



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

**ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ
ΣΠΟΥΔΩΝ ΥΠΟΛΟΓΙΣΤΙΚΗΣ ΜΗΧΑΝΙΚΗΣ**

ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Προγραμματισμός μεθόδου συναρτήσεων ακτινικής βάσης
(RBF) για τη μετατόπιση υπολογιστικών πλεγμάτων σε
κάρτες γραφικών**

Κωνσταντίνος Κολοβός

ΕΠΙΒΛΕΠΩΝ:

Κυριάκος Χ. Γιαννάκογλου, Καθηγητής ΕΜΠ

**ΣΧΟΛΗ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΕΜΠ
ΤΟΜΕΑΣ ΡΕΥΣΤΩΝ
ΜΟΝΑΔΑ ΠΑΡΑΛΛΗΛΗΣ ΥΠΟΛΟΓΙΣΤΙΚΗΣ
ΡΕΥΣΤΟΔΥΝΑΜΙΚΗΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ**

**ΑΘΗΝΑ
ΟΚΤΩΒΡΙΟΣ, 2015**

ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Προγραμματισμός μεθόδου συναρτήσεων ακτινικής βάσης
(RBF) για τη μετατόπιση υπολογιστικών πλεγμάτων σε
κάρτες γραφικών**

Κωνσταντίνος Κολοβός

ΕΠΙΒΛΕΠΩΝ:

Κυριάκος Χ. Γιαννάκογλου, Καθηγητής ΕΜΠ

Ευχαριστίες

Στο σημείο αυτό θα ήθελα να ευχαριστήσω όσους με υποστήριξαν κατά την διάρκεια των μεταπτυχιακών σπουδών μου και όσους συνέβαλαν στην εκπόνηση της παρούσας μεταπτυχιακής εργασίας. Ιδιαίτερα θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κ. Κυριάκο Γιαννάκογλου για την ευκαιρία που μου έδωσε να ασχοληθώ με το συγκεκριμένο αντικείμενο. Η εργατικότητα, η σοβαρότητα και η υπευθυνότητα την οποία επιδεικνύει ο κ. Γιαννάκογλου στην αντιμετώπιση των διδακτικών και ερευνητικών του υποχρεώσεων αποτέλεσε για μένα σπουδαίο μάθημα. Οφείλω λοιπόν τις πρώτες μου και πιο θερμές ευχαριστίες να τις εκφράσω σ' αυτόν.

Ευχαριστώ τους καθηγητές μου που κατά τη διάρκεια των μεταπτυχιακών σπουδών, μου μετέδωσαν πολύτιμες συμβουλές και γνώσεις, διαμόρφωσαν τον τρόπο σκέψης και αναζήτησης. Ιδιαίτερα θα ήθελα να ευχαριστήσω την Δρ. Βαρβάρα Ασούτη, η οποία ήταν πάντα πρόθυμη να με καθοδηγήσει κατά τη διάρκεια της ανάπτυξης του κώδικα. Θα ήθελα επίσης να ευχαριστήσω θερμά όλα τα μέλη της Μονάδας Παράλληλης Υπολογιστικής Ρευστοδυναμικής & Βελτιστοποίησης του Εργαστηρίου Θερμικών Στροβιλομηχανών της Σχολής Μηχανολόγων Μηχανικών του ΕΜΠ για τον χρόνο που μου αφιέρωσαν και για τις εύστοχες παρατηρήσεις. Ένα μεγάλο ευχαριστώ, θέλω να πω στην οικογένειά μου για την ηθική συμπαράσταση και στήριξη που μου παρείχε κατά τη διάρκεια των σπουδών μου.

Περίληψη

Η μετακίνηση ενός υπολογιστικού πλέγματος, μαζί ή χωρίς τη μετακίνηση των ορίων του, είναι πολύ χρήσιμη στις εφαρμογές Υπολογιστικής Ρευστοδυναμικής. Στα προβλήματα βελτιστοποίησης μορφής η μετατόπιση του υπολογιστικού πλέγματος είναι απαραίτητη, είτε στην περίπτωση όπου χρησιμοποιείται παραμετροποίηση της προς σχεδιασμό μορφής, η οποία τροποποιεί σε κάθε κύκλο της βελτιστοποίησης τα όρια του υπολογιστικού χωρίου και αυτό που χρειάζεται είναι να προσαρμοστεί κατάλληλα το εσωτερικό πλέγμα, είτε στην παραλλαγή που ενιαίος μηχανισμός αναλαμβάνει, ελλείπει παραμετροποίησης, την ταυτόχρονη και συμβατή μετακίνηση όλων των κόμβων ενός υπολογιστικού πλέγματος, συμπεριλαμβανομένων των ορίων. Από την άλλη πλευρά, στην αεροελαστικότητα, τα τοιχώματα ενός στερεού σώματος, λ.χ. μιας πτέρυγας, μετακινούνται με το χρόνο και σε κάθε χρονικό βήμα, πρέπει να αναπροσαρμοστεί το πλέγμα σε αυτά. Σε όλες τις παραπάνω περιπτώσεις, η επαναπλεγματοποίηση είναι πολύ χρονοβόρα και ειδικά σε τριδιάστατες γεωμετρίες.

Η παρούσα μεταπτυχιακή εργασία αφορά τεχνικές μετακίνησης πλεγμάτων σε διδιάστατες και τριδιάστατες γεωμετρίες, αναπτύσσοντας και εφαρμόζοντας δίκτυα συναρτήσεων ακτινικής βάσης (RBF). Ο κώδικας που αναπτύχθηκε αποτελείται από δύο τμήματα. Το βασικό τμήμα αποτελεί τον κώδικα που αναπτύχθηκε σε γλώσσα προγραμματισμού C/C++ για την εκτέλεση σε CPU. Το άλλο τμήμα προγραμματίστηκε σε γλώσσα προγραμματισμού CUDA C για την υλοποίηση παράλληλα σε Μονάδα Επεξεργασίας Γραφικών (GPU) επειδή σε μεγάλα προβλήματα βιομηχανικής κλίμακας, τα δίκτυα RBF έχουν μεγάλο υπολογιστικό κόστος. Επίσης στην εργασία χρησιμοποιήθηκε το υπολογιστικό πακέτο CULA(CUDA LINEAR ALGEBRA) για την επίλυση των γραμμικών συστημάτων. Στην εργασία παρουσιάζονται και χρησιμοποιούνται μετρικές ποιότητας μη-δομημένων τριγωνικών και μη-δομημένων τετραεδρικών πλεγμάτων, με τις οποίες πραγματοποιείται έλεγχος της ποιότητας κάθε προκύπτοντος παραμορφωμένου πλέγματος. Ο κώδικας δοκιμάστηκε σε δύσκολες γεωμετρίες με διαφορετικές συναρτήσεις βάσεις. Η παρούσα μεταπτυχιακή εργασία επικεντρώνεται στην υλοποίηση, προγραμματισμό και δοκιμή των δικτύων RBF και στην εκτέλεση τους σε μονάδα επεξεργασίας γραφικών με σκοπό την επιτάχυνση της φάσης της εκπαίδευσής τους.

NATIONAL TECHNICAL UNIVERSITY OF ATHENS
JOINT POSTGRADUATE PROGRAM: COMPUTATIONAL MECHANICS
SCHOOL OF MECHANICAL ENGINEERING
FLUIDS DEPARTMENT
LABORATORY OF THERMAL TURBOMACHINES
PARALLEL CFD & OPTIMIZATION UNIT

Graduate Thesis

Programming of a Method for the Displacement of Computational Mesh using Radial Basis Functions (RBF) on Graphics Processing Unit

Konstantinos Kolovos

Advisor: Kyriakos C. Giannakoglou, Professor NTUA

Athens, October 2015

Abstract

Mesh displacement, where the term “mesh” may refer either to internal nodes or may include the boundary nodes, is very useful in Computational Fluid Dynamic. In shape optimization problems, this technique is necessary either if a shape parameterization tool is used, in which cases the boundaries of the computational mesh are modified in each cycle of the optimization loop and a mechanism propagating the surface displacements to the interior mesh nodes must be devised, or in parameterization-free methods, where the same mechanism undertakes the displacement of both internal and boundary nodes. Furthermore, in aeroelasticity, the walls of a solid body, e.g. a wing, oscillate and therefore, in every time step, the mesh should be readjusted to the new boundaries. In all the above cases, remeshing is very time-consuming, especially in three dimensional geometries.

The present graduate thesis focuses on mesh displacement techniques in two-dimensional and three-dimensional geometries by developing and applying Radial Basis Function (RBF) networks. The code that was developed consists of two parts. The main part is the part of the code that was developed in C/C++ programming language to run on CPU. The other part was programmed in CUDA C programming language for the parallel implementation in Graphics Processing Unit (GPU), because of the high computational cost of RBF networks in large scale industrial problems. In addition, the CULA - CUDA LINEAR ALGEBRA computational library was used for solving linear systems. In this thesis are presented and used unstructured triangular and tetrahedral mesh quality metrics, according to which, the displacement mesh quality was tested. The code has been tested in difficult geometries with different basis functions. The present thesis focuses on the implementation, programming and test of RBF networks and the execution of them on GPU, aiming at the acceleration of RBF network training phase.

Ακρωνύμια

ΕΜΠ	Εθνικό Μετσόβιο Πολυτεχνείο
ΜΠΥΡ&Β	Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής & Βελτιστοποίησης
ΜΔΕ	Μερικές Διαφορικές Εξισώσεις
ΥΡΔ	Υπολογιστική Ρευστοδυναμική
CFD	Computational Fluid Dynamics
NTUA	National Technical University of Athens
RBF	Radial Basis Functions
GPGPU	General Purpose computing on Graphics Processing Units
CUDA	Compute Unified Device Architecture
CULA	CUDA Linear Algebra
CAD	Computer Aided Design

Περιεχόμενα

1	Εισαγωγή	15
1.1	Η βελτιστοποίηση στην αεροδυναμική	15
1.2	Ανάγκη για παραμόρφωση πλέγματος	15
1.3	Άλλες μέθοδοι παραμόρφωσης πλέγματος	16
2	Δίκτυα Συναρτήσεων Ακτινικής Βάσης	18
2.1	Μετατόπιση πλέγματος με δίκτυα (RBF)	18
2.2	Συναρτήσεις Ακτινικής Βάσης (RBF)	21
2.3	Τα Δίκτυα RBF ως εργαλείο μετακίνησης πλέγματος	24
2.3.1	Χρησιμότητα των δικτύων RBF	24
2.3.2	Εκπαίδευση και χρήση του δικτύου RBF	25
3	Η GPU και η τεχνολογία CUDA	32
3.1	Η αρχιτεκτονική CUDA	32
3.1.1	Η GPU– Αρχιτεκτονική Single Instruction – Multiple Threads	32
3.1.2	Οργάνωση δεδομένων και νημάτων	39
3.1.3	Τύποι μνήμης	41
3.1.4	Ασύγχρονη και ταυτόχρονη εκτέλεση	45
3.1.5	Πληρότητα των πολυεπεξεργαστών και τεχνικά χαρακτηριστικά	47
3.2	CUDA/C : Μία επέκταση των C/C++	49
3.2.1	Εισαγωγή στην γλώσσα προγραμματισμού CUDA/C	49
3.2.2	Ειδικοί ορισμοί συναρτήσεων	49
3.2.3	Ο kernel, οι ειδικές μεταβλητές του και οι συναρτήσεις συσκευής	50
3.2.4	Καλές πρακτικές και μέγιστη απόδοση	52
3.3	Η εργαλειοθήκη CUDA Toolkit	53
3.3.1	Ο μεταγλωττιστής nvcc	54
3.3.2	CULA Περιγραφή Λογισμικού	54
4	Μέθοδοι Επίλυσης	57
4.1	Παραγοντοποίηση CHOLESKY	57
5	Εφαρμογές σε 2D πλέγμα γύρω από γεωμετρία φύλλου πλατάνου	58
5.1	Περιγραφή του προβλήματος	58
5.2	Παρουσίαση αποτελεσμάτων	60
5.2.1	Αποτελέσματα με τη <i>Wendland C2</i> , $R = 2$	60
5.2.2	Αποτελέσματα με τη <i>Wendland C2</i> , $R = 3$	62
5.2.3	Αποτελέσματα με τη <i>Wendland C2</i> , $R = 9$	66
5.2.4	Αποτελέσματα με τη συνάρτηση ολικής επιρροής Gauss	73
5.2.5	Αποτελέσματα με τη συνάρτηση ολικής επιρροής Inverse Multiquadric Biharmonics	77
5.3	Σχόλια	82
6	Εφαρμογές σε πλέγμα γύρω από αεροτομή	84
6.1	Περιγραφή του προβλήματος	84
6.2	Παρουσίαση αποτελεσμάτων	85
6.2.1	Αποτελέσματα με τη <i>Wendland C2</i> , $R = 2$	85
6.3	Σχόλια	89

7	Εφαρμογές σε 3D πλέγμα γύρω από αεροσκάφος	90
7.1	Περιγραφή του προβλήματος.....	90
7.2	Εφαρμογές με τη <i>Wendland C2</i>	92
7.2.1	Αποτελέσματα με τη <i>Wendland C2</i> και $R=21$	92
7.2.2	Αποτελέσματα με τη <i>Wendland C2</i> και $R=40$	94
7.3	Σχόλια	96
8	Αποτελέσματα και συγκριτικές επιδόσεις.....	98
8.1	Σύγκριση ανάμεσα στο πακέτο CULA και στη συνάρτηση CUDA kernel	98
9	Ανακεφαλαίωση και Συμπεράσματα.....	100
9.1	Συμπεράσματα για τα δίκτυα RBF	101
9.2	Συμπεράσματα για τη βιβλιοθήκη CULA	102
10	Βιβλιογραφία.....	104

1 Εισαγωγή

1.1 Η βελτιστοποίηση στην αεροδυναμική

Τον τελευταίο καιρό υπάρχει έντονο ενδιαφέρον στον τομέα της βελτιστοποίησης για το σχεδιασμό αεροδυναμικών ή υδροδυναμικών μορφών (πτερυγίων, πτερύγων, αεροτομών, μορφολογία σχήματος αυτοκινήτων κλπ) με καλύτερη απόδοση σε κάποιο εύρος λειτουργίας [1], [2], [3], ή επιδιώκοντας ικανοποιητική απόδοση σε ένα εύρος λειτουργίας [4]. Το επόμενο, δηλαδή, βήμα όταν είναι υπολογιστικά εφικτή η ανάλυση μιας μηχανολογικής συνιστώσας, είναι η βελτιστοποίησή της βάσει προδιαγραφών και απαιτήσεων. Η βελτιστοποίηση των μορφών αυτών στην αεροδυναμική ή την υδροδυναμική, επιτυγχάνεται μέσω μεθόδων βελτιστοποίησης [5] και εμπλέκει μεθόδους πρόλεξης πεδίων ροής, δηλαδή μεθόδους Υπολογιστικής Ρευστοδυναμικής (ΥΡΔ, Computational Fluid Dynamics, CFD).

Η μέθοδος της βελτιστοποίησης είναι το εργαλείο το οποίο ανιχνεύει τον χώρο των υποψήφιων λύσεων και οδηγεί στην επιλογή της βέλτιστης από όλες τις υποψήφιες λύσεις. Για αυτόν το λόγο, απαιτείται ο καθορισμός ενός ή περισσοτέρων στόχων που φέρονται στη μορφή συναρτήσεων στόχου (objective functions) οι οποίες πρέπει να ελαχιστοποιηθούν ή να μεγιστοποιηθούν, των περιορισμών (constraints) ανισότητας ή ανισότητας του προβλήματος καθώς και των (ελεύθερων) μεταβλητών σχεδιασμού (design variables), η επιλογή συνόλου τιμών των οποίων οδηγεί στη βέλτιστη λύση ενός προβλήματος. Σαφώς, κατά την ανίχνευση του χώρου των υποψήφιων λύσεων, την επιλογή δηλαδή των βέλτιστων τιμών μεταβλητών σχεδιασμού, απαιτείται υποστήριξη και από ένα υπολογιστικό εργαλείο. Αυτό ονομάζεται λογισμικό αξιολόγησης, που βαθμολογεί και αξιολογεί ως προς τους στόχους που τέθηκαν τις υποψήφιες λύσεις. Ένα ενδεικτικό παράδειγμα λογισμικού αξιολόγησης στην αεροδυναμική βελτιστοποίηση είναι ο κώδικας επίλυσης των εξισώσεων ροής γύρω λ.χ. από μία πτέρυγα ενός αεροσκάφους. Ένα είδος βελτιστοποίησης που χρησιμοποιείται πολύ συχνά είναι η βελτιστοποίηση μορφής (shape optimization), όπου το πρόβλημα ανάγεται στον προσδιορισμό της μορφής, λ.χ. μιας αεροτομής [6] που θα παρουσιάζει μέγιστο συντελεστή άνωσης CL υπό κάποιους περιορισμούς.

1.2 Ανάγκη για παραμόρφωση πλέγματος

Αντικείμενο μιας ευρείας περιοχής της αεροδυναμικής βελτιστοποίησης είναι η εύρεση βέλτιστων μορφών με χρήση λογισμικού αξιολόγησης της γεωμετρίας με σκοπό την επίλυση προβλημάτων ενός στόχου με μία αντικειμενική συνάρτηση (π.χ συνάρτηση υπολογισμού του συντελεστή άνωσης της αεροτομής) ή την αντιμετώπιση προβλημάτων πολλών στόχων με περισσότερες από μία αντικειμενικές συναρτήσεις. Για την εύρεση της βέλτιστης λύσης πρέπει, πρώτα, να αξιολογηθούν οι υποψήφιες λύσεις με τη χρήση ενός επιλύτη της ροής (λ.χ η ανάλυση μίας υποψήφιας αεροτομής γίνεται με λογισμικό όπως είναι οι κώδικες Navier-Stokes) [48]. Επομένως, επειδή τα προβλήματα αυτά λύνονται με τη χρήση Ηλεκτρονικού Υπολογιστή, χρήζουν χωρικής διακριτοποίησης, απαιτείται δηλαδή η γένεση κατάλληλου υπολογιστικού πλέγματος [7]. Η διαδικασία της βελτιστοποίησης, όμως, δεν πραγματοποιείται σε ένα βήμα, αλλά αντιθέτως είναι επαναληπτική, γεγονός που σημαίνει ότι σε κάθε κύκλο βελτιστοποίησης, όποια μέθοδος και αν χρησιμοποιείται, τροποποιείται η γεωμετρία της προς βελτιστοποίηση μορφής. Ως εκ τούτου πρέπει να προσαρμοστεί το πλέγμα στη νέα αυτή γεωμετρία ώστε να υποστηρίξει τον επιλύτη στο νέο κύκλο βελτιστοποίησης. Πολλές φορές, επιλέγεται η βελτιστοποίηση γεωμετριών με χρήση αιτιοκρατικών ή στοχαστικών μεθόδων βελτιστοποίησης.

Από την πλευρά των στοχαστικών μεθόδων οι εξελικτικοί αλγόριθμοι (evolutionary algorithms) [8], οι οποίοι με χρήση αντίστοιχης με τη φύση διαδικασίας προσαρμογής στο περιβάλλον επιλύουν διάφορα προβλήματα. Για το λόγο αυτό χρησιμοποιούν στοχαστικούς μηχανισμούς

εξέλιξης προερχόμενους από τη φύση και βασίζονται στις αρχές εξέλιξης των ειδών όπως αυτές διατυπώθηκαν αρχικά από το Δαρβίνο. Οι εξελικτικοί αλγόριθμοι χειρίζονται πληθυσμούς υποψήφιων λύσεων και όχι μία μεμονωμένη λύση σε κάθε επανάληψη για την αξιολόγηση της γεωμετρίας. Οι εξελικτικοί αλγόριθμοι είναι πιο αποδοτικοί όταν χειρίζονται λίγες μεταβλητές και για αυτό το λόγο χρησιμοποιείται παραμετροποίηση της προς βελτιστοποίηση μορφής. Με αυτόν τον τρόπο, μειώνεται ο αριθμός των άγνωστων και έτσι το λογισμικό που χρησιμοποιείται, λ.χ. ο EASY (Evolutionary Algorithms System) [9] της ΜΠΥΡ&Β/ΕΜΠ, μπορεί να ανταπεξέλθει.

Από την πλευρά των αιτιοκρατικών μεθόδων υπάρχει η συζυγής μέθοδος βελτιστοποίησης η οποία βασίζεται στον υπολογισμό της κλίσης της αντικειμενικής συνάρτησης. Για παράδειγμα το πρόβλημα αντίστροφου σχεδιασμού της αεροτομής, δηλαδή η αναζήτηση εκείνων των γεωμετρικών χαρακτηριστικών της αεροτομής που θα δώσουν προκαθορισμένη κατανομή πίεσης ή ταχύτητας. Επίσης η αναζήτηση της αεροτομής με βέλτιστη άνωση, ελάχιστη οπισθέλκουσα ή συνδυασμό τους για δεδομένες συνθήκες ροής. Οι συζυγείς μέθοδοι βελτιστοποίησης (adjoint – based optimization) [12], [13] είναι μαθηματικά υπολογιστικά εργαλεία υπολογισμού της κλίσης της αντικειμενικής συνάρτησης ως προς τις μεταβλητές σχεδιασμού, εξασφαλίζοντας ταυτόχρονα την ικανοποίηση των βασικών εξισώσεων του προβλήματος (λ.χ. των εξισώσεων Navier Stokes). Σε όλες τις παραπάνω περιπτώσεις, είναι αναγκαία η ενιαία και συμβατή κίνηση του υπολογιστικού πλέγματος αλλά και της προς βελτιστοποίηση μορφής. Επιπροσθέτως, σε προβλήματα αεροελαστικότητας, όπου λαμβάνει χώρα αλληλεπίδραση ρευστού με στερεό σώμα, όπως λ.χ. ένα πτερύγιο μιας ανεμογεννήτριας, της οποίας τα στερεά τοιχώματα μετακινούνται λόγω της χρονικά μεταβαλλόμενης δύναμης που ασκείται από τον αέρα, σε κάθε χρονικό βήμα απαιτείται η προσαρμογή του πλέγματος στο νέο σχήμα του πτερυγίου. Μια ακόμα περίπτωση είναι αυτή σε χωρία ροής με κινούμενα όρια, λ.χ. μία παλλόμενη αεροτομή (pitching airfoil), όπου απαιτείται η προσαρμογή του πλέγματος στη νέα θέση της κάθε χρονική στιγμή.

Επειδή η επαναπλεγματοποίηση σε όλες τις παραπάνω περιπτώσεις και ειδικά σε βιομηχανικές εφαρμογές, με μεγάλα τριδιάστατα πλέγματα, έχει απαγορευτικό κόστος, εφαρμόζεται μία τεχνική παραμόρφωσης πλέγματος που υλοποιείται με συναρτήσεις ακτινικής βάσης (Radial Basis Functions, RBF) [10], [11]. Τα δίκτυα RBF είναι μία τεχνική παρεμβολής η οποία μπορεί να χρησιμοποιηθεί ως «εργαλείο παραμόρφωσης» (δηλαδή, μετακίνησης κάποιων κόμβων του πλέγματος) σε όλες τις προαναφερθείσες περιπτώσεις, αρκεί να είναι γνωστές οι μετατοπίσεις κάποιων άλλων κόμβων της προς βελτιστοποίηση μορφής. Στη μέθοδο βελτιστοποίησης, με συζυγείς μεθόδους [14] υπολογίζονται οι παράγωγοι της αντικειμενικής συνάρτησης ως προς τις κομβικές τιμές των συντεταγμένων των κόμβων πάνω στο σώμα λ.χ., και βάσει του κανόνα της αλυσίδας, είναι επιθυμητό να βρεθούν οι παράγωγοι αυτών των μεταβλητών (πρακτικά συντελεστών) ως προς τις παραμέτρους του δικτύου RBF. Με αυτόν τον τρόπο, η διαδικασία της βελτιστοποίησης επιταχύνεται, κάτι το οποίο είναι απαραίτητο στη σημερινή εποχή μιας και ο ανταγωνισμός είναι μεγάλος.

1.3 Άλλες μέθοδοι παραμόρφωσης πλέγματος

Η ανάγκη αποφυγής της επαναπλεγματοποίησης οδήγησε στην ανάπτυξη πολλών μεθόδων προσαρμογής πλέγματος στη νέα γεωμετρία. Μερικές από αυτές εξασφαλίζουν καλύτερη ποιότητα πλέγματος έναντι άλλων μεθόδων αλλά με μεγαλύτερο υπολογιστικό κόστος ή το ανάποδο. Γενικότερα, η αύξηση της ποιότητας προσαρμογής του υπολογιστικού πλέγματος συνεπάγεται και μεγαλύτερο υπολογιστικό κόστος.

Μία πρώτη κατηγορία μεθόδων παραμόρφωσης πλέγματος είναι η μέθοδος στην οποία το υπολογιστικό πλέγμα αντιμετωπίζεται ως ένα ελαστικό στερεό σώμα [15], [16]. Οι ελαστικές εξισώσεις εκφράζουν ισορροπία των ασκούμενων δυνάμεων και ροπών σε στοιχειώδες τμήμα

του όγκου αυτού. Αυτές οι εξισώσεις εκφράζουν το σχήμα του υπολογιστικού πλέγματος, το οποίο προέκυψε από την μετατόπιση των επιφανειακών κόμβων.

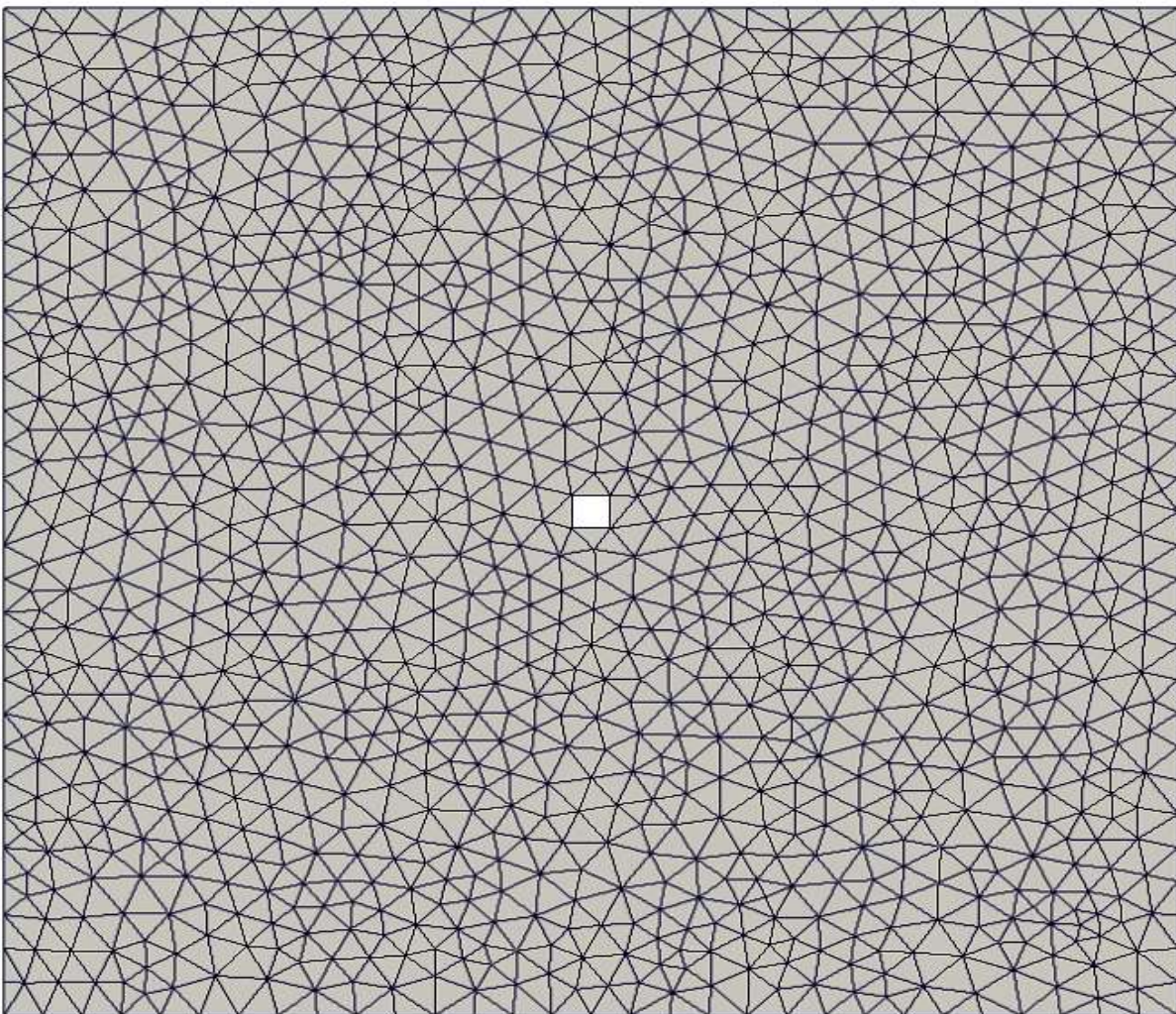
Μία άλλη μέθοδος είναι η μέθοδος των ελατηρίων [17], [18]. Σε αυτήν, το πλέγμα αντιμετωπίζεται ως ένα σύστημα ελατηρίων. Σε αυτή τη μέθοδο θεωρείται ότι η παραμόρφωση ενός στοιχείου του πλέγματος περιορίζεται από γραμμικά ή στρεπτικά ελατήρια, τοποθετημένα πάνω στους κόμβους ή στις ακμές που ενώνουν τους κόμβους του υπολογιστικού πλέγματος. Η μετατόπιση των οριακών κόμβων του πλέγματος επηρεάζει την μετατόπιση των εσωτερικών, η οποία υπολογίζεται από επίλυση των εξισώσεων ισορροπίας δυνάμεων και ροπών στους κόμβους του πλέγματος. Επίσης, υπάρχει και η μέθοδος στην οποία ορίζεται μία συνάρτηση στόχος [19], η οποία ποσοτικοποιεί την ποιότητα του πλέγματος και λύνοντας το πρόβλημα ελαχιστοποίησης, προκύπτουν οι νέες θέσεις του υπολογιστικού πλέγματος.

2 Δίκτυα Συναρτήσεων Ακτινικής Βάσης

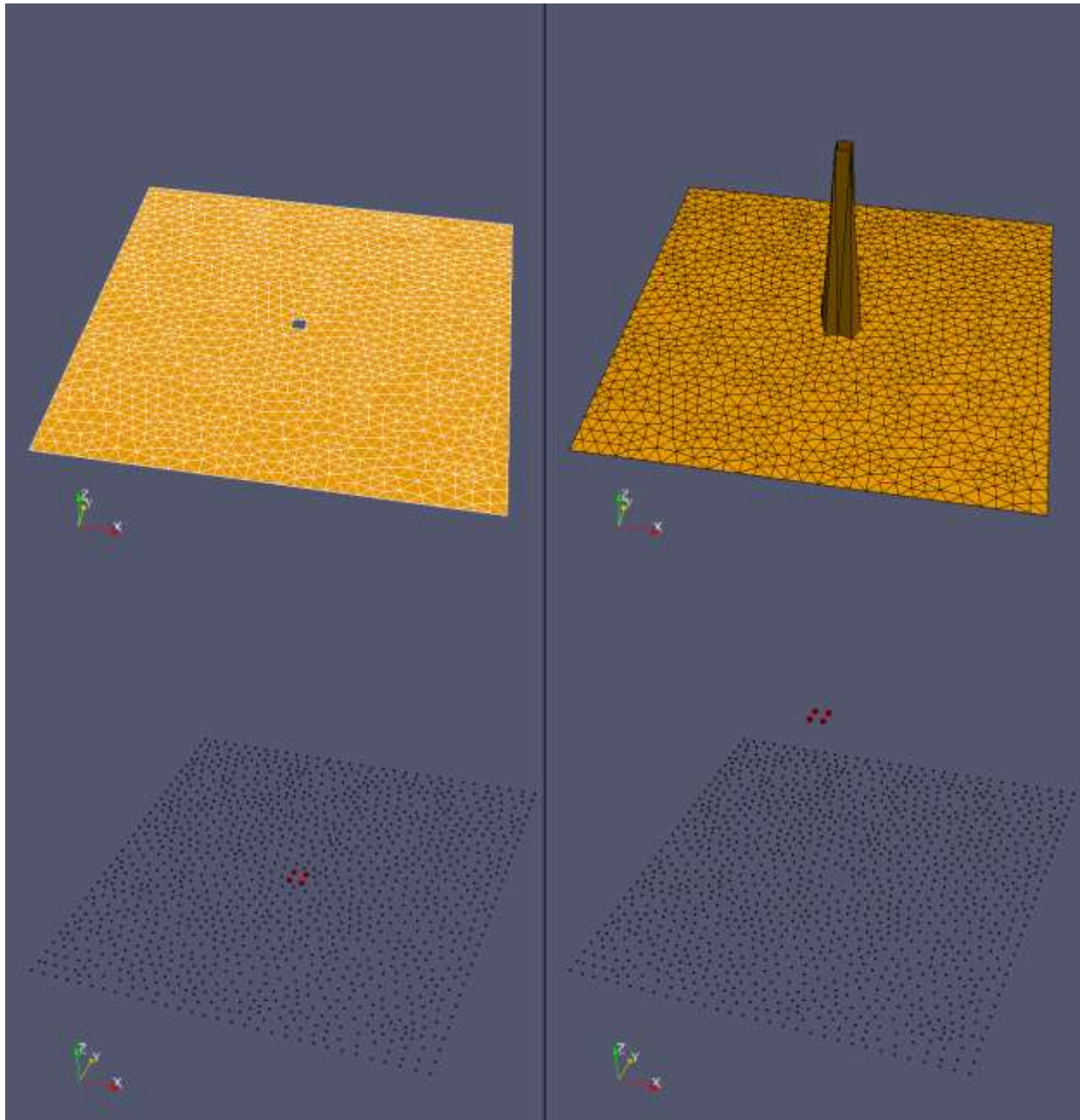
Σκοπός αυτής της ενότητας είναι η επεξήγηση της έννοιας της μετατόπισης πλέγματος με δίκτυα RBF [20],[21],[22], η οποία για λόγους απλότητας θα πραγματοποιηθεί με τη βοήθεια ενός παραδείγματος.

2.1 Μετατόπιση πλέγματος με δίκτυα (RBF)

Έστω ότι το πλέγμα είναι αυτό του σχήματος 2.1 και 2.2, το οποίο αποτελείται από τριγωνικά στοιχεία και έχει 4 κόμβους στο εσωτερικό όριο οι οποίοι περιγράφουν την εσωτερική γεωμετρία 120 κόμβους στο εξωτερικό όριο οι οποίοι περιγράφουν την εξωτερική γεωμετρία και 1110 εσωτερικούς κόμβους.



Σχήμα 2.1: Μη-δομημένο πλέγμα τριγωνικών στοιχείων με $K=4$ κόμβους της εσωτερικής γεωμετρίας, 120 κόμβους της εξωτερικής γεωμετρίας και $N=1110$ εσωτερικούς κόμβους στο x - y επίπεδο.



Σχήμα 2.2: Μη-δομημένο πλέγμα τριγωνικών στοιχείων με $K=4$ κόμβους εσωτερικής γεωμετρίας, 120 εξωτερικούς οριακούς κόμβους και $N=1110$ εσωτερικούς κόμβους στον τριδιάστατο χώρο. Δεξιά παρουσιάζεται το πλέγμα με μετατοπισμένους τους οριακούς εσωτερικούς κόμβους. **Κάτω Αριστερά:** Με κόκκινο χρώμα εμφανίζονται τα κέντρα του δικτύου δηλαδή όλοι εκείνοι οι κόμβοι ή υποσύνολο αυτών, των οποίων η τιμή της μετατόπισης είναι γνωστή. **Κάτω Δεξιά:** Με κόκκινο χρώμα εμφανίζονται οι μετατοπισμένοι εκείνοι κόμβοι, των οποίων η τιμή της μετατόπισης είναι γνωστή

Για να είναι εφικτή η χρήση ενός μοντέλου RBF πρέπει πρώτα να οριστούν τα K σε πλήθος κέντρα του, τα οποία είναι οι οριακοί κόμβοι ή υποσύνολο αυτών, των οποίων η τιμή της μετατόπισης είναι γνωστή ακόμη και όταν αυτή θα είναι μηδενική, σχήματα 2.1 και 2.2. Το μοντέλο μετατόπισης ενός οποιουδήποτε εσωτερικού κόμβου που βρίσκεται στη θέση με διάνυσμα \vec{x} υποθέτοντας ότι τα κέντρα βρίσκονται στη θέση με διάνυσμα \vec{x}_{c_i}) θα είναι

$$\begin{aligned}\Delta x &= \sum_{i=1}^K w_{x,i} \Phi(\|\bar{x}_{c_i} - \vec{x}\|) \\ \Delta y &= \sum_{i=1}^K w_{y,i} \Phi(\|\bar{x}_{c_i} - \vec{x}\|) \\ \Delta z &= \sum_{i=1}^K w_{z,i} \Phi(\|\bar{x}_{c_i} - \vec{x}\|)\end{aligned}\tag{2.1}$$

όπου $\bar{w}_x = (w_{x,1}, \dots, w_{x,K})$, $\bar{w}_y = (w_{y,1}, \dots, w_{y,K})$, $\bar{w}_z = (w_{z,1}, \dots, w_{z,K})$ είναι τα βάρη του δικτύου RBF των οποίων οι τιμές είναι άγνωστες και Φ μία συνάρτηση βάσης. Για την εύρεση αυτών των βαρών, πρέπει να εκπαιδευτεί το σύστημα, να εφαρμοστούν, δηλαδή, οι εξισώσεις 2.1, K φορές στα K κέντρα του δικτύου και έτσι δημιουργείται το σύστημα

$$\begin{bmatrix} \Phi(\|\bar{x}_{c_1} - \bar{x}_{c_1}\|) & \Phi(\|\bar{x}_{c_1} - \bar{x}_{c_2}\|) & \dots & \Phi(\|\bar{x}_{c_1} - \bar{x}_{c_K}\|) \\ \Phi(\|\bar{x}_{c_2} - \bar{x}_{c_1}\|) & \Phi(\|\bar{x}_{c_2} - \bar{x}_{c_2}\|) & \dots & \Phi(\|\bar{x}_{c_2} - \bar{x}_{c_K}\|) \\ \vdots & \vdots & \ddots & \vdots \\ \Phi(\|\bar{x}_{c_K} - \bar{x}_{c_1}\|) & \Phi(\|\bar{x}_{c_K} - \bar{x}_{c_2}\|) & \dots & \Phi(\|\bar{x}_{c_K} - \bar{x}_{c_K}\|) \end{bmatrix} \begin{bmatrix} w_{x1} \\ w_{x2} \\ \vdots \\ w_{xK} \end{bmatrix} = \begin{bmatrix} \Delta x_{c1} \\ \Delta x_{c2} \\ \vdots \\ \Delta x_{cK} \end{bmatrix}$$

για το x (ομοίως και με ίδιο μητρώο συντελεστών θα είναι και για y και για το z σε μία τριδιάστατη περίπτωση) ή, ισοδύναμα

$$\begin{aligned}A\bar{w}_x &= \bar{\Delta x}_c \\ A\bar{w}_y &= \bar{\Delta y}_c \\ A\bar{w}_z &= \bar{\Delta z}_c\end{aligned}\tag{2.2}$$

Το μητρώο A, είναι ίδιο για κάθε κατεύθυνση, είναι συμμετρικό και έχει τιμή μονάδα σε κάθε στοιχείο στη διαγώνιά του. Επίσης, αν βρεθούν τα βάρη, επόμενο βήμα είναι η χρήση του δικτύου RBF για τον υπολογισμό των μετατοπίσεων των εσωτερικών κόμβων. Αν οι εσωτερικοί κόμβοι είναι N σε πλήθος, τότε για να βρεθεί η μετατόπιση τους εφαρμόζεται ο τύπος 2.1, N φορές για τους N εσωτερικούς κόμβους, αφού αυτός ο τύπος είναι γενικός.

Όσον αφορά το υπολογιστικό κόστος του δικτύου RBF, αυτό εξαρτάται τόσο από το πλήθος K των κέντρων του, όσο και από το πλήθος N των εσωτερικών κόμβων των οποίων η μετατόπιση ζητείται να βρεθεί.

1. Αν το πλήθος των κέντρων του δικτύου (K) είναι μεγάλο, τότε η εκπαίδευση έχει μεγάλο υπολογιστικό κόστος. Το γεγονός ότι το μητρώο των συντελεστών είναι το ίδιο για x, y, z αποτελεί σημαντικό πλεονέκτημα, αλλά το υπολογιστικό κόστος δεν παύει να είναι μεγάλο.
2. Αν το πλήθος των εσωτερικών κόμβων (N) είναι μεγάλο, τότε παρόλο που ο τύπος υπολογισμού των μετατοπίσεών τους είναι εύκολος, δεν παύει να είναι μεγάλος ο βρόχος, γεγονός που οδηγεί σε μεγάλο υπολογιστικό κόστος.
3. Αν συμβαίνει ταυτόχρονα να είναι μεγάλο το πλήθος των κέντρων του δικτύου RBF και το πλήθος των εσωτερικών κόμβων, τότε το υπολογιστικό κόστος είναι πολύ μεγάλο.

2.2 Συναρτήσεις Ακτινικής Βάσης (RBF)

Οι συναρτήσεις ακτινικής βάσης (Radial Basis Functions, RBF) έχουν ως πεδίο τιμών το σύνολο των πραγματικών αριθμών και η τιμή τους εξαρτάται από την απόστασή τους από την αρχή των αξόνων, έτσι ώστε $\Phi(\vec{x}) = \Phi(\|\vec{x}\|)$. Εναλλακτικά, αντί της αρχής των αξόνων, μπορεί να εξαρτάται από την απόστασή του από ένα σημείο x_{cj} , το οποίο λέγεται κέντρο, τέτοιο ώστε $\Phi(\vec{x}, \vec{x}_{cj}) = \Phi(\|\vec{x} - \vec{x}_{cj}\|)$. Οποιαδήποτε συνάρτηση έχει την ιδιότητα $\Phi(\vec{x}) = \Phi(\|\vec{x}\|)$, λέγεται ακτινική συνάρτηση. Τις περισσότερες φορές, νόρμα είναι η Ευκλείδεια απόσταση. Οι συναρτήσεις RBF, αν χρησιμοποιηθούν σε ένα τεχνητό νευρωνικό δίκτυο ως συναρτήσεις ενεργοποίησης, μπορούν να εφαρμοστούν ως μέσο παρεμβολής. Υπάρχουν πολλών ειδών συναρτήσεις RBF οι οποίες μπορούν να χρησιμοποιηθούν για παρεμβολή και χωρίζονται σε δύο κατηγορίες, στις συναρτήσεις **ολικής επιρροής** (global support) και **τοπικής επιρροής** (compact support). Οι συναρτήσεις RBF τοπικής επιρροής, έχουν την ιδιότητα

$$\Phi(r, R) = \begin{cases} f(r, R) & 0 \leq x \leq R \\ 0 & x > R \end{cases} \quad (2.3)$$

$r = \|\vec{x} - \vec{x}_{cj}\|_2$, $f(r, R) \geq 0$, \vec{x} το διάνυσμα θέσης ενός σημείου στο χώρο και \vec{x}_{cj} το διάνυσμα θέσης του j κέντρου του δικτύου. Η τιμή της ακτίνας R , δρα ως κατώφλι επιρροής, καθορίζεται από τον χρήστη ή, μερικές φορές, υπολογίζεται από εμπειρικές σχέσεις. Αν $\xi = r/R$, συνηθίζεται και η αδιάστατη γραφή

$$\Phi(\xi) = \begin{cases} f(\xi) & 0 \leq \xi \leq 1 \\ 0 & \xi > 1 \end{cases} \quad (2.4)$$

Είναι σημαντικό ότι ένα κέντρο, έστω το \vec{x}_{cj} , των συναρτήσεων RBF τοπικής επιρροής επηρεάζει αποκλειστικά εκείνα τα σημεία που ανήκουν σε κύκλο, αν είναι διδιάστατο, ή σφαίρα, αν είναι τριδιάστατο, με κέντρο το \vec{x}_{cj} και ακτίνα R . Επομένως, τα σημεία τα οποία δεν ανήκουν σε αυτόν το κύκλο ή τη σφαίρα αντίστοιχα, δεν επηρεάζονται καθόλου από το κέντρο \vec{x}_{cj} [20]. Στον πίνακα 2.1 φαίνονται μερικές από τις συχνά χρησιμοποιούμενες συναρτήσεις ακτινικής βάσης τοπικής επιρροής όπως προτάθηκαν από τον Wendland [21]. Στον πίνακα αυτό οι τέσσερις πρώτες συναρτήσεις βασίζονται σε πολυώνυμα (Continuous Polynomial), ενώ οι τέσσερις τελευταίες βασίζονται στη συνάρτηση Thin Plate Spline, για την οποία ισχύει $f(x) = x^2 \ln(x)$. Στα σχήματα 2.3 και 2.4 φαίνεται η μορφή των συναρτήσεων ακτινικής βάσης που παρουσιάζονται στον πίνακα 2.1.

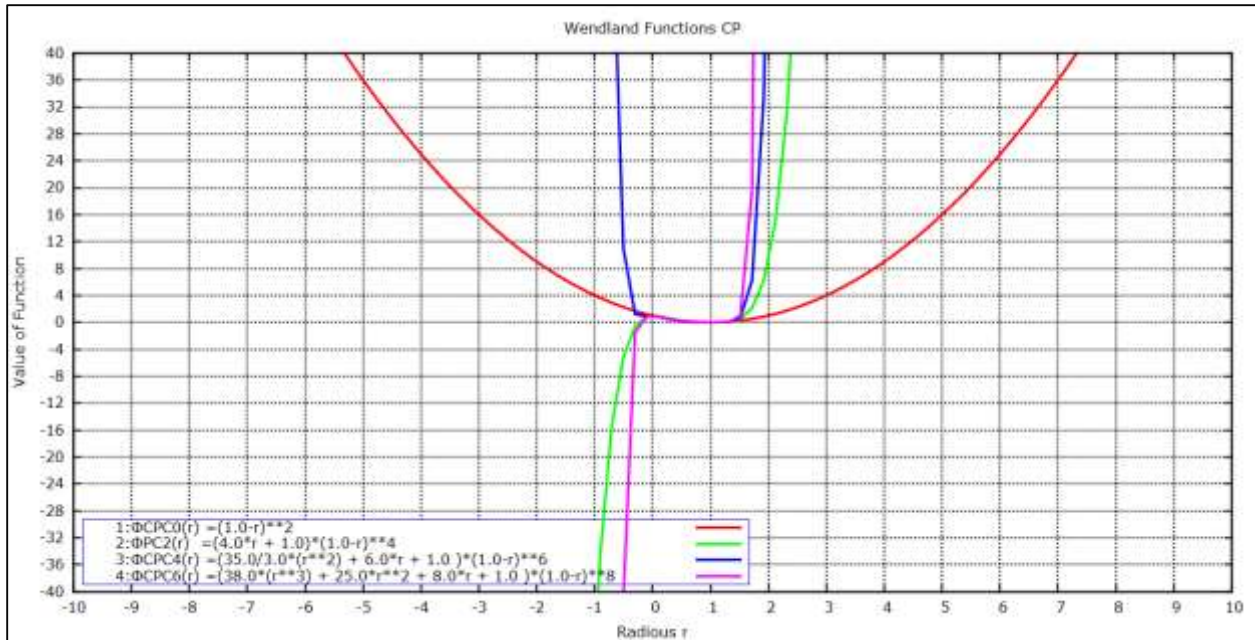
Στον αντίποδα, υπάρχουν και συναρτήσεις ολικής επιρροής, συναρτήσεις δηλαδή για τις οποίες ισχύει ότι το εκάστοτε κέντρο \vec{x}_{cj} επηρεάζει όλα τα σημεία/κόμβους του χώρου στον οποίο γίνεται η παρεμβολή, άσχετα από το πόσο μακριά είναι το υπόψη κέντρο [20]. Υπάρχουν και σε αυτήν την κατηγορία αρκετές συναρτήσεις ακτινικής βάσης μερικές από τις οποίες φαίνονται στον πίνακα 2.2 [20], [22]. Ο συντελεστής a που υπάρχει στη συνάρτηση βάσης Multiquadric Biharmonics και στη Inverse Multiquadric Biharmonics ελέγχει τη μορφή της συνάρτησης βάσης. Μεγάλη τιμή αυτής της παραμέτρου δίνει στη συνάρτηση επίπεδη μορφή ενώ η μικρή τιμή δίνει σχήμα κώνου. Στα σχήματα 2.5 και 2.6 φαίνεται η μορφή των συναρτήσεων ολικής επιρροής του πίνακα 2.2. Επίσης στα σχήματα 2.10 ως 2.15 παρουσιάζεται η επίδραση τόσο των συναρτήσεων τοπικής επιρροής όσο και των ολικής επιρροής στη συγκεκριμένη εφαρμογή του κεφαλαίου. Στα σχήματα 2.6 ως 2.8 απεικονίζεται η επίδραση μίας συνάρτησης τοπικής επιρροής για διαφορετική ακτίνα επιρροής, τα κέντρα του δικτύου καθώς και οι σφαίρες επιρροής για κάθε κέντρο. Στο σχήμα 2.8 παρουσιάζεται η επίδραση της αύξησης της ακτίνας επιρροής καθώς και η επίδραση της μεγάλης απόστασης ενός τυχαίου εσωτερικού κόμβου από τα κέντρα του δικτύου.

Όνομα	$\Phi(\xi)$
$CP C^0$	$(1 - \xi)^2$
$CP C^2$	$(1 - \xi)^4(4\xi + 1)$
$CP C^4$	$(1 - \xi)^6\left(\frac{35}{3}\xi^2 + 6\xi + 1\right)$
$CP C^6$	$(1 - \xi)^8(38\xi^3 + 25\xi^2 + 8\xi + 1)$
$CTPS C^0$	$(1 - \xi)^5$
$CTPS C^1$	$1 + \frac{80}{3}\xi^2 - 40\xi^3 + 15\xi^4 + \frac{8}{3}\xi^5 + 20\xi^2 \ln(\xi)$
$CTPS C_a^2$	$1 - 30\xi^2 - 10\xi^3 + 45\xi^4 - 6\xi^5 - 60\xi^3 \ln(\xi)$
$CTPS C_b^2$	$1 - 20\xi^2 + 80\xi^3 - 45\xi^4 - 16\xi^5 + 60\xi^4 \ln(\xi)$

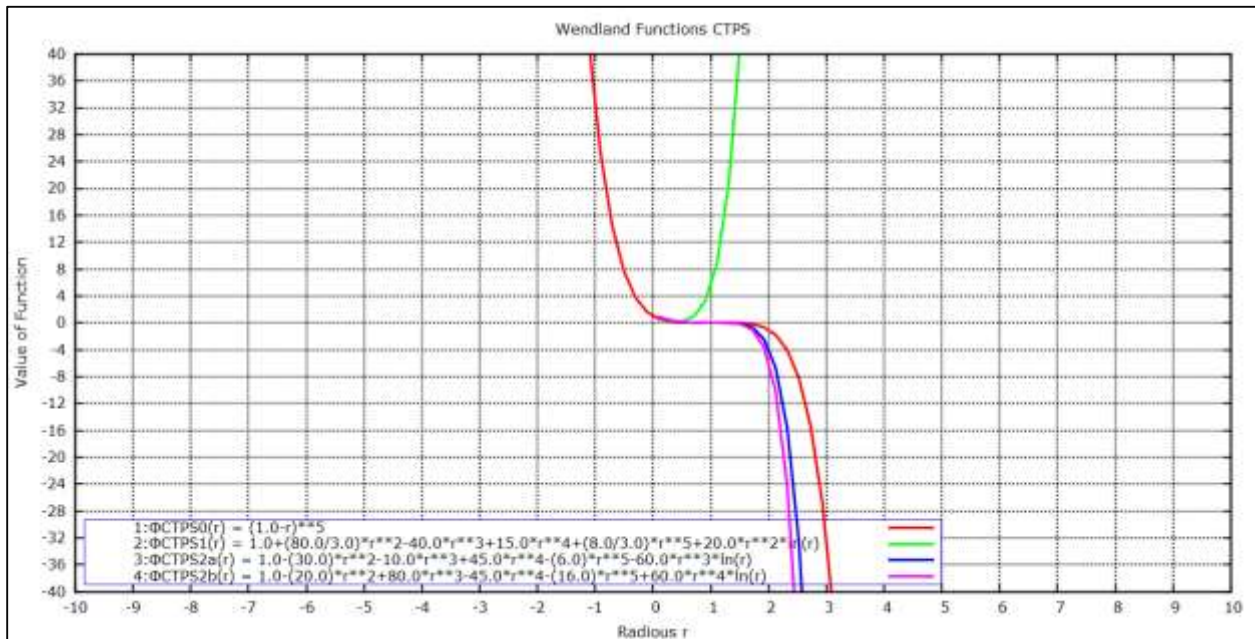
Πίνακας 2.1: Συναρτήσεις ακτινικής βάσης τοπικής επιρροής όπως προτάθηκαν από τον Wendland [21]. Οι πρώτες τέσσερις βασίζονται σε συνεχή πολυώνυμα.

Όνομα	$\Phi(r)$
<i>Gaussian</i>	$\exp(-r^2)$
<i>Multiquadric Biharmonics</i>	$\sqrt{a^2 + r^2}$
<i>Inverse Multiquadric Biharmonics</i>	$\frac{1}{\sqrt{a^2 + r^2}}$
<i>Quadric Biharmonics</i>	$\frac{1}{1 + r^2}$
<i>Inverse Quadric Biharmonics</i>	$\frac{1}{1 + r^2}$
<i>Thin Plate Spline</i>	$r^2 \ln(r)$
<i>Polyharmonic Spline</i>	$r^{2k-d}, k = \text{order of smoothness}, d = \text{dimensions}$
	$r^{2k-d} \ln(r), k = \text{order of smoothness}, d = \text{dimensions}$

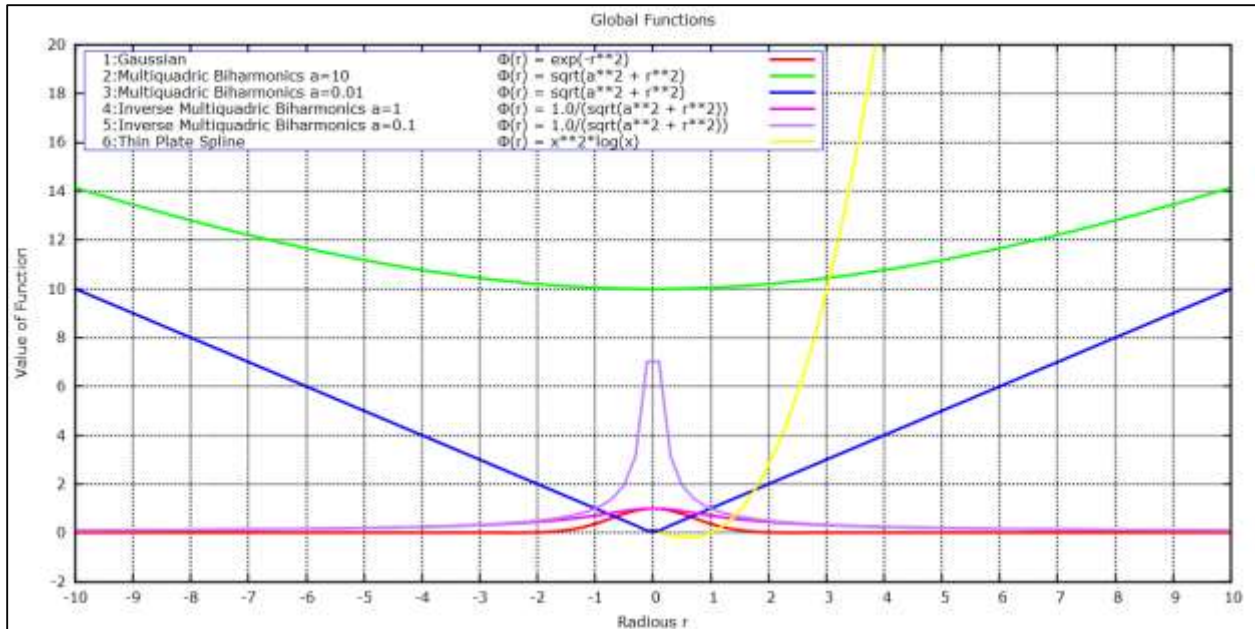
Πίνακας 2.2: Συναρτήσεις ακτινικής βάσης ολικής επιρροής που μπορούν να χρησιμοποιηθούν ως συναρτήσεις ενεργοποίησης σε ένα δίκτυο RBF.



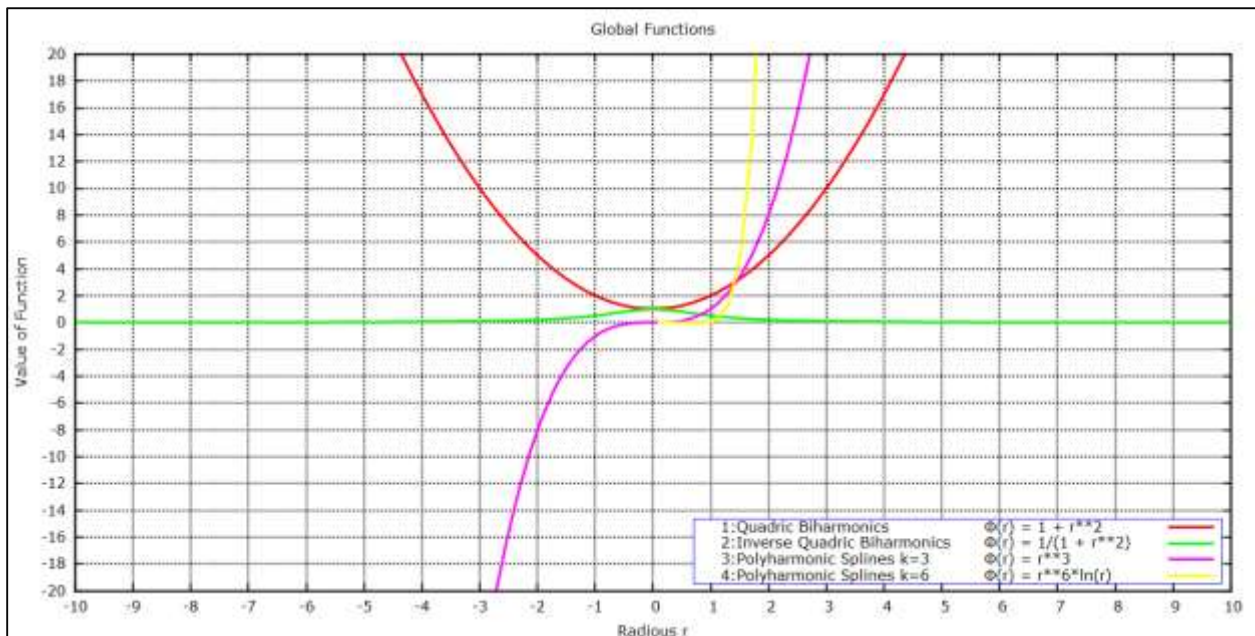
Σχήμα 2.3: Γραφική απεικόνιση των τεσσάρων πρώτων συναρτήσεων τοπικής επιρροής του πίνακα 2.1. Για τις συναρτήσεις αυτές είναι αναγκαίος και ο καθορισμός μιας ακτίνας επιρροής R , για την οποία ισχύει ότι, όταν $r > R$, τότε $\Phi(r) = 0$.



Σχήμα 2.4: Γραφική απεικόνιση των τεσσάρων τελευταίων συναρτήσεων τοπικής επιρροής του πίνακα 2.1. Για τις συναρτήσεις αυτές είναι αναγκαίος και ο καθορισμός μιας ακτίνας επιρροής R , για την οποία ισχύει ότι, όταν $r > R$, τότε $\Phi(r) = 0$.



Σχήμα 2.5: Γραφική απεικόνιση συναρτήσεων ολικής επιρροής του πίνακα 2.2. Όταν χρησιμοποιείται μια από αυτές τις συναρτήσεις, η κίνηση κάθε κέντρου του δικτύου επηρεάζει όλα τα σημεία στο χώρο.



Σχήμα 2.6: Γραφική απεικόνιση συναρτήσεων ολικής επιρροής του πίνακα 2.2. Όταν χρησιμοποιείται μια από αυτές τις συναρτήσεις, η κίνηση κάθε κέντρου του δικτύου επηρεάζει όλα τα σημεία στο χώρο.

2.3 Τα Δίκτυα RBF ως εργαλείο μετακίνησης πλέγματος

2.3.1 Χρησιμότητα των δικτύων RBF

Τα προβλήματα τα οποία καλείται να αντιμετωπίσει ένας μηχανικός στον τομέα της ΥΡΔ, χρήζουν Ηλεκτρονικού Υπολογιστή, αφού πραγματοποιεί πράξεις πολύ γρήγορα και με μεγάλη ακρίβεια. Πριν όμως οδηγηθεί το πρόβλημα στον υπολογιστή, πρέπει να δημιουργηθεί υπολογιστικό πλέγμα και να διακριτοποιηθούν οι μερικές διαφορικές εξισώσεις που διέπουν το πρόβλημα (λ.χ.

να εκφραστούν οι παράγωγοι ως πεπερασμένες διαφορές ή με κάποιο ισοδύναμο σχήμα) σε αυτό. Με την έννοια διακριτοποίησης του χώρου νοείται η γένεση πλέγματος, το οποίο αποτελείται από κόμβους και τη συνδεσμολογία τους. Τα πλέγματα αυτά, ανάλογα με τη δομή τους χωρίζονται σε δομημένα (structured grids) και μη-δομημένα (unstructured grids).

Υπάρχουν πολλές εφαρμογές στην ΥΡΔ, στις οποίες είναι απαραίτητη η μετακίνηση του υπολογιστικού πλέγματος. Τέτοιου είδους προβλήματα συναντώνται κυρίως στον τομέα της βελτιστοποίησης, της αεροελαστικότητας και σε προβλήματα ροής με κινούμενα όρια. Για παράδειγμα, στα προβλήματα βελτιστοποίησης μορφής, ελλείψει κλασικής παραμετροποίησης της προς σχεδιασμό μορφής (λ.χ. παραμετροποίηση με πολυώνυμα Bezier, NURBS και άλλα), είναι απαραίτητη η ταυτόχρονη και συμβατή μετακίνηση των κόμβων του υπολογιστικού πλέγματος, συμπεριλαμβανομένων και των ορίων του. Μια άλλη περίπτωση είναι όταν χρησιμοποιούνται μέθοδοι αιτιοκρατικής βελτιστοποίησης (gradient-based methods), όπου χρησιμοποιείται παραμετροποίηση της προς σχεδιασμό μορφής, η οποία τροποποιεί σε κάθε κύκλο βελτιστοποίησης τα όρια του υπολογιστικού χωρίου και αυτό που χρειάζεται είναι ένας μηχανισμός ο οποίος αναλαμβάνει να προσαρμόζει κατάλληλα το εσωτερικό πλέγμα. Όταν λ.χ. χρησιμοποιείται η συζυγής μέθοδος για τη βελτιστοποίηση της μορφής μιας αεροτομής, οι παράγωγοι ευαισθησίας της αντικειμενικής συνάρτησης - στόχου ως προς της μεταβλητές της παραμετροποίησης, τροποποιούν τη μορφή της αεροτομής, αλλάζοντας το σχήμα της και πρέπει να προσαρμοστεί το προηγούμενο πλέγμα στη νέα γεωμετρία, έτσι ώστε να συνεχιστεί η διαδικασία βελτιστοποίησης.

Από την άλλη πλευρά, στην αεροελαστικότητα ή σε χωρία CFD με κινούμενα όρια, τα τοιχώματα ενός στερεού σώματος λ.χ. μιας πτέρυγας ή μιας αεροτομής, μετακινούνται με το χρόνο και, σε κάθε χρονικό βήμα, πρέπει να αναπροσαρμόζεται το πλέγμα σε αυτά ώστε λ.χ. να βρεθούν οι σωστές στιγμιαίες ασκούμενες δυνάμεις. Σε όλες τις παραπάνω περιπτώσεις, η επαναπλεγματοποίηση είναι παντελώς ασύμφορη και χρονοβόρα και, ειδικά σε βιομηχανικής κλίμακας εφαρμογές, μάλλον απαγορευτική. Το πρόβλημα αυτό αντιμετωπίζεται με την ανάπτυξη ενός δικτύου με συναρτήσεις ακτινικής βάσης RBF, το οποίο αναλαμβάνει τη μετακίνηση παραμόρφωση του υπολογιστικού πλέγματος, συμπεριλαμβάνοντας τους κόμβους των ορίων ή όχι. Η μέθοδος αυτή εφαρμόζεται σε δομημένα ή μη-δομημένα πλέγματα, η μεγάλη τους αξία, όμως, φαίνεται στα μη-δομημένα. Η διαδικασία της μετακίνησης του υπολογιστικού πλέγματος με ένα δίκτυο RBF προϋποθέτει την επιλογή των κέντρων του δικτύου και μιας συνάρτησης ενεργοποίησης σε πρώτη φάση και, σε δεύτερη φάση, την εκπαίδευσή του, με σκοπό να βρεθούν οι συντελεστές βάρους του κάθε κέντρου του δικτύου για κάθε κατεύθυνση, το οποίο αναλύεται σε επόμενη παράγραφο.

2.3.2 Εκπαίδευση και χρήση του δικτύου RBF

Για να βρεθεί, λοιπόν, η μετατόπιση των κόμβων του υπολογιστικού πλέγματος, συμπεριλαμβάνοντας ή όχι τους οριακούς κόμβους, πρέπει πρώτα να επιλεγούν τα K κέντρα του δικτύου RBF και μία συνάρτηση ενεργοποίησης από τον πίνακες 2.1 και 2.2 ή κάποια άλλη διαθέσιμη στη βιβλιογραφία. Ως κέντρα του δικτύου επιλέγονται όλοι εκείνοι οι κόμβοι ή υποσύνολο αυτών, των οποίων η τιμή της μετατόπισης είναι γνωστή. Στα προβλήματα βελτιστοποίησης, όπου δεν γίνεται χρήση κλασικής παραμετροποίησης της προς σχεδιασμό μορφής, είναι δυνατόν τα δίκτυα RBF να αναλάβουν ταυτόχρονα την τροποποίηση του σχήματος της και τη μετατόπιση των εσωτερικών κόμβων του υπολογιστικού πλέγματος. Σε αυτήν την περίπτωση, το πλήθος των κέντρων του δικτύου RBF είναι αρκετά μικρό. Όταν όμως χρησιμοποιείται εργαλείο παραμετροποίησης και αναζητείται η βέλτιστη μορφή, λ.χ μιας πτέρυγας, ή όταν υπολογίζονται οι δυνάμεις σε μια παλλόμενη αεροτομή, τότε τροποποιείται η γεωμετρία της μορφής που βελτιστοποιείται ή υπολογίζεται αντίστοιχα και για να προσαρμοστεί

το εσωτερικό πλέγμα στη νέα γεωμετρία πρέπει να επιλεγούν ως κέντρα του δικτύου όλα ή ένα υποσύνολο των κόμβων της τροποποιημένης γεωμετρίας. Σε αυτήν την περίπτωση, το πλήθος των κέντρων είναι αρκετά μεγάλο, ειδικά σε προβλήματα βιομηχανικής κλίμακας που τα πλέγματα είναι τεράστια και αφορούν τριδιάστατες γεωμετρίες.

Η επόμενη φάση, πριν τη χρησιμοποίηση του δικτύου RBF, περιλαμβάνει την εκπαίδευση του δικτύου, την εύρεση δηλαδή των βαρών κάθε κέντρου για κάθε κατεύθυνση, η οποία διαδικασία είναι ίδια για όλες τις περιπτώσεις στις οποίες μπορεί να χρησιμοποιηθούν τα δίκτυα RBF, ανεξάρτητα από το πλήθος των κέντρων του. Έστω ότι, χωρίς βλάβη της γενικότητας, είναι γνωστές οι μετατοπίσεις $\Delta \vec{r}_{cx}$, $\Delta \vec{r}_{cy}$ και $\Delta \vec{r}_{cz}$ των K οριακών κόμβων ενός υπολογιστικού πλέγματος για την x, y και z κατεύθυνση αντιστοίχως (οι οποίοι είναι και τα κέντρα του δικτύου RBF). Για την εκπαίδευση του δικτύου RBF και την εύρεση των βαρών τους πρέπει να διατυπωθούν και λυθούν τα τρία γραμμικά συστήματα της σχέσης 2.2 ένα για κάθε κατεύθυνση (αν το πρόβλημα είναι διδιάστατο, τότε δεν λαμβάνεται υπόψη το τρίτο σύστημα της σχέσης 2.2).

Σε περίπτωση που χρησιμοποιείται συνάρτηση ολικής επιρροής, η τιμή στην i γραμμή και στη j στήλη του μητρώου A, της σχέσης 2.2 θα είναι ίση με $\alpha_{ij} = \varphi(\|\vec{x}_{ci} - \vec{x}_{cj}\|_2)$, ενώ αν χρησιμοποιείται μια συνάρτηση τοπικής επιρροής τότε $\alpha_{ij} = \varphi(\frac{\|\vec{x}_{ci} - \vec{x}_{cj}\|_2}{R})$, όπου R η ακτίνα επιρροής, \vec{x}_{ci} και \vec{x}_{cj} τα διανύσματα θέσης του i και j κέντρου του δικτύου RBF και φ η συνάρτηση ενεργοποίησης. Διευκρινίζεται ότι το μητρώο αυτό είναι σε όλες τις περιπτώσεις συμμετρικό, θετικά ορισμένο και στη διαγώνιο έχει παντού τιμή ίση με τη μονάδα. Είναι το ίδιο μητρώο και στα τρία συστήματα τα οποία μπορούν να λυθούν με οποιαδήποτε μέθοδο επίλυσης συστημάτων, όπως με Gauss-Seidel, LU decomposition και άλλες.

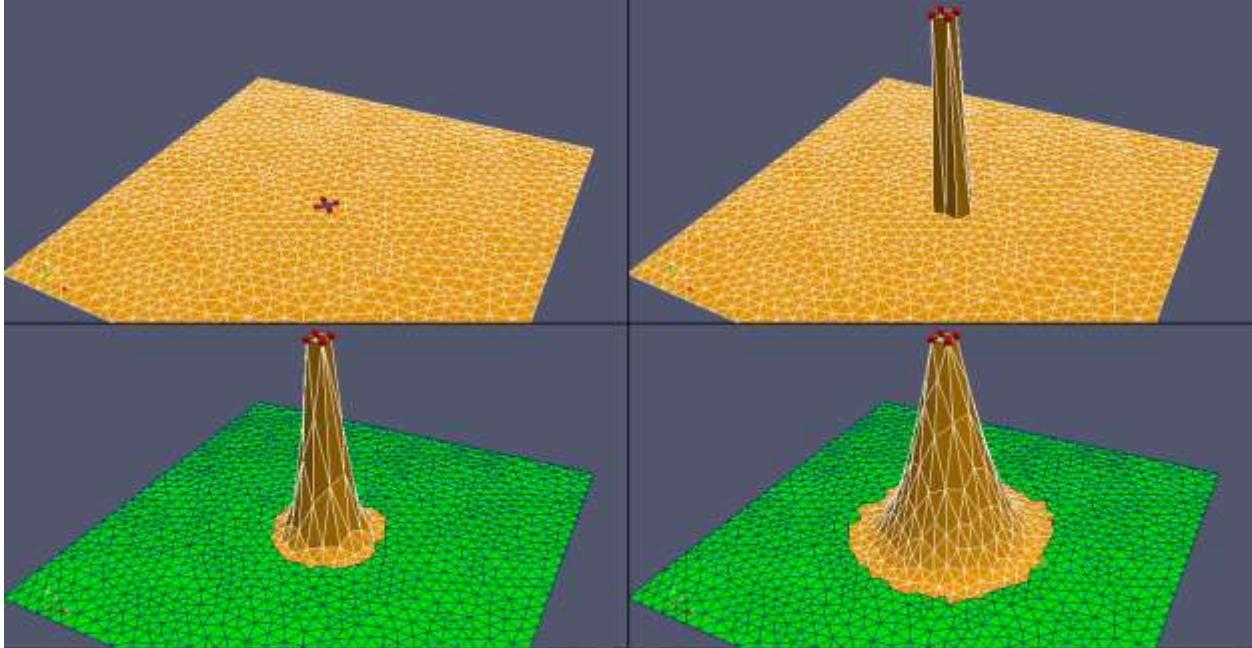
Όταν το πλήθος των κέντρων του δικτύου RBF είναι αρκετά μεγάλο, η ευθεία επίλυση των τριών συστημάτων έχει μεγάλο υπολογιστικό κόστος και για αυτό το λόγο χρησιμοποιούνται επαναληπτικές μέθοδοι επίλυσης των συστημάτων όπως για παράδειγμα η μέθοδος των συζυγών κλίσεων (Conjugate Gradient). Μετά τη διαδικασία που πραγματοποιήθηκε για την εύρεση των συντελεστών βάρους του δικτύου RBF, πλέον μπορούν να χρησιμοποιηθούν για τον υπολογισμό των μετατοπίσεων κάθε κόμβου του εσωτερικού πλέγματος. Αν οι κόμβοι αυτοί, των οποίων η νέα θέση πρέπει να υπολογιστεί, είναι N σε πλήθος, τότε οι μετατοπίσεις τους υπολογίζονται ως εξής

$$\overrightarrow{\Delta x_n} = M \overrightarrow{w_x} \quad (2.5)$$

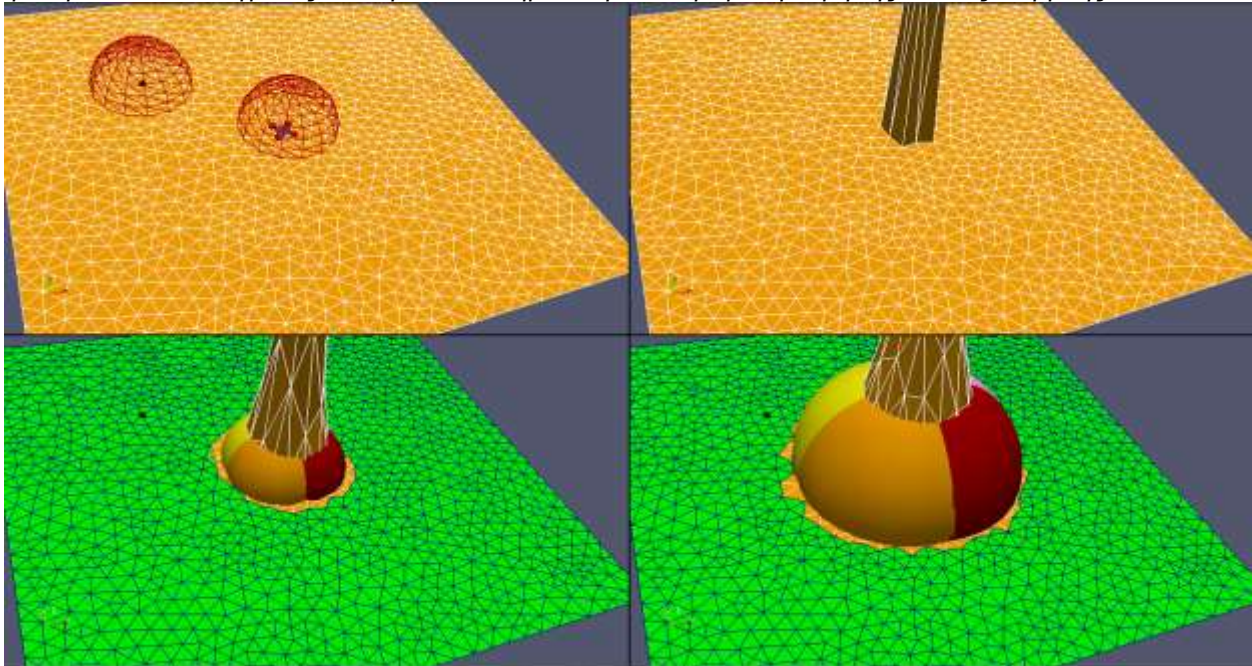
$$\overrightarrow{\Delta y_n} = M \overrightarrow{w_y} \quad (2.6)$$

$$\overrightarrow{\Delta z_n} = M \overrightarrow{w_z} \quad (2.7)$$

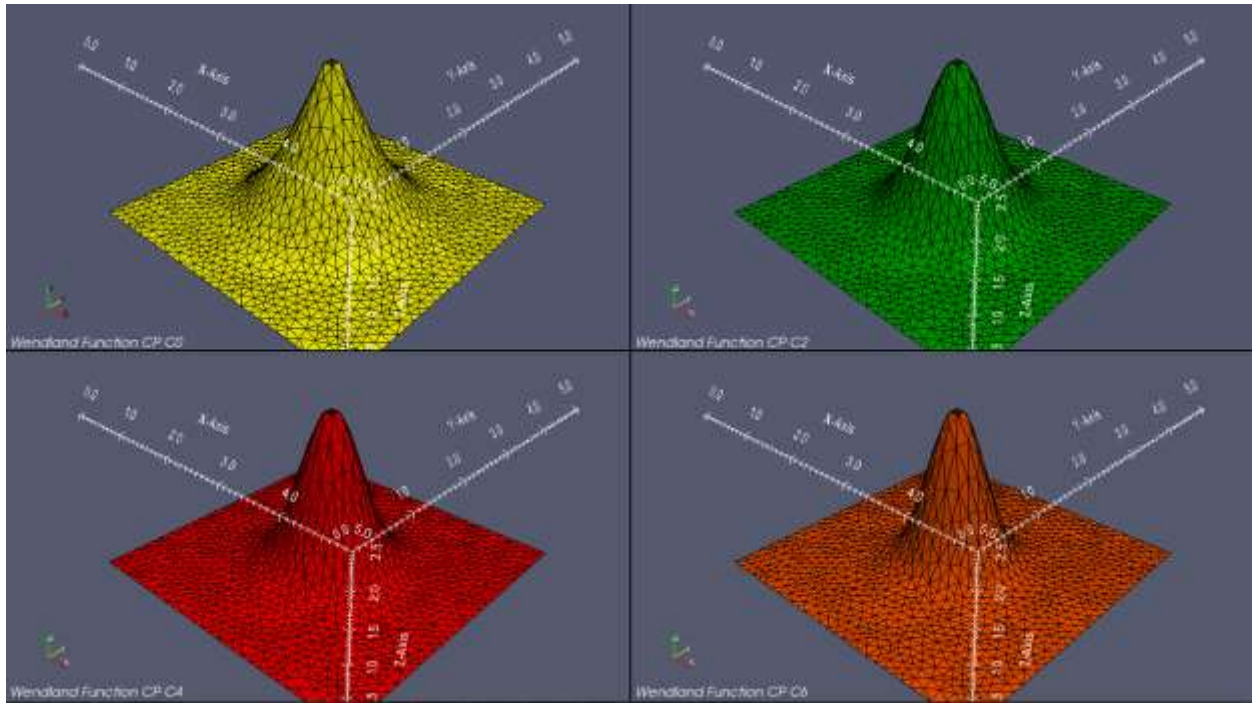
όπου $\overrightarrow{w_x}$, $\overrightarrow{w_y}$ και $\overrightarrow{w_z}$ τα γνωστά βάρη που υπολογίστηκαν κατά τη φάση της εκπαίδευσης, $\overrightarrow{\Delta x_n}$, $\overrightarrow{\Delta y_n}$ και $\overrightarrow{\Delta z_n}$ οι ζητούμενες μετατοπίσεις των κόμβων του εσωτερικού πλέγματος σε κάθε κατεύθυνση και το μητρώο M, διάστασης N×K, περιέχει τη σχέση του κάθε κόμβου του εσωτερικού πλέγματος με όλα τα κέντρα του δικτύου RBF. Η τιμή του μητρώου αυτού στην i γραμμή και στη j στήλη είναι ίση με $m_{ij} = \varphi(\|\vec{x}_i - \vec{x}_{cj}\|_2)$ (ή $m_{ij} = \varphi(\frac{\|\vec{x}_i - \vec{x}_{cj}\|_2}{R})$ σε περίπτωση που χρησιμοποιείται συνάρτηση ενεργοποίησης τοπικής επιρροής). Με \vec{x}_i ορίζεται το διάνυσμα θέσης του i κόμβου, του οποίου η νέα θέση ζητείται να υπολογιστεί. Ο πίνακας M είναι ένα μητρώο N×K και είναι ίδιος για όλα τα συστήματα. Το υπολογιστικό κόστος αυτή της φάσης είναι μικρότερο από αυτό της εκπαίδευσης του δικτύου RBF και εξαρτάται από το πλήθος N των κόμβων του εσωτερικού πλέγματος (Αυτό ισχύει για τις περιπτώσεις που το πλήθος των κέντρων του δικτύου RBF είναι μεγάλο). Πρακτικά, το κόστος είναι ο πολλαπλασιασμός του μητρώου M διάστασης N×K με τα διανύσματα-στήλη $\overrightarrow{w_x}$, $\overrightarrow{w_y}$ και $\overrightarrow{w_z}$.



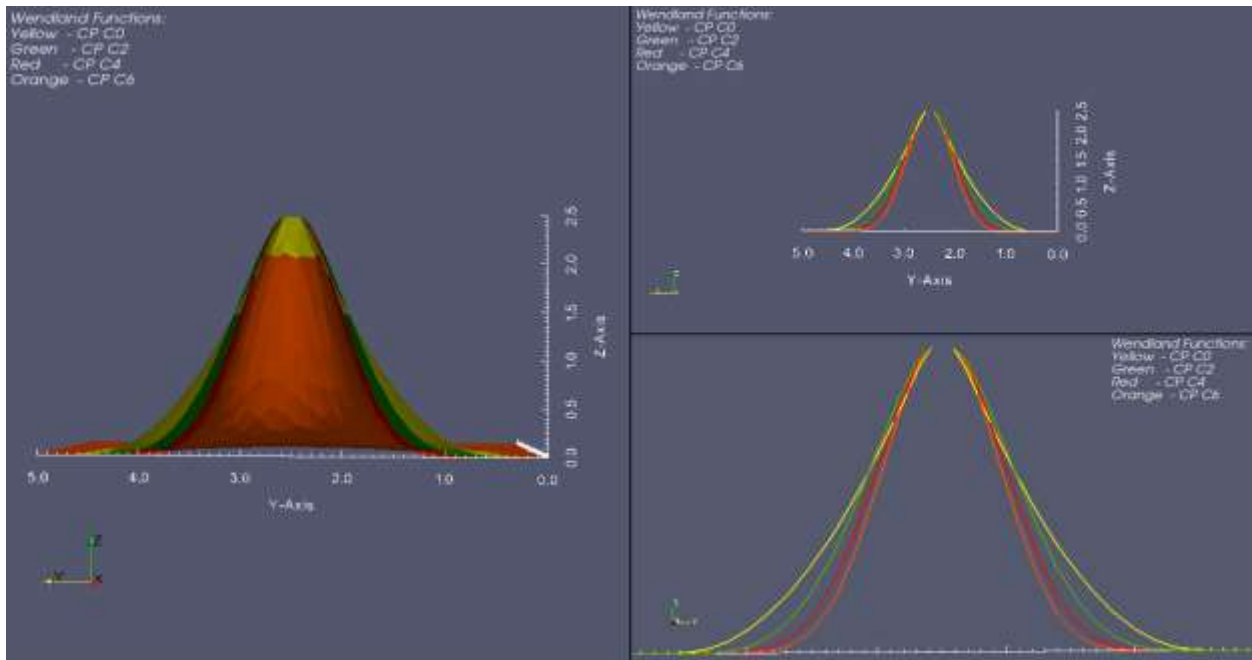
Σχήμα 2.7: Μετακίνηση πλέγματος με εφαρμογή δικτύων RBF χρησιμοποιώντας μία συνάρτηση τοπικής επιρροής. **Πάνω αριστερά:** Το αρχικό πλέγμα με τα κέντρα του δικτύου RBF. **Πάνω δεξιά:** Οι μετατοπισμένοι κόμβοι του πλέγματος. **Κάτω αριστερά:** Το αναμορφωμένο πλέγμα με ακτίνα επιρροής $R=0.5$. **Κάτω δεξιά:** Το αναμορφωμένο πλέγμα με ακτίνα επιρροής $R=1.0$. Με πράσινο χρώμα εμφανίζεται η επιφάνεια του πλέγματος που έμεινε ανεπηρέαστη ανάλογα με την τιμή της ακτίνας επιρροής.



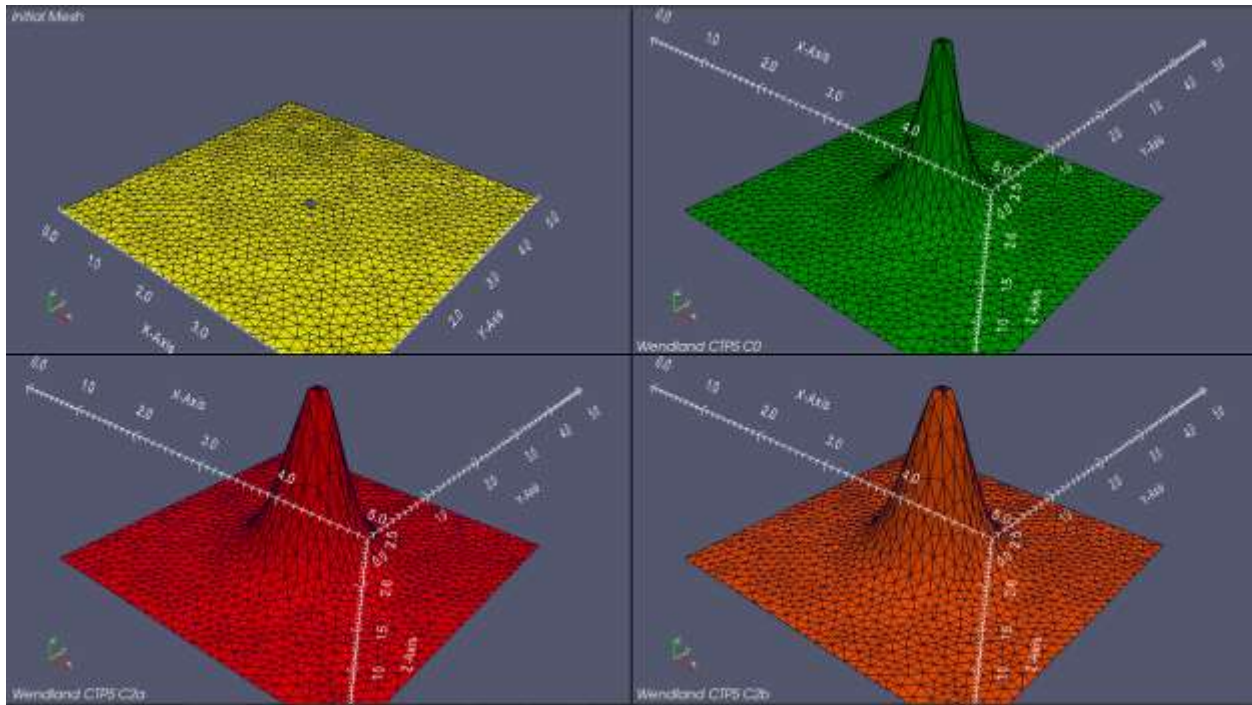
Σχήμα 2.8: Μετακίνηση πλέγματος με εφαρμογή δικτύων RBF χρησιμοποιώντας μία συνάρτηση τοπικής επιρροής. **Πάνω αριστερά:** Στο αρχικό πλέγμα παρατηρούμε τα 4 κέντρα του δικτύου και την σφαίρα-κύκλο επιρροής που αντιστοιχεί μόνο στο ένα κέντρο (για ακτίνα $R=0.5$). Με μαύρο εμφανίζεται ένας εσωτερικός κόμβος του πλέγματος και γύρω από αυτόν η σφαίρα επιρροής. **Πάνω δεξιά:** Οι μετατοπισμένοι κόμβοι του πλέγματος (κέντρα). **Κάτω αριστερά:** Το αναμορφωμένο πλέγμα με ακτίνα επιρροής $R=0.5$. **Κάτω δεξιά:** Το αναμορφωμένο πλέγμα με ακτίνα επιρροής $R=1.0$. Με διαφορετικά χρώματα εμφανίζονται οι σφαίρες επιρροής για κάθε κέντρο και με πράσινο χρώμα εμφανίζεται η επιφάνεια του πλέγματος που έμεινε ανεπηρέαστη ανάλογα με την τιμή της ακτίνας επιρροής.



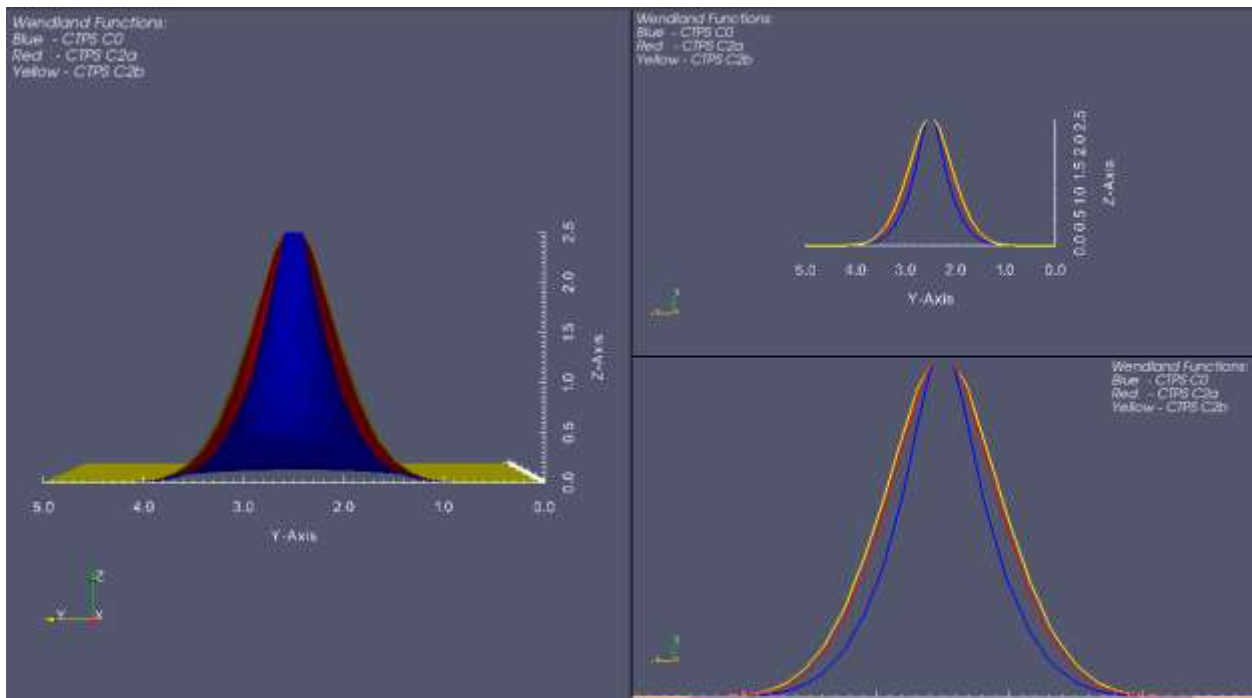
Σχήμα 2.9: Μετακίνηση πλέγματος με εφαρμογή δικτύων RBF χρησιμοποιώντας τις τέσσερις πρώτες συναρτήσεις τοπικής επιρροής του πίνακα 2.1. **Πάνω αριστερά:** Wendland Function CP C⁰. **Πάνω δεξιά:** Wendland Function CP C². **Κάτω αριστερά:** Wendland Function CP C⁴. **Κάτω δεξιά:** Wendland Function CP C⁶.



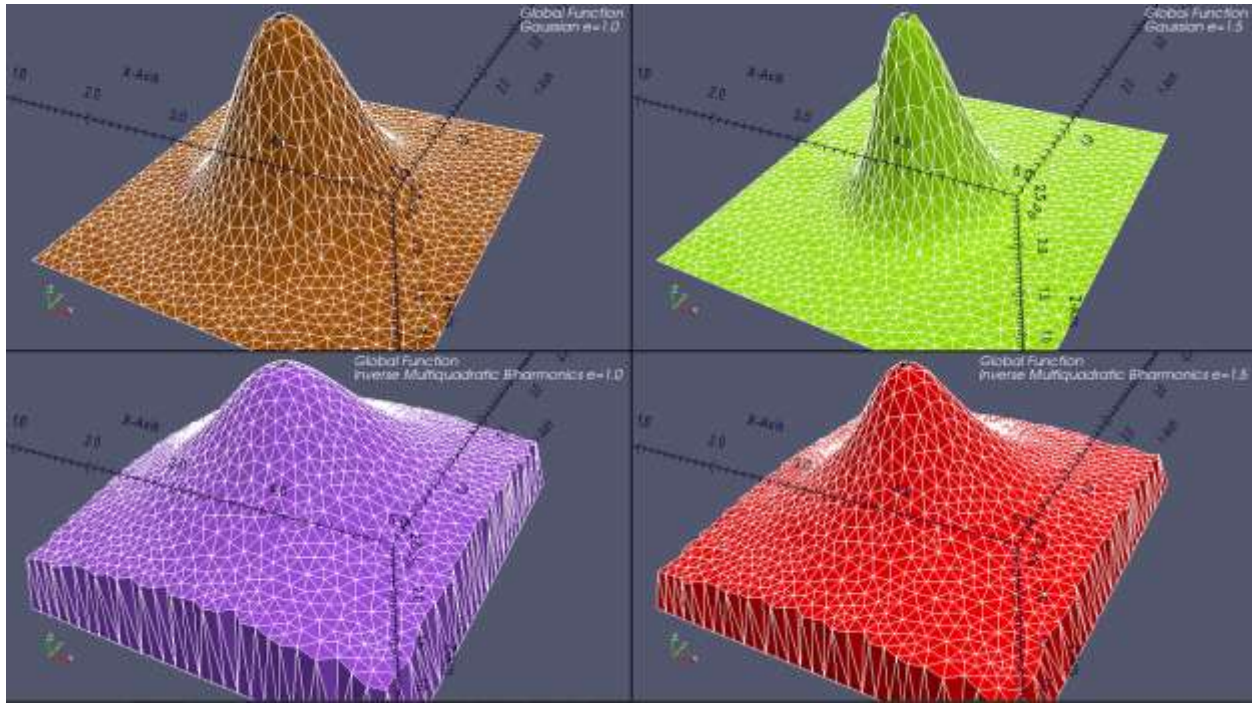
Σχήμα 2.10: Σύγκριση των τεσσάρων πρώτων συναρτήσεων τοπικής επιρροής του πίνακα 2.1. **Αριστερά:** Τομές των αναμορφωμένων πλεγμάτων.



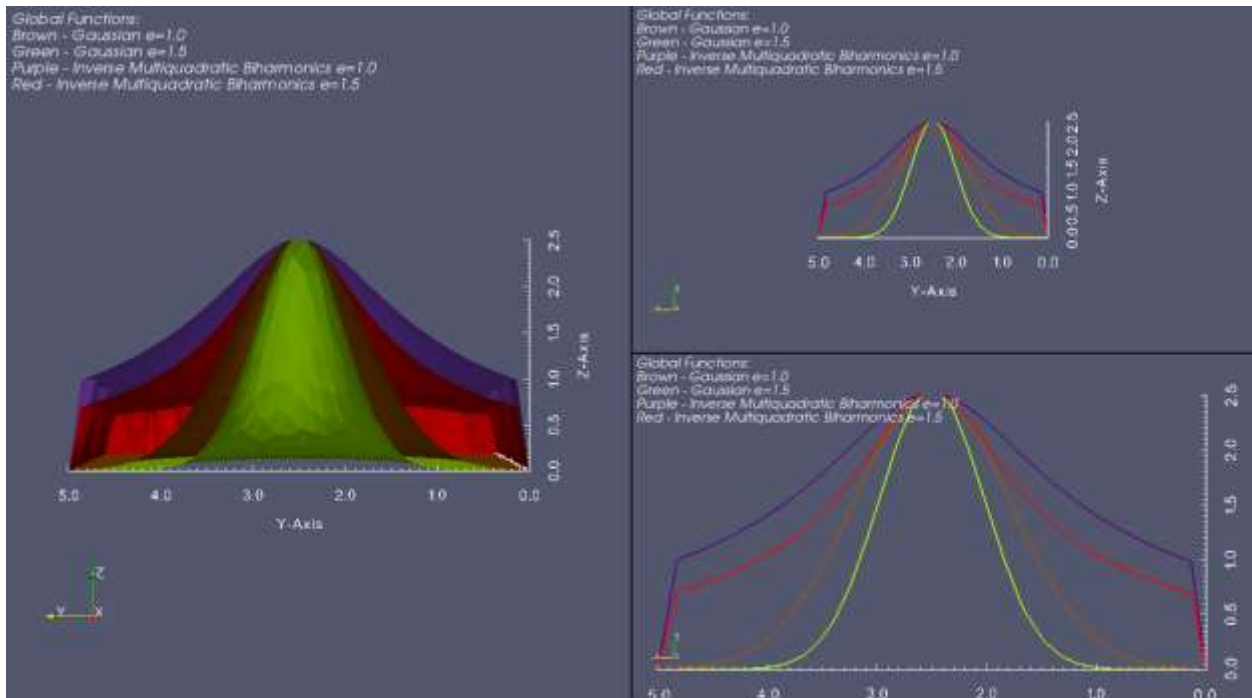
Σχήμα 2.11: Μετακίνηση πλέγματος με εφαρμογή δικτύων RBF χρησιμοποιώντας τις συναρτήσεις τοπικής επιρροής του πίνακα 2.1. **Πάνω αριστερά:** Το αρχικό πλέγμα. **Πάνω δεξιά:** Wendland Function CTPS C^0 . **Κάτω αριστερά:** Wendland Function CTPS C^2_a . **Κάτω δεξιά:** Wendland Function CTPS C^2_b .



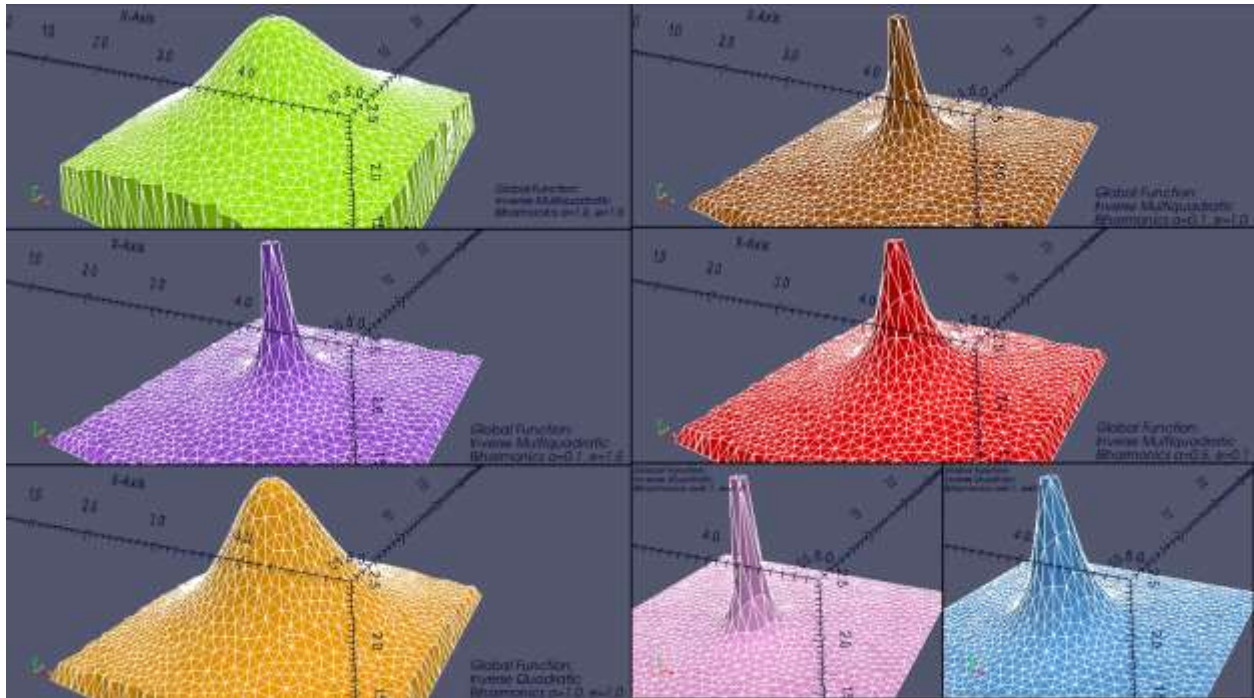
Σχήμα 2.12: Σύγκριση των συναρτήσεων τοπικής επιρροής του πίνακα 2.1. **Αριστερά:** Τομές των αναμορφωμένων πλεγμάτων.



Σχήμα 2.13: Μετακίνηση πλέγματος με εφαρμογή δικτύων RBF χρησιμοποιώντας τις συναρτήσεις ολικής επιρροής του πίνακα 2.2. **Πάνω** : Gaussian: $\exp(-\epsilon r^2)$. **Κάτω**: Inverse Multiquadratic Biharmonic $\frac{1}{\sqrt{1+\epsilon r^2}}$.



Σχήμα 2.14: Σύγκριση των συναρτήσεων ολικής επιρροής του πίνακα 2.2. **Αριστερά**: Τομές των αναμορφωμένων πλεγμάτων.



Σχήμα 2.15: Μετακίνηση πλέγματος με εφαρμογή δικτύων RBF χρησιμοποιώντας τις συναρτήσεις ολικής επιρροής του πίνακα 2.2.

3 Η GPU και η τεχνολογία CUDA

Η μονάδα επεξεργασίας γραφικών – GPU (**Graphics Processing Unit**) είναι μια κάρτα γραφικών, η οποία συχνά αποκαλείται και *device*. Σε αντιδιαστολή με την CPU (**Central Processing Unit**), δηλαδή τον κεντρικό επεξεργαστή ο οποίος συχνά αποκαλείται και *host*. Η GPU διαθέτει μεγάλο πλήθος επεξεργαστικών μονάδων συνήθως αρκετές δεκάδες ή εκατοντάδες που καλούνται πυρήνες (*cores*) [25],[41],[43]. Οι πυρήνες είναι διαθέσιμοι στον προγραμματιστή ώστε μέσω μιας κατάλληλης βιβλιοθήκης να μπορούν να χρησιμοποιηθούν για παράλληλη εκτέλεση προγραμμάτων. Η κάρτα γραφικών είναι ένας «συνεπηρεργαστής» με βασικό ρόλο την επεξεργασία των δεδομένων (γραφικών) που θα καταλήξουν στην οθόνη μας. Επομένως, δεν μπορεί να υπάρξει αυτόνομη από έναν κεντρικό επεξεργαστή. Εμάς θα μας απασχολήσει η τεχνική χρήση των καρτών γραφικών για υπολογισμούς σε εφαρμογές που παραδοσιακά γίνονταν σε κανονικούς επεξεργαστές και μόνο η υλοποίηση της NVIDIA, η αρχιτεκτονική CUDA με την γλώσσα CUDA/C, που είναι η πιο διαδεδομένη και με τις περισσότερες δυνατότητες.

3.1 Η αρχιτεκτονική CUDA

Το όνομα CUDA είναι ακρώνυμο για την ονομασία Compute Unified Device Architecture [25],[26],[27],[43]. Αυτή η αρχιτεκτονική μπορεί να βρεθεί σε όλες τις τρεις τελευταίες γενεές καρτών γραφικών της εταιρίας NVIDIA. Υπάρχει αντίστοιχη αρχιτεκτονική και για τις κάρτες γραφικών ATI. Ουσιαστικά δεν υπάρχουν μεγάλες διαφορές στην αρχιτεκτονική. Η πλατφόρμα CUDA (Compute Unified Device Architecture) είναι ένα προγραμματιστικό μοντέλο της εταιρίας NVidia το οποίο υποστηρίζει την εκτέλεση παράλληλων προγραμμάτων χρησιμοποιώντας τους πυρήνες (*cores*) μιας ή περισσότερων καρτών GPU. Η CUDA επιτρέπει στον προγραμματιστή να δημιουργήσει εκατοντάδες ή και χιλιάδες νήματα τα οποία εκτελούνται στους πυρήνες μιας ή περισσότερων GPU. Με τον τρόπο αυτό μπορούν να υλοποιηθούν αλγόριθμοι μαζικής παράλληλης επεξεργασίας. Σειριακοί αλγόριθμοι ή αλγόριθμοι με μικρό παραλληλισμό δε συμφέρει να υλοποιούνται με CUDA.

3.1.1 Η GPU– Αρχιτεκτονική Single Instruction – Multiple Threads

Ωθούμενη από την αυξανόμενη ανάγκη για τρισδιάστατα γραφικά υψηλής ευκρίνειας, η GPU ή αλλιώς η μονάδα επεξεργασίας γραφικών, εξελίχθηκε σε μία πανίσχυρη, προγραμματιζόμενη και σχετικά ευέλικτη μονάδα επεξεργασίας. Ένα βασικό συστατικό της αρχιτεκτονικής των καρτών γραφικών είναι ότι δομούνται από πολλούς πολυεπεξεργαστές (SM) οι οποίοι με την σειρά τους αποτελούνται από επεξεργαστικές μονάδες (ALU - Arithmetic Logical Unit) οι οποίες ονομάζονται SP - Streaming Processors [26], [27], [43],[48].

Αν κοιτάξουμε το σχήμα 3.1 μπορούμε να δούμε άμεσα την διαφορά στην φιλοσοφία μεταξύ κεντρικού επεξεργαστή και επεξεργαστή γραφικών. Ο πρώτος αφιερώνει περισσότερα κυκλώματα του στον έλεγχο ροής και την cache ενώ αποτελείται από λίγους γρήγορους πυρήνες. Ο δεύτερος αφιερώνει τα περισσότερα κυκλώματα στην επεξεργασία δεδομένων με πολλούς μικρούς πυρήνες. Κάθε ένας από αυτούς είναι πολύ αργός σε σύγκριση με μία μονάδα του κεντρικού επεξεργαστή, αλλά μπορεί να δουλέψει πολύ αποδοτικά ταυτόχρονα με τους υπολοίπους καθώς όλη η αρχιτεκτονική είναι βελτιστοποιημένη για μαζική παράλληλη επεξεργασία.

Γενικά, αν το πρόβλημα μας είναι παραλληλοποιήσιμο και ο λόγος των αριθμητικών διεργασιών ως προς τις διεργασίες μνήμης (μεταφορές - συγχρονισμός) είναι μεγάλος, τότε είναι ιδανικό για υλοποίηση σε κάρτα γραφικών.



Σχήμα 3.1: Σύγκριση στην φιλοσοφία μεταξύ κεντρικού επεξεργαστή και επεξεργαστή γραφικών.

Ένας κεντρικός επεξεργαστής είναι τύπου MIMD - Multiple Instruction Multiple Data ενώ οι κάρτες γραφικών είναι SIMT - Single Instruction Multiple Threads που είναι μία παραλλαγή των επεξεργαστών vector - Single Instruction Multiple Data. Με λίγα λόγια ο κεντρικός επεξεργαστής είναι βελτιστοποιημένος για να επεξεργάζεται μεγάλο όγκο δεδομένων με πολλές αποκλίσεις στην ροή (λ.χ. if..else ή αποκλίνοντα loops), ενώ μία κάρτα γραφικών προτιμά όσο το δυνατόν περισσότερα δεδομένα, όμοια διαχωρισμένα στους πολυεπεξεργαστές, και μικρές αλλαγές στην ροή των νημάτων (αν και εφόσον τηρούνται κάποιες προϋποθέσεις θα δούμε αργότερα ότι μερικές προσεγγιζόμενες διακλαδώσεις δεν θα έχουν αρνητικές συνέπειες στις επιδόσεις). Ο λόγος που οι κάρτες γραφικών δεν θέλουν μεγάλες αποκλίσεις στην ροή των νημάτων δεν είναι μόνον τα λιγότερα κυκλώματα ελέγχου ροής, αλλά και ο τρόπος που οργανώνουν τα νήματα όπως θα δούμε αργότερα αναλυτικά στην οργάνωση των δεδομένων και των νημάτων. Ο επεξεργαστής γραφικών δεν έχει την δυνατότητα άμεσου Input / Output, όπως να τυπώσει άμεσα στην οθόνη, ή να γράψει και να διαβάσει ένα αρχείο.

Οι αρχιτεκτονικές CUDA κατηγοριοποιούνται στις υπολογιστικές ικανότητες όπως φαίνονται στον πίνακα 2.1. Η 2.x είναι αρκετά πρόσφατη αρχιτεκτονική με το όνομα Fermi. Λύνει πάρα πολλά από τα προβλήματα και τους περιορισμούς που θα παρουσιάσουμε στην συνέχεια. Είναι μία αρχιτεκτονική που απευθύνεται περισσότερο στον επιστημονικό προγραμματισμό, παρά τα παιχνίδια! Οι αρχιτεκτονικές πριν την Fermi, συγκεκριμένα στις 1.1 και 1.3 που υπάρχουν στα περισσότερα μηχανήματα αξίζει να αναφερθεί εδώ ότι δεν υποστηρίζουν αριθμούς κινητής υποδιαστολής διπλής ακρίβειας (double, ή 64bit float), ενώ παρόλο που η 1.3 τους υποστηρίζει, η εφαρμογή τους σε αυτή την αρχιτεκτονική είναι σε εμβρυακό στάδιο, με πολλές μαθηματικές εντολές να λείπουν και μεγάλη θυσία σε ταχύτητα.

Αρχιτεκτονική	Κάρτες		
Kepler (compute capability 3.x)	GeForce Σειρά 600	Quadro Σειρά Kepler	Tesla K20 Tesla K10
Fermi (compute capability 2.x)	GeForce Σειρά 500 GeForce Σειρά 400	Quadro Σειρά Fermi	Tesla Σειρά 20

Tesla (compute capability 1.x)	GeForce Σειρά 200 GeForce Σειρά 9 GeForce Σειρά 8	Quadro Σειρά FX Quadro Σειρά Plex Quadro Σειρά NVS	Tesla Σειρά 10
Τυπικές εφαρμογές:	Διασκέδαση	Επαγγελματικά Γραφικά	Παράλληλη Επεξεργασία

Πίνακας 3.1: Οι αρχιτεκτονικές CUDA κατηγοριοποιούνται ανάλογα με τις υπολογιστικές ικανότητες.

3.1.1.1 Η αρχιτεκτονική TESLA

Η αρχιτεκτονική TESLA [47] αποτελεί την πρόταση της NVidia για την υλοποίηση μιας GPGPU (General Purpose computing on Graphics Processing Units). Η NVidia αποκαλεί τους πολυεπεξεργαστές της μονάδας επεξεργασίας ως Streaming Multiprocessors (SMs). Ένα κατάλληλα γραμμένο πρόγραμμα σε CUDA όταν εκτελείται στη GPU δημιουργεί μια πληθώρα από νήματα (threads), καθένα από το οποίο ανήκει σε ένα block του grid. Τα blocks χαρακτηρίζονται από έναν αύξοντα αριθμό και διανέμονται στους SMs, με το κάθε block να εκτελείται εξολοκλήρου σε έναν SM παράλληλα και ανεξάρτητα από τα υπόλοιπα. Σε κάθε SM είναι έτοιμα προς επεξεργασία παραπάνω του ενός block (πρέπει να δίνεται μέριμνα ώστε ο αριθμός των block να υπερκαλύπτει τον αριθμό των SMs της κάθε κάρτας για βέλτιστη απόδοση), έτσι ώστε μόλις ένα block ολοκληρώσει τους υπολογισμούς του να αντικατασταθεί άμεσα από ένα άλλο.

Ο κάθε πολυεπεξεργαστής (SM) αποτελείται από βαθμωτούς πυρήνες (SP - Scalar Processor) μονής ακριβείας MAD (Multiply-and-Add), πυρήνα διπλής ακριβείας MAD, ειδικές μονάδες για χειρισμό transcendentals όπως πράξεις με ημίτονα (SFU - Special Function Unit), μονάδα που αναλαμβάνει την οργάνωση των threads (instruction unit) και τέλος 16KB on-chip κοινής μνήμης για όλα τα νήματα του block (shared memory) [23]. Το overhead για την δημιουργία, τον προγραμματισμό και την εκτέλεση των νημάτων είναι πρακτικά μηδενικό, επιτρέποντας την δημιουργία block με μεγάλο αριθμό από threads. Επίσης η εντολή συγχρονισμού των threads ενός block `__syncthreads()` απαιτεί μόνο μία εντολή μηχανής. Αυτά τα δύο χαρακτηριστικά επιτρέπουν την υψηλή διακριτοποίηση της παράλληλης διαδικασίας με τη δημιουργία πληθώρας ανεξάρτητων threads, αντιστοιχώντας ακόμα και ένα στοιχείο των δεδομένων προς επεξεργασία σε κάθε thread.

Η NVidia ονομάζει αυτόν τον τρόπο επεξεργασίας SIMT (Single Instruction – Multiple Threaded) [24]. Ο κάθε πυρήνας δημιουργεί και χειρίζεται τα νήματα του σε συνεχόμενες ομάδες των 32, οι οποίες αποκαλούνται warps (CUDA Warp είναι ομάδα των 32 threads). Τα threads ενός warp αρχίζουν την εκτέλεση τους από μία κοινή διεύθυνση της μνήμης, με το καθένα να έχει τη δική του διεύθυνση εκτέλεσης (instruction address) και τους δικούς του καταχωρητές, και συνεχίζουν την εκτέλεση τους ανεξάρτητα. Εάν κάποιο νήμα ακολουθήσει κάποιο διαφορετικό κομμάτι του κώδικα μετά από εντολή διακλάδωσης, η εκτέλεση του θα γίνει ανεξάρτητα και επιπλέον (όχι παράλληλα) από τα υπόλοιπα, οπότε και τέτοιες διακλαδώσεις πρέπει να αποφεύγονται με τον έξυπνο διαχωρισμό του προβλήματος στα blocks (ή και σε warps, αφού κάθε warp εκτελείται αυτόνομα).

Σε αντίθεση με τις συνήθεις αρχιτεκτονικές SIMD που απαιτούν το μέγεθος του διανύσματος παράλληλης επεξεργασίας (vector length) για να μεταγλωττιστούν (compile) για μία συγκεκριμένη αρχιτεκτονική, στην SIMT το μέγεθος του warp δεν απαιτείται, οπότε και ένα πρόγραμμα γραμμένο σε CUDA θα εκτελεστεί σε οποιαδήποτε GPU, με τους SM να αναλαμβάνουν τον διαχωρισμό και τον προγραμματισμό των threads. Πρακτικά ο προγραμματιστής ορίζει το μήκος του vector length με τον ορισμό του μεγέθους του block. Επιπρόσθετα, τα νήματα κάθε warp στην

αρχιτεκτονική SIMT ακολουθούν ανεξάρτητες μεταξύ τους αλληλουχίες εντολών, αντίθετα με τις διανυσματικές αρχιτεκτονικές (vector machines) SIMD όπου ο προγραμματιστής πρέπει να προβλέψει και να χειριστεί τις παρεκκλίσεις από την κοινή ακολουθία εντολών. Σε κάθε χρόνο εκτέλεσης εντολής ο SM επιλέγει ένα warp που είναι έτοιμο προς εκτέλεση και εκτελεί την εντολή ταυτόχρονα για όλα τα νήματα του warp. Απώροια αυτού είναι ότι τα block πρέπει να περιέχουν πάντα περισσότερο από 32 νήματα, με δοκιμές να δείχνουν ότι 64 είναι αρκετά για να υπάρχει συνέχεια μία πληθώρα στοιχείων έτοιμων για επεξεργασία χωρίς να εξαντλείται η on-chip μνήμη από τα πολλά warps που είναι έτοιμα προς εκτέλεση [23].

Η μνήμη που υπάρχει πάνω (on-chip) σε κάθε SM είναι

- Ένα set 32bit καταχωρητών (registers).
- Μία κοινή μνήμη (shared memory) για όλους τους πυρήνες (SPs).
- Μία μνήμη μόνο ανάγνωσης constant cache για όλους τους SPs που επιταχύνει τις αναγνώσεις από τον χώρο constant memory της κύριας μνήμης της κάρτας γραφικών.
- Μία μνήμη μόνο ανάγνωσης texture cache για όλους τους SPs που επιταχύνει τις αναγνώσεις από τον χώρο texture memory της κύριας μνήμης της κάρτας γραφικών.

Η κύρια μνήμη της κάρτας γραφικών χαρακτηρίζεται ως local ή global μνήμη και η προσπέλαση σε αυτήν δεν γίνεται διαμέσου κάποιας μνήμης cache.

3.1.1.2 Η αρχιτεκτονική Fermi

Η αρχιτεκτονική Fermi [26] έφερε σημαντικές αλλαγές στις επιδόσεις και τα χαρακτηριστικά των καρτών NVidia. Εκτός από την αναμενόμενη αύξηση της ταχύτητας μέσω των περισσότερων Cuda cores, ως σημαντικές αναβαθμίσεις θα μπορούσαν να θεωρηθούν η ως και 8 φορές αύξηση της ταχύτητας σε διπλής ακρίβειας αριθμητικές πράξεις καθώς και η δυνατότητα ταυτόχρονης εκτέλεσης πολλών πυρήνων μέσω των streams. Αυτό σημαίνει ότι όταν πληρούνται ορισμένες προϋποθέσεις, έως και 16 διαφορετικοί πυρήνες μπορούν να εκτελούνται ταυτόχρονα, όταν σε αντίθεση περίπτωση θα εκτελούνταν σειριακά. Οι προϋποθέσεις αυτές αφορούν τον μέγιστο αριθμό των threads (η πιο πρακτικά των warps) που μπορούν να εκτελούνται ταυτόχρονα. Αυτό σημαίνει ότι όταν ένας πυρήνας απασχολεί πλήρως την κάρτα δεν υπάρχει δυνατότητα εκτέλεσης κάποιου άλλου που θα βρίσκεται σε άλλο stream. Το όριο βέβαια της πλήρους κατάληψης της κάρτας δεν αφορά αποκλειστικά τα threads αλλά εξαρτάται και από την χρήση των registers και της shared μνήμης. Παρόλα αυτά παρέχεται η δυνατότητα παύσης κάποιων warps όταν εκτελούν αργές διαδικασίες, όπως πρόσβαση στην μνήμη, και την αντικατάσταση τους από άλλα αυξάνοντας έτσι τα παραπάνω όρια, κάτι που ουσιαστικά αναφέρεται ως η διαδικασία του context switching. Τέλος, η αρχιτεκτονική Fermi δίνει τη δυνατότητα επιλογής ανάμεσα σε 48 KB shared μνήμης και 16 KB L1 cache η το αντίθετο (16 KB shared, 48 KB L1), επιλογή που μπορεί να ρυθμιστεί μέσω ειδικής εντολής.

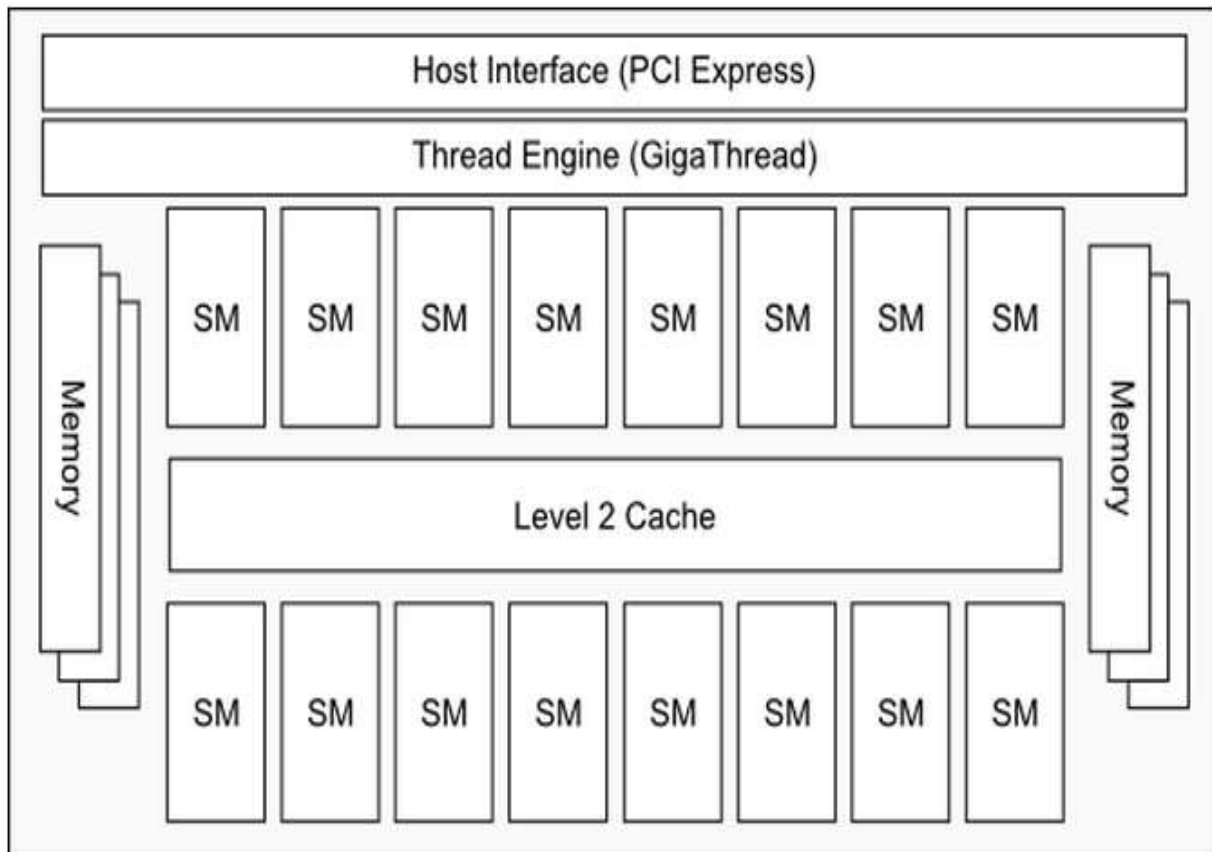
Σήμερα υπάρχουν διαθέσιμες κάρτες με την νέα αρχιτεκτονική Kepler η οποία δεν αποτέλεσε σημαντική αναβάθμιση στα χαρακτηριστικά της Cuda, αλλά κυρίως εστιάστηκε στη μεγάλη βελτίωση της κατανάλωσης ενέργειας της κάρτας. Ως μοναδικό νέο χαρακτηριστικό αναφέρεται η δυνατότητα εκτέλεσης νέων πυρήνων μέσα από ήδη ενεργούς πυρήνες. Στο μέλλον, αναμένονται κάρτες της γενιάς Maxwell και Volta, ενώ σημαντικό στοιχείο αποτελεί η ολοένα και πιο βελτιωμένη διαδικασία προσπέλασης της κύριας μνήμης και της διαδικασίας του coalescing access. Μάλιστα, η γενιά Volta υπόσχεται νέο σχεδιασμό, ώστε η κύρια μνήμη να βρίσκεται στο ίδιο chip με τους πολυεπεξεργαστές προσφέροντας εύρος έως και 1 TB/s.

Τα τελευταία χρόνια υπήρξαν ραγδαίες εξελίξεις στο αρχιτεκτονικό μοντέλο των καρτών GPU. Αν και υπήρχαν ήδη από το 1999 ο προγραμματισμός τους με σκοπό την υλοποίηση αλγορίθμων γενικού σκοπού (δηλ. όχι μόνο graphics) ήταν εξαιρετικά δύσκολος. Το μοντέλο ορόσημο που

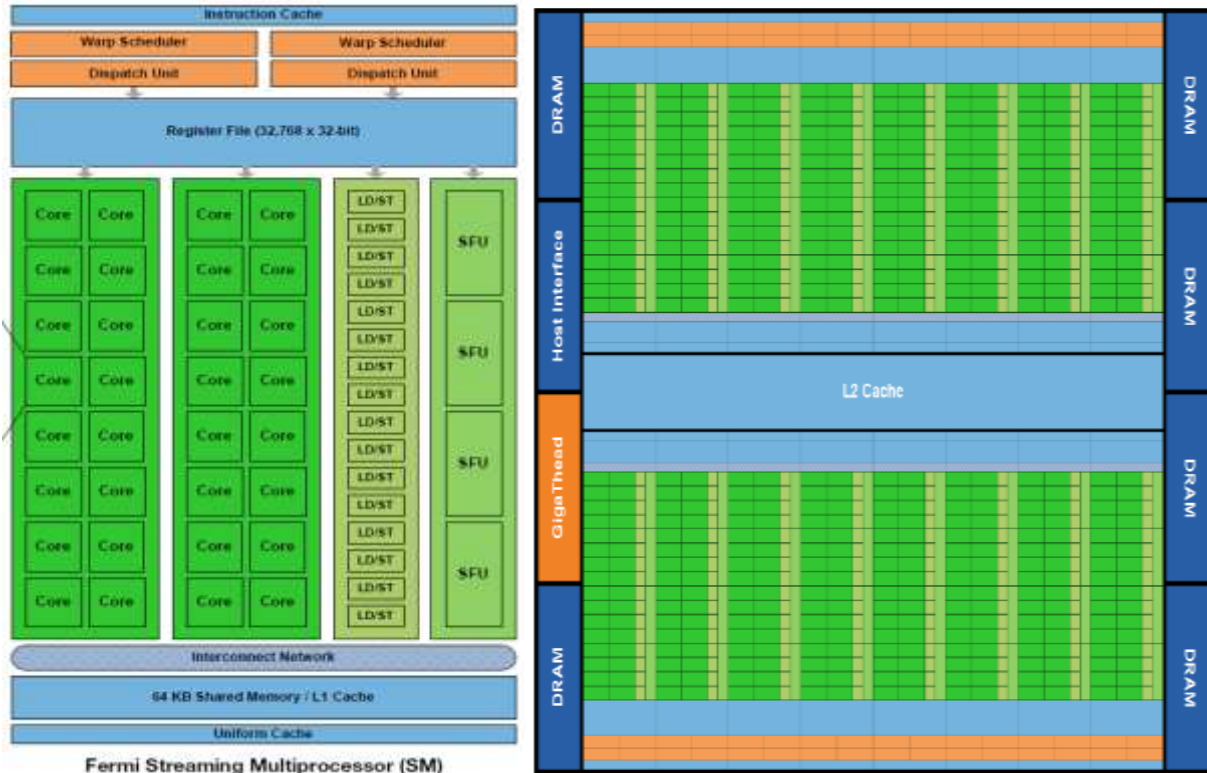
γέννησε την ιδέα των καρτών GPU για προγραμματισμό γενικού σκοπού ήταν η κάρτα GeForce 8800 που βασίστηκε στην αρχιτεκτονική G80 της NVidia. Συγκεκριμένα η αρχιτεκτονική G80

- ήταν η πρώτη αρχιτεκτονική GPU που υποστήριζε τη γλώσσα C διέθετε για πρώτη φορά ένα ενιαίο επεξεργαστή για υπολογισμούς graphics (vertex, geometry, pixel) και πράξεις γενικού σκοπού.
- εισήγαγε για πρώτη φορά το προγραμματιστικό μοντέλο *Single-Instruction Multiple Thread (SIMT)* όπου πολλοί κλώνοι του ίδιου προγράμματος εκτελούνται ως διαφορετικά νήματα με διαφορετικά id.
- εισήγαγε την κοινή μνήμη (shared memory) και τον συγχρονισμό μέσω φραγμάτων (barrier synchronization) για την επικοινωνία μεταξύ νημάτων.

Αν και η αρχιτεκτονική G80 βελτιώθηκε με το λανσάρισμα της G200 το 2008, η δεύτερη γενιά αρχιτεκτονικής GPU θεωρείται η Αρχιτεκτονική Fermi. Τα παρακάτω σχήματα δείχνουν το αρχιτεκτονικό διάγραμμα ενός chip GPU που ακολουθεί την Αρχιτεκτονική Fermi.



Σχήμα 3.2: Το αρχιτεκτονικό διάγραμμα ενός chip GPU που ακολουθεί την Αρχιτεκτονική Fermi.



Σχήμα 3.3: Το αρχιτεκτονικό διάγραμμα ενός chip GPU και ενός Streaming Multiprocessor που ακολουθούν την Αρχιτεκτονική Fermi.

Στο chip αυτό υπάρχουν:

16 Streaming Processors (SM) με 32 πυρήνες (cores) ο καθένας οπότε συνολικά υπάρχουν 512 πυρήνες. Οι SM είναι τοποθετημένοι γύρω από μια κοινή μνήμη cache 2ου επιπέδου μεγέθους 768 KB.

6 τράπεζες μνήμης 64bit που υποστηρίζουν μέχρι 6GB μνήμης τύπου GDDR5 DRAM.

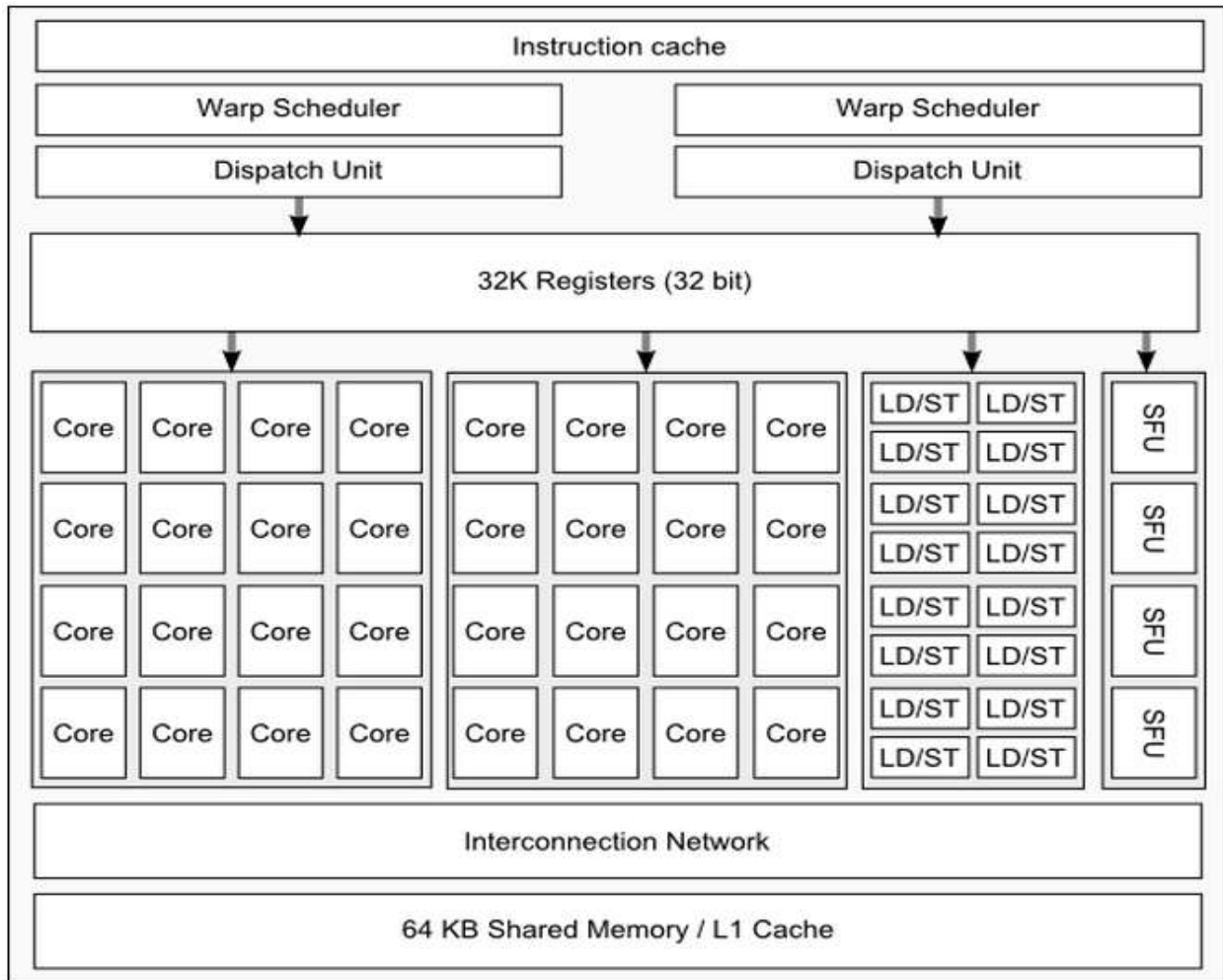
Ένα host-interface που συνδέει την GPU με την CPU μέσω διαύλου PCI Express.

Ο δρομολογητής GigaThread engine ο οποίος κατανέμει τα blocks των νημάτων στους SM.

Η ιεραρχία των threads της CUDA (grid → blocks of threads → threads) αντιστοιχεί στην ιεραρχία των επεξεργασιών της GPU (GPU → SMs → cores):

Ένα πλέγμα (grid)	εκτελείται σε	μια GPU
Ένα block νημάτων (threads)		ένα SM
Ένα νήμα (thread)		ένα πυρήνα (core)

Ο Streaming Multiprocessor εκτελεί ένα ή περισσότερα block νημάτων. Αποτελείται από πολλούς πυρήνες (cores) όπως φαίνεται στο παρακάτω σχήμα. Το σχήμα 3.4 δείχνει το αρχιτεκτονικό διάγραμμα ενός SM της αρχιτεκτονικής Fermi.



Σχήμα 3.4: Το αρχιτεκτονικό διάγραμμα ενός SM της αρχιτεκτονικής Fermi.

Ένας SM της αρχιτεκτονικής Fermi περιέχει:

- 32 πυρήνες (cores). Κάθε πυρήνας είναι ένας πολύ απλός επεξεργαστής ο οποίος περιέχει μια πλήρως pipelined Integer Arithmetic Logic Unit (ALU) και μια Floating Point Unit (FPU).
- 16 units Load/Store.
- 4 Special Function Units (SFU) οι οποίες εκτελούν πολύπλοκες μαθηματικές συναρτήσεις όπως ημίτονο (sine), συνημίτονο (cosine), αντίστροφο ($1/x$) και τετραγωνική ρίζα.
- 32768 (Registers) μήκους 32 bit.
- Μια Κοινή μνήμη μεγέθους 64 KB.
- Μια κοινή Data Cache επιπέδου 1 και μια Instruction Cache.
- Ένα κύκλωμα δικτύου για διασύνδεση με τους υπόλοιπους SM.
- Δύο warp scheduler's και δυο instruction dispatch units.

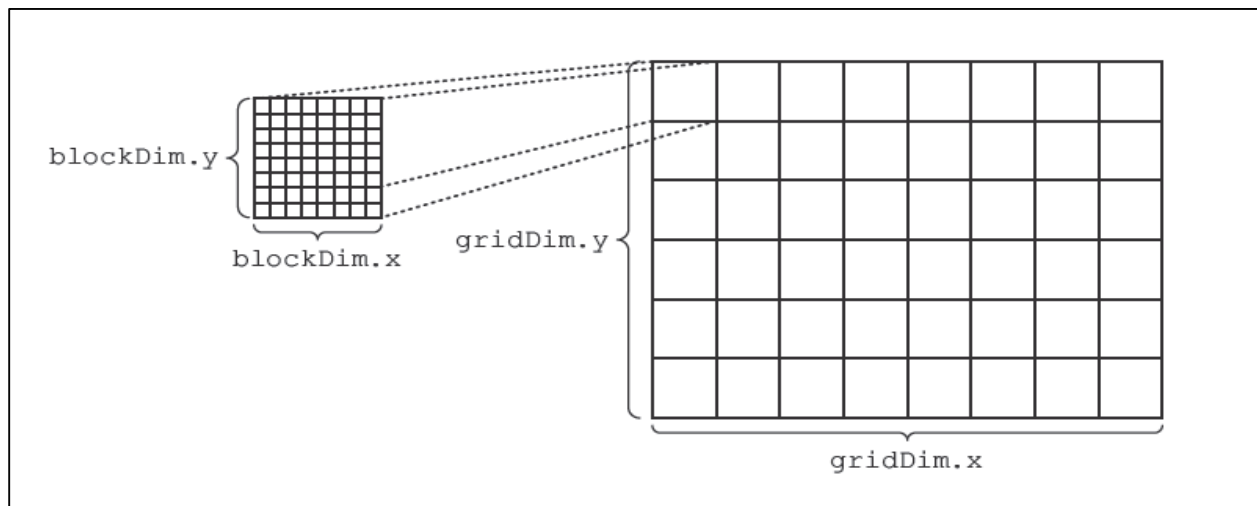
Ένας SM δρομολογεί νήματα σε groups που καλούνται warps. Το πλήθος των νημάτων σε ένα warp σχετίζεται άμεσα με το πλήθος των πυρήνων που διαθέτει ο SM. Στο συγκεκριμένο παράδειγμα κάθε warp έχει 32 threads. Υπάρχουν 2 warp-schedulers και δύο μονάδες ανάθεσης εντολών (instruction dispatch units). Δηλαδή, ο Fermi SM εκτελεί ταυτόχρονα 2 εντολές προερχόμενες από 2 διαφορετικά warps. Μια εντολή από ένα warp μπορεί, ανάλογα με το είδος της, να ανατεθεί

- είτε σε ένα πυρήνα από μια 16-άδα πυρήνων.
- είτε σε μια από τις 16 Load/Store units.
- είτε σε ένα από τα 4 Special Function Units.

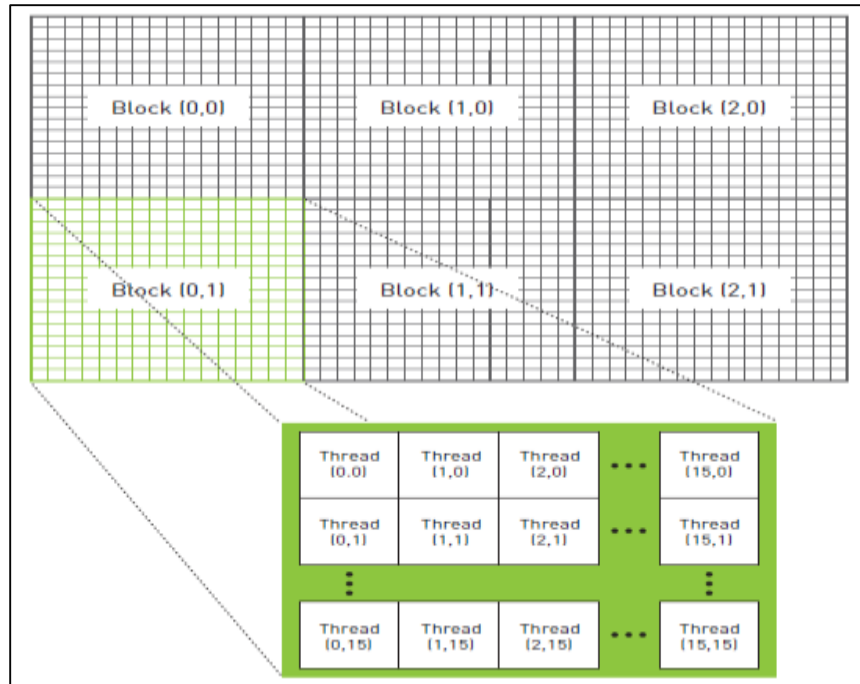
3.1.2 Οργάνωση δεδομένων και νημάτων

Τα νήματα, χωρίζονται σε grids και αμέσως μετά σε blocks. Το μέγιστο μέγεθος για κάθε διάσταση (ανεξάρτητα) ενός grid είναι { 65535, 65535, 65535 }, ενώ το μέγιστο μέγεθος για κάθε διάσταση ενός block είναι { 1024, 1024, 64}. Ένα block δεν μπορεί να περιέχει όμως πάνω από 1024 νήματα για υπολογιστική ικανότητα 1.3 και άνω, που είναι και ο μέγιστος αριθμός των ενεργών νημάτων σε κάθε στιγμή ανά πολυεπεξεργαστή. Το μέγεθος των grids και των blocks που θα χρησιμοποιήσουμε μπορεί να καθοριστεί κατά την εκτέλεση του προγράμματος. Τα νήματα οργανώνονται ως προς την εκτέλεση στα warps των 32 threads. Τα warps εξαρτώνται από την αρχιτεκτονική του ίδιου του επεξεργαστή γραφικών και το μέγεθος τους είναι δεδομένο και σταθερό, επομένως δεν εξαρτάται από το μέγεθος των blocks ή των grids. Κάθε φορά στην κάρτα γραφικών θα εκτελείται ένα grid μόνο με τα blocks που του αναλογούν, δηλαδή το σύνολο των νημάτων όλων των blocks που ανήκουν στο συγκεκριμένο grid. Τα νήματα αυτά θα οργανωθούν σε warps από τον επεξεργαστή και για κάθε warp ετοιμάζονται οι εντολές για τα νήματα του.

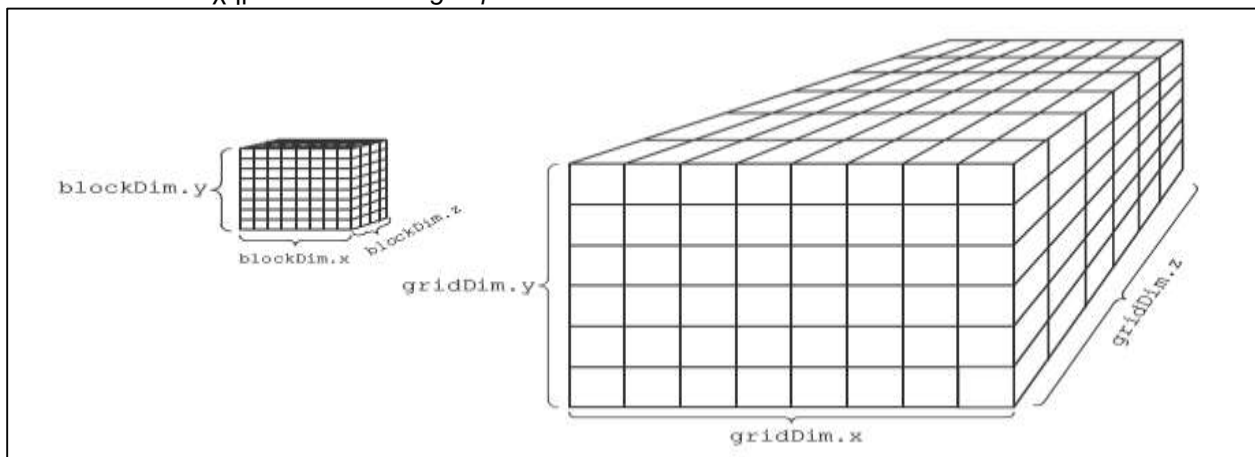
Ο τρόπος με τον οποίο θα εκτελεστούν τα warps δεν είναι απλά παράλληλος, η κάρτα γραφικών θα προσπαθεί πάντα να κρατάει όλους τους πολυεπεξεργαστές της πλήρως απασχολημένους. Αυτό σημαίνει ότι δεν μπορεί κανείς να μας εγγυηθεί καμία τάξη στην σειρά εκτέλεσης των threads. Αν ένα warp πρέπει να περιμένει εξαιτίας λ.χ. μίας εντολής ανάγνωσης από ένα thread του προς την μνήμη, η κάρτα γραφικών θα βάλει το warp στην αναμονή μέχρι να ολοκληρωθεί η εντολή αυτή και θα αρχίσει να εκτελεί κάποια άλλο warp. Επίσης, περνάει από όλες τα warp, εκτελώντας μόνο ένα κομμάτι του προγράμματος που τους αναλογεί κάθε φορά. Αν και υπάρχουν αρκετοί τρόποι να επιβάλουμε κάποια τάξη σε αυτό το χάος, γενικά, όπως θα δούμε και στην παρουσίαση του προγραμματισμού σε κάρτα γραφικών, ο τρόπος που επεξεργαζόμαστε τα δεδομένα μας πρέπει να είναι όσο πιο ασυσχέτιστος, ή αλλιώς, αυθεντικά παράλληλος γίνεται.



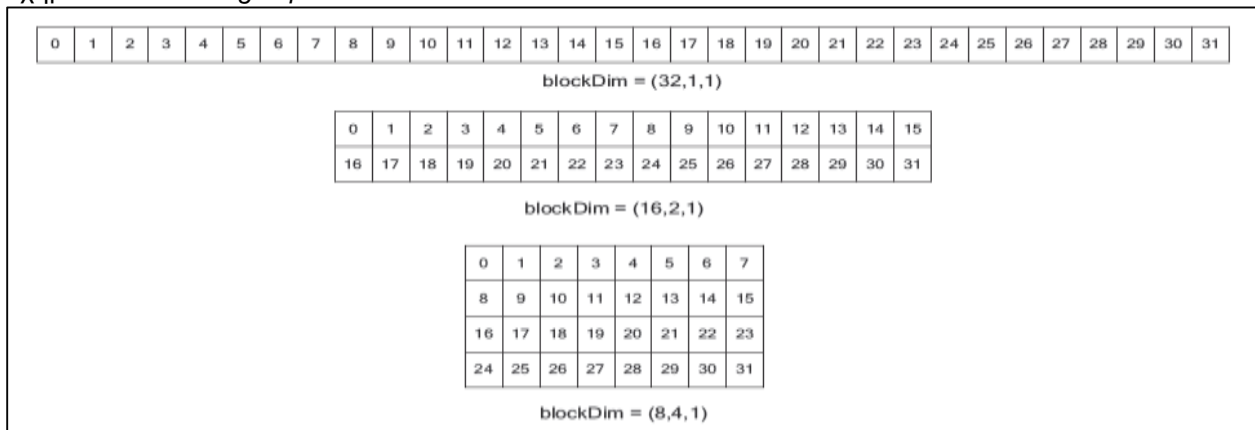
Σχήμα 3.5: Ένα 2D grid με 2D thread block.



Σχήμα 3.6: Ένα 2D grid με 2D thread block.



Σχήμα 3.7: Ένα 3D grid με 3D thread block.

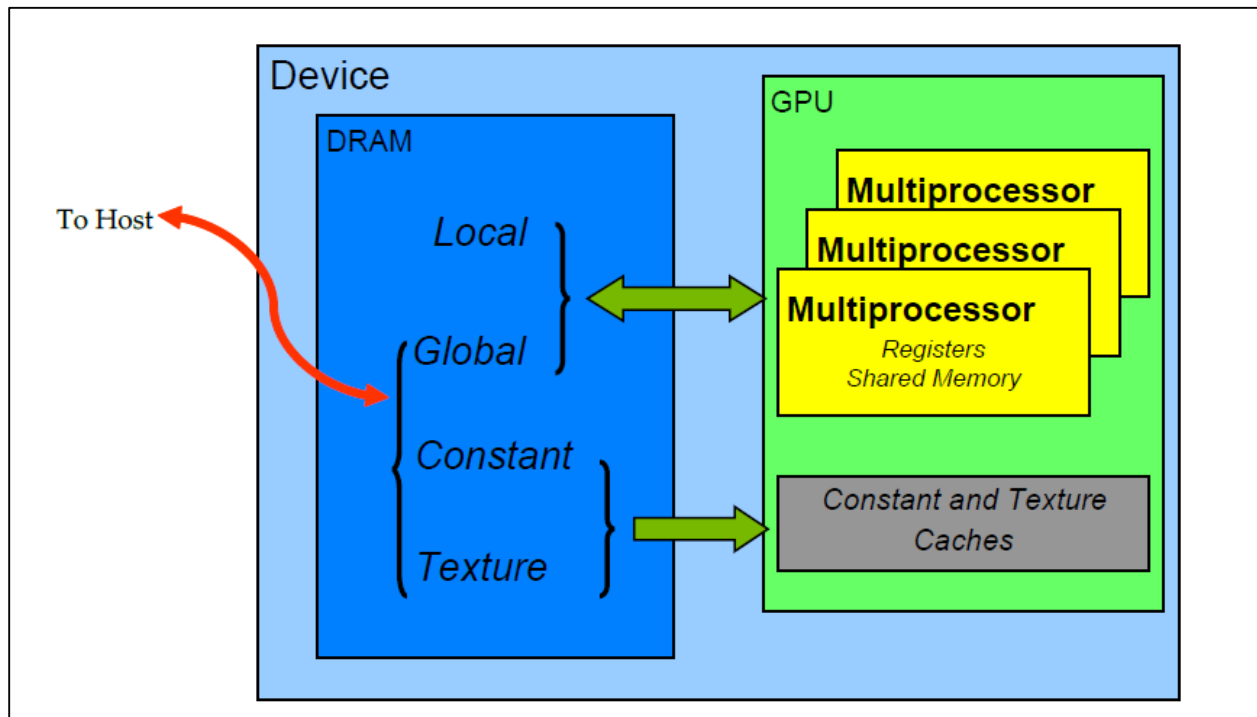


Σχήμα 3.8: Blocks αποτελούμενα από 32 threads. Πρώτο: 1D block. Δεύτερο και τρίτο: 2D block.

3.1.3 Τύποι μνήμης

Η κάρτα γραφικών δεν μπορεί να διαβάσει ή να γράψει άμεσα στην μνήμη του υπολογιστή. Αρχικά θα πρέπει να αναφέρουμε τους τέσσερις φυσικούς τύπους μνήμης και στην συνέχεια τρεις ειδικές υλοποιήσεις φυσικών μνημών:

1. Την καθολική μνήμη (Global memory).
2. Την κοινόχρηστη μνήμη (Shared memory).
3. Την τοπική μνήμη (Local Memory).
4. Τους καταχωρητές (Registers).
5. Τη μνήμη σταθερών (Constant Memory).
6. Την μνήμη υφής (Texture Memory).



Σχήμα 3.9: Τύποι μνήμης στην συσκευή GPU.

1. Η καθολική μνήμη (Global memory)

Η καθολική μνήμη ή αλλιώς “Global memory” είναι η πιο βασική και είναι μέρος της μνήμης DRAM της συσκευής. Θα μπορούσε κανείς να την αντιστοιχήσει με την μνήμη RAM του υπολογιστή. Τα δεδομένα που θα χρησιμοποιηθούν πρέπει να μεταφερθούν στην καθολική μνήμη. Είναι η πιο μεγάλη μνήμη σε χωρητικότητα αλλά και η πιο αργή. Σε σύγκριση με την μνήμη RAM του υπολογιστή όμως έχει σχεδόν την δεκαπλάσια ταχύτητα (~ 120Gbyte/s). Για να δει η κάρτα γραφικών τα δεδομένα πρέπει να μεταφερθούν στην καθολική μνήμη πρώτα και αντίστοιχα, αν πρέπει να εξαχθούν σε ένα αρχείο ή την οθόνη, πρέπει να μεταφερθούν πίσω. Εδώ, ήδη διαπιστώνεται ένα πρόβλημα, αυτή η επικοινωνία είναι ακριβή, καθώς ο δίαυλος που ενώνει αυτά τα δύο είναι συγκριτικά πολύ αργός. Πρέπει να μην μεταφερθούν άσκοπα δεδομένα. Αν, όπως στην περίπτωση μας, υπάρχει ένα πλέγμα από το οποίο χρειάζονται μόνο δύο ποσότητες, δεν θα μεταφερθεί όλο το πλέγμα, αλλά θα υπολογιστούν αυτές οι ποσότητες μέσα στην κάρτα, ακόμα και αν κάτι τέτοιο δεν είναι αποδοτικό (π.χ. μπορεί να γίνει μόνο σειριακά ή με πολύ λίγα νήματα). Από τη στιγμή που υπάρχουν τα δεδομένα στην καθολική μνήμη, όλα τα νήματα μπορούν να διαβάσουν και να γράψουν σε αυτή. Η καθολική μνήμη είναι προσβάσιμη με συναλλαγές των 32, 64 ή 128 byte από τα warps. Μία διαδικασία ανάγνωσης ή εγγραφής από ή προς την καθολική

μνήμη μπορεί να μεταφέρει ως και 128 byte σε μία συναλλαγή. Πρέπει να σημειωθεί ότι αυτό το σκέλος, παρόλο που φαίνεται ως λεπτομέρεια, είναι ένα από τα βασικά χαρακτηριστικά για ένα αποτελεσματικό πρόγραμμα. Η καθολική μνήμη δεν είναι απλά αργή, έχει και μεγάλη καθυστέρηση της τάξης των 500-700 κύκλων.

Στην αρχιτεκτονική Fermi τα πράγματα αλλάζουν πολύ, καθώς πλέον η καθολική μνήμη μπορεί να είναι cached, επομένως μία ανάγνωση μπορεί να κοστίζει λίγο, όσο ένας κύκλος του ρολογιού του πολυεπεξεργαστή. Οι προσβάσεις στην μνήμη από τα νήματα ενός warp πρέπει να είναι ευθυγραμμισμένες με αυτά τα μεγέθη (ανάλογα το μέγεθος της λέξης), δηλαδή η θέση της μνήμης που θα διαβάσει το πρώτο νήμα να είναι πολλαπλάσιο αυτών. Όταν τα νήματα ενός warp δίνουν μία εντολή ανάγνωσης από την καθολική μνήμη, το warp θα προσπαθήσει να οργανώσει αυτές τις εντολές για κάθε νήμα. Σκοπός είναι με όσο το δυνατόν λιγότερες συναλλαγές να μεταφερθούν όλα τα απαιτούμενα δεδομένα.

2. Η κοινόχρηστη μνήμη (Shared memory)

Η κοινόχρηστη μνήμη ή αλλιώς Shared memory αποτελεί μνήμη η οποία βρίσκεται στα κυκλώματα του κάθε πολυεπεξεργαστή (on chip). Είναι η πιο γρήγορη μνήμη (Tbyte/s) αλλά είναι πολύ περιορισμένη σε μέγεθος. Έχει συνολική χωρητικότητα μόλις 16 Kbyte σε κάθε πολυεπεξεργαστή ενώ στην αρχιτεκτονική Fermi μπορεί να φτάσει τα 48 Kbyte και έχει την ζωή του block στο οποίο ορίζεται. Η κοινόχρηστη μνήμη είναι ορατή μόνο στα νήματα ενός block. Νήματα ενός άλλου block δεν μπορούν να διαβάσουν ή να γράψουν στην κοινόχρηστη μνήμη που είναι δεσμευμένη σε ένα άλλο block. Πώς χωρίζεται τότε αυτή η μνήμη;

Η κοινόχρηστη μνήμη μπορεί να καταμεριστεί όπως είναι απαραίτητο για κάθε περίπτωση σε block. Η κοινόχρηστη μνήμη βέβαια, κατανέμεται ισόποσα και καθολικά, επομένως ορίζεται μόνον πόση από αυτή την μνήμη θα χρησιμοποιεί ένα block, κάθε block τότε θα έχει αυτή την διαμόρφωση. Ένας περιορισμός είναι ότι αν οριστεί 16 Kbyte κοινόχρηστης μνήμης σε κάθε block τότε κάθε επεξεργαστής δεν θα μπορεί να τρέξει πάνω από ένα block ταυτόχρονα. Αυτό θα δημιουργήσει προβλήματα με την πληρότητα του κάθε πολυεπεξεργαστή. Η πληρότητα ενός πολυεπεξεργαστή ορίζεται ως ο λόγος των ενεργών warps ενός πολυεπεξεργαστή ως προς τον μέγιστο δυνατό αριθμό ενεργών warps ανά πολυεπεξεργαστή, και θα αναλύσουμε την σημασία του αργότερα. Εκτός του καταμερισμού της κοινόχρηστης μνήμης ανά block υπάρχει και ένας άλλος καταμερισμός ο οποίος βασίζεται στην αρχιτεκτονική. Η κοινόχρηστη μνήμη χωρίζεται φυσικά σε πολλές ομάδες των δεκαέξι τραπεζών με την χωρητικότητα της κάθε τράπεζας να είναι 4byte.

Η κοινόχρηστη μνήμη ορίζεται μόνο στο εσωτερικό ενός block και επομένως ένα νήμα από άλλο block δεν μπορεί να δει την κοινόχρηστη μνήμη ενός άλλου block. Δεν είναι κοινόχρηστη για όλα τα νήματα, για αυτή την δουλειά υπάρχει η καθολική μνήμη. Επομένως ως μνήμη έχει την ζωή ενός block. Τα νήματα ενός warp πρέπει να συναλλάσσονται με μία ομάδα τραπεζών με ανεξάρτητο και γραμμικό τρόπο, με την ζητούμενη διεύθυνση μνήμης του πρώτου νήματος να είναι η πρώτη τράπεζα και του τελευταίου νήματος να είναι η τελευταία τράπεζα. Αν αυτή η συνθήκη ικανοποιείται, τότε η συναλλαγή θα ολοκληρωθεί ταυτόχρονα και για τα δεκαέξι νήματα.

3. Η τοπική μνήμη (Local memory)

Η τοπική μνήμη είναι η προσωπική μνήμη κάθε νήματος. Κανένα άλλο νήμα δεν έχει πρόσβαση σε αυτή, επομένως οι συναλλαγές είναι πάντα γραμμικές και ευθυγραμμισμένες. Η τοπική μνήμη βρίσκεται στην DRAM της κάρτας γραφικών. Έχει, επομένως, την ίδια ταχύτητα και υστέρηση με την καθολική μνήμη. Εδώ δημιουργείται ένα ερώτημα, γιατί το κάθε νήμα να έχει αργή μνήμη; Η ταχύτητα μίας τοπικής μνήμης θα πρέπει να είναι της τάξης της κοινόχρηστης. Ουσιαστικά πάρα πολύ σπάνια χρησιμοποιείται αυτή η μνήμη, γιατί δεν χρειάζεται σχεδόν ποτέ.

Οι πολυεπεξεργαστές έχουν μία μορφή μνήμης, τους καταχωρητές, η οποία παίζει τον ρόλο της τοπικής μνήμης. Ουσιαστικά η τοπική μνήμη είναι η τελευταία διέξοδος όταν οι απαιτήσεις σε μνήμη κάθε νήματος είναι πολύ μεγάλες. Η τοπική μνήμη δεν ορίζεται. Όποτε αφιερώνεται μνήμη σε κάθε νήμα, ο μεταγλωττιστής όταν ερμηνεύει την εντολή θα δεσμεύσει έναν ή περισσότερους καταχωρητές. Τοπική μνήμη θα δεσμευτεί εφόσον περάσουμε το όριο των διαθέσιμων από την αρχιτεκτονική καταχωρητών ή αν δεσμεύσουμε πίνακα στο νήμα που είναι μεγάλος ή ο μεταγλωττιστής δεν μπορεί να καταλάβει αν θα είναι σταθερός ή/και μικρός σε μέγεθος. Η μεταφορά δέσμευσης της μνήμης από τους καταχωρητές στην τοπική μνήμη ονομάζεται "spilling" και έχει δραματικές επιπτώσεις στην επίδοση ενός πολυεπεξεργαστή. Το γιατί όμως κάποιος να θέλει να πιέσει τα όρια της μνήμης καταχωρητών δημιουργώντας άμεσο κίνδυνο να κάνει το πρόγραμμα του πολύ πιο αργό λόγω spilling θα το δούμε αμέσως αφού αναλύσουμε τον ρόλο και την λειτουργία των καταχωρητών.

4. Οι καταχωρητές (Registers)

Οι καταχωρητές είναι αυτοί που επωμίζονται τις απαιτήσεις σε μνήμη του κάθε νήματος. Οι καταχωρητές είναι μέρος των κυκλωμάτων κάθε πολυεπεξεργαστή και το μέγεθος τους εξαρτάται από την αρχιτεκτονική. Οι κάρτες γραφικών με αρχιτεκτονική 1.0 και 1.1 έχουν μόλις 8192 καταχωρητές των 32bit, ενώ στις αρχιτεκτονικές 1.2 και 1.3 υπάρχουν 16834 καταχωρητές των 32bit. Η ταχύτητα τους είναι εφάμιλλη της κοινόχρηστης μνήμης αφού ένα μισό της δύνης μπορεί να γράψει στους καταχωρητές παράλληλα σε μία συναλλαγή, αλλά χωρίς την πολυπλοκότητα που είδαμε στην καθολική και την κοινόχρηστη μνήμη. Εφόσον τα blocks (που είναι και ο κύριος δομικός λίθος για ένα πρόγραμμα σε κάρτα γραφικών) αποτελούνται από μία ομάδα νημάτων και αυτά απαιτούν μνήμη από τους καταχωρητές, θα μετράμε το πλήθος των καταχωρητών με βάση τα blocks. Επομένως, αν ένα block έχει μέγεθος 256 νήματα και το κάθε ένα απαιτεί 10 καταχωρητές, τότε θα χρειαστούμε συνολικά 2560 καταχωρητές. Λαμβάνοντας υπόψιν ότι το μέγιστο δυνατό πλήθος των ενεργών νημάτων κυμαίνεται από 768 ως 1024 και για την 2.x είναι 1536, δηλαδή τρία ως τέσσερα blocks με την δεδομένη διαμόρφωση, οι καταχωρητές δεν αποτελούν εμπόδιο.

Για παράδειγμα ένα πιο απαιτητικό πρόγραμμα που χρειάζεται 20 καταχωρητές ανά νήμα, τότε στις αρχιτεκτονικές 1.0 και 1.1 θα έχουμε λιγότερα από 768 νήματα ενεργά. Όμως δεν μπορούμε να έχουμε ελεύθερα νήματα σε έναν πολυεπεξεργαστή, πρέπει να αποτελούν ένα block. Επομένως, σε αυτές τις αρχιτεκτονικές θα μπορεί να εκτελείται μόλις ένα block με 256 νήματα. Αμέσως, η πληρότητα των πολυεπεξεργαστών πέφτει στο 33%, Στην περίπτωση των αρχιτεκτονικών 1.2 και 1.3 θα μπορούν να εκτελούνται μόνο 3 block με 768 νήματα από το μέγιστο στον 1024 νημάτων. Η μεγάλη χρήση καταχωρητών μπορεί να δημιουργήσει κάποια προβλήματα στον αριθμό των ενεργών νημάτων. Οι καταχωρητές χρησιμοποιούνται από τον μεταγλωττιστή αυτόματα και για την αποφυγή υπολογισμού παραστάσεων που χρησιμοποιούνται πάνω από μία φορά χωρίς να αλλάξει η τιμή τους στο ενδιάμεσο, κλασικό παράδειγμα είναι οι δείκτες (indexes) τις μνήμης. Αυτό, να μεν επιταχύνει κάπως ένα πρόγραμμα αλλά η διαφορά μπορεί να είναι πολύ μικρή.

5. Η μνήμη σταθερών (Constant memory)

Η μνήμη σταθερών είναι το αντίστοιχο του ορισμού const μνήμης ενός υπολογιστή. Η μνήμη, δηλαδή, περιέχει κάποια σταθερή μεταβλητή. Είναι σταθερή μόνο στο τμήμα του προγράμματος που εκτελείται από την κάρτα γραφικών. Εκτός αυτών των τμημάτων υπάρχει η δυνατότητα να μεταβληθεί μέσα στην ροή του προγράμματος, και η κάρτα γραφικών θα δει κανονικά τη νέα τιμή στο επόμενο τμήμα που θα χρησιμοποιηθεί. Θα ήταν χρήσιμο να δηλώνονται οι διαστάσεις ενός πλέγματος σε αυτήν την μνήμη. Η μνήμη σταθερών είναι και αυτή κομμάτι της DRAM της κάρτας γραφικών. Είναι όμως πολύ γρήγορη καθώς είναι cached, είναι βελτιστοποιημένη για

broadcasting (εκπομπή) και το κόστος ανάγνωσης της είναι αμελητέο εφόσον τηρούνται κάποιοι κανόνες. Όντας βελτιστοποιημένη για broadcasting μίας πληροφορίας σε όλα τα νήματα, αυτά θα πρέπει να διαβάζουν όλα την ίδια περιοχή μίας μεταβλητή της μνήμης σταθερών κάθε φορά. Δηλαδή, αν τα νήματα θα πρέπει να διαβάσουν την x διάσταση ενός πίνακα, τότε η πληροφορία θα μεταδοθεί σε όλα τα νήματα με συνήθως μηδενικό κόστος. Αν όμως, τα μισά πρέπει να διαβάσουν την x διάσταση και τα άλλα μισά την y διάσταση, τότε το κόστος θα είναι ισοδύναμο με ανάγνωση από την καθολική μνήμη. Το μέγεθος της είναι μόλις 8 Kbyte για όλη την κάρτα γραφικών.

6. Η μνήμη υφής (Texture memory)

Η μνήμη υφής είναι αρκετά διαφορετική από τις μνήμες που γνωρίζουμε. Οι πολυεπεξεργαστές διαθέτουν ειδικά κυκλώματα για την επεξεργασία υφών στα γραφικά. Μέσω της μνήμης υφής έχουμε τη δυνατότητα να χρησιμοποιούνται αυτά τα κυκλώματα ως προς όφελος μας. Η μνήμη υφής λοιπόν αποτελείται από καθολική μνήμη την οποία "δένουμε" πάνω στα κυκλώματα υφής. Η μνήμη υφής είναι ορατή σε όλα τα νήματα, αλλά σε αντίθεση με την καθολική, μπορούν μόνο να την διαβάσουν μέσω των κυκλωμάτων αυτών. Η μνήμη υφής είναι cached και βελτιστοποιημένη για αναγνώσεις. Με το πέρας της εκτέλεσης ενός τμήματος της κάρτας γραφικών η μνήμη υφής θα ανανεωθεί αυτόματα και σε νέα κλήση ενός τμήματος (συνάρτησης) της κάρτας γραφικών θα διαβαστούν νέες τιμές.

Η μνήμη υφής έχει διαστάσεις, επομένως μπορεί να οριστεί ως μονοδιάστατη, διδιάστατη ή τριδιάστατη και η προσπέλαση της να γίνεται από τον αντίστοιχο αριθμό συντεταγμένων. Το κάθε στοιχείο ονομάζεται texel (texture element) και η θέση του υπολογίζεται από ειδικά κυκλώματα του πολυεπεξεργαστή. Επίσης, υποστηρίζει και μία άλλη μορφή μεταβλητών που ορίζονται στις κάρτες γραφικών, τα διανύσματα. Ένα texel τελικά μπορεί να επιστέφει μία τιμή ή ένα διάνυσμα διάστασης δύο ή τέσσερα. Μία υφή μπορεί να είναι δεμένη σε ένα κλασσικό γραμμικό κομμάτι μνήμης ή σε έναν συμπαγή τύπο πίνακα που ονομάζεται CUDA array. Στην περίπτωση που είναι δεμένη σε γραμμικό κομμάτι μνήμης, αν γράψουμε σε αυτό το κομμάτι μέσα σε ένα τμήμα του προγράμματος της κάρτας γραφικών, η υφή θα ενημερωθεί αυτόματα με το πέρας αυτού. Τα πράγματα δεν είναι τόσο απλά όμως και για τα CUDA Arrays, σε αυτά δεν έχουμε πρόσβαση εγγραφής παρά μόνο μέσω ειδικών συναρτήσεων εντός του τμήματος του προγράμματος που τρέχει στον κεντρικό επεξεργαστή. Ο λόγος είναι ότι τα CUDA Arrays δεν μπορούμε να τα δούμε ως γραμμική μνήμη. Τα δεδομένα μέσα στα συμπαγή CUDA Arrays είναι κατανομημένα με ειδικούς αλγορίθμους, ώστε να επιτυγχάνεται χωρικός εντοπισμός. Αν φανταστούμε έναν μικρό κύβο και το κέντρο του, τα στοιχεία θα είναι υποθηκευμένα έτσι ώστε πάντα τα στοιχεία γύρω από αυτό το κέντρο να είναι κοντά σε αυτό από την σκοπιά της γενικής μονοδιάστατης αναπαράστασης.

Για να το εξηγηθεί αυτό καλύτερα πρέπει να αναφερθεί, ότι πάντα κάθε μνήμη σε υπολογιστή και σε κάρτα γραφικών, είναι μονοδιάστατη. Αν ορίσουμε έναν πίνακα τριών διαστάσεων με N_x, N_y, N_z την κάθε μία και τοποθετηθεί για κάθε z όλα τα (y, z) και θα κάθε y όλα τα (x, y) τότε ο μετασχηματισμός από τις εικονικές τρεις διαστάσεις στην "φυσική" μία θα είναι:

$$x + N_x * (y + N_y * z)$$

Εδώ παρατηρείται ότι ακόμα και για μικρές διαστάσεις του πίνακα, από την μονοδιάστατη σκοπιά, ένα στοιχείο θα είναι πάντα μακριά από ένα σε θέση $y + 1$ ή ειδικότερα $z + 1$. Μία από τις ιδιότητες της μνήμης υφής είναι ότι δεν περιορίζεται στην προσπέλαση μέσω των "κλασσικών" συντεταγμένων όπως συμβαίνει με όλες τις μνήμες. Μπορεί να προσπελαστεί με αριθμούς κινητής υποδιαστολής (float) ως συντεταγμένες, ακόμα και κανονικοποιημένους ως προς την διάσταση. Όταν χρησιμοποιηθούν αριθμοί κινητής υποδιαστολής ως συντεταγμένες, τότε η θέση ενός texel είναι το κέντρο του. Δηλαδή, το στοιχείο που έχει συντεταγμένες (x, y) υπό κανονικές συνθήκες, τώρα θα έχει συντεταγμένες $(x + 0, 5, y + 0, 5)$. Μην ξεχνάμε ότι οι κάρτες γραφικών

φτιάχτηκαν για γραφικά και εξαιτίας αυτού στα κυκλώματα υφής, υπάρχουν ολοκληρωμένα που αναλαμβάνουν να εκτελούν γραμμικές παρεμβολές (linear interpolation).

Οπότε εδώ παρουσιάζονται κάποια βασικά πράγματα.

1. Στην καθολική μνήμη έχουν πρόσβαση ανάγνωσης και εγγραφής όλα τα νήματα.
2. Στην κοινόχρηστη μνήμη έχουν πρόσβαση ανάγνωσης και εγγραφής μόνο τα νήματα του block στο οποίο αυτή ορίζεται. Έχει την ζωή του block.
3. Στην τοπική μνήμη και στους καταχωρητές έχει πρόσβαση ανάγνωσης και εγγραφής μόνο το νήμα στο οποίο αυτή είναι ορισμένη. Έχει την ζωή του νήματος.
4. Στην μνήμη σταθερών έχουν πρόσβαση ανάγνωσης όλα τα νήματα.
5. Στην μνήμη υφής έχουν πρόσβαση ανάγνωσης όλα τα νήματα.

3.1.4 Ασύγχρονη και ταυτόχρονη εκτέλεση

Ταυτόχρονη εκτέλεση μεταξύ CPU και GPU

Πολλές από τις συναρτήσεις τις CUDA που τρέχουν σε μία κάρτα γραφικών, είναι ασύγχρονες ως προς τον κεντρικό επεξεργαστή. Με το που ξεκινήσει η εκτέλεση μίας τέτοια συνάρτησης, ο έλεγχος επιστρέφει αμέσως πίσω στον κεντρικό επεξεργαστή. Αυτό με λίγα λόγια σημαίνει ότι ο κεντρικός επεξεργαστής δεν την περιμένει να ολοκληρωθεί για να συνεχίσει να εκτελεί τις επόμενες εντολές όπως συμβαίνει στα συμβατικά προγράμματα. Στο σχήμα 3.10 φαίνεται η γενική ιδέα του ετερογενούς προγραμματισμού, αρκεί να φανταστούμε ότι το κομμάτι που εκτελείται στην κάρτα γραφικών (device) επικαλύπτει το τμήμα που εκτελείται αμέσως μετά από τον κεντρικό επεξεργαστή (host) έως ότου ο τελευταίος βρεθεί αντιμέτωπος με μία άλλη κλήση συνάρτησης της κάρτας γραφικών ή σε μία από τις ειδικές συναρτήσεις που επιβάλλουν συγχρονισμό μεταξύ των δύο συσκευών. Συναρτήσεις με τέτοια ασύγχρονη συμπεριφορά ως προς τον κεντρικό επεξεργαστή είναι:

Οι συναρτήσεις κλήσης προγράμματος προς την κάρτα γραφικών (kernels)

Οι μεταφορές/αντιγραφές μνήμης εσωτερικά στην κάρτα γραφικών

Οι συναρτήσεις που δεσμεύουν μνήμη στην κάρτα γραφικών

Μία κατηγορία συναρτήσεων μεταφοράς μνήμης μεταξύ επεξεργαστή - κάρτας γραφικών, που ονομάζονται async.

Αλληλοεπικάλυψη μεταφορών μνήμης και εκτέλεσης συναρτήσεων kernel

Μερικές συσκευές με υπολογιστική ικανότητα 1.1 και άνω, έχουν την δυνατότητα να μεταφέρουν δεδομένα μεταξύ κλειδωμένης μνήμης συστήματος και την μνήμη της συσκευής DRAM. Αυτός ο τρόπος μεταφοράς υποστηρίζεται μόνο για γραμμική μνήμη (όχι CUDA Arrays).

Ταυτόχρονη εκτέλεση συναρτήσεων kernel

Μερικές συσκευές υπολογιστικής ικανότητας 2.0 (αρχιτεκτονική Fermi) μπορούν να εκτελέσουν πάνω από μία συναρτήσεις εκτέλεσης δεδομένων (kernel) ταυτόχρονα, έως τέσσερις συγκεκριμένα. Το πόσες συναρτήσεις θα "φορτωθούν" και θα εκτελεστούν ταυτόχρονα στην κάρτα γραφικών εξαρτάται από τις απαιτήσεις τους. Εδώ τίθεται άλλος ένας περιορισμός, όπως έχουμε αναφέρει, η κάρτα γραφικών ελέγχεται από τον κεντρικό επεξεργαστή. Πιο συγκεκριμένα όταν ένα νήμα του κεντρικού επεξεργαστή εκτελεί την πρώτη συνάρτηση που σχετίζεται με την κάρτα γραφικών (πχ για μεταφορά μνήμης), τότε αυτό το νήμα δημιουργεί ένα "πλαίσιο" (context) με την κάρτα γραφικών. Οι συναρτήσεις kernel δεν πρέπει να ανήκουν στο ίδιο πλαίσιο για να μπορούν να εκτελεστούν ταυτόχρονα. Δηλαδή, αν εκτελούνται τέσσερα ανεξάρτητα προγράμματα που χρησιμοποιούν την κάρτα γραφικών τότε οι συναρτήσεις kernel τους πιθανό να μπορέσουν να εκτελεστούν ταυτόχρονα. Αντίθετα σε ένα πρόγραμμα δεν μπορούν να εκτελεστούν τέσσερις συναρτήσεις kernel ταυτόχρονα εκτός και αν χρησιμοποιηθεί κάποια πολυνηματική βιβλιοθήκη

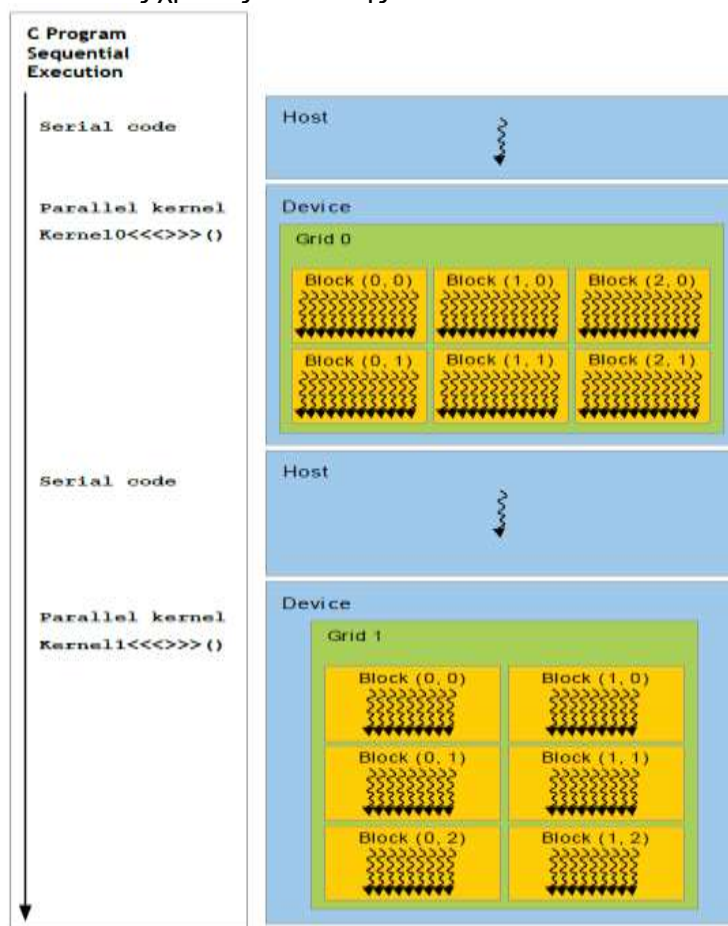
όπως το OpenMP, όπου αν οριστούν 4 νήματα και καθένα δημιουργήσει το δικό της πλαίσιο με την κάρτα, οι συναρτήσεις kernel θα μπορούν να εκτελεστούν ταυτόχρονα. Σε όλες τις παλαιότερες αρχιτεκτονικές, σε όλες τις περιπτώσεις μία συνάρτηση kernel θα περιμένει πάντα την προηγούμενη διεργασία στην κάρτα γραφικών να τελειώσει, ανεξάρτητα αν είναι από άλλο πρόγραμμα ή από το ίδιο αλλά από διαφορετικό πλαίσιο.

Ταυτόχρονες μεταφορές μνήμης

Οι συσκευές με υπολογιστική ικανότητα 2.0 μπορούν να εκτελέσουν ταυτόχρονα μία αντιγραφή μνήμης από κλειδωμένη μνήμη συστήματος στην μνήμη της κάρτας γραφικών μαζί με μία αντιγραφή μνήμης από την μνήμη σε κλειδωμένη μνήμη συστήματος.

Τα ρεύματα (streams)

Ένα ρεύμα είναι ένα σύνολο εντολών οι οποίες εκτελούνται σε σειρά. Διαφορετικά ρεύματα όμως θα μπορούν να εκτελέσουν τις εντολές τους εκτός σειράς ή μία ως προς την άλλη, ή ακόμα και παράλληλα. Τώρα που αναφέρθηκε η ταυτόχρονη μεταφορά μνήμη και εκτέλεσης αλλά και τα ρεύματα, μπορεί να αναλυθεί αυτό το σχήμα καλύτερα. Στην πρώτη περίπτωση, πρώτα εκτελείται μία συνάρτηση kernel και μετά γίνεται η αντιγραφή των δεδομένων από την καθολική μνήμη πίσω στην κλειδωμένη μνήμη συστήματος. Αν χωριστεί η συνάρτηση kernel σε τέσσερα ανεξάρτητα τμήματα και τα οριστούν ως διαφορετικά ρεύματα, τότε μπορούν να επιστρέψουν το αποτέλεσμα της κάθε συνάρτησης kernel ενώ εκτελείται η συνάρτηση kernel του επόμενου ρεύματος. Έτσι μπορεί να μειωθεί ο συνολικός χρόνος εκτέλεσης.



Σχήμα 3.10: Εκτέλεση κώδικα σειριακά στην CPU και εκτέλεση τμήματος του κώδικα παράλληλα στην GPU.

3.1.5 Πληρότητα των πολυεπεξεργαστών και τεχνικά χαρακτηριστικά

Αναφέρθηκε στην υποενότητα 3.1.3 η πληρότητα των πολυεπεξεργαστών. Η πληρότητα ενός πολυεπεξεργαστή ορίζεται ως:

$$\frac{Warps_{effective}}{Warps_{maximum}}$$

όπου warps είναι οι ομάδες των 32 threads. $Warps_{effective}$ είναι ο αριθμός των ενεργών warps και $Warps_{maximum}$ ο μέγιστος αριθμός των ενεργών warps που ορίζεται από τα τεχνικά χαρακτηριστικά της κάρτας γραφικών. Ο αριθμός των ενεργών warps εξαρτάται από της απαιτήσεις και την παραμετροποίηση του προγράμματος μας:

Το μέγεθος μνήμης καταχωρητών ανά νήμα.

Το μέγεθος της κοινόχρηστης μνήμης ανά block.

Το μέγεθος ενός block (σε νήματα).

Το μέγιστο πλήθος νημάτων ανά πολυεπεξεργαστή.

Την υπολογιστική ικανότητα της κάρτας γραφικών (που σχετίζεται και με τα δύο πρώτα).

Για παράδειγμα μία κάρτα γραφικών με υπολογιστική ικανότητα 1.3. Κάποιος μπορεί να κάνει τους ίδιους υπολογισμούς παίρνοντας τις παραμέτρους από τους πίνακες 3.2 και 3.3 για οποιαδήποτε άλλη κάρτα γραφικών αρχιτεκτονικής CUDA.

Θεωρώντας ότι έχουμε ένα πρόγραμμα παραμετροποιημένο έτσι ώστε ένα block να έχει διαστάσεις 16*16, δηλαδή 256 νήματα. Έχοντας ορίσει το μέγεθος της κοινόχρηστης μνήμης να είναι 1024 στοιχεία ακεραίων (4byte το καθένα) και στην συνέχεια μετρώντας πόσες μεταβλητές δεν είναι ορισμένες ως κοινόχρηστες και δεν είναι ορίσματα της συνάρτησης υπάρχουν μέσα στην συνάρτηση. Αυτό μας δίνει μία γενική ιδέα του πλήθους των καταχωρητών. Στην υποενότητα 3.1.3 αναφέρθηκε ότι αυτό δεν είναι αρκετό, καθώς τι πάει στους καταχωρητές και τι όχι ελέγχεται από τον μεταγλωττιστή. Παρόλα αυτά, για τώρα θα θεωρηθεί ότι καταχωρητές είναι όσοι αναφέρθηκαν παραπάνω +4. Ας πούμε ότι μετρήθηκαν 14 + 4. Επομένως υπάρχουν 18 καταχωρητές.

Σε κάρτα γραφικών με υπολογιστική ικανότητα 1.3 και πάνω, το μέγιστο πλήθος των ενεργών νημάτων ανά πολυεπεξεργαστή είναι 1024. Επομένως, λαμβάνοντας μόνο αυτή την παράμετρο υπόψη, θα μπορούν να φορτωθούν 4 block στον πολυεπεξεργαστή. Το όριο των blocks που μπορούν να επεξεργαστούν παράλληλα από έναν πολυεπεξεργαστή είναι 8. Επομένως, δεν έχουμε νήματα τα οποία δεν "χωράνε" στον πολυεπεξεργαστή. Στην κοινόχρηστη μνήμη. Ορίστηκαν 1024 στοιχεία ανά block των 4byte (integer, float), επομένως 4Kbyte κοινόχρηστης μνήμης χρειάζονται για κάθε block. Η διαθέσιμη μνήμη σε έναν πολυεπεξεργαστή είναι 16kbyte, επομένως λαμβάνοντας μόνο αυτή την παράμετρο υπόψη, μπορούν να φορτωθούν 4 blocks στον πολυεπεξεργαστή. Το πλήθος 32bit καταχωρητών ανά πολυεπεξεργαστή είναι 16.384. Εδώ πρέπει να πάρουμε τα προηγούμενα αποτελέσματα, αν τα συνδυάσουμε βρίσκουμε ότι μπορούν φορτωθούν 4 blocks. Τα 4 αυτά blocks περιέχουν 1024 νήματα και το καθένα θέλει 18 καταχωρητές. Επομένως, $18 * 1024 > 16.384$.

Δεν χρειάζεται πραγματικά να κάνουμε όλη αυτή την ανάλυση. Αυτό που χρειάζεται όμως είναι να έχουμε μία γενική ιδέα του πως θα επιλέξουμε τις παραμέτρους για τον αλγόριθμό μας. Τις λεπτομέρειες θα της συμπεριλάβουμε αφού είμαστε σε σημείο να μεταγλωττίσουμε το πρόγραμμα και επομένως ο μεταγλωττιστής να μας πει ακριβώς πόσοι καταχωρητές χρειάστηκαν. Τότε θα

βάλουμε όλα τα δεδομένα στον υπολογιστή πληρότητας και θα μας πει την πραγματική πληρότητα και από κει θα κάνουμε ότι αλλαγές είναι απαραίτητες.

Στον πίνακα 3.2 αναφέρονται όλα τα τεχνικά χαρακτηριστικά κάθε υπολογιστικής ικανότητας. Αξίζει να σημειωθεί μία επίσης πολύ σημαντική διαφορά μεταξύ των καρτών γραφικών υπολογιστικής ικανότητας 1.1 και κάτω με τις κάρτες γραφικών υπολογιστικής ικανότητας 1.2 και άνω. Οι δεύτερες υποστηρίζουν αριθμούς κινητής υποδιαστολής διπλής ακρίβειας (double ή 64bit float).

Στον πίνακα 3.3 αναφέρονται μερικά από τα γενικότερα τεχνικά χαρακτηριστικά (υπολογιστική ικανότητα, αριθμός πολυεπεξεργαστών και συνολικός αριθμός πυρήνων) για μερικές από τις πιο ευρέως διαθέσιμες κάρτες γραφικών αρχιτεκτονικής CUDA.

GPU	G80	GT200	Fermi
Transistors	681 million	1.4 billion	3.0 billion
CUDA Cores	128	240	512
Double Precision Floating Point Capability	None	30 FMA ops / clock	256 FMA ops /clock
Single Precision Floating Point Capability	128 MAD ops/clock	240 MAD ops / clock	512 FMA ops /clock
Special Function Units (SFUs) / SM	2	2	4
Warp schedulers (per SM)	1	1	2
Shared Memory (per SM)	16 KB	16 KB	Configurable 48 KB or 16 KB
L1 Cache (per SM)	None	None	Configurable 16 KB or 48 KB
L2 Cache	None	None	768 KB
ECC Memory Support	No	No	Yes
Concurrent Kernels	No	No	Up to 16
Load/Store Address Width	32-bit	32-bit	64-bit

Πίνακας 3.2: Τεχνικές δυνατότητες τριών διαφορετικών καρτών γραφικών. G80 και GT200 είναι αρχιτεκτονικής Tesla.

	FERMI GF100	FERMI GF104	KEPLER GK104	KEPLER GK110	KEPLER GK210
Compute Capability	2.0	2.1	3.0	3.5	3.7
Threads / Warp	32				
Max Threads / Thread Block	1024				
Max Warps / Multiprocessor	48		64		
Max Threads / Multiprocessor	1536		2048		
Max Thread Blocks / Multiprocessor	8		16		
32-bit Registers / Multiprocessor	32768		65536		131072
Max Registers / Thread Block	32768		65536		65536
Max Registers / Thread	63			255	
Max Shared Memory / Multiprocessor	48K				112K
Max Shared Memory / Thread Block	48K				
Max X Grid Dimension	2 ¹⁶ -1		2 ³² -1		
Hyper-Q	No			Yes	
Dynamic Parallelism	No			Yes	

Πίνακας 3.3: Τεχνικά χαρακτηριστικά της αρχιτεκτονικής Fermi και Tesla.

3.2 CUDA/C : Μία επέκταση των C/C++

Στη συνέχεια παρουσιάζεται ο τρόπος με τον οποίο προγραμματίζεται μία εφαρμογή στην μονάδα επεξεργασίας γραφικών.

3.2.1 Εισαγωγή στην γλώσσα προγραμματισμού CUDA/C

Η γλώσσα προγραμματισμού CUDA/C διαφέρει από τις C/C++ κυρίως στο ότι είναι γλώσσα ετερογενούς προγραμματισμού. Από τη μία πλευρά, ο κεντρικός επεξεργαστής (στην CUDA/C ονομάζεται host) εκτελεί σειριακό κώδικα και έχει πρόσβαση στη μνήμη όπως στα κλασικά προγράμματα, και από την άλλη η μονάδα επεξεργασίας γραφικών (device) εκτελεί παράλληλο κώδικα και έχει ειδική πρόσβαση σε περιοχές της μνήμης. Αυτό το είδος προγραμματισμού το είδαμε στο σχήμα 3.10. Η γλώσσα CUDA/C απευθύνεται μόνο σε κάρτες γραφικών της εταιρίας NVIDIA. Η έννοια του ετερογενούς προγραμματισμού δεν περιορίζεται στην CUDA/C, υπάρχουν αρκετές άλλες γλώσσες προγραμματισμού, όπως η OpenCL, η Direct Compute και η Brook++, αλλά και γλώσσες προγραμματισμού που δεν απευθύνονται καν σε κάρτες γραφικών αλλά σε εξειδικευμένα κυκλώματα FPGA σε παράλληλη χρήση με τον κεντρικό επεξεργαστή.

Το CUDA API είναι ένα σύνολο συναρτήσεων και ορισμών που αναλαμβάνουν τον συνδυαστικό ρόλο ανάμεσα στον κεντρικό επεξεργαστή και την κάρτα γραφικών. Υπάρχουν τρία είδη του API, το API εκτέλεσης σε χαμηλό επίπεδο, το API εκτέλεσης σε υψηλό επίπεδο και τέλος το API εκτέλεσης σε επίπεδο οδηγού της κάρτας. Τα δύο πρώτα διαφέρουν ελάχιστα στον προγραμματισμό, ουσιαστικά είναι τα ίδια, με το δεύτερο να δίνει κάποιες από της δυνατότητας της C++ στον κώδικα για την κάρτα γραφικών. Το API εκτέλεσης σε επίπεδο οδηγού (driver) της κάρτας γραφικών είναι πολύ διαφορετικό και δεν θα μας απασχολήσει καθόλου καθώς είναι αρκετά πιο πολύπλοκο και χρησιμοποιείται μόνο σε ειδικές περιπτώσεις. Εν τέλει, η CUDA/C ελάχιστα διαφέρει από την C στο μεγαλύτερο κομμάτι της, γεγονός που την κάνει εξαιρετικά εύκολη στην εκμάθηση αφήνοντας έτσι το κύριο μέρος της προσπάθειας εκμάθησης προγραμματισμού σε κάρτα γραφικών να είναι το πως επεξεργάζεται η κάρτα γραφικών τις εντολές μας και πως θα την παραμετροποιήσουμε σωστά για τις ανάγκες του προγράμματος μας.

3.2.2 Ειδικό ορισμοί συναρτήσεων

Αρχικά, θα ξεκινήσουμε με τους νέους τύπους μεταβλητών και της ειδικές συναρτήσεις που θα συναντήσουμε αρκετές φορές από εδώ και στο εξής. Η μεταβλητή τύπου `enum cudaMemcryKind kind` ορίζει την κατεύθυνση αντιγραφής της μνήμης. Χρησιμοποιείται μόνο στις ειδικές συναρτήσεις αντιγραφής δεδομένων. Οι επιλογές είναι:

Η μεταβλητή τύπου `enum cudaMemcryKindTokind`

`CudaMemcryHostToDevice`: Αντιγραφή από την μνήμη RAM στην καθολική μνήμη. Είναι σύγχρονη ως προς τον κεντρικό επεξεργαστή.

`CudaMemcryDeviceToHost`: Αντιγραφή από την καθολική μνήμη στην μνήμη Ram. Επίσης είναι σύγχρονη ως προς τον κεντρικό επεξεργαστή.

`CudaMemcryDeviceToDevice`: Εσωτερική αντιγραφή μεταξύ δύο δεικτών καθολικής μνήμης. Είναι ασύγχρονη ως προς τον κεντρικό επεξεργαστή.

Μεταβλητές τύπου `dim3` και η συνάρτηση `dim3()`

Οι μεταβλητές `dim3` είναι διανύσματα ακεραίων (32bit integer) με διάσταση 3. Βάση αυτών ορίζεται ένα από τα πιο βασικά μεγέθη στην CUDA, τις διαστάσεις των blocks και των grids. Οι μεταβλητές `dim3` παίρνουν τιμή μέσω της συνάρτησης: `dim3(int x, int y, int z)`. Είναι δυνατόν οι τιμές των στοιχείων του διανύσματος να δηλώνονται άμεσα αρκεί να ληφθεί υπόψη η αναπαράσταση της συνάρτησης:

```

struct dim3{
int x;
int y;
int z;
}

```

Θεωρώντας ότι έχουμε έναν πίνακα διάστασης (N_x, N_y) με δεδομένα τα οποία είναι χρήσιμο να μοιράζονται έτσι ώστε κάθε νήμα να επεξεργάζεται ένα στοιχείο. Έχοντας δει ως τώρα ότι γενικά έναν καλός αριθμός για το μέγεθος ενός block είναι 256 νήματα, θα το εφαρμοστεί και στο συγκεκριμένο παράδειγμα. Ξεκινάμε ορίζοντας το μέγεθος του block:

```
dim3 blocksize = dim3 ( 16, 16, 1 )
```

Ειδική ορισμοί συναρτήσεων

Εδώ είναι χρήσιμο να οριστούν και διαχωριστούν οι συναρτήσεις που αναφέρονται στην κάρτα γραφικών (device) και αυτές που αναφέρονται στον κεντρικό επεξεργαστή (host).

__device__ Μία συνάρτηση η οποία είναι ορισμένη ως **__device__**, εκτελείται στην κάρτα γραφικών και μπορεί να κληθεί από την κάρτα γραφικών μόνο.

__global__ Μία συνάρτηση η οποία είναι ορισμένη ως **__global__**, εκτελείται στην κάρτα γραφικών και μπορεί να κληθεί από τον κεντρικό επεξεργαστή μόνο.

__host__ Ως **__host__** ορίζεται μία συνάρτηση η οποία θα εκτελείται στον κεντρικό επεξεργαστή και θα μπορεί να κληθεί μόνο από αυτόν (οι συναρτήσεις που εκτελούνται στην κάρτα γραφικών εξορισμού δεν μπορούν να κάνουν κλήση συναρτήσεων που θα εκτελεστούν στον κεντρικό επεξεργαστή). Όταν μία συνάρτηση δεν περιέχει κανέναν από τους τρεις ορισμούς τότε θα εκτελείται όπως μία **__host__**. Εδώ ο ορισμός **__host__** φαίνεται να είναι άχρηστος. Στην ουσία υπάρχει για ευκολία, καθώς επιτρέπεται να χρησιμοποιηθεί πάνω από έναν ορισμό για μία συνάρτηση. Αν ορίσει μία συνάρτηση ως **__host__** και **__device__**, τότε ο μεταγλωττιστής θα την μεταγλωττίσει εις διπλούν, μία έκδοση για τον κεντρικό επεξεργαστή και μία έκδοση για την κάρτα γραφικών.

3.2.3 Ο kernel, οι ειδικές μεταβλητές του και οι συναρτήσεις συσκευής

Ο kernel είναι η καρδιά ενός προγράμματος για GPU. Cuda Kernels ονομάζονται οι κύριες συναρτήσεις της κάρτας γραφικών. Οι συναρτήσεις αυτές, δηλαδή, που αναλαμβάνουν να επεξεργαστούν τα δεδομένα μας. Ένας kernel εφόσον εκτελείται στην κάρτα γραφικών δεν έχει τη συνήθη πρόσβαση I/O που απολαμβάνουν οι κλασικές συναρτήσεις που τρέχουν στον κεντρικό επεξεργαστή. Οι συναρτήσεις kernel ορίζονται ως **__global__**, πρέπει να επιστρέφουν πάντα τιμή void και τα αρχεία που τις περιέχουν πρέπει να έχουν την κατάληξη .cu. Ας δούμε ένα παράδειγμα ορισμού:

```
__global__ void this_will_run_in_the_gpu ( float * input , float * output );
```

Οι συναρτήσεις kernel διαφέρουν από όλες τις άλλες συναρτήσεις στον τρόπο που καλούνται. Σε μία συνάρτηση kernel εκτός από τα κλασικά ορίσματα που περιέχει κάθε κλήση συνάρτησης, υπάρχουν και μερικά ειδικά ορίσματα με τις παραμέτρους του kernel. Οι παράμετροι του kernel είναι τα blocks, grids αλλά και η μέγιστη τιμή δυναμικά δεσμευμένης κοινόχρηστης μνήμης:

```
this_will_run_in_the_gpu <<< grids , blocks , max_dynamic_sharedmem >>> ( in , out );
```

Τα <<<...>>> αναλαμβάνουν να δώσουν τις παραμέτρους στον kernel μας. Στην πραγματικότητα, συνήθως η παράμετρος της δυναμικής κοινόχρηστης μνήμης δεν αναγράφεται καν ή απλά είναι μονάδα. Συνήθως ορίζεται το μέγεθος των block στατικά και κατά κανόνα η κοινόχρηστη μνήμη ανά block ορίζεται ως συνάρτηση του μεγέθους του block. Μία συνάρτηση kernel είναι ουσιαστικά το κομμάτι του κώδικα που θα τρέξει σε κάθε νήμα των πολυεπεξεργαστών. Είναι, δηλαδή, ένα

loop από το νήμα 0 ως το νήμα $grid_size * block_size - 1$ στην απλουστευμένη περίπτωση που το block έχει μόνο διάσταση x. Σε αυτό το loop όμως δεν υπάρχει ο ορισμός της τάξης. Τα νήματα δεν θα εκτελεστούν με αύξοντα αριθμό, αλλά σχεδόν τυχαία. Η κάρτα γραφικών θα αποφασίσει πώς θα τα εκτελέσει.

Ειδικές μεταβλητές

Μέσα σε κάθε kernel υπάρχουν μερικές ειδικές μεταβλητές οι οποίες ορίζονται από το σύστημα. Αυτές οι μεταβλητές δίνουν ανά πάσα στιγμή πιο νήμα ενός block εκτελείται, σε πιο block, ποια είναι η διάσταση του block και ποια η διάσταση του grid. Ας τις δούμε πιο αναλυτικά, αφού είναι βασικές για τον προγραμματισμό του kernel.

int3 threadIdx: Είναι ο δείκτης κάθε νήματος σε ένα block. Παίρνει τιμές $[0..blockDim - 1]$.

int3 blockIdx: Μας δίνει τον δείκτη κάθε block. Παίρνει τιμές $[0..gridDim - 1]$.

int3 blockDim: Μας δίνει τις διαστάσεις κάθε block. Η τιμή του είναι σταθερή αφού έχει οριστεί από τις παραμέτρους που περάσαμε στον kernel.

int3 gridDim: Μας δίνει τις διαστάσεις του grid, το πλήθος δηλαδή των blocks σε κάθε διάσταση. Η τιμή του επίσης είναι προκαθορισμένη από τις παραμέτρους του kernel.

Στην περίπτωση των δύο διαστάσεων θα υπολογιστεί ο καθολικά γραμμικός δείκτης ενός νήματος (οι προσπελάσεις στην μνήμη είναι γραμμικές). Ένας μετασχηματισμός από τις δύο διαστάσεις σε μία είναι ο εξής:

$$index1D = x + Nx * y$$

Το πρόβλημα θα γίνει πιο εύκολα κατανοητό με την βοήθεια των σχημάτων 3.11.

0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31	24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63	56	57	58	59	60	61	62	63
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31	24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	64	65	66	67	68	69	70	71
72	73	74	75	76	77	78	79	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	80	81	82	83	84	85	86	87
88	89	90	91	92	93	94	95	88	89	90	91	92	93	94	95
96	97	98	99	100	101	102	103	96	97	98	99	100	101	102	103
104	105	106	107	108	109	110	111	104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119	112	113	114	115	116	117	118	119
120	121	122	123	124	125	126	127	120	121	122	123	124	125	126	127
128	129	130	131	132	133	134	135	128	129	130	131	132	133	134	135
136	137	138	139	140	141	142	143	136	137	138	139	140	141	142	143
144	145	146	147	148	149	150	151	144	145	146	147	148	149	150	151
152	153	154	155	156	157	158	159	152	153	154	155	156	157	158	159
160	161	162	163	164	165	166	167	160	161	162	163	164	165	166	167
168	169	170	171	172	173	174	175	168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183	176	177	178	179	180	181	182	183
184	185	186	187	188	189	190	191	184	185	186	187	188	189	190	191
192	193	194	195	196	197	198	199	192	193	194	195	196	197	198	199
200	201	202	203	204	205	206	207	200	201	202	203	204	205	206	207
208	209	210	211	212	213	214	215	208	209	210	211	212	213	214	215
216	217	218	219	220	221	222	223	216	217	218	219	220	221	222	223
224	225	226	227	228	229	230	231	224	225	226	227	228	229	230	231
232	233	234	235	236	237	238	239	232	233	234	235	236	237	238	239
240	241	242	243	244	245	246	247	240	241	242	243	244	245	246	247
248	249	250	251	252	253	254	255	248	249	250	251	252	253	254	255

Σχήμα 3.11: a) Αριστερά: Γραμμική θέση κάθε νήματος σε ένα block. b) Δεξιά: Γραμμική θέση κάθε νήματος στο σύνολο του grid.

Στο σχήμα 3.11a το μαύρο τμήμα είναι το block με συντεταγμένες (0,0), το μπλέ block με συντεταγμένες (1,0), το κίτρινο (0,1) και το πράσινο (1,1). Όπως αναφέρθηκε το διάνυσμα *threadIdx* δίνει τις συντεταγμένες τοπικά στο block, δηλαδή, σε κάθε block παίρνει πάντα τις τιμές $[0..block_size-1]$, με το *block_size* στη περίπτωση του σχήματός να είναι 8x8. Στη σχέση, αν για x οριστεί *threadIdx.x*, για Nx *blockDim.x* και για y *threadIdx.y* θα υπολογιστούν οι γραμμικές θέσεις σε κάθε block όπως φαίνεται στο σχήμα 3.11a. Για τον υπολογισμό της καθολικής γραμμικής θέσης κάθε νήματος θα πρέπει πρώτα να υπολογιστούν οι τοπικές συντεταγμένες. Για τη διάσταση x, θα είναι:

$$x_{global} = threadIdx.x + blockDim.x * blockIdx.x$$

και αντίστοιχα για τη y διάσταση θα είναι:

$$y_{global} = threadIdx.y + blockDim.y * blockIdx.y$$

Έχοντας ορίσει τις νέες συντεταγμένες x_{global} και y_{global} στην σχέση με x την συνολική διάσταση του x , $blockDim.x * gridDim.x$ (μέγεθος του block στην διάσταση x επί το πλήθος τους), θα υπολογιστούν οι καθολικές γραμμικές θέσεις, όπως φαίνονται και στο σχήμα 3.11b. Αυτές είναι ιδιαίτερα χρήσιμες καθώς στην πλειονότητα των περιπτώσεων, όλες οι συναλλαγές με την καθολική μνήμη θα γίνονται με δείκτη αντίστοιχο του καθολικού και όλες οι συναλλαγές με την κοινόχρηστη μνήμη με τον τοπικό δείκτη.

Ο καθολικός δείκτης ενός νήματος είναι:

$$tid_{global} = threadIdx.x + blockIdx.x * blockDim.x + \backslash (gridDim.x * blockDim.x) * (threadIdx.y + blockIdx.y * blockDim.y);$$

Υποθέτοντας έναν πίνακα 8x8 χωρισμένο σε 4 δισδιάστατα blocks με αναλογία 1-1 στοιχείων - νημάτων. Για κάποιον λόγο, πρέπει γίνει επεξεργασία με διαφορετικό τρόπο το πάνω και το κάτω μισό. Ο kernel είναι σαν ένα loop πάνω σε όλα τα νήματα:

```
_global_ void this_will_run_in_the_gpu ( float * i n t , float * out ) {
int tid_local = threadIdx.x + blockDim.x * blockIdx.x ;
int tid_global = threadIdx.x + blockDim.x * blockIdx.x + \
( gridDim.x * blockDim.x ) * ( threadIdx.y + blockIdx.y * blockDim.y ) ;
#Στο πάνω μισό θέλουμε την τιμή tid_global ,
#Στο κάτω μισό την τιμή tid_local
if ( blockIdx.y < blockDim.y / 2 ) out [ tid_global ] = tid_global ;
else out [ tid_global ] = tid_local ;
}
```

Ένας kernel έχει την ικανότητα να περιέχει "αδρανή" νήματα. Νήματα τα οποία δεν θα κάνουν τίποτα. Αυτό χρησιμεύει όταν τα δεδομένα μας δεν χωράνε ακριβώς στα blocks. Τότε χρειάζονται ένα ή δύο blocks (ανάλογα σε μία ή δύο διαστάσεις) επιπλέον. Τα νήματα που είναι εκτός εμβέλειας των δεδομένων, απλά δεν θα κάνουν τίποτα.

3.2.4 Καλές πρακτικές και μέγιστη απόδοση

Ποια προβλήματα μπορούν να επιταχυνθούν σημαντικά από μία κάρτα γραφικών:

Λόγω των μεγάλων διαφορών μεταξύ ενός επεξεργαστή γραφικών και ενός κεντρικού επεξεργαστή, ένα πρόβλημα πρέπει να χωρίζεται έτσι ώστε ο κάθε επεξεργαστής να εκτελεί το κομμάτι του προβλήματος στο οποίο είναι καλύτερος.

- Η κάρτα γραφικών είναι ιδανική για προβλήματα ή τμήματα τους, τα οποία περιέχουν υπολογισμούς οι οποίοι μπορούν να εκτελεστούν σε πολλά στοιχεία δεδομένων ταυτόχρονα (παράλληλα). Συνήθως τέτοια προβλήματα περιέχουν αριθμητικούς υπολογισμούς σε μεγάλες ενότητες δεδομένων (πίνακες), όπου η ίδια διεργασία μπορεί να γίνει ταυτόχρονα σε χιλιάδες ή εκατομμύρια στοιχεία. Γενικά, για να είναι ένας αλγόριθμος αποδοτικός στην κάρτα γραφικών θα πρέπει να κάνει χρήση μεγάλου αριθμού νημάτων [32].

- Οι προσβάσεις στην μνήμη από τα νήματα πρέπει να έχουν κάποια συνοχή. Πρέπει να τηρούνται ορισμένα πρότυπα κατά τις προσβάσεις στην μνήμη ώστε αυτές να γίνονται με όσο το δυνατόν λιγότερες συναλλαγές.

- Για να χρησιμοποιηθεί η κάρτα γραφικών, πρέπει πρώτα να μεταφερθούν τα δεδομένα σε αυτήν μέσω του διαύλου PCI Express (PCIe). Αυτές οι μεταφορές είναι γενικά αργές και η πολυπλοκότητα των υπολογισμών που θα εκτελεστούν στην κάρτα γραφικών πρέπει να δικαιολογεί το πλήθος και μέγεθος των μεταφορών. Για παράδειγμα, αν μεταφερθούν δύο πίνακες $N \times N$ μόνο για να προστεθούν τα στοιχεία τους και να επιστρέψουν το αποτέλεσμα, η επιτάχυνση δεν θα είναι σημαντική.

- Τα δεδομένα πρέπει να είναι δυνατό να μείνουν στην μνήμη της συσκευής για όσο το δυνατό μεγαλύτερο διάστημα. Επειδή οι μεταφορές μνήμης μεταξύ των δύο συστημάτων είναι ακριβές, πρέπει τα δεδομένα να μένουν στην κάρτα γραφικών κατά διαδοχικές εκτελέσεις kernel.

Μέγιστη απόδοση

Για να επιτευχθεί η μέγιστη απόδοση για τον αλγόριθμό, πρέπει να ακολουθηθούν κάποιοι κανόνες οι οποίοι αναφέρθηκαν παραπάνω. Εδώ θα κατηγοριοποιηθούν αυτοί οι κανόνες ως προς την σημαντικότητα τους [32], [33], [34].

1. Πρώτα κάποιος πρέπει να επικεντρωθεί στην παραλληλοποίηση του σειριακού κώδικα. Το κέρδος σε ταχύτητα που θα έχει ένας αλγόριθμος εξαρτάται σχεδόν ολοκληρωτικά από το πόσο μπορεί να παραλληλοποιηθεί. Κώδικας ο οποίος δεν μπορεί να γίνει παράλληλος σε μεγάλο βαθμό θα πρέπει να εκτελείται από τον κεντρικό επεξεργαστή. Το μεγαλύτερο (σχετικό) κέρδος σε ταχύτητα έναντι σειριακού κώδικα, μας το δίνει η μεγιστοποίηση του παράλληλου τμήματος ενός αλγορίθμου, και όχι η μεγιστοποίηση των διαθέσιμων επεξεργαστών.

2. Εφαρμογή των κανόνων συναλλαγών της καθολικής μνήμης όπου αυτό είναι δυνατό. Οι συναλλαγές πρέπει να είναι όσο το δυνατό συνεχείς ως προς το τμήμα της μνήμης και ευθυγραμμισμένες σε πολλαπλάσια των 32, 64 ή 128 bytes από την αρχή του τμήματος της μνήμης.

3. Ελαχιστοποίηση των μεταφορών μνήμης μεταξύ κεντρικού επεξεργαστή και κάρτας γραφικών. Ακόμα και αν αυτό συνεπάγεται εκτέλεση κάποιων μικρών kernel οι οποίοι δεν είναι αποτελεσματικοί στην κάρτα γραφικών, έτσι ώστε να αποφύγουμε ενδιάμεσες μεταφορές. Επίσης, η αποδοτικότητα μίας μεγάλης μεταφοράς, είναι μεγαλύτερη από την αποδοτικότητα πολλών μικρών μεταφορών.

4. Αποφυγή των διακλαδώσεων της ροής ενός προγράμματος μέσα σε ένα warp.

5. Πρέπει να έχουμε το λιγότερο 192 ενεργά νήματα συνολικά σε κάθε πολυεπεξεργαστή για να καλύψουμε τις καθυστερήσεις των αριθμητικών εντολών. Αντίστοιχα, για τις προσβάσεις στην μνήμη, 256 νήματα είναι συνήθως αρκετά.

6. Οι προσβάσεις στην κοινόχρηστη μνήμη πρέπει να γίνονται με τρόπο ώστε να αποφεύγονται τα συγκρούσεις τραπεζών.

7. Το πλήθος των νημάτων ανά block πρέπει να είναι κάποιο πολλαπλάσιο του 32, επειδή σε αυτήν την περίπτωση έχουμε την καλύτερη υπολογιστική αποδοτικότητα και συνήθως έτσι επιτυγχάνεται και η συνεχής και ευθυγραμμισμένη πρόσβαση στην μνήμη αρκετά πιο εύκολα.

8. Χρήση της κοινόχρηστης μνήμης, αντί της καθολικής, όπου αυτό καθίσταται δυνατό.

9. Αποφυγή συγχρονισμών όσο αυτό είναι δυνατό.

10. Χρήση των γρήγορων μαθηματικών συναρτήσεων χαμηλής ακρίβειας, όπου η ακρίβεια δεν είναι σημαντική.

11. Όλα τα νήματα του warp θα πρέπει να έχουν πρόσβαση στην ίδια θέση της μνήμης σταθερών.

3.3 Η εργαλειοθήκη CUDA Toolkit

Για τον μεταγλωττισμό ενός προγράμματος σε κάρτας γραφικών χρειάζεται ειδικός μεταγλωττιστής παράλληλα με τον συνηθισμένο μεταγλωττιστή. Για να εκτελεστεί χρειάζεται κάποιες ειδικές βιβλιοθήκες, για να αναλυθεί το πρόγραμμα και να βρεθούν προβλήματα χρειάζονται ειδικά εργαλεία. Εδώ γίνεται αναφορά μόνο στην έκδοση της εργαλειοθήκης για

λειτουργικό σύστημα Linux, αλλά ο τρόπος και τα εργαλεία δεν διαφέρουν σχεδόν καθόλου με την έκδοση της εργαλειοθήκης για άλλα λειτουργικά συστήματα όπως τα Windows και το Mac OSX. Αρχικά κάποιος πρέπει να κατεβάσει την νεότερη έκδοση CUDA οδηγού για την κάρτα γραφικών του από την ιστοσελίδα της NVIDIA και να την εγκαταστήσει. Στη συνέχεια, πρέπει να κατεβάσει την εργαλειοθήκη που απευθύνεται στο σύστημα του (32bit, 64bit και διανομή του Linux) από την ίδια ιστοσελίδα και να τρέξει το script εγκατάστασης. Παράλληλα με την εργαλειοθήκη, για τη μεταγλώττιση χρειάζεται και ο gcc με μεγάλη έκδοση 4.3.x. Ο gcc συνήθως είναι εξ αρχής εγκατεστημένος στις διανομές Linux, αλλά είναι πολύ πιθανό να έχουμε την νεότερη έκδοση 4.4.x η οποία δεν υποστηρίζεται. Σε κάθε περίπτωση θα πρέπει να γίνει εγκατάσταση της έκδοσης 4.3.x. Το λειτουργικό Linux υποστηρίζει πολλαπλές εκδόσεις του gcc, οπότε δεν θα υπάρξει πρόβλημα. Στη συνέχεια, παρουσιάζονται κάποια paths του συστήματος, έτσι ώστε να βλέπουν τα εκτελέσιμα και τις βιβλιοθήκες της εργαλειοθήκης.

```
export PATH = /usr/local/cuda/bin:\$PATH
export LD_LIBRARY_PATH = /usr/local/cuda/lib:\$LD_LIBRARY_PATH
```

3.3.1 Ο μεταγλωττιστής nvcc

Εφόσον το πρόγραμμα είναι έτοιμο, μπορεί να μεταγλωττιστεί με τον nvcc. Τα αρχεία που περιέχουν κλήσεις συναρτήσεων kernel ή CUDA θα πρέπει να έχουν κατάληξη .cu. Έστω ότι το πρόγραμμα αποτελείται από τα αρχεία main.cu, kernel.cu. Θα χρησιμοποιηθεί ο nvcc όπως κάθε άλλος μεταγλωττιστής:

```
nvcc main.cu kernel.cu -o program.exe
```

3.3.2 CULA Περιγραφή Λογισμικού

Ο λόγος για τον οποίο στην μεταπτυχιακή εργασία γίνεται χρήση του υπολογιστικού πακέτου CULA (CUDA LINEAR ALGEBRA) [29], είναι η μελέτη της απόδοσης, του χρόνου εκτέλεσης, του υπολογιστικού πακέτου CULA σε απαιτητικές εφαρμογές με πολύ μεγάλα συστήματα. Επίσης αναζητείται σε ποιο βαθμό υπερτερεί το συγκεκριμένο πακέτο σε σχέση με συναρτήσεις Cuda kernel που έχουν προγραμματιστεί παράλληλα. Το CULA χρησιμοποιεί επεξεργαστές γραφικών (GPUs), προκειμένου να επιλύσει παράλληλα μεγάλα και αραιά γραμμικά συστήματα αλγεβρικών εξισώσεων, χρησιμοποιώντας επιλύτες τύπου Krylov. Το πακέτο CULA χρησιμοποιεί τη βιβλιοθήκη CUDA για τη παράλληλη επεξεργασία. Το πακέτο CULA συγκρίνεται, ως προς το χρόνο εκτέλεσης, με συνάρτηση CUDA kernel.

Η συνάρτηση CUDA kernel διασπά το συμμετρικό και θετικά ορισμένο μητρώο A (σχέση 2.2) σε άνω τριγωνικό μητρώο U ($A = LU = U^T U$) με την παραγοντοποίηση CHOLESKY (κεφάλαιο 4). Η συνάρτηση CUDA kernel έχει προγραμματιστεί παράλληλα και χρησιμοποιεί τους Cuda cores της GPU όπως και το CULA. Από την πλευρά του πακέτου CULA επιλέγεται η συνάρτηση DPOTRF (πίνακας 3.5) για την διάσπαση του μητρώου A . Και οι δύο συναρτήσεις καλούνται μέσω πηγαίου κώδικα C/C++. Στο κεφάλαιο 7 παρουσιάζονται τα αποτελέσματα των συγκρίσεων και αποδεικνύεται το μεγάλο πλεονέκτημα της GPU επεξεργασίας κατά την επεξεργασία πολύ μεγάλων συστημάτων.

Εδώ περιγράφεται αναλυτικά το υπολογιστικό πακέτο που χρησιμοποιείται στη παρούσα εργασία. Πρόκειται για το CULA που δημιουργήθηκε από την EM Photonics [28]. Το πακέτο CULA (CUDA Linear Algebra) [30] είναι ένα υπολογιστικό πακέτο που βοηθά στην παράλληλη επίλυση μεγάλων γραμμικών συστημάτων σε επεξεργαστές γραφικών (GPUs). Δημιουργήθηκε από μια ομάδα μηχανικών της EM Photonics [28] σε συνεργασία με την NVIDIA [31]. Διανέμεται σε 2 εκδόσεις, μια μόνο για "πυκνά" συστήματα (CULA DENSE) και μια για "αραιά" συστήματα (CULA

SPARSE). Καθεμιά από τις εκδόσεις αυτές περιέχει γνωστούς επιλύτες των ανάλογων γραμμικών συστημάτων οι οποίοι στην ουσία "τρέχουν" σε βιβλιοθήκες γραμμένες σε CUDA (NVIDIA CULADENSE, NVIDIA CUSPARSE, κ.α.). Διανέμεται με συμβατότητα για όλες τις γνωστές υπολογιστικές πλατφόρμες (Windows, Linux, MacOSx), ενώ επίσης περιέχει βιβλιοθήκες (bindings) για τη χρήση της σε κώδικες γραμμένους πάνω στις πιο γνωστές γλώσσες προγραμματισμού όπως οι C, C++, Fortran. Στη παρούσα εργασία εξετάζονται μόνο πυκνά γραμμικά συστήματα, επομένως γίνεται περιγραφή του πακέτου CULA DENSE.

Το πακέτο CULA DENSE διαθέτει ένα σύνολο από εξισώσεις γραμμικής άλγεβρας, επιλύτες (solvers) για την επίλυση πυκνών (dense) συστημάτων. Μια προεπισκόπηση των περιεχομένων του πακέτου παρατίθεται στον πίνακα 3.4 και 3.5:

Matrix Type	Operation	S	C	D	Z
General	Matrix-matrix multiply	SGEMM	CGEMM	DGEMM	ZGEMM
	Matrix-vector multiply	SGEMV	CGEMV	DGEMV	ZGEMV
Triangular	Triangular matrix-matrix multiply	STRMM	CTRMM	DTRMM	ZTRMM
	Triangular matrix solve	STRSM	CTRSM	DTRSM	ZTRSM
Symmetric	Symmetric matrix-matrix multiply	SSYMM	CSYMM	DSYMM	ZSYMM
	Symmetric rank 2k update	SSYR2K	CSYR2K	DSYR2K	ZSYR2K
	Symmetric rank k update	SSYRK	CSYRK	DSYRK	ZSYRK
Hermitian	Hermitian matrix-matrix multiply		CHEMM		ZHEMM
	Hermitian rank 2k update		CHER2K		ZHER2K
	Hermitian rank k update		CHERK		ZHERK

Πίνακας 3.4: Περιεχόμενες εξισώσεις γραμμικής άλγεβρας στο πακέτο CULA DENSE.

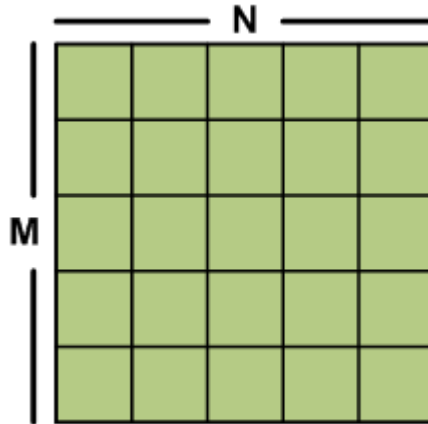
Matrix Type	Operation	S	C	D	Z
General	Factorize and solve	SGESV	CGESV	DGESV	ZGESV
	Factorize and solve with iterative refinement			DSGESV	ZCGESV
	LU factorization	SGETRF	CGETRF	DGETRF	ZGETRF
	Solve using LU factorization	SGETRS	CGETRS	DGETRS	ZGETRS
	Invert using LU factorization	SGETRI	CGETRI	DGETRI	ZGETRI
Positive Definite	Factorize and solve	SPOSV	CPOSV	DPOSV	ZPOSV
	Cholesky factorization	SPOTRF	CPOTRF	DPOTRF	ZPOTRF
Triangular	Invert triangular matrix	STRTRI	CTRTRI	DTRTRI	ZTRTRI
	Solve triangular system	STRTRS	CTRTRS	DTRTRS	ZTRTRS
Banded	LU factorization	SGBTRF	CGBTRF	DGBTRF	ZGBTRF
Pos Def Banded	Cholesky factorization	SPBTRF	CPBTRF	DPBTRF	ZPBTRF

Πίνακας 3.5: Περιεχόμενοι επιλύτες γραμμικών συστημάτων στο πακέτο CULA DENSE.

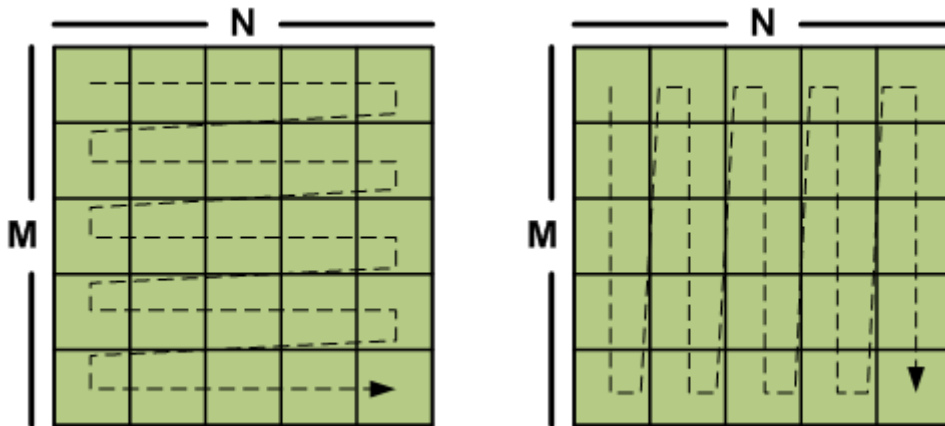
Όλες οι εξισώσεις και οι επιλύτες μπορούν να συνδυαστούν μεταξύ τους. Το πακέτο CULA χρησιμοποιεί μια ονομαστική σύμβαση για την κλήση των υπορουτινών των καλούμενων εξισώσεων και επιλυτών, μια σύμβαση που μπορεί εύκολα να απομνημονευτεί [35] [36].

Τύποι Δεδομένων

Το CULA υποστηρίζει φυσικούς δεκαδικούς διπλής ακριβείας καθώς και μιγαδικούς δεκαδικούς διπλής ακριβείας. Τα αναγνωριστικά είναι το "D" για τους πραγματικούς δεκαδικούς και το "Z" για τους μιγαδικούς με πραγματικό και φανταστικό μέρος, δεκαδικούς διπλής ακριβείας. Η δομή των πινάκων πρέπει να έχει συγκεκριμένη μορφή, θα πρέπει να ακολουθεί αποθήκευση στήλης "column major". Αν έχουμε ένα πίνακα $N \times N$ όπως στο σχήμα 3.12 τότε για να λειτουργήσει σωστά η συνάρτηση του πακέτου CULA DENSE πρέπει ο πίνακας να αποθηκευτεί όπως δείχνει το σχήμα 3.13 b.



Σχήμα 3.12: Πίνακας $N \times N$.



Σχήμα 3.13: a) Αριστερά: Πίνακας $N \times N$ με αποθήκευση "Row major". b) Δεξιά: Πίνακας $N \times N$ με αποθήκευση "Column major".

4 Μέθοδοι Επίλυσης

Στο κεφάλαιο αυτό γίνεται περιγραφή της μεθόδου επίλυσης Cholesky η οποία επιλέχθηκε ως επιλύτης από το πακέτο CULA [35] [36] στα παραδείγματα που θα ακολουθήσουν. Γίνεται μία σύντομη περιγραφή του θεωρητικού υπόβαθρου και του αλγορίθμου επίλυσης.

4.1 Παραγοντοποίηση CHOLESKY

Έχουμε το ακόλουθο γραμμικό σύστημα $Ax = b$. Η επίλυση του παραπάνω συστήματος έχει κυρίαρχο ρόλο στα επιστημονικά προβλήματα και συναντάται συχνά σε αυτά. Για αυτό έχουν αναπτυχθεί τρόποι επίλυσής τους που μετατρέπουν το παραπάνω πρόβλημα σε ένα ευκολότερο με την προϋπόθεση ότι πληρούνται κάποιες συνθήκες σχετικά με τον πίνακα A. Η πιο γνωστή μέθοδος τέτοιου τύπου είναι η μέθοδος Gauss, η οποία ουσιαστικά μετατρέπει το παραπάνω πρόβλημα σε πρόβλημα τριγωνικών πινάκων που είναι πιο εύκολο να επιλυθούν. Οι μέθοδοι, όπως αυτή του Gauss λέγονται μέθοδοι LU παραγοντοποίησης (LU Factorization), και μία από αυτές είναι και η μέθοδος που εξετάζουμε. Προκειμένου να χρησιμοποιηθεί η μέθοδος της παραγοντοποίησης Cholesky, όπως αυτή παρουσιάζεται στα [37] και [38], πρέπει ο πίνακας A να είναι ένας τετραγωνικός πίνακας $n \times n$, συμμετρικός και θετικά ορισμένος. Τα υπόλοιπα στοιχεία του συστήματος είναι τα γνωστά, δηλαδή το διάνυσμα b είναι ένα γνωστό διάνυσμα μεγέθους n και x είναι το διάνυσμα των αγνώστων. Η μετατροπή του παραπάνω γραμμικού συστήματος σε ένα τριγωνικό σύστημα και η επίλυσή του περιλαμβάνει δύο βασικά βήματα. Αρχικά γίνεται ο υπολογισμός του κάτω τριγωνικού πίνακα L, ο οποίος ονομάζεται παράγοντας Cholesky (Cholesky Factor), με βάση την σχέση $A = LL^T$ και στην συνέχεια έχουμε την τριγωνική επίλυση προκειμένου να υπολογίσουμε το διάνυσμα x με βάση τις επόμενες δύο σχέσεις

$$Ly = b \text{ και } L^T x = y \quad (4.1)$$

Η επίλυση μπορεί να γίνει και με άνω τριγωνικούς πίνακες ωστόσο συνηθίζεται η χρήση των κάτω τριγωνικών πινάκων. Για τον υπολογισμό των στοιχείων του πίνακα L ακολουθεί το επόμενο παράδειγμα. Υποθέτοντας ότι ο πίνακας A είναι ένας 3×3 πίνακας και πληρεί τις προϋποθέσεις που αναφέρθηκαν παραπάνω

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad (4.2)$$

Επομένως, με βάση τη σχέση που συνδέει τον πίνακα A με τον πίνακα L έχουμε:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{bmatrix} = LL^T \quad (4.3)$$

Επομένως, βάση των παραπάνω, για τον υπολογισμό των στοιχείων του διανύσματος Cholesky χρησιμοποιούνται οι γενικοί τύποι 4.4 για τα διαγώνια στοιχεία και 4.5 για τα κάτω από την διαγώνιο:

$$l_{kk} = \sqrt{a_{kk} - \sum_{j=1}^{k-1} l_{kj}^2} \quad (4.4)$$

$$l_{ik} = \frac{1}{l_{kk}} \left(a_{ik} - \sum_{j=1}^{k-1} l_{ij} l_{kj} \right) \quad (4.5)$$

Επομένως με τους τύπους αυτούς γίνεται ο υπολογισμός των στοιχείων του διανύσματος Cholesky.

5 Εφαρμογές σε 2D πλέγμα γύρω από γεωμετρία φύλλου πλατάνου

5.1 Περιγραφή του προβλήματος

Η ενότητα αυτή ασχολείται με την εφαρμογή της μεθόδου της μετακίνησης του πλέγματος με δίκτυα συναρτήσεων ακτινικής βάσης. Σκοπός είναι η εφαρμογή της μεθόδου και η διερεύνησή της για το πώς η στροφή της εσωτερικής γεωμετρίας (φύλλο πλατάνου τοποθετημένο μέσα σε ένα κύκλο και με τον ενδιάμεσο χώρο να έχει πλεγματοποιηθεί με τριγωνικά στοιχεία με τη μέθοδο του προελαύνοντος μετώπου “advancing front method”) γύρω από το κέντρο του, επηρεάζει την ποιότητα του πλέγματος. Πρόκειται μεν για μια, εκ πρώτης όψεως, απλή εφαρμογή, αλλά που μπορεί να δώσει χρήσιμα συμπεράσματα για την απόδοση και τη συμπεριφορά της μεθόδου μετατόπισης πλέγματος. Για αυτό το λόγο φτιάχτηκε ένα φύλλο πλατάνου όπως φαίνεται στο σχήμα 5.1. Στη συνέχεια, δημιουργήθηκε μη-δομημένο πλέγμα με τριγωνικά στοιχεία, με τη βοήθεια λογισμικού της ΜΠΥΡ&B/ΕΜΠ, γύρω από το φύλλο πλατάνου.

Στο κεφάλαιο αυτό πραγματοποιούνται διαφορετικού μέτρου στροφές της εσωτερικής γεωμετρίας δεξιόστροφα, προσαρμόζεται το αρχικό πλέγμα στο αλλαγμένο περίγραμμα μέσω του δικτύου RBF και διερευνάται η ποιότητα του πλέγματος. Πρωτίστως ενδιαφέρον προκαλεί το πότε δημιουργούνται ανεπιθύμητα ανεστραμμένα (inverted) τρίγωνα. Ως συναρτήσεις ενεργοποίησης του δικτύου RBF επιλέχθηκαν μία συνάρτηση τοπικής επιρροής και δύο συναρτήσεις ολικής επιρροής. Η συνάρτηση τοπικής επιρροής που χρησιμοποιήθηκε είναι η Wendland $CP C^2$ με τιμή που δίνεται από τη σχέση

$$\Phi(\xi) = \begin{cases} (1 - \xi)^4(4\xi + 1) & 0 \leq \xi \leq 1 \\ 0 & \xi > 1 \end{cases} \quad (5.1)$$

η πρώτη συνάρτηση ολικής επιρροής είναι η συνάρτηση Gauss σύμφωνα με τη σχέση

$$\Phi(r) = e^{-r^2} \quad (5.2)$$

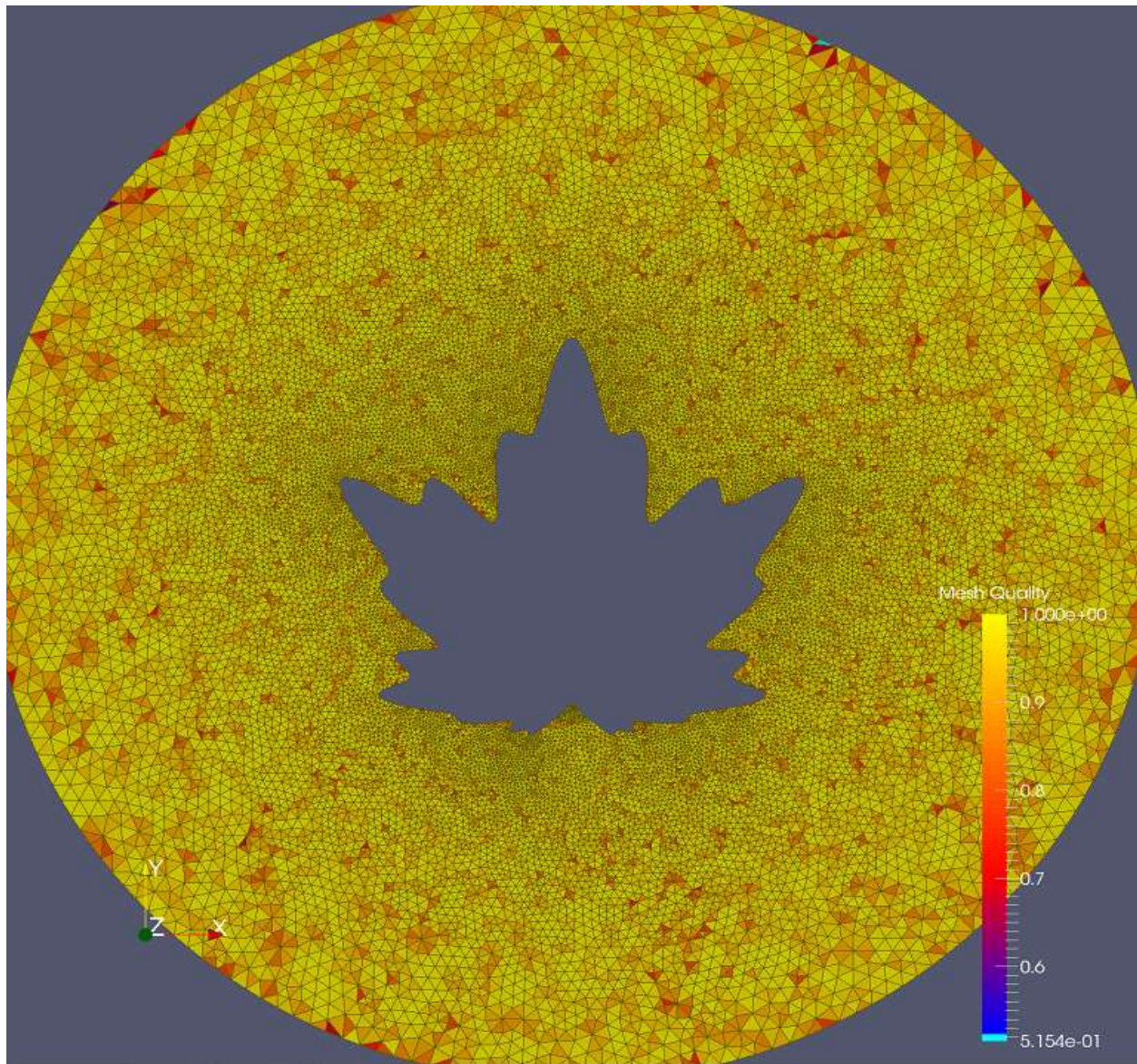
Και η δεύτερη συνάρτηση ολικής επιρροής είναι η συνάρτηση Inverse Multiquadric Biharmonics σύμφωνα με τη σχέση

$$\Phi(r) = \frac{1}{\sqrt{a^2 + r^2}} \quad (5.3)$$

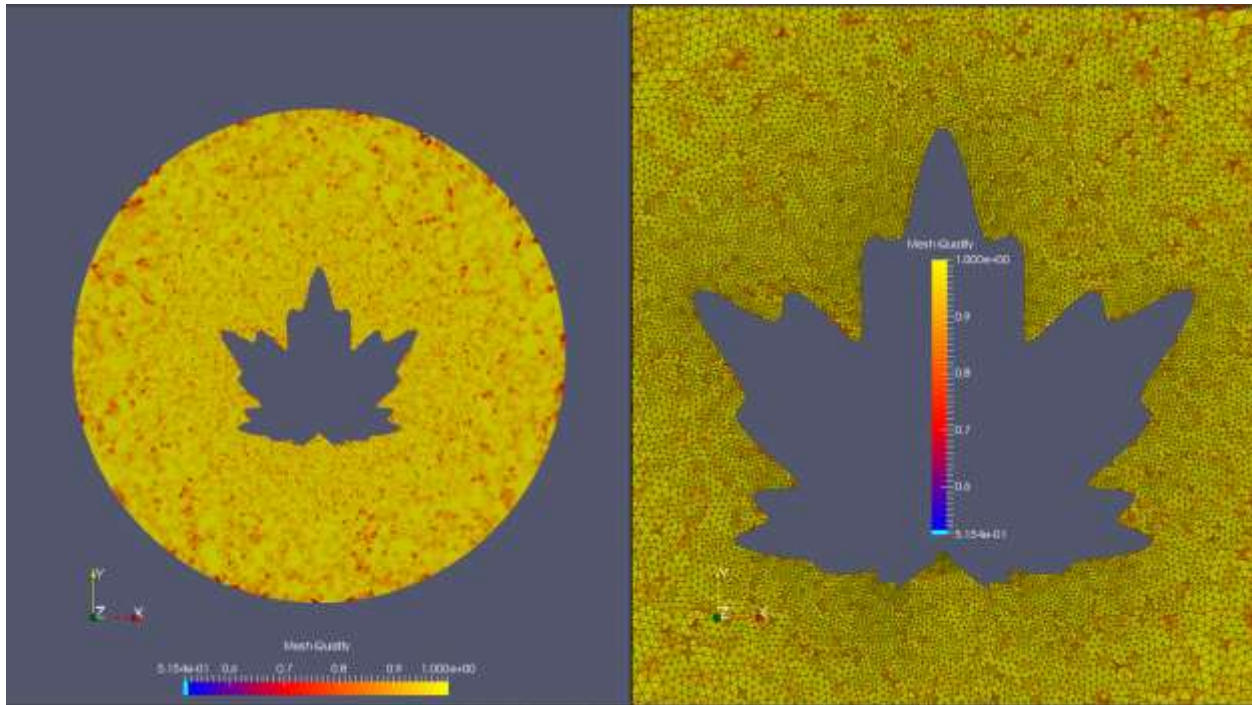
Υπενθυμίζεται ότι $\xi = \frac{r}{R}$, όπου $r = \|\vec{x} - \vec{x}_{c_j}\|$ και \vec{x}, \vec{x}_{c_j} τα διανύσματα θέσης δύο σημείων στο χώρο. Συγκεκριμένα, για τη συνάρτηση τοπικής επιρροής ορίστηκε μία ακτίνα R , τέτοια ώστε κάθε κέντρο \vec{x}_{c_j} του δικτύου να επηρεάζει εκείνους τους κόμβους του χωρικού πλέγματος, εξαιρώντας τους οριακούς κόμβους, οι οποίοι βρίσκονται μέσα σε κύκλο ακτίνας R και κέντρο το εκάστοτε \vec{x}_{c_j} κέντρο του δικτύου RBF.

Επομένως, για τις συναρτήσεις ενεργοποίησης, διερευνάται το πώς η στροφή της εσωτερικής γεωμετρίας επηρεάζει τόσο την ποιότητα του πλέγματος, που προσαρμόζεται με τη μέθοδο της μετακίνησης πλέγματος με δίκτυα RBF, όσο και τη δημιουργία ανεπιθύμητων ανεστραμμένων τριγώνων. Για τη συνάρτηση τοπικής επιρροής, το πρόγραμμα εκτελείται για διάφορες ακτίνες R . Στη συνέχεια, παρουσιάζονται σχόλια και συμπεράσματα για τα αποτελέσματα που προκύπτουν από την εφαρμογή των συναρτήσεων ενεργοποίησης 5.1 και 5.2 του δικτύου RBF στο πλέγμα

των σχημάτων 5.1 και 5.2. Το περίγραμμα του φύλλου (εσωτερική γεωμετρία) αποτελείται από 463 κόμβους, ενώ το περίγραμμα του κύκλου (εξωτερική γεωμετρία), αποτελείται από 200 κόμβους. Στην εσωτερική γεωμετρία πραγματοποιούνται διαφορετικού μέτρου στροφές, προσαρμόζεται το αρχικό πλέγμα (το οποίο αποτελείται από 15677 εσωτερικούς κόμβους με 32017 τριγωνικά στοιχεία) στο στραμμένο περίγραμμα μέσω δικτύου RBF και γίνεται διερεύνηση της ποιότητας του πλέγματος.



Σχήμα 5.1: Το αρχικό πλέγμα.



Σχήμα 5.2: Σχήμα 5.1: Αριστερά :Το αρχικό πλέγμα χωρίς πλεγματικές γραμμές. Δεξιά : Εστίαση στην περιοχή του περιγράμματος του φύλλου.

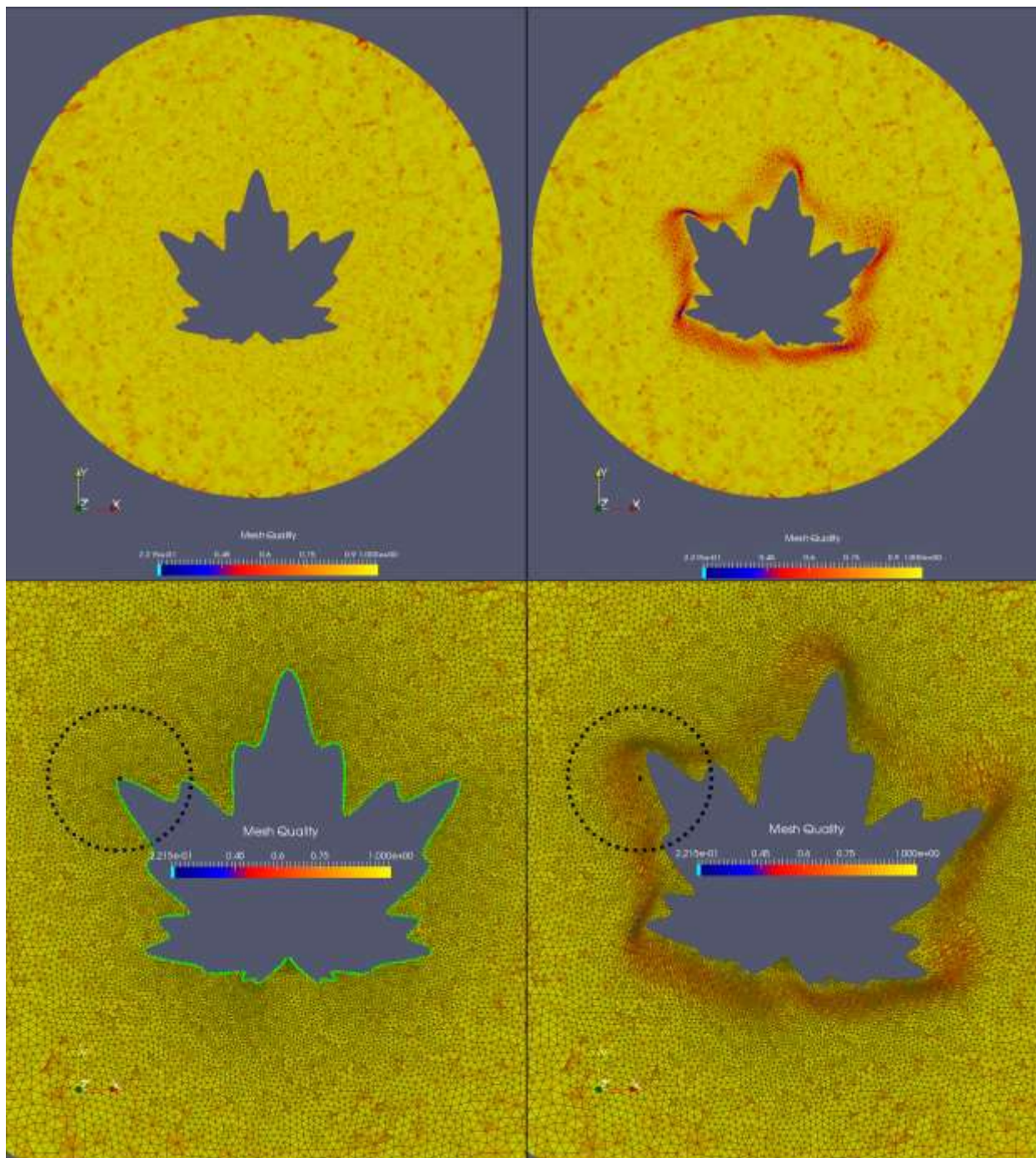
5.2 Παρουσίαση αποτελεσμάτων

5.2.1 Αποτελέσματα με τη $Wendland C^2, R = 2$

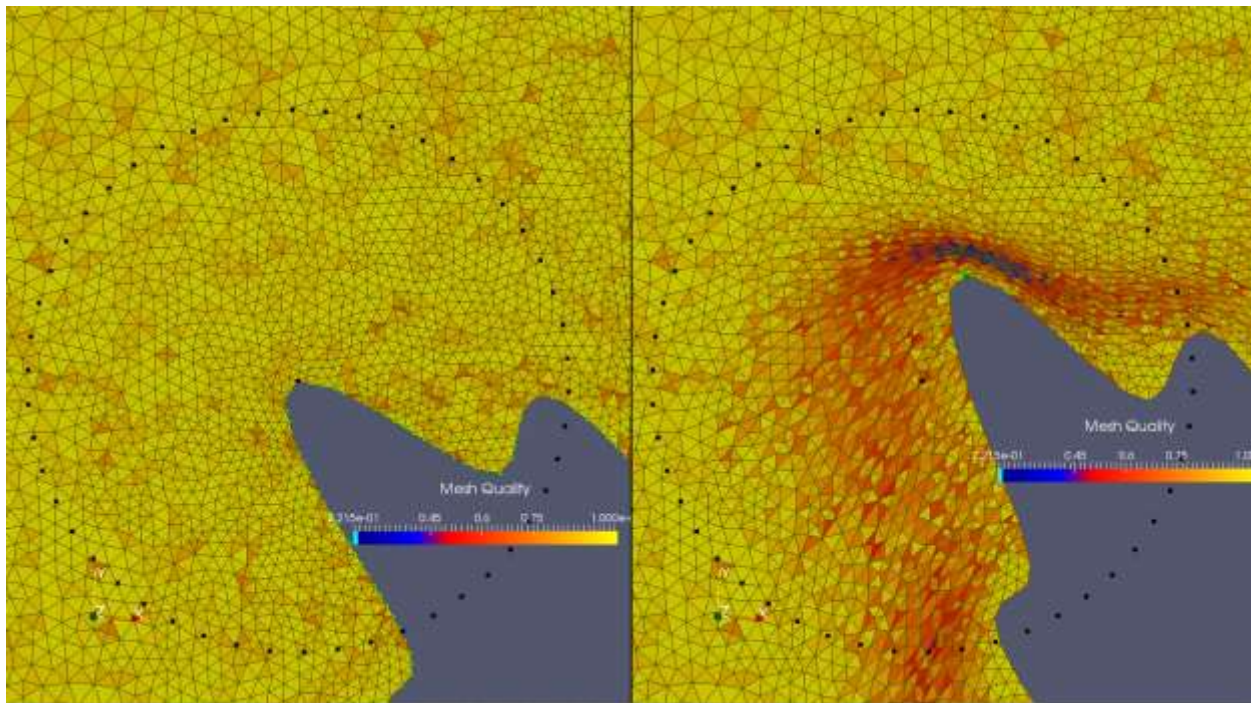
Πρώτη ακτίνα επιρροής που επιλέχθηκε για τη συνάρτηση ενεργοποίησης $Wendland C^2$ είναι $R=2$. Έπειτα έγινε στροφή της εσωτερικής γεωμετρίας για γωνία στροφής, θ . Στον πίνακα 5.1 φαίνονται η μέση τιμή (\bar{q}_e), η τυπική απόκλιση (σ_{qe}) και η ελάχιστη τιμή ($q_{e,min}$) της μετρικής ποιότητας του για τη γωνία στροφής. Στα σχήματα 5.3 και 5.4 φαίνεται το φύλλο πλατάνου σε τυχαία επιλεγμένη γωνία στροφής μέχρι την οποία δεν έχει εμφανίσει ανεστραμμένα τριγωνικά στοιχεία αλλά η ελάχιστη τιμή της μετρικής ποιότητας έχει αρκετά χαμηλή τιμή, καθώς και το πως διαμορφώθηκε το πλέγμα μετά την εφαρμογή της μεθόδου. Η στροφή της εσωτερικής γεωμετρίας σταμάτησε στις 10 μοίρες, γιατί το πλέγμα εμφάνισε πολύ χαμηλή ελάχιστη τιμή μετρικής ποιότητας πλέγματος και αμέσως μετά εμφανίζει τα πρώτα ανεστραμμένα τριγωνικά στοιχεία.

θ (μοίρες)	0	10
\bar{q}_e	0.958755	0.908281
σ_{qe}	0.0428128	0.111562
$q_{e,min}$	0.515426	0.221473

Πίνακας 5.1: Αποτελέσματα μέσης τιμής \bar{q}_e , τυπικής απόκλισης σ_{qe} και ελάχιστη τιμή $q_{e,min}$ της μετρικής ποιότητας του πλέγματος για την αρχική κατάσταση του πλέγματος και για γωνία στροφής της εσωτερικής γεωμετρίας. Ως συνάρτηση ενεργοποίησης είναι η $Wendland C^2$ με ακτίνα $R=2$.



Σχήμα 5.3: Εφαρμογή της μεθόδου της μετακίνησης του πλέγματος με δίκτυο RBF με γωνία στροφής 10 μοιρών του περιγράμματος του φύλλου, με χρήση της Wendland C^2 με $R=2$ και τα αποτελέσματα στην μετρική ποιότητας. **Πάνω αριστερά:** Το αρχικό πλέγμα χωρίς πλεγματικές γραμμές. Η ποιότητα του κάθε τριγωνικού στοιχείου εμφανίζεται με διαφορετικό χρώμα. **Πάνω δεξιά:** Το αναμορφωμένο πλέγμα χωρίς πλεγματικές γραμμές για γωνία στροφής κατά 10 μοίρες και τα αποτελέσματα στην μετρική ποιότητας του πλέγματος. **Κάτω αριστερά:** Το αρχικό πλέγμα με εστίαση στην περιοχή του φύλλου. Με πράσινο χρώμα εμφανίζονται τα κέντρα του δικτύου (όλοι εκείνοι οι κόμβοι ή υποσύνολο αυτών, των οποίων η τιμή της μετατόπισης είναι γνωστή). Με μαύρο χρώμα εμφανίζεται ο κύκλος επιρροής (με ακτίνα επιρροής $R=2$) ενός εκ των κέντρων. **Κάτω δεξιά:** Το αναμορφωμένο πλέγμα και η ανεπιθύμητη παραμόρφωση του πλέγματος. Σε αυτή την περίπτωση, για γωνία μεγαλύτερη των 10 μοιρών, το πλέγμα θα έχει εμφανίσει έναν πολύ μικρό αριθμό ανεστραμμένων τριγώνων, θα έχει δηλαδή κριθεί ακατάλληλο για χρήση με λογισμικό επίλυσης ΜΔΕ.



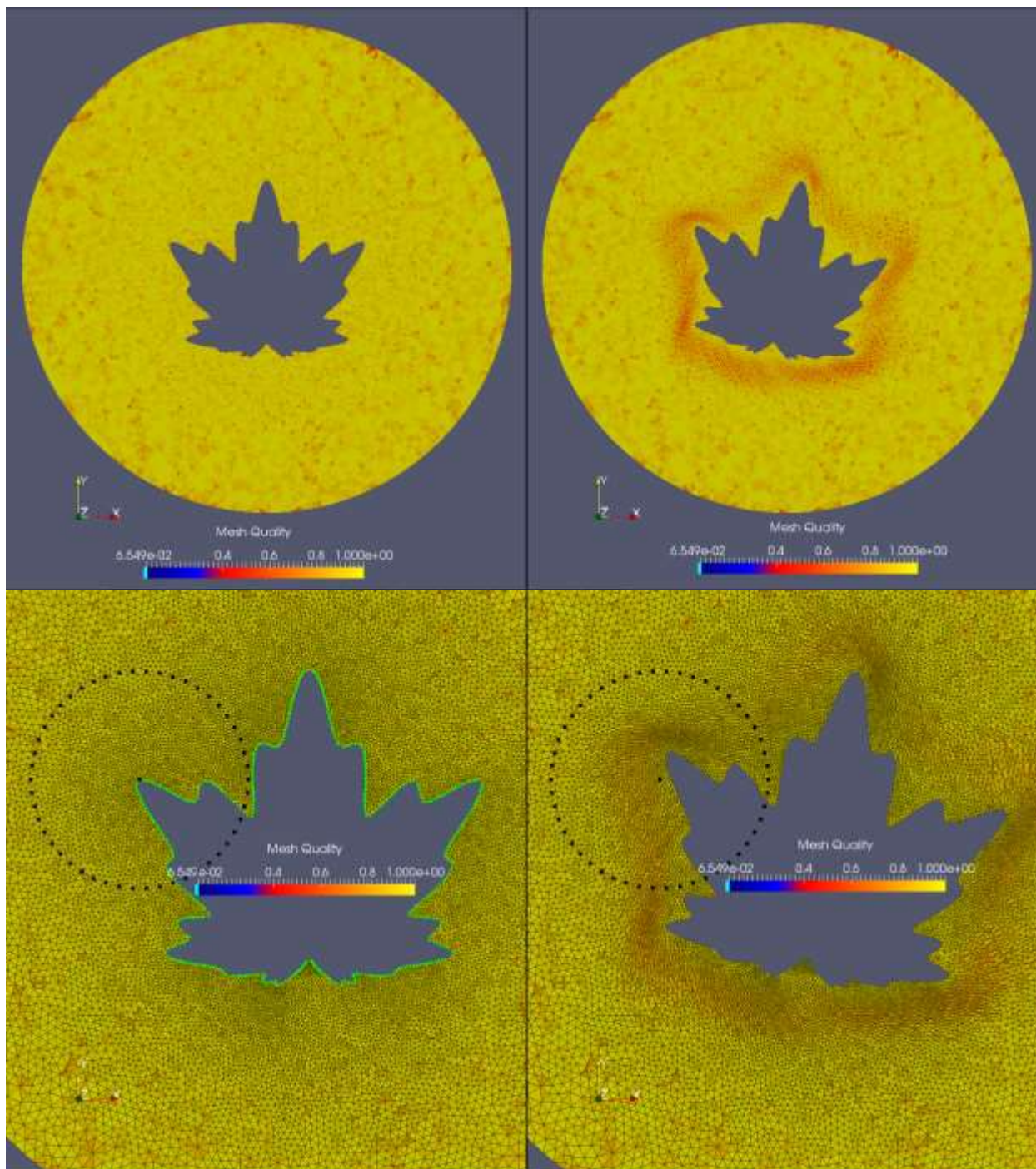
Σχήμα 5.4: **Αριστερά:** Το αρχικό πλέγμα. Με πράσινο χρώμα εμφανίζονται τα κέντρα του δικτύου (όλοι εκείνοι οι κόμβοι ή υποσύνολο αυτών, των οποίων η τιμή της μετατόπισης είναι γνωστή). Με μαύρο χρώμα εμφανίζεται ο κύκλος επιρροής (με ακτίνα επιρροής $R=2$) ενός εκ των κέντρων. **Δεξιά:** Το αναμορφωμένο πλέγμα και η ανεπιθύμητη παραμόρφωση του πλέγματος. Με πράσινο χρώμα εμφανίζεται το μετατοπισμένο κέντρο.

5.2.2 Αποτελέσματα με τη $Wendland C^2$, $R = 3$

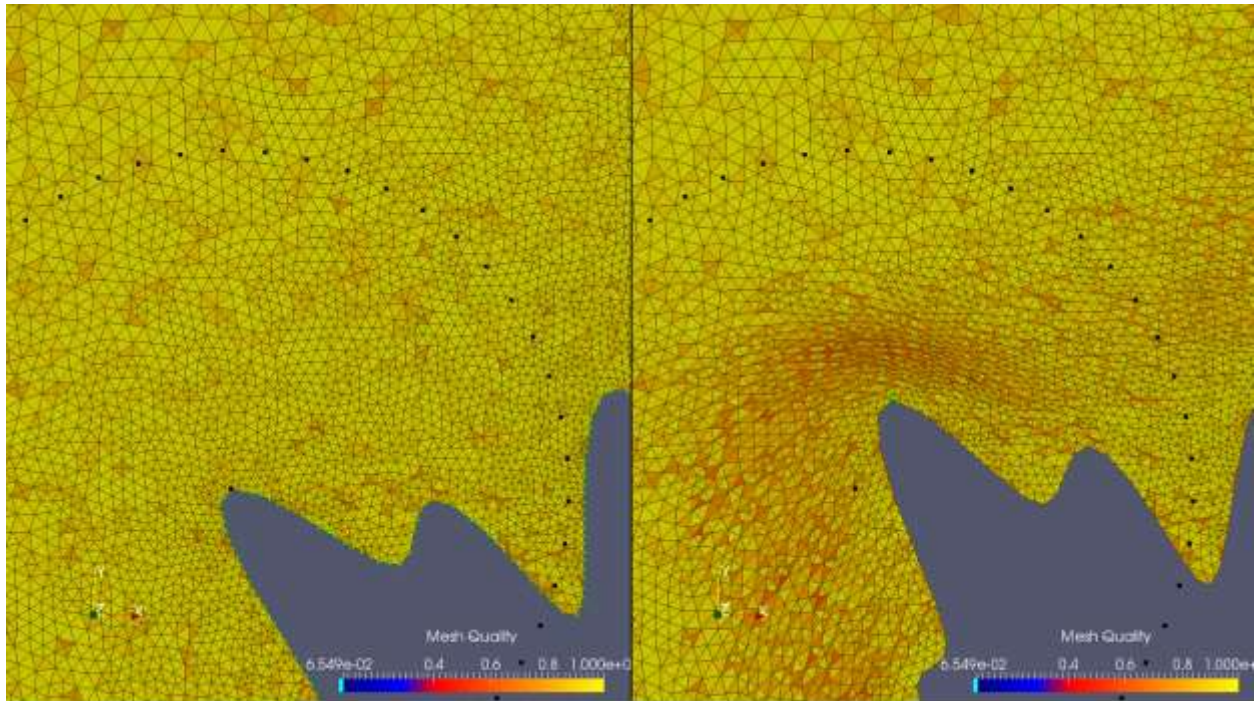
Η επόμενη ακτίνα που επιλέχθηκε για την εφαρμογή της μεθόδου, είναι $R=3$. Ομοίως, στον πίνακα 5.2 φαίνονται η μέση τιμή (\bar{q}_e), η τυπική απόκλιση (σ_{q_e}) και η ελάχιστη τιμή ($q_{e,min}$) της μετρικής ποιότητας για κάθε γωνία στροφής θ , που επιλέχθηκε για την συγκεκριμένη περίπτωση. Στα σχήματα 5.5, 5.6, 5.7 και 5.8 απεικονίζεται αντιστοίχως το ίδιο πλέγμα συναρτήσεως της γωνίας στροφής σε μοίρες 10 και 18. Στα σχήματα 5.6 και 5.8 φαίνεται το πως προσαρμόστηκε το αρχικό πλέγμα στο αλλαγμένο περίγραμμα, το οποίο έχει στραφεί κατά 10 και 18 μοίρες αντίστοιχα. Σε αυτή την περίπτωση, που η ακτίνα επιρροής είναι μεγαλύτερη, το πλέγμα άντεξε μεγαλύτερη γωνία στροφής, όπως αναμενόταν, πριν κριθεί ακατάλληλο.

θ (μοίρες)	0	10	18
\bar{q}_e	0.958755	0.922248	0.864021
σ_{q_e}	0.0428128	0.079114	0.153823
$q_{e,min}$	0.515426	0.392005	0.0654887

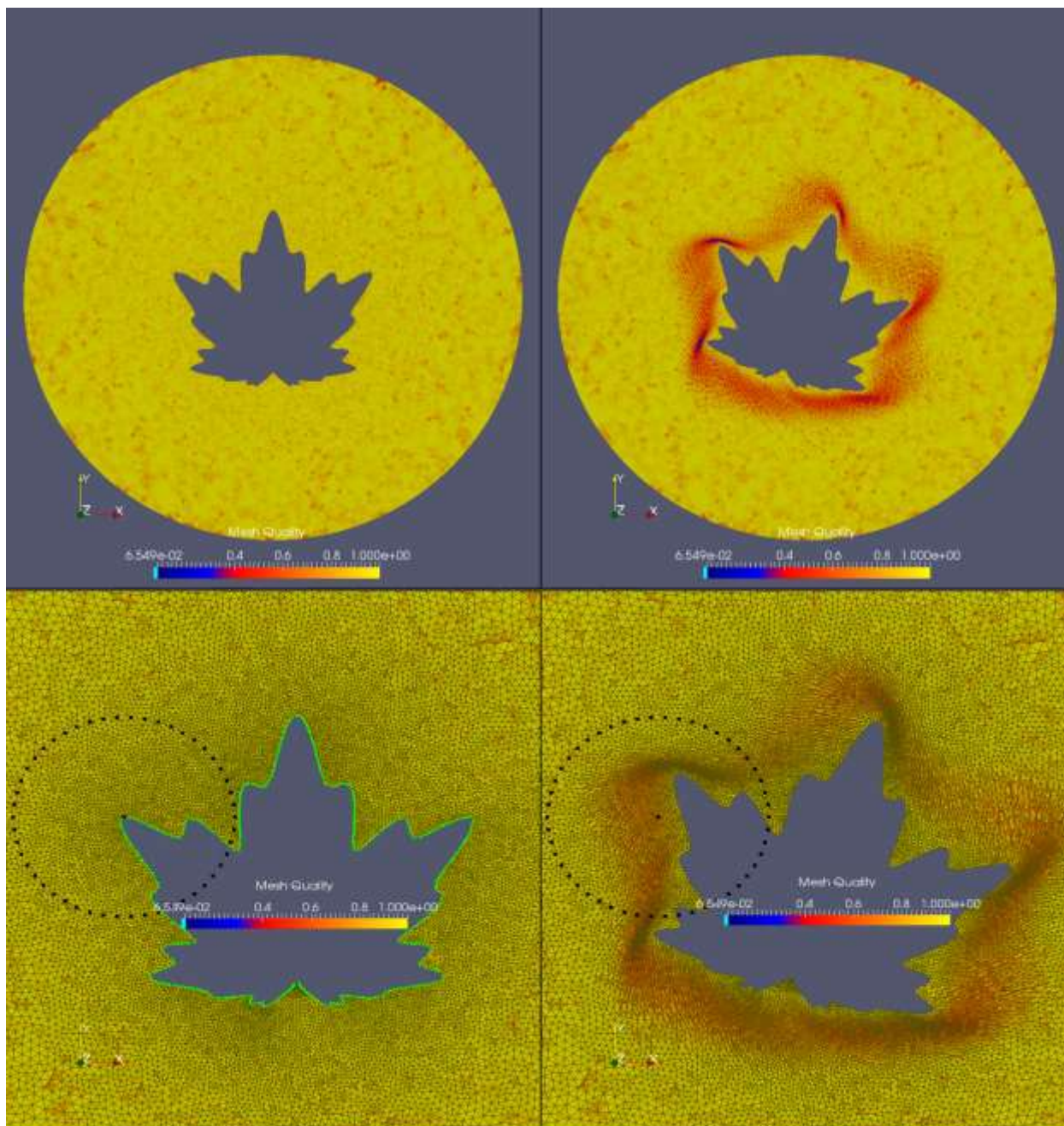
Πίνακας 5.2: Αποτελέσματα μέσης τιμής \bar{q}_e , τυπικής απόκλισης σ_{q_e} και ελάχιστη τιμή $q_{e,min}$ της μετρικής ποιότητας του πλέγματος για την αρχική κατάσταση του πλέγματος και για γωνία στροφής της εσωτερικής γεωμετρίας. Ως συνάρτηση ενεργοποίησης είναι η $Wendland C^2$ με ακτίνα $R=3$.



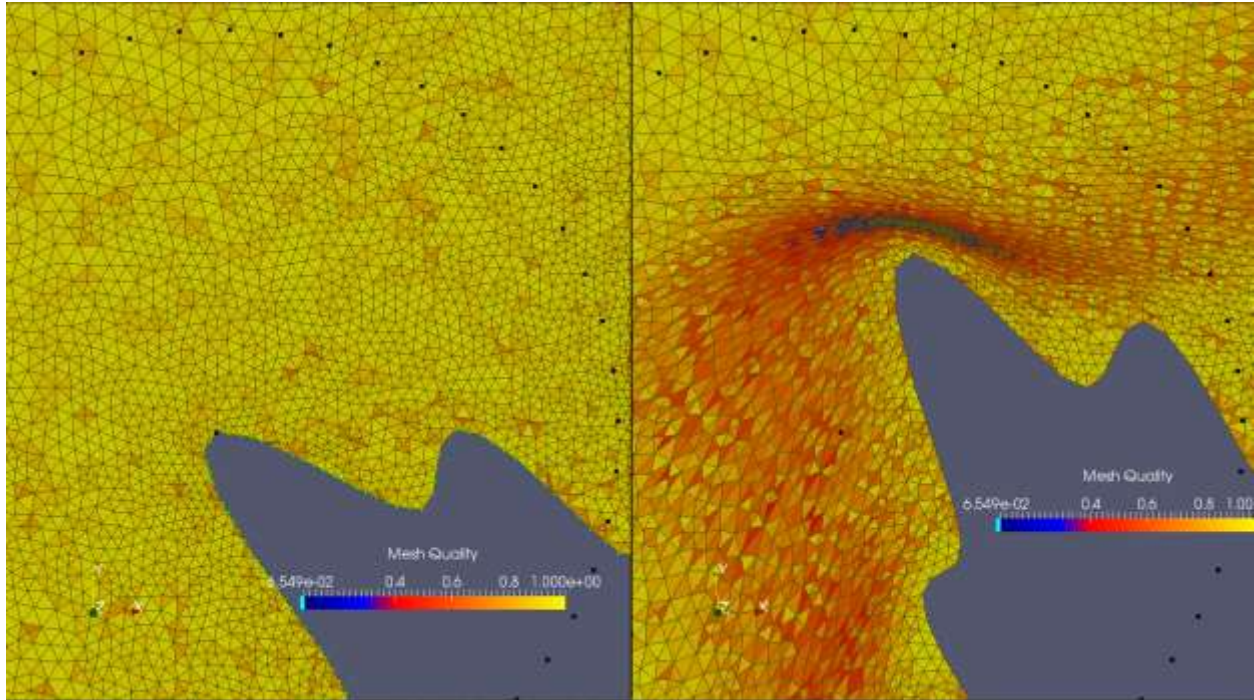
Σχήμα 5.5: Εφαρμογή της μεθόδου της μετακίνησης του πλέγματος με δίκτυο RBF με γωνία στροφής 10 μοιρών του περιγράμματος του φύλλου, με χρήση της Wendland C^2 με $R=3$ και τα αποτελέσματα στην μετρική ποιότητας. **Πάνω αριστερά:** Το αρχικό πλέγμα χωρίς πλεγματικές γραμμές. Η ποιότητα του κάθε τριγωνικού στοιχείου εμφανίζεται με διαφορετικό χρώμα. **Πάνω δεξιά:** Το αναμορφωμένο πλέγμα χωρίς πλεγματικές γραμμές για γωνία στροφής κατά 10 μοίρες και τα αποτελέσματα στην μετρική ποιότητας του πλέγματος. **Κάτω αριστερά:** Το αρχικό πλέγμα με εστίαση στην περιοχή του φύλλου. Με πράσινο χρώμα εμφανίζονται τα κέντρα του δικτύου (όλοι εκείνοι οι κόμβοι ή υποσύνολο αυτών, των οποίων η τιμή της μετατόπισης είναι γνωστή). Με μαύρο χρώμα εμφανίζεται ο κύκλος επιρροής (με ακτίνα επιρροής $R=3$) ενός εκ των κέντρων. **Κάτω δεξιά:** Το αναμορφωμένο πλέγμα και η ανεπιθύμητη παραμόρφωση του πλέγματος.



Σχήμα 5.6: **Αριστερά:** Το αρχικό πλέγμα. Με πράσινο χρώμα εμφανίζονται τα κέντρα του δικτύου (όλοι εκείνοι οι κόμβοι ή υποσύνολο αυτών, των οποίων η τιμή της μετατόπισης είναι γνωστή). Με μαύρο χρώμα εμφανίζεται ο κύκλος επιρροής (με ακτίνα επιρροής $R=3$) ενός εκ των κέντρων. **Δεξιά:** Το αναμορφωμένο πλέγμα και η ανεπιθύμητη παραμόρφωση του πλέγματος. Με πράσινο χρώμα εμφανίζεται το μετατοπισμένο κέντρο.



Σχήμα 5.7: Εφαρμογή της μεθόδου της μετακίνησης του πλέγματος με δίκτυο RBF με γωνία στροφής 18 μοιρών του περιγράμματος του φύλλου, με χρήση της Wendland C^2 με $R=3$ και τα αποτελέσματα στην μετρική ποιότητας. **Πάνω αριστερά:** Το αρχικό πλέγμα χωρίς πλεγματικές γραμμές. Η ποιότητα του κάθε τριγωνικού στοιχείου εμφανίζεται με διαφορετικό χρώμα. **Πάνω δεξιά:** Το αναμορφωμένο πλέγμα χωρίς πλεγματικές γραμμές για γωνία στροφής κατά 18 μοίρες και τα αποτελέσματα στην μετρική ποιότητας του πλέγματος. **Κάτω αριστερά:** Το αρχικό πλέγμα με εστίαση στην περιοχή του φύλλου. Με πράσινο χρώμα εμφανίζονται τα κέντρα του δικτύου (όλοι εκείνοι οι κόμβοι ή υποσύνολο αυτών, των οποίων η τιμή της μετατόπισης είναι γνωστή). Με μαύρο χρώμα εμφανίζεται ο κύκλος επιρροής (με ακτίνα επιρροής $R=3$) ενός εκ των κέντρων. **Κάτω δεξιά:** Το αναμορφωμένο πλέγμα και η ανεπιθύμητη παραμόρφωση του πλέγματος. Σε αυτή την περίπτωση, για γωνία μεγαλύτερη των 18 μοιρών, το πλέγμα θα έχει εμφανίσει έναν πολύ μικρό αριθμό αντεστραμμένων τριγώνων, θα έχει δηλαδή κριθεί ακατάλληλο για χρήση με λογισμικό επίλυσης ΜΔΕ.



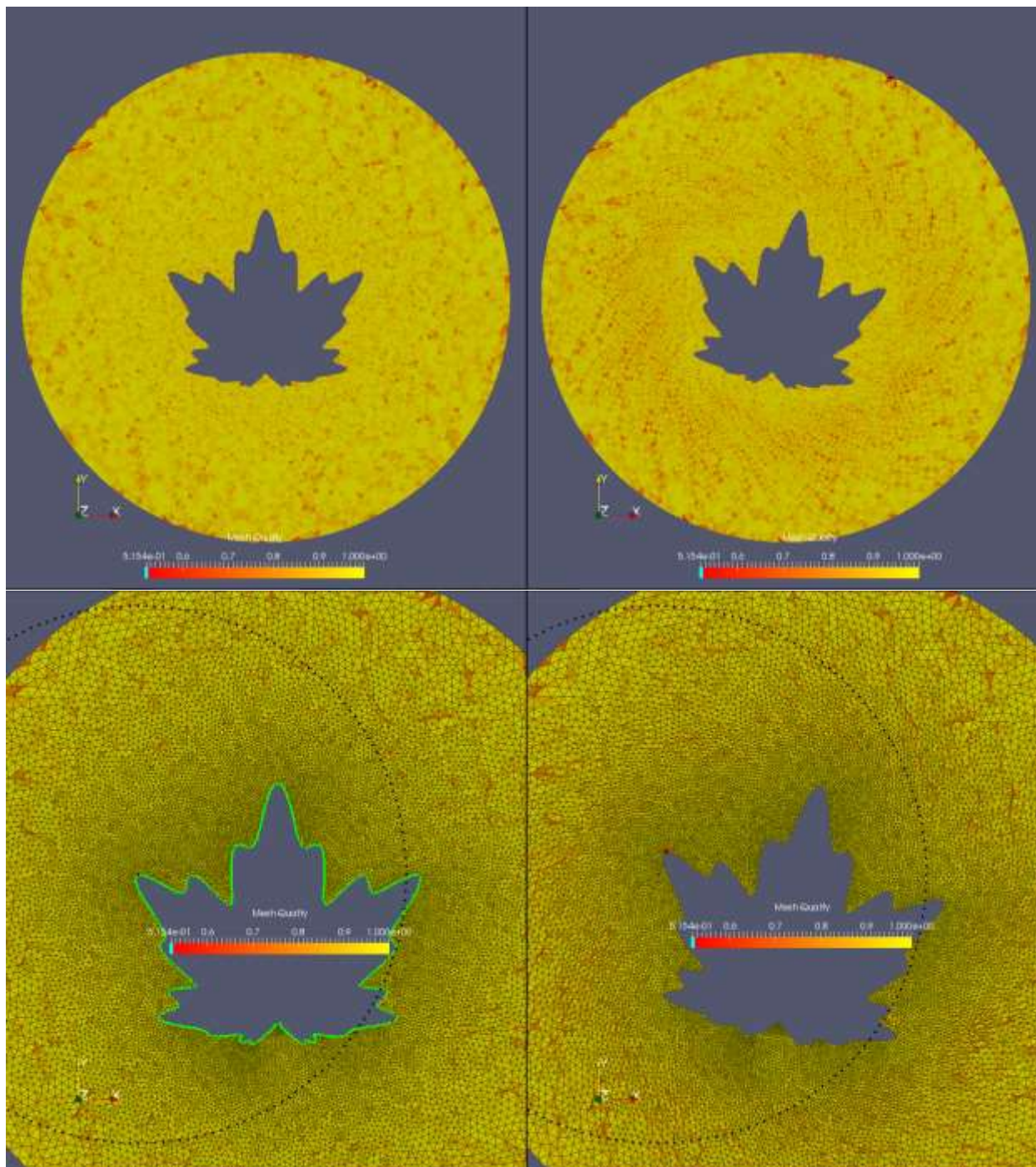
Σχήμα 5.8: **Αριστερά:** Το αρχικό πλέγμα. Με πράσινο χρώμα εμφανίζονται τα κέντρα του δικτύου (όλοι εκείνοι οι κόμβοι ή υποσύνολο αυτών, των οποίων η τιμή της μετατόπισης είναι γνωστή). Με μαύρο χρώμα εμφανίζεται ο κύκλος επιρροής (με ακτίνα επιρροής $R=3$) ενός εκ των κέντρων. **Δεξιά:** Το αναμορφωμένο πλέγμα και η ανεπιθύμητη παραμόρφωση του πλέγματος. Με μπλε χρώμα εμφανίζονται τα στοιχεία με χαμηλή τιμή μετρικής ποιότητας. Θα μπορούσαμε να παρατηρήσουμε την «συμπύεση» των πλεγματικών γραμμών που λαμβάνει χώρα μέσα στην περιοχή του μπλε.

5.2.3 Αποτελέσματα με τη $Wendland C^2$, $R = 9$

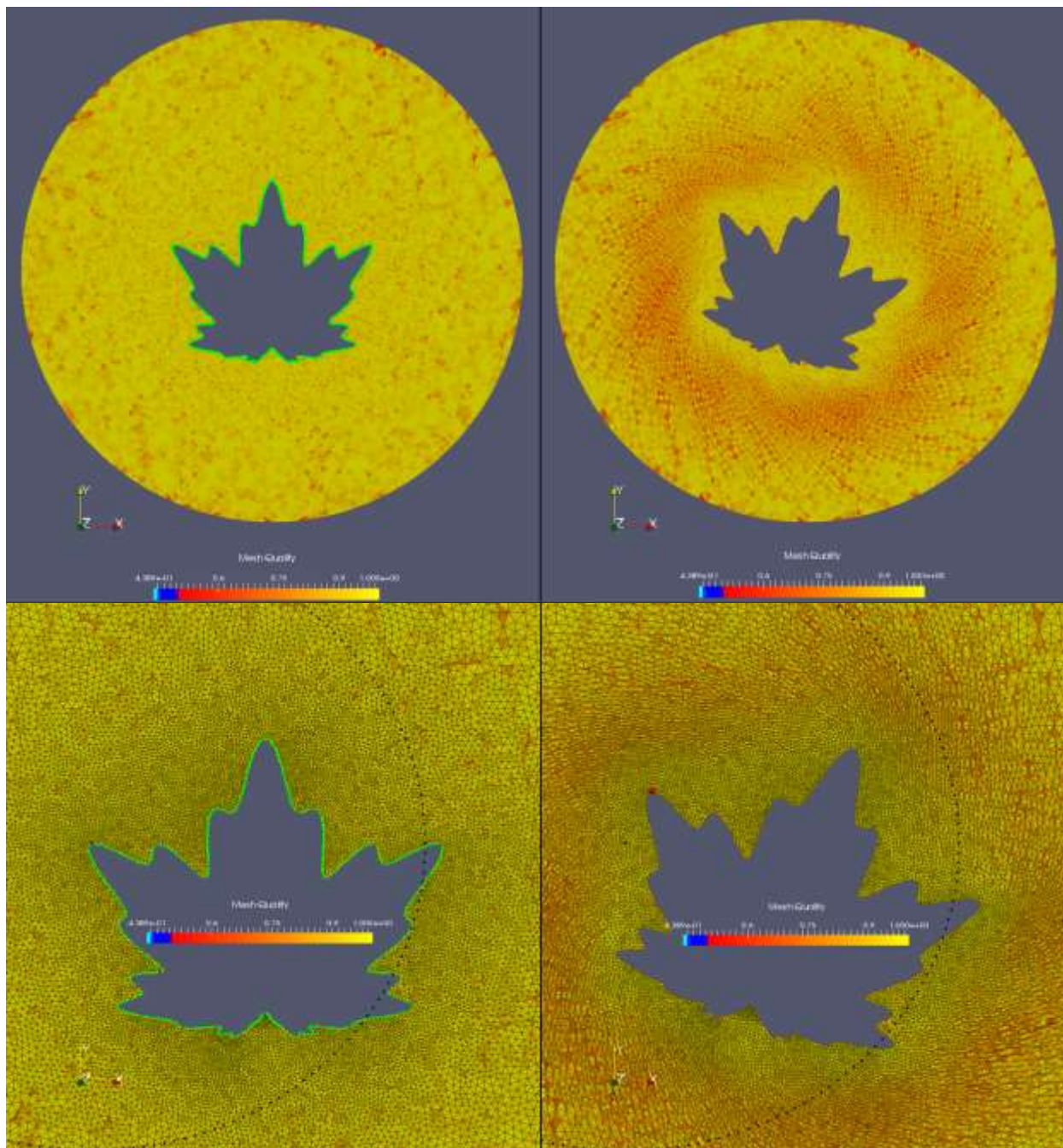
Η ακτίνα που επιλέχτηκε για την εφαρμογή της μεθόδου, είναι $R=9$. Όπως και στις προηγούμενες περιπτώσεις, έγιναν τα αντίστοιχα σχήματα και πίνακες, τα οποία φαίνονται στον πίνακα 5.3 και στα σχήματα 5.9 έως 5.15. Η επιλεγείσα ακτίνα είναι τέτοια ώστε περισσότερα κέντρα του δικτύου RBF να επηρεάζονται μεταξύ τους και ταυτόχρονα να επηρεάζουν περισσότερους κόμβους του εσωτερικού πλέγματος. Σε αυτή την περίπτωση, που η ακτίνα επιρροής είναι μεγαλύτερη, το πλέγμα άντεξε μεγαλύτερη γωνία στροφής, όπως αναμενόταν, πριν κριθεί ακατάλληλο.

θ (μοίρες)	0	10	20	30	40	50	60
\bar{q}_e	0.958755	0.943765	0.902786	0.843617	0.774818	0.703037	0.632358
σ_{qe}	0.0428128	0.0530309	0.081259	0.122306	0.169567	0.218115	0.265349
$q_{e,min}$	0.515426	0.516089	0.438899	0.325037	0.228654	0.148105	0.004881

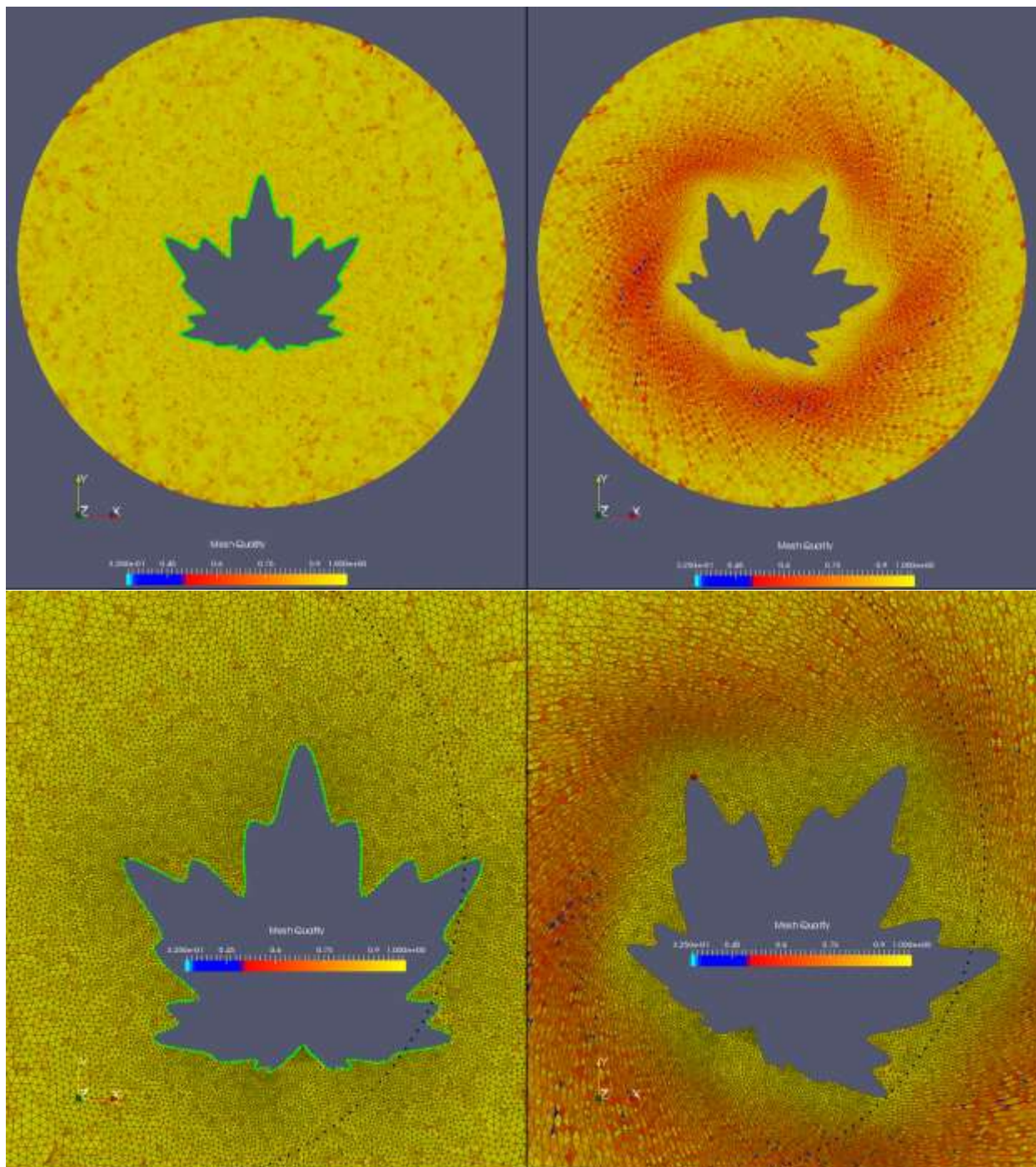
Πίνακας 5.3: Αποτελέσματα μέσης τιμής \bar{q}_e , τυπικής απόκλισης σ_{qe} και ελάχιστη τιμή $q_{e,min}$ της μετρικής ποιότητας του πλέγματος για την αρχική κατάσταση του πλέγματος και για γωνία στροφής της εσωτερικής γεωμετρίας. Ως συνάρτηση ενεργοποίησης είναι η $Wendland C^2$ με ακτίνα $R=9$.



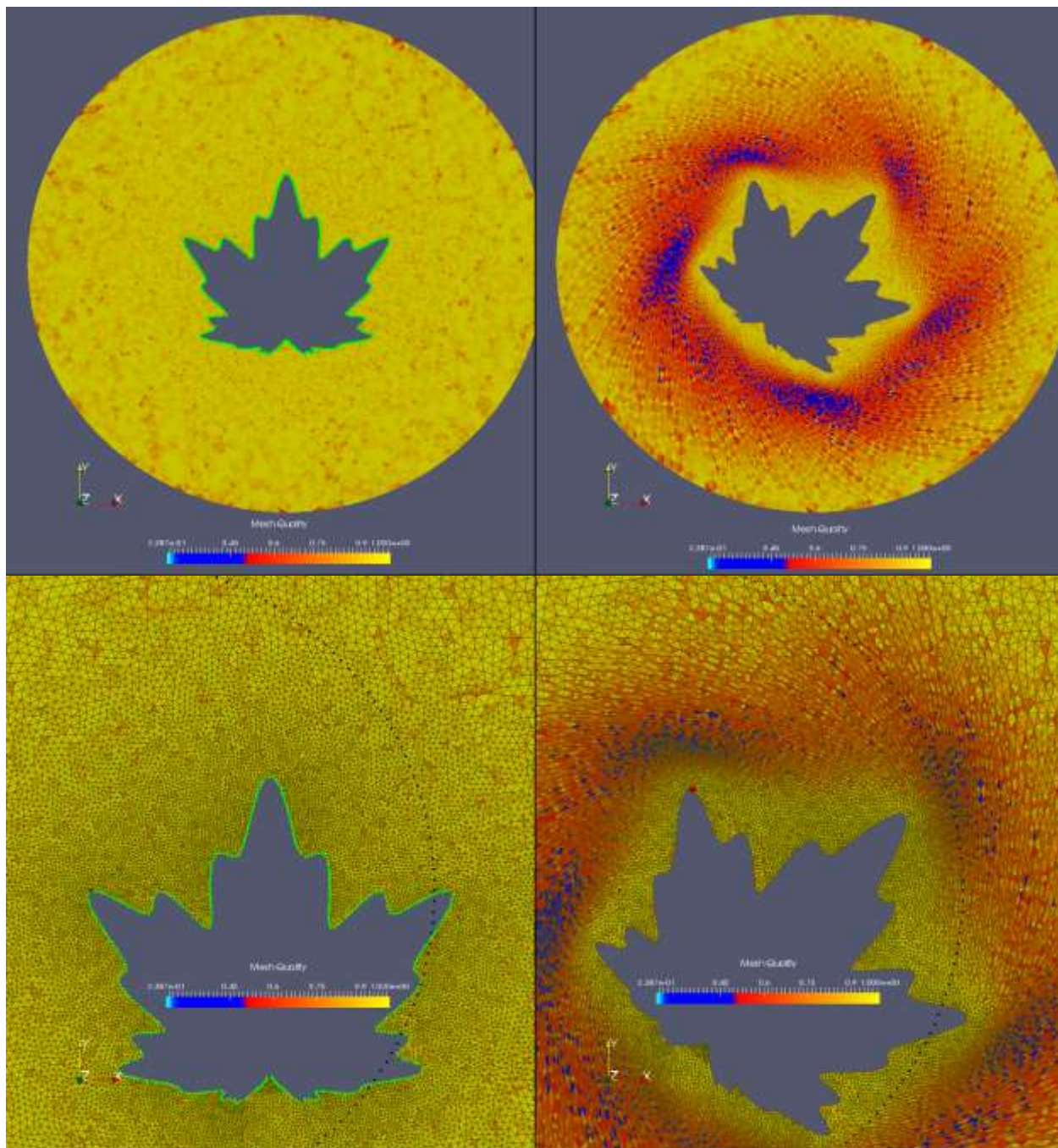
Σχήμα 5.9 Εφαρμογή της μεθόδου της μετακίνησης του πλέγματος με δίκτυο RBF με γωνία στροφής 10 μοιρών του περιγράμματος του φύλλου, με χρήση της Wendland C^2 με $R=9$ και τα αποτελέσματα στην μετρική ποιότητας. **Πάνω αριστερά:** Το αρχικό πλέγμα χωρίς πλεγματικές γραμμές. Η ποιότητα του κάθε τριγωνικού στοιχείου εμφανίζεται με διαφορετικό χρώμα. **Πάνω δεξιά:** Το αναμορφωμένο πλέγμα χωρίς πλεγματικές γραμμές για γωνία στροφής κατά 10 μοίρες και τα αποτελέσματα στην μετρική ποιότητας του πλέγματος. **Κάτω αριστερά:** Το αρχικό πλέγμα με εστίαση στην περιοχή του φύλλου. Με πράσινο χρώμα εμφανίζονται τα κέντρα του δικτύου (όλοι εκείνοι οι κόμβοι ή υποσύνολο αυτών, των οποίων η τιμή της μετατόπισης είναι γνωστή). Με μαύρο χρώμα εμφανίζεται ο κύκλος επιρροής (με ακτίνα επιρροής $R=9$) ενός εκ των κέντρων. **Κάτω δεξιά:** Το αναμορφωμένο πλέγμα. Με κόκκινο εμφανίζεται το μετατοπισμένο κέντρο.



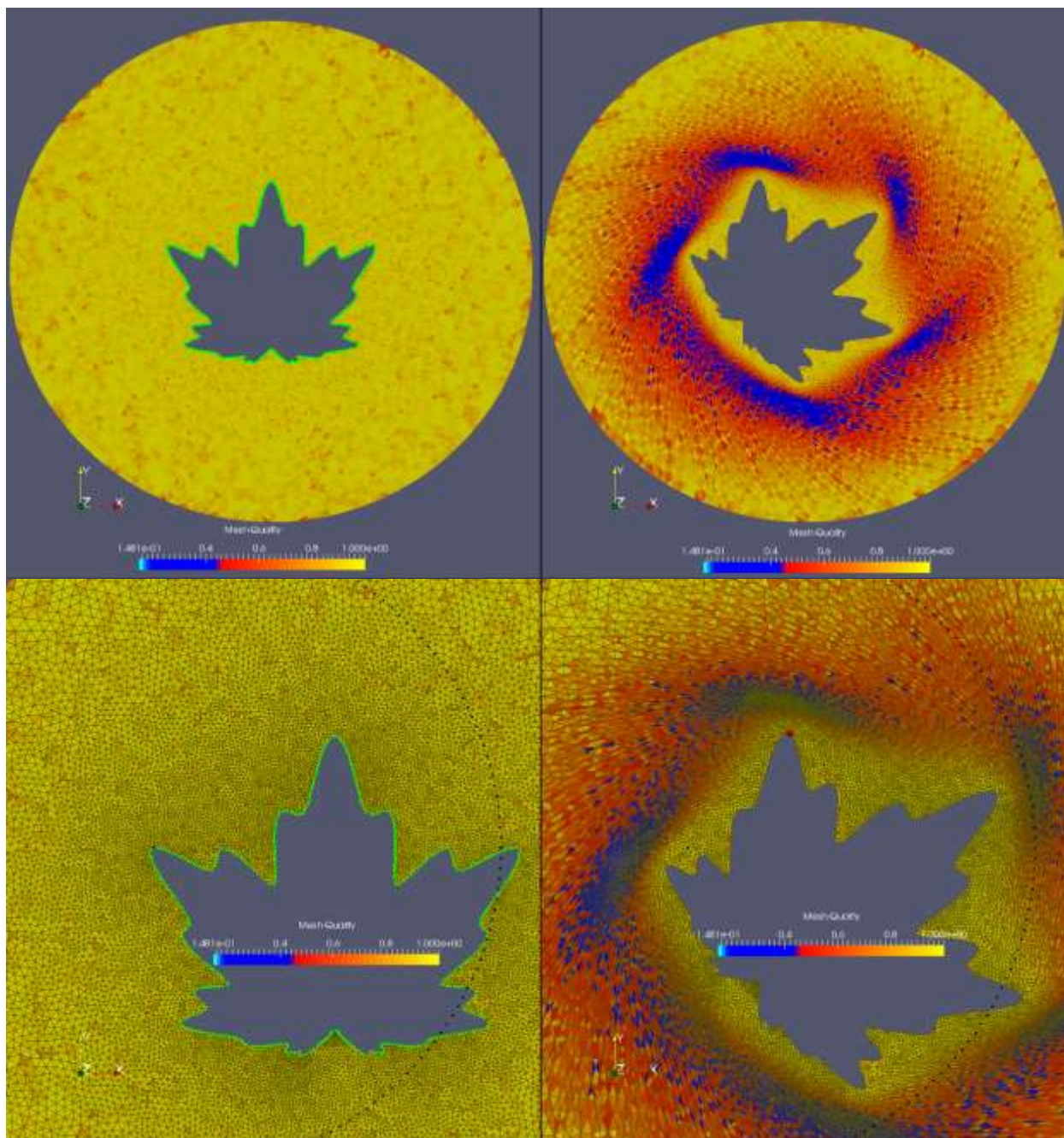
Σχήμα 5.10: Εφαρμογή της μεθόδου της μετακίνησης του πλέγματος με δίκτυο RBF με γωνία στροφής 20 μοιρών του περιγράμματος του φύλλου, με χρήση της Wendland C^2 με $R=9$ και τα αποτελέσματα στην μετρική ποιότητας. **Πάνω αριστερά:** Το αρχικό πλέγμα χωρίς πλεγματικές γραμμές. Η ποιότητα του κάθε τριγωνικού στοιχείου εμφανίζεται με διαφορετικό χρώμα. **Πάνω δεξιά:** Το αναμορφωμένο πλέγμα χωρίς πλεγματικές γραμμές για γωνία στροφής κατά 20 μοίρες και τα αποτελέσματα στην μετρική ποιότητας του πλέγματος. **Κάτω αριστερά:** Το αρχικό πλέγμα με εστίαση στην περιοχή του φύλλου. Με πράσινο χρώμα εμφανίζονται τα κέντρα του δικτύου (όλοι εκείνοι οι κόμβοι ή υποσύνολο αυτών, των οποίων η τιμή της μετατόπισης είναι γνωστή). Με μαύρο χρώμα εμφανίζεται ο κύκλος επιρροής (με ακτίνα επιρροής $R=9$) ενός εκ των κέντρων. **Κάτω δεξιά:** Το αναμορφωμένο πλέγμα. Με κόκκινο εμφανίζεται το μετατοπισμένο κέντρο. Μπορούμε να παρατηρήσουμε ότι αρχίζουν να εμφανίζονται τα πρώτα «ανεπιθύμητα» στοιχεία.



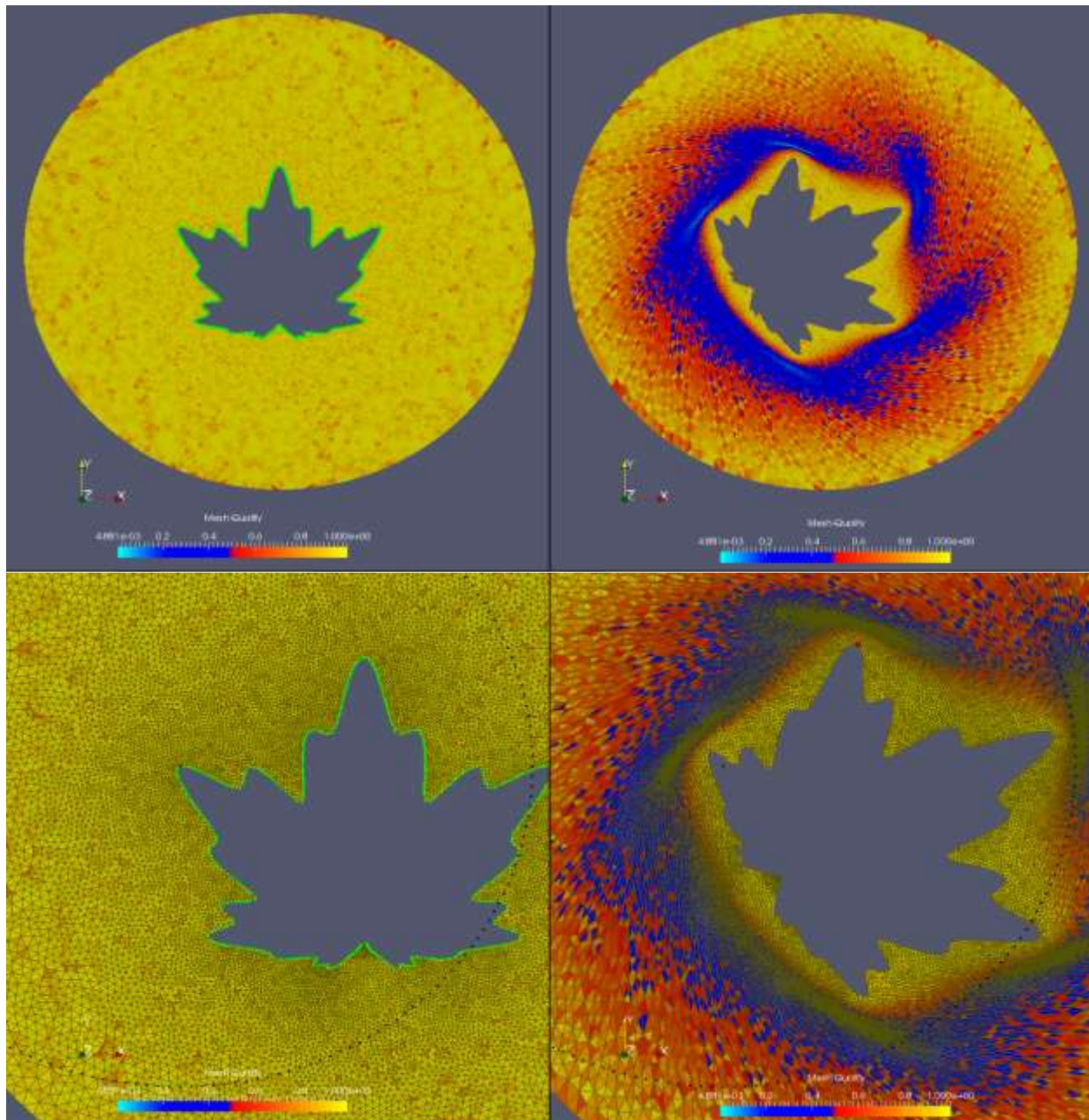
Σχήμα 5.11: Εφαρμογή της μεθόδου της μετακίνησης του πλέγματος με δίκτυο RBF με γωνία στροφής 30 μοιρών του περιγράμματος του φύλλου, με χρήση της Wendland C^2 με $R=9$ και τα αποτελέσματα στην μετρική ποιότητας. **Πάνω αριστερά:** Το αρχικό πλέγμα χωρίς πλεγματικές γραμμές. Η ποιότητα του κάθε τριγωνικού στοιχείου εμφανίζεται με διαφορετικό χρώμα. **Πάνω δεξιά:** Το αναμορφωμένο πλέγμα χωρίς πλεγματικές γραμμές για γωνία στροφής κατά 30 μοίρες και τα αποτελέσματα στην μετρική ποιότητας του πλέγματος. **Κάτω αριστερά:** Το αρχικό πλέγμα με εστίαση στην περιοχή του φύλλου. Με πράσινο χρώμα εμφανίζονται τα κέντρα του δικτύου (όλοι εκείνοι οι κόμβοι ή υποσύνολο αυτών, των οποίων η τιμή της μετατόπισης είναι γνωστή). Με μαύρο χρώμα εμφανίζεται ο κύκλος επιρροής (με ακτίνα επιρροής $R=9$) ενός εκ των κέντρων. **Κάτω δεξιά:** Το αναμορφωμένο πλέγμα. Με κόκκινο εμφανίζεται το μετατοπισμένο κέντρο. Με μπλε χρώμα εμφανίζονται τα στοιχεία με τιμή μετρικής ποιότητας χαμηλότερη του 0.5.



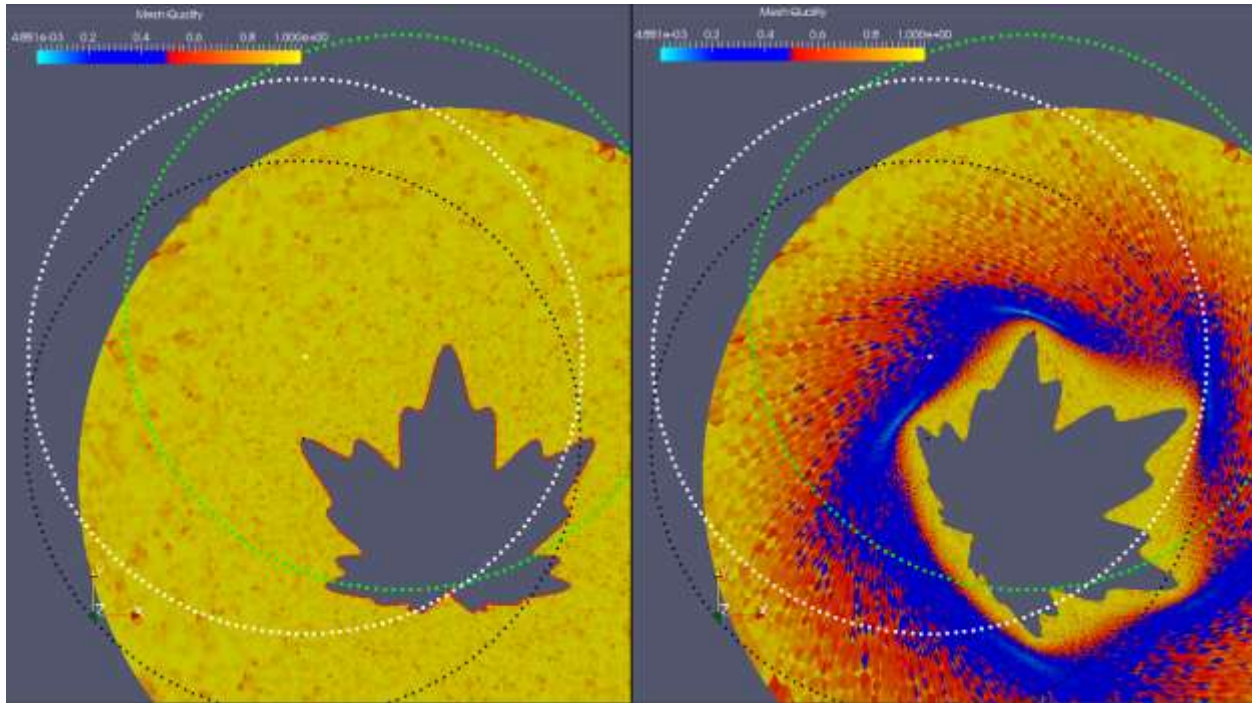
Σχήμα 5.12: Εφαρμογή της μεθόδου της μετακίνησης του πλέγματος με δίκτυο RBF με γωνία στροφής 40 μοιρών του περιγράμματος του φύλλου, με χρήση της Wendland C^2 με $R=9$ και τα αποτελέσματα στην μετρική ποιότητας. **Πάνω αριστερά:** Το αρχικό πλέγμα χωρίς πλεγματικές γραμμές. Η ποιότητα του κάθε τριγωνικού στοιχείου εμφανίζεται με διαφορετικό χρώμα. **Πάνω δεξιά:** Το αναμορφωμένο πλέγμα χωρίς πλεγματικές γραμμές για γωνία στροφής κατά 40 μοίρες και τα αποτελέσματα στην μετρική ποιότητας του πλέγματος. **Κάτω αριστερά:** Το αρχικό πλέγμα με εστίαση στην περιοχή του φύλλου. Με πράσινο χρώμα εμφανίζονται τα κέντρα του δικτύου (όλοι εκείνοι οι κόμβοι ή υποσύνολο αυτών, των οποίων η τιμή της μετατόπισης είναι γνωστή). Με μαύρο χρώμα εμφανίζεται ο κύκλος επιρροής (με ακτίνα επιρροής $R=9$) ενός εκ των κέντρων. **Κάτω δεξιά:** Το αναμορφωμένο πλέγμα. Με κόκκινο εμφανίζεται το μετατοπισμένο κέντρο. Με μπλε χρώμα εμφανίζονται τα στοιχεία με τιμή μετρικής ποιότητας χαμηλότερη του 0.5. Θα μπορούσαμε να παρατηρήσουμε την «συμπύκνωση» των πλεγματικών γραμμών που λαμβάνει χώρα μέσα στην περιοχή του μπλε.



Σχήμα 5.13: Εφαρμογή της μεθόδου της μετακίνησης του πλέγματος με δίκτυο RBF με γωνία στροφής 50 μοιρών του περιγράμματος του φύλλου, με χρήση της Wendland C^2 με $R=9$ και τα αποτελέσματα στην μετρική ποιότητας. **Πάνω αριστερά:** Το αρχικό πλέγμα χωρίς πλεγματικές γραμμές. Η ποιότητα του κάθε τριγωνικού στοιχείου εμφανίζεται με διαφορετικό χρώμα. **Πάνω δεξιά:** Το αναμορφωμένο πλέγμα χωρίς πλεγματικές γραμμές για γωνία στροφής κατά 50 μοίρες και τα αποτελέσματα στην μετρική ποιότητας του πλέγματος. **Κάτω αριστερά:** Το αρχικό πλέγμα με εστίαση στην περιοχή του φύλλου. Με πράσινο χρώμα εμφανίζονται τα κέντρα του δικτύου (όλοι εκείνοι οι κόμβοι ή υποσύνολο αυτών, των οποίων η τιμή της μετατόπισης είναι γνωστή). Με μαύρο χρώμα εμφανίζεται ο κύκλος επιρροής (με ακτίνα επιρροής $R=9$) ενός εκ των κέντρων. **Κάτω δεξιά:** Το αναμορφωμένο πλέγμα. Με κόκκινο εμφανίζεται το μετατοπισμένο κέντρο. Με μπλε χρώμα εμφανίζονται τα στοιχεία με τιμή μετρικής ποιότητας χαμηλότερη του 0.5. Θα μπορούσαμε να παρατηρήσουμε την «συμπίεση» των πλεγματικών γραμμών που λαμβάνει χώρα μέσα στην περιοχή του μπλε.



Σχήμα 5.14: Εφαρμογή της μεθόδου της μετακίνησης του πλέγματος με δίκτυο RBF με γωνία στροφής 60 μοιρών του περιγράμματος του φύλλου, με χρήση της Wendland C^2 με $R=9$ και τα αποτελέσματα στην μετρική ποιότητας. **Πάνω αριστερά:** Το αρχικό πλέγμα χωρίς πλεγματικές γραμμές. Η ποιότητα του κάθε τριγωνικού στοιχείου εμφανίζεται με διαφορετικό χρώμα. **Πάνω δεξιά:** Το αναμορφωμένο πλέγμα χωρίς πλεγματικές γραμμές για γωνία στροφής κατά 60 μοίρες και τα αποτελέσματα στην μετρική ποιότητας του πλέγματος. **Κάτω αριστερά:** Το αρχικό πλέγμα με εστίαση στην περιοχή του φύλλου. Με πράσινο χρώμα εμφανίζονται τα κέντρα του δικτύου (όλοι εκείνοι οι κόμβοι ή υποσύνολο αυτών, των οποίων η τιμή της μετατόπισης είναι γνωστή). Με μαύρο χρώμα εμφανίζεται ο κύκλος επιρροής (με ακτίνα επιρροής $R=9$) ενός εκ των κέντρων. **Κάτω δεξιά:** Το αναμορφωμένο πλέγμα. Με κόκκινο εμφανίζεται το μετατοπισμένο κέντρο. Με μπλε χρώμα εμφανίζονται τα στοιχεία με τιμή μετρικής ποιότητας χαμηλότερη του 0.5. Θα μπορούσαμε να παρατηρήσουμε την «συμπίεση» των πλεγματικών γραμμών που λαμβάνει χώρα μέσα στην περιοχή του μπλε. Σε αυτή την περίπτωση, για γωνία μεγαλύτερη των 60 μοιρών, το πλέγμα θα έχει εμφανίσει έναν πολύ μικρό αριθμό ανεστραμμένων τριγώνων, θα έχει δηλαδή κριθεί ακατάλληλο για χρήση με λογισμικό επίλυσης ΜΔΕ.



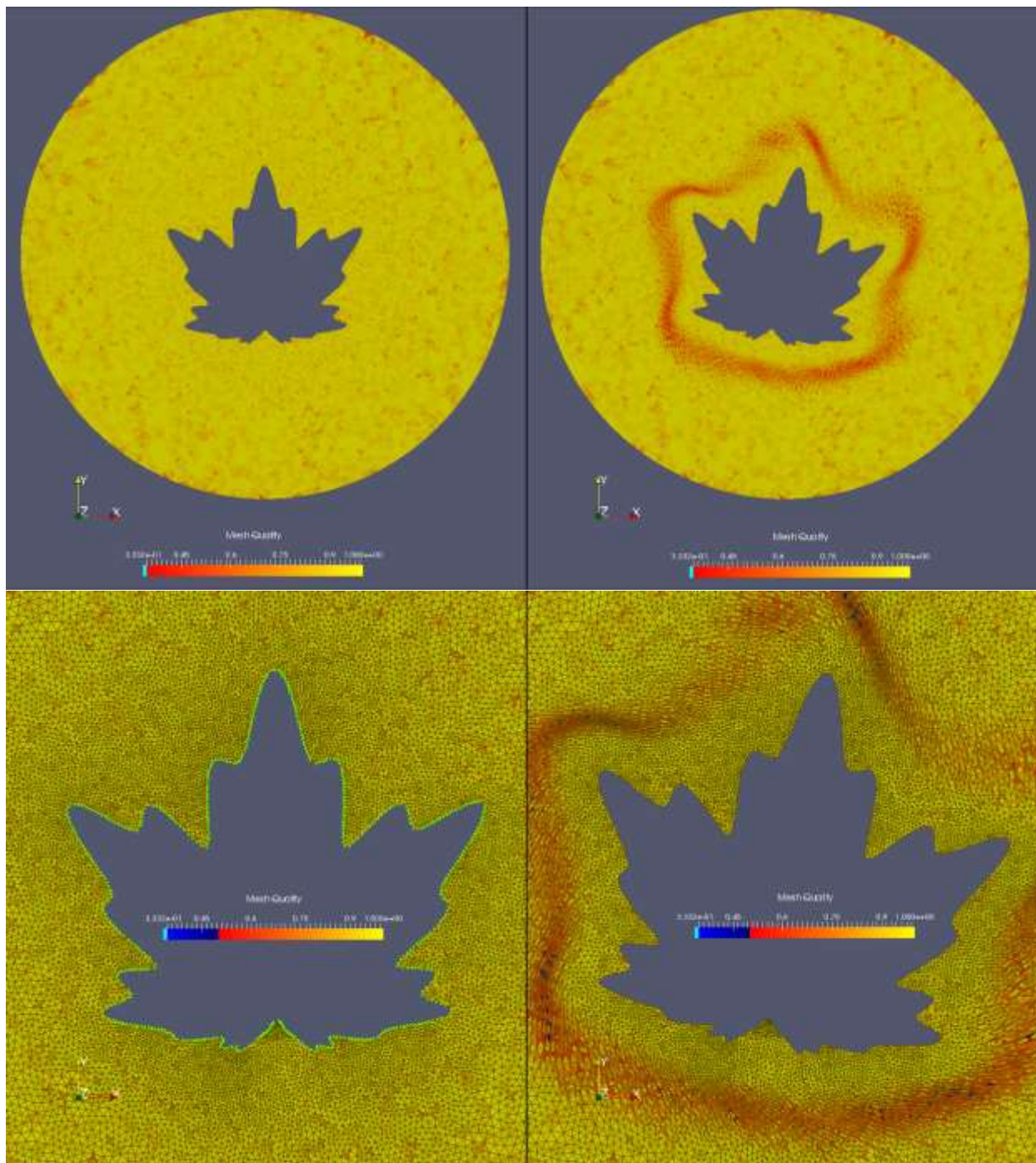
Σχήμα 5.15: **Αριστερά:** Το αρχικό πλέγμα. Με κόκκινο χρώμα εμφανίζονται τα κέντρα του δικτύου (όλοι εκείνοι οι κόμβοι ή υποσύνολο αυτών, των οποίων η τιμή της μετατόπισης είναι γνωστή). Με μαύρο χρώμα εμφανίζεται ο κύκλος επιρροής (με ακτίνα επιρροής $R=9$) ενός εκ των κέντρων. Με λευκό και πράσινο χρώμα εμφανίζεται ένας εσωτερικός κόμβος (με τον αντίστοιχο κύκλο επιρροής) πριν και μετά την αναμόρφωση του πλέγματος. **Δεξιά:** Το αναμορφωμένο πλέγμα και η ανεπιθύμητη παραμόρφωση του πλέγματος. Με κόκκινο χρώμα εμφανίζεται το μετατοπισμένο κέντρο. Με μπλε χρώμα εμφανίζονται τα στοιχεία με χαμηλή τιμή μετρικής ποιότητας (<0.5). Θα μπορούσαμε να παρατηρήσουμε τις περιοχές δημιουργία των «οριακά» ανεστραμμένων τριγωνικών στοιχείων που απεικονίζονται με λευκό χρώμα.

5.2.4 Αποτελέσματα με τη συνάρτηση ολικής επιρροής Gauss

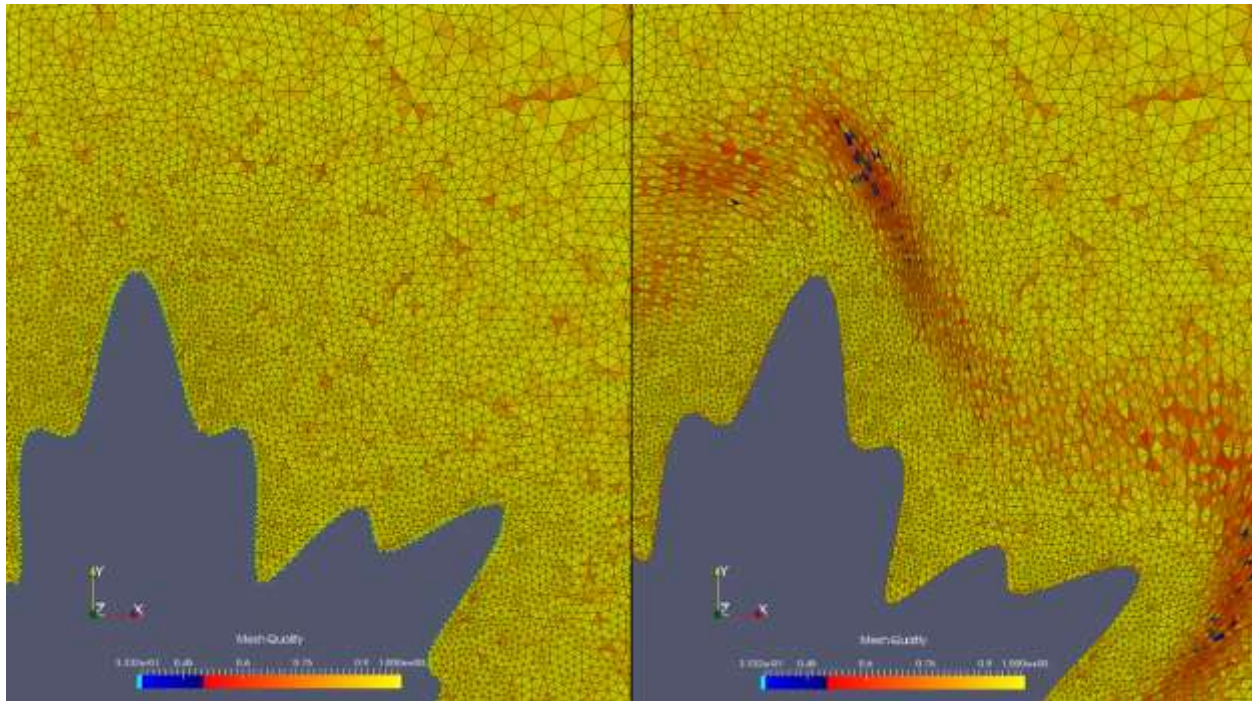
Στην ενότητα αυτή επιλέχθηκε ως συνάρτηση ενεργοποίησης του δικτύου RBF η συνάρτηση Gauss (σχέση 5.2), μια συνάρτηση ολικής επιρροής, με σκοπό να παρουσιαστούν και εδώ η μέση τιμή (\bar{q}_e), η τυπική απόκλιση (σ_{q_e}) και η ελάχιστη τιμή ($q_{e,min}$) της μετρικής ποιότητας του πλέγματος. Στα σχήματα 5.16 ως 5.17 φαίνεται το φύλλο πλατάνου σε τυχαία επιλεγμένη γωνία στροφής μέχρι την οποία δεν έχει εμφανίσει αντεστραμμένα τριγωνικά στοιχεία αλλά η ελάχιστη τιμή της μετρικής ποιότητας έχει αρκετά χαμηλή τιμή, καθώς και το πως διαμορφώθηκε το πλέγμα μετά την εφαρμογή της μεθόδου. Η στροφή της εσωτερικής γεωμετρίας σταμάτησε στις 14 μοίρες, γιατί το πλέγμα εμφάνισε πολύ χαμηλή τιμή ποιότητας πλέγματος.

θ (μοίρες)	0	10	14
\bar{q}_e	0.958755	0.915115	0.882724
σ_{q_e}	0.0428128	0.0995814	0.152845
$q_{e,min}$	0.515426	0.33321	0.0516738

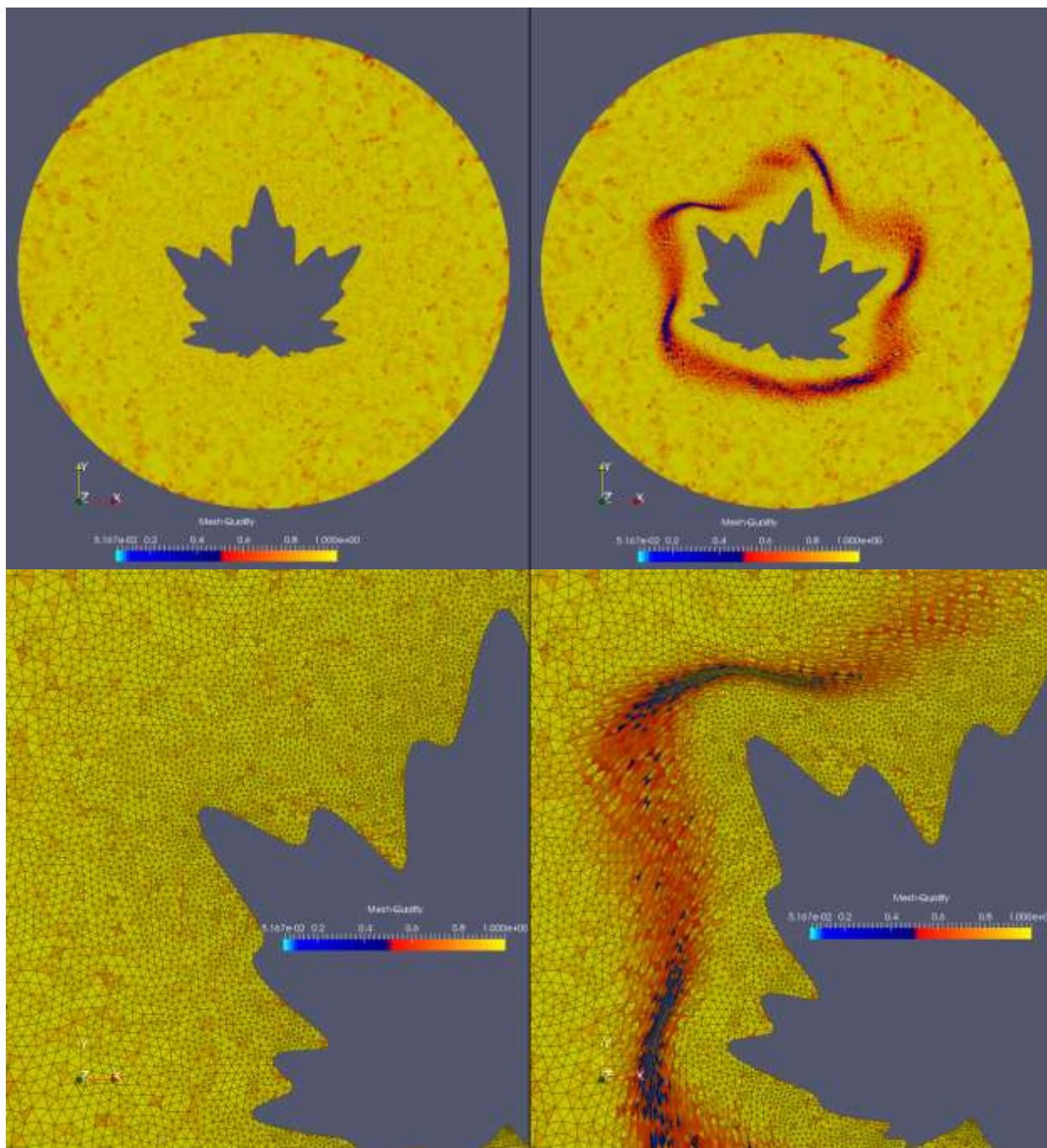
Πίνακας 5.4: Αποτελέσματα μέσης τιμής \bar{q}_e , τυπικής απόκλισης σ_{q_e} και ελάχιστη τιμή $q_{e,min}$ της μετρικής ποιότητας του πλέγματος για την αρχική κατάσταση του πλέγματος και για γωνία στροφής της εσωτερικής γεωμετρίας. Ως συνάρτηση ενεργοποίησης είναι η Gauss.



Σχήμα 5.16: Εφαρμογή της μεθόδου της μετακίνησης του πλέγματος με δίκτυο RBF με γωνία στροφής 10 μοιρών του περιγράμματος του φύλλου, με χρήση της Gauss και τα αποτελέσματα στην μετρική ποιότητας. **Πάνω αριστερά:** Το αρχικό πλέγμα χωρίς πλεγματικές γραμμές. Η ποιότητα του κάθε τριγωνικού στοιχείου εμφανίζεται με διαφορετικό χρώμα. **Πάνω δεξιά:** Το αναμορφωμένο πλέγμα χωρίς πλεγματικές γραμμές για γωνία στροφής κατά 10 μοίρες και τα αποτελέσματα στην μετρική ποιότητας του πλέγματος. **Κάτω αριστερά:** Το αρχικό πλέγμα με εστίαση στην περιοχή του φύλλου. Με πράσινο χρώμα εμφανίζονται τα κέντρα του δικτύου (όλοι εκείνοι οι κόμβοι ή υποσύνολο αυτών, των οποίων η τιμή της μετατόπισης είναι γνωστή). **Κάτω δεξιά:** Το αναμορφωμένο πλέγμα και η ανεπιθύμητη παραμόρφωση του πλέγματος.



Σχήμα 5.17: **Αριστερά:** Το αρχικό πλέγμα. Με πράσινο χρώμα εμφανίζονται τα κέντρα του δικτύου (όλοι εκείνοι οι κόμβοι ή υποσύνολο αυτών, των οποίων η τιμή της μετατόπισης είναι γνωστή). Με μαύρο χρώμα εμφανίζεται ο κύκλος επιρροής (με ακτίνα επιρροής $R=3$) ενός εκ των κέντρων. **Δεξιά:** Το αναμορφωμένο πλέγμα και η ανεπιθύμητη παραμόρφωση του πλέγματος. Με πράσινο χρώμα εμφανίζεται το μετατοπισμένο κέντρο.



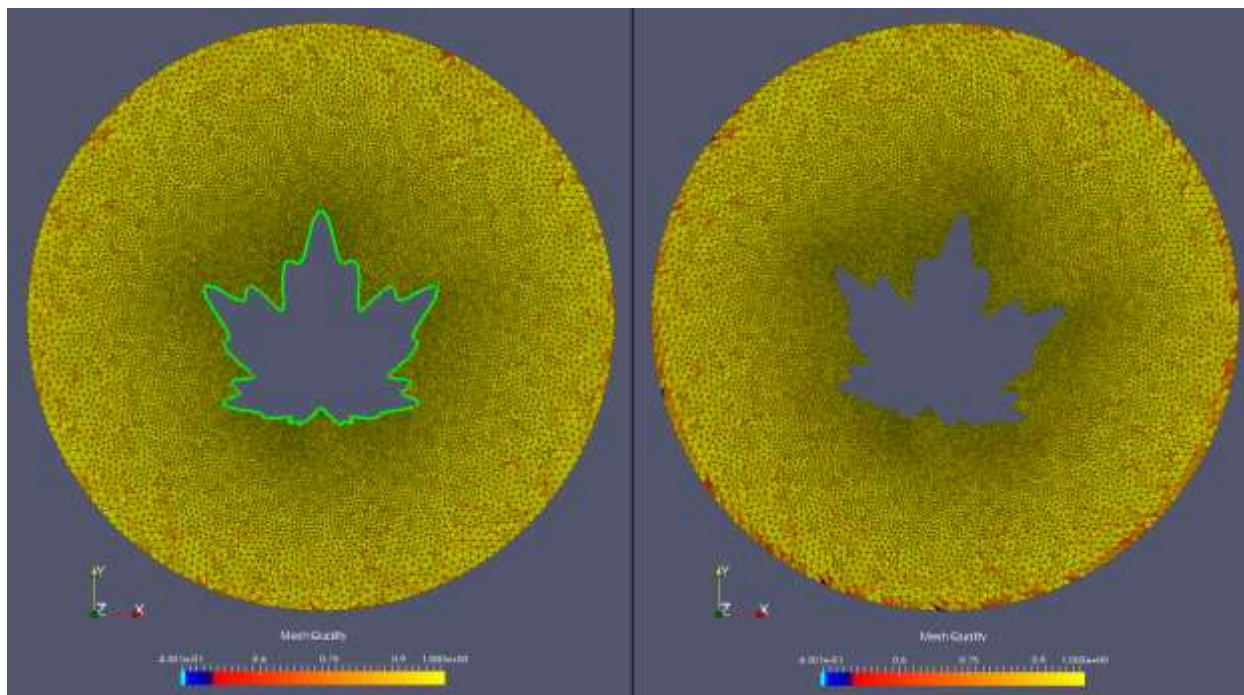
Σχήμα 5.18: Εφαρμογή της μεθόδου της μετακίνησης του πλέγματος με δίκτυο RBF με γωνία στροφής 18 μοιρών του περιγράμματος του φύλλου, με χρήση της Gauss και τα αποτελέσματα στην μετρική ποιότητας. **Πάνω αριστερά:** Το αρχικό πλέγμα χωρίς πλεγματικές γραμμές. Η ποιότητα του κάθε τριγωνικού στοιχείου εμφανίζεται με διαφορετικό χρώμα. **Πάνω δεξιά:** Το αναμορφωμένο πλέγμα χωρίς πλεγματικές γραμμές για γωνία στροφής κατά 18 μοίρες και τα αποτελέσματα στην μετρική ποιότητας του πλέγματος. **Κάτω αριστερά:** Το αρχικό πλέγμα με εστίαση στην περιοχή του φύλλου. Με πράσινο χρώμα εμφανίζονται τα κέντρα του δικτύου (όλοι εκείνοι οι κόμβοι ή υποσύνολο αυτών, των οποίων η τιμή της μετατόπισης είναι γνωστή). **Κάτω δεξιά:** Το αναμορφωμένο πλέγμα και η ανεπιθύμητη παραμόρφωση του πλέγματος. Σε αυτή την περίπτωση, για γωνία μεγαλύτερη των 18 μοιρών, το πλέγμα θα έχει εμφανίσει έναν πολύ μικρό αριθμό αντεστραμμένων τριγώνων, θα έχει δηλαδή κριθεί ακατάλληλο για χρήση με λογισμικό επίλυσης ΜΔΕ.

5.2.5 Αποτελέσματα με τη συνάρτηση ολικής επιρροής Inverse Multiquadric Biharmonics

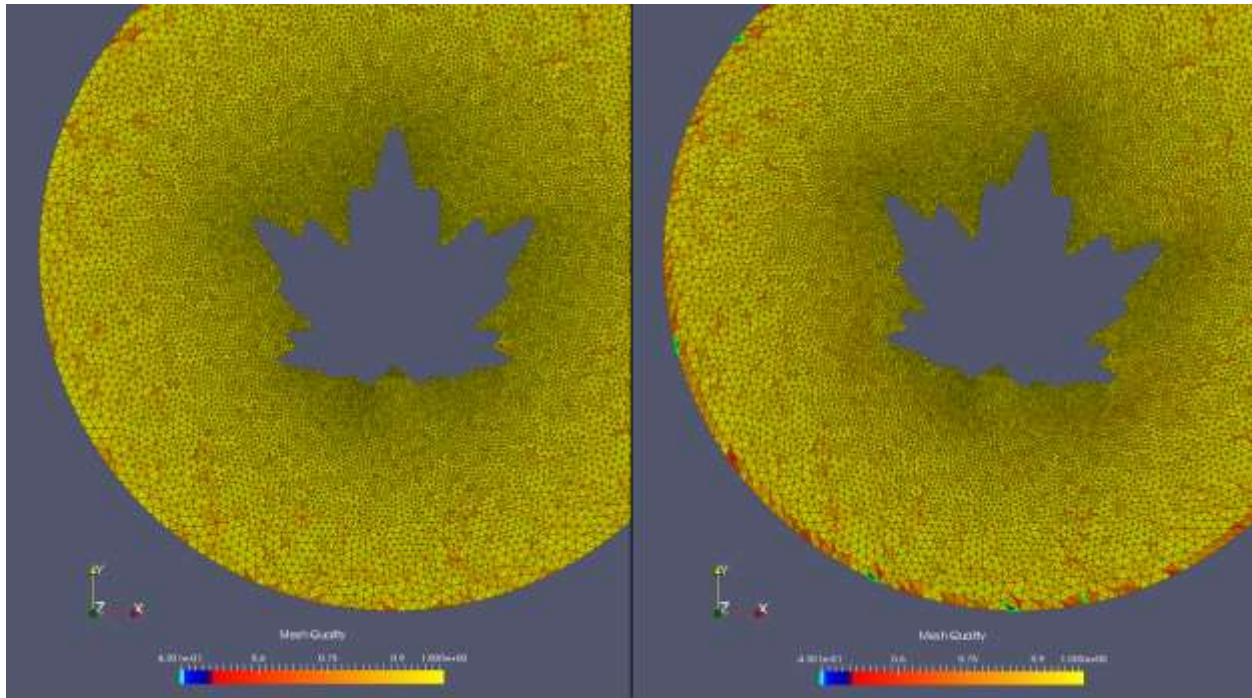
Στην ενότητα αυτή επιλέχθηκε ως συνάρτηση ενεργοποίησης του δικτύου RBF η συνάρτηση Inverse Multiquadric Biharmonics (σχέση 5.3), μια συνάρτηση ολικής επιρροής, με σκοπό να παρουσιαστούν και εδώ η μέση τιμή (\bar{q}_e), η τυπική απόκλιση (σ_{q_e}) και η ελάχιστη τιμή ($q_{e,min}$) της μετρικής ποιότητας του πλέγματος, τα οποία φαίνονται στον πίνακα 5.5. Στα σχήματα 5.19 ως 5.24 φαίνεται το φύλλο πλατάνου σε τυχαία επιλεγμένη γωνία στροφής μέχρι την οποία δεν έχει εμφανίσει αντεστραμμένα τριγωνικά στοιχεία αλλά η ελάχιστη τιμή της μετρικής ποιότητας έχει αρκετά χαμηλή τιμή, καθώς και το πως διαμορφώθηκε το πλέγμα μετά την εφαρμογή της μεθόδου. Η στροφή της εσωτερικής γεωμετρίας σταμάτησε στις 30 μοίρες, γιατί το πλέγμα εμφάνισε πολύ χαμηλή τιμή ποιότητας πλέγματος.

θ (μοίρες)	0	10	20	30
\bar{q}_e	0.958755	0.941092	0.893278	0.840448
σ_{q_e}	0.0428128	0.057302	0.0986602	0.145307
$q_{e,min}$	0.515426	0.43011	0.109917	0.00939905

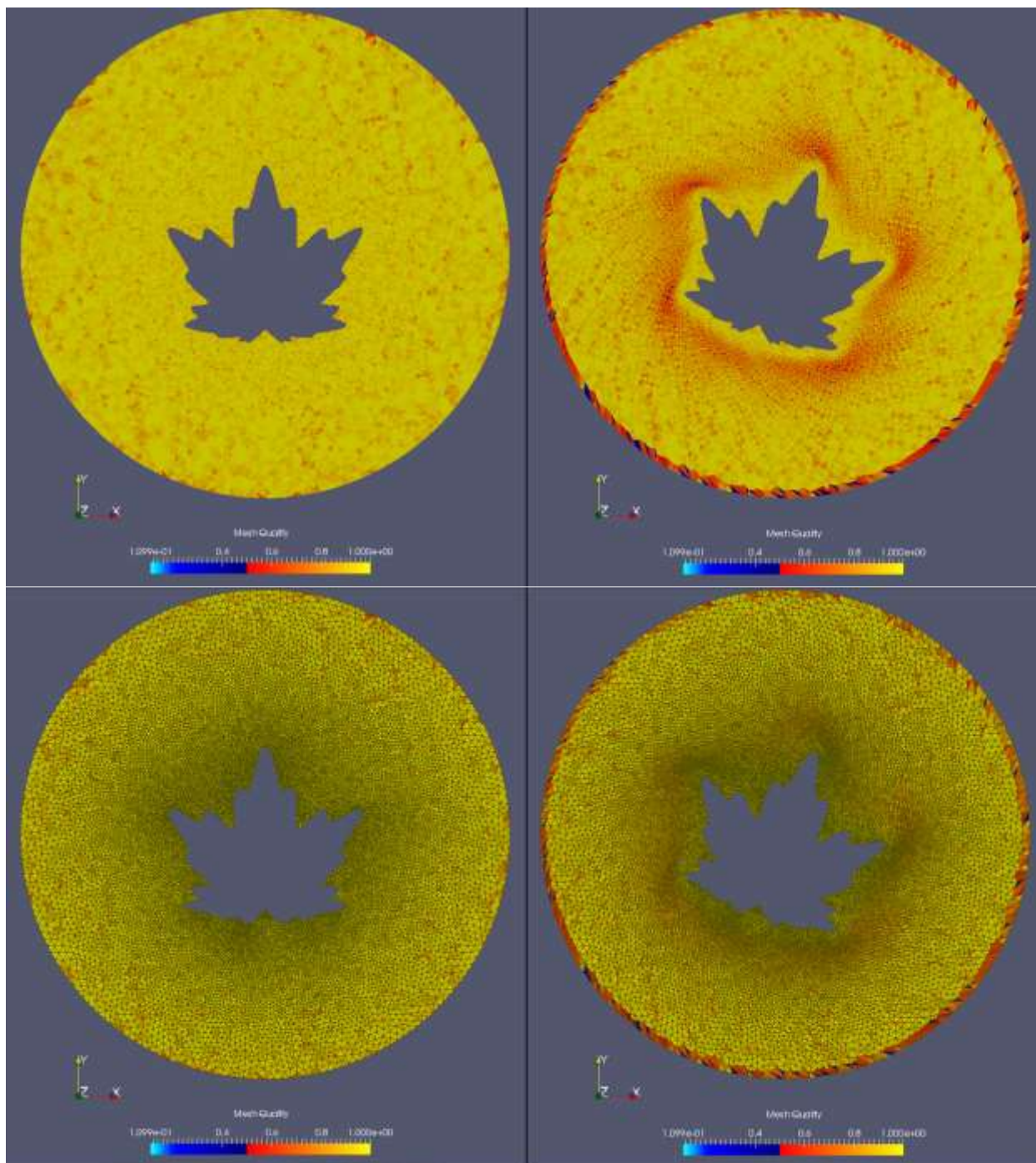
Πίνακας 5.5: Αποτελέσματα μέσης τιμής \bar{q}_e , τυπικής απόκλισης σ_{q_e} και ελάχιστη τιμή $q_{e,min}$ της μετρικής ποιότητας του πλέγματος για την αρχική κατάσταση του πλέγματος και για γωνία στροφής της εσωτερικής γεωμετρίας. Ως συνάρτηση ενεργοποίησης είναι η Inverse Multiquadric Biharmonics.



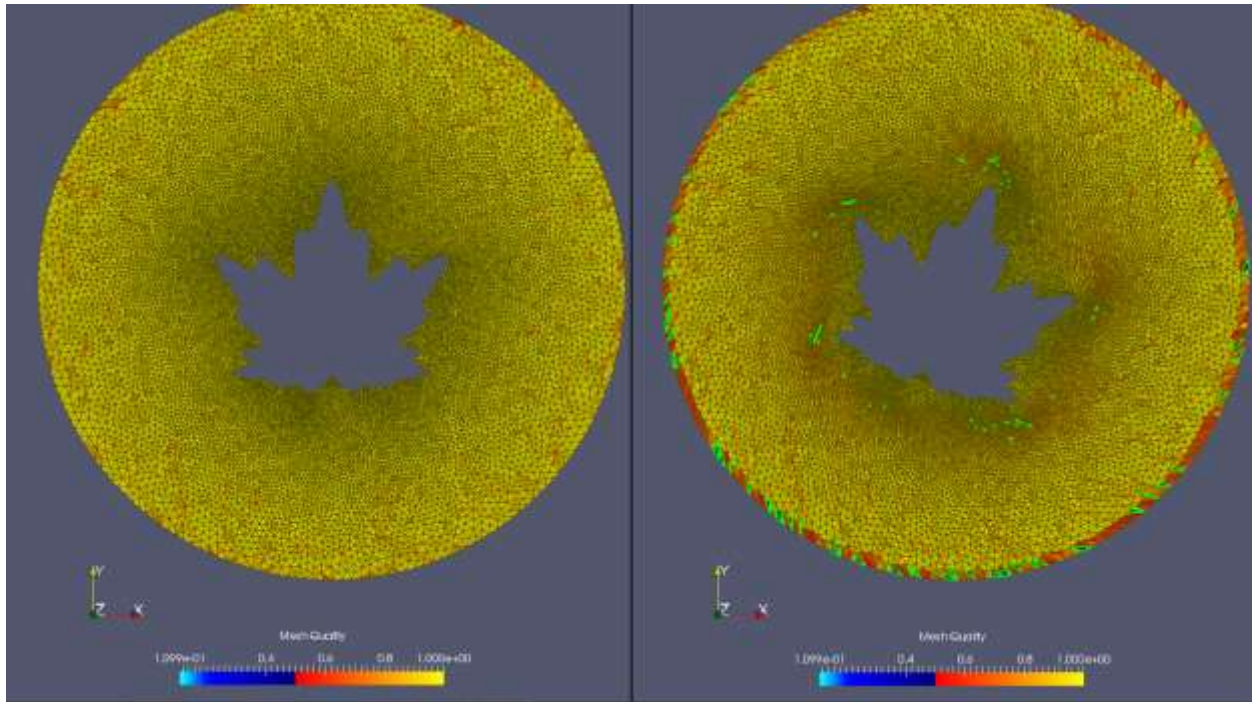
Σχήμα 5.19: Εφαρμογή της μεθόδου της μετακίνησης του πλέγματος με δίκτυο RBF με γωνία στροφής 10 μοιρών του περιγράμματος του φύλλου, με χρήση της Inverse Multiquadric Biharmonics και τα αποτελέσματα στην μετρική ποιότητας. **Αριστερά:** Το αρχικό πλέγμα. Με πράσινο χρώμα εμφανίζονται τα κέντρα του δικτύου (όλοι εκείνοι οι κόμβοι ή υποσύνολο αυτών, των οποίων η τιμή της μετατόπισης είναι γνωστή). **Δεξιά:** Το αναμορφωμένο. Μπορούμε να παρατηρήσουμε την ολική επίδραση της Inverse Multiquadric Biharmonics αφού βλέπουμε ότι επηρεάζει έντονα τους εξωτερικούς οριακούς κόμβους.



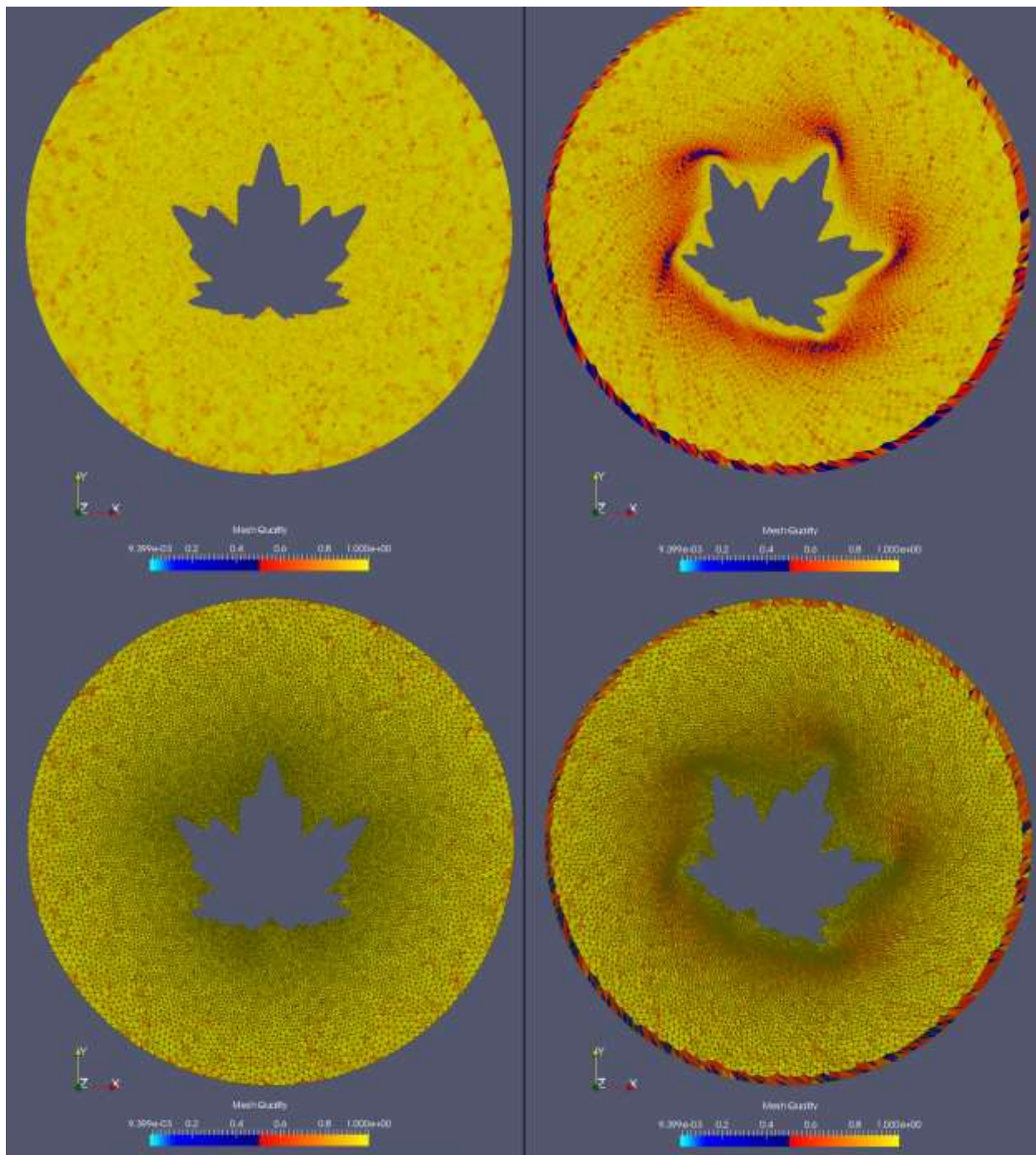
Σχήμα 5.20: **Αριστερά:** Το αρχικό πλέγμα. **Δεξιά:** Το αναμορφωμένο πλέγμα για γωνία 10 μοιρών και η ανεπιθύμητη παραμόρφωση του πλέγματος. Με πράσινο χρώμα εμφανίζονται τα τριγωνικά στοιχεία με τιμή μετρικής ποιότητας χαμηλότερη του 0.5. Μπορούμε να παρατηρήσουμε την ολική επίδραση της *Inverse Multiquadric Biharmonics* αφού βλέπουμε ότι επηρεάζει έντονα τους εξωτερικούς οριακούς κόμβους.



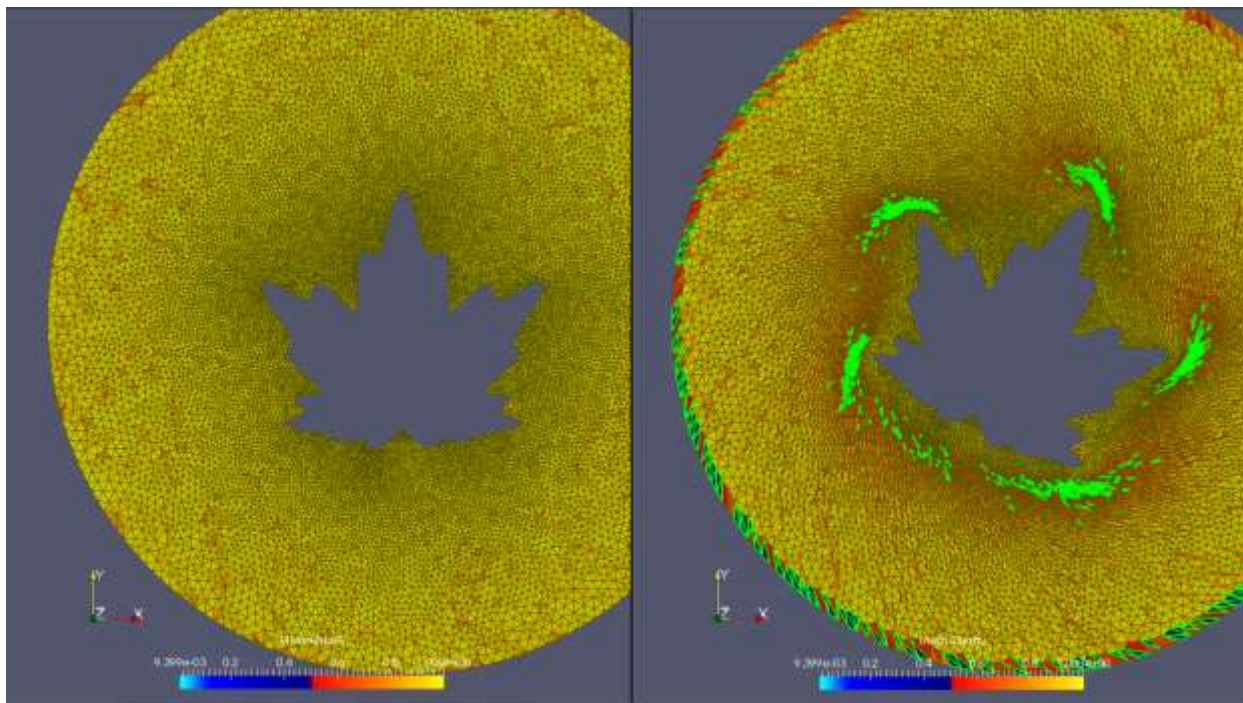
Σχήμα 5.21: Εφαρμογή της μεθόδου της μετακίνησης του πλέγματος με δίκτυο RBF με γωνία στροφής 20 μοιρών του περιγράμματος του φύλλου, με χρήση της *Inverse Multiquadric Biharmonics* και τα αποτελέσματα στην μετρική ποιότητας. **Πάνω αριστερά:** Το αρχικό πλέγμα. Η ποιότητα του κάθε τριγωνικού στοιχείου εμφανίζεται με διαφορετικό χρώμα. **Πάνω δεξιά:** Το αναμορφωμένο πλέγμα για γωνία στροφής κατά 20 μοίρες και τα αποτελέσματα στην μετρική ποιότητας του πλέγματος. **Κάτω αριστερά:** Το αρχικό πλέγμα. Με πράσινο χρώμα εμφανίζονται τα κέντρα του δικτύου (όλοι εκείνοι οι κόμβοι ή υποσύνολο αυτών, των οποίων η τιμή της μετατόπισης είναι γνωστή). **Κάτω δεξιά:** Το αναμορφωμένο πλέγμα. . Μπορούμε να παρατηρήσουμε την ολική επίδραση της *Inverse Multiquadric Biharmonics* αφού βλέπουμε ότι επηρεάζει έντονα τους εξωτερικούς οριακούς κόμβους.



Σχήμα 5.22: **Αριστερά:** Το αρχικό πλέγμα. **Δεξιά:** Το αναμορφωμένο πλέγμα για γωνία 20 μοιρών και η ανεπιθύμητη παραμόρφωση του πλέγματος. Με πράσινο χρώμα εμφανίζονται τα τριγωνικά στοιχεία με τιμή μετρικής ποιότητας χαμηλότερη του 0.5. Μπορούμε να παρατηρήσουμε την ολική επίδραση της *Inverse Multiquadric Biharmonics* αφού βλέπουμε ότι επηρεάζει έντονα τους εξωτερικούς οριακούς κόμβους.



Σχήμα 5.23: Εφαρμογή της μεθόδου της μετακίνησης του πλέγματος με δίκτυο RBF με γωνία στροφής 30 μοιρών του περιγράμματος του φύλλου, με χρήση της *Inverse Multiquadric Biharmonics* και τα αποτελέσματα στην μετρική ποιότητας. **Πάνω αριστερά:** Το αρχικό πλέγμα. Η ποιότητα του κάθε τριγωνικού στοιχείου εμφανίζεται με διαφορετικό χρώμα. **Πάνω δεξιά:** Το αναμορφωμένο πλέγμα για γωνία στροφής κατά 30 μοίρες και τα αποτελέσματα στην μετρική ποιότητας του πλέγματος. **Κάτω αριστερά:** Το αρχικό πλέγμα. Με πράσινο χρώμα εμφανίζονται τα κέντρα του δικτύου (όλοι εκείνοι οι κόμβοι ή υποσύνολο αυτών, των οποίων η τιμή της μετατόπισης είναι γνωστή). **Κάτω δεξιά:** Το αναμορφωμένο για γωνία 30 μοιρών. . Μπορούμε να παρατηρήσουμε την ολική επίδραση της *Inverse Multiquadric Biharmonics* αφού βλέπουμε ότι επηρεάζει έντονα τους εξωτερικούς οριακούς κόμβους.



Σχήμα 5.24: **Αριστερά:** Το αρχικό πλέγμα. **Δεξιά:** Το αναμορφωμένο πλέγμα για γωνία 30 μοιρών και η ανεπιθύμητη παραμόρφωση του πλέγματος. Με πράσινο χρώμα εμφανίζονται τα τριγωνικά στοιχεία με τιμή μετρικής ποιότητας χαμηλότερη του 0.5. Μπορούμε να παρατηρήσουμε την ολική επίδραση της *Inverse Multiquadric Biharmonics* αφού βλέπουμε ότι επηρεάζει έντονα τους εξωτερικούς οριακούς κόμβους. Όμως δεν αποφεύγεται η ανεπιθύμητη παραμόρφωση του πλέγματος κοντά στην γεωμετρία.

5.3 Σχόλια

Όσον αφορά τη συνάρτηση ενεργοποίησης *Wendland C²*, παρατηρείται ότι όσο η ακτίνα επιρροής αυξάνεται, τόσο πιο μεγάλη γωνία στροφής αντέχει η εσωτερική γεωμετρία. Αξίζει να σημειωθεί ότι, η μέση τιμή της μετρικής ποιότητας του πλέγματος, όπως εκφράστηκε με τη μετρική αυτή της μέσης αναλογίας, δεν καθορίζει το πότε το πλέγμα κρίνεται ακατάλληλο, αφού όσο αυξάνεται η ακτίνα επιρροής η ποιότητα είναι χειρότερη από την ποιότητα της κατάστασης στην οποία οριακά το πλέγμα γίνεται ακατάλληλο με άλλες ακτίνες R . Η Ιακωβιανή ορίζουσα του κάθε τριγωνικού στοιχείου είναι αυτή που καθορίζει το πότε είναι ακατάλληλο το πλέγμα. Όταν η ορίζουσα αυτή ενός, τουλάχιστον, τριγωνικού στοιχείου του πλέγματος αποκτήσει αρνητική τιμή, δηλαδή η αντιστροφή της φοράς, με την οποία έχουν οριστεί οι κόμβοι, αλλάζει λόγω του ανεστραμμένου τριγώνου, το οποίο καθιστά το πλέγμα ακατάλληλο. Το ανεστραμμένο στοιχείο έχει μετακινηθεί ανεπιθύμητα πάνω από κάποιο άλλο στοιχείο.

Επίσης, από όλα τα παραπάνω σχήματα του κεφαλαίου, παρατηρείται ότι σε όλες τις περιπτώσεις (και σε αυτές που χρησιμοποιήθηκε η *Wendland C²*, η *Gauss* και σε αυτές που χρησιμοποιήθηκε η συνάρτηση *Inverse Multiquadric Biharmonics* ως συνάρτηση ενεργοποίησης του δικτύου RBF), η μέση τιμή μειώνεται, αλλά η τυπική απόκλιση αυξάνεται καθώς αυξάνεται η γωνία στροφής του εσωτερικού τετραγώνου, γεγονός που δείχνει ότι υπάρχει αυξημένη ποικιλία στην μετρική ποιότητας των τριγωνικών στοιχείων.

Επιπλέον, στον πίνακα 5.6 φαίνεται, για κάθε ακτίνα επιρροής της συνάρτησης *Wendland C²*, η κρίσιμη γωνία, φ_{crit} , μετά την οποία παρουσιάστηκε τουλάχιστον ένα ανεστραμμένο τριγωνικό στοιχείο. Για τις γωνίες αυτές, παρατίθενται η μέση τιμή (\bar{q}_e), η τυπική απόκλιση (σ_{q_e}) και η ελάχιστη τιμή ($q_{e,min}$) της μετρικής ποιότητας του πλέγματος. Τα ίδια μεγέθη φαίνονται στον

πίνακα 5.7 και 5.8 αντίστοιχα για την συνάρτηση *Gauss* και για την Inverse Multiquadric Biharmonics του δικτύου RBF. Παρατηρείται ότι με τη συνάρτηση *Gauss* το πλέγμα κρίθηκε ακατάλληλο μετά τις 14 μοίρες στροφής της εσωτερικής γεωμετρίας, ενώ με την *Wendland C²* το πλέγμα κρίθηκε ακατάλληλο στην ίδια γωνία περίπου για ακτίνα επιρροής ανάμεσα σε R=2 και R=3. Η διαφορά είναι ότι, στην τελευταία συνάρτηση, το πλέγμα είχε λίγο καλύτερη μέση τιμή της μετρικής ποιότητας του πλέγματος. Τέλος, αξίζει να αναφερθεί ότι τόσο μεγάλες γωνίες στροφής, λ.χ. 60 μοιρών, δεν θα χρειαστεί σχεδόν ποτέ να εφαρμοστούν σε έναν αλγόριθμο βελτιστοποίησης. Αν όμως η μέθοδος αυτή εφαρμοστεί λ.χ. σε μια παλλόμενη αεροτομή, έχει νόημα, γιατί σε αυτήν την περίπτωση κανείς δεν ξέρει τι μπορεί να αντιμετωπίσει. Στον πίνακα 5.8 παρατηρούμε επίσης ότι για τη συνάρτηση ολικής επιρροής Inverse Multiquadric Biharmonics το πλέγμα κρίνεται ακατάλληλο μετά τις 30 μοίρες και θα μπορούσε να αυξηθεί αν στη συγκεκριμένη εφαρμογή τα εξωτερικά όρια του πλέγματος μπορούσαν να μετακινηθούν ή ήταν σε μεγαλύτερη απόσταση από τα κέντρα του δικτύου. Αυτό συμβαίνει γιατί όπως παρουσιάστηκε σε προηγούμενο κεφάλαιο η συγκεκριμένη συνάρτηση όπως και άλλες έχουν σημαντική επιρροή στις μεγάλες αποστάσεις από το σημείο εφαρμογής.

Ακτίνα επιρροής R	2	3	9
φ_{crit}	10	18	60
\bar{q}_e	0.908281	0.864021	0.632358
σ_{qe}	0.111562	0.153823	0.265349
$q_{e,min}$	0.221473	0.0654887	0.004881

Πίνακας 5.6: Αποτελέσματα μέσης τιμής ποιότητας, τυπικής απόκλισης και ελάχιστης τιμής μετρικής ποιότητας πλέγματος για εφαρμογή της μεθόδου της μετακίνησης πλέγματος με δίκτυα RBF και συνάρτηση ενεργοποίησης, τη *Wendland C²*. Παρουσιάζεται, για κάθε ακτίνα επιρροής, η κρίσιμη γωνία στροφής της γεωμετρίας του φύλλου πλατάνου, μετά την οποία το πλέγμα κρίθηκε ακατάλληλο.

φ_{crit}	\bar{q}_e	σ_{qe}	$q_{e,min}$
14	0.882724	0.152845	0.0516738

Πίνακας 5.7: Αποτελέσματα μέσης τιμής ποιότητας, τυπικής απόκλισης και ελάχιστης τιμής μετρικής ποιότητας πλέγματος για εφαρμογή της μεθόδου της μετακίνησης πλέγματος με δίκτυα RBF και συνάρτηση ενεργοποίησης τη *Gauss*. Παρουσιάζεται η κρίσιμη γωνία στροφής της γεωμετρίας του φύλλου πλατάνου, μετά την οποία το πλέγμα κρίθηκε ακατάλληλο.

φ_{crit}	\bar{q}_e	σ_{qe}	$q_{e,min}$
30	0.840448	0.145307	0.00939905

Πίνακας 5.8: Αποτελέσματα μέσης τιμής ποιότητας, τυπικής απόκλισης και ελάχιστης τιμής μετρικής ποιότητας πλέγματος για εφαρμογή της μεθόδου της μετακίνησης πλέγματος με δίκτυα RBF και συνάρτηση ενεργοποίησης τη *Inverse Multiquadric Biharmonics*. Παρουσιάζεται η κρίσιμη γωνία στροφής της γεωμετρίας του φύλλου πλατάνου, μετά την το πλέγμα κρίθηκε ακατάλληλο.

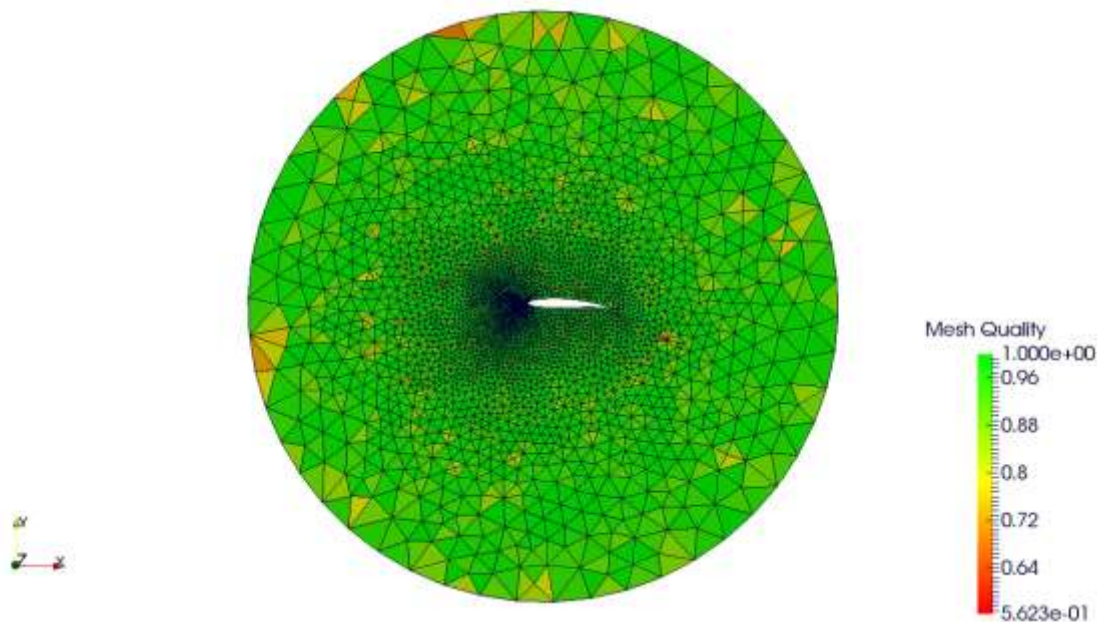
6 Εφαρμογές σε πλέγμα γύρω από αεροτομή

6.1 Περιγραφή του προβλήματος

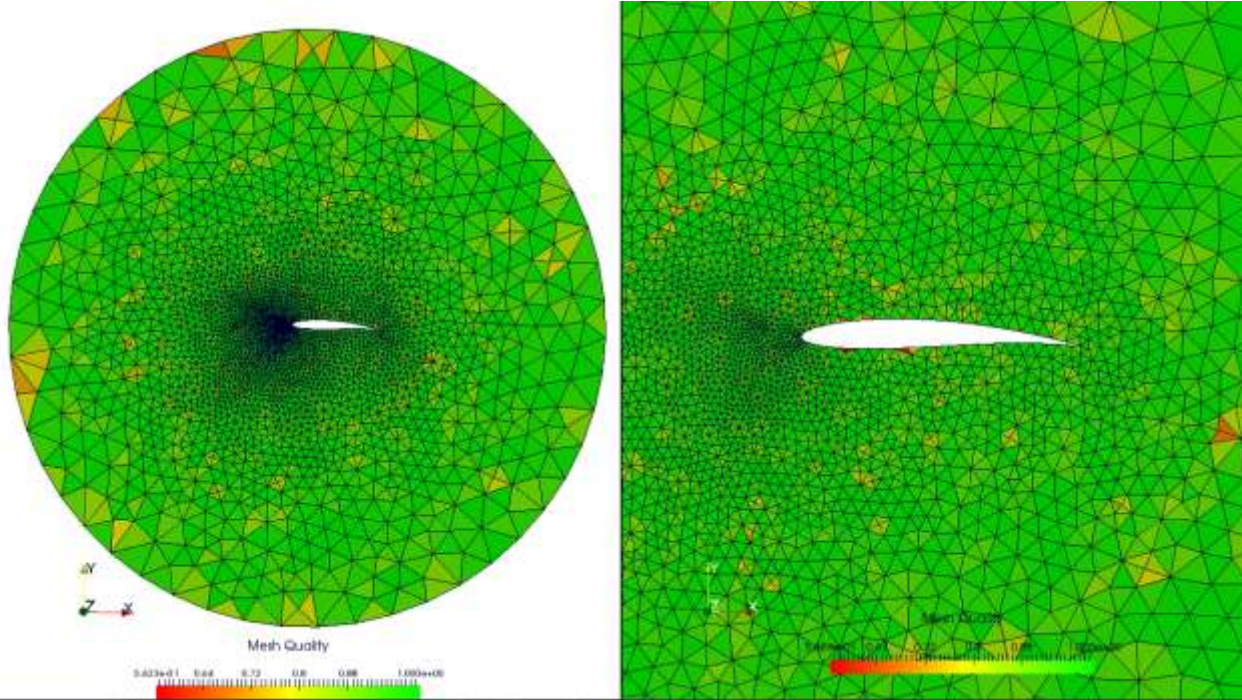
Σκοπός είναι η εφαρμογή της μεθόδου και η διερεύνησή της για το πώς η στροφή της εσωτερικής γεωμετρίας, (το περίγραμμα της αεροτομής τοποθετημένο μέσα σε ένα κύκλο και με τον ενδιάμεσο χώρο να έχει πλεγματοποιηθεί με τριγωνικά στοιχεία με τη μέθοδο του προελαύνοντος μετώπου “advancing front method”) επηρεάζει την ποιότητα του πλέγματος. Για αυτό το λόγο δημιουργήθηκε μία αεροτομή με μήκος χορδής ίσο με τη 0.61, όπως φαίνεται στο σχήμα 6.1 και 6.2, γύρω από αυτή κύκλος με ακτίνα $\rho=2$. Το περίγραμμα της αεροτομής αποτελείται από 50 κόμβους, ενώ το περίγραμμα του κύκλου(εξωτερική γεωμετρία), αποτελείται από 100 κόμβους. Το αρχικό πλέγμα αποτελείται από 2961 εσωτερικούς κόμβους με 6022 τριγωνικά στοιχεία.

Στη συνέχεια, δημιουργήθηκε μη-δομημένο πλέγμα με τριγωνικά στοιχεία, με τη βοήθεια λογισμικού της ΜΠΥΡ&Β/ΕΜΠ, γύρω από την αεροτομή. Στο κεφάλαιο αυτό πραγματοποιούνται διαφορετικού μέτρου στροφές της εσωτερικής γεωμετρίας δεξιόστροφα, προσαρμόζεται το αρχικό πλέγμα στο αλλαγμένο περίγραμμα μέσω του δικτύου RBF και διερευνάται η ποιότητα του πλέγματος. Πρωτίστως ενδιαφέρον προκαλεί το πότε δημιουργούνται ανεπιθύμητα ανεστραμμένα (inverted) τρίγωνα. Ως συνάρτηση ενεργοποίησης του δικτύου RBF επιλέχθηκε συνάρτηση τοπικής επιρροής. Η συνάρτηση τοπικής επιρροής που χρησιμοποιήθηκε είναι η Wendland $CP C^2$ με τιμή που δίνεται από τη σχέση 5.1.

Για τη συνάρτηση ενεργοποίησης, διερευνάται το πώς η στροφή της εσωτερικής γεωμετρίας επηρεάζει τόσο την ποιότητα του πλέγματος, που προσαρμόζεται με τη μέθοδο της μετακίνησης πλέγματος με δίκτυα RBF, όσο και τη δημιουργία ανεπιθύμητων ανεστραμμένων τριγώνων. Για τη συνάρτηση τοπικής επιρροής, το πρόγραμμα εκτελείται για ακτίνα $R=2$. Στη συνέχεια, από το σχήμα 6.3 ως 6.5.παρουσιάζονται τα αποτελέσματα για κάθε εφαρμογή.



Σχήμα 6.1: Το περίγραμμα της αεροτομής(εσωτερική γεωμετρία) αποτελείται από 50 κόμβους, ενώ το περίγραμμα του κύκλου(εξωτερική γεωμετρία), αποτελείται από 100 κόμβους. Το αρχικό πλέγμα αποτελείται από 2961 εσωτερικούς κόμβους με 6022 τριγωνικά στοιχεία.



Σχήμα 6.2: Το αρχικό πλέγμα. **Δεξιά:** έχει γίνει εστίαση στην περιοχή του περιγράμματος της αεροτομής.

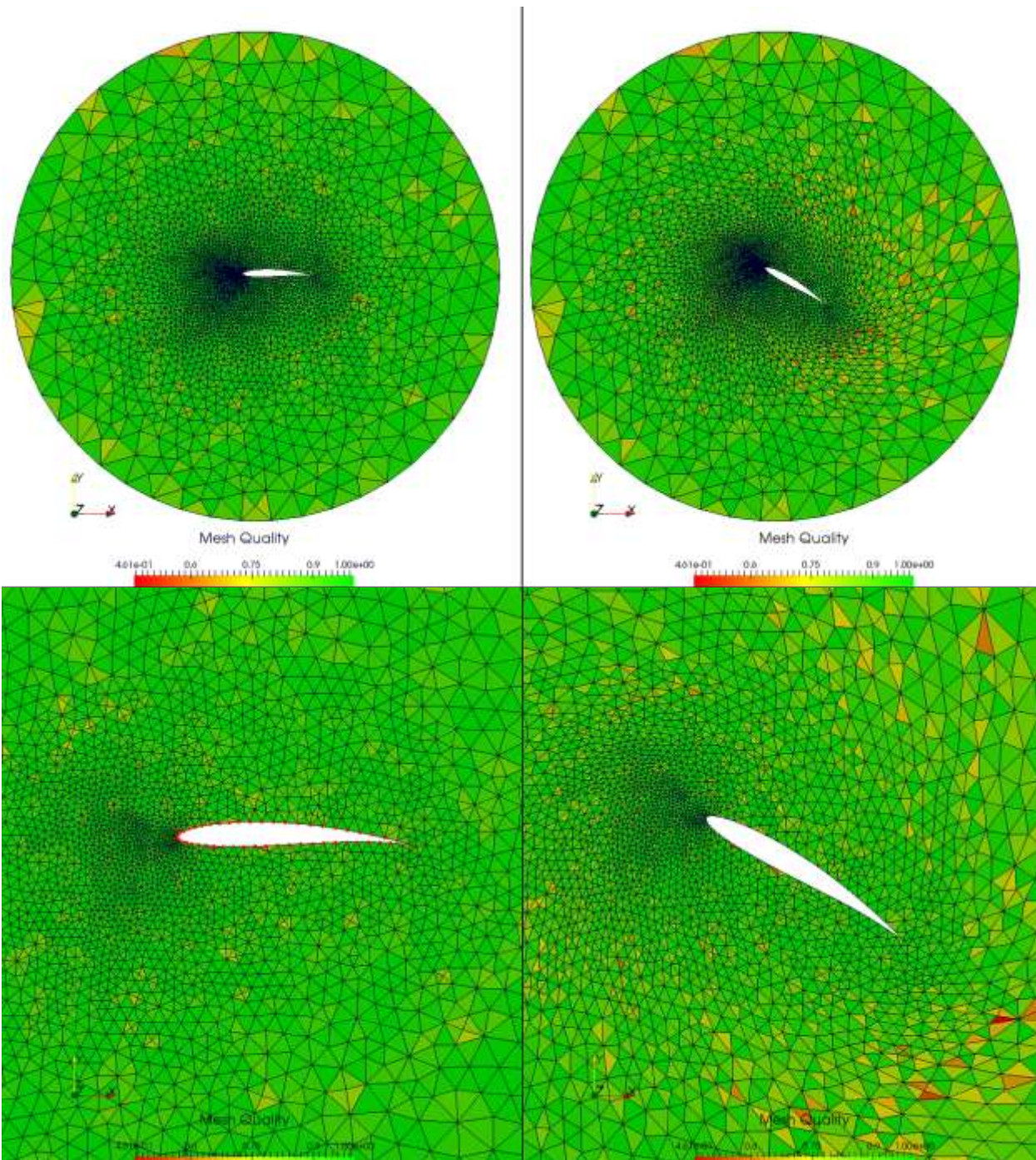
6.2 Παρουσίαση αποτελεσμάτων

6.2.1 Αποτελέσματα με τη $Wendland C^2$, $R = 2$

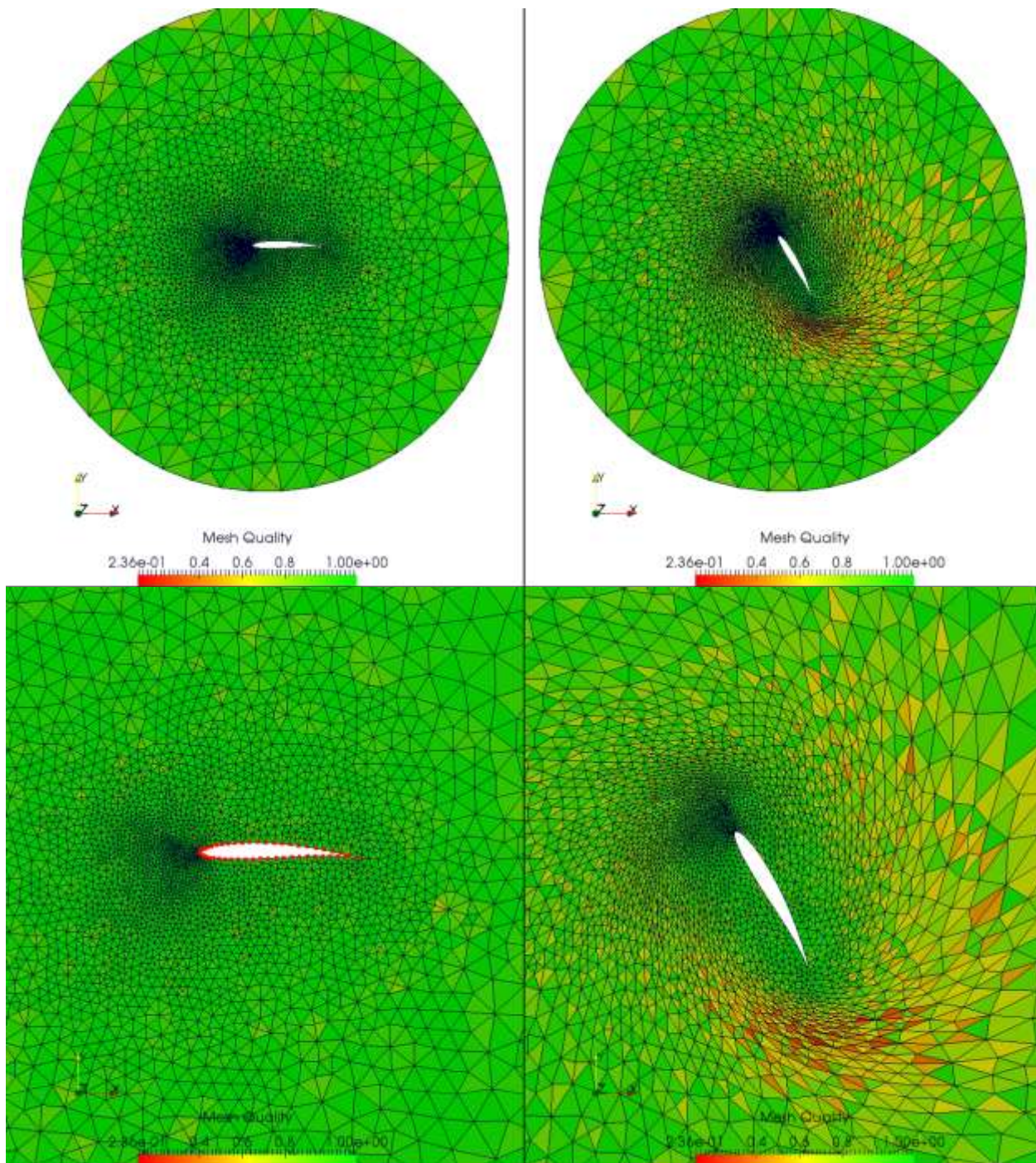
Η ακτίνα επιρροής που επιλέχθηκε για τη συνάρτηση ενεργοποίησης $Wendland C^2$ είναι $R=2$. Έπειτα έγινε στροφή της εσωτερικής γεωμετρίας για γωνία στροφής, θ . Στον πίνακα 6.1 φαίνονται η μέση τιμή (\bar{q}_e), η τυπική απόκλιση (σ_{qe}) και η ελάχιστη τιμή ($q_{e,min}$) της μετρικής ποιότητας του για τη γωνία στροφής. Στα σχήματα 6.3 ως 6.5 παρουσιάζεται η αεροτομή σε τυχαία επιλεγμένη γωνία στροφής μέχρι την οποία δεν έχει εμφανίσει ανεστραμμένα τριγωνικά στοιχεία αλλά η ελάχιστη τιμή της μετρικής ποιότητας έχει αρκετά χαμηλή τιμή. Η στροφή της εσωτερικής γεωμετρίας σταμάτησε στις 83 μοίρες, γιατί το πλέγμα εμφάνισε πολύ χαμηλή ελάχιστη τιμή μετρικής ποιότητας πλέγματος και αμέσως μετά εμφανίζει τα πρώτα ανεστραμμένα τριγωνικά στοιχεία.

θ (μοίρες)	0	30	60	83
\bar{q}_e	0.957215	0.931122	0.855778	0.766303
σ_{qe}	0.0428867	0.0624743	0.115619	0.180968
$q_{e,min}$	0.562287	0.460745	0.235805	0.0199818

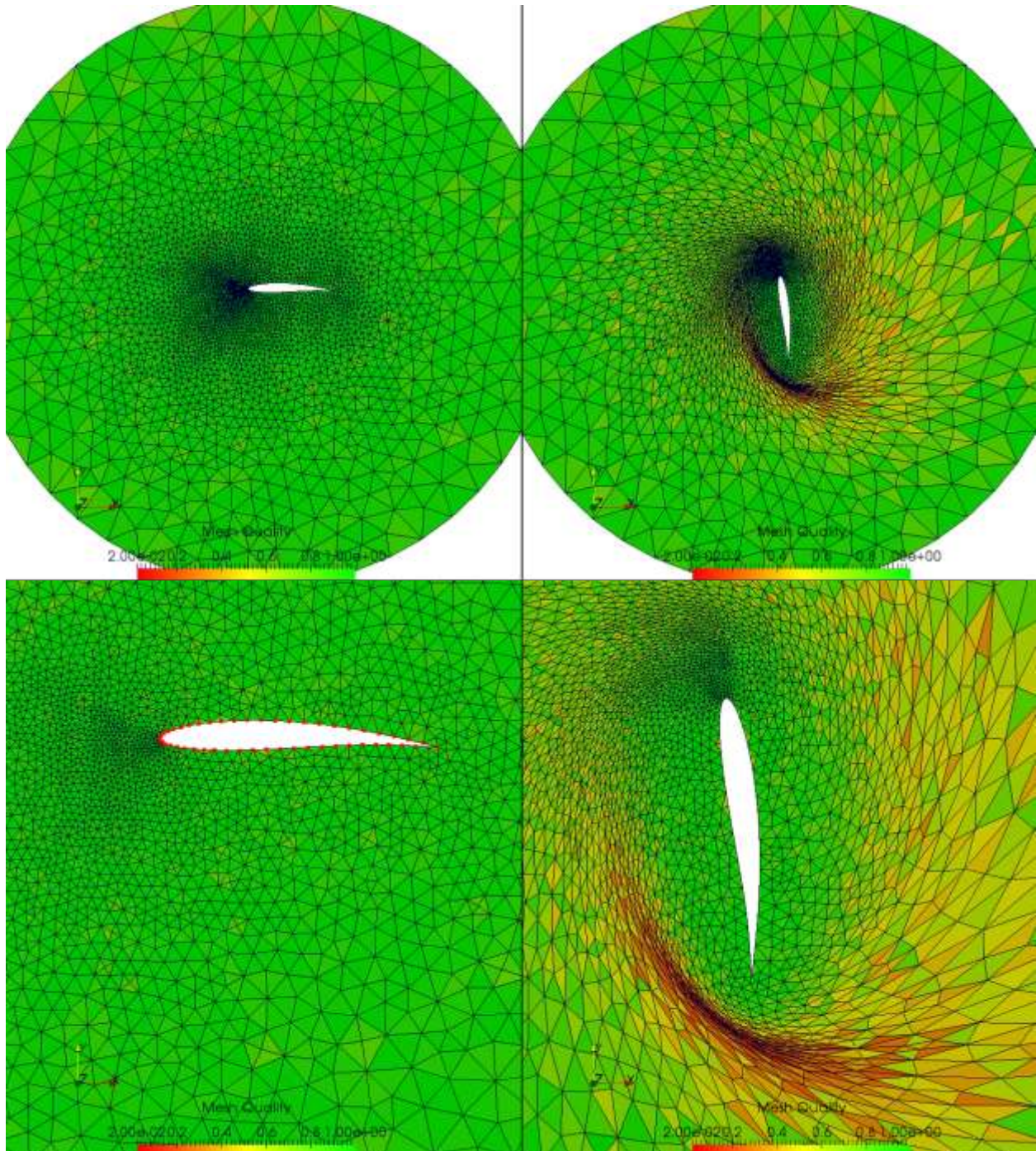
Πίνακας 6.1: Αποτελέσματα μέσης τιμής \bar{q}_e , τυπικής απόκλισης σ_{qe} και ελάχιστη τιμή $q_{e,min}$ της μετρικής ποιότητας του πλέγματος για την αρχική κατάσταση του πλέγματος και για γωνία στροφής της εσωτερικής γεωμετρίας. Ως συνάρτηση ενεργοποίησης είναι η $Wendland C^2$ με ακτίνα $R=2$.



Σχήμα 6.3: Εφαρμογή της μεθόδου της μετακίνησης του πλέγματος με δίκτυο RBF με γωνία στροφής 30 μοιρών του περιγράμματος της αεροτομής, με χρήση της Wendland C^2 με $R=2$ και τα αποτελέσματα στην μετρική ποιότητας. **Πάνω αριστερά:** Το αρχικό πλέγμα. Η ποιότητα του κάθε τριγωνικού στοιχείου εμφανίζεται με διαφορετικό χρώμα. **Πάνω δεξιά:** Το αναμορφωμένο πλέγμα για γωνία στροφής κατά 30 μοίρες και τα αποτελέσματα στην μετρική ποιότητας του πλέγματος. **Κάτω αριστερά:** Το αρχικό πλέγμα. Με κόκκινο χρώμα εμφανίζονται τα κέντρα του δικτύου (όλοι εκείνοι οι κόμβοι ή υποσύνολο αυτών, των οποίων η τιμή της μετατόπισης είναι γνωστή). **Κάτω δεξιά:** Το αναμορφωμένο πλέγμα και η παραμόρφωση του πλέγματος.



Σχήμα 6.4: Εφαρμογή της μεθόδου της μετακίνησης του πλέγματος με δίκτυο RBF με γωνία στροφής 60 μοιρών του περιγράμματος της αεροτομής, με χρήση της Wendland C^2 με $R=2$ και τα αποτελέσματα στην μετρική ποιότητας. **Πάνω αριστερά:** Το αρχικό πλέγμα. Η ποιότητα του κάθε τριγωνικού στοιχείου εμφανίζεται με διαφορετικό χρώμα. **Πάνω δεξιά:** Το αναμορφωμένο πλέγμα για γωνία στροφής κατά 60 μοίρες και τα αποτελέσματα στην μετρική ποιότητας του πλέγματος. **Κάτω αριστερά:** Το αρχικό πλέγμα. Με κόκκινο χρώμα εμφανίζονται τα κέντρα του δικτύου (όλοι εκείνοι οι κόμβοι ή υποσύνολο αυτών, των οποίων η τιμή της μετατόπισης είναι γνωστή). **Κάτω δεξιά:** Το αναμορφωμένο πλέγμα και η ανεπιθύμητη παραμόρφωση του πλέγματος.



Σχήμα 6.5: Εφαρμογή της μεθόδου της μετακίνησης του πλέγματος με δίκτυο RBF με γωνία στροφής 83 μοιρών του περιγράμματος της αεροτομής, με χρήση της Wendland C^2 με $R=2$ και τα αποτελέσματα στην μετρική ποιότητας. **Πάνω αριστερά:** Το αρχικό πλέγμα. Η ποιότητα του κάθε τριγωνικού στοιχείου εμφανίζεται με διαφορετικό χρώμα. **Πάνω δεξιά:** Το αναμορφωμένο πλέγμα για γωνία στροφής κατά 83 μοίρες και τα αποτελέσματα στην μετρική ποιότητας του πλέγματος. **Κάτω αριστερά:** Το αρχικό πλέγμα. Με κόκκινο χρώμα εμφανίζονται τα κέντρα του δικτύου (όλοι εκείνοι οι κόμβοι ή υποσύνολο αυτών, των οποίων η τιμή της μετατόπισης είναι γνωστή). **Κάτω δεξιά:** Το αναμορφωμένο πλέγμα και η ανεπιθύμητη παραμόρφωση του πλέγματος. Σε αυτήν την περίπτωση, για γωνία μεγαλύτερη των 83 μοιρών, το πλέγμα θα έχει εμφανίσει έναν πολύ μικρό αριθμό ανεστραμμένων τριγώνων, θα έχει δηλαδή κριθεί ακατάλληλο για χρήση με λογισμικό επίλυσης ΜΔΕ.

6.3 Σχόλια

Είναι σημαντικό να σημειωθεί ότι ο συγκεκριμένος κώδικας δεν έχει προγραμματιστεί με σκοπό να χρησιμοποιηθεί σε ακραίες εφαρμογές όπως αυτή του σχήματος 6.5. Επίσης, η μέση τιμή της μετρικής ποιότητας του πλέγματος, όπως εκφράστηκε με τη μετρική αυτή της μέσης αναλογίας, δεν καθορίζει το πότε το πλέγμα κρίνεται ακατάλληλο. Η Ιακωβιανή ορίζουσα του κάθε τριγωνικού στοιχείου είναι αυτή που καθορίζει το πότε είναι ακατάλληλο το πλέγμα. Όταν η ορίζουσα αυτή ενός, τουλάχιστον, τριγωνικού στοιχείου του πλέγματος αποκτήσει αρνητική τιμή, τότε η αντιστροφολογική φορά, με την οποία έχουν οριστεί οι κόμβοι αλλάζει λόγω του ανεστραμμένου τριγώνου, το οποίο καθιστά το πλέγμα ακατάλληλο. Το ανεστραμμένο στοιχείο έχει μετακινηθεί ανεπιθύμητα πάνω σε κάποιο άλλο στοιχείο.

Επιπλέον, στον πίνακα 6.1 φαίνεται, για τη συνάρτηση *Wendland* C^2 με ακτίνα επιρροής $R=2$, η κρίσιμη γωνία, φ_{crit} , μετά την οποία παρουσιάστηκε τουλάχιστον ένα ανεστραμμένο τριγωνικό στοιχείο είναι η $\varphi_{crit} = 83$.

7 Εφαρμογές σε 3D πλέγμα γύρω από αεροσκάφος

7.1 Περιγραφή του προβλήματος

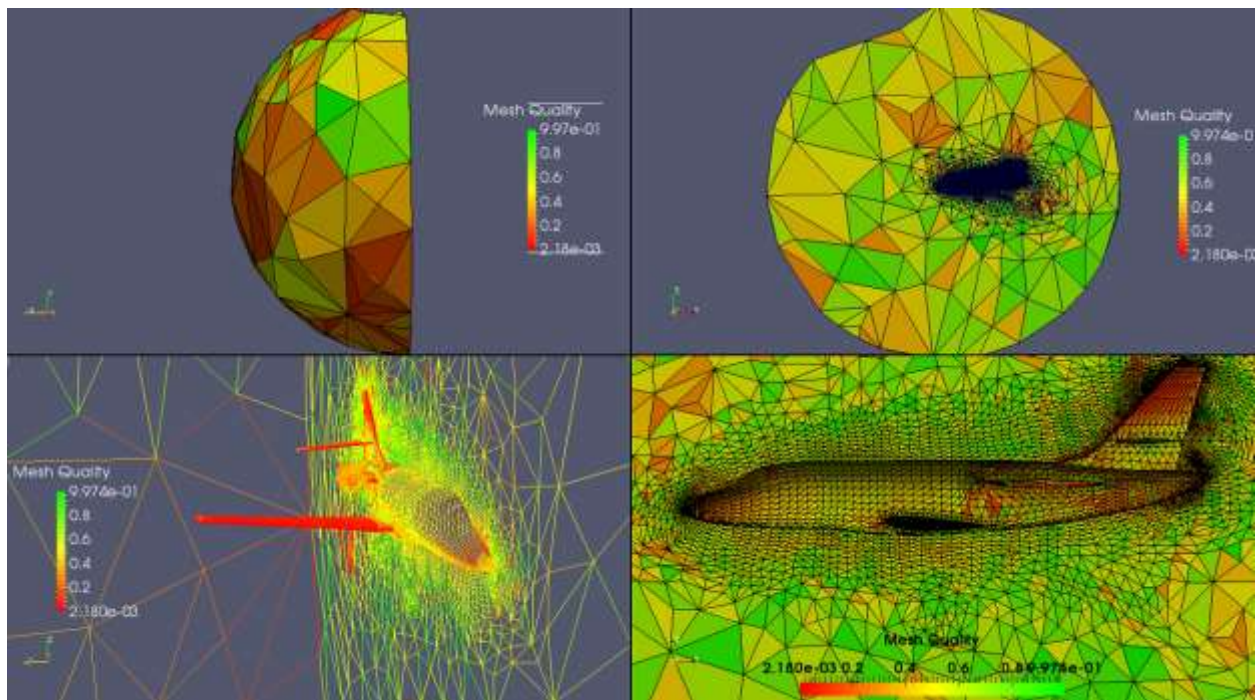
Το θέμα αυτής της ενότητας αφορά την εφαρμογή της μεθόδου αναμόρφωσης του πλέγματος με δίκτυα RBF. Κύριος σκοπός είναι η εφαρμογή του λογισμικού που αναπτύχθηκε και υλοποιεί τα δίκτυα συναρτήσεων RBF σε τριδιάστατο πλέγμα καθώς και η διερεύνηση για το πώς η στροφή μίας δύσκολης γεωμετρίας όπως το αεροσκάφος, επηρεάζει την ποιότητα του αρχικού πλέγματος, με χαμηλή αρχική τιμή μετρικής ποιότητας. Ο χώρος γύρω από το αεροσκάφος έχει πλεγματοποιηθεί με τετραεδρικά στοιχεία. Πρόκειται για ένα αεροσκάφος, το οποίο είναι συμμετρικό κατά τον άξονα y όπως φαίνεται στο σχήμα 7.1, έχει μήκος ατράκτου $L=20$ και μήκος πτερυγίου $P=10$, το οποίο παρουσιάζεται στα σχήματα 7.1 ως 7.3. Τα ποιοτικά χαρακτηριστικά του αναμορφωμένου πλέγματος μετά την στροφή της γεωμετρίας παρουσιάζονται στον πίνακα 7.1. Δοκιμάζεται η μέθοδος προσαρμογής πλέγματος με δίκτυα RBF σε τριδιάστατο πλέγμα, εφαρμόζεται στροφή στο αεροσκάφος και εξετάζεται η ποιότητα του πλέγματος που προκύπτει.

Η μέθοδος που αναπτύχθηκε δοκιμάζεται με τη συνάρτηση ενεργοποίησης τοπικής επιρροής. Ως συνάρτηση τοπικής επιρροής επιλέχθηκε η *Wendland C^2* με τιμή που δίνεται από τη σχέση 5.1. Υπενθυμίζεται ότι $\xi = \frac{r}{R}$, όπου $r = \|\vec{x} - \vec{x}_{c_j}\|$ και \vec{x}, \vec{x}_{c_j} τα διανύσματα θέσης δύο σημείων στο χώρο. Συγκεκριμένα, για τη συνάρτηση τοπικής επιρροής ορίστηκε μία ακτίνα R , τέτοια ώστε κάθε κέντρο \vec{x}_{c_j} του δικτύου να επηρεάζει εκείνους τους κόμβους του χωρικού πλέγματος, εξαιρώντας τους οριακούς κόμβους, οι οποίοι βρίσκονται μέσα σε κύκλο ακτίνας R και κέντρο το εκάστοτε \vec{x}_{c_j} κέντρο του δικτύου RBF. Επομένως, διερευνάται το πώς η στροφή της εσωτερικής γεωμετρίας επηρεάζει τόσο την ποιότητα του πλέγματος που προσαρμόζεται με τη μέθοδο RBF, όσο και τη δημιουργία ανεπιθύμητων ανεστραμμένων τριγώνων. Για τη συνάρτηση τοπικής επιρροής, το πρόγραμμα αυτό εκτελείται για ακτίνες επιρροής $R=21$ και $R=40$. Στη συνέχεια, παρουσιάζονται σχόλια και συμπεράσματα για τα αποτελέσματα που προκύπτουν.

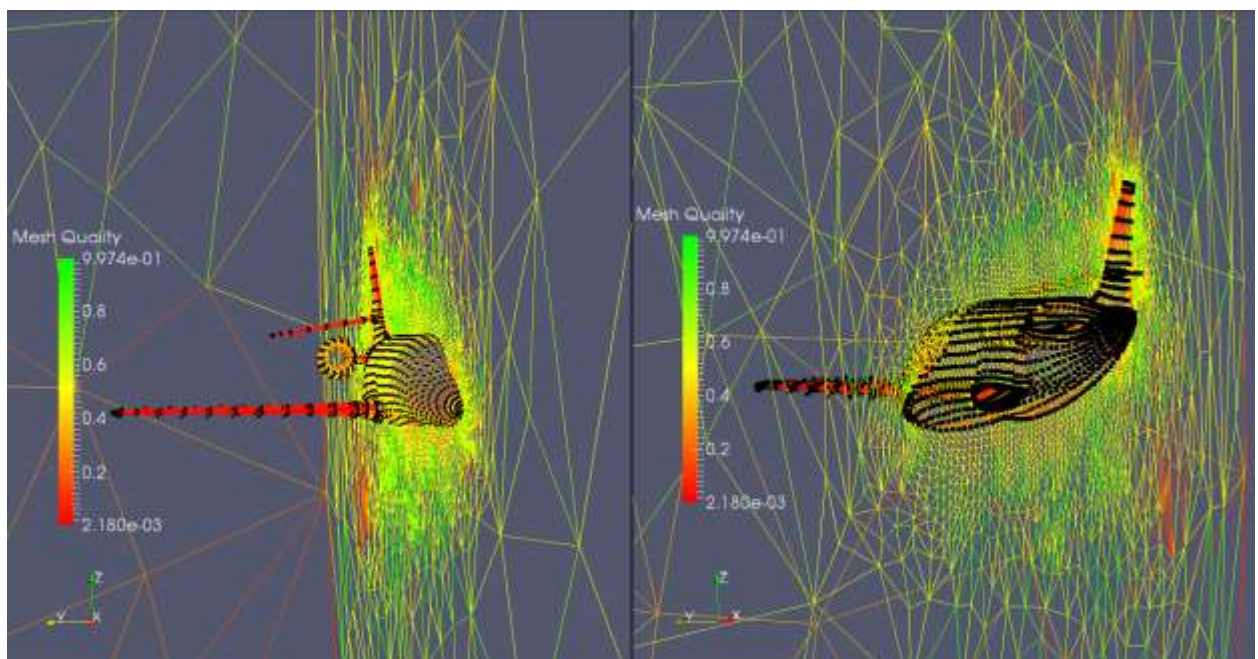
Το μη δομημένο πλέγμα γύρω από το αεροσκάφος του σχήματος 7.1 ως 7.9 αποτελείται από 45387 κόμβους. 2992 οριακούς κόμβους οι οποίοι αποτελούν το περίγραμμα του αεροσκάφους και χρησιμοποιούνται από το δίκτυο RBF ως κέντρα, 42248 εσωτερικούς κόμβους και 147 εξωτερικούς οριακούς. Ο χώρος γύρω από το αεροσκάφος έχει πλεγματοποιηθεί με τετραεδρικά στοιχεία. Το πλέγμα αποτελείται συνολικά από 255944 τετραεδρικά στοιχεία. Στον πίνακα 7.1 παρουσιάζεται η τιμή του λόγου της μέσης τιμής μετρικής ποιότητας του αναμορφωμένου πλέγματος \bar{q}_{eRBF} προς την μέση τιμή μετρικής ποιότητας του αντίστοιχου αρχικού πλέγματος \bar{q}_{eInit} .

Ακτίνα επιρροής R	21	40
$\frac{\varphi_{crit}}{\bar{q}_{eRBF}}$	5	15
$\frac{\bar{q}_{eRBF}}{\bar{q}_{eInit}}$	1.000	0.99

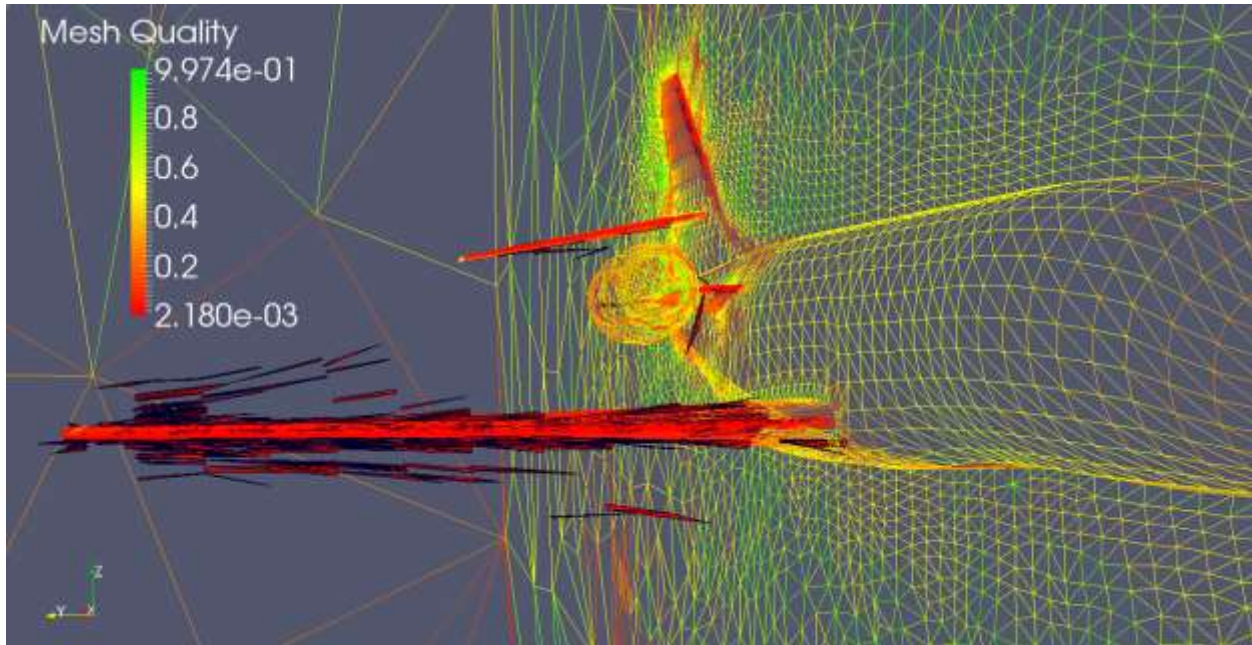
Πίνακας 7.1: Παρατίθενται η τιμή του λόγου της μέσης τιμής μετρικής ποιότητας του αναμορφωμένου πλέγματος \bar{q}_{eRBF} προς την μέση τιμή μετρικής ποιότητας του αντίστοιχου αρχικού πλέγματος \bar{q}_{eInit} .



Σχήμα 7.1: Το μη δομημένο πλέγμα συμμετρικό κατά τον άξονα y γύρω από το αεροσκάφος. **Κάτω αριστερά** :Το περίγραμμα του αεροσκάφους εσωτερικά του πλέγματος. **Κάτω δεξιά** :Το περίγραμμα του αεροσκάφους εξωτερικά του πλέγματος.



Σχήμα 7.2: Το περίγραμμα του αεροσκάφους του σχήματος αποτελείται από 2992 οριακούς κόμβους, οι οποίοι χρησιμοποιούνται από το δίκτυο RBF ως κέντρα. Με μαύρο χρώμα εμφανίζονται τα κέντρα του δικτύου (όλοι εκείνοι οι κόμβοι ή υποσύνολο αυτών, των οποίων η τιμή της μετατόπισης είναι γνωστή).



Σχήμα 7.3: Το περίγραμμα του αεροσκάφους. Με μαύρο χρώμα εμφανίζονται οι ακμές των τετραεδρικών στοιχείων με τιμή μετρικής ποιότητας πλέγματος χαμηλότερη του 0.03. Τα τετραεδρικά στοιχεία με τιμή χαμηλότερη του 0.03 είναι 2293.

7.2 Εφαρμογές με τη $Wendland C^2$

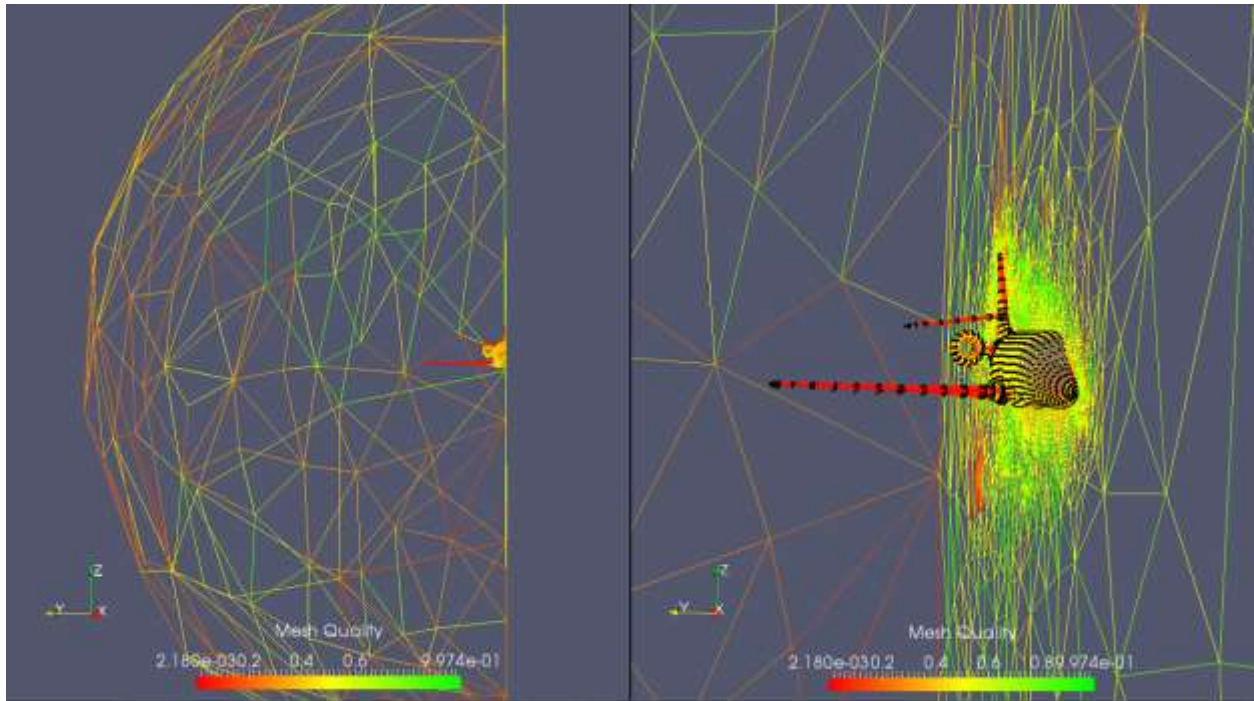
7.2.1 Αποτελέσματα με τη $Wendland C^2$ και $R=21$

Σε αυτήν την περίπτωση, μελετάται η συμπεριφορά της μεθόδου μετακίνησης πλέγματος σε αεροσκάφος. Η ακτίνα επιρροής που επιλέχθηκε για τη συνάρτηση ενεργοποίησης $Wendland C^2$ είναι $R=21$ έτσι περικλείει ολόκληρο το αεροσκάφος και έχει ως αποτέλεσμα την επιρροή κάθε κέντρου με κάθε άλλο κέντρο που έχει επιλεγθεί. Έπειτα έγινε στροφή του αεροσκάφους για γωνία, φ . Στον πίνακα 6.2 φαίνεται για τη κρίσιμη γωνία στροφής φ_{crit} πριν το πλέγμα κριθεί ακατάλληλο ο λόγος της μέσης τιμής μετρικής ποιότητας του αναμορφωμένου πλέγματος $\overline{q_{eRBF}}$ προς τη μέση τιμή μετρικής ποιότητας του αντίστοιχου αρχικού πλέγματος $\overline{q_{eInit}}$.

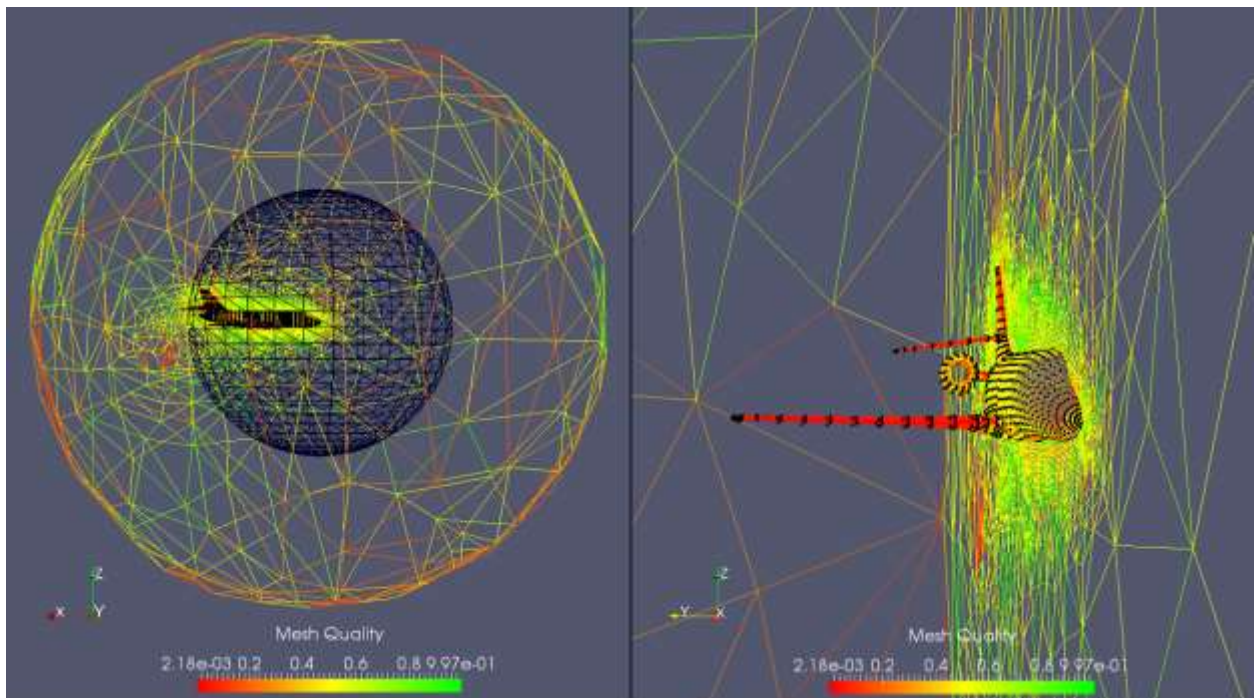
Ενώ στο σχήμα 7.6 φαίνεται το αεροσκάφος στην επιλεγμένη γωνία στροφής, καθώς και το πως διαμορφώθηκε το πλέγμα μετά την εφαρμογή της μεθόδου. Στο σχήμα αυτό έχει γίνει εστίαση στην περιοχή του αεροσκάφους για λόγους ευκρίνειας. Η στροφή της γεωμετρίας σταμάτησε στις 5 μοίρες, πριν το πλέγμα εμφάνισε ένα πολύ μικρό αριθμό ανεστραμμένων τετραεδρικών στοιχείων.

Ακτίνα επιρροής R	21
$\frac{\varphi_{crit}}{\overline{q_{eRBF}}}$	5
$\frac{\overline{q_{eRBF}}}{\overline{q_{eInit}}}$	1.000

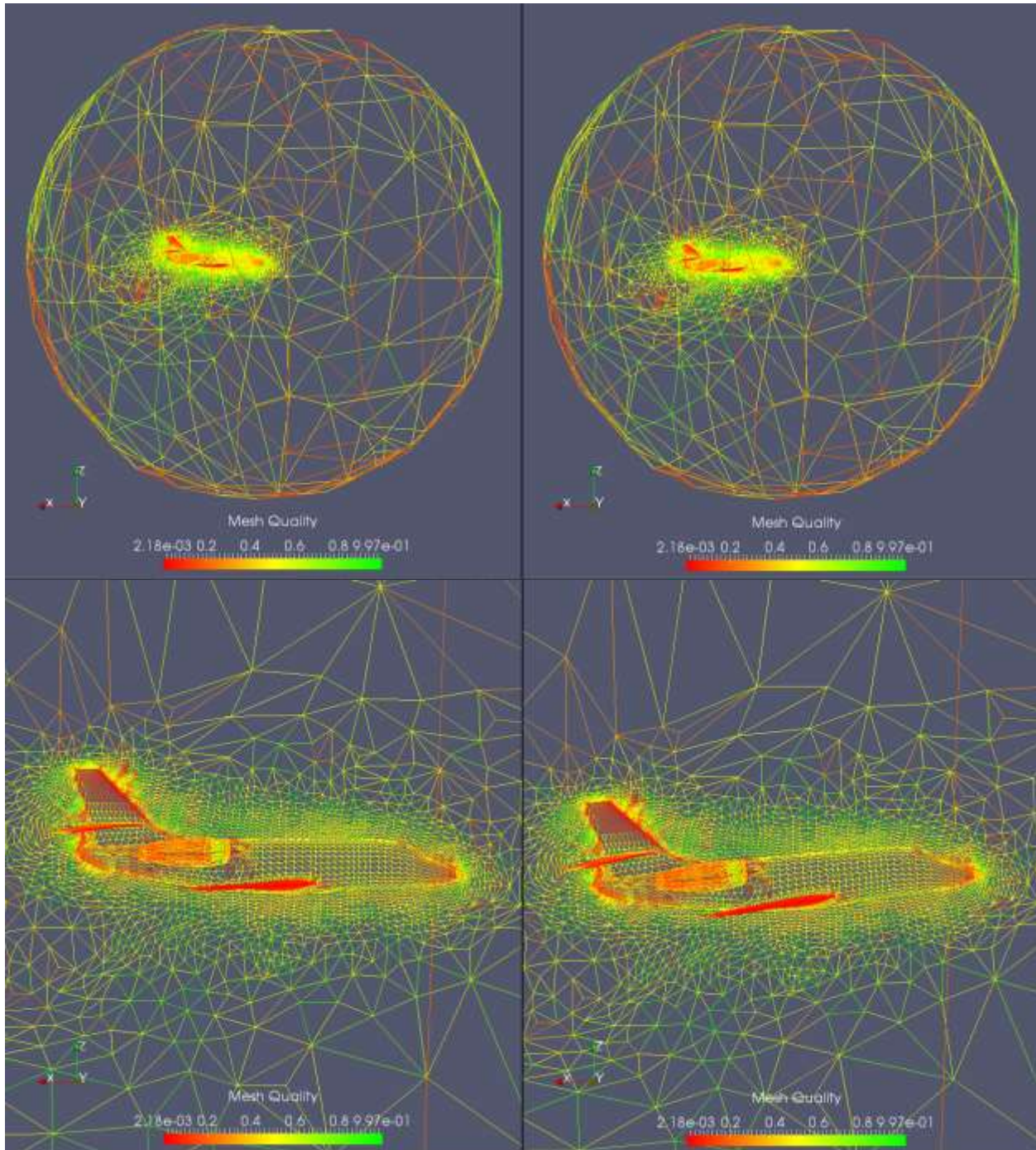
Πίνακας 7.2: Παρατίθενται η τιμή του λόγου της μέσης τιμής μετρικής ποιότητας του αναμορφωμένου πλέγματος $\overline{q_{eRBF}}$ προς την μέση τιμή μετρικής ποιότητας του αντίστοιχου αρχικού πλέγματος $\overline{q_{eInit}}$.



Σχήμα 7.4: Το περίγραμμα του αεροσκάφους αποτελείται από 2992 οριακούς κόμβους, οι οποίοι χρησιμοποιούνται από το δίκτυο RBF ως κέντρα. Με μαύρο χρώμα εμφανίζονται τα κέντρα του δικτύου (όλοι εκείνοι οι κόμβοι ή υποσύνολο αυτών, των οποίων η τιμή της μετατόπισης είναι γνωστή).



Σχήμα 7.5: Με μαύρο χρώμα εμφανίζονται τα κέντρα του δικτύου. Με μαύρο χρώμα εμφανίζεται η σφαίρα επιρροής (με ακτίνα επιρροής $R=21$) ενός εκ των κέντρων το οποίο είναι το κέντρο που βρίσκεται στην αρχή του αεροσκάφους.



Σχήμα 7.6: **Αριστερά:** Το αρχικό πλέγμα. **Δεξιά:** Το αναμορφωμένο πλέγμα για στροφή 5 μοιρών.

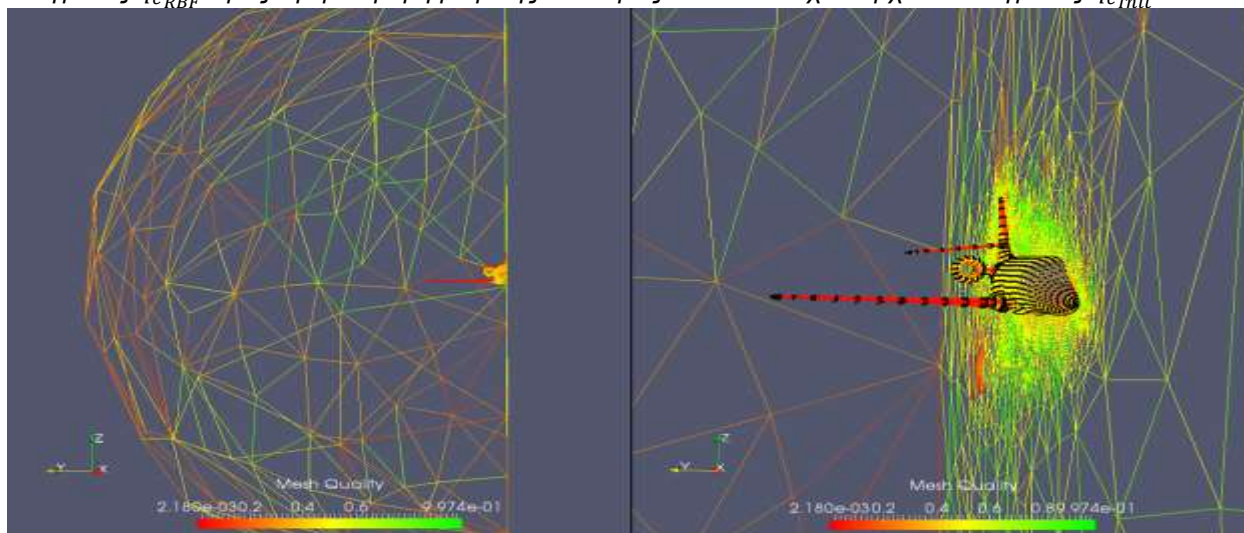
7.2.2 Αποτελέσματα με τη $Wendland C^2$ και $R=40$

Σε αυτήν την περίπτωση, η ακτίνα επιρροής που επιλέχθηκε για τη συνάρτηση ενεργοποίησης $Wendland C^2$ είναι $R=40$. Έπειτα έγινε στροφή του αεροσκάφους για γωνία, φ . Στον πίνακα 7.3 φαίνεται για τη κρίσιμη γωνία στροφής φ_{crit} πριν το πλέγμα κριθεί ακατάλληλο ο λόγος της μέσης τιμής μετρικής ποιότητας του αναμορφωμένου πλέγματος \bar{a}_{eRBF} προς την μέσης τιμής μετρικής ποιότητας του αντίστοιχου αρχικού πλέγματος \bar{a}_{eInit} .

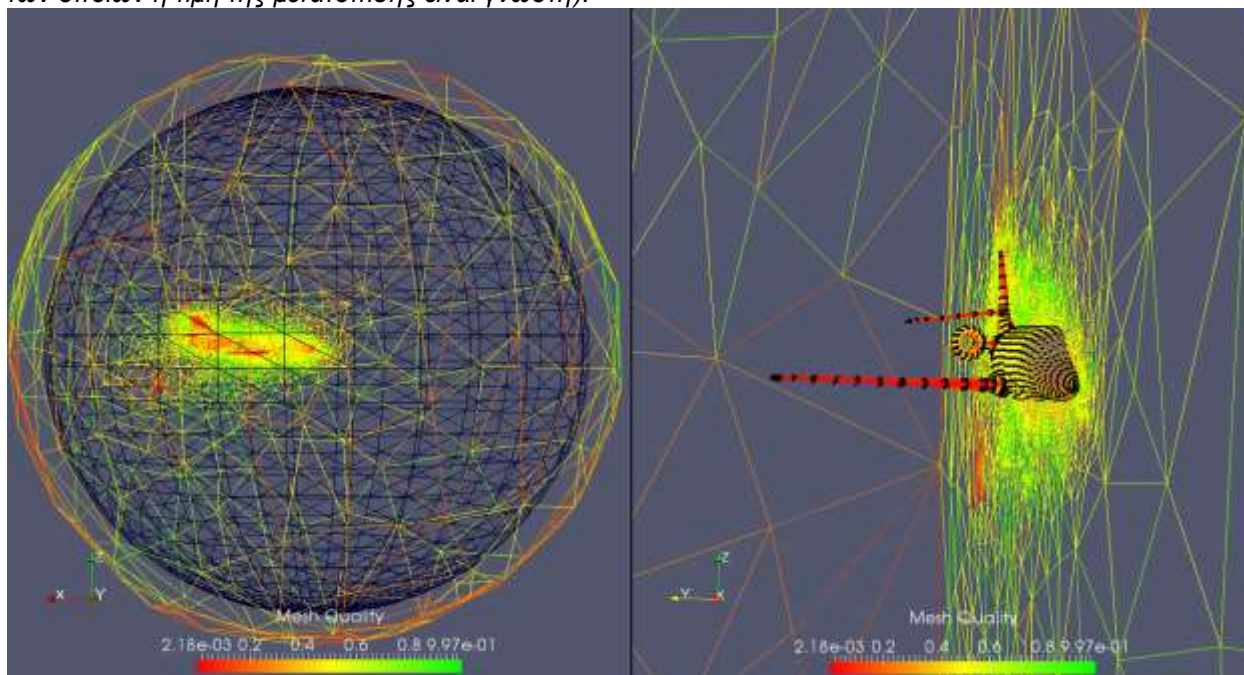
Ενώ στο σχήμα 7.9 φαίνεται το αεροσκάφος στην επιλεγμένη γωνία στροφής, καθώς και το πως διαμορφώθηκε το πλέγμα μετά την εφαρμογή της μεθόδου. Στο σχήμα αυτό έχει γίνει εστίαση στην περιοχή του αεροσκάφους για λόγους ευκρίνειας. Η στροφή της γεωμετρίας σταμάτησε στις 15 μοίρες, πριν το πλέγμα εμφάνισε ένα πολύ μικρό αριθμό ανεστραμμένων στοιχείων.

Ακτίνα επιρροής R	40
$\frac{\varphi_{crit}}{\bar{q}_{eRBF}}$	15
$\frac{\varphi_{eLimit}}{\bar{q}_{eLimit}}$	0.98

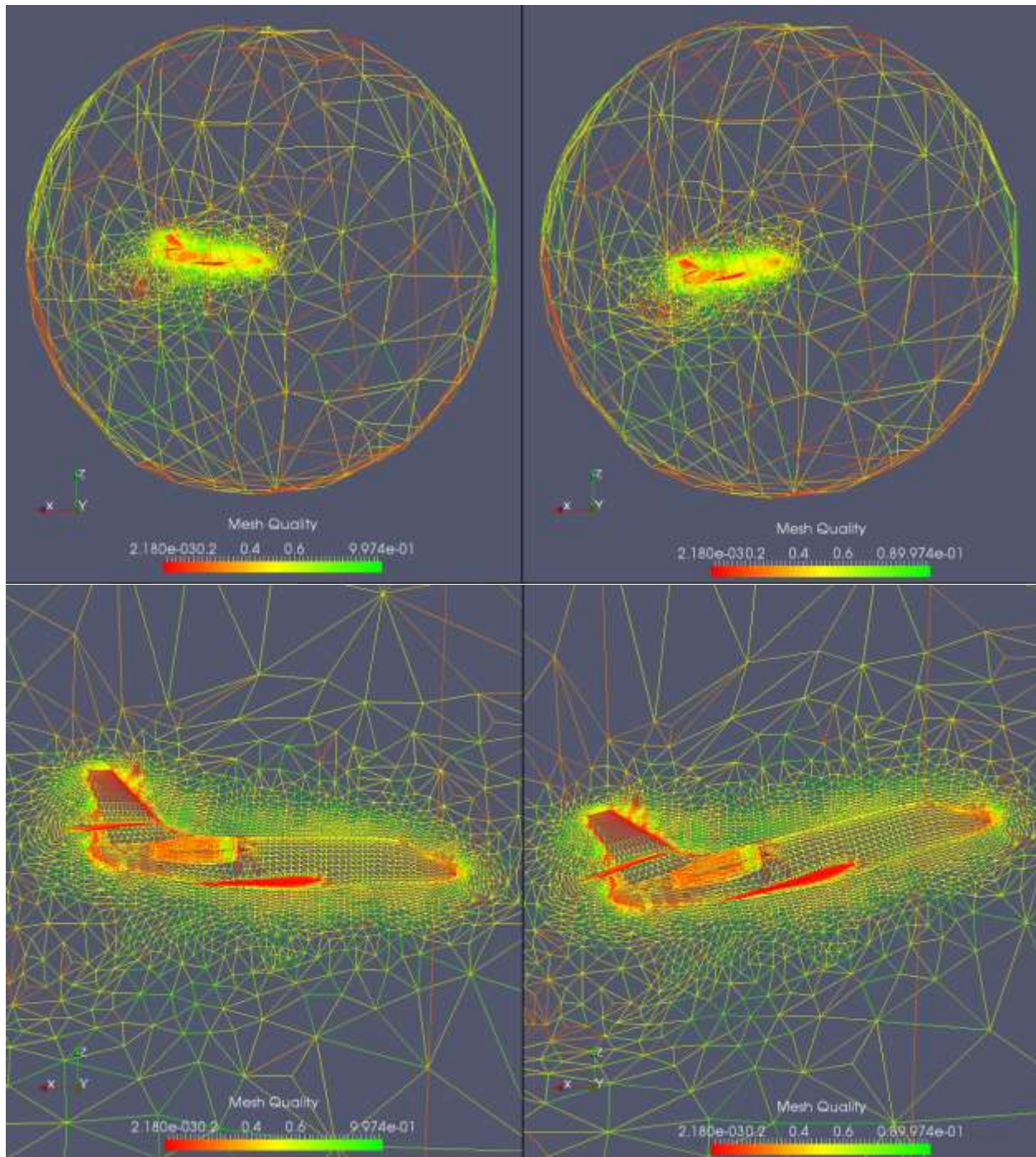
Πίνακας 7.3: Παρατίθενται η τιμή του λόγου της μέσης τιμής μετρικής ποιότητας του αναμορφωμένου πλέγματος \bar{q}_{eRBF} προς την μέση τιμή μετρικής ποιότητας του αντίστοιχου αρχικού πλέγματος \bar{q}_{eLimit} .



Σχήμα 7.7: Με μαύρο χρώμα εμφανίζονται τα κέντρα του δικτύου (όλοι εκείνοι οι κόμβοι ή υποσύνολο αυτών, των οποίων η τιμή της μετατόπισης είναι γνωστή).



Σχήμα 7.8: Με μαύρο χρώμα εμφανίζονται τα κέντρα του δικτύου. Με μαύρο χρώμα εμφανίζεται η σφαίρα επιρροής (με ακτίνα επιρροής $R=40$) ενός εκ των κέντρων το οποίο είναι το κέντρο που βρίσκεται στην αρχή του αεροσκάφους.



Σχήμα 7.9: **Αριστερά:** Το αρχικό πλέγμα. **Δεξιά:** Το αναμορφωμένο πλέγμα για στροφή 15 μοιρών.

7.3 Σχόλια

Σκοπός της εφαρμογής αυτής είναι, όπως αναφέρθηκε, η χρήση του προγραμματισθέντος λογισμικού παραμόρφωσης πλεγμάτων με δίκτυα συναρτήσεων RBF που αναπτύχθηκε, αλλά και η αξιολόγηση του προκύπτοντος πλέγματος. Πραγματοποιήθηκαν διαφορετικού μέτρου στροφές της γεωμετρίας του αεροσκάφους και εξετάστηκε η δημιουργία ανεπιθύμητων ανεστραμμένων τριγώνων. Η δοκιμή αυτή πραγματοποιήθηκε στην ίδια τριδιάστατη γεωμετρία και

χρησιμοποιήθηκε η συνάρτηση ενεργοποίησης $Wendland C^2$ τοπικής επιρροής με ακτίνες επιρροής $R=21$ και $R=40$. Αξίζει να σημειωθεί ότι, η μέση τιμή της μετρικής ποιότητας του πλέγματος, όπως εκφράστηκε με τη μετρική αυτή της μέσης αναλογίας, δεν καθορίζει το πότε το πλέγμα είναι ακατάλληλο, απλά ποσοτικοποιεί την ποιότητα του πλέγματος. Η Ιακωβιανή ορίζουσα του κάθε τετραεδρικού στοιχείου είναι αυτή που καθορίζει το πότε είναι ακατάλληλο το πλέγμα. Όταν η ορίζουσα αυτή ενός, τουλάχιστον, στοιχείου του πλέγματος αποκτήσει αρνητική τιμή, τότε σημαίνει ότι υπάρχει τουλάχιστον ένα ανεστραμμένο στοιχείο, το οποίο καθιστά το πλέγμα ακατάλληλο. Όταν η ορίζουσα ενός τετραεδρικού στοιχείου του πλέγματος αποκτήσει αρνητική τιμή, η αντιωρολογιακή φορά, με την οποία έχουν οριστεί οι κόμβοι αλλάζει λόγω του ανεστραμμένου τετράεδρου, το οποίο καθιστά το πλέγμα ακατάλληλο. Το ανεστραμμένο τετράεδρο έχει μετακινηθεί ανεπιθύμητα μέσα σε κάποιο άλλο τετραεδρικό στοιχείο.

Επίσης τα τετραεδρικά στοιχεία που είναι κοντά στους οριακούς κόμβους, οι οποίοι έχουν επιλεγεί ως κέντρα του δικτύου, επηρεάζονται περισσότερο από τα πιο μακρινά στοιχεία, με αποτέλεσμα να αλλάζει σε μεγαλύτερο βαθμό η ποιότητά τους. Επίσης, στη συγκεκριμένη εφαρμογή λόγω της χαμηλής τιμής της μετρικής ποιότητας των τετραεδρικών στοιχείων του αρχικού πλέγματος (σχήμα 7.3), τα οποία βρίσκονται πολύ κοντά στα επιλεγμένα κέντρα, παρατηρείται ότι είναι δύσκολη η αναμόρφωση του πλέγματος ακόμη και για μικρές στροφές (σχήμα 7.6, 7.9). Για τον λόγο αυτό είναι αναγκαία η μεγάλη αύξηση της ακτίνας επιρροής μέχρι εκεί που δεν θα επηρεάζει τους εξωτερικά οριακούς κόμβους όπως παρουσιάζεται στο σχήμα 7.8.

Αυτό που παρατηρήθηκε είναι ότι επηρεάζεται το πόσο μπορεί να στραφεί η γεωμετρία από την ελάχιστη τιμή μετρικής ποιότητας του αρχικού πλέγματος, όσο χαμηλότερη είναι η ποιότητα εξ αρχής με αποτέλεσμα να υπάρχει μεγαλύτερη ανάγκη για αύξηση της ακτίνας επιρροής. Στην περίπτωση της $Wendland C^2$ με $R=21$, το πλέγμα κρίθηκε ακατάλληλο μετά τις 5 μοίρες περίπου, ενώ όταν εφαρμόστηκε η $Wendland C^2$ με $R=40$ το πλέγμα κρίθηκε ακατάλληλο μετά τις 15 μοίρες. Αξίζει να σημειωθεί ότι οι υπόλοιπες συναρτήσεις δεν αναφέρονται γιατί δημιουργούσαν ανεπιθύμητα τετράεδρα για στροφή πάνω από 3 μοίρες. Επιπλέον, στον πίνακα 6.4 φαίνεται, για κάθε ακτίνα επιρροής της συνάρτησης $Wendland C^2$ η επίδραση στο αναμορφωμένο πλέγμα.

Ακτίνα επιρροής R	21	40
$\frac{\varphi_{crit}}{\bar{q}_{e_{RBF}}}$	5	15
$\frac{\bar{q}_{e_{Int}}}{\bar{q}_{e_{RBF}}}$	1.000	0.99

Πίνακας 7.4: Παρατίθενται η τιμή του λόγου της μέσης τιμής μετρικής ποιότητας του αναμορφωμένου πλέγματος $\bar{q}_{e_{RBF}}$ προς την μέση τιμή μετρικής ποιότητας του αντίστοιχου αρχικού πλέγματος $\bar{q}_{e_{Int}}$.

8 Αποτελέσματα και συγκριτικές επιδόσεις

Στο κεφάλαιο αυτό παρουσιάζονται οι συγκριτικές επιδόσεις ανάμεσα στο πακέτο CULA χρησιμοποιώντας την συνάρτηση DPOTRF (πίνακας 3.5) και στη συνάρτηση Cuda kernel που έχει προγραμματιστεί παράλληλα. Και οι δύο συναρτήσεις διασπούν το συμμετρικό και θετικά ορισμένο μητρώο εκπαίδευσης του δικτύου RBF σε άνω τριγωνικό μητρώο U και σε κάτω τριγωνικό μητρώο U^T , ($A = LU = U^T U$) μέσω της μεθόδου παραγοντοποίησης CHOLESKY. Η ανάπτυξη, η πιστοποίηση και η αξιολόγηση του κώδικα πραγματοποιήθηκε σε κάρτα γραφικών της εταιρίας NVIDIA, της οποίας τα χαρακτηριστικά παρουσιάζονται στον πίνακα 8.1.

CUDA Device	
Major revision number:	2
Minor revision number:	1
Name:	GeForce GT 635M
Total global memory (bytes):	2147155968
Maximum memory pitch (bytes):	2147483647
Total shared memory per block (bytes):	49152
Total registers per block:	32768
Warp size:	32
Number of Stream Multiprocessors:	2
Number of max Threads Per Stream Multiprocessors:	1536
Maximum threads per block:	1024
Maximum dimension 0:x of block:	1024
Maximum dimension 1:y of block:	1024
Maximum dimension 2:z of block:	64
Maximum dimension 0:x of grid:	65535
Maximum dimension 1:y of grid:	65535
Maximum dimension 2:z of grid:	65535
Clock rate (MHz):	950
Memory Bus Width (bits):	128
Total constant memory (bytes):	65536
Peak Memory Bandwidth (GB/s):	28.8
Memory Clock Rate (MHz):	900
L2 cache size (Kbytes):	131
Cuda Cores (SMs x Fermi Cores)	2 x 32=64

Πίνακας 8.1: Παρουσιάζονται τα χαρακτηριστικά της συγκεκριμένης GPU που χρησιμοποιήθηκε στις εφαρμογές.

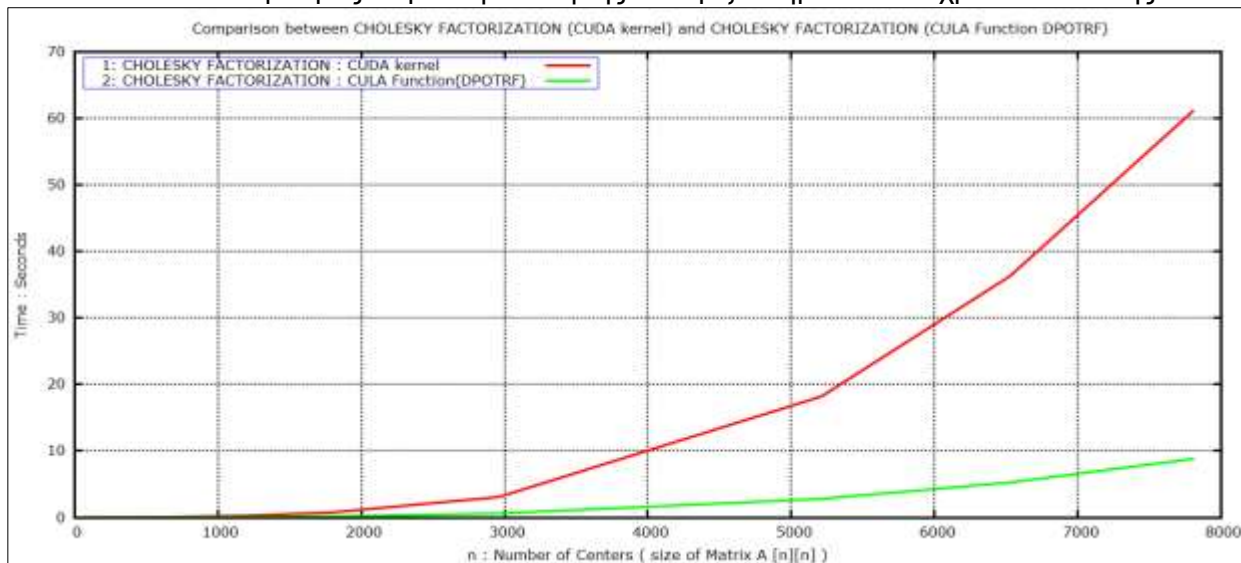
Από τα πιο πάνω χαρακτηριστικά μπορούμε να ισχυριστούμε ότι η κάρτα γραφικών είναι μία κάρτα σχετικά νέας αρχιτεκτονικής (Fermi) αλλά όχι σχεδιασμένη για παράλληλη επεξεργασία (πίνακας 3.1).

8.1 Σύγκριση ανάμεσα στο πακέτο CULA και στη συνάρτηση CUDA kernel

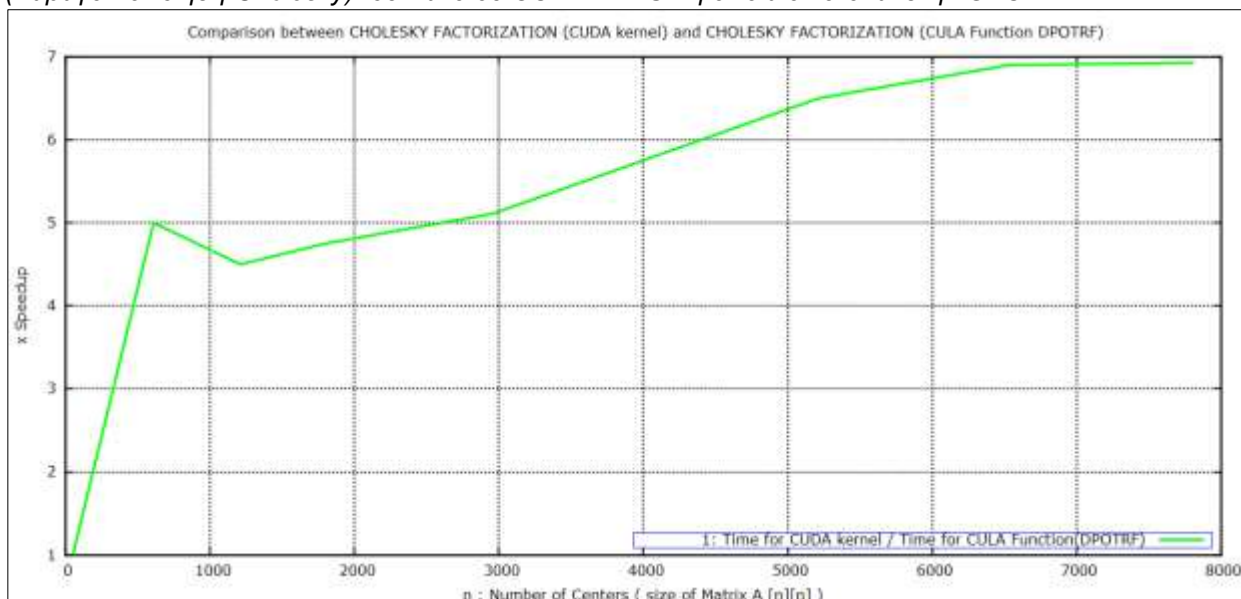
Η συνάρτηση CUDA kernel διασπά το συμμετρικό και θετικά ορισμένο μητρώο A (σχέση 2.2) με τη μέθοδο παραγοντοποίησης CHOLESKY. Η συνάρτηση CUDA kernel έχει προγραμματιστεί παράλληλα και χρησιμοποιεί τους CUDA cores της GPU όπως και το πακέτο CULA. Από την πλευρά του πακέτου CULA επιλέγεται η συνάρτηση DPOTRF (παραγοντοποίησης Cholesky,

πίνακας 3.5) για τη διάσπαση του μητρώου A. Και οι δύο συναρτήσεις καλούνται μέσω πηγαίου κώδικα C/C++. Στη συνέχεια, παρουσιάζονται τα αποτελέσματα των συγκρίσεων.

Στο σχήμα 8.1 παρουσιάζεται το διάγραμμα του χρόνου εκτέλεσης. Όπως φαίνεται στο σχήμα 8.1 όσο ο αριθμός των κέντρων-κόμβων παραμένει χαμηλός η διαφορά στο χρόνο εκτέλεσης δεν είναι μεγάλη. Παράλληλα στο σχήμα 8.2 παρουσιάζεται η επιτάχυνση που εμφανίζει η συνάρτηση του πακέτου CULA σε σχέση με τη συνάρτηση CUDA kernel. Με την αύξηση του αριθμού των κέντρων του δικτύου RBF υπάρχει ξεκάθαρη ανοδική τάση της επιτάχυνσης της συνάρτησης του πακέτου CULA. Όπως φαίνεται και στα διαγράμματα, η εκτέλεση της συνάρτησης DPOTRF του πακέτου CULA είναι 7 φορές πιο γρήγορη από την εκτέλεση της συνάρτησης CUDA kernel. Αποδεικνύεται ότι ο βαθμός παραλληλοποίησης καθορίζει σημαντικά το χρόνο εκτέλεσης.



Σχήμα 8.1: Οι χρόνοι εκτέλεσης της συνάρτησης CUDA kernel σε γλώσσα CUDA C (για την διάσπαση του μητρώου A με τη μέθοδο παραγοντοποίησης Cholesky) και αντίστοιχα της συνάρτησης DPOTRF (παραγοντοποίηση Cholesky) του πακέτου CULA DENSE η οποία εκτελείται στην GPU.



Σχήμα 8.2: Στο διάγραμμα παρουσιάζεται η επιτάχυνση της συνάρτησης DPOTRF (παραγοντοποίησης Cholesky) του πακέτου CULA DENSE η οποία εκτελείται στην GPU σε σχέση με τη συνάρτησης CUDA kernel σε γλώσσα CUDA C (για την διάσπαση του μητρώου A με τη μέθοδο παραγοντοποίησης Cholesky).

9 Ανακεφαλαίωση και Συμπεράσματα

Για την επίλυση ενός προβλήματος αεροδυναμικού σχεδιασμού με κώδικα υπολογιστικής ρευστοδυναμικής (CFD) χρησιμοποιείται η ακόλουθη μεθοδολογία.

1. Παραμετροποίηση του γεωμετρικού μοντέλου (CAD): Κατασκευή του γεωμετρικού μοντέλου που θα μελετηθεί, το οποίο αποτελεί το υπολογιστικό χωρίο του προβλήματος. Διακριτοποίηση του υπολογιστικού χωρίου με τη δημιουργία κατάλληλου υπολογιστικού πλέγματος (ανάλυση ευαισθησίας στη λύση).
2. Επίλυση του συστήματος CFD: Καθορισμός συνωριακών συνθηκών. Αριθμητική επίλυση του μαθηματικού μοντέλου που περιγράφει το πρόβλημα.

Η ανάγκη για αποδοτικότερη λειτουργία και συμπεριφορά ενός προϊόντος οδήγησε στην επαναληπτική διαδικασία της μεθοδολογίας που μόλις αναφέρθηκε, με σκοπό την αξιολόγηση του μεταβαλλόμενο γεωμετρικού μοντέλου, δηλαδή στην δημιουργία ενός κλειστού βρόχου βελτιστοποίησης. Κατά τη διάρκεια ενός βρόχου βελτιστοποίησης των γεωμετρικών χαρακτηριστικών μίας μορφής (αεροτομή), προκύπτει η ανάγκη επαναληπτικής πλεγματοποίησης του υπολογιστικού χωρίου. Στη διαδικασία της βελτιστοποίησης τα γεωμετρικά χαρακτηριστικά του μοντέλου μεταβάλλονται και σε κάθε κύκλο βελτιστοποίησης αξιολογείται η νέα μορφή. Γίνεται αναγκαίο λοιπόν το βήμα της επαναπλεγματοποίησης του νέου υπολογιστικού χωρίου για να επιλυθούν οι εξισώσεις ροής. Η διαδικασία που περιγράφεται στα επόμενα βήματα ονομάζεται «παραμετροποίηση του γεωμετρικού μοντέλου μέσα στο βρόχο βελτιστοποίησης» (CAD in the loop)

- 1. Παραμετροποίηση του γεωμετρικού μοντέλου (CAD): Κατασκευή του γεωμετρικού μοντέλου που θα μελετηθεί, το οποίο αποτελεί το υπολογιστικό χωρίο του προβλήματος. Διακριτοποίηση του υπολογιστικού χωρίου με τη δημιουργία κατάλληλου πλέγματος (ανάλυση ευαισθησίας στη λύση).
2. Επίλυση του συστήματος CFD: Καθορισμός συνωριακών συνθηκών. Επίλυση του μαθηματικού μοντέλου που περιγράφει το πρόβλημα.
3. Βελτιστοποίηση: Μεταβολή των γεωμετρικών χαρακτηριστικών της μορφής (αεροτομή).

Στη προσέγγιση “CAD in the loop” οι ενέργειες είναι οι ακόλουθες:

- Αλλαγή των γεωμετρικών παραμέτρων της μορφής(CAD)
- Επαναδημιουργία του γεωμετρικού μοντέλου (CAD)
- Δημιουργία νέου υπολογιστικού πλέγματος (CAD)
- Εισαγωγή του νέου διακριτοποιημένου υπολογιστικού χωρίου προς επίλυση (CFD)

Δυστυχώς η διαδικασία “CAD in the loop” είναι πολύπλοκη και δύσκολη για να αυτοματοποιηθεί. Σε περιπτώσεις με πολύπλοκα γεωμετρικά μοντέλα η νέα διακριτοποίηση του υπολογιστικού χωρίου μπορεί να αποτύχει. Η επαναπλεγματοποίηση δεν ενδείκνυται, ιδιαίτερα σε βιομηχανικές εφαρμογές με μεγάλα τριδιάστατα πλέγματα, δημιουργήθηκε η ανάγκη για την αποφυγή της και την αντικατάσταση αυτής με μία μέθοδο μετατόπισης του υπολογιστικού πλέγματος. Η διαδικασία βελτιστοποίησης με τη μέθοδο μετατόπισης πλέγματος ονομάζεται «παραμετροποίηση του γεωμετρικού μοντέλου έξω από το βρόχο βελτιστοποίησης»(CAD out of the loop).

Στη προσέγγιση “CAD out of the loop” ο σχεδιασμός του γεωμετρικού μοντέλου και η δημιουργία του υπολογιστικού πλέγματος υλοποιείται μία φορά στην αρχή του βρόχου βελτιστοποίησης. Ενώ η μετατόπιση του υπολογιστικού πλέγματος, σύμφωνα με τις μεταβολές των γεωμετρικών χαρακτηριστικών του μοντέλου λόγω της βελτιστοποίησης, είναι αυτή η διαδικασία που ακολουθεί τον βρόχο βελτιστοποίησης. Η επιθυμητή παραμόρφωση του υπολογιστικού πλέγματος και η αξιολόγηση της ποιότητας του μετατοπισμένου πλέγματος είναι αυτά που εισέρχονται στη διαδικασία βελτιστοποίησης.

Στη προσέγγιση “CAD out of the loop” η διαδικασία είναι η εξής:

1. Παραμετροποίηση του γεωμετρικού μοντέλου (CAD): Κατασκευή του γεωμετρικού μοντέλου που θα μελετηθεί, το οποίο αποτελεί το υπολογιστικό χωρίο του προβλήματος. Διακριτοποίηση του υπολογιστικού χωρίου με τη δημιουργία κατάλληλου πλέγματος (ανάλυση ευαισθησίας στη λύση).
2. Βελτιστοποίηση: Μεταβολή των γεωμετρικών χαρακτηριστικών της μορφής (αεροτομή).
3. Μετατόπιση του υπολογιστικού πλέγματος.
4. Επίλυση του συστήματος CFD: Καθορισμός συννοριακών συνθηκών. Επίλυση του μαθηματικού μοντέλου που περιγράφει το πρόβλημα.
5. Εξαγωγή των γεωμετρικών χαρακτηριστικών της νέας γεωμετρίας

Η διαδικασία “CAD out of the loop” επιτρέπει την αποφυγή της επαναληπτικής πλεγματοποίησης του υπολογιστικού χωρίου με άμεσο αποτέλεσμα την επιτάχυνση της διαδικασίας βελτιστοποίησης, τη μείωση της πιθανότητας αποτυχίας της αυτοματοποιημένης διαδικασίας και τη μείωση του συνολικού χρόνου σχεδιασμού της μορφής – προϊόντος.

Σε αυτήν τη μεταπτυχιακή εργασία αναπτύχθηκε και προγραμματίστηκε λογισμικό μετατόπισης πλέγματος με δίκτυα RBF ώστε να επιταχυνθεί ο κύκλος βελτιστοποίησης και στη συνέχεια αυτό χρησιμοποιήθηκε σε διάφορες εφαρμογές. Όμως η μετατόπιση πλέγματος με δίκτυα RBF παρουσιάζει το μειονέκτημα ότι έχει μεγάλο υπολογιστικό κόστος, ιδίως όταν το πλήθος των κέντρων του είναι μεγάλο. Για αυτό το λόγο, αναπτύχθηκε κώδικας σε γλώσσα προγραμματισμού CUDA C για να εκτελείται παράλληλα σε μονάδα επεξεργασίας γραφικών – GPU με σκοπό την επιτάχυνση της εκπαίδευσης του δικτύου το οποίο συνήθως έχει, όπως αναφέρθηκε, μεγάλο υπολογιστικό κόστος. Η ανάγκη για ταχύτητα στην μετατόπιση του υπολογιστικού χωρίου απαιτεί υψηλό βαθμό παραλληλοποίησης του κώδικα που υλοποιεί τα δίκτυα RBF για αυτό το λόγο έγινε χρήση της βιβλιοθήκης CULA όπου αυτό ήταν αναγκαίο και χρήσιμο. Ένα τμήμα της εκπαίδευσης του δικτύου είναι η επίλυση τριών (αν το πρόβλημα είναι τριδιάστατο) ή δύο (αν το πρόβλημα είναι διδιάστατο) γραμμικών συστημάτων, τα οποία έχουν ίδιο μητρώο συντελεστών. Στην παρούσα εργασία τα γραμμικά συστήματα επιλύονται με τη μέθοδο Cholesky διότι το μητρώο είναι συμμετρικό και θετικά ορισμένο, χρησιμοποιώντας τη συνάρτηση DPOSV του υπολογιστικού πακέτου CULA η οποία εκτελείται παράλληλα στην GPU.

9.1 Συμπεράσματα για τα δίκτυα RBF

Μετά την αξιολόγηση των εφαρμογών στα κεφάλαια 5, 6 και 7 έχουν εξαχθεί τα ακόλουθα συμπεράσματα σε ότι αφορά τη χρήση των δικτύων RBF για την μετατόπιση του πλέγματος.

1. Αποτελεσματικά στην μετατόπιση υπολογιστικών πλεγμάτων όπου το γεωμετρικό μοντέλο χαρακτηρίζεται από μεγάλες και απότομες μεταβολές των γεωμετρικών του χαρακτηριστικών όπως το φύλλο πλατάνου.
2. Αποτελεσματικά στην μετατόπιση υπολογιστικών πλεγμάτων με μεγάλες στροφές του γεωμετρικού μοντέλου όπως στην περίπτωση της αεροτομής.
3. Λειτουργικά στην μετατόπιση «δύσκολου» υπολογιστικού πλέγματος με χαμηλή αρχική ποιότητα πλεγματικών στοιχείων όπως στην περίπτωση του αεροσκάφους.
4. Διατήρηση της ποιότητας του υπολογιστικού πλέγματος σε ικανοποιητικά επίπεδα ακόμη και σε μεγάλες μεταβολές των γεωμετρικών χαρακτηριστικών του γεωμετρικού μοντέλου.
5. Χρήσιμο εργαλείο για περιπτώσεις λ.χ. αεροελαστικότητας και αεροακουστικής.
6. Χρήσιμα από την πλευρά της συνάρτησης επιρροής. Με τη χρήση συναρτήσεων τοπικής επιρροής, άμεσα γίνεται επιλογή της περιοχής του υπολογιστικού χωρίου που είναι απαραίτητο να επηρεαστούν και έμμεσα δημιουργείται αραιό μητρώο εκπαίδευσης του δικτύου RBF. Με την κατάλληλη επιλογή αποθήκευσης του μητρώου εκπαίδευσης μπορεί να γίνει μεγάλη εξοικονόμηση στην μνήμη της CPU αλλά και της GPU. Επίσης δίδεται η ευκαιρία για χρήση της έκδοσης SPARSE της βιβλιοθήκης CULA με συνδυασμό του κατάλληλου επιλύτη και προσταθεροποιητή για την συγκεκριμένη περίπτωση.

9.2 Συμπεράσματα για τη βιβλιοθήκη CULA

Τα τελευταία χρόνια, έχει παρουσιαστεί μεγάλο ενδιαφέρον από τη Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής & Βελτιστοποίησης του ΕΜΠ για την εκμετάλλευση των δυνατοτήτων των GPU σε εφαρμογές υπολογιστικής ρευστοδυναμικής και βελτιστοποίησης. Στον τομέα της υπολογιστικής ρευστοδυναμικής έχει παρουσιάσει εντυπωσιακά αποτελέσματα σε επιλύτες ροής ατρίβους και συνεκτικού ρευστού με επιταχύνσεις μεγαλύτερες από 40 φορές [39], [40].

Στην παρούσα μεταπτυχιακή εργασία, πιο συγκεκριμένα, αναπτύχθηκε κώδικας παραμόρφωσης πλέγματος (RBF displacement), για δομημένο, μη δομημένο και υβριδικό πλέγμα. Ο κώδικας προγραμματίστηκε σε γλώσσα CUDA C, η οποία αποτελεί επέκταση της γλώσσας C/C++, με στόχο την, όσο το δυνατόν, πιο αποδοτική εκμετάλλευση των δυνατοτήτων μίας GPU αρχιτεκτονικής Fermi. Τα αποτελέσματα που προκύπτουν από την επίλυση με τον κώδικα που αναπτύχθηκε στο πλαίσιο της παρούσας εργασίας είναι τα αποτελέσματα που παρουσιάστηκαν πιο πάνω. Η σύγκριση των επιδόσεων του υπολογιστικού πακέτου CULA (μέθοδος παραγοντοποίησης CHOLESKY) με τη συνάρτηση Cuda kernel (μέθοδος παραγοντοποίησης CHOLESKY) η οποία είχε προγραμματιστεί παράλληλα (κεφάλαιο 8) παρουσίασε επιτάχυνση ως 7x. Δε τίθεται θέμα σύγκρισης όταν τα συστήματα που έχουμε να επιλύσουμε είναι μικρά. Όμως ο βαθμός παραλληλοποίησης κρίνεται καθοριστικός σε μεγάλα συστήματα για αυτό το λόγο κρίνεται αποτελεσματική η επιλογή και η χρήση του υπολογιστικού πακέτου CULA.

Μετά την αξιολόγηση των αποτελεσμάτων της σύγκρισης που έγινε στο κεφάλαιο 8 αλλά και των συγκρίσεων που έγιναν και δεν αναφέρθηκαν στην εργασία έχουν εξαχθεί τα ακόλουθα συμπεράσματα σε ότι αφορά τη χρήση της βιβλιοθήκης CULA.

1. Ταχύτερη ως 7 φορές η συνάρτηση παραγοντοποίησης του μητρώου εκπαίδευσης, DPOTRF(μέθοδος Cholesky) της βιβλιοθήκης CULA σε σύγκριση με παράλληλο

κώδικα(CUDA kernel) που παραγραμματίστηκε για την παραγοντοποίηση του μητρώου με την ανάλογη μέθοδο.

2. Ταχύτερη ως 2 φορές η συνάρτηση πολλαπλασιασμού του μητρώου M με το διάνυσμα των βαρών για κάθε κατεύθυνση σε σύγκριση με ανάλογο παράλληλο κώδικα(CUDA kernel) που αναπτύχθηκε για εκτέλεση στην GPU.
3. Για μικρό αριθμό αγνώστων του γραμμικού συστήματος δεν είναι αναγκαία η χρήση της.
4. Υψηλή απόδοση σε μεγάλης τάξης προβλήματα.
5. Σε πραγματικής κλίμακας προβλήματα με αριθμό αγνώστων της τάξης των εκατομμυρίων η χρήση της βιβλιοθήκης γίνεται επιτακτική.

Στο προσεχές μέλλον θα παρουσιαστούν ακόμα μεγαλύτερες επιταχύνσεις, τόσο σε εφαρμογές υπολογιστικής ρευστοδυναμικής όσο και σε άλλους τομείς, καθώς αναπτύσσεται περισσότερο η τεχνολογία αλλά και η τεχνογνωσία γύρω από τη χρήση των GPU σε εφαρμογές γενικού προγραμματισμού. Τελικά, η μετατροπή ενός ήδη υπάρχοντα κώδικα της CPU σε κώδικα κατάλληλο για εκτέλεση από την GPU είναι χρονοβόρα και απαιτητική εργασία καθώς πρέπει ένας σειριακός κώδικας, προορισμένος για την CPU η οποία εκτελεί πράξεις σειριακά, να μετατραπεί σε έναν πλήρως παράλληλο κώδικα για να είναι αποδοτική η χρήση της GPU, η οποία εκμεταλλεύεται πλήρως παράλληλα τους επεξεργαστές της. Όμως, μία τελική επιτάχυνση όπως η παραπάνω κάνει εμφανές το γιατί όλο και περισσότερο αυξάνεται το ενδιαφέρον της επιστημονικής κοινότητας για το GPGPU σε επιστημονικές εφαρμογές, όπως είναι η επιτάχυνση εφαρμογών για υπολογιστικά πλέγματα και ειδικότερα για προβλήματα βελτιστοποίησης μορφής.

10 Βιβλιογραφία

- [1] Παπαδημητρίου, Δ.: Συζυγείς (Adjoint) διατυπώσεις για την ανάλυση–σχεδίαση πτερυγώσεων στροβιλομηχανών και τη βέλτιστη προσαρμογή πλέγματος με *aposteriori* εκτίμηση σφάλματος. Διδακτορική διατριβή, Εργαστήριο Θερμικών Στροβιλομηχανών, Ε.Μ.Π., Αθήνα, 2007.
- [2] Καμπόλης, Ι.: Πολυεπίπεδοι, πολυεπεξεργαστικοί αλγόριθμοι αεροδυναμικής βελτιστοποίησης στις στροβιλομηχανές. Διδακτορική διατριβή, Εργαστήριο Θερμικών Στροβιλομηχανών, Ε.Μ.Π., Αθήνα, 2009.
- [3] Ζυμάρης, Α.: Συζυγείς μέθοδοι για το σχεδιασμό μορφών με βέλτιστη αεροδυναμική συμπεριφορά σε στρωτές και τυρβώδεις ροές. Διδακτορική διατριβή, Εργαστήριο Θερμικών Στροβιλομηχανών, Ε.Μ.Π., Αθήνα, 2010.
- [4] Παπουτσής-Κιαχαγιάς, Ε.: Συζυγείς Μέθοδοι για το Στιβαρό Σχεδιασμό Αεροδυναμικών Μορφών με Υπολογισμό Τρίτων Παραγώγων. Εφαρμογή σε Μονοδιάστατους Αγωγούς. Διπλωματική Εργασία. Εργαστήριο Θερμικών Στροβιλομηχανών, Ε.Μ.Π., 2009.
- [5] Χ., Γιαννάκογλου Κ.: Μέθοδοι Βελτιστοποίησης στην Αεροδυναμική. Πανεπιστημιακές Εκδόσεις ΕΜΠ, 2006.
- [6] Elizarov, A.M., Il'inskiy, N.B., and Potashev, A.V.: *Mathematical Methods of Airfoil Design: Inverse Boundary Value Problems of Aerohydrodynamics*. Wiley, 1997.
- [7] Λιακόπουλος, Π.: Γένεση μη-δομημένων πλεγμάτων και διαχείρισή τους σε μεθόδους ανάλυσης και βελτιστοποίησης συνιστωσών στροβιλομηχανών και εφαρμογές, αξιοποιώντας τεχνολογίες πλέγματος (*Grid computing*). Διδακτορική διατριβή, Εργαστήριο Θερμικών Στροβιλομηχανών, Ε.Μ.Π., Αθήνα, 2008.
- [8] Kampolis I.C., Zymaris A.C, Asouti V.G. and Giannakoglou K.C.: *Multilevel Optimization Strategies based on Metamodel-Assisted Evolutionary Algorithms, for Computationally Expensive Problems*. 2007 IEEE Congression Evolutionary Computation CEC, Singapore, September 25-28, 2007.
- [9] Κοντολέοντος, Ε.: Ανάλυση και βελτιστοποίηση σύνθετων ρευστομηχανικών διατάξεων με χρήση υπολογιστικής ρευστοδυναμικής και εξελικτικών αλγορίθμων. Διδακτορική διατριβή, Εργαστήριο Θερμικών Στροβιλομηχανών, Ε.Μ.Π., Αθήνα, 2012.
- [10] M.E., Biancolini, Viola, Ignazio Maria, and M., Riotte: *Sails trim optimization using CFD and RBF mesh morphing*, 2014. Export Date: 6 March 2014 Source: Scopus.
- [11] Sieger, D. and Menzel, S. and Botsch, M.: *RBF Morphing Techniques for Simulation-based Design Optimization*. *Engineering with Computers*, 30(2):161–174, 2014.
- [12] Παπουτσής-Κιαχαγιάς, Ε.: Συζυγείς μέθοδοι χαμηλού υπολογιστικού κόστους για τυρβώδεις ροές, στην αεροδυναμική βελτιστοποίηση. Διδακτορική διατριβή, Εργαστήριο Θερμικών Στροβιλομηχανών, Ε.Μ.Π., Αθήνα, σε εξέλιξη.
- [13] Ζερβογιάννης, Θ.: Μέθοδοι βελτιστοποίησης στην αεροδυναμική και τις στροβιλομηχανές με χρήση συζυγών τεχνικών, υβριδικών πλεγμάτων και του ακριβούς εσσιανού μητρώου. Διδακτορική διατριβή, Εργαστήριο Θερμικών Στροβιλομηχανών, Ε.Μ.Π., Αθήνα, 2011.
- [14] Καββαδίας, Ι.: Αεροδυναμική βελτιστοποίηση για τυρβώδεις ροές με χρήση συζυγών μεθόδων και επεξεργαστών καρτών γραφικών. Διδακτορική διατριβή, Εργαστήριο Θερμικών Στροβιλομηχανών, Ε.Μ.Π., Αθήνα, σε εξέλιξη.
- [15] Lynch, D.: *Unified approach to simulation on deforming elements with application to phase change problems*. *Journal of Computational Physics*, 47(3):387 – 411, 1982.

- [16] Lynch, D. and O'Neil, K.: *Elastic grid deformation for moving boundary problems in two space dimensions*. In *3rd International Conference, Mississippi University, Oxford, May 19-23 1980*.
- [17] Batina, J.: *Unsteady Euler algorithms with unstructured dynamic mesh for complex-aircraft aerodynamic analysis*. *AIAA Journal*, 29(3):327-333, 200.
- [18] Degand, C. and Farhat, C.: *A three-dimensional torsional spring analogy method for unstructured dynamic meshes*. *Computers & Structures*, 80(3-4):305-316, 2002.
- [19] Braess, H. and Wriggers, P.: *Arbitrary Lagrangian Eulerian finite element analysis of free surface flow*. *Computer Methods in Applied Mechanics & Engineering*, 190(1-2):95-109, 2000.
- [20] Boer, A. de, Schoot, M. S. van der, and Bijl, H.: *Mesh deformation based on radial basis function interpolation*. *Comput. Struct.*, 85(11-14):784-795, jun 2007.
- [21] Wendland, H.: *Konstruktion und Untersuchung radialer Basisfunktionen mit kompaktem Trager*, 1996.
- [22] *Radial Basis Functions*, webpage. en.wikipedia.org/wiki/Radial_basis_functions.
- [23] V. Volkov, and J. W. Demmel, *Benchmarking GPUs to tune dense linear algebra*, SC08
- [24] NVIDIA, *NVIDIA CUDA Compute Unified Device Architecture Programming Guide*, version 2.3, Aug 2009.
- [25] Jason Sanders. Edward Kandrot. :*CUDA by Example - An Introduction to General-Purpose GPU Programming*. Addison-Wesley, 2010.
- [26] *Whitepaper. NVIDIA'S Next Generation CUDA Compute Architecture: Fermi*
- [27] *Whitepaper. NVIDIA'S Next Generation CUDA Compute Architecture: Kepler GK110/210*
- [28] <http://www.emphotonics.com/>
- [29] <http://www.culatools.com/>
- [30] *CULA Reference Manual*. www.culatools.com. Release 1.1. EM Photonics, Inc. November 25, 2009.
- [31] <http://www.nvidia.com/>
- [32] *Design Guide*. NVIDIA Co., *CUDA C Best Practices Guide*. September 2015
- [33] Mark Harris. NVIDIA Co. *Optimizing Parallel Reduction in CUDA*.
- [34] Damir A. Jamsek, IBM Research (2009), *Designing and optimizing compute kernels on NVIDIA GPU*, *Proceedings of the 2009 Asia and South Pacific Design Automation Conference*.
- [35] *CULA Reference Manual*. www.culatols.com. Release R17 (CUDA 5.0) EM Photonics, Inc. www.emphotonics.com. April 15, 2014
- [36] *CULA Programmer's Guide*. www.culatools.com. Release R17 (CUDA 5.0) EM Photonics, Inc. www.emphotonics.com. April 15, 2014
- [37] G. H. Golub, C. F. Van Loan, *Matrix computations*, Third edition, 1996

[38] M. T. Heath, E. Ng, B. W. Peyton, *Parallel algorithms for sparse linear systems*, *Parallel Algorithms for Matrix Computations*, 1987

[39] V. G. Asouti, X. S. Trompoukis, I. C. Kampolis, K. C. Giannakoglou. *Unsteady CFD computations using vertex-centered finite volumes for unstructured grids on Graphics Processing Units*, *International journal for numerical methods in fluids*, *Int. J. Numer. Meth. Fluids* (2010).

[40] Βαλσαμάκης Γεώργιος. Αριθμητική επίλυση μη-μόνιμου πεδίου ροής σε κάρτες γραφικών με απεικόνιση του σε «πραγματικό» χρόνο, Διπλωματική εργασία, Εργαστήριο Θερμικών Στροβιλομηχανών ΕΜΠ, Φεβρουάριος 2010.

[41] <http://www.nvidia.com/object/what-is-gpu-computing.html>

[42] <http://www.nvidia.com/object/cuda-parallel-computing-platform.html>

[43] Jayshree Ghorpade, Jitendra Parande, Madhura Kulkarni, Amit Bawaskar.: *GPGPU PROCESSING IN CUDA ARCHITECTURE*, *Advanced Computing: An International Journal*, Vol.3, No.1, January 2012

[44] *Programming Massively Parallel Processors: A Hands-on Approach (Applications of GPU Computing Series) 1st Edition* by David B. Kirk (Author), Wen-mei W. Hwu (Author)

[45] *CUDA Handbook: A Comprehensive Guide to GPU Programming, The 1st Edition* by Nicholas Wilt (Author)

[46] *GPU Computing Gems Jade Edition (Applications of GPU Computing Series) 1st Edition* by Wenmei W. Hwu (Editor)

[47] Erik Lindholm, John Nickolls, Stuart Oberman, John Montrym. *NVIDIA. TESLA: A UNIFIED GRAPHICS AND COMPUTING ARCHITECTURE*

[48] Τρομπούκης, Ξ.: *Αριθμητική Επίλυση Προβλημάτων Αεροδυναμικής – Αεροελαστικότητας σε Επεξεργαστές Καρτών Γραφικών. Διδακτορική διατριβή, Εργαστήριο Θερμικών Στροβιλομηχανών, Ε.Μ.Π., Αθήνα, 2012.*