

RAICES POLINOMIO DE SEGUNDO GRADO

Santiago Orjuela Convers

Abril 2020

Las raíces de un polinomio de segundo grado pueden ser calculadas de varias formas, pero en especial, se utiliza la fórmula cuadrática que para un polinomio $ax^2 + bx + c$ está dada por la ecuación:

$$x_1 = \frac{-b^2 + \sqrt{b^2 - 4ac}}{2a}$$
$$x_2 = \frac{-b^2 - \sqrt{b^2 - 4ac}}{2a}$$

Los dos números que son el resultado de la ecuación cuadrática son las raíces del polinomio $ax^2 + bx + c$. Es útil poseer herramientas en computador que permitan calcular las raíces de polinomios de grados cuyos coeficientes no permiten un cálculo sencillo. Se implementó un código en R que calcula las raíces de un polinomio de grado 2, donde sus parámetros son los coeficientes del polinomio. El código es:

```
raices <- function(a,b,c){  
  
  d <- b^2-4*a*c  
  
  if(a==0) {stop("No es cuadrática")}  
  
  if(d==0){  
    x1 <- -b/(2*a)  
    cat("Una raíz real x1 = ", x1)  
  }  
  if(d>0){  
    x1 <- (-b+sqrt(d))/(2*a)  
    x2 <- (-b-sqrt(d))/(2*a)  
    cat("Dos raíces reales \n", "x1 = ", x1, "\n", "x2 = ", x2)  
  }  
  if(d<0){ cat("Las raíces son complejas")  
  }  
}
```

Algunos ejemplos usando el código son los siguientes:


```

raices <- function(a,b,c){
d <- mpfr(b^2-4*a*c,64)

if(a==0) {stop("No es cuadrática")}

if(d==0){
  x1 <- -b/(2*a)
  cat("Una raíz real x1 = ", x1)
}
if(d>0){
  x1 <- (-b+sqrt(d))/(2*a)
  x2 <- (-b-sqrt(d))/(2*a)
  print(x1)
  print(x2)
}
if(d<0){ cat("Las raíces son complejas")
}
}

```

La función que se invoca, llamada mpfr tiene dos parámetros que son: el número y la precisión en bits, en este caso se utiliza una precisión de 64 bits. Esta herramienta arregla el problema de tener una diferencia de números muy cercana al cero y que el computador la aproxime a cero, lo cual daría una respuesta incorrecta.

Para observar el arreglo que otorga el paquete mpfr, considere el siguiente polinomio: $\frac{7}{4}x^2 + 2x + 0,571428571415$. El valor del discriminante en este polinomio es de 0.000000000094. Luego, colocando este polinomio en la función raíces con el paquete Rmpfr se obtiene:

```

> raices()
a = 1.75
b = 2
c = 0.571428571415
1 'mpfr' number of precision 64 bits
[1] -0.571425786636580938793
1 'mpfr' number of precision 64 bits
[1] -0.571431356220561918334

```