



Τοπολογική Ανάλυση Δεδομένων

Εαρινό Εξάμηνο 2026

(Νέο Μάθημα Επιλογής)

Φροντιστηριακή Διάλεξη 1: Εισαγωγή

Διδάσκοντες:

Αθανάσιος Ανδρικόπουλος

Ιωάννης Γουναρίδης

Επικουρικό Έργο: Αλέξανδρος - Μάριος Αφράτης

Πάτρα, 5 Μαρτίου 2026

Εισαγωγή σε Python & TDA

Απλοτοπικά Συμπλέγματα & Φιλτράρισμα

Δέντρο Απλοτόπων

Θεμελιώδη Απλοτοπικά Συμπλέγματα

Οπτικοποίηση Απλοτοπικών Συμπλεγμάτων

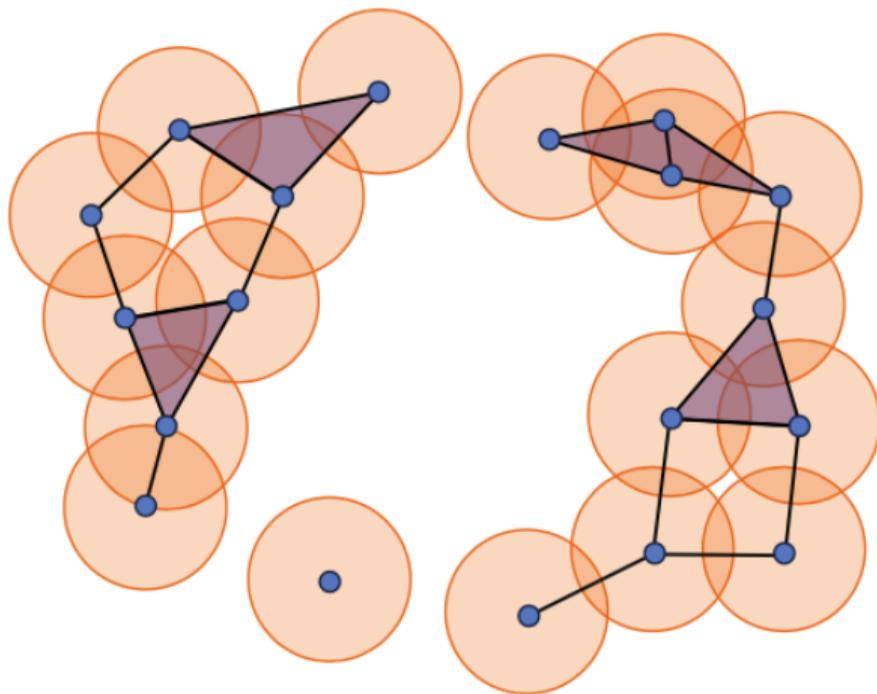
Σύνοψη



PyCharm  
JETBRAINS IDE



python™



# Τι είναι η Python; (What is Python?)

## Ορισμός (Definition)

Η Python είναι μια υψηλού επιπέδου, γενικής χρήσης γλώσσα προγραμματισμού με έμφαση στην αναγνωσιμότητα κώδικα.

## Χαρακτηριστικά (Features)

- Απλή σύνταξη, εύκολη εκμάθηση
- Δυναμική γλώσσα (dynamically typed)
- Ερμηνευόμενη (interpreted)
- Πλούσιο οικοσύστημα βιβλιοθηκών
- Ανοικτού κώδικα (open source)

## Εφαρμογές (Applications)

- Επιστήμη Δεδομένων (Data Science)
- Μηχανική Μάθηση (Machine Learning)
- Επιστημονικός Υπολογισμός
- Τοπολογική Ανάλυση Δεδομένων (TDA)
- Ανάπτυξη Λογισμικού

## Ορισμός (Definition)

Η **Τοπολογική Ανάλυση Δεδομένων** (Topological Data Analysis) εξαγεί τοπολογικές υπογραφές (topological signatures) από νέφη σημείων (point clouds) σε  $\mathbb{R}^d$  ή σε γενικούς μετρικούς χώρους.

## Βασική Ιδέα (Main Idea)

- Μελετάμε την τοπολογία ενώσεων σφαιρών με κέντρα τα σημεία του νέφους (offsets)
- Μη διακριτά σύνολα (π.χ. offsets) κωδικοποιούνται ως πεπερασμένες διακριτές δομές
- Χρησιμοποιούμε **απλοτοπικά συμπλέγματα** (simplicial complexes) για προσέγγιση

## Επιστημονικές Βιβλιοθήκες

- `numpy` – αριθμητικοί υπολογισμοί
- `matplotlib` – οπτικοποίηση
- `scipy` – επιστημονικοί αλγόριθμοι
- `pickle` – σειριοποίηση δεδομένων

## Βιβλιοθήκες TDA

- `gudhi` – κύρια βιβλιοθήκη TDA
- `plotly` – διαδραστική οπτικοποίηση
- `ipywidgets` – διαδραστικά widgets
- `ripser` – γρήγορη επίμονη ομολογία

## Βασικές εισαγωγές για TDA

```
1 import numpy as np
2 import gudhi as gd
3 import matplotlib.pyplot as plt
4 from mpl_toolkits.mplot3d import Axes3D
5 import pickle
6
```

## Σημείωση

Η βιβλιοθήκη `gudhi` (Geometry Understanding in Higher Dimensions) παρέχει εργαλεία για:

- Κατασκευή απλοτοπικών συμπλεγμάτων (simplicial complexes)
- Υπολογισμό επίμονης ομολογίας (persistent homology)
- Οπτικοποίηση τοπολογικών δομών

# Πίνακας Περιεχομένων

Εισαγωγή σε Python & TDA

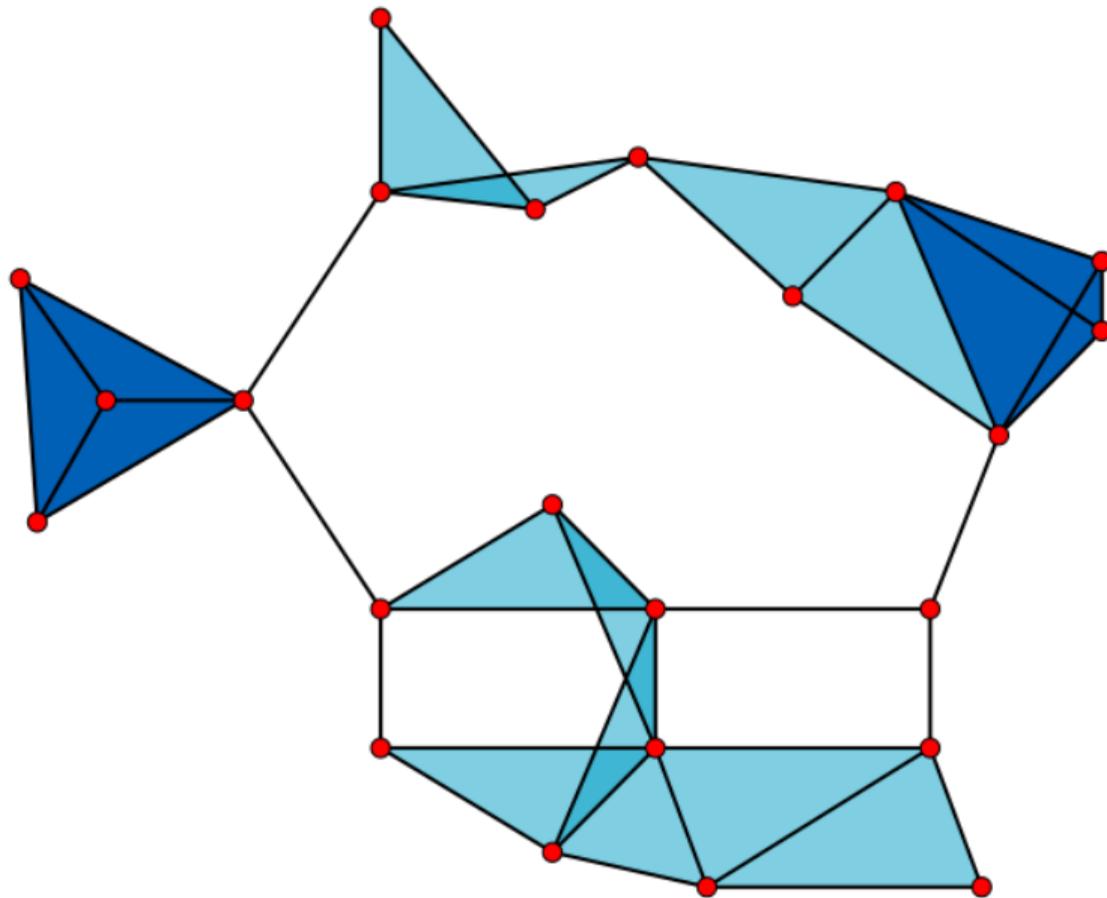
Απλοτοπικά Συμπλέγματα & Φιλτράρισμα

Δέντρο Απλοτόπων

Θεμελιώδη Απλοτοπικά Συμπλέγματα

Οπτικοποίηση Απλοτοπικών Συμπλεγμάτων

Σύνοψη



**Απλοτοπικά Συμπλέγματα & Φιλτράρισμα**

# Απλοτοπικά Συμπλέγματα (Simplicial Complexes)

## Ορισμός

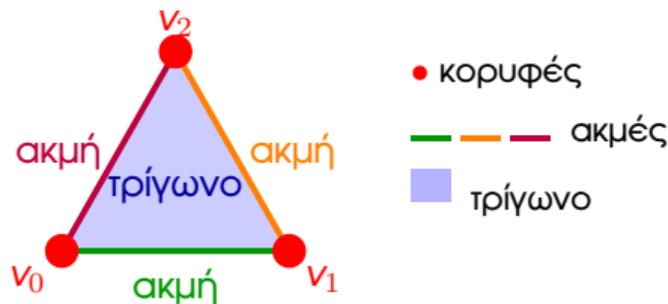
Ένα **απλοτοπικό σύμπλεγμα** (*simplicial complex*)  $\mathcal{K}$  είναι ένα σύνολο **απλοτόπων** (*simplices*).  
Αποτελεί γενίκευση υψηλότερης διάστασης ενός γραφήματος.

## Τύποι Απλοτόπων (Types of Simplices)

- **0-απλότοπος**: κορυφή (vertex)
- **1-απλότοπος**: ακμή (edge)
- **2-απλότοπος**: τρίγωνο (triangle)
- **k-απλότοπος**: k-διάστατος απλότοπος

## Από Παράδειγμα

Το παρακάτω τρίγωνο είναι ένα σύνολο απλοτόπων (κορυφές & ακμές).



# Φιλτράρισμα (Filtration)

## Ορισμός (Definition)

Το **φιλτράρισμα** (filtration) είναι μια αύξουσα ακολουθία υπο-συμπλεγμάτων ενός απλοτοπικού συμπλέγματος  $\mathcal{K}$ :

$$\emptyset = \mathcal{K}_0 \subseteq \mathcal{K}_1 \subseteq \dots \subseteq \mathcal{K}_n = \mathcal{K}$$

όπου:

- $\mathcal{K}_0 = \emptyset$ : ξεκινάμε με **κενό** σύμπλεγμα
- $\mathcal{K}_i \subseteq \mathcal{K}_{i+1}$ : σε κάθε βήμα **προσθέτουμε** απλοτόπους, ποτέ δεν αφαιρούμε
- $\mathcal{K}_n = \mathcal{K}$ : στο τέλος έχουμε **ολόκληρο** το σύμπλεγμα

## Τιμή Φιλτραρίσματος (Filtration Value)

- Κάθε απλότοπος έχει μια **τιμή φιλτραρίσματος** (filtration value) – μια χρονική σφραγίδα που δείχνει πότε εμφανίστηκε
- **Ιδιότητα συμπερίληψης**: κάθε έδρα απλοτόπου με τιμή  $t$  έχει τιμή  $\leq t$

# Πίνακας Περιεχομένων

Εισαγωγή σε Python & TDA

Απλοτοπικά Συμπλέγματα & Φιλτράρισμα

**Δέντρο Απλοτόπων**

Θεμελιώδη Απλοτοπικά Συμπλέγματα

Οπτικοποίηση Απλοτοπικών Συμπλεγμάτων

Σύνοψη

## Ορισμός στην GUDHI

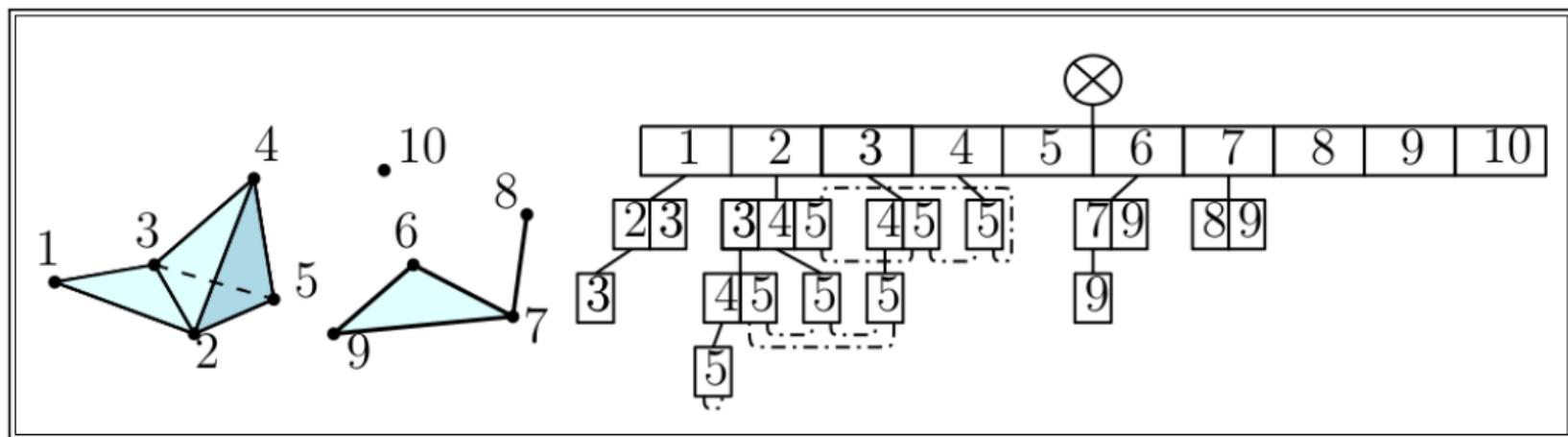
Στη βιβλιοθήκη GUDHI, τα φιλτραρισμένα απλοτοπικά συμπλέγματα κωδικοποιούνται μέσω μιας δομής δεδομένων που ονομάζεται **Δέντρο Απλοτόπων** (Simplex Tree).

## Αναπαράσταση

- Κορυφές: ακέραιοι αριθμοί
- Ακμές: λίστα 2 ακεραίων
- Τρίγωνα: λίστα 3 ακεραίων
- k-απλότοποι: λίστα k+1 ακεραίων

## Βασικές Μέθοδοι (Basic Methods)

- `insert()` – εισαγωγή
- `find()` – εύρεση
- `get_filtration()` – λήψη φιλτραρίσματος
- `dimension()` – διάσταση
- `num_vertices()` – αριθμός κορυφών
- `num_simplices()` – αριθμός απλοτόπων



Σχ. 1.1: Χρήση δέντρου απλοτόπων ως αναπαράστασης απλοτοπικών συμπλεγμάτων από σημεία (σχηματικών) δεδομένων

## Εισαγωγή ακμών (Inserting edges)

```
1 import gudhi as gd
2
3 st = gd.SimplexTree() # Κενό δέντρο απλοτόπων
4
5 # Εισαγωγή ακμών αυτόματα( εισάγονται και οι κορυφές)
6 st.insert([0, 1])
7 st.insert([1, 2])
8 st.insert([3, 1])
9
10 # Εκτύπωση όλων των απλοτόπων
11 for splx in st.get_filtration():
12     print(splx)
13 # Αποτέλεσμα:
14 # ([0], 0.0), ([1], 0.0), ([0, 1], 0.0)
15 # ([2], 0.0), ([1, 2], 0.0)
16 # ([3], 0.0), ([1, 3], 0.0)
17
```

## Σημαντική Σημείωση (Important Note)

Η εισαγωγή μιας ακμής **αυτόματα εισάγει** τις κορυφές της αν δεν υπάρχουν ήδη, για να ικανοποιηθεί η ιδιότητα συμπερίληψης.

## Βασικές Πληροφορίες

```
1 # Διάσταση συμπλέγματος
2 st.dimension()
3 # >> 1
4
5 # Αριθμός κορυφών
6 st.num_vertices()
7 # >> 4
8
9 # Αριθμός απλοτόπων
10 st.num_simplices()
11 # >> 7
12
13 # Σκελετός διάστασης 1
14 print(st.get_skeleton(1))
15
16 # Εύρεση απλοτόπου
17 st.find([2, 4])
18 # >> False
19
```

## d-Σκελετός (d-Skeleton)

### Ορισμός

Ο **d-σκελετός** (*d-skeleton*) είναι η ένωση όλων των απλοτόπων διάστασης  $\leq d$ .

### Περιγραφή Μεθόδων

- `dimension()`: μέγιστη διάσταση
- `num_vertices()`: αριθμός κορυφών
- `num_simplices()`: σύνολο απλοτόπων
- `get_skeleton(d)`: *d*-σκελετός
- `find([...])`: έλεγχος ύπαρξης

## Εισαγωγή με τιμές φιλτραρίσματος

```
1 # Εισαγωγή τριγώνων με διαφορετικές τιμές φιλτραρίσματος
2 st.insert([0, 1, 2], filtration=0.1)
3 st.insert([1, 2, 3], filtration=0.2)
4 st.insert([0, 1, 3], filtration=0.4)
5
6 # Αλλαγή τιμής φιλτραρίσματος συγκεκριμένου απλοτόπου
7 st.assign_filtration([3], filtration=0.8)
8
9 # Επαναφορά μηψθίνουσας- ιδιότητας
10 st.make_filtration_non_decreasing()
11
12 # Ανάκτηση τιμής φιλτραρίσματος
13 print(st.filtration([2, 3])) # >> 0.8
14
```

## Ιδιότητα Μη-Φθίνοντος Φιλτραρίσματος

Κάθε έδρα ενός απλοτόπου με τιμή  $t$  πρέπει να έχει τιμή  $\leq t$ . Η `make_filtration_non_decreasing()` διορθώνει τυχόν παραβάσεις.

# Πίνακας Περιεχομένων

Εισαγωγή σε Python & TDA

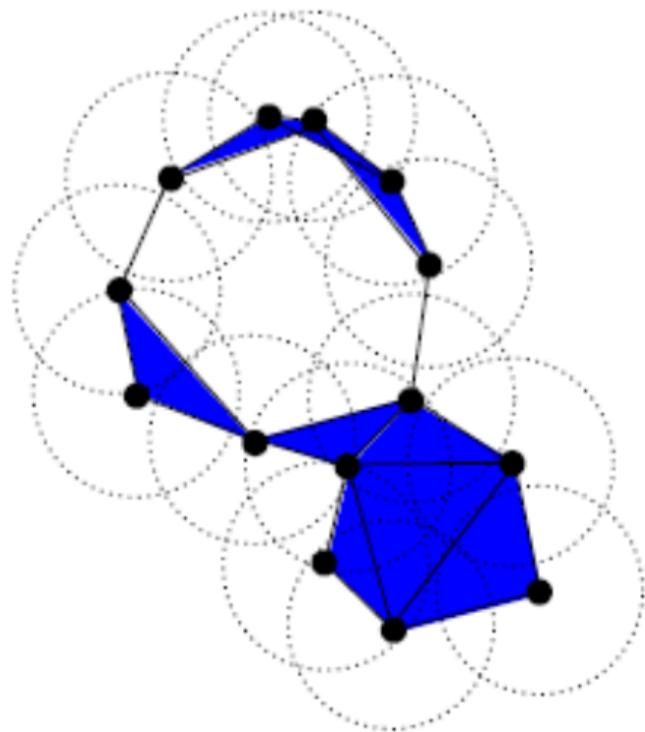
Απλοτοπικά Συμπλέγματα & Φιλτράρισμα

Δέντρο Απλοτόπων

Θεμελιώδη Απλοτοπικά Συμπλέγματα

Οπτικοποίηση Απλοτοπικών Συμπλεγμάτων

Σύνοψη



Θεμελιώδη Απλοτοπικά Συμπλέγματα

# Σύμπλεγμα Vietoris-Rips (Vietoris-Rips Complex)

## Ορισμός

Το σύμπλεγμα  $\alpha$ -Rips ενός νέφους σημείων  $\mathbb{X}$  σε  $(\mathbb{R}^d, \|\cdot\|_2)$  ορίζεται σχηματίζοντας έναν απλότοπο για κάθε πεπερασμένο υποσύνολο του  $\mathbb{X}$  με διάμετρο το πολύ  $\alpha$ .

## Κατασκευή στην GUDHI

1. Κατασκευή 1-σκελετού (RipsComplex)
2. Επέκταση σε υψηλότερες διαστάσεις (create\_simplex\_tree)

## Παράμετροι

- `points`: το νέφος σημείων
- `max_edge_length`: μέγιστο μήκος ακμής  $\alpha$
- `max_dimension`: μέγιστη διάσταση

## Προσοχή

Ο αριθμός απλοτόπων αυξάνεται **πολύ γρήγορα** με τον αριθμό σημείων και τη διάσταση!

## Κατασκευή Rips Συμπλέγματος

```
1 import numpy as np
2 import gudhi as gd
3 import pickle
4
5 # Φόρτωση δεδομένων επιταχυνσιόμετρου (accelerometer data)
6 f = open("./datasets/data_acc", "rb")
7 data = pickle.load(f)
8 f.close()
9 data_A = data[0]
10 data_A_sample = data_A[0] # 200 σημεία στο R^3
11
12 # Κατασκευή σκελετού 1- με max ακμή = 0.2
13 skeleton = gd.RipsComplex(points=data_A_sample, max_edge_length=0.2)
14
15 # Δημιουργία δέντρου απλοτόπων μέχρι διάσταση 3
16 Rips_st = skeleton.create_simplex_tree(max_dimension=3)
17
18 print("Διάσταση:", Rips_st.dimension()) # 3
19 print("Κορυφές:", Rips_st.num_vertices()) # 200
20 print("Απλότοποι:", Rips_st.num_simplices())
21
```

# Σύμπλεγμα Alpha (Alpha Complex)

## Ορισμός

Το σύμπλεγμα Alpha κατασκευάζεται από τα πεπερασμένα κελιά μιας **Τριγωνοποίησης Delaunay** (Delaunay triangulation). Είναι υπο-σύμπλεγμα του Delaunay συμπλέγματος.

## Πλεονεκτήματα έναντι Rips

- Περιέχει **λιγότερους απλοτόπους**
- Πιο αποδοτικό υπολογιστικά
- Γεωμετρικό (geometric) σύμπλεγμα

## Σύγκριση (Comparison)

	Rips	Alpha
Απλότοποι	Πολλοί	Λιγότεροι
Ταχύτητα	Αργό	Γρήγορο
Γεωμετρικό	Όχι	Ναι

## Κατασκευή Συμπλέγματος Alpha

```
1 # Κατασκευή συμπλέγματος Alpha
2 alpha_complex = gd.AlphaComplex(points=data_A_sample)
3
4 # max_alpha_square: τετράγωνο μέγιστης τιμής alpha
5 st_alpha = alpha_complex.create_simplex_tree(max_alpha_square=0.2**2)
6
7 # Σύγκριση με Rips
8 print("Alpha -- Διάσταση:", st_alpha.dimension()) # 3
9 print("Alpha -- Κορυφές:", st_alpha.num_vertices()) # 200
10 print("Alpha -- Απλότοποι:", st_alpha.num_simplices())
11 # Αποτέλεσμα: πολύ λιγότεροι απλότοποι από το Rips!
12
13 # Οπτικοποίηση τροχιάς επιταχύνσεων
14 import matplotlib.pyplot as plt
15 from mpl_toolkits.mplot3d import Axes3D
16 fig = plt.figure()
17 ax = fig.add_subplot(111, projection='3d')
18 ax.scatter(data_A_sample[:, 0], data_A_sample[:, 1], data_A_sample[:, 2])
19 plt.show()
20
```

# Πίνακας Περιεχομένων

Εισαγωγή σε Python & TDA

Απλοτοπικά Συμπλέγματα & Φιλτράρισμα

Δέντρο Απλοτόπων

Θεμελιώδη Απλοτοπικά Συμπλέγματα

Οπτικοποίηση Απλοτοπικών Συμπλεγμάτων

Σύνοψη

## Matplotlib

- Κλασική βιβλιοθήκη Python
- Στατικές απεικονίσεις
- `plot_trisurf()` για τριγωνισμούς
- Υποστήριξη 3D (`Axes3D`)
- Απλή χρήση

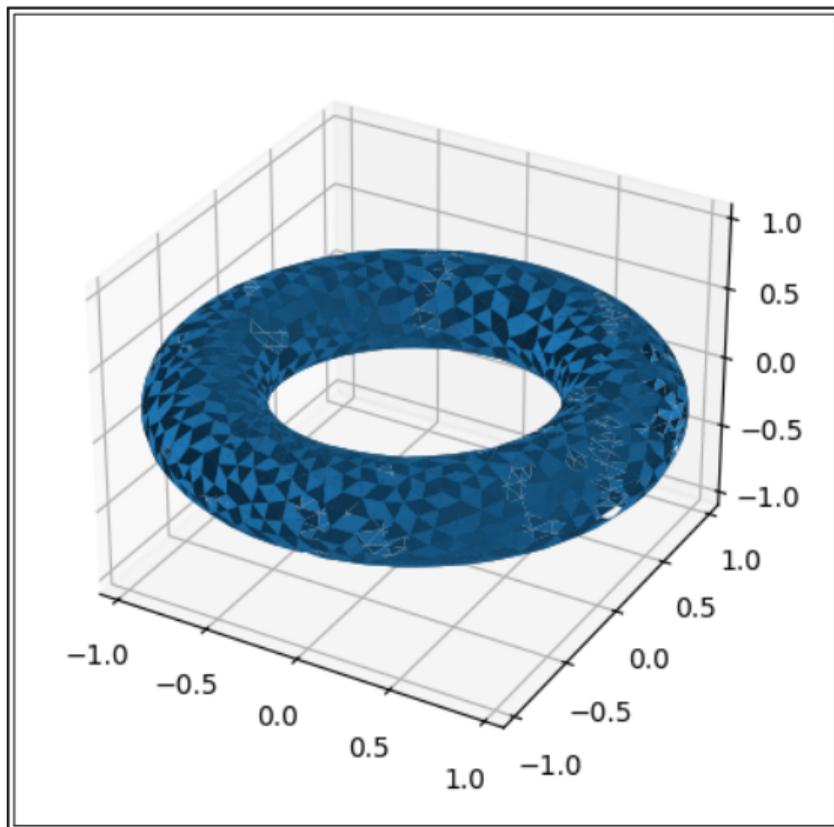
## Άλλες Βιβλιοθήκες (Other Libraries)

Mayavi, iPyVolume, εργαλεία GUDHI (`plot_rips_complex.py`,  
`plot_alpha_complex.py`)

## Τόρος σε 3D από Σύμπλεγμα Alpha

```
1 import numpy as np
2 import gudhi as gd
3 import matplotlib.pyplot as plt
4 from mpl_toolkits.mplot3d import Axes3D
5
6 # Φόρτωση σημείων από τόρο
7 torus = gd.read_points_from_off_file(off_file='datasets/tore3D_1307.off')
8 ac = gd.AlphaComplex(points=torus)
9 st = ac.create_simplex_tree()
10
11 # Ανάκτηση συντεταγμένων και τριγώνων
12 points = np.array([ac.get_point(i) for i in range(st.num_vertices())])
13 triangles = np.array([s[0] for s in st.get_skeleton(2)
14 if len(s[0]) == 3 and s[1] <= 0.005])
15
16 # Οπτικοποίηση τριγωνοποίησης
17 fig = plt.figure()
18 ax = fig.add_subplot(projection='3d')
19 ax.plot_trisurf(points[:, 0], points[:, 1], points[:, 2],
20 triangles=triangles)
21 plt.show()
22
```

## Αποτέλεσμα Οπτικοποίησης (Visualization Result)



Σχ. 1.2: Τόρος σε 3D από Σύμπλεγμα Αλφα

# Πίνακας Περιεχομένων

Εισαγωγή σε Python & TDA

Απλοτοπικά Συμπλέγματα & Φιλτράρισμα

Δέντρο Απλοτόπων

Θεμελιώδη Απλοτοπικά Συμπλέγματα

Οπτικοποίηση Απλοτοπικών Συμπλεγμάτων

Σύνοψη

### Κύριες Έννοιες που Καλύφθηκαν

- **Απλότοποι** (Simplices) & **Απλοτοπικά Συμπλέγματα** (Simplicial Complexes)
- **Φιλτράρισμα** (Filtration) και τιμές φιλτραρίσματος
- **Δέντρο Απλοτόπων** (Simplex Tree) – δομή δεδομένων GUDHI
- **Vietoris-Rips** σύμπλεγμα από νέφος σημείων
- **Alpha σύμπλεγμα** από Delaunay τριγωνοποίηση
- **Οπτικοποίηση** με Matplotlib

Σας ευχαριστώ θερμά για την προσοχή σας!

Ερωτήσεις;



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΠΑΤΡΩΝ  
UNIVERSITY OF PATRAS