Maximiliano Cartes - Jan Seiffert - Diego Zapata
Cálculo Avanzado - Stefan Martin Berres

Cálculo desde la distancia

Introducción

Se tiene por entendido que las aplicaciones del cálculo en contexto de la vida cotidiana son muchas. En este caso se nos presenta el caso de Kamilla, una alumna de topografía la cual para hacer su práctica tiene que lograr medir la altura de las torres del paine con la aplicación del cálculo para ello. Teniendo en cuenta que desde nuestro punto de vista, poniéndonos en el lugar de Kamilla, estamos separados de las torres por un lago, Debemos calcular la altura a partir de nuestra distancia hasta el pueblo situado en un valle a los pies de las torres.

Para este proyecto se espera un planteamiento del problema, en el cual se presentará su modelamiento en el que definiremos la información que conocemos, como la que se desconoce. De esta manera se podrá llegar a una resolución del problema de una forma óptima y más eficaz. Por otro lado también se busca resolver el sistema de ecuaciones formulado numéricamente para que, de esta manera, se pueda comprobar la prueba de concepto para la situación dada.

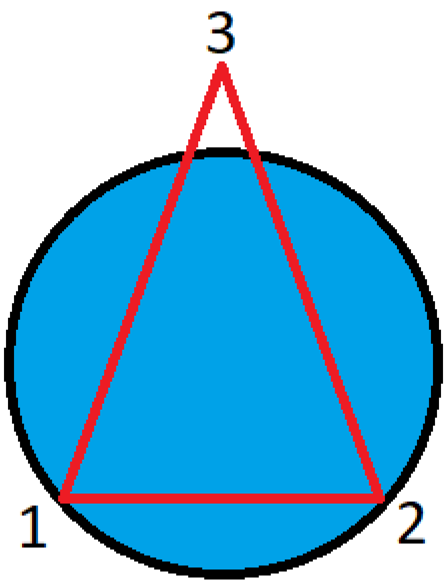
(2) Modelamiento del problema.

El problema planteado en esta situación toma origen en la novela "**Las Torres**", para ser específicos en el capítulo titulado "**La Caminata**", en el que la protagonista debe calcular la altura de las torres en respecto a su posición frente a un lago. En este capítulo se le explica que mediante la superposición de una cantidad de triángulos formada entre la base de estos ubicada en el lago y la punta como punto final en la altura de las torres, se puede conseguir efectuar un sistema de ecuaciones para que de esta manera, se logre calcular con exactitud la altura de estas.

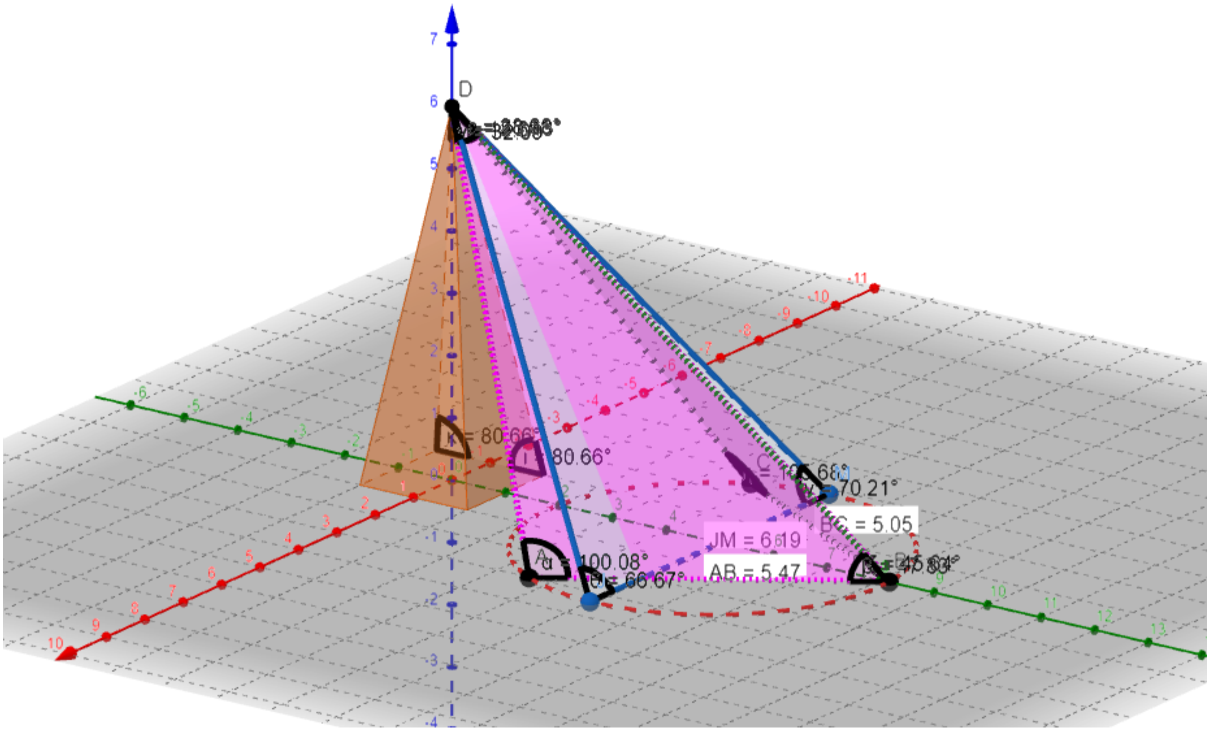
A lo largo del escrito, se puede observar que surgen diversos problemas, los cuales se tienen que tomar en consideración para continuar y de esta manera resolver el cálculo de una manera óptima. Los problemas que encontramos son los siguientes:

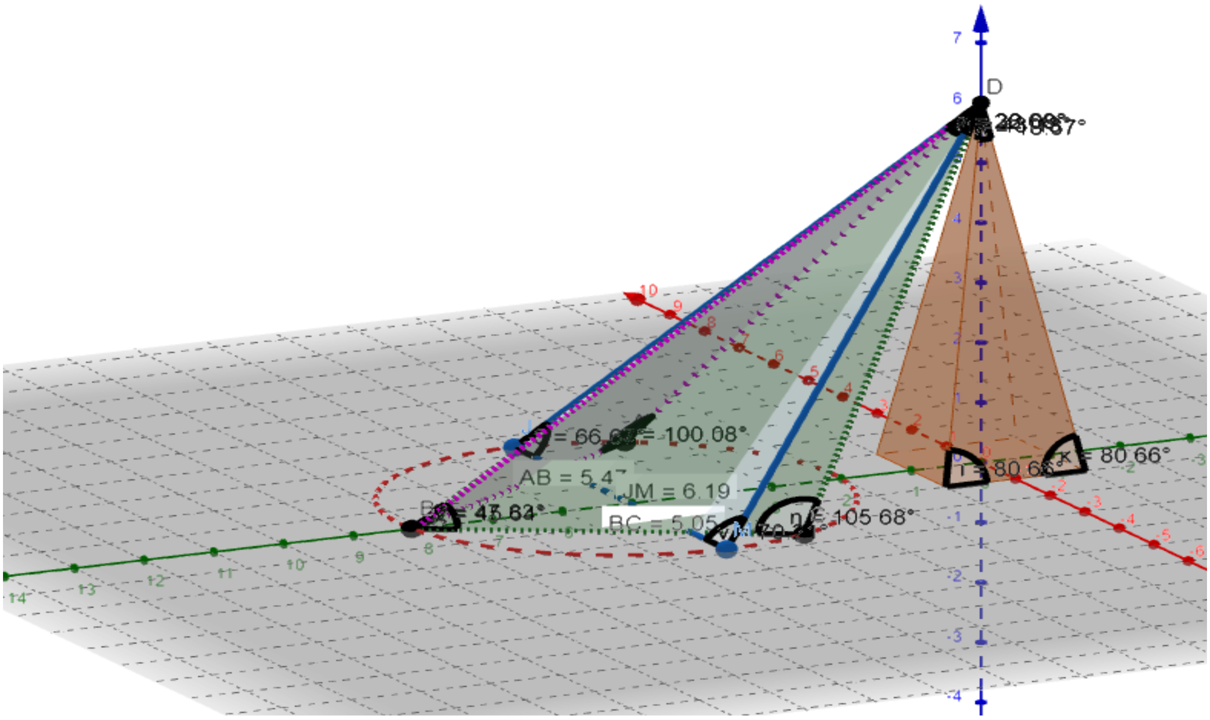
1. No se indican las características principales de la torre (tales como su altura, su inclinación, etc).
2. Se desconoce en qué posición está ubicada exactamente las torres.
3. No se sabe la longitud exacta del lago.

Para una mayor comprensión se creará la siguiente imagen, en donde "**1**" y "**2**" se definirán como los puntos que conforman la base de los triángulos y "**3**" corresponderá a la punta de las torres.



Como el problema está llevado en un ámbito de 3 dimensiones, nuestro plano tendrá que actualizarse, quedando de la siguiente manera:





Como podemos observar en nuestro modelo contamos con 3 triángulos, estos tienen un punto en común llamado D que es la punta de la torre, Cada triángulo tiene los siguientes nombres.

- El triángulo azul es DMJ.
- El triángulo verde es DBC.
- El triángulo rosa es DAB.

Con la información anterior nos corresponde crear el correspondiente sistema de ecuaciones, para esto debemos tomar en cuenta que:

- No conocemos el punto D.
- Conocemos los puntos A,B,C,D,J,M.
- Conocemos los ángulos en cada uno de estos puntos.
- Conocemos además los segmentos de la base de nuestros triángulos, estos son |BC|, |AB| y |MJ|.

Con la información anterior podemos establecer nuestro sistema de ecuaciones:

$$f_1(x_1, x_2, x_3) = (x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2 - |AD|^2$$

$$f_2(x_1, x_2, x_3) = (x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2 - |BD|^2$$

$$f_3(x_1, x_2, x_3) = (x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2 - |CD|^2$$

$$f_4(x_1, x_2, x_3) = (x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2 - |JD|^2$$

$$f_5(x_1, x_2, x_3) = (x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2 - |MD|^2$$

Donde se tiene que **la distancia** está definida por la siguiente fórmula:

$$d_{AD} = \frac{d(P_A, P_B) \cdot sen(\alpha\, P_B)}{sen(\alpha\, P_D)}$$

$$d_{BD} = \frac{d(P_B, P_A) \cdot sen(\alpha\, P_A)}{sen(\alpha\, P_D)}$$

$$d_{CD} = \frac{d(P_B, P_C) \cdot sen(\alpha\, P_C)}{sen(\alpha\, P_D)}$$

$$d_{JD} = \frac{d(P_M, P_J) \cdot sen(\alpha\, P_J)}{sen(\alpha\, P_D)}$$

$$d_{MD} = \frac{d(P_J, P_M) \cdot sen(\alpha\, P_M)}{sen(\alpha\, P_D)}$$

(3) Solución del problema:

Para realizar la correcta solución de este problema, tomaremos los valores que nos entrega geogebra por defecto con la coordenada de cada punto

	x1	x2	x3
Punto A	3.02	3.61	0
Punto B	0	8.16	0
Punto C	-2.51	3.78	0
Punto J	3.39	5.01	0
Punto M	-2.8	5.03	0

Ahora se debe tener en cuenta también los ángulos de la cada esquina de los triángulos formados, para estos ángulos usaremos los siguientes valores:

Ángulos para el triángulo ABD:
- Ángulo del Punto ABD = 47.83°
- Ángulo del Punto BAD = 100.08°
- Ángulo del Punto ADB = 32.09°

Ángulos para el triángulo BCD:
- Ángulo del punto DBC = 45.64°
- Ángulo del punto BCD = 105.68°
- Ángulo del punto BDC = 28.64°

Ángulos para el triángulo MJD:
- Ángulo del punto MJD = 66.67°
- Ángulo del punto JMD = 70.21°
- Ángulo del punto MDJ = 43.13°

Usando la función de geogebra que nos entrega la distancia entre dos puntos tendremos que |AB| = 5.47 ; |BC| = 5.05 ; |MJ| = 6.19

Luego y recopilando todos los datos anteriores establecidos se procederá a calcular distancia AD, distancia BD, distancia CD, distancia MD y distancia JD:

$$d_{AD} = \frac{5{,}47 * sen(47{,}83°)}{sen(32.09°)} =- 5.68$$

$$d_{BD} = \frac{5.47 * sen(100.08°)}{sen(32.09°)} =- 3.81$$

$$d_{CD} = \frac{5.05 * sen(105.68°)}{sen(28.64°)} = 12.79$$

$$d_{JD} = \frac{6.19 * sen(66.67°)}{sen(43.13°)} = 5.27$$

$$d_{MD} = \frac{6.19 * sen(70.21°)}{sen(43.13°)} =- 7.30$$

Finalizando y teniendo las |AD|, |BD|, |CD| y |MJ|, además de que se cuenta con la tabla de valores solo quedaría reemplazar en las funciones que se establecieron anteriormente, quedando el sistema de ecuaciones definido de la siguiente manera (Como no pueden existir longitudes negativas, los valores que tenemos arriba para las distancias se tomarán como valores positivos):

$$f_1(3.02; 3.61; 0) = (3.02 - y_1)^2 + (3.61 - y_2)^2 + (0 - y_3)^2 - 5.68^2$$

$$f_2(0; 8.16; 0) = (0 - y_1)^2 + (8.16 - y_2)^2 + (0 - y_3)^2 - 3.81^2$$

$$f_3(- 2.51; 3.78; 0) = (- 2.51 - y_1)^2 + (3.78 - y_2)^2 + (0 - y_3)^2 - 12.79^2$$

$$f_4(3.39; 5.01; 0) = (3.39 - y_1)^2 + (5.01 - y_2)^2 + (0 - y_3)^2 - 5.27^2$$

$$f_5(- 2.8; 5.03; 0) = (- 2.8 - y_1)^2 + (5.03 - y_2)^2 + (0 - y_3)^2 - 7.30^2$$

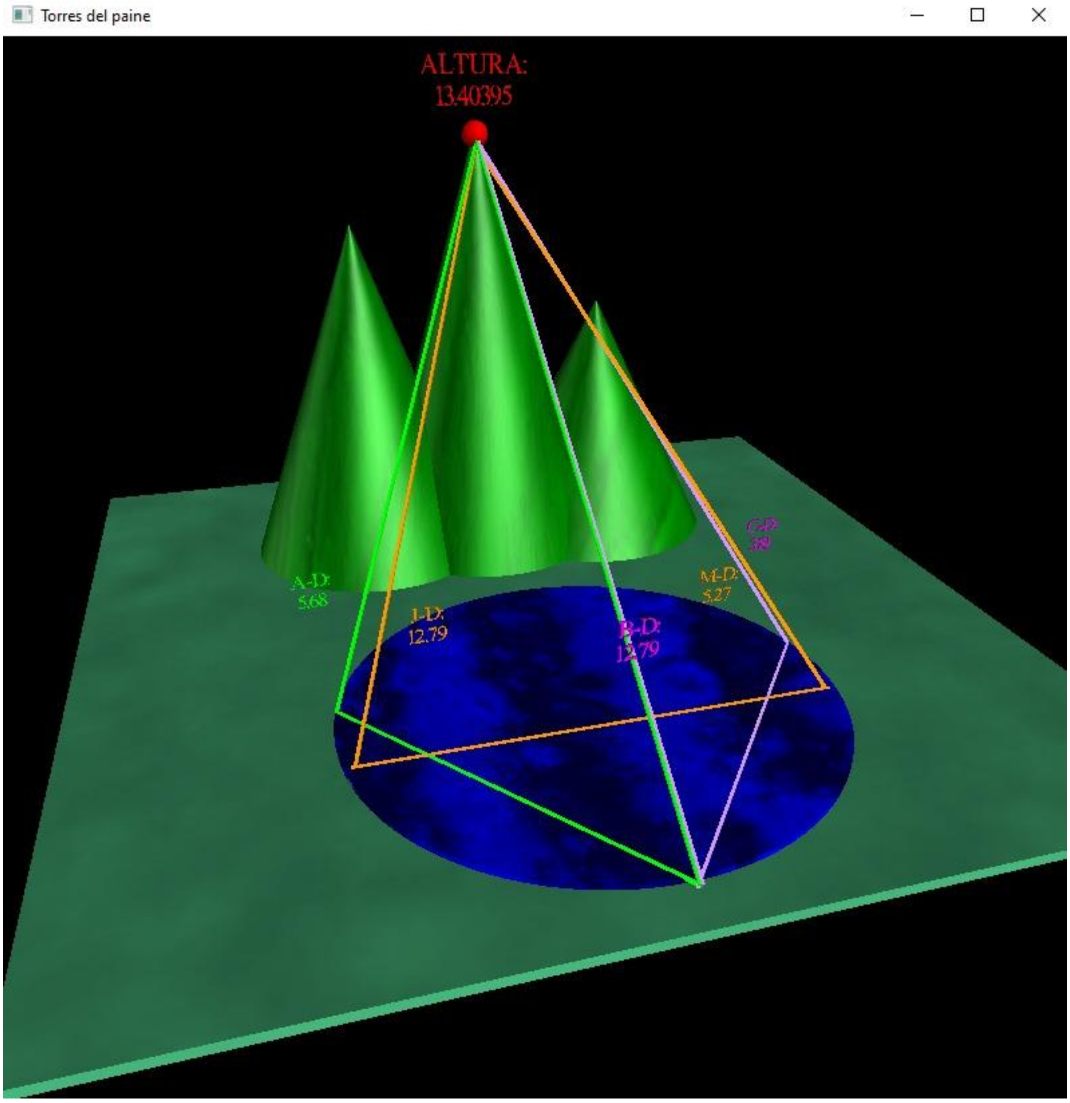
(4) Captura de pantalla de simulación realizada en python:

Se adjunta un screenshot de una simulación que se realizó mediante el uso de vpython + numpy en donde se aprecia la altura total que tendrá nuestra torre usando nuestro sistema de ecuaciones.

Código:

```python
#LIBRERIAS
from visual import * ; import numpy as np ; import numpy.linalg as la
#CALCULOS-NUMPY
A = np.array([  [3.02, 3.61, 1],
                [0, 8.16, 1],
                [-2.51, 3.78, 1],
                [3.39, 5.01, 1],
                [-2.8, 5.03, 1]
            ])
b = np.array([5.68,
            3.81,
            12.79,
            5.27,
            7.30
            ])
C = la.lstsq(A, b, rcond=None)[0]
pantalla = display(title='Torres del paine',x=0, y=0, width=800, height=600,center=(0,0,0))
base = box(length=100,width=100,height=1,color=color.rgb_to_hsv((0.07,0.6,0.09)),material=materials.rough)
laguna = cylinder(z=20,axis=vector(0,1,0),radius=25,height=1,color=color.blue,material=materials.marble)
torre1 = cone(pos=vector(0,0,-25),axis=vector(0,50,0),radius=15,color=color.green,material=materials.wood)
torre2 = cone(pos=vector(15,0,-25),axis=vector(0,30,0),radius=15,color=color.green,material=materials.wood)
torre3 = cone(pos=vector(-15,0,-25),axis=vector(0,40,0),radius=15,color=color.green,material=materials.wood)
#TRIANGULO1
t1v1 = box(x=12,y=25,z=-5,length=0.3,width=0.3,height=67,color=color.rgb_to_hsv(vector(1,0.4,1)),material=materials.emissive )
t1v1.rotate(angle=(radians(44)),axis=vector(-1,0,0))#arribaabajo
t1v1.rotate(angle=(radians(32)),axis=vector(0,1,0))#lados
t1v2 = box(x=0.1,y=25,z=11,length=0.3,width=0.3,height=85,color=color.rgb_to_hsv(vector(1,0.4,1)),material=materials.emissive )
t1v2.rotate(angle=-0.98,axis=vector(1,0,0))
t1v3 = box(x=12,y=1,z=30.5,length=0.3,width=0.3,height=39.5,color=color.rgb_to_hsv(vector(1,0.4,1)),material=materials.emissive )
t1v3.rotate(angle=(radians(90)),axis=vector(-1,0,0))#arribaabajo
t1v3.rotate(angle=(-radians(38)),axis=vector(0,1,0))#lados
#TRIANGULO2
t2v1 = box(x=-12,y=25,z=-5,length=0.3,width=0.3,height=67,color=color.green,material=materials.emissive )
t2v1.rotate(angle=(radians(44)),axis=vector(-1,0,0))#arribaabajo
t2v1.rotate(angle=(-radians(32)),axis=vector(0,1,0))#lados
t2v2 = box(x=-0.1,y=25,z=11,length=0.3,width=0.3,height=85,color=color.green,material=materials.emissive )
t2v2.rotate(angle=-0.98,axis=vector(1,0,0))
t2v3 = box(x=-12,y=1,z=30.5,length=0.3,width=0.3,height=39.5,color=color.green,material=materials.emissive )
t2v3.rotate(angle=(radians(90)),axis=vector(-1,0,0))#arribaabajo
t2v3.rotate(angle=(radians(38)),axis=vector(0,1,0))#lados
#TRIANGULO3
t3v1 = box(x=0,y=1,z=24,length=0.3,width=0.3,height=48,color=color.orange,material=materials.emissive )
t3v1.rotate(angle=(radians(90)),axis=vector(0,0,1))
t3v2 = box(x=-12,y=24,z=0,length=0.3,width=0.3,height=73,color=color.orange,material=materials.emissive )
t3v2.rotate(angle=(radians(49)),axis=vector(-1,0,0))#arribaabajo
t3v2.rotate(angle=(radians(26)),axis=vector(0,-1,0))#lados
t3v3 = box(x=12,y=24,z=0,length=0.3,width=0.3,height=73,color=color.orange,material=materials.emissive )
t3v3.rotate(angle=(radians(49)),axis=vector(-1,0,0))#arribaabajo
t3v3.rotate(angle=(-radians(26)),axis=vector(0,-1,0))#lados
puntoD = sphere(pos=vector(0,49,-25),radius=1.5,color=color.red)
AD = text(text='A-D:\n'+str(b[0]),pos=vector(-25,10,5),align='center', color=color.green,height=1.5,length=1.2)
BD = text(text='B-D:\n'+str(b[2]),pos=vector(0,15,31),align='center', color=color.magenta,height=1.5,length=1.2)
CD = text(text='C-D:\n'+str(b[1]),pos=vector(25,10,5),align='center', color=color.magenta,height=1.5,length=1.2)
JD = text(text='J-D:\n'+str(b[2]),pos=vector(-15,10,15),align='center', color=color.orange,height=1.5,length=1.2)
MD = text(text='M-D:\n'+str(b[3]),pos=vector(15,10,15),align='center', color=color.orange,height=1.5,length=1.2)
D = text(text='ALTURA:\n'+str(C[2])[0:8],pos=vector(0,55,-25),align='center', color=color.red,height=2,length=2)
```

Simulación:


Torres del paine
ALTURA:
13.40395
A-D:
5.68
J-D:
12.79
B-D:
12.79
M-D:
5.27

Bibliografía

Berres, S. (s. f.). *Las torres*.

¿Cómo se mide la altura de las montañas? (2016, 27 octubre). [Vídeo]. YouTube. https://www.youtube.com/watch?v=qPhsyk3FdF4

Linear algebra (numpy.linalg) — NumPy v1.20 Manual. (2021). numpy.org. https://numpy.org/doc/stable/reference/routines.linalg.html

numpy.linalg.lstsq — NumPy v1.20 Manual. (2021). numpy.org. https://numpy.org/doc/stable/reference/generated/numpy.linalg.lstsq.html

VPython Help. (2021). Vpython. https://www.glowscript.org/docs/VPythonDocs/index.html