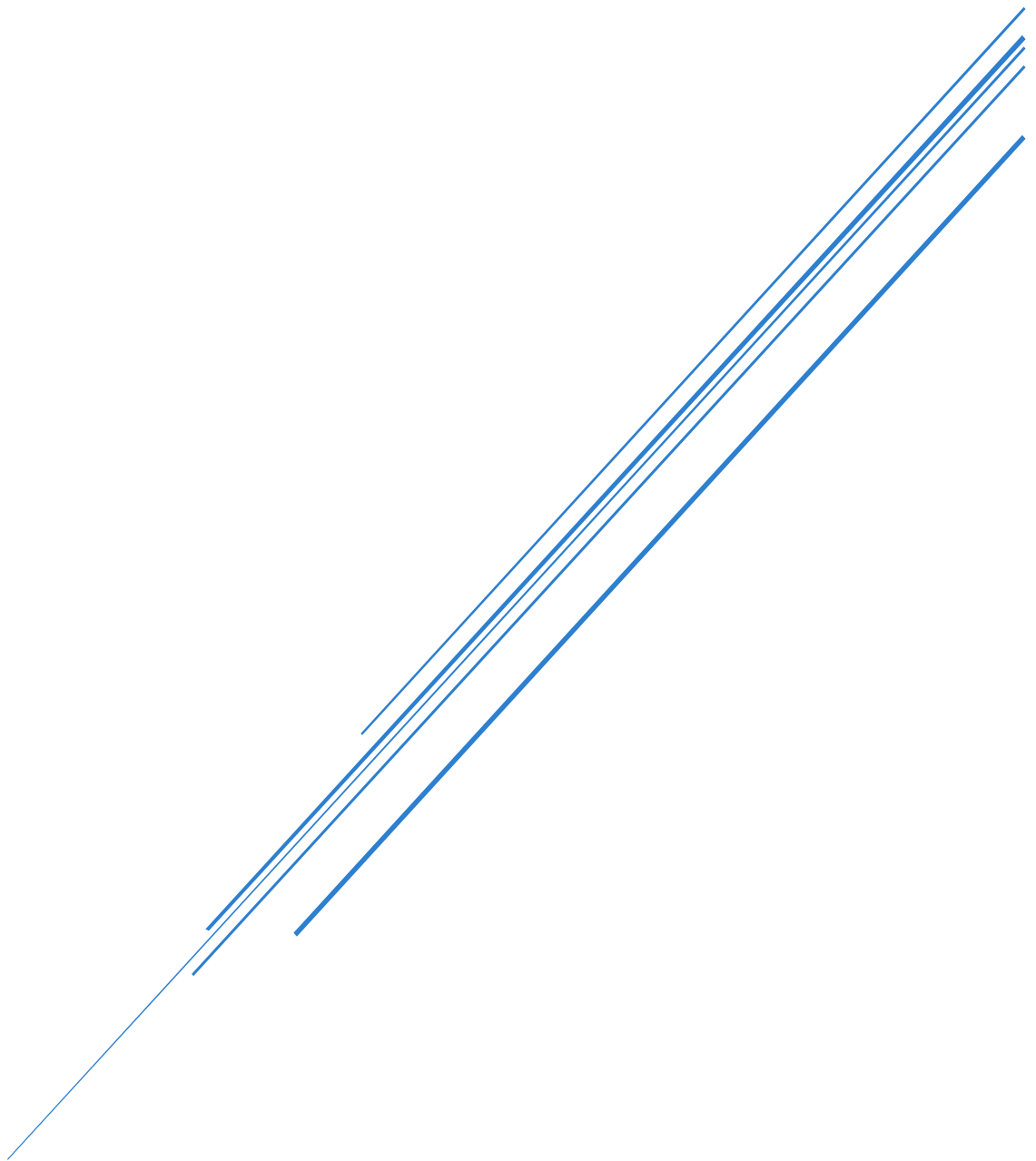


3Δ ΥΠΟΛΟΓΙΣΤΙΚΗ ΓΕΩΜΕΤΡΙΑ ΚΑΙ ΟΡΑΣΗ

4^η Εργαστηριακή Άσκηση



1092813
Γέροντας Νικόλαος

Περιεχόμενα

Ασκήσεις 1 έως και 7:.....	2
Άσκηση 8 ^η :.....	2
Άσκηση 9 ^η :.....	4

Ασκήσεις 1 έως και 7:

Βρείτε το κέντρο μάζας των κορυφών του μοντέλου.

Μετατοπίστε το μοντέλο ώστε το κέντρο μάζας των κορυφών να έρθει στο (0, 0, 0).

Διατηρώντας το κέντρο μάζας στο (0, 0, 0), εκτελέστε κλιμάκωση του μοντέλου, έτσι ώστε να εφάπτεται στη μοναδιαία σφαίρα.

Βρείτε το Axis Aligned Bounding Box του μοντέλου.

Απεικονίστε τα principal components του μοντέλου, που προκύπτουν από την εφαρμογή της μεθόδου Principal Component Analysis (PCA).

Υπόδειξη: Χρησιμοποιήστε την κλάση *Arrow3D*.

Βρείτε τα τρίγωνα του μοντέλου που τέμνονται με το επίπεδο και χρωματίστε τα.

Διαγράψτε τα τρίγωνα της τομής του μοντέλου με το επίπεδο και απομακρύνετε μεταξύ τους τα 2 τμήματα στα οποία χωρίζεται το μοντέλο.

Υπόδειξη: Χρησιμοποιήστε τη φορά που δείχνει το *normal* του επιπέδου για την απομάκρυνση.

Απάντηση:

✓ Ολοκληρώθηκαν κατά τη διάρκεια του εργαστηρίου!

Άσκηση 8^η:

Βρείτε τα ευθύγραμμα τμήματα των τριγώνων που αποτελούν την ακριβή τομή του μοντέλου με το επίπεδο. Προσθέστε τα σε ένα **LineSet3D**, ώστε να δείξετε την ακριβή τομή.

Απάντηση:

Ο στόχος μας είναι να βρούμε την τομή ανάμεσα στο επίπεδο και τα τρίγωνα του λαγού με ακρίβεια, δηλαδή τα σημεία όπου κόβει το επίπεδο κάθε τρίγωνο! Αναλυτικότερα:

- ✓ Από το Task 6, έχουμε ήδη εντοπίσει ποια τρίγωνα τέμνονται με το επίπεδο (λίστα intersections).
- ✓ Στο task_8_exact_intersection, για κάθε τέτοιο τρίγωνο:
 - Καλείται η **intersect_triangle_with_plane()** για να βρει τα ακριβή σημεία τομής στις ακμές του τριγώνου.
 - Τα σημεία τομής (2 ανά τρίγωνο) δημιουργούν γραμμές, οι οποίες απεικονίζονται με κόκκινο (LineSet3D).

Έτσι προκύπτει μια καμπύλη, που δείχνει την ακριβή τομή του αντικειμένου με το επίπεδο!

✎ **intersect_triangle_with_plane()**

- ✓ Για κάθε ακμή του τριγώνου:
Αν τα άκρα της βρίσκονται σε διαφορετικές πλευρές του επιπέδου (δηλαδή, $d1 * d2 < 0$), τότε τέμνεται η **ακμή**.

Συγκεκριμένα:

$$\vec{x}(t) = (1 - t)\vec{v}_i + t\vec{v}_j, t \in [0, 1]$$

Εξίσωση ευθυγράμμου τμήματος {ακμής} ($t = \lambda$)
 v = κάποιο vertex του τριγώνου

✓ Εμείς θέλουμε να βρούμε τη **τιμή του t που αντιστοιχεί στο σημείο τομής με το επίπεδο**, δηλαδή το σημείο που:

$$\vec{n} \cdot \vec{x}(t) = d$$

Αντικατάσταση:

$$\vec{n} \cdot [(1-t)\vec{v}_i + t\vec{v}_j] = d \Rightarrow (1-t)(\vec{n} \cdot \vec{v}_i) + t(\vec{n} \cdot \vec{v}_j) = d$$

Λύνοντας ως προς t:

$$t = \frac{d_i}{d_i - d_j}$$

Επομένως:

$$\vec{p} = (1-t)\vec{v}_i + t\vec{v}_j$$

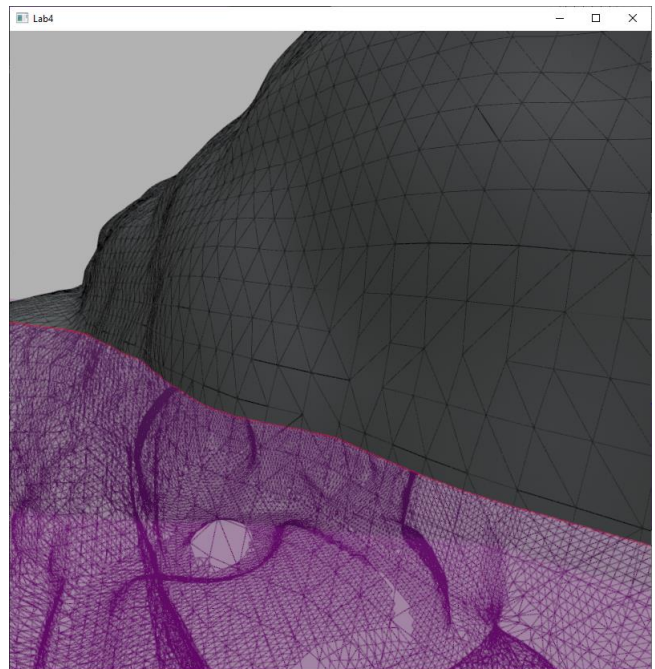
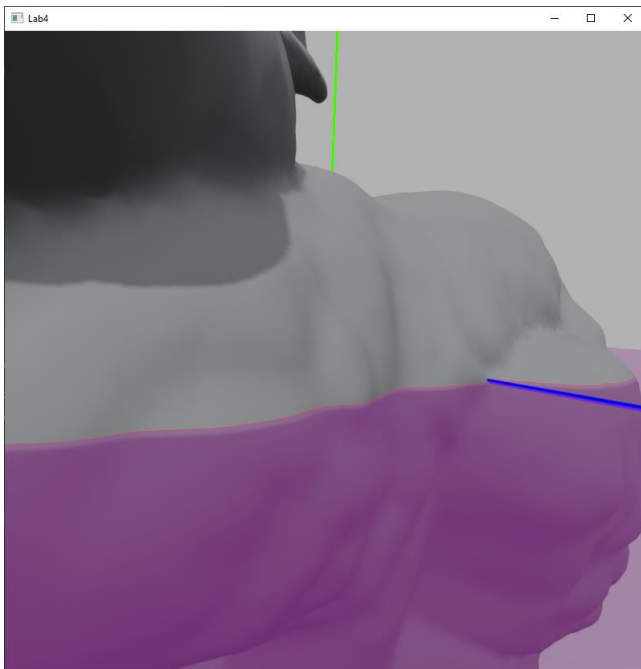
σημείο τομής

```
def intersect_triangle_with_plane(triangle: np.ndarray, n: np.ndarray, d: float) -> list[np.ndarray]:
    ''' n * x = d {Plane Vector Equation} '''

    distances = np.dot(triangle, n) - d # Όπως στο task_6_plane_intersection

    edges = [(0,1), (1,2), (2,0)] # Όλες οι ακμές του τριγώνου
    intersection_points = []
    for (i, j) in edges:
        d1 = distances[i]
        d2 = distances[j]
        if d1 * d2 < 0:
            t = d1 / (d1 - d2) # t = λ
            p = t * triangle[j] + (1 - t) * triangle[i]
            intersection_points.append(p)

    return intersection_points;
```



Άσκηση 9^η:

Κόψτε το μοντέλο στα δύο σύμφωνα με την **ακριβή τομή**.

Υπόδειξη: Η τομή δημιουργεί ένα τρίγωνο και ένα τετράπλευρο. Αφού εντοπίσετε τα τρίγωνα που τέμνονται με το επίπεδο, πραγματοποιήστε εκ νέου τριγωνοποίηση και, έπειτα, απομάκρυνση των δύο τμημάτων του μοντέλου.

Απάντηση:

Στην υλοποίηση του Task 9 **πραγματοποιείται ακριβής τομή του μοντέλου με το επίπεδο**. Αρχικά εντοπίζονται τα τρίγωνα που τέμνονται με το επίπεδο και στη συνέχεια υπολογίζονται τα σημεία τομής στις ακμές κάθε τέτοιου τριγώνου. Ανάλογα με τη θέση των κορυφών (πάνω ή κάτω από το μοβ επίπεδο), δημιουργούνται νέα τρίγωνα ώστε να αναπαρασταθεί σωστά το σχήμα και στα 2 μέρη!

Κάθε τεμνόμενο τρίγωνο μετατρέπεται σε 1 τρίγωνο και 1 τετράπλευρο, το οποίο στη συνέχεια τριγωνοποιείται. Έτσι, το μοντέλο χωρίζεται με ακρίβεια σε 2 «υπομοντέλα»: το επάνω και το κάτω. Τέλος, γίνεται μετατόπιση του πάνω τμήματος `{self.model_top.vertices = top_vertices + 0.005 * self.plane[:3]}`, ώστε να είναι διακριτά στην απεικόνιση!

Προστέθηκε το κουμπί E για την απεικόνιση των polygons στο Task 9. Επιπλέον, καθώς μετακινείται το επίπεδο τομής, το μοντέλο προσαρμόζεται δυναμικά!

