Universidad Autónoma de Nuevo León

Facultad de Ciencias Físico Matemáticas

Actividad 3

Autores:

Diego Alejandro Téllez Martínez	1941559
Alexia Sofía Ibarra García	1841890
Cristian Joel Gallegos Yáñez	1990064
Gustavo Hernández Ángeles	1989978

Materia:
Física Computacional

Profesor:
Alfredo Tlahuice Flores *30 de enero de 2023*

Programa en fortran para método de Newton

```
program Newtom
implicit none
integer :: i, imax
real ::  x , xnuevo, toler
parameter (imax = 100)  ! Definimos las iteraciones maximas que debe correr el programa
parameter (toler = 0.0000001)  ! Tolerancia de error
print*, 'Valor inicial ' ; read*, x
! Para empezar con la interacion y pasos
xnuevo = 0.5 ; i = 0
    print*, '------------------------------------------------------------'
    print*, '| Pasos |     Aproximacion Raiz    |         f(x)        | '
! Metodo Newton (Si |x nuevo| > x )
do while ((abs(xnuevo-x)>toler) .and. (i<imax))
  i = i + 1                          ! Contador
  x = xnuevo                         ! Cambio de x
  xnuevo = x - f(x)/df(x)            ! Idea de Newton (y Raphson?)
  write(*,'(2x,i4,9x,f17.13,2x,f17.8)') i, xnuevo, f(xnuevo)
end do
! Si llega a las iteraciones maximas, marcar error (Depende tolerancia)
if (i>imax-2) then
    write(*,*) '# Error: Este metodo no converge, podrias reducir &
              el valor de la tolerancia'
    else
    write(*,'(a14,f20.17)') 'Tolerancia : ', toler
    print*, '------------------------------------------------------------'
    write(*,'(a8,f30.25)') 'Raiz : ', xnuevo
    print*, '------------------------------------------------------------'
endif
contains
! Funcion a evaluar
function f(z)
real ::  z, f
! f = z*z*z - 8
! f = -0.8*z*z*z*z + 6.6*z*z*z - 16*z*z + 11.7*z + 10
! f = z*exp(0.5*z) + 1.2*z - 5
! f = cos(z)*cosh(z) + 1
f = z - 2*exp(-z)
end function f
! Derivada de la funcion
function df(z)
real z, df
! df = 3*z*z
! df =- 3.2*z*z*z + 19.8*z*z - 32*z + 11.7
! df = exp(0.5*z)*(0.5*z + 1) + 1.2
! df = sinh(z)*cos(z) - sin(z)*cosh(z)
df = 1 + 2*exp(-z)
end function df
end program Newtom
```

1. Es posible por el método de Newton estimar el cero de la función $y = \sqrt{x^2 - 9}$

1.0.1. Calculo a mano

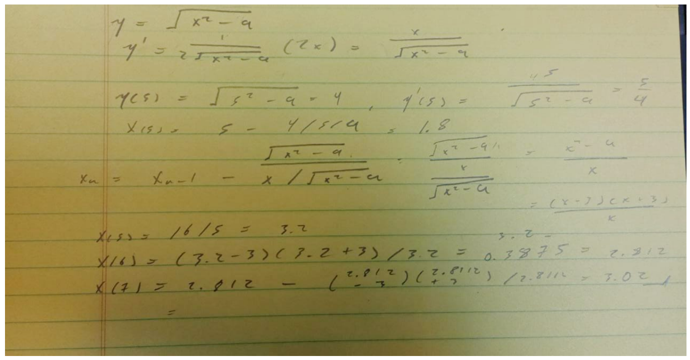


Figura 1: Solución a manual de $y = x^3 - 8$

1.0.2. Gráfica de la función

```
set xlabel 'x'
set ylabel 'f(x)'
set title "Grafica de f(x)"
unset border
set zeroaxis lt -1
set grid xtics mytics
set grid ytics mytics
set mytics 5
plot [-7:7][-5:5] sqrt(x*x - 9) w lp pt 3
replot 0 lw 1 linecolor black
```

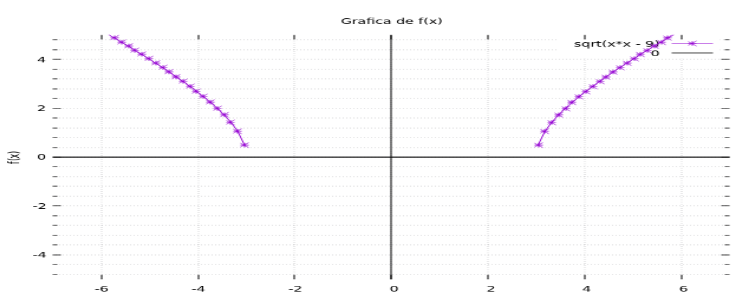


Figura 2: Gráfica de $y = \sqrt{x^2 - 9}$

Analíticamente, podemos concluir que la solución es 3, la gráfica nos ayuda a confirmar dicho resultado, vamos a calcularlo de manera numérica con el método de Newton

1.0.3. Solución numérica

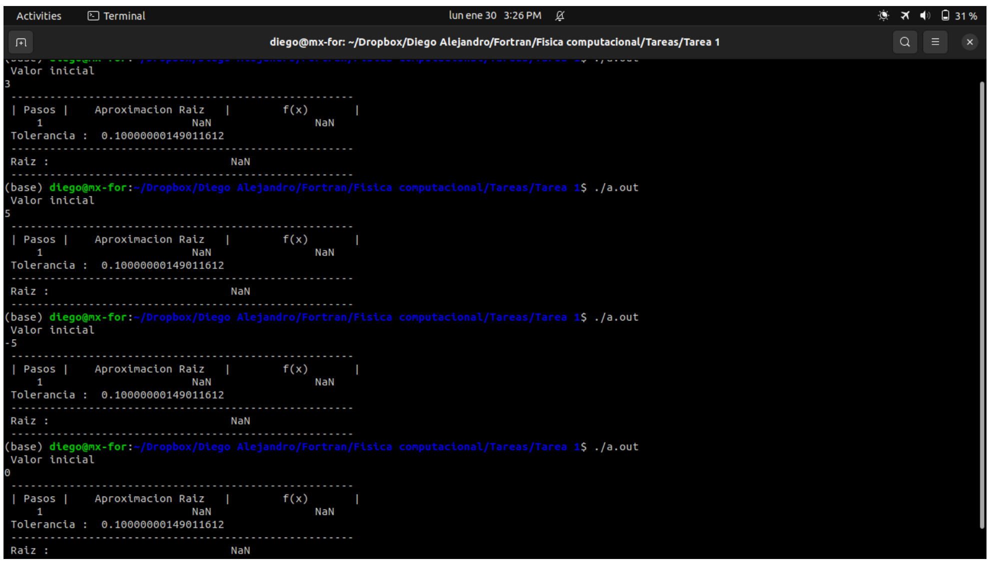


Figura 3: Solución numérica de$y = \sqrt{x^2 - 9}$

Corrimos el programa para diferentes valores iniciales, intentamos con x0 = 3, 5, -5 y 0. Para todos la solución nos daba un valor indeterminado, esto se puede deber a que la pendiente de la tangente puede aproximarse en gran magnitud a cero, entonces el método puede fallar o presentar lenta convergencia.

El método de Newton en gran parte se debe al valor inicial que uno ingresa y es decisivo poner atención en los máximos, mínimos y los puntos de inflexión, porque es cuando el método falla, también en darle una tolerancia de error muy pequeña, lo cual haría que el método use todas sus iteraciones y llegue a no converger a ninguna solución.

2. Encontrar las raíces de las siguientes funciones

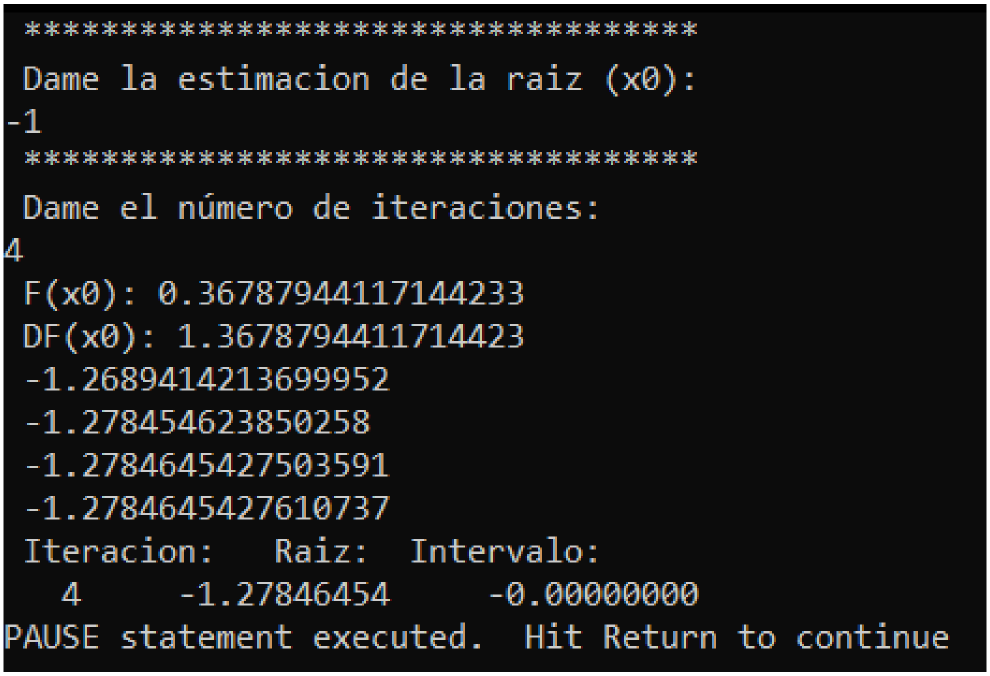
2.1. a) $e^x + x + 1 = 0$

Para resolver esta ecuación se nos pide iterar 4 veces, por lo cual previo a realizarlo verificamos la grágica mediante Gnuplot de forma que observemos el comportamiento de la ecuación así como la localización de las soluciones:

```
 **********************************
 Dame la estimacion de la raiz (x0):
-1
 **********************************
 Dame el número de iteraciones:
4
 F(x0): 0.36787944117144233
 DF(x0): 1.3678794411714423
 -1.2689414213699952
 -1.278454623850258
 -1.2784645427503591
 -1.2784645427610737
 -1.2784645427610737
 Iteracion:   Raiz:  Intervalo:
   5     -1.27846454      0.00000000
PAUSE statement executed.  Hit Return to continue
```

Figura 4: Gráfica de la función $e^x + x + 1 = 0$

Donde observamos una solución entre el intervalo de [-2,-1] por lo cual podremos emplear como valor inicial $x_0 = -1$. Sin embargo primero modificamos el código dado agregando un iteradr y modificando el ciclo para lo cual generalizamos un poco preguntando al usuario el número de iteraciones deseadas obteniendo el siguiente resultado:



```
***********************************
Dame la estimacion de la raiz (x0):
-1
***********************************
Dame el número de iteraciones:
4
F(x0): 0.36787944117144233
DF(x0): 1.3678794411714423
-1.2689414213699952
-1.278454623850258
-1.2784645427503591
-1.2784645427610737
Iteracion:   Raiz:  Intervalo:
  4     -1.27846454     -0.00000000
PAUSE statement executed.  Hit Return to continue
```

Figura 5: Solución de la función $e^x + x + 1 = 0$

Siendo la solución buscada dadas 4 iteraciones.

2.2. b) $x^2 - 16 = 0$

Para encontrar las raíces se nos pide tomar $x_0 = 3$ y x_0, sin embargo observemos la gráfica de la función obtenida mediante Gnuplot:

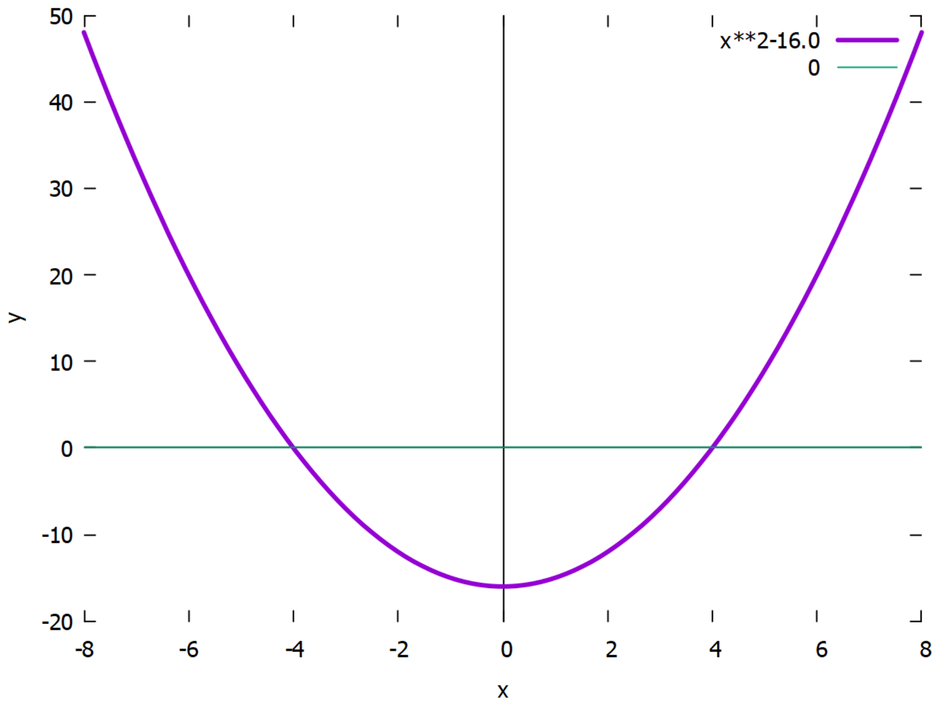


Figura 6: Gráfica de la función $x^2 - 16 = 0$

Donde observemos que se observan dos raíces en ± 4 por lo cual no podemos emplear $x_0 = 4$ sin embargo al utilizar ± 3 obtenemos las siguientes soluciones siendo las esperadas:

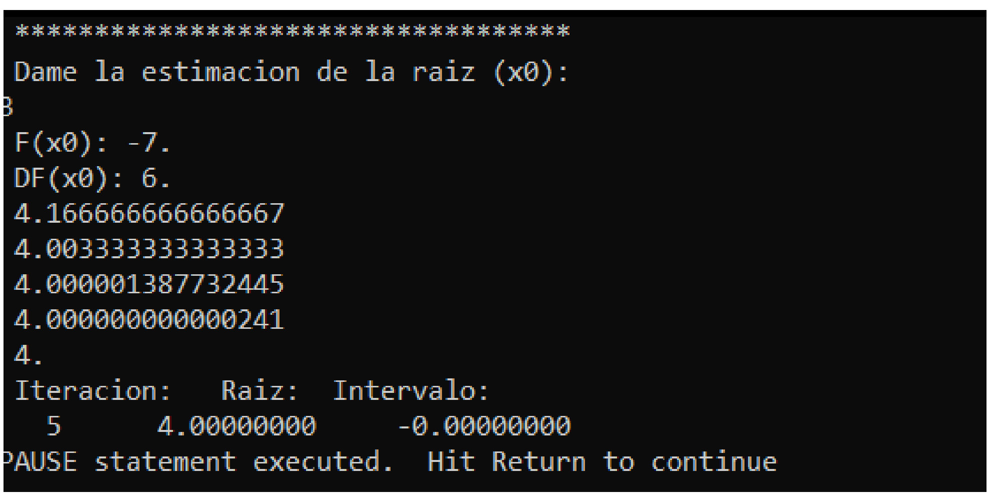


Figura 7: Solución 1 de la función $x^2 - 16 = 0$

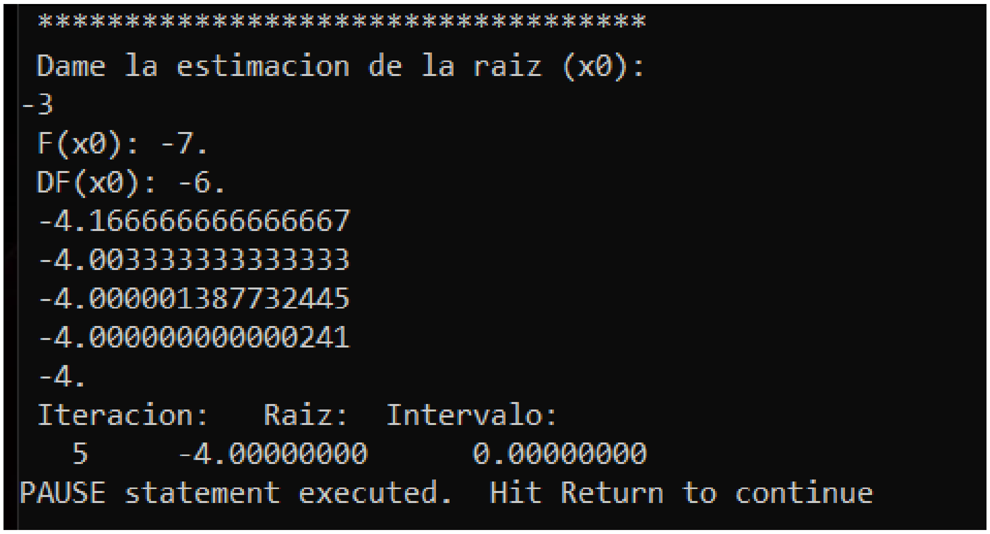


Figura 8: Solución 2 de la función $x^2 - 16 = 0$

Obteniendo en ambos casos las soluciones ± 4 después de 5 iteraciones.

2.3. c) $x^4 - x^2 = 0$

Para este polinomio de orden 4 se pide que se busque la raíz en el punto $x_0 = -0.5\sqrt{2.0}$. Se ingresa la función y su derivada $4x^3 - 2x$. Se obtiene que se realizan 29 iteraciones, cuando lo común es que este método converja rápidamente. Después de las 29 iteraciones el programa logra encontrar la raíz más cercana:

```
 Dame la estimacion de la raiz (x0):
-0.707
 F(x0): -0.249999977199
 DF(x0): 0.00042702800000015274
 584.7346506620422
 438.5512017690226
 328.9136863569401
 246.68564480837503
 185.0147403282154
 138.76173087787427
 104.07219900653749
 78.05535039960768
 58.543114358836355
 43.90947125918411
 32.93495094906384
 24.705010322223202
 18.533821592468534
 13.907120452439408
 10.439351837365175
 7.841542991006134
 5.897228667172214
 4.444427087323743
 3.3621758364094827
 2.5605307272334827
 1.973246386904809
 1.552615257995146
 1.2660397283270532
 1.0930253032264816
 1.0164399354800915
 1.0006399168926954
 1.00000102151157
 1.0000000000026088
 1.
 Iteracion:   Raiz:  Intervalo:
  29      1.00000000     -0.00000000
PAUSE statement executed.  Hit Return to continue
```

Figura 9: Solución numérica de$x^4 - x^2 = 0$

Esta raíz es 1.0000 y el programa logra calcularla exactamente.

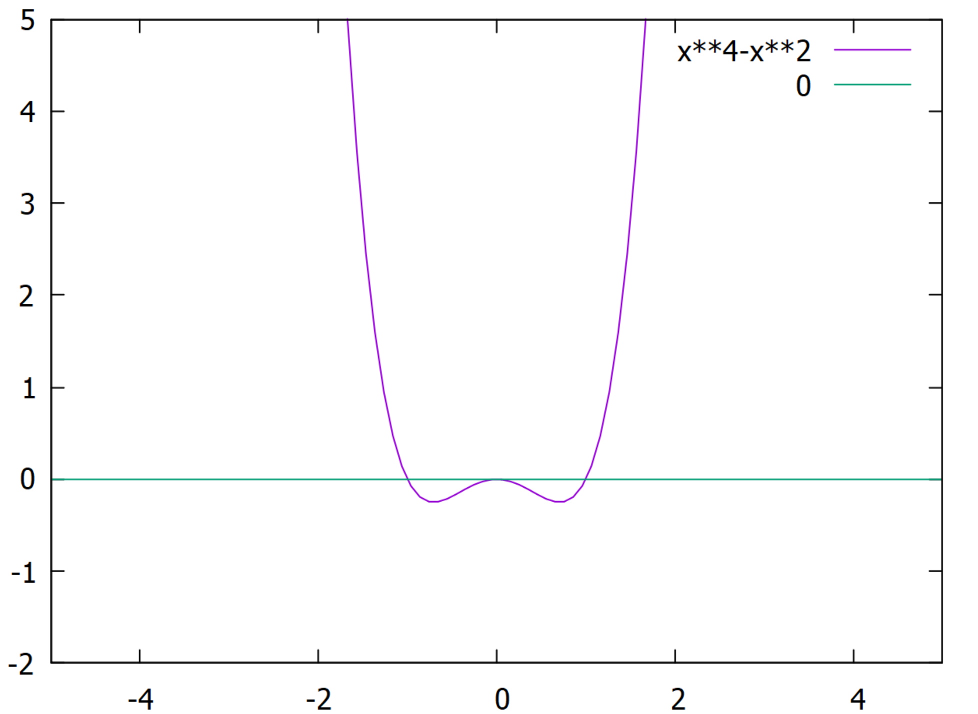


Figura 10: Gráfica de$x^4 - x^2 = 0$

Lo que sucede en este caso es que el programa tiene problemas para converger en $x_0 = -0.5\sqrt{2.0}$ específicamente para esta función.
Verificando con la solución analítica:

c) $f(x) = x^4 - x^2 \quad ; \quad f'(x) = 4x^3 - 2x$

$x^4 - x^2 = 0$

La ecuación para el método de Newton:

$$X_{n+1} = X_n + \frac{f(x)}{f'(x)}$$

Tomamos como $X_0 = -0.5\sqrt{2.0}$ que numéricamente es:

$$X_0 = -\frac{\sqrt{2}}{2} = -0.707$$

$$X_{0+1} = -0.707 - \frac{(-0.707)^4 - (-0.707)^2}{4(-0.707)^3 - 2(-0.707)}$$

$$X_1 = -584.73$$

$$X_2 = -584.73 - \frac{(-584.73)^4 - (-584.73)^2}{4(-584.73)^3 - 2(-584.73)}$$

$$X_2 = -438.54$$

$$X_3 = -438.54 - \frac{(-438.54)^4 - (-438.54)^2}{4(-438.54)^3 - 2(-438.54)}$$

$$X_3 = -328.90$$

$$X_4 = -328.90 - \frac{(-328.90)^4 - (-328.90)^2}{4(-328.90)^3 - 2(-328.90)}$$

$$X_4 = -246.67$$

$$X_5 = -246.67 - \frac{(-246.67)^4 - (-328.90)^2}{4(-246.67)^3 - 2(-246.67)}$$

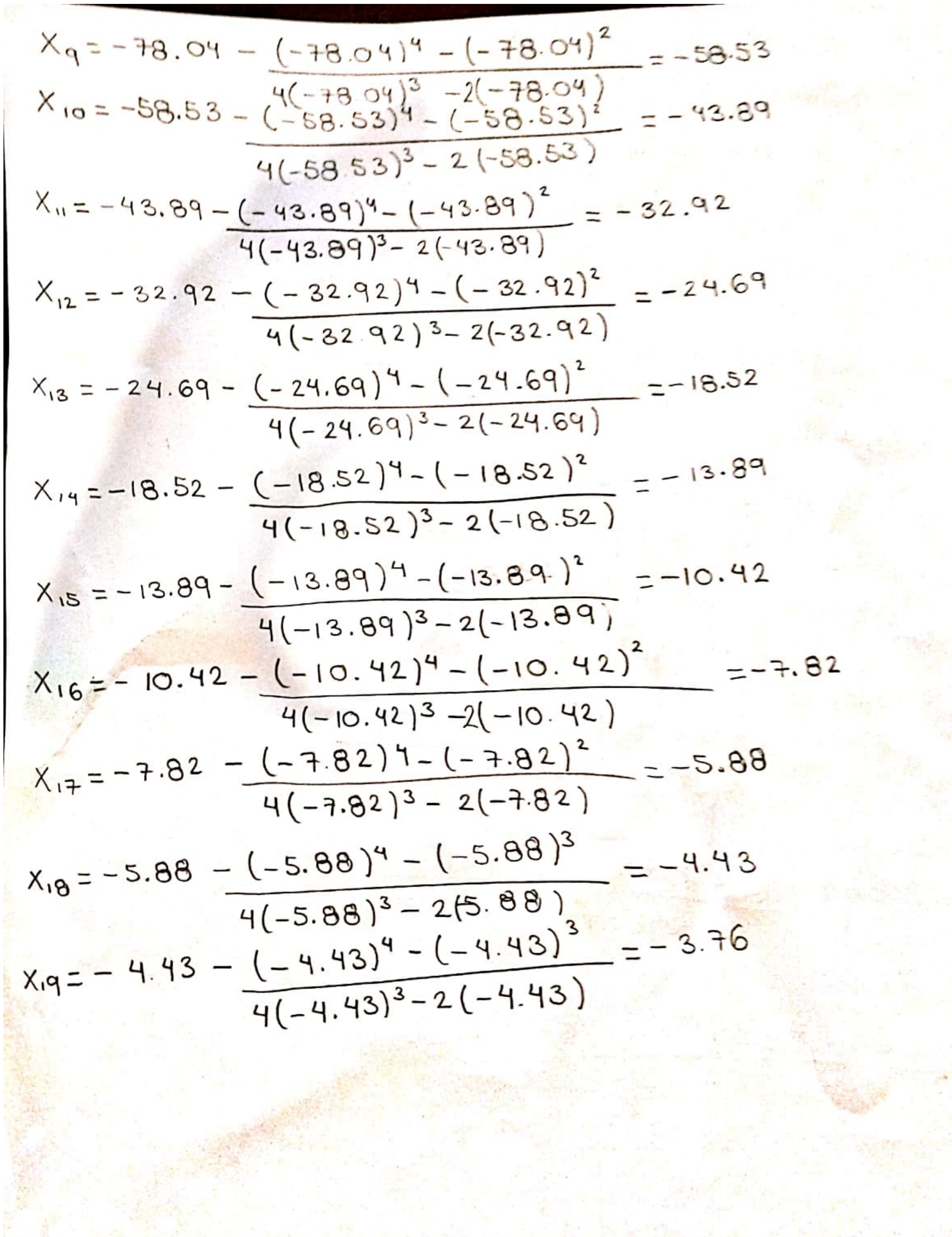
$$X_5 = -185.00$$

$$X_6 = -185 - \frac{(-185.00)^4 - (-185.00)^2}{4(-185.00)^3 - 2(-185.00)} = -138.75$$

$$X_7 = -138.75 - \frac{(-138.75)^4 - (-138.75)^2}{4(-138.75)^3 - 2(-138.75)} = -104.06$$

$$X_8 = -104.06 - \frac{(-104.06)^4 - (-104.06)^2}{4(-104.06)^3 - 2(-104.06)} = -78.04$$

Figura 11: Solución analítica de$f(x) = -x - e^x - 2$



$$X_9 = -78.04 - \frac{(-78.04)^4 - (-78.04)^2}{4(-78.04)^3 - 2(-78.04)} = -58.53$$

$$X_{10} = -58.53 - \frac{(-58.53)^4 - (-58.53)^2}{4(-58.53)^3 - 2(-58.53)} = -43.89$$

$$X_{11} = -43.89 - \frac{(-43.89)^4 - (-43.89)^2}{4(-43.89)^3 - 2(-43.89)} = -32.92$$

$$X_{12} = -32.92 - \frac{(-32.92)^4 - (-32.92)^2}{4(-32.92)^3 - 2(-32.92)} = -24.69$$

$$X_{13} = -24.69 - \frac{(-24.69)^4 - (-24.69)^2}{4(-24.69)^3 - 2(-24.69)} = -18.52$$

$$X_{14} = -18.52 - \frac{(-18.52)^4 - (-18.52)^2}{4(-18.52)^3 - 2(-18.52)} = -13.89$$

$$X_{15} = -13.89 - \frac{(-13.89)^4 - (-13.89)^2}{4(-13.89)^3 - 2(-13.89)} = -10.42$$

$$X_{16} = -10.42 - \frac{(-10.42)^4 - (-10.42)^2}{4(-10.42)^3 - 2(-10.42)} = -7.82$$

$$X_{17} = -7.82 - \frac{(-7.82)^4 - (-7.82)^2}{4(-7.82)^3 - 2(-7.82)} = -5.88$$

$$X_{18} = -5.88 - \frac{(-5.88)^4 - (-5.88)^3}{4(-5.88)^3 - 2(5.88)} = -4.43$$

$$X_{19} = -4.43 - \frac{(-4.43)^4 - (-4.43)^3}{4(-4.43)^3 - 2(-4.43)} = -3.76$$

Figura 12: Solución analítica de$f(x) = -x - e^x - 2$

2.4. d) $f(x) = -x + e^x - 2$

Para esta función las raíces se encuentran gráficamente en:

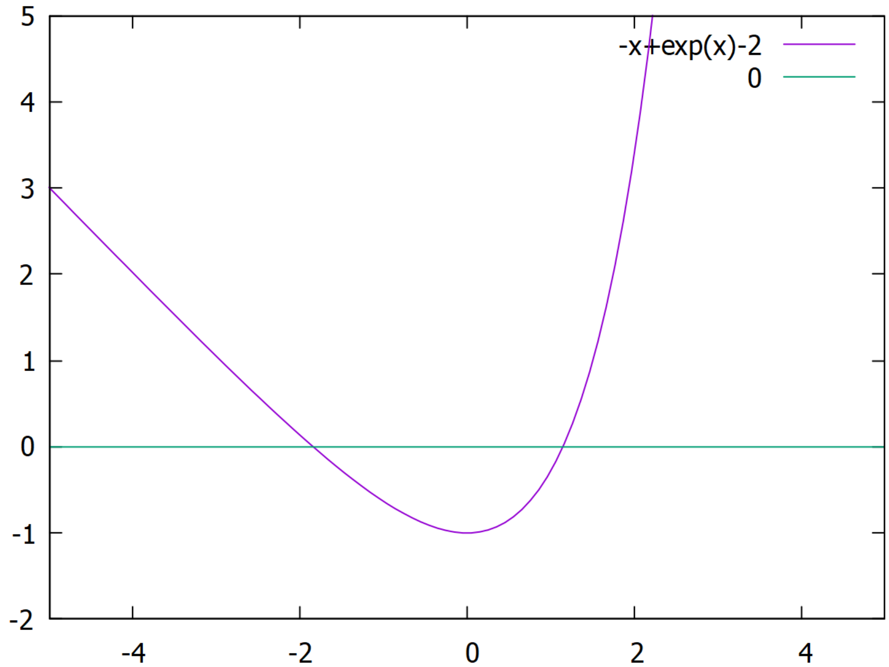


Figura 13: Gráfica de $f(x) = -x - e^{x} - 2$

Una cerca de -2 y otra cerca 1.
Tomamos el primer caso con una primera aproximación en el punto -2:

```
 ***********************************
 Dame la estimacion de la raiz (x0):
-2
 F(x0): 0.1353352832366127
 DF(x0): -0.8646647167633873
 -1.8434823572503343
 -1.8414060661579263
 -1.8414056604369762
 Iteracion:   Raiz:   Intervalo:
   3     -1.84140566      0.00000041
PAUSE statement executed.  Hit Return to continue
```

Figura 14: Primera raíz de $f(x) = -x - e^{x} - 2$

Solo le tomó 3 iteraciones y arroja un resultado de raíz aproximado de -1.84140566.
Para la segunda raíz tomamos como punto aproximado 1:

```
 ********************************
 Dame la estimacion de la raiz (x0):
1
 F(x0): -0.28171817154095474
 DF(x0): 1.7182818284590453
 1.163953413738653
 1.1464211850430086
 1.1461932587044987
 1.1461932206205836
 Iteracion:   Raiz:  Intervalo:
   4      1.14619322     -0.00000004
PAUSE statement executed.  Hit Return to continue
```

Figura 15: Seguna raíz de$f(x) = -x - e^x - 2$

Para la cuál solo se realizaron 4 iteraciones y se encontró que la solución analítica para esta raíz es 1.1419322.
De forma analítica para ambas raíces:

d) $f(x) = -x + e^{x} - 2$

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

Para la primera raiz se toma como $x_0 = -2$

$$x_1 = x_0 - \frac{f(x)}{f'(x)} = -2 - \frac{(-2 + e^{-2} - 2)}{-1 + e^{-2}} = -2.53$$

$$x_2 = 2.46 - \frac{(-2.53 + e^{2.53} - 2)}{-1 + e^{-2.53}} = \underline{-1.8355}$$

Para la segunda raiz tomamos $x_0 = 1$

$$x_1 = 1 - \frac{(-1 + e - 2)}{-1 + e} = 1.16$$

$$x_2 = 1.16 - \frac{(-1.16 + e^{1.16} - 2)}{-1 + e^{1.16}} = \underline{1.1463}$$

Figura 16: Soluciones analíticas de$f(x) = -x - e^{x} - 2$

2.5. e) $f(x) = x \sin x - \ln x + x - 2.6$

Con el objetivo de encontrar el intervalo donde se encuentran las 5 raíces requeridas graficamos $f(x)$ en GNUPLOT. El script utilizado es el siguiente.

```
set terminal epslatex color
set output 'e_graph.tex'
set zeroaxis lt -1 lc -1
unset key
set xlabel '$x$'
set ylabel '$f(x)$'

plot [-0.1:11] x*sin(x)-log(x)+x-2.6 title "$x\\sin(x)-\\ln(x)+x-2.6$" w lp pt 12 lw 2 ps 0.6
```

La gráfica nos muestra los intervalos en donde resulta conveniente hacer las iteraciones para encontrar las raices.

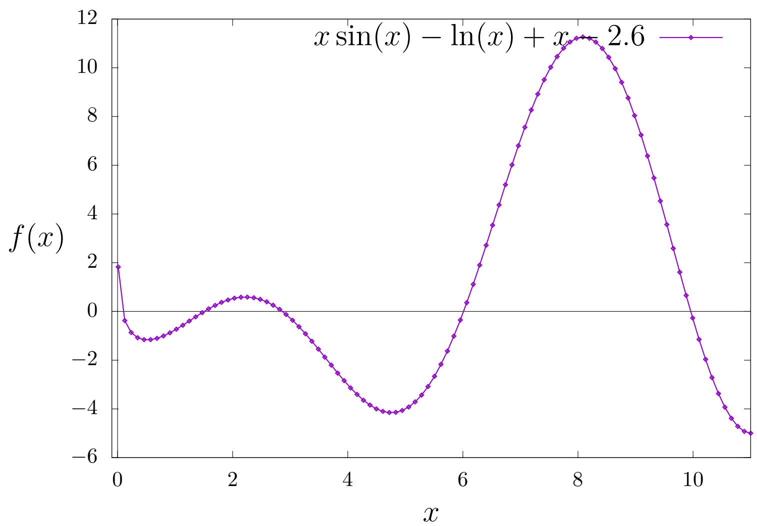
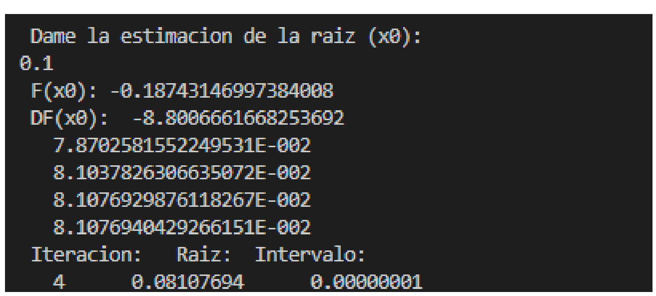


Figura 17: Gráfica de la función $f(x)$

De la figura encontramos las 5 raíces para los cuales seleccionaremos valores cercanos para asegurar una convergencia; desde la primera raíz hasta la quinta seleccionaremos los valores $0.1, 1.5, 3.0, 6.0$ y 10.0, respectivamente. Así obtenemos las raíces numéricamente:



```
 Dame la estimacion de la raiz (x0):
0.1
 F(x0): -0.18743146997384008
 DF(x0):  -8.8006661668253692
   7.8702581552249531E-002
   8.1037826306635072E-002
   8.1076929876118267E-002
   8.1076940429266151E-002
 Iteracion:   Raiz:  Intervalo:
   4      0.08107694      0.00000001
```

Figura 18: Primera solución numérica

```
 Dame la estimacion de la raiz (x0):
1.5
 F(x0):  -9.2225328346511315E-003
 DF(x0):   1.4369341224389420
   1.5064182015658432
   1.5064314174268594
   1.5064314174838573
 Iteracion:   Raiz:  Intervalo:
   3      1.50643142      0.00000000
```

Figura 19: Segunda solución numérica

```
 Dame la estimacion de la raiz (x0):
3
 F(x0): -0.27525216912107653
 DF(x0):  -2.1621908150748022
   2.8726975587899006
   2.8622355599047995
   2.8621582889568020
   2.8621582847198570
 Iteracion:   Raiz:  Intervalo:
   4      2.86215828     -0.00000000
```

Figura 20: Tercera solución numérica

```
 Dame la estimacion de la raiz (x0):
6
 F(x0):  -6.8252363054178478E-002
 DF(x0):   6.3149395550366032
   6.0108080785982727
   6.0107749233356920
   6.0107749230267746
 Iteracion:   Raiz:  Intervalo:
   3      6.01077492     -0.00000000
```

Figura 21: Cuarta solución numérica

```
 Dame la estimacion de la raiz (x0):
10
 F(x0): -0.34279610652031156
 DF(x0):  -8.0347364016538947
   9.9573357370566935
   9.9577392121045047
   9.9577392453597806
 Iteracion:   Raiz:  Intervalo:
   3      9.95773925      0.00000003
```

Figura 22: Quinta solución numérica

Por lo tanto, nuestras soluciones serían: $x_1 \simeq 0.081$, $x_2 \simeq 1.5061$, $x_3 \simeq 2.862$, $x_4 \simeq 6.010$ y $x_5 \simeq 9.957$.

2.6. $f(x) = 19500 - 800\frac{(1+\frac{x}{4})^{20}}{\frac{x}{4}}$

Graficamos esta función con el objetivo de obtener un valor aproximado de la raíz y proceder con un método numérico. El script de GNUPLOT y la gráfica son los siguientes:

```
set terminal epslatex color
set output 'f_graph.tex'
set zeroaxis lt -1 lc -1
unset key
set xlabel '$x$'
set ylabel '$f(x)$'

plot [-0.5:0.5] 19500-800*((1+(x/4))**(20)-1)/(x/4) title "$19500-800\\frac{(1+\\frac{x}{4})
^{20}}{\\frac{x}{4}}$" w lp pt 12 lw 2 ps 0.6
```

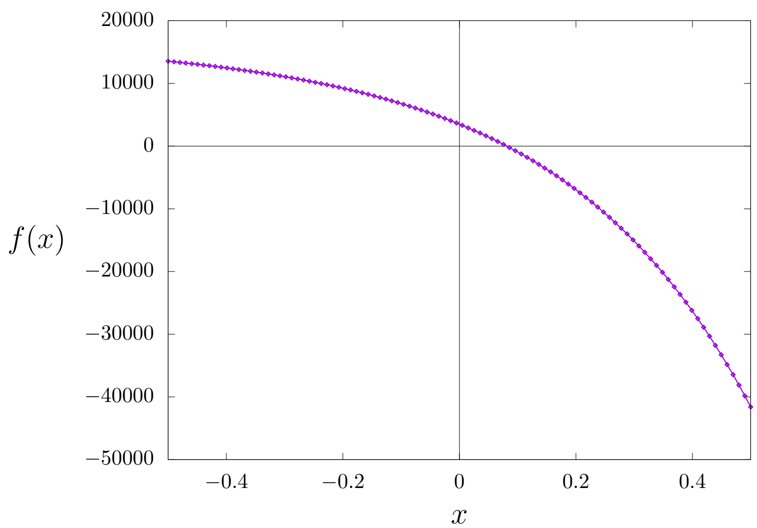


Figura 23: Gráfica de la función $f(x)$

En donde hallamos que un valor inicial conveniente para el método de Newton-Raphson es $x_0 = 0.2$, por lo que obtenemos el siguiente resultado.

```
 Dame la estimacion de la raiz (x0):
0.2
 F(x0):  -6952.7632823107569
 DF(x0):  -69892.199218497437
  0.10052161242522956
   8.1834019602678068E-002
   8.1281414869487340E-002
   8.1280950009675557E-002
 Iteracion:   Raiz:  Intervalo:
   4      0.08128095     -0.00000046
```

Figura 24: Solución numérica para $f(x) = 0$

Por lo tanto, nuestra solución para la ecuación es $x \simeq 0.0812$.

3. Conclusiones

Alexia Sofía Ibarra García: El método de Newton permite calcular las raíces de cualquier función de forma numérica, es rápido, sencillo y eficiente. Sin embargo, se presentan ciertas complicaciones, como tener que calcular la derivada que en ocasiones es demasiado compleja o que en el proceso de la iteraciones se encuentre con ciertos problemas o que se llegue a ciclar. Las aplicaciones de este tipo de métodos son vastas y permiten la resolución de problemas complejos con poco poder computacional.

Gustavo Hernández Ángeles: El método de Newton-Raphson nos provee soluciones muy acertadas con un requerimiento de poder computacional relativamente bajo comparado con el método de bisección. A pesar de contar con las desventajas sobre funciones cuya derivada tenga un valor cercano a cero, produce muy buenos resultados en pocas iteraciones.

Diego Alejandro Téllez Martínez: Mi conclusión es que este método es universal para solucionar ecuaciones que de manera analítica no se podrían determinar y los resultados obtenidos son bastante precisos.

Excepto para funciones donde la función tiene la pendiente de la tangente puede aproximarse en gran magnitud a cero, entonces el método puede fallar o presentar lenta convergencia

Cristian Joel Gallegos Yañez: El método de Newton ha demostrado ser una excelente opción para realizar cálculos de manera rápida y eficaz pese a la condición de la existencia de la derivada de la función en cuestión. Sin embargo es de suma importancia aumentar los conocmientos para poder aplicar los mismos en problemas cada vez más complejos al tiempo que implementamos nuevas herramientas computacionales anunadas a funciones cada vas con mayor complejidad.