International Institute of Information Technology Bangalore

Basic Computational Topology
SM 402

BCT Implementation Assignment

April 19, 2022

Group 1
Karanjit Saha (IMT2020003)
Arya Kondawar (IMT2020084)
Paras Vekariya (IMT2020547)
Anshul Madurwar(IMT2020554)

Problem Statement

Given any input simplicial complex (up to 3-dimensional), compute β_0 using the boundary matrix method.

Algorithm

We have used the formula given below in our code to calculate β_0:

$$\beta_0 = dim(H_0(K)) = dim(C_0(K)) - dim(Im(\partial_1)) \tag{1}$$

In our program we take vertices,edges and faces(it is redundant) as input. We then create a matrix corresponding to the linear transformation ∂_0 and them compute $dim(Im(\partial_1))$, i.e. the $rank(\partial_1)$.

As we also know $C_0(K)$ is the vector space of 0-chains $\implies$ $dim(C_0(K))$= number of vertices.

By using all the above facts we can easily calculate β_0 for a simplicial complex.

Implementation Steps

1. First we ask for input from user. From here we get the number of vertices in the input simplicial complex, i.e. $dim(C_0(K))$.
2. In the next step we create the matrix corresponding to ∂_0 using the edges of the input.
3. In the last and the final step we calculate β_0 using equation 1.

Steps to run the code

1. Open the terminal.
2. Enter the commmand "pip3 install sympy".
3. Enter the command "python3 topo.py".
4. Enter the number of vertices, edges and faces respectively.
5. Enter the vertices.
6. Enter the edges.
7. Press Enter to get the final result.

NOTE:- Here we have not taken faces as input since faces do not play any role in calculation of β_0.

GitHub Link

Please visit this for an example based explanation.
https://github.com/KaranjitSaha/TOPOLOGY-PROJECT

Demo Results

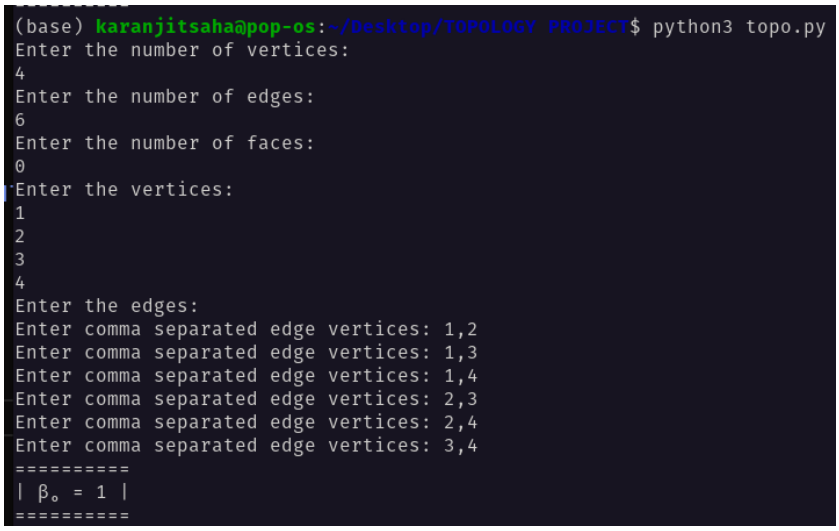

Figure 1: Testcase 1:- a tetrahedron

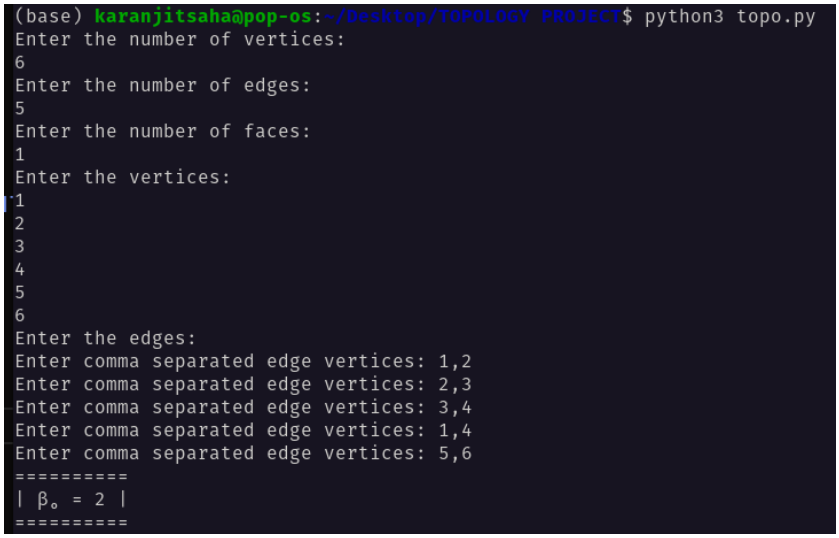

Figure 2: Testcase 2:- a square and a line segment

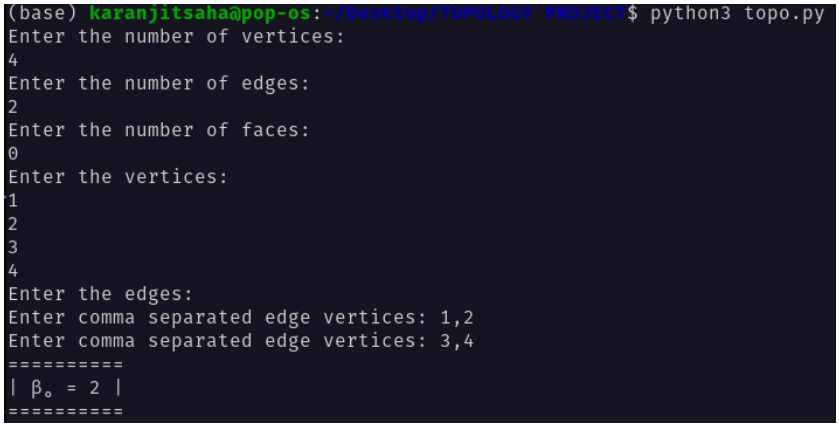

Figure 3: Testcase 3:- two line segments

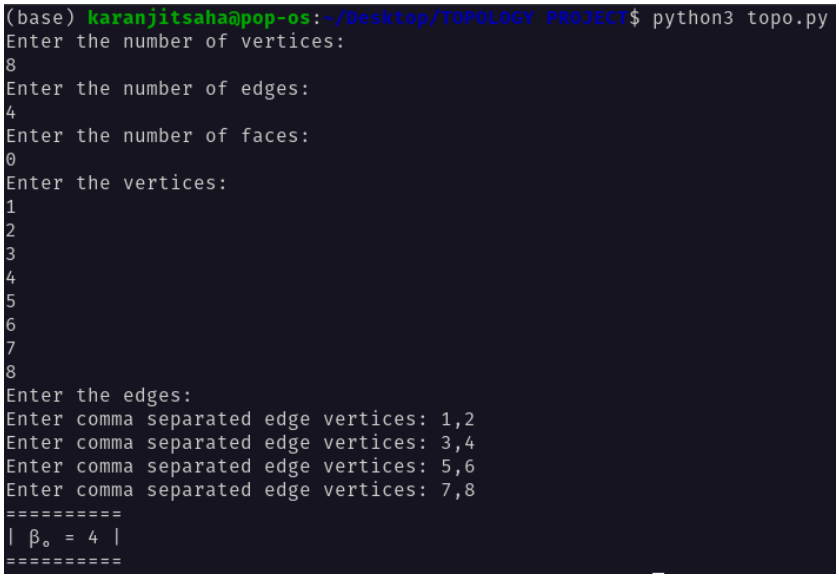

Figure 4: Testcase 4:- four line segments

Python Code

Python code for calculating β_0

```
import sympy as sym

print("Enter the number of vertices: ")
num_ver = int(input())
print("Enter the number of edges: ")
num_edg = int(input())
print("Enter the number of faces: ")
num_fac = int(input())

ver = []
edges = []
faces =[]

print("Enter the vertices: ")
for i in range(0, num_ver):
    j = int(input())
    ver.append(j)

print("Enter the edges: ")
for i in range(0, num_edg):

    edge_1 = list(map(int, input("Enter comma separated edge vertices: ").split(",")))
    edges.append(edge_1)

# print(edges)

img_space = []

for i in range(0, num_ver):
    temp1=[]
    for j in range(0, num_edg):
        if(edges[j][0] == (i+1)):
            temp1.append(-1)

        elif(edges[j][1] == i+1):
            temp1.append(1)

        else:
            temp1.append(0)

    img_space.append(temp1)
```

```
42
43  # print()
44  # print(img_space)
45  # for i in range(num_ver):
46  #     for j in range(num_edg):
47  #         if(img_space[i][j] ≥ 0):
48  #             print("  ", img_space[i][j], end = " ")
49  #         else:
50  #             print(" ",img_space[i][j], end = " ")
51  #     print()
52
53  rank = sym.Matrix(img_space).rank()
54
55  betti_0 = num_ver - rank
56
57  print("==========")
58  print('| \N{GREEK SMALL LETTER BETA}\N{SUBSCRIPT ZERO} =',betti_0,'|')
59  print("==========")
```

References

1. http://web.cse.ohio-state.edu/~wang.1016/courses/788/Lecs/lec7-qichao.pdf
2. https://www.math.rug.nl/~gert/documents/2006/rv_ecg_book_chapter7.pdf
3. https://jeremykun.com/2013/04/10/computing-homology/
4. https://jeremykun.com/2014/01/23/fixing-bugs-in-computing-homology/
5. https://en.wikipedia.org/wiki/Quotient_space_(linear_algebra
6. https://en.wikipedia.org/wiki/Rank%E2%80%93nullity_theorem