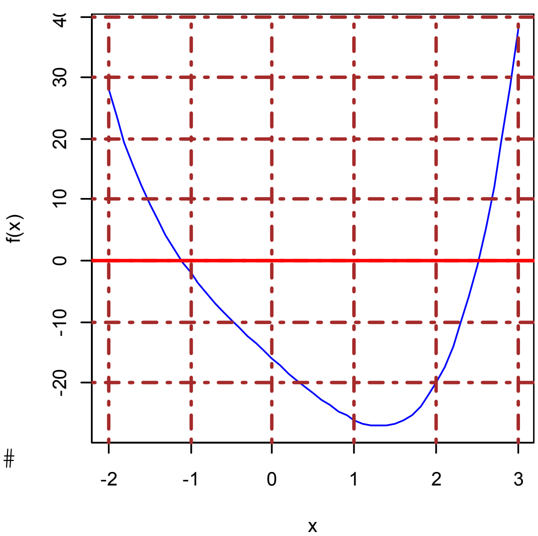
Universidad Nacional Agraria – La Molina
Dpto. Estadística e Informática
Métodos numéricos y simulación
Practica # 2 (26 setiembre de 2012)

Puede utilizar su calculadora programable para facilitar su trabajo. En cada caso, debe localizar la raíz y establecer el error máximo posible que pueda tener la raíz.

1. Dada la función: f(x)= x^4 + x^2 –12*X – 16
Hallar la raíz con al menos 2 cifras decimales significativos por el método de bisección



\#

En R:

```
# Localización de raíces
f<-function(x)x^4+x^2-12*x-16
x<-seq(-3,3,0.1)
plot(x,f(x),type="l",col="blue")
grid(col="brown",lty=4,lwd=2)
abline(h=0,col="red",lwd=2)
# Hallar las raices con R.
cbind(polyroot(c(-16,-12,1,0,1)))
Raices:
 [1,] -1.106431
 [2,] -0.702926+2.293901i
 [3,] -0.702926-2.293901i
 [4,]  2.512282
```

```
Raices reales:
# X1 = -1.106431
# X2 = 2.512282
#
Raíz negativa en I1 = ]-2, -1[
Raíz positiva en I2 = ]2, 3[

La error máximo que se puede cometer es  0.5*10^(m-2+1) =
Error = 0.5*10^(-1)=0.05.

Por Bissecion, Raiz negativa
a=-2
b=-1
d = (a+b)/2 = -1.5
E=abs(a-d) = 0.5 > 0.05
# Ubicando el nuevo intervalo
f(a)*f(d) = 260 es positivo
a=-1.5
b=-1
d = (a+b)/2 = -1.25
E=abs(a-d) = 0.25 > 0.05

# Se continúa el proceso
f(a)*f(d) = 27 es positivo
```

```
a=-1.25
b=-1
d = (a+b)/2 = -1.125
E=abs(a-d) = 0.125 > 0.05

f(a)*f(d) = 1.10373 es positivo
a=-1.125
b=-1
d = (a+b)/2 = -1.0625
E=abs(a-d) = 0.0625 > 0.05
f(a)*f(d) = -0.3110913 es negativo
a=-1.125
b=-1.0625
d = (a+b)/2 = -1.09375
E=abs(a-d) = 0.03125 < 0.05 Satisface.

X1 = -1.09375 +/- 0.03125
```

i	**a**	**b**	**d**	**E**	**error**
0	-2	-1	-1.5	0.5	0.05
1	-1.5	-1	-1.25	0.25	0.05
2	-1.25	-1	-1.125	0.125	0.05
3	-1.125	-1	-1.0625	0.0625	0.05
4	-1.0625	-1	-1.09375	0.03125	0.05

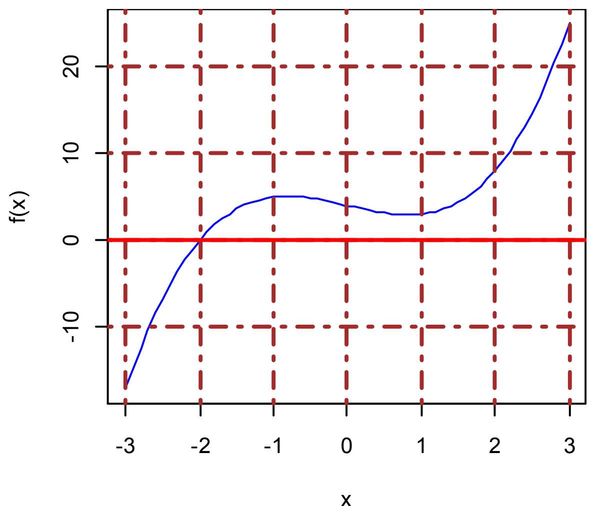
Aplicando a la segunda raíz.

```
a=2; b=3; d=(a+b)/2; E=abs(a-d)
data.frame(a,b,d,f(a)*f(d)<0,E,E<0.05)
  a b   d f.a....f.d....0   E E...0.05
1 2 3 2.5             FALSE 0.5     FALSE
a=d; d=(a+b)/2; E=abs(a-d)
data.frame(a,b,d,f(a)*f(d)<0,E,E<0.05)
    a b    d f.a....f.d....0    E E...0.05
1 2.5 3 2.75             TRUE 0.25     FALSE
a=2.5; b=d; d=(a+b)/2; E=abs(a-d)
data.frame(a,b,d,f(a)*f(d)<0,E,E<0.05) # Es positivo
    a    b     d f.a....f.d....0     E E...0.05
1 2.5 2.75 2.625             TRUE 0.125     FALSE
a=2.5; b=d; d=(a+b)/2; E=abs(a-d)
data.frame(a,b,d,f(a)*f(d)<0,E,E<0.05) # Es positivo
    a     b      d f.a....f.d....0      E E...0.05
1 2.5 2.625 2.5625             TRUE 0.0625     FALSE
a=2.5; b=d; d=(a+b)/2; E=abs(a-d)
data.frame(a,b,d,f(a)*f(d)<0,E,E<0.05) # Es positivo
    a      b       d f.a....f.d....0       E E...0.05
1 2.5 2.5625 2.53125             TRUE 0.03125      TRUE
```

i	**a**	**b**	**d**	**E**	**error**
0	2	3	2.5	0.5	0.05
1	2.5	3	2.75	0.25	0.05
2	2.5	2.75	2.625	0.125	0.05
3	2.5	2.625	2.6625	0.0625	0.05
4	2.5	2.5625	2.53125	0.03125	0.05

En la quinta iteración, X2=2.53125 +/- 0.03125

2. **El siguiente polinomio x^3 -2x + 4 tiene una solución cercana a cero. Ilustre el proceso para determinar el grado de convergencia al aplicar el método de Newton. Hallar la raíz redondeado a 2 cifras decimales significativos.**



En R:

```
# Localización de raíces
f<-function(x)x^3-2*x+4
f1<-function(x) 3*x^2-2
x<-seq(-3,3,0.1)
plot(x,f(x),type="l",col="blue")
grid(col="brown",lty=4,lwd=2)
abline(h=0,col="red",lwd=2)

cbind(polyroot(c(4,-2,0,1)))
Raices:
[1,]  1 + 1i
[2,] -2
[3,]  1 - 1i
```

```
La raíz mas cerca a 0 es negativa y esta en el intervalo ]-1, -3[

Para newton se requiere un valor inicial, X0 = -1

Xi+1 = Xi - f(Xi) / f1(Xi)

Xi+1 = Xi - (Xi^3-2*Xi+4) /(3*Xi^2-2)

error = 0.5*10^(m-n+1) = 10^(0-2+1) = 0.1

Con R:
Con X <- -1
error = 0.1
i<-1
X<-NULL ; E<-NULL
X[1]<- -1
E[1]<-1 # Un error grande
while( E[i] > error) {
i<-i+1
X[i]<-X[i-1]-f(X[i-1])/f1(X[i-1])
E[i]<- abs(f(X[i])/f1(X[i]))
}
cbind(X,E)
             X          E
[1,] -1.000000 1.00000000
[2,] -6.000000 1.88679245
[3,] -4.113208 1.17654158
[4,] -2.936666 0.64730573
[5,] -2.289360 0.24922241
[6,] -2.040138 0.03920372

X* = -2.040138 +/- 0.03920372
```

```
con X<- -3

cbind(X,E)
              X          E
[1,] -3.000000 1.00000000
[2,] -2.320000 0.27193848
[3,] -2.048062 0.04673103

X* = -2.048062 +/- 0.04673103
```

3. **Suponga que desea hallar el valor de la raíz cúbica de x mediante el método de falsa posición. aplicar a la raíz cúbica de 4. Sugerencia: x^3 - 4 = 0 representa a la raíz cubica de 4.**

```
La raíz cubica de 4 es 1.587401  (Hallado con la calculadora)

Planteado como una función: X^3 – 4 = 0

Dado que se conoce el verdadero valor, E = |Xi –X*|

Aplicando el método de falsa posición se requiere un intervalo ]a, b[

f<-function(x) x^3 - 4
f1<-function(x) 3*x^2
a<- 1
b<- 2

d <- (a*f(b)-b*f(a))/(f(b)-f(a))
d
1.428571

f(a)*f(d)
3.253644, Es positivo.
a<-d
E<-abs(a-b)
E
[1] 0.5714286
E1 <- abs(f(d)/f1(d))

d <- (a*f(b)-b*f(a))/(f(b)-f(a))
d
1.550459

f(a)*f(d)
0.2958841, Es positivo
a<-d
E<-abs(a-b)
E
0.4495413
E1 <- abs(f(d)/f1(d))

d <- (a*f(b)-b*f(a))/(f(b)-f(a))
d
1.579162
f(a)*f(d)
0.01690445, Es positivo
```

```
a<-d
E<-abs(a-b)
E
0.4208382
E1 <- abs(f(d)/f1(d))

d <- (a*f(b)-b*f(a))/(f(b)-f(a))
d
1.585581
f(a)*f(d)
0.0008513812, Es positivo
a<-d
E<-abs(a-b)
E
0.4208382
E1 <- abs(f(d)/f1(d))
```

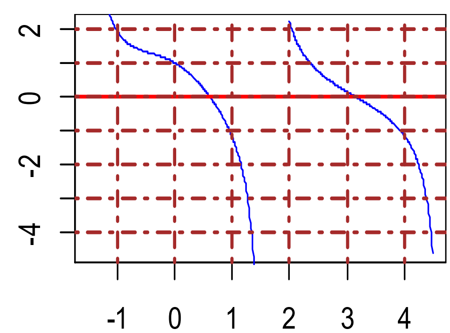
i	a	b	d	E	E1	E.real
0	1	2	1.428571	0.5714286	0.1771429	0.15883
1	1.428571	2	1.550459	0.4495413	0.03782954	0.036942
2	1.550459	2	1.579162	0.4208382	0.008282356	0.008239
3	1.579162	2	1.585581	0.4144186	0.001821786	0.00182

```
X1 = 1.585581 +/- 0.00182
```

4. Dada la función: $f(x)= \exp(-x^2) – tg(x)$.

Hallar la raíz por el método de la secante con un margen de error del 10%

```
Con R:
par(mar=c(3,3,0,0))
f<-function(x) exp(-x^2)- tan(x)
x<-seq(2.0,4.5,0.01)
plot(x,f(x),type="l",col="blue",
xlim=c(-1.5,4.5))
x<-seq(-1.5,1.5,0.01)
lines(x,f(x),col="blue")
abline(h=0,col="red",lwd=2)
grid(col="brown",lty=4,lwd=2)
```



Una raíz está en el intervalo]0,1[, otra en]2.5 y 3.5[

Por secante: $X_{i+1} = (X_{i-1}*f(X_i)- X_i*f(X_{i-1}))/(f(X_{i-1})-f(X_i))$

```
F<-expression(exp(-x^2)- tan(x))
Fx<-D(F,"x")
Fx
-(exp(-x^2) * (2 * x) + 1/cos(x)^2)
X<-NULL; E<-NULL
X[1]<-0; X[2]<-1; E[1:2]<-1; E1<-E; E.real<-E; iter<-NULL; iter[1:2]<-0
error <- 0.1  # 0.10*|X|
# raíz exacta: 0.605855437424296
exacto<- 0.605855437424296
i<-1
while( E[i] > error) {
i<-i+1
X[i+1]<-(X[i-1]*f(X[i])- X[i]*f(X[i-1]))/ (f(X[i])- f(X[i-1]))
E[i+1]<- abs(X[i-1]-X[i])
```

```
x<-X[i+1]
E1[i+1] <- abs(f(x)/eval(Fx))
E.real[i+1] <- abs(x-exacto)
iter[i+1] <- i-1
}

cbind(iter,X,E,E1,E.real)
     iter         X           E           E1       E.real
[1,]    0 0.0000000 1.000000000 1.000000e+00 1.000000e+00
[2,]    0 1.0000000 1.000000000 1.000000e+00 1.000000e+00
[3,]    1 0.4567194 1.000000000 1.615591e-01 1.491361e-01
[4,]    2 0.5719887 0.543280620 3.446857e-02 3.386671e-02
[5,]    3 0.6085748 0.115269348 2.715476e-03 2.719331e-03
[6,]    4 0.6058073 0.036586040 4.814189e-05 4.814068e-05
[7,]    5 0.6058554 0.002767471 6.825481e-08 6.825481e-08

> round(E*100/E2,2)
100   100  670.53    1604.17    4238.89   75998.18 4054617.29

> round(E1*100/E2,2)
100 100 108.33 101.78  99.86 100.00 100.00
```

X1 = 0.6058554 +/- 6.8x10^(-08)

Otra raíz puede ser encontrada en el intervalo]2,4[

```
X<-NULL; E<-NULL
X[1]<-2; X[2]<-4; E[1:2]<-1; E1<-E; E.real<-E; iter<-NULL; iter[1:2]<-0
error <- 0.1  # 0.10*|X|
# raíz exacta: 3.141644359974672
exacto<- 3.141644359974672
. . .
cbind(iter,X,E,E1,E.real)

     iter        X            E           E1       E.real
[1,]    0 2.000000 1.0000000000 1.000000e+00 1.000000e+00
[2,]    0 4.000000 1.0000000000 1.000000e+00 1.000000e+00
[3,]    1 3.311062 2.0000000000 1.662078e-01 1.694176e-01
[4,]    2 3.191604 0.6889380017 4.987834e-02 4.995954e-02
[5,]    3 3.142258 0.1194581003 6.133730e-04 6.133728e-04
[6,]    4 3.141645 0.0493461653 4.910749e-07 4.910749e-07
[7,]    5 3.141644 0.0006128817 2.139438e-13 2.140510e-13
```