



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE
CC3501 MODELACIÓN Y COMPUTACIÓN GRÁFICA

TAREA 3-C

TORMENTA EN ERUPCIÓN VOLCÁNICA

Por: Bryan Ortiz.
Profesor: Daniel Calderón
Fecha de Entrega: 10 de junio de 2019

Solución Propuesta

La tarea consiste en un programa realizado en *Python 3.7*, en el cual se modela el potencial eléctrico en el entorno de un volcán en erupción. Para ello se discretiza un espacio en 2D asignando valores a cada punto a partir de condiciones de borde especificadas y siguiendo la ecuación de Laplace $\nabla^2 V = 0$. Para la realización de la tarea se utilizan funciones de las librerías **SciPy**, **numpy** y **matplotlib**.

Para encontrar la solución, lo primero que se hizo fue simplificar el dominio, resolviendo primero para un rectángulo condiciones de borde tipo Dirichlet arriba y a ambos costados, y condiciones de tipo Neumann en el borde inferior. Este problema se resuelve a través de un sistema lineal matricial, del cual se obtiene el valor de cada punto en el dominio. Una vez realizado lo anterior, se procedió a limitar la condición de borde tipo Neumann de todo el borde inferior a sólo una parte de este, el centro (que podría imaginarse como el cráter pero desde una vista más elevada). Finalmente se eleva la condición del cráter y se añaden condiciones de borde tipo Dirichlet a sus costados.

Para el caso del interior del volcán, por motivos gráficos y prácticos, se le asigna a toda esa parte del dominio un potencial 0. Luego, esto se grafica de distintas formas mediante distintos scripts utilizando las funciones de las librerías **matplotlib**.

Por último, para encontrar un valor de E tal que $\Delta V = 3 \cdot 10^6 [V]$, se propone iterar el programa anterior con distintos valores de E y calculando ΔV en cada iteración hasta obtener la deseada. El valor obtenido es $E \sim 2300$. A continuación se muestra un gráfico de lo obtenido:

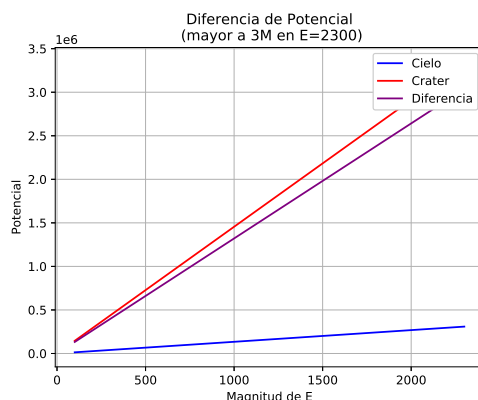


Figura 1: V en función de E

En la figura se aprecia un crecimiento lineal del potencial en función del campo eléctrico. Esto tiene mucho sentido pues mientras mayor sea el campo eléctrico, el potencial de un punto cercano al origen del campo con respecto a una referencia lejana será mucho mayor.

Dificultades Encontradas

Durante el desarrollo de esta tarea se encontraron diversas dificultades; las dos mas destacables fueron:

1.- Imponer condiciones de borde en una diagonal: Cuando se trata de un dominio específico con límites rectangulares, el imponer condiciones de borde es mucho más simple que en una diagonal. Hallar una solución para este caso tomó un poco más de tiempo del esperado ya que requiere pensar bien en la geometría del espacio. Esto se resuelve planteando la ecuación de la recta para cada diagonal, de modo que en los casos borde, algún punto adyacente al punto en cuestión se encuentra bajo la recta que define el volcán, mientras que el punto en sí se encuentra sobre esta recta.

2.- Graficar: El estar acostumbrado a graficar típicamente con programas como **Matlab** o **MS Excel**, ocasionan que uno raramente recurra a **Python** para estas tareas, por lo tanto no hay mucho conocimiento acerca de las librerías en cuestión y hubo que investigar un poco más al respecto para poder utilizar estas herramientas.

Cabe destacar que existieron también otras dificultades de por medio, como errores de código, adaptar un código ya existente a nuevos requerimientos, etc. Considerando lo anterior, el tiempo total invertido en la tarea es cercano a las 6 horas de trabajo continuo.

Instrucciones de Ejecución

El programa principal corresponde a **volcanic_eruption.py** el cual se ejecuta por consola como:

```
>> python volcanic_eruption.py h E solution.npy
```

donde **h** corresponde al largo del espaciado de la grilla, y **E** la magnitud del campo eléctrico en el cráter del volcán. Por otro lado **solution.npy** es el nombre del archivo donde se guardarán los datos de la solución obtenida, para un correcto funcionamiento de los demás *scripts* se recomienda utilizar el mismo nombre indicado en el ejemplo.

Los *scripts* de visualización como imagen, superficie y curvas de nivel, se ejecutan por consola respectivamente como:

```
>> python script_img.py
```

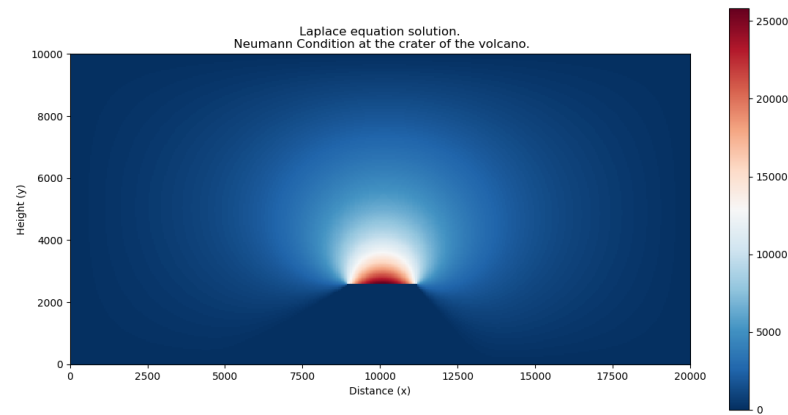
```
>> python script_levelCurves.py
```

```
>> python script_surface.py
```

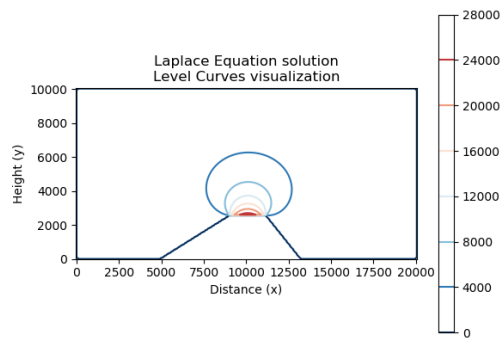
Nota: los scripts cargan la matriz con la información, la cual debe llamarse por defecto **solution.npy**, si se guarda con otro nombre en el programa principal, habrá que modificar el *script* para que cargue los datos de dicha matriz.

Resultados

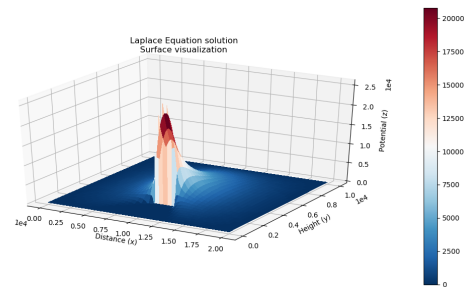
A continuación se muestran los 3 gráficos generados a partir del problema.



(a) Visualización como imagen.



(b) Curvas de nivel



(c) Vista como superficie

Figura 2: Capturas de las visualizaciones obtenidas.