

# Omar Espinel Santamaria

## Parcial 1 Análisis numérico

### Parcial 1 Código en R

```
##Omar Espinel Santamaria
#Documento CC: 1019130596
#Parcial 1 Analisis Numerico
#Punto 2 A
```

```
rm(list=ls())
```

```
funcXn <- function(xa,xb) {

  # Fx <- function(x) tan(pi*x)
  Fx <- function(x) tan(pi*x) - sin(pi*x) #se igualan las dos funciones y se despejan para cero
  x<-seq(xa,xb,0.01)
  plot(Fx(x),type="l")
  abline(h=0)
  c<-1
  i<-0
  # se deciden estos valores iniciales de manera arbitraria pero aun asi cercanos al punto que desea
  el usuario
  x<-xa+.002
  x1<-xa+.001
  x2<-xa
  while ( (c > 1.e-9) || (x<=xb) ) {
    i<-i+1
    x2<-x1
    x1<-x

    if((Fx(x1) - Fx(x2))==0){
      cat ("Error, Indeterminacion por division entre cero\n")
      break
    }

    x<-( x1 - ( ( Fx(x1) * (x1-x2) )/(Fx(x1) -Fx(x2)) ) ) #Se aplica la funcion para encontrar el valor de x

    if( (x-(floor(x))<0.5) || (x-(floor(x))>(3/2)) ){
      #si el valor de x hace que la funcion tan(pi*x) salga de sus limites se entra a este if y se
      recalcula x para evitar que la funcion tienda a inf
      i<-i+1
      x2<-x1
      x1<-x
      x<-( x1 - ( ( Fx(x1) * (x1-x2) )/(Fx(x1) -Fx(x2)) ) )
    }
  }
}
```

```

if((Fx(x)==0) || (abs(Fx(x))<1.e-9) ){
  #Si la raiz da igual a cero o muy cercana (1e-9) a cero
  cat ("Raiz encontrada en",x,"\n")
  break
}
c<-abs(Fx(x)-Fx(x1))/abs(Fx(x))
#se calcula el error
points(x,0, cex = .8, col = "red")
cat("Solucion=",x,"Funcion=",Fx(x),"\\t Error=",c,"\\t Iteracion=",i,"\\n")
}
}
funcXn(0.5,4)

```

# Punto 2 B

#Se usa el metodo de biseccion para calcular la solucion y se mira la cantidad de operaciones, se agrega un if para descartar indeterminaciones en la funcion  $\tan(\pi \cdot x)$

```

funcBiseccion<-function(a,b){
  Fx <- function(x) tan(pi*x) - sin(pi*x)
  x<-seq(a,b,0.01)
  plot(x,Fx(x),type="l")
  abline(h=0)
  i<-0
  x<-b
  c<-abs(a-b)/2
  while (c > 1.e-8)
  {
    if (Fx(x) == 0) break
    if (Fx(x)*Fx(a) < 0) b <- x
    else {a <- x}
    i<-i+1
    x<-(a+b)/2
    if( (x-(floor(x))<0.5) || (x-(floor(x))>(3/2)) ){
      i<-i+1
      x<-(a+b)/2
    }
    c<-abs(a-b)/2
    points(x,0, cex = .8, col = "red")
    cat("Solucion=",x,"\\t Error=",c,"\\t Iteracion=",i,"\\n")
  }
}
funcBiseccion(0.5,4)

```

## Se puede apreciar por la cantidad de iteraciones que el método inicial es mucho mejor al método de bisección, teniendo menos de la mitad de iteraciones para encontrar una raíz

## Resultados del código:

> funcXn(0.5,4)

Solucion= 0.5030063	Funcion= -106.879	Error= 0.4984492	Iteracion= 1
Solucion= 0.5050251	Funcion= -64.33922	Error= 0.6611803	Iteracion= 2
Solucion= 0.5080784	Funcion= -40.394	Error= 0.5927914	Iteracion= 3
Solucion= 0.5132291	Funcion= -25.04664	Error= 0.6127515	Iteracion= 4
Solucion= 0.521635	Funcion= -15.68776	Error= 0.5965721	Iteracion= 5
Solucion= 0.5357253	Funcion= -9.866185	Error= 0.5900533	Iteracion= 6
Solucion= 0.5596051	Funcion= -6.260265	Error= 0.576001	Iteracion= 7
Solucion= 0.601063	Funcion= -3.993087	Error= 0.567776	Iteracion= 8
Solucion= 0.674081	Funcion= -2.496622	Error= 0.5993958	Iteracion= 9
Solucion= 0.7959004	Funcion= -1.344563	Error= 0.8568278	Iteracion= 10
Solucion= 0.9380753	Funcion= -0.3903517	Error= 2.444491	Iteracion= 11
Solucion= 0.9962366	Funcion= -0.02364636	Error= 15.5079	Iteracion= 12
Solucion= 0.999987	Funcion= -8.141885e-05	Error= 289.4286	Iteracion= 13

Raiz encontrada en 1

> funcBiseccion(0.5,4)

Solucion= 2.25	Error= 1.75	Iteracion= 2
Solucion= 3.125	Error= 0.875	Iteracion= 4
Solucion= 3.5625	Error= 0.4375	Iteracion= 5
Solucion= 3.34375	Error= 0.21875	Iteracion= 7
Solucion= 3.453125	Error= 0.109375	Iteracion= 9
Solucion= 3.507812	Error= 0.0546875	Iteracion= 10
Solucion= 3.480469	Error= 0.02734375	Iteracion= 12
Solucion= 3.494141	Error= 0.01367188	Iteracion= 14
Solucion= 3.500977	Error= 0.006835938	Iteracion= 15
Solucion= 3.497559	Error= 0.003417969	Iteracion= 17
Solucion= 3.499268	Error= 0.001708984	Iteracion= 19
Solucion= 3.500122	Error= 0.0008544922	Iteracion= 20
Solucion= 3.499695	Error= 0.0004272461	Iteracion= 22
Solucion= 3.499908	Error= 0.000213623	Iteracion= 24
Solucion= 3.500015	Error= 0.0001068115	Iteracion= 25
Solucion= 3.499962	Error= 5.340576e-05	Iteracion= 27
Solucion= 3.499989	Error= 2.670288e-05	Iteracion= 29
Solucion= 3.500002	Error= 1.335144e-05	Iteracion= 30
Solucion= 3.499995	Error= 6.67572e-06	Iteracion= 32
Solucion= 3.499999	Error= 3.33786e-06	Iteracion= 34
Solucion= 3.5	Error= 1.66893e-06	Iteracion= 35
Solucion= 3.499999	Error= 8.34465e-07	Iteracion= 37
Solucion= 3.5	Error= 4.172325e-07	Iteracion= 39
Solucion= 3.5	Error= 2.086163e-07	Iteracion= 40
Solucion= 3.5	Error= 1.043081e-07	Iteracion= 42
Solucion= 3.5	Error= 5.215406e-08	Iteracion= 44
Solucion= 3.5	Error= 2.607703e-08	Iteracion= 45
Solucion= 3.5	Error= 1.303852e-08	Iteracion= 47
Solucion= 3.5	Error= 6.519258e-09	Iteracion= 49