International Institute of Information Technology
Bangalore

SUBJECT Basic Computational Topology
COURSECODE SM 402

COURSE PROJECT

Ganga Sagar (IMT2020134)

Vamshi guggulla (IMT2020080)

Sougandh (IMT2020120)

B Sathiya Naraayanan (IMT2020534)

1 CONCEPT

- A vertex (x,y,z) is present in super level set then f(x,y,z)>=c for given c value
- To calculate β_0 we need $dim(Ker(\partial_0))$ and $dim(img(\partial_1))$. $dim(ker(\partial_0)) = dim(C_0(K))$ and $dim(img(\partial_1))$ is $rank(\partial_1)$. $\beta_0 = dim(ker(\partial_0)) - dim(img(\partial_1)) = dim(C_0(K)) - rank(\partial_1)$

2 ALGORITHM AND IMPLEMENTATION

We will visit every vertex(x,y,z) in the grid and check whether the vertex is present in super level set by comparing A[x][y][z] with c value, if the vertex (x,y,z) is present in super level set then we will check three neighbouring vertices (x+1,y,z),(x,y+1,z),(x,y,z+1) whether these vertices present in super level set (by comparing vertex mapping value with c value)if any these vertices present in superset then there will be edge from (x,y,z) to these vertices. Compute boundary matrix from these edges using ∂_1 operator.

$\partial_1(v_iv_j) = v_j - v_i$

$dim(img(\partial_1))$ is same as rank of boundary matrix and

$dim(ker(\partial_0))$ is $dim(C_0(k))$(this is no vertices present in super level set)

so compute using this formulae,

$\beta_0 = dim(ker(\partial_0)) - dim(img(\partial_1)) = dim(C_0(K)) - rank(\partial_1)$

3 CODE

```
import numpy as np

filename = input("Enter a input file name:")
#"marschner_lobb_41x41x41_uint8.raw"
dim_x=int(input("Enter the dimension of x:"))
dim_y=int(input("Enter the dimension of y:"))
dim_z=int(input("Enter the dimension of z:"))
A = np.fromfile(filename, dtype='uint8', sep="")
A = A.reshape((dim_x, dim_y, dim_z))
c=int(input("Enter the c value: "))
v=dim_x*dim_y*dim_z
e=0
a=0
max=0
for i in range(0,dim_x):
    for j in range(0,dim_y):
        for k in range(0,dim_z):
            if(A[i][j][k]>=c):
                #if(max<A[i][j][k]):
                    #max=A[i][j][k]
                a+=1
                if(i+1 < dim_x):
                    if(A[i+1][j][k]>=c):
                        e+=1
                if(j+1 < dim_y):
                    if(A[i][j+1][k]>=c):
                        e+=1
                if(k+1<dim_z):
                    if(A[i][j][k+1]>=c):
                        e+=1
```

```
print("\n")
#print('max:',max)
print('no of vertices: ',a)
print('no of edges: ',e)

t=0
arr=[ [0] * int(e) for i in range(int(v))]
for i in range(0,dim_x):
    for j in range(0,dim_y):
        for k in range(0,dim_z):
            if(A[i][j][k]>=c):
                if(i+1 < dim_x):
                    if(A[i+1][j][k]>=c):
                        x=i+j*dim_x+k*dim_x*dim_y
                        y=(i+1)+j*dim_x+k*dim_x*dim_y
                        arr[x][t]=-1
                        arr[y][t]=1
                        t+=1
                if(j+1 < dim_y):
                    if(A[i][j+1][k]>=c):
                        x=i+(j)*dim_x+k*dim_x*dim_y
                        y=(i)+(j+1)*dim_x+k*dim_x*dim_y
                        arr[x][t]=-1
                        arr[y][t]=1
                        t+=1
                if(k+1<dim_z):
                    if(A[i][j][k+1]>=c):
                        x=i+(j)*dim_x+k*dim_x*dim_y
                        y=(i)+(j)*dim_x+(k+1)*dim_x*dim_y
                        arr[x][t]=-1
                        arr[y][t]=1
                        t+=1

rank=np.linalg.matrix_rank(arr)
print("Rank: " + str(rank))
betti_0 = a - rank

print("+---------+")
print('|\N{GREEK SMALL LETTER BETA}\N{SUBSCRIPT ZERO} =',betti_0,'|')
print("+---------+")
```

4 OUTPUT

```
sagar@sagar:~/Desktop/topology_11$ python3 topo.py
Enter a input file name:hydrogenAtom.raw
Enter the dimension of x:128
Enter the dimension of y:128
Enter the dimension of z:128
Enter the c value: 100


no of vertices:  19
no of edges:  30
Rank: 18
+---------+
|β₀ = 1 |
+---------+
sagar@sagar:~/Desktop/topology_11$
```

```
sagar@sagar:~/Desktop/topology_11$ python3 topo1.py
Enter a input file name:Tangle_20.txt
Enter the dimension of x:20
Enter the dimension of y:20
Enter the dimension of z:20
Enter the c value: 2000


max: 4687.16
no of vertices:  872
no of edges:  1888
Rank: 868
+---------+
|β₀ = 4 |
+---------+

Execution time: 3.5692696571350098
sagar@sagar:~/Desktop/topology_11$
```

```
sagar@sagar:~/Desktop/topology_11$ python3 topo.py
Enter a input file name:marschner_lobb_41x41x41_uint8.raw
Enter the dimension of x:41
Enter the dimension of y:41
Enter the dimension of z:41
Enter the c value: 254


max: 255
no of vertices:  242
no of edges:  153
Rank: 126
+---------+
|β₀ = 116 |
+---------+

Execution time: 1.3267788887023926
sagar@sagar:~/Desktop/topology_11$
```