積分函數 f(x) = exp(-x)(cos(x^2))^2 ,從 0 到無窮大.
因梯型法和辛普森法對於積分範圍限制 ,取從 0 到 50 ,因 f(50)= 1.113533e-22
結果，梯型法算出積分值和真實值的誤差小於 1e-03，所需要的分割數量為 35440.
另外 ,
辛普森法算出積分值和真實值的誤差小於 1e-03，所需要的分割數量為 208.

```
f<-function(x){exp(-x)*(cos(x^2))^2}          #積分函數 f(x) = exp(-x)(cos(x^2))^2
a<-0;b<-50
取從 0 到  50 ( f(50)= 1.113533e-22
I<-integrate(f,a,Inf)$value                  #用 R 內建算的真實值
I=0.702596463614

#Trapezoidal rule(梯型法)
I_T<-I+1;n<-1
while(abs(I_T-I)>=10^-3){
   n<-n+1
   cat("Now is",n-1,"-th iteration","\n",I_T,"")
   h<-seq(a,b,(b-a)/n)
   for(i in 2:(n+1)){ I_T<-I_T+(f(h[i-1])+f(h[i]))/2 }
   I_T<-I_T*(b-a)/n
}
print(n)
```

```
#Simpsons rule
I_S<-I+1;n_s<-2
while(abs(I_S-I)>=10^-3){
   n_s<-n_s+2
   h<-seq(a,b,(b-a)/n_s)
   z1<-0;z2<-0
   for(i in 2:(n_s/2)){     z1<-z1+2*f(h[2*i-1])    }
   for(i in 1:(n_s/2)){     z2<-z2+4*f(h[2*i])    }
   I_S<-(f(h[1])+f(h[n_s+1])+z1+z2)*(b-a)/n_s/3
   cat("Now is",n_s/2-1,"-th iteration","\n",I_S," ")
}   ;rm(z1,z2)
print(n_s)
```