

# Reporte de Tarea 3

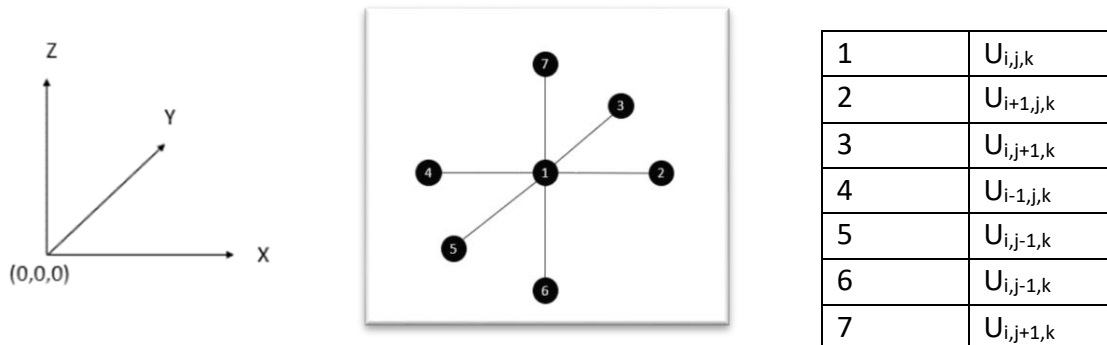
Este reporte describe la arquitectura realizada para solucionar la tarea 3 del curso CC3501- Modelamiento y computación grafica para ingenieros. En esta ocasión la opción seleccionada fue la opción b, la cual consiste en resolver la ecuación del calor para el agua contenida en un acuario y luego hacer una aplicación para visualizar el acuario con peces. Esta tarea se desarrollo con lenguaje de Python.

## Solución propuesta

La Tarea se divide en dos partes , la primera en la resolución de la EDP y la segunda en la visualización de un acuario usando la solución obtenida de la EDP

### Parte 1: Solución de la EDP

Lo primero que se realizo fue discretizar el dominio 3D usando un stencil de 7 puntos , donde cada punto  $(i,j,k)$  se representa con un  $U_{i,j,k}$  . A continuación, se muestra un ejemplo de esta discretización .



Al tener todo el cubo de dimensiones  $W$  ,  $L$  y  $H$  , discretizado de esta forma se procedió a identificar y a resolver todos los posibles casos que podría haber en este cubo los cuales resultaron ser 29 ,contando a los calefactores. Luego se procedió a resolver el sistema de ecuaciones planteados , el cual nos entregaba un vector al cual se procede a transformarlo nuevamente al dominio 3D original para finalmente plotear la matriz obtenida para que el usuario pueda observar las distintas temperaturas que tendría el acuario bajo las condiciones que se entrego .

Esta parte de la tarea esta adaptado para recibir un json con toda la información requerida para crear esta ilustración , el único parámetro que se encuentra fijo es el valor

del  $h$  , el cual es la distancia entre cada punto , este parámetro tiene un valor de 0.2 para tener una mejor visualización, pero la desventaja es que se tarda un poco en cargar .

## Parte 2: visualización del acuario

Esta parte del problema consistía en la visualización de un acuario con tres tipos de peces distintos .

Los tres modelos de peces fueron creados usando grafos de escena , los cuales se dividían en tres grandes partes , el cuerpo , los ojos y las aletas. Para los cuerpos se usó esfera con iluminación a la cual se le aplicó diferentes transformaciones para obtener las distintas formas de los cuerpos de cada pez , para las distintas aletas se usaron dos curvas de Bézier unidas y luego transformadas para darle distintas formas para cada modelo , por último, se creó los ojos usando círculos negros alargados .

La estructura de la pecera al igual que los peces se creó usando un grafo de escena ,el cual se compuso de 12 rectángulos alargados para simular el marco , 4 cuadrados azules para simular los vidrios y dos imágenes , una para simular el suelo del acuario y la otra para simular la superficie del agua de la pecera.

Para una mejor visualización de lo que se encuentre dentro del acuario, el usuario tendrá control sobre la cámara de la visualización , si presiona la tecla DERECHA ,la cámara se moverá hacia la derecha rotando sobre la pecera , si selecciona la tecla IZQUIERDA se moverá hacia esta dirección rotando sobre la pecera . Para realizar Zoom el usuario solamente tendrá que presionar la tecla ARRIBA para acercarse y para alejarse debe presionar la tecla ABAJO del teclado. Además de estos comandos , se agregan dos adicionales , el primero de ellos consiste en que al presionar la tecla ENTER del ordenador, los vidrios como la superficie del agua desaparecerán , dejando solamente los marcos del acuario y para volver a colocarlos solo se debe apretar ENTER nuevamente y el segundo comando es que al apretar CTRL izquierdo , la cámara bajara como si el usuario se agachara y si se presiona nuevamente la misma tecla la cámara volverá a subir .

La aplicación recibirá un archivo json , la cual especificara la cantidad de peces de cada modelo  $n_A$ ,  $n_B$  y  $n_C$  , pero además entregara la temperatura a la cual cada pez le gusta vivir ,  $T_A$ ,  $T_B$  y  $T_C$  . Por ende, como un objetivo de esta aplicación es observar los voxels donde cada pez vivirá, se ha habilitado 3 comandos para activar y desactivar la visualización de estos voxels.

- Al presionar la tecla A , los voxels del pez A se resaltará de color de verde
- Al presionar la tecla B , los voxels del pez B se resaltará de color de rojo
- Al presionar la tecla C , los voxels del pez C se resaltará de color de naranja

El archivo json también traerá el nombre del archivo que posee la matriz , la aplicación usará esta matriz para crear la pecera, pero siempre lo llevará a un tamaño estándar de 10x19x11.

Los peces se distribuirán aleatoriamente dentro de cada una de sus zonas ,de igual modo se orientarán de forma y su aleteo aleatorio

Adicionalmente se han agregado paredes con imágenes de acuarios , un suelo de arena y una mesa donde se coloca encima el acuario , en forma de decoración .

## Instrucciones de ejecución

Las dos partes de la tarea se pueden ejecutar en el terminal de anaconda . Para visualizar la primera parte se debe colocar el comando Python aquarium-solver.py archivo.json, donde el archivo.json es un archivo a elección , en el siguiente ejemplo se puede ver lo mencionado usando como archivo.json el ejemplo entregado por el profe.

```
(base) C:\Users\karen>conda activate python-cg
(python-cg) C:\Users\karen>cd C:\Users\karen\OneDrive\Escritorio\GitLab\lavados-jilbert_karen-constanza\Tarea3a
(python-cg) C:\Users\karen\OneDrive\Escritorio\GitLab\lavados-jilbert_karen-constanza\Tarea3a>python aquarium-solver.py
problem-setup.json
C:\Users\karen\anaconda3\envs\python-cg\lib\site-packages\scipy\sparse\linalg\dsolve\linsolve.py:137: SparseEfficiencyWarning:
  warn('spsolve requires A be CSC or CSR matrix format',
```

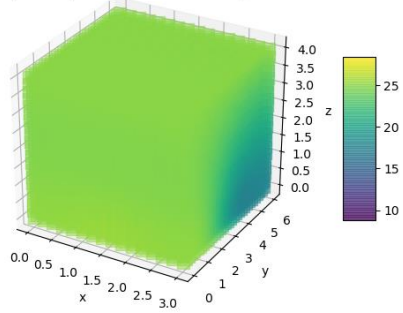
Para la segunda parte es análoga pero el comando es el siguiente Python Aquarium-view.py archivo.json , donde lo ultimo es un archivo a elección del usuario . A continuación, se puede ver la ejecución de este comando con el json entregado como ejemplo por el profesor .

```
(base) C:\Users\karen>conda activate python-cg
(python-cg) C:\Users\karen>cd C:\Users\karen\OneDrive\Escritorio\GitLab\lavados-jilbert_karen-constanza\Tarea3a
(python-cg) C:\Users\karen\OneDrive\Escritorio\GitLab\lavados-jilbert_karen-constanza\Tarea3a>python aquarium-view.py vi
ew-setup.json
```

## Resultados

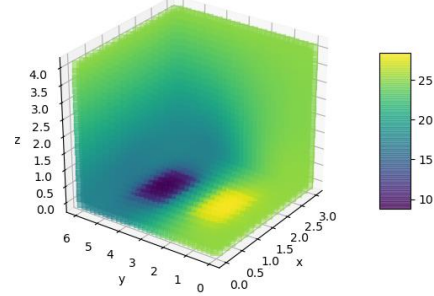
A continuación, se presentarán algunos screenshots de la tarea realizada , en las cuales se puede ver la solución de la EDP y también algunas imágenes del acuario final en distintos casos.

Laplace equation solution from aquarium

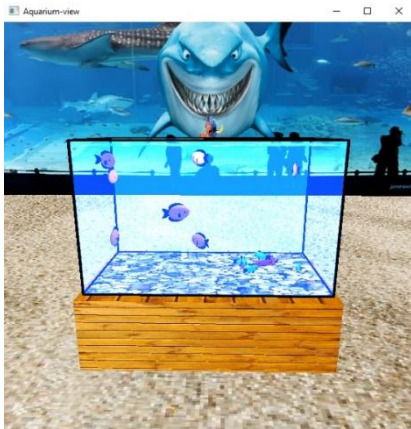


*Solución vista desde el frente*

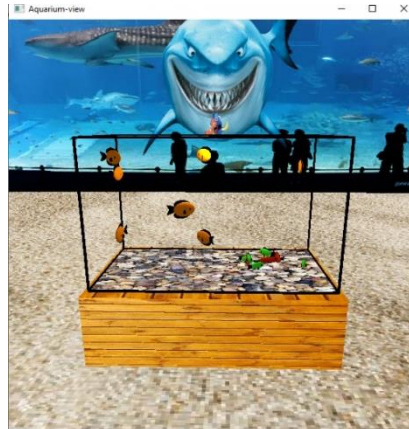
Laplace equation solution from aquarium



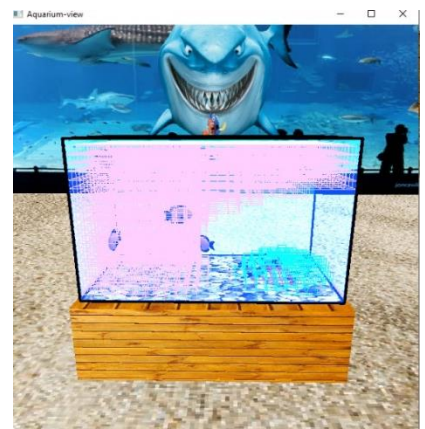
*Solución vista desde atrás*



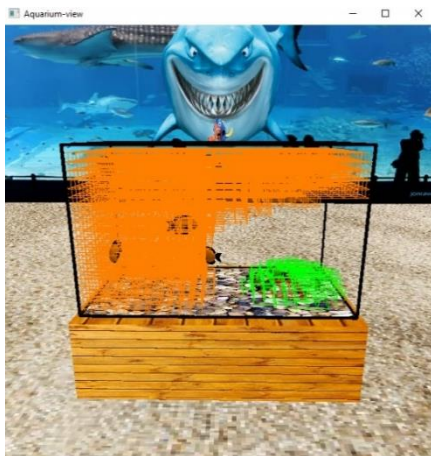
*visualización del acuario con vidrios y superficie de agua*



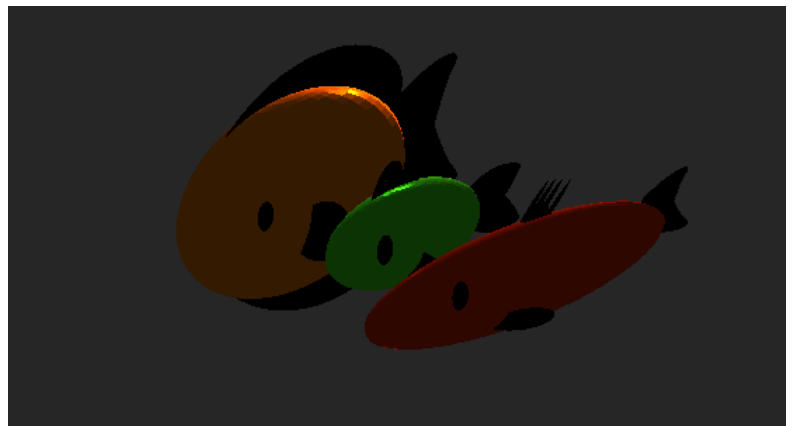
*visualización del acuario sin vidrios y sin superficie de agua*



*visualización del acuario con voxels de la distribución de los peces usando el vidrio y la superficie del agua*



*visualización del acuario con voxels de la distribución de los peces sin el vidrio ni con la superficie del agua*



*Los tres modelos de los peces*

## Bibliografía:

The Walt Disney Company, 2003. *Personajes De La Película Nemo*.