

**SM 402: Basic Computational Topology**  
**Implementation Project**

By

IMT2020042 Gousepeer Arella

IMT2020085 Harshadeep Donapati

IMT2020082 Prudhvi Nath Reddy

IMT2020522 Prasanth Reddy Lomada

# Contents

<b>1</b>	<b>Problem Statement</b>	<b>1</b>
<b>2</b>	<b>Algorithm</b>	<b>1</b>
<b>3</b>	<b>Steps to run the code</b>	<b>1</b>
<b>4</b>	<b>Code</b>	<b>1</b>
4.1	Github Link . . . . .	4
<b>5</b>	<b>Note</b>	<b>4</b>
<b>6</b>	<b>Screenshots</b>	<b>5</b>
<b>7</b>	<b>References</b>	<b>6</b>

# 1 Problem Statement

Problem Number : 4

Given any input simplicial complex (up to 3 -dimensional), compute  $\beta_3$  using the boundary matrix method.

# 2 Algorithm

Formula used to calculate  $\beta_3$ :

$$\beta_3 = \dim(Ker(\partial_3)) - \dim(Im(\partial_4)) \quad (1)$$

In our python file, we take the filename (.gts file) as an input from the user and process the file.

We know that  $\partial_3$  and  $\partial_4$  are defined as:

$$\partial_3 : C_3(K) \rightarrow C_2(K)$$

$$\partial_4 : C_4(K) \rightarrow C_3(K)$$

$\partial_k$  boundary matrix row labels are basis for  $C_{k-1}(X)$  and column labels are the basis for  $C_k(X)$ .

Kernel of a matrix is nullspace of the matrix and image is column space of a matrix.

# 3 Steps to run the code

Make sure you have `python3` and `numpy` and `plotly` installed. Use the command `pip3 install numpy plotly` to install them.

For best results for tetrahedron, use `tetra.py`. Use `Tetrahedron.gts` as your input file.

For Torus and TangleCube, use the command `python3 only.py` to run it. Then enter the name of the .gts file.

# 4 Code

`only.py`

```
import time
import numpy as np
import os
```

```

import plotly.graph_objects as go

begin = time.time()

def convert_float(nested):
    return [[float(x) for x in lst] for lst in nested]

def convert_int(nested):
    return [[int(x) for x in lst] for lst in nested]

def open_file(file_name):
    try:
        with open(file_name, 'r') as f:
            line = f.readline()
            lst = line.split(' ')
            no_vertices = int(lst[0])
            no_edges = int(lst[1])
            no_faces = int(lst[2])
            print("Number of vertices:", no_vertices)
            print("Number of edges:", no_edges)
            print("Number of faces:", no_faces)

            vertices = []
            edges = []
            faces = []

            for i in range(no_vertices):
                line = f.readline()
                vertices.append(list(line.replace('\n', ' ').split(' ')))

```

```

        for i in range(no_edges):
            line = f.readline()
            edges.append(list(line.replace('\n', '').split(' ')))

        for i in range(no_faces):
            line = f.readline()
            faces.append(list(line.replace('\n', '').split(' ')))

except OSError as e:
    print(e.strerror)

print("vertices:", convert_float(vertices))
print("edges:", convert_int(edges))
print("faces:", convert_int(faces))

try:
    os.mknod('vertices.txt')

except FileExistsError as e:
    print(e.strerror)

with open("vertices.txt", "a") as f:
    for lst in vertices:
        f.write(lst[0] + ' ' + lst[1] + ' ' + lst[2] + '\n')

pts = np.loadtxt(np.DataSource().open(
    'vertices.txt'))

x, y, z = pts.T

fig = go.Figure(

```

```

        data=[go.Mesh3d(x=x, y=y, z=z, color='cyan', opacity=0.5)])

fig.show()

os.remove('vertices.txt')

if __name__ == '__main__':
    file_name = input("Enter the name of the file: ")
    open_file(file_name)
    end = time.time()
    print("Time elapsed:", end - begin, 's')

```

## 4.1 Github Link

You can also visit the below link.

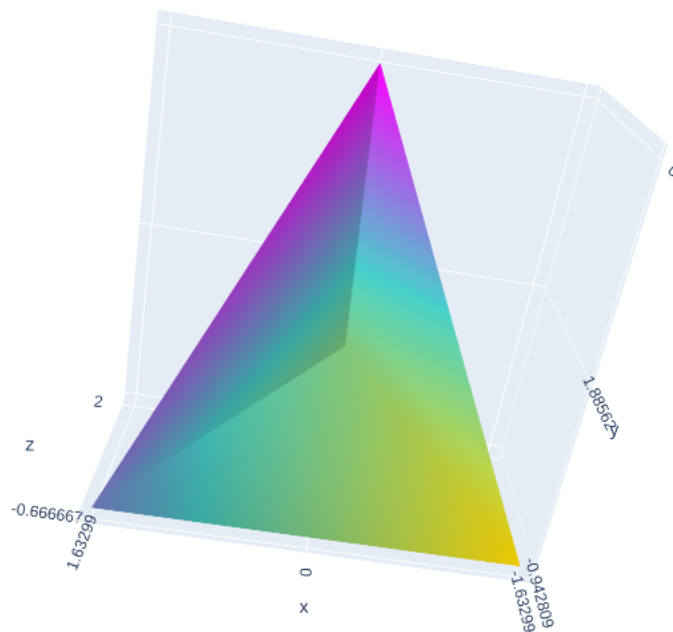
<https://github.com/harsha-deep/BCTImplementationProject>

## 5 Note

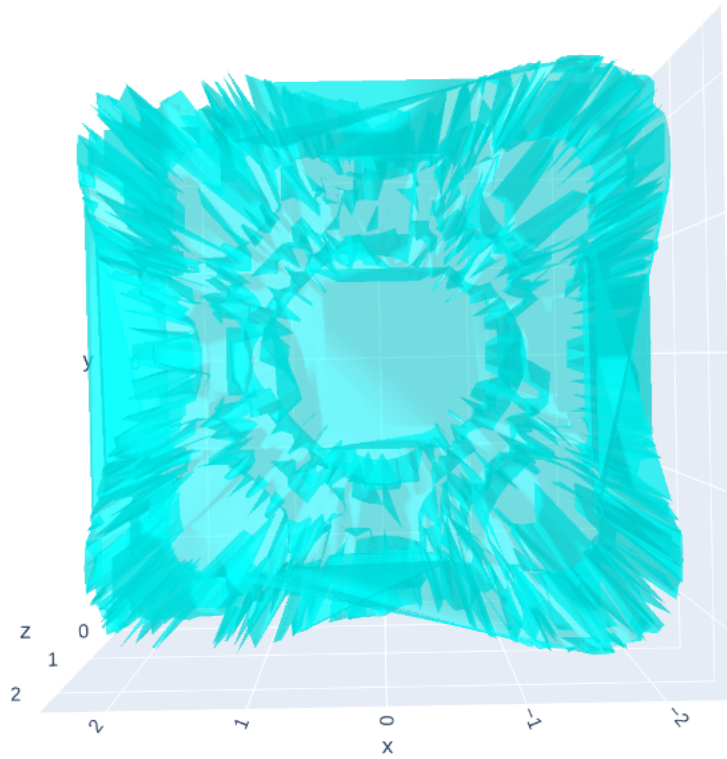
The code is not complete. We are unable to calculate  $\partial_3$  and  $\partial_4$ .

## 6 Screenshots

```
Terminal - harsha@xfce: ~/Semester-4/Topo/Impl_Project
File Edit View Terminal Tabs Help
harsha@xfce:~/Semester-4/Topo/Impl_Project$ python3 tetra.py
Enter the name of file: Tetrahedron.gts
Number of vertices: 4
Number of edges: 6
Number of faces: 4
vertices: [[-1.63299, -0.942809, -0.666667], [0.0, 1.88562, -0.666667], [1.63299, -0.942809, -0.666667], [0.0, 0.0, 2.0]]
edges: [[2, 1], [3, 2], [2, 4], [1, 3], [4, 1], [3, 4]]
faces: [[1, 2, 4], [5, 3, 1], [3, 6, 2], [6, 5, 4]]
harsha@xfce:~/Semester-4/Topo/Impl_Project$
```



Tetrahedron.



Tangle cube.

## 7 References

1. [https://en.wikipedia.org/wiki/Simplicial\\_complex](https://en.wikipedia.org/wiki/Simplicial_complex)
2. <https://jeremykun.com/2013/04/10/computing-homology/>
3. <https://jeremykun.com/2014/01/23/fixing-bugs-in-computing-homology/>
4. <http://gts.sourceforge.net/samples.html>
5. <https://plotly.com/python/3d-mesh/>