Análisis numérico Taller 1
Johan García, Esteban Casas, Camilo Cruz
Punto 1

```
#include
<iostream>
            #include <conio.h>
            #include <math.h>
            using namespace std;

            int main ( )
             {
                int grado, m, remp;
                cout << "Grado del polimonoi";
                cout<<endl;
                cin >> grado;
                int a[grado], b[grado];
                cout << "Ingrese los coeficientes con su signo correspondiente ";
                cout<<endl;
                for( int i=0; i<=grado; i++){
                    m = grado-i;
                    cout << "ingrese le numero que acompania la base del exponente a("<<
            m <<") : > ";
                    cout<<endl;
                    cin >> a[grado-i];
                }
                 cout << " usted ingreso: P(x) = ";
                 cout<<endl;
                 for(int i=0; i<=grado; i++){
                       m=grado-i;
                       if(i!=grado){
                            cout << " " << a[m] << " x' " << m << " + ";
                        }
                       else{
                           cout << " " << a[m] << " ";
                         }
                  }
                cout << " Coloque el valor para evaluar el P(x): ";
                cin >> remp;
                int mul=0, sum=0;
                b[grado] = a[grado];
                cout<<b[grado]<<"\n";
                for(int k=(grado-1); k>=0; k--){
                    b[k]=remp*b[k+1];
```

```
        cout<<b[k]<<" 1 "<<endl;
        b[k]=b[k]+a[k];
        cout<<b[k]<<" 2 "<<endl;
         mul++;
         sum++;
    }

    cout << " Solucion:  " << b[0];
    cout << endl << endl;
    cout<< "el numero de sumas y restas es:" << sum<<endl;
    cout<< "el numero de productos es:" << mul<<endl;
    return 0;
}
```

Resultados:

1.

$$P(x) = 2x^4 - 3x^2 + 3x - 4 \qquad \text{en } x_0 = -2$$

```
Solucion:  10


el numero de sumas y restas es:4

el numero de productos es:4
```

2.

$$P(x) = 7x^5 + 6x^4 - 6x^3 + 3x - 4 \qquad \text{en } x_0 = 3$$

```
     Solucion:  2030


el numero de sumas y restas es:5
el numero de productos es:5
```

3.

$$P(x) = -5x^6 + 3x^4 + 2x^2 - 4x \qquad \text{en } x_0 = -1$$

```
Solucion:  4

el numero de sumas y restas es:6
el numero de productos es:6
```

Punto 2

b) es muy similar a una búsqueda binaria que tiene una complejidad de orden logarítmico (O(log n))

ya que en cada iteración la variable a se vera reducida en la mitad

Punto 3

3.Utilice el método de Newton para resolver el problema, muestre gráficamente cómo se comporta la convergencia a la solución

Ejemplo. Una partícula se mueve en el espacio con el vector de posición **R(t) = (2cos(t), sen(t), 0)**. Se requiere conocer el tiempo en el que el objeto se encuentra más cerca del punto **P(2, 1, 0).** Utilice el método de Newton con cuatro decimales de precisión.

Solución

Distancia entre dos puntos:

$$d(t) = \sqrt{(2\cos(t)-2)^2 + (\text{sen}(t)-1)^2 + (0-0)^2}$$

Modelo matemático: f(t) = d'(t) = 0

$$f(t) = d'(t) = \frac{2\cos(t)\,(\text{sen}(t)-1) - 4\text{sen}(t)(2\cos(t)-2)}{2\sqrt{(2\cos(t)-2)^2 + (\text{sen}(t)-1)^2}} = 0$$

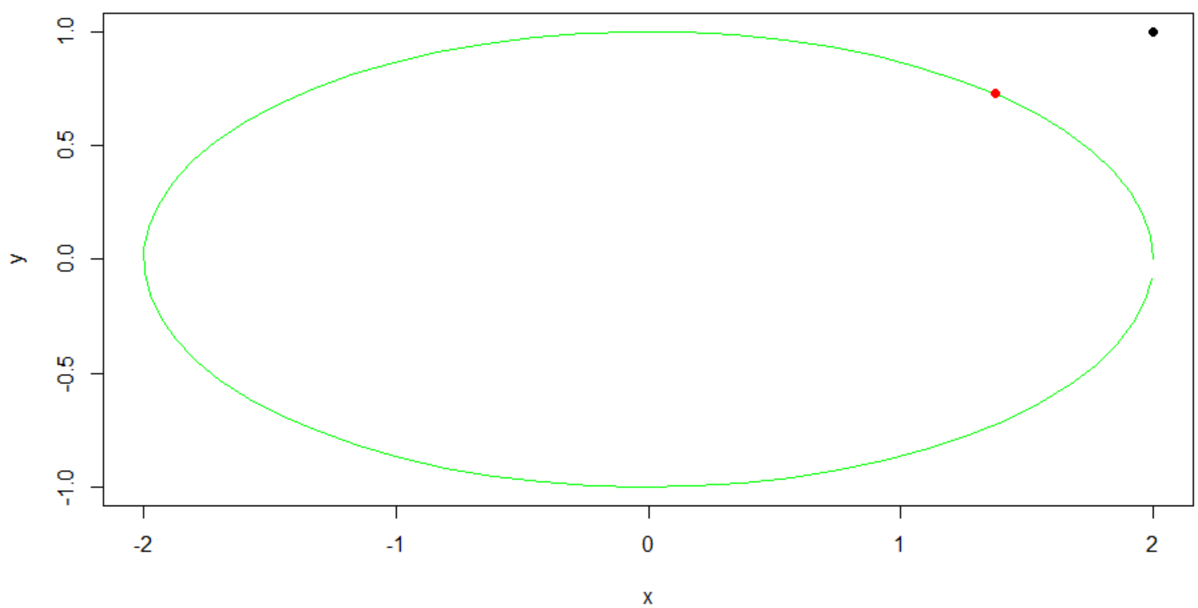
$$f(t) = \cos(t)\,(\text{sen}(t)-1) - 4\,\text{sen}(t)\,(\cos(t)-1) = 0$$
$$= 3\text{sen}(t)\cos(t) - 4\text{sen}(t) + \cos(t) = 0$$

$$f'(t) = 4\cos(t) + \text{sen}(t) - 3\cos(t)^2 + 3\text{sen}(t)^2$$

```
f<function(x){
  return(sqrt((2-(2*cospi(x)))**2+(1-sinpi(x))**2))
}
df<-function(x)
{
  return(((4*sinpi(x) )- (((3*sinpi(x)) +1)*cospi(x)))/(sqrt((sinpi(x) * sinpi(x))-(2*sinpi(x))+(4*(cospi(x) * cospi(x)))-(8*cospi(x))+5)))
}
dat<- data.frame(t=seq(0, 2*pi, by=0.1) )
xhrt <- function(t) 2*cos(t)
yhrt <- function(t) sin(t)
dat$y=yhrt(dat$t)
dat$x=xhrt(dat$t)
with(dat, plot(x,y, type="l",col="Green"))
newton<-function(f,df,E,xo)
{
  k=0
  repeat{
```

```
  c=f(xo)/df(xo)
  x1=xo-c
  dx=abs(c)
  xo=x1
  k=k+1
  if(dx<E ||k>100)
    break;
 }
 y1<-sin(x1)
 cat("# de iteraciones: ",k," El valor de xmax: ",format(x1,nsmall =
4),"rad, f(x): ",f(x1), ",Error Estimado: ", c)
 cat("\nEl punto es: (",format(2*cos(x1),nsmall = 4),",",format(y1,nsmall
= 4),", 0.0000)\n")
 points(x=2,y=1,pch=19)
 points(x=2*cos(x1),y=y1,pch=19,col="Red")
}
newton(f,df,0.0001,1)
```

Resultados:



de iteraciones: 101 El valor de xmax: -816.0000 rad, f(x): 1 ,Error Estimado: 17
El punto es: (1.373061 , 0.7271011 , 0.0000)

Punto 4

```
#Metodo de newton
rm (list = ls ())
f1 <- función (x) 2 + cos (3 * x)
f2 <- función (x) 2-exp (x)
```

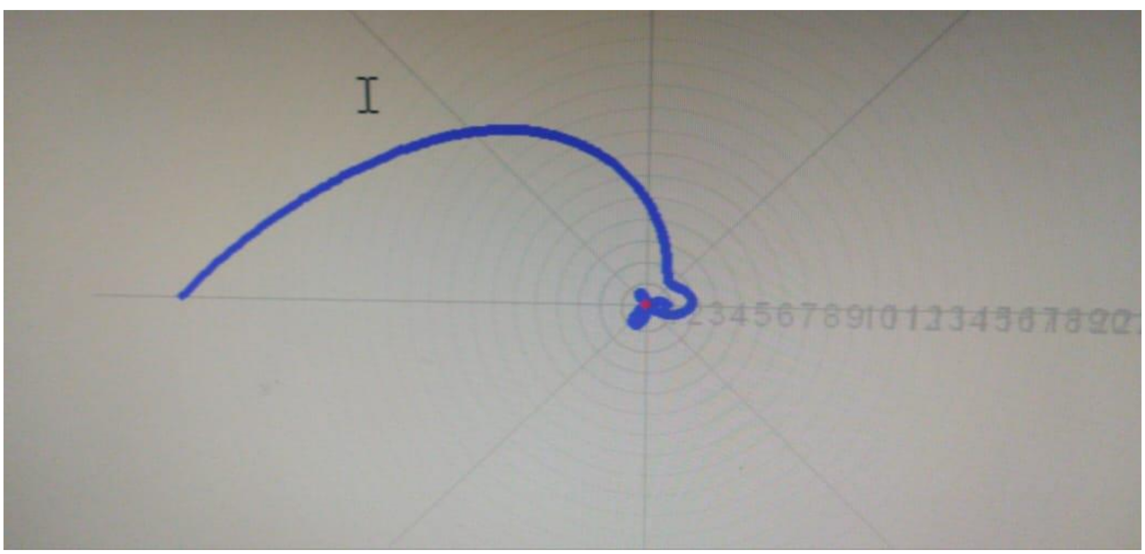
```
nw <-function(a,b,t){
  if(f1(a)*f2(b)<0){
    error<-1
    numeroanterior=t
    while(error>1.e-4){
      t<-t-f1(t0)/dfdt(t)
      error<-abs(t-numeroanteriro)/abs(t)
      numeroanterior<-t
    }
  }else{
    cat("El intervalo escogido no tiene una raiz única.")
  }
  return(t)
}
nw(0,pi/4)


#Metodo secante
f1<-función (x) 2 + cos (3 * x)
f2<-función (x) 2-exp (x)
Sc<-función (a, b)
{
  x <- (f1(b) * a-f1(a) * b) / (f1(b) -f1(a))
  error <-1
  mientras (error> 1.e-4)
  {
    a<-b
    b<-x
    x<- (f1(b) * a-f1(a) * b) / (f1(b) -f1(a))
    if(f1(x) == 0){
       break
    }
    error <-abs (f1(x) / f2(x))
    cat ("r =", x, "\ t", "E =", error, "\ n")
  }
}
secante (3 * pi / 2,pi)
```

Graficacion a polares

variabame r= nueva ecuacion que sale apartir de igual los dos iniciales
punto de oprximacion= -0.697329122

points(cir,fi,pch="-",col="gray",cex=0.3)
text(ejex+0.2,-0.2,ejex,col="gray")
ejex <- ejex + 1
aux <- aux - 1
}
abline(v=((max(cir)+min(cir))/2),col="gray")
abline(h=((max(cir)+min(cir))/2),col="gray")
points(x,y,pch=20,col=color,cex=1)
}
polar(dim,r,"blue")



Punto 13.
Formula iterativa para calcular la raíz real n-enésima de un número real.
y = yo + (x-xo)/n (xo^1/n)^(1-n)

Donde
yo:= la raiz nésima conocida mas cercana al valor buscado
xo:= el valor cuya enésima raiz es aquella yo
x:= el valor cuya raíz deseamos conocer
y := la raíz enésima que deseamos conocer de aquella x

Ejemplo: Deseamos sacar la raiz 5a de 40, si te das cuenta la raiz mas cercana es la de 32, por lo que la raiz 5a de 32 es 2, esto es:
n = 5
xo = 32
yo = 2
x=40
y=?
Sustituyendo

y = 2 + (40 - 32) /5[32^1/5]^(1-5)
= 2+ (8) /5 (2)^-4
=2+ 8 /(5*16)
= 2+1/10
= 2.1
Ahora si revisas en una calculadora verás que raiz 5a de 40 es
2.0912.... redondeando es 2.1

Punto 14
a) Para que exista una raíz tiene que haber una función f(x) continua en un intervalo [a,b] en el que f(a) y f(b) son de diferente signo, por consiguiente habrá por lo menos un punto en el que f(c)=0. Adicionalmente, para que sea única, usando el teorema de Rolle podemos suponer que si hay mas de una raíz tendríamos que f(c1)=f(c2), siendo c1 y c2 las dos raíces, en donde la función es continua y derivable entre ambos puntos, tenemos que tiene que haber un punto intermedio c en el que f'(c)=0, al realizar la derivada de la función original tendríamos que saldrán unos puntos en los que esta es cero, para demostrar que es única estos puntos tendrían que estar justo en los extremos o fuera de nuestro intervalo inicial.
b) orden de convergencia lineal
c)
```
def f (x):
   return (x**3+2*x**2-x+1)
# E es el error a evaluar
E=1E-8
```

```
print("ingrese a")
numero=int (input())
print ("ingrese b")
numero2=int (input())
d=numero2-numero
d=d/10
x=numero
bol=True
while bol:
   num=f(x)
   x=x+d
   if num<0 and f(x)>0 or num>0 and f(x)<0:
      x=x-d
      d=d/10
   if d<E:
      bol= False
      print ("la raiz es:" , x)
```

Parámetros:
Intervalo [a,b]
Resultados:
la raiz es: -2.5468182799999988

Punto 15
Punto b y d se encuentran en el código.

a) La ecuación que se obtiene al realizar la integral es la siguiente:

$f(x) = -e^{x} + 5x - 1 = 0$

c) Para obtener un g(x) se le adiciona una x ambos lados de la ecuación. $\boldsymbol{x=-e^{x}+5x-1+x}$

$\boldsymbol{x = -e^{x} + 6x - 1}$

Para encontrar el intervalo de convergencia se resolvió la siguiente desigualdad.

$\boldsymbol{-1<-e^{x}+6x-1<1}$ $\boldsymbol{0 < -e^{x} + 6x < 2}$