Punto fijo

Uriel Paluch

4/9/2021

Método de punto fijo

Un punto fijo para una función es un número en el que el valor de la función no cambia cuando se apica la función. Decimos quqe el número p es un punto fijo para una función dada g si $g(p) = p$

Los problemas para enontrar la raíz y los de punto fijo son equivalentes en el siguiente sentido:

Dado un problema para encontrar la raíz $f(p) = 0$, podemos definir las funciones g con un punto fijo en p en diferentes formas, por ejemplo:

$$g(x) = x - f(x)$$

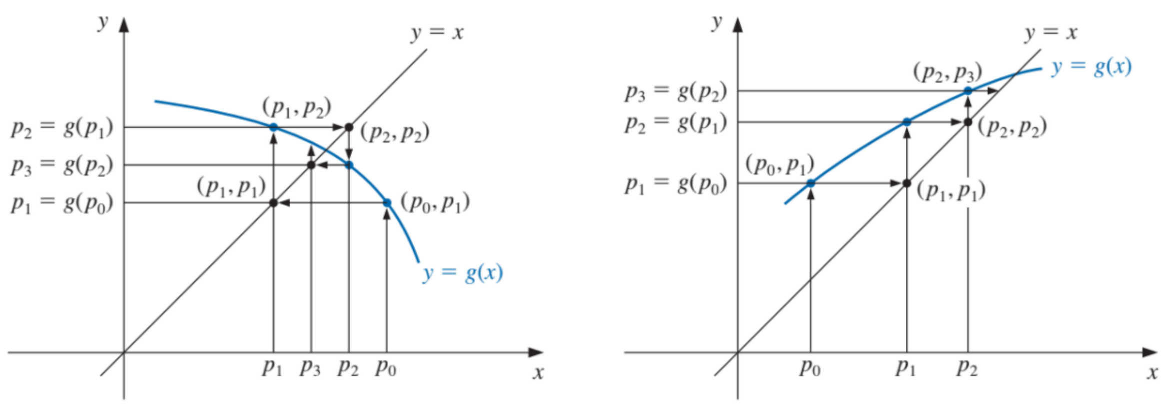
Si la función g tiene un punto fijo en p, entonces la función tiene un cero en p.

Teorema:

1. Si la función es continua en el intervalo $[a; b]$
2. Si, ademas, $g'(x)$ existe en todo $(a; b)$ y hay una constante $0 < k < 1$ con:

$$|g'(x)| \leq k$$

Método de iteración de punto fijo:



a) b)

Figure 1: Método de iteración de punto fijo

```r
#Método de punto fijo
PuntoFijo <- function(p0, n = 100, tol){
```

```r
  #Donde p0 es la aproximación inicial
  #El número máximo de iteraciones n viene por default en 100
  #Y tol es la toleranacia al error

  #Instancio las listas vacias
  lista_p <- c(NULL)
  lista_gp <- c(NULL)

  for (i in 1:n) {
    #Calculo p
    p <- f(p0)

    lista_p[i] <- p0
    lista_gp[i] <- p

    if(abs(p-p0) <= tol){
      #Creo un data frame con las listas
      datos <- data.frame(lista_p, lista_gp)
      colnames(datos) <- c("P", "G(P)")
      print(datos)
      return(p)
    }

    p0 <- p
  }

  #En el caso de que falle el método
  return(paste('El método falla luego de: ', n, ' iteraciones'))
}
```

Es muy importante dejar expresadas las ecuaciones con la forma de punto fijo:

$$x = g(x)$$

No todas las formas de la expresión pueden ser correctas, la ecuación puede resultar en un punto fijo, diverger o volverse indefinida dependiendo de el cumplimiento del teorema.

Ejercicios:

- Aplicar directamente el algoritmo:
  1. $x = g_1(x) = x - x^3 - 4x^2 + 10$
  2. $x = g_2(x) = (\frac{10}{x} - 4x)^{\frac{1}{2}}$
  3. $x = g_3(x) = \frac{1}{2}(10 - x^3)^{\frac{1}{2}}$
  4. $x = g_4(x) = (\frac{10}{4+x})^{\frac{1}{2}}$
  5. $x = g_5(x) = x - \frac{x^3+4x^2-10}{3x^2+8x}$
- Reescribir como problemas de punto fijo y hallar la solución de:
  1. $cos(x) = \sqrt{x}$
  2. $2 + cos(e^x - 2) = e^x$
  3. $x^3 - 7x^2 + 14x - 6 = 0$

Solución:

Aplicando directamente el algoritmo:

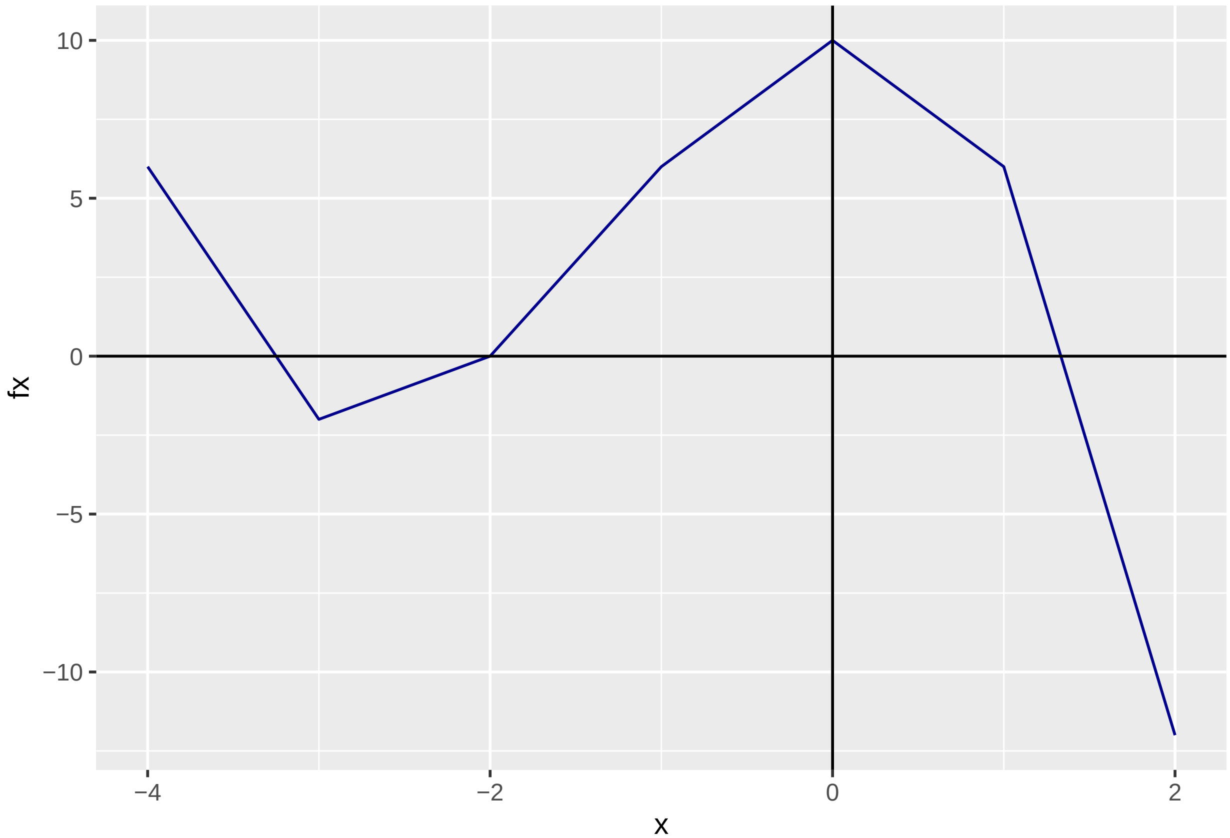
Ejercicio 1:

Grafico la función:

$x = g_1(x) = x - x^3 - 4x^2 + 10$

Al ser un polinomio completo de grado 3 debe tener 3 raices

```
f <- function(x){
  return(x - x^3 - 4*x^2 + 10)
}

#Instancio un vector que me va a indicar los puntos en la función
x <- seq(-4, 2, by = 1)

#Genero los puntos
fx <- f(x)

#Creo un data frame con los x e y
df <- data.frame(x, fx)

#Instancio los datos
gg_fx <- ggplot(data = df)

#Agrego la capa con los datos
gg_fx <- gg_fx + aes(x = x, y = fx)

#Est grafica una linea
gg_fx <- gg_fx + geom_line(linetype = 1, colour = "darkblue")

#Agrego el eje X
gg_fx <- gg_fx + geom_vline(xintercept = 0, linetype = 1)

#Agrego el eje Y
gg_fx <- gg_fx + geom_hline(yintercept = 0, linetype = 1)

#Grafico
gg_fx
```



Se observa graficamente que hay raices en los siguientes intervalos $[-4;-3]$, $[-2.5;-1.5]$ y $[1;2]$.

Evaluo el método con $p_0 = 1$

```
this_could_go_wrong <- tryCatch(
  print(PuntoFijo(p0 = 1, tol = 0.0001)),
  error = function(e){print("Error")})
```

```
## [1] "Error"
```

Tira error porque probablemente no cumpla con las condiciones del teorema. Vamos a comprobarlo:

1. Se observa graficamente que la función es continua en todo el intervalo.
2. Evaluo la derivada en diferentes puntos del intervalo $[1;2]$

```
fprima <- D(expression(x - x^3 - 4*x^2 + 10), "x")

fprima
```

```
## 1 - 3 * x^2 - 4 * (2 * x)
```

```
fprima <- function(x){
  return(1-3*x^2-4*(2*x))
}

x <- seq(1, 2, by =0.1)

#Genero los puntos
fprimax <- fprima(x)
```

```r
#Creo un data frame con los x e y
df <- data.frame(x, fprimax)

#Lo imprimo
df
```

```
##      x fprimax
## 1  1.0  -10.00
## 2  1.1  -11.43
## 3  1.2  -12.92
## 4  1.3  -14.47
## 5  1.4  -16.08
## 6  1.5  -17.75
## 7  1.6  -19.48
## 8  1.7  -21.27
## 9  1.8  -23.12
## 10 1.9  -25.03
## 11 2.0  -27.00
```

Se observa que no cumple con:

$$|g'(x)| \leq 1$$

Evaluo el método con $p_0 = -2$

```r
this_could_go_wrong <- tryCatch(
  print(PuntoFijo(p0 = -2, tol = 0.0001)),
  error = function(e){print("Error")})
```

```
## [1] "Error"
```

Tira error porque probablemente no cumpla con las condiciones del teorema. Vamos a comprobarlo:

1. Se observa graficamente que la función es continua en todo el intervalo.
2. Evaluo la derivada en diferentes puntos del intervalo $[-2.5, -1.5]$

```r
x <- seq(-2.5, -1.5, by =0.1)

#Genero los puntos
fprimax <- fprima(x)

#Creo un data frame con los x e y
df <- data.frame(x, fprimax)

#Lo imprimo
df
```

```
##       x fprimax
## 1  -2.5    2.25
## 2  -2.4    2.92
## 3  -2.3    3.53
## 4  -2.2    4.08
## 5  -2.1    4.57
## 6  -2.0    5.00
## 7  -1.9    5.37
## 8  -1.8    5.68
## 9  -1.7    5.93
```

```
## 10 -1.6    6.12
## 11 -1.5    6.25
```

Se observa que no cumple con:

$$|g'(x)| \leq 1$$

Evaluo el método con $p_0 = -3$

```r
this_could_go_wrong <- tryCatch(
  print(PuntoFijo(p0 = -3, tol = 0.0001)),
  error = function(e){print("Error")})
```

```
## [1] "Error"
```

Tira error porque probablemente no cumpla con las condiciones del teorema. Vamos a comprobarlo:

1. Se observa graficamente que la función es continua en todo el intervalo.
2. Evaluo la derivada en diferentes puntos del intervalo $[-4, -3]$

```r
x <- seq(-4, -3, by =0.1)

#Genero los puntos
fprimax <- fprima(x)

#Creo un data frame con los x e y
df <- data.frame(x, fprimax)

#Lo imprimo
df
```

```
##       x fprimax
## 1  -4.0  -15.00
## 2  -3.9  -13.43
## 3  -3.8  -11.92
## 4  -3.7  -10.47
## 5  -3.6   -9.08
## 6  -3.5   -7.75
## 7  -3.4   -6.48
## 8  -3.3   -5.27
## 9  -3.2   -4.12
## 10 -3.1   -3.03
## 11 -3.0   -2.00
```

Se observa que no cumple con:

$$|g'(x)| \leq 1$$

Ejercicio 2:

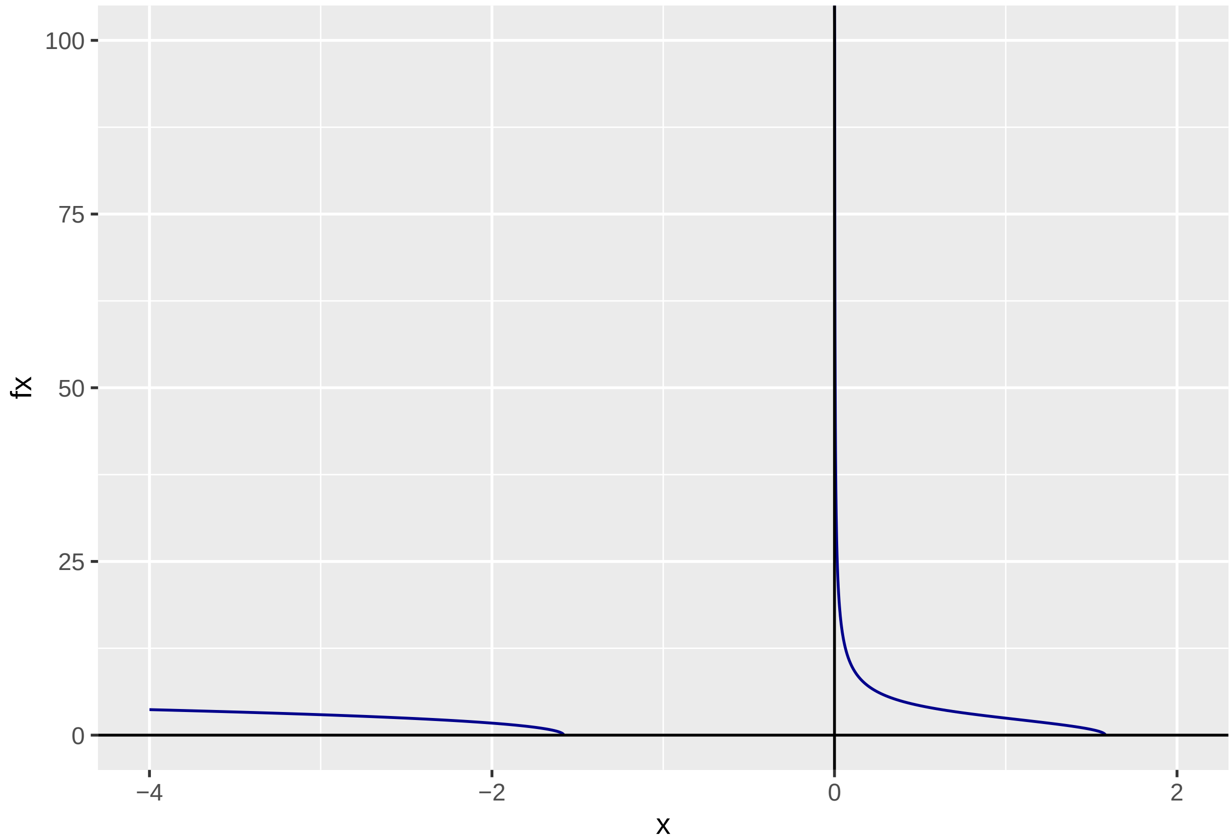
Grafico la función:

$x = g_2(x) = (\frac{10}{x} - 4x)^{\frac{1}{2}}$

```r
f <- function(x){
  return((10/x-4*x)^(1/2))
}
```

```r
#Instancio un vector que me va a indicar los puntos en la función
x <- seq(-4, 2, by = 0.001)

#Genero los puntos
fx <- f(x)

#Creo un data frame con los x e y
df <- data.frame(x, fx)

#Instancio los datos
gg_fx <- ggplot(data = df)

#Agrego la capa con los datos
gg_fx <- gg_fx + aes(x = x, y = fx)

#Est grafica una linea
gg_fx <- gg_fx + geom_line(linetype = 1, colour = "darkblue")

#Agrego el eje X
gg_fx <- gg_fx + geom_vline(xintercept = 0, linetype = 1)

#Agrego el eje Y
gg_fx <- gg_fx + geom_hline(yintercept = 0, linetype = 1)

#Grafico
gg_fx
```

```
## Warning: Removed 419 row(s) containing missing values (geom_path).
```



Se observa graficamente que la función no cumple con la primera condición del teorema:
1. Ser continua.

Por lo tanto, no es aplicable el método de punto fijo para encontrar sus raices

---

Ejercicio 3:

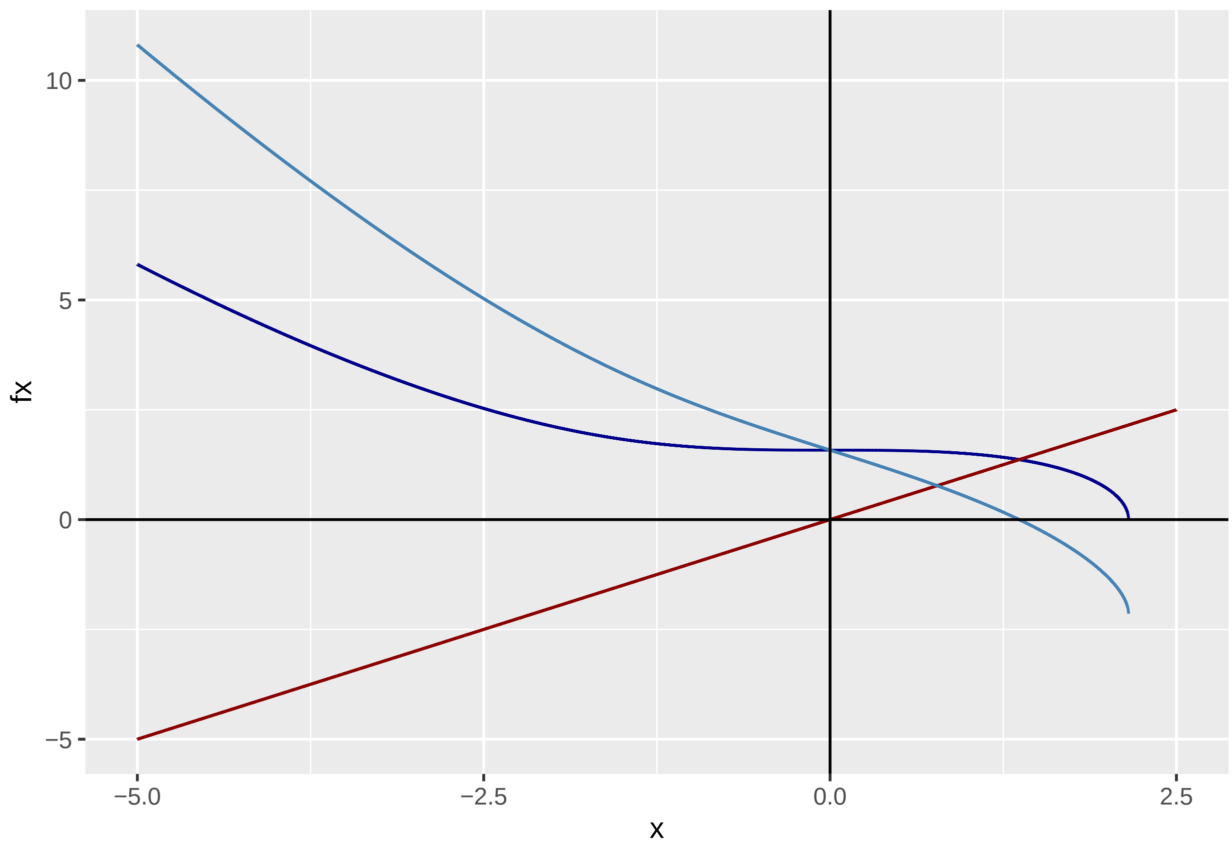
Grafico la función:

$x = g_3(x) = \frac{1}{2}(10 - x^3)^{\frac{1}{2}}$

```
#La función del ejercicio
f <- function(x){
  return((1/2)*(10-x^3)^(1/2))
}
#La función para graficar la raiz
g <- function(x){
  return(-x+(1/2)*(10-x^3)^(1/2))
}

#Instancio un vector que me va a indicar los puntos en la función
x <- seq(-5, 2.5, by = 0.0001)

#Genero los puntos
fx <- f(x)
```

```r
#Creo un data frame con los x e y
df <- data.frame(x, fx)

#Instancio los datos
gg_fx <- ggplot(data = df)

#Agrego la capa con los datos
gg_fx <- gg_fx + aes(x = x, y = fx)

#Grafica la función del ejercicio
gg_fx <- gg_fx + geom_line(linetype = 1, colour = "darkblue")

#Gráfico x = y
gg_fx <- gg_fx + geom_line(aes(y = x), colour = "darkred")

#Gráfico la función del ejercicio donde esta la raiz
gg_fx <- gg_fx + geom_line(aes(x = x, y = g(x)), colour = "steelblue")

#Agrego el eje X
gg_fx <- gg_fx + geom_vline(xintercept = 0, linetype = 1)

#Agrego el eje Y
gg_fx <- gg_fx + geom_hline(yintercept = 0, linetype = 1)

#Grafico
gg_fx
```

```
## Warning: Removed 3456 row(s) containing missing values (geom_path).

## Warning: Removed 3456 row(s) containing missing values (geom_path).
```




```
this_could_go_wrong <- tryCatch(
  PuntoFijo(p0 = 2, tol = 0.001),
  error = function(e){print("Error")})
```

```
##             P      G(P)
## 1  2.0000000 0.7071068
## 2  0.7071068 1.5529365
## 3  1.5529365 1.2504919
## 4  1.2504919 1.4181474
## 5  1.4181474 1.3367782
## 6  1.3367782 1.3794210
## 7  1.3794210 1.3578691
## 8  1.3578691 1.3689731
## 9  1.3689731 1.3633071
## 10 1.3633071 1.3662127
## 11 1.3662127 1.3647264
## 12 1.3647264 1.3654877
```

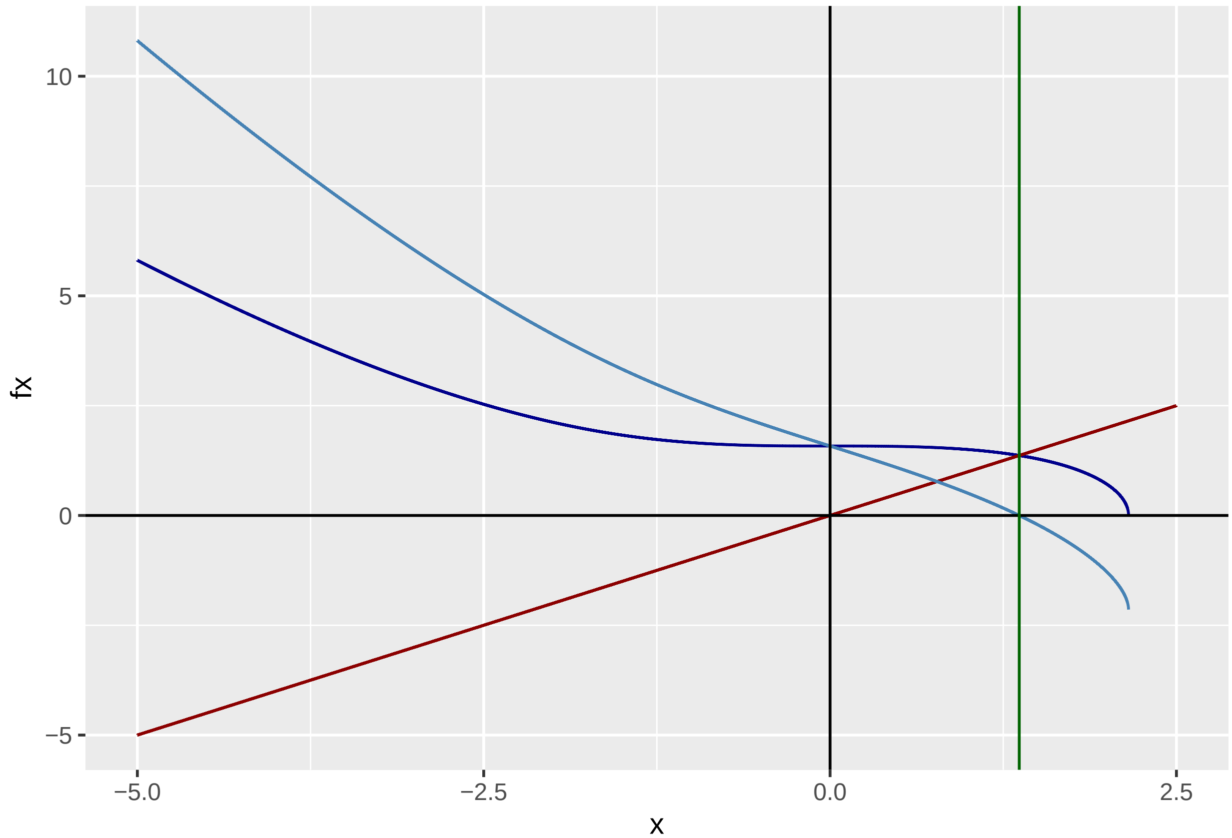
```
if (this_could_go_wrong != "Error"){
  raiz <- this_could_go_wrong
  print(paste("La raiz es: ", raiz))
}
```

```
## [1] "La raiz es:  1.36548770206867"
```

```
#Agrego la linea recta que une los puntos entre la función del ejercicio, el punto fijo y la raiz
gg_fx <- gg_fx + geom_vline(xintercept = raiz, linetype = 1, colour="darkgreen")
```

```
#Grafico
gg_fx
```

```
## Warning: Removed 3456 row(s) containing missing values (geom_path).

## Warning: Removed 3456 row(s) containing missing values (geom_path).
```



Verifico que se cumplan las condiciones del teorema

1. Se observa graficamente que la función es continua en todo el intervalo.
2. Evaluo la derivada en diferentes puntos del intervalo $[1; 1.5]$

```
fprima <- D(expression((1/2)*(10-x^3)^(1/2)), "x")

fprima
```

```
## -((1/2) * ((10 - x^3)^((1/2) - 1) * ((1/2) * (3 * x^2))))
```

```
fprima <- function(x){
  return((1/2) * ((10 - x^3)^((1/2) - 1) * ((1/2) * (3 * x^2))))
}

x <- seq(1, 1.5, by =0.1)

#Genero los puntos
fprimax <- fprima(x)

#Creo un data frame con los x e y
```

```
df <- data.frame(x, fprimax)

#Lo imprimo
df
```

```
##     x   fprimax
## 1 1.0 0.2500000
## 2 1.1 0.3082209
## 3 1.2 0.3755074
## 4 1.3 0.4537506
## 5 1.4 0.5457185
## 6 1.5 0.6556180
```

Ejercicio 4:

Grafico la función:

$x = g_4(x) = (\frac{10}{4+x})^{\frac{1}{2}}$

```
#La función del ejercicio
f <- function(x){
  return((10/(4+x))^(1/2))
}
#La función para graficar la raiz
g <- function(x){
  return(-x+(10/(4+x))^(1/2))
}

#Instancio un vector que me va a indicar los puntos en la función
x <- seq(-4, 2.5, by = 0.0001)

#Genero los puntos
fx <- f(x)

#Creo un data frame con los x e y
df <- data.frame(x, fx)

#Instancio los datos
gg_fx <- ggplot(data = df)

#Agrego la capa con los datos
gg_fx <- gg_fx + aes(x = x, y = fx)

#Grafica la función del ejercicio
gg_fx <- gg_fx + geom_line(linetype = 1, colour = "darkblue")

#Gráfico x = y
gg_fx <- gg_fx + geom_line(aes(y = x), colour = "darkred")

#Gráfico la función del ejercicio donde esta la raiz
gg_fx <- gg_fx + geom_line(aes(x = x, y = g(x)), colour = "steelblue")

#Agrego el eje X
```

```r
gg_fx <- gg_fx + geom_vline(xintercept = 0, linetype = 1)

#Agrego el eje Y
gg_fx <- gg_fx + geom_hline(yintercept = 0, linetype = 1)

#Grafico
gg_fx
```

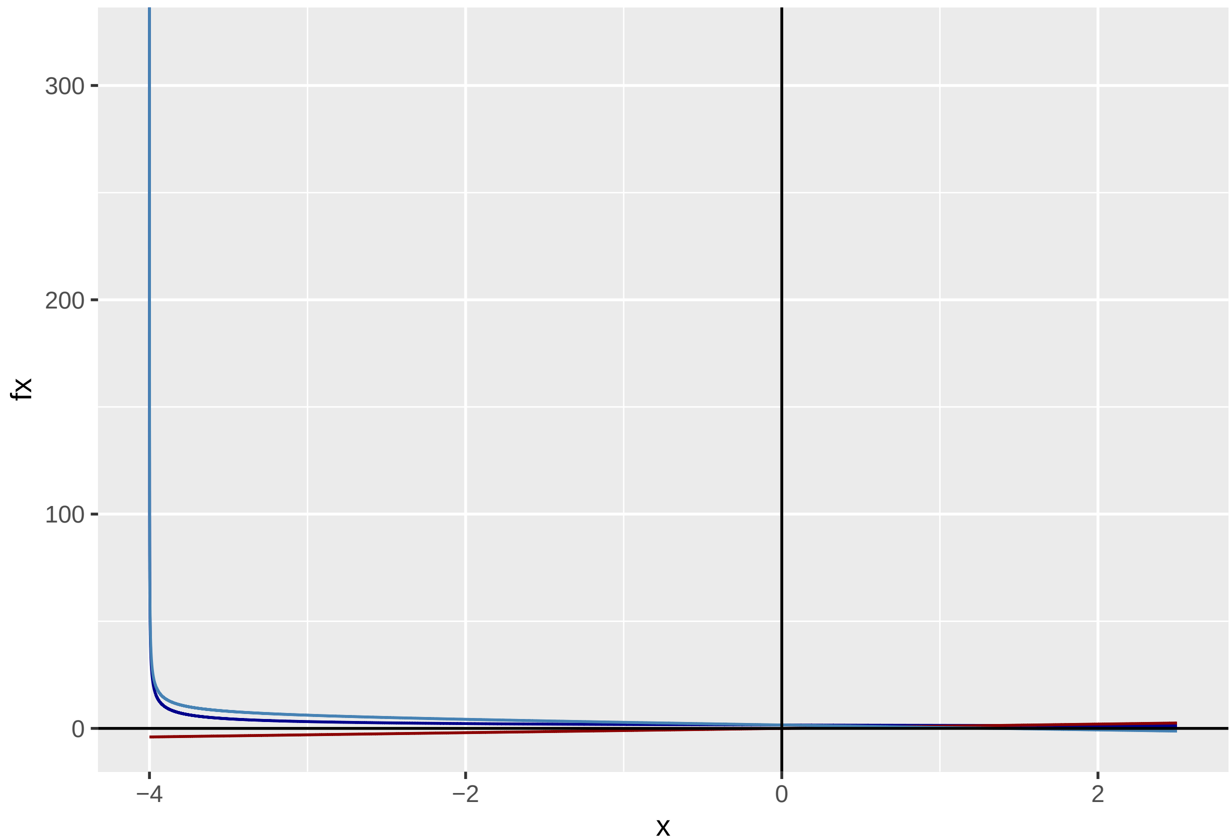
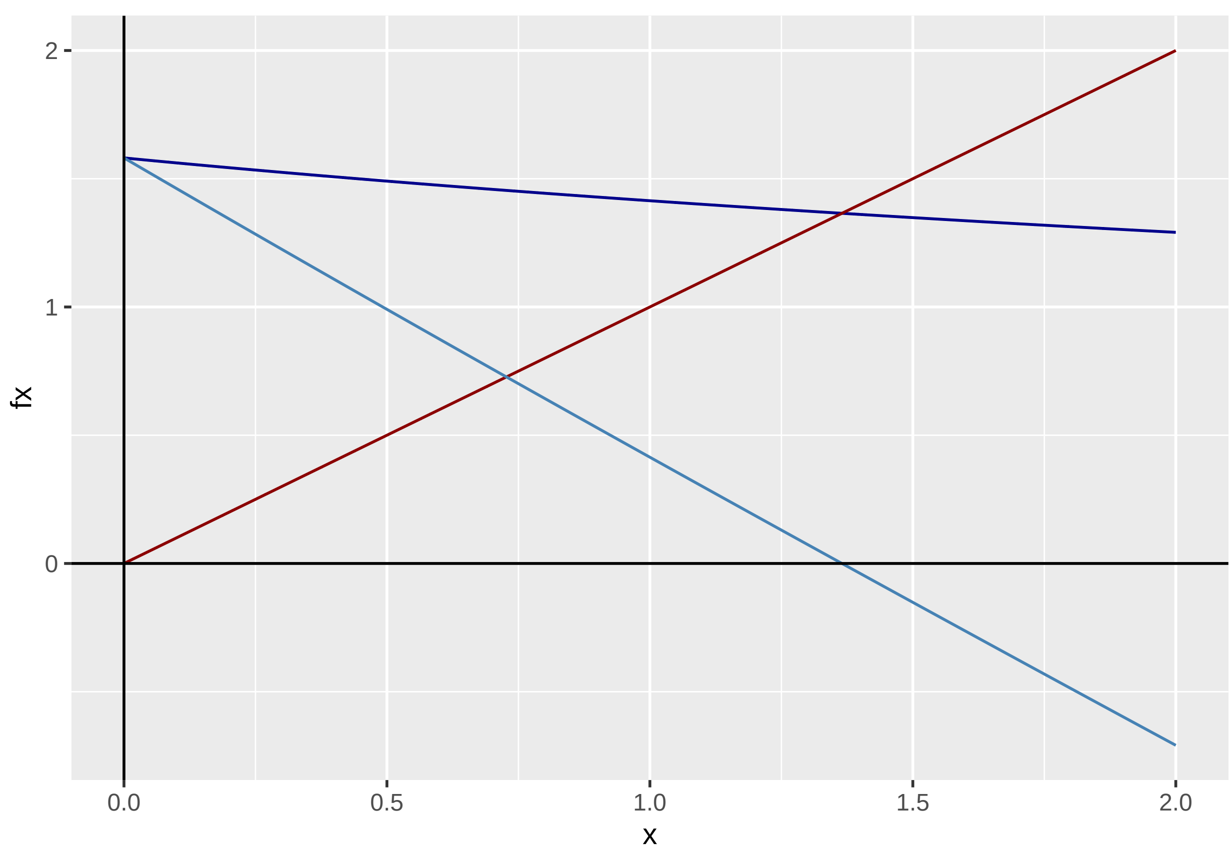


Grafico nuevamente la función en el intervalo $[0; 2]$ para tener una visión mas clara.

```r
#La función del ejercicio
f <- function(x){
  return((10/(4+x))^(1/2))
}
#La función para graficar la raiz
g <- function(x){
  return(-x+(10/(4+x))^(1/2))
}

#Instancio un vector que me va a indicar los puntos en la función
x <- seq(0, 2, by = 0.0001)

#Genero los puntos
fx <- f(x)

#Creo un data frame con los x e y
```

```r
df <- data.frame(x, fx)

#Instancio los datos
gg_fx <- ggplot(data = df)

#Agrego la capa con los datos
gg_fx <- gg_fx + aes(x = x, y = fx)

#Grafica la función del ejercicio
gg_fx <- gg_fx + geom_line(linetype = 1, colour = "darkblue")

#Gráfico x = y
gg_fx <- gg_fx + geom_line(aes(y = x), colour = "darkred")

#Gráfico la función del ejercicio donde esta la raiz
gg_fx <- gg_fx + geom_line(aes(x = x, y = g(x)), colour = "steelblue")

#Agrego el eje X
gg_fx <- gg_fx + geom_vline(xintercept = 0, linetype = 1)

#Agrego el eje Y
gg_fx <- gg_fx + geom_hline(yintercept = 0, linetype = 1)

#Grafico
gg_fx
```



Se observa una raíz cerca de $p_0 = 1.25$

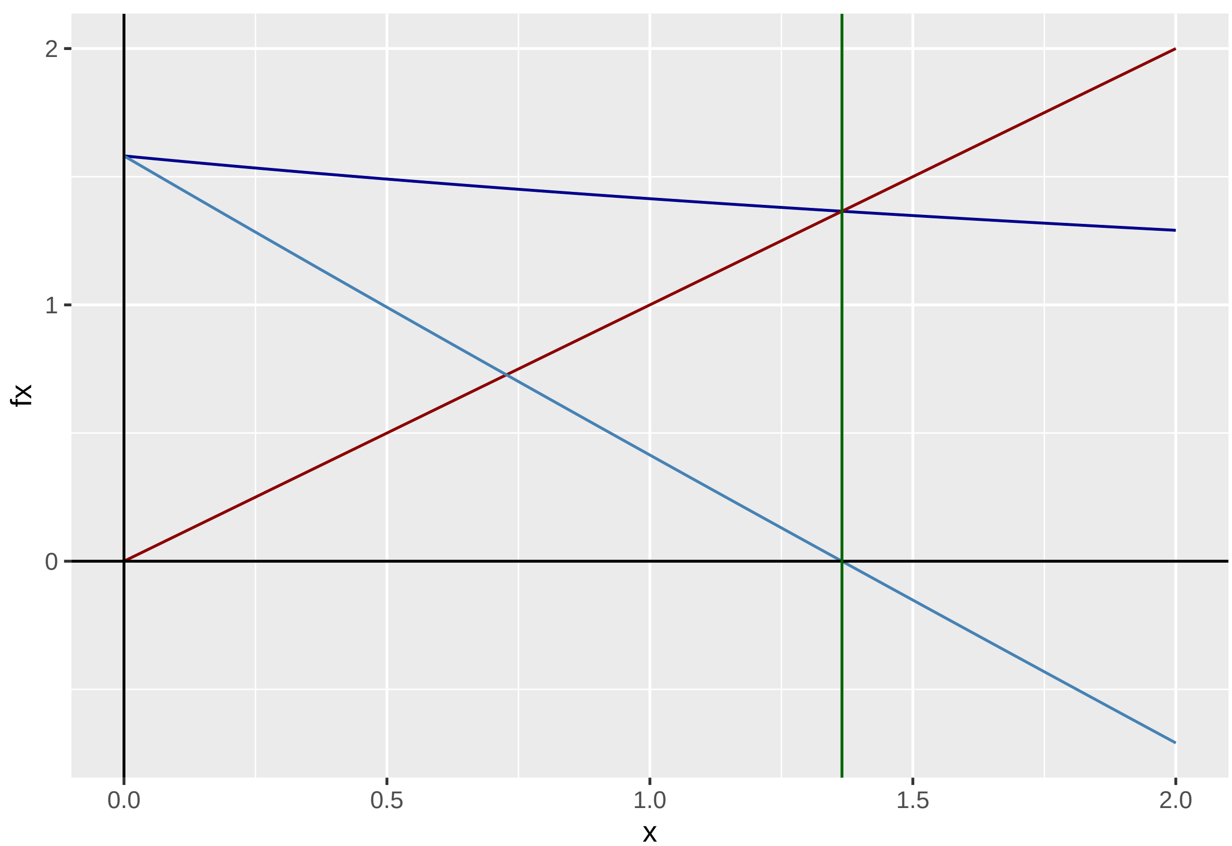
```
this_could_go_wrong <- tryCatch(
  PuntoFijo(p0 = 1.25, tol = 0.001),
  error = function(e){print("Error")})
```

```
##          P     G(P)
## 1 1.250000 1.380131
## 2 1.380131 1.363338
## 3 1.363338 1.365471
## 4 1.365471 1.365199
```

```
if (this_could_go_wrong != "Error"){
  raiz <- this_could_go_wrong
  print(paste("La raiz es: ", raiz))
}
```

```
## [1] "La raiz es:  1.36519938124783"
```

```
#Agrego la linea recta que une los puntos entre la función del ejercicio, el punto fijo y la raiz
gg_fx <- gg_fx + geom_vline(xintercept = raiz, linetype = 1, colour="darkgreen")

#Grafico
gg_fx
```



Verifico que se cumplan las condiciones del teorema

1. Se observa graficamente que la función es continua en todo el intervalo.
2. Evaluo la derivada en diferentes puntos del intervalo $[0; 2]$

```r
fprima <- D(expression((10/(4+x))^(1/2)), "x")

fprima
```

```
## -((10/(4 + x))^((1/2) - 1) * ((1/2) * (10/(4 + x)^2)))
```

```r
fprima <- function(x){
  return((10/(4 + x))^((1/2) - 1) * ((1/2) * (10/(4 + x)^2)))
}

x <- seq(1, 1.5, by =0.1)

#Genero los puntos
fprimax <- fprima(x)

#Creo un data frame con los x e y
df <- data.frame(x, fprimax)

#Lo imprimo
df
```

```
##     x   fprimax
## 1 1.0 0.1414214
## 2 1.1 0.1372824
## 3 1.2 0.1333414
## 4 1.3 0.1295854
## 5 1.4 0.1260026
## 6 1.5 0.1225818
```

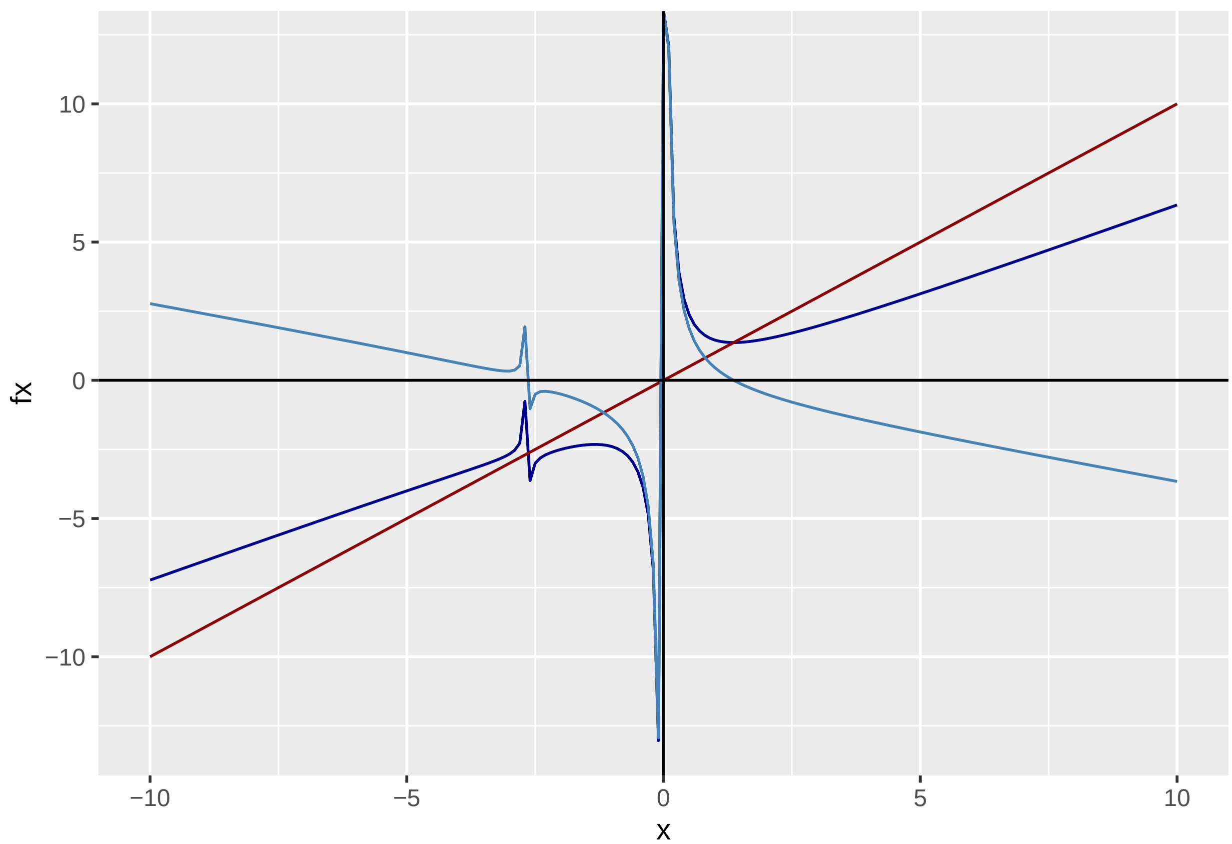
Ejercicio 5:

Grafico la función:

$x = g_5(x) = x - \frac{x^3+4x^2-10}{3x^2+8x}$

```r
#La función del ejercicio
f <- function(x){
  return(x-(x^3+4*x^2-10)/(3*x^2+8*x))
}
#La función para graficar la raiz
g <- function(x){
  return(-x+x-(x^3+4*x^2-10)/(3*x^2+8*x))
}

#Instancio un vector que me va a indicar los puntos en la función
x <- seq(-10, 10, by = 0.1)

#Genero los puntos
fx <- f(x)

#Creo un data frame con los x e y
df <- data.frame(x, fx)

#Instancio los datos
```

```r
gg_fx <- ggplot(data = df)

#Agrego la capa con los datos
gg_fx <- gg_fx + aes(x = x, y = fx)

#Grafica la función del ejercicio
gg_fx <- gg_fx + geom_line(linetype = 1, colour = "darkblue")

#Gráfico x = y
gg_fx <- gg_fx + geom_line(aes(y = x), colour = "darkred")

#Gráfico la función del ejercicio donde esta la raiz
gg_fx <- gg_fx + geom_line(aes(x = x, y = g(x)), colour = "steelblue")

#Agrego el eje X
gg_fx <- gg_fx + geom_vline(xintercept = 0, linetype = 1)

#Agrego el eje Y
gg_fx <- gg_fx + geom_hline(yintercept = 0, linetype = 1)

#Grafico
gg_fx
```



Se observa una raiz en el intervalo $[1; 2.5]$

Tomo $p_0 = 1$

```
this_could_go_wrong <- tryCatch(
  PuntoFijo(p0 = 1, tol = 0.001),
  error = function(e){print("Error")})
```

```
##          P     G(P)
## 1 1.000000 1.454545
## 2 1.454545 1.368900
## 3 1.368900 1.365237
## 4 1.365237 1.365230
```

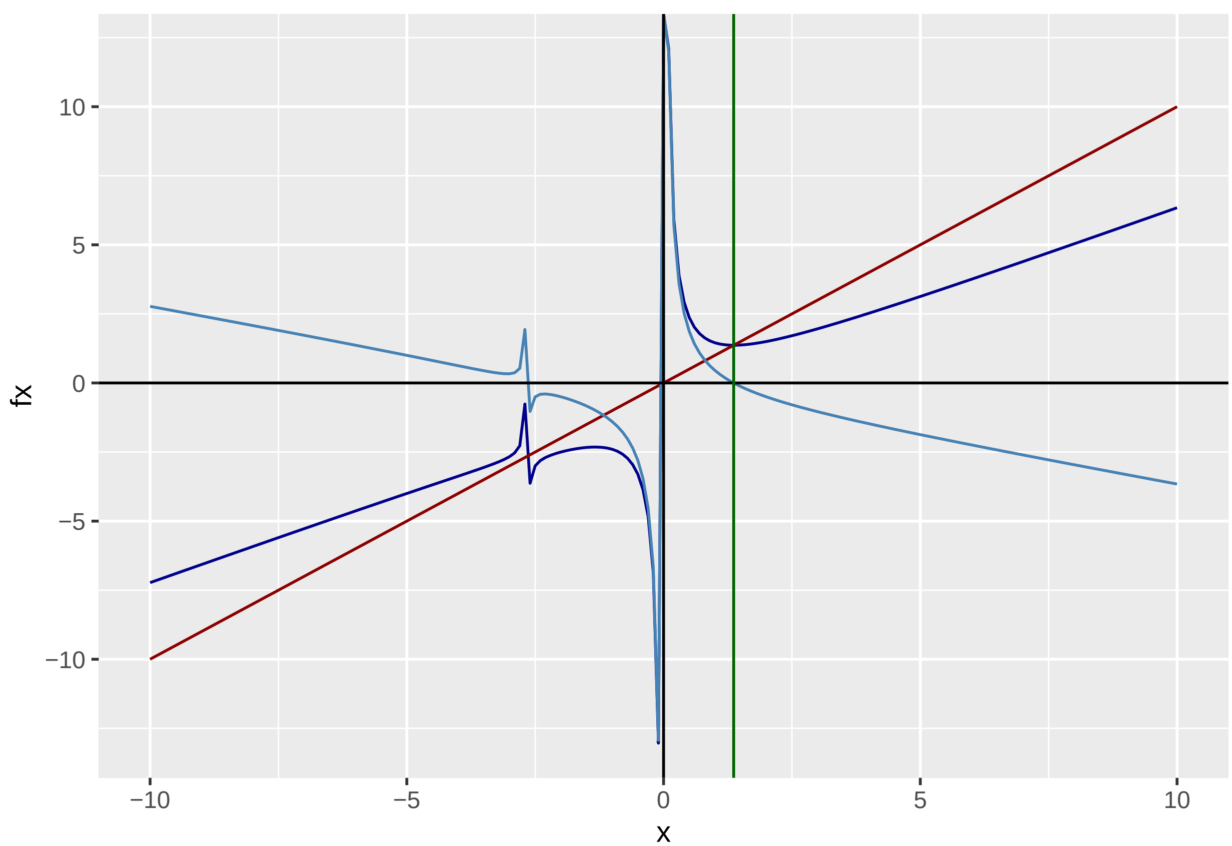
```
print(this_could_go_wrong)
```

```
## [1] 1.36523
```

```
if (this_could_go_wrong != "Error"){
  raiz <- this_could_go_wrong
  print(paste("La raiz es: ", raiz))
}
```

```
## [1] "La raiz es:  1.36523001343537"
```

```
#Agrego la linea recta que une los puntos entre la función del ejercicio, el punto fijo y la raiz
gg_fx <- gg_fx + geom_vline(xintercept = raiz, linetype = 1, colour="darkgreen")

gg_fx
```



Verifico que se cumplan las condiciones del teorema

1. Se observa graficamente que la función es continua en todo el intervalo.
2. Evaluo la derivada en diferentes puntos del intervalo $[1; 1.5]$

```
fprima <- D(expression(x-(x^3+4*x^2-10)/(3*x^2+8*x)), "x")

fprima
```

```
## 1 - ((3 * x^2 + 4 * (2 * x))/(3 * x^2 + 8 * x) - (x^3 + 4 * x^2 -
##     10) * (3 * (2 * x) + 8)/(3 * x^2 + 8 * x)^2)
```

```
fprima <- function(x){
  return(1 - ((3 * x^2 + 4 * (2 * x))/(3 * x^2 + 8 * x) - (x^3 + 4 * x^2 -
    10) * (3 * (2 * x) + 8)/(3 * x^2 + 8 * x)^2))
}

x <- seq(1, 2.5, by =0.1)

#Genero los puntos
fprimax <- fprima(x)

#Creo un data frame con los x e y
df <- data.frame(x, fprimax)

#Lo imprimo
df
```

```
##      x     fprimax
## 1  1.0 -0.57851240
## 2  1.1 -0.36182283
## 3  1.2 -0.19705377
## 4  1.3 -0.06885901
## 5  1.4  0.03283076
## 6  1.5  0.11484444
## 7  1.6  0.18194580
## 8  1.7  0.23753990
## 9  1.8  0.28411288
## 10 1.9  0.32351424
## 11 2.0  0.35714286
## 12 2.1  0.38607258
## 13 2.2  0.41113880
## 14 2.3  0.43299928
## 15 2.4  0.45217759
## 16 2.5  0.46909469
```

Solución:

Reescribir como problemas de punto fijo y hallar la solución de:

Ejercicio 1:

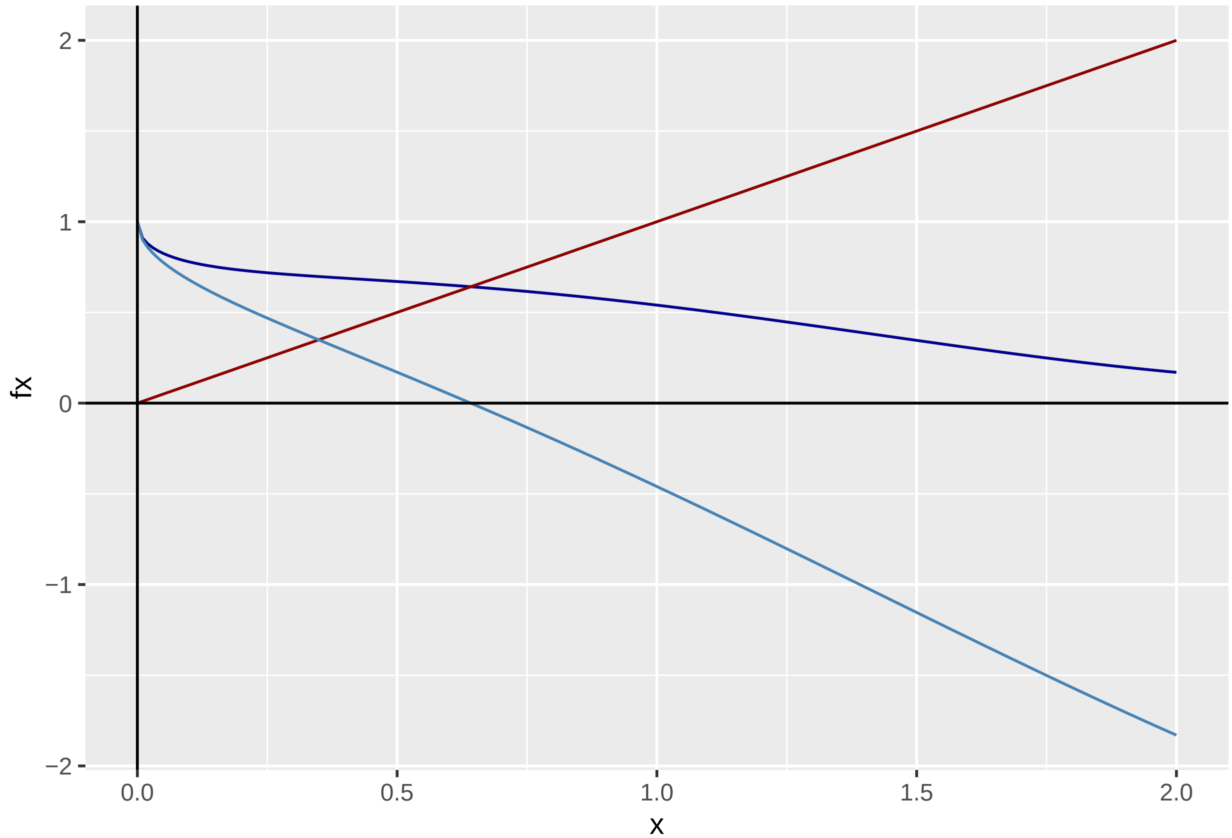
$cos(x) = \sqrt{x}$

Reescribo la función:

$x = \sqrt(x) * \frac{tan(x)}{sen(x)} * x$

Grafico la función:

```r
#La función del ejercicio
f <- function(x){
  return(-sqrt(x)+cos(x)+x)
}
#La función para graficar la raiz
g <- function(x){
  return(-sqrt(x)+cos(x))
}

#Instancio un vector que me va a indicar los puntos en la función
x <- seq(0, 2, by = 0.01)

#Genero los puntos
fx <- f(x)

#Creo un data frame con los x e y
df <- data.frame(x, fx)

#Instancio los datos
gg_fx <- ggplot(data = df)

#Agrego la capa con los datos
gg_fx <- gg_fx + aes(x = x, y = fx)

#Grafica la función del ejercicio
gg_fx <- gg_fx + geom_line(linetype = 1, colour = "darkblue")

#Gráfico x = y
gg_fx <- gg_fx + geom_line(aes(y = x), colour = "darkred")

#Gráfico la función del ejercicio donde esta la raiz
gg_fx <- gg_fx + geom_line(aes(x = x, y = g(x)), colour = "steelblue")

#Agrego el eje X
gg_fx <- gg_fx + geom_vline(xintercept = 0, linetype = 1)

#Agrego el eje Y
gg_fx <- gg_fx + geom_hline(yintercept = 0, linetype = 1)

#Grafico
gg_fx
```



Se observa una raiz en el intevalo $[0.5; 1]$

Tomo $p_0 = 1$

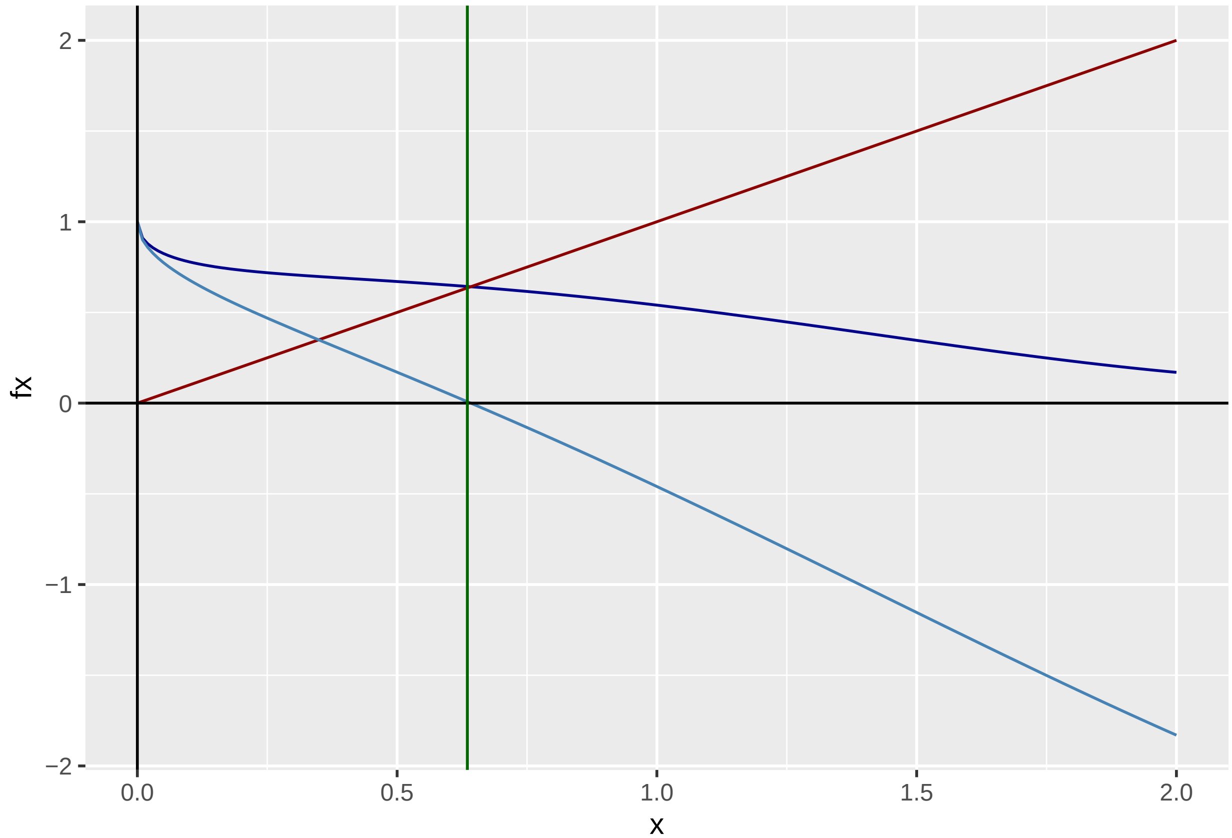
```
this_could_go_wrong <- tryCatch(
  PuntoFijo(p0 = 0.5, tol = 0.1),
  error = function(e){print("Error")})
```

```
##           P      G(P)
## 1 0.5000000 0.6704758
## 2 0.6704758 0.6351760
```

```
if (this_could_go_wrong != "Error"){
  raiz <- this_could_go_wrong
  print(paste("La raiz es: ", raiz))
}
```

```
## [1] "La raiz es:  0.635176049798848"
```

```
#Agrego la linea recta que une los puntos entre la función del ejercicio, el punto fijo y la raiz
gg_fx <- gg_fx + geom_vline(xintercept = raiz, linetype = 1, colour="darkgreen")

gg_fx
```



Verifico que se cumplan las condiciones del teorema

1. Se observa graficamente que la función es continua en todo el intervalo.
2. Evaluo la derivada en diferentes puntos del intervalo $[0.5; 1]$

```
fprima <- D(expression(-sqrt(x)+cos(x)+x), "x")

fprima
```

```
## 1 - (sin(x) + 0.5 * x^-0.5)
```

```
fprima <- function(x){
  return(1 - (sin(x) + 0.5 * x^-0.5))
}

x <- seq(0.5, 1, by =0.1)

#Genero los puntos
fprimax <- fprima(x)

#Creo un data frame con los x e y
df <- data.frame(x, fprimax)

#Lo imprimo
df
```

```
##     x    fprimax
## 1 0.5 -0.1865323
```

```
## 2 0.6 -0.2101397
## 3 0.7 -0.2418320
## 4 0.8 -0.2763731
## 5 0.9 -0.3103732
## 6 1.0 -0.3414710
```

Ejercicio 2:

$2 + cos(e^x - 2) = e^x$

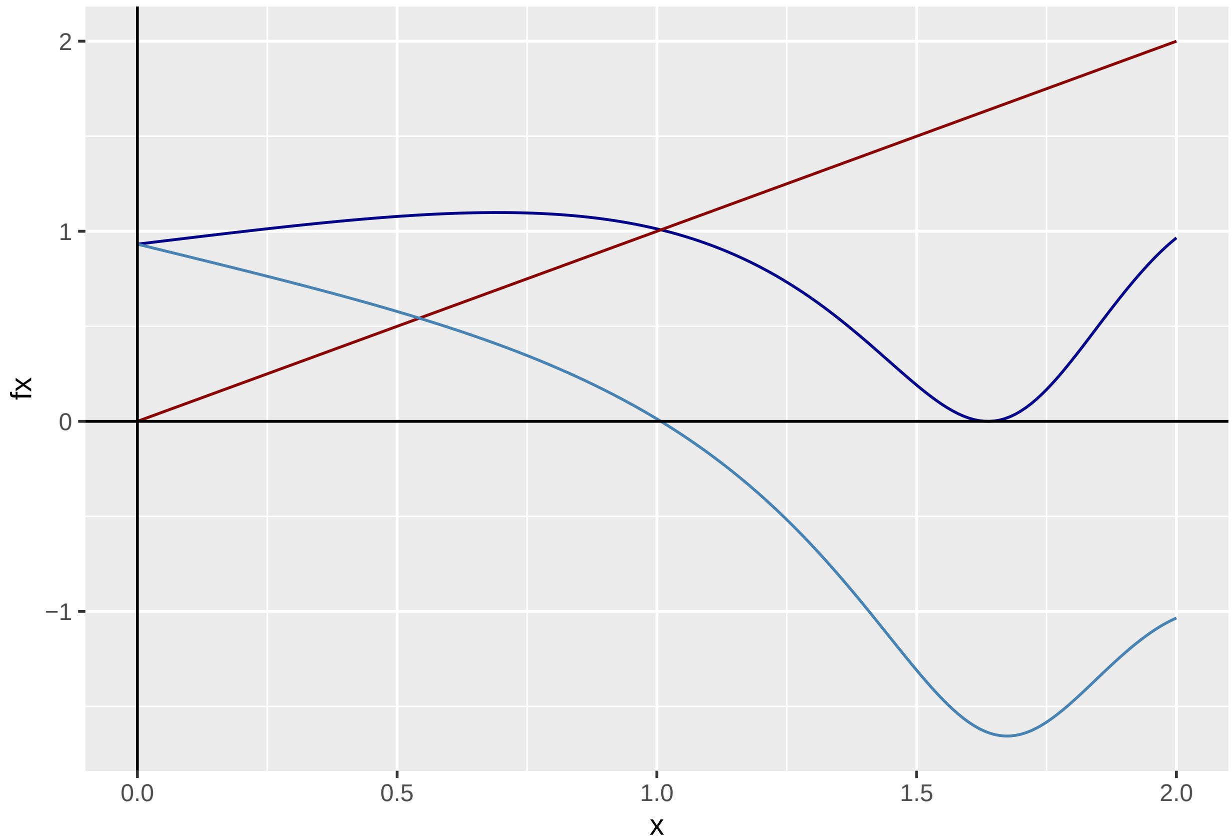
Reescribo la función:

$x = ln(2 + cos(e^x - 2))$

Grafico la función:

```
#La función del ejercicio
f <- function(x){
  return(log((2+ cos(exp(x)-2)), base = exp(1)))
}
#La función para graficar la raiz
g <- function(x){
  return(-x+log((2+ cos(exp(x)-2)), base = exp(1)))
}

#Instancio un vector que me va a indicar los puntos en la función
x <- seq(0, 2, by = 0.01)

#Genero los puntos
fx <- f(x)

#Creo un data frame con los x e y
df <- data.frame(x, fx)

#Instancio los datos
gg_fx <- ggplot(data = df)

#Agrego la capa con los datos
gg_fx <- gg_fx + aes(x = x, y = fx)

#Grafica la función del ejercicio
gg_fx <- gg_fx + geom_line(linetype = 1, colour = "darkblue")

#Gráfico x = y
gg_fx <- gg_fx + geom_line(aes(y = x), colour = "darkred")

#Gráfico la función del ejercicio donde esta la raiz
gg_fx <- gg_fx + geom_line(aes(x = x, y = g(x)), colour = "steelblue")

#Agrego el eje X
gg_fx <- gg_fx + geom_vline(xintercept = 0, linetype = 1)

#Agrego el eje Y
gg_fx <- gg_fx + geom_hline(yintercept = 0, linetype = 1)
```

```r
#Grafico
gg_fx
```



Se observa una raiz en 1

Tomo $p_0 = 1$

```r
this_could_go_wrong <- tryCatch(
  PuntoFijo(p0 = 1, tol = 0.1),
  error = function(e){print("Error")})
```

```
##   P     G(P)
## 1 1 1.012669
```

```r
if (this_could_go_wrong != "Error"){
  raiz <- this_could_go_wrong
  print(paste("La raiz es: ", raiz))
}
```

```
## [1] "La raiz es:  1.01266854332195"
```