# Monte Carlo Integration with Variance Reduction

*YIK LUN, KEI*
*allen29@ucla.edu*

## Example 1 $\int_0^1 (cos(20x) + sin(50x))^2 dx$

```
f<-function(x) {(cos(20*x)+sin(50*x))^2}
integrate(f,0,1)
```

```
## 1.045276 with absolute error < 2.1e-10
```

```
n<-1e4;iter<-1e3
```

### Original Method

```
first<-function(n) {
  x<-runif(n)
  return( mean( (cos(20*x)+sin(50*x))^2 ) )
}
original <- replicate(iter,first(n))
```

### Antithetic

```
second<-function(n){
  x<-runif(n);y<-1-x
  return( 0.5 * mean( (cos(20*x)+sin(50*x))^2 + (cos(20*y)+sin(50*y))^2) )
}
antithetic <- replicate(iter,second(n))
```

### Results

```
mean(original)
```

```
## [1] 1.045348
```

```
mean(antithetic)
```

```
## [1] 1.04533
```

```
var(original)
```

```
## [1] 0.0001173453
```

```
var(antithetic)
```

```
## [1] 4.031384e-05
```

# Example 2 $\int_2^5 e^{\frac{-x^2}{2}}/\sqrt{2\pi}dx$

```
g=function(x) { exp(-x^2/2)/sqrt(2*pi) }
integrate(g,2,5)
```

```
## 0.02274985 with absolute error < 2.5e-16
```

## Original Method

```
a=2
b=5
f=function(y) { (g(a+(b-a)*y)-c)/(d-c) }
c=min(g(a:b))
d=max(g(a:b))
first<-function(n){
  x=runif(n)
  return(sum((b-a)*(d-c)*f(x)+c*(b-a))/n)
}
original <- replicate(iter,first(n))
```

## Antithetic

```
second<-function(n){
  x<-runif(n);y<-1-x
  return((b-a)*(d-c)*sum(f(x)+f(y))/(2*n)+c*(b-a))
}
antithetic <- replicate(iter,second(n))
```

# Results

```
mean(original)
```

```
## [1] 0.02276065
```

```r
mean(antithetic)
```

```
## [1] 0.02275149
```

```r
var(original)
```

```
## [1] 1.441215e-07
```

```r
var(antithetic)
```

```
## [1] 4.818039e-08
```

## Example 3

```r
f<-function(x){ (sqrt(abs(x))/sqrt(2*pi))*exp(-x^2/2) }
integrate(f,0,1)
```

```
## 0.2170295 with absolute error < 8.8e-05
```

```r
norm<-rnorm(n,0,1)
unif<-runif(n,0,1)
h<-function(x){ sqrt(abs(x)) }
I<-mean(h(norm))
I
```

```
## [1] 0.8228145
```

```r
I2<-mean(f(unif))
I2
```

```
## [1] 0.2165779
```