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) |