Tarea 5 - Métodos numéricos
Giovanni Gamaliel López Padilla

Problema 1

A lighthouse L is located on a small island 5 km north of a point A on a straight east-west shoreline. A cable is to be laid from L to point B on the shoreline 10 km east of A. The cable will be laid through the water in a straight line from L to a point C on the shoreline between A and B, and from there to B along the shoreline. (see Figure 1). The part of the cable lying in the water costs \$5.000/km, and the part along the shoreline costs \$3, 000/km. Where should C be chosen to minimize the total cost of the cable?

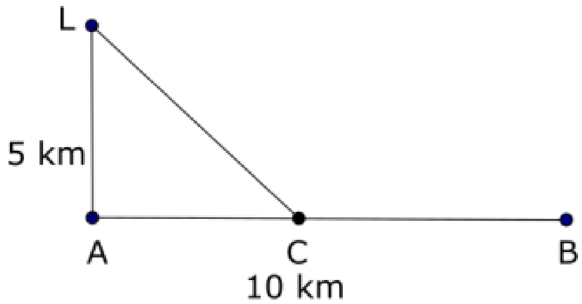

Figura 1: Problema 1

Llamnado x al segmento de recta $\overline{AC}$, y a $\overline{LC}$, entonces el segmento $\overline{CB}$ puede ser calculado como $10 - x$. Las definiciones antes mencionadas se encuentran en la figura 2.

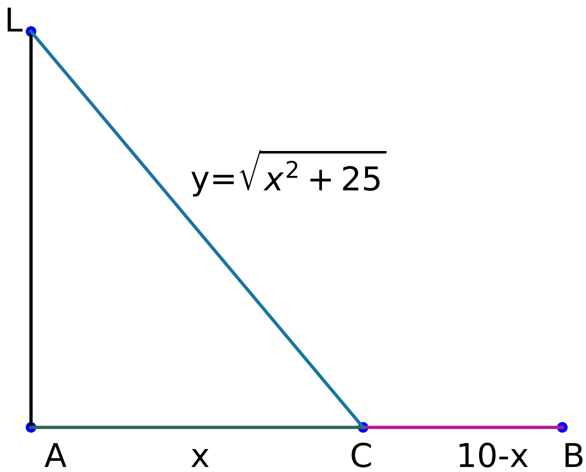

Figura 2: Representación de las definiciones de cada segmento de linea.

Definiendo la función de costo de cada linea se obtiene la función 1.

$$C(x) = 3000(10 - x) + 5000\sqrt{x^2 + 25} \tag{1}$$

Realizando la derivada con respecto a x de le función 1 para encontrar los valores críticos.

$$\frac{dC(x)}{dx} = -3000 + \frac{5000(x)}{\sqrt{x^2+25}}$$

Encontrando los valores criticos se obtiene lo siguiente:

$$\begin{aligned}
-3000 + \frac{5000(x)}{\sqrt{x^2+25}} &= 0 \\
\frac{5000(x)}{\sqrt{x^2+25}} &= 3000 \\
5x &= 3\sqrt{x^2+25} \\
25x^2 &= 9(x^2+25) \\
16x^2 - 225 &= 0 \\
(4x-15)(4x+15) &= 0 \\
x_1 &= \frac{15}{4} \\
x_2 &= -\frac{15}{4}
\end{aligned}$$

La solución x_2 es despreciada, ya que su sentido físico no admisible, por lo tanto el punto C debe estar a 3.75km del punto A.

Problema 2

Implement the following algorithms: Bisection, Newton, and Secant methods for optimization in 1D.

Método de bisección

El método de bisección supone que la función f es unimodal, esto quiere decir que existe un único valor crítico. La función f también debe de cumplir que es continuamente diferenciable. Dada una x en el intervalo $[a_0, b_0]$, el método obtiene una secuencia de intervalos tal que:

$$[a_0, b_0] \supset [a_1, b_1] \supset [a_2, b_2] \supset \cdots \supset [a_k, b_k]$$

En cada intervalo, el valor crítico estara contenido. El algoritmo para obtener el valor mínimo de una función es el siguiente:

```
1      input: df(x), [a,b], tolerancia
2      output: x, approximacion valor minimo
3      x=(a+b)/2
4      while |b-a| > tolerancia:
5          if (df(x) > 0):
6              b=x
7          else:
8              a=x
9          x=(a+b)/2
10     return x
```

En el caso que se quiera aplicar el algoritmo para obtener el valor máximo de una función se deberá realizar una modificación en la linea 4, ya que si no, el algoritmo arrojara como resultado alguno de los extremos del intervalo. La modificación sería la siguiente:

```
input: df(x), [a,b], tolerancia
output: x, approximacion valor maximo
x=(a+b)/2
while |b-a| > tolerancia:
    if (df(x) < 0):
        b=x
    else:
        a=x
    x=(a+b)/2
return x
```

La implementación del algoritmo se encuentra en la carpeta Problema_3a y Problema_3b en el archivo bisection.h.

Método de Newton

El método de Newton supone que tenemos una f, la cual es de clase C^2, lo cual quiere decir que la primer y segunda derivada de f son continuas. El algoritmo del descendiente es el siguiente: Dado un valor pequeño de h, donde $h > 0$, entonces el siguiente valor de x es:

$$x_{k+1} = x_k - hf'(x_k) \tag{2}$$

Con la ecuación 2 se obtienen los siguientes casos:

- Si $f'(x_0) > 0$ entonces $x_{k+1} < x_0$, ya que existe $\delta > 0$ para $f'(x) > 0$ para todo $|x - x_0| < \delta$. Por lo tanto f es creciente en el interalo.
- Si $f'(x_0) < 0$ entonces $x_{k+1} > x_0$, ya que existe $\delta > 0$ para $f'(x) < 0$ para todo $|x - x_0| < \delta$. Por lo tanto f es decreciente en el interalo.

Podemos aproximar la función f usando una serie de Taylor hasta segundo orden. Entnces:

$$f(x) \approx q(x) \overset{def}{=} f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}f''(x_0)(x - x_0)^2$$

Por lo que al querer minimizar f, el método de Newton genera la siguiente secuencia

$$x_{k+1} = argmin_x q(x)$$

Calculando la primer derivada de $q(x)$, obtenemos lo siguiente:

$$q'(x) = f'(x_k) + f''(x_k)(x - x_k)$$

En donde al encontrar sus valores críticos, obtenemos la secuencia para el método de Newton (ecuación 3).

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)} \tag{3}$$

El algoritmo del método de Newton es el siguiente:

```
input: df, ddf, x0, tolerancia
output: x, aproxximacion de sus valores criticos
while |df| > tolerancia:
    x = x- df/dff
return x
```

La implementación del método de Newton se encuentra en las carpetas Problema_3a y Problema_3b en el archivo newton.h.

Método de secante

El método de la secante es una alternativa al método de Newton. Supongamos que no se tiene la manera de obtener la segunda derivada de f. Entonces podemos aproximar este valor con la ecuación 4.

$$f''(x_k) = \frac{f'(x_k) - f'(x_{k-1})}{x_k - x_{k-1}} \tag{4}$$

Sustituyendo la ecuación 4 en 3, obtenemos lo siguiente:

$$\begin{aligned} x_{k+1} &= x_k - \frac{x_k - x_{k-1}}{f'(x_k) - f'(x_{k-1})} f'(x_k) \\ x_{k+1} &= \frac{f'(x_k)x_{k-1} - f'(x_k)x_k}{f'(x_k) - f'(x_{k-1})} \end{aligned} \tag{5}$$

Por lo tanto, el algoritmo del método de la secante es el siguiente:

```
input: df, x0, x1, tolerancia
output: x1, aproxximacion de sus valores criticos
while |df| > tolerancia:
    x2 = x1 - df(x1)(x1-x0)/(df(x1)-df(x0))
    x0, x1 = x1, x2
return x1
```

La implementación del método de Newton se encuentra en las carpetas Problema_3a y Problema_3b en el archivo secant.h.

Problmea 3

Problema 3a

Find the minimum value and minimum point of the function 6 on the interval [-1, 1] using the previous implemented algorithms. Compare the results in terms of number of iterations.

$$f(x) = -sin(x) + x^2 + 1 \tag{6}$$

La implementación de este problema se encuentra en la carpeta Problema_3a. El outut del programa es el siguiente:

```
Los valores minimos de la funcion son:
    Biseccion:
            Iteraciones     Aproximacion     Tolerancia
                      1     0.500000000000   1.000000000000
                      2     0.250000000000   0.500000000000
                      3     0.375000000000   0.250000000000
                      4     0.437500000000   0.125000000000
                      5     0.468750000000   0.062500000000
                      6     0.453125000000   0.031250000000
                      7     0.445312500000   0.015625000000
                      8     0.449218750000   0.007812500000
                      9     0.451171875000   0.003906250000
                     10     0.450195312500   0.001953125000
                     11     0.449707031250   0.000976562500
                     12     0.449951171875   0.000488281250
                     13     0.450073242188   0.000244140625
                     14     0.450134277344   0.000122070312
                     15     0.450164794922   0.000061035156
                     16     0.450180053711   0.000030517578
                     17     0.450187683105   0.000015258789
```

```
                    18      0.450183868408  0.000007629395
                    19      0.450181961060  0.000003814697
                    20      0.450182914734  0.000001907349
                    21      0.450183391571  0.000000953674
                    22      0.450183153152  0.000000476837
                    23      0.450183033943  0.000000238419
                    24      0.450183093548  0.000000119209
                    25      0.450183123350  0.000000059605
                    26      0.450183108449  0.000000029802
                    27      0.450183115900  0.000000014901
                    28      0.450183112174  0.000000007451
                    29      0.450183110312  0.000000003725
                    30      0.450183111243  0.000000001863
                    31      0.450183111709  0.000000000931
                    32      0.450183111476  0.000000000466
                    33      0.450183111359  0.000000000233
                    34      0.450183111301  0.000000000116
                    35      0.450183111330  0.000000000058
                    36      0.450183111345  0.000000000029
                    37      0.450183111352  0.000000000015
                    38      0.450183111348  0.000000000007
                    39      0.450183111347  0.000000000004
                    40      0.450183111346  0.000000000002
                    41      0.450183111345  0.000000000001
            Solucion: x = 0.450183111345
            Punto de interes: (0.450183111345 , 0.767534424842)
    Newton:
            Iteraciones     Aproximacion    Tolerancia
                    1       0.486293632179  0.088517391084
                    2       0.450418282152  0.000572696779
                    3       0.450183104164  0.000000017319
                    4       0.450183111276  0.000000000000
            Solucion: x = 0.450183111276
            Punto de interes: (0.450183111276 , 0.767534424842)
    Secante:
            Iteraciones     Aproximacion    Tolerancia
                    1       0.270150596267  2.540301727194
                    2       0.524212950303  0.423428390794
                    3       0.447630513809  0.182708997709
                    4       0.450149038877  0.006212974468
                    5       0.450183127397  0.000082970297
                    6       0.450183111303  0.000000039191
                    7       0.450183111303  0.000000000000
            Solucion: x = 0.450183111303
            Punto de interes: (0.450183111303 , 0.767534424842)
```

Comparando estos resultados por las iteracione que tienen que realizar para obtener una respuesta vemos que el método de Newton es el más eficiente de todos, esto se debe que esta basado en un descendiente que se dirige hacia los valores críticos que queremos encontrar. El menos eficiente es el de bisección, esto se debe a que esta basado en una diferencia en los límites del intervalo.

Comparando las soluciones observamos que para el redondeo a 9 decimales las tres tienen la misma solución.

El comando para compilar el programa es el siguiente:

```
gcc -Wall -Wextra -Werror -pedantic -ansi -o main.out main.c -lm -std=c11
```

Problema 3b

Compare and comment the results obtained for each algorithm on the interval [-1, 1] with function 7.

$$f(x) = sin(x) - x^2 + 1 \tag{7}$$

La implementación de este problema se encuentra en la carpeta Problema_3b. El outut del programa es el siguiente:

```
    Los valores minimos de la funcion son:
        Biseccion:
                Iteraciones     Aproximacion    Tolerancia
                        1       0.500000000000  1.000000000000
                        2       0.250000000000  0.500000000000
                        3       0.375000000000  0.250000000000
                        4       0.437500000000  0.125000000000
                        5       0.468750000000  0.062500000000
                        6       0.453125000000  0.031250000000
                        7       0.445312500000  0.015625000000
                        8       0.449218750000  0.007812500000
                        9       0.451171875000  0.003906250000
                        10      0.450195312500  0.001953125000
                        11      0.449707031250  0.000976562500
                        12      0.449951171875  0.000488281250
                        13      0.450073242188  0.000244140625
                        14      0.450134277344  0.000122070312
                        15      0.450164794922  0.000061035156
                        16      0.450180053711  0.000030517578
                        17      0.450187683105  0.000015258789
                        18      0.450183868408  0.000007629395
                        19      0.450181961060  0.000003814697
                        20      0.450182914734  0.000001907349
                        21      0.450183391571  0.000000953674
                        22      0.450183153152  0.000000476837
                        23      0.450183033943  0.000000238419
                        24      0.450183093548  0.000000119209
                        25      0.450183123350  0.000000059605
                        26      0.450183108449  0.000000029802
                        27      0.450183115900  0.000000014901
                        28      0.450183112174  0.000000007451
                        29      0.450183110312  0.000000003725
                        30      0.450183111243  0.000000001863
                        31      0.450183111709  0.000000000931
                        32      0.450183111476  0.000000000466
                        33      0.450183111359  0.000000000233
                        34      0.450183111418  0.000000000116
                        35      0.450183111388  0.000000000058
                        36      0.450183111374  0.000000000029
                        37      0.450183111367  0.000000000015
                        38      0.450183111370  0.000000000007
                        39      0.450183111368  0.000000000004
                        40      0.450183111368  0.000000000002
                        41      0.450183111367  0.000000000001
                Solucion: x = 0.450183111367
                Punto de interes: (0.450183111367 , 1.232465575158)
        Newton:
                Iteraciones     Aproximacion    Tolerancia
                        1       0.486253486285  0.088418337763
                        2       0.450418281731  0.000572695669
                        3       0.450183125641  0.000000035083
                        4       0.450183111233  0.000000000000
                Solucion: x = 0.450183111233
                Punto de interes: (0.450183111233 , 1.232465575158)
        Secante:
                Iteraciones     Aproximacion    Tolerancia
                        1       0.270150596150  2.540301726750
                        2       0.524212950377  0.423428391017
                        3       0.447630513789  0.182708997931
                        4       0.450149038813  0.006212974357
                        5       0.450183127333  0.000082970297
                        6       0.450183111285  0.000000039080
                        7       0.450183111285  0.000000000000
```

```
                    Solucion: x = 0.450183111285
                    Punto de interes: (0.450183111285 , 1.232465575158)
```

De igual manera se comprueba que en terminos del número de iteraciones, el método de Newton es el más eficiente y el menos eficiente es el método de bisección. Y las diferencias entre resultados se presenta a partir del añadir más de 9 decimales a la solución.
En la grafica 3 se visualizando graficamente la solución del problema 3a y 3b.

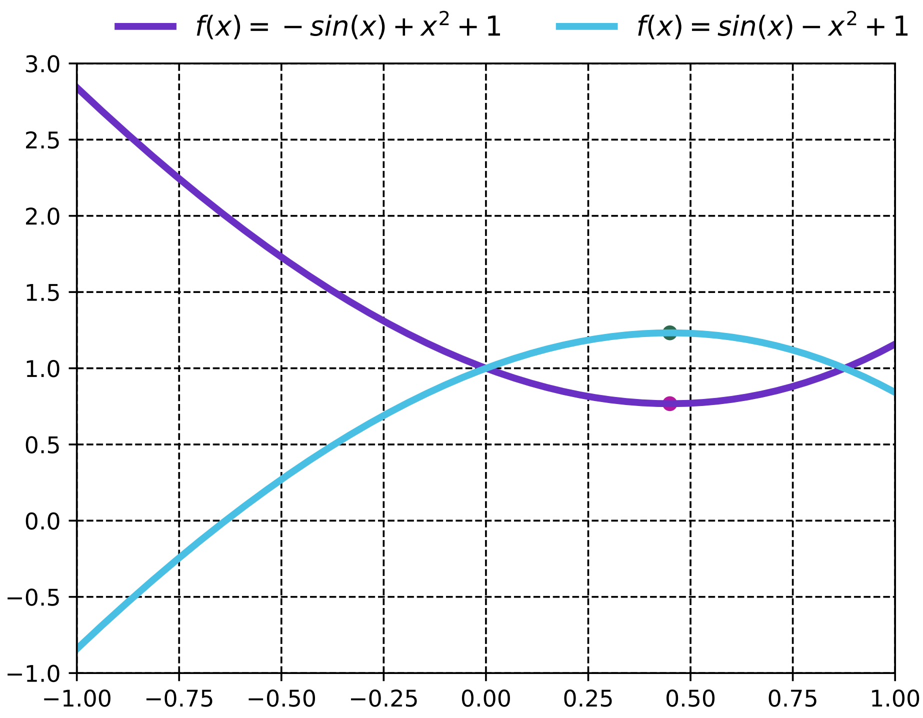

Figura 3: Grafica de la función 6 y 7 con su valor mínimo y máximo respectivamente.

El comando para compilar el programa es el siguiente:

```
gcc -Wall -Wextra -Werror -pedantic -ansi -o main.out main.c -lm -std=c11
```