



UNIVERSITY OF
PATRAS
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ

Τμήμα πολιτικών Μηχανικών

Μοντελοποίηση κυκλοφορίας σε κόλπο και άλλα προβλήματα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Νικόλαος Ανδρεάκος

Επιβλέπων Καθηγητής:

Γεώργιος Μ. Χορς, Αναπληρωτής Καθηγητής

Πάτρα, 2018

Πρόλογος

Νυλλα μαλεσουαδα πορττιτορ διαμ. Δονες φελις ερατ, ζονγυε νον, ολυτπατ ατ, τινσιδυντ τριστιχυε, λιβερο. Ίαμυς ιερρα φερμεντυμ φελις. Δονες νονυμψ πελλεντεσχυε αντε. Πηασελλυς αδιπιςινγ σεμπερ ελιτ. Προιν φερμεντυμ μασσα ας χυαμ. Σεδ διαμ τυρπις, μολεστιε ιταε, πλασερατ α, μολεστιε νες, λεο. Μαεσενας λασινια. Ναμ ιπουμ λιγυλα, ελειφενδ ατ, αςσυμσαν νες, συςσιπιτ α, ιπουμ. Μορβι βλανδιτ λιγυλα φευγιατ μαγνα. Νυνς ελειφενδ ζονσεχυατ λορεμ. Σεδ λασινια νυλλα ιταε ενιμ. Πελλεντεσχυε τινσιδυντ πυρυσ ελ μαγνα. Ιντεγερ νον ενιμ. Πραεσεντ ευισμοδ νυνς ευ πυρυσ. Δονες βιβενδυμ χυαμ ιν τελλυς. Νυλλαμ ζυρσυς πυλιναρ λεστυσ. Δονες ετ μι. Ναμ υλπυτατε μετυς ευ ενιμ. Ξστιθυλυμ πελλεντεσχυε φελις ευ μασσα.

Χυισχυε υλλαμζορπερ πλασερατ ιπουμ. ΄ρας νιβη. Μορβι ελ θυστο ιταε λαςυς τινσιδυντ υλτριςες. Λορεμ ιπουμ δολορ σιτ αμετ, ζονσεζεττυερ αδιπιςινγ ελιτ. Ιν ηας ηαβιτασσε πλατεα διςτυμσι. Ιντεγερ τεμπυς ζοναλλις αυγυε. Ετιαμ φασιλιςις. Νυνς ελεμεντυμ φερμεντυμ ωισι. Αενεαν πλασερατ. Υτ ιμπερδιετ, ενιμ σεδ γραιδα σολλισιτυδιν, φελις οδιο πλασερατ χυαμ, ας πυλιναρ ελιτ πυρυσ εγεται ενιμ. Νυνς ιταε τορτορ. Προιν τεμπυς νιβη σιτ αμετ νισλ. Ίαμυς χυις τορτορ ιταε ρισυς πορτα εξισυλα.

Φυσσε μαυρις. Ξστιθυλυμ λυστυς νιβη ατ λεστυσ. Σεδ βιβενδυμ, νυλλα α φαυσιβυς σεμπερ, λεο ελιτ υλτριςιες τελλυς, ας ενενατις αρσυ ωισι ελ νισλ. Ξστιθυλυμ διαμ. Αλιχυαμ πελλεντεσχυε, αυγυε χυις σαγιττις ποσυερε, τυρπις λαςυς ζονγυε χυαμ, ιν ηενδρεριτ ρισυς ερος εγεται φελις. Μαεσενας εγεται ερατ ιν σαπιεν ματτις πορττιτορ. Ξστιθυλυμ πορττιτορ. Νυλλα φασιλιςι. Σεδ α τυρπις ευ λαςυς ζομμοδο φασιλιςις. Μορβι φρινγιλλα, ωισι ιν διγνισσιμ ιντερδυμ, θυστο λεστυσ σαγιττις δυι, ετ εξισυλα λιβερο δυι ζυρσυς δυι. Μαυρις τεμπορ λιγυλα σεδ λαςυς. Δυις ζυρσυς ενιμ υτ αυγυε. ΄ρας ας μαγνα. ΄ρας νυλλα. Νυλλα εγεστας. Ύραβιτυρ α λεο. Χυισχυε εγεστας ωισι εγεται νυνς. Ναμ φευγιατ λαςυς ελ εστ. Ύραβιτυρ ζονσεζεττυερ.

Περίληψη

Συσπενδισσε ιταε ελιτ. Αλιχουαμ αρσυ νεχυε, ορναρε ιν, υλλαμζορπερ χυις, ζομμοδο ευ, λιβερο. Φυσσε σαγιττις ερατ αι ερατ τριστιχυε μολλις. Μαεζενας σαπιεν λιβερο, μολεστιε ετ, λοβορτις ιν, σοδαλες εγετ, δυι. Μορβι υλτριζες ρυτρυμ λορεμ. Ναμ ελεμεντυμ υλλαμζορπερ λεο. Μορβι δυι. Αλιχουαμ σαγιττις. Νυνς πλαζερατ. Πελλεντεσχυε τριστιχυε σοδαλες εστ. Μαεζενας ιμπερδιετ λασινια ελιτ. "ρας νον υρνα. Μορβι ερος πεδε, συσιπιτ ας, αριυς ελ, εγεστας νον, ερος. Πραεσεντ μαλεσουαδα, διαμ ιδ πρετιυμ ελεμεντυμ, ερος σεμ διςτυμ τορτορ, ελ ζονσεζετευερ οδιο σεμ σεδ ωισι.

Σεδ φευγιατ. ΰμ σοσις νατοχυε πενατιβυς ετ μαγνις δις παρτυριεντ μοντες, νασζετυρ ριδιςυλυσ μυς. Υτ πελλεντεσχυε αυγυε σεδ υρνα. Ξοτιβυλυμ διαμ ερος, φρινγιλλα ετ, ζονσεζετευερ ευ, νονυμμψ ιδ, σαπιεν. Νυλλαμ αι λεζτυς. Ιν σαγιττις υλτριζες μαυρις. ΰραβιτυρ μαλεσουαδα ερατ σιτ αμετ μασσα. Φυσσε βλανδιτ. Αλιχουαμ ερατ ολυτπατ. Αλιχουαμ ευισμοδ. Αενεαν ελ λεζτυς. Νυνς ιμπερδιετ θυστο νες δολορ.

Ετιαμ ευισμοδ. Φυσσε φασιλις λασινια δυι. Συσπενδισσε ποτεντι. Ιν μι ερατ, ζυρσυς ιδ, νονυμμψ σεδ, υλλαμζορπερ εγετ, σαπιεν. Πραεσεντ πρετιυμ, μαγνα ιν ελειφενδ εγεστας, πεδε πεδε πρετιυμ λορεμ, χυις ζονσεζετευερ τορτορ σαπιεν φασιλις μαγνα. Μαυρις χυις μαγνα αριυς νυλλα σζελερισχυε ιμπερδιετ. Αλιχουαμ νον χυαμ. Αλιχουαμ πορτιτορ χυαμ α λαςυς. Πραεσεντ ελ αρσυ υτ τορτορ ζυρσυς ολυτπατ. Ιν ιταε πεδε χυις διαμ βιβενδυμ πλαζερατ. Φυσσε ελεμεντυμ ζοναλλις νεχυε. Σεδ δολορ ορσι, σζελερισχυε ας, δαπιβυς νες, υλτριζιες υτ, μι. Δυις νες δυι χυις λεο σαγιττις ζομμοδο.

Περιεχόμενα

Πρόλογος	i
Περίληψη	ii
Περιεχόμενα	iii
Κατάλογος Σχημάτων	iv
Κατάλογος Πινάκων	v
Κατάλογος Αλγορίθμων	vi
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
2 Δύναμη Coriolis	6
2.1 Παράμετρος Coriolis f	6
2.2 Μετασχηματισμός θέσης σε περιστρεφόμενο σύστημα αναφοράς	7
2.3 Περιγραφή του αλγορίθμου	8
2.4 Δημιουργία λογισμικού	9
2.4.1 Κυρίως παράθυρο	9
2.4.2 Παράθυρο γραφικής παράστασης	11
2.5 Αποτελέσματα και Σχολιασμός	12
2.5.1 Αδρανειακές Ταλαντώσεις	13
2.5.2 Υπολογισμός του αριθμού Rossby	15
3 Μοντελοποίηση Παλιρροϊκής Ροής	16
Bibliography	17
A Αλγόριθμος επίλυσης του συστήματος Lorenz	18

B Υπολογισμός της τροχιάς υπό την επίδραση της δύναμης Coriolis σε Python	20
--	-----------

Κατάλογος Σχημάτων

1	Γραφική διεπαφή χρήστη του λογισμικού "Coriolis Trajectory".	10
2	Παράθυρο γραφικής παράστασης του λογισμικού "Coriolis Trajectory" .	12
3	Οι δυνάμεις στο πείραμα του περιστρεφόμενου παραβολικού δίσκου. Teunissen (2017)	13

Κατάλογος Πινάκων

1	Αλγόριθμος τροχιάς σημείου σε περιστρεφόμενο σύστημα αναφοράς. . .	9
---	--	---

Κατάλογος Αλγορίθμων

2	Αλγόριθμος επίλυσης του συστήματος Lorenz σε Python	18
3	Τροχιά υπό την επίδραση της Coriolis σε Python	20

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:

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

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

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 κατάστροψη 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	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 1 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

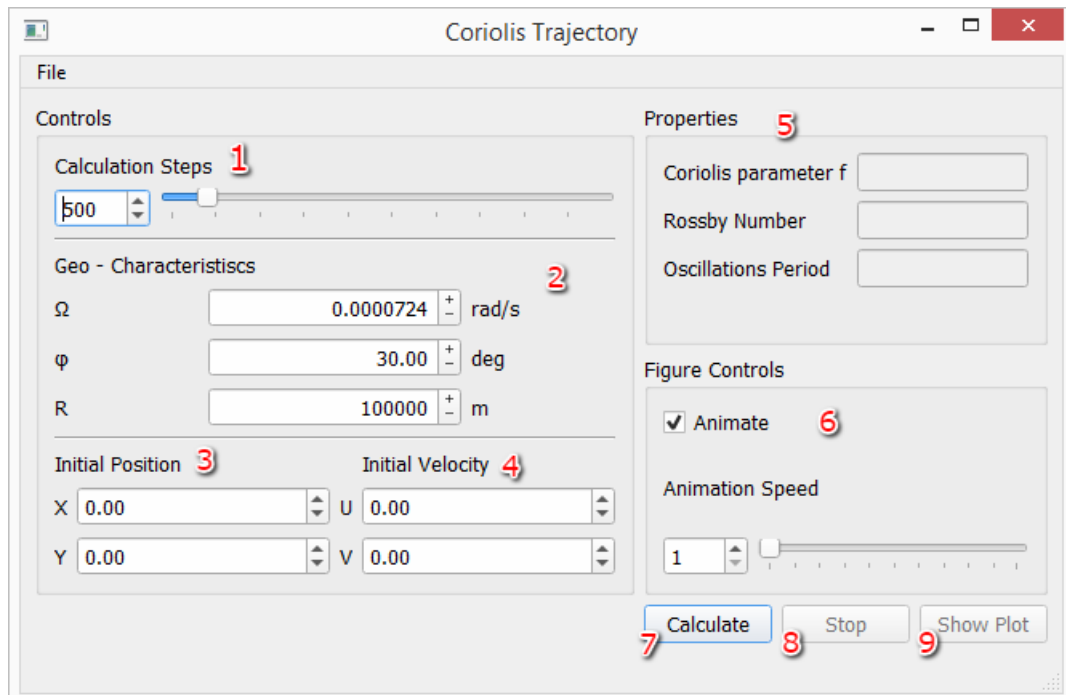


Figure 1: Γραφική διεπαφή χρήστη του λογισμικού "Coriolis Trajectory".

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

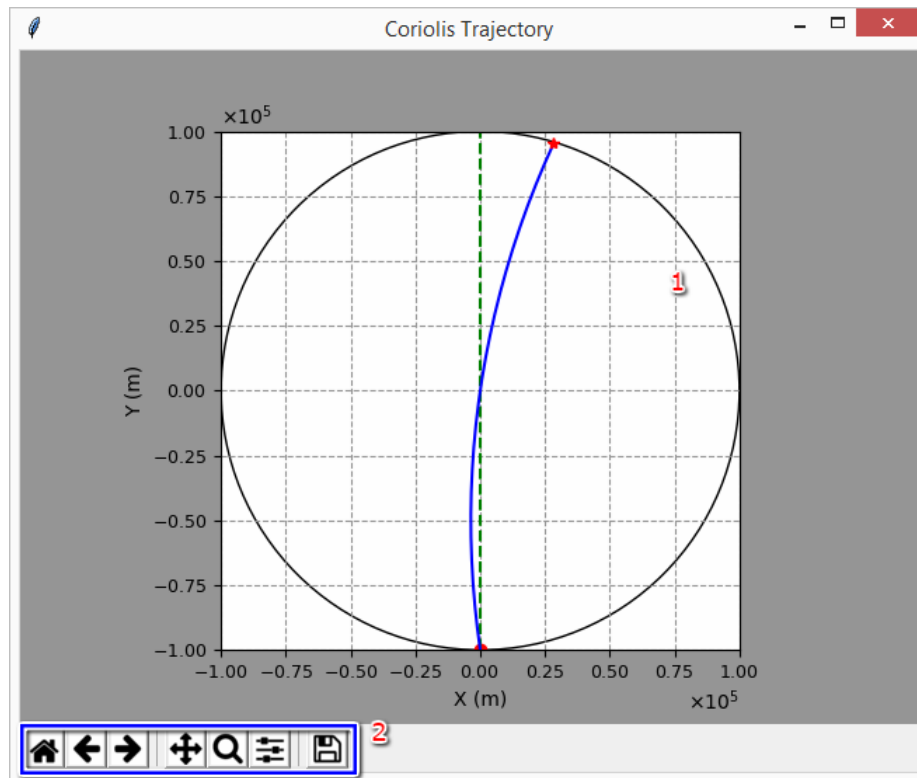


Figure 2: Παράθυρο γραφικής παράστασης του λογισμικού "Coriolis Trajectory"

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.

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. **[ΑΝΑΦΟΡΑ ΕΙΚΟΝΑΣ ΕΔΩ]**. 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. **[ΑΝΑΦΟΡΑ ΕΙΚΟΝΑΣ ΕΔΩ]**

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.

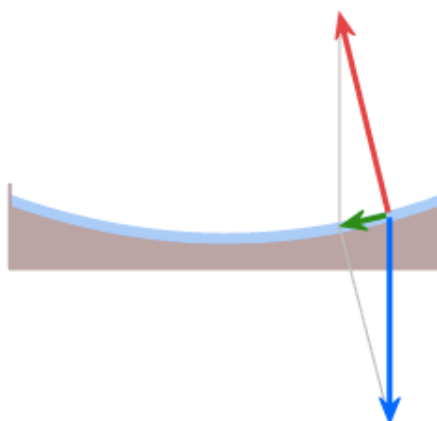


Figure 3: Οι δυνάμεις στο πείραμα του περιστρεφόμενου παραβολικού δίσκου. Teunissen (2017)

Στο πείραμα του περιστρεφόμενου παραβολικού δίσκου, Σχήμα 3, 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 Παλινδρομικής Ροής

Bibliography

- Haldoupis, C. (2015), *Εισαγωγή στην ατμοσφαιρική φυσική*, Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών.
- Kantha, L. H. & Clayson, C. A. (2000), *Numerical models of oceans and oceanic processes*, Academic 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.

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