

Código de modelación volcánica

Parte1, correr código:

¡Requerimientos!:

tener instalado:

- python3.8.x
- numpy
- matplotlib
- mpl_toolkits (incluido en matplotlib, pero es requerido en caso de tener instalación especial)

dentro de la carpeta, encontrás un archivo con nombre:

main.py

Este es el archivo principal de donde corre todo.

Para esta entrega, cree 3 presets que puedes disfrutar con solo correrlo

el primero es una grafica usando solamente un angulo d 45° y una magnitud de 141.4213562.

También tiene un radio de 10.

la segunda grafica se calcula un poco diferente, se van a crear 10 diferentes lineas, con una velocidad inicial en X yendo desde 50 a 200 m/s, y con una velocidad en Y desde 50 a 200 m/s. El radio cambia de 0.01 a 50.

Se podrán apreciar tanto las lineas con y sin resistencia al aire.

La ultima grafica sera una grafica en 3D. En esta grafica, se da la velocidad inicial de Y como un numero aleatorio de 50 a 100. Los valores de velocidad inicial en X y Z se dan con un numero aleatorio de -50 a 50. Estos valores aseguran que el angulo este desde los 45° a los 90° en todas las direcciones, y jamas llegue a ser uno menor.

Parte2, dataCreator.py:

dataCreator se divide en 3 partes:

- variables globales
- clase 'rock'
- clase 'plotter'

variables globales:

aquí es donde se definen las variables globales, estas incluyen:

- `delta_t = 0.02`
 - define el cambio de posiciones, en caso de tener una computadora no tan buena, es recomendado poner 0.1
- `rangeInTime = (0.0,100.0)`
 - define el rango de donde a donde construir los tiempos, por predeterminado, se crea una lista de 0 a 100 s
- `densityRho = 1.29`
 - densidad del aire, se encuentra en https://airdensityonline.com/track-results/Autodromo_Hermanos_Rodriguez/
- `gravity = 9.8`
 - gravedad de la tierra
- `dragCoefficient = 0.45`
 - coeficiente de arraste según la forma de una roca (esfera), https://www.engineeringtoolbox.com/drag-coefficient-d_627.html
- `time = np.array(time, dtype=np.float)`
 - la lista creada con base en tiempo, (todos sus puntos)
- `globalArrayLength = time.shape[0]`
 - El tamaño universal de todas las listas.

También se incluye una función para cambiar fácilmente todas las variables, esta se llama `updateBaseData`.

Rock:

Esta clase es donde se guarda toda la información fundamental de la piedra, incluye listas como:

`x` → posiciones de coordenadas en x
`y` → posiciones de coordenadas en y
`z` → posiciones de coordenadas en z

`x_dragForce` → fuerza de resistencia en un punto de x
`y_dragForce` → fuerza de resistencia en un punto de y
`z_dragForce` → fuerza de resistencia en un punto de z

`x_velocity` → velocidad en el vector x
`y_velocity` → velocidad en el vector y
`z_velocity` → velocidad en el vector z

`x_acceleration` → aceleración en el vector x
`y_acceleration` → aceleración en el vector y
`z_acceleration` → aceleración en el vector z

también guarda variables como:

`area` → área de efecto del arrastre

mass → masa del objeto

weight → el peso del objeto (masa * gravedad)

draffForceCoefficient → calculando el coeficiente de arrastre, con la formula
$$\frac{\rho AC}{2}$$

También, si calculateWithoutDrag es verdadero, se va a calcular la linea sin consideración de arrastre

CalculateNext (parte de rock):

Aquí es donde ocurren todos los cálculos matemáticos para la trayectoria.

El proceso es:

crear un loop que se repita por todos los objetos dentro del codigo

sacar la aceleración de x en el siguiente punto usando

$$\frac{-D}{m} \times \sqrt{V_x(t)^2 + V_y(t)^2 + V_z(t)^2} \times V_x(t)$$

sacar la velocidad de x en el siguiente punto $V_x(t - \Delta t) + a_x(t - \Delta t) \times \Delta t$

sacar el siguiente punto de x considerando lo anterior $V_x(t) \times \Delta t + \frac{a_x(t)}{2} \times \Delta t^2$

sacar la aceleración de y en el siguiente punto usando

$$-g - \frac{D}{m} \times \sqrt{V_x(t)^2 + V_y(t)^2 + V_z(t)^2} \times V_y(t)$$

sacar la velocidad de y en el siguiente punto $V_y(t - \Delta t) + a_y(t - \Delta t) \times \Delta t$

sacar el siguiente punto de y considerando lo anterior $V_y(t) \times \Delta t + \frac{a_y(t)}{2} \times \Delta t^2$

sacar la aceleración de z en el siguiente punto usando

$$\frac{-D}{m} \times \sqrt{V_x(t)^2 + V_y(t)^2 + V_z(t)^2} \times V_z(t)$$

sacar la velocidad de z en el siguiente punto $V_z(t - \Delta t) + a_z(t - \Delta t) \times \Delta t$

sacar el siguiente punto de z considerando lo anterior $V_z(t) \times \Delta t + \frac{a_z(t)}{2} \times \Delta t^2$

si y es menor a 0, el programa dejara de sacar mas valores para ahorrar tiempo.

Plotter:

tiene 2 funciones extras:

- plot
- getSpecialPoints

Plotter toma los rangos del disparo que gustes agregar, y la cantidad de disparos que gustas, al igual que la cantidad de disparos que serán calculados, al igual que rangos de tamaño del objeto.

Crea las lineas, y las calcula usando el método calculateNext de Rock.

Plot:

Crea una tabla con todas las lineas en 3D y lo representa en pantalla

GetSpecialPoints:

Crea una lista de los puntos especiales, como la distancia recorrida, altura máxima, etc.

Le da un formato lindo

Parte3 razón de elección de variables:

El rango de tiempo fue elegido de 0 a 100 s porque todo cabe dentro de ese rango.

El cambio de tiempo elegí 0.02 porque es un rango decente que mi computadora soporta, y no he tenido problemas con ello.

Elegí valores de velocidades de inicio que cayeran dentro de 100 a 200 m/s porque en promedio los proyectiles son expulsado entre esas velocidades, con la máxima velocidad posible (pero improbable) siendo 212.13 m/s, algo posible, pero improbable. De la misma forma, me asegure que el menor angulo de disparo sea 45 grados, ya que de esa forma me aseguro que lo que es expulsado supere en angulo mínimo de un volcán.

Adicionalmente, encontré que la piedra mas abundante del popocatepetl es la andesita, el cual tiene una densidad de entre 2.4 a 2.8 g/cm³ sabiendo esto, utilicé la formula $a \times d$, pero el problema es que mi medida esta en g y cm, entonces antes tuve que cambiar mi densidad a kg/m³ y para hacer esto, no hay mejor forma que multiplicarlo por 1000, lo cual me deja con una densidad de 2600 (porque tome el promedio de densidades de andesita). Lo que esto deja, es solamente sacar el área del objeto. ¿Recuerdas que al definir el valor de plotter, damos un rango de pesos?, bueno, para eso se pidió, lo podemos sacar lo la formula de área, y nuestra formula final seria :

$$\pi r^3 \times 2600$$

para sacar el área que esta en contacto con el aire, es fácil, si aplanamos nuestra esfera teórica, tendremos una buena aproximación de que área esta tocando al aire.

$$\pi r^2$$

y de ahí ya tenemos todas nuestras formulas.

Para sacar el tamaño de las piedras, se considera un numero aleatorio desde 0.01 m de radio, hasta los 1 m, creando un peso máximo de una roca de 8168.14kg, ya que la piedra volcánica mas grande expulsada por un volcán ha sido de 5 m, creo que es seguro creer que una piedra de 2 m de altitud es posible.

Finalmente, la densidad del aire es la densidad en la CDMX, planeo en el futuro cambiar la densidad a una formula según la altura, pero por el momento es una constante.

Mis planes a futuro son:

- Agregar soporte de GPU, para que la computacion de millones de lineas sea mas rapido
- Guardar lineas en una tabla para tener que computar solo 1 vez todo, y poderlas usarlas mas adelante
- Hacer un UI para que pueda ser mas interactivo

Referencias:

Current weather* for Autodromo Hermanos Rodriguez. (2020). Retirado Octubre 09, 2020, desde

https://airdensityonline.com/track-results/Autodromo_Hermanos_Rodriguez/

Developers, G. (2016). Densities of Igneous Rocks¶. Retirado Octubre 09, 2020, desde

https://gpg.geosci.xyz/content/physical_properties/tables/density_igneous_rocks.html

Drag Coefficient. (s.f.). Retirado Octubre 09, 2020, desde

https://www.engineeringtoolbox.com/drag-coefficient-d_627.html

INEGI. (2020). Relieve. Retirado Octubre 09, 2020, desde

<http://cuentame.inegi.org.mx/monografias/informacion/df/territorio/relieve.aspx?tema=me>

Laacher, V. (s.f.). 23. Lavabombe Strohn (Strohn Lava Bomb): A hot snowball. Retirado Octubre

09, 2020, desde <https://www.deutsche-vulkanstrasse.com/en/geology/lavabombe>

Volcán Popocatepetl. (s.f.). Retirado Octubre 09, 2020, desde

<https://www.geoenciclopedia.com/popocatepetl/>

referencias que había usado previamente, pero construí sobre ellas:

Global Volcanism Program: Current Eruptions. (2020). Retirado Septiembre 24, 2020, de https://volcano.si.edu/gvp_currenteruptions.cfm

Drag Coefficient. (s.f.). Retirado Septiembre 30, 2020, de

https://www.engineeringtoolbox.com/drag-coefficient-d_627.html

OpenStax. (s.f.). Physics. retirado Septiembre 30, 2020, de

<https://courses.lumenlearning.com/physics/chapter/5-2-drag-forces/>

Kalenda, P. (2002). Mathematical modelling of tektite formation. Retirado Septiembre 30, 2020, de

https://www.researchgate.net/publication/258997162_Mathematical_modelling_of_tektite_formation