

Económicas, UBA. Actuario. Análisis Numérico.

Cuatrimestre 2, 2021. Primer Examen Parcial.

Paluch Uriel - 895700

07/Octubre/2021

Contents

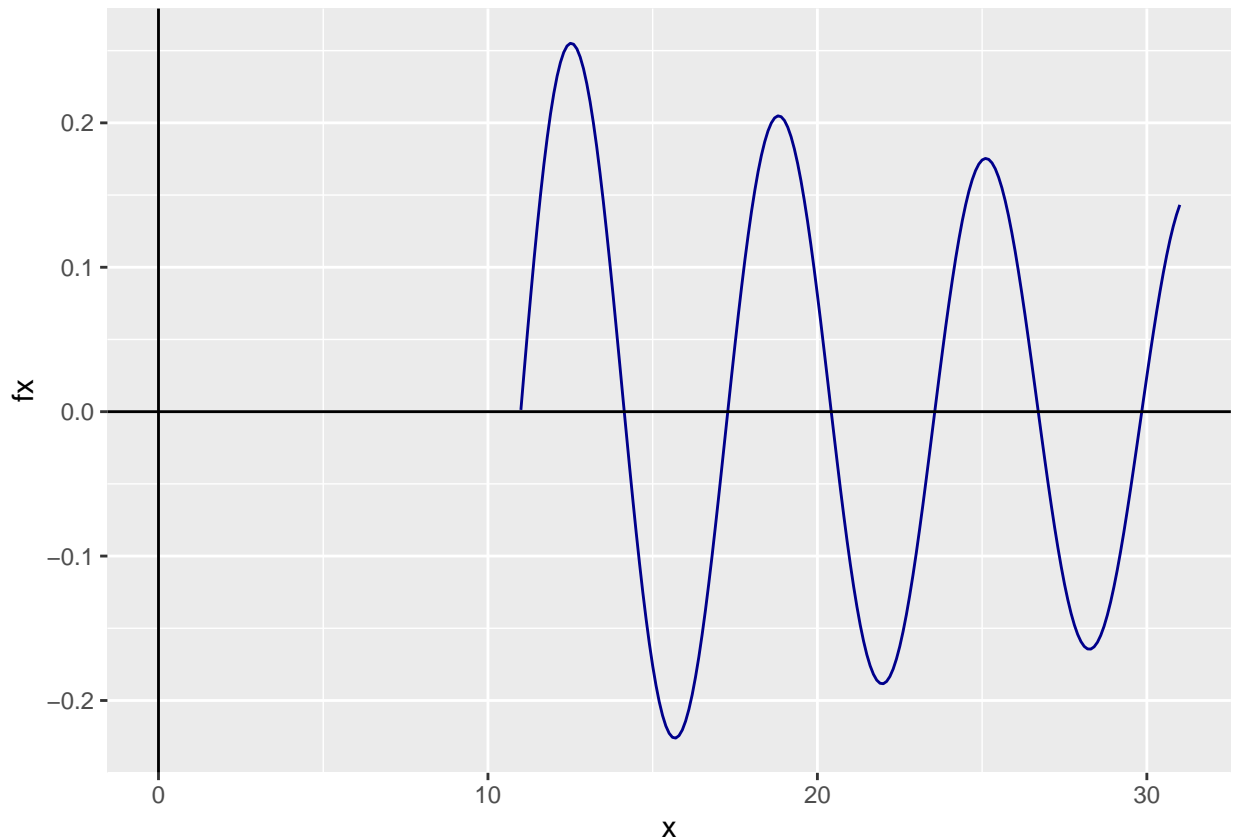
1 Resolución de Ecuaciones: Secante. (24 puntos)	2
1.1 Graficar función e identificar raíces	2
1.2 Corregir algoritmo	2
1.3 Hallar raíces	3
1.4 Iteraciones	3
2 Resolución de Ecuaciones: Newton-Raphson. (24 puntos)	5
2.1 Teoría	5
2.2 Hallar raíces	5
2.3 Iteraciones	5
2.4 Graficar función y marcar raíces	5
3 Factorización de Matrices. (12 puntos)	7
3.1 Factorización de Cholesky	7
3.2 Factorización LU	7
4 Sistemas de Ecuaciones No Lineales. (10 puntos)	8
5 SQL. (20 puntos)	9
5.1 Consulta Empleados	9
5.2 Consulta Proveedores	9
6 Aprendizaje automático. (10 puntos)	10
## Warning: package 'flextable' was built under R version 4.1.1	
## Warning: package 'tidyverse' was built under R version 4.1.1	
## Warning: package 'tibble' was built under R version 4.1.1	
## Warning: package 'tidyr' was built under R version 4.1.1	
## Warning: package 'readr' was built under R version 4.1.1	
## Warning: package 'purrr' was built under R version 4.1.1	
## Warning: package 'dplyr' was built under R version 4.1.1	
## Warning: package 'stringr' was built under R version 4.1.1	
## Warning: package 'forcats' was built under R version 4.1.1	

1 Resolución de Ecuaciones: Secante. (24 puntos)

Para este ejercicio, considere la función $h(x) = \cos(x)/x^{0.54}$.

1.1 Graficar función e identificar raíces

Grafique la función $h(x)$ en el intervalo $[11, 31]$ de manera tal que pueda identificar todas las soluciones de la ecuación en el intervalo.



Respuesta:

1.2 Corregir algoritmo

CORREGIR el algoritmo “Secante” en el siguiente bloque de código.

COMENTAR los cambios que realizó (use “#” al final de cada línea modificada).

[Notar que, para el resto del ejercicio, no debe utilizar su propio algoritmo, sino que debe usar el algoritmo dado y corregido, sin agregar ninguna línea de código adicional.]

Respuesta:

```
# Edite las líneas que considere erróneas
# Comente al final de cada línea editada
Secante <- function(f,p0,p1,TOL,N){
  i <- 2
  q0 <- f(p0)
  q1 <- f(p1)
  while (i <= N){ #Mientras que i sea menor al numero máximo de iteraciones
    p = p1 - q1*(p1-p0)/(q1-q0) #Corregi la formula
    if (abs(p-p1) < TOL){ #La tolerancia se calcula en base a p, no q
```

```

    return(p)
  }
  i <- i + 1
  p0 = p1 #p0 = p1
  q0 = q1
  p1 = p #p1 = p
  q1 = f(p) #f(p)
}
return(paste('El metodo fallo luego de ', N, ' iteraciones')) #R es case sensitive
}

```

1.3 Hallar raíces

Utilizando el algoritmo del punto 1.2, halle todas las raíces identificadas en el punto 1.1.

Respuesta:

```

## [1] 14.13717
## [1] 17.27876
## [1] 20.42035
## [1] 26.70354
## [1] 29.84513

```

1.4 Iteraciones

Tome el algoritmo del punto 1.2 (copie y pegue) y agregue las líneas de código que considere necesarias para poder visualizar (imprimir) cada iteración del algoritmo. Una vez editado el algoritmo, imprima 8 iteraciones del algoritmo iniciando en $x_0 = 16$ y $x_1 = 19$. ¿A cuál de las raíces convergería el algoritmo en este caso?

Respuesta:

```

## [1] "En la iteracion: 3"
## [1] "p vale: 17.5456577303667"
## [1] "p0 vale: 19"
## [1] "q0 vale: 0.201622789126561"
## [1] "p1 vale: 17.5456577303667"
## [1] "q1 vale: 0.0561468628097437"
## [1] "En la iteracion: 4"
## [1] "p vale: 16.9843500421538"
## [1] "p0 vale: 17.5456577303667"
## [1] "q0 vale: 0.0561468628097437"
## [1] "p1 vale: 16.9843500421538"
## [1] "q1 vale: -0.0628685374988462"
## [1] "En la iteracion: 5"
## [1] "p vale: 17.2808544722805"
## [1] "p0 vale: 16.9843500421538"
## [1] "q0 vale: -0.0628685374988462"
## [1] "p1 vale: 17.2808544722805"
## [1] "q1 vale: 0.000449648456801175"
## [1] "En la iteracion: 6"
## [1] "p vale: 17.2787488724844"
## [1] "p0 vale: 17.2808544722805"
## [1] "q0 vale: 0.000449648456801175"
## [1] "p1 vale: 17.2787488724844"

```

```
## [1] "q1 vale: -2.30159908064475e-06"  
## [1] "Converge a la raiz ubicada en: 17.2787595954536"
```

2 Resolución de Ecuaciones: Newton-Raphson. (24 puntos)

Considere la siguiente ecuación: $4\sin(x) = e^{46/x^3}$.

2.1 Teoría

Describa el método de *Newton-Raphson* que se utiliza para hallar raíces de funciones.

Respuesta (escriba a continuación):

El método de la forma presentada en el libro esta basada en los polinomios de Taylor.

Supongamos que p_0 existe en un intervalo $[a; b]$ y es una aproximación para p , de tal forma que $f'(p_0) \neq 0$ y $|p - p_0|$ es “pequeño”. Consideremos el primer polinomio de Taylor expandido de p_0 y evaluado en $x = p$:

$$f(p) = f(p_0) + (p - p_0) * f'(p_0) + \frac{(p - p_0)^2}{2} * f''(\xi(p))$$

Puesto que, $f(p) = 0$ y $(p - p_0)^2$ es pequeño, reexpresamos de modo que:

$$p = p_0 - \frac{f(p_0)}{f'(p_0)}$$

Los consecutivos puntos p_n se encuentran mediante el cruce de la recta tangente propuesta con el eje de las X.

2.2 Hallar raíces

Plantee la ecuación de la forma $f(x) = 0$ y halle todas las raíces en el intervalo $[40, 50]$ utilizando el método de Newton-Raphson.

Respuesta:

[1] 40.58785

[1] 44.23511

[1] 46.87109

2.3 Iteraciones

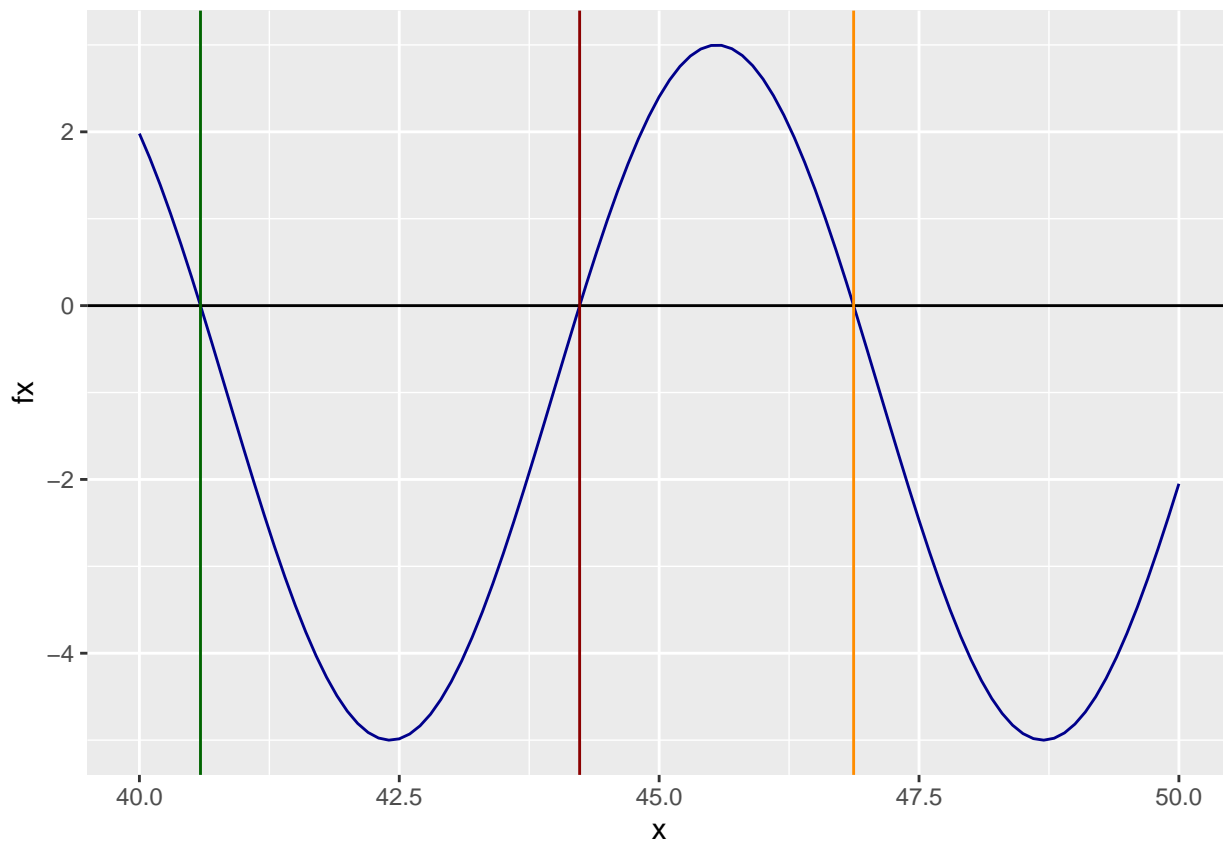
Considere el punto anterior. Realice 6 iteraciones del algoritmo de Newton-Raphson, utilizando en $x_0 = 41$. ¿A cuál de las raíces convergería el algoritmo en este caso?

Respuesta:

[1] 40.58785

2.4 Graficar función y marcar raíces

Grafique la función $f(x)$ en el intervalo $[40, 50]$, identifique todas las raíces halladas en el punto 2.2 y marque cada una con un color distinto en el gráfico.



Respuesta:

3 Factorización de Matrices. (12 puntos)

3.1 Factorización de Cholesky

Realice la factorización de Cholesky de la siguiente matriz:

$$A = \begin{bmatrix} 1 & 0.0481 & 0.0105 \\ 0.0481 & 1 & 0.0116 \\ 0.0105 & 0.0116 & 1 \end{bmatrix}$$

Respuesta:

```
##          [,1]          [,2]          [,3]
## [1,] 1.0000 0.00000000 0.0000000
## [2,] 0.0481 0.99884253 0.0000000
## [3,] 0.0105 0.01110781 0.9998832
```

3.2 Factorización LU

Realice la factorización de LU de la siguiente matriz:

$$B = \begin{bmatrix} 10 & 8 & 6 & 8 \\ 6 & 5 & 19 & 8 \\ 9 & 11 & 5 & 16 \\ 9 & -7 & 6 & 9 \end{bmatrix}$$

Respuesta:

```
## $L
##          [,1] [,2]          [,3] [,4]
## [1,] 1.0    0 0.000000 0
## [2,] 0.6    1 0.000000 0
## [3,] 0.9   19 1.000000 0
## [4,] 0.9  -71 -3.733788 1
##
## $U
##          [,1] [,2]          [,3]          [,4]
## [1,] 10 8.0    6.0    8.000
## [2,] 0 0.2   15.4    3.200
## [3,] 0 0.0 -293.0 -52.000
## [4,] 0 0.0    0.0   34.843
```

4 Sistemas de Ecuaciones No Lineales. (10 puntos)

Utilice el método de Newton para resolver el siguiente Sistema de Ecuaciones No Lineales, tomando como punto inicial $(1, 1, -1)$.

$$3x - \cos(yz) = 0.5$$

$$9x^2 - 93(y + 1/10)^2 + \sin(z) = -0.52$$

$$e^{-xy} + 21z = 4 - (14/3)\pi$$

Aproxime la solución con una tolerancia de 10^{-6}

Respuesta:

```
##          [,1]
## [1,]  0.4998437
## [2,]  0.0552840
## [3,] -0.5539767
```


5 SQL. (20 puntos)

Utilice la base de datos de la siguiente web para preparar sus códigos de SQL: https://www.w3schools.com/sql/trysql.asp?filename=trysql_select_all

5.1 Consulta Empleados

Escriba una consulta SQL que contenga los empleados con más de 24 ventas realizadas o cuyo total vendido es mayor a \$13500. La salida debe contener los siguientes campos: Nombre y Apellido del Empleado, Cantidad de ventas, Total (calculado como la suma total del Precio*Cantidad).

Respuesta (escriba a continuación):

```
SELECT
E.FirstName AS "Nombre del empleado",
E.LastName AS "Apellido del empleado",
COUNT(DISTINCT(O.OrderID)) AS "Cantidad de ventas",
E.EmployeeID,
ROUND(SUM(P.Price * OD.Quantity),2) AS "Total"
FROM Employees E
INNER JOIN Orders O ON O.EmployeeID = E.EmployeeID
INNER JOIN OrderDetails OD ON OD.OrderID = O.OrderID
INNER JOIN Products P ON P.ProductID = OD.ProductID
GROUP BY O.EmployeeID
HAVING COUNT(DISTINCT(O.OrderID)) > 24 OR SUM(P.Price * OD.Quantity) > 13500
```

5.2 Consulta Proveedores

Escriba una consulta SQL que contenga un resumen de los productos vendidos de los proveedores de Italia. La salida debe contener los siguientes campos: Nombre del Proveedor, ciudad y país del proveedor, Contacto del proveedor, Cantidad de ventas y Total (calculado como la suma total del Precio*Cantidad) y tiene que estar ordenada de forma descendente por la cantidad.

Respuesta (escriba a continuación):

```
SELECT
S.SupplierName AS "Nombre del proveedor",
S.City AS "Ciudad del proveedor",
S.Country AS "Pais del proveedor",
S.ContactName AS "Contacto del proveedor",
COUNT(DISTINCT(OD.OrderID)) AS "Cantidad de Ventas",
ROUND(SUM(P.Price * OD.Quantity),2) AS "Total" FROM Suppliers S
INNER JOIN Products P ON P.SupplierID = S.SupplierID
INNER JOIN OrderDetails OD ON OD.ProductID = P.ProductID
GROUP BY S.SupplierID
HAVING S.Country = "Italy"
ORDER BY COUNT(DISTINCT(OD.OrderID)) DESC
```

6 Aprendizaje automático. (10 puntos)

Defina el concepto de “Big Data”, identifique los impactos que tiene en la actualidad y relaciónelo con el aprendizaje automático.

Respuesta (escriba a continuación):

Se llama Big Data a la masiva cantidad de datos generados por los dispositivos electrónicos de todo tipo. Puede ir desde un dispositivos GPS hasta interacciones en las redes sociales. Se pueden recolectar de muchas formas, con el ejercicio de twitter hicimos text mining pero, tambien podríamos haber hecho web scrapping para recolectar los datos directamente de la web.

La Big Data tiene las denominadas tres v:

Volumen: refiere la cantidad

Velocidad: al día de hoy, la internet permite tener los datos en tiempo real.

Variedad: Los datos pueden ser de todo tipo, imagenes, audios, texto localizaciones.

Y una versión extendida, las 5 v:

Veracidad: son los datos reales?

Valor: los datos por si solos no tienen ningún valor. Sirven por su capacidad predictiva. Surge la necesidad de analizar este gran volumen de datos, con algoritmos acordes (Ejemplo: Machine Learning, se basa en la detección de patrones sin la intervención humana).

La Big Data impacta a diario en nuestra vida, en el ambito tecnológico surgieron nuevas infraestructuras para adaptarse a los grandes volúmenes (Ejemplo: cloud storage). También tiene un gran impacto social. ¿Cómo se utilizan estos datos? ¿Qué saben realmente de nosotros? Solo basta con recordar el caso de Cambridge Analytics donde utilizaron la data de millones de usuarios de redes sociales para influir en el resultado electoral.

Los algoritmos aprenden a través de 3 métodos:

- Supervisado: cada uno de los datos tiene una “etiqueta” o respuesta asociada.
- No supervisado: los datos no tienen “etiqueta”.
- Aprendizaje reforzado: se da una recompensa al algoritmo cuando hace algo bien. Un ejemplo son las AI que juegan ajedrez.