



UNIVERSIDAD AUTÓNOMA DE NUEVO LÉON
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS



DIVISIÓN DE LAS CARAS DEL ICOSAEDRO
QUÍMICA COMPUTACIONAL PARA NANOTECNOLOGÍA

JOSÉ ABRAHAM MORALES VIDALES

1941505

NOMBRE DEL MAESTRO:

TLAHUICE FLORES ALFREDO

06 de marzo de 2020

Enunciado:

Hacer un código que divida las caras del icosaedro.

Reporte:

Lo que se hizo fue leer 2 posiciones del icosaedro (A y B) primeramente, en caso de que estos vértices formen una arista se lee una tercera posición (C). Después se pregunta si las tres posiciones forman una cara por medio de preguntar si A y C forman una arista y si las aristas B – A y C – A forman un ángulo de 60°. Si las condiciones se cumplen se inicia el proceso de división de caras.

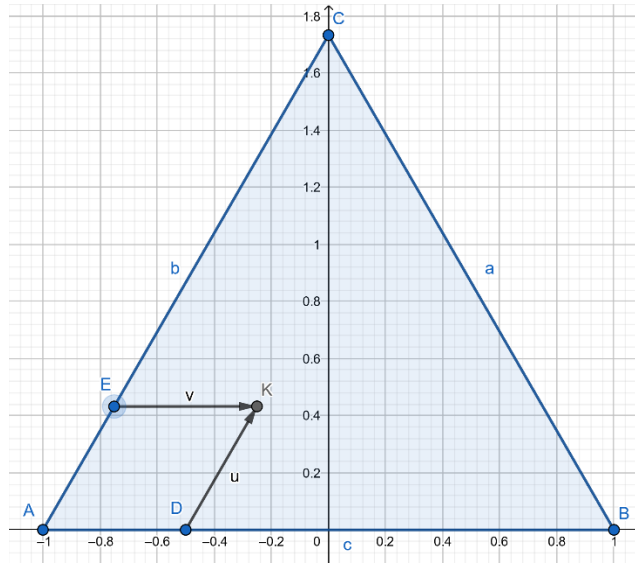


Figura 1. Representación de la división de una de las caras del icosaedro.

En la figura 1 se están dividiendo las aristas del icosaedro en n partes iguales, esto es fácil de hacer por medio de la siguiente fórmula:

$$\vec{D} = \vec{A} + \frac{\vec{B} - \vec{A}}{n} l \quad (1)$$

$$\vec{E} = \vec{A} + \frac{\vec{C} - \vec{A}}{n} l \quad (2)$$

Donde l va de 0 a n . A partir de la figura 1 se puede determinar que el punto K se encuentra sumando los vectores v y u partiendo del punto A . Entonces el vector v está en la misma dirección que vector $\vec{B} - \vec{A}$ y el vector u está en la misma dirección que el vector $\vec{C} - \vec{A}$. Entonces la fórmula para determinar el punto K juntando las fórmulas (1) y (2).

$$\vec{K} = \vec{A} + \frac{\vec{B} - \vec{A}}{n} l + \frac{\vec{C} - \vec{A}}{n} m \quad (3)$$

Antes de aplicar la fórmula (3) dentro de un código hay que añadir otra condición. Se debe añadir un contador que haga que la variable m no vaya de 0 a n , sino de 0 a $n - 1$ y en el

siguiente ciclo a $n-2$, $n-3$ y así. Esto es porque las rectas que marcan la división de las caras se van haciendo más pequeñas (véase la Figura 2).

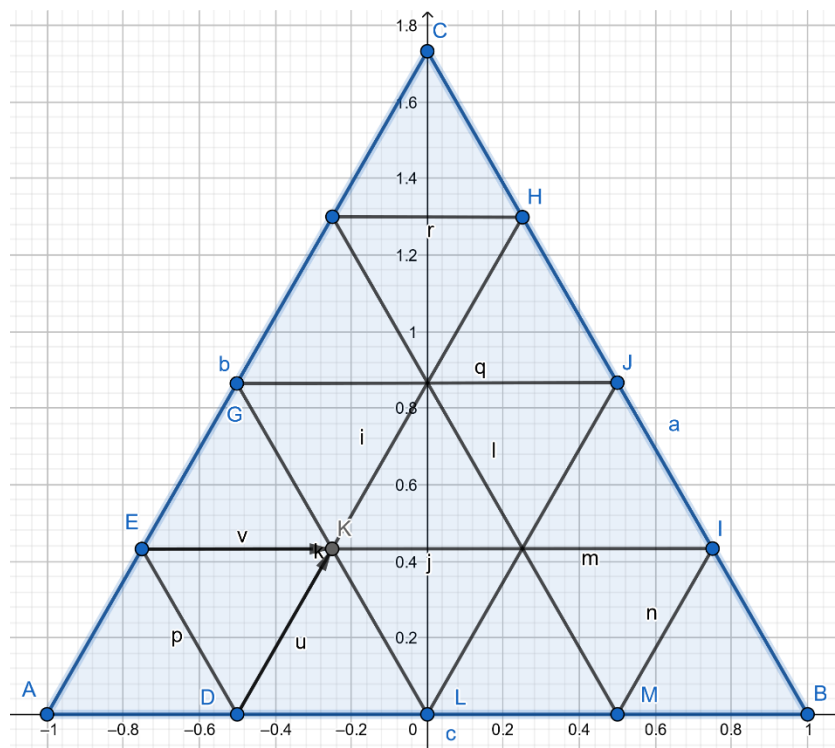


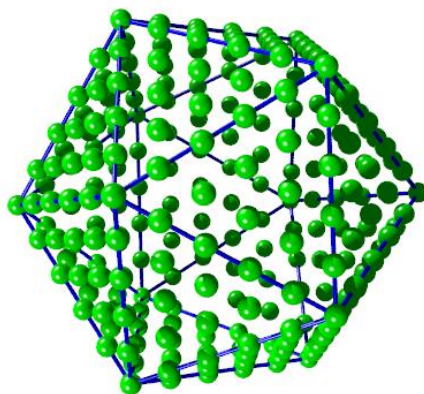
Figura 2. Cara triangular dividida en 16 partes iguales.

Entonces se implementó la fórmula iterativa (3) dentro de dos ciclos anidados tipo *for*.

```
#declare cont = 0;
#for(1,0,ndiv,1)
  #declare cont = cont+1;
  #for(m,0,ndiv-cont+1,1)
    #declare pos = ico[i] + (1/ndiv)*(ico[j]-ico[i]) + (m/ndiv)*(ico[k]-ico[i]);
    sphere[pos, 0.15 pigment{color Green} finish{phong 1}]
    write(Salida,"Au", " ", vstr(3,pos, " ",5,7), "\n")
```

Figura 3. Implementación de la fórmula (3) en el código de POV-Ray.

Con el código se obtuvieron los siguientes resultados:



Conclusión

Esta actividad fue interesante porque hay un gran variedad de nanopartículas que tienen una forma icosaedral y es importante saber cómo programar una molécula de este tipo para después poder hacer cálculos con estructuras similares. El código en sí, no tuvo mayor complicación, el utilizar dibujos ayuda bastante a entender cómo se van dividiendo los triángulos utilizando vectores. Se espera que este código pueda ser de utilidad en proyectos futuros en los que se hagan cálculos de optimización de estructuras.