

SM 402: Basic Computational Topology
Implementation Project

By

IMT2020042 Gousepeer Arella

IMT2020085 Harshadeep Donapati

IMT2020082 Prudhvi Nath Reddy

Contents

1	Problem Statement	1
2	Algorithm	1
3	Steps to run the code	1
4	Code	1
4.1	Github Link	3
5	Note	3
6	References	3

1 Problem Statement

Given any input simplicial complex (up to 3 -dimensional), compute β_3 using the boundary matrix method.

2 Algorithm

Formula used to calculate β_3 :

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

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

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

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

∂_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 columnspace of a matrix.

3 Steps to run the code

Make sure you have `python3` and `numpy` installed. Use the command `pip3 install numpy` to install it. After, use the command `python3 only.py` to run it. Then enter the name of the of the `.gts` file.

4 Code

```
import time
```

```
import numpy as np
```

```
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))

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 ∂_3 and ∂_4 .

6 References

- 1 . 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>