



UNIVERSITY OF
PATRAS
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ

Τμήμα πολιτικών Μηχανικών

Μοντελοποίηση παλιρροϊκής
κυκλοφορίας και σχετικών αλγορίθμων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Νικόλαος Ανδρεάκος

Επιβλέπων Καθηγητής:

Γεώργιος Μ. Χορς, Αναπληρωτής Καθηγητής

Πάτρα, 2019

Πρόλογος

Η παρούσα διπλωματική εργασία εκπονήθηκε στα πλαίσια του προπτυχιακού προγράμματος Πολιτικών Μηχανικών, του Πανεπιστημίου Πατρών. Κύριος στόχος της εργασίας είναι η μοντελοποίηση της κυκλοφορίας σε έναν εξιδανικευμένο κόλπο, με κύρια διέγερση ένα παλιρροϊκό κύμα. Ακόμα, αναφέρονται και κάποια άλλα στοιχεία πολύ χρήσιμα για την επιστήμη του υδραυλικού πολιτικού μηχανικού, η φαινόμενη δύναμη Coriolis και το σύστημα Lorenz.

Κύριος, προσωπικός, στόχος μου ήταν η ανάπτυξη λογισμικού, με σύγχρονες προδιαγραφές, όπου να μπορεί να επιλύσει τα φαινόμενα που περιγράφονται στη διπλωματική εργασία επαρκώς, αλλά και ταχέως.

Εκτός από τις γνώσεις που έλαβα κατά τη διάρκεια υλοποίησης του λογισμικού και της συγγραφής της παρούσας διπλωματικής, το υλικό που παρουσιάζεται είναι ιδιαίτερα χρήσιμο σε κάποιον που ξεκινά να μελετά φαινόμενα κυκλοφορίας.

Σε αυτό το σημείο, θα ήθελα να ευχαριστήσω θερμά τον κ. Γεώργιο Χορς, Αναπληρωτή Καθηγητή, για την επιστημονική καθοδήγησή του, καθώς και την πολύτιμη βοήθεια που μου προσέφερε κατά τη διάρκεια εκπόνησης αυτής της εργασίας. Η παρουσία του κρίθηκε απαραίτητη στα τελευταία χρόνια των σπουδών μου. Ακόμα, είναι απαραίτητο να αναφερθώ και να ευχαριστήσω την οικογένειά μου και τους φίλους μου στάθηκαν δίπλα μου και με υποστήριξαν στο έργο μου.

Πάτρα, 2019

Περίληψη

Η παρούσα διπλωματική εργασία διαρθρώνεται σε τρία κύρια μέρη.

Το πρώτο μέρος παρουσιάζεται το σύστημα Lorenz. Αναφέρεται ο τρόπος που ανακαλύφθηκε, η χρήση του στο φυσικό κόσμο, καθώς και προτείνεται μία αριθμητική επίλυσή του. Ακόμα, παρουσιάζεται ένα πρόγραμμα με γραφικό περιβάλλον, που αναπτύχθηκε για τα πλαίσια αυτής της διπλωματικής εργασίας, όπου επιλύεται το σύστημα για διάφορες αρχικές συνθήκες που ορίζει ο χρήστης και εμφανίζει ένα κινούμενο γραφικό αποτέλεσμα.

Στο δεύτερο κεφάλαιο γίνεται αναφορά στη δύναμη Coriolis και πώς επηρεάζει φαινόμενα που εξελίσσονται στη γη. Γίνεται ανάλυση των ιδιοτήτων αυτής της δύναμης και περιγράφεται το αποτέλεσμά της σε μία κίνηση σε περιστρεφόμενο σύστημα αναφοράς. Επίσης, γίνεται αναφορά και σε φαινόμενα που συνήθως η δύναμη Coriolis αποτελεί καθοριστικό παράγοντα, όπως για παράδειγμα τα αδρανειακά κύματα. Τέλος, παρουσιάζεται ένα λογισμικό, που επίσης αναπτύχθηκε στα πλαίσια αυτής της διπλωματικής εργασίας, όπου ο χρήστης μπορεί να ορίσει τις αρχικές συνθήκες και να λάβει μία γραφική απεικόνιση της τροχιάς που θα ακολουθήσει ένα σωματίδιο. Το λογισμικό επιστρέφει, ακόμα, κάποιες χρήσιμες παραμέτρους, όπως ο αριθμός Rossby.

Στο τρίτο και τελευταίο κεφάλαιο μοντελοποιείται η κυκλοφορία σε ένα κόλπο, εξιδανικευμένων διαστάσεων, που διεγείρεται από ένα ημιτονοειδές κύμα, που προσομοιάζει ένα αντίστοιχο παλιρροϊκό κύμα. Γίνεται εκτενής ανάλυση και περιγραφή των εξισώσεων που διέπουν ένα τέτοιο πρόβλημα, καθώς και προτείνεται η αριθμητική του λύση μέσω των πεπερασμένων διαφορών. Τέλος, παρουσιάζονται και αναλύονται τα αποτελέσματα διαφόρων επιλύσεων, σε δύο μεγέθη πλεγμάτων και για διαφορετικό χρόνο εκτέλεσης της επίλυσης.

Περιεχόμενα

Πρόλογος	i
Περίληψη	ii
Περιεχόμενα	iv
Κατάλογος Σχημάτων	v
Κατάλογος Πινάκων	vi
Κατάλογος Αλγορίθμων	vii
1 Οπτικό μοντέλο του συστήματος Lorenz	1
1.1 Το σύστημα Lorenz	1
1.2 Υπολογιστική Επίλυση	2
1.3 Αλγόριθμος επίλυσης	3
1.4 Γραφικό Περιβάλλον	5
1.4.1 Κυρίως παράθυρο	5
2 Δύναμη Coriolis	7
2.1 Παράμετρος Coriolis f	7
2.2 Μετασχηματισμός θέσης σε περιστρεφόμενο σύστημα αναφοράς	8
2.3 Περιγραφή του αλγορίθμου	9
2.4 Δημιουργία λογισμικού	10
2.4.1 Κυρίως παράθυρο	10
2.4.2 Παράθυρο γραφικής παράστασης	12
2.5 Αποτελέσματα και Σχολιασμός	13
2.5.1 Αδρανειακές Ταλαντώσεις	15
2.5.2 Υπολογισμός του αριθμού Rossby	17
3 Μοντελοποίηση κυκλοφορίας σε ανοιχτό από το ένα άκρο κόλπο	18
3.1 Παλίρροια και παλιρροϊκή κυκλοφορία	18
3.2 Εξισώσεις ρηχού στρώματος - SWE	19
3.2.1 Τελικό μοντέλο	20

3.3 Διακριτικοποίηση των εξισώσεων	21
3.3.1 Ευστάθεια της λύσης	22
3.4 Διαδικασία επίλυσης	23
3.4.1 Δεδομένα του προβλήματος	23
3.4.2 Υπολογισμοί	24
3.4.3 Αρχικές και Συνοριακές συνθήκες	25
3.5 Αποτελέσματα επίλυσης	26
3.5.1 Διαγράμματα επιλύσεων	26
3.5.2 Σχολιασμός αποτελεσμάτων	41
Βιβλιογραφία	42
A Αλγόριθμος επίλυσης του συστήματος Lorenz	43
B Υπολογισμός της τροχιάς υπό την επίδραση της δύναμης Coriolis σε Python	45
C Υπολογισμός ανύψωσης / ταπείνωσης ελεύθερης επιφάνειας	50
D Υπολογισμός διανύσματος ταχύτητας	54

Κατάλογος Σχημάτων

1	Γραφική διεπαφή χρήστη του λογισμικού "Interactive Lorenz Attractor".	5
2	Επίλυση συστήματος με αρχικά σημεία $(-5, 0, 10)$ και $(5, -10, 6)$	6
3	Γραφική διεπαφή χρήστη του λογισμικού "Coriolis Trajectory".	11
4	Παράθυρο γραφικής παράστασης του λογισμικού "Coriolis Trajectory" .	13
5	Αποτελέσματα τροχιάς σωματιδίου για μεγάλο και μικρό αριθμό Rossby	14
6	Οι δυνάμεις στο πείραμα του περιστρεφόμενου παραβολικού δίσκου. Teunissen (2017)	15
7	Κάτοψη του κόλπου. Με διακεκομένη γραμμή φαίνεται το ανοιχτό όριο.	24
8	Αποτελέσματα 1a	27
9	Αποτελέσματα 1b	28
10	Αποτελέσματα 2a	29
11	Αποτελέσματα 2b	30
12	Αποτελέσματα 3a	31
13	Αποτελέσματα 3b	32
14	Αποτελέσματα 4a	33
15	Αποτελέσματα 4b	34
16	Αποτελέσματα 5a	35
17	Αποτελέσματα 5b	36
18	Συγκριτικά αποτελέσματα χρονοσειρών 1a-2a	37
19	Συγκριτικά αποτελέσματα χρονοσειρών 1b-2b	38
20	Συγκριτικά αποτελέσματα χρονοσειρών 3a-4a	39
21	Συγκριτικά αποτελέσματα χρονοσειρών 3b-4b	40

Κατάλογος Πινάκων

1	Αλγόριθμος τροχιάς σημείου σε περιστρεφόμενο σύστημα αναφοράς. . .	10
2	Δεδομένα προβλήματος	24
3	Διαφορετικές επιλύσεις του προβλήματος	26

Κατάλογος Αλγορίθμων

2	Αλγόριθμος επίλυσης του συστήματος Lorenz σε Python	43
3	Τροχιά υπό την επίδραση της Coriolis σε Python	45
4	Υπολογισμός ανύψωσης / ταπείνωσης ελεύθερης επιφάνειας	50
5	Υπολογισμός διανύσματος ταχύτητας	54

1 Οπτικό μοντέλο του συστήματος Lorenz

1.1 Το σύστημα Lorenz

Το σύστημα Lorenz, που μελετήθηκε για πρώτη φορά από τον Edward Lorenz στο έργο του *Deterministic nonperiodic flow*, Lorenz (1963), είναι ένα σύστημα που αποτελείται από συνήθεις διαφορικές εξισώσεις και αποτελεί ένα αξιοσημείωτο παράδειγμα χαοτικού προβλήματος.

Για συγκεκριμένες τιμές των παραμέτρων, το σύστημα συμπεριφέρεται χαοτικά. Μικρές μεταβολές στις αρχικές συνθήκες καταλήγουν σε τελείως διαφορετική λύση του προβλήματος, που παραμένει ωστόσο ντετερμινιστικό.

Ο ελκυστήρας του Lorenz, όπως είναι γνωστό το σύστημα, αποτελεί μία απλοποιημένη μορφή ενός συστήματος εξισώσεων που περιγράφουν την διδιάστατη ροή ενός ρευστού, με ομοίομορφο πάχος H και διαφορά θερμοκρασίας ΔT , υπό την επίρεια της βαρυτικής επιτάχυνσης g , με άνωση α , συντελεστή θερμικής μετάδοσης κ και κινηματικό ιξώδες ν , Tabor (1989)

$$\frac{\partial}{\partial t} (\nabla^2 \psi) = \frac{\partial \psi}{\partial z} \frac{\partial}{\partial x} (\nabla^2 \psi) - \frac{\partial \psi}{\partial x} \frac{\partial}{\partial z} (\nabla^2 \psi) + \nu \nabla^2 (\nabla^2 \psi) + g \alpha \frac{dT}{dx} \quad (1.1.1)$$

$$\frac{\partial T}{\partial t} = \frac{\partial T}{\partial z} \frac{\partial \psi}{\partial x} - \frac{\partial \theta}{\partial x} \frac{\partial \psi}{\partial z} + \kappa \nabla^2 T + \frac{\Delta T}{H} \frac{\partial \psi}{\partial x} \quad (1.1.2)$$

όπου ψ η ροϊκή συνάρτηση τέτοια ώστε το διάνυσμα της ταχύτητας $\mathbf{u} = (u, w)$ είναι

$$u = \frac{\partial \psi}{\partial z} \quad (1.1.3)$$

$$w = -\frac{\partial \psi}{\partial x} \quad (1.1.4)$$

Ο Lorenz ανακάλυψε τυχαία ότι σε ένα απλοποιημένο σύστημα, του οποίου η λύση είναι συγκεκριμένης μορφής, παρουσιάζεται χαοτική συμπεριφορά

$$\begin{cases} \frac{dx}{dt} = \sigma (y - x) \\ \frac{dy}{dt} = x (\rho - z) - y \\ \frac{dz}{dt} = xy - \beta z \end{cases} \quad (1.1.5)$$

Το απλοποιημένου του σύστημα συσχετίζει τις ιδιότητες ενός διδιάστατου ρευστού στρώματος, όπου το x είναι ανάλογο του ρυθμού μετάδοσης, το y είναι ανάλογο

της διαφοράς θερμοκρασίας κατά μήκος και το z είναι ανάλογο της κατακόρυφης κατανομής της θερμοκρασίας.

Ακόμα εισήγαγε τις θετικές παραμέτρους σ , ρ που είναι ανάλογες του αριθμού Prandtl και του αριθμού Rayleigh, καθώς και το β που είναι χαρακτηριστικό για των φυσικών διαστάσεων του ρευστού στρώματος.

Στο σύστημα εμφανίζονται τρία κρίσιμα σημεία. Το πρώτο $(0, 0, 0)$ που αντιστοιχεί σε μηδενική μετάδοση και τα

$$\left(\sqrt{\beta(\rho - 1)}, \sqrt{\beta(\rho - 1)}, \rho - 1 \right) \quad (1.1.6)$$

$$\left(-\sqrt{\beta(\rho - 1)}, -\sqrt{\beta(\rho - 1)}, \rho - 1 \right) \quad (1.1.7)$$

που αντιστοιχούν σε σταθερή μετάδοση θερμότητας. Το ζεύγος των σημείων είναι ευσταθές μόνο αν

$$\rho < \sigma \frac{\sigma + \beta + 3}{\sigma - \beta - 1} \Rightarrow \sigma > \beta + 1 \quad (1.1.8)$$

Οι τιμές που ο Lorenz επέλεξε για τις παραμέτρους, για τις οποίες παρουσιάζεται χαοτική συμπεριφορά, είναι

$$\sigma = 10 \quad (1.1.9)$$

$$\beta = 8/3 \quad (1.1.10)$$

$$\rho = 23 \quad (1.1.11)$$

Το σύστημα είναι δύσκολο να αναλυθεί και συνήθως είναι πιο εύκολη η οπτικοποίησή του, μέσω της λύσης του με κάποια υπολογιστική τεχνική. Το οπτικό αποτέλεσμα εμφανίζει τη μορφή μια πεταλούδας ή του αριθμού 8 και αυτός είναι ο λόγος που πήρε το όνομά του το *φαινόμενο της πεταλούδας*, όρο τον οποίο χρησιμοποίησε για πρώτη φορά ο Edward Lorenz.

1.2 Υπολογιστική Επίλυση

Το σύστημα των διαφορικών εξισώσεων μπορεί να λυθεί εύκολα κάνοντας χρήση μίας υπολογιστικής μεθόδου. Για την παρούσα εργασία επιλέχθηκε η επίλυση μέσω της μεθόδου LSODA, κυρίως λόγω της της ευκολίας στην υλοποίηση, αλλά και της δυνατότητας του αλγορίθμου να προσαρμόζεται στο σύστημα και να προσαρμόζει και το βήμα ολοκλήρωσης (*variable step size intergration*).

Για την υπολογιστική επίλυση χρησιμοποιήθηκε η γλώσσα *Python* και το πακέτο αριθμητικής ολοκλήρωσης διαφορικών εξισώσεων *scipy.integrate.odeint*, που χρησιμοποιεί τη βιβλιοθήκη της *LSODA* που προέρχεται από τη *FORTTRAN*. Στο documentation της βιβλιοθήκης αριθμητικής ολοκλήρωσης διαφορικών εξισώσεων *scipy.integrate.odeint* περιγράφεται ότι

Solve a system of ordinary differential equations using lsoda from the FORTRAN library odepack.

Solves the initial value problem for stiff or non-stiff systems of first order ode-s:

$$dy/dt = \text{func}(y, t, \dots) \quad [\text{or } \text{func}(t, y, \dots)]$$

where y can be a vector.

Αντίστοιχα το documentation της *LSODA* αναφέρει ότι

LSODA, written jointly with L. R. Petzold, solves systems $dy/dt = f$ with a dense or banded Jacobian when the problem is stiff, but it automatically selects between nonstiff (Adams) and stiff (BDF) methods. It uses the nonstiff method initially, and dynamically monitors data in order to decide which method to use.

1.3 Αλγόριθμος επίλυσης

```
import random
import numpy as np
from scipy.integrate import odeint
import file_utils

def lorenz(state, t):
    x = state[0]
    y = state[1]
    z = state[2]
```

```
sigma = 10.0
rho = 28.0
beta = 8.0 / 3.0

# compute state derivatives
xd = sigma * (y - x)
yd = (rho - z) * x - y
zd = x * y - beta * z

# return the state derivatives
return [xd, yd, zd]

def get_lorenz(step, ts, tf, x0):
    td = int((tf - ts) / step)
    t = np.linspace(ts, tf, td)
    x_t = np.asarray([odeint(lorenz, x0i, t) for x0i in x0])
    if len(x_t[0]) == td:
        print('New system calculated')
    return x_t

if __name__ == "__main__":
    step = 0.01
    ts = 0
    tf = 100
    num_traj = 2
    x0 = [(random.randint(-25, 25),
            random.randint(-35, 35),
            random.randint(5, 55)) for traj in range(num_traj)]
    x_t = get_lorenz(step, ts, tf, x0)
    file_utils.save_to_file(x_t)
    array = file_utils.open_file()
    print(array == x_t)
```

Κώδικας 1: Lorenz System Solver

1.4 Γραφικό Περιβάλλον

Για τη δημιουργία ενός απλού λογισμικού με γραφική διεπαφή χρήστη (GUI) έγινε χρήση της γλώσσας προγραμματισμού Python. Μέσω της γραφικής διεπαφής ο χρήστης έχει τη δυνατότητα να εισάγει διάφορα αρχικά σημεία και να λάβει μία κινούμενη τριδιάστατη επίλυση του συστήματος Lorenz.

1.4.1 Κυρίως παράθυρο

Στο σχήμα 1 φαίνεται το κυρίως παράθυρο του λογισμικού. Το παράθυρο αποτελείται από δύο βασικά μέρη. Το πρώτο, στα αριστερά, περιλαμβάνει όλες τις βασικές λειτουργίες, καθώς και τις παραμετροποιήσεις που μπορεί να εισάγει στο σύστημα ο χρήστης. Στο δεύτερο μισό, στα δεξιά, εμφανίζεται η γραφική επίλυση του συστήματος, της οποίας την απεικόνιση μπορεί να επεξεργαστεί ο χρήστης.

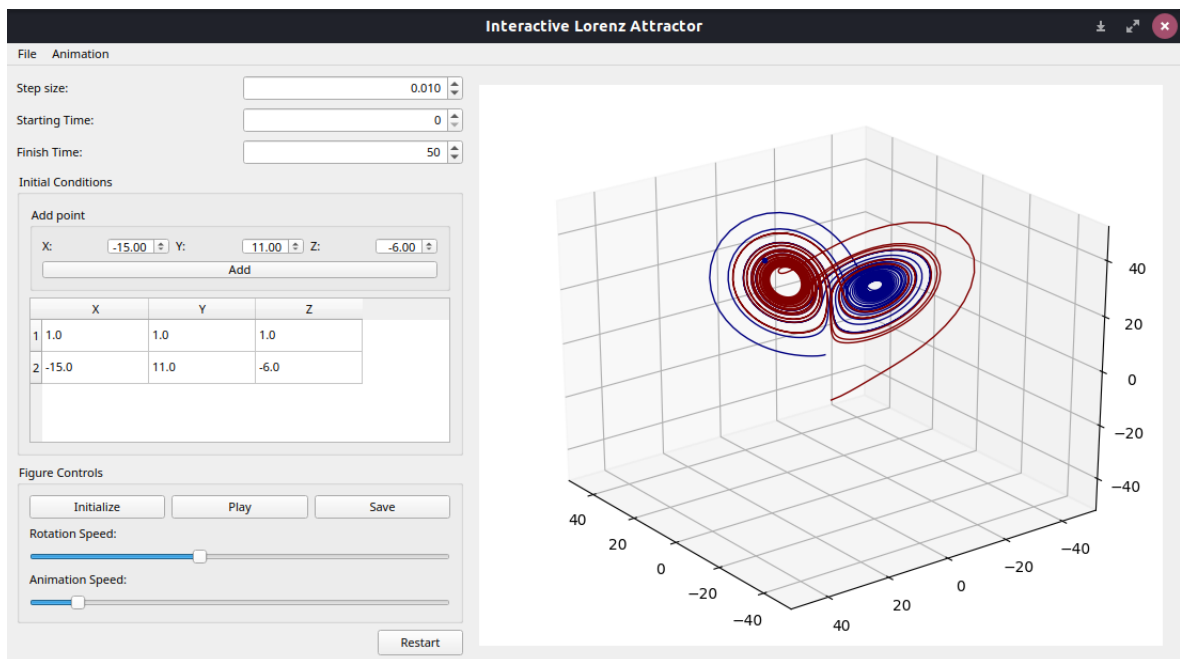


Figure 1: Γραφική διεπαφή χρήστη του λογισμικού "Interactive Lorenz Attractor".

Ο χρήστης επιλέγοντας πρώτα τις συντεταγμένες του αρχικού σημείου, το εισάγει στη λίστα με το κουμπί **Add**. Με διπλό κλικ σε οποιοδήποτε αντικείμενο της λίστας, μπορεί να διαγράψει ένα αρχικό σημείο. Με τα κουμπια **Initialize** και **Play/Stop**, καθώς και με τους κυλιόμενους δείκτες, είναι δυνατή η παραμετροποίηση της εμφάνισης του

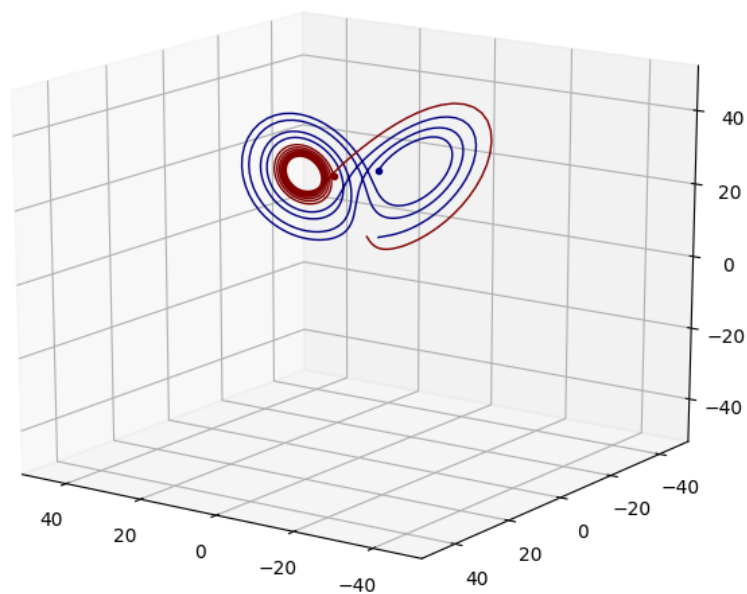


Figure 2: Επίλυση συστήματος με αρχικά σημεία $(-5, 0, 10)$ και $(5, -10, 6)$.

διαγράμματος. Τέλος, προσφέρεται η δυνατότητα αποθήκευσης, ως αρχείο εικόνας, του διαγράμματος, οποιαδήποτε στιγμή, όπως φαίνεται στο σχήμα.

2 Δύναμη Coriolis

Για την κίνηση των σωμάτων θεμελιώθηκαν οι κινηματικοί νόμοι από το Νεύτωνα με το έργο του Principia (Philosophiæ Naturalis Principia Mathematica, 1687). Η νευτώνια κινηματική περιγράφει την κίνηση των αντικειμένων σε ένα αδρανειακό, μη κινούμενο δηλαδή, σύστημα αναφοράς. Όταν εξετάζονται κινήσεις μεγάλης κλίμακας, τέτοιας ώστε η περιστροφή της γης γύρω από τον άξονά της να τις επηρεάζει, η νευτώνια κινηματική δεν μπορεί να τις περιγράψει πλήρως.

Στις εξισώσεις κίνησης ενός σώματος, το οποίο κινείται σε σχέση με ένα περιστρεφόμενο σύστημα αναφοράς, σαν αυτό της περιστρεφόμενης γης, εμφανίζεται μία φαινόμενη δύναμη, η δύναμη Coriolis. Η δύναμη αυτή πήρε το όνομά της από το Γάλλο επιστήμονα Γκασπάρ - Γκουστάβ ντε Κοριόλις (γαλλ. Gaspard-Gustave de Coriolis).

Η επιρροή του περιστρεφόμενου συστήματος στην κίνηση του σώματος μπορεί να προσδιοριστεί υπολογίζοντας τον αριθμό *Rossby*

$$Ro = \frac{U}{Lf} \quad (2.0.1)$$

όπου U και L είναι η χαρακτηριστική ταχύτητα και το χαρακτηριστικό μήκος αντίστοιχα, ενώ f η παράμετρος Coriolis. Ο αριθμός Coriolis είναι στην ουσία ο λόγος των αδρανειακών δυνάμεων προς τις δυνάμεις Coriolis και είναι χαρακτηριστικός για το σύστημα. Οι δυνάμεις Coriolis είναι εμφανείς και ισχυρές και υπερισχύουν των αδρανειακών δυνάμεων, όταν ο αριθμός Rossby είναι μικρός. Το σύστημα επέρχεται σε ισορροπία με τη δράση της δύναμης Coriolis και της βαθμίδας πίεσης. Η ισορροπία αυτή ονομάζεται *γεωστροφική ισορροπία*. Αντίθετα, όταν ο αριθμός Rossby είναι μεγάλος, οι αδρανειακές δυνάμεις υπερισχύουν και το σύστημα έρχεται σε ισορροπία υπό την επίδραση της κεντρομόλου και της φυγόκεντρου δύναμης, όπως ονομάζεται *κυκλοστροφική ισορροπία* (Χαλδούπης, 2015).

2.1 Παράμετρος Coriolis f

Η παράμετρος Coriolis, γνωστή και ως συχνότητα Coriolis, εξαρτάται από την ταχύτητα περιστροφής της γης Ω και το γεωγραφικό πλάτος φ εξέλιξης του φαινομένου. Η καμπυλότητα της επιφάνειας της γης μπορεί να αγνοηθεί για αποστάσεις μικρότερες

των 100 km. Έτσι, τοποθετώντας το σύστημα συντεταγμένων στην επιφάνεια της θάλασσας, με τον άξονα z κατακόρυφα σε αυτή, η παράμετρος Coriolis f μπορεί να θεωρηθεί σταθερή. Η τιμή της παραμέτρου υπολογίζεται ως

$$f = 2\Omega \sin\varphi \quad (2.1.1)$$

Η τιμή αλλάζει πρόσημο κατά τη μετάβαση από το βόρειο προς το νότιο ημισφαίριο, και αντίθετα, ενώ γίνεται 0 στον ισημερινό.

Το όνομα συχνότητα Coriolis προήλθε από τις αδρανειακές ταλαντώσεις, που έχουν συχνότητα ίση με την παράμετρο Coriolis. Οι ταλαντώσεις αυτές ονομάζονται και αδρανειακά κύματα. Τα κύματα αυτά είναι εγκάρσια και παρατηρούνται τόσο στην ατμόσφαιρα (πχ. *γεωστροφικός άνεμος*), όσο και στους ωκεανούς και στις λίμνες (*γεωστροφικά ρεύματα*). Τα κύματα τα οποία επηρεάζονται από την κλίση του πυθμένα ονομάζονται κύματα Rossby.

2.2 Μετασχηματισμός θέσης σε περιστρεφόμενο σύστημα αναφοράς

Θεωρώντας ένα σωματίδιο ακίνητο σε καρτεσιανό αδρανειακό σύστημα αναφοράς, η θέση του περιγράφεται από το διάνυσμα $\mathbf{X} = e_1x_1 + e_2x_2 + e_3x_3$. Σε μητρική μορφή, το διάνυσμα θέσης στο αδρανειακό σύστημα είναι

$$\mathbf{X} = \begin{pmatrix} e_1 & e_2 & e_3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad (2.2.1)$$

Σε ένα άλλο καρτεσιανό σύστημα, το οποίο έχει περιστραφεί *αριστερόστροφα* σε σχέση με το προηγούμενο κατά μία γωνία θ , η θέση του σωματιδίου περιγράφεται από το διάνυσμα

$$\mathbf{X}' = \begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{pmatrix} x_1 \cos\theta + x_2 \sin\theta \\ -x_1 \sin\theta + x_2 \cos\theta \\ x_3 \end{pmatrix} \quad (2.2.2)$$

Παρατηρώντας την (2.2.2), μπορεί να εκλεχθεί ένα μητρώο μετασχηματισμού \mathbb{R} και η θέση του σωματιδίου να περιγράφεται από το διάνυσμα $\mathbf{X}' = \mathbb{R}\mathbf{X}$. Το μητρώο αυτό

είναι

$$\mathbb{R} = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.2.3)$$

2.3 Περιγραφή του αλγορίθμου

Η διαδικασία του μετασχηματισμού της θέσης ενός σωματιδίου, που περιγράφηκε παραπάνω, είναι εύκολο να γίνει αλγοριθμικά. Πρέπει να σημειωθεί ότι η διαδικασία αυτή μπορεί να περιγράψει κάθε είδους κίνηση επί του αδρανειακού επιπέδου, καθώς το μόνο απαραίτητο στοιχείο είναι οι καρτεσιανές συντεταγμένες της θέσης του σωματιδίου, οποιαδήποτε στιγμή. Στην κατάστρωση του αλγορίθμου επιλέχθηκε το σώμα να βάλλεται με μία αρχική ταχύτητα (u, v) , αντίστοιχα σε κάθε άξονα και έπειτα να ακολουθεί την ευθύγραμμη ομαλή κίνηση. Ακόμα επιλέχθηκε το περιστρεφόμενο σύστημα να περιστρέφεται με σταθερή γωνιακή ταχύτητα, παρόλο που η διαδικασία μπορεί να εφαρμοστεί σε κάθε περίπτωση. Ο λόγος που έγιναν αυτές οι παραδοχές είναι ώστε ο αλγόριθμος να περιγράφει και να αναλύει την κίνηση ενός σωματιδίου στο επίπεδο της γης, καθώς αυτή περιστρέφεται γύρω από τον άξονα της.

Όπως είναι γνωστό από τη φυσική η θέση ενός σωματιδίου κάθε χρονική στιγμή i δίνεται αναδρομικά από την προηγούμενη

$$x_i = x_{i-1} + u\Delta t \quad (2.3.1)$$

$$y_i = y_{i-1} + v\Delta t \quad (2.3.2)$$

όπου Δt είναι ένα απειροστό διάστημα χρόνου και (u, v) οι ταχύτητες σε κάθε άξονα.

Έτσι γνωρίζοντας τη θέση του σωματιδίου στο αδρανειακό σύστημα και ακολουθώντας το μετασχηματισμό (2.2.2) γίνεται ο υπολογισμός της θέσης του στο περιστρεφόμενο σύστημα. Ο αλγόριθμος, ο οποίος επιστρέφει τις συντεταγμένες κάθε σημείου της τροχιάς που εκτελεί το σωματίδιο, όπως αυτή αποτυπώνεται στο περιστρεφόμενο σύστημα αναφοράς φαίνεται παρακάτω (βλ. Πίνακα 1).

Table 1: Αλγόριθμος τροχιάς σημείου σε περιστρεφόμενο σύστημα αναφοράς.

1	ΑΡΧΗ
2	$x_0 = \text{αρχική θέση στον άξονα } x$
3	$y_0 = \text{αρχική θέση στον άξονα } y$
4	$\vartheta = 0$
5	Για $t = 1$ έως $t = t_{\text{Τελικό}}$ με_βήμα $= \Delta t$:
6	$\vartheta = \vartheta + \Omega * \Delta t$
7	$x_1 = x_0 + u * \Delta t$
8	$y_1 = y_0 + v * \Delta t$
9	$x_{\text{prime1}} = x_1 * \cos(\vartheta) + y_1 * \sin(\vartheta)$
10	$y_{\text{prime1}} = -x_1 * \sin(\vartheta) + y_1 * \cos(\vartheta)$
11	$\text{τύπωσε}(x_{\text{prime}}, y_{\text{prime}})$
12	$x_0 = x_1$
13	$y_0 = y_1$
14	τέλος Για
15	ΤΕΛΟΣ

2.4 Δημιουργία λογισμικού

Για τη δημιουργία ενός απλού λογισμικού με γραφική διεπαφή χρήστη (GUI) έγινε χρήση της γλώσσας προγραμματισμού Python. Μέσω της γραφικής διεπαφής ο χρήστης έχει τη δυνατότητα να παράξει γραφικές παραστάσεις της τροχιάς του σωματιδίου, έχοντας παράλληλα πλήρη έλεγχο επί των παραμέτρων. Ο κώδικας που υλοποιεί τον υπολογισμό παρατίθεται στο παράρτημα Β.

2.4.1 Κυρίως παράθυρο

Στο σχήμα 3 φαίνεται το κύριο παράθυρο του λογισμικού. Στην αριστερή μεριά του παραθύρου ο χρήστης έχει τη δυνατότητα να παραμετροποιήσει τη λύση, ενώ στη δεξιά πλευρά εμφανίζονται μερικές χρήσιμες χαρακτηριστικές τιμές καθώς και οι λειτουργίες ελέγχου των παραγόμενων γραφημάτων. Κατά σειρά αρίθμησης

1. Ορισμός του αριθμού βημάτων επίλυσης

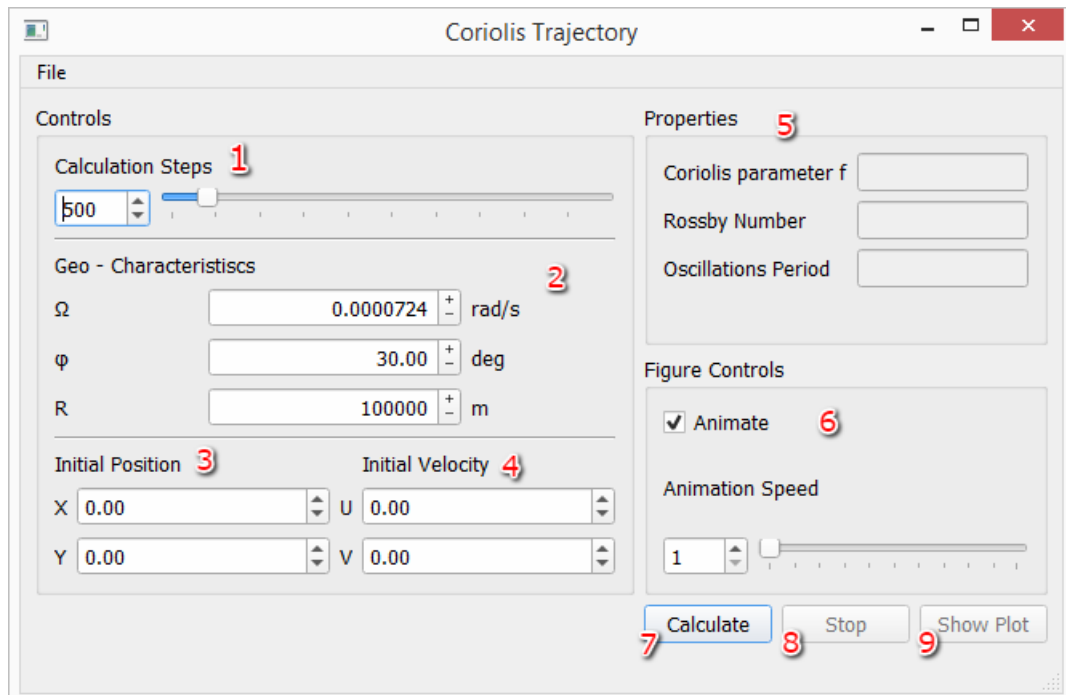


Figure 3: Γραφική διεπαφή χρήστη του λογισμικού "Coriolis Trajectory".

2. Χαρακτηριστικά που αφορούν το επίπεδο επίλυσης

Ω Ο ρυθμός περιστροφής του επιπέδου, $7.24e^{-5}rad/s$ για το ρυθμό περιστροφής της γης γύρω από τον άξονά της

φ Το γεωγραφικό πλάτος εξέλιξης του φαινομένου. Επιτρέπονται τιμές από -90° έως 90° . Για τα σωματίδια που εκτελούν κίνηση στον ισημερινό, δηλαδή για $\varphi = 90^\circ$ δεν ορίζεται η δύναμη Coriolis και το πρόγραμμα ειδοποιεί το χρήστη με σχετικό μήνυμα. Ανάλογα με το πρόσημο \pm του γεωγραφικού πλάτους, στο βόρειο ή νότιο ημισφαίριο, το λογισμικό επιλέγει αυτόματα το πρόσημο της γωνιακής ταχύτητας, έτσι ώστε τα αποτελέσματα να έχουν και φυσική υπόσταση.

R Η ακτίνα του περιστρεφόμενου δακτυλίου. Προτείνεται να μην ξεπερνάει τα 100km, καθώς αυτή είναι η οριακή τιμή για την οποία μπορούμε να θεωρήσουμε σταθερή τη συχνότητα Coriolis f .

3. Αρχική θέση του σωματιδίου

4. Αρχική ταχύτητα του σωματιδίου, στις συνιστώσες (u, v) του καρτεσιανού επιπέδου

5. Χαρακτηριστικά της κίνησης

Παράμετρος Coriolis f

Η συχνότητα Coriolis

Αριθμός Rossby

Ο αριθμός Rossby , χαρακτηριστικός του φαινομένου.

Συχνότητα αδρανειακών ταλαντώσεων

Η συχνότητα της ταλάντωσης που εκτελεί ένα σωματίδιο ρευστού, κατά την οποία οι αδρανειακές δυνάμεις που του ασκούνται εξισορροπούνται πλήρως από τη δύναμη Coriolis .

6. Επιλογή για την απεικόνιση των αποτελεσμάτων σε μορφή βίντεο (animation).
7. Το βασικό κουμπί εκτέλεσης του προγράμματος. Ελέγχει τα δεδομένα και στο τέλος των υπολογισμών εμφανίζει τη γραφική παράσταση του αποτελέσματος
8. Στην περίπτωση που οι υπολογισμοί διαρκούν πολύ ώρα, δίνεται στο χρήστη δυνατότητα τερματισμού
9. Εμφανίζει τη γραφική παράσταση, σύμφωνα με τις επιλογές του χρήστη στο (6), χωρίς την επανέναρξη του υπολογισμού

Πρέπει να σημειωθεί ότι έχει υλοποιηθεί προγραμματισμός που κάνει χρήση παραπάνω τους ενός πυρήνα του επεξεργαστή, δίνοντας στο χρήστη τη δυνατότητα να υπολογίσει και να εμφανίσει ταυτόχρονα πολλά διαφορετικά σενάρια επίλυσης.

2.4.2 Παράθυρο γραφικής παράστασης

Το δεύτερο παράθυρο του λογισμικού εμφανίζει γραφικά τα αποτελέσματα των υπολογισμών. Τα αποτελέσματα εμφανίζονται είτε με τη μορφή βίντεο, είτε σταθερά, αναλόγως με την επιλογή του χρήστη. Ακόμα, υπάρχει η δυνατότητα εξαγωγής και αποθήκευσης του γραφήματος σε ένα αρχείο εικόνας. Στο παράθυρο φαίνονται

1. Το κυρίως γράφημα

Μαύρη γραμμή

Τα όρια του περιστρεφόμενου κυκλικού δίσκου

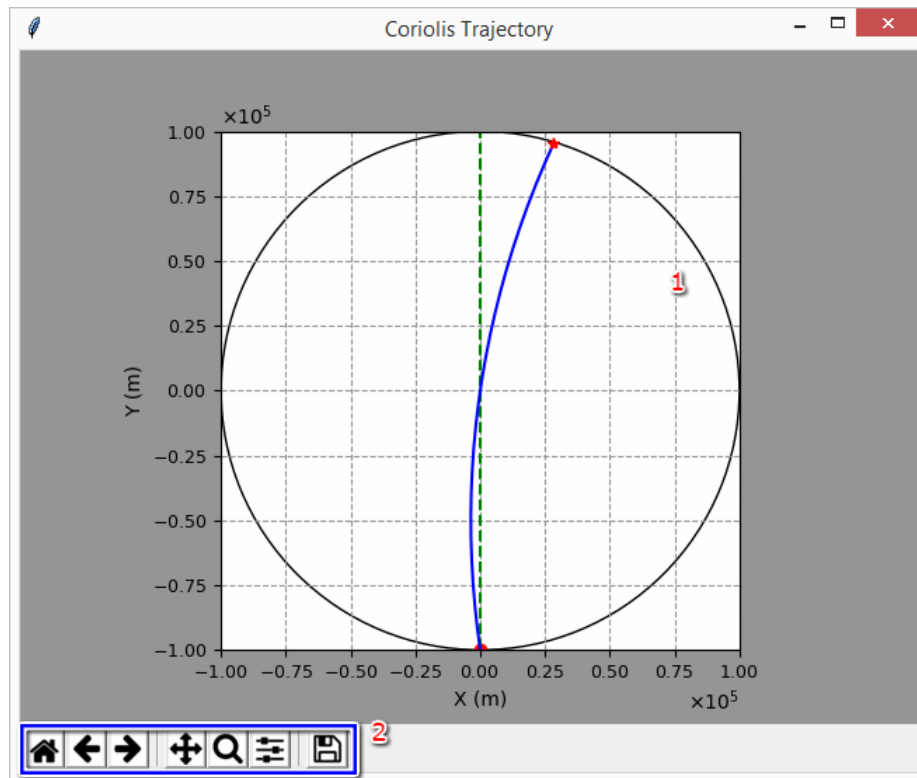


Figure 4: Παράθυρο γραφικής παράστασης του λογισμικού "Coriolis Trajectory"

Πράσινη γραμμή

Η τροχιά του σωματιδίου στο αδρανειακό σύστημα αναφοράς

Μπλε γραμμή

Η τροχιά του σωματιδίου στο περιστρεφόμενο σύστημα αναφοράς

Κόκκινος κύκλος

Η αρχική θέση του σωματιδίου

Κόκκινο αστέρι

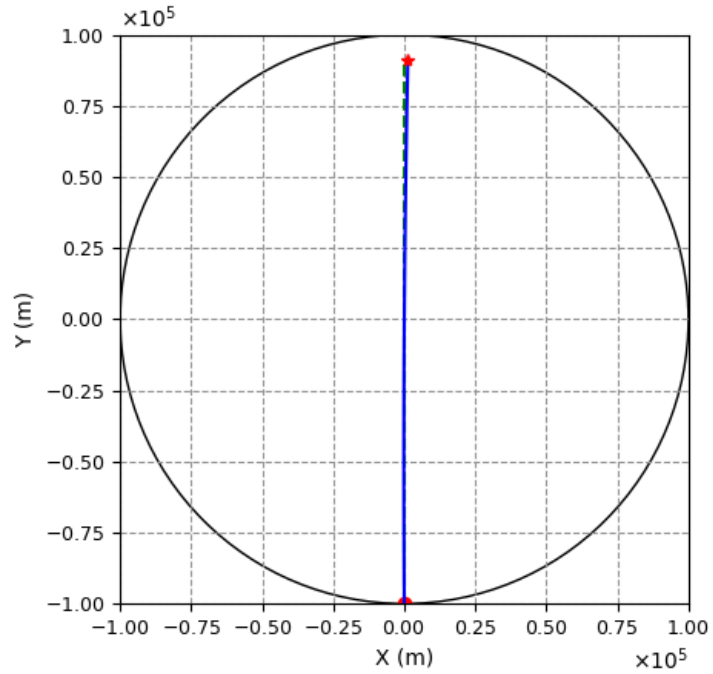
Η τελική θέση του σωματιδίου

2. Επιλογές για την παραμετροποίηση της εμφάνισης του γραφήματος, καθώς και αποθήκευσής του σε αρχείο εικόνας

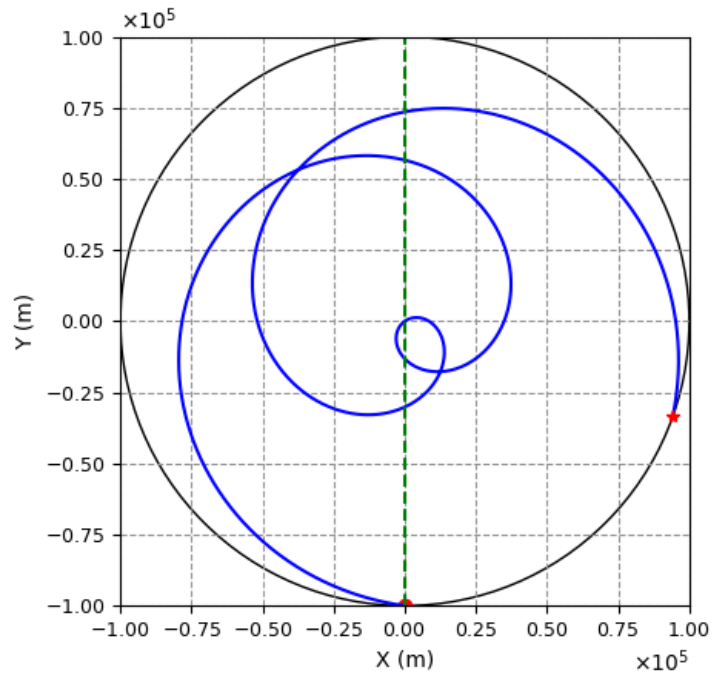
2.5 Αποτελέσματα και Σχολιασμός

Το λογισμικό επιστρέφει στο χρήστη μερικές χρήσιμες τιμές, ώστε αυτός να μπορεί να αξιολογήσει τα αποτελέσματα. Μία από αυτές είναι ο αριθμός Rossby . Ο αριθμός

αυτός αποτελεί τον αδιάστατο λόγο των αδρανειακών δυνάμεων προς τις δυνάμεις Coriolis.



(a) Μεγάλος αριθμός Rossby, $R = 1.38 \cdot 10^2$



(b) Μικρός αριθμός Rossby, $R = 1.38 \cdot 10^{-1}$

Figure 5: Αποτελέσματα τροχιάς σωματιδίου για μεγάλο και μικρό αριθμό Rossby

Η τάξη μεγέθους αυτού του αριθμού δηλώνει κατά πόσο η τροχιά ενός σωματιδίου επηρεάζεται από την περιστροφή του συστήματος συντεταγμένων, όπως για παράδειγμα όταν ένα σωματιδίου που εκτελεί μία κίνηση επηρεάζεται από την περιστροφή της γης γύρω από τον άξονα της. Όταν η τιμή του αριθμού Rossby είναι αρκετά μεγάλη, δηλαδή οι αδρανειακές δυνάμεις υπερσχύουν των δυνάμεων Coriolis και το σύστημα έρχεται σε κυκλοστροφική ισορροπία, φαίνεται ότι η τροχιά του σωματιδίου δεν επηρεάζεται από την περιστροφή του συστήματος, σχήμα 5(a). Στην αντίθετη περίπτωση που ο αριθμός Rossby είναι μικρός, το σύστημα έρχεται σε γεωστροφική ισορροπία και οι δυνάμεις Coriolis υπερσχύουν των αδρανειακών δυνάμεων, η τροχιά του σωματιδίου επηρεάζεται σε μεγάλο βαθμό από την περιστροφή του συστήματος συντεταγμένων, σχήμα 5(b).

2.5.1 Αδρανειακές Ταλαντώσεις

Οι αδρανειακές ταλαντώσεις είναι η πιο απλή μορφή κίνησης στη γη, η οποία επηρεάζεται από τη χρονική της διάρκεια. Στην πραγματικότητα, τέτοιου είδους κίνηση δεν εμφανίζεται μόνο στην περίπτωση της περιστρεφόμενης γης. Ένα απλό παράδειγμα για την κατανόηση των δυνάμεων που επηρεάζουν τέτοιες κινήσεις, είναι το παράδειγμα του περιστρεφόμενου παραβολικού δίσκου.

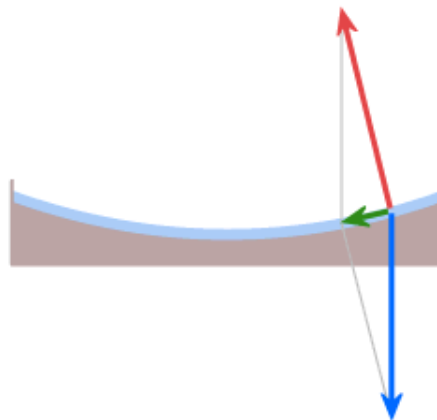


Figure 6: Οι δυνάμεις στο πείραμα του περιστρεφόμενου παραβολικού δίσκου. Teunissen (2017)

Στο πείραμα του περιστρεφόμενου παραβολικού δίσκου, Σχήμα 6, αφήνεται ένα σώμα να εκτελέσει μία ελεύθερη ταλάντωση στην άτριβη επιφάνεια. Η δύναμη επαναφοράς είναι ανάλογη της απόστασης από το κέντρο της παραβολής.

Παρατηρώντας αυτήν την κίνηση από ένα αδρανειακό σύστημα, φαίνεται ότι το σώμα εκτελεί μία ελλειπτική κίνηση, η οποία ονομάζεται αδρανειακή ταλάντωση.

Στο φυσικό κόσμο, η δύναμη επαναφοράς είναι η φαινόμενη δύναμη Coriolis. Η περιστροφή της γης επηρεάζει τις κινήσεις που είναι πολύ "αργές" σε σχέση με την περιστροφή της, όπως τα κύματα με μεγάλο μήκος και μικρή συχνότητα, τα αδρανειακά κύματα (inertial waves).

Τα αδρανειακά κύματα είναι εγκάρσια μηχανικά κύματα, που σε αντίθεση με τα επιφανειακά, λαμβάνουν χώρα στα εσωτερικά στρώματα ενός ρευστού. Τέτοια κύματα εμφανίζονται συνήθως στην ατμόσφαιρα της γης, παραδείγματος χάρη τα κύματα Rossby και τα γεωστροφικά ρεύματα.

Τέτοια κύματα συναντώνται και στον ωκεανό, με τη μορφή κυμάτων Poincare. Κύμα Poincare μπορεί να θεωρηθεί και η διάδοση της παλίρροιας, καθώς και όλα τα κύματα ρηχού στρώματος (SWW), μακριά από τη στεριά.

Η συχνότητα των επιφανειακών κυμάτων Poincare είναι ίδια με την συχνότητα κυμάτων ρηχού στρώματος, με την προσθήκη της παραμέτρου Coriolis, f .

$$\omega = \sqrt{f^2 + gHk^2} \quad (2.5.1)$$

Η ταχύτητα ομάδας (Group velocity) είναι

$$C_g = \frac{\partial \omega}{\partial k} = \frac{gHk}{\sqrt{f^2 + gHk^2}} \quad (2.5.2)$$

Όταν $f \rightarrow 0$ ή $k^2 \gg \frac{f^2}{gH}$ τότε $C_g = \sqrt{gH}$ και το κύμα συμπεριφέρεται σαν απλό κύμα ρηχού στρώματος, χωρίς της επιρροή της περιστροφής της γης. Στην αντίθετη περίπτωση, όπου $k \rightarrow 0$ τότε $C_g = 0$, δεν υπάρχει διάδοση του κύματος και εκτελείται μία αδρανειακή ταλάντωση.

Η επίλυση στη μία διάσταση ενός κύματος Poincare είναι

$$u = \frac{\omega}{kH} H_0 \cos(kx - \omega t) \quad (2.5.3)$$

$$v = \frac{f}{kH} H_0 \sin(kx - \omega t) \quad (2.5.4)$$

Εύκολα, μπορούμε να υπολογίσουμε πότε η περιστροφή της γης είναι σημαντική. Η χαρακτηριστική κλίμακα μήκους είναι

$$L \rightarrow 1/k \Rightarrow L_R = \frac{\sqrt{gH}}{H} \quad (2.5.5)$$

και ονομάζεται **Rossby radius of deformation**. Για μήκος μικρότερο από την τιμή της κλίμακας η περιστροφή δεν επηρεάζει την κίνηση.

2.5.2 Υπολογισμός του αριθμού Rossby

Ο αριθμός Rossby αποτελεί έναν αδιάστατο αριθμό χαρακτηριστικό για το κάθε πρόβλημα, ο δε υπολογισμός του είναι εύκολος κάνοντας χρήση και ανάλυση των εγγενών κλιμάκων του προβλήματος.

Έχοντας ως βασική κλίμακα το **χρόνο**, μπορεί να προσδιοριστεί τόσο ο χρόνος που απαιτείται ώστε ένα σημείο επάνω στο όριο του κύκλου που περιγράφει το περιστρεφόμενο σύστημα αναφοράς να κάνει μία πλήρη περιστροφή, όσο και ο χρόνος που απαιτείται ώστε ένα σωματίδιο, που εκτελεί την εξεταζόμενη κίνηση, να διαγράψει μία διαμετρική πορεία.

Ο χρόνος που απαιτείται ώστε ένα σημείο του κύκλου να διαγράψει μία πλήρη περιστροφή είναι

$$t_R = \frac{2\pi}{\Omega} \quad (2.5.6)$$

ενώ ο χρόνος που απαιτείται ώστε ένα σωματίδιο να εκτελέσει ευθύγραμμη ομαλή κίνηση και να διαγράψει απόσταση ίση με τη διάμετρο του κυκλικού δίσκου είναι

$$t_I = \frac{2R}{V} \quad (2.5.7)$$

όπου Ω είναι η σταθερή γωνιακή ταχύτητα του κυκλικού δίσκου, R η διάμετρός του και V η χαρακτηριστική ταχύτητα της ευθύγραμμης κίνησης.

Συνδυάζοντας τις 2.5.6 και 2.5.7, που έχουν ως κλίμακα το χρόνο, $[T]$, προκύπτει ο αδιάστατος αριθμός Rossby

$$Ro = \frac{t_R}{t_I} \Rightarrow Ro = \frac{\frac{2\pi}{\Omega}}{\frac{2R}{V}} \quad (2.5.8)$$

Κάνοντας τις απλοποιήσεις και απαλοίζοντας το σταθερό όρο π , προκύπτει ο αριθμός Rossby που χρησιμοποιήθηκε στο λογισμικό

$$Ro = \frac{V}{\Omega R} \quad (2.5.9)$$

3 Μοντελοποίηση κυκλοφορίας σε ανοιχτό από το ένα άκρο κόλπο

Στο κεφάλαιο αυτό παρουσιάζεται η μοντελοποίηση της κυκλοφορίας σε έναν εξιδανικευμένο κόλπο, όπου το ένα άκρο του είναι ανοιχτό (open-sea-boundary), ενώ τα υπόλοιπα κλειστά (στεριά).

Ο κόλπος έχει εξιναδικευτεί ώστε να έχει κάτοψη ορθογωνίου, με σταθερό βάθος και άτριβο πυθμένα. Το μοντέλο περιγράφει τη διάδοση ενός παλιρροϊκού κύματος από την ανοιχτή θάλασσα και το κλειστό άκρο, στο οποίο προσπίπτει το κύμα, προκαλεί τέλεια ανάκλαση.

3.1 Παλίρροια και παλιρροϊκή κυκλοφορία

Το φαινόμενο της παλίρροιας αναφέρεται στην περιοδική κίνηση του νερού της θάλασσας και οφείλεται στην διαφορά των ελκτικών δυνάμεων του Ηλίου και της Σελήνης, στις διάφορες φάσεις της. Ως παλίρροια αναφέρεται η περιοδική ανύψωση και ταπείνωση της στάθμης της θάλασσας και συνοδεύεται από το παλλιροϊκό ρεύμα, δηλαδή την οριζόντια κίνηση του νερού. Και οι δύο κινήσεις αποτελούν μέρη του ίδιου φαινομένου.

Η περιοδική ανύψωση και ταπείνωση της στάθμης της θάλασσας είναι επί της ουσίας ένα μακρύ κύμα το οποίο διαδίδεται. Η περίοδος ενός τέτοιου κύματος είναι συγκρίσιμη με το χρόνο περιστροφής της γης, αφού η παλίρροια μπορεί να έχει περίοδο 6, 12, ακόμα και 24 ώρες, οπότε επηρεάζεται από αυτή. Για αυτό στις εξισώσεις ορμής είναι απαραίτητη η προσθήκη του όρου της δύναμης Coriolis, εφόσον το φαινόμενο εξελίσσεται σε μεταξύ διαφορετικών γεωγραφικών πλατών.

Ένα μακρύ κύμα περιγράφεται από την εξίσωση

$$\eta = \frac{H}{2} \cos(kx - \omega t) \quad (3.1.1)$$

όπου H το πλάτος κύματος, $k = \frac{2\pi}{L}$ ο κυματαριθμός, $\omega = \frac{2\pi}{T}$ η κυκλική συχνότητα, L το μήκος κύματος και T η περίοδος.

3.2 Εξισώσεις ρηχού στρώματος - SWE

Πολλοί τύποι ροών μπορούν να χαρακτηριστούν ως "ροές ρηχού στρώματος". Με τον όρο ροή δεν είναι απαραίτητη η παραπομπή μόνο σε ροή ύδατος. Ροές όπως η κυκλοφορία ατμοσφαιρικών μαζών, η παλίρροια, πλημυρικά ρέματα, η παράκτια κυκλοφορία, τα tsunamis κ.ά., μπορούν να περιγραφούν με τις εξισώσεις ρηχού στρώματος. Το βασικό χαρακτηριστικό τέτοιων ροών είναι ότι η κλίμακα του βάθους είναι πολύ μικρότερη από την οριζόντια χαρακτηριστική κλίμακα.

$$\frac{h}{L} \ll 1 \quad (3.2.1)$$

Επομένως το αν θα εμπίπτει μία ροή στην κατηγορία εξισώσεων ρηχού στρώματος έχει σχέση με τις γεωμετρικές διαστάσεις της ίδιας της ροής και όχι του ρευστού όγκου που εξετάζεται. Για παράδειγμα η παλιρροϊκή ροή σε έναν ωκεανό περιγράφεται από τις εξισώσεις ρηχού στρώματος, όμως η ανεμογενής κυκλοφορία σε έναν ταμιευτήρα όχι, παρόλο που το βάθος του ταμιευτήρα είναι πολύ μικρό.

Στο μοντέλο του ρηχού στρώματος υπεισέρχεται η βασική απλοποίηση ότι η κατακόρυφη συνιστώσα της ταχύτητας, οπότε και της επιτάχυνσης είναι αμελητέες. Έτσι, η κατακόρυφη εξίσωση ορμής αποτελείται από υδροστατική ισορροπία.

Το σύστημα των διαφορικών εξισώσεων που περιγράφουν το πρόβλημα είναι οι εξισώσεις ορμής Navier - Stokes και η εξίσωση συνέχειας. Ύστερα από ανάλυση της κλίμακας κάθε όρου των εξισώσεων και απαλοιφής των ασήμαντων, λόγω κλίμακας, όρων οι εξισώσεις ορμής για τους άξονες x, y, z είναι αντίστοιχα

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} = f v - \frac{1}{\rho} \frac{\partial p}{\partial x} + \frac{\partial}{\partial z} \left(\varepsilon_z \frac{\partial u}{\partial z} \right) \quad (3.2.2)$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} = f u - \frac{1}{\rho} \frac{\partial p}{\partial y} + \frac{\partial}{\partial z} \left(\varepsilon_z \frac{\partial v}{\partial z} \right) \quad (3.2.3)$$

$$0 = -\frac{1}{\rho} \frac{\partial p}{\partial z} - g \quad (3.2.4)$$

Το σύστημα συμπληρώνεται με την εξίσωση συνέχειας

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \quad (3.2.5)$$

Η ύπαρξη ελεύθερης επιφάνειας του νερού εισάγει έναν ακόμα άγνωστο στο πρόβλημα της κυκλοφορίας. Η ανάπτυξη οποιουδήποτε ρεύματος συνεπάγεται τη

μετατόπιση της ελεύθερης επιφάνειας από τη θέση ισορροπίας της, με αποτέλεσμα το πεδίο ορισμού της λύσης να είναι άγνωστο.

Ένας τρόπος αντιμετώπισης αυτού του προβλήματος είναι να δημιουργήσουμε μία εξίσωση που να περιγράφει την θέση της ελεύθερης επιφάνειας κάθε χρονική στιγμή. Για το λόγο αυτό ολοκληρώνεται ως προς το βάθος η εξίσωση συνέχειας.

Έστω ότι με $\zeta(t, x, y)$ συμβολίζεται το τοπικό βάθος της υδάτινης μάζας. Τότε η ολοκληρωμένη ως προς το βάθος εξίσωση συνέχειας δίνει

$$\frac{\partial \zeta}{\partial t} + \frac{\partial}{\partial x} (\zeta U) + \frac{\partial}{\partial y} (\zeta V) = 0 \quad (3.2.6)$$

όπου U και V οι μέσες ως προς το βάθος τιμές των u και v αντίστοιχα.

Ακόμα ολοκληρώνοντας την υδροστατική εξίσωση ως προς το βάθος, καταλήγουμε

$$-\frac{1}{\rho} \frac{\partial p}{\partial x} = -g \frac{\partial \eta}{\partial x} - \frac{1}{\rho} \frac{\partial p_\alpha}{\partial x} \quad (3.2.7)$$

$$-\frac{1}{\rho} \frac{\partial p}{\partial y} = -g \frac{\partial \eta}{\partial y} - \frac{1}{\rho} \frac{\partial p_\alpha}{\partial y} \quad (3.2.8)$$

Μετά από τις κατάλληλες αντικαταστάσεις στο αρχικό σύστημα καταλήγουμε στις εξισώσεις ορμής, που μαζί με την ολοκληρωμένη ως προς το βάθος εξίσωση συνέχειας, αποτελούν τη βάση του μοντέλου ρηχού στρώματος

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} = f v - g \frac{\partial \eta}{\partial x} - \frac{1}{\rho} \frac{\partial p_\alpha}{\partial x} + \frac{\partial}{\partial z} \left(\varepsilon_z \frac{\partial u}{\partial z} \right) \quad (3.2.9)$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} = f u - g \frac{\partial \eta}{\partial y} - \frac{1}{\rho} \frac{\partial p_\alpha}{\partial y} + \frac{\partial}{\partial z} \left(\varepsilon_z \frac{\partial v}{\partial z} \right) \quad (3.2.10)$$

3.2.1 Τελικό μοντέλο

Το μοντέλο που εξετάζεται στην παρούσα διπλωματική είναι αρκετά απλοποιημένο και δεν είναι απαραίτητη η χρήση όλων των όρων των εξισώσεων. Οι μη γραμμικοί όροι στις παραπάνω εξισώσεις μεταθέτουν μεν την ορμή, χωρίς όμως να μεταβάλλουν την ολική τιμή της. Καθώς η κυκλοφορία καθορίζεται από την ισορροπία των δυνάμεων, η μορφή της διατηρείται και στη γραμμικοποιημένη μορφή των εξισώσεων. Ακόμα, θεωρούμε ότι ο πυθμένας του κόλπου είναι άτριβος, η ατμοσφαιρική πίεση είναι σταθερή σε όλη την κάτοψη, καθώς και η εξέλιξη του φαινομένου γίνεται στο ίδιο γεωγραφικό πλάτος, με αποτέλεσμα η δύναμη Coriolis να μην επηρεάζει τη λύση.

Με βάση όλες αυτές τις παραδοχές, οι εξισώσεις 3.2.9, 3.2.10 και 3.2.6 γίνονται

$$\frac{\partial u}{\partial t} = -g \frac{\partial \eta}{\partial x} \quad (3.2.11)$$

$$\frac{\partial v}{\partial t} = -g \frac{\partial \eta}{\partial y} \quad (3.2.12)$$

$$\frac{\partial \eta}{\partial t} + h \frac{\partial U}{\partial x} + h \frac{\partial V}{\partial y} = 0 \quad (3.2.13)$$

Μετά από παραγωγή των εξισώσεων 3.2.11 και 3.2.12 ως προς x και y αντίστοιχα και παραγωγή της 3.2.13 ως προς t , καταλήγουμε στην εξίσωση κύματος

$$\frac{\partial^2 \eta}{\partial t^2} = gh \left(\frac{\partial^2 \eta}{\partial x^2} + \frac{\partial^2 \eta}{\partial y^2} \right) \quad (3.2.14)$$

η οποία είναι μια υπερβολική διαφορική εξίσωση της μορφής $\frac{\partial^2 \vec{u}}{\partial t^2} = c^2 \nabla^2 \vec{u}$, με $\vec{u} = \vec{u}(x_1, x_2, \dots, x_n)$ βαθμωτή συνάρτηση της οποίας οι τιμές αποτελούν τη μετατόπιση ενός κύματος. Γίνεται αμέσως φανερό ότι η ταχύτητα του κύματος είναι

$$c = \sqrt{gh} \quad (3.2.15)$$

3.3 Διακριτοποίηση των εξισώσεων

Η αριθμητική επίλυση του προβλήματος γίνεται διακριτοποιώντας τις εξισώσεις (3.2.11), (3.2.12) και (3.2.14). Επιλέχθηκε να γίνει χρήση της μεθόδου *πεπερασμένων διαφορών*, έναντι της μεθόδου πεπερασμένων στοιχείων, προς χάρη της απλότητάς της. Βασιζόμενοι στη λογική της απλότητας χρησιμοποιήθηκε ένα ρητό σχήμα, διακριτοποιώντας το χρόνο με πρόσω διαφορές και το χώρο με κεντρικές, όπως συνιθίζεται σε φαινόμενα μεταφοράς.

Έτσι η διακριτοποιημένη μορφή της εξίσωσης κύματος (3.2.14) είναι

$$\eta_{i,j}^{n+1} = 2\eta_{i,j}^n - \eta_{i,j}^{n-1} + \frac{c^2 \Delta t^2}{\Delta x^2} (\eta_{i+1,j}^n - 2\eta_{i,j}^n + \eta_{i-1,j}^n) + \frac{c^2 \Delta t^2}{\Delta y^2} (\eta_{i,j+1}^n - 2\eta_{i,j}^n + \eta_{i,j-1}^n) \quad (3.3.1)$$

όπου c είναι η ταχύτητα κύματος, όπως υπολογίζεται στην εξίσωση (3.2.15).

Για λόγους ευπαρουσίασης της εξίσωσης (3.3.1), συμβολίζουμε ως

$$H_x = \eta_{i+1,j}^n - 2\eta_{i,j}^n + \eta_{i-1,j}^n \quad (3.3.2)$$

$$H_y = \eta_{i,j+1}^n - 2\eta_{i,j}^n + \eta_{i,j-1}^n \quad (3.3.3)$$

$$C_{xs} = \frac{c^2 \Delta t^2}{\Delta x^2} \quad (3.3.4)$$

$$C_{ys} = \frac{c^2 \Delta t^2}{\Delta y^2} \quad (3.3.5)$$

οπότε η (3.3.1) γίνεται

$$\eta_{i,j}^{n+1} = 2\eta_{i,j}^n - \eta_{i,j}^{n-1} + C_{xs}H_x + C_{ys}H_y \quad (3.3.6)$$

Οι αριθμοί (3.3.4) και (3.3.4), είναι το τετράγωνο του αριθμού Courant για κάθε μία από τις δύο διευθύνσεις.

Για τη διακριτοποίηση των διαφορικών εξισώσεων της ταχύτητας (3.2.11) και (3.2.12) έγινε χρήση του ίδιου σχήματος, επομένως οι διακριτοποιημένες εξισώσεις, χρησιμοποιώντας τον παραπάνω συμβολισμό είναι

$$u_{i,j}^{n+1} = u_{i,j}^n - g \frac{\Delta t}{\Delta x^2} H_x \quad (3.3.7)$$

$$v_{i,j}^{n+1} = v_{i,j}^n - g \frac{\Delta t}{\Delta y^2} H_y \quad (3.3.8)$$

3.3.1 Ευστάθεια της λύσης

Όπως κάθε ρητό σχήμα επίλυσης, έτσι και αυτό που αναπτύχθηκε παραπάνω, είναι απαραίτητη η ύπαρξη μιας σχέσης που συνδέει το χρονικό βήμα, με το χωρικό βήμα.

Σύμφωνα με τον περιορισμό Courant-Friedrichs-Lewy (**CFL condition**) το χρονικό βήμα πρέπει να έχει τιμή μικρότερη από ένα συγκεκριμένο χρονικό διάστημα ώστε η αριθμητική λύση μιας υπερβολικής διαφορικής εξίσωσης να συγκλίνει.

Η μορφή του περιορισμού CFL έχει σχέση με το πεδίο ορισμού της εξίσωσης που πρόκειται να λυθεί. Στο πρόβλημα της κυκλοφορίας σε ένα κόλπο, η επίλυση γίνεται στις δύο οριζόντιες διαστάσεις x , y και η υπόθεση είναι

$$C = c\Delta t \frac{\Delta x + \Delta y}{\Delta x \Delta y} \quad (3.3.9)$$

που προέκυψε από πρόσθεση του αριθμού Courant για κάθε διεύθυνση.

Από φυσική άποψη, ο περιορισμός CFL αναφέρεται στο γεγονός ότι το χρονικό βήμα πρέπει να είναι μικρότερο από το χρόνο που χρειάζεται το κύμα για να διασχίσει ένα

χωρικό βήμα. Όταν η επίλυση του κύματος γίνεται στη μία διάσταση, αρκεί $C \leq 1$. Στην περίπτωση όμως που η επίλυση γίνεται στις δύο διαστάσεις, όπως στην παρούσα διπλωματική εργασία, ο περιορισμός γίνεται πιο αυστηρός και πρέπει $C \leq \frac{1}{\sqrt{2}}$. Η αυστηρότητα του περιορισμού έγκυται στο γεγονός ότι όταν ένα κύμα κινείται υπό γωνία στο πλέγμα επίλυσης πρέπει να διανύσει τη διαγώνια απόσταση ενός κελιού. Αυτό συχνά αναφέρεται και ως *αριθμητικό ιξώδες* ή *αριθμητική διάχυση*.

Με βάση λοιπόν τα παραπάνω, υπάρχει ένας περιορισμός για το μέγιστο χρονικό βήμα που είναι δυνατό να χρησιμοποιηθεί κατά την επίλυση και αυτό έχει άμεση σχέση με το χωρικό βήμα που επιλέγεται σε κάθε διεύθυνση. Επομένως, με γνωστό Δx και Δy , το χρονικό βήμα υπολογίζεται

$$\Delta t = \sqrt{2} \frac{\Delta x \Delta y}{\Delta x + \Delta y} \quad (3.3.10)$$

3.4 Διαδικασία επίλυσης

Η επίλυση του προβλήματος γίνεται σε τρία βασικά στάδια, την αρχικοποίηση των μεταβλητών με τα δεδομένα του προβλήματος, τους βασικούς υπολογισμούς και, τέλος, την οπτικοποίηση των αποτελεσμάτων.

3.4.1 Δεδομένα του προβλήματος

Ο σκοπός αυτής της διπλωματικής, πέραν όσων αναφέρθηκαν στα προηγούμενα κεφάλαια, είναι να υπολογισθεί αριθμητικά και να παρουσιασθεί η κυκλοφορία σε έναν ανοιχτό από τη μία πλευρά κόλπο. Επιλέχθηκε οι διαστάσεις του κόλπου αυτού να είναι $25km$ σε πλάτος και $40km$ σε μήκος, ενώ το βάθος του να είναι σταθερό $50m$.

Τα ύδατα του κόλπου τη χρονική στιγμή $t = 0$ βρίσκονται σε ηρεμία και η διαδικασία επίλυσης αρχίζει με τη διέγερση του ανοιχτού άκρου από ένα ημιτονοειδές κύμα πλάτους $40cm$. Πρέπει να τονισθεί ότι τα σημεία στο άκρο του κόλπου διεγείρονται όλα ταυτόχρονα, οπότε προσομοιάζεται ένα εξωτερικό κύμα που εισέχεται στον κόλπο κάθετα, όπως φαίνεται στο σχήμα 7.

Στον Πίνακα 2 φαίνονται συγκεντρωμένα όλα τα δεδομένα του προβλήματος. Με g συμβολίζεται η βαρυτική επιτάχυνση, ενώ με N_x και N_y ο αριθμός των κόμβων που επιλέχθηκαν για να δομήσουν το πλέγμα της επίλυσης.

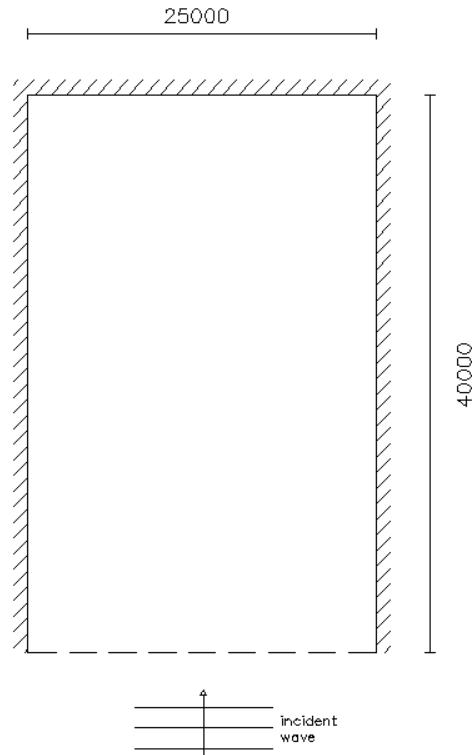


Figure 7: Κάτοψη του κόλπου. Με διακεκομμένη γραμμή φαίνεται το ανοιχτό όριο.

L_x	$25km$	T	$4000s$
L_y	$40km$	A	$0.4m$
H	$50m$	N_x	50
g	$9.807m/s^2$	N_y	50

Table 2: Δεδομένα προβλήματος

3.4.2 Υπολογισμοί

Οι υπολογισμοί για την επίλυση του προβλήματος διακρίνονται σε τρία στάδια.

- Υπολογισμός χρονικού βήματος από την εξίσωση (3.3.10)
- Για κάθε κόμβο του πλέγματος υπολογίζεται
 - Η ανύψωση - ταπείνωση της ελεύθερης επιφάνειας η από την εξίσωση (3.3.6)
 - Η οριζόντια ταχύτητα u από την εξίσωση (3.3.7)
 - Η κάθετη ταχύτητα v από την εξίσωση (3.3.8)
- Μεταβολή του χρόνου κατά dt και επανάληψη του προηγούμενου βήματος

Ο προγραμματισμός της μεθόδου επίλυσης έγινε σε περιβάλλον *Python* και το αντίστοιχο κομμάτι κώδικα φαίνεται στο παράρτημα C που υπολογίζεται η ανύψωση και ταπείνωση της ελεύθερης επιφάνειας, ενώ ο υπολογισμός της ταχύτητας φαίνεται στο παράρτημα D.

3.4.3 Αρχικές και Συνοριακές συνθήκες

Ο κόλπος πριν διεγερθεί από το εξωτερικό κύμα βρίσκεται σε κατάσταση ηρεμίας. Άρα οι αρχικές συνθήκες του προβλήματος είναι προφανές ότι θα είναι 0 για την ανύψωση της ελεύθερης επιφάνειας, αλλά και για την ταχύτητα. Η δυσκολία βρίσκεται στην εκλογή κατάλληλων συνοριακών συνθηκών, τέτοιων ώστε να προσομοιώνεται όσο το δυνατόν καλύτερα το πρόβλημα, με τους περιορισμούς που έχουμε θέσει.

Τα κλειστά άκρα του κόλπου (στεριά) προκαλούν πλήρη ανάκλαση των κυματισμών. Επομένως η επιλογή μία συνθήκης τύπου Neumann είναι η πιο κατάλληλη. Για να υπάρχει πλήρης ανάκλαση των κυματισμών είναι απαραίτητο να μηδενίζεται η κλίση της ανύψωσης - ταπείνωσης της ελεύθερης επιφάνειας του νερού και όχι να μηδενίζεται η ίδια η τιμή της. Το ίδιο συμβαίνει και με την ταχύτητα. Επομένως οι συνοριακή συνθήκη για τα κλειστά άκρα του κόλπου είναι

$$\frac{\partial \eta}{\partial x} = \frac{\partial \eta}{\partial y} = 0 \quad (3.4.1)$$

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} = 0 \quad (3.4.2)$$

Η εφαρμογή των παραπάνω συνθηκών έγινε προγραμματιστικά με την εισαγωγή *κόμβων-φαντασμάτων* όπου η τιμή τους αυτόματα παίρνει την τιμή του προηγούμενου κόμβου. Έτσι, η διαφορά τους είναι 0 και η κλίση του διανύσματος είναι, επίσης, 0.

Για το ανοιχτό άκρο η εκλογή μια συνοριακής συνθήκης είναι πιο περίπλοκη, γιατί υπάρχει το προσπίπτον κύμα που μπαίνει στον κόλπο, αλλά και τα ανακλανόμενα κύματα που εξέρχονται από τον κόλπο. Το πρόβλημα είναι δυνατόν να λυθεί με χρήση της *ελεύθερης ακτινοβολίας* που προτείνει ο Koutitas (1988).

$$\frac{\partial \zeta_r}{\partial t} + \sqrt{gh} \frac{\partial \zeta_r}{\partial y} = 0 \quad (3.4.3)$$

η διακριτοποίηση της οποίας ακολουθεί το ίδιο σχήμα με τη διακριτοποίηση της ανύψωσης της ελεύθερης επιφάνειας, με πρόσω διαφορές στο χρόνο και κεντρικές

διαφορές στο χώρο.

$$\zeta_{i,j}^{n+1} = 2\zeta_{i,j}^n - \zeta_{i,j}^{n-1} + gh \frac{\Delta t^2}{2\Delta y^2} (\zeta_{i+1,j}^n - 2\zeta_{i,j}^n + \zeta_{i,j+1}^n) \quad (3.4.4)$$

3.5 Αποτελέσματα επίλυσης

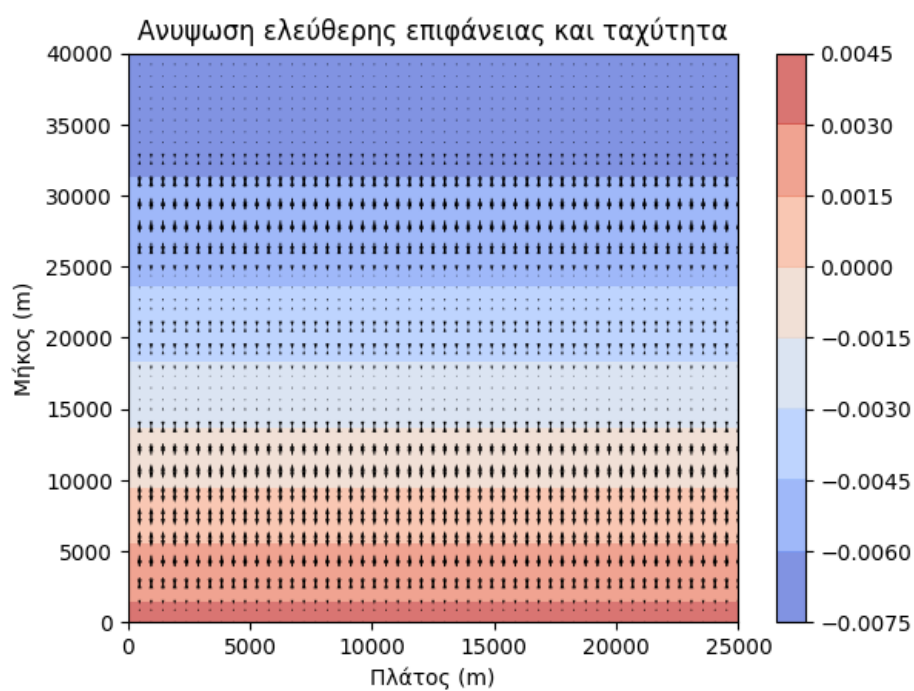
Η επίλυση έγινε για διάφορες χρονικές περιόδους και αριθμό κόμβων, ώστε να διαπιστωθεί η ευστάθεια της μεθόδου. Σαν οπτικό αποτέλεσμα για κάθε επίλυση, το πρόγραμμα επιστρέφει μία σειρά διαγραμμάτων. Μία κάτοψη του κόλπου που εμφανίζεται η ελεύθερη επιφάνεια μαζί με τα διανύσματα της ταχύτητας για την τελευταία χρονική στιγμή της επίλυσης και δύο διαγράμματα χρονοσειράς για τη μεταβολή της ελεύθερης επιφάνειας και τη μεταβολή της κατά y ταχύτητας ενός συγκεκριμένου σημείου κατά τη διαδικασία επίλυσης, δηλαδή για όλες τις χρονικές στιγμές.

Κωδικός Επίλυσης	Χρονική διάρκεια	N_x	N_y	Σημείο	Δx [m]	Δy [m]	Δt [s]	
1	a	0.5d	50	50	μέσο αν. ορίου	500	800	30
	b	0.5d	50	50	κέντρο κόλπου	500	800	30
2	a	0.5d	25	25	μέσο αν. ορίου	1000	1600	39
	b	0.5d	25	25	κέντρο κόλπου	1000	1600	39
3	a	1d	50	50	μέσο αν. ορίου	500	800	30
	b	1d	50	50	κέντρο κόλπου	500	800	30
4	a	1d	25	25	μέσο αν. ορίου	1000	1600	39
	b	1d	25	25	κέντρο κόλπου	1000	1600	39
5	a	5d	25	25	μέσο αν. ορίου	1000	1600	39
	b	5d	25	25	κέντρο κόλπου	1000	1600	39

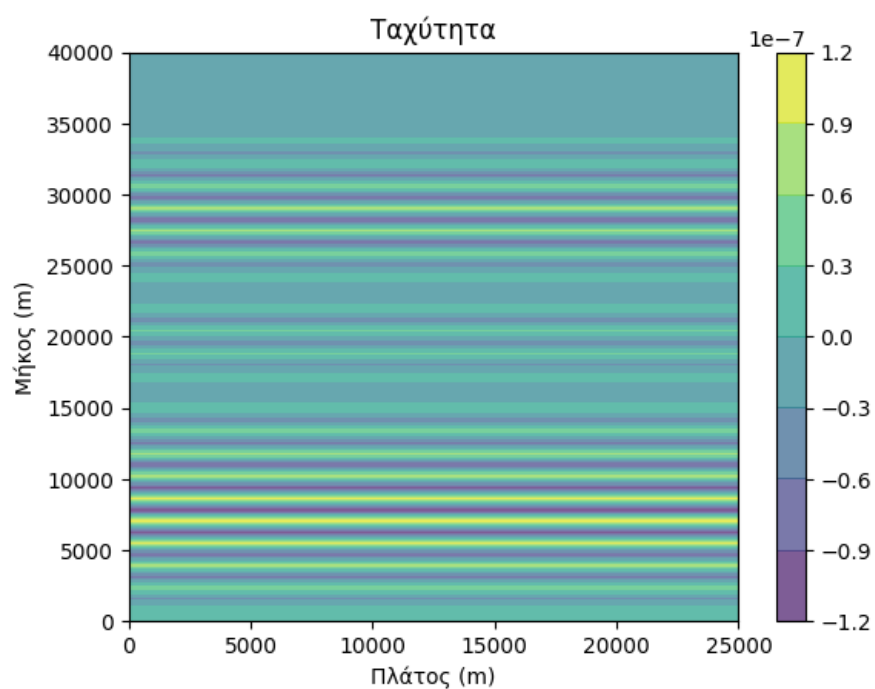
Table 3: Διαφορετικές επιλύσεις του προβλήματος

Στον Πίνακα 3 φαίνονται οι διαφορετικές επιλύσεις που έγιναν για χρόνους επίλυσης $12h$, $1day$ και $5days$, με διαφορετικό αριθμό κόμβων κάθε φορά. Ακόμα φαίνονται τα υπολογισμένα μεγέθη Δx , Δy και Δt .

3.5.1 Διαγράμματα επιλύσεων

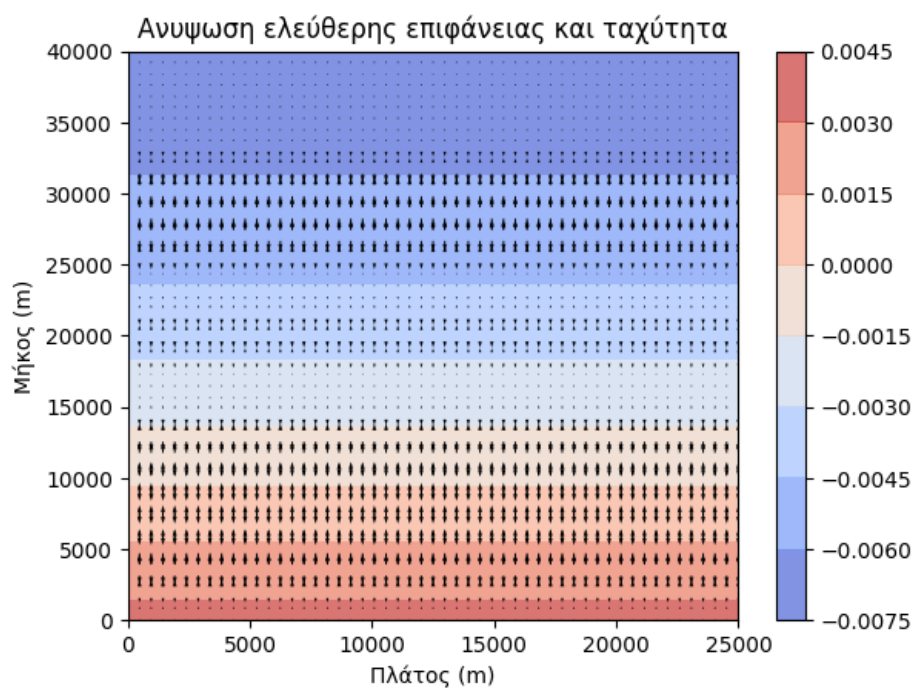


(a) Διάγραμμα Ε.Ε. και ταχύτητας

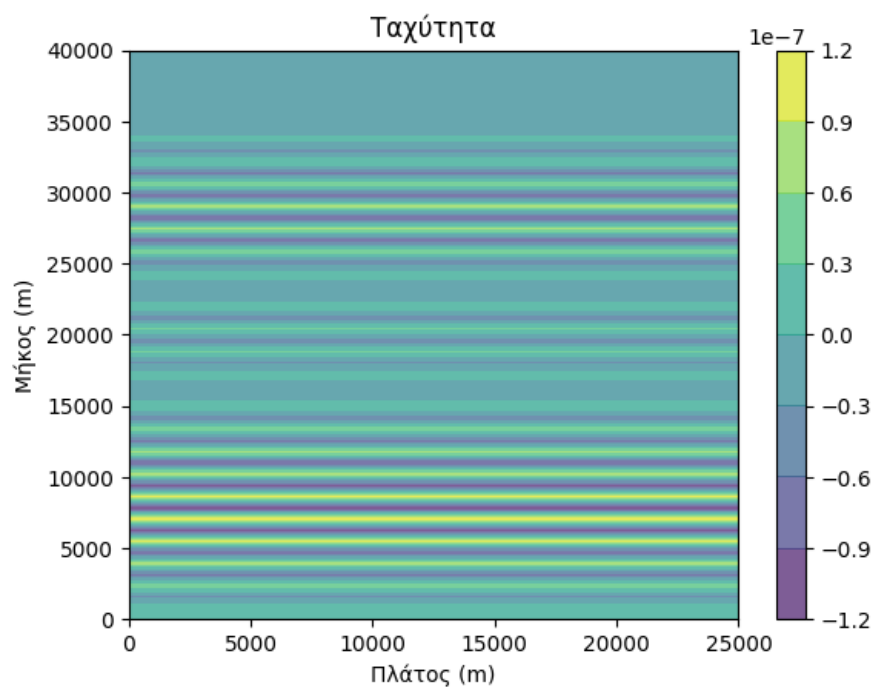


(b) Διάγραμμα ταχύτητας σε κάτοψη

Figure 8: Αποτελέσματα 1a

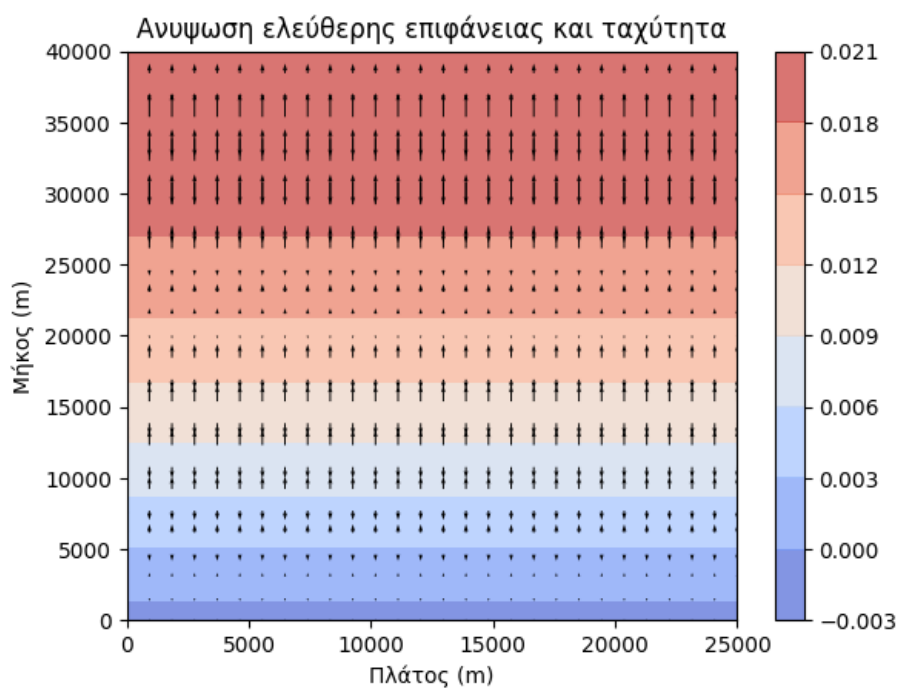


(a) Διάγραμμα Ε.Ε. και ταχύτητας

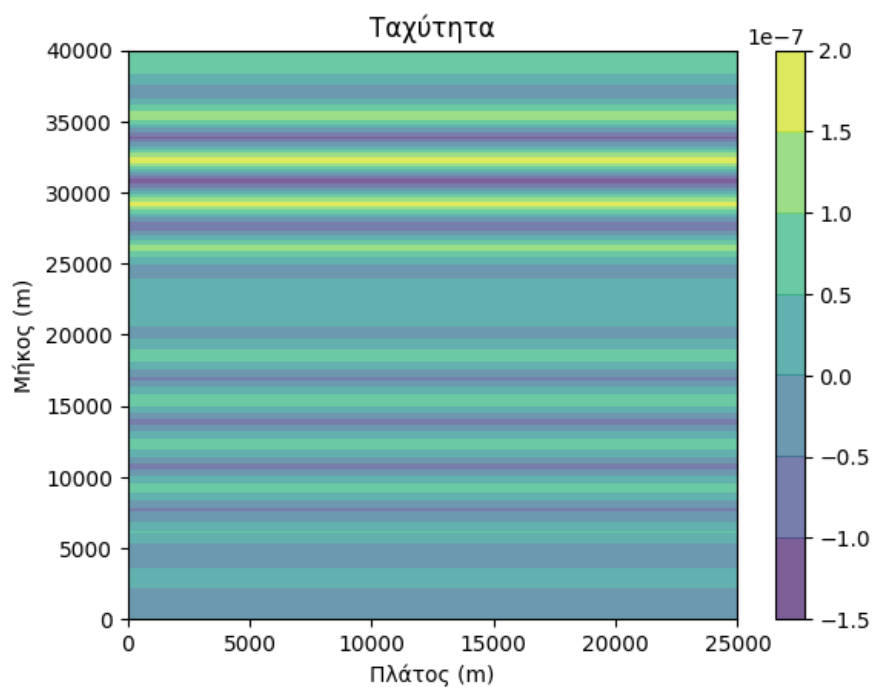


(b) Διάγραμμα ταχύτητας σε κάτοψη

Figure 9: Αποτελέσματα 1b

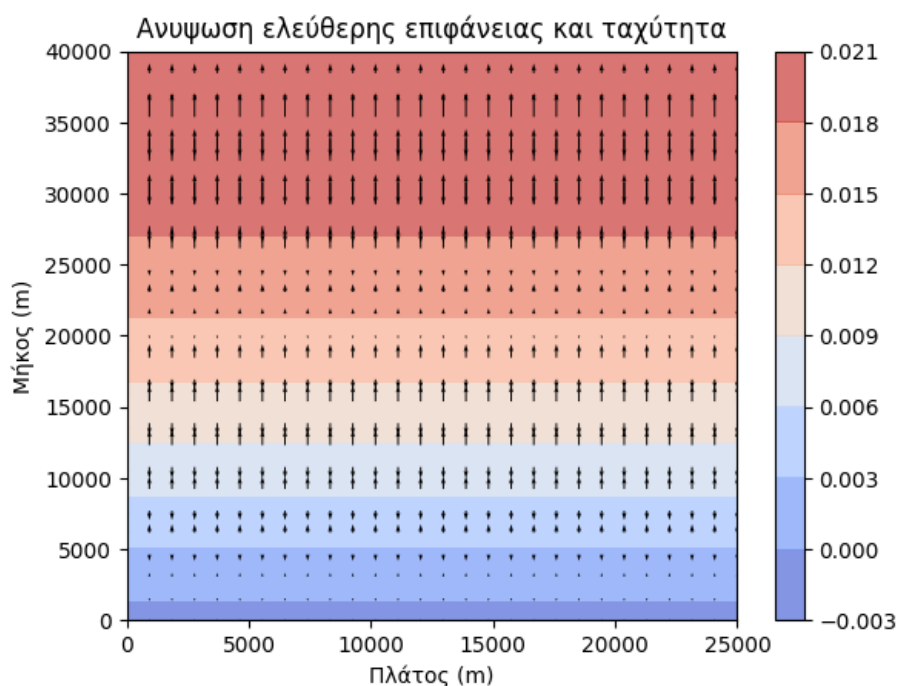


(a) Διάγραμμα Ε.Ε. και ταχύτητας

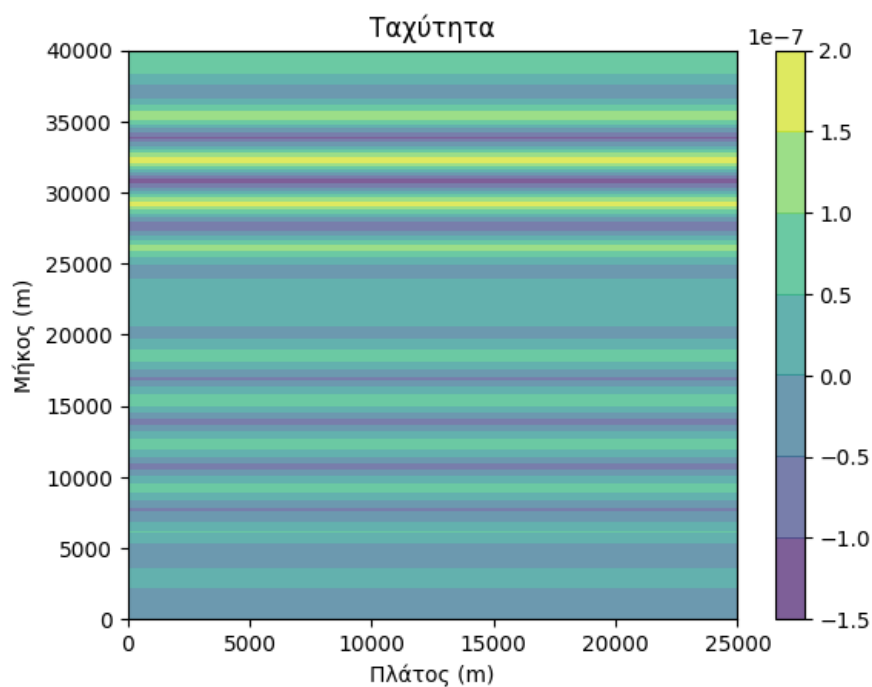


(b) Διάγραμμα ταχύτητας σε κάτοψη

Figure 10: Αποτελέσματα 2a

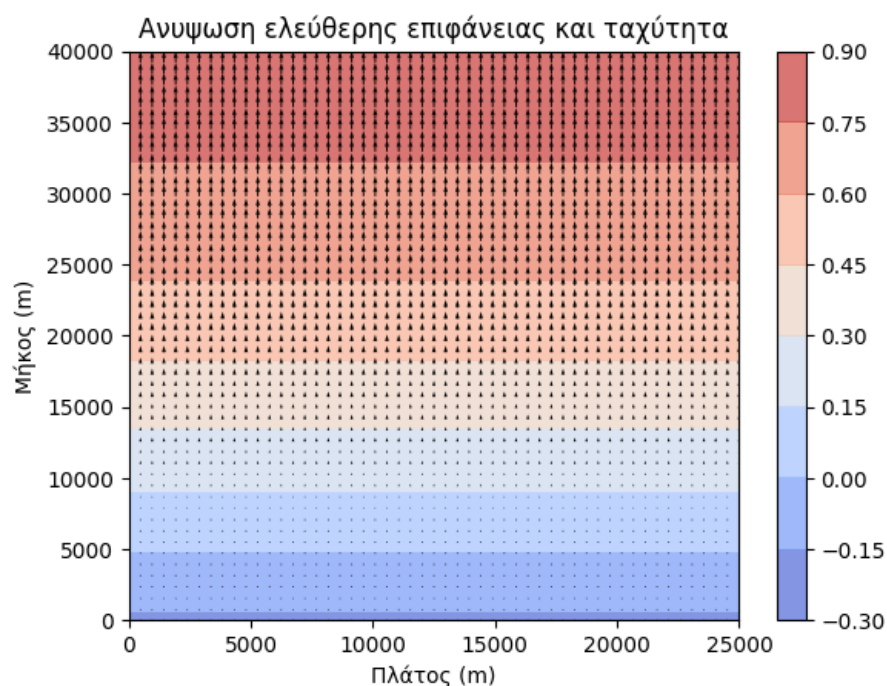


(a) Διάγραμμα Ε.Ε. και ταχύτητας

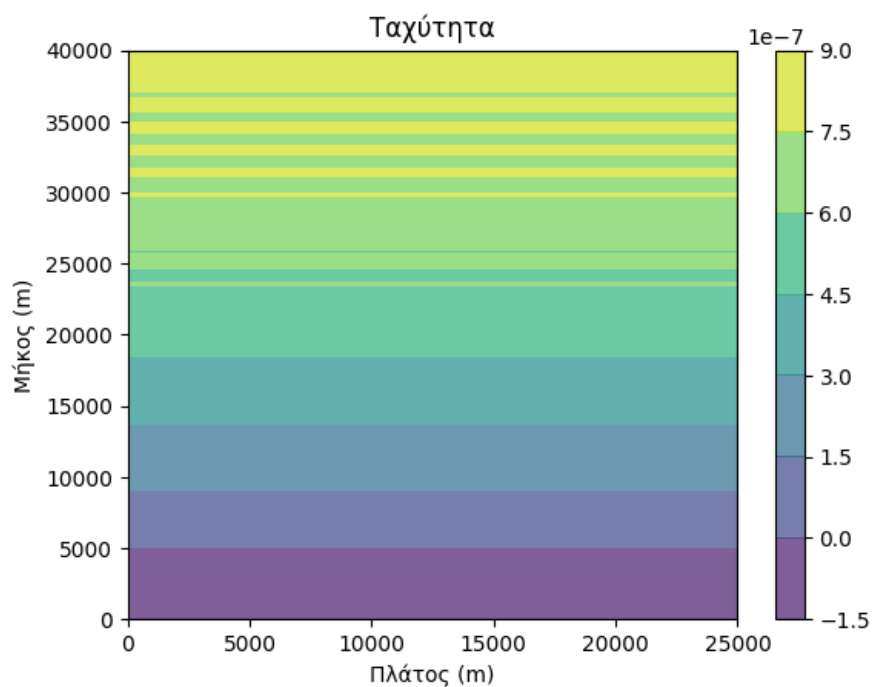


(b) Διάγραμμα ταχύτητας σε κάτοψη

Figure 11: Αποτελέσματα 2b

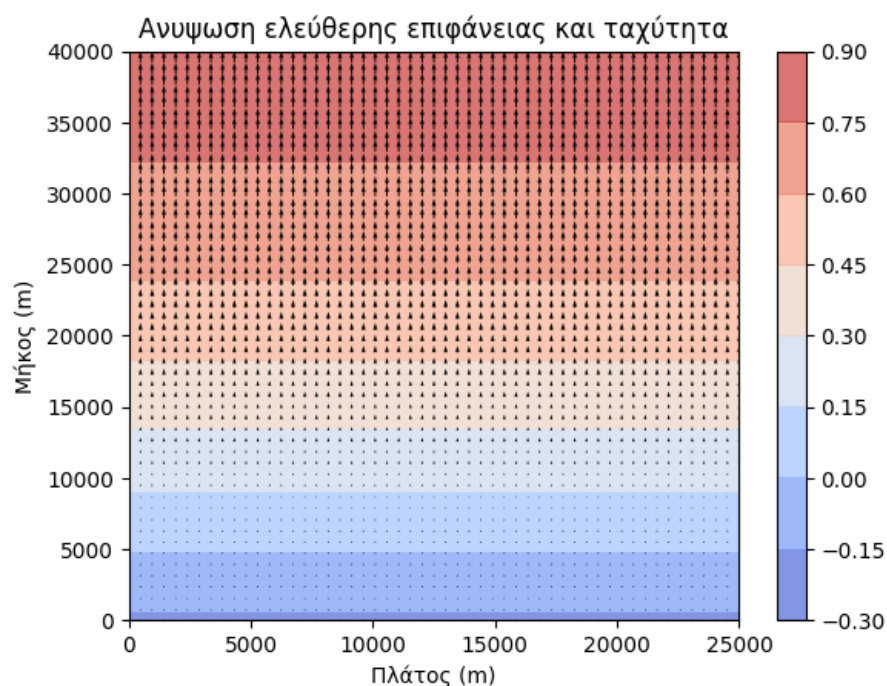


(a) Διάγραμμα Ε.Ε. και ταχύτητας

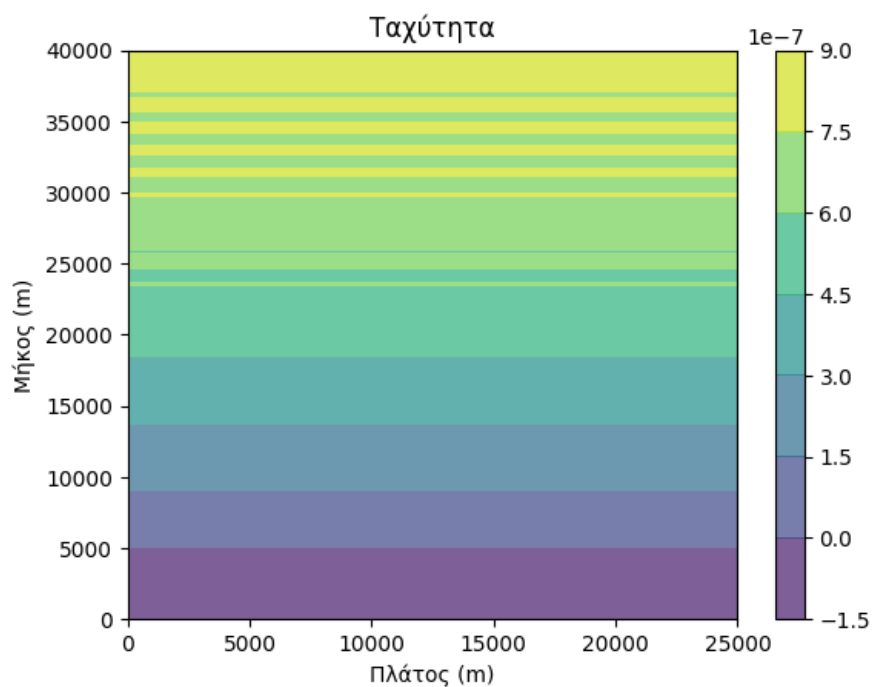


(b) Διάγραμμα ταχύτητας σε κάτοψη

Figure 12: Αποτελέσματα 3a

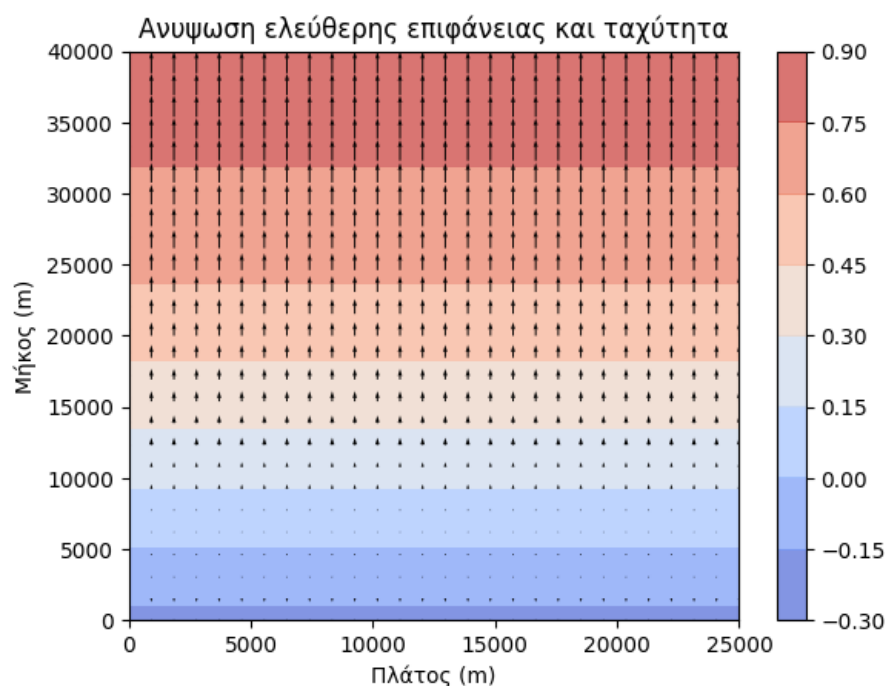


(a) Διάγραμμα Ε.Ε. και ταχύτητας

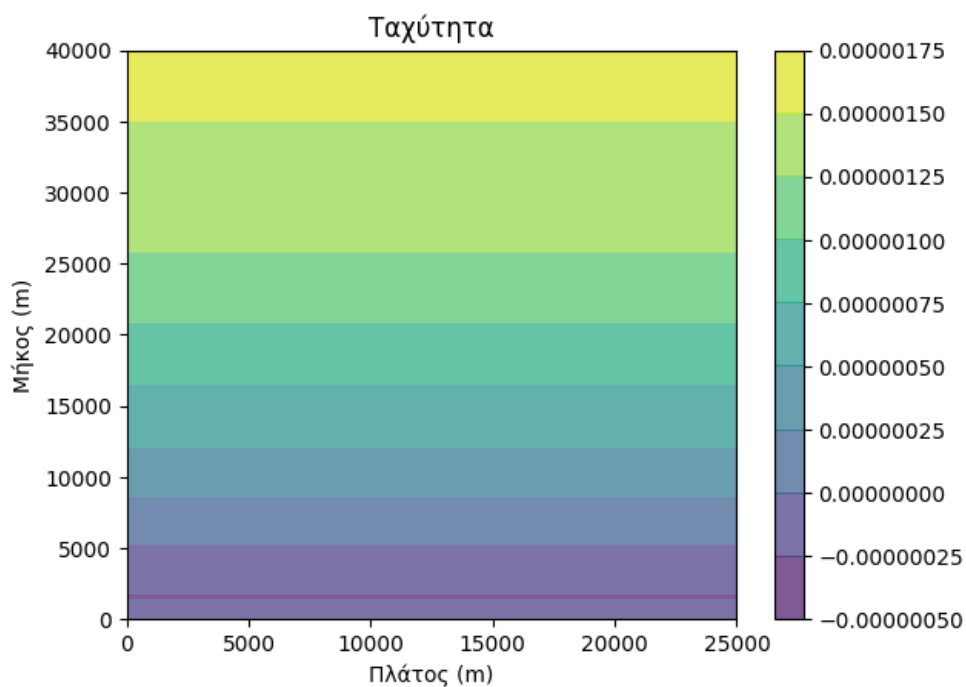


(b) Διάγραμμα ταχύτητας σε κάτοψη

Figure 13: Αποτελέσματα 3b

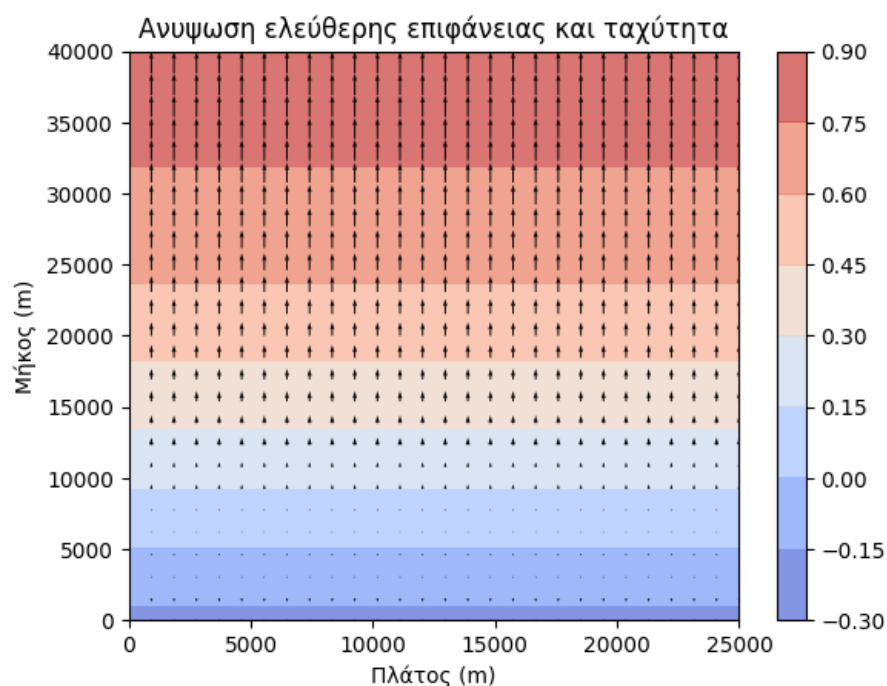


(a) Διάγραμμα Ε.Ε. και ταχύτητας

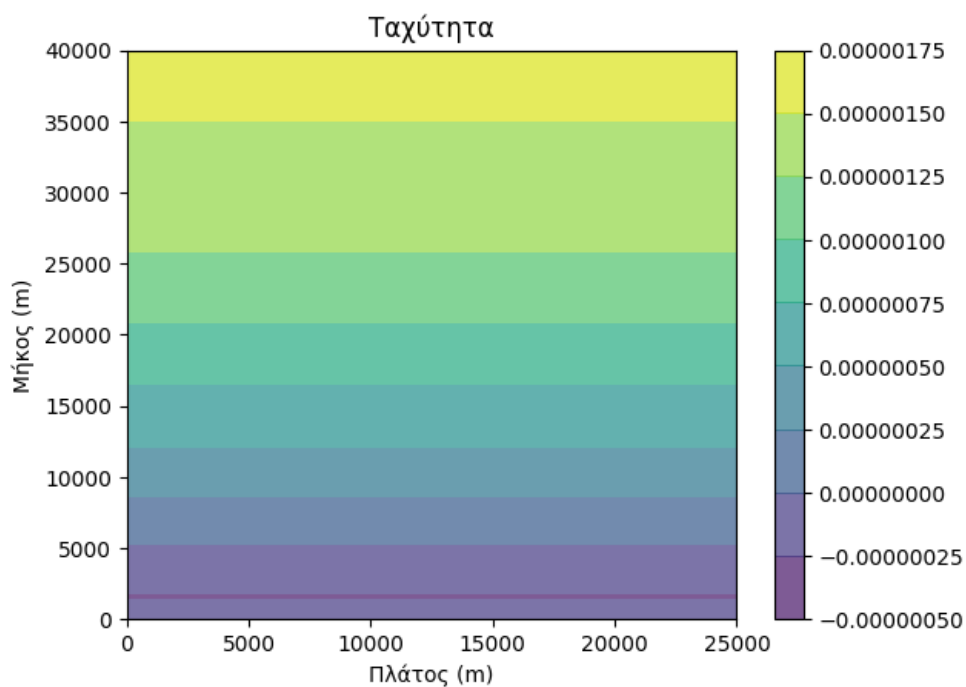


(b) Διάγραμμα ταχύτητας σε κάτοψη

Figure 14: Αποτελέσματα 4a

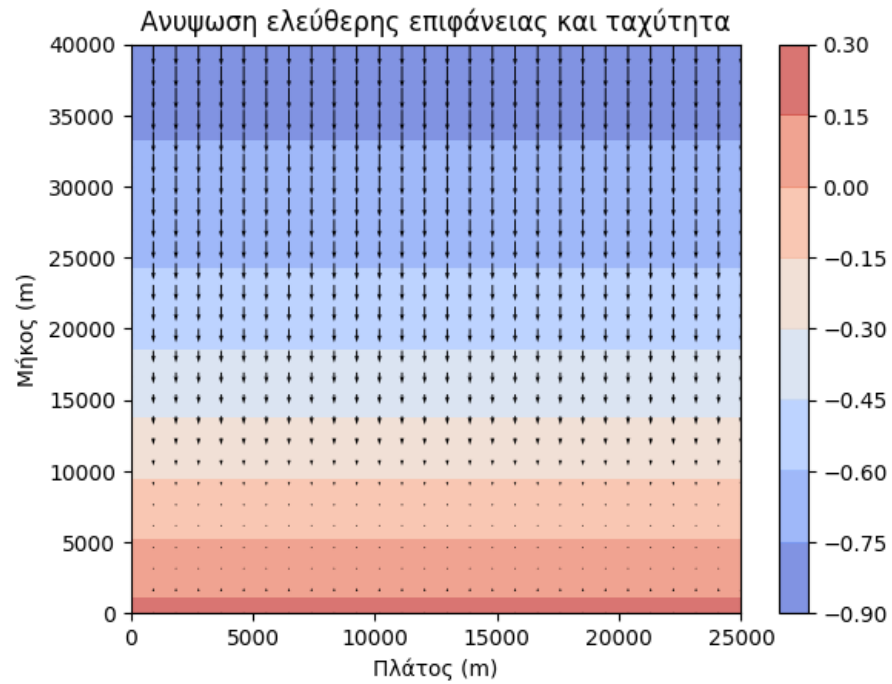


(a) Διάγραμμα Ε.Ε. και ταχύτητας

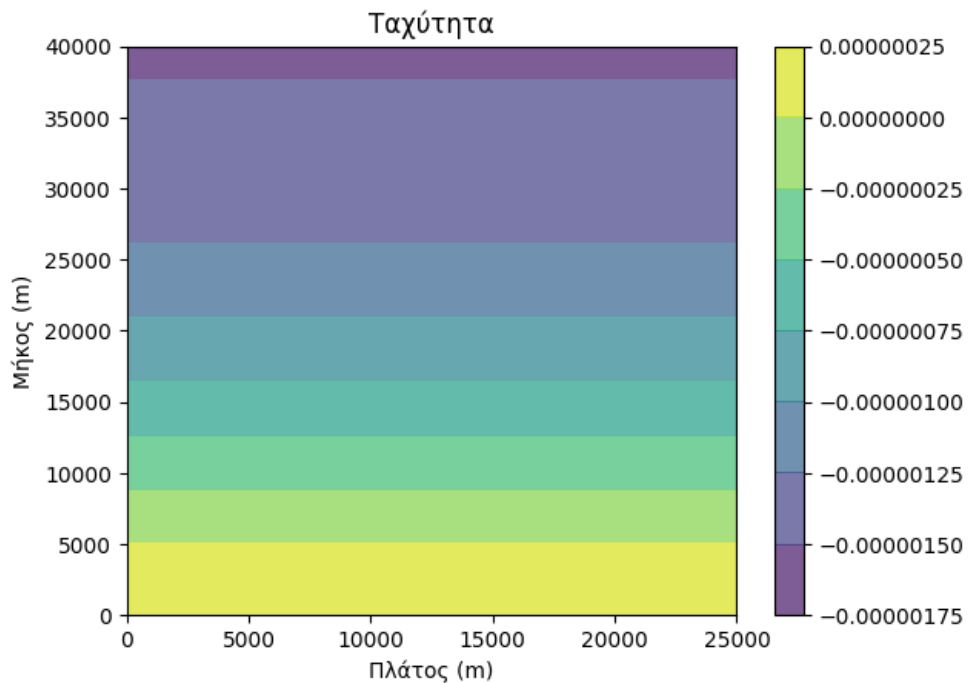


(b) Διάγραμμα ταχύτητας σε κάτοψη

Figure 15: Αποτελέσματα 4b

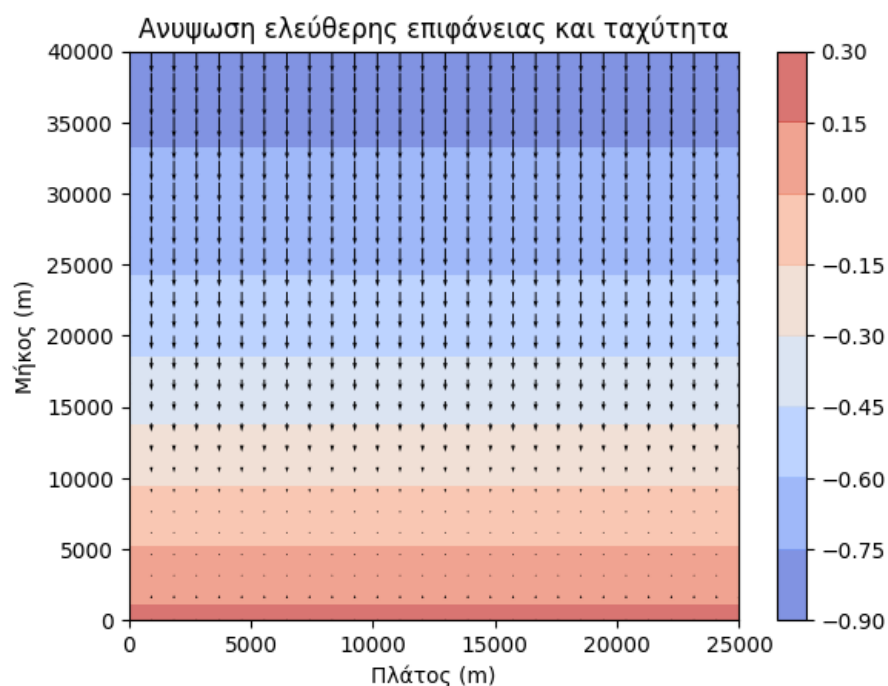


(a) Διάγραμμα Ε.Ε. και ταχύτητας

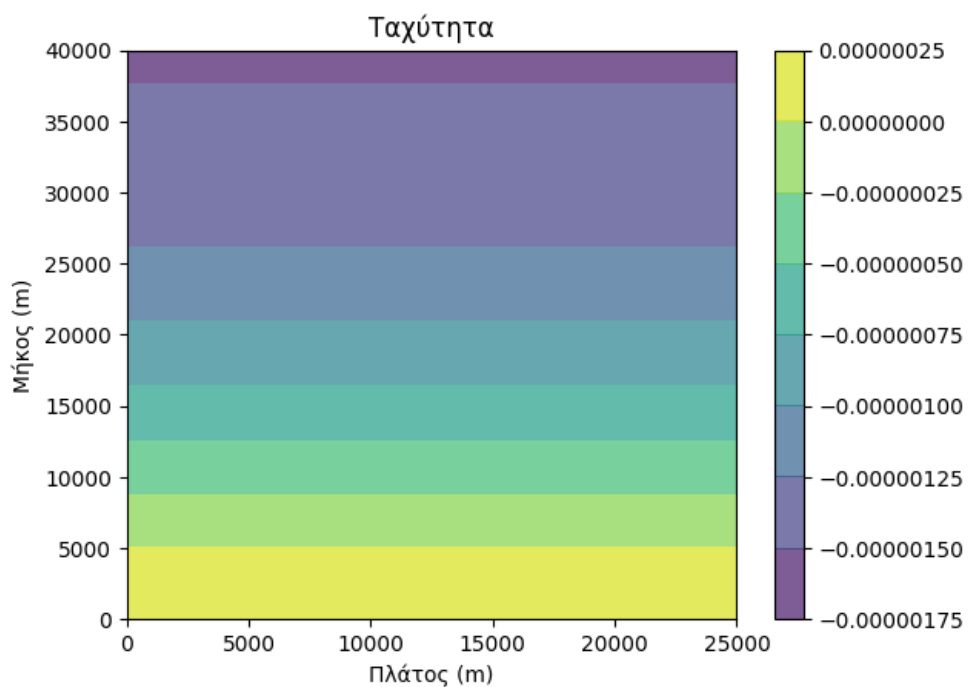


(b) Διάγραμμα ταχύτητας σε κάτοψη

Figure 16: Αποτελέσματα 5a

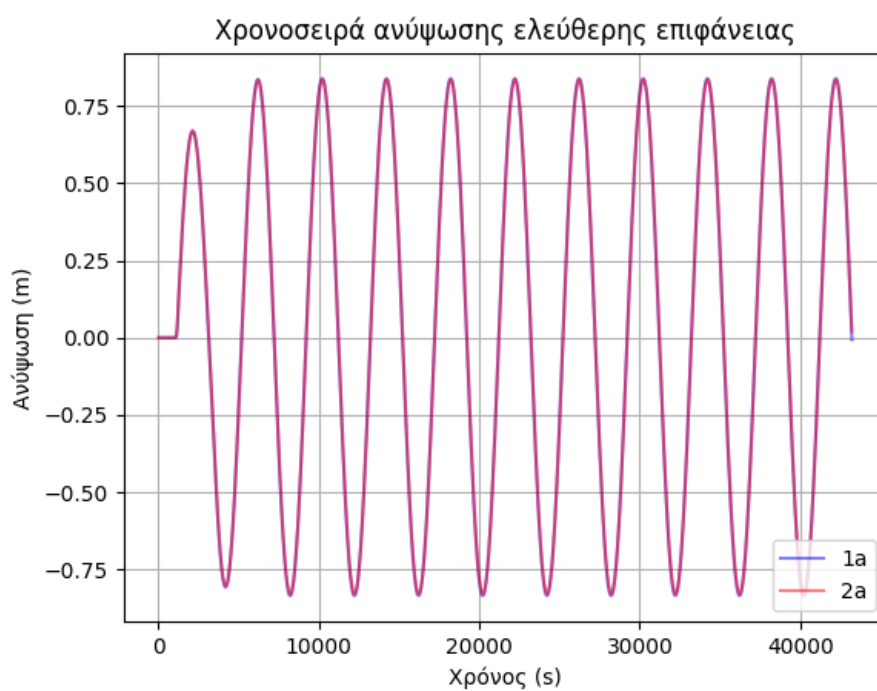


(a) Διάγραμμα Ε.Ε. και ταχύτητας

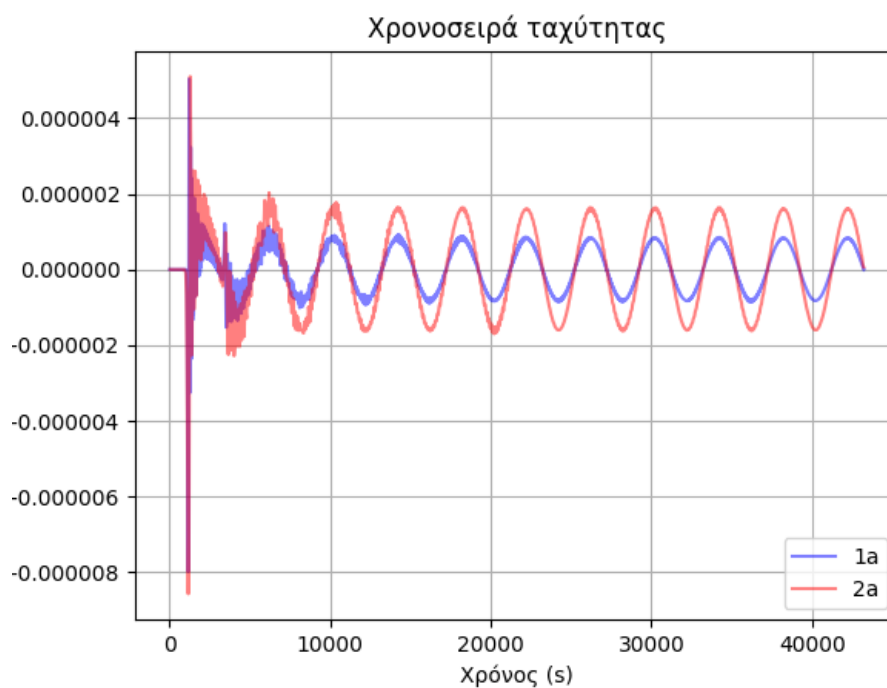


(b) Διάγραμμα ταχύτητας σε κάτοψη

Figure 17: Αποτελέσματα 5b

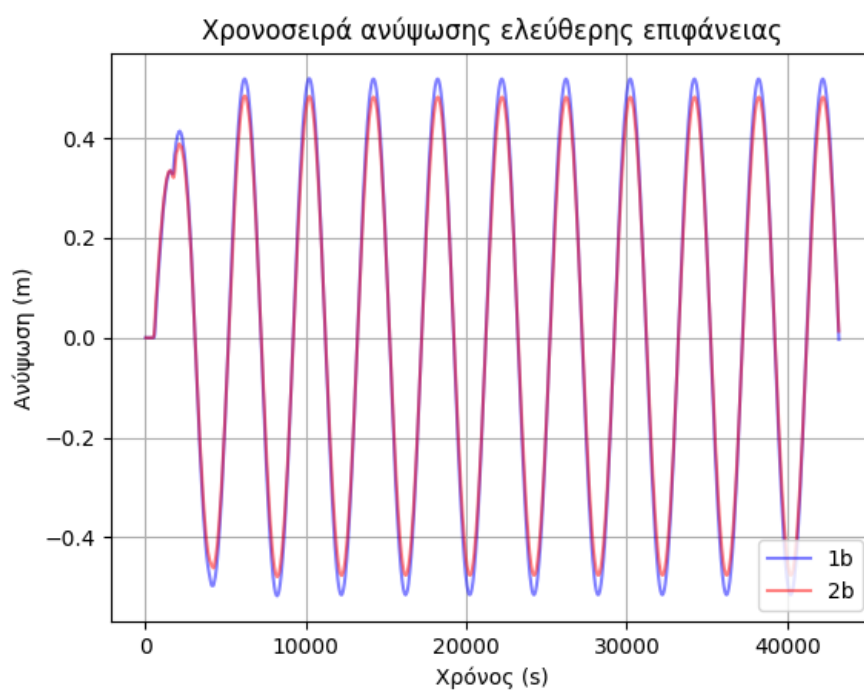


(a) Χρονοσειρά ανύψωσης ελεύθερης επιφάνειας

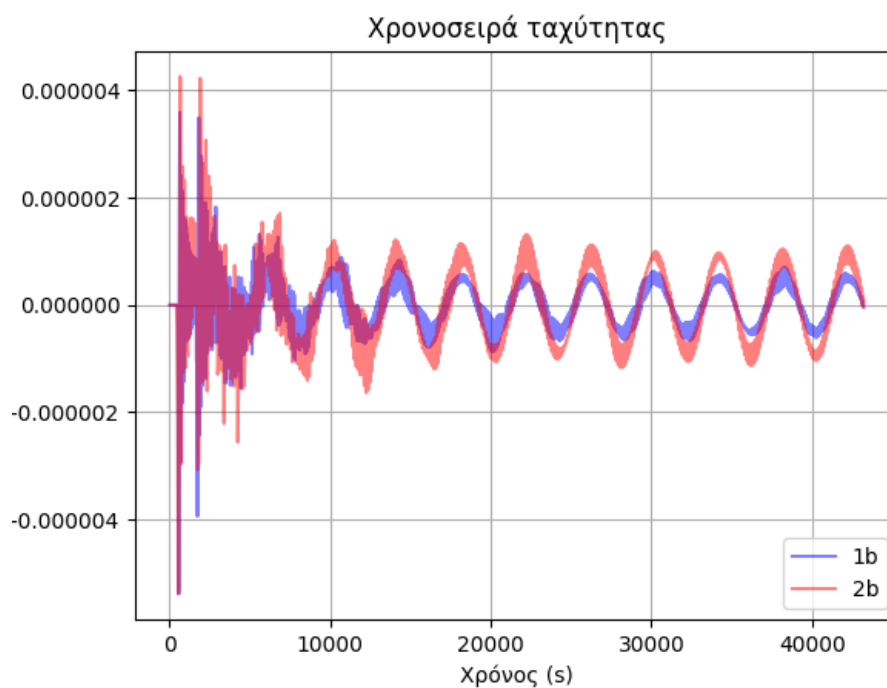


(b) Χρονοσειρά κατακόρυφης ταχύτητας

Figure 18: Συγκριτικά αποτελέσματα χρονοσειρών 1a-2a

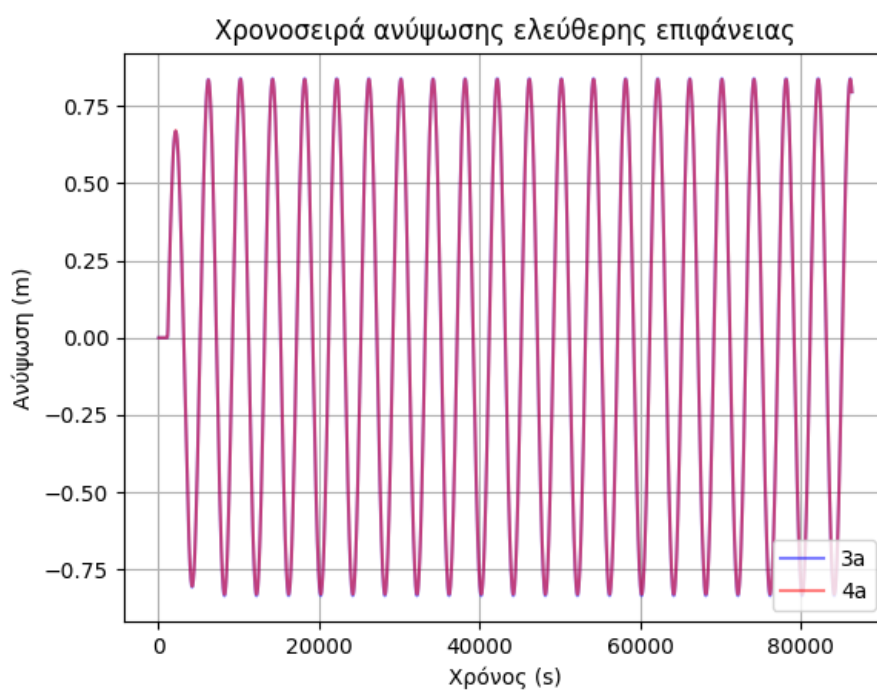


(a) Χρονοσειρά ανύψωσης ελεύθερης επιφάνειας

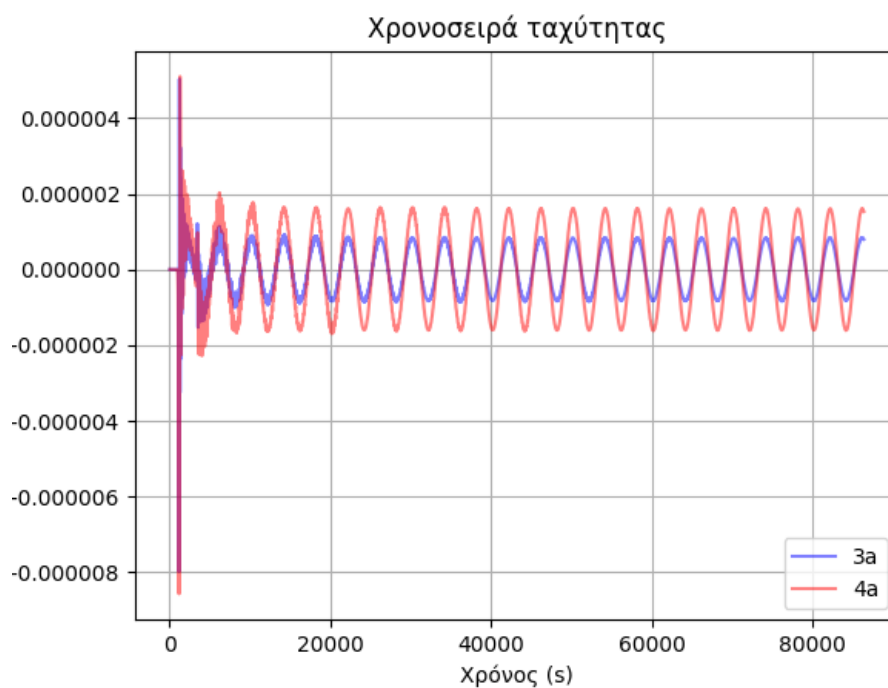


(b) Χρονοσειρά κατακόρυφης ταχύτητας

Figure 19: Συγκριτικά αποτελέσματα χρονοσειρών 1b-2b

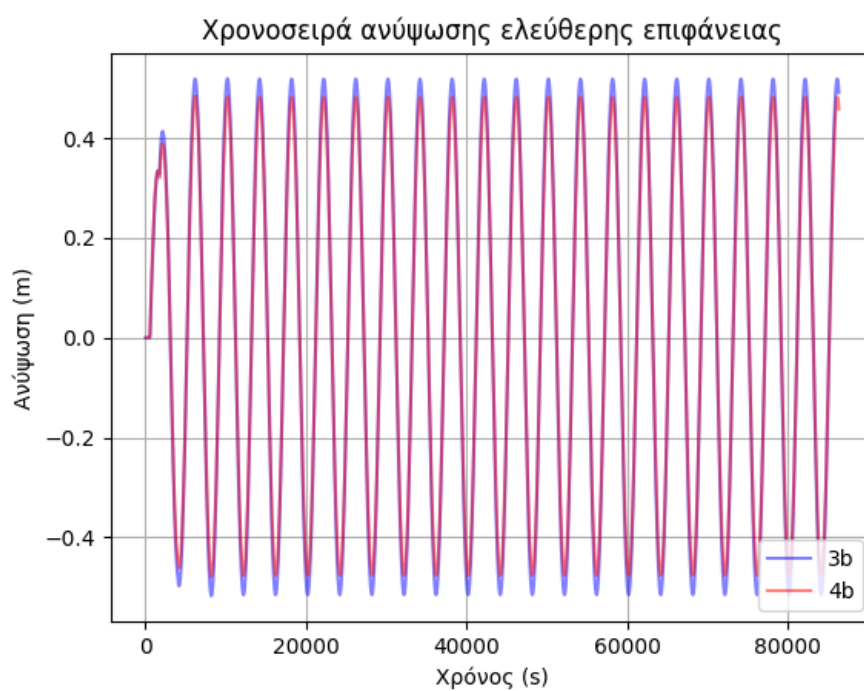


(a) Χρονοσειρά ανύψωσης ελεύθερης επιφάνειας

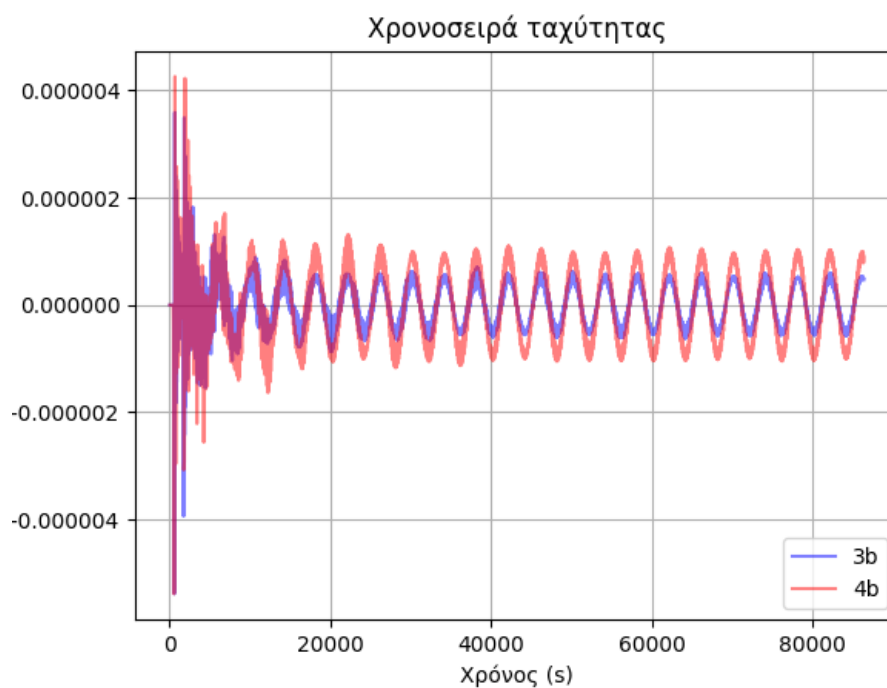


(b) Χρονοσειρά κατακόρυφης ταχύτητας

Figure 20: Συγκριτικά αποτελέσματα χρονοσειρών 3a-4a



(a) Χρονοσειρά ανύψωσης ελεύθερης επιφάνειας



(b) Χρονοσειρά κατακόρυφης ταχύτητας

Figure 21: Συγκριτικά αποτελέσματα χρονοσειρών 3b-4b

3.5.2 Σχολιασμός αποτελεσμάτων

Τα αποτελέσματα από την παρουσίαση των διαγραμμάτων παρουσιάζουν ιδιαίτερο ενδιαφέρον. Όπως θα περίμενε κανείς όταν το πλέγμα γίνεται πιο πυκνό να υπάρχει μεγαλύτερη ακρίβεια στον υπολογισμό των μεταβλητών του προβλήματος. Αντίθετα, φαίνεται, ότι κάτι τέτοιο δε συμβαίνει. Αυτό έγγυται στον περιορισμό CFL που ανάλογα με το χωρικό βήμα αλλάζει αυτόματα και το χρονικό βήμα. Παρ' όλα αυτά, οι λύσεις συγκλίνουν και παρουσιάζουν συνοχή, ιδιαιτέρως σε μεγαλύτερους χρόνους επίλυσης (1d, 5d). Σε μεγάλους χρόνους επίλυσης, η ταχύτητα παρουσιάζει την αναμενόμενη συμπεριφορά, δηλαδή να κρατά σταθερό πρόσημο σε όλο το πλέγμα κάθε χρονική στιγμή, όπως φαίνεται από τα διαγράμματα 14(b), 15(b), 16(b) και 17(b).

Ενδιαφέρον παρουσιάζουν και τα διαγράμματα χρονοσειρών της ταχύτητας, καθώς παρατηρούμε το μεγάλο σφάλμα στον υπολογισμό για τις 2 πρώτες ώρες περίπου. Η λύση, όμως, ομαλοποιείται και λαμβάνει την αναμενόμενη ημιτονοειδή μορφή με το πέρασμα του χρόνου, όπου και σταθεροποιείται.

Όσον αφορά τα διαγράμματα χρονοσειράς της ανύψωσης της ελεύθερης επιφάνειας, παρατηρούμε ότι δημιουργείται στάσιμο κύμα πλάτους $2A \approx 0.8m$, όπως φυσικά θα περίμενε κανείς, λόγω της πλήρους ανάκλασης.

Βιβλιογραφία

- Courant, R., Friedrichs, K. & Lewy, H. (1928), 'Über die partiellen differenzengleichungen der mathematischen physik', *Mathematische annalen* **100**(1), 32–74.
- Haldoupis, C. (2015), *Εισαγωγή στην ατμοσφαιρική φυσική*, Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών.
- Horsch, G. M. (2012), *Υδροδυναμική Κόλπων και Ταμειυτήρων (Σημειώσεις για το μάθημα)*, Πανεπιστήμιο Πατρών, Τμήμα Πολιτικών Μηχανικών.
- Kantha, L. H. & Clayson, C. A. (2000), *Numerical models of oceans and oceanic processes*, Academic Press.
- Koutitas, C. G. (1988), *Mathematical models in coastal engineering*, Pentech Press.
- Kämpf, J. (2009), *Ocean Modeling for Beginners, Using Open-Source Software*, Springer.
- Lorenz, E. N. (1963), 'Deterministic nonperiodic flow', *Journal of the Atmospheric Sciences* **20**(2), 130–141.
- Tabor, M. (1989), *Chaos and integrability in nonlinear dynamics: an introduction*, Wiley.
- Teunissen, C. (2017), *Inertial oscillations*. http://www.cleonis.nl/physics/phys256/inertial_oscillations.php.
- Vreugdenhil, C. B. (2013), *Numerical methods for shallow-water flow*, Vol. 13, Springer Science & Business Media.

A Αλγόριθμος επίλυσης του συστήματος Lorenz

```
1 import random
2 import numpy as np
3 from scipy.integrate import odeint
4 import file_utils
5
6 def lorenz(state, t):
7     x = state[0]
8     y = state[1]
9     z = state[2]
10
11     sigma = 10.0
12     rho = 28.0
13     beta = 8.0 / 3.0
14     # compute state derivatives
15     xd = sigma * (y - x)
16     yd = (rho - z) * x - y
17     zd = x * y - beta * z
18     # return the state derivatives
19     return [xd, yd, zd]
20
21 def get_lorenz(step, ts, tf, x0):
22     td = int((tf - ts) / step)
23     t = np.linspace(ts, tf, td)
24     x_t = np.asarray([odeint(lorenz, x0i, t) for x0i in x0])
25     if len(x_t[0]) == td:
26         print('New system calculated')
27     return x_t
28
29 if __name__ == "__main__":
30     step = 0.01
31     ts = 0
32     tf = 100
```

```
33     num_traj = 2
34     x0 = [(random.randint(-25, 25),
35           random.randint(-35, 35),
36           random.randint(5, 55)) for traj in range(num_traj)]
37     x_t = get_lorenz(step, ts, tf, x0)
38     file_utils.save_to_file(x_t)
39     array = file_utils.open_file()
40     print(array == x_t)
```

Κώδικας 2: Αλγόριθμος επίλυσης του συστήματος Lorenz σε Python

B Υπολογισμός της τροχιάς υπό την επίδραση της δύναμης Coriolis σε Python

```
1 import numpy as np
2 import matplotlib.ticker as ticker
3 import matplotlib.pyplot as plt
4 import matplotlib.animation as animation
5
6 def plot_circle(center, radius):
7     """
8     Plots a circle using the parametric equations
9     :param center: A list of the circle's center: [x,y]
10    :param radius: The radius of the circle
11    :return: None
12    """
13    fig = plt.figure(facecolor='#969696')
14    fig.canvas.set_window_title('Coriolis Trajectory')
15
16    center = tuple(center)
17    ax = fig.add_subplot(111, aspect='equal')
18    ax.set_xlim(-radius, radius)
19    ax.set_ylim(-radius, radius)
20
21    ax.tick_params(axis='both', which='major', labelsize=9)
22
23    tickfmt = ticker.ScalarFormatter(useMathText=True)
24    tickfmt.set_powerlimits((-3, 3))
25    ax.xaxis.set_major_formatter(tickfmt)
26    ax.yaxis.set_major_formatter(tickfmt)
27
28    ax.set_xlabel('X (m)')
29    ax.set_ylabel('Y (m)')
```

```
31     ax.grid(color='#969696', linestyle='--')
32     circle = plt.Circle(center, radius, color='k', fill=False)
33     ax.add_patch(circle)
34
35     ax.plot()
36     plt.show(block=False)
37     return
38
39 def update(num, x1, y1, x2, y2, line01, line02):
40     line01.set_data(x1[:num], y1[:num])
41     line02.set_data(x2[:num], y2[:num])
42     return line01, line02
43
44 def calculate(omega, numSteps, phi, R, x0, y0, u0, v0):
45     # Calculation of the Coriolis parameter
46     f = 2 * omega * np.sin(np.deg2rad(phi))
47
48     # Calculation of the inertial oscillation characteristics
49     T = 2 * np.pi / abs(f)
50     Th = T / 3600 # Period in hours
51
52     Dt = T / numSteps
53
54     if not u0 == v0:
55         vel = max(abs(u0), abs(v0))
56         Ro = abs(vel / (R * omega))
57     else:
58         Ro = abs(u0 / (R * omega))
59
60     xi = [x0]
61     yi = [y0]
62     xc = [x0]
63     yc = [y0]
64
```

```
65     theta = 0
66     t = 0
67     i = 0
68
69     while True:
70         i = i + 1
71         t = t + Dt
72         theta = theta + omega * Dt
73
74         # Parcel's Position on the Inertial
75         x1 = x0 + Dt * u0
76         y1 = y0 + Dt * v0
77
78         # Parcel's Position translated to the
79         # rotating frame
80         xc1 = x1 * np.cos(theta) + y1 * np.sin(theta)
81         yc1 = x1 * np.sin(theta) + y1 * np.cos(theta)
82         if xc1 ** 2 + yc1 ** 2 > R ** 2:
83             break
84
85         xi.append(x1)
86         yi.append(y1)
87         xc.append(xc1)
88         yc.append(yc1)
89
90         x0 = x1
91         y0 = y1
92
93     return [f, Ro, Th, xc, yc, xi, yi]
94
95 def plot(radius, xc, yc, xi, yi, animate, speed):
96     plot_circle([0, 0], radius)
97
98     ax = plt.gca()
```

```

99     ax.plot(xc[0], yc[0], 'ro')
100     if animate:
101         line1, = ax.plot([], [], 'b')
102         line2, = ax.plot([], [], 'g--')
103         ani = animation.FuncAnimation(plt.gcf(), update,
104                                         len(xc),
105                                         fargs=[xc, yc,
106                                                  xi, yi,
107                                                  line1, line2],
108                                         interval=speed,
109                                         blit=True, repeat=False)
110     else:
111         ax.plot(xi, yi, 'g--')
112         ax.plot(xc, yc, 'b')
113     ax.plot(xc[len(xc) - 1], yc[len(yc) - 1], 'r*')
114     plt.show()
115
116 def run_cli(omega, numSteps, phi, R,
117             x0, y0, u0, v0,
118             animate, speed):
119     [f, Ro, Th, xc, yc, xi, yi] = calculate(omega,
120                                             numSteps,
121                                             phi, R,
122                                             x0, y0,
123                                             u0, v0)
124
125     print('f: ', f)
126     print('Rossby: ', Ro)
127     print('T: ', Th)
128
129     plot(R, xc, yc, xi, yi, animate, speed)
130
131 if __name__ == '__main__':
132     # Test Case 1

```



```
133     omega = -0.0000724
134     numSteps = 1500
135     phi = 30
136     R = 100000
137     x0 = 100000
138     y0 = 0
139     u0 = -1
140     v0 = 0
141     run_cli(omega, numSteps, phi, R, x0, y0, u0, v0,
142             animate=False, speed=5)
143
144     # Test Case 2
145     omega = -0.0000724
146     numSteps = 500
147     phi = 30
148     R = 100000
149     x0 = 0
150     y0 = -100000
151     u0 = 0
152     v0 = 1
153     run_cli(omega, numSteps, phi, R, x0, y0, u0, v0,
154             animate=True, speed=5)
```

Κώδικας 3: Τροχιά υπό την επίδραση της Coriolis σε Python

C Υπολογισμός ανύψωσης / ταπείνωσης ελεύθερης επιφάνειας

```
1 import numpy
2 import sys
3 import velocity
4 from math import sqrt
5
6 def sin_wave(t, prop=0.0):
7     global A
8     return A * numpy.sin(2 * numpy.pi * t / T - prop)
9
10 def variables(Nx, Ny, Lx, Ly, c, g_, H, Tin, A0):
11     global A, T, g
12     g = g_
13     A = A0
14     T = Tin
15     x = numpy.linspace(0, Lx, Nx + 3)
16     y = numpy.linspace(0, Ly, Ny + 2)
17     dx = Lx / Nx
18     dy = Ly / Ny
19     dt = round(sqrt(2) * dx * dy / c / (dx + dy))
20     print('Calculated Dx: {}'.format(dx))
21     print('Calculated Dy: {}'.format(dy))
22     print('Calculated Dt: {}'.format(dt))
23     if not dt <= numpy.sqrt(2*dx/(g*H)) and dt <= numpy.sqrt(2*dy/(g*H)):
24         print("CFL criterion not meet and the solution will be unstable")
25         f = numpy.sqrt(2*dx/(g*H))
26         s = numpy.sqrt(2*dy/(g*H))
27         print(dt, f, s)
28
29 if (T / dt) < 10: # CFL Condition
30     print('The calculated Dt is not at least an order\n'
```

```

31     'of magnitude less than the tidal wave period.\n'
32     'Please change Nx and/or Ny')
33     sys.exit()
34     return x, y, dx, dy, dt
35
36
37 def calculate_water_level(Lx, Ly, Nx, Ny, dx, dy, dt, c, days, P=None):
38     # P : tuple of a point's coord to get the time series
39     print('Calculating water level profile for {} days...'.format(days))
40     # Useful variables
41     Cxs = (c * dt / dx) ** 2 # Courant number for X - squared
42     Cys = (c * dt / dy) ** 2 # Courant number for Y - squared
43
44     # Instantiating arrays
45     # numpy.zeros(rows, columns)
46     u = numpy.zeros((Ny + 2, Nx + 3)) # u(n+1)(i,j) -> solution array
47     u_1 = numpy.zeros((Ny + 2, Nx + 3)) # u(n)(i,j) -> solution at t-1
48     u_2 = numpy.zeros((Ny + 2, Nx + 3)) # u(n-1)(i,j) -> solution at t-2
49
50     # For time series plot
51     u_ = []
52     uvel_ = []
53     vvel_ = []
54     t_ = []
55
56     t = 0
57     n = 0
58     # Loop Start
59     while True:
60         t = t + dt
61         n = n + 1
62         for i in range(1, Ny + 1):
63             for j in range(1, Nx + 2):
64                 u_xx = u_1[i + 1, j] - 2 * u_1[i, j] + u_1[i - 1, j]

```

```

65         u_yy = u_1[i, j + 1] - 2 * u_1[i, j] + u_1[i, j - 1]
66         u[i, j] = 2 * u_1[i, j] - u_2[i, j] + Cxs * u_xx + Cys * u_yy
67
68     # Open sea boundary
69     for j in range(Nx + 2):
70         if t-dt > dy/c:
71             u1 = u_1[0, j] - sin_wave(t - dt)
72             u2 = u_1[1, j] - sin_wave(t - dt, dy / Ly)
73             u1 = u1 + dt / dy * c * (u2 - u1)
74         else:
75             u1 = 0
76             u[0, j] = u1 + sin_wave(t)
77
78     # Closed boundary with ghost cells
79     # Fully reflective boundaries (Neumann Conditions)
80     # * top (i = Ny)
81     u[-1, :] = u[-2, :]
82     # * left (j = Nx)
83     u[:, -1] = u[:, -2]
84     # * right (j = 0)
85     u[:, 0] = u[:, 1]
86
87     if P is not None:
88         xp = P[0]
89         yp = P[1]
90         u_.append(u[xp, yp])
91         t_.append(t)
92
93     u_2 = numpy.copy(u_1)
94     u_1 = numpy.copy(u)
95
96     u_vel, v_vel = velocity.velocity(dx, dy, dt, g, u, Nx, Ny)
97     if P is not None:
98         xp = P[0]

```

```
99     yp = P[1]
100     uvel_.append(u_vel[xp, yp])
101     vvel_.append(v_vel[xp, yp])
102
103     if t > 3600 * 24 * days:
104         break
105
106     return u, u_vel, v_vel, u_, uvel_, vvel_, t_
```

Κώδικας 4: Υπολογισμός ανύψωσης / ταπείνωσης ελεύθερης επιφάνειας

D Υπολογισμός διανύσματος ταχύτητας

```

1 import numpy
2
3 def velocity(dx, dy, dt, g, z, Nx, Ny):
4     # where z is the calculated water profile
5     # for the specific timestep we do the calculation
6     # Instantiate arrays with 0s for better performance
7     u = numpy.zeros((Ny + 2, Nx + 3))
8     u1 = numpy.zeros((Ny + 2, Nx + 3))
9     v = numpy.zeros((Ny + 2, Nx + 3))
10    v1 = numpy.zeros((Ny + 2, Nx + 3))
11
12    for i in range(1, Ny + 1):
13        for j in range(1, Nx + 2):
14            u1[i, j] = u[i, j] - (g*dt/dx**2) * (z[i+1, j] -
15                2*z[i, j] + z[i-1, j])
16            v1[i, j] = v[i, j] - (g*dt/dy**2) * (z[i, j+1] -
17                2*z[i, j+1] + z[i, j+1])
18
19    # * top (i = Ny)
20    u1[-1, :] = u1[-2, :]
21    v1[-1, :] = v1[-2, :]
22    # * left (j = Nx)
23    u1[:, -1] = u1[:, -2]
24    v1[:, -1] = v1[:, -2]
25    # * right (j = 0)
26    u1[:, 0] = u1[:, 1]
27    v1[:, 0] = v1[:, 1]
28
29    u = numpy.copy(u1)
30    v = numpy.copy(v1)
31

```

32 **return** u, v

Κώδικας 5: Υπολογισμός διανύσματος ταχύτητας