R Notebook

TALLER 1

1. Evaluar el valor de un polinomio es una tarea que involucra para la maquina realizar un número de operaciones la cual debe ser mínimas. Para cada uno de los siguientes polinomios

$$P(\dot{x}) = 2x^4 - 3x^2 + 3x - 4 \quad x = -2$$
$$P(\dot{x}) = 7x^5 + 6x^4 - 6x^3 + 3x - 4 \quad x = 3$$
$$P(\dot{x}) = -5x^6 + 3x^4 + 2x^2 - 4x \quad x = -1$$

```
#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 acompaC1a 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]<<"hdhdh\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;
}
```

2. Dado el siguiente algoritmo

$$Leer \quad n$$
$$Mientras n > 0 \quad repita$$
$$d \leftarrow mod(n, 2)$$
$$n \leftarrow fix(n, 2)$$
$$mostrar$$
$$fin$$

a. Recorra el algoritmo con n = 73

```r
t = function(maxiter = 100){
  while(maxiter > 0){
    d=maxiter%%2
    maxiter = maxiter%/%2
    cat(d, maxiter, "\n")
  }
}
t(73)
```

```
## 1 36
## 0 18
## 0 9
## 1 4
## 0 2
## 0 1
## 1 0
```

b. Encuentra T(n) y exprésela con la notación O(). Para obtener T(n) observe el hecho de que en cada ciclo el valor de N se reduce aproximadamente a la mitad

Para calcular la complejidad de este algoritmo, inicialmente en número de elementos por analizar es n. Tras la primera división, el número será como mucho n/2 (pues nos hemos quedado con la mitad de elementos); tras la segunda división, el número será como mucho n/4; y así sucesivamente. Por lo general, tras la división número i, el número de elementos por analizar será como mucho:

$$T(n) = \begin{cases} \sigma(1) \\ T\frac{n}{2} + \sigma(n) \end{cases}$$

De acuerdo al teorema maestro podemos afirmar ya que

$$a \geq 1, b > 1, k \geq 0$$

f(n) sea una funcion del tipo:

$$T(n) = a * T\frac{n}{2} + f(n)$$

entonces:
$$T(n) = a \times T\frac{n}{2} + f(n) \to T(n) \in \theta(n^2) \to \log_2(n)$$

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

Una particula se mueve en el espacion con el vector de posicion R(t)=(2 cos(t), sen(t), 0). Se requiere conocer el tiempo en el que se encuentra mas cerca del punto P(2,1,0). Utilice el meto de newton con cuatro decimales

```
punto3 = function(f, x0, tol, maxiter){
  fp = function(x) { h = 1e-15
  (f(x+h) - f(x-h)) / (2*h)
  }
  k = 0
  cat(formatC( c("x_k"," f(x_k)","Error est."), width = -20, format = "f", flag = " "), "\n")
  repeat{
    nume = f(x0)/fp(x0)
    dx = abs(nume)
    x1 = x0 - nume
    cat(formatC( c(x1 ,f(x1), dx), digits=15, width = -15, format = "f", flag = " "), "\n")
    x0 = x1
    k = k+1
    if(dx <= tol || k > maxiter ) break;
  }
  cat("--------------------------------------------------------------------------\n")
  if(k > maxiter){
    cat("Se alcanzó el máximo número de iteraciones.\n")
    cat("\nError estimado <= ", nume)
    cat("\nTiempo estimado.",x1,"\n")
  } else {
    cat("\nError estimado <= ", nume)
    cat("\nTiempo estimado.",x1,"\n")
    }
}

f = function(x) x-cos(x)
punto3(f, 2, 1, 0)
```

```
## x_k                     f(x_k)                Error est.
##  0.639830250906736 -0.162366868805647  1.360169749093264
## --------------------------------------------------------------------------
## Se alcanzó el máximo número de iteraciones.
##
## Error estimado <=  1.36017
## Tiempo estimado. 0.6398303
```

4. Solución en R. Para la siguiente ecuación, utilice dos métodos diferentes, grafique las soluciones y comparar
$$\begin{cases} r = 2 + cos(3 \times t) \\ r = 2 - e^t \end{cases}$$

5. El número épsilon de una maquina el cual se denota como e, es la distancia entre el 1 y el menor número de punto flotante mayor que 1. Para el punto flotante de precision doble se tiene que:
$$\varepsilon_{maq} = 2^{-52}$$

Teniendo en cuenta lo anterior, encuentre el error de redondeo para x = 0.4

```
t = function(maxiter = 100){
  n = 0;
  while(1.0+(maxiter*0.5) > 1.0){
    n=n+1
    maxiter = maxiter*0.5
  }
  cat("Epsilon de la maquina en forma hexadecimal = ", maxiter, "\n")
  cat("Epsilon de la maquina en forma binaria = 2^-", n, "\n")
  suma = 1
  i = 1
  while(i<=1000){
    suma = suma + 0.40000000000
    i = i + 1
  }
  maxiter = 10000 * 0.40000000000 + 1

  cat("El error acumulado es  = ", suma, "\n")
}
t(1.0)
```

```
## Epsilon de la maquina en forma hexadecimal =  2.220446e-16
## Epsilon de la maquina en forma binaria = 2^- 52
## El error acumulado es  =  401
```

6. Resolver los ejercicios 13, 14

a. Encuentre una formula iterativa de convergencia cuadratica y defina un intervalo de convergencia apropiado para calcular la raíz n-ésima de un numero real. El algoritmo solamente debe incluir operaciones aritmeticas elementales

```
punto6a = function(num,x1){
    xk = 1
    xk1 = 1
    a= 1
    while (a!=0){
        xk = xk1
        xk1 = 0.5 * (xk+(num/(xk * (x1-1))))
        if (abs(xk-xk1) < 0.00000001){
            a=0
        }
    }

    cat("La raiz es = ", xk1 )
}

punto6a(27, 3)
```

```
## La raiz es =  3.674235
```

b. El siguiente es un proceso intuitivo para calculr la raiz real positiva de la ecuacion f(x) = 0 en un intervalo [a,b] con precision E

b.1.De manera formal escriba las condiciones necesarias para que la raiz exista, sea unica y pueda ser calculada

$$En\ matemática,\ se\ conoce\ como\ raíz\ (o\ cero)\ de\ un\ polinomio\ o\ de\ una\ función(definida\ sobre\ un\ cierto\ cuerpo\ algebraico)$$

b.2.Indique el orden de convergencia y estime el factor de convergencia

b.3.Describa el procedimiento anterior en notación algoritmica

```
punto6b = function(g,a,b, E=1e-9, maxIt=50){
  k = 1
  x = a
  anterior = 0
  d =  ((b-a)/10)
  repeat{
    x1 = g(x)
    simbolo = x1 * anterior
    anterior = x1
    if (simbolo > 0 && k > 1 ){
      x = x+d
    }
    if (simbolo < 0 && k > 1){
      x = x-d
      d = d/10
    }
    if ((d < E && k > 1) || k > maxIt || (x1==0 && k>1)){
      cat("Iteracion", k, "  x= ", x1, "\n")
      break;
    }

    cat("Iteracion", k, "  x= ", x1, "\n")
    k = k+1
  }
  if( k == maxIt+1 ){
    cat("No hubo convergencia ")
  } else{
    cat("x es aproximadamente ", x, " con error menor a ", E)
  }
}
```