UNIVERSIDAD CATÓLICA DE TEMUCO | FACULTAD DE INGENIERÍA

Etapa 2- Actividad 1

José Moya Fernando Torres Maximiliano Gonzalez

Eduardo Lobos

Abril 2021

1. Introducción

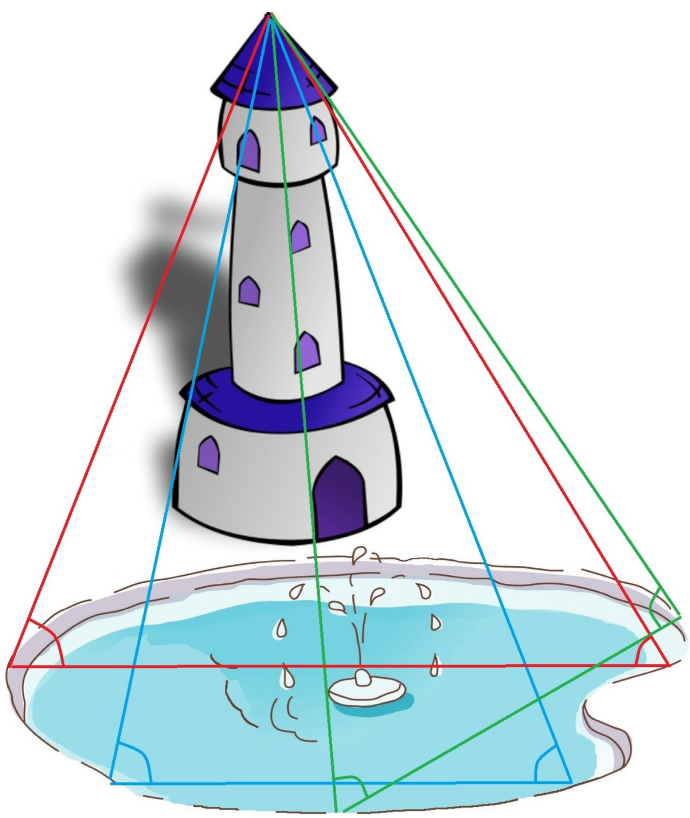
A modo de introducción podemos decir que para realizar la actividad debemos utilizar el concepto de distancia y sistemas tridimensionales, como también leer la novela Las Torres resaltando con mayor énfasis la sección La caminata, donde se plantea que el objetivo es finalizar la tarea de K, ya que ella no la pudo completar. Para ello a través de estos apuntes entregados debemos determinar la altura de la torre mediante métodos como distancia , entre otros que se irán adhiriendo en el transcurso de las actividades, como objetivo anexo es poder ir utilizando un modelo matemático propuesto por nuestro grupo de trabajo que debiera ser un sistema de ecuaciones lineales y no-lineales, para reforzar nuestro planteamiento
El texto de las torres se trata de una joven topógrafa que llega a un pueblo para hacer su practica , el cual ese se basaba en reglas ya establecidas por el dueño del lugar, para poder estar o irse de este dicho pueblo tenia que tener permiso del dueño de los terrenos, al día siguiente esta fue a una entrevista donde le platearon el de determinar la distancia de la torre ,podemos determinar la altura de las torres. Si hay una laguna, entonces debería tener la forma de un círculo con el radio indicado.
El texto de la caminata a las torres la dejaron para el penúltimo día de la práctica como actividad de desempeño integrado.El objetivo del aprendizaje de la caminata era finalmente formular un método para medir la altura de las torres ,Pero sí conoces la ubicación de los puntos de la medición en este lado de la laguna, y la distancia de cada uno de estos puntos al tope. Eso corresponde a tres triángulos, o más, que corresponden a seis puntos, o más, en nuestro lado. El profesor Abrió el armario, se vio un sin número de apuntes, hojas, libros, cuadernos con resolución de problemas, resolución sin problemas, problemas en contexto, problemas sin contexto, problemas resueltos, problemas no resueltos, unos apretados al lado de otros, todos cuidadosamente etiquetados y descritos.
Para poder realizar la presente actividad debemos saber que es un sistemas tridimensionales de coordenadas que se utilizan para ubicar un punto en un plano, se requieren dos números. Por lo tanto, el plano se llama bidimensional. Cualquier punto en el espacio está representado por un triple ordenado de números reales.

2. Planteamiento

Como se explicó en la introducción, el objetivo de este problema es calcular la altura de la torre ubicada en las afueras de lago. Sin embargo, existen ciertas limitaciones que debemos de tomar en cuenta a la hora de plantear nuestra solución.

- No sabemos si la torre esta inclinada
- No sabemos la longitud del lago
- No sabemos la posición exacta de la torre

Pero, existen ciertos datos que podemos calcular para enfrentar este problema, por ejemplo, podemos asignar puntos ubicados que se encuentren en nuestro lado del lago. Con esta pista nosotros podemos formular un modelo matemático basado en las distancias en un plano 3D.



Podemos trazar triángulos cuya base se defina desde el lago y la punta del triángulo esté ubicada en la punta de la torre, de esta manera, utilizando fórmulas trigonométricas podemos calcular la altura de la torre sin importar si está inclinada.

$$d(x,y) = \sqrt{(X_1) - (Y_1)^2 + (X_2) - (Y_2)^2 + (X_3) - (Y_3)^2}$$

Definiremos 3 triángulos, cuya base este en nuestro lado de la laguna y su punta este en la punta de la torre, sus bases estarán dadas por los puntos, B y C para el primero, D y E para el segundo y G y F para el tercero, todos tienen su punta en A.
Con esto ya podemos plantear nuestro sistema de ecuaciones.

No conocemos: A
Conocemos: B,C,D,E,F,G, los ángulos donde esta cada uno de estos puntos , |BC|,|DE|,|FG|.
primero debemos calcularla distancia que hay de lados muestra puntos hasta A.

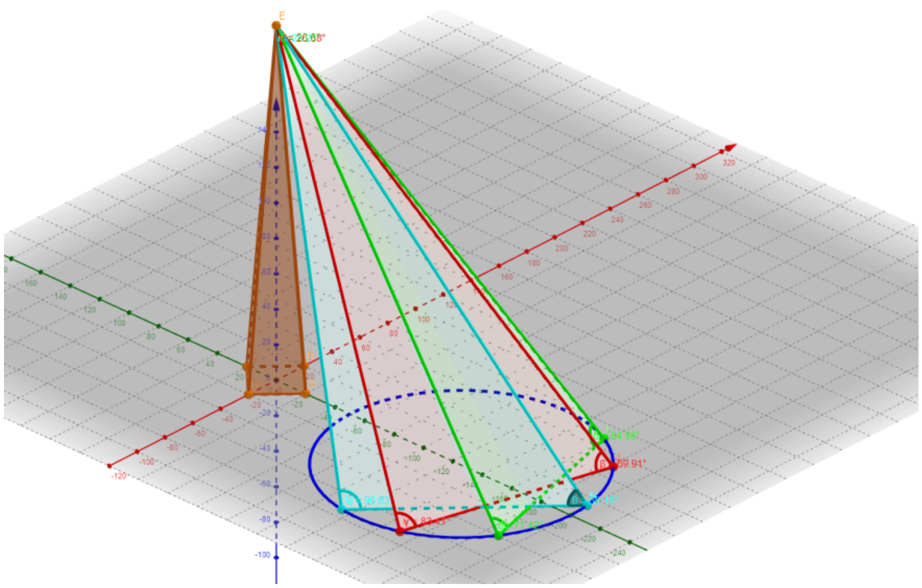
$$|BA| = \frac{|BC| * sen(ang(C))}{sen(ang(A))}$$

$$|BA| = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + (a_3 - b_3)^2}$$

$$(|BA|)^2 = (a_1 - b_1)^2 + (a_2 - b_2)^2 + (a_3 - b_3)^2$$
$$f_1(a_1, a_2, a_3) = (a_1 - b_1)^2 + (a_2 - b_2)^2 + (a_3 - b_3)^2 - (|BA|)^2$$
$$f_2(a_1, a_2, a_3) = (a_1 - c_1)^2 + (a_2 - c_2)^2 + (a_3 - c_3)^2 - (|CA|)^2$$
$$f_3(a_1, a_2, a_3) = (a_1 - d_1)^2 + (a_2 - d_2)^2 + (a_3 - d_3)^2 - (|DA|)^2$$
$$f_4(a_1, a_2, a_3) = (a_1 - e_1)^2 + (a_2 - e_2)^2 + (a_3 - e_3)^2 - (|EA|)^2$$
$$f_5(a_1, a_2, a_3) = (a_1 - f_1)^2 + (a_2 - f_2)^2 + (a_3 - f_3)^2 - (|FA|)^2$$
$$f_6(a_1, a_2, a_3) = (a_1 - g_1)^2 + (a_2 - g_2)^2 + (a_3 - g_3)^2 - (|GA|)^2$$

3. Validación

Para validar el sistema planteado, debemos de asignarles valores a nuestros puntos y resolver los sistemas de ecuaciones. Para facilitar todo el proceso de validación, decidimos modelar toda la escena en algún programa 3D, con el fin de visualizar de mejor manera toda la situación y obtener algunos valores iniciales que nos servirán para trabajar de manera más rápida y fácil. El programa que decidimos utilizar es GeoGebra3D, ya que en esta se puede modelar escenas y obtener en un plano cartesiano las ubicaciones, ángulos y distancia entre los puntos que usaremos, de esta manera, ubicando todos los puntos en la escena, pudimos obtener los ángulos que utilizaban las lineas trazadas desde la base del lago hacia la punta de la torre, los cuales se pueden observar en la siguiente imagen:



Tomamos la punta de la torre (y la de todos los triángulos por esto mismo)como A, después el triangulo celeste esta formado por ABC, el rojo por ADE y el verde por AGF, la posición de estos puntos esta listada a continuación.

$$B = (-73,41; -113,86; 0)$$
$$C = (12,71; -188,05; 0)$$
$$D = (-69,92; -150,08; 0)$$
$$E = (50,21; -181,39; 0)$$
$$G = (67,89; -158,45; 0)$$
$$F = (-39,57; -188,24; 0)$$

GeoGebra también nos da la distancia entre puntos, de esto nosotros sacamos solo lo que por ejercicio conocíamos, que es la distancia de los puntos a nuestro lado de la laguna, estas distancias son.

$$|BC| = 121,13$$
$$|DE| = 124,14$$
$$|GF| = 111,52$$

Con estas distancias y los ángulos internos de los triángulos podemos usar el teorema del seno para conseguir las distancias de los lados de los triángulos que van hacia la torre, nos quedara así.

$$|BA| = \frac{121,13 * sen(58,16)}{sen(25,21)} |CA| = \frac{121,13 * sen(96,63)}{sen(25,21)}$$
$$|BA| = \frac{124,14 * sen(69,91)}{sen(26,68)} |CA| = \frac{124,14 * sen(83,41)}{sen(26,68)}$$
$$|BA| = \frac{111,52 * sen(71,42)}{sen(23,6)} |CA| = \frac{111,52 * sen(84,98)}{sen(23,6)}$$
$$|BA| = 241,6$$
$$|DA| = 282,5$$
$$|CA| = 259,7$$
$$|EA| = 274,6$$
$$|FA| = 264,0$$
$$|GA| = 277,5$$

Con los datos que tenemos ya podemos reemplazar los valores y quedar con un sistema de ecuaciones funcional.

$$F_1 = (a_1 + 73,41)^2 + (a_2 + 113,86)^2 + (a_3 - 0)^2 - (241,6)^2$$
$$F_2 = (a_1 - 12,71)^2 + (a_2 + 188,05)^2 + (a_3 - 0)^2 - (282,5)^2$$
$$F_3 = (a_1 + 69,92)^2 + (a_2 + 150,08)^2 + (a_3 - 0)^2 - (259,7)^2$$
$$F_4 = (a_1 - 50,21)^2 + (a_2 + 181,39)^2 + (a_3 - 0)^2 - (274,6)^2$$
$$F_5 = (a_1 - 67,89)^2 + (a_2 + 158,45)^2 + (a_3 - 0)^2 - (264)^2$$
$$F_6 = (a_1 + 39,57)^2 + (a_2 + 188,24)^2 + (a_3 - 0)^2 - (277,5)^2$$

Ahora usaremos un programa hecho en Python que nos ayudara a resolver el sistema, para esto necesitaremos 3 de las ecuaciones anteriores, ya que si usamos mas tendríamos un sistema con mas ecuaciones que variables, lo que no nos daría solución, el código de este programa esta en la siguiente imagen

```python
import numpy as np
def newthon_method(F,J,x0,tol):
    x=x0
    error=1e3
    n=0
    while error>tol:
        dx=-np.linalg.solve(J(*x),F(*x))
        error=np.linalg.norm(dx)/np.linalg.norm(x)
        x+=dx
        n+=1
    print("Numero de Iteraciones: ",n,"\n")
    print("Resultados Finales:")
    print(x)
    print("Resultados Finales Redonados:")
    print(round(x[0]),",",round(x[1]),",",round(x[2]))

F=lambda x1,x2,x3: [
                (((x1+73.41)**2)+((x2+113.86)**2)+((x3-0)**2)-241.6**2),
                (((x1-12.71)**2)+((x2+199.05)**2)+((x3-0)**2)-282.5**2),
                (((x1+69.92)**2)+((x2+150.09)**2)+((x3-0)**2)-259.7**2)
                #(((x1-50.21)**2)+((x2+181.39)**2)+((x3-0)**2)-274.6**2),
                #(((x1-67.89)**2)+((x2+158.45)**2)+((x3-0)**2)-264**2),
                #(((x1+39.57)**2)+((x2+188.24)**2)+((x3-0)**2)-277.5**2)
                ]
J=lambda x1,x2,x3: [
                [2*(x1+73.41),2*(x2+113.86),2*(x3)],
                [2*(x1-12.71),2*(x2+199.05),2*(x3)],
                [2*(x1+69.92),2*(x2+150.09),2*(x3)]
```

```
                 #[2*(x1-50.21),2*(x2+181.39),2*(x3)],
                 #[2*(x1-67.89),2*(x2+158.45),2*(x3)],
                 #[2*(x1+39.57),2*(x2+188.24),2*(x3)]
                 ]

ValorInicial=[1,1,1]
Tolerancia=1e-5
newthon_method(F,J,ValorInicial,Tolerancia)
```

Al ejecutar el programa, realiza las iteraciones necesarias para llegar al resultado, que corresponde al punto (0,0,200), con el cual si lo comprobamos con nuestro modelo en GeoGebra3D podemos observar que es el valor correcto, y que por ende, el modelo planteado y el programa son correctos.

4. Método de Newton

El método de Newton-Raphson es un algoritmo que permite encontrar las raíces de una función, es decir, cuando una función f(x)=0. También puede ser utilizada para encontrar el máximo o mínimo de una función, sin embargo, para esta actividad se utiliza una adaptación de este algoritmo utilizada para encontrar la solución de un sistema de ecuaciones. El método de Newton-Raphson para sistemas de ecuaciones se utiliza para resolver N ecuaciones con N incógnitas, es un requerimiento que tengamos la misma cantidad de ecuaciones e incógnitas. El algoritmo original consiste en utilizar la tangente trazada desde el punto inicial dado y su intersección con la recta X, el método adaptado para sistemas indica que para una iteración k, los puntos de la iteración van a ser dados por la ecuación

$$x_{k+1} = x_k - J^{-1} * F$$

donde J corresponde a la matriz Jacobiana originada de las derivadas parciales de nuestras funciones originales.
El programa creado en Python toma como valores iniciales las funciones proporcionadas, la matriz Jacobiana, los valores iniciales para las funciones, y una tolerancia, la cual corresponde a la tolerancia de errores de valores. Estos datos entran al proceso newthon method donde recibe estos datos, y en un ciclo while, se va calculando la solución del sistema con las funciones, la matriz jacobiana y los valores de las x utilizando el método de numpy linalg.solve, y posteriormente se calcula el error dado por la division de las matrices de la solución y los valores de x por medio del método numpy linalg norm, la cual calcula el error del modelo dado. Luego se repite el ciclo esta vez con nuevos valores de X. Finalmente, cuando el error de cálculo sea mayor a la tolerancia, se detiene el ciclo y el algoritmo termina, dando como resultado una aproximación muy cercana a los valores que resuelven nuestro sistema de ecuaciones. Para mejorar el entendimiento de estos valores, se aproximan al entero más cercano.