

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ  
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
Τμήμα Ηλεκτρολόγων & Ηλεκτρονικών Μηχανικών



[www.eee.uniwa.gr](http://www.eee.uniwa.gr)

Θηβών 250, Αθήνα-Αιγάλεω 12244  
Τηλ. +30 210 538-1225, Fax. +30 210 538-1226

Πρόγραμμα Μεταπτυχιακών Σπουδών  
Ηλεκτρικές & Ηλεκτρονικές Επιστήμες μέσω  
Έρευνας

UNIVERSITY of WEST ATTICA  
FACULTY OF ENGINEERING  
Department of Electrical & Electronics  
Engineering

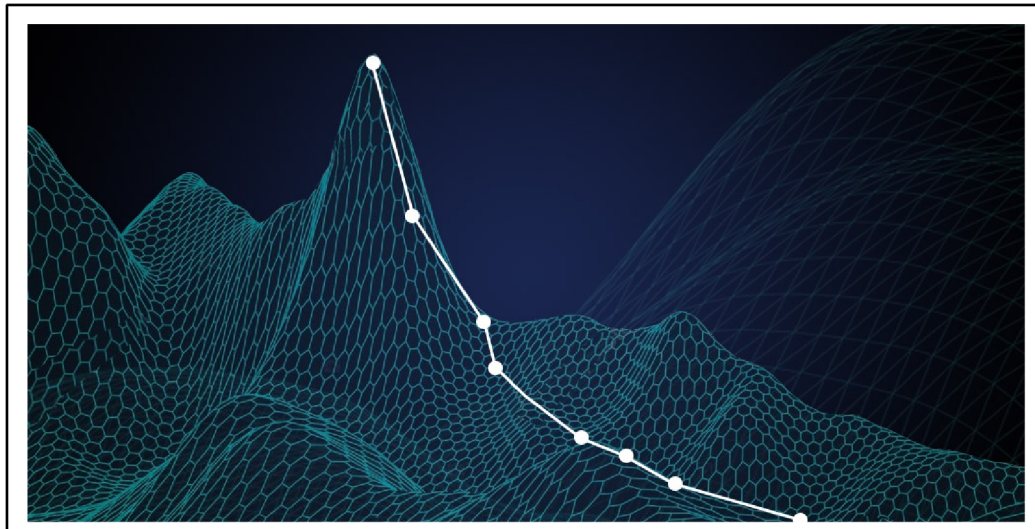
[www.eee.uniwa.gr](http://www.eee.uniwa.gr)

250, Thivon Str., Athens, GR-12244, Greece  
Tel: +30 210 538-1225, Fax: +30 210 538-1226

Master of Science By Research in  
Electrical & Electronics Engineering

## ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**« Βελτιστοποίηση Τεχνικών Εκπαίδευσης Νευρωνικών δικτύων  
εμπρόσθιας τροφοδότησης για την επίλυση Διαφορικών Εξισώσεων ».**



Μεταπτυχιακή Φοιτήτρια: Βασιλική Καλούτσα, ΑΜ 0009  
Επιβλέπων: Ιωάννης Θ. Φαμέλης, Καθηγητής

ΑΙΓΑΛΕΩ, 15/02/2020

---

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ  
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
Τμήμα Ηλεκτρολόγων & Ηλεκτρονικών Μηχανικών



[www.eee.uniwa.gr](http://www.eee.uniwa.gr)

Θηβών 250, Αθήνα-Αιγάλεω 12244  
Τηλ. +30 210 538-1225, Fax. +30 210 538-1226

Πρόγραμμα Μεταπτυχιακών Σπουδών  
Ηλεκτρικές & Ηλεκτρονικές Επιστήμες μέσω  
Έρευνας

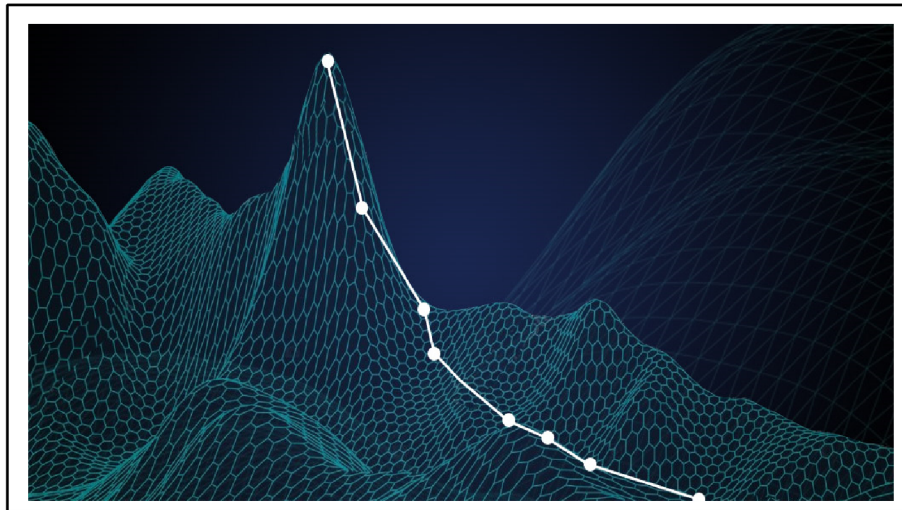
UNIVERSITY of WEST ATTICA  
FACULTY OF ENGINEERING  
Department of Electrical & Electronics  
Engineering  
[www.eee.uniwa.gr](http://www.eee.uniwa.gr)

250, Thivon Str., Athens, GR-12244, Greece  
Tel: +30 210 538-1225, Fax: +30 210 538-1226

Master of Science By Research in  
*Electrical & Electronics Engineering*

## MSc Thesis

***“ Optimization of Feedforward Neural Network Techniques  
for solving Differential Equations ”.***



Student: Kaloutsa, Vasiliki, Registration Number 0009

MSc Thesis Supervisor: Famelis, Ioannis, Professor

ATHENS-EGALEO, 15/02/2020

Η Μεταπτυχιακή Διπλωματική Εργασία έγινε αποδεκτή, εξετάστηκε και βαθμολογήθηκε από την εξής τριμελή εξεταστική επιτροπή:

Επιβλέπων/ουσα	Μέλος	Μέλος
Ιωάννης Θ. Φαμέλης	Αλέξανδρος Αλεξανδρίδης	Στυλιανός Ποτηράκης
Καθηγητής	Καθηγητής	Καθηγητής

## **ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ**

Η κάτωθι υπογεγραμμένη ΚΑΛΟΥΤΣΑ ΒΑΣΙΛΙΚΗ του ΔΗΜΗΤΡΙΟΥ, με αριθμό μητρώου ΑΜ0009 φοιτήτρια του Προγράμματος Μεταπτυχιακών Σπουδών “ΗΛΕΚΤΡΙΚΕΣ & ΗΛΕΚΤΡΟΝΙΚΕΣ ΕΠΙΣΤΗΜΕΣ ΜΕΣΩ ΕΡΕΥΝΑΣ” του Τμήματος ΗΛΕΚΤΡΟΛΟΓΩΝ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ της Σχολής ΜΗΧΑΝΙΚΩΝ του Πανεπιστημίου Δυτικής Αττικής, δηλώνω ότι:

«Είμαι η συγγραφέας αυτής της μεταπτυχιακής διπλωματικής εργασίας και κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του τίτλου μου».

Η Δηλούσα



Μεταπτυχιακή Φοιτήτρια

ΚΑΛΟΥΤΣΑ ΒΑΣΙΛΙΚΗ

## ΠΕΡΙΛΗΨΗ

Οι Διαφορικές Εξισώσεις αποτελούν μαθηματικά εργαλεία για τη μοντελοποίηση πληθώρας προβλημάτων τα οποία περιέχουν ποσότητες που μεταβάλλονται και τα οποία συναντώνται σε πολλούς τομείς όπως αυτούς της μηχανικής, της οικονομίας, της βιολογίας και της κοινωνιολογίας. Η εύρεση επομένως αποδοτικών μεθόδων επίλυσης, οι οποίες έχουν όσο το δυνατόν μικρότερο υπολογιστικό κόστος, καθίσταται αναγκαία. Μία αρκετά σύγχρονη μέθοδος για την επίλυση Διαφορικών Εξισώσεων, είναι η χρήση Τεχνητών Νευρωνικών Δικτύων (ΤΝΔ). Στη συγκεκριμένη εργασία γίνεται μια προσπάθεια για τη βελτιστοποίηση τεχνικών εκπαίδευσης ΤΝΔ εμπρόσθιας τροφοδότησης για την επίλυση συγκεκριμένων κλάσεων Διαφορικών Εξισώσεων. Οι Διαφορικές Εξισώσεις που επιλύονται ανήκουν σε μία ιδιαίτερη κατηγορία προβλημάτων, αυτή των άκαμπτων προβλημάτων αρχικών τιμών (Π.Α.Τ.), η οποία παρουσιάζει δυσκολίες στην επίλυση με τη χρήση κλασικών μεθόδων. Επιπρόσθετα, επιλύονται άκαμπτα προβλήματα τα οποία περιέχουν παράμετρο ή οικογένεια παραμέτρων, η οποία επηρεάζει το πόσο άκαμπτο είναι το πρόβλημα. Από τα αποτελέσματα της έρευνας, προκύπτει το συμπέρασμα ότι η αρχιτεκτονική και η μέθοδος εκπαίδευσης των ΤΝΔ που χρησιμοποιήθηκαν για την επίλυση των προαναφερθέντων προβλημάτων, παράγουν ικανοποιητικές προσεγγιστικές λύσεις, με πολύ καλές ιδιότητες γενίκευσης, όπως επίσης ότι μπορούν τα ΤΝΔ να εκπαιδευθούν ώστε να λυθούν κλάσεις τέτοιων προβλημάτων που εξαρτώνται από την παράμετρο.

**ΛΕΞΕΙΣ – ΚΛΕΙΔΙΑ:** *Τεχνητά Νευρωνικά Δίκτυα, Διαφορικές Εξισώσεις, άκαμπτα προβλήματα, προβλήματα με παράμετρο, προσεγγιστικές λύσεις*

## ABSTRACT

Differential Equations are mathematical tools, used for modelling of a majority of problems that contain quantities that change. These problems have a variety of applications in many fields, such as mechanics, economy, biology and sociology. Thus, efficient solving methods, with low computational cost are urgent. A modern method for solving Differential Equations is the use of Artificial Neural Networks (ANN). The present Thesis is an effort of Optimization of Feedforward Neural Network Techniques for solving a specific class of Differential Equations. The Differential Equations that are solved are “stiff” problems and “stiff” problems present difficulties when are solved using classical methods. Furthermore, “stiff problems” that contain parameter or family of parameters, which affect the stiffness of the problem, are solved. The results of this research indicate that the architecture and the training method of ANN used to solve the aforementioned problems, produce satisfying approximate solutions with very good generalization properties and that ANN can be trained in order to solve types of problems that depend of the parameters.

**KEYWORDS:** *Artificial Neural Networks, Differential Equations, Stiff problems, problems with parameter, approximation method*

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Ιωάννη Θ. Φαμέλη για την πολύτιμη βοήθεια του και για την άψογη συνεργασία καθ' όλη τη διάρκεια του μεταπτυχιακού διπλώματος.

## ΠΙΝΑΚΑΣ ΣΥΜΒΟΛΩΝ-ΑΚΡΩΝΥΜΙΩΝ- ΣΥΝΤΟΜΟΓΡΑΦΙΩΝ

ΤΝΔ	Τεχνητά Νευρωνικά Δίκτυα
ΣΔΕ	Συνήθεις Διαφορικές Εξισώσεις
ΜΔΕ	Μερικές Διαφορικές Εξισώσεις
Π.Α.Τ.	Προβλήματα Αρχικών Τιμών



# ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Εισαγωγή .....	11
<b>Κεφάλαιο 1 : Θεωρητικό πλαίσιο του θέματος - Ανασκόπηση του πεδίου</b>	
1.1.1 Τεχνητή Νοημοσύνη.....	14
1.1.2 Τεχνητά Νευρωνικά Δίκτυα.....	14
1.1.3 Αρχιτεκτονική & τύποι Τεχνητών Νευρωνικών Δικτύων.....	16
1.1.4 Συναρτήσεις ενεργοποίησης .....	18
1.1.5 Τεχνητά Νευρωνικά Δίκτυα Εμπρόσθιας τροφοδότησης.....	20
1.1.6 Κυριότεροι αλγόριθμοι εκπαίδευσης .....	21
1.1.6.1 Error Back-propagation algorithm.....	22
1.1.6.2 Levenberg-Marquardt algorithm.....	23
1.1.7 Overfitting.....	24
1.2.1 Διαφορικές Εξισώσεις & μέθοδοι επίλυσης.....	25
1.2.2 Άκαμπτα (stiff) προβλήματα.....	26
1.2.3 Επίλυση Διαφορικών Εξισώσεων με Νευρωνικά.....	27
1.2.4 Μέθοδος Λαγάρη, Λίκα και Φωτιάδη.....	28
<b>Κεφάλαιο 2 : Μεθοδολογία της έρευνας .....</b>	<b>30</b>
<b>Κεφάλαιο 3 : Η προτεινόμενη μέθοδος – Θεμελίωση, Σχεδίαση, Ανάπτυξη</b>	
3.1 Τροποποίηση της αρχιτεκτονικής των Λαγάρη, Λίκα & Φωτιάδη.....	32
3.2 Παραμετρικοποιημένα προβλήματα.....	33
3.3 Περιγραφή κώδικα για την επίλυση ΣΔΕ με τη χρήση ΤΝΔ.....	34
<b>Κεφάλαιο 4 : Εφαρμογή και Αποτελέσματα</b>	
4.1 Άκαμπτα προβλήματα.....	36
4.2 Άκαμπτα προβλήματα με παράμετρο.....	41
4.3 Άκαμπτα προβλήματα με οικογένεια παραμέτρων.....	61
4.4 Έλεγχος overfitting.....	73
<b>Κεφάλαιο 5: Συμπεράσματα – Προτάσεις</b>	
5.1 Συμπεράσματα.....	77
5.2 Προτάσεις για μελλοντική έρευνα.....	78
<b>Βιβλιογραφία-Πηγές.....</b>	<b>80</b>
<b>Παράρτημα Α.....</b>	<b>82</b>
<b>Παράρτημα Β .....</b>	<b>91</b>

<b>Παράρτημα Γ</b> .....	<b>96</b>
<b>Παράρτημα Δ</b> .....	<b>99</b>

Οι Διαφορικές Εξισώσεις είναι ευρέως χρησιμοποιούμενα μαθηματικά εργαλεία, που χρησιμοποιούνται για τη μοντελοποίηση προβλημάτων, τα οποία περιλαμβάνουν ποσότητες που μεταβάλλονται στο χρόνο ή το χώρο. Τα προβλήματα αυτά ανήκουν στο χώρο της μηχανικής, της βιολογίας, της οικονομίας, κ.α. Η λύση των Διαφορικών Εξισώσεων δίνει τη δυνατότητα για πρόβλεψη του κόσμου, αφού οι Διαφορικές Εξισώσεις περιγράφουν προβλήματα όπως η αύξηση του πληθυσμού και η μεταβολή της απόδοσης μιας επένδυσης σε σχέση με το χρόνο.

Έπειτα από τη μοντελοποίηση των διαφόρων προβλημάτων, σημαντική είναι η επίλυση αυτών. Ωστόσο, η επίλυση των Διαφορικών Εξισώσεων δεν είναι πάντα μια εύκολη διαδικασία, καθώς η εύρεση της αναλυτικής λύσης είναι δύσκολη ή και αδύνατη πολλές φορές. Για αυτό το λόγο έχουν αναπτυχθεί μέθοδοι οι οποίες προσεγγίζουν την αναλυτική λύση. Μερικές από τις πιο γνωστές από αυτές τις μεθόδους για την επίλυση Συνήθων Διαφορικών Εξισώσεων (ΣΔΕ) δηλαδή εξισώσεων των οποίων οι συναρτήσεις αποτελούνται από μία ανεξάρτητη μεταβλητή, είναι οι: Runge-Kutta, Euler, Predictor-Corrector, Collocation κ.α ενώ για την επίλυση Μερικών Διαφορικών Εξισώσεων (ΜΔΕ), οι συναρτήσεις των οποίων περιέχουν περισσότερες από μια ανεξάρτητη μεταβλητή, είναι οι: Μέθοδος των Πεπερασμένων Στοιχείων, Πεπερασμένων Διαφορών, κ.α. Στις προαναφερθείσες μεθόδους προστίθενται τα Τεχνητά Νευρωνικά Δίκτυα (ΤΝΔ), τα οποία μπορούν να χρησιμοποιηθούν τόσο για την επίλυση Συνήθων, όσο και Μερικών Διαφορικών Εξισώσεων [1].

Τα ΤΝΔ είναι μαθηματικά μοντέλα που προσομοιώνουν τη λειτουργία των βιολογικών νευρώνων και αποτελούν ένα δίκτυο συνδεδεμένων υπολογιστικών στοιχείων, των νευρώνων. Τα ΤΝΔ μέσω της διαδικασίας της εκπαίδευσης μαθαίνουν τον τρόπο με τον οποίο συνδέονται τα δεδομένα που τους παρέχονται και στη συνέχεια μπορούν να χρησιμοποιηθούν για να επιλύσουν το συγκεκριμένο πρόβλημα με τη χρήση νέων δεδομένων. Χρησιμοποιούνται με επιτυχία για αναγνώριση προτύπων, βελτιστοποίηση, προβλέψεις κ.α. [1].

Σύμφωνα με τους Hayati και Karami [2], οι κοινές μέθοδοι εύρεσης των προσεγγιστικών λύσεων των Διαφορικών Εξισώσεων, όπως η Runge-Kutta, υπολογίζουν επαρκώς τη λύση, όμως καταναλώνουν αρκετό υπολογιστικό χρόνο και μνήμη. Σε αντίθεση τα ΤΝΔ, που σύμφωνα με τους Λάγαρη κ.α. [3] απαιτούν μικρή μνήμη, λόγω του μικρού αριθμού των παραμέτρων μοντελοποίησης που χρησιμοποιούνται. Επίσης τα ΤΝΔ παρέχουν λύσεις με πολύ καλές ικανότητες γενίκευσης, οι οποίες είναι διαφορίσιμες, ενώ οι περισσότερες άλλες μέθοδοι προσφέρουν διακριτές λύσεις, ή λύσεις περιορισμένης διαφορισιμότητας.

Η εύρεση αποδοτικών μεθόδων προσέγγισης της αναλυτικής λύσης κρίνεται μέγιστης σημασίας, για την παραγωγή όσο το δυνατόν λύσεων με μικρότερο σφάλμα, μικρότερο υπολογιστικό κόστος και μεγαλύτερες ικανότητες γενίκευσης.

Για αυτό το λόγο, η συγκεκριμένη διπλωματική εργασία αποσκοπεί στη βελτιστοποίηση των τεχνικών εκπαίδευσης των ΤΝΔ για την επίλυση Συνήθων Διαφορικών Εξισώσεων και κυρίως ενός είδους αυτών που ονομάζονται «άκαμπτα».

Ως «άκαμπτες» (stiff) Συνήθειες Διαφορικές Εξισώσεις ορίζονται προβλήματα τα οποία, όταν επιλύονται με μεταβλητού βήματος μεθόδους, οι λύσεις αποτυγχάνουν, γιατί η μέθοδος αναγκάζεται να μικραίνει συνεχώς το βήμα χωρίς να μπορεί να επιτύχει την επιθυμητή ακρίβεια, ενώ όταν επιλύονται με σταθερού βήματος μεθόδους, πάλι αποτυγχάνουν για τιμή του βήματος η οποία είναι συνήθως αρκετά μικρή [10].

Για την επίλυση Διαφορικών Εξισώσεων με τη χρήση ΤΝΔ έχουν γίνει αρκετές προσπάθειες, οι οποίες αρχικά περιορίστηκαν κυρίως στην περίπτωση της επίλυσης συστημάτων αλγεβραϊκών εξισώσεων που προκύπτουν από τη διακριτοποίηση του χώρου των λύσεων. Στη συγκεκριμένη περίπτωση η λύση βρίσκεται με την ελαχιστοποίηση της συνάρτησης ενέργειας του ΤΝΔ. Μία άλλη μέθοδος για την επίλυση Διαφορικών Εξισώσεων με ΤΝΔ βασίζεται στη χρήση  $B_1$  splines ως συναρτήσεις βάσεως. Η λύση σε αυτή τη προσέγγιση μπορεί να βρεθεί με την επίλυση ενός συστήματος γραμμικών ή μη γραμμικών εξισώσεων για τον καθορισμό των συντελεστών των splines και στη συνέχεια αυτή η λύση εισέρχεται σε ένα ΤΝΔ εμπρόσθιας τροφοδότησης, στο οποίο η κάθε spline αντικαθίσταται από ένα γραμμικό άθροισμα συναρτήσεων ενεργοποίησης σε κάθε νευρώνα. Η συγκεκριμένη μέθοδος ωστόσο, παρουσιάζει κάποια μειονεκτήματα, όπως το ότι δεν είναι εύκολο να χρησιμοποιηθεί σε πολυδιάστατους χώρους(domains) [1].

Η μέθοδος που πρότειναν οι Λαγαρής κ.α [3] βασίζεται στην ικανότητα των ΤΝΔ εμπρόσθιας τροφοδότησης να προσεγγίζουν συναρτήσεις και την παράγωγό τους και μπορεί να χρησιμοποιηθεί για την επίλυση τόσο Συνήθων όσο και Μερικών Διαφορικών Εξισώσεων με την κατασκευή κατάλληλης μορφής της δοκιμαστικής λύσης. Η συγκεκριμένη μέθοδος έχει πολύ καλές ικανότητες γενίκευσης και μπορεί να χρησιμοποιηθεί για domains υψηλότερων διαστάσεων. Μια ακόμη μέθοδος η οποία βασίζεται στη χρήση MLP παρουσιάζεται από τους He κ.α [13]. Στην μέθοδο αυτή χρησιμοποιείται ο extended back-propagation αλγόριθμος για την εκπαίδευση της παραγωγού ενός ΤΝΔ εμπρόσθιας τροφοδότησης και μέσω αυτής επιλύεται μία κλάση από πρώτης τάξης Μερικές Διαφορικές Εξισώσεις. Το μειονέκτημα της συγκεκριμένης μεθόδου είναι ότι δεν μπορεί να χρησιμοποιηθεί σε μεγάλο διάστημα τρεξίματος. Επιπλέον, οι Mall και Chakraverty [16] πρότειναν έναν νέο αλγόριθμο για την επίλυση singular Προβλημάτων Αρχικών Τιμών (Π.Α.Τ.) και εξισώσεων τύπου Emden–Fowler με τη χρήση Chebyshev ΤΝΔ μονής στιβάδας. Τα προβλήματα τα οποία επίλυσαν είναι 2<sup>ns</sup> τάξης Μερικές Διαφορικές Εξισώσεις και έχουν πολλές εφαρμογές στην αστροφυσική και τη κβαντομηχανική. Το μοντέλο το οποίο ανέπτυξαν και βασίζεται στη χρήση ενός ΤΝΔ εμπρόσθιας τροφοδότησης με τη χρήση του error-back-propagation αλγορίθμου για την ελαχιστοποίηση της συνάρτησης κόστους, έχει μικρότερο υπολογιστικό κόστος από τα MLP ΤΝΔ, ενώ είναι εύκολο στην εφαρμογή του.

Όσον αφορά την επίλυση άκαμπτων Διαφορικών Εξισώσεων με τη χρήση ΤΝΔ έχουν γίνει λίγες προσπάθειες μέχρι στιγμής, όπως αυτή των Lazovskaya και

Tarkhon, οι οποίοι επίλυσαν μια Συνήθης Διαφορική Εξίσωση, η οποία περιέχει παράμετρο [15]. Ωστόσο, στη συγκεκριμένη δημοσίευση το πρόβλημα το οποίο παρουσιάζεται δεν παρουσιάζει σημαντική ακαμψία. Τέλος πολύ σημαντικό είναι το βιβλίο των Yadan κ.α. το οποίο συνοψίζει πολλές από τις μεθόδους επίλυσης Διαφορικών Εξισώσεων με τη χρήση ΤΝΔ [1].

Στην παρούσα εργασία η λύση της εκάστοτε Διαφορικής Εξίσωσης που παράγεται με τη χρήση ΤΝΔ, βασίζεται στη δημοσίευση των Λαγαρή κ.α [3] και η λύση που προκύπτει συγκρίνεται με την αναλυτική λύση της Διαφορικής Εξίσωσης. Επιπλέον γίνεται μια προσπάθεια για την εκπαίδευση ΤΝΔ που επιλύουν Διαφορικές Εξισώσεις που περιέχουν παραμέτρους είτε σταθερές είτε διαστήματα αυτών. Οι Διαφορικές Εξισώσεις οι οποίες επιλύονται ανήκουν στην κατηγορία των άκαμπτων (stiff) προβλημάτων.

Στο Κεφάλαιο 1 αναφέρονται εισαγωγικές πληροφορίες για τα ΤΝΔ, τις Διαφορικές Εξισώσεις και τα άκαμπτα προβλήματα και περιγράφεται αναλυτικά η μέθοδος των Λαγαρή κ.α. . Στο Κεφάλαιο 2 περιγράφεται η μεθοδολογία που ακολουθήθηκε στην εργασία για την επίλυση των Διαφορικών Εξισώσεων με τη χρήση ΤΝΔ. Στο Κεφάλαιο 3 παρουσιάζονται οι τροποποιήσεις που έγιναν στη μέθοδο των Λαγαρή κ.α και μελετώνται, η διερεύνηση που γίνεται για την επίλυση Διαφορικών Εξισώσεων που περιέχουν παράμετρο και η περιγραφή του κώδικα για την επίλυση ΣΔΕ με τη χρήση ΤΝΔ. Στο Κεφάλαιο 4 φαίνονται τα προβλήματα τα οποία επιλύονται για τη διερεύνηση των προτεινόμενων μεθόδων καθώς και τα αποτελέσματά τους, ενώ στο Κεφάλαιο 5 αναλύονται τα αποτελέσματα που προκύπτουν για το κάθε πρόβλημα. Επιπλέον στο Κεφάλαιο 6 αναφέρονται τα συμπεράσματα της παρούσας έρευνας και μελλοντικές προτάσεις έρευνας. Τέλος στο Παράρτημα Α παρουσιάζονται οι πίνακες επίλυσης με μεθόδους εκτός των ΤΝΔ για κάθε πρόβλημα, στο Παράρτημα Β οι τιμές σφάλματος που προκύπτουν από τα ΤΝΔ για κάθε πρόβλημα και στο Παράρτημα Γ ενδεικτικοί κώδικες που χρησιμοποιήθηκαν για την εξαγωγή αποτελεσμάτων.

# ΚΕΦΑΛΑΙΟ 1: Θεωρητικό πλαίσιο του θέματος – Ανασκόπηση του πεδίου

---

## **1.1.1 Τεχνητή Νοημοσύνη**

Ο σχεδιασμός αλγοριθμικών μοντέλων για την επίλυση ολοένα πιο περίπλοκων προβλημάτων έχει πλέον σηματοδοτηθεί από τη χρήση «ευφών αλγορίθμων» όπως τα ΤΝΔ (Artificial Neural Networks), ο Εξελικτικός υπολογισμός (Evolutionary Computation), η Νοημοσύνη Σμήνους (Swarm Intelligence), το Τεχνητό Ανοσοποιητικό Σύστημα (Artificial Immune System) και τα Ασαφή Συστήματα (Fuzzy Systems). Οι προαναφερθείσες μέθοδοι ανήκουν στον τομέα της Τεχνητής Νοημοσύνης και ειδικότερα σε μία υποκατηγορία αυτής που ονομάζεται «Υπολογιστική Νοημοσύνη».

Η «Υπολογιστική Νοημοσύνη» ασχολείται με τη μελέτη προσαρμοστικών μηχανισμών που επιτρέπουν ή διευκολύνουν την «έξυπνη» συμπεριφορά σε σύνθετα και μεταβαλλόμενα περιβάλλοντα. Οι μηχανισμοί αυτοί έχουν την ικανότητα να μαθαίνουν από νέες καταστάσεις και να προσαρμόζονται σε αυτές, να γενικεύουν, να ανακαλύπτουν και να συνδέονται. Επιπλέον ο κάθε ένας τομέας έχει τις ρίζες του σε βιολογικά και φυσικά συστήματα [4]. Γενικότερα, η υπολογιστική νοημοσύνη βοηθάει στην επίλυση πολύπλοκων προβλημάτων στα οποία η μαθηματική ή η κλασσική μοντελοποίηση δεν είναι επαρκής είτε λόγω της ισχυρής πολυπλοκότητας που δεν μπορεί να αποτυπωθεί με μαθηματικούς συλλογισμούς είτε λόγω της ύπαρξης αβεβαιοτήτων κατά τη διάρκεια της διαδικασίας, είτε εξαιτίας της στοχαστικής φύσης της διαδικασίας [5].

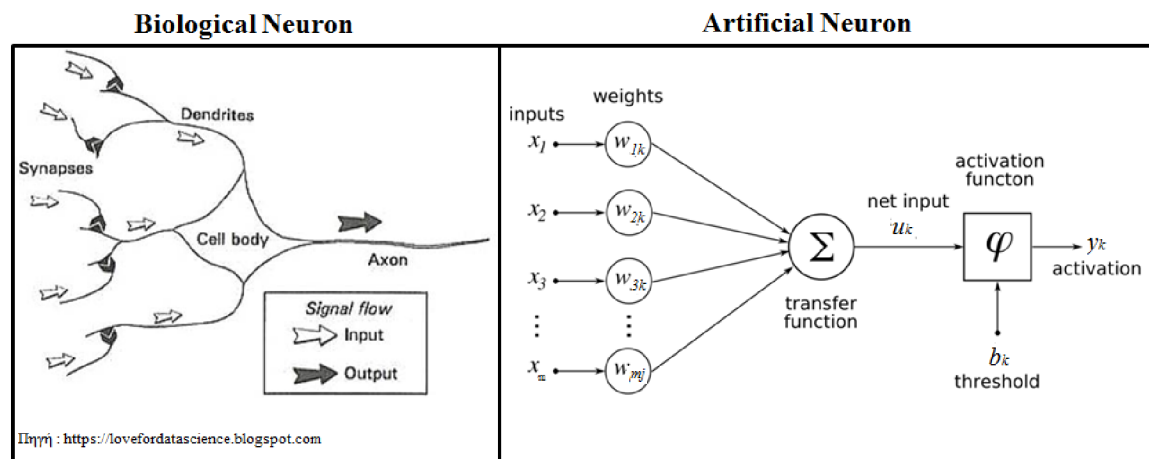
## **1.1.2 Τεχνητά Νευρωνικά Δίκτυα**

Η μελέτη των ΤΝΔ στηρίζεται στο γεγονός ότι ο ανθρώπινος εγκέφαλος πραγματοποιεί τους υπολογισμούς με εντελώς διαφορετικό τρόπο από τον ψηφιακό υπολογιστή. Η βασική διαφορά τους βρίσκεται στο ότι ο ψηφιακός υπολογιστής δουλεύει σειριακά, σε αντίθεση με τον ανθρώπινο εγκέφαλο, ο οποίος είναι ένας τρομερά πολύπλοκος, μη γραμμικός, παράλληλα συνδεδεμένος υπολογιστής, ο οποίος έχει την ικανότητα να οργανώνει τις δομικές του μονάδες, δηλαδή τους νευρώνες, για την επίτευξη ορισμένων υπολογισμών [6].

Επομένως τα ΤΝΔ αποτελούν ένα μαθηματικό μοντέλο προσομοίωσης της λειτουργίας του ανθρώπινου εγκεφάλου για να εκτελέσει μια συγκεκριμένη διεργασία. Δύο βασικές κατηγορίες στις οποίες χρησιμοποιούνται είναι η προσέγγιση συνάρτησης, στην οποία σκοπός είναι η εκμάθηση των λειτουργικών σχέσεων μεταξύ των εισερχομένων και των επιθυμητών διανυσμάτων εξόδου και

βρίσκει συχνά εφαρμογές σε προβλέψεις και η αναγνώριση προτύπων, στην οποία το ΤΝΔ προσπαθεί να ανιχνεύσει ένα μοτίβο που συνδέεται καλύτερα με ένα δεδομένο διάνυσμα εισόδου και χρησιμοποιείται σε εφαρμογές όπως η υπολογιστική όραση και η αναγνώριση φωνής [4].

Τα ΤΝΔ αποτελούνται από νευρώνες οι οποίοι προσομοιάζουν τους αντίστοιχους βιολογικούς και παρουσιάζουν πολλές αναλογίες, όπως διακρίνεται στην Εικόνα 1.1.



Εικόνα 1.1 : Σύγκριση βιολογικού & τεχνητού νευρώνα

Ο βιολογικός νευρώνας αποτελείται από το κυτταρικό σώμα, το οποίο δέχεται πληροφορίες από το περιβάλλον του, μέσω των δενδριτών και των συνάψεων που δημιουργεί με άλλους νευρώνες και στη συνέχεια παράγει το σήμα εξόδου. Ο τεχνητός νευρώνας αποτελείται από τον κόμβο που περιλαμβάνει το σταθμικό άθροισμα και τη συνάρτηση ενεργοποίησης. Ο κόμβος αυτός παραλαμβάνει την πληροφορία που δέχεται ο νευρώνας από τα σήματα εισόδου, τα οποία ενισχύονται ή αποδυναμώνονται ανάλογα με τις τιμές των συναπτικών βαρών και στο τέλος παράγει το δικό του σήμα εξόδου.

Κάθε τεχνητός νευρώνας δέχεται πληροφορία από το περιβάλλον του ( $x_j$ ). Στη συνέχεια το σήμα αυτό ενισχύεται ή αποδυναμώνεται ανάλογα με την τιμή του *συναπτικού βάρους* ( $w_{kj}$ ) με το οποίο πολλαπλασιάζεται και έπειτα υπολογίζεται το άθροισμα όλων των εισερχόμενων σημάτων, μέσω ενός γραμμικού αθροιστή, όπως φαίνεται στη σχέση (1.1). Τέλος το άθροισμα αυτό εισέρχεται σε μια συνάρτηση ενεργοποίησης, η οποία καθορίζει το εύρος του εξαγόμενου σήματος του νευρώνα σε μία διακριτή τιμή.

$$u_n = \sum_{n=1}^m w_{n,j} x_n \quad (1.1)$$

Επομένως, η ισχύς του εξαγόμενου σήματος ενός τεχνητού νευρώνα ( $y_k$ ), ο τύπος του οποίου περιγράφεται στην (1.2) καθορίζεται από:

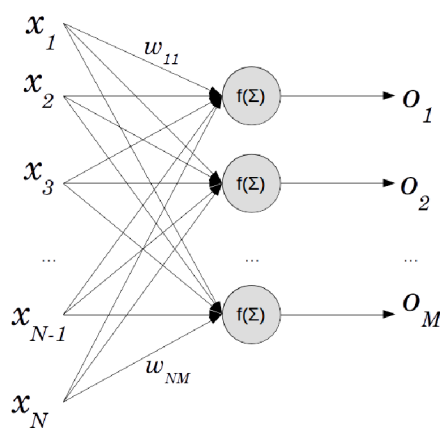
- a) την τιμή εισόδου
- b) το συναπτικό βάρος με το οποίο πολλαπλασιάζεται η είσοδος
- c) τη συνάρτηση ενεργοποίησης που θα χρησιμοποιηθεί
- d) από μία επιπλέον τιμή η οποία ονομάζεται πόλωση ( $b_k$ ) [4,6] .

$$y_k = \varphi(u_k + b_k) \quad (1.2)$$

Όσον αφορά τη σύγκριση του βιολογικού και του τεχνητού νευρώνα, το κυτταρικό σώμα του βιολογικού νευρώνα είναι ανάλογο του κόμβου του τεχνητού, ενώ τα συναπτικά βάρη του τεχνητού είναι ανάλογα των συνάψεων του βιολογικού. Τέλος ένα ΤΝΔ αποτελείται από διασυνδεδεμένους τεχνητούς νευρώνες, όπως θα αναλυθεί παρακάτω.

### 1.1.3 Αρχιτεκτονική & τύποι Τεχνητών Νευρωνικών Δικτύων

Η Εικόνα 1.2 παρουσιάζει ένα ΤΝΔ μονής στιβάδας, το οποίο αποτελείται από συνδεδεμένους τεχνητούς νευρώνες. Κάθε μία παράλληλη διάταξη νευρώνων αποτελεί μία στιβάδα (“layer”). Υπάρχουν νευρωνικά τα οποία δεν αποτελούνται από κρυφές στιβάδες και ονομάζονται “single layer networks” και υπάρχουν και νευρωνικά τα οποία αποτελούνται από μία ή περισσότερες κρυφές στιβάδες, τα “multilayer networks”.

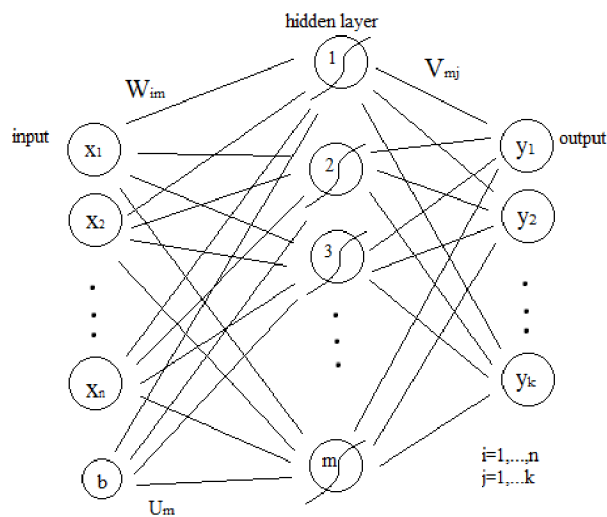


Πηγή: Neural-networks.io

Εικόνα 1.2: ΤΝΔ μονής στιβάδας

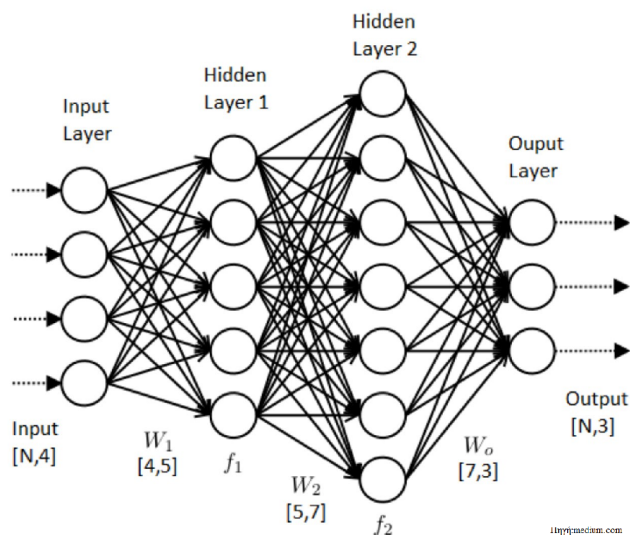
Στην Εικόνα 1.3 παρουσιάζεται ένα ΤΝΔ με μία κρυφή στιβάδα και τα συναπτικά βάρη της στιβάδας εισόδου με την κρυφή στιβάδα ( $W$ ) και τα συναπτικά βάρη της κρυφής στιβάδας με τη στιβάδα εξόδου ( $V$ ), όπως και η πόλωση ( $b$ ) και τα συναπτικά βάρη αυτής ( $U$ ). Τα  $W, V, U$  αποτελούν τις παραμέτρους του ΤΝΔ.





Εικόνα 1.3 : ΤΝΔ μίας κρυφής στιβάδας

Στην Εικόνα 1.4 παρουσιάζεται ένα multilayer ΤΝΔ 2 κρυφών στιβάδων.



Εικόνα 1.4 : ΤΝΔ δύο κρυφών στιβάδων

Γενικότερα ο αριθμός και η δομή των νευρώνων που απαρτίζουν το ΤΝΔ αποτελούν την αρχιτεκτονική του. Η αρχιτεκτονική συνδέεται άμεσα με τον κανόνα εκπαίδευσης που χρησιμοποιείται. Εκπαίδευση είναι ο καθορισμός τω συναπτικών βαρών και της πόλωσης, έτσι ώστε να πραγματοποιείται μία λειτουργία. Υπάρχει η μάθηση με επίβλεψη, χωρίς επίβλεψη, η ανταγωνιστική μάθηση και η μάθηση με ενίσχυση.

Όσον αφορά τη μάθηση με επίβλεψη, απαιτείται ένα σετ εκπαίδευσης το οποίο αποτελείται από διανύσματα εισόδου και ένα διάνυσμα-στόχο το οποίο συνδέεται με κάθε διάνυσμα εισόδου. Το διάνυσμα-στόχος χρησιμοποιείται ως κριτήριο για το πόσο καλά έχει εκπαιδευτεί το ΤΝΔ και για να καθοριστούν οι ρυθμίσεις στα βάρη, έτσι ώστε να μειώνεται το σφάλμα.

### 1.1.4 Συναρτήσεις Ενεργοποίησης

Ανάλογα με την επιλογή της συναρτήσεως ενεργοποίησης που θα χρησιμοποιηθεί γίνεται μια αντιστοίχιση των δεδομένων, οι τιμές των οποίων ανήκουν στο χώρο των πραγματικών αριθμών, σε ένα διάστημα αυτού το οποίο συνήθως είναι το (-1,1) ή το (0,1).

Μερικές από τις συναρτήσεις ενεργοποίησης που χρησιμοποιούνται παρουσιάζονται παρακάτω:

- ⌚ *Σιγμοειδής (Sigmoid)* : Η σιγμοειδής συνάρτηση, που διακρίνεται στη σχέση 1.3, είναι μία από τις πιο συχνά χρησιμοποιούμενες συναρτήσεις ενεργοποίησης. Η παράμετρος  $\lambda$  ελέγχει την κλίση της συνάρτησης, η οποία τις περισσότερες φορές λαμβάνει την τιμή  $\lambda=1$ . Η έξοδος που παράγεται ανήκει στο εύρος τιμών (0,1).

$$\varphi(u_k + b_k) = \frac{1}{1 + e^{-\lambda(u_k + b_k)}} \quad (1.3)$$

- ⌚ *Υπερβολική εφαπτομένη ( Hyperbolic tangent)*: Η Υπερβολική εφαπτομένη, που φαίνεται στη σχέση (1.4) δημιουργεί έξοδο που ανήκει στο εύρος τιμών (-1,1).

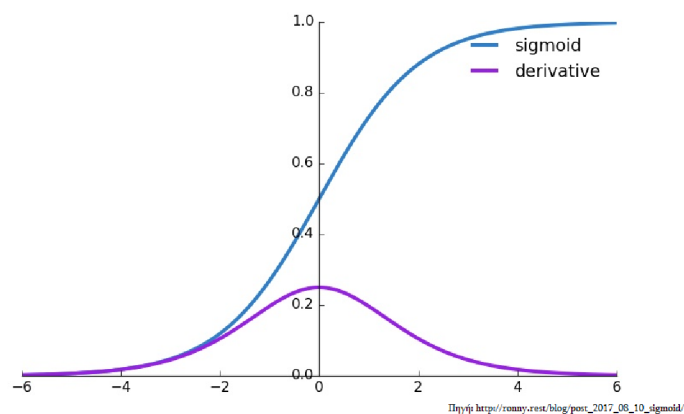
$$\varphi(u_k + b_k) = \frac{e^{\lambda(u_k + b_k)} - e^{-\lambda(u_k + b_k)}}{e^{\lambda(u_k + b_k)} + e^{-\lambda(u_k + b_k)}} = \tanh \quad (1.4)$$

Υπάρχει μία γενίκευση της υπερβολικής εφαπτομένης, η οποία φαίνεται στη σχέση (1.5) και η οποία ως συμμετρική σιγμοειδής έχει πιο γρήγορη σύγκλιση από την απλή σιγμοειδή [18].

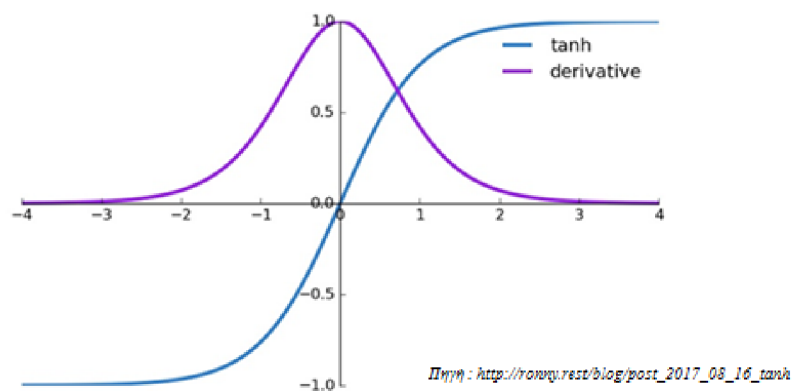
$$\varphi(u_k + b_k) = 1.7159 \tanh\left(\frac{2}{3}x\right) \quad (1.5)$$

Ένα βασικό πρόβλημα των tanh και sigmoid είναι το “vanishing gradients problem”, το οποίο προκύπτει στα ΤΝΔ που αποτελούνται από πολλές κρυφές στιβάδες λόγω της μορφής της παραγώγου αυτών των δύο συναρτήσεων ενεργοποίησης. Όπως διακρίνεται στην Εικόνα 1.5 για τη σιγμοειδή και στην 1.6 για την υπερβολική

εφαπτομένη όσο πιο μεγάλες είναι οι τιμές των δεδομένων τόσο πιο πολύ τείνουν στο 0 οι τιμές των παραγώγων τους, κάτι το οποίο δημιουργεί πρόβλημα στη χρήση της διαδικασίας ελαχιστοποίησης που χρησιμοποιείται στην εκπαίδευση του ΤΝΔ.



Εικόνα 1.5 : Σιγμοειδής συνάρτηση & η παράγωγός της



Εικόνα 1.6 : Υπερβολική εφαπτομένη & η παράγωγός της

Για να ξεπεραστεί το “vanishing gradients problem” στα ΤΝΔ πολλών στιβάδων, χρησιμοποιούνται κυρίως δύο άλλες συναρτήσεις ενεργοποίησης: η Softplus και η ReLU. Η ReLU είναι γραμμική οπότε οι υπολογισμοί στη διαδικασία εκπαίδευσης είναι πολύ

πιο απλοί οπότε και πιο γρήγοροι. Ωστόσο, η Softplus είναι πιο ομαλή, κάτι το οποίο κάνει τη συγκεκριμένη συνάρτηση ενεργοποίησης πιο σταθερή, ενώ η ReLU έχει μια ασυνέχεια στην κλίση στο σημείο 0. Επίσης η Softplus έχει μη μηδενική τιμή της κλίσης όταν η είσοδος είναι αρνητική, σε αντίθεση με τη ReLU [7].

🕒 **Softplus:** Η softplus δίνει εύρος τιμών  $(0, \infty)$  και ο τύπος της δίνεται από τον τύπο (1.6).

$$\varphi(u_k + b_k) = \ln(1 + e^{(u_k + b_k)}) \quad (1.6)$$

- ⌚ *Rectified Linear Units (ReLU)*: Όπως φαίνεται από την (1.7) η ReLU, είναι μία γραμμική συνάρτηση για τιμές μεγαλύτερες του 0, ωστόσο είναι κατά κλάδους γραμμική και συνεχής. Λόγω αυτής της ιδιαιτερότητας, η ReLU διατηρεί πολλές από τις ιδιότητες που κάνουν τα γραμμικά μοντέλα να βελτιστοποιούνται εύκολα με “gradient-based” μεθόδους, όπως και το να γενικεύουν καλά.

$$\varphi(u_k + b_k) = \max(0, (u_k + b_k)) \quad (1.7)$$

### 1.1.5 Τεχνητά Νευρωνικά Δίκτυα Εμπρόσθιας τροφοδότησης

Υπάρχουν διάφοροι τύποι νευρωνικών δικτύων τα οποία μαθαίνουν με επίβλεψη, όπως τα νευρωνικά πολλών στιβάδων (MLP), τα “recurrent”, τα “time-delays”, κ.α. Ένας τύπος των MLP είναι τα feedforward νευρωνικά δίκτυα, τα οποία λαμβάνουν τα εξωτερικά σήματα και τα προωθούν σε όλες τις στιβάδες για να λάβουν τα αποτελέσματα, χωρίς να υπάρχουν ανατροφοδοτήσεις με προηγούμενες στιβάδες.

Μια πολύ σημαντική ιδιότητα των feedforward νευρωνικών είναι ότι αυτά που έχουν μονοτονικά αυξανόμενες διαφοροποιήσιμες συναρτήσεις ενεργοποίησης μπορούν να προσεγγίσουν κάθε συνεχή συνάρτηση και την παράγωγό της, με την ύπαρξη ακόμα και μόνο μίας κρυφής στιβάδας, με την προϋπόθεση ότι η κρυφή στιβάδα θα αποτελείται από αρκετούς νευρώνες [4].

Η σχέση (1.8) περιγράφει το σήμα εξόδου ενός νευρώνα της εξωτερικής στιβάδας, όπου ως  $o_k$  είναι η μονάδα εξόδου και  $y_j$  η κρυφή μονάδα και  $z_{i,p}$  η τιμή της μονάδας εισόδου  $z_i$  του μοτίβου εισόδου  $z_p$ .

$$o_{k,p} = \varphi_{ok}(net_{ok,p}) = \varphi_{ok}\left(\sum_{j=1}^{J+1} w_{kj} f_{yj}(net_{yj,p})\right) = \varphi_{ok}\left(\sum_{j=1}^{J+1} w_{kj} f_{yj}\left(\sum_{i=1}^{I+1} u_{ji} z_{i,p}\right)\right) \quad (1.8)$$

Σημαντικό είναι επίσης το ότι η συνάρτηση ενεργοποίησης δεν είναι απαραίτητο να είναι η ίδια σε κάθε νευρώνα [4].

### 1.1.6 Κυριότεροι Αλγόριθμοι Εκπαίδευσης.

Η ικανότητα εκμάθησης από το περιβάλλον και η βελτίωση της απόδοσης του ΤΝΔ, ονομάζεται εκπαίδευση του δικτύου και αποσκοπεί στον καθορισμό των συναπτικών βαρών και της πόλωσης έτσι ώστε να ελαχιστοποιείται το σφάλμα μεταξύ πραγματικής και επιθυμητής τιμής, όταν προσεγγίζουμε τιμές συνάρτησης.

Η εκπαίδευση είναι μια επαναληπτική διαδικασία, η οποία σταματάει όταν έχει περάσει ένας προκαθορισμένος αριθμός επαναλήψεων, είτε όταν το ελάχιστο τετραγωνικό σφάλμα στο σετ της εκπαίδευσης είναι αρκετά μικρό.

Η πιο συχνά χρησιμοποιούμενη μέθοδος βελτιστοποίησης που χρησιμοποιείται για την εύρεση του ελάχιστου σφάλματος είναι η “gradient descent” και αποτελεί και τη βάση του backpropagation algorithm.

Στα feedforward δίκτυα το άθροισμα του τετραγώνου του σφάλματος, που φαίνεται στη σχέση (1.9) , είναι η αντικειμενική συνάρτηση, δηλαδή η συνάρτηση της οποίας αναζητούμε το ελάχιστο.

$$\varepsilon_p = \frac{1}{2} \left( \frac{\sum_{k=1}^K (t_{k,p} - o_{k,p})^2}{K} \right) \quad (1.9)$$

Στη σχέση (1.9) ως K συμβολίζεται το πλήθος των μονάδων εξόδου, ως t η επιθυμητή τιμή εξόδου της k μονάδας και ως o η πραγματική τιμή εξόδου της k μονάδας εξόδου.

#### 1.1.6.1 Error Back-propagation algorithm:

Ο πιο διαδεδομένος αλγόριθμος, ο οποίος χρησιμοποιείται στη μάθηση με επίβλεψη των multilayer networks είναι ο error back-propagation algorithm. Ο συγκεκριμένος αλγόριθμος αποτελείται από 2 περάσματα του σήματος μέσω των διαφορετικών στιβάδων του δικτύου, το forward pass και το backward pass.

Στο forward pass τα δεδομένα εισόδου και το αποτέλεσμα τους διαδίδονται από στιβάδα σε στιβάδα. Τελικά προκύπτει ένα σετ δεδομένων εξόδου, που είναι η απόκριση του συστήματος. Κατά τη διάρκεια του, τα συναπτικά βάρη του δικτύου είναι όλα προσδιορισμένα και παραμένουν αναλλοίωτα σε όλο το δίκτυο, ενώ τα σήματα λειτουργίας του δικτύου υπολογίζονται από νευρώνα σε νευρώνα [4].

Στο backward pass τα συναπτικά βάρη είναι όλα προσαρμοσμένα με βάση ένα error-correction learning rule. Η πραγματική απόκριση του δικτύου προκύπτει από μία επιθυμητή απόκριση παραγωγής ενός σήματος σφάλματος, το οποίο στη συνέχεια διαδίδεται αντίθετα από την κατεύθυνση των συναπτικών συνδέσεων.

Τα συναπτικά βάρη αρχικά είναι τυχαία αλλά στη συνέχεια προσαρμόζονται για να επιτρέψουν στην πραγματική απόκριση του δικτύου να προσεγγίσει όσο το δυνατόν καλύτερα την επιθυμητή.

Ο back-propagation algorithm αποτελεί μια μέθοδο βελτιστοποίησης, η οποία λειτουργεί με τη μέθοδο της ταχύτερης καθόδου, δηλαδή σε κάθε βήμα αναζητείται η ελάχιστη τιμή της κλίσης της συνάρτησης. Ο στόχος του αλγορίθμου είναι να μάθει τις συσχετίσεις μεταξύ των σετ των δεδομένων εισόδου-εξόδου  $\{(p_1, t_1), (p_2, t_2), \dots, (p_Q, t_Q)\}$  και ως πρώτο βήμα υπολογίζει τα σήματα εξόδου των νευρώνων με βάση τις σχέσεις (1.10) –(1.11).

$$a^0 = p \quad (1.10)$$

$$a^{k+1} = \varphi^{k+1}(W^{k+1}a^k + b^{k+1}) \quad (1.11)$$

Στη συνέχεια κατά το backward pass μεταδίδει προς τα πίσω την πληροφορία που υπάρχει στο σφάλμα πρόβλεψης ( $\delta$ ), που φαίνεται στον τύπο (1.12).

$$\delta^k = F^k(\eta^k)W^{(k+1)T} \delta^{k+1} \quad (1.12)$$

Τέλος γίνεται η αναβάθμιση των συναπτικών βαρών και των bias με βάση τις σχέσεις (1.13) και (1.14) αντίστοιχα [6].

$$\Delta w_{kj} = n \left( -\frac{\partial E}{\partial w_{kj}} \right) = -n \frac{\partial E}{\partial o_k} \frac{\partial o_k}{\partial w_{kj}} = -n \delta_{ok} y_j \quad (1.13)$$

$$\Delta u_{ji} = n \left( -\frac{\partial E}{\partial u_{ji}} \right) = -n \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial u_{ji}} = -n \delta_{y_j} z_i \quad (1.14)$$

#### 1.1.6.2 Levenberg-Marquardt algorithm:

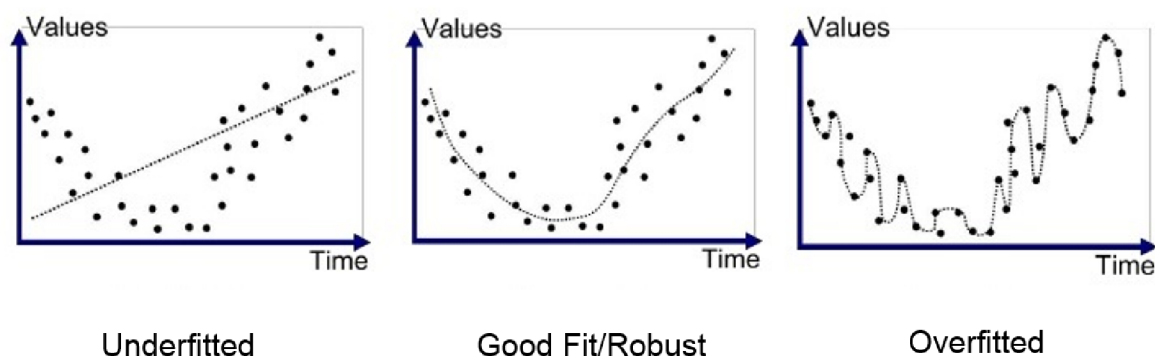
Ο λόγος για τον οποίο δημιουργήθηκε ο συγκεκριμένος αλγόριθμος ήταν για την επιτάχυνση της σύγκλισης του back-propagation αλγορίθμου. Ενώ ο back-propagation algorithm είναι αλγόριθμος της «πιο απότομης καθόδου», ο Marquardt-Levenberg είναι ένας συνδυασμός της «απότομης καθόδου» και της «μεθόδου Newton». Η μέθοδος της «απότομης καθόδου» συγκλίνει συνήθως αργά, αλλά εγγυημένα, σε αντίθεση με τη “Newton” η οποία συγκλίνει πολύ πιο γρήγορα, αλλά υπάρχει το ενδεχόμενο εάν δεν επιλεγεί η σωστή αρχική τιμή να μη συγκλίνει. Επιπλέον, παρόλο που οι Newton μέθοδοι θεωρούνται πιο αποδοτικές, οι υπολογιστικές και αποθηκευτικές τους απαιτήσεις αυξάνονται με την αύξηση του μεγέθους του τεχνητού νευρωνικού δικτύου, κάτι το οποίο ενισχύει την αποδοτικότητα της Marquardt-Levenberg σε σχέση με την απλή μέθοδο της απότομης καθόδου και της Newton.

Στη συγκεκριμένη μέθοδο υπολογίζονται τα δεδομένα εξόδου από τις σχέσεις (1.10) και (1.11) και τα σφάλματα της αντικειμενικής συνάρτησης σε όλες τις εισόδους. Στη συνέχεια υπολογίζεται ένας Ιακωβιανός πίνακας, ο οποίος περιγράφεται από τη σχέση (1.15).

$$J(x) = \left[ \frac{\partial e_N(x)}{\partial x_n} \right] \quad (1.15)$$

### 1.1.7 Overfitting

Μια σημαντική κατάσταση η οποία μπορεί να εμφανιστεί στα ΤΝΔ είναι το “overfitting”. Το “overfitting” ενός σετ εκπαίδευσης σημαίνει ότι το ΤΝΔ απομνημονεύει τα πρότυπα εκπαίδευσης, με αποτέλεσμα να χάνει συνεχώς την ικανότητα του να γενικεύει καλά. Επομένως, τα ΤΝΔ που προκύπτουν δεν μπορούν να προβλέψουν επαρκώς πρότυπα δεδομένων τα οποία δεν έχουν συναντήσει στην εκπαίδευση. Ένα τυπικό παράδειγμα “overfitting” φαίνεται στην Εικόνα 1.7.



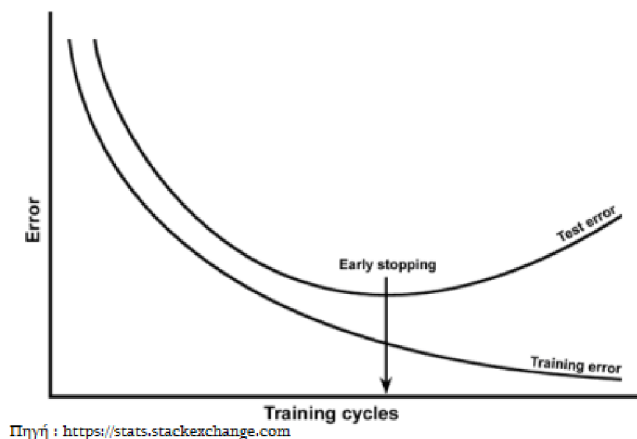
Πηγή : <https://medium.com>

Εικόνα 1.7: Παράδειγμα “overfitting”

Το “overfitting” συμβαίνει όταν η αρχιτεκτονική των ΤΝΔ είναι μεγάλη, δηλαδή όταν υπάρχουν πολλές ελεύθεροι παράμετροι (συναπτικά βάρη). Επίσης αν το ΤΝΔ εκπαιδεύεται για μεγάλο χρονικό διάστημα, οι περισσευούμενες ελεύθερες παράμετροι αρχίζουν να απομνημονεύουν όλα τα πρότυπα εκπαίδευσης, ακόμα και το θόρυβο που υπάρχει στο σετ εκπαίδευσης.

Για την αποφυγή του “overfitting” και επομένως για την αύξηση της αποδοτικότητας του ΤΝΔ πρέπει να βελτιστοποιείται η αρχιτεκτονική και να χρησιμοποιούνται τα κατάλληλα πρότυπα εκπαίδευσης. Επιπλέον εκτιμήσεις του σφάλματος γενίκευσης

κατά τη διάρκεια της εκπαίδευσης μπορούν να χρησιμοποιηθούν για να βρεθεί το σημείο στο οποίο αρχίζει το “overfitting”. Η πιο απλή μέθοδος για να βρεθεί το σημείο «overfitting» παρουσιάζεται στην Εικόνα 1.8, στην οποία όπως φαίνεται από την αρχή της εκπαίδευσης τόσο τα σφάλματα της εκπαίδευσης και της γενίκευσης μειώνονται, ενώ στο σημείο που αρχίζει το “overfitting” το σφάλμα της γενίκευσης αρχίζει να αυξάνεται και σε αυτό το σημείο πρέπει να διακόπτεται η εκπαίδευση.



Εικόνα 1.8: Σημείο overfitting

Για την εύρεση του σημείου “overfitting” το αρχικό σετ δεδομένων χωρίζεται σε τρία υποσύνολα: το σετ εκπαίδευσης, το σετ γενίκευσης και το σετ αξιολόγησης. Από το σετ αξιολόγησης υπολογίζεται το σφάλμα γενίκευσης και ο κινούμενος μέσος όρος αυτού ( $E_V$ ) και η τυπική απόκλιση του σετ αυτού. Το “overfitting” εντοπίζεται όταν ισχύει η σχέση (1.16).

$$E_V > E_V + \sigma E_V \quad (1.16)$$

Τέλος παρουσιάζεται ένας ακόμη τρόπος για την εύρεση του σημείου “overfitting”, το οποίο βασίζεται στον παράγοντα γενίκευσης  $\rho = E_V / E_T$ , όπου  $E_V$  και  $E_T$  είναι το μέσο τετραγωνικό σφάλμα στο σετ αξιολόγησης  $D_V$  και στο τρέχων σετ εκπαίδευσης  $D_T$ . Ο παράγοντας γενίκευσης δείχνει το σφάλμα το οποίο συμβαίνει μόνο στην εκπαίδευση στο τρέχων σετ εκπαίδευσης και όχι όταν εκπαιδεύεται όλος ο χώρος εισόδου. Το “overfitting” εντοπίζεται όταν  $\rho(t) > \phi_\rho(t)$ , όπου  $\phi_\rho(t) = \min\{\phi_\rho(t-1), \bar{\rho} + \sigma_\rho, 1.0\}$ ,  $t$  είναι η τρέχουσα επανάληψη,  $\bar{\rho}$  είναι ο μέσος όρος του παράγοντα γενίκευσης και το  $\sigma_\rho$  είναι η τυπική απόκλιση. Για την προσέγγιση συναρτήσεων για την τιμή  $\rho < 1$  σημαίνει ότι το σφάλμα αξιολόγησης είναι μικρότερο από το σφάλμα εκπαίδευσης το οποίο είναι επιθυμητό [4].



### 1.2.1 Διαφορικές Εξισώσεις & μέθοδοι επίλυσης

Οι Διαφορικές Εξισώσεις, είναι εξισώσεις οι οποίες περιέχουν άγνωστες ποσότητες και τους ρυθμούς μεταβολής τους και χρησιμοποιούνται για τη μοντελοποίηση προβλημάτων που συναντώνται σε πολλούς τομείς, όπως της φυσικής, της μηχανικής, της βιολογίας, της οικονομίας, κ.α. Κατηγοριοποιούνται στις Συνήθεις Διαφορικές (ODEs), τις Μερικές (PDEs), τις Συναρτησιακές Διαφορικές (FDEs), τις Στοχαστικές (SDEs) και τις Αλγεβρικές Διαφορικές Εξισώσεις (DAE). Στις Συνήθεις Διαφορικές η άγνωστη συνάρτηση είναι συνάρτηση μίας ανεξάρτητης μεταβλητής, ενώ στις Μερικές είναι συνάρτηση τουλάχιστον δύο ανεξάρτητων μεταβλητών [1].

Υπάρχει ένας επιπλέον διαχωρισμός των ODE σε δύο ομάδες:

- ⌚ Προβλήματα Αρχικών τιμών (ΠΑΤ) : Υπάρχει προσδιορισμός της λύσης της Δ.Ε ή των παραγώγων αυτής σε ένα αρχικό σημείο.
- ⌚ Προβλήματα Συνοριακών τιμών (ΠΣΤ) : Υπάρχει προσδιορισμός της λύσης της Δ.Ε ή των παραγώγων αυτής στα άκρα ενός διαστήματος της λύσης.

Για τις περισσότερες Διαφορικές Εξισώσεις δεν μπορεί να υπολογισθεί αναλυτική λύση, για αυτό το λόγο επιλύονται προσεγγιστικά. Υπάρχουν μονοβηματικές μέθοδοι, πολυβηματικές και μεταβλητού βήματος, άμεσες και έμμεσες.

### 1.2.2 Άκαμπτα (stiff) προβλήματα.

Υπάρχουν κάποια προβλήματα ΣΔΕ, τα οποία όταν επιλύονται με μεταβλητού βήματος μεθόδους, οι λύσεις τους αποτυγχάνουν, γιατί η μέθοδος αναγκάζεται να μικραίνει συνεχώς το βήμα χωρίς να μπορεί να επιτύχει την επιθυμητή ακρίβεια. Επιπλέον, αποτυγχάνουν να επιλυθούν με σταθερού βήματος μεθόδους όπως η Runge-Kutta και η Euler, από μία τιμή του βήματος και πάνω, η οποία συνήθως είναι αρκετά μικρή.

Δεν υπάρχει ξεκάθαρος ορισμός της ακαμψίας. Μερικοί από τους ορισμούς που υπάρχουν στη βιβλιογραφία παρατίθενται παρακάτω:

1. Γραμμικά άκαμπτα προβλήματα χαρακτηρίζονται από πίνακες τα οποία έχουν ιδιοτιμές με αρνητικό πραγματικό μέρος, αλλά ενώ κάποια έχουν μικρές τιμές, άλλα έχουν αρκετά μεγάλες. Επομένως, εάν η αναλογία που φαίνεται στη σχέση (1.17) είναι μεγάλη, τότε το πρόβλημα είναι άκαμπτο [9].

$$\frac{\max_j -R(\lambda_j)}{\min_j -R(\lambda_j)} \quad (1.17)$$

2. Το πρόβλημα ολοκληρώνεται σε εξαιρετικά μεγάλα διαστήματα ολοκλήρωσης και εμφανίζει “stiffness”.

3. Οι λύσεις του προβλήματος έχουν όρους που φθίνουν πολύ πιο γρήγορα από άλλους.
4. Εάν μια αριθμητική μέθοδος με πεπερασμένη περιοχή απόλυτης σταθερότητας που εφαρμόζεται σε ένα σύστημα με οποιαδήποτε αρχική συνθήκη εξαναγκάζεται να χρησιμοποιήσει σε συγκεκριμένο διάστημα ολοκλήρωσης βήμα το οποίο είναι αρκετά μικρό σε σχέση με την ομαλότητα της συγκεκριμένης λύσης σε αυτό το διάστημα, τότε το πρόβλημα είναι άκαμπτο στο συγκεκριμένο διάστημα.

Για την επίλυση των άκαμπτων προβλημάτων χρησιμοποιούνται έμμεσες μέθοδοι, όπου σε κάθε νέο βήμα πρέπει να επιλύεται ένα μη γραμμικό σύστημα, κάτι το οποίο αυξάνει το υπολογιστικό κόστος. Τα ζεύγη έμμεσων RK, η έμμεση μέθοδος του Euler που ανήκει στις μονοβηματικές μεθόδους και η μέθοδος Simpson, που ανήκει στις πολυβηματικές μεθόδους είναι παραδείγματα τέτοιων μεθόδων [9,10].

### **1.2.3 Επίλυση Διαφορικών Εξισώσεων με Νευρωνικά**

Τα ΤΝΔ εμπρόσθιας τροφοδότησης μπορούν να επιλύσουν τόσο Συνήθεις όσο και Μερικές Διαφορικές Εξισώσεις, λόγω της ιδιότητας τους να προσεγγίζουν οποιαδήποτε συνάρτηση. Η λύση που προκύπτει είναι διαφορίσιμη και σε κλειστή, αναλυτική μορφή. Το ΤΝΔ αποτελεί το βασικό στοιχείο προσέγγισης, του οποίου οι παράμετροι προσαρμόζονται ώστε να ελαχιστοποιείται η συνάρτηση κόστους, η οποία συνάρτηση κόστους δείχνει πόσο αποτυγχάνει η προτεινόμενη λύση να προσεγγίσει την Διαφορική Εξίσωση. Η διαδικασία της ελαχιστοποίησης γίνεται με οποιαδήποτε τεχνική βελτιστοποίησης.

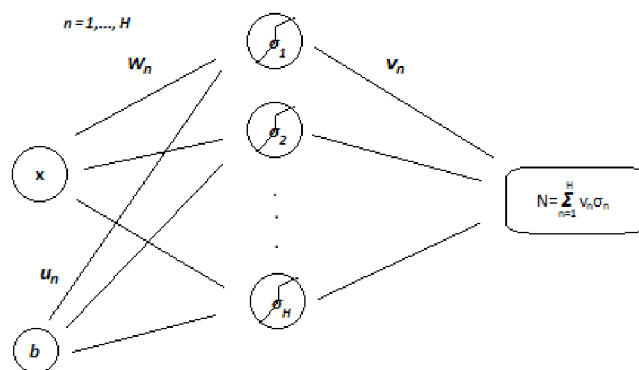
Τα θετικά της επίλυσης Διαφορικών Εξισώσεων με ΤΝΔ είναι, εκτός από τη κλειστή αναλυτική μορφή που προαναφέρθηκε, η ύπαρξη πολύ καλών ιδιοτήτων σε αντίθεση με άλλες υπολογιστικές μεθόδους, η υλοποίηση της μεθόδου σε παράλληλες αρχιτεκτονικές και η δυνατότητα που δίνουν τα ΤΝΔ για την επίλυση σε πραγματικό χρόνο δύσκολων Διαφορικών Εξισώσεων που προκύπτουν σε επιστημονικές εφαρμογές και εφαρμογές μηχανικών [1].

Αρκετές από τις έρευνες που έχουν γίνει στο συγκεκριμένο θέμα έχουν χρησιμοποιήσει ΤΝΔ εμπρόσθιας τροφοδότησης, πολλών στιβάδων (MLP). Οι Λαγαρής, Λίκας και Φωτιάδης [3], η μέθοδος των οποίων παρουσιάζεται αναλυτικά στην παράγραφο 1.2.4 αποτέλεσε την έρευνα στην οποία βασίστηκε η συγκεκριμένη εργασία. Αποτελεί μια μέθοδο η οποία έχει πολύ καλές ικανότητες γενίκευσης και μπορεί να επιλύσει τόσο Συνήθεις όσο και Μερικές Διαφορικές Εξισώσεις. Μια ακόμα μέθοδος η οποία βασίζεται στη χρήση του backpropagation αλγορίθμου παρουσιάζεται από τους He κ.α., στην οποία επιλύεται μία κλάση από πρώτης τάξης Μερικές Διαφορικές Εξισώσεις που χρησιμοποιούνται ως είσοδος για την προσέγγιση γραμμικών συστημάτων [13].

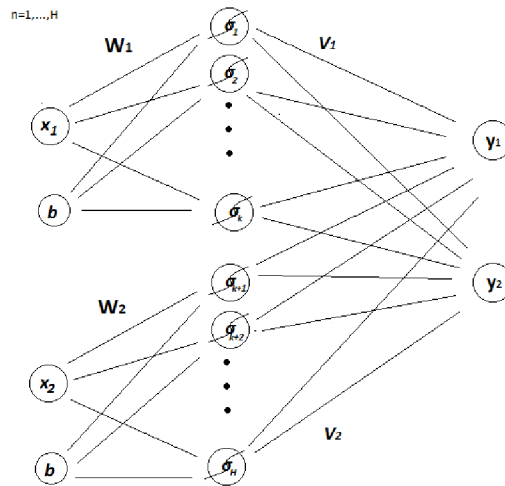
Υπάρχουν επιπλέον έρευνες οι οποίες συνδυάζουν τη χρήση ΤΝΔ και εξελικτικών αλγορίθμων. Μία από αυτές πραγματοποιήθηκε από τους Aarts και Veer [14], οι οποίοι ανέπτυξαν μια μέθοδο για την επίλυση Μερικών Διαφορικών Εξισώσεων και των συνοριακών ή αρχικών τους συνθηκών. Για τη μέθοδο τους χρησιμοποιήθηκαν ΤΝΔ με μία κρυφή στιβάδα, με πολλές εισόδους, αλλά μόνο μία έξοδο και χωρίς τη χρήση πόλωσης, τα οποία έχουν την ιδιότητα να προσεγγίζουν οποιαδήποτε συνάρτηση και της παραγώγους της. Επιπλέον οι Smaoui και Al-Enezi [15] ανέπτυξαν μια μέθοδο που συνδυάζει τα ΤΝΔ και το “Karhunen- Loeve decomposition” για την επίλυση δύο μη-γραμμικών Μερικών Διαφορικών Εξισώσεων.

#### 1.2.4 Μέθοδος Λαγαρή, Λίκα & Φωτιάδη

Μία από τις σημαντικές έρευνες για την επίλυση Συνήθων αλλά και Μερικών Διαφορικών Εξισώσεων έγινε από τους Λαγαρή, Λίκα και Φωτιάδη το 1998 [3]. Η ιδέα τους βασίστηκε στην ιδιότητα των ΤΝΔ εμπρόσθιας τροφοδότησης να προσεγγίζουν οποιαδήποτε συνάρτηση. Η σχηματική αναπαράσταση του ΤΝΔ που πρότειναν φαίνεται στην Εικόνα 1.9 και πιο συγκεκριμένα για την επίλυση συστημάτων ΣΔΕ 2x2 στην Εικόνα 1.10.



Εικόνα 1.9: ΤΝΔ για επίλυση ΣΔΕ & ΜΔΕ όπως προτείνεται από Λαγαρή κ.α



Εικόνα 1.10: ΤΝΔ για επίλυση ΣΔΕ & ΜΔΕ συστημάτων όπως προτείνεται από Λαγαρή κ.α

Η γενική μορφή της εξίσωσης δίνεται από τη σχέση (1.18), με την οποία έπειτα από μια διακριτοποίηση του χώρου  $D$  και του ορίου του  $S$ , σε ένα σύνολο σημείων, το πρόβλημα μετατρέπεται σε ένα σύστημα εξισώσεων, όπως φαίνεται στη σχέση (1.19).

$$G(\vec{x}, \Psi(\vec{x}), \nabla \Psi(\vec{x}), \nabla^2 \Psi(\vec{x})) = 0, \quad \vec{x} \in D \quad (1.18)$$

$$G(\vec{x}_i, \Psi(\vec{x}), \nabla \Psi(\vec{x}_i), \nabla^2 \Psi(\vec{x}_i)) = 0, \quad \vec{x}_i \in D \quad (1.19)$$

Επομένως προκύπτει ένα πρόβλημα ελαχιστοποίησης, όπως φαίνεται στη σχέση (1.20), το οποίο δείχνει πόσο η δοκιμαστική λύση ( $\Psi$ ) αποτυγχάνει να προσεγγίσει την διαφορική εξίσωση. Η δοκιμαστική λύση είναι μία λύση η οποία δοκιμάζεται και η οποία θεωρείται ότι θα ικανοποιήσει τη Διαφορική Εξίσωση.

$$E[\vec{p}] = \sum_i \left\{ \frac{dy_t(x_i)}{dx} - f(x_i, \Psi_t(x_i)) \right\}^2 \quad (1.20)$$

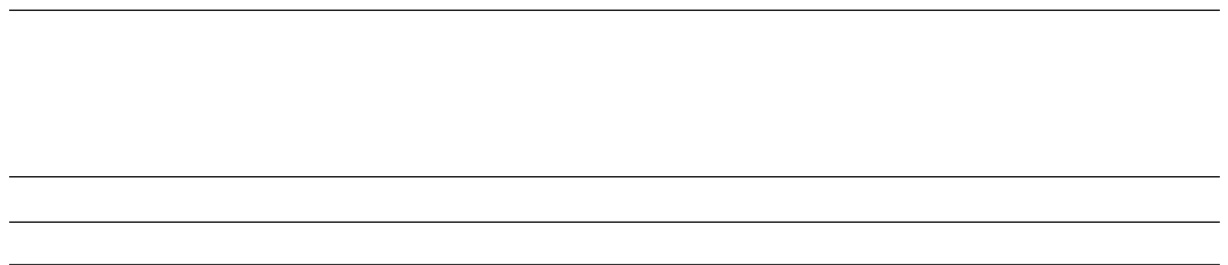
Η δοκιμαστική λύση που προτείνουν οι Λαγαρή, Λίκας και Φωτιάδης για τα Π.Α.Τ 1<sup>ης</sup> τάξης δίνεται από τη σχέση (1.21) και αποτελείται από δύο όρους. Ο πρώτος ( $A(x)$ ) ικανοποιεί τις συνοριακές και αρχικές συνθήκες και δεν αποτελείται από ρυθμιζόμενες παραμέτρους, ενώ ο δεύτερος όρος ( $F(x, N(x, p))$ ) αποτελείται από ένα ΤΝΔ εμπρόσθιας τροφοδότησης, το οποίο έχει μία κρυφή στιβάδα.

$$\Psi_t = A(x) + F(x, N(x, p)) \quad (1.21)$$

Η μορφή αυτής της μεθόδου, μετατρέπει το αρχικό πρόβλημα από ένα πρόβλημα βελτιστοποίησης με περιορισμούς σε ένα πρόβλημα χωρίς περιορισμούς, καθώς αυτοί ικανοποιούνται. Επομένως οι παράμετροι του νευρωνικού (p), δηλαδή τα συναπτικά βάρη (W, V) και η πόλωση (b), πρέπει να ρυθμίζονται έτσι ώστε να ελαχιστοποιηθεί μία συνάρτηση κόστους.

Για την ελαχιστοποίηση της συνάρτησης κόστους η οποία φαίνεται στη σχέση (1.22), πρέπει να βρεθούν οι καλύτερες τιμές των παραμέτρων, κάτι το οποίο επιτυγχάνεται με τη διαδικασία της εκπαίδευσης, η οποία στην προκειμένη περίπτωση πραγματοποιείται με τη μέθοδο quasi-Newton BFGS [3].

$$\min_{\vec{p}} \left( \sum_{\vec{x}_i \in D} G \left( \vec{x}_i, \Psi_c(\vec{x}_i, \vec{p}), \nabla \Psi_c(\vec{x}_i, \vec{p}), \nabla^2 \Psi_c(\vec{x}_i, \vec{p}) \right)^2 \right) \quad (1.22)$$



## ΚΕΦΑΛΑΙΟ 2: Μεθοδολογία της έρευνας

---

Σκοπός της παρούσας εργασίας είναι η εκπαίδευση ΤΝΔ για την επίλυση άκαμπτων Δ.Ε, επομένως πρόκειται για μια ποιοτική έρευνα με τρεξίματα κώδικα. Τα εργαλεία που χρησιμοποιήθηκαν είναι τα ακόλουθα:

- ⌚ *MATLAB* : για την επίλυση των Δ.Ε τόσο με τη χρήση ΤΝΔ, όσο και με τις μεθόδους που έχει ενσωματωμένες ήδη.
- ⌚ *MATHEMATICA* : για την εύρεση της αναλυτικής λύσης των Δ.Ε.

Η μεθοδολογία που ακολουθήθηκε αναλύεται παρακάτω:

### **1. Έλεγχος της ακαμψίας του προβλήματος**

Αρχικά, εξετάζεται εάν το πρόβλημα είναι άκαμπτο (stiff) με βάση τους ορισμούς που περιγράφονται στο κεφάλαιο 1.2.2. Για τη διαπίστωση της ακαμψίας των προβλημάτων που επιλύθηκαν χρησιμοποιήθηκαν άμεσες και έμμεσες μέθοδοι επίλυσης της *MATLAB* και από τα αποτελέσματά του σφάλματος, του αριθμού βημάτων και το υπολογιστικό κόστος εξάγεται το συμπέρασμα αν υπάρχει ακαμψία στο πρόβλημα. Τα αποτελέσματα για κάθε πρόβλημα φαίνονται στους πίνακες του Παραρτήματος Α.

### **2. Λύση του άκαμπτου προβλήματος με τη χρήση Τεχνητών Νευρωνικών Δικτύων.**

#### *2.1 Εκπαίδευση του ΤΝΔ*

Πρώτο βήμα είναι η εκπαίδευση των ΤΝΔ εμπρόσθιας τροφοδότησης με μία κρυφή στιβάδα, με τη μέθοδο Levenberg-Marquardt και με τη χρήση διαφορετικών συναρτήσεων ενεργοποίησης. Η εκπαίδευση γίνεται σε σημεία εκπαίδευσης του διαστήματος τιμών στο οποίο επιλύεται το πρόβλημα.

#### *2.2 Αξιολόγηση του ΤΝΔ.*

Για την αξιολόγηση του ΤΝΔ εξετάζονται περισσότερα σημεία, από τα σημεία εκπαίδευσης εντός του διαστήματος επίλυσης, στα οποία υπολογίζεται το ολικό σφάλμα, δηλαδή η διαφορά της τιμής που προκύπτει από το ΤΝΔ και της τιμής που προκύπτει από την αναλυτική λύση.

### 3 Σύγκριση με ήδη υπάρχουσες μεθόδους του Matlab.

Για τη σύγκριση των σφαλμάτων που προκύπτουν από το TND με άλλες μεθόδους για επίλυση ΣΔΕ χρησιμοποιήθηκαν οι μέθοδοι της MATLAB που παρουσιάζονται παρακάτω [11]:

*Ode23*: Είναι ένας επιλυτής μη-άκαμπτων προβλημάτων και ένας μικρής τάξης επιλυτής. Η μέθοδος αυτή αποτελεί υλοποίηση της Bogacki-Shampine (2,3), η οποία είναι μία μέθοδος Runge-Kutta τρίτης τάξης, με 4 στάδια με FSAL ιδιότητα, επομένως χρησιμοποιεί προσεγγιστικά τρεις υπολογισμούς συναρτήσεων σε κάθε βήμα. Είναι μονοβηματική μέθοδος και μπορεί να είναι πιο αποδοτική από την *Ode45* σε έντονες διακυμάνσεις και στην ύπαρξη ακαμψίας στο πρόβλημα.

*Ode45*: Είναι ένας επιλυτής μη-άκαμπτων προβλημάτων, μεσαίας τάξης. Η μέθοδος *Ode45* βασίζεται σε μία άμεση Runge-Kutta (4,5), την Dormand-Prince pair. Είναι μία μέθοδος που έχει 7 στάδια αλλά χρησιμοποιεί μόνο 6 υπολογισμούς συναρτήσεων σε κάθε βήμα, λόγω του ότι έχει FSAL ιδιότητα.

*Ode15s*: Η *Ode15s* είναι ένας μεταβλητού βήματος και μεταβλητής τάξης επιλυτής ο οποίος βασίζεται σε αριθμητικές διαφορίσιμες μεθόδους τάξεως 1-5. Είναι ένας πολυβηματικός επιλυτής ο οποίος χρησιμοποιείται έναντι της *Ode45* όταν η τελευταία αποτυγχάνει ή είναι αρκετά ανεπαρκής και όταν πρόκειται για ένα άκαμπτο πρόβλημα.

*Ode23s*: Η μέθοδος *ode23s* προτείνεται ως μια εναλλακτική της μεθόδου *Ode15s* για την επίλυση άκαμπτων προβλημάτων, κυρίως όταν υπάρχουν έντονες διακυμάνσεις, όταν μια μονοβηματική μέθοδος έχει πλεονεκτήματα έναντι μεθόδων με μνήμη και όταν οι Ιακωβιανοί πίνακες έχουν ιδιοτιμές κοντά στον άξονα των φανταστικών αριθμών.

*Radau5*: Η *Radau5* χρησιμοποιεί μία έμμεση Runge-Kutta μέθοδο (*Radau IIa*) 5<sup>ης</sup> τάξης, τριών σταδίων. Αποτελεί μια μέθοδο τουλάχιστον συγκρίσιμη με τους καλύτερους από τους άλλους ολοκληρωτές όσον αφορά την ταχύτητα και υπερέρχει όσον αφορά την ακρίβεια [12].

Η προαναφερθείσα μεθοδολογία χρησιμοποιείται για την επίλυση κάθε προβλήματος που αναφέρεται στην παρούσα εργασία. Τα προβλήματα τα οποία επιλύονται ανήκουν στις ακόλουθες τρεις κατηγορίες:

- Ακαμπτα προβλήματα.
- Ακαμπτα προβλήματα με σταθερή παράμετρο.
- Ακαμπτα προβλήματα με οικογένεια παραμέτρων

## ΚΕΦΑΛΑΙΟ 3: Η προτεινόμενη μέθοδος – Θεμελίωση, Σχεδίαση, Ανάπτυξη

### 3.1 Τροποποίηση της αρχιτεκτονικής των Λαγαρή, Λίκα & Φωτιάδη

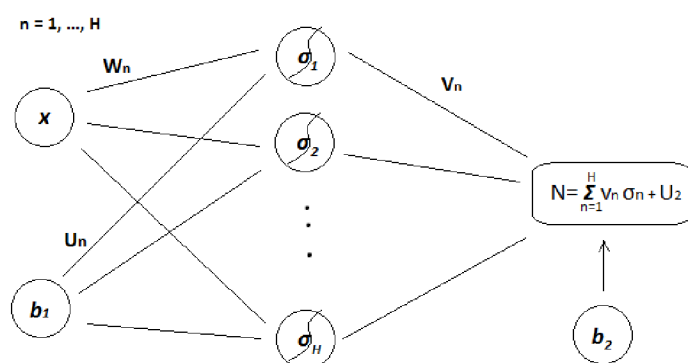
Η παρούσα εργασία στηρίζεται στη μέθοδο που πρότειναν οι Λαγαρή κ.α, για την επίλυση Δ.Ε και κυρίως Σ.Δ.Ε. Η μέθοδος τους περιγράφεται αναλυτικά στο κεφάλαιο 1.2.4. Έπειτα από τη μελέτη της ήδη υπάρχουσας μεθόδου και ιδίως της αρχιτεκτονικής του ΤΝΔ που αναπτύχθηκε στη μέθοδο αυτή, διερευνάται η χρήση μιας τροποποιημένου μεθόδου για τη γενική επίλυση Σ.Δ.Ε και μιας διαφορετικής αρχιτεκτονικής για την επίλυση συστημάτων Σ.Δ.Ε 2x2 .

Πιο συγκεκριμένα, για τη γενική επίλυση Σ.Δ.Ε μελετάται η χρήση μιας ακόμα πόλωσης στην έξοδο του ΤΝΔ κάτι το οποίο μετατρέπει την αρχική σχέση 3.1 των Λαγαρή κ.α, στην (3.2).

$$Neural\_network = (V * activation(W * x + U)) \quad (3.1)$$

$$Neural\_network = (V * activation(W * x + U) + U2) \quad (3.2)$$

Στην Εικόνα 3.1 διακρίνεται η μορφή του ΤΝΔ που μελετάται για τη γενική επίλυση ΣΔΕ και περιγράφεται από την Εξίσωση 3.2. Σε σύγκριση με την Εικόνα 1.5 φαίνεται η ύπαρξη της δεύτερης πόλωσης στην έξοδο του ΤΝΔ.

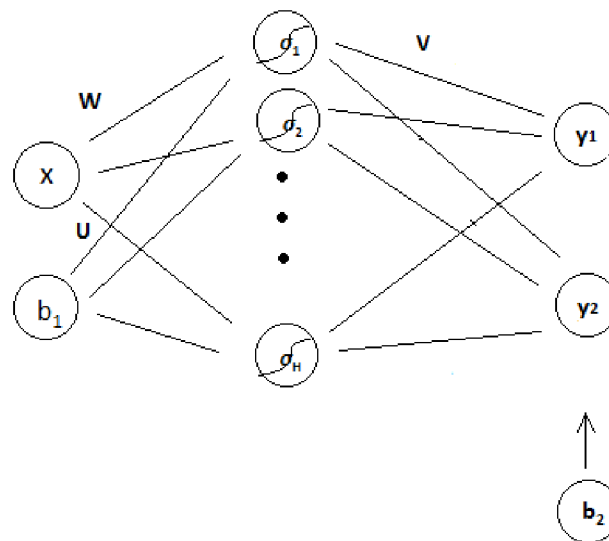


Εικόνα 3.1 : ΤΝΔ που μελετάται για επίλυση προβλημάτων ΣΔΕ

Στην Εικόνα 3.2 φαίνεται η αρχιτεκτονική του ΤΝΔ που μελετάται για την επίλυση συστημάτων ΣΔΕ 2 x 2, στο οποίο γίνεται χρήση ενός ΤΝΔ για την παραγωγή των δύο λύσεων αντί για δύο ΤΝΔ τα οποία συνδέονται μόνο στην τελική στιβάδα για να



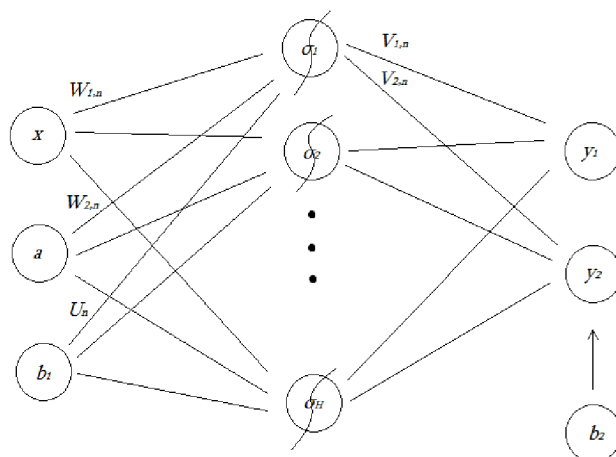
παράγουν τις τελικές λύσεις, όπως φαίνεται στην Εικόνα 1.6 και προτείνουν οι Λαγαρής κ.α.



Εικόνα 3.2 : ΤΝΔ που μελετάται για επίλυση συστημάτων Σ.Δ.Ε 2x2

### 3.2 Παραμετρικοποιημένα προβλήματα

Μία επιπλέον διερεύνηση έγινε στην επίλυση Δ.Ε με τη χρήση ΤΝΔ τα οποία εκπαιδεύονται να έχουν ως δεδομένα εισόδου μία ή περισσότερες παραμέτρους του προβλήματος. Με αυτόν τον τρόπο μπορούν να επιλυθούν οικογένειες προβλημάτων, αντί για μεμονωμένες εξισώσεις. Η έρευνα που πραγματοποιείται βασίζεται στη δημοσίευση των Lazovskaya και Tarkhon, που ασχολείται με την επίλυση μίας άκαμπτης Σ.Δ.Ε για ένα διάστημα τιμών της παραμέτρου [15]. Η αρχιτεκτονική του ΤΝΔ που μελετάται φαίνεται στην Εικόνα 3.3.



Εικόνα 3.3: ΤΝΔ που μελετάται για επίλυση παραμετρικοποιημένων προβλημάτων

### 3.3 Περιγραφή κώδικα για την επίλυση ΣΔΕ με τη χρήση ΤΝΔ.

Ο κώδικας ο οποίος αναπτύχθηκε για την επίλυση των προαναφερθέντων κατηγοριών προβλημάτων και φαίνεται αναλυτικά στο Παράρτημα Δ περιλαμβάνει τα εξής βήματα:

#### 1. Επιλέγεται το πρόβλημα το οποίο επιλύεται

Τα προβλήματα μπορούν να ανήκουν στην γενικότερη κατηγορία των άκαμπτων προβλημάτων, στην κατηγορία των άκαμπτων προβλημάτων με την ύπαρξη παραμέτρου ή στην κατηγορία των άκαμπτων προβλημάτων με οικογένεια παραμέτρων.

#### 2. Ορισμός του ΤΝΔ.

Οι παράμετροι του ΤΝΔ που ορίζονται σε αυτό το στάδιο είναι:

- ⌚ Ο αριθμός των σημείων τα οποία θα χρησιμοποιηθούν στην εκπαίδευση. Τα  $n+1$  αυτά σημεία, διαμερίζουν το διάστημα τιμών στο οποίο επιλύεται το πρόβλημα σε  $n$  ισαπέχοντα διαστήματα.
- ⌚ Ο αριθμός των νευρώνων της κρυφής στιβάδας.
- ⌚ Ο αριθμός των επαναλήψεων της διαδικασίας εκπαίδευσης του ΤΝΔ, δηλαδή πόσα “forward” και “backward” περάσματα θα πραγματοποιηθούν για όλα τα σημεία εκπαίδευσης.
- ⌚ Η συνάρτηση ενεργοποίησης που θα χρησιμοποιηθεί και ο υπολογισμός της πρώτης και της δεύτερης παραγώγου της.

### 3. Εκπαίδευση του ΤΝΔ

Αρχικά θέτονται τυχαίες αρχικές τιμές για τις παραμέτρους του ΤΝΔ., δηλαδή τα συναπτικά βάρη μεταξύ στιβάδας εισόδου-κρυφής στιβάδας και κρυφής στιβάδας-στιβάδας εξόδου και οι πολώσεις. Στη συνέχεια με τη χρήση του έτοιμου αλγορίθμου Levenberg-Marquardt της Matlab, καθορίζονται οι τιμές των παραμέτρων του ΤΝΔ που ελαχιστοποιούν τη συνάρτηση σφάλματος, η οποία φαίνεται από τη σχέση (1.21).

### 4. Αξιολόγηση του ΤΝΔ

Έπειτα από την εκπαίδευση του ΤΝΔ, γίνεται μία αξιολόγηση αυτού σε περισσότερα σημεία από αυτά που έχει εκπαιδευτεί με τη χρήση των παραμέτρων οι οποίες έχουν προκύψει από την εκπαίδευση. Το σφάλμα (*sfalma*) το οποίο υπολογίζεται από τον τύπο 3.4 είναι η μέγιστη τιμή της απόλυτης διαφοράς μεταξύ της αναλυτικής λύσης και της λύσης που προκύπτει από τον τύπο 3.3. Όσο μεγαλύτερο είναι το σφάλμα το οποίο προκύπτει, τόσο λιγότερο επιτυγχάνει το ΤΝΔ να προσεγγίσει την αναλυτική λύση.

$$res = y0 + (x_{space1} - x0) * neuralnetwork \quad (3.3)$$

$$sfalma = \max(abs(y_{sol}(x_{space1}) - res)) \quad (3.4)$$

### 5. Έλεγχος *overfitting*

Τέλος, αφού έχουν προκύψει από την εκπαίδευση και την αξιολόγηση οι καλύτερες τιμές των παραμέτρων του ΤΝΔ και συνεπώς η βέλτιστη τιμή σφάλματος που επιτυγχάνεται, πραγματοποιείται έλεγχος για την ύπαρξη “*overfitting*”. Ο έλεγχος αυτός πραγματοποιείται με τον υπολογισμό του ολικού σφάλματος και τον υπολογισμό του σφάλματος στα δεδομένα εκπαίδευσης μόνο και τη σύγκρισή τους. Στην περίπτωση διαφοράς μεταξύ των δύο σφαλμάτων υπάρχει “*overfitting*”, καθώς το ΤΝΔ δεν μπορεί να γενικεύσει σωστά.

## ΚΕΦΑΛΑΙΟ 4: Εφαρμογή και Αποτελέσματα

---

Όλα τα προβλήματα τα οποία επιλύθηκαν είναι Προβλήματα Αρχικών Τιμών (Π.Α.Τ.) στα οποία είναι γνωστή η αναλυτική λύση. Αρχικά επιλύθηκαν ΣΔΕ και συστήματα ΣΔΕ 2x2 που ανήκουν στη γενικότερη κατηγορία των άκαμπτων προβλημάτων, όπως τα προβλήματα 1 και 2. Σε όλα τα προβλήματα έχουν γίνει 40 επαναλήψεις στο ΤΝΔ. Επιπλέον γίνεται scaling στα ΤΝΔ, ενώ η τιμή η οποία υπολογίζεται για κάθε πρόβλημα και συγκρίνεται με τις μεθόδους της Matlab [11,20], είναι ο αριθμός ψηφίων ακριβείας στο σφάλμα. Στις μεθόδους της Matlab έχει επιλεγεί η προκαθορισμένη τιμή για την ανοχή, δηλαδή e-03 και e-06.

### 4.1 Άκαμπτα προβλήματα

#### Πρόβλημα 1:

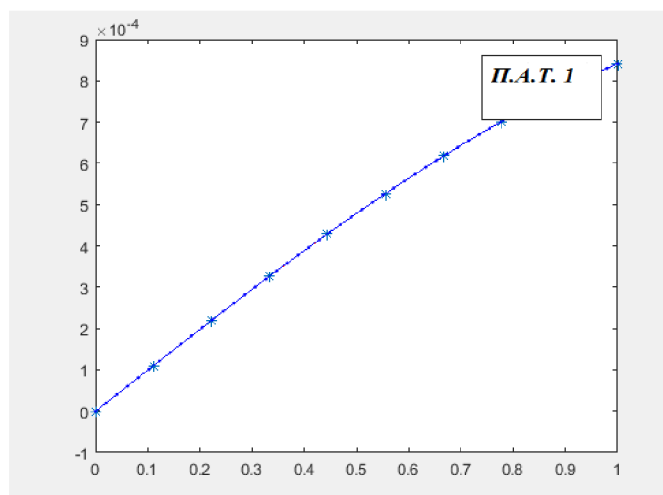
Το πρώτο πρόβλημα αποσκοπεί στη μελέτη ενός μοντέλου (Εικόνα 3.1 ) το οποίο είναι πιο γενικό από αυτό που χρησιμοποίησαν οι Λάγαρης κ.α [3], καθώς η ύπαρξη μίας ακόμα πόλωσης στην έξοδο του ΤΝΔ, εισάγει ακόμα μία παράμετρο, η οποία σε περίπτωση που δεν βοηθάει στην επίλυση ορισμένων προβλημάτων μπορεί να λάβει την τιμή 0.

Στο Π.Α.Τ. (4.1) παρουσιάζεται μια ΣΔΕ, η οποία έχει ένα κυριαρχούν γραμμικό όρο. Στη σχέση (4.2) παρουσιάζεται η αναλυτική λύση. Τα διαστήματα τιμών για τα οποία επιλύεται το συγκεκριμένο πρόβλημα είναι το [0,1] και το [0,10].

$$y' = -1000y + \sin(t), \quad y(0) = \frac{-1}{1000001} \quad (4.1)$$

$$y(t) = \frac{1000 \sin(t) - \cos(t)}{1000001} \quad (4.2)$$

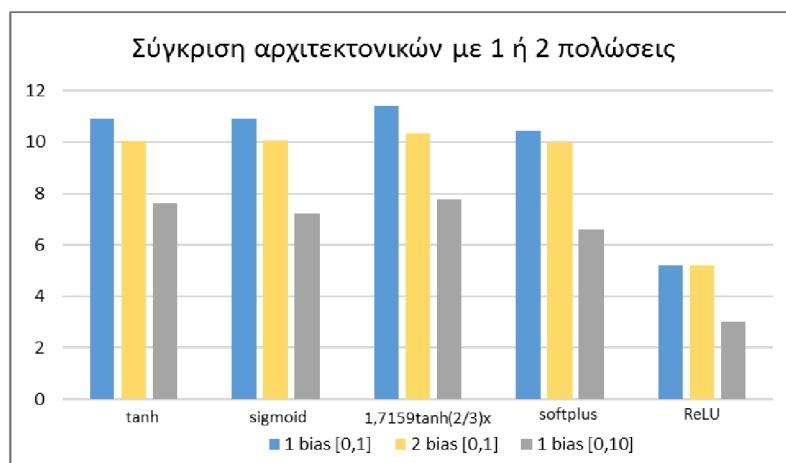
Στο Διάγραμμα 4.1 παρουσιάζεται η αναλυτική λύση του Π.Α.Τ. 1



Διάγραμμα 4.1 : Αναλυτική λύση του Π.Α.Τ. 1

Όπως διακρίνεται από τον Πίνακα Α.1 του Παραρτήματος Α, το πρόβλημα είναι άκαμπτο, κάτι το οποίο αποδεικνύεται και από τον Πίνακα Α.27, επειδή ο αριθμός των βημάτων που κάνει ένας stiff επιλυτής είναι πολύ λιγότερα από τα βήματα των non-stiff επιλυτών.

Στο Διάγραμμα 4.2 παρουσιάζεται μία σύγκριση των αποτελεσμάτων που προκύπτουν με τη χρήση 1 και 2 πολώσεων για κάθε συνάρτηση ενεργοποίησης στο διάστημα τιμών  $[0,1]$ , ενώ παράλληλα φαίνεται η βέλτιστη τιμή που προκύπτει για κάθε συνάρτηση ενεργοποίησης στο διάστημα τιμών  $[0,10]$ . Τα αποτελέσματα φαίνονται αναλυτικά στον πίνακα Β.1 στο Παράρτημα Β.

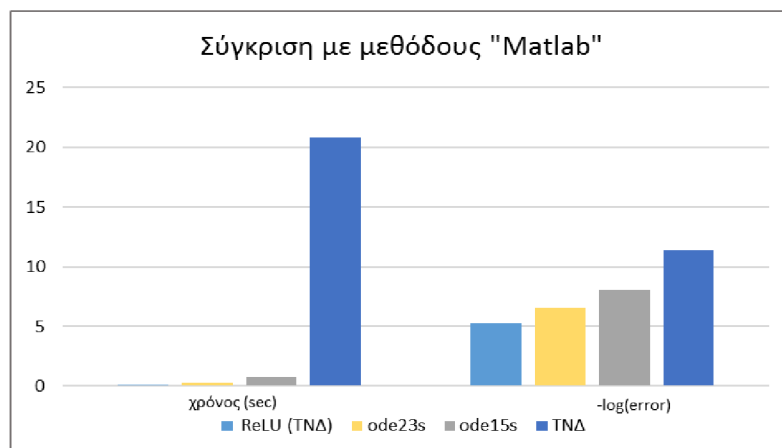


Διάγραμμα 4.2: Σύγκριση μεθόδου με 1 και 2 “bias” για το Π.Α.Τ. 1.

Όπως φαίνεται από το Διάγραμμα 4.2 η ύπαρξη μίας ακόμα πολώσης δεν επηρεάζει ιδιαίτερα την τάξη του σφάλματος του προβλήματος, ενώ η καλύτερη τιμή σφάλματος (11 ψηφία ακρίβειας) στο διάστημα τιμών  $[0,1]$ , με τη χρήση ΤΝΔ, επιτυγχάνεται με τη χρήση της συναρτήσεως ενεργοποίησης  $1.7159 \tanh(2/3x)$  και τη

χρήση μίας πόλωσης. Όσον αφορά την επέκταση της λύσης στο διάστημα τιμών  $[0,10]$  παρατηρούμε ότι το ΤΝΔ δίνει πολύ ικανοποιητικές λύσεις οι οποίες ξεπερνούν τα 6 ψηφία σε ακρίβεια.

Στο Διάγραμμα 4.3 φαίνεται μία σύγκριση των μεθόδων της “Matlab” με τις τιμές δύο ΤΝΔ ( του βέλτιστου ΤΝΔ και του ΤΝΔ με τη χρήση της ReLU) που προκύπτουν για το Π.Α.Τ. 1. Η σύγκριση γίνεται τόσο για την τιμή του σφάλματος που προκύπτει όσο και για το χρόνο που χρειάζεται κάθε μέθοδος για την υλοποίηση της. Τα αποτελέσματα φαίνονται αναλυτικά στον Πίνακα Α.1 στο Παράρτημα Α.



Διάγραμμα 4.3: Σύγκριση με μεθόδους “Matlab” για το Π.Α.Τ. 1.

Από το Διάγραμμα 4.2 παρατηρείται η υπεροχή όλων των άλλων συναρτήσεων ενεργοποίησης έναντι της ReLU, όσον αφορά την ακρίβεια που επιτυγχάνεται, καθώς καταφέρνει να επιτύχει μόνο 5 ψηφία ακρίβειας. Ωστόσο, από το Διάγραμμα 4.3 διακρίνεται η υπεροχή της ReLU στο χρόνο που απαιτείται για την υλοποίηση της, τόσο έναντι των άλλων συναρτήσεων ενεργοποίησης, αλλά και των μεθόδων της “Matlab” που προτείνονται για την επίλυση «άκαμπτων» προβλημάτων. Το χαρακτηριστικό αυτό παρατηρείται λόγω της φύσης της ReLU, η οποία είναι γραμμική και στη συγκεκριμένη περίπτωση χρησιμοποιείται για την επίλυση ενός προβλήματος που διαθέτει ένα κυριαρχούν γραμμικό όρο.

Από το Π.Α.Τ. 1 φαίνεται ότι η χρήση της δεύτερης πόλωσης φαίνεται να μην επηρεάζει αρνητικά τα αποτελέσματα του ΤΝΔ, ενώ η χρήση της ReLU φαίνεται να είναι κατάλληλη για την επίλυση συστημάτων με κυριαρχούν γραμμικό όρο.

### **Πρόβλημα 2:**

Το πρόβλημα 2 είναι ένα “mildly stiff” πρόβλημα, το οποίο επιλύεται στο διάστημα τιμών  $[0,10]$  και χρησιμοποιείται για τη λήψη περισσότερων πληροφοριών σχετικά με την αρχιτεκτονική με τις 2 πολώσεις, αλλά και για την αρχιτεκτονική της Εικόνας 3.2.

Το Π.Α.Τ (4.3), μπορεί να μετατραπεί σε πινακική μορφή που φαίνεται στη σχέση (4.4) και περιγράφει την ψύξη μιας κούπας καφέ η οποία επιλύεται στο διάστημα τιμών [0,10].

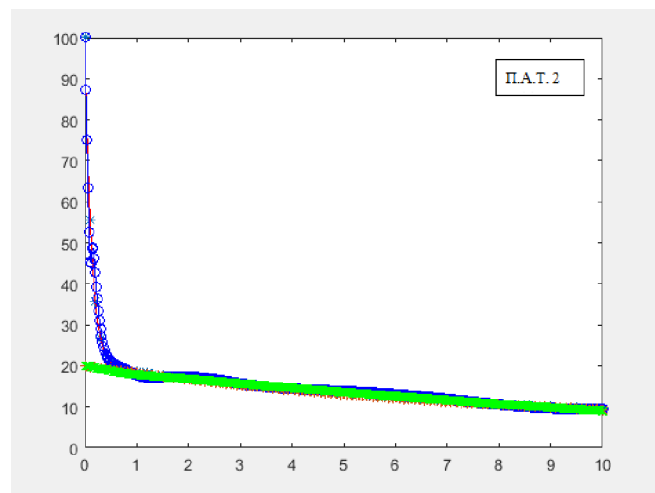
$$x''(t) - ax'(t) - bx(t) = f(t), \quad x(0) = \xi, x'(0) = \eta \quad (4.3)$$

$$x'(t) = Ax(t) + g, \quad x(t) = \begin{bmatrix} u(t) \\ v(t) \end{bmatrix}, A = \begin{bmatrix} -8 & 8 \\ 0 & -\frac{1}{8} \end{bmatrix}, g = \begin{bmatrix} 0 \\ 5/8 \end{bmatrix}, x(0) = [100 \ 20] \quad (4.4)$$

Το  $u(t)$  που περιγράφεται στη σχέση (4.4) περιγράφει τη θερμοκρασία του υγρού σε χρόνο  $t$ . Οι αρχικές συνθήκες του συστήματος είναι  $100^\circ\text{C}$ , που είναι η θερμοκρασία του υγρού και  $20^\circ\text{C}$ , που είναι η θερμοκρασία του δωματίου. Το δωμάτιο ψύχεται ταυτόχρονα με το υγρό με μια αναλογία  $1/8$  και εξωτερική θερμοκρασία  $5^\circ\text{C}$ .

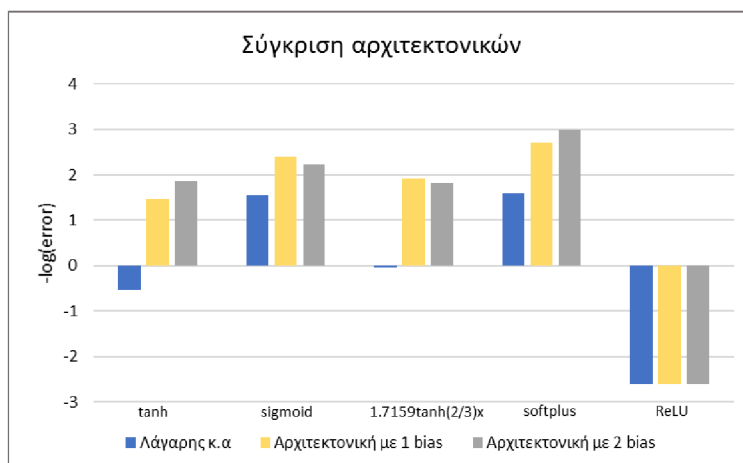
Οι ιδιοτιμές του προβλήματος είναι  $\lambda = -1/8$  και  $\lambda = -8$  κάτι το οποίο καθιστά το πρόβλημα άκαμπτο με βάση τον πρώτο ορισμό του υποκεφαλαίου 1.2.2 [9]. Η ακαμψία του προβλήματος επιβεβαιώνεται και από τον Πίνακα Α.28 του παραρτήματος Α, επειδή ο αριθμός των βημάτων που κάνει ένας stiff επιλυτής είναι πολύ λιγότερα από τα βήματα των non-stiff επιλυτών.

Στο Διάγραμμα 4.4 παρουσιάζεται η αναλυτική λύση του Π.Α.Τ. 2.



Διάγραμμα 4.4: Αναλυτική λύση του Π.Α.Τ. 2

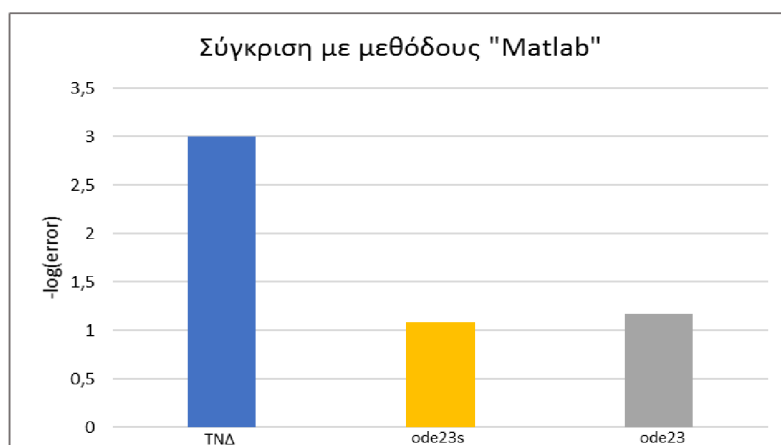
Στο Διάγραμμα 4.5 παρουσιάζονται οι βέλτιστες τιμές των σφαλμάτων που προκύπτουν από τα ΤΝΔ με τη χρήση διαφορετικών συναρτήσεων ενεργοποίησης. Παράλληλα παρουσιάζεται μία σύγκριση μεταξύ της αρχιτεκτονικής που προτείνουν οι Λαγαρές κ.α για την επίλυση συστημάτων ΣΔΕ 2x2 και της αρχιτεκτονικής που μελετάται τόσο με τη χρήση 1 bias όσο και με 2 bias. Στον πίνακα Β.2 του παραρτήματος Β, φαίνονται αναλυτικά τα αποτελέσματα αυτών των συγκρίσεων.



Διάγραμμα 4.5: Σύγκριση αρχιτεκτονικών για το Π.Α.Τ. 2.

Από το Διάγραμμα 4.5 παρατηρείται ότι η αρχιτεκτονική που μελετάται και φαίνεται στην Εικόνα 3.2, παράγει καλύτερα αποτελέσματα από την αρχιτεκτονική που προτείνουν οι Λαγαρής κ.α για την επίλυση συστημάτων ΣΔΕ 2x2. Επιπλέον η χρήση 2 πολώσεων δίνει το καλύτερο αποτέλεσμα με ακρίβεια 3 ψηφίων και τη χρήση της συναρτήσεως ενεργοποίησης Softplus. Η χρήση της ReLU δίνει τα χειρότερα αποτελέσματα.

Στο Διάγραμμα 4.6 φαίνεται μία σύγκριση των μεθόδων της “Matlab” με την τιμή του ΤΝΔ που προκύπτει για το Π.Α.Τ. 2. Τα αποτελέσματα φαίνονται αναλυτικά στον Πίνακα Α.2 στο Παράρτημα Α.



Διάγραμμα 4.6: Σύγκριση αρχιτεκτονικών για το Π.Α.Τ. 2.

Από το Διάγραμμα 4.6 διακρίνεται ότι το ΤΝΔ επιτυγχάνει καλύτερη ακρίβεια από τις συγκρινόμενες μεθόδους της “Matlab”. Η σύγκριση γίνεται με τη μέθοδο ode23 η οποία είναι “non-stiff” επιλυτής και μπορεί να είναι πιο αποδοτική σε σχέση με την ode45 σε προβλήματα που έχουν μέτρια ακαμψία. Επίσης γίνεται σύγκριση με τη μέθοδο ode23s η οποία είναι ένας “stiff” επιλυτής και επιλύει προβλήματα στα οποία η ode15s δεν είναι τόσο αποδοτική.

Από το Π.Α.Τ. 2 προκύπτει το συμπέρασμα ότι η αρχιτεκτονική της Εικόνας 3.2 είναι καλύτερη από την προτεινόμενη αρχιτεκτονική των Λαγαρή κ.α., όπως επίσης ότι η χρήση 2<sup>ης</sup> πολώσης δίνει καλύτερα αποτελέσματα.



- Τα επόμενα άκαμπτα προβλήματα που επιλύονται και περιέχουν σταθερή παράμετρο, η οποία χρησιμοποιείται ως δεδομένο εισόδου στο ΤΝΔ, είναι συστήματα ΣΔΕ 2x2. Τα συστήματα αυτά επιλύονται με τη χρήση ΤΝΔ τα οποία λαμβάνουν τη μορφή της Εικόνας 3.2 και εξετάζονται για τρεις διαφορετικές συναρτήσεις ενεργοποίησης. Η χρήση της αρχιτεκτονικής των Λαγαρή κ.α. απορρίφθηκε έπειτα από την εξέταση των Προβλημάτων 1 & 2 , όπως επίσης απορρίφθηκε και η χρήση των συναρτήσεων softplus και ReLU, καθώς η softplus καταναλώνει υπερβολικό υπολογιστικό χρόνο, χωρίς να δίνει πολύ καλύτερα αποτελέσματα από τις υπόλοιπες συναρτήσεις, ενώ η ReLU δεν παράγαγε καλά αποτελέσματα. Η επίλυση των παρακάτω προβλημάτων βασίστηκε στη δημοσίευση των Lazovskaya και Tarkhov [15].

## 4.2 Άκαμπτα προβλήματα με παράμετρο

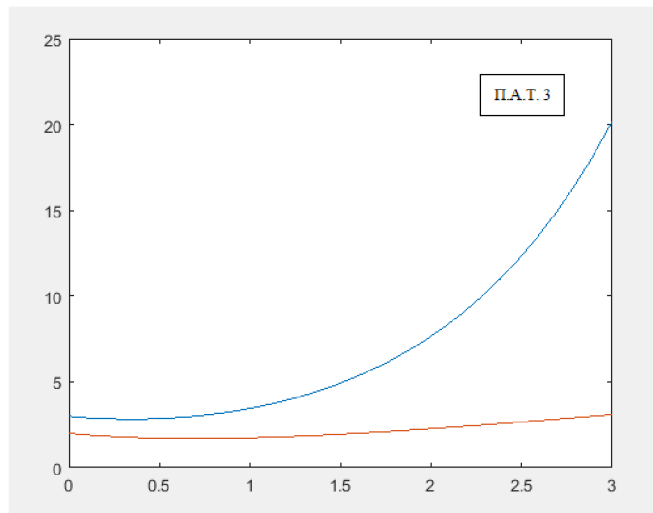
### Πρόβλημα 3:

Το πρώτο άκαμπτο πρόβλημα το οποίο περιέχει παράμετρο  $a$  και επιλύεται με την είσοδο στο ΤΝΔ σταθερής τιμής για την παράμετρο είναι το Π.Α.Τ. που περιγράφεται στη σχέση (4.5). Η αναλυτική λύση του συστήματος δίνεται από τη σχέση (4.6). Το διάστημα τιμών στο οποίο επιλύεται το πρόβλημα είναι το  $[0,3]$ . Η παράμετρος  $a$ , η οποία δεν εμπεριέχεται στην αναλυτική λύση έχει λάβει τις τιμές -8, -98, -998, -9998. Για τις τιμές της παραμέτρου  $a$ , έχει γίνει κανονικοποίηση των δεδομένων σε διάστημα τιμών  $[0,10]$  πριν την εκπαίδευση του ΤΝΔ.

$$\begin{aligned} y_1'(t) &= -2y_1(t) + y_2(t) + 3e^t - t, & y_1(0) &= 3 \\ y_2'(t) &= -ay_1(t) + (a-1)y_2(t) + ae^t - (a-1)t + 1, & y_2(0) &= 2 \end{aligned} \quad (4.5)$$

$$\begin{aligned} y_1(t) &= 2e^{-t} + e^t \\ y_2(t) &= 2e^{-t} + t \end{aligned} \quad (4.6)$$

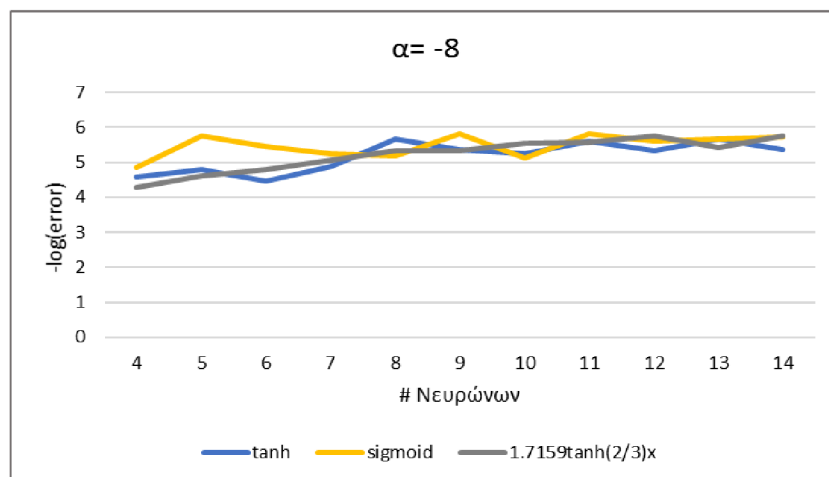
Στο Διάγραμμα 4.7 διακρίνεται η αναλυτική λύση του Π.Α.Τ. 3.



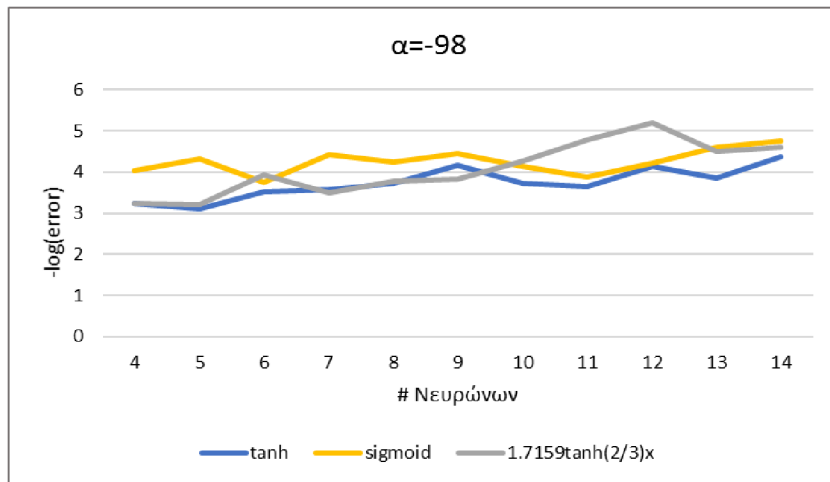
Διάγραμμα 4.7 : Αναλυτική λύση του Π.Α.Τ. 3

Από τους πίνακες Α.3- Α.6 του παραρτήματος Α, από τα αποτελέσματά του σφάλματος, του αριθμού βημάτων και το υπολογιστικό κόστος φαίνεται ότι το Π.Α.Τ. 3 είναι άκαμπτο.

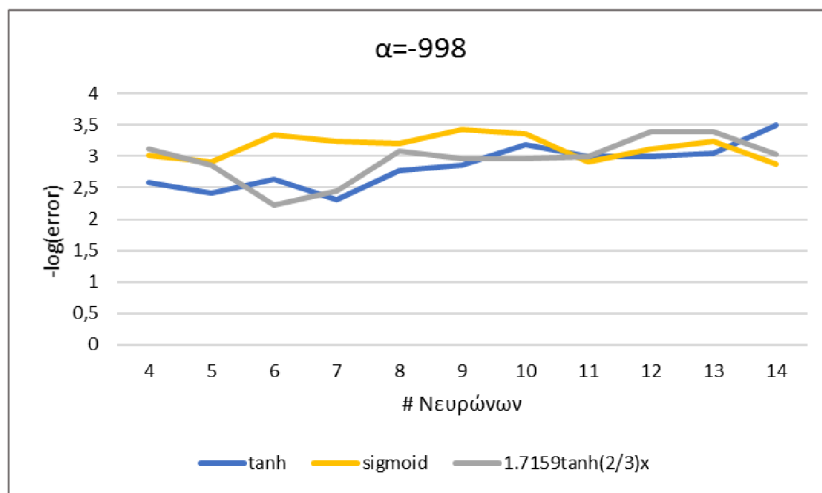
Στα Διαγράμματα 4.8-4.11 φαίνονται οι τιμές των δεκαδικών λογαρίθμων των σφαλμάτων που προκύπτουν για τις διάφορες τιμές της παραμέτρου  $\alpha$ . Οι νευρώνες οι οποίοι χρησιμοποιούνται είναι από 4 έως 14, ενώ παράλληλα γίνεται μία σύγκριση μεταξύ των τριών συναρτήσεων ενεργοποίησης ( $\tanh$ ,  $\text{sigmoid}$  &  $1.7159\tanh(4/3)bx$ ) που εξετάστηκαν. Οι πίνακες των αποτελεσμάτων παρουσιάζονται στο παράρτημα Β στους πίνακες Β.3 και Β.4.



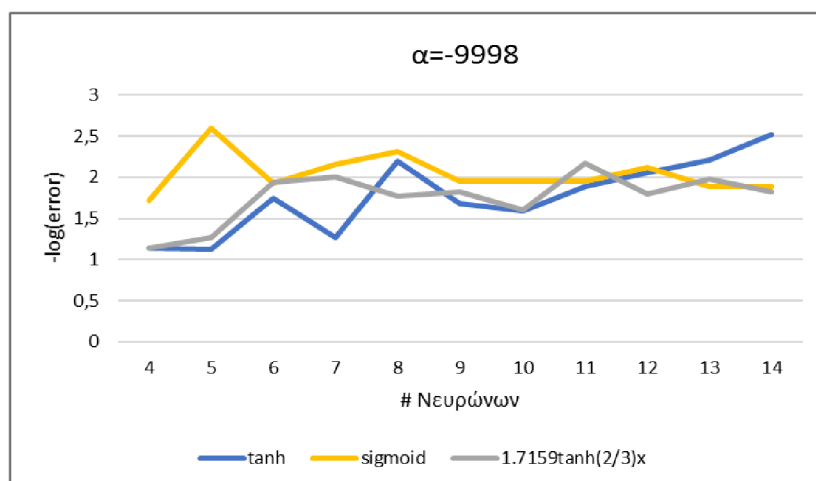
Διάγραμμα 4.8: Σφάλματα ΤΝΔ για παράμετρο  $\alpha = -8$ .



Διάγραμμα 4.9: Σφάλματα ΤΝΔ για παράμετρο  $\alpha = -98$ .



Διάγραμμα 4.10: Σφάλματα ΤΝΔ για παράμετρο  $\alpha = -998$ .

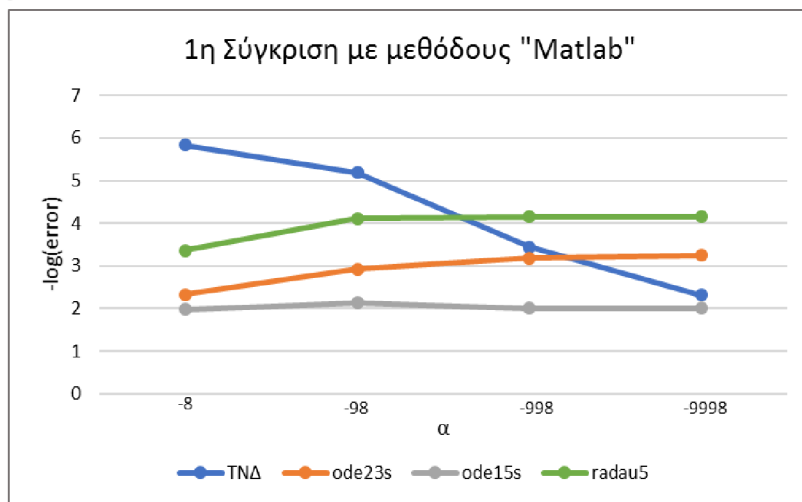


Διάγραμμα 4.11: Σφάλματα ΤΝΔ για παράμετρο  $\alpha = -9998$ .

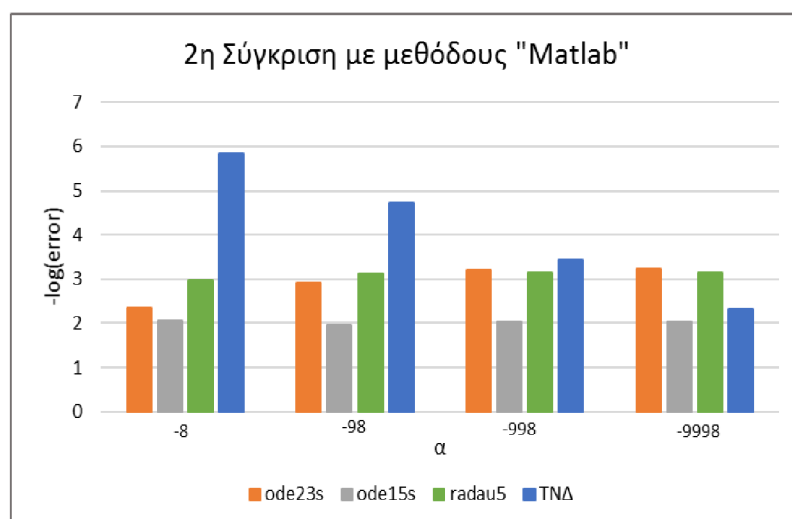
Για το Π.Α.Τ. 3 και για την τιμή της παραμέτρου 8 η καλύτερη τιμή σφάλματος του ΤΝΔ δίνεται από τη σιγμοειδή συνάρτηση ενεργοποίησης με ακρίβεια 6 δεκαδικών ψηφίων και για 9 νευρώνες, ενώ για την τιμή της παραμέτρου 98, δίνεται πάλι λύση με 6 δεκαδικά ψηφία, με συνάρτηση  $1.7159 \tanh(2/3x)$  και με 12 νευρώνες. Για  $\alpha = -998$  η καλύτερη τιμή παράγεται με τη χρήση 14 νευρώνων, με τη χρήση της  $\tanh$  και ακρίβεια 4 ψηφίων και για  $\alpha = -9998$  το ΤΝΔ επιτυγχάνει την καλύτερη τιμή, δηλαδή ακρίβεια 3 ψηφίων με τη sigmoid και 5 νευρώνες.

Όπως φαίνεται από τα διαγράμματα 4.8-4.11 και τους Πίνακες Β.3 και Β.4 η χρήση των διαφορετικών συναρτήσεων ενεργοποίησης παράγει παρόμοιας τάξης αποτελέσματα, ενώ όπως είναι αναμενόμενο, όσο αυξάνεται η τιμή της παραμέτρου και το πρόβλημα γίνεται πιο «άκαμπτο» η ακρίβεια που επιτυγχάνεται από το ΤΝΔ είναι μικρότερη.

Στα Διαγράμματα 4.12-4.13 φαίνονται δύο συγκρίσεις των δεκαδικών λογαρίθμων των σφαλμάτων που δημιουργούνται με τη χρήση της βέλτιστης τιμής του ΤΝΔ με τις μεθόδους της “Matlab” για τις τιμές της παραμέτρου  $\alpha$ . Αναλυτικά τα αποτελέσματα που προκύπτουν από τις μεθόδους Matlab φαίνονται στο παράρτημα Α, στους Πίνακες Α.3-Α.6 και Α.22.



Διάγραμμα 4.12: Σφάλματα ΤΝΔ & σφάλματα μεθόδων “Matlab”.



Στο Διάγραμμα 4.12 γίνεται σύγκριση με τις μεθόδους της “Matlab” που προτείνονται για την επίλυση stiff προβλημάτων, στα σημεία στα οποία «πατάνε» οι μέθοδοι. Τα ΤΝΔ υπερέχουν όλων των άλλων μεθόδων για τις τιμές  $\alpha = -8$  και  $\alpha = -98$ , όμως για την τιμή  $\alpha = -998$  η Radau5 δίνει καλύτερα αποτελέσματα, όπως και για την τιμή  $\alpha = -9998$ , στην οποία εκτός από την Radau5 και η Ode23s δίνει καλύτερα αποτελέσματα από τα αντίστοιχα ΤΝΔ.

Η σύγκριση στο Διάγραμμα 4.12 γίνεται για τα σημεία τα οποία εκπαιδεύονται οι μέθοδοι, ενώ στο Διάγραμμα 4.13 γίνεται σύγκριση με τις μεθόδους σε όλο το διάστημα τιμών και εδώ παρατηρείται ότι τα ΤΝΔ παράγουν καλύτερα αποτελέσματα για  $\alpha = -8$ ,  $\alpha = -98$  και  $\alpha = -998$ . Για  $\alpha = -9998$  η καλύτερη τιμή δίνεται από την Ode23s. Όπως διακρίνεται, η Radau5 δεν μπορεί να γενικεύσει καλά και έχει χάσει ψηφία από την ακρίβεια της.

Για το Π.Α.Τ. 3 τα ΤΝΔ παράγουν ικανοποιητικής ακρίβειας αποτελέσματα για μέτριας ακαμψίας προβλήματα, δηλαδή για μικρές τιμές της παραμέτρου  $\alpha$ , οι οποίες είναι καλύτερες από τις άλλες μεθόδους. Όσο αυξάνεται η τιμή της παραμέτρου η απόδοση των ΤΝΔ μειώνεται, παράγοντας ωστόσο ικανοποιητικές προσεγγιστικές λύσεις, συγκρίσιμες με τα αποτελέσματα των άλλων μεθόδων. Η χρήση των διαφορετικών συναρτήσεων ενεργοποίησης γενικά δεν δημιουργεί μεγάλες διαφοροποιήσεις.

#### **Πρόβλημα 4 :**

Ένα ακόμα Π.Α.Τ το οποίο περιέχει παράμετρο περιγράφεται στη σχέση (4.7) , ενώ η αναλυτική του λύση από τη σχέση (4.8). Οι αρχικές συνθήκες υπολογίζονται για την αρχική τιμή 0.5 και το σύστημα επιλύεται στο διάστημα τιμών  $[0,3]$ . Η παράμετρος  $\alpha$  λαμβάνει τις σταθερές τιμές -8, -98, -998,-9998. Και σε αυτό το πρόβλημα γίνεται κανονικοποίηση των τιμών του  $\alpha$  στο διάστημα τιμών  $[0,10]$ .

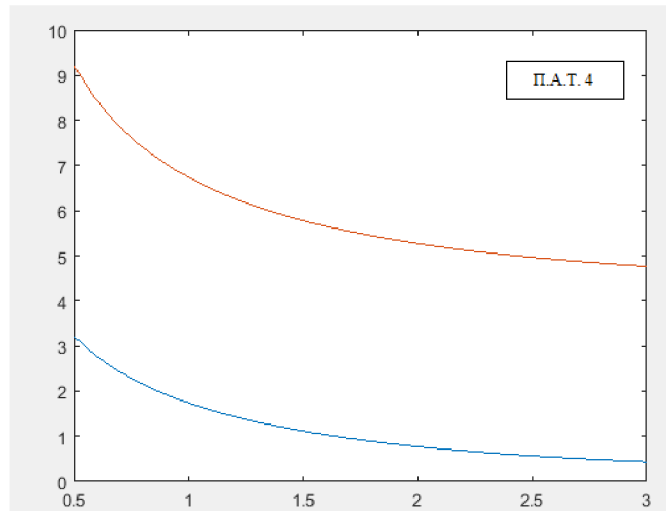
$$y'_1(t) = -2y_1(t) + y_2(t) - t^{-2} - 4, \quad y_1(0.5) = 2$$

$$y'_2(t) = -\alpha y_1(t) + (\alpha - 1)y_2(t) - 2t^{-2} + t^{-1}(2 - \alpha) + 4(1 - \alpha), \quad y_2(0.5) = 6 \quad (4.7)$$

$$y_1(t) = 2e^{-t} + t^{-1}$$

$$y_2(t) = 2e^{-t} + 2t^{-1} + 4 \quad (4.8)$$

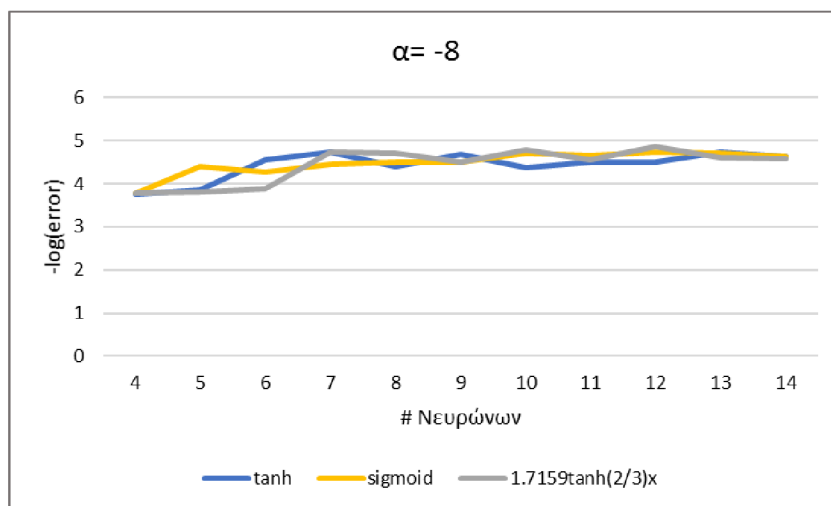
Στο Διάγραμμα 4.14 φαίνεται η αναλυτική λύση του Π.Α.Τ. 4:



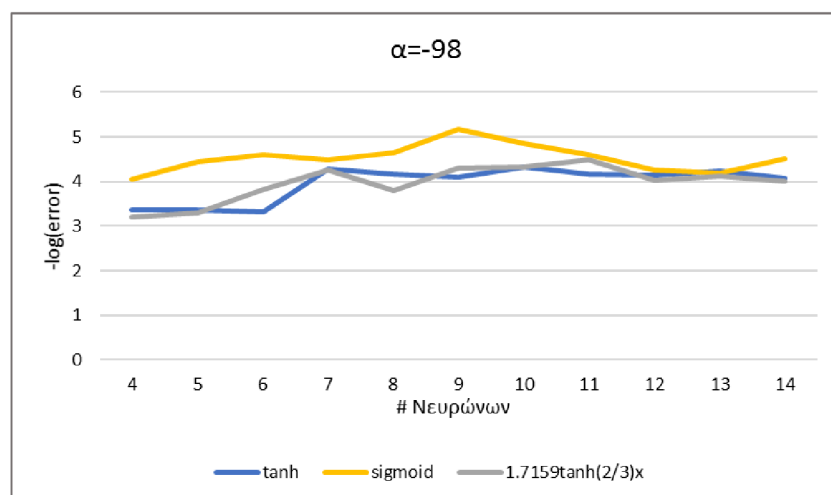
Διάγραμμα 4.14 : Αναλυτική λύση του Π.Α.Τ. 4

Από τους πίνακες Α.7- Α.10 του παραρτήματος Α, από τα αποτελέσματά του σφάλματος, του αριθμού βημάτων και το υπολογιστικό κόστος φαίνεται ότι το Π.Α.Τ. 4 είναι άκαμπτο.

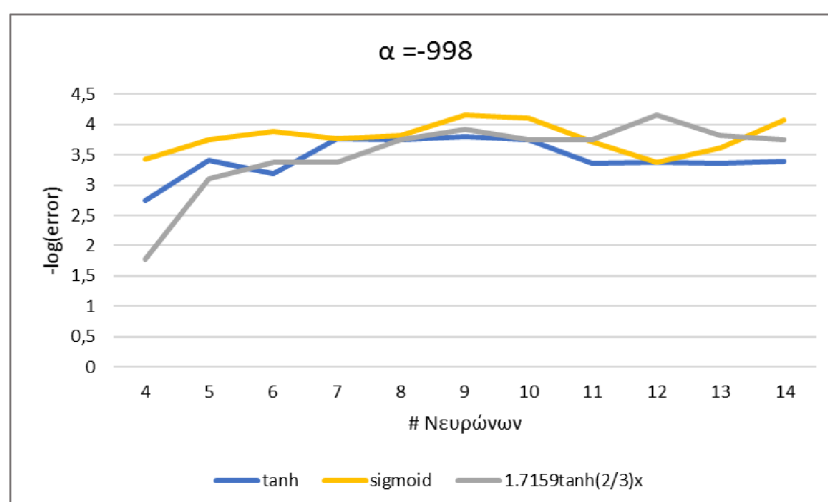
Στα Διαγράμματα 4.15-4.18 διακρίνονται οι τιμές των δεκαδικών λογαρίθμων των σφαλμάτων που προκύπτουν για τις διάφορες τιμές της παραμέτρου  $\alpha$ . Παρουσιάζονται τα αποτελέσματα για τρεις συναρτήσεις ενεργοποίησης ( $\tanh$ , sigmoid &  $1.7159\tanh(2/3)x$ ) και οι νευρώνες που χρησιμοποιούνται είναι από 4 έως 14. Οι πίνακες των αποτελεσμάτων παρουσιάζονται στο παράρτημα Β στους πίνακες Β.5 και Β.6.



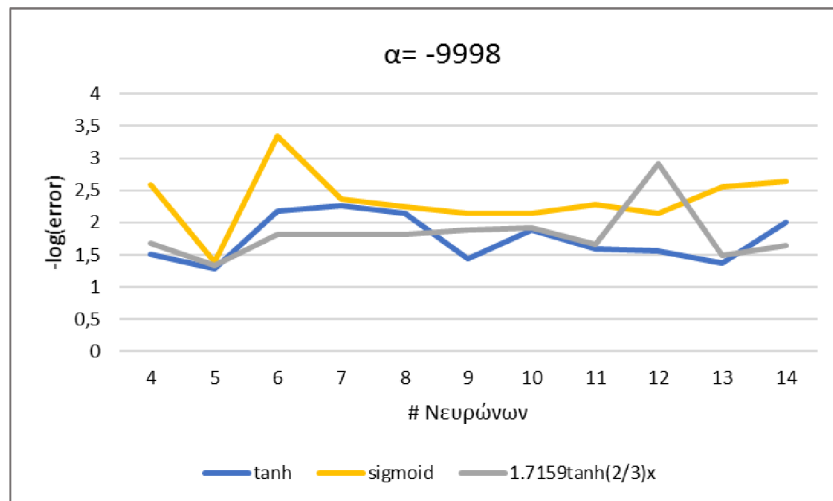
Διάγραμμα 4.15: Σφάλματα ΤΝΔ για παράμετρο  $\alpha = -8$ .



Διάγραμμα 4.16: Σφάλματα ΤΝΔ για παράμετρο  $\alpha = -98$ .



Διάγραμμα 4.17: Σφάλματα ΤΝΔ για παράμετρο  $\alpha = -998$ .

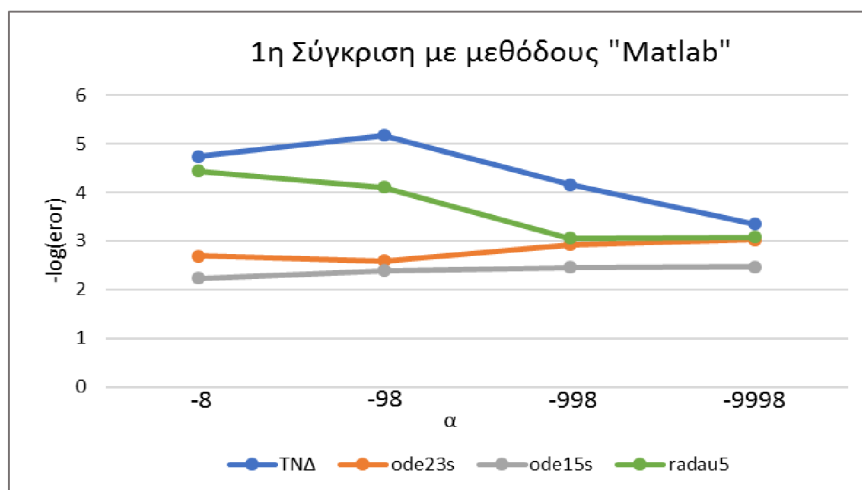


Διάγραμμα 4.18: Σφάλματα ΤΝΔ για παράμετρο α = -9998.

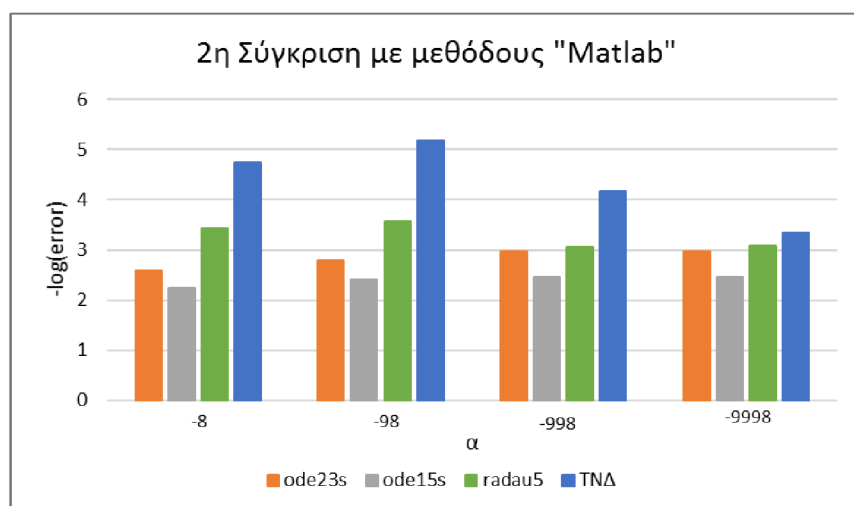
Για το Π.Α.Τ. 4 η καλύτερη τιμή του ΤΝΔ, για την παράμετρο α=-8, προκύπτει με τη χρήση 12 νευρώνων και τη συνάρτηση  $1.7159 \tanh((2/3)x)$  και παράγει ακρίβεια είναι 5 ψηφίων. Για το α=-98, η σιγμοειδής συνάρτηση ενεργοποίησης συμπεριφέρεται καλύτερα από τις άλλες συναρτήσεις και δίνει αποτέλεσμα με ακρίβεια 6 δεκαδικών ψηφίων. Για το α=-998, παρόλο που η  $1.7159 \tanh((2/3)x)$  έχει το καλύτερο αποτέλεσμα, η συμπεριφορά της sigmoid είναι αρκετά καλύτερη, όπως και για α=-9998 η sigmoid δίνει το καλύτερο αποτέλεσμα.

Στα Διαγράμματα 4.19-4.20 γίνεται μια σύγκριση των δεκαδικών λογαρίθμων των σφαλμάτων που δημιουργούνται με τη χρήση της βέλτιστης τιμής των ΤΝΔ για κάθε τιμή της παραμέτρου α με τις μεθόδους της “Matlab”.





Διάγραμμα 4.19: Σφάλματα TNA & σφάλματα μεθόδων “Matlab”



Διάγραμμα 4.20: Σφάλματα TNA & σφάλματα μεθόδων “Matlab”

Στα Διαγράμματα 4.19 και 4.20 πραγματοποιείται σύγκριση των TNA με μεθόδους της “Matlab”, στα οποία φαίνεται ότι τα αποτελέσματα των TNA υπερτερούν των άλλων μεθόδων, είτε η σύγκριση γίνεται στα σημεία στα οποία «πατάνε» οι μέθοδοι της “Matlab” είτε σε όλο το διάστημα τιμών. Τα αποτελέσματα φαίνονται στους πίνακες A.7-A.10 και A.23.

Γενικά στο Π.Α.Τ. 4 η σιγμοειδής συνάρτηση ενεργοποίησης έχει καλύτερη συμπεριφορά από τις υπόλοιπες συναρτήσεις ενεργοποίησης και ακόμα και για μικρό αριθμό παραμέτρων τα αποτελέσματα τα οποία παράγονται δεν απέχουν αρκετά από τα αποτελέσματα που παράγονται για μεγαλύτερο αριθμό παραμέτρων. Για το Π.Α.Τ. 4 τα T.N.Δ που εκπαιδεύθηκαν έχουν καλύτερα αποτελέσματα από τις μεθόδους με τις οποίες συγκρίνονται.

### Πρόβλημα 5:

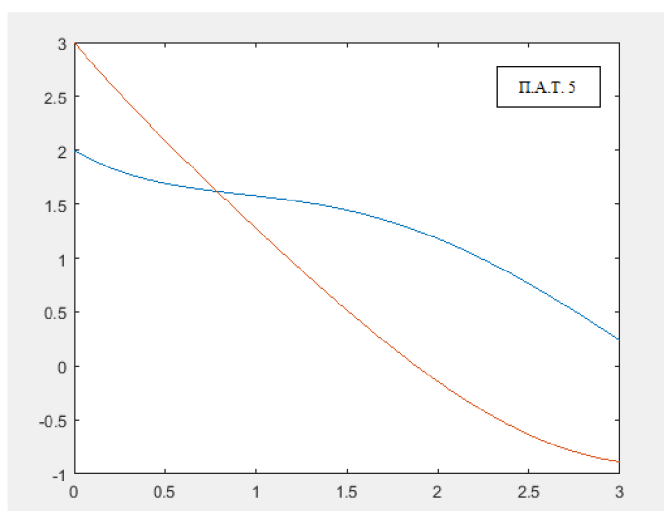
Το επόμενο άκαμπτο παραμετρικοποιημένο πρόβλημα που επιλύθηκε είναι και αυτό ένα σύστημα ΣΔΕ 2x2, το οποίο περιγράφεται από τη σχέση (4.9) με αναλυτική λύση που φαίνεται στη σχέση (4.10). Οι αρχικές συνθήκες υπολογίζονται στο 0 και το διάστημα τιμών στο οποίο επιλύεται είναι το [0 3]. Η παράμετρος α λαμβάνει τις σταθερές τιμές -8, -98, -998, -9998, ενώ γίνεται κανονικοποίηση των τιμών του α στο διάστημα τιμών [0, 10].

$$\begin{aligned} y_1'(t) &= -2y_1(t) + y_2(t) + 2\sin(t), & y_1(0) &= 2 \\ y_2'(t) &= -\alpha y_1(t) + (\alpha - 1)y_2(t) - (\alpha - 1)(\cos(t) - \sin(t)), & y_2(0) &= 3 \end{aligned} \quad (4.9)$$

$$y_1(t) = 2e^{-t} + \sin(t)$$

$$y_2(t) = 2e^{-t} + \cos(t) \quad (4.10)$$

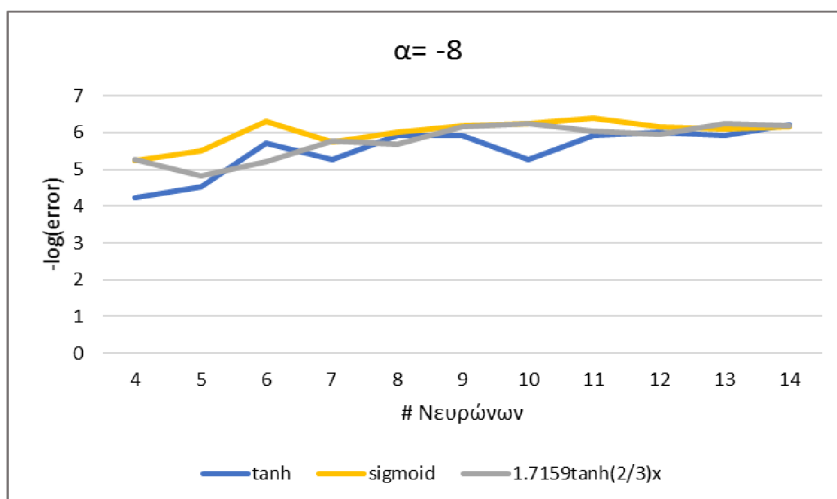
Στο Διάγραμμα 4.21 παρουσιάζεται η αναλυτική λύση του Π.Α.Τ. 5



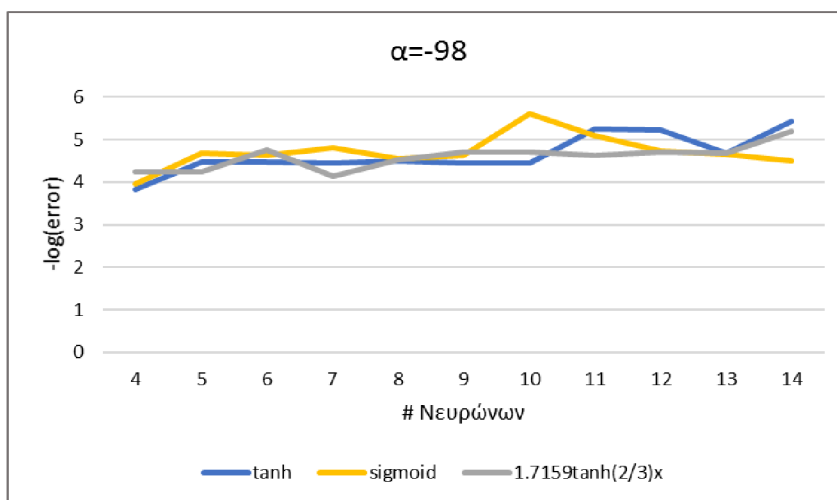
Διάγραμμα 4.21: Αναλυτική λύση Π.Α.Τ. 5

Από τους Πίνακες Α.11-Α.14 του παραρτήματος Α, από τα αποτελέσματά του σφάλματος, του αριθμού βημάτων και το υπολογιστικό κόστος φαίνεται ότι το Π.Α.Τ. 5 είναι άκαμπτο.

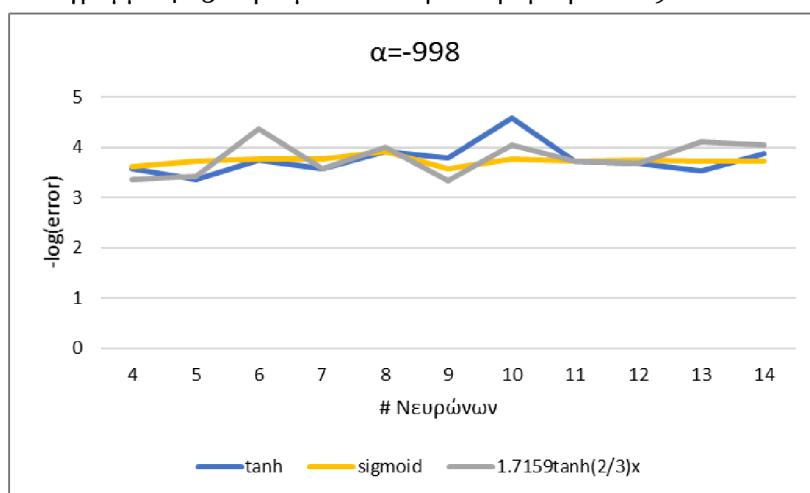
Στα Διαγράμματα 4.22-4.25 απεικονίζονται οι δεκαδικοί λογάριθμοι των σφαλμάτων των ΤΝΔ για την επίλυση του Π.Α.Τ. 5 σε σχέση με τον αριθμό νευρώνων που χρησιμοποιούνται, συγκρίνοντας τρεις διαφορετικές συναρτήσεις ενεργοποίησης. Τα αποτελέσματα που παρουσιάζονται στα διαγράμματα φαίνονται αναλυτικά στο Παράρτημα Β, στους πίνακες Β.7 και Β.8.



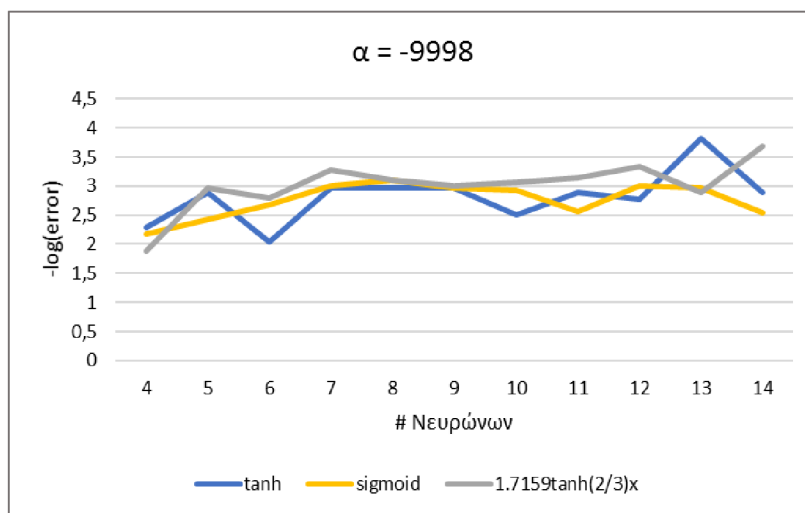
Διάγραμμα 4.22: Σφάλματα ΤΝΔ για παράμετρο  $\alpha = -8$ .



Διάγραμμα 4.23: Σφάλματα ΤΝΔ για παράμετρο  $\alpha = -98$ .



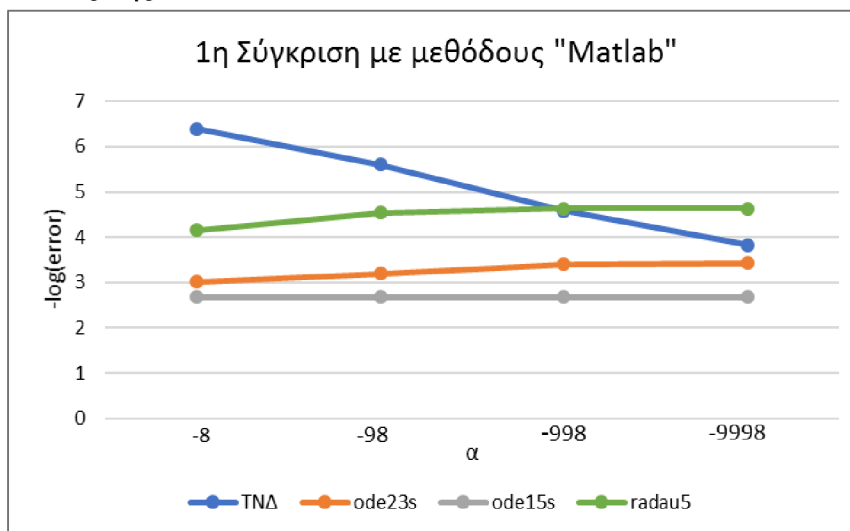
Διάγραμμα 4.24: Σφάλματα ΤΝΔ για παράμετρο  $\alpha = -998$ .



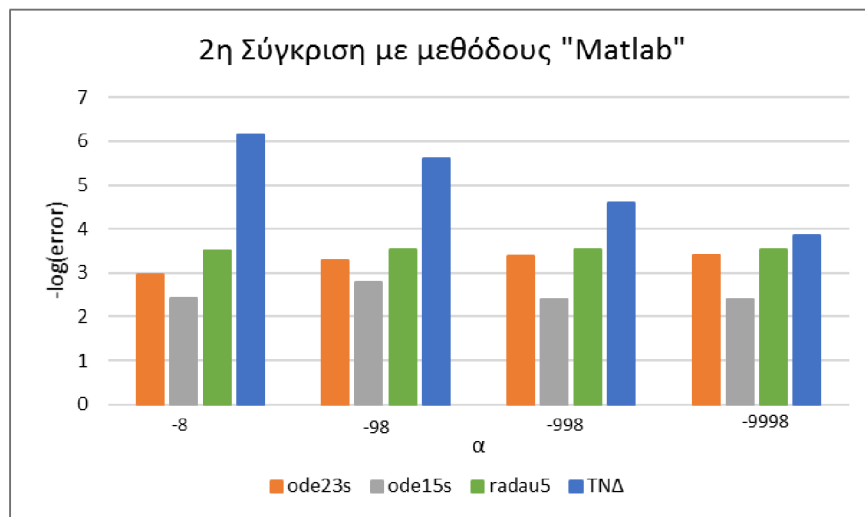
Διάγραμμα 4.25: Σφάλματα ΤΝΔ για παράμετρο α = -9998.

Για το Π.Α.Τ. 5 η καλύτερη τιμή του ΤΝΔ, για την παράμετρο α=-8, προκύπτει με τη χρήση 11 νευρώνων και τη σιγμοειδή συνάρτηση και παράγει ακρίβεια 7 ψηφίων. Για το α=-98, η σιγμοειδής συνάρτηση ενεργοποίησης συμπεριφέρεται καλύτερα από τις άλλες συναρτήσεις και δίνει αποτέλεσμα με ακρίβεια 6 δεκαδικών ψηφίων και χρήση 10 νευρώνων. Για το α=-998, η tanh έχει το καλύτερο αποτέλεσμα, με ακρίβεια 6 ψηφίων και τη χρήση 10 νευρώνων και για α=-9998 η tanh δίνει το καλύτερο αποτέλεσμα με τη χρήση 13 νευρώνων.

Στο Διάγραμμα 4.26-4.27 διακρίνονται οι βέλτιστες τιμές των δεκαδικών λογαρίθμων των σφαλμάτων του ΤΝΔ για τις τιμές της παραμέτρου α που προαναφέρθηκαν, σε σύγκριση με τις τιμές που προκύπτουν κατά την επίλυση του Π.Α.Τ με τις μεθόδους της "Matlab".



Διάγραμμα 4.26: Σφάλματα ΤΝΔ & σφάλματα μεθόδων "Matlab"



Διάγραμμα 4.27: Σφάλματα TND & σφάλματα μεθόδων "Matlab"

Όσον αφορά τη σύγκριση που πραγματοποιείται στα Διαγράμματα 4.26 και 4.27 και φαίνεται αναλυτικά στα Α.11–Α.14 και στο Α.24 παρατηρείται ότι όταν γίνεται η σύγκριση στα σημεία στα οποία «πατάνε» οι υπόλοιπες μέθοδοι, τα TND υπερτερούν όλων των μεθόδων για τιμές της παραμέτρου  $\alpha = -8$  και  $\alpha = -98$ , ενώ για τις τιμές  $\alpha = -998$  και  $\alpha = -9998$  η Radau5 εμφανίζει καλύτερα αποτελέσματα. Όμως όταν η σύγκριση πραγματοποιείται σε όλο το διάστημα τιμών, η Radau5 χάνει ψηφία ακρίβειας, με αποτέλεσμα τα TND να υπερτερούν για όλες τις τιμές της παραμέτρου  $\alpha$ .

Γενικά παρόλο που η tanh παράγει στις περισσότερες περιπτώσεις τα καλύτερα αποτελέσματα, η χρήση και των υπόλοιπων συναρτήσεων ενεργοποίησης παράγει εξίσου καλά αποτελέσματα, ακόμα και σε αριθμό νευρώνων αρκετά μικρό. Επιπλέον τα TND τα οποία έχουν εκπαιδευθεί για τις διάφορες τιμές της παραμέτρου παράγουν καλύτερες λύσεις από τις συγκρινόμενες μεθόδους σε όλο το διάστημα επίλυσης.

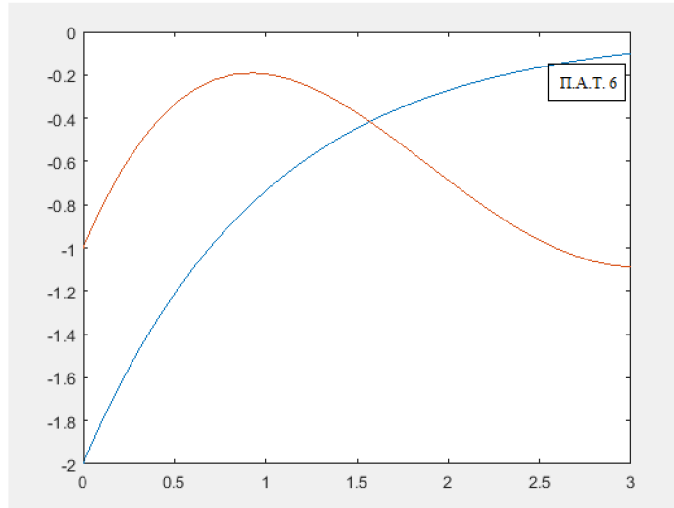
### Πρόβλημα 6:

Το τελευταίο πρόβλημα που επιλύθηκε με ύπαρξη σταθερής παραμέτρου στο σύστημα εξισώσεων περιγράφεται από τη σχέση (4.11), ενώ η αναλυτική λύση δίνεται από τη σχέση (4.12). Οι αρχικές συνθήκες υπολογίστηκαν στο 0 και η παράμετρος  $\alpha$  λαμβάνει τις τιμές -8, -98, -998, -9998 και κανονικοποιείται στο διάστημα τιμών [0,10].

$$\begin{aligned}
 y_1'(t) &= -2y_1(t) + y_2(t) - \cos(t) , & y_1(0) &= 2 \\
 y_2'(t) &= -\alpha y_1(t) + (\alpha - 1)y_2(t) - \sin(t) - \alpha \cos(t) + \cos(t) , & y_2(0) &= 3
 \end{aligned} \tag{4.11}$$

$$\begin{aligned}
 y_1(t) &= 2e^{-t} \\
 y_2(t) &= 2e^{-t} + \cos(t) \quad (4.12)
 \end{aligned}$$

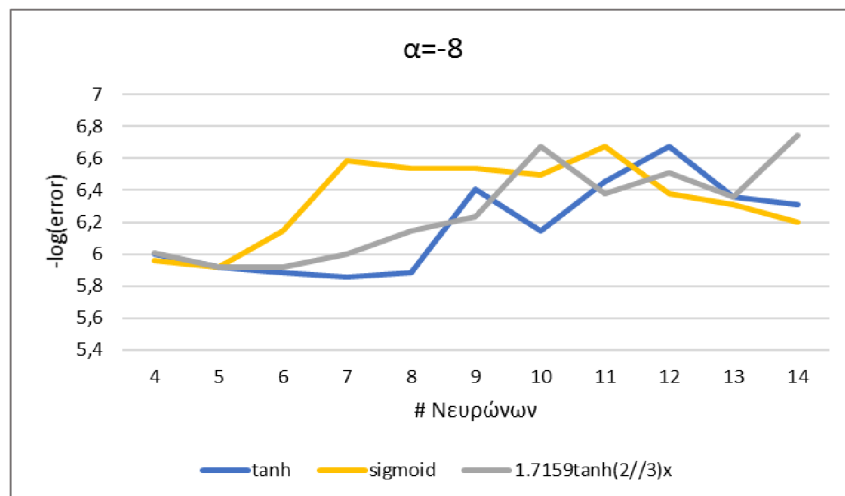
Στο Διάγραμμα 4.28 διακρίνεται η αναλυτική λύση του Π.Α.Τ. 6:



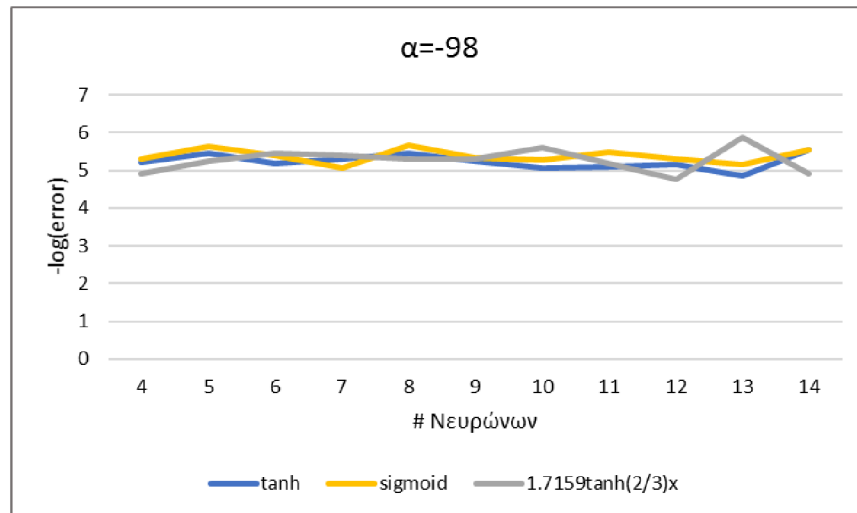
Διάγραμμα 4.28: Αναλυτική λύση Π.Α.Τ. 6

Από τους Πίνακες Α.15-Α.18 του παραρτήματος Α, από τα αποτελέσματά του σφάλματος, του αριθμού βημάτων και το υπολογιστικό κόστος φαίνεται ότι το Π.Α.Τ. 6 είναι άκαμπτο.

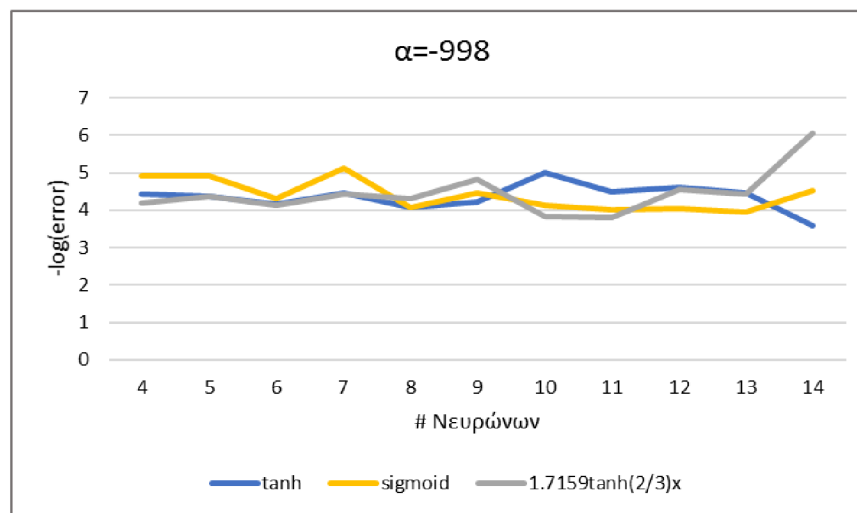
Στα Διαγράμματα 4.29 – 4.32 παρουσιάζονται οι δεκαδικοί λογάριθμοι των σφαλμάτων των ΤΝΔ για την επίλυση του Π.Α.Τ. 6 σε σχέση με τον αριθμό νευρώνων που χρησιμοποιούνται, συγκρίνοντας τρεις διαφορετικές συναρτήσεις ενεργοποίησης. Τα αποτελέσματα φαίνονται αναλυτικά στους πίνακες Β.9 και Β.10 του παραρτήματος Β



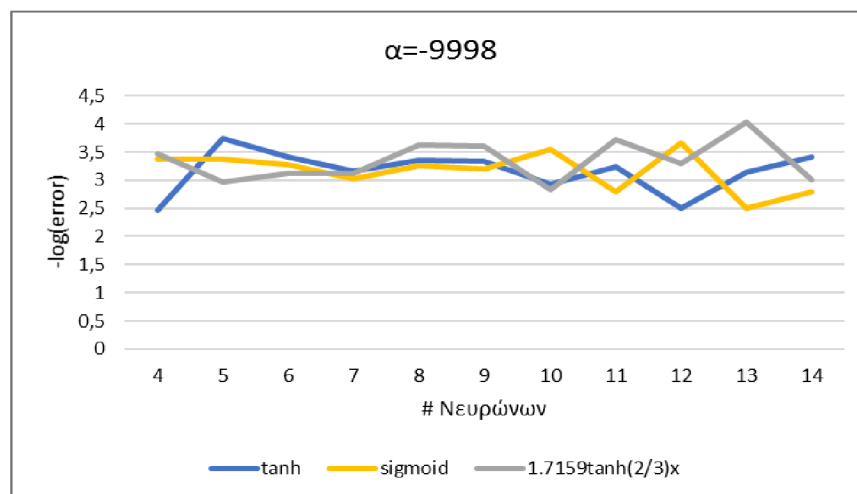
Διάγραμμα 4.29: Σφάλματα ΤΝΔ για παράμετρο  $\alpha = -8$ .



Διάγραμμα 4.30: Σφάλματα ΤΝΔ για παράμετρο  $\alpha = -98$ .



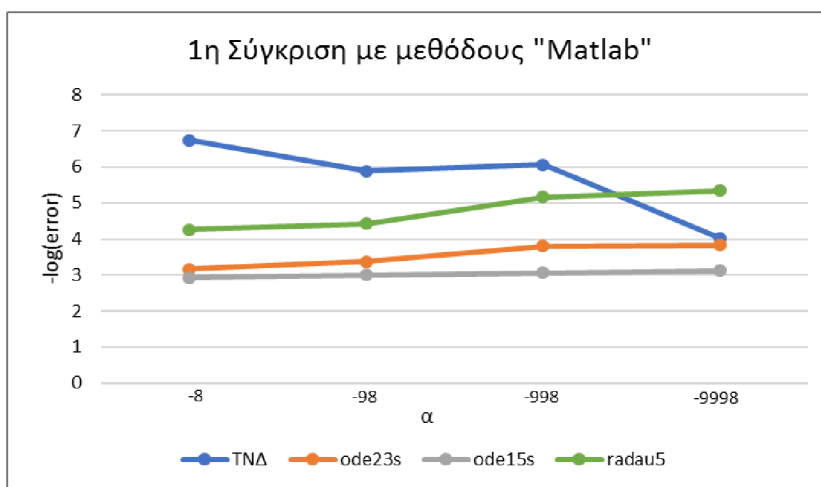
Διάγραμμα 4.31: Σφάλματα ΤΝΔ για παράμετρο  $\alpha = -998$ .



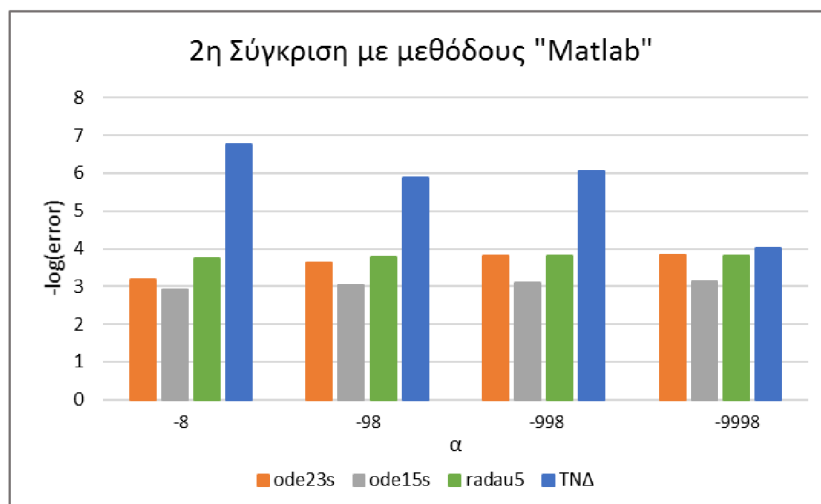
Διάγραμμα 4.32: Σφάλματα ΤΝΔ για παράμετρο  $\alpha = -9998$ .

Για το Π.Α.Τ. 6 οι καλύτερες τιμές του ΤΝΔ, παράγονται για όλες τις παραμέτρους με τη χρήση της  $1.7159 \tanh((2/3)x)$  συνάρτησης ενεργοποίησης και τη χρήση 13 ή 14 νευρώνων, όπως φαίνεται από τα Διαγράμματα 4.29-4.32 και τους πίνακες Β.9 και Β.10. Και σε αυτό το πρόβλημα όμως παρατηρείται η εξαγωγή αξιολογών αποτελεσμάτων και με τη χρήση λιγότερων νευρώνων, ενώ οι διάφορες συναρτήσεις ενεργοποίησης δεν διαφοροποιούν υπερβολικά το αποτέλεσμα.

Στα Διαγράμματα 4.33-4.34 παρουσιάζεται μια σύγκριση των βέλτιστων τιμών των δεκαδικών λογαρίθμων των σφαλμάτων του ΤΝΔ για τις τιμές της παραμέτρου  $\alpha$  που προαναφέρθηκαν, σε σύγκριση με τις τιμές που προκύπτουν κατά την επίλυση του Π.Α.Τ με τις μεθόδους της “Matlab”.



Διάγραμμα 4.33 : Σφάλματα ΤΝΔ & σφάλματα μεθόδων “Matlab”



Διάγραμμα 4.34 : Σφάλματα ΤΝΔ & σφάλματα μεθόδων “Matlab”



Στα Διαγράμματα 4.33-4.34 και στους πίνακες Α.15-Α.18 και Α.26 διακρίνεται ξανά το χαρακτηριστικό που προκύπτει από τις συγκρίσεις των μεθόδων στα προηγούμενα προβλήματα. Στην πρώτη σύγκριση τα ΤΝΔ υπερτερούν των άλλων μεθόδων, εκτός της τιμής  $a=-9998$  στην οποία είναι καλύτερη η Radau5, γεγονός το οποίο όμως στη 2<sup>η</sup> σύγκριση ανατρέπεται, καθώς η Radau5 χάνει σε ακρίβεια και τα ΤΝΔ είναι η καλύτερη μέθοδος για όλες τις τιμές της παραμέτρου  $a$ .

Για το Π.Α.Τ.6 η χρήση διαφορετικών συναρτήσεων ενεργοποίησης παράγει παρόμοια αποτελέσματα. Επίσης τα ΤΝΔ τα οποία έχουν εκπαιδευθεί για τις διάφορες τιμές της παραμέτρου, παράγουν καλύτερα αποτελέσματα σε σύγκριση με τις άλλες μεθόδους, όταν η σύγκριση γίνεται στο σύνολο του διαστήματος επίλυσης .

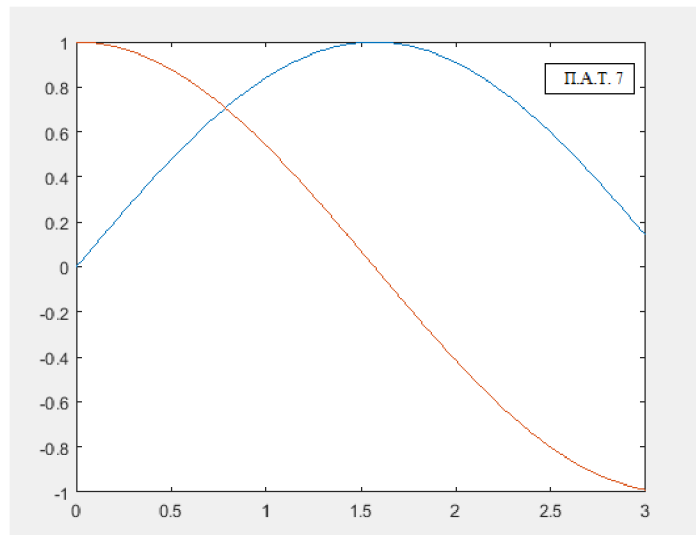
### **Πρόβλημα 7**

Το πρόβλημα 7 έχει μία βασική διαφορά από τα υπόλοιπα προβλήματα, η οποία είναι στο ότι υπάρχει παράμετρος στις αρχικές συνθήκες. Το Π.Α.Τ. 7 είναι ένα άκαμπτο σύστημα ΣΔΕ το οποίο περιγράφεται στη σχέση (4.13) και περιέχει την παράμετρο  $a$  τόσο στην αναλυτική λύση που φαίνεται στη σχέση (4.14) όσο και στις αρχικές συνθήκες. Το διάστημα τιμών στο οποίο επιλύεται το πρόβλημα είναι το  $[0,3]$  και η παράμετρος  $a$  λαμβάνει τις τιμές 0, 1, 2.

$$\begin{aligned} \begin{bmatrix} y_1'(t) \\ y_2'(t) \end{bmatrix} &= \begin{bmatrix} -2 & 1 \\ 998 & -999 \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} + \begin{bmatrix} 2\sin(t) \\ 999(\cos(t) - \sin(t)) \end{bmatrix}, \\ \begin{bmatrix} y_1(0) \\ y_2(0) \end{bmatrix} &= \begin{bmatrix} a \\ a + 1 \end{bmatrix} \end{aligned} \quad (4.13)$$

$$\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = ae^{-t} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} \sin(t) \\ \cos(t) \end{bmatrix} \quad (4.14)$$

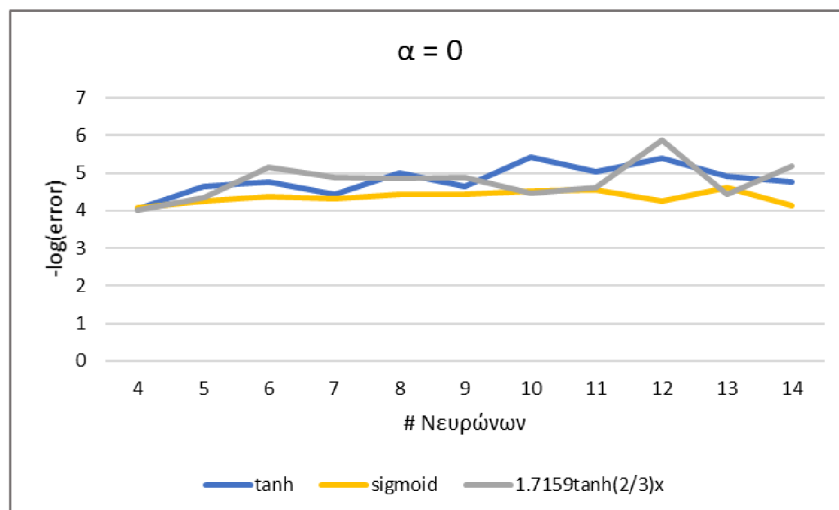
Στο Διάγραμμα 4.35 φαίνεται η αναλυτική λύση του Π.Α.Τ. 7 :



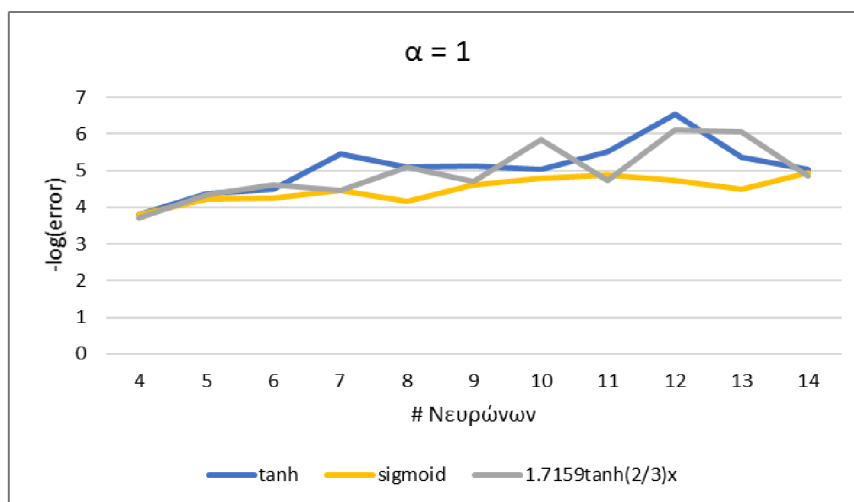
Διάγραμμα 4.35 : Αναλυτική λύση του Π.Α.Τ. 7

Από τους Πίνακες Α.19-Α.21 του παραρτήματος Α, από τα αποτελέσματά του σφάλματος, του αριθμού βημάτων και το υπολογιστικό κόστος φαίνεται ότι το Π.Α.Τ. 7 είναι άκαμπτο.

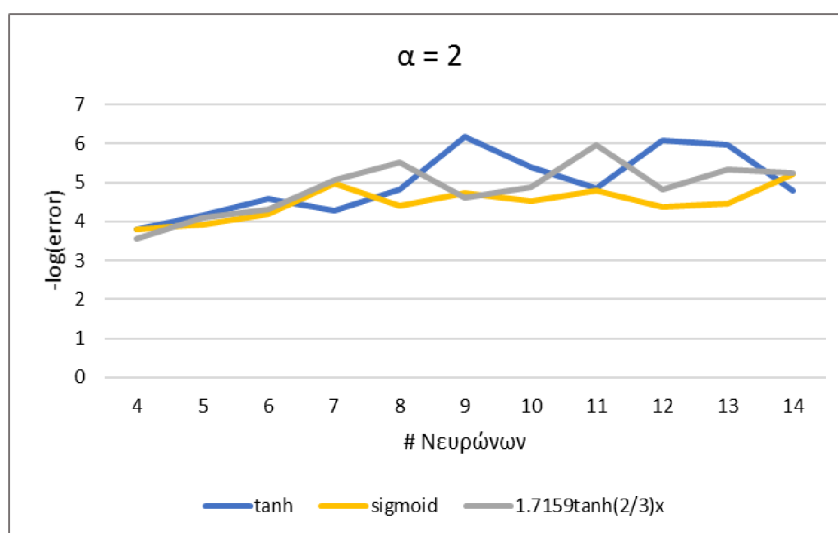
Στα Διαγράμματα 4.36 – 4.38 παρουσιάζονται οι δεκαδικοί λογάριθμοι των σφαλμάτων των ΤΝΔ για την επίλυση του Π.Α.Τ. 7 σε σχέση με τον αριθμό νευρώνων που χρησιμοποιούνται, συγκρίνοντας τρεις διαφορετικές συναρτήσεις ενεργοποίησης. Τα αποτελέσματα που φαίνονται στα Διαγράμματα, παρουσιάζονται αναλυτικά στους πίνακες Β.11 και Β.12 του παραρτήματος Β.



Διάγραμμα 4.35: Σφάλματα ΤΝΔ για παράμετρο  $\alpha=0$ .



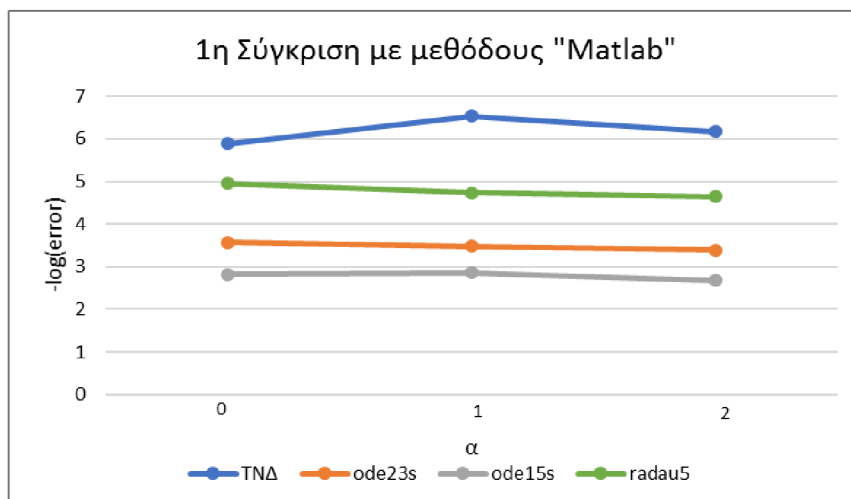
Διάγραμμα 4.36: Σφάλματα ΤΝΔ για παράμετρο  $\alpha = 1$ .



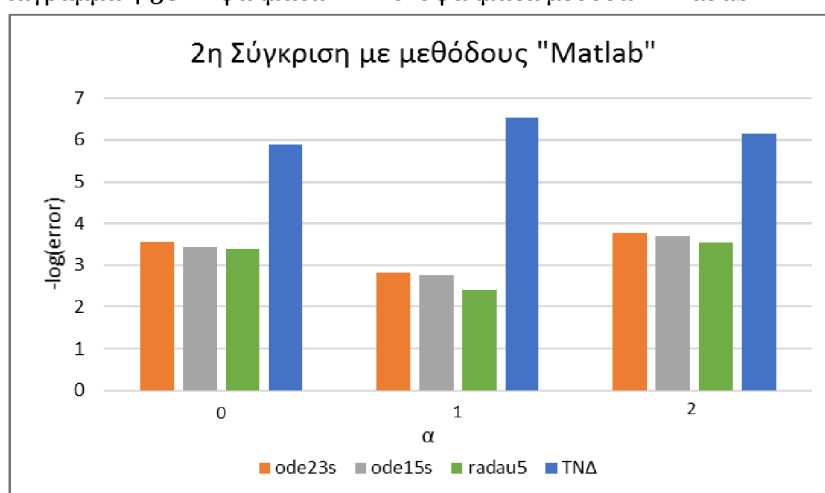
Διάγραμμα 4.37: Σφάλματα ΤΝΔ για παράμετρο  $\alpha = 2$ .

Για τιμή της παραμέτρου  $\alpha=0$  η καλύτερη τιμή από το ΤΝΔ λαμβάνεται για 12 νευρώνες από την  $1.7159 \tanh((2/3)x)$ . Για την τιμή  $\alpha=1$  η βέλτιστη τιμή επιτυγχάνεται με τη χρήση της  $\tanh$  και 12 νευρώνων, όπως και για την τιμή  $\alpha=2$  η  $\tanh$  δίνει το καλύτερο αποτέλεσμα με 9 νευρώνες. Όπως προκύπτει από τα Διαγράμματα 4.35-4.37 και τους Πίνακες Β.11-Β.12 στο Π.Α.Τ. 7 η συνάρτηση ενεργοποίησης  $\tanh$  παρουσιάζει καλύτερη συμπεριφορά, ωστόσο η διαφορά από τις υπόλοιπες συναρτήσεις δεν είναι έντονη. Επιπλέον ακόμα και με τη χρήση λίγων παραμέτρων, δηλαδή μικρού αριθμού νευρώνων, παράγονται πολύ καλά αποτελέσματα.

Στα Διαγράμματα 4.38-4.39 και στους Πίνακες Α.19-Α.21 και Α.27 παρουσιάζεται μια σύγκριση των βέλτιστων τιμών των δεκαδικών λογαρίθμων των σφαλμάτων του ΤΝΔ για τις τιμές της παραμέτρου  $\alpha$  που προαναφέρθηκαν, σε σύγκριση με τις τιμές που προκύπτουν κατά την επίλυση του Π.Α.Τ 7 με τις μεθόδους της “Matlab”.



Διάγραμμα 4.38 : Σφάλματα TNΔ & σφάλματα μεθόδων “Matlab”



Διάγραμμα 4.39 : Σφάλματα TNΔ & σφάλματα μεθόδων “Matlab”

Στα Διαγράμματα 4.38-4.39 και στους Πίνακες A.19-A.21 και A.27 που πραγματοποιούνται οι συγκρίσεις με τις μεθόδους της Matlab, τα TNΔ για τις διάφορες τιμές της παραμέτρου  $\alpha$  είναι καλύτερα για όλες τις τιμές της παραμέτρου, είτε η σύγκριση γίνεται στα σημεία στα οποία «πατάνε» οι άλλες μέθοδοι, είτε στο ευρύτερο διάστημα επίλυσης.

Στο Π.Α.Τ. 7 επομένως παρατηρείται η εξαγωγή αποτελεσμάτων ικανοποιητικής ακρίβειας με τη χρήση οποιασδήποτε συναρτήσεως ενεργοποίησης, ενώ τα TNΔ που εκπαιδεύθηκαν έχουν καλύτερα σφάλματα από τις υπόλοιπες προσεγγιστικές μεθόδους που χρησιμοποιήθηκαν.

Τα αποτελέσματα των άκαμπτων ΣΔΕ με τη χρήση σταθερής παραμέτρου  $\alpha$  παρουσιάστηκαν στο επιστημονικό συνέδριο «ICNAAM 2019» και δημοσιεύθηκαν στο περιοδικό του συνεδρίου [19].

- Στα επόμενα προβλήματα πραγματοποιείται μία γενίκευση της μεθόδου και επιλύονται προβλήματα με διάστημα τιμών των παραμέτρων. Στη συγκεκριμένη περίπτωση το ΤΝΔ εκπαιδεύεται σε συγκεκριμένες τιμές της παραμέτρου και καλείται να γενικεύσει σε άλλες τιμές μέσα στο διάστημα αυτό.

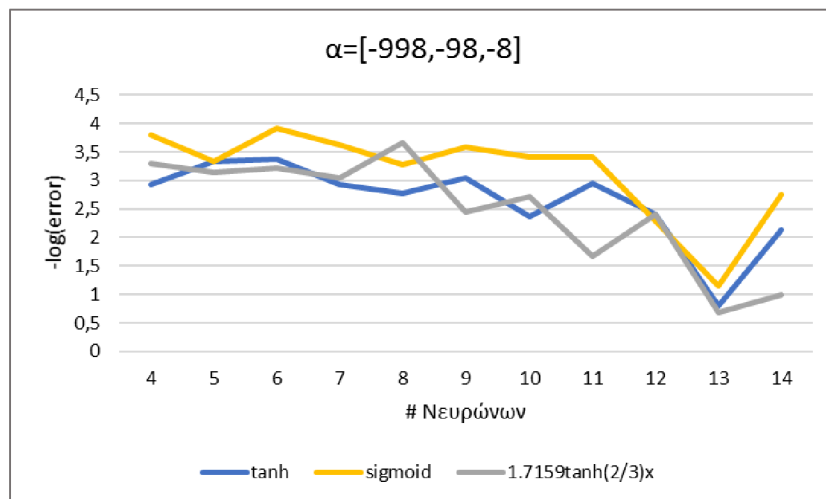
### **4.3 Άκαμπτα προβλήματα με οικογένεια παραμέτρων**

#### **Πρόβλημα 8**

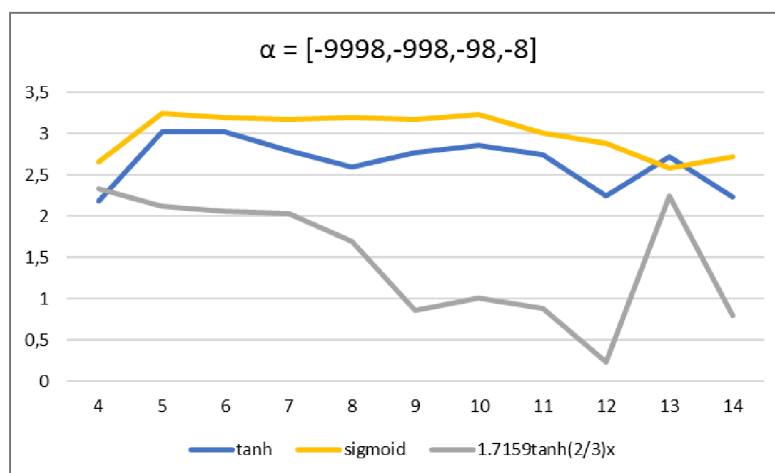
Το πρώτο Π.Α.Τ. το οποίο επιλύθηκε με τη χρήση οικογένειας παραμέτρων περιγράφεται από τις σχέσεις (4.5) και (4.6). Το διάστημα τιμών στο οποίο επιλύεται το πρόβλημα είναι το  $[0,3]$ , ενώ η παράμετρος  $a$  λαμβάνει δύο διαστήματα τιμών, το  $[-998,-98,-8]$  και το  $[-9998,-998,-98,-8]$ . Για τις τιμές της παραμέτρου  $a$ , έχει γίνει κανονικοποίηση των δεδομένων σε διάστημα τιμών  $[0,10]$  πριν την εκπαίδευση του νευρωνικού.

Η αναλυτική λύση διακρίνεται στο Διάγραμμα 4.7, ενώ από τους Πίνακες Α.3 -Α.6 του Παραρτήματος Α και από τα αποτελέσματά του σφάλματος, του αριθμού βημάτων και το υπολογιστικό κόστος φαίνεται ότι το Π.Α.Τ. 8 είναι άκαμπτο.

Τα αποτελέσματα του προβλήματος αυτού παρουσιάζονται αναλυτικά στους πίνακες Β.3 και Β.4 του παραρτήματος Β, ενώ στα Διαγράμματα 4.40-4.41 φαίνονται οι τιμές των δεκαδικών λογαρίθμων των σφαλμάτων που προκύπτουν για τα διαστήματα τιμών της παραμέτρου  $a$  και μια σύγκριση των τριών συναρτήσεων ενεργοποίησης που χρησιμοποιήθηκαν . Οι νευρώνες οι οποίοι χρησιμοποιούνται είναι από 4 έως 14.



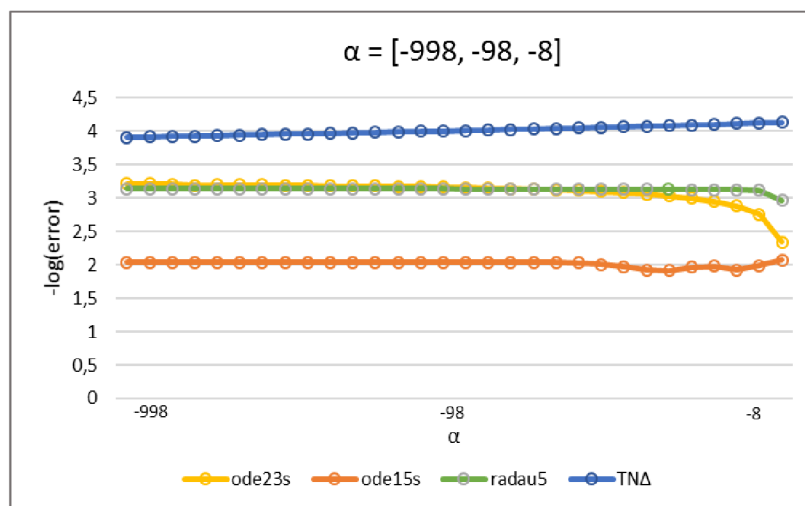
Διάγραμμα 4.40: Σφάλματα ΤΝΔ για παράμετρο  $\alpha = [-998, -98, -8]$ .



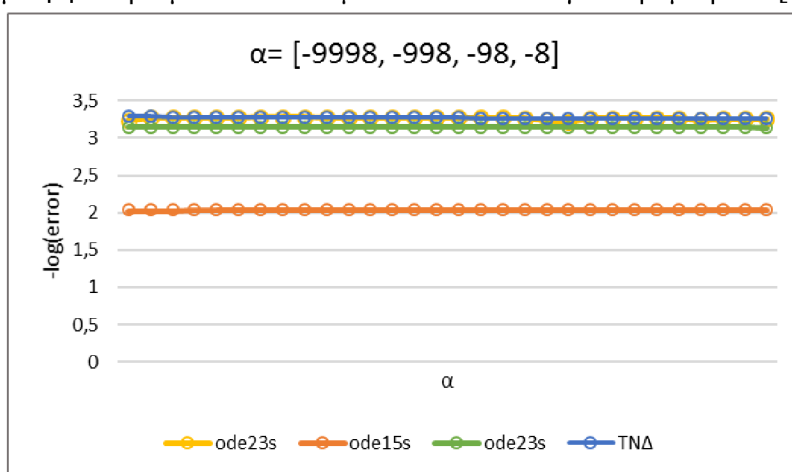
Διάγραμμα 4.41: Σφάλματα ΤΝΔ για παράμετρο  $\alpha = [-9998, -998, -98, -8]$ .

Για το Π.Α.Τ. 8, όπως διακρίνεται στα Διαγράμματα 4.40-4.41 και στους Πίνακες Β.3 και Β.4 και για τα δύο διαστήματα τιμών που επιλύεται το πρόβλημα, η συνάρτηση ενεργοποίησης που παράγει τα καλύτερα αποτελέσματα είναι η σιγμοειδής, ενώ ο αριθμός των νευρώνων που χρησιμοποιούνται είναι μικρός, 6 και 5 νευρώνες αντίστοιχα. Ο αριθμός των παραμέτρων ωστόσο είναι περίπου ίδιος με τον αριθμό των παραμέτρων που χρησιμοποιήθηκαν για την επίλυση του Π.Α.Τ. 3, λόγω του αριθμού των τιμών που απαρτίζουν το διάστημα τιμών της παραμέτρου  $\alpha$ .

Στα Διαγράμματα 4.42-4.43 και στον Πίνακα Α.29 του Παραρτήματος Α, παρουσιάζεται μία σύγκριση των ΤΝΔ που έδωσαν την καλύτερη τιμή σφάλματος με μεθόδους της “Matlab” για τα διαστήματα τιμών  $[-998, -98, -8]$  και  $[-9998, -998, -98, -8]$ .



Διάγραμμα 4.42: Σφάλματα TNΔ και μεθόδων “Matlab” για παράμετρο  $\alpha = [-998, -98, -8]$ .



Διάγραμμα 4.43: Σφάλματα TNΔ και μεθόδων “Matlab” για παράμετρο  $\alpha = [-9998, -998, -98, -8]$ .

Όσον αφορά τη σύγκριση του TNΔ με των μεθόδων της “Matlab” που φαίνεται στα Διαγράμματα 4.42-4.43 και στους Πίνακες Α.3-Α.6 του Παραρτήματος Β, παρατηρείται η παραγωγή καλύτερων αποτελεσμάτων από το TNΔ στο διάστημα τιμών  $\alpha = [-998, -98, -8]$ , ενώ το TNΔ στο  $\alpha = [-9998, -998, -98, -8]$  υπερέρχει των Ode15s και Radau5, αλλά παράγει παρόμοια αποτελέσματα με την Ode23s αν και υπερέρχει ελάχιστα και αυτής.

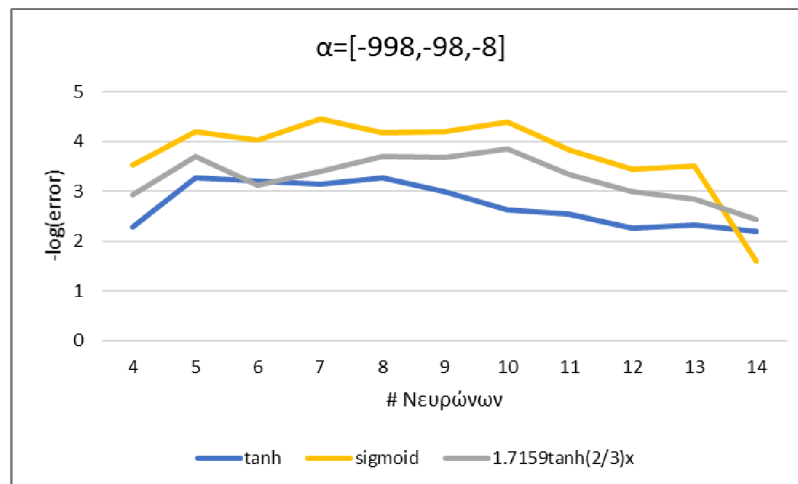
Για το Π.Α.Τ. 8 η χρήση της σιγμοειδούς συνάρτησης παράγει τα καλύτερα αποτελέσματα με μικρό σχετικά αριθμό παραμέτρων. Η γενίκευση επομένως είναι επιτυχημένη. Όσον αφορά τη σύγκριση με τις άλλες προσεγγιστικές μεθόδους, τα TNΔ είναι πιο επιτυχημένα.

## Πρόβλημα 9

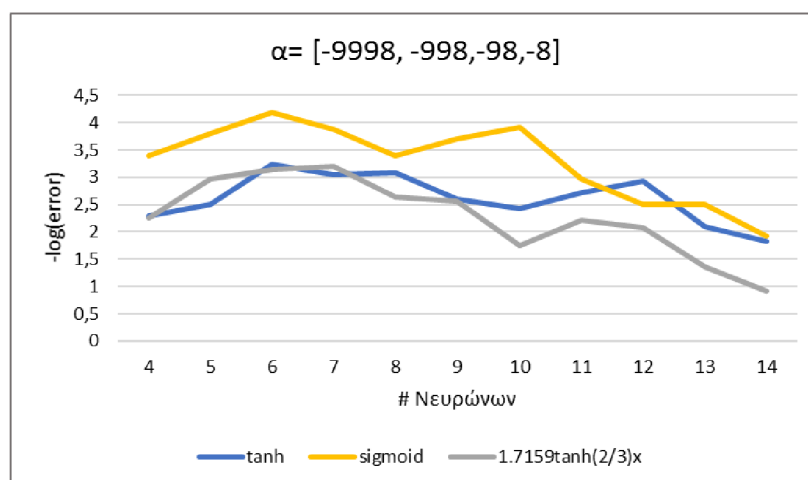
Το επόμενο Π.Α.Τ το οποίο περιέχει οικογένεια παραμέτρων περιγράφεται στη σχέση (4.7), ενώ η αναλυτική του λύση από τη σχέση (4.8). Οι αρχικές συνθήκες υπολογίζονται για την αρχική τιμή 0.5 και το σύστημα επιλύεται στο διάστημα τιμών  $[0, 3]$ . Η παράμετρος  $\alpha$  λαμβάνει τα διαστήματα τιμών  $[-998, -98, -8]$  και  $[-9998, -998, -98, -8]$ . Και σε αυτό το πρόβλημα γίνεται κανονικοποίηση των τιμών του  $\alpha$  στο διάστημα τιμών  $[0,10]$  και οι νευρώνες για τους οποίους εξετάζεται το πρόβλημα είναι από 4 έως 14.

Στο Διάγραμμα 4.14 παρουσιάζεται η αναλυτική λύση του προβλήματος. Από τους πίνακες Α.7- Α.10 του παραρτήματος Α, από τα αποτελέσματά του σφάλματος, του αριθμού βημάτων και το υπολογιστικό κόστος φαίνεται ότι το Π.Α.Τ. 9 είναι άκαμπτο.

Στους πίνακες Β.5 και Β.6 του παραρτήματος Β παρουσιάζονται αναλυτικά τα αποτελέσματα τιμών και στα Διαγράμματα 4.44-4.45 φαίνονται οι τιμές των δεκαδικών λογαρίθμων των σφαλμάτων που προκύπτουν για τα διαστήματα τιμών της παραμέτρου  $\alpha$  και μια σύγκριση των τριών συναρτήσεων ενεργοποίησης που χρησιμοποιήθηκαν.



Διάγραμμα 4.44: Σφάλματα ΤΝΔ για παράμετρο  $\alpha = [-998, -98, -8]$ .

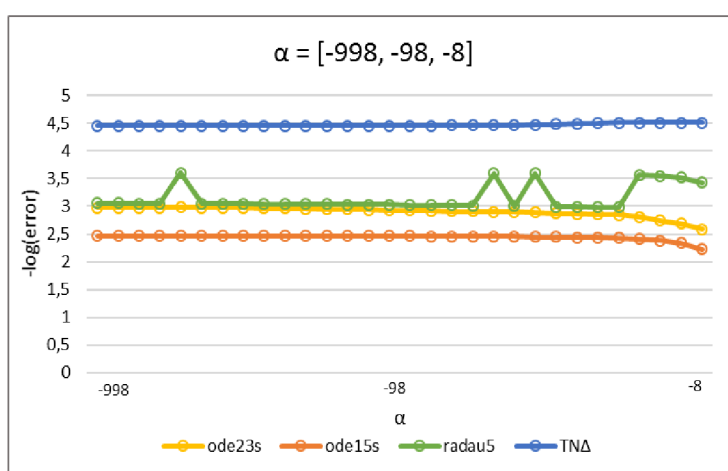




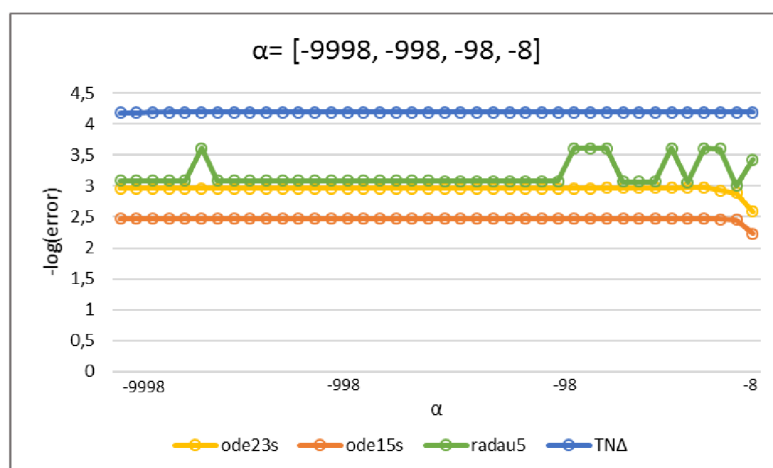
Διάγραμμα 4.45: Σφάλματα ΤΝΔ για παράμετρο  $\alpha = [-9998, -998, -98, -8]$ .

Και σε αυτό το πρόβλημα, όπως διακρίνεται από τα Διαγράμματα 4.44-4.45 και στους Πίνακες Β.5 και Β.6, η συνάρτηση ενεργοποίησης που έχει την καλύτερη συμπεριφορά είναι η σιγμοειδής, ενώ πάλι η καλύτερη τιμή και για τα δύο διαστήματα τιμών παράγεται με τη χρήση μικρού αριθμού νευρώνων (7 και 6 νευρώνων) επιτυγχάνοντας ακρίβεια 5 ψηφίων και στις δύο περιπτώσεις.

Στα Διαγράμματα 4.46-4.47 και στον Πίνακα Α.30 του Παραρτήματος Α, παρουσιάζεται μία σύγκριση των ΤΝΔ που έδωσαν την καλύτερη τιμή σφάλματος με μεθόδους της “Matlab” για τα διαστήματα τιμών  $[-998, -98, -8]$  και  $[-9998, -998, -98, -8]$ .



Διάγραμμα 4.46: Σφάλματα ΤΝΔ και μεθόδων “Matlab” για παράμετρο  $\alpha = [-998, -98, -8]$ .



Διάγραμμα 4.47: Σφάλματα ΤΝΔ και μεθόδων “Matlab” για παράμετρο  $\alpha = [-9998, -998, -98, -8]$ .

Όπως διακρίνεται στα Διαγράμματα 4.46-4.47, στη σύγκριση των αποτελεσμάτων του ΤΝΔ με τα αποτελέσματα των άλλων μεθόδων, το ΤΝΔ υπερτερεί πολύ έντονα.

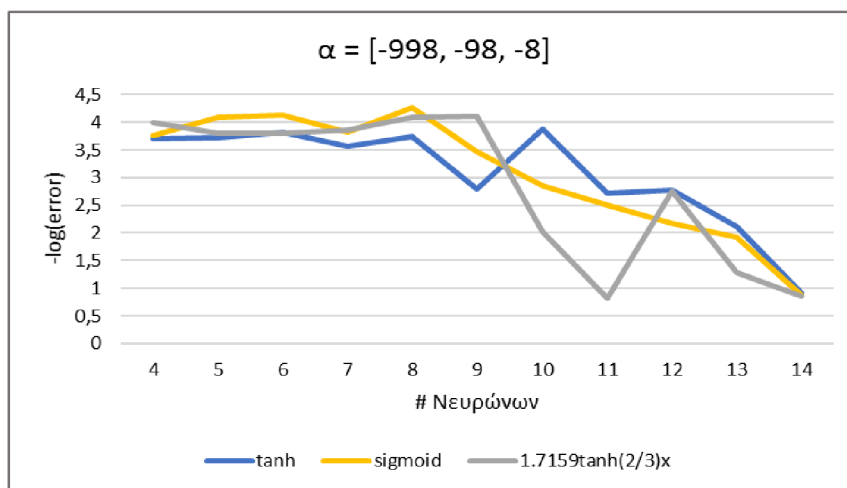
Για το Π.Α.Τ. 9 τα ΤΝΔ δίνουν ικανοποιητικής ακρίβειας προσεγγιστικές λύσεις και για τα δύο διαστήματα της παραμέτρου  $\alpha$ , οι οποίες είναι καλύτερες από όλες τις υπόλοιπες συγκρινόμενες μεθόδους.

### Πρόβλημα 10

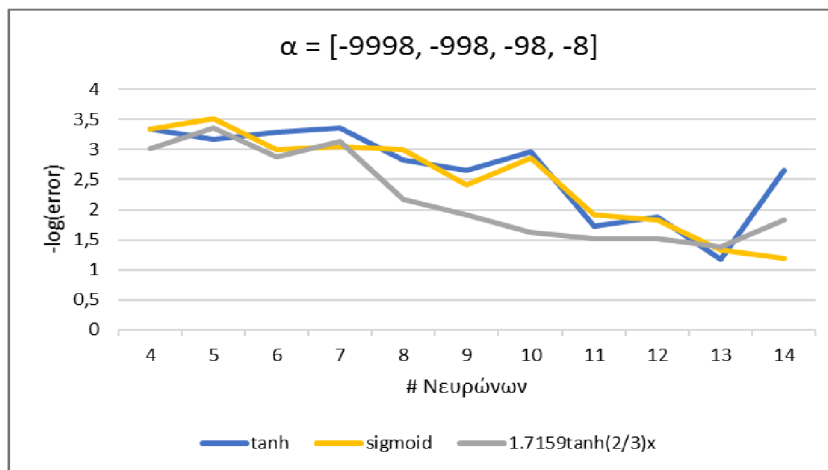
Το Π.Α.Τ. 10 περιγράφεται από τη σχέση (4.9) με αναλυτική λύση που φαίνεται στη σχέση (4.10). Οι αρχικές συνθήκες υπολογίζονται στο 0 και το διάστημα τιμών στο οποίο επιλύεται είναι το  $[0,3]$ . Η παράμετρος  $\alpha$  λαμβάνει τα διαστήματα τιμών  $[-998, -98, -8]$  και  $[-9998, -998, -98, -8]$  ενώ γίνεται κανονικοποίηση των τιμών του  $\alpha$  στο διάστημα τιμών  $[0,10]$ .

Στο Διάγραμμα 4.21 παρουσιάζεται η αναλυτική λύση του Π.Α.Τ. 10, ενώ από τους Πίνακες Α.11-Α.14 του παραρτήματος Α, από τα αποτελέσματά του σφάλματος, του αριθμού βημάτων και το υπολογιστικό κόστος φαίνεται ότι το Π.Α.Τ. 10 είναι άκαμπτο.

Στα Διαγράμματα 4.48-4.49 απεικονίζονται οι τιμές των δεκαδικών λογαρίθμων των σφαλμάτων που προκύπτουν για τα διαστήματα τιμών της παραμέτρου  $\alpha$ . Τα αποτελέσματα προκύπτουν για αριθμό νευρώνων από 4-14 και για τρεις συναρτήσεις ενεργοποίησης και παρουσιάζονται αναλυτικά στο Παράρτημα Β, στους πίνακες Β.7 και Β.8.



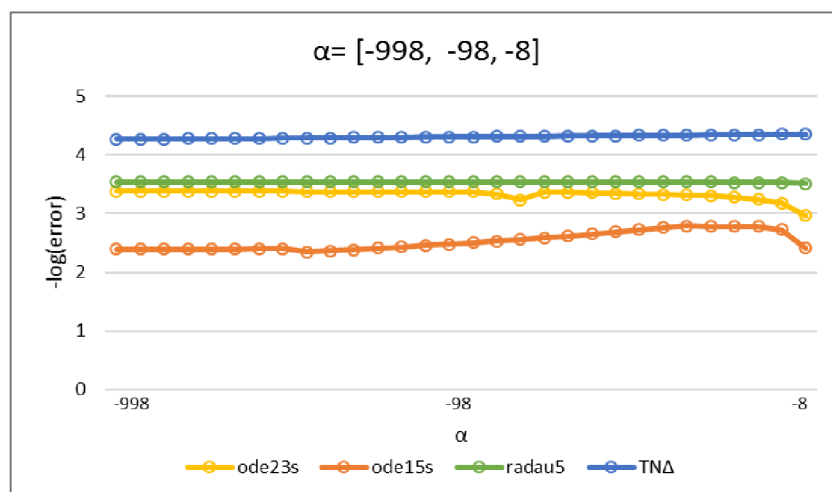
Διάγραμμα 4.48: Σφάλματα ΤΝΔ για παράμετρο  $\alpha=[-998, -98, -8]$ .



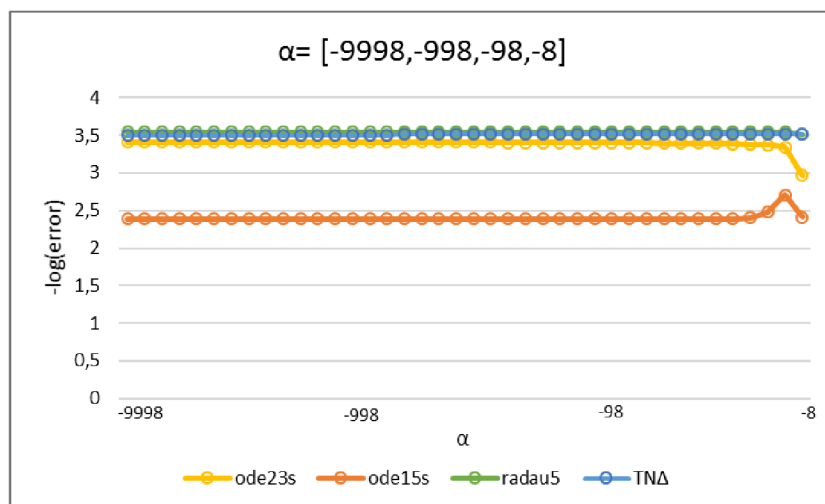
Διάγραμμα 4.49: Σφάλματα ΤΝΔ για παράμετρο  $\alpha = [-9998, -998, -98, -8]$ .

Στα Διαγράμματα 4.48-4.49 και στους Πίνακες Β.7 και Β.8 του παραρτήματος Β παρόλο που ξανά η σιγμοειδής συνάρτηση δίνει τις βέλτιστες τιμές και για τα δύο διαστήματα τιμών, η συμπεριφορά των άλλων συναρτήσεων ενεργοποίησης είναι εξίσου καλή. Οι βέλτιστες τιμές των ΤΝΔ είναι για μικρό αριθμό νευρώνων.

Στα Διαγράμματα 4.50-4.51 και στον Πίνακα Α.31 του παραρτήματος Α, παρουσιάζεται μία σύγκριση των ΤΝΔ που έδωσαν την καλύτερη τιμή σφάλματος με μεθόδους της “Matlab” για τα διαστήματα τιμών  $[-998, -98, -8]$  και  $[-9998, -998, -98, -8]$ .



Διάγραμμα 4.50: Σφάλματα ΤΝΔ και μεθόδων “Matlab” για παράμετρο  $\alpha = [-998, -98, -8]$ .



Διάγραμμα 4.51: Σφάλματα TNΔ και μεθόδων “Matlab” για παράμετρο  $\alpha = [-9998, -998, -98, -8]$ .

Από τα Διαγράμματα 4.50-4.51 παρατηρείται ότι ενώ για το πρώτο διάστημα τιμών του  $\alpha$  το TNΔ υπερέχει των υπόλοιπων μεθόδων, στο δεύτερο διάστημα τιμών, το οποίο περιέχει την υψηλότερη τιμή ακαμψίας, το TNΔ παράγει ισάξια αποτελέσματα με τη Radau5.

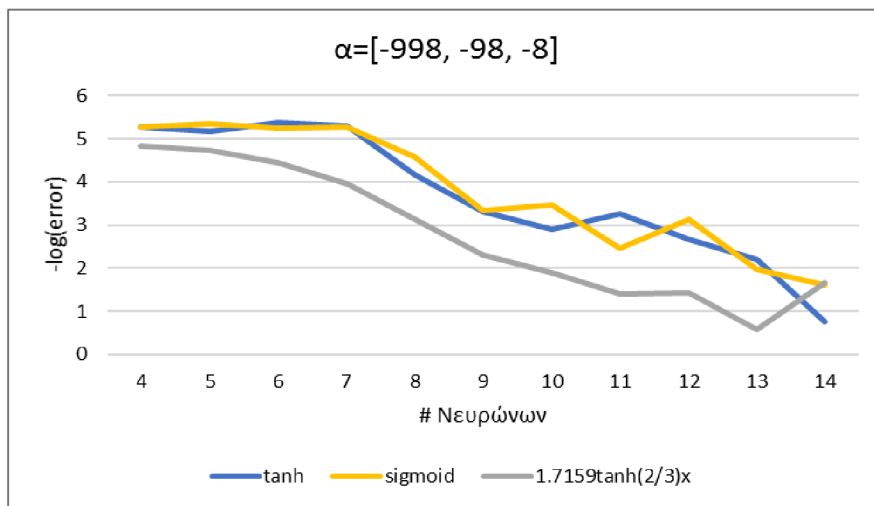
Για το Π.Α.Τ. 10 επομένως όλες οι συναρτήσεις ενεργοποίησης παράγουν ισάξια αποτελέσματα, καλύτερα γενικά σε σύγκριση με τις υπόλοιπες μεθόδους.

### Πρόβλημα 11

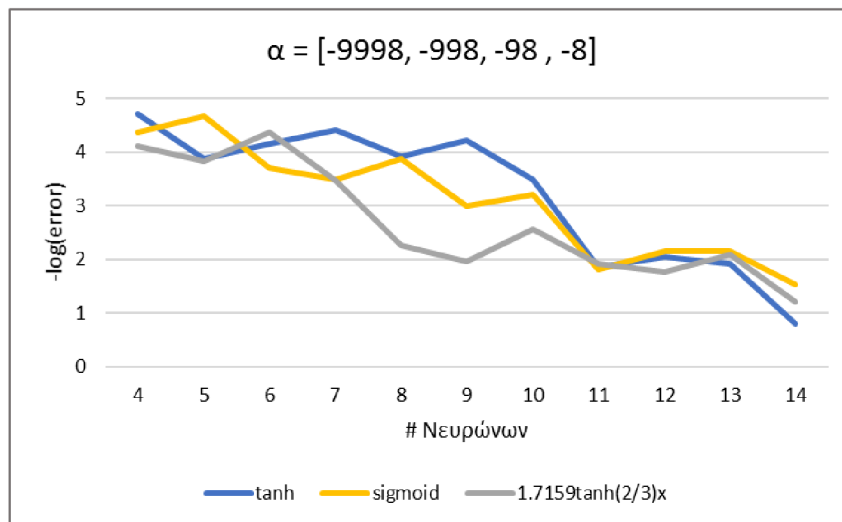
Το Π.Α.Τ. 11 περιγράφεται από τη σχέση (4.11), ενώ η αναλυτική λύση δίνεται από τη σχέση (4.12). Οι αρχικές συνθήκες υπολογίστηκαν στο 0, το σύστημα επιλύεται στο διάστημα τιμών  $[0, 3]$  και η παράμετρος  $\alpha$  λαμβάνει τα διαστήματα τιμών  $[-998, -98, -8]$  και  $[-9998, -998, -98, -8]$  και κανονικοποιείται στο διάστημα τιμών  $[0, 10]$ .

Στο Διάγραμμα 4.28 διακρίνεται η αναλυτική λύση του Π.Α.Τ. 11. Από τους Πίνακες A.15-A.18 του παραρτήματος A, από τα αποτελέσματά του σφάλματος, του αριθμού βημάτων και το υπολογιστικό κόστος φαίνεται ότι το Π.Α.Τ. 11 είναι άκαμπτο.

Τα Διαγράμματα 4.52-4.53 απεικονίζουν τις τιμές των δεκαδικών λογαρίθμων των σφαλμάτων που προκύπτουν για τα διαστήματα τιμών της παραμέτρου  $\alpha$ . Τα αποτελέσματα προκύπτουν για αριθμό νευρώνων από 4-14 και για τρεις συναρτήσεις ενεργοποίησης και παρουσιάζονται αναλυτικά στο Παράρτημα B, στους πίνακες B.9 και B.10.



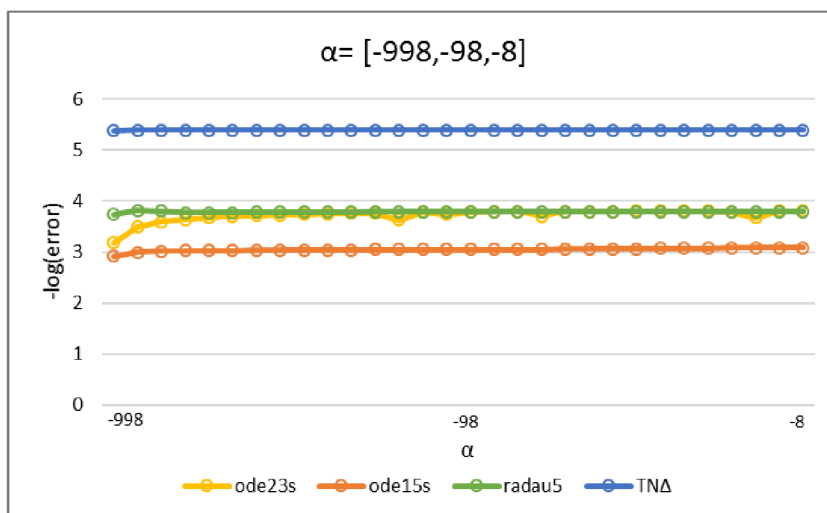
Διάγραμμα 4.52: Σφάλματα ΤΝΔ για παράμετρο  $\alpha = [-998, -98, -8]$ .



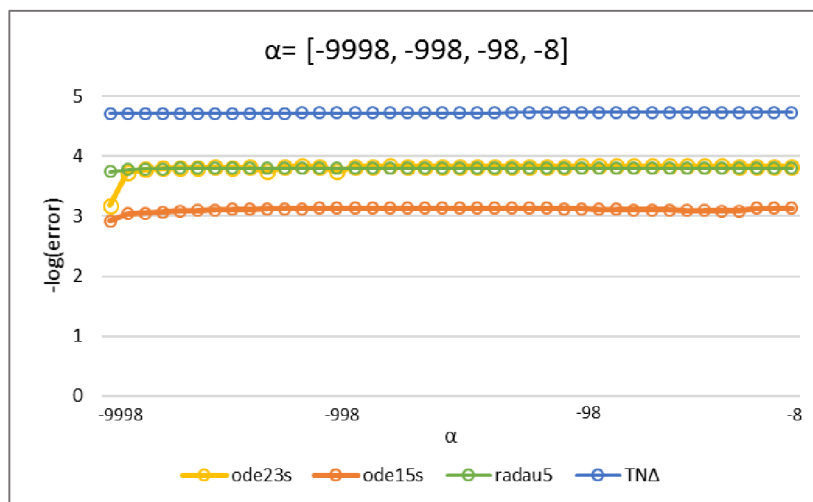
Διάγραμμα 4.53: Σφάλματα ΤΝΔ για παράμετρο  $\alpha = [-9998, -998, -98, -8]$ .

Στο Π.Α.Τ. 11 η tanh συνάρτηση ενεργοποίησης είναι καλύτερη από τις υπόλοιπες, ενώ η  $1.7159 \tanh((2/3)x)$  δεν έχει τόσο καλή απόκριση και για τα δύο διαστήματα τιμών. Ο αριθμός νευρώνων για τον οποίο επιτυγχάνεται η καλύτερη τιμή είναι μικρός και στα δύο διαστήματα, ενώ όσο αυξάνεται ο αριθμός των νευρώνων, τόσο μειώνεται η απόδοση των ΤΝΔ.

Στα Διαγράμματα 4.54-4.55 και στον Πίνακα Α.32 του Παραρτήματος Α, παρουσιάζεται μία σύγκριση των ΤΝΔ που έδωσαν την καλύτερη τιμή σφάλματος με μεθόδους της “Matlab” για τα διαστήματα τιμών  $[-998, -98, -8]$  και  $[-9998, -998, -98, -8]$ .



Διάγραμμα 4.54: Σφάλματα TNA και μεθόδων “Matlab” για παράμετρο  $\alpha = [-998, -98, -8]$ .



Διάγραμμα 4.55: Σφάλματα TNA και μεθόδων “Matlab” για παράμετρο  $\alpha = [-9998, -998, -98, -8]$ .

Από τα Διαγράμματα 4.54-4.55 φαίνεται ότι και για τα δύο TNA οι τιμές των σφαλμάτων είναι πολύ καλύτερες από όλες τις συγκρινόμενες προσεγγιστικές μεθόδους.

Για το Π.Α.Τ. 11 επομένως παράγονται πολύ καλής ακρίβειας προσεγγιστικές λύσεις, οι οποίες υπερέρχουν των υπόλοιπων προσεγγιστικών μεθόδων με τις οποίες συγκρίνονται.

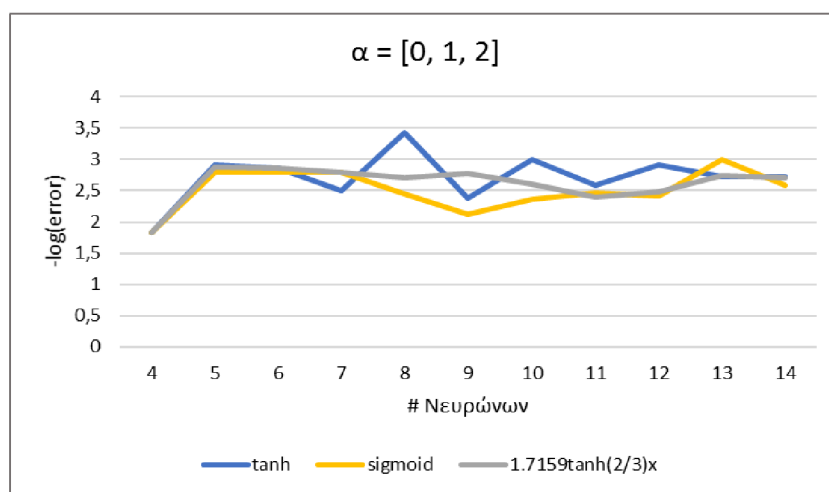
## Πρόβλημα 12

Το Π.Α.Τ. 12 περιγράφεται από τη σχέση (4.13), ενώ η αναλυτική λύση δίνεται από τη σχέση (4.14). Οι αρχικές συνθήκες υπολογίστηκαν στο 0, το σύστημα επιλύεται

στο διάστημα τιμών  $[0, 3]$  και η παράμετρος  $\alpha$  λαμβάνει το διάστημα τιμών  $[0,1,2]$  και κανονικοποιείται στο διάστημα τιμών  $[0,1]$ .

Στο Διάγραμμα 4.35 φαίνεται η αναλυτική λύση του Π.Α.Τ. 12. Από τους Πίνακες A.19-A.21 του παραρτήματος A, από τα αποτελέσματά του σφάλματος, του αριθμού βημάτων και το υπολογιστικό κόστος φαίνεται ότι το Π.Α.Τ. 12 είναι άκαμπτο.

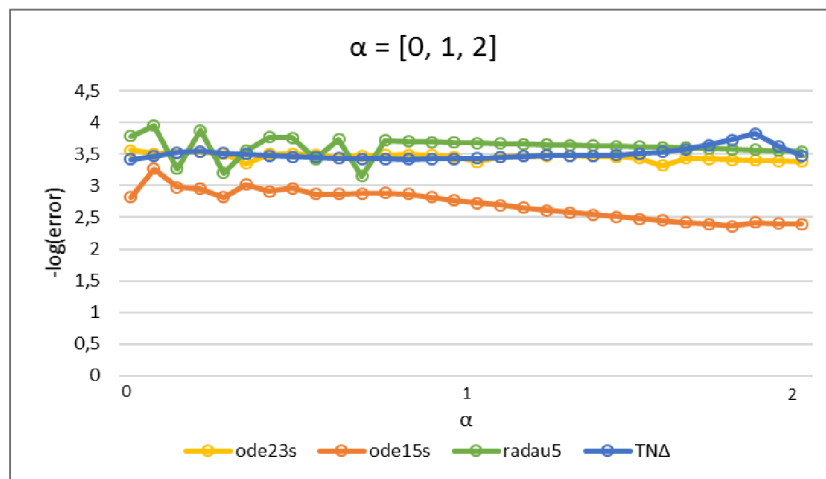
Το Διάγραμμα 4.56 απεικονίζει τις τιμές των δεκαδικών λογαρίθμων των σφαλμάτων που προκύπτουν για τα διαστήματα τιμών της παραμέτρου  $\alpha$ . Τα αποτελέσματα προκύπτουν για αριθμό νευρώνων από 4-14 και για τρεις συναρτήσεις ενεργοποίησης και παρουσιάζονται αναλυτικά στο Παράρτημα B, στους πίνακες B.11 και B.12.



Διάγραμμα 4.56: Σφάλματα ΤΝΔ για παράμετρο  $\alpha=[0, 1, 2]$ .

Για το Π.Α.Τ. 12 η βέλτιστη τιμή του ΤΝΔ έχει ακρίβεια 4 ψηφίων και επιτυγχάνεται με τη χρήση της tanh, για 8 νευρώνες, δηλαδή μικρό αριθμό νευρώνων, ενώ η χρήση των υπόλοιπων συναρτήσεων ενεργοποίησης δεν τροποποιεί ιδιαίτερα τα αποτελέσματα.

Στο Διάγραμμα 4.57 και στον Πίνακα A.33 του Παραρτήματος A, παρουσιάζεται μία σύγκριση του ΤΝΔ που έδωσε την καλύτερη τιμή σφάλματος, με μεθόδους της “Matlab” για το διάστημα τιμών  $[0,1,2]$ .



Διάγραμμα 4.57: Σφάλματα TNΔ και μεθόδων “Matlab” για παράμετρο  $\alpha=[0, 1, 2]$

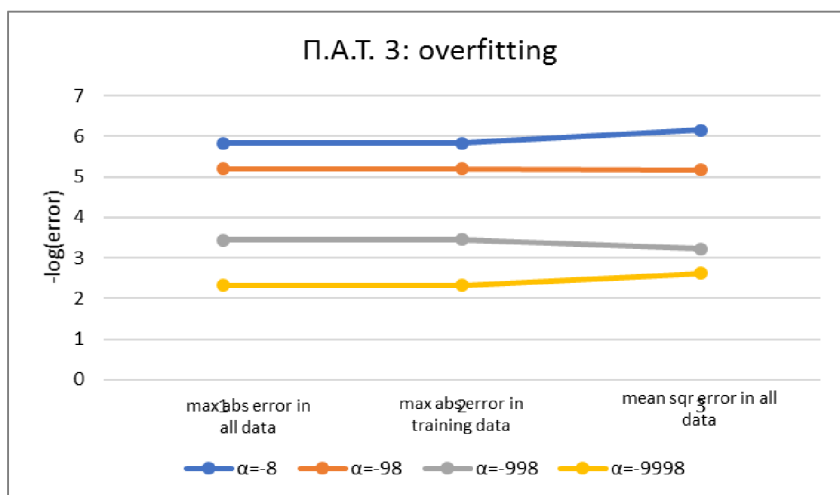
Από το Διάγραμμα 4.57 φαίνεται ότι το TNΔ για το μεγαλύτερο μέρος του διαστήματος τιμών συμπεριφέρεται χειρότερα από τη Radau5 και αρκετά παρόμοια με την Ode23s, ωστόσο όσο γίνεται πιο άκαμπτο το πρόβλημα το TNΔ δίνει καλύτερα αποτελέσματα από την Ode23s και για ορισμένες τιμές του  $\alpha$  λαμβάνει καλύτερες τιμές και από τη Radau5.

Για το Π.Α.Τ. 12 επομένως παράγονται ικανοποιητικής ακρίβειας προσεγγιστικές λύσεις, οι οποίες είναι ισάξιες των προσεγγιστικών μεθόδων με τις οποίες συγκρίνονται.

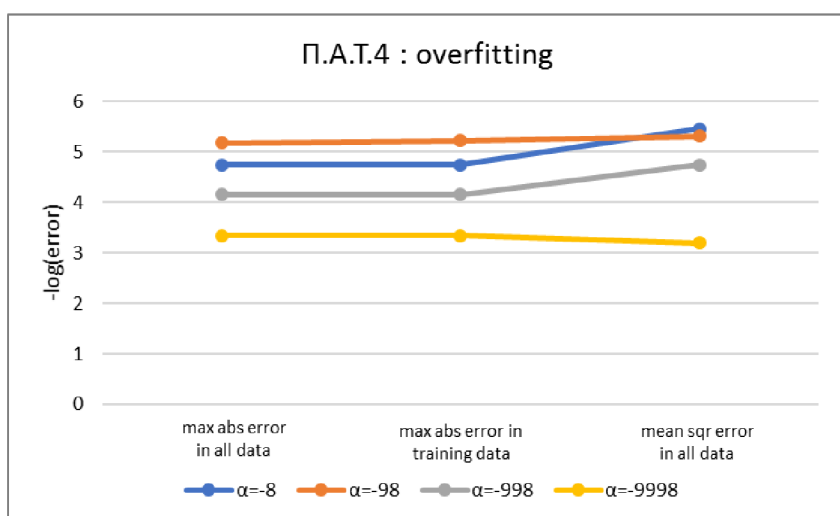
#### 4.4 Έλεγχος overfitting

Έπειτα από την παρουσίαση των αποτελεσμάτων των Π.Α.Τ. που επιλύθηκαν, παρουσιάζονται και τα αποτελέσματα από τον έλεγχο του “overfitting” στα Διαγράμματα 4.58-4.67, για τα Π.Α.Τ. 3-12. Τα αποτελέσματα φαίνονται αναλυτικά στους Πίνακες Γ.1-Γ.10 στο Παράρτημα Γ. Σε κάθε διάγραμμα παρουσιάζονται τα TNΔ που έχουν επιτύχει την καλύτερη τιμή σφάλματος για τις διάφορες τιμές της παραμέτρου. Η σύγκριση γίνεται με βάση τον ορισμό του 1 κεφαλαίου και οι τιμές οι οποίες εντοπίζονται είναι το μέγιστο απόλυτο σφάλμα στα δεδομένα εκπαίδευσης, το μέγιστο απόλυτο σφάλμα σε όλα τα δεδομένα και το μέσο τετραγωνικό σφάλμα σε όλα τα δεδομένα.

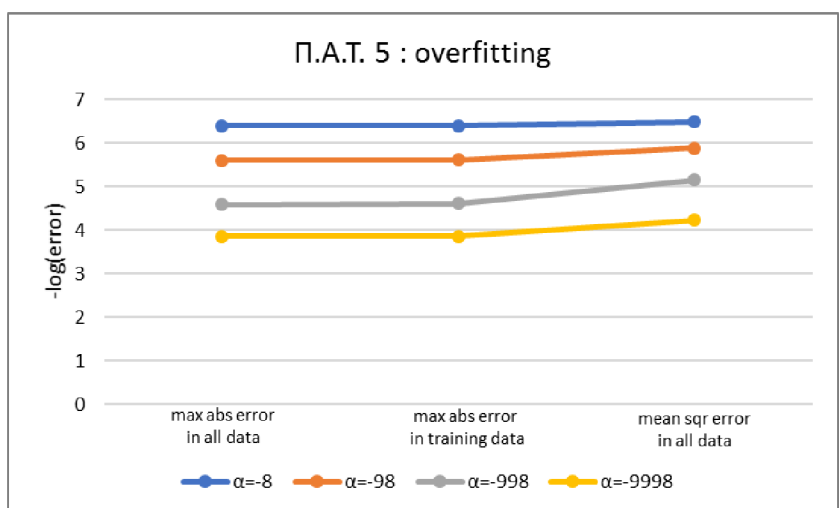




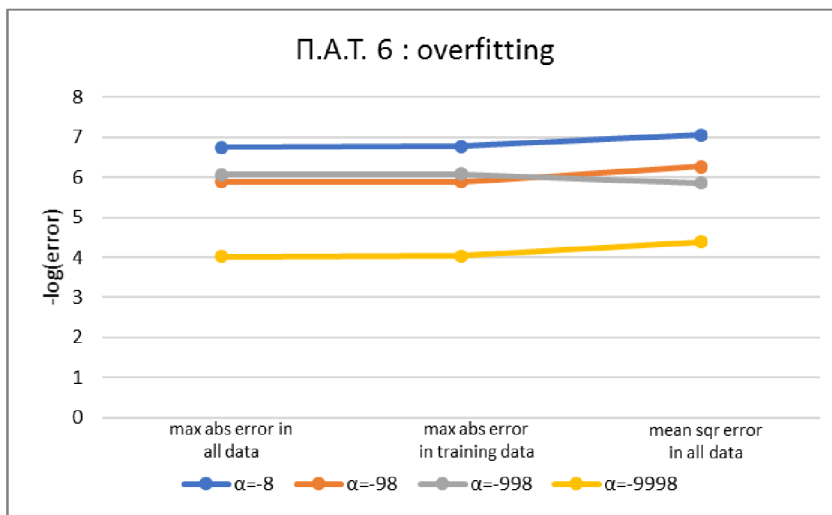
Διάγραμμα 4.58: Έλεγχος overfitting για το Π.Α.Τ. 3



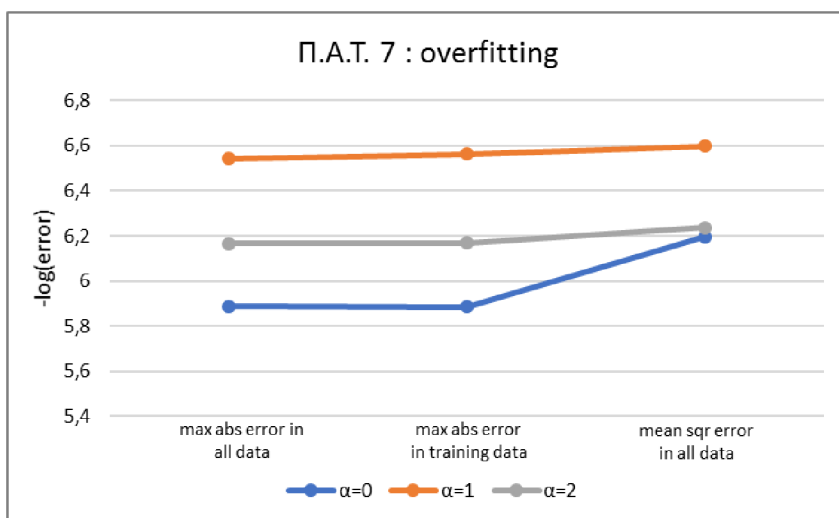
Διάγραμμα 4.59: Έλεγχος overfitting για το Π.Α.Τ. 4



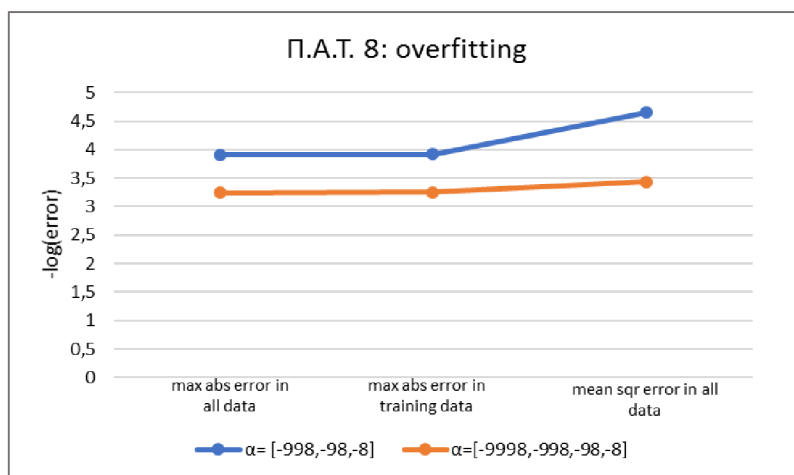
Διάγραμμα 4.60: Έλεγχος overfitting για το Π.Α.Τ. 5



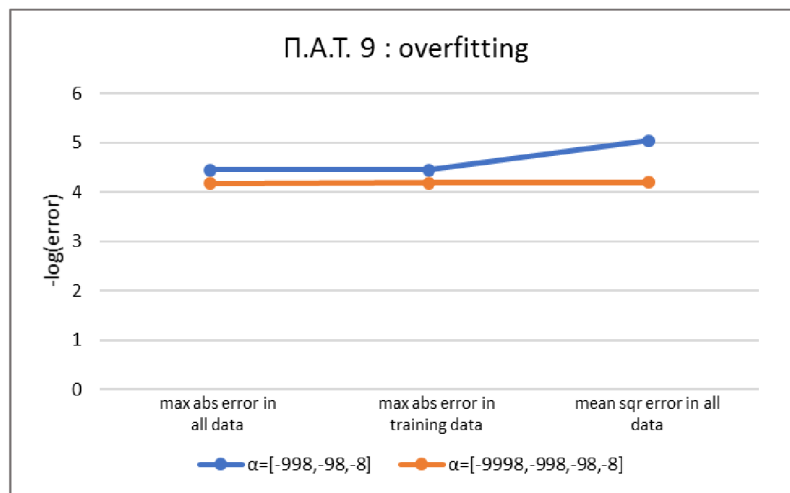
Διάγραμμα 4.61: Έλεγχος overfitting για το Π.Α.Τ. 6



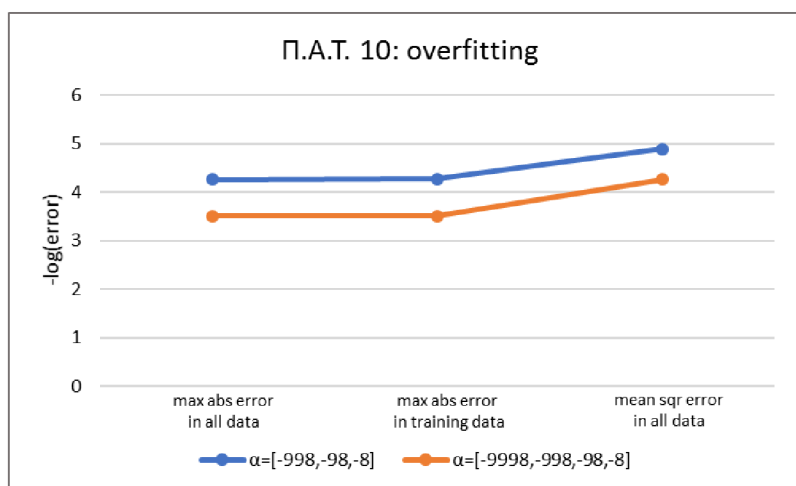
Διάγραμμα 4.62: Έλεγχος overfitting για το Π.Α.Τ. 7



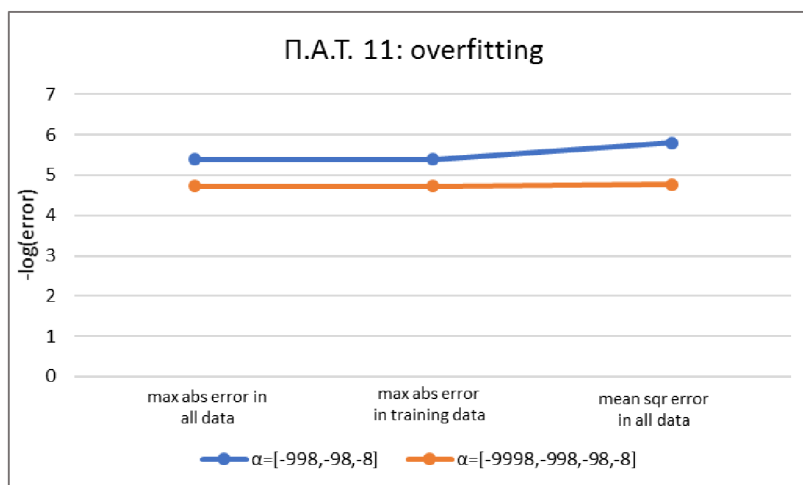
Διάγραμμα 4.63: Έλεγχος overfitting για το Π.Α.Τ. 8



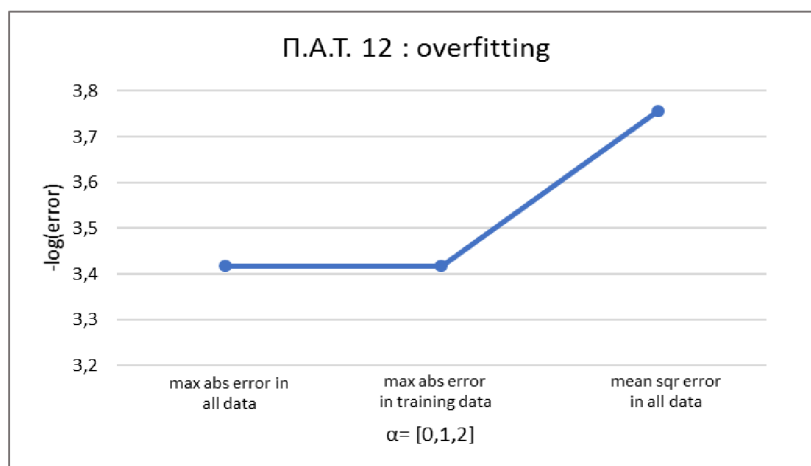
Διάγραμμα 4.64: Έλεγχος overfitting για το Π.Α.Τ. 9



Διάγραμμα 4.65: Έλεγχος overfitting για το Π.Α.Τ. 10



Διάγραμμα 4.66: Έλεγχος overfitting για το Π.Α.Τ. 11



Διάγραμμα 4.67: Έλεγχος overfitting για το Π.Α.Τ. 12

Όπως διακρίνεται από τα διαγράμματα 4.58-4.67, σε όλα τα Π.Α.Τ. το μέγιστο απόλυτο σφάλμα σε όλα τα δεδομένα έχει ίδιας τάξης σφάλμα με το μέγιστο απόλυτο σφάλμα στα δεδομένα εκπαίδευσης. Επομένως δεν υπάρχει overfitting σε κανένα από τα Π.Α.Τ. που έχουν επιλυθεί. Το μέσο τετραγωνικό σφάλμα δείχνει κατά πόσο το μοντέλο της δοκιμαστικής λύσης που χρησιμοποιείται είναι επαρκές. Σε όλα τα διαγράμματα παρατηρείται ότι το μέσο τετραγωνικό σφάλμα λαμβάνει ίδιας τάξης τιμή σφάλματος ή καλύτερες τιμές από τα υπόλοιπα σφάλματα, κάτι το οποίο αποδεικνύει ότι το μοντέλο της δοκιμαστικής λύσης που χρησιμοποιήθηκε για την επίλυση των Διαφορικών Εξισώσεων είναι επαρκές και στις περιπτώσεις που έχει λάβει καλύτερες τιμές, δηλαδή για όλες τις τιμές της παραμέτρου  $\alpha$  στα Π.Α.Τ. 4, 6, 7, 8, 10, 12 και για  $\alpha = -8$  και  $\alpha = -98$  στο Π.Α.Τ. 5 και για  $\alpha = [-998, -98, -8]$  για το Π.Α.Τ. 9 και για το Π.Α.Τ. 11 για  $\alpha = [-998, -98, -8]$ , ότι το ΤΝΔ μπορεί να επιτύχει ακόμα καλύτερες τιμές σφαλμάτων.

### 5.1 Συμπεράσματα

Τα συμπεράσματα που προκύπτουν από το σύνολο της εργασίας παρουσιάζονται παρακάτω:

#### **Γενικά:**

Τα ΤΝΔ μονής κρυφής στιβάδας μπορούν να εκπαιδευθούν για να επιλύσουν Διαφορικές Εξισώσεις και κυρίως άκαμπτες Διαφορικές Εξισώσεις, οι οποίες περιέχουν ή όχι σταθερή παράμετρο ή διάστημα παραμέτρων, με πολύ καλές ικανότητες γενίκευσης. Οι λύσεις οι οποίες προκύπτουν είναι συγκρίσιμες και πολλές φορές καλύτερης ακρίβειας από προσεγγιστικές λύσεις που προκύπτουν από κλασσικούς άκαμπτους επιλυτές.

Με την εκπαίδευση ΤΝΔ που περιέχουν παράμετρο στο σύστημα εξισώσεων τους ή κυριότερα στις αρχικές τους συνθήκες επιτυγχάνεται η επίλυση οικογένειας προβλημάτων με μικρό υπολογιστικό κόστος.

Η αρχιτεκτονική που μελετάται στην Εικόνα 3.2 προτείνεται για την επίλυση συστημάτων ΣΔΕ 2x2, έναντι της αρχιτεκτονικής που προτείνεται στη δημοσίευση των Λαγαρή κ.α [3].

Με την προτεινόμενη αρχιτεκτονική, δεν εντοπίστηκε overfitting στις λύσεις των ΤΝΔ που παρουσιάζονται.

Επίσης όσο μικρότερο είναι το διάστημα εκπαίδευσης στην παράμετρο τόσο καλύτερα προσαρμόζεται το ΤΝΔ, κάτι αναμενόμενο, ωστόσο αρκετά σημαντικό είναι ότι τελικά το ΤΝΔ εκπαιδεύεται και μαθαίνει να αποδίδει και σε άλλες τιμές της παραμέτρου εκτός από αυτές στις οποίες εκπαιδεύτηκε.

Τέλος τα αποτελέσματα τα οποία διεξήχθησαν ήταν στατιστικά σωστά, καθώς έχουν πραγματοποιηθεί 66 διαδικασίες εκπαίδευσης, κάθε μία από τις οποίες έχει επαναληφθεί 40 φορές.

#### **Συναρτήσεις ενεργοποίησης:**

Ενδιαφέρον αποτέλεσμα ήταν τα σχετικά καλά σε ακρίβεια αποτελέσματα με ελάχιστο υπολογιστικό κόστος στο πρόβλημα όπου οι γραμμικοί όροι κυριαρχούσαν στη λύση όταν χρησιμοποιήθηκε η ReLU. Είναι γνωστό ότι η συγκεκριμένη συνάρτηση ενεργοποίησης είναι κατάλληλη για ΤΝΔ στα οποία

υπάρχουν πολλά δεδομένα και χρησιμοποιούνται πολλές εσωτερικές στιβάδες (βαθιά μάθηση).

Για την γενικότερη επίλυση άκαμπτων Διαφορικών Εξισώσεων τα οποία είτε διαθέτουν είτε όχι παράμετρο στο σύστημα των εξισώσεων τους και εκπαιδεύονται με τη μέθοδο Levenberg-Marquardt προτείνεται η χρήση της σιγμοειδούς συνάρτησης ενεργοποίησης, η οποία επέδειξε γενικότερα την καλύτερη συμπεριφορά.

### ***Εκμετάλλευση των αποτελεσμάτων της έρευνας:***

Όλοι οι επιτυχείς άκαμπτων προβλημάτων της “Matlab” που χρησιμοποιήσαμε μπορούν να ολοκληρώσουν ικανοποιητικά σε μεγάλα διαστήματα τιμών για τα άκαμπτα προβλήματα, δίνοντας ικανοποιητικής ακρίβειας αποτελέσματα στα σημεία στα οποία επιλέγουν να προσεγγίσουν τη λύση. Όλες αυτές οι μέθοδοι παράγουν τις ακριβείς αυτές λύσεις στα σημεία που «πατάνε» . Για άλλα σημεία εκτός αυτών χρησιμοποιούν παρεμβολικές (“interpolatory”) ή “collocation” τεχνικές. Τα αριθμητικά πειράματα που διεξήχθησαν στην παρούσα έρευνα έδειξαν ότι αυτή η ακρίβεια μπορεί να χαθεί όταν οι μέθοδοι αυτοί χρησιμοποιούνται σε σημεία εκτός της διαμέρισης λύσης της μεθόδου, σε αντίθεση με τα ΤΝΔ, τα οποία δεν μπορούν να εκπαιδευθούν ικανοποιητικά σε μεγάλα διαστήματα, ωστόσο έχουν πολύ καλές ικανότητες γενίκευσης, διατηρώντας την ακρίβεια της μεθόδου τους ακόμα και όταν η προσεγγιστική λύση υπολογίζεται σε σημεία στα οποία δεν έχει εκπαιδευθεί, αλλά ανήκουν στο διάστημα εκπαίδευσης.

Οικογένεια προβλημάτων και προβλήματα με παράμετρο που επηρεάζουν την παράμετρο ακαμψίας ή την αρχική συνθήκη (επομένως και τη λύση) θα μπορούσαν να εκπαιδευθούν για ένα πρόβλημα της βιομηχανίας το οποίο θα γενικεύει ανάλογα με τις αρχικές συνθήκες που αλλάζουν, μειώνοντας με αυτό τον τρόπο το υπολογιστικό κόστος.

## ***6.2 Προτάσεις για μελλοντική έρευνα***

Η χρήση ΤΝΔ για την επίλυση άκαμπτων Διαφορικών Εξισώσεων και κυρίως εξισώσεων που περιέχουν παράμετρο είναι ένας τομέας έρευνας ο οποίος δεν έχει διερευνηθεί αρκετά αλλά είναι αρκετά υποσχόμενος για την επίλυση εφαρμογών που απαιτούν μεγάλο υπολογιστικό κόστος. Παρακάτω παρουσιάζονται μερικές προτάσεις για μελλοντική έρευνα που προέκυψαν από την παρούσα εργασία.

1. Επίλυση διαφορετικών τύπων άκαμπτων προβλημάτων και γενίκευση των αποτελεσμάτων σε μεγαλύτερες οικογένειες προβλημάτων.
2. Περαιτέρω μελέτη και βελτιστοποίηση της χρήσης κλασικών αλγορίθμων εκπαίδευσης ή μελέτη επιλογής αλγορίθμων με διαφορετική για την εκπαίδευση προσέγγιση, όπως οι διαφοροεξελικτικοί αλγόριθμοι (“Differential evolution”) ή οι αλγόριθμοι σμήνους (“Particle Swarm

Optimization”), με σκοπό την αποδοτικότερη απόδοση της εκπαίδευσης και παραγωγή καλύτερων προσεγγιστικών λύσεων των ΤΝΔ.

3. Εξέταση διαφορετικών τύπων ΤΝΔ για την επίλυση άκαμπτων προβλημάτων, όπως τα “Radial basis function”
4. Εξέταση της ReLU συναρτήσεως ενεργοποίησης για επίλυση γραμμικών συστημάτων.

## ΒΙΒΛΙΟΓΡΑΦΙΑ - ΠΗΓΕΣ

---

- [1] N.Yadav, A.Yadav, M.Kumar, An Introduction to Neural Network Method Differential Equations, Studies in Computational Intelligence, Springer, London, 2015
- [2] M. Hayati, B. Karami, Feedforward neural network for solving partial differential equations, Journal of Applied Sciences. URL 10.3923/jas.2007.2812.2817
- [3] I. Lagaris, A. Likas, D. Fotiadis, Artificial Neural Networks for Solving Ordinary and Partial Differential Equations (1998).
- [4] A. P. Engelbrecht, Computational Intelligence An Introduction, 2nd Edition, John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, 2007.
- [5] S. Nazmul, A. Hojjat, Computational Intelligence: Synergies of Fuzzy Logic, Neural Networks and Evolutionary Computing, John Wiley & Sons , 2013.
- [6] S. Haykin, Neural Networks A Comprehensive Foundation, 2nd Edition, Pearson Education, India, 1999.
- [7] H. Zheng, Z. Yang, W. Liu, J. Liang, Y.Li, Improving Deep Neural Networks Using Softplus Units, 2015 International Joint Conference on Neural Networks (IJCNN), 2015
- [8] I.Goodfellow, Y. Bengio, A. Courville, Deep Learning (Adaptive Computation and Machine Learning series), The MIT Press, 2016
- [9] D. F. Griffiths, D. J. Highman, Numerical Methods for Ordinary Differential Equations Initial Value Problems, Springer Undergraduate Mathematics Series, Springer-Verlag, London, 2010.
- [10] J.D.Lambert, Numerical Methods for Ordinary Differential Systems, Wiley, 1991
- [11] Matlab, *MATLAB* version 7.10.0, Natick, Massachusetts: The MathWorks Inc.,2010.
- [12] E.Hairer, G.Wanner, Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems, Second Revised Edition, Springer-Verlag, Berlin, Heidelberg, 1996
- [13] S.He, K. Reif, R. Unbehauen, Multilayer networks for solving a class of partial differential equations.Neural Netw, 2000
- [14] L.P. Aarts, P.V. Veer, Neural Network method for partial differential equations. Neural Process. Lett. 14,261-271, 2001
- [15] T. V. Lazovskaya, D.A. Tarkhov, Fresh approaches to the construction of parameterized neural network solutions of a stiff differential equation, St. Petersburg Polytechnical University Journal: Physics and Mathematics, 2015
- [16] S. Mall, S. Chakraverty, Numerical solution of nonlinear singular initial value problems of Emden–Fowler type using Chebyshev Neural Network method (2014)
- [17] A. J. Owens, D. L. Filkin, EFFICIENT TRAINING OF THE BACK PROPAGATION NETWORK BY SOLVING A SYSTEM OF STIFF ORDINARY DIFFERENTIAL EQUATIONS, International 1989 Joint Conference on Neural Networks, IEEE, Washington, DC, USA, 2002
- [18] Y. Le Cun, EFFICIENT LEARNING AND SECOND–ORDER METHODS, Tutorial presented at Neural Information Processing Systems 5,1993.



[19] V. Kaloutsa, I. Th. Famelis, On the Neural Network solution of stiff Initial Value problems, ICNAAM, Rhodes, Greece, 2019

[20]E. Hairer (2004) Fortran and Matlab Codes, Retrieved from:  
<http://www.unige.ch/~hairer/software.html>

## ΠΑΡΑΡΤΗΜΑ Α

Στους Πίνακες Α.1-Α.21 φαίνονται τα σφάλματα των Π.Α.Τ. για τις προσεγγιστικές μεθόδους που συγκρίνονται με τα ΤΝΔ.

1 <sup>ο</sup> πρόβλημα						
	ΤΝΔ (βέλτιστη)	ΤΝΔ (ReLU)	Ode23s	Ode23	Ode45	Ode15s
Βέλτιστη τιμή σφάλματος	3.9e-12	5.9 e-06	2.5-07	1.6e-06	1.1e-06	9.4e-09
Χρόνος	20.827	0.035	0.298	0.941	0.768	0.757

Πίνακας Α.1: Σύγκριση Μεθόδων “Matlab” & ΤΝΔ για τη βέλτιστη τιμή σφάλματος & το χρόνο.

2 <sup>ο</sup> πρόβλημα			
	Τ.Ν.Δ (βέλτιστη)	Ode23s	Ode23
Βέλτιστη τιμή σφάλματος	1 e-03	8.3 e-02	6.8 e-02

Πίνακας Α2: Σύγκριση Μεθόδων “Matlab” & ΤΝΔ για τη βέλτιστη τιμή σφάλματος & το χρόνο

α = -8	Ode23s	Ode23	Ode45	Ode15s	Radau5
successful steps	39	27	18	15	8
failed attempts	0	0	1	0	0
function evaluations	197	82	115	34	37
partial derivatives	39	-	-	1	1
LU decompositions	39	-	-	5	7
solutions of linear systems	117	-	-	30	9
errorODE	4.6238e-03	6.78e-03	7.807e-04	1.035e-02	4.344e-04

Πίνακας Α.3: Μέθοδοι “Matlab” για α = -8 για το Π.Α.Τ. 3

α = -98	Ode23s	Ode23	Ode45	Ode15s	Radau5
successful steps	76	120	100	16	8
failed attempts	1	18	3	0	0
function evaluations	384	415	619	36	37
partial derivatives	76	-	-	1	1
LU decompositions	77	-	-	5	7
solutions of linear systems	231	-	-	32	9
errorODE	1.175e-03	4.381e-03	1.479e-03	7.424e-03	7.776e-05

Πίνακας Α.4: Μέθοδοι “Matlab” για α = -98 για το Π.Α.Τ. 3

α = -998	Ode23s	Ode23	Ode45	Ode15s	Radau5
successful steps	103	1191	904	15	8
failed attempts	1	8	21	0	0

function evaluations	1519	3598	5551	34	37
partial derivatives	103	-	-	1	1
LU decompositions	104	-	-	5	7
solutions of linear systems	312	-	-	30	9
errorODE	6.531e-04	3.2014e-03	2.843e-03	9.856e-03	7.038e-05

Πίνακας A.5: Μέθοδοι “Matlab” για  $\alpha = -998$  για το Π.Α.Τ. 3

$\alpha = -9998$	Ode23s	Ode23	Ode45	Ode15s	Radau5
successful steps	110	11930	9035	15	8
failed attempts	1	10	607	0	0
function evaluations	554	35821	57853	34	37
partial derivatives	110	-	-	1	1
LU decompositions	111	-	-	5	7
solutions of linear systems	333	-	-	30	9
errorODE	5.567e-04	3.177e-03	3.249e-03	9.865e-03	6.975e-05

Πίνακας A.6: Μέθοδοι “Matlab” για  $\alpha = -9998$  για το Π.Α.Τ. 3

$\alpha = -8$	Ode23s	Ode23	Ode45	Ode15s	Radau5
successful steps	15	14	12	23	9
failed attempts	0	0	0	0	0
function evaluations	77	43	73	50	41
partial derivatives	15	-	-	1	1
LU decompositions	15	-	-	6	9
solutions of linear systems	45	-	-	46	10
errorODE	2.038e-03	5.259e-03	1.841e-03	5.880e-03	3.607e-05

Πίνακας A.7: Μέθοδοι “Matlab” για  $\alpha = -8$  για το Π.Α.Τ. 4

$\alpha = -98$	ode23s	Ode23	Ode45	Ode15s	radau
successful steps	22	99	77	23	9
failed attempts	0	7	1	0	0
function evaluations	112	319	469	33	44
partial derivatives	22	-	-	1	1
LU decompositions	22	-	-	6	9
solutions of linear systems	66	-	-	29	11
errorODE	2.581e-03	1.219e-02	6.864e-03	4.015e-03	7.902e-05

Πίνακας A.8: Μέθοδοι “Matlab” για  $\alpha = -98$  για το Π.Α.Τ. 4

$\alpha = -998$	Ode23s	Ode23	Ode45	Ode15s	Radau5
successful steps	26	993	753	23	14
failed attempts	1	5	49	0	3
function evaluations	134	2995	4813	33	85
partial derivatives	26	-	-	1	4

LU decompositions	27	-	-	6	10
solutions of linear systems	81	-	-	29	20
errorODE	1.186e-03	1.355e-02	8.927e-03	3.436e-03	8.788e-04

Πίνακας A.9: Μέθοδοι “Matlab” για  $\alpha = -998$  για το Π.Α.Τ. 4

$\alpha = -9998$	Ode23s	Ode23	Ode45	Ode15s	Radau5
successful steps	27	9943	7530	23	14
failed attempts	1	11	505	0	3
function evaluations	139	29863	48211	33	85
partial derivatives	27	-	-	1	4
LU decompositions	28	-	-	6	10
solutions of linear systems	84	-	-	29	20
errorODE	9.415e-04	1.612e-02	9.969e-03	3.433e-03	8.218e-04

Πίνακας A.10: Μέθοδοι “Matlab” για  $\alpha = -9998$  για το Π.Α.Τ. 4

$\alpha = -8$	Ode23s	Ode23	Ode45	Ode15s	Radau5
successful steps	35	24	26	17	8
failed attempts	3	1	2	0	1
function evaluations	183	76	109	38	43
partial derivatives	35	-	-	2	2
LU decompositions	38	-	-	5	7
solutions of linear systems	114	-	-	34	10
errorODE	9.7e-04	4.1e-03	8e-04	2.1e-03	7e-05

Πίνακας A.11: Μέθοδοι “Matlab” για  $\alpha = -8$  για το Π.Α.Τ. 5

$\alpha = -98$	Ode23s	Ode23	Ode45	Ode15s	Radau5
successful steps	65	122	97	17	8
failed attempts	4	28	8	1	1
function evaluations	335	451	631	40	43
partial derivatives	65	-	-	1	2
LU decompositions	69	-	-	6	8
solutions of linear systems	207	-	-	36	10
errorODE	6.3e-04	3.9e-03	2.4e-03	2.1 e-03	2.9e-05

Πίνακας A.12: Μέθοδοι “Matlab” για  $\alpha = -98$  για το Π.Α.Τ. 5

$\alpha = -998$	Ode23s	Ode23	Ode45	Ode15s	Radau5
successful steps	90	1191	904	17	8
failed attempts	6	14	36	1	1
function evaluations	464	3616	5641	40	43
partial derivatives	90	-	-	1	2
LU decompositions	96	-	-	6	8
solutions of linear systems	288	-	-	36	10
errorODE	4e-04	5.1e-03	2.8e-03	2.1e-03	2.3e-05

Πίνακας A.13: Μέθοδοι “Matlab” για  $\alpha = -998$  για το Π.Α.Τ. 5

$\alpha = -998$	Ode23s	Ode23	Ode45	Ode15s	Radau5
successful steps	98	11932	9035	17	8
failed attempts	5	10	608	1	1
function evaluations	502	35827	57859	40	43
partial derivatives	98	-	-	1	2
LU decompositions	103	-	-	6	8
solutions of linear systems	309	-	-	36	10
errorODE	3.7e-04	3.9e-03	3.3e-03	2.1e-03	2.3e-05

Πίνακας A.14: Μέθοδοι “Matlab” για  $\alpha = -9998$  για το Π.Α.Τ. 5

$\alpha = -8$	Ode23s	Ode23	Ode45	Ode15s	Radau5
successful steps	33	22	17	20	7
failed attempts	1	3	1	0	0
function evaluations	169	76	109	44	33
partial derivatives	33	-	-	1	1
LU decompositions	34	-	-	6	6
solutions of linear systems	102	-	-	40	8
errorODE	6.8e-04	1.7e-03	5.2e-04	1.2e-03	5.5e-05

Πίνακας A.15: Μέθοδοι “Matlab” για  $\alpha = -8$  για το Π.Α.Τ. 6

$\alpha = -98$	Ode23s	Ode23	Ode45	Ode15s	Radau5
successful steps	58	119	93	21	7
failed attempts	1	27	2	0	0
function evaluations	294	439	571	46	33
partial derivatives	58	-	-	1	1
LU decompositions	59	-	-	6	6
solutions of linear systems	177	-	-	42	8
errorODE	4.2e-04	1.3e-03	5.5e-04	9.9e-04	3.8e-05

Πίνακας A.16: Μέθοδοι “Matlab” για  $\alpha = -98$  για το Π.Α.Τ. 6

$\alpha = -998$	Ode23s	Ode23	Ode45	Ode15s	Radau5
successful steps	75	1191	904	22	8
failed attempts	2	7	37	2	0
function evaluations	381	3595	5647	52	37
partial derivatives	75	-	-	1	1
LU decompositions	77	-	-	7	7
solutions of linear systems	231	-	-	48	9
errorODE	1.6e-04	1.6e-03	1.2e-03	8.6e-04	7e-06

Πίνακας A.17: Μέθοδοι “Matlab” για  $\alpha = -998$  για το Π.Α.Τ. 6

$\alpha = -9998$	Ode23s	Ode23	Ode45	Ode15s	Radau5
successful steps	79	11930	9037	21	8
failed attempts	3	8	604	4	0
function evaluations	403	35815	57847	54	37
partial derivatives	79	-	-	1	1
LU decompositions	82	-	-	9	7
solutions of linear systems	246	-	-	50	9
errorODE	1.5e-04	1.7e-03	1.1e-03	7.7e-04	4.6e-06

Πίνακας A.18: Μέθοδοι “Matlab” για  $\alpha = -9998$  για το Π.Α.Τ. 6

$\alpha = 0$	ode23s	Ode23	Ode45	Ode15s	radau
successful steps	101	1193	906	18	9
failed attempts	6	7	40	0	1
function evaluations	519	3601	5677	27	53
partial derivatives	101	-	-	1	2
LU decompositions	107	-	-	5	9
solutions of linear systems	321	-	-	23	13
errorODE	2.7e-04	1.7e - 3	1.1e - 3	1.5e-03	1.119e-05

Πίνακας A.19: Μέθοδοι “Matlab” για  $\alpha = 0$  για το Π.Α.Τ. 7

$\alpha = 1$	ode23s	Ode23	Ode45	Ode15s	radau
successful steps	94	1191	903	17	8
failed attempts	6	12	47	1	1
function evaluations	484	3610	5701	40	43
partial derivatives	94	-	-	1	2
LU decompositions	100	-	-	6	8
solutions of linear systems	300	-	-	36	10
errorODE	3.269e-04	2.709e-03	1.967e-03	1.381e-03	1.842e-05

Πίνακας A.20: Μέθοδοι “Matlab” για  $\alpha = 1$  για το Π.Α.Τ. 7

$\alpha = 2$	ode23s	Ode23	Ode45	Ode15s	radau
successful steps	90	1191	904	17	8
failed attempts	6	14	36	1	1
function evaluations	464	3616	5641	40	43
partial derivatives	90	-	-	1	2
LU decompositions	96	-	-	6	8
solutions of linear systems	288	-	-	36	10
errorODE	4.013e-04	5.086e-03	2.849e - 3	2.125e-3	2.284e-05

Πίνακας A.21: Μέθοδοι “Matlab” για  $\alpha = 2$  για το Π.Α.Τ. 7

Π.Α.Τ. 3			
<b>Ode23s</b>	<b>Ode15s</b>	<b>Radau5</b>	<b>TNΔ</b>
0,004616055	0,008496606	0,00108768	1,46E-06
0,001190501	0,010985376	0,000753914	1,84E-05
0,000613545	0,009134915	0,000726038	0,000365536
5,82E-04	9,14E-03	0,000719072	0,004798432

A.22: Μέθοδοι “Matlab” και TNΔ για σύγκριση σε όλο το διάστημα τιμών για το Π.Α.Τ.3

Π.Α.Τ. 4			
<b>Ode23s</b>	<b>Ode15s</b>	<b>Radau5</b>	<b>TNΔ</b>
0,002581	0,005891	0,000372	1,82E-05
0,001651	0,003942	0,000273	6,55E-06
0,001053	0,003413	0,000879	6,95E-05
0,001084	0,00341	0,000822	0,000453

A.23: Μέθοδοι “Matlab” και TNΔ για σύγκριση σε όλο το διάστημα τιμών για το Π.Α.Τ. 4

Π.Α.Τ. 5			
<b>Ode23s</b>	<b>Ode15s</b>	<b>Radau5</b>	<b>TNΔ</b>
0,00109	0,003853	0,000305	6,88E-07
0,000537	0,001656	0,000292	2,49E-06
0,00041	0,004055	0,000288	2,55E-05
0,000389	0,004057	0,000288	0,000145

A.24: Μέθοδοι “Matlab” και TNΔ για σύγκριση σε όλο το διάστημα τιμών για το Π.Α.Τ. 5

Π.Α.Τ. 6			
<b>Ode23s</b>	<b>Ode15s</b>	<b>Radau5</b>	<b>TNΔ</b>
0,000663	0,001199	0,000182	1,75E-07
0,000236	0,000945	0,000167	1,34E-06
0,000157	0,000818	0,000161	8,56E-07
0,00015	0,000739	0,000159	9,38E-05

A.25: Μέθοδοι “Matlab” και TNΔ για σύγκριση σε όλο το διάστημα τιμών για το Π.Α.Τ. 6

Π.Α.Τ. 7			
<b>Ode23s</b>	<b>Ode15s</b>	<b>Radau5</b>	<b>TNΔ</b>
2,73E-04	0,000365	0,00041	1,30E-06
1,51E-03	0,001769	0,004055	2,87E-07
1,64E-04	0,000208	0,000288	6,83E-07
2,73E-04	0,000365	0,00041	1,30E-06

A.26: Μέθοδοι “Matlab” και TNΔ για σύγκριση σε όλο το διάστημα τιμών για το Π.Α.Τ. 7

Π.Α.Τ.1	Ode23s	Ode23	Ode45	Ode15s
successful steps	176	3972	3008	37
failed attempts	9	7	189	4
function evaluations	724	11938	19183	55
partial derivatives	176	-	-	1

LU decompositions	185	-	-	11
solutions of linear systems	555	-	-	52

A.27: Διερεύνηση Ακαμψίας για το Π.Α.Τ.1

Π.Α.Τ.2	Ode23s	Ode23	Ode45	Ode15s
successful steps	39	329	248	63
failed attempts	0	3	16	3
function evaluations	197	997	1585	136
partial derivatives	39	-	-	1
LU decompositions	39	-	-	16
solutions of linear systems	117	-	-	132

A.28: Διερεύνηση Ακαμψίας για το Π.Α.Τ.2

α	ode23s	ode15s	radau5	TNA	α	ode23s	ode15s	radau5	TNA
[.998,-98,-8]	6.14E-04	9.13E-03	7.26E-04	1.23E-04	[.998,-98,-8]	5.82E-04	9.14E-03	7.19E-04	5.16E-04
	6.12E-04	9.13E-03	7.26E-04	1.21E-04		5.29E-04	9.14E-03	7.19E-04	5.18E-04
	6.17E-04	9.13E-03	7.27E-04	1.20E-04		5.29E-04	9.14E-03	7.19E-04	5.19E-04
	6.50E-04	9.13E-03	7.27E-04	1.18E-04		5.29E-04	9.14E-03	7.19E-04	5.20E-04
	6.27E-04	9.13E-03	7.27E-04	1.16E-04		5.28E-04	9.14E-03	7.19E-04	5.22E-04
	6.27E-04	9.13E-03	7.27E-04	1.14E-04		5.28E-04	9.14E-03	7.19E-04	5.23E-04
	6.33E-04	9.13E-03	7.28E-04	1.13E-04		5.27E-04	9.14E-03	7.19E-04	5.24E-04
	6.43E-04	9.13E-03	7.28E-04	1.11E-04		5.27E-04	9.14E-03	7.19E-04	5.26E-04
	6.41E-04	9.13E-03	7.28E-04	1.09E-04		5.27E-04	9.14E-03	7.19E-04	5.27E-04
	6.55E-04	9.13E-03	7.29E-04	1.08E-04		5.28E-04	9.14E-03	7.19E-04	5.28E-04
	6.54E-04	9.13E-03	7.29E-04	1.06E-04		5.28E-04	9.14E-03	7.19E-04	5.30E-04
	6.61E-04	9.13E-03	7.30E-04	1.04E-04		5.29E-04	9.14E-03	7.19E-04	5.31E-04
	6.69E-04	9.13E-03	7.30E-04	1.03E-04		5.30E-04	9.14E-03	7.19E-04	5.32E-04
	6.76E-04	9.13E-03	7.30E-04	1.01E-04		5.31E-04	9.14E-03	7.20E-04	5.34E-04
	6.83E-04	9.13E-03	7.31E-04	9.91E-05		5.31E-04	9.14E-03	7.20E-04	5.35E-04
	6.95E-04	9.13E-03	7.31E-04	9.74E-05		5.32E-04	9.14E-03	7.20E-04	5.36E-04
	7.07E-04	9.13E-03	7.32E-04	9.57E-05		5.33E-04	9.14E-03	7.20E-04	5.38E-04
	7.19E-04	9.13E-03	7.33E-04	9.40E-05		5.33E-04	9.14E-03	7.20E-04	5.39E-04
	7.33E-04	9.12E-03	7.33E-04	9.23E-05		5.39E-04	9.14E-03	7.20E-04	5.40E-04
	7.52E-04	9.13E-03	7.34E-04	9.06E-05		5.75E-04	9.14E-03	7.20E-04	5.42E-04
	7.74E-04	9.39E-03	7.34E-04	8.89E-05		6.12E-04	9.14E-03	7.20E-04	5.43E-04
	8.00E-04	9.88E-03	7.35E-04	8.72E-05		5.40E-04	9.14E-03	7.20E-04	5.44E-04
	8.32E-04	1.07E-02	7.34E-04	8.55E-05		5.38E-04	9.14E-03	7.20E-04	5.46E-04
	8.73E-04	1.19E-02	7.39E-04	8.38E-05		5.40E-04	9.14E-03	7.20E-04	5.47E-04
	9.38E-04	1.21E-02	7.45E-04	8.22E-05		5.42E-04	9.14E-03	7.20E-04	5.48E-04
	1.01E-03	1.09E-02	7.50E-04	8.05E-05		5.44E-04	9.14E-03	7.21E-04	5.50E-04
	1.12E-03	1.05E-02	7.50E-04	7.89E-05		5.75E-04	9.14E-03	7.21E-04	5.51E-04
	1.32E-03	1.19E-02	7.59E-04	7.69E-05		5.52E-04	9.14E-03	7.21E-04	5.53E-04
	1.76E-03	1.02E-02	7.68E-04	7.44E-05		5.50E-04	9.14E-03	7.21E-04	5.54E-04
	4.62E-03	8.50E-03	1.09E-03	7.38E-05		5.55E-04	9.14E-03	7.22E-04	5.55E-04

A.29 : Μέθοδοι “Matlab” και TNA για σύγκριση σε όλο το διάστημα τιμών για το Π.Α.Τ. 8



$\alpha$	ode23s	ode15s	radau5	TNΔ	$\alpha$	ode23s	ode15s	radau5	TNΔ
[-998,-98,-8]	0,001053	0,003413	0,000879	3,51E-05	[-9998,-998,-98,-8]	0,001084	0,00341	0,000822	6,48E-05
	0,001051	0,003413	0,000881	3,51E-05		0,001084	0,00341	0,000822	6,48E-05
	0,001049	0,003413	0,000884	3,51E-05		0,001084	0,00341	0,000822	6,48E-05
	0,001046	0,003414	0,000886	3,50E-05		0,001084	0,00341	0,000822	6,47E-05
	0,001043	0,003414	0,000247	3,50E-05		0,001084	0,00341	0,000823	6,47E-05
	0,001049	0,003414	0,000892	3,50E-05		0,001084	0,00341	0,000243	6,47E-05
	0,001056	0,003414	0,000895	3,50E-05		0,001084	0,00341	0,000823	6,46E-05
	0,001064	0,003414	0,000899	3,50E-05		0,001085	0,00341	0,000823	6,46E-05
	0,001074	0,003414	0,000903	3,50E-05		0,001085	0,00341	0,000823	6,46E-05
	0,001089	0,003415	0,000907	3,50E-05		0,001085	0,00341	0,000824	6,46E-05
	0,001104	0,003415	0,000911	3,50E-05		0,001085	0,00341	0,000824	6,45E-05
	0,001112	0,003415	0,000916	3,50E-05		0,001085	0,00341	0,000824	6,45E-05
	0,001137	0,003415	0,000921	3,49E-05		0,001085	0,00341	0,000825	6,45E-05
	0,001153	0,003419	0,000927	3,49E-05		0,001085	0,00341	0,000825	6,44E-05
	0,00117	0,003426	0,000933	3,49E-05		0,001085	0,00341	0,000825	6,44E-05
	0,001186	0,003435	0,00094	3,48E-05		0,001085	0,003411	0,000826	6,44E-05
	0,001202	0,003444	0,000947	3,47E-05		0,001085	0,003411	0,000826	6,43E-05
	0,001218	0,003455	0,000956	3,46E-05		0,001085	0,003411	0,000827	6,43E-05
	0,001233	0,003468	0,000964	3,45E-05		0,001085	0,003411	0,000827	6,43E-05
	0,001246	0,003484	0,000252	3,43E-05		0,001085	0,003411	0,000828	6,42E-05
	0,001263	0,003503	0,000984	3,40E-05		0,001085	0,003411	0,000828	6,42E-05
	0,001291	0,003528	0,000254	3,36E-05		0,001085	0,003411	0,000829	6,42E-05
	0,001322	0,003559	0,001007	3,31E-05		0,001085	0,003411	0,00083	6,41E-05
	0,001354	0,0036	0,001019	3,25E-05		0,001085	0,003411	0,000831	6,41E-05
	0,001386	0,003658	0,00103	3,16E-05		0,001085	0,003411	0,000832	6,41E-05
	0,001413	0,003742	0,001037	3,11E-05		0,001085	0,003411	0,000833	6,40E-05
	0,001558	0,003874	0,00027	3,11E-05		0,001084	0,003411	0,000834	6,40E-05
	0,001809	0,004106	0,00028	3,10E-05		0,001084	0,003411	0,000836	6,40E-05
	0,002032	0,004585	0,000301	3,09E-05		0,001083	0,003411	0,000244	6,40E-05
	0,002581	0,005891	0,000372	3,06E-05		0,001083	0,003411	0,000244	6,39E-05
-	-	-	-	-		0,001081	0,003411	0,000244	6,39E-05
-	-	-	-	-		0,001079	0,003412	0,000846	6,39E-05
-	-	-	-	-		0,001077	0,003412	0,00085	6,38E-05
-	-	-	-	-		0,001073	0,003412	0,000856	6,38E-05
-	-	-	-	-		0,001066	0,003412	0,000246	6,38E-05
-	-	-	-	-		0,001055	0,003413	0,000877	6,37E-05
-	-	-	-	-		0,00106	0,003414	0,000247	6,37E-05
-	-	-	-	-		0,001169	0,003426	0,000249	6,37E-05
-	-	-	-	-		0,001306	0,003542	0,001001	6,36E-05
-	-	-	-	-		0,002581	0,005891	0,000372	6,36E-05

Α.30 : Μέθοδοι “Matlab” και TNΔ για σύγκριση σε όλο το διάστημα τιμών για το Π.Α.Τ. 9

$\alpha$	ode23s	ode15s	radau5	TNΔ	$\alpha$	ode23s	ode15s	radau5	TNΔ
[-998,-98,-8]	0,00041	0,004055	0,000288	5,41E-05	[-9998,-998,-98,-8]	0,000389	0,004057	0,000288	0,000315
	0,000411	0,004055	0,000288	5,38E-05		0,000389	0,004057	0,000288	0,000314
	0,000412	0,004055	0,000288	5,34E-05		0,000389	0,004057	0,000288	0,000314
	0,000413	0,004055	0,000288	5,31E-05		0,000389	0,004057	0,000288	0,000314
	0,000414	0,004055	0,000288	5,28E-05		0,000389	0,004057	0,000288	0,000313
	0,000415	0,004026	0,000288	5,24E-05		0,000389	0,004057	0,000288	0,000313
	0,000416	0,003972	0,000288	5,21E-05		0,000389	0,004057	0,000288	0,000313
	0,000418	0,003916	0,000288	5,17E-05		0,000389	0,004057	0,000288	0,000313
	0,000419	0,004467	0,000288	5,14E-05		0,00039	0,004057	0,000288	0,000312
	0,00042	0,004284	0,000288	5,11E-05		0,00039	0,004057	0,000288	0,000312
	0,000421	0,004089	0,000288	5,07E-05		0,00039	0,004057	0,000288	0,000312
	0,000422	0,003898	0,000288	5,04E-05		0,00039	0,004057	0,000288	0,000311
	0,000424	0,003711	0,000288	5,00E-05		0,00039	0,004057	0,000288	0,000311
	0,000425	0,003515	0,000288	4,97E-05		0,000391	0,004057	0,000288	0,000311
	0,000426	0,003334	0,000288	4,94E-05		0,000391	0,004057	0,000288	0,000311
	0,000427	0,003136	0,000289	4,90E-05		0,000391	0,004057	0,000288	0,00031
	0,000465	0,00296	0,000289	4,87E-05		0,000391	0,004057	0,000288	0,00031
	0,000583	0,002764	0,000289	4,84E-05		0,000392	0,004057	0,000288	0,00031
	0,00043	0,002592	0,000289	4,80E-05		0,000392	0,004057	0,000288	0,000309
	0,000433	0,0024	0,000289	4,77E-05		0,000392	0,004057	0,000288	0,000309
	0,000442	0,002232	0,000289	4,73E-05		0,000393	0,004056	0,000288	0,000309
	0,000452	0,002047	0,000289	4,70E-05		0,000393	0,004056	0,000288	0,000309
	0,000462	0,001888	0,00029	4,67E-05		0,000394	0,004056	0,000288	0,000308
	0,000472	0,001719	0,00029	4,63E-05		0,000394	0,004056	0,000288	0,000308
	0,000482	0,00161	0,00029	4,60E-05		0,000395	0,004056	0,000288	0,000308
	0,000495	0,00164	0,000291	4,56E-05		0,000395	0,004056	0,000288	0,000307
	0,000526	0,001656	0,000292	4,53E-05		0,000396	0,004056	0,000288	0,000307
	0,000564	0,001641	0,000294	4,50E-05		0,000397	0,004056	0,000288	0,000307
	0,000655	0,001858	0,000297	4,46E-05		0,000397	0,004056	0,000288	0,000307
	0,00109	0,003853	0,000305	4,43E-05		0,000398	0,004056	0,000288	0,000306
-	-	-	-	-		0,000399	0,004056	0,000288	0,000306
-	-	-	-	-		0,000401	0,004056	0,000288	0,000306
-	-	-	-	-		0,000402	0,004056	0,000288	0,000305
-	-	-	-	-		0,000403	0,004055	0,000288	0,000305
-	-	-	-	-		0,000404	0,004055	0,000288	0,000305
-	-	-	-	-		0,000409	0,004055	0,000288	0,000305
-	-	-	-	-		0,000417	0,003944	0,000288	0,000304
-	-	-	-	-		0,000426	0,003336	0,000288	0,000304
-	-	-	-	-		0,000457	0,00197	0,000289	0,000304
-	-	-	-	-		0,00109	0,003853	0,000305	0,000303

Α.31 : Μέθοδοι “Matlab” και TNΔ για σύγκριση σε όλο το διάστημα τιμών για το Π.Α.Τ. 10

$\alpha$	ode23s	ode15s	radau5	TNΔ	$\alpha$	ode23s	ode15s	radau5	TNΔ
[.998-.98-.8]	0.000663	0.001199	0.000182	4.09E-06	[.9998-.998-.98-.8]	0.000663	0.001199	0.000182	1.95E-05
	0.000321	0.001007	0.000154	4.09E-06		0.000184	0.000918	0.000165	1.94E-05
	0.000255	0.000958	0.000156	4.09E-06		0.000165	0.000885	0.000161	1.94E-05
	0.000229	0.000939	0.000171	4.09E-06		0.000159	0.000856	0.000161	1.94E-05
	0.000211	0.000932	0.000169	4.09E-06		0.000155	0.000813	0.00016	1.93E-05
	0.000196	0.000929	0.000168	4.09E-06		0.000155	0.000796	0.00016	1.93E-05
	0.00019	0.000925	0.000165	4.09E-06		0.000152	0.00078	0.00016	1.93E-05
	0.000186	0.000921	0.000165	4.09E-06		0.000157	0.000769	0.00016	1.92E-05
	0.000182	0.000916	0.000165	4.09E-06		0.000151	0.000763	0.00016	1.92E-05
	0.000178	0.000911	0.000164	4.08E-06		0.000182	0.000757	0.00016	1.92E-05
	0.000175	0.000906	0.000164	4.08E-06		0.000151	0.000751	0.00016	1.91E-05
	0.000172	0.000901	0.000163	4.08E-06		0.000147	0.000745	0.00016	1.91E-05
	0.000234	0.000896	0.000162	4.08E-06		0.000151	0.00074	0.000159	1.91E-05
	0.000166	0.000891	0.000162	4.08E-06		0.000179	0.00074	0.000159	1.90E-05
	0.000184	0.000886	0.000161	4.08E-06		0.000148	0.00074	0.000159	1.90E-05
	0.000166	0.000885	0.000161	4.08E-06		0.000149	0.00074	0.000159	1.90E-05
	0.000161	0.000886	0.000161	4.08E-06		0.000147	0.00074	0.000159	1.89E-05
	0.00016	0.000886	0.000161	4.07E-06		0.000148	0.000739	0.000159	1.89E-05
	0.000195	0.000883	0.000161	4.07E-06		0.000149	0.000739	0.000159	1.89E-05
	0.000159	0.000877	0.000161	4.07E-06		0.00015	0.000739	0.000159	1.88E-05
	0.000161	0.000871	0.000161	4.07E-06		0.00015	0.000739	0.000159	1.88E-05
	0.00016	0.000865	0.000161	4.07E-06		0.000149	0.000739	0.000159	1.88E-05
	0.000159	0.000859	0.000161	4.07E-06		0.00015	0.000739	0.000159	1.87E-05
	0.000156	0.000853	0.000161	4.07E-06		0.00015	0.000739	0.000159	1.87E-05
	0.000156	0.000847	0.000161	4.07E-06		0.000149	0.000739	0.000159	1.87E-05
	0.000158	0.000842	0.000161	4.07E-06		0.000149	0.000743	0.000159	1.86E-05
	0.000165	0.000836	0.000161	4.06E-06		0.000148	0.00075	0.000159	1.86E-05
	0.000212	0.00083	0.000161	4.06E-06		0.000147	0.000757	0.000159	1.86E-05
	0.000156	0.000824	0.000161	4.06E-06		0.000146	0.000765	0.000159	1.86E-05
	0.000157	0.000818	0.000161	4.06E-06		0.000145	0.000772	0.000159	1.86E-05
-	-	-	-	-		0.000145	0.000779	0.000159	1.86E-05
-	-	-	-	-		0.000146	0.000786	0.000159	1.86E-05
-	-	-	-	-		0.000146	0.000792	0.000159	1.86E-05
-	-	-	-	-		0.000146	0.000799	0.000159	1.86E-05
-	-	-	-	-		0.000146	0.000805	0.000159	1.87E-05
-	-	-	-	-		0.000147	0.000811	0.000159	1.87E-05
-	-	-	-	-		0.000148	0.000818	0.000159	1.87E-05
-	-	-	-	-		0.000149	0.000739	0.000159	1.87E-05
-	-	-	-	-		0.000149	0.000739	0.000159	1.87E-05
-	-	-	-	-		0.00015	0.000739	0.000159	1.87E-05

A.32 : Μέθοδοι “Matlab” και TNΔ για σύγκριση σε όλο το διάστημα τιμών για το Π.Α.Τ. 11

$\alpha$	ode23s	ode15s	radau5	TNΔ
[0, 1, 2]	0.000273	0.001509	0.000164	0.000383
	0.00031	0.000535	0.000112	0.000343
	0.000296	0.00105	0.000542	0.000299
	0.00029	0.001118	0.000133	0.000288
	0.000312	0.001504	0.000631	0.000302
	0.000443	0.000958	0.000281	0.000317
	0.000313	0.001215	0.000171	0.000331
	0.000309	0.001091	0.000175	0.000343
	0.000325	0.001354	0.000376	0.000354
	0.000361	0.00134	0.000184	0.000364
	0.000334	0.001323	0.000702	0.000371
	0.000328	0.001306	0.000193	0.000376
	0.000323	0.001361	0.000197	0.000378
	0.000326	0.001522	0.000201	0.000377
	0.000348	0.001682	0.000206	0.000373
	0.000418	0.001853	0.00021	0.000365
	0.000345	0.002036	0.000214	0.000353
	0.00034	0.002221	0.000219	0.000337
	0.000333	0.002424	0.000223	0.000327
	0.000333	0.002624	0.000227	0.000333
	0.000338	0.002845	0.000232	0.000333
	0.000345	0.003061	0.000236	0.000326
	0.000365	0.003299	0.00024	0.000311
	0.000481	0.003533	0.000245	0.00029
	0.000367	0.003786	0.000249	0.000262
	0.000377	0.004041	0.000257	0.000228
	0.000386	0.004306	0.000265	0.000187
	0.000393	0.003774	0.000273	0.00015
	0.000401	0.003915	0.00028	0.000238
	0.00041	0.004055	0.000288	0.000336

A.33 : Μέθοδοι “Matlab” και TNΔ για σύγκριση σε όλο το διάστημα τιμών για το Π.Α.Τ. 12

## ΠΑΡΑΡΤΗΜΑ Β

Στους Πίνακες Β.1-Β.12 φαίνονται τα σφάλματα που προκύπτουν από τα ΤΝΔ για κάθε Π.Α.Τ.

1 <sup>ο</sup> πρόβλημα	
$x \in [0,1]$ με 1 bias	
Βέλτιστη τιμή σφάλματος	3.9 e-12 (3)
$x \in [0,1]$ με 2 bias	
Βέλτιστη τιμή σφάλματος	4.6 e-11 (7)
$x \in [0,10]$ με 1 bias	
Βέλτιστη τιμή σφάλματος	1.7 e-08 (5)

Πίνακας Β.1 : Σύγκριση ΤΝΔ με χρήση 1 & 2 πολώσεων στο [0,1] & σύγκριση με τη χρήση 1 πόλωσης στο [0,1] και στο [0,10]

2 <sup>ο</sup> πρόβλημα	
<i>Μέθοδος Λαγαρή, Λίκα &amp; Φωτιάδη (<math>x \in [0,10]</math>)</i>	
Βέλτιστη τιμή σφάλματος	2.6 e-02 (5)
<i>Αρχιτεκτονική που μελετάται με 1 bias (<math>x \in [0,10]</math>)</i>	
Βέλτιστη τιμή σφάλματος	2 e-03 (7)
<i>Αρχιτεκτονική που μελετάται με 2 bias (<math>x \in [0,10]</math>)</i>	
Βέλτιστη τιμή σφάλματος	1 e-03 (6)

Πίνακας Β.2: Σύγκριση μεθόδου Λαγαρή κ.α με μέθοδο που μελετάται για επίλυση του Π.Α.Τ 2

<b>Πρόβλημα Αρχικών Τιμών 3</b>	
<i>α : σταθερή τιμή</i>	<i>Βέλτιστη τιμή σφάλματος</i>
-8	1.5 e-06 (9)
-98	6.4 e-06 (12)
-998	3.2 e-04 (14)
-9998	2.5 e-03 (5)
<i>α: διάστημα τιμών</i>	<i>Βέλτιστη τιμή σφάλματος</i>
<i>[-998,-98,-8]</i>	1.2 e-04 (6)
<i>[-9998,-998,-98,-8]</i>	4.7 e-03 (4)

Πίνακας Β.3: Βέλτιστες τιμές σφαλμάτων ΤΝΔ για τα Π.Α.Τ. 3 & 8

<b>3ο Πρόβλημα</b>											
<b><math>\alpha = -8</math></b>											
	4	5	6	7	8	9	10	11	12	13	14
tanh	2,6e-05	1,6e-05	3,5e-05	1,3e-05	<b>2,1e-06</b>	4,4e-06	5,8e-06	2,4e-06	4,6e-06	2,2e-06	4,2e-06
sigmoid	1,4e-05	1,7e-06	3,6e-06	5,6e-06	6,4e-06	<b>1,5e-06</b>	7,4e-06	1,5e-06	2,5e-06	2,2e-06	1,9e-06
atanhbx	5,3e-05	2,5e-05	1,6e-05	8,5e-06	4,5e-06	4,6e-06	2,9e-06	2,6e-06	<b>1,8e-06</b>	3,7e-06	1,8e-06
<b><math>\alpha = -98</math></b>											
	4	5	6	7	8	9	10	11	12	13	14
tanh	5,7e-04	7,7e-04	3,0e-04	2,7e-04	1,9e-04	6,7e-05	1,9e-04	2,2e-04	7,1e-05	1,4e-04	<b>4,2e-05</b>
sigmoid	9,1e-05	4,9e-05	1,74e-04	3,7e-05	5,8e-05	3,6e-05	7,3e-05	1,3e-04	6,1e-05	2,5e-05	<b>1,8e-05</b>
atanhbx	5,7e-04	6,3e-04	1,2e-04	3,3e-04	1,7e-04	1,5e-04	5,4e-05	1,7e-05	<b>6,4e-06</b>	3,2e-05	2,5e-05
<b><math>\alpha = -998</math></b>											
	4	5	6	7	8	9	10	11	12	13	14
tanh	2,6e-03	3,9e-03	2,3e-03	4,9e-03	1,7e-03	1,4e-03	6,6e-04	1e-03	1e-03	8,9e-04	<b>3,2e-04</b>
sigmoid	9,7e-04	1,2e-03	4,5e-04	5,7e-04	6,3e-04	<b>3,7e-04</b>	4,4e-04	1,2e-03	7,5e-04	5,7e-04	1,3e-03
atanhbx	7,6e-04	1,4e-03	5,9e-03	3,6e-03	8,3e-04	1,1e-03	1,1e-03	1e-03	<b>4e-04</b>	4,1e-04	9,2e-04
<b><math>\alpha = -9998</math></b>											
	4	5	6	7	8	9	10	11	12	13	14
tanh	7,3e-02	7,5e-02	1,8e-02	5,4e-02	6,3e-03	2,1e-02	2,6e-02	1,3e-02	8,9e-03	6,2e-03	<b>3e-03</b>
sigmoid	1,9e-02	<b>2,5e-03</b>	1,2e-02	7e-03	4,8e-03	1,1e-02	1,1e-02	1,1e-02	7,6e-03	1,3e-02	1,3e-02
atanhbx	7,3e-02	5,4e-02	1,14e-02	1e-02	1,7e-02	1,5e-02	2,5e-02	<b>6,8e-03</b>	1,61e-02	1,06e-02	1,5e-02
<b><math>\alpha = [-998, -98, -8]</math></b>											
	4	5	6	7	8	9	10	11	12	13	14
tanh	1,2e-03	4,7e-04	<b>4,3e-04</b>	1,2e-03	1,7e-03	9,16e-04	4,3e-03	1,147e-03	3,9e-03	0,16	7,3e-03
sigmoid	1,6e-04	4,7e-04	<b>1,2e-04</b>	2,4e-04	5,3e-04	2,6e-04	3,8e-04	3,9e-04	5,1e-03	7,1e-02	1,8e-03
atanhbx	5e-04	7,1e-04	6e-04	9,2e-04	<b>2,2e-04</b>	3,6e-03	1,9e-03	2,1e-02	4e-03	2,1e-01	1e-01
<b><math>\alpha = [-9998, -998, -98, -8]</math></b>											
	4	5	6	7	8	9	10	11	12	13	14
tanh	6,58e-03	<b>9,5e-04</b>	9,6e-04	1,6e-03	2,5e-03	1,7e-03	1,4e-03	1,8e-03	5,6e-03	1,9e-03	5,8e-03
sigmoid	2,2e-03	<b>5,7e-04</b>	6,4e-04	6,8e-04	6,3e-04	6,7e-04	5,8e-04	9,9e-04	1,3e-03	2,6e-03	1,9e-03
atanhbx	<b>4,7e-03</b>	7,6e-03	8,7e-03	9,3e-03	2e-02	1,4e-01	9,9e-02	1,3e-01	5,9e-01	5,6e-03	1,6e-01

Πίνακας Β.4: Τιμές σφαλμάτων ΤΝΔ για το Π.Α.Τ. 3 & 8

<b>Πρόβλημα Αρχικών Τιμών 4</b>	
<i><math>\alpha</math> : σταθερή τιμή παραμέτρου</i>	<i>Βέλτιστη τιμή σφάλματος</i>
-8	1.4 e-05 (12)
-98	6.6 e-06 (9)
-998	6.9 e-05 (12)
-9998	4.5 e-04 (6)
<i><math>\alpha</math> : διάστημα τιμών</i>	<i>Βέλτιστη τιμή σφάλματος</i>
<i>[-998, -98, -8]</i>	3.5 e-05 (7)
<i>[-9998, -998, -98, -8]</i>	6.5e-05 (6)

Πίνακας Β.5: Βέλτιστες τιμές σφαλμάτων ΤΝΔ για το Π.Α.Τ. 4&9

<b>4ο Πρόβλημα</b>											
$\alpha = -8$											
	4	5	6	7	8	9	10	11	12	13	14
tanh	1,8e-04	1,4e-04	2,8e-05	<b>1,8e-05</b>	4e-05	2,1e-05	4,2e-05	3,2e-05	3,1e-05	1,9e-05	2,3e-05
sigmoid	1,7e-04	4,1e-05	5,3e-05	3,6e-05	3,2e-05	3,1e-05	2e-05	2,2e-05	<b>1,9e-05</b>	2e-05	2,3e-05
atanhbx	1,7e-04	1,6e-04	1,3e-04	1,8e-05	2e-05	3,2e-05	1,6e-05	2,8e-05	<b>1,4e-05</b>	2,5e-05	2,6e-05
$\alpha=-98$											
	4	5	6	7	8	9	10	11	12	13	14
tanh	4,4e-04	4,4e-04	4,8e-04	5,4e-05	7e-05	8,1e-05	<b>4,8e-05</b>	6,7e-05	7,4e-05	5,9e-05	8,5e-05
sigmoid	9,1e-05	3,6e-05	2,5e-05	3,2e-05	2,3e-05	<b>6,6e-06</b>	1,4e-05	2,5e-05	5,6e-05	6,4e-05	3,1e-05
atanhbx	6,3e-04	5e-04	1,5e-04	5,5e-05	1,6e-04	5,1e-05	4,7e-05	<b>3,2e-05</b>	9,4e-05	7,8e-05	9,8e-05
$\alpha=-998$											
	4	5	6	7	8	9	10	11	12	13	14
tanh	1,8e-03	3,9e-04	6,6e-04	<b>1,7e-04</b>	1,8e-04	1,6e-04	1,8e-04	4,4e-04	4,3e-04	4,4e-04	4,1e-04
sigmoid	3,8e-04	1,8e-04	1,3e-04	1,7e-04	1,5e-04	<b>7,1e-05</b>	7,9e-05	1,9e-04	4,3e-04	2,4e-04	8,5e-05
atanhbx	1,7e-02	7,8e-04	4,2e-04	4,3e-04	1,8e-04	1,2e-04	1,8e-04	1,8e-04	<b>6,9e-05</b>	1,5e-04	1,8e-04
$\alpha=-9998$											
	4	5	6	7	8	9	10	11	12	13	14
tanh	3,1e-02	5,2e-02	6,6e-03	<b>5,4e-03</b>	7,2e-03	3,6e-02	1,3e-02	2,5e-02	2,7e-02	4,3e-02	9,8e-03
sigmoid	2,6e-03	41,e-03	<b>4,5e-04</b>	4,3e-03	5,7e-03	7,1e-03	7,2e-03	5,23e-03	7,2e-03	2,8e-03	2,3e-03
atanhbx	2,1e-02	4,6e-02	1,5e-02	1,5e-02	1,5e-02	1,3e-02	1,2e-02	2,2e-02	<b>1,2e-03</b>	3,2e-02	2,3e-02
$\alpha=[-998,-98,-8]$											
	4	5	6	7	8	9	10	11	12	13	14
tanh	5,2e-03	5,4e-04	6,3e-04	7,2e-04	<b>5,2e-04</b>	1e-03	2,3e-03	2,8e-03	5,5e-03	4,8e-03	6,3e-03
sigmoid	2,9e-04	6,4e-05	9,3e-05	<b>3,5e-05</b>	6,6e-05	6,4e-05	4,1e-05	1,5e-04	3,5e-04	3,1e-04	2,5e-02
atanhbx	1,2e-03	2e-04	7,5e-04	4e-04	2e-04	2,1e-04	<b>1,4e-04</b>	4,5e-04	1e-03	1,4e-03	3,6e-03
$\alpha= [-9998,-998,-98,-8]$											
	4	5	6	7	8	9	10	11	12	13	14
tanh	5,2e-03	3,2e-03	<b>5,8e-04</b>	9,2e-04	8,2e-04	2,5e-03	3,7e-03	1,9e-03	1,2e-03	7,9e-03	1,5e-02
sigmoid	4e-04	1,6e-04	<b>6,5e-05</b>	1,3e-04	4e-04	2e-04	1,2e-04	1,1e-03	3,2e-03	3,2e-03	1,2e-02
atanhbx	5,5e-03	1,1e-03	7,3e-04	<b>6,4e-04</b>	2,3e-03	2,7e-03	1,8e-02	6e-03	8,2e-03	4,3e-02	1,2e-01

Πίνακας Β.6: Τιμές σφαλμάτων ΤΝΔ για το Π.Α.Τ. 4 & 9

<b>Πρόβλημα Αρχικών Τιμών 5</b>	
$\alpha$ : σταθερή τιμή παραμέτρου	Βέλτιστη τιμή σφάλματος
-8	4 e-07 (11)
-98	2.5 e-06 (10)
-998	2.6 e-05 (10)
-9998	1.5 e-04 (13)
$\alpha$ : διάστημα τιμών	Βέλτιστη τιμή σφάλματος
$[-998,-98,-8]$	5.4 e-05 (8)
$[-9998,-998,-98,-8]$	3.1 e-04 (5)

Πίνακας Β.7: Βέλτιστες τιμές σφαλμάτων ΤΝΔ για το Π.Α.Τ. 5 & 10

<b>5ο Πρόβλημα</b>											
<b><math>\alpha = -8</math></b>											
	4	5	6	7	8	9	10	11	12	13	14
tanh	5,8e-05	3e-05	1,9e-06	5,2e-06	1,2e-06	1,2e-06	5,4e-06	1,2e-06	9,9e-07	1,2e-06	<b>5,9e-07</b>
sigmoid	5,6e-06	3,1e-06	4,8e-07	1,8e-06	1e-06	6,6e-07	5,7e-07	<b>4e-07</b>	6,8e-07	7,8e-07	6,9e-07
atanhbx	5,2e-06	1,5e-05	6,3e-06	1,7e-06	2e-06	6,8e-07	<b>5,5e-07</b>	9,3e-07	1,1e-06	5,8e-07	6,6e-07
<b><math>\alpha = -98</math></b>											
	4	5	6	7	8	9	10	11	12	13	14
tanh	1,5e-04	3,3e-05	3,3e-05	3,5e-05	3,2e-05	3,6e-05	3,5e-05	5,8e-06	5,9e-06	2,1e-05	<b>3,7e-06</b>
sigmoid	1,1e-04	2,1e-05	2,3e-05	1,6e-05	2,9e-05	2,3e-05	<b>2,5e-06</b>	7,9e-06	1,9e-05	2,2e-05	3,1e-05
atanhbx	5,9e-05	5,9e-05	1,8e-05	7,4e-05	3e-05	2e-05	2e-05	2,4e-05	2e-05	2,1e-05	<b>6,5e-06</b>
<b><math>\alpha = -998</math></b>											
	4	5	6	7	8	9	10	11	12	13	14
tanh	2,6e-04	4,3e-04	1,8e-04	2,6e-04	1,2e-04	1,6e-04	<b>2,6e-05</b>	1,9e-04	2,1e-04	2,9e-04	1,3e-04
sigmoid	2,4e-04	1,9e-04	1,7e-04	1,7e-04	<b>1,2e-04</b>	2,6e-04	1,7e-04	1,9e-04	1,8e-04	1,9e-04	1,9e-04
atanhbx	4,4e-04	3,7e-04	<b>4,3e-05</b>	2,7e-04	1e-04	4,5e-04	9,1e-05	1,9e-04	2,1e-04	7,7e-05	8,9e-05
<b><math>\alpha = -9998</math></b>											
	4	5	6	7	8	9	10	11	12	13	14
tanh	5,1e-03	1,3e-03	9,1e-03	1,1e-03	1,1e-03	1,1e-03	3,2e-03	1,3e-03	1,7e-03	<b>1,5e-04</b>	1,3e-03
sigmoid	6,7e-03	3,8e-03	2,1e-03	1e-03	<b>7,8e-04</b>	1,1e-03	1,2e-03	2,8e-03	9,8e-04	1,1e-03	2,9e-03
atanhbx	1,3e-02	1,1e-03	1,6e-03	5,4e-04	7,9e-04	9,8e-04	8,5e-04	7,2e-04	4,7e-04	1,3e-03	<b>2,1e-04</b>
<b><math>\alpha = [-998, -98, -8]</math></b>											
	4	5	6	7	8	9	10	11	12	13	14
tanh	2e-04	1,9e-04	1,5e-04	2,7e-04	1,8e-04	1,6e-03	<b>1,3e-04</b>	1,9e-03	1,7e-03	7,7e-03	1,2e-01
sigmoid	1,7e-04	8,3e-05	7,4e-05	1,5e-04	<b>5,4e-05</b>	3,4e-04	1,4e-03	3,2e-03	6,8e-03	1,2e-02	1,3e-01
atanhbx	1e-04	1,6e-04	1,6e-04	1,4e-04	8,3e-05	<b>7,9e-05</b>	9,7e-03	1,5e-01	1,8e-03	5,2e-02	1,4e-01
<b><math>\alpha = [-9998, -998, -98, -8]</math></b>											
	4	5	6	7	8	9	10	11	12	13	14
tanh	4,5e-04	6,8e-04	5,2e-04	<b>4,3e-04</b>	1,5e-03	2,2e-03	1,1e-03	1,9e-02	1,3e-02	6,8e-02	2,2e-03
sigmoid	4,6e-04	<b>3,1e-04</b>	1e-03	9e-04	9,9e-04	3,8e-03	1,4e-03	1,2e-02	1,5e-02	4,7e-02	6,4e-02
atanhbx	9,6e-04	<b>4,4e-04</b>	1,3e-03	7,4e-04	6,8e-03	1,2e-02	2,4e-02	3e-02	3e-02	4,1e-02	1,5e-02

Πίνακας Β.8: Τιμές σφαλμάτων ΤΝΔ για το Π.Α.Τ. 5 & 10

<b>Πρόβλημα Αρχικών Τιμών 6</b>	
<i><math>\alpha</math> : σταθερή τιμή παραμέτρου</i>	<i>Βέλτιστη τιμή σφάλματος</i>
-8	1.8 e-07 (14)
-98	1.3 e-06 (13)
-998	8.6 e-07 (14)
-9998	9.4 e-05 (13)
<i><math>\alpha</math> : διάστημα τιμών</i>	<i>Βέλτιστη τιμή σφάλματος</i>
<i>[-998, -98, -8]</i>	4.1e-06 (6)
<i>[-9998, -998, -98, -8]</i>	1.9e-05 (4)

Πίνακας Β.9: Βέλτιστες τιμές σφαλμάτων ΤΝΔ για το Π.Α.Τ. 6 & 11

<b>6ο Πρόβλημα</b>											
$\alpha = -8$											
	4	5	6	7	8	9	10	11	12	13	14
tanh	1e-06	1,2e-06	1,3e-06	1,4e-06	1,3e-06	3,9e-07	7,2e-07	3,5e-07	<b>2,1e-07</b>	4,4e-07	4,9e-07
sigmoid	1,1e-06	1,2e-06	7,1e-07	2,6e-07	2,9e-07	2,9e-07	3,2e-07	<b>2,1e-07</b>	4,2e-07	4,9e-07	6,3e-07
atanhbx	9,9e-07	1,2e-06	1,2e-06	1e-06	7,2e-07	5,8e-07	2,1e-07	4,2e-07	3,1e-07	4,4e-07	<b>1,8e-07</b>
$\alpha = -98$											
	4	5	6	7	8	9	10	11	12	13	14
tanh	5,9e-06	3,4e-06	6,6e-06	5e-06	3,5e-06	5,6e-06	8,4e-06	8,2e-06	7,1e-06	1,4e-05	<b>2,8e-06</b>
sigmoid	4,8e-06	2,3e-06	4,1e-06	8,7e-06	<b>2,2e-06</b>	4,7e-06	5,4e-06	3,3e-06	4,9e-06	7,1e-06	2,9e-06
atanhbx	1,2e-05	5,7e-06	3,6e-06	3,9e-06	4,9e-06	4,9e-06	2,5e-06	6,7e-06	1,7e-05	<b>1,3e-06</b>	1,2e-05
$\alpha = -998$											
	4	5	6	7	8	9	10	11	12	13	14
tanh	3,7e-05	4,1e-05	6,7e-05	3,4e-05	8,7e-05	6e-05	<b>1e-05</b>	3,1e-05	2,4e-05	3,5e-05	2,5e-04
sigmoid	1,2e-05	1,2e-05	4,9e-05	<b>7,3e-06</b>	8,3e-05	3,5e-05	7,1e-05	9,9e-05	8,9e-05	1,1e-04	3e-05
atanhbx	6,3e-05	4,1e-05	7,5e-05	3,7e-05	4,9e-05	1,5e-05	1,5e-04	1,6e-04	2,7e-05	3,7e-05	<b>8,6e-07</b>
$\alpha = -9998$											
	4	5	6	7	8	9	10	11	12	13	14
tanh	3,4e-03	<b>1,8e-04</b>	3,9e-04	6,8e-04	4,4e-04	4,7e-04	1,2e-03	5,7e-04	3,2e-03	7,1e-04	3,9e-04
sigmoid	4,2e-04	4,3e-04	5,2e-04	9,3e-04	5,5e-04	6,4e-04	2,8e-04	1,6e-03	<b>2,2e-04</b>	3,1e-03	1,6e-03
atanhbx	3,4e-04	1,1e-03	7,5e-04	7,4e-04	2,4e-04	2,5e-04	1,5e-03	1,9e-04	5,1e-04	<b>9,4e-05</b>	9,7e-04
$\alpha = [-998, -98, -8]$											
	4	5	6	7	8	9	10	11	12	13	14
tanh	5,2e-06	6,7e-06	<b>4,1e-06</b>	5e-06	6,8e-05	4,8e-04	1,3e-03	5,4e-04	2,1e-03	6,3e-03	1,7e-01
sigmoid	5,2e-06	<b>4,5e-06</b>	5,6e-06	5,2e-06	2,7e-05	4,7e-04	3,5e-04	3,5e-03	7,3e-04	1,1e-02	2,5e-02
atanhbx	<b>1,5e-05</b>	1,9e-05	3,6e-05	1,1e-04	7,5e-04	4,9e-03	1,3e-02	3,9e-02	3,7e-02	2,7e-01	2,2e-02
$\alpha = [-9998, -998, -98, -8]$											
	4	5	6	7	8	9	10	11	12	13	14
tanh	<b>1,9e-05</b>	1,3e-04	6,9e-05	3,8e-05	1,2e-04	5,9e-05	3,2e-04	1,4e-02	8,8e-03	1,2e-02	1,6e-01
sigmoid	4,3e-05	<b>2,1e-05</b>	2e-04	3,2e-04	1,3e-04	1e-03	6,3e-04	1,5e-02	6,8e-03	6,9e-03	2,9e-02
atanhbx	7,7e-05	1,5e-04	<b>4,2e-05</b>	3,4e-04	5,4e-03	1,1e-02	2,7e-03	1,2e-02	1,7e-02	7,9e-03	6,1e-02

Πίνακας Β.10: Τιμές σφαλμάτων ΤΝΔ για το Π.Α.Τ. 6 & 11

<b>Πρόβλημα Αρχικών Τιμών 7</b>	
$\alpha$ : σταθερή τιμή παραμέτρου	Βέλτιστη τιμή σφάλματος
0	1.3e-06 (12)
1	2.9 e-07 (12)
2	6.8e-07 (9)
$\alpha$ : διάστημα τιμών	Βέλτιστη τιμή σφάλματος
[0,1,2]	3.8e-04 (8)

Πίνακας Β.11: Βέλτιστες τιμές σφαλμάτων ΤΝΔ για το Π.Α.Τ. 7

**Π.Α.Τ. 7**

$\alpha = 0$											
	4	5	6	7	8	9	10	11	12	13	14
tanh	8,9e-05	2,3e-05	1,7e-05	3,7e-05	1e-05	2,3e-05	<b>3,7e-06</b>	9,3e-06	4,1e-06	1,2e-05	1,7e-05
sigmoid	8,6e-05	5,5e-05	4,3e-05	4,7e-05	3,8e-05	3,8e-05	2,9e-05	2,7e-05	5,4e-05	<b>2,5e-05</b>	7,2e-05
atanhbx	1e-04	4,4e-05	6,8e-06	1,3e-05	1,4e-05	1,3e-05	3,5e-05	2,4e-05	<b>1,3e-06</b>	3,6e-05	6,5e-06
$\alpha=1$											
	4	5	6	7	8	9	10	11	12	13	14
tanh	1,6e-04	4,2e-05	3,3e-05	3,6e-06	7,8e-06	7,4e-06	9,5e-06	3e-06	<b>2,9e-07</b>	4,4e-06	9,5e-06
sigmoid	1,6e-04	6,1e-05	5,7e-05	3,4e-05	7e-05	2,4e-05	1,6e-05	1,3e-05	1,9e-05	3,1e-05	<b>1,1e-05</b>
atanhbx	2e-04	4,6e-05	2,5e-05	3,4e-05	8,2e-06	2e-05	1,4e-06	1,9e-05	<b>7,8e-07</b>	8,7e-07	1,4e-05
$\alpha=2$											
	4	5	6	7	8	9	10	11	12	13	14
tanh	1,6e-04	6,9e-05	2,6e-05	5,3e-05	1,5e-05	<b>6,8e-07</b>	4e-06	1,4e-05	8,4e-07	1,1e-06	1,6e-05
sigmoid	1,6e-04	1,2e-04	6,5 e-05	1 e-05	4,1 e-05	1,9 e-05	2,9 e-05	1,6 e-05	4,2 e-05	3,5 e-05	<b>6,2e-06</b>
atanhbx	2,7e-04	7,8e-05	4,9e-05	8,9e-06	3,1e-06	2,4 e-05	1,3e-05	1e-06	1,5e-05	4,6e-06	5,6e-06
$\alpha=[0, 1, 2]$											
	4	5	6	7	8	9	10	11	12	13	14
tanh	1,5e-02	1,2e-03	1,4e-03	3,2e-03	<b>3,8e-04</b>	4,1e-03	1e-03	2,6e-03	1,2e-03	1,9e-03	1,9e-03
sigmoid	1,5e-02	1,6e-03	1,6e-03	1,6e-03	3,5e-03	7,5e-03	4,3e-03	3,4e-03	3,8e-03	<b>1e-03</b>	2,6e-03
atanhbx	1,5e-02	<b>1,3e-03</b>	1,4e-03	1,6e-03	2e-03	1,7e-03	2,5e-03	4e-03	3,3e-03	1,8e-03	2e-03

Πίνακας Β.12: Τιμές σφαλμάτων ΤΝΔ για το Π.Α.Τ. 7



## ΠΑΡΑΡΤΗΜΑ Γ

Στους Πίνακες Γ.1-Γ.10 διακρίνονται για κάθε τιμή της παραμέτρου  $\alpha$  σε κάθε Π.Α.Τ. αν υπάρχει overfitting.

<b>Π.Α.Τ. 3</b>		
<b>Max abs error in all data</b>	<b>Max abs error in training data</b>	<b>Mean sqr error in all data</b>
1,46E-06	1,46E-06	6,98E-07
6,41E-06	6,40E-06	6,70E-06
3,66E-04	3,47E-04	5,94E-04
4,80E-03	4,77E-03	2,43E-03

Πίνακας Γ.1 : Overfitting για το Π.Α.Τ.3

<b>Π.Α.Τ. 4</b>		
<b>Max abs error in all data</b>	<b>Max abs error in training data</b>	<b>Mean sqr error in all data</b>
1,80E-05	1,80E-05	3,50E-06
6,60E-06	6,00E-06	4,90E-06
7,10E-05	7,10E-05	1,80E-05
4,50E-04	4,50E-04	6,40E-04

Πίνακας Γ.2 : Overfitting για το Π.Α.Τ.4

<b>Π.Α.Τ. 5</b>		
<b>Max abs error in all data</b>	<b>Max abs error in training data</b>	<b>Mean sqr error in all data</b>
4,00E-07	4,00E-07	3,30E-07
2,50E-06	2,40E-06	1,30E-06
2,60E-05	2,40E-05	7,10E-06
1,40E-04	1,40E-04	5,90E-05

Πίνακας Γ.3 : Overfitting για το Π.Α.Τ.5

<b>Π.Α.Τ. 6</b>		
<b>Max abs error in all data</b>	<b>Max abs error in training data</b>	<b>Mean sqr error in all data</b>
1,80E-07	1,70E-07	8,80E-08
1,30E-06	1,30E-06	5,50E-07
8,60E-07	8,30E-07	1,40E-06
9,40E-05	9,10E-05	4,10E-05

Πίνακας Γ.4 : Overfitting για το Π.Α.Τ.6

<b>Π.Α.Τ. 7</b>		
<b>Max abs error in all data</b>	<b>Max abs error in training data</b>	<b>Mean sqr error in all data</b>
1,30E-06	1,30E-06	6,34E-07
2,87E-07	2,74E-07	2,52E-07
6,82E-07	6,77E-07	5,80E-07

Πίνακας Γ.5 : Overfitting για το Π.Α.Τ.7

<b>Π.Α.Τ. 8</b>		
<b>Max abs error in all data</b>	<b>Max abs error in training data</b>	<b>Mean sqr error in all data</b>
1,23E-04	1,21E-04	2,22E-05
5,69E-04	5,64E-04	3,70E-04

Πίνακας Γ.6 : Overfitting για το Π.Α.Τ.8

<b>Π.Α.Τ. 9</b>		
<b>Max abs error in all data</b>	<b>Max abs error in training data</b>	<b>Mean sqr error in all data</b>
3,50E-05	3,50E-05	9,10E-06
6,48E-05	6,43E-05	6,41E-05

Πίνακας Γ.7 : Overfitting για το Π.Α.Τ.9

<b>Π.Α.Τ. 10</b>		
<b>Max abs error in all data</b>	<b>Max abs error in training data</b>	<b>Mean sqr error in all data</b>
5,40E-05	5,30E-05	1,30E-05
3,15E-04	3,10E-04	5,50E-05

Πίνακας Γ.8 : Overfitting για το Π.Α.Τ.10

<b>Π.Α.Τ. 11</b>		
<b>Max abs error in all data</b>	<b>Max abs error in training data</b>	<b>Mean sqr error in all data</b>
4,10E-06	4,10E-06	1,60E-06
1,90E-05	1,90E-05	1,70E-05

Πίνακας Γ.9 : Overfitting για το Π.Α.Τ.11

<b>Π.Α.Τ. 12</b>		
<b>Max abs error in all data</b>	<b>Max abs error in training data</b>	<b>Mean sqr error in all data</b>
3,83E-04	3,82E-04	1,75E-04

Πίνακας Γ.10 : Overfitting για το Π.Α.Τ.12

## ΠΑΡΑΡΤΗΜΑ Δ

---

Ενδεικτικοί κώδικες για επίλυση παραμετροποιημένων προβλημάτων.

### Κώδικας για επίλυση Π.Α.Τ:

```
clear all;
format long e;

tic

%SET THE PARAMETERS
vima=10;
epochs= 30000;
activation_function=3;
r=40; % # epanalhpsewn
training_points=30;
training_points_2=1;
neur_layer=1;
bound=1;
% _____

sfalma(1:r,1:neur_layer)=zeros;
min_res=inf;

%***** Set the problem *****
x0=0; %arxh euroys timwn pou ginetai h ekpaideysh
xf=3;
a0=-98;
af=-98;
y1_sol=@(x) 2*exp(-x)+exp(x);
y2_sol=@(x) 2*exp(-x)+x;
y01=y1_sol(0);
y02=y2_sol(0);
f1=@(x,a,y1,y2) -2*y1+y2+3*exp(x)-x;
f2=@(x,a,y1,y2) -a.*y1+(a-1).*y2+a.*exp(x)-(a-1).*x+1;

%***** Set the network *****
nepchs=epochs;
x_space=linspace(x0,xf,training_points);
a_space=[-98];
x_space2=kron(ones(1,length(a_space)'),'x_space');
a_space2=kron(a_space',ones(length(x_space),1));

%***** activation function *****
% _____tanh_____
if activation_function==1
activation=@(x) (exp(2.*x)-1)/(1+exp(2.*x));
activation_grad=@(x) 4.*exp(2.*x)/((exp(2.*x)+1).^2);

% _____sigmoid_____
elseif activation_function==2
activation=@(x) 1./(1+exp(-x));
activation_grad=@(x) exp(x)/(1+exp(x)).^2;

% _____1.7159tan(2/3x)_____
elseif activation_function==3
activation=@(x) 1.7159*(exp((4/3).*x)-1)/(1+exp((4/3).*x));
activation_grad=@(x) (1.7159*8/3)*(exp((4/3).*x)/((exp((4/3).*x)+1).^2));

for i=1:r;
```

```

% _____ 2 inputs (x,a) _____
neural_network=@(W1,W2,V,U,U1,x,a)
(V'*activation(W1*x'+(W2*(a'/10))+U*ones(1,length(x)))+U1)';
d_neural_network_dx=@(W1,W2,V,U,x,a)
(transpose((W1.*V))*activation_grad(W1*x'+(W2*(a'/10))+U*ones(1,length(x))))';

nneterr=@(P)[neural_network(P(1:neurons_used),P(neurons_used+1:2*neurons_used),P(3
*neurons_used+1:4*neurons_used),P(2*neurons_used+1:3*neurons_used),P(5*neurons_used
+1),x_space2,a_space2)+...
(x_space2-
x0).*d_neural_network_dx(P(1:neurons_used),P(neurons_used+1:2*neurons_used),P(3*neu
rons_used+1:4*neurons_used),P(2*neurons_used+1:3*neurons_used),x_space2,a_space2)-.
...
f1(x_space2,a_space2,y01+(x_space2-
x0).*neural_network(P(1:neurons_used),P(neurons_used+1:2*neurons_used),P(3*neurons_
used+1:4*neurons_used),P(2*neurons_used+1:3*neurons_used),P(5*neurons_used+1),x_spa
ce2,a_space2),...
y02+(x_space2-
x0).*neural_network(P(1:neurons_used),P(neurons_used+1:2*neurons_used),P(4*neurons_
used+1:5*neurons_used),P(2*neurons_used+1:3*neurons_used),P(5*neurons_used+2),x_spa
ce2,a_space2)];

neural_network(P(1:neurons_used),P(neurons_used+1:2*neurons_used),P(4*neurons_used+
1:5*neurons_used),P(2*neurons_used+1:3*neurons_used),P(5*neurons_used+2),x_space2,a
_space2)+...
(x_space2-
x0).*d_neural_network_dx(P(1:neurons_used),P(neurons_used+1:2*neurons_used),P(4*neu
rons_used+1:5*neurons_used),P(2*neurons_used+1:3*neurons_used),x_space2,a_space2)-.
...
f2(x_space2,a_space2,y01+(x_space2-
x0).*neural_network(P(1:neurons_used),P(neurons_used+1:2*neurons_used),P(3*neurons_
used+1:4*neurons_used),P(2*neurons_used+1:3*neurons_used),P(5*neurons_used+1),x_spa
ce2,a_space2),...
y02+(x_space2-
x0).*neural_network(P(1:neurons_used),P(neurons_used+1:2*neurons_used),P(4*neurons_
used+1:5*neurons_used),P(2*neurons_used+1:3*neurons_used),P(5*neurons_used+2),x_spa
ce2,a_space2)]];
%
%Training
rng('shuffle')
W1=-bound+2*bound*rand(neurons_used,1);
W2=-bound+2*bound*rand(neurons_used,1);
V1=-bound+2*bound*rand(neurons_used,1);
U=-bound+2*bound*rand(neurons_used,1);
V2=-bound+2*bound*rand(neurons_used,1);
U1=-bound+2*bound*rand;
U2=-bound+2*bound*rand;

P0=[W1;W2;U;V1;V2;U1;U2];
options = optimoptions('lsqnonlin','Jacobian','off','Algorithm','levenberg-
marquardt','InitDamping',.001,'TolX',1e-16,'TolFun',1e-
16,'MaxFunEvals',200000,'MaxIter',nepchs);
[parsol,resnorm,residual]=lsqnonlin(nneterr,P0,[],[],options);

% ***** evaluate the nnet at more points than the one trained
x_space3=linspace(x0,xf,5*training_points);
a_space3=linspace(a0,af,10*training_points_2);
x_space1=kron(ones(1,length(a_space3)),x_space3);
a_space1=kron(a_space3,ones(length(x_space3),1));
W1=parsol(1:neurons_used);
W2=parsol(neurons_used+1:2*neurons_used);
U=parsol(2*neurons_used+1:3*neurons_used);
V1=parsol(3*neurons_used+1:4*neurons_used);
V2=parsol(4*neurons_used+1:5*neurons_used);
U1=parsol(5*neurons_used+1);

```

```

U2=parsol(5*neurons_used+2);
y1_space=y1_sol(x_space2);
y2_space=y2_sol(x_space2);
y1_space1=y1_sol(x_space1);
y2_space1=y2_sol(x_space1);

% __system-(x,a) eisodoi
res1=y01+(x_space1-x0).*(neural_network(W1,W2,V1,U,U1,x_space1,a_space1));
res2=y02+(x_space1-x0).*neural_network(W1,W2,V2,U,U2,x_space1,a_space1);

sfalma_system(i,neurons_used)=max(max(abs(y1_space1-res1)),max(abs(y2_space1-
res2)));

if sfalma_system(i,neurons_used)<min_res,
    min_res=sfalma_system(i,neurons_used);
    W1_best=W1;
    W2_best=W2;
    V1_best=V1;
    U_best=U;
    U1_best=U1;
    U2_best=U2;
    V2_best=V2;

    x_space_best=x_space;
    x_space1_best=x_space1;
    y_space1_best=y1_space1;
    y_space2_best=y2_space1;
    neurons_best=neurons_used;
end
end
end
toc
accuracy_system=min(sfalma_system(:,6:6));
best_accuracy=min(accuracy_system)
save('parametric_a_pinakas_lo_problhma_[-98]_tanhabx')

```

### Κώδικας για εύρεση σφαλμάτων από τις υπόλοιπες προσεγγιστικές μεθόδους.

```

clear all
format long e
opts = odeset('Stats','on');
method=7;
tspan=[0 3];
a=-8;
x0=0;
xf=3;
y1_sol=@(x)-2*exp(-x);
y2_sol=@(x)-2*exp(-x)+cos(x);
y01=y1_sol(0); %arxikes sun8hkes
y02=y2_sol(0);
y0s=[y01;y02];
fs=@(x,y)[-2*y(1)+y(2)-cos(x);-a.*y(1)+(a-1).*y(2)-sin(x)-a.*cos(x)+cos(x)];
if method==1;
    disp('ode23s')
    [t,ynum]=ode23s(fs,tspan,y0s,opts);
    plot(t,ynum)
    errorODE_1=max(abs(ynum(:,1)-y1_sol(t)));
    errorODE_2=max(abs(ynum(:,2)-y2_sol(t)));
    errorODE_mat=[errorODE_1,errorODE_2];
    errorODE=max(errorODE_mat)
elseif method==2;

```

```

        disp('ode23')
[t,ynum] =ode23( fs, tspan, y0s,opts);
plot(t,ynum)
    errorODE_1=max(abs(ynum(:,1)-y1_sol(t)));
    errorODE_2=max(abs(ynum(:,2)-y2_sol(t)));
    errorODE_mat=[errorODE_1,errorODE_2];
    errorODE=max(errorODE_mat)
    %
    elseif method==3;
        disp('ode45')
[t,ynum] =ode45( fs, tspan, y0s,opts);
plot(t,ynum)
    errorODE_1=max(abs(ynum(:,1)-y1_sol(t)));
    errorODE_2=max(abs(ynum(:,2)-y2_sol(t)));
    errorODE_mat=[errorODE_1,errorODE_2];
    errorODE=max(errorODE_mat)
    elseif method==4;
        disp('ode15s')
[t,ynum] =ode15s( fs, tspan, y0s,opts);
plot(t,ynum)
    errorODE_1=max(abs(ynum(:,1)-y1_sol(t)));
    errorODE_2=max(abs(ynum(:,2)-y2_sol(t)));
    errorODE_mat=[errorODE_1,errorODE_2];
    errorODE=max(errorODE_mat)
    elseif method==5;
        disp('ode23t')
[t,ynum] =ode23t( fs, tspan, y0s,opts);
plot(t,ynum)
    errorODE_1=max(abs(ynum(:,1)-y1_sol(t)));
    errorODE_2=max(abs(ynum(:,2)-y2_sol(t)));
    errorODE_mat=[errorODE_1,errorODE_2];
    errorODE=max(errorODE_mat)
    elseif method==6;
        disp('ode23tb')
[t,ynum] =ode23tb( fs, tspan, y0s,opts);
plot(t,ynum)
    errorODE_1=max(abs(ynum(:,1)-y1_sol(t)));
    errorODE_2=max(abs(ynum(:,2)-y2_sol(t)));
    errorODE_mat=[errorODE_1,errorODE_2];
    errorODE=max(errorODE_mat)

else
    disp('radau')
    opts.Complex='off';
[t,ynum] = radau5( 'lams2p1_k2_0', tspan, y0s,opts,1);
plot(t,ynum)
    errorODE_1=max(abs(ynum(:,1)-y1_sol(t)));
    errorODE_2=max(abs(ynum(:,2)-y2_sol(t)));
    errorODE_mat=[errorODE_1,errorODE_2];
    errorODE=max(errorODE_mat)
end

```

## **Κώδικας για υπολογισμό overfitting**

```

clear all
load('parametric_a_pinakas_lo_problhma_[-998]_sigmoid.mat', 'W1_best', 'W2_best', 'V1_best', 'V2_best', 'U_best', 'U1_best', 'U2_best')
load('parametric_a_pinakas_lo_problhma_[-998]_sigmoid.mat', 'neurons_best')
W1=W1_best;
W2=W2_best;
V1=V1_best;
U=U_best;
U1=U1_best;

```

```

U2=U2_best;
V2=V2_best;
P=[W1;W2;U;V1;V2;U1;U2];
neurons_used=neurons_best;
vima=10;
epochs= 30000;
activation_function=2;
r=40;
training_points=30;
training_points_2=1;
neur_layer=1;
bound=1;
sfalma(1:r,1:neur_layer)=zeros;
min_res=inf;

x0=0;
xf=3;
a0=-998;
af=-998;
y1_sol=@(x) 2*exp(-x)+exp(x);
y2_sol=@(x) 2*exp(-x)+x;
y01=y1_sol(0);
y02=y2_sol(0);
f1=@(x,a,y1,y2) -2*y1+y2+3*exp(x)-x;
f2=@(x,a,y1,y2) -a.*y1+(a-1).*y2+a.*exp(x)-(a-1).*x+1;
nepchs=epochs;
x_space=linspace(x0,xf,training_points);
a_space=[-998];
x_space2=kron(ones(1,length(a_space)),x_space);
a_space2=kron(a_space,ones(length(x_space),1));
%_____simplified tanh_____
if activation_function==1
activation=@(x) (exp(2.*x)-1)/(1+exp(2.*x));
activation_grad=@(x) 4.*exp(2.*x)/((exp(2.*x)+1).^2);

%_____sigmoid_____
elseif activation_function==2
activation=@(x) 1./(1+exp(-x));
activation_grad=@(x) exp(x)/(1+exp(x)).^2;

%_____1.7159tan(2/3x)_____
elseif activation_function==3
activation=@(x) 1.7159*(exp((4/3).*x)-1)/(1+exp((4/3).*x));
activation_grad=@(x) (1.7159*8/3)*(exp((4/3).*x)/((exp((4/3).*x)+1).^2));

%_____smooth RElu_____
else
activation=@(x) log(1+exp(x));
activation_grad=@(x) 1./(1+exp(-x));

end
neural_network =@(W1,W2,V,U,U1,x,a)
(V'*activation(W1*x'+(W2*(a'/100))+U*ones(1,length(x)))+U1)';
d_neural_network_dx=@(W1,W2,V,U,x,a)
(transpose((W1.*V))*activation_grad(W1*x'+(W2*(a'/100))+U*ones(1,length(x))))';

nneterr=@(P) [ neural_network(P(1:neurons_used),P(neurons_used+1:2*neurons_used),P(3
*neurons_used+1:4*neurons_used),P(2*neurons_used+1:3*neurons_used),P(5*neurons_used
+1),x_space2,a_space2)+...
(x_space2-
x0).*d_neural_network_dx(P(1:neurons_used),P(neurons_used+1:2*neurons_used),P(3*neu
rons_used+1:4*neurons_used),P(2*neurons_used+1:3*neurons_used),x_space2,a_space2)-.
...
f1(x_space2,a_space2,y01+(x_space2-
x0).*neural_network(P(1:neurons_used),P(neurons_used+1:2*neurons_used),P(3*neurons_

```

```

used+1:4*neurons_used),P(2*neurons_used+1:3*neurons_used),P(5*neurons_used+1),x_spa
ce2,a_space2),...
    y02+(x_space2-
x0).*neural_network(P(1:neurons_used),P(neurons_used+1:2*neurons_used),P(4*neurons_
used+1:5*neurons_used),P(2*neurons_used+1:3*neurons_used),P(5*neurons_used+2),x_spa
ce2,a_space2));

neural_network(P(1:neurons_used),P(neurons_used+1:2*neurons_used),P(4*neurons_used+
1:5*neurons_used),P(2*neurons_used+1:3*neurons_used),P(5*neurons_used+2),x_space2,a
_space2)+...
    (x_space2-
x0).*d_neural_network_dx(P(1:neurons_used),P(neurons_used+1:2*neurons_used),P(4*neu
rons_used+1:5*neurons_used),P(2*neurons_used+1:3*neurons_used),x_space2,a_space2)-.
..
    f2(x_space2,a_space2,y01+(x_space2-
x0).*neural_network(P(1:neurons_used),P(neurons_used+1:2*neurons_used),P(3*neurons_
used+1:4*neurons_used),P(2*neurons_used+1:3*neurons_used),P(5*neurons_used+1),x_spa
ce2,a_space2),...
    y02+(x_space2-
x0).*neural_network(P(1:neurons_used),P(neurons_used+1:2*neurons_used),P(4*neurons_
used+1:5*neurons_used),P(2*neurons_used+1:3*neurons_used),P(5*neurons_used+2),x_spa
ce2,a_space2)]);
%
x_space3=linspace(x0,xf,5*training_points);
a_space3=linspace(a0,af,10*training_points_2);
x_space1=kron(ones(1,length(a_space3))',x_space3');
a_space1=kron(a_space3',ones(length(x_space3),1));
y1_space=y1_sol(x_space2);
y2_space=y2_sol(x_space2);
y1_space1=y1_sol(x_space1);
y2_space1=y2_sol(x_space1);
%xspace1 4500 x1 aspace 1 4500x1
res1=y01+(x_space1-x0).*neural_network(W1,W2,V1,U,U1,x_space1,a_space1);
res2=y02+(x_space1-x0).*neural_network(W1,W2,V2,U,U2,x_space1,a_space1);
disp('max abs error in all data')
max(max(abs(y1_space1-res1)),max(abs(y2_space1-res2)))

% xspace2 30x1 aspace2 30x1
res01=y01+(x_space2-x0).*neural_network(W1,W2,V1,U,U1,x_space2,a_space2));
res02=y02+(x_space2-x0).*neural_network(W1,W2,V2,U,U2,x_space2,a_space2);
disp('max abs error in training data')
max(max(abs(y1_space-res01)),max(abs(y2_space-res02)))
disp('mean sqr error in all data')
sqrt(sum(nneterr(P).^2)/numel(nneterr(P)))

```