



ΑΛΕΞΑΝΔΡΕΙΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**“Σχεδιασμός και ανάπτυξη λογισμικού για την επίλυση
συστήματος διαφορικών εξισώσεων”
(Κωδικός: 09205ΕΣ)**

Σπουδαστές: Βαφειάδης Αντώνιος (Κ.Α.Σ.: 502016)
Ρίτση Μαρία-Μυρτώ (Κ.Α.Σ.: 502558)

Επιβλέπουσα: Παπαδοπούλου Μαρία
Φυσικός-Ραδιοηλεκτρολόγος

Ανάληψη πτυχιακής εργασίας : Σεπτέμβριος 2009
Περάτωση πτυχιακής εργασίας : Ιούνιος 2010

Θεσσαλονίκη 2010

κενή σελίδα

Περίληψη

Σκοπός της πτυχιακής εργασίας είναι η δημιουργία ενός λογισμικού για τη συστηματική μελέτη της συμπεριφοράς του συστήματος διαφορικών εξισώσεων Lorenz. Η ανάπτυξη του λογισμικού γίνεται σε προγραμματιστικό περιβάλλον Matlab/ Simulink. Το Matlab/ Simulink είναι μία γλώσσα προγραμματισμού τέταρτης γενιάς κατάλληλη για μαθηματικές εφαρμογές και χρησιμοποιείται ευρέως στην ψηφιακή επεξεργασία σημάτων και στη μελέτη δυναμικών συστημάτων.

Το λογισμικό που αναπτύχθηκε προσομοιώνει τη συμπεριφορά του συστήματος Lorenz. Ο χρήστης έχει τη δυνατότητα, μέσω ενός εύχρηστου γραφικού περιβάλλοντος, να λαμβάνει μια σειρά από διαφορετικά γραφήματα, να ελέγχει συνεχώς τις παραμέτρους της προσομοίωσης και, τελικά, να εξάγει χρήσιμα συμπεράσματα για τη δυναμική του συστήματος. Επίσης, ο χρήστης μπορεί να μεταβάλλει τις παραμέτρους των εξισώσεων, να θέσει τις συνθήκες στις οποίες θέλει να επικεντρώσει τη μελέτη του και να λάβει αποτελέσματα σε μία, δύο ή και τρεις διαστάσεις.

Abstract

The purpose of this graduation project is the development of a software for the systematic study of Lorenz's equations system. The design and development of this software is in Matlab/Simulink, which is a fourth generation programming language suitable for mathematical applications. It is widely used in the study of digital signal processing and dynamical systems.

The final software simulates the behaviour of Lorenz's system. The user is able, via an intuitive graphical user interface, to gather a number of different charts, to continuously monitor the parameters of the simulation and, ultimately, to draw useful conclusions about the dynamics of the system. Also, the user could change the parameters of the equations, focus in specific simulation's conditions and observe results in one, two or three dimensions.

Πρόλογος

Ο Edward Lorenz (1917-2008) μελετώντας τη δυνατότητα πρόβλεψης των καιρικών φαινομένων και αναλύοντας τα κλιματολογικά δεδομένα κατέληξε σε ένα σύστημα τριών εξισώσεων, που πήραν και το όνομά του. Κατά την επίλυσή τους δημιουργούν μία περίπλοκη μορφή (fractal) στον τρισδιάστατο χώρο των φάσεων, που ονομάστηκε «ελκυστής Lorenz». Ο «ελκυστής» Lorenz είναι ένα γράφημα τριών διαστάσεων που παρουσιάζει, κάτω από ορισμένες προϋποθέσεις, χαοτική συμπεριφορά.

Οι Kevin Cuomo και Alan Oppenheim, ψάχνοντας να βρουν έναν τρόπο για την ασφαλή μετάδοση μηνυμάτων, κατασκεύασαν ένα ισοδύναμο ηλεκτρονικό κύκλωμα των εξισώσεων Lorenz. Το κύκλωμα δημιουργούσε ένα χαοτικό σήμα στον πομπό, το οποίο στη συνέχεια το προσθέτανε στο μήνυμα που ήθελαν να μεταδώσουν, ώστε σε περίπτωση υποκλοπής το μήνυμα να μην ήταν κατανοητό. Στο δέκτη, με ένα ίδιο κύκλωμα, δημιουργούσαν το ίδιο χαοτικό σήμα και το αφαιρούσαν από το μεταδιδόμενο σήμα, ώστε να ανακτήσουν το αρχικό μήνυμα.

Σκοπός αυτής της εργασίας είναι η δημιουργία ενός λογισμικού το οποίο θα προσομοιώνει τη συμπεριφορά των εξισώσεων Lorenz και θα δίνει στο χρήστη τη δυνατότητα της συστηματικής μελέτης τους. Με τη βοήθεια ενός εύχρηστου γραφικού περιβάλλοντος είναι εφικτή η μελέτη του συστήματος. Ο χρήστης έχει τη δυνατότητα να αλλάζει τις παραμέτρους του συστήματος, να θέτει τις συνθήκες στις οποίες θέλει να επικεντρώσει τη μελέτη του και να βλέπει τα αποτελέσματα σε πραγματικό χρόνο.

Στην παρούσα πτυχιακή μελετούμε το σύστημα Lorenz, τη χαοτική συμπεριφορά του και το ισοδύναμο ηλεκτρονικό κύκλωμα των Kevin Cuomo και Alan Oppenheim. Στη συνέχεια παρουσιάζεται ο σχεδιασμός και η ανάπτυξη του λογισμικού και των λειτουργιών του. Τέλος παρουσιάζουμε ορισμένα αριθμητικά αποτελέσματα, με τη μορφή γραφημάτων, που ελήφθησαν με τη χρήση του λογισμικού. Πιο συγκεκριμένα:

- Στο πρώτο κεφάλαιο γίνεται μια γενική αναφορά στο πρόγραμμα Matlab/Simulink και στα χρήσιμα εργαλεία που μας παρέχει (π.χ. τη δυνατότητα δημιουργίας γραφικού περιβάλλοντος), στις δυνατότητές και τους σκοπούς που εξυπηρετεί.
- Στο δεύτερο κεφάλαιο γίνεται μια γενική αναφορά στο μοντέλο Lorenz, καθώς και μια ιστορική αναδρομή σχετικά με το πώς ανακαλύφθηκε. Τέλος γίνεται επεξήγηση

ορισμένων όρων που είναι απαραίτητοι για την κατανόηση του μοντέλου από τον αναγνώστη.

- Στο τρίτο κεφάλαιο παρουσιάζεται και αναλύεται το ισοδύναμο ηλεκτρονικό κύκλωμα του συστήματος Lorenz, των Kevin Cuomo και Alan Oppenheim, καθώς και μια ιστορική αναδρομή σχετικά με το πώς ανακαλύφθηκε.
- Στο τέταρτο κεφάλαιο γίνεται η παρουσίαση του λογισμικού που αναπτύχθηκε. Ακολουθεί επεξήγηση της λειτουργίας του προγράμματος και των δυνατοτήτων του.
- Στο πέμπτο κεφάλαιο παρουσιάζονται ορισμένα αριθμητικά αποτελέσματα επίλυσης των εξισώσεων Lorenz με τη βοήθεια διαγραμμάτων, για διαφορετικές παραμέτρους και αριθμητικές μεθόδους.
- Τέλος, στο έκτο και τελευταίο κεφάλαιο παρουσιάζονται συμπεράσματα και προοπτικές βελτίωσης της δουλειάς που έχει γίνει.

Οι συγγραφείς
Αντώνιος Βαφειάδης, Μαρία-Μυρτώ Ρίτση

Θεσσαλονίκη, Μάιος 2010

ΠΕΡΙΕΧΟΜΕΝΑ

<i>Κεφάλαιο 1</i>	1
Εισαγωγή στο Matlab/ Simulink.....	1
1.1 Matlab.....	1
1.2 Simulink.....	2
1.3 Graphical User Interface (GUI).....	2
<i>Κεφάλαιο 2</i>	3
Μοντέλο Lorenz.....	3
2.1 Ο ελκυστής Lorenz	3
2.2 Χαοτική συμπεριφορά ελκυστή Lorenz	4
2.3 Ορισμός του χάους	6
<i>Κεφάλαιο 3</i>	8
Κύκλωμα Υλοποίησης Εξισώσεων Lorenz	8
3.1 Ιστορική αναδρομή.....	8
3.2 Ηλεκτρονικό κύκλωμα υλοποίησης εξισώσεων Lorenz	9
<i>Κεφάλαιο 4</i>	13
Λογισμικό Επίλυσης Εξισώσεων Lorenz.....	13
4.1 Εισαγωγή.....	13
4.2 Το σύστημα Lorenz στο Simulink.....	14
4.2.1 Υλοποίηση συστήματος στο Simulink	14
4.2.2 Ρύθμιση παραμέτρων συστήματος Lorenz.....	16
4.3 Δημιουργία γραφικού περιβάλλοντος	17
4.3.1 Εισαγωγή	17
4.3.2 Περιγραφή στοιχείων.....	18
4.3.3 Σχεδιασμός GUI	18
4.3.4 Παράμετροι συστήματος (System Parameters) – Figure 1	19
4.3.5 Χρόνος προσομοίωσης (Simulation Time) – Figure 1	20
4.3.6 Αριθμητική μέθοδος επίλυσης – Figure 1.....	20
4.3.7 Τρισδιάστατη απεικόνιση – Figure 1.....	21
4.3.8 Εμφάνιση εικόνας ισοδύναμου κυκλώματος – Figure 1.....	21
4.3.9 Πίνακας ελέγχου – Figure 1.....	21

4.3.10 Πρόοδος και κατάσταση προσομοίωσης – Figure 1	21
4.3.11 Φόρμα με μεταβαλλόμενο ρ – Figure 2	22
4.4 Παρουσίαση εφαρμογής	24
4.4.1 Εισαγωγή	24
4.4.2 Περιβάλλον εφαρμογής – Βασικό περιβάλλον	25
4.4.3 Περιβάλλον εφαρμογής – Μεταβλητό ρ	33
Κεφάλαιο 5	36
Αποτελέσματα Προσομοίωσης	36
5.1 Εισαγωγή	36
5.2 Προσομοιώσεις σταθερού βήματος με τη χρήση διαφορετικών αριθμητικών μεθόδων ολοκλήρωσης	37
5.3 Προσομοιώσεις μεταβλητού βήματος με τη χρήση διαφορετικών αριθμητικών μεθόδων ολοκλήρωσης	45
5.4 Προσομοιώσεις για διάφορες τιμές της παραμέτρου ρ	50
Κεφάλαιο 6	64
Συμπεράσματα	64
6.1 Ανακεφαλαίωση	64
6.2 Συμπεράσματα	65
6.3 Προοπτικές	65
ΑΝΑΦΟΡΕΣ	66

Κεφάλαιο 1

Εισαγωγή στο Matlab/ Simulink

1.1 Matlab

Το matlab (“matrix laboratory” = “εργαστήριο πινάκων”) είναι μια γλώσσα προγραμματισμού τέταρτης γενιάς που προορίζεται για μαθηματικές εφαρμογές και έχει ευρεία χρήση στον τομέα της ψηφιακής επεξεργασίας σημάτων. Το matlab παρέχει πολλές δυνατότητες στο χρήστη, όπως το χειρισμό πινάκων, τη σχεδίαση διαγραμμάτων συναρτήσεων ή δεδομένων, τη δημιουργία και εφαρμογή αλγορίθμων, τη δημιουργία περιβάλλοντος προσιτού στο χρήστη και τη δυνατότητα συνεργασίας με προγράμματα που βασίζονται σε άλλες γλώσσες προγραμματισμού. Επίσης περιλαμβάνει χρήσιμα εργαλεία, όπως είναι το simulink, για τη σχεδίαση και προσομοίωση μοντέλων συστημάτων και το graphical user interface (gui) για τη δημιουργία γραφικού περιβάλλοντος φιλικό προς το χρήστη.

Το matlab δημιουργήθηκε στα τέλη της δεκαετίας του 1970 από τον Cleve Moler, πρόεδρο του τμήματος ηλεκτρονικών υπολογιστών του πανεπιστημίου του Νέου Μεξικού. Το σχεδίασε για ακαδημαϊκούς σκοπούς και σύντομα διαδόθηκε σε άλλα πανεπιστήμια και απέκτησε στην πορεία πολλούς υποστηρικτές στον τομέα των εφαρμοσμένων μαθηματικών. Το 1983 ο μηχανικός Jack Little γνώρισε το matlab και αναγνωρίζοντας την εμπορική του αξία, μαζί με τους Cleve Moler και Steve Bangert, ξαναέγραψαν το matlab σε γλώσσα C και ίδρυσαν την εταιρεία The Mathworks το 1984 ώστε να συνεχίσουν την ανάπτυξή του.

Ο πρώτος τομέας στον οποίο βρήκε εφαρμογή το matlab ήταν στα συστήματα αυτοματισμού, σύντομα όμως διαδόθηκε σε πολλές άλλες ειδικότητες. Σήμερα

χρησιμοποιείται και στον τομέα της εκπαίδευσης, πιο συγκεκριμένα στη διδασκαλία γραμμικής άλγεβρας και αριθμητικής ανάλυσης, και είναι δημοφιλές σε όσους ασχολούνται με την επεξεργασία εικόνας. Το 2004 η Mathworks ισχυρίζεται ότι το matlab χρησιμοποιήθηκε από περισσότερους από ένα εκατομμύριο ανθρώπους του βιομηχανικού και ακαδημαϊκού κόσμου.

1.2 Simulink

Το simulink είναι ένα πολύ χρήσιμο εργαλείο που προσφέρει το matlab και δίνει τη δυνατότητα στο χρήστη για τη σχεδίαση, προσομοίωση και ανάλυση δυναμικών συστημάτων. Στο κύριο περιβάλλον του βρίσκει κανείς το εργαλείο σχεδιασμού μπλοκ διαγραμμάτων καθώς και μεγάλο αριθμό βιβλιοθηκών. Αυτές οι βιβλιοθήκες ανανεώνονται συνεχώς από τη mathworks παρέχοντας στο χρήστη καινούρια εξαρτήματα και δυνατότητες σχεδιασμού. Το simulink συνεργάζεται με το matlab και μπορεί κανείς είτε να το χρησιμοποιήσει σαν εργαλείο του matlab, είτε αυτόνομα. Το simulink χρησιμοποιείται ευρέως στα συστήματα αυτοματισμού και στην επεξεργασία ψηφιακών σημάτων για τη σχεδίαση και προσομοίωσή τους.

1.3 Graphical User Interface (GUI)

Ακόμη μία δυνατότητα που μας προσφέρει το matlab είναι η δημιουργία γραφικού περιβάλλοντος χρήστη (Graphical User Interface). Το γραφικό περιβάλλον χρήστη είναι μία πιο προσιτή στο χρήστη επικοινωνία με το πρόγραμμα σε σχέση με την απλή πληκτρολόγηση εντολών. Το περιβάλλον αυτό προσφέρει στο χρήστη εικόνες και γραφικές συντομεύσεις που αντιπροσωπεύουν πλήρως τις επιλογές του, τις λειτουργίες και τις πληροφορίες του προγράμματος. Ο χρήστης δε χρειάζεται να λειτουργεί το πρόγραμμα ψάχνοντας πλήθος εντολών και μεγάλης έκτασης κώδικα, αλλά έχει απευθείας πρόσβαση στις λειτουργίες του προγράμματος μέσω των γραφικών στοιχείων.

Κεφάλαιο 2

Μοντέλο Lorenz

2.1 Ο ελκυστής Lorenz

Ο ελκυστής Lorenz (Lorenz attractor), που πήρε το όνομά του από τον Edward N. Lorenz (1917-2008), αποτελεί μία δομή fractal που είναι αντίστοιχη με το αποτέλεσμα της συμπεριφοράς του ταλαντωτή Lorenz. Η δομή fractal (fractal structure) είναι ένα περίπλοκο γεωμετρικό σχήμα που όσο και αν μεγεθυνθεί παρουσιάζει ένα εξίσου περίπλοκο σχέδιο με μερική ή ολική επανάληψη του αρχικού. Ο ταλαντωτής Lorenz είναι ένα δυναμικό σύστημα τριών διαστάσεων που παρουσιάζει χαοτική συμπεριφορά και είναι γνωστό για το γεωμετρικό του σχήμα.

Ο ελκυστής και οι εξισώσεις Lorenz, από τις οποίες προκύπτει αυτή η μορφή, παρουσιάστηκαν από τον Edward Lorenz το 1963. Κατέληξε σε αυτό το σύστημα τριών διαστάσεων από ένα απλοποιημένο μοντέλο της διάδοσης θερμότητας και μάζας στην ατμόσφαιρα. Οι εξισώσεις Lorenz μελετήθηκαν αρχικά με σκοπό την πρόβλεψη του καιρού και την ανάλυση των κλιματολογικών συνθηκών και πέρα από την αξία και το ενδιαφέρον τους στον τομέα των μαθηματικών συναντώνται και σε άλλες εφαρμογές όπως σε απλοποιημένα μοντέλα laser (Haken 1975) και ηλεκτρογεννητριών (Knobloch 1981).

Το σύστημα Lorenz απαρτίζεται από τις παρακάτω εξισώσεις:

$$\frac{dx}{dt} = \sigma(y - x) \quad (1.1)$$

$$\frac{dy}{dt} = \rho x - y - xz \quad (1.2)$$

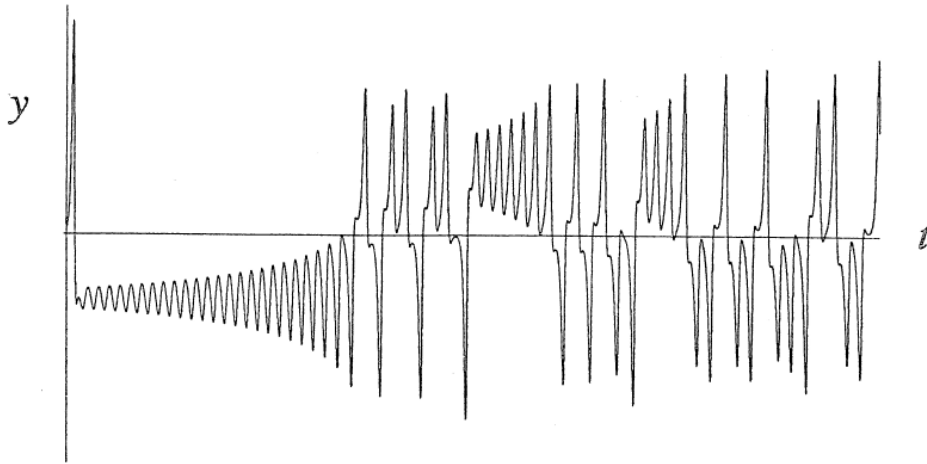
$$\frac{dz}{dt} = xy - \beta z \quad (1.3)$$

όπου σ , ρ , β είναι οι παράμετροι του συστήματος. Το σ ονομάζεται αριθμός Prandtl, το ρ ονομάζεται αριθμός Rayleigh και το β δεν έχει όνομα. Τα σ , ρ και β είναι θετικοί αριθμοί και συνήθως οι τιμές τους είναι $\sigma=10$, $\beta=8/3$, ενώ το ρ μεταβάλλεται. Το σύστημα παρουσιάζει χαοτική συμπεριφορά για $\rho=28$ αλλά παρουσιάζει περιοδικές τροχιές για άλλες τιμές του ρ .

Ο Lorenz ανακάλυψε ότι αυτό το φαινομενικά απλό σύστημα μπορεί να έχει ακανόνιστη συμπεριφορά: για μεγάλο πλήθος παραμέτρων, οι λύσεις του συστήματος παρουσιάζουν ακανόνιστες διακυμάνσεις, χωρίς ποτέ να επαναλαμβάνονται, αλλά συνεχώς παραμένουν εντός ενός συγκεκριμένου χώρου στο χώρο των φάσεων. Όταν σχεδίασε το διάγραμμα των λύσεων του συστήματος στις τρεις διαστάσεις, διαπίστωσε ότι δημιουργείται ένα πολύπλοκο σχήμα που ονόμασε «ελκυστή». Σε αντίθεση με τα διαγράμματα που απεικονίζουν σταθερά σημεία και οριοθετημένους κύκλους, ο ελκυστής δεν είναι κάποιο συνηθισμένο σχήμα αλλά ένα μορφόκλασμα.

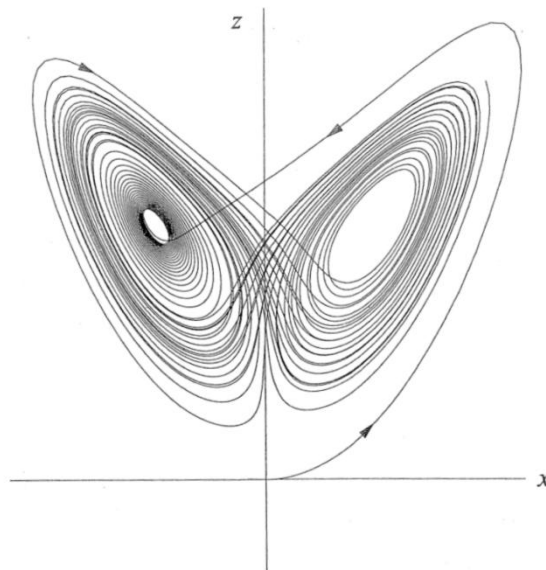
2.2 Χαοτική συμπεριφορά ελκυστή Lorenz

Ο Lorenz δοκίμασε εφαρμόζοντας αριθμητική ολοκλήρωση να βρει πώς θα απεικονιστεί το διάγραμμα των λύσεων των εξισώσεων σε μεγάλο χρονικό διάστημα. Μελέτησε την περίπτωση όπου οι σταθερές είχαν τις τιμές $\sigma=10$, $\beta=8/3$ και $\rho=28$. Την τιμή αυτή του ρ τη διάλεξε ώστε να είναι λίγο μεγαλύτερη από την τιμή $r_h = \sigma \cdot \frac{\sigma + \beta + 3}{\sigma - \beta - 1} \approx 24,74$, που είναι η τιμή του θεωρήματος της διακλάδωσης Hopf (Hopf bifurcation), περιμένοντας ότι έτσι θα προέκυπτε κάποιο περίεργο αποτέλεσμα. Φυσικά κάτι περίεργο θα μπορούσε να συμβεί για οποιοδήποτε λόγο αφού οι υπολογιστές εκείνων των ημερών δεν ήταν ούτε αξιόπιστοι ούτε εύκολοι στη χρήση και ο Lorenz θα έπρεπε να ερμηνεύσει τα αποτελέσματά του με προσοχή. Ξεκίνησε την ολοκλήρωση με αρχικές συνθήκες $(-1, 1, 0)$. Στο σχήμα 1.1 παρουσιάζεται η γραφική παράσταση στην οποία κατέληξε για το $y(t)$. Μετά από την αρχική κατάσταση, το γράφημα παρουσιάζει μια ακανόνιστη ταλάντωση που συνεχίζεται καθώς $t \rightarrow \infty$, αλλά ποτέ δεν επαναλαμβάνεται ακριβώς. Η ταλάντωση δεν είναι περιοδική.



Σχήμα 1.1: Αποτελέσματα Lorenz για το $y(t)$

Ο Lorenz παρατήρησε ότι δημιουργείται ένα όμορφο σχήμα όταν βλέπουμε το διάγραμμα στο χώρο των φάσεων. Για παράδειγμα, αν απεικονίσουμε το $z(t)$ ως προς το $x(t)$, προκύπτει ένα σχήμα που μοιάζει με πεταλούδα (βλ. σχήμα 1.2).



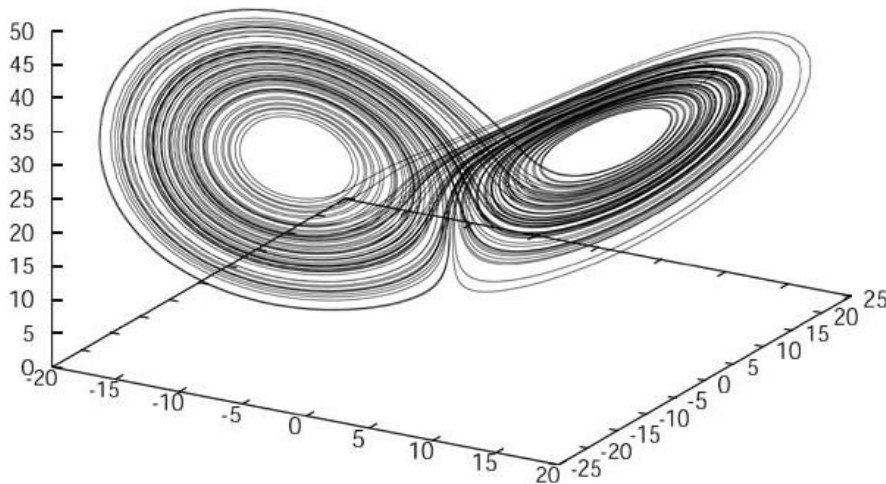
Σχήμα 1.2: Ελκυστής Lorenz

Η τροχιά των σημείων φαίνεται να “διασταυρώνεται” με τον εαυτό της συνεχώς αλλά αυτό είναι αποτέλεσμα της απεικόνισης μίας τρισδιάστατης τροχιάς σε επίπεδο δύο διαστάσεων. Αν προβάλλαμε το διάγραμμα σε τρεις διαστάσεις δε θα υπήρχαν αλληλοεπικαλύψεις.

Αναλύοντας το διάγραμμα παρατηρούμε ότι η τροχιά των σημείων ξεκινάει από ένα σημείο πολύ κοντά στην αρχή των αξόνων, ανεβαίνει πάνω και δεξιά και μετά βυθίζεται στο

κέντρο των σπειρών που βρίσκονται αριστερά. Μετά το σχηματισμό μίας σπείρας στα αριστερά επιστρέφει στη δεξιά πλευρά και σχηματίζει μερικές σπείρες. Στη συνέχεια επιστρέφει στη δεξιά πλευρά και σχηματίζει μερικές ακόμα σπείρες κ.ο.κ. Ο αριθμός των περιστροφών που κάνει σε κάθε πλευρά αλλάζει απρόβλεπτα από τον έναν κύκλο στον επόμενο και για την ακρίβεια η συχνότητα του αριθμού των κύκλων που διαγράφει είναι τυχαία.

Στην απεικόνιση στις τρεις διαστάσεις (βλ. σχήμα 1.3) φαίνεται ότι δημιουργούνται επιφάνειες οι οποίες καλύπτουν η μία την άλλη αλλά στην πραγματικότητα αυτό είναι μία παραίσθηση. Αυτό συμβαίνει εξαιτίας του μεγάλου αριθμού σημείων στο διάγραμμα και της ανεπαρκούς ανάλυσης του. Στην πραγματικότητα έχουμε άπειρες επιφάνειες, τη μία πολύ κοντά στην άλλη, και έτσι δημιουργείται το πολύπλοκο τελικό σχήμα (μορφόκλασμα – fractal structure). Είναι ένα σύνολο σημείων με μηδενικό όγκο και άπειρη έκταση επιφάνειας. Τα αριθμητικά πειράματα έδειξαν ότι έχει διάσταση περίπου 2.05.



Σχήμα 1.3: Τρισδιάστατη απεικόνιση ελκυστή Lorenz

2.3 Ορισμός του χάους

Κανένας ορισμός της έννοιας του χάους δεν έχει σήμερα παγκόσμια αποδοχή, αλλά σχεδόν όλοι θα συμφωνούσαν στα τρία συστατικά που αναφέρονται στην ερμηνεία που ακολουθεί.

Το χάος είναι μία απεριοδική μακροπρόθεσμη συμπεριφορά ενός ντετερμινιστικού (αιτιοκρατικού) συστήματος που παρουσιάζει ευαίσθητη εξάρτηση από τις αρχικές συνθήκες.

- 1) “Απεριοδική μακροπρόθεσμη συμπεριφορά” σημαίνει ότι οι σχηματισμοί των σημείων λύσεων του συστήματος καθώς ο χρόνος τείνει στο άπειρο ($t \rightarrow \infty$) δεν συμπίπτουν με σταθερά σημεία, περιοδικές τροχιές ή εν μέρει περιοδικές τροχιές.
- 2) “Ντετερμινιστικό” ή αιτιοκρατικό σημαίνει ότι το σύστημα δεν έχει εισόδους ή παραμέτρους τυχαίες ή με θόρυβο. Η ακανόνιστη συμπεριφορά δεν είναι αποτέλεσμα αυτών των εισόδων αλλά προκύπτει από τη μη γραμμικότητα του συστήματος και όχι από το θόρυβο.
- 3) “Ευαίσθητη εξάρτηση από τις αρχικές συνθήκες” σημαίνει ότι οι αποστάσεις ανάμεσα σε διαδοχικά κομμάτια της διαγραφόμενης τροχιάς ή σημεία μπορεί να μεταβάλλονται εκθετικά.

Κεφάλαιο 3

Κύκλωμα Υλοποίησης Εξισώσεων Lorenz

3.1 Ιστορική αναδρομή

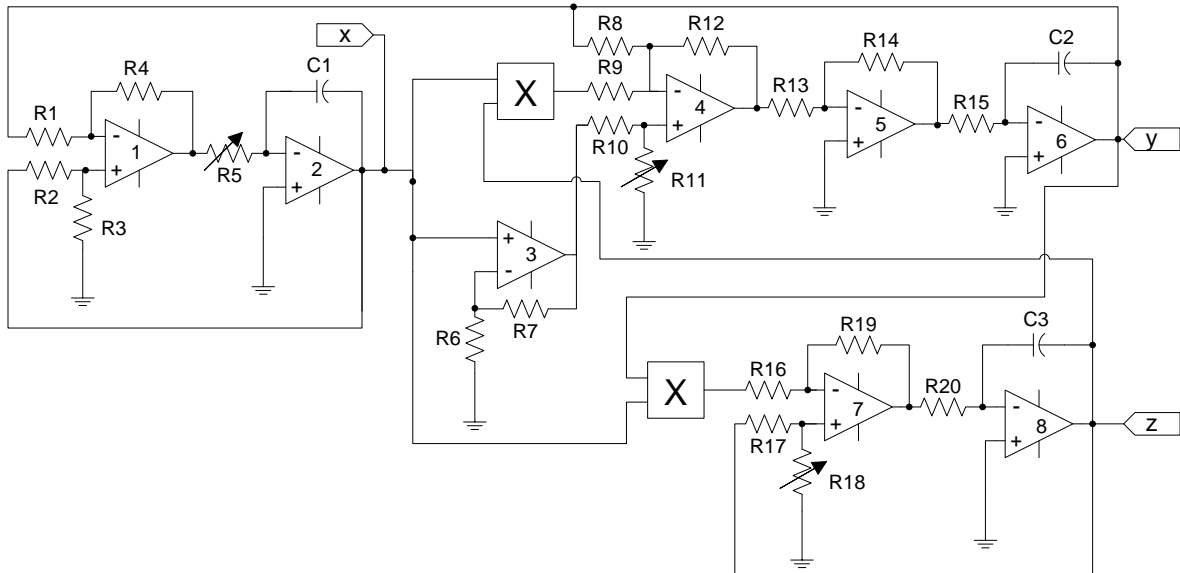
Μία από τις πιο εντυπωσιακές πρόσφατες ανακαλύψεις στον τομέα της μη γραμμικής δυναμικής είναι η συνειδητοποίηση ότι το χάος μπορεί να φανεί χρήσιμο. Κανονικά, κάποιος θα θεωρούσε το χάος σαν ένα συναρπαστικό και παράδοξο φαινόμενο στην καλύτερη περίπτωση ή σαν μία ανοησία στη χειρότερη, σε κάθε περίπτωση σαν κάτι που πρέπει να αποφεύγεται. Από το 1990 όμως, οι άνθρωποι έχουν ανακαλύψει τρόπους να εκμεταλλευτούν τις ιδιότητες του χάους για τη δημιουργία εντυπωσιακών και πρακτικών εφαρμογών. Μία εφαρμογή αφορά στις απόρρητες επικοινωνίες, ένα πρόβλημα που απασχολεί τους ανθρώπους από τότε που άρχισαν να έχουν μυστικά που αξίζει να κρατήθουν κρυφά.

Ο Kevin Cuomo και ο Alan Oppenheim (1992, 1993) έκαναν μία νέα προσέγγιση σε αυτό το πρόβλημα βασιζόμενοι στην ανακάλυψη των Pecora και Carroll (1990) ότι δύο χαοτικά συστήματα είναι δυνατόν να συγχρονιστούν. Το σκεπτικό είναι το εξής: Όταν κάποιος στέλνει ένα μήνυμα σε κάποιον, το καλύπτει με ένα χαοτικό σήμα με μεγαλύτερη ένταση. Αν προσπαθήσει κάποιος άλλος να ακούσει το μήνυμα θα ακούσει μόνο το χαοτικό σήμα, δηλαδή ένα θόρυβο χωρίς νόημα. Αν ο παραλήπτης του μηνύματος έχει ένα δέκτη ο οποίος παράγει το χαοτικό σήμα, μπορεί να το αφαιρέσει από το μήνυμα που λαμβάνει και να ανακτήσει το αρχικό μήνυμα.

Ο Kevin Cuomo έκανε μία επίδειξη της προσέγγισής του, τη δημιουργία δηλαδή του χαοτικού σήματος, κατασκευάζοντας μία ηλεκτρονική εφαρμογή των εξισώσεων Lorenz.

3.2 Ηλεκτρονικό κύκλωμα υλοποίησης εξισώσεων Lorenz

Το ηλεκτρονικό κύκλωμα υλοποίησης του συστήματος εξισώσεων Lorenz παρουσιάζεται στο σχήμα 3.1.



Σχήμα 3.1: Ηλεκτρονικό κύκλωμα υλοποίησης εξισώσεων Lorenz

Στο παραπάνω κύκλωμα οι τάσεις στα σημεία x , y , z είναι τα αντίστοιχα των μεταβλητών x , y , z των εξισώσεων Lorenz. Για να υπολογίσουμε τις τιμές των τάσεων στα σημεία x , y , z αναλύουμε το κύκλωμα. Εφαρμόζοντας το θεώρημα της υπέρθεσης στο κύκλωμα του τελεστικού 1 έχουμε:

$$V_{o1+} = \left(1 + \frac{R_4}{R_1}\right) \left(\frac{R_3}{R_2 + R_3}\right) \cdot x \quad (3.1)$$

$$V_{o1-} = -\left(\frac{R_4}{R_1}\right) \cdot y \quad (3.2)$$

$$V_{o1} = V_{o1+} + V_{o1-} = \left(1 + \frac{R_4}{R_1}\right) \left(\frac{R_3}{R_2 + R_3}\right) \cdot x - \left(\frac{R_4}{R_1}\right) \cdot y \quad (3.3)$$

Στην έξοδο του ολοκληρωτή που αποτελείται από τον τελεστικό 2 και τα στοιχεία R_5 και C_1 έχουμε:

$$V_{o2} = -\left(\frac{1}{R_5 C_1}\right) \cdot V_{o1} \quad (3.4)$$

Έτσι καταλήγουμε στη σχέση για το \dot{x} που είναι η αντίστοιχη της εξίσωσης Lorenz (1.1):

$$\dot{x} = V_{o2} = -\left(\frac{1}{R_5 C_1}\right) \cdot \left[\left(1 + \frac{R_4}{R_1}\right) \left(\frac{R_3}{R_2 + R_3}\right) \cdot x - \left(\frac{R_4}{R_1}\right) \cdot y \right] \quad (3.5)$$

Στην έξοδο του μη αναστρέφοντος τελεστικού ενισχυτή 3 έχουμε:

$$V_{o3} = \left(1 + \frac{R_7}{R_6}\right) \cdot x \quad (3.6)$$

Εφαρμόζοντας το θεώρημα της υπέρθεσης για τον τελεστικό 4:

$$V_{o4+} = \left(1 + \frac{\frac{R_{12}}{R_8 R_9}}{\frac{R_8 + R_9}{R_8 + R_9}}\right) \left(\frac{R_{11}}{R_{10} + R_{11}}\right) \cdot V_{o3} = \left(1 + \frac{R_{12}}{R_8} + \frac{R_{12}}{R_9}\right) \left(\frac{R_{11}}{R_{10} + R_{11}}\right) \cdot V_{o3} \quad (3.7)$$

$$V_{o4-(1)+} = -\frac{R_{12}}{R_9} xz \quad (3.8)$$

$$V_{o4-(2)+} = -\frac{R_{12}}{R_8} y \quad (3.9)$$

Στην έξοδο του τελεστικού 4 έχουμε:

$$V_{o4} = \left(1 + \frac{R_{12}}{R_8} + \frac{R_{12}}{R_9}\right) \left(\frac{R_{11}}{R_{10} + R_{11}}\right) \cdot V_{o3} - \frac{R_{12}}{R_9} xz - \frac{R_{12}}{R_8} y \quad (3.10)$$

Στην έξοδο του αναστρέφοντος τελεστικού ενισχυτή 5:

$$V_{o5} = -\frac{R_{14}}{R_{13}} \cdot V_{o4} \quad (3.11)$$

Στην έξοδο του ολοκληρωτή που αποτελείται από τον τελεστικό 6 και τα στοιχεία R_{15} και C_2 έχουμε:

$$V_{o6} = -\frac{1}{R_{15}C_2} \cdot V_{o5} \quad (3.12)$$

Έτσι καταλήγουμε στη σχέση για το \dot{y} που είναι η αντίστοιχη της εξίσωσης Lorenz (1.2):

$$\dot{y} = V_{o6} = \frac{1}{R_{15}C_2} \cdot \frac{R_{14}}{R_{13}} \left[\left(1 + \frac{R_{12}}{R_8} + \frac{R_{12}}{R_9} \right) \left(\frac{R_{11}}{R_{10} + R_{11}} \right) \cdot \left(1 + \frac{R_7}{R_6} \right) \cdot x - \frac{R_{12}}{R_9} xz - \frac{R_{12}}{R_8} y \right] \quad (3.13)$$

Εφαρμόζοντας το θεώρημα της υπέρθεσης για τον τελεστικό 7:

$$V_{o7+} = \left(1 + \frac{R_{19}}{R_{16}} \right) \left(\frac{R_{18}}{R_{17} + R_{18}} \right) \cdot z \quad (3.14)$$

$$V_{o7-} = -\left(\frac{R_{19}}{R_{16}} \right) \cdot xy \quad (3.15)$$

Άρα στην έξοδο του τελεστικού 7 έχουμε:

$$V_{o7} = \left(1 + \frac{R_{19}}{R_{16}} \right) \left(\frac{R_{18}}{R_{17} + R_{18}} \right) \cdot z - \left(\frac{R_{19}}{R_{16}} \right) \cdot xy \quad (3.16)$$

Στην έξοδο του ολοκληρωτή που αποτελείται από τον τελεστικό 8 και τα R_{20} και C_3 έχουμε:

$$V_{o8} = -\frac{1}{R_{20}C_3} \cdot V_{o7} \quad (3.17)$$

Έτσι καταλήγουμε στη σχέση για το \dot{z} που είναι η αντίστοιχη της εξίσωσης Lorenz (1.3):

$$\dot{z} = V_{o8} = \frac{1}{R_{20}C_3} \left[\left(\frac{R_{19}}{R_{16}} \right) \cdot xy - \left(1 + \frac{R_{19}}{R_{16}} \right) \left(\frac{R_{18}}{R_{17} + R_{18}} \right) \cdot z \right] \quad (3.18)$$

Από τις παραπάνω εξισώσεις συμπεραίνουμε ότι μεταβάλλοντας τις μεταβλητές αντιστάσεις R_5 , R_{11} και R_8 , μεταβάλλουμε τις παραμέτρους σ , ρ και β αντίστοιχα του συστήματος Lorenz. Επίσης μεταβάλλοντας τους πυκνωτές C_1 , C_2 και C_3 , μεταβάλλουμε τους χρόνους του κυκλώματος.

Κεφάλαιο 4

Λογισμικό Επίλυσης Εξισώσεων Lorenz

4.1 Εισαγωγή

Στο κεφάλαιο που ακολουθεί θα παρουσιάσουμε το λογισμικό που σχεδιάστηκε και προσομοιώνει το σύστημα Lorenz. Ο χρήστης με τη βοήθεια ενός εύχρηστου γραφικού περιβάλλοντος έχει τον πλήρη έλεγχο της προσομοίωσης. Είναι σε θέση να εισάγει δεδομένα και να αλλάζει μεθόδους ολοκλήρωσης με σκοπό τη μελέτη της συμπεριφοράς του συστήματος.

Το πρόγραμμα δέχεται δεδομένα από το χρήστη. Πιο συγκεκριμένα, ο χρήστης επιλέγει τις τιμές των παραμέτρων σ , β και ρ , που είναι οι παράμετροι των εξισώσεων Lorenz:

$$\frac{dx}{dt} = \sigma(y - x) \quad (4.1)$$

$$\frac{dy}{dt} = \rho x - y - xz \quad (4.2)$$

$$\frac{dz}{dt} = xy - \beta z \quad (4.3)$$

Επιλέγει επίσης το χρόνο έναρξης και λήξης της προσομοίωσης, τη μαθηματική μέθοδο επίλυσης του συστήματος των διαφορικών εξισώσεων, την επιλογή του τύπου βήματος ολοκλήρωσης (σταθερό ή μεταβλητό) καθώς επίσης και την τιμή του βήματος ολοκλήρωσης.

Με το πάτημα ενός κουμπιού ανοίγει το μοντέλο προσομοίωσης (στο simulink) και μεταφέρει τα δεδομένα που δέχτηκε από το χρήστη. Σύμφωνα με αυτά ξεκινάει την


προσομοίωση και προβάλλει τις γραφικές παραστάσεις των λύσεων του συστήματος. Ο χρήστης έχει τη δυνατότητα να “παγώσει” την προσομοίωση ή να τη σταματήσει, όποτε αυτός επιθυμεί. Μπορεί επίσης να σώσει αυτά τα διαγράμματα (σε μορφή εικόνας .jpg) και τις τιμές των μεταβλητών τους (σε πίνακα σε μορφή κειμένου .txt) έτσι ώστε να έχει πρόσβαση σε αυτά αργότερα και εκτός περιβάλλοντος matlab.

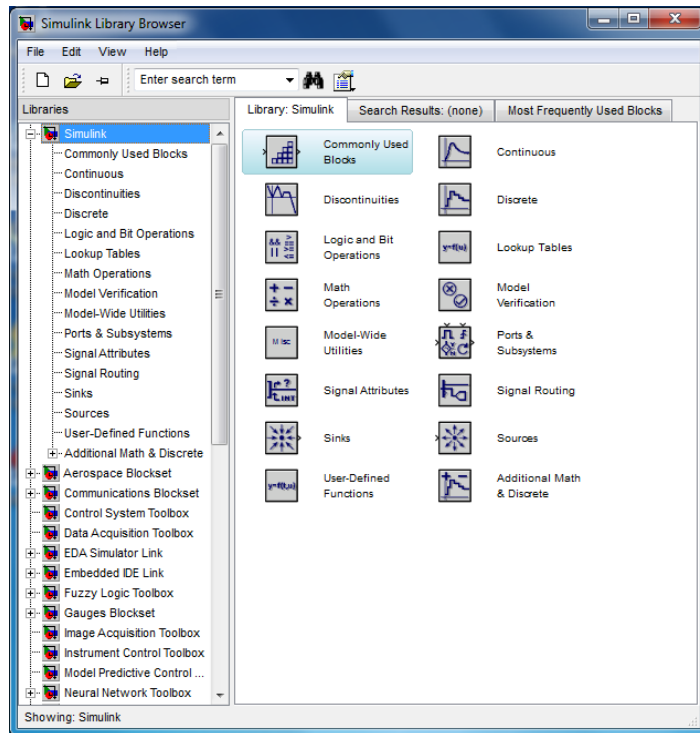
Τα αποτελέσματα της προσομοίωσης μπορεί ο χρήστης να τα εμφανίσει και σε τρισδιάστατο γράφημα ώστε να μελετήσει καλύτερα ή πιο αποτελεσματικά τη συμπεριφορά του μοντέλου. Του προσφέρεται δυνατότητα αποθήκευσης και του τρισδιάστατου γραφήματος.

Μία επιπλέον λειτουργία του προγράμματος είναι η μελέτη του συστήματος με μεταβαλλόμενη την παράμετρο ρ . Πατώντας ένα κουμπί εμφανίζεται ένα καινούριο παράθυρο όπου ο χρήστης εισάγει την αρχική και τελική τιμή για το ρ καθώς και το βήμα μεταβολής του. Με το πάτημα ενός κουμπιού ξεκινάει η προσομοίωση και εμφανίζονται οι γραφικές παραστάσεις των λύσεων του συστήματος, τις οποίες μπορεί ο χρήστης να μελετήσει και να αποθηκεύσει.

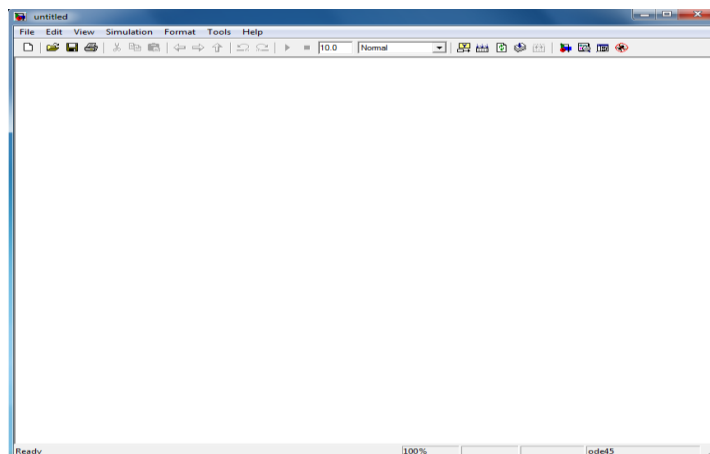
4.2 Το σύστημα Lorenz στο Simulink

4.2.1 Υλοποίηση συστήματος στο Simulink

Μέσα από το κεντρικό περιβάλλον του matlab επιλέγουμε το κατάλληλο εικονίδιο () ώστε να μπούμε στο περιβάλλον του simulink. Το πρώτο πράγμα που βλέπουμε είναι οι βιβλιοθήκες με τα εξαρτήματα που μας παρέχει το simulink (βλ. σχήμα 4.1). Πατώντας file → new → model δημιουργείται ο χώρος σχεδίασης μέσα στον οποίο θα υλοποιήσουμε το μπλοκ διάγραμμα του συστήματος (βλ. σχήμα 4.2). Για τη σχεδίαση επιλέγουμε κάθε εξάρτημα που θα χρησιμοποιήσουμε από τις βιβλιοθήκες και τα τοποθετούμε στο χώρο σχεδίασης. Αφού τοποθετήσουμε όλα τα εξαρτήματα τα οποία χρειαζόμαστε για το σύστημά μας ενώνουμε τα εξαρτήματα μεταξύ τους με αγωγούς.

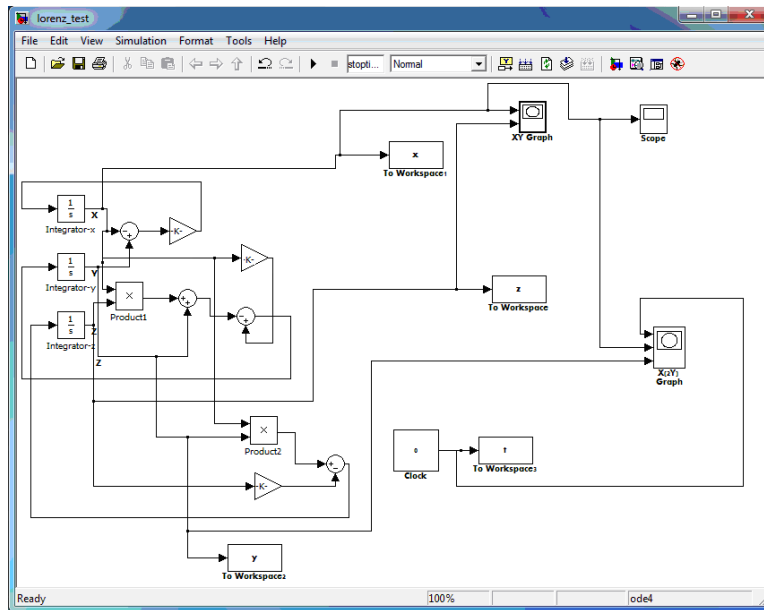


Σχήμα 4.1: Αρχικό παράθυρο Simulink



Σχήμα 4.2: Επιφάνεια εργασίας Simulink

Επιλέγουμε όλα τα στοιχεία που θα χρειαστούμε από τις βιβλιοθήκες του simulink (π.χ. ολοκληρωτές, πολλαπλασιαστές, αθροιστές κ.λπ.) και τα τοποθετούμε στην επιφάνεια εργασίας του Simulink. Πραγματοποιούμε τις απαραίτητες ρυθμίσεις για το σύστημά μας και στη συνέχεια ενώνουμε τα επιμέρους στοιχεία. Τοποθετούμε ό,τι άλλο στοιχείο μπορεί να χρειαστούμε στη μελέτη του συστήματός μας, π.χ. γραφήματα, ρολόι, score κ.λπ. Με αυτόν τον τρόπο λαμβάνουμε το τελικό υλοποιημένο σύστημα που παρουσιάζεται στο σχήμα 4.3.



Σχήμα 4.3: Σύστημα Lorenz


4.2.2 Ρύθμιση παραμέτρων συστήματος Lorenz

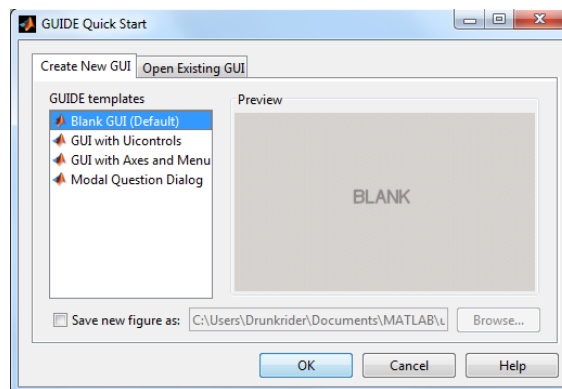
Το simulink μας δίνει τη δυνατότητα να επεξεργαστούμε τις παραμέτρους της λειτουργίας της προσομοίωσης του συστήματος. Στο μοντέλο που δημιουργήσαμε, επιλέγοντας Simulation → Configuration Parameters, ανοίγει ένα παράθυρο μέσα από το οποίο μπορούμε να ρυθίσουμε διάφορες παραμέτρους σχετικά με την προσομοίωση. Πολλές από αυτές τις παραμέτρους θα τις ελέγχουμε από το πρόγραμμά μας.

Στην κατηγορία Solver βρίσκονται οι παράμετροι που αφορούν στη διαδικασία εκτέλεσης της προσομοίωσης. Αρχίζοντας από την υποκατηγορία Simulation Time βλέπουμε τις παραμέτρους Start Time και Stop Time. Η παράμετρος Start Time δηλώνει τη χρονική στιγμή έναρξης της προσομοίωσης, ενώ η Stop Time δηλώνει τη χρονική στιγμή λήξης της. Έχουμε επίσης τη δυνατότητα επιλογής Solver Options, δηλαδή τον τύπο (Type) της προσομοίωσης (σταθερού ή μεταβλητού βήματος). Η κάθε επιλογή από αυτές έχει τις δικές της παραμέτρους. Έχοντας επιλεγμένη την παράμετρο Fixed-step οι παράμετροι που εμφανίζονται είναι οι Solver και Fixed-step size. Έχοντας επιλεγμένη την παράμετρο Variable-step, οι παράμετροι που μας ενδιαφέρουν είναι οι Solver, Max step size και Min step size. Η παράμετρος Solver μας δίνει όπως και στην προηγούμενη περίπτωση τη δυνατότητα να επιλέξουμε την αριθμητική μέθοδο επίλυσης του συστήματος (π.χ. ode45, ode23 κ.λπ.). Κάθε μέθοδος από αυτές δίνει διαφορετική ακρίβεια στους υπολογισμούς μας. Αυτό θα φανεί και στις προσομοιώσεις που θα κάνουμε.

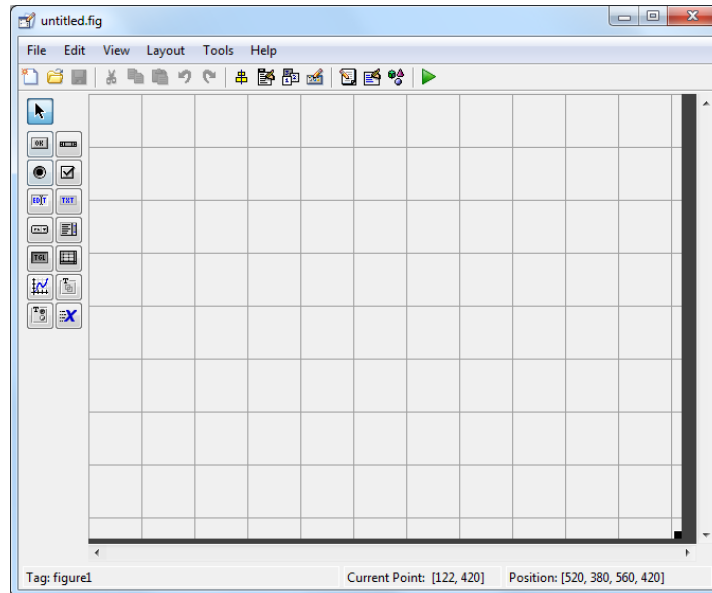
4.3 Δημιουργία γραφικού περιβάλλοντος

4.3.1 Εισαγωγή

Στο κύριο περιβάλλον του matlab κάνουμε κλικ στο κατάλληλο εικονίδιο (Guide ) ώστε να μπούμε στο παράθυρο δημιουργίας γραφικού περιβάλλοντος (βλ. σχήματα 4.4 και 4.5). Σε αυτό το χώρο θα τοποθετήσουμε τα κατάλληλα στοιχεία που θα εξυπηρετήσουν τις λειτουργίες του προγράμματος.



Σχήμα 4.4: Επιλογή κενού GUI



Σχήμα 4.5: Χώρος σχεδίασης γραφικού περιβάλλοντος

Στα αριστερά του παραθύρου που εμφανίζεται βρίσκονται τα στοιχεία που μπορούμε να προσθέσουμε στο figure, όπως push button, edit text, static text, pop-up menu κλπ. Όλα αυτά τα στοιχεία είναι τροποποιήσιμα (μέγεθος, όνομα, χρώμα κ.λπ.) έτσι ώστε να τα

επεξεργαστούμε σύμφωνα με τις ανάγκες μας. Για να ρυθμίσουμε τις παραμέτρους των στοιχείων χρησιμοποιούμε το παράθυρο Properties Inspector το οποίο εμφανίζεται κάνοντας διπλό κλικ πάνω στο κάθε στοιχείο που έχουμε βάλει στο figure.

4.3.2 Περιγραφή στοιχείων

Τα στοιχεία που θα χρησιμοποιήσουμε στο figure που θα δημιουργήσουμε είναι τα ακόλουθα: Push Button, Edit Text, Static Text, Pop-up Menu, Axes, Panel και Button Group, μια περιγραφή των οποίων ακολουθεί.

Push Button: Προσθέτουμε τον κατάλληλο κώδικα ώστε να μπορεί να εκτελέσει διάφορες λειτουργίες.

Toggle Button: Κουμπί το οποίο έχει δύο καταστάσεις, πατημένο ή μη πατημένο, και παραμένει σε κάθε κατάσταση μέχρι ο χρήστης να το αλλάξει.

Edit Text: Κείμενο το οποίο ο χρήστης του προγράμματος μπορεί να τροποποιήσει έτσι ώστε να εισάγει τα δεδομένα που θέλει για την εκτέλεση του προγράμματος.

Static Text: Στατικό κείμενο, δηλαδή κείμενο το οποίο ο χρήστης δεν έχει τη δυνατότητα να τροποποιήσει και παραμένει σταθερό.

Pop-up Menu: Μενού με κάποιες επιλογές που έχει καθορίσει ο προγραμματιστής και αποτελούν τις επιλογές που έχει ο χρήστης κατά τη διάρκεια εκτέλεσης του.

Axes: Σύστημα αξόνων δύο ή τριών διαστάσεων πάνω στο οποίο μπορούμε να εμφανίσουμε κάποια γραφική παράσταση.

Panel: Πλαίσιο που μπορεί να βάλει ο προγραμματιστής ώστε να ομαδοποιήσει κάποια από τα στοιχεία που βρίσκονται στο figure.

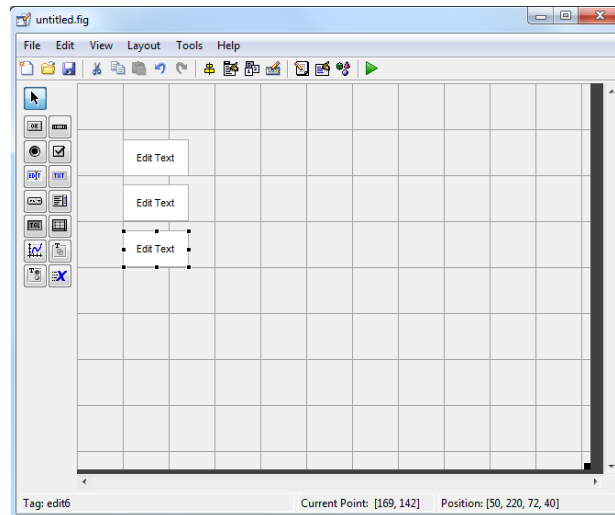
Button Group: Πλαίσιο παρόμοιο με το Panel, με τη διαφορά ότι χρησιμοποιείται για να οργανώσει γενικά κουμπιά εκτέλεσης εντολών.

4.3.3 Σχεδιασμός GUI

Στο σχεδιασμό του προγράμματος χρησιμοποιήσουμε δύο figures, το ένα είναι το βασικό πρόγραμμα (figure1), ενώ το άλλο (figure2) καλείται μέσα από το βασικό πρόγραμμα. Στη συνέχεια παραθέτουμε τη διαδικασία σχεδιασμού χωρίζοντας το κάθε figure σε τμήματα, ανάλογα με τις λειτουργίες που εξυπηρετούν. Θα ξεκινήσουμε από το βασικό figure.

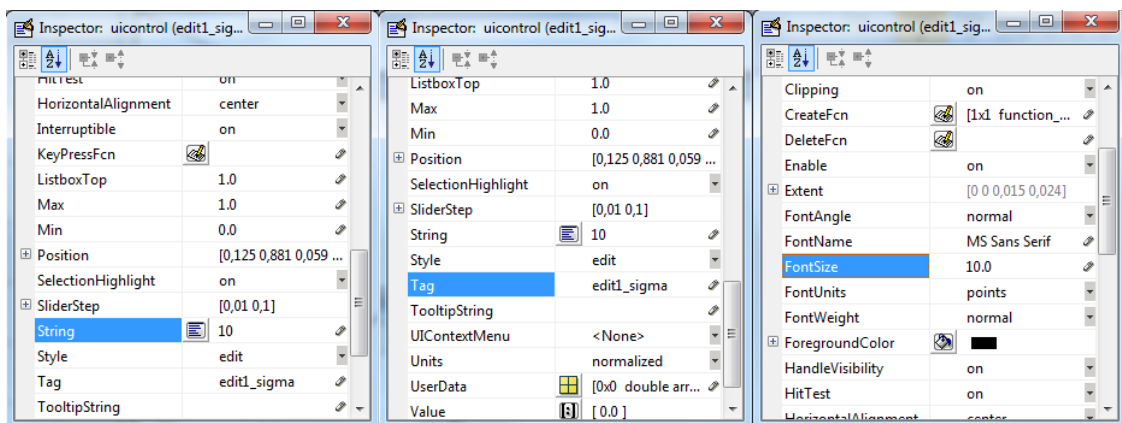
4.3.4 Παράμετροι συστήματος (System Parameters) – Figure 1

Ξεκινάμε από το κομμάτι που αφορά στην εισαγωγή των τριών παραμέτρων σ , β και ρ . Εισάγουμε τρία Edit Texts στα οποία θα δηλώνονται οι τρεις παράμετροι του συστήματος (βλ. σχήμα 4.6).



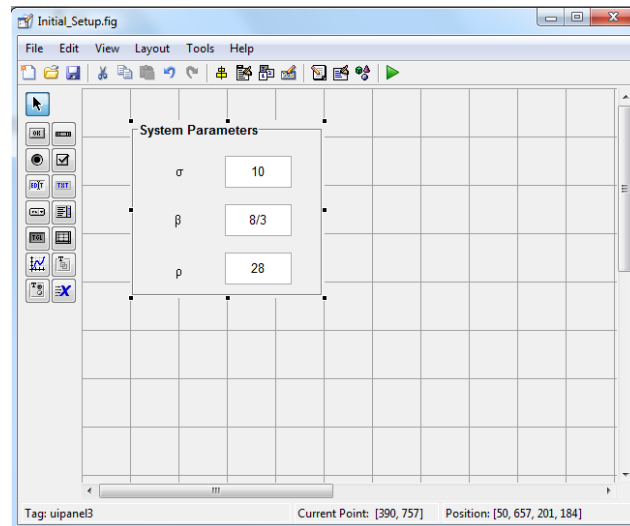
Σχήμα 4.6: Γραφικό περιβάλλον με τρία Edit Texts

Κάνοντας διπλό κλικ πάνω στο Edit Text ανοίγουμε το παράθυρο Properties Inspector μέσα από το οποίο θα τροποποιήσουμε τις ιδιότητές του (βλ. σχήμα 4.7). Αρχικοποιούμε τα τρία Edit Text με τις τιμές 10, $8/3$ και 28 για τα σ , β και ρ αντίστοιχα, αφού αυτές είναι οι τιμές τις οποίες θέλουμε να μελετήσουμε κυρίως στο πρόγραμμα καθώς για αυτές τις τιμές το σύστημα Lorenz εμφανίζει την αξιοσημείωτη συμπεριφορά που αναφέραμε παραπάνω. Φυσικά ο χρήστης όταν “τρέχει” το πρόγραμμα θα μπορεί να αλλάξει αυτές τις τιμές και να βάλει όποιες ο ίδιος επιθυμεί.



Σχήμα 4.7: Αλλαγή παραμέτρων για το Edit Text στο οποίο εισάγεται η μεταβλητή σ

Στη συνέχεια προσθέτουμε τρία Static Texts, ένα για κάθε Edit Text, στα οποία θα γράψουμε τα ονόματα των μεταβλητών ώστε να γνωρίζει ο χρήστης το κάθε Edit Text σε ποια μεταβλητή αντιστοιχεί. Για να ομαδοποιήσουμε τα παραπάνω στοιχεία που χρησιμοποιούνται για την εισαγωγή των παραμέτρων του συστήματος (System Parameters) τοποθετούμε ένα Panel (βλ. σχήμα 4.8).



Σχήμα 4.8: Εισαγωγή Panel για την ομαδοποίηση των στοιχείων

4.3.5 Χρόνος προσομοίωσης (Simulation Time) – Figure 1

Σε αυτό το κομμάτι του προγράμματος δίνεται η δυνατότητα στο χρήστη να επιλέξει τις χρονικές στιγμές έναρξης και λήξης της προσομοίωσης. Τοποθετούμε δύο Edit Texts στα οποία ο χρήστης θα εισάγει την αρχική και την τελική χρονική στιγμή. Αρχική και τελική τιμή των παραπάνω χρόνων επιλέγονται 0 και 500 αντίστοιχα

4.3.6 Αριθμητική μέθοδος επίλυσης – Figure 1

Σε αυτήν την ομάδα στοιχείων πρώτα βάζουμε δύο Pop-up Menus. Στο πρώτο θα δίνεται στο χρήστη η δυνατότητα να επιλέξει αν θέλει η επίλυση του συστήματος να γίνει με μέθοδο σταθερού βήματος (Fixed-step) ή μεταβλητού (Variable-step). Στο δεύτερο Pop-up Menu ανάλογα με το τύπο βήματος χρόνου που έχει επιλέξει στο πρώτο Pop-up Menu, θα εμφανίζονται οι δυνατοί μέθοδοι επίλυσης του συστήματος. Δηλαδή αν στο πρώτο Pop-up Menu έχει επιλέξει Fixed Step στο δεύτερο θα εμφανίζονται οι επιλογές ode4, ode5 και ode3, ενώ αν στο πρώτο έχει επιλέξει Variable Step στο δεύτερο θα εμφανίζονται οι επιλογές ode23 και ode45.

4.3.7 Τρισδιάστατη απεικόνιση – Figure 1

Για την τρισδιάστατη απεικόνιση του ελκυστή του συστήματος τοποθετούμε στο figure το στοιχείο Axes. Ρυθμίζουμε το μέγεθος που επιθυμούμε κατά την τοποθέτησή του ή και αφού το έχουμε εισάγει στο figure επεκτείνοντάς το από κάποια γωνία ή άκρη με τη βοήθεια του ποντικιού. Στη συνέχεια προσθέτουμε τρία Push Buttons με τα οποία ο χρήστης θα εκτελεί εντολές που αφορούν στο τρισδιάστατο γράφημα. Πιο συγκεκριμένα βάζουμε ένα Push Button το οποίο θα σχεδιάζει το τρισδιάστατο γράφημα στους άξονες, ένα Push Button το οποίο θα καθαρίζει τους άξονες και ένα Push Button το οποίο θα αποθηκεύει το γράφημα

4.3.8 Εμφάνιση εικόνας ισοδύναμου κυκλώματος – Figure 1

Για να γίνει πιο ευπαρουσίαστο και ολοκληρωμένο το πρόγραμμα προσθέτουμε την εικόνα του ισοδύναμου κυκλωματικού του συστήματος Lorenz. Αυτό το υλοποιούμε χρησιμοποιώντας το στοιχείο Axes, το οποίο μας δίνει τη δυνατότητα να εμφανίσουμε μια εικόνα. Προσθέτουμε επομένως στο figure ένα στοιχείο Axes και προγραμματιστικά του δίνουμε την εντολή να εμφανίσει την εικόνα που θέλουμε.

4.3.9 Πίνακας ελέγχου – Figure 1

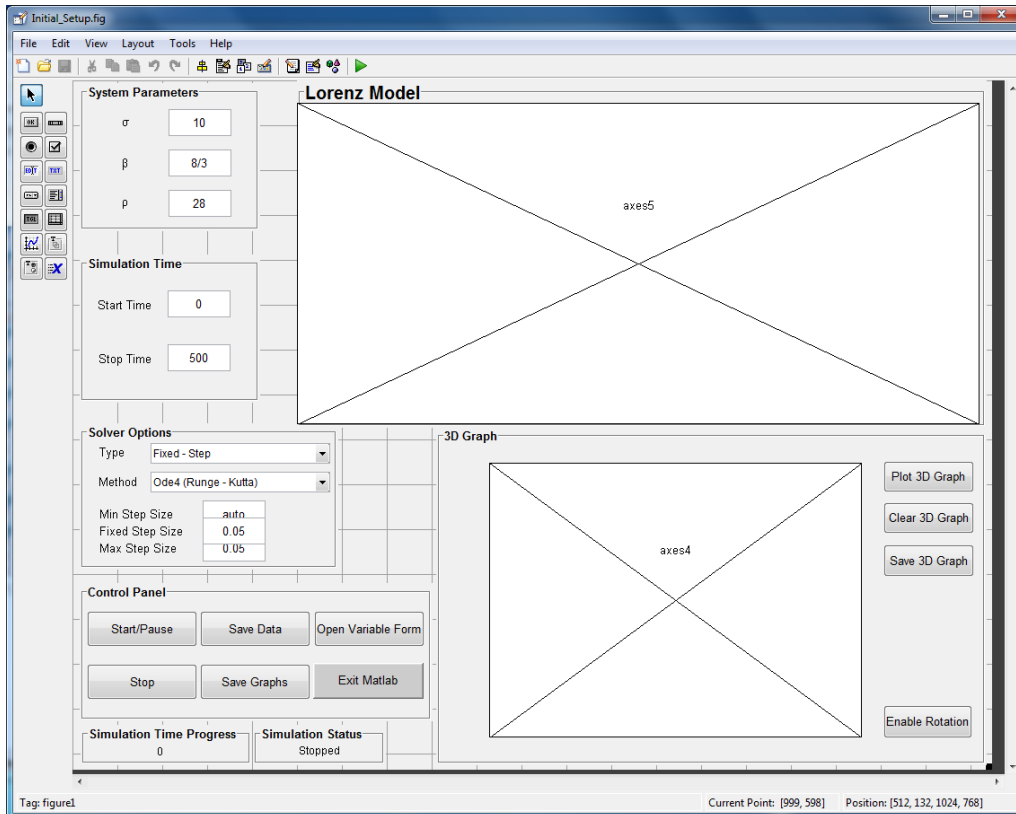
Αυτό το κομμάτι του figure αποτελείται από τα Push Buttons που εκτελούν εντολές που ελέγχουν την προσομοίωση. Τοποθετούμε έξι Push Buttons. Το πρώτο δίνει τη δυνατότητα στο χρήστη να ξεκινάει και να “παγώνει” την προσομοίωση. Το δεύτερο Push Button σταματάει την προσομοίωση. Το τρίτο Push Button δίνει τη δυνατότητα αποθήκευσης των δεδομένων x , y , z και t σε αρχείο κειμένου (.txt). Το τέταρτο Push Button δίνει τη δυνατότητα αποθήκευσης των γραφημάτων xz , xy , yz και $x(t)$. Στη συνέχεια τοποθετούμε ένα Push Button το οποίο θα ανοίγει το δεύτερο figure που θα είναι η φόρμα στην οποία θα λειτουργεί η προσομοίωση με μεταβαλλόμενο ρ . Τέλος προσθέτουμε και ένα Push Button το οποίο θα κλείνει το πρόγραμμα. Οργανώνουμε όλα τα Push Buttons μέσα σε ένα Button Group ,με την ονομασία Control Panel.

4.3.10 Πρόδος και κατάσταση προσομοίωσης – Figure 1

Για να ολοκληρώσουμε το πρώτο figure βάζουμε ένα Static Text το οποίο θα δείχνει την πρόοδο της προσομοίωσης με τον κατάλληλο προγραμματιστικό κώδικα και το αρχικοποιούμε βάζοντας $\text{String} \rightarrow 0$. Τέλος βάζουμε ένα Static Text το οποίο θα δείχνει την

κατάσταση στην οποία βρίσκεται η προσομοίωση (stopped, running, paused) και το αρχικοποιούμε βάζοντας String \rightarrow Stopped.

Πραγματοποιώντας τις παραπάνω διαδικασίες σχεδιασμού, ανάπτυξης και προγραμματισμού, καταλήγουμε στη φόρμα του σχήματος 4.9.



Σχήμα 4.9: Τελική όψη της πρώτης φόρμας (Initial_Setup.fig)

4.3.11 Φόρμα με μεταβαλλόμενο ρ – Figure 2

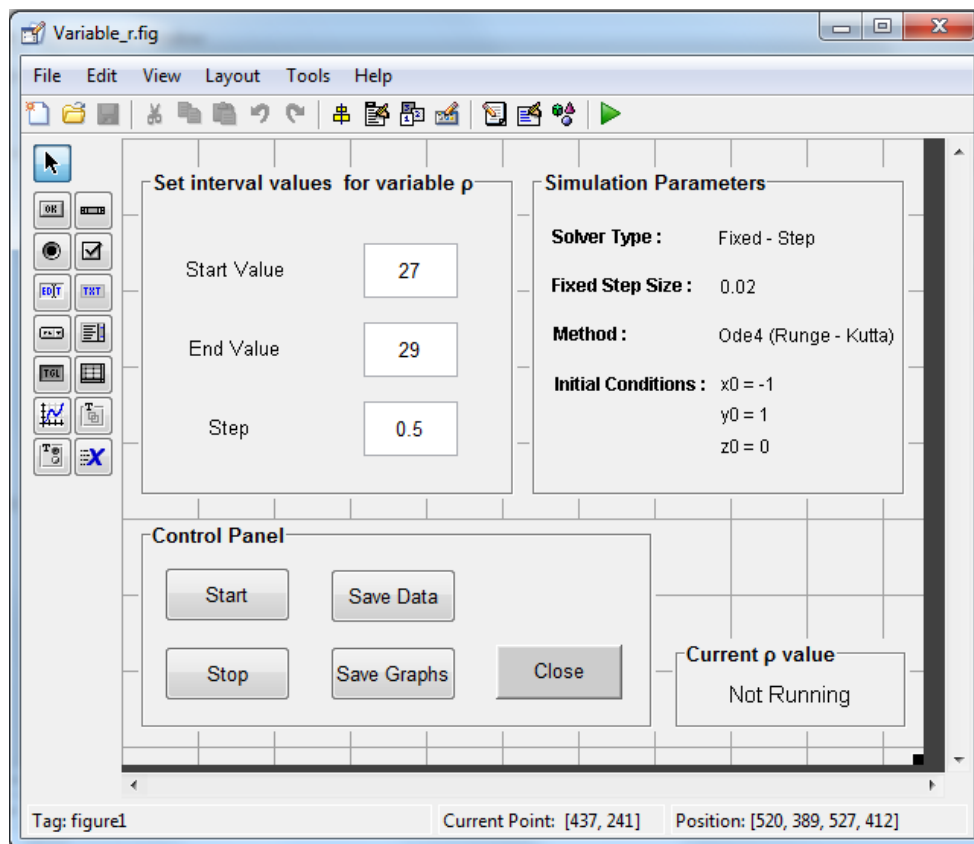
Σε αυτή τη φόρμα ο χρήστης θα δηλώνει την αρχική τιμή της μεταβλητής ρ , την τελική τιμή της μεταβλητής ρ και το βήμα με το οποίο θα αυξάνεται η μεταβλητή ρ για κάθε 100 δείγματα της κάθε τιμής. Για παράδειγμα αν ο χρήστης δώσει αρχική τιμή 27, τελική τιμή 29 και βήμα 0.5 το πρόγραμμα θα υπολογίζει την έξοδο και θα σχεδιάζει τη γραφική παράσταση στην οποία τα 100 πρώτα δείγματα θα είναι για $\rho = 27$, τα επόμενα 100 για $\rho = 27,5$, τα επόμενα 100 για $\rho = 28$ κ.ο.κ.

Θα ξεκινήσουμε από το κομμάτι όπου ο χρήστης θα δηλώνει την αρχική τιμή, την τελική τιμή και το βήμα για τη μεταβλητή ρ . Χρησιμοποιούμε τρία Edit Texts στα οποία θέτουμε και τις προεπιλεγμένες τιμές. Για να γνωρίζει ο χρήστης που θα εισάγει τις τιμές που

αφορούν στην μεταβλητή ρ , βάζουμε τρία Static Texts. Τέλος, περικλείουμε τα παραπάνω σε ένα Panel.

Στη συνέχεια προσθέτουμε τα Push Buttons με τα οποία θα ελέγχονται οι λειτουργίες του προγράμματος. Βάζουμε ένα Push Button το οποίο θα ξεκινάει την προσομοίωση, ένα Push Button το οποίο θα σταματάει την προσομοίωση και δύο Push Buttons τα οποία θα διαχειρίζονται την αποθήκευση των δεδομένων και των γραφικών παραστάσεων που θέλουμε. Τέλος βάζουμε ένα Push Button το οποίο θα κλείνει τη φόρμα και θα επιστρέφουμε στο κύριο πρόγραμμα. Οργανώνουμε όλα τα Push Buttons μέσα σε ένα Panel με την ονομασία Control Panel. Προσθέτουμε Static Texts που αναγράφουν τις συνθήκες κάτω από τις οποίες πραγματοποιείται η προσομοίωση. Τέλος προσθέτουμε ένα Static Text στο οποίο θα αναγράφεται η τιμή του ρ για την οποία εκτελείται η προσομοίωση κάθε στιγμή και το περικλείουμε σε ένα panel με τίτλο Current ρ Value.

Αφού τελειώσουμε με τη σχεδίαση αποθηκεύουμε το figure με την ονομασία Variable_r.fig (βλ. σχήμα 4.10).



Σχήμα 4.10: Τελική μορφή δεύτερης φόρμας (Variable_r.fig)

4.4 Παρουσίαση εφαρμογής

4.4.1 Εισαγωγή

Το λογισμικό, που παρουσιάζεται στη συνέχεια, αναπτύχθηκε εξολοκλήρου σε Matlab/ Simulink. Με τη βοήθεια ενός εύχρηστου γραφικού περιβάλλοντος, ο χρήστης μπορεί να μελετήσει τη συμπεριφορά του συστήματος Lorenz. Πιο συγκεκριμένα, δίνεται στο χρήστη η δυνατότητα να δώσει τις επιθυμητές τιμές για τις παραμέτρους σ , ρ και β του συστήματος Lorenz, αλλά και να επιλέξει μεταξύ διαφορετικών αριθμητικών μεθόδων επίλυσης διαφορικών εξισώσεων.

Ο χρήστης εισάγει τις παραμέτρους του συστήματος και της προσομοίωσης στο γραφικό περιβάλλον που έχει δημιουργηθεί. Με την έναρξη της προσομοίωσης, οι τιμές αυτές εισάγονται στο μοντέλο Simulink, χωρίς να απαιτείται επιπλέον ρύθμιση στο περιβάλλον του Simulink. Όλες οι απαραίτητες αλλαγές πραγματοποιούνται με αυτοματοποιημένο τρόπο στο παρασκήνιο.

Κατά τη διάρκεια της προσομοίωσης εμφανίζονται γραφήματα, σε πραγματικό χρόνο, ώστε να μπορεί ο χρήστης να παρατηρεί τη συμπεριφορά και την εξέλιξη των μεταβλητών του συστήματος κάθε χρονική στιγμή. Παρέχεται επίσης η δυνατότητα εμφάνισης τρισδιάστατου γραφήματος των μεταβλητών, ώστε να μπορεί να παρατηρήσει ο χρήστης τη συμπεριφορά του συστήματος στο χώρο των φάσεων. Ο χρήστης μπορεί να αποθηκεύσει όλα τα παραπάνω διαγράμματα σε μορφή εικόνας .jpg, καθώς και τις μεταβλητές του συστήματος σε πίνακα σε αρχείο εξόδου .txt.

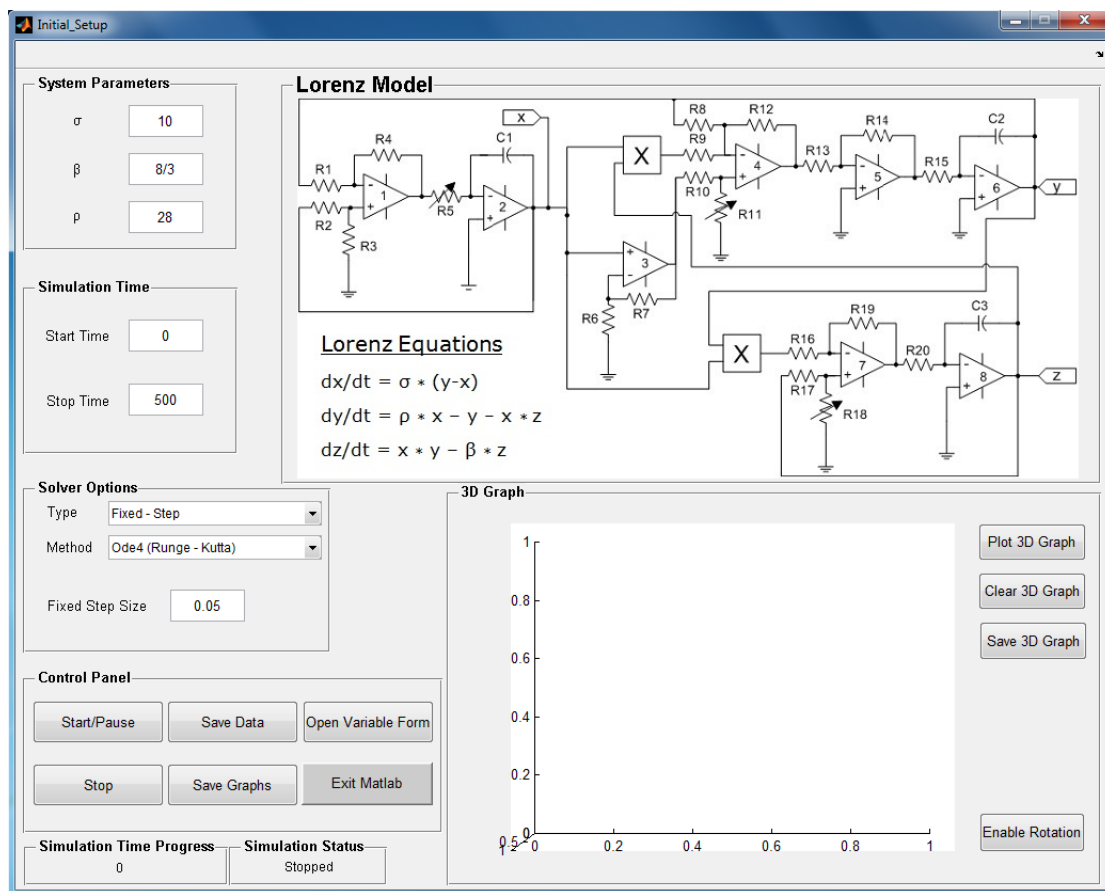
Ο χρήστης μπορεί να παγώσει την προσομοίωση, οποιαδήποτε χρονική στιγμή, αλλά και να την συνεχίσει ή να την σταματήσει, αν το κρίνει αναγκαίο. Υπάρχει συνεχής ενημέρωση, καθ' όλη τη διάρκεια της προσομοίωσης, για τη χρονική στιγμή στην οποία βρίσκεται αυτή με ένα Static Text, το οποίο ανανεώνεται ανά τακτά χρονικά διαστήματα.

Τέλος, υπάρχει μία επιπλέον εφαρμογή στο πρόγραμμα για τη μελέτη της επίδρασης της παραμέτρου ρ στη συμπεριφορά του συστήματος. Πιο συγκεκριμένα, ο χρήστης μπορεί να επιλέξει την επιθυμητή αρχική και τελική τιμή για την παράμετρο ρ , καθώς και το βήμα με το οποίο αυτή θα μεταβάλλεται ανά 100 χρονικά σημεία. Οι υπόλοιπες παράμετροι του συστήματος και της προσομοίωσης έχουν προεπιλεγεί και παραμένουν σταθερές. Για παράδειγμα, αν ο χρήστης επιλέξει για την παράμετρο ρ αρχική τιμή 25, τελική τιμή 27 και βήμα 0.5, η προσομοίωση θα τρέξει ως εξής: τη χρονική περίοδο 0-100 για $\rho=25$, τη χρονική

περίοδο 100-200 για $\rho=25.5$, τη χρονική περίοδο 200-300 για $\rho=26$, τη χρονική περίοδο 300-400 για $\rho=26.5$ και, τέλος, τη χρονική περίοδο 400-500 για $\rho=27$, οπότε και θα σταματήσει. Στην επόμενη παράγραφο παρουσιάζονται αναλυτικά οι δυνατότητες της εφαρμογής.

4.4.2 Περιβάλλον εφαρμογής – Βασικό περιβάλλον

Μόλις ο χρήστης ανοίξει το πρόγραμμα εμφανίζεται στην οθόνη του το βασικό περιβάλλον του προγράμματος (σχήμα 4.11).

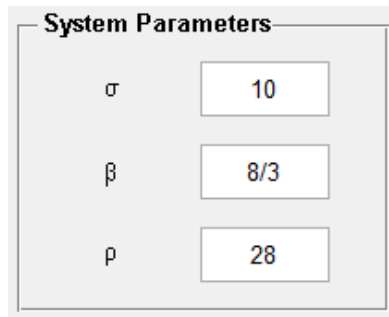


Σχήμα 4.11: Βασικό περιβάλλον του προγράμματος

Στο βασικό περιβάλλον του προγράμματος βλέπουμε τα εξής επιμέρους τμήματα (panels): System Parameters, Simulation Time, Solver Options, Control Panel, Simulation Time Progress, Simulation Status, 3D Graph και Lorenz Model.

Μέσα από το panel System Parameters ο χρήστης μπορεί να εισάγει, σε κατάλληλα πεδία, τις τιμές των παραμέτρων σ , ρ και β του συστήματος. Προεπιλεγμένες είναι οι τιμές 10, $8/3$ και 28 για τις παραμέτρους σ , ρ και β αντίστοιχα, που είναι και οι τιμές, για τις οποίες το σύστημα Lorenz παρουσιάζει την αξιοσημείωτη συμπεριφορά που περιγράψαμε

στα προηγούμενα κεφάλαια, αλλά ο χρήστης μπορεί να εισάγει οποιαδήποτε αριθμητική τιμή επιθυμεί (σχήμα 4.12). Αξίζει να σημειώσουμε ότι το πρόγραμμα σχεδιάστηκε έτσι ώστε να δέχεται μόνο αριθμητικές τιμές σε αυτά τα πεδία. Αν ο χρήστης εισάγει κάποιο γράμμα σε κάποιο από τα πεδία το πρόγραμμα το λαμβάνει σαν τον αριθμό 0. Το ίδιο ισχύει και για όλα τα πεδία τα οποία δέχονται αριθμητικές τιμές και όχι γράμματα.



The image shows a panel titled "System Parameters" with three input fields. The first field is labeled with the Greek letter sigma (σ) and contains the value "10". The second field is labeled with the Greek letter beta (β) and contains the value "8/3". The third field is labeled with the Greek letter rho (ρ) and contains the value "28".

Σχήμα 4.12: Εισαγωγή των σ , ρ και β στο panel System Parameters

Μέσα από το panel Simulation Time ο χρήστης εισάγει τις επιθυμητές τιμές για τους χρόνους έναρξης (Start Time) και λήξης (Stop Time) της προσομοίωσης. Προεπιλεγμένες είναι οι τιμές 0 και 500 για το χρόνο έναρξης και λήξης αντίστοιχα (σχήμα 4.13).

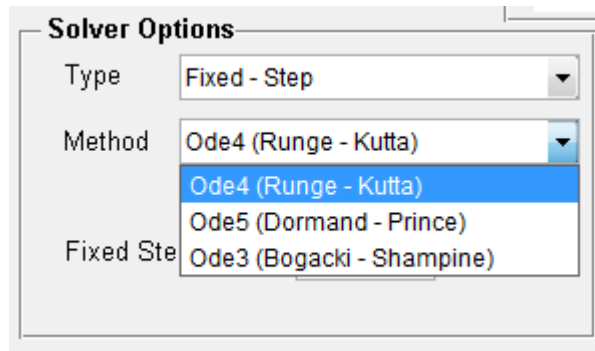


The image shows a panel titled "Simulation Time" with two input fields. The first field is labeled "Start Time" and contains the value "0". The second field is labeled "Stop Time" and contains the value "500".

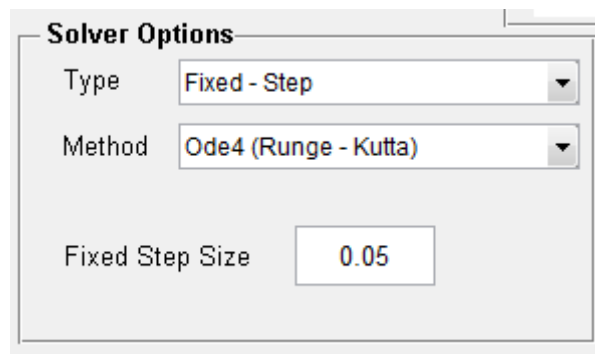
Σχήμα 4.13: Εισαγωγή χρόνων έναρξης και λήξης της προσομοίωσης στο panel Simulation Time

Μέσα από το panel Solver Options ο χρήστης εισάγει τις αριθμητικές μεθόδους επίλυσης, χρησιμοποιώντας δύο Pop-up Menus, αλλά και την τιμή των βημάτων ολοκλήρωσης, βάσει των οποίων θα πραγματοποιηθεί η προσομοίωση. Αξίζει να αναφέρουμε ότι όλα τα στοιχεία μέσα σε αυτό το panel εξαρτώνται από την επιλογή που θα κάνει ο χρήστης στο πρώτο Pop-up menu, όπου επιλέγει τον τύπο της μεθόδου επίλυσης. Πιο συγκεκριμένα, αν ο χρήστης επιλέξει επίλυση με μέθοδο σταθερού βήματος (Fixed Step), τότε το δεύτερο Pop-up menu, όπου επιλέγεται η αριθμητική μέθοδος επίλυσης, εμφανίζονται οι επιλογές

Ode4 (Runge Kutta), Ode3 (Dormand Prince) και Ode5 (Bogacki Shampine) και τέλος εμφανίζεται ένα Edit Text όπου εισάγεται το βήμα (Fixed Step Size) (σχήματα 4.14 και 4.15).

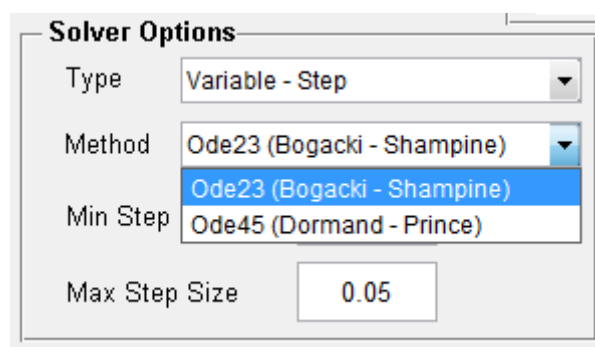


Σχήμα 4.14: Επιλογή αριθμητικής μεθόδου επίλυσης σταθερού βήματος

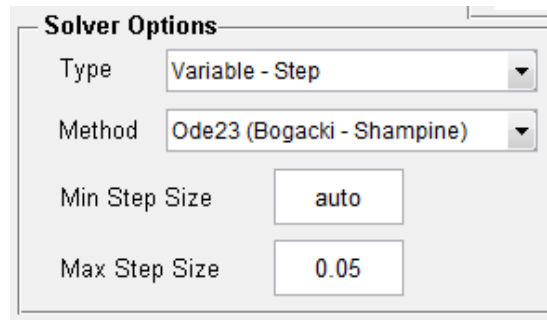


Σχήμα 4.15: Εισαγωγή σταθερού βήματος ολοκλήρωσης

Αν ο χρήστης επιλέξει επίλυση με μεταβλητό βήμα (Variable Step), τότε στο δεύτερο Pop-up menu, όπου επιλέγεται η αριθμητική μέθοδος επίλυσης, θα εμφανίζονται οι επιλογές, Ode23 (Bogacki Shampine) και Ode45 (Dormand Prince) και, τέλος, θα εμφανίζονται δύο Edit Text όπου εισάγονται το ελάχιστο και μέγιστο βήμα ολοκλήρωσης (Min Step Size και Max Step Size αντίστοιχα) (σχήματα 4.16 και 4.17).



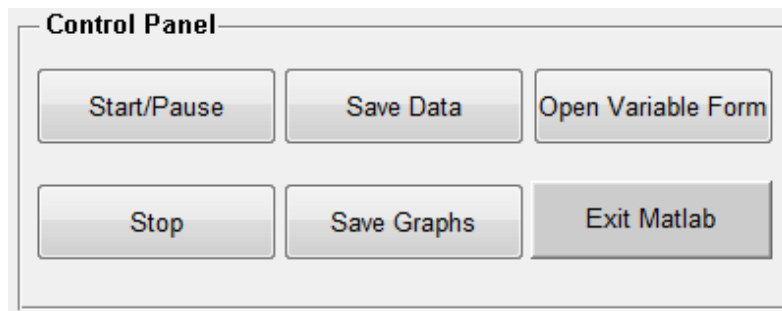
Σχήμα 4.16: Επιλογή αριθμητικής μεθόδου επίλυσης για μεταβλητό βήμα ολοκλήρωσης



Σχήμα 4.17: Εισαγωγή ελάχιστου και μέγιστου βήματος χρόνου

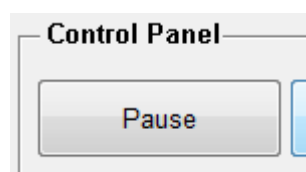
Ανάλογα λοιπόν με τον τύπο επίλυσης που επιλέγεται, εμφανίζονται και οι κατάλληλες αριθμητικές μέθοδοι επίλυσης και τα κατάλληλα πεδία εισαγωγής βημάτων ολοκλήρωσης. Με την έναρξη του προγράμματος έχουν ορισθεί οι ακόλουθες προεπιλεγμένες ρυθμίσεις: Type: Fixed Step, Method: Ode4 (Runge Kutta) και Fixed Step Size: 0.05.

Στο panel με τίτλο Control Panel, βρίσκονται τα κουμπιά με τα οποία ο χρήστης ελέγχει και δίνει εντολές στο πρόγραμμα (σχήμα 4.18). Ακολουθεί επεξήγηση των λειτουργιών κάθε κουμπιού.



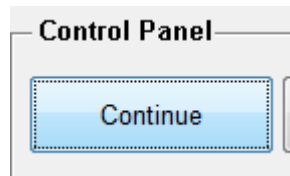
Σχήμα 4.18: Control Panel

Start/Pause: Με το πλήκτρο αυτό ο χρήστης (αφού έχει επιλέξει όλες τις επιμέρους παραμέτρους) μπορεί να ξεκινήσει, αλλά και να παγώσει την προσομοίωση. Πιο συγκεκριμένα, πριν πατηθεί, το κουμπί γράφει Start/Pause, δηλαδή τις λειτουργίες που εκτελεί. Μόλις ο χρήστης πατήσει το κουμπί ξεκινάει η προσομοίωση και το κουμπί γράφει Pause (παύση), δηλαδή τη λειτουργία που θα εκτελέσει αν πατηθεί ξανά (σχήμα 4.19).



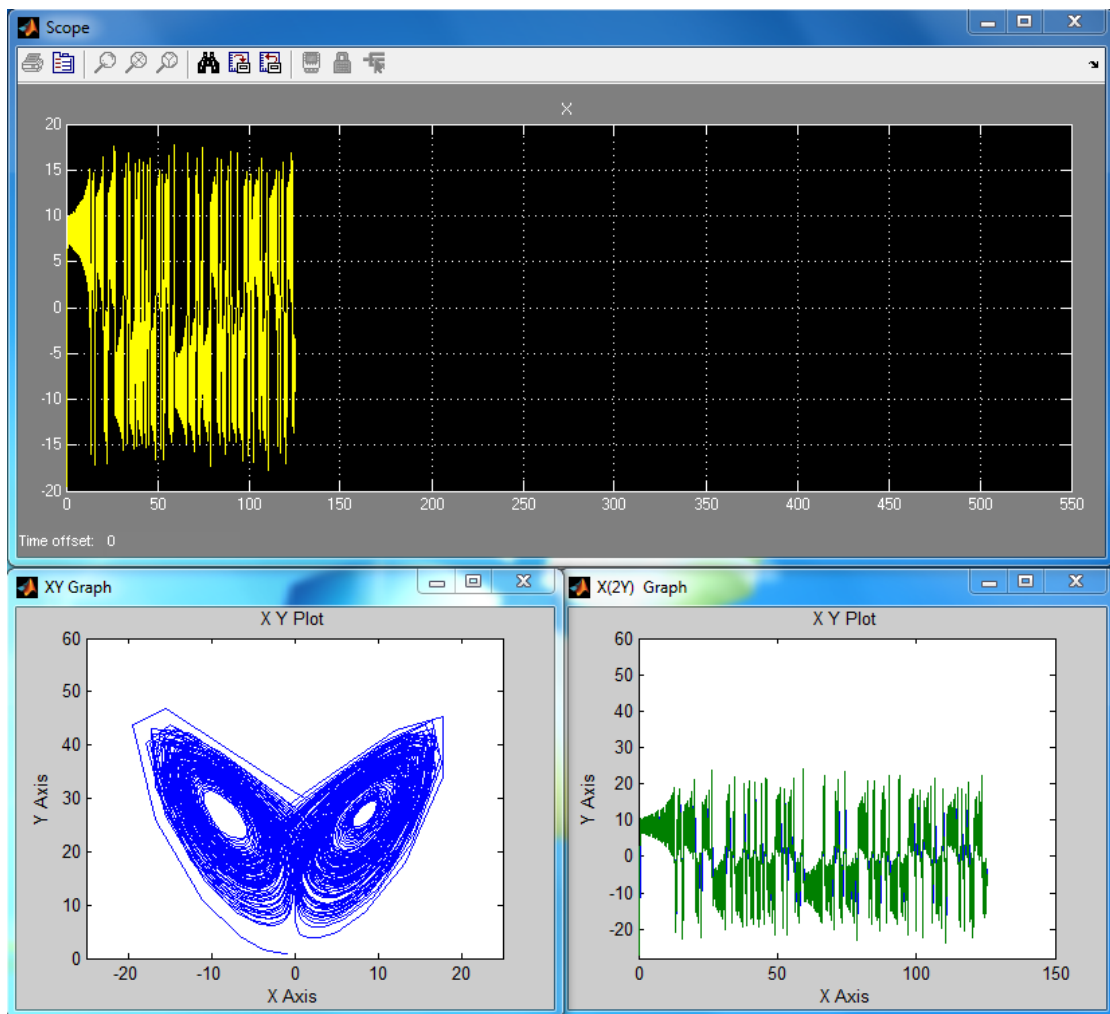
Σχήμα 4.19: Παύση προσομοίωσης

Αν ο χρήστης πατήσει πάλι το κουμπί τότε η προσομοίωση θα παγώσει αμέσως και το κουμπί θα γράφει Continue (συνέχεια), δηλαδή τη λειτουργία που θα εκτελέσει το κουμπί αν πατηθεί ξανά (σχήμα 4.20). Αν ο χρήστης πατήσει πάλι το κουμπί η προσομοίωση θα συνεχιστεί από το χρονικό σημείο στο οποίο πάγωσε κ.ο.κ.



Σχήμα 4.20: Συνέχεια προσομοίωσης

Όταν ξεκινάει η προσομοίωση εμφανίζονται τα διαγράμματα των μεταβλητών του συστήματος τα οποία ανανεώνονται σε πραγματικό χρόνο (σχήμα 4.21).



Σχήμα 4.21: Διαγράμματα που εμφανίζονται κατά τη διάρκεια της προσομοίωσης

Stop: Αυτό το πλήκτρο σταματάει την προσομοίωση και επαναφέρει το κουμπί Start/Pause στην αρχική του μορφή. Έτσι αν πατηθεί πάλι το Start/Pause η προσομοίωση θα αρχίσει και πάλι από την αρχή με τις νέες παραμέτρους που έχει εισάγει ο χρήστης.

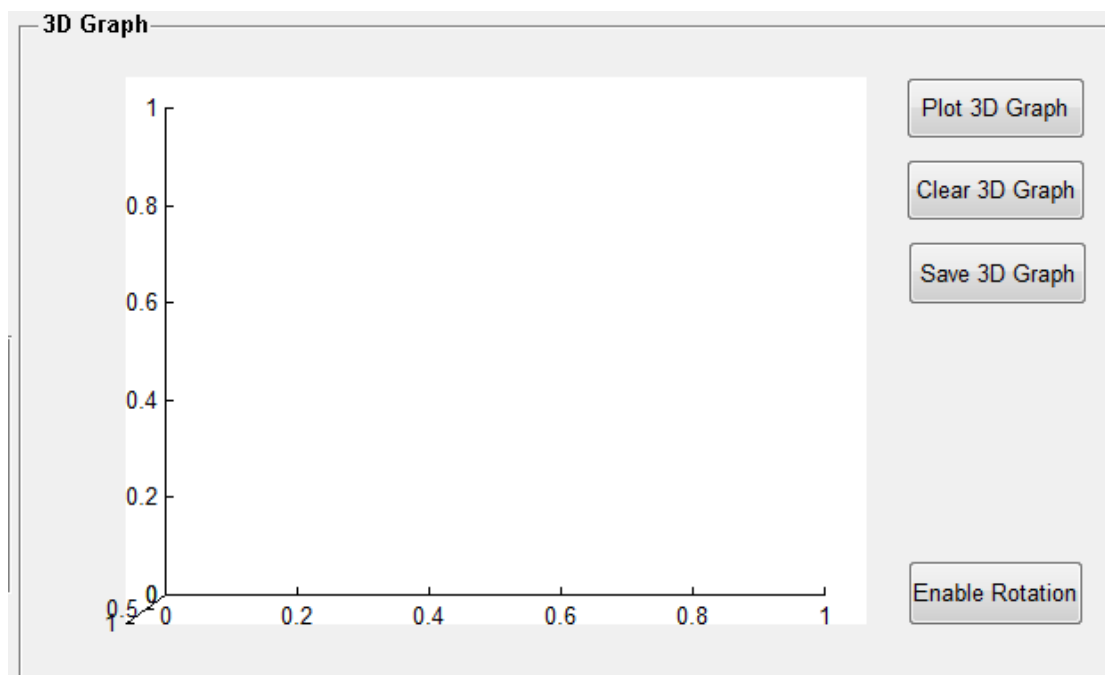
Save Data: Με αυτό το πλήκτρο δημιουργείται ένας πίνακας με τις τιμές των μεταβλητών του συστήματος για κάθε χρονική στιγμή, ο οποίος αποθηκεύεται σε μορφή αρχείου κειμένου .txt. Πιο συγκεκριμένα, στον πίνακα που δημιουργείται η πρώτη στήλη είναι ο χρόνος, η δεύτερη η μεταβλητή x , η τρίτη η μεταβλητή y και η τέταρτη η μεταβλητή z , όλες σε αντιστοιχία μεταξύ τους. Δηλαδή από τον πίνακα του αποθηκευμένου αρχείου μπορούμε να βρούμε τις τιμές των μεταβλητών x , y , z για κάθε χρονική στιγμή της προσομοίωσης. Το αρχείο αποθηκεύεται με την ονομασία Data.txt.

Save Graphs: Αυτό το κουμπί αποθηκεύει τέσσερις γραφικές παραστάσεις σε αρχεία εικόνας .jpg. Πιο συγκεκριμένα, αποθηκεύει τα γραφήματα $x(t)$, xy , xz και yz με τις ονομασίες $X(t)_Graph.jpg$, $XY_Graph.jpg$, $XZ_Graph.jpg$ και $YZ_Graph.jpg$, αντίστοιχα.

Exit Matlab: Αυτό το κουμπί κλείνει όλα τα παράθυρα και το πρόγραμμα Matlab.

Open Variable Form: Αυτό το κουμπί εμφανίζει μία νέα φόρμα, στην οποία πραγματοποιείται μελέτη του συστήματος για μεταβλητό ρ .

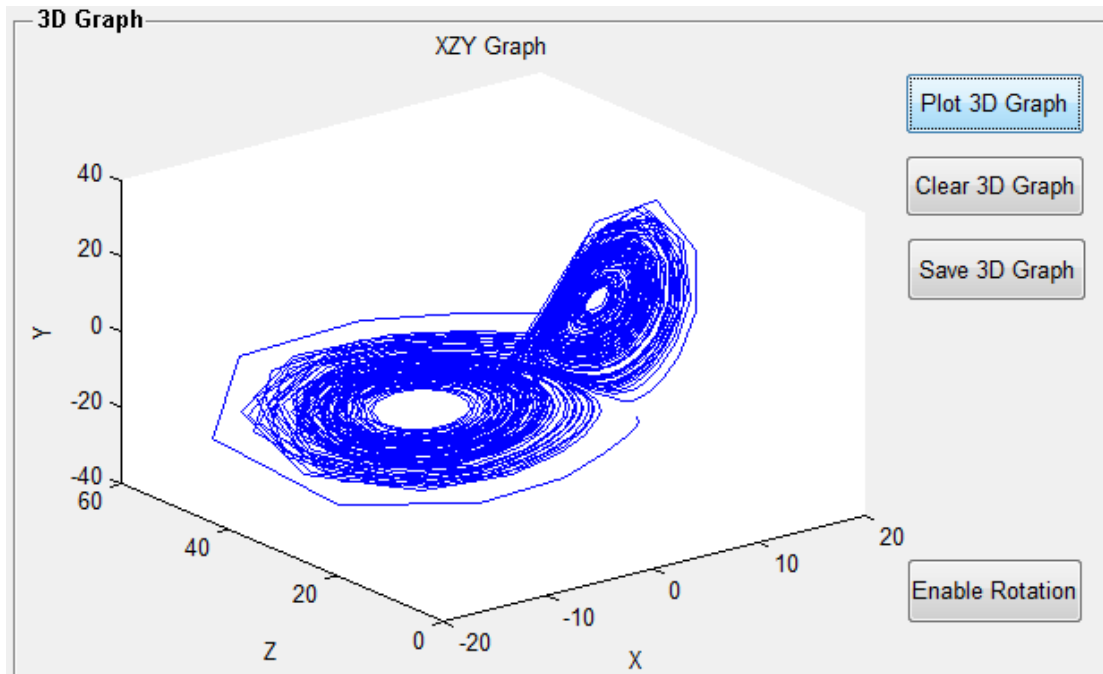
Το panel 3D Graph αποτελείται από ένα σύστημα τριών αξόνων, στο οποίο εμφανίζεται το 3D γράφημα xzy , και από τέσσερα κουμπιά, για επιμέρους λειτουργίες (σχήμα 4.22).



Σχήμα 4.22: 3D Graph panel

Ακολουθεί επεξήγηση των λειτουργιών κάθε κουμπιού.

Plot 3D Graph: Με αυτό το πλήκτρο εμφανίζεται το τρισδιάστατο γράφημα xyz (σχήμα 4.23).

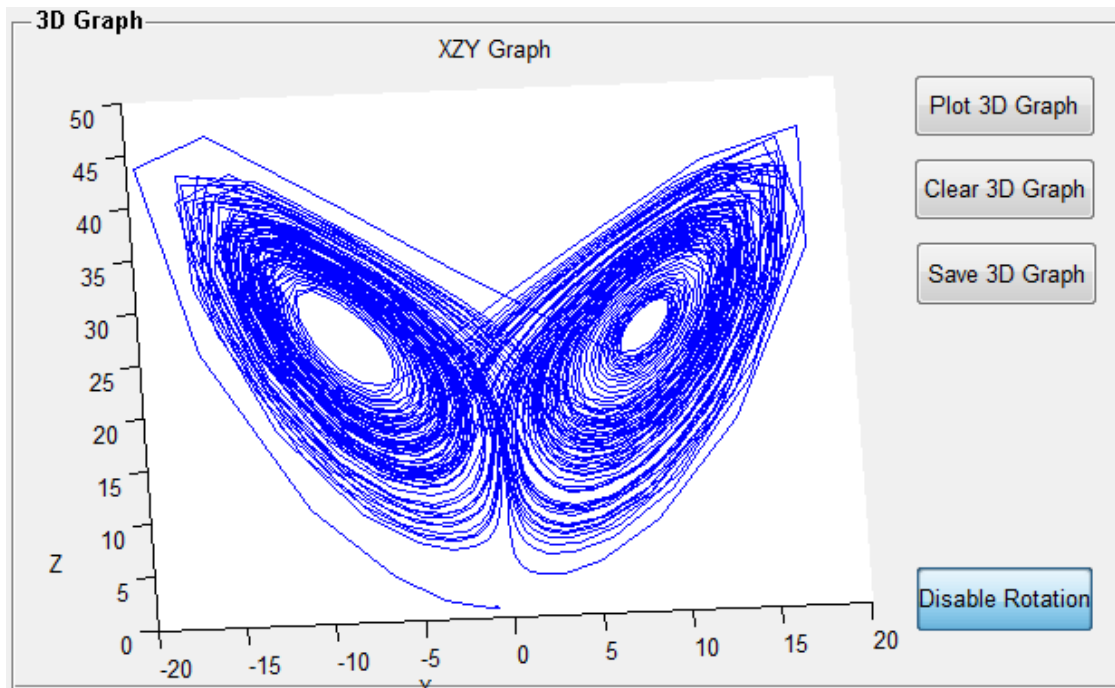


Σχήμα 4.23: Εμφάνιση τρισδιάστατου γραφήματος

Clear 3D Graph: Αυτό το πλήκτρο σβήνει τα υπάρχοντα δεδομένα από το τρισδιάστατο γράφημα (σύστημα τριών αξόνων) και το επαναφέρει στην αρχική του μορφή. Ο χρήστης στη συνέχεια μπορεί να εμφανίσει άλλο διάγραμμα.

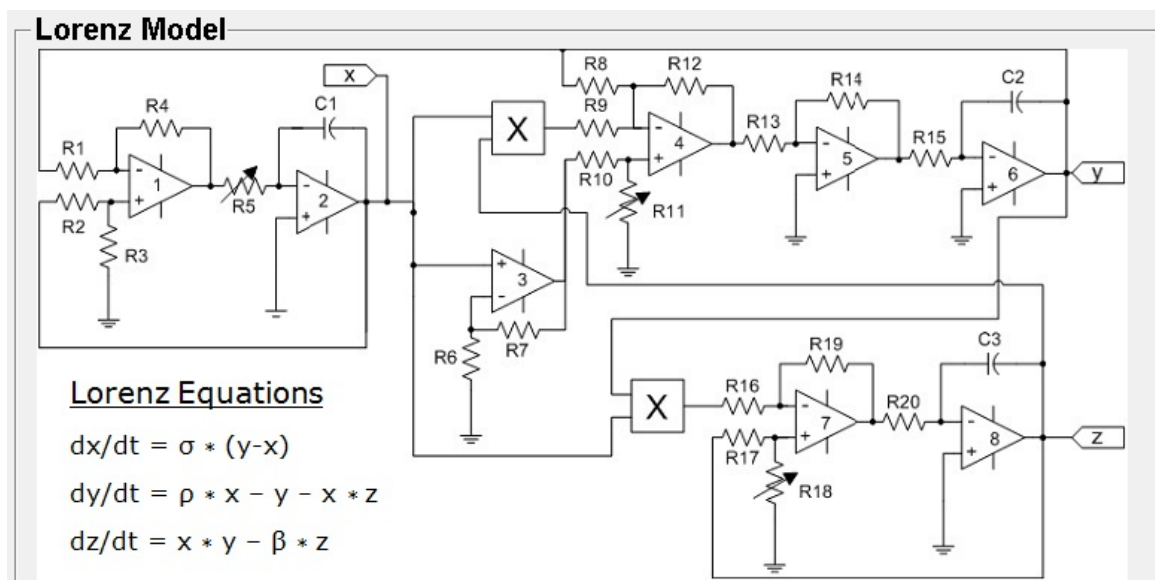
Save 3D Graph: Αυτό το πλήκτρο αποθηκεύει το τρισδιάστατο γράφημα σε αρχείο εικόνας .jpg με την ονομασία XYZ_Graph.jpg. Η αποθήκευση γίνεται στο φάκελο εργασίας του Matlab.

Enable Rotation: Αυτό το πλήκτρο είναι ένα Toggle button. Όταν είναι πατημένο επιτρέπει στο χρήστη να περιστρέψει το γράφημα, ώστε να βρει την επιθυμητή οπτική γωνία από την οποία θέλει να μελετήσει το διάγραμμα. Έχοντας πατημένο το αριστερό πλήκτρο του ποντικιού μπορεί να μετακινήσει και να αλλάξει τη θέση των αξόνων στο χώρο. Όταν ο χρήστης το πατήσει τότε γράφει Disable Rotation που σημαίνει πως αν πατηθεί πάλι δεν θα επιτρέπει πλέον στο χρήστη να το περιστρέψει και θα «κλειδώσει» την οπτική γωνία που επέλεξε ο χρήστης (σχήμα 4.24). Αν το πατήσει για μια ακόμη φορά τότε θα μπορεί και πάλι να το περιστρέψει κ.ο.κ.



Σχήμα 4.24: Περιστροφή τρισδιάστατου γραφήματος

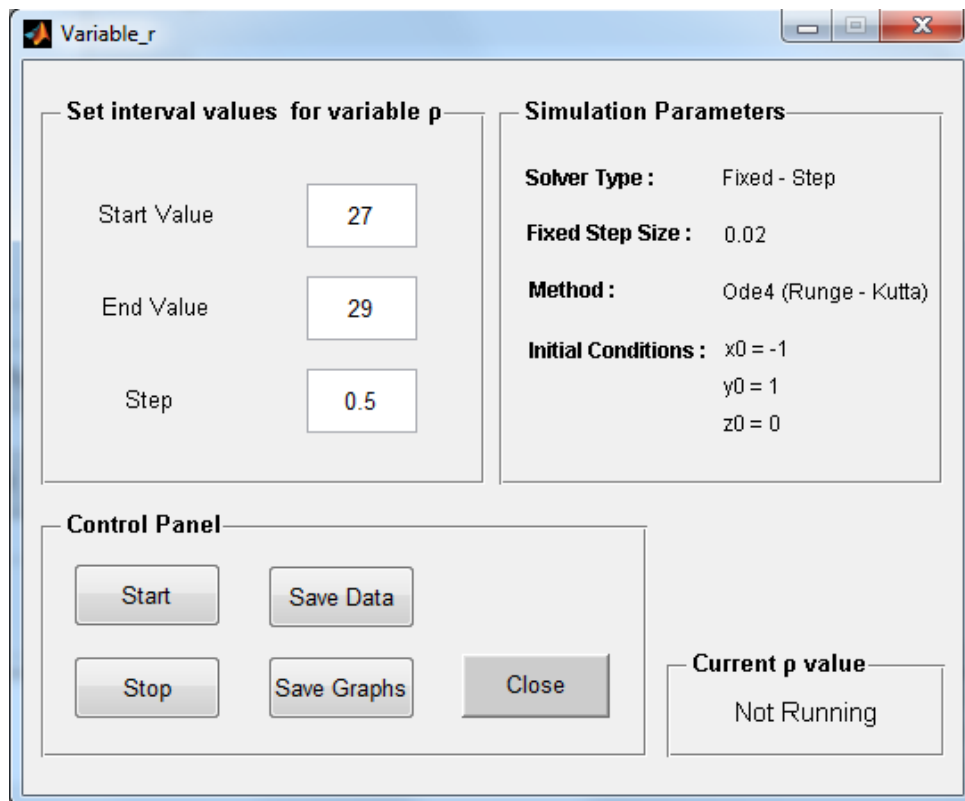
Το τελευταίο panel που υπάρχει στο βασικό περιβάλλον του προγράμματος είναι το Lorenz Model και περιέχει απλά μια εικόνα του ισοδύναμου ηλεκτρονικού κυκλώματος του μοντέλου Lorenz, με σκοπό να κάνει το περιβάλλον του προγράμματος πιο καλαίσθητο (σχήμα 4.25).



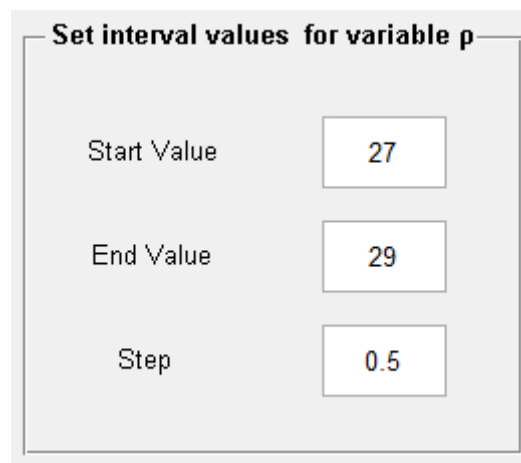
Σχήμα 4.25: Ισοδύναμο ηλεκτρονικό κύκλωμα συστήματος Lorenz

4.4.3 Περιβάλλον εφαρμογής – Μεταβλητό ρ

Πατώντας το κουμπί Open Variable Form, από το βασικό περιβάλλον του προγράμματος, εμφανίζεται η φόρμα στην οποία εκτελείται η λειτουργία μεταβλητού ρ (σχήμα 4.26). Ο χρήστης, στο panel Set interval values for ρ , εισάγει την αρχική τιμή, την τελική και το βήμα μεταβολής της παραμέτρου ρ , το οποίο θα αλλάζει για κάθε 100 δείγματα χρόνου (σχήμα 4.27).

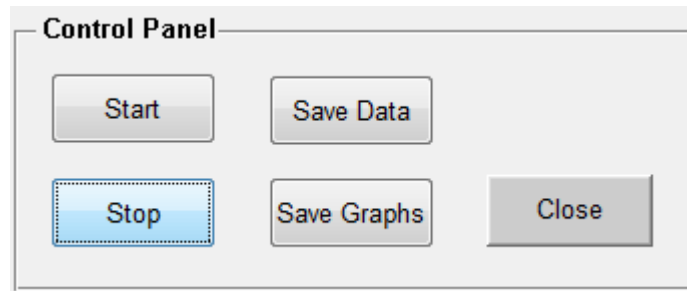


Σχήμα 4.26: Φόρμα λειτουργίας μεταβλητού ρ



Σχήμα 4.27: Εισαγωγή αρχικής τιμής, τελικής και βήματος για την παράμετρο ρ

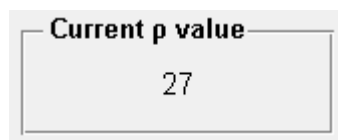
Στο panel με την ονομασία Control Panel βρίσκονται τα κουμπιά με τα οποία ο χρήστης ελέγχει την προσομοίωση (σχήμα 4.28).



Σχήμα 4.28: Control Panel

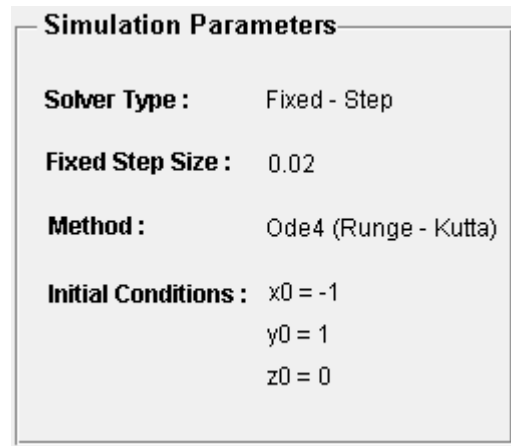
Πιο συγκεκριμένα το κουμπί Start ξεκινάει την προσομοίωση και το κουμπί Stop την σταματάει. Το κουμπί Save Data δημιουργεί έναν πίνακα με τις τιμές των μεταβλητών του συστήματος για κάθε χρονική στιγμή και τον αποθηκεύει με τη μορφή αρχείου κειμένου .txt. Στον πίνακα αυτό η πρώτη στήλη είναι το ρ , η δεύτερη ο χρόνος, η τρίτη η μεταβλητή x , η τέταρτη η μεταβλητή y και η πέμπτη η μεταβλητή z , όλες σε αντιστοιχία μεταξύ τους. Δηλαδή από τον πίνακα του αποθηκευμένου αρχείου μπορούμε να βρούμε τις τιμές των μεταβλητών ρ , x , y και z για κάθε χρονική στιγμή της προσομοίωσης. Το αρχείο αποθηκεύεται με την ονομασία Data.txt. Το Save Graphs αποθηκεύει τέσσερις γραφικές παραστάσεις σε αρχεία εικόνας .jpg. Πιο συγκεκριμένα, αποθηκεύει τα γραφήματα $x(t)$, xy , xz και yz με τις ονομασίες Variable_r_X(t)_Graph.jpg, Variable_r_XY_Graph.jpg, Variable_r_XZ_Graph.jpg και Variable_r_YZ_Graph.jpg, αντίστοιχα. Τέλος, το κουμπί Close κλείνει αυτή τη φόρμα και μας επιστρέφει στη βασική φόρμα του προγράμματος.

Στο panel Current ρ Value αναγράφεται κάθε φορά η τιμή του ρ για την οποία τρέχει τη συγκεκριμένη χρονική στιγμή η προσομοίωση (σχήμα 4.29).



Σχήμα 4.29: Εμφάνιση τρέχουσας τιμής του ρ

Τέλος στο panel Simulation Parameters αναγράφονται οι παράμετροι βάσει των οποίων πραγματοποιείται η προσομοίωση και τις οποίες δεν μπορεί να μεταβάλλει ο χρήστης σε αυτήν τη λειτουργία του προγράμματος (σχήμα 4.30).



Σχήμα 4.30: Εμφάνιση παραμέτρων στο panel Simulation Parameters

Κεφάλαιο 5

Αποτελέσματα Προσομοίωσης

5.1 Εισαγωγή

Στο κεφάλαιο που ακολουθεί θα παραθέσουμε ορισμένα αριθμητικά αποτελέσματα που ελήφθησαν με τη χρήση της εφαρμογής που σχεδιάστηκε και παρουσιάστηκε σε προηγούμενα κεφάλαια. Στον πίνακα I παρουσιάζονται συγκεντρωτικά οι παράμετροι των προσομοιώσεων που θα ακολουθήσουν.

Πίνακας I: Συγκεντρωτικός πίνακας αριθμητικών προσομοιώσεων

α/α	σ	β	ρ	Start time	Stop time	Solver type	method	step
1	10	8/3	28	0	200	fixed	Ode3	0.05
2	10	8/3	28	0	200	fixed	Ode4	0.05
3	10	8/3	28	0	200	fixed	Ode5	0.05
4	10	8/3	28	0	200	variable	Ode 23	0.05
5	10	8/3	28	0	200	variable	Ode 45	0.05
6	10	8/3	22-28 με βήμα 2.0	0	100	fixed	Ode4	0.02
7	10	8/3	22	0	100	fixed	Ode4	0.02
8	10	8/3	24	0	100	fixed	Ode4	0.02
9	10	8/3	26	0	100	fixed	Ode4	0.02
10	10	8/3	28	0	100	fixed	Ode4	0.02

5.2 Προσομοιώσεις σταθερού βήματος με τη χρήση διαφορικών αριθμητικών μεθόδων ολοκλήρωσης

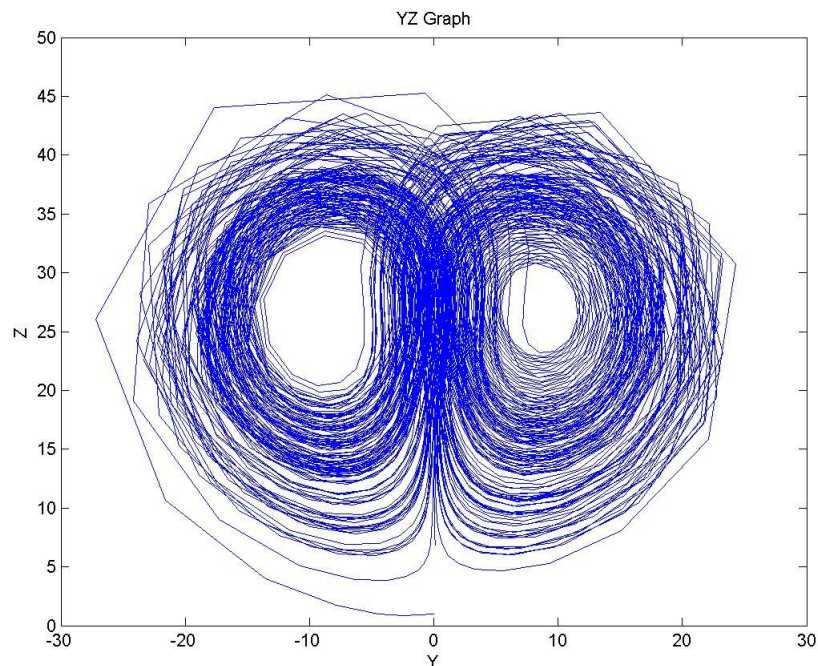
Αρχικά πραγματοποιούμε προσομοιώσεις κρατώντας σταθερές τις παραμέτρους του συστήματος και μεταβάλλοντας κάθε φορά τη σταθερού βήματος μέθοδο ολοκλήρωσης. Οι παράμετροι οι οποίες διατηρούνται σταθερές έχουν τις εξής τιμές:

$$\sigma=10, \beta=8/3, \rho=28$$

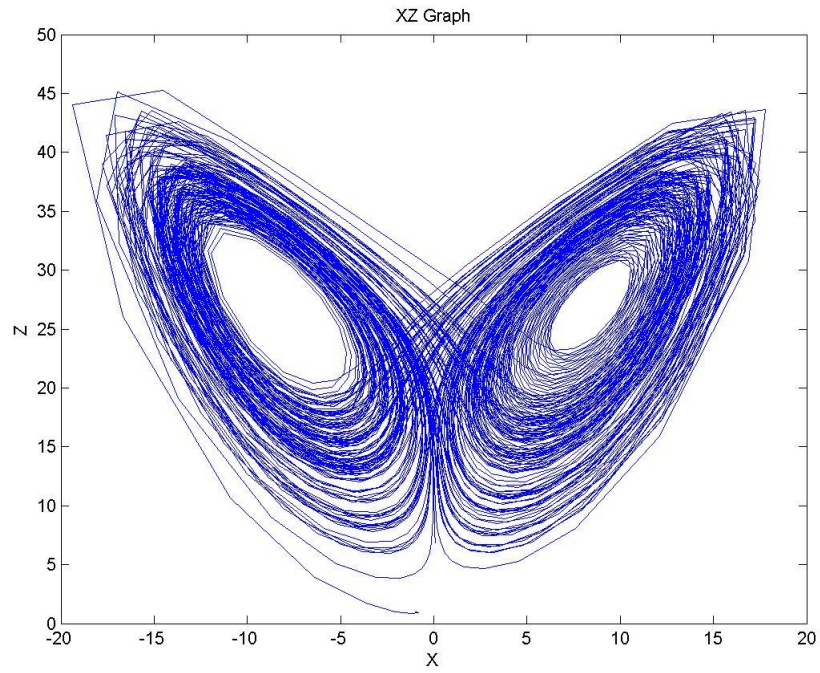
Start Time \rightarrow 0, Stop Time \rightarrow 200

Solver Type \rightarrow fixed, method \rightarrow OdeX, Step \rightarrow 0.05

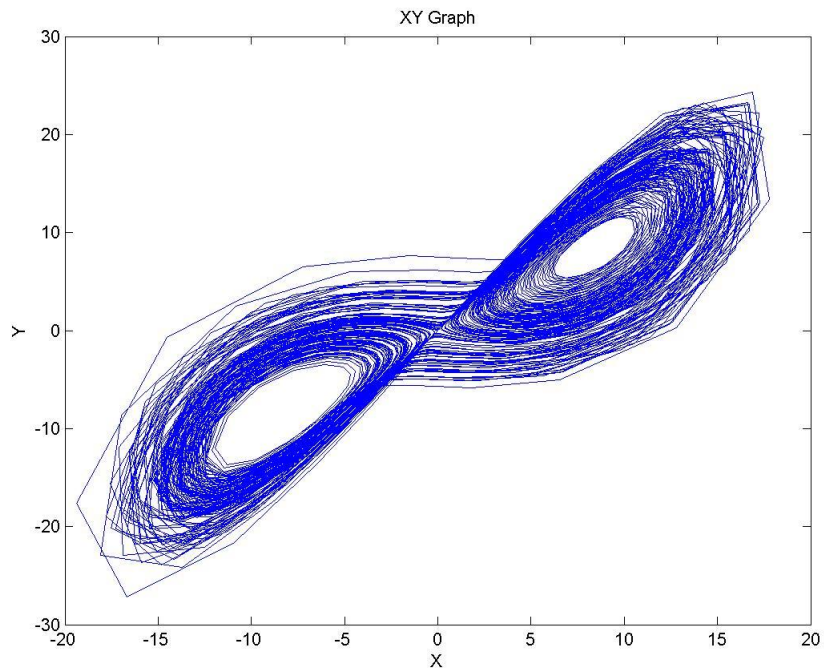
Οι μέθοδοι που χρησιμοποιούμε είναι οι Ode3 (Bogacki-Shampine), Ode4 (Runge-Kutta) και Ode5 (Dormand-Prince).



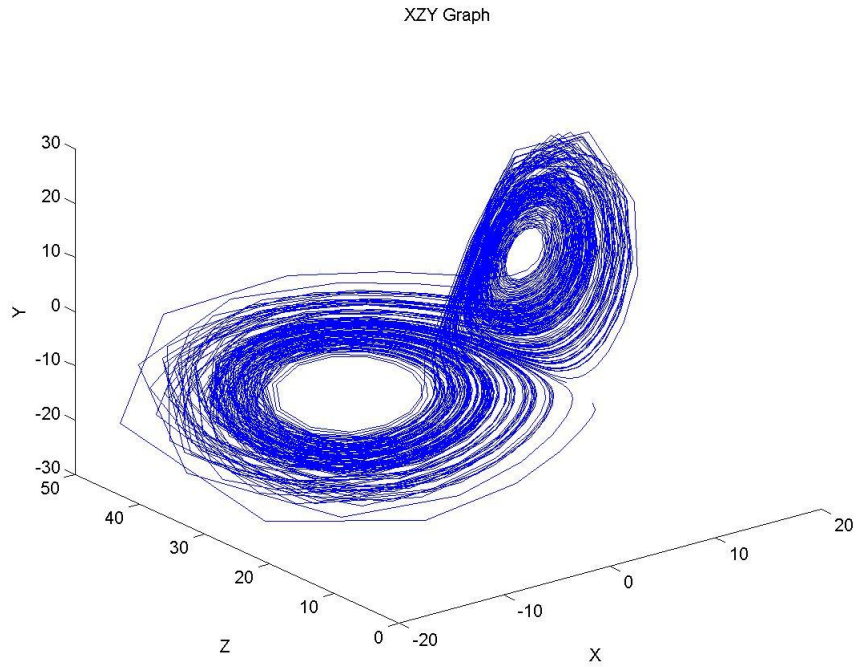
Σχήμα 5.1: Διάγραμμα yz με τη χρήση της Ode3 (Bogacki-Shampine)



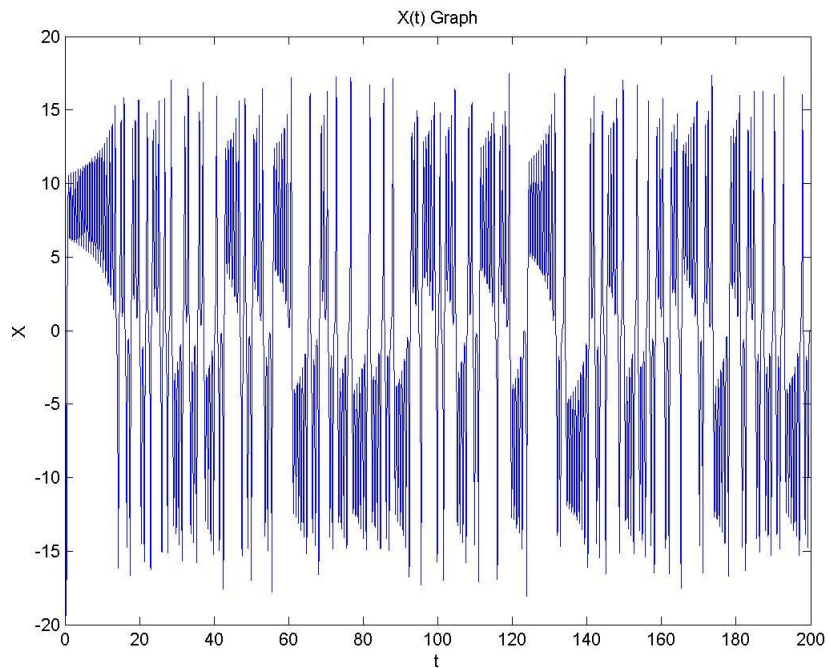
Σχήμα 5.2: Διάγραμμα xz με τη χρήση της Ode3 (Bogacki-Shampine)



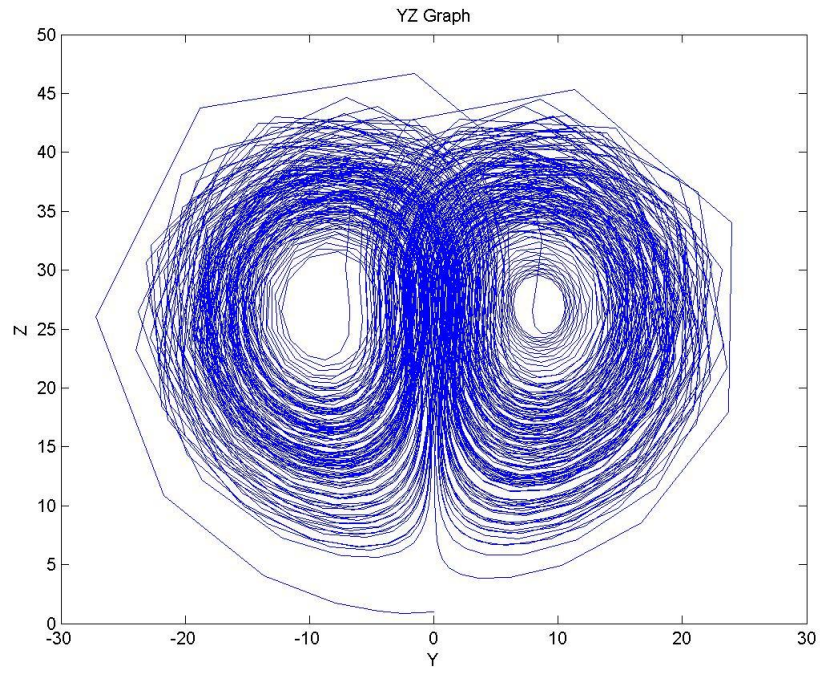
Σχήμα 5.3: Διάγραμμα xy με τη χρήση της Ode3 (Bogacki-Shampine)



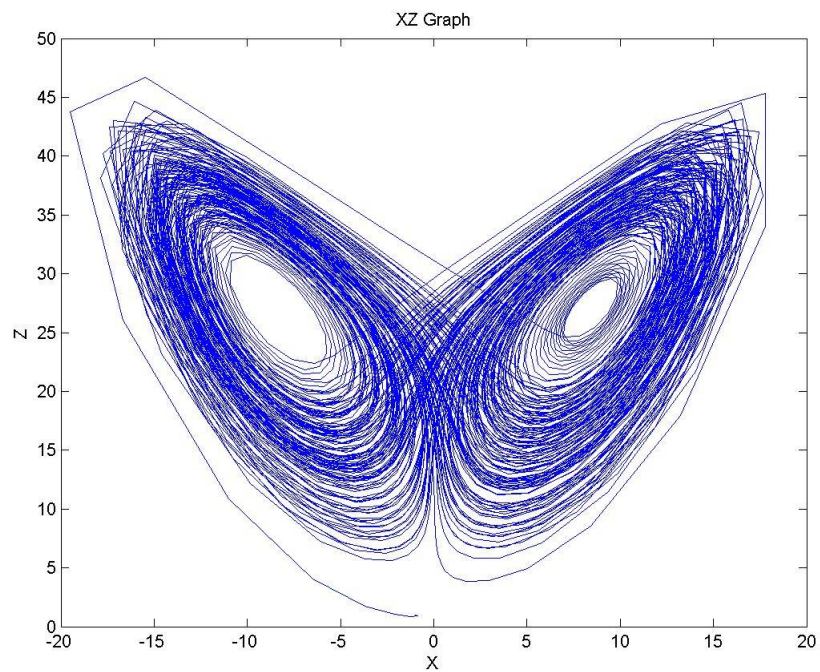
Σχήμα 5.4: Διάγραμμα 3D - xyz με τη χρήση της Ode3 (Bogacki-Shampine)



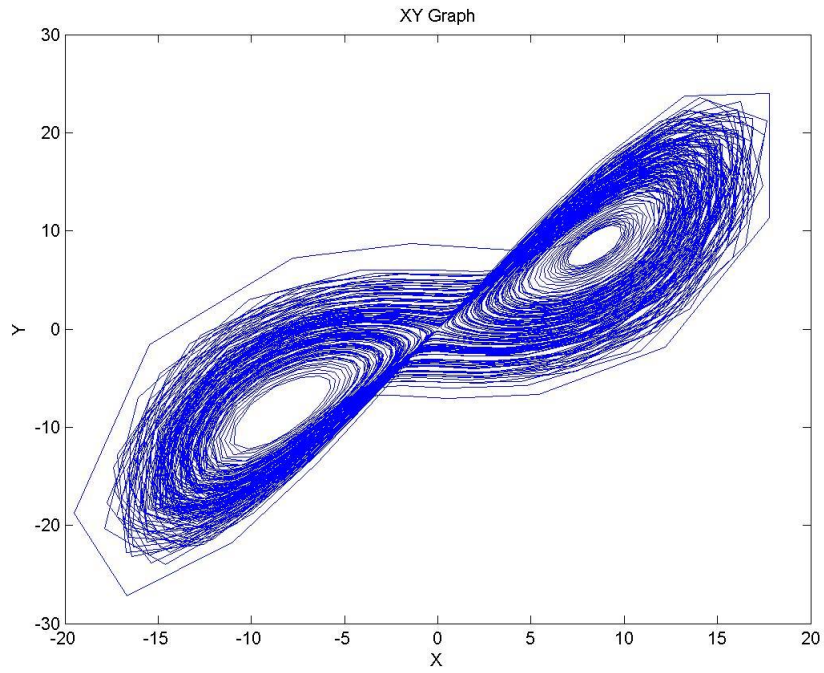
Σχήμα 5.5: Κυματομορφή $x(t)$ με τη χρήση της Ode3 (Bogacki-Shampine)



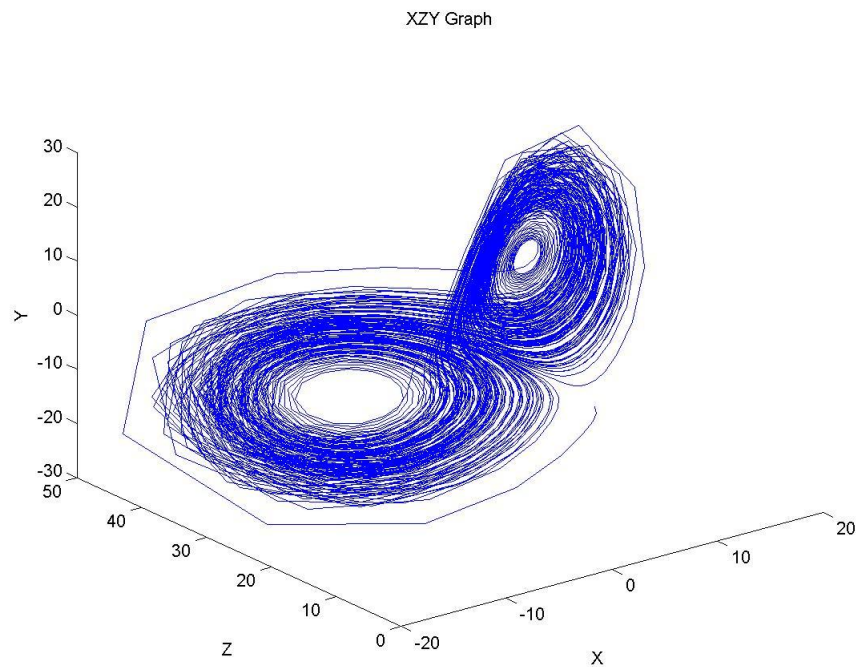
Σχήμα 5.6: Διάγραμμα yz με τη χρήση της Ode4 (Runge-Kutta)



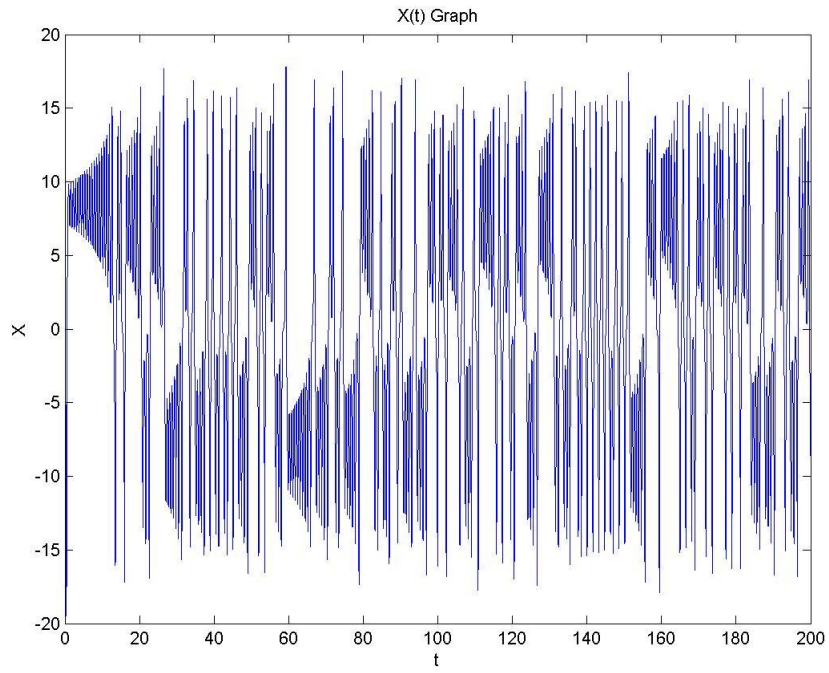
Σχήμα 5.7: Διάγραμμα xz με τη χρήση της Ode4 (Runge-Kutta)



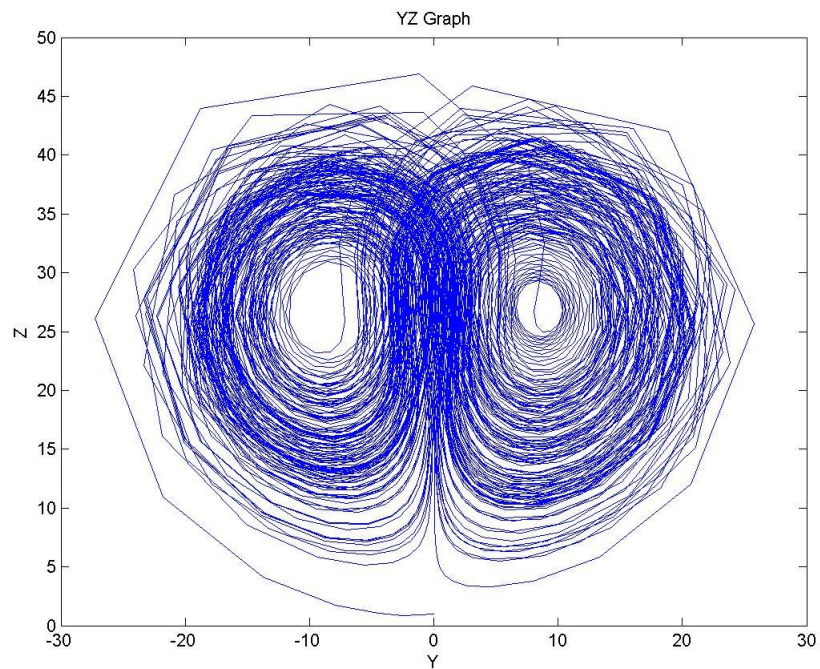
Σχήμα 5.8: Διάγραμμα xy με τη χρήση της Ode4 (Runge-Kutta)



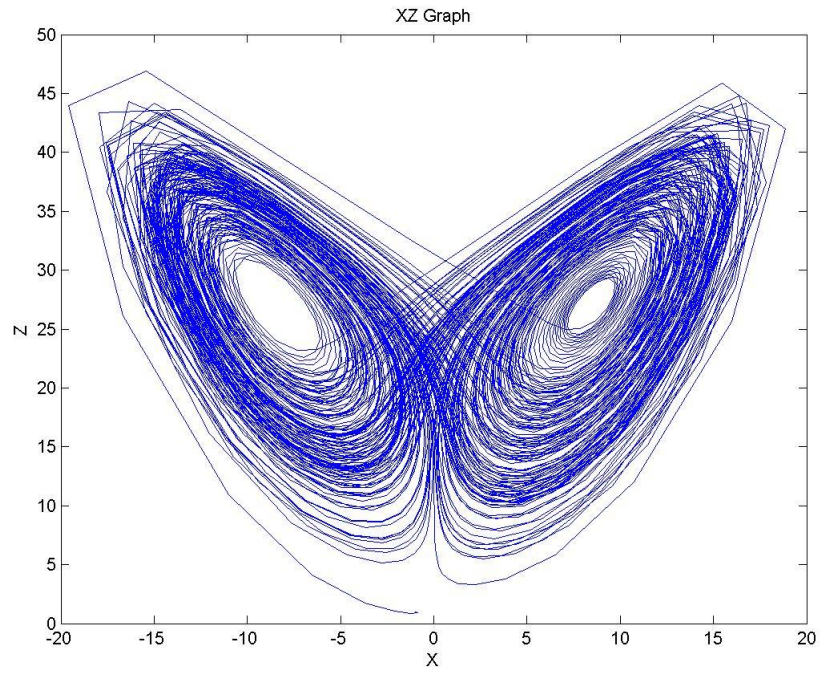
Σχήμα 5.9: Διάγραμμα 3D – xyz με τη χρήση της Ode4 (Runge-Kutta)



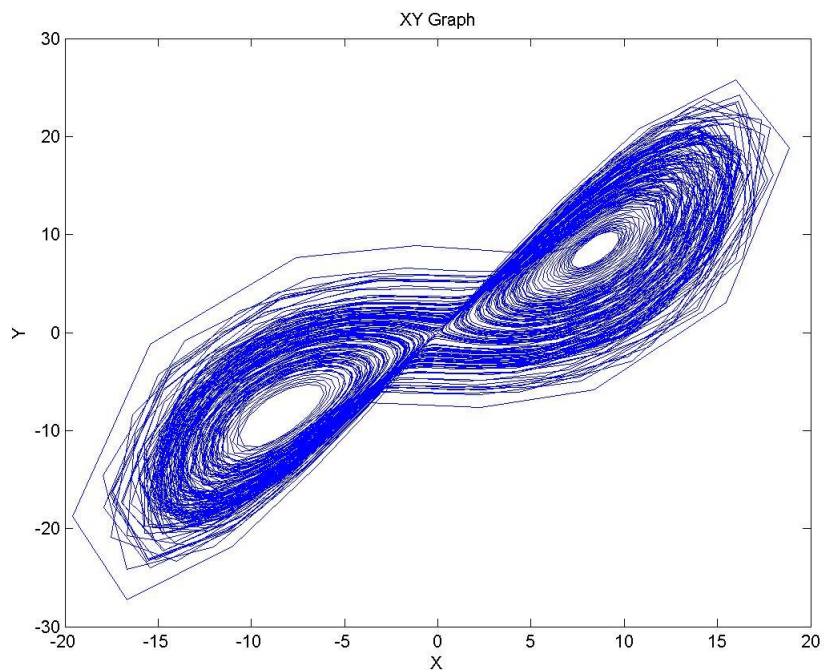
Σχήμα 5.10: Κυματομορφή $x(t)$ με τη χρήση της Ode4 (Runge-Kutta)



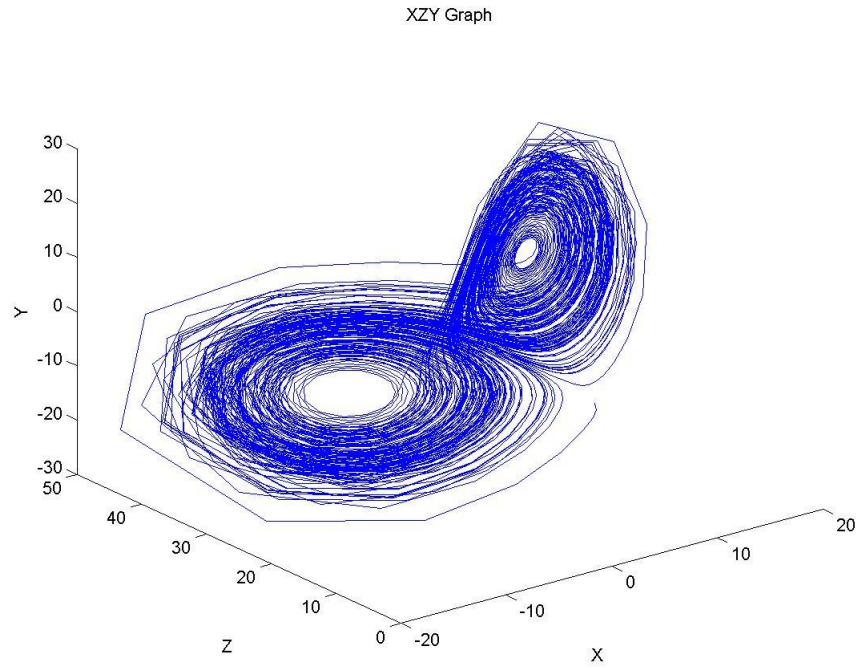
Σχήμα 5.11: Διάγραμμα yz με τη χρήση της Ode5 (Dormand-Prince)



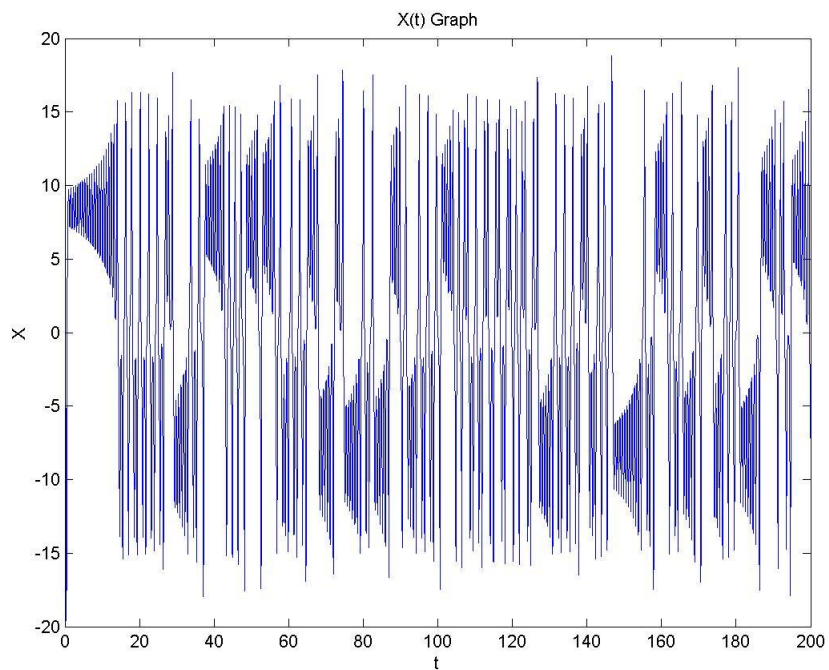
Σχήμα 5.12: Διάγραμμα xz με τη χρήση της Ode5 (Dormand-Prince)



Σχήμα 5.13: Διάγραμμα xy με τη χρήση της Ode5 (Dormand-Prince)



Σχήμα 5.14: Διάγραμμα 3D – xyz με τη χρήση της Ode5 (Dormand-Prince)



Σχήμα 5.15: Κυματομορφή $x(t)$ με τη χρήση της Ode5 (Dormand-Prince)

5.3 Προσομοιώσεις μεταβλητού βήματος με τη χρήση διαφορικών αριθμητικών μεθόδων ολοκλήρωσης

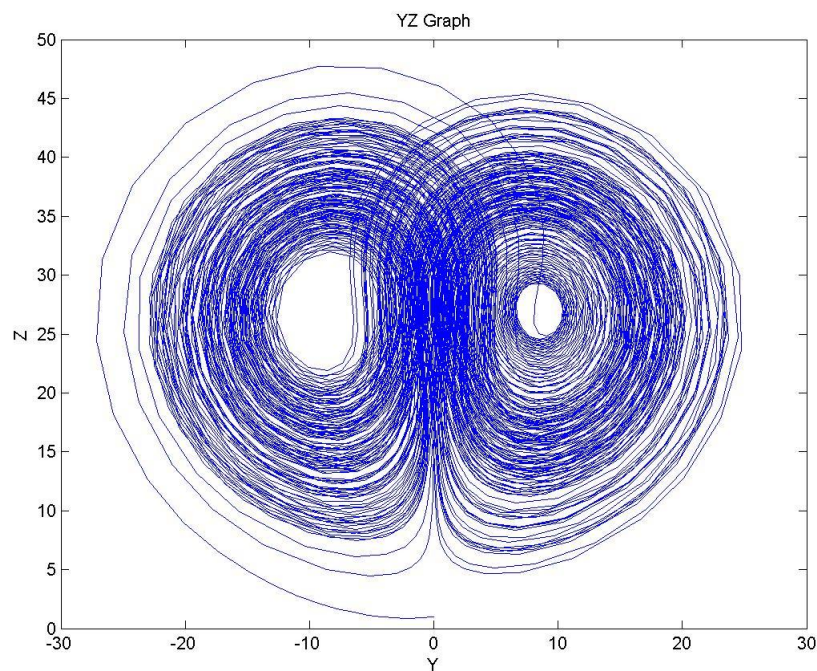
Στη συνέχεια, πραγματοποιούμε προσομοιώσεις κρατώντας σταθερές τις παραμέτρους του συστήματος και μεταβάλλοντας κάθε φορά τη μεταβλητού βήματος μέθοδο ολοκλήρωσης. Οι παράμετροι οι οποίες διατηρούνται σταθερές έχουν τις εξής τιμές:

$$\sigma=10, \beta=8/3, \rho=28$$

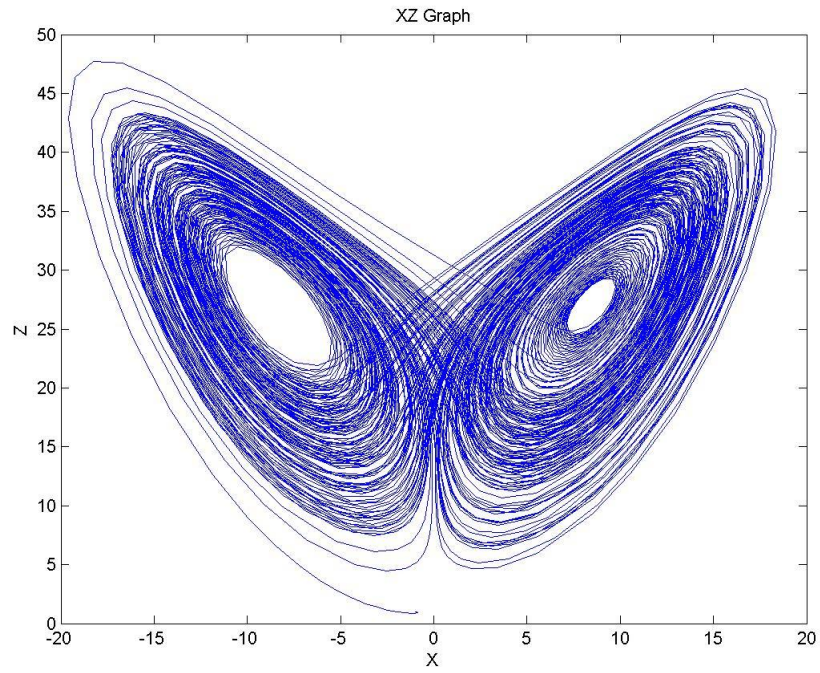
Start Time \rightarrow 0, Stop Time \rightarrow 200

Solver Type \rightarrow variable, method \rightarrow OdeXX, Step \rightarrow 0.05

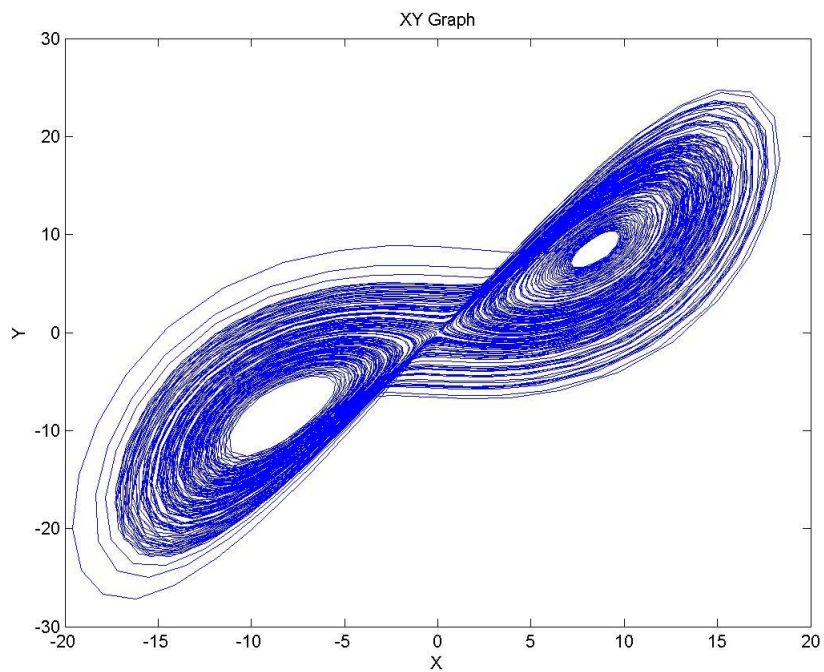
Οι μέθοδοι που χρησιμοποιούμε είναι οι Ode23 (Bogacki-Shampine), Ode45 (Dormand-Prince).



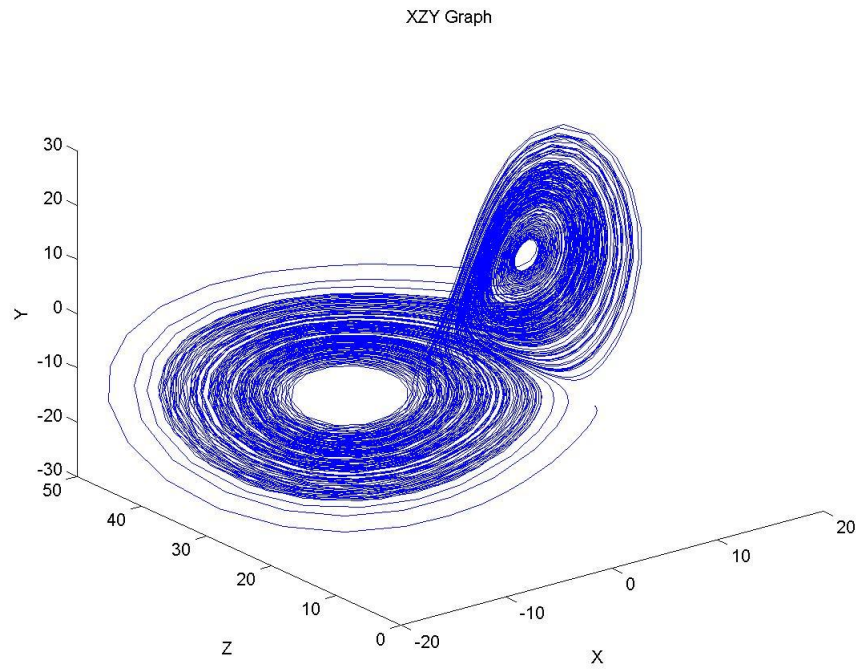
Σχήμα 5.16: Διάγραμμα yz με τη χρήση της Ode23 (Bogacki-Shampine)



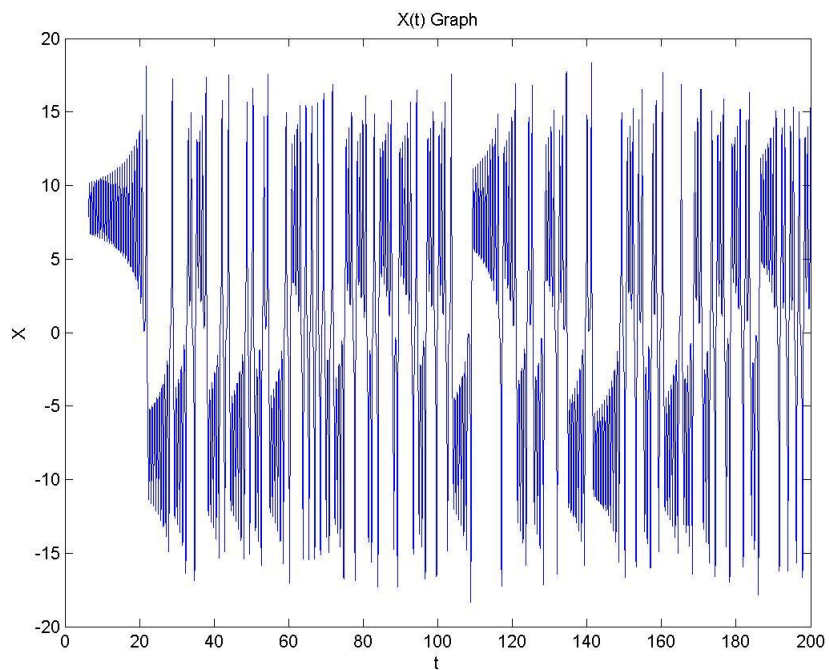
Σχήμα 5.17: Διάγραμμα xz με τη χρήση της Ode23 (Bogacki-Shampine)



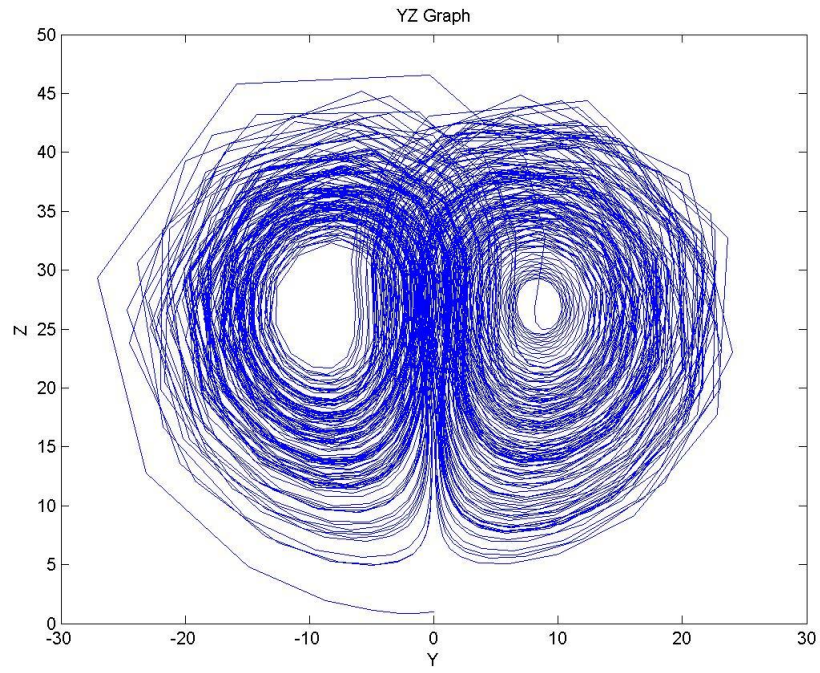
Σχήμα 5.18: Διάγραμμα xy με τη χρήση της Ode23 (Bogacki-Shampine)



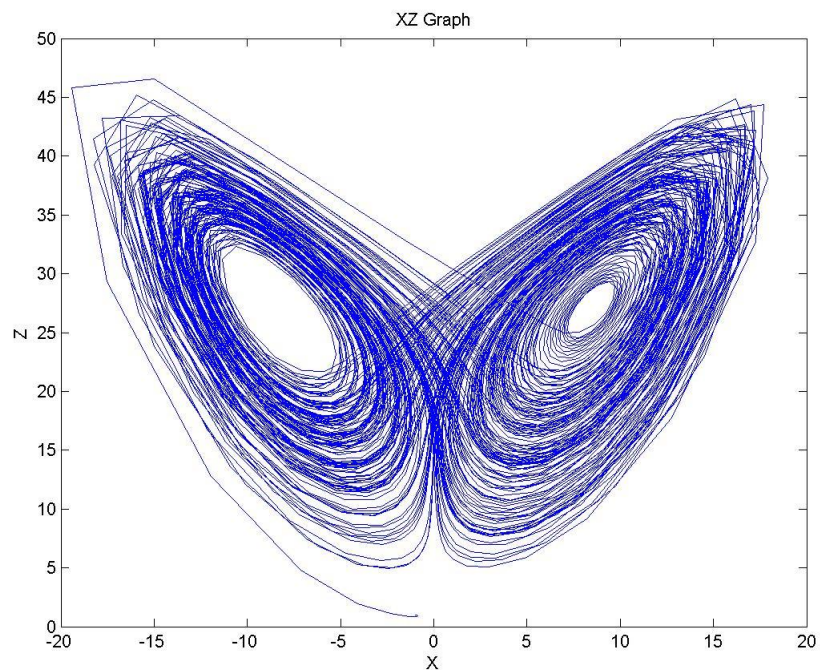
Σχήμα 5.19: Διάγραμμα 3D - xzy με τη χρήση της Ode23 (Bogacki-Shampine)



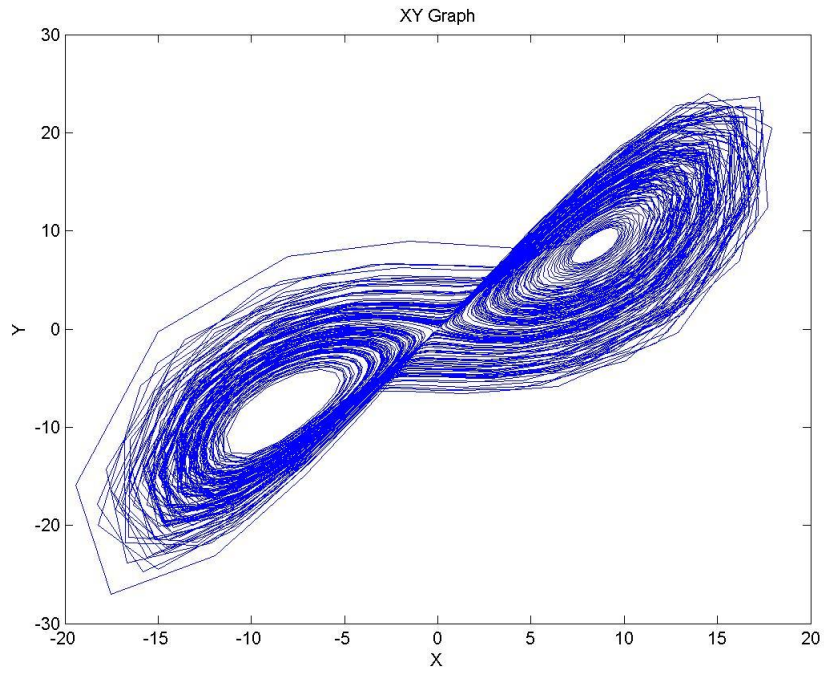
Σχήμα 5.20: Κυματομορφή $x(t)$ με τη χρήση της Ode23 (Bogacki-Shampine)



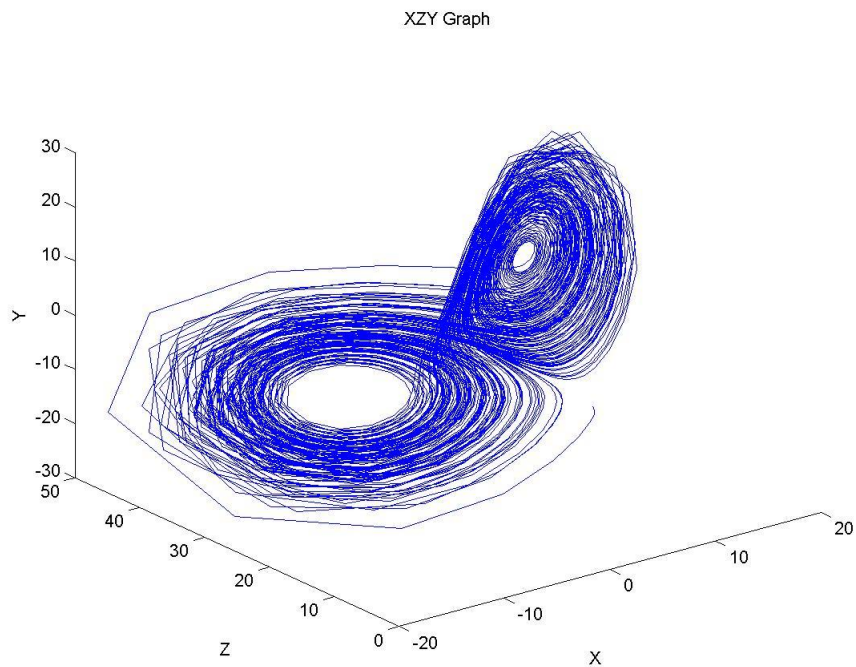
Σχήμα 5.21: Διάγραμμα yz με τη χρήση της Ode45 (Dormand-Prince)



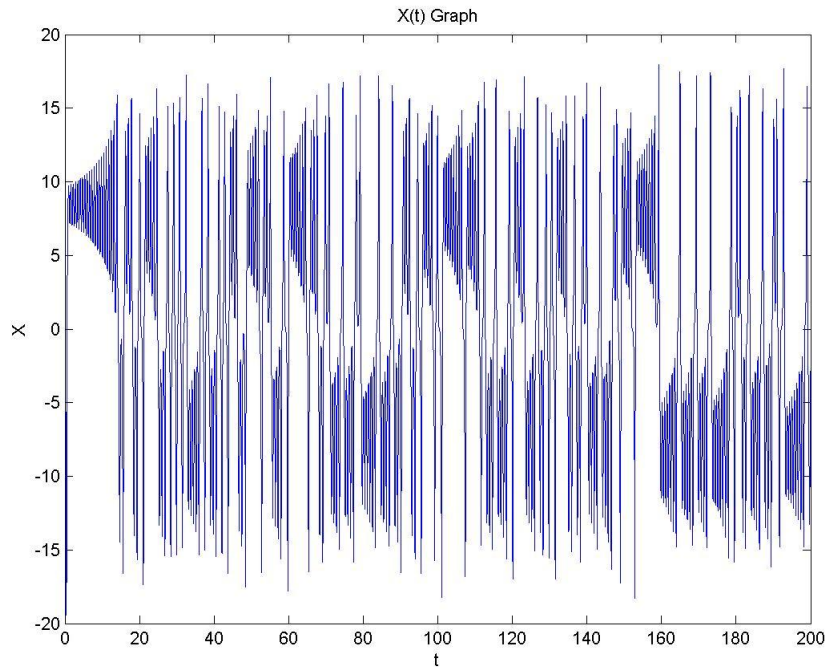
Σχήμα 5.22: Διάγραμμα xz με τη χρήση της Ode45 (Dormand-Prince)



Σχήμα 5.23: Διάγραμμα xy με τη χρήση της Ode45 (Dormand-Prince)



Σχήμα 5.24: Διάγραμμα 3D - xzy με τη χρήση της Ode45 (Dormand-Prince)



Σχήμα 5.25: Κυματομορφή $x(t)$ με τη χρήση της Ode45 (Dormand-Prince)

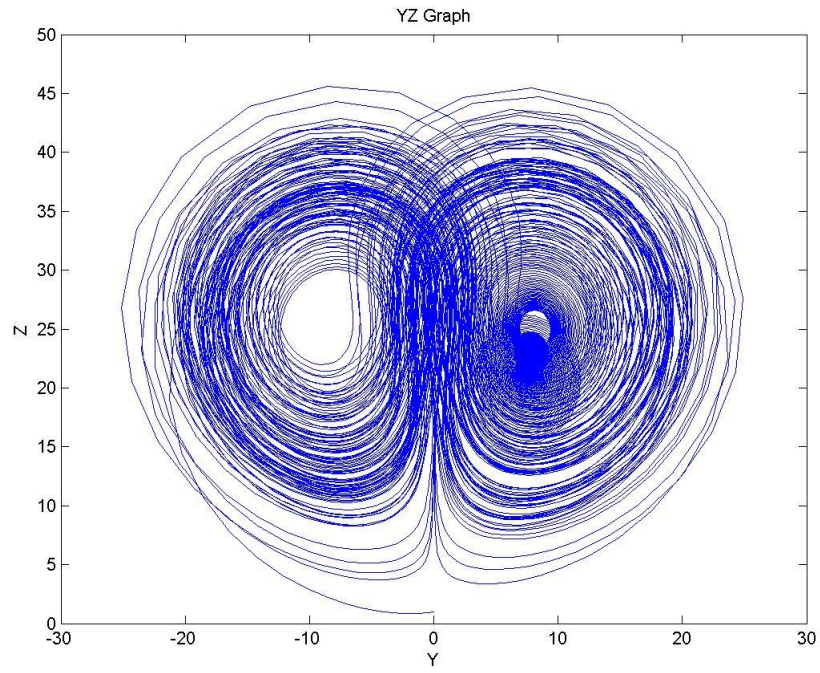
5.4 Προσομοιώσεις για διάφορες τιμές της παραμέτρου ρ

Στη συνέχεια, πραγματοποιούμε προσομοιώσεις κρατώντας σταθερές τις παραμέτρους του συστήματος και μεταβάλλοντας κάθε φορά την παράμετρο ρ του συστήματος των εξισώσεων Lorenz. Οι παράμετροι οι οποίες διατηρούνται σταθερές έχουν τις εξής τιμές:

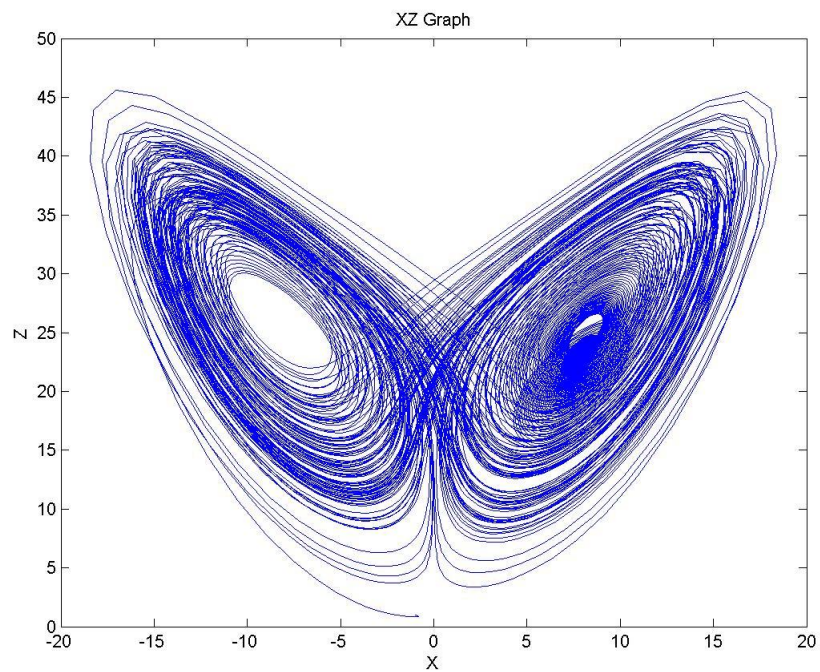
$\sigma=10$, $\beta=8/3$, ρ = μεταβλητό

Start Time $\rightarrow 0$, Stop Time $\rightarrow 100$

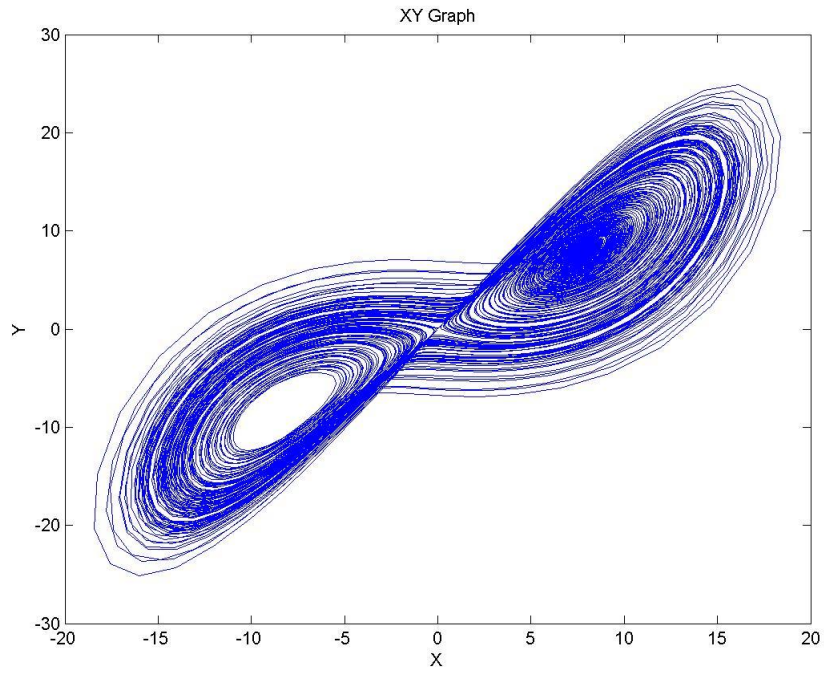
Solver Type \rightarrow fixed, method \rightarrow Ode4 (Runge-Kutta), Step $\rightarrow 0.02$



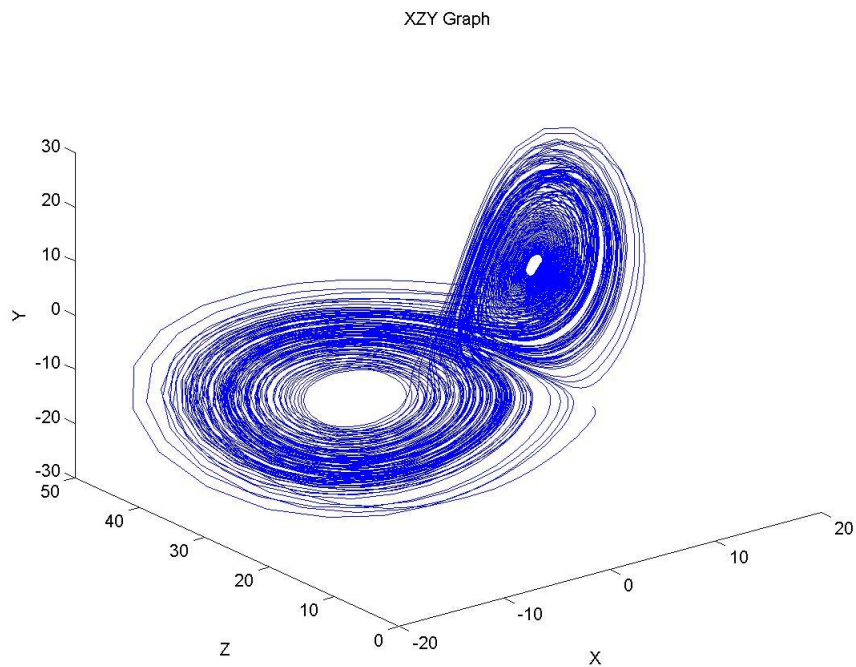
Σχήμα 5.26: Διάγραμμα yz για $\rho=22-28$ με βήμα 2



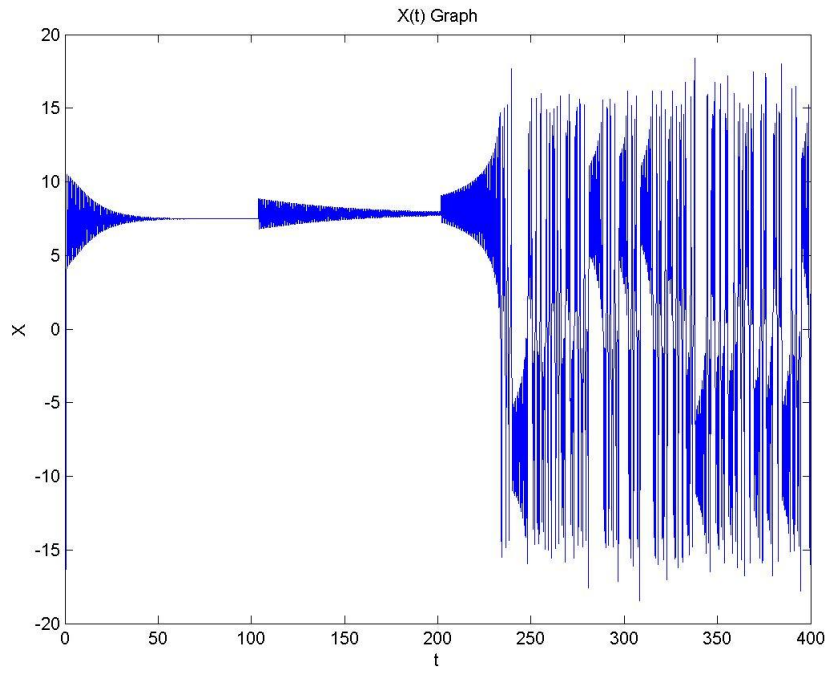
Σχήμα 5.27: Διάγραμμα xz για $\rho=22-28$ με βήμα 2



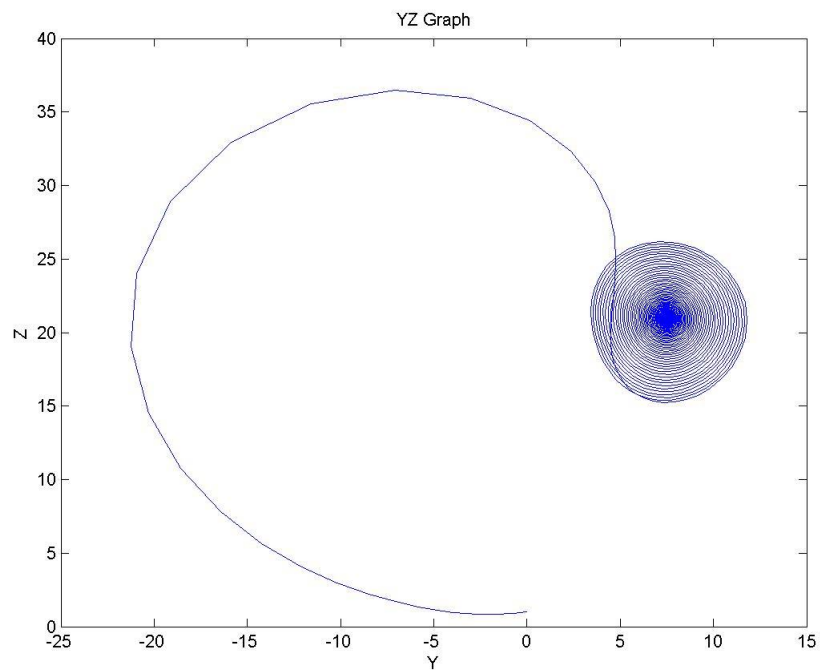
Σχήμα 5.28: Διάγραμμα xy για $\rho=22-28$ με βήμα 2



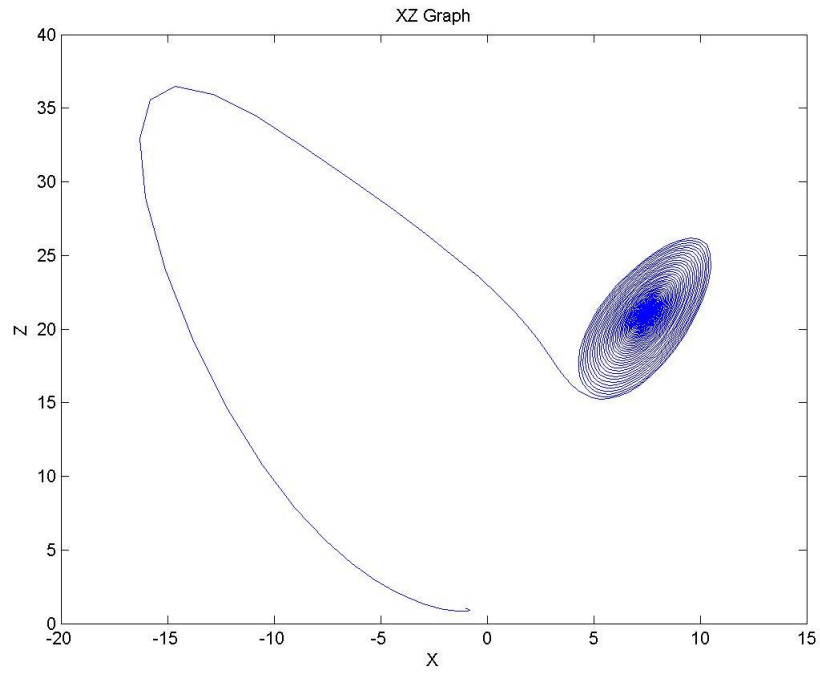
Σχήμα 5.29: Διάγραμμα 3D - xyz για $\rho=22-28$ με βήμα 2



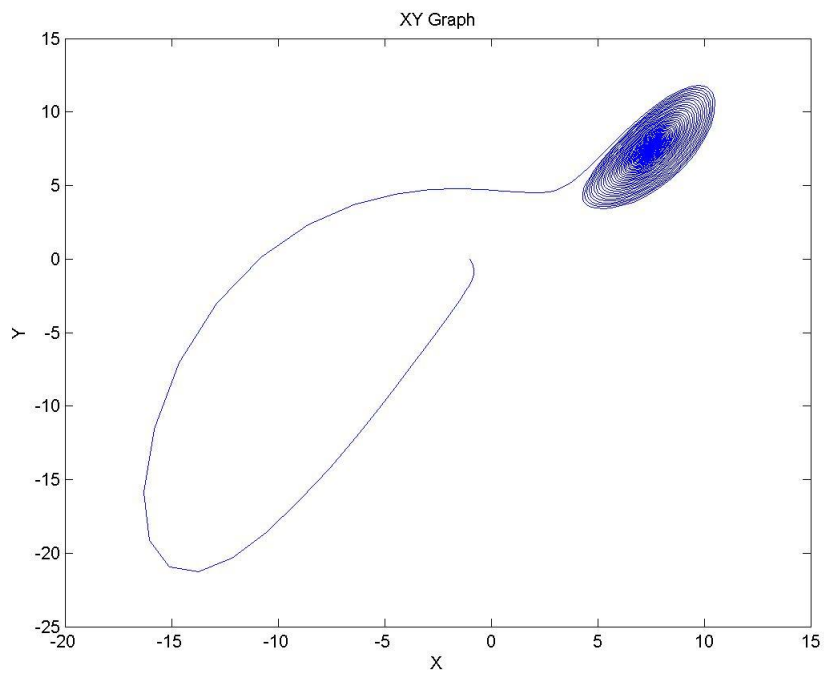
Σχήμα 5.30: Κυματομορφή $x(t)$ για $\rho=22-28$ με βήμα 2



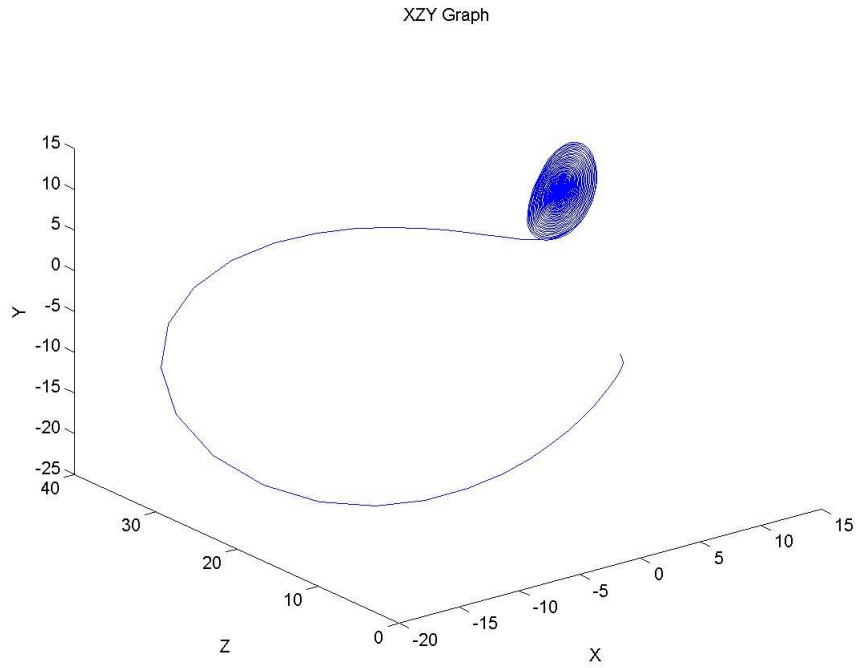
Σχήμα 5.31: Διάγραμμα yz για $\rho=22$



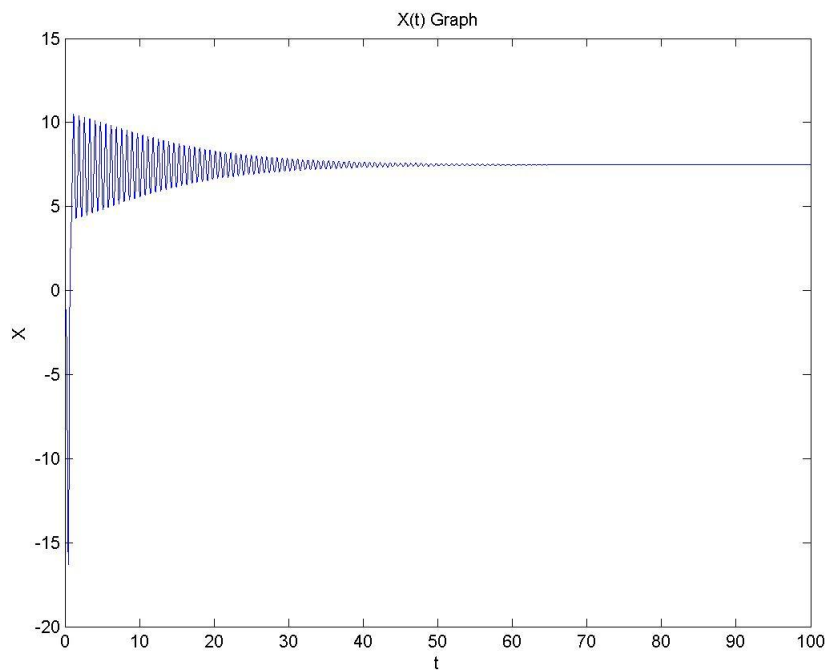
Σχήμα 5.32: Διάγραμμα xz για $\rho=22$



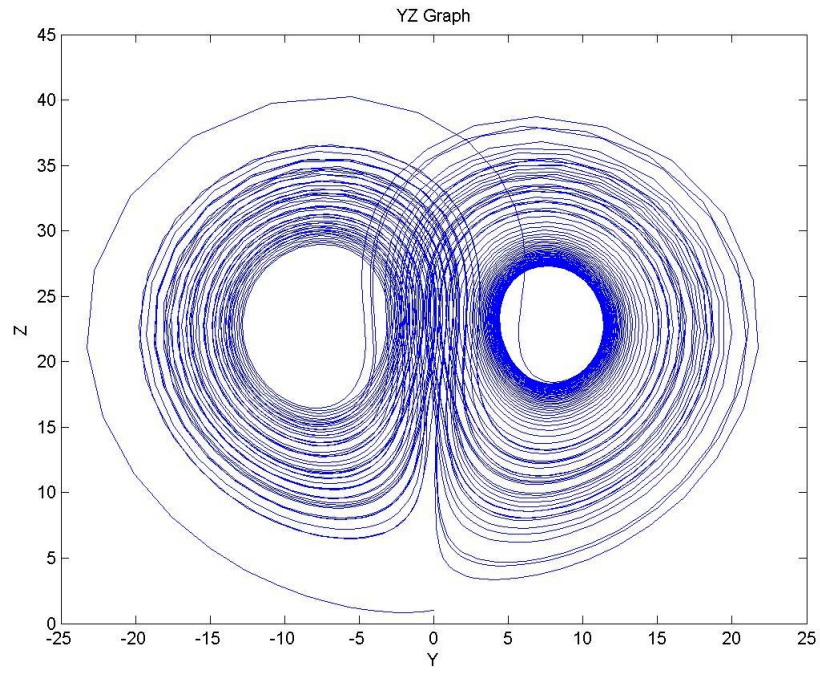
Σχήμα 5.33: Διάγραμμα xy για $\rho=22$



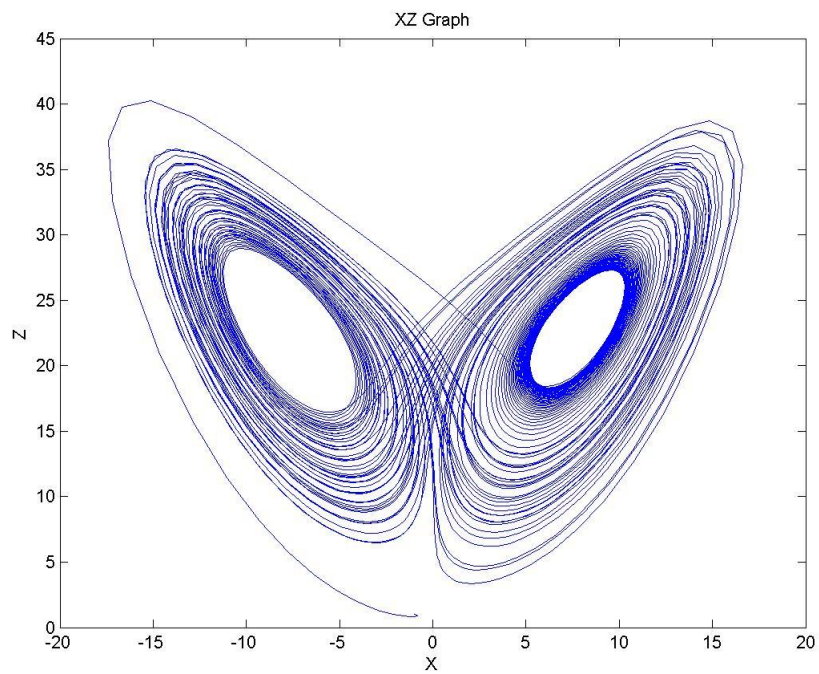
Σχήμα 5.34: Διάγραμμα 3D - xyz για $\rho=22$



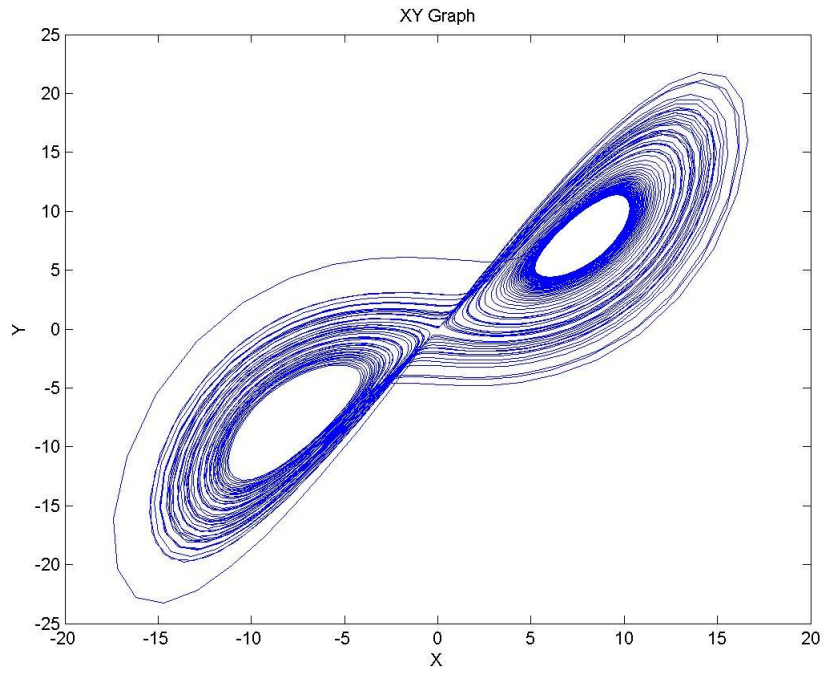
Σχήμα 5.35: Κυματομορφή $x(t)$ για $\rho=22$



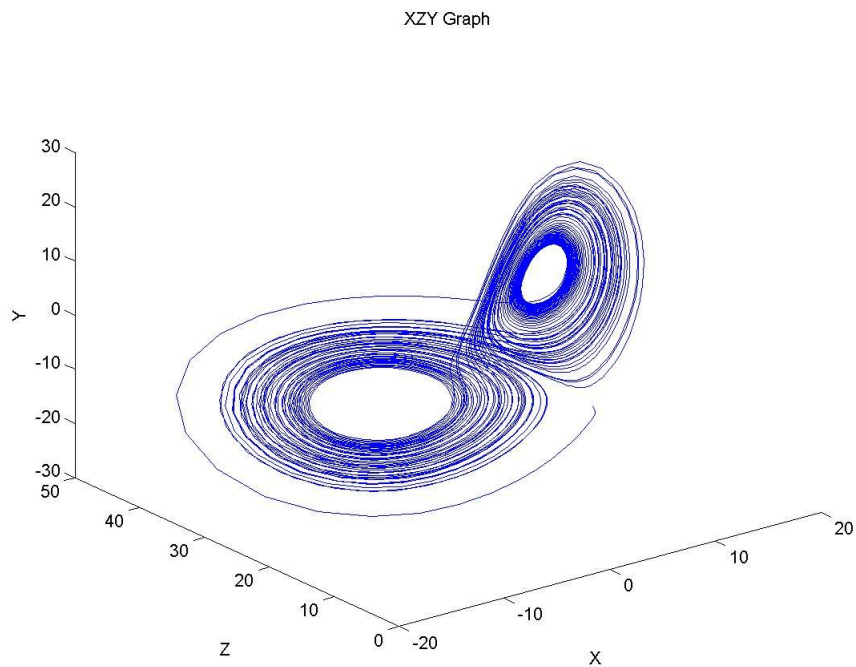
Σχήμα 5.36: Διάγραμμα yz για $\rho=24$



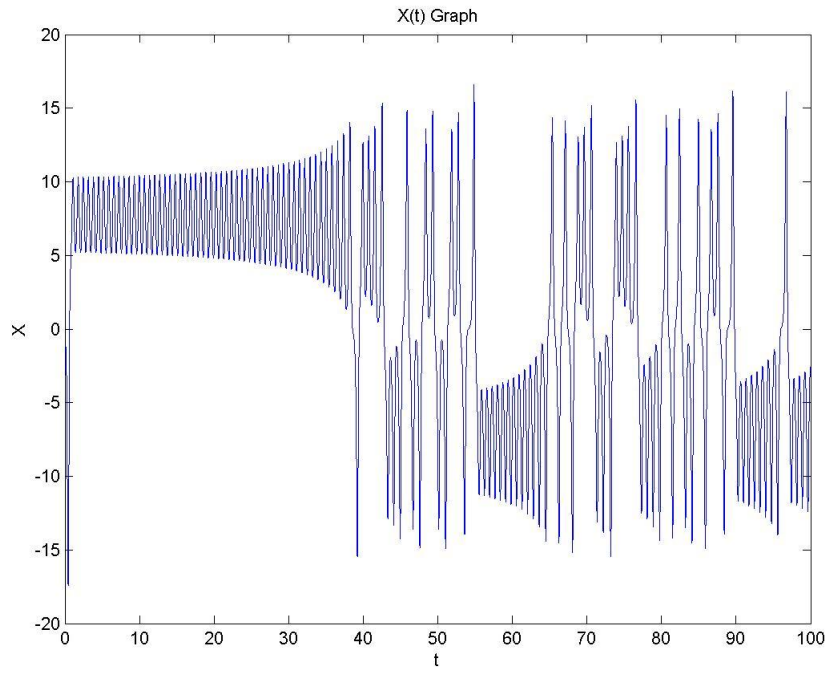
Σχήμα 5.37: Διάγραμμα xz για $\rho=24$



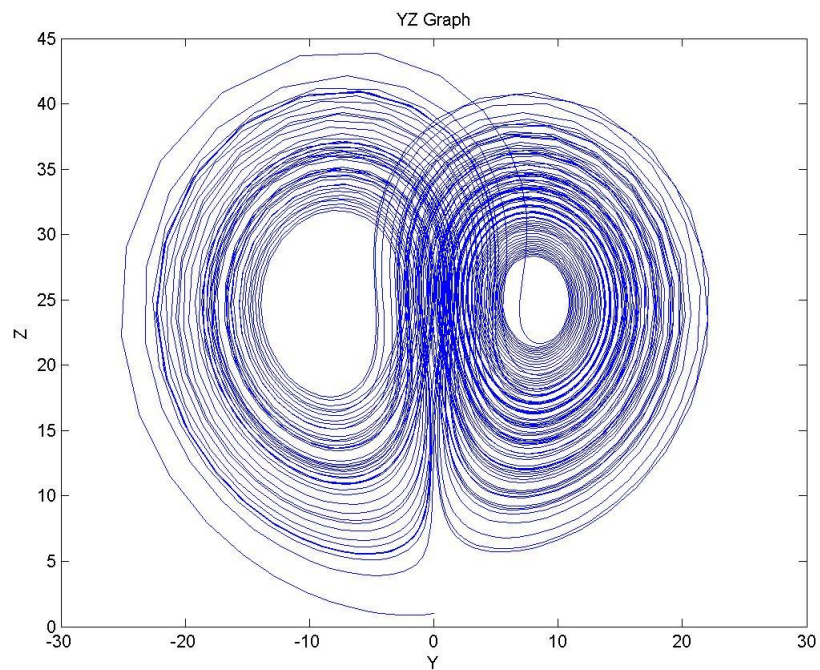
Σχήμα 5.38: Διάγραμμα xy για $\rho=24$



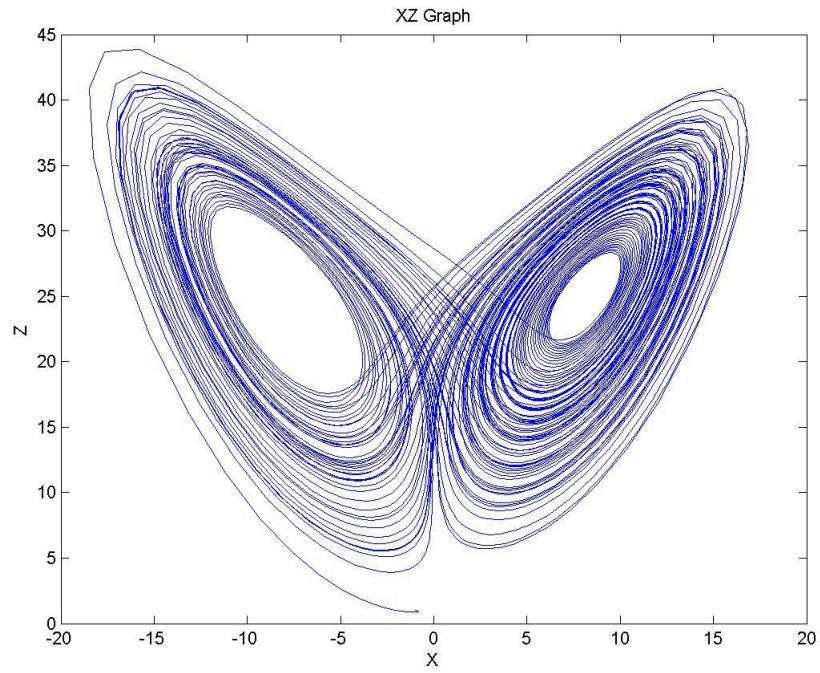
Σχήμα 5.39: Διάγραμμα 3D - xyz για $\rho=24$



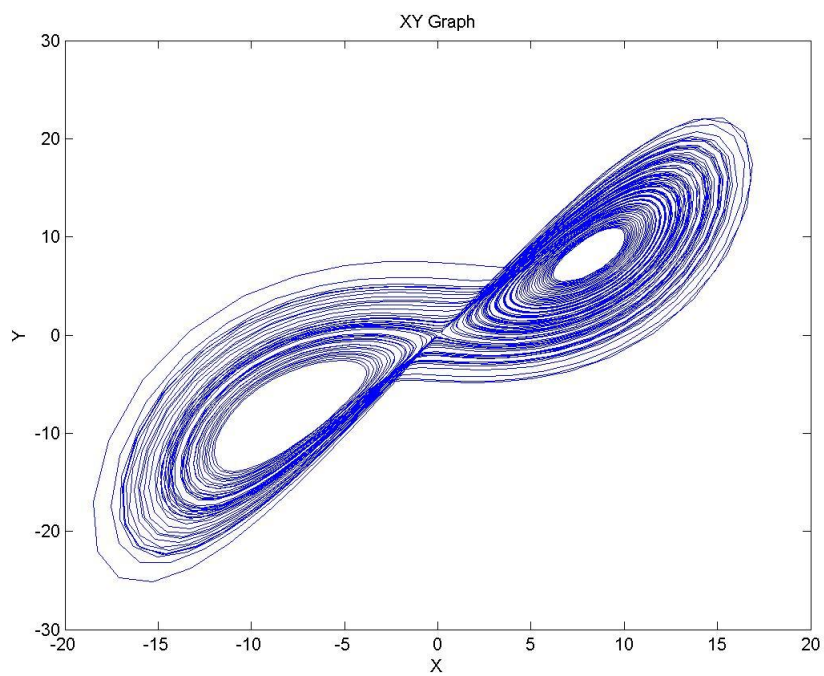
Σχήμα 5.40: Κυματομορφή $x(t)$ για $\rho=24$



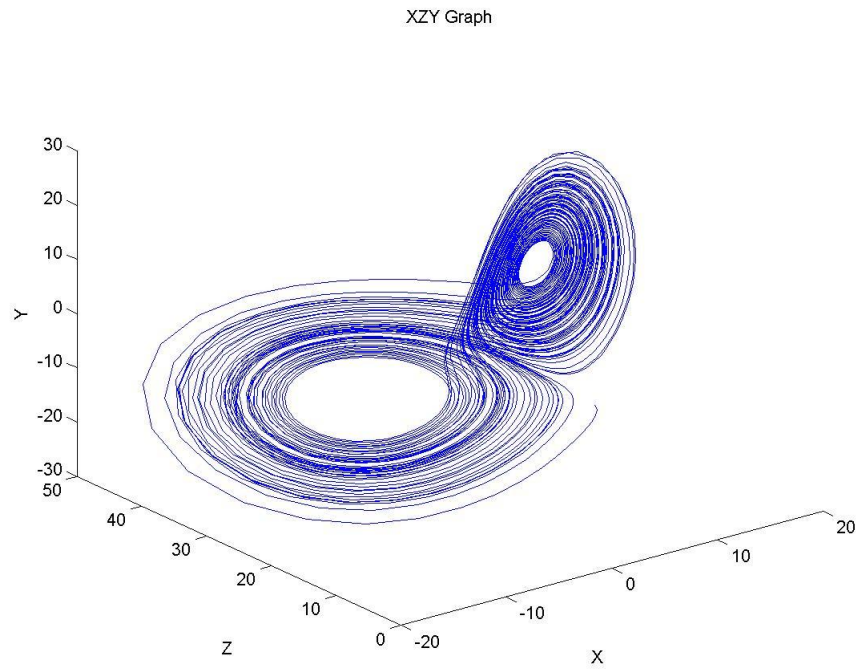
Σχήμα 5.41: Διάγραμμα yz για $\rho=26$



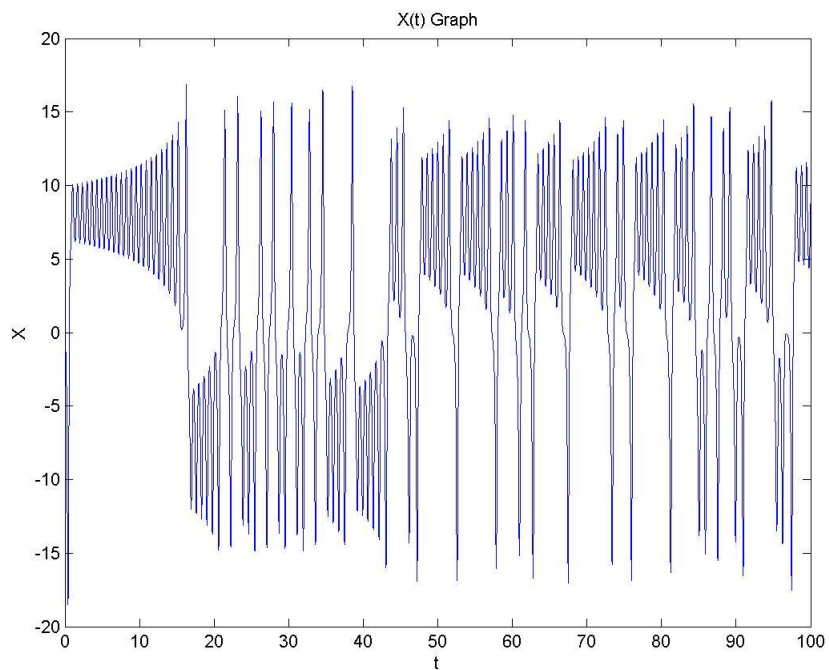
Σχήμα 5.42: Διάγραμμα xz για $\rho=26$



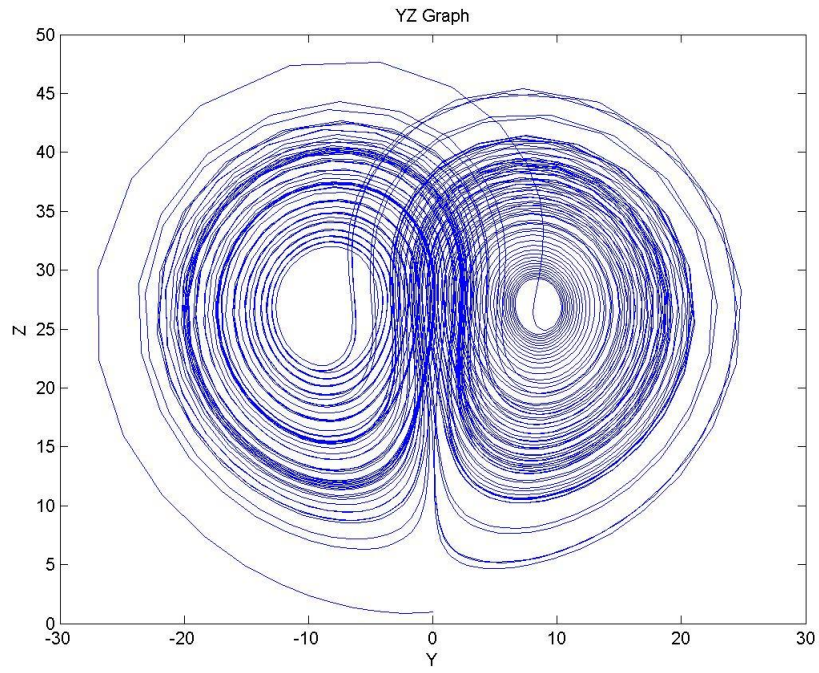
Σχήμα 5.43: Διάγραμμα xy για $\rho=26$



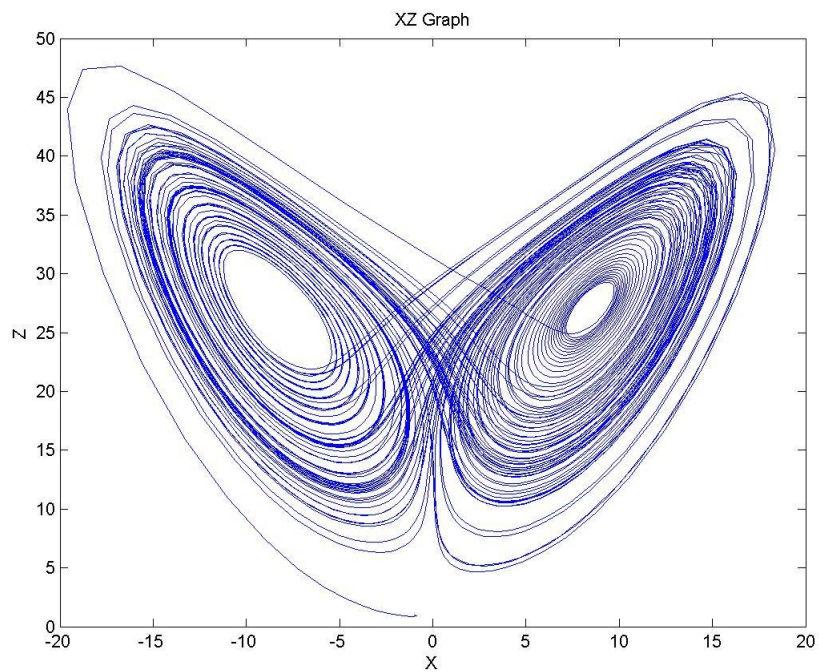
Σχήμα 5.44: Διάγραμμα 3D - xyz για $\rho=26$



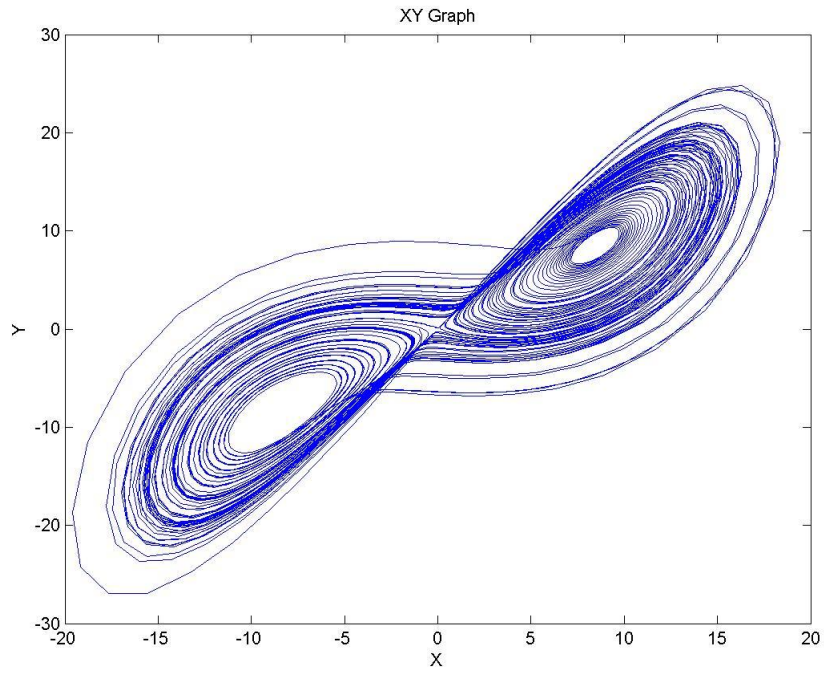
Σχήμα 5.45: Κυματομορφή $x(t)$ για $\rho=26$



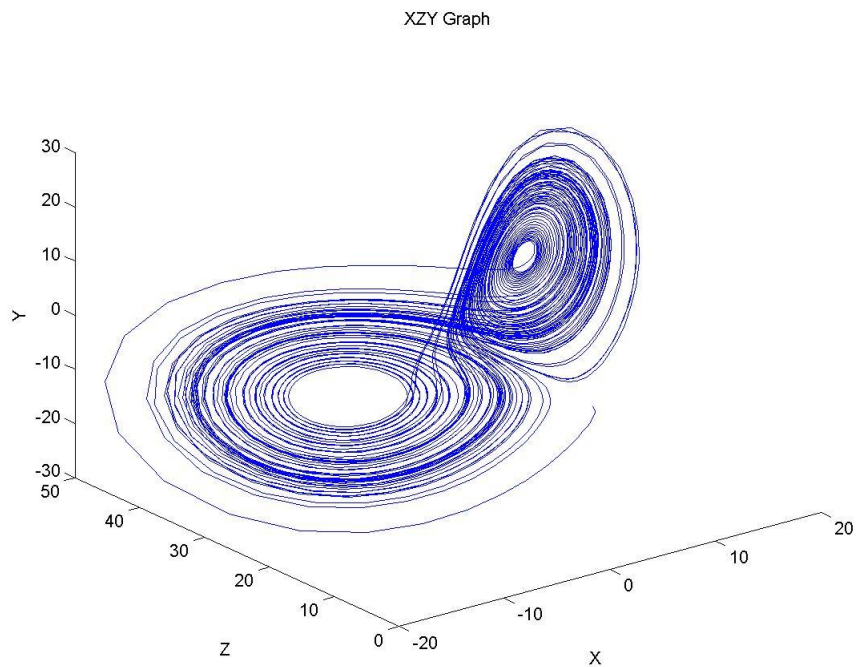
Σχήμα 5.46: Διάγραμμα yz για $\rho=28$



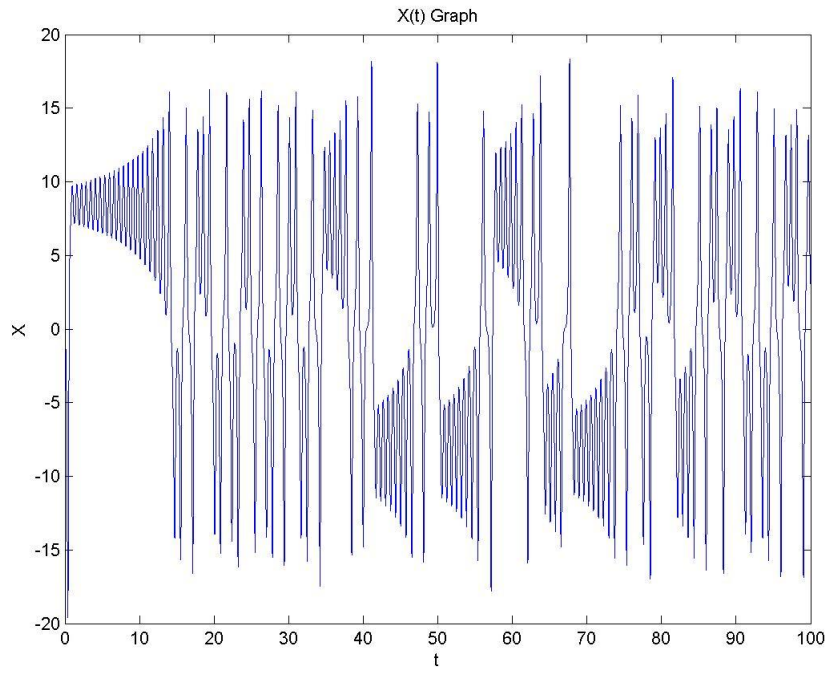
Σχήμα 5.47: Διάγραμμα xz για $\rho=28$



Σχήμα 5.48: Διάγραμμα xy για $\rho=28$



Σχήμα 5.49: Διάγραμμα 3D - xyz για $\rho=28$



Σχήμα 5.50: Κυματομορφή $x(t)$ για $\rho=28$

Κεφάλαιο 6

Συμπεράσματα

6.1 Ανακεφαλαίωση

Στην παρούσα πτυχιακή εργασία σχεδιάστηκε και αναπτύχθηκε ένα λογισμικό επίλυσης του συστήματος Lorenz. Το σύστημα αυτό είναι ένα δυναμικό σύστημα τριών διαφορικών εξισώσεων οι οποίες περιγράφονται από τις σχέσεις:

$$\frac{dx}{dt} = \sigma(y - x) \quad (6.1)$$

$$\frac{dy}{dt} = \rho x - y - xz \quad (6.2)$$

$$\frac{dz}{dt} = xy - \beta z \quad (6.3)$$

όπου σ , ρ , β είναι οι παράμετροι του συστήματος. Το σ ονομάζεται αριθμός Prandtl, το ρ ονομάζεται αριθμός Rayleigh και το β δεν έχει όνομα.

Το λογισμικό που αναπτύχθηκε προσομοιώνει τη συμπεριφορά του συστήματος Lorenz και δίνει στο χρήστη τη δυνατότητα, μέσω ενός εύχρηστου γραφικού περιβάλλοντος, να απεικονίσει και να αποθηκεύσει τις μεταβλητές του συστήματός του. Ο χρήστης μπορεί να αλλάξει τις παραμέτρους του συστήματος, να θέσει τις συνθήκες στις οποίες θέλει να επικεντρώσει τη μελέτη του και να δει τα αποτελέσματα σε μία, δύο ή τρεις διαστάσεις. Η υλοποίηση του προγράμματος έγινε με τη χρήση γραφικού περιβάλλοντος (Graphical User Interface) σε συνδυασμό με matlab/ simulink.

6.2 Συμπεράσματα

Σχεδιάζοντας, υλοποιώντας αλλά και χρησιμοποιώντας το λογισμικό για την επίλυση των διαφορικών εξισώσεων Lorenz καταλήγουμε σε μία σειρά χρήσιμων συμπερασμάτων. Αρχικά, καθίσταται φανερό το πλεονέκτημα που μας προσφέρει η χρήση γραφικού περιβάλλοντος. Δίνεται η δυνατότητα δημιουργίας πολύπλοκων εφαρμογών που μπορούν να προσαρμοστούν στις ανάγκες και τις ιδιαιτερότητες του κάθε χρήστη, ο οποίος δεν είναι απαραίτητο να γνωρίζει τον τρόπο ανάπτυξης τους. Η χρήση του λογισμικού δεν προϋποθέτει τη γνώση του τρόπου δημιουργίας του.

Επίσης, τα Matlab/ Simulink μας δίνουν τη δυνατότητα χρησιμοποίησης έτοιμων πολύπλοκων αριθμητικών μεθόδων επίλυσης διαφορικών εξισώσεων. Με αυτόν τον τρόπο γίνεται φανερή η χρησιμότητα του Matlab σε συνδυασμό με το Simulink για το σχεδιασμό και την επίλυση μη-γραμμικών διαφορικών συστημάτων και γενικότερα συστημάτων τα οποία παρουσιάζουν χαστική συμπεριφορά.

Με την ανάπτυξη του συγκεκριμένου λογισμικού έγινε εφικτή η συγκέντρωση μιας πληθώρας αριθμητικών μεθόδων επίλυσης διαφορικών εξισώσεων σε μία φόρμα εφαρμογής. Με αυτόν τον τρόπο είναι απλή και εύχρηστη η σύγκριση διαφορετικών μεθόδων και αποτελεσμάτων για το ίδιο σύστημα. Με εύκολο τρόπο έχουμε πρόσβαση σε όλες τις παραμέτρους του συστήματος, άρα και πλήρη έλεγχο των αποτελεσμάτων.

Τέλος, μπορούμε εύκολα και γρήγορα να λάβουμε αποτελέσματα για διαφορετικές αριθμητικές μεθόδους ολοκλήρωσης και επομένως να επιλέξουμε ποια μέθοδος είναι η καταλληλότερη για τις εκάστοτε επιλεγόμενες ρυθμίσεις του συστήματος μας.

6.3 Προοπτικές

Το λογισμικό που αναπτύχθηκε μας δίνει τη δυνατότητα μελέτης του συστήματος Lorenz. Μια μελλοντική επέκταση αυτού είναι η δυνατότητα λήψης διαγράμματος διακλάδωσης δύο ή και τριών διαστάσεων. Επίσης, θα ήταν χρήσιμο σε μελλοντικές χρήσεις του να αναπτυχθούν και διαγράμματα που σε πραγματικό χρόνο θα παρουσιάζουν διαφορετικές τροχιές του συστήματος για διαφορετικές ρυθμίσεις του. Γενικά, οι δυνατότητες του λογισμικού περιορίζονται μόνο από τη δημιουργικότητα του προγραμματιστή.

ΑΝΑΦΟΡΕΣ

Βιβλιογραφία

- [1] K.M. Cuomo, A.V. Oppenheim, *Circuit Implementation Of Synchronized Chaos with Application to Communications*, Physical Review Letters, Vol. 71, No. 1, pp. 65-68, 1993.
- [2] K.M. Cuomo, A.V. Oppenheim, S.H. Strogatz, *Synchronization of Lorenz-Based Chaotic Circuits with Application to Communications*, IEEE Transactions On Circuits and Systems, Vol. 40, No. 10, pp. 626-633, 1993.
- [3] J.B. Dabney, T.L. Harman, *Mastering Simulink*, Pearson Prentice Hall, 2004.
- [4] J. Gleick, *Chaos: Making a New Science*, Heinemann, London, 1988.
- [5] D.C. Hanselman, B.L. Littlefield, *Mastering MATLAB 7*, Pearson Prentice Hall, 2005.
- [6] S.T. Karris, *Introduction to Simulink with Engineering Applications*, Orchard Publications, 2006.
- [7] T.H. Liao, C.K. Chen, *Design and Circuit Simulation of observer-based chaotic synchronization and communication systems*, International Journal of Electronics, Vol. 86, No. 12, pp. 1432-1440, 1999.
- [8] E.N. Lorenz, *Deterministic Nonperiodic Flow*, J. Atmos. Sci., 20, pp. 130-141, 1963.
- [9] C. Sparrow, *The Lorenz Equations: Bifurcations, Chaos, and Strange Attractors*, Springer-Verlag, 1982.
- [10] S.H. Strogatz, *Nonlinear Dynamics and Chaos*, Addison-Wesley, 1994.
- [11] R.F. Williams, *The Structure of Lorenz Attractors*, Publ. Math. IHES, 50, pp. 321-347, 1979.
- [12] Μ.Σ. Παπαδοπούλου, *Δημιουργία Γραφικής Διεπαφής Χρήστη για τη λήψη Μετρήσεων*, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης-Τμήμα Φυσικής, 2008.

Ηλεκτρονικές πηγές

- [1] <http://en.wikipedia.org>
- [2] <http://www.eece.maine.edu/mm/>
- [3] <http://www.mathworks.com>