**Statistical Computing**

610611104 Pan,Hsing-Cheng

HW8

在區間中找出function 的最小值。利用 The golden-section method。因為端點會一直收縮所以我停止條件設為端點距離的絕對值小於。

|  |  |  |  |
| --- | --- | --- | --- |
| n | Middle value | f(x) | Choosen |
| 32 | 0.7808819 | -24.3696 | 1/2 |
|  |  |  |  |

利用Newton method找出function的一次微分的根，所以在過程中會用到二次微分的函數，所以在二次微分的函數的地方用微分定義在對一次微分函數作用取代。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x0 | n | xn | df(xn) | error |
| 1 | 3 | 0.7808789 | -7.761969e-6 |  |
| 0.5 | 4 | 0.7808791 | 7.105427e-10 |  |
| -1 | 5 | 0.7808791 | -1.715961e-7 |  |
| 2 | 1 | 59815.43 | 8.559012e+14 |  |
| 10 | 7 | 5.95719 | 5.684342e-8 |  |
| 10 | 151 | 5.95719 | -1.705303e-8 |  |

HW8 coding

|  |
| --- |
| f<-function(x){  x^4-14\*x^3+60\*x^2-70\*x  }  plot(c(-1,3),c(-25,5),type = "n")  a<--1  b<-3  c<-1.5  n<-1  conv<-FALSE  while(!conv){  if(abs(b-c)>abs(a-c)){  y<-(b+c)/2  if(f(y)>=f(c)){  b<-y  } else{  a<-c  c<-y  }  } else{  y<-(a+c)/2  if(f(y)>=f(c)){  a<-y  } else{  b<-c  c<-y  }  }  points(a,f(a),col="red",type ="p",pch = 1)  points(b,f(b),col="blue",type ="p",pch=1)  if(abs(a-b)<10^-5 | n>150){conv<-TRUE}  cat("now is",n,"iteration and xa value is:",a,"and xb is:",b,"\n")  cat("middle value is:",(a+b)/2,"middle function value is:",f((a+b)/2),"\n\n")  n<-n+1  }  ####  f<-function(x){  x^4-14\*x^3+60\*x^2-70\*x  }  df<-function(x)  {  h<-10^-5  y<-(f(x+h)-f(x))/h  return(y)  }  conv<-FALSE  xa<- 10  n<-1  h<-10^-5  while(!conv){  xb<-xa  xn<-xb-df(xb)/((df(xb+h)-df(xb))/h)  xa<-xn  if(abs(df(xa))<10^-10 | n>150){conv<-TRUE}  cat("now is",n,"iteration and x value is:",xa,"and df(x) is:",df(xa),"\n")  n<-n+1  } |