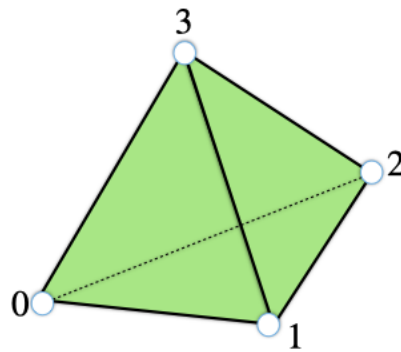


Ejemplos Homología

October 9, 2020

1 Ejemplo 1

Consideramos el 3-símplice $(0,1,2,3)$



```
[4]: sc=SimplicialComplex([(0,1,2,3)])
```

```
[5]: sc.boundarymatrix(1)
```

```
[5]: array([[1., 1., 1., 0., 0., 0.],  
          [1., 0., 0., 1., 1., 0.],  
          [0., 1., 0., 1., 0., 1.],  
          [0., 0., 1., 0., 1., 1.]])
```

```
[6]: sc.boundarymatrix(2)
```

```
[6]: array([[1., 1., 0., 0.],  
          [1., 0., 1., 0.],  
          [0., 1., 1., 0.],  
          [1., 0., 0., 1.],  
          [0., 1., 0., 1.],  
          [0., 0., 1., 1.]])
```

```
[7]: sc.boundarymatrix(3)
```

```
[7]: array([[1.],
           [1.],
           [1.],
           [1.]])
```

```
[8]: print(sc.Betti_number(0),sc.Betti_number(1),sc.Betti_number(2),sc.
      ↪Betti_number(3))
```

```
1 0 0 0
```

2 Ejemplo 2

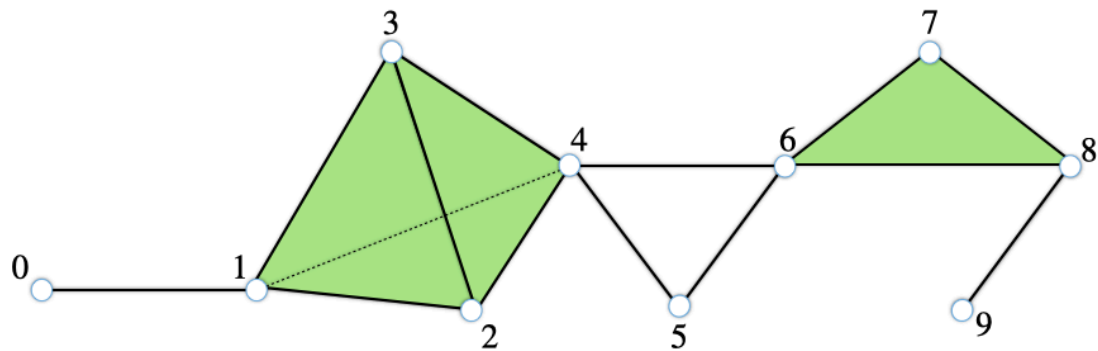
Borde del tetraedro (triangula la 2-esfera)

```
[9]: sc1=SimplicialComplex(list(sc.n_faces(2)))
```

```
[10]: print(sc1.Betti_number(0),sc1.Betti_number(1),sc1.Betti_number(2))
```

```
1 0 1
```

3 Ejemplo 3



Símplices

maximales: (0,1), (1,2,3,4), (4,5), (5,6), (4,6), (6,7,8), (8,9)

```
[11]: sc=SimplicialComplex([(0,1),(1,2,3,4),(4,5),(5,6),(4,6),(6,7,8),(8,9)])
```

```
[12]: sc.boundarymatrix(1)
```

```
[12]: array([[1., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
            [1., 1., 1., 1., 0., 0., 0., 0., 0., 0.],
            [0., 1., 0., 0., 1., 1., 0., 0., 0., 0.],
            [0., 0., 1., 0., 1., 0., 1., 0., 0., 0.]])
```

```
[0., 0., 0., 1., 0., 1., 1., 1., 1., 0., 0., 0., 0., 0.],
[0., 0., 0., 0., 0., 0., 0., 1., 0., 1., 0., 0., 0., 0.],
[0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 1., 1., 1., 0., 0.],
[0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 1., 0.],
[0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 1., 1.],
[0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1.]]
```

```
[13]: sc.boundarymatrix(2)
```

```
[13]: array([[0., 0., 0., 0., 0.],
            [1., 1., 0., 0., 0.],
            [1., 0., 1., 0., 0.],
            [0., 1., 1., 0., 0.],
            [1., 0., 0., 1., 0.],
            [0., 1., 0., 1., 0.],
            [0., 0., 1., 1., 0.],
            [0., 0., 0., 0., 0.],
            [0., 0., 0., 0., 0.],
            [0., 0., 0., 0., 0.],
            [0., 0., 0., 0., 1.],
            [0., 0., 0., 0., 1.],
            [0., 0., 0., 0., 1.],
            [0., 0., 0., 0., 0.]])
```

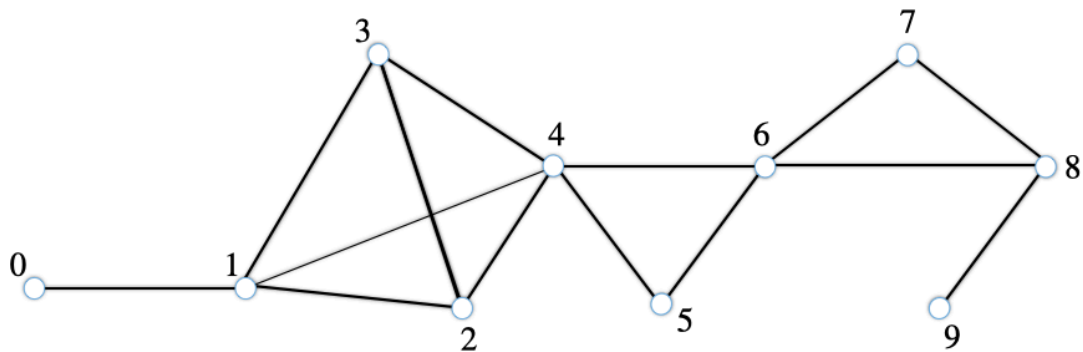
```
[14]: sc.boundarymatrix(3)
```

```
[14]: array([[1.],
            [1.],
            [1.],
            [1.],
            [0.]])
```

```
[15]: print(sc.Betti_number(0),sc.Betti_number(1),sc.Betti_number(2),sc.
            ↪Betti_number(3))
```

```
1 1 0 0
```

4 Ejemplo 4

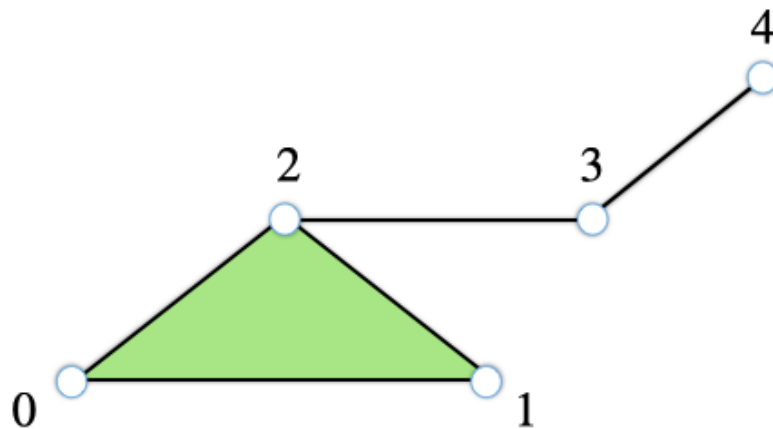


```
[16]: sc1=SimplicialComplex(list(sc.n_faces(1)))
```

```
[17]: print(sc1.Betti_number(0),sc1.Betti_number(1))
```

1 5

5 Ejemplo 5



Símplices maximales: $(0,1,2), (2,3), (3,4)$

```
[18]: sc=SimplicialComplex([(0,1,2),(2,3),(3,4)])
```

```
[19]: sc.boundarymatrix(1)
```

```
[19]: array([[1., 1., 0., 0., 0.],
            [1., 0., 1., 0., 0.],
            [0., 1., 1., 1., 0.],
            [0., 0., 0., 1., 1.],
            [0., 0., 0., 0., 1.]])
```

```
[20]: sc.boundarymatrix(2)
```

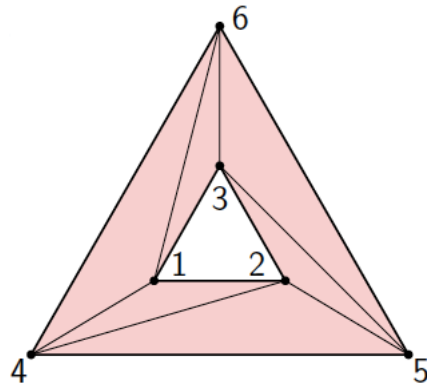
```
[20]: array([[1.],
            [1.],
            [1.],
            [0.],
            [0.]])
```

```
[21]: print(sc.Betti_number(0),sc.Betti_number(1),sc.Betti_number(2))
```

```
1 0 0
```

6 Ejemplo 6

Triangulación del anillo cerrado



Símplices maximales: (1,2,4), (1,3,6), (1,4,6), (2,3,5), (2,4,5), (3,5,6)

```
[22]: sc=SimplicialComplex([(1,2,4),(1,3,6),(1,4,6),(2,3,5),(2,4,5),(3,5,6)])
```

```
[23]: sc.boundarymatrix(1)
```

```
[23]: array([[1., 1., 1., 1., 0., 0., 0., 0., 0., 0., 0., 0.],
            [1., 0., 0., 0., 1., 1., 1., 0., 0., 0., 0., 0.],
            [0., 1., 0., 0., 1., 0., 0., 1., 1., 0., 0., 0.],
            [0., 0., 1., 0., 0., 1., 0., 0., 0., 1., 1., 0.],
            [0., 0., 0., 0., 0., 0., 1., 1., 0., 1., 0., 1.],
            [0., 0., 0., 1., 0., 0., 0., 0., 1., 0., 1., 1.]])
```

```
[24]: sc.boundarymatrix(2)
```

```
[24]: array([[1., 0., 0., 0., 0., 0.],  
          [0., 1., 0., 0., 0., 0.],  
          [1., 0., 1., 0., 0., 0.],  
          [0., 1., 1., 0., 0., 0.],  
          [0., 0., 0., 1., 0., 0.],  
          [1., 0., 0., 0., 1., 0.],  
          [0., 0., 0., 1., 1., 0.],  
          [0., 0., 0., 1., 0., 1.],  
          [0., 1., 0., 0., 0., 1.],  
          [0., 0., 0., 0., 1., 0.],  
          [0., 0., 1., 0., 0., 0.],  
          [0., 0., 0., 0., 0., 1.]])
```

```
[25]: print(sc.Betti_number(0),sc.Betti_number(1),sc.Betti_number(2))
```

```
1 1 0
```

7 Ejemplo 7

1-esqueleto del anillo

```
[26]: sc1=SimplicialComplex(list(sc.n_faces(1)))
```

```
[27]: print(sc1.Betti_number(0),sc1.Betti_number(1))
```

```
1 7
```

8 Ejemplo 8

El toro

[0., 0., 1., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0.],
 [1., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0.],
 [0., 0., 1., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0.],
 [0., 0., 0., 1., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0.],
 [0., 1., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0.],
 [0., 0., 0., 0., 0., 0., 1., 1., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0.],
 [1., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0.,
 0., 0.],
 [0., 0., 0., 0., 0., 0., 1., 0., 1., 0., 0., 0., 0., 0., 0., 0.,
 0., 0.],
 [0., 1., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0.,
 0., 0.],
 [0., 0., 0., 0., 0., 0., 0., 1., 0., 1., 0., 0., 0., 0., 0., 0.,
 0., 0.],
 [0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 1., 0., 0., 0., 0., 0.,
 0., 0.],
 [0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0.,
 0., 0.],
 [0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0.,
 0., 0.],
 [0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 1., 0., 0., 0., 0.,
 0., 0.],
 [0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 1., 0., 0., 0.,
 0., 0.],
 [0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0.,
 0., 0.],
 [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 1.,
 0., 0.],
 [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 1.,
 0., 0.],
 [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 1.,
 0., 0.],
 [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0.,
 1., 0.],
 [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1.,
 0., 1.],
 [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0.,
 0., 1.],
 [0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0.],


```

1., 0.],
[0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 1., 0.,
0., 0.],
[0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0.,
0., 1.]]

```

```
[31]: print(sc.Betti_number(0),sc.Betti_number(1),sc.Betti_number(2))
```

```
1 2 1
```

9 Ejemplo 8

1-esqueleto del toro

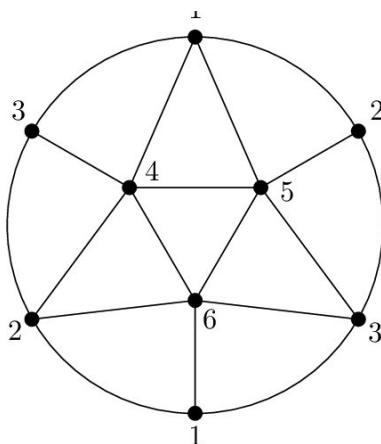
```
[32]: sc1=SimplicialComplex(list(sc.n_faces(1)))
```

```
[33]: print(sc1.Betti_number(0),sc1.Betti_number(1))
```

```
1 19
```

10 Ejemplo 9

El plano proyectivo



Símplices maximales: (1,2,6), (2,3,4), (1,3,4), (1,2,5), (2,3,5), (1,3,6), (2,4,6), (1,4,5), (3,5,6), (4,5,6)

```
[34]: sc=SimplicialComplex([(1,2,6), (2,3,4), (1,3,4), (1,2,5), (2,3,5), (1,3,6),
↪(2,4,6), (1,4,5), (3,5,6), (4,5,6)])
```

```
[35]: sc.boundarymatrix(1)
```

```
[35]: array([[1., 1., 1., 1., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
[1., 0., 0., 0., 0., 1., 1., 1., 1., 0., 0., 0., 0., 0., 0.]])
```

```
[0., 1., 0., 0., 0., 1., 0., 0., 0., 1., 1., 1., 0., 0., 0.],
[0., 0., 1., 0., 0., 0., 1., 0., 0., 1., 0., 0., 1., 1., 0.],
[0., 0., 0., 1., 0., 0., 0., 1., 0., 0., 1., 0., 1., 0., 1.],
[0., 0., 0., 0., 1., 0., 0., 0., 1., 0., 0., 1., 0., 1., 1.]])
```

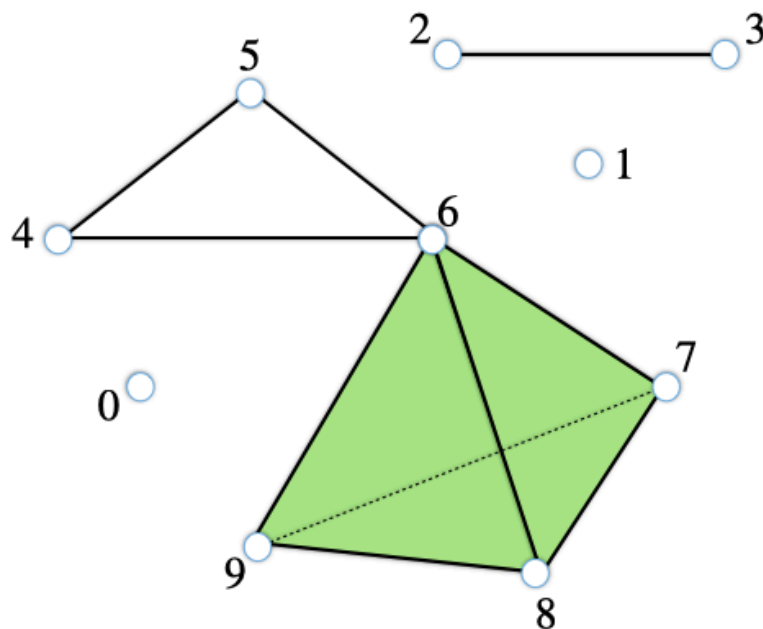
```
[36]: sc.boundarymatrix(2)
```

```
[36]: array([[1., 1., 0., 0., 0., 0., 0., 0., 0., 0.],
            [0., 0., 1., 1., 0., 0., 0., 0., 0., 0.],
            [0., 0., 1., 0., 1., 0., 0., 0., 0., 0.],
            [1., 0., 0., 0., 1., 0., 0., 0., 0., 0.],
            [0., 1., 0., 1., 0., 0., 0., 0., 0., 0.],
            [0., 0., 0., 0., 0., 1., 1., 0., 0., 0.],
            [0., 0., 0., 0., 0., 1., 0., 1., 0., 0.],
            [1., 0., 0., 0., 0., 0., 1., 0., 0., 0.],
            [0., 1., 0., 0., 0., 0., 0., 1., 0., 0.],
            [0., 0., 1., 0., 0., 1., 0., 0., 0., 0.],
            [0., 0., 0., 0., 0., 0., 1., 0., 1., 0.],
            [0., 0., 0., 1., 0., 0., 0., 0., 1., 0.],
            [0., 0., 0., 0., 1., 0., 0., 0., 0., 1.],
            [0., 0., 0., 0., 0., 0., 0., 1., 0., 1.],
            [0., 0., 0., 0., 0., 0., 0., 0., 1., 1.]])
```

```
[37]: print(sc.Betti_number(0),sc.Betti_number(1),sc.Betti_number(2))
```

```
1 1 1
```

11 Ejemplo 10



Símplices maximales: $(0,)$, $(1,)$, $(2,3)$, $(4,5)$, $(5,6)$, $(4,6)$, $(6,7,8,9)$

```
[38]: sc=SimplicialComplex([(0,), (1,), (2,3), (4,5), (5,6), (4,6), (6,7,8,9)])
```

```
[39]: sc.boundarymatrix(1)
```

```
[39]: array([[0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
            [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
            [1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
            [1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
            [0., 1., 1., 0., 0., 0., 0., 0., 0., 0., 0.],
            [0., 1., 0., 1., 0., 0., 0., 0., 0., 0., 0.],
            [0., 0., 1., 1., 1., 1., 1., 0., 0., 0., 0.],
            [0., 0., 0., 0., 1., 0., 0., 1., 1., 0., 0.],
            [0., 0., 0., 0., 0., 1., 0., 1., 0., 1., 0.],
            [0., 0., 0., 0., 0., 0., 1., 0., 1., 1.]])
```

```
[40]: sc.boundarymatrix(2)
```

```
[40]: array([[0., 0., 0., 0.],
            [0., 0., 0., 0.],
            [0., 0., 0., 0.],
            [0., 0., 0., 0.],
            [1., 1., 0., 0.]])
```

```
[1., 0., 1., 0.],  
[0., 1., 1., 0.],  
[1., 0., 0., 1.],  
[0., 1., 0., 1.],  
[0., 0., 1., 1.]])
```

```
[41]: print(sc.Betti_number(0),sc.Betti_number(1),sc.Betti_number(2),sc.  
↪Betti_number(3))
```

```
4 1 0 0
```