



Εθνικό Μετσόβιο Πολυτεχνείο  
Σχολή Χημικών Μηχανικών  
Τομέας Ανάλυσης, Σχεδιασμού και  
Ανάπτυξης Διεργασιών και Συστημάτων

**Διπλωματική Εργασία:**

**Εξαγωγή μακροσκοπικών/αδρομερών διαφορικών εξισώσεων από υπολογιστικά πειράματα μικροκλίμακας με χρήση μεθόδων μηχανικής μάθησης**

**ΑΘΑΝΑΣΑΚΗΣ ΠΑΥΛΟΣ**

**Επιβλέπων Καθηγητής:  
ΚΑΒΟΥΣΑΝΑΚΗΣ ΜΙΧΑΗΛ**

**ΑΘΗΝΑ, ΣΕΠΤΕΜΒΡΗΣ 2023**



Εθνικό Μετσόβιο Πολυτεχνείο  
Σχολή Χημικών Μηχανικών  
Τομέας Ανάλυσης, Σχεδιασμού και  
Ανάπτυξης Διεργασιών και Συστημάτων

**Διπλωματική Εργασία:**

**Εξαγωγή μακροσκοπικών/αδρομερών διαφορικών εξισώσεων από υπολογιστικά πειράματα μικροκλίμακας με χρήση μεθόδων μηχανικής μάθησης**

**ΑΘΑΝΑΣΑΚΗΣ ΠΑΥΛΟΣ**

**Τριμελής επιτροπή:**

**ΚΑΒΟΥΣΑΝΑΚΗΣ ΜΙΧΑΗΛ ΚΑΘΗΓΗΤΗΣ Χ.Μ. Ε.Μ.Π.  
ΚΟΚΚΟΡΗΣ ΓΕΩΡΓΙΟΣ ΚΑΘΗΓΗΤΗΣ Χ.Μ. Ε.Μ.Π.  
ΠΑΥΛΑΤΟΥ ΕΥΑΓΓΕΛΙΑ ΚΑΘΗΓΗΤΡΙΑ Χ.Μ. Ε.Μ.Π.**

**ΑΘΗΝΑ, ΣΕΠΤΕΜΒΡΗΣ 2023**

## Ευχαριστίες

Η παρούσα διπλωματική εργασία σηματοδοτεί το πέρας του 5ετούς κύκλου σπουδών μου στο Εθνικό Μετσόβιο Πολυτεχνείο. Ολοκληρώνοντας αυτό το ταξίδι θα ήθελα να ευχαριστήσω τους ανθρώπους που με βοήθησαν και που συνέβαλαν στην προσπάθειά μου.

Αρχικά θα ήθελα να ευχαριστήσω τον καθηγητή Καβουσανάκη Μιχαήλ ο οποίος με εμπιστεύτηκε με αυτό το θέμα και με στήριξε καθ' όλη τη διάρκεια της εκπόνησής του. Ο χρόνος που αφιέρωσε, οι πολύτιμες συμβουλές που μου προσέφερε και η βοήθεια που παρείχε κατά την εκπόνηση του κώδικα υπήρξαν καταλυτικά στην προσπάθειά μου.

Επίσης θα ήθελα να ευχαριστήσω τις καθηγήτριες Μικέδη Αικατερίνη και Κροκίδα Μαγδαληνή που πίστεψαν σε εμένα και με έφεραν σε επαφή με τον ακαδημαϊκό κόσμο και την έρευνα από τα πρώτα βήματα μου σε αυτή τη σχολή. Υπό την καθοδήγησή τους είχα μια πρώτη επαφή με το πως λειτουργεί η έρευνα σε ακαδημαϊκό επίπεδο και ξεκίνησε να παίρνει μορφή η αγάπη μου για την επιστήμη του χημικού μηχανικού και την προσφορά σε αυτή.

Τέλος θα ήθελα να ευχαριστήσω την οικογένεια μου και ιδιαίτερα τον παππού μου, Άγη, ο οποίος υπήρξε ο μεγαλύτερος υποστηρικτής μου από το σημείο μηδέν.

## Περίληψη

Στην παρούσα διπλωματική εργασία εξετάζεται η χρήση μεθόδων μηχανικής μάθησης για την εκμάθηση και επίλυση μερικών διαφορικών εξισώσεων όπου είναι άγνωστο το δεξί μέλος τους. Στόχος της εργασίας είναι η θεμελίωση μεθόδου με την οποία θα μπορεί να εξαχθεί η λύση της διαφορικής εξίσωσης από δεδομένα πειραμάτων μικροκλίμακας, χωρίς καμία γνώση της έκφρασης της εξίσωσης.

Αρχικά εφαρμόζονται οι μέθοδοι που εξετάζονται στην παρούσα εργασία σε δύο προβλήματα όπου είναι γνωστή η έκφραση της διαφορικής εξίσωσης, επαληθεύοντας την δύναμη των εργαλείων. Μελετώνται το πρόβλημα της μοριακής διάχυσης και το πρόβλημα Bratu. Η επίλυση των διαφορικών εξισώσεων γίνεται με την εντολή του MATLAB, ode15s, και έπειτα προσεγγίζονται αριθμητικά οι χωρικές και η χρονική παράγωγος τους με την μέθοδο των πεπερασμένων διαφορών. Εξετάζεται η δυναμική των προβλημάτων σε μία διάσταση. Εφαρμόζονται Γκαουσιανές Διεργασίες σε ένα σετ 4800 δεδομένων για την λύση, την πρώτη και δεύτερη χωρική παράγωγο και την χρονική παράγωγο των εξισώσεων για την εύρεση των σημαντικών παραμέτρων των εξισώσεων. Στο πρόβλημα της μοριακής διάχυσης προσδιορίζεται ως σημαντική μόνο η δεύτερη παράγωγος της λύσης ως προς την διεύθυνση  $x$  ενώ στο πρόβλημα Bratu αναδεικνύεται η δεύτερη παράγωγος της λύσης ως προς τη διεύθυνση  $x$  και η ίδια η λύση. Έπειτα, χρησιμοποιώντας ως δεδομένα εκπαίδευσης τα δεδομένα για τις σημαντικές παραμέτρους που χρησιμοποιήθηκαν και για την εφαρμογή γκαουσιανών διεργασιών, εκπαιδεύονται ρηγά τεχνητά νευρωνικά δίκτυα στην επίλυση των δύο προβλημάτων.

Για το πρόβλημα της μοριακής διάχυσης διαπιστώνεται βέλτιστη αρχιτεκτονική με 1 νευρώνα στο στρώμα εισόδου και μόνο ένα κρυφό στρώμα, με 3 νευρώνες. Η προσαρμογή του δικτύου εξετάστηκε σε δεδομένα από αρχικές κατανομές διαφορετικές από αυτές στις οποίες έγινε η εκπαίδευση και για μεγαλύτερους χρόνους από αυτούς στους οποίους επιλύθηκαν οι διαφορικές και υπολογίστηκαν οι δείκτες αξιολόγησης, ο AUF της τάξης  $10^{-5} - 10^{-8}$  και ο MAPE της τάξης  $10^{-5} - 10^{-7}$ .

Για το πρόβλημα Bratu διαπιστώνεται βέλτιστη αρχιτεκτονική με 1 νευρώνα στο στρώμα εισόδου και μόνο ένα κρυφό στρώμα, με 5 νευρώνες. Η προσαρμογή του δικτύου εξετάστηκε σε δεδομένα από αρχικές κατανομές διαφορετικές από αυτές στις οποίες έγινε η εκπαίδευση και για μεγαλύτερους χρόνους από αυτούς στους οποίους επιλύθηκαν οι διαφορικές και υπολογίστηκαν οι δείκτες αξιολόγησης, ο AUF της τάξης  $10^{-4} - 10^{-8}$  και ο MAPE της τάξης  $10^{-4} - 10^{-6}$ .

Τέλος εξετάστηκε μοντέλο προσομοιωτή μικροσκοπικής κλίμακας, το μοντέλο των Random Walkers. Αποτελεί ένα πείραμα Monte Carlo που προσομοιάζει τη φαινόμενο της μοριακής διάχυσης. Έγινε επίλυση για τις θέσεις ενός πληθυσμού  $10^4$  βηματιστών σε διάφορες χρονικές στιγμές και υπολογίστηκε η συνάρτηση της συγκέντρωσής τους στο χωρίο που μελετάται. Με εφαρμογή Γκαουσιανών Διεργασιών στα δεδομένα του πειράματος αναδείχθηκε η δεύτερη παράγωγος της συνάρτησης της συγκέντρωσης βηματιστών στη διεύθυνση  $x$  ως η μοναδική σημαντική μεταβλητή του προβλήματος. Έπειτα εκπαιδεύτηκε νευρωνικό δίκτυο στην επίλυση του προσομοιωτή υπολογίστηκαν οι δείκτες αξιολόγησης, ο AUF της τάξης  $10^{-3} - 10^{-4}$  και ο MAPE της τάξης  $10^{-3} - 10^{-4}$ .

Η εκπόνηση του κώδικα της διπλωματικής έγινε σε περιβάλλον MATLAB.

Λέξεις κλειδιά: Μηχανική μάθηση, τεχνητά νευρωνικά δίκτυα, Γκαουσιανές διεργασίες, επίλυση διαφορικών εξισώσεων, προσομοιωτής μικροκλίμακας, μοριακή διάχυση, πρόβλημα Bratu, αριθμητική προσέγγιση, πεπερασμένες διαφορές

## Abstract

In this thesis machine learning methods are examined for learning and solving partial differential equations (PDEs) where the right hand side is unknown. The goal of this study is developing a method for extracting the solution of a differential equation only through data from microscale experiments, without knowledge of the equation.

Initially, the methods examined in this study are used in two problems where the differential equation is known, in order to validate the power of the tools. The problem of molecular diffusion and the Bratu problem are studied. The differential equations are solved by the MATLAB command `ode15s` and the derivatives of the solutions are approximated numerically by the method of finite differences. The problems are studied in only one spatial dimension. Gaussian Processes are applied on a set of 4800 data points of the solution, the first and second spatial derivative and the time derivative of the solutions of the PDEs, in order to identify the important variables of the equations. In the problem of molecular diffusion only the second spatial derivative with respect to dimension  $x$  is identified as important while in the Bratu problem the solution of the PDE and the second spatial derivative are identified as important variables. Then using the data of the important variables used to implement Gaussian Processes, are trained shallow artificial neural networks at solving the PDEs.

In the problem of molecular diffusion the best identified network architecture is a structure with 1 neuron on the first layers and only 1 hidden layer with 3 neurons. The fit of the network is assessed on data from new initial conditions different than the ones the networks were trained and for larger times than the ones the PDEs were solved for. The assessment indices were calculated, AUF in the order of  $10^{-5} - 10^{-8}$  and MAPE in the order of  $10^{-4} - 10^{-6}$ .

In the Bratu problem the best identified network architecture is a structure with 1 neuron on the first layers and only 1 hidden layer with 5 neurons. The fit of the network is assessed on data from new initial conditions than the on the ones the networks were trained and for larger times than the ones the PDEs were solved for. The assessment indices were calculated, AUF in the order of  $10^{-4} - 10^{-8}$  and MAPE in the order of  $10^{-4} - 10^{-6}$ .

Lastly we examined the model of a microscale simulator, the Rndom Walkers model. It is a Monte Carlo experiment simulating the dynamics of molecular diffusion. It was solved for the positions of a population of  $10^5$  walkers in different time steps and their density function in their domain is calculated. Gaussian Processess on the experiments' data showed that the most important variable of the problem is the second spatial derivative of the density function of the walkers in the  $x$  direction. Then a shallow neural network was trained on solving the microscale simulator model. The assessment indices were calculated, AUF in the order of  $10^{-3} - 10^{-4}$  and MAPE in the order of  $10^{-3} - 10^{-4}$ .

All the code was written in a MATLAB environment.

Keywords: Machine learning, artificial neural networks, Gaussian Processess, solving differential equations, microscale simulator, molecular diffusion, Bratu problem, numerical approximation, finite differences

## Περιεχόμενα

.....	1
Ευχαριστίες.....	3
Περίληψη.....	4
Abstract .....	5
Περιεχόμενα .....	6
Κατάλογος Εικόνων .....	9
Κατάλογος Πινάκων.....	13
1 Κεφάλαιο 1 <sup>ο</sup> : Εισαγωγή.....	14
1.1 Διαφορικές Εξισώσεις.....	14
1.1.1 Ταξινόμηση Διαφορικών Εξισώσεων.....	14
1.1.2 Επίλυση Διαφορικών Εξισώσεων.....	15
1.1.3 Προβλήματα Αρχικών και Συνοριακών τιμών .....	15
1.1.4 Αριθμητική Επίλυση Διαφορικών Εξισώσεων.....	16
1.1.5 Πρόβλημα Bratu .....	17
1.2 Μέθοδοι Monte Carlo.....	19
1.3 Μηχανική Μάθηση.....	20
1.3.1 Ταξινόμηση μεθόδων Μηχανικής Μάθησης.....	21
1.3.2 Τεχνητά Νευρωνικά Δίκτυα .....	21
1.3.3 Γκαουσιανές διεργασίες .....	27
2 Κεφάλαιο 2 <sup>ο</sup> : Μεθοδολογία/ Επίλυση Δ.Ε. με μεθόδους Μηχανικής Μάθησης σε περιβάλλον MATLAB.....	29
2.1 Μοριακή Διάχυση σε Μια Διάσταση.....	29
2.1.1 Καθορισμός Αρχικών Κατανομών .....	29
2.1.2 Αριθμητική Επίλυση της Δ.Ε. ....	31
2.1.3 Γκαουσιανές Διεργασίες.....	31
2.1.4 Εκπαίδευση Νευρωνικού Δικτύου .....	32
2.2 Bratu σε Μια Διάσταση.....	34
2.2.1 Καθορισμός Αρχικών Κατανομών .....	34
2.2.2 Αριθμητική Επίλυση του προβλήματος .....	34
2.2.3 Γκαουσιανές Διεργασίες.....	35
2.2.4 Εκπαίδευση Νευρωνικού Δικτύου .....	35
2.3 Random Walkers .....	36
2.3.1 Αρχικές Θέσεις Βηματιστών .....	36
2.3.2 Επίλυση Μοντέλου.....	38
2.3.3 Γκαουσιανές Διεργασίες.....	38
2.3.4 Εκπαίδευση Νευρωνικού Δικτύου .....	38

2.4	Εφαρμογή σε περιβάλλον MATLAB.....	39
2.4.1	Επίλυση Δ.Ε. με την ode15s/μέθοδο πεπερασμένων διαφορών.....	39
2.4.2	Επίλυση μοντέλου τυχαίων βηματιστών.....	40
2.4.3	Εφαρμογή διεργασιών Γκάους.....	40
2.4.4	Εκπαίδευση νευρωνικών δικτύων.....	41
3	Κεφάλαιο 3 <sup>ο</sup> : Παρουσίαση Αποτελεσμάτων και Σχολιασμός.....	43
3.1	Αποτελέσματα 1-D διάχυσης.....	43
3.1.1	Αποτελέσματα επίλυσης της Δ.Ε. μοριακής διάχυσης.....	43
3.1.2	Αποτελέσματα Γκαουσιανών Διεργασιών.....	44
3.1.3	Αποτελέσματα και προσαρμογή νευρωνικού δικτύου.....	46
3.2	Αποτελέσματα 1-D Bratu.....	48
3.2.1	Αποτελέσματα επίλυσης της Δ.Ε. του προβλήματος Bratu.....	48
3.2.2	Αποτελέσματα Γκαουσιανών Διεργασιών.....	50
3.2.3	Αποτελέσματα και προσαρμογή νευρωνικού δικτύου.....	51
3.3	Αποτελέσματα Μοντέλου Τυχαίων Βηματιστών.....	54
3.3.1	Αξιολόγηση ορθότητας των υπολογισμού του μοντέλου τυχαίων βηματιστών 54	
3.3.2	Διερεύνηση της επίδρασης του θορύβου στα δεδομένα της επίλυσης του μοντέλου των τυχαίων βηματιστών.....	57
3.3.3	Εφαρμογή γκαουσιανών διεργασιών.....	66
3.3.4	Αποτελέσματα και προσαρμογή νευρωνικού δικτύου.....	67
4	Κεφάλαιο 4 <sup>ο</sup> : Συμπεράσματα και Προτάσεις.....	70
4.1	Διεργασίες Γκάους.....	70
4.2	Εκπαίδευση Νευρωνικών Δικτύων.....	70
4.3	Μεθοδολογία.....	70
4.4	Προτάσεις για περαιτέρω έρευνα.....	70
4.4.1	Επιτάχυνση μεθοδολογίας.....	70
4.4.2	Επίλυση των προβλημάτων σε ανώτερες διαστάσεις.....	70
4.4.3	Μελέτη πιο περίπλοκων διαφορικών εξισώσεων.....	70
4.4.4	Μελέτη πιο περίπλοκων μοντέλων προσομοιωτών μικροκλίμακας.....	71
5	Βιβλιογραφία.....	73
6	Παράρτημα.....	75
6.1	Αριθμητική επίλυση της Δ.Ε. μοριακής διάχυσης σε μεταβατική κατάσταση, σε μια διάσταση με την μέθοδο των πεπερασμένων διαφορών.....	75
6.2	Αριθμητική επίλυση του προβλήματος Bratu σε μεταβατική κατάσταση, σε μια διάσταση με την μέθοδο των πεπερασμένων διαφορών.....	76
6.3	Κώδικας επίλυσης Δ.Ε. μοριακής διάχυσης σε μεταβατική κατάσταση, σε μία διάσταση.....	76

6.4	Κώδικας επίλυσης προβλήματος Bratu σε μεταβατική κατάσταση, σε μία διάσταση 78	
6.5	Κώδικας επίλυσης μοντέλου τυχαίων βηματιστών .....	80
6.6	Κώδικας για εφαρμογή Γκαουσιανών Διεργασιών .....	84
6.7	Κώδικας εκπαίδευσης νευρωνικών δικτύων .....	84
6.8	Κώδικας για αξιολόγηση νευρωνικών δικτύων για τα προβλήματα Δ.Ε. ....	86
6.9	Κώδικας για αξιολόγηση νευρωνικών δικτύων για το μοντέλο τυχαίων βηματιστών 87	



## Κατάλογος Εικόνων

Εικόνα 1.1: Αναλυτική επίλυση του προβλήματος Bratu/ Οι δύο καμπύλες που προκύπτουν για $\lambda=1$ και η μοναδική λύση για $\lambda=\lambda_c$ .....	18
Εικόνα 1.2: Παραμετρική ανάλυση του μονοδιάστατου προβλήματος Bratu για διερεύνηση της ευστάθειας συναρτήσεως του συντελεστή $\lambda$ .....	18
Εικόνα 1.3: Πείραμα "Βελόνα του Buffon".....	19
Εικόνα 1.4: Μοντέλο ενός νευρώνα.....	22
Εικόνα 1.5: Συνάρτηση ενεργοποίησης hardlim.....	22
Εικόνα 1.6: Συνάρτηση ενεργοποίησης purelin.....	23
Εικόνα 1.7: Συνάρτηση ενεργοποίησης logsig.....	23
Εικόνα 1.8: Συνάρτηση ενεργοποίησης tansig.....	23
Εικόνα 1.9: Τυπική δομή νευρωνικού δικτύου προώθησης 3 στρωμάτων.....	24
Εικόνα 1.10: Συνάρτηση με πολλά τοπικά ελάχιστα και saddle points.....	26
Εικόνα 1.11: Εικόνα 1.11-α: Επιλογή 4 τυχαίων συναρτήσεων από το αρχικό σετ με μέση τιμή 0. Εικόνα 1.11-β: Μετά από την εισαγωγή 2 παρατηρήσεων επιλέγονται συναρτήσεις που περνάνε από τα σημεία. Οι συναρτήσεις φαίνεται με τις διακεκομμένες γραμμές και η μέση πρόβλεψη (μέση τιμή των συναρτήσεων) με συνεχή γραμμή. Η σκιασμένη περιοχή αντιστοιχεί στο διπλάσιο της τυπικής απόκλισης σε κάθε $x$ .....	27
Εικόνα 2.1: Αρχικές κατανομές για το πείραμα της μοριακής διάχυσης.....	30
Εικόνα 2.2: Εικόνα 2.2: Αρχικές κατανομές για το πείραμα των τυχαίων βηματιστών. Εικόνα 2.1-A: Αρχικές κατανομές 1 έως 4, Εικόνα 2.1-B: Αρχικές κατανομές 5 έως 8.....	37
Εικόνα 2.3: Εικόνα 2.3: Διεπιφάνεια επαφής και διαγράμματα εκπαίδευσης νευρωνικών δικτύων. 2.3-A (αριστερά): Διεπιφάνεια επαφής νευρωνικού δικτύου, Εικόνα 2.3-A (δεξιά): Διαγράμματα παλινδρόμησης στα δεδομένα εκπαίδευσης, επαλήθευσης, τεστ και συνολικά, Εικόνα 2.3-B (κάτω): Διάγραμμα σφάλματος κατά την εκπαίδευση για τα 3 σετ δεδομένων...42	
Εικόνα 3-1: Αποτελέσματα επίλυσης της Δ.Ε. μοριακής διάχυσης για χρόνο $t=100$ , με συντελεστή διάχυσης $D=0,1$ για κάθε αρχική κατανομή.....	43
Εικόνα 3-2: Πρώτη χωρική παράγωγος ως προς τη διεύθυνση $x$ της λύσης της Δ.Ε. μοριακής διάχυσης για χρόνο $t=100$ , με συντελεστή διάχυσης $D=0,1$ .....	43
Εικόνα 3-3: Δεύτερη χωρική παράγωγος ως προς τη διεύθυνση $x$ της λύσης της Δ.Ε. μοριακής διάχυσης για χρόνο $t=100$ , με συντελεστή διάχυσης $D=0,1$ .....	44
Εικόνα 3-4: Χρονική παράγωγος της λύσης της Δ.Ε. μοριακής διάχυσης για χρόνο $t=100$ , με συντελεστή διάχυσης $D=0,1$ .....	44
Εικόνα 3-5: Τα δεδομένα που χρησιμοποιήθηκαν από την λύση του προβλήματος μοριακής διάχυσης και τις παραγώγους του για την εφαρμογή Γκαουσιανών Διεργασιών.....	45
Εικόνα 3-6: Λογάριθμοι των τιμών των παραμέτρων Kernel για κάθε μεταβλητή για το πρόβλημα μοριακής διάχυσης.....	45
Εικόνα 3-7: Δεδομένα εκπαίδευσης νευρωνικού δικτύου για την επίλυση του προβλήματος της μοριακής διάχυσης.....	46

Εικόνα 3-8: Προσαρμογή βέλτιστου νευρωνικού δικτύου στην 7η αρχική κατανομή σε χρόνους $t=100,1000$ για συντελεστή διάχυσης $D=0,1$ . Με μπλε διακεκομμένες γραμμές απεικονίζονται οι αρχικές κατανομές. Με μαύρες συνεχείς γραμμές απεικονίζονται οι λύσεις που υπολογίζονται σε κάθε χρόνο. Με κόκκινους κύκλους απεικονίζονται οι τιμές που υπολογίζονται από το νευρωνικό δίκτυο σε κάθε κόμβο.....	48
Εικόνα 3-9: Προσαρμογή βέλτιστου νευρωνικού δικτύου στην 8 η αρχική κατανομή σε χρόνους $t=100,1000$ για συντελεστή διάχυσης $D=0,1$ . Με μπλε διακεκομμένες γραμμές απεικονίζονται οι αρχικές κατανομές. Με μαύρες συνεχείς γραμμές απεικονίζονται οι λύσεις που υπολογίζονται σε κάθε χρόνο. Με κόκκινους κύκλους απεικονίζονται οι τιμές που υπολογίζονται από το νευρωνικό δίκτυο σε κάθε κόμβο.....	48
Εικόνα 3-10: Αποτελέσματα επίλυσης του προβλήματος Bratu για χρόνο $t=1$ , με συντελεστή $D=1, \lambda=1$ .....	49
Εικόνα 3-11: Πρώτη χωρική παράγωγος ως προς τη διεύθυνση $x$ της λύσης του προβλήματος Bratu για χρόνο $t=1$ , με συντελεστή διάχυσης $D=1, \lambda=1$ .....	49
Εικόνα 3-12: Δεύτερη χωρική παράγωγος ως προς τη διεύθυνση $x$ της λύσης του προβλήματος Bratu για χρόνο $t=1$ , με συντελεστή διάχυσης $D=1, \lambda=1$ .....	49
Εικόνα 3-13: Χρονική παράγωγος της λύσης του προβλήματος Bratu για χρόνο $t=1$ , με συντελεστή διάχυσης $D=1, \lambda=1$ .....	50
Εικόνα 3-14: Τα δεδομένα που χρησιμοποιήθηκαν από την λύση του προβλήματος Bratu και τις παραγώγους του για την εφαρμογή Γκαουσιανών Διεργασιών.....	50
Εικόνα 3-15: Λογάριθμοι των τιμών των παραμέτρων Kernel για κάθε μεταβλητή για το πρόβλημα Bratu.....	51
Εικόνα 3-16: Δεδομένα εκπαίδευσης νευρωνικού δικτύου για την επίλυση του προβλήματος Bratu.....	51
Εικόνα 3-17: Προσαρμογή βέλτιστου νευρωνικού δικτύου στην 7η αρχική κατανομή σε χρόνους $t=0.1, 1, 10$ για συντελεστή $D=1, \lambda=1$ . Με μπλε διακεκομμένες γραμμές απεικονίζονται οι αρχικές κατανομές. Με μαύρες συνεχείς γραμμές απεικονίζονται οι λύσεις που υπολογίζονται σε κάθε χρόνο. Με κόκκινους κύκλους απεικονίζονται οι τιμές που υπολογίζονται από το νευρωνικό δίκτυο σε κάθε κόμβο.....	53
Εικόνα 3-18: Προσαρμογή βέλτιστου νευρωνικού δικτύου στην 7η αρχική κατανομή σε χρόνους $t=0.1, 1, 10$ για συντελεστή $D=1, \lambda=1$ . Με μπλε διακεκομμένες γραμμές απεικονίζονται οι αρχικές κατανομές. Με μαύρες συνεχείς γραμμές απεικονίζονται οι λύσεις που υπολογίζονται σε κάθε χρόνο. Με κόκκινους κύκλους απεικονίζονται οι τιμές που υπολογίζονται από το νευρωνικό δίκτυο σε κάθε κόμβο.....	53
Εικόνα 3.19: Αντιπαραβολή των δεδομένων επίλυσης του μοντέλου τυχαίων βηματιστών με το μοντέλο της μοριακής διάχυσης. Επίλυσης για την 1η αρχική κατανομή/1 Source at center σε χρόνο $t=0.5, D=10$ . Με μπλε συνεχόμενη γραμμή παρουσιάζονται τα δεδομένα του μοντέλου της διάχυσης ενώ με κίτρινη των τυχαίων βηματιστών.....	54
Εικόνα 3.20: Αντιπαραβολή των δεδομένων επίλυσης του μοντέλου τυχαίων βηματιστών με το μοντέλο της μοριακής διάχυσης. Επίλυσης για την 2η αρχική κατανομή/Saw σε χρόνο $t=0.5, D=10$ . Με μπλε συνεχόμενη γραμμή παρουσιάζονται τα δεδομένα του μοντέλου της διάχυσης ενώ με κίτρινη των τυχαίων βηματιστών.....	55

Εικόνα 3.21 Αντιπαραβολή των δεδομένων επίλυσης του μοντέλου τυχαίων βηματιστών με το μοντέλο της μοριακής διάχυσης. Επίλυσης για την 3η αρχική κατανομή/Steps σε χρόνο $t=0.5$ , $D=10$ . Με μπλε συνεχόμενη γραμμή παρουσιάζονται τα δεδομένα του μοντέλου της διάχυσης.....	55
Εικόνα 3.22: Αντιπαραβολή των δεδομένων επίλυσης του μοντέλου τυχαίων βηματιστών με το μοντέλο της μοριακής διάχυσης. Επίλυσης για την 4η αρχική κατανομή/1 Larger Source at center σε χρόνο $t=0.5$ , $D=10$ . Με μπλε συνεχόμενη γραμμή παρουσιάζονται τα δεδομένα του μοντέλου της διάχυσης ενώ με κίτρινη των τυχαίων βηματιστών.....	56
Εικόνα 3.23: Αντιπαραβολή των δεδομένων επίλυσης του μοντέλου τυχαίων βηματιστών με το μοντέλο της μοριακής διάχυσης. Επίλυσης για την 5η αρχική κατανομή/Normal Distribution σε χρόνο $t=0.5$ , $D=10$ . Με μπλε συνεχόμενη γραμμή παρουσιάζονται τα δεδομένα του μοντέλου της διάχυσης ενώ με κίτρινη των τυχαίων βηματιστών.....	56
Εικόνα 3.24: Αντιπαραβολή των δεδομένων επίλυσης του μοντέλου τυχαίων βηματιστών με το μοντέλο της μοριακής διάχυσης. Επίλυσης για την 6η αρχική κατανομή/Pyramid σε χρόνο $t=0.5$ , $D=10$ . Με μπλε συνεχόμενη γραμμή παρουσιάζονται τα δεδομένα του μοντέλου της διάχυσης ενώ με κίτρινη των τυχαίων βηματιστών.....	57
Εικόνα 3.25: Οι CDF των θέσεων των βηματιστών για κάθε αρχική κατανομή σε χρόνο $t=0.5$ για πληθυσμό $10^4$ βηματιστών, 10 επαναλήψεις χωρίς φιλτράρισμα θορύβου.....	58
Εικόνα 3.26: Δεδομένα για την δεύτερη χωρική παράγωγο της CDF των θέσεων των βηματιστών ως προς τη διεύθυνση $\chi$ (Εικόνα 3.26-A) και για τη χρονική παράγωγο της CDF (Εικόνα 3.26-B) για κάθε αρχική κατανομή σε χρόνο $t=0.5$ για πληθυσμό $10^4$ βηματιστών, 10 επαναλήψεις χωρίς φιλτράρισμα θορύβου.....	58
Εικόνα 3.27: Οι CDF των θέσεων των βηματιστών για κάθε αρχική κατανομή σε χρόνο $t=0.5$ για πληθυσμό $10^4$ βηματιστών, 10 επαναλήψεις με φιλτράρισμα θορύβου.....	59
Εικόνα 3.28: Δεδομένα για την δεύτερη χωρική παράγωγο της CDF των θέσεων των βηματιστών ως προς τη διεύθυνση $\chi$ (Εικόνα 3.28-A) και για τη χρονική παράγωγο της CDF (Εικόνα 3.28-B) για κάθε αρχική κατανομή σε χρόνο $t=0.5$ για πληθυσμό $10^4$ βηματιστών, 10 επαναλήψεις με φιλτράρισμα θορύβου.....	59
Εικόνα 3.29: Οι CDF των θέσεων των βηματιστών για κάθε αρχική κατανομή σε χρόνο $t=0.5$ για πληθυσμό $10^4$ βηματιστών, 100 επαναλήψεις χωρίς φιλτράρισμα θορύβου.....	60
Εικόνα 3.30: Δεδομένα για την δεύτερη χωρική παράγωγο της CDF των θέσεων των βηματιστών ως προς τη διεύθυνση $\chi$ (Εικόνα 3.30-A) και για τη χρονική παράγωγο της CDF (Εικόνα 3.30-B) για κάθε αρχική κατανομή σε χρόνο $t=0.5$ για πληθυσμό $10^4$ βηματιστών, 100 επαναλήψεις χωρίς φιλτράρισμα θορύβου.....	60
Εικόνα 3.31: Οι CDF των θέσεων των βηματιστών για κάθε αρχική κατανομή σε χρόνο $t=0.5$ για πληθυσμό $10^4$ βηματιστών, 100 επαναλήψεις με φιλτράρισμα θορύβου.....	61
Εικόνα 3.32: Δεδομένα για την δεύτερη χωρική παράγωγο της CDF των θέσεων των βηματιστών ως προς τη διεύθυνση $\chi$ (Εικόνα 3.32-A) και για τη χρονική παράγωγο της CDF (Εικόνα 3.32-B) για κάθε αρχική κατανομή σε χρόνο $t=0.5$ για πληθυσμό $10^4$ βηματιστών, 100 επαναλήψεις με φιλτράρισμα θορύβου.....	61
Εικόνα 3.33: Οι CDF των θέσεων των βηματιστών για κάθε αρχική κατανομή σε χρόνο $t=0.5$ για πληθυσμό $10^5$ βηματιστών, 10 επαναλήψεις χωρίς φιλτράρισμα θορύβου.....	62

Εικόνα 3.34: Δεδομένα για την δεύτερη χωρική παράγωγο της CDF των θέσεων των βηματιστών ως προς τη διεύθυνση $\chi$ (Εικόνα 3.34-A) και για τη χρονική παράγωγο της CDF (Εικόνα 3.34-B) για κάθε αρχική κατανομή σε χρόνο $t=0.5$ για πληθυσμό $10^5$ βηματιστών, 10 επαναλήψεις χωρίς φιλτράρισμα θορύβου.....	62
Εικόνα 3.35: Οι CDF των θέσεων των βηματιστών για κάθε αρχική κατανομή σε χρόνο $t=0.5$ για πληθυσμό $10^5$ βηματιστών, 10 επαναλήψεις με φιλτράρισμα θορύβου.....	63
Εικόνα 3.36: Δεδομένα για την δεύτερη χωρική παράγωγο της CDF των θέσεων των βηματιστών ως προς τη διεύθυνση $\chi$ (Εικόνα 3.36-A) και για τη χρονική παράγωγο της CDF (Εικόνα 3.36-B) για κάθε αρχική κατανομή σε χρόνο $t=0.5$ για πληθυσμό $10^5$ βηματιστών, 10 επαναλήψεις με φιλτράρισμα θορύβου.....	63
Εικόνα 3.37: Οι CDF των θέσεων των βηματιστών για κάθε αρχική κατανομή σε χρόνο $t=0.5$ για πληθυσμό $10^5$ βηματιστών, 100 επαναλήψεις χωρίς φιλτράρισμα θορύβου.....	64
Εικόνα 3.38: Δεδομένα για την δεύτερη χωρική παράγωγο της CDF των θέσεων των βηματιστών ως προς τη διεύθυνση $\chi$ (Εικόνα 3.38-A) και για τη χρονική παράγωγο της CDF (Εικόνα 3.38-B) για κάθε αρχική κατανομή σε χρόνο $t=0.5$ για πληθυσμό $10^5$ βηματιστών, 100 επαναλήψεις χωρίς φιλτράρισμα θορύβου.....	64
Εικόνα 3.39: Οι CDF των θέσεων των βηματιστών για κάθε αρχική κατανομή σε χρόνο $t=0.5$ για πληθυσμό $10^5$ βηματιστών, 100 επαναλήψεις με φιλτράρισμα θορύβου.....	65
Εικόνα 3.40: Δεδομένα για την δεύτερη χωρική παράγωγο της CDF των θέσεων των βηματιστών ως προς τη διεύθυνση $\chi$ (Εικόνα 3.40-A) και για τη χρονική παράγωγο της CDF (Εικόνα 3.40-B) για κάθε αρχική κατανομή σε χρόνο $t=0.5$ για πληθυσμό $10^5$ βηματιστών, 100 επαναλήψεις με φιλτράρισμα θορύβου.....	65
Εικόνα 3.41: Τα δεδομένα που χρησιμοποιήθηκαν από την λύση του μοντέλου τυχαίων βηματιστών και τις παραγώγους του για την εφαρμογή Γκαουσιανών Διεργασιών.....	66
Εικόνα 3.42: Λογάριθμοι των τιμών των παραμέτρων Kernel για κάθε μεταβλητή για το μοντέλο τυχαίων βηματιστών.....	67
Εικόνα 3.43: Δεδομένα εκπαίδευσης νευρωνικού δικτύου για την επίλυση του μοντέλου των τυχαίων βηματιστών.....	67
Εικόνα 3-43: Προσαρμογή βέλτιστου νευρωνικού δικτύου στην 5 η αρχική κατανομή σε χρόνους $t=0.1, 0.5$ . Με μπλε διακεκομμένες γραμμές απεικονίζονται οι αρχικές κατανομές. Με μαύρες συνεχείς γραμμές απεικονίζονται οι λύσεις που υπολογίζονται σε κάθε χρόνο. Με κόκκινους κύκλους απεικονίζονται οι τιμές που υπολογίζονται από το νευρωνικό δίκτυο σε κάθε κόμβο.....	69
Εικόνα 3-44: Προσαρμογή βέλτιστου νευρωνικού δικτύου στην 7η αρχική κατανομή σε χρόνους $t=0.1, 0.5$ . Με μπλε διακεκομμένες γραμμές απεικονίζονται οι αρχικές κατανομές. Με μαύρες συνεχείς γραμμές απεικονίζονται οι λύσεις που υπολογίζονται σε κάθε χρόνο. Με κόκκινους κύκλους απεικονίζονται οι τιμές που υπολογίζονται από το νευρωνικό δίκτυο σε κάθε κόμβο.....	69
Εικόνα 4.1: Πληθυσμός κυττάρων μετά από 1 ώρα και 300 επαναλήψεις.....	71
Εικόνα 4.2: Πληθυσμός κυττάρων μετά από 6 ώρες και 668 επαναλήψεις.....	72

## Κατάλογος Πινάκων

Πίνακας 2.1: Δεδομένα εκπαίδευσης νευρωνικών δικτύων για το πρόβλημα της μοριακής διάχυσης.....	33
Πίνακας 2.2: Δομές νευρωνικών δικτύων που εκπαιδεύτηκαν και εξετάστηκαν για την εύρεση της βέλτιστης αρχιτεκτονικής για το πρόβλημα της μοριακής διάχυσης.....	33
Πίνακας 2.3: Δεδομένα εκπαίδευσης νευρωνικών δικτύων για το πρόβλημα Bratu.....	35
Πίνακας 2.4: Δομές νευρωνικών δικτύων που εκπαιδεύτηκαν και εξετάστηκαν για την εύρεση της βέλτιστης αρχιτεκτονικής για το πρόβλημα Bratu .....	36
Πίνακας 2.5: Δεδομένα εκπαίδευσης νευρωνικών δικτύων για το πρόβλημα των τυχαίων βηματιστών.....	39
Πίνακας 2.6: Δομές νευρωνικών δικτύων που εκπαιδεύτηκαν και εξετάστηκαν για την εύρεση της βέλτιστης αρχιτεκτονικής για το πρόβλημα των τυχαίων βηματιστών.....	39
Πίνακας 3.1: Δεδομένα αξιολόγησης νευρωνικών δικτύων στο πρόβλημα της μοριακής διάχυσης/ Δείκτης AUF.....	46
Πίνακας 3.2: Δεδομένα αξιολόγησης νευρωνικών δικτύων στο πρόβλημα της μοριακής διάχυσης/ Δείκτης MSE.....	47
Πίνακας 3.3: Δεδομένα αξιολόγησης νευρωνικών δικτύων στο πρόβλημα Bratu/ Δείκτης AUF.....	52
Πίνακας 3.4: Δεδομένα αξιολόγησης νευρωνικών δικτύων στο πρόβλημα Bratu/ Δείκτης MSE.....	52
Πίνακας 3.5: Χρόνος ολοκλήρωσης υπολογισμών επίλυσης μοντέλου τυχαίων βηματιστών για κάθε συνδιασμό πληθυσμού βηματιστών και επαναλήψεων.....	65
Πίνακας 3.6: Δεδομένα αξιολόγησης νευρωνικών δικτύων στο μοντέλο τυχαίων βηματιστών/ Δείκτης AUF.....	68
Πίνακας 3.7: Δεδομένα αξιολόγησης νευρωνικών δικτύων στο μοντέλο τυχαίων βηματιστών/ Δείκτης MAPE.....	68

# 1 Κεφάλαιο 1<sup>ο</sup>: Εισαγωγή

Στο πρώτο κεφάλαιο αναπτύσσεται η θεωρία πάνω στην οποία βασίζεται η παρούσα διπλωματική. Αρχικά θεμελιώνονται οι διαφορικές εξισώσεις και οι μέθοδοι επίλυσης τους. Έπειτα στοιχειοθετείται το πρόβλημα Bratu ως παράδειγμα διαφορικής εξίσωσης. Μετά παρουσιάζεται η βάση των μοντέλων Monte Carlo, πάνω στα οποίες βασίζεται ο προσομοιωτής μικροκλίμακας των τυχαίων βηματιστών. Τέλος παρουσιάζεται μια μελέτη μεθόδων μηχανικής μάθησης και επεξηγούνται τα εργαλεία που χρησιμοποιούνται στην ανάλυση των παραπάνω μοντέλων, οι Γκαουσιανές Διεργασίες και τα νευρωνικά δίκτυα.

## 1.1 Διαφορικές Εξισώσεις

Κατά την μελέτη της συμπεριφοράς του φυσικού κόσμου και των φαινομένων που τον διέπουν πολύ συχνά προκύπτουν μαθηματικές σχέσεις οι οποίες συνδέουν το φαινόμενο υπό μελέτη, όχι μόνο με το περιβάλλον του, αλλά και με κάποιο ή κάποιους ρυθμούς μεταβολής. Τέτοιες σχέσεις, οι οποίες περιέχουν παραγώγους μεγεθών, ονομάζονται διαφορικές εξισώσεις και όταν χρησιμοποιούνται για την περιγραφή κάποιου φυσικού φαινομένου ονομάζονται μαθηματικό μοντέλο αυτού. [1]

Στον τομέα της Χημικής Μηχανικής οι διαφορικές εξισώσεις αποτελούν ένα εξαιρετικά σημαντικό εργαλείο στην περιγραφή, ανάλυση και μοντελοποίηση των φαινομένων που μελετώνται. Παραδείγματος χάριν διαφορικές εξισώσεις αποτελούν τα ισοζύγια ορμής, μάζας και ενέργειας, τα οποία αποτελούν ακρογωνιαίο λίθο στην επιστήμη της χημικής μηχανικής.

### 1.1.1 Ταξινόμηση Διαφορικών Εξισώσεων

Ως διαφορική εξίσωση ορίζεται κάθε εξίσωση που συσχετίζει μια σχέση με την χρονική μεταβολή της. Υπάρχουν πολλοί τρόποι να ταξινομηθούν αυτές οι εξισώσεις, παρακάτω παρουσιάζονται οι κυριότεροι:

- **Συνήθεις και μερικές διαφορικές εξισώσεις**

Αυτή η μέθοδος ταξινόμησης ελέγχει τις ανεξάρτητες μεταβλητές από τις οποίες εξαρτάται η άγνωστη συνάρτηση.

Εάν εξαρτάται από μόνο μία μεταβλητή τότε η εξίσωση περιέχει μόνο συνήθεις παραγώγους και αποκαλείται Συνήθης Διαφορική Εξίσωση (Σ.Δ.Ε). Τυπικό παράδειγμα Σ.Δ.Ε. αποτελεί η εξίσωση που περιγράφει την μεταβολή της συγκέντρωσης ενός αντιδρώντος που συμμετέχει σε μονόδρομη στοιχειώδη αντίδραση 1<sup>ης</sup> τάξης[2]:

$$\frac{dC}{dt} = -kC, \quad (1.1)$$

όπου C: συγκέντρωση αντιδρώντος, k: σταθερά της αντίδρασης

Εάν η συνάρτηση εξαρτάται από περισσότερες από μία, ανεξάρτητες μεταξύ τους, μεταβλητές, τότε αποκαλείται Μερική Διαφορική Εξίσωση. Τυπικό παράδειγμα Μ.Δ.Ε. αποτελεί η εξίσωση της διάχυσης[3]:

$$\frac{\partial u}{\partial t} = D\nabla^2 u. \quad (1.2)$$

Όπου u: η συγκέντρωση, D: ο συντελεστής διάχυσης,  $\nabla^2$ : ο τελεστής Laplace.

Εδώ στην έκφραση της εξίσωσης υπάρχουν μερικά διαφορικά καθώς η συνάρτηση της συγκέντρωσης εξαρτάται όχι μόνο από τον χρόνο αλλά και από τις χωρικές διαστάσεις. Σε μία διάσταση η εξίσωση (1.2) γράφεται:

$$\frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial x^2} \quad (1.3)$$

- **Συστήματα Διαφορικών Εξισώσεων**

Αυτή η μέθοδος ταξινόμησης ελέγχει το πλήθος των άγνωστων εξισώσεων που πρέπει να προσδιοριστούν.

Ως τώρα έχουν παρουσιαστεί παραδείγματα με μόνο μία άγνωστη εξίσωση. Συχνά όμως το φαινόμενο υπό μελέτη εξαρτάται από πολλούς διαφορετικούς παράγοντες. Για παράδειγμα ένα σύστημα αντιδράσεων εντός αντιδραστήρα απαιτεί την ταυτόχρονη επίλυση τόσων διαφορικών εξισώσεων όσα και τα αντιδρώντα που συμμετέχουν στις αντιδράσεις.

- **Τάξη**

Αυτή η μέθοδος ελέγχει την τάξη της ανώτερης παραγώγου που εμφανίζεται στην εξίσωση. Με αυτή την ταξινόμηση η εξίσωση (1.1) είναι μια Δ.Ε. 1<sup>ης</sup> τάξης ενώ η εξίσωση (1.3) είναι μια Δ.Ε. 2<sup>ης</sup> τάξης.

- **Γραμμικότητα**

Μια διαφορική εξίσωση καλείται γραμμική όταν είναι μια γραμμική συνάρτηση των παραγώγων της άγνωστης συνάρτησης και δεν περιέχει μη γραμμικές συναρτήσεις. Σύμφωνα με αυτή την ταξινόμηση όλες οι εξισώσεις που έχουν παρουσιαστεί ως τώρα αποτελούν γραμμικές Δ.Ε. Αν όμως εξεταστεί η εξίσωση που περιγράφει τη γωνία  $\theta$  που διαγράφει ένα ταλαντούμενο εκκρεμές με την κατακόρυφη διεύθυνση συναρτήσει του χρόνου[4]:

$$\frac{d^2\theta}{dt^2} + \frac{g}{L} \sin \theta = 0, \quad (1.4)$$

όπου  $\theta$ : η γωνία,  $g$ : η επιτάχυνση της βαρύτητας,  $L$ : το μήκος του εκκρεμούς [1]

είναι εμφανές πως είναι μια μη γραμμική Δ.Ε.

### 1.1.2 Επίλυση Διαφορικών Εξισώσεων

Ανάλογα την φύση και την ταξινόμηση της εκάστοτε διαφορικής εξίσωσης έχουν αναπτυχθεί διάφορες μέθοδοι επίλυσης και εύρεσης της αναλυτικής λύσης. Μερικές από τις κυριότερες μεθόδους παρουσιάζονται ενδεικτικά παρακάτω:

- **Δ.Ε 1ης τάξης:** Μέθοδος ολοκληρώνοντα παράγοντα
- **Δ.Ε. 2ης τάξης (γραμμικές):** Μέθοδος προσδιοριστέων συντελεστών/ Μέθοδος δυναμοσειρών
- **Μερικές Διαφορικές εξισώσεις:** Χωρισμός μεταβλητών

Με τις παραπάνω μεθόδους είναι δυνατή η επίλυση διαφορικών εξισώσεων για τον προσδιορισμό της άγνωστης συνάρτησης. Πρέπει όμως να τονιστεί πως το αποτέλεσμα της επίλυσης αποτελεί μία οικογένεια συναρτήσεων οι οποίες ικανοποιούν την Δ.Ε. και δίνονται συναρτήσει κάποιων σταθερών. Για παράδειγμα, με επίλυση της εξίσωσης (1.3) προκύπτει[2]:

$$C(t) = C_0 e^{-kt}. \quad (1.5)$$

Είναι εμφανές ότι για τον ακριβή προσδιορισμό της συνάρτησης  $C=f(t)$  είναι απαραίτητο να προσδιοριστεί η σταθερά  $C_0$ . Τα προβλήματα που περιλαμβάνουν και τον προσδιορισμό αυτών των σταθερών ονομάζονται Προβλήματα Αρχικών Τιμών ή Π.Α.Τ.

### 1.1.3 Προβλήματα Αρχικών και Συνοριακών τιμών

Τα Π.Α.Τ. εστιάζουν, πέρα από την επίλυση της Δ.Ε., στον προσδιορισμό της ακριβούς λύσης η οποία περιγράφει το υπό μελέτη σύστημα. Αυτό είναι εφικτό με την χρήση κάποιων αρχικών τιμών και συνοριακών εξισώσεων, δεδομένων τα οποία περιγράφουν το συγκεκριμένο σύστημα.

Οι αρχικές τιμές μπορεί να είναι δεδομένα όπως η τιμή της συγκέντρωσης ενός αντιδρώντος σε αντιδραστήρα σε χρόνο μηδέν ή η τιμή της θερμοκρασίας μιας μεταλλικής ράβδου σε κάθε σημείο αυτής, δηλαδή μπορεί να είναι μια τιμή ή μια συνάρτηση, που θα αποτελέσει την βάση της επίλυσης.

Οι συνοριακές συνθήκες περιγράφουν την συμπεριφορά του φαινομένου στα άκρα του συστήματος. Υπάρχουν 3 είδη συνοριακών συνθηκών[5, 6]:

#### **Συνθήκη τύπου Dirichlet**

Ορίζει μια συγκεκριμένη σταθερή τιμή για την άγνωστη συνάρτηση στα όρια του συστήματος:

$$u_{boundary} = c \quad (1.6)$$

Για παράδειγμα σε σύστημα θερμαινόμενης ράβδου ορίζεται στο άκρο που εφαρμόζεται η θέρμανση σταθερή θερμοκρασία  $T=80^{\circ}C:const$ .

#### **Συνθήκη τύπου Neumann**

Ορίζει μια συγκεκριμένη σταθερή τιμή για τον ρυθμό μεταβολής της άγνωστης συνάρτησης στα όρια του συστήματος:

$$\vec{n} \cdot \nabla u|_{boundary} = c \quad (1.7)$$

όπου  $\vec{n}$  το κάθετο στο σύνορο διάνυσμα και  $c$  η τιμή του ρυθμού μεταβολής της άγνωστης συνάρτησης.

Για παράδειγμα σε σύστημα θερμαινόμενης ράβδου ορίζεται σε όλα τα άκρα εκτός από αυτό που εφαρμόζεται η θέρμανση μηδενική απώλεια ή ροή θερμότητας.

#### **Μικτή συνθήκη τύπου Robin**

Ορίζει μια συγκεκριμένη συνάρτηση για τον ρυθμό μεταβολής της άγνωστης συνάρτησης στα όρια του συστήματος:

$$\vec{n} \cdot \nabla u|_{boundary} = f(t, x, y, z) \quad (1.8)$$

όπου  $\vec{n}$  το κάθετο στο σύνορο διάνυσμα.

Για παράδειγμα σε σύστημα θερμαινόμενης ράβδου όπου ορίζεται συνοριακή συνθήκη συναγωγής σε κάποιο άκρο, αυτή διαμορφώνεται ως εξής[3]:

$$-k \frac{\partial T(t, x_1, y_1, z_1)}{\partial t} = h(T_{\infty} - T(t, x_1, y_1, z_1)). \quad (1.9)$$

Όπου  $k$ : η θερμική αγωγιμότητα του μέσου,  $h$ : ο συντελεστής μεταφοράς θερμότητας,  $T_{\infty}$ =η θερμοκρασία του περιβάλλοντος στη διεπιφάνεια του μέσου

#### **1.1.4 Αριθμητική Επίλυση Διαφορικών Εξισώσεων**

Έως τώρα έχει δοθεί έμφαση στις μεθόδους αναλυτικής επίλυσης Δ.Ε. Στο πως είναι δηλαδή δυνατόν δεδομένης της Δ.Ε. να ληφθεί μια ακριβής εξίσωση που να περιγράφει το υπό μελέτη φαινόμενο. Παρόλο που αυτό είναι και το ιδανικό σενάριο, πολλές φορές στον πραγματικό κόσμο είναι ανέφικτο. Είτε επειδή δεν επιλύεται μια Δ.Ε. με τις υπαρκτές μεθόδους ή η εφαρμογή τους θα ήταν εξαιρετικά περίπλοκη, πολλά προβλήματα που συναντά ένας μηχανικός σήμερα, έχουν μεν γνωστή Δ.Ε., αλλά δεν είναι δυνατόν να υπολογιστεί η ακριβής λύση. Σε αυτές τις περιπτώσεις, προκειμένου να παρακαμφθούν οι δυσκολίες της αναλυτικής επίλυσης, η Δ.Ε. μπορεί να λυθεί αριθμητικά και να δώσει μια ικανοποιητική προσέγγιση της άγνωστης συνάρτησης.[6]



Μια από τις απλούστερες μεθόδους αριθμητικής επίλυση Δ.Ε. είναι η μέθοδος των πεπερασμένων διαφορών, η οποία βασίζεται στα αναπτύγματα Taylor της άγνωστης συνάρτησης για να προσεγγίσει τις παραγώγους της. Ανάλογα με τις ανάγκες για το βήμα και την ακρίβεια μπορεί να γίνει η επιλογή ανάμεσα στα σχήματα εμπρός, πίσω και κεντρικών διαφορών (τα οποία προκύπτουν με κατάλληλο συνδυασμό των αναπτυγμάτων Taylor. Ενδεικτικά εξάγεται το σχήμα κεντρικών διαφορών για την προσέγγιση της 1<sup>ης</sup> παραγώγου[5]:

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(x) + \dots, \quad (1.10)$$

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{6}f'''(x) + \dots. \quad (1.11)$$

Με αφαίρεση της (1.11) από την (1.10) προκύπτει:

$$f'(x) = \frac{f(x+h)-f(x-h)}{2h} + O(h^2), \quad (1.12)$$

όπου  $h$ : βήμα και η έκφραση  $O(h^2)$  υποδηλώνει ότι με αυτή τη σχέση προκύπτει σφάλμα στον υπολογισμό της παραγώγου της τάξης του  $h^2$ .

Παρακάτω παρουσιάζονται οι σχέσεις για την προσέγγιση της 1<sup>ης</sup> και της 2<sup>ης</sup> παραγώγου σύμφωνα με το κάθε σχήμα[5]:

- **Εμπρός διαφορές**

$$f'(x) = \frac{f(x+h)-f(x)}{h} \quad (1.13)$$

$$f''(x) = \frac{f(x+2h)-2f(x+h)+f(x)}{h^2} \quad (1.14)$$

- **Πίσω διαφορές**

$$f'(x) = \frac{f(x)-f(x-h)}{h} \quad (1.15)$$

$$f''(x) = \frac{f(x)-2f(x-h)+f(x-2h)}{h^2} \quad (1.16)$$

- **Κεντρικές διαφορές**

$$f'(x) = \frac{f(x+h)-f(x-h)}{2h} \quad (1.17)$$

$$f''(x) = \frac{f(x+h)-2f(x)+f(x-h)}{h^2} \quad (1.18)$$

Από τα 3 σχήματα αυτό με την μεγαλύτερη ακρίβεια είναι το σχήμα κεντρικών διαφορών, με σφάλμα  $O(h^2)$ . [5]

### 1.1.5 Πρόβλημα Bratu

Το πρόβλημα Bratu διατυπώνεται ως εξής:

$$\frac{\partial u}{\partial t} = \nabla^2 u + \lambda e^u \quad (1.19)$$

όπου  $\lambda$ : θετική σταθερά του προβλήματος. [7]

Το πρόβλημα Bratu είναι μια μη γραμμική μερική διαφορική εξίσωση, της οποίας η συμπεριφορά χαρακτηρίζεται από την τιμή του συντελεστή  $\lambda$ . Για διάφορες τιμές του προκύπτουν διαφορετικές λύσεις του προβλήματος. Συγκεκριμένα, ορίζεται κρίσιμη τιμή του συντελεστή,  $\lambda_c$ , σύμφωνα με την οποία γίνεται η ανάλυση[7]:

- $\lambda = \lambda_c$

Υπάρχει μοναδική λύση του προβλήματος.

- $\lambda > \lambda_c$

Δεν υπάρχει λύση του προβλήματος.

- $\lambda < \lambda_c$

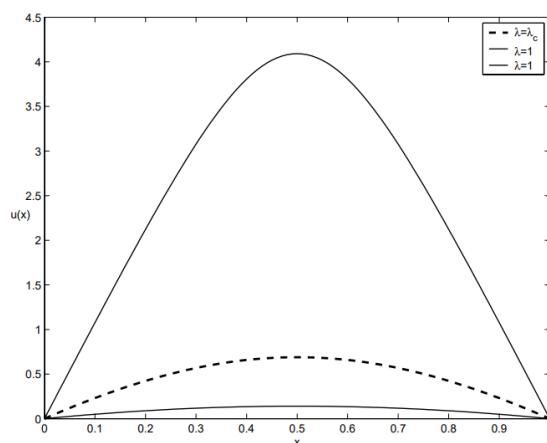
Η παραπάνω ανάλυση του προβλήματος Bratu αποδεικνύεται και από την αναλυτική επίλυση του προβλήματος σε μία διάσταση [8, 9], στο διάστημα  $L=[0,1]$ , για συνοριακές τιμές  $u(0)=u(1)=0$ . Διαγραμματική αναπαράσταση της επίλυσης παρουσιάζεται στην Εικόνα 1.1:

$$u(x) = -2 \ln \left[ \frac{\cosh\left(\left(x-\frac{1}{2}\right)\frac{\theta}{2}\right)}{\cosh\left(\frac{\theta}{4}\right)} \right] \quad (1.20)$$

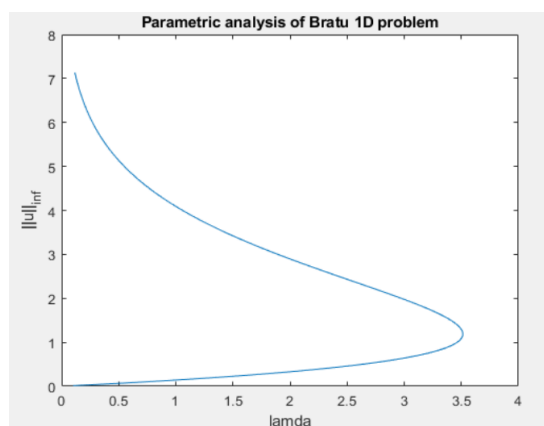
$$\theta = \sqrt{2\lambda \cosh^2\left(\frac{\theta}{4}\right)} \quad (1.21)$$

Με παραγωγή της σχέσης (1.21) και λύνοντας για την ρίζα της εξίσωσης που προκύπτει, υπολογίζεται η εξάρτηση της κρίσιμης τιμής του συντελεστή  $\lambda$ ,  $\lambda_c$ , από την σταθερά  $\theta$ :

$$\theta = \frac{1}{2} \lambda_c \cosh\left(\frac{\theta}{4}\right) \sinh\left(\frac{\theta}{4}\right). \quad (1.22)$$



Εικόνα 1.1: Αναλυτική επίλυση του προβλήματος Bratu/ Οι δύο καμπύλες που προκύπτουν για  $\lambda=1$  και η μοναδική λύση για  $\lambda=\lambda_c$  [9]



Εικόνα 1.2: Παραμετρική ανάλυση του μονοδιάστατου προβλήματος Bratu για διερεύνηση της ευστάθειας συναρτήσει του συντελεστή  $\lambda$  [10]

Υπάρχουν δύο λύσεις του προβλήματος οι οποίες εμφανίζονται ως δύο κλάδοι, ένας ευσταθής και ένας ασταθής, οι οποίοι συναντιούνται στη κρίσιμη τιμή της παραμέτρου (Εικόνα 1.2). [7, 10]

Αυτή η παραμετρική ανάλυση του προβλήματος, συναρτήσει του συντελεστή  $\lambda$ , αποτελεί έναν από τους σημαντικότερους τομείς μελέτης του συγκεκριμένου προβλήματος όμως δεν είναι ο λόγος για τον οποίο αξιοποιείται στη παρούσα εργασία. Έμφαση δίνεται στην μη γραμμικότητα του προβλήματος εξαιτίας του εκθετικού όρου και το πρόβλημα και η ανάλυση αυτού αξιοποιούνται για την αξιολόγηση της δυνατότητας εκπαίδευσης νευρωνικών δικτύων στην επίλυση μη γραμμικών Δ.Ε

## 1.2 Μέθοδοι Monte Carlo

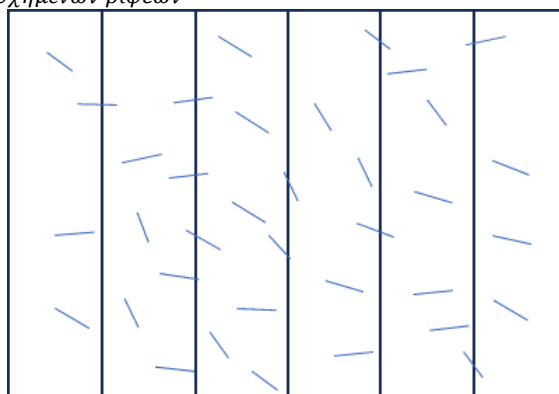
Ως Monte Carlo (Μόντε Κάρλο-MC) αναφέρονται οι μέθοδοι οι οποίες χρησιμοποιούν τυχαίους αριθμούς προκειμένου να επιλύσουν ένα πρόβλημα ή να αναπτύξουν μια αξιόπιστη προσομοίωση. Όπως περιγράφηκε από τον Halton (1970), οι μέθοδοι MC λειτουργούν με την θεωρία πως η λύση του προβλήματος που μελετάται μπορεί να προσομοιωθεί ως μια παράμετρος ενός υποθετικού πληθυσμού και έπειτα να προσεγγιστεί ικανοποιητικά λαμβάνοντας τα στατιστικά μέτρα της παραμέτρου αυτής, μέσω μιας σειράς τυχαίων αριθμών που θα προσομοιάζει δείγμα αυτού του πληθυσμού.[11]

Δύο χαρακτηριστικά προβλήματα που μπορούν να υπολογιστούν με τη χρήση μεθόδων Μόντε Κάρλο αποτελούν τα παρακάτω:

- **Η βελόνα του Buffon**

Σκοπός αυτού του πειράματος είναι η προσέγγιση της τιμής της σταθεράς  $\pi$ . Όπως το πείραμα αυτό περιγράφηκε από τον Buffon το 1777, απαιτεί τον σχεδιασμό παράλληλων ευθειών σε μια επίπεδη επιφάνεια και έπειτα την τυχαία ρίψη μικρών βελόνων στην διάταξη. Επιτυχημένη ρίψη θα θεωρείται αυτή κατά την οποία η βελόνα θα καταλήγει πάνω σε μία από τις γραμμές. Αναπαράσταση του πειράματος παρουσιάζεται στην παρακάτω εικόνα (0). Στο τέλος η τιμή του  $\pi$  μπορεί να προσεγγιστεί από τον τύπο[11, 12]:

$$\pi = 2 * \frac{\text{πλήθος συνολικών ρίψεων}}{\text{πλήθος επιτυχημένων ρίψεων}} \quad (1.23)$$



Εικόνα 1.3: Πείραμα "Βελόνα του Buffon"

- **Υπολογισμός της έκτασης μιας χώρας**

Σε αυτό το πείραμα το ζητούμενο είναι ο υπολογισμός της έκτασης μιας χώρας με ακανόνιστα σύνορα. Απαραίτητα για το πείραμα είναι ένας ορθογωνικός χάρτης που περιλαμβάνει τη χώρα και βελάκια. Τα βελάκια πρέπει να ριφθούν, τυχαία, πάνω στον χάρτη και έπειτα να υπολογιστεί ο λόγος του πλήθους τους που έπεσε μέσα στα σύνορα της χώρας πάνω στο χάρτη, προς το πλήθος των συνολικών ρίψεων. Το γινόμενο αυτού του λόγου με την συνολική επιφάνεια του χάρτη, δίνει μια εκτίμηση της έκτασης της χώρας.[12]

Αντίστοιχα το ίδιο πείραμα θα μπορούσε να γίνει τοποθετώντας πάνω από τον χάρτη ένα κανονικό δίκτυο σημείων και έπειτα να αντιμετωπιστούν αυτά όπως και τα βελάκια που αναφέρθηκαν προηγουμένως.[12]

Είναι προφανές πως η δύναμη της μεθόδου ΜΚ έγκειται στη χρήση πολύ μεγάλων στατιστικών δειγμάτων. Όσες περισσότερες βελόνες ριφθούν στην πρώτη διάταξη, όπως και όσες περισσότερα βελάκια ριφθούν επάνω στον χάρτη, με τόση μεγαλύτερη ακρίβεια θα υπολογιστούν τα επιθυμητά μεγέθη. Αυτό εκφράζεται και από το μαθηματικό υπόβαθρο πάνω στο οποίο στηρίζεται η μέθοδος ΜC, τα δύο παρακάτω θεωρήματα:

- **Νόμος των Μεγάλων Αριθμών**

Έστω  $\chi_1, \chi_2, \dots, \chi_N$  τυχαίες μεταβλητές που χαρακτηρίζονται από μέση τιμή  $E(S_n) = \mu$  και διακύμανση  $V(S_n) = \sigma^2$ . Επίσης έστω  $S_n = \sum_{k=1}^N x_k$ . Τότε ισχύει:

$$P\left(\lim_{N \rightarrow \infty} \frac{S_n}{N} = \mu\right) = 1. \quad (1.24)$$

Η παραπάνω σχέση ορίζει πως το άθροισμα των τυχαίων μεταβλητών  $S_n$  έχει κατανομή που προσεγγίζει την κανονική, για πολύ μεγάλες τιμές του N.[12, 13]

- **Κεντρικό Οριακό Θεώρημα**

Έστω  $\chi_1, \chi_2, \dots, \chi_N$  τυχαίες μεταβλητές που χαρακτηρίζονται από μέση τιμή  $E(S_n) = \mu$  και διακύμανση  $V(S_n) = \sigma^2 < \infty$ . Επίσης έστω  $S_n = \sum_{k=1}^N x_k$ . Τότε:

$$\lim_{N \rightarrow \infty} P\left(\frac{S_n - N\mu}{\sigma\sqrt{N}} \leq \chi\right) = \Phi(\chi), \quad (1.25)$$

όπου  $\Phi(\chi)$ : η συνάρτηση αθροιστικής κατανομής της κανονικής κατανομής

Η παραπάνω σχέση ορίζει πως το άθροισμα των τυχαίων μεταβλητών  $S_n$  έχει κατανομή που προσεγγίζει την κανονική, με μέσο όρο που προσεγγίζει την τιμή  $E(S_n) = N\mu$  και διακύμανση που προσεγγίζει την τιμή  $V(S_n) = N\sigma^2$ , για πολύ μεγάλες τιμές του N.[12, 13]

### 1.3 Μηχανική Μάθηση

Οι προσπάθειες της ανθρωπότητας επικεντρώνονται στην κατανόηση του κόσμου που την περιβάλλει, στην συσσώρευση γνώσης και στην βελτίωση της καθημερινής ζωής. Αυτό γίνεται μέσω παρατηρήσεων, πειραμάτων και επεξεργασίας των αποτελεσμάτων. Τα τελευταία χρόνια, με τις αλματώδεις εξελίξεις που έχουν σημειωθεί στον τομέα την πληροφορικής, έχει προστεθεί ένα εξαιρετικά δυνατό εργαλείο στην διαδικασία αυτή, ο υπολογιστής. Με την βοήθεια των υπολογιστών έχει αυξηθεί δραματικά ο όγκος των δεδομένων που μπορούν να ληφθούν και να επεξεργαστούν για ένα πείραμα, όπως επίσης έχει μειωθεί δραματικά ο απαιτούμενος χρόνος για την διεκπεραίωση πράξεων που άλλοτε θα χρειαζόταν μήνες ή και χρόνια για να ολοκληρωθούν.[14]

Ο τομέας της Μηχανικής Μάθησης αναπτύχθηκε για να εκμεταλλευτεί την ικανότητα των υπολογιστών να επιταχύνουν την διαδικασία που περιγράφηκε παραπάνω. Ιδιαίτερη έμφαση έχει δοθεί όχι μόνο στην επιτάχυνση αλλά και στην αυτοματοποίηση της παραπάνω διαδικασίας. Ο τομέας της Μηχανικής Μάθησης ασχολείται με την αυτοματοποιημένη συλλογή, ταξινόμηση και επεξεργασία προκειμένου να ληφθεί το επιθυμητό αποτέλεσμα.[15]

Η δύναμη των τεχνικών του τομέα της Μηχανικής Μάθησης έγκειται στην ικανότητά τους να εντοπίζουν και να αξιοποιούν μοτίβα και κρυμμένες σχέσεις σε μεγάλες ομάδες δεδομένων. Μερικοί τομείς που αξιοποιούνται τέτοιες τεχνικές: [16, 17]

- Αναγνώριση εικόνας [18, 19]
- Αναγνώριση κειμένου[20, 21]
- Αναγνώριση προφορικού λόγου[22]
- AI assistants[23]
- Ρομποτική[24]
- Αυτόνομα οχήματα[25]

### 1.3.1 Ταξινόμηση μεθόδων Μηχανικής Μάθησης

Σε κάθε σύστημα Μηχανικής Μάθησης παρέχεται ένα σετ δεδομένων με βάση το οποίο ρυθμίζεται. Στο σετ αυτό περιλαμβάνονται τα εξής:

- Δεδομένα εισαγωγής:  $\bar{x} = [x_1 x_2 \dots x_N]$
- Δεδομένα στόχου:  $\bar{y} = [y_1 y_2 \dots y_N]$

Παρακάτω παρουσιάζονται οι κυριότερες κατηγορίες Μηχανικής Μάθησης, με βάση τα δεδομένα που δίνονται αρχικά στο σύστημα και το αναμενόμενο αποτέλεσμα:

- **Εποπτευόμενη Μάθηση**

Κατά την εποπτευόμενη μάθηση παρέχονται τα  $\bar{x}, \bar{y}$  ως ένα σετ δεδομένων όπου κάθε στοιχείο του  $\bar{x}$  αντιστοιχίζεται σε ένα του  $\bar{y}$ . Στόχος αποτελεί το σύστημα να μπορεί να βρει μια συνάρτηση που θα πραγματοποιεί σωστά αυτή την αντιστοίχιση. Τέτοια συστήματα αξιοποιούνται σε προβλήματα ταξινόμησης και παλινδρόμησης.[16]

- **Μη εποπτευόμενη μάθηση**

Κατά την μη εποπτευόμενη μάθηση στο σύστημα παρέχονται μόνο τα δεδομένα του  $\bar{x}$  χωρίς τα αντίστοιχα  $\bar{y}$  και στόχος πλέον είναι η εύρεση της δομής που διέπει τα δεδομένα.[16]

- **Ενισχυτική μάθηση**

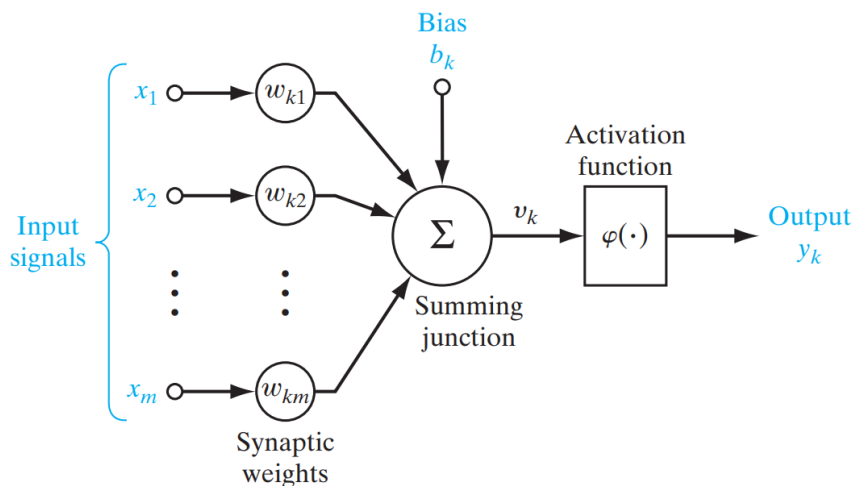
Κατά την ενισχυτική μάθηση παρέχονται στο σύστημα πληροφορίες και δεδομένα για ένα συγκεκριμένο περιβάλλον, όπου είναι δυνατόν να ληφθούν συγκεκριμένες ενέργειες με συγκεκριμένο τρόπο. Πλέον από το σύστημα δεν απαιτείται να αντιστοιχίσει κάποια δεδομένα εισαγωγής σε κάποια δεδομένα στόχου, ή να εντοπίσει κάποια δομή, αλλά ορίζεται ένας στόχος εντός του περιβάλλοντος που πρέπει να επιτευχθεί και μια συνάρτηση επιβράβευσης η οποία επιβραβεύει το σύστημα ανάλογα με το πόσο κοντά έφτασε στην επίτευξη του ορισμένου στόχου με τις ενέργειες που επέλεξε. Αποτελεί μία διαδικασία δοκιμής και σφάλματος όπου το σύστημα, μέσα από διαδοχικές δοκιμές, βρίσκει τη βέλτιστη μέθοδο επίτευξης του ορισμένου στόχου, προσπαθώντας να μεγιστοποιήσει την επιβράβευση που λαμβάνει.[26]

### 1.3.2 Τεχνητά Νευρωνικά Δίκτυα

Τα Τεχνητά Νευρωνικά Δίκτυα, ή Νευρωνικά Δίκτυα (N.N.) για απλούστευση, αποτελούν μια από τις πιο δημοφιλείς μεθόδους μηχανικής μάθησης. Βασίζονται στο μοντέλο του εγκεφάλου και ο σκοπός είναι η προσομοίωση της λειτουργίας του. Ένα N.N. αποτελείται από πολλούς νευρώνες, διατεταγμένους και συνδεδεμένους ο ένας με τον άλλο, όπως ο εγκέφαλος.[27]

#### 1.3.2.1 Μοντέλο ενός νευρώνα

Οι νευρώνες είναι οι δομικοί λίθοι ενός νευρωνικού δικτύου. Επεξεργάζονται την εισερχόμενη πληροφορία και ανάλογα με το πως είναι διασυνδεδεμένοι παράγουν το τελικό σήμα. Το μοντέλο του ενός νευρώνα παρουσιάζεται στην παρακάτω εικόνα (Εικόνα 1.4):



Εικόνα 1.4: Μοντέλο ενός νευρώνα[28]

Στο μοντέλο του ενός νευρώνα αναγνωρίζονται 3 από τα κυριότερα στοιχεία ενός νευρωνικού δικτύου[28]:

- **Οι συνάψεις**

Οι συνάψεις είναι οι εισοδοί του σήματος σε κάθε νευρώνα. Κάθε σύναψη χαρακτηρίζεται από ένα βάρος (weight) με το οποίο πολλαπλασιάζεται το σήμα πριν φτάσει στον νευρώνα. Αυτά τα βάρη είναι που ρυθμίζονται κατά την εκπαίδευση του νευρωνικού δικτύου.

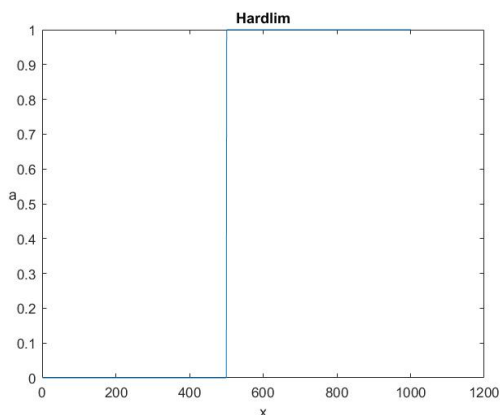
- **Στοιχείο άθροισης**

Αθροίζει τα σήματα από όλες τις συνάψεις και προσθέτει τη βάση (bias) του νευρώνα.

- **Η συνάρτηση ενεργοποίησης**

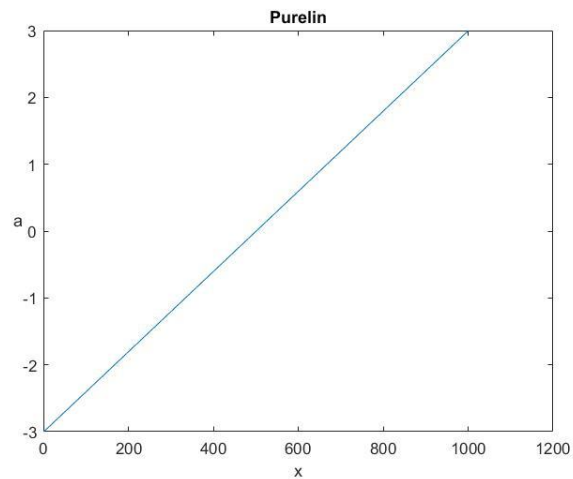
Είναι το στοιχείο που ρυθμίζει την απόκριση του νευρώνα (activation function). Λαμβάνει το σήμα από το στοιχείο άθροισης και ανάλογα την συνάρτηση απόκρισης που έχει επιλεγεί κανονικοποιεί το σήμα ή λειτουργεί σαν διακόπτης. Στη περίπτωση που λειτουργεί ως διακόπτης παίρνει τιμές 0 και 1, ανάλογα με το αν το σήμα από το στοιχείο άθροισης είναι ίσο ή μεγαλύτερο από μια τιμή-κατώφλι. Οι συνηθέστερες συναρτήσεις ενεργοποίησης παρουσιάζονται παρακάτω[29]:

➤ **Hardlim:**  $a = \begin{cases} 0, & n < 0 \\ 1, & n \geq 0 \end{cases}$  (1.26)



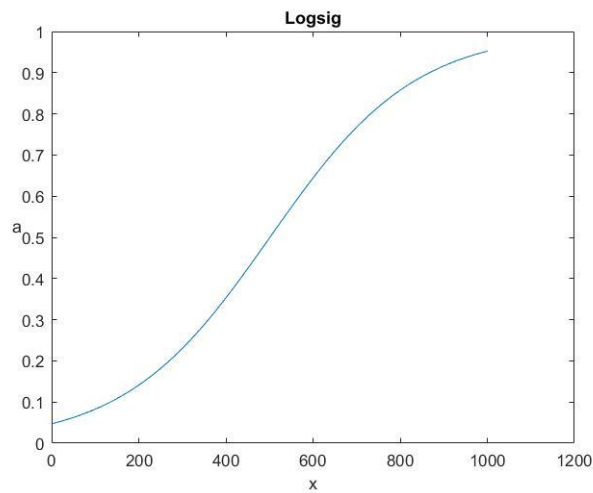
Εικόνα 1.5: Συνάρτηση ενεργοποίησης hardlim

➤ **Purelin:**  $a = n$  (1.27)



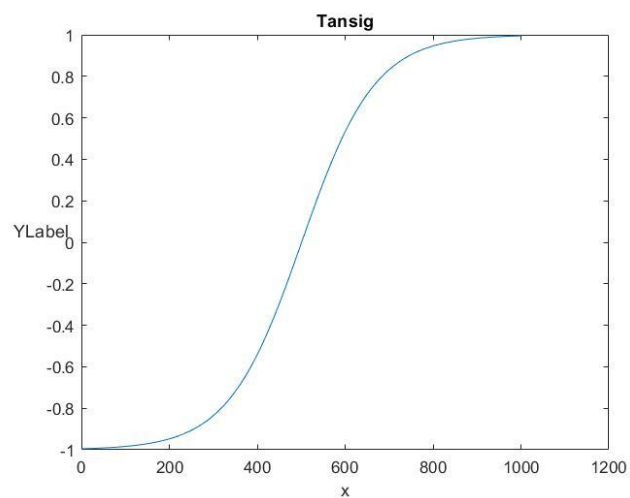
Εικόνα 1.6: Συνάρτηση ενεργοποίησης purelin

➤ **Logsig:**  $a = \frac{1}{1+e^{-n}}$  (1.28)



Εικόνα 1.7: Συνάρτηση ενεργοποίησης logsig

➤ **Tansig:**  $a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$  (1.29)



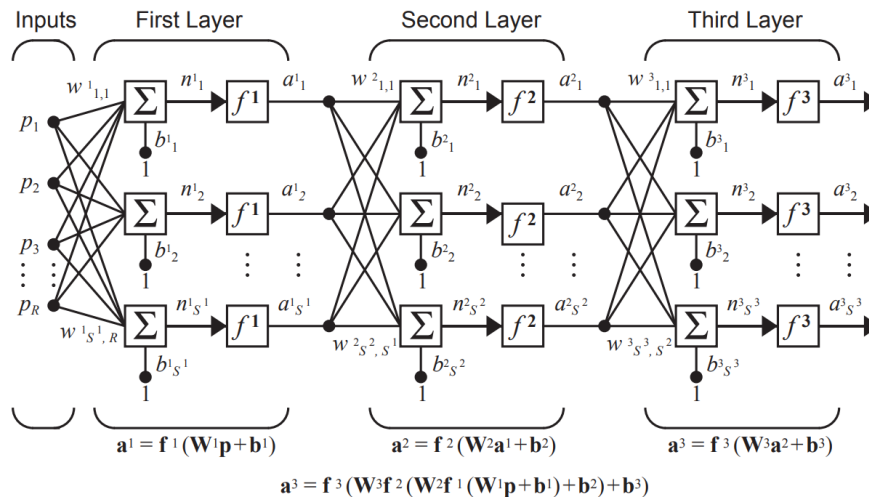
Εικόνα 1.8: Συνάρτηση ενεργοποίησης tansig

Το μαθηματικό μοντέλο που περιγράφει τον απλό νευρώνα είναι:

$$y = \varphi \left( \sum x_i w_{ki} + b_k \right). \quad (1.30)$$

### 1.3.2.2 Μοντέλο με πολλούς νευρώνες

Πιο περίπλοκα δίκτυα με περισσότερους νευρώνες έχουν και πιο περίπλοκη αρχιτεκτονική. Οι νευρώνες διατάσσονται σε στρώματα, εισόδου, εξόδου και κρυφά στα οποία οι υπολογισμοί που γίνονται είναι «κρυφοί» στον χρήστη (Εικόνα 1.9). Τέτοια δίκτυα χαρακτηρίζονται ως δίκτυα προώθησης (feedforward) καθώς η πληροφορία προχωρά σε μια συγκεκριμένη κατεύθυνση.[27]



Εικόνα 1.9: Τυπική δομή νευρωνικού δικτύου προώθησης 3 στρωμάτων[29]

### 1.3.3.3 Εκπαίδευση νευρωνικών δικτύων

Η εκπαίδευση των νευρωνικών δικτύων βασίζεται στην προσαρμογή των βαρών στις εισόδους κάθε κόμβου (στις συνάψεις). Ορίζεται μια συνάρτηση σφάλματος, η οποία είναι μια σχέση των βαρών, και το δίκτυο καλείται να την ελαχιστοποιήσει, χρησιμοποιώντας ένα σετ δεδομένων εκπαίδευσης ως δεδομένα αναφοράς. Η συνάρτηση σφάλματος υπολογίζει το μέσο τετραγωνικό σφάλμα ανάμεσα στην απόκριση του δικτύου και στα δεδομένα αναφοράς. Η διαδικασία αυτή είναι μια επαναληπτική διαδικασία που επαναλαμβάνεται μέχρι το σφάλμα να είναι μικρότερο από μια αποδεκτή τιμή.[30]

Μία μέθοδος που χρησιμοποιείται για την ελαχιστοποίηση της συνάρτησης σφάλματος είναι η μέθοδος της απότομης κατάβασης. Με προσαρμογή των βαρών του δικτύου αναζητείται η ελάχιστη τιμή της συνάρτησης σφάλματος. Με μια επαναληπτική διαδικασία είναι δυνατόν να βρεθεί η βέλτιστη τιμή με βάση την κλίση της συνάρτησης σφάλματος συναρτήσει των διαδοχικών προσαρμογών των βαρών.[29]

Ο αλγόριθμος backpropagation Είναι βοηθητικός αλγόριθμος για τον υπολογισμό κλίσεων (μερικών παραγώγων) της συνάρτησης σφάλματος και βασίζεται στο κανόνα της αλυσίδας. Διαδίδει το σφάλμα από το εξωτερικό στρώμα στα εσωτερικά, συσχετίζοντας το σφάλμα του στρώματος m με αυτό του m+1, για να προσδιοριστούν τα νέα βάρη. Σα δείκτη προσαρμογής χρησιμοποιεί το μέσο τετραγωνικό σφάλμα. Το μοντέλο του αλγόριθμου backpropagation:[27, 30]

Έστω p το διάνυσμα μεταβλητών εισόδου, t: το διάνυσμα μεταβλητών εξόδου  $a_i$ : το διάνυσμα των αποτελεσμάτων μετά από το i επίπεδο του νευρωνικού, W: ο πίνακας που περιλαμβάνει



τα διάφορα βάρη όλων των νευρώνων του νευρωνικού και  $b$ : το διάνυσμα των βάσεων όλων των επιπέδων του νευρωνικού.[29]

Το MSE υπολογίζεται:

$$F(x) = \sum_{q=1}^Q (t_q - a_q)^T (t_q - a_q) = \sum_{q=1}^Q e_q^T e_q = \sum_{i=1}^N (n_i)^2. \quad (1.31)$$

Πρώτα διαδίδεται το σήμα εισόδου στο νευρωνικό δίκτυο:

$$\begin{cases} a^0 = p & (1.32) \\ a^{m+1} = f^{m+1}(W^{m+1} a^m + b^{m+1}), m = 0, 1, \dots, M - 1 & (1.33) \\ a = a^M. & (1.34) \end{cases}$$

Έπειτα διαδίδεται το σφάλμα στο δίκτυο αντίστροφα:

$$s^M = -2F^M(n^M)(t - a), \quad (1.35)$$

$$s^m = -2F^m(n^m)(W^{m+1})^T s^{m+1}, m = M - 1, \dots, 2, 1. \quad (1.36)$$

Τέλος υπολογίζονται τα νέα βάρη και οι βάσεις:

$$W^m(k + 1) = W^m(k) - a s^m (a^{m-1})^T, \quad (1.37)$$

$$b^m(k + 1) = b^m(k) - a s^m. \quad (1.38)$$

Όπου  $a$ : ο ρυθμός μάθησης, παράμετρος της μεθόδου απότομης κατάβασης

#### 1.3.3.4 Προβλήματα κατά την εκπαίδευση νευρωνικών δικτύων

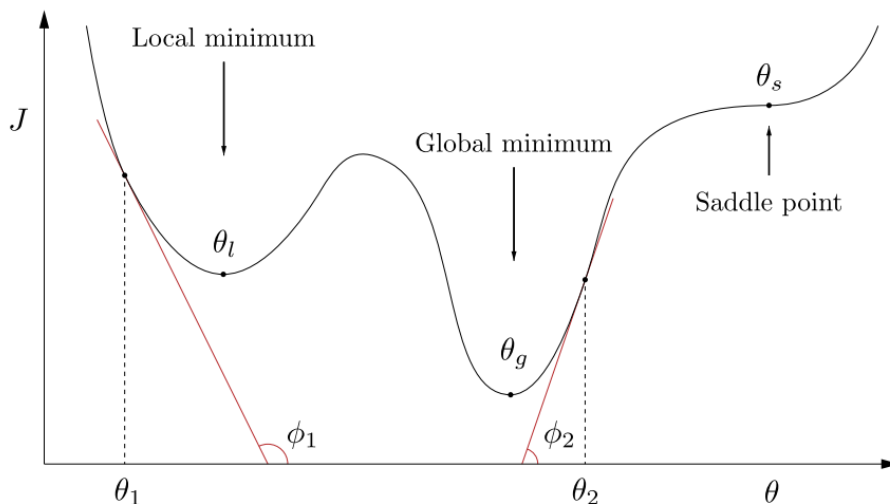
##### ➤ Τοπικά και ολικά ελάχιστα

Η επαναληπτική διαδικασία με την οποία η μέθοδος της απότομης κατάβασης βρίσκει ελάχιστα μιας συνάρτησης είναι:

$$\theta(\text{new}) = \theta(\text{old}) - a \frac{\partial J}{\partial \theta}, \quad (1.39)$$

όπου  $a$  είναι ο ρυθμός μάθησης.

Η αδυναμία της μεθόδου είναι ότι αν η συνάρτηση εμφανίζει πολλά τοπικά ελάχιστα (Εικόνα 1.10) τότε η μέθοδος μπορεί να συγκλίνει σε κάποιο από τα τοπικά ελάχιστα αντί του ολικού.



Εικόνα 1.10: Συνάρτηση με πολλά τοπικά ελάχιστα και saddle points[16]

Η συνέπεια αυτού του προβλήματος είναι πως μπορεί η προσαρμογή του δικτύου στα δεδομένα εκπαίδευσης να μην είναι ικανοποιητική. Παρακάμπτεται με διαδοχικές αρχικοποιήσεις των βαρών του δικτύου και εκπαιδεύσεις εκ νέου έως ότου το MSE στα δεδομένα εκπαίδευσης να είναι ικανοποιητικό.[16, 29]

### ➤ Overfitting

Η ποιότητα της εκπαίδευσης ενός νευρωνικού δικτύου βασίζεται πάνω στα δεδομένα εκπαίδευσης που παρέχονται. Όσο μεγαλύτερο είναι το σετ δεδομένων και όσο μεγαλύτερο εύρος του φαινομένου που μελετάται καλύπτει, τόσο καλύτερη θα είναι και η εκπαίδευση του νευρωνικού δικτύου. Ένας άλλος πολύ σημαντικός παράγοντας είναι η περιπλοκότητα, ή το βάθος, του νευρωνικού δικτύου που σχεδιάζεται (το πλήθος κρυφών στρωμάτων και νευρώνων). Συνήθως πιο περίπλοκες αρχιτεκτονικές οδηγούν σε καλύτερη προσαρμογή στα δεδομένα εκπαίδευσης καθώς το δίκτυο έχει μεγαλύτερη ελευθερία να προσαρμοστεί στο θόρυβο που περιλαμβάνεται στα δεδομένα.[30] Προκύπτει πρόβλημα όταν το δίκτυο προσαρμόζεται τόσο καλά στα δεδομένα εκπαίδευσης και στο θόρυβο που μπορεί να έχουν που εν τέλει χάνει τη δυναμική του φαινομένου. Δίνει δηλαδή τέλεια προσαρμογή στα δεδομένα εκπαίδευσης αλλά παρουσιάζει μεγάλη απόκλιση από την πραγματικότητα. Όσο πιο περιορισμένα είναι τα δεδομένα εκπαίδευσης, ή όσο πιο περίπλοκη η αρχιτεκτονική του δικτύου, τόσο μεγαλύτερος είναι ο κίνδυνος του overfitting.[27, 30].

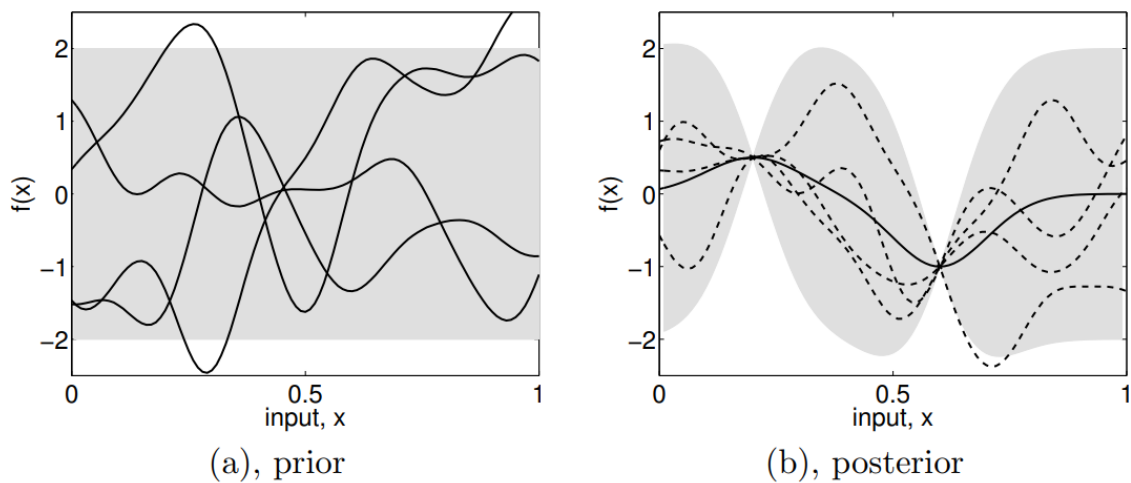
Μία μέθοδο αποφυγής του overfitting είναι η κανονικοποίηση. Η μέθοδος αυτή προσθέτει στην συνάρτηση σφάλματος έναν όρο που εξαρτάται από την περιπλοκότητα του δικτύου οπότε κατά την εκπαίδευση του το δίκτυο μειώνει το πλήθος των νευρώνων που είναι ενεργοί προκειμένου να ελαχιστοποιήσει τη συνάρτηση σφάλματος.[27, 30]

Άλλη μια μέθοδος είναι το early stopping. Αυτή η μέθοδος χωρίζει το αρχικό σετ δεδομένων σε δεδομένα εκπαίδευσης, επαλήθευσης και τεστ. Όσο προχωράει η εκπαίδευση του δικτύου στα δεδομένα εκπαίδευσης (οι επαναλήψεις του υπολογισμού της συνάρτησης σφάλματος και της προσαρμογής των βαρών) υπολογίζεται το MSE της προσαρμογής του δικτύου στα δεδομένα επαλήθευσης. Όσο το αυτό το MSE μειώνεται η επαναληπτική διαδικασία συνεχίζεται. Όταν αρχίσει το MSE να αυξάνεται διακόπτεται η εκπαίδευση του δικτύου καθώς πλέον υπάρχει overfitting και γίνεται η αξιολόγησή του στα δεδομένα τεστ.[27, 28]

### 1.3.3 Γκαουσιανές διεργασίες

Οι Γκαουσιανές Διεργασίες (Γκ.Δ.) είναι ένα πολύ ισχυρό στατιστικό εργαλείο μοντελοποίησης δεδομένων. Βρίσκουν εφαρμογή σε προβλήματα παλινδρόμησης και ταξινόμησης δεδομένων τόσο γραμμικών όσο και μη γραμμικών συναρτήσεων.

Συνήθως για την επίλυση τέτοιων προβλημάτων επιλέγεται ένας τύπος συνάρτησης (πολωνυμική, εκθετική, λογαριθμική, γραμμική, κλπ...) και γίνεται διερεύνηση των παραμέτρων αυτής. Οι Γκ.Δ. αντί να περιορίζουν την ανάλυση με αυτό τον τρόπο, αναθέτουν σε κάθε πιθανή συνάρτηση την πιθανότητα να είναι η βέλτιστη. Έπειτα επιλέγεται ένα τυχαίο σύνολο από αυτές τις κατανομές (προγενέστερες κατανομές), τα χαρακτηριστικά του οποίου ακολουθούν γκαουσιανή κατανομή, και γίνεται αντιστοίχιση αυτού με ένα σετ παρατηρήσεων των δεδομένων που μελετώνται (μεταγενέστερες κατανομές).[31, 32] Αυτή η διαδικασία επιλογής περιγράφεται και διαγραμματικά στην εικόνα 1.11:



Εικόνα 1.11: Εικόνα 1.11-α: Επιλογή 4 τυχαίων συναρτήσεων από το αρχικό σετ με μέση τιμή 0. Εικόνα 1.11-β: Μετά από την εισαγωγή 2 παρατηρήσεων επιλέγονται συναρτήσεις που περνάνε από τα σημεία. Οι συναρτήσεις φαίνεται με τις διακεκομμένες γραμμές και η μέση πρόβλεψη (μέση τιμή των συναρτήσεων) με συνεχή γραμμή. Η σκιασμένη περιοχή αντιστοιχεί στο διπλάσιο της τυπικής απόκλισης σε κάθε  $x$ [31]

Το σετ των συναρτήσεων που περιγράφεται παραπάνω χαρακτηρίζεται απόλυτα από την μέση τιμή και την συνδιακύμανση του[16]:

$$\mu_x = E[f(x)], \quad (1.40)$$

$$\text{cov}_f(x, x') = E[(f(x) - \mu_x)(f(x') - \mu_{x'})] = \kappa(x, x'). \quad (1.41)$$

Η συνάρτηση της συνδιακύμανσης ονομάζεται και συνάρτηση Kernel της Γκ.Δ. και μέσω αυτής γίνεται η επιλογή του σετ συναρτήσεων. Επιλέγοντας την συνάρτηση Kernel της Γκ.Δ. προσδιορίζονται και τα στοιχεία που θα χαρακτηρίζουν τις συναρτήσεις που θα επιλεγούν. Συνήθεις συναστήσεις Kernel:[16]

#### Γραμμική Kernel

$$\kappa(x, x') = x^T x'. \quad (1.42)$$

#### Rational Quadratic Kernel

$$\kappa(x, x') = (1 + \|x - x'\|^2)^{-a}, a \geq 0. \quad (1.43)$$

### Τετραγωνισμένη Εκθετική Kernel (Squared Exponential or Gaussian Kernel)

$$\kappa(x, x') = \theta_0 \exp\left(-\frac{\|x - x'\|^2}{2\theta^2}\right), \quad (1.44)$$

όπου  $h$  είναι μια υπερπαράμετρος του συστήματος που εκφράζει την προσαρμογή των δεδομένων της συνάρτησης στις παρατηρήσεις.[33]

### Radial Basis Function

$$\kappa(x, x') = \theta_0 \exp\left(-\frac{1}{2} \frac{\|x - x'\|^2}{\theta}\right). \quad (1.45)$$

Στις δύο παραπάνω συναρτήσεις Kernel  $\theta$  είναι ένα διάνυσμα  $\theta = [\theta_0 \theta_1 \dots \theta_N]$ , διαστάσεων  $N+1$ , όπου  $N$  είναι οι παρατηρήσεις στις οποίες προσαρμόζονται οι συναρτήσεις.

Καθώς τα  $x, x'$  είναι διανύσματα, η συνάρτηση Kernel είναι ένας πίνακας:

$$K_{ij} = \kappa(x(i), x(i)). \quad (1.46)$$

Μία μέθοδος υπολογισμού των παραμέτρων  $\theta$  είναι η μέθοδος Quasi-Newton optimizer L-BFGS, σύμφωνα με την οποία ελαχιστοποιείται η παρακάτω σχέση:[34]

$$-\log(p(x'|\theta)) = -\frac{1}{2} \log(K) + \frac{1}{2} x'^T K^{-1} x' + \frac{N}{2} \log(2\pi),$$

Όπου  $x'$  είναι το διάνυσμα των παρατηρήσεων,  $p \sim N(0, K)$  ανήκει σε γκαουσιανή κατανομή με μέση τιμή 0 και συνδιακύμανση  $K$  και  $K$  η εξίσωση Kernel.

Πρέπει να σημειωθεί ότι η ύπαρξη του όρου  $\log(K)$  στην παραπάνω εξίσωση διασφαλίζει μειωμένη περιπλοκότητα του μοντέλου καθώς συνδέει τη περιπλοκότητα του μοντέλου με την συνάρτηση σφάλματος που ελαχιστοποιείται, «τιμωρώντας» πιο περίπλοκες δομές.

Εκτός από προβλήματα ταξινόμησης και παλινδρόμησης οι Γκ.Δ. βρίσκουν εφαρμογή και σε προβλήματα όπου η συνάρτηση που μελετάται δεν είναι γνωστή και αναζητούνται οι σημαντικές μεταβλητές της.[34] Σε αυτές τις περιπτώσεις αντί για επιλογή συναρτήσεων από ένα τυχαίο σετ επιλέγονται οι συναρτήσεις που εξετάζονται για την προσαρμογή τους στα πειραματικά δεδομένα. Υπολογίζοντας τον πίνακα της συνάρτησης Kernel μπορούν να προσδιοριστούν οι υπερπαράμετροι  $\theta$  που αντιστοιχούν σε κάθε συνάρτηση (και άρα μεταβλητή) που εξετάζονται και να προσδιοριστούν αυτές που έχουν τη μεγαλύτερη επιρροή στις παρατηρήσεις.

## 2 Κεφάλαιο 2<sup>ο</sup>: Μεθοδολογία/ Επίλυση Δ.Ε. με μεθόδους Μηχανικής Μάθησης σε περιβάλλον MATLAB

Σκοπός της παρούσας διπλωματικής εργασίας είναι η θεμελίωση μιας μεθόδου σύμφωνα με την οποία θα είναι δυνατόν να εκπαιδευτεί ένα νευρωνικό δίκτυο να επιλύει εξισώσεις της μορφής  $\frac{\partial u}{\partial t} = f(t, x, y, z)$ , δηλαδή Δ.Ε. όπου το δεξί μέλος είναι άγνωστο, χρησιμοποιώντας ως δεδομένα εισόδου πειραματικά αποτελέσματα.

Αρχικά καταστρώνεται διαδικασία όπου εκπαιδευτεί νευρωνικό δίκτυο στην επίλυση Δ.Ε. όπου το δεξί μέλος είναι γνωστό. Συγκεκριμένα στην ενότητα 2.1 εξετάζεται το μοντέλο της μοριακής διάχυσης, ενώ στην ενότητα 2.2 εξετάζεται το πρόβλημα Bratu. Σε αυτές τις ενότητες τα δεδομένα εκπαίδευσης των νευρωνικών δικτύων παράγονται με αριθμητική επίλυση των γνωστών Δ.Ε. Τα δεξιά μέλη των Δ.Ε. αντιμετωπίζονται ως άγνωστα μόνο για την εφαρμογή γκαουσιανών διεργασιών, όπου χρησιμοποιούνται για την επικύρωση των αποτελεσμάτων που προκύπτουν.

Πρέπει να σημειωθεί πως η ανάλυση που πραγματοποιείται στις ενότητες 2.1 και 2.2 εστιάζει στις σημαντικές συναρτήσεις που παρουσιάζονται στις εκφράσεις των παραγώγων των δύο προβλημάτων που εξετάζονται. Οι μεταβλητές των δύο προβλημάτων,  $D$  και  $\lambda$ , θεωρούνται σταθερές καθώς στόχος της ανάλυσης είναι η κατάστρωση μεθοδολογίας σχεδιασμού νευρωνικού δικτύου που να λειτουργεί ως μαύρο κουτί για την επίλυση των Δ.Ε. και όχι η γενίκευση αυτού για όλη την οικογένεια λύσεων των δύο προβλημάτων.

Έπειτα, στην ενότητα 2.3, εκπαιδευτεί νευρωνικό δίκτυο στην επίλυση μοντέλου προσομοιωτή μικροσκοπικής κλίμακας. Για την παραγωγή των δεδομένων εισόδου χρησιμοποιείται το μοντέλο των Random Walkers, το οποίο αποτελεί ένα πρόβλημα Monte Carlo. Οι παράμετροι του μοντέλου ρυθμίζονται έτσι ώστε να προσομοιώνει το μοντέλο της μοριακής διάχυσης. Σε αυτή τη περίπτωση τα αποτελέσματα είναι οι θέσεις των τυχαίων βηματιστών που αντιστοιχούν στα μόρια τα οποία διαχέονται αντί για την συγκέντρωση. Εφόσον είναι άγνωστη η Δ.Ε. και άρα δεν είναι γνωστό ποιες είναι οι σημαντικές παράμετροι του μοντέλου, είναι απαραίτητο να προηγηθεί ανάλυση γκαουσιανών διεργασιών πριν την εκπαίδευση του νευρωνικού δικτύου για τον προσδιορισμό των μεταβλητών εισόδου.

### 2.1 Μοριακή Διάχυση σε Μια Διάσταση

Αρχικά εξετάζεται το πρόβλημα εκπαίδευσης νευρωνικού δικτύου στην επίλυση της Δ.Ε. της μοριακής διάχυσης, σε μία διάσταση, σε μεταβατική κατάσταση:

$$\frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial x^2} \quad (2.1)$$

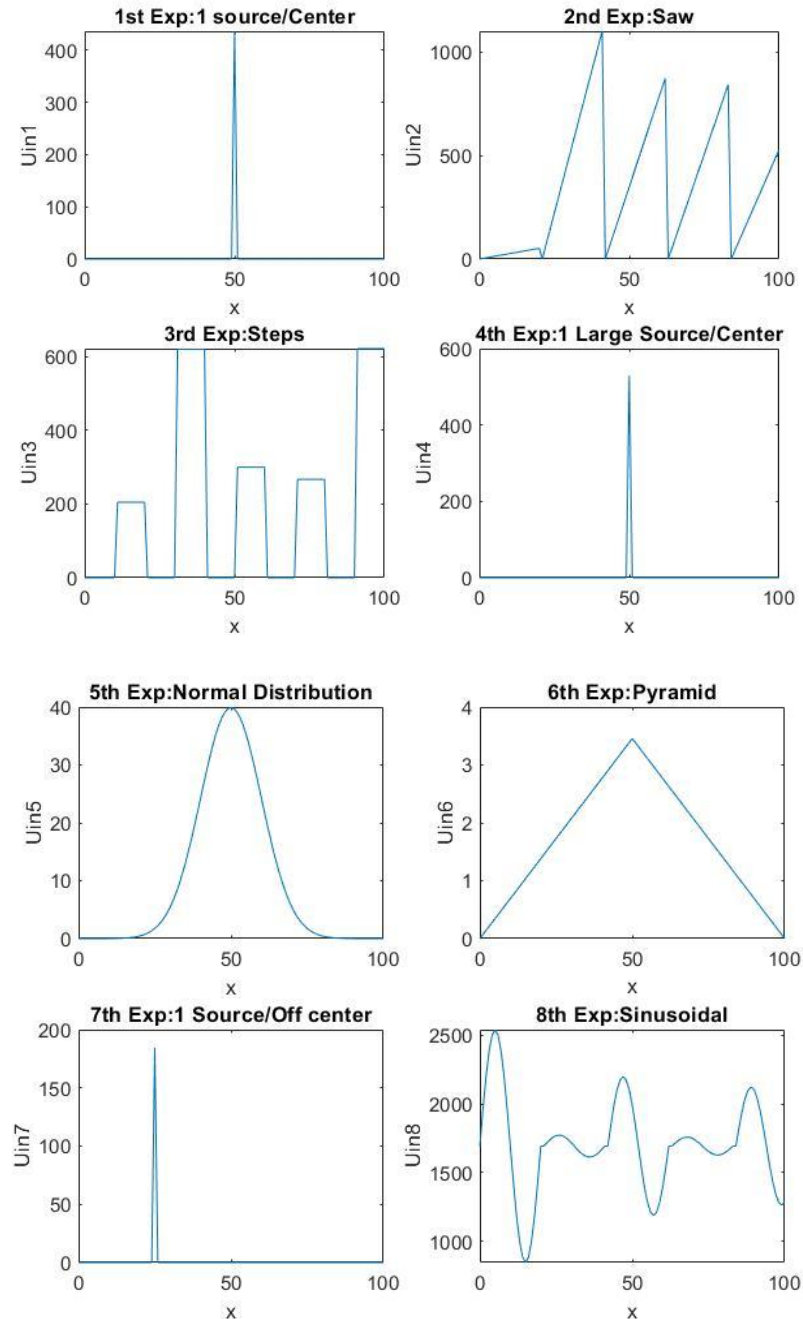
Επιλέχθηκε αυτό το μοντέλο καθώς αποτελεί ένα από τα απλούστερα παραδείγματα στα οποία θα μπορούσε να εφαρμοστεί η διαδικασία που μελετάται στη παρούσα διπλωματική.

Η εξίσωση της διάχυσης βρίσκει εφαρμογή μέσω του νόμου του Fick στην επίλυση προβλημάτων διάχυσης μάζας και θερμότητας. Παραδειγματικά, χρησιμοποιείται στην πρόβλεψη του προφίλ της θερμοκρασίας πτερυγίων ψύξης σε εναλλάκτες θερμότητας.

#### 2.1.1 Καθορισμός Αρχικών Κατανομών

Συνολικά δημιουργούνται 8 αρχικές κατανομές, κάθε μία συμβολίζει και ένα διαφορετικό πείραμα. Οι 6 από αυτές χρησιμοποιούνται για την παραγωγή του σετ δεδομένων εκπαίδευσης του νευρωνικού δικτύου ενώ οι 2 χρησιμοποιούνται για την αξιολόγηση του σε κατανομές πέρα από αυτές στις οποίες εκπαιδεύτηκες και σε πολύ μεγαλύτερους χρόνους.

Οι αρχικές κατανομές που χρησιμοποιούνται έχουν σχεδιαστεί για να εκπαιδευτεί το νευρωνικό δίκτυο σε μια μεγάλη ποικιλία συναρτήσεων και συνθηκών, είτε είναι μια σημειακή πηγή είτε μια συνεχής κατανομή. Η επιλογή τους έχει μεγάλη σημασία καθώς αν το νευρωνικό δίκτυο εκπαιδευτεί επί περιορισμένων δεδομένων τότε θα περιοριστεί σημαντικά και η ικανότητα του να επιλύει το πρόβλημα για το οποίο σχεδιάζεται. [27, 30]



Εικόνα 2.1: Αρχικές κατανομές για το πείραμα της μοριακής διάχυσης.

Οι παραπάνω κατανομές εκπαίδευσης (Εικόνα 2.1) παρέχουν δεδομένα για σημειακές πηγές, για σταθερές κατανομές, για απότομες μεταβολές από μεγάλες μη μηδενικές τιμές σε μηδενικές και για κατανομές με μεγάλο εύρος και διακύμανση τιμών.

### 2.1.2 Αριθμητική Επίλυση της Δ.Ε.

Όπως περιγράφηκε και στην ενότητα 1.4.2 η αδυναμία των νευρωνικών δικτύων είναι η ανάγκη από πειραματικές μετρήσεις, από την αναλυτική ή την αριθμητική επίλυση της Δ.Ε. Στην παρούσα διπλωματική εργασία επιλέχθηκε για αυτό το σκοπό μια μέθοδος αριθμητικής επίλυσης, η μέθοδος των πεπερασμένων διαφορών.

Το χωρίο στο οποίο γίνεται η επίλυση είναι το διάστημα  $x \in [0,100]$  με βήμα  $dh=1$  και διακριτοποιείται σε 101 κόμβους. Το χρονικό διάστημα επίλυσης είναι το  $t \in [0,100]$  με χρονικό βήμα  $\Delta t=0,1$ .

Η επίλυση της Δ.Ε. πραγματοποιήθηκε με εφαρμογή της εντολής `ode15s` σε περιβάλλον MATLAB. Για την έκφραση της Δ.Ε. χρησιμοποιήθηκε προσέγγιση του δεξιού μέλους με χρήση σχήματος κεντρικών διαφορών. Ορίστηκε το χρονικό διάστημα επίλυσης που περιγράφεται παραπάνω και το χρονικό βήμα  $\Delta t$  ορίστηκε αυτόματα από την εντολή.

Η διακριτοποίηση του προβλήματος παρουσιάζεται στο παράρτημα 6.1.

Έχοντας υπολογίσει την λύση της Δ.Ε. για κάθε αρχική κατανομή το επόμενο βήμα είναι ο υπολογισμός των χωρικών παραγώγων κάθε λύσης σε κάθε χρονική στιγμή. Χρησιμοποιώντας το σχήμα εμπρός διαφορών για τις πρώτες παραγώγους ως προς την διεύθυνση  $x$ , το σχήμα κεντρικών διαφορών για τις δεύτερες παραγώγους ως προς την διεύθυνση  $x$  και εφαρμόζοντας τις παρακάτω συνοριακές συνθήκες:

$$\bullet \quad \left. \frac{\partial u}{\partial x} \right|_{\text{άκρο}} = 0, \quad (2.2)$$

$$\bullet \quad \left. \frac{\partial^2 u}{\partial x^2} \right|_{\text{άκρο}} = 0, \quad (2.3)$$

προσεγγίζονται αριθμητικά οι χωρικές παράγωγοι.

Παρόλο που ουσιαστικά είναι γνωστή η Δ.Ε. πραγματοποιούμε τους υπολογισμούς μας "υποκρινόμενοι" ότι δεν τη γνωρίζουμε. Προσεγγίζεται η χρονική παράγωγος και οι χωρικές παράγωγοι χρησιμοποιώντας διακριτές τιμές της συνάρτησης  $u$  και στη συνέχεια ανιχνεύονται ποιες από αυτές τις μεταβλητές έχουν ουσιαστική επίδραση στη χρονική παράγωγο. Αυτό γίνεται με τις Γκαουσιανές Διεργασίες.

### 2.1.3 Γκαουσιανές Διεργασίες

Κατά την επίλυση του προβλήματος της μοριακής διάχυσης οι Γκ.Δ εφαρμόστηκαν ως μέθοδος επικύρωσης και όχι διερευνητικά καθώς είναι γνωστή η Δ.Ε. που επιλύεται και άρα και οι σημαντικές μεταβλητές της. Αναμενόμενο αποτέλεσμα των Γκ.Δ. είναι να αναδειχθεί η δεύτερη χωρική παράγωγος της λύσης,  $u_{xx}$ , ως η μοναδική σημαντική μεταβλητή του προβλήματος.

Για την εφαρμογή Γκ.Δ. χρησιμοποιήθηκαν 1200 σετ δεδομένων  $[u \ u_x \ u_{xx} \ u_t]$ , τα οποία λήφθηκαν από 10 διαφορετικούς χρόνους σε 20 από τους 40 κεντρικούς κόμβους του χωρίου.

Για την σήμανση των κόμβων στους οποίους έγινε δειγματοληψία για την συμπλήρωση του παραπάνω σετ δεδομένων επιλέχθηκαν τυχαία 20 από τον 30° έως τον 70° κόμβο.

Οι χρόνοι στους οποίους έγινε δειγματοληψία επιλέχθηκαν ούτως ώστε να ακολουθούν λογαριθμική κλίμακα. Σκοπός αυτού ήταν η επιλογή περισσότερων χρόνων προς το τέλος της προσομοίωσης όπου πλέον θα είχε αναπτυχθεί το φαινόμενο προκειμένου να υπάρχει πλήρη εικόνα της εξέλιξης του, χωρίς όμως να παραληφθούν τελείως μικροί, συγκριτικά, χρόνοι, όπου θα ήταν μικρότερες οι μεταβολές.

Από την παραπάνω δειγματοληψία συλλέχθηκαν 1200 σετ τυχαία διατεταγμένων δεδομένων. Διατηρήθηκε μόνο ο συσχετισμός των κόμβων μεταξύ των 4 σετ δεδομένων, ένα για κάθε μεταβλητή. Δηλαδή, παρόλο που συνολικά η δειγματοληψία έγινε σε τυχαία διατεταγμένους κόμβους, σε κάθε σετ δεδομένων κάθε στοιχείο αντιστοιχεί στον ίδιο κόμβο με κάθε άλλο σετ.

Για την εφαρμογή Γκ.Δ. χρησιμοποιήθηκαν ως παρατηρήσεις τα δεδομένα της χρονικής παραγωγού και διερευνήθηκε η προσαρμογή των δεδομένων της λύσης και των υπόλοιπων παραγωγών αυτής. Ως συνάρτηση Kernel χρησιμοποιήθηκε η συνάρτηση squared exponential για τον υπολογισμό της υπερπαραμέτρου κάθε μεταβλητής. Έπειτα αξιολογήθηκε η προσαρμογή των συναρτήσεων με βάση τις υπερπαραμέτρους, αυτές με τις χαμηλότερες τιμές εμφανίζουν καλύτερη προσαρμογή.

#### 2.1.4 Εκπαίδευση Νευρωνικού Δικτύου

Όπως είναι γνωστό από την έκφραση του προβλήματος της μοριακής διάχυσης, αλλά και όπως προέκυψε από την ανάλυση των Γκ.Δ., η σημαντική μεταβλητή του προβλήματος είναι η δεύτερη παράγωγος ως προς την διεύθυνση  $x$ . Η σχέση δηλαδή είναι της μορφής:

$$\frac{\partial u}{\partial t} = f \left( \frac{\partial^2 u}{\partial x^2} \right) \quad (2.4)$$

Ο σκοπός είναι να χρησιμοποιηθεί το νευρωνικό δίκτυο στην θέση της συνάρτησης  $f$  για να παράγει αυτόματα την λύση σε κάθε επιθυμητό χρόνο. Δηλαδή:

$$\frac{\partial u}{\partial t} = net \left( \frac{\partial^2 u}{\partial x^2} \right) \quad (2.5)$$

Η εκπαίδευση του νευρωνικού δικτύου ακολουθεί την μέθοδο της εποπτευόμενης μάθησης, οπότε σαν δεδομένα εισόδου παρέχονται και τα δεδομένα εισαγωγής και τα δεδομένα στόχου. Δεδομένα εισαγωγής αποτελούν τα δεδομένα της δεύτερης παραγωγού της λύσης ως προς τη διεύθυνση  $x$  ενώ σα δεδομένα στόχου εισάγοντα τα δεδομένα της πρώτης παραγωγού της λύσης ως προς το χρόνο που αντιστοιχούν στα δεδομένα της  $u_{xx}$  που εισήχθησαν ως δεδομένα εισαγωγής. Εκπαιδεύεται λοιπόν το δίκτυο να λαμβάνει δεδομένα για την  $u_{xx}$  και να αποδίδει τα δεδομένα που αντιστοιχούν στην  $u_t$ .

Καθώς δεν υπάρχει συγκεκριμένη μεθοδολογία σύμφωνα με την οποία μπορεί να καθοριστεί η βέλτιστη αρχιτεκτονική για την αντιμετώπιση του εκάστοτε προβλήματος, ακολουθείται η προσέγγιση της δοκιμής και σφάλματος. Υποθέτοντας πως ένα ρηχό νευρωνικό δίκτυο θα μπορεί να ανταπεξέλθει στις απαιτήσεις του προβλήματος της μοριακής διάχυσης, δοκιμάζονται διάφορες αρχιτεκτονικές μέχρι να βρεθεί αυτή που παρουσιάζει το μικρότερο σφάλμα.

Πρέπει να σημειωθεί πως η αξιολόγηση του δικτύου πραγματοποιείται και αυτόματα από τα εργαλεία σχεδιασμού νευρωνικών δικτύων του MATLAB για την προσαρμογή των συντελεστών κάθε node, οπότε από τα δεδομένα εισαγωγής ένα τμήμα ορίζεται ως training set. Βέβαια αυτή η αξιολόγηση εξετάζει την προσαρμογή του δικτύου στα δεδομένα εισαγωγής, για αυτό και μετά πραγματοποιείται και αξιολόγηση της προσαρμογής του δικτύου σε νέες κατανομές.

Η εκπαίδευση όλων των νευρωνικών δικτύων έγινε με τις εξής παραμέτρους:



Πίνακας 2.1: Δεδομένα εκπαίδευσης νευρωνικών δικτύων για το πρόβλημα της μοριακής διάχυσης

Επιλογή δεδομένων εισόδου	Τυχαία
Ποσοστό δεδομένων εκπαίδευσης	70%
Ποσοστό δεδομένων αξιολόγησης	15%
Ποσοστό δεδομένων τεστ	15%
Αλγόριθμος εκπαίδευσης	Levenberg Marquardt
Αλγόριθμος επίδοσης	Mean Squared Error (MSE)
Μέγιστος αριθμός επαναλήψεων εκπαίδευσης	2000
Συνάρτηση μεταφοράς εσωτερικών νευρώνων	Tansig
Συνάρτηση μεταφοράς εξωτερικών νευρώνων	Purelin

Συνολικά εξετάστηκαν 21 διαφορετικές αρχιτεκτονικές για την εύρεση της βέλτιστης, οι οποίες παρουσιάζονται στον παρακάτω πίνακα.

Πίνακας 2.2: Δομές νευρωνικών δικτύων που εκπαιδεύτηκαν και εξετάστηκαν για την εύρεση της βέλτιστης αρχιτεκτονικής για το πρόβλημα της μοριακής διάχυσης

networks	layers	neurons-1	neurons-2	neurons3
net-1	1	1		
net-2	1	2		
net-3	1	5		
net-4	1	7		
net-5	1	10		
net-6	2	1	1	
net-7	2	1	3	
net-8	2	1	5	
net-9	2	2	1	
net-10	2	2	2	
net-11	2	3	1	
net-12	3	1	1	1
net-13	3	1	2	2
net-14	3	1	3	3
net-15	3	1	2	3
net-16	3	2	2	2
net-17	3	2	3	1
net-18	3	2	3	3
net-19	3	2	3	5
net-20	3	1	3	5
net-21	3	2	5	3

Για κάθε αρχιτεκτονική που εξετάζεται γίνονται 10 διαδοχικές εκπαιδεύσεις του δικτύου και αποθηκεύεται εν τέλει αυτή που παρουσίασε το χαμηλότερο MSE προκειμένου να διασφαλιστεί η καλύτερη δυνατή προσαρμογή στα δεδομένα εκπαίδευσης. Έπειτα, για κάθε ένα από τα δίκτυα που έχουν εκπαιδευτεί εξετάζεται η προσαρμογή σε νέα αρχική κατανομή και σε χρόνο πολύ μεγαλύτερο από αυτών στον οποίο παράχθηκαν τα δεδομένα εκπαίδευσης. Υπολογίζεται η λύση της Δ.Ε., με την νέα αρχική κατανομή, για χρόνους 100 και 1000, πρώτα με την μεθοδολογία που περιγράφεται στην ενότητα 2.1.2 και έπειτα με το νευρωνικό δίκτυο.

Μετά υπολογίζονται οι δείκτες προσαρμογής AUF (Area under function) και MAPE (Mean absolute percentage error) για να αξιολογηθεί η προσαρμογή κάθε νευρωνικού δικτύου σε νέες κατανομές.

- **AUF (Area under functions)**

$$AUF = \left| \frac{\int_2^{N-1} u_{calc} dx - \int_2^{N-1} u_{net} dx}{\int_2^{N-1} u_{calc} dx} \right| \quad (2.6)$$

- **MAPE (Mean absolute percentage error)**

$$MAPE = \sum_{i=2}^{N-1} \left| \frac{u_{calc}(i) - u_{net}(i)}{u_{calc}(i)} \right| \frac{100\%}{N-2} \quad (2.7)$$

όπου  $u_{calc}(i)$ : η υπολογισμένη τιμή της λύσης  $u$  στον κόμβο  $i$ ,  $u_{net}(i)$ : η τιμή της λύσης στον κόμβο  $i$  που υπολογίζεται από το νευρωνικό δίκτυο,  $N$ : το πλήθος των κόμβων

Το άθροισμα διαιρείται με  $N-2$  αντί για  $N$  για τον υπολογισμό του μέσου όρου διότι στα άκρα του διαστήματος εφαρμόζονται συνοριακές τιμές και άρα δεν συμμετέχουν στον υπολογισμό του σφάλματος.

Πρέπει να σημειωθεί ότι τα δεδομένα εξόδου του νευρωνικού δικτύου είναι οι τιμές της πρώτης παραγώγου της λύσης ως προς τον χρόνο, για τον χρόνο που έχει οριστεί, οπότε για την σύγκριση των δεδομένων του νευρωνικού δικτύου με αυτά που παράγονται από την αριθμητική επίλυση, θα πρέπει να υπολογιστεί η λύση από αυτά. Όπως γίνεται και στην ενότητα 2.1.2, η λύση υπολογίζεται με την εντολή του MATLAB ode15s.

Τελικά επιλέγεται το νευρωνικό δίκτυο που παρουσιάζει το ελάχιστο MSE ανάμεσα στις 10 πρώτες εκπαιδεύσεις και τις ελάχιστες τιμές των συντελεστών αξιολόγησης με την 7<sup>η</sup> και την 8<sup>η</sup> αρχική κατανομή σε χρόνους 100 και 1000.

## 2.2 Bratu σε Μια Διάσταση

Όπως περιγράφηκε στην ενότητα 1.1.5 το πρόβλημα Bratu διατυπώνεται από την εξίσωση 1.19. Σε μεταβατική κατάσταση, σε μία διάσταση, το πρόβλημα διατυπώνεται:

$$\frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial x^2} + \lambda e^u \quad (2.8)$$

Η ανάλυση του προβλήματος θα εστιάσει στην συνεισφορά του μη γραμμικού, εκθετικού όρου στην λύση του προβλήματος και στην εφαρμογή μεθόδων μηχανικής μάθησης. Θα εξεταστεί το κατά πόσο θα μπορέσουν οι γκαουσιανές διεργασίες να αποδώσουν σωστά αποτελέσματα σχετικά με τις σημαντικές μεταβλητές του προβλήματος και η επιτυχία και ακρίβεια προσαρμογής ενός ρηχού νευρωνικού δικτύου.

### 2.2.1 Καθορισμός Αρχικών Κατανομών

Θα χρησιμοποιηθούν οι ίδιες αρχικές κατανομές που αξιοποιήθηκαν και στο πρόβλημα της μοριακής διάχυσης, στην ενότητα 2.1.1, καθώς σκοπός είναι το να παραχθεί ένα όσο το δυνατόν πιο αντιπροσωπευτικό σετ δεδομένων της δυναμικής του προβλήματος.

### 2.2.2 Αριθμητική Επίλυση του προβλήματος

Η αριθμητική επίλυση της εξίσωσης 2.12 για την παραγωγή του σετ δεδομένων που απαιτούνται ακολουθεί την ίδια διαδικασία με αυτή που περιγράφηκε στην ενότητα 2.1.2.

Το χωρίο στο οποίο γίνεται η επίλυση είναι το διάστημα  $\chi \in [0,1]$  με βήμα  $dh=0.01$  και διακριτοποιείται σε 101 κόμβους. Το χρονικό διάστημα επίλυσης είναι το  $t \in [0,1]$  με χρονικό

βήμα  $\Delta t = 10^{-5}$ . Αλλάζει η διακριτοποίηση του χωρίου και του χρονικού διαστήματος καθώς επιθυμητή είναι μόνο η ευσταθής λύση και άρα παράγεται μόνο αυτός ο κλάδος.

Από την αριθμητική επίλυση του προβλήματος προκύπτει η λύση  $u$  και έπειτα προσεγγίζονται οι χωρικές και η χρονική παράγωγο σε κάθε σημείο, σε κάθε χρόνο.

### 2.2.3 Γκαουσιανές Διεργασίες

Όπως και στο πρόβλημα της μοριακής διάχυσης, έτσι και στην ανάλυση του προβλήματος Bratu είναι γνωστές ως σημαντικές μεταβλητές η λύση της εξίσωσης και η δεύτερη χωρική της παράγωγος.

Για την εφαρμογή των Γκ.Δ επιλέγονται 1200 σετ δεδομένων, επιλεγμένων με την ίδια μέθοδο της ενότητας 2.1.3, από την λύση του προβλήματος, την πρώτη και δεύτερη χωρική παράγωγό της και την χρονική της παράγωγο. Τα δεδομένα που συλλέχθηκαν διατάχθηκαν με τυχαίο τρόπο πριν την εφαρμογή της μεθόδου. Συγκρίνεται η προσαρμογή της λύσης και των χωρικών παραγώγων της πάνω στα δεδομένα της χρονικής παραγώγου. Επιλέγεται πάλι η συνάρτηση Kernel η συνάρτηση squared exponential. Έπειτα υπολογίζονται οι υπερπαραμέτροι των μεταβλητών για τον προσδιορισμό των σημαντικών μεταβλητών.

### 2.2.4 Εκπαίδευση Νευρωνικού Δικτύου

Στην περίπτωση του προβλήματος Bratu εκπαιδεύεται νευρωνικό δίκτυο:

$$\frac{\partial u}{\partial t} = \text{net} \left( u, \frac{\partial^2 u}{\partial x^2} \right), \quad (2.9)$$

καθώς η λύση του προβλήματος και η δεύτερη παράγωγος της λύσης ως προς την διεύθυνση  $x$  αναδείχθηκαν οι σημαντικές μεταβλητές.

Η κατασκευή και η αξιολόγηση του δικτύου ακολουθούν την ίδια διαδικασία της ενότητας 2.1.4.

Οι παράμετροι εκπαίδευσης του νευρωνικού δικτύου για το πρόβλημα Bratu:

Πίνακας 2.3: Δεδομένα εκπαίδευσης νευρωνικών δικτύων για το πρόβλημα Bratu

Επιλογή δεδομένων εισόδου	Τυχαία
Ποσοστό δεδομένων εκπαίδευσης	70%
Ποσοστό δεδομένων αξιολόγησης	15%
Ποσοστό δεδομένων τεστ	15%
Αλγόριθμος εκπαίδευσης	Levenberg Marquardt
Αλγόριθμος επίδοσης	Mean Squared Error (MSE)
Μέγιστος αριθμός επαναλήψεων εκπαίδευσης	2000
Συνάρτηση μεταφοράς εσωτερικών νευρώνων	Tansig
Συνάρτηση μεταφοράς εξωτερικών νευρώνων	Purelin

Η εκπαίδευση νευρωνικού δικτύου για την επίλυση του προβλήματος Bratu ακολουθεί το ίδιο πνεύμα με τη μεθοδολογία που αναπτύχθηκε για το πρόβλημα της μοριακής διάχυσης. Η διαφορά είναι ότι στην προκειμένη περίπτωση πρέπει να αντιμετωπιστεί το ζήτημα της μη γραμμικότητας που προκύπτει από την ύπαρξη του όρου  $le^u$ . Αναμένεται ότι η βέλτιστη αρχιτεκτονική που θα περιγράψει το πρόβλημα Bratu δεν θα διαφέρει πολύ από αυτή που προσδιορίστηκε για το πρόβλημα της μοριακής διάχυσης. Για αυτό εξετάζονται πολύ λιγότερες

περιπτώσεις σε σχέση με αυτές που περιγράφονται στην ενότητα 2.1.4, εστιάζοντας στην αρχιτεκτονική που αναδείχθηκε βέλτιστη και σε λίγο πιο περίπλοκες αρχιτεκτονικές από αυτή.

Πίνακας 2.4: Δομές νευρωνικών δικτύων που εκπαιδεύτηκαν και εξετάστηκαν για την εύρεση της βέλτιστης αρχιτεκτονικής για το πρόβλημα Bratu

networks	layers	neurons-1	neurons-2
net-1	1	5	
net-2	1	3	
net-3	1	7	
net-4	2	1	3
net-5	2	2	3
net-6	2	3	3
net-7	2	3	1

Για κάθε αρχιτεκτονική που εξετάζεται γίνονται 10 διαδοχικές εκπαιδεύσεις και διατηρείται αυτή με το χαμηλότερο MSE. Έπειτα τα δίκτυα εξετάζονται στη προσαρμογή τους στις καινούργιες κατανομές για χρόνους 0,1 και 1 με τους δείκτες αξιολόγησης AUF και MAPE. Τελικά επιλέγεται η αρχιτεκτονική που παρουσίασε τις χαμηλότερες τιμές σε όλους τους δείκτες αξιολόγησης στα τεστ.

## 2.3 Random Walkers

Το μοντέλο των Random Walkers ή αλλιώς τυχαίων βηματιστών είναι ένας προσομοιωτής μικροσκοπικής κλίμακας. Όπως περιγράφεται στην ενότητα 1.2 για τα μοντέλα MC, χρησιμοποιείται ένας μεγάλος πληθυσμός βηματιστών που κινούνται τυχαία, με τον οποίο προσομοιώνεται το φαινόμενο της μοριακής διάχυσης.

### 2.3.1 Αρχικές Θέσεις Βηματιστών

Η αρχικοποίηση της θέσης των βηματιστών ακολουθεί παρόμοια διαδικασία με αυτή των προηγούμενων κεφαλαίων. Η διαφορά είναι ότι στα προηγούμενα μοντέλα η λύση της Δ.Ε. ήταν η συγκέντρωση μορίων σε θέση  $x$  ενώ τώρα οι βηματιστές είναι τα μόρια. Πρέπει λοιπόν να γίνει μετατροπή των αρχικών κατανομών που χρησιμοποιήθηκαν στα προηγούμενα προβλήματα προκειμένου να είναι συμβατά με τις θέσεις των βηματιστών. Αυτή η μετατροπή γίνεται με την διαδικασία του lifting:[35]

Το διάστημα στο οποίο γίνονται οι υπολογισμοί,  $x$ , έχει διακριτοποιηθεί σε ένα πλήθος  $N$  κόμβων. Σε κάθε κόμβο είναι γνωστή η τιμή της συγκέντρωσης μορίων. Με αυτά τα δεδομένα υπολογίζεται η συνάρτηση αθροιστικής κατανομής των μορίων (CDF):

$$f_i = \int_{x_0}^{x_i} u dx \quad (2.10)$$

όπου  $u$  η αρχική συνάρτηση συγκέντρωσης των μορίων και  $i$  ο κόμβος που υπολογίζεται το η CDF.

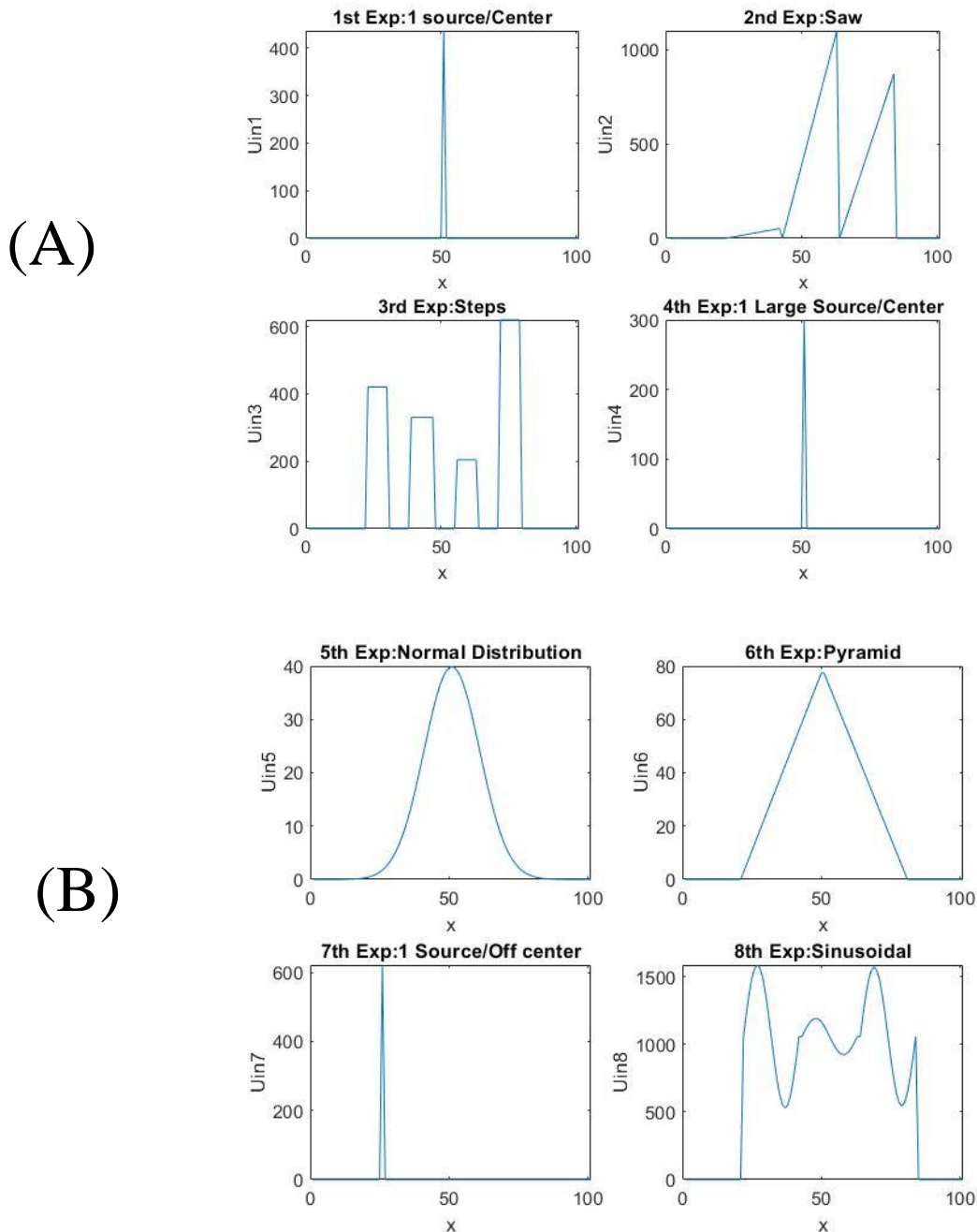
Έπειτα υπολογίζεται η συνάρτηση πυκνότητας πιθανότητας των μορίων στο μέσο των διαστημάτων ανάμεσα στους κόμβους:

$$N_{i-\frac{dh}{2}} = \frac{f_i - f_{i-1}}{x_i - x_{i-1}} \quad (2.11)$$

Η τιμή της  $N$  αντιστοιχεί στην συγκέντρωση μορίων εντός του διαστήματος ανάμεσα στους κόμβους  $i$  και  $i-1$ .

Τέλος κατανέμεται ομοιόμορφα ανάμεσα στους δύο κόμβους το πλήθος των βηματιστών που αντιστοιχεί στην υπολογισμένη συγκέντρωση.

Πρέπει να σημειωθεί ότι η συνάρτηση PDF είναι κανονικοποιημένη και η τιμή της σε κάθε κόμβο αντιστοιχεί σε ποσοστό του συνολικού πληθυσμού των βηματιστών. Διαιρείται με το άθροισμα των τιμών της και πολλαπλασιάζεται με το συνολικό πλήθος των βηματιστών για να δώσει τις σωστές τιμές για τον πληθυσμό που έχει επιλεγθεί. Οι αρχικές κατανομές που χρησιμοποιούνται για τον υπολογισμό των αρχικών θέσεων των βηματιστών για κάθε πείραμα παρουσιάζονται παρακάτω:



Εικόνα 2.2: Αρχικές κατανομές για το πείραμα των τυχαίων βηματιστών. Εικόνα 2.1-A: Αρχικές κατανομές 1 έως 4, Εικόνα 2.1-B: Αρχικές κατανομές 5 έως 8

### 2.3.2 Επίλυση Μοντέλου

Το χωρίο στο οποίο γίνεται η επίλυση είναι το διάστημα  $x \in [0,100]$  με βήμα  $dh=1$  και διακριτοποιείται σε 101 κόμβους. Το χρονικό διάστημα επίλυσης είναι το  $t \in [0,1]$  με χρονικό βήμα  $\Delta t=10^{-4}$ . Επιλέγονται πληθυσμοί  $10^4 - 10^5$  βηματιστών.

Στα άκρα του χωρίου εφαρμόζονται συνοριακές συνθήκες τύπου Dirichlet, ορίζεται η τιμή της συγκέντρωσης των βηματιστών στα άκρα μηδενική.

Ο πυρήνας του συγκεκριμένου προβλήματος είναι η επαναληπτική διαδικασία που υπολογίζει τις θέσεις των βηματιστών σε κάθε νέο χρόνο. Επιλέγεται για κάθε βηματιστή μία τυχαία τιμή, από το 0 μέχρι το 1, και η νέα θέση του βηματιστή υπολογίζεται:

$$X_{new} = X_{old} + \begin{cases} (2 * p - 1)dx, & p \geq 0.5 \\ -dx, & p < 0.5 \end{cases} \quad (2.12)$$

όπου  $X$  η θέση του βηματιστή,  $p$  η τυχαία τιμή που υπολογίζεται,  $dx$  το βήμα της MC.

Για την εφαρμογή Γκ.Δ λαμβάνονται τιμές για τις θέσεις των βηματιστών σε 10 χρονικές στιγμές. Μετά από κάθε λήψη γίνεται ταξινόμηση των θέσεων, οργανώνοντάς τες από την μικρότερη στη μεγαλύτερη. Έτσι προκύπτει η συνάρτηση CDF των θέσεων. Επειδή ο πληθυσμός των βηματιστών είναι σχετικά μικρός οι τιμές που προκύπτουν έχουν θόρυβο. Αυτό αντιμετωπίζεται επαναλαμβάνοντας τη διαδικασία 10 με 100 φορές και υπολογίζοντας τη μέση τιμή των μετρήσεων.

Τελικά για κάθε πείραμα προκύπτουν αποτελέσματα για τις θέσεις των βηματιστών σε 10 χρονικές στιγμές. Για να συνάδει η κλίμακα της επίλυσης με τις αρχικές κατανομές από τις οποίες υπολογίζεται πολλαπλασιάζεται με την τιμή του ολοκληρώματος της εκάστοτε αρχικής κατανομής. Επίσης, επειδή ακόμα υπάρχει θόρυβος στα δεδομένα, ο οποίος θα δημιουργήσει πρόβλημα στην αριθμητική προσέγγιση των παραγώγων, γίνεται και φιλτράρισμα.

Έπειτα προσεγγίζονται αριθμητικά η πρώτη και η δεύτερη χωρική παράγωγος της CDF των θέσεων των βηματιστών ως προς την διεύθυνση  $x$  και η χρονική παράγωγός της.

### 2.3.3 Γκαουσιανές Διεργασίες

Στο πρόβλημα των τυχαίων βηματιστών δεν είναι γνωστές οι σημαντικές μεταβλητές στην έκφραση της Δ.Ε. Θα πρέπει να αναδειχτούν μέσω ανάλυσης γκαουσιανών διεργασιών. Εφόσον τα δεδομένα συμπεριλαμβάνουν θόρυβο θα χρησιμοποιηθεί μεγαλύτερο σετ απ' ό,τι στις προηγούμενες εφαρμογές. Γίνεται δειγματοληψία των 80 εσωτερικών κόμβων (10 έως 90) σε 10 χρονικές στιγμές. Δημιουργούνται λοιπόν 4374 σετ τιμών της CDF, της πρώτης και δεύτερης παραγώγου της ως προς  $x$  και της χρονικής παραγώγου της για την εφαρμογή Γκ.Δ. Επιλέγεται πάλι η συνάρτηση Kernel η συνάρτηση squared exponential. Έπειτα υπολογίζονται οι υπερπαραμέτροι των μεταβλητών για τον προσδιορισμό των σημαντικών.

### 2.3.4 Εκπαίδευση Νευρωνικού Δικτύου

Από τις Γκ.Δ. αναδεικνύεται ως σημαντική μεταβλητή η δεύτερη παράγωγος της CDF ως προς τη διεύθυνση  $x$ . Οπότε διαμορφώνεται το πρόβλημα ως εξής:

$$\frac{\partial CDF}{\partial t} = net \left( \frac{\partial^2 CDF}{\partial x^2} \right) \quad (2.13)$$

Πρέπει να τονιστεί πως η CDF είναι το ολοκλήρωμα της λύσης της Δ.Ε. οπότε η δεύτερη παράγωγος της CDF αντιστοιχίζεται με τη πρώτη παράγωγο ως προς τη διεύθυνση  $x$  της λύσης της Δ.Ε.

Η κατασκευή και η αξιολόγηση του δικτύου ακολουθούν την ίδια διαδικασία της ενότητας 2.1.4.

Οι παράμετροι εκπαίδευσης του νευρωνικού δικτύου για το πρόβλημα των τυχαίων βηματιστών:

Πίνακας 2.5: Δεδομένα εκπαίδευσης νευρωνικών δικτύων για το πρόβλημα των τυχαίων βηματιστών

Επιλογή δεδομένων εισόδου	Τυχαία
Ποσοστό δεδομένων εκπαίδευσης	70%
Ποσοστό δεδομένων αξιολόγησης	15%
Ποσοστό δεδομένων τεστ	15%
Αλγόριθμος εκπαίδευσης	Levenberg Marquardt
Αλγόριθμος επίδοσης	Mean Squared Error (MSE)
Μέγιστος αριθμός επαναλήψεων εκπαίδευσης	2000
Συνάρτηση μεταφοράς εσωτερικών νευρώνων	Tansig
Συνάρτηση μεταφοράς εξωτερικών νευρώνων	Purelin

Κατά την επιλογή των αρχιτεκτονικών που εξετάζονται για το πρόβλημα των τυχαίων βηματιστών πρέπει να ληφθεί υπ' όψει η αδυναμία του μοντέλου, δηλαδή ο θόρυβος που περιλαμβάνεται στα δεδομένα των υπολογισμών. Για αυτό εξετάζονται δομές ρηχών αρχιτεκτονικών δικτύων, μέχρι 2 στρωμάτων, παρόμοιας περιπλοκότητας με αυτά που εξετάστηκαν στα προβλήματα της διάχυσης και Bratu.

Πίνακας 2.6: Δομές νευρωνικών δικτύων που εκπαιδεύτηκαν και εξετάστηκαν για την εύρεση της βέλτιστης αρχιτεκτονικής για το πρόβλημα των τυχαίων βηματιστών

networks	layers	neurons-1	neurons-2
net-1	1	1	
net-2	1	3	
net-3	1	5	
net-4	1	7	
net-5	2	1	1
net-6	2	1	3
net-7	2	1	5
net-8	2	1	7
net-9	2	2	1
net-10	2	3	1

Για κάθε αρχιτεκτονική που εξετάζεται γίνονται 10 διαδοχικές εκπαιδεύσεις και διατηρείται αυτή με το χαμηλότερο MSE. Έπειτα τα δίκτυα εξετάζονται στη προσαρμογή τους στις καινούργιες κατανομές για χρόνους 0,1 και 0,5 με τους δείκτες αξιολόγησης AUF και MAPE. Τελικά επιλέγεται η αρχιτεκτονική που παρουσίασε τις χαμηλότερες τιμές σε όλους τους δείκτες αξιολόγησης στα τεστ.

## 2.4 Εφαρμογή σε περιβάλλον MATLAB

### 2.4.1 Επίλυση Δ.Ε. με την ode15s/μέθοδο πεπερασμένων διαφορών

Στα παραρτήματα 6.3 και 6.4 παρουσιάζεται ο κώδικας MATLAB για την επίλυση των Δ.Ε. της μοριακής διάχυσης και του προβλήματος Bratu.

Αρχικά γίνεται η διακριτοποίηση του χωρίου, σε 101 κόμβους και ορίζεται το χρονικό διάστημα όπως περιγράφηκε στις ενότητες 2.1.2 και 2.2.2. Έπειτα ορίζονται οι αρχικές κατανομές και οι μεταβλητές  $D$  και  $\lambda$ . Οι αρχικές κατανομές είναι κοινές και για τα δύο προβλήματα και οι μεταβλητές διατηρούνται σταθερές και η ανάλυση πραγματοποιείται για συγκεκριμένη τιμή τους.

Η επίλυση των δύο προβλημάτων γίνεται με την χρήση της εντολής `ode15`, εσωτερική εντολή επίλυσης Δ.Ε. του MATLAB. Δέχεται ως ορίσματα της Δ.Ε. προς επίλυση, το χρονικό διάστημα επίλυσης και τις αρχικές κατανομές. Η Δ.Ε. προς επίλυση ορίζεται σε ξεχωριστό function το οποίο καλείται από την `ode15s`.

Τα αποτελέσματα που δίνει η εντολή `ode15s` έχουν την μορφή ενός πίνακα, διαστάσεων  $m \times n$ , όπου  $m$  είναι οι χρόνοι στους οποίους η εντολή δίνει την επίλυση της Δ.Ε. και  $n$  είναι οι κόμβοι στους οποίους υπολογίζονται οι λύσεις. Ο αριθμός  $n$  είναι ίσος με 101 καθώς αυτή η διακριτοποίηση έχει διατηρηθεί. Ο αριθμός  $m$  αλλάζει σε κάθε διαφορετική κατανομή καθώς προκύπτει αυτόματα από την εντολή `ode15s`. Οπότε για κάθε αρχική κατανομή προκύπτει ένας πίνακας λύσεων όπου κάθε στήλη αναπαριστά ένα κόμβο και αποθηκεύει τις τιμές που λαμβάνει η λύση σε αυτών ενώ κάθε γραμμή αναπαριστά τους χρόνους στους οποίους υπολογίζονται οι λύσεις και αποθηκεύει τις τιμές της λύσης για όλους τους κόμβους στον χρόνο αυτό.

Επόμενο βήμα είναι η προσέγγιση της πρώτης και δεύτερης χωρικής παραγώγου της λύσης ως προς τη διεύθυνση  $x$ . Ορίζεται function στο οποίο γίνεται αριθμητική προσέγγιση των τιμών των παραγώγων και καλείται από τον κύριο κώδικα για τον υπολογισμό των παραγώγων για τις λύσεις που προκύπτουν από κάθε αρχική κατανομή.[36]

#### 2.4.2 Επίλυση μοντέλου τυχαίων βηματιστών

Στο παράρτημα 6.5 παρουσιάζεται ο κώδικας MATLAB για τη επίλυση του μοντέλου των τυχαίων βηματιστών.

Το μοντέλο των τυχαίων βηματιστών επιλύεται με τις ίδιες αρχικές κατανομές με τα προβλήματα Δ.Ε.. Αυτό γίνεται προκειμένου να είναι τα αποτελέσματα συγκρίσιμα και επαληθεύσιμα. Γίνεται λοιπόν μετατροπή των αρχικών κατανομών ώστε τα δεδομένα της συγκέντρωσης να μεταφραστούν σε θέσεις βηματιστών.

Έπειτα υπολογίζονται οι λύσεις μέσω της επαναληπτικής διαδικασίας που υπολογίζει τις ανανεωμένες θέσεις των βηματιστών. Υπολογίζεται η CDF της λύσεις για κάθε κατανομή και έπειτα προσεγγίζονται αριθμητικά η πρώτη και δεύτερη χωρική παράγωγός της στη διεύθυνση  $x$  και η χρονική της παράγωγος.

Το φιλτράρισμα των αποτελεσμάτων για μείωση του θορύβου γίνεται με την εντολή του MATLAB, `sgolayfilt`. Αυτή η εντολή εφαρμόζει ένα φίλτρο εξομάλυνσης Savitzky-Golay finite impulse response (FIR) στα δεδομένα με παραμέτρους την πολυωνυμική τάξη του και το μήκος κάθε frame όπου εφαρμόζεται.

#### 2.4.3 Εφαρμογή διεργασιών Γκάους

Στο παράρτημα 6.6 παρουσιάζεται ο κώδικας για την εφαρμογή Γκ.Δ.

Για την εφαρμογή Γκ.Δ. χρησιμοποιείται η εσωτερική εντολή του MATLAB, `fitgrp`. Ως όρισμα εισόδου δέχεται τα δεδομένα που εξετάζονται ως προς την συσχέτισή τους με την πρώτη χρονική παράγωγο της λύσης. Τα δεδομένα της χρονικής παραγώγου ορίζονται ως δεδομένα απόκρισης. Επίσης προσδιορίζεται η συνάρτηση Kernel. Στα συγκεκριμένα προβλήματα έχει επιλεγεί η `squared exponential` με διαφορετικές υπερπαραμέτρους για κάθε μεταβλητή, η οποία επιλέγεται με την εντολή `"ardsquaredexponential"`.



.Για τα δεδομένα εισόδου σχηματίζεται ο πίνακας `mat`, διαστάσεων 1200x3 με τα δεδομένα της λύσης και της πρώτης και της δεύτερης παραγώγου αυτής ως προς τη διεύθυνση  $x$ .

Η δειγματοληψία αυτών των δεδομένων ακολουθεί την διαδικασία που περιγράφεται στην ενότητα 2.1.3. Συγκεκριμένα, με χρήση της εντολής `randperm`, γίνεται τυχαία επιλογή 20 κόμβων από τους 40 κεντρικούς κόμβους του χωρίου (κόμβοι 30 έως 70). Σε αυτούς τους κόμβους θα ληφθούν οι τιμές  $[u \ u_x \ u_{xx} \ u_t]$ . Οι χρονικές τιμές που γίνεται η δειγματοληψία επιλέγονται συναρτήσει λογαριθμικής κλίμακας. Τα δεδομένα που λαμβάνονται διατάσσονται με τέτοιο τρόπο ώστε κάθε γραμμή του πίνακα `mat` να περιλαμβάνει δεδομένα που λήφθηκαν από τον ίδιο κόμβο την ίδια χρονική στιγμή.

Η αξιολόγηση της προσαρμογής των δεδομένων του πίνακα `mat` στα δεδομένα της χρονική παραγώγου για την εύρεση των σημαντικών μεταβλητών του προβλήματος γίνεται μέσω των παραμέτρων `Kernel` κάθε μεταβλητής. Τα δεδομένα κάθε στήλης του πίνακα `mat` αντιστοιχούν σε μία μεταβλητή και μετά την προσαρμογή του μοντέλου υπολογίζονται οι παράμετροι `Kernel` για κάθε μια. Όσο μεγαλύτερη τιμή έχει η παράμετρος `Kernel` που υπολογίζεται για μία μεταβλητή συγκριτικά με τις υπόλοιπες τόσο χαμηλότερη συγγένεια παρουσιάζει με τα δεδομένα απόκρισης.

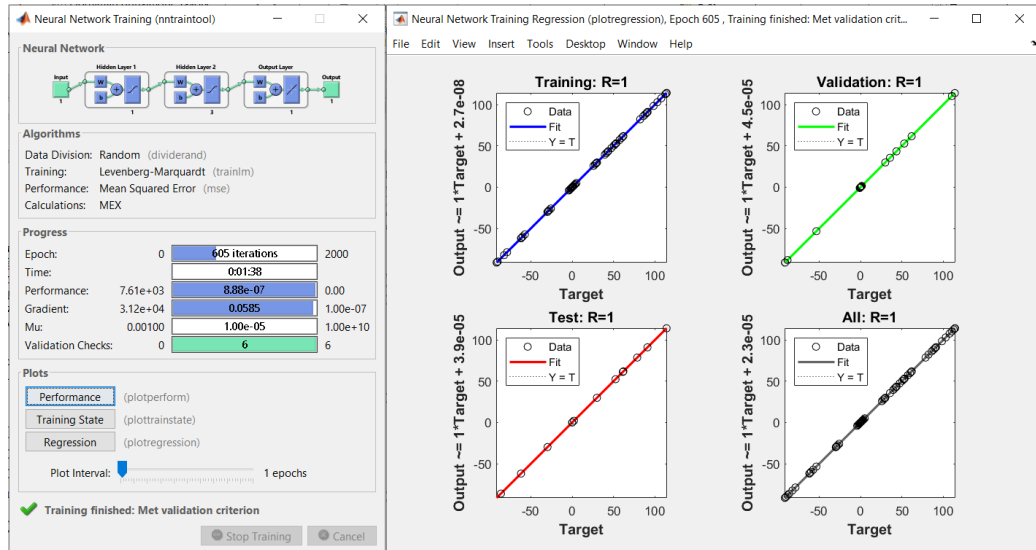
#### 2.4.4 Εκπαίδευση νευρωνικών δικτύων

Ο κώδικας για την εκπαίδευση ρηχών νευρωνικών δικτύων παρουσιάζεται στο παράρτημα 6.7.

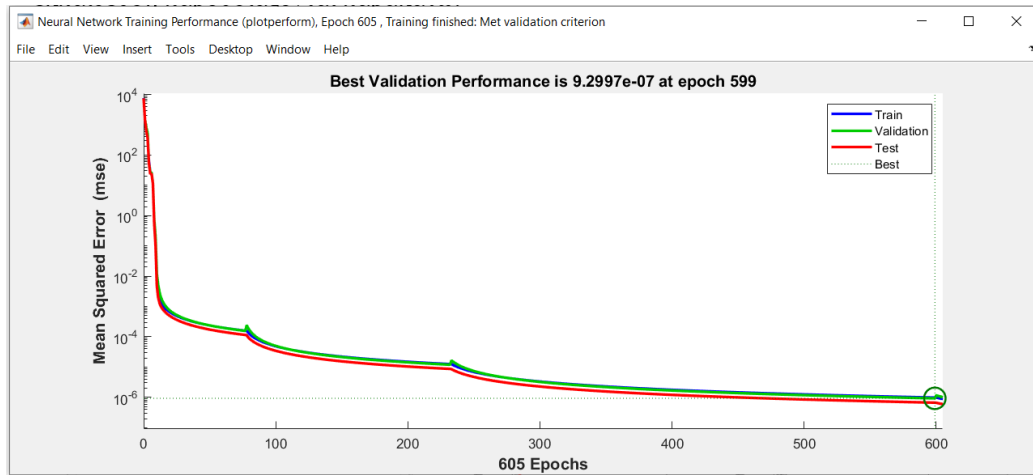
Ο κώδικας εκπαίδευσης νευρωνικών δικτύων παράγεται από το MATLAB και γίνονται επ' αυτού οι απαραίτητες αλλαγές. Η εκπαίδευση των νευρωνικών δικτύων έγινε μέσω της εφαρμογής `Neural Net Fitting` της εργαλειοθήκης `Neural Network Toolbox` του MATLAB. Καλώντας την εντολή `nnstart` ανοίγει επιφάνεια διεπαφής (GUI) μέσω της οποίας ο χρήστης έχει τη δυνατότητα να ρυθμίσει τις απαραίτητες μεταβλητές εκπαίδευσης των νευρωνικών δικτύων και να παρακολουθήσει την πορεία εκπαίδευσης. Ορίζονται οι μεταβλητές εκπαίδευσης που έχουν οριστεί στις ενότητες 2.1.4 και 2.2.4.

Η διεπιφάνεια επαφής που ανοίγει και τα διαγράμματα μέσω των οποίων παρακολουθείται η εκπαίδευση παρουσιάζονται παρακάτω (Εικόνα 2.3):

(A)



(B)



Εικόνα 2.3: Διεπιφάνεια επαφής και διαγράμματα εκπαίδευσης νευρωνικών δικτύων. 2.3-A (αριστερά): Διεπιφάνεια επαφής νευρωνικού δικτύου, Εικόνα 2.3-A (δεξιά): Διαγράμματα παλινδρόμησης στα δεδομένα εκπαίδευσης, επαλήθευσης, τεστ και συνολικά, Εικόνα 2.3-B (κάτω): Διάγραμμα σφάλματος κατά την εκπαίδευση για τα 3 σετ δεδομένων

Ο κώδικας για την αξιολόγηση των νευρωνικών δικτύων για τα προβλήματα Δ.Ε. σύμφωνα με την μεθοδολογία που περιγράφεται στις ενότητες 2.1.4 και 2.2.4 παρουσιάζεται στο παράρτημα 6.8.

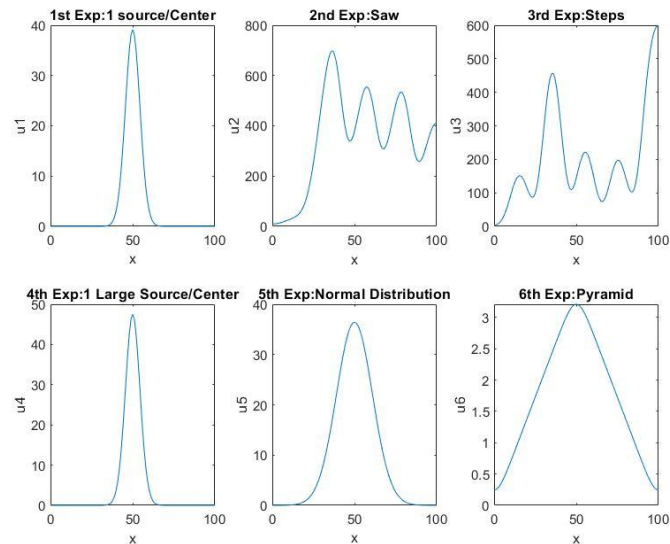
Ο κώδικας για την αξιολόγηση των νευρωνικών δικτύων για το μοντέλο των τυχαίων βηματιστών παρουσιάζεται στο παράρτημα 6.9.

### 3 Κεφάλαιο 3<sup>ο</sup>: Παρουσίαση Αποτελεσμάτων και Σχολιασμός

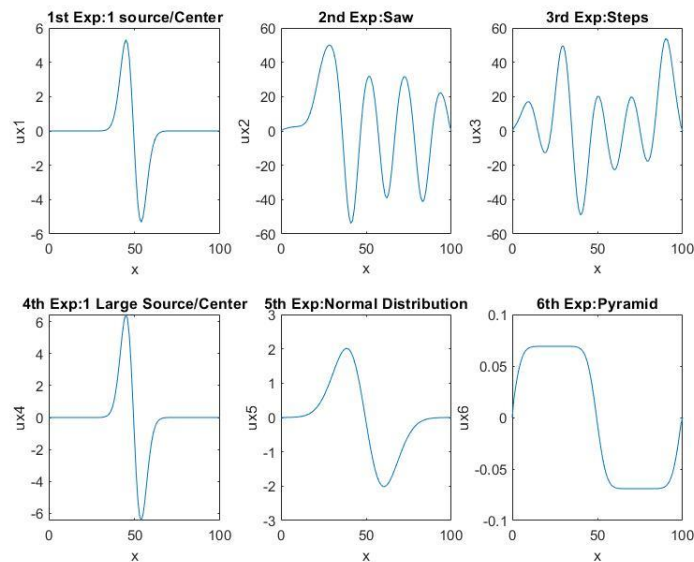
#### 3.1 Αποτελέσματα 1-D διάχυσης

##### 3.1.1 Αποτελέσματα επίλυσης της Δ.Ε. μοριακής διάχυσης

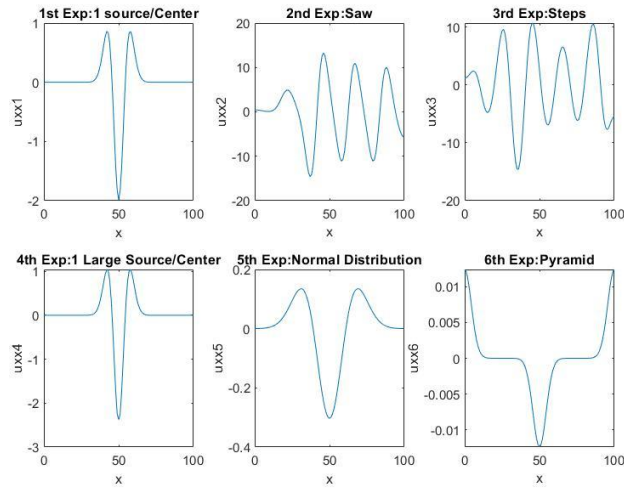
Χρησιμοποιώντας τις αρχικές κατανομές της παραγράφου 2.1.1 παράγονται οι λύσεις της Δ.Ε. από την εντολή ode15. Ενδεικτικά σε χρόνο 100 παράγονται οι εξής λύσεις (Εικόνα 3.1-4):



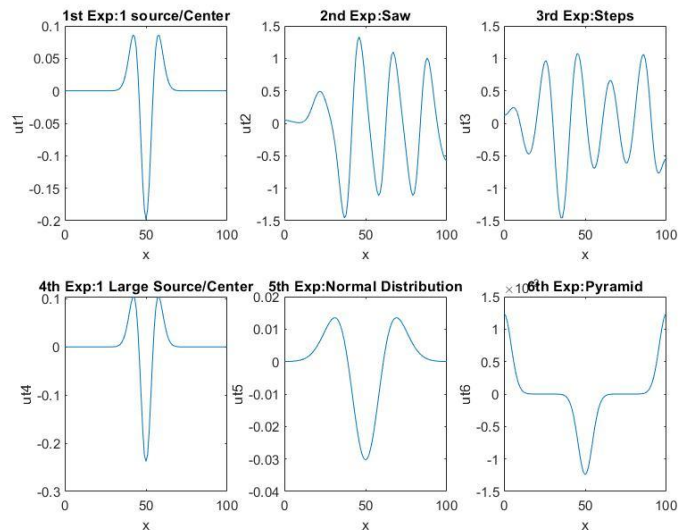
Εικόνα 3-1: Αποτελέσματα επίλυσης της Δ.Ε. μοριακής διάχυσης για χρόνο  $t=100$ , με συντελεστή διάχυσης  $D=0,1$  για κάθε αρχική κατανομή



Εικόνα 3-2: Πρώτη χωρική παράγωγος ως προς τη διεύθυνση  $x$  της λύσης της Δ.Ε. μοριακής διάχυσης για χρόνο  $t=100$ , με συντελεστή διάχυσης  $D=0,1$



Εικόνα 3-3: Δεύτερη χωρική παράγωγος ως προς τη διεύθυνση  $x$  της λύσης της Δ.Ε. μοριακής διάχυσης για χρόνο  $t=100$ , με συντελεστή διάχυσης  $D=0,1$

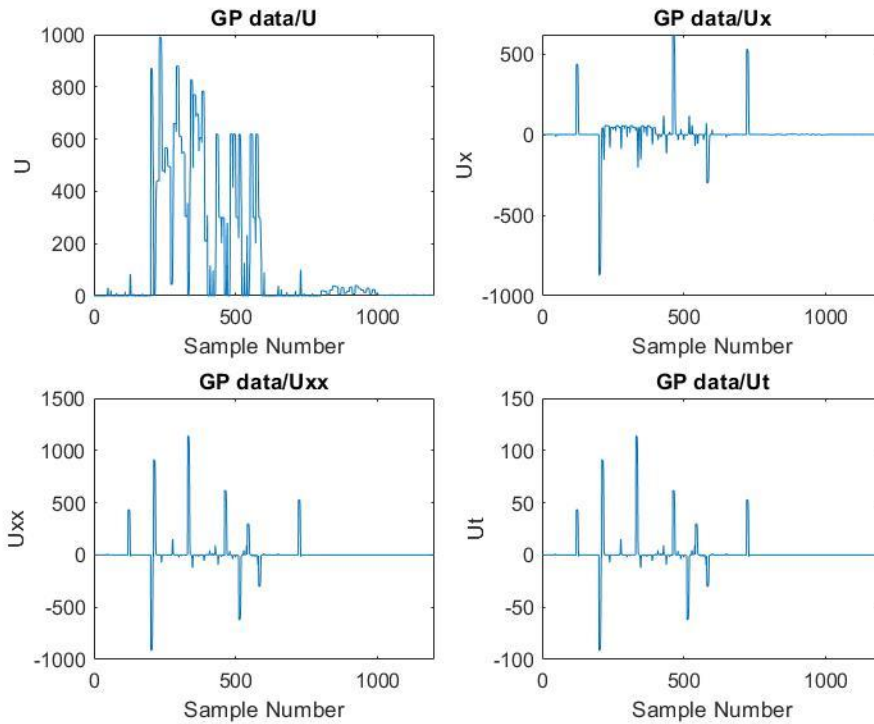


Εικόνα 3-4: Χρονική παράγωγος της λύσης της Δ.Ε. μοριακής διάχυσης για χρόνο  $t=100$ , με συντελεστή διάχυσης  $D=0,1$

Είναι εμφανές και από τα διαγράμματα ότι η δεύτερη παράγωγος της λύσης ως προς τη διεύθυνση  $x$  και η χρονική παράγωγος της λύσης παρουσιάζουν πολύ μεγάλη ομοιότητα ως προς τη μορφολογία, με διαφορά στην κλίμακα.

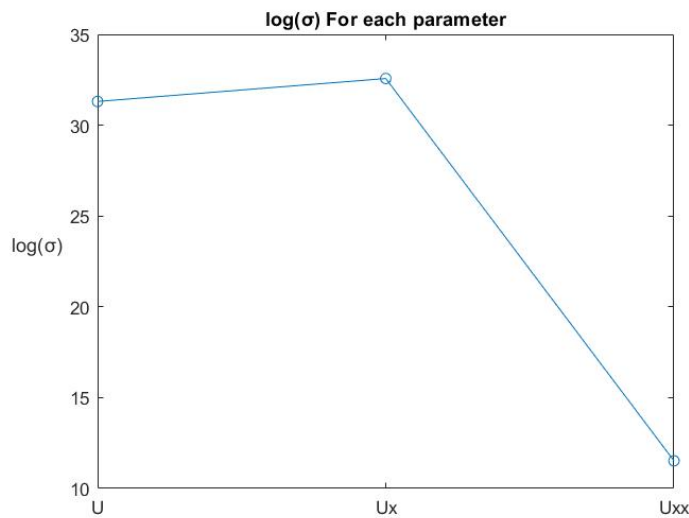
### 3.1.2 Αποτελέσματα Γκαουσιανών Διεργασιών

Παρουσιάζονται τα δεδομένα τα οποία χρησιμοποιήθηκαν για την εφαρμογή Γκ.Δ στο πρόβλημα της μοριακής διάχυσης (Εικόνα 3.5):



Εικόνα 3-5: Τα δεδομένα που χρησιμοποιήθηκαν από την λύση του προβλήματος μοριακής διάχυσης και τις παραγώγους του για την εφαρμογή Γκαουσιανών Διεργασιών

Στη παρακάτω εικόνα (Εικόνα 3.6) φαίνονται οι λογάριθμοι των τιμών των παραμέτρων Kernel που υπολογίζονται από την εφαρμογή Γκ.Δ στα παραπάνω δεδομένα:

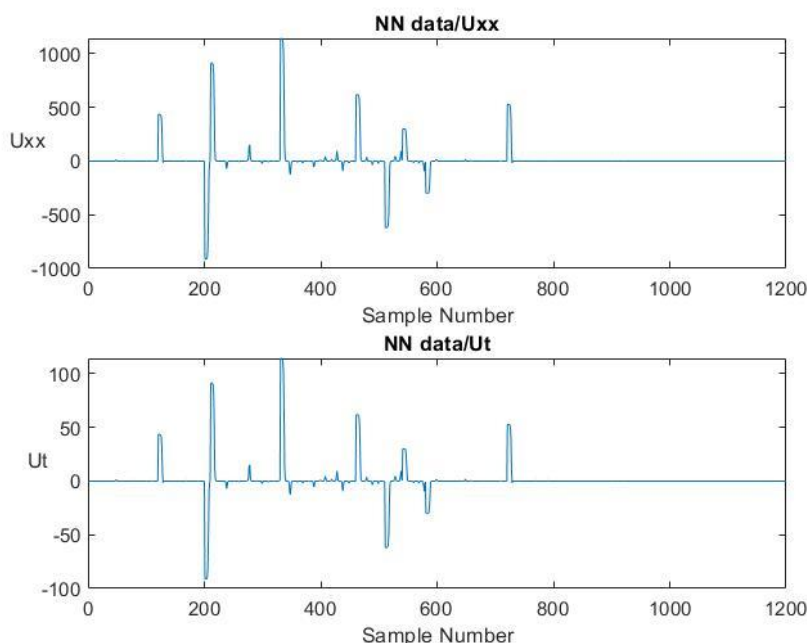


Εικόνα 3-6: Λογάριθμοι των τιμών των παραμέτρων Kernel για κάθε μεταβλητή για το πρόβλημα μοριακής διάχυσης

Όπως φαίνεται ξεκάθαρα στο παραπάνω διάγραμμα, αναγνωρίζεται η δεύτερη παράγωγος της λύσης ως προς τη διεύθυνση  $x$  ως η μοναδική σημαντική μεταβλητή του προβλήματος. Παρουσιάζει σχεδόν 20 τάξεις μεγέθους διαφορά από τις άλλες μεταβλητές του προβλήματος, με αυτή να έχει την μικρότερη τιμή.

### 3.1.3 Αποτελέσματα και προσαρμογή νευρωνικού δικτύου

Για την εκπαίδευση του νευρωνικού δικτύου χρησιμοποιήθηκαν τα ίδια δεδομένα με αυτά που χρησιμοποιήθηκαν για την εφαρμογή Γκ.Δ., για τις σημαντικές μεταβλητές (Εικόνα 3.7):



Εικόνα 3-7: Δεδομένα εκπαίδευσης νευρωνικού δικτύου για την επίλυση του προβλήματος της μοριακής διάχυσης

Όπως περιγράφεται στην ενότητα 2.1.4 εξετάζονται 21 διαφορετικές αρχιτεκτονικές ως προς την προσαρμογή τους στα δεδομένα εκπαίδευσης και σε νέες αρχικές κατανομές σε χρόνους πολύ μεγαλύτερους από τον χρόνο εκπαίδευσης με βάση τους δείκτες αξιολόγησης AUF και MAPE.

Παρακάτω παρουσιάζονται οι πίνακες αξιολόγησης των νευρωνικών δικτύων:

Πίνακας 3.1: Δεδομένα αξιολόγησης νευρωνικών δικτύων στο πρόβλημα της μοριακής διάχυσης/ Δείκτης AUF

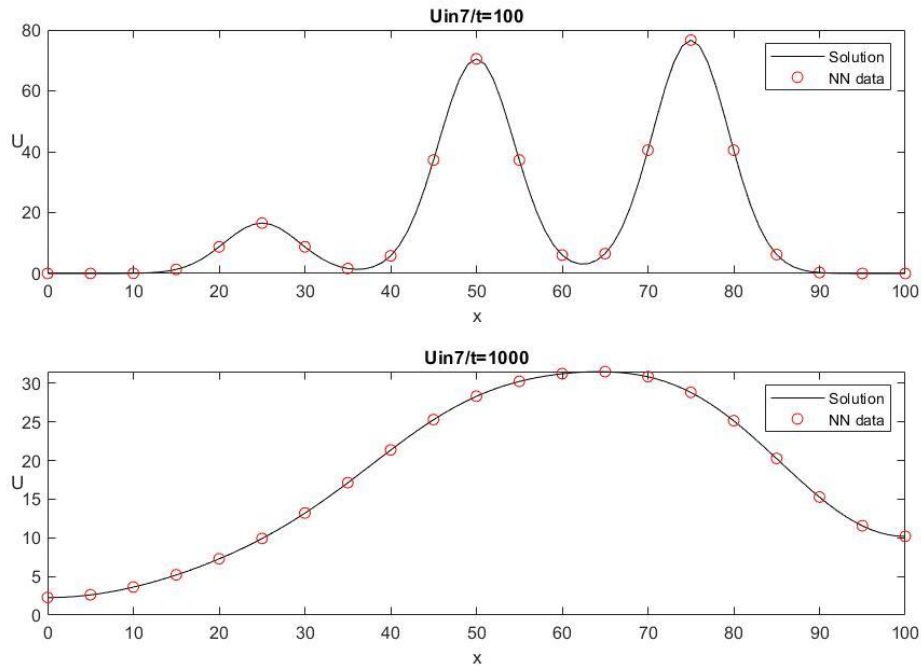
networks	layers	neurons-1	neurons-2	neurons3	MSE-Mean	MSE-Best	STDV	Area under points			
								U_in_7/t100	U_in_7/t1000	U_in_8/t100	U_in_8/t1000
net-1	1	1			6,68E-04	7,33E-05	5,55E-04	6,50E-03	6,91E-02	1,00E-04	8,00E-04
net-2	1	2			1,18E-03	1,47E-05	2,15E-03	3,70E-03	3,96E-02	7,00E-04	4,00E-04
net-3	1	5			2,28E-04	2,31E-08	3,47E-04	3,21E-05	9,04E-04	9,00E-04	1,00E-04
net-4	1	7			7,46E-04	2,09E-08	2,04E-03	<b>2,36E-05</b>	2,57E-04	2,00E-07	2,40E-06
net-5	1	10			2,15E+00	3,12E-03	2,33E+00	8,00E-04	7,50E-03	7,78E-06	8,17E-05
net-6	2	1	1		1,63E+02	5,63E-04	2,94E+02	8,58E-03	9,03E-02	9,80E-05	9,89E-04
net-7	2	1	3		<b>9,48E-05</b>	<b>3,284E-10</b>	<b>1,80E-04</b>	7,58E-05	<b>9,94E-05</b>	<b>9,00E-08</b>	<b>9,60E-07</b>
net-8	2	1	5		1,88E-03	4,89E-06	4,25E-03	7,73E-04	7,43E-03	8,12E-06	8,14E-05
net-9	2	2	1		2,71E-03	3,04E-06	7,08E-03	3,13E-04	3,87E-03	3,87E-06	3,92E-05
net-10	2	2	2		2,20E-03	1,49E-06	5,04E-03	5,28E-04	5,18E-03	5,54E-06	5,67E-05
net-11	2	3	1		2,12E+02	2,76E-06	3,79E+02	3,12E-03	3,35E-02	3,60E-05	3,65E-04
net-12	3	1	1	1	2,46E-03	4,39E-04	2,63E-03	1,38E+01	1,39E+02	1,51E-01	1,52E+00
net-13	3	1	2	2	1,15E+02	2,06E-05	1,95E+02	2,14E-04	2,82E-03	2,92E-06	3,01E-05
net-14	3	1	3	3	3,50E+01	3,50E-07	1,00E+02	2,51E-03	2,65E-02	2,85E-05	2,89E-04
net-15	3	1	2	3	5,17E-03	6,91E-06	8,41E-03	6,76E+01	6,81E+02	7,42E-01	7,43E+00
net-16	3	2	2	2	1,24E+02	6,09E-08	2,28E+02	5,91E-01	6,05E+00	6,10E-01	6,05E+00
net-17	3	2	3	1	1,03E+02	5,19E-06	2,08E+02	1,94E-03	1,97E-02	2,13E-05	2,16E-04
net-18	3	2	3	3	1,83E+02	1,50E-06	3,00E+02	1,53E-03	1,51E-02	1,64E-05	1,68E-04
net-19	3	2	3	5	8,67E+00	4,70E-08	2,68E+01	5,44E-05	6,56E-04	1,54E-07	2,06E-06
net-20	3	1	3	5	1,62E+00	2,11E-07	2,68E+01	5,06E-04	4,97E-03	5,25E-06	5,39E-05
net-21	3	2	5	3	2,51E+01	1,52E-07	5,08E+00	3,38E-05	9,83E-04	1,02E-06	1,07E-05

Πίνακας 3.2: Δεδομένα αξιολόγησης νευρωνικών δικτύων στο πρόβλημα της μοριακής διάχυσης / Δείκτης MSE

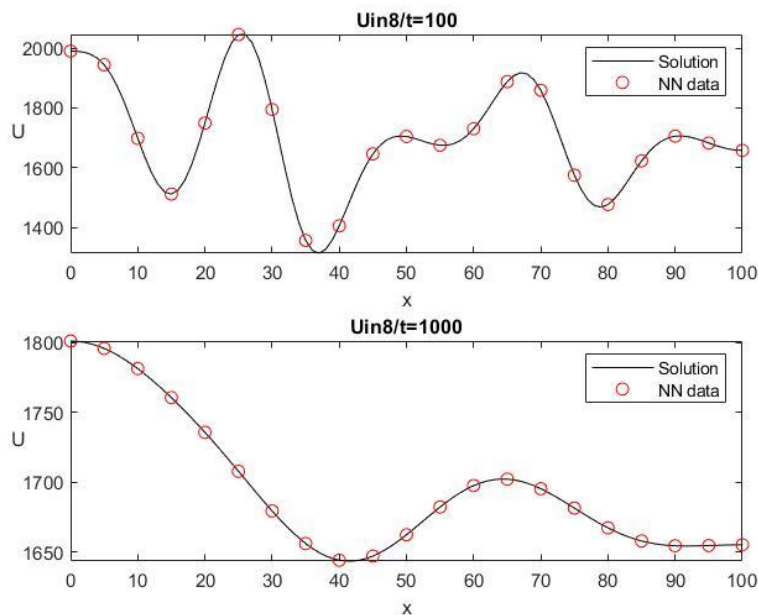
networks	layers	neurons-1	neurons-2	neurons3	MSE-Mean	MSE-Best	STDV	MAPE			
								U_in_7/t100	U_in_7/t1000	U_in_8/t100	U_in_8/t1000
net-1	1	1			6,68E-04	7,33E-05	5,55E-04	2,50E-03	5,41E-02	1,00E-04	8,00E-04
net-2	1	2			1,18E-03	1,47E-05	2,15E-03	1,30E-03	3,10E-02	3,00E-04	4,00E-04
net-3	1	5			2,28E-04	2,31E-08	3,47E-04	<b>4,47E-05</b>	7,08E-04	1,00E-06	1,01E-05
net-4	1	7			7,46E-04	2,09E-08	2,04E-03	5,89E-05	2,16E-04	2,00E-04	2,40E-06
net-5	1	10			2,15E+00	3,12E-03	2,33E+00	3,29E-04	5,83E-03	8,14E-06	8,25E-05
net-6	2	1	1		1,63E+02	5,63E-04	2,94E+02	3,32E-03	7,07E-02	1,04E-04	9,99E-04
net-7	2	1	3		<b>9,48E-05</b>	<b>3,284E-10</b>	<b>1,80E-04</b>	9,27E-05	<b>7,36E-05</b>	<b>1,00E-07</b>	<b>9,70E-07</b>
net-8	2	1	5		1,88E-03	4,89E-06	4,25E-03	3,60E-04	5,81E-03	8,52E-06	8,22E-05
net-9	2	2	1		2,71E-03	3,04E-06	7,08E-03	6,41E-05	3,09E-03	3,58E-06	3,95E-05
net-10	2	2	2		2,20E-03	1,49E-06	5,04E-03	2,46E-04	4,05E-03	5,84E-06	5,72E-05
net-11	2	3	1		2,12E+02	2,76E-06	3,79E+02	1,07E-03	2,62E-02	3,63E-05	3,68E-04
net-12	3	1	1	1	2,46E-03	4,39E-04	2,63E-03	5,00E+00	1,09E+02	1,42E-01	1,53E+00
net-13	3	1	2	2	1,15E+02	2,06E-05	1,95E+02	1,29E-05	2,22E-03	2,65E-06	3,04E-05
net-14	3	1	3	3	3,50E+01	3,50E-07	1,00E+02	8,17E-04	2,07E-02	2,71E-05	2,92E-04
net-15	3	1	2	3	5,17E-03	6,91E-06	8,41E-03	2,61E+01	5,33E+02	7,32E-01	7,49E+00
net-16	3	2	2	2	1,24E+02	6,09E-08	2,28E+02	3,50E-03	3,70E-01	5,50E-03	3,70E-01
net-17	3	2	3	1	1,03E+02	5,19E-06	2,08E+02	7,71E-04	1,53E-02	2,16E-05	2,18E-04
net-18	3	2	3	3	1,83E+02	1,50E-06	3,00E+02	5,95E-04	1,17E-02	1,60E-05	1,70E-04
net-19	3	2	3	5	8,67E+00	4,70E-08	2,68E+01	6,57E-05	6,47E-04	2,56E-07	2,09E-06
net-20	3	1	3	5	1,62E+00	2,11E-07	2,68E+01	2,37E-04	3,89E-03	5,53E-06	5,44E-05
net-21	3	2	5	3	2,51E+01	1,52E-07	5,08E+00	4,02E-05	7,76E-04	1,19E-06	1,08E-05

Στους παραπάνω πίνακς οι στήλες MSE-Mean, MSE-Best και STDV είναι κοινές και αφορούν την προσαρμογή των νευρωνικών δικτύων στα δεδομένα εκπαίδευσης. Παρουσιάζουν τα δεδομένα για το μέσο και βέλτιστο MSE και την τυπική απόκλιση των τιμών του MSE που προκύπτουν κατά τις 10 διαδοχικές επαναλήψεις εκπαίδευσης κάθε αρχιτεκτονικής.

Στους πίνακες έχουν σημειωθεί με Bold οι βέλτιστες τιμές κάθε στήλης για το προσδιορισμό της βέλτιστης αρχιτεκτονικής. Όπως φαίνεται αναδεικνύεται το net-7 ως το βέλτιστο, με 2 κρυμμένα στρώματα και 1 νευρώνα στο πρώτο και 3 στο δεύτερο. Παρακάτω παρουσιάζεται η προσαρμογή του 7<sup>ου</sup> δικτύου στις κατανομές αξιολόγησης και διαγραμματικά (Εικόνα 3.5):



Εικόνα 3-8: Προσαρμογή βέλτιστου νευρωνικού δικτύου στην 7<sup>η</sup> αρχική κατανομή σε χρόνους  $t=100, 1000$  για συντελεστή διάχυσης  $D=0,1$ . Με μαύρες συνεχείς γραμμές απεικονίζονται οι λύσεις που υπολογίζονται σε κάθε χρόνο. Με κόκκινους κύκλους απεικονίζονται οι τιμές που υπολογίζονται από το νευρωνικό δίκτυο σε κάθε κόμβο.



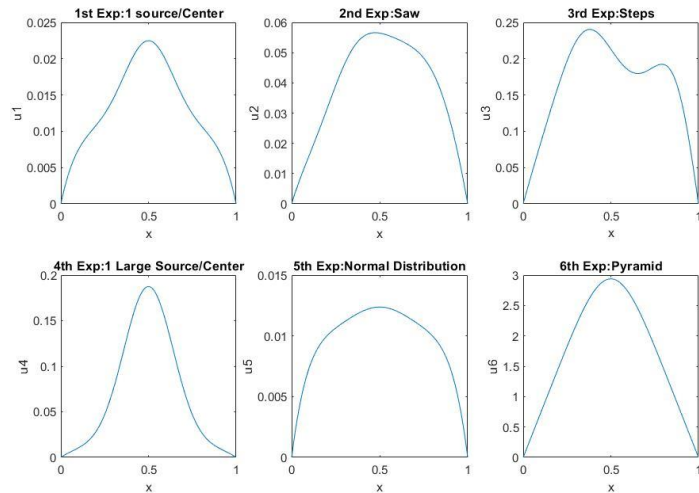
Εικόνα 3-9: Προσαρμογή βέλτιστου νευρωνικού δικτύου στην 8<sup>η</sup> αρχική κατανομή σε χρόνους  $t=100, 1000$  για συντελεστή διάχυσης  $D=0,1$ . Με μαύρες συνεχείς γραμμές απεικονίζονται οι λύσεις που υπολογίζονται σε κάθε χρόνο. Με κόκκινους κύκλους απεικονίζονται οι τιμές που υπολογίζονται από το νευρωνικό δίκτυο σε κάθε κόμβο.

## 3.2 Αποτελέσματα 1-D Bratu

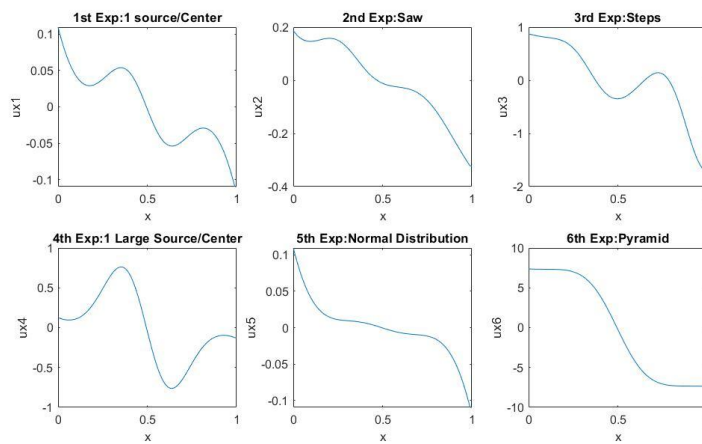
### 3.2.1 Αποτελέσματα επίλυσης της Δ.Ε. του προβλήματος Bratu

Χρησιμοποιώντας τις αρχικές κατανομές της παραγράφου 2.2.1 παράγονται οι λύσεις της Δ.Ε. από την εντολή ode15. Ενδεικτικά σε χρόνο 1 παράγονται οι εξής λύσεις (Εικόνα 3.10-13):

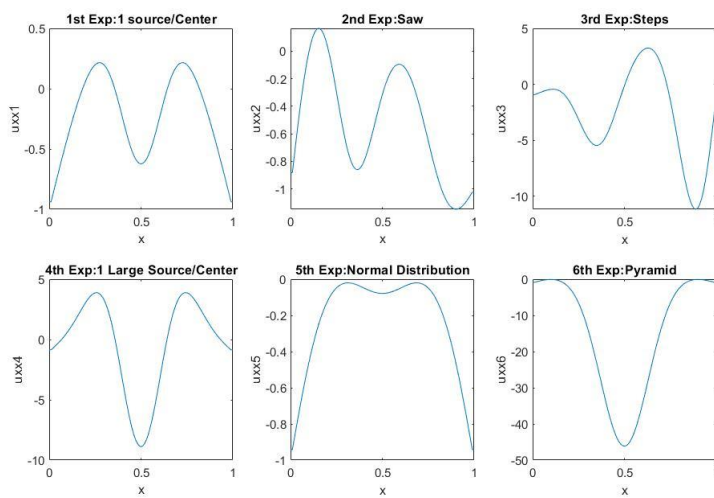




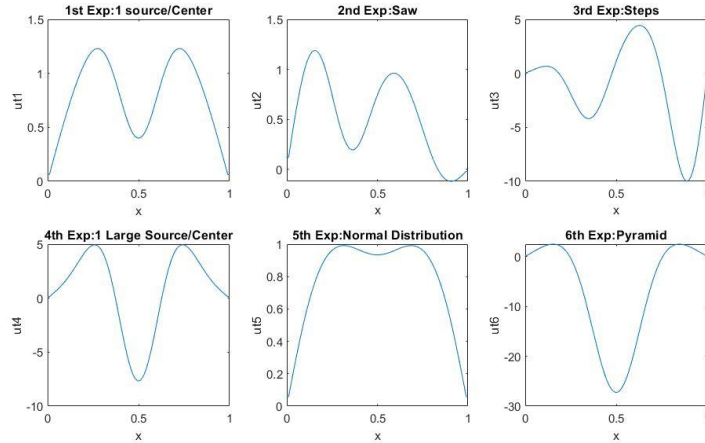
Εικόνα 3-10: Αποτελέσματα επίλυσης του προβλήματος Bratu για χρόνο  $t=1$ , με συντελεστή  $D=1$ ,  $\lambda=1$



Εικόνα 3-11: Πρώτη χωρική παράγωγος ως προς τη διεύθυνση  $x$  της λύσης του προβλήματος Bratu για χρόνο  $t=1$ , με συντελεστή διάχυσης  $D=1$ ,  $\lambda=1$



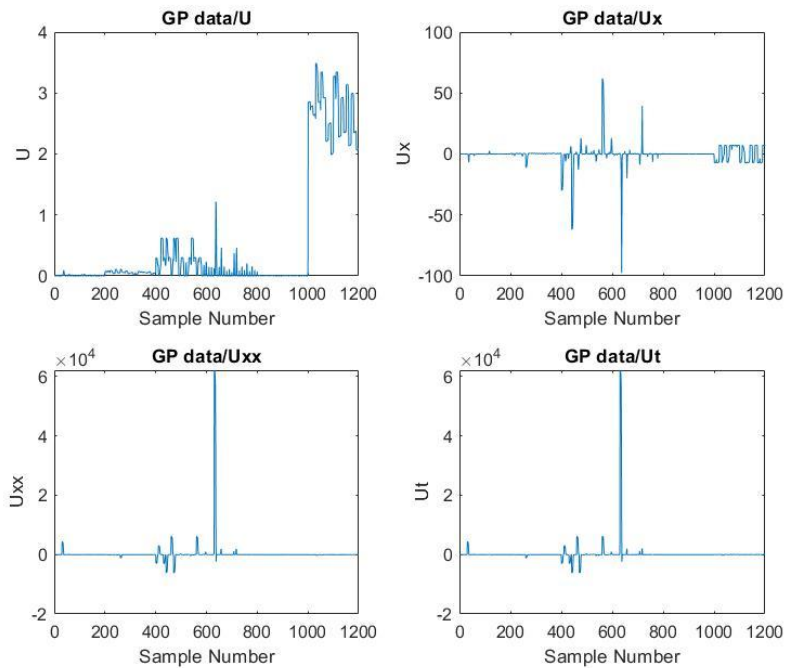
Εικόνα 3-12: Δεύτερη χωρική παράγωγος ως προς τη διεύθυνση  $x$  της λύσης του προβλήματος Bratu για χρόνο  $t=1$ , με συντελεστή διάχυσης  $D=1$ ,  $\lambda=1$



Εικόνα 3-13: Χρονική παράγωγος της λύσης του προβλήματος Bratu για χρόνο  $t=1$ , με συντελεστή διάχυσης  $D=1$ ,  $\lambda=1$

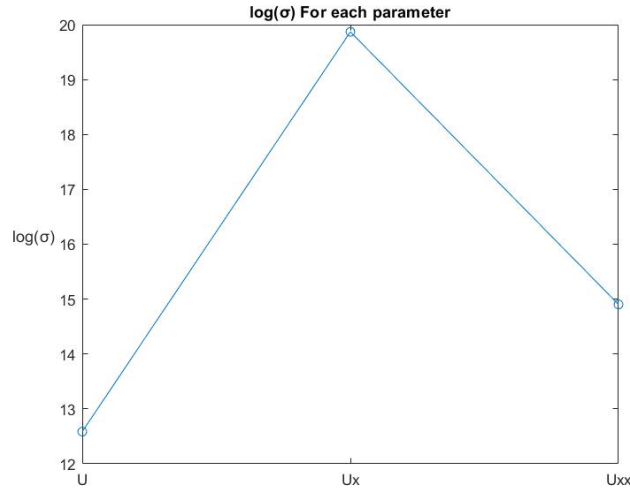
### 3.2.2 Αποτελέσματα Γκαουσιανών Διεργασιών

Παρουσιάζονται τα δεδομένα τα οποία χρησιμοποιήθηκαν για την εφαρμογή Γκ.Δ στο πρόβλημα Bratu (Εικόνα 3.14):



Εικόνα 3-14: Τα δεδομένα που χρησιμοποιήθηκαν από την λύση του προβλήματος Bratu και τις παραγώγους του για την εφαρμογή Γκαουσιανών Διεργασιών

Στη παρακάτω εικόνα (Εικόνα 3.15) φαίνονται οι λογάριθμοι των τιμών των παραμέτρων Kernel που υπολογίζονται από την εφαρμογή Γκ.Δ στα παραπάνω δεδομένα:

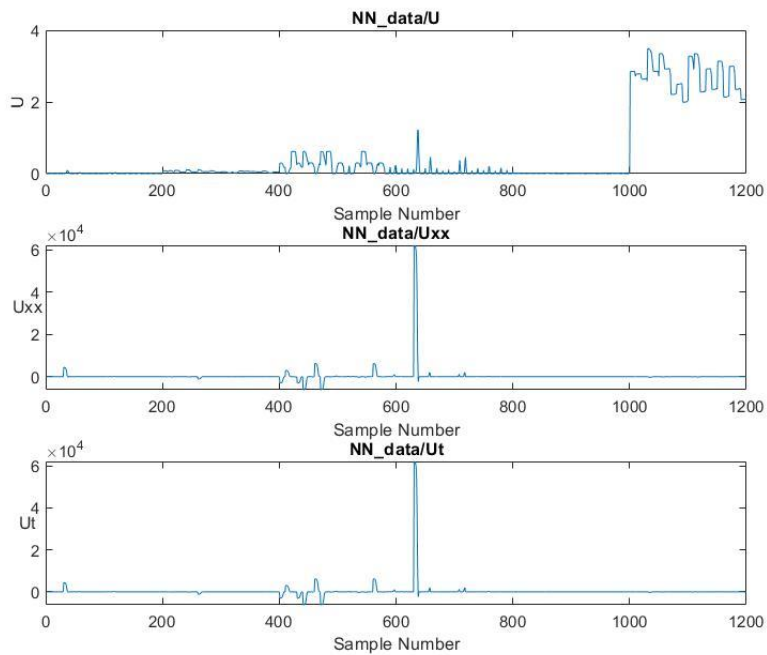


Εικόνα 3-15: Λογάριθμοι των τιμών των παραμέτρων Kernel για κάθε μεταβλητή για το πρόβλημα Bratu

Στο παραπάνω διάγραμμα αναδεικνύονται ως σημαντικές μεταβλητές η λύση του προβλήματος,  $u$ , και η δεύτερη παράγωγός της ως προς τη διεύθυνση  $x$ . Οι τιμές των παραμέτρων είναι λίγο μεγαλύτερες από αυτές που παρατηρήθηκαν στο πρόβλημα της διάχυσης όμως είναι ξεκάθαρη η διαφορά μεγέθους ανάμεσα στις παραμέτρους των μεταβλητών που αναγνωρίζονται ως σημαντικές και στην παράμετρο της πρώτης παραγώγου ως προς την διεύθυνση  $x$ .

### 3.2.3 Αποτελέσματα και προσαρμογή νευρωνικού δικτύου

Για την εκπαίδευση του νευρωνικού δικτύου χρησιμοποιήθηκαν τα ίδια δεδομένα με αυτά που χρησιμοποιήθηκαν για την εφαρμογή Γκ.Δ., για τις σημαντικές μεταβλητές (Εικόνα 3.9):



Εικόνα 3-16: Δεδομένα εκπαίδευσης νευρωνικού δικτύου για την επίλυση του προβλήματος Bratu

Για την αξιολόγηση των αρχιτεκτονικών σε νέες αρχικές κατανομές χρησιμοποιούνται οι ίδιοι δείκτες που χρησιμοποιήθηκαν στην ενότητα 3.1.3.

Οι πίνακες αξιολόγησης των νευρωνικών δικτύων παρουσιάζονται παρακάτω (Πίνακες 3.3-4):

Πίνακας 3.3: Δεδομένα αξιολόγησης νευρωνικών δικτύων στο πρόβλημα Bratu/ Δείκτης AUF

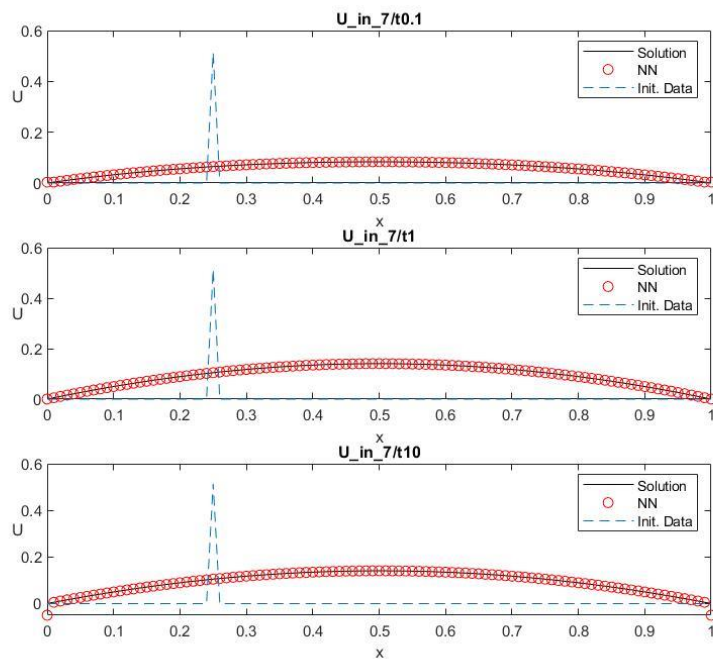
networks	layers	neurons-1	neurons-2	neurons3	MSE-Mean	MSE-Best	STDV	AUF			
								U_in_7/t0,1	U_in_7/t1	U_in_9/t0,1	U_in_9/t1
net-1	1	5			<b>9,51E-02</b>	<b>4,40E-05</b>	<b>1,68E-01</b>	<b>6,19E-05</b>	<b>1,45E-04</b>	3,50E-03	<b>5,40E-06</b>
net-2	1	3			2,55E+00	1,21E-03	4,02E+00	2,40E-03	8,10E-03	<b>3,20E-03</b>	8,10E-03
net-3	2	1	3		5,78E+00	3,11E+00	1,79E+00	2,34E-01	2,99E-01	4,09E-01	5,87E-01
net-4	2	2	3		4,90E+00	3,02E-01	7,61E+00	1,24E-02	2,06E-01	1,19E-01	2,08E-01
net-5	2	3	3		3,43E+03	8,07E-05	1,08E+04	1,30E-03	2,30E-03	<b>3,20E-03</b>	2,40E-03
net-6	1	7			1,95E-01	4,15E-03	4,67E-01	4,30E-03	7,10E-03	1,08E-02	7,20E-03
net-7	2	3	1		4,39E-01	7,85E-04	5,08E-01	8,10E-03	1,92E-02	1,31E-02	1,93E-02

Πίνακας 3.4: Δεδομένα αξιολόγησης νευρωνικών δικτύων πρόβλημα Bratu/ Δείκτης MSE

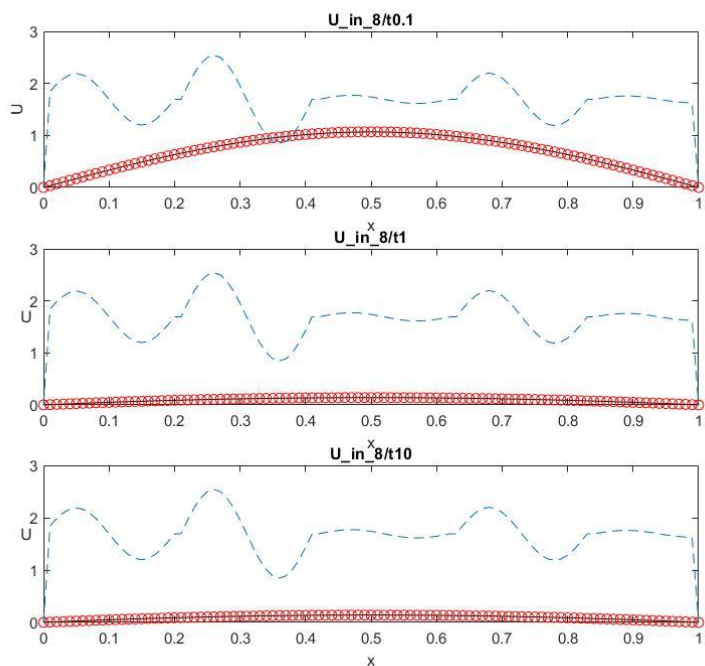
networks	layers	neurons-1	neurons-2	neurons3	MSE-Mean	MSE-Best	STDV	MAPE			
								U_in_7/t0,1	U_in_7/t1	U_in_9/t0,1	U_in_9/t1
net-1	1	5			<b>9,51E-02</b>	<b>4,40E-05</b>	<b>1,68E-01</b>	<b>6,80E-05</b>	<b>1,45E-04</b>	3,60E-03	<b>6,00E-06</b>
net-2	1	3			2,55E+00	1,21E-03	4,02E+00	2,50E-03	8,30E-03	<b>3,20E-03</b>	8,30E-03
net-3	2	1	3		5,78E+00	3,11E+00	1,79E+00	2,34E-01	3,10E-01	4,14E-01	6,05E-01
net-4	2	2	3		4,90E+00	3,02E-01	7,61E+00	1,56E-02	2,12E-01	1,19E-01	2,14E-01
net-5	2	3	3		3,43E+03	8,07E-05	1,08E+04	1,30E-03	2,40E-03	<b>3,20E-03</b>	2,50E-03
net-6	1	7			1,95E-01	4,15E-03	4,67E-01	4,40E-03	7,20E-03	1,10E-02	7,30E-03
net-7	2	3	1		4,39E-01	7,85E-04	5,08E-01	8,30E-03	1,96E-02	1,34E-02	1,97E-02

Όπως και στην ενότητα 3.1.3, οι στήλες MSE-Mean, MSE-Best και STDV είναι κοινές και αφορούν την προσαρμογή των νευρωνικών δικτύων στα δεδομένα εκπαίδευσης. Παρουσιάζουν τα δεδομένα για το μέσο και βέλτιστο MSE και την τυπική απόκλιση των τιμών του MSE που προκύπτουν κατά τις 10 διαδοχικές επαναλήψεις εκπαίδευσης κάθε αρχιτεκτονικής. Με bold έχουν σημαθεί οι βέλτιστες τιμές κάθε στήλης. Αναδεικνύεται η πρώτη αρχιτεκτονική ως βέλτιστη. Επισημαίνεται η ομοιότητα με την συμπεριφορά που παρατηρήθηκε στην εκπαίδευση νευρωνικού δικτύου για το πρόβλημα της μοριακής διάχυσης. Στην προηγούμενη περίπτωση εξαιρετικά καλή προσαρμογή είχε παρουσιάσει το δίκτυο με ένα μόνο στρώμα, με 3 νευρώνες, ενώ τώρα αναδεικνύεται ως βέλτιστη η αρχιτεκτονική με ένα στρώμα με 5 νευρώνες. Αυτό οφείλεται στην ανάγκη για προσαρμογή του δικτύου στον μη γραμμικό όρο. Στην περίπτωση της μοριακής διάχυσης το δίκτυο έπρεπε απλά να αναγνωρίσει τον κατάλληλο συντελεστή D, ενώ τώρα έπρεπε να αναγνωρίσει τον κατάλληλο συνδυασμό της δεύτερης παραγώγου ως προς την διεύθυνση  $\chi$  με την λύση του προβλήματος.

Παρακάτω παρουσιάζεται η προσαρμογή του 1<sup>ου</sup> δικτύου στις κατανομές αξιολόγησης και διαγραμματικά (Εικόνα 3.17):



Εικόνα 3-17: Προσαρμογή βέλτιστου νευρωνικού δικτύου στην 7<sup>η</sup> αρχική κατανομή σε χρόνους  $t=0.1, 1, 10$  για συντελεστή  $D=1, \lambda=1$ . Με μπλε διακεκομμένες γραμμές απεικονίζονται οι αρχικές κατανομές. Με μαύρες συνεχείς γραμμές απεικονίζονται οι λύσεις που υπολογίζονται σε κάθε χρόνο. Με κόκκινους κύκλους απεικονίζονται οι τιμές που υπολογίζονται από το νευρωνικό δίκτυο σε κάθε κόμβο.



Εικόνα 3-18: Προσαρμογή βέλτιστου νευρωνικού δικτύου στην 7<sup>η</sup> αρχική κατανομή σε χρόνους  $t=0.1, 1, 10$  για συντελεστή  $D=1, \lambda=1$ . Με μπλε διακεκομμένες γραμμές απεικονίζονται οι αρχικές κατανομές. Με μαύρες συνεχείς γραμμές απεικονίζονται οι λύσεις που υπολογίζονται σε κάθε χρόνο. Με κόκκινους κύκλους απεικονίζονται οι τιμές που υπολογίζονται από το νευρωνικό δίκτυο σε κάθε κόμβο.

### 3.3 Αποτελέσματα Μοντέλου Τυχαίων Βηματιστών

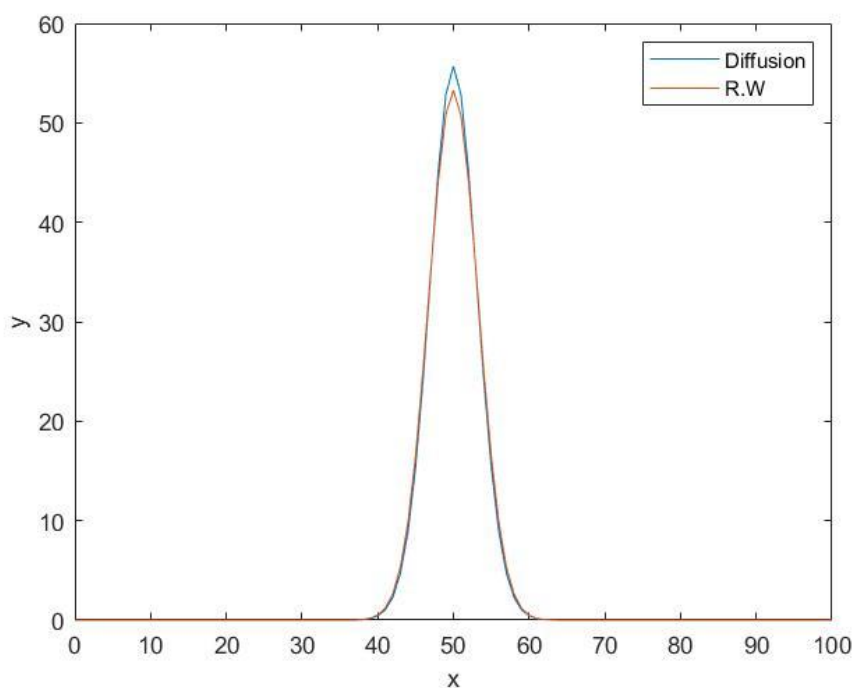
Χρησιμοποιώντας τις αρχικές κατανομές που παρουσιάζονται στην παράγραφο 2.3.1 υπολογίζονται οι CDF των θέσεων των βηματιστών για κάθε κατανομή. Γίνεται διερεύνηση τόσο της ορθότητας του μοντέλου όσο και της συμπεριφοράς του όσον αφορά τον θόρυβο.

#### 3.3.1 Αξιολόγηση ορθότητας των υπολογισμών του μοντέλου τυχαίων βηματιστών

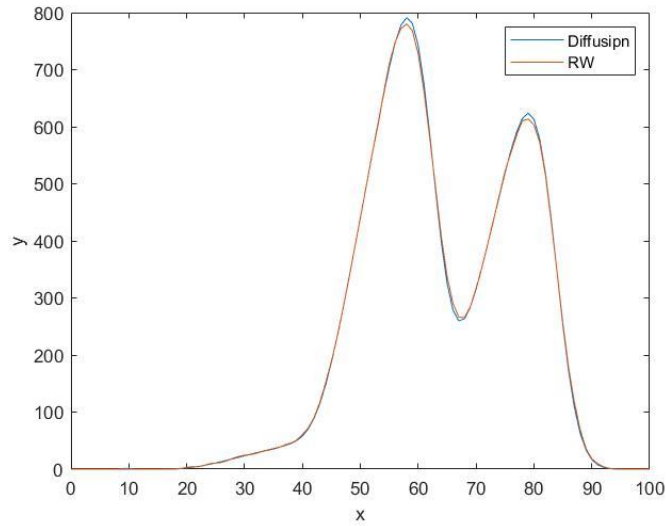
Το μοντέλο των τυχαίων βηματιστών είναι σχεδιασμένο για να απροσμοιώνει το φαινόμενο της μοριακής διάχυσης. Αντιπαραβάλλοντας τα αποτελέσματα του με αυτά της διάχυσης όπως επιλύθηκε στην ενότητα 2.1 μπορεί να αξιολογηθεί η ορθότητα των υπολογισμών.

Για την σύγκριση εφαρμόζονται οι αρχικές συνθήκες της παραγράφου 2.3.1 και στα δύο μοντέλα ( το μοντέλο της διάχυσης επιλύεται με τα δεδομένα της συγκέντρωσης ενώ το μοντέλο των τυχαίων βηματιστών επιλύεται με τα δεδομένα για τις θέσεις των βηματιστών όπως υπολογίζονται από την διαδικασία του lifting). Η επίλυση γίνεται για χρόνο  $t=[0, 0.5]$ , σε χωρίο 101 κόμβων,  $\chi=[0, 100]$ , με συντελεστή διάχυσης  $D=10$ , με πληθυσμό  $10^5$  και με 10 επαναλήψεις για τον υπολογισμό του μέσου όρου σε κάθε μέτρηση.

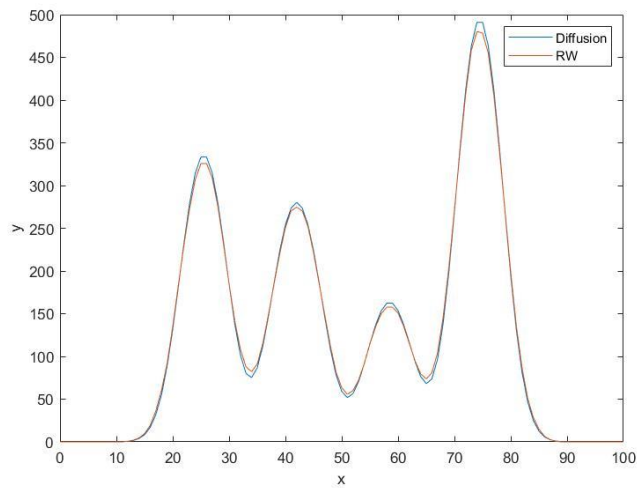
Παρακάτω παρουσιάζονται τα δεδομένα αντιπαραβολής των λύσεων των δύο μοντέλων (Εικόνες 3.19-24):



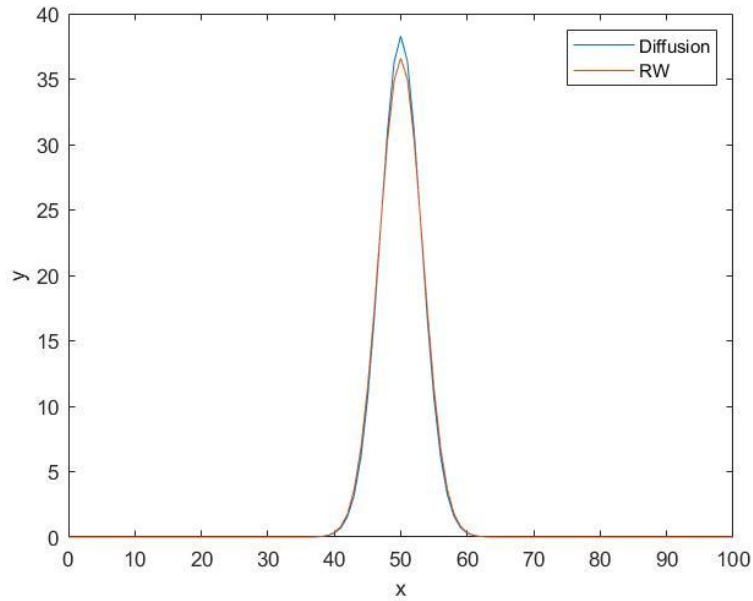
Εικόνα 3.19: Αντιπαραβολή των δεδομένων επίλυσης του μοντέλου τυχαίων βηματιστών με το μοντέλο της μοριακής διάχυσης, δεδομένα  $y$ . Επίλυση για την 1<sup>η</sup> αρχική κατανομή/1 Source at center σε χρόνο  $t=0.5$ ,  $D=10$ . Με μπλε συνεχόμενη γραμμή παρουσιάζονται τα δεδομένα του μοντέλου της διάχυσης ενώ με κίτρινη των τυχαίων βηματιστών.



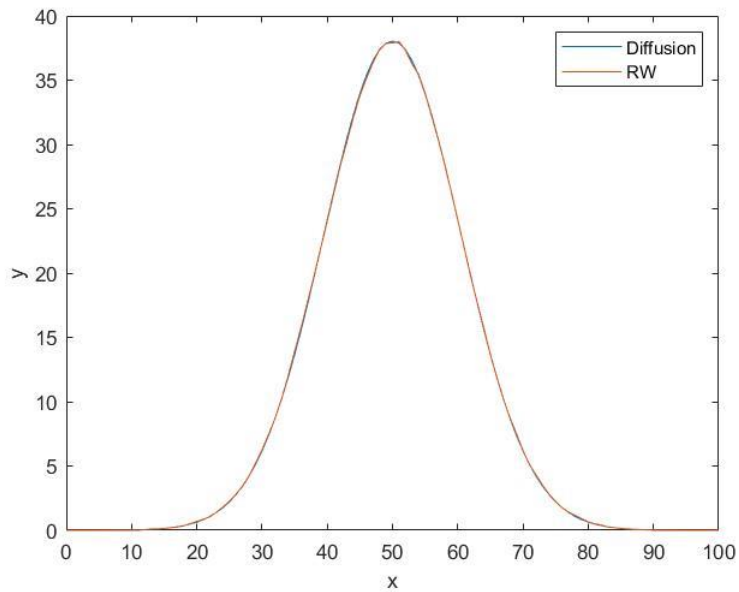
Εικόνα 3.20: Αντιπαραβολή των δεδομένων επίλυσης του μοντέλου τυχαίων βηματιστών με το μοντέλο της μοριακής διάχυσης, δεδομένα  $y$ . Επίλυσης για την 2η αρχική κατανομή/Saw σε χρόνο  $t=0.5$ ,  $D=10$ . Με μπλε συνεχόμενη γραμμή παρουσιάζονται τα δεδομένα του μοντέλου της διάχυσης ενώ με κίτρινη των τυχαίων βηματιστών.



Εικόνα 3.21 Αντιπαραβολή των δεδομένων επίλυσης του μοντέλου τυχαίων βηματιστών με το μοντέλο της μοριακής διάχυσης, δεδομένα  $y$ . Επίλυσης για την 3η αρχική κατανομή/Steps σε χρόνο  $t=0.5$ ,  $D=10$ . Με μπλε συνεχόμενη γραμμή παρουσιάζονται τα δεδομένα του μοντέλου της διάχυσης

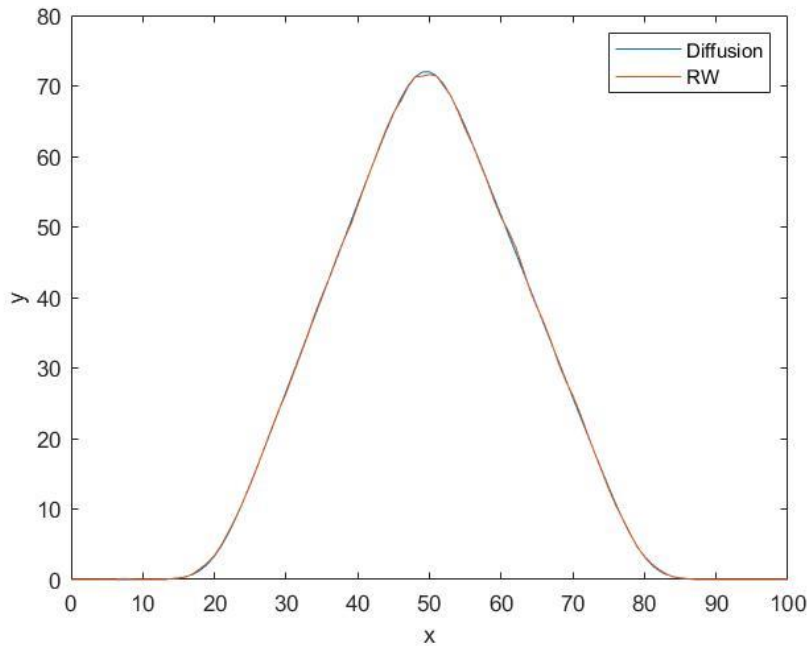


Εικόνα 3.22: Αντιπαραβολή των δεδομένων επίλυσης του μοντέλου τυχαίων βηματιστών με το μοντέλο της μοριακής διάχυσης, δεδομένα  $y$ . Επίλυσης για την 4η αρχική κατανομή/1 Larger Source at center σε χρόνο  $t=0.5$ ,  $D=10$ . Με μπλε συνεχόμενη γραμμή παρουσιάζονται τα δεδομένα του μοντέλου της διάχυσης ενώ με κίτρινη των τυχαίων βηματιστών.



Εικόνα 3.23: Αντιπαραβολή των δεδομένων επίλυσης του μοντέλου τυχαίων βηματιστών με το μοντέλο της μοριακής διάχυσης, δεδομένα  $y$ . Επίλυσης για την 5η αρχική κατανομή/Normal Distribution σε χρόνο  $t=0.5$ ,  $D=10$ . Με μπλε συνεχόμενη γραμμή παρουσιάζονται τα δεδομένα του μοντέλου της διάχυσης ενώ με κίτρινη των τυχαίων βηματιστών.





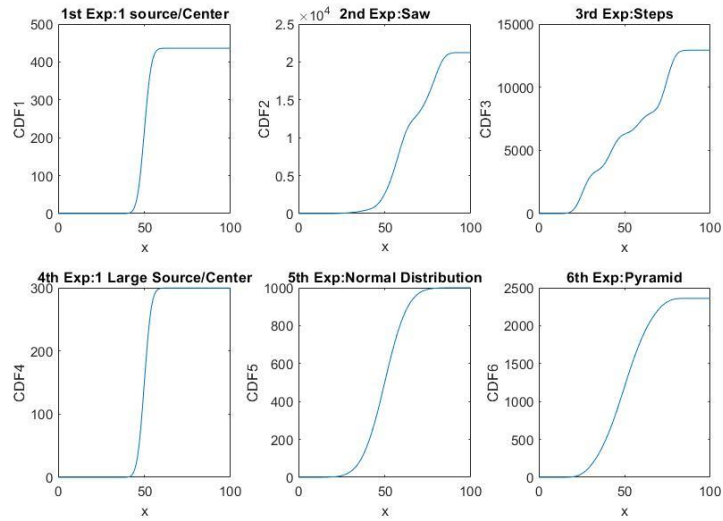
Εικόνα 3.24: Αντιπαραβολή των δεδομένων επίλυσης του μοντέλου τυχαίων βηματιστών με το μοντέλο της μοριακής διάχυσης, δεδομένα  $y$ . Επίλυσης για την  $\delta\eta$  αρχική κατανομή/Pyramid σε χρόνο  $t=0.5$ ,  $D=10$ . Με μπλε συνεχόμενη γραμμή παρουσιάζονται τα δεδομένα του μοντέλου της διάχυσης ενώ με κίτρινη των τυχαίων βηματιστών.

Είναι εμφανές από τα παραπάνω σχήματα πως το μοντέλο των βηματιστών μπορεί να προσεγγίσει ικανοποιητικά το φαινόμενο της διάχυσης. Οι μικρές διαφορές που προκύπτουν οφείλονται στην ύπαρξη θορύβου στα δεδομένα του μοντέλου των βηματιστών εξαιτίας του μεγέθους του πληθυσμού βηματιστών που χρησιμοποιείται. Όπως περιγράφηκε στην παράγραφο 1.2 όσο μεγαλύτερος είναι ο πληθυσμός που χρησιμοποιείται τόσο μεγαλύτερη ακρίβεια θα έχουν τα δεδομένα που προκύπτουν. Είναι όμως ανέφικτο να χρησιμοποιηθεί άπειρος πληθυσμός και τιμές πολύ μεγαλύτερες από αυτές που επιλέχθηκαν θα έκαναν την επίλυση του προβλήματος απαγορευτική λόγω του χρόνου επίλυσης και της απαιτούμενης μνήμης συστήματος.

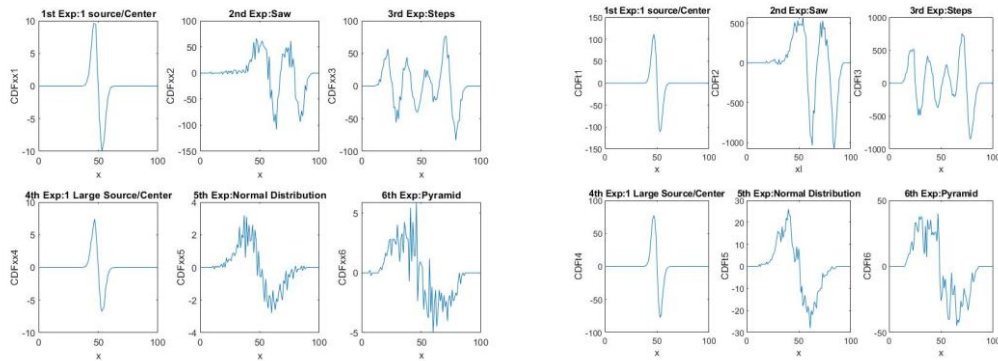
### 3.3.2 Διερεύνηση της επίδρασης του θορύβου στα δεδομένα της επίλυσης του μοντέλου των τυχαίων βηματιστών

Όπως φάνηκε παραπάνω ο θόρυβος στα δεδομένα αποτελεί ένα από τα μεγαλύτερα εμπόδια στην επίλυση του μοντέλου των τυχαίων βηματιστών. Αυτό μπορεί να αντιμετωπιστεί με δύο τρόπους. Είτε με επίλυση σε μεγαλύτερους πληθυσμούς βηματιστών είτε με επεξεργασία των δεδομένων για μείωση του θορύβου. Ακόμη, ο θόρυβος μπορεί να μειωθεί υπολογίζοντας την λύση σε κάποια χρονική τιμή πολλές φορές και λαμβάνοντας την μέση τιμή όλων των μετρήσεων. Οπότε γίνεται παραμετρική ανάλυση για πληθυσμούς  $10^4$  και  $10^5$  βηματιστές, για 10 και 100 επαναλήψεις για κάθε μέτρηση και με εφαρμογή ή χωρίς επεξεργασίας θορύβου στα δεδομένα της δεύτερης χωρικής παραγώγου στη διεύθυνση  $x$  και της χρονικής παραγώγου (όπως θα παρουσιαστεί παρακάτω αναδεικνύεται η δεύτερη χωρική παράγωγος ως προς  $x$  η μόνη σημαντική μεταβλητή στην έκφραση της Δ.Ε.). Η επίλυση του μοντέλου των βηματιστών γίνεται για χρόνο  $t=[0 \ 0.5]$  με 5 δειγματοληψίες στο χρόνο για την επιτάχυνση της διαδικασίας. Εξετάζονται 8 περιπτώσεις:

**Πληθυσμός  $10^4$  βηματιστών, 10 επαναλήψεις, χωρίς φιλτράρισμα θορύβου**  
 Χρόνος ολοκλήρωσης υπολογισμών: 17.001764 δευτερόλεπτα



Εικόνα 3.25: Οι CDF των θέσεων των βηματιστών για κάθε αρχική κατανομή σε χρόνο  $t=0.5$  για πληθυσμό  $10^4$  βηματιστών, 10 επαναλήψεις χωρίς φιλτράρισμα θορύβου

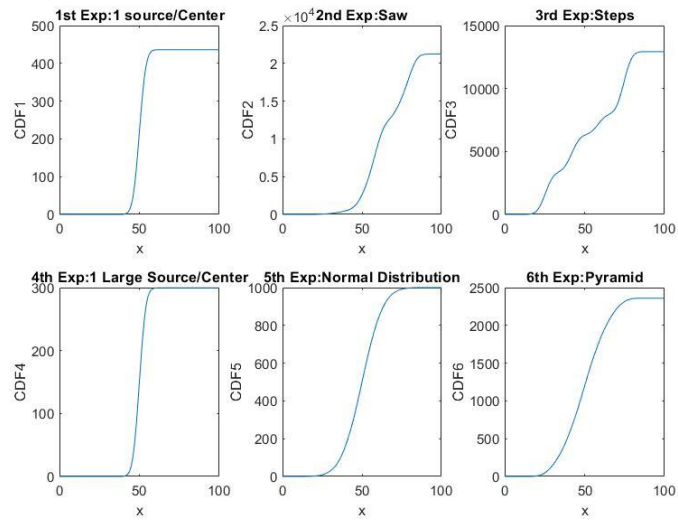


(A)

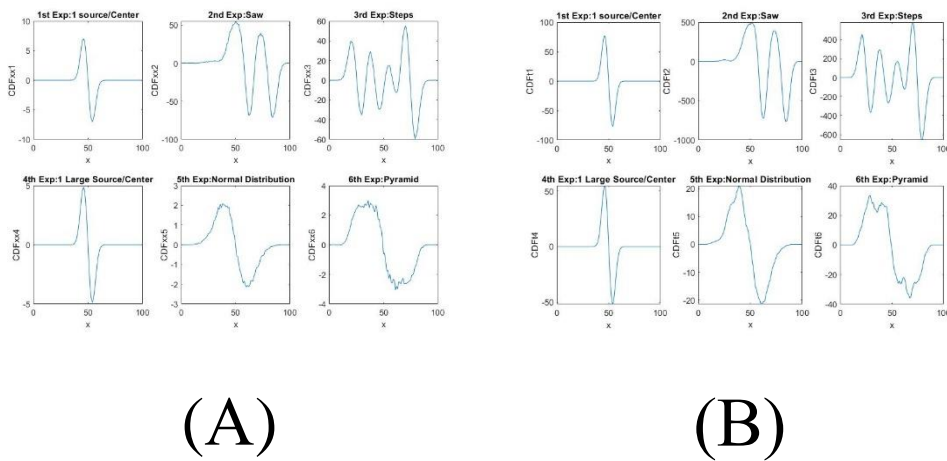
(B)

Εικόνα 3.26: Δεδομένα για την δεύτερη χωρική παράγωγο της CDF των θέσεων των βηματιστών ως προς τη διεύθυνση  $x$  (Εικόνα 3.26-A) και για τη χρονική παράγωγο της CDF (Εικόνα 3.26-B) για κάθε αρχική κατανομή σε χρόνο  $t=0.5$  για πληθυσμό  $10^4$  βηματιστών, 10 επαναλήψεις χωρίς φιλτράρισμα θορύβου

Πληθυσμός  $10^4$  βηματιστών, 10 επαναλήψεις, με φιλτράρισμα θορύβου

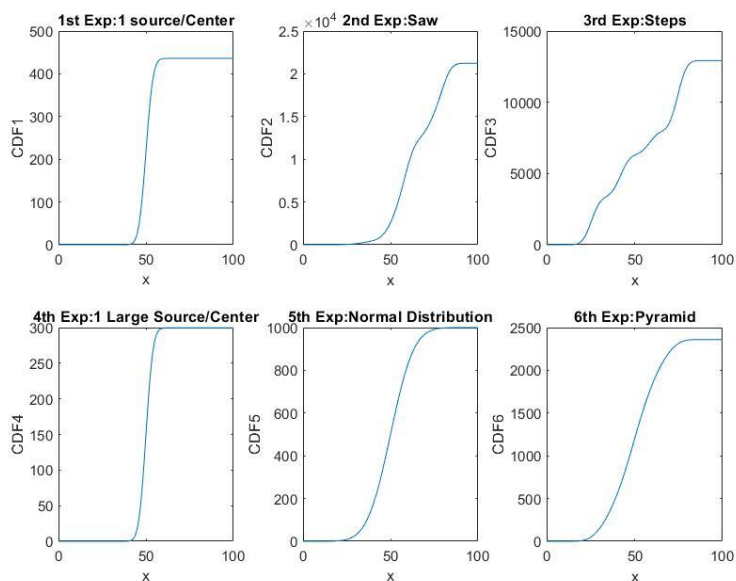


Εικόνα 3.27: Οι CDF των θέσεων των βηματιστών για κάθε αρχική κατανομή σε χρόνο  $t=0.5$  για πληθυσμό  $10^4$  βηματιστών, 10 επαναλήψεις με φιλτράρισμα θορύβου

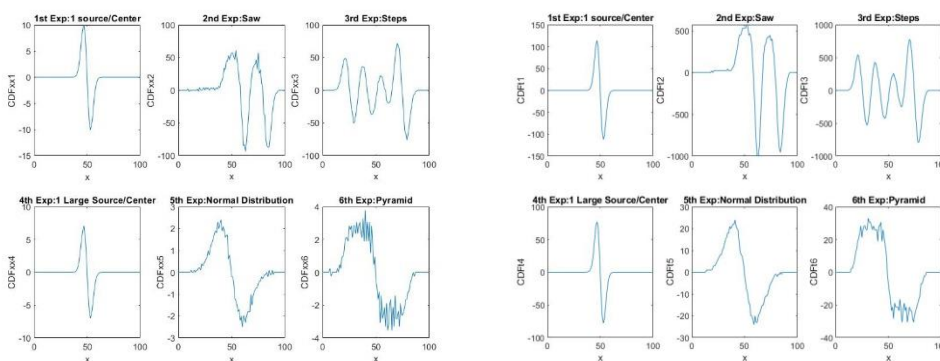


Εικόνα 3.28: Δεδομένα για την δεύτερη χωρική παράγωγο της CDF των θέσεων των βηματιστών ως προς τη διεύθυνση  $x$  (Εικόνα 3.28-A) και για τη χρονική παράγωγο της CDF (Εικόνα 3.28-B) για κάθε αρχική κατανομή σε χρόνο  $t=0.5$  για πληθυσμό  $10^4$  βηματιστών, 10 επαναλήψεις με φιλτράρισμα θορύβου

**Πληθυσμός  $10^4$  βηματιστών, 100 επαναλήψεις, χωρίς φιλτράρισμα θορύβου**  
 Χρόνος ολοκλήρωσης υπολογισμών: 186.206601 δευτερόλεπτα



Εικόνα 3.29: Οι CDF των θέσεων των βηματιστών για κάθε αρχική κατανομή σε χρόνο  $t=0.5$  για πληθυσμό  $10^4$  βηματιστών, 100 επαναλήψεις χωρίς φιλτράρισμα θορύβου

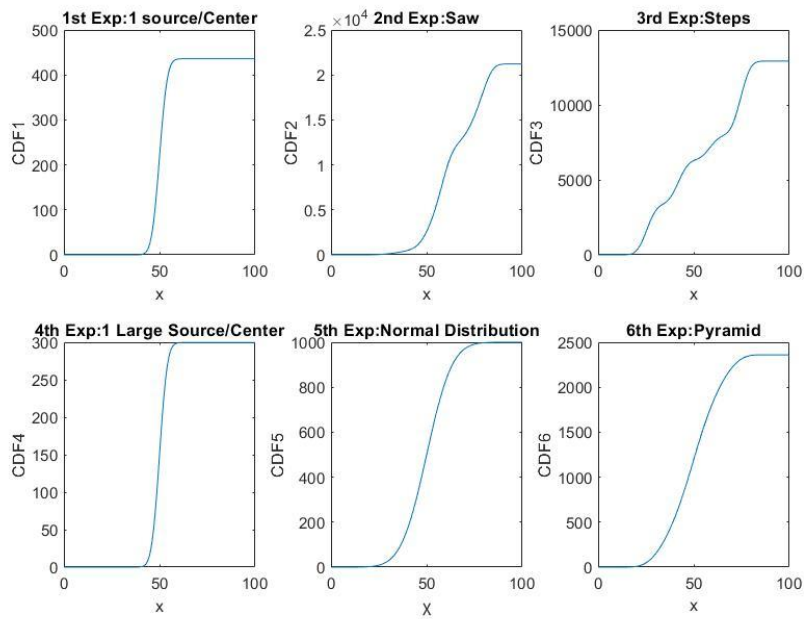


(A)

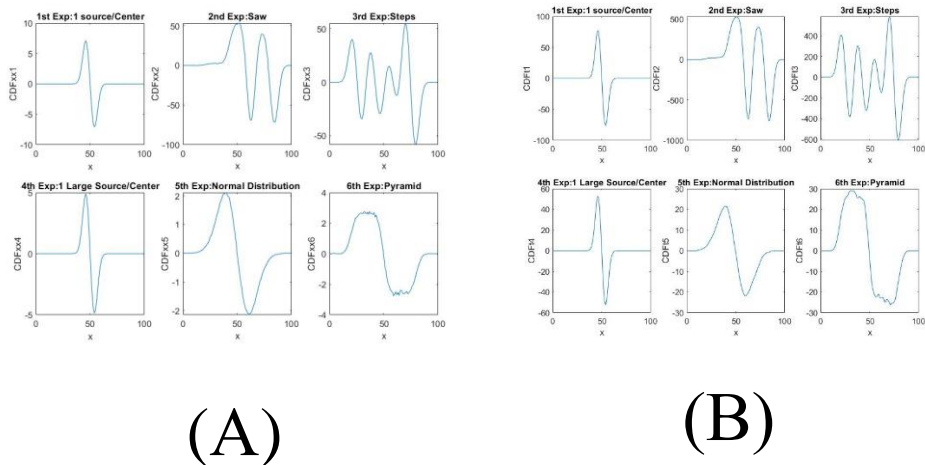
(B)

Εικόνα 3.30: Δεδομένα για την δεύτερη χωρική παράγωγο της CDF των θέσεων των βηματιστών ως προς τη διεύθυνση  $\chi$  (Εικόνα 3.30-A) και για τη χρονική παράγωγο της CDF (Εικόνα 3.30-B) για κάθε αρχική κατανομή σε χρόνο  $t=0.5$  για πληθυσμό  $10^4$  βηματιστών, 100 επαναλήψεις χωρίς φιλτράρισμα θορύβου

**Πληθυσμός  $10^4$  βηματιστών, 100 επαναλήψεις, με φιλτράρισμα θορύβου**

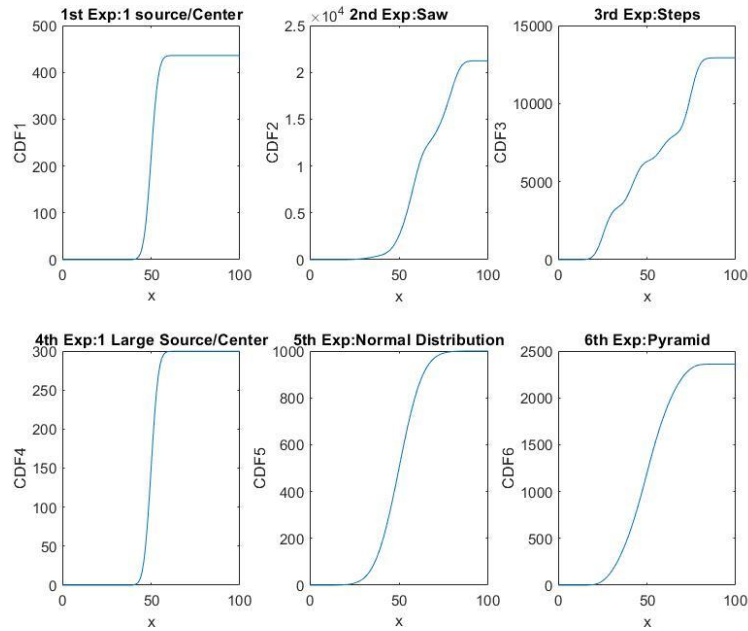


Εικόνα 3.31: Οι CDF των θέσεων των βηματιστών για κάθε αρχική κατανομή σε χρόνο  $t=0.5$  για πληθυσμό  $10^4$  βηματιστών, 100 επαναλήψεις με φιλτράρισμα θορύβου

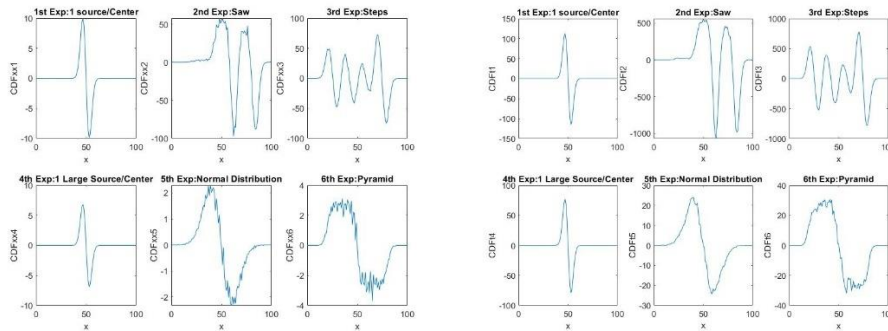


Εικόνα 3.32: Δεδομένα για την δεύτερη χωρική παράγωγο της CDF των θέσεων των βηματιστών ως προς τη διεύθυνση  $x$  (Εικόνα 3.32-A) και για τη χρονική παράγωγο της CDF (Εικόνα 3.32-B) για κάθε αρχική κατανομή σε χρόνο  $t=0.5$  για πληθυσμό  $10^4$  βηματιστών, 100 επαναλήψεις με φιλτράρισμα θορύβου

Πληθυσμός  $10^5$  βηματιστών, 10 επαναλήψεις, χωρίς φιλτράρισμα θορύβου  
 Χρόνος ολοκλήρωσης υπολογισμών: 155.814951 δευτερόλεπτα



Εικόνα 3.33: Οι CDF των θέσεων των βηματιστών για κάθε αρχική κατανομή σε χρόνο  $t=0.5$  για πληθυσμό  $10^5$  βηματιστών, 10 επαναλήψεις χωρίς φιλτράρισμα θορύβου

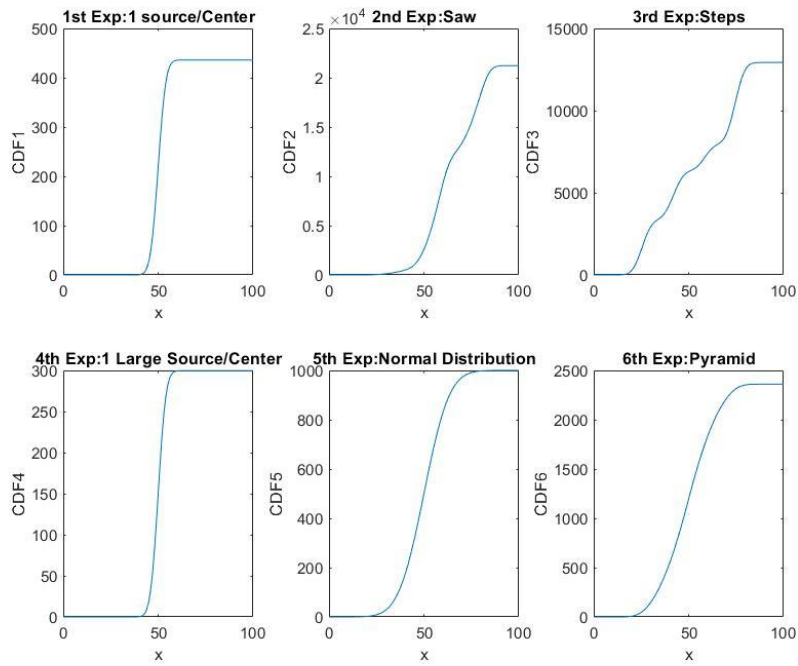


(A)

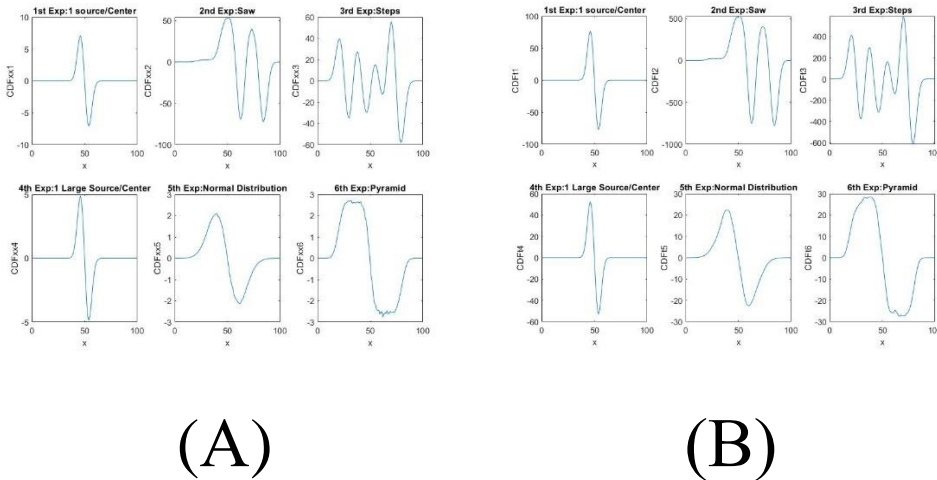
(B)

Εικόνα 3.34: Δεδομένα για την δεύτερη χωρική παράγωγο της CDF των θέσεων των βηματιστών ως προς τη διεύθυνση  $\chi$  (Εικόνα 3.34-A) και για τη χρονική παράγωγο της CDF (Εικόνα 3.34-B) για κάθε αρχική κατανομή σε χρόνο  $t=0.5$  για πληθυσμό  $10^5$  βηματιστών, 10 επαναλήψεις χωρίς φιλτράρισμα θορύβου

Πληθυσμός  $10^5$  βηματιστών, 10 επαναλήψεις, με φιλτράρισμα θορύβου

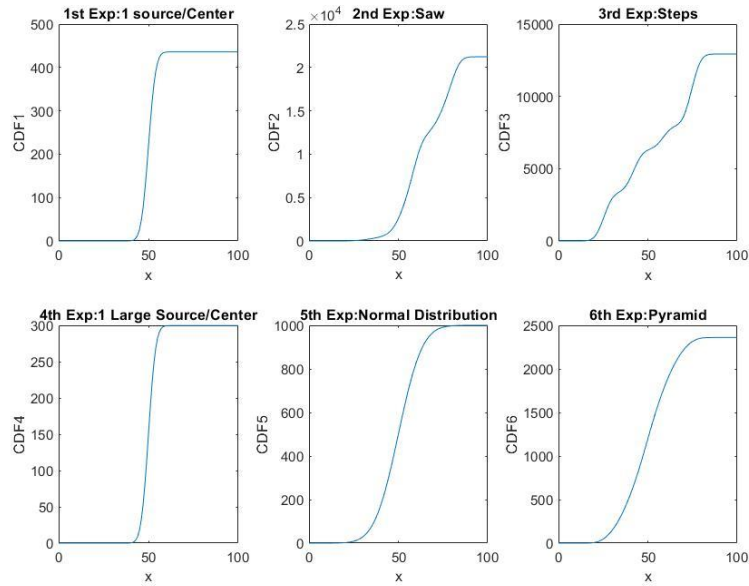


Εικόνα 3.35: Οι CDF των θέσεων των βηματιστών για κάθε αρχική κατανομή σε χρόνο  $t=0.5$  για πληθυσμό  $10^5$  βηματιστών, 10 επαναλήψεις με φιλτράρισμα θορύβου

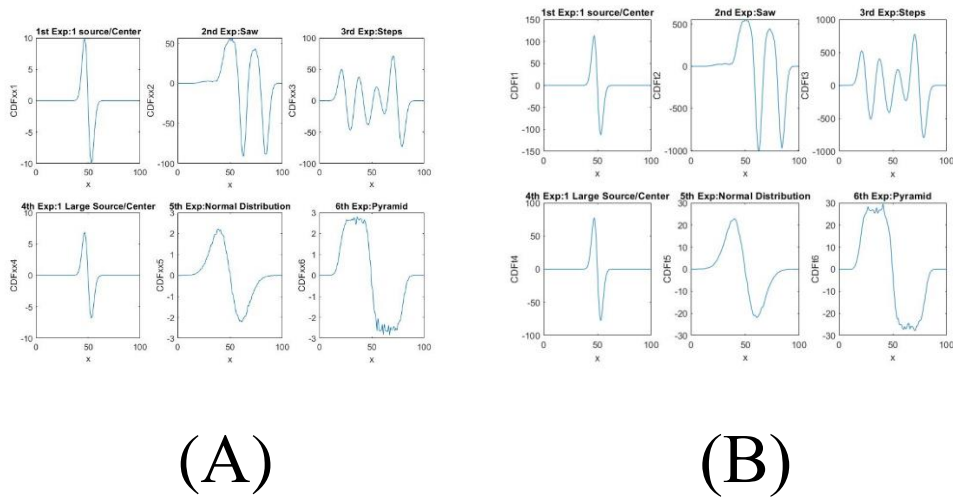


Εικόνα 3.36: Δεδομένα για την δεύτερη χωρική παράγωγο της CDF των θέσεων των βηματιστών ως προς τη διεύθυνση  $x$  (Εικόνα 3.36-A) και για τη χρονική παράγωγο της CDF (Εικόνα 3.36-B) για κάθε αρχική κατανομή σε χρόνο  $t=0.5$  για πληθυσμό  $10^5$  βηματιστών, 10 επαναλήψεις με φιλτράρισμα θορύβου

**Πληθυσμός  $10^5$  βηματιστών, 100 επαναλήψεις, χωρίς φιλτράρισμα θορύβου**  
 Χρόνος ολοκλήρωσης υπολογισμών: 1602.54 δευτερόλεπτα



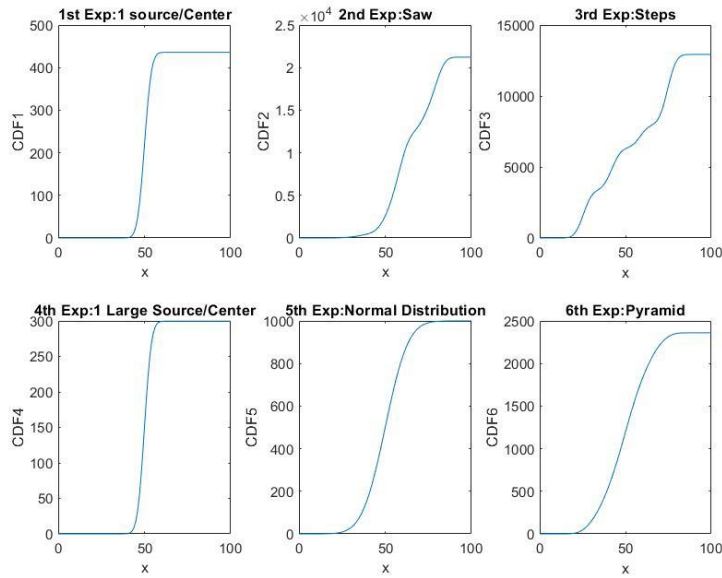
Εικόνα 3.37: Οι CDF των θέσεων των βηματιστών για κάθε αρχική κατανομή σε χρόνο  $t=0.5$  για πληθυσμό  $10^5$  βηματιστών, 100 επαναλήψεις χωρίς φιλτράρισμα θορύβου



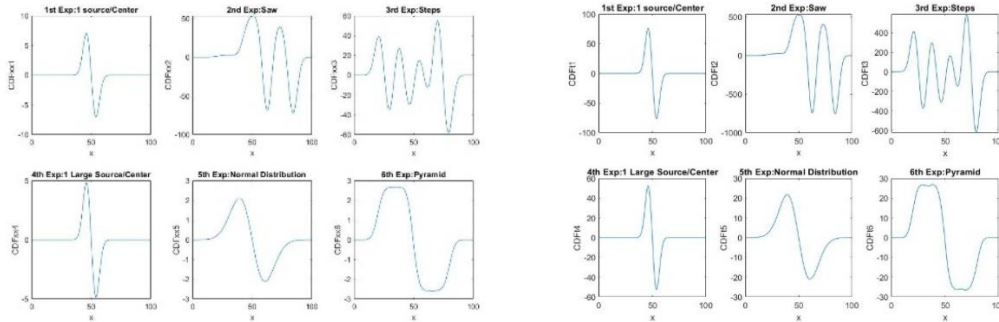
Εικόνα 3.38: Δεδομένα για την δεύτερη χωρική παράγωγο της CDF των θέσεων των βηματιστών ως προς τη διεύθυνση  $x$  (Εικόνα 3.38-A) και για τη χρονική παράγωγο της CDF (Εικόνα 3.38-B) για κάθε αρχική κατανομή σε χρόνο  $t=0.5$  για πληθυσμό  $10^5$  βηματιστών, 100 επαναλήψεις χωρίς φιλτράρισμα θορύβου



### Πληθυσμός $10^5$ βηματιστών, 100 επαναλήψεις, με φιλτράρισμα θορύβου



Εικόνα 3.39: Οι CDF των θέσεων των βηματιστών για κάθε αρχική κατανομή σε χρόνο  $t=0.5$  για πληθυσμό  $10^5$  βηματιστών, 100 επαναλήψεις με φιλτράρισμα θορύβου



(A)

(B)

Εικόνα 3.40: Δεδομένα για την δεύτερη χωρική παράγωγο της CDF των θέσεων των βηματιστών ως προς τη διεύθυνση  $x$  (Εικόνα 3.40-A) και για τη χρονική παράγωγο της CDF (Εικόνα 3.40-B) για κάθε αρχική κατανομή σε χρόνο  $t=0.5$  για πληθυσμό  $10^5$  βηματιστών, 100 επαναλήψεις με φιλτράρισμα θορύβου

Πίνακας 3.5: Χρόνος ολοκλήρωσης υπολογισμών επίλυσης μοντέλου τυχαίων βηματιστών για κάθε συνδυασμό πληθυσμού βηματιστών και επαναλήψεων

	πληθυσμός βηματιστών	
επαναλήψεις	$10^4$	$10^5$
10	17,00	155,81
100	186,21	1.602,54

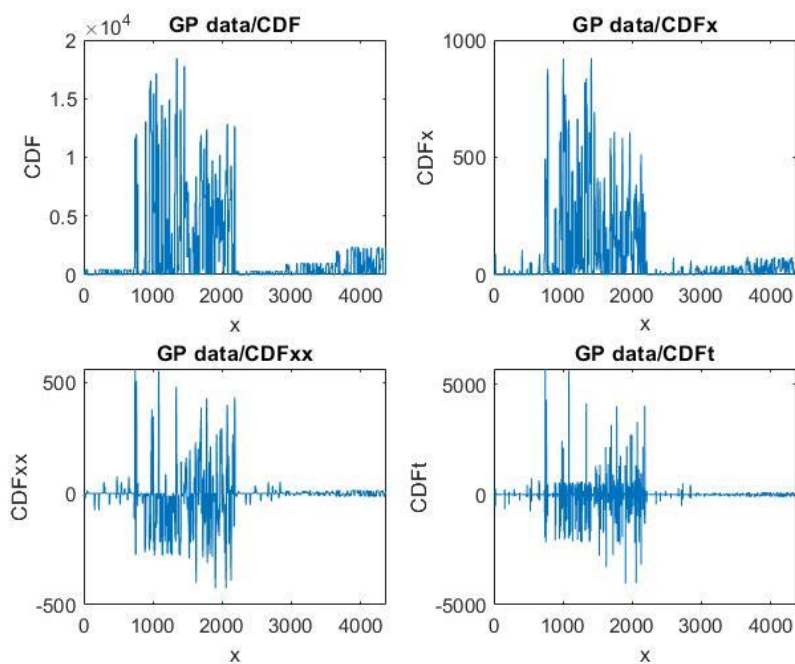
Δεν παρουσιάζονται οι χρόνοι ολοκλήρωσης υπολογισμών στις περιπτώσεις που γίνεται και επεξεργασία στο θόρυβο με φιλτράρισμα καθώς η διαδικασία είναι ταχύτερη σε όλες τις περιπτώσεις και δεν επηρεάζει τον τελικό χρόνο ολοκλήρωσης.

Όπως φαίνεται από τα παραπάνω σχήματα ο θόρυβος μειώνεται με αύξηση του πληθυσμού, με αύξηση των επαναλήψεων και με φιλτράρισμα. Ο συνδυασμός των τριών, το 8<sup>ο</sup> σενάριου που μελετάται (εικόνες 3.39-40 παρουσιάζει τα χαμηλότερα επίπεδα θορύβου. Αυτό φαίνεται σε όλες τις κατανομές και ειδικότερα στα αποτελέσματα της 6<sup>ης</sup> αρχικής κατανομής.

Παρόλο που ο 8<sup>ος</sup> συνδυασμός (Πληθυσμός  $10^5$  βηματιστών, 100 επαναλήψεις, με φιλτράρισμα θορύβου) παρουσιάζει το χαμηλότερο επίπεδο θορύβου οι υπολογισμοί για τα δεδομένα που θα χρησιμοποιηθούν για τις γκαουσιανές διεργασίες και στην εκπαίδευση των νευρωνικών δικτύων γίνονται με τον 4<sup>ο</sup> συνδυασμό (Πληθυσμός  $10^4$  βηματιστών, 100 επαναλήψεις, με φιλτράρισμα θορύβου) καθώς οι υπολογισμοί ολοκληρώνονται ταχύτερα (περίπου 3 λεπτά αντί για περίπου 27 λεπτά).

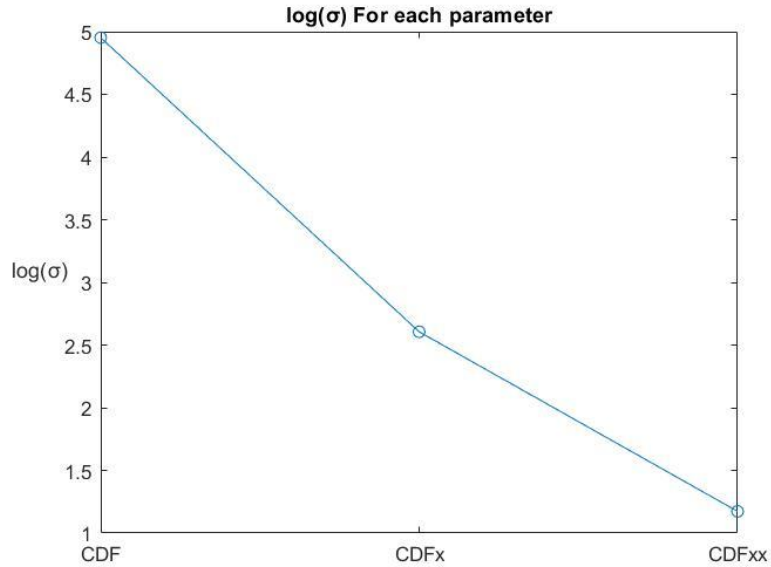
### 3.3.3 Εφαρμογή γκαουσιανών διεργασιών

Εφόσον έχει διαπιστευθεί πως τα αποτελέσματα του μοντέλου των τυχαίων βηματιστών προσομοιώνουν ικανοποιητικά το φαινόμενο της διάχυσης και έχει επιλεγεί το σετ παραμέτρων που παρέχουν το βέλτιστο συνδυασμό ελαχιστοποίησης θορύβου και χρόνου ολοκλήρωσης υπολογισμών, εφαρμόζονται γκαουσιανές διεργασίες. Παρακάτω παρουσιάζονται τα δεδομένα που χρησιμοποιήθηκαν για τις γκαουσιανές διεργασίες (Εικόνα 3.41):



Εικόνα 3.41: Τα δεδομένα που χρησιμοποιήθηκαν από την λύση του μοντέλου τυχαίων βηματιστών και τις παραγώγους του για την εφαρμογή Γκαουσιανών Διεργασιών

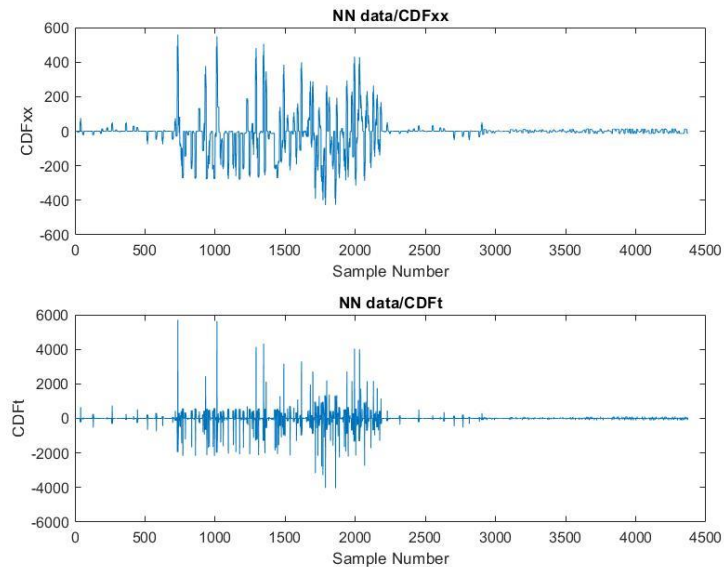
Στη παρακάτω εικόνα (Εικόνα 3.42) φαίνονται οι λογάριθμοι των τιμών των παραμέτρων Kernel που υπολογίζονται από την εφαρμογή Γκ.Δ στα παραπάνω δεδομένα:



Εικόνα 3.42: Λογάριθμοι των τιμών των παραμέτρων Kernel για κάθε μεταβλητή για το μοντέλο τυχαίων βηματιστών

### 3.3.4 Αποτελέσματα και προσαρμογή νευρωνικού δικτύου

Για την εκπαίδευση του νευρωνικού δικτύου χρησιμοποιήθηκαν τα ίδια δεδομένα με αυτά που χρησιμοποιήθηκαν για την εφαρμογή Γκ.Δ., για τις σημαντικές μεταβλητές (Εικόνα 3.43):



Εικόνα 3.43: Δεδομένα εκπαίδευσης νευρωνικού δικτύου για την επίλυση του μοντέλου των τυχαίων βηματιστών

Για την αξιολόγηση των αρχιτεκτονικών σε νέες αρχικές κατανομές χρησιμοποιούνται οι ίδιοι δείκτες που χρησιμοποιήθηκαν στην ενότητα 3.1.3.

Οι πίνακες αξιολόγησης των νευρωνικών δικτύων παρουσιάζονται παρακάτω (Πίνακας 3.6.7):

Πίνακας 3.6: Δεδομένα αξιολόγησης νευρωνικών δικτύων στο μοντέλο τυχαίων βηματιστών/ Δείκτης AUF

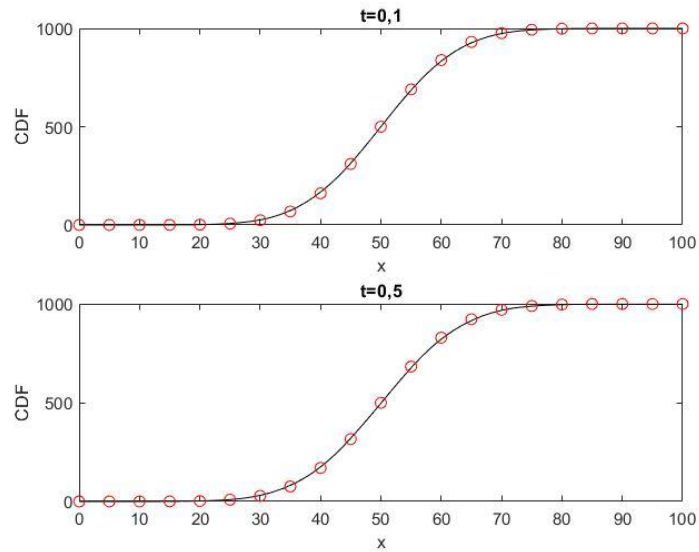
networks	layers	neurons-1	neurons-2	neurons-3	MSE-Mean	MSE-Best	STDV	AUF			
								U_in_5/t0,1	U_in_5/t0,5	U_in_8/t0,1	U_in_8/t0,5
net-1	1	3			<b>3,51E-05</b>	<b>1,19E-08</b>	<b>4,58E-05</b>	<b>1,30E-04</b>	1,10E-03	<b>8,00E-05</b>	<b>2,29E-04</b>
net-2	1	5			4,51E-06	8,25E-07	5,82E-06	8,26E-04	9,27E-04	1,07E-04	3,44E-04
net-3	1	7			3,31E-01	1,26E-07	1,05E+00	1,00E-03	9,00E-04	9,68E-05	4,86E-04
net-4	2	1	1		1,44E-04	6,58E-05	7,78E-05	1,00E-03	9,00E-04	1,47E-04	7,55E-04
net-5	2	1	3		1,20E-03	5,38E-08	2,27E-03	9,85E-04	9,42E-01	<b>8,05E-05</b>	4,81E-01
net-6	2	1	5		<b>9,51E-07</b>	<b>5,29E-12</b>	<b>2,01E-06</b>	9,93E-04	9,46E-04	8,77E-05	4,48E-04
net-7	2	1	7		1,84E-04	1,06E-11	5,78E-04	8,49E-04	9,22E-04	8,67E-05	4,40E-04
net-8	2	2	1		1,09E+04	1,13E-06	1,75E+04	1,00E-03	9,00E-04	8,27E-05	4,98E-04
net-9	2	3	1		6,81E+03	1,14E-06	1,46E+04	8,98E-04	9,36E-04	8,79E-05	4,33E-04

Πίνακας 3.7: Δεδομένα αξιολόγησης νευρωνικών δικτύων στο μοντέλο τυχαίων βηματιστών/ Δείκτης MAPE

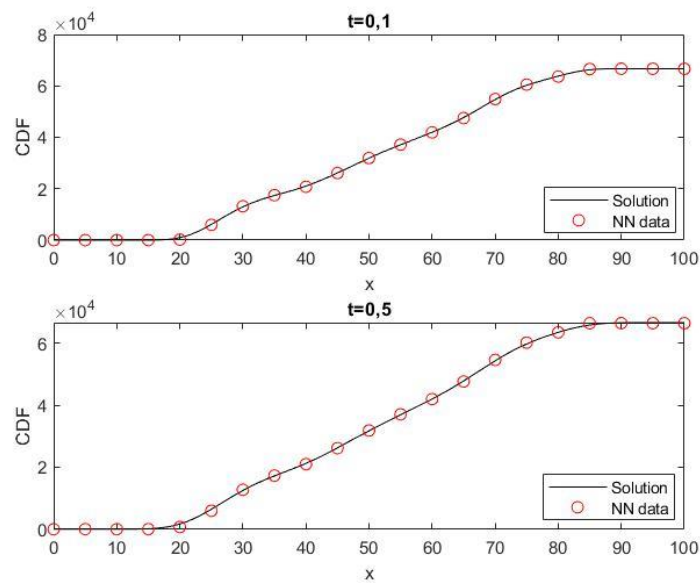
networks	layers	neurons-1	neurons-2	neurons-3	MSE-Mean	MSE-Best	STDV	MAPE			
								U_in_5/t0,1	U_in_5/t0,5	U_in_8/t0,1	U_in_8/t0,5
net-1	1	3			<b>3,51E-05</b>	<b>1,19E-08</b>	<b>4,58E-05</b>	<b>1,00E-04</b>	2,60E-03	<b>9,00E-04</b>	<b>1,50E-03</b>
net-2	1	5			4,51E-06	8,25E-07	5,82E-06	5,00E-04	<b>2,16E-03</b>	1,00E-03	1,40E-03
net-3	1	7			3,31E-01	1,26E-07	1,05E+00	3,00E-04	3,37E-03	9,00E-04	1,40E-03
net-4	2	1	1		1,44E-04	6,58E-05	7,78E-05	2,00E-04	3,32E-03	9,00E-04	1,80E-03
net-5	2	1	3		1,20E-03	5,38E-08	2,27E-03	3,00E-04	3,13E-03	1,00E-03	1,40E-03
net-6	2	1	5		<b>9,51E-07</b>	<b>5,29E-12</b>	<b>2,01E-06</b>	3,00E-04	3,38E-03	1,00E-03	<b>1,40E-03</b>
net-7	2	1	7		1,84E-04	1,06E-11	5,78E-04	5,00E-04	2,59E-03	1,00E-03	1,40E-03
net-8	2	2	1		1,09E+04	1,13E-06	1,75E+04	3,00E-04	2,60E-03	1,00E-03	1,40E-03
net-9	2	3	1		6,81E+03	1,14E-06	1,46E+04	4,00E-04	3,24E-03	1,00E-03	1,40E-03

Όπως και στην ενότητα 3.1.3, οι στήλες MSE-Mean, MSE-Best και STDV είναι κοινές και αφορούν την προσαρμογή των νευρωνικών δικτύων στα δεδομένα εκπαίδευσης. Παρουσιάζουν τα δεδομένα για το μέσο και βέλτιστο MSE και την τυπική απόκλιση των τιμών του MSE που προκύπτουν κατά τις 10 διαδοχικές επαναλήψεις εκπαίδευσης κάθε αρχιτεκτονικής. Με bold έχουν σημειωθεί οι βέλτιστες τιμές κάθε στήλης. Αναδεικνύεται η πρώτη αρχιτεκτονική ως βέλτιστη. Παρότι η 6<sup>η</sup> αρχιτεκτονική παρουσιάζει χαμηλότερο σφάλμα προσαρμογής στα δεδομένα εκπαίδευσης δεν επιλέγεται αυτή ως βέλτιστη εξαιτίας του κινδύνου υπερπροσαρμογής. Τα δεδομένα εκπαίδευσης εξακολουθούν να έχουν θόρυβο παρά την επεξεργασία που έχει γίνει οπότε ένα τόσο χαμηλό σφάλμα υποδηλώνει πως το νευρωνικό δίκτυο καταλήγει να αποκλίνει από το φαινόμενο.

Παρακάτω παρουσιάζεται η προσαρμογή του 1<sup>ου</sup> δικτύου στις κατανομές αξιολόγησης και διαγραμματικά (Εικόνα 3.44-45):



Εικόνα 3-43: Προσαρμογή βέλτιστου νευρωνικού δικτύου στην 5<sup>η</sup> αρχική κατανομή σε χρόνους  $t=0.1, 0.5$ . Με μπλε διακεκομμένες γραμμές απεικονίζονται οι αρχικές κατανομές. Με μαύρες συνεχείς γραμμές απεικονίζονται οι λύσεις που υπολογίζονται σε κάθε χρόνο. Με κόκκινους κύκλους απεικονίζονται οι τιμές που υπολογίζονται από το νευρωνικό δίκτυο σε κάθε κόμβο.



Εικόνα 3-44: Προσαρμογή βέλτιστου νευρωνικού δικτύου στην 7<sup>η</sup> αρχική κατανομή σε χρόνους  $t=0.1, 0.5$ . Με μπλε διακεκομμένες γραμμές απεικονίζονται οι αρχικές κατανομές. Με μαύρες συνεχείς γραμμές απεικονίζονται οι λύσεις που υπολογίζονται σε κάθε χρόνο. Με κόκκινους κύκλους απεικονίζονται οι τιμές που υπολογίζονται από το νευρωνικό δίκτυο σε κάθε κόμβο.

## 4 Κεφάλαιο 4<sup>ο</sup>: Συμπεράσματα και Προτάσεις

### 4.1 Διεργασίες Γκάους

Σε κάθε πρόβλημα που εφαρμόστηκε η μέθοδος των Διεργασιών Gauss κατάφερε να αναγνωρίσει με επιτυχία τις σημαντικές μεταβλητές του προβλήματος. Παρότι η εφαρμογή τους απαιτεί μεγάλο όγκο δεδομένων χαμηλού θόρυβου, για τις συναρτήσεις των οποίων μελετάται η προσαρμογή στα δεδομένα στόχου, φάνηκε η δύναμη της μεθόδου.

### 4.2 Εκπαίδευση Νευρωνικών Δικτύων

Όπως και οι γκαουσιανές διεργασίες έτσι και τα νευρωνικά δίκτυα εμφανίζουν μεγάλη ευαισθησία στα δεδομένα εκπαίδευσης και απαιτούν μεγάλο σετ δεδομένων για ικανοποιητική προσαρμογή. Παρόλα αυτά, με επαρκές σετ δεδομένων εκπαίδευσης από διαφορετικές αρχικές κατανομές, ήταν δυνατόν να εκπαιδευτούν ρηχά νευρωνικά δίκτυα χαμηλής περιπλοκότητας τα οποία εμφάνιζαν εξαιρετική προσαρμογή στα δεδομένα στόχου ακόμα και για νέες κατανομές και μεγάλους χρόνους. Ακόμη, πρέπει να τονιστεί η ταχύτητα επίλυσης των διαφορικών εξισώσεων. Εκεί που μια επαναληπτική διαδικασία αριθμητικής επίλυσης απαιτούσε μερικά δευτερόλεπτα, το νευρωνικό δίκτυο παρήγαγε μία λύση ικανοποιητικής ακρίβειας αστραπιαία. Παρόλο που για προβλήματα χαμηλού αριθμού επαναλήψεων και πολυπλοκότητας η διαφορά είναι μικρή, όσο πιο περίπλοκη είναι η σχέση που επιλύεται και άρα πιο χρονοβόρες οι πράξεις που απαιτούνται να γίνουν εντός μιας επαναληπτικής διαδικασίας (πολλαπλασιασμός και εύρεση αντιστρόφου πολύ μεγάλων πινάκων, προβλήματα ελαχιστοποίησης, δομής επιλογής if) τόσο μεγαλύτερη θα είναι και η διαφορά.

### 4.3 Μεθοδολογία

Στην παρούσα εργασία αναπτύχθηκε με επιτυχία μεθοδολογία εύρεσης λύσης χωρίς γνώση της έκφρασης της διαφορικής εξίσωσης που διέπει το φαινόμενο με δεδομένα από αριθμητική επίλυση διαφορικών εξισώσεων όπως και από προσομοιωτή μικροκλίμακας.

### 4.4 Προτάσεις για περαιτέρω έρευνα

#### 4.4.1 Επιτάχυνση μεθοδολογίας

Μεγάλο ενδιαφέρον θα είχε η διερεύνηση μεθόδων επιτάχυνσης της μεθοδολογίας που αναπτύσσεται στην ενότητα 2 για επίλυση περιπλοκότερων προβλημάτων. Θα μπορούσαν να επιστρατευτούν μέθοδοι επιτάχυνσης της επίλυσης των διαφορικών εξισώσεων και του μοντέλου μικροκλίμακας με σκοπό την επιτάχυνση της παραγωγής των δεδομένων εκπαίδευσης.

#### 4.4.2 Επίλυση των προβλημάτων σε ανώτερες διαστάσεις

Τα προβλήματα που μελετώνται στην παρούσα διπλωματική επιλύονται σε μόνο μία χωρική διάσταση. Μια νέα μελέτη θα μπορούσε να εμβαθύνει στη εφαρμογή των μεθόδων που περιγράφονται σε 2 και 3 χωρικές διαστάσεις. Αναμένεται η επίλυση των μοντέλων για την παραγωγή δεδομένων εκπαίδευσης να μην είναι πολύ πιο περίπλοκη αλλά να έχει μεγαλύτερες απαιτήσεις σε χρόνο τρεξίματος. Θα έχει ενδιαφέρον η εφαρμογή Γκαουσιανών μεθόδων σε δεδομένα ανώτερων χωρικών διαστάσεων για τη μελέτη της συμπεριφοράς των μεθόδων απέναντι στις μερικές παραγώγους  $\frac{\partial^2}{\partial x_i \partial x_j}$ , κλπ. Αντίστοιχα για την εφαρμογή νευρωνικών δικτύων θα πρέπει να αξιολογηθεί η δυνατότητα προσαρμογής τους σε δεδομένα πολλαπλών διαστάσεων και το σφάλμα που θα προκύπτει.

#### 4.4.3 Μελέτη πιο περίπλοκων διαφορικών εξισώσεων

Στη παρούσα εργασία εξετάζονται δύο σχετικά απλές διαφορικές εξισώσεις. Αξίζει να μελετηθεί εάν η μεθοδολογία που αναπτύσσεται θα μπορούσε να εφαρμοστεί σε πιο περίπλοκα,

μη γραμμικά προβλήματα τα οποία περιλαμβάνουν περισσότερες σημαντικές μεταβλητές. Για παράδειγμα θα μπορούσε να μελετηθεί το πρόβλημα κινούμενου μετώπου Nagumo[35]:

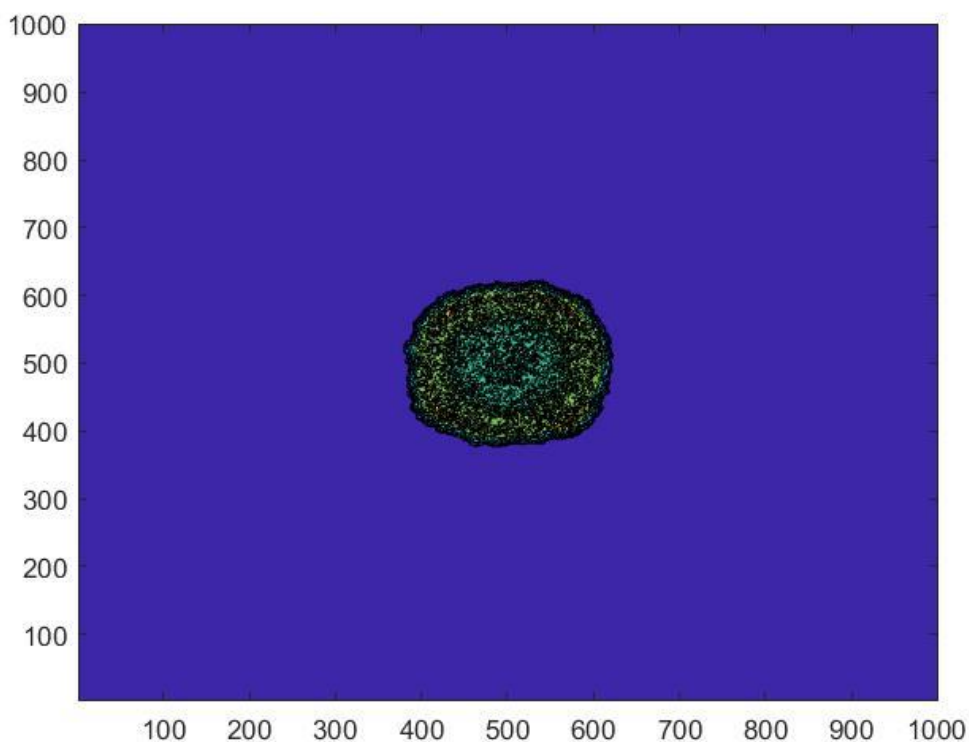
$$\frac{\partial u}{\partial t} = D \frac{d^2 u}{dx^2} + u(1-u)(u-a) \quad (4.1)$$

όπου D είναι ο συντελεστής διάχυσης και a μια κινητική παράμετρος του προβλήματος, ή κάποιο πρόβλημα αντίδρασης-διάχυσης το οποίο να περιλαμβάνει όρο συναγωγής,  $u\nu$ .

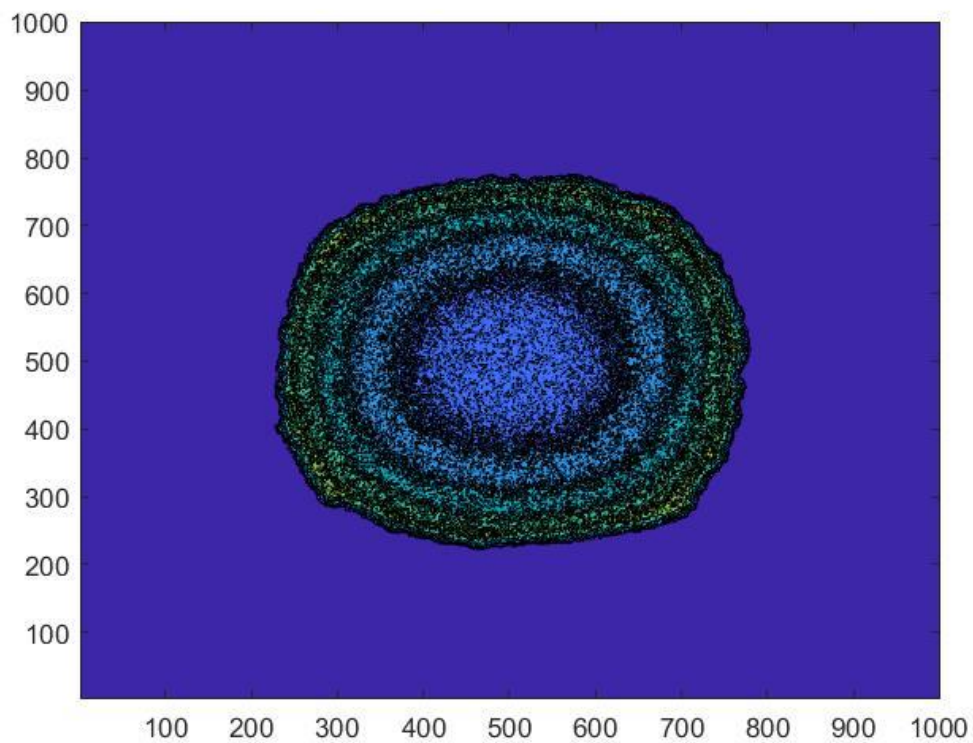
#### 4.4.4 Μελέτη πιο περίπλοκων μοντέλων προσομοιωτών μικροκλίμακας

Το μοντέλο των τυχαίων βηματιστών αποτελεί ένα σχετικά απλό πείραμα MC σε μία διάσταση. Θα άξιζε να μελετηθεί η εφαρμογή της μεθοδολογίας της εργασίας σε πιο περίπλοκους προσομοιωτές μικροκλίμακας οι οποίοι διέπονται από πολύπλοκα μοντέλα επαναληπτικής διαδικασίας που καθορίζουν το πληθυσμό που μελετάται και δε μπορούν να αντιστοιχιστούν με κάποια διαφορική εξίσωση που να περιγράφει ικανοποιητικά το μοντέλο. Ένα τέτοιο παράδειγμα είναι το κυτταρικό αυτόματο.[37, 38]

Ξεκινώντας με ένα κύτταρο στο κέντρο του διδιάστατου χωρίου υπολογίζονται οι κατανομές των θρεπτικών συστατικών και οι ενέργειες που θα λάβει το κύτταρο (μετακίνηση, μίτωση, θάνατος) εξαρτώνται από τη συγκέντρωση των θρεπτικών συστατικών και ένα τυχαίο παράγοντα. Παρακάτω παρουσιάζονται οι κατανομές που παράγονται μετά από 1 ώρα και 6 ώρες, στις 300 και 668 επαναλήψεις (Εικόνες 4.1 και 4.2):



Εικόνα 4.1: Πληθυσμός κυττάρων μετά από 1 ώρα και 300 επαναλήψεις



Εικόνα 4.2: Πληθυσμός κυττάρων μετά από 6 ώρες και 668 επαναλήψεις

Στις δύο παραπάνω εικόνες φαίνεται ο ρυθμός εξέλιξης του μετώπου των κυττάρων. Όσο αυξάνονται οι επαναλήψεις και άρα και πληθυσμός των κυττάρων για τα οποία λαμβάνονται αποφάσεις για το τι ενέργεια θα λάβουν, τόσο αυξάνεται ο απαιτούμενος χρόνος και χώρος μνήμης. Εάν μπορέσει να εκπαιδευτεί νευρωνικό δίκτυο πάνω σε αυτά τα δεδομένα το οποίο θα μπορεί να προβλέπει ικανοποιητικά τις κατανομές θα επιταχυνθεί σημαντικά η επίλυση του προβλήματος καθώς από ώρες θα απαιτεί δευτερόλεπτα.



## 5 Βιβλιογραφία

- [1] W. E. Boyce, R. C. DiPrima, and D. B. Meade, *Elementary differential equations and boundary value problems*, Twelfth edition. ed. Hoboken: Wiley, 2022, p. pages cm.
- [2] H. S. Fogler, *Elements of chemical reaction engineering*, Fifth edition. ed. Boston: Prentice Hall, 2016, pp. xxxiii, 957 pages.
- [3] F. P. Incropera and F. P. Incropera, *Fundamentals of heat and mass transfer*, 6th ed. Hoboken, NJ: John Wiley, 2007, pp. xxv, 997 p.
- [4] H. D. Young, R. A. Freedman, and H. D. Young, *University physics with modern physics*, 15th edition. ed. Hoboken, N.J.: Pearson Higher Education, 2020, p. volumes cm.
- [5] Ι. Ε. Σ.-Θ. Καρακασίδης, *Αριθμητικές Μέθοδοι και Εφαρμογές για Μηχανικούς με Παραδείγματα στο Matlab*. Εκδόσεις Τζιόλα, 2019.
- [6] D. G. Zill and M. R. Cullen, *Differential equations with boundary-value problems*, 7th ed. Belmont, CA: Brooks/Cole, Cengage Learning, 2009, pp. xv, 526, 23, 30, 9 p.
- [7] A. Mohsen, "A simple solution of the Bratu problem," *Computers & Mathematics with Applications*, vol. 67, no. 1, pp. 26-33, 2014/01/01/ 2014, doi: <https://doi.org/10.1016/j.camwa.2013.10.003>.
- [8] A. Ahmad, M. Sulaiman, A. J. Aljohani, A. Alhindi, and H. Alrabaiah, "Design of an efficient algorithm for solution of Bratu differential equations," *Ain Shams Engineering Journal*, vol. 12, no. 2, pp. 2211-2225, 2021/06/01/ 2021, doi: <https://doi.org/10.1016/j.asej.2020.11.007>.
- [9] R. Buckmire, "On exact and numerical solutions of the one-dimensional planar Bratu problem," 2003.
- [10] Η. ΕΥΣΤΑΘΙΟΣ, "Διερεύνηση μη γραμμικών φαινομένων με χρήση μεθόδων μηχανικής μάθησης," Τομέας Ανάλυσης, Σχεδιασμού και Ανάπτυξης Διεργασιών και Συστημάτων, Εθνικό Μετσόβιο Πολυτεχνείο, 2023.
- [11] F. James, "Monte Carlo theory and practice," *Reports on Progress in Physics*, 1980.
- [12] W. L. Dunn and J. K. Shultis, *Exploring Monte Carlo methods*. Amsterdam ; Boston: Elsevier/Academic Press, 2012, pp. xvi, 384 p.
- [13] R. Y. Rubinstein and D. P. Kroese, *Simulation and the Monte Carlo method / Reuven Y. Rubinstein, Technion, Dirk P. Kroese, University of Queensland*, Third edition. ed. (Wiley series in probability and statistics). Hoboken, New Jersey: Wiley, 2017, pp. xvii, 414 pages.
- [14] B. K. NATARAJA, *Machine Learning. A Theoretical Approach*. Morgan Kaufmann Publishers, Inc. , 1991.
- [15] Y. Ureel, M. R. Dobbelaere, Y. Ouyang, K. De Ras, M. K. Sabbe, G. B. Marin, and K. M. Van Geem, "Active Machine Learning for Chemical Engineers: A Bright Future Lies Ahead!," *Engineering*, 2023/08/01/ 2023, doi: <https://doi.org/10.1016/j.eng.2023.02.019>.
- [16] Θ. Σ., *Machine learning. A Bayesian and optimization perspective*. Elsevier, 2021.
- [17] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
- [18] M. H. Modarres, R. Aversa, S. Cozzini, R. Ciancio, A. Leto, and G. P. Brandino, "Neural Network for Nanoscience Scanning Electron Microscope Image Recognition," *Scientific Reports*, vol. 7, no. 1, p. 13282, 2017/10/16 2017, doi: 10.1038/s41598-017-13565-z.
- [19] B. B. Traore, B. Kamsu-Foguem, and F. Tangara, "Deep convolution neural network for image recognition," *Ecological Informatics*, vol. 48, pp. 257-268, 2018/11/01/ 2018, doi: <https://doi.org/10.1016/j.ecoinf.2018.10.002>.

- [20] B. Su and S. Lu, "Accurate Scene Text Recognition Based on Recurrent Neural Network," in *Computer Vision – ACCV 2014*, Cham, D. Cremers, I. Reid, H. Saito, and M.-H. Yang, Eds., 2015// 2015: Springer International Publishing, pp. 35-48.
- [21] M. Yousef, K. F. Hussain, and U. S. Mohammed, "Accurate, data-efficient, unconstrained text recognition with convolutional neural networks," *Pattern Recognition*, vol. 108, p. 107482, 2020/12/01/ 2020, doi: <https://doi.org/10.1016/j.patcog.2020.107482>.
- [22] G. Dede and M. H. Sazli, "Speech recognition with artificial neural networks," *Digital Signal Processing*, vol. 20, no. 3, pp. 763-768, 2010/05/01/ 2010, doi: <https://doi.org/10.1016/j.dsp.2009.10.004>.
- [23] A. Maedche *et al.*, "AI-Based Digital Assistants," *Business & Information Systems Engineering*, vol. 61, no. 4, pp. 535-544, 2019/08/01 2019, doi: 10.1007/s12599-019-00600-8.
- [24] M. B. a. D. D. T. a. D. D. a. B. F. a. B. F. a. P. G. a. L. D. J. a. M. M. a. U. M. a. J. Zhang, "End to End Learning for Self-Driving Cars," *arXiv*, 2016, doi: 10.48550/arXiv.1604.07316.
- [25] V. Shreyas, S. N. Bharadwaj, S. Srinidhi, K. U. Ankith, and A. B. Rajendra, "Self-driving Cars: An Overview of Various Autonomous Driving Systems," in *Advances in Data and Information Sciences*, Singapore, M. L. Kolhe, S. Tiwari, M. C. Trivedi, and K. K. Mishra, Eds., 2020// 2020: Springer Singapore, pp. 361-371.
- [26] R. S. Sutton and A. G. Barto, *Reinforcement learning : an introduction* (Adaptive computation and machine learning). Cambridge, Mass.: MIT Press, 2018, pp. xviii, 322 p.
- [27] C. C. Aggarwal, *Neural Networks and Deep Learning*. Springer, 2018.
- [28] S. S. Haykin and S. S. Haykin, *Neural networks and learning machines*, 3rd ed. New York: Prentice Hall, 2009, pp. xxx, 906 p.
- [29] M. T. Hagan, H. B. Demuth, and M. H. Beale, *Neural network design*, 1st ed. Boston: PWS Pub., 1996.
- [30] K. Gurney, *An introduction to neural networks*. London: UCL Press, 1997, pp. xi, 234 p.
- [31] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning* (Adaptive computation and machine learning). Cambridge, Mass.: MIT Press, 2006, pp. xviii, 248 p.
- [32] K. P. Murphy, *Machine learning : a probabilistic perspective* (Adaptive computation and machine learning series). Cambridge, MA: MIT Press, 2012, pp. xxix, 1067 p.
- [33] C. M. Bishop, *Pattern recognition and machine learning* (Information science and statistics). New York: Springer, 2006, pp. xx, 738 p.
- [34] P. P. b. Maziar Raissi, George Em Karniadakis "Machine learning of linear differential equations using Gaussian processes," *Journal of Computational Physics*, 2017.
- [35] M. E. Kavousanakis, R. Erban, A. G. Boudouvis, C. W. Gear, and I. G. Kevrekidis, "Projective and coarse projective integration for problems with continuous symmetries," *Journal of Computational Physics*, vol. 225, no. 1, pp. 382-407, 2007/07/01/ 2007, doi: <https://doi.org/10.1016/j.jcp.2006.12.003>.
- [36] <https://www.mathworks.com/products/matlab.html> (accessed).
- [37] S. Ferreira and M. Vilela, "Reaction-diffusion model for the growth of avascular tumor," *Physical review. E, Statistical, nonlinear, and soft matter physics*, vol. 65, p. 021907, 03/01 2002, doi: 10.1103/PhysRevE.65.021907.
- [38] D. G. Mallet and L. G. De Pillis, "A cellular automata model of tumor-immune system interactions," *Journal of Theoretical Biology*, vol. 239, no. 3, pp. 334-350, 2006/04/07/ 2006, doi: <https://doi.org/10.1016/j.jtbi.2005.08.002>.

## 6 Παράρτημα

### 6.1 Αριθμητική επίλυση της Δ.Ε. μοριακής διάχυσης σε μεταβατική κατάσταση, σε μια διάσταση με την μέθοδο των πεπερασμένων διαφορών

Σύμφωνα με την μέθοδο των πεπερασμένων διαφορών, προσεγγίστηκαν οι παράγωγοι στην εξίσωση 2.1, η πρώτη παράγωγος ως προς το χρόνο με σχήμα εμπρός διαφορών και η δεύτερη παράγωγος ως προς τη διεύθυνση  $\chi$  με σχήμα κεντρικών διαφορών:

$$\frac{u_i^{m+1} - u_i^m}{\Delta t} = D \frac{u_{i+1}^m + u_{i-1}^m - 2u_i^m}{\Delta \chi^2} \rightarrow u_i^{m+1} = ru_{i+1}^m + ru_{i-1}^m + (1 - 2r)u_i^m \quad (6.1)$$

$$\text{όπου } r = \frac{D\Delta t}{\Delta \chi^2}.$$

Μια εμφανής αδυναμία της μεθόδου πεπερασμένων διαφορών είναι το ότι δεν μπορεί να εφαρμοστεί στους ακριανούς κόμβους του χωρίου. Αυτό συμβαίνει διότι σε αυτούς τους κόμβους απαιτούνται οι τιμές κόμβων εκτός του χωρίου που μελετάται. Το πρόβλημα αυτό αντιμετωπίζεται ανάλογα με τις συνοριακές συνθήκες που εφαρμόζονται στα άκρα: [5]

#### Συνοριακές Συνθήκες τύπου Neumann

Σε προβλήματα όπου χρησιμοποιούνται συνοριακές συνθήκες τύπου Neumann, όπου δηλαδή είναι σταθερή η τιμή της παραγώγου της συνάρτησης και όχι της ίδιας της συνάρτησης, οι εξωτερικοί κόμβοι που προκύπτουν αντιμετωπίζονται ως εικονικοί. Με την μέθοδο των πεπερασμένων στοιχείων προσεγγίζονται αριθμητικά οι παράγωγοι στους ακριανούς κόμβους και από την σχέση που προκύπτει αντικαθίσταται η τιμή του εικονικού κόμβου με αυτή ενός εσωτερικού στην διακριτοποίηση του προβλήματος.

Εφαρμογή συνοριακών συνθηκών τύπου Neumann στα άκρα:

$$\left. \frac{\partial u}{\partial x} \right|_{\text{άκρο}} = 0 \rightarrow \frac{u_{i+1} - u_{i-1}}{\Delta \chi} = 0 \rightarrow u_{i+1} = u_{i-1} \quad (6.2)$$

Δηλαδή στα άκρα δεν υπάρχει μοριακή διάχυση.

Με εφαρμογή της παραπάνω μεθοδολογίας υπολογίζεται η διακριτοποίηση του προβλήματος στους ακριανούς κόμβους:

#### Κόμβος $i=1$

Γίνεται χρήση του εικονικού κόμβου  $i=0$ .

$$(6.2): u_2 = u_0 \quad (6.3)$$

$$(6.1): u_1^{m+1} = ru_2^m + ru_0^m + (1 - 2r)u_1^m \rightarrow u_1^{m+1} = 2ru_2^m + (1 - 2r)u_1^m \quad (6.4)$$

#### Κόμβος $i=L$ (τελευταίος κόμβος)

Γίνεται χρήση του εικονικού κόμβου  $i=L+1$ .

$$(2.3): u_{L+1} = u_{L-1} \quad (6.5)$$

$$(6.1): u_L^{m+1} = ru_{L+1}^m + ru_{L-1}^m + (1 - 2r)u_L^m \rightarrow u_L^{m+1} = 2ru_{L-1}^m + (1 - 2r)u_L^m \quad (6.6)$$

Οπότε προκύπτει η συνολική διακριτοποίηση του προβλήματος:

$$\left\{ \begin{array}{l} i = 1: u_1^{m+1} = 2ru_2^m + (1 - 2r)u_1^m \\ i = [2, L - 1]: u_i^{m+1} = ru_{i+1}^m + ru_{i-1}^m + (1 - 2r)u_i^m \\ i = L: u_L^{m+1} = 2ru_{L-1}^m + (1 - 2r)u_L^m \end{array} \right\}$$

## 6.2 Αριθμητική επίλυση του προβλήματος Bratu σε μεταβατική κατάσταση, σε μια διάσταση με την μέθοδο των πεπερασμένων διαφορών

Σύμφωνα με την μέθοδο των πεπερασμένων διαφορών, προσεγγίστηκαν οι παράγωγοι στην εξίσωση 2.6, η πρώτη παράγωγος ως προς το χρόνο με σχήμα εμπρός διαφορών και η δεύτερη παράγωγος ως προς τη διεύθυνση  $x$  με σχήμα κεντρικών διαφορών:

$$\frac{u_i^{m+1} - u_i^m}{\Delta t} = D \frac{u_{i+1}^m + u_{i-1}^m - 2u_i^m}{\Delta x^2} + \lambda e^{u_i^m} \rightarrow$$

$$u_i^{m+1} = r' u_{i+1}^m + r' u_{i-1}^m + (1 - 2r') u_i^m + \lambda \Delta t e^{u_i^m} \quad (6.7)$$

όπου  $r' = \frac{D\Delta t}{\Delta x^2}$ .

Στα άκρα του χωρίου εφαρμόζονται συνοριακές συνθήκες Dirichlet:  $u(0) = u(1) = 0$ .

Οπότε προκύπτει η διακριτοποίηση του προβλήματος:

$$\left\{ \begin{array}{l} i = 1: u_1^{m+1} = 0 \\ i = [2, L - 1]: u_i^{m+1} = r' u_{i+1}^m + r' u_{i-1}^m + (1 - 2r') u_i^m + \lambda \Delta t e^{u_i^m} \\ i = L: u_L^{m+1} = 0 \end{array} \right\}$$

## 6.3 Κώδικας επίλυσης Δ.Ε. μοριακής διάχυσης σε μεταβατική κατάσταση, σε μία διάσταση

```
%Diffusion_NN
clear;clc;

% Define Grid
dh=1 ;
L=100 ;
x=0:dh:L; x(end)=L;
n=length(x); % number of nodes in x direction

% Time domain
dt=0.1;
t_final=100 ;
t=(0.1:dt:t_final);

% Boundary Conditions
ux_left=0;
ux_right=0;

% Initial Conditions
U_in_1=zeros(n,1);U_in_4=zeros(n,1);U_in_7=zeros(n,1);
b=0:20;

U_in_1((n+1)/2)=rand*1000; %source located at center
U_in_2=[b*rand*100 b*rand*100 b*rand*100 b*rand*100 b*rand*100]; U_in_2(102:105)=[];
U_in_2=U_in_2'; %Saw
U_in_3=[0 zeros(1,10) ones(1,10)*rand*1000 zeros(1,10) ones(1,10)*rand*1000
zeros(1,10) ones(1,10)*rand*1000 zeros(1,10) ones(1,10)*rand*1000 zeros(1,10)
ones(1,10)*rand*1000]; U_in_3=U_in_3'; %Steps
U_in_4((n+1)/2)=rand*1000; %one larger source
U_in_5=normpdf(x,50,10)*1000; U_in_5=U_in_5'; %Normal Distribution located at center
```

```

U_in_6(1:50)=0:1:49; U_in_6(51:101)=linspace(50,0,51); U_in_6=U_in_6*rand*rand;
U_in_6=U_in_6'; %Pyramid
U_in_7((n-1)/4+1)=rand*1000; %source located off center
U_in_8=[sin(b*2*pi/20)*rand*1000 sin(b*2*pi/20)*rand*1000 sin(b*2*pi/20)*rand*1000
sin(b*2*pi/20)*rand*1000 sin(b*2*pi/20)*rand*1000]; U_in_8=U_in_8-
ones(1,length(U_in_8))*min(U_in_8)*2; U_in_8(102:105)=[]; U_in_8=U_in_8'; %Sinusoidal

U_in=[U_in_1 U_in_2 U_in_3 U_in_4 U_in_5 U_in_6];

D=0.1;

% Mark Nodes
left=1;
inner=2:n-1;
right=n;

% Experiment 1
time=t_final;
[t1,u1]=ode15s(@(t,y) Diffusion(t,y,x,D),[0 time],U_in(:,1));
[t2,u2]=ode15s(@(t,y) Diffusion(t,y,x,D),[0 time],U_in(:,2));
[t3,u3]=ode15s(@(t,y) Diffusion(t,y,x,D),[0 time],U_in(:,3));
[t4,u4]=ode15s(@(t,y) Diffusion(t,y,x,D),[0 time],U_in(:,4));
[t5,u5]=ode15s(@(t,y) Diffusion(t,y,x,D),[0 time],U_in(:,5));
[t6,u6]=ode15s(@(t,y) Diffusion(t,y,x,D),[0 time],U_in(:,6));

[Ux_1,Uxx_1]=Dif_der(x,u1,t1);
[Ux_2,Uxx_2]=Dif_der(x,u2,t2);
[Ux_3,Uxx_3]=Dif_der(x,u3,t3);
[Ux_4,Uxx_4]=Dif_der(x,u4,t4);
[Ux_5,Uxx_5]=Dif_der(x,u5,t5);
[Ux_6,Uxx_6]=Dif_der(x,u6,t6);

```

```

function dydt=Diffusion(t,y,x,D)

n=length(x);
dh=x(2)-x(1);

dydt=zeros(n,1);

y_new=y;

dydt(1)=(2*y_new(2)-2*y_new(1))/(dh^2);
for m=2:(length(x)-1)
    dydt(m)=(y_new(m+1)+y_new(m-1)-2*y_new(m))/(dh^2);
end
dydt(length(x))=(2*y_new(length(x)-1)-2*y_new(length(x)))/(dh^2);

dydt=D*dydt;

```

```

function [Ux,Uxx]=Dif_der(x,u,t)

n=length(x);
dh=x(2)-x(1);

for i=1:length(t)
    U_new=u(i,:);

    Ux(i,1)=0;

```

```

for m=2:(length(x)-1)
    Ux(i,m)=(U_new(m+1)-U_new(m))/(dh);
end
Ux(i,length(x))=0;

Uxx(i,1)=(2*U_new(2)-2*U_new(1))/(dh^2);
for m=2:(length(x)-1)
    Uxx(i,m)=(U_new(m+1)+U_new(m-1)-2*U_new(m))/(dh^2);
end
Uxx(i,length(x))=(2*U_new(length(x)-1)-2*U_new(length(x)))/(dh^2);

end

```

#### 6.4 Κώδικας επίλυσης προβλήματος Bratu σε μεταβατική κατάσταση, σε μία διάσταση

```

%Bratu
clear;clc;

% Define Grid
dh=0.01 ;
L=1 ;
x=0:dh:L; x(end)=L;
n=length(x); % nodes in x direction

% Time domain
dt=10^-5;
t_final=0.01 ;
% t=(10^-4:dt:t_final);

% Initial Conditions
U_in_1=zeros(n,1);
U_in_4=zeros(n,1);U_in_7=zeros(n,1); U_in_8=zeros(n,1);
b=0:20;

U_in_1((n+1)/2)=rand; %source located at center
U_in_2=[b*rand b*rand b*rand b*rand b*rand]; U_in_2(102:105)=[]; U_in_2=U_in_2/100;
U_in_2=U_in_2'; %Saw
U_in_3=[0 zeros(1,10) ones(1,10)*rand zeros(1,10) ones(1,10)*rand zeros(1,10)
ones(1,10)*rand zeros(1,10) ones(1,10)*rand zeros(1,10) ones(1,10)]; U_in_3=U_in_3';
%Steps
U_in_4((n+1)/2)=rand*10; %one larger source
U_in_5=normpdf(x,5/10,1/10)/1000; U_in_5=U_in_5'; %Normal Distribution located at
center
U_in_6(1:50)=0:1:49; U_in_6(51:101)=linspace(50,0,51); U_in_6=U_in_6*rand*rand;
U_in_6=U_in_6'; %Pyramid
U_in_7((n-1)/4+1)=rand; %source located off center
U_in_8=[sin(b*2*pi/20)*rand sin(b*2*pi/20)*rand sin(b*2*pi/20)*rand
sin(b*2*pi/20)*rand sin(b*2*pi/20)*rand]; U_in_8=U_in_8-
ones(1,length(U_in_8))*min(U_in_8)*2; U_in_8(102:105)=[]; U_in_8=U_in_8'; %Sinusoidal

U_in=[U_in_1 U_in_2 U_in_3 U_in_4 U_in_5 U_in_6];
U_in(1,:)=zeros(1,6); U_in(end,:)=zeros(1,6);

D=1;
lamda=1;

```

```

% Mark Nodes
left=1;
inner=2:n-1;
right=n;

% Experiment
time=t_final;
[t1,u1]=ode15s(@(t,y) Bratu(t,y,x,D,lamda),[0 time],U_in(:,1));
[t2,u2]=ode15s(@(t,y) Bratu(t,y,x,D,lamda),[0 time],U_in(:,2));
[t3,u3]=ode15s(@(t,y) Bratu(t,y,x,D,lamda),[0 time],U_in(:,3));
[t4,u4]=ode15s(@(t,y) Bratu(t,y,x,D,lamda),[0 time],U_in(:,4));
[t5,u5]=ode15s(@(t,y) Bratu(t,y,x,D,lamda),[0 time],U_in(:,5));
[t6,u6]=ode15s(@(t,y) Bratu(t,y,x,D,lamda),[0 time],U_in(:,6));

[Ux_1,Uxx_1]=Bratu_der(x,u1,t1);
[Ux_2,Uxx_2]=Bratu_der(x,u2,t2);
[Ux_3,Uxx_3]=Bratu_der(x,u3,t3);
[Ux_4,Uxx_4]=Bratu_der(x,u4,t4);
[Ux_5,Uxx_5]=Bratu_der(x,u5,t5);
[Ux_6,Uxx_6]=Bratu_der(x,u6,t6);

```

```

function dydt=Bratu(t,y,x,D,lamda)

n=length(x);
dh=x(2)-x(1);

dydt=zeros(n,1);

y_new=y;
y_new(1)=0; y_new(end)=0;

for m=2:(length(x)-1)
    dydt(m)=(y_new(m+1)+y_new(m-1)-2*y_new(m))/(dh^2);
end

dydt=D*dydt+lamda*exp(y_new);
dydt(1)=0; dydt(end)=0;

```

```

function [Ux,Uxx]=Bratu_der(x,u,t)

n=length(x);
dh=x(2)-x(1);

for i=1:length(t)
    U_new=u(i,:);

    Ux(i,1)=0;
    for m=2:(length(x)-1)
        Ux(i,m)=(U_new(m+1)-U_new(m))/(dh);
    end
    Ux(i,length(x))=0;

    Uxx(i,1)=0;
    for m=2:(length(x)-1)
        Uxx(i,m)=(U_new(m+1)+U_new(m-1)-2*U_new(m))/(dh^2);
    end
    Uxx(i,length(x))=0;

```

end

## 6.5 Κώδικας επίλυσης μοντέλου τυχαίων βηματιστών

```
%Walkers2Diff_2
clear;clc;

% number of random walkers
Nw=10000;

% Define Grid
dh=1 ;
L=100 ;
x=0:dh:L; x(end)=L;
n=length(x); % number of nodes in x direction

% Time domain
dt=1e-4;
t_final=1 ;
t=(1e-4:dt:t_final);

% Initial Conditions
U_in_1=zeros(n,1);U_in_4=zeros(n,1);U_in_7=zeros(n,1); U_in_8=zeros(n,1);
b=0:20;

U_in_1((n+1)/2)=rand*1000; %source located at center
U_in_2=[b*0 b*rand*100 b*rand*100 b*rand*100 b*0]; U_in_2(102:105)=[]; U_in_2=U_in_2';
%Saw
U_in_3=[zeros(1,22) ones(1,8)*rand*1000 zeros(1,8) ones(1,9)*rand*1000 zeros(1,8)
ones(1,8)*rand*1000 zeros(1,8) ones(1,8)*rand*1000 zeros(1,22)]; U_in_3=U_in_3';
%Steps
U_in_4((n+1)/2)=rand*1000; %one larger source
U_in_5=normpdf(x,50,10)*1000; U_in_5=U_in_5'; %Normal Distribution located at center
U_in_6(1:20)=zeros(1,20); U_in_6(21:50)=0:1:29; U_in_6(51:81)=linspace(29,0,31);
U_in_6(82:101)=zeros(1,20); U_in_6=U_in_6*rand*10; U_in_6=U_in_6'; %Pyramid
U_in_7((n-1)/4+1)=rand*1000; %source located off center
U_in_8=[b*0 sin(b*2*pi/20)*rand*1000 sin(b*2*pi/20)*rand*1000 sin(b*2*pi/20)*rand*1000
b*0]; U_in_8=U_in_8-ones(1,length(U_in_8))*min(U_in_8)*2; U_in_8(102:105)=[];
U_in_8(1:21)=zeros(1,21); U_in_8(85:101)=zeros(1,17); U_in_8=U_in_8'; %Sinusoidal

s_1=trapz(x,U_in_1);
s_2=trapz(x,U_in_2);
s_3=trapz(x,U_in_3);
s_4=trapz(x,U_in_4);
s_5=trapz(x,U_in_5);
s_6=trapz(x,U_in_6);
s_7=trapz(x,U_in_7);
s_8=trapz(x,U_in_8);

U_in=[U_in_1 U_in_2 U_in_3 U_in_4 U_in_5 U_in_6 U_in_7 U_in_8];

X_in_1=Converter(x,U_in_1,Nw)';
X_in_2=Converter(x,U_in_2,Nw)';
X_in_3=Converter(x,U_in_3,Nw)';
X_in_4=Converter(x,U_in_4,Nw)';
X_in_5=Converter(x,U_in_5,Nw)';
```



```

X_in_6=Converter(x,U_in_6,Nw)';
X_in_7=Converter(x,U_in_7,Nw)';
X_in_8=Converter(x,U_in_8,Nw)';

X_in=[X_in_1' X_in_2' X_in_3' X_in_4' X_in_5' X_in_6' X_in_7' X_in_8'];

% Number of steps
nsteps = length(t);
D=10;

dx=sqrt(2*dt*D); %Monte Carlo step

N_t=10; %Amount of time steps at which we sampled y values

% Experiment
k_tr=100; %Repetitions of each exp for mean value

y_1=R_W_y(X_in_1,k_tr,N_t,dx,nsteps,Nw,x)*s_1;
y_2=R_W_y(X_in_2,k_tr,N_t,dx,nsteps,Nw,x)*s_2;
y_3=R_W_y(X_in_3,k_tr,N_t,dx,nsteps,Nw,x)*s_3;
y_4=R_W_y(X_in_4,k_tr,N_t,dx,nsteps,Nw,x)*s_4;
y_5=R_W_y(X_in_5,k_tr,N_t,dx,nsteps,Nw,x)*s_5;
y_6=R_W_y(X_in_6,k_tr,N_t,dx,nsteps,Nw,x)*s_6;

[yx_1,yxx_1,yt_1]=R_W_Deriv(N_t,x,dt,y_1);
[yx_2,yxx_2,yt_2]=R_W_Deriv(N_t,x,dt,y_2);
[yx_3,yxx_3,yt_3]=R_W_Deriv(N_t,x,dt,y_3);
[yx_4,yxx_4,yt_4]=R_W_Deriv(N_t,x,dt,y_4);
[yx_5,yxx_5,yt_5]=R_W_Deriv(N_t,x,dt,y_5);
[yx_6,yxx_6,yt_6]=R_W_Deriv(N_t,x,dt,y_6);

for i=1:N_t
    yxx_1(i,:)=sgolayfilt(yxx_1(i,:),1,7);
    yxx_2(i,:)=sgolayfilt(yxx_2(i,:),1,7);
    yxx_3(i,:)=sgolayfilt(yxx_3(i,:),1,7);
    yxx_4(i,:)=sgolayfilt(yxx_4(i,:),1,7);
    yxx_5(i,:)=sgolayfilt(yxx_5(i,:),1,7);
    yxx_6(i,:)=sgolayfilt(yxx_6(i,:),1,7);
end

for i=N_t-1
    yt_1(i,:)=sgolayfilt(yt_1(i,:),1,7);
    yt_2(i,:)=sgolayfilt(yt_2(i,:),1,7);
    yt_3(i,:)=sgolayfilt(yt_3(i,:),1,7);
    yt_4(i,:)=sgolayfilt(yt_4(i,:),1,7);
    yt_5(i,:)=sgolayfilt(yt_5(i,:),1,7);
    yt_6(i,:)=sgolayfilt(yt_6(i,:),1,7);
end
toc

function [N_1,N_2,N_3,N_4,N_5,N_6,N_7,N_8,N_9,N_10 ]=R_W(k_tr,dx,nsteps,Nw,X)

N_1=zeros(k_tr,Nw); N_2=zeros(k_tr,Nw); N_3=zeros(k_tr,Nw); N_4=zeros(k_tr,Nw);
N_5=zeros(k_tr,Nw);
N_6=zeros(k_tr,Nw); N_7=zeros(k_tr,Nw); N_8=zeros(k_tr,Nw); N_9=zeros(k_tr,Nw);
N_10=zeros(k_tr,Nw);

Xin=X;

```

```

for k=1:k_tr
    x=xin;

    for i=1:nsteps

        p=rand(1,Nw);
        X = X + (2*(p>=0.5)-1)*dx;

        if i==1000
            N_1(k,:)=sort(X); %histc(x0,xpt);
        end
        if i==2000
            N_2(k,:)=sort(X); %histc(x0,xpt);
        end
        if i==3000
            N_3(k,:)=sort(X); %histc(x0,xpt);
        end
        if i==4000
            N_4(k,:)=sort(X); %histc(x0,xpt);
        end
        if i==5000
            N_5(k,:)=sort(X); %histc(x0,xpt);
        end
        if i==6000
            N_6(k,:)=sort(X); %histc(x0,xpt);
        end
        if i==7000
            N_7(k,:)=sort(X); %histc(x0,xpt);
        end
        if i==8000
            N_8(k,:)=sort(X); %histc(x0,xpt);
        end
        if i==9000
            N_9(k,:)=sort(X); %histc(x0,xpt);
        end
        if i==10000
            N_10(k,:)=sort(X); %histc(x0,xpt);
        end
    end
end
end

```

```
function [y]=R_w_y(X,k_tr,N_t,dx,nsteps,Nw,xpt)
```

```

y=zeros(N_t,length(xpt));
y_N=(1:Nw)/Nw;

X=X;

[N_1,N_2,N_3,N_4,N_5,N_6,N_7,N_8,N_9,N_10]=R_w(k_tr,dx,nsteps,Nw,X);
%

M_1=mean(N_1,1); [i_1,j_1]=unique(M_1);
y(1,:)=interp1(M_1(j_1),y_N(j_1),xpt,'nearest','extrap');
M_2=mean(N_2,1); [i_2,j_2]=unique(M_2);
y(2,:)=interp1(M_2(j_2),y_N(j_2),xpt,'nearest','extrap');
M_3=mean(N_3,1); [i_3,j_3]=unique(M_3);

```

```

y(3,:)=interp1(M_3(j_3),y_N(j_3),xpt,'nearest','extrap');
M_4=mean(N_4,1); [i_4,j_4]=unique(M_4);
y(4,:)=interp1(M_4(j_4),y_N(j_4),xpt,'nearest','extrap');
M_5=mean(N_5,1); [i_5,j_5]=unique(M_5);
y(5,:)=interp1(M_5(j_5),y_N(j_5),xpt,'nearest','extrap');
M_6=mean(N_1,1); [i_6,j_6]=unique(M_6);
y(6,:)=interp1(M_6(j_1),y_N(j_6),xpt,'nearest','extrap');
M_7=mean(N_2,1); [i_7,j_7]=unique(M_7);
y(7,:)=interp1(M_7(j_2),y_N(j_7),xpt,'nearest','extrap');
M_8=mean(N_3,1); [i_8,j_8]=unique(M_8);
y(8,:)=interp1(M_8(j_3),y_N(j_8),xpt,'nearest','extrap');
M_9=mean(N_4,1); [i_9,j_9]=unique(M_9);
y(9,:)=interp1(M_9(j_4),y_N(j_9),xpt,'nearest','extrap');
M_10=mean(N_5,1); [i_10,j_10]=unique(M_10);
y(10,:)=interp1(M_10(j_10),y_N(j_10),xpt,'nearest','extrap');

```

```

function [yx,yxx,yt]=R_W_Deriv(N_t,xpt,dt,y)

```

```

yx=zeros(N_t,length(xpt)); yxx=zeros(N_t,length(xpt)); yt=zeros(N_t-1,length(xpt));

```

```

np=length(xpt);
dh=xpt(2)-xpt(1);

```

```

for i=2:(np-1)
    yx(1,i)=(y(1,i+1)-y(1,i-1))/(2*dh);
    yx(2,i)=(y(2,i+1)-y(2,i-1))/(2*dh);
    yx(3,i)=(y(3,i+1)-y(3,i-1))/(2*dh);
    yx(4,i)=(y(4,i+1)-y(4,i-1))/(2*dh);
    yx(5,i)=(y(5,i+1)-y(5,i-1))/(2*dh);
    yx(6,i)=(y(6,i+1)-y(6,i-1))/(2*dh);
    yx(7,i)=(y(7,i+1)-y(7,i-1))/(2*dh);
    yx(8,i)=(y(8,i+1)-y(8,i-1))/(2*dh);
    yx(9,i)=(y(9,i+1)-y(9,i-1))/(2*dh);
    yx(10,i)=(y(10,i+1)-y(10,i-1))/(2*dh);
end

```

```

for i=2:(np-1)
    yxx(1,i)=(y(1,i+1)+y(1,i-1)-2*y(1,i))/(dh^2);
    yxx(2,i)=(y(2,i+1)+y(2,i-1)-2*y(2,i))/(dh^2);
    yxx(3,i)=(y(3,i+1)+y(3,i-1)-2*y(3,i))/(dh^2);
    yxx(4,i)=(y(4,i+1)+y(4,i-1)-2*y(4,i))/(dh^2);
    yxx(5,i)=(y(5,i+1)+y(5,i-1)-2*y(5,i))/(dh^2);
    yxx(6,i)=(y(6,i+1)+y(6,i-1)-2*y(6,i))/(dh^2);
    yxx(7,i)=(y(7,i+1)+y(7,i-1)-2*y(7,i))/(dh^2);
    yxx(8,i)=(y(8,i+1)+y(8,i-1)-2*y(8,i))/(dh^2);
    yxx(9,i)=(y(9,i+1)+y(9,i-1)-2*y(9,i))/(dh^2);
    yxx(10,i)=(y(10,i+1)+y(10,i-1)-2*y(10,i))/(dh^2);
end

```

```

dt=dt*1000;
for i=1:np
    yt(1,i)=(y(2,i)-y(1,i))/dt;
    yt(2,i)=(y(3,i)-y(2,i))/dt;
    yt(3,i)=(y(4,i)-y(3,i))/dt;
    yt(4,i)=(y(5,i)-y(4,i))/dt;
    yt(5,i)=(y(6,i)-y(5,i))/dt;
    yt(6,i)=(y(7,i)-y(6,i))/dt;
    yt(7,i)=(y(8,i)-y(7,i))/dt;
end

```

```

yt(8,i)=(y(9,i)-y(8,i))/dt;
yt(9,i)=(y(10,i)-y(9,i))/dt;
end

```

## 6.6 Κώδικας για εφαρμογή Γκαουσιανών Διεργασιών

```

x_1=x(30:70);
p=randperm(length(x_1),20);
x_2=x_1(p);

LG_idx_1=[1 2 round(logspace(log10(5),log10(length(t1)),8),0)];
LG_idx_2=[1 2 round(logspace(log10(5),log10(length(t2)),8),0)];
LG_idx_3=[1 2 round(logspace(log10(5),log10(length(t3)),8),0)];
LG_idx_4=[1 2 round(logspace(log10(5),log10(length(t4)),8),0)];
LG_idx_5=[1 2 round(logspace(log10(5),log10(length(t5)),8),0)];
LG_idx_6=[1 2 round(logspace(log10(5),log10(length(t6)),8),0)];

% Gauss Process
U_st=[reshape((u1(LG_idx_1,x_2)),200,1); reshape((u2(LG_idx_2,x_2)),200,1);
reshape((u3(LG_idx_3,x_2)),200,1); reshape((u4(LG_idx_4,x_2)),200,1);
reshape((u5(LG_idx_5,x_2)),200,1); reshape((u6(LG_idx_6,x_2)),200,1)];
Ux_st=[reshape((Ux_1(LG_idx_1,x_2)),200,1); reshape((Ux_2(LG_idx_2,x_2)),200,1);
reshape((Ux_3(LG_idx_3,x_2)),200,1); reshape((Ux_4(LG_idx_4,x_2)),200,1);
reshape((Ux_5(LG_idx_5,x_2)),200,1); reshape((Ux_6(LG_idx_6,x_2)),200,1)];
Uxx_st=[reshape((Uxx_1(LG_idx_1,x_2)),200,1); reshape((Uxx_2(LG_idx_2,x_2)),200,1);
reshape((Uxx_3(LG_idx_3,x_2)),200,1); reshape((Uxx_4(LG_idx_4,x_2)),200,1);
reshape((Uxx_5(LG_idx_5,x_2)),200,1); reshape((Uxx_6(LG_idx_6,x_2)),200,1)];
Ut_st=D*Uxx_st;
mat=[U_st Ux_st Uxx_st];
GP=fitrgrp(mat,Ut_st,'KernelFunction','ardsquaredexponential');
sigmaK=GP.KernelInformation.KernelParameters(1:end-1,1);
el_k=log(sigmaK);

```

## 6.7 Κώδικας εκπαίδευσης νευρωνικών δικτύων

```

% Solve an Input-Output Fitting problem with a Neural Network
% Script generated by Neural Fitting app
%
% This script assumes these variables are defined:
%
% Uxx_st - input data.
% Ut_st - target data.

x = Uxx_st';
t = Ut_st';

% Choose a Training Function
% For a list of all training functions type: help nntrain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. Suitable in low memory situations.
trainFcn = 'trainlm'; % Levenberg-Marquardt backpropagation.

% Create a Fitting Network
H1 = 1;
H2 = 3;

```

```

net = fitnet([H1 H2],trainFcn);

% Choose Input and Output Pre/Post-Processing Functions
% For a list of all processing functions type: help nprocess
net.input.processFcns = {'removeconstantrows','mapminmax'};
net.output.processFcns = {'removeconstantrows','mapminmax'};

% Setup Division of Data for Training, Validation, Testing
% For a list of all data division functions type: help nndivision
net.divideFcn = 'dividerand'; % Divide data randomly
net.divideMode = 'sample'; % Divide up every sample
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% Choose a Performance Function
% For a list of all performance functions type: help nperformance
net.performFcn = 'mse'; % Mean Squared Error

% Choose Plot Functions
net.plotFcns = {'plotperform','plotregression'};
net.layers{1}.transferFcn = 'tansig';
net.layers{2}.transferFcn = 'purelin';

%Training parameters.
%net.trainParam.min_grad=1e-5; %minimum gradient. Under this value,the training stops.
%net.trainParam.show=10; %Number of epochs after the plots are performed.
%net.trainParam.lr=1; %Learning rate,
%net.trainParam.lr_inc=1.05; % Increasing learning rate value.
%net.trainParam.lr_dec=0.7; % Decreasing learning rate value.
net.trainParam.epochs=10000; % Number of epochs.
%net.trainParam.goal=0.0; % MSE goal value.
%net.trainParam.max_fail=6; % if the MSE of validation set does not decrease after 6
epochs, the training stops.

% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)

% view the Network
view(net)

% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotregression(t,y)
%figure, plotfit(net,x,t)

```

## 6.8 Κώδικας για αξιολόγηση νευρωνικών δικτύων για τα προβλήματα Δ.Ε.

```

%Neural Network Test
clear;clc;

% Define Grid
dh=1 ;
L=100 ; % dh=0.01 and L=1 for Bratu
x=0:dh:L; x(end)=L;
n=length(x); % number of nodes in x direction

% Initial Conditions
b=0:20;
U_in_7=zeros(n,1);
U_in_7((n-1)/4+1)=rand*1000; %source located off center
U_in_8=[sin(b*2*pi/20)*rand*1000 sin(b*2*pi/20)*rand*1000 sin(b*2*pi/20)*rand*1000
sin(b*2*pi/20)*rand*1000 sin(b*2*pi/20)*rand*1000]; U_in_8=U_in_8-
ones(1,length(U_in_8))*min(U_in_8)*2; U_in_8(102:105)=[]; U_in_8=U_in_8'; %Sinusoidal

D=0.1;

% NN-testing
y=U_in_7;
time=100;
load('net location on computer'); % call the net you want to test
[t,u0]=ode15s(@(t,y) Diffusion(t,y,x,D),[0 time],y); %Bratu αντι για Diffusion για την
αξιολόγηση NN στο πρόβλημα Bratu
figure(1); plot(x,u0(end,:), 'k')
hold on
[t,u1]=ode15s(@(t,y) diffnet(t,y,x,network7),[0 time],y); % diffnetBratu αντι για
diffnet για την αξιολόγηση NN στο πρόβλημα Bratu
plot(x,u1(end,:), 'ro', "MarkerIndices", [1:5:101])
)
% plot(x,y,'--','color',[0 0.4470 07410])
hold off
AUF=abs((trapez(u0(end,[2:n-1]))-trapez(u1(end,[2:n-1])))/trapez(u0(end,[2:n-1])));
MAPE=sum(abs((u0(end,[2:n-1])-u1(end,[2:n-1])))/(u0(end,[2:n-1]))))*100/99;

function dydt=diffnet(t,y,x,net)

n=length(x);
dh=x(2)-x(1);

y_new=y;

Uxx=zeros(n,1);

Uxx(1)=(2*y_new(2)-2*y_new(1))/(dh^2);
for m=2:(length(x)-1)
    Uxx(m)=(y_new(m+1)+y_new(m-1)-2*y_new(m))/(dh^2);
end
Uxx(length(x))=(2*y_new(length(x)-1)-2*y_new(length(x)))/(dh^2);

dydt=net(Uxx');
dydt=dydt';

```

```

function dydt=diffnetBratu(t,y,x,A_U,net)

n=length(x);
dh=x(2)-x(1);

U_new=y;
U_new(1)=0; U_new(end)=0;

Uxx(1)=(U_new(3)-2*U_new(2))/(dh^2);
for m=2:(length(x)-1)
    Uxx(m)=(U_new(m+1)+U_new(m-1)-2*U_new(m))/(dh^2);
end
Uxx(length(x))=(U_new(length(x)-2)-2*U_new(length(x)-1))/(dh^2);

U_nn=[U_new';Uxx];
dydt=net(U_nn);
dydt=dydt';

```

## 6.9 Κώδικας για αξιολόγηση νευρωνικών δικτύων για το μοντέλο τυχαίων βηματιστών

```

%Neural Network Test
clear;clc;

% Define Grid
dh=1 ;
L=100 ; % dh=0.01 and L=1 for Bratu
x=0:dh:L; x(end)=L;
n=length(x); % number of nodes in x direction

% Initial Conditions
b=0:20;
U_in_5=normpdf(x,50,10)*1000; U_in_5=U_in_5'; %Normal Distribution located at center
U_in_8=[sin(b*2*pi/20)*rand*1000 sin(b*2*pi/20)*rand*1000 sin(b*2*pi/20)*rand*1000
sin(b*2*pi/20)*rand*1000 sin(b*2*pi/20)*rand*1000]; U_in_8=U_in_8-
ones(1,length(U_in_8))*min(U_in_8)*2; U_in_8(102:105)=[]; U_in_8=U_in_8'; %Sinusoidal

D=0.1;
Nw=10^4;

% NN-testing
U=U_in_8;
s=trapz(x,U);
X=Converter(x,U,Nw)';
y=X;
y_N=(1:Nw)/Nw;
y=sort(y);
[i_1,j_1]=unique(y); y=interp1(y(j_1),y_N(j_1),x,'nearest','extrap')*s;

time=0.1;
load('net location on computer');
u0=RW(X,dx,time,x,Nw,D)*s;
figure(7); subplot(2,1,1), plot(x,u0,'k')
hold on
[t,u1]=ode15s(@(t,y) Rwnet(t,y,x,Nw,network1),[0 time],y);
plot(x,u1(end,:), 'ro')

```

```

% plot(x,y,'--','color',[0 0.4470 0.7410])
hold off
AUF=abs((trapez(u0(2:n-1))-trapez(u1(end,[2:n-1])))/trapez(u0(2:n-1)));
MAPE=sum(abs((u0(2:n-1)-u1(end,[2:n-1]))/(u0(2:n-1))))*100/99;

function u0=RW(X,dx,time,x,Nw,D)

dt=1e-4;
t_final=time ;
t=(1e-4:dt:t_final);
nsteps = length(t);

for i=1:nsteps
    p=rand(1,Nw);
    x = x + (2*(p>=0.5)-1)*dx;
end
N=sort(x)';

y_N=(1:Nw)/Nw;

[i_1,j_1]=unique(N); u0=interp1(N(j_1),y_N(j_1),x,'nearest','extrap');
u0=sgolayfilt(u0,1,7);

dh=x(2)-x(1);

Uxx=zeros(1,length(x));
Uxx(1)=(U_new(3)-2*U_new(2))/(dh^2);
for m=2:(length(x)-1)
    Uxx(m)=(U_new(m+1)+U_new(m-1)-2*U_new(m))/(dh^2);
end
Uxx(length(x))=(U_new(length(x)-2)-2*U_new(length(x)-1))/(dh^2);

dydt=D*Uxx;
dydt=sgolayfilt(dydt,1,7);

function dydt=RWnet(t,y,x,Nw,net)

dh=x(2)-x(1);
U_new=y;

Uxx(1)=(U_new(3)-2*U_new(2))/(dh^2);
for m=2:(length(x)-1)
    Uxx(m)=(U_new(m+1)+U_new(m-1)-2*U_new(m))/(dh^2);
end
Uxx(length(x))=(U_new(length(x)-2)-2*U_new(length(x)-1))/(dh^2);

Uxx=sgolayfilt(Uxx,

```