



Εθνικό Μετσόβιο Πολυτεχνείο  
Σχολή Μηχανολόγων Μηχανικών  
Τομέας Ρευστών  
Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής  
& Βελτιστοποίησης

Ένταξη CAD στην Αεροδυναμική Βελτιστοποίηση  
Μορφής & Χρήση Βαθιών Νευρωνικών Δικτύων για  
Προβλέψεις Ροών

Διπλωματική Εργασία

Γεώργιος Τσάμης

Επιβλέπων: Κυριάκος Χ. Γιαννάκογλου, Καθηγητής ΕΜΠ

Αθήνα, 2020



## Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω τον καθηγητή μου κ. Κυριάκο Χ. Γιαννάκογλου για το χρόνο που διέθεσε και την καθοδήγηση που μου παρείχε, παρά το βεβαρυμένο πρόγραμμά του.

Επίσης, θα ήθελα να ευχαριστήσω το προσωπικό της ΜΠΤΡΒ, της σχολής Μηχανολόγων Μηχανικών του ΕΜΠ, και συγκεκριμένα την Μαρίνα Κοντού για την πολύτιμη βοήθειά της, όπως επίσης και τη Δρ. Βαρβάρα Ασούτη για την επίλυση διάφορων τεχνικών δυσκολιών.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου, που πάντα είναι εκεί για να με στηρίζουν και τους φίλους μου, που στάθηκαν πλάι μου από την αρχή της φοίτησής μου μέχρι και το τέλος.



Εθνικό Μετσόβιο Πολυτεχνείο  
Σχολή Μηχανολόγων Μηχανικών  
Τομέας Ρευστών  
Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής  
& Βελτιστοποίησης

## Ένταξη CAD στην Αεροδυναμική Βελτιστοποίηση Μορφής & Χρήση Βαθιών Νευρωνικών Δικτύων για Προβλέψεις Ροών

Διπλωματική Εργασία

Γεώργιος Τσάμης

Επιβλέπων: Κυριάκος Χ. Γιαννάκογλου, Καθηγητής ΕΜΠ

Αθήνα, 2020

Περίληψη

Σκοπός της εργασίας αυτής είναι η υλοποίηση μιας μεθοδολογίας για την ένταξη ενός λογισμικού CAD σε διαδικασίες βελτιστοποίησης μορφής μέσω στοχαστικών μεθόδων ανίχνευσης, στην υπολογιστική ρευστοδυναμική, αλλά και η χρήση βαθιών νευρωνικών δικτύων (Deep Neural Networks, DNN) για την πρόβλεψη ροών ως υποκατάστατο λογισμικού υπολογιστικής ρευστοδυναμικής. Η διπλωματική αυτή εργασία, δομείται έχοντας επιλέξει ως εφαρμογή τη βελτιστοποίηση της γεωμετρίας ενός αγωγού και παρουσιάζει τη σχεδίαση του αρχικού σχήματός του στο εμπορικό πακέτο CAD Solidworks, την παραμετροποίηση του σχήματός του, την αυτοματοποιημένη γένεση πλέγματος στον πλεγματοποιητή Pointwise με χρήση script και μια εκτενή μελέτη της προσομοίωσης της ροής στο εσωτερικό του. Με βάση τα παραπάνω, πραγματοποιείται η βελτιστοποίηση του σχήματός του έχοντας ως συνάρτηση-στόχο τις απώλειες ολικής πίεσης, με τη βοήθεια ενός εξελικτικού αλγορίθμου που περιέχεται στο λογισμικό βελτιστοποίησης EASY. Ως λογισμικό αξιολόγησης χρησιμοποιείται ο οικείος υψηλής ακρίβειας επιλύτης υπολογιστικής ρευστοδυναμικής – CFD PUMA, ο οποίος μαζί με το λογισμικό EASY έχουν αναπτυχθεί στη ΜΠΥΡΒ της Σχολής Μηχανολόγων Μηχανικών του ΕΜΠ. Η μηχανική μάθηση και συγκεκριμένα τα νευρωνικά δίκτυα DNN, έχουν αποδειχθεί ικανά στην επίλυση σύνθετων μη-γραμμικών προβλημάτων, στα οποία περιλαμβάνονται προβλήματα ρευστοδυναμικής. Με κριτήριο το υψηλό υπολογιστικό κόστος της χρήσης λογισμικού CFD στο βρόχο της βελτιστοποίησης, προτείνεται η εκπαίδευση βαθιών νευρωνικών δικτύων (DNN), ύστερα από κατάλληλη ανάλυση, σε δεδομένα που έχουν ληφθεί από το λογισμικό CFD, με σκοπό την αντικατάστασή του εντός του βρόχου της βελτιστοποίησης. Με την προαναφερθείσα μέθοδο, μειώνεται σημαντικά ο απαιτούμενος χρόνος για τον εντοπισμό της βέλτιστης γεωμετρίας, ενώ οι απώλειες που εμφανίζονται στην ακρίβεια είναι επαρκώς μικρές και ελεγχόμενες.



**National Technical University of Athens**  
**School of Mechanical Engineering**  
**Fluids Department**  
**Parallel CFD & Optimization Unit**

## **CAD into the Aerodynamic Shape Optimization Loop & Use of Deep Neural Networks for Flow Predictions**

Diploma Thesis

**George Tsamis**

Advisor: Kyriakos C. Giannakoglou, Professor NTUA

Athens, 2020

Abstract

The purpose of this thesis is to incorporate the CAD software into the optimization loop, based on stochastic, population-based methods for shape optimization in the field of CFD, but also to implement the use of Deep Neural Networks (DNN) for flow prediction, as a CFD software substitute. Throughout this thesis, the shape optimization of a duct is used as a working example. This thesis includes the duct's initial design using CAD software Solidworks, its shape parameterization, its automatic mesh generation with scripts using Pointwise and an extensive study (including some parametric studies from the CFD point of view) for its flow simulation. Based on the above, duct's shape optimization targets the minimization of total pressure losses between its inlet and outlet and this is carried out using the optimization software EASY based on evolutionary algorithms. PUMA, a highly-accurate and computationally CFD solver, which, like EASY, has been developed by the PCOpt Unit of NTUA, is used as the evaluation software.

On the other hand, machine learning and neural networks in particular, which include ANN and DNN, have been proved to be able to solve complex non-linear problems, such as those in fluid dynamics. Given the high computational cost of the use of CFD software in the optimization loop, DNN training - after the proper analysis - using CFD data is proposed, in order to replace the CFD software in the optimization loop. Using the method mentioned above, the time required to find the optimal geometry is significantly reduced, while the loss in accuracy is small and controlled.

## Ακρωνύμια

ΕΜΠ	Εθνικό Μετσόβιο Πολυτεχνείο
ΕΘΣ	Εργαστήριο Θερμικών Στροβιλομηχανών
ΜΠΥΡΒ	Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής & Βελτιστοποίησης
ΥΡΔ	Υπολογιστική Ρευστοδυναμική

---

CFD	Computational Fluid Dynamics
NTUA	National Technical University of Athens
PCopt	Parallel CFD & Optimization unit
PUMA	<b>P</b> arallel solver for <b>U</b> nstructured grids for <b>M</b> ult-blade row computations including <b>A</b> djoint
EASY	<b>E</b> volutionary <b>A</b> lgorithms <b>S</b> ystem
EA	Evolutionary Algorithm
GA	Genetic Algorithm
MAEA	Metamodel-Assisted Evolutionary Algorithm
CAD	Computer-Aided Design
CAE	Computer-Aided Engineering
DNN	Deep Neural Network

# Περιεχόμενα

Περιεχόμενα	i
<b>1 Εισαγωγή</b>	<b>1</b>
1.1 Ένταξη Λογισμικού CAD στο Βρόχο Βελτιστοποίησης . . . . .	1
1.2 Νευρωνικά Δίκτυα . . . . .	2
1.3 Δομή της Εργασίας . . . . .	2
<b>2 Σχεδίαση Γεωμετρίας και Δημιουργία Πλέγματος</b>	<b>4</b>
2.1 Σχεδίαση Γεωμετρίας . . . . .	4
2.2 Γένεση Πλέγματος . . . . .	8
2.3 Αυτοματοποιημένη Παραγωγή Γεωμετρίας και Πλέγματος . . . . .	14
<b>3 Υπολογιστική Προσομοίωση της Ροής (CFD)</b>	<b>17</b>
3.1 Επίλυση της ροής . . . . .	17
3.2 Ανεξαρτησία Πλέγματος . . . . .	22
3.2.1 Μελέτη 1 . . . . .	22
3.2.2 Μελέτη 2 . . . . .	23
3.2.3 Μελέτη 3 . . . . .	25
3.2.4 Συμπεράσματα . . . . .	28
3.3 Σύγκριση Μοντέλων Τύρβης . . . . .	29
3.3.1 Προσδιορισμός της Έντασης της Τύρβης . . . . .	29
3.3.2 Σύγκριση Ολικών Πιέσεων . . . . .	31
3.3.3 Συμπεράσματα . . . . .	34
3.4 Επιλογή της Τελικής Μοντελοποίησης . . . . .	34

<b>4</b>	<b>Βελτιστοποίηση Γεωμετρίας του Αγωγού - MAEA &amp; CFD</b>	<b>35</b>
4.1	Περιγραφή της Περίπτωσης . . . . .	37
4.2	Αποτελέσματα και Τελική Λύση . . . . .	42
4.2.1	Αποτελέσματα . . . . .	42
4.2.2	Η ροή στον αγωγό της τελικής λύσης . . . . .	44
<b>5</b>	<b>Πρόβλεψη της Ροής με χρήση Νευρωνικών Δικτύων</b>	<b>47</b>
5.1	Τα βασικά των Νευρωνικών Δικτύων . . . . .	48
5.1.1	Η βασική λειτουργία των Νευρωνικών Δικτύων . . . . .	48
5.1.2	Συναρτήσεις ενεργοποίησης . . . . .	49
5.1.3	Εκπαίδευση . . . . .	50
5.1.4	Συναρτήσεις κόστους . . . . .	51
5.1.5	Οπισθοδιάδοση (Back-propagation) . . . . .	51
5.1.6	Gradient-Based Αλγόριθμοι Εκπαίδευσης . . . . .	53
5.1.7	Βάση Δεδομένων (Database) . . . . .	54
5.1.8	Κανονικοποίηση των Δεδομένων (Data Normalization) . . . . .	55
5.1.9	Υπερπαραμέτροι . . . . .	55
5.2	Υλοποίηση της εκπαίδευσης . . . . .	58
5.2.1	Σύνθεση Βάσης Δεδομένων . . . . .	58
5.2.2	Προετοιμασία των Δεδομένων . . . . .	60
5.3	Μεθοδολογία χρήσης DNN για την Πρόβλεψη της Ροής στον Αγωγό .	61
5.3.1	Α΄ Προσέγγιση . . . . .	62
5.3.2	Β΄ Προσέγγιση . . . . .	64
5.3.3	Γ΄ Προσέγγιση . . . . .	66
5.3.4	Σύγκριση των Προσεγγίσεων Α΄,Β΄ και Γ΄ . . . . .	68
5.4	Ανάλυση της επίδρασης των Υπερπαραμέτρων στην εκπαίδευση DNN .	71
5.4.1	Επίδραση της Αρχιτεκτονικής . . . . .	71
5.4.2	Επίδραση Dropout . . . . .	73
5.4.3	Επίδραση του Ρυθμού Μάθησης (Learning Rate) . . . . .	77
5.4.4	Επίδραση της Βάσης Δεδομένων . . . . .	78



5.5	Εκπαίδευση DNN και Πρόβλεψη της Ροής στον Αγωγό . . . . .	81
5.6	Βελτιστοποίηση Γεωμετρίας Αγωγού με χρήση EA & DNN . . . . .	82
5.6.1	Παρουσίαση της Περίπτωσης . . . . .	82
5.6.2	Υλοποίηση της διαδικασίας . . . . .	83
5.6.3	Αποτελέσματα . . . . .	83
5.6.4	Σύγκριση με χρήση λογισμικού CFD . . . . .	84
<b>6</b>	<b>Ανακεφαλαίωση - Συμπεράσματα</b>	<b>89</b>
6.1	Ανακεφαλαίωση . . . . .	89
6.2	Συμπεράσματα . . . . .	90
6.3	Προτάσεις για μελλοντική έρευνα . . . . .	91
	<b>A' Scripts</b>	<b>93</b>
A'.0.1	SwPw.bat . . . . .	93
A'.0.2	Preprocessor.py . . . . .	95
A'.0.3	main.bat . . . . .	97
A'.0.4	remote.bat (Χρησιμοποιώντας MobaXTerm) . . . . .	98
A'.0.5	eval.bat (Χρησιμοποιώντας MobaXTerm) . . . . .	99
A'.0.6	dnn.py (Χρησιμοποιώντας tensorflow) . . . . .	100
	<b>Βιβλιογραφία</b>	<b>103</b>



# Κεφάλαιο 1

## Εισαγωγή

### 1.1 Ένταξη Λογισμικού CAD στο Βρόχο Βελτιστοποίησης

Ο ταχύτατα αναπτυσσόμενος χώρος του σχεδιασμού που πραγματεύεται προβλήματα ρευστοδυναμικής, απαιτεί, πλέον, διαδικασίες βελτιστοποίησης μορφών, με σκοπό την εξαγωγή βελτιωμένων γεωμετριών, που επιτυγχάνουν καλύτερη απόδοση. Αναλόγως την εφαρμογή, οι γεωμετρίες συνήθως παραμετροποιούνται χρησιμοποιώντας καμπύλες Bezier, καμπύλες και επιφάνειες NURBS κ.ο.κ. Βέβαια, σε πολλές εφαρμογές, η γεωμετρία είναι δύσκολο να παραμετροποιηθεί χρησιμοποιώντας τα παραπάνω. Από την άλλη μεριά, τα ολοένα και πιο εξελιγμένα λογισμικά CAD, δίνουν τη δυνατότητα παραμετροποίησης των διαστάσεων του μοντέλου προς σχεδίαση. Συνεπώς, εύλογο είναι το κίνητρο που παρουσιάζεται για την παραμετροποίηση γεωμετριών εντός του λογισμικού CAD, το οποίο, εντάσσοντάς το στο βρόχο της βελτιστοποίησης, δύναται να παράγει τη γεωμετρία της εκάστοτε υποψήφιας λύσης. Αυτό, φυσικά, σημαίνει πως ο μηχανικός δεν έχει πρόσβαση στις μαθηματικές σχέσεις που περιγράφουν τη γεωμετρία, και συνεπώς δεν υπάρχει δυνατότητα εξαγωγής των απαιτούμενων παραγώγων για τη χρήση αιτιοκρατικών μεθόδων βελτιστοποίησης. Έτσι, στη Διπλωματική αυτή Εργασία, παρουσιάζεται μια μέθοδος υλοποίησης της ένταξης λογισμικού CAD στο βρόχο της βελτιστοποίησης μορφής, με χρήση στοχαστικών μεθόδων βελτιστοποίησης.

## 1.2 Νευρωνικά Δίκτυα

Η μηχανική μάθηση, υποπεδίο της επιστήμης των υπολογιστών, αναπτύχθηκε και χρησιμοποιείται κατά κύριο λόγο για την αναγνώριση προτύπων σε πολλούς τομείς. Τα τεχνητά νευρωνικά δίκτυα (Artificial Neural Networks - ANN), ένας αλγόριθμος μάθησης, εμπνεόμενα από τη βιολογική δομή των νευρώνων του εγκεφάλου, χρησιμοποιούνται σε ένα ευρύ φάσμα πεδίων και έχει αποδειχθεί η αποτελεσματικότητά τους για την επίλυση μη-γραμμικών προβλημάτων. Η πτώση των τιμών των υπολογιστών σε συνδυασμό με τη ραγδαία ανάπτυξη της υπολογιστικής δύναμης, οδήγησε στην ανάπτυξη της βαθιάς μάθησης (Deep Learning), που περιλαμβάνει τη χρήση βαθιών νευρωνικών δικτύων (Deep Neural Networks - DNN), τα οποία έχουν υψηλές δυνατότητες στην επίλυση σύνθετων προβλημάτων. Τα προβλήματα ρευστοδυναμικής, ως προβλήματα που διέπονται από σύνθετους μη-γραμμικούς νόμους, μπορούν να επιλυθούν χρησιμοποιώντας σύγχρονους επιλύτες CFD υψηλής ακρίβειας, όπως για παράδειγμα το λογισμικό PUMA, το οποίο αξιοποιεί κάρτες γραφικών GPU για την επίλυση της ροής. Το παραπάνω, δεν αναιρεί το γεγονός πως η επίλυση προβλημάτων ρευστοδυναμικής είναι μια διαδικασία χρονοβόρα. Έτσι, στην παρούσα εργασία, παρουσιάζεται ένας τρόπος εκπαίδευσης βαθιών νευρωνικών δικτύων για την πρόβλεψη της ροής με μικρά και ελεγχόμενα σφάλματα.

## 1.3 Δομή της Εργασίας

Η εργασία αυτή, παρουσιάζει την υλοποίηση μιας μεθόδου ένταξης του λογισμικού CAD Solidworks στο βρόχο της βελτιστοποίησης, η οποία πραγματοποιείται χρησιμοποιώντας εξελικτικούς αλγόριθμους. Η παραπάνω μεθοδολογία, βρίσκει εφαρμογή στη βελτιστοποίηση της μορφής ενός αγωγού που διακινεί αέρα, με συνάρτηση-στόχο τις απώλειες ολικής πίεσης. Έστερα από τα παραπάνω, παρουσιάζεται μια τεχνική της χρήσης βαθιών νευρωνικών δικτύων για την πρόβλεψη της ροής στον εν λόγω αγωγό, με σκοπό την αντικατάσταση του λογισμικού CFD με ένα εκπαιδευμένο δίκτυο DNN στο βρόχο της βελτιστοποίησης. Το κάθε κεφάλαιο πραγματεύεται τα εξής:

- Στο κεφάλαιο 2 παρουσιάζεται, αρχικά, η σχεδίαση του αγωγού μέσω του λογισμικού CAD Solidworks και ο τρόπος γένεσης πλέγματος στον πλεγματοποιητή Pointwise. Επίσης, παρουσιάζεται η τεχνική αυτόματης παραγωγής της παραμετροποιημένης γεωμετρίας και της γένεσης του πλέγματος.
- Στο κεφάλαιο 3 παρουσιάζεται μια εκτενής ανάλυση για την επίλυση της ροής εντός του αγωγού, χρησιμοποιώντας το υψηλής ακρίβειας και ταχύτητας λογισμικό CFD PUMA.
- Στο κεφάλαιο 4 παρουσιάζεται η υλοποίηση της βελτιστοποίησης σχήματος του αγωγού, με στόχο την ελαχιστοποίηση των απωλειών ολικής πίεσης. Η βελτιστοποίηση πραγματοποιείται με χρήση εξελικτικού αλγόριθμου που διαθέτει το

λογισμικό EASY, χρησιμοποιώντας ως λογισμικό αξιολόγησης τον επιλύτη CFD PUMA.

- Στο κεφάλαιο 5, ο αναγνώστης εισάγεται στην έννοια των νευρωνικών δικτύων, παρουσιάζοντας κάποια βασικά γνωρίσματά τους. Ύστερα, αναλύεται η διαδικασία της εκπαίδευσης βαθιών νευρωνικών δικτύων (DNN) σε δεδομένα από ροές που έχουν επιλυθεί χρησιμοποιώντας το λογισμικό CFD PUMA, καθώς και η ρύθμιση διαφόρων παραμέτρων των δικτύων, με σκοπό την επίτευξη χαμηλών σφαλμάτων. Τέλος, παρουσιάζεται η αποτελεσματικότητα ενός εκπαιδευμένου δικτύου DNN ως αντικαταστάτης του λογισμικού CFD στον βρόχο βελτιστοποίησης.
- Στο κεφάλαιο 6 πραγματοποιείται μια ανακεφαλαίωση της εργασίας και παρουσιάζονται τα τελικά συμπεράσματα.

# Κεφάλαιο 2

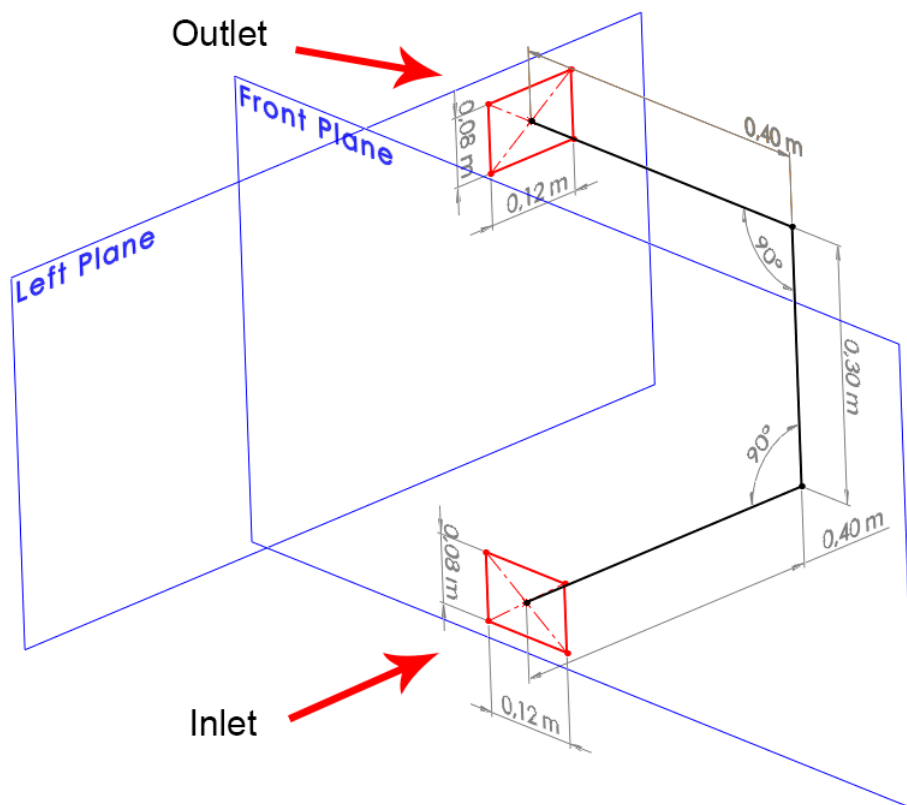
## Σχεδίαση Γεωμετρίας και Δημιουργία Πλέγματος

### 2.1 Σχεδίαση Γεωμετρίας

**Λογισμικό σχεδιασμού** Η σχεδίαση της γεωμετρίας απαιτεί τη χρήση ενός σχεδιαστικού λογισμικού. Για το σκοπό αυτό, χρησιμοποιήθηκε το εμπορικό πακέτο Solidworks. Το Solidworks (δημοσιευμένο από την Dassault Systèmes) είναι ένα λογισμικό solid-modeling CAD-CAE, το οποίο εγκαθίσταται σε λειτουργικό σύστημα Microsoft Windows και χρησιμοποιεί μια παραμετρική feature-based προσέγγιση. Το λογισμικό είναι γραμμένο σε Parasolid-kernel.

Η σχεδίαση ενός μοντέλου στο Solidworks συνήθως αρχίζει με ένα διδιάστατο (2-D) σκαρίφημα (sketch), αν και υπάρχει δυνατότητα τριδιάστατου σκαριφήματος (3-D sketch). Το σκαρίφημα αποτελείται από γεωμετρικά στοιχεία όπως: σημεία, ευθύγραμμα τμήματα, καμπύλες, κωνικές τομές (εκτός από υπερβολή) και splines. Έπειτα, στο σκαρίφημα προστίθενται διαστάσεις, ώστε να καθοριστεί το μέγεθος αλλά και η θέση της γεωμετρίας. Επίσης, δίνεται η δυνατότητα να προστεθούν σχέσεις μεταξύ των γεωμετρικών στοιχείων - λ.χ. επαπτομενικότητα, παραλληλότητα, καθετότητα και ομοκεντρότητα. Με τις διάφορες δυνατότητες που παρέχει το λογισμικό, το διδιάστατο (ή και τριδιάστατο) σκαρίφημα, παίρνει μορφή στο τελικό τριδιάστατο μοντέλο. Επίσης, δυνατή είναι η δημιουργία συναρμογών (assemblies). Η παραμετρική φύση του Solidworks σημαίνει πως οι διαστάσεις και οι γεωμετρικές σχέσεις οδηγούν τη γεωμετρία, και όχι το αντίστροφο. Οι διαστάσεις του σκαριφήματος μπορούν να ελεγχθούν ανεξάρτητα ή σε συνάρτηση με άλλες παραμέτρους, εντός ή εκτός του σκαριφήματος. ([1])

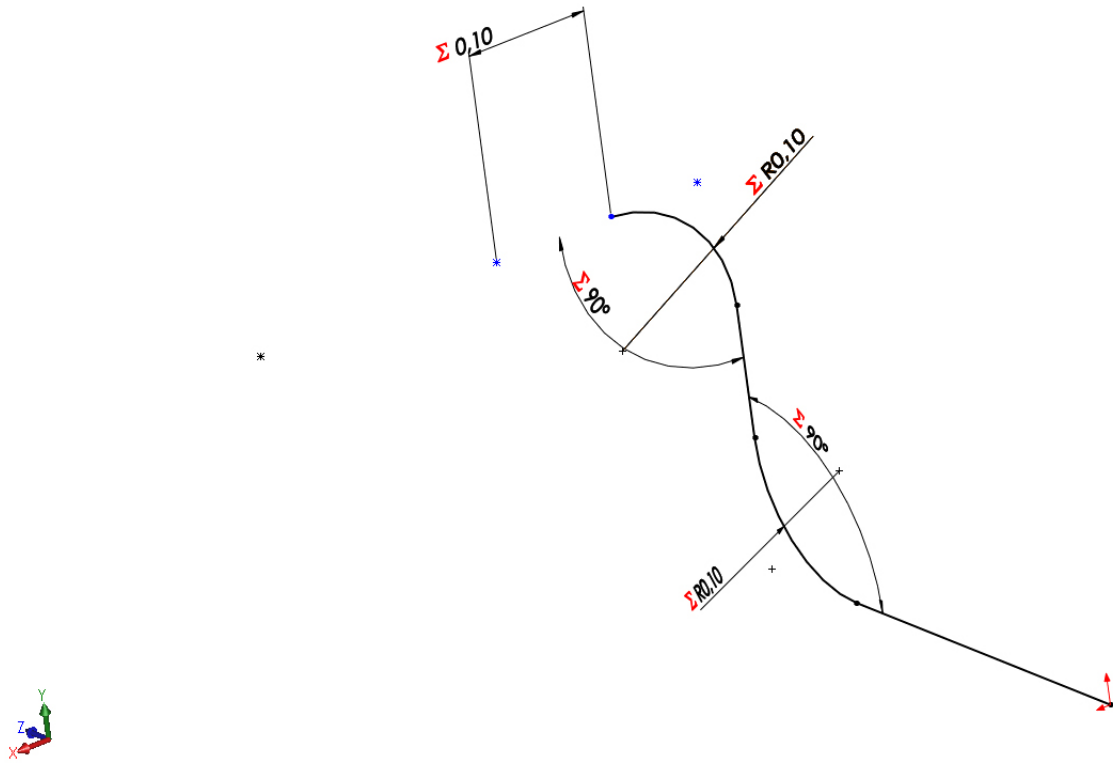
**Σχεδίαση του αγωγού** Το παράδειγμα που διαπραγματεύεται η εργασία αυτή, ανάγεται στη σχεδίαση ενός αγωγού με δύο γωνίες (Double Elbow Duct) ορθογωνικής διατομής, ο οποίος συνδέει δύο διατομές (είσοδος και έξοδος) που βρίσκονται σε κάθετα επίπεδα (σχήμα 2.1). Το παραπάνω πραγματοποιείται με έναν αγωγό που έχει δύο ορθές γωνίες. Ο αγωγός προορίζεται για αέρα χαμηλής ταχύτητας (λ.χ. ως κανάλι εξαερισμού, θέρμανσης χώρων)



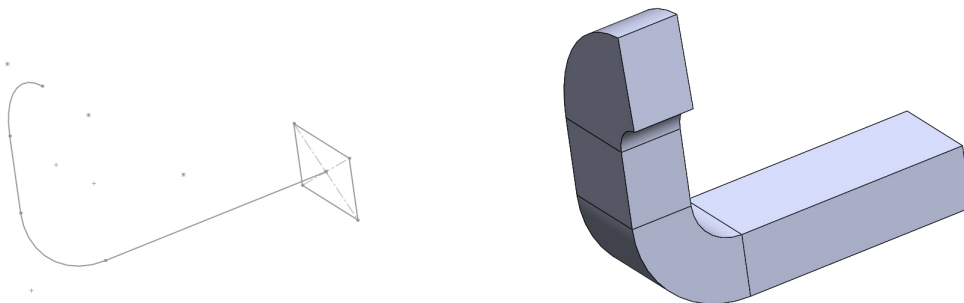
**Σχήμα 2.1:** Οι διατομές εισόδου/εξόδου και οι διαστάσεις του χώρου

Η σχεδίαση ξεκινάει με το 3D σκαρίφημα του σχήματος 2.2, στο οποίο ορίζονται οι εικονιζόμενες διαστάσεις. Αφού σχεδιαστεί και η διατομή εισόδου, με τη λειτουργία "sweep", διαμορφώνεται το 3D μοντέλο της εικόνας 2.3. Σημειώνεται πως το 3D μοντέλο σχεδιάζεται ως συμπαγές, και όχι κοίλο όπως είναι ένας αγωγός. Η μοντελοποίηση αυτή έγινε με σκοπό να διευκολυνθεί η δημιουργία του πλέγματος, στο οποίο επιλέγονται μόνο τα τοιχώματα (βλ. παράγραφο 2.2). Η ορθογωνική διατομή αλλάζει δύο φορές προσανατολισμό, σε σχέση με την είσοδο της διάταξης, μετά από τις δύο γωνίες, με συνέπεια να μην έχει τον ίδιο προσανατολισμό με τη διατομή στην έξοδο. Το παραπάνω πρόβλημα λύνεται με μια αλλαγή της διατομής εντός συγκεκριμένου μήκους (loft), το οποίο φαίνεται στο σχήμα 2.4. Το μήκος του loft ορίστηκε στο αρχικό σκαρίφημα. Τέλος, η προσαρμοσμένη διατομή ενώνεται με την ευθυγραμμισμένη διατομή

εξόδου με τη λειτουργία του "loft".

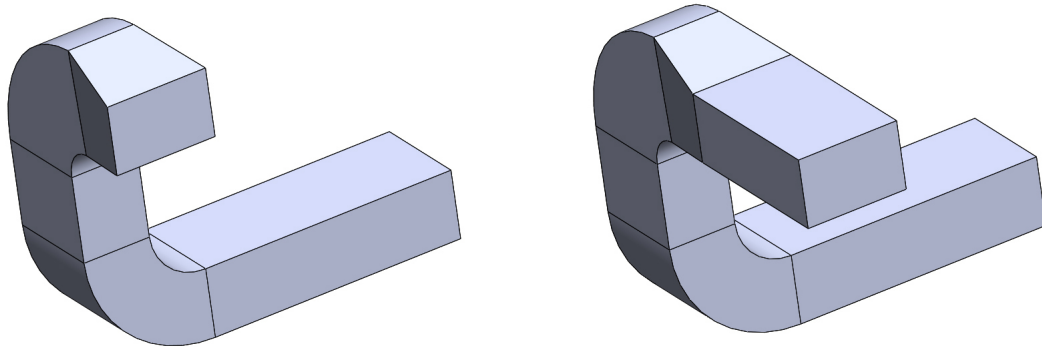


Σχήμα 2.2: Το αρχικό, 3D σκαρίφημα - Μέση γραμμή του αγωγού



Σχήμα 2.3: Με τη λειτουργία "sweep", το σκαρίφημα μαζί με τη διατομή εισόδου παίρνει 3D μορφή





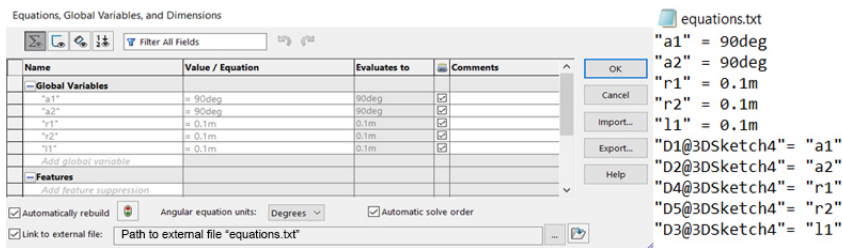
**Σχήμα 2.4:** Προσαρμογή της διατομής με loft (αριστερά), ένωση με τη διατομή εξόδου (δεξιά)

Συνεπώς, η γεωμετρία περιγράφεται με τις εξής παραμέτρους:

- Διατομή εισόδου
- Διατομή εξόδου
- Πρώτη γωνία ( $a_1$ )
- Δεύτερη γωνία ( $a_2$ )
- Στρογγύλεμα (Fillet) της γωνίας  $a_1$  ( $r_1$ )
- Στρογγύλεμα (Fillet) της γωνίας  $a_2$  ( $r_1$ )
- Μήκος του loft ( $l_1$ )

Ορίζοντας τις διαστάσεις των παραπάνω παραμέτρων, η γεωμετρία ορίζεται πλήρως. Θεωρώντας τις διαστάσεις των διατομών εισόδου/εξόδου σταθερές, η γεωμετρία εξαρτάται από τις πέντε τελευταίες παραμέτρους. Οι διαστάσεις τους, μπορούν να ονομαστούν στο εξής ως μεταβλητές σχεδιασμού. Με τη λειτουργία "equations", ορίζονται ως μεταβλητές και συνδέονται αυτόματα με εξωτερικό αρχείο ("equations.txt"), στο οποίο περιέχονται οι τιμές των μεταβλητών (σχήμα 2.5). Η αυτοματοποιημένη παραγωγή της γεωμετρίας με script αναλύεται στην παράγραφο 2.3. Να σημειωθεί πως οι τιμές των πέντε μεταβλητών πρέπει να είναι 'λογικές', ειδάλλως η γεωμετρία δεν ορίζεται. Η γεωμετρία αποθηκεύεται σε αρχείο μορφής \*.SLDPRT, το οποίο ανοίγεται από το Solidworks, και σε αρχείο Parasolid \*.x.t ([2], [3]), το οποίο χρησιμοποιείται αργότερα για την κατασκευή του πλέγματος.

Η μέθοδος που χρησιμοποιήθηκε για την παραμετροποίηση αυτού του αγωγού, μπορεί να εφαρμοστεί σε οποιαδήποτε γεωμετρία που σχεδιάζεται στο λογισμικό Solidworks. Ο αγωγός σχεδιάστηκε ως ένα παράδειγμα για να υποστηρίξει την επίδειξη των τεχνικών που αναπτύχθηκαν και δεν αποτελεί αυτοσκοπό της εργασίας αυτής.



Σχήμα 2.5: Η λειτουργία equations του Solidworks (αριστερά) και το εξωτερικό αρχείο "equations.txt" (δεξιά)

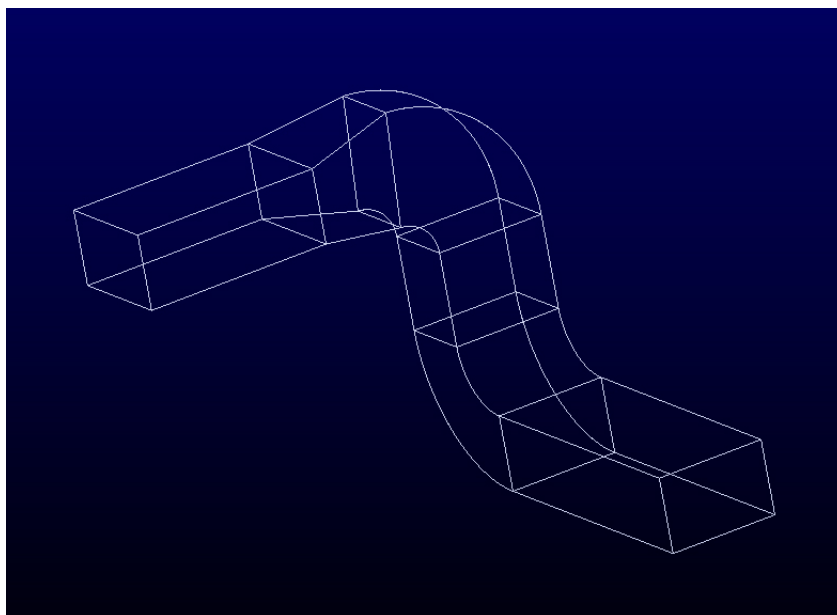
## 2.2 Γένεση Πλέγματος

**Λογισμικό γένεσης πλέγματος** Η γένεση του πλέγματος πραγματοποιήθηκε χρησιμοποιώντας το λογισμικό Pointwise ([4]), το οποίο είναι ένα εμπορικό λογισμικό παραγωγής πλεγμάτων της Pointwise, Inc. Το Pointwise μπορεί να διαχειριστεί και να παράξει διαφόρων ειδών πλέγματα (π.χ. structured quad and hex, unstructured tri and tet, hybrid prism, tet, hex, pyramid, κ.α.), ενώ είναι διαθέσιμο στα λειτουργικά Windows, Linux και Mac OS X. Διαθέτει πολλά εργαλεία με σκοπό την επιθυμητή τοποθέτηση των κόμβων (π.χ. με χρήση ελλειπτικών και υπερβολικών μ.δ.ε.), ενώ υποστηρίζει πολλές μορφές αρχείων προοριζόμενα για διάφορους επιλύτες (π.χ. ANSYS CFX, OpenFOAM, SU2, Tecplot, κ.α.). Ένα από τα χαρακτηριστικά του Pointwise που οδήγησαν στη χρήση του, είναι το γεγονός πως υποστηρίζει μεγάλη ποικιλία αρχείων εισόδου, ιδιαίτερα προερχόμενα από λογισμικά CAD (IGES, CATIA, Pro/E & Creo, Parasolid, SolidWorks, κ.α.). Επίσης σημαντικό χαρακτηριστικό που προσφέρει το λογισμικό Pointwise είναι η δυνατότητα για scripting. Για τον σκοπό της εργασίας, το λογισμικό Pointwise παρέχεται από τη ΜΠΥΡΒ/ΕΜΠ.

**Γένεση του πλέγματος** Η περίπτωση του κλειστού αγωγού είναι ένα κλασικό παράδειγμα εσωτερικής ρευστοδυναμικής, αφού το υπολογιστικό πλέγμα περιβάλλεται από τοιχώματα. Η γένεση του πλέγματος είναι μια διαδικασία η οποία πρέπει να γίνει με προσοχή. Ένα αραιό σε κόμβους πλέγμα δεν θα δώσει ικανοποιητικής ακρίβειας αποτελέσματα, ενώ ένα πυκνό σε κόμβους πλέγμα θα καταστήσει την επίλυση της ροής κοστοβόρα. Στην παράγραφο αυτή, περιγράφεται η μέθοδος με την οποία προέκυψε το αρχικό πλέγμα, το οποίο αποτελεί σημείο αναφοράς για την πορεία μέχρι την επιλογή του τελικού πλέγματος. Η μελέτη για την επιλογή του τελικού πλέγματος γίνεται στην παράγραφο 3.2.

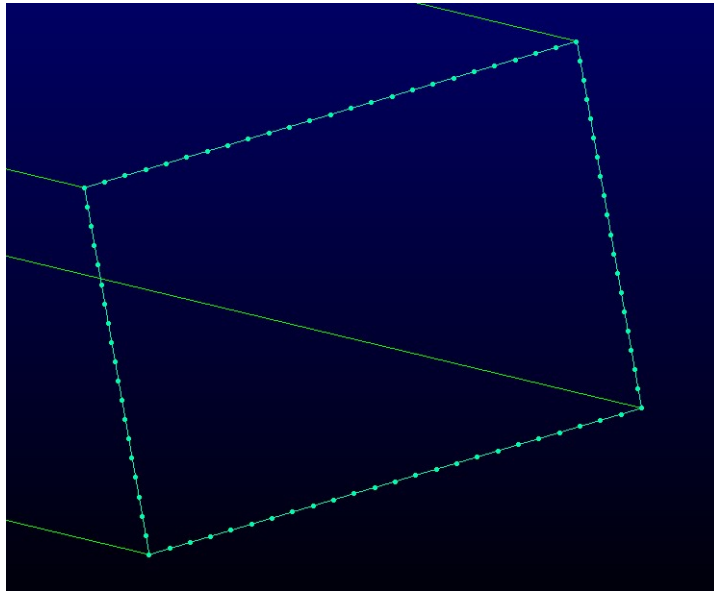
Τα πλέγματα μπορούν να κατηγοριοποιηθούν σύμφωνα με τη δομή τους σε: δομημένα, μη δομημένα και υβριδικά. Το κάθε είδος πλέγματος μπορεί να έχει πλεονεκτήματα και μειονεκτήματα, αναλόγως την εφαρμογή. Στη συγκεκριμένη περίπτωση, το πλέγμα, εφόσον η γεωμετρία το επιτρέπει, επιλέγεται να είναι δομημένο, λόγω της υψηλής τους ακρίβειας και της ευκολίας κατά την επίλυση. Στην παρακάτω παράγραφο αναλύεται ο τρόπος γένεσης του πλέγματος.

Το λογισμικό Pointwise δέχεται την εισαγωγή αρχείων \*.x\_t και έτσι εισάγεται το αντίστοιχο αρχείο που παράγεται από το Solidworks και περιέχει τη γεωμετρία (σχήμα 2.6).

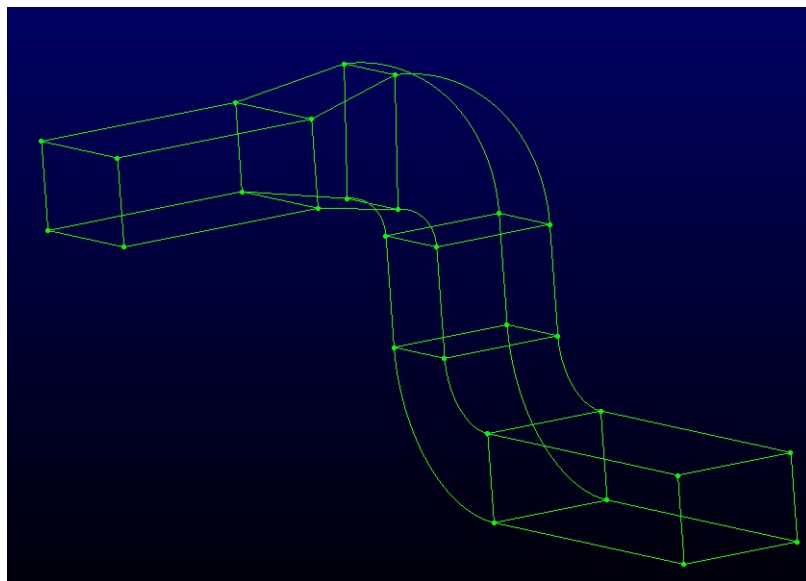


**Σχήμα 2.6:** Το μοντέλο που λαμβάνεται εισάγοντας το αρχείο \*.x\_t στο Pointwise

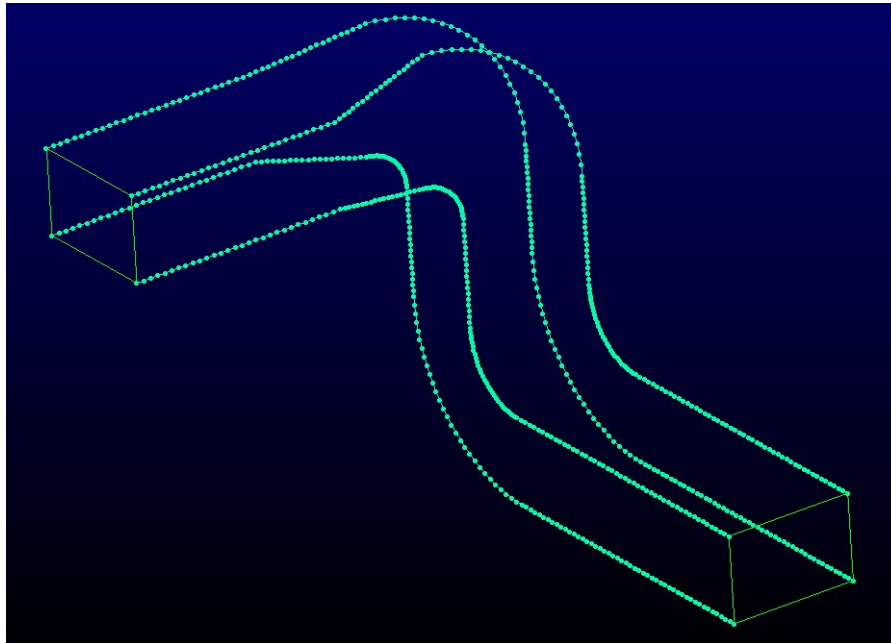
Στη συνέχεια, ορίζεται αριθμός κόμβων σε κάθε πλευρά της ορθογωνικής διατομής του αγωγού (σχήμα 2.7). Κατα μήκος ο αγωγός χωρίζεται σε έξι τμήματα (σχήμα 2.8) από το λογισμικό Solidworks, για το καθένα από τα οποία ορίζεται αριθμός κόμβων (σχήμα 2.9). Στις ενώσεις των τμημάτων, ορίζεται το διάκενο που έχουν οι κόμβοι, ώστε να υπάρχει συνέχεια. Ύστερα, το πλέγμα πυκνώνεται σε περιοχές οι οποίες είναι πιο απαιτητικές στην επίλυση της ροής (είσοδος και έξοδος αγωγού, γωνίες και περιοχές που αλλάζει η διατομή), αλλά κυρίως κοντά στα τοιχώματα (σχήμα 2.10). Για την πυκνωση των κόμβων κοντά στα τοιχώματα, με τη λειτουργία του Pointwise 'spacing constraints', εισάγεται το επιθυμητό διάκενο του πρώτου κόμβου από το τοίχωμα, ενώ οι υπόλοιποι κόμβοι εξαπλώνονται με τη συνάρτηση της υπερβολικής εφαιπτομένης (tanh). Η συνάρτηση εξάπλωση των κόμβων μπορεί να επιλεγθεί στην καρτέλα Distribute (σχήμα 2.11).



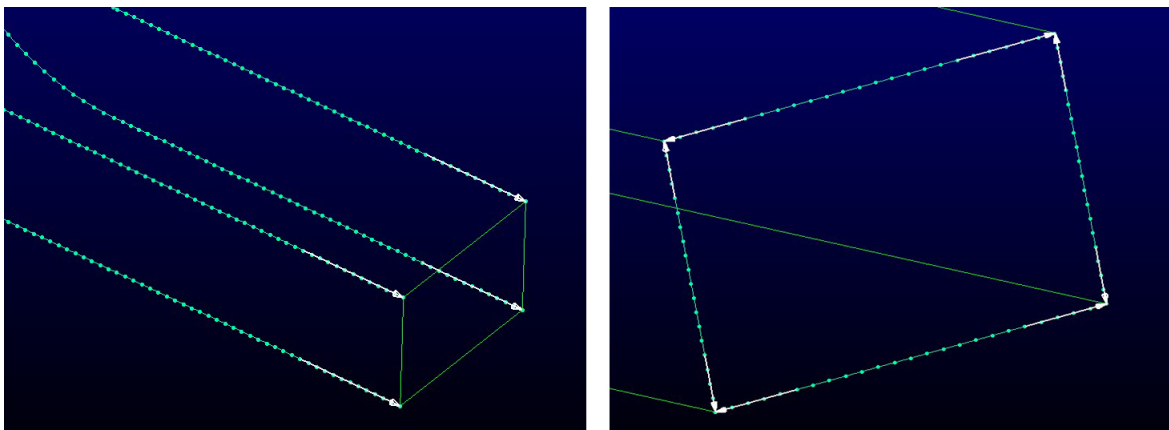
Σχήμα 2.7: Η ορθογωνική διατομή του αγωγού, στην οποία ορίστηκε ένας αριθμός κόμβων σε κάθε πλευρά



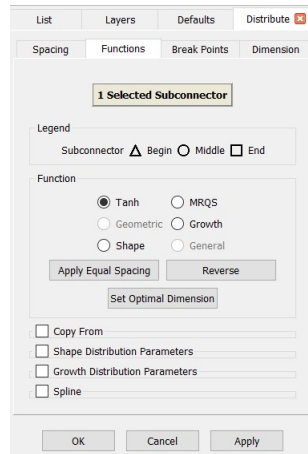
Σχήμα 2.8: Ο αγωγός χωρίζεται σε έξι τμήματα από το λογισμικό *Solidworks*



**Σχήμα 2.9:** Ορίζοντας αριθμό κόμβων σε κάθε τμήμα του αγωγού, συγκροτείται ένα σύνολο κόμβων κατά μήκος του

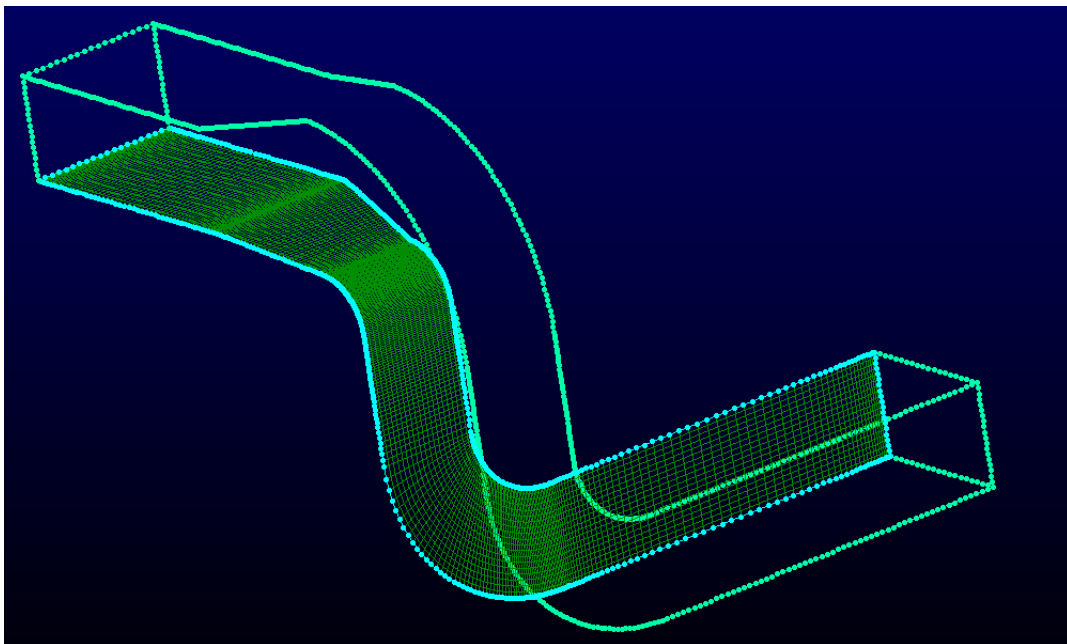


**Σχήμα 2.10:** Τα άσπρα βέλη δείχνουν την κατεύθυνση στην οποία θα μικρύνουν τα διάκενα μεταξύ των κόμβων. Το πλέγμα θα πυκνώσει στις περιοχές αυτές

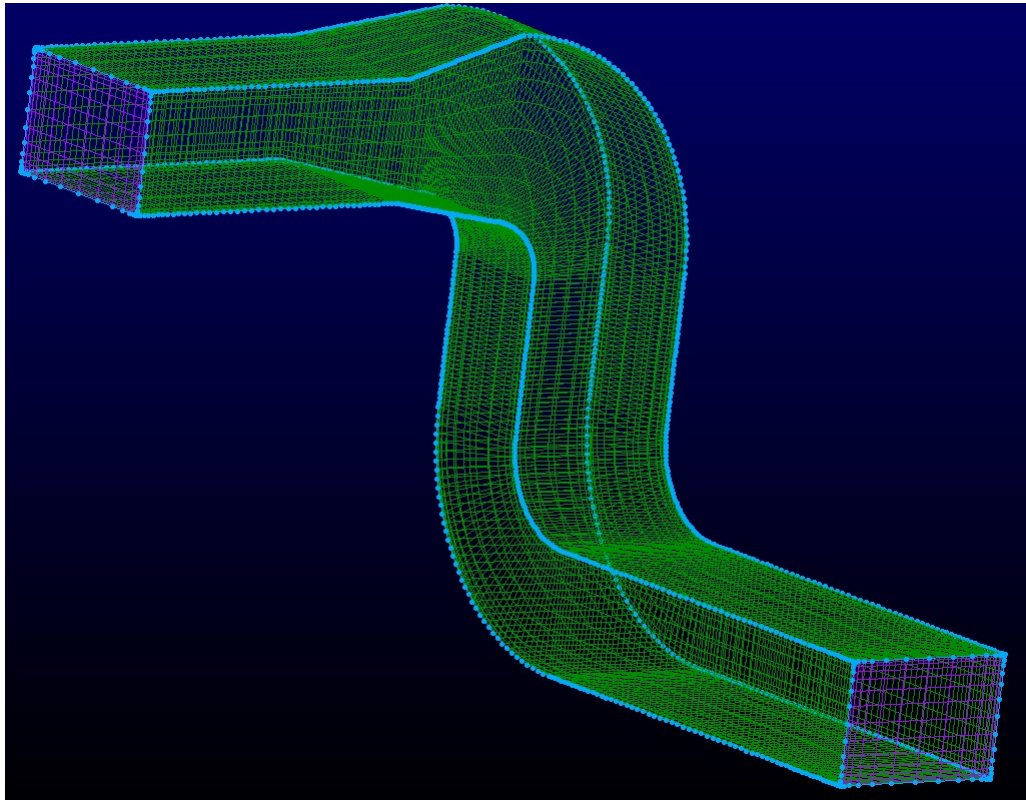


Σχήμα 2.11: Επιλογή συνάρτησης εξάπλωσης των κόμβων, στο αντίστοιχο τμήμα

Μετά την παραπάνω διαδικασία, σε αυτή τη γεωμετρία, οι ακμές συνθέτουν έξι επιφάνειες (την είσοδο, την έξοδο και τέσσερα στερεά τοιχώματα). Με τη λειτουργία Assemble Domains και επιλέγοντας τις κατάλληλες ακμές, δημιουργούνται επιφάνειες, όπως του σχήματος 2.12. Ακολουθώντας την ίδια διαδικασία, τελικά συντίθεται ένα 3D πλέγμα (block). Το πλέγμα έχει αριθμό κόμβων  $A \cdot B \cdot L$ , όπου  $A$  ο αριθμός των κόμβων στην οριζόντια πλευρά της διατομής,  $B$  ο αριθμός των κόμβων στην κατακόρυφη πλευρά της διατομής και  $L$  ο συνολικός αριθμός των κόμβων κατά μήκος του αγωγού (σχήμα 2.13).



Σχήμα 2.12: Ένα από τα στερεά τοιχώματα του πλέγματος

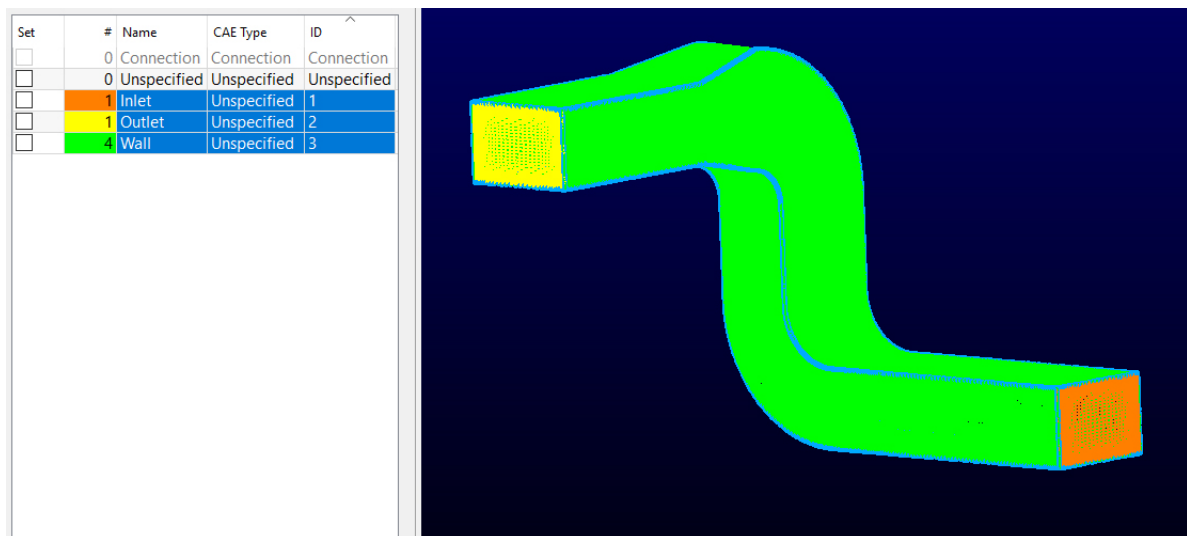


Σχήμα 2.13: Η τελική εικόνα του δομημένου, τριδιάστατου πλέγματος

Εν συνεχεία, πρέπει να ονομαστούν οι επιφάνειες στις οποίες θα επιβληθούν οι οριακές συνθήκες. Έτσι ορίζονται η είσοδος, η έξοδος και τα τοιχώματα, όπως φαίνεται το σχήμα 2.14. Το πλέγμα εξάγεται από το Pointwise σε μορφή \*.su2, καθώς μέσω αυτής της μορφής αρχείου πλεγμάτων μπορεί να γίνει μετατροπή σε μορφή πλεγμάτων LTT, με την οποία είναι συμβατή το χρησιμοποιούμενο λογισμικό επίλυσης της ροής PUMA. Η μετατροπή του αρχείου \*.su2 σε μορφή LTT πραγματοποιείται με εκτελέσιμο αρχείο που διαθέτει η ΜΠΥΡΒ/ΕΜΠ.

**Μορφή αρχείων πλεγμάτων LTT** Η μορφή αρχείων πλεγμάτων LTT αποτελείται από τρία αρχεία. Τα αρχεία αυτά είναι:

- \*.hyb: Περιλαμβάνει τη συνδεσιμότητα (connectivity) των κόμβων
- \*.nod: Περιλαμβάνει τις καρτεσιανές συντεταγμένες των κόμβων
- \*.patch: Περιλαμβάνει πληροφορία για τα patches του πλέγματος



**Σχήμα 2.14:** Οι επιφάνειες στις οποίες θα επιβληθούν οριακές συνθήκες, με τα αντίστοιχα ονόματα

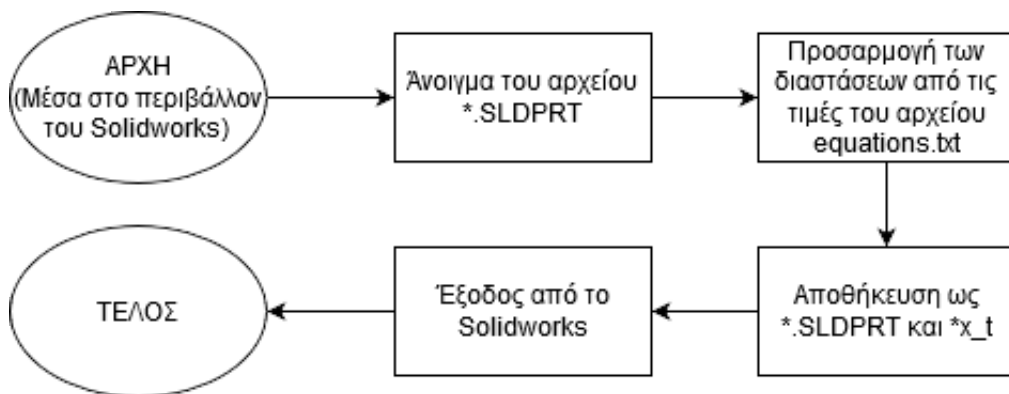
## 2.3 Αυτοματοποιημένη Παραγωγή Γεωμετρίας και Πλέγματος

Η παραπάνω διαδικασία του σχεδιασμού της γεωμετρίας και της γένεσης του πλέγματος είναι ένα στάδιο που υποχρεωτικά πρέπει να γίνει χειροκίνητα, καθώς όπως φάνηκε στις προηγούμενες παραγράφους, υπάρχουν πολλές σχεδιαστικές παράμετροι που ένας μηχανικός καλείται να ρυθμίσει. Ωστόσο, αφού ο μηχανικός επιλέξει τις παραμέτρους αυτές και καταλήξει σε μια γεωμετρία και ένα πλέγμα, η παραπάνω διαδικασία μπορεί να πραγματοποιηθεί αυτόματα, για άλλες τιμές των παραμέτρων, με τη χρήση των macro - scripts.

**Παραγωγή γεωμετρίας** Όπως ήδη αναφέρθηκε, η γεωμετρία του αγωγού εξαρτάται από πέντε μεταβλητές, οι τιμές των οποίων βρίσκονται στο αρχείο "equations.txt", το οποίο διαβάζει το λογισμικό Solidworks και παράγει το 3D μοντέλο. Με τη χρήση macro (στην αντίστοιχη λειτουργία του λογισμικού), προγραμματίζεται ο παρακάτω αλγόριθμος (σχήμα 2.15). Το macro αποθηκεύεται στη μορφή \*.swp και ανοίγεται με το Solidworks. Ουσιαστικά, με τη λειτουργία macro, ο χρήστης καταγράφει τις εντολές που δίνει στο λογισμικό κατά τη διάρκεια της σχεδίασης του μοντέλου. Η λειτουργία macro αποθηκεύει τις εντολές αυτές σε ένα script, το οποίο ο χρήστης μπορεί να εκτελέσει και να επεξεργαστεί στο Solidworks. Αρχικά, ο χρήστης πρέπει να έχει σχεδιάσει τη γεωμετρία και να έχει συνδέσει τις διαστάσεις με το εξωτερικό αρχείο "equations.txt". Πριν την καταγραφή του macro, ο χρήστης πρέπει να αλλάξει τις τιμές των μεταβλητών στο αρχείο "equations.txt" στις επιθυμητές. Ύστερα στο Soli-

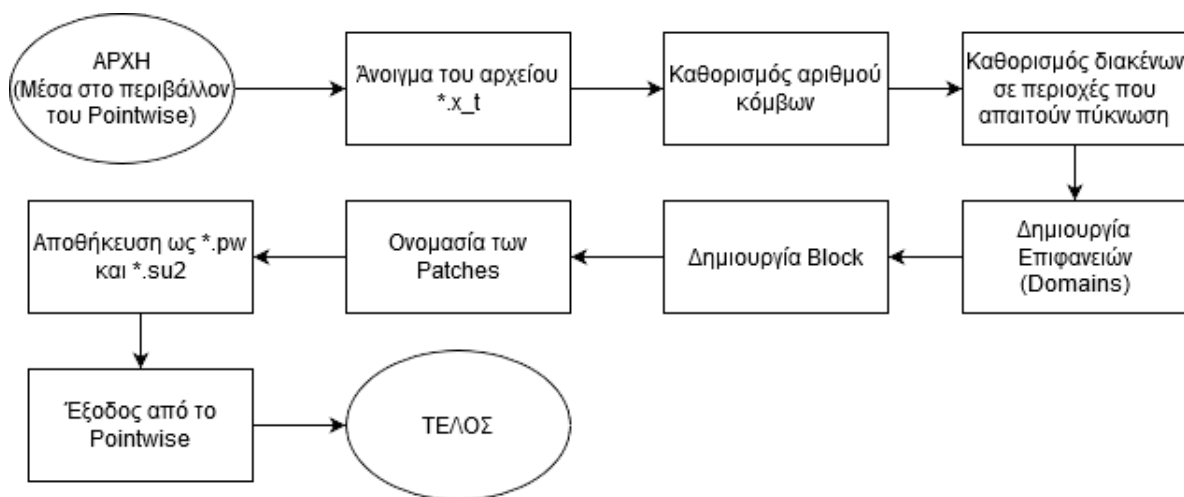


dworks, ξεκινώντας την καταγραφή του macro, ο χρήστης πρέπει να ανοίξει το αρχείο \*.SLDPRT, να πατήσει την εντολή "Rebuild" (η οποία προσαρμόζει τις διαστάσεις με βάση το συνδεδεμένο εξωτερικό αρχείο) και να αποθηκεύσει (και κάνει over-write) το νέο αρχείο \*.x\_t.



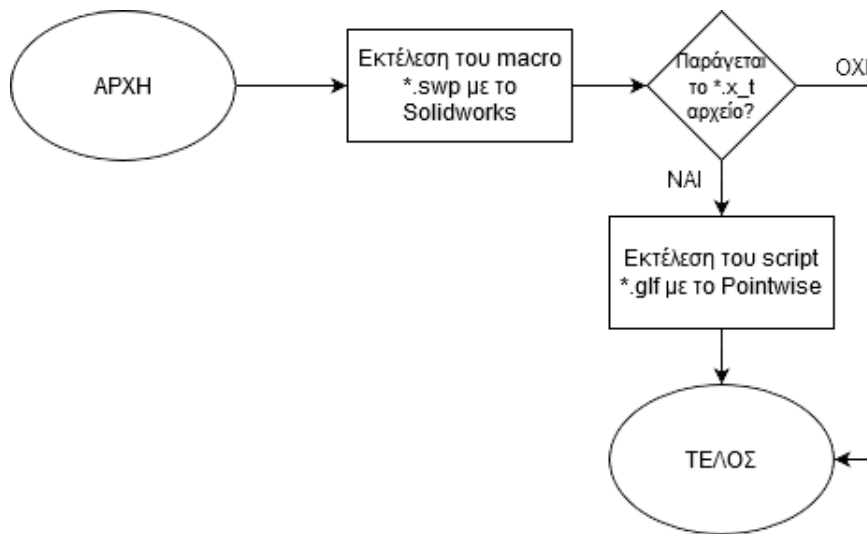
Σχήμα 2.15: Το διάγραμμα ροής του macro \*.swp

**Γένεση πλέγματος** Όπως και το Solidworks, το λογισμικό Pointwise διαθέτει λειτουργία scripting, με την οποία είναι δυνατό να παραχθεί το πλέγμα αυτόματα. Έτσι, συντίθεται ένα script που ακολουθεί το παρακάτω διάγραμμα ροής (σχήμα 2.16). Το script αποθηκεύεται στη μορφή \*.glf και ανοίγεται με το λογισμικό Pointwise. Αντίστοιχα με το Solidworks, το λογισμικό Pointwise με τη λειτουργία scripting, καταγράφει τις εντολές που εκτελεί ο χρήστης κατά τη γένεση του πλέγματος. Οι εντολές αποθηκεύονται στο script που αναφέρθηκε, το οποίο εκτελείται με το Pointwise και μπορεί να επεξεργαστεί με text editor.



Σχήμα 2.16: Το διάγραμμα ροής του script \*.glf

Έχοντας ετοιμάσει το macro του Solidworks και το script του Pointwise, είναι δυνατό να κληθούν με τη σειρά με ένα script της μορφής \*.bat:



**Σχήμα 2.17:** Το διάγραμμα ροής του εκτελέσιμου \*.bat

Να σημειωθεί πως είναι απαραίτητο να ελεγχθεί με μια εντολή IF εντός του εκτελέσιμου \*.bat αν το macro \*.swp παράγει το αρχείο γεωμετρίας \*.x\_t, αλλιώς το script \*.glf δεν θα εκτελεστεί σωστά, και το Pointwise δεν θα τερματιστεί.

Με τα παραπάνω, ο χρήστης μπορεί να ετοιμάσει το αρχείο equations.txt, να τρέξει το εκτελέσιμο \*.bat και να παράξει το αρχείο πλέγματος \*.su2. Η μετατροπή σε μορφή LTT γίνεται σε επόμενο βήμα, πριν επιλυθεί η ροή με το λογισμικό CFD. Το script \*.bat που χρησιμοποιήθηκε για την αυτόματη παραγωγή γεωμετρίας και πλέγματος περιλαμβάνεται στο παράρτημα Α' ως SwPw.bat.

## Κεφάλαιο 3

# Υπολογιστική Προσομοίωση της Ροής (CFD)

Στο κεφάλαιο αυτό παρουσιάζεται η μελέτη που πραγματοποιήθηκε για τη μοντελοποίηση της ροής. Η υπολογιστική ρευστοδυναμική αποτελεί απαραίτητο εργαλείο ενός μηχανικού που εκτελεί έργο στο χώρο του σχεδιασμού σε προβλήματα μηχανικής των ρευστών. Η χρήση ενός αξιόπιστου λογισμικού επίλυσης προβλημάτων ρευστομηχανικής θα οδηγήσει σε αποδεκτά αποτελέσματα. Ωστόσο, η επιλογή των παραμέτρων του επιλύτη χρήζει ανάλυσης, ειδάλλως η τελική εικόνα της ροής θα είναι πιθανόν αμφίβολης ακρίβειας, ενώ θα έχουν καταναλωθεί πόροι και χρόνος.

**Λογισμικό Επίλυσης της Ροής** Για την επίλυση της ροής χρησιμοποιήθηκε το λογισμικό PUMA, το οποίο αναπτύχθηκε στη ΜΠΥΡΒ/ΕΜΠ. Το PUMA είναι ένα υψηλής ακρίβειας και ταχύτητας CFD λογισμικό, που χρησιμοποιεί τη μέθοδο Πεπερασμένων Όγκων. Σε όλες τις παρακάτω μελέτες, χρησιμοποιήθηκε έκδοση του PUMA που εκτελείται σε επεξεργαστές κάρτες γραφικών NVIDIA-GPUs. Το λογισμικό PUMA εκτελείται τις υπολογιστικές συστοιχίες της ΜΠΥΡΒ/ΕΜΠ ([5], [6], [7]).

### 3.1 Επίλυση της ροής

Η επίλυση της ροής προϋποθέτει, αρχικά, την επιλογή των βασικών χαρακτηριστικών της, καθώς επίσης και του ρευστού. Το ρευστό που διαρρέει τον αγωγό επιλέγεται αέρας χαμηλής ταχύτητας, κινηματικής συνεκτικότητας  $\nu = 15.5 \cdot 10^{-6} m^2/s$ . Ως γνωστόν, ένας δείκτης καίριας σημασίας για τη μηχανική των ρευστών είναι ο αριθμός Reynolds

που, στην περίπτωση κλειστών αγωγών, υπολογίζεται ως εξής([8]):

$$Re = \frac{uD_h}{\nu}$$

όπου  $u$  η ταχύτητα του ρευστού και  $D_h$  η υδραυλική διάμετρος. Η υδραυλική διάμετρος ορίζεται ως:

$$D_h = \frac{4A}{P}$$

όπου  $A$  το εμβαδόν της διατομής και  $P$  η βρεχόμενη περίμετρος.

Σύμφωνα με τη βιβλιογραφία [9], σε διατάξεις αγωγών εξαερισμού, η ταχύτητα του αέρα δεν πρέπει να ξεπερνά ορισμένες τιμές, ώστε να αποφευχθούν οι υψηλές απώλειες πίεσης, αλλά και ο θόρυβος να διατηρείται χαμηλός. Για ζεστό αέρα προοριζόμενο για οικιακή θέρμανση, ένα ενδεικτικό εύρος τιμών ταχύτητας είναι  $0.8 - 1\text{m/s}$ . Συνεπώς, υπολογίζεται ο αριθμός Reynolds:

$$\text{Διαστάσεις διατομής αγωγού: } 0.08\text{m} \times 0.12\text{m} \Rightarrow A = 0.0096\text{m}^2, P = 0.4\text{m}$$

$$\Rightarrow D_h = 0.096\text{m}$$

$$Re = 4955 - 6193$$

**Εξισώσεις** Η συμπιεστότητα του ρευστού θεωρείται αμελητέα, με το σκεπτικό πως ο αριθμός Mach εντός του αγωγού είναι πολύ κοντά στο μηδέν, συνεπώς το ρευστό θεωρείται ασυμπίεστο. Οι εξισώσεις που επιλύονται για τη ροή είναι οι Reynolds-Averaged Navier-Stokes (RANS). Το μοντέλο τύρβης που χρησιμοποιείται είναι το μοντέλο των Spalart-Allmaras ([10]), το οποίο προσθέτει μία μεταβλητή και μία μερική διαφορική εξίσωση στο σύστημα. Οι εξισώσεις που περιγράφουν τη ροή είναι οι παρακάτω:

- Κατά Reynolds εξίσωση διατήρησης της μάζας

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0 \quad (3.1)$$

- Κατά Reynolds εξίσωση διατήρησης της ορμής

$$\bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j} \left[ (\nu + \nu_t) \left( \frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) \right] \quad (3.2)$$

- Εξίσωση του μοντέλου Spalart - Allmaras [11]

$$u_j \frac{\partial \tilde{\nu}}{\partial x_j} = c_{b1}(1 - f_{t2})\tilde{S}\tilde{\nu} - \left[ c_{w1}f_w - \frac{c_b1}{\kappa^2}f_{t2} \right] \left( \frac{\tilde{\nu}}{d} \right)^2 + \frac{1}{\sigma} \left[ \frac{\partial}{\partial x_j} \left( (\nu + \tilde{\nu}) \frac{\partial \tilde{\nu}}{\partial x_j} \right) + c_{b2} \frac{\partial \tilde{\nu}}{\partial x_i} \frac{\partial \tilde{\nu}}{\partial x_i} \right] \quad (3.3)$$

όπου η τυρβώδης συνεκτικότητα (turbulent or eddy viscosity) δίνεται από τη σχέση

$$\mu_t = \rho \tilde{\nu} f_{u1}$$

οπου:

$$f_{u1} = \frac{\chi^3}{\chi^3 + c_{u1}^3}$$

$$\chi = \frac{\tilde{\nu}}{\nu}$$

$$\tilde{S} = \Omega + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{u2}$$

$$\Omega = \sqrt{2W_{ij}W_{ij}}$$

$$f_{u2} = 1 - \frac{\chi}{1 + \chi f_{u1}}, \quad f_w = g \left[ \frac{1 + c_{w3}^6}{g^6 + c_{w3}^6} \right]^{1/6}$$

$$g = r + c_{w2}(r^6 - r)$$

$$r = \min \left[ \frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2}, 10 \right]$$

$$f_{t2} = c_{t3} \exp(-c_{t4}\chi^2)$$

$$c_{b1} = 0.1355 \quad \sigma = 2/3 \quad c_{b2} = 0.622 \quad \kappa = 0.41$$

$$c_{w2} = 0.3 \quad c_{w3} = 2 \quad c_{u1} = 7.1 \quad c_{t3} = 1.2 \quad c_{t4} = 0.5$$

$$c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{1 + c_{b2}}{\sigma}$$

**Ρυθμίσεις** Το πλέγμα για τη δεύτερη προσέγγιση έχει αριθμό κόμβων  $50 \times 50 \times 600 = 1500000$ . Το (ασυμπίεστο) ρευστό έχει ιδιότητες: πυκνότητα  $\rho = 1.2 \text{ kg/m}^3$  και κινηματική συνεκτικότητα  $\nu = 1.55 \cdot 10^{-5} \text{ m}^2/\text{s}$ . Οι ρυθμίσεις του επιλύτη επιλέχθηκαν ως εξής:

- Μέθοδος Ολοκλήρωσης στον Ψευδο-χρόνο : 3 - Stage Runge Kutta με κεντροκομβική διατύπωση

- Σχήμα Διακριτοποίησης : Roe 2ης τάξης
- Μοντέλο Τύρβης : Spalart-Allmaras με χρήση συναρτήσεων τοίχου (Wall Functions)

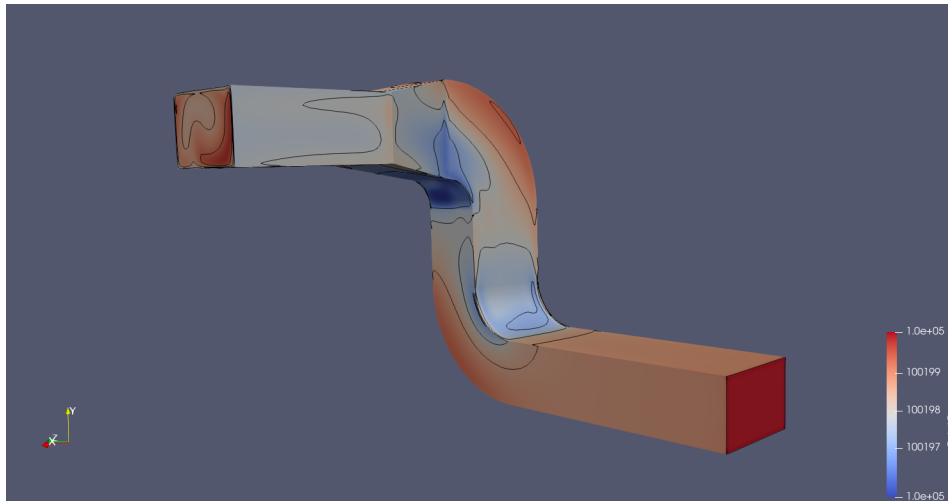
Οι οριακές συνθήκες είναι οι εξής:

- Είσοδος αγωγού (inlet) :  $P_t = 100200 Pa$ , ροή παράλληλη με το κάθετο διάνυσμα της διατομής εισόδου, λόγος συνεκτικότητας (viscosity ratio) = 10
- Έξοδος αγωγού (outlet) :  $P_{st} = 100197.6 Pa$
- Τοιχώματα (walls) : Αδιαβατικά

Ως αρχικοποίηση του πεδίου, ορίστηκε πίεση  $P_t = 100200 Pa$ , ομοιόμορφη ταχύτητα  $u = 1 m/s$  στον άξονα X.

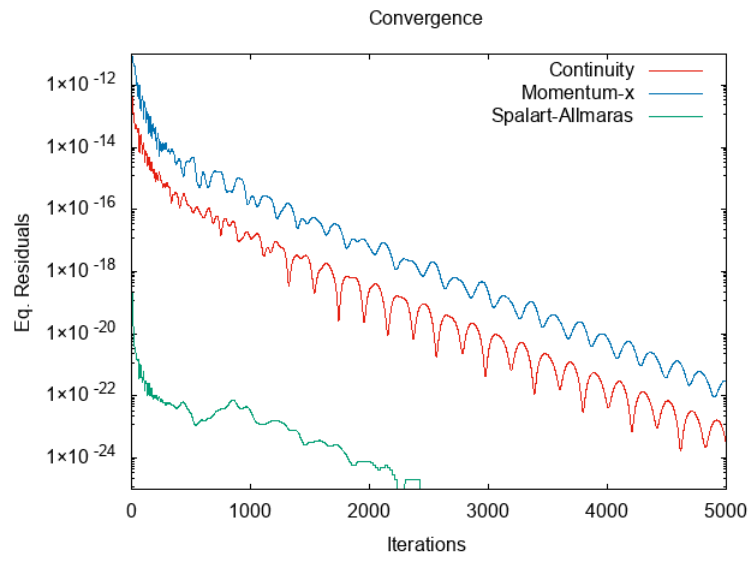
**Αποτελέσματα** Η μέση ολική πίεση στην έξοδο υπολογίζεται:  $P_{t,out} = 100198.675 Pa$ , ενώ η παροχή μάζας που διέρχεται από τον αγωγό:  $\dot{m} = 0.01542 kg/s$ . Γνωρίζοντας την παροχή μάζας, υπολογίζεται η ταχύτητα και ο αριθμός Re:

$$u = \frac{\dot{m}}{\rho \cdot A} \rightarrow u = 1.34 m/s \rightarrow Re = 8290.5$$

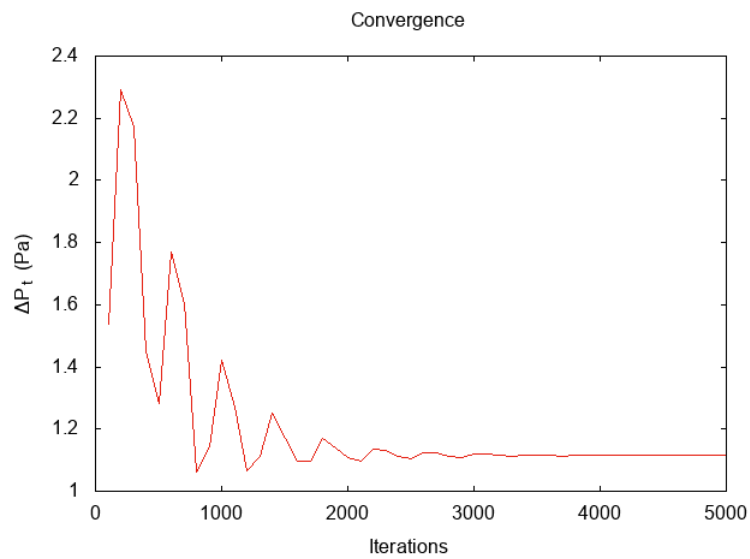


**Σχήμα 3.1:** Διάγραμμα ισογραμμών ολικής πίεσης κατά μήκος του αγωγού

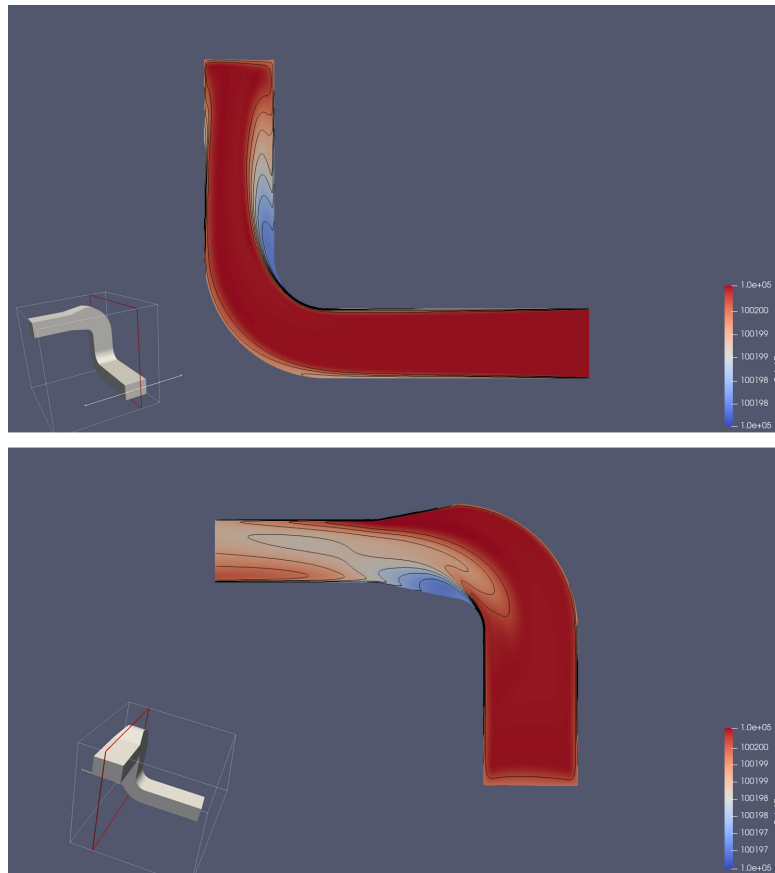
Παραπάνω, φαίνεται το σχήμα 3.1, στο οποίο παρουσιάζεται το διάγραμμα contour της ολικής πίεσης κατά μήκος του αγωγού, ενώ στο σχήμα 3.4 φαίνονται οι ισογραμμές ολικής πίεσης στα μεσημβρινά επίπεδα του αγωγού. Ενδεικτικά, παρουσιάζεται και η σύγκλιση της υπολογιστικής προσομοίωσης (σχήμα 3.2), για την εξίσωση της συνέχειας (continuity), για την εξίσωση ορμής κατά τον άξονα x (Momentum-x) και για την εξίσωση της μοντελοποίησης της τύρβης (Spalart-Allmaras).



**Σχήμα 3.2:** Σύγκλιση - Οριζόντιος άξονας: Αριθμός επαναλήψεων. Κατακόρυφος άξονας: Υπόλοιπα (Residuals) των εξισώσεων



**Σχήμα 3.3:** Σύγκλιση απωλειών ολικής πίεσης



Σχήμα 3.4: Ισογραμμές ολικής πίεσης στα μεσημβρινά επίπεδα

## 3.2 Ανεξαρτησία Πλέγματος

Η μελέτη για την ανεξαρτησία του πλέγματος είναι στάδιο απαραίτητο για την εξαγωγή αξιόπιστων αποτελεσμάτων, τα οποία εξαρτώνται μόνο από τις οριακές συνθήκες. Συνεπώς, ο σκοπός είναι η εξαγωγή λύσης, η οποία είναι ανεξάρτητη από την πυκνότητα του πλέγματος. Πραγματοποιήθηκαν τρεις διαφορετικές μελέτες, η κάθε μία από τις οποίες αναλύεται παρακάτω.

### 3.2.1 Μελέτη 1

Η πρώτη μελέτη αποτελείται από επτά περιπτώσεις, δηλαδή επτά υπολογιστικές προσομοιώσεις με πλέγματα διαφορετικής πυκνότητας. Οι περιπτώσεις είναι οι εξής:

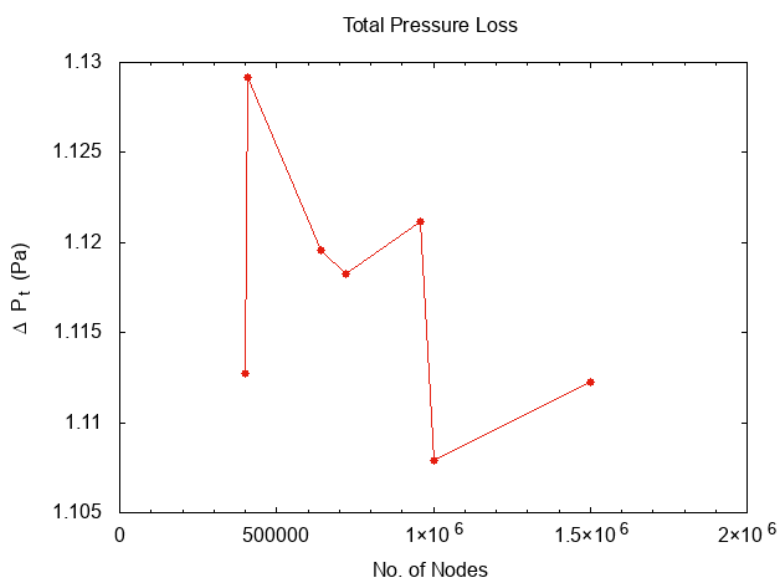
- Περίπτωση 1: No. of Nodes =  $50 \times 50 \times 600 = 1500000$
- Περίπτωση 2: No. of Nodes =  $50 \times 50 \times 400 = 1000000$
- Περίπτωση 3: No. of Nodes =  $40 \times 40 \times 600 = 960000$



- Περίπτωση 4: No. of Nodes =  $40 \times 40 \times 450 = 720000$
- Περίπτωση 5: No. of Nodes =  $40 \times 40 \times 400 = 640000$
- Περίπτωση 6: No. of Nodes =  $32 \times 32 \times 400 = 409600$
- Περίπτωση 7: No. of Nodes =  $40 \times 40 \times 250 = 400000$

Το συγκρινόμενο μέγεθος για τη μελέτη είναι η διαφορά ολικής πίεσης  $\Delta P_t = P_{t,in} - P_{t,out}$ , όπου  $P_{t,in}$  και  $P_{t,out}$  είναι η μέση ολική πίεση στην είσοδο και αντίστοιχα στην έξοδο του αγωγού.

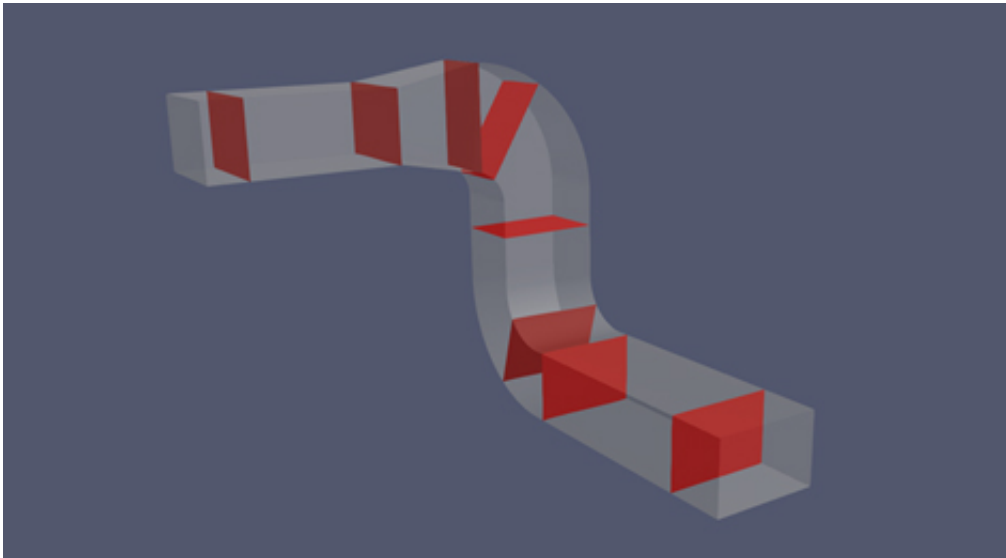
Στο σχήμα 3.5, φαίνονται τα αποτελέσματα. Εκ πρώτης όψεως, οι απώλειες ολικής πίεσης δείχνουν να είναι πρακτικά ανεξάρτητες από τον αριθμό των κόμβων (σε αυτό το πλήθος κόμβων), καθώς τα δεδομένα δεν φαίνεται να έχουν μονότονη συμπεριφορά.



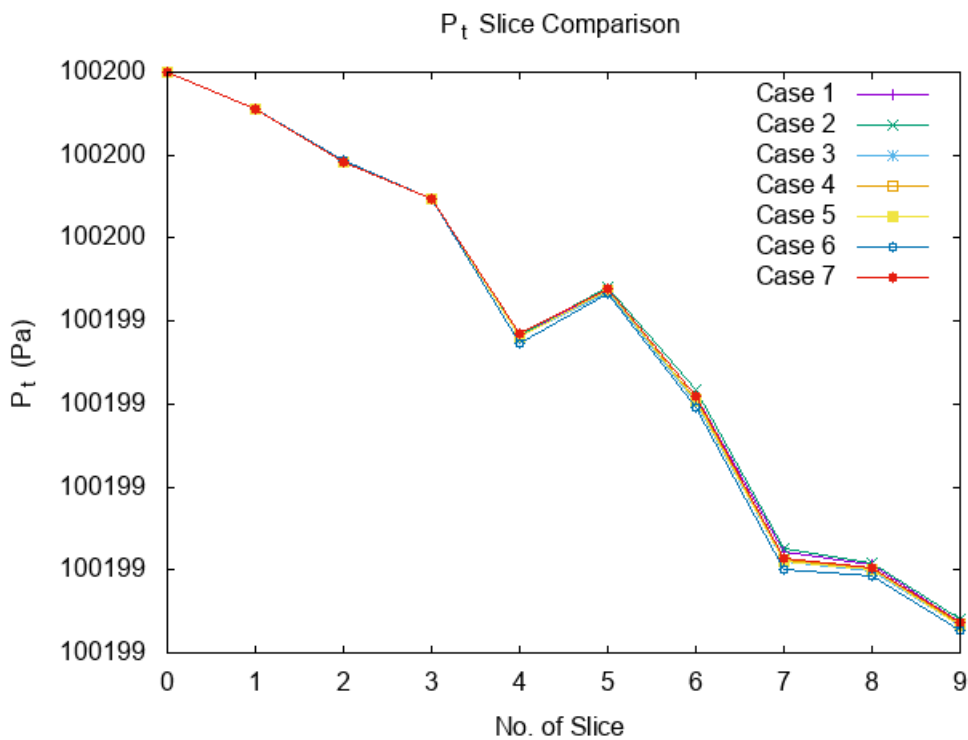
**Σχήμα 3.5:** Διάγραμμα απωλειών ολικής πίεσης συναρτήσει του αριθμού των κόμβων. Κάθε σημείο αντιπροσωπεύει μια περίπτωση

### 3.2.2 Μελέτη 2

Στη δεύτερη μελέτη γίνεται σύγκριση των ίδιων περιπτώσεων με την πρώτη μελέτη, όμως η σύγκριση των αποτελεσμάτων έγινε με διαφορετικό τρόπο. Επιλέχθηκαν δέκα διατομές κατά μήκος του αγωγού (slices) - η είσοδος του αγωγού, οκτώ ενδιάμεσες διατομές και η έξοδος, στις οποίες υπολογίζεται η αριθμητική μέση τιμή της ολικής πίεσης ( $P_t$ ) (σχήμα 3.6). Σε κάθε περίπτωση, οι δέκα διατομές είναι ταυτοτικά ίδιες. Στο σχήμα 3.7 φαίνεται η σύγκριση της ολικής πίεσης σε κάθε διατομή, για τις επτά περιπτώσεις.



Σχήμα 3.6: Οι οκτώ ενδιάμεσες διατομές (slices)

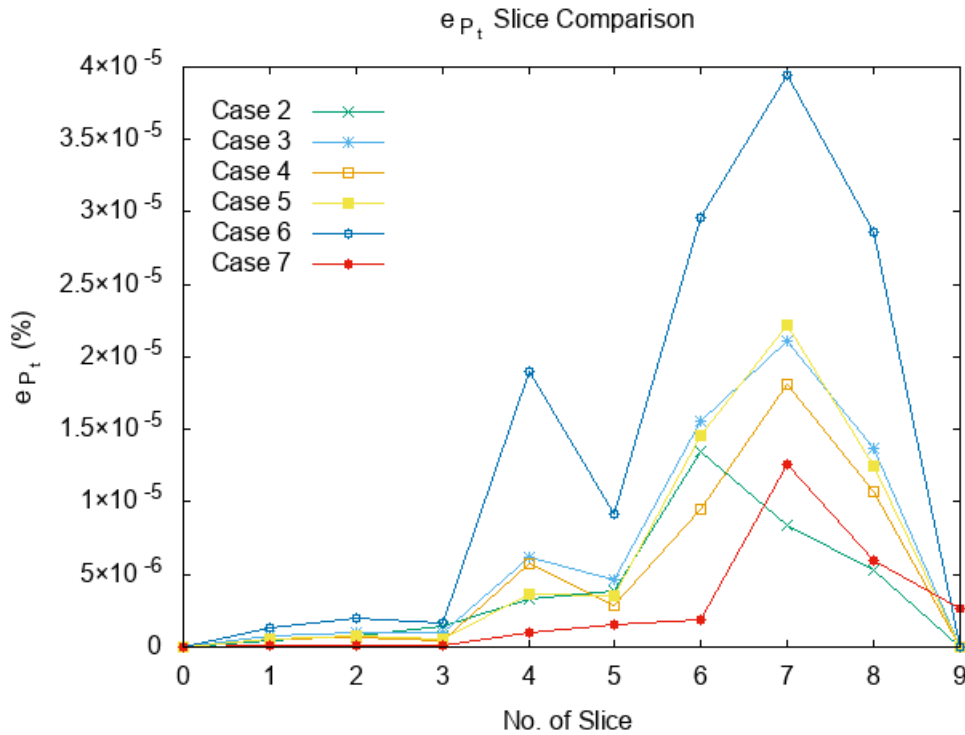


Σχήμα 3.7: Διάγραμμα ολικής πίεσης

Όπως φαίνεται, οι διαφορές είναι μικρές. Για τον λόγο αυτό, η σύγκριση γίνεται με την ποσοστιαία απόκλιση των περιπτώσεων 2 έως 7, σε σύγκριση με την πρώτη, η οποία

έχει το πυκνότερο πλέγμα, και άρα την ακριβέστερη λύση (σχήμα 3.8):

$$e_{pt} = \frac{|P_{t_{Case1}} - P_{t_{Casei}}|}{P_{t_{Case1}}} \cdot 100\%$$



Σχήμα 3.8: Διάγραμμα απόκλισης της ολικής πίεσης

Το αναμενόμενο αποτέλεσμα είναι πως για πλέγματα με μικρότερο αριθμό κόμβων θα υπήρχαν μεγαλύτερες αποκλίσεις. Αυτό, όμως, δεν συμβαίνει (λ.χ. η έκτη περίπτωση παρουσιάζει μεγαλύτερες αποκλίσεις από την έβδομη). Το συμπέρασμα που μπορεί να εξαχθεί είναι πως ο αριθμός των κόμβων στη διατομή έχει διαφορετική επίδραση στην ακρίβεια της λύσης από τον αριθμό των κόμβων κατά μήκος του αγωγού.

### 3.2.3 Μελέτη 3

Η τρίτη μελέτη χωρίζεται σε δύο μέρη. Το πρώτο μέρος πραγματεύεται την επίδραση των κόμβων κατά μήκος του αγωγού στην ακρίβεια της λύσης (Length Dependence), ενώ το δεύτερο με την επίδραση των κόμβων στη διατομή (Cross Section Dependence). Συγκρινόμενο μέγεθος είναι οι απώλειες ολικής πίεσης.

**Επίδραση του αριθμού των κόμβων κατά μήκος του αγωγού - Length Dependence** Όλες οι περιπτώσεις αυτού του μέρους της ανάλυσης αποτελούνται από τον ίδιο αριθμό κόμβων στη διατομή. Συγκεκριμένα:

- Περίπτωση 1:  $40 \times 40 \times 1400 = 2240000$
- Περίπτωση 2:  $40 \times 40 \times 1050 = 1680000$
- Περίπτωση 3:  $40 \times 40 \times 790 = 1264000$
- Περίπτωση 4:  $40 \times 40 \times 600 = 960000$
- Περίπτωση 5:  $40 \times 40 \times 450 = 720000$
- Περίπτωση 6:  $40 \times 40 \times 340 = 544000$
- Περίπτωση 7:  $40 \times 40 \times 250 = 400000$

**Επίδραση του αριθμού των κόμβων στη διατομή του αγωγού - Cross Section Dependence** Σε αυτό το μέρος της ανάλυσης, όλες οι περιπτώσεις αποτελούνται από τον ίδιο αριθμό κόμβων κατά μήκος του αγωγού. Συγκεκριμένα:

- Περίπτωση 1:  $60 \times 60 \times 600 = 2160000$
- Περίπτωση 2:  $55 \times 55 \times 600 = 1815000$
- Περίπτωση 3:  $45 \times 45 \times 600 = 1215000$
- Περίπτωση 4:  $40 \times 40 \times 600 = 960000$
- Περίπτωση 5:  $35 \times 35 \times 600 = 735000$
- Περίπτωση 6:  $30 \times 30 \times 600 = 540000$

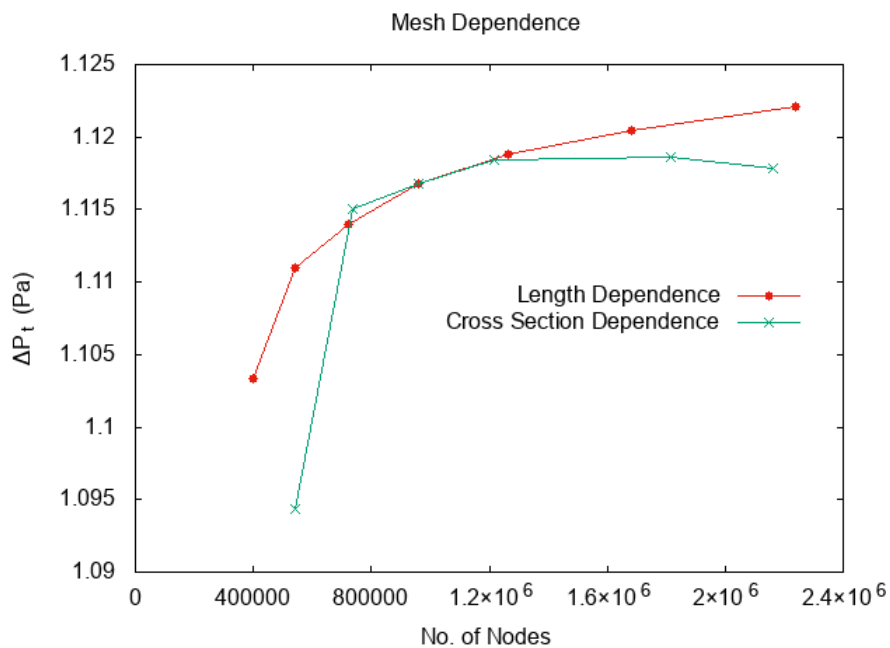
Στον πίνακα 3.1 παρουσιάζονται τα αποτελέσματα των υπολογιστικών προσομοιώσεων για κάθε μέρος της μελέτης. Για κάθε περίπτωση αναφέρεται ο χρόνος μέχρι την περάτωση, το μέγεθος  $y^+$ , οι απώλειες ολικής πίεσης σε Pa, και η παροχή μάζας που διέρχεται από τον αγωγό σε kg/s.

Στο σχήμα 3.9 φαίνεται η εξάρτηση της λύσης από το εκάστοτε πλέγμα:

Length Dependence					
Case	Total Nodes	Runtime	$\gamma_{plus}$	$\Delta P_t$	Massflow
1	2240000	1:43:49	12.19281	1.122111	0.015352
2	1680000	1:17:07	12.18598	1.120472	0.015362
3	1264000	0:58:31	12.18675	1.118858	0.015371
4	960000	0:43:38	12.18484	1.116823	0.015383
5	720000	0:38:40	12.19030	1.113991	0.015399
6	544000	0:25:15	12.19178	1.110998	0.015417
7	400000	0:19:59	12.20758	1.103341	0.015464

Cross Section Dependence					
Case	Total Nodes	Runtime	$\gamma_{plus}$	$\Delta P_t$	Massflow
1	2160000	1:37:59	8.69303	1.117802	0.015384
2	1815000	1:21:55	9.26148	1.118583	0.015373
3	1215000	0:55:41	10.99385	1.118452	0.015369
4	960000	0:43:38	12.18484	1.116823	0.015383
5	735000	0:34:39	13.42061	1.114998	0.015400
6	540000	0:25:11	16.26431	1.094338	0.015557

**Πίνακας 3.1:** Πίνακας αποτελεσμάτων - Επίδραση του αριθμού των κόμβων κατά μήκος και στη διατομή του αγωγού

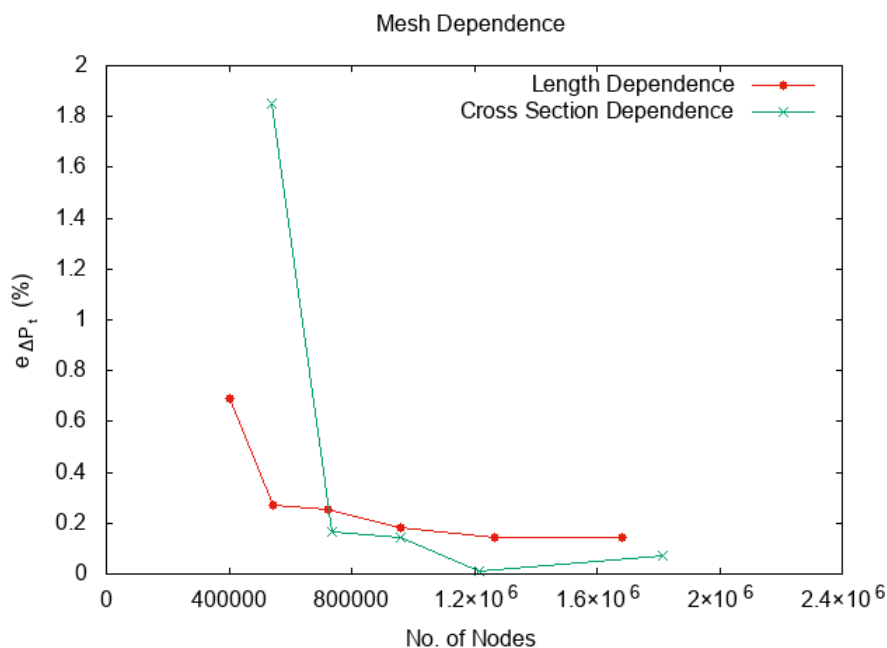


**Σχήμα 3.9:** Διάγραμμα απωλειών ολικής πίεσης συναρτήσει του συνολικού αριθμού των κόμβων

Από τα παραπάνω διαγράμματα επιβεβαιώνεται η παρατήρηση πως ο αριθμός των κόμβων στη διατομή έχει διαφορετική επίδραση στην ακρίβεια της λύσης από τον αριθμό των κόμβων κατά μήκος του αγωγού. Χρήσιμο μέγεθος σύγκρισης στην ανεξαρτησία του πλέγματος είναι η απόκλιση της λύσης του ελάχιστου πλέγματος από τη λύση του αμέσως

πυκνότερου. Αυτή η απόκλιση ορίζεται ως:

$$e_{\Delta P_t} = \frac{|\Delta P_{t_{Casei}} - \Delta P_{t_{Casei-1}}|}{\Delta P_{t_{Casei-1}}} \cdot 100\%$$



Σχήμα 3.10: Διάγραμμα απόκλισης

### 3.2.4 Συμπεράσματα

Από τις τρεις μελέτες που πραγματοποιήθηκαν, η τρίτη ήταν η πιο αποτελεσματική, καθώς δίνει πιο ξεκάθαρα αποτελέσματα. Συνεπώς, με βάση τα εξαχθέντα αποτελέσματα, μπορεί να γίνει η επιλογή του τελικού πλέγματος, κατά την οποία πρέπει να ληφθούν υπόψη η ακρίβεια, το υπολογιστικό κόστος, αλλά φυσικά και οι ιδιαιτερότητες που μπορεί να έχει η εφαρμογή. Κρίνοντας από τα σχήματα 3.9 και 3.10, η ανεξαρτησία του πλέγματος για τον αριθμό των κόμβων κατά μήκος του αγωγού δεν είναι ξεκάθαρη, καθώς τα σημεία έχουν μια γενθώς αύξουσα συμπεριφορά. Ωστόσο, η ακρίβεια που προσφέρει το πλέγμα της τέταρτης περίπτωσης (περίπτωση 4), είναι ικανοποιητική. Η ανεξαρτησία του πλέγματος για τον αριθμό των κόμβων στη διατομή είναι πιο εμφανής, καθώς για πλέγματα πυκνότερα από της πέμπτης περίπτωσης (περίπτωση 5), οι αποκλίσεις των λύσεων είναι αμελητέες. Έτσι, κρίνεται κατάλληλο το πλέγμα με αριθμό κόμβων:  $35 \times 35 \times 595 = 728875$ .

### 3.3 Σύγκριση Μοντέλων Τύρβης

Στην παράγραφο αυτή παρουσιάζεται η μελέτη που έγινε για τη σύγκριση των λύσεων που παράγει το μοντέλο Spalart-Allmaras ([10]) και το μοντέλο k- $\omega$  SST ([12]), ως ένα μέσο επιβεβαίωσης της ακρίβειας του μοντέλου Spalart-Allmaras.

Το μοντέλο τύρβης που χρησιμοποιείται κατά κύριο λόγο στην εργασία αυτή, είναι το μοντέλο Spalart-Allmaras. Το μοντέλο Spalart-Allmaras ως μοντέλο μίας εξίσωσης προτιμάται σε σχέση με το μοντέλο k- $\omega$  SST, ως προς την ταχύτητα επίλυσης. Φυσικά, για την πιστοποίηση της ακρίβειας της λύσης που παράγει το μοντέλο Spalart-Allmaras, πραγματοποιείται σύγκριση με το μοντέλο k- $\omega$  SST, το οποίο ως πολυχρησιμοποιημένο μοντέλο τύρβης, παράγει αποτελέσματα υψηλής ακρίβειας. Για τις προσομοιώσεις και των δύο περιπτώσεων, χρησιμοποιήθηκε το ίδιο πλέγμα, το οποίο είναι το τελικό πλέγμα της προηγούμενης παραγράφου.

Το μοντέλο Spalart-Allmaras, ως μοντέλο τύρβης μίας εξίσωσης, απαιτεί ως οριακή συνθήκη στην είσοδο του αγωγού ένα μέγεθος, το οποίο ονομάζεται λόγος συνεκτικότητας (viscosity ratio). Το μοντέλο k- $\omega$  SST ως μοντέλο δύο εξισώσεων απαιτεί ένα επιπλέον μέγεθος, που ονομάζεται ένταση της τύρβης (turbulence intensity). Η τιμή του μεγέθους αυτού εξαρτάται από την εκάστοτε εφαρμογή. Για τις δύο περιπτώσεις, ως οριακή συνθήκη του λόγου συνεκτικότητας στην είσοδο δόθηκε η ίδια τιμή. Για τον προσδιορισμό της κατάλληλης τιμής της έντασης της τύρβης ως οριακή συνθήκη στην είσοδο, πραγματοποιήθηκε η ανάλυση της παρακάτω παραγράφου.

#### 3.3.1 Προσδιορισμός της Έντασης της Τύρβης

Η ένταση της τύρβης ([13]), μέγεθος γνωστό και ως επίπεδο της τύρβης, ορίζεται ως:

$$I = \frac{\overline{u'}}{U}$$

όπου  $u'$  η RMS (Root Mean Square) τιμή των διαταραχών της τυρβώδους ταχύτητας και  $U$  η κατά Reynolds μέση ταχύτητα (Reynolds Averaged). Με γνωστή την κινητική ενέργεια της τύρβης  $k$ , μπορεί να υπολογιστεί το μέγεθος  $u'$  ως εξής:

$$u' = \sqrt{\frac{2}{3}k}$$

Το μέγεθος  $U$  μπορεί να υπολογιστεί ως:

$$U = \sqrt{U_x^2 + U_y^2 + U_z^2}$$

Για να οριστεί η οριακή συνθήκη της έντασης της τύρβης στην εκάστοτε εφαρμογή, συνήθως απαιτούνται μετρήσεις από προηγούμενως διεξαχθέντα πειράματα σε παρόμοια περίπτωση, ώστε να υπάρχει μια καλή εκτίμηση. Σε περιπτώσεις μέτριας τύρβης, κατηγορία που περιλαμβάνει ροές σε γεωμετρίες χαμηλής ή μέτριας πολυπλοκότητας, όπως λ.χ. σε αγωγούς, η ένταση της τύρβης συνήθως βρίσκεται στο εύρος 1% – 5%.

Πραγματοποιώντας βιβλιογραφική επισκόπηση([13]), για πλήρως ανεπτυγμένες ροές εντός αγωγών, η ένταση της τύρβης μπορεί να προσδιοριστεί ως :

$$I = 0.16Re_{dh}^{-\frac{1}{8}}$$

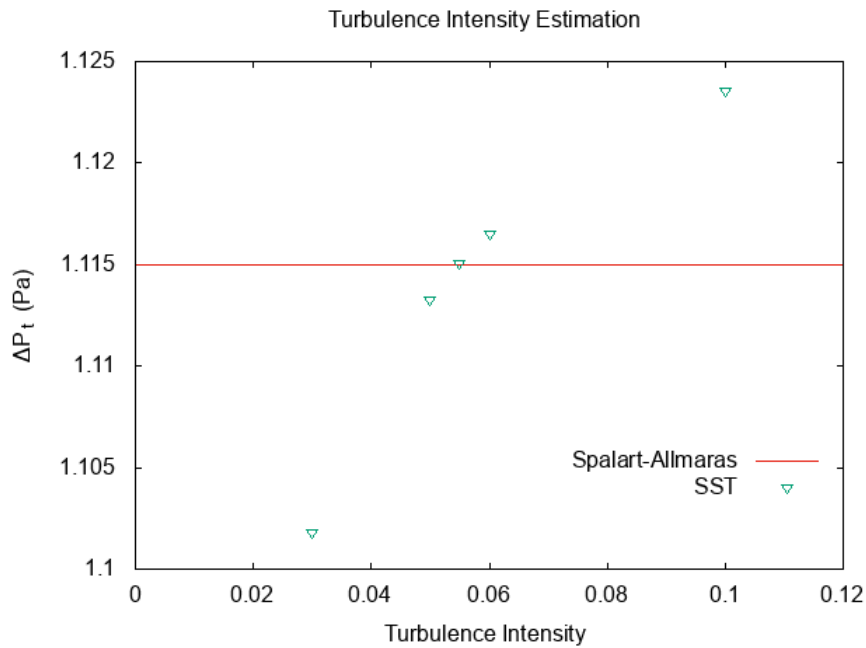
όπου  $Re_{dh}$  είναι ο αριθμός Reynolds βασισμένος στην υδραυλική διάμετρο  $D_h$ . Χρησιμοποιώντας την παραπάνω σχέση, για  $Re = 8290.5$ , υπολογίζεται:  $I = 0.0518$ . Παρακάτω, παρουσιάζεται η ανάλυση που πραγματοποιήθηκε για την επιβεβαίωση της παραπάνω σχέσης. Η κατάλληλη τιμή της έντασης της τύρβης, είναι εκείνη που παράγει στην τελική λύση τις ίδιες απώλειες ολικής πίεσης στον αγωγό με το μοντέλο τύρβης Spalart-Allmaras. Συνεπώς, με δοκιμές στην τιμή - κοντά στην τιμή που υπολογίστηκε παραπάνω - βρίσκεται η τελική τιμή της έντασης της τύρβης. Πραγματοποιήθηκαν πέντε δοκιμές και παρακάτω, στο σχήμα 3.2, φαίνεται ο πίνακας των αποτελεσμάτων, ενώ στο σχήμα 3.11 το διάγραμμα των απωλειών ολικής πίεσης συναρτήσει της έντασης της τύρβης. Υπενθυμίζονται οι οριακές συνθήκες:

- Είσοδος αγωγού (inlet) :  $P_t = 100200Pa$ , ροή παράλληλη με το κάθετο διάνυσμα της διατομής εισόδου, λόγος συνεκτικότητας (viscosity ratio) = 10 &  $I = const.$  για το μοντέλο k- $\omega$ . Για την τιμή του μεγέθους  $I$  πραγματοποιήθηκαν δοκιμές και τα αποτελέσματα φαίνονται στον πίνακα 3.2 και το σχήμα 3.11.
- Έξοδος αγωγού (outlet) :  $P_{st} = 100197.6Pa$
- Τοιχώματα (walls) : Αδιαβατικά

Turb Model	gplus	Runtime	Turb. Int.	Turb. Res.	Ptin (Pa)	Ptout (Pa)	$\Delta Pt$ (Pa)	Rel. Err.(SA)
Sp.-Allm.	13.42061	0:34:39	0	-31	100200	100198.9	1.114998	-
k- $\omega$ 1	15.15267	0:36:50	0.1	-25.4497	100200	100198.9	1.123551	0.767%
k- $\omega$ 2	15.18105	0:39:54	0.05	-25.2886	100200	100198.9	1.113264	0.156%
k- $\omega$ 3	15.20666	0:38:16	0.03	-25.2031	100200	100198.9	1.101806	1.183%
k- $\omega$ 4	15.17412	0:37:54	0.06	-25.3459	100200	100198.9	1.116513	0.136%
k- $\omega$ 5	15.17719	0:37:46	0.055	-25.3156	100200	100198.9	1.115057	0.005%

**Πίνακας 3.2:** Πίνακας αποτελεσμάτων - Σύγκριση μοντέλου Spalart-Allmaras και k- $\omega$





**Σχήμα 3.11:** Διάγραμμα απωλειών ολικής πίεσης συναρτήσει της έντασης της τύρβης στην είσοδο του αγωγού

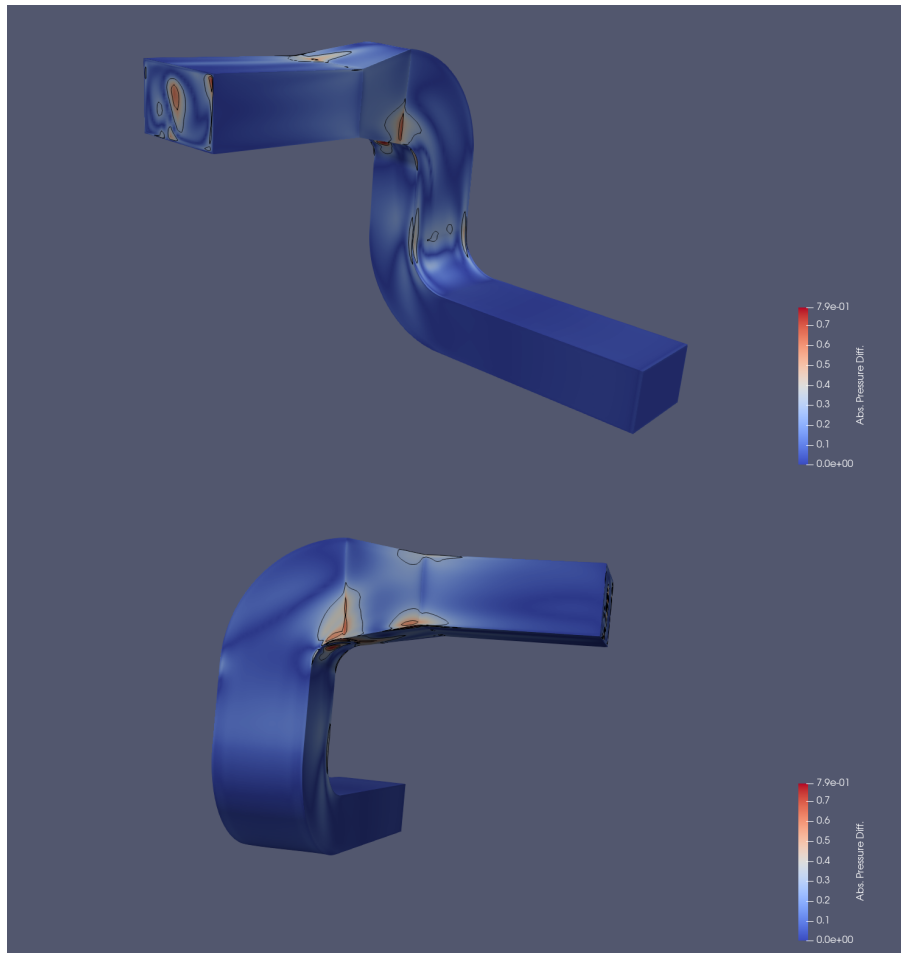
Όπως φαίνεται στα παραπάνω σχήματα, η πέμπτη δοκιμή με  $I = 0.055$  παράγει λύση που επαληθεύει τις απώλειες ολικής πίεσης. Με τις παραπάνω δοκιμές, φαίνεται να επιβεβαιώνεται (τουλάχιστον σε αυτήν την περίπτωση) η αξιοπιστία της σχέσης που χρησιμοποιήθηκε.

### 3.3.2 Σύγκριση Ολικών Πιέσεων

Έχοντας, πλέον, προσδιορίσει τη βασική παράμετρο του δεύτερου μοντέλου τύρβης, μπορεί να γίνει η σύγκριση των λύσεων που παράγει το κάθε μοντέλο. Συγκεκριμένα, συγκρίνονται οι τιμές της ολικής πίεσης των δύο προσομοιώσεων, δηλαδή σε κάθε κόμβο υπολογίζεται η απόκλιση:

$$e_{abs,p_t} = |P_{t,Spalart-Allmaras} - P_{t,SST}|$$

Στο σχήμα 3.12, παρουσιάζεται η παραπάνω διαφορά σε ισογραμμές, κατά μήκος του αγωγού:



Σχήμα 3.12: Ισογραμμές απόκλισης των ολικών πιέσεων σε δύο όψεις

Παρατηρείται πως εν γένει οι απόλυτες διαφορές είναι μικρές, όμως σε συγκεκριμένες περιοχές υπάρχουν και ελαφρώς αυξημένες αποκλίσεις. Για περαιτέρω σύγκριση, επιλέγονται εννιά εγχάρσιες διατομές (slices) (η είσοδος, η έξοδος και επτά ενδιάμεσες, οι οποίες είναι διαφορετικές από αυτές που παρουσιάστηκαν στην προηγούμενη παράγραφο), στις οποίες υπολογίζεται η σχετική απόκλιση της ολικής πίεσης, ως εξής:

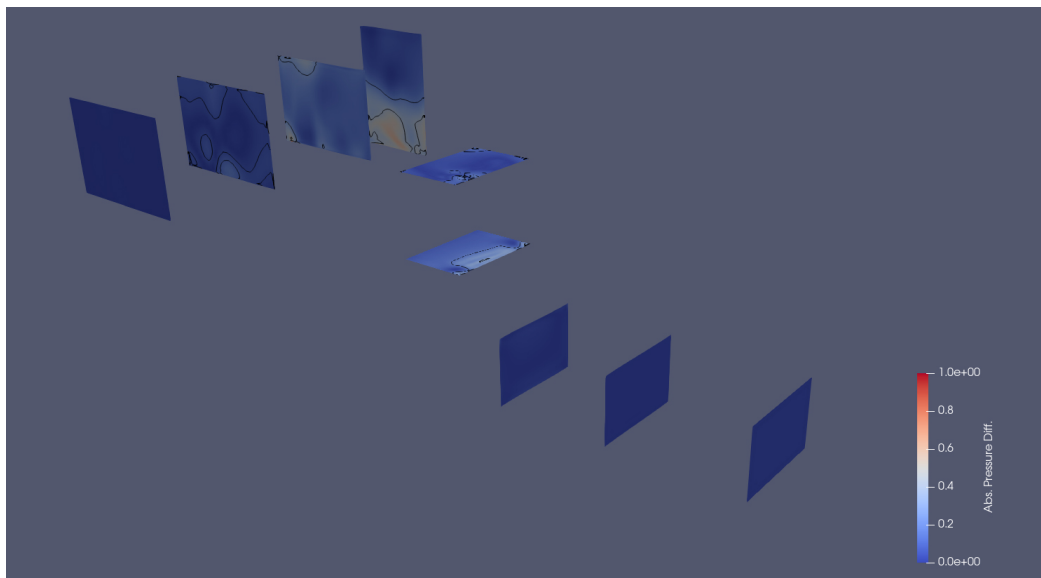
$$e_{rel,p} = \frac{|p_{SA} - p_{SST}|}{|p_{SA} - p_{inlet}|}$$

Όπου:

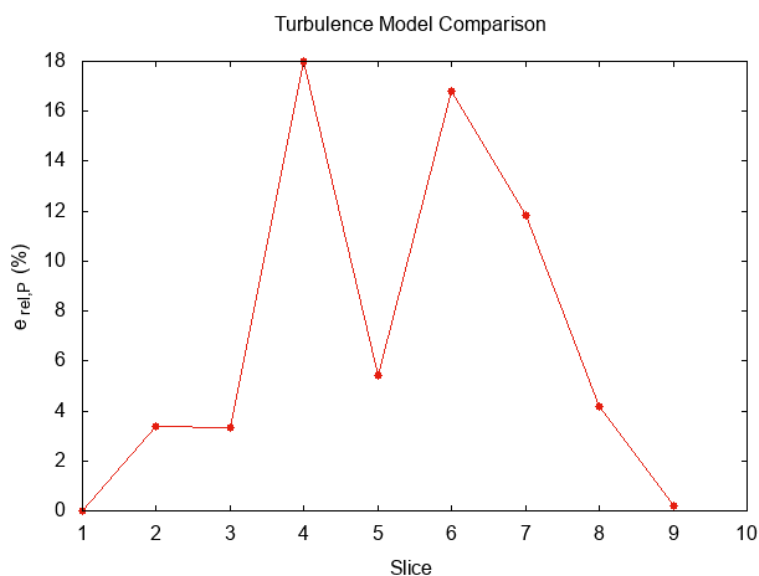
- $p_{SA} = \frac{\int_{slice} p_t dA}{A}$ : η μέση πίεση σε κάθε τομή, από το μοντέλο Spalart-Allmaras
- $p_{SST} = \frac{\int_{slice} p_t dA}{A}$ : η μέση πίεση σε κάθε τομή, από το μοντέλο k- $\omega$
- $p_{inlet}$ : η μέση πίεση στη διατομή εισόδου

Στο σχήμα 3.13 φαίνονται οι εν λόγω διατομές, με τις οποίες πραγματοποιείται εποπτεία

των αποκλίσεων των πιέσεων στο εσωτερικό του αγωγού, ενώ στο σχήμα 3.14 φαίνεται το διάγραμμα της σχετικής απόκλισης των δύο μοντέλων για την κάθε διατομή. Στον πίνακα 3.3 συγκρίνονται οι απώλειες ολικής πίεσης χρησιμοποιώντας τα δύο μοντέλα τύρβης.



**Σχήμα 3.13:** Οι εγκάρσιες διατομές και οι ισογραμμές των αποκλίσεων των ολικών πιέσεων



**Σχήμα 3.14:** Διάγραμμα σχετικής απόκλισης της ολικής πίεσης

	Sp.-Allm.	SST k- $\omega$
$\Delta P_t$ (Pa)	1.1150	1.1151

**Πίνακας 3.3:** Πίνακας αποτελεσμάτων - Σύγκριση απωλειών ολικής πίεσης

### 3.3.3 Συμπεράσματα

Μικρές διαφορές παρατηρούνται σε 'δύσκολα' σημεία για την επίλυση, όπως οι γωνίες του αγωγού (βλ. διατομή 4 και 6 στο σχήμα 3.13). Συνολικά, το πεδίο των πιέσεων και οι απώλειες ολικής πίεσης έχουν πολύ μικρή απόκλιση. Με τα παραπάνω, πιστοποιείται η ακρίβεια του μοντέλου Spalart-Allmaras και κρίνεται κατάλληλο για τη συγκεκριμένη εφαρμογή.

Παρακάτω, το μοντέλο τύρβης που χρησιμοποιείται είναι το Spalart-Allmaras.

## 3.4 Επιλογή της Τελικής Μοντελοποίησης

Η τελική μοντελοποίηση της ροής στο λογισμικό CFD PUMA, περιλαμβάνει:

- Πλέγμα:  $35 \times 35 \times 595 = 728875$  κόμβων
- Ασυμπίεστο ρευστό
- Μόνιμη, τυρβώδης ροή
- Μοντέλο τύρβης: Spalart-Allmaras
- CFD Runtime:  $\approx 35mins$  σε έναν επεξεργαστή GPU Nvidia Tesla K20m

## Κεφάλαιο 4

# Βελτιστοποίηση Γεωμετρίας του Αγωγού - ΜΑΕΑ & CFD

**Ορισμός της Βελτιστοποίησης** Ένα πρόβλημα βελτιστοποίησης με  $M_o$  αντικειμενικές συναρτήσεις ( $f$ ) και  $M_c$  περιορισμούς ( $c$ ) ανάγεται στην ελαχιστοποίηση του διανύσματος στόχου:

$$\min \vec{f}(\vec{b}) = (f_1(\vec{b}), \dots, f_{M_o}(\vec{b})) \in \mathbb{R}^{M_o}$$

$$\text{subject to } c_k(\vec{b}) \leq 0, k = 1, \dots, M_c$$

όπου  $\vec{b} \in \mathbb{R}^N$  το διάνυσμα των μεταβλητών σχεδιασμού και  $N$  ο αριθμός των μεταβλητών. Η βελτιστοποίηση δε λείπει από το χώρο της υπολογιστικής ρευστοδυναμικής και της αεροδυναμικής. Υπάρχουν πολλές μέθοδοι που έχουν αναπτυχθεί και χρησιμοποιούνται και κατηγοριοποιούνται στις αιτιοκρατικές (λ.χ. η μέθοδος καθόδου (Steepest Descent), η μέθοδος Newton, η συζυγής μέθοδος (Adjoint Method) ) και στις στοχαστικές. Στο πλαίσιο της εργασίας αυτής, χρησιμοποιούνται οι εξελικτικοί αλγόριθμοι, που είναι μια στοχαστική μέθοδος βελτιστοποίησης που χειρίζεται πληθυσμούς υποψήφιων λύσεων ([14]).

**Εξελικτικοί Αλγόριθμοι** Οι εξελικτικοί αλγόριθμοι (EA) ([14]) είναι μια στοχαστική μέθοδος βελτιστοποίησης. Τα βασικά τους χαρακτηριστικά είναι το ενδιαφέρον, μη-μαθηματικό υπόβαθρό τους, η ευκολία με την οποία προσαρμόζονται σε κάθε νέο πρόβλημα και, το βασικότερο, η δυνατότητά τους να μην εγκλωβίζονται σε τοπικά ακρότατα. Εμπνευσμένη από τη βιολογική εξελικτική θεωρία, η βασική λογική των EA είναι ότι χειρίζονται πληθυσμούς υποψήφιων λύσεων (population-based methods)

και όχι μια μεμονωμένη λύση σε κάθε επανάληψη. Οι ΕΑ αρχικά αναπτύχθηκαν για την επίλυση προβλημάτων ενός στόχου, όμως με κατάλληλες μετατροπές μπορούν να χειριστούν και προβλήματα πολλαπλών στόχων. Μία συνήθης κατηγορία εξελικτικών αλγορίθμων, ιδίως στην περιοχή της αεροδυναμικής, είναι οι γενετικοί αλγόριθμοι (GA) ([14]).

**Λογισμικό Βελτιστοποίησης EASY** Το λογισμικό που χρησιμοποιείται για την υλοποίηση της βελτιστοποίησης σε αυτή τη διπλωματική εργασία είναι η πλατφόρμα EASY, που έχει δημιουργηθεί και αναπτύσσεται στη ΜΠΤΡΒ/ΕΜΠ. Το λογισμικό EASY είναι μια πλατφόρμα που βασίζεται σε εξελικτικούς αλγορίθμους ως το κύριο εργαλείο και υποστηρίζει βελτιστοποιήσεις ενός στόχου (Single Objective Optimization - SOO), αλλά και πολλαπλών (Multi Objective Optimization - MOO) ([15], [16], [17], [18]).

Κατά τη διαδικασία της βελτιστοποίησης, το λογισμικό EASY πρέπει να επικοινωνεί με το λογισμικό αξιολόγησης. Η επικοινωνία αυτή, είναι ευθύνη του χρήστη, και πρέπει να γίνει με προσοχή. Το λογισμικό EASY, για να πραγματοποιήσει μία αξιολόγηση, παράγει το αρχείο "task.dat", το οποίο περιέχει τις τιμές των μεταβλητών σχεδιασμού και έχει τη μορφή του σχήματος 4.1, ενώ αντίστοιχα διαβάζει το αρχείο "task.res", το οποίο περιέχει τις τιμές των συναρτήσεων-στόχου και έχει τη μορφή που φαίνεται στο σχήμα 4.1. Τα λογισμικά αξιολόγησης χρησιμοποιούν συνήθως δική τους μορφολογία αρχείων, και για τον λόγο αυτό, το λογισμικό EASY καλεί ένα script, με το οποίο ο χρήστης κάνει τις απαραίτητες αλλαγές στη μορφή των αρχείων.

"task.dat"	"task.res"
$N$	$f_1$
$b_1$	$f_2$
$b_2$	$:$
$:$	$f_{M_o}$
$b_N$	

**Σχήμα 4.1:** Η μορφή των αρχείων "task.dat" και "task.res"

**Μεταπρότυπα** Οι εξελικτικοί αλγόριθμοι είναι μια πολύ αποτελεσματική μέθοδος εύρεσης βελτιστοποιημένων μορφών σε προβλήματα ρευστοδυναμικής και όχι μόνο. Για την εξαγωγή της τελικής λύσης, απαιτείται ένας αριθμός αξιολογήσεων από το λογισμικό αξιολόγησης, που στην προκειμένη περίπτωση είναι το λογισμικό CFD PUMA. Η διαδικασία αξιολόγησης των λύσεων από τα λογισμικά αξιολόγησης είναι συνήθως χρονοβόρα και έχει υψηλό υπολογιστικό κόστος. Για τον λόγο αυτόν, εντάσσονται τα μεταπρότυπα στη φάση της προσεγγιστικής προ-αξιολόγησης (Low-Cost Pre-Evaluation - LCPE), τα οποία έχουν σαφώς μικρότερο υπολογιστικό κόστος από το λογισμικό

αξιολόγησης. Τα μεταπρότυπα εκπαιδεύονται από τις υποψήφιας λύσεις που έχουν ήδη αξιολογηθεί, και χρησιμοποιούνται συνδυαστικά με το λογισμικό αξιολόγησης, βοηθώντας έτσι τον εξελικτικό αλγόριθμο να καταλήξει στην τελική λύση γρηγορότερα. Για την εργασία αυτή, επιλέχθηκαν τα δίκτυα συναρτήσεων ακτινικής βάσης Radial Basis Function networks - RBFN ([19], [20]), τα οποία εκπαιδεύονται σε κάθε υποψήφια λύση και λαμβάνουν υπόψη τις παραγώγους ευαισθησίας της κάθε μεταβλητής σχεδιασμού στη συνάρτηση κόστους. Ο εξελικτικός αλγόριθμος που υποβοηθάται από μεταπρότυπα ονομάζεται Metamodel-Assisted Evolutionary Algorithm και στο εξής αναφέρεται ως MAEA.

## 4.1 Περιγραφή της Περίπτωσης

Η βελτιστοποίηση γίνεται με σκοπό την εύρεση της λύσης που επιτυγχάνει την ελάχιστη τιμή της συνάρτησης-στόχου. Στην προκειμένη περίπτωση, σκοπός είναι η εύρεση του αγωγού με τις ελάχιστες απώλειες ολικής πίεσης. Οι πέντε μεταβλητές σχεδιασμού (όπως παρουσιάστηκαν στο πρώτο κεφάλαιο), είναι:

- η πρώτη γωνία του αγωγού ( $a_1$ )
- η δεύτερη γωνία του αγωγού ( $a_2$ )
- η ακτίνα του πρώτου fillet ( $r_1$ )
- η ακτίνα του δεύτερου fillet ( $r_2$ )
- το μήκος στο οποίο προσαρμόζεται η διατομή με loft ( $l_1$ )

Οι μεταβλητές σχεδιασμού, συνήθως συγχροτούνται σε ένα διάνυσμα  $\vec{b}$ . Συνεπώς:

$$\vec{b} = [b_1 \ b_2 \ b_3 \ b_4 \ b_5] = [a_1 \ a_2 \ r_1 \ r_2 \ l_1]$$

**Εύρος μεταβλητών σχεδιασμού** Όπως αναφέρθηκε και στο πρώτο κεφάλαιο, η γεωμετρία δεν ορίζεται για οποιεσδήποτε τιμές των μεταβλητών σχεδιασμού. Για παράδειγμα, η πρώτη γωνία δεν μπορεί να ισούται με  $180^\circ$  και η δεύτερη με  $30^\circ$ , καθώς δεν θα ορίζεται η γεωμετρία. Έτσι, σημαντικό είναι να οριστεί ένα επιτρεπτό εύρος των τιμών των μεταβλητών σχεδιασμού. Το εύρος αυτό, γνωστό και ως χώρος των μεταβλητών, επιλέγεται να είναι το εξής:

- $80^\circ < b_1 < 111^\circ$
- $80^\circ < b_2 < 111^\circ$
- $0.08m < b_3 < 0.12m$
- $0.08m < b_4 < 0.12m$
- $0.08m < b_5 < 0.12m$

**Συνάρτηση Κόστους** Η εφαρμογή ανάγεται σε βελτιστοποίηση ενός στόχου, χωρίς περιορισμούς (constraints). Συνάρτηση-στόχος της βελτιστοποίησης είναι οι απώλειες ολικής πίεσης του αγωγού:

$$F_{obj} = \Delta P_t$$

όπου  $\Delta P_t = P_{t,in} - P_{t,out}$

**Κωδικοποίηση μεταβλητών** Ένας εξελικτικός αλγόριθμος ψάχνει τη λύση μέσα στο χώρο των μεταβλητών σχεδιασμού, ο οποίος διακριτοποιείται με δυαδικά ψηφία. Για κάθε μεταβλητή ορίζεται ένας αριθμός δυαδικών ψηφίων, επιτυγχάνοντας μια αντίστοιχη ακρίβεια. Η επιλογή των δυαδικών ψηφίων κάθε μεταβλητής χρήζει προσοχής, καθώς με λίγα δυαδικά ψηφία ο αλγόριθμος δεν θα μπορέσει να βρει τη βέλτιστη λύση, ενώ με πολλά η βελτιστοποίηση θα γίνει χρονοβόρα. Η ακρίβεια της διακριτοποίησης δίνεται από τον παρακάτω τύπο:

$$\delta b = \frac{b_{upper-lim} - b_{lower-lim}}{2^k - 1}$$

όπου  $b_{upper-lim}$  το άνω όριο του εύρους μιας μεταβλητής,  $b_{lower-lim}$  το κάτω όριο του εύρους και  $k$  ο αριθμός των bits.

Η διακριτοποίηση των μεταβλητών που επιλέχθηκε για την εφαρμογή, φαίνεται παρακάτω:

- $b_1$  :  $k = 5$ , που αντιστοιχεί σε ακρίβεια  $1^\circ$
- $b_2$  :  $k = 5$ , που αντιστοιχεί σε ακρίβεια  $1^\circ$
- $b_3$  :  $k = 7$ , που αντιστοιχεί σε ακρίβεια  $\approx 3.15 \cdot 10^{-4}m$
- $b_4$  :  $k = 7$ , που αντιστοιχεί σε ακρίβεια  $\approx 3.15 \cdot 10^{-4}m$
- $b_5$  :  $k = 7$ , που αντιστοιχεί σε ακρίβεια  $\approx 3.15 \cdot 10^{-4}m$

**Ρυθμίσεις Εξελικτικού αλγόριθμου** Στην παράγραφο αυτή, αναφέρονται συνοπτικά οι ρυθμίσεις που επιλέχθηκαν για τον εξελικτικό αλγόριθμο.

- Αριθμός δήμεων: 1
- Αριθμός γονέων: 12
- Αριθμός απογόνων: 22
- Αριθμός γονέων για έναν απόγονο: 3
- Αριθμός Επίλεκτων: 1
- Πιθανότητα για επιλογή Επίλεκτων: 0.1
- Tournament size: 3

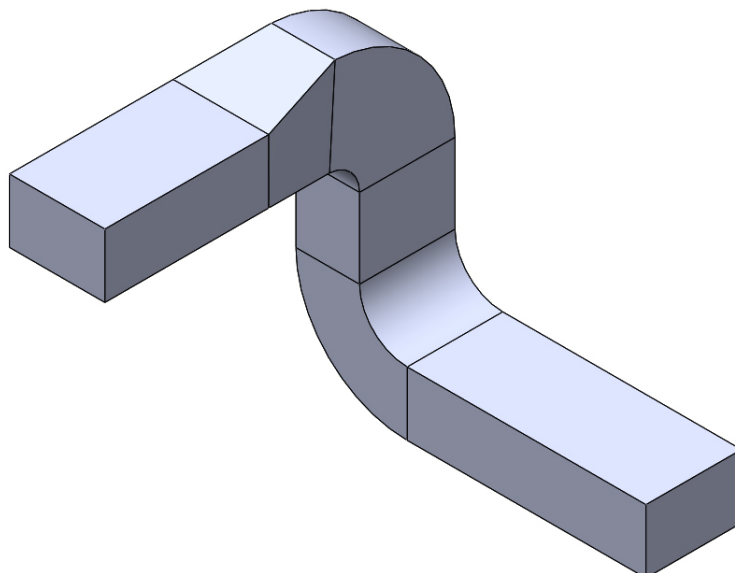


- Tournament probability: 0.9
- Κωδικοποίηση: Binary-Gray
- Πιθανότητα διασταύρωσης: 0.95 - Two point/variable
- Πιθανότητα μετάλλαξης: 0.05
- Τύπος μεταπροτύπων: RBF
- Exact evaluation: Min:2 Max:4 (ακριβείς αξιολογήσεις ανά γενιά και πληθυσμό)
- Training Pattern: Min:14 Max:20 (δείγματα εκπαίδευσης μεταπροτύπων)
- IPE pause gen.: 3 (αδρανείς γενιές για την παύση των μεταπροτύπων)
- Minimum Database entries: 40 (ελάχιστες αξιολογήσεις, στις οποίες εκπαιδεύονται τα μεταπρότυπα, για την εκκίνησή τους)

**Baseline Γεωμετρία** Η Baseline γεωμετρία είναι η πρώτη υποψήφια λύση στον εξελικτικό αλγόριθμο (σχήμα 4.2). Εν προκειμένω:

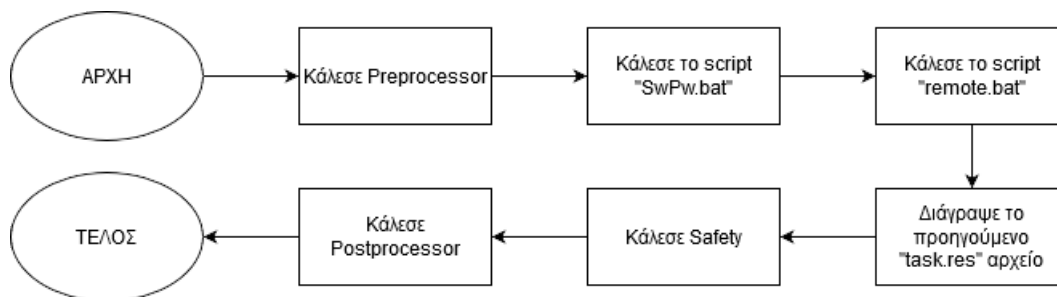
$$\vec{b} = [b_1 \ b_2 \ b_3 \ b_4 \ b_5] = [90 \ 90 \ 0.09984^* \ 0.09984^* \ 0.09984^*]$$

\* Η διακριτοποίηση με τη συγκεκριμένη δυαδική κωδικοποίηση δεν πιάνει τον αριθμό 0.1



Σχήμα 4.2: Η Baseline γεωμετρία

**Υλοποίηση της διαδικασίας** Η υλοποίηση της βελτιστοποίησης περιέχει την προετοιμασία του script \*.bat, με το οποίο το λογισμικό EASY λαμβάνει το αρχείο "task.res". Το script ονομάστηκε "main.bat" και ακολουθεί το παρακάτω διάγραμμα ροής:

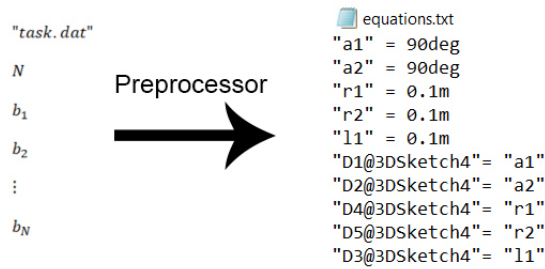


**Σχήμα 4.3:** Το διάγραμμα ροής του script "main.bat"

Παρακάτω, γίνεται επεξήγηση των συνιστωσών:

- Preprocessor: Script, το οποίο δέχεται ως είσοδο το αρχείο "task.dat" του EASY και παράγει το αρχείο "equations.txt", με το οποίο παράγεται η γεωμετρία (σχήμα 4.4)
- "SwPw.bat": Script, το οποίο παράγει αυτόματα τη γεωμετρία και το πλέγμα (βλ. παράγραφο 2.3)
- "remote.bat": Script, του οποίου το διάγραμμα ροής φαίνεται στο σχήμα 4.5
  - "eval.bat": Script, το οποίο πραγματοποιεί σύνδεση με το υπολογιστικό σύστημα της ΜΠΥΡΒ/ΕΜΠ και εκτελεί το script "task.sh"
  - "task.sh": Script, του οποίου το διάγραμμα ροής φαίνεται στο σχήμα 4.7
- Safety: Script, το οποίο ελέγχει εάν το αρχείο που λήφθηκε από το υπολογιστικό σύστημα της ΜΠΥΡΒ περιέχει πλήρη σύγκλιση του λογισμικού PUMA. Το αρχείου εξόδου του PUMA που λαμβάνεται είναι το αρχείο της μορφής '\*.conν' που φαίνεται στο σχήμα 4.6. Εάν επιστραφεί αρχείο '\*.conν' που δεν περιέχει όλο το πλήθος των επαναλήψεων, τότε η υποψήφια λύση απορρίπτεται.
- Postprocessor: Script, το οποίο δέχεται ως είσοδο το αρχείο εξόδου του λογισμικού PUMA '\*.conν' και παράγει το αρχείο "task.res", χρησιμοποιώντας τις στήλες 6 και 7 που περιέχουν τη μέση ολική πίεση εισόδου και εξόδου αντίστοιχα (σχήμα 4.6).

Τα απαραίτητα scripts από τα παραπάνω, περιλαμβάνονται στο παράρτημα Α'.



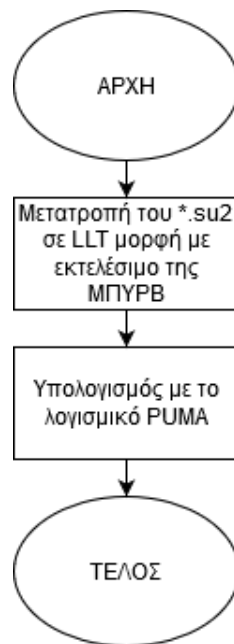
Σχήμα 4.4: Η λειτουργία του script Preprocessor



Σχήμα 4.5: Το διάγραμμα ροής του script "remote.bat"

```
# variables =
# ( 1 ) iterations,
# ( 2- 3 ) mass flow,
# ( 4- 5 ) static pressure,
# ( 6- 7 ) total pressure,
# ( 8- 9 ) total pressure (relative),
# ( 10 ) pressure losses
# ( 11 ) Hist
# ( 12 ) outlet peripheral/axial angle (swirl)
# ( 13 ) outlet radial/axial angle
# ( 14 ) outlet angle (RR)
# ( 15 ) mass flow (kg/s)
# zone t = OptimizationCycle0
100, -0.003156982372486, 0.010976508959630,
200, -0.008815726522231, 0.003820575868467,
300, -0.013315077566530, 0.006336171193308,
. . .
4800, -0.014066654879505, 0.014080168603655,
4900, -0.014069371872906, 0.014077529636717,
5000, -0.014071064806585, 0.014076527014875,
```

Σχήμα 4.6: Μορφή του αρχείου '\*.conv'



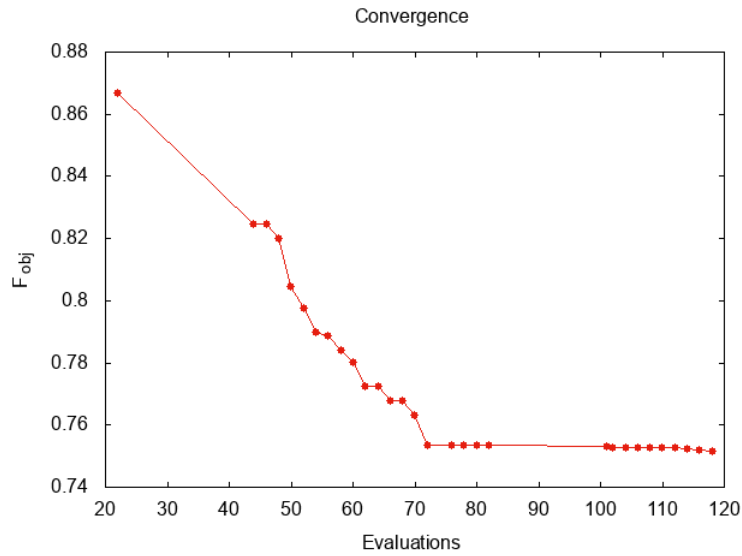
Σχήμα 4.7: Το διάγραμμα ροής του script "task.sh"

Με την παραπάνω διαδικασία, επικοινωνούν τέσσερα διαφορετικά λογισμικά (EASY, Solidworks, Pointwise, PUMA), σε συνολικά 2 συστήματα (το τοπικό και του εργαστηρίου). Τα λογισμικά EASY, Solidworks και Pointwise τρέχουν στον προσωπικό υπολογιστή του χρήστη (PC), ενώ το λογισμικό PUMA εκτελείται στη συστοιχία της ΜΠΥΡΒ/ΕΜΠ.

## 4.2 Αποτελέσματα και Τελική Λύση

### 4.2.1 Αποτελέσματα

Ο εξελικτικός αλγόριθμος τερματίστηκε στη γενιά 40. Η κάθε αξιολόγηση χρειάζεται περίπου 35 min σε έναν επεξεργαστή GPU Nvidia Tesla K20m, ενώ πραγματοποιήθηκαν συνολικά 167 ακριβείς αξιολογήσεις (exact evaluations). Έτσι, το κόστος της βελτιστοποίησης σε πραγματικό χρόνο υπολογίζεται στις 97 ώρες (περίπου τέσσερις ημέρες). Η πορεία της βελτιστοποίησης φαίνεται στο σχήμα 4.8.



**Σχήμα 4.8:** Η πορεία της βελτιστοποίησης. Στον οριζόντιο άξονα φαίνεται ο αριθμός των αξιολογήσεων, ενώ στον κατακόρυφο η καλύτερη λύση μέχρι την αντίστοιχη αξιολόγηση. Μετά την 118<sup>η</sup> αξιολόγηση δεν βρέθηκε καλύτερη λύση

Η τελική λύση είναι η παρακάτω:

$$\vec{b} = [b_1 \ b_2 \ b_3 \ b_4 \ b_5] = [111 \ 111 \ 0.12 \ 0.12 \ 0.12]$$

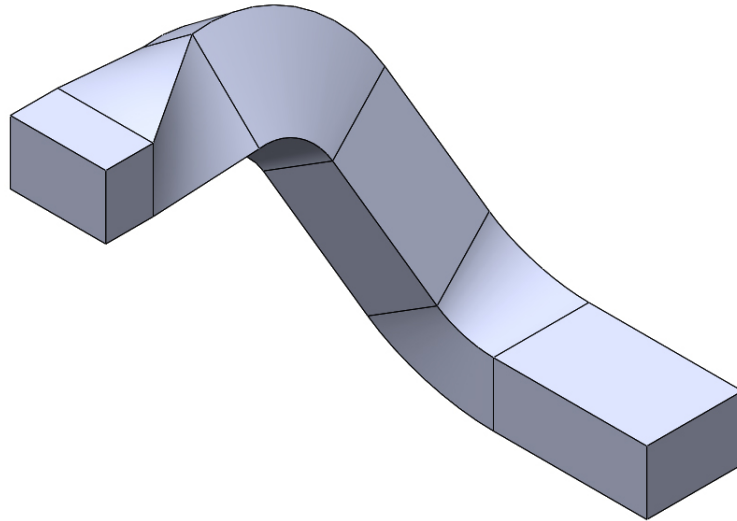
με τιμή της συνάρτησης-στόχου:

$$F_{obj} = 0.75139Pa$$

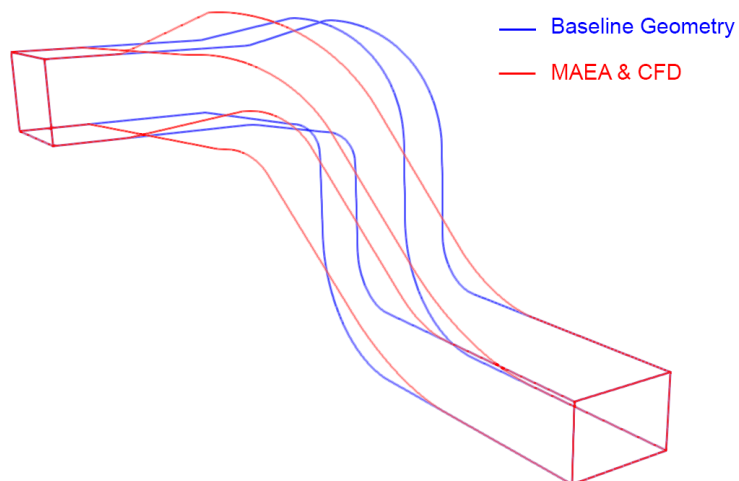
Η συνάρτηση-στόχος μειώθηκε κατά:

$$\frac{F_{baseline} - F_{final}}{F_{baseline}} \cdot 100\% \approx 32\%$$

Η μορφή του αγωγού της τελικής λύσης φαίνεται στο σχήμα 4.10. Ως φυσική ερμηνεία της τελικής λύσης, μπορεί να παρατηρηθεί πως με το συγκεκριμένο σετ γωνιών ( $a_1 = 111^\circ$  και  $a_2 = 111^\circ$ ) το συνολικό μήκος του αγωγού μειώνεται, με το συγκεκριμένο σετ fillet ( $r_1 = 0.12m$  και  $r_2 = 0.12m$ ) οι γωνίες είναι λιγότερο απότομες και με το μήκος του loft να είναι το μέγιστο ( $l_1 = 0.12m$ ), το ρευστό υποβάλλεται σε λιγότερο βίαιη μεταβολή της ορμής του.



Σχήμα 4.9: Η γεωμετρία της τελικής λύσης



Σχήμα 4.10: Η γεωμετρία της τελικής λύσης σε σύγκριση με την αρχική γεωμετρία

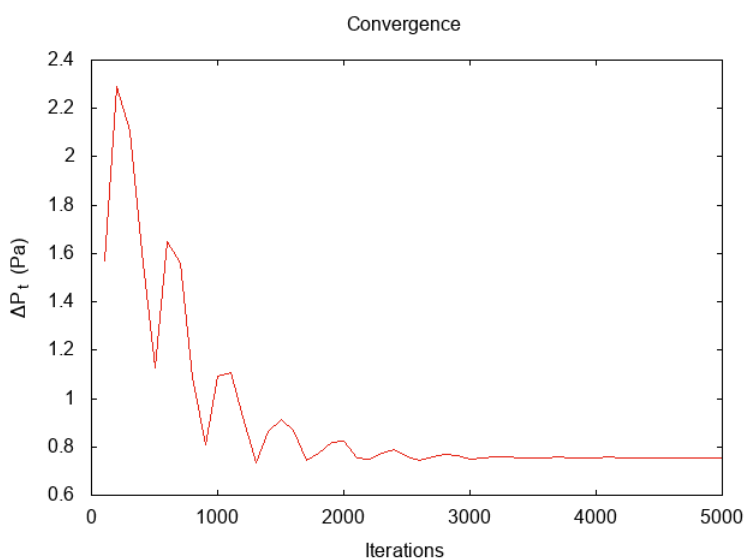
#### 4.2.2 Η ροή στον αγωγό της τελικής λύσης

Στην παράγραφο αυτή, παρουσιάζεται η σύγκλιση της επαναληπτικής επίλυσης της ροής για τον αγωγό της τελικής λύσης, καθώς επίσης και το πεδίο των πιέσεων.

**Σύγκλιση** Στο σχήμα 4.1 φαίνεται ένας πίνακας με τα υπόλοιπα (residuals) των εξισώσεων, στην αρχή της επίλυσης και στο πέρας (5000 επαναλήψεις), ενώ στο σχήμα 4.11 απεικονίζεται η σύγκλιση των απωλειών ολικής πίεσης.

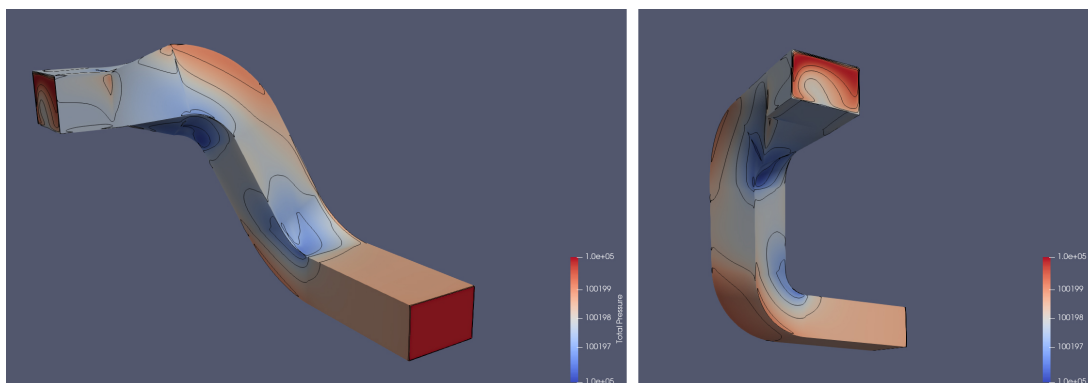
Iteration	Equation Residuals (Powers of 10)				
	Continuity	Momentum - x	Momentum - y	Momentum - z	Spalart-Allmaras
1	-12.3	-11.4	-12.8	-11.6	-18.5
5000	-29.5	-28.6	-29.1	-29.5	-31

**Πίνακας 4.1:** Οι δεκαδικοί λογάριθμοι των υπολοίπων των εξισώσεων στην αρχή και στο πέρας της υπολογιστικής προσομοίωσης



**Σχήμα 4.11:** Η σύγκλιση των απωλειών ολικής πίεσης συναρτήσει του αριθμού επαναλήψεων

**Πεδίο Ολικών Πίεσεων** Στο σχήμα Α'.1 απεικονίζεται το πεδίο των ολικών πιέσεων κατά μήκος του αγωγού με ισογραμμές.

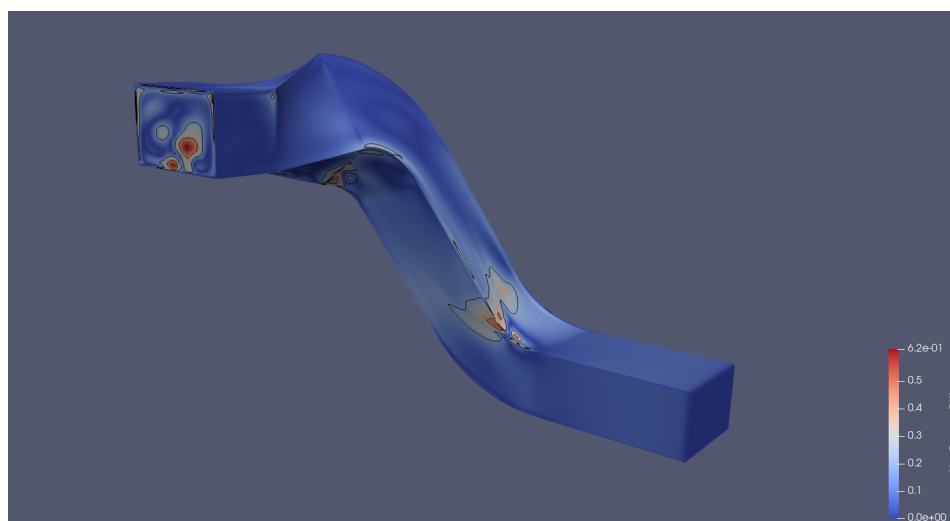


**Σχήμα 4.12:** Το πεδίο των πιέσεων στον αγωγό της τελικής λύσης

**Σύγκριση με μοντέλο k- $\omega$  SST** Για την πιστοποίηση της ακρίβειας της λύσης, συγκρίνονται τα πεδία ολικών πιέσεων που παράγουν τα μοντέλα Spalart-Allmaras και k- $\omega$  SST. Στον πίνακα 4.2 συγκρίνονται οι απώλειες ολικής πίεσης που παράγουν τα δύο μοντέλα, ενώ οι αποκλίσεις των ολικών πιέσεων κατά μήκος του αγωγού φαίνονται στο σχήμα 4.13.

	Sp.-Allm.	SST k- $\omega$
$\Delta Pt$ (Pa)	0.7553	0.7586

**Πίνακας 4.2:** Οι απώλειες ολικής πίεσης με τα μοντέλα Spalart-Allmaras και k- $\omega$  SST



**Σχήμα 4.13:** Οι αποκλίσεις των ολικών πιέσεων για τα δύο μοντέλα τύρβης



## Κεφάλαιο 5

# Πρόβλεψη της Ροής με χρήση Νευρωνικών Δικτύων

Το κεφάλαιο αυτό πραγματεύεται τη χρήση των νευρωνικών δικτύων στην πρόβλεψη του πεδίου πιέσεων στον αγωγό της εφαρμογής. Σε όλες τις εφαρμογές σχεδιασμού που ανάγονται σε προβλήματα ρευστομηχανικής, η υπολογιστική ρευστοδυναμική (CFD) δεν λείπει. Τα λογισμικά και οι επιλύτες CFD έχουν εξελιχθεί πολύ τα τελευταία χρόνια και χρησιμοποιούνται ακόμα και επεξεργαστές GPUs για την επίλυση ροών (λ.χ. PUMA). Ωστόσο, όπως παρουσιάστηκε σε προηγούμενα κεφάλαια, το υπολογιστικό κόστος είναι μεγάλο, ενώ η επίλυση της ροής χρονοβόρα.

Τα νευρωνικά δίκτυα (ANNs), και συγκεκριμένα τα δίκτυα DNNs, έχουν αποδειχθεί ικανά για την επίλυση σύνθετων, μη-γραμμικών προβλημάτων. Το γεγονός αυτό, σε συνδυασμό με το χαμηλό υπολογιστικό κόστος τους, κάνει τη χρήση τους σε προβλήματα ρευστομηχανικής μια λογική λύση. Φυσικά, ένα δίκτυο DNN για να είναι αξιόπιστο, πρέπει να συνοδεύεται από βάσεις δεδομένων (databases) με ποσότητα πληροφορίας που να αντεπεξέρχεται στην πολυπλοκότητα του προβλήματος. Για τη σύγχρονη βιομηχανία, αυτό δεν είναι πρόβλημα. Εταιρίες κρατούν αρχείο από παρόμοιους σχεδιασμούς εδώ και πολλά χρόνια, χτίζοντας αντίστοιχες βάσεις δεδομένων, οι οποίες χρησιμοποιούνται ακόμα και για πειραματικές εφαρμογές. Συνεπώς, τα λογισμικά επίλυσης της ροής αντικαθίστανται, σε ορισμένες περιπτώσεις, με τα χαμηλού υπολογιστικού κόστους νευρωνικά δίκτυα, τα οποία εκπαιδεύονται σε πλούσιες βάσεις δεδομένων.

Η χρήση των νευρωνικών δικτύων σε προβλήματα αεροδυναμικής/ρευστομηχανικής χωρίζεται σε δύο κατηγορίες. Η πρώτη κατηγορία περιλαμβάνει τη χρήση δικτύων ANN για την πρόβλεψη συγκεκριμένων ολοκληρωματικών μεγεθών, που περιγράφουν την εφαρμογή (λ.χ. την άνοση και την οπισθέλκουσα μιας αεροτομής (Lift - Drag) και τους αντίστοιχους συντελεστές  $C_L$ ,  $C_D$ ) [21]. Η δεύτερη κατηγορία, χρησιμοποιεί τα

DNNs (μια ειδική κατηγορία των ANN) για την πρόβλεψη ολόκληρων των πεδίων ροής (λ.χ. πεδίο των πιέσεων γύρω από μια πτέρυγα αεροπλάνου) ([15], [21]). Η εργασία αυτή εστιάζει στη δεύτερη κατηγορία.

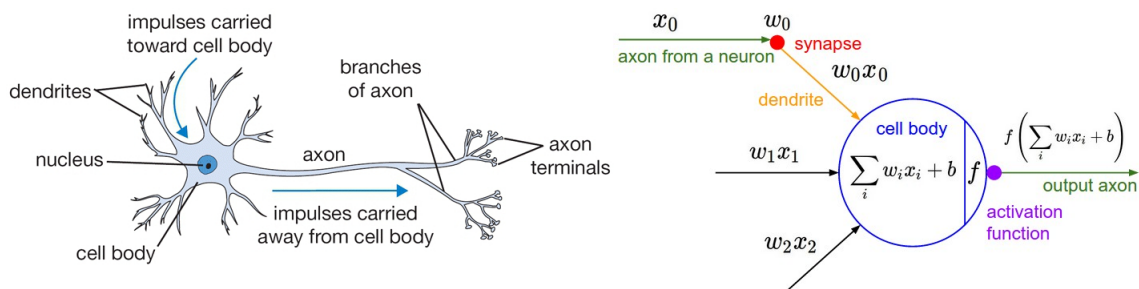
## 5.1 Τα βασικά των Νευρωνικών Δικτύων

### 5.1.1 Η βασική λειτουργία των Νευρωνικών Δικτύων

Τα νευρωνικά δίκτυα (Artificial Neural Networks - ANNs) είναι υπολογιστικά συστήματα εμπνευσμένα από τις βιολογικές νευρωνικές συνδέσεις ενός εγκεφάλου. Η υπολογιστική τους ικανότητα βασίζεται στους τεχνητούς νευρώνες (neurons), οι οποίοι οργανώνονται σε επίπεδα (layers) που συνδέονται μεταξύ τους. Η μοντελοποίηση των δικτύων βασίζεται στη δομή και στις συνάψεις των βιολογικών νευρώνων, οι οποίες μεταφέρουν πληροφορία/σήμα από και προς τους νευρώνες. Κάθε νευρώνας δέχεται σήματα εισόδου από τους δενδρίτες του και παράγει μια τιμή εξόδου στον άξονά του. Έπειτα, ο άξονας διακλαδίζεται και συνδέεται μέσω των συνάψεων σε δενδρίτες άλλων νευρώνων (σχήμα 5.1). Στο υπολογιστικό μοντέλο ενός νευρώνα, τα σήματα που μεταφέρονται στους άξονες (λ.χ.  $x_0$ ) αλληλεπιδρούν με την πράξη του πολλαπλασιασμού με τα βάρη (weights) των δενδριτών (λ.χ.  $w_0x_0$ ). Τα βάρη είναι τιμές που προσαρμόζονται κατά την εκπαίδευση, και ελέγχουν την επίδραση που έχει ο ένας νευρώνας στον άλλον. Σε κάθε νευρώνα αντιστοιχεί μια αριθμητική τιμή που ονομάζεται bias, η οποία προστίθεται στο παραπάνω γινόμενο. Τελικά, η τιμή που μεταφέρεται στον νευρώνα είναι το άθροισμα των γινομένων, συνυπολογίζοντας και την τιμή bias. Μέσω της συνάρτησης ενεργοποίησης (activation function), ο νευρώνας είτε ενεργοποιείται και εξάγει μια αριθμητική τιμή, είτε όχι, παράγοντας έτσι μηδενική τιμή. Συγκεκριμένα, η τιμή που εξάγεται (output) υπολογίζεται ως:

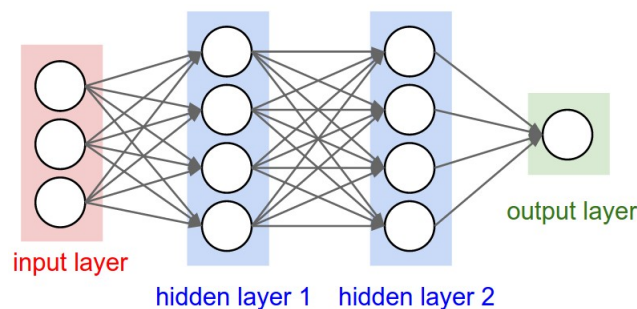
$$o_j = f \left( \sum_{i=1}^r w_{ij}x_i + b_j \right)$$

όπου  $w_{ij}$  το βάρος που συνδέει τον νευρώνα  $i$  με τον νευρώνα  $j$ ,  $x_i$  το σήμα που προέρχεται από τον νευρώνα  $i$ ,  $b_j$  η τιμή bias του  $j$  νευρώνα,  $f$  η συνάρτηση ενεργοποίησης και  $r$  ο αριθμός των νευρώνων που συνδέονται με τον  $j$  νευρώνα.



**Σχήμα 5.1:** Αριστερά: Αναπαράσταση του βιολογικού νευρώνα, Δεξιά: Αναπαράσταση του υπολογιστικού νευρώνα

Οι νευρώνες ομαδοποιούνται σε επίπεδα (layers), όπως φαίνεται στο σχήμα 5.2, συγκροτώντας έτσι ένα νευρωνικό δίκτυο. Το πρώτο επίπεδο ονομάζεται επίπεδο εισόδου (input layer), το τελευταίο επίπεδο ονομάζεται επίπεδο εξόδου (output layer), ενώ όλα τα ενδιάμεσα ονομάζονται κρυμμένα επίπεδα (hidden layers). Βαθύ ονομάζεται το νευρωνικό δίκτυο που περιέχει αρκετά (συνήθως παραπάνω από δύο) κρυμμένα επίπεδα. Ένα νευρωνικό δίκτυο μπορεί να αναπαρασταθεί από μια πολύ σύνθετη μαθηματική συνάρτηση, που περιλαμβάνει όλες τις μαθηματικές εκφράσεις του κάθε νευρώνα που το αποτελεί. Το σύνολο των παραμέτρων ενός δικτύου περιλαμβάνει τα βάρη και τις τιμές bias, που ρυθμίζονται κατά τη φάση της εκπαίδευσης.

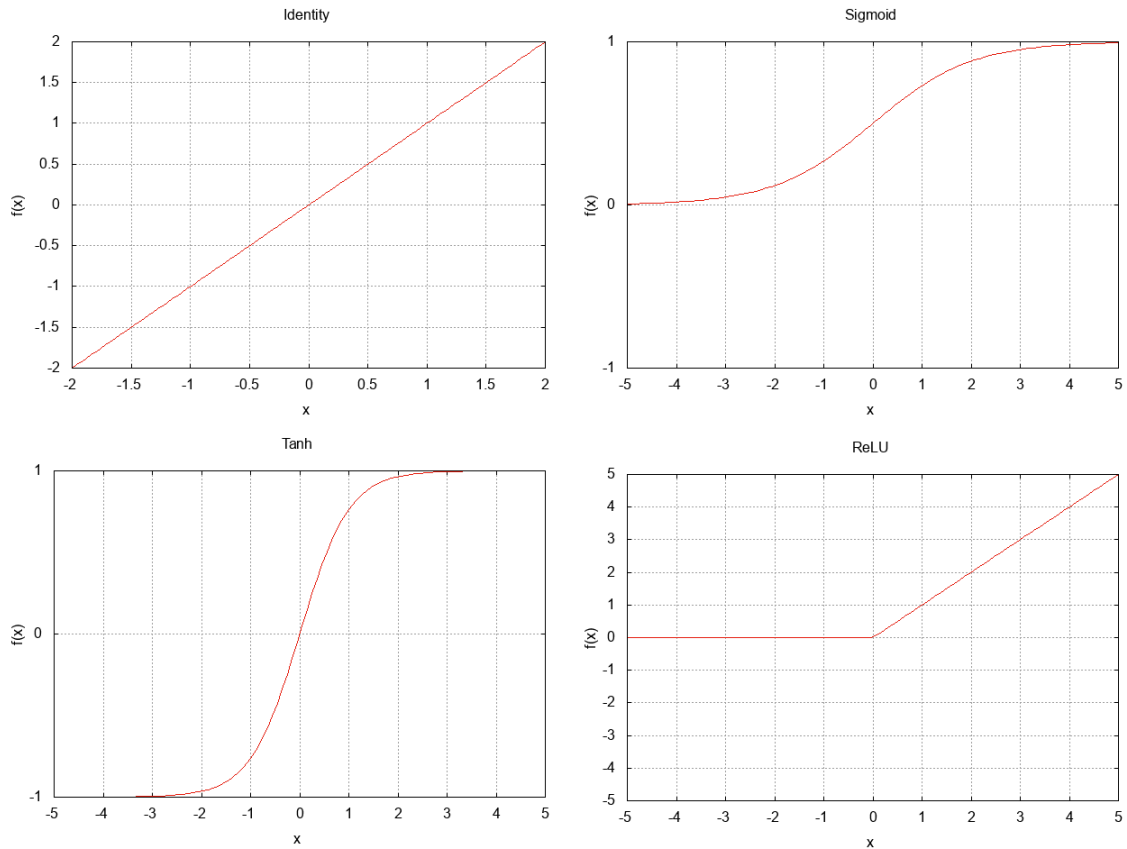


**Σχήμα 5.2:** Οι νευρώνες ομαδοποιούνται σε επίπεδα (layers)

### 5.1.2 Συναρτήσεις ενεργοποίησης

Υπάρχουν πολλές συναρτήσεις ενεργοποίησης που χρησιμοποιούνται αναλόγως την εφαρμογή. Οι πιο συνηθισμένες είναι οι παρακάτω:

- Ταυτοτική (Identity):  $f(x) = x$
- Σιγμοειδής (Sigmoid):  $f(x) = \frac{1}{1+e^{-x}}$
- Υπερβολική εφαπτομένη (Tanh):  $f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$
- Rectified Linear Unit - ReLU:  $f(x) = \begin{cases} 0 & , x < 0 \\ x & , x \geq 0 \end{cases}$



Σχήμα 5.3: Οι συναρτήσεις ενεργοποίησης

### 5.1.3 Εκπαίδευση

Η διεργασία που καλείται ένα δίκτυο να διεκπεραιώσει, είναι να εξάγει τις τιμές εξόδου ( $\vec{y}$ ), με δοσμένες τις τιμές εισόδου ( $\vec{x}$ ) - συχνά αυτή η διεργασία ονομάζεται πρόβλεψη. Κατά τη φάση της εκπαίδευσης, δίνονται στο δίκτυο οι τιμές που καλείται να προβλέψει (πραγματικές τιμές), οι οποίες αντιστοιχούν στις τιμές εισόδου. Ο γνωστός συνδυασμός τιμών εισόδου - πραγματικών τιμών εξόδου ονομάζεται δείγμα εκπαίδευσης (training pattern). Η απόκλιση της πρόβλεψης του δικτύου από τις πραγματικές τιμές υπολογίζεται με μία συνάρτηση κόστους (cost function - συχνά αναφερόμενη και ως loss function), η οποία βαθμολογεί την ικανότητα του δικτύου να προβλέψει τις πραγματικές τιμές. Ουσιαστικά, η εκπαίδευση ενός δικτύου ανάγεται σε πρόβλημα βελτιστοποίησης, με σκοπό την ελαχιστοποίηση της συνάρτησης κόστους. Αν  $N$  ο αριθμός των δειγμάτων εκπαίδευσης, τότε η συνάρτηση κόστους εκφράζεται ως:

$$C = \frac{1}{N} \sum_{i=1}^N C_i(w, b, \vec{x}^i, \vec{y}^i)$$

όπου  $C_i$  είναι η συνάρτηση κόστους υπολογισμένη για το  $i$  δείγμα εκπαίδευσης. Αν  $v$  μια παράμετρος που ρυθμίζεται κατά τη φάση της εκπαίδευσης, τότε η ανανέωση της τιμής της πραγματοποιείται με τη μέθοδο της απότομης καθόδου (steepest descent):

$$v^{new} = v^{old} - \eta \frac{\partial C}{\partial v}$$

όπου  $\eta$  ο ρυθμός μάθησης (learning rate).

Για την πρώτη εποχή της εκπαίδευσης, οι παράμετροι του δικτύου (βάρη και τιμές bias) αρχικοποιούνται συνήθως χρησιμοποιώντας τυχαία αρχικοποίηση. Η φάση της εκπαίδευσης έχει ως αποτέλεσμα το δίκτυο να προβλέπει με όσο το δυνατόν καλύτερη ακρίβεια τα δείγματα εκπαίδευσης.

#### 5.1.4 Συναρτήσεις κόστους

Υπάρχουν διάφορες συναρτήσεις κόστους που μπορούν να χρησιμοποιηθούν. Η επιλογή, κρίνεται αναλόγως την εκάστοτε εφαρμογή. Παρακάτω παρουσιάζονται συνήθεις συναρτήσεις κόστους. Με  $y_i$  συμβολίζεται η πραγματική (επιθυμητή) τιμή εξόδου, ενώ με  $\hat{y}_i$  η πρόβλεψη του δικτύου.

- Απόλυτη διαφορά (Absolute Difference):  $C_i = |y_i - \hat{y}_i|$
- Τετραγωνική απόκλιση (Mean Squared Error - MSE):  $C_i = (y_i - \hat{y}_i)^2$
- Τετραγωνική λογαριθμική απόκλιση:  $C_i = (\log(y_i + 1) - \log(\hat{y}_i + 1))^2$

#### 5.1.5 Οπισθοδιάδοση (Back-propagation)

Οι παράγωγοι που απαιτούνται για την εκπαίδευση των δικτύων μπορούν να εκφραστούν αναλυτικά. Ωστόσο, η αναλυτική έκφραση έχει αποδειχθεί πολύ αργή για την εκπαίδευση των δικτύων. Η πιο διαδεδομένη τεχνική για τον υπολογισμό των παραγώγων στα δίκτυα είναι ο αλγόριθμος της οπισθοδιάδοσης (back-propagation), ο οποίος είναι βασισμένος στον κανόνα της αλυσίδας. Η οπισθοδιάδοση στέλνει την πληροφορία από τη συνάρτηση κόστους από την έξοδο προς την είσοδο, σε όλους τους νευρώνες του δικτύου, προσαρμόζοντας έτσι τα βάρη και τις τιμές bias. Τα μαθηματικά σύμβολα που θα χρησιμοποιηθούν παρακάτω, είναι τα εξής:

- $w_{ij}^k$ : το βάρος για τον  $j$  νευρώνα του  $k$  επιπέδου, που συνδέεται με τον  $i$  νευρώνα του προηγούμενου
- $b_i^k$ : η τιμή bias του  $i$  νευρώνα του  $k$  επιπέδου
- $a_i^k$ : η αριθμητική τιμή που φτάνει στον  $i$  νευρώνα του  $k$  επιπέδου
- $o_i^k$ : η τιμή εξόδου του  $i$  νευρώνα του  $k$  επιπέδου

- $r^k$ : ο αριθμός των νευρώνων του  $k$  επιπέδου
- $f$ : η συνάρτηση ενεργοποίησης

Οι τιμές εξόδου των νευρώνων υπολογίζονται ως:

$$a_i^k = \sum_l^{r^k} w_{li}^k o_l^{k-1} + b_i^k$$

$$o_i^k = f(a_i^k)$$

Χάρην απλότητας, παρουσιάζεται η οπισθοδιάδοση ενός δικτύου με μία τιμή εξόδου. Το επίπεδο εξόδου (output layer) ενεργοποιείται από τις τιμές  $\hat{f} = f(a_i^m)$ . Υπενθυμίζεται πως η συνάρτηση κόστους  $C$  πρέπει να ελαχιστοποιηθεί, ενώ οι παράμετροι που πρέπει να ρυθμιστούν είναι τα βάρη  $w$  και οι τιμές bias  $b$ . Η παράγωγος ως προς τα βάρη υπολογίζεται ως:

$$\frac{\partial C}{\partial w_{ij}^k} = \frac{\partial C}{\partial a_j^k} \frac{\partial a_j^k}{\partial w_{ij}^k}$$

Οι παράγωγοι που εμφανίστηκαν, υπολογίζονται ως:

$$\frac{\partial a_j^k}{\partial w_{ij}^k} = \frac{\partial}{\partial w_{ij}^k} \left( \sum_{l=0}^{r^{k-1}} w_{lj}^k o_l^{k-1} + b_j^k \right) = o_i^{k-1}$$

$$\frac{\partial C}{\partial a_j^k} = \delta_j^k$$

όπου  $\delta_j^k$  είναι η απόκλιση του νευρώνα  $j$  στο επίπεδο  $k$  (όπου  $\delta$  το δέλτα του Kronecker). Συνεπώς, η παράγωγος του κόστους ως προς τα βάρη γράφεται ως:

$$\frac{\partial C}{\partial w_{ij}^k} = \delta_j^k o_i^{k-1}$$

Η απόκλιση  $\delta_j^k$  υπολογίζεται ως εξής: για το επίπεδο εξόδου εξαρτάται από τη συνάρτηση κόστους, ενώ για τα υπόλοιπα επίπεδα ισχύει:

$$\delta_j^k = \frac{\partial C}{\partial a_j^k} = \sum_{l=1}^{r^{k+1}} \frac{\partial C}{\partial a_l^{k+1}} \frac{\partial a_l^{k+1}}{\partial a_j^k}$$

ή,

$$\delta_j^k = \sum_{l=1}^{r^{k+1}} \delta_l^{k+1} \frac{\partial a_l^{k+1}}{\partial a_j^k}$$

και από τον ορισμό του  $a_l^{k+1} = \sum_{j=1}^{r^k} w_{jl}^{k+1} f(a_j^k) + b_l^{k+1}$ :

$$\frac{\partial a_l^{k+1}}{\partial a_j^k} = w_{jl}^{k+1} f'(a_j^k)$$

Συνθέτοντας όλα τα παραπάνω:

$$\delta_j^k = f'(a_j^k) \sum_{l=1}^{r^{k+1}} \delta_l^{k+1} w_{jl}^{k+1}$$

Οι παράγωγοι της συνάρτησης κόστους ως προς τις τιμές bias είναι:

$$\frac{\partial C}{\partial b_j^k} = \frac{\partial C}{\partial a_j^k} \frac{\partial a_j^k}{\partial b_j^k} = \delta_j^k \frac{\partial a_j^k}{\partial b_j^k}$$

όμως:  $\frac{\partial a_j^k}{\partial b_j^k} = 1$ , συνεπώς:

$$\frac{\partial C}{\partial b_j^k} = f'(a_j^k) \sum_{l=1}^{r^{k+1}} \delta_l^{k+1} w_{jl}^{k+1}$$

Από τις τελικές εκφράσεις των παραγώγων, είναι εμφανές πως η πληροφορία ρέει από την έξοδο του εκάστοτε επιπέδου προς την είσοδο (εξ ου και η ονομασία του αλγορίθμου). Η παραπάνω διαδικασία επαναλαμβάνεται για όλα τα επίπεδα, μέχρι το επίπεδο εισόδου. Έπειτα, μέσω της εκπαίδευσης οι τιμές των παραμέτρων ανανεώνονται. Επειδή, όπως φάνηκε, οι παράγωγοι και οι αποκλίσεις εξαρτώνται από τις τιμές εξόδου των νευρώνων, η οπισθοδιάδοση διεκπεραιώνεται μετά τον υπολογισμό της τιμής εξόδου του δικτύου κάθε φορά.

### 5.1.6 Gradient-Based Αλγόριθμοι Εκπαίδευσης

Η εκπαίδευση των νευρωνικών δικτύων γίνεται με τη χρήση Gradient-Based μεθόδων, οι οποίες είναι γνωστές για την ταχύτητα με την οποία βρίσκουν βέλτιστες λύσεις, όμως συχνά 'παγιδεύονται' σε τοπικά ακρότατα. Για να ξεπεραστεί αυτό το εμπόδιο, έχουν αναπτυχθεί κάποιοι αλγόριθμοι, οι οποίοι έχουν βασιστεί στη μέθοδο της απότομης

καθόδου (steepest descent).

- **Stochastic Gradient Descent (SGD) with Momentum:** Ο αλγόριθμος αυτός χρησιμοποιεί την όρμή από τον υπολογισμό των προηγούμενων παραγώγων, ώστε να υπάρχει μεταβολή ακόμα και όταν οι τοπικές παράγωγοι είναι μηδέν. Με τον τρόπο αυτό, ο αλγόριθμος μπορεί να ξεφύγει από τοπικά ελάχιστα. Βασίζεται στους εκθετικά σταθμισμένους μέσους (Exponentially Weighted Averages - EWA) και εστιάζει στη ρύθμιση των πιο σημαντικών παραμέτρων, ενώ μειώνει τις ταλαντώσεις της σύγκλισης. Το αρνητικό αυτής της μεθόδου είναι πως η όρμή μπορεί να είναι αρκετά μεγάλη, ώστε ο αλγόριθμος να μην σταματήσει ούτε στο ολικό ακρότατο.
- **RMSProp:** Ο αλγόριθμος Root Mean Squared Propagation χρησιμοποιεί τους εκθετικά σταθμισμένους μέσους (EWA) για να ρυθμίσει το ρυθμό μάθησης ξεχωριστά για την κάθε παράμετρο. Οι τιμές των παραμέτρων ρυθμίζονται από παραγώγους που υπολογίζονται από τις προηγούμενες επαναλήψεις, μέσω του τετραγωνικού μέσου όρου (Root Mean Square - RMS). Η μέθοδος αυτή μειώνει επίσης τις ταλαντώσεις στη σύγκλιση, όμως καθώς ο παρονομαστής του μέσου όρου μεγαλώνει με τις επαναλήψεις, ο ρυθμός μάθησης μπορεί να γίνει πολύ μικρός, 'παγώνοντας' έτσι τη μεταβολή των παραμέτρων.
- **Adam:** Ο αλγόριθμος Adaptive Moment Estimation - ADAM χρησιμοποιεί χαρακτηριστικά από τους παραπάνω δύο αλγόριθμους, εκμεταλλεύοντας τα πλεονεκτήματά τους. Η σύνθετη αυτή μέθοδος είναι γρήγορη και αποτελεσματική, όμως έχει μεγαλύτερο υπολογιστικό κόστος. Στο πλαίσιο της εργασίας αυτής, χρησιμοποιήθηκε κατά κόρον.

### 5.1.7 Βάση Δεδομένων (Database)

Όπως αναφέρθηκε παραπάνω, με τη διαδικασία της εκπαίδευσης, ένα νευρωνικό δίκτυο μπορεί να προσαρμόσει τις παραμέτρους του με τέτοιο τρόπο, ώστε να κάνει ακριβείς προβλέψεις σε σύνθετα και μη-γραμμικά προβλήματα. Στη διαδικασία της εκπαίδευσης, δίνονται στο δίκτυο οι τιμές εισόδου και οι επιθυμητές τιμές εξόδου. Ο συνδυασμός αυτός, όπως αναφέρθηκε, ονομάζεται δείγμα εκπαίδευσης (training pattern). Το δίκτυο, ύστερα, υπολογίζει την απόκλιση της πρόβλεψής του μέσω της συνάρτησης κόστους και οι παράμετροί του ρυθμίζονται κατάλληλα, ώστε οι προβλέψεις του να γίνουν πιο ακριβείς. Φυσικά, ένα πολύ απλό δίκτυο είναι αρκετό για να προβλέπει τις τιμές εξόδου μόνο ενός δείγματος εκπαίδευσης, όμως σε περίπτωση που κληθεί να προβλέψει τιμές εξόδου ενός άγνωστου για αυτό σετ τιμών εισόδου, είναι πολύ πιθανό πως θα έχει φτωχή απόδοση. Το γεγονός αυτό είναι συχνό στη μηχανική μάθηση και ονομάζεται υπερ-προσαρμογή (overfitting). Για το λόγο αυτό, κατά τη φάση της εκπαίδευσης, πρέπει να δοθεί μια πληθώρα από δείγματα εισόδου, ώστε το δίκτυο να αφομοιώσει την ποικιλία της πληροφορίας και να μπορεί να ανταποκρίνεται καλύτερα σε περιπτώσεις που οι τιμές εισόδου είναι διαφορετικές από αυτές που έχει 'μάθει'. Εκτός από δείγματα εκπαίδευσης, λοιπόν, πρέπει να υπάρχουν και κάποια δείγματα άγνωστα για το δίκτυο,



ώστε να δοκιμάζεται η ακρίβειά του. Τα δείγματα αυτά είναι γνωστά και ως δείγματα πιστοποίησης (validation patterns). Το σύνολο των δειγμάτων εκπαίδευσης και πιστοποίησης, ονομάζεται βάση δεδομένων (Database) του δικτύου, και η συγκρότησή της είναι σημείο άξιο προσοχής.

### 5.1.8 Κανονικοποίηση των Δεδομένων (Data Normalization)

Συχνά στη μηχανική μάθηση, οι τιμές εισόδου στα δείγματα εκπαίδευσης προέρχονται από μεγέθη, τα οποία συνοδεύονται συχνά από διαφορετικές μονάδες το καθένα. Συνεπώς, οι τιμές εισόδου συνήθως δεν βρίσκονται υπό την ίδια κλίμακα και το δίκτυο δυσκολεύεται να προσαρμοστεί σε μη-σταθμισμένα δεδομένα. Για να λυθεί το πρόβλημα αυτό, εφαρμόζεται η τεχνική της κανονικοποίησης των δεδομένων, στην οποία όλες οι τιμές των δειγμάτων (τιμές εισόδου και εξόδου), κανονικοποιούνται σε ενιαία κλίμακα. Υπάρχουν διάφορες μέθοδοι κανονικοποίησης (λ.χ. κανονικοποίηση Z (Z Normalization), κανονικοποίηση μοναδιαίου διανύσματος (unit vector normalization), κ.α.). Στο πλαίσιο της εργασίας αυτής, εφαρμόζεται η κανονικοποίηση μεγίστου-ελαχίστου (Min-Max Normalization), η οποία παρουσιάζεται παρακάτω.

Εάν με  $X$  συμβολίσουμε το μητρώο που περιέχει τα δεδομένα των δειγμάτων εκπαίδευσης, τότε αυτό θα έχει διαστάσεις  $N \times D$ , όπου  $N$  είναι ο αριθμός των καταχωρήσεων (entries) και  $D$  είναι ο αριθμός των χαρακτηριστικών (features). Έτσι, η κανονικοποίηση γίνεται ως εξής:

$$\hat{X}[:, i] = \frac{X[:, i] - \min(X[:, i])}{\max(X[:, i]) - \min(X[:, i])}$$

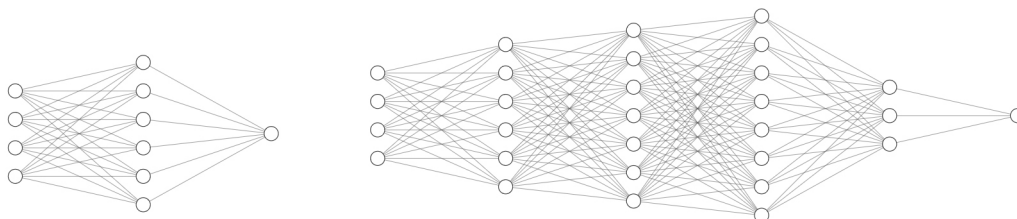
όπου  $\hat{X}$  το νέο, αδιαστατοποιημένο μητρώο.

### 5.1.9 Υπερπαράμετροι

Πριν ξεκινήσει η φάση της εκπαίδευσης, υπάρχουν κάποιες παράμετροι που πρέπει να ρυθμιστούν, οι οποίες επηρεάζουν άμεσα την πρόοδο της εκπαίδευσης. Το σύνολο των παραμέτρων αυτών περιλαμβάνει την αρχιτεκτονική του δικτύου (λ.χ. τον αριθμό των νευρώνων και των επιπέδων), το ρυθμό μάθησης, τον αριθμό εποχών και τον αριθμό των ομάδων. Οι παράμετροι αυτές ονομάζονται υπερπαράμετροι και δεν πρέπει να συγχέονται με τις παραμέτρους που ρυθμίζονται με την εκπαίδευση (βάρη, τιμές bias). Η ρύθμιση των υπερπαραμέτρων είναι μια δύσκολη διαδικασία, που συχνά αναφέρεται και ως tuning, ενώ πολλές φορές γίνεται εμπειρικά.

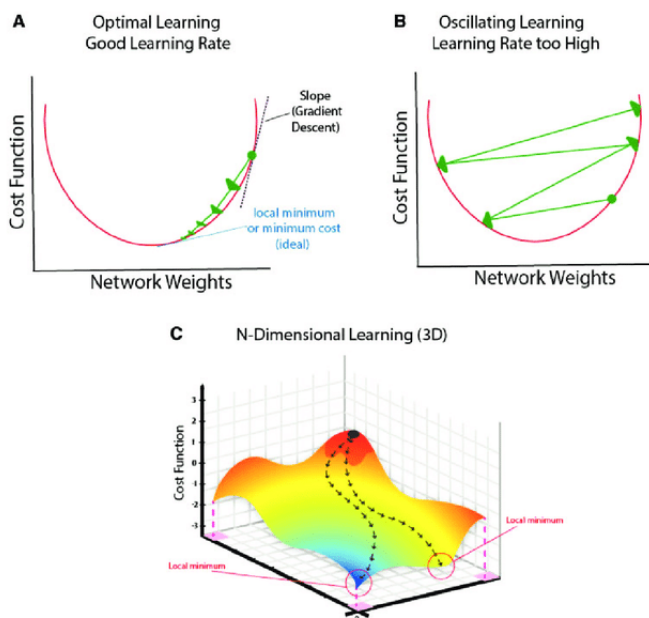
**Αρχιτεκτονική Δικτύου** Με τον όρο αρχιτεκτονική δικτύου εννοείται ο αριθμός των επιπέδων (layers), ο τρόπος που συνδέονται μεταξύ τους και ο αριθμός των

νευρώνων ανά επίπεδο. Η αρχιτεκτονική του δικτύου καθορίζει κατά κύριο λόγο την ικανότητά του να κάνει προβλέψεις σε σύνθετα προβλήματα. Ωστόσο, πιο 'βαριές' αρχιτεκτονικές (πολλά επίπεδα, πολυπληθή σε νευρώνες επίπεδα) τείνουν να χρειάζονται μεγαλύτερο όγκο δεδομένων για τη φάση της εκπαίδευσης (μεγαλύτερες βάσεις δεδομένων).



**Σχήμα 5.4:** Αριστερά: μια απλή αρχιτεκτονική με ένα κρυμμένο επίπεδο, Δεξιά: μια πιο σύνθετη αρχιτεκτονική

**Ρυθμός Μάθησης** Ο ρυθμός μάθησης (learning rate) είναι μια υπερπαράμετρος που ρυθμίζεται και από τον αλγόριθμο εκπαίδευσης, όπως αναφέρθηκε παραπάνω. Ωστόσο, ο αλγόριθμος θα ακολουθήσει την τιμή που ορίζει ο χρήστης. Η επιλογή μεγάλης τιμής στο ρυθμό μάθησης μπορεί να οδηγήσει στην προσπέραση του ολικού ελαχίστου (overshooting). Η επιλογή μικρής τιμής μπορεί να κάνει τη σύγκλιση πολύ χρονοβόρα, ενώ υπάρχει κίνδυνος το δίκτυο να παγιδευτεί σε τοπικό ακρότατο.



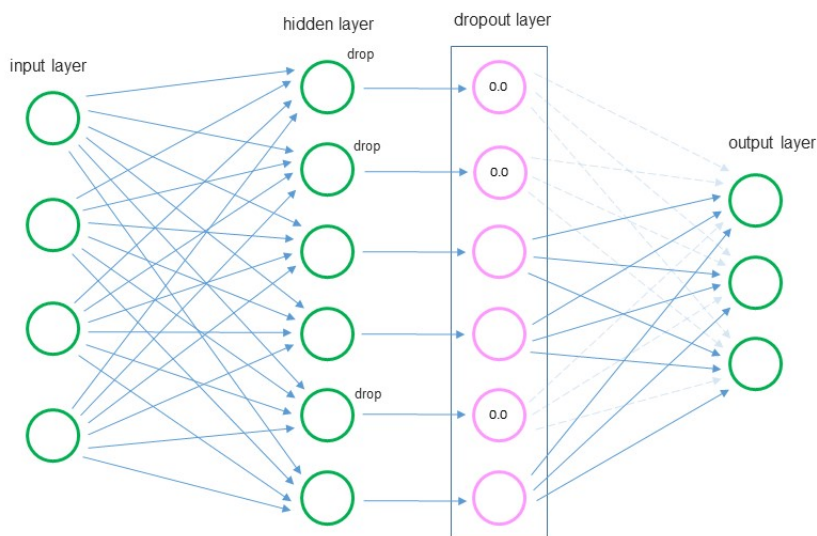
**Σχήμα 5.5:** Αναπαράσταση της επίδρασης του ρυθμού μάθησης

**Αριθμός Ομάδων (Batches)** Συχνά, το σύνολο των δειγμάτων εκπαίδευσης χωρίζεται σε ομάδες (batches). Με τον τρόπο αυτό, το δίκτυο ανανεώνει τις παραμέτρους

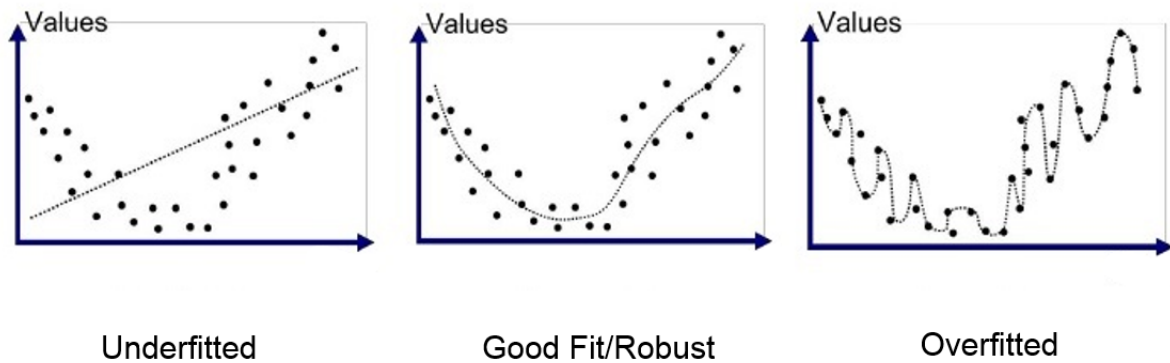
του με την προσπέλαση των δειγμάτων εκπαίδευσης περισσότερες φορές από μία και συγκεκριμένα ισούται με τον αριθμό των ομάδων. Για παράδειγμα, εάν τα δείγματα εκπαίδευσης είναι 1000, και χωριστούν σε 10 ομάδες, τότε το δίκτυο θα ανανεώσει 10 φορές τις παραμέτρους του με μία προσπέλαση των δειγμάτων εκπαίδευσης, δηλαδή το δίκτυο θα εκπαιδεύεται ανά 100 δείγματα. Αντιθέτως, εάν δεν χωριστούν σε ομάδες (δηλαδή ο αριθμός των ομάδων είναι ίσος με τη μονάδα), τότε θα ανανεώσει τις παραμέτρους του μία μόνο φορά, δηλαδή θα εκπαιδεύεται ανά 1000 δείγματα σε κάθε επανάληψη.

**Αριθμός Εποχών (Epochs)** Μια εποχή ολοκληρώνεται όταν όλα τα δείγματα εκπαίδευσης προσπελαστούν μία φορά. Συνεπώς, ο αριθμός των εποχών καθορίζει πόσες φορές θα επαναληφθεί αυτή η διαδικασία. Εν γένει, με το πέρασμα των εποχών ακολουθεί και η σύγκλιση του δικτύου σε ακρότατο. Ωστόσο, μετά από έναν αριθμό εποχών, το δίκτυο κατά κανόνα αρχίζει να προσαρμόζεται υπερβολικά στα δείγματα εκπαίδευσης, προκαλώντας έτσι υπερ-προσαρμογή.

**Dropout** Μια τεχνική που εφαρμόζεται για τη μείωση του φαινομένου της υπερ-προσαρμογής (overfitting - σχήμα 5.7), είναι η προσθήκη επιπέδων Dropout. Τα επίπεδα Dropout προστίθενται μετά από ένα επίπεδο του δικτύου και η λειτουργία τους είναι αντίστοιχη με ένα τυχαίο φίλτρο: κάθε φορά που οι νευρώνες ενός επιπέδου ενεργοποιούνται και εξάγουν τιμές, τυχαία ένα ποσοστό των νευρώνων δεν ενεργοποιείται και έτσι δεν παράγει αριθμητική τιμή. Το ποσοστό αυτό είναι επίσης μια υπερπαραμέτρος, όπως επίσης και ο αριθμός των επιπέδων Dropout.



Σχήμα 5.6: Αναπαράσταση της επίδρασης ενός επιπέδου Dropout



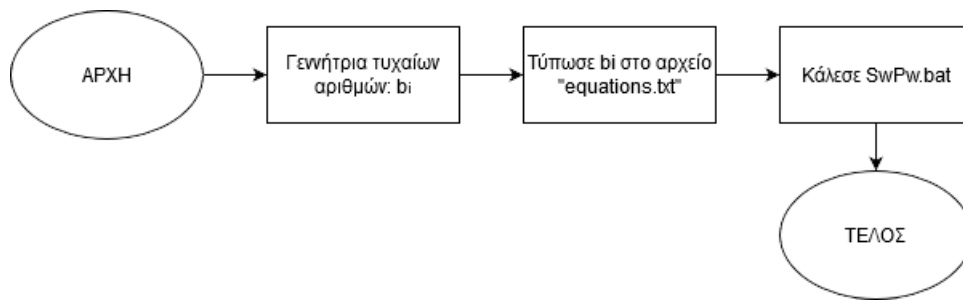
Σχήμα 5.7: Αναπαράσταση της υπερ-προσαρμογής (*overfitting*)

## 5.2 Υλοποίηση της εκπαίδευσης

Όπως παρουσιάστηκε στην ενότητα της βελτιστοποίησης του σχήματος του αγωγού, η διαδικασία της βελτιστοποίησης είναι αρκετά κοστοβόρα, καθώς η αξιολόγηση της εκάστοτε υποψήφιας λύσης με χρήση λογισμικών CFD έχει υψηλό υπολογιστικό κόστος, ακόμα και με χρήση μεταμοντέλων. Ο στόχος της πρόβλεψης της ροής με χρήση βαθιών νευρωνικών δικτύων είναι η αντικατάσταση της υψηλού υπολογιστικού κόστους επίλυσης της ροής με χρήση λογισμικού CFD, με ένα βαθύ νευρωνικό δίκτυο χαμηλού υπολογιστικού κόστους, το οποίο θα μπορεί να προβλέπει με μικρά σφάλματα τη ροή. Ουσιαστικά, σκοπός των δικτύων είναι η πρόβλεψη των ολικών πιέσεων, ώστε να μη χρειάζεται η επίλυση της ροής για την εξαγωγή των απωλειών ολικής πίεσης κατά τον βρόχο της βελτιστοποίησης. Για την εκπαίδευση ενός τέτοιου δικτύου, απαιτείται η συγκρότηση μιας βάσης δεδομένων, τα οποία έχουν συλλεγεί από ροές που έχουν επιλυθεί με χρήση του λογισμικού CFD PUMA.

### 5.2.1 Σύνθεση Βάσης Δεδομένων

Για τη συγκρότηση της βάσης δεδομένων, απαιτούνται αποτελέσματα από ροές που έχουν επιλυθεί με λογισμικό CFD. Το λογισμικό που χρησιμοποιείται και σε αυτήν την ενότητα είναι το PUMA. Το βαθύ νευρωνικό δίκτυο πρέπει να εκπαιδευτεί σε ροές από διάφορες γεωμετρίες του αγωγού, έτσι ώστε να είναι ικανό να μπορεί να προβλέπει τη ροή σε οποιαδήποτε μορφή της γεωμετρίας του αγωγού. Η σύνθεση της βάσης δεδομένων μπορεί να γίνει με διάφορους τρόπους, οργανωμένους ή μη. Στη συγκεκριμένη περίπτωση επιλέχθηκε η συγκρότηση βάσης δεδομένων που περιέχει τυχαίες γεωμετρίες του αγωγού. Συνεπώς, χρησιμοποιώντας μια γεννήτρια τυχαίων αριθμών, παράγονται τυχαίες τιμές για τις πέντε μεταβλητές σχεδιασμού του αγωγού (σχήμα 5.8). Φυσικά οι τυχαίες τιμές βρίσκονται στο καθορισμένο εύρος που αναφέρθηκε στην ενότητα της βελτιστοποίησης του αγωγού.



**Σχήμα 5.8:** Διάγραμμα ροής της παραγωγής τυχαίας γεωμετρίας

Όπως έχει αναλυθεί σε προηγούμενη ενότητα, η γεωμετρία παράγεται αυτόματα και ύστερα επιλύεται η ροή για την εκάστοτε γεωμετρία χρησιμοποιώντας το λογισμικό PUMA. Η βάση δεδομένων που συγκροτήθηκε, αποθηκεύτηκε στο υπολογιστικό σύστημα της ΜΠΥΡΒ/ΕΜΠ και αποτελείται από τους παρακάτω δύο τύπους αρχείων:

- Αρχείο 'pointdata.dat'
- Αρχείο 'vars.dat'

**Αρχείο 'pointdata.dat'** Το λογισμικό PUMA παράγει αρχείο αποτελεσμάτων σε μορφή \*.vtu, το οποίο έχει την εξής δομή:

```

...
(x, y, z)
...
Node Connectivity
...
Pst/ρ
...
u
v
w
...
μt
μ̃t

```

**Σχήμα 5.9:** Δομή του αρχείου \*.vtu. Στην ενότητα κάθε μεγέθους περιέχονται οι αριθμητικές τιμές για όλους τους κόμβους (728875 κόμβοι)

Χρησιμοποιώντας τις συνιστώσες της ταχύτητας και τις τιμές της στατικής πίεσης

διαμεμένης με τη σταθερή πυκνότητα, υπολογίζονται μέσω της εξίσωσης Bernoulli οι τιμές των ολικών πιέσεων για όλους τους κόμβους:

$$P_t = \rho \frac{P_{st}}{\rho} + \frac{1}{2} \rho (u^2 + v^2 + w^2)$$

Μαζί με τις συντεταγμένες των κόμβων από το αρχείο \*.vtu, συγκροτείται το αρχείο 'pointdata.dat', το οποίο σε κάθε γραμμή του αναγράφει τις συντεταγμένες του κόμβου και την αντίστοιχη ολική πίεση.

```

pointdata.dat
x1 y1 z1 pt1
x2 y2 z2 pt2


...

xN yN zN ptN

```

**Σχήμα 5.10:** Δομή του αρχείου 'pointdata.dat', όπου  $N$  ο αριθμός των κόμβων του πλέγματος, δηλαδή το αρχείο περιέχει  $N=728875$  γραμμές

**Αρχείο 'vars.dat'** Το αρχείο 'vars.dat' περιλαμβάνει τις τιμές των πέντε μεταβλητών σχεδιασμού, που αναγράφονται στο αρχείο 'equations.txt' της εκάστοτε γεωμετρίας (σχήμα 5.11)

<pre> equations.txt "a1" = 90deg "a2" = 90deg "r1" = 0.1m "r2" = 0.1m "l1" = 0.1m "D1@3DSketch4"= "a1" "D2@3DSketch4"= "a2" "D4@3DSketch4"= "r1" "D5@3DSketch4"= "r2" "D3@3DSketch4"= "l1" </pre>		<pre> vars.dat 90 90 0.1 0.1 0.1 </pre>
---	---	---

**Σχήμα 5.11:** Δομή των αρχείων 'equations.txt' και 'vars.dat'

## 5.2.2 Προετοιμασία των Δεδομένων

Ένα δίκτυο, όπως αναφέρθηκε, δεν μπορεί να εκπαιδευτεί σε μη-σταθμισμένα δεδομένα. Έτσι, οι τιμές των αρχείων 'pointdata.dat' και 'vars.dat' αδιαστατοποιούνται με την

κανονικοποίηση μεγίστου-ελαχίστου,

$$\hat{X}[:, i] = \frac{X[:, i] - \min(X[:, i])}{\max(X[:, i]) - \min(X[:, i])}$$

Για τις μεταβλητές σχεδιασμού, η ως μέγιστες και ελάχιστες τιμές χρησιμοποιούνται τα ανώ και κάτω όρια από το εύρος τους  $b_{i,upper-lim}$ ,  $b_{i,lower-lim}$ . Για τις συντεταγμένες (x, y, z) και τις ολικές πιέσεις, βρίσκονται οι μέγιστες και οι ελάχιστες τιμές για το κάθε μέγεθος από όλη τη βάση δεδομένων (db - Data Base):  $x_{max,db}$ ,  $x_{min,db}$ , κλπ. Στο εξής, όποτε αναφέρονται τα μεγέθη αυτά ως δεδομένα εισόδου και εξόδου από το δίκτυο, θα εννοούνται οι αδιαστατοποιημένες μορφές τους. Τέλος, το δίκτυο εκπαιδεύεται χρησιμοποιώντας τα μητρώα των δειγμάτων εκπαίδευσης 'dbInput' και 'dbOutput', ενώ η ακρίβειά του επαληθεύεται με τα δείγματα πιστοποίησης 'valInput' και 'valOutput'. Το περιεχόμενο των μητρώων αυτών αναλύεται στις επόμενες παραγράφους.

### 5.3 Μεθοδολογία χρήσης DNN για την Πρόβλεψη της Ροής στον Αγωγό

Τα βαθιά νευρωνικά δίκτυα που εκπαιδεύτηκαν στο πλαίσιο της εργασίας αυτής, προγραμματίστηκαν σε γλώσσα python, χρησιμοποιώντας τη βιβλιοθήκη tensorflow, ενώ χρησιμοποιήθηκε επεξεργαστής GPU Nvidia Tesla K20m της ΜΠΥΡΒ/ΕΜΠ. Για την ακρίβεια των προβλέψεων του εκάστοτε δικτύου, χρησιμοποιείται η έννοια του σφάλματος MSE - Mean Squared Error:

$$MSE = \frac{1}{N} \sum_1^N (y_i - \hat{y}_i)^2$$

όπου  $y_i$  η πραγματική τιμή εξόδου, δηλαδή η επιθυμητή τιμή της πρόβλεψης,  $\hat{y}_i$  η πρόβλεψη του δικτύου και  $N$  το σύνολο των προβλέψεων του δικτύου.

Η αντικατάσταση του λογισμικού CFD στον βρόχο της βελτιστοποίησης μπορεί να πραγματοποιηθεί ακολουθώντας τις παρακάτω προσεγγίσεις. Υπενθυμίζεται πως η συνάρτηση-στόχος στη βελτιστοποίηση της γεωμετρίας του αγωγού είναι οι απώλειες ολικής πίεσης  $\Delta P_t$ . Να σημειωθεί πως το λογισμικό CFD PUMA σταθμίζει τις ολικές πιέσεις με την παροχή. Για την εξαγωγή των ολικών πιέσεων εισόδου και εξόδου χρησιμοποιώντας τα δίκτυα, η στάθμιση με την παροχή δεν μπορεί να γίνει.

- Α' Προσέγγιση: Πρόβλεψη της ολικής πίεσης (μία τιμή) σε κάθε κόμβο του πλέγματος
- Β' Προσέγγιση: Πρόβλεψη της ολικής πίεσης (μία τιμή) σε κάθε κόμβο της διατομής εξόδου

- Γ' Προσέγγιση: Πρόβλεψη όλων των ολικών πιέσεων (πλήθος τιμών) στη διατομή εξόδου

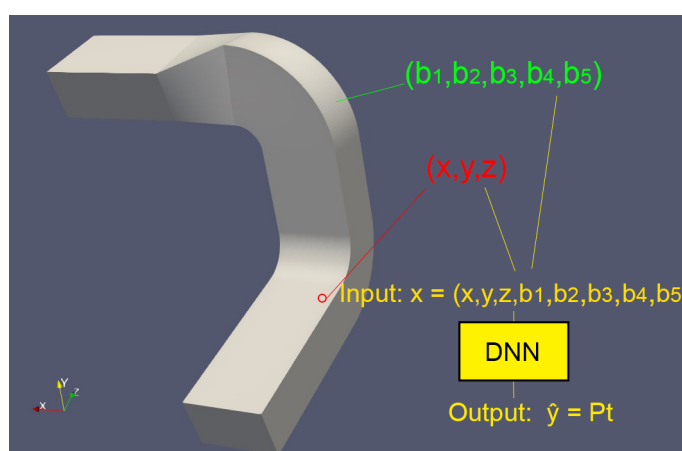
Για το σκοπό της εργασίας, πραγματοποιήθηκαν οι προσεγγίσεις A, B και Γ, οι οποίες αναλύονται παρακάτω. Για τη σύγκριση των προσεγγίσεων αυτών, εκπαιδεύτηκαν ορισμένα δίκτυα στην ίδια βάση δεδομένων των 40 δειγμάτων εκπαίδευσης.

### 5.3.1 Α' Προσέγγιση

Στην πρώτη προσέγγιση το δίκτυο χρησιμοποιείται για να προβλέπει την ολική πίεση του κάθε κόμβου ξεχωριστά, προσπελαύνοντας όλους τους κόμβους του αγωγού. Ως είσοδος στο δίκτυο δίνονται οι συντεταγμένες του κόμβου  $(x, y, z)$  και οι μεταβλητές σχεδιασμού  $b_1, b_2, b_3, b_4, b_5$  που χαρακτηρίζουν τη γεωμετρία στην οποία ανήκει ο κόμβος. Έτσι, συγκροτείται το διάνυσμα εισόδου:

$$\vec{x} = [x, y, z, b_1, b_2, b_3, b_4, b_5]$$

Η έξοδος του δικτύου, είναι η ολική πίεση που αντιστοιχεί στον κόμβο που δώθηκε ως είσοδος (σχήμα 5.12). Το πλέγμα του αγωγού αποτελείται από 728875 κόμβους, συνεπώς το ίδιο δίκτυο πραγματοποιεί 728875 προβλέψεις ανά γεωμετρία. Τα μητρώα εισόδου dbInput και εξόδου dbOutput της εκπαίδευσης φαίνονται στο σχήμα 5.13. Το μητρώο εισόδου dbInput περιέχει σε κάθε γραμμή το διάνυσμα εισόδου που δίνεται στο δίκτυο, ενώ το αρχείο εξόδου dbOutput περιέχει σε κάθε γραμμή την πραγματική ολική πίεση του αντίστοιχου κόμβου, με την οποία υπολογίζεται το σφάλμα μεταξύ αυτής και της πρόβλεψης. Έτσι, το μητρώο dbInput έχει διαστάσεις  $[NbTP \cdot 728875, 8]$ , όπου  $NbTP$  ο αριθμός των δειγμάτων εκπαίδευσης (Number of Training Patterns), ενώ το μητρώο dbOutput  $[NbTP \cdot 728875, 1]$ . Αντίστοιχα, για ένα δείγμα πιστοποίησης (validation case), τα μητρώα εισόδου valInput και εξόδου valOutput φαίνονται στο σχήμα 5.13, ενώ έχουν διαστάσεις  $[728875, 8]$  και  $[728875, 1]$  αντιστοίχως.



Σχήμα 5.12: Πρόβλεψη της ολικής πίεσης σε έναν τυχαίο κόμβο μιας γεωμετρίας



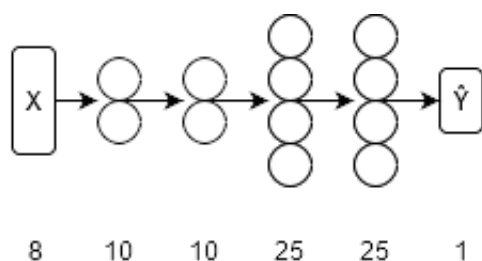
dblInput								dbOutput
x1,1	y1,1	z1,1	b1,1	b1,2	b1,3	b1,4	b1,5	Pt1,1
x1,2	y1,2	z1,2	b1,1	b1,2	b1,3	b1,4	b1,5	Pt1,2
		...						...
x1,N	y1,N	z1,N	b1,1	b1,2	b1,3	b1,4	b1,5	Pt1,N
		...						...
		...						PtNbTP,N
		...						
xNbTP,N	yNbTP,N	zNbTP,N	bNbTP,1	bNbTP,2	bNbTP,3	bNbTP,4	bNbTP,5	

**Σχήμα 5.13:** Τα μητρώα εισόδου και εξόδου που διαβάζει το δίκτυο κατά τη φάση της εκπαίδευσης ( $N=728875$  κόμβοι)

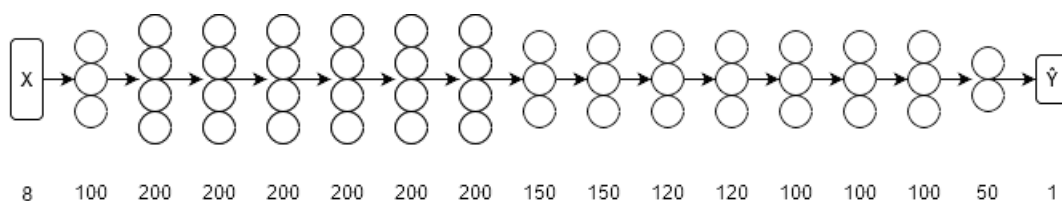
valInput								valOutput
x1,1	y1,1	z1,1	b1,1	b1,2	b1,3	b1,4	b1,5	Pt1,1
x1,2	y1,2	z1,2	b1,1	b1,2	b1,3	b1,4	b1,5	Pt1,2
		...						...
x1,N	y1,N	z1,N	b1,1	b1,2	b1,3	b1,4	b1,5	Pt1,N

**Σχήμα 5.14:** Τα μητρώα εισόδου και εξόδου που διαβάζει το δίκτυο κατά τη φάση της εκπαίδευσης ( $N=728875$  κόμβοι)

Επειδή τα μητρώα είναι αρκετά μεγάλα σε μέγεθος, τα δεδομένα χωρίζονται σε ομάδες (batches). Ο αριθμός των ομάδων που επιλέχθηκε, εξαρτώμενος από το υπολογιστικό κόστος της αρχιτεκτονικής του δικτύου, κυμάνθηκε από 10 έως 80. Ο χρόνος για την εκπαίδευση ενός DNN με την προσέγγιση αυτή είναι αρκετά μεγάλος, κυμαινόμενος από  $1h/200epochs$  για απλές αρχιτεκτονικές έως  $1h/10epochs$  για πιο σύνθετες. Έτσι, για 2000 εποχές (που περίπου απαιτούνται για τη σύγκλιση σε αυτήν την προσέγγιση) απαιτείται χρόνος από 10 ώρες για απλές αρχιτεκτονικές έως 200. Ενδεικτικά για την πρώτη προσέγγιση, παρουσιάζονται στην παράγραφο της σύγκρισης των προσεγγίσεων τα δίκτυα A1 και A2. Στα σχήματα 5.15 και 5.16 φαίνονται οι αρχιτεκτονικές των δύο δικτύων. Στο δίκτυο A1 δόθηκε μια απλή αρχιτεκτονική, ενώ στο δίκτυο A2 μια αρκετά πιο σύνθετη.



**Σχήμα 5.15:** Η αρχιτεκτονική του DNN A1. Οι νευρώνες του σχήματος είναι ενδεικτικοί - ο αριθμός των νευρώνων για κάθε επίπεδο φαίνεται από κάτω



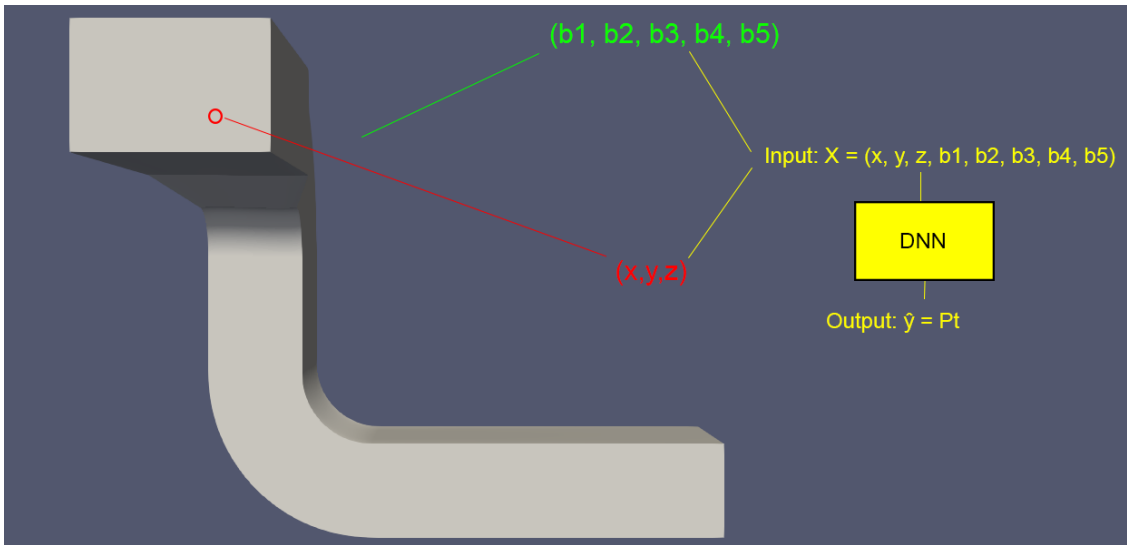
**Σχήμα 5.16:** Η αρχιτεκτονική του DNN A2

### 5.3.2 Β' Προσέγγιση

Στη δεύτερη προσέγγιση το δίκτυο χρησιμοποιείται για να προβλέπει την ολική πίεση του κάθε κόμβου ξεχωριστά, προσπελώνοντας αυτή τη φορά μόνο τους κόμβους της διατομής εξόδου, στο σύνολο των 1225 κόμβων ανά γεωμετρία ( $35 \times 35 = 1225$  κόμβοι στην έξοδο). Η λογική της προσέγγισης αυτής, είναι πως για να εξαχθούν οι απώλειες ολικής πίεσης του αγωγού απαιτούνται μόνο οι ολικές πιέσεις στην είσοδο και στην έξοδο. Η ολική πίεση στην είσοδο έχει επιβληθεί ως οριακή συνθήκη του προβλήματος. Φυσικά, προσεγγίζοντας το πρόβλημα με αυτόν τον τρόπο, χάνεται η πληροφορία των πιέσεων εντός του αγωγού. Ωστόσο, η προσέγγιση αυτή υπερτερεί από άποψη χρόνου εκπαίδευσης του δικτύου, καθώς ο αριθμός των κόμβων μειώθηκε σημαντικά, ενώ η κατανομή των πιέσεων στη διατομή εξόδου περιέχει εξίσου ενδιαφέρουσα πληροφορία. Το διάνυσμα εισόδου που συγκροτείται είναι το ίδιο:

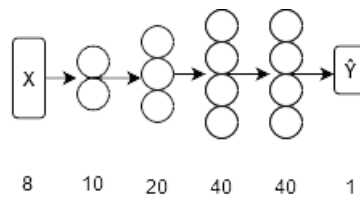
$$\vec{x} = [x, y, z, b_1, b_2, b_3, b_4, b_5]$$

όπου  $(x, y, z)$  οι συντεταγμένες του εκάστοτε κόμβου της διατομής εξόδου (σχήμα 5.17).

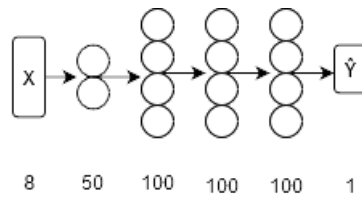


**Σχήμα 5.17:** Πρόβλεψη της ολικής πίεσης σε έναν τυχαίο κόμβο της διατομής εξόδου μιας γεωμετρίας

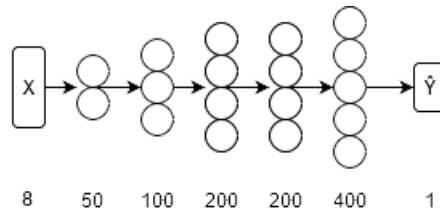
Τα μητρώα εισόδου έχουν αντίστοιχη μορφή με της προσέγγισης Α', όμως με πολύ μικρότερη διάσταση γραμμών. Το μητρώο dbInput πλέον έχει διαστάσεις  $[NbTP \cdot 1225, 8]$ , το μητρώο dbOutput  $[NbTP \cdot 1225, 1]$  και τα μητρώα valInput και valOutput  $[1225, 8]$  και  $[1225, 1]$  αντίστοιχα, για ένα δείγμα πιστοποίησης. Επειδή τα μητρώα έχουν σαφώς μικρότερο μέγεθος από τα αντίστοιχα της προσέγγισης Α', τα δεδομένα δε χωρίζονται σε ομάδες. Ο χρόνος εκπαίδευσης των δικτύων χρησιμοποιώντας αυτήν την τεχνική μειώθηκε σημαντικά στο  $1min/200epochs$ , εξαρτώμενος όμως πάλι από τη συνθετότητα της αρχιτεκτονικής. Με μικρότερο υπολογιστικό κόστος εκπαίδευσης σε σχέση με την πρώτη προσέγγιση, η προσέγγιση Β' δίνει τη δυνατότητα για επιλογή περισσότερων εποχών εκπαίδευσης. Για παράδειγμα, για 2000 εποχές απαιτούνται μόλις 10 λεπτά (σε σύγκριση με τις 10 ώρες της γρηγορότερης περίπτωσης της προσέγγισης Α'). Έτσι, εκπαιδεύοντας ένα δίκτυο για 50000 εποχές, οδηγώντας το σε πλήρη σύγκλιση, το υπολογιστικό κόστος σε χρόνο ανέρχεται σε περίπου 4 ώρες. Αξίζει να σημειωθεί πως ο αριθμός των εποχών που απαιτούνται για τη σύγκλιση των δικτύων σε κάθε προσέγγιση είναι διαφορετικός. Η προσέγγιση Β' οδηγεί σε πιο ακριβή αποτελέσματα, καθώς το δίκτυο εστιάζει καλύτερα στις λιγότερες τιμές. Για τη δεύτερη προσέγγιση, επιλέχθηκαν τα δίκτυα B1 (σχήμα 5.18), B2 (5.19), B3 (5.20) και B4 (5.22).



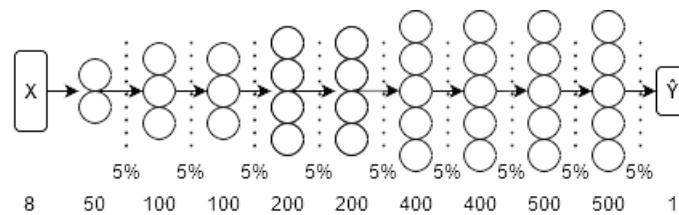
**Σχήμα 5.18:** Η αρχιτεκτονική του DNN B1



Σχήμα 5.19: Η αρχιτεκτονική του DNN B2



Σχήμα 5.20: Η αρχιτεκτονική του DNN B3



Σχήμα 5.21: Η αρχιτεκτονική του DNN B4. Η αρχιτεκτονική αυτή περιλαμβάνει επίπεδα Dropout με ποσοστό 5% μετά από κάθε κρυμμένο επίπεδο

### 5.3.3 Γ' Προσέγγιση

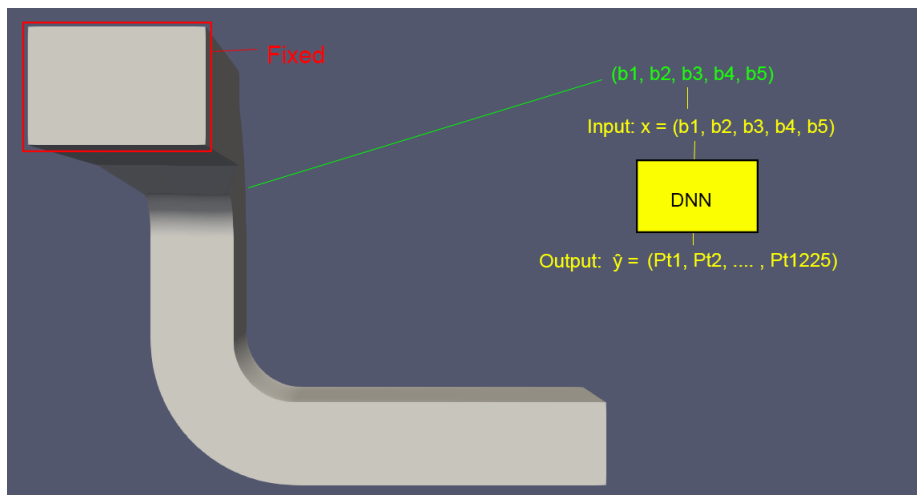
Στην τρίτη προσέγγιση το δίκτυο χρησιμοποιείται για την πρόβλεψη όλων των ολικών πιέσεων της διατομής εξόδου. Σε αντίθεση με τις προηγούμενες δύο προσεγγίσεις, πλέον η έξοδος του δικτύου δεν είναι μία τιμή, αλλά 1225 (όσοι είναι και οι κόμβοι της διατομής εξόδου). Από τη στιγμή που το δίκτυο δεν προσπελαύνει ανά έναν τους κόμβους, δεν δίνονται ως είσοδος συντεταγμένες. Άλλωστε, από το σχεδιασμό του αγωγού, οι διατομές εισόδου και εξόδου έχουν οριστεί ως σταθερές. Το γεγονός αυτό, σε συνδυασμό με το ότι οι γεωμετρίες που συνθέτουν τη βάση δεδομένων παράχθηκαν αυτόματα από τα ίδια scripts (SwPw.bat), έχει ως αποτέλεσμα οι κόμβοι της διατομής εξόδου να βρίσκονται στις ίδιες θέσεις στα αρχεία πλεγμάτων και φυσικά να έχουν τις ίδιες συντεταγμένες. Έτσι, οι μόνες μεταβλητές που επηρεάζουν το πεδίο πιέσεων είναι οι μεταβλητές σχεδιασμού της εκάστοτε γεωμετρίας. Συνεπώς, το δίκτυο δέχεται ως είσοδο τις μεταβλητές σχεδιασμού:

$$\vec{x} = [b_1, b_2, b_3, b_4, b_5]$$

και εξάγει ως έξοδο ένα διάνυσμα με 1225 πιέσεις (σχήμα 5.23):

$$\vec{y} = [p_{t1}, p_{t2}, \dots, p_{t1225}]$$

Τα μητρώα εισόδου και εξόδου στην προσέγγιση αυτή είναι διαφορετικά. Το μητρώο εισόδου dbInput περιέχει σε κάθε σειρά τις μεταβλητές σχεδιασμού της αντίστοιχης γεωμετρίας και έχει διαστάσεις  $[NbTP, 5]$ . Το μητρώο εξόδου dbOutput περιέχει σε κάθε σειρά τις 1225 πιέσεις και έχει διαστάσεις  $[NbTP, 1225]$  (σχήμα 5.23). Αντίστοιχα, τα μητρώα valInput και valOutput για ένα δείγμα πιστοποίησης έχουν διαστάσεις  $[1, 5]$  και  $[1, 1225]$  αντίστοιχα.



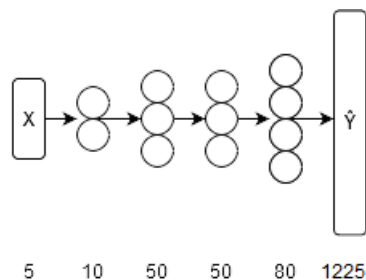
**Σχήμα 5.22:** Πρόβλεψη όλων των ολικών πιέσεων της διατομής εξόδου μιας γεωμετρίας

dbInput					dbOutput			
b1,1	b1,2	b1,3	b1,4	b1,5	Pt1,1	Pt1,2	. . .	Pt1,1225
b2,1	b2,2	b2,3	b2,4	b2,5	Pt2,1	Pt2,2	. . .	Pt2,1225
	⋮	⋮				⋮		
bNbTP,1	bNbTP,2	bNbTP,3	bNbTP,4	bNbTP,5	PtNbTP,1	PtNbTP,2	. . .	PtNbTP,1225

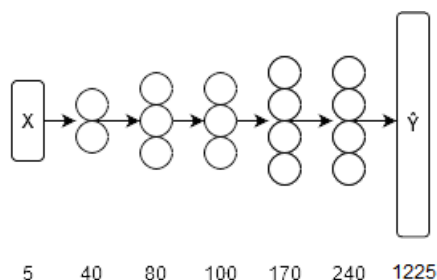
**Σχήμα 5.23:** Τα μητρώα εισόδου και εξόδου για τη φάση της εκπαίδευσης

Όπως φαίνεται, τα μητρώα συνολικά έχουν μικρότερο μέγεθος από τα αντίστοιχα των προηγούμενων προσεγγίσεων. Έτσι, τα δεδομένα δεν χρειάστηκε να χωριστούν σε ομάδες, ενώ ο χρόνος εκπαίδευσης μειώθηκε κατακόρυφα στο  $1min/6000epochs$ . Η προσέγγιση Γ', λοιπόν, παρέχει τη δυνατότητα για πολύ γρήγορη εκπαίδευση, καθώς για πραγματικό χρόνο 10 λεπτά, με την προσέγγιση Γ' έχουν ολοκληρωθεί 60000 εποχές, σε σύγκριση με τις 2000 εποχές της προσέγγισης Β' στον ίδιο χρόνο. Για την

προσέγγιση αυτή, απαιτούνται περίπου 80000 εποχές για τη σύγκλιση. Τα δίκτυα που αντιπροσωπεύουν την προσέγγιση αυτή στην παρακάτω σύγκριση είναι τα δίκτυα C1 (σχήμα 5.24) και C2 (σχήμα 5.25).



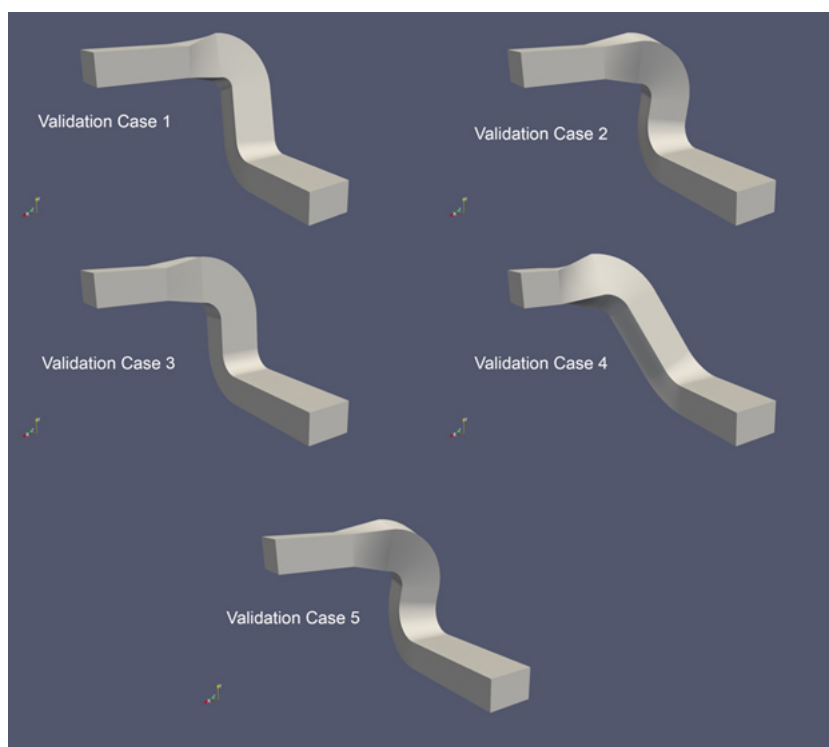
Σχήμα 5.24: Η αρχιτεκτονική του DNN C1



Σχήμα 5.25: Η αρχιτεκτονική του DNN C2

### 5.3.4 Σύγκριση των Προσεγγίσεων Α', Β' και Γ'

Για να συγκριθεί η αποτελεσματικότητα των προσεγγίσεων, τα δίκτυα της κάθε προσέγγισης δοκιμάζονται σε πέντε ίδια δείγματα πιστοποίησης, δηλαδή τυχαίες γεωμετρίες που τα δίκτυα δεν έχουν "δει" κατά τη φάση της εκπαίδευσης. Οι πέντε αυτές γεωμετρίες φαίνονται στο σχήμα 5.26. Για την κάθε περίπτωση υπολογίζεται το σφάλμα MSE. Τα αποτελέσματα φαίνονται στο σχήμα 5.1



Σχήμα 5.26: Οι πέντε τυχαίες γεωμετρίες - Δείγματα πιστοποίησης

MSE	A1	A2	B1	B2	B3	B4	C1	C2
Case 1	0.1926	0.1637	0.0599	0.0189	0.0203	0.0456	0.0113	0.0071
Case 2	0.1937	0.1789	0.0584	0.0296	0.0312	0.0690	0.0159	0.0267
Case 3	0.2014	0.1780	0.0595	0.0230	0.0174	0.0495	0.0140	0.0098
Case 4	0.2148	0.1938	0.0872	0.0344	0.0359	0.0862	0.0300	0.0386
Case 5	0.1974	0.1865	0.0683	0.0392	0.0393	0.0847	0.0216	0.0330

Πίνακας 5.1: Σύγκριση του σφάλματος MSE των δικτύων κάθε προσέγγισης για τις πέντε validation cases

Όπως φαίνεται και στον παραπάνω πίνακα, η προσέγγιση με τα μεγαλύτερα σφάλματα ήταν η πρώτη, ενώ τα μικρότερα σφάλματα τα εμφάνισε η τρίτη προσέγγιση. Βέβαια, η σημασία του σφάλματος MSE δεν είναι εύκολα κατανοητή. Έτσι, παρακάτω αναλύεται η σημασία των σφαλμάτων αυτών με παραδείγματα.

Λαμβάνοντας τις προβλέψεις του δικτύου A1 για όλους τους κόμβους της πρώτης περίπτωσης και διαστατοποιώντας αυτές σε ολικές πιέσεις, μπορεί να γίνει μια σύγκριση μεταξύ των πραγματικών τιμών και των εκτιμήσεων του δικτύου. Για παράδειγμα, στην πρώτη στήλη του παρακάτω πίνακα (σχήμα 5.2), φαίνονται οι πραγματικές ολικές πιέσεις σε πέντε τυχαίους κόμβους της πρώτης γεωμετρίας. Στη δεύτερη στήλη, φαίνονται οι προβλέψεις του δικτύου και στην τρίτη στήλη υπολογίζεται το ποσοστιαίο σφάλμα ως προς το εύρος των πιέσεων:

$$Err_i(\%) = \frac{|P_{t,i} - \hat{P}_{t,i}|}{P_{t,max} - P_{t,min}} \cdot 100\%$$

όπου:  $P_{t,max}$  η μέγιστη τιμή ολικής πίεσης και  $P_{t,min}$  η ελάχιστη. Υπολογίζοντας το ποσοστιαίο σφάλμα για όλες τις προβλέψεις, υπολογίζεται το μέσο σφάλμα των προβλέψεων για την πρώτη γεωμετρία:

$$\overline{Err}(\%) = \frac{1}{N} \sum_1^N Err_i(\%) = 19.26\%$$

Pt	Pt - Pred	Err(%)
100199.537	100199.112	8.58
100198.972	100197.947	20.69
100199.324	100198.821	10.16
100199.319	100199.117	4.08
100199.314	100199.119	3.94

**Πίνακας 5.2:** Ποσοστιαίο σφάλμα εκτίμησης του δικτύου A1 για την πρώτη περίπτωση

Αντίστοιχα, στο σχήμα 5.3 φαίνονται τα ποσοστιαία σφάλματα των προβλέψεων του δικτύου B3 για πέντε τυχαίους κόμβους της πρώτης γεωμετρίας. Οι εκτιμήσεις είναι πιο ακριβείς από του δικτύου A1, ενώ το αντίστοιχο μέσο σφάλμα των εκτιμήσεων για την πρώτη περίπτωση είναι:  $\overline{Err}(\%) = 4.36\%$

Pt	Pt - Pred	Err(%)
100199.537	100200.029	19.40
100199.297	100199.220	3.05
100199.290	100199.296	0.24
100199.271	100199.304	1.28
100199.267	100200.034	30.25

**Πίνακας 5.3:** Ποσοστιαίο σφάλμα εκτίμησης του δικτύου B3 για την πρώτη περίπτωση

Στο σχήμα 5.4 φαίνονται τα αντίστοιχα σφάλματα του δικτύου C3, το οποίο παρουσίασε μέσο σφάλμα  $\overline{Err}(\%) = 1.02\%$

Pt	Pt - Pred	Err (%)
100199.412	100199.428	0.650
100199.260	100199.280	0.813
100198.418	100198.384	1.337
100200.039	100200.041	0.075
100198.644	100198.605	1.552

**Πίνακας 5.4:** Ποσοστιαίο σφάλμα εκτίμησης του δικτύου C3 για την πρώτη γεωμετρία

Ολοκληρώνοντας την παραπάνω σύγκριση, γίνεται κατανοητό πως η τρίτη προσέγγιση κρίνεται κατάλληλη για την πρόβλεψη της ροής με χρήση DNN, καθώς τα δίκτυα που εκπαιδεύτηκαν με αυτήν την προσέγγιση παρουσιάζουν σφάλματα τάξης μεγέθους μικρότερα. Άλλος ένας λόγος που η τρίτη προσέγγιση υπερτερεί των άλλων δύο, είναι



ο χρόνος εκπαίδευσης, ο οποίος είναι σαφώς μικρότερος. Στον πίνακα 5.5 φαίνεται ο αριθμός των εποχών που εκπαιδεύτηκε το κάθε δίκτυο έως τη σύγκλιση και ο απαιτούμενος πραγματικός χρόνος. Η ανάλυση που γίνεται στις παρακάτω παραγράφους γίνεται χρησιμοποιώντας την τρίτη προσέγγιση.

Training Phase	A1	A2	B1	B2	B3	B4	C1	C2
Epochs	1000	1000	20000	20000	20000	20000	100000	100000
Total time	22h	60h	42min	50min	2h	5h	18min	30min

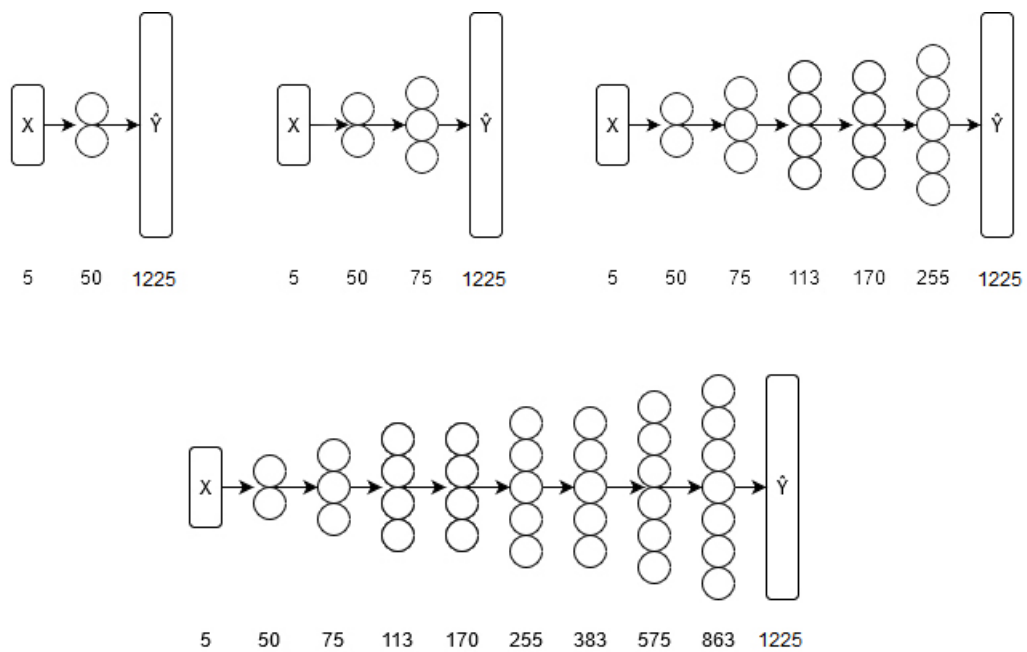
Πίνακας 5.5: Κόστος εκπαίδευσης των δικτύων της κάθε περίπτωσης

## 5.4 Ανάλυση της επίδρασης των Υπερπαραμέτρων στην εκπαίδευση DNN

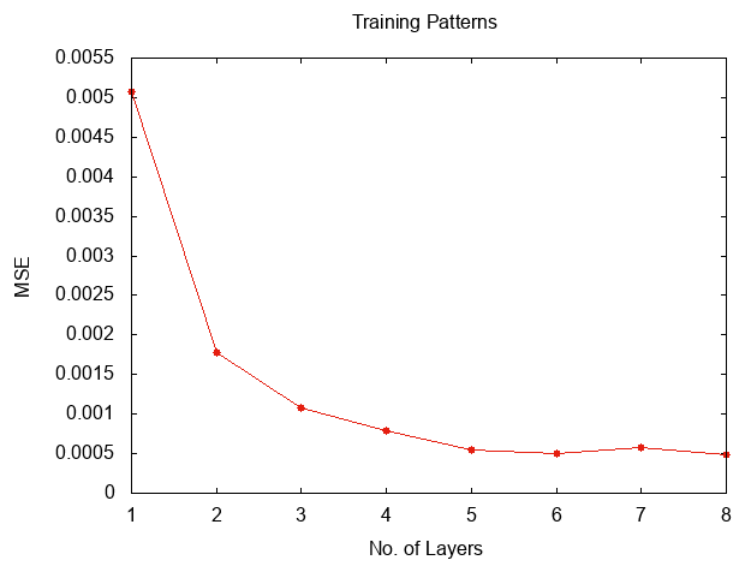
Στην ενότητα αυτή, διερευνάται η επίδραση που έχουν διάφορες υπερπαραμέτροι στην ακρίβεια των δικτύων. Η ανάλυση αυτή συνέβαλε στη ρύθμιση (tuning) των υπερπαραμέτρων για την εκπαίδευση του τελικού DNN.

### 5.4.1 Επίδραση της Αρχιτεκτονικής

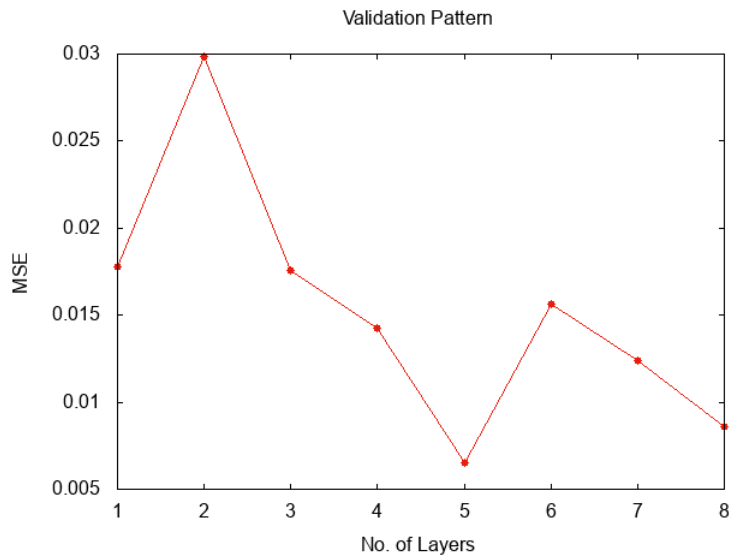
Στην παράγραφο αυτή μελετάται η επίδραση που έχουν τα επίπεδα (layers) στην ακρίβεια των δικτύων. Εκπαιδεύτηκαν οκτώ δίκτυα με βάση δεδομένων που περιέχει 60 δείγματα εκπαίδευσης, ενώ πραγματοποιήθηκε επαλήθευση σε μία περίπτωση πιστοποίησης. Στο πρώτο δίκτυο δόθηκε ένα χτυπημένο επίπεδο, στο δεύτερο δύο, κ.ο.κ. Κάθε επίπεδο που προστίθεται έχει αριθμό νευρώνων μιάμιση φορά μεγαλύτερο από τον αριθμό νευρώνων του προηγούμενου επιπέδου (λ.χ. πρώτο επίπεδο: 50 νευρώνες, δεύτερο επίπεδο: 75 νευρώνες). Στο σχήμα 5.27 φαίνονται οι αρχιτεκτονικές των δικτύων αυτών. Τα δίκτυα εκπαιδεύτηκαν για 100000 εποχές, με σταθερό ρυθμό μάθησης 0.001. Στο σχήμα 5.28 φαίνεται το διάγραμμα των σφαλμάτων MSE για τα δείγματα εκπαίδευσης συναρτήσει του αριθμού των χτυπημένων επιπέδων, ενώ στο σχήμα 5.29 φαίνεται το διάγραμμα των σφαλμάτων MSE για την περίπτωση πιστοποίησης.



Σχήμα 5.27: Αρχιτεκτονικές δικτύων με ένα, δύο, πέντε και οκτώ κρυμμένα επίπεδα



Σχήμα 5.28: Σφάλμα MSE για τα δείγματα εκπαίδευσης συναρτήσει του αριθμού των επιπέδων



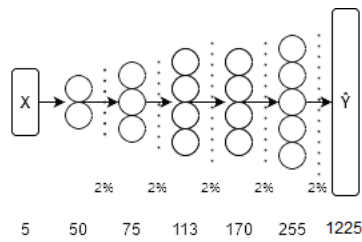
**Σχήμα 5.29:** Σφάλμα  $MSE$  για το δείγμα πιστοποίησης συναρτήσει του αριθμού των επιπέδων

Είναι φανερό πως αυξάνοντας τα κρυμμένα επίπεδα του δικτύου, αυξάνεται και η ικανότητά του να προβλέπει με μεγαλύτερη ακρίβεια τα δείγματα της εκπαίδευσης. Αντίστοιχα, για το δείγμα επαλήθευσης παρατηρείται μια αντίστοιχη συμπεριφορά, που όμως επηρεάζεται εύκολα από την τυχαία αρχικοποίηση των παραμέτρων του δικτύου.

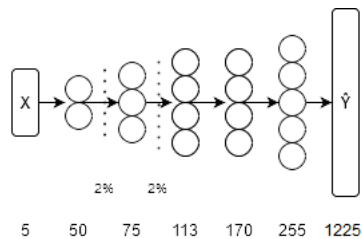
#### 5.4.2 Επίδραση Dropout

Όπως αναφέρθηκε και σε προηγούμενη παράγραφο, τα επίπεδα Dropout βοηθούν στην εξάλειψη του φαινομένου της υπερπροσαρμογής. Στην παράγραφο αυτή, γίνεται μια σύντομη ανάλυση για το πώς τα επίπεδα αυτά επηρεάζουν την ακρίβεια του δικτύου στις προβλέψεις των δειγμάτων εκπαίδευσης και του δείγματος επαλήθευσης. Τα δίκτυα που χρησιμοποιούνται σε αυτήν την ανάλυση εκπαιδεύτηκαν για 100000 εποχές με σταθερό ρυθμό μάθησης 0.001 σε βάση δεδομένων 60 δείγματα εκπαίδευσης. Η επαλήθευση γίνεται χρησιμοποιώντας ένα δείγμα πιστοποίησης.

**Επίδραση σε αρχιτεκτονική πέντε κρυμμένων επιπέδων** Χρησιμοποιώντας ένα δίκτυο πέντε κρυμμένων επιπέδων, τοποθετήθηκαν στη μία περίπτωση πέντε επίπεδα dropout (ένα μετά από κάθε κρυμμένο επίπεδο) με ποσοστά 2%, 5% και 10%, ενώ στην άλλη περίπτωση τοποθετήθηκαν δύο επίπεδα dropout με ποσοστά 2% και 5% (σχήματα 5.30 και 5.31).

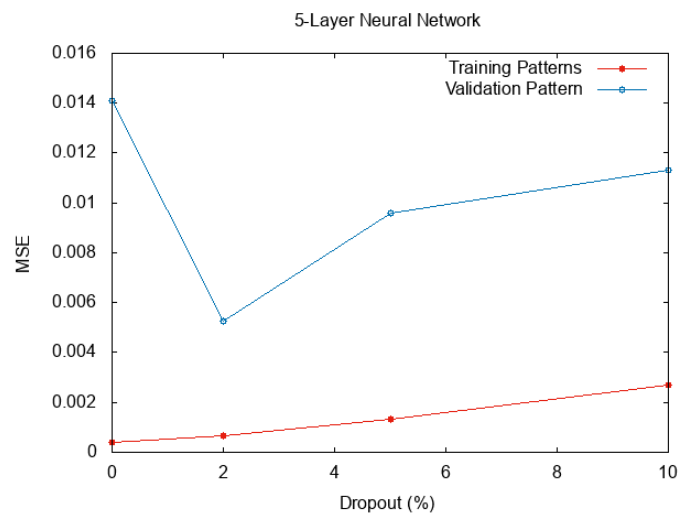


**Σχήμα 5.30:** Αρχιτεκτονική δικτύου με πέντε κρυμμένα επίπεδα και πέντε επίπεδα dropout με ποσοστό 2%



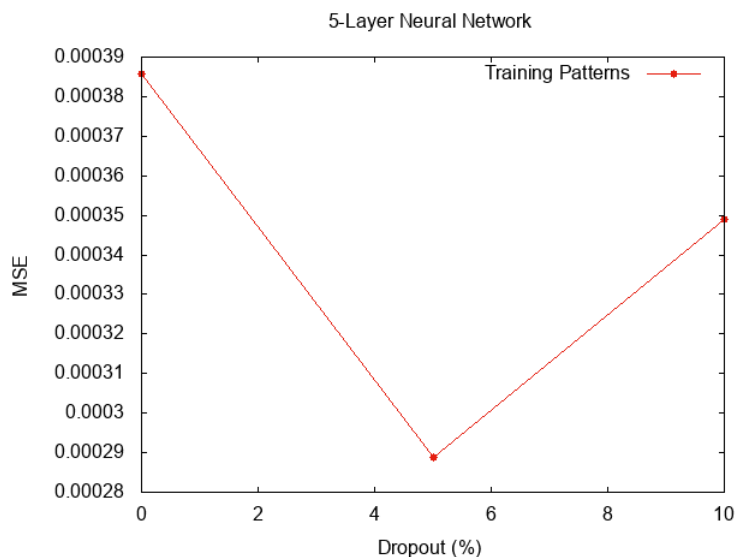
**Σχήμα 5.31:** Αρχιτεκτονική δικτύου με πέντε κρυμμένα επίπεδα και δύο επίπεδα dropout με ποσοστό 2%

Παρακάτω, στο σχήμα 5.32 φαίνεται η επίδραση που έχουν τα πέντε επίπεδα dropout στα σφάλματα MSE του δικτύου για τα δείγματα εκπαίδευσης και για το δείγμα πιστοποίησης. Με 0% dropout εννοείται το αντίστοιχο δίκτυο χωρίς τη χρήση επιπέδων dropout. Το αποτέλεσμα είναι το αναμενόμενο, καθώς αυξάνοντας το ποσοστό των νευρώνων που δεν ενεργοποιούνται, η ακρίβεια του δικτύου στις προβλέψεις των δειγμάτων εκπαίδευσης μειώνεται, ενώ για το δείγμα πιστοποίησης αυξάνεται σε ένα βαθμό.

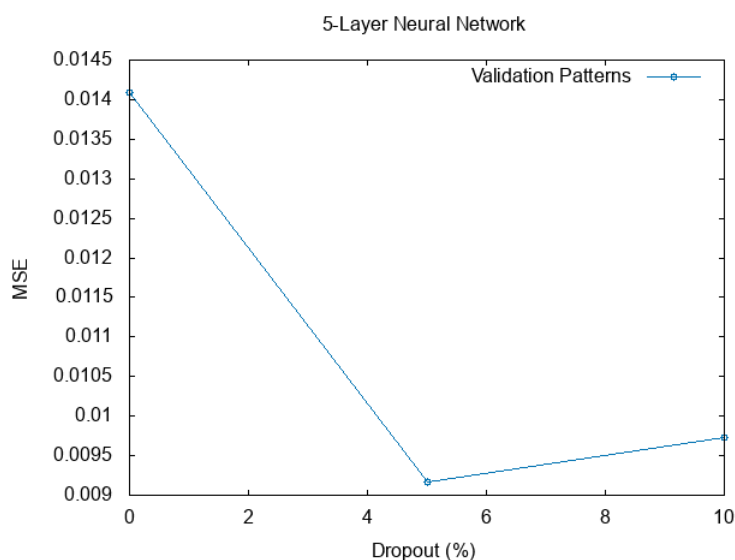


**Σχήμα 5.32:** Επίδραση του ποσοστού των νευρώνων που δεν ενεργοποιούνται σε κάθε επίπεδο στο σφάλμα MSE του δικτύου, χρησιμοποιώντας πέντε επίπεδα dropout

Στα σχήματα 5.33 και 5.34 παρουσιάζεται η επίδραση που έχουν τα δύο επίπεδα dropout αντίστοιχα. Το σφάλμα για το δείγμα επαλήθευσης φαίνεται να μειώνεται και σε αυτή την περίπτωση. Όπως φαίνεται, το σφάλμα μειώνεται και για το δείγμα εκπαίδευσης.

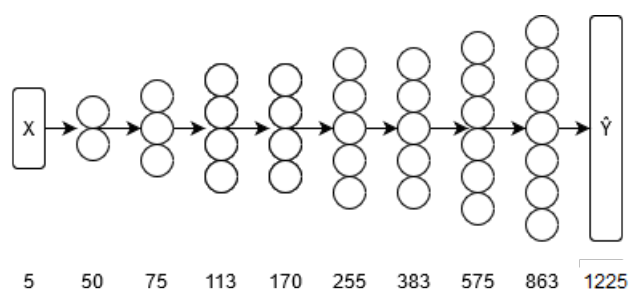


**Σχήμα 5.33:** Επίδραση του ποσοστού των νευρώνων που δεν ενεργοποιούνται σε κάθε επίπεδο στο σφάλμα MSE του δικτύου για τα δείγματα εκπαίδευσης, χρησιμοποιώντας δύο επίπεδα dropout



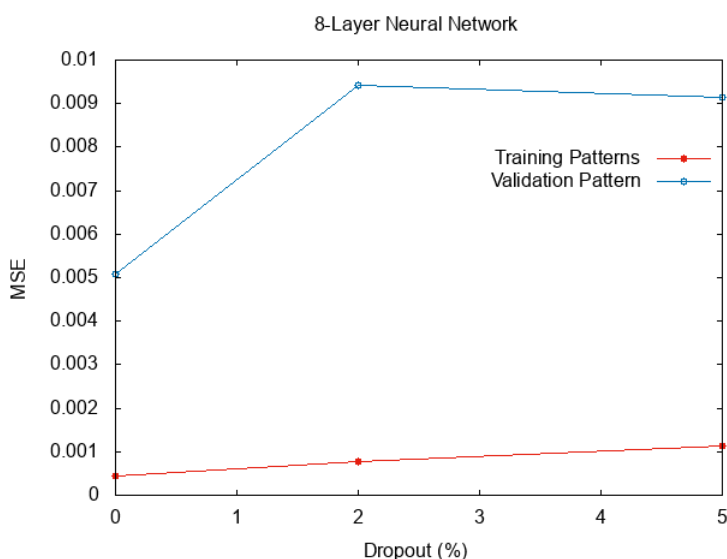
**Σχήμα 5.34:** Επίδραση του ποσοστού των νευρώνων που δεν ενεργοποιούνται σε κάθε επίπεδο στο σφάλμα MSE του δικτύου για το δείγμα πιστοποίησης, χρησιμοποιώντας δύο επίπεδα dropout

**Επίδραση σε αρχιτεκτονική οκτώ κρυμμένων επιπέδων** Στη συνέχεια παρουσιάζεται η επίδραση που έχει η χρήση επιπέδων dropout σε αρχιτεκτονική με οκτώ κρυμμένα επίπεδα. Όπως φάνηκε σε παραπάνω παράγραφο, τα οκτώ κρυμμένα επίπεδα εμφανίζουν (όπως είναι λογικό) μεγαλύτερη ικανότητα μάθησης των δειγμάτων εκπαίδευσης. Στη μία περίπτωση τοποθετήθηκαν οκτώ επίπεδα dropout (ένα μετά από κάθε κρυμμένο επίπεδο) με ποσοστά 2% και 5%, ενώ στην άλλη τοποθετήθηκαν δύο επίπεδα με ποσοστό 5%.

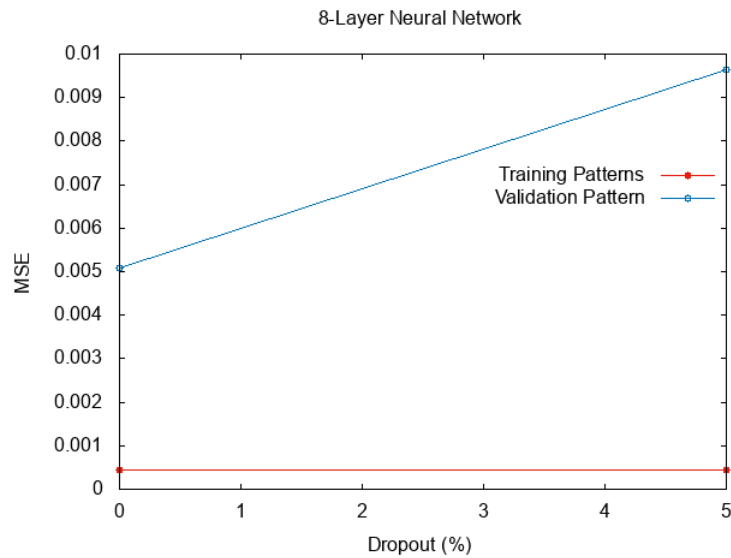


**Σχήμα 5.35:** Το δίκτυο οκτώ κρυμμένων επιπέδων

Στο σχήμα 5.36 φαίνεται η επίδραση του ποσοστού των νευρώνων που δεν ενεργοποιούνται σε κάθε επίπεδο, για την περίπτωση των οκτώ επιπέδων dropout. Παρατηρείται πως η χρήση dropout για τη συγκεκριμένη αρχιτεκτονική μειώνει γενικά την ακρίβεια του δικτύου. Αντίστοιχα, στο σχήμα 5.37 φαίνονται τα αποτελέσματα για την περίπτωση των δύο επιπέδων dropout, στα οποία παρατηρείται επίσης μείωση της ακρίβειας.



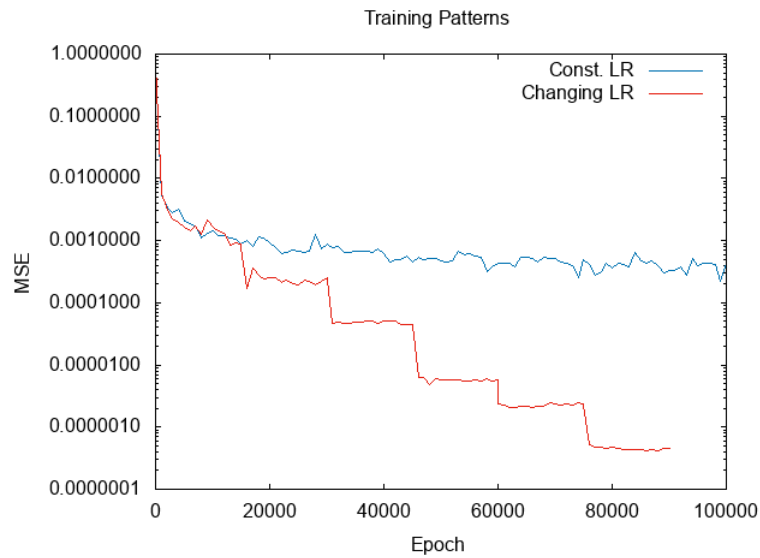
**Σχήμα 5.36:** Επίδραση του ποσοστού των νευρώνων που δεν ενεργοποιούνται σε κάθε επίπεδο στο σφάλμα MSE του δικτύου, χρησιμοποιώντας οκτώ επίπεδα dropout



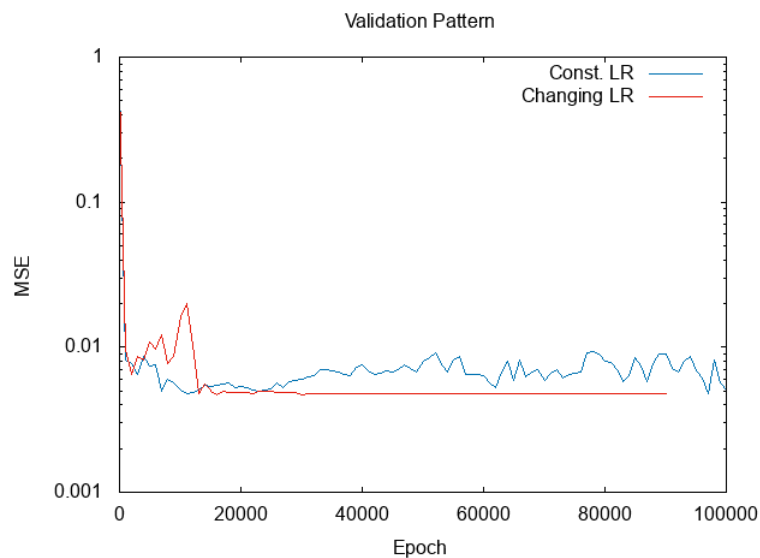
**Σχήμα 5.37:** Επίδραση του ποσοστού των νευρώνων που δεν ενεργοποιούνται σε κάθε επίπεδο στο σφάλμα MSE του δικτύου, χρησιμοποιώντας δύο επίπεδα dropout

### 5.4.3 Επίδραση του Ρυθμού Μάθησης (Learning Rate)

Μία υπερπαράμετρος που τίθεται προς ρύθμιση, είναι ο ρυθμός μάθησης. Εν γένει, με τη χρήση υψηλού ρυθμού μάθησης το δίκτυο θα δυσκολευτεί να βρεί τοπικό ελάχιστο, καθώς παρουσιάζεται ταλάντωση στις τιμές του σφάλματος. Αντίστοιχα, με τη χρήση πολύ μικρού ρυθμού μάθησης, το δίκτυο θα παγιδευτεί σε τοπικό ελάχιστο, που όμως μπορεί να μην παρέχει ικανοποιητική ακρίβεια. Φυσικά, οι όροι 'μεγάλος' και 'μικρός' ρυθμός μάθησης δεν μπορούν να αποδοθούν με τιμές στη γενική περίπτωση, καθώς αυτές εξαρτώνται από την εκάστοτε εφαρμογή. Για την ανάλυση της επίδρασης του ρυθμού μάθησης, εκπαιδεύτηκε δίκτυο οκτώ κρυμμένων επιπέδων σε βάση δεδομένων 60 δειγμάτων εκπαίδευσης, ενώ έγινε επαλήθευση με ένα δείγμα πιστοποίησης. Στη μία περίπτωση, το δίκτυο εκπαιδεύτηκε για 100000 εποχές με σταθερό ρυθμό μάθησης 0.001, ενώ στη δεύτερη περίπτωση το δίκτυο εκπαιδεύτηκε για 90000 εποχές μειώνοντας το ρυθμό μάθησης χειροκίνητα από 0.002 έως  $8 \cdot 10^{-6}$  ανά 15000 εποχές. Το σφάλμα MSE για τα δείγματα εκπαίδευσης φαίνεται στο σχήμα 5.38, ενώ στο σχήμα 5.39 φαίνεται το σφάλμα για το δείγμα πιστοποίησης. Παρατηρείται έντονη διαφορά στην ακρίβεια του δικτύου στις προβλέψεις των δειγμάτων εκπαίδευσης, καθώς το σφάλμα MSE μειώνεται αρκετές τάξεις μεγέθους. Για το δείγμα πιστοποίησης, παρατηρείται πως εξαλείφθηκε η ταλαντωτική συμπεριφορά που εμφάνιζε το δίκτυο με σταθερό ρυθμό μάθησης.



**Σχήμα 5.38:** Σφάλμα MSE για τα δείγματα εκπαίδευσης του δικτύου συναρτήσει των εποχών, για τις περιπτώσεις σταθερού και ρυθμιζόμενου ρυθμού μάθησης



**Σχήμα 5.39:** Σφάλμα MSE για το δείγμα πιστοποίησης του δικτύου συναρτήσει των εποχών, για τις περιπτώσεις σταθερού και ρυθμιζόμενου ρυθμού μάθησης

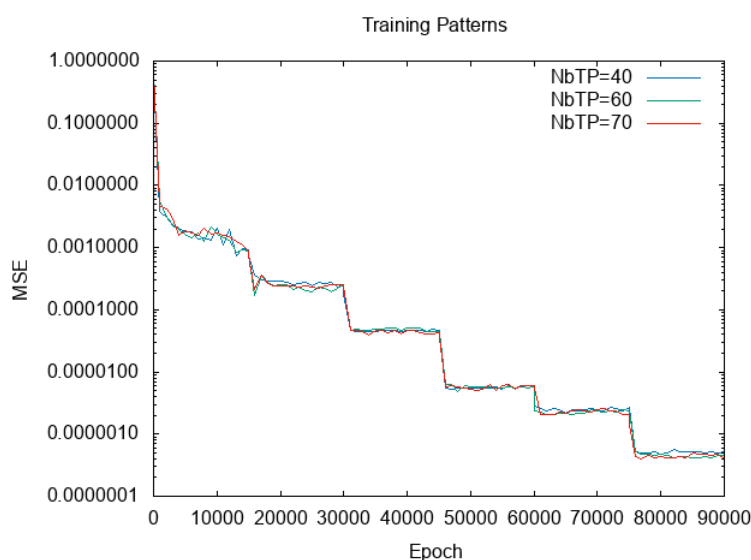
#### 5.4.4 Επίδραση της Βάσης Δεδομένων

Στην παράγραφο αυτή αναλύεται η επίδραση που έχει η βάση δεδομένων στην ακρίβεια των DNN σε περιπτώσεις που δεν περιλαμβάνονται στα δείγματα εκπαίδευσης. Για το σκοπό αυτό, συγκροτήθηκαν τρεις βάσεις δεδομένων: η πρώτη με 40 δείγματα εκπαίδευ-



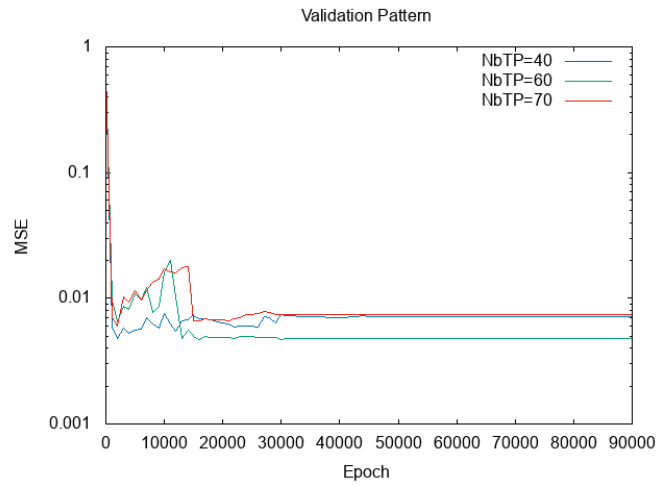
σης, η δεύτερη εμπλουτισμένη με άλλα 20 περιλαμβάνει σύνολο 60 δείγματα εκπαίδευσης και η τρίτη εμπλουτισμένη με άλλα 10 περιλαμβάνει σύνολο 70. Έτσι, επιλέχθηκε το δίκτυο των οκτώ κρυμμένων επιπέδων που χρησιμοποιήθηκε και σε παραπάνω παραγράφους και εκπαιδεύτηκε για 90000 εποχές με ρυθμιζόμενο ρυθμό μάθησης.

Στο σχήμα 5.40 φαίνεται το σφάλμα MSE συναρτήσει των εποχών για τις τρεις διαφορετικές βάσεις δεδομένων. Παρατηρείται πως δεν υπάρχει μεγάλη διαφορά στην ακρίβεια του δικτύου για τα δείγματα εκπαίδευσης, πλην της ελαφρώς ταλαντωτικής συμπεριφοράς του δικτύου στις πιο 'πτωχές' βάσεις δεδομένων.

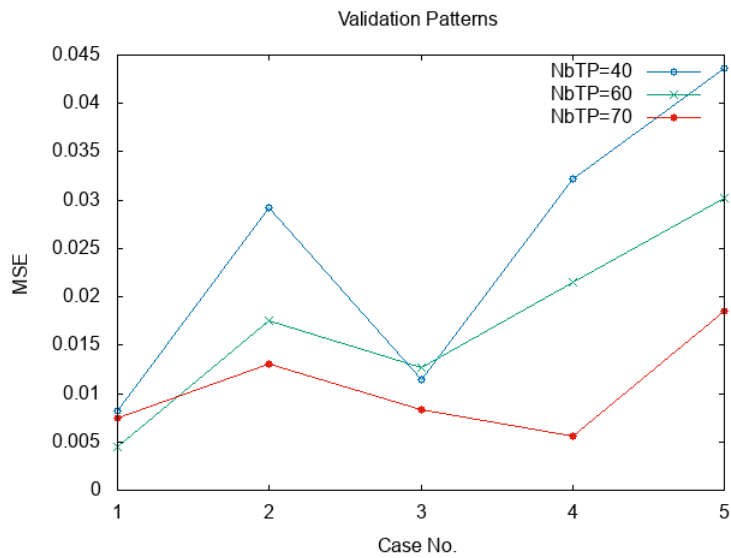


**Σχήμα 5.40:** Σφάλμα MSE για τα δείγματα εκπαίδευσης του δικτύου συναρτήσει των εποχών, για τις περιπτώσεις των διαφορετικών βάσεων δεδομένων

Στο σχήμα 5.41 παρουσιάζεται το σφάλμα MSE συναρτήσει των εποχών για το ίδιο δείγμα πιστοποίησης. Αξιοσημείωτο είναι το γεγονός πως το δίκτυο που εκπαιδεύτηκε σε 60 δείγματα εκπαίδευσης προβλέπει με μεγαλύτερη ακρίβεια το δείγμα πιστοποίησης. Αυτό δεν είναι το αναμενόμενο αποτέλεσμα και για το λόγο αυτό, πραγματοποιήθηκε περαιτέρω σύγκριση των τριών περιπτώσεων σε άλλες τέσσερις περιπτώσεις πιστοποίησης. Οι πέντε σε σύνολο περιπτώσεις πιστοποίησης είναι οι ίδιες που παρουσιάστηκαν σε προηγούμενη ενότητα (σχήμα 5.26). Τα αποτελέσματα φαίνονται στο σχήμα 5.42. Παρατηρείται το αναμενόμενο αποτέλεσμα, καθώς φαίνεται πως οι πιο 'πλούσιες' βάσεις δεδομένων οδηγούν το δίκτυο σε πιο ακριβείς εκτιμήσεις.



**Σχήμα 5.41:** Σφάλμα *MSE* για το ίδιο δείγμα πιστοποίησης του δικτύου συναρτήσει των εποχών, για τις περιπτώσεις των διαφορετικών βάσεων δεδομένων



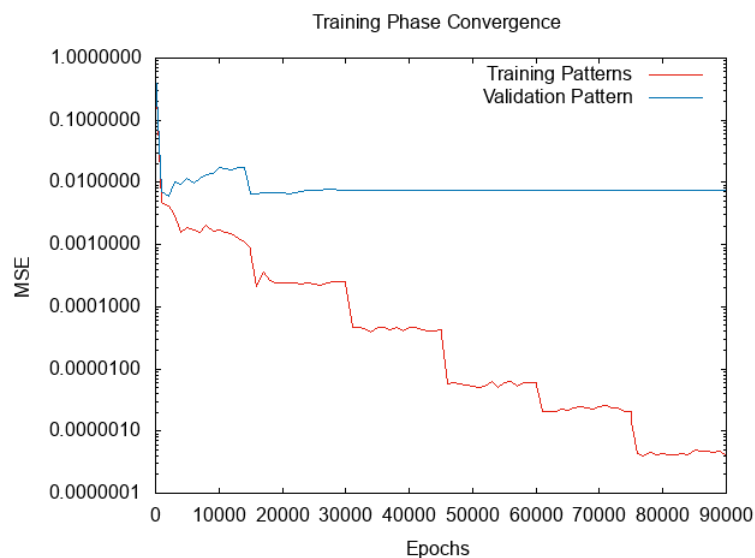
**Σχήμα 5.42:** Σφάλμα *MSE* για τα ίδια δείγματα πιστοποίησης, για τις περιπτώσεις των διαφορετικών βάσεων δεδομένων

## 5.5 Εκπαίδευση DNN και Πρόβλεψη της Ροής στον Αγωγό

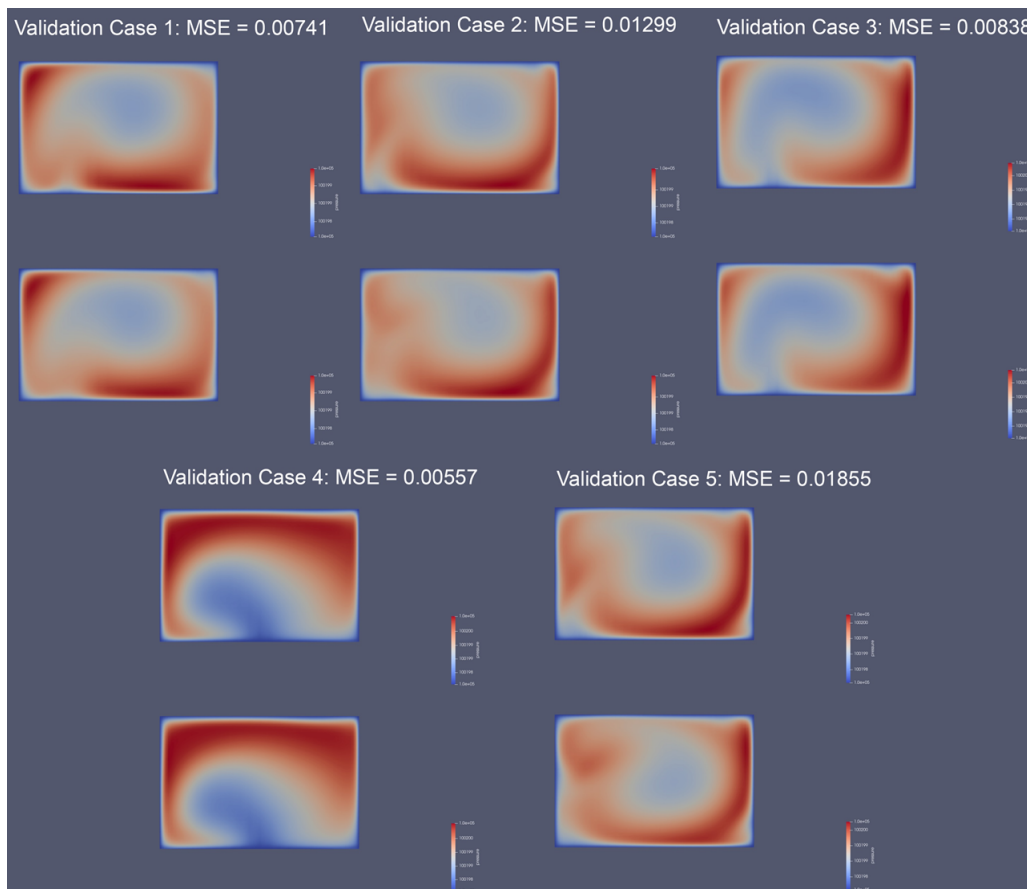
Μετά την παραπάνω ανάλυση, μπορεί να γίνει η ρύθμιση των υπερπαραμέτρων ενός δικτύου, το οποίο μετά απο εκπαίδευση θα είναι ικανό να προβλέπει με ικανοποιητική ακρίβεια τις ολικές πιέσεις της διατομής εξόδου σε οποιαδήποτε γεωμετρία. Έτσι, επιλέχθηκαν οι παρακάτω παράμετροι:

- Δίκτυο με αρχιτεκτονική οκτώ κρυμμένων επιπέδων, χωρίς χρήση dropout
- Βάση δεδομένων με 70 δείγματα εκπαίδευσης
- Ρυθμός μάθησης που ρυθμίζεται χειροκίνητα ανά 15000 εποχές, ξεκινώντας από 0.002 και καταλήγοντας στην τιμή  $8 \cdot 10^{-6}$
- Συνολικός αριθμός εποχών για την εκπαίδευση: 90000

Η σύγκλιση του δικτύου φαίνεται στο σχήμα 5.43. Η εκπαίδευση διήρκησε περίπου 30mins σε έναν επεξεργαστή GPU Nvidia Tesla K20m, χρησιμοποιώντας (όπως έχει ήδη αναφερθεί) τη βιβλιοθήκη tensorflow της γλώσσας python. Στο σχήμα 5.44 παρουσιάζονται οι εκτιμήσεις του δικτύου στις πέντε περιπτώσεις πιστοποίησης.



Σχήμα 5.43: Σφάλμα MSE συναρτήσει των εποχών του δικτύου που επιλέχθηκε



**Σχήμα 5.44:** Οι εκτιμήσεις του δικτύου στις πέντε γεωμετρίες πιστοποίησης. Πάνω: Πραγματικές ολικές πιέσεις (CFD). Κάτω: Προβλέψεις του δικτύου DNN

## 5.6 Βελτιστοποίηση Γεωμετρίας Αγωγού με χρήση EA & DNN

Στην ενότητα αυτή επαναλαμβάνεται η βελτιστοποίηση της γεωμετρίας του αγωγού, αυτή τη φορά χρησιμοποιώντας στη θέση του λογισμικού CFD το δίκτυο DNN που παρουσιάστηκε παραπάνω.

### 5.6.1 Παρουσίαση της Περίπτωσης

Η περίπτωση της βελτιστοποίησης είναι η ίδια με του τρίτου κεφάλαιου. Υπενθυμίζονται οι μεταβλητές σχεδιασμού:

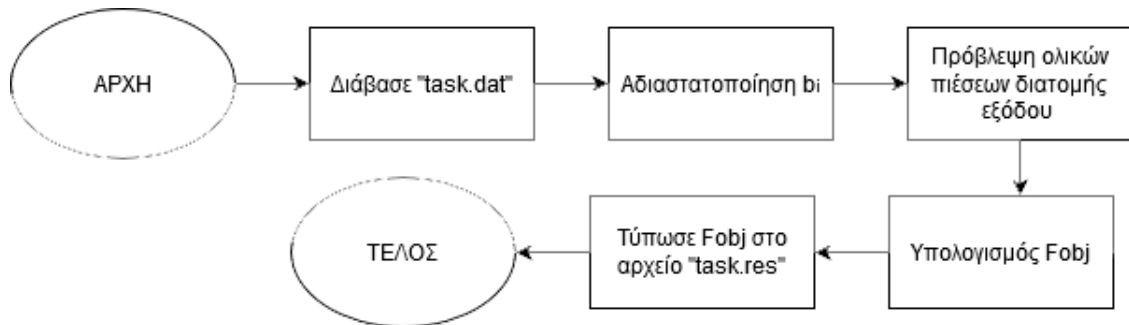
$$\vec{b} = [b_1 \ b_2 \ b_3 \ b_4 \ b_5] = [a_1 \ a_2 \ r_1 \ r_2 \ l_1]$$

Η συνάρτηση στόχος είναι οι απώλειες ολικής πίεσης. Να σημειωθεί πως οι πιέσεις που χρησιμοποιούνται για τον υπολογισμό των απωλειών ολικής πίεσης, δεν είναι σταθμισμένες με την παροχή (όπως στην περίπτωση του λογισμικού CFD). Συνεπώς, στη διατομή εξόδου (που έχει 1225 κόμβους) η μέση πίεση υπολογίζεται ως αριθμητική μέση τιμή:  $P_{t,out} = \frac{1}{1225} \sum_1^{1225} p_{t,i}$

Οι ρυθμίσεις του εξελικτικού αλγορίθμου είναι οι ίδιες που χρησιμοποιήθηκαν στο κεφάλαιο 3, ενώ πάλι χρησιμοποιήθηκε το λογισμικό EASY, χωρίς τη χρήση μετα-προτύπων (MAEA) αυτή τη φορά. Στο εξής, θα γίνεται αναφορά στην περίπτωση της βελτιστοποίησης με χρήση του βαθιού νευρωνικού δικτύου ως EA & DNN.

### 5.6.2 Υλοποίηση της διαδικασίας

Το λογισμικό EASY παράγει το αρχείο 'task.dat' και μέσω της αξιολόγησης της υποψήφιας λύσης αναμένει την επιστροφή του αρχείου 'task.res'. Την αξιολόγηση των λύσεων, πλέον, δεν αναλαμβάνει το λογισμικό CFD, και συνεπώς δεν χρησιμοποιούνται τα scripts που παρουσιάστηκαν στο τρίτο κεφάλαιο. Στη θέση του λογισμικού CFD εισέρχεται το εκπαιδευμένο δίκτυο DNN. Έτσι, για την αξιολόγηση των υποψήφιας λύσεων ακολουθείται το παρακάτω διάγραμμα ροής, μέσω ενός script:



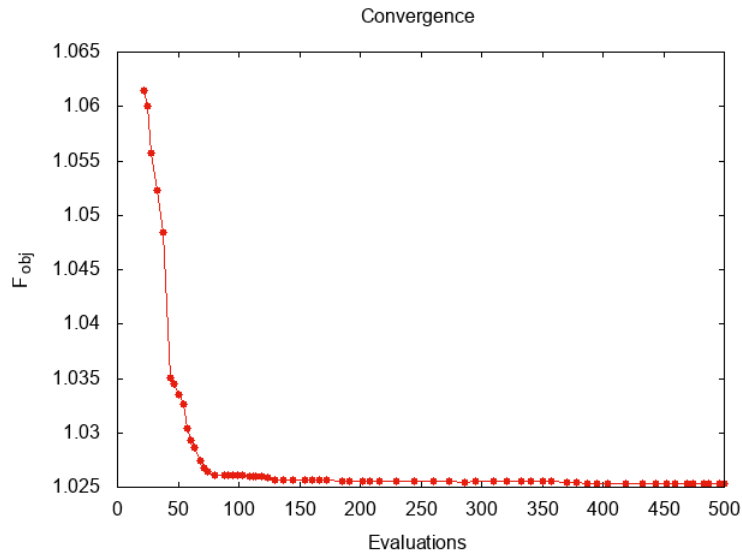
Σχήμα 5.45: Διάγραμμα ροής της αξιολόγησης των υποψήφιας λύσεων

### 5.6.3 Αποτελέσματα

Χρησιμοποιώντας το λογισμικό EASY, η βελτιστοποίηση τερματίστηκε στις 500 αξιολογήσεις του βαθιού νευρωνικού δικτύου, ενώ διήρκτησε:  $15sec/eval \times 500evals = 125mins \approx 2h$ . Η σύγκλιση της βελτιστοποίησης φαίνεται στο σχήμα 5.46. Η τελική λύση είναι η εξής:

$$\hat{\vec{b}} = [111 \ 109 \ 0.10803 \ 0.11874 \ 0.12]$$

όπου με  $\hat{\vec{b}}$  συμβολίζεται η λύση που βρέθηκε χρησιμοποιώντας DNN για την αξιολόγηση των λύσεων.



**Σχήμα 5.46:** Η σύγκλιση της βελτιστοποίησης χρησιμοποιώντας DNN ως λογισμικό αξιολόγησης. Οριζόντιος άξονας: Αριθμός αξιολογήσεων. Κατακόρυφος άξονας: Τιμή συνάρτησης-στόχου

#### 5.6.4 Σύγκριση με χρήση λογισμικού CFD

Στην παράγραφο αυτή, γίνεται μια σύγκριση των βελτιστοποιήσεων μεταξύ της χρήσης λογισμικού CFD (MAEA & CFD) και της χρήσης DNN (EA & DNN) για την αξιολόγηση των λύσεων.

##### Κόστος βελτιστοποίησης με χρήση CFD

$$35min/eval_{CFD} \times 167evals_{CFD} \approx 97h$$

##### Κόστος βελτιστοποίησης EA & DNN

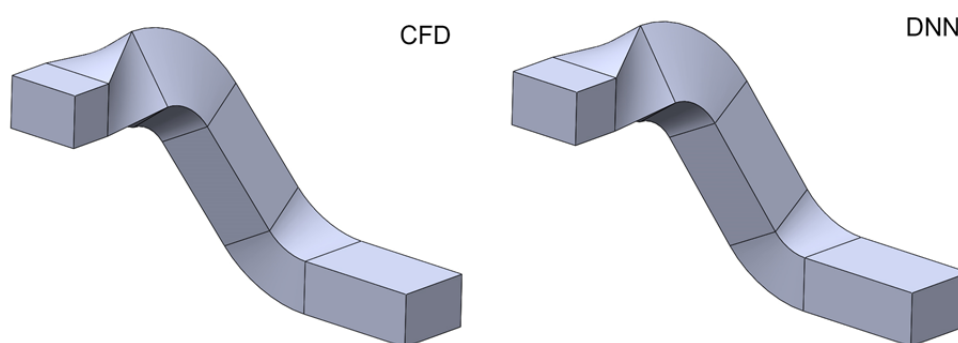
- Κόστος σύνθεσης βάσης δεδομένων:  $(70TP+5VP) \times 35min/CFD_{run} \approx 43.5h$
- Κόστος εκπαίδευσης DNN:  $30mins$
- Κόστος βελτιστοποίησης:  $15sec/eval_{DNN} \times 500evals_{DNN} \approx 2h$

Έτσι, το συνολικό κόστος ανέρχεται στις  $46h$ , δηλαδή μειώθηκε κατά  $\approx 52\%$ . Η τελική λύση που βρέθηκε χρησιμοποιώντας το DNN δεν είναι η ίδια που βρέθηκε χρησιμοποιώντας το λογισμικό CFD, ωστόσο οι δύο λύσεις είναι πολύ κοντά γεωμετρικά (σχήματα 5.47 και 5.48). Η απόκλιση των τελικών λύσεων οφείλεται στα σφάλματα των προβλέψεων του DNN και στο γεγονός πως οι απώλειες ολικής πίεσης στην περίπτωση MAEA & CFD υπολογίζονται από τη μέση ολική πίεση εξόδου, σταθμισμένη από την

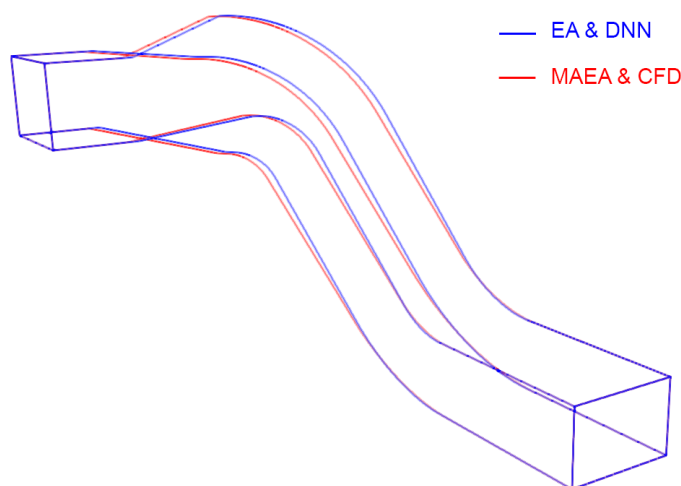
παροχή, ενώ στην περίπτωση EA & DNN υπολογίζονται με την αριθμητική μέση τιμή των ολικών πιέσεων εξόδου.

$$\hat{\vec{b}} = [111 \ 109 \ 0.10803 \ 0.11874 \ 0.12]$$

$$\vec{b} = [111 \ 111 \ 0.12 \ 0.12 \ 0.12]$$



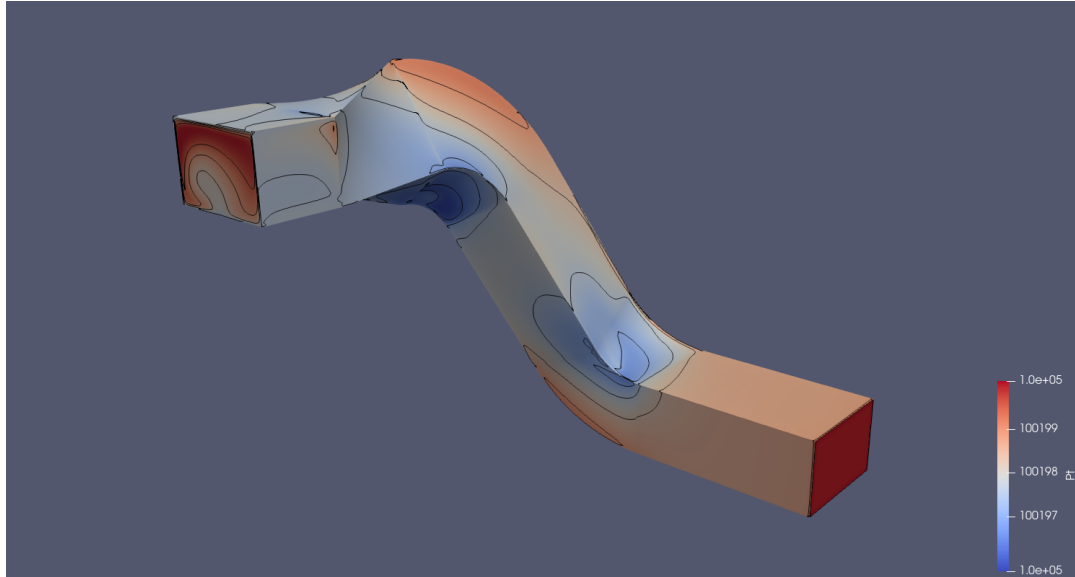
Σχήμα 5.47: Οι τελικές λύσεις, MAEA & CFD και EA & DNN



Σχήμα 5.48: Οι τελικές λύσεις, MAEA & CFD και EA & DNN

**Επαν-αξιολόγηση της τελικής λύσης με το λογισμικό CFD** Η ροή επιλύεται για την τελική λύση χρησιμοποιώντας το λογισμικό CFD PUMA. Ως μοντέλο τύρβης χρησιμοποιείται το μοντέλο Spalart-Allmaras. Για τη σύγκριση των τελικών λύσεων MAEA & CFD και EA & DNN που φαίνεται στον πίνακα 5.6, χρησιμοποιούνται

οι σταθμισμένες απώλειες ολικής πίεσης που εξάγει το λογισμικό PUMA. Παρατηρείται πως οι δύο λύσεις έχουν παρόμοιες απώλειες ολικής πίεσης.



Σχήμα 5.49: Το πεδίο των ολικών πιέσεων για την τελική λύση EA & DNN

	MAEA&CFD	EA&DNN
$\Delta Pt$ (Pa)	0.7553	0.7632

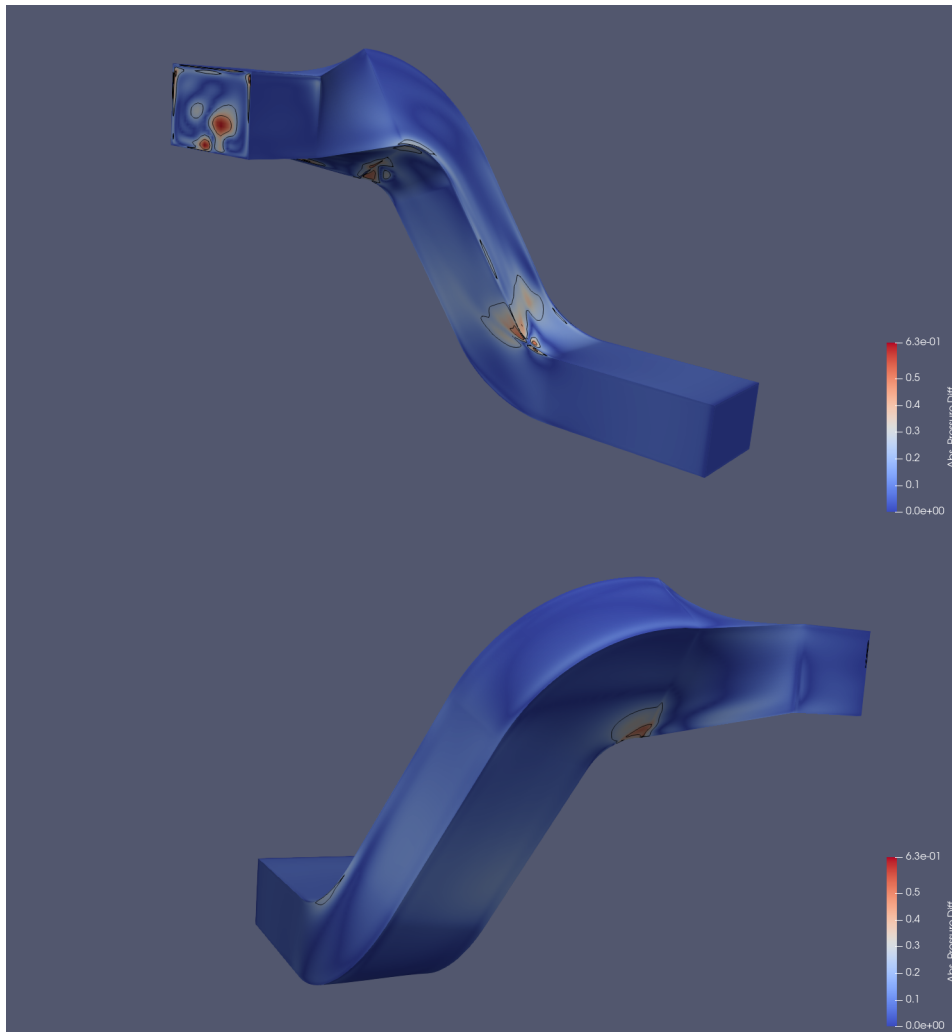
Πίνακας 5.6: Οι σταθμισμένες απώλειες ολικής πίεσης για τις βέλτιστες λύσεις των περιπτώσεων MAEA & CFD και EA & DNN

**Πιστοποίηση της λύσης EA & DNN με το μοντέλο τύρβης k- $\omega$  SST**  
 Για την πιστοποίηση της ακρίβειας της τελικής λύσης EA & DNN, η ροή επιλύεται ξανά, χρησιμοποιώντας το μοντέλο τύρβης k- $\omega$  SST.

$\Delta Pt$ (Pa)	Sp.-Allm.	SST k- $\omega$
EA&DNN	0.7632	0.7673

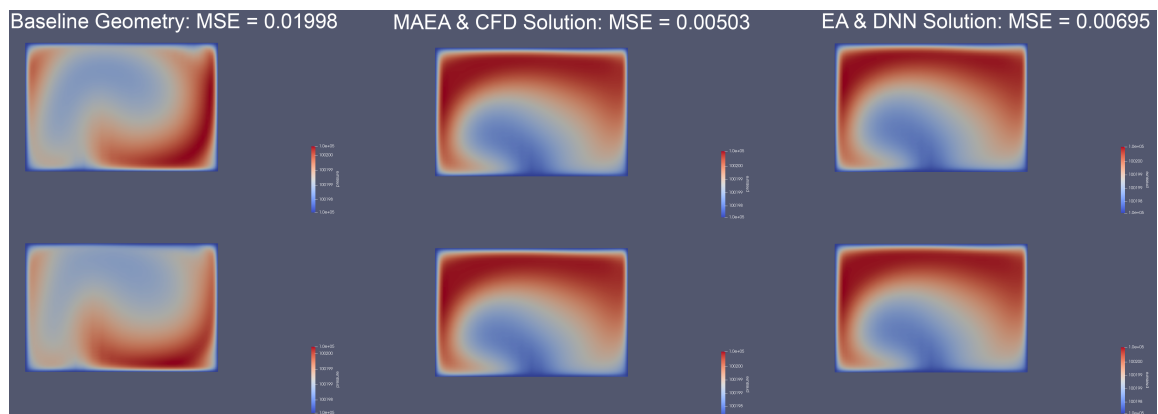
Πίνακας 5.7: Οι σταθμισμένες απώλειες ολικής πίεσης για τη λύση EA & DNN χρησιμοποιώντας τα δύο μοντέλα τύρβης





**Σχήμα 5.50:** Οι αποκλίσεις των ολικών πιέσεων για τη λύση EA & DNN των μοντέλων *Spalart-Allmaras* και *k- $\omega$  SST*

**Πιστοποίηση του DNN στην αρχική γεωμετρία και τις δύο τελικές λύσεις** Για περαιτέρω πιστοποίηση του δικτύου, το ίδιο χρησιμοποιείται για την πρόβλεψη των πιέσεων στις περιπτώσεις της αρχικής γεωμετρίας, της τελικής λύσης από τη βελτιστοποίηση MAEA & CFD και της τελικής λύσης EA & DNN (σχήμα 5.51). Η σύγκριση γίνεται μεταξύ των πεδίων πιέσεων από τα αποτελέσματα του λογισμικού CFD PUMA και των προβλέψεων του δικτύου.



**Σχήμα 5.51:** Οι εκτιμήσεις του δικτύου και τα αντίστοιχα σφάλματα *MSE* για την αρχική γεωμετρία και για τις δύο τελικές λύσεις (*MAEA* & *CFD* και *EA* & *DNN*). Πάνω: Πραγματικές ολικές πιέσεις (*CFD*). Κάτω: Προβλέψεις του δικτύου *DNN*

# Κεφάλαιο 6

## Ανακεφαλαίωση - Συμπεράσματα

### 6.1 Ανακεφαλαίωση

Στη διπλωματική αυτή εργασία, παρουσιάστηκε, αρχικά, μια μέθοδος για την ένταξη ενός λογισμικού CAD στο βρόχο της βελτιστοποίησης του σχήματος ενός αγωγού. Στο πρώτο κεφάλαιο, παρουσιάστηκε ο αγωγός που χρησιμοποιείται ως παράδειγμα σε αυτή την εργασία. Στο δεύτερο κεφάλαιο, παρουσιάστηκε ο τρόπος με τον οποίο πραγματοποιήθηκε η σχεδίαση της γεωμετρίας στο λογισμικό Solidworks και η γένεση του πλέγματος στο λογισμικό Pointwise, όπως επίσης και ένας τρόπος να πραγματοποιηθούν τα παραπάνω αυτόματα με τη χρήση των λειτουργιών 'equations' και macro και τη χρήση script.

Στο τρίτο κεφάλαιο, παρουσιάστηκε μια εκτενής ανάλυση της προσομοίωσης της ροής, η οποία πραγματοποιήθηκε χρησιμοποιώντας το λογισμικό CFD PUMA. Η ανάλυση περιελάμβανε τη μελέτη της ανεξαρτησίας του πλέγματος και τη σύγκριση δύο μοντέλων τύρβης. Με κριτήριο τα παραπάνω, επιλέχθηκε η τελική μοντελοποίηση του προβλήματος, όσον αφορά την επίλυση της ροής.

Στο τέταρτο κεφάλαιο, παρουσιάστηκε η υλοποίηση της διαδικασίας βελτιστοποίησης του σχήματος του αγωγού, η οποία πραγματοποιήθηκε χρησιμοποιώντας το λογισμικό εξελικτικών αλγορίθμων βελτιστοποίησης EASY. Εκτός από την επιλογή των κατάλληλων ρυθμίσεων του λογισμικού, παρουσιάστηκε μια μέθοδος ένταξης του λογισμικού CAD και του πλεγματοποιητή στο βρόχο της βελτιστοποίησης, χρησιμοποιώντας scripts για την επικοινωνία μεταξύ των λογισμικών Solidworks και Pointwise. Χρησιμοποιώντας ως λογισμικό αξιολόγησης τον επιλύτη CFD PUMA, παρουσιάστηκαν τα αποτελέσματα και το κόστος της βελτιστοποίησης.

Στο πέμπτο κεφάλαιο, εισήχθη η έννοια και παρουσιάστηκε η βασική λειτουργία των νευρικών δικτύων, ενώ αναλύθηκε η έννοια των υπερπαραμέτρων. Ύστερα, παρου-

σιάστηκε η υλοποίηση της διαδικασίας της εκπαίδευσης των DNN, όπως επίσης και διάφορες μεθοδολογίες χρήσης των DNN για την πρόβλεψη της ροής, ως υποκατάστατο χρήσης λογισμικού CFD. Επιλέγοντας την κατάλληλη μεθοδολογία για το πλαίσιο της εργασίας αυτής, παρουσιάστηκε μια εκτενής ανάλυση της επίδρασης των υπερπαραμέτρων στην ακρίβεια των δικτύων. Αφού πραγματοποιήθηκε η επιλογή του τελικού δικτύου και η εκπαίδευσή του, παρουσιάστηκε η ακρίβειά του σε πέντε περιπτώσεις επαλήθευσης. Τέλος, το εκπαιδευμένο δίκτυο DNN χρησιμοποιήθηκε ως λογισμικό αξιολόγησης στο βρόχο της βελτιστοποίησης σχήματος του αγωγού, αντικαθιστώντας με αυτόν τον τρόπο το λογισμικό CFD και παρουσιάστηκε το μειωμένο κόστος της βελτιστοποίησης.

Συνοψίζοντας, η ένταξη λογισμικών CAD στο βρόχο της βελτιστοποίησης ανοίγει τους ορίζοντες στο χώρο του σχεδιασμού σε προβλήματα υπολογιστικής ρευστοδυναμικής, καθώς με τον τρόπο αυτόν μπορούν να σχεδιαστούν γεωμετρίες, για τις οποίες δεν απαιτείται παραμετροποίηση με μαθηματικές σχέσεις. Τα βαθιά νευρωνικά δίκτυα, ενώ βρίσκουν εφαρμογή σε πολλές περιπτώσεις προβλημάτων, κρίνονται αξιόπιστα και για την πρόβλεψη ροών σε προβλήματα ρευστοδυναμικής, καθώς με την κατάλληλη εκπαίδευσή τους, μπορούν να αντικαταστήσουν το λογισμικό CFD στο βρόχο της βελτιστοποίησης και να λειτουργήσουν ως το αντίστοιχο λογισμικό αξιολόγησης. Με τον τρόπο αυτό, μειώνεται σημαντικά το κόστος της βελτιστοποίησης, ενώ το σφάλμα είναι ικανοποιητικά μικρό.

## 6.2 Συμπεράσματα

Ανακεφαλαιώνοντας, παρακάτω παρουσιάζονται τα συμπεράσματα που εξήχθη για αυτήν την διπλωματική εργασία:

- Υλοποιήθηκε η αλληλουχία κλήσεων λογισμικών Solidworks (CAD), Pointwise (πλεγματοποιητής) και PUMA (CFD). Με την παραπάνω αλυσίδα λογισμικών, πραγματοποιήθηκε βελτιστοποίηση σχήματος στο σχεδιασμό ενός αγωγού (MAEA & CFD). Φυσικά, ο αγωγός αποτελεί απλώς ένα παράδειγμα και η παραπάνω τεχνική μπορεί να χρησιμοποιηθεί (με πιθανώς μικρές τροποποιήσεις) και σε άλλες περιπτώσεις, δηλαδή η μέθοδος αυτή έχει το σημαντικό χαρακτηριστικό της εναλλαξιμότητας.
- Παρουσιάστηκε η επιλογή τριών προσεγγίσεων για την πρόβλεψη της ροής με χρήση βαθιού νευρωνικού δικτύου και αναδεικνύεται η αποτελεσματικότητα της τρίτης από αυτές.
- Παρουσιάστηκε η δυνατότητα αντικατάστασης του λογισμικού αξιολόγησης στο βρόχο της βελτιστοποίησης με εκπαιδευμένο βαθύ νευρωνικό δίκτυο (EA & DNN) και οι απαιτήσεις που προκύπτουν για την υλοποίηση αυτής. Παρουσιάστηκε η ακρίβεια της μεθόδου και το όφελος σε υπολογιστικό κόστος (μείωση

αυτού κατά περίπου 50%).

- Πραγματοποιήθηκε η πιστοποίηση του μοντέλου τύρβης Spalart-Allmaras σε σύγκριση με το k- $\omega$  SST. Τα αποτελέσματα για την αρχική γεωμετρία και τις τελικές λύσεις MAEA & CFD και EA & DNN συνοψίζονται στον πίνακα 6.1

Σημειώνεται πως όλα τα σχόλια που περιέχει αυτή η Διπλωματική εργασία βασίζονται στην περίπτωση του αγωγού που παρουσιάστηκε. Η επαλήθευση αυτών οφείλει να πραγματοποιηθεί σε διαφορετικές περιπτώσεις/εφαρμογές.

$\Delta P_t$ (Pa)	Sp.-Allm.	SST k- $\omega$
Baseline	1.1150	1.1151
MAEA&CFD	0.7553	0.7586
EA&DNN	0.7632	0.7673

**Πίνακας 6.1:** Σύγκριση των απωλειών ολικής πίεσης για τις προσομοιώσεις CFD με χρήση των μοντέλων Spalart-Allmaras και k- $\omega$  SST.

### 6.3 Προτάσεις για μελλοντική έρευνα

Η πορεία της εργασίας αυτής, ανέδειξε σημεία άξια περαιτέρω έρευνας. Παρακάτω παρουσιάζονται τα σημεία αυτά:

- Αυτοματοποιημένη γένεση πλέγματος σε περίπτωση μη δομημένου πλέγματος. Η γεωμετρία του αγωγού που λειτούργησε ως παράδειγμα στην εργασία αυτή, επέτρεψε τη χρήση δομημένου πλέγματος. Η τεχνική αυτόματης γένεσης γεωμετρίας και πλέγματος που χρησιμοποιήθηκε αφήνει περιθώρια εξέλιξής της σε γεωμετρίες που απαιτούν μη-δομημένα και υβριδικά πλέγματα.
- Περαιτέρω πιστοποίηση μοντέλου τύρβης Spalart-Allmaras σε πρόβλημα που υπάρχουν διαθέσιμα πειραματικά δεδομένα. Το μοντέλο τύρβης Spalart-Allmaras, όπως παρουσιάστηκε στην εργασία, παράγει αποτελέσματα ικανοποιητικής ακρίβειας. Ωστόσο, η πιστοποίησή του πραγματοποιήθηκε με τη χρήση ενός άλλου μοντέλου τύρβης. Συνεπώς, υπάρχει περιθώριο για περαιτέρω πιστοποίηση του μοντέλου Spalart-Allmaras, η οποία μπορεί να πραγματοποιηθεί με πειραματικά δεδομένα.
- Εύρεση βελτιστοποιημένου τρόπου ρύθμισης των υπερπαραμέτρων των νευρωνικών δικτύων. Η ρύθμιση των υπερπαραμέτρων στα νευρωνικά δίκτυα είναι μια δύσκολη διαδικασία και διαφέρει αναλόγως την εφαρμογή. Ευεργετική θα ήταν η εύρεση ενός μεθοδευμένου τρόπου (ενδεχομένως τρέχοντας έναν εσωτερικό βρόχο βελτιστοποίησης) να πραγματοποιείται η ρύθμιση αυτή.
- Ανάλυση της επίδρασης της ακρίβειας νευρωνικού δικτύου σε σχέση με τη βάση δεδομένων. Η βάση δεδομένων που χρησιμοποιήθηκε δομήθηκε από τυχαίες περιπτώσεις. Πιθανώς να υπάρξει όφελος στην ακρίβεια του δικτύου, συνθέτοντας

βάση δεδομένων με την τεχνική DoE - Design of Experiments, επιλέγοντας δηλαδή τα δείγματα εκπαίδευσης με στατιστικές διαδικασίες/μεθόδους.

# Παράρτημα Α΄

## Scripts

### A΄.0.1 SwPw.bat

Το script SwPw.bat δημιουργήθηκε με σκοπό την αλληπάλληλη κλήση των λογισμικών Solidworks και Pointwise σε λειτουργικό σύστημα Windows. Πριν καλέσει το λογισμικό Solidworks για την παραγωγή του αρχείου γεωμετρίας, διαγράφει πιθανά αρχεία γεωμετρίας και πλέγματος από προηγούμενη περίπτωση. Αφού τερματιστεί το macro του Solidworks, με την εντολή if, το script ελέγχει εάν παρήχθη αρχείο γεωμετρίας. Εάν το παραπάνω δεν παραχθεί από το Solidworks (πιθανώς σε εσφαλμένες τιμές των μεταβλητών σχεδιασμού), το Pointwise δεν θα μπορέσει να παράξει αρχείο πλέγματος, και η εφαρμογή δεν θα τερματιστεί. Για να μην δημιουργεί πρόβλημα μια τέτοια περίπτωση, είναι αναγκαίος ο έλεγχος με την εντολή if. Χρησιμοποιώντας το παρακάτω script, ο χρήστης αρκεί να τοποθετήσει δίπλα από το όρισμα κάθε μεταβλητής το path που αντιστοιχεί ή το όνομα του εκάστοτε αρχείου.

```
SET WORKDIR= Path με τον φάκελο αρχείων
SET SLDWPATH= Path εγκατάστασης του Solidworks
SET PWPATH= Path εγκατάστασης του Pointwise
SET SLDWXPORT= Όνομα αρχείου *.x_t
SET PWXPORT= Όνομα αρχείου *.su2
SET SLDWMACRO= Όνομα του macro του Solidworks *.swp
SET PWMACRO= Όνομα του script του Pointwise *.glf
```

```
del "%WORKDIR%\%SLDWXPORT%" /f /q
del "%WORKDIR%\%PWXPORT%" /f /q
"%SLDWPATH%\SLDWORKS.exe" /m "%WORKDIR%\%SLDWMACRO%"
```

```
if not exist %SLDWXPORT% goto tkill
```

```
"%PWPATH%\Pointwise.exe" "%WORKDIR%\%PWMACRO%"
```

```
:tkill
taskkill /f /im "Pointwise.exe"
taskkill /f /im "SLDWORKS.exe"
```



## A'.0.2 Preprocessor.py

Η λειτουργία του script σε γλώσσα python Preprocessor.py είναι η ανάγνωση του αρχείου 'task.dat' του λογισμικού EASY και η παραγωγή του αρχείου 'equations.txt', με το οποίο το λογισμικό Solidworks δημιουργεί τη γεωμετρία και παράγει το αντίστοιχο αρχείο. Όπως έχει παρουσιαστεί στην εργασία, το αρχείο 'equations.txt' έχει μια συγκεκριμένη μορφή. Αρχικά, αναφέρει τα ονόματα των μεταβλητών, τις τιμές τους και τις μονάδες τους και ύστερα αναφέρει τον τρόπο σύνδεσης των μεταβλητών με διαστάσεις της γεωμετρίας. Για να παράγεται αρχείο 'equations.txt' με την επιθυμητή μορφή, το script Preprocessor.py απαιτεί την ανάγνωση ενός αρχείου που δημιουργείται από το χρήστη 'eq\_form.txt'. Το αρχείο αυτό ωφείλει να έχει την εξής μορφή:

```
eq_form.txt
"a1" =
"a2" = "ΟΝΟΜΑ ΜΕΤΑΒΛΗΤΗΣ" =
"r1" =
"r2" =
"l1" =
"D1@3DSketch4"= "a1"
"D2@3DSketch4"= "a2"
"D4@3DSketch4"= "r1"
"D5@3DSketch4"= "r2"
"D3@3DSketch4"= "l1"
deg
deg
m          ΜΟΝΑΔΕΣ
m
m
```

Σχήμα A'.1: Μορφή του βοηθητικού αρχείου 'eq\_form.txt'

Τον τρόπο σύνδεσης των μεταβλητών με τις διαστάσεις, ο χρήστης μπορεί να τον λάβει από το πρώτο αρχείο 'equations.txt', που θα δημιουργηθεί κατά τη σχεδίαση της πρώτης γεωμετρίας.

```

f = open("task.dat", "r")
file = f.read().splitlines()
f.close()
N = int(file[0])
var = [None]*N
k=0
for line in file[1:]:
    var[k] = line
    k+=1

f = open("eq_form.txt" , "r")
file = f.read().splitlines()
f.close()

names = file[0:N]
text = file[N:2*N]
units = file [2*N:3*N]

new_file = [None]*2*N
for k in range(N):
    new_file[k] = str(names[k]) + ' ' + str(var[k]) + str(units[k])

for k in range(N):
    new_file[N+k] = str(text[k])

f = open("equations.txt" , "w")
for line in new_file:
    f.write(line + "\n")
f.close()

```

### A'.0.3 main.bat

Το script 'main.bat' καλείται από το λογισμικό EASY και η λειτουργία του είναι να καλεί τα επιμέρους scripts. Γίνεται χρήση αλληπάληλων scripts, καθώς τα scripts της μορφής \*.bat κάνουν τη χρήση εμφωλευμένων εντολών δύσκλη.

```
SET PYTHONPATH=Path εγκατάστασης της Python
SET WORKDIR=Path φακέλου βελτιστοποίησης
SET PREPROCESSOR=Όνομα του script Preprocessor
SET SAFETY=Όνομα του script Safety
SET POSTPROCESSOR=Όνομα του script Postprocessor
SET RESULTFILE=task.res
```

```
"%PYTHONPATH%" "%PREPROCESSOR%"
call SwPw.bat
call remote.bat
del "%RESULTFILE%"
"%PYTHONPATH%" "%SAFETY%"
"%PYTHONPATH%" "%POSTPROCESSOR%"
```

## A'.0.4 remote.bat (Χρησιμοποιώντας MobaXTerm)

Για την απομακρυσμένη σύνδεση του προσωπικού υπολογιστή του χρήστη με τις συστοιχίες της ΜΠΥΡΒ/ΕΜΠ, χρησιμοποιήθηκε το λογισμικό 'MobaXTerm' με τη λειτουργία ssh.

```
SET WORKDIR=Path φακέλου αρχείων βελτιστοποίησης
SET RESFILE=Όνομα αρχείου αποτελεσμάτων του PUMA *.conv

del "%WORKDIR%\%RESFILE%"

SET GPU0=IP των GPU
SET WORKDIR=Path φακέλου αρχείων βελτιστοποίησης στις GPU
SET FILE=Όνομα αρχείου *.su2
SET USERNAME=username στις GPU
SET ORGDIR=Path φακέλου μετατροπής *.su2 σε LTT στις GPU
SET GPUX=Αριθμός της GPU που εκτελείται το PUMA
SET OPTIMDIR=Path φακέλου που εκτελείται το PUMA (flow folder)
SET HOMEDIR=/home/%USERNAME%
SET MESHDIR=Path φακέλου πλέγματος το PUMA (mesh folder)
SET HYB=Όνομα αρχείου *.hyb
SET NOD=Όνομα αρχείου *.nod

MobaXterm.exe -exec "ssh -tt %USERNAME%@%GPU0% 'yes | rm
%ORGDIR%/%FILE%' " -exitwhendone

MobaXterm.exe -exec "ssh -tt %USERNAME%@%GPU0% 'rm %MESHDIR%/%NOD% ; rm
%MESHDIR%/%HYB%' " -exitwhendone

MobaXterm.exe -exec "cd %WORKDIR% && scp %FILE%
%USERNAME%@%GPU0%:%ORGDIR%" -exitwhendone

call eval.bat

MobaXterm.exe -exec "scp %USERNAME%@%GPU0%:%OPTIMDIR%/%RESFILE%
%WORKDIR%" -exitwhendone
```

## A'.0.5 eval.bat (Χρησιμοποιώντας MobaXTerm)

SET USERNAME=username στις GPU

SET GPU0=IP των GPU

SET GPUX=Αριθμός GPU που τρέχει το PUMA

SET EVALEXEPATH=Path για το script task.sh

```
MobaXterm.exe -exec "ssh -tt %USERNAME%@%GPU0% 'ssh -tt %GPUX% \  
%EVALEXEPATH% '" -exitwhendone
```

## A'.0.6 dnn.py (Χρησιμοποιώντας tensorflow)

Το script 'dnn.py' περιέχει τον κώδικα που χρησιμοποιήθηκε για τη δημιουργία των νευρωνικών δικτύων. Στην ενότητα # Read, Normalize and store data#, ο χρήστης πρέπει να γράψει κώδικα που να διαβάζει τη βάση δεδομένων, να κανονικοποιεί τα μεγέθη και να τα αποθηκεύει σε μητρώα. Στην ενότητα # Store input and output data# ο χρήστης πρέπει να γράψει κώδικα που θα διαχωρίζει τα μητρώα εισόδου και εξόδου για τα δείγματα εκπαίδευσης και για τα δείγματα πιστοποίησης αντίστοιχα. Έπειτα το script καλεί τη συνάρτηση 'train', η οποία εκπαιδεύει το δίκτυο που αναφέρεται εντός αυτής. Στις πρώτες γραμμές της συνάρτησης 'train', ο χρήστης πρέπει να καθορίσει τη μία διάσταση των μητρώων εξόδου (εν προκειμένω 1225). Στην ενότητα # DNN MODEL #, ο χρήστης πρέπει να καθορίσει πόσα κρυμμένα επίπεδα θα έχει η αρχιτεκτονική του δικτύου και πόσους νευρώνες το κάθε ένα. Πρέπει επίσης να διαλέξει συναρτήσεις ενεργοποίησης. Στη συνέχεια, ο χρήστης πρέπει να καθορίσει τη συνάρτηση κόστους (loss), τον αλγόριθμο εκπαίδευσης με το ρυθμό μάθησης και παρακάτω τον αριθμό των εποχών (epoch) και των ομάδων (batchN).

```

import os
import math
import time
import shutil
import sys
import random
import zipfile
import tensorflow as tf
import numpy as np
import random
import sys

def train(input,output,input_val,output_val):
    with tf.Graph().as_default():
        x_db = input
        x_val = input_val
        y_db = np.reshape(output , (output .shape[0],1225))
        y_val = np.reshape(output_val, (output_val.shape[0],1225))
        x = tf.placeholder(shape =
[None,input.shape[1]],dtype=tf.float64)
        y = tf.placeholder(shape = [None,
],dtype=tf.float64)

        print("#x_db = ",x_db.shape)
        print("#y_db = ",y_db.shape)
        print("#x_val = ",x_val.shape)
        print("#y_val = ",y_val.shape)

        #DNN MODEL#
        h = tf.layers.dense(x,50,activation='relu',use_bias=True)
        h = tf.layers.dense(h,75,activation='relu',use_bias=True)
        h = tf.layers.dense(h,113,activation='relu',use_bias=True)
        h = tf.layers.dense(h,170,activation='relu',use_bias=True)
        h = tf.layers.dense(h,255,activation='relu',use_bias=True)
        h = tf.layers.dense(h,383,activation='relu',use_bias=True)
        h = tf.layers.dense(h,575,activation='relu',use_bias=True)
        h = tf.layers.dense(h,863,activation='relu',use_bias=True)
        o = tf.layers.dense(h,1225,activation='tanh',use_bias=True)

        loss = tf.losses.absolute_difference(o,y)
        optimizer = tf.train.AdamOptimizer(learning_rate=0.002)
        train_op = optimizer.minimize(loss)
        variables = tf.global_variables()
        saver = tf.train.Saver(variables)
        init = tf.global_variables_initializer()
        config = tf.ConfigProto()
        config.gpu_options.allow_growth = True
        sess = tf.Session(config=config)
        sess.run(init)
        save_rest = tf.train.Saver(variables)
        ckpt = tf.train.get_checkpoint_state("checkpoint/")

```

```

if ckpt is not None:
    try:
        save_rest.restore(sess,ckpt.model_checkpoint_path)
    except:
        print("there was a problem using variables in checkpoint, random
init will be used instead")
    nepoch = 15001
    batchN = 1
    step    = int(int(x_db.shape[0])/batchN)

for iters in range(nepoch):
    for i in range(batchN):
        x_ = x_db[i*step:(i+1)*step,:]
        y_ = y_db[i*step:(i+1)*step,:]
        sess.run(train_op, feed_dict={x: x_, y: y_})
        if iters % 1000 == 0:
            saver.save(sess,'checkpoint/save')
            mse = loss.eval(session=sess,feed_dict={x: x_ ,y: y_})
            pred = sess.run(o, feed_dict={x: x_})
            np.savetxt("pred",pred)
            sys.stdout.flush()
            predv = sess.run(o, feed_dict={x: x_val})
            predError = loss.eval(session=sess,feed_dict={x: x_val, y:
y_val})

            print(iters, "\tMSE:", mse , "\t" , predError )

return predv

#####STARTING THE PROGRAM MAIN#####

#####
# Read, normalize and store data #
#####

#####
# Store input and output data #
#####

# Call train function #
predicted = train(dbInput,dbOutput,valInput,valOutput)

```



## Βιβλιογραφία

- [1] Wikipedia: *Solidworks*, Accessed 2.9.2020.  
<https://en.wikipedia.org/wiki/SolidWorks>.
- [2] Solidworks: *Parasolid files*, Accessed 4.9.2020. [https://help.solidworks.com/2014/english/SolidWorks/sldworks/c\\_parasolid\\_files.htm](https://help.solidworks.com/2014/english/SolidWorks/sldworks/c_parasolid_files.htm).
- [3] Wikipedia: *Parasolid*, Accessed 4.9.2020.  
<https://en.wikipedia.org/wiki/Parasolid>.
- [4] CFD-Wiki: *Pointwise*, Accessed 2.9.2020.  
<https://www.cfd-online.com/Wiki/Pointwise>.
- [5] Trompoukis, X., Tsiakas, K.T., Nejad, M.G., Asouti, V.G., and Giannakoglou, K.C.: *The continuous adjoint method on graphics processing units for compressible flows*. OPT-i 2014 - 1st International Conference on Engineering and Applied Sciences Optimization, Proceedings, pages 2026–2038, January 2014.
- [6] Asouti, V., Trompoukis, X., Kampolis, I., and Giannakoglou, K.C.: *Unsteady cfd computations using vertex[U+2010]centered finite volumes for unstructured grids on graphics processing units*. International Journal for Numerical Methods in Fluids, 67:232 – 246, September 2011. 10.1002/fld.2352.
- [7] Trompoukis, X.S., Asouti, V.G., Kampolis, I.C., and Giannakoglou, K.C.: *CUDA Implementation of Vertex-Centered, Finite Volume CFD Methods on Unstructured Grids with Flow Control Applications*, pages 207–223. December 2012, ISBN 9780123859631.
- [8] Engineering-Toolbox: *Reynolds number*, 2003 - Accessed 28.3.2020.  
[https://www.engineeringtoolbox.com/reynolds-number-d\\_237.html](https://www.engineeringtoolbox.com/reynolds-number-d_237.html).
- [9] Engineering-Toolbox: *Typical air duct velocities*, 2003 - Accessed 30.3.2020.  
[https://www.engineeringtoolbox.com/amp/flow-velocity-air-ducts-d\\_388.html](https://www.engineeringtoolbox.com/amp/flow-velocity-air-ducts-d_388.html).
- [10] Spalart, P. and Allmaras, S.: *A one-equation turbulence model for aerodynamic flows*. AIAA, 439, January 1992. 10.2514/6.1992-439.

- [11] NASA: *The spalart-allmaras turbulence model*, Accessed 6.9.2020.  
<https://turbmodels.larc.nasa.gov/spalart.html>.
- [12] Menter, F. R.: *Two-equation eddy-viscosity turbulence models for engineering applications*. 1994. <https://doi.org/10.2514/3.12149>.
- [13] CFD-Wiki: *Turbulence intensity*, Accessed 20.4.2020.  
[https://www.cfd-online.com/Wiki/Turbulence\\_intensity](https://www.cfd-online.com/Wiki/Turbulence_intensity).
- [14] Κ.Χ. Γιαννάκογλου, Καθηγητής Ε.Μ.Π.: *Μέθοδοι Βελτιστοποίησης στην Αεροδυναμική*, 4η Έκδοση. Αθήνα, 2006.
- [15] Kapsoulis, D. H.: *Low-Cost Metamodel-Assisted Evolutionary Algorithms with Application in Shape Optimization in Fluid Dynamics*. PhD thesis, School of Mechanical Engineering, NTUA, Athens, 2019.
- [16] Γιώτης, Α.: *Χρήση εξελικτικών τεχνικών, υπολογιστικής ευφύιας και μεθόδων υπολογιστικής ρευστομηχανικής στη βελτιστοποίηση - αντίστροφη σχεδίαση περρωσέων στροβιλομηχανών, μέσω παράλληλης επεξεργασίας*. Διδακτορική διατριβή, Εργαστήριο Θερμικών Στροβιλομηχανών, Ε.Μ.Π., Αθήνα, 2003.
- [17] Kampolis, I.C.: *Parallel, multilevel algorithms for the aerodynamic optimization in turbomachines*. PhD thesis, NTUA, 2009.
- [18] Karakasis, M.K.: *Hierarchical, distributed evolutionary algorithms and computational intelligence in aerodynamic shape optimization, on multiprocessing systems*. PhD thesis, NTUA, 2006.
- [19] Karakasis, M.K., Giotis, A.P., and Giannakoglou, K.C.: *Inexact information aided, low-cost, distributed genetic algorithms for aerodynamic shape optimization*. International Journal for Numerical Methods in Fluids, 43(10 [U+2010] 11):1149–1166, 2003.  
<https://onlinelibrary.wiley.com/doi/abs/10.1002/flid.575>,  
<https://doi.org/10.1002/flid.575>.
- [20] Kapsoulis, D., Tsiakas, K., Trompoukis, X., Asouti, V., and Giannakoglou, K.C.: *Evolutionary multi-objective optimization assisted by metamodels, kernel pca and multi-criteria decision making techniques with applications in aerodynamics*. Applied Soft Computing, 64:1 – 13, 2018, ISSN 1568-4946.  
<http://www.sciencedirect.com/science/article/pii/S1568494617307093>,  
<https://doi.org/10.1016/j.asoc.2017.11.046>.
- [21] Suresh, S., Omkar, S.N., Mani, V., and Guru Prakash, T.N.: *Lift coefficient prediction at high angle of attack using recurrent neural network*. Aerospace Science and Technology, 7(8):595 – 602, 2003, ISSN 1270-9638. <http://www.sciencedirect.com/science/article/pii/S1270963803000531>,  
[https://doi.org/10.1016/S1270-9638\(03\)00053-1](https://doi.org/10.1016/S1270-9638(03)00053-1).