

重要抽样或者权重抽样: 程序, 计算

$f(x) = \exp(-x^2 - 0.25 * x - 0.05 * x^2 * x^2) * \cos(2.1 * x)$

的值, 这个函数可以用 `mathematica` 直接计算, 所以二者可以比较。

本课程要掌握这个方法。这个方法可以被推广到任意维度。

作业: 用这个方法计算

$f(x, y) = \exp(-x^2 - y^2 - 3.5 * \cos(0.25 * x + 2.2 * y) - 0.05 * x^2 * x^2 - 0.05 * y^2 * y^2) * \cos(2.1 * x + 1.35 * y + 0.5 * y * y)$

```
program main
use random

integer, parameter :: Nc = 900000000
double precision :: val, x, f
double precision, parameter :: Pi2 = sqrt(6.283185307179586d0)
integer :: i

val = 0.0

do i=1, Nc
    x = random_normal()
    f = exp(-x*x - 0.25 * x - 0.05 * x*x*x*x) * cos(2.1*x)
    f = f * Pi2 / exp(-0.5*x*x) // 核心, 计算 f/g
    val = val + f
end do

val = val/Nc

write(*, *) "true value = 0.615081736130362, we find val = ", val

end program
```

最终结果:

Nc=900000000

true value = 1.740636625084, we find val = 1.7408722256046805

如果 Nc = 9000000

true value = 0.615081736130362, we find val = 0.61479488769904633

! 3d integration

! 可以推广到任何维度，比如 20 维，都可以。

! 测试结果表明，可以以较快的速度较高的精度计算数值结果。

```
program main
use random

integer, parameter :: Nc = 900000000
double precision :: val, x1, x2, x3, f1, f2, f3, f
double precision, parameter :: Pi2 = sqrt(6.283185307179586d0)
integer :: i

val = 0.0

do i=1, Nc
    x1 = random_normal()
    x2 = random_normal()
    x3 = random_normal()
    f1 = exp(-x1*x1 - 0.25 * x1 - 0.05 *x1*x1*x1*x1) * cos(2.1*x1) * Pi2 / exp(-
0.5*x1*x1)
    f2 = exp(-x2*x2 - 0.25 * x2 - 0.05 *x2*x2*x2*x2) * cos(2.1*x2) * Pi2 / exp(-
0.5*x2*x2)
    f3 = exp(-x3*x3 - 0.25 * x3 - 0.05 *x3*x3*x3*x3) * cos(2.1*x3) * Pi2 / exp(-
0.5*x3*x3)
    f = f1*f2*f3
    val = val + f
end do

val = val/Nc

write(*, *) "1d value = 0.615081736130362, 3d value = 0.615**3 =
0.232701131270331"
write(*, *) "we find val = ", val

end program
```