Parcial 1

Juan Castañeda

Punto 1b

```
library(pracma)

A=matrix(c(4,-1,-1,-1,
            -1,4,-1,-1,
            -1,-1,4,-1,
            -1,-1,-1,4),nrow=4,byrow=4)

Ap=matrix(c(4,-1,-1,-1,
            -1.15,4,-1,-1,
            -1,-1,4,-1,
            -1,-1,-1,4),nrow=4,byrow=4)

b=c(-exp(1),5,6,0)

punto = function(A,Ap) {
  C = Ap-A  ## Calulamos el error en la matriz original
  cat ("Error en A\n\n")
  print(C)
  cat ("\n\n")
  SA = solve(A,b)    # Resolvemos la matriz para la matriz original
  SAp = solve (Ap,b) # Resolvemos la matriz para la matriz modificada
  C2 = SAp - SA      # Calculamos el error en las soluciones
  cat ("Error en Soluciones\n\n")
  print(C2)
  cat ("\n\n")
  Ea=(norm(C, type = "I"))/(norm(A,type ="I")) #Calculamos el error relativo de la matriz
  Esol = (max(C2)) / (max(SAp))    #Calculamos el error relativo de las soluciones
  cond = norm(A, type = "I")* (norm( inv(A), type = "I")) #Calculamos el numero de
                                                          #condicion para la matriz
  cota = cond *Ea    #Calculamos la cota de error
  cat("La cota de error es: ",cota*100,"% \n\n")
  cat(round(Ea*100,digits = 2),"% de distorcion en la matriz produjo una distorcion de  ",
      round(Esol*100,digits = 2) ,"% en la solución \n")
}
punto(A,Ap)
```

```
## Error en A
##
##       [,1] [,2] [,3] [,4]
## [1,]  0.00    0    0    0
## [2,] -0.15    0    0    0
## [3,]  0.00    0    0    0
## [4,]  0.00    0    0    0
##
##
## Error en Soluciones
##
```

```
## [1] 0.03441301 0.06882602 0.03441301 0.03441301
##
##
## La cota de error es:  15 %
##
## 2.14 % de distorcion en la matriz produjo una distorcion de   2.38 % en la solución
```

Primero se calcula el error relativo de la matriz restando la matriz modificada con la original y a la matriz resultante se le saco la norma y luego se dividio esta entre la norma de la matriz original para obtener el errror relativo de la matriz.

Pra la cota de error se tomo el error relativo de la matriz calculado anteriormente y se multiplico por el numero de condicion de la matriz original.

Para calcular la variacion en las solucion se resolvio la matriz A original y la matriz A con el cambio y luego se restaron ambas para ver en cuento habia cambiado, luego con la ayuda de la norma de ese vector resultante y la norma de la solucion del vector modificado se pudo hallar el error relativo de la solucion.

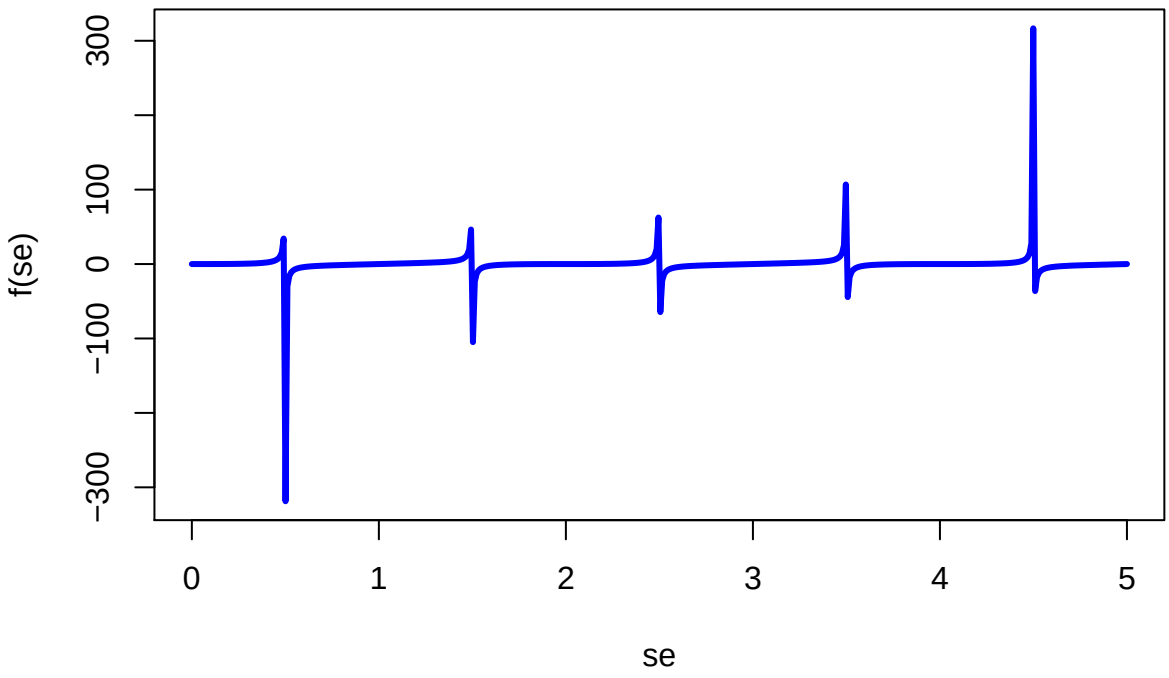
Con la calculado podemos decir que tanto varia la solucion de la matriz con una variacion en la matriz original

Punto 2a

```
f <- function(x){
  tan(pi*x)-sin(pi*x)
}

parteA=function(x0){
  error=1000; #Inicialisamos el error para que pueda entrar al ciclo

  x0=x0-0.1; #-|-->Se varian x1 y x2 en un pequeño valor para poder empezar a hacer el
  x2=x0+1;   #-|   el metodo, se disminuye x0 en 1 para que siempre muestre la raiz a
  x1=x0+0.5; #-|   la derecha del numero dado

  cont=0;
  while( error > 10e-9 && cont <= 100 ){
    x0 = x1 - ( (f(x1) * (x1-x2)) / ((f(x1)-f(x2))) )#Definimos la formula iterativa
                                                      #dada en el parcial

    error = abs(x0-x1) #Calculamos el error para ver si hay mas itercaiones
    x2=x1 #Actualizamos
    x1=x0 #Actualizamos
    cont=cont+1
  }
  if (cont < 100)
    cat( "Raiz ",x0, " Iteraciones = ",cont, "\n")
  else
    cat ("No converge\n")
}
cat ("Metodo Iterativo\n")
```

```
## Metodo Iterativo
```

```
se=seq(0,5,length=500)
plot(se,f(se),type="l",col="blue",lwd=3)
```




```
parteA(0)
```

```
## Raiz  1  Iteraciones =  8
```

```
parteA(2.2)
```

```
## Raiz  3  Iteraciones =  5
```

El metodo funciona, para cualquier numero que se encuentre entre los reales, ya sea decimal o entero. Este entrega la raiz mas proxima hacia la derecha del numero dado