数字水印与信息隐藏

同济大学计算机系
钟计东
EMAIL: zhongjidong@tongji.edu.cn

课件存放在以下邮箱

EMAIL: tongji_zhong@163.com

密码：tongji_cs

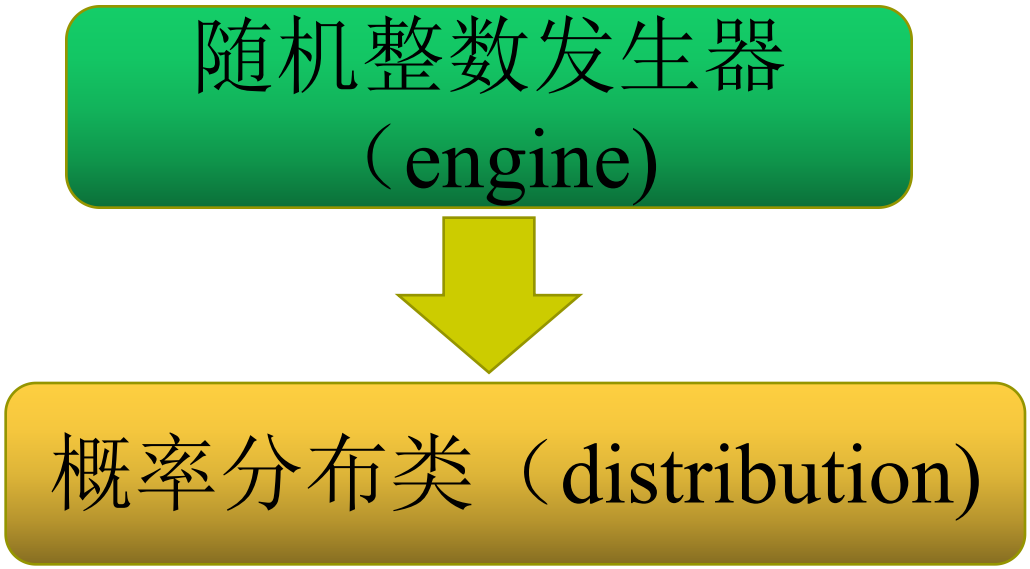
实验一

伪随机数发生器

伪随机数发生器

- 广泛用途：
  - 实验仿真
  - 游戏
  - 安全

C++产生随机数的方式



例：
```
mt19937_64 gen(100);
uniform_real_distribution<double> dis(0, 1);
double ran = dis(gen);
```

均匀分布随机数

- 可以直接利用产生[0, 1]均匀分布的类（C++标准库中有这样的类）：
  uniform_real_distribution
  - 产生的是[a, b)之间的均匀分布的随机数（不包括b。
  - 需要一个整数的引擎(Engine)
    - Mersenne Twister(64位): mt19937_64，构造函数需要一个种子（可以作为一个密钥），可以自己输入，当然种子也可以使用seed成员函数设置

随机整数发生器

- C++提供以下整数发生器，通常这些发生器产生的是一定区间内的伪随机的整数。
  - linear_congruential_engin：线性同余发生器
  - mersenne_twister_engine：日本人发明，http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html
  - subtract_with_carry_engine

随机整数发生器

- C++还提供一些利用上面基本发生器产生随机整数的类：
  - discard_block_engine
  - independent_bits_engine
  - shuffle_order_engine

随机整数发生器

- C++还提供一些模板实例化的发生器：
  - linear_congruential_engine的实例化：default_random_engine
  - mersenne_twister_engine的实例化：mt19937，mt19937_64

需要注意的问题

- 不建议使用rand()函数，因为它产生的数是16位的

需要注意的问题

- 假设rand()产生[0,1500]之间的均匀整数(每一个数概率为1/1501)
  - 假如你试图这样产生[0,1]之间的均匀分布X = (rand()%1001)/1000.0，这样做存在问题：
    - 精度不好（精度0.001, 更好的精度可以是1/1500
    - 均匀性不好（下面公式可以看到）

$$P(X = x) = \begin{cases} 2/1501 & \text{if } 0 \le x \le 0.499 \\ 1/1501 & \text{if } 0.5 \le x \le 1.0 \end{cases}$$

需要注意的问题

- 避免log 0
  - uniform_real_distribution可以产生[0, 1)之间的均匀分布的随机数
  - uniform_real_distribution<double> gen(0, 1);
  - log(1- gen())就不会遇到log 0问题

广义高斯分布CDF理论值

广义高斯分布GGD(c, β)的CDF理论值如下（参考讲义），其中F_E为Gamma($1/c, 1/\beta^c$)分布的CDF

$$F_X(x) = \begin{cases} 0.5 \cdot F_E(x^c) + 0.5, & if \ x \geq 0 \\ 0.5 \cdot [1 - F_E((-x)^c)], & if \ x < 0 \end{cases}$$

广义高斯分布CDF理论值

Gamma(α,β)的CDF如下，可以用不完备*Gamma*函数(Incomplete Gamma Function)计算, 代码在《Numerical Recipe》中

$$F_{Gamma(\alpha,\beta)}(x)=\frac{1}{\Gamma(\alpha)\beta^{\alpha}}\int_0^x t^{\alpha-1}e^{-t/\beta}dt$$

$$=\frac{1}{\Gamma(\alpha)}\int_0^{x/\beta} t^{\alpha-1}e^{-t}dt$$

广义高斯分布CDF实验值

- 获得CDF实验值方法：
  - 将实验数据拷入Origin
  - Statistics/Descriptive Statistics/Frequency Counts
  - Computation Control: 调整间隔和范围
  - Quantities to compute: bin end + cumulative frequency
  - 绘制实验CDF：bin end为X轴， cumulative frequency为Y轴
  - 理论CDF：F(x), 此处x用bin end代替来计算

关于概率密度问题

- 假设F为累计分布函数，f为概率密度
  - 中值定理：$F(b) - F(a) = F'(t)(b-a) = f(t)(b-a)$，其中$t \in (a, b)$
  - 实验中区间(b-a)足够小，那么$[F(b) - F(a)]/(b-a)$可以近似看成概率密度值