻	0.35	(0, 0.35)
一般新闻	0.45	[0.35, 0.80)
差新闻	0.20	[0.80, 1)



需求量	好新闻的随机数区间	一般新闻的随机数区间	差新闻的随机数区间
40	(0.00,0.03]	(0.00,0.10]	(0.00,0.44]
50	[0.03,0.08)	[0.10,0.28)	[0.44,0.66)
60	[0.08,0.23)	[0.28,0.68)	[0.80,1.00)
70	[0.23,0.43)	[0.68,0.88)	[0.80,1.00)
80	[0.43,0.78)	[0.88,0.96)	[0.80,1.00)
90	[0.78,0.93)	[0.96,1.00)	[100, 102]
100	[0.93,1.00]	[8, 100]	[104, 125]

计算机仿真的流程:

- 1) 令每天的报纸订购数变化, 40--100;
- 2) 让时间从1开始变化(循环)到365;
- 3) 产生新闻种类的随机数, 确定当天的新闻类型;
- 4) 产生需求量随机数, 确定当天的报纸需求量;
- 5) 计算当天的收入, 计算累积利润,
- 6) 比较得出最优定货量。



Matlab编程(baotong程序)

```
x1=rand(365,1);
x2=rand(365,1);
for n=4:10
    paper=n*10;
    sb(n)=0;
    for i=1:365
        if x1(i)<0.35
            if x2(i)<0.03
                news=40;
            elseif x2(i)<0.08
                news=50;
            elseif x2(i)<0.23
                news=60;
            elseif x2(i)<0.43
                news=70;
```

```
elseif x2(i)<0.78
            news=80;
        elseif x2(i)<0.93
            news=90;
        else news=100;
    end
elseif x1(i)<0.8
    if x2(i)<0.10
        news=40;
    elseif x2(i)<0.28
        news=50;
    elseif x2(i)<0.68
        news=60;
    elseif x2(i)<0.88
        news=70;
```




```
elseif x2(i)<0.96
    news=80;
else news=90;
end
else
    if x2(i)<0.44
        news=40;
    elseif x2(i)<0.66
        news=50;
    elseif x2(i)<0.82
        news=60
    elseif x2(i)<0.94
        news=70;
```

```
    else news=80;
    end
end

    if paper>=news
        sale=news;
        remand=paper-news;
    else
        sale=paper;remand=0;
    end
    sb(n)=sb(n)+2*sale-
        1.3*paper+0.2*remand;
    end
end
```



```
optnews=40;  
optmoney=sb(4);  
[40,sb(4)/365]  
for n=5:10  
    if sb(n)>=optmoney  
        optnews=n*10;  
        optmoney=sb(n);  
    end  
    [n,sb(n)/365]  
end  
[optnews,optmoney,optmoney/365]
```

经过计算机仿真后得到最优购货量是每天60份，
平均每天利润34.4元。



应用举例-赶火车仿真

例10 赶火车过程仿真

- 一列火车从A站经过B站开往C站，某人每天赶往B站乘这趟火车。已知火车从A站到B站运行时间为均值30分钟、标准差为2分钟的正态随机变量。火车大约在下午1点离开A站。离开时刻的频率分布为

出发时刻 (T)	1: 00	1: 05	1: 10
频率	0.7	0.2	0.1



这个人到达B站时的频率分布为：

到达时刻 (T)	1:28	1:30	1:32	1: 34
频率	0.3	0.4	0.2	0.1

用计算机仿真火车开出、火车到达B站、这个人到达B站情况，并给出能赶上火车的仿真结果。

引入以下变量：

T1 火车从A站开出的时刻；

T2 火车从A站运行到B站所需要的时间；

T3 此人到达B站的时刻；

T1(分)	0	5	10
P (频率)	0.7	0.2	0.1



开车时间的仿真测试

```
s1=0; s2=0; s3=0;  
x=rand(10000, 1);  
for i=1:10000  
    if x(i)<0.7  
        s1=s1+1;  
    end  
    if x(i)>0.9  
        s3=s3+1;  
    end  
end  
[s1/10000, 1-s1/10000-s3/10000, s3/10000]
```



人到达时刻仿真测试

T2 (分)	28	30	32	34
P(频率)	0.3	0.4	0.2	0.1

```
s1=0; s2=0; s3=0;s4=0;  
x=rand(10000,1);  
for i=1:10000  
    if x(i)<0.3  
        s1=s1+1;  
    elseif x(i)<0.7  
        s2=s2+1;  
    else  
        if x(i)<0.9  
            s3=s3+1;  
        else  
            s4=s4+1;  
        end  
    end  
end  
end  
[s1/10000, s2/10000,s3/10000,s4/10000]
```



火车运行时间的仿真测试

```
x=randn(10000,1);  
for i=1:10000  
    y(i)=30+2*x(i);  
end
```

赶上火车的仿真结果

```
s=0;  
x1=rand(10000,1);  
x2=rand(10000,1);  
x3=randn(10000,1);  
for i=1:10000  
    if x1(i)<0.7  
        T1=0;  
    elseif x1(i)<0.9  
        T1=5;  
    else  
        T1=10;  
    end  
    T2=30+2*x3(i);
```

```
    if x2(i)<0.3  
        T3=28;  
    elseif x2(i)<0.7  
        T3=30;  
    elseif x2(i)<0.9  
        T3=32;  
    else  
        T3=34;  
    end  
    if T3<T1+T2  
        s=s+1;  
    end  
end  
[s/10000]
```



用蒙特卡洛法解非线性规划问题

对于非线性规划问题:

$$\begin{aligned} \min & f(X) \quad X \in E^n \\ \text{s. t.} \quad & g_i(X) \geq 0 \quad i=1, 2, \dots, m \\ & a_j \leq x_j \leq b_j \quad j=1, 2, \dots, n \end{aligned}$$

用蒙特卡洛法求解的基本思想是: 在估计的区域

$$\{(x_1, x_2, \dots, x_n) \mid x_j \in [a_j, b_j], j=1, 2, \dots, n\}$$

内随机取若干实验点, 然后从试验点中找出可行点, 再从可行点中选择最小点.



基本假设 试验点的第 j 个分量 x_j 服从 $[a_j, b_j]$ 内的均匀分布.

符号假设

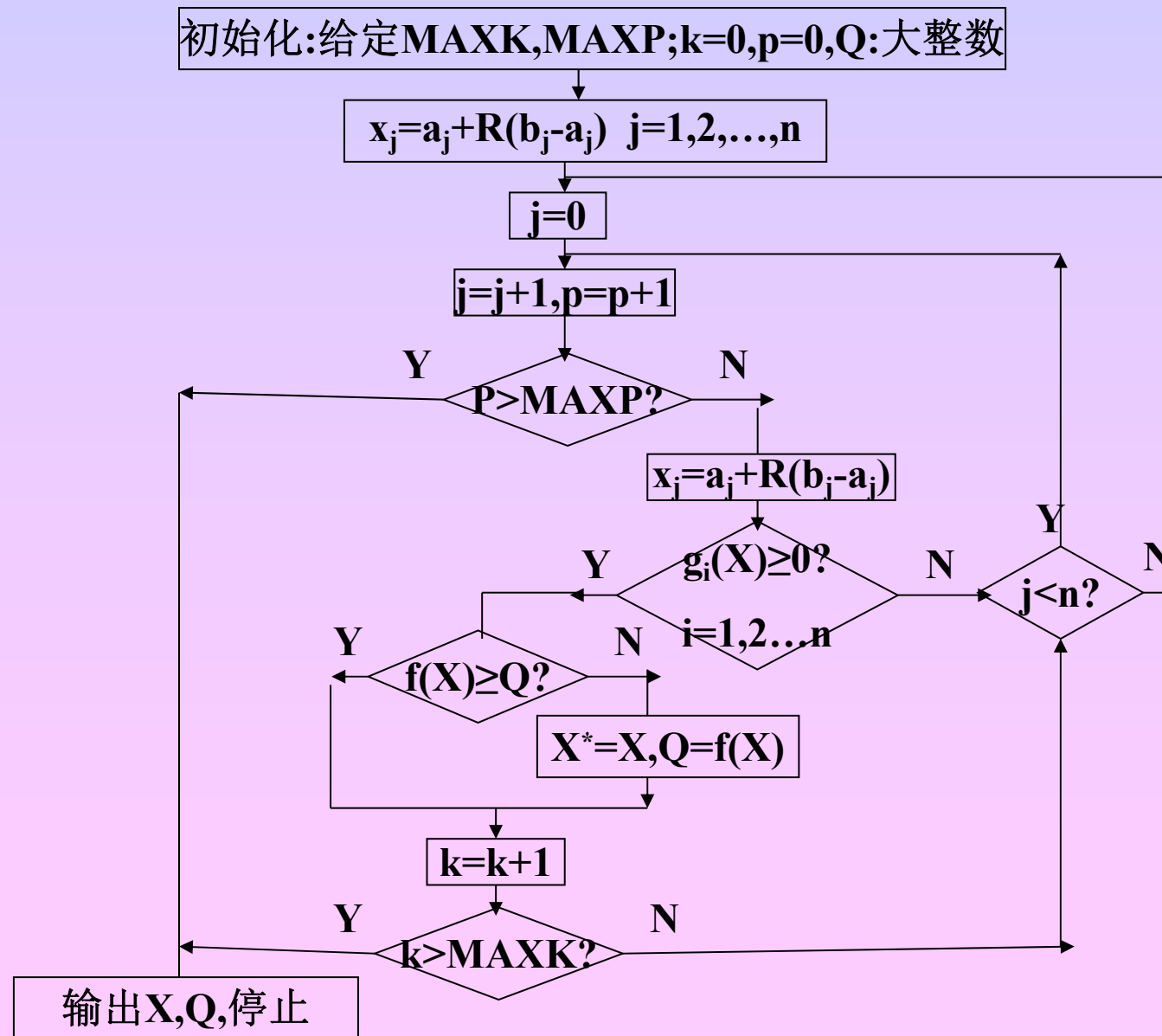
P: 试验点总数; MAXP: 最大试验点总数;
K: 可行点总数; MAXK: 最大可行点数;
 X^* : 迭代产生的最优点;
Q: 迭代产生的最小值 $f(X^*)$, 其初始值为计算机所能表示的最大数.

求解过程

先产生一个随机数作为初始试验点, 以后则将上一个试验点的第 j 个分量随机产生, 其它分量不变而产生一新的试验点. 这样, 每产生一个新试验点只需一个新的随机数分量. 当 $K > \text{MAXK}$ 或 $P > \text{MAXP}$ 时停止迭代.



框图



$$\text{例11 } \max z = -2x_1^2 - x_2^2 + x_1x_2 + 8x_1 + 3x_2$$

$$\text{s. t } 3x_1 + x_2 = 10$$

$$x_1 > 0$$

$$x_2 > 0$$

在Matlab软件包中编程，共需三个M—文件:randlp.m, mylp.m, lpconst.m. 主程序为randlp.m.

% mylp.m

function z=mylp(x)

%目标函数

z=2*x(1)^2+x(2)^2-x(1)*x(2)-8*x(1)-3*x(2); %转化为求最小值问题

% lpconst.m

function lpc=lpconst(x)

%约束条件

if 3*x(1)+x(2)-10<=0.5 & 3*x(1)+x(2)-10>=-0.5 %约束条件的误差为±0.5

lpc=1;

else

lpc=0;

end



```
% randlp.m
```

```
%function [sol,r1,r2]=randlp(a,b,n)      %随机模拟解非线性规划
```

```
debug=1;
```

```
a=0;                                     %试验点下界
```

```
b=10;                                    %试验点上界
```

```
n=1000;                                  %试验点个数
```

```
r1=unifrnd(a,b,n,1);                    %n*1阶的 [a,b] 均匀分布随机数矩阵
```

```
r2=unifrnd(a,b,n,1);
```

```
sol=[r1(1) r2(1)];
```

```
z0=inf;
```

```
for i=1:n
```

```
    x1=r1(i);
```

```
    x2=r2(i);
```

```
    lpc=lpconst([x1 x2]);
```

```
    if lpc==1
```

```
        z=mylp([x1 x2]);
```

```
        if z<z0
```

```
            z0=z;
```

```
            sol=[x1 x2];
```

```
        end
```

```
    end
```

```
end
```

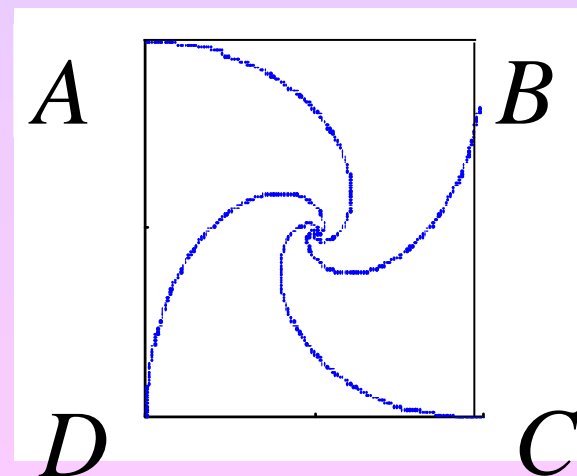
```
sol
```



其他：连续系统模拟实例——追逐问题

状态随时间连续变化的系统称为连续系统。对连续系统的计算机模拟只能是近似的，只要这种近似达到一定的精度，也就可以满足要求。

例12 追逐问题：如图,正方形ABCD的四个顶点各有一人.在某一时刻,四人同时出发以匀速 $v=1$ 米/秒按顺时针方向追逐下一人,如果他们始终保持对准目标,则最终按螺旋状曲线于中心点O.试求出这种情况下每个人的行进轨迹.



求解过程:

1. 建立平面直角坐标系: $A(x_1, y_1)$, $B(x_2, y_2)$, $C(x_3, y_3)$, $D(x_4, y_4)$.
2. 取时间间隔为 Δt , 计算每一点在各个时刻的坐标.

设某点在 t 时刻的坐标为: (x_i, y_i)

则在 $t + \Delta t$ 时刻的坐标为: $(x_i + v\Delta t \cos \alpha, y_i + v\Delta t \sin \alpha)$

其中 $\cos \alpha = \frac{x_{i+1} - x_i}{d}$ $\sin \alpha = \frac{y_{i+1} - y_i}{d}$

$$d = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$$

3. 取足够小的 ε , $d < \varepsilon$ 时结束算法.
4. 对每一个点, 连接它在各时刻的位置, 即得所求运动轨迹.



计算程序:

```
v=1;
```

```
dt=0.05;
```

```
x=[0 0 10 10];
```

```
y=[0 10 10 0];
```

```
for i=1:4
```

```
    plot(x(i),y(i),'.'),hold on
```

```
end
```

```
d=20;
```

```
while(d>0.1)
```

```
    x(5)=x(1);y(5)=y(1);
```

```
    for i=1:4
```

```
        d=sqrt((x(i+1)-x(i))^2+(y(i+1)-y(i))^2);
```

```
        x(i)=x(i)+v*dt*(x(i+1)-x(i))/d;
```

```
        y(i)=y(i)+v*dt*(y(i+1)-y(i))/d;
```

```
        plot(x(i),y(i),'.'),hold on
```

```
    end
```

```
end
```

