Método Punto Fijo

Juan Angarita, Héctor Rodríguez, Aldemar Ramirez

3/8/2019

Problema

Determinar la raiz de una función de la forma f(x).

Solución

Lenguaje de programación: R

Función principal- Método de Punto Fijo

Parametros:

-f <- función

-x_1<- valor desde el que se busca en la raiz

-maxIteraciones <- cantidad máxima de iteraciones que puede realizar la función

-tol <- tolerancia mínima que debe tener la función

Valores de retorno:

"x_n" = resultados$x,

"errorAbsoluto" = resultados$errorAbsoluto, <- valor del error absoluto a través de las iteraciones

"errorRelactivo" = resultados$errorRelativo <- valor del error relativo a través de las iteraciones

```
#1e-9 = 0.000000001
puntofijo =function(g, x0, tol=1e-9, maxIteraciones=100){
  k = 1
  errorAbsoluto = c()
  errorRelativo = c()
  x = c()
    # iteración hasta que abs(x1 - x0) <= tol o se alcance maxIteraciones
    repeat{
    x1 = g(x0)
    dx = abs(x1 - x0)
    x0 = x1
    #Imprimir estado
    #cat("x_", k, "= ", x1, "\n")
    k = k+1
    x <- c(x,x1)
    errorAbsoluto <- c(errorAbsoluto,dx)
    errorRelativo <- c(errorRelativo,dx/(abs(x0)))
    #until
    if(dx< tol|| k > maxIteraciones) break;
  }
  # Mensaje de salida
```

```
  if( dx > tol ){
    cat("No hubo convergencia ")
    #return(NULL)
  } else{

    my_list <- list("resultado" = x1,
                    "errorAbsoluto" = errorAbsoluto,
                    "errorRelativo" = errorRelativo,
                    "x" = x)

    return(my_list)
  }
}
```

El método trata de encontrar un $x\ tal\ que\ x = g(x)$.

Un número $x = x*$ que satisfaga la ecuación se denomina como el punto fijo. La función verifica que la resta entre x1 y x0 sea menor o igual al nivel de tolerancia previamente establecido. Además cabe aclarar que se verifica que en los alrededores de $x = x*$ la derivada sea menor que 1. En dado caso en la derivada sea mayor a 1, o que la resta sea mayor al nivel de tolerancia, o que se alcance el límite de iteraciones se indica que no hay convergencia.

Implementacion

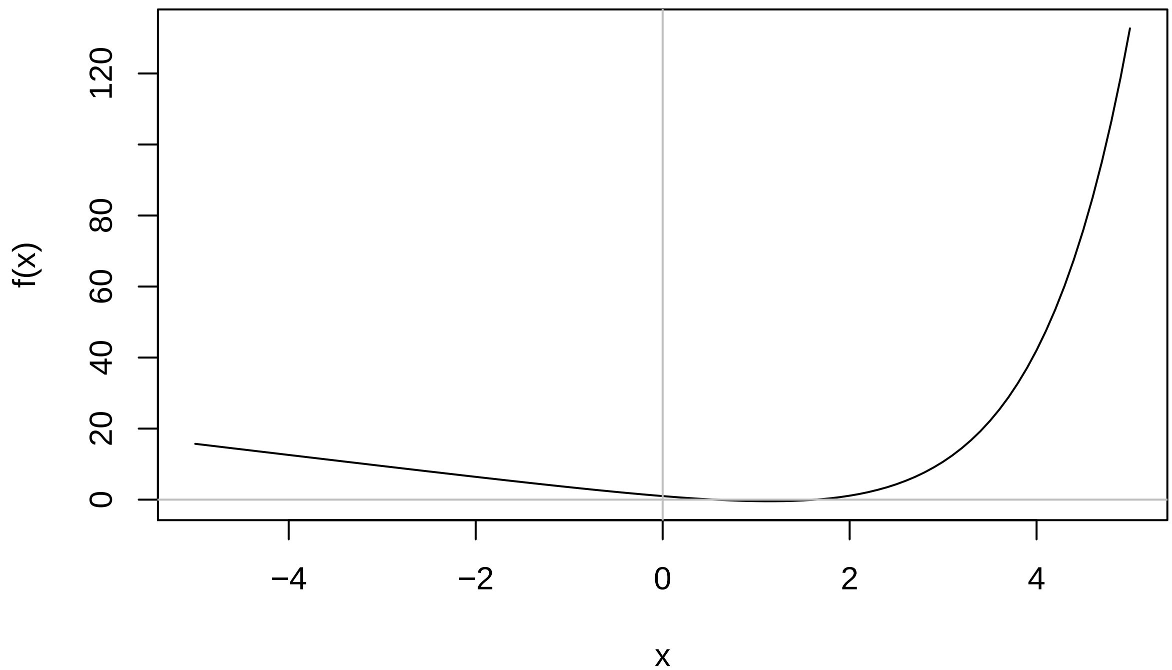
La iteración del punto fijo se hace de la siguiente manera:

Dado $x_i + _1 = g(x_i), i = 0, 2, ...$

Se ve representado en las siguientes gráficas :

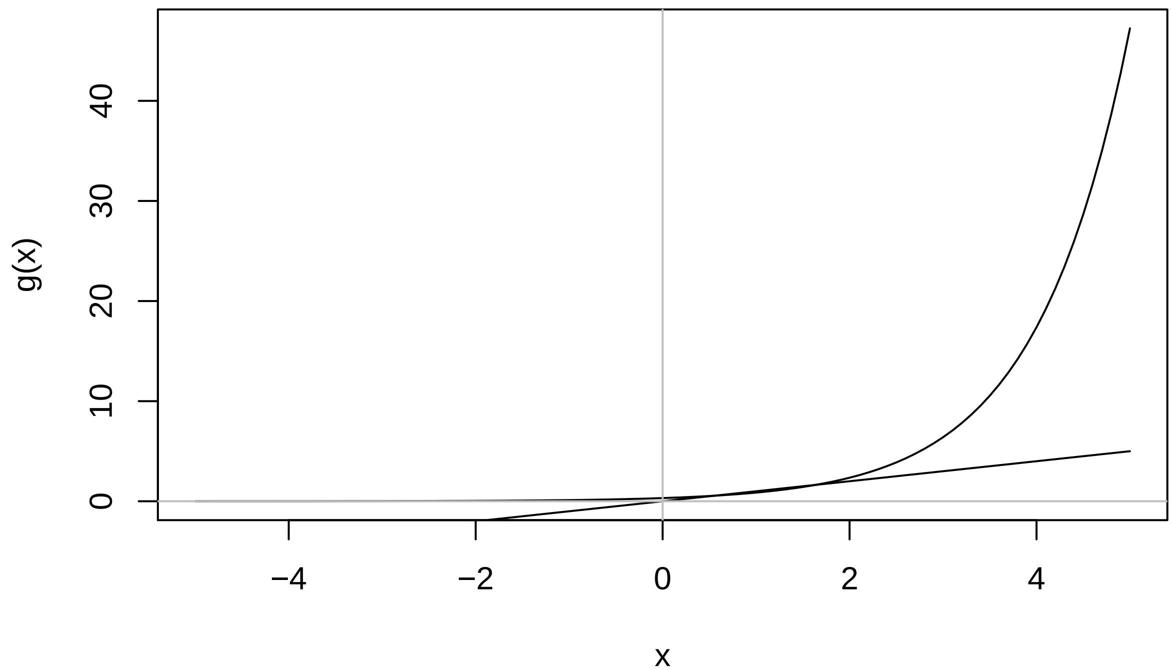
Gráfica de la función

```
f = function(x) (exp(1)^x)-(pi*x)
curve(f, -5, 5) #gráfico para decidir un intervalo
abline(h = 0, v = 0, col = "gray")
```



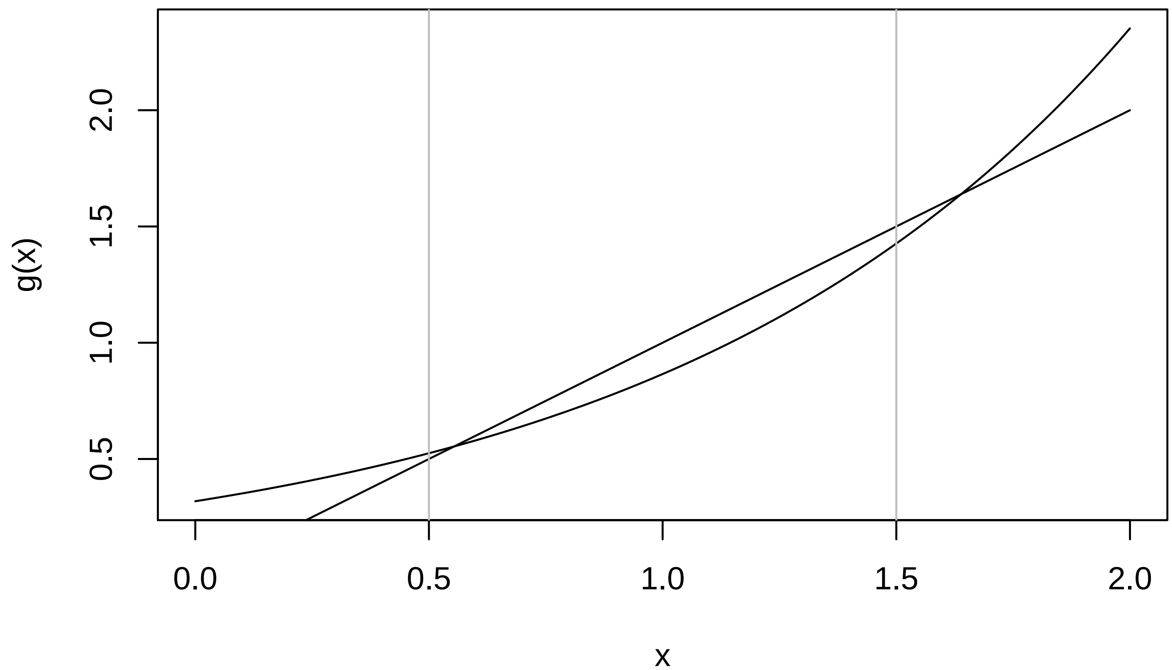
Gráfica de la función $e^x - pi * X$

```
g = function(x)  (exp(1)^x)/(pi)
y = function(x) x
curve(g, -5, 5)
curve(y, -5, 5, add=T) #gráfico para decidir un intervalo
abline(h = 0, v = 0, col = "gray")
```



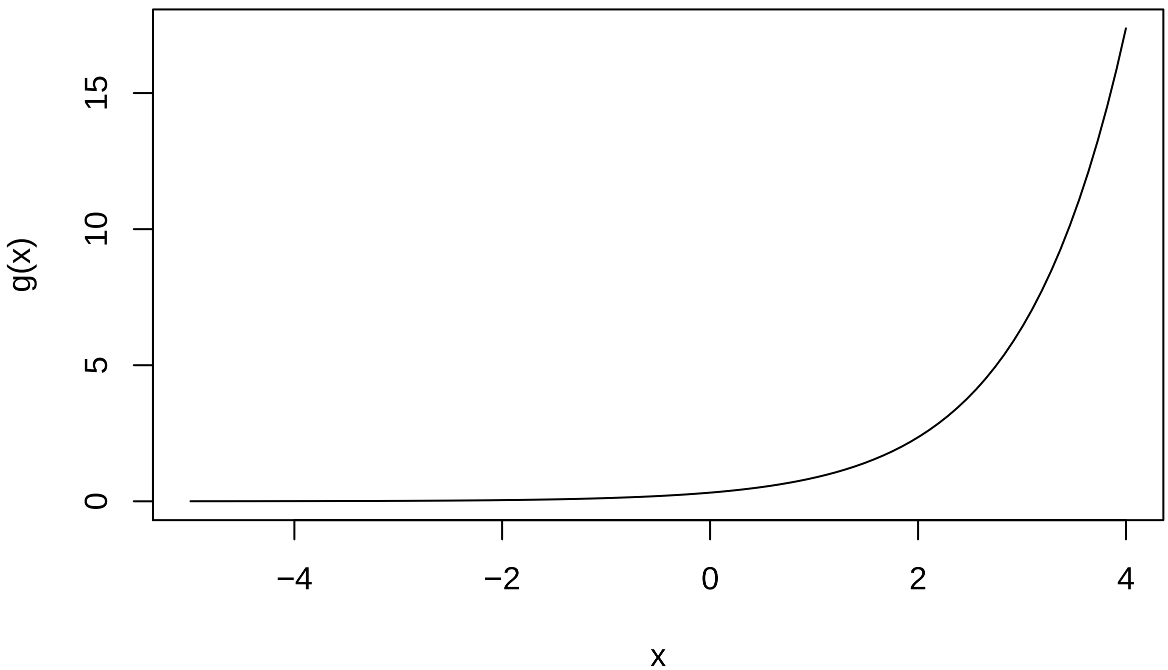
Gráfica correspondiente a la función cuando se despeja x

```
g = function(x) (exp(1)^x)/(pi)
y = function(x) x
curve(g, 0, 2)
curve(y, 0, 2, add=T)
abline(v = 0.5, col = "gray")
abline(v = 1.5, col = "gray")
```



Gráfica correspondiente al acercamiento a g sobre el punto fijo

```
dg <- function(x) {}
body(dg) <- D(body(g), 'x')
curve(dg, -5, 4, ylab = "g(x)")
```



Gráfica correspondiente a la primera derivada de g(x)

Tabla de resultados a través de las iteraciones

```r
resultados<-puntofijo(g, 1, 1e-9)
tablaErrores <- data.frame(
  "iteraciones" = 1:length(resultados$errorAbsoluto),
  "x_n" = resultados$x,
  "errorAbsoluto" = resultados$errorAbsoluto,
  "errorRelactivo" = resultados$errorRelativo
)
print(tablaErrores)
```

```
##    iteraciones       x_n errorAbsoluto errorRelactivo
## 1            1 0.8652560  1.347440e-01   1.557273e-01
## 2            2 0.7561815  1.090745e-01   1.442438e-01
## 3            3 0.6780404  7.814107e-02   1.152455e-01
## 4            4 0.6270748  5.096562e-02   8.127519e-02
## 5            5 0.5959163  3.115850e-02   5.228671e-02
## 6            6 0.5776347  1.828157e-02   3.164901e-02
## 7            7 0.5671706  1.046413e-02   1.844970e-02
## 8            8 0.5612666  5.904000e-03   1.051907e-02
## 9            9 0.5579626  3.303955e-03   5.921463e-03
## 10          10 0.5561222  1.840441e-03   3.309419e-03
## 11          11 0.5550996  1.022569e-03   1.842136e-03
## 12          12 0.5545323  5.673375e-04   1.023092e-03
## 13          13 0.5542178  3.145177e-04   5.674985e-04
```

```
## 14          14 0.5540435  1.742839e-04   3.145672e-04
## 15          15 0.5539469  9.655244e-05   1.742991e-04
## 16          16 0.5538934  5.348234e-05   9.655710e-05
## 17          17 0.5538638  2.962273e-05   5.348378e-05
## 18          18 0.5538474  1.640671e-05   2.962317e-05
## 19          19 0.5538383  9.086741e-06   1.640685e-05
## 20          20 0.5538333  5.032563e-06   9.086782e-06
## 21          21 0.5538305  2.787194e-06   5.032575e-06
## 22          22 0.5538290  1.543631e-06   2.787198e-06
## 23          23 0.5538281  8.549067e-07   1.543632e-06
## 24          24 0.5538276  4.734712e-07   8.549071e-07
## 25          25 0.5538274  2.622213e-07   4.734713e-07
## 26          26 0.5538272  1.452253e-07   2.622214e-07
## 27          27 0.5538271  8.042974e-08   1.452253e-07
## 28          28 0.5538271  4.454417e-08   8.042974e-08
## 29          29 0.5538271  2.466977e-08   4.454417e-08
## 30          30 0.5538271  1.366278e-08   2.466977e-08
## 31          31 0.5538270  7.566820e-09   1.366278e-08
## 32          32 0.5538270  4.190709e-09   7.566820e-09
## 33          33 0.5538270  2.320928e-09   4.190709e-09
## 34          34 0.5538270  1.285393e-09   2.320928e-09
## 35          35 0.5538270  7.118853e-10   1.285393e-09
```

```r
cat("Resultado: iteraciones",length(resultados$errorAbsoluto),"\n")
```

```
## Resultado: iteraciones 35
```

```r
cat("x = ",resultados$x[length(resultados$errorAbsoluto)],"\n")
```

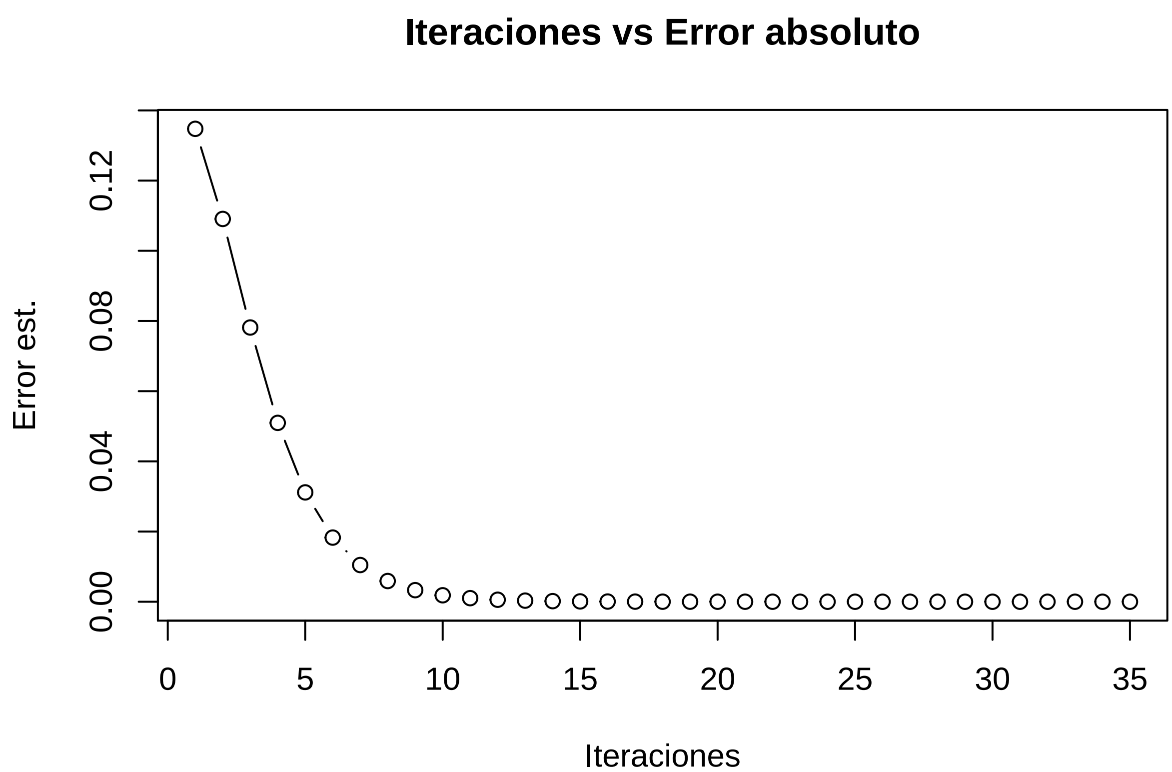
```
## x =  0.553827
```

```r
cat("Error estimado <= ", resultados$errorRelativo[length(resultados$errorAbsoluto)],"\n")
```

```
## Error estimado <=  1.285393e-09
```

Gráfica de iteraciones vs error absoluto

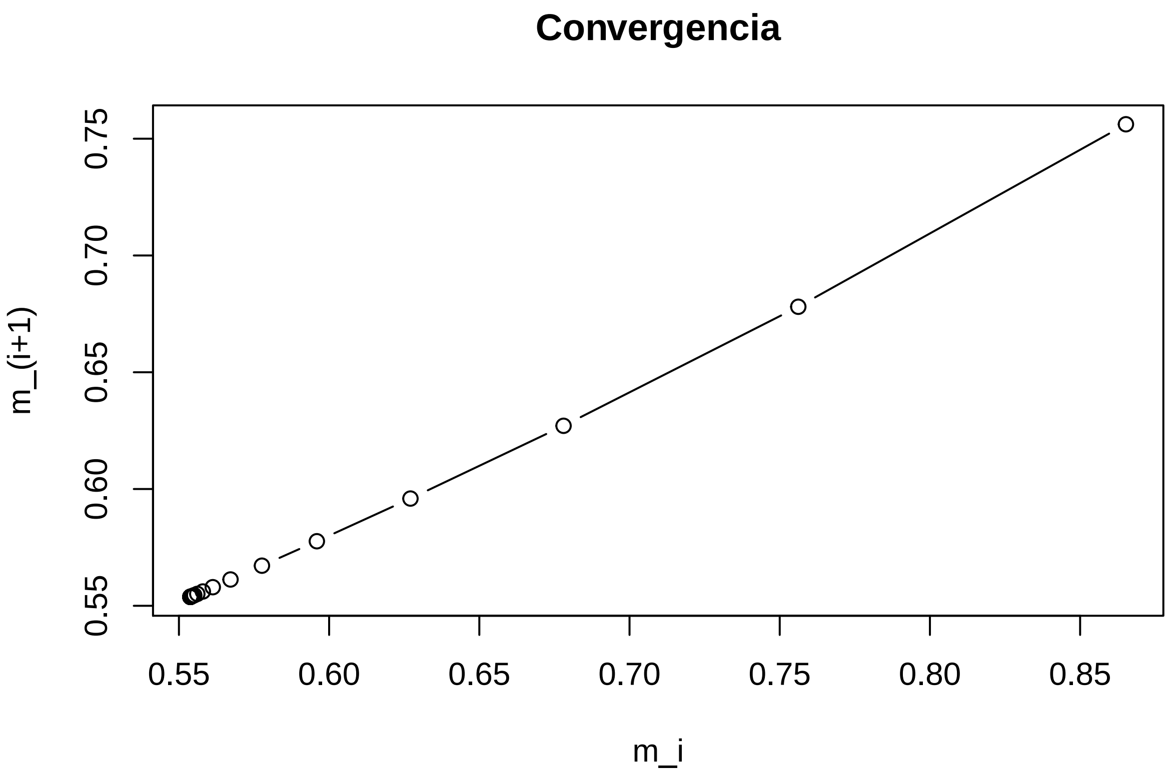
```r
plot(x = 1:length(resultados$x), y = resultados$errorAbsoluto, xlab = "Iteraciones",
     ylab = "Error est.", type="b", main = "Iteraciones vs Error absoluto")
```



La gráfica denota claramente como a medida que incrementan las iteraciones el error va disminuyendo y tiende a cero.

Grafica de x(i) vs x(i+1)

```
m_i =  resultados$x[-length(resultados$x)]
m_i2 = resultados$x
m_i2 = m_i2[-1]
plot(x =m_i, y =m_i2, xlab = "m_i", ylab = "m_(i+1)", type="b",main = "Convergencia")
```

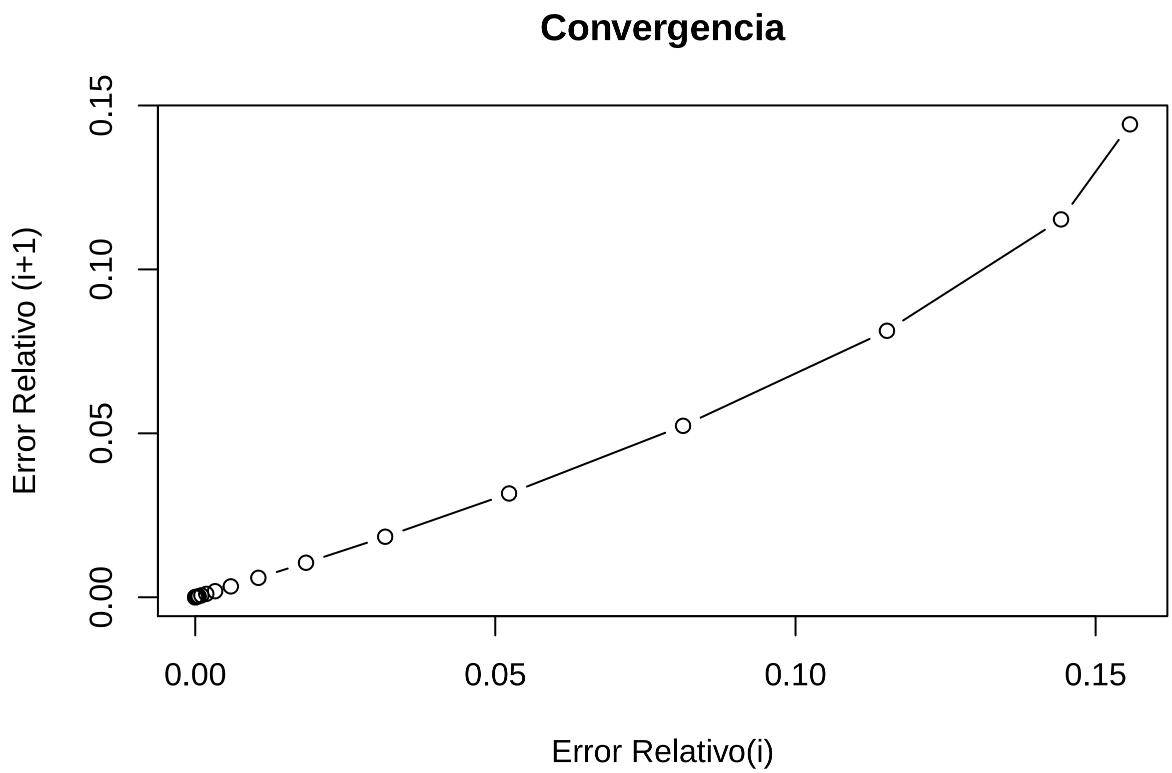


La grafica nos confirma como el método de punto fijo tiene una convergencia que presenta una tendencia lineal.

Gráfica de Error Relativo(i) vs Error Relativo(i+1)

```
m_i = resultados$errorRelativo[-length(resultados$errorRelativo)]
m_i2 = resultados$errorRelativo
m_i2 = m_i2[-1]


plot(x =m_i, y = m_i2, xlab = "Error Relativo(i)",
     ylab = "Error Relativo (i+1)",
     main = "Convergencia", type = "b")
```

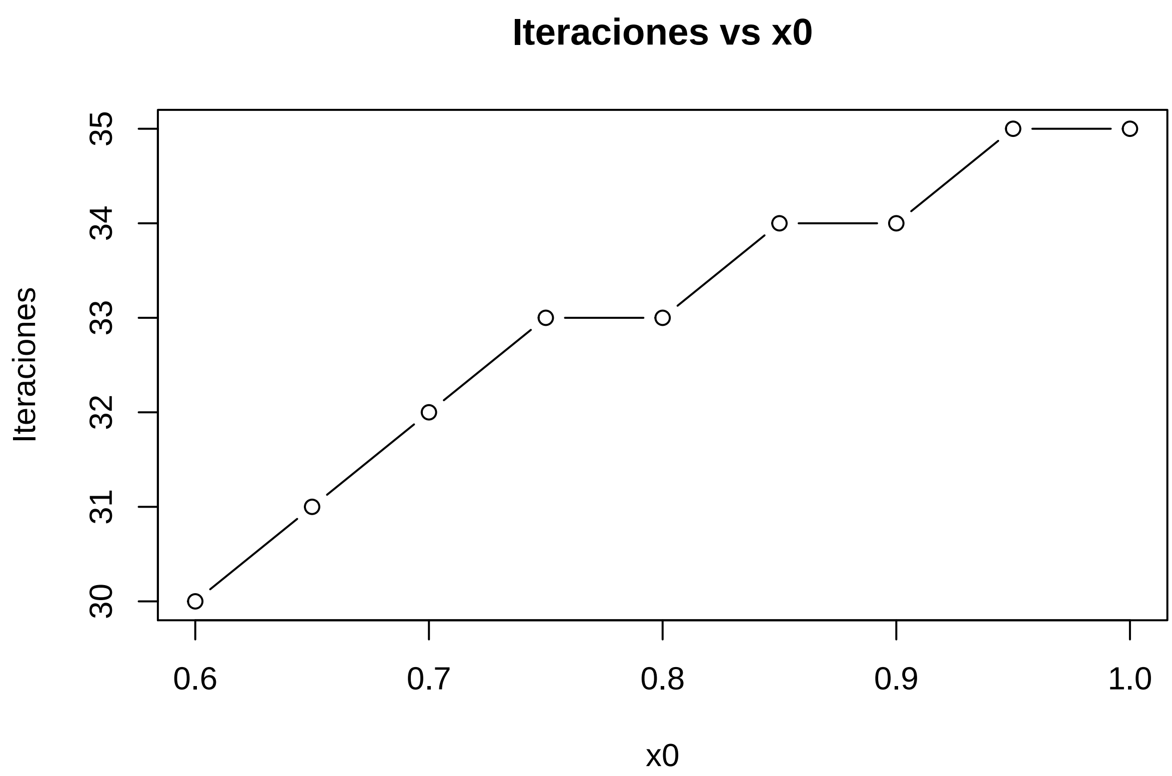


Grafica de iteraciones vs x0 (Aproximación por derecha)

```
l=1
v = c()
iteraciones = c()
while(l>=0.550) {
  v <- c(v,l)
  resultados<-puntofijo(g, l, 1e-9)

  tablaErrores <- data.frame(
    "iteraciones" = 1:length(resultados$errorAbsoluto),
    "x_n" = resultados$x,
    "errorAbsoluto" = resultados$errorAbsoluto,
    "errorRelactivo" = resultados$errorRelativo
  )
  iteraciones <- c(iteraciones,length(resultados$errorAbsoluto))
  l = l-0.050
  #print(tablaErrores)
}

plot(x =v, y =iteraciones, xlab = "x0", ylab = "Iteraciones", type="b",main = "Iteraciones vs x0")
```



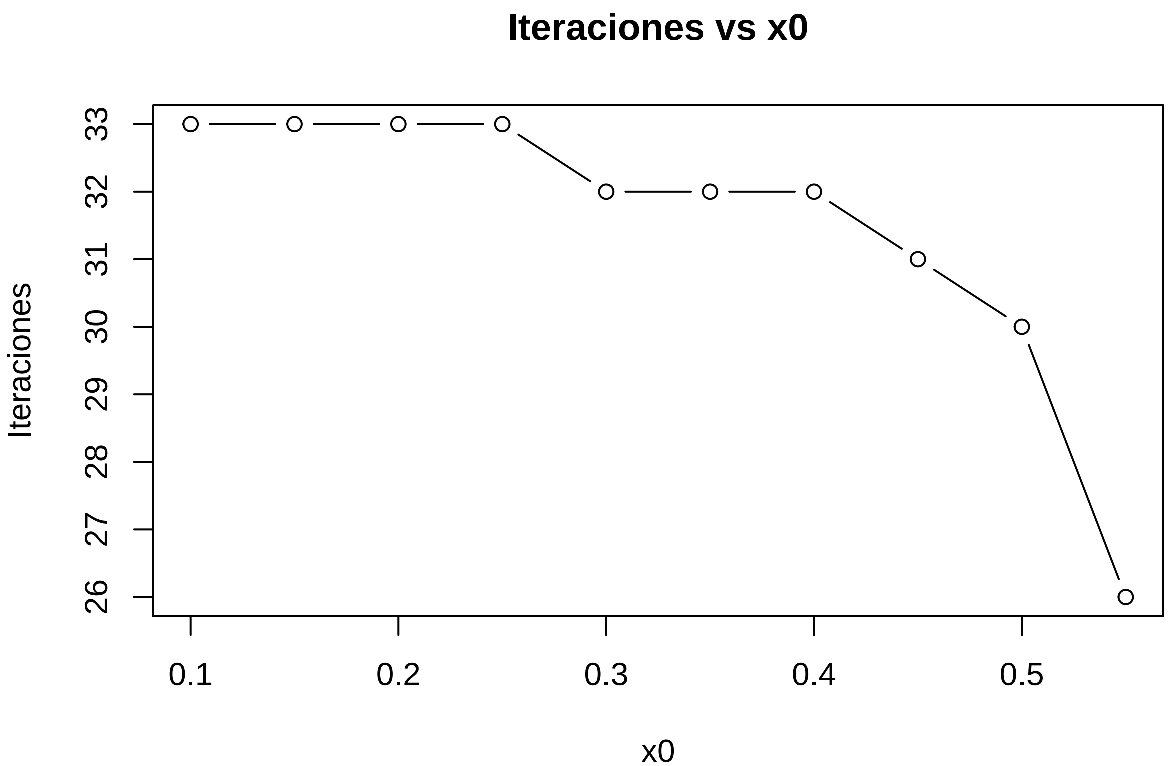
Entre más alejado esté el valor incial del valro real de la raíz, más iteraciones va a tomar el algoritmo.

Grafica de iteraciones vs x0 (Aproximación por izquierda)

```
l=0.1
v = c()
iteraciones = c()
while(l<=0.550) {
  v <- c(v,l)
  resultados<-puntofijo(g, l, 1e-9)

  tablaErrores <- data.frame(
    "iteraciones" = 1:length(resultados$errorAbsoluto),
    "x_n" = resultados$x,
    "errorAbsoluto" = resultados$errorAbsoluto,
    "errorRelactivo" = resultados$errorRelativo
  )
  iteraciones <- c(iteraciones,length(resultados$errorAbsoluto))
  l = l+0.050
  #print(tablaErrores)
}

plot(x =v, y =iteraciones, xlab = "x0", ylab = "Iteraciones", type="b",main = "Iteraciones vs x0")
```

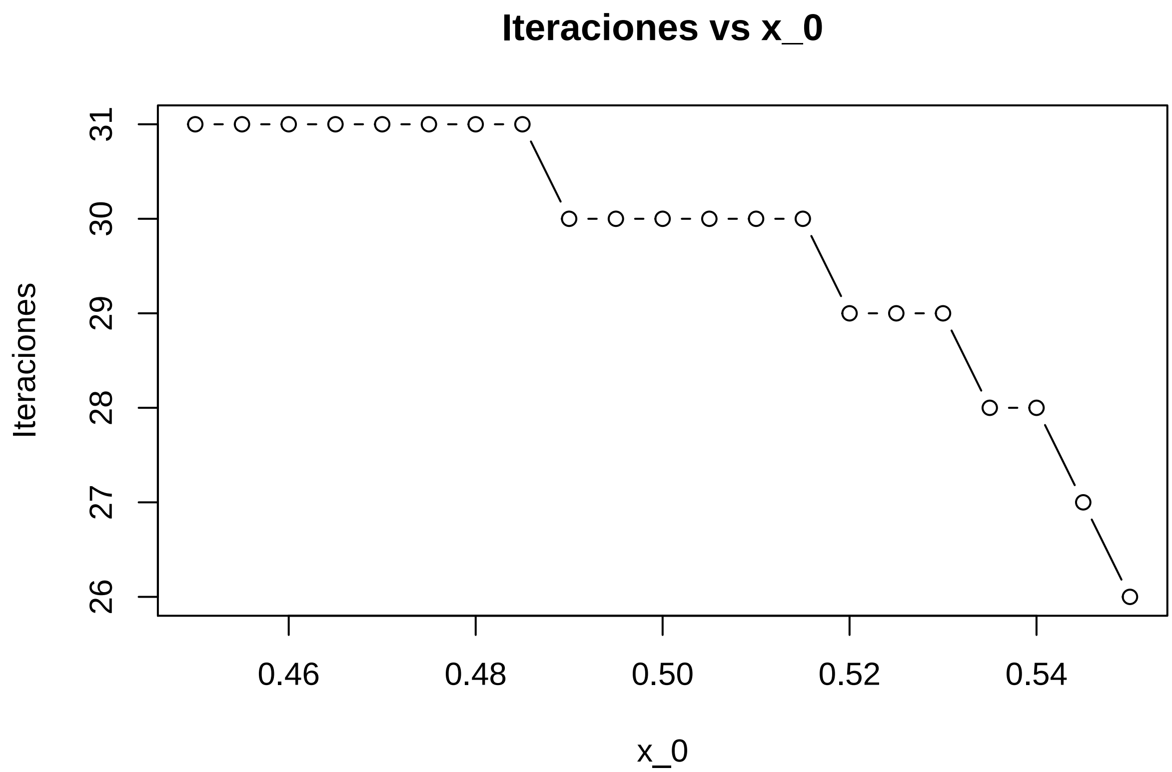


Cuando comparamos la gráfica de aproximación por la izquirda con la de aproximación por derecha encontramos que ambas son prácticamente igual de eficientes, teniendo la aproximación por izquierda una ligera ventaja.

Grafica de iteraciones vs x0 (Aproximacion de x__0 a la raiz)

```r
l=0.45
v = c()
iteraciones = c()
while(l<=0.550) {
  v <- c(v,l)
  resultados<-puntofijo(g, l, 1e-9)

  tablaErrores <- data.frame(
    "iteraciones" = 1:length(resultados$errorAbsoluto),
    "x_n" = resultados$x,
    "errorAbsoluto" = resultados$errorAbsoluto,
    "errorRelactivo" = resultados$errorRelativo
  )
  iteraciones <- c(iteraciones,length(resultados$errorAbsoluto))
  l = l+0.005
  #print(tablaErrores)
}
plot(x =v, y =iteraciones, xlab = "x_0", ylab = "Iteraciones",
     type="b",main = "Iteraciones vs x_0")
```

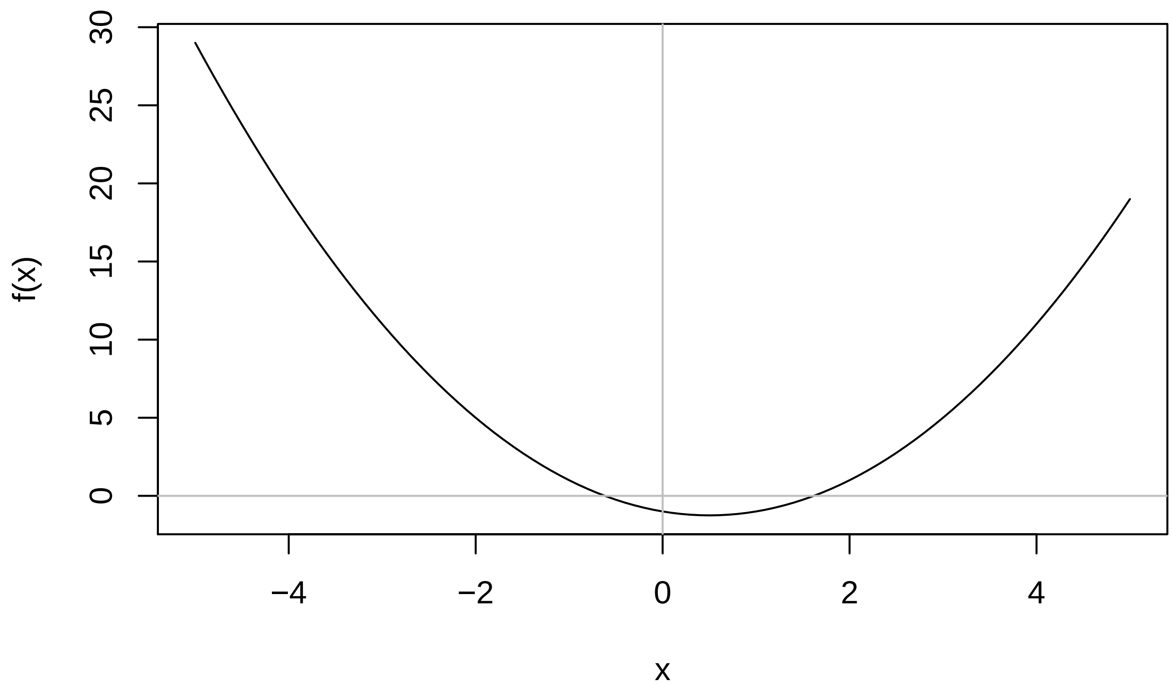


Al realizar una aproximación de x_0 con un valor cercano a la raiz encontramos que el número de iteraciones no cambia con variaciones bajas de x_0(es este caso variamos x_0 de a 0.004).

Caso especial: No hay convergencia

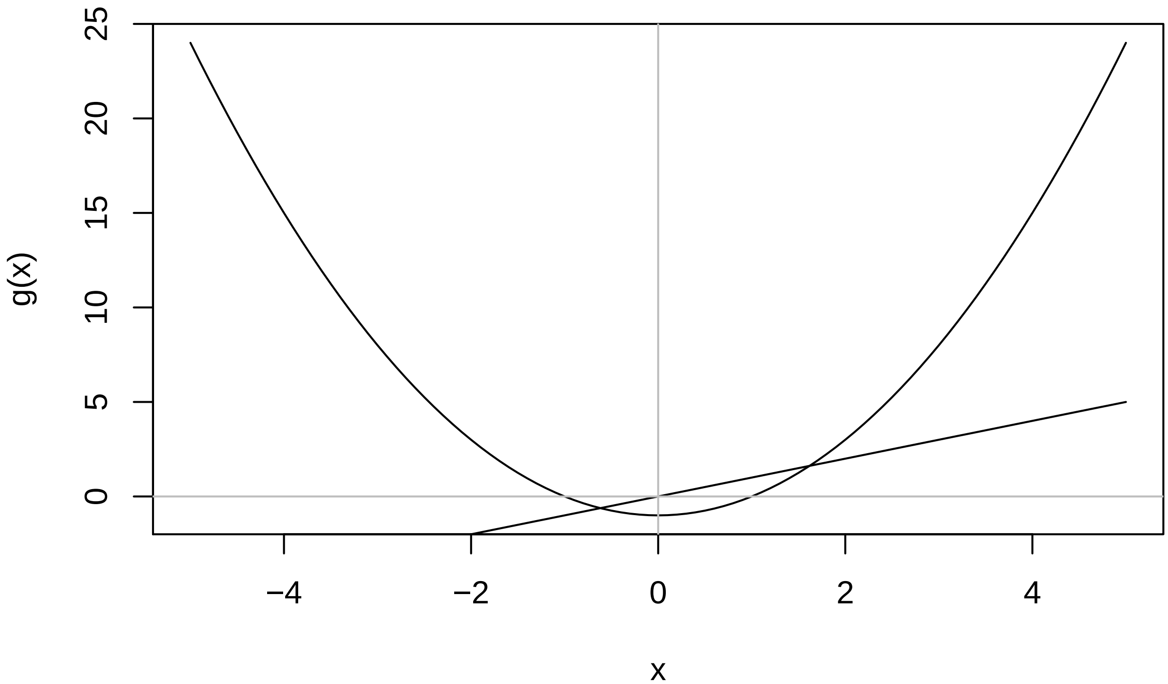
Gráfica de la función

```
f = function(x) (x^2)-1-x
curve(f, -5, 5) #gráfico para decidir un intervalo
abline(h = 0, v = 0, col = "gray")
```



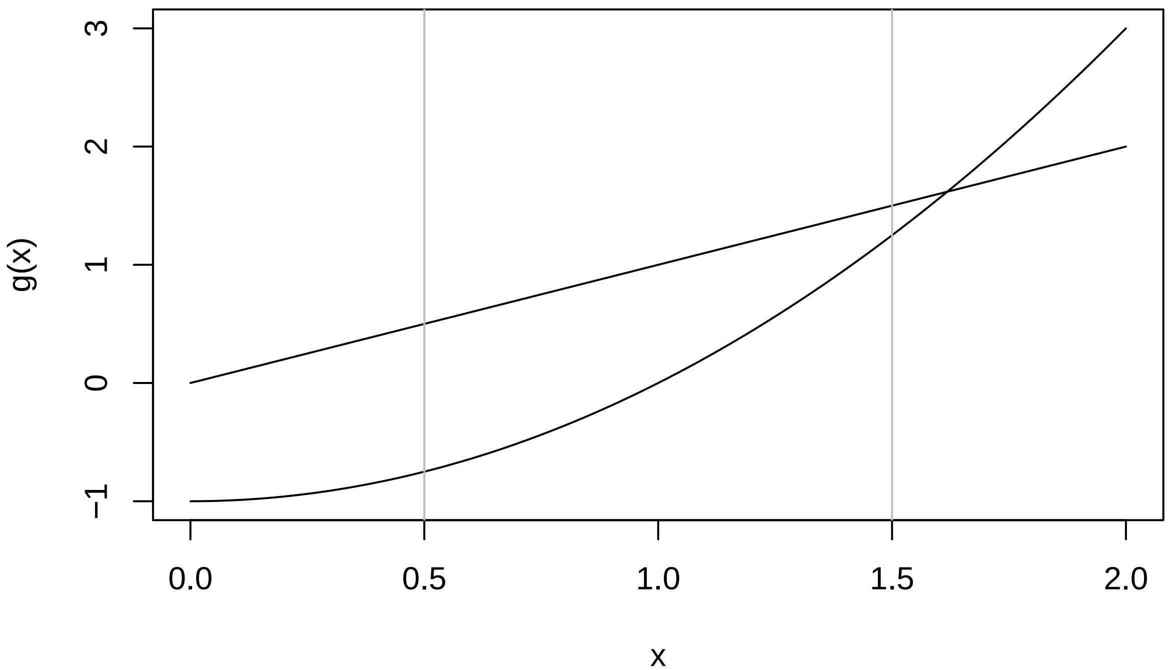
Gráfica de la función \$ (x^2)-1\$

```
g = function(x) (x^2)-1
y = function(x) x
curve(g, -5, 5)
curve(y, -5, 5, add=T) #gráfico para decidir un intervalo
abline(h = 0, v = 0, col = "gray")
```



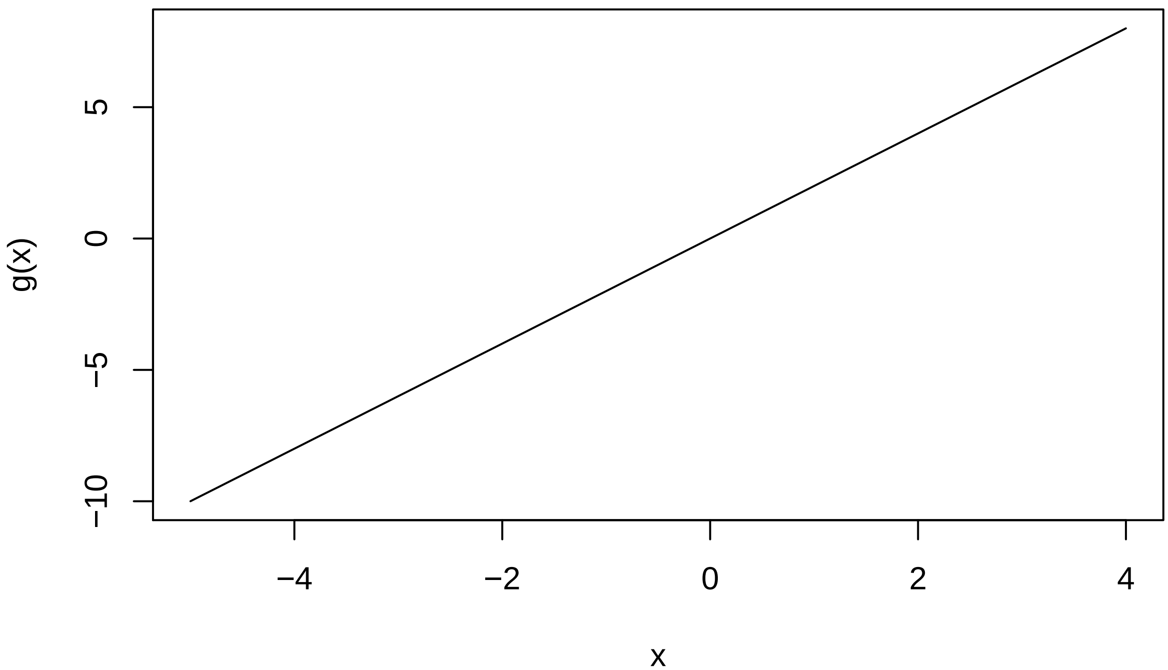
Gráfica correspondiente a la función cuando se despeja x

```
g = function(x) (x^2)-1
y = function(x) x
curve(g, 0, 2)
curve(y, 0, 2, add=T)
abline(v = 0.5, col = "gray")
abline(v = 1.5, col = "gray")
```



Gráfica correspondiente al acercamiento a g sobre el punto fijo

```
dg <- function(x) {}
body(dg) <- D(body(g), 'x')
curve(dg, -5, 4, ylab = "g(x)")
```



Gráfica correspondiente a la primera derivada de g(x)

Tabla de resultados a traves de las iteraciones

```r
resultados<-puntofijo(g, 1.5, 1e-9)
```

```
## No hubo convergencia
```

Vemos entonces como se alcanzó el número máximo de iteraciones sin encontrar convergencia alguna.