



TECHNICAL UNIVERSITY OF CRETE

SCHOOL OF PRODUCTION ENGINEERING & MANAGEMENT

**On the Numerical Solution of
Compressible Fluid Flow and Radiative Heat Transfer Problems**

By

Georgios N. Lygidakis

A dissertation submitted in partial fulfilment of the
requirements for the degree of
Doctor of Philosophy (PhD)

Supervisor: Dr. Ioannis K. Nikolos, Associate Professor

Chania, March 10th, 2015



TECHNICAL UNIVERSITY OF CRETE

SCHOOL OF PRODUCTION ENGINEERING & MANAGEMENT

On the Numerical Solution of Compressible Fluid Flow and Radiative Heat Transfer Problems

By

Georgios N. Lygidakis

Supervised and approved by:

Dr. Ioannis K. Nikolos

Associate Professor
Technical University of Crete
School of Production Engineering
& Management

Dr. Anargyros I. Delis

Associate Professor
Technical University of Crete
School of Production Engineering
& Management

Dr. Dimitrios V. Rovas

Assistant Professor
Technical University of Crete
School of Production Engineering
& Management

Approved by:

Dr. Georgios Karatzas

Professor
Technical University of Crete
School of Environmental
Engineering

Dr. Sokrates Tsangaris

Professor
National Technical University of
Athens
School of Mechanical Engineering

Dr. George Papadakis

Reader in Aerodynamics
Imperial College
Faculty of Engineering, Department
of Aeronautics

Dr. Ioannis Anagnostopoulos

Associate Professor
National Technical University of
Athens
School of Mechanical Engineering

"Intentionally left blank"

To my family
Vili, Nick and Nadia

"Intentionally left blank"

Abstract

In this study the development of a methodology for the numerical solution of steady-state compressible fluid flow and radiative heat transfer problems is reported. For the discretization of the computational domains, three-dimensional unstructured hybrid grids with tetrahedral, prismatic and pyramidal elements are employed along with a node-centered finite-volume scheme. Flow modelling is achieved via the Reynolds-Averaged Navier-Stokes (RANS) equations, along with appropriate two-equation turbulence models, namely, $k-\varepsilon$ (in three versions), $k-\omega$ and SST . For the computation of the inviscid fluxes an upwind method, applying Roe's approximate Riemann solver, is implemented, together with a higher-order accurate spatial scheme, while for the viscous ones the required gradients are evaluated with an element-based (edge-dual volume) or a nodal-averaging approach. The time advancement of the aforementioned equations is achieved with either an explicit scheme, applying a second-order temporal accurate four-stage Runge-Kutta (RK(4)) method, or an implicit one, implementing the Jacobi or the Gauss-Seidel algorithm. For the modelling of radiative heat transfer in general enclosures through absorbing, emitting, and either isotropically or anisotropically scattering gray media, the time-dependent or steady (non time-dependent) Radiative Transfer Equation (RTE) is employed. Similarly to fluid flow, a second-order accurate spatial scheme along with appropriate slope limiters is applied to increase accuracy of the solution, especially at the boundary regions, while time integration is obtained with simple iterative approximations or the same to flow model explicit scheme. In order to increase the efficiency of the proposed methodology additional acceleration techniques are used, namely, an edge-based data structure, a parallelization strategy based on the domain decomposition approach and MPI library functions, and an agglomeration multigrid method employed in isotropic or directional formulation for the flow solver and in spatial, angular or nested spatial/angular one for radiative heat transfer algorithm. Finally, the h-refinement technique is incorporated to increase accuracy at pre-selected regions of the examined grid, by enriching them with more nodes during the solution procedure; as a result, the generation of dense meshes from the very beginning is avoided. Based on the aforementioned methods, an academic CFD code, named *Galatea*, was developed; it has been validated against three- and quasi-three-dimensional benchmark test cases presented in the open literature, while its results have been compared with wind tunnel experimental data as well as results obtained by reference numerical solutions, confirming its capability to effectively perform such simulations in terms of accuracy, geometric flexibility and computational efficiency.

Keywords: Compressible flow, RANS equations, turbulence modelling, radiative heat transfer, three-dimensional unstructured hybrid grids, node-centered finite-volume scheme, parallel processing, agglomeration multigrid method, FAS, FMG, h-refinement.

Περίληψη

(Extended Abstract in Greek)

Σκοπός της παρούσας Διδακτορικής Διατριβής ήταν η ανάπτυξη μεθοδολογίας για την αριθμητική επίλυση προβλημάτων μόνιμης ροής συμπιεστού ρευστού και μετάδοσης θερμότητας μέσω ακτινοβολίας. Η υπόψη μέθοδος δύναται να χρησιμοποιηθεί σε ένα ευρύ φάσμα εφαρμογών τόσο σε ακαδημαϊκό όσο και σε βιομηχανικό επίπεδο, π.χ., σε θαλάμους καύσης, βιομηχανικούς φούρνους, εσωτερική ή εξωτερική αεροδυναμική ροή, κινητήρες τύπου ramjet, κ.λπ.

Πιο συγκεκριμένα, αναπτύχθηκε μεθοδολογία τύπου χρονοπροέλασης για την προσομοίωση ατρίβους και συνεκτικής στρωτής ή τυρβώδους μόνιμης συμπιεστής ροής. Η διακριτοποίηση των εξισώσεων Navier-Stokes επί τρισδιάστατων τετραεδρικών ή υβριδικών μη-δομημένων υπολογιστικών πλεγμάτων επιτυγχάνεται με την κεντροκομβική μέθοδο πεπερασμένων όγκων (FVM/Finite Volume Method). Για την μοντελοποίηση τυρβώδους ροής εφαρμόζονται οι κατά Reynolds ολοκληρωμένες εξισώσεις Navier-Stokes (RANS/Reynolds-Averaged Navier-Stokes), χρησιμοποιώντας την υπόθεση Boussinesq και επακόλουθα τον όρο της τυρβώδους συνεκτικότητας, για τον υπολογισμό της οποίας συμπεριλήφθηκαν τρία διαφορετικά μοντέλα τύρβης, το k-ε, το k-ω και το SST (Shear Stress Transport). Για την εκτίμηση των μη-συνεκτικών διανυσμάτων ροής εφαρμόζεται ο προσεγγιστικός επιλύτης του Roe, θεωρώντας ένα τοπικό μονοδιάστατο πρόβλημα Riemann στη διεπαφή των γειτονικών όγκων ελέγχου. Αύξηση της ακρίβειας του προαναφερθέντος υπολογισμού επιτυγχάνεται με την εφαρμογή σχήματος δεύτερης τάξης χωρικής ακρίβειας, βασισμένου στην τεχνική MUSCL (Monotonic Upwind Scheme for Conservation Laws). Το εν λόγω σχήμα συνδυάζεται με κατάλληλη συνάρτηση περιορισμού (Van Albada-Van Leer, Min-mod ή Barth-Jespersen) προκειμένου να διασφαλιστεί η μονοτονία μεταξύ των τιμών των μεταβλητών των γειτονικών όγκων ελέγχου. Η εκτίμηση των συνεκτικών διανυσμάτων ροής προϋποθέτει τον πρωτύπο υπολογισμό των παραγώγων των συνιστωσών της ταχύτητας και της θερμοκρασίας στη διεπαφή των όγκων ελέγχου, η οποία συμπίπτει με το μέσο της ακμής που συνδέει τους αντίστοιχους υπολογιστικούς κόμβους. Για τον υπόψη υπολογισμό εισήχθησαν δύο τεχνικές στην παρούσα μεθοδολογία, εκ των οποίων η πρώτη βασίζεται στη δημιουργία νέων δυικών όγκων ελέγχου γύρω από την υπό εξέταση ακμή (edge-dual volume method), ενώ σύμφωνα με τη δεύτερη μέθοδο οι επιθυμητές παράγωγοι προκύπτουν από τις αντίστοιχες των ακραίων κόμβων της ακμής (nodal-averaging method). Ο υπολογισμός τόσο των μη-συνεκτικών όσο και των συνεκτικών διανυσμάτων ροής εκτελείται με σάρωση των ακμών του πλέγματος, χρησιμοποιώντας κατάλληλες δομές δεδομένων (edge-based data structures), προκειμένου να μειωθεί όσο το δυνατόν ο απαιτούμενος υπολογιστικός χρόνος. Η χρονική ολοκλήρωση και τελική κατάσταση της ροής προσεγγίζεται επαναληπτικά, είτε με ρητό σχήμα, εφαρμόζοντας την μέθοδο Runge-Kutta τεσσάρων βημάτων (RK(4)) και δεύτερης τάξης χρονικής ακρίβειας, είτε με σημειακά πεπλεγμένο σχήμα, εφαρμόζοντας τον αλγόριθμο Jacobi ή τον αλγόριθμο Gauss-Seidel. Για την επιτάχυνση της επίλυσης εφαρμόζεται επιπρόσθετα η τεχνική του τοπικού ψευδο-χρονικού βήματος (local time-stepping technique). Τέλος, σημειώνεται ότι για τις εξισώσεις των μοντέλων τύρβης ακολουθείται παρόμοια με τις εξισώσεις ροής διαδικασία χρονικής ολοκλήρωσης και υπολογισμού των διανυσμάτων ροής, εκτός του προσεγγιστικού επιλύτη του Roe και του σχήματος δεύτερης τάξης χωρικής ακρίβειας, καθώς η κύρια συνεισφορά στα εν λόγω μοντέλα προέρχεται από τους συνεκτικούς όρους.

Για την μοντελοποίηση της μετάδοσης θερμότητας μέσω ακτινοβολίας σε γκρι μέσο με δυνατότητα απορρόφησης, εκπομπής και σκέδασης εφαρμόζεται η εξίσωση μεταφοράς ακτινοβολίας (RTE/Radiative Transfer Equation) με παρουσία (time-dependent) ή μη (non time-dependent) χρονικού όρου. Καθώς η υπόψη εξίσωση θα πρέπει να επιλυθεί για κάθε πεπερασμένο όγκο ελέγχου και για κάθε πεπερασμένη στερεά γωνία ελέγχου, διακριτοποιείται χωρικά και χρονικά. Η εν λόγω χωρική διακριτοποίηση επιτυγχάνεται με τη κεντροκομβική μέθοδο πεπερασμένων όγκων, ομοίως των εξισώσεων ροής, ενώ η γωνιακή παρομοιάζεται με το χωρισμό της επιφάνειας «σφαίρας» (περιμέτρου «κύκλου» στις δύο διαστάσεις) σε μικρότερες αντίστοιχες των επιθυμητών στερεών γωνιών ελέγχου. Για τον υπολογισμό των διανυσμάτων θερμοροής στη διεπαφή των όγκων ελέγχου εφαρμόζεται απλό άναντες σχήμα (step scheme), σύμφωνα με το οποίο η ένταση της ακτινοβολίας στη διεπαφή τίθεται ίση με αυτή του κόμβου στα ανάντη της θερμοροής. Για την αύξηση της ακρίβειας της λύσης χρησιμοποιείται σχήμα δεύτερης τάξης χωρικής ακρίβειας, αντίστοιχο αυτού για τα μη-συνεκτικά διανύσματα ροής των εξισώσεων Navier-Stokes. Το υπόψη σχήμα συνδυάζεται με τη συνάρτηση περιορισμού Van Albada-Van Leer ή Min-mod, προκειμένου να διασφαλιστεί η μονοτονία μεταξύ των τιμών της έντασης της ακτινοβολίας των γειτονικών κόμβων, ιδιαίτερα κοντά στις οριακές επιφάνειες. Το πρόβλημα επικάλυψης όγκων ελέγχου και στερεών γωνιών ελέγχου, στο οποίο αναπόφευκτα οδηγείται η εν λόγω μέθοδος λόγω συνδυασμού μη-δομημένων πλεγμάτων και γωνιακής διακριτοποίησης, αντιμετωπίζεται με την εφαρμογή της Bold Approximation ή της Pixelation Technique. Αν και κατά τη συνήθη τακτική οι οριακές συνθήκες επιβάλλονται ρητά σε τέτοιους είδους μεθοδολογίες, στην παρούσα εργασία υιοθετήθηκε η πεπλεγμένη επιβολή τους, επιτρέποντας τη χρήση αραιότερων πλεγμάτων στην περιοχή των οριακών επιφανειών, καθώς και την ανάπτυξη και εφαρμογή οριακών συνθηκών τύπου συμμετρίας. Τέλος, η προσέγγιση της τελικής λύσης της εξίσωσης επιτυγχάνεται είτε με επαναληπτική διόρθωση των τιμών (non time-dependent RTE) είτε με χρονική ολοκλήρωση και εφαρμογή της μεθόδου Runge-Kutta τεσσάρων βημάτων (RK(4)) και δεύτερης τάξης χρονικής ακρίβειας (time-dependent RTE).

Με στόχο την επιτάχυνση της αριθμητικής επίλυσης τόσο των προβλημάτων ροής όσο και των προβλημάτων μετάδοσης θερμότητας μέσω ακτινοβολίας (επιπλέον της διάρθρωσης των δεδομένων κατά τις ακμές του υπολογιστικού πλέγματος και της εφαρμογής της τεχνικής του τοπικού ψευδο-χρονικού βήματος), χρησιμοποιήθηκε μεθοδολογία παράλληλης επεξεργασίας και πολυπλέγματος. Η μέθοδος παραλληλοποίησης βασίζεται στην τεχνική διαμέρισης του υπολογιστικού πλέγματος σε μικρότερα υποχωρία (domain decomposition approach), ώστε να καταστεί δυνατή η ταυτόχρονη επίλυση των εξισώσεων σε αυτά. Η όλη διαδικασία ξεκινάει στον κύριο επεξεργαστή με την εφαρμογή του λογισμικού METIS, το οποίο διαχωρίζει τους κόμβους του αρχικού πλέγματος σε μικρότερα σύνολα κόμβων (core nodes). Ωστόσο, με αυτόν τον τρόπο τα στοιχεία του πλέγματος στα όρια των υπο-συνόλων (υπο-πλεγμάτων) παραμένουν ανολοκλήρωτα, καθώς δεν περιέχονται όλοι οι κόμβοι τους στα αντίστοιχα σύνολα. Για την ολοκλήρωση τους προστίθενται οι ελλείποντες κόμβοι (ghost nodes) στα αντίστοιχα υπο-πλέγματα, δημιουργώντας ταυτόχρονα μία περιοχή αλληλοκάλυψης ανάμεσα τους, ενώ στη συνέχεια διανέμονται τα απαραίτητα δεδομένα και στους υπόλοιπους επεξεργαστές. Η ανταλλαγή πληροφορίας μεταξύ των γειτονικών υπο-πλεγμάτων, που αφορά στις τιμές των μεταβλητών καθώς και των παραγώγων αυτών, επιτυγχάνεται μέσω των κόμβων στις περιοχές αλληλοκάλυψης και τις εντολές του πρωτόκολλου επικοινωνίας MPI (Message Passing Interface). Οι προαναφερθείσες τιμές των ghost κόμβων δεν υπολογίζονται από τις εξισώσεις μεταφοράς, αλλά λαμβάνονται απευθείας από τους αντίστοιχους core κόμβους των γειτονικών υπο-πλεγμάτων.

Η μέθοδος πολυπλέγματος (multigrid method) στην παρούσα Διδακτορική Διατριβή εφαρμόζεται σε χωρική μορφή για την περίπτωση προβλημάτων ροής και μετάδοσης θερμότητας μέσω ακτινοβολίας,

καθώς και σε γωνιακή και συνδυασμένη χωρική-γωνιακή μορφή μόνο για προβλήματα ακτινοβολίας. Η χωρική μέθοδος βασίζεται στη δημιουργία μίας σειράς αραιότερων του αρχικού πλεγμάτων και στην επίλυση προσεγγιστικών εξισώσεων σε αυτά, με σκοπό την ταχύτερη σύγκλιση στην τελική λύση. Για τη δημιουργία αυτών των πλεγμάτων εφαρμόζεται η τεχνική συσσωμάτωσης (agglomeration approach), σύμφωνα με την οποία οι όγκοι ελέγχου των γειτονικών κόμβων ενώνονται δημιουργώντας ένα νέο υπερ-κόμβο του αραιότερου υπο-πλέγματος. Η υπόψη συσσωμάτωση, η οποία δύναται να είναι είτε ισότροπη (isotropic) είτε κατευθυνόμενη (directional), πραγματοποιείται με τρόπο τέτοιο ώστε να διατηρεί την αρχική τοπολογία, ενώ ξεκινάει από τις οριακές επιφάνειες και εκτείνεται προς το εσωτερικό του πλέγματος προσομοιάζοντας την τεχνική του προελαύνοντος μετώπου (advancing front technique). Ωστόσο, η εν λόγω ένωση υπόκειται σε προκαθορισμένους περιορισμούς, που αφορούν κυρίως στους κόμβους των εσωτερικών και εξωτερικών ορίων των υπο-πλεγμάτων, π.χ., ένας οριακός κόμβος δύναται να ενωθεί μόνο με έναν άλλο οριακό του ίδιου είδους, ενώ οι "ghost" κομβοί δε λαμβάνονται υπόψη κατά την κύρια διαδικασία αλλά ενώνονται σύμφωνα με τη συσσωμάτωση που έχουν υποστεί οι αντίστοιχοι τους core κόμβοι στα γειτονικά υπο-πλέγματα. Λαμβάνοντας υπόψη τα ανωτέρω, η όλη διαδικασία ξεκινάει σε κάθε υπο-πλέγμα με την ένωση των όγκων ελέγχου των οριακών κόμβων στερεάς επιφάνειας με αυτούς των γειτονικών τους, επίσης οριακών κόμβων, ενώ στη συνέχεια καταρτίζεται λίστα με τους κόμβους που έχουν έρθει σε επαφή με το "μέτωπο" της συσσωμάτωσης (seed nodes). Η ένωση των κόμβων και η δημιουργία υπερ-κόμβων συνεχίζεται με τη συσσωμάτωση των όγκων ελέγχου των "seed" κόμβων με αυτούς των γειτονικών τους. Κατόπιν καταρτίζεται μία νέα λίστα "seed" κόμβων με τον ίδιο τρόπο και η όλη διαδικασία επαναλαμβάνεται έως ότου εξεταστούν όλοι οι κόμβοι του υπολογιστικού πλέγματος. Στην περίπτωση της κατευθυνόμενης συσσωμάτωσης, η οποία εφαρμόζεται σε πλέγματα υβριδικού τύπου και δύναται να είναι είτε μερική (semi-directional) είτε ολική (full-directional), η διαδικασία ξεκινάει από τους οριακούς κόμβους των πρισματικών στοιχείων και συνεχίζεται στους κόμβους των επόμενων πρισματικών στρωμάτων, διατηρώντας το μοτίβο συσσωμάτωσης των πρώτων. Αφού ολοκληρωθεί η ένωση των πρισματικών κόμβων, η διαδικασία συνεχίζεται ισότροπα για τους υπόλοιπους κόμβους, εκκινώντας από τους γειτονικούς των ανώτερων πρισματικών στρωμάτων. Όπως αναφέρθηκε παραπάνω, οι "ghost" κόμβοι δε λαμβάνονται υπόψη κατά την κύρια διαδικασία, αλλά ενώνονται σύμφωνα με τη συσσωμάτωση που έχουν υποστεί οι αντίστοιχοι τους "core" κόμβοι στα γειτονικά υπο-πλέγματα. Με τον τρόπο αυτό δημιουργούνται "ghost" ψευδο-υπερ-κόμβοι, καθώς ο περικλειόμενος αριθμός κόμβων τους πιθανόν να διαφέρει από αυτόν στους αντίστοιχους core υπερ-κόμβους. Εφόσον απαιτείται ακόμη αραιότερο πλέγμα, η όλη διαδικασία επαναλαμβάνεται. Η επίλυση των εξισώσεων στη εν λόγω σειρά πλεγμάτων πραγματοποιείται με το σχήμα FAS (Full Approximation Scheme) σε κύκλο σχήματος V, σύμφωνα με το οποίο στα αραιότερα πλέγματα επιλύεται μία προσεγγιστική μορφή των εξισώσεων μεταφοράς. Η μεταφορά πληροφορίας σε έναν υπερ-κόμβο από τους περικλειόμενους κόμβους του (restriction) αφορά στις σταθμισμένες κατ' όγκο συντηρητικές μεταβλητές τους και στο άθροισμα των διανυσμάτων ροής τους. Αντίθετα, από τον υπερ-κόμβο μεταφέρονται στους περικλειόμενους κόμβους του (prolongation) οι διορθώσεις των μεταβλητών είτε με απλή μεταφορά στην περίπτωση ατρίβους ροής είτε με μεταφορά σταθμισμένη με την απόσταση μεταξύ τους στην περίπτωση συνεκτικής ροής. Για την επίτευξη ακόμη μεγαλύτερης επιτάχυνσης κατά τα πρώτα στάδια του σχήματος FAS, εφαρμόζεται το σχήμα FMG (Full Multigrid).

Η γωνιακή μέθοδος πολυπλέγματος εφαρμόζεται με ανάλογο της χωρικής τρόπο στη γωνιακή διακριτοποίηση των προβλημάτων μετάδοσης θερμότητας μέσω ακτινοβολίας. Η συσσωμάτωση συνίσταται στην ένωση των γειτονικών στερεών γωνιών ελέγχου, μειώνοντάς τες στο ένα τέταρτο των αρχικών κάθε φορά. Σε αντίθεση με τη χωρική, η γωνιακή συσσωμάτωση υπόκειται σε έναν μόνο

περιορισμό που απαγορεύει την ένωση στερεών γωνιών που ανήκουν σε διαφορετικά τεταρτημόρια. Με αυτόν τον τρόπο διασφαλίζεται η συνέχεια κατά τη μεταφορά της πληροφορίας από την πυκνότερη στην αραιότερη διακριτοποίηση και αντίστροφα. Η επίλυση της RTE στα διαδοχικά αραιότερα γωνιακά "πλέγματα" πραγματοποιείται με το σχήμα FAS ομοίως της χωρικής μεθόδου πολυπλέγματος, ενώ η στάθμιση των μεταβλητών κατά τη μεταφορά τους από το πυκνότερο στο αραιότερο υλοποιείται κατά γωνία (όχι κατ' όγκο). Τέλος, στην παρούσα εργασία αναπτύχθηκε και συνδυασμένη χωρική-γωνιακή μεθοδολογία για την επιτάχυνση προβλημάτων μετάδοσης θερμότητας μέσω ακτινοβολίας. Σύμφωνα με αυτήν, σε κάθε επίπεδο του χωρικού σχήματος πολυπλέγματος εκτελείται ένας πλήρης FAS V-κύκλος του γωνιακού σχήματος, ενώ η μεταφορά της έντασης της ακτινοβολίας, των διανυσμάτων ροής και των διορθώσεων εκτελούνται ομοίως προς τις απλές μεθόδους.

Επιπρόσθετα, με σκοπό την περαιτέρω αύξηση της ακρίβειας της παρούσας μεθόδου και πέραν της εφαρμογής των προαναφερθέντων σχημάτων υψηλότερης τάξης χωρικής ακρίβειας, αναπτύχθηκε μεθοδολογία αυτόματης τοπικής πύκνωσης του πλέγματος. Με αυτόν τον τρόπο δύνανται να αυξηθούν οι βαθμοί ελευθερίας του υπό εξέταση πλέγματος, αποφεύγοντας ταυτόχρονα την απαίτηση κατασκευής ενός νέου εξαρτή. Η υπόψη τεχνική αναδεικνύεται ιδιαίτερα πολύτιμη σε περιπτώσεις εμφάνισης τοπικών φαινομένων, π.χ. περιοχές κυμάτων κρούσης, αποκόλλησης ροής, κ.λπ., καθώς και σε προβλήματα που αντιμετωπίζονται για πρώτη φορά, με συνέπεια να μην είναι εκ των προτέρων γνωστό το απαιτούμενο επίπεδο πύκνωσης στις διάφορες περιοχές του πλέγματος. Η όλη διαδικασία δύναται να διαιρεθεί σε τέσσερα βασικά βήματα: α) Εντοπισμός των περιοχών προς πύκνωση και σημείωση των αντίστοιχων ακμών βάσει ενός προκαθορισμένου κριτηρίου, π.χ. ακμές που περιλαμβάνουν κόμβους με τιμή του αριθμού Mach υψηλότερη της μονάδας. β) Διάχυση της πληροφορίας σημείωσης στις γειτονικές ακμές, ώστε να καταστεί δυνατή η διαίρεση των αντίστοιχων στοιχείων του πλέγματος. γ) Διαίρεση των σημειωμένων ακμών και εισαγωγή νέων κόμβων στο μέσο τους. δ) Διαίρεση των αντίστοιχων πλευρών και στοιχείων του πλέγματος, βάσει προκαθορισμένων κανόνων διαίρεσης, π.χ., ένα πρισματικό στοιχείο δύναται να διαιρεθεί σε δύο ή τέσσερα νέα πρισματικά στοιχεία.

Εφαρμόζοντας τις ανωτέρω τεχνικές αναπτύχθηκε ο ακαδημαϊκός κώδικας Galatea (Γαλάτεια) στα πλαίσια της παρούσας Διδακτορικής Διατριβής. Για την αξιολόγηση του εν λόγω λογισμικού και κατ' επέκταση των προαναφερθεισών μεθόδων εξετάστηκαν διάφορα προβλήματα αναφοράς, ενώ τα αποτελέσματα του συγκρίθηκαν ποιοτικά και ποσοτικά με διαθέσιμα πειραματικά δεδομένα, καθώς και με αριθμητικά αποτελέσματα αντίστοιχων αλγορίθμων αναφοράς. Από τις υπόψη συγκρίσεις διαφαίνεται η δυνατότητα της εν λόγω μεθοδολογίας για τέτοιου είδους προσομοιώσεις τόσο από την πλευρά της ακρίβειας όσο και από την πλευρά της αποδοτικότητας. Εν κατακλείδι, λαμβάνοντας υπόψη τις αντίστοιχες διαθέσιμες στη διεθνή βιβλιογραφία μελέτες, η συνεισφορά της παρούσας εργασίας συνοψίζεται στα κάτωθι:

- Στην ανάπτυξη της γωνιακής μεθόδου πολυπλέγματος.
- Στην ανάπτυξη της συνδυασμένης χωρικής-γωνιακής μεθόδου πολυπλέγματος.
- Στην εφαρμογή σχήματος δεύτερης τάξης χωρικής ακρίβειας με συναρτήσεις περιορισμού για την προσομοίωση της μετάδοσης θερμότητας μέσω ακτινοβολίας.
- Στην ανάπτυξη της πεπλεγμένης επιβολής των οριακών συνθηκών σε προβλήματα μετάδοσης θερμότητας μέσω ακτινοβολίας.
- Στην τοπική προσαρμογή του πλέγματος για την προσομοίωση της μετάδοσης θερμότητας μέσω ακτινοβολίας.

- Στη χρήση υβριδικών πλεγμάτων σε συνδυασμό με την κεντροκομβική μέθοδο πεπερασμένων όγκων σε προβλήματα μετάδοσης θερμότητας μέσω ακτινοβολίας.
- Στη χωρική μέθοδο πολυπλέγματος (δημιουργία ghost ψευδο-υπερ-κόμβων, μεταφορά διορθώσεων από το αραιότερο στο πυκνότερο πλέγμα βάσει απόστασης των κόμβων, κ.λπ.).

Acknowledgments

It is a pleasant duty to express my sincere gratitude to all those who have contributed to the accomplishment of this "mission".

First and foremost, I would like to thank my supervisor Associate Professor Ioannis K. Nikolos for giving me the opportunity to complete this PhD and for introducing me to the world of research, especially to the fascinating field of CFD. His guidance and his support through stimulating discussions was endless during these years, while sometimes it was extended beyond the scientific issues. I am deeply grateful for his encouragement, kindness and ethics.

I would like also to thank Associate Professor Anargyros I. Delis and Assistant Professor Dimitrios V. Rovas for their valuable suggestions and efforts as co-advisors. Moreover, I would like to thank all the members of the examination committee, honouring me by participating to it, and for their valuable comments and suggestions, which greatly improved the quality of the final document.

Many thanks to Mr. Sotirios S. Sarakinos, for all the work we did together and especially for providing the methodology of parallel processing.

The financial support by the ELXIS Engineering Consultants S.A. and the EUROSTARS Project (E!5292) "Structural and aerodynamic design of TUNnels under Fire Emergency Conditions" for a part of this work is gratefully acknowledged.

Last but certainly not least, my gratitude goes to my lovely wife Vili for her patience and support during all these years, and to our children Nick and Nadia. This thesis is dedicated to them.

"Intentionally left blank"

Contents

Abstract	V
Acknowledgements	XI
Contents	XIII
Nomenclature	XVII
1. Introduction	1-1
1.1. Objectives	1-1
1.2. Literature review	1-1
1.2.1. Fluid flow modelling	1-1
1.2.2. Radiative heat transfer modelling	1-4
1.2.3. Parallel computation	1-7
1.2.4. Multigrid methodology	1-8
1.2.5. Grid adaptation	1-9
1.3. Present study	1-10
1.3.1 Overview	1-10
1.3.2 Contributions	1-11
1.3.3 Outline	1-12
1.4. List of publications	1-12
2. Fluid flow	2-1
2.1. Governing equations	2-1
2.1.1. The Navier-Stokes equations	2-1
2.1.2. The Reynolds Averaged Navier-Stokes (RANS) equations	2-4
2.1.3. Turbulence models	2-7
2.1.3.1. Turbulence model $k-\epsilon$	2-7
2.1.3.2. Turbulence model $k-\omega$	2-11
2.1.3.3. Turbulence model SST	2-11
2.2. Spatial discretization	2-13
2.3. Numerical fluxes	2-16
2.3.1. Inviscid fluxes	2-16
2.3.2. Higher-order accurate schemes and slope limiters	2-17
2.3.3. Viscous fluxes	2-20
2.3.4. Turbulence models' fluxes	2-21
2.3.5. Boundary conditions	2-21
2.3.6. Source terms	2-24
2.4. Time integration	2-24
2.4.1. Point-implicit scheme	2-25
2.4.2. Explicit scheme	2-29
3. Radiative heat transfer	3-1
3.1. Governing equation	3-1
3.2. Spatial and angular discretization	3-2
3.3. Mathematical analysis	3-3
3.3.1. Formulation of discrete equations	3-3
3.3.2. Boundary conditions	3-10
3.3.3. High-order accurate spatial scheme	3-11
3.3.4. Iterative solution	3-13

4. Parallel computation strategy	4-1
4.1. Domain decomposition	4-1
4.2. Data exchange	4-2
5. Agglomeration multigrid method	5-1
5.1. Spatial agglomeration multigrid scheme	5-1
5.1.1. Spatial agglomeration strategy	5-1
5.1.1.1. Isotropic agglomeration	5-2
5.1.1.2. Directional agglomeration	5-6
5.1.2. Flux computation and numerical solution	5-9
5.2. Angular agglomeration multigrid scheme	5-13
5.2.1. Angular agglomeration strategy	5-13
5.2.2. Flux computation and numerical solution	5-15
5.3. Combined spatial/angular agglomeration multigrid scheme	5-17
6. Grid adaptation	6-1
6.1. H-refinement	6-1
6.2. Subdivision method	6-4
6.3. Procedure description	6-5
7. Fluid flow numerical results	7-1
7.1. Definitions	7-1
7.2. Inviscid flow over a rectangular wing with a NACA0012 airfoil	7-2
7.3. Laminar flow over a rectangular wing with a NACA0012 airfoil	7-4
7.4. Turbulent flow over a rectangular wing with a RAE2822 airfoil	7-10
7.5. Turbulent flow over a rectangular wing with a NACA0012 airfoil	7-12
7.6. Turbulent flow over the ONERA M6 wing	7-13
7.7. Turbulent flow over the DLR-F6 aircraft	7-17
7.7.1. DLR-F6 wing-body (WB) aircraft configuration	7-18
7.7.2. DLR-F6 wing-body-nacelles-pylons (WBNP) aircraft configuration	7-26
7.8. Turbulent flow over the common research model (CRM)	7-29
7.8.1. CRM wing-body (WB) aircraft configuration	7-30
7.8.2. CRM wing-body-horizontal tail (WBHT) aircraft configuration	7-39
8. Radiative heat transfer numerical results	8-1
8.1. Definitions	8-1
8.2. A hexahedral enclosure with quadratic bases	8-1
8.3. A cubic enclosure	8-5
8.4. A prismatic enclosure	8-18
8.5. A cylindrical enclosure	8-26
8.6. A hemi-cylindrical enclosure	8-33
8.7. An L-shaped enclosure	8-36
8.8. A J-shaped enclosure	8-37
8.9. A hexahedral enclosure with trapezoidal bases	8-39
8.10. A cubic enclosure with three baffles	8-45
8.11. An annular sector	8-50
8.12. A tetrahedral enclosure	8-54

9. Conclusions	9-1
9.1. Summary	9-1
9.2. Contributions	9-3
9.3. Ongoing work	9-6
9.4. Future developments	9-6
10. References	10-1
Appendix A: Jacobian matrix decomposition	A-1
Appendix B: Newton linearization of the viscous fluxes	B-1
B.1. Flow model	B-1
B.1.1. Linearization based on the edge-dual volume method	B-2
B.1.2. Linearization based on the nodal-averaging scheme	B-4
B.2. Turbulence models	B-6
B.2.1. Linearization based on the edge-dual volume method	B-7
B.2.2. Linearization based on the nodal-averaging scheme	B-7
Appendix C: Newton linearization of the free-slip convective fluxes	C-1

"Intentionally left blank"

Nomenclature

$\underline{A}, \underline{L}$	Jacobian matrices	q_i	thermal tensor
A_H	spatial forcing function	q	incident radiative heat flux, W/m^2
A^{MN}	angular forcing function	\vec{R}	flux balance vector
c	propagation speed of radiation in the medium	Re	Reynolds number
\tilde{c}_P	sound speed of node P	R_g	gas constant ($R_g=287.04 \text{ m}^2\text{sec}^{-2}\text{K}^{-1}$)
C_j	Legendre polynomials' coefficients	\vec{S}	source term
c_p	constant pressure specific heat	S_{ij}	strain rate
c_v	constant volume specific heat	\hat{s}	unit vector in s direction of a solid angle
CFL	Courant-Friedrichs-Lewy number	t	time
D_{ci}^{mn}	directional weight of solid angle mn	T_u	turbulence intensity
e	energy per unit mass	T	temperature (K only for radiative heat transfer problems)
$\hat{e}_x, \hat{e}_y, \hat{e}_z$	unit vectors in x, y and z directions	u, v, w	components of the velocity
E	total energy per unit mass (reference energy in radiative heat transfer problems)	\tilde{U}	averaged Roe value of a primitive variable
$\vec{F}, \vec{G}, \vec{J}$	Navier-Stokes and turbulence models PDE's vectors	u_τ	friction velocity
G^*	dimensionless average radiative heat flux	V_P	volume of control volume of a node P
h_t	specific total enthalpy	\vec{W}	conservative variables' vector
I	unit matrix	x, y, z	cartesian coordinates
$I(\vec{r}, \hat{s})$	radiative intensity, $W/m^2 \text{ sr}$	y^+	dimensionless distance
k	turbulent kinetic energy	α_{ci}^{mn}	pixelation coefficient
k_α	absorption coefficient, m^{-1}	β	extinction coefficient, $\beta=k_a+\sigma_s$
M	Mach number	γ	ideal gas constant ($\gamma=1.4$)
N_p	number of nodes (finest grid)	δ_{ij}	Kronecker's delta
N_θ, N_ϕ	number of polar and azimuthal angles (finest resolution)	ΔA_i	part i of surface area of control volume
p	pressure	$\Delta \Omega^{mn}$	discrete control angle
P_j	Legendre polynomials of order j	ε	dissipation rate
P_k	turbulent energy production term	ε_w	wall emissivity
Pr	laminar Prandtl number ($Pr=0.72$)	θ	polar angle
Pr_t	turbulent Prandtl number ($Pr_t=0.9$)	κ	Von Karman constant ($\kappa=0.41$)

μ	laminar viscosity	<i>Subscripts</i>	
μ_t	turbulent viscosity	b	blackbody
ρ	density	h	first non-agglomerated grid
σ	Stefan-Boltzmann constant ($\sigma = 5.67 \cdot 10^{-8} \text{ W/m}^2 \text{ K}^4$)	H	agglomerated grid
σ_s	scattering coefficient, m^{-1}	in	inlet
τ_{ij}	stress tensor	k, ε, ω	equations of turbulence models
τ_w	friction stress tensor	m	medium
φ	azimuthal angle	out	outgoing
Φ	scattering phase function	PQ	edge connecting P and Q nodes
$\bar{\Phi}$	average scattering phase function	P, p	present control volume
ω	specific dissipation rate (scattering albedo $\omega = \sigma_s / \beta$ in radiative heat transfer problems)	Q, q	adjacent control volume
<i>Superscripts</i>		r	reflected
inv	inviscid	rad	radiative
mn	first non-agglomerated angular resolution	t	turbulent
MN	agglomerated angular resolution	ref	<i>reference</i>
vis	viscous	w	wall
\sim	normalized variable		

1. Introduction

1.1. Objectives

As the title of this thesis suggests, its main objective is the development of methodologies capable of solving numerically steady-state compressible fluid flow and radiative heat transfer problems. Such a methodology has many academic and industrial engineering applications, e.g., in aircrafts, engine inlets, industrial furnaces, combustion chambers, ramjet engines, etc. Considering the aforementioned goal, the following objectives were accomplished, while main attention was directed toward the improvement of the obtained solutions' accuracy as well as of the efficiency of the simulations:

- A node-centered finite-volume algorithm for the prediction of inviscid and viscous (laminar and turbulent) compressible fluid flows on unstructured tetrahedral or hybrid three-dimensional grids.
- A corresponding methodology simulating radiative heat transfer.
- Improvement of solutions' accuracy, employing:
 - a) Higher-order accurate schemes along with appropriate limiting functions, especially in case of radiative heat transfer.
 - b) A grid adaptation technique (h-refinement).
- Improvement of methods efficiency, implementing:
 - a) A parallelization strategy.
 - b) A spatial agglomeration multigrid scheme.
 - c) An angular and a combined (nested) spatial/angular agglomeration multigrid methodology for radiative heat transfer simulations.
- Validation of the developed method against benchmark test cases, via the qualitative and quantitative comparison of the obtained results with experimental data as well as with the results of acclaimed corresponding solvers.

This Chapter is continued with a literature review, concerning the main features of this type of algorithms, i.e., fluid flow and radiative heat transfer modelling, parallel computation, multigrid methodology and grid adaptation, while in its final section an overview of the proposed methodology along with its original contributions is summarized.

1.2. Literature review

1.2.1. Fluid flow modelling

The attempt of English mathematician L.F. Richardson around 1920 to predict weather by solving Partial Differential Equations (PDE's) is assumed to be the beginning of Computational Fluid Dynamics or CFD for short, as it included the main four features of CFD: a practical problem, PDE's to model mathematically this problem, a numerical scheme and *computers* (the original sense of word *computers* is utilized here - people, and not machines, doing calculations) [Tor97]. Nevertheless, many years passed for CFD history to actually start, mainly due to the limited computing capabilities; its beginning is dated about the early 1970's [Bla01]. Since then, CFD has denoted the combination of physics, numerical analysis and computer science, employed to simulate fluid flows, while it has routinely been applied to the scientific fields of aeronautics, marine engineering, turbomachinery, meteorology, oceanography, astrophysics, oil recovery and architecture [Tor97, Bla01].

A significant effort has been exerted during the past decades in the field of aerodynamics for the development of industrial, commercial and academic CFD algorithms, mainly due to their capability for analyzing the aerodynamic behaviour of complete aircraft configurations in a relatively short period of time; one of the first such codes was developed by Jameson et al. [Jam86]. Nowadays, the aerodynamics designers rely strongly on the CFD algorithms' results, as in that way they may avoid (initially at least) a

large fraction of the extremely time and money consuming experiments in wind tunnels. The decreased effort, obtained by the employment of CFD codes, especially those based on the Reynolds Averaged Navier-Stokes (RANS) equations, produces finally a reduced design cycle time. Nevertheless, the results obtained even by the most acclaimed methodologies appear not to always match with the corresponding wind tunnel experimental results. As the accuracy of the numerically derived flow affects significantly the lift, drag and general aerodynamic payload of an aircraft, which subsequently influences its final design, a continuous need arises for more accurate and more efficient methods, making CFD an evolving scientific field. Therefore, although various academic and commercial compressible flow solvers have been developed in the past years, many issues concerning the methods of grid generation, discretization, flux computation, turbulence modelling, etc., are still subjects of continuous research.

Considering the previous state, the first issue to be defined for a CFD algorithm is the type of the grid used to represent the physical domain under examination. For the first solvers, structured grids were employed, constructed either by algebraic methods or by using PDE's; in test cases requiring meshes with increased complexity overlapping grids were also utilized, derived by the Chimera technique [Bla01]. As excessive time was demanded for their construction (even weeks in case of a military aircraft), the researchers started to focus on the development of unstructured grid generators [And94, Bla01]. Unstructured grids, usually consisted of tetrahedral elements (triangular faces in two dimensions), offer the largest possible flexibility in the treatment of complex geometries along with reduced time and minimum user intervention for their generation/adaptation (compared to structured ones). In case of a viscous flow simulation a hybrid mesh should be selected instead, including prismatic or hexahedral elements at boundary layer region to resolve effectively the severe anisotropy of the flow at this area which demands *a priori* smaller spacing besides excessive geometric flexibility; tetrahedra are used for the rest of the field, while pyramids fill the transition region (from prismatic to tetrahedral elements) [Kim03, Kal05a]. In that way, not only the accuracy of the final solution is improved, but the number of required elements, faces and edges is decreased, compared to a tetrahedral grid with appropriate spacing, resulting in reduced computational effort and memory requirements [Bla01]. Considering the aforementioned advantages of unstructured grids, they appear to be preferable compared to structured ones for problems concerning external aerodynamic flow over a complex geometry, in which the mesh size is adapted to local geometrical and flow features, e.g., in case of an aircraft. Despite their arbitrary nodal distribution, which typically results to approximately twice the number of edges of a corresponding structured hexahedral mesh and consequently to double computational and memory cost when an edge-based technique is employed, they remain the preferred choice for such simulations [Bar92, Kal96, Bla01, Sor03, Kal05a]. In addition, the unstructured meshes have another important feature as analyzed at a next paragraph, the easy adaptation; it can be performed during the solution procedure to capture more efficiently localized phenomena such as shocks [Loh92, Kal96, Bla01, Kal05a].

For the discretization of the computational field and subsequently the numerical approximation of the governing equations an appropriate modelling strategy is required, such as the finite-volume method, probably being the most widely applied in three-dimensional flow simulations [Jam81, Bar92, Kal96, Lan98, Bla01, Lyr02, Kim03, Kou03, Kal05a, Del11]. According to this scheme the computational domain is discretized into a certain number of arbitrary finite control volumes, at each of which the magnitude of variables is assumed to be constant (for first order schemes) [Kim01]. This method can be implemented to both structured and unstructured grids, while it can be categorized in two main types, the cell-centered and the node-centered (also named vertex-centered), depending on the location of the computational nodes surrounded by the corresponding control volumes [Del11]. In the first case, the mesh elements represent the aforementioned control volumes, while the variables are stored at their barycenters. For the node-centered approach, the control volumes are constructed around the nodes of the grid; several methodologies have been developed for their construction, e.g., by connecting lines defined by edge midpoints, barycenters of faces and barycenters of elements, sharing a node [Kim03, Kal05a]. Although one should expect the wider implementation of the cell-centered scheme, considering that for the same computational grid it includes many more Degrees of Freedom (DoF's) compared to the node-centered one, both options are almost equally used nowadays [Bla01]. Moreover, there exist studies

indicating the equal or even superior potential of the node-centered method in terms of accuracy, compared to the cell-centered one for the same number of Degrees of Freedom [Del11].

Another important issue in CFD methodologies concerns the scheme implemented for the computation of the numerical fluxes. The upwind method, considering a one-dimensional Riemann problem in the direction of the normal vector of each face of a finite control volume (interface between adjacent control cells), appears to be strongly established as the most appropriate model for the evaluation of the inviscid fluxes [Bar89, Dec92, Bla01]. For the solution of this problem approximate Riemann solvers have been developed (Roe, Osher-Engquist, HLLC, etc.), as the exact solution would require excessive computational effort [Roe81, Osh84, Bar89, Bar92, Dec92, Lan98, Bla01, Sor03]. Unfortunately, the discretization of the continuous computational field into finite control volumes results in significant values of numerical diffusion; a remedy to this drawback is the implementation of higher-order spatial schemes, such as the Monotonic Upstream Scheme for Conservation Laws (MUSCL), that reconstructs the variables' values at computational nodes using Taylor series expansion [VanA82, VanL85, Bar89, Bar92, And94, Kal96, Lan98, Bla01, Kal05a]. Moreover, to control the total variation of the reconstructed field and assure monotonicity between the adjacent control volumes, appropriate slope limiters accompany the aforementioned scheme, such as the Van Albada-Van Leer, Min-mod, Barth-Jespersen, Superbee, etc., [VanA82, Swe84, Bar89, Bar92, Bla01]. For the viscous fluxes, the temperature and velocity components' gradients have to be evaluated at the control volumes' interfaces; usually a nodal-averaging or an element-based method (approximate Galerkin finite-element approach, face-centered control volume approach, etc.) are applied [Bra96, Kal96, Bla01, Kal05a]. Although the first approach, especially when considering a simple arithmetic nodal-averaging, is susceptible to odd/even oscillations, it is frequently selected due to the relatively increased computational effort required by the element-based approach [Bra96].

For the simulation of turbulent flows the Direct Numerical Simulation (DNS) and the Large Eddy Simulation (LES) models appear to be the methods producing the most accurate numerical results; nevertheless, their computing requirements for excessively decreased spatial and temporal discretization cannot yet be afforded widely [Bla01]. As such, the RANS equations with an appropriate statistical turbulence model, such as $k-\varepsilon$, $k-\omega$, SST (Shear Stress Transport) or Spalart-Allmaras, is still the most widely employed methodology in engineering applications [Lau74, Saf74, Men03a, Du05, Kal05a]. The aforementioned one- or two-equation models are typical examples of PDE's with source terms, based on the Boussinesq assumption, according to which the turbulent shear stress is associated with the mean rate strain with a linear relationship [Bla01, Kou03, Du05]. One of the most popular turbulence models is revealed to the $k-\varepsilon$ [Lau74, Lar91, Kun92, And94, Ste94, Mav94, Jon97, Yod99, Koo00, Kou03, Du05], being a well established method for the prediction of unsteady mean characteristics of a class of low-speed vortex dominated flows [Koo00]. It has been developed in various formulations, depending mainly on the boundary wall treatment; for example it can be applied at the solid wall boundary region via wall functions [Koo00] or with a one-equation model employed at the same area [Jon97]. Although strong non-linearities of the model may interact with numerical errors in such a way that positivity of the variables k and ε is lost, appropriate treatment can overcome these difficulties quite easily [Lar91, Du05]. Another also widely applied turbulence model is the $k-\omega$ of Wilcox [Saf74, Lar91, Men94, Chi96, Liu96, Pen99, Bre02, Kim03, Men03a], which unlikely the $k-\varepsilon$ does not usually consider wall functions at the solid wall region but allows simple Dirichlet boundary conditions to be specified (low Reynolds number approach). Despite its high level accuracy at this area, it appears to be relatively sensitive to the value of free-stream dissipation rate [Men94, Men03a, Men03b]. This was one of the main motivations for the development of the SST scheme [Men94], which is actually a combination of the aforementioned models. It utilizes the $k-\omega$ at the inner region of the boundary layer and switches via an appropriate blending function to the standard $k-\varepsilon$ at the outer region and at the free shear flow. As such, it has a similar performance to the $k-\omega$ model, but avoids at the same time its strong free-stream sensitivity [Men94, Men03a, Men03b]. Significant efforts have been made for the efficient treatment of source term of the pre-described models, in order to alleviate its rapid increase during the iterative solution of corresponding equations (especially in case of an explicit iterative scheme), a phenomenon usually arising at stagnation

points [Lar91, Mav94, Dur96, Du05]. As a result, various methods have been developed, such as the explicit limiting of the obtained corrections of model's variables [Mav94], use of different time step for the flow and turbulence model PDE's [Mav94], implicit treatment of the source term in case of an explicit iterative scheme [Bla01], utilization of only negative contributions of the source term for the construction of the corresponding Jacobian matrix [Lar91, Koo00], etc. All these models were originally applied in aeronautical applications, but since then have made their way into most industrial, commercial and many research codes. Although they give satisfactory results in a computationally acceptable time, they are still statistical models, so they do not derive turbulent variables' distributions in total agreement with DNS data [Men94].

As CFD concerns numerical methods, the last main issue, which has to be defined is the iterative approximation scheme employed for the solution of flow and turbulence model's equations (in a strongly or loosely coupled manner) [Bla01]. The simplest means of solving iteratively a system of PDE's is the implementation of an explicit scheme [Jam81, And94, Mav94, Ven95, Kal96, Kim03, Sor03, Kal05a], e.g., a multi-stage Runge-Kutta method [Lal88a]. According to this scheme the temporal derivative of conservative variables is discretized using a finite-difference formulation at each time step, after evaluating the flux sum at the same step [Ven95]; in that way the explicit methods call for only simple updates, resulting in easy parallelization and reduced memory requirements. Nevertheless, for large-scale flow simulations, such as flows over aircrafts necessitating millions or tens of millions of computational nodes, the rate of convergence slows down dramatically, resulting in relatively inefficient solution schemes. Implicit schemes appear under certain circumstances more appropriate for fine grids, as they allow for the use of larger CFL (Courant-Friedrichs-Lewy) numbers and consequently for larger time steps, accelerating the solution process [Ven91, Ven95, Gri98, Joh04, Pet07]. Nevertheless, they aren't so simple to implement, compared to the explicit ones, especially in a parallel computational environment, while they call for increased memory storage. The point-implicit schemes (a category of implicit iterative methods), usually implemented with the Jacobi or the Gauss-Seidel iterative algorithms, preserve relatively the simple implementation of explicit schemes, even in parallelized simulations, along with the utilization of large time steps [Kou03].

In order to increase the convergence rate and consequently accelerate the solution procedure independently of the iterative scheme (explicit or implicit), especially in large-scale flow problems, various techniques have been developed during the past years. Considering the use of unstructured grids and the increase of computational effort and memory requirements which entail (especially the hybrid ones), a more sophisticated data structure with indirect addressing, such as an edge-based one, can be utilized [Bar92, Eva92, Bla01, Sor03]. According to this approach, the solver gets information from the examined mesh as sets of nodes connected by an edge; in that way the nature of the hybrid mesh is concealed from the main calculation loops, enhancing the solver with the capability of handling grids composed of complex elements [Eva92]. Besides the main solver, h-refinement methodologies, analyzed at a next paragraph, take advantage from the edge-based data storage, considering that the new nodes, enriching the mesh, are introduced at the middle of the existing edges [Eva92, Loh92]. Another widely applied acceleration method is the local time-stepping technique, defining for each computational node the maximum allowable time step, which consequently results to the improvement of convergence of the simulation [Bla01]. In case of an explicit iterative scheme usually entailing a relatively low convergence rate, an implicit residual smoothing technique may be employed, which enhances the procedure with implicit features and allows for the utilization of larger CFL numbers [Kim03]. Last but certainly not least methods reducing the required wall-clock simulation time are the parallel computation [Ven95, Lan96, Smi96, Kar99] and the multigrid method [Mav94, Mav96, Mav97, Mav98, Car00, Bla01, Sor03, Nis10, Nis11, Nis13], analyzed in detail in the following sections.

1.2.2. Radiative heat transfer modelling

Radiative heat transfer has to be considered and therefore be modelled in the simulation of any engineering problem including combustion, e.g., fire in the internal area of a building or a tunnel, combustion chambers, etc. Therefore, various methodologies were developed during the past decades,

mainly divided in two categories, those using exchange view factors (zonal type approaches) and those utilizing flux models (CFD type approaches) [Bor13].

The first model employing the Radiative Transfer Equation (RTE), which describes mathematically radiative heat transfer in a general enclosure, was the Hottel's zone method [Hot67, Loc81]. Although it has been widely applied, it appears to be relatively inefficient, due to its requirement for pre-computation of the necessary exchange view factors. For complex geometries with a large number of discrete surfaces the time for these evaluations increases excessively; therefore its implementation is limited to enclosures including few discrete surfaces [Loc81]. This shortcoming was alleviated partially by the Monte Carlo method, which considers the automatic computation of exchange view factors, tracking in a random manner selected energy releases [Ste71, Loc81]. Based on the feature of geometric flexibility, the Monte Carlo method became quite popular, though it didn't also achieve satisfactory convergence rates [Loc81]. Nevertheless, for both aforementioned methods belonging to the first category, which depend on the utilization of exchange view factors, adequate accuracy is achieved only in the expense of a relatively high computational cost [Sas95].

The second category of methods, employed for the prediction of radiative heat transfer, is based on flux models (also named differential approximations), which, similarly to CFD, depend on the exchange and balance of fluxes at control cells in which a computational domain is discretized. They have been widely implemented in engineering applications, as they appear to be relatively more computationally efficient, compared to the pre-mentioned zonal type approaches. In a similar manner to CFD solvers, extra accuracy is gained with the increase of DoF's, e.g., by applying a grid adaptation technique, in expense however of an also increased simulation time. Various algorithms involving flux models have been developed and tested during the past decades, such as the six-flux model [Loc81, Hof88, Sas95, Ker00], the Finite-Element Method (FEM) [Raz83, Fiv94, Fum04, Liu04, Liu05, Liu06], the Discrete Ordinates Method (DOM) [Car68, Fiv84, Fiv87, Fiv88, Jam92, Kim98, Rai99, Gri08, Hun11, Coe14] and the Finite-Volume Method (FVM) [Rai90, Chu92, Chu93a, Chu93b, Mur98a, Mur98b, Mur98c, Mat99, Rai99, Kim01, Kim08, Kim10, Cap10, Coe14].

The six-flux model [Hof88] was one of the first CFD type approaches, which became quite popular, mainly due to its capability to retain the important effects of hot enclosure cases along with computational economy by solving directly on the flow domain spatial grid and without needing any special description of the geometry [Ker00]. Hybrid versions of this model have been also developed, such as the hybrid six-flux/zone model. The Discrete Transfer Model (DTM) [Loc81, Sas95], which combines also characteristics of zonal type methodologies [Loc81], was widely applied, due to its potential of preserving the physics of the problem with relatively simple mathematics and controlling its accuracy by increasing the number of rays projected from each physical surface and the number of zones that the domain is divided into; this increase of DoF's results of course to a corresponding increase of required computational resources [Ker00]. Therefore, an accurate surface model, being necessary for the examined geometry to be described, along with accurately shaped control volumes and positioning of the rays are prerequisites in order also accurate solutions to be obtained by this method [Ker00].

Other broadly employed flux models for radiative heat transfer simulations are the Finite-Element Method (FEM) [Raz83, Fiv94, Fum04, Liu04, Liu05, Liu06] and the Discrete Ordinates Method (DOM) [Rai90, Chu92, Chu93a, Chu93b, Mur98a, Mur98b, Mur98c, Mat99, Rai99, Kim01, Kim08, Kim10, Cap10, Coe14]. The popularity of the first model derives mainly from its feature that the radiative intensity in a computational cell can differentiate across this domain and that this variation can be traced to increase degrees of approximation [Liu06]. Despite DOM's inflexibility in anisotropic scattering modelling and angular discretization [Rai99], especially in cylindrically configured enclosures under axisymmetric or non-axisymmetric radiative heat transfer [Car68, Jam92, Gri08, Hun11], it has been applied in various multidimensional problems, providing accurate and computationally efficient solutions [Fiv84, Fiv87, Fiv88, Kim98, Coe14]. For axisymmetric radiative heat transfer in cylindrical enclosures the Modified Discrete Ordinates Method (MDOM) has been also proposed, according to which any set of arbitrary control angles can be selected, while the simplicity of the conventional DOM is retained [Bae98a].

The Finite-Volume Method (FVM), proposed initially by Raithby and Chui [Rai90], is revealed as one of the most viable and robust tools for analyzing radiative heat transfer, mainly due to its simplicity and its applicability to complex geometries. Considering these characteristics, many researchers have focused on this methodology, studying its potential for effective radiative heat transfer prediction in various shapes of enclosures, represented by structured or unstructured tetrahedral (triangular for 2D), hybrid or polygonal grids and Cartesian or cylindrical coordinates. For example, Chui et al. [Chu92, Chu93a, Chu93b] as well as Grissa et al. [Gri08] employed this method on cylindrical enclosures, Kim and Baek treated unstructured meshes, including complex geometries and obstacles [Bae97, Kim01], Murthy and Mathur [Mur98a, Mur98b, Mur98c, Mat99] implemented it on structured, unstructured and hybrid grids, extending it to account also for periodic boundaries, while Kim et al. applied it in unstructured polygonal meshes [Kim08, Kim10], etc. Modifications of this method were also developed, as its coupling with the Mie theory for anisotropically scattering media by Trivic et al. [Tri04a, Tri08], its combination with the lattice Boltzmann method in conjunction with genetic algorithms for transient conduction-radiation heat transfer problems by Das et al. [Das08], the corresponding to MDOM Modified Finite-Volume Method (MFVM) for axisymmetric radiative heat transfer in cylindrical enclosures by Kim and Baek [Kim05b], etc. Other researchers worked on the improvement of its efficiency, such as Kim et al. proposing a parallelized version of it [Kim05a], or on the increase of its accuracy, such as Guedri et al. implementing a higher-order spatial differencing scheme [Gue09] and Hassanzadeh and Raithby employing a second-order radiative transfer equation [Has08].

According to the finite-volume scheme, the RTE is discretized and solved in a discrete number of control cells (similarly to CFD) as well as for a discrete number of finite solid control angles; it is integrated for each direction over the control volume under examination. Conservation of energy, spatial discretization, application of boundary conditions, and, formulation and solution of discrete equations are common procedures for fluid flow and radiation [Rai99, Kim05a]. It is this the main feature making this method attractive for CFD scientists, as it allows radiative heat transfer to be treated in a way familiar to them, compared for example with Monte-Carlo method, which requires additional knowledge and experience considering the employed exchange view factors [Rai99, Ste07]. Consequently, via the finite-volume modelling of radiative heat transfer, the assets of well established in CFD techniques can be utilized, e.g., the employment of hybrid unstructured grids with highly stretched elements at boundary surfaces, resulting in increased accuracy at the same regions, or the incorporation of acceleration techniques, such as parallel computation [Ven95, Lan96, Smi96, Kar99, Kim05a] and multigrid scheme [Mav94, Mav96, Mav97, Mav98, Car00, Bla01, Sor03, Nis10, Nis11, Nis13]. Despite the aforementioned similarities, an important difference exists between them; heat transfer via radiation occurs simultaneously in all directions [Rai99]. Nevertheless, considering its main features it appears to be the first choice for performing radiative heat transfer simulations by scientists familiar with CFD [Rai99, Kim05a], and for enhancing existing finite-volume CFD solvers with additional radiative heat transfer computation capability [Kim96, Ko08].

Based on the test cases solved with the finite-volume scheme during the past years, two main shortcomings have been identified, causing reduction of the method's accuracy: the ray effect and the false scattering [Cha93, Coe02, Cap10]. The first drawback derives from the discretization of the continuous angular field in finite solid control angles, and can be alleviated with a denser angular resolution [Cha93]. It doesn't depend at all on the selected spatial resolution; therefore, it can be observed even to a very fine mesh. On the other hand, the false scattering is associated only to the spatial discretization, resembling false or numerical diffusion in CFD scientific field [Coe02], while its effects can be subdued by the utilization of finer grids (globally or locally employing grid adaptation) or more accurate spatial schemes [Cap10]. Significant efforts have been exerted during the last years to moderate the consequences of this drawback and improve the accuracy of the finite-volume method, such as the development of the second-order RTE [Has08] or of higher-order accurate spatial schemes [Cap10, Gue09].

As mentioned in the previous section a higher-order accurate spatial scheme, based on the MUSCL methodology, considers the reconstruction of variables' values (radiative intensity in case of RTE)

employing the Taylor series expansions [VanA82, VanL85, Bar89, Bar92, And94, Kal96, Lan98, Kal05a, Bla01]; in that way more mesh nodes are taken into account for the computation of radiative intensity numerical fluxes, avoiding significant amounts of false scattering. Limiting by appropriate functions, such as those of Van Albada-Van Leer [VanA82] and Min-mod [Swe84], is required mainly for boundary surfaces, where strong discontinuities arise similarly to shock waves in compressible flow. Unfortunately, higher-order schemes may lead sometimes to negative - unphysical - values of radiative intensity; a fix-up procedure, reverting them to zero or to their previous values [Cap10], can be employed to subdue this problem. Despite this temporary remedy, such values along with the common practice of solving RTE with simple iterative approximations, a procedure being susceptible to oscillations, may lead the simulation even to failure. Although the time-dependent RTE is applied usually to transient-unsteady simulations [Das08, Hun11, Tan02, Cha03], it can be employed along with more sophisticated iterative schemes to mitigate the aforementioned problem. Such a scheme is the multi-stage Runge-Kutta method [Lal88a], formulated suitably to provide a second-order temporal discretization [Bla01], considering that a higher-order spatial scheme shall be accompanied by an also higher-order temporal one.

1.2.3. Parallel computation

Large-scale test cases involving grids with millions or tens of millions DoF's, e.g., in flows over aircrafts, result in very time-consuming simulations; thus, considerable efforts were exerted during the past decades for the development of appropriate strategies to enhance the existing numerical solvers with the capability of solution of such problems in a parallel computation environment. These efforts were significantly augmented by the rapid increase of computing capabilities during the same decades, as well as by the widespread availability of multi-processor computer systems. According to the so called Flynn's taxonomy [PARCOMP], proposed by the American Professor Michael J. Flynn in 1966, the following classifications of multi-processor computer architecture, based on the two independent dimensions of Instruction Stream and Data Stream, can be distinguished [Fly98, PARCOMP]:

- SISD (Single Instruction-Single Data Stream),
- SIMD (Single Instruction-Multiple Data Stream),
- MISD (Multiple Instruction-Single Data Stream),
- MIMD (Multiple Instruction-Multiple Data Stream).

From the aforementioned categorization, the first group does not actually correspond to parallel computing (a single processor/process is assigned), while the SIMD or the slightly different SPMD (Single Program-Multiple Data Stream, corresponding to the same program but not necessarily to the same instruction) appears to be the most widely applied one in CFD numerical methods, based on the domain decomposition methods [Loh92, Ven95, Lan96, Smi96, Gri98, Sor03, Kim05a]. Considering the aforementioned governing PDE's, the domain decomposition approach entails the process of distributing and exchanging data among the processors/processes in a distributed memory computational system. In this context, domain decomposition indicates the methodologies developed for decomposing a data structure; thus, data decomposition is perhaps a more appropriate term [Smi96].

In CFD simulations the domain decomposition approach is implemented via the division of the examined grid among the available processors/processes (almost always one process corresponds to one processor); a mesh partitioning application, such as METIS, can be used for this division. Special care is required in order decomposition to be performed in a judicious manner, since it has a significant impact on the parallel performance of the solver via the work balance and the communication load among the processors [Tai03, Kal05a]. Since initial partitioning of computational field is accomplished, two main strategies can be employed in order the communication among the adjacent sub-domains to be established; the first is based on overlapping of interface cells, whose nodes are assigned to multiple partitions, as well as the exchange of values of the variables and nodal gradients, while the second considers straightforward the exchange of numerical fluxes at interfacing faces with no overlapping areas

[Tai03, Kal05a]. Although the first strategy might seem more memory consuming, it appears to be quite efficient, as for an explicit especially iterative scheme it is not the memory usage that should concern, but the communication cost between the adjacent sub-grids [Kal05a]. Finally, an appropriate interface protocol, such as the Message Passing Interface (MPI), is required in order *internal boundary* data of each sub-domain to be exchanged and the iterative communication to be achieved [Loh92, Ven95, Sor03, Tai03, Kal05a].

1.2.4. Multigrid methodology

Although unstructured mesh solvers offer the largest possible flexibility in the treatment of complex geometries, they appear to be relatively inferior in terms of efficiency compared to structured ones [Mav99, Car00]. A remedy to this significant shortcoming is the multigrid methodology; a scheme originally introduced to increase the convergence rate of the numerical solution of elliptic problems [Bra84, Mav97, Fer02], but since then it has made its way into most types of numerical simulations in CFD. Its main idea derives from an observation identified to most of the well-established iterative methods; they converge more slowly on finer resolutions as information is carried out to only one node per iteration (one-dimensional problem) [Fer02]. As a result information has to go back and forth on the computational domain several times during an iterative solution, generating both high and low frequency errors, whose damping defines actually the convergence rate of the simulation [Fer02]. The commonly implemented iterative schemes succeed in efficient relaxation of the first errors, but they seem to be relatively inefficient against the low frequency ones [Fer02]. The multigrid method, which is based on the solution of governing PDE's on successively coarser grids, actually transforms these low frequency errors in high frequency ones on the coarser resolutions, allowing for their efficient damping [Fer02]. The solutions, obtained by the coarser meshes in a relatively negligible computation time, are prolonged to these of the finer grids up to the initial finest one, requiring as such an associating relation between two successive grids. More specifically, appropriate operators, namely restriction and prolongation operators, have to be defined; the first one considers a method for transferring smoothed values of flux balances and/or conservative variables from the finer to the coarser grid (restriction procedure), while the second operator includes a scheme for interpolating conservative variables' corrections from the coarser to the finer grid (prolongation) [Fer02]. Many researchers have focused on the multigrid methodology during the past years, developing various types; their differences are mainly identified on the generation of the coarser resolutions as well as on their associating relation [Mav94, Bra96, Liu96, Mav96, Mav97, Mav98, Car00, Dar06, Bla01, Fer02, Han02, Sor03, Tai03, Liu06, Kat09, Vak09, Nis10, Nis11, Nis13].

Considering the way the coarser resolutions are generated, the multigrid method can be divided in two main types, the geometrical and the agglomeration approach [Bla01]. According to the first classification, the sequence of the coarser grids is obtained either by the construction from scratch of completely independent grids with a corresponding application (non-nested grids approach) or by the generation of associated grids (nested) [Bla01]. If the second (topological) method is selected, the successive grids can be obtained either by starting from the coarsest one enriching it with more mesh nodes via a grid adaptation technique, such as h-refinement, or by starting from the finest one removing nodes and implementing re-triangulation [Bla01]. In a similar way to the previous topological approach, the second (agglomeration) multigrid type, initially proposed by M.H. Lallemand [Lal88b], begins from the initial finest grid generating the sequence of required successive coarser grids via merging the adjacent control volumes; in this way coarser meshes of irregular polyhedral elements are constructed. Despite the satisfactory performance of this scheme for inviscid flow problems along with tetrahedral grids, some studies indicate the reduced gained acceleration in case of its implementation for viscous flow simulations accompanied by hybrid meshes with highly stretched elements at the boundary layer region [Mav97, Mav98, Lam04, Nis10, Nis11, Nis13]. A semi-coarsening or directional-coarsening agglomeration technique was proposed by D.J. Mavriplis [Mav97, Mav98] to alleviate the effects of the aforementioned shortcoming. According to this method, the control volumes of nodes belonging to stretched elements (prisms or hexahedrons) are fused together only if they are located in the normal to the boundary surface direction, while for the rest nodes of tetrahedral elements the isotropic procedure is

employed; thus, the grid anisotropy is moderated as the generated coarser meshes appear to be less stretched than the initial finest one [Lam04]. Alternatively, full coarsening directional agglomeration can be implemented, which considers the fusion of control cells on a topology-preserving framework, resembling the advancing front technique [Han02, Nis10, Nis11, Nis13]. In accordance with this method, the procedure begins by merging the solid wall boundary control cells, while next a line-agglomeration step is performed fusing control volumes along implicit lines starting directly above the boundary volumes [Nis11]; compared to semi-coarsening technique, this agglomeration type seems to result to more efficient solutions, as it allows for greater reduction of DoF's between two successive grids. Finally, an alternative to the aforementioned types of multigrid scheme (geometrical and agglomeration) is the Algebraic Multigrid (AMG), which considers the construction of a coarsening matrix rather than the generation of any grid topology [Vak09].

The second feature distinguishing types of multigrid method, concerns the associating relation and consequently the way in which data are transferred between two successive grids. The initially introduced multigrid schemes were based on the Full Multigrid (FMG) method [Bra84], according to which after fully or partially relaxing the governing PDE's on the coarsest grid the obtained solution is interpolated to the finer one and utilized as an initial guess. The procedure is repeated up to the finest resolution, achieving in that way a better and simultaneously cheaper initial guess/solution, considering that most simulations begin from a uniform unphysical state [Fer02]. Alternatively, the Full Approximation Scheme (FAS) can be implemented, in which at each multigrid cycle the governing PDE's are solved only for the finest resolution, while for the coarser ones approximate versions of the same equations are relaxed [Fer02, Sor03, Lam04, Nis10, Nis11, Nis13]; the solutions obtained by the coarser grids correspond actually to smoothed versions of the finest mesh solution [Fer02]. More specifically, since a solver's external iteration is complete on the finest grid, the values of conservative variables and flux balances are restricted to the coarser level; this process of solution and restriction is repeated up to the coarsest mesh [Fer02]. At next, the conservative variables' corrections are transferred via prolongation from each coarser level to its finer one, up to the initial finest mesh. In this way a V-cycle process is accomplished, being probably the most commonly applied strategy; otherwise a W-cycle strategy can be followed [Bla01]. In addition, a nested combination of the pre-mentioned methods has been proposed in the open literature aiming to improve even more the gained acceleration; it considers the incorporation of FAS approach in the FMG scheme and the division of the solution procedure in two main stages, namely the preliminary and the main one [Lam04]. At the first stage the governing PDE's are relaxed, beginning from the coarsest grid and, as the number of iterative cycles increases, the solution extends successively to the finer meshes up to the initial finest one, at which point the main stage begins. In that way a better initial guess is obtained for each finer FAS level, resulting subsequently to an even faster convergence of the simulation to the final steady-state [Fer02].

1.2.5. Grid adaptation

As already mentioned, in order for the effects of numerical diffusion (or false scattering for radiative heat transfer problems) to be subdued, finer grids can be employed, which however increase the memory and computing requirements [Cap10]. Alternatively, a grid adaptation technique can be implemented to improve the accuracy at the region of interest, without increasing significantly the computational requirements. The same methodology can be followed in test cases including sufficiently localized phenomena, such as shocks in compressible fluid flows, or in problems encountered for the first time, for which no previous knowledge exists about the necessary spatial resolution [Loh92]. The main idea is to refine the mesh in regions with large solution gradients and maintain or even coarsen it in areas where the solution doesn't vary considerably [Kal05b]. Considering additionally the easiness of its application along with solvers utilizing unstructured grids and an edge-based data structure [Eva92, Loh92], it can be employed during the solution procedure to increase the accuracy at user- or automatically-defined areas, achieving significant computational savings, as it avoids the generation of a new mesh from scratch. Many studies have been performed for grid adaptation, resulting in methods, classified mainly in two types, redistribution and refinement [Kal96, Kha00].

The first classification considers the reposition of the mesh nodes in order a more effective resolution in a selected region to be obtained; it targets test cases involving grids, constituted by a sufficient total number of nodes to capture the whole phenomenon, but with insufficient point distribution in a specific area of the field for satisfactory accuracy [Kha00]. For example, redistribution of prismatic nodes based on their dimensionless distance (y^+) values has been implemented in the study of Kallinderis and Baron [Kal92] to resolve more accurately the viscous stresses at this region [Kal05b]. Moreover, in [Fel03], [Ili99] and [Pai01] edge and face swapping strategies are followed in order to align the mesh under study with flow behaviour [Kal05b]. In general, two main redistribution approaches have prevailed, the Spring Systems, handling computational meshes as systems of springs, and the Moving Finite-Element method, according to which the positions of mesh nodes are assumed unknown in order to be computed as well [Loh92]. Despite this method appears to be relatively simple, it encounters difficulties in cases of complex geometries, requiring more DoF's.

The second category (refinement), considers the enrichment of the mesh under examination with more DoF's, following specific rules; it is basically divided in two types, namely p-refinement and h-refinement. According to the first one, extra DoF's are added following hierarchical shape functions [Loh92], while it even allows for different orders of spatial approximation throughout the computational domain [Kal05b]. The h-refinement technique effects the enrichment of the examined grid by splitting the existing elements into new ones [Kal05a]; it begins with the detection of desired areas for adaptation and the division of the corresponding mesh edges along with the embedding of new nodes as midpoints of them. The division is extended then to the faces and the elements of the targeted regions following pre-defined rules [Loh92]. Special treatment is required for hybrid grids (including hexahedral, prismatic, tetrahedral and pyramidal elements), due to the variety of element types and division ways of them [Kal05a]. Despite it is assumed a relatively efficient process, in some studies such as this of Kavouklis and Kallinderis [Kav10] its parallel implementation has been proposed in order the required wall-clock time to be further decreased. Moreover, the aforementioned methodology can be coupled with p-refinement or de-refinement approach [Kal05b]; the latter is employed in an analogous to h-refinement manner at regions where the solution doesn't vary significantly [Kal05b].

1.3. Present study

1.3.1 Overview

According to the main objective of this thesis, methodologies to support the numerical simulation of compressible fluid flow and radiative heat transfer were developed. For the representation of the examined computational domains, three-dimensional unstructured tetrahedral grids are employed. However, if a viscous flow is simulated, prisms and pyramids are also included (hybrid grids) to resolve effectively the severe anisotropy of the flow at the boundary layer region; for radiative heat transfer problems, highly stretched elements are utilized at the areas of boundary surfaces in case increased accuracy is required [Lyg12b, Lyg14a, Lyg15]. The discretization and the corresponding reformulation of governing PDE's (RANS, turbulence model equations and RTE) is implemented with a node-centered finite-volume scheme [Lyg13c, Lyg14a, Lyg15].

For the computation of the RANS PDE's inviscid fluxes, an upwind scheme is employed, assuming a one-dimensional Riemann problem at each face of a finite control-volume, while its solution is succeeded using the Roe's approximate Riemann solver. The pre-mentioned scheme is enhanced by a higher-order accurate spatial reconstruction method, based on the MUSCL technique, along with appropriate slope limiters, namely Van Albada-Van Leer, Barth-Jespersen and Min-mod. These limiting functions are required to control the variation of reconstructed values and maintain monotonicity between the states at the interface of each two adjacent control volumes. For viscous fluxes, the same discretization is utilized, while the necessary velocity and temperature gradients are calculated implementing either an element-based method, which considers the construction of edge-dual volumes, or a nodal-averaging one. For turbulent flows, three two-equation turbulence models, namely $k-\epsilon$ (in three versions), $k-\omega$ and SST , have been incorporated; no additional model is included to allow for transition modelling [Lyg11]. For time

integration and iterative approximation of the flow and turbulence equations, either an explicit scheme, applying a second-order temporal accurate four-stage Runge-Kutta (RK(4)) method, or an implicit one, employing the Jacobi or the Gauss-Seidel algorithm, is implemented [Lyg11].

In a similar way, a node-centered finite-volume discretized, time- or non-time-dependent RTE is employed to allow for the prediction of radiative heat transfer in general enclosures through absorbing, emitting, and either isotropically or anisotropically scattering gray media; the considered forward scattering ratio (F2) is based on the Legendre polynomial expansions [Lyg12a, Lyg12b]. The effects of overhang problem, caused by the combination of unstructured grids and the division of angular domain in finite solid angles, are alleviated with either the bold approximation or the pixelation method. The methodology is enhanced with a second-order accurate spatial scheme, based on the MUSCL reconstruction of the radiative intensity values, while Van Albada-Van Leer or Min-mod slope limiter is used to bound the reconstructed values, especially at the boundary surfaces' regions [Lyg13b, Lyg13c]. The final steady-state solution is achieved by either iterative approximations (steady RTE, non time-dependent) [Lyg12a, Lyg12b] or a second-order temporal accurate four-stage Runge-Kutta (RK(4)) method (time-dependent RTE) [Lyg13b, Lyg13c].

In order to increase accuracy and efficiency of the proposed methodology, additional enhancing schemes have been incorporated to both fluid flow and radiative heat transfer solvers. An edge-based data structure along with the local-time stepping technique, suitably defined for each solver, is used for the computation of numerical fluxes, resulting in reduced memory requirements and computational effort [Lyg13c, Lyg14a, Lyg15]. To alleviate increased computational requirements especially in large-scale simulations a parallelization strategy was developed, based on the domain decomposition approach and the MPI library [Lyg12b, Lyg14a, Lyg15]. In addition, a spatial agglomeration multigrid methodology is incorporated, resulting in more acceleration of the solution procedure; it considers the implementation of FAS (included in FMG) on successively coarser spatial resolutions, derived from the initial finest grid through the fusion of adjacent control volumes in a way similar to advancing front technique [Lyg14b, Lyg14c, Lyg14d, Lyg14f]. In case of radiative heat transfer simulations, the aforementioned method is extended to an angular version, which considers coarsening of the angular resolution by the fusion of neighbouring solid control angles, along with the FAS employment in a corresponding to spatial scheme way. Furthermore, a combined (nested) spatial/angular agglomeration multigrid method is introduced, according to which a complete angular FAS cycle is accomplished at each level of the spatial multigrid scheme [Lyg14b, Lyg14f, Lyg14g]. Finally, the h-refinement technique is incorporated to enrich the desired regions (either automatically selected or user defined) of unstructured, tetrahedral or hybrid grids and consequently increase accuracy of the final steady-state solution at the same areas; it can be performed during the solution procedure, avoiding the generation of a new mesh from scratch [Lyg13b, Lyg13c].

Based on the aforementioned methods an academic CFD code, named *Galatea* (one of the fifty goddess-nymphs of the sea, the *Nereides*), was developed; it has been validated against three- and quasi-three-dimensional benchmark test cases, while its results have been compared qualitatively and quantitatively with wind tunnel experimental data as well as with those obtained by acclaimed corresponding solvers. The solutions results confirm that the proposed methodology/code is capable at effectively simulating compressible fluid flows and radiative heat transfer in terms of accuracy, geometric flexibility and computational efficiency [Lyg11, Lyg12b, Lyg13c, Lyg14a, Lyg14e, Lyg14g, Lyg15].

1.3.2 Contributions

Considering similar studies available in the open literature, the following original contributions have been introduced in this thesis:

- A combination of a parallel node-centered finite-volume method with three-dimensional unstructured hybrid grids for radiative heat transfer prediction [Lyg12b].

- A second-order accurate spatial scheme jointed or not with the slope limiter of Van Albada-Van Leer or Min-mod, improving accuracy of radiative heat transfer finite-volume solutions, especially at boundary surfaces' regions with large radiative intensity gradients [Lyg13c].
- Implicit treatment of diffusively and specularly (mirroring) reflecting surfaces' boundary conditions [Lyg13c].
- A grid adaptation method, employing h-refinement technique to improve the accuracy of radiative heat transfer solution at specific regions of the examined mesh [Lyg13c].
- A parallel spatial agglomeration multigrid method enhancing both flow and radiative heat transfer algorithms with improved computational performance [Lyg14b, Lyg14c, Lyg14d, Lyg14f, Lyg14g].
- A parallel angular agglomeration multigrid scheme for radiative heat transfer simulations [Lyg14b, Lyg14f, Lyg14g].
- An extension of previous methods in a combined (nested) spatial/angular agglomeration multigrid scheme for radiation problems [Lyg14b, Lyg14f, Lyg14g].

The aforementioned contributions have been published in sixteen Journal and Conferences' papers, listed below.

1.3.3 Outline

The structure of this thesis is as follows: Chapter 2 describes the fluid flow model, including governing equations, discretization method, computation of numerical fluxes and iterative solution, while in Chapter 3 the corresponding topics are analyzed for the radiative heat transfer model. In Chapter 4 the implemented parallelization strategy is summarized, while Chapter 5 includes the spatial, the angular and the combined (nested) spatial/angular agglomeration multigrid scheme, developed to improve the computational performance of the proposed (flow and radiative heat transfer) methodology. Chapter 6 contains the employed automatic mesh enrichment method (h-refinement), enhancing the algorithm with the capability of increasing solution's accuracy at specific locations, defined either automatically depending on selected criteria, or by the user. Finally, in Chapters 7 and 8 the validation of the proposed method against benchmark test cases is analyzed, including qualitative and quantitative comparisons of the obtained results with the available experimental data as well as other solvers, while Chapter 9 contains some conclusions and information on ongoing and future work.

1.4. List of publications

A. International journals

1. G.N. Lygidakis and I.K. Nikolos, Numerical Analysis of Flow over the NASA Common Research Model Using the Academic Computational Fluid Dynamics Code *Galatea*, *ASME Journal of Fluids Engineering*, vol. 137, FE-14-1602, 2015, DOI: 10.1115/1.4029730.
2. G.N. Lygidakis and I.K. Nikolos, Using a Parallel Spatial/Angular Agglomeration Multigrid Scheme to Accelerate the FVM Radiative Heat Transfer Computation - Part I: Methodology, *Numerical Heat Transfer Part B: Fundamentals*, vol. 66, pp. 471-497, 2014.
3. G.N. Lygidakis and I.K. Nikolos, Using a Parallel Spatial/Angular Agglomeration Multigrid Scheme to Accelerate the FVM Radiative Heat Transfer Computation - Part II: Numerical Results, *Numerical Heat Transfer Part B: Fundamentals*, vol. 66, pp. 498-525, 2014.

4. G.N. Lygidakis and I.K. Nikolos, Using a High-Order Spatial/Temporal Scheme and Grid Adaptation with a Finite-Volume Method for Radiative Heat Transfer, *Numerical Heat Transfer Part B: Fundamentals*, vol. 64, pp. 89-117, 2013.
5. G.N. Lygidakis and I.K. Nikolos, Using the Finite-Volume Method and Hybrid Unstructured Meshes to Compute Radiative Heat Transfer in 3-D Geometries, *Numerical Heat Transfer Part B: Fundamentals*, vol. 62, pp. 289-314, 2012.

B. Proceeding of international conferences

1. S.S. Sarakinos, G.N. Lygidakis and I.K. Nikolos, Acceleration Strategies for Simulating Compressible and Incompressible Flows, *Proceedings of the 4th International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering, PARENG2015*, Dubrovnik, Croatia, 24-27 March 2015.
2. G.N. Lygidakis and I.K. Nikolos, Using the DLR-F6 Aircraft Model for the Evaluation of the Academic CFD Code "Galatea", *Proceedings of the International Mechanical Engineering Congress and Exposition, ASME-IMECE2014*, Montreal, Canada, 14-20 November 2014, IMECE2014-39756.
3. S.S. Sarakinos, G.N. Lygidakis and I.K. Nikolos, Evaluation of a Parallel Agglomeration Multigrid Finite-Volume Algorithm, named Galatea-I, for the Simulation of Incompressible Flows on 3D Hybrid Unstructured Grids, *Proceedings of the International Mechanical Engineering Congress and Exposition, ASME-IMECE2014*, Montreal, Canada, 14-20 November 2014, IMECE2014-39759.
4. G.N. Lygidakis, S.S. Sarakinos and I.K. Nikolos, A Parallel Agglomeration Multigrid Method Incompressible Flow Simulations, *Proceedings of the 9th International Conference on Engineering Computational Technology, ECT 2014*, Naples, Italy, 2-5 September 2014.
5. G.N. Lygidakis and I.K. Nikolos, A Parallel Agglomeration Multigrid Method for the Acceleration of Compressible Flow Computations on 3D Hybrid Unstructured Grids, *Proceedings of the 11th World Congress on Computational Mechanics (WCCM XI), 5th European Conference on Computational Mechanics (ECCM V), 6th European Conference on Computational Fluid Dynamics (ECFD VI), IACM-ECCOMAS*, Barcelona, Spain, 20-25 July 2014, pp. 4852-4863.
6. G.N. Lygidakis and I.K. Nikolos, A Parallel Finite-Volume Spatial/Angular Agglomeration Multigrid Method for Radiative Heat Transfer Computation, *Proceedings of the HEAT TRANSFER 2014, 13th International Conference on Simulation and Experiments in Heat Transfer and its Applications (WIT (Wessex Institute of Technology) Transactions on Engineering Sciences)*, A Coruna, Spain, 2-4 July 2014, vol. 83, pp. 275-286.
7. G.N. Lygidakis and I.K. Nikolos, Assessment of the Academic CFD Code "Galatea" Using the NASA Common Research Model (CRM), *Proceedings of the 12th Biennial Conference on Engineering Systems Design and Analysis, ASME-ESDA2014*, Copenhagen, Denmark, 25-27 June 2014, ESDA2014-20265.
8. G.N. Lygidakis and I.K. Nikolos, Improving the Accuracy of a Finite-Volume Method for Computing Radiative Heat Transfer in Three-Dimensional Unstructured Meshes, *Proceedings of the ECCOMAS Special Interest Conference - SEECCM 2013: 3rd South-East European*

- Conference on Computational Mechanics - An IACM Special Interest Conference 2013*, Kos, Greece, 12-14 June 2013, pp. 599-620.
9. G.N. Lygidakis and I.K. Nikolos, Evaluating a Parallel Node-Centered Finite-Volume Algorithm, named Galatea, in simulation of 3D Compressible Flows, *Proceedings of the 10th HSTAM (Hellenic Society for Theoretical and Applied Mechanics) International Congress on Mechanics*, Chania, Crete, Greece, 25-27 May 2013.
 10. G.N. Lygidakis and I.K. Nikolos, A Parallelized Node-Centered Finite Volume Method for Computing Radiative Heat Transfer on 3D Unstructured Hybrid Grids, *Proceedings of the 11th Biennial Conference on Engineering Systems Design and Analysis, ASME-ESDA2012*, Nantes, France, 2-4 July 2012, vol.1, pp. 177-186, ESDA2012-82331.
 11. G.N. Lygidakis and I.K. Nikolos, An Unstructured Node-Centered Finite Volume Method for Computing 3D Viscous Compressible Flows on Hybrid Grids, *Proceedings of the 7th GRACM (Greek Association of Computational Mechanics) International Congress on Computational Mechanics*, Athens, 30 June - 2 July 2011.

2. Fluid flow

2.1. Governing equations

2.1.1. The Navier-Stokes equations

A compressible viscous flow is described by the Navier-Stokes equations (named of the French mathematician Claude Louis Marie Henri Navier and the English engineer Sir George Gabriel Stokes), which are derived by the conservation laws of mass, momentum and energy [Bla01]. In three dimensions, arranged into convective (inviscid), diffusive (viscous) and source terms, they can be written in differential form as:

$$\frac{\partial \vec{W}}{\partial t} + \frac{\partial \vec{F}^{inv}}{\partial x} + \frac{\partial \vec{G}^{inv}}{\partial y} + \frac{\partial \vec{J}^{inv}}{\partial z} - \frac{\partial \vec{F}^{vis}}{\partial x} - \frac{\partial \vec{G}^{vis}}{\partial y} - \frac{\partial \vec{J}^{vis}}{\partial z} = \vec{S} \quad (2.1)$$

The conservative variables' vector $\vec{W} = (\rho, \rho u, \rho v, \rho w, \rho E)^T$, the inviscid flux vectors $\vec{F}^{inv}, \vec{G}^{inv}, \vec{J}^{inv}$, the viscous flux vectors $\vec{F}^{vis}, \vec{G}^{vis}, \vec{J}^{vis}$ and the vector of the source term \vec{S} are expressed in terms of the primitive variables (ρ, u, v, w, p) . The source term is set to zero in this work, while the inviscid and viscous vectors are defined as [Koo00, Lyg14e]:

$$\vec{F}^{inv} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (\rho E + p)u \end{pmatrix}, \quad \vec{G}^{inv} = \begin{pmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vw \\ (\rho E + p)v \end{pmatrix}, \quad \vec{J}^{inv} = \begin{pmatrix} \rho w \\ \rho wu \\ \rho wv \\ \rho w^2 + p \\ (\rho E + p)w \end{pmatrix} \quad (2.2)$$

$$\vec{F}^{vis} = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + q_x \end{pmatrix}, \quad \vec{G}^{vis} = \begin{pmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ u\tau_{yx} + v\tau_{yy} + w\tau_{yz} + q_y \end{pmatrix}, \quad \vec{J}^{vis} = \begin{pmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ u\tau_{zx} + v\tau_{zy} + w\tau_{zz} + q_z \end{pmatrix} \quad (2.3)$$

For an inviscid flow only the corresponding flux vectors $\vec{F}^{inv}, \vec{G}^{inv}, \vec{J}^{inv}$ are considered, leading in the so-called Euler equations (named of the Swiss mathematician and physicist Leonhard Euler). On the other hand, for a viscous flow the diffusive flux vectors $\vec{F}^{vis}, \vec{G}^{vis}, \vec{J}^{vis}$ have to be taken into account, based on the components of the stress tensor $(\tau_{xx}, \tau_{xy}, \tau_{xz}, \tau_{yx}, \tau_{yy}, \tau_{yz})$, which are defined for Newtonian fluids (shear stresses proportional to the velocity gradients) as [Hir90]

$$\tau_{ij} = \mu \left[\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} (\nabla \cdot \vec{V}) \delta_{ij} \right] \quad (2.4)$$

where μ is the dynamic viscosity coefficient, which for a perfect gas depends strongly on the temperature but only weakly on the pressure [Bla01]. Considering this state, it can be computed from the local temperature of the fluid (in K) via the Sutherland formula as [Luo05]

$$\mu = \frac{c_1 T^{3/2}}{T + c_2} \quad (2.5)$$

where the coefficients c_1 and c_2 are equal to $1.458E-6 \text{ kg m}^{-1} \text{ sec}^{-1} \text{ K}^{-1/2}$ and 110.4 K respectively, e.g., the obtained dynamic viscosity for air at 300 K equals to $1.846E-5 \text{ kg m}^{-1} \text{ sec}^{-1}$. Using another formulation of the same equation, which is based on the reference dynamic viscosity μ_{ref} and the reference temperature T_{ref} (usually the ones in the far field) it can be computed as [Luo05]:

$$\mu = \mu_{ref} \left(\frac{T}{T_{ref}} \right)^{3/2} \frac{T_{ref} + c_2}{T + c_2} \quad (2.6)$$

The three-dimensional Navier-Stokes equations are expressed by a set of five equations with five conservative unknown variables ($\rho, \rho u, \rho v, \rho w, \rho E$); nevertheless, they contain seven unknown flow field variables, namely (ρ, u, v, w, E, p, T), necessitating two more equations. Considering that in pure aerodynamics, the fluid can quite reasonably be assumed to behave like a perfect gas, the perfect gas state equation is included in the equation set [Lan98]

$$p = \rho R_g T \quad (2.7)$$

where the gas constant R_g equals to $287.04 \text{ m}^2 \text{ sec}^{-2} \text{ K}^{-1}$ and it is associated with the constant pressure and volume specific heat coefficients with the following equations

$$R_g = c_p - c_v, \quad \gamma = c_p / c_v \quad (2.8)$$

while these coefficients are defined as follows

$$h = c_p T, \quad e = c_v T \quad (2.9)$$

where h and e are the enthalpy and internal energy of the gas per unit mass. The specific heat coefficients are assumed constants, although for different gases they take different values; for air the constant pressure specific heat coefficient c_p equals to $1004.64 \text{ m}^2 \text{ sec}^{-2} \text{ K}^{-1}$, the constant volume specific heat coefficient c_v equals to $717.6 \text{ m}^2 \text{ sec}^{-2} \text{ K}^{-1}$ and the dimensionless coefficient γ equals to 1.4 [Lan98]. The last expression completing the equation set, associates the pressure p with the total energy per unit volume ρE [Bla01]

$$\begin{aligned} \rho E &= \rho e + \frac{1}{2} \rho (u^2 + v^2 + w^2) = \rho T c_v + \frac{1}{2} \rho (u^2 + v^2 + w^2) = \\ &= \frac{p}{R_g} c_v + \frac{1}{2} \rho (u^2 + v^2 + w^2) = \frac{p}{(\gamma - 1)} + \frac{1}{2} \rho (u^2 + v^2 + w^2) \end{aligned} \quad (2.10)$$

where ρe is the internal energy per unit volume. The corresponding specific total enthalpy h_t then is associated with the pressure p and the total energy per unit volume ρE as:

$$h_t = \frac{\rho E + p}{\rho} = \frac{\gamma p}{\rho(\gamma - 1)} + \frac{1}{2} (u^2 + v^2 + w^2) \quad (2.11)$$

The heat flux vector (q_x, q_y, q_z) in the energy equation is defined accordingly to the stress tensor as [Bla01]

$$q_i = \chi \nabla T, \quad \chi = \frac{\mu c_p}{Pr} \quad (2.12)$$

where the conductivity coefficient χ depends on the dimensionless Prandtl number Pr (named in honor of the German professor Ludwig Prandtl), and is commonly considered constant in the entire flow field and equal to 0.72 for perfect gases.

Following the common strategy in CFD as well as in experimental fluid mechanics, the governing equations are expressed in dimensionless form; the normalization of the variables is performed utilizing a characteristic length L_{ref} , the free-stream velocity V_{ref} , the free-stream density ρ_{ref} , the free-stream dynamic viscosity μ_{ref} and the constant volume specific heat coefficient c_v as:

$$\tilde{x}_i = \frac{x_i}{L_{ref}}, \quad \tilde{u}_i = \frac{u_i}{V_{ref}}, \quad \tilde{\rho} = \frac{\rho}{\rho_{ref}}, \quad \tilde{\mu} = \frac{\mu}{\mu_{ref}}, \quad \tilde{R}_g = \frac{R_g}{c_v} = \gamma - 1 \quad (2.13)$$

Considering the previous normalizations, the rest of the variables, included in the equations (2.1)-(2.3) and (2.10)-(2.11), are expressed as

$$\tilde{p} = \frac{p}{\rho_{ref} V_{ref}^2}, \quad \tilde{\rho} E = \frac{\rho E}{\rho_{ref} V_{ref}^2}, \quad \tilde{h}_t = \frac{h_t}{V_{ref}^2}, \quad \tilde{T} = \frac{T}{V_{ref}^2 / c_v}, \quad \tilde{t} = \frac{t}{L_{ref} / V_{ref}} \quad (2.14)$$

while taking into account the dimensionless Reynolds number (named of the British engineer Osborne Reynolds), defined as [Mun98]

$$Re = \frac{\rho_{ref} V_{ref} L_{ref}}{\mu_{ref}} \quad (2.15)$$

the dynamic viscosity μ in the equations for the stress tensor and heat flux vector, is substituted by the term $\tilde{\mu}/Re$:

$$\tilde{\tau}_{ij} = \frac{\tilde{\mu}}{Re} \left[\left(\frac{\partial \tilde{u}_i}{\partial \tilde{x}_j} + \frac{\partial \tilde{u}_j}{\partial \tilde{x}_i} \right) - \frac{2}{3} (\nabla \cdot \tilde{\vec{V}}) \delta_{ij} \right] \quad (2.16)$$

$$\tilde{q}_i = \frac{\tilde{\mu} \gamma}{Re Pr} \nabla \tilde{T} \quad (2.17)$$

Additionally, the constant pressure and the constant volume specific heat coefficients are normalized ($\tilde{c}_p = \gamma$ and $\tilde{c}_v = 1$), while the perfect gas equation is transformed as:

$$p = \rho R_g T \Rightarrow \tilde{p} \rho_{ref} V_{ref}^2 = \tilde{\rho} \rho_{ref} \tilde{R}_g \tilde{T} (V_{ref}^2 / c_v) \Rightarrow \tilde{p} = \tilde{\rho} \tilde{R}_g \tilde{T} \Rightarrow \tilde{p} = \tilde{\rho} (\gamma - 1) \tilde{T} \quad (2.18)$$

As far as the dimensionless dynamic viscosity is concerned, it can be assumed quite safely to be equal to unity in compressible flow simulations as these encountered in this work or preferably evaluated via the Sutherland law using the dimensionless temperature as [Mar07]

$$\begin{aligned}\tilde{\mu} &= \left(\frac{T}{T_{ref}} \right)^{3/2} \frac{T_{ref} + c_2}{T + c_2} = \left(\frac{\tilde{T}(V_{ref}^2/c_v)}{\tilde{T}_{ref}(V_{ref}^2/c_v)} \right)^{3/2} \frac{\tilde{T}_{ref}(V_{ref}^2/c_v) + c_2}{\tilde{T}(V_{ref}^2/c_v) + c_2} = \\ &= \left(\frac{\tilde{T}}{\tilde{T}_{ref}} \right)^{3/2} \frac{\tilde{T}_{ref} + \frac{c_2}{(V_{ref}^2/c_v)}}{\tilde{T} + \frac{c_2}{(V_{ref}^2/c_v)}} = \left(\frac{\tilde{T}}{\tilde{T}_{ref}} \right)^{3/2} \frac{\tilde{T}_{ref} + \frac{c_2 \tilde{T}_{ref}}{T_{ref}}}{\tilde{T} + \frac{c_2 \tilde{T}_{ref}}{T_{ref}}}\end{aligned}\quad (2.19)$$

where \tilde{T} is the dimensionless local temperature, while T_{ref} and \tilde{T}_{ref} are the dimensional and the dimensionless reference temperature (usually the one in the far field). Finally, two more expressions used in this work, concern the computation of the local speed of sound at a node P [Lan98]

$$\tilde{c}_P = \sqrt{\gamma \tilde{R}_g \tilde{T}_P} = \sqrt{\gamma(\gamma-1) \tilde{T}_P} = \sqrt{\frac{\gamma \tilde{p}_P}{\tilde{\rho}_P}} \quad (2.20)$$

and the computation of corresponding Mach number [Mun98]:

$$M_P = \frac{\sqrt{\tilde{u}_P^2 + \tilde{v}_P^2 + \tilde{w}_P^2}}{\tilde{c}_P} \quad (2.21)$$

For simplification reasons the superscript "~", denoting the normalized variables, will be neglected at the next sections.

2.1.2. The Reynolds Averaged Navier-Stokes (RANS) equations

Although the Navier-Stokes equations can account theoretically for turbulent flows (DNS-Direct Numerical Simulation), their utilization is limited only to relatively simple flow problems at low Reynolds numbers, due to their requirements for excessive spatial resolution and computation time. Instead of them, approximation methods are preferred, such as the Large Eddy Simulation (LES) or the Reynolds Averaged Navier Stokes (RANS) equations, the latter implemented in this work. The first modelling approach was introduced by Osborne Reynolds in 1895, in accordance with which the flow variables are decomposed into a mean and a fluctuating part, e.g., a flow variable U_i is analyzed as follows

$$U_i = \bar{U}_i + U'_i, \quad \bar{U}'_i = 0 \quad (2.22)$$

where the turbulent fluctuation is described by a prime and the mean value by an overbar, the latter defined as:

$$\bar{U}_i = \lim_{T \rightarrow \infty} \frac{1}{T} \int_t^{t+T} U_i dt \quad (2.23)$$

Nevertheless, the previous equations can be implemented only in incompressible flows or in flows in which the value of density can be assumed constant [Bla01]. Thus, for compressible flows the density (mass) or the Favre averaging method is applied to certain quantities (velocity, enthalpy, energy), while for the rest variables (density and pressure) the Reynolds averaging approach is maintained [Bla01]. As such the previous equations are transformed for the velocity, enthalpy and energy as:

$$U_i = \tilde{U}_i + U_i'', \quad \widetilde{\rho U_i''} = 0 \quad (2.24)$$

$$\tilde{U}_i = \frac{1}{\bar{\rho}} \lim_{T \rightarrow \infty} \frac{1}{T} \int_t^{t+T} \rho U_i dt \quad (2.25)$$

Furthermore, in the governing equations the turbulent kinetic energy k has to be considered, which is defined, using the velocity fluctuations, as follows:

$$\bar{\rho} \tilde{k} = \frac{1}{2} \widetilde{\rho u_i'' u_i''} \quad (2.26)$$

With the above approaches the Favre and Reynolds Averaged Navier-Stokes (RANS) equations are derived in the following formulation (implemented in this study) [Lyg14e]

$$\frac{\partial \bar{W}}{\partial t} + \frac{\partial \bar{F}^{inv}}{\partial x} + \frac{\partial \bar{G}^{inv}}{\partial y} + \frac{\partial \bar{J}^{inv}}{\partial z} - \frac{\partial \bar{F}^{vis}}{\partial x} - \frac{\partial \bar{G}^{vis}}{\partial y} - \frac{\partial \bar{J}^{vis}}{\partial z} = \bar{0} \quad (2.27)$$

$$\bar{F}^{inv} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (\rho E + p)u \end{pmatrix}, \quad \bar{G}^{inv} = \begin{pmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vw \\ (\rho E + p)v \end{pmatrix}, \quad \bar{J}^{inv} = \begin{pmatrix} \rho w \\ \rho wu \\ \rho wv \\ \rho w^2 + p \\ (\rho E + p)w \end{pmatrix} \quad (2.28)$$

$$\bar{F}^{vis} = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + q_x + E_{tx} \end{pmatrix}, \quad \bar{G}^{vis} = \begin{pmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ u\tau_{yx} + v\tau_{yy} + w\tau_{yz} + q_y + E_{ty} \end{pmatrix}, \quad (2.29)$$

$$\bar{J}^{vis} = \begin{pmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ u\tau_{zx} + v\tau_{zy} + w\tau_{zz} + q_z + E_{tz} \end{pmatrix}$$

where the mean total energy per unit volume ρE and the corresponding mean specific total enthalpy h_t , which include the contribution of the turbulent kinetic energy k , can be evaluated as [Koo00]:

$$\rho E = \frac{p}{(\gamma-1)} + \frac{1}{2} \rho (u^2 + v^2 + w^2) + \rho k \quad (2.30)$$

$$h_t = \frac{\rho E + p}{\rho} = \frac{\gamma p}{\rho(\gamma-1)} + \frac{1}{2} (u^2 + v^2 + w^2) + k \quad (2.31)$$

Similarly, the viscous stress tensor is extended in the momentum and energy equation by the corresponding Favre averaged turbulent stress tensor τ'_{ij} [Bla01]:

$$\tau'_{ij} = -\bar{\rho} \widetilde{u'_i u'_j} \quad (2.32)$$

Based on the Boussinesq hypothesis (named in honor of the French mathematician and physicist Joseph Valentin Boussinesq), which assumes a linear relationship between the turbulent shear stress and the mean rate strain, the total averaged stress tensor τ_{ij} , divided in the laminar part denoted by the superscript l and the turbulent part denoted by the superscript t , is defined as [Koo00]:

$$\begin{aligned} \tau_{ij} = \tau_{ij}^l + \tau_{ij}^t = & \left[\frac{\mu}{\text{Re}} \left[\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} (\nabla \cdot \vec{V}) \delta_{ij} \right] \right] + \left[\mu_t \left[\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} (\nabla \cdot \vec{V}) \delta_{ij} \right] - \frac{2}{3} \rho k \delta_{ij} \right] = \\ & \left(\frac{\mu}{\text{Re}} + \mu_t \right) \left[\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} (\nabla \cdot \vec{V}) \delta_{ij} \right] - \frac{2}{3} \rho k \delta_{ij} \end{aligned} \quad (2.33)$$

The total averaged thermal vector q_i is defined accordingly to the total averaged stress tensor as [Lyg14e]

$$q_i = q_i^l + q_i^t = \frac{\mu \gamma}{\text{RePr}} \nabla T + \frac{\mu_t \gamma}{\text{Pr}_t} \nabla T \quad (2.34)$$

where Pr_t is the turbulent Prandtl number, which is in general assumed to be constant in the entire flow field and equal to 0.9 for perfect gases or air. In case of the k - ε turbulence model we define term E_{tx_i} , which corresponds to the partial derivative of the turbulent kinetic energy k , as

$$E_{tx_i} = \left(\frac{\mu}{\text{Re}} + \frac{\mu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_i} \quad (2.35)$$

while in case of the k - ω and SST turbulence model as:

$$E_{tx_i} = \left(\frac{\mu}{\text{Re}} + \mu_t \sigma_k \right) \frac{\partial k}{\partial x_i} \quad (2.36)$$

The term μ_t is the turbulent dynamic viscosity, while σ_k is a constant; for their evaluation as well as for the computation of the turbulent kinetic energy k , appropriate turbulence models have to be employed along with the RANS equations.

2.1.3. Turbulence models

Three two-equation turbulence models have been incorporated in the present algorithm, namely k - ε , k - ω and SST , to predict the features of high Reynolds flows. The interaction between the turbulence model PDE's and the Navier-Stokes ones is obtained mainly via the turbulent dynamic viscosity. No additional model was included to simulate transition.

2.1.3.1. Turbulence model k - ε

The turbulence model k - ε , which is one of the most widely applied two-equation eddy-viscosity models, is based on the solution of PDE's for the turbulent kinetic energy k and the turbulent dissipation rate ε [Lau74]. Its dimensionless differential formulation is similar to this of the flow model with the exception of the non-zero source term:

$$\frac{\partial \vec{W}_{k\varepsilon}}{\partial t} + \frac{\partial \vec{F}_{k\varepsilon}^{inv}}{\partial x} + \frac{\partial \vec{G}_{k\varepsilon}^{inv}}{\partial y} + \frac{\partial \vec{J}_{k\varepsilon}^{inv}}{\partial z} - \frac{\partial \vec{F}_{k\varepsilon}^{vis}}{\partial x} - \frac{\partial \vec{G}_{k\varepsilon}^{vis}}{\partial y} - \frac{\partial \vec{J}_{k\varepsilon}^{vis}}{\partial z} = \vec{S}_{k\varepsilon} \quad (2.37)$$

The $\vec{W}_{k\varepsilon} = (\rho k, \rho \varepsilon)^T$ is the conservative variables vector, while the advective (inv), the diffusive (vis), and the source term vectors are defined as

$$\vec{F}_{k\varepsilon}^{inv} = \begin{pmatrix} \rho u k \\ \rho u \varepsilon \end{pmatrix}, \vec{G}_{k\varepsilon}^{inv} = \begin{pmatrix} \rho v k \\ \rho v \varepsilon \end{pmatrix}, \vec{J}_{k\varepsilon}^{inv} = \begin{pmatrix} \rho w k \\ \rho w \varepsilon \end{pmatrix} \quad (2.38)$$

$$\vec{F}_{k\varepsilon}^{vis} = \begin{pmatrix} \left(\frac{\mu}{\text{Re}} + \frac{\mu_t}{\sigma_k} \right) \frac{\partial k}{\partial x} \\ \left(\frac{\mu}{\text{Re}} + \frac{\mu_t}{\sigma_\varepsilon} \right) \frac{\partial \varepsilon}{\partial x} \end{pmatrix}, \vec{G}_{k\varepsilon}^{vis} = \begin{pmatrix} \left(\frac{\mu}{\text{Re}} + \frac{\mu_t}{\sigma_k} \right) \frac{\partial k}{\partial y} \\ \left(\frac{\mu}{\text{Re}} + \frac{\mu_t}{\sigma_\varepsilon} \right) \frac{\partial \varepsilon}{\partial y} \end{pmatrix}, \vec{J}_{k\varepsilon}^{vis} = \begin{pmatrix} \left(\frac{\mu}{\text{Re}} + \frac{\mu_t}{\sigma_k} \right) \frac{\partial k}{\partial z} \\ \left(\frac{\mu}{\text{Re}} + \frac{\mu_t}{\sigma_\varepsilon} \right) \frac{\partial \varepsilon}{\partial z} \end{pmatrix} \quad (2.39)$$

$$\vec{S}_{k\varepsilon} = \begin{pmatrix} P_k - \rho \varepsilon + S_k^{LR} \\ c_1 f_1 P_k \frac{\varepsilon}{k} - c_2 f_2 \rho \frac{\varepsilon^2}{k} + S_\varepsilon^{LR} \end{pmatrix} \quad (2.40)$$

where the terms σ_k , σ_ε , c_1 , c_2 , f_1 and f_2 in the source term vector are constants with values $\sigma_k=1$, $\sigma_\varepsilon=1.3$, $c_1=1.44$, $c_2=1.92$, $f_1=1$ and $f_2=1$, while the terms S_k^{LR} and S_ε^{LR} are assumed equal to zero for high Reynolds number flows (as those examined in our study) [Koo00]. The turbulent viscosity μ_t is evaluated as [Koo00, Kou03, ANSYS06]

$$\mu_t = f_\mu c_\mu \frac{\rho k^2}{\varepsilon} \quad (2.41)$$

with the constants c_μ and f_μ being equal to 0.09 and 1 respectively, while the turbulent energy production term P_k is computed as [Koo00]:

$$\begin{aligned}
 P_k = & -\frac{2}{3}\rho k \nabla \cdot \vec{c} + \tau_{ij}^t \frac{\partial u_i}{\partial x_j} \\
 & -\frac{2}{3}\rho k \nabla \cdot \vec{c} - \frac{2}{3}\mu_t (\nabla \cdot \vec{c})^2 + \mu_t \left[\left(\frac{\partial u_i}{\partial x_j} \right)^2 + \frac{\partial u_i}{\partial x_j} \frac{\partial u_j}{\partial x_i} \right] = \\
 & -\frac{2}{3}\rho k \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) - \frac{2}{3}\mu_t \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right)^2 + \\
 & + \mu_t \left[2 \left(\frac{\partial u}{\partial x} \right)^2 + 2 \left(\frac{\partial v}{\partial y} \right)^2 + 2 \left(\frac{\partial w}{\partial z} \right)^2 + \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right)^2 \right]
 \end{aligned} \tag{2.42}$$

The second term $(-(2/3)\mu_t (\nabla \cdot \vec{c})^2)$ of the turbulent energy production term P_k is decreased by some researchers to $(-2\mu_t (\nabla \cdot \vec{c})^2)$, in order to prevent the values of k and ε of becoming too large through shocks [ANSYS06]; this situation becomes progressively worse as the mesh is refined at shocks' regions.

For boundary layer modelling appropriate boundary conditions are implemented at the solid wall region; two methods were incorporated in the present study, namely wall functions and an one-equation model. According to the first methodology, suitable functions are implemented instead of the governing equations for the computation of the kinetic energy k and the turbulent dissipation rate ε of the first layer nodes next to the solid boundaries. Considering the common practice of using highly stretched grids at this area, the aforementioned functions can be employed to the boundary nodes instead; in that case free-slip boundary conditions are required instead of no-slip ones. Although this methodology is based on simplifications of boundary layer modelling, it appears to be quite popular, due to its capability to derive sufficiently accurate results even in coarse grids. The aforementioned functions are described as follows [Hir90, Mav94, Koo00, Kou03]

$$k = \frac{u_\tau^2}{C_\mu^{0.5}}, \quad \varepsilon = \frac{u_\tau^3}{\kappa y} \tag{2.43}$$

where u_τ is the friction velocity and y is the distance of the boundary node from the next internal one. As the wall functions are based on the theoretical division of the boundary layer in two discrete areas, namely the linear (laminar) and the logarithmic one, u_τ is defined as [Kou03]

$$u^+ = \frac{V_t}{u_\tau} = \begin{cases} y^+ & y^+ < 11.6 \\ \frac{1}{\kappa} \ln y^+ + B & y^+ \geq 11.6 \end{cases} \tag{2.44}$$

where the terms κ and B are constants (0.41 and 5.5 respectively), while u^+ is the dimensionless velocity and V_t is the tangential component of velocity. y^+ is the dimensionless distance of the boundary node from the next internal one, evaluated as [Hir90, Koo00, Kou03]:

$$y^+ = \frac{\rho u_\tau y}{\mu} \text{Re} \tag{2.45}$$

A value of y^+ between 20 and 200 has been identified by many researchers [Kou03], as appropriate to capture sufficiently the features of the logarithmic area of the boundary layer; thus, the selection of this

turbulence model has to be considered during the mesh generation procedure. The dimensionless distance y^+ and the friction velocity u_τ are evaluated through an iterative procedure (Newton-Raphson) utilizing equations (2.44) and (2.45), while then the friction stress tensor $\tau_w = \rho u_\tau^2$ is computed to contribute to the viscous terms' fluxes of the flow governing equations [Koo00].

Except the previously described formulation (named at next WF1), according to which Dirichlet conditions are imposed for both the turbulent kinetic energy k and the dissipation rate ε of the boundary nodes, another formulation of wall functions (named at next WF2) has been included in the present algorithm considering Dirichlet conditions only for ε , while k is computed via equation (2.37). In accordance with this method, the turbulent dissipation rate ε of boundary nodes is evaluated as follows [Kou03]

$$\varepsilon = \frac{c_\mu^{3/4} k^{3/2}}{\kappa y} \quad (2.46)$$

while the friction stress, the dimensionless distance, the dimensionless velocity and the turbulent energy production term are defined as [Kou03]:

$$\tau_w = \begin{cases} \frac{\mu V_t}{\text{Re } y} & y^+ < 11.6 \\ \rho c_\mu^{1/4} k^{1/2} \frac{V_t}{\frac{1}{\kappa} \ln(y^+) + B} & y^+ \geq 11.6 \end{cases} \quad (2.47)$$

$$y^+ = \frac{\rho c_\mu^{1/4} k^{1/2} y}{\mu} \text{Re} \quad (2.48)$$

$$u^+ = \frac{V_t c_\mu^{1/4} k^{1/2}}{\tau_w} \quad (2.49)$$

$$P_k = \begin{cases} \tau_w \frac{V_t}{y} - \frac{\rho c_\mu^{3/4} k^{3/2} u^+}{y} \\ \tau_w \frac{c_\mu^{1/4} k^{1/2}}{\kappa y} - \frac{\rho c_\mu^{3/4} k^{3/2}}{y} \left[\frac{1}{\kappa} \ln(y^+) + B \right] \end{cases} \quad (2.50)$$

The second formulation considers the implementation of an one-equation model near the solid wall boundaries along with no-slip boundary conditions. Equation (2.37) is still being utilized for turbulent kinetic energy k , while for dissipation rate ε and turbulent dynamic viscosity μ_t the following equations are employed [Jon 97]

$$(\rho \varepsilon)_{\text{leq}} = \rho \frac{k^{3/2}}{l_\varepsilon} \quad (2.51)$$

$$\mu_t = \rho c_\mu l_\mu \sqrt{k} \quad (2.52)$$

where l_μ and l_ε are functions of y^+ [Jon97]:

$$l_\mu = \kappa c_\mu^{-3/4} y \left[1 - \exp(-y^+ / 70) \right] \quad (2.53)$$

$$l_\varepsilon = \kappa c_\mu^{-3/4} y \left[1 - \exp\left(-y^+ / \left(2\kappa c_\mu^{-3/4}\right)\right) \right] \quad (2.54)$$

$$y^+ = \frac{\rho y \sqrt{k}}{\mu} \text{Re} \quad (2.55)$$

Though the wall functions are implemented explicitly on the first layer or boundary nodes, the one-equation model is employed via an indirect transition scheme. The transition from the one-equation model to the two-equation governing equations, while getting away from the solid wall boundaries, is achieved by using a blending interpolation coefficient λ , taking values between 0 and 1. Thus, the expressions for dissipation rate ε and turbulent dynamic viscosity μ_t become [Jon97]

$$\frac{\partial(\rho\varepsilon)}{\partial t} + \lambda \left[\frac{\partial(\rho\varepsilon u_j)}{\partial x_j} - \left(\frac{\mu}{\text{Re}} + \frac{\mu_t}{\sigma_\varepsilon} \right) \frac{\partial\varepsilon}{\partial x_j} \right] = S_\varepsilon \quad (2.56)$$

$$S_\varepsilon = \lambda \left(c_1 f_1 P_k \frac{\varepsilon}{k} - c_2 f_2 \frac{\varepsilon^2}{k} \right) + (1 - \lambda) \alpha \left(\frac{\rho k^{3/2}}{l_\varepsilon} - \rho\varepsilon \right) \quad (2.57)$$

$$\mu_t = \lambda f_\mu c_\mu \frac{\rho k^2}{\varepsilon} + (1 - \lambda) \rho c_\mu k^{1/2} l_\mu \quad (2.58)$$

where α is a time constant taking values of the order of unity, adjusting time stiffness of ε at the boundary layer region [Jon97]. The interpolation coefficient λ controls not only the distance from the boundaries, in which transition takes place, but also the length of this transition over the computational mesh; it is computed via a hyperbolic tangential function as [Jon97]

$$\lambda = \frac{1}{2} \left[1 + \tanh \left(\frac{\text{Re}_y - \text{Re}_y^*}{A} \right) \right] \quad (2.59)$$

$$\text{Re}_y = \frac{\rho y \sqrt{k}}{\mu} \text{Re} \quad (2.60)$$

where Re_y is the local Reynolds number, depending on the distance from the closest wall and the laminar kinetic energy as shown in equation (2.60), while Re_y^* is the criterion for transition from the one-equation model to the two-equation one, with typical values of 50-200 [Jon97]. Constant A with typical values between 1 and 10, controls the number of cells, within which the transition is completed, and consequently the sharpness of this transition.

Finally, in order to control the unphysical increase of the source term (independently of the implemented formulation of boundary conditions), especially at stagnation points, special modifications

have been proposed in the open literature, such as the Kato-Launder modification, the Yap correction method, an explicit limitation to the computed corrections, the implicit treatment of the source term in case of an explicit iterative scheme, etc. [Kun92, Mav94, Dur96, Bla01, Kou03, Lyg14a, Lyg15], the latter being employed in this work.

2.1.3.2. Turbulence model $k\text{-}\omega$

The second incorporated two-equation turbulence model is $k\text{-}\omega$ [Saf74], which is described by similar to the previous model dimensionless differential governing equations (equation (2.37)), while its inviscid, viscous and source terms, based on the conservative variables (ρk , $\rho\omega$), are defined as

$$\vec{F}_{k\omega}^{inv} = \begin{pmatrix} \rho u k \\ \rho u \omega \end{pmatrix}, \vec{G}_{k\omega}^{inv} = \begin{pmatrix} \rho v k \\ \rho v \omega \end{pmatrix}, \vec{J}_{k\omega}^{inv} = \begin{pmatrix} \rho w k \\ \rho w \omega \end{pmatrix} \quad (2.61)$$

$$\vec{F}_{k\omega}^{vis} = \begin{pmatrix} \left(\frac{\mu}{\text{Re}} + \mu_t \sigma_k \right) \frac{\partial k}{\partial x} \\ \left(\frac{\mu}{\text{Re}} + \mu_t \sigma_\omega \right) \frac{\partial \omega}{\partial x} \end{pmatrix}, \vec{G}_{k\omega}^{vis} = \begin{pmatrix} \left(\frac{\mu}{\text{Re}} + \mu_t \sigma_k \right) \frac{\partial k}{\partial y} \\ \left(\frac{\mu}{\text{Re}} + \mu_t \sigma_\omega \right) \frac{\partial \omega}{\partial y} \end{pmatrix}, \vec{J}_{k\omega}^{vis} = \begin{pmatrix} \left(\frac{\mu}{\text{Re}} + \mu_t \sigma_k \right) \frac{\partial k}{\partial z} \\ \left(\frac{\mu}{\text{Re}} + \mu_t \sigma_\omega \right) \frac{\partial \omega}{\partial z} \end{pmatrix} \quad (2.62)$$

$$\vec{S}_{k\omega} = \begin{pmatrix} P_k - \beta^* \rho \omega k \\ \alpha \frac{\omega}{k} P_k - \beta \rho \omega^2 \end{pmatrix} \quad (2.63)$$

with constant values $\sigma_k=0.5$, $\sigma_\omega=0.5$, $\alpha=5/9$, $\beta=3/40$ and $\beta^*=0.09$ [Kim03]. The turbulent energy production term P_k is evaluated similarly to $k\text{-}\varepsilon$ model, while the turbulent viscosity is computed as

$$\mu_t = \alpha^* \frac{\rho k}{\omega} \quad (2.64)$$

where constant α^* is set equal to unity. Special treatment of the model's source term is required as well, to minimize a possible excessive and sudden increase in the values of the conservative variables; therefore, besides the implicit treatment of the source term in case of an explicit iterative scheme [Bla01], the turbulent energy production term is limited as follows [Kim03, Men03a]:

$$P_k = \min(P_k, 10\beta^* \rho \omega k) \quad (2.65)$$

2.1.3.3. Turbulence model SST

The last incorporated turbulence model is SST (Shear Stress Transport) [Men94, Men03a, Men03b, NASA], which is a combination of the previously presented $k\text{-}\omega$ and $k\text{-}\varepsilon$ models. The transition from the first to the second, while getting away from solid wall boundaries, is achieved by using appropriate blending functions. The inviscid and viscous terms of the SST model's governing equations are exactly the same with those of $k\text{-}\omega$, while the source term slightly differs as [Men03b, NASA]:

$$\vec{S}_{SST} = \vec{S}_{k\omega} + \vec{D}_{k\omega} \quad (2.66)$$

$$\vec{S}_{k\omega} = \begin{pmatrix} P_k - \beta^* \rho \omega k \\ \frac{\gamma^* \rho}{\mu_t} P_k - \beta \rho \omega^2 \end{pmatrix}, \quad \vec{D}_{k\omega} = \begin{pmatrix} 0 \\ 2(1-F_1) \frac{\rho \sigma_{\omega 2}}{\omega} \nabla k \nabla \omega \end{pmatrix} \quad (2.67)$$

The additional vector in the source term $\vec{D}_{k\omega}$ accounts for the switching between the two models. It includes the first blending function F_1 , which is equal to zero far away from the surface (k - ε model) and switches over to unity inside the boundary layer region (k - ω model). The turbulent energy production term is computed via equation (2.42), while the first blending function F_1 is evaluated as [Men03b]:

$$\begin{aligned} F_1 &= \tanh(\arg_1^4) \\ \arg_1 &= \min \left[\max \left(\frac{\sqrt{k}}{\beta^* \omega d}, \frac{500\mu}{\rho d^2 \omega \text{Re}} \right), \frac{4\rho \sigma_{\omega 2} k}{CD_{k\omega} d^2} \right] \\ CD_{k\omega} &= \max \left(2\rho \sigma_{\omega 2} \frac{1}{\omega} \nabla k \nabla \omega, 10^{-10} \right) \end{aligned} \quad (2.68)$$

The value of the turbulent viscosity is also based on a blending function; for its calculation the invariant of strain rates tensor is utilized [Men03, NASA]:

$$\mu_t = \frac{\rho \alpha_1 k}{\max(\alpha_1 \omega, SF_2)} \quad (2.69)$$

$$\begin{aligned} S &= \sqrt{2(S_{ij}S_{ij})} = \sqrt{2 \left(\frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right)} = \\ &= \left[2 \left(\frac{\partial u}{\partial x} \right)^2 + 2 \left(\frac{\partial v}{\partial y} \right)^2 + 2 \left(\frac{\partial w}{\partial z} \right)^2 + \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right)^2 \right]^{\frac{1}{2}} \end{aligned} \quad (2.70)$$

$$S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (2.71)$$

$$\begin{aligned} F_2 &= \tanh(\arg_2^2) \\ \arg_2 &= \max \left(\frac{2\sqrt{k}}{\beta^* \omega d}, \frac{500\mu}{\rho d^2 \omega \text{Re}} \right) \end{aligned} \quad (2.72)$$

The ϕ coefficients of the model, are also obtained by a blending function, which defines their value between a minimum value ϕ_1 and maximum value ϕ_2 (β^* and α_1 are constant values equal to 0.09 and 0.31, respectively). This function, utilizing minimum and maximum constants $\beta_1=0.075$, $\beta_2=0.0828$, $\gamma_1^*=0.555$, $\gamma_2^*=0.44$, $\sigma_{k1}=0.85$, $\sigma_{k2}=1$, $\sigma_{\omega 1}=0.5$ and $\sigma_{\omega 2}=0.856$, is described as follows:

$$\phi = F_1 \phi_1 + (1 - F_1) \phi_2 \quad (2.73)$$

Because of the activation of $k-\omega$ model inside the boundary layer region, special treatment of the source term of SST model is performed in the same to $k-\omega$ approach way [Liu96, Kim03, Men03b, Lyg14a, Lyg15].

2.2. Spatial discretization

For the discretization of the computational field and consequently the computation of numerical fluxes of flow and turbulence models, a node-centered finite-volume scheme is employed. According to this scheme the examined grid is divided into a number of finite control volumes, in each of which the magnitude of the conservative variables is assumed constant. In Figure 2.1 a median dual control volume for a node P in a two-dimensional triangular mesh is illustrated; it is defined by an assembly of linear segments, where each segment is connected to the midpoint of an edge and to the barycenter of its neighbouring triangular face [Kal96, Kal05a, Lyg12b, Sar14].

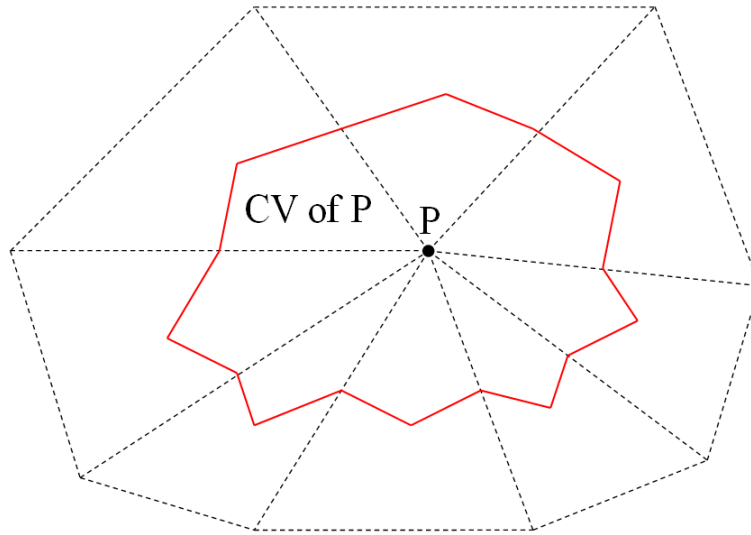


Figure 2.1: Median dual control volume of a node P in a two-dimensional triangular grid.

In three dimensions the employed median dual control volume of a node P is constructed by connecting lines defined by edge midpoints, barycenters of faces and barycenters of elements, sharing this node [Mav94, Kal96, Mav96, Koo00, Bla01, Kim03, Kou03, Kal05a, Lyg12b]. According to this definition the volume of each element is equally divided to each of its nodes' control volumes. Contributions to the control volume of a node P from different types of elements (prismatic, pyramidal and tetrahedral) are shown in Figure 2.2. The volume of a tetrahedral element can be computed, based on the coordinates of its nodes, as [Wolfram]

$$V_T = \frac{1}{3!} \begin{vmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \\ x_4 & y_4 & z_4 & 1 \end{vmatrix} \quad (2.74)$$

while the computation of the volumes of the prismatic and pyramidal elements is performed with their suitable division in tetrahedral ones.

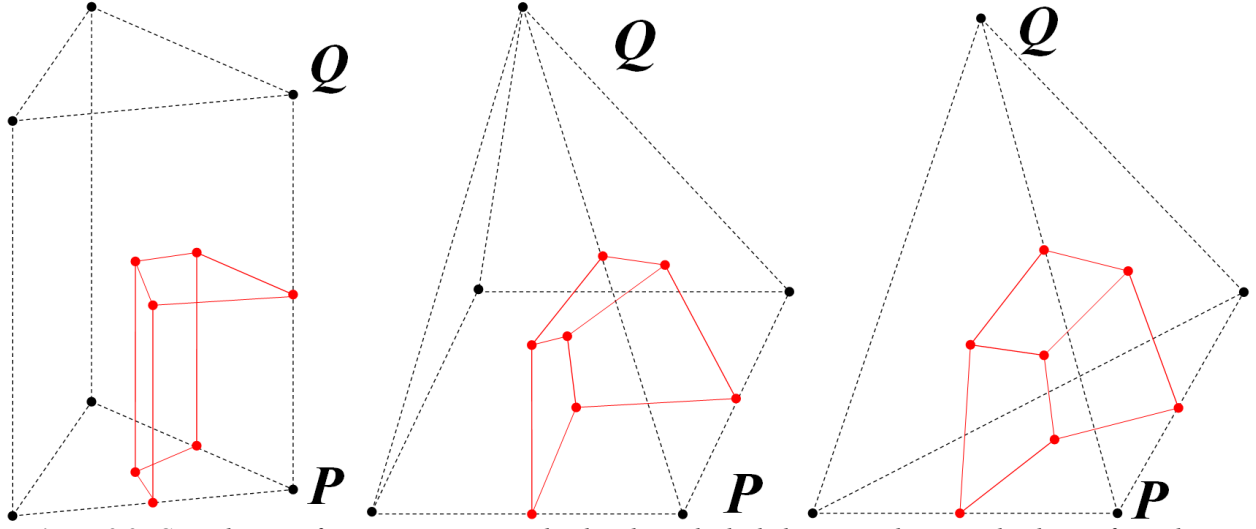


Figure 2.2: Contribution of a prismatic, pyramidal and tetrahedral element to the control volume of a node P .

Considering the aforementioned discretization scheme, equation (2.1) is integrated over the control volume CV_P of each node P as

$$\iiint_{CV_P} \frac{\partial \vec{W}}{\partial t} dx dy dz + \iiint_{CV_P} \frac{\partial \vec{F}^{inv}}{\partial x} + \frac{\partial \vec{G}^{inv}}{\partial y} + \frac{\partial \vec{J}^{inv}}{\partial z} - \frac{\partial \vec{F}^{vis}}{\partial x} - \frac{\partial \vec{G}^{vis}}{\partial y} - \frac{\partial \vec{J}^{vis}}{\partial z} dx dy dz = \iiint_{CV_P} \vec{S} dx dy dz \quad (2.75)$$

while employing the Green-Gauss divergence theorem is transformed as

$$\iiint_{CV_P} \frac{\partial \vec{W}}{\partial t} dx dy dz + \iint_{\partial CV_P} (\vec{H}^{inv} - \vec{H}^{vis}) ds = \iiint_{CV_P} \vec{S} dx dy dz \quad (2.76)$$

where ∂CV_P denotes the boundaries of the control volume of node P , defined by the facets constructed around the edges connecting node P with each neighbouring node Q . If ∂CV_{PQ} is the interfacing part of ∂CV_P and ∂CV_Q , $K_N(P)$ is the set of neighbouring nodes to P , and Γ is the domain's external boundary, then ∂CV_P is defined as:

$$\partial CV_P = \bigcup_{Q \in K_N(P)} \partial CV_{PQ} + (\partial CV_P \cap \Gamma) \quad (2.77)$$

\vec{H} is the vector of the inviscid or viscous numerical fluxes, evaluated at the midpoint of an edge connected to node P ; this midpoint actually coincides with the interface between the adjacent control volumes of nodes P and Q connected with this edge. As mentioned in Introduction, an edge-based data structure is used to reduce the required computational effort and accelerate the solution. Utilizing the outward unit normal vector \vec{n}_{PQ} of the corresponding ∂CV_{PQ} face of the control volume, the aforementioned vectors are described as [Koo00, Kou03]

$$\begin{aligned}\vec{\hat{H}}^{inv} &= \hat{n}_{PQ,x} \vec{F}^{inv} + \hat{n}_{PQ,y} \vec{G}^{inv} + \hat{n}_{PQ,z} \vec{J}^{inv} \\ \vec{\hat{H}}^{vis} &= \hat{n}_{PQ,x} \vec{F}^{vis} + \hat{n}_{PQ,y} \vec{G}^{vis} + \hat{n}_{PQ,z} \vec{J}^{vis}\end{aligned}\quad (2.78)$$

$$\vec{\hat{n}}_{PQ} = \frac{\vec{n}_{PQ}}{|\vec{n}_{PQ}|} = (\hat{n}_{PQ,x}, \hat{n}_{PQ,y}, \hat{n}_{PQ,z}) \quad (2.79)$$

where \vec{n}_{PQ} is defined as the vector sum of the outward normal vectors of all the facets forming ∂CV_{PQ} . Their computation is achieved by a single element-loop in a pre-computation stage, deriving actually the main edge-wise data structure of the algorithm. In Figure 2.3 part of such a vector \vec{n}_{PQ} contributed by a tetrahedron is presented; the illustrated barycenter of the tetrahedron G , medians of the faces G_1 , G_2 and of the edge M are utilized for the computation of this vector.

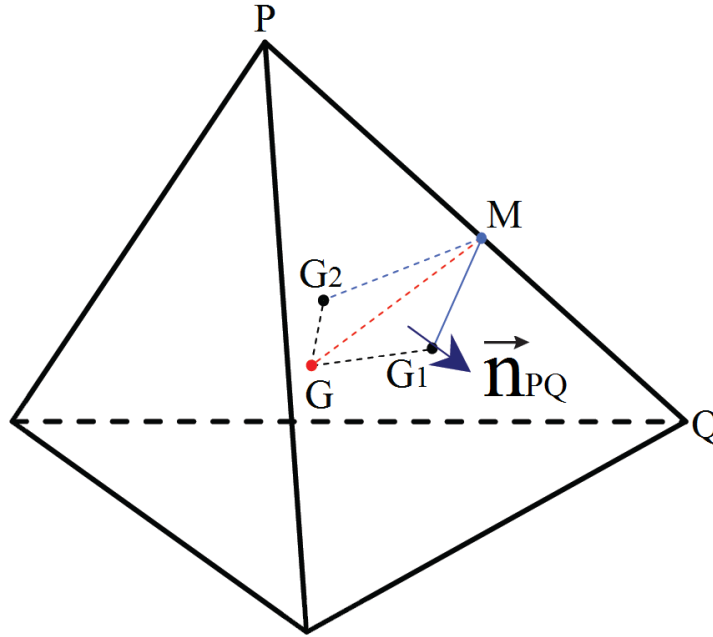


Figure 2.3: Part of vector \vec{n}_{PQ} contributed by a tetrahedron.

Thus, equation (2.76) is transformed as follows:

$$\iiint_{CV_P} \frac{\partial \vec{W}}{\partial t} dx dy dz + \sum_{Q \in K_N(P)} \iint_{\partial CV_{PQ}} (\vec{\hat{H}}^{inv} - \vec{\hat{H}}^{vis}) ds + \iint_{\partial CV_P \cap \Gamma} (\vec{\hat{H}}^{inv} - \vec{\hat{H}}^{vis}) ds = \iiint_{CV_P} \vec{S} dx dy dz \quad (2.80)$$

Assuming that the conservative variables at node P are equal to their mean values over CV_P , (main idea of the finite-volume method) the first term of equation (2.80) becomes:

$$\iiint_{CV_P} \frac{\partial \vec{W}}{\partial t} dx dy dz = \left(\frac{d \vec{W}}{dt} \right)_P \iiint_{CV_P} dx dy dz = \left(\frac{d \vec{W}}{dt} \right)_P V_P \quad (2.81)$$

Expressing then the integrals of the numerical fluxes as summations of fluxes through the faces composing the control volume of node P , equation (2.80) is transformed as

$$\left(\frac{d\vec{W}}{dt}\right)_P V_P + \sum_{Q \in K_N(P)} \vec{\Phi}_{PQ}^{inv} + \sum_{(K_{out} \in \partial CV_P \cap \Gamma)} \vec{\Phi}_{P,out}^{inv} - \sum_{Q \in K_N(P)} \vec{\Phi}_{PQ}^{vis} - \sum_{(K_{out} \in \partial CV_P \cap \Gamma)} \vec{\Phi}_{P,out}^{vis} = \iiint_{CV_P} \vec{S} dx dz dy \quad (2.82)$$

$$\begin{aligned} \vec{\Phi}_{PQ}^{inv} &= \iint_{\partial CV_{PQ}} \vec{H}^{inv} ds = \vec{f}(\vec{W}_{PQ}^L, \vec{W}_{PQ}^R, \vec{n}_{PQ}), \quad \vec{\Phi}_{PQ}^{vis} = \iint_{\partial CV_{PQ}} \vec{H}^{vis} ds = \vec{g}(\vec{W}_{PQ}^L, \vec{W}_{PQ}^R, \vec{n}_{PQ}) \\ \vec{\Phi}_{P,out}^{inv} &= \iint_{\partial CV_P \cap \Gamma} \vec{H}^{inv} ds = \vec{f}(\vec{W}_P, \vec{W}_{out}, \vec{n}_{out}), \quad \vec{\Phi}_{P,out}^{vis} = \iint_{\partial CV_P \cap \Gamma} \vec{H}^{vis} ds = \vec{g}(\vec{W}_P, \vec{W}_{out}, \vec{n}_{out}) \end{aligned} \quad (2.83)$$

where \vec{W}_{PQ}^L and \vec{W}_{PQ}^R are the vectors of the conservative variables on the left and right side of the edge PQ respectively, while \vec{W}_{out} is the corresponding vector on the boundary.

2.3. Numerical fluxes

2.3.1. Inviscid fluxes

For the computation of the flow equations' convective numerical fluxes, a one-dimensional Riemann problem in the direction of the normal vector of each face of the control volume of a node P is considered. As its exact solution would require excessive computational effort [Lan98], an upwind scheme using Roe's approximate Riemann solver [Roe81] is employed to evaluate the inviscid fluxes at the midpoint of the edge PQ , as follows

$$\vec{\Phi}_{PQ}^{inv} = \frac{1}{2} \left(\vec{H}^{inv}(\vec{W}_{PQ}^L, \vec{n}_{PQ}) + \vec{H}^{inv}(\vec{W}_{PQ}^R, \vec{n}_{PQ}) \right) - \frac{1}{2} |\underline{\tilde{A}}_{PQ}| (\vec{W}_{PQ}^R - \vec{W}_{PQ}^L) \quad (2.84)$$

where $\underline{\tilde{A}}$ is the Jacobian matrix of the convective flux vector \vec{H}^{inv} , also evaluated at the midpoint of the corresponding edge PQ by utilizing Roe's averaged values of the primitive variables (denoted with tilde $\tilde{\cdot}$) [Roe81, Ven95, Lan98, Koo00, Kou03], defined as

$$\vec{\tilde{U}}_{PQ} = \frac{\sqrt{\rho_L} \vec{U}_L + \sqrt{\rho_R} \vec{U}_R}{\sqrt{\rho_L} + \sqrt{\rho_R}} \quad (2.85)$$

where \vec{U}_L and \vec{U}_R are the values of primitive variables at the left and right side of the edge PQ respectively, assuming a first-order accurate scheme; in case of a higher-order one these values have to be reconstructed using the Taylor series expansions, as analyzed in next section. Additional information on the computation of the Jacobian matrix of the convective flux vector is given in Appendix A.

Based on the following equation (2.86), equation (2.84) is transformed in an equivalent formulation (2.87), implemented finally in this work [Roe81, Lan98]:

$$\vec{H}^{inv}(\vec{W}_{PQ}^R) - \vec{H}^{inv}(\vec{W}_{PQ}^L) = \underline{\tilde{A}}_{PQ} (\vec{W}_{PQ}^R - \vec{W}_{PQ}^L) \quad (2.86)$$

$$\bar{\Phi}_{PQ}^{inv} = \bar{H}^{inv}(\bar{W}_{PQ}^L, \bar{n}_{PQ}) + \bar{A}_{PQ}^-(\bar{W}_{PQ}^R - \bar{W}_{PQ}^L) \quad (2.87)$$

Considering the edge-wise data structure of the algorithm, evaluation of the convective fluxes for all the mesh nodes is achieved with a single edge-loop, as no information is needed about the cell topology [Lyg14a, Lyg15].

2.3.2. Higher-order accurate schemes and slope limiters

For a first-order accurate spatial scheme the left and right states of an edge PQ are approximated by the values at the end-points P and Q respectively. In a higher-order accurate scheme, these states are reconstructed with the Taylor series expansions, taking into account the corresponding values of more mesh nodes when computing the numerical fluxes. The incorporated higher-order accurate scheme is based on the MUSCL (Monotonic Upstream Scheme for Conservation Laws) reconstruction of the primitives or conservatives variables, using additionally a slope limiter (Van Albada -Van Leer [VanA82], Min-mod [Swe84] or Barth-Jespersen [Bar89]) to control the total variation of the reconstructed field [VanL85, Bar92, And94, Lan98, Bla01, ANSYS06, Lyg13b]. Thus, the left and right states for a primitive or a conservative variable U at the midpoint of an edge PQ are approximated as [Bar92, And94, Bla01, ANSYS06, Lyg13c, Sar14]

$$\begin{aligned} U_{PQ}^L &= U_P + \frac{1}{2} \cdot (\nabla U)^L \cdot \vec{r}_{PQ} \\ U_{PQ}^R &= U_Q - \frac{1}{2} \cdot (\nabla U)^R \cdot \vec{r}_{PQ} \end{aligned} \quad (2.88)$$

where, the first R/H side terms are the left and right nodes' values of variable U and \vec{r}_{PQ} is the vector connecting these nodes. In case of a second-order accurate scheme the extrapolation gradients $(\nabla U)^L$ and $(\nabla U)^R$ are equal to the gradients $(\nabla U)_P$ and $(\nabla U)_Q$ at the nodes P and Q respectively. For the evaluation of these derivatives the element-by-element approach [Bar92] can be employed, which for a node P , being the common vertex of the neighbouring tetrahedra T , is described as [Bar92]:

$$(\nabla U)_P = \frac{1}{V_P} \sum_{T \in K_T(P)} \frac{V_T}{4} (\nabla U)_T \quad (2.89)$$

where V_P and V_T are the volumes of the control volume of node P and adjacent element T . Nevertheless, in this study an equivalent expression, derived by the Green-Gauss linear representation method, is employed, which is more suitable for the utilized edge-based data structure of the algorithm [Bar92, Bla01]:

$$(\nabla U)_P = \frac{1}{V_P} \sum_{Q \in K_N(P)} \frac{1}{2} (U_P + U_Q) \cdot \vec{n}_{PQ} \quad (2.90)$$

In case of a boundary node (Figure 2.5) the previous equation is modified to include also the boundary interfaces as follows [Lyg13c]:

$$(\nabla U)_P = \frac{1}{V_P} \left(\sum_{Q \in K_N(P)} \frac{1}{2} (U_P + U_Q) \cdot \vec{n}_{PQ} + \sum_{(K_{out} \in \partial C_P \cap \Gamma)} U_P \cdot \vec{n}_{out} \right) \quad (2.91)$$

For the minimization of the total variation in the reconstructed field three slope limiters have been incorporated in the present algorithm, namely the Van Albada -Van Leer [VanA82], the Min-mod [Swe84] and the Barth-Jespersen [Bar89].

The first limiter is actually an expansion of the MUSCL scheme, maintaining monotonicity between the values at the end-points P and Q , while its formulation is based on the centered $(\nabla U)_{PQ}^c$ and upwind gradients $(\nabla U)_P^u, (\nabla U)_Q^u$, defined as:

$$\begin{aligned} (\nabla U)_{PQ}^c \cdot \vec{r}_{PQ} &= U_Q - U_P \\ (\nabla U)_P^u &= 2(\nabla U)_P - (\nabla U)_{PQ}^c \\ (\nabla U)_Q^u &= 2(\nabla U)_Q - (\nabla U)_{PQ}^c \end{aligned} \quad (2.92)$$

Once these gradients are obtained, equation (2.88) including a non-linear slope limiter is formulated as [Bla01, Lyg13c]

$$\begin{aligned} U_{PQ}^L &= U_P + \frac{1}{2} \cdot \Phi \left((\nabla U)_P^u \cdot \vec{r}_{PQ}, (\nabla U)_{PQ}^c \cdot \vec{r}_{PQ} \right) \\ U_{PQ}^R &= U_Q - \frac{1}{2} \cdot \Phi \left((\nabla U)_Q^u \cdot \vec{r}_{PQ}, (\nabla U)_{PQ}^c \cdot \vec{r}_{PQ} \right) \end{aligned} \quad (2.93)$$

where Φ is the limiter function, described as [VanA82]

$$\Phi(a, b) = \begin{cases} \frac{(a^2 + e)b + (b^2 + e)a}{a^2 + b^2 + 2e} & \text{if } ab > 0 \\ 0 & \text{if } ab \leq 0 \end{cases} \quad (2.94)$$

where e is a very small number with a typical value of $1.0E-16$, used to prevent division by zero.

The second incorporated Min-mod limiter is characterized by the smoothing of solution, choosing always the slope with the smallest magnitude; its formulation is exactly the same to this of the Van Albada-Van Leer (equations (2.93) and (2.94)), while the corresponding limiting function Φ is defined as [Swe84, Lyg13c]:

$$\Phi(a, b) = \begin{cases} a & \text{if } |a| < |b| \text{ and } ab > 0 \\ b & \text{if } |b| < |a| \text{ and } ab > 0 \\ 0 & \text{if } ab \leq 0 \end{cases} \quad (2.95)$$

The Barth-Jespersen [Bar89] approach extends limitation further of the values at the two end-nodes P and Q , to these of all adjacent points connected with an edge to the pre-mentioned nodes. It is formulated similarly to the previous limiters as [Bar89, Lyg11]

$$\begin{aligned} U_{PQ}^L &= U_P + \Psi_P \cdot \frac{1}{2} \cdot (\nabla U)_P \cdot \vec{r}_{PQ} \\ U_{PQ}^R &= U_Q - \Psi_Q \cdot \frac{1}{2} \cdot (\nabla U)_Q \cdot \vec{r}_{PQ} \end{aligned} \quad (2.96)$$

where the coefficients Ψ_P, Ψ_Q are defined as:

$$\Psi_P = \min \{ \Psi_{PQ}, Q \in K_N(P) \cap \{P\} \} \quad (2.97)$$

$$\Psi_{PQ} = \begin{cases} \min \{ 1, \Delta_{1,\max} / \Delta_2 \} & \Delta_2 > 0 \\ \min \{ 1, \Delta_{1,\min} / \Delta_2 \} & \Delta_2 < 0 \\ 1 & \Delta_2 = 0 \end{cases} \quad (2.98)$$

$$\begin{aligned} \Delta_2 &= U_{rec,L} - U_P \\ \Delta_{1,\max} &= U_{\max} - U_P \\ \Delta_{1,\min} &= U_{\min} - U_P \\ U_{\max} &= \max \{ U_Q, Q \in K_N(P) \cap \{P\} \} \\ U_{\min} &= \min \{ U_Q, Q \in K_N(P) \cap \{P\} \} \\ U_{rec,L} &= U_P + \frac{1}{2} \cdot (\nabla U)_P \cdot \vec{r}_{PQ} \end{aligned} \quad (2.99)$$

Although this scheme is assumed to be more effective in assuring monotonicity comparing to Van Albada-Van Leer and Min-mod limiters, it is susceptible of exhibiting oscillations and consequently poor convergence due to the overreacted limiting of small amplitude noise it entails [And94, Del13].

In case of an even higher-order accurate scheme, the extrapolation gradients are obtained via a blending function, utilizing the centered (^c) and upwind (^u) derivatives again, as [Kaz12, Smi07]

$$\begin{aligned} (\nabla U)^L \cdot \vec{r}_{PQ} &= \frac{1}{2} \left[(1-\beta) \cdot (\nabla U)_P^u \cdot \vec{r}_{PQ} + (1+\beta) \cdot (\nabla U)_{PQ}^c \cdot \vec{r}_{PQ} \right] \\ (\nabla U)^R \cdot \vec{r}_{PQ} &= \frac{1}{2} \left[(1-\beta) \cdot (\nabla U)_Q^u \cdot \vec{r}_{PQ} + (1+\beta) \cdot (\nabla U)_{PQ}^c \cdot \vec{r}_{PQ} \right] \end{aligned} \quad (2.100)$$

where parameter β can be selected suitably to derive a family of reconstruction schemes; in structured grids $\beta=1$ yields a fully upwind scheme, $\beta=0$ a semi-upwind approximation (Fromm's scheme), and $\beta=-1$ a central differencing one [Kaz12]. If it is set equal to $1/3$, it leads to a third-order accurate upwind scheme, although third-order accuracy is strictly correct only for linear problems [Kaz12, Smi07]. Considering these gradients along with the requirement for minimization of oscillations in the reconstructed field, Van Albada - Van Leer limiter in a slightly different formulation to the pre-mentioned one is used as follows [Kaz12, Smi07]

$$\begin{aligned} U_{PQ}^L &= U_P + \frac{1}{2} \cdot X \left((\nabla U)_P^u \cdot \vec{r}_{PQ}, (\nabla U)_{PQ}^c \cdot \vec{r}_{PQ} \right) (\nabla U)^L \cdot \vec{r}_{PQ} \\ U_{PQ}^R &= U_Q - \frac{1}{2} \cdot X \left((\nabla U)_Q^u \cdot \vec{r}_{PQ}, (\nabla U)_{PQ}^c \cdot \vec{r}_{PQ} \right) (\nabla U)^R \cdot \vec{r}_{PQ} \end{aligned} \quad (2.101)$$

where X is the non-linear limiter function defined as:

$$X(a, b) = \frac{ab + |ab| + e}{a^2 + b^2 + e} \quad (2.102)$$

2.3.3. Viscous fluxes

The computation of viscous fluxes is based on the gradients of the velocity components and temperature at the middle of each edge (interface of adjacent control volumes) of the grid. For the evaluation of these derivatives two methods were included in the proposed algorithm, an element-based [Bra96, Kal96, Bla01, Kal05a, Lyg14a, Lyg15] and a nodal-averaging one [Bla01, Lyg14a, Lyg15].

The first methodology introduces new control volumes, the edge-dual volumes, which include all the neighbouring cells sharing a common edge [Kal96, Kal05a, Lyg14e]; in Figure 2.4. such volumes around an edge e ($\sim PQ$) are illustrated by prismatic, tetrahedral and different types of adjacent mesh elements. The divergence theorem is then implemented on these new finite volumes and the gradients of primitive variables at the midpoints of each edge are calculated, performing the surface integrals along the edge-dual boundaries as [Kal96, Kal05a, Ahn06, Ahn07, Lyg14a, Lyg15]

$$\left(\frac{\partial U}{\partial x}\right)_e = \frac{1}{V_e} \int_{V_e} U \hat{n}_x ds = \frac{1}{V_e} \sum_{k=1}^m n_{x,k} \sum_{l=1}^{L_k} \frac{U_l}{L_k} \quad (2.103)$$

where V_e is the volume of the edge-dual volume, m is the number of its boundary surfaces and L_k is the number of nodes at each such surface. Despite the capability of this method to avoid spurious oscillations [Kal05a], it leads to significantly increased computational effort as it requires a loop over an element-based or a cell-based data structure [Lyg14a, Lyg14e, Lyg15].

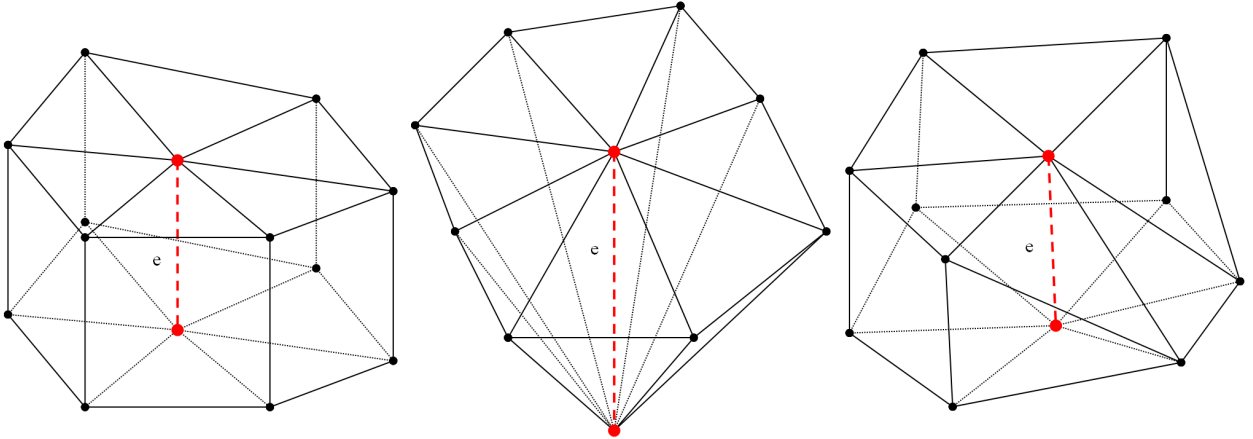


Figure 2.4: Edge-dual volumes of edge e composed of prisms, tetrahedra and different types of elements.

Considering the previous state along with the edge-wise evaluation of gradients at each node of the mesh by implementing equations (2.90) and (2.91), another scheme, the nodal-averaging one, was incorporated in the present algorithm [Bla01]. This method is particularly attractive, as gradients' calculation can be completed by a single edge-loop, which is employed for the convective fluxes anyway, without requiring any additional storage. Utilizing the corresponding gradients at the end-points P and Q of the examined edge PQ , as well as the directional derivative along these nodes $(\partial U / \partial l)_{PQ}$, this scheme is described mathematically as [Bla01, Lyg14a, Lyg15]:

$$(\nabla \bar{U})_{PQ} = \frac{1}{2} \left((\nabla U)_P + (\nabla U)_Q \right) \quad (2.104)$$

$$\left(\frac{\partial U}{\partial l}\right)_{PQ} \approx \frac{U_Q - U_P}{|\vec{r}_{PQ}|}, \quad \vec{r}_{PQ} = \frac{\vec{r}_{PQ}}{|\vec{r}_{PQ}|} \quad (2.105)$$

$$(\nabla U)_{PQ} = (\nabla \bar{U})_{PQ} - \left[(\nabla \bar{U})_{PQ} \cdot \vec{r}_{PQ} - \left(\frac{\partial U}{\partial l}\right)_{PQ} \right] \vec{r}_{PQ} \quad (2.106)$$

Furthermore, as it depends on the edges of the grid and not on the type of the elements (tetrahedral, prismatic, pyramidal), it appears even more valuable in case of utilization of hybrid grids.

2.3.4. Turbulence models' fluxes

For the computation of turbulence models' fluxes the same node-centered finite-volume discretization scheme is implemented. The convective fluxes are evaluated at the midpoint of each edge PQ , employing a simple first-order accurate upwind scheme as [Lar91, And94, Koo00, Kou03, ANSYS06, Lyg14a, Lyg15]

$$\Phi_{\phi, PQ}^{inv} = (\rho V_n)^+ \phi_P + (\rho V_n)^- \phi_Q \quad (2.107)$$

$$(\rho V_n)^+ = \max(\rho V_n, 0), \quad (\rho V_n)^- = \min(\rho V_n, 0) \quad (2.108)$$

where ϕ is each primitive variable of the model and $V_n = \vec{V} \cdot \vec{n}_{PQ}$ is the value of the normal to the face around the edge PQ velocity. In general, a higher-order accurate scheme isn't necessary for the computation of these numerical fluxes, as the viscous terms are the main reacting ones in turbulence models' PDE's; the numerical diffusion imported due to the first-order method can be assumed quite reasonably negligible [And94, Kou03, Lyg14a, Lyg15].

As far as the computation of the diffusive fluxes at the midpoint of each edge is concerned, the gradients of the corresponding primitive variables are needed to be pre-computed in the same way velocity components' gradients are evaluated for the flow equations. Since these derivatives are obtained, the viscous fluxes in case of turbulence model $k-\varepsilon$ are calculated for an edge PQ as [Kou03]:

$$\vec{\Phi}_{PQ}^{vis} = \left(\left(\frac{\mu}{\text{Re}} + \frac{\mu_t}{\sigma_k} \right) \left(\frac{\partial k}{\partial x} n_{PQ,x} + \frac{\partial k}{\partial y} n_{PQ,y} + \frac{\partial k}{\partial z} n_{PQ,z} \right) \right) \quad (2.109)$$

$$\left(\left(\frac{\mu}{\text{Re}} + \frac{\mu_t}{\sigma_\varepsilon} \right) \left(\frac{\partial \varepsilon}{\partial x} n_{PQ,x} + \frac{\partial \varepsilon}{\partial y} n_{PQ,y} + \frac{\partial \varepsilon}{\partial z} n_{PQ,z} \right) \right)$$

For $k-\omega$ and SST turbulence models a corresponding relation is extracted, whose differences to the previous one are focused on the specific dissipation rate ω equation.

2.3.5. Boundary conditions

The contribution of boundary surfaces has to be also considered in the flux balance of the corresponding nodes; in this study, wall, inlet, outlet and symmetry boundary conditions are encountered. In general such fluxes are computed at the barycenters of boundary faces, using the arithmetic averages of conservative variables of their nodes, and assigned to the same nodes, weighted by the area of the face, which corresponds to them, e.g., solid wall boundary fluxes in case of inviscid flow or inlet/outlet ones.

As far as the solid wall boundary nodes are concerned, a free-slip condition is implemented for inviscid flows, while for laminar and turbulent ones a no-slip condition is used. The free-slip condition is

implemented implicitly, by adding a flux with zero normal to the boundary face velocity V_n , described as [Mav94]

$$V_n = \vec{V} \cdot \vec{n}_{out} = 0 \quad (2.110)$$

where $\vec{n}_{out} = (n_{out,x}, n_{out,y}, n_{out,z})$ is the normal to the boundary face vector. Such a vector is shown in Figure 2.5; the illustrated median point of face M, medians of edges M_P , M_Q , M_R and median point of tetrahedron G are utilized for the computation of this normal vector. The added free-slip convective flux is calculated finally as follows:

$$\vec{H}_{freeslip} = \begin{pmatrix} \rho V_n \\ \rho u V_n + p n_{out,x} \\ \rho v V_n + p n_{out,y} \\ \rho w V_n + p n_{out,z} \\ (\rho E + p) V_n \end{pmatrix} = \begin{pmatrix} 0 \\ p n_{out,x} \\ p n_{out,y} \\ p n_{out,z} \\ 0 \end{pmatrix} \quad (2.111)$$

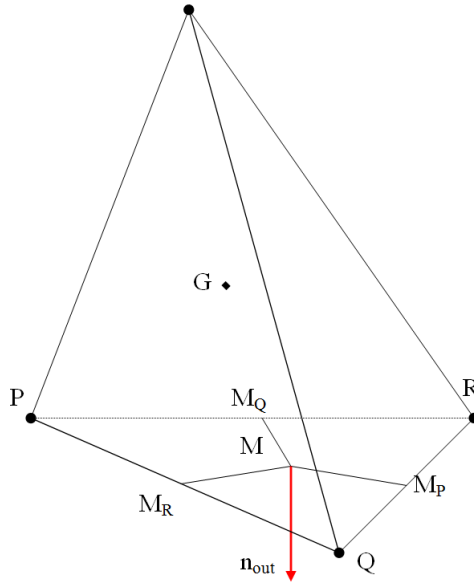


Figure 2.5: Normal to the boundary face PQR vector \vec{n}_{out} .

The same free-slip condition is applied in case of employing the turbulence model $k-\varepsilon$ along with wall functions WF1 or WF2; although wall functions should be imposed at the first layer of nodes from the solid surface, they are implemented on the boundary nodes, considering that in the usually employed grids, which are highly stretched in this area, the distance of the first layer from the solid walls is very small [Mav94, Kou03]. In this case, the turbulent kinetic energy k and the dissipation rate ε at wall boundary nodes are evaluated using the wall functions, while the convective flux of equation (2.111), along with the diffusive flux caused by the computed friction stress tensor τ_w are added to the flux balance of the corresponding nodes [Koo00, Kou03].

Unlike the aforementioned flows, for laminar as well as turbulent flows employing $k-\varepsilon$ with one-equation model near the solid wall region, $k-\omega$ or SST model, no-slip conditions are imposed explicitly at the corresponding nodes, by zeroizing their components of velocity (low Reynolds number approach). As far as the turbulence models' variables are concerned, the turbulent kinetic energy k and the turbulent

dynamic viscosity μ_t are set similarly equal to zero at these nodes; the dissipation rate ε for the first model is computed by employing equation (2.56), while the specific dissipation rate for the $k-\omega$ and SST models is evaluated as

$$\rho\omega_{wall} = \frac{60\mu}{\beta y^2 \text{Re}} \quad (2.112)$$

where y is the distance of the wall boundary node from the nearest non-boundary one of the grid. Due to this condition $k-\omega$ and SST require finer grids in this region with y^+ less than 2.5 [Saf74, Men94, Bla01, Bre02, Kim03, Kou03, Men03a, Men03b].

In inlet boundary faces, an one-dimensional Riemann problem is assumed again between the face's barycenter and far field for the computation of the flow PDE's convective fluxes; the obtained fluxes are shared then to the corresponding boundary nodes. The aforementioned problem is solved by employing the Steger-Warming scheme [Ste81, Lan98] via the following equation:

$$\vec{H}_{K,out}^{inv} = \tilde{A}_K^+ \vec{W}_K + \tilde{A}_K^- \vec{W}_{out} \quad (2.113)$$

where subscript K denotes the barycenter of the boundary face, while subscript out denotes the far field; the values of the variables of vector \vec{W}_{out} are obtained either from the far field or the boundary barycenter, depending on the type of the flow (internal or external) [Hir90, Bla01]. Further information for the computation of Jacobian matrices is given in Appendix A. Following the common practice in the open literature [Kou03], the diffusive fluxes are omitted due to their negligible contribution away from solid obstacles. If a turbulent flow is encountered, Dirichlet conditions are imposed for the turbulence models' variables at this region. In case of $k-\varepsilon$ model, their values are defined, based on the turbulence intensity T_u and the ratio of turbulent to laminar viscosity a in inlet area, as [Lau74, Kim03, Kou03]:

$$k_{in} = \frac{3}{2} (T_u |\vec{V}_{in}|)^2, \quad \mu_t = \frac{a\mu}{\text{Re}}, \quad \varepsilon_{in} = \frac{c_\mu f_\mu k_{in}^2}{a} \quad (2.114)$$

Typical values for this ratio and turbulence intensity at the inlet boundary are 0.1 and 1% respectively. In the same way, for $k-\omega$ and SST turbulence models the imposed values are defined as [NASA]

$$\omega_{in} = C_1 \frac{|\vec{V}_{in}|}{L}, \quad \mu_t = \frac{a\mu}{\text{Re}}, \quad k_{in} = \frac{\mu_t \omega_{in}}{\rho_{in}} \quad (2.115)$$

where L is a characteristic length (equal to unity in this algorithm due to its dimensionless formulation), C_1 is a constant with value between 1 and 10, and a is again the ratio of turbulent to laminar viscosity with value between $1.0E-2$ and $1.0E-5$ [NASA].

Similarly to inlet boundary faces, a one-dimensional Riemann problem is considered and solved at outlet ones for the computation of the flow PDE's convective fluxes, while the values of the variables of vector \vec{W}_{out} are obtained again either from the far field or the boundary barycenter depending on the type of the flow (internal or external); the viscous fluxes are neglected again [Hir90, Bla01]. However, in turbulence models the corresponding variables are not evaluated with Dirichlet conditions; the convective numerical flux for each primitive variable ϕ in an outlet boundary node P is calculated via a simple upwind scheme, expressed as follows

$$\Phi_{\phi,P-out}^{inv} = (\rho V_n) \phi_P \quad (2.116)$$

while its diffusive flux is omitted in the same way to the flow equations.

Finally, for a symmetry surface free-slip boundary conditions are imposed to the flow equations similarly to these for solid wall boundaries along with an inviscid flow.

2.3.6. Source terms

As mentioned in the first section of this Chapter, the source term of flow equations is set equal to zero, while this of turbulence models is computed, based on the finite-volume discretization, as follows [Bla01, Lyg14e]:

$$\iiint_{C_P} \vec{S} dx dy dz = \vec{S}_P V_P \quad (2.117)$$

2.4. Time integration

For time integration two schemes have been incorporated in the proposed methodology for both the Navier-Stokes and the turbulence models PDE's (either selected by the user), a point-implicit one using the Jacobi or the Gauss-Seidel algorithm [Ven95, Kou03, Lyg11] and an explicit one utilizing a second-order temporal accurate four-stage Runge-Kutta (RK(4)) method [Lal88a, Kal96, Bla01, Sor03, Lyg14a, Lyg15]. Equation (2.82) is transformed in the following formulation for their implementation

$$\begin{aligned} -V_P \left(\frac{d\vec{W}}{dt} \right)_P &= -V_P \frac{\Delta \vec{W}_P^{n+1}}{\Delta t_P} = \\ \sum_{Q \in K_N(P)} \vec{\Phi}_{PQ}^{inv} + \sum_{(K_{out} \in \partial CV_P \cap \Gamma)} \vec{\Phi}_{P,out}^{inv} - \sum_{Q \in K_N(P)} \vec{\Phi}_{PQ}^{vis} - \sum_{(K_{out} \in \partial CV_P \cap \Gamma)} \vec{\Phi}_{P,out}^{vis} - \vec{S}_P V_P &= \vec{R}_P^n \end{aligned} \quad (2.118)$$

where Δt_P is the local time step at node P , which depending on the type of the flow (inviscid, laminar or turbulent) is computed as [Kim03, Lyg11]

$$\begin{aligned} \Delta t_P^{inv} &= CFL \cdot \frac{0.5\alpha_{min\,edge,P}}{|\vec{U}_P| + c_P} \\ \Delta t_P^{lam} &= CFL \cdot \frac{(0.5\alpha_{min\,edge,P})^2}{(|\vec{U}_P| + c_P) 0.5\alpha_{min\,edge,P} + \frac{2\gamma\mu}{\rho_P \text{Re Pr}}} \\ \Delta t_P^{turb} &= CFL \frac{V_P}{(\lambda_c^x + \lambda_c^y + \lambda_c^z) + 4 \cdot (\lambda_v^x + \lambda_v^y + \lambda_v^z)} \end{aligned} \quad (2.119)$$

where $|\vec{U}_P|$ is the value of velocity at node P , c_P is the speed of sound evaluated on the same node and $\alpha_{min\,edge,P}$ is the length of the shortest edge connected to P . The convective spectral radii λ_c and the viscous spectral radii λ_v are evaluated as [Kim03, Kal05a]

$$\lambda_c^i = (|U_i| + c_P) \Delta A_i^j, \quad j = x, y, z \quad (2.120)$$

$$\lambda_v^i = \max \left(\frac{4}{3\rho}, \frac{\gamma}{\rho} \right) \left(\frac{\mu}{\text{Pr Re}} + \frac{\mu_t}{\text{Pr t}} \right) \frac{(\Delta A_i^j)^2}{V_p}, \quad j = x, y, z \quad (2.121)$$

where ΔA_i^j is the sum of the absolute projections of the node P control volume surfaces' normal vectors in direction j , calculated as [Kim03]:

$$\Delta A_i^j = \frac{1}{2} \sum_{CS_p} |\Delta A^j|_{CS_{p,j}}, \quad j = x, y, z \quad (2.122)$$

The local time stepping technique is commonly employed in CFD, enhancing each control volume with the maximum permissible time step, which consequently results in the acceleration of the convergence to the steady-state solution [Bla01]; in case a global time step is required, it is defined as the smallest of the local time steps of all the nodes of the mesh.

2.4.1. Point-implicit scheme

For the employment of the point-implicit scheme, the R/H side of equation (2.118) has to be linearized per Newton as [Ven 95]:

$$R(\vec{W}^{n+1}) = R(\vec{W}^n + \delta \vec{W}) = R(\vec{W}^n) + \left(\frac{\partial R}{\partial \vec{W}} \right)^n \cdot \delta \vec{W} + O(\delta \vec{W}^2) \quad (2.123)$$

where n denotes the current pseudo time step, in which the numerical fluxes are computed. Considering the desired formulation $A \cdot x = b$ with unknowns the corrections of conservative variables, equation (2.118) becomes as follows [Ven95, Kou03]

$$\begin{aligned} \frac{V}{\Delta t} \cdot \delta \vec{W} + \left(\frac{\partial R}{\partial \vec{W}} \right)^n \cdot \delta \vec{W} &= -R(\vec{W}^n) \Rightarrow \\ \left(\frac{V}{\Delta t} \cdot I + \left(\frac{\partial R}{\partial \vec{W}} \right)^n \right) \cdot \delta \vec{W} &= -R(\vec{W}^n) \end{aligned} \quad (2.124)$$

where I is a unit matrix with dimensions 5×5 (2×2 for the turbulence models). According to this approach the inviscid and viscous numerical fluxes have to be linearized similarly to the way they are computed.

The inviscid numerical flux at the midpoint of an edge PQ is defined for the node P via Newton linearization as [Ven95, Kou03]

$$(\vec{\Phi}_{PQ}^{inv})^{n+1} = (\vec{\Phi}_{PQ}^{inv})^n + \underline{B}_d^n \delta \vec{W}_P^{n+1} + \underline{B}_o^n \delta \vec{W}_Q^{n+1} \quad (2.125)$$

where \underline{B}_d^n and \underline{B}_o^n are diagonal and off-diagonal matrices, calculated as follows [Kou03]:

$$\begin{aligned} \underline{B}_d^n &= \tilde{A}_{PQ}^+ (\vec{W}_{PQ}, \vec{n}_{PQ}) \\ \underline{B}_o^n &= \underline{A}_Q (\vec{W}_Q, \vec{n}_{PQ}) - \tilde{A}_{PQ}^+ (\vec{W}_{PQ}, \vec{n}_{PQ}) \end{aligned} \quad (2.126)$$

The previous equations (2.125) and (2.126) are obtained using the alternative formulation of the Roe's approximate Riemann solver, which for time step $n+1$ is described as [Lan98]:

$$\left(\vec{\Phi}_{PQ}^{inv}\right)^{n+1} = \vec{H}^{inv}\left(\vec{W}_{PQ}^R, \vec{n}_{PQ}\right)^{n+1} - \tilde{\underline{A}}_{PQ}^+\left(\vec{W}_{PQ}^{R,n+1} - \vec{W}_{PQ}^{L,n+1}\right) \quad (2.127)$$

The inviscid flux vector $\vec{H}^{inv}(\vec{W}_{PQ}^L, \vec{n}_{PQ})$ at the midpoint of edge PQ , based on the conservative variables of L/H node P and the corresponding vector \vec{n}_{PQ} for the time step $n+1$, can be calculated as [Ven95, Kou03]

$$\vec{H}^{inv}(\vec{W}_{PQ}^L, \vec{n}_{PQ}) = \underline{A}_P \cdot \vec{W}_{PQ}^L \quad (2.128)$$

$$\left(\vec{H}_P^{inv}\right)^{n+1} = \left(\vec{H}_P^{inv}\right)^n + \left(\frac{\partial \vec{H}^{inv}}{\partial \vec{W}}\right)_P^n \cdot \delta \vec{W}_P = \underline{A}_P \cdot \vec{W}_P^n + \underline{A}_P \cdot \delta \vec{W}_P \quad (2.129)$$

while considering equation (2.129), equation (2.127) obtains its final formulation [Ven95]:

$$\begin{aligned} \left(\vec{\Phi}_{PQ}^{inv}\right)^{n+1} &= \left(\vec{H}_Q^{inv}\right)^n + \underline{A}_Q \delta \vec{W}_Q - \tilde{\underline{A}}_{PQ}^+(\vec{W}_Q^n + \delta \vec{W}_Q - \vec{W}_P^n - \delta \vec{W}_P) \\ &= \left(\vec{H}_Q^{inv}\right)^n - \tilde{\underline{A}}_{PQ}^+(\vec{W}_Q^n - \vec{W}_P^n) + (\underline{A}_Q - \tilde{\underline{A}}_{PQ}^+) \delta \vec{W}_Q + \tilde{\underline{A}}_{PQ}^+ \delta \vec{W}_P \\ &= \left(\vec{\Phi}_{PQ}^{inv}\right)^n + (\underline{A}_Q - \tilde{\underline{A}}_{PQ}^+) \delta \vec{W}_Q + \tilde{\underline{A}}_{PQ}^+ \delta \vec{W}_P \end{aligned} \quad (2.130)$$

Although a higher-order accurate scheme is implemented for the computation of convective fluxes, the corresponding linearized terms in the L/H side of equation (2.124) are based on a first-order one for simplification reasons; nevertheless, they don't constrain the accuracy in case of steady-state flows.

In a similar to the inviscid fluxes way, the viscous ones at the midpoint of an edge PQ are defined for the node P via Newton linearization as [Kou03]

$$\left(\vec{\Phi}_{PQ}^{vis}\right)^{n+1} = \left(\vec{\Phi}_{PQ}^{vis}\right)^n + \underline{C}_d^n \delta \vec{W}_P^{n+1} + \underline{C}_o^n \delta \vec{W}_Q^{n+1} + \underline{C}_{adj}^n \delta \vec{W}_R^{n+1} \quad (2.131)$$

where \underline{C}_d^n and \underline{C}_o^n are also diagonal and off-diagonal matrices, used with the conservative variables' corrections of nodes P and Q , while \underline{C}_{adj}^n is combined with the corresponding corrections of the rest nodes R adjacent to the edge PQ ; their computation is based on the scheme employed for the evaluation of velocity components and temperature gradients. Further information for their formulation is contained in Appendix B.

The numerical fluxes derived by the boundary conditions (free-slip and inlet/outlet) have also to be linearized per Newton. Thus, the inviscid diagonal contribution at the inlet/outlet boundary nodes is defined as [Ven95, Kou03]

$$\left(\vec{H}_{P,out}^{inv}\right)^{n+1} = \left(\vec{H}_{P,out}^{inv}\right)^n + \tilde{\underline{A}}_K^+ \cdot \delta \vec{W}_P^{n+1} \quad (2.132)$$

while at the free-slip surfaces' nodes as [Kou03]:

$$\left(\bar{H}_{P, \text{freeslip}}^{\text{inv}}\right)^{n+1} = \left(\bar{H}_{P, \text{freeslip}}^{\text{inv}}\right)^n + \frac{\left(\partial \bar{H}_{P, \text{freeslip}}^{\text{inv}}\right)^n}{\mathcal{G}\bar{U}_P^n} \cdot \frac{\mathcal{G}\bar{U}_P^n}{\mathcal{G}\bar{W}_P^n} \cdot \delta \bar{W}_P^{n+1} \quad (2.133)$$

Further information for its computation is included in Appendix C. The free-slip and inlet/outlet viscous diagonal contributions are assumed negligible and are omitted similarly to the corresponding internal fluxes (section 2.3.5). Nevertheless, in case of turbulent flow modelled with $k-\varepsilon$ WF1 or WF2, these solid wall surfaces are taken into account; as the friction stress tensor τ_w flux is added to the viscous terms of Navier-Stokes PDE's, its linearized contribution has to be considered in the diagonal matrix \underline{C}_d^n as:

$$\left(\bar{H}_{P, \tau_w}^{\text{vis}}\right)^{n+1} = \left(\bar{H}_{P, \tau_w}^{\text{vis}}\right)^n + \frac{\left(\partial \bar{H}_{K, \tau_w}^{\text{vis}}\right)^n}{\partial \bar{W}_K^n} \cdot \delta \bar{W}_P^{n+1} \quad (2.134)$$

Since the linearization of all the components of R/H side of equation (2.118) is accomplished, equation (2.124) becomes [Kou03]:

$$\underline{D}_P^n \cdot \delta \bar{W}_P^{n+1} + \sum_{Q \in K_N(P)} \underline{Q}_{PQ}^n \cdot \delta \bar{W}_Q^{n+1} + \sum_{R \in \text{adj}(PQ), Q \in K_N(P)} \underline{C}_{adj}^n \cdot \delta \bar{W}_R^{n+1} = -\bar{R}_P^n \quad (2.135)$$

where \underline{D}_P^n and \underline{Q}_{PQ}^n are the diagonal and off-diagonal matrices including both the inviscid and viscous contributions, while $\delta \bar{W}_P^{n+1}$ and $\delta \bar{W}_Q^{n+1}$ are the computed corrections of conservative variables of nodes P and Q . \underline{C}_{adj}^n denotes the corresponding matrix for the rest nodes R neighbouring the examined edge PQ , which is used with the corresponding computed corrections $\delta \bar{W}_R^{n+1}$. The Jacobi or the Gauss-Seidel algorithm can be implemented then to re-compute the values of the conservative variables' corrections, via internal iterations, the number of which should be carefully chosen in order an acceptable convergence to be succeeded [Ven95, Kou03]. Employing the Jacobi algorithm, the corrections of conservative variables for time step $n+1$ are computed as follows

$$\delta \bar{W}_P^{n+1, k+1} = \left(\underline{D}_P^n\right)^{-1} \cdot \left(-\bar{R}_P^n - \sum_{Q \in K_N(P)} \underline{Q}_{PQ}^n \cdot \delta \bar{W}_Q^{n+1, k} - \sum_{R \in \text{adj}(PQ), Q \in K_N(P)} \underline{C}_{adj}^n \cdot \delta \bar{W}_R^{n+1, k} \right) \quad (2.136)$$

while in case the Gauss-Seidel algorithm along with the Red-Black procedure [Dec92, Ven95] is selected, as:

$$\begin{aligned} \delta \bar{W}_P^{n+1, k+1} = & \left(\underline{D}_P^n\right)^{-1} \cdot \left(-\bar{R}_P^n - \sum_{Q \in K_N(P), \text{mod}(Q, 2)=0} \underline{Q}_{PQ}^n \cdot \delta \bar{W}_Q^{n+1, k} - \sum_{Q \in K_N(P), \text{mod}(Q, 2)>0} \underline{Q}_{PQ}^n \cdot \delta \bar{W}_Q^{n+1, k+1} \right) \\ & - \left(\underline{D}_P^n\right)^{-1} \cdot \left(\sum_{R \in \text{adj}(PQ), Q \in K_N(P), \text{mod}(R, 2)=0} \underline{C}_{adj}^n \cdot \delta \bar{W}_R^{n+1, k} + \sum_{R \in \text{adj}(PQ), Q \in K_N(P), \text{mod}(R, 2)>0} \underline{C}_{adj}^n \cdot \delta \bar{W}_R^{n+1, k+1} \right) \end{aligned} \quad (2.137)$$

In the first internal iteration ($k=0$) for both methods the corrections of conservative variables are evaluated as:

$$\delta \vec{W}_p^{n+1,0} = (\underline{D}_p^n)^{-1} \cdot (-\vec{R}_p^n) \quad (2.138)$$

The incorporated turbulence models are managed in the same way, but separately to flow equations (loose coupling); the flow variables are updated firstly and then the corrections of turbulence models' variables are obtained employing the respective to (2.136) or (2.137) equations. Newton linearization is implemented to their convective fluxes as well, deriving the following components of diagonal and off-diagonal matrices:

$$\underline{B}_{t,d}^n = \underline{B}_{t,o}^n = \begin{bmatrix} un_x + vn_y + wn_z & 0 \\ 0 & un_x + vn_y + wn_z \end{bmatrix} \quad (2.139)$$

An analysis for the corresponding diffusive components is included in Appendix B. The boundary conditions' diagonal contributions are omitted, except for the inviscid ones at the outlet surfaces, defined as follows

$$\left(\vec{H}_{t,P,out}^{inv} \right)^{n+1} = \left(\vec{H}_{t,P,out}^{inv} \right)^n + \underline{L}_{t,K} \cdot \delta \vec{W}_{t,P}^{n+1} \quad (2.140)$$

where $\underline{L}_{t,K}$ is computed as:

$$\underline{L}_{t,K} = \begin{bmatrix} u_K n_{x,out} + v_K n_{y,out} + w_K n_{z,out} & 0 \\ 0 & u_K n_{x,out} + v_K n_{y,out} + w_K n_{z,out} \end{bmatrix} \quad (2.141)$$

The main difference between the turbulence models' and Navier-Stokes' PDE's is the existence of the source term in the first set, which has to be linearized as well for the implementation of the point-implicit scheme:

$$\vec{S}_{t,P}^{n+1} = \vec{S}_{t,P}^n + \underline{L}_{t,P}^S \delta \vec{W}_{t,P}^{n+1} \quad (2.142)$$

In order to reinforce the diagonal dominance of Jacobian matrix $\underline{L}_{t,P}^S$, only the negative source term components $\vec{S}_{t,P}^-$ are considered for its computation, resulting for k - ε turbulence model WF1 and WF2 in the following relations [Lar91, Koo00]

$$\vec{S}_{k\varepsilon-WF,P}^- = \begin{pmatrix} -\rho\varepsilon \\ -c_2 f_2 \rho \frac{\varepsilon^2}{k} \end{pmatrix} = \begin{pmatrix} -\frac{c_\mu f_\mu}{\mu_t} (\rho k)^2 \\ -c_2 f_2 \frac{(\rho\varepsilon)^2}{\rho k} \end{pmatrix} \quad (2.143)$$

$$\underline{L}_{k\varepsilon-WF,P}^S = \begin{pmatrix} \frac{-2f_\mu C_\mu \rho k}{\mu_t} & 0 \\ c_2 f_2 \frac{(\rho\varepsilon)^2}{(\rho k)^2} & -2c_2 f_2 \frac{\rho\varepsilon}{\rho k} \end{pmatrix} \quad (2.144)$$

while for the blended one with the one-equation method near the solid wall boundaries in:

$$\vec{S}_{k\varepsilon-1eq,P}^- = \begin{pmatrix} -\frac{c_\mu f_\mu}{\mu_t} (\rho k)^2 \\ \lambda \left(-c_2 f_2 \frac{(\rho \varepsilon)^2}{\rho k} \right) + (1-\lambda) \alpha (-\rho \varepsilon) \end{pmatrix} \quad (2.145)$$

$$\underline{L}_{k\varepsilon-1eq,P}^{S^-} = \begin{pmatrix} \frac{-2f_\mu C_\mu \rho k}{\mu_t} & 0 \\ \lambda \left(c_2 f_2 \frac{(\rho \varepsilon)^2}{(\rho k)^2} \right) & \lambda \left(-2c_2 f_2 \frac{\rho \varepsilon}{\rho k} \right) - (1-\lambda) \alpha \end{pmatrix} \quad (2.146)$$

If turbulence model $k-\omega$ or SST is selected instead, the corresponding vectors are similarly computed as [Chi96, Kim03]:

$$\vec{S}_{k\omega,P}^- = \begin{pmatrix} -\beta^* \rho \omega k \\ -\beta \rho \omega^2 \end{pmatrix} = \begin{pmatrix} -\beta^* \rho \omega k \\ -\frac{\beta (\rho \omega)^2}{\rho} \end{pmatrix} \quad (2.147)$$

$$\underline{L}_{k\omega,P}^{S^-} = \begin{pmatrix} -\beta^* \omega & -\beta^* k \\ 0 & -2\beta \omega \end{pmatrix} \Rightarrow \underline{L}_{k\omega,P}^{S^-} = \begin{pmatrix} -\beta^* \omega & 0 \\ 0 & -2\beta \omega \end{pmatrix} \quad (2.148)$$

It is common practice to omit the second term in the first row of the previous vector (2.148); in that way the completely uncoupled solution of these equations is obtained, resulting in increased stability of the methodology [Chi96]. If a negative value of a conservative variable is extracted during the iterative procedure, the value of previous iteration is restored for this node.

2.4.2. Explicit scheme

If an explicit scheme is selected instead, a four-stage Runge-Kutta (RK(4)) method [Lal88a, Kal96, Bla01, Sor03, Lyg14a, Sar14, Lyg15] is employed to solve equation (2.118) iteratively as follows

$$\begin{aligned} \vec{W}_P^{n+1,0} &= \vec{W}_P^n \\ \vec{W}_P^{n+1,k} &= \vec{W}_P^n - \alpha_k \frac{\Delta t_P}{V_P} \vec{R}(\vec{W}_P^{n+1,k-1}), \quad k=1,\dots,4 \\ \vec{W}_P^{n+1} &= \vec{W}_P^{n+1,4} \end{aligned} \quad (2.149)$$

where k is the number of current internal stage of the scheme. In this study, constants α_1 , α_2 , α_3 and α_4 of the method with values 0.11, 0.26, 0.5 and 1.0 respectively, are used attributing second-order temporal accuracy to the procedure [Bla01].

Considering the relatively low convergence rate of explicit methods, an implicit residual smoothing technique has been incorporated, enhancing it with implicit features [Bla01, Kim03] and consequently

allowing for the utilization of larger CFL numbers. The modified residual for a node P is defined as [Kim03]

$$R_p^{m+1} = \frac{R_p^0 + \varepsilon \sum_{j=1}^l R_{Q_j}^m}{1 + \varepsilon \sum_{j=1}^l 1} \quad (2.150)$$

where Q_j are the neighbouring nodes of node P and ε is a coefficient with typical values 0.5-0.8, defining the blending degree [Bla01].

The rapid change of turbulence models' source term can cause instability to the solution or even its complete failure; a remedy to this shortcoming appears to be the implicit handling of this component [Kim03]. While this treatment is a prerequisite for implicit methods, for explicit ones constitutes an additional technique, implemented by a similar Newton linearization procedure only for the source term [Bla01, Kim03]. Considering this approach, the corresponding equation (2.118) for turbulence models becomes

$$\begin{aligned} -V_P \frac{\Delta \vec{W}_{t,P}^{n+1}}{\Delta t_P} = \\ \sum_{Q \in K_N(P)} \vec{\Phi}_{t,PQ}^{inv} + \sum_{(K_{out} \in \partial C_P \cap \Gamma)} \vec{\Phi}_{t,P,o}^{inv} u \bar{t} - \sum_{Q \in K_N(P)} \vec{\Phi}_{t,PQ}^{vis} - \sum_{(K_{out} \in \partial C_P \cap \Gamma)} \vec{\Phi}_{t,P,o}^{vis} u \bar{t} (\vec{S}_{t,P}^n + \underline{L}_{t,P}^S \delta \vec{W}_{t,P}^{n+1}) V_P \end{aligned} \quad (2.151)$$

$$\begin{aligned} -V_P \frac{\Delta \vec{W}_{t,P}^{n+1}}{\Delta t_P} + \underline{L}_{t,P}^S \delta \vec{W}_{t,P}^{n+1} V_P = \\ \sum_{Q \in K_N(P)} \vec{\Phi}_{t,PQ}^{inv} + \sum_{(K_{out} \in \partial C_P \cap \Gamma)} \vec{\Phi}_{t,P,o}^{inv} u \bar{t} - \sum_{Q \in K_N(P)} \vec{\Phi}_{t,PQ}^{vis} - \sum_{(K_{out} \in \partial C_P \cap \Gamma)} \vec{\Phi}_{t,P,o}^{vis} u \bar{t} \vec{S}_{t,P}^n V_P \end{aligned} \quad (2.152)$$

$$\left(\frac{I}{\Delta t_P} - \underline{L}_{t,P}^S \right) \delta \vec{W}_{t,P}^{n+1} = -\frac{\vec{R}_{t,P}^n}{V_P} \quad (2.153)$$

where I is a unit matrix with dimensions 2×2 . Thus, the iterative step for the Runge-Kutta method is transformed as [Kim03]:

$$\vec{W}_{t,P}^{n+1,k} = \vec{W}_{t,P}^n - \frac{a_k}{V_P} \left(\frac{I}{\Delta t_P} - \underline{L}_{t,P}^S \right)^{-1} \vec{R}(\vec{W}_{t,P}^{n+1,k-1}), \quad k=1, \dots, 4 \quad (2.154)$$

Similarly to equations (2.144), (2.146) and (2.148) only the negative source term components are taken into account for the computation of the reverse matrices; for turbulence model k - ε WF1 and WF2 the latter is defined as

$$\left(\frac{I}{\Delta t_P} - \underline{L}_{k\varepsilon-WF,P}^{S-} \right)^{-1} = \begin{pmatrix} \frac{\mu_t \Delta t_P}{\mu_t + 2f_\mu C_\mu \rho k \Delta t_P} & 0 \\ 0 & \frac{k \Delta t_P}{k + 2c_2 f_2 \varepsilon \Delta t_P} \end{pmatrix} \quad (2.155)$$

while for the blended one with the one-equation method near the solid wall boundaries as:

$$\left(\frac{I}{\Delta t_p} - \underline{L}_{k\varepsilon-1eq,P}^{S^-} \right)^{-1} = \begin{pmatrix} \frac{\mu_t \Delta t_p}{\mu_t + 2f_\mu C_\mu \rho k \Delta t_p} & 0 \\ 0 & \frac{k \Delta t_p}{k + \lambda 2c_2 f_2 \varepsilon \Delta t_p + (1 - \lambda) \alpha k \Delta t_p} \end{pmatrix} \quad (2.156)$$

If turbulence model $k-\omega$ or SST is implemented instead, the corresponding vector is computed accordingly as:

$$\left(\frac{I}{\Delta t_p} - \underline{L}_{k\omega,P}^{S^-} \right)^{-1} = \begin{pmatrix} \frac{\Delta t_p}{1 + \Delta t_p \beta^* \omega} & 0 \\ 0 & \frac{\Delta t_p}{1 + \Delta t_p 2\beta \omega} \end{pmatrix} \quad (2.157)$$

"Intentionally left blank"

3. Radiative heat transfer

3.1. Governing equation

The radiative intensity I for a gray medium at any node P in position \vec{r} along a path \hat{s} through an absorbing, emitting and scattering medium is computed by radiative transfer equation (RTE) [Kim88, Rai90, Chu92, Cha93, Chu93a, Chu93b, Cha94a, Cha94b, Bae97, Bae98a, Bae98b, Kim98, Rai99, Kim01, Cha03, Kim05a, Kim05b, Kim08, Kim10, Lyg12b] as:

$$\frac{dI_p(\vec{r}, \hat{s})}{ds} = -(k_\alpha + \sigma_s) I_p(\vec{r}, \hat{s}) + k_\alpha I_b(\vec{r}) + \frac{\sigma_s}{4\pi} \int_{4\pi} I_p'(\vec{r}, \hat{s}_i') \Phi(\hat{s}, \hat{s}_i') d\omega \quad (3.1)$$

Similarly to Navier-Stokes conservation PDE's, the RTE expresses conservation of radiative energy, while its terms denote the following: The left hand side represents the rate of change of intensity in position/direction \vec{r}/\hat{s} ; the first right hand side term expresses the attenuation by absorption and scattering to other directions; the second right hand side term is the emission source term and the last one is the in-scattering integral, describing radiation obtained by scattering from all the other directions to the examined one [Hos06]. Alternatively, the time-dependent RTE can be utilized [Hun11, Lyg13b, Lyg13c, Lyg14f]

$$\frac{1}{c} \frac{dI_p(\vec{r}, \hat{s}, t)}{dt} + \frac{dI_p(\vec{r}, \hat{s}, t)}{ds} = -(k_\alpha + \sigma_s) I_p(\vec{r}, \hat{s}, t) + k_\alpha I_b(\vec{r}, t) + \frac{\sigma_s}{4\pi} \int_{4\pi} I_p'(\vec{r}, \hat{s}_i', t) \Phi(\hat{s}, \hat{s}_i') d\omega \quad (3.2)$$

which can be employed for both transient and steady-state simulations with real and pseudo-time steps respectively; in that case the left hand side terms express the rate of change of intensity in time t and in position/direction \vec{r}/\hat{s} .

Intensity I_p depends on spatial position \vec{r} , angular direction \hat{s} and time t , which correspond for a three-dimensional problem to six variables, $I(x, y, z, \theta, \varphi, t)$; the position vector is defined by the Cartesian coordinates x, y, z , while the intensity direction by the polar θ and azimuthal φ angles [Kim01, Tal05, Hun11, Lyg13c]. In case of a steady-state simulation, the blackbody intensity $I_b(\vec{r}, t)$ depends only on the spatial position \vec{r} , as its computation is based only on the temperature of the medium at the specific position [Lie02, Lyg12b]. The Scattering Phase Function (SPF) $\Phi(\hat{s}, \hat{s}_i')$ expresses the radiative energy scattered from a solid angle defined by direction \hat{s}_i' to the direction \hat{s} [Tri04a, Tri08, Lyg12b]; it is equal to unity for isotropic scattering and takes different values in various directions \hat{s}, \hat{s}_i' for anisotropic scattering, depending on the model used for its calculation (e.g., with Legendre polynomials) [Rai99, Tri04a, Hos06, Tri08, Lyg12b].

For predicting radiative heat transfer in a general enclosure by using RTE, boundary conditions for walls have also to be modelled and imposed. A wall can be fully emitting or partially emitting and reflecting, depending on the value of wall emissivity (or conversely reflectivity) [Lie02]. For a black surface, emission is considered equal to the black body one (being always diffusive), while for a gray one a reflected portion of incoming rays, which can be diffusive or specular, is added to the blackbody energy emitted from the surface. When reflection is diffusive, as in this study, there is no preferred direction for outgoing rays, while when it is fully specular each incoming ray derives a discrete outgoing reflected ray, e.g., in mirroring surfaces [Lie02, Lyg12b]. In Figure 3.1 diffusively emitting radiative rays are illustrated, while in Figure 3.2 partially diffusively and specularly reflecting radiative rays, caused by an incoming ray, are presented. In accordance with this approach, for a diffusively emitting and reflecting surface the total radiative energy $q_{w,out}$ leaving the wall can be computed as [Lie02, Lyg12b]

$$q_{w,out} = \varepsilon_w q_{w,b} + (1 - \varepsilon_w) q_{w,r} \quad (3.3)$$

where $q_{w,b}$ denotes the blackbody energy and $q_{w,r}$ the heat derived from reflection of incoming radiative rays, multiplied both with the appropriate coefficients ε_w and $1 - \varepsilon_w$, which stand for wall emissivity and reflectivity respectively ($0 \leq \varepsilon_w \leq 1$) [Hos02, Lie02, Kim05b, Lyg13c]. Boundary walls are assumed to behave as diffusive reflectors when $\varepsilon_w < 1$, while when $\varepsilon_w = 1$ they are considered to be black (only emitting, non-reflecting).

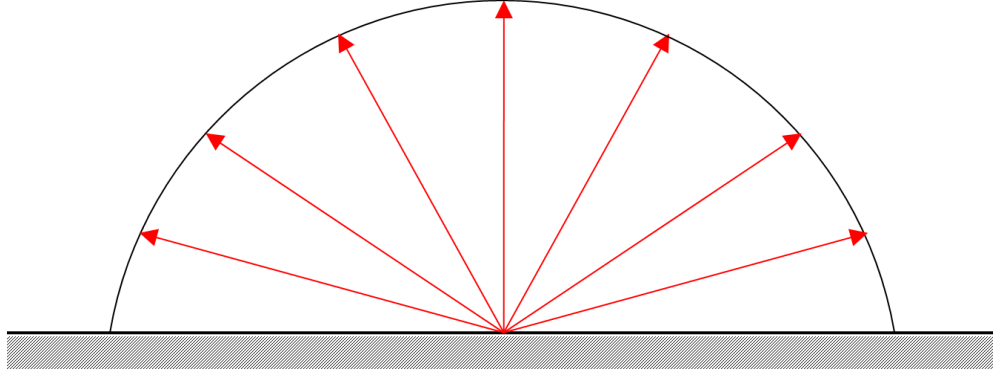


Figure 3.1: Diffusively emitting radiative rays from a boundary surface.

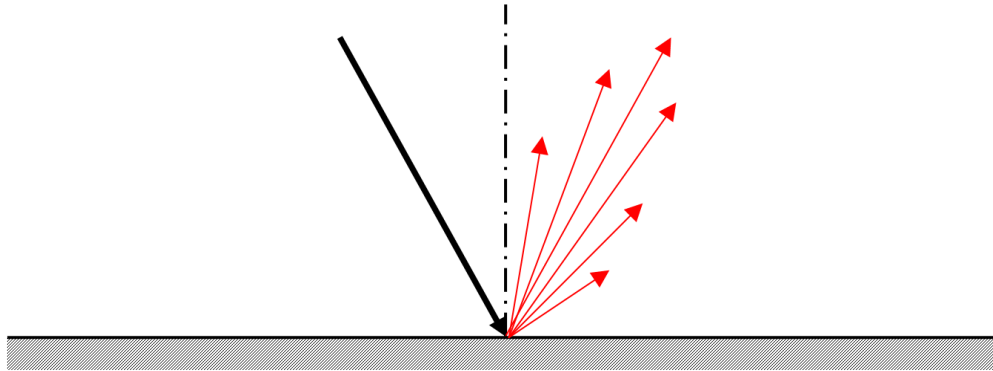


Figure 3.2: Partially diffusively and specularly reflecting radiative rays from a boundary surface.

In addition, the surfaces as well as the enclosed medium are assumed to exhibit approximately gray behaviour, neglecting wave lengths and emitting only monochromatically; although this approximation does not stand for real bodies and gases, many of them behave in such a way under certain circumstances [Lie02, Tri04b, Tal11, Lyg12b].

3.2. Spatial and angular discretization

As intensity I_P at a node P depends on its spatial position \vec{r} and the angular direction \hat{s} of radiative ray, RTE is solved for a discrete number of finite solid angles, which requires angular discretization, and is integrated for each direction over the control volume of the examined node P , which requires spatial discretization. As far as the spatial discretization is concerned, the node-centered finite-volume method, analysed in Chapter 2, is implemented on unstructured tetrahedral or hybrid grids to divide them in finite control volumes. For angular discretization the directional domain is subdivided in a finite number of solid control angles, which should exactly fill the directional domain without overlapping; although this can be achieved in many ways, it is quite common and efficient to select the solid angles associated with

areas on the surface of a *sphere* defined by lines of constant longitude and constant latitude [Rai99, Lyg12b]. In Figure 3.3 such a control angle discretization is illustrated along with a discrete solid angle \hat{s} .

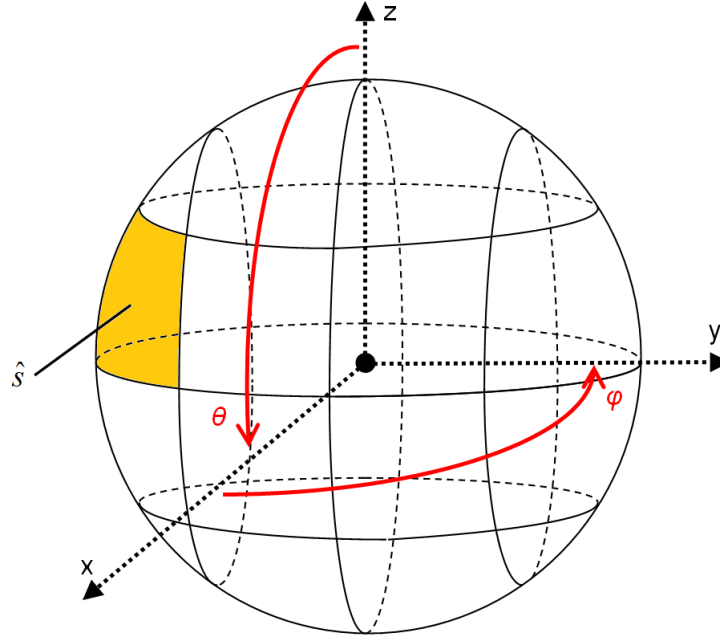


Figure 3.3: Angular discretization and a solid angle \hat{s} .

Considering the aforementioned spatial and angular discretization, the main concept is the integration of RTE over the finite control volumes and finite solid control angles; thus, the transport equation of radiative energy is redefined as an integral over the surface of each finite volume, constituted by all the faces of the examined control volume, and over every finite solid angle, composed also by a pre-defined number of sub-solid control angles. Nevertheless, it is quite common for simplicity reasons, the finite solid angles to be selected in a way that they do not need to be further subdivided [Rai99, Lyg12b]. To complete the equation set and obtain the algebraic equation for I , the radiative intensity on each face for a specific solid angle is approximated by another algebraic equation that involves control volume geometry and nodal values; the main idea is the conservation of radiative energy within a discrete direction \hat{s} , crossing the boundary face of any control volume. Since each step in the angular discretization is similar with the corresponding step in the spatial discretization, the employed method could be termed as finite-volume (in space) / finite-volume (in direction) method [Rai99, Lyg12b].

3.3. Mathematical analysis

3.3.1. Formulation of the discrete equations

Integrating equation (3.1) for a node P over the spatial finite volume V_P and over the finite solid angle Ω^{mn} (m denotes current polar angle, while n the corresponding azimuthal one) derives:

$$\int_{\Omega^{mn}} \int_{V_P} \frac{dI_P(\vec{r}, \hat{s})}{ds} dV d\omega = \int_{\Omega^{mn}} \int_{V_P} \left(-(k_\alpha + \sigma_s) I_P(\vec{r}, \hat{s}) + k_\alpha I_b(\vec{r}) + \frac{\sigma_s}{4\pi} \int_{4\pi} I_P'(\vec{r}, \hat{s}_i') \Phi(\hat{s}, \hat{s}_i') d\omega \right) dV d\omega \quad (3.4)$$

Equation (3.2) is transformed in a similar way as follows [Hun11, Lyg13c]:

$$\begin{aligned} \int_{\Omega^{mn}} \int_{V_P} \left(\frac{1}{c} \frac{dI_P(\vec{r}, \hat{s}, t)}{dt} + \frac{dI_P(\vec{r}, \hat{s}, t)}{ds} \right) dV d\omega = \\ \int_{\Omega^{mn}} \int_{V_P} \left(-(k_\alpha + \sigma_s) I_P(\vec{r}, \hat{s}, t) + k_\alpha I_b(\vec{r}, t) + \frac{\sigma_s}{4\pi} \int_{4\pi} I_P'(\vec{r}, \hat{s}_i', t) \Phi(\hat{s}, \hat{s}_i') d\omega \right) dV d\omega \end{aligned} \quad (3.5)$$

Applying divergence theorem on their left hand side and considering that in finite-volume method the magnitude of radiative intensity is taken constant over the examined control angle and control volume, the following formulations can be obtained [Kim01, Hun11, Lyg12b, Lyg13c]

$$\sum_i I_i^{mn} \Delta A_i D_{ci}^{mn} = \left[-(k_\alpha + \sigma_s) I_P^{mn} + S_R^{mn} \right] V_P \Delta \Omega^{mn} \quad (3.6)$$

$$\Delta I_P^{mn} \frac{V_P \Delta \Omega^{mn}}{c \Delta t} + \sum_i I_i^{mn} D_{ci}^{mn} \Delta A_i = \left[-(k_\alpha + \sigma_s) I_P^{mn} + S_R^{mn} \right] V_P \Delta \Omega^{mn} \quad (3.7)$$

where $\Delta \Omega^{mn}$ and D_{ci}^{mn} are the magnitude and directional weight respectively of solid control angle Ω^{mn} , while S_R^{mn} is the sum of emissive and in-scattering terms, all defined as follows [Kim01, Lyg13c]:

$$\Delta \Omega^{mn} = \int_{\theta^{m-}}^{\theta^{m+}} \int_{\varphi^{n-}}^{\varphi^{n+}} \sin \theta d\theta d\varphi \quad (3.8)$$

$$D_{ci}^{mn} = \int_{\theta^{m-}}^{\theta^{m+}} \int_{\varphi^{n-}}^{\varphi^{n+}} (\hat{s} \cdot \hat{n}_i) \sin \theta d\theta d\varphi \quad (3.9)$$

$$\hat{s} = \sin \theta \cos \varphi \hat{e}_x + \sin \theta \sin \varphi \hat{e}_y + \cos \theta \hat{e}_z \quad (3.10)$$

$$\hat{n}_i = n_x \hat{e}_x + n_y \hat{e}_y + n_z \hat{e}_z \quad (3.11)$$

$$S_R^{mn} = k_\alpha I_b + \frac{\sigma_s}{4\pi} \int_{4\pi} I_P^{mn'} \Phi(mn', mn) d\omega \quad (3.12)$$

Each solid angle Ω^{mn} is obtained by the division of 4π steradians into $N_\theta \times N_\varphi$ directions, where θ is a polar angle ranging from 0 to π and φ is an azimuthal angle ranging from 0 to 2π , while N_θ and N_φ are the numbers of pre-defined polar and azimuthal angles respectively (Figures 3.3 and 3.4). Although a control angle can be selected arbitrarily, it is usually equally divided such that $\Delta \theta^m = \theta^{m+} - \theta^{m-} = \pi / N_\theta$ and $\Delta \varphi^n = \varphi^{n+} - \varphi^{n-} = 2\pi / N_\varphi$ (a technique also adopted in this work) [Kim01, Lyg13c].

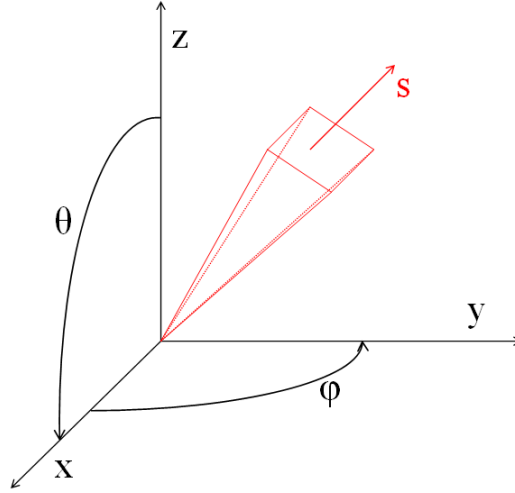


Figure 3.4: Solid angle \hat{s} .

Once the angular discretization is accomplished and limits for each control angle are defined, the directional weight D_{ci}^{mn} , which determines the direction of flow of radiative energy across the control volume face depending on its sign, is computed as [Kim01, Lyg13c]

$$\begin{aligned} D_{ci}^{mn} &= D_{ci,x}^{mn} + D_{ci,y}^{mn} + D_{ci,z}^{mn} = \int_{\theta^{m-}}^{\theta^{m+}} \int_{\varphi^{n-}}^{\varphi^{n+}} (\hat{s} \cdot \hat{n}_i) \sin \theta d\theta d\varphi = \\ &= \int_{\theta^{m-}}^{\theta^{m+}} \int_{\varphi^{n-}}^{\varphi^{n+}} (\hat{s} \cdot (n_x \hat{e}_x + n_y \hat{e}_y + n_z \hat{e}_z)) \sin \theta d\theta d\varphi \end{aligned} \quad (3.13)$$

$$\begin{aligned} D_{ci,x}^{mn} &= \int_{\theta^{m-}}^{\theta^{m+}} \int_{\varphi^{n-}}^{\varphi^{n+}} (\hat{s}_x \cdot n_x \hat{e}_x) \sin \theta d\theta d\varphi = \int_{\theta^{m-}}^{\theta^{m+}} \int_{\varphi^{n-}}^{\varphi^{n+}} n_x \cos \varphi \sin \theta \sin \theta d\theta d\varphi = \\ &= \frac{n_x}{2} (\sin \varphi^{n+} - \sin \varphi^{n-}) [(\theta^{m+} - \theta^{m-}) - (\cos \theta^{m+} \sin \theta^{m+} - \cos \theta^{m-} \sin \theta^{m-})] \end{aligned} \quad (3.14)$$

$$\begin{aligned} D_{ci,y}^{mn} &= \int_{\theta^{m-}}^{\theta^{m+}} \int_{\varphi^{n-}}^{\varphi^{n+}} (\hat{s}_y \cdot n_y \hat{e}_y) \sin \theta d\theta d\varphi = \int_{\theta^{m-}}^{\theta^{m+}} \int_{\varphi^{n-}}^{\varphi^{n+}} n_y \sin \varphi \sin \theta \sin \theta d\theta d\varphi = \\ &= \frac{n_y}{2} (\cos \varphi^{n-} - \cos \varphi^{n+}) [(\theta^{m+} - \theta^{m-}) - (\cos \theta^{m+} \sin \theta^{m+} - \cos \theta^{m-} \sin \theta^{m-})] \end{aligned} \quad (3.15)$$

$$\begin{aligned} D_{ci,z}^{mn} &= \int_{\theta^{m-}}^{\theta^{m+}} \int_{\varphi^{n-}}^{\varphi^{n+}} (\hat{s}_z \cdot n_z \hat{e}_z) \sin \theta d\theta d\varphi = \int_{\theta^{m-}}^{\theta^{m+}} \int_{\varphi^{n-}}^{\varphi^{n+}} n_z \cos \theta \sin \theta d\theta d\varphi = \\ &= \frac{n_z}{2} (\varphi^{n+} - \varphi^{n-}) (\sin^2 \theta^{m+} - \sin^2 \theta^{m-}) \end{aligned} \quad (3.16)$$

while each discrete solid angle of the *sphere* of 4π steradians is defined as [Kim01, Lyg13c]:

$$\Delta\Omega^{mn} = \int_{\theta^{m-}}^{\theta^{m+}} \int_{\varphi^{n-}}^{\varphi^{n+}} \sin\theta d\theta d\varphi = (\varphi^{n+} - \varphi^{n-}) (\cos\theta^{m-} - \cos\theta^{m+}) \quad (3.17)$$

In order to complete the equation set, the i^{th} facial intensity I_i^{mn} should be expressed in terms of the nodal intensity [Rai99, Chu93b, Kim01, Lyg13c]; several approaches have been developed so far, as the finite-volume method does not constrain this choice as long as conservation is maintained. The step scheme is utilized in this study, according to which the value of radiative intensity of a downstream face is set equal to that of the upstream node [Rai99, Kim01, Lyg13c]. It is associated with a simple upwind scheme in CFD, ensuring as such the positivity of the intensity values. In Figure 3.5 such an example is illustrated; the value of radiative intensity for a control angle s at face $ip=1$ is equal to the value of node $P=1$.

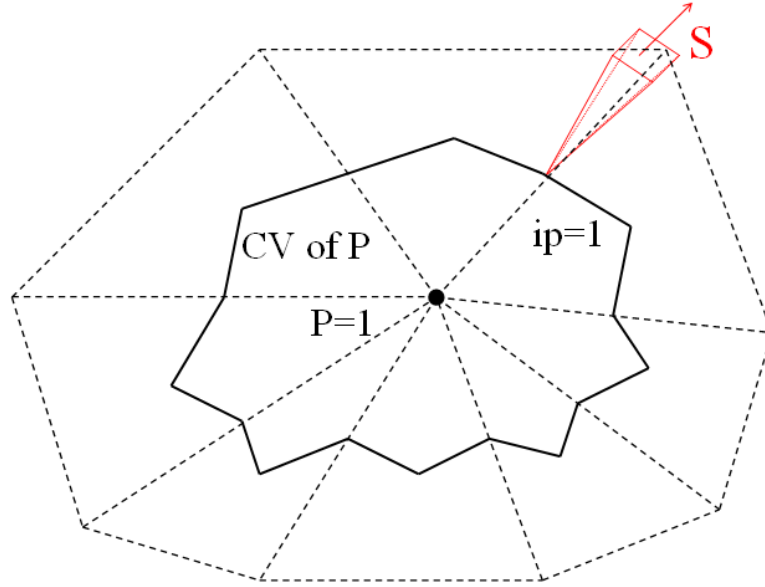


Figure 3.5: Associating facial (ip) and nodal (P) radiative intensity.

Considering the definition of the step scheme, a typical relation between facial and nodal intensity is described as [Kim01, Lyg13c]

$$I_i^{mn} D_{ci}^{mn} = I_P^{mn} D_{ci,out}^{mn} + I_Q^{mn} D_{ci,in}^{mn} \quad (3.18)$$

where I_P^{mn} and I_Q^{mn} are the values of radiative intensity at control volumes of neighbouring nodes P and Q respectively (Figure 3.6).

Depending on the manipulation of solid control angles, values of directional weights $D_{ci,out}^{mn}$ and $D_{ci,in}^{mn}$ can differ. In a structured mesh around a hexahedral geometry a control angle usually lays tangentially to its corresponding control volume; however the situation is changed for an unstructured one, as in these grids a control angle doesn't match exactly with the corresponding control volume face, becoming inevitably overlapped (overhang problem) and thus requiring appropriate manipulation for a more accurate solution [Cha94a, Kim01, Lyg13c]. Such remedies of this drawback are the bold approximation and the pixelation method.

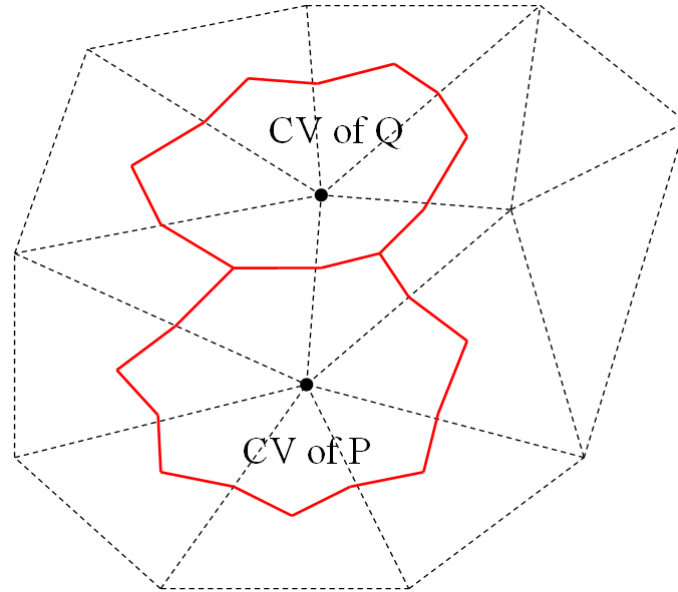


Figure 3.5: Neighbouring control volumes (CVs) of nodes P and Q .

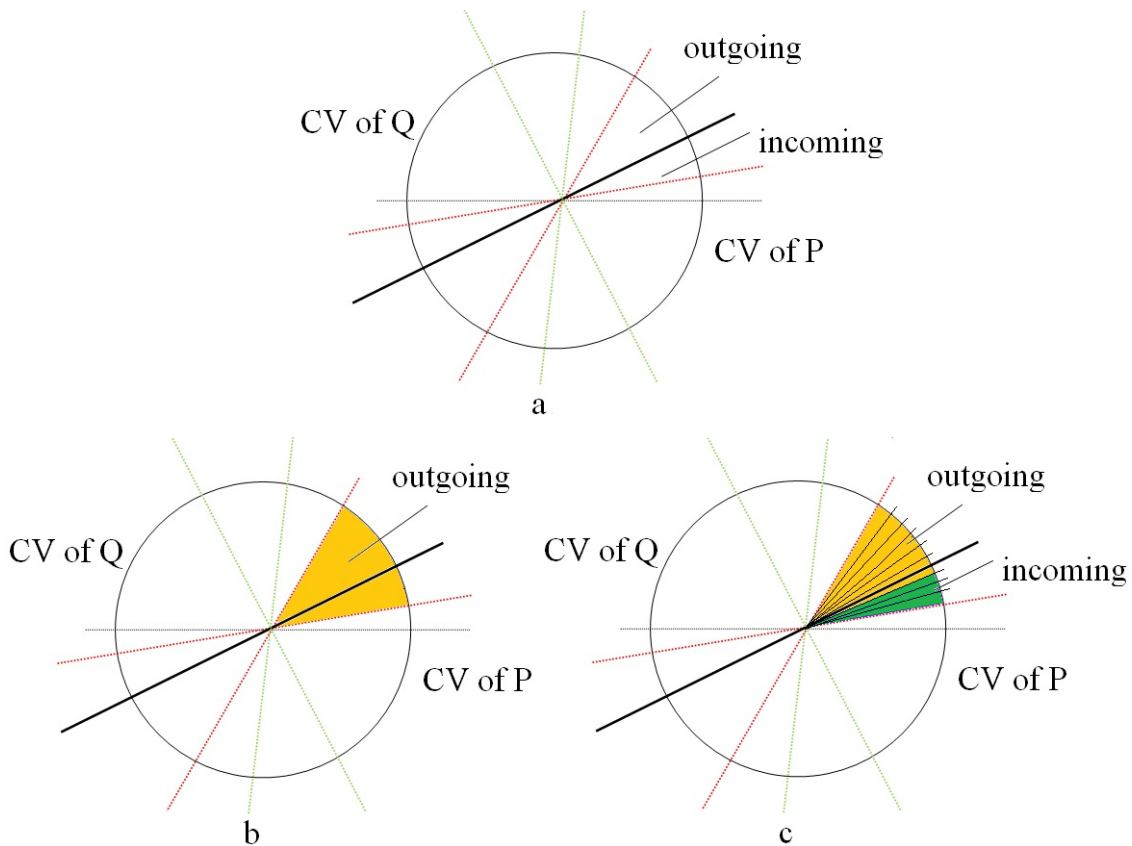


Figure 3.6: a) Schematic representation of the control angle overhang problem. b) Implementation of bold approximation. c) Implementation of pixelation method.

According to the first approach, a solid angle is assumed wholly either outgoing or incoming, depending on the sign of its directional weight [Kim01, Lyg13c]. Depending on previous illustration of Figure 3.5, Figure 3.6a presents the problem of a simultaneously incoming and outgoing solid angle in two dimensions; Figure 3.6b includes the solution obtained by the bold approximation (wholly outgoing

radiation). Mathematically this methodology derives the following expressions for directional weights [Cha94a, Kim01, Lyg13c]

$$\begin{aligned}
 D_{ci,out,bold}^{mn} &= \alpha_{ci,bold}^{mn} \left| (D_{ci}^{mn}) \right| \\
 D_{ci,in,bold}^{mn} &= -(1 - \alpha_{ci,bold}^{mn}) \left| (D_{ci}^{mn}) \right| \\
 \alpha_{ci,bold}^{mn} &= \begin{cases} 1 & \text{if } D_{ci}^{mn} \geq 0 \\ 0 & \text{if } D_{ci}^{mn} < 0 \end{cases}
 \end{aligned} \tag{3.19}$$

where $\alpha_{ci,bold}^{mn}$ is the method coefficient based on the sign of the corresponding directional weight.

Unlike this approach, pixelation method considers the division of the examined solid control angle in smaller ones; although their number $N_m \times N_n$ can be arbitrarily selected, in this work it is defined as $N_\theta \times N_\varphi$ [Mur98a, Kim01, Kim05b, Lyg13c]. For each new sub-angle the bold approximation is employed, as illustrated in Figure 3.6c, while the following expressions for directional weights are derived

$$\begin{aligned}
 D_{ci,out,pix}^{mn} &= \alpha_{ci,pix}^{mn} \sum_{m_i=1}^{N_m} \sum_{n_i=1}^{N_n} |D_{ci}^{m_i n_i}| \\
 D_{ci,in,pix}^{mn} &= -(1 - \alpha_{ci,pix}^{mn}) \sum_{m_i=1}^{N_m} \sum_{n_i=1}^{N_n} |D_{ci}^{m_i n_i}| \\
 \alpha_{ci,pix}^{mn} &= \frac{\sum_{m_i=1}^{N_m} \sum_{n_i=1}^{N_n} \max(D_{ci}^{m_i n_i}, 0)}{\sum_{m_i=1}^{N_m} \sum_{n_i=1}^{N_n} |D_{ci}^{m_i n_i}|}
 \end{aligned} \tag{3.20}$$

where the coefficient $\alpha_{ci,pix}^{mn}$ is evaluated by the sum of the positive sub-angles' directional weights, divided by the sum of the absolute values of them. Most of researchers utilize the pixelation approach only for solid angles-control volume faces, at which overhang is exhibited [Mur98a, Kim01, Kim05b]. In this work it is employed to the total number of nodes and control angles at a pre-computation stage; considering that the values of coefficients α_{ci}^{mn} , as well as of directional weights D_{ci}^{mn} , don't change during the solution of the RTE, they are calculated only once at this stage for both approaches (bold approximation and pixelation method). As a result a reduced computational cost per iteration is succeeded [Lyg13c].

The last term, which has to be computed in equations (3.6) and (3.7), is S_R^{mn} , representing the emissive (blackbody) and in-scattering contributions; it can be evaluated as follows [Kim01, Lie02, Lyg14f]

$$S_R^{mn} = k_\alpha I_b + \frac{\sigma_s}{4\pi} \int_{4\pi} I_p^{m'n'} \Phi(m'n', mn) d\omega = k_\alpha \frac{\sigma T^4}{\pi} + \frac{\sigma_s}{4\pi} \sum_{m'_i=1}^{N_\theta} \sum_{n'_i=1}^{N_\varphi} I_p^{m'_i n'_i} \bar{\Phi}(m'_i n'_i, mn) \Delta\Omega^{m'_i n'_i} \tag{3.21}$$

where the quantity $\bar{\Phi}(m'_i n'_i, mn)$ is the normalized or average SPF (Scattering Phase Function), denoting the diffusive contribution from the control angle $m'_i n'_i$ (incident angle) to the control angle mn (in-

scattering angle) [Tri04a, Tri04b, Tri08, Lyg12b]. This quantity is assumed equal to unity for isotropic scattering, while for anisotropic one it has to be further analyzed, depending on the utilized model (e.g., with Legendre polynomials), as [Cla57, Hun70, Rai99, Jim03, Tri04a, Tri08, Lyg14f]:

$$\bar{\Phi}(m'_i n'_i, mn) = \frac{\int \int_{mn, m'_i n'_i} \Phi(m'_i n'_i, mn) d\omega' d\omega}{\Delta\Omega^{mn} \Delta\Omega^{m'_i n'_i}} = \frac{\sum_{(mn)_{kk}} \sum_{(m'_i n'_i)_{ll}} \Phi((m'_i n'_i)_{ll}, (mn)_{kk}) \Delta\Omega^{(mn)_{kk}} \Delta\Omega^{(m'_i n'_i)_{ll}}}{\Delta\Omega^{mn} \Delta\Omega^{m'_i n'_i}} \quad (3.22)$$

According to the previous equation, the examined incident and in-scattering control angles are divided in smaller ones (the number of which is defined in this work as $N_\theta \times N_\phi$), approximating the summed term. Considering that the values of average SPF don't change during the solution, their evaluation can be performed before the beginning of the iterative procedure, succeeding similarly to bold approximation/pixelation method coefficients a reduced computational cost per iteration.

In this study the Legendre polynomial expansions are employed to model anisotropic scattering as follows [Jim03, Tri04a, Tri08, Lyg12b]

$$\Phi((m'_i n'_i)_{ll}, (mn)_{kk}) = \Phi(\cos \Psi) = \sum_{j=0}^N C_j P_j(\cos \Psi) \quad (3.23)$$

where Ψ is the angle between the divided incoming and scattered direction, while P_j and C_j denote the Legendre polynomials of order j and their coefficients respectively; for the forward scattering function F2 (up to order 8) these polynomials are described as [Wolfram]

$$\begin{aligned} P_0(\cos \Psi) &= 1 \\ P_1(\cos \Psi) &= \cos \Psi \\ P_2(\cos \Psi) &= 1/2(3 \cos^2 \Psi - 1) \\ P_3(\cos \Psi) &= 1/2(5 \cos^3 \Psi - 3 \cos \Psi) \\ P_4(\cos \Psi) &= 1/8(35 \cos^4 \Psi - 30 \cos^2 \Psi + 3) \\ P_5(\cos \Psi) &= 1/8(63 \cos^5 \Psi - 70 \cos^3 \Psi + 15 \cos \Psi) \\ P_6(\cos \Psi) &= 1/16(231 \cos^6 \Psi - 315 \cos^4 \Psi + 105 \cos^2 \Psi - 5) \\ P_7(\cos \Psi) &= 1/16(429 \cos^7 \Psi - 693 \cos^5 \Psi + 315 \cos^3 \Psi - 35 \cos \Psi) \\ P_8(\cos \Psi) &= 1/128(6435 \cos^8 \Psi - 12012 \cos^6 \Psi + 6930 \cos^4 \Psi - 1260 \cos^2 \Psi + 35) \end{aligned} \quad (3.24)$$

while their coefficients as [Cla57, Hun70, Ryb96, Lyg12b]:

$$\begin{array}{lll} C_0=1.0000 & C_3=0.6741 & C_6=0.0067 \\ C_1=2.0092 & C_4=0.2222 & C_7=0.0007 \\ C_2=1.5634 & C_5=0.0473 & C_8=0.0001 \end{array} \quad (3.25)$$

The cosine of angle Ψ , included in the previous equations, can be calculated as follows [Jim03, Tri04a, Tri08, Lyg12b]:

$$\begin{aligned}\cos \Psi &= \mu \mu' + (1 - \mu^2)^{1/2} (1 - \mu'^2)^{1/2} \cos(\varphi' - \varphi) \\ \mu &= \cos \theta\end{aligned}\quad (3.26)$$

Finally, utilizing the above supplementary equations, (3.6) and (3.7) can be further modified in the following ones:

$$\sum_i I_P^{mn} D_{ci,out}^{mn} \Delta A_i + \sum_i I_Q^{mn} D_{ci,in}^{mn} \Delta A_i = \left[-(k_\alpha + \sigma_s) I_P^{mn} + S_R^{mn} \right] V_P \Delta \Omega^{mn} \quad (3.27)$$

$$\Delta I_P^{mn} \frac{V_P \Delta \Omega^{mn}}{c \Delta t} + \sum_i I_P^{mn} D_{ci,out}^{mn} \Delta A_i + \sum_i I_Q^{mn} D_{ci,in}^{mn} \Delta A_i = \left[-(k_\alpha + \sigma_s) I_P^{mn} + S_R^{mn} \right] V_P \Delta \Omega^{mn} \quad (3.28)$$

3.3.2. Boundary conditions

The numerical contribution of the boundaries has also to be considered for the prediction of radiative heat transfer in a general enclosure; in this algorithm both explicit and implicit treatment of boundary conditions has been incorporated. Additionally, besides the most commonly seen boundary conditions in radiative heat transfer, i.e., opaque and diffuse surfaces, symmetry boundaries are also modelled in case of implicit manipulation.

According to the first explicit approach the boundary mesh nodes get their intensity values simply summing their blackbody intensity due to their temperature and the reflected ones by the incoming rays. As such, radiative intensity of a node P on an opaque and diffuse surface can be computed as [Rai99, Kim01, Kim05b, Lyg12b]

$$\begin{aligned}I_w^{mn} &= \varepsilon_w I_{bw} + \frac{1 - \varepsilon_w}{\pi} \sum_{\hat{s}_i \cdot \hat{n}_w < 0} I_w^{m_i n_i} |D_{cw,out}^{m_i n_i}| = \varepsilon_w I_{bw} + \frac{1 - \varepsilon_w}{\pi} \sum_{m_i=1}^{N_\theta} \sum_{n_i=1}^{N_\phi} I_P^{m_i n_i} |D_{cw,out}^{m_i n_i}| \quad \hat{s}_i \cdot \hat{n}_w < 0 \\ D_{cw,out}^{mn} &= \int_{\Delta \Omega^{mn}} (\hat{s} \cdot \hat{n}_w) d\Omega \quad \hat{s} \cdot \hat{n}_w > 0\end{aligned}\quad (3.29)$$

where I_w^{mn} is the radiative intensity coming into the computational domain, ε_w is the wall emissivity, I_{bw} is the blackbody emission of the boundary node P , while $I_w^{m_i n_i}$ and $D_{cw,out}^{mn}$ are its radiative intensity and directional weight respectively going toward the bounding wall; the second term is used in case of reflecting surfaces only.

As the common tactic of explicit treatment requires *a priori* finer computational meshes in the boundary region, the proposed algorithm was enhanced with the capability to manipulate boundary conditions in an implicit way, contributing to the flux balance of the corresponding nodes. By this approach, the step scheme is applied between the boundary node and a *ghost* node outside the computational domain; thus, the intensity of the boundary node is calculated by the time-dependent RTE (a corresponding equation is derived for non-time-dependent RTE) as [Lyg13c]

$$\begin{aligned}\Delta I_P^{mn} \frac{V_P \Delta \Omega^{mn}}{c \Delta t} + \sum_i I_P^{mn} (D_{ci,out}^{mn} \Delta A_i + D_{ci,out,w}^{mn} \Delta A_w) + \sum_i (I_Q^{mn} D_{ci,in}^{mn} \Delta A_i + I_w^{mn} D_{ci,in,w}^{mn} \Delta A_w) = \\ \left[-(k_\alpha + \sigma_s) I_P^{mn} + S_R^{mn} \right] V_P \Delta \Omega^{mn}\end{aligned}\quad (3.30)$$

where I_w^{mn} is the radiative intensity of the *ghost* node, while $D_{ci,out,w}^{mn}$ is the directional weight going toward

the boundary wall (outward to the control volume of node P) and $D_{ci,in,w}^{mn}$ the corresponding directional weight coming into the mesh (inward to the control volume of node P). For the evaluation of these weights, either equation (3.19) or (3.20) is utilized, depending on the implemented scheme (bold approximation or pixelation method). For the *ghost* node the radiative intensity is computed summing the blackbody intensity of node P and its reflected intensities going toward the boundaries as [Lyg13c]:

$$I_w^{mn} = \varepsilon_w I_{b,P} + \frac{1 - \varepsilon_w}{\pi} \sum_{m_i=1}^{N_\theta} \sum_{n_i=1}^{N_\phi} I_P^{m_i n_i} D_{ci,out,w}^{m_i n_i} \quad (3.31)$$

The mirroring boundary conditions are implemented in the same implicit way, while the radiative intensity of the boundary surface I_w^{mn} is evaluated such as the net heat flux to become zero at the symmetry plane. Thus, the intensities of a boundary node P , I_P^{mn} going out of a surface are placed as intensities I_w^{mn} of the same node coming into this surface in the mirroring direction (Figure 3.7). This attitude can be resembled with this of a specularly reflecting wall [Liu00, Lie02, Lyg13c]. The utilization of symmetry surfaces is common practice in CFD, as it allows for the use of smaller domains with significantly reduced numbers of Degrees of Freedom (DoF's) and consequently for reduced computational effort and memory requirements. A drawback of this methodology is that the mirroring direction has to be selected to coincide with a discrete one; in order to implement it on more directions, an interpolation technique should be employed [Liu00]. Nevertheless, as the symmetry planes are usually designed parallel to the axes' planes (as in this work), this shortcoming appears to be relatively negligible [Lyg13c].

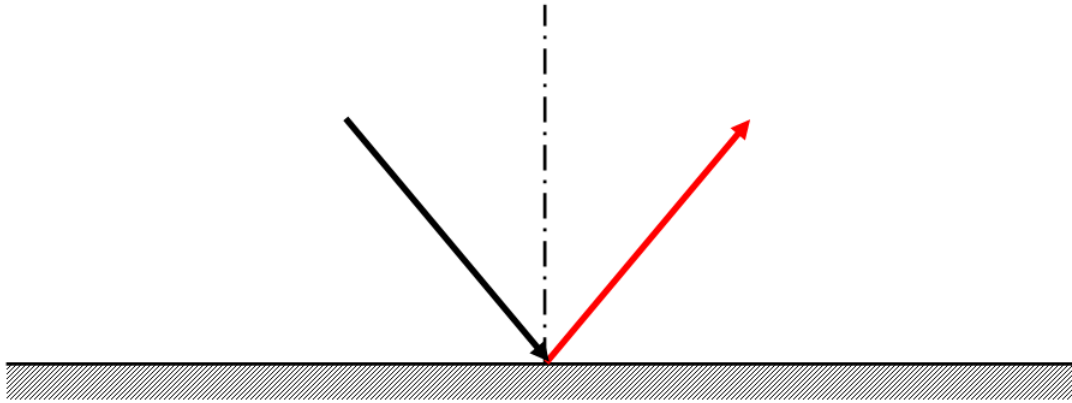


Figure 3.7: Symmetry boundary condition on surface.

3.3.3. High-order accurate spatial scheme

For a first-order accurate spatial scheme the left and right states of an edge PQ (interface between adjacent control volumes) are approximated by the values of radiative intensity at the end-points P and Q respectively. For a higher-order one, these states are reconstructed with the Taylor series expansions taking into account the values of more mesh nodes when computing the numerical fluxes similarly to CFD. In this study the higher-order methodology is based on the MUSCL reconstruction of radiative intensity values, using additionally a slope limiter to control the total variation of the reconstructed field [VanA82, Swe84, Bar92, Lyg13c]. As such, radiative intensity I^{mn} at the left and right states of midpoint of an edge PQ is approximated as [Bar92, Lyg13b]

$$\begin{aligned} (I_{PQ}^{mn})^L &= I_P^{mn} + \frac{1}{2}(\nabla I^{mn})_P \cdot \vec{r}_{PQ} \\ (I_{PQ}^{mn})^R &= I_Q^{mn} - \frac{1}{2}(\nabla I^{mn})_Q \cdot \vec{r}_{PQ} \end{aligned} \quad (3.32)$$

where, the first right hand side terms are the intensities of left and right nodes respectively, while \vec{r}_{PQ} is the vector connecting them; as such, the step scheme, defined in equation (3.18), can be described as follows:

$$I_i^{mn} D_{ci}^{mn} = (I_{PQ}^{mn})^L D_{ci,out}^{mn} + (I_{PQ}^{mn})^R D_{ci,in}^{mn} \quad (3.33)$$

In order to implement the aforementioned second-order accurate spatial scheme, the extrapolation gradients $(\nabla I^{mn})_P$ and $(\nabla I^{mn})_Q$ have to be evaluated at nodes P and Q respectively, applying the Green-Gauss linear representation method as [Bar92, Lyg13c]

$$(\nabla I^{mn})_P = \frac{1}{V_P} \left(\sum_{i, \text{edge } PQ} \frac{1}{2} (I_P^{mn} + I_Q^{mn}) \hat{n}_i \Delta A_i + \sum_{i, P\text{-wall}} I_P^{mn} \cdot \hat{n}_{i,w} \Delta A_w \right) \quad (3.34)$$

where V_P is the volume of the control volume of node P and Q is its neighbouring node; the second summing term is added only in case node P is a boundary node.

In addition, two slope limiters have been incorporated in the present algorithm, the Van Albada-Van Leer [VanA82] and the Min-mod [Swe84], in order the total variation in the reconstructed field to be diminished. As mentioned in previous Chapter, the first one is actually an expansion of MUSCL methodology, preserving monotonicity between the values at end-points, while the second one is characterized by the smoothing of solution, choosing always the slope with the smallest magnitude. Their mathematical formulation is based on centered $(\nabla I_{PQ}^{mn})^c$ and upwind $(\nabla I_P^{mn})^u, (\nabla I_Q^{mn})^u$ gradients, defined as follows [VanA82, Swe84, Lyg13c]:

$$\begin{aligned} (\nabla I_{PQ}^{mn})^c \cdot \vec{r}_{PQ} &= I_Q^{mn} - I_P^{mn} \\ (\nabla I_P^{mn})^u &= 2(\nabla I_P^{mn}) - (\nabla I_{PQ}^{mn})^c \\ (\nabla I_Q^{mn})^u &= 2(\nabla I_Q^{mn}) - (\nabla I_{PQ}^{mn})^c \end{aligned} \quad (3.35)$$

Since these gradients are computed, equation (3.32) including additionally a non-linear slope limiter is modified as [VanA82, Swe84, Lyg13c]

$$\begin{aligned} (I_{PQ}^{mn})^L &= I_P^{mn} + \frac{1}{2} \cdot X \left((\nabla I_P^{mn})^u \cdot \vec{r}_{PQ}, (\nabla I_{PQ}^{mn})^c \cdot \vec{r}_{PQ} \right) \\ (I_{PQ}^{mn})^R &= I_Q^{mn} - \frac{1}{2} \cdot X \left((\nabla I_Q^{mn})^u \cdot \vec{r}_{PQ}, (\nabla I_{PQ}^{mn})^c \cdot \vec{r}_{PQ} \right) \end{aligned} \quad (3.36)$$

where X is the limiter function, defined in case of Van Albada-Van Leer as [VanA82]

$$X(a, b) = \begin{cases} \frac{(a^2 + e)b + (b^2 + e)a}{a^2 + b^2 + 2e} & ab > 0 \\ 0 & ab \leq 0 \end{cases} \quad (3.37)$$

where e is a very small number with a typical value of $1.0E-16$, used to prevent division by zero. Similarly the Min-mod function is described as follows [Swe84]:

$$X(a, b) = \begin{cases} a & \text{if } |a| < |b| \text{ and } ab > 0 \\ b & \text{if } |b| < |a| \text{ and } ab > 0 \\ 0 & \text{if } ab \leq 0 \end{cases} \quad (3.38)$$

In case a negative unphysical value of radiative intensity is derived by the aforementioned method (usually in the initial iterations), it is reverted to its non-reconstructed value (first-order accurate scheme). Based on the previous supplementary equations for left and right intensity values, equation (3.30) (a corresponding equation is derived for non-time-dependent RTE) is transformed as:

$$\begin{aligned} \Delta I_P^{mn} \frac{V_P \Delta \Omega^{mn}}{c \Delta t} &= \left[-(k_\alpha + \sigma_s) I_P^{mn} + S_R^{mn} \right] V_P \Delta \Omega^{mn} \\ - \sum_i \left((I_{PQ}^{mn})^L D_{ci,out}^{mn} \Delta A_i + I_P^{mn} D_{ci,out,w}^{mn} \Delta A_w \right) &- \sum_i \left((I_{PQ}^{mn})^R D_{ci,in}^{mn} \Delta A_i + I_w^{mn} D_{ci,in,w}^{mn} \Delta A_w \right) = R_P^{mn} \end{aligned} \quad (3.39)$$

3.3.4. Iterative solution

For the solution of steady (non time-dependent) RTE, equation (3.27) is modified in the following formulation [Chu92, Cha94b, Bae98b, Rai99, Kim01, Lyg12b]:

$$\sum_i I_P^{mn} D_{ci,out}^{mn} \Delta A_i + (k_\alpha + \sigma_s) I_P^{mn} V_P \Delta \Omega^{mn} = S_R^{mn} V_P \Delta \Omega^{mn} - \sum_i I_Q^{mn} D_{ci,in}^{mn} \Delta A_i \quad (3.40)$$

$$I_P^{mn} \left(\sum_i D_{ci,out}^{mn} \Delta A_i + (k_\alpha + \sigma_s) V_P \Delta \Omega^{mn} \right) = S_R^{mn} V_P \Delta \Omega^{mn} - \sum_i I_Q^{mn} D_{ci,in}^{mn} \Delta A_i \quad (3.41)$$

$$I_P^{mn} = \frac{S_R^{mn} V_P \Delta \Omega^{mn} - \sum_i I_Q^{mn} D_{ci,in}^{mn} \Delta A_i}{\left(\sum_i D_{ci,out}^{mn} \Delta A_i + (k_\alpha + \sigma_s) V_P \Delta \Omega^{mn} \right)} \quad (3.42)$$

The previous relation is implemented iteratively, until either a desired radiative intensity residual convergence or a desired number of iterations is achieved.

In case of time-dependent RTE (equation (3.39)), an explicit temporal scheme with a four-stage Runge-Kutta (RK(4)) method [Lal88a, Bla01] is employed instead for time integration and solution as [Lyg13c]

$$\Delta I_P^{mn} \frac{V_P \Delta \Omega^{mn}}{c \Delta t_p} = R_P^{mn} \quad (3.43)$$

$$\begin{aligned}
 \left(I_P^{mn}\right)^{l+1,0} &= \left(I_P^{mn}\right)^l \\
 \left(I_P^{mn}\right)^{l+1,k} &= \left(I_P^{mn}\right)^l + \alpha_k \frac{c\Delta t_P}{V_P \Delta \Omega^{mn}} R_P^{mn} \left(\left(I_P^{mn}\right)^{l+1,k-1}\right), \quad k=1,\dots,4 \\
 \left(I_P^{mn}\right)^{l+1} &= \left(I_P^{mn}\right)^{l+1,4}
 \end{aligned} \tag{3.44}$$

where k and l are the numbers of current internal (Runge-Kutta method) and external iterations respectively, while R_P^{mn} is the total flux balance of node P for the solid control angle Ω^{mn} . The employed in this study constants of the method α_k attribute the procedure with a second-order accuracy in time [Lal88a]; their values are defined similarly to the CFD algorithm (Chapter 2) [Bla01, Lyg13c].

Δt_P is the local pseudo-time step for node P [Fit99, Bla01], based on the length $\alpha_{\min l \text{ edge}, P}$ of the shortest edge of the grid connected to node P and defined for a steady-state simulation as [Hun11, Lyg13c]:

$$c\Delta t_P \leq \frac{1}{2} \min(\alpha_{\min l \text{ edge}, P}) \tag{3.45}$$

The local time stepping technique is a common practice in CFD, enhancing each control volume with the maximum permissible time step and consequently accelerating the convergence to the steady-state solution [Bla01, Lyg13c]. Nevertheless, it cannot be applied to an unsteady-transient simulation, in which a global time step should be used instead, evaluated as the smallest of the local time steps of all the nodes of the examined mesh.

For both equations (with or not time-dependency), if a negative unphysical value of intensity is obtained (usually in the initial iterations along with a higher-order accurate spatial scheme), a fix-up procedure is employed, setting this value equal to zero [Cap10, Lyg14f].

4. Parallel computation strategy

4.1. Domain decomposition

As mentioned in the Introduction, the incorporated SPMD (Single Program-Multiple Data) parallelization strategy, is based on the domain decomposition approach. According to this methodology the computational domain is divided into a number of sub-domains equal to the number of available processors/processes [Ven95, Smi96, Lan96, Tai03, Kal05a, Lyg12b, Sar15].

The whole procedure begins with the implementation of METIS, which is an acclaimed multi-level graph partitioning application used for dividing unstructured meshes; it is capable of providing high quality partitions with less CPU time workload balancing [METIS, Kar99]. METIS derives a number of sets of non-common nodes, the *core* nodes, depending on the number of available computational processors/processes. Based on these sets (sub-domains), a single process is assigned to [Lyg12b, Lyg14a, Lyg15, Sar15]:

a) Divide the elements of the initial grid accordingly; each element is assigned to the partition including its first node.

b) Identify the elements situated at the interface between two adjacent sub-domains; these elements are included in both partitions, despite remaining incomplete.

c) Identify and add as extra nodes the missing ones (named *ghost* nodes), completing the deficient elements, which subsequently form an overlapping region included to both neighbouring partitions; it is this region that allows for interaction between the corresponding sub-grids via the exchange of data (flow, turbulence and radiative heat transfer model variables and nodal gradients) with the MPI (Message Passing Interface) communication protocol [Tai03, MPI].

d) Distribute to other processors/processes the corresponding new data, including overlapping region nodes and elements, coordinates of *core* and *ghost* nodes, initial values of conservative and primitive variables, etc.

Figure 4.1 presents the initial division (by METIS) of a two-dimensional triangular unstructured grid, as well as the same mesh including the constructed overlapping region (coloured orange). Similarly, in Figure 4.2, the decomposition of a three-dimensional tetrahedral unstructured mesh in two sub-domains for parallel process is illustrated.

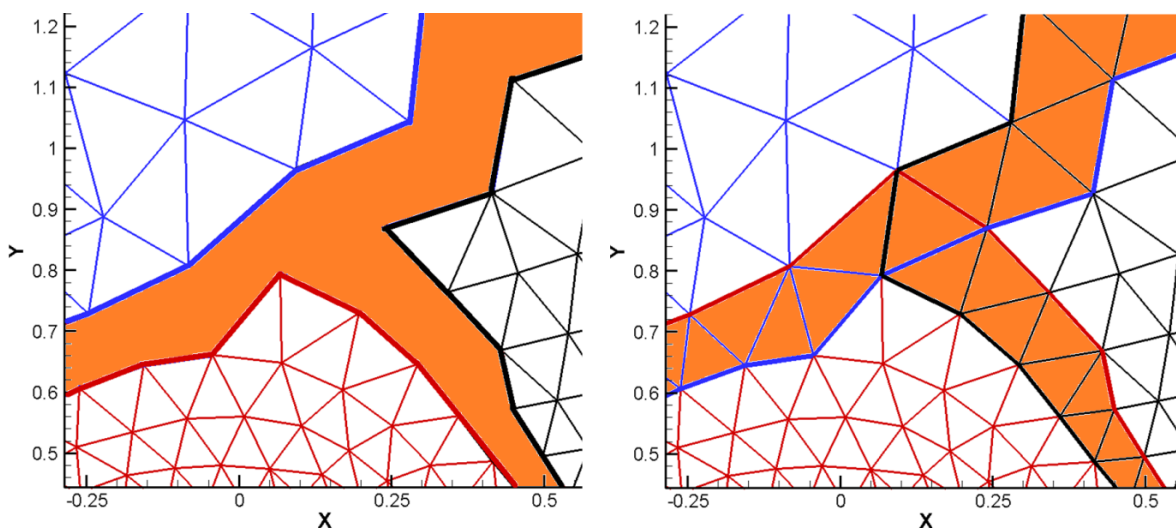


Figure 4.1: Partitioning of a two-dimensional triangular unstructured grid, before (left) and after (right) the construction of the overlapping region (coloured orange).

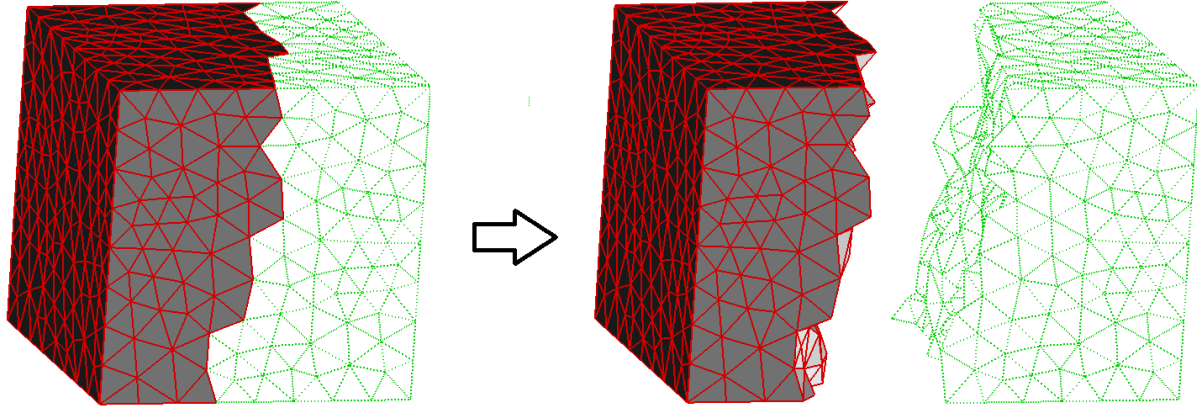


Figure 4.2: Decomposition of a three-dimensional tetrahedral unstructured mesh in two sub-domains.

4.2. Data exchange

The interaction between the neighbouring partitions is succeeded via the exchange of data (model variables and nodal gradients) of the nodes constituting the overlapping area; these shared nodes either send (*core*) or receive (*ghost*) data to or from the corresponding identical nodes of the adjacent sub-domains [Ven95, Lan96, Tai03, Lyg14a, Lyg15, Sar15]. While for *core* nodes the appropriate governing equations are normally implemented, no calculation is performed for the *ghost* ones; their values are explicitly received by their corresponding *core* nodes at the neighbouring processors/processes, ensuring in that way the consistency of the solution across the virtual internal boundaries. Although this strategy might seem memory consuming at first glance it appears to be quite efficient, as for an explicit iterative scheme it is not the memory usage that should concern, but the communication cost between the adjacent partitions [Kal05a, Lyg12b, Sar15].

Since partitioning by METIS application is accomplished, each available processor/process is assigned a unique rank; the one with a rank equal to zero is identified as the master processor/process, which will perform the tasks analyzed in the previous section. Additional information, stored for each processor/process in simple array data structures, is required to supplement the communication procedure [Lyg12b, Lyg14a, Lyg15, Sar15]; more specifically, these data structures are the following:

- a) ***nneighbours***: an integer representing the number of adjacent processors/processes.
- b) ***ineighbours(n)***: an integer array with size *nneighbours*, including the ranks of neighbouring sub-domains.
- c) ***nreceivenodes(n)***: an integer array with size *nneighbours*, which includes the number of *ghost* nodes receiving information from each of the adjacent partitions.
- d) ***nsendnodes(n)***: an integer array with size *nneighbours*, which includes the number of *core* nodes sending information to each of the adjacent partitions.
- e) ***ireceivenodes(n,m)***: a matrix with *nneighbours* rows; each row, constituted by *nreceivenodes(n)* integers, includes the indexes of the *ghost* nodes receiving data from the neighbouring sub-domain *n*.
- f) ***isendnodes(n,m)***: a matrix with *nneighbours* rows; each row, constituted by *nsendnodes(n)* integers, includes the indexes of the *core* nodes sending data to the neighbouring sub-domain *n*.

In order to ensure that data sent from a specific *core* node are received by its corresponding *ghost* one at the adjacent partition, master processor/process fills in the same order the *isendnodes(n,m)* and *ireceivenodes(n,m)* matrices. Figure 4.3 includes a schematic representation of the *core* and *ghost* nodes of a sub-domain with rank 0; according to the previous definitions the additional data structures for this partition are the following:

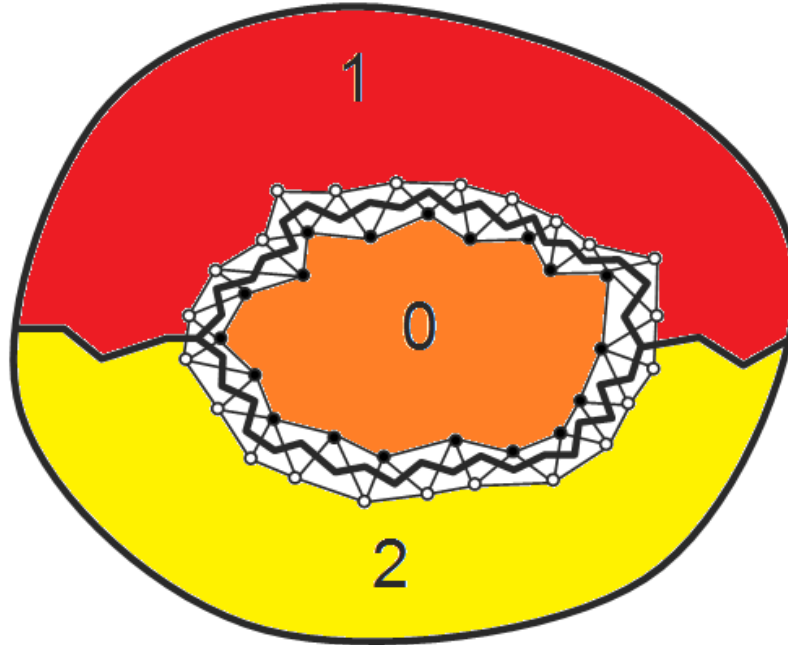


Figure 4.3: Schematic representation of the core and ghost nodes of processor/process with rank 0.

- **nneighbours** = 2
- **ineighbours(1)** = 1
- **ineighbours(2)** = 2
- **nreceivenodes(1)** = 12
- **nreceivenodes(2)** = 11
- **nsendnodes(1)** = 11
- **nsendnodes(2)** = 10

As mentioned above, the exchange of data is based on the MPI communication protocol libraries, which enable point-to-point communication as well as collective communication procedure [MPI]. Nevertheless, considering the unique mapping of corresponding *core* and *ghost* nodes of neighbouring partitions, a blocking point-to-point communication was selected for the proposed algorithm. This communication procedure is performed at the end of each step of the iterative procedure (internal and external), while to assure synchronicity of processors/processes the MPI_BARRIER function of the MPI library is used, pausing the faster processes until the slowest one executes this command [MPI]. In case a second-order accurate spatial scheme is employed, the nodal gradients have to be exchanged as well, since they are computed; the same stands for the conservative variables' corrections if an implicit iterative scheme is selected. The whole procedure is presented in pseudo-code formulation as

Initialization of simulation by the master process

Initialization of parallelization (domain decomposition, overlapping region construction, etc.) by the master process

Distribution of data to the rest processes

Iterative procedure

Do $i = 1$ to max iterations

Do $j = 1$ to iterative method steps

Perform relaxation of governing equations

MPI_BARRIER

Exchanging data

```
MPI_BARRIER  
End  
End
```

while the exchange procedure as:

```
Do i = 0 to number of processes - 1  
  If (process rank = i) then  
    Do j = 1 to nneighbours  
      Send data of isendnodes(j,m)  
    End  
  Else  
    Receive data from process i and store them in ireiceivenodes(i,m) (if applicable)  
  End  
End
```

Employing the aforementioned methodology, equally accurate but more computationally efficient solutions compared to serial schemes, can be obtained [Ven95, Lan96]; its contribution appears especially valuable in large-scale problems, such as flows over aircrafts usually requiring grids with millions or tens of millions nodes, which consequently result to excessive time-consuming simulations [Lyg12b, Lyg14a, Lyg15, Sar15]. Besides the validation of obtained solutions' accuracy in Chapters 7 and 8, the proposed parallelization strategy is evaluated against benchmark test cases, such as this considering turbulent flow over the NASA Common Research Model (CRM).

5. Agglomeration multigrid methods

5.1. Spatial agglomeration multigrid scheme

5.1.1. Spatial agglomeration strategy

The first issue, which has to be defined for the implementation of the agglomeration multigrid method, is the generation of the coarser grids, i.e., the agglomeration strategy [Mav97, Mav98, Car00, Bla01, Nis10, Nis11, Nis13, Lyg14c, Lyg14d, Lyg14f, Lyg14g]. The fusion of the control volumes in this study is performed on a topology-preserving framework at each partition, resembling the advancing front technique [Nis11]; both isotropic and directional agglomeration methodologies (either selected by the user) have been developed to be employed on tetrahedral and hybrid grids respectively [Bla01, Mav97, Nis11, Lyg14c].

The procedure is limited though by pre-defined rules in order consistency between the solution of the initial finest grid and those of the coarser ones to be preserved at external (especially at the boundary layer region) [Nis11, Lyg14c] and internal boundaries (between the sub-domains constructed for parallel implementation of the algorithm). More specifically, the following constraints are employed [Lyg14c, Lyg14d, Lyg14f]:

a) The control volume of a node located at the internal region of a grid can be fused only with its neighbouring, also internal, nodes' control volumes, to construct a new virtual *supernode* [Nis10, Nis11, Nis13].

b) In the same way, a boundary node can be agglomerated only with its adjacent boundary nodes of the same type [Nis11], e.g., a node positioned at a solid wall surface (or at a diffusively reflecting surface in case of radiative heat transfer simulation) can be associated only with another node belonging to the same surface.

c) A node belonging to two or more boundary-condition-type closures (e.g., a node belonging simultaneously to inlet and solid wall surfaces) isn't agglomerated and remains singleton at the next multigrid level [Nis11]. Nevertheless, the nodes belonging to two boundary-condition-type closures, with the one being a symmetry surface, are excluded from this limitation; they can be agglomerated with an adjacent node having the same two boundary condition types.

d) A boundary node belonging to two or more boundary slope discontinuities isn't agglomerated and remains singleton at the next multigrid level [Nis11]; examples of such nodes are those situated among three different surfaces or among two different walls of the same type forming a 90° angle. In this algorithm the limit set to restrict the fusion procedure is equal to 30° .

e) A boundary node can be merged with its adjacent one, only if their control areas unit normal vectors form an angle less than 10° .

f) Although the control volumes of *core* nodes at overlapping region (sending information to neighbouring partitions) [Ven95, Lyg12b] are considered for fusion, these of *ghost* ones (receiving information from adjacent sub-domains) aren't merged during each level main agglomeration procedure; at the end of each level they are fused accordingly to the agglomeration of their corresponding *core* nodes at neighbouring partitions, creating actually virtual *ghost supernodes*.

g) In simulations, concerning turbulent flows and consequently relatively fine hybrid grids with highly stretched elements in boundary layer region, a maximum threshold of merged nodes to a supernode (eight for internal and four for boundary) is imposed to preserve even more the topology of the initial finest grid [Nis11].

Considering the aforementioned limitations, the isotropic and directional agglomeration methodologies are analyzed in the next sections.

5.1.1.1. Isotropic agglomeration

In case of a tetrahedral grid, the isotropic spatial agglomeration procedure is performed in the following sequence [Lyg14c, Lyg14d, Lyg14f]:

1) The nodes of the initial mesh, prohibited for agglomeration due to pre-mentioned constraints (c) and (d), are identified and marked.

2) The nodes, marked for confined "agglomeration", remain singletons (simply transferred to the next level). The derived *supernodes* are marked in the same way for prohibited fusion at the next level agglomeration.

3) The solid wall boundary nodes of each sub-domain are selected to create the list of the so-called *seed* nodes [Sor03, Nis11]. In case a partition includes no such boundary nodes, the *core* nodes of the overlapping region are assigned as *seed* ones.

4) At this step the main agglomeration procedure begins by looping over the nodes of the aforementioned list and examining the eligibility for fusion of all their adjacent nodes (due to the pre-mentioned constraints or in case they are already fused to another *supernode*); if no limitation is identified, their control volumes are added to the control volume of the *seed* node, constructing the control volume of a new *supernode*. Figure 5.1 illustrates the schematic representation of the generation of a *supernode* via agglomeration at the prismatic region of a hybrid grid. In case no appropriate for fusion (already agglomerated or constrained) adjacent node is identified, further examination is performed; the seed node's neighbouring *supernodes* are examined, in order to include it. If more than one *supernodes* are identified as eligible for this merging, the one with the lower number of included nodes is selected, while if no suitable *supernode* is found, the seed node remains as singleton to the next multigrid level. The procedure is assumed to be complete when all the seed nodes are agglomerated or become singletons.

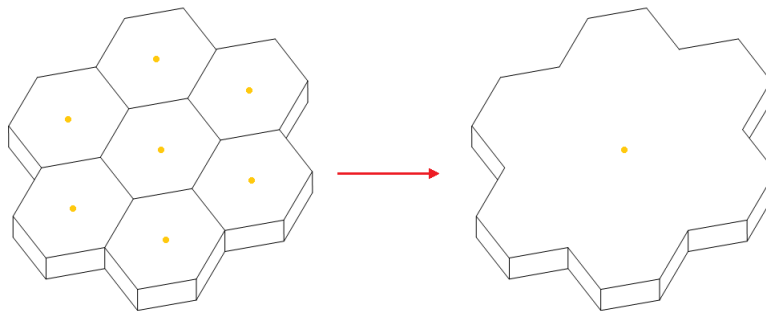


Figure 5.1: Construction of a *supernode* via agglomeration at a prismatic region.

5) A new list of *seed* nodes is constructed, including the nodes whose control volumes have been touched by the agglomeration front, which are actually the neighbouring, non-agglomerated yet, nodes of the already fused ones [Han02]. In this list a priority hierarchy is applied, taking into account the number of times a node is touched by the agglomeration front, as well as the number of its adjacent control cells. Therefore, the set of nodes which have the maximum touch number will be initially examined for agglomeration, while for the same set the procedure begins from the node with the minimum number of adjacent nodes [Han02].

6) Steps 4 and 5 are repeated until all the nodes of each partition (except the *ghost* ones) are fused or assigned as singletons. In Figure 5.2 the progress of the agglomeration front as well as the merging procedure are presented schematically in a part of two-dimensional triangular grid, beginning from the solid wall boundary surface and extending then to the internal area of the computational domain.

7) The generated unconstrained singleton *supernodes* are examined once more to be merged with their adjacent ones, according to the procedure described in step 4.

8) A loop over the derived *supernodes* is performed in order the completely surrounded by other ones to be identified (they are connected with only one *supernode*); in such case they are agglomerated with their surrounding ones [Sor03].

9) The *ghost* nodes at the overlapping region of each partition (coloured black in Figure 5.3) are agglomerated or become singletons according to their corresponding *core* nodes' fusion at the adjacent sub-grids. Therefore, virtual *ghost supernodes* are actually generated, as the number of their merged nodes may differ from this of their overlapping *core supernodes*; such a two-dimensional example is illustrated in Figure 5.4, where it is obvious that the virtual *ghost supernodes* do not coincide with their overlapping *core* ones. Nevertheless, the *ghost supernodes* get the correct data (volume, coordinates, etc.) of their overlapping *core* ones by exchange of information.

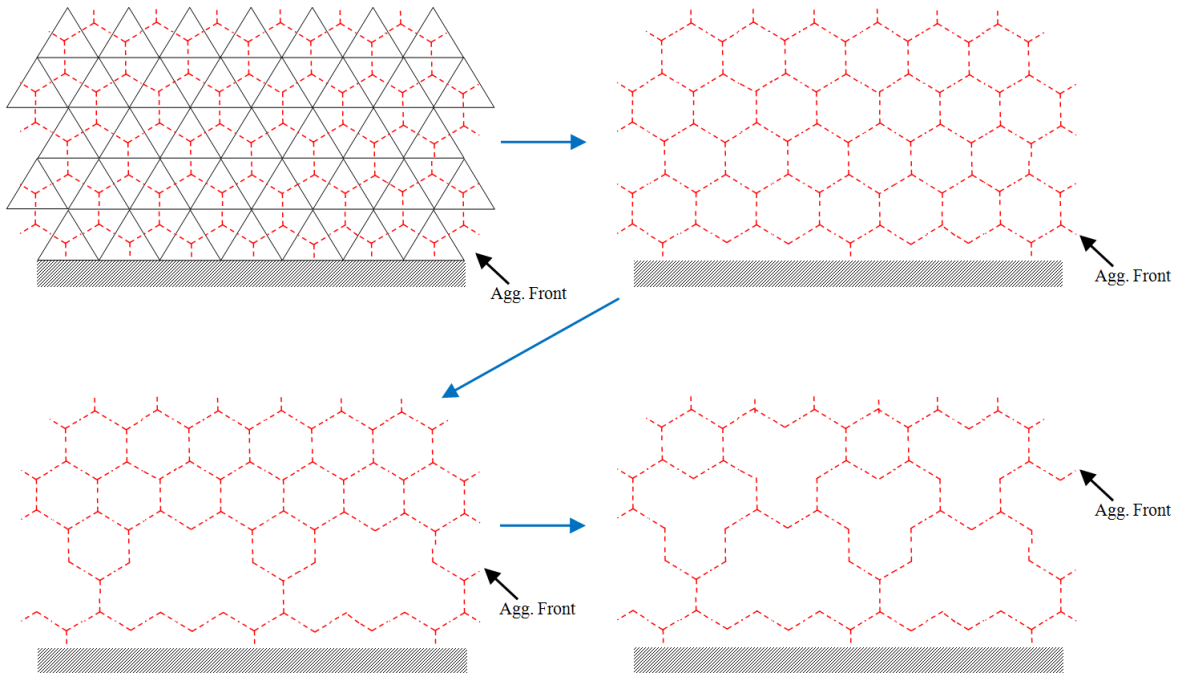


Figure 5.2: Progress of the agglomeration front and fusion procedure in a part of a two-dimensional triangular grid.

10) The new virtual *superedges*, connecting the *supernodes* obtained by the previous steps, are constructed. At each *super* control volume the internal edges are deleted, while the external ones are used for the evaluation of the new vectors \vec{n}_{PQ} at the interface areas with its neighbouring *super* control volumes; such vectors are simply evaluated by summing the vectors of the corresponding surfaces interfacing two adjacent *super* control volumes [Sor03]. As unstructured grids are utilized, special attention is required in order to avoid summing vectors with opposite signs and subsequently opposite directions (inward and outward to the examined control volume).

11) At the final step of the main spatial agglomeration procedure the *supernodes* that remained singletons as well as their neighbouring ones are identified and marked, in order to become also singletons at the next level (coarser) grid. In that way the generation of adjacent control volumes with large differences in size is avoided, and consequently a possible harsh irregularity at next agglomeration levels is alleviated.

12) The whole procedure is repeated in case an even coarser mesh is required. Depending on the test cases encountered in this study, a number of three or four spatial multigrid levels has been identified to be adequate.

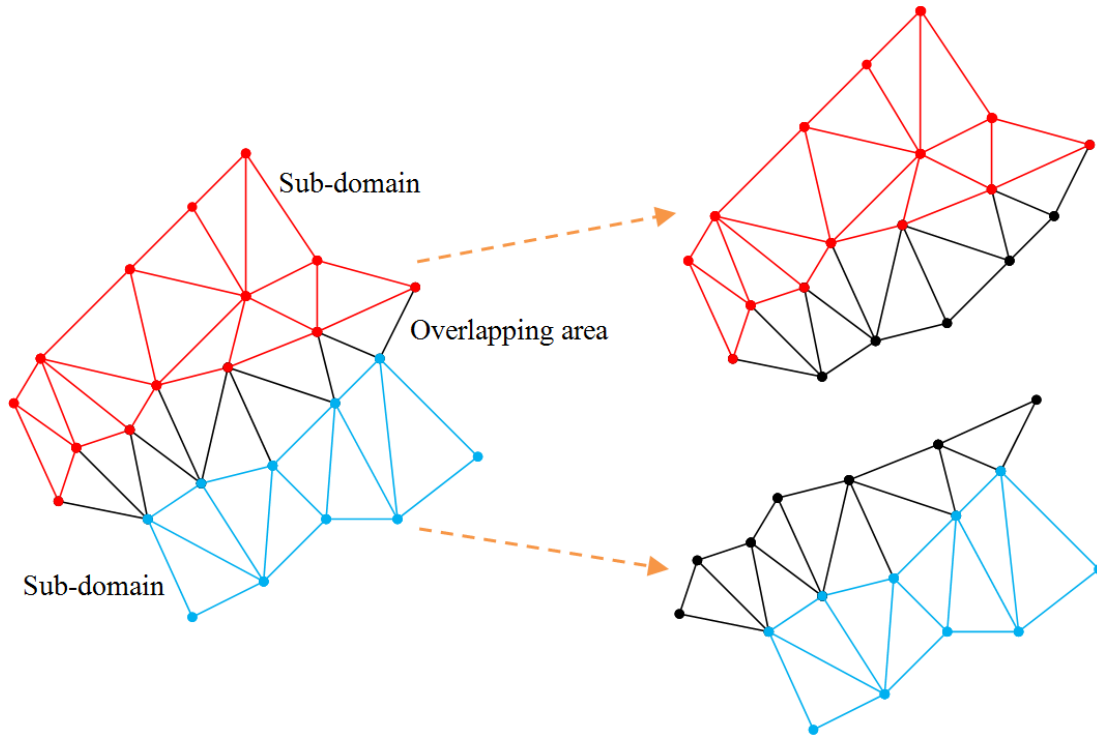


Figure 5.3: Sub-domains and overlapping region in a part of a two-dimensional triangular grid.

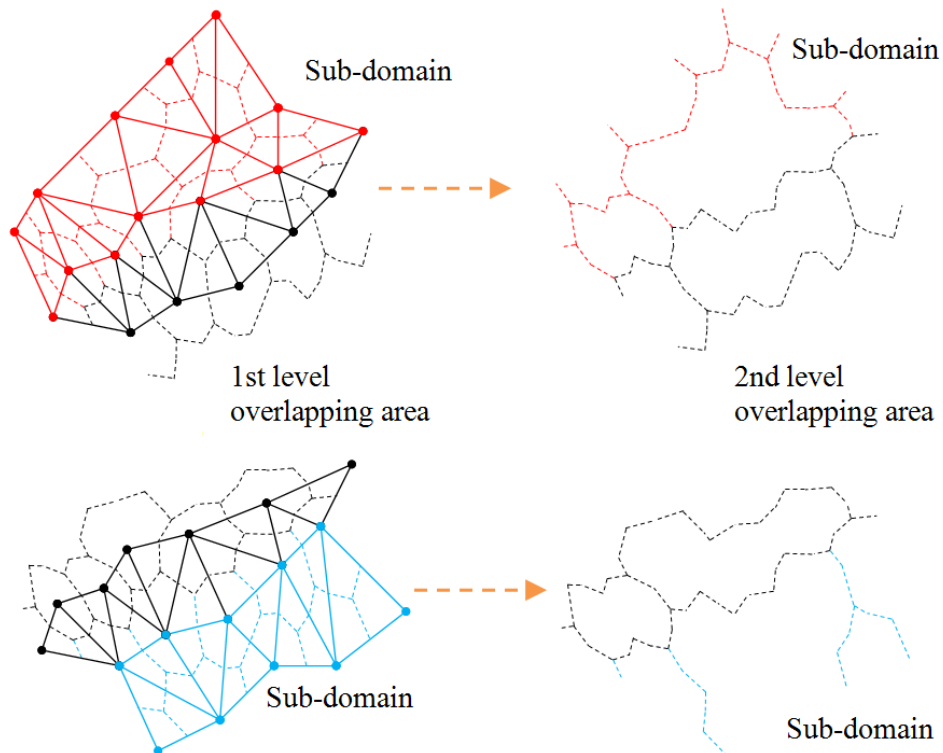


Figure 5.4: Generation of corresponding core and ghost supernodes in a part of a two-dimensional triangular grid.

Implementing the previously described steps on an unstructured tetrahedral grid representing the domain over a rectangular wing with a NACA0012 airfoil, the illustrations of symmetry surface included in Figure 5.5 are obtained.

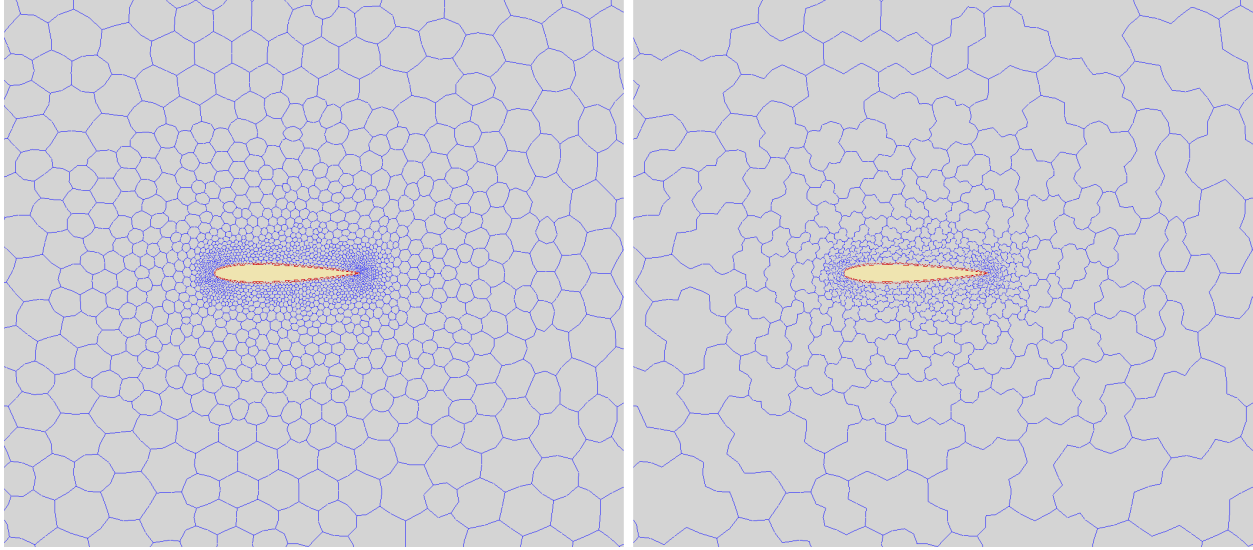


Figure 5.5: Symmetry surface of the initial and isotropically generated coarser control volume grid over a rectangular wing with a NACA0012 airfoil.

Since the agglomeration procedure is complete, the required variables for the new coarser grids, such as the volumes of *super* control volumes, the pseudo-coordinates of *supernodes*, the normal vectors of boundary faces included to new control volumes, the directional weights and the pixelation coefficients of *superedges* in case of a radiative heat transfer problem, have to be evaluated; these computations are performed, based on the corresponding values of merged nodes and edges. In addition, the data for the new *core* and *ghost supernodes* of overlapping regions have to be added to the existing structures (initially including data only for the overlapping *core* and *ghost* nodes of the finest mesh), in order interaction between the agglomerated partitions to be established [Lyg14f].

Moreover, the expression deriving the local time step for each *supernode* is reformulated; as virtual edges are considered in coarser grids, the time step computation is based on the volume and the total surface of the control volume of examined *supernode* P independently of the type of flow. Therefore, the time step is defined similarly to Chapter 2 as:

$$\Delta t_p = CFL \frac{V_p}{\left(\lambda_c^x + \lambda_c^y + \lambda_c^z \right) + 4 \cdot \left(\lambda_v^x + \lambda_v^y + \lambda_v^z \right)} \quad (5.1)$$

In case an inviscid flow problem is encountered the term $4 \cdot \left(\lambda_v^x + \lambda_v^y + \lambda_v^z \right)$ is neglected, while if a laminar flow simulation is performed the turbulent viscosity included in the viscous spectral radii λ_v (equation (2.121)) is assumed equal to zero. In the same way, for a radiative heat transfer problem, the time step utilized in Runge-Kutta method is calculated as [Lyg14f]:

$$c\Delta t_p = \frac{1}{2} \frac{V_p}{\sum_{\partial CV_p} \Delta A_i} \quad (5.2)$$

5.1.1.2. Directional agglomeration

In case of a laminar or turbulent viscous flow problem, usually encountered along with a hybrid grid, full-coarsening directional agglomeration [Car00, Nis11, Nis13] is performed instead, described as follows [Lyg14c, Lyg14d]:

- 1) Step 1 of isotropic agglomeration procedure is performed.
- 2) The solid wall boundary nodes at the prismatic region are identified and assigned an index, e.g., an ascending number. The same index is assigned to the corresponding nodes of the next prismatic layers; in this way lines of prismatic nodes with the same index (implicit lines) [Nis11, Nis13] are constructed.
- 3) Step 2 of isotropic agglomeration procedure is accomplished.
- 4) The boundary prismatic nodes are merged with their eligible for agglomeration adjacent ones, which belong to the same surface, generating in that way new boundary *supernodes* (bottoms of implicit lines) [Mav97, Nis11]. Their indexes are stored together in a data structure in order the nodes of the next prismatic layers to be fused accordingly. If a semi-coarsening process [Mav97, Mav98] is selected instead, the examined boundary prismatic nodes are simply transferred as singletons to the next coarser level; only one node is included at the bottom of each implicit line.
- 5) Similarly to isotropic agglomeration, a list of *seed* nodes is created, including the nodes touched by the agglomeration front, which are actually the nodes of the next prismatic layer.
- 6) A loop over this list is performed, in order the *seed* nodes to be merged with their adjacent non-agglomerated and non-constrained ones, similarly to isotropic strategy. Nevertheless, an extra limitation is imposed to these nodes, according to which a node at prismatic region can be fused with another one only if their indexes are associated and consequently if their corresponding boundary nodes (bottoms of implicit lines) have already been merged. Thus, the nodes belonging to the same implicit line (defined by the corresponding bottom agglomeration) at each two successive prismatic layers are fused together.
- 7) A new *seed* list is constructed similarly to isotropic procedure, including though only prismatic nodes.
- 8) The previous steps 6 and 7 are repeated, until all the prismatic nodes have been transferred to the next coarser level either agglomerated or as singletons. In addition, another limitation is imposed to preserve the topology of the initial mesh up to the coarsest generated one; after the first agglomeration process the nodes of the lowest prismatic layer are not allowed to be merged with the nodes of the next layer, while the same constraint is applied to the nodes of the second prismatic layer after the second process, and so forth. Figure 5.6 includes the schematic representation of the proposed directional agglomeration procedure at the quadrilateral area of a two-dimensional hybrid grid.
- 9) The directional agglomeration is accomplished by performing steps 4 to 8 for the prismatic nodes, whose implicit lines' bottoms belong to a different partition; the *core* nodes of overlapping region are considered as lines' bottoms, instead of the boundary ones.
- 10) A new *seed* list is constructed including the nodes next to the already agglomerated prismatic ones, belonging to tetrahedral or pyramidal elements.
- 11) The isotropic agglomeration procedure is performed for the remaining nodes, beginning from step 4.

Implementing the previously described steps on a hybrid unstructured grid representing the domain over a rectangular wing with a NACA0012 airfoil, the illustrations of symmetry surface included in Figure 5.7 are obtained, while in Figure 5.8 the flow chart of isotropic and directional agglomeration is presented.

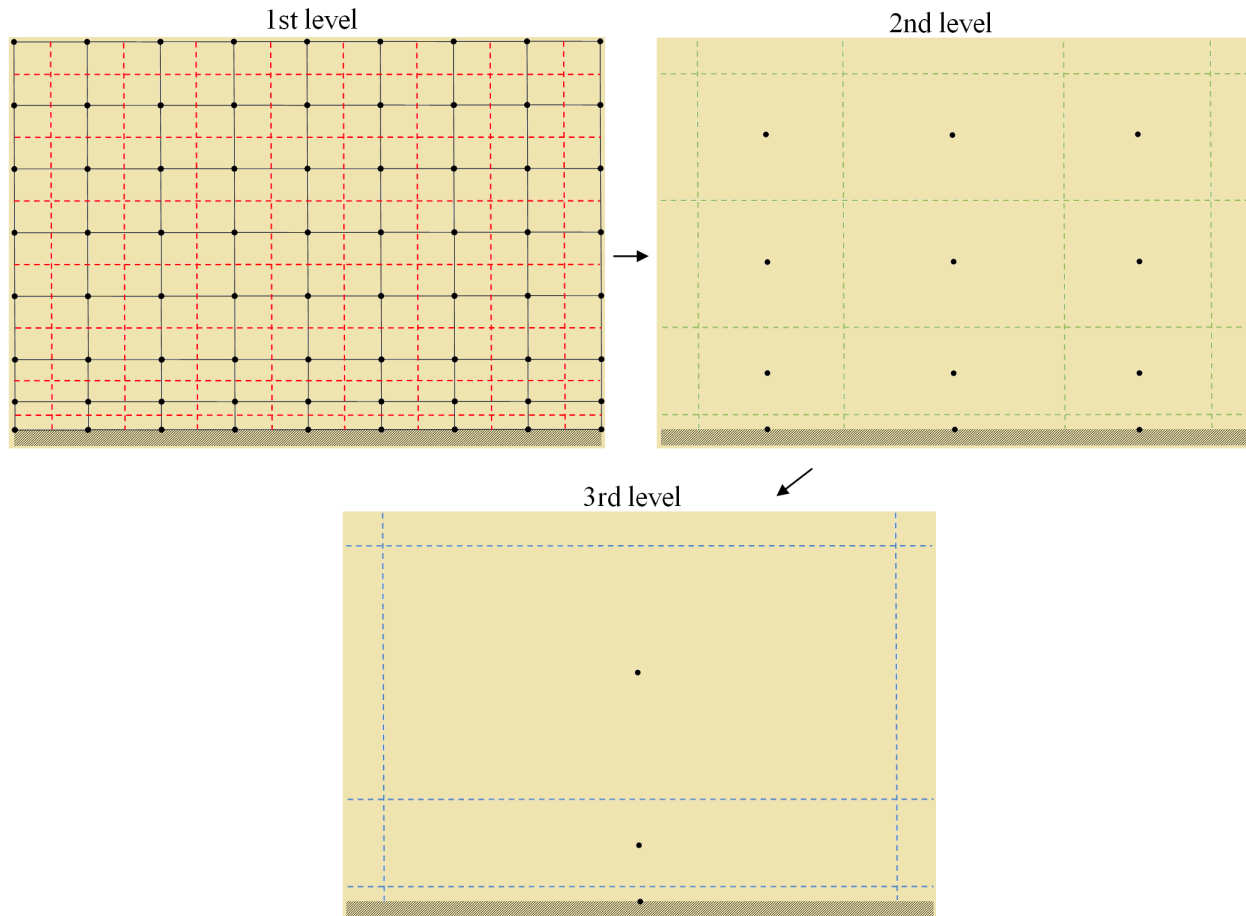


Figure 5.6: Directional agglomeration at the quadrilateral area of a two-dimensional hybrid grid (level 1-red, level 2-green, level 3-blue).

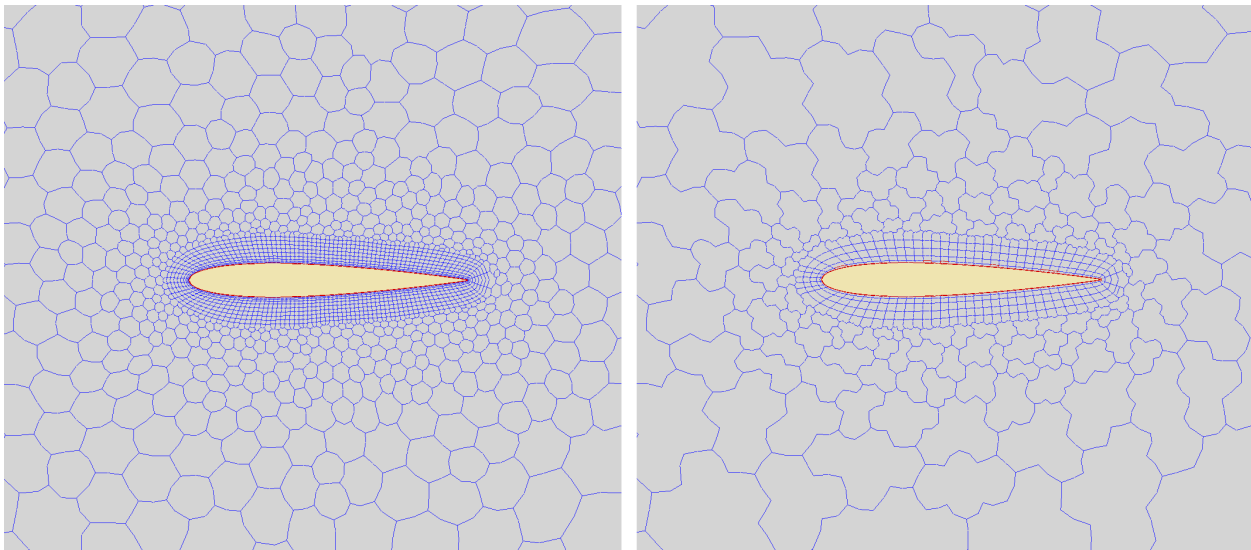


Figure 5.7: Symmetry surface of the initial and directionally generated coarser control volume grid over a rectangular wing with a NACA0012 airfoil.

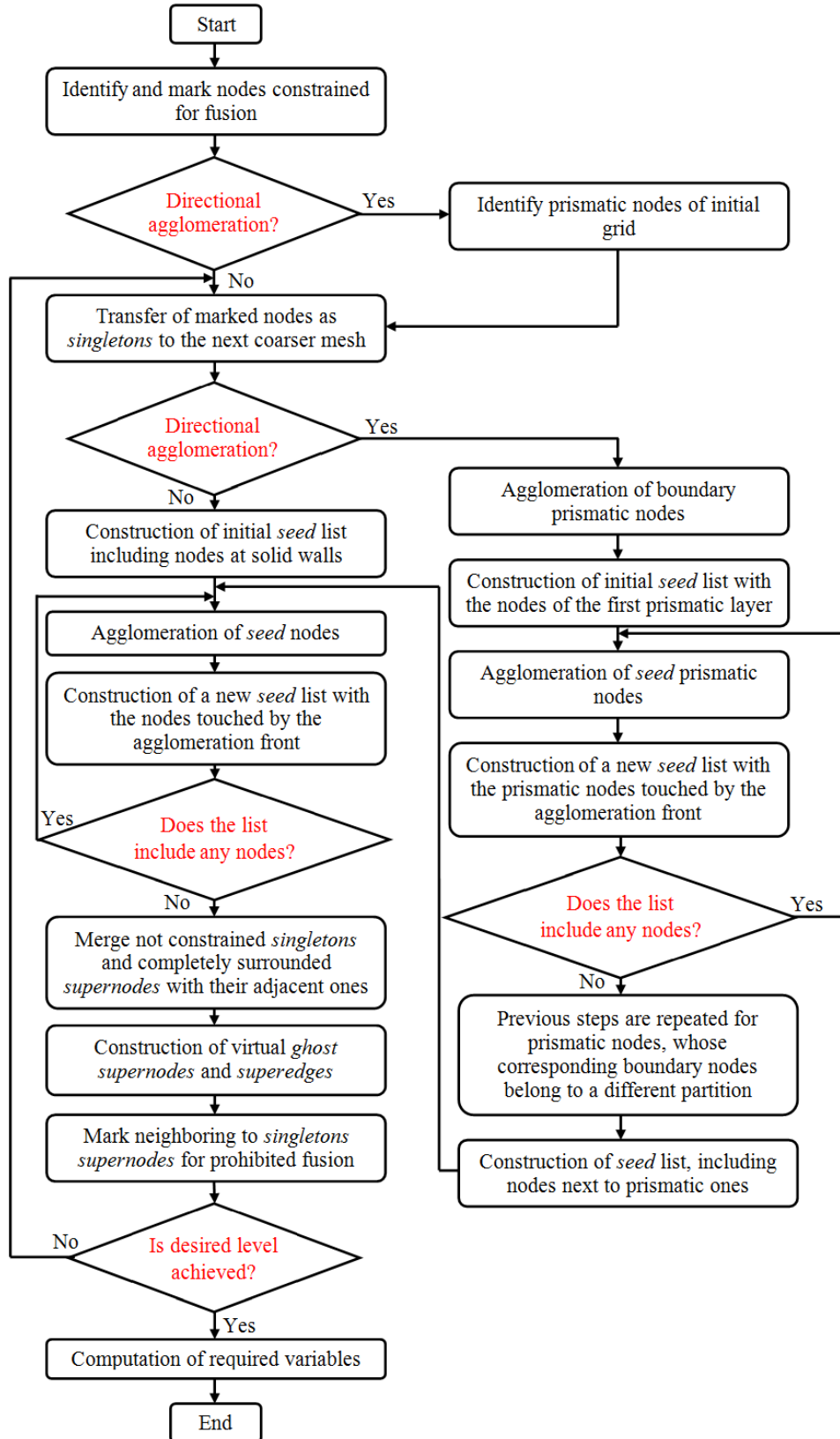


Figure 5.8: Flow chart of isotropic and directional agglomeration procedures.

5.1.2. Flux computation and numerical solution

For a multigrid accelerated iterative solution FAS is implemented, in accordance with which equation (2.118) (and/or equation (3.43) in case of radiative heat transfer) is solved at the first non-agglomerated level but at the coarser ones an approximate version of the same equation is used [Bla01]. The interaction between two successive multigrid levels is obtained via the restriction of conservative variables and flux balances from the finer to the coarser level, as well as the prolongation of conservative variables' corrections from the coarser to the finer level [Sor03, Ni11, Lyg14c]. FAS is implemented in a $V(v_1, v_2)$ -cycle process, where v_1 denotes the number of relaxations before proceeding to the coarser mesh from the finer one, while v_2 the number of relaxations after obtaining the solution from the coarser grid [Nis11, Lyg14c]. For flow simulations performed with an explicit scheme a $V(1,0)$ process is employed, while for those encountered with an implicit one a $V(2,1)$ is used; in test cases involving turbulent flow a $V(1,1)$ and $V(2,2)$ strategy, respectively, is followed instead. If radiative heat transfer is considered, FAS is employed for RTE in a $V(1,0)$ -cycle strategy. For simplification reasons the procedure is analyzed below only for flow equations; its steps are also applicable to RTE by substituting \vec{W}_p with I_p^{mn} and \vec{R}_p with R_p^{mn} , unless otherwise stated.

At each cycle the procedure begins with the solution of equation (2.118) for the first non-agglomerated level (denoted at next with the subscript h), deriving updated values for conservative variables and flux balances of each computational node p . These values are restricted by means of smoothing to the next coarser grid (denoted at next with the subscript H), using the restriction operators $(I_W)_h^H$ and $(I_R)_h^H$ respectively as follows [Man97, Car00, Sor03, Ni11, Lyg14c]:

$$\vec{W}_{P,restricted} = (I_W)_h^H \vec{W}_p = \frac{\sum \vec{W}_p \cdot V_p}{V_P} \quad (5.3)$$

$$\vec{R}_{P,restricted} = (I_R)_h^H \vec{R}_p = \sum \vec{R}_p \quad (5.4)$$

$\vec{W}_{P,restricted}$ is actually the volume-weighted sum vector of the values of conservative variables of nodes p merged to construct the *supernode* P , while $\vec{R}_{P,restricted}$ is the sum vector of flux balances of the same nodes p ; Figure 5.10 illustrates the schematic representation of the spatial agglomeration multigrid restriction process at a prismatic region.

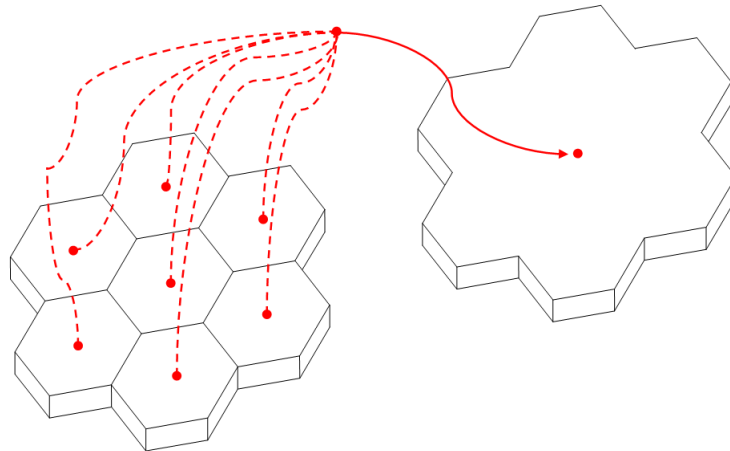


Figure 5.10: Spatial agglomeration multigrid restriction at a prismatic region.

Since the restriction is complete, the approximated equation (2.118) has to be relaxed for the coarser mesh, which is obtained by substituting the right hand side term of this equation with the following one [Mav97, Car00, Bla01, Nis11, Lyg14c, Lyg14f]

$$\bar{R}_{P,FAS} = \bar{R}_P(\bar{W}_P) + \underbrace{\left[\bar{R}_{P,restricted} - \bar{R}_P(\bar{W}_{P,restricted}) \right]}_{\bar{A}_H} \quad (5.5)$$

where $\bar{R}_P(\bar{W}_P)$ is the current flux balance of node P in the coarse grid. Considering the edge-based structure of this algorithm, as well as the derived *superedges* by the agglomeration process, the computation of fluxes appears to be a straightforward procedure; for the inviscid fluxes a first-order accurate spatial scheme is employed (the solution on the coarser levels does not influence the final solution [Bla01, Lyg14f]), while for the viscous ones the necessary gradients are obtained either by applying the nodal-averaging method (described in Chapter 2) or by restricting them similarly to equation (5.3) for conservative variables. The same stands for boundary surfaces' fluxes, as the control volume boundary surface of a *supernode* is the sum of the corresponding surfaces of its incorporated nodes. The next two terms included in the forcing function \bar{A}_H [Bla01, Lyg14f] denote respectively the restricted flux balance to node P from the finer level and the flux balance of the same node generated by using the restricted values of conservative variables. It is quite obvious that at the first internal iteration of Runge-Kutta method or at the first external of Jacobi or Gauss-Seidel algorithm, $\bar{R}_{P,FAS}$ is equal to the restricted flux balance $\bar{R}_{P,restricted}$, confirming the approximation character of FAS; it actually derives a smoothed version of the finest grid solution at the coarser levels [Bla01, Fer02, Lyg14f]. As such, equation (2.118) is reformulated and solved via the iterative scheme for each agglomerated *supernode* P as [Mav96, Mav97, Mav98, Bla01, Fer02, Sor03, Nis10, Nis11, Nis13, Lyg14f]:

$$\Delta \bar{W}_P = -\frac{\Delta t_P}{V_P} \bar{R}_{P,FAS} = -\frac{\Delta t_P}{V_P} \left(\bar{R}_P(\bar{W}_P) + \bar{A}_H \right) \quad (5.6)$$

If an implicit scheme is selected, the Newton-linearized inviscid, viscous and source terms have to be defined before the relaxation of the previous equation; they are evaluated in the same way with this applied for the initial finest grid [Nis11]. However, as elements do not exist in coarser resolutions, only the linearization based on the nodal-averaging computation of gradients can be implemented for viscous terms [Lyg14f].

In case an even coarser mesh has been generated via agglomeration, the restriction process (equations (5.3)-(5.4)) as well as the relaxation process (equations (5.5)-(5.6)) are employed similarly for this grid. As mentioned above, only a first-order accurate spatial scheme is used for the higher agglomerated levels [Sor03, Nis11, Lyg14c]; the flux sum of the finest grid, transferred not only to the next coarser grid but to all the multigrid levels, influences via the forcing function the accuracy of their solutions [Bla01]. The previous state is confirmed by the following analytical equation of the forcing function \bar{A}_{2H} for the second agglomerated mesh [Lyg14f]:

$$\begin{aligned} \bar{A}_{2H} &= \bar{R}_{2P,restricted} - \bar{R}_{2P}(\bar{W}_{2P,restricted}) = (I_R)_{2H}^{2H} \bar{R}_P - \bar{R}_{2P}(\bar{W}_{2P,restricted}) = \\ &= (I_R)_{2H}^{2H} \left(\bar{R}_P(\bar{W}_P) + \bar{A}_H \right) - \bar{R}_{2P}(\bar{W}_{2P,restricted}) \end{aligned} \quad (5.7)$$

Since the solution at the coarsest grid is obtained, the corrections of conservative variables are prolonged by means of smoothing to the next finer mesh [Fer02, Nis11, Lyg14c]; in case of inviscid flow or radiative heat transfer a simple point injection scheme, namely the prolongation operator $(I_W)_H^h$, is implemented as [Car00, Lyg14f]

$$\Delta \vec{W}_p = (I_W)_H^h \Delta \vec{W}_p = \Delta \vec{W}_p = \vec{W}_p - \vec{W}_{p,restricted} \quad (5.8)$$

while in case of laminar or turbulent viscous flow a distance-based scheme is employed, considering only the nodes of the same type (internal or boundary ones of the same surface) [Nis11] as follows [Kat09]

$$\Delta \vec{W}_p = (I_W)_H^h \Delta \vec{W}_p = \frac{a_{p-p} \Delta \vec{W}_p + \sum_{Q \in K_N(p)} a_{p-Q} \Delta \vec{W}_Q}{a_{p-p} + \sum_{Q \in K_N(p)} a_{p-Q}} \quad (5.9)$$

$$a_{p-Q} = |\vec{r}_{pQ}|^{-4}$$

where \vec{r}_{pQ} denotes the vector connecting the agglomerated node p and the *supernode* Q , while $\Delta \vec{W}_p$ and $\Delta \vec{W}_Q$ represent respectively their corrections. The distance-based operator is downgraded to the simple point injection one for singleton supernodes, as a_{p-Q} is zeroized in such cases. Figure 5.11 illustrates the schematic representation of the spatial agglomeration multigrid prolongation procedure at a prismatic region.

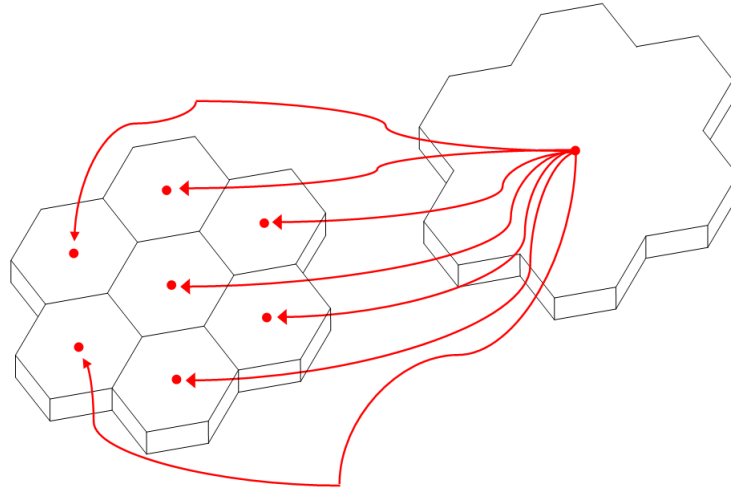


Figure 5.11: Spatial agglomeration multigrid (point-injected) prolongation at a prismatic region.

Although not necessary, an averaging operator can be used to smooth the total error components before updating the conservative variables of the finer grid nodes as [Car00, Lyg14f]

$$\Delta \vec{W}_{p,averaged} = \frac{\Delta \vec{W}_p V_p + \sum_{q \in K_N(p)} \Delta \vec{W}_q V_q}{V_p + \sum_{q \in K_N(p)} V_q} \quad (5.10)$$

where q are the neighbouring nodes of each node p . The updated final values \vec{W}_p^{l+} of the finer mesh nodes at l multigrid iteration/cycle are computed as follows [Car00, Sor03, Nis11, Lyg14f]

$$\vec{W}_p^{l+} = \vec{W}_p^l + \Delta \vec{W}_p \quad (5.11)$$

where \vec{W}_p^l is the vector of previously calculated and not restricted values of conservative variables. The prolongation process is repeated up to the initial finest mesh, accomplishing in that way the V-cycle procedure.

The previously described V-cycle scheme (V(1,0)-cycle strategy) is illustrated schematically in Figure 5.12, in which the bullets denote the relaxation of the solution by the Runge-Kutta iterative method, while at next the whole procedure is presented in pseudo-code formulation [Lyg14f]:

```

Do i = 1 to NS (NS stands for the number of spatial agglomeration levels)
  If (i > 1) then (1 corresponds to the finest non-agglomerated grid)
    Perform spatial restriction
  End if
  Perform the Runge-Kutta (RK(4)) scheme
End do
Do i = NS-1 to 1, step= -1
  Perform spatial prolongation
End do
    
```

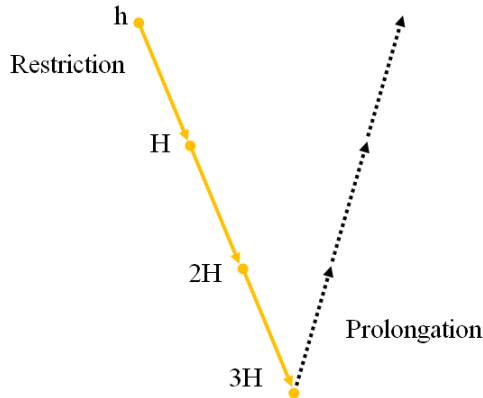


Figure 5.12: Schematic representation of the spatial agglomeration multigrid V(1,0)-cycle scheme.

If an unphysical value of a conservative variable is obtained, e.g., a negative value of pressure, turbulent kinetic energy or radiative intensity, either by the iterative method or by the prolongation process, a fix-up procedure is implemented [Cap10, Lyg14f]. According to this methodology the conservative variables of flow or turbulence model get their previous acceptable values. It is implemented similarly in case of radiative heat transfer; for the finest grid the value is set equal to zero if it was derived by the Runge-Kutta method and equal to its previous value if the unphysical intensity was a result of prolongation from the coarser level, while for the agglomerated grids the negative values are always restored to their previous values. The same procedure is followed for the corresponding flux balances. Independently of the employed scheme (explicit or implicit), each level process of the FAS is completed firstly for the flow equations and then for the turbulence model (loose coupling); in case of radiative heat transfer a similar FAS procedure is followed.

In this study, the FAS is incorporated in the Full Multigrid (FMG) scheme [Fer02, Lyg14c], in order further acceleration to be obtained; following this strategy the solution is decomposed in two stages,

namely the preliminary and the main one (not applicable to radiative heat transfer). At the first stage the flow and turbulence model PDE's are relaxed beginning from the coarsest grid and, as the number of iterative cycles increases, the solution extends successively to the finer meshes up to the initial finest one, at which point the main stage begins; in this way a better initial guess is obtained for each finer level [Lyg14c]. Figure 5.13 includes a schematic representation of the developed combined FMG-FAS (V(1,0)-cycle strategy) procedure.

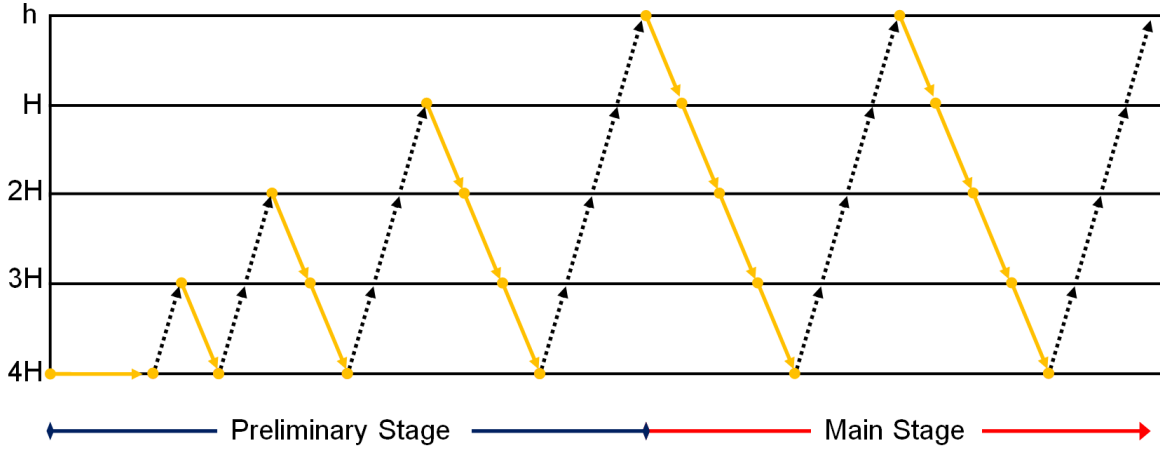


Figure 5.13: FMG-FAS procedure.

5.2. Angular agglomeration multigrid scheme

5.2.1. Angular agglomeration strategy

As mentioned in Introduction, the agglomeration multigrid scheme for RTE is extended in this study beyond the coarsening of spatial resolution to the coarsening of angular one; it is performed in a similar manner, based on the solution of RTE on successively coarser angular levels, requiring as such the implementation of a corresponding angular agglomeration strategy [Lyg14b, Lyg14f, Lyg14g]. The angular fusion is performed by merging neighbouring solid control angles of the finer angular discretization and creating new solid control *superangles*. Although being a similar process to the spatial scheme, it is much simpler, mainly due to the absence of the large number of restrictions of the spatial one.

For each angular level the procedure begins with the identification and fusion of every two neighbouring azimuthal angles; it extends then similarly to the polar direction, deriving the new solid control *superangles* (respectively to *supernodes* derived by the spatial agglomeration scheme). Therefore, the equal division of 4π steradians into $N_\theta \times N_\phi$ directions employed for the single-grid level, is

reformulated into $N_\theta/2 \times N_\phi/2$ ones, such that $\Delta\theta^m = \theta^{m+} - \theta^{m-} = \pi/2N_\theta$ and $\Delta\phi^n = \phi^{n+} - \phi^{n-} = 2\pi/2N_\phi$ [Lyg14f]. Considering the commonly used *circle* and *sphere*, representing the two- and three-dimensional discretization respectively [Rai99], the angular agglomeration derives finally a *circle* with the half number of angles and a *sphere* with the quarter number of solid angles. In case an even coarser angular discretization is required the procedure is repeated, resulting in $N_\theta/4 \times N_\phi/4$ directions; depending on the test cases encountered in this study, a number of two or three angular multigrid levels has been identified to be adequate. Figure 5.14 illustrates the schematic representation of the angular merging process in two dimensions, in which sixteen control angles of the initial resolution are agglomerated successively to eight and four ones for the second and third multigrid level respectively. In Figure 5.15 the corresponding three-dimensional procedure is presented, in which thirty two solid control angles are agglomerated in eight new *superangles*.

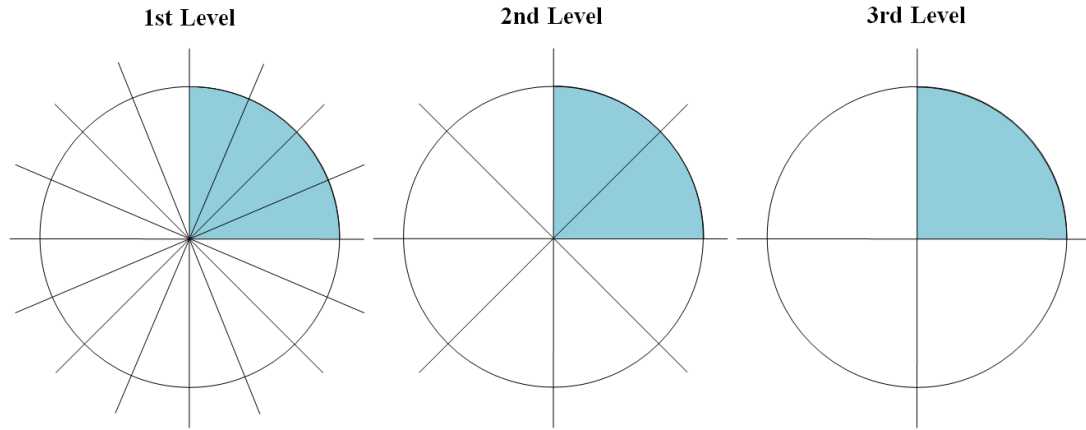


Figure 5.14: Angular agglomeration in two dimensions.

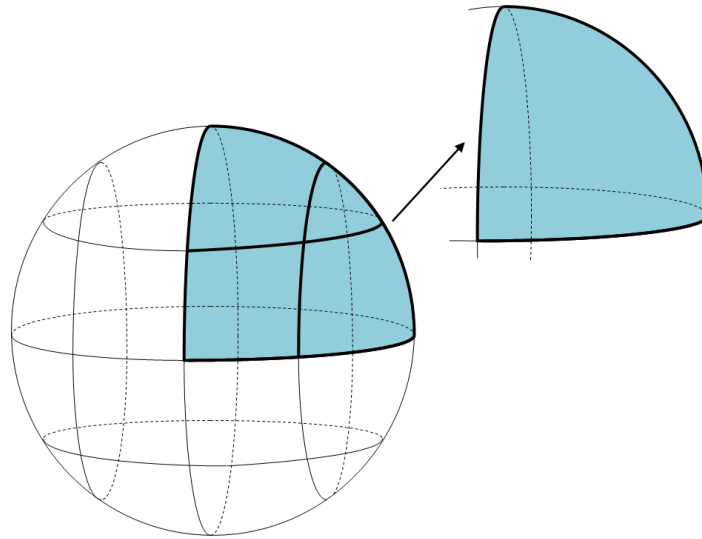


Figure 5.15: Angular agglomeration in three dimensions.

Although the spatial agglomeration methodology is constrained by a non-trivial number of limitations, only one is applied to the angular one; the adjacent angles, either in azimuthal or polar direction, examined for merging, shall belong to the same quadrant of the directional *circle* or *sphere* [Lyg14b, Lyg14f]. In that way the fused angles as well as the derived *superangles* have the same sign, ensuring the consistency of the restriction and prolongation relations for two successive angular discretizations. As the common practice is to employ an even number of angles in both directions for symmetry reasons, the agglomeration procedure is not influenced significantly by this restriction. In addition, this limitation defines a minimum allowed number of control *superangles* for each direction, four in the azimuthal and two in the polar direction; it is applied separately in each direction, in order not to affect the number of the pre-defined angular multigrid levels. For example in a quasi-3D test case, in which sixteen azimuthal and four polar angles are considered along with three multigrid resolutions, the angular agglomeration method derives eight azimuthal and two polar angles for the second level and four azimuthal and two polar ones for the third angular *grid*.

Since the coarser angular resolutions are ready, the corresponding required variables, such as the directional weights and the pixelation coefficients, are computed [Lyg14b, Lyg14f]. This calculation can be performed either analytically, using the new solid control *superangles* for each edge and boundary surface, or by simply summing the absolute values of directional weights, as well as their positive ones, for the merged control angles; the latter is incorporated in this method. Because of the aforementioned

limitation there is no need for special treatment, in order to avoid summing adverse directional weights. As far as the parallelization strategy is concerned, additional data structures for the new *superangles* have to be constructed for the interaction between the partitions via their overlapping regions. Finally, the angular agglomeration procedure is completed with the computation of the average scattering phase functions for the new *superangles* and the definition of the corresponding mirroring directions [Lyg14b, Lyg14f].

5.2.2. Flux computation and numerical solution

For the angular multigrid accelerated iterative solution of RTE, the FAS via a V(1,0)-cycle is implemented again. Similarly to spatial multigrid methodology, each cycle begins with the solution of equation (3.43) for the first non-agglomerated level (denoted at next with the superscript mn) updating the values of radiative intensity as well as those of flux balance; these values are transferred then to the next coarser resolution (denoted at next with the superscript MN), utilizing the restriction operators $(I_R)_{mn}^{MN}$ and $(I_I)_{mn}^{MN}$ respectively as follows [Lyg14b, Lyg14f]

$$R_{p,restricted}^{MN} = (I_R)_{mn}^{MN} R_p^{mn} = \sum R_p^{mn} \quad (5.12)$$

$$I_{p,restricted}^{MN} = (I_I)_{mn}^{MN} I_p^{mn} = \frac{\sum I_p^{mn} \cdot \Delta\Omega^{mn}}{\Delta\Omega^{MN}} \quad (5.13)$$

where $R_{p,restricted}^{MN}$ denotes the sum of flux balances of merged solid control angles mn to the *superangle* MN , while $I_{p,restricted}^{MN}$ represents the angular-weighted sum of radiative intensity values of the same fused control angles. Figure 5.16 includes the schematic representation of restriction procedure in two dimensions, in which the bullets stand for the summation/weighted-summation of transferred variables (radiative intensity and flux balance).

The solution at the coarser angular resolution is succeeded once more via the implementation of the approximate equation (3.43), in which the right hand side term is reformulated similarly to this of equation (5.5) as [Lyg14b, Lyg14f]

$$R_{p,FAS}^{MN} = R_p^{MN} (I_p^{MN}) + \underbrace{\left[R_{p,restricted}^{MN} - R_p^{MN} (I_{p,restricted}^{MN}) \right]}_{A^{MN}} \quad (5.14)$$

where $R_p^{MN} (I_p^{MN})$ is the current flux balance in direction MN of the coarse angular *grid*, while A^{MN} is the angular forcing function, controlling the accuracy of the coarser angular levels. Similarly to spatial multigrid technique, the computation of the required fluxes of the first right hand side term is a straightforward process; a first-order accurate spatial scheme is applied to the coarser resolutions, considering that they do not influence the accuracy of the final steady-state solution. For an even coarser angular *grid* the restriction as well as the relaxation with the Runge-Kutta method are repeated.

Since the solution at the coarsest angular discretization is obtained, the corrections of radiative intensity are prolonged by means of interpolation to the next angularly denser level, considering a simple point-injection scheme, which utilizes the prolongation operator $(I_I)_{MN}^{mn}$ as [Lyg14b, Lyg14f]

$$\Delta I_p^{mn} = (I_I)_{MN}^{mn} \Delta I_p^{MN} = \Delta I_p^{MN} = I_p^{MN} - I_{p,restricted}^{MN} \quad (5.15)$$

where ΔI_p^{mn} is the correction for the agglomerated angle mn , while ΔI_p^{MN} is the respective correction for the *superangle* MN . Figure 5.17 presents the schematic representation of the prolongation procedure in two dimensions.

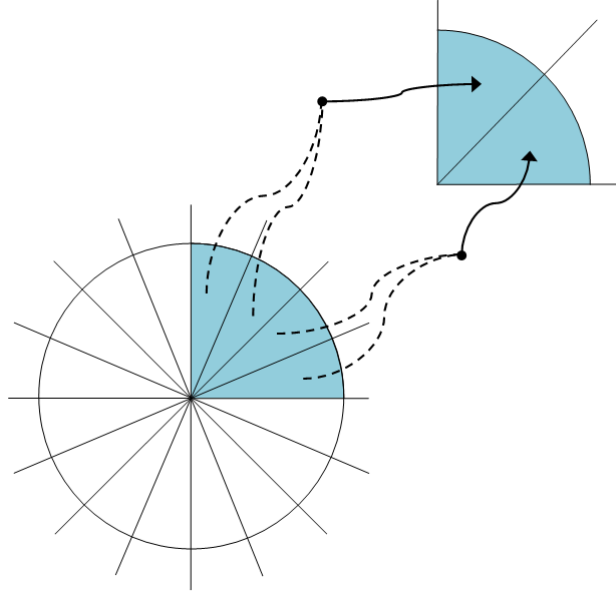


Figure 5.16: Angular agglomeration multigrid restriction in two dimensions.

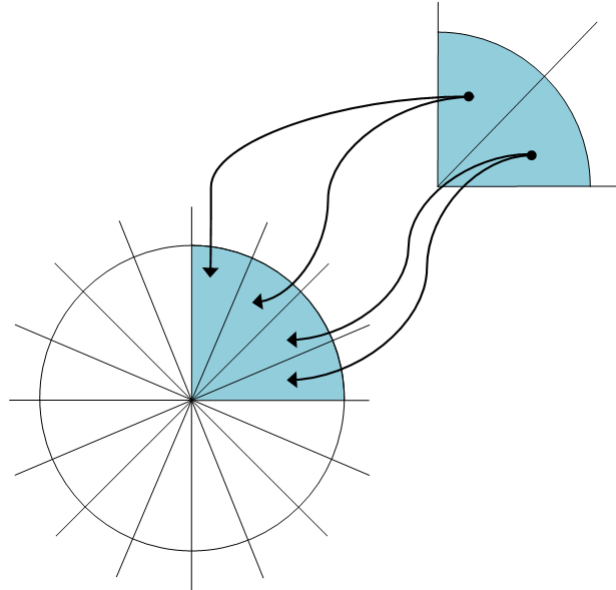


Figure 5.17: Angular agglomeration multigrid prolongation in two dimensions.

For the computation of the updated final value of radiative intensity at the initial densest resolution equation (5.11) is implemented again. The whole procedure is presented schematically in Figure 5.18, while at next it is outlined in pseudo-code formulation as [Lyg14f]:

```

Do  $j = 1$  to  $NA$  ( $NA$  stands for the number of angular agglomeration levels).
  If ( $j > 1$ ) then ( $1$  corresponds to the densest angular resolution)
    Perform angular restriction
    
```



```

End if
Perform the Runge-Kutta (RK(4)) scheme
End do
Do j = NA-1 to 1, step = -1
    Perform angular prolongation
End do

```

In case a negative, unphysical radiative intensity is obtained by the prolongation process, its previous value, as well as the previous value of the corresponding flux balance, is restored; if it is derived through the Runge-Kutta method at the densest level, its value is set equal to zero [Cap10, Lyg14f].

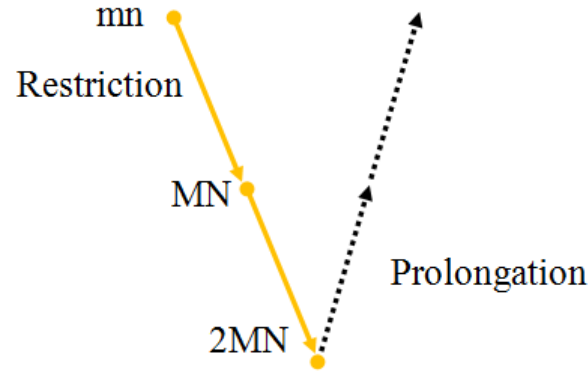


Figure 5.18: Schematic representation of the angular agglomeration multigrid $V(1,0)$ -cycle scheme.

5.3. Combined spatial/angular agglomeration multigrid scheme

A nested combination of the aforementioned (spatial and angular) agglomeration multigrid schemes was finally developed for additional acceleration in radiative heat transfer computations [Lyg14b, Lyg14f]. The whole procedure begins with the spatial agglomeration of the initial finest mesh and the generation of the coarser grids; it extends then to the construction of the coarser angular resolutions. For the accelerated solution of RTE a complete angular FAS V-cycle is employed at each level of the spatial multigrid scheme, described in Figure 5.19 and the following pseudo-code [Lyg14f]:

```

Do i = 1 to NS (NS stands for the number of spatial agglomeration levels)
    If (i > 1) then (1 corresponds to the finest non-agglomerated grid)
        Perform spatial restriction for i level (angular level is 1)
    End if
    Do j = 1 to NA (NA stands for the number of angular agglomeration levels).
        If (j > 1) then (1 corresponds to the densest angular resolution)
            Perform angular restriction for i and j level respectively
        End if
        Perform the Runge-Kutta (RK(4)) scheme for i and j level respectively
    End do
    Do j = NA-1 to 1, step = -1
        Perform angular prolongation for i level and j level respectively
    End do
    Perform the Runge-Kutta (RK(4)) scheme for i level (angular level is 1)
End do
Do i = NS-1 to 1, step = -1
    Perform spatial prolongation for i level (angular level is 1)
End do

```

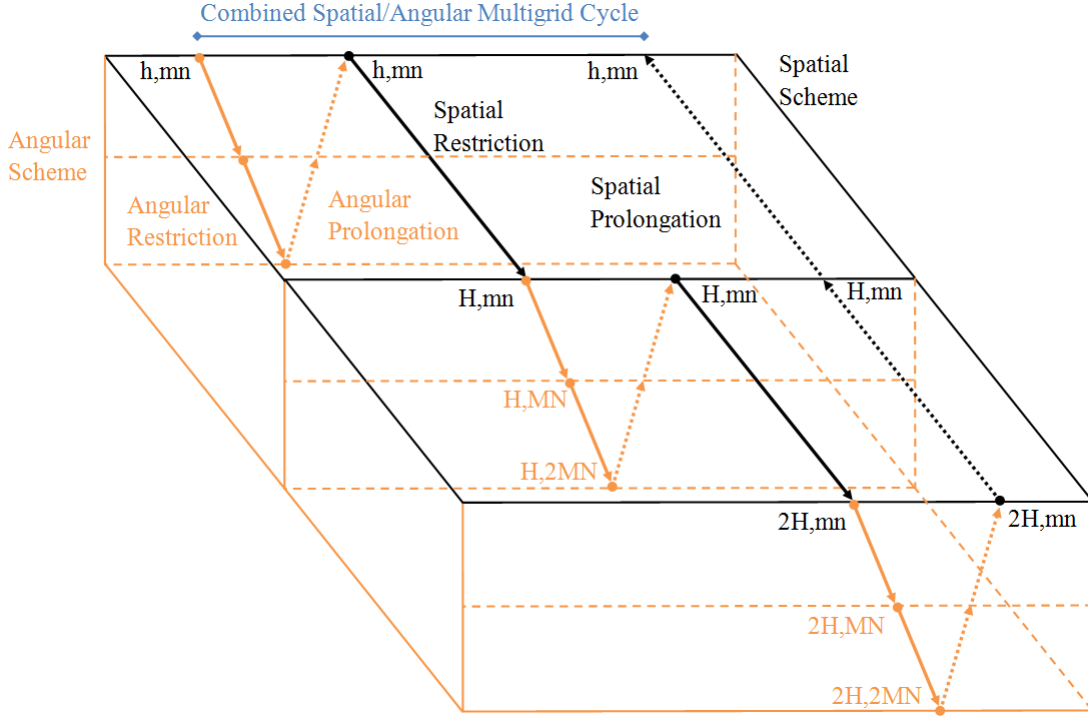


Figure 5.19: Schematic representation of the combined (nested) spatial/angular agglomeration multigrid scheme.

Considering the implementation of a complete angular multigrid scheme at each spatial agglomeration level, the computation of the appropriate directional weights and pixelation coefficients is performed for all the combinations of spatial grids and angular resolutions at a pre-processing stage [Lyg14f]. The spatial and angular, restriction and prolongation operators are applied similarly to the only-spatial and only-angular multigrid schemes, while the evaluation of the numerical fluxes is performed exactly in the same way. The main difference concerns the forcing function, which, for the nested angular multigrid stage, includes not only the angular function A^{MN} but also the spatial one A_H , as indicated by the following equation for a coarser spatial and angular resolution:

$$A_H^{MN} = (I_R)_{mn}^{MN} R_{P,FAS}^{mn} - R_P^{MN} \left((I_I)_{mn}^{MN} I_P^{mn} \right) = (I_R)_{mn}^{MN} (R_P^{mn} + A_H^{mn}) - R_P^{MN} \left((I_I)_{mn}^{MN} I_P^{mn} \right) \quad (5.16)$$

Moreover, an additional difference is identified to the double implementation of the Runge-Kutta method per multigrid cycle for each spatial level at the initial densest angular discretization; this second relaxation is necessary in order the values of radiative intensity and flux balances to be updated after being prolonged angularly and before being restricted spatially. Similarly to the aforementioned methods, the second-order accurate spatial scheme and the zero fix up procedure [Cap10, Lyg14f] are employed only for the finest spatial and angular resolutions.

6. Grid adaptation

6.1. H-refinement

Numerical diffusion and false scattering in flow and radiative heat transfer predictions respectively can be mitigated by the utilization of finer meshes, resulting however in increased memory requirements and computational effort. In such cases as well as in simulations with sufficiently localized phenomena or in problems faced for the first time, for which no previous knowledge exists for the required resolution, local grid adaptation techniques can be implemented during the iterative procedure in order to resolve the physical behaviour and increase the accuracy of the final solution, along with significant computational savings [Loh92, Lyg13c, Lyg14a, Lyg15]. As mentioned in Introduction, various methods have been developed for mesh adaptation divided mainly in two categories, namely redistribution and refinement; the h-refinement has been incorporated in this algorithm [Kal96, Kha00].

According to this methodology, the whole procedure begins with the detection of desired areas for enrichment, usually concerning regions with discontinuities, shocks, stagnation points, etc., [Loh92]. Thus, appropriate criteria have to be implemented to capture the characteristics of the intent local physical behaviour of the flow or radiative heat transfer; these criteria target either areas with a specific value of a variable/function (Mach number, velocity, pressure, radiative intensity, etc.) or areas designated by the corresponding experimental data (shock location), e.g., the region over the surface of a wing with Mach number greater equal to 0.95, if the shock location is desired to be enriched [Lyg13b, Lyg13c, Lyg14a, Lyg15].

After the appropriate regions are selected for refinement, the corresponding mesh edges are divided, while next the division is extended to the faces and elements following pre-defined rules [Loh92, Kha00, Kav10]. This edge-based feature makes h-refinement particularly attractive for implementation in hybrid grids, including various types of elements [Lyg13c]. In the next paragraphs the aforementioned rules are analyzed, as well as the steps of the whole procedure are summarized. The pre-defined rules, concerning the division of the existing edges, faces and elements, are described as follows [Lyg13c]:

1) Every new, embedded in an edge, node of the grid must be a common point of all the neighbouring faces and elements; hanging-nodes are not accepted. Figure 6.1 illustrates an acceptable and an unacceptable embedded node in a part of a two-dimensional triangular grid.

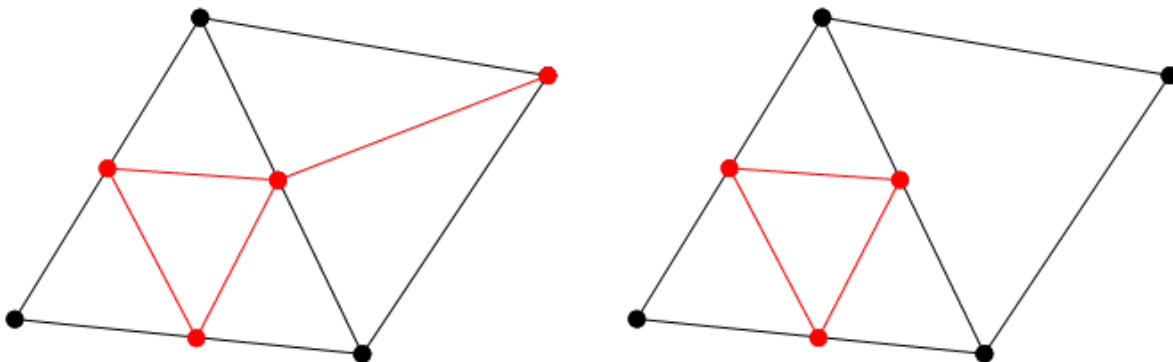


Figure 6.1: Acceptable (left) and unacceptable (right) embedded node in a part of a two-dimensional triangular grid.

2) The new nodes are embedded only at the midpoints of the marked edges of the mesh; their coordinates (as well as the variables' values for initialization purposes) are set equal to the arithmetic averages of the corresponding values at the two end-points.

3) Every new edge is a common edge of all the neighbouring elements.

4) A tetrahedron can be divided in five ways (as shown in Figure 6.2) resulting in [Loh92, Kal96, Kha00]:

- (a) two new tetrahedra,
- (b) four new tetrahedra,
- (c) eight new tetrahedra,
- (d) one new tetrahedron and one new pyramid,
- (e) two new tetrahedra, one new pyramid and one new prism.

The last two types are implemented only in the transition region of the grid, from the prismatic to the tetrahedral elements. For tetrahedra divided in eight new ones, three possible ways of division can be employed, depending on the selected internal diagonal edge (7-9, 5-10, 6-8), as depicted in Figure 6.3. The shortest inner diagonal edge should always be chosen, decreasing in that way the possibility of producing distorted new tetrahedra [Loh92].

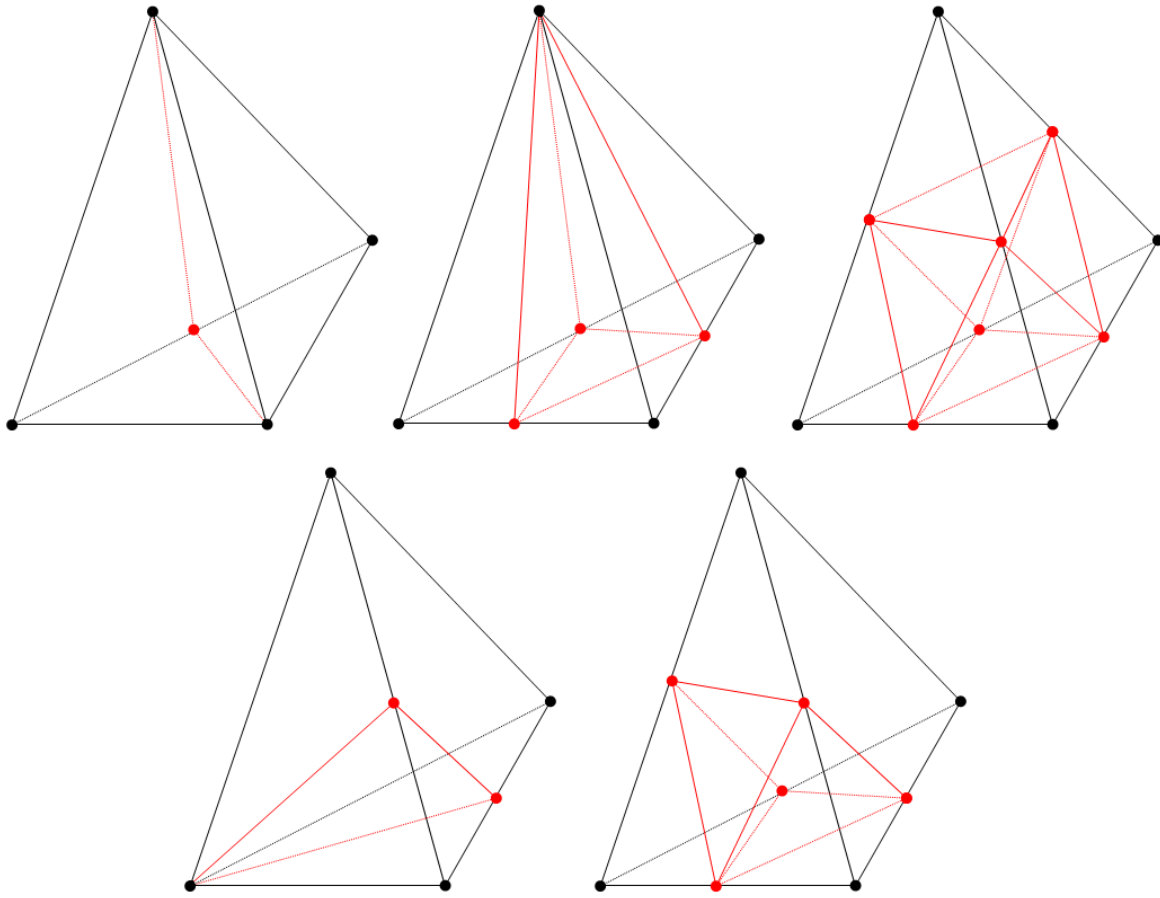


Figure 6.2: Permitted ways of division for tetrahedra.

5) Respectively, a prism can be divided in two (a) or four (b) new prisms, as illustrated in Figure 6.4 [Kha00]. As their height is usually pre-defined, the refinement is implemented only in the longitudinal direction.

6) Considering that the pyramidal elements are located usually only in the transition region of the grid from the prismatic to the tetrahedral elements, a pyramid can be divided in three ways [Kal96, Kha00], as shown in Figure 6.5, producing:

- (a) two new pyramids,
- (b) three new prisms and a new pyramid,

(c) a new prism and a new pyramid.

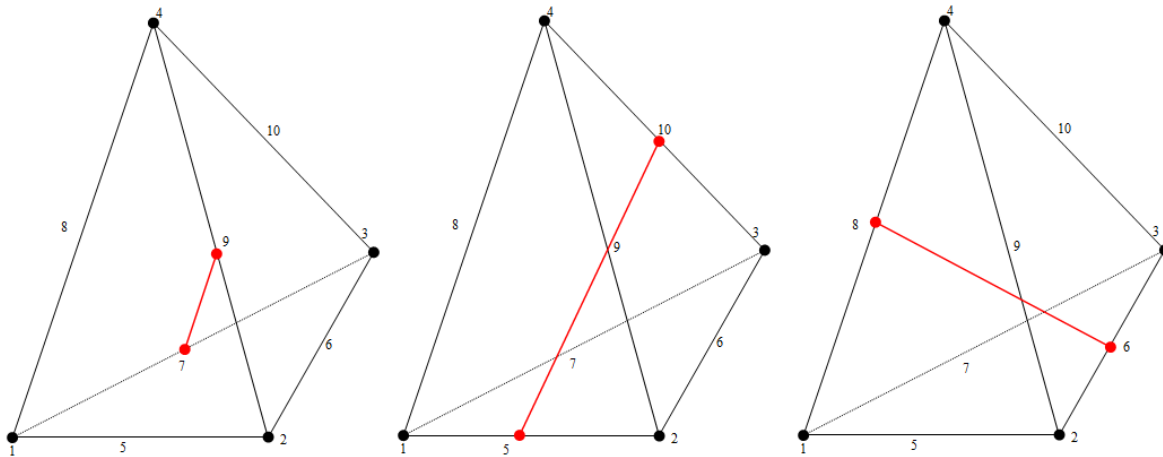


Figure 6.3: Possible inner diagonal edges for the division of a tetrahedron in eight new ones.

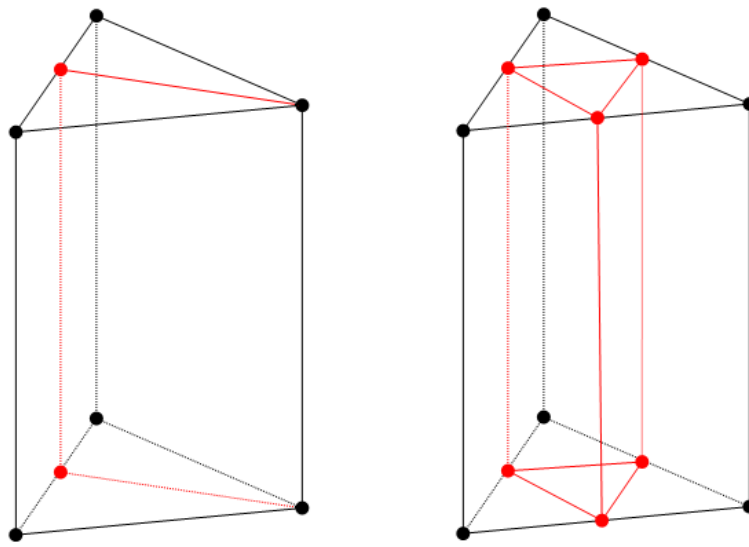


Figure 6.4: Permitted ways of division for prisms.

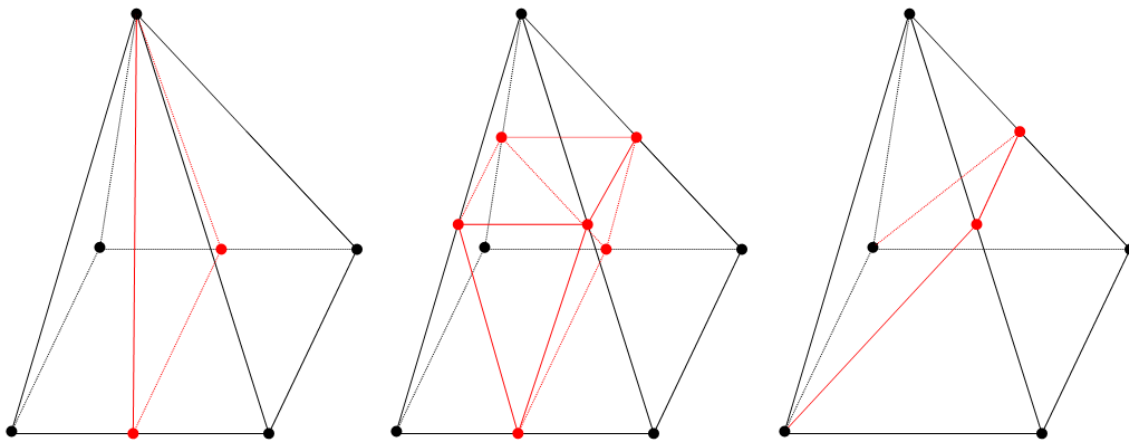


Figure 6.5: Permitted ways of division for pyramids.

7) Similar rules have to be followed for the division of the triangular and rectangular faces, as well; as illustrated in Figure 6.6 a triangular face can be divided in two or four new triangular faces, while a rectangular one can be divided only in two new rectangular faces [Loh92].

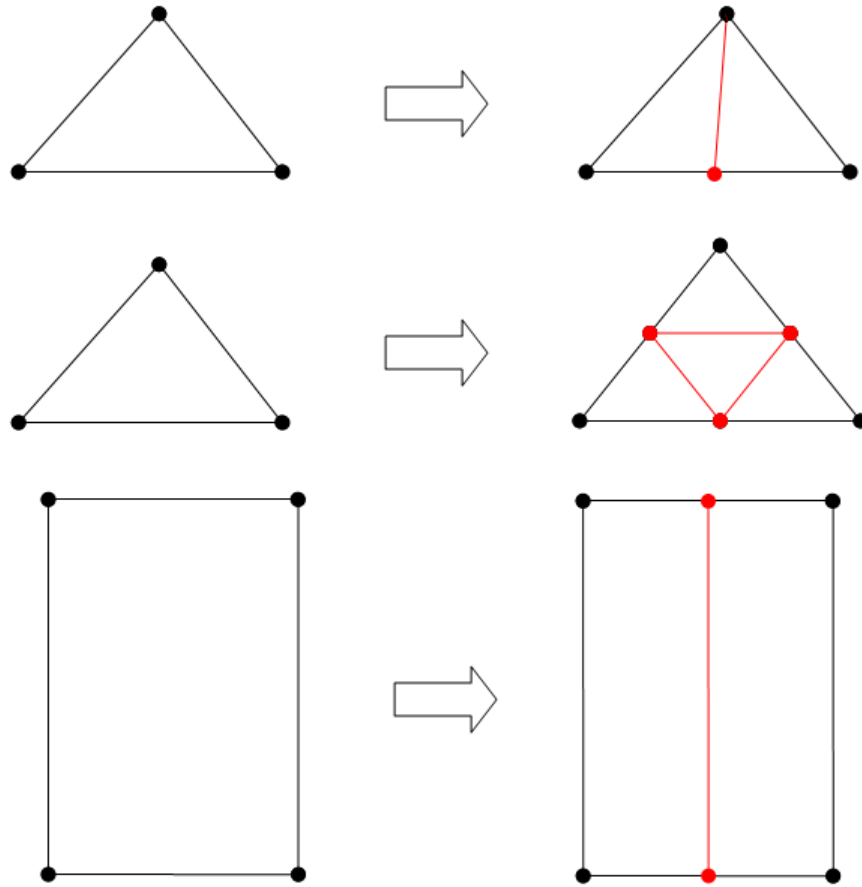


Figure 6.6: Permitted ways of division for triangular and rectangular faces.

6.2. Subdivision method

According to afore-mentioned rule (2), the new nodes are embedded as midpoints of the marked edges of the mesh, getting as coordinates' and variables' values the arithmetic averages of the corresponding values at the two end-points. Although not necessarily, for the new nodes embedded at boundary edges a subdivision technique can be implemented for the corresponding computation instead of arithmetic averaging [Ull06, Zor98]; in accordance to this method the values of more neighbouring nodes are considered in order to avoid distorting the boundary surfaces. Figure 6.7 illustrates such an example in a part of a two-dimensional triangular grid with a distorted boundary edge derived by the implementation of the arithmetic average technique, as well as its smoothed correction, taking into account more adjacent mesh nodes.

In three-dimensional grids, such as those used with the proposed algorithm, a diamond scheme is employed for the computation of the coordinates' and variables' values of the new embedded boundary nodes. Mathematically is formulated as follows [Ull06]

$$x_M = \frac{(x_K + x_L + 3x_P + 3x_Q)}{8} \quad (6.1)$$

where subscripts denote the nodes of the two adjacent triangular boundary faces, as illustrated in Figure 6.8.

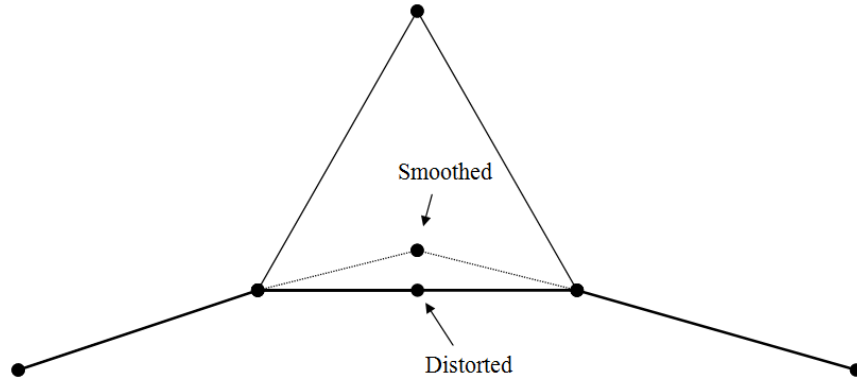


Figure 6.7: Distorted boundary edge and its smoothed correction in a part of a two-dimensional triangular grid.

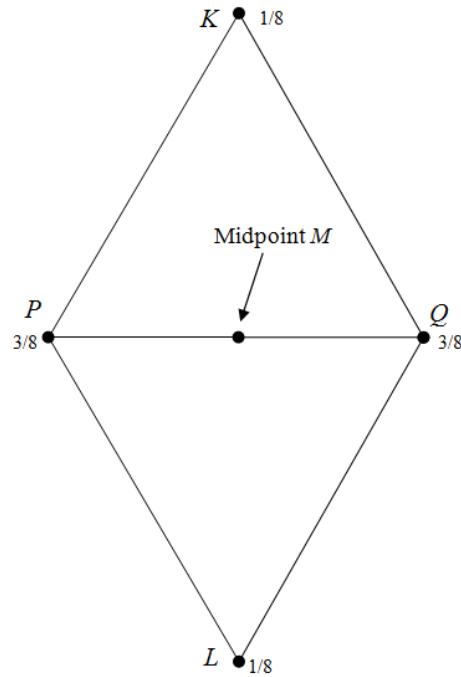


Figure 6.8: Embedding a new node in a boundary edge PQ of a three-dimensional unstructured grid, using the subdivision technique.

6.3. Procedure description

Considering the previous rules/limitations, the h-refinement procedure can be divided in four main steps [Lyg13b, Lyg13c, Lyg14a, Lyg15]

- Initial marking of the edges, according to the selected criteria,
- spread of marking information to the neighbouring edges,
- division of targeted edges, and
- Division of faces and elements,

while in Figure 6.9. the flow chart of the whole procedure is presented.

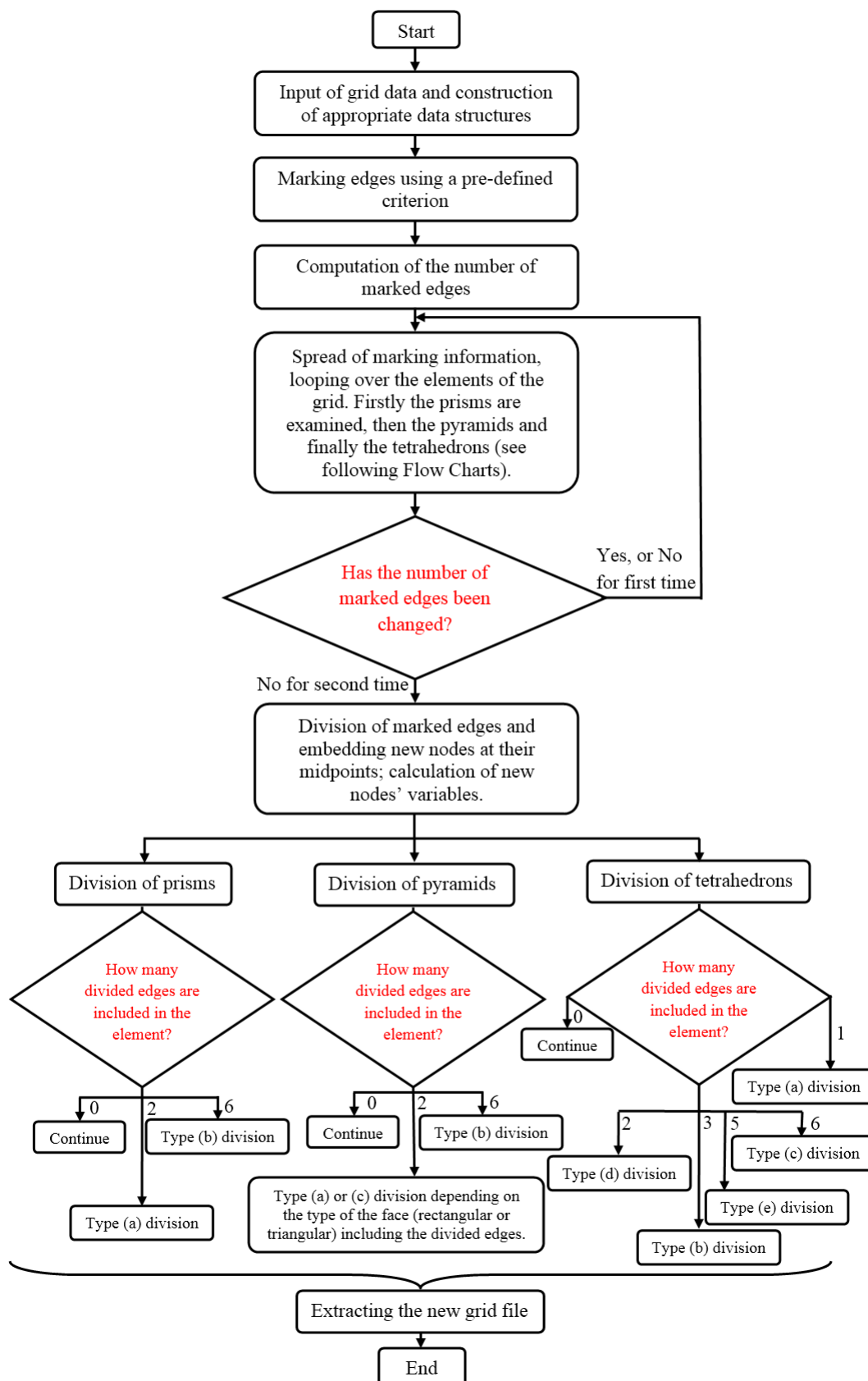


Figure 6.9: Flow chart of h-refinement procedure.

As it can be observed, the procedure begins with the input of grid data and the construction of required data structures, i.e., since nodes' connectivities and elements of the mesh have been obtained, their faces as well as their edges are defined. The process continues with marking the edges at desired regions for enrichment (step a); an appropriate criterion is employed, defined by user's requirements, e.g., targeting edges which include endpoints with value of Mach number greater or equal to unity. At this point the marked edges are enumerated in order this number to be used later as a criterion for the completion of the procedure. The derived list of marked edges is still however in a preliminary status and cannot lead to an admissible refinement pattern for the construction of a new mesh. Therefore, their adjacent edges shall be also examined for marking (and consequently division), in order an admissible pattern to be reached. The procedure is accomplished by looping several times over the elements of the mesh (step b) [Loh92, Lyg14a, Lyg15]; firstly the prismatic edges are examined, while at next a loop over the pyramidal and the tetrahedral elements of the mesh is performed. This sequence derives from rule 5 that forbids division of prismatic edges, located along the stretched direction of the element. If however a prohibited for division edge of a prism is identified as marked, it is unmarked and flagged as forbidden for remarking during this stage. For the corresponding edges of pyramids there is no need for such an examination, as in case they have been marked during the initial marking entailed by the employed criterion, they would have been unmarked during the loop over the prismatic elements. In order to distinguish the transitional tetrahedra (located at the transition region from prismatic to pyramidal and tetrahedral elements) from the non-transitional ones, their edges are examined for existence of a prohibited for division edge among them; if one is identified, type (d) or (e) division is implemented. In Figures 6.10-6.12 the corresponding flow charts of spread of information among the different types of elements are depicted.

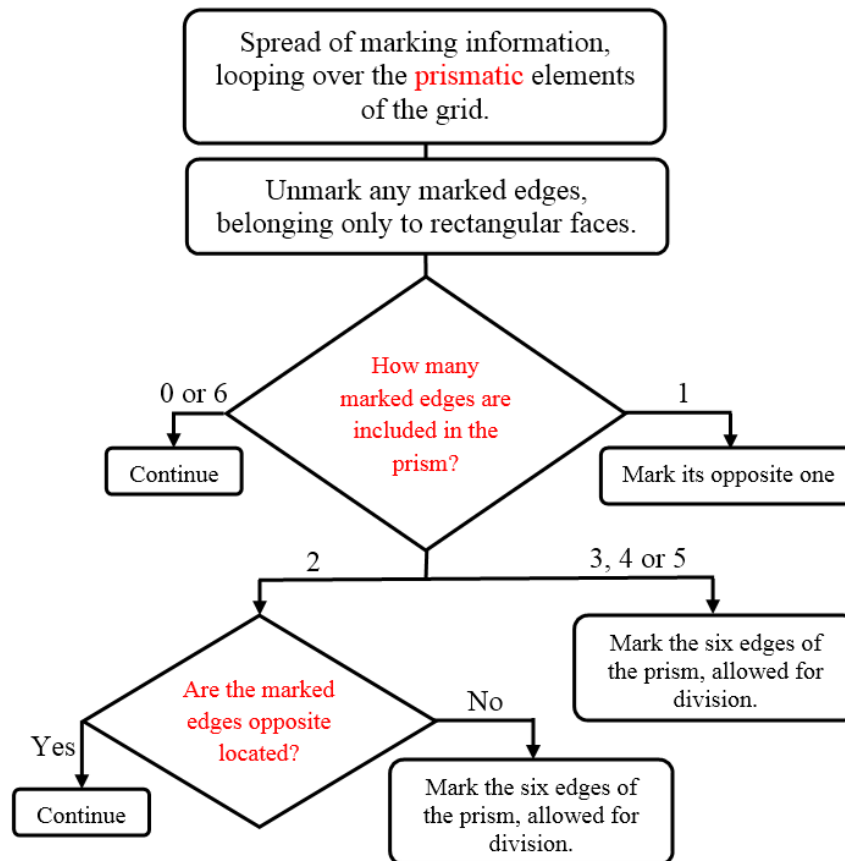


Figure 6.10: Flow chart for the spread of marking information among the edges of the prismatic elements.

For instance a tetrahedron is examined; during the first step of the procedure, one up to six edges of it can be marked. If three ones are assigned for division according to the employed criterion, further examination is required as two possible ways of division exist; in case those edges are located at the same face the element is divided in four new tetrahedra, while if they belong to different faces three more edges should be marked in order the element to be divided in eight new tetrahedra. The previous example is applicable for a tetrahedron located far from the prismatic region; if the tetrahedron lies at the transition region (from prismatic to tetrahedral elements) further examination is required, as two more ways of division can be employed (type (d) and (e) ones, producing one new tetrahedron and one new pyramid or two new tetrahedra, one new pyramid and one new prism respectively). In case of prismatic or pyramidal elements, the aforementioned procedure is much simpler, due to the fewer number of permitted ways of division (two for prisms and three for pyramids).

Once the final number of the selected edges for division is defined (twice), the refinement procedure begins, deriving new edges with half the length of the initial ones, as well as new nodes embedded at their midpoints (step c); the variables of new nodes are then computed. Subsequently, the new elements along with the new faces are constructed, following the rules analyzed previously (step d). Although the refinement procedure can be implemented in a parallelized computational system [Kav10], it is still performed serially in our algorithm, for simplification reasons; nevertheless, it appears significantly efficient as it requires less than a few minutes to refine locally a grid with about 4,500,000 nodes, producing a new one with about 6,500,000 nodes (CRM WBHT aircraft) [Lyg14a, Lyg15].

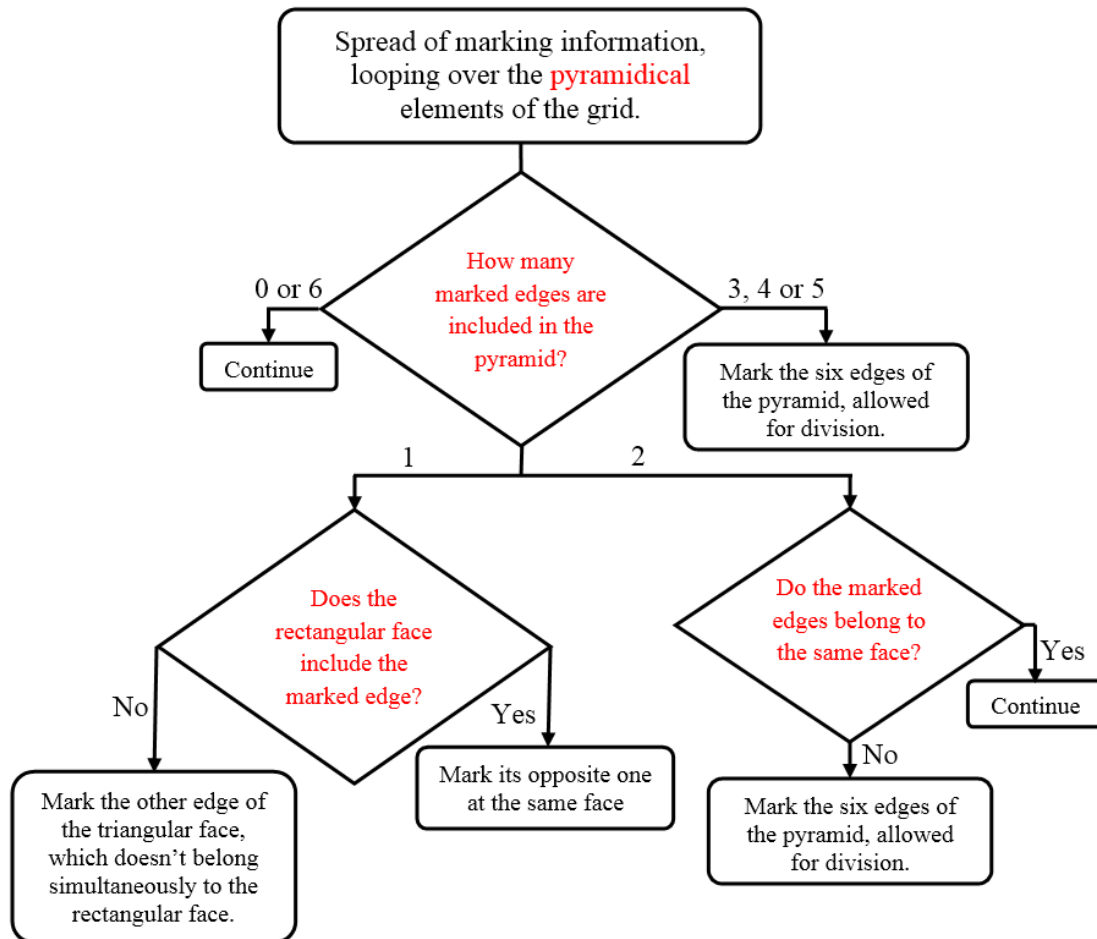


Figure 6.11: Flow chart for the spread of marking information among the edges of the pyramidal elements.

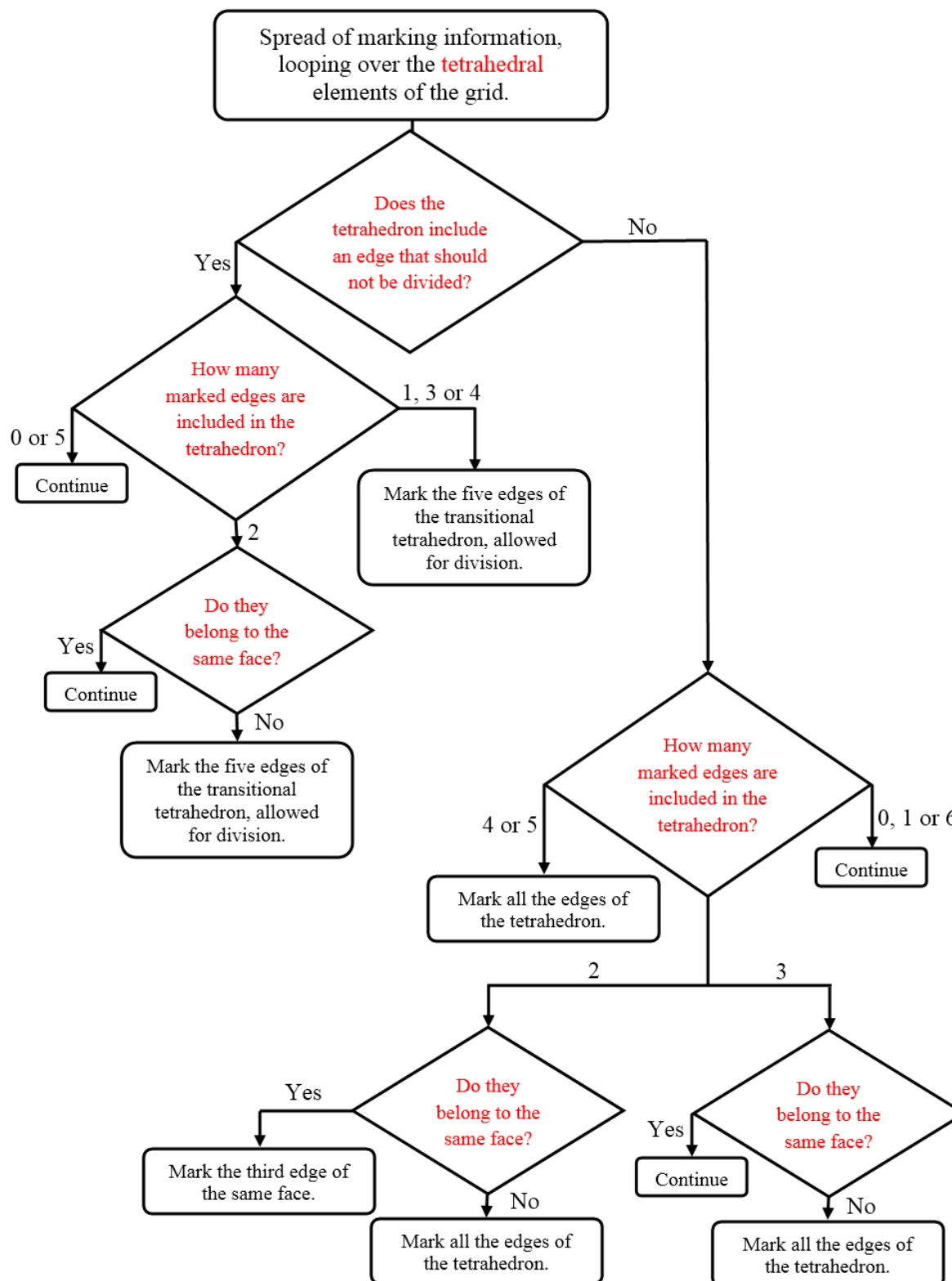


Figure 6.12: Flow chart for the spread of marking information among the edges of the tetrahedral elements.

"Intentionally left blank"

7. Fluid flow numerical results

7.1. Definitions

The developed fluid flow solver has been validated (especially in terms of accuracy and efficiency) against two- and three-dimensional benchmark test cases, while its results have been compared with experimental data as well as with the results obtained by other solvers. The compared solutions concern mainly the distributions of pressure coefficient, defined as [Abb49, Mun98]

$$C_p = \frac{2(p - p_\infty)}{\rho_\infty \vec{V}_\infty^2} \quad (7.1)$$

where p denotes the local pressure, while p_∞ , ρ_∞ and \vec{V}_∞ the pressure, density and velocity at far-field respectively. In problems considering fluid flow over aircrafts, the lift and drag coefficients are also used for comparison reasons; they can be computed as follows [Abb49, Mun98]

$$C_L = \frac{2L}{\rho_\infty \vec{V}_\infty^2 S} \quad (7.2)$$

$$C_D = \frac{2D}{\rho_\infty \vec{V}_\infty^2 S} \quad (7.3)$$

where S is a characteristic area of the aircraft (usually the projection of the wing area on the horizontal plane). The total lift L and drag D forces can be evaluated by a single loop over the boundary nodes or faces, using the (pressure and viscous) stress tensor (near-field analysis) [Vos10, Vos13]. Besides the dimensionless pressure and Mach number contours, the isentropic Mach number can be utilized for the visualization of the extracted solutions, which is defined as [Mun98]

$$M_{isentropic} = \sqrt{\left(\left(\frac{p_o^\infty}{p} \right)^{\frac{\gamma}{\gamma-1}} - 1 \right) \cdot \frac{2}{\gamma-1}} \quad (7.4)$$

where p denotes the local static pressure, while p_o^∞ the total pressure at the far-field; for the latter the value at the inlet boundary surface is usually assigned. In order to validate the present method's computational performance, the density residual is computed at each cycle (of the multigrid scheme) or at each iteration (of the single-grid scheme) as [Lyg13c, Lyg14f]

$$residual = \frac{\sum_{p=1}^{N_p} |\rho_p^{l+1} - \rho_p^l|}{N_p} \quad (7.5)$$

where l is the number of the current multigrid cycle or single-grid iteration, while N_p is the number of nodes of the initial finest mesh; a corresponding expression is employed for the residual of turbulent kinetic energy in case of turbulent flows.

This section is continued with the evaluation of the proposed methodology against test cases, considering inviscid, viscous laminar and viscous turbulent flows over wings and aircrafts.

7.2. Inviscid flow over a rectangular wing with a NACA0012 airfoil

The first (quasi-3D) test case considers inviscid flow with Mach number 0.8 and angle of attack 0° over a rectangular wing with a NACA0012 airfoil [Bae09, Lyg11, Lyg14c]. The utilized grid is composed of $625,250$ nodes and $3,500,243$ tetrahedra, while for parallel computation on a DELL T7500 workstation with two Intel^(R) Xeon^(R)-X5550 four-core processors at 2.67 GHz it was divided in four partitions; Figure 7.1 illustrates the employed grid as well as its density on symmetry surface. Accuracy improvement was succeeded with the incorporated second-order spatial accurate scheme, coupled with Van Albada-Van Leer limiter. Both an explicit (Runge-Kutta method) and implicit (Jacobi algorithm) scheme were implemented for the iterative solution of the problem (without of course any difference in their final results); for the first method a CFL number equal to unity was employed, while for the second one CFL was linearly increased up to 5.0 . The parameters of the simulation are summarized in the following Table 7.1.

Table 7.1: Parameters of simulation (inviscid flow over a rectangular wing with a NACA0012 airfoil).

Parameters	
Type of flow	Inviscid
Mach number	0.8
Reynolds number	-
Angle of attack (deg.)	0°
Grid density	$625,250$ nodes and $3,500,243$ tetrahedra
Number of partitions	4
Iterative scheme	Runge-Kutta method (CFL= 1.0) and Jacobi algorithm (CFL= $1.0-5.0$)
Computational system	DELL T7500 workstation with two Intel ^(R) Xeon ^(R) -X5550 four-core processors at 2.67 GHz

Nevertheless, the aforementioned grid was identified to be insufficient for the accurate capturing of flow phenomena, especially those located at shock formations. Therefore, the developed h-refinement methodology was applied, resulting in an increased number of DoF's and consequently in improved accuracy at the shock region; its implementation was based on a criterion automatically targeting and enriching areas with value of Mach number between 0.95 and 1.05 . The new mesh, whose density on the mirroring surface is presented in Figure 7.2, compared with the initial one, consists of $1,084,667$ nodes and $6,171,465$ tetrahedra. The iterative solution was continued, since the up to that moment computed primitive variables were interpolated to the adapted grid; in that way the generation of a new mesh from scratch was avoided and consequently significant computational savings were achieved.

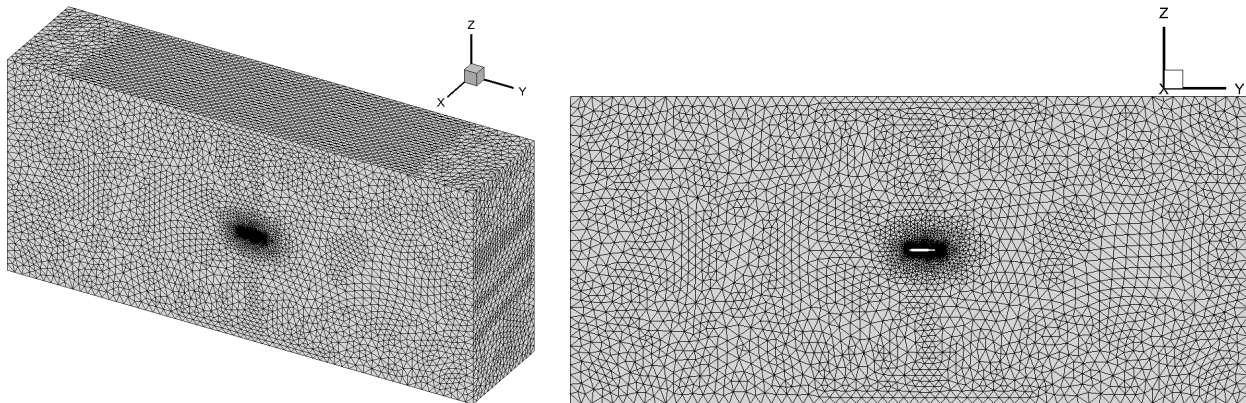


Figure 7.1: Employed grid and its density on symmetry surface (inviscid flow over a rectangular wing with a NACA0012 airfoil).

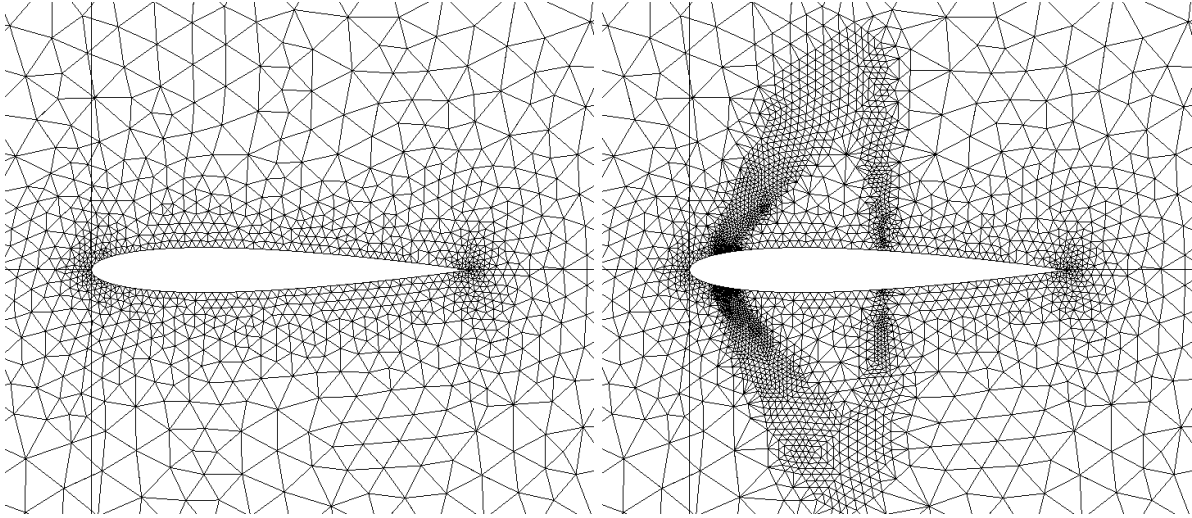


Figure 7.2: Mesh density on the symmetry surface prior (left) and after (right) h -refinement (inviscid flow over a rectangular wing with a NACA0012 airfoil).

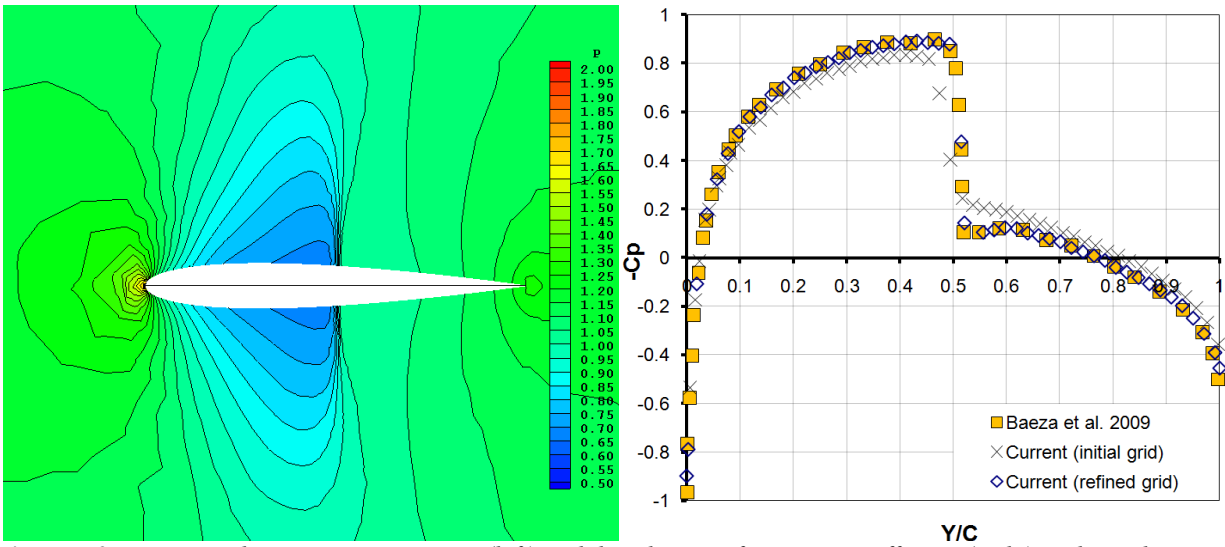


Figure 7.3: Dimensionless pressure contours (left) and distribution of pressure coefficient (right) at the mid-span of the wing (inviscid flow over a rectangular wing with a NACA0012 airfoil).

Figure 7.3 illustrates the obtained dimensionless pressure contours at the mid-span of the refined wing, as well as the distributions of pressure coefficient C_p along the same slice for both grids (initial and refined), compared with the corresponding one of Baeza et al. [Bae09]. A sufficient agreement is identified between the reference and the extracted by the adapted grid results, indicating the potential of the developed solver for such simulations. Additionally, the mis-capturing of the shock by the initial grid, compared to the improved shock prediction by the refined one, demonstrates the capability of h -refinement methodology for improving the solution's accuracy.

In order to evaluate the developed multigrid methodology, three-coarser resolutions of the initial mesh were created via isotropic agglomeration. In Figure 7.4 far views of the initial and agglomerated control volume grids are illustrated, while Figure 7.5 includes the corresponding close-up views at symmetry surface. In case of the explicit scheme, iterations/cycles and time speed-up coefficients equal to ~ 10.5 and ~ 8.8 respectively were succeeded for a final residual equal to $1.0E-10$ (only FAS); in Figure 7.6 the density convergence histories per number of iterations/cycles and time are presented for the single-grid and multigrid simulations. Figure 7.7 illustrates the corresponding history for the runs with the

implicit scheme, performed on a workstation with an AMD FX^(tm)-8350 eight-core processor at 4.0 GHz; iterations/cycles and temporal acceleration coefficients equal to ~ 14.0 and ~ 3.5 were obtained (only FAS). Despite that a higher acceleration was achieved in case of the explicit scheme, significantly reduced wall-clock computation time (~ 0.8 hours) was achieved by the implicit method, mainly due to the increased CFL value. Independently of the employed scheme, the aforementioned coefficients reveal the proposed multigrid method's potential for considerably improved computational performance.

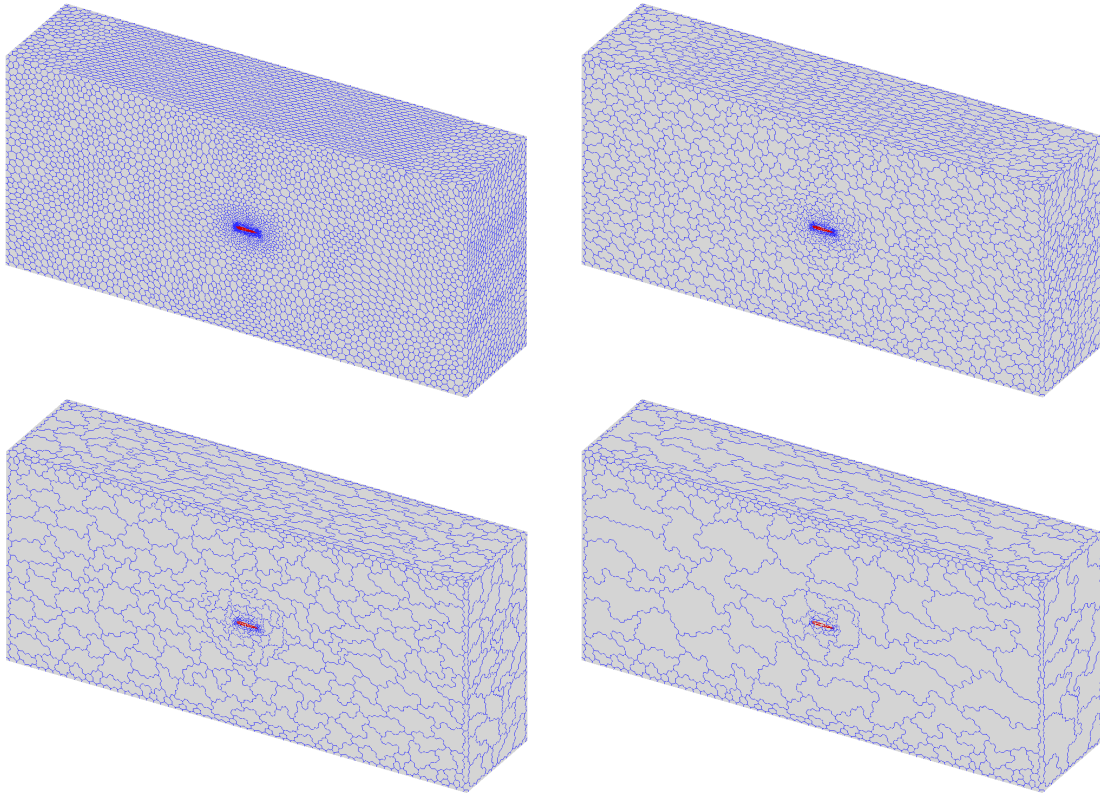


Figure 7.4: Initial and agglomerated control volume grids (inviscid flow over a rectangular wing with a NACA0012 airfoil).

7.3. Laminar flow over a rectangular wing with a NACA0012 airfoil

The second (quasi-3D) problem encountered, concerns laminar flow with 0° angle of attack, Mach number 0.85 and Reynolds number 500 over a rectangular wing with a NACA0012 airfoil [Mit98, Lyg11, Lyg14c]. The utilized hybrid grid consists of 305,978 nodes, 566,245 tetrahedra and 394,760 prisms, the latter located on the solid wall region (wing) to allow for the effective prediction of boundary layer region; Figure 7.8 illustrates the employed grid as well as its density on the symmetry surface. For the computation of velocity components' gradients, which are necessary to viscous terms' evaluation, the nodal-averaging method was followed, while for the inviscid fluxes the incorporated second-order accurate spatial scheme was accompanied by the Min-mod limiter. Both an explicit (Runge-Kutta method) and implicit (Jacobi algorithm) scheme were implemented for the iterative solution of the problem (without of course any difference in their final results); for the first method a CFL number equal to unity was employed, while for the second one CFL was linearly increased up to 2.0. In order to accelerate the solution procedure on a DELL T7500 workstation with two Intel^(R) Xeon^(R)-X5650 six-core processors at 2.67 GHz, the developed parallelization strategy, as well as the incorporated agglomeration multigrid scheme were implemented (FMG-FAS). Therefore, the initial grid was divided in four sub-domains, for which three-coarser resolutions were generated via directional agglomeration. In Figure 7.9 far views of the initial and agglomerated control volume grids are illustrated, while Figure 7.10 includes

the corresponding close-up views of the symmetry surface, where the topology-preserving framework, according to which the proposed multigrid methodology is applied, is clearly demonstrated. The parameters of the simulation are summarized in the following Table 7.2.

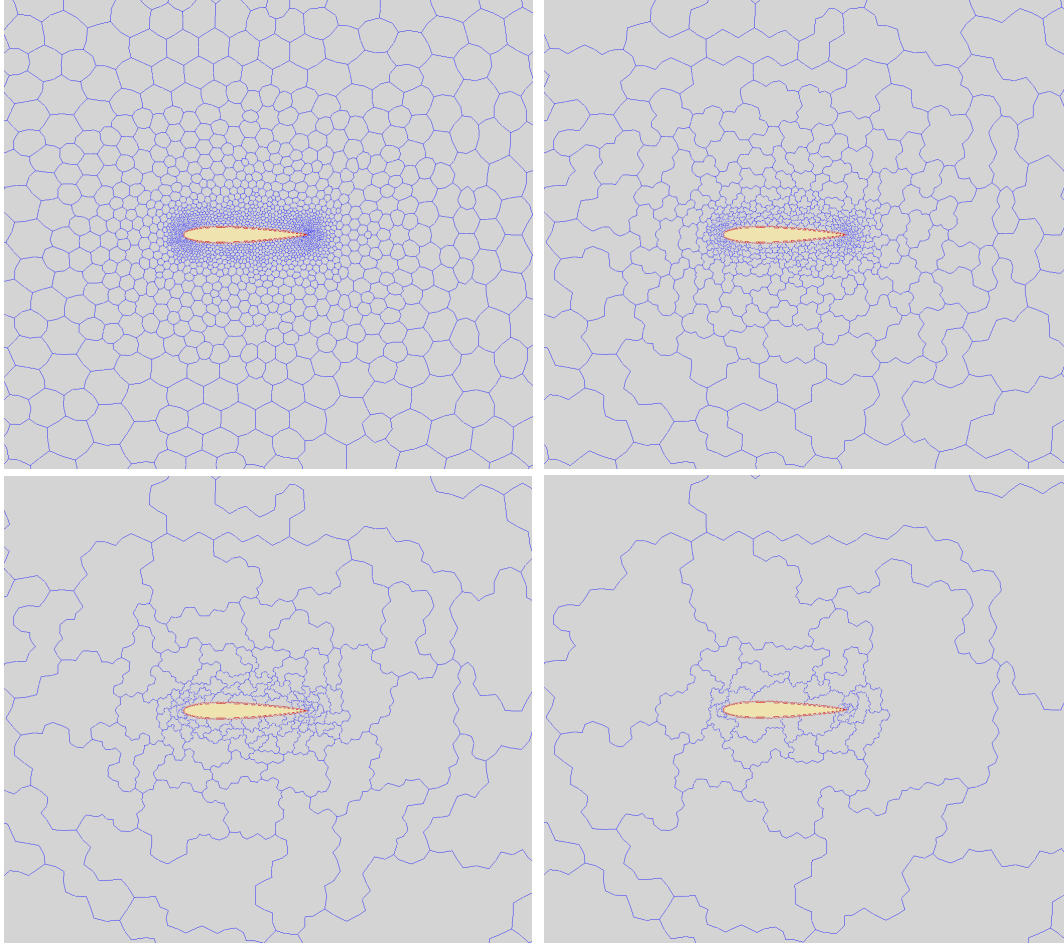


Figure 7.5: Mesh density on the symmetry surface of the initial and agglomerated control volume grids (inviscid flow over a rectangular wing with a NACA0012 airfoil).

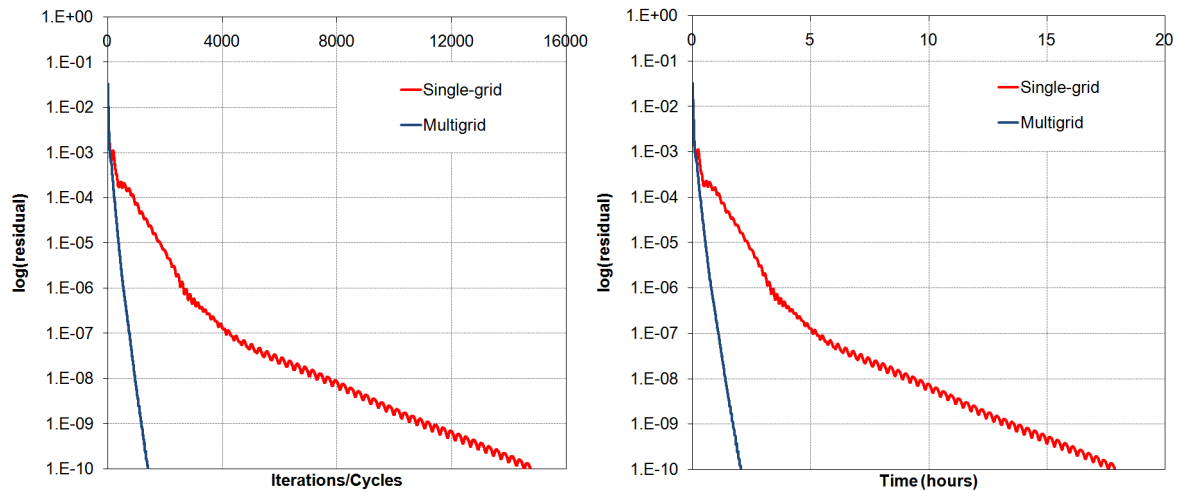


Figure 7.6: Density convergence history per iterations/cycles and time (explicit scheme, inviscid flow over a rectangular wing with a NACA0012 airfoil).

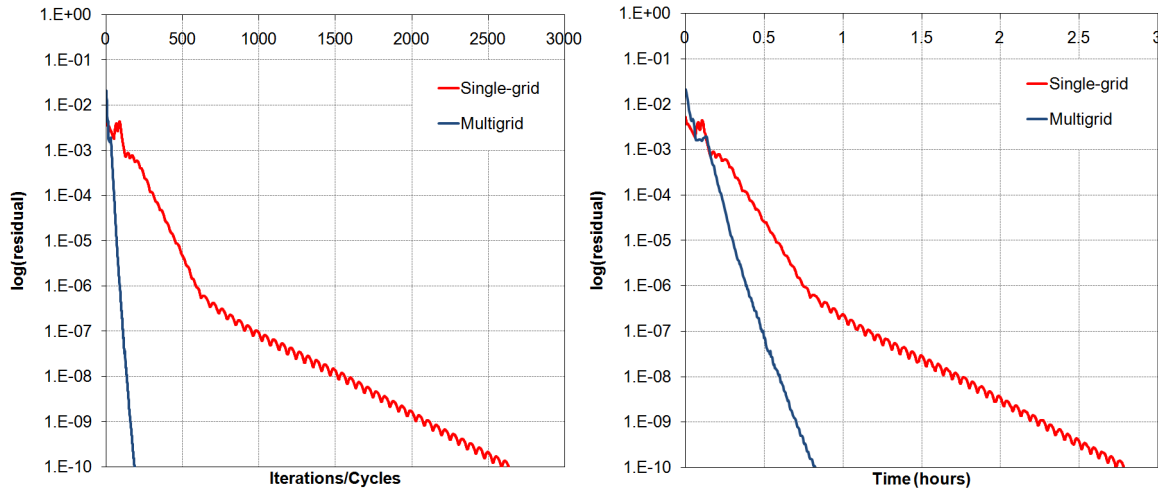


Figure 7.7: Density convergence history per iterations/cycles and time (implicit scheme, inviscid flow over a rectangular wing with a NACA0012 airfoil).

In Figure 7.11 the extracted Mach number contours at the mid-span of the wing are presented along with the distribution of pressure coefficient C_p at the same slice, compared with the corresponding one of Mittal [Mit98]. The obtained results agree very well with those of the reference solver, indicating the capability of the proposed methodology for such simulations in terms of accuracy.

Table 7.2: Parameters of the simulation (laminar flow over a rectangular wing with a NACA0012 airfoil).

Parameters	
Type of flow	Laminar
Mach number	0.85
Reynolds number	500
Angle of attack (deg.)	0°
Grid density	305,978 nodes, 566,245 tetrahedra and 394,760 prisms
Number of partitions	4
Number of multigrid levels	4
Iterative scheme	Runge-Kutta method (CFL=1.0) and Jacobi algorithm (CFL=1.0-2.0)
Computational system	DELL T7500 workstation with two Intel ^(R) Xeon ^(R) -X5650 six-core processors at 2.67 GHz

Besides the initial multigrid runs (explicit and implicit), single-grid simulations were also performed, revealing the superiority of the former in terms of efficiency. In case of the explicit scheme, iterations/cycles and time speed-up coefficients equal to ~ 7.2 and ~ 6.0 respectively were achieved for a final residual equal to $1.0E-9$; in Figure 7.12 the density convergence histories per number of iterations/cycles and time are presented for the single-grid and the four-level multigrid simulation. In addition, a two- and three-level FMG-FAS multigrid run was performed, with its convergence history included in the same figure; as expected, additional acceleration is obtained for any additional spatial multigrid level. Similarly Figure 7.13 shows the corresponding history for the implicit scheme; iterations/cycles and temporal acceleration coefficients equal to ~ 12.8 and ~ 3.9 were succeeded considering a three-level multigrid scheme.

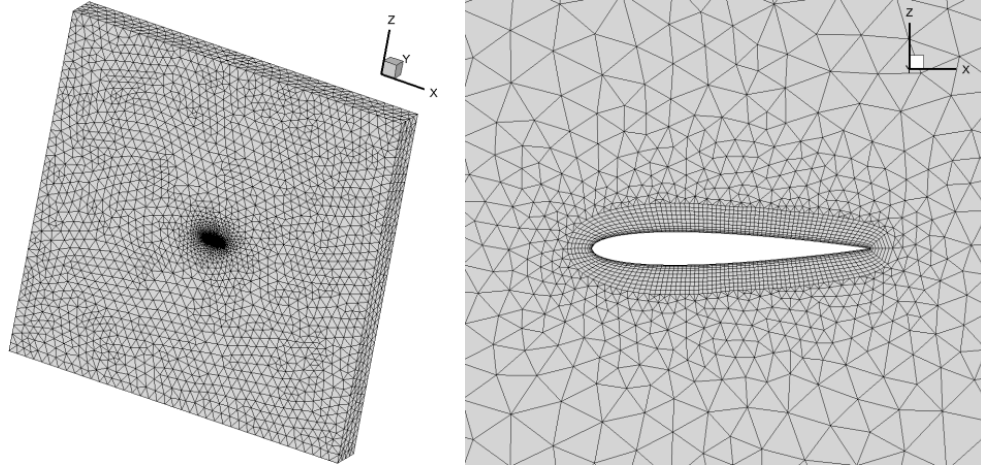


Figure 7.8: Employed grid and its density on the symmetry surface (laminar flow over a rectangular wing with a NACA0012 airfoil).

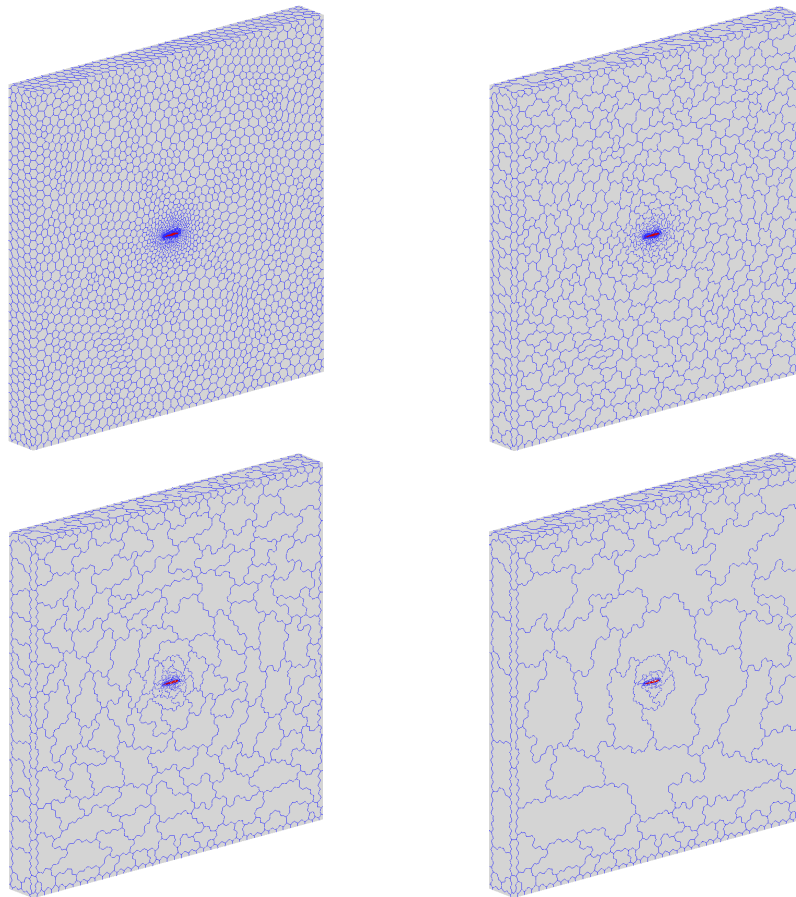


Figure 7.9: Initial and agglomerated control volume grids (laminar flow over a rectangular wing with a NACA0012 airfoil).

In addition, two more simulations (single-grid and 4-level multigrid, only FAS) were performed, implementing the incorporated explicit scheme along with the element-based method (edge-dual volume approach) for the velocity components and temperature gradients' computation; CFL number was set equal to unity. In Figure 7.14 the density convergence history per number of iterations and time is presented for these runs; a temporal speed-up coefficient equal to ~ 7.1 was achieved. Although a larger

time speed-up coefficient was succeeded, comparing to the corresponding simulations with nodal-averaging scheme, the wall clock computation time was significantly increased (~ 3.3 hours); coefficients per number of iterations are equal (~ 7.2) for both sets of simulations. Thus, the superior computational efficiency of the nodal-averaging scheme, compared to the edge-dual volume one, is demonstrated independently of implementing or not the multigrid methodology.

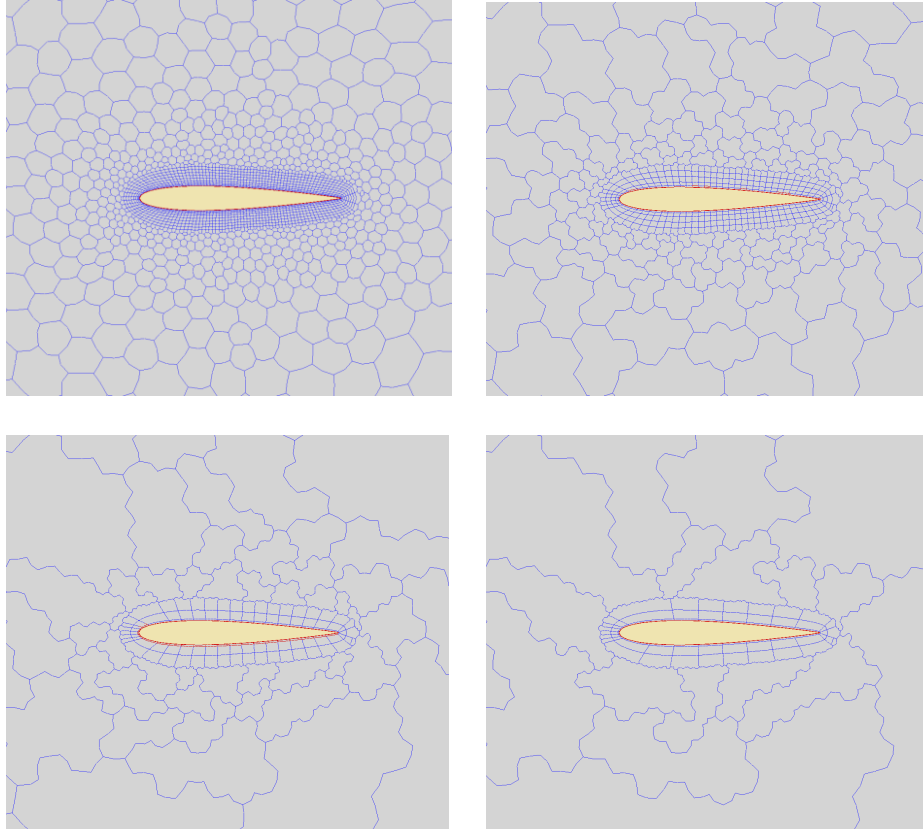


Figure 7.10: Mesh density on the symmetry surface of the initial and agglomerated control volume grids (laminar flow over a rectangular wing with a NACA0012 airfoil).

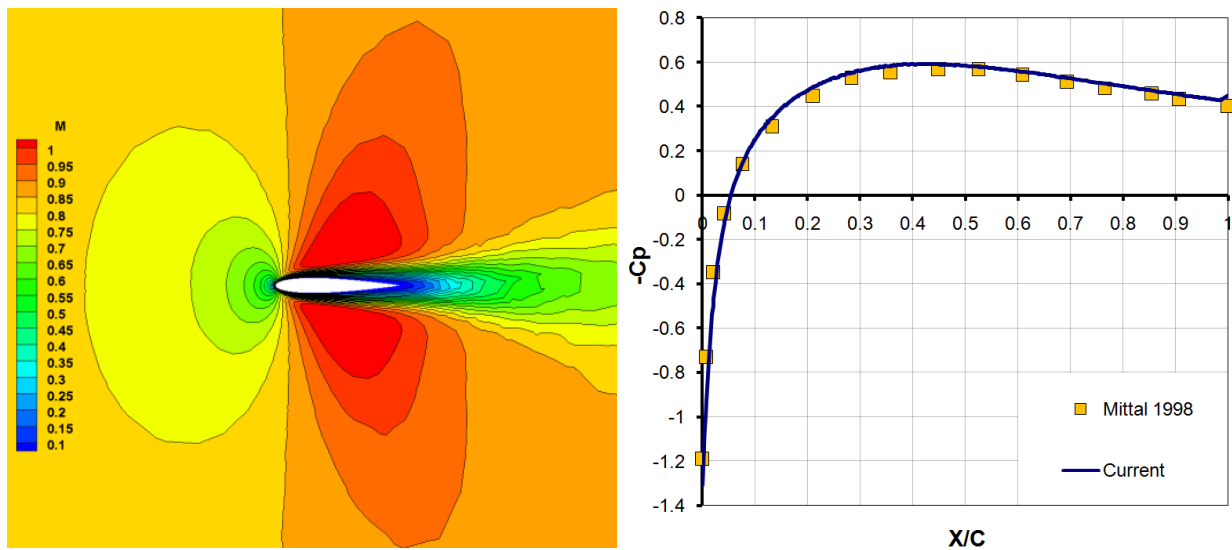


Figure 7.11: Mach number contours (left) and distribution of pressure coefficient (right) at the mid-span of the wing (laminar flow over a rectangular wing with a NACA0012 airfoil).

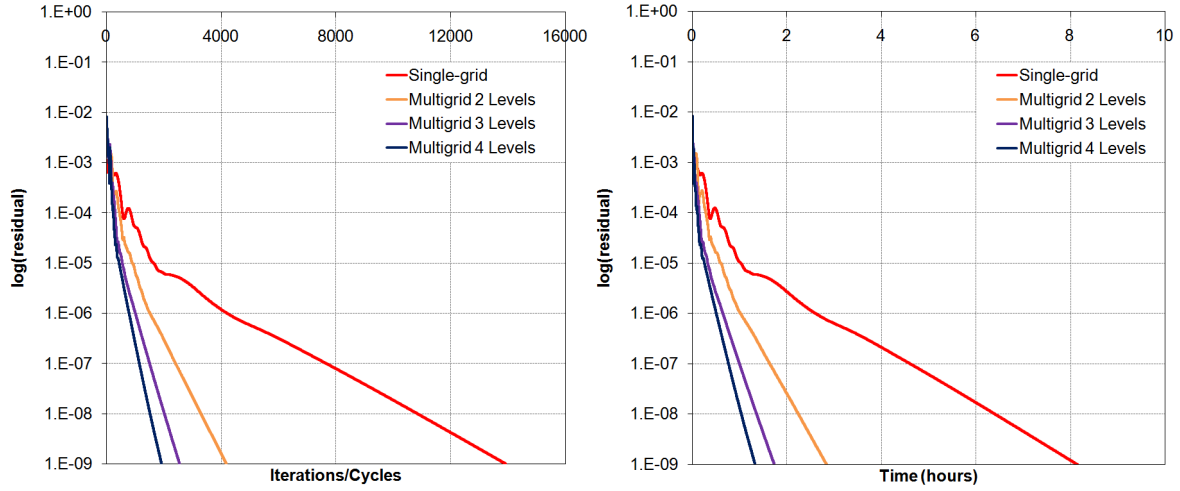


Figure 7.12: Density convergence history per iterations/cycles and time (2-, 3- and 4-level FMG-FAS multigrid explicit scheme, CFL=1.0, laminar flow over a rectangular wing with a NACA0012 airfoil).

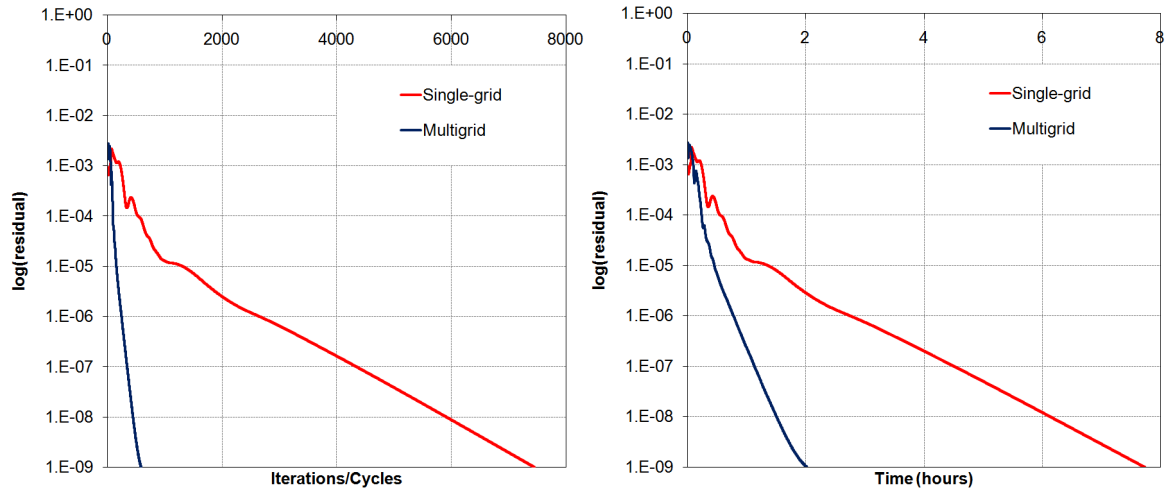


Figure 7.13: Density convergence history per iterations/cycles and time (3-level FAS multigrid implicit scheme CFL=1.0-2.0, laminar flow over a rectangular wing with a NACA0012 airfoil).

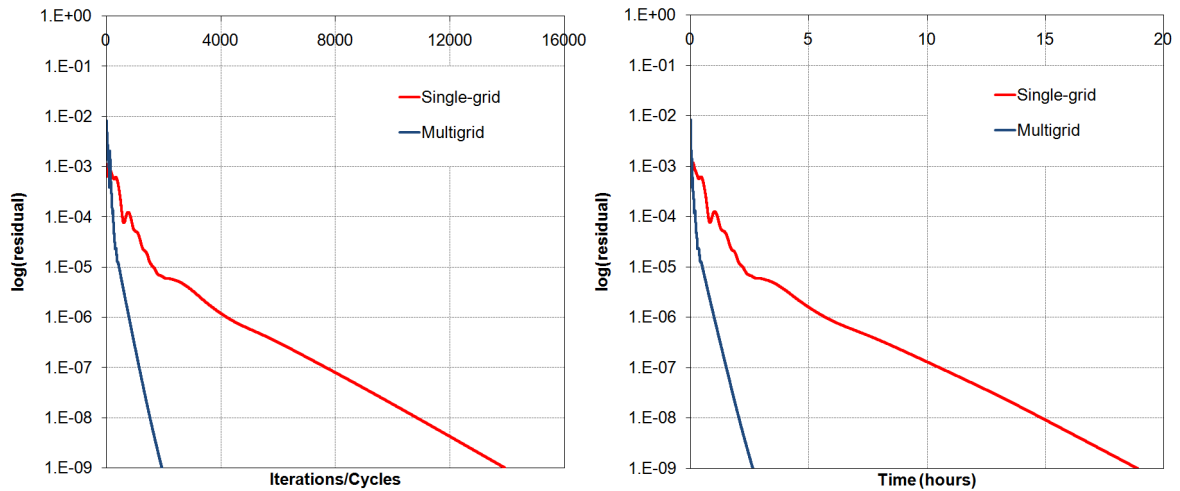


Figure 7.14: Density convergence history per iterations/cycles and time (explicit scheme, element-based method, laminar flow over a rectangular wing with a NACA0012 airfoil).

7.4. Turbulent flow over a rectangular wing with a RAE2822 airfoil

The third (quasi-3D) benchmark test case considers "steady-state" fully turbulent flow over a rectangular wing with a RAE2822 airfoil [Lyg11]; experimental data for this problem have been published by Cook et al. [Coo79]. The Mach and Reynolds numbers are set equal to 0.725 and $6.5E+6$ respectively, while the angle of attack is set to 2.6° in order to account for the side effects of the wind tunnel. The employed hybrid mesh is composed of $683,346$ nodes, $2,165,362$ tetrahedra, $587,465$ prisms and $5,201$ pyramids, while for parallel solution of the problem on a DELL T7400 workstation with two Intel^(R) Xeon^(R)-E5410 four-core processors at 2.33 GHz, it was divided in eight sub-domains; in Figure 7.15 the utilized grid as well as its density on the symmetry surface is presented. For the iterative approximation of the solution the second-order time accurate Runge-Kutta method was employed along with a CFL number equal to 0.5 . The gradients at control volumes interfaces, required for the calculation of viscous fluxes, were obtained with the edge-dual volume approach, while inviscid fluxes' accuracy was improved with a second-order accurate spatial scheme, coupled with the Van Albada-Van Leer limiting function. The parameters of the simulation are summarized in the following Table 7.3.

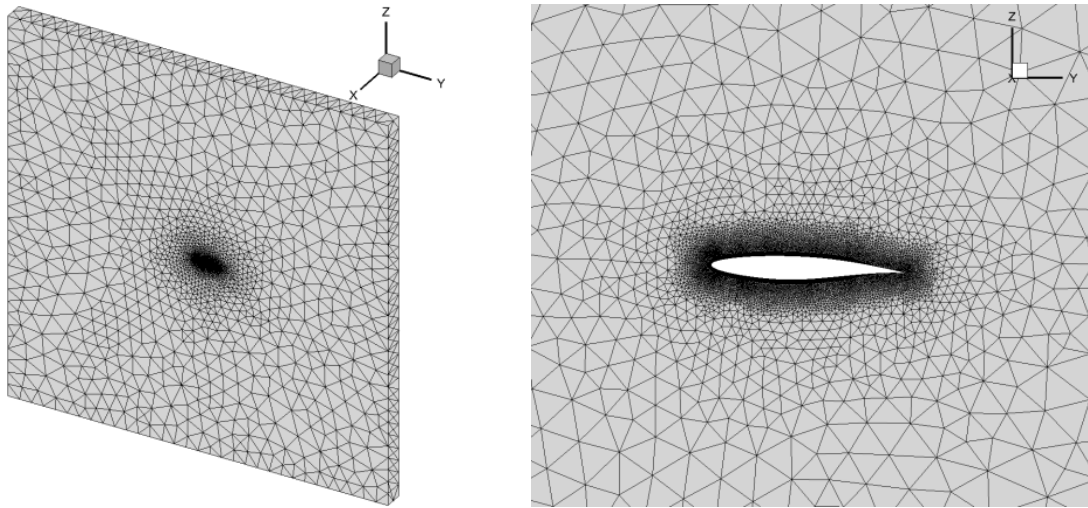


Figure 7.15: Employed grid and its density on the symmetry surface (turbulent flow over a rectangular wing with a RAE2822 airfoil).

Table 7.3: Parameters of simulation (turbulent flow over a rectangular wing with a RAE2822 airfoil).

Parameters	
Type of flow	Turbulent
Mach number	0.725
Reynolds number	$6.5E+6$
Angle of attack (deg.)	2.6°
Grid density	$683,346$ nodes, $2,165,362$ tetrahedra, $587,465$ prisms and $5,201$ pyramids
Number of partitions	8
Iterative scheme	Runge-Kutta method (CFL= 0.5)
Computational system	DELL T7400 workstation with two Intel ^(R) Xeon ^(R) -E5410 four-core processors at 2.33 GHz

The main goal of this test case is the evaluation of the performance of the three incorporated in the proposed algorithm turbulence models (without any transition scheme), namely $k-\epsilon$ with wall functions (WF1 and WF2), $k-\omega$ and SST . In Figure 7.16 the derived by the $k-\omega$ model dimensionless pressure

contours at the mid-span of the wing are depicted, while Figure 7.17 illustrates the pressure coefficient distributions at the same slice for all the employed models, compared with the corresponding results of Cook et al. [Coo79]; in general all the applied turbulence models provided equally satisfactory results, agreeing very well with the experimental data.

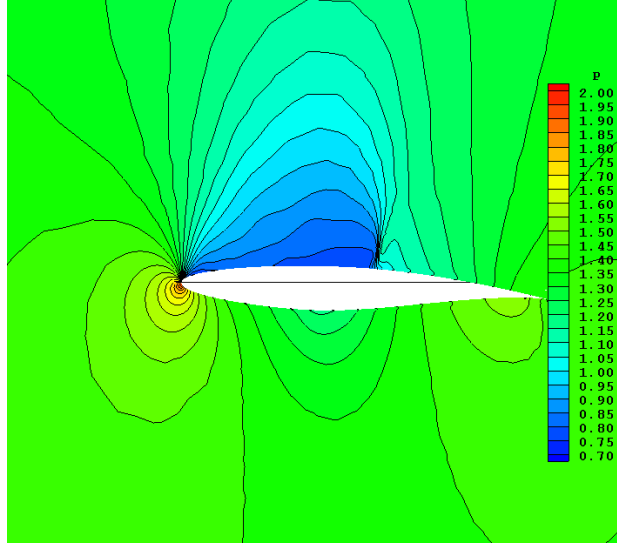


Figure 7.16: Dimensionless pressure contours at the mid-span of the wing predicted with $k-\omega$ model (turbulent flow over a rectangular wing with a RAE2822 airfoil).

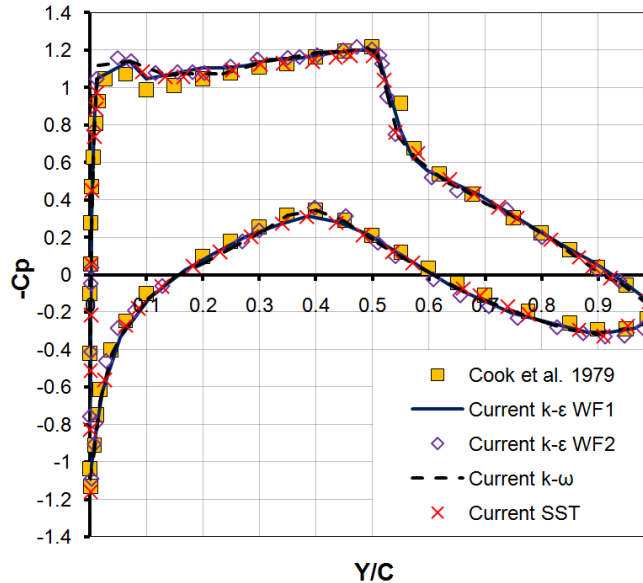


Figure 7.17: Distribution of pressure coefficient at the mid-span of the wing (turbulent flow over a rectangular wing with a RAE2822 airfoil).

Besides the effectiveness of turbulence modelling, the scalability provided by the adopted parallelization strategy was evaluated with this test case, utilizing a DELL T7500 workstation with two Intel^(R) Xeon^(R)-X5650 six-core processors at 2.67 GHz. Therefore, the sub-case including $k-\omega$ model was calculated in a parallel environment of two, four, eight and twelve processors, deriving the speed-up per number of processors distribution, illustrated in Figure 7.18; the relatively slight deviation between the ideal and the current computation derives from the increase of communication load with the increase of employed processors.

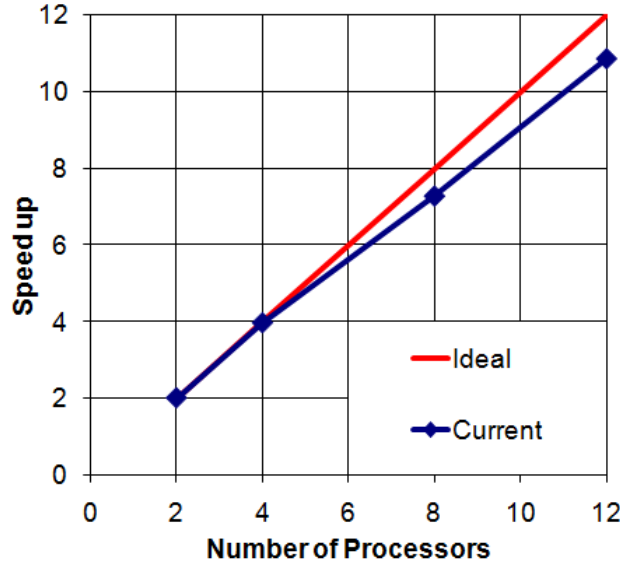


Figure 7.18: Scalability of parallel implementation ($k-\omega$ model, turbulent flow over a rectangular wing with a RAE2822 airfoil).

7.5. Turbulent flow over a rectangular wing with a NACA0012 airfoil

A quasi-3D test case involving "steady-state" fully turbulent flow over a rectangular wing with a NACA0012 airfoil [Per07] was used for the evaluation of the $k-\varepsilon$ model blended with the one-equation model at the solid wall region. The Mach and Reynolds numbers are assumed equal to 0.85 and $1.85E+6$ respectively, while the angle of attack is set to 0° . The utilized hybrid grid consists of $1,900,703$ nodes, $3,017,043$ tetrahedra, $2,702,647$ prisms and $4,154$ pyramids, while for its employment on the same to the previous test case computational system it was divided in eight partitions; in Figure 7.19 the employed grid as well as its density on the symmetry surface is presented.

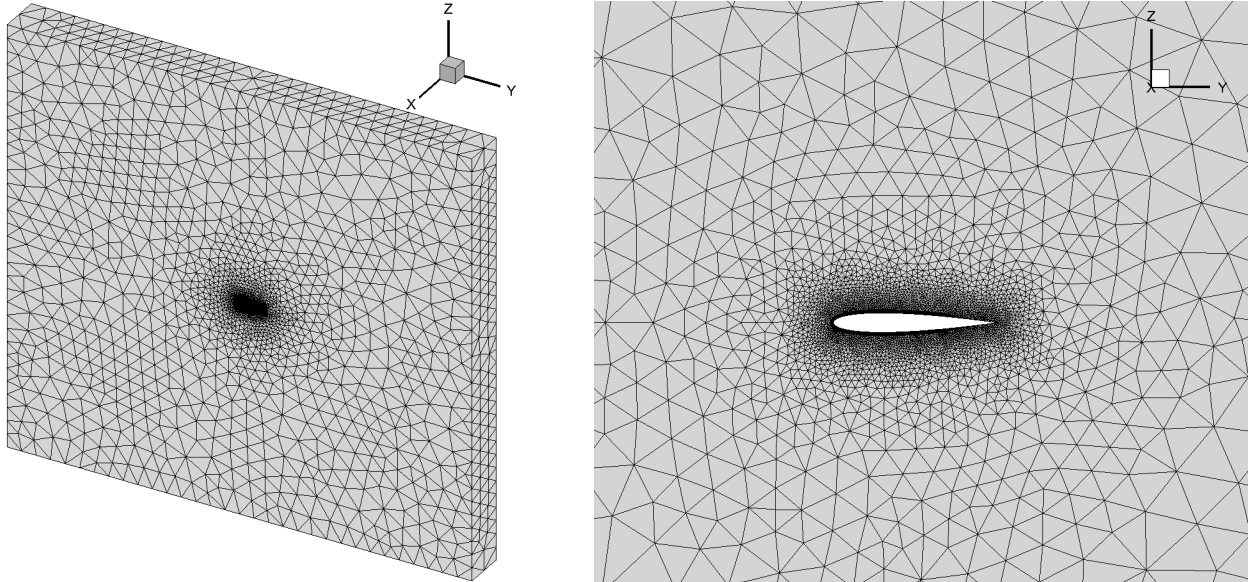


Figure 7.19: Employed grid and its density on the symmetry surface (turbulent flow over a rectangular wing with a NACA0012 airfoil).

For the iterative approximation of the final steady-state solution the second-order accurate in time Runge-Kutta method was employed along with a CFL number equal to 0.5 . The gradients at the middle of

each edge, required for the calculation of viscous fluxes, were obtained with the edge-dual volume method, while inviscid fluxes' accuracy was improved with a second-order accurate spatial scheme, coupled with the Van Albada-Van Leer limiting function. The parameters of the simulation are summarized in Table 7.4. In Figure 7.20 the obtained Mach number contours at the mid-span of the wing are illustrated along with the derived distribution of pressure coefficient at the same slice, compared with the corresponding one of Persson [Per07]; a satisfactory agreement is obtained between the results of the present and the reference solver.

Table 7.4: Parameters of simulation (turbulent flow over a rectangular wing with a NACA0012 airfoil).

Parameters	
Type of flow	Turbulent
Mach number	0.85
Reynolds number	$1.85E+6$
Angle of attack (deg.)	0°
Grid density	1,900,703 nodes, 3,017,043 tetrahedra, 2,702,647 prisms and 4,154 pyramids
Number of partitions	8
Iterative scheme	Runge-Kutta method (CFL=0.5)
Computational system	DELL T7400 workstation with two Intel ^(R) Xeon ^(R) -E5410 four-core processors at 2.33 GHz

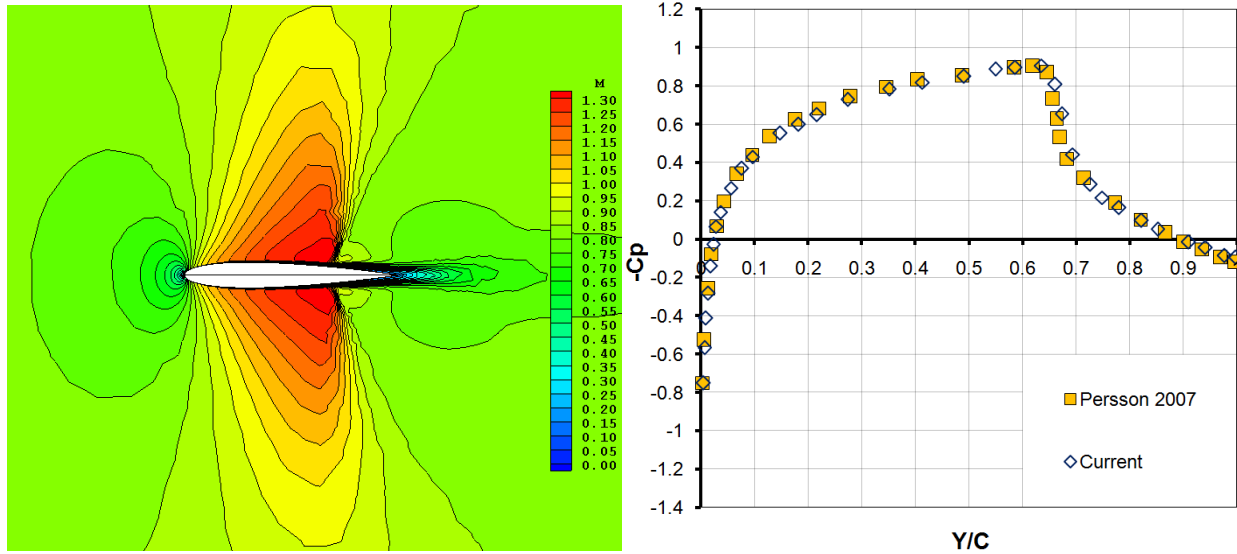


Figure 7.20: Mach number contours (left) and distribution of pressure coefficient (right) at the mid-span of the wing (turbulent flow over a rectangular wing with a NACA0012 airfoil).

7.6. Turbulent flow over the ONERA M6 wing

The next benchmark problem encountered in this study considers the prediction of three-dimensional "steady-state" fully turbulent flow over the ONERA M6 wing [Sch79, Sor03, Ara10, Lyg13a, PAB3D, WIND]. The free-stream Mach number is assumed equal to 0.8395, the angle of attack equal to 3.06° and the Reynolds number (based on the mean aerodynamic chord) equal to $11.72E+6$. The generated hybrid grid includes 4,676,354 nodes, 7,231,670 tetrahedra and 6,775,840 prisms, while for its utilization on the parallel computational environment, utilizing a DELL T7500 workstation with two Intel^(R) Xeon^(R)-X5650 six-core processors at 2.67 GHz, it was decomposed in twelve partitions; Figure 7.21 illustrates the

utilized grid as well as its density on the symmetry surface. For the iterative solution the four-stage Runge-Kutta method was implemented with a CFL number equal to unity, while for turbulence prediction the *SST* model was employed along with an implicit source term treatment. The gradients, required for the calculation of viscous fluxes, were obtained with the nodal-averaging scheme, while inviscid fluxes' accuracy was improved with a second-order accurate spatial scheme, coupled with Van Albada-Van Leer limiting function. The parameters of the simulation are summarized in the following Table 7.5.

Table 7.5: Parameters of simulation (turbulent flow over the ONERA M6 wing).

Parameters	
Type of flow	Turbulent (<i>SST</i>)
Mach number	0.8395
Reynolds number	11.72E+6
Angle of attack (deg.)	3.06°
Grid density	4,676,354 nodes, 7,231,670 tetrahedra and 6,775,840 prisms
Number of partitions	12
Iterative scheme	Runge-Kutta method (CFL=1.0)
Computational system	DELL T7500 workstation with two Intel ^(R) Xeon ^(R) -X5650 six-core processors at 2.67 GHz

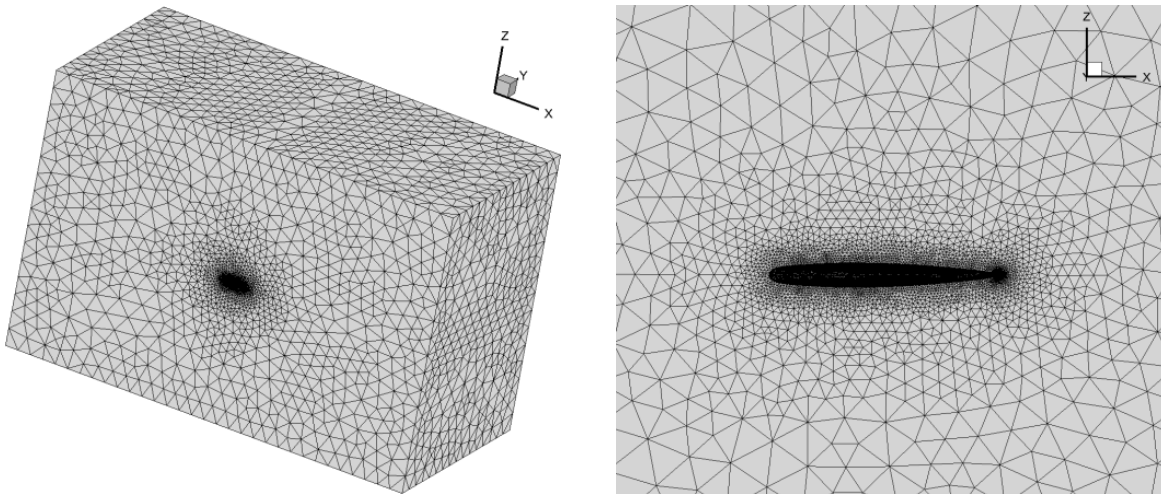


Figure 7.21: Employed grid and its density on the symmetry surface (turbulent flow over the ONERA M6 wing).

Nevertheless, this grid was identified insufficient for the accurate capturing of the flow phenomena, especially in the area of shock formations and for span-wise sections greater than 44%. Therefore, the h-refinement technique was applied at this region, based on a criterion targeting the areas with Mach number greater than 0.95; the derived adapted mesh was composed of 5,302,368 nodes, 8,315,389 tetrahedra and 7,653,360 prisms. As the second mesh proved to be also insufficient, an additional grid adaptation was implemented, targeting the shock regions defined by the available experimental data, resulting in the final grid including 6,431,378 nodes, 8,978,240 tetrahedra and 11,048,643 prisms. Figures 7.22 and 7.23 illustrate the mesh density at the wing span-wise section 40%, and the wing surface respectively prior and after h-refinement.

Figure 7.24 illustrates the predicted dimensionless pressure contours on the upper surface of the wing for the initial and the refined mesh, while Figures 7.25-7.31 include the finally obtained distributions of pressure coefficient at seven different span-wise sections of the wing, compared with the available experimental data [Sch79] as well as with the computed ones by reference solvers [Ara10, PAB3D]. The

current results appear to be in satisfactory agreement with the reference ones (experimental and computed); even more, at wing section 80% the proposed method appears to predict more accurately the shock at the upper side of the wing, comparing to the reference solvers.

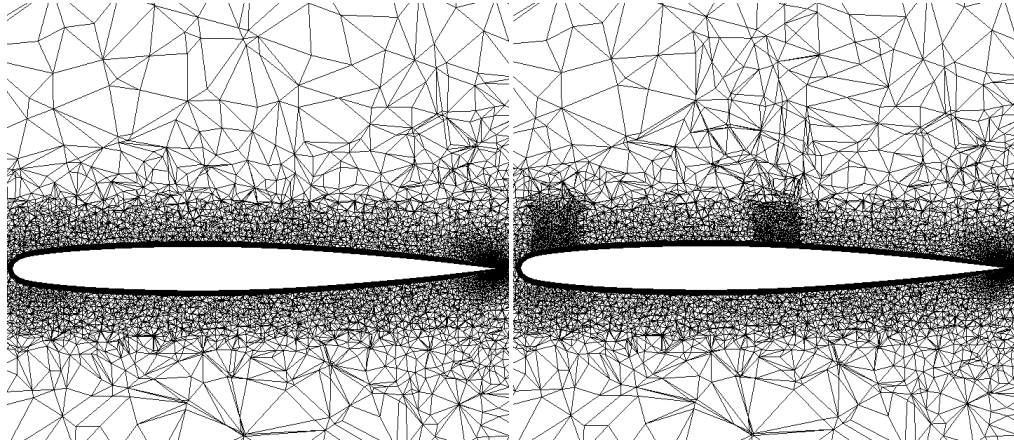


Figure 7.22: Mesh density at the wing span-wise section 40% prior (left) and after (right) h-refinement (turbulent flow over the ONERA M6 wing).

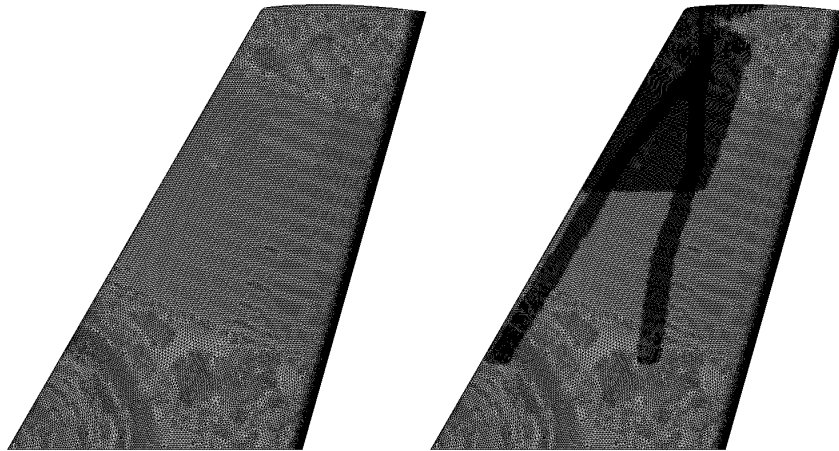


Figure 7.23: Mesh density on the wing surface prior (left) and after (right) h-refinement (turbulent flow over the ONERA M6 wing).

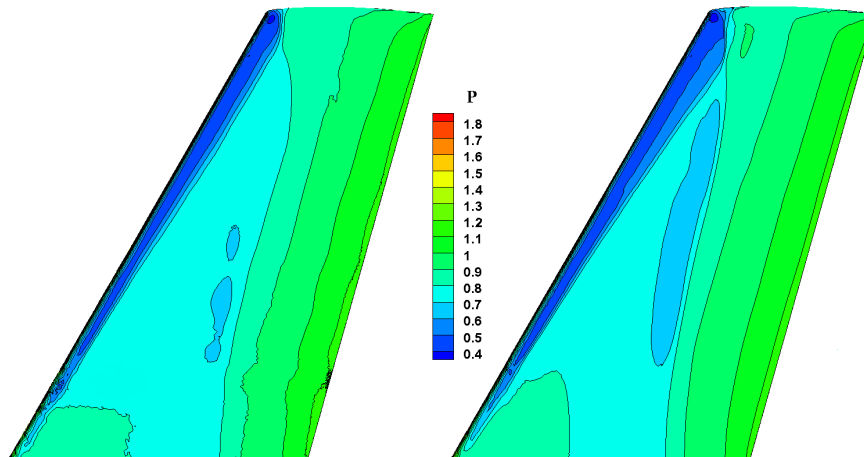


Figure 7.24: Predicted dimensionless pressure contours on the wing surface prior (left) and after (right) h-refinement (turbulent flow over the ONERA M6 wing).

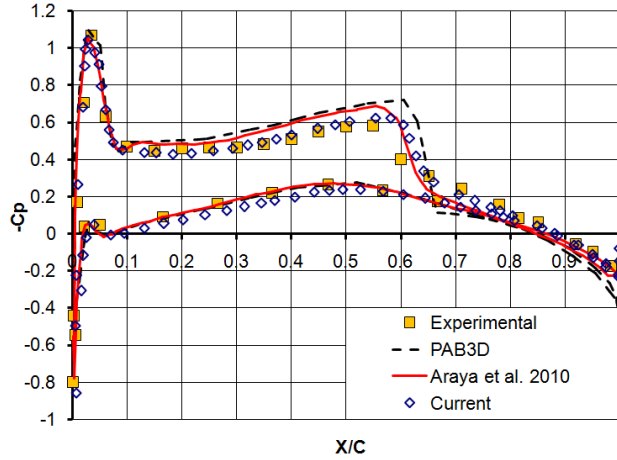


Figure 7.25: Distribution of pressure coefficient at the span-wise section 20% (turbulent flow over the ONERA M6 wing).

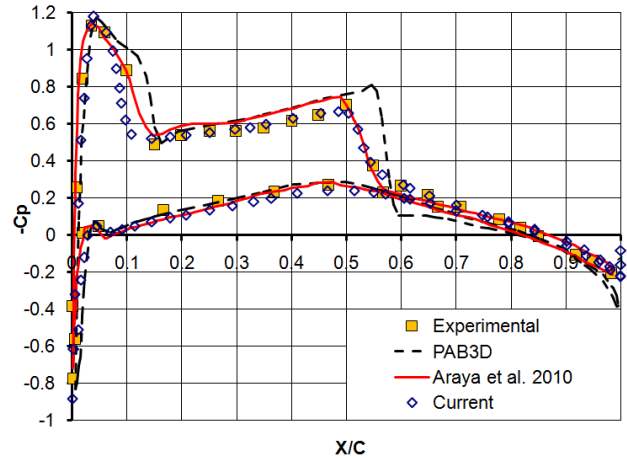


Figure 7.26: Distribution of pressure coefficient at the span-wise section 44% (turbulent flow over the ONERA M6 wing).

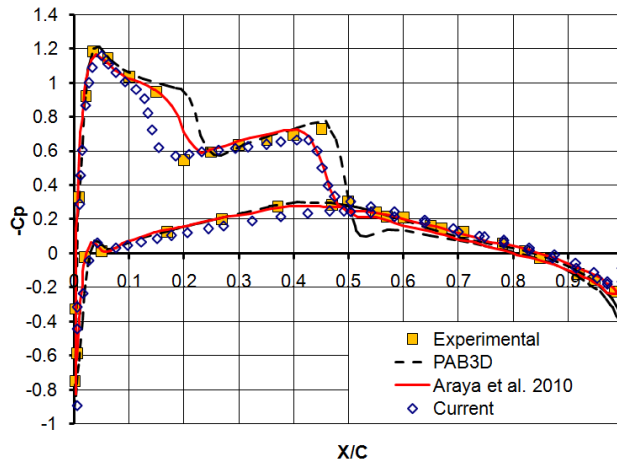


Figure 7.27: Distribution of pressure coefficient at the span-wise section 65% (turbulent flow over the ONERA M6 wing).

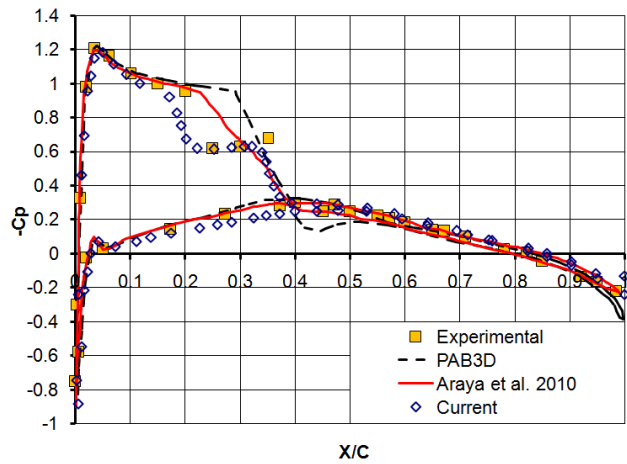


Figure 7.28: Distribution of pressure coefficient at the span-wise section 80% (turbulent flow over the ONERA M6 wing).

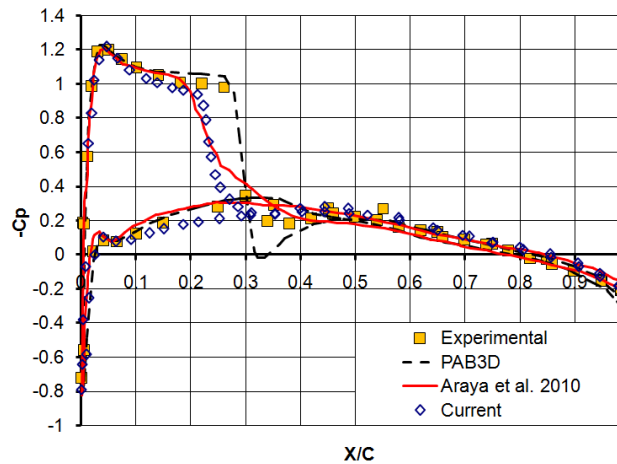


Figure 7.29: Distribution of pressure coefficient at the span-wise section 90% (turbulent flow over the ONERA M6 wing).

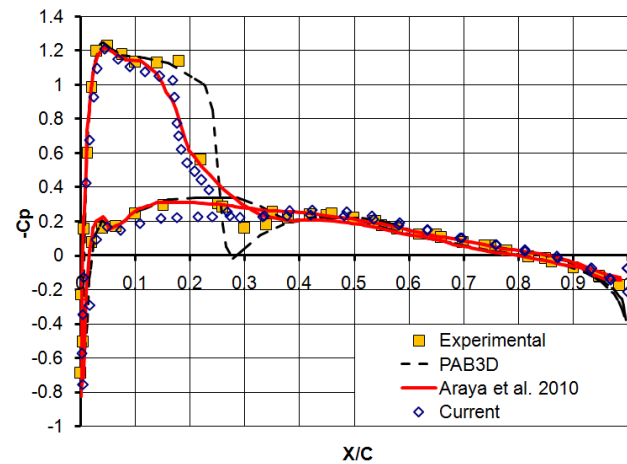


Figure 7.30: Distribution of pressure coefficient at the span-wise section 95% (turbulent flow over the ONERA M6 wing).

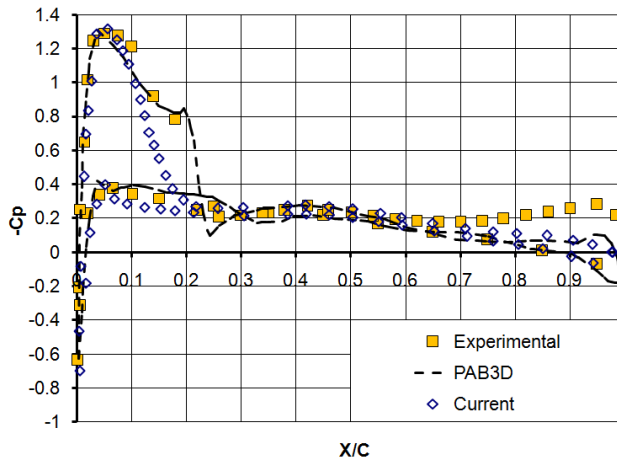


Figure 7.31: Distribution of pressure coefficient at the span-wise section 99% (turbulent flow over the ONERA M6 wing).

7.7. Turbulent flow over the DLR-F6 aircraft

A validation of the developed methodology was performed against the test cases of the second AIAA Drag Prediction Workshop (DPW II), held in Orlando in June 2003; the fully turbulent flow over the DLR-F6 aircraft with (Wing-Body-Nacelles-Pylons, WBNP) and without nacelles-pylons (Wing-Body, WB), representing a twin engine wide body aircraft of Airbus type, is examined [Bro01, Laf05, Lan05, Luo05, Lyg13a, Lyg14e]. The utilized meshes, which were provided by the NASA Langley Research Center (LARC) (<http://aaac.larc.nasa.gov>), include only the half configuration, while they were re-dimensionalized such as the mean aerodynamic chord to become equal to unity and consequently the grids to be eligible for simulation with the proposed dimensionless methodology. In Figure 7.32 the employed grid for the DLR F-6 WB configuration as well as a close-up view of its solid/symmetry surface is illustrated.

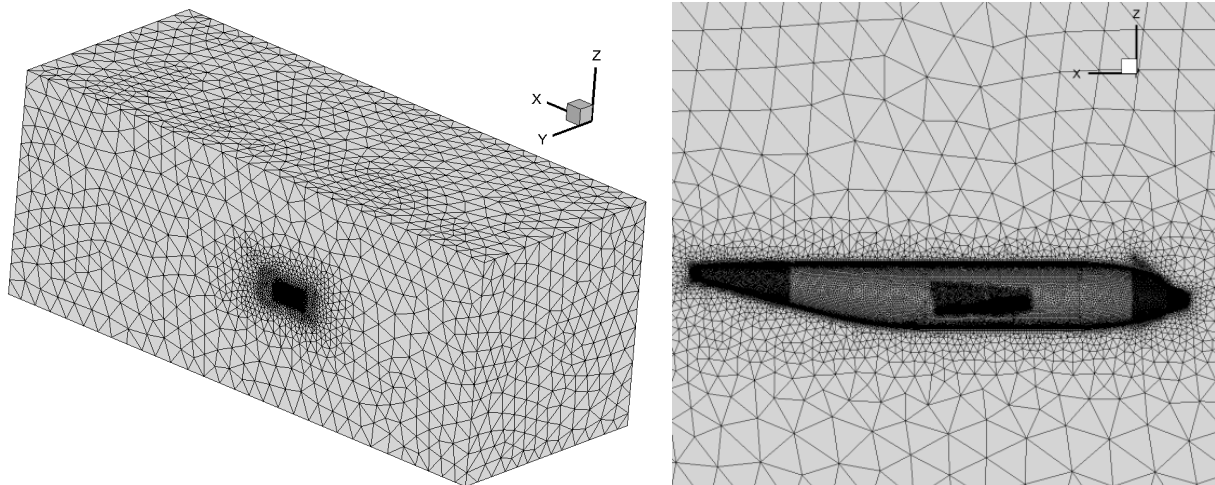


Figure 7.32: Employed grid and its density on the symmetry surface (turbulent flow over the DLR F-6 WB aircraft).

The flow for both aircraft configurations (WB and WBNP) is characterized by free-stream Mach number and Reynolds number (based on the unit mean aerodynamic chord) equal to 0.75 and $3.0E+6$, respectively [Luo05]. For the iterative approximation of the final steady-state solution, the incorporated four-stage Runge-Kutta method was applied with a CFL number equal to unity, while turbulence

prediction was succeeded utilizing the *SST* model; for the latter one implicit treatment of the source term was additionally employed [Bla01]. The gradients at control volumes' interfaces, required for the calculation of viscous fluxes, were obtained with the nodal-averaging scheme, while inviscid fluxes' accuracy was improved with a second-order accurate spatial scheme, coupled with Van Albada-Van Leer limiting function; for multigrid evaluation the Min-mod limiter was used instead, due to its smoothing attitude and consequently its better convergence behaviour.

7.7.1. DLR-F6 wing-body (WB) aircraft configuration

The utilized grid for the WB configuration is composed of 5,666,335 nodes, 4,175,553 tetrahedra, 9,768,149 prisms and 2,587 pyramids, while for parallel processing on a DELL T7500 workstation with two Intel^(R) Xeon^(R)-X5660 six-core processors at 2.80 GHz it was divided in twelve partitions. The angle of attack is set equal to 0.275° , in order the desired lift coefficient equal to 0.5 to be obtained [Luo05]. The parameters of the simulation are summarized in the following Table 7.6.

Table 7.6: Parameters of simulation (turbulent flow over the DLR F-6 WB aircraft).

Parameters	
Type of flow	Turbulent (<i>SST</i>)
Mach number	0.75
Reynolds number	$3.0E+6$
Angle of attack (deg.)	0.275°
Grid density	5,666,335 nodes, 4,175,553 tetrahedra, 9,768,149 prisms and 2,587 pyramids
Number of partitions	12
Iterative scheme	Runge-Kutta method (CFL=1.0)
Computational system	DELL T7500 workstation with two Intel ^(R) Xeon ^(R) -X5660 six-core processors at 2.80 GHz

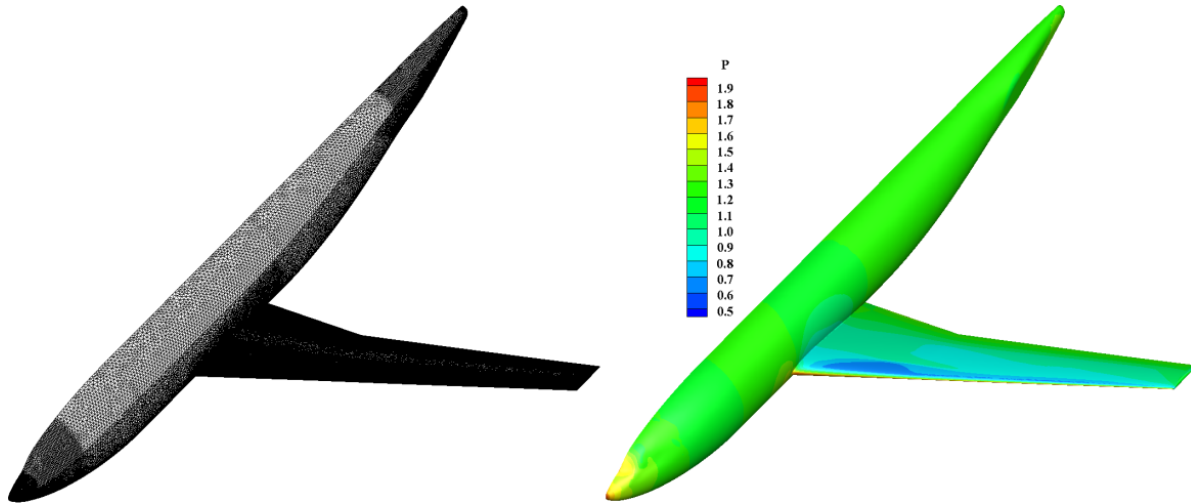


Figure 7.33: Mesh density (left) and predicted dimensionless pressure contours (right) on the surface of the aircraft (turbulent flow over the DLR F-6 WB aircraft).

In Figure 7.33 the mesh density and the predicted contours of the dimensionless pressure on the surface of the aircraft are presented; the latter, compared to the corresponding illustration of Luo et al. [Luo05], demonstrates qualitatively the effectiveness of the proposed methodology. Figure 7.34 includes a close-up view of the predicted dimensionless pressure contours/velocity traces on the upper wing

surface (left) and the obtained isentropic Mach number contours on the lower one (right); the traces reveal a wing-body junction separation on the upper surface area of the wing [Lan05].

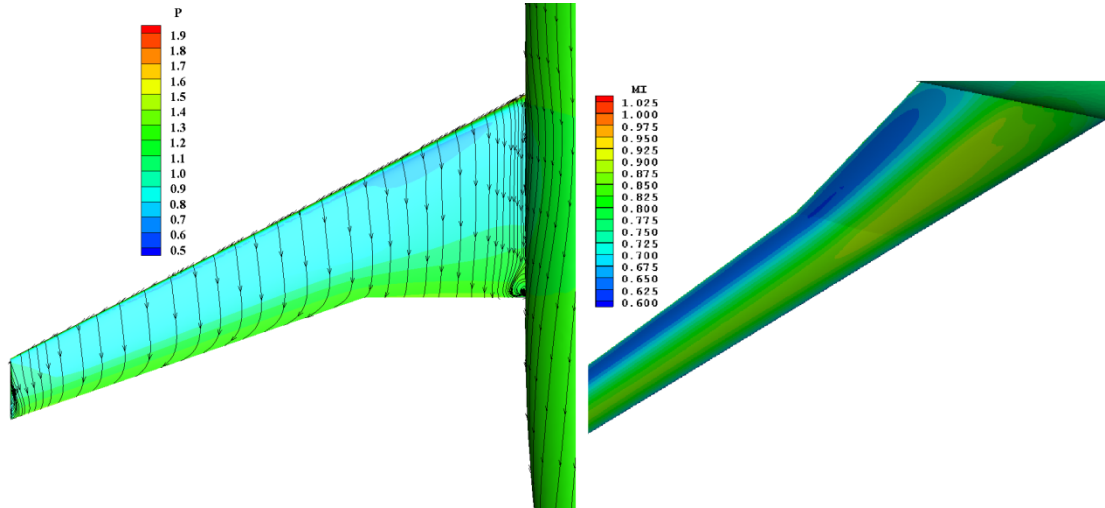


Figure 7.34: Predicted dimensionless pressure contours/velocity traces on the upper wing surface (left) and isentropic Mach number contours (right) on the lower one (turbulent flow over the DLR F-6 WB aircraft).

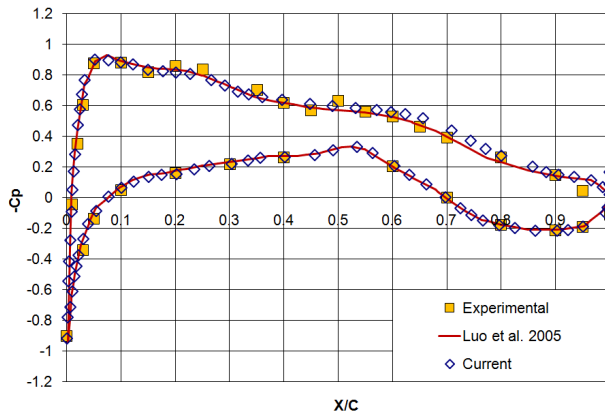


Figure 7.35: Distribution of pressure coefficient at the span-wise section 15% (turbulent flow over the DLR F-6 WB aircraft).

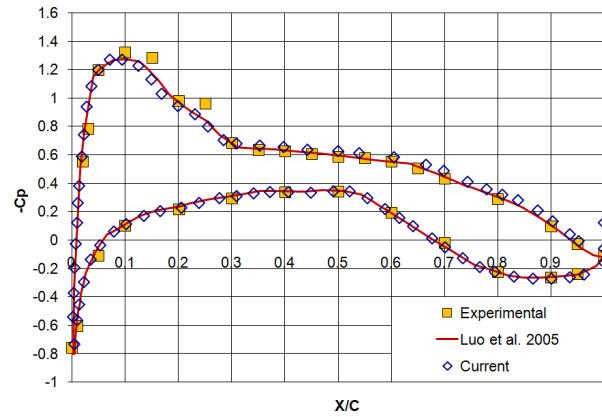


Figure 7.36: Distribution of pressure coefficient at the span-wise section 23.9% (turbulent flow over the DLR F-6 WB aircraft).

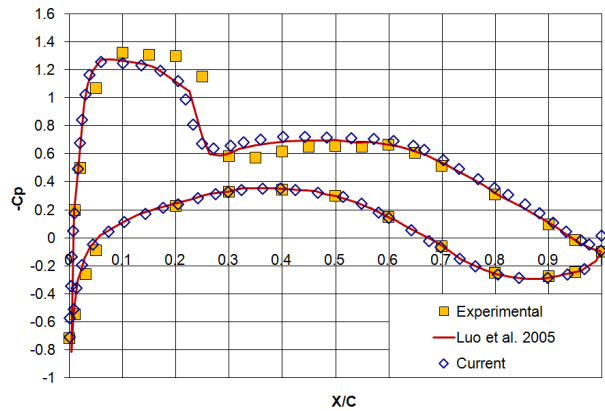


Figure 7.37: Distribution of pressure coefficient at the span-wise section 33.1% (turbulent flow over the DLR F-6 WB aircraft).

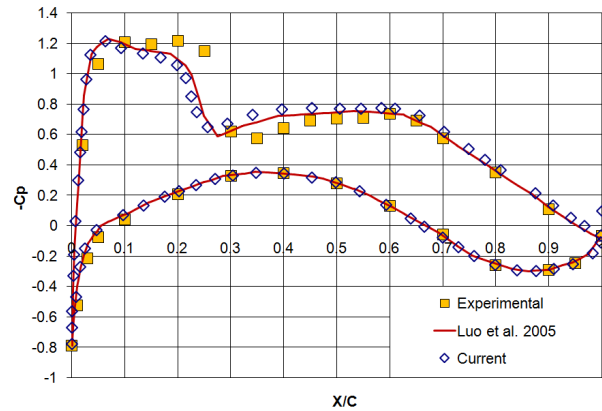


Figure 7.38: Distribution of pressure coefficient at the span-wise section 37.7% (turbulent flow over the DLR F-6 WB aircraft).

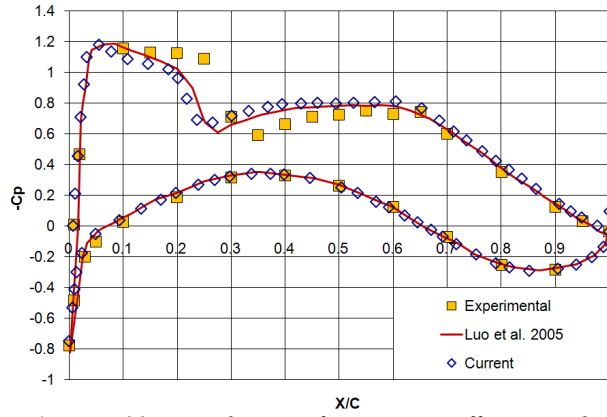


Figure 7.39: Distribution of pressure coefficient at the span-wise section 41.1% (turbulent flow over the DLR F-6 WB aircraft).

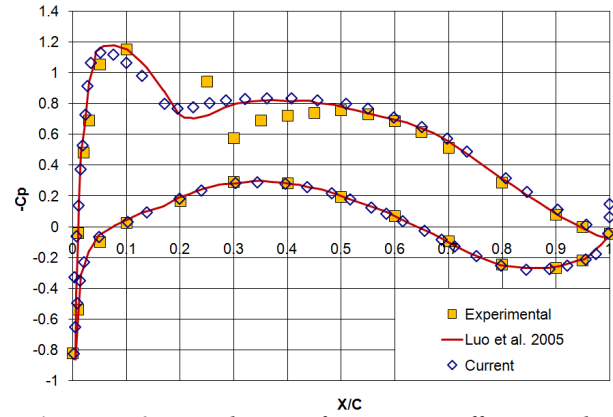


Figure 7.40: Distribution of pressure coefficient at the span-wise section 51.4% (turbulent flow over the DLR F-6 WB aircraft).

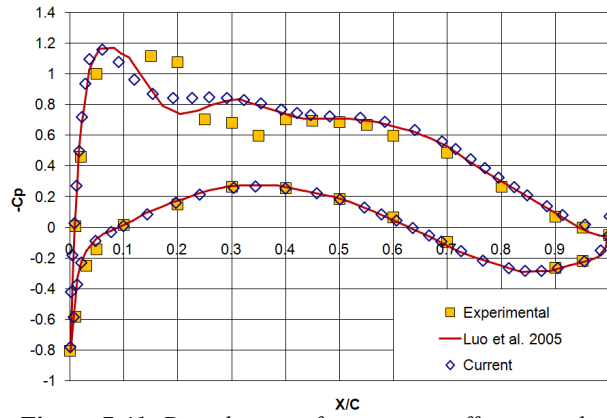


Figure 7.41: Distribution of pressure coefficient at the span-wise section 63.8% (turbulent flow over the DLR F-6 WB aircraft).

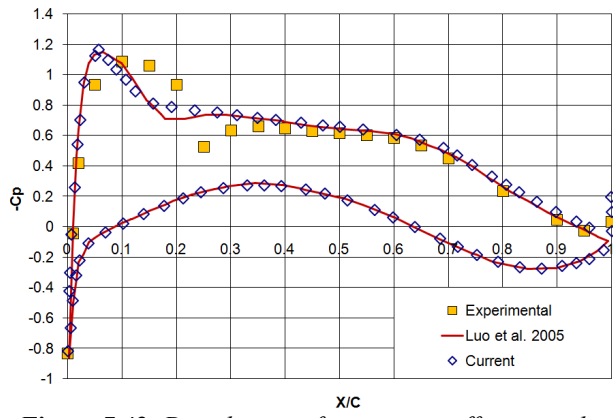


Figure 7.42: Distribution of pressure coefficient at the span-wise section 84.7% (turbulent flow over the DLR F-6 WB aircraft).

In Figures 7.35 to 7.42 the distributions of pressure coefficient C_p at different span-wise sections of aircraft's wing are illustrated, compared with the experimental as well as with the computed by the corresponding simulation of Luo et al. [Luo05]; the obtained results compare well, both with the experimental and the computed reference ones, confirming the equal potential of the developed algorithm for such simulations.

Besides for the qualitative and quantitative validation of accuracy of the final steady-state solution, the incorporated spatial agglomeration multigrid methodology was evaluated against this test case, utilizing initially a relatively coarse grid of the WB configuration, composed of 622,445 nodes, 1,217,387 tetrahedra, 790,934 prisms and 78 pyramids. The single-grid and multigrid (only FAS) runs, using the explicit Runge-Kutta method or the Jacobi algorithm, were performed on a workstation with an AMD FX^(tm)-8350 eight-core processor at 4.0 GHz. For all the encountered simulations the CFL number was set equal to unity (in case of the implicit scheme it was gradually increased up to 2.0), while the initial grid was decomposed in eight sub-domains for parallel processing. Three coarser sub-grids were generated via directional agglomeration for each partition; in Figure 7.43 far views of the initial and the agglomerated control volumes are illustrated, while Figure 7.44 includes the corresponding close-up views on the symmetry surface.

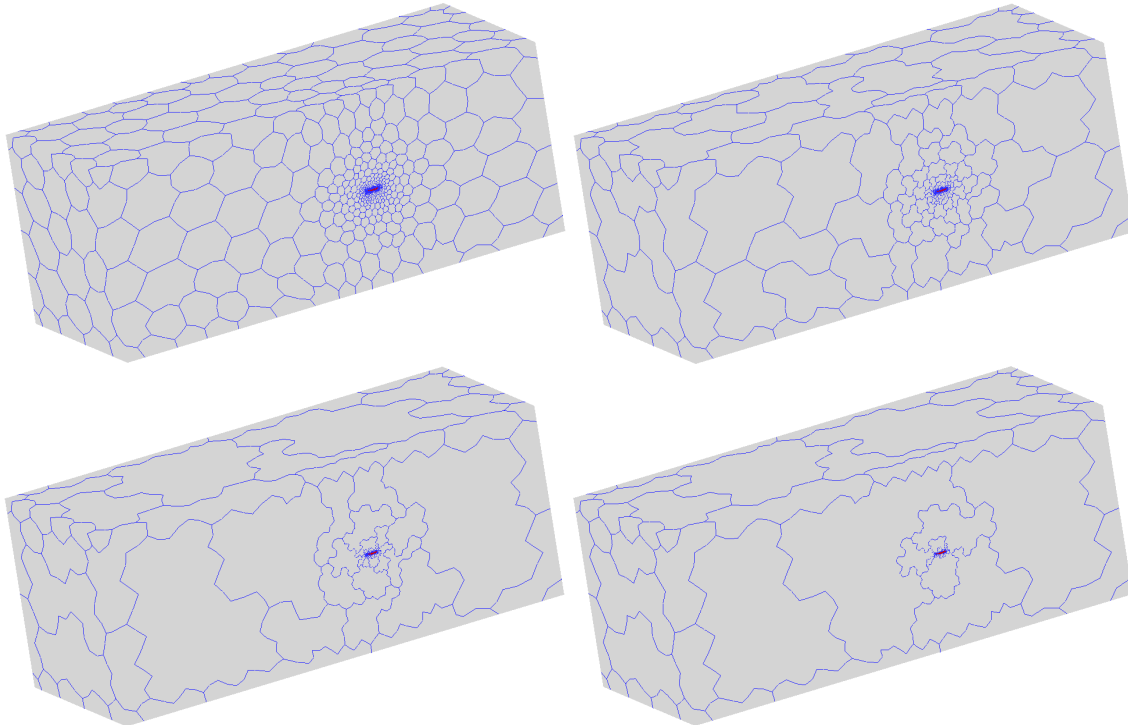


Figure 7.43: Initial and agglomerated control volume grids (turbulent flow over the DLR F-6 WB aircraft).

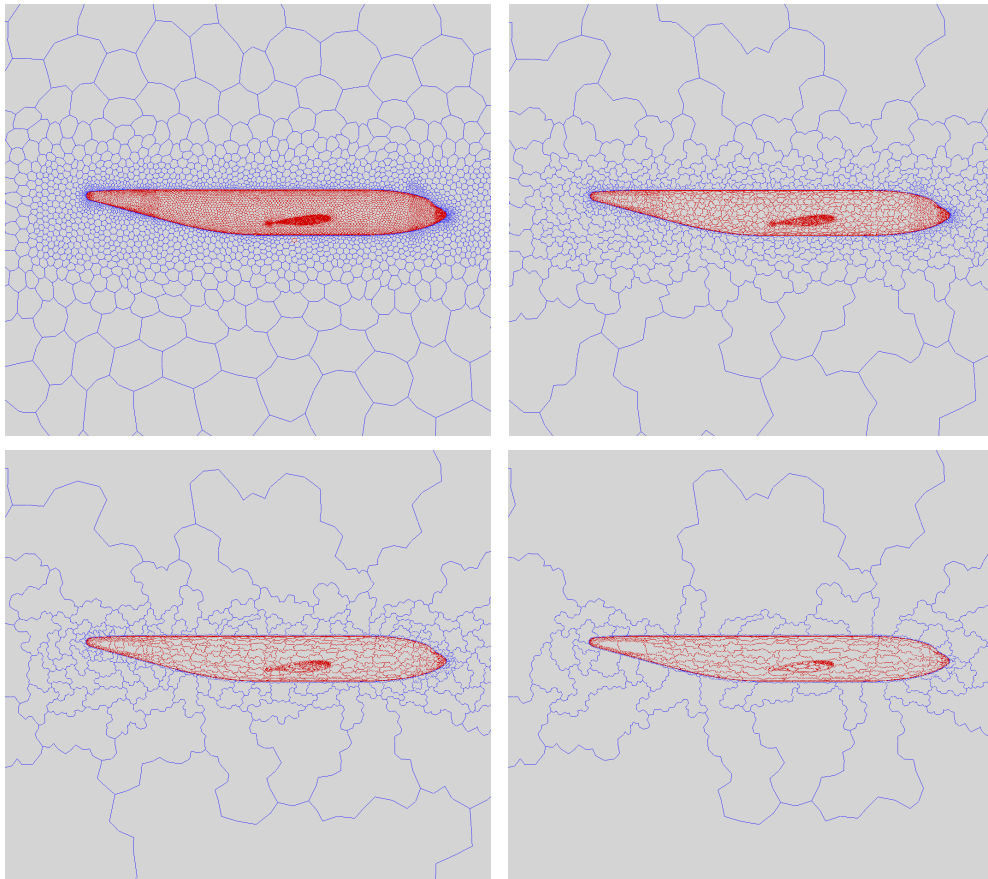


Figure 7.44: Mesh density on the symmetry surface of the initial and agglomerated control volume grids (turbulent flow over the DLR F-6 WB aircraft).

In case of the explicit scheme, iterations/cycles and time speed-up coefficients equal to ~ 7.4 and ~ 3.2 respectively were achieved for a final density residual equal to $1.0E-9$; for turbulent kinetic energy the corresponding coefficients were ~ 7.2 and ~ 3.1 . In Figures 7.45 and 7.46 the single-grid and four-level multigrid simulations' convergence histories per number of iterations/cycles and wall-clock time are presented for density and turbulent kinetic energy respectively. In case of the implicit method, density iterations/cycles and temporal acceleration coefficients equal to ~ 9.3 and ~ 1.9 were obtained, while for turbulent kinetic energy the respective coefficients were ~ 10.0 and ~ 2.0 ; Figures 7.47 and 7.48 include the corresponding convergence histories per number of iterations/cycles and wall-clock time. The implicit runs were relatively slow compared to the explicit ones, due to the small utilized CFL number; efficiency of implicit schemes derives mainly from their capability to handle large CFL numbers. Nevertheless, the aforementioned acceleration coefficients indicate the developed multigrid methodology's potential for significantly improved computational performance even for relatively coarse grids.

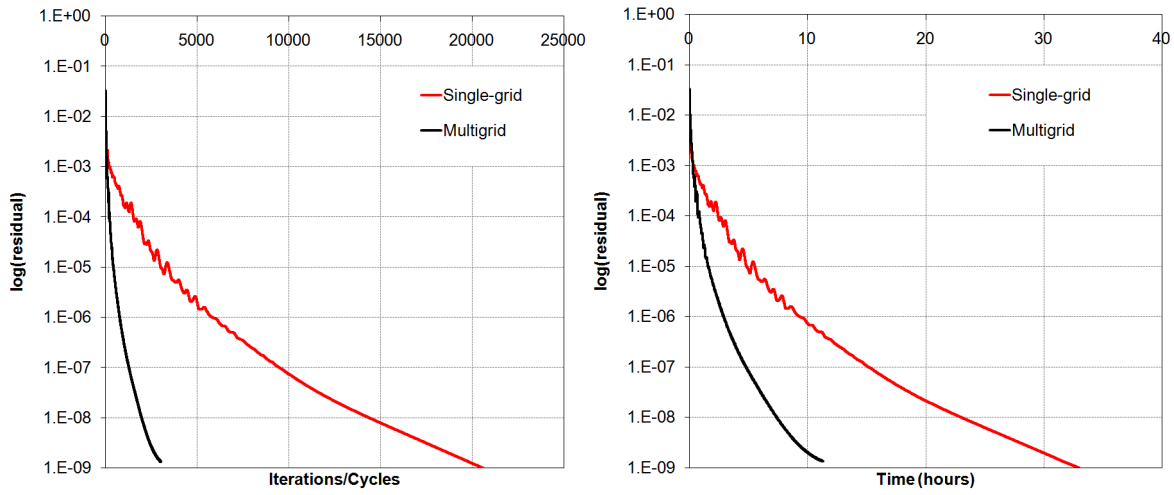


Figure 7.45: Density convergence history per iterations/cycles and time (explicit scheme, turbulent flow over the DLR F-6 WB aircraft).

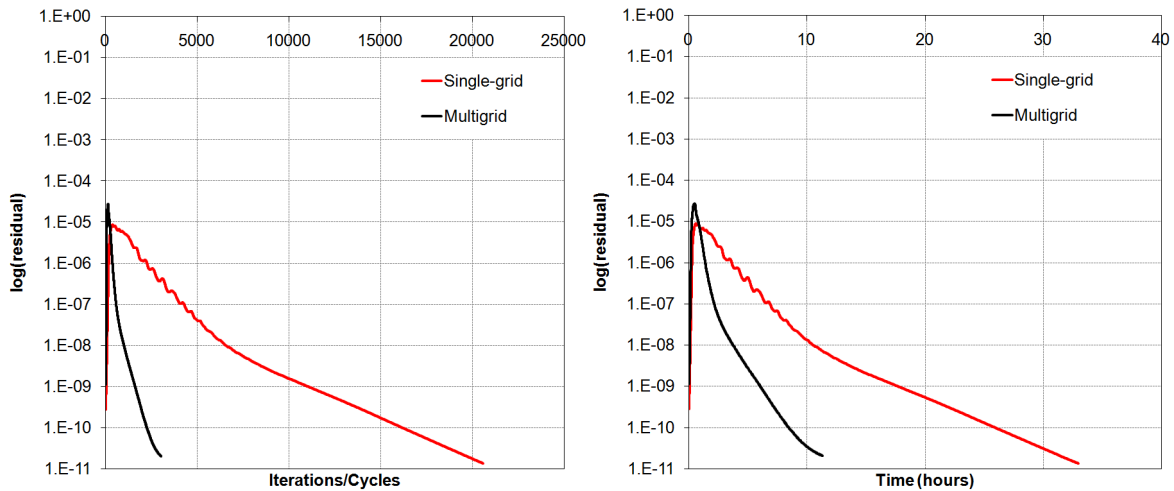


Figure 7.46: Turbulent kinetic energy convergence history per iterations/cycles and time (explicit scheme, turbulent flow over the DLR F-6 WB aircraft).

Besides the aforementioned coarse mesh, the fine one (5,666,335 nodes, 4,175,553 tetrahedra, 9,768,149 prisms and 2,587 pyramids), employed for the validation of the proposed methodology's

accuracy, was utilized to study the influence of the grid size on the gained speed-up due to the multigrid algorithm. The single-grid and multigrid simulations, using the explicit Runge-Kutta method ($CFL=1.0$), were performed on a DELL T7500 workstation with two Intel^(R) Xeon^(R)-X5550 four-core processors at 2.67 GHz, while for parallel processing the initial grid was divided in eight sub-domains. Three coarser sub-grids were generated via full-coarsening and semi-coarsening directional agglomeration for each partition; in Figure 7.49 far views of the first and second full-coarsening agglomerated control volumes grids are illustrated, while Figures 7.50 and 7.51 include the corresponding close-up views on the symmetry surface and near aircraft nose. Table 7.7 includes the number of DoF's (nodes) per sub-domain of the initial and coarser grids, for both semi- and full-coarsening cases; the *ghost* nodes at overlapping regions are also included in this Table, resulting in an increased summed number of DoF's at first level compared to the total one. Moreover, in order to confirm the additional acceleration, gained by the combined FMG-FAS procedure compared to the only FAS one, simulations with both of them were encountered. In Figures 7.52 and 7.53 the single-grid and four multigrid simulations' convergence histories per number of iterations/cycles and time are presented for density and turbulent kinetic energy respectively.

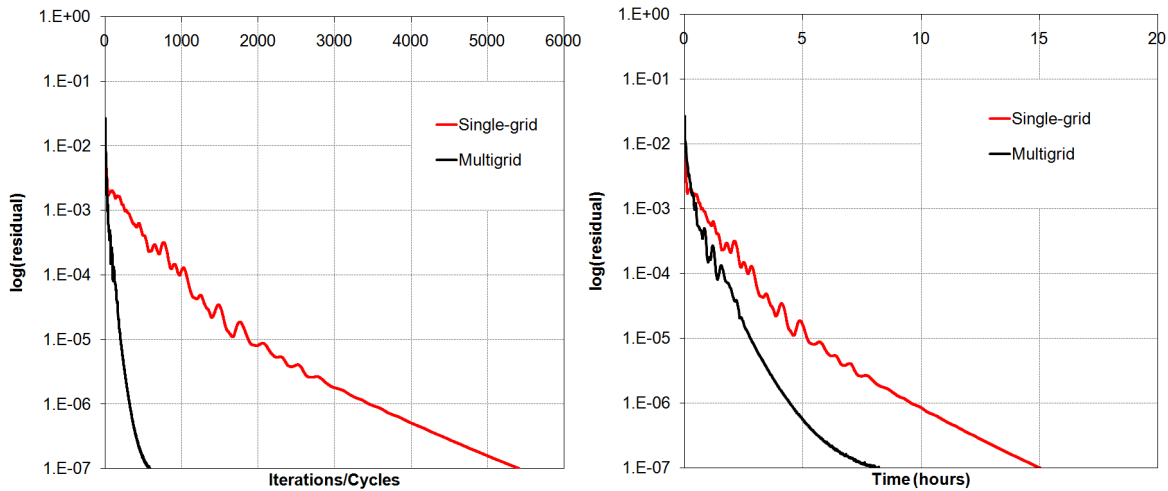


Figure 7.47: Density convergence history per iterations/cycles and time (implicit scheme, turbulent flow over the DLR F-6 WB aircraft).

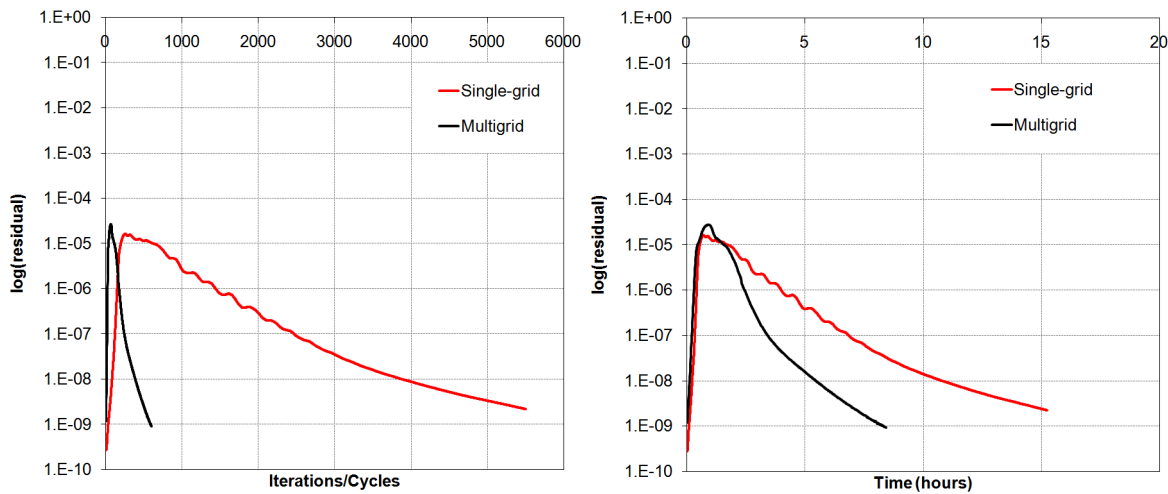


Figure 7.48: Turbulent kinetic energy convergence history per iterations/cycles and time (implicit scheme, turbulent flow over the DLR F-6 WB aircraft).

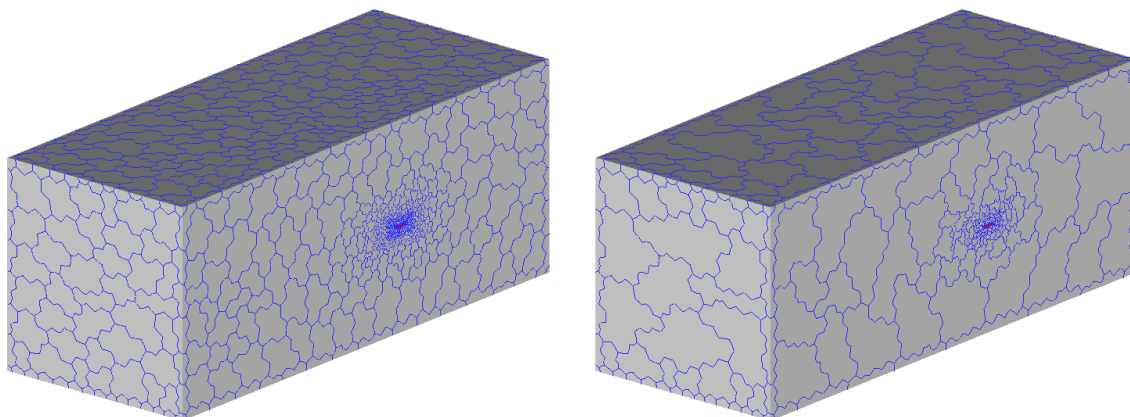


Figure 7.49: First and second full-coarsening agglomerated control volume grids (fine mesh, turbulent flow over the DLR F-6 WB aircraft).

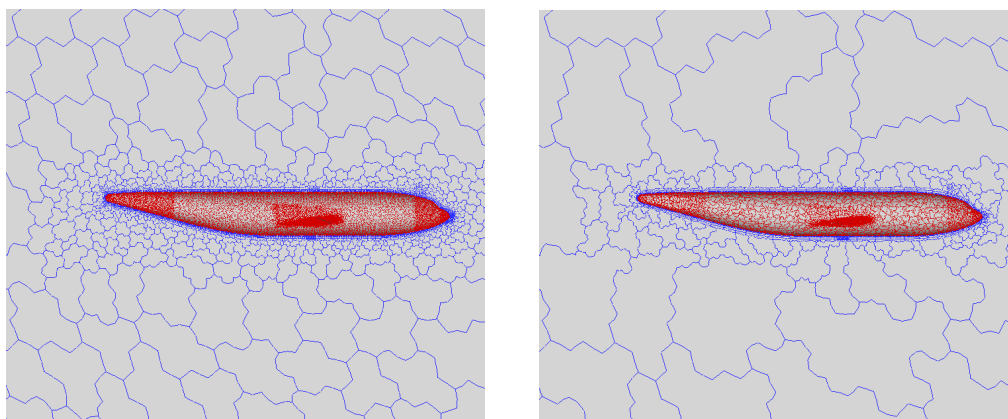


Figure 7.50: Mesh density on the symmetry surface of the first and second full-coarsening agglomerated control volume grids (fine mesh, turbulent flow over the DLR F-6 WB aircraft).

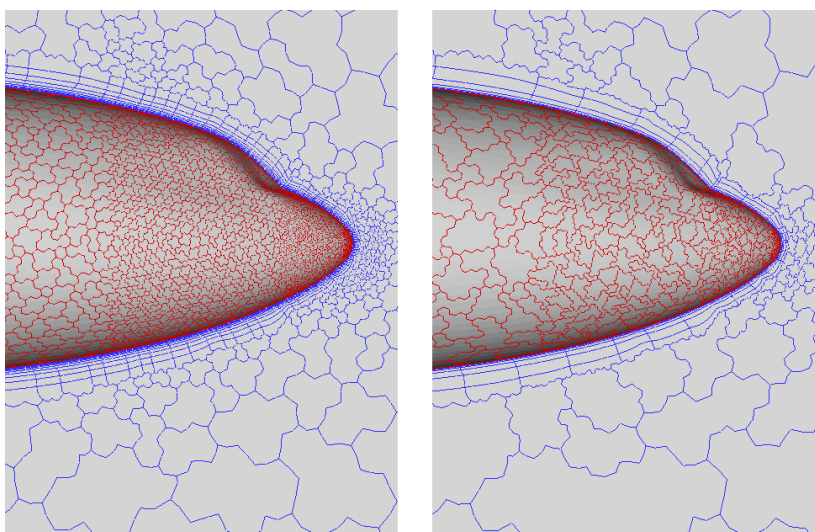


Figure 7.51: Boundary density near aircraft nose of the first and second full-coarsening agglomerated control volume grids (fine mesh, turbulent flow over the DLR F-6 WB aircraft).

An acceleration increase is evident, compared to the corresponding results on the coarser initial mesh; the greater the number of computational nodes, the greater is the acceleration obtained [Lyg14g].

The maximum density speed-up coefficient equal to ~ 5.3 was obtained by the full-coarsening agglomeration FMG-FAS. As expected, full-coarsening agglomeration multigrid scheme was revealed much more efficient than the semi-coarsening one, due to the more extended DoF's decrease it entails, which subsequently results to an elevated acceleration for the same computational system. Unlike this merging strategy the semi-coarsening one seems to be useless against this test case, as it accelerated negligibly the corresponding run comparing to the single-grid one. Finally, the combined FMG-FAS runs derived better temporal speed-up coefficients, comparing to those performed with only FAS, although the latter ones succeeded greater ratios per number of iterations/cycles; in case of full-coarsening fusion FMG-FAS succeeded an acceleration ratio of ~ 9.5 and ~ 5.3 per iterations and time respectively, while FAS ~ 10.6 and ~ 4.6 respectively. However, the convergence history curves of the FMG-FAS runs (especially for the full-coarsening fusion strategy) seem to diverge after several orders decrease of the residuals; therefore, the corresponding only FAS runs appear to be more time-effective this point forward. This FMG-FAS methodology deficiency derives from the insufficient and irreversible approximation derived at the preliminary stage. Despite the FMG-FAS is assumed in general more efficient than only FAS, it is susceptible to be confined to local maxima or minima and therefore generate oscillations or even fail.

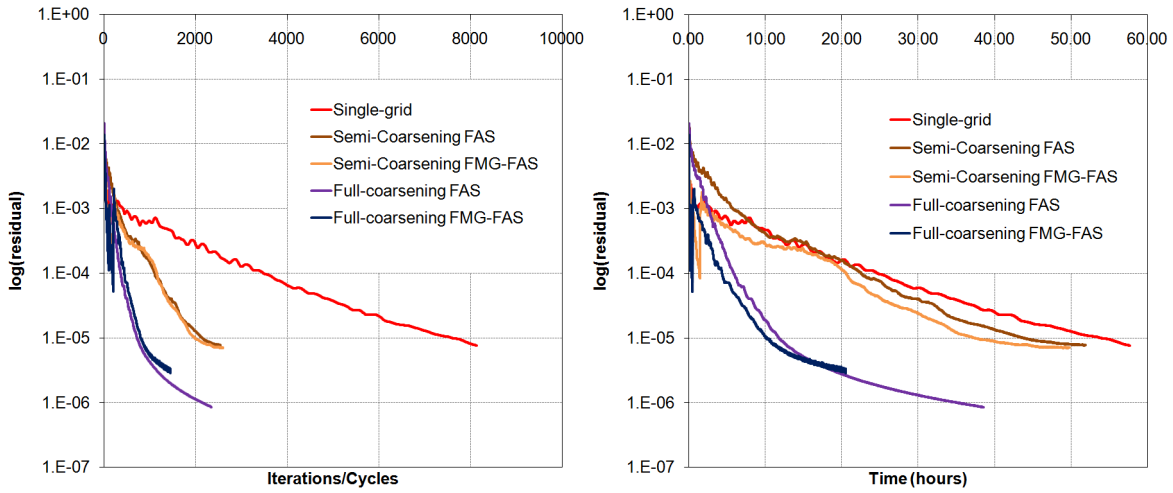


Figure 7.52: Density convergence history per iterations/cycles and time (fine grid, turbulent flow over the DLR F-6 WB aircraft).

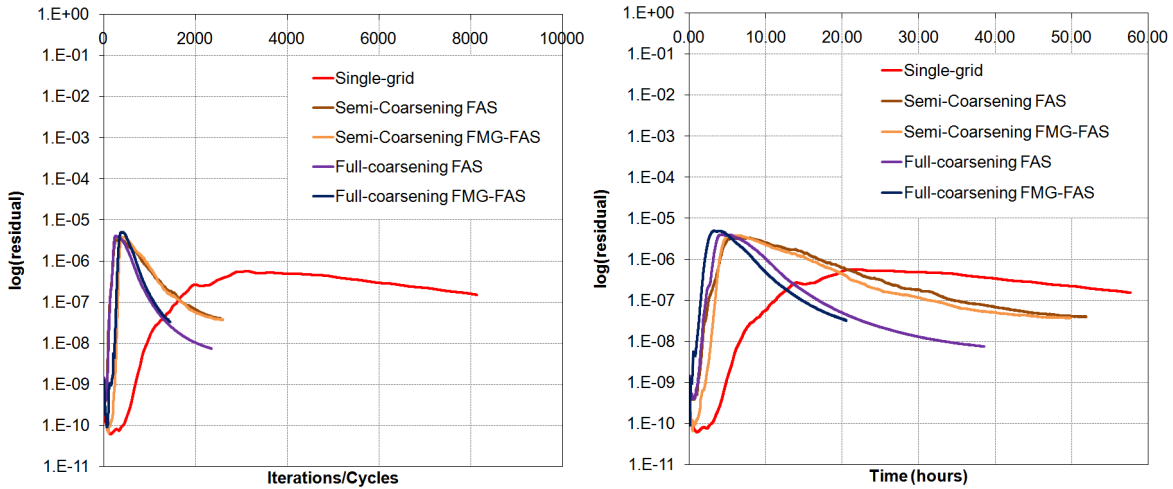


Figure 7.53: Turbulent kinetic energy convergence history per iterations/cycles and time (fine grid, turbulent flow over the DLR F-6 WB aircraft).

Table 7.7: Grid coarsening by semi- and full-coarsening directional agglomeration.

	Sub-domains	Level 1 (DoF's)	Level 2 (DoF's)	Level 3 (DoF's)	Level 4 (DoF's)
Semi-coarsening	1st	728,628	354,399	202,529	155,316
	2nd	725,734	359,573	211,640	167,769
	3rd	717,934	323,993	175,954	125,165
	4th	722,500	350,838	198,323	147,571
	5th	732,802	359,146	213,165	158,405
	6th	722,019	351,405	198,571	151,802
	7th	727,505	363,072	211,652	157,629
	8th	720,742	357,037	206,502	145,302
Full-coarsening	1st	728,628	135,299	35,738	16,737
	2nd	725,734	137,450	37,101	16,994
	3rd	717,934	135,718	35,414	17,645
	4th	722,500	138,859	39,259	19,571
	5th	732,802	143,627	40,518	21,072
	6th	722,019	129,100	32,056	15,993
	7th	727,505	119,366	24,911	9,859
	8th	720,742	118,164	24,928	10,164

7.7.2. DLR-F6 wing-body-nacelles-pylons (WBNP) aircraft configuration

The sub-case considering the WBNP aircraft configuration was examined, utilizing a mesh consisting of 5,887,240 nodes, 3,037,873 tetrahedra, 10,594,701 prisms and 9,114 pyramids. For parallel computation on a DELL T7500 workstation with two Intel^(R) Xeon^(R)-X5650 six-core processors at 2.67 GHz it was divided in twelve sub-domains. The angle of attack is set equal to 0.8° , in order the desired lift coefficient equal to 0.5 to be obtained [Luo05]. The parameters of the simulation are summarized in the following Table 7.8.

Table 7.8: Parameters of simulation (turbulent flow over the DLR F-6 WBNP aircraft).

Parameters	
Type of flow	Turbulent (SST)
Mach number	0.75
Reynolds number	$3.0E+6$
Angle of attack (deg.)	0.8°
Grid density	5,887,240 nodes, 3,037,873 tetrahedra, 10,594,701 prisms and 9,114 pyramids
Number of partitions	12
Iterative scheme	Runge-Kutta method (CFL=1.0)
Computational system	DELL T7500 workstation with two Intel ^(R) Xeon ^(R) -X5650 six-core processors at 2.67 GHz

In Figure 7.54 the mesh density and the obtained dimensionless pressure distribution on the surface of the WBNP aircraft are presented; the latter illustration indicates a qualitative agreement with the corresponding computed one of Luo et al. [Luo05]. Figure 7.55 includes close-up views of the predicted dimensionless pressure contours/velocity traces on the upper (top) and lower (bottom) wing surfaces; compared to the experimental oil flow visualizations contained in the study of Langtry et al. [Lan05] sufficient agreement can be observed. On the upper wing surface area a wing-body junction separation is noticed at the wing root for both the experimental and simulated results [Lan05]. Similarly on the lower

one, the presence of the nacelle-pylon produces supersonic flow at the internal area of the wing, which results after the shock in flow separation at the wing-pylon junction [Lan05].

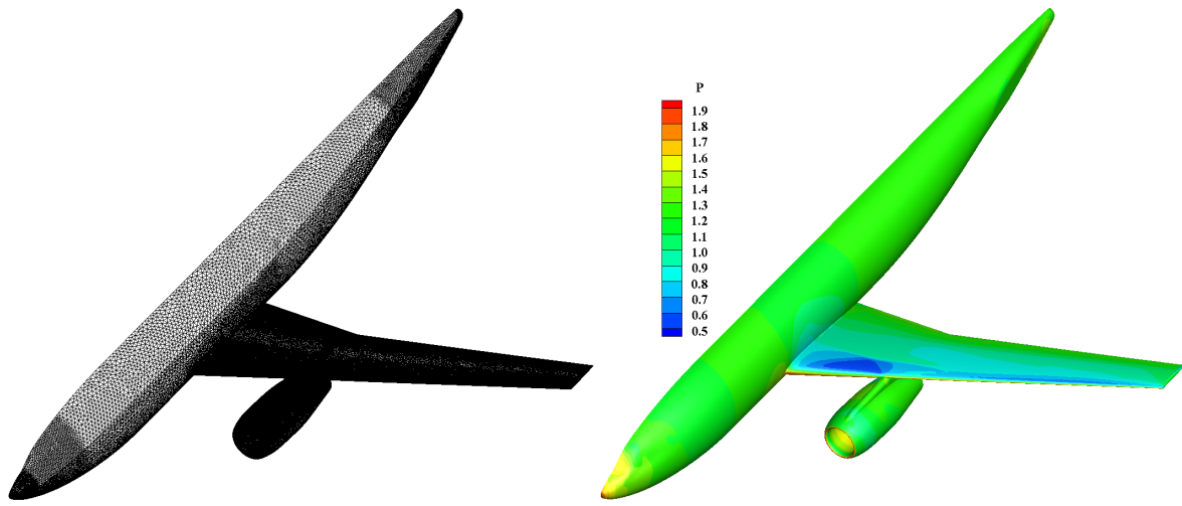


Figure 7.54: Mesh density (left) and predicted dimensionless pressure contours (right) on the surface of the aircraft (turbulent flow over the DLR F-6 WBNP aircraft).

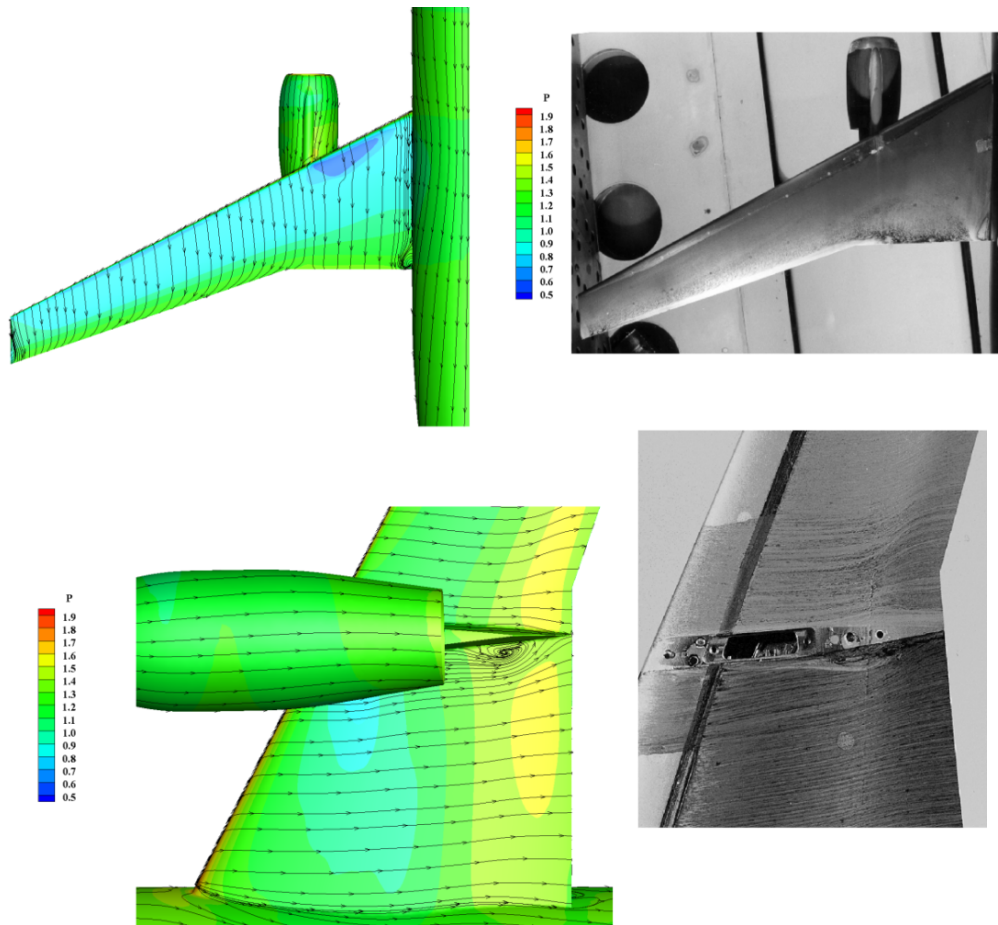


Figure 7.55: Predicted dimensionless pressure contours/velocity traces on the upper (top) and lower (bottom) wing surface of the aircraft, compared to the experimental oil flow visualizations contained in the study of Langtry et al. [Lan05] (turbulent flow over the DLR F-6 WBNP aircraft).

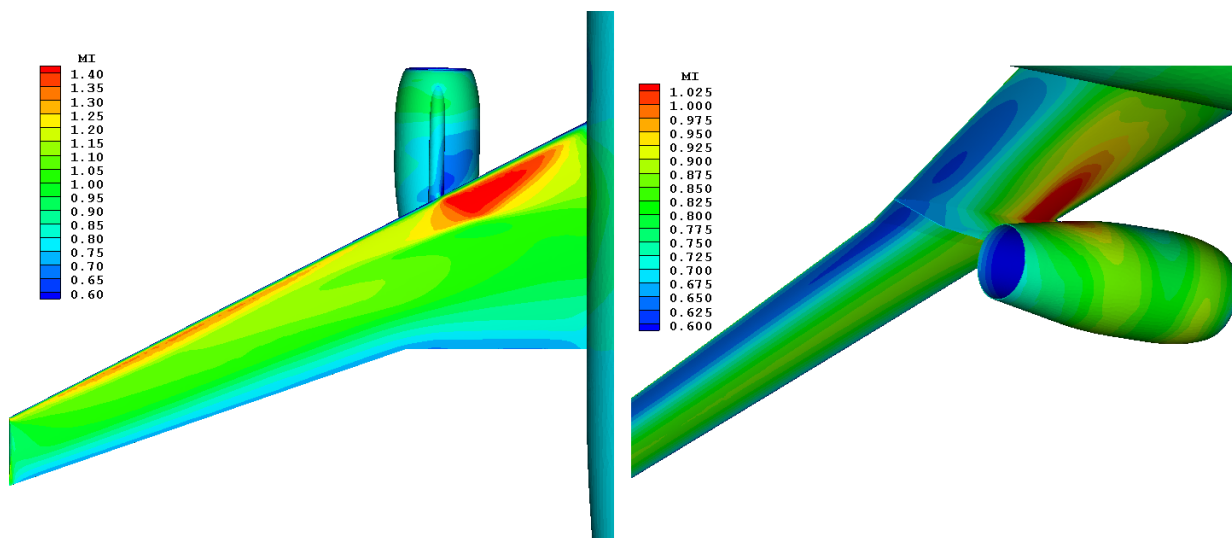


Figure 7.56: Predicted isentropic Mach number contours on the upper (left) and lower (right) wing surface of the aircraft (turbulent flow over the DLR F-6 WBNP aircraft).

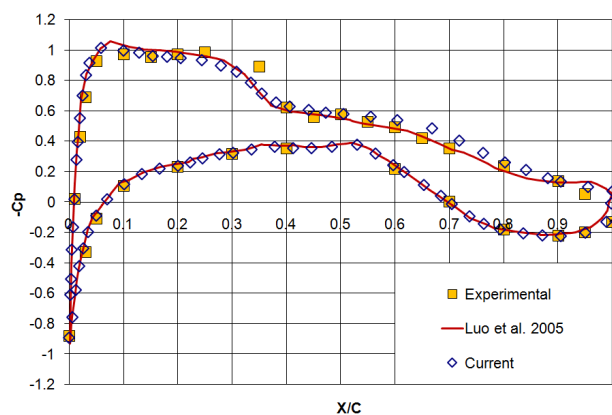


Figure 7.57: Distribution of pressure coefficient at the span-wise section 15% (turbulent flow over the DLR F-6 WBNP aircraft).

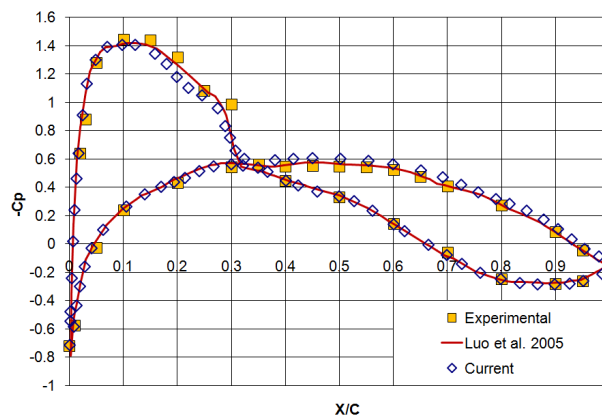


Figure 7.58: Distribution of pressure coefficient at the span-wise section 23.9% (turbulent flow over the DLR F-6 WBNP aircraft).

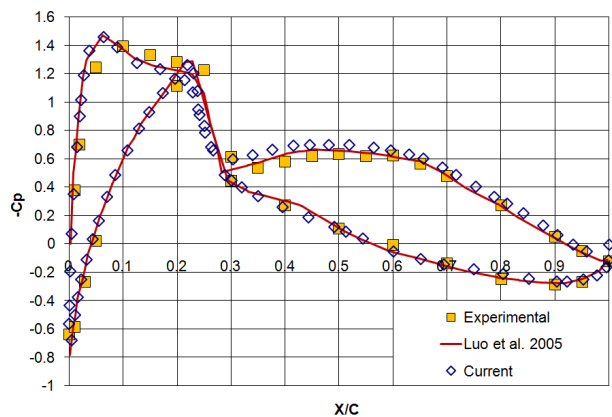


Figure 7.59: Distribution of pressure coefficient at the span-wise section 33.1% (turbulent flow over the DLR F-6 WBNP aircraft).

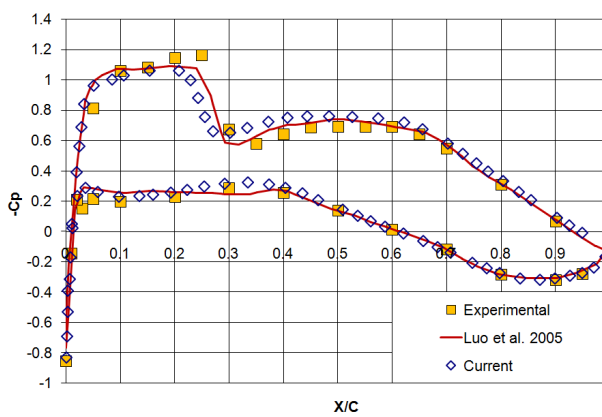


Figure 7.60: Distribution of pressure coefficient at the span-wise section 37.7% (turbulent flow over the DLR F-6 WBNP aircraft).

In Figure 7.56 the predicted isentropic Mach number contours are presented for the same wing surfaces (upper and lower), confirming the aforementioned flow separations. Figures 7.57 to 7.64 contain the distributions of pressure coefficient C_p at different span-wise sections of aircraft's wing, compared with the experimental ones and those computed by Luo et al. [Luo05]; the obtained results achieve a satisfactory agreement both with the experimental and the computed reference ones, revealing the potential of this algorithm for such predictions.

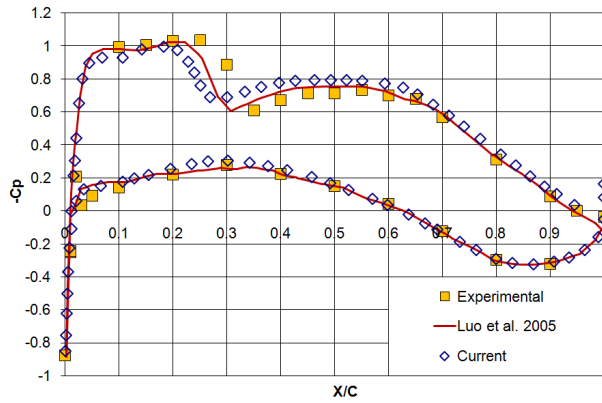


Figure 7.61: Distribution of pressure coefficient at the span-wise section 41.1% (turbulent flow over the DLR F-6 WBNP aircraft).

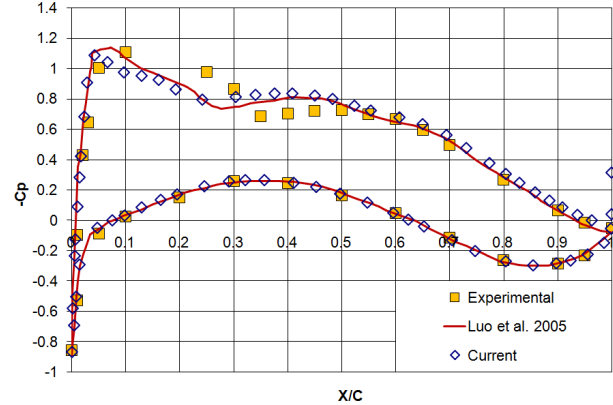


Figure 7.62: Distribution of pressure coefficient at the span-wise section 51.4% (turbulent flow over the DLR F-6 WBNP aircraft).

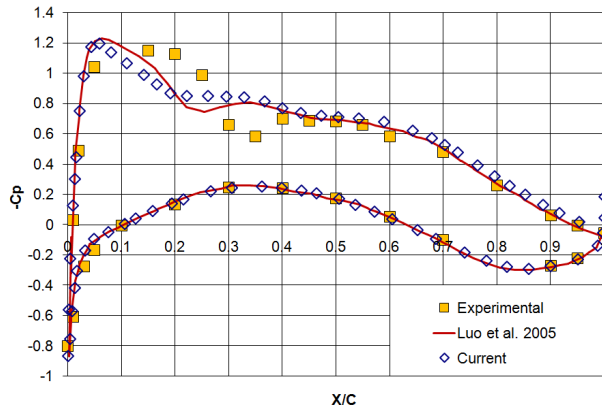


Figure 7.63: Distribution of pressure coefficient at the span-wise section 63.8% (turbulent flow over the DLR F-6 WBNP aircraft).

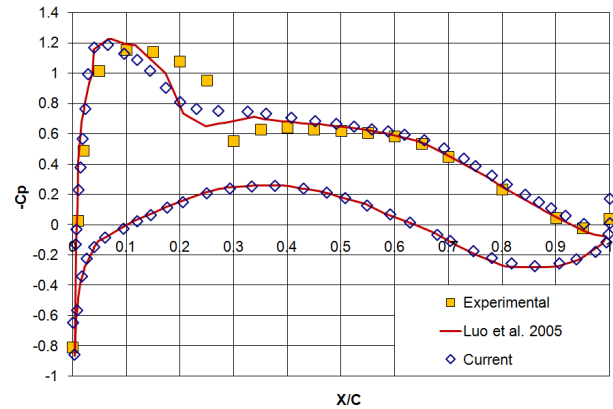


Figure 7.64: Distribution of pressure coefficient at the span-wise section 84.7% (turbulent flow over the DLR F-6 WBNP aircraft).

7.8. Turbulent flow over the common research model (CRM)

The last fluid flow test case encountered in this study, concerns the fully turbulent flow over the NASA CRM aircraft with (Wing-Body-Horizontal Tail, WBHT) and without horizontal tail (Wing-Body, WB) [Lee10, Mav10, Cez13, Mur13, Par13, Sca13, Scl13, Vos10, Vos13, Lyg14a, Lyg15, Sar15]; the aforementioned problem was examined by the participants of the fourth and fifth AIAA Drag Prediction Workshop (DPW), held in San Antonio in 2009 and in New Orleans in 2012, respectively. The CRM, which is a 2.7% scaled representative of a modern commercial transonic flow aircraft, was designed specifically for the evaluation of CFD codes [Scl13]. Therefore, the geometry parameters as well as the experimental (wind tunnel) results are available; furthermore, most of the grids used by the participants of the pre-mentioned DPW's can be obtained from the NASA Langley Research Center (LARC) (<http://aaac.larc.nasa.gov/tsab/cfdlarc/aiaadpw>). The provided grids, utilized in this study, include only the half aircraft, while they were re-dimensionalized, in order the mean aerodynamic chord to become

equal to unity and, consequently, to be used with the developed dimensionless methodology; in Figure 7.65 the utilized grid for the CRM WB configuration as well as a close-up view of its solid/symmetry surface are illustrated.

For both aircrafts (WB and WBHT), the free-stream flow is characterized by Mach number 0.85 , Reynolds number (based on the unit mean aerodynamic chord) $5.0E+6$ and reference temperature $310,927\text{ K}$ (100° F). The final steady-state solution was approximated by employing the incorporated second-order time accurate four-stage Runge-Kutta method with a CFL number equal to unity, while turbulence prediction was succeeded with the *SST* model, along with an implicit treatment of its source term; no transition was assumed. The simulations were accelerated with the developed directional agglomeration multigrid scheme (full-coarsening) and parallelization strategy, while no further optimization for computational performance has been carried out.

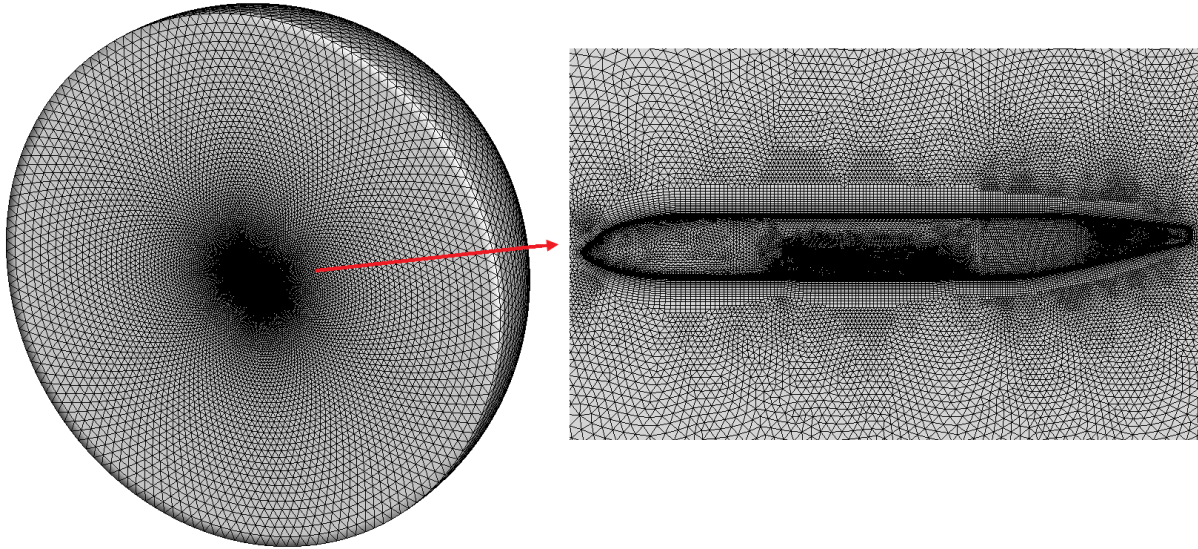


Figure 7.65: Employed grid and its density on symmetry surface (turbulent flow over the CRM WB aircraft).

7.8.1. CRM wing-body (WB) aircraft configuration

A relatively coarse grid, consisted of $3,342,884$ nodes, $5,042,388$ tetrahedra, $4,425,652$ prisms and $33,072$ pyramids, was utilized for the representation of the computational domain around the WB aircraft. The angle of attack of the free-stream flow was set equal to 2.15° , considering the desired value equal to 0.5 of the lift coefficient. The gradients, required for viscous fluxes, were obtained with the nodal-averaging method, while the second-order spatial accurate scheme, used for the inviscid ones, was coupled with Van Albada-Van Leer slope limiter. Implementing a three-level multigrid scheme along with a twelve-partitions parallel processing on a DELL T7500 workstation with two Intel^(R) Xeon^(R)-X5660 six-core processors at 2.80 GHz , the density residual was decreased approximately three orders of magnitude in 24 hours. The parameters of the simulation are summarized in the following Table 7.9. Nevertheless, the employed mesh revealed to be insufficient for the accurate prediction of the expected flow phenomena, especially in the region of shock waves and for wing span-wise sections approximately greater than 28.28% . The developed h-refinement methodology was implemented to this area, while a second refinement was used for the outer half wing compartment due to its significantly coarse initial density, deriving the finally used grid, which consists of $4,570,180$ nodes, $6,721,680$ tetrahedra, $6,651,436$ prisms and $34,318$ pyramids; Figure 7.66 depicts the mesh density on the wing surface prior (left) and after (right) h-refinement.

Since the results for the initial mesh were interpolated on the refined grid, the simulation was continued deriving the final steady-state solution. In Figure 7.67 the final mesh density and predicted dimensionless pressure contours on the aircraft surface are illustrated. Figures 7.68 and 7.69 include the corresponding Mach number contours on the symmetry surface and a section at 37% of the wing span,

allowing for qualitative comparison of the derived results with those of other solvers [Cez13, Mur13, Par13]; the equal potential of the proposed solver for such simulations is indicated.

Table 7.9: Parameters of simulation (turbulent flow over the CRM WB aircraft).

Parameters	
Type of flow	Turbulent (SST)
Mach number	0.85
Reynolds number	$5.0E+6$
Angle of attack (deg.)	2.15°
Grid density	3,342,884 nodes, 5,042,388 tetrahedra, 4,425,652 prisms and 33,072 pyramids
Number of partitions	12
Number of multigrid levels	3
Iterative scheme	Runge-Kutta method (CFL=1.0)
Computational system	DELL T7500 workstation with two Intel ^(R) Xeon ^(R) -X5660 six-core processors at 2.80 GHz

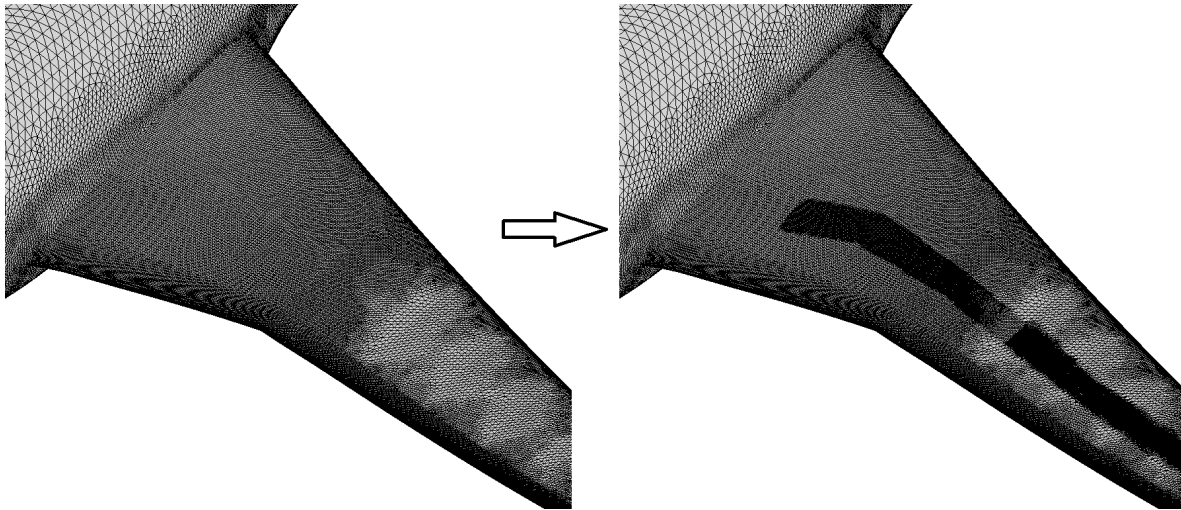


Figure 7.66: Mesh density on the wing surface prior (left) and after (right) *h*-refinement (turbulent flow over the CRM WB aircraft).

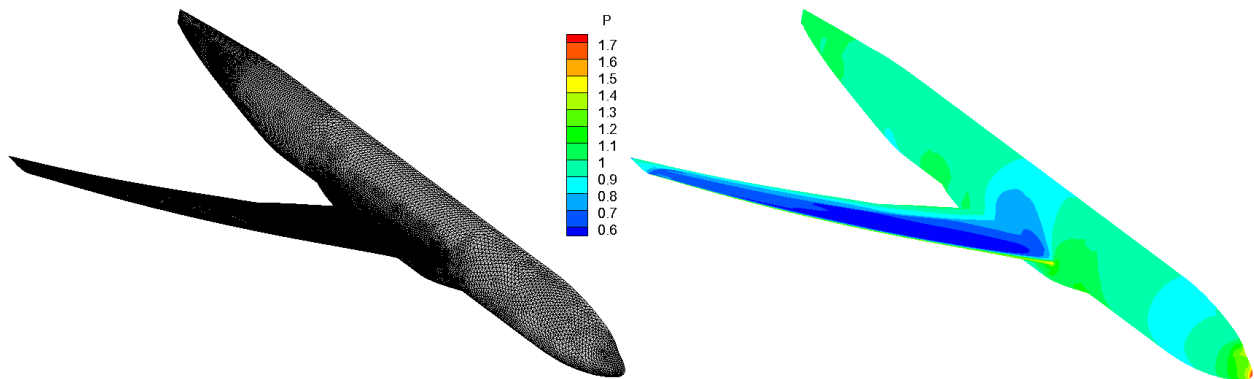


Figure 7.67: Mesh density (left) and predicted dimensionless pressure contours (right) on the surface of the aircraft (turbulent flow over the CRM WB aircraft).



Figure 7.68: Predicted Mach number contours on the symmetry surface of the aircraft (turbulent flow over the CRM WB aircraft).

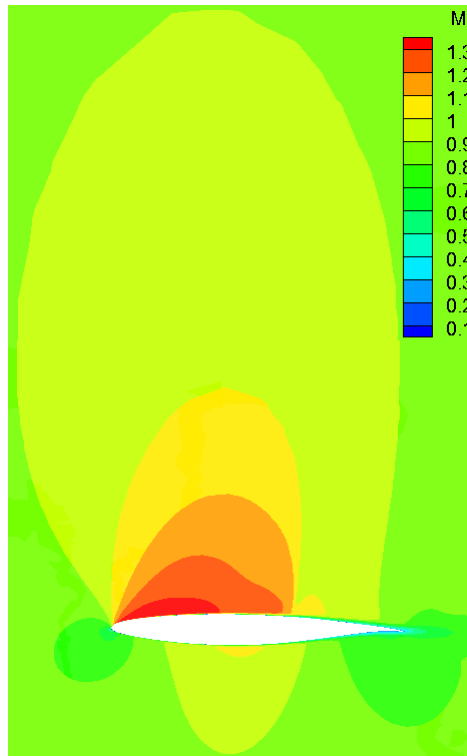


Figure 7.69: Predicted Mach number contours on the wing span-wise section 37% of the aircraft (turbulent flow over the CRM WB aircraft).

No considerable wing-body juncture separation at the upper side of the wing was observed, similarly to some of the participants of the DPW's [Cez13, Par13, Sca13, Scl13]; its absence might derive from the utilization of a relatively coarse mesh along with the *SST* turbulence model [Scl13]. Figures 7.70 up to 7.75 contain the pressure coefficient distributions at different wing span-wise sections of the aircraft, compared to the corresponding experimental and reference calculated ones [Sca13]. A satisfactory agreement is identified between the obtained results and the reference experimental and computed data [Sca13], considering the relatively small number of DoF's of the finally utilized mesh as well as the differences, especially in the area of shock formations, between the results of the DPW's' participants [Lev13]. Additionally, the significant contribution of the h-refinement methodology is demonstrated with the better shock capturing at wing span-wise sections greater equal than 50.24% of the adapted mesh, comparing to the initial one. Local grid adaptation had though a negligible impact to the lift and drag coefficients; thus, angle of attack of free-stream flow wasn't adjusted. In Table 7.10 the aforementioned coefficients (lift and idealized drag ones), derived by near-field analysis [Vos10, Vos13] from the initial

and refined mesh, are presented in comparison with the experimental ones [Vas10] as well as with the calculated ones by Vos and Sanchi [Vos10]; the idealized drag coefficient is defined as follows [Lee10]

$$C_{Didealized} = C_D - \frac{C_L^2}{\pi \cdot AR} \quad (7.6)$$

where AR is the wing aspect ratio [Lee10]. The obtained lift coefficients are very close with both the reference computed and experimental one [Lee10, Vas10, Vos10], while grid adaptation entails a slight increase. As far as the idealized drag results are concerned, a satisfactory agreement is achieved with the reference computed one [Vos10]. Nevertheless, an overestimation is obvious when it is compared to the experimental one, deriving from the relatively coarse utilized grid [Sca13]; drag is reduced when going from a coarser to a finer resolution [Vos10]. The previous state is confirmed with the corresponding comparison between the initial and refined grid, as well as with the utilization of a medium grid in the next section, concerning the WBHT configuration.

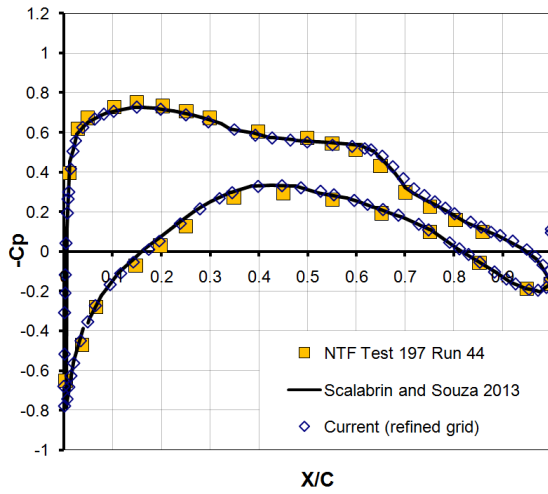


Figure 7.70: Pressure coefficient distribution at wing span-wise section 13.06% (turbulent flow over the CRM WB aircraft).

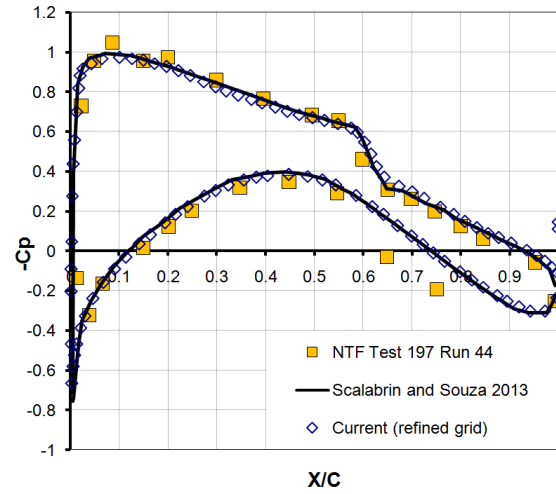


Figure 7.71: Pressure coefficient distribution at wing span-wise section 28.28% (turbulent flow over the CRM WB aircraft).

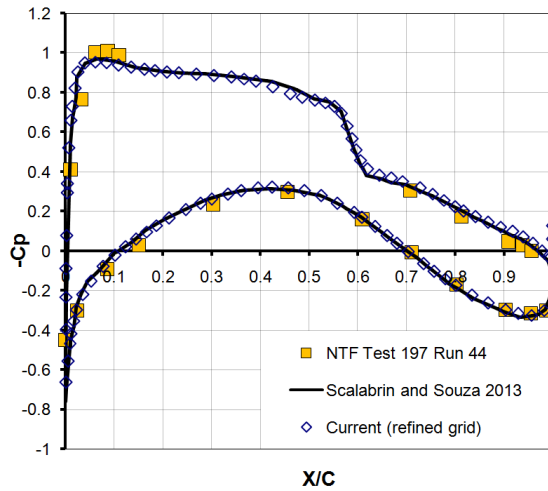


Figure 7.72: Pressure coefficient distribution at wing span-wise section 39.71% (turbulent flow over the CRM WB aircraft).

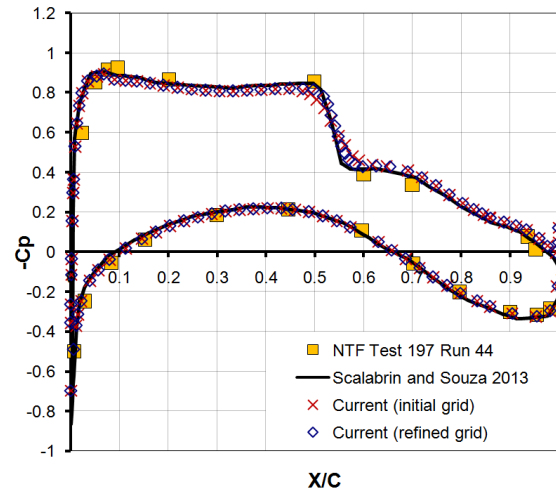


Figure 7.73: Pressure coefficient distribution at wing span-wise section 50.24% (turbulent flow over the CRM WB aircraft).

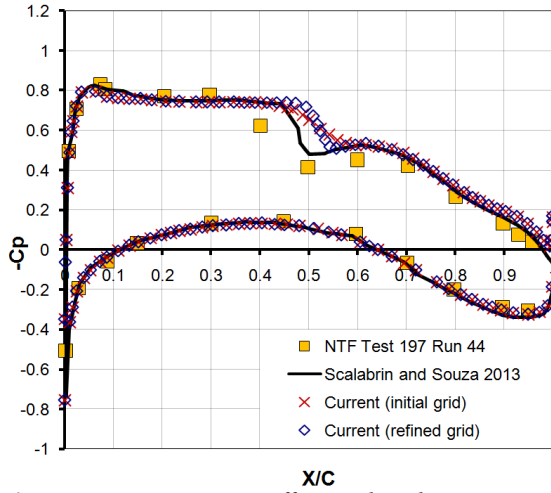


Figure 7.74: Pressure coefficient distribution at wing span-wise section 72.68% (turbulent flow over the CRM WB aircraft).

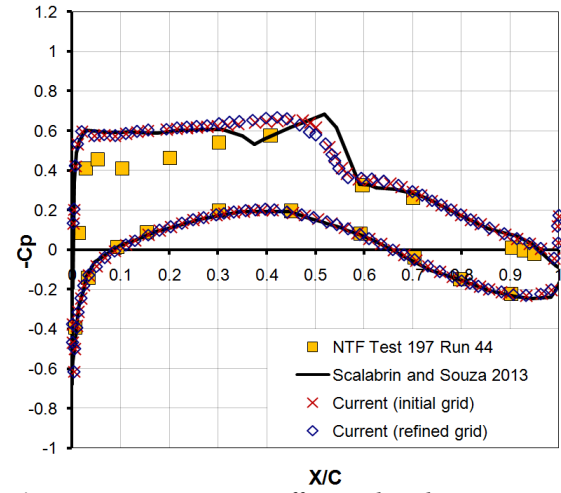


Figure 7.75: Pressure coefficient distribution at wing span-wise section 95.00% (turbulent flow over the CRM WB aircraft).

Table 7.10: Lift and idealized drag coefficients ($C_L \sim 0.5$, turbulent flow over the CRM WB aircraft).

	C_L	$C_{D_{idealized}}$
Current (initial coarse grid)	0.5014	0.01939
Current (refined coarse grid)	0.5016	0.01934
Vos and Sanchi 2010 [Vos10]	~ 0.5	~ 0.01960
NTF Run 44 [Vas10]	~ 0.5	~ 0.01671

An additional analysis was performed, considering the different available options in *Galatea* (slope limiters and schemes, implemented for the computation of inviscid and viscous fluxes, respectively). The corresponding runs were accomplished on a DELL T7500 workstation with two Intel^(R) Xeon^(R)-X5550 four-core processors at 2.67 GHz, utilizing the refined grid, divided in eight partitions for parallel processing. In order to implement the developed multigrid scheme, three coarser polyhedral sub-grids were constructed for each sub-domain via full-coarsening directional agglomeration.

The first part of this sensitivity analysis includes simulations with the limiting functions of Van Albada-Van Leer, Min-mod and Barth-Jespersen; for all of them the nodal-averaging scheme for gradients' computation and consequently diffusive fluxes' evaluation was employed. In Figure 7.76 the density (left) and turbulent kinetic energy (right) convergence history per multigrid cycles for the runs with these three different slope limiters is presented. Density residual was reduced approximately four orders of magnitude with Van Albada-Van Leer and Min-mod, while only two with Barth-Jespersen, resulting in an unconverged and, consequently, wrong solution; the limiter of Barth-Jespersen is acknowledged in some studies to exhibit poor convergence characteristics due to the overreacted limiting of small amplitude noise [Del13]. As far as the rest two limiting approaches (Van Albada-Van Leer and Min-mod) are concerned, both qualitative and quantitative differences were observed between them, despite their similar convergence behaviour. Figure 7.77 illustrates the finally extracted Mach number contours on a section at 70% of the wing span for both of them; Van Albada-Van Leer (left) produced a more accurate solution compared to Min-mod (right), identified by the sharper shock wave as well as by its more forward position on this section. Inaccuracy for the second limiter derives from its smoothing behaviour, as it always chooses the slope with the smallest magnitude [Lyg13c]. The aforementioned qualitative differentiation is verified quantitatively with the comparison of the corresponding pressure coefficient distributions at wing span-wise sections 50.24% and 72.68%, shown in Figures 7.78 and 7.79,

respectively. Van Albada-Van Leer slope limiter superiority is clearly demonstrated at shock wave formation region as well as ahead of it (especially at section 72.68%).

The second part of this analysis includes runs with either the nodal-averaging or the element-based (edge-dual volume) scheme, used for the computation of the gradients at the middle of each edge and consequently the evaluation of the viscous fluxes at the same point. Considering the results of the previous part, the simulations were performed with the superior Van Albada-Van Leer slope limiter. Although the element-based approach is assumed to be more accurate [Bla01], no differences were observed between the extracted results as well as between the convergence histories per multigrid cycles. Nevertheless, differences do exist regarding the corresponding density (left) and turbulent kinetic energy (right) convergence history curves per wall-clock computation time, as illustrated in Figure 7.80. The edge-dual volume run lasted almost twice the time required by the nodal-averaging one, in order the density residual to be reduced at least three order of magnitude. Thus, the superior efficiency of the nodal-averaging scheme is revealed once more, resolving capability from the edge-based structure it uses, whose dimensions are *a priori* smaller from those of the element-based methodology.

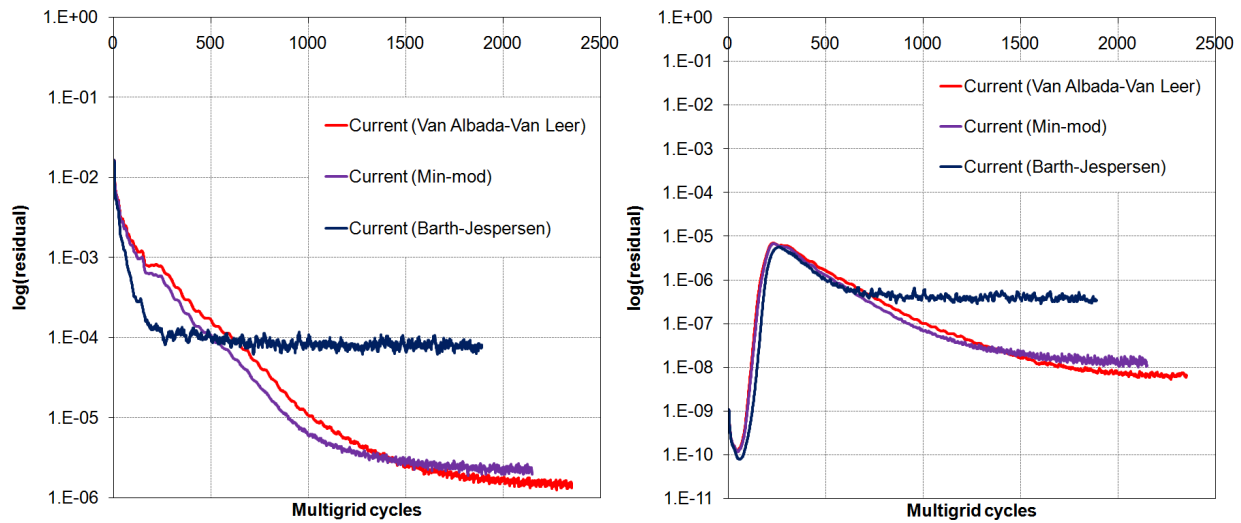


Figure 7.76: Density (left) and turbulent kinetic energy (right) convergence history for three different slope limiters (turbulent flow over the CRM WB aircraft).

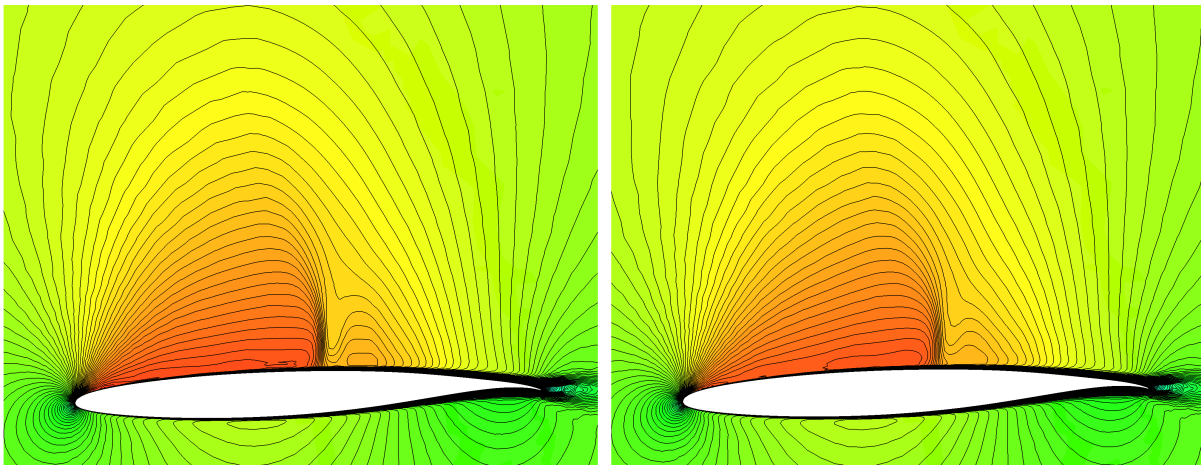


Figure 7.77: Mach number contours on a section at 70% of the wing span obtained with limiters of Van Albada-Van Leer (left) and Min-mod (right) (turbulent flow over the CRM WB aircraft).

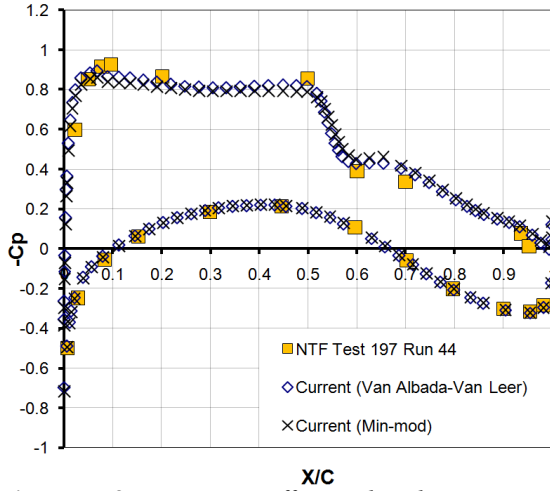


Figure 7.78: Pressure coefficient distribution at wing span-wise section 50.24% with different limiters (turbulent flow over the CRM WB aircraft).

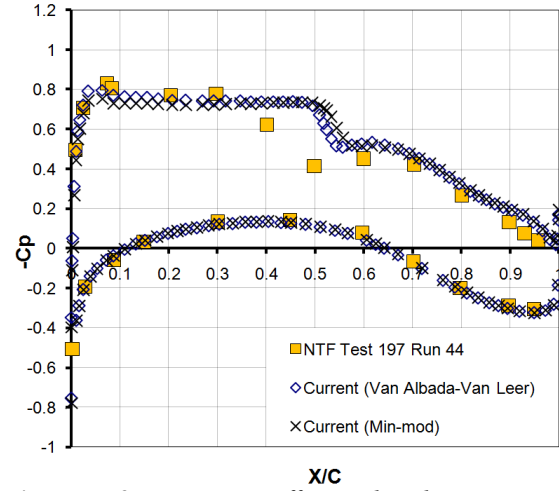


Figure 7.79: Pressure coefficient distribution at wing span-wise section 72.68% with different limiters (turbulent flow over the CRM WB aircraft).

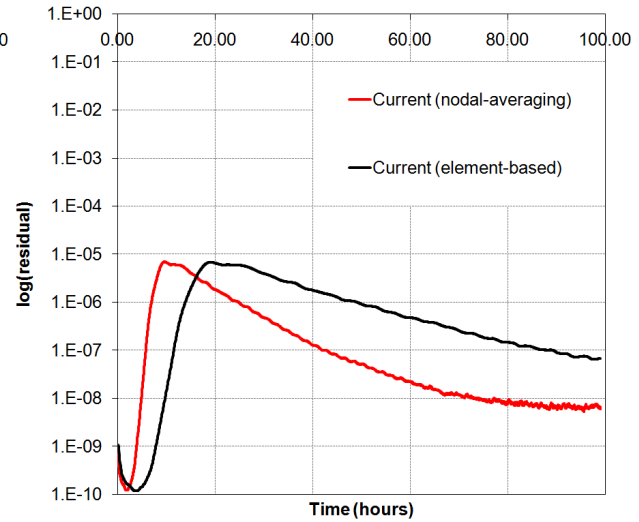
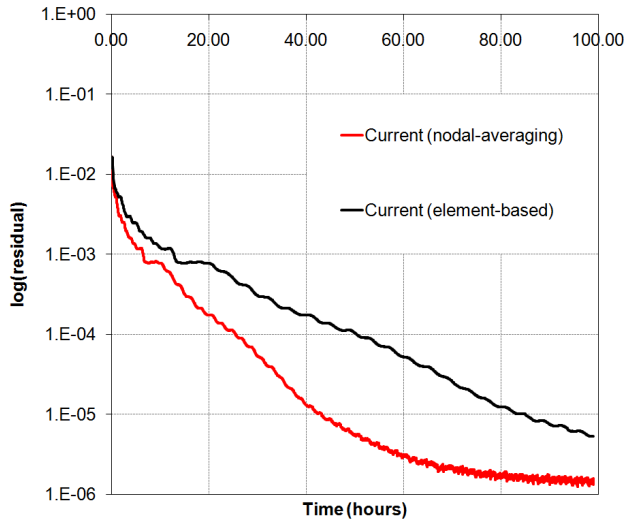


Figure 7.80: Density (left) and turbulent kinetic energy (right) convergence histories, obtained with nodal-averaging and element-based scheme (turbulent flow over the CRM WB aircraft).

Since the most suitable options of *Galatea* solver (Van Albada-Van Leer limiting function and nodal-averaging approach) were identified for this problem, a downwash study was performed similarly to these of DPW's participants; the flow on the refined grid but in different angles of attack, namely 0° , 1° , 3° and 4° was considered. Figure 7.81 includes the distribution of lift coefficient for the aforementioned angles of attack, while in Figure 7.82 the same distribution is illustrated for different values of idealized drag coefficient. The obtained and reference lift coefficient distributions are close across all the encountered angles of attack; a break in lift curve is identified prior to 4° , similarly to other codes applying the *SST* turbulence model [Lev13]. An also satisfactory comparison is observed for the lift coefficient - idealized drag coefficient curve [Vos10]; an overestimation is identified again comparing to experimental data [Vas10] due to previously analyzed reasons.

The incorporated spatial agglomeration multigrid methodology was also validated on this test case; single-grid and multigrid runs were performed on a DELLTM R815 Poweredge server with four AMD OpteronTM 6380 sixteen-core processors at 2.50 Ghz. Three coarser grids were generated via semi- and full-coarsening directional agglomeration, while for parallelization reasons the initial mesh was decomposed in eight partitions; in Figure 7.83 far views of the initial and agglomerated control volumes

are illustrated, while Figure 7.84 includes the corresponding close-up views on the symmetry surface. Inviscid fluxes were obtained with a second-order accurate spatial scheme, coupled with the Min-mod limiter, while the computation of the viscous ones was based on the nodal-averaging method. Figures 7.85 and 7.86 present the convergence history for density and turbulent kinetic energy respectively per number of iterations/cycles and time; in case of more efficient full-coarsening run, time speed-up coefficients (for density and turbulent kinetic energy) equal to ~ 4.9 and ~ 4.3 were succeeded respectively, considering the final residual of the single-grid simulation.

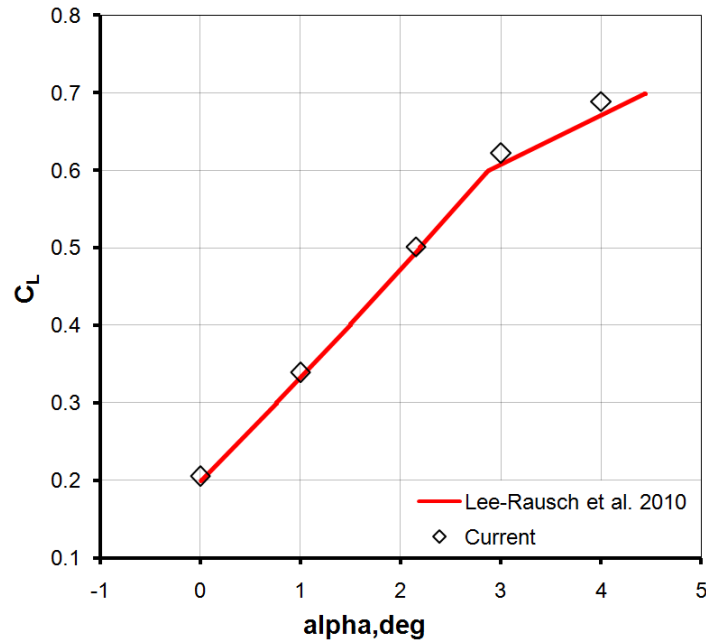


Figure 7.81: Lift coefficient for different values of angle of attack (turbulent flow over the CRM WB aircraft).

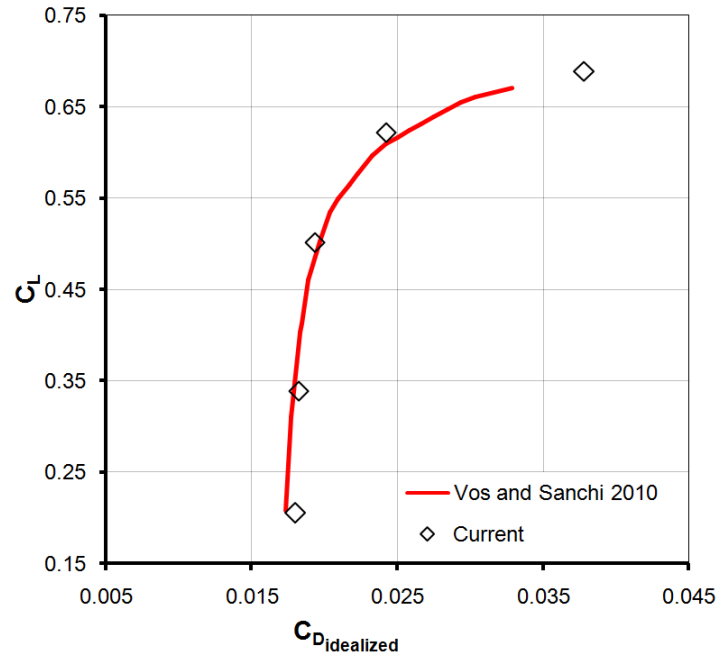


Figure 7.82: Lift coefficient - idealized drag coefficient curve (turbulent flow over the CRM WB aircraft).

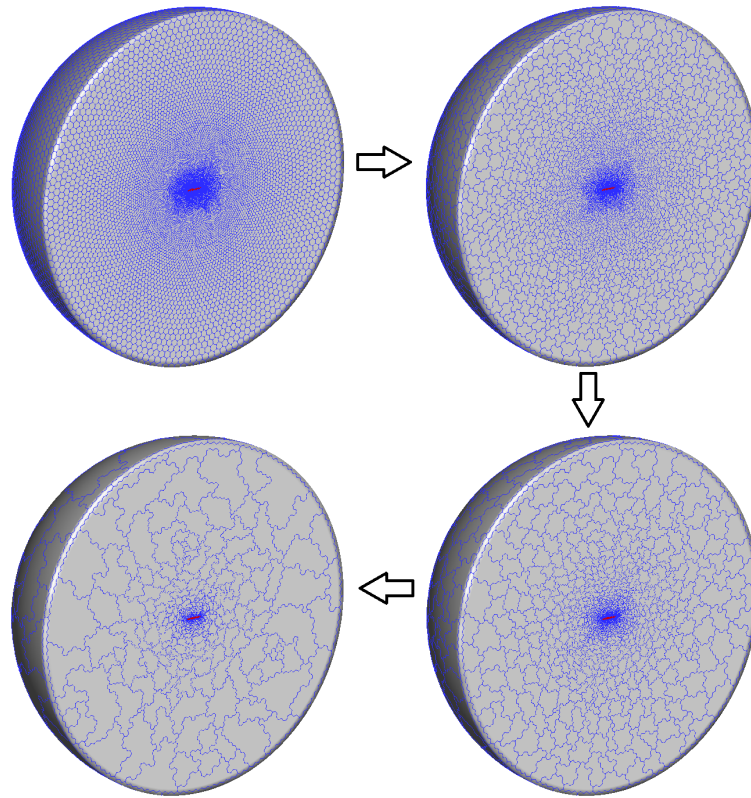


Figure 7.83: Initial and agglomerated control volume grids (turbulent flow over the CRM WB aircraft).

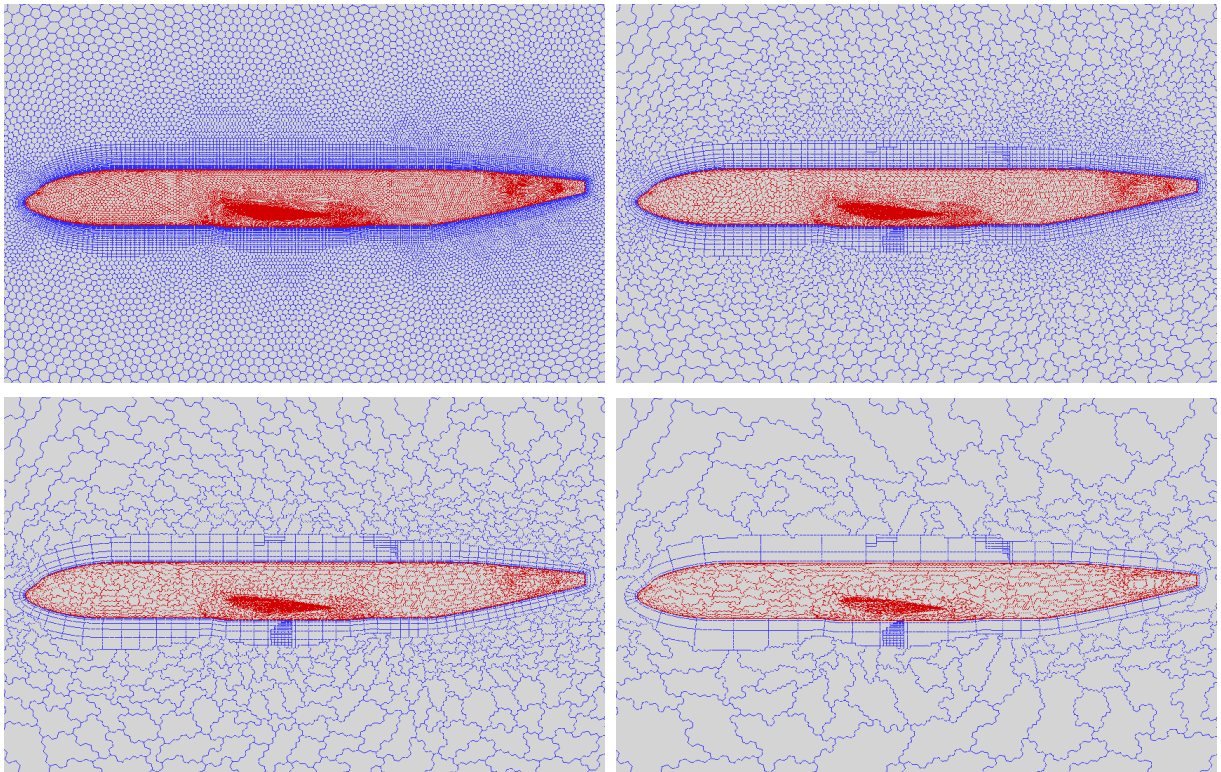


Figure 7.84: Mesh density on the symmetry surface of the initial and agglomerated control volume grids (turbulent flow over the CRM WB aircraft).

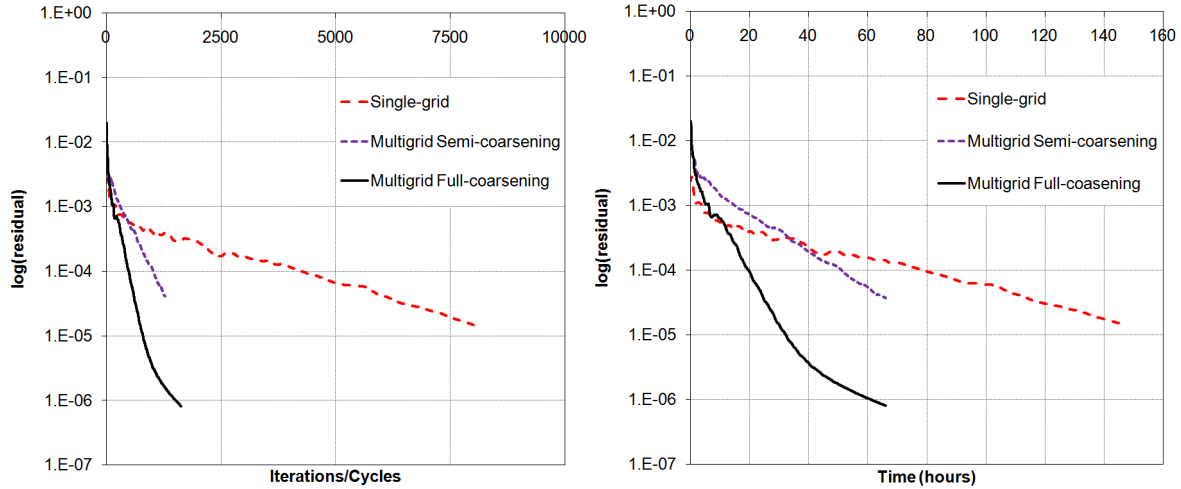


Figure 7.85: Density convergence history per iterations/cycles and time (turbulent flow over the CRM WB aircraft).

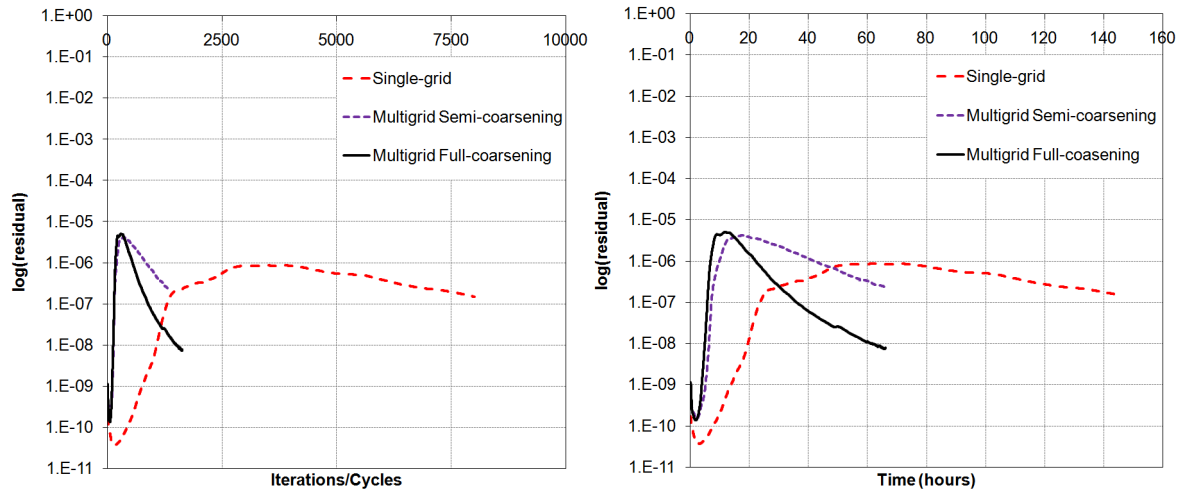


Figure 7.86: Turbulent kinetic energy convergence history per iterations/cycles and time (turbulent flow over the CRM WB aircraft).

7.8.2. CRM wing-body-horizontal-tail (WBHT) aircraft configuration

The utilized grid for the aircraft configuration with horizontal-tail (WBHT) consisted of more DoF's than the WB one to adequately describe the extra surfaces; it includes 4,653,134 nodes, 8,226,704 tetrahedra, 2,685,704 prisms and 53,630 pyramids. The angle of attack of the free-stream flow was set equal to 2.33° , considering again the desired value equal to 0.5 of the lift coefficient. Implementing a three-level multigrid scheme along with a twelve-partitions parallel processing on a DELL T7500 workstation with two Intel^(R) Xeon^(R)-X5650 six-core processors at 2.67 GHz, the density residual was decreased approximately three orders of magnitude in 28 hours. Similarly to the WB configuration, the nodal-averaging approach was implemented for gradients' computation, while inviscid fluxes were computed with a second-order accurate spatial scheme utilizing the Van Albada-Van Leer limiting function. The parameters of the simulation are summarized in the following Table 7.11. As the grid wasn't fine enough to capture the shock formations at the upper side of the wing, a similar to WB aircraft test case mesh refinement was implemented, deriving a new grid with 6,334,399 nodes, 10,920,171 tetrahedra, 4,571,468 prisms and 55,697 pyramids; the initial grid results were interpolated on the new resolution and the simulation was continued. In addition, in order to verify that h-refinement methodology indeed improves the accuracy of the final steady-state solution, the same test case (same simulation features) was faced with a medium grid, composed of 9,265,094 nodes, 7,702,336 prisms, 15,711,044

tetrahedra and 96,285 pyramids; the area above the outer half wing compartment at this medium mesh was also h-refined, producing a new grid, consisted of 9,761,321 nodes, 8,009,980 prisms, 16,593,580 tetrahedra and 96,881 pyramids. The angle of attack wasn't adjusted for neither of the two meshes (refined coarse and medium).

Table 7.11: Parameters of simulation (turbulent flow over the CRM WBHT aircraft).

Parameters	
Type of flow	Turbulent (SST)
Mach number	0.85
Reynolds number	5.0E+6
Angle of attack (deg.)	2.33°
Grid density	4,653,134 nodes, 8,226,704 tetrahedra, 2,685,704 prisms and 53,630 pyramids
Number of partitions	12
Number of multigrid levels	3
Iterative scheme	Runge-Kutta method (CFL=1.0)
Computational system	DELL T7500 workstation with two Intel ^(R) Xeon ^(R) -X5650 six-core processors at 2.67 GHz

In Figure 7.87 the refined coarse mesh density and predicted dimensionless pressure contours on the aircraft surface are illustrated, demonstrating their qualitative agreement with those of other solvers [Lee10, Mav10]; similarly to the WB aircraft and independently of the mesh density no significant wing-body juncture separation was observed, while the same stands for horizontal tail-body juncture separation (probably due to the SST turbulence model utilized [Sch13]). Figures 7.88-7.101 contain the pressure coefficient distributions at different wing and horizontal tail span-wise sections for the initial coarse, the refined coarse, the initial medium and the refined medium mesh, compared to those of the study of Lee-Rausch et al. [Lee10]; in Figures 7.88-7.90 and 7.95-7.101 the distributions, derived by the initial grids (coarse and medium) are not contained, as they are the same to those of the corresponding refined ones. As expected, the medium grids produced more accurate results than the coarse ones, while the adapted grids provided sharper pressure distributions at shock regions compared to the initial unrefined ones. It should be highlighted that the slope limiters reduce the order of the spatial discretization scheme at shock areas; therefore, the accurate prediction of this discontinuity is mainly based on the local grid density. In that way the importance of the incorporated h-refinement methodology is revealed; depending on the examined test case, the construction of a new finer mesh from the very beginning might be avoided along with significant computational savings. Moreover, in wing tip section (95%) *Galatea* is revealed to predict more accurately the shock at the upper side of the wing, independently of the density of the employed grid.

Correspondingly to Table 7.10 for WB configuration, Table 7.12 contains the lift and idealized drag coefficients obtained with the initial coarse, the refined coarse, the initial medium and the refined medium grid for WBHT aircraft, in comparison with the experimental and the computed by Vos and Sanchi [Vos10] ones. Despite no adjustment was made to the angle of attack, the extracted lift coefficients agree sufficiently with both the reference computed and experimental ones [Vas10, Vos10]; similarly to WB aircraft a slight change of lift coefficient is observed with grid adaptation and the utilization of the medium mesh. The idealized drag coefficients obtained with the initial and refined coarse grids are close enough with the reference computed ones [Vos10]; however, their values are overestimated, compared to the experimental coefficients [Vas10]. In case of medium grids this difference is decreased considerably; it is closer to the experimental than the reference computed one, indicating therefore the insufficiency of the employed initial and refined coarse grids.

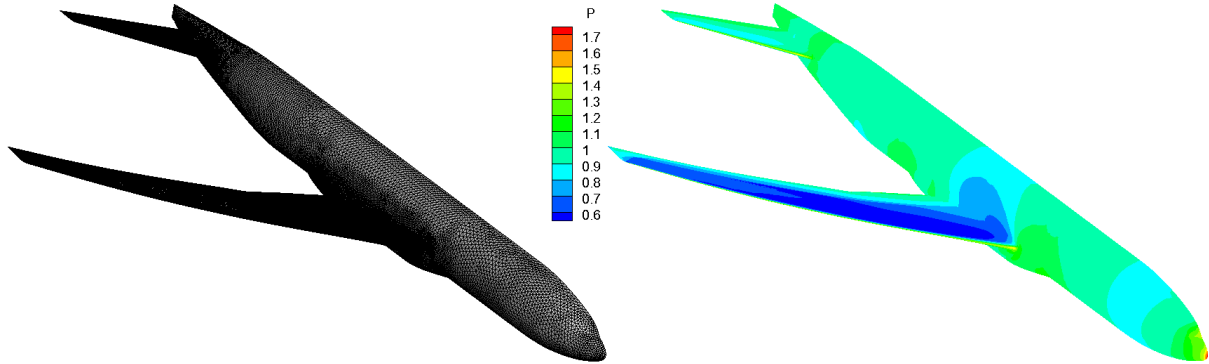


Figure 7.87: Mesh density (left) and predicted dimensionless pressure contours (right) on the surface of the aircraft (turbulent flow over the CRM WBHT aircraft).

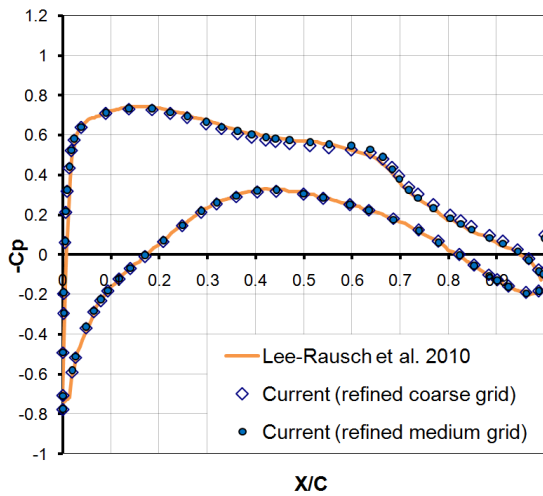


Figure 7.88: Pressure coefficient distribution at wing span-wise section 13.06% (turbulent flow over the CRM WBHT aircraft).

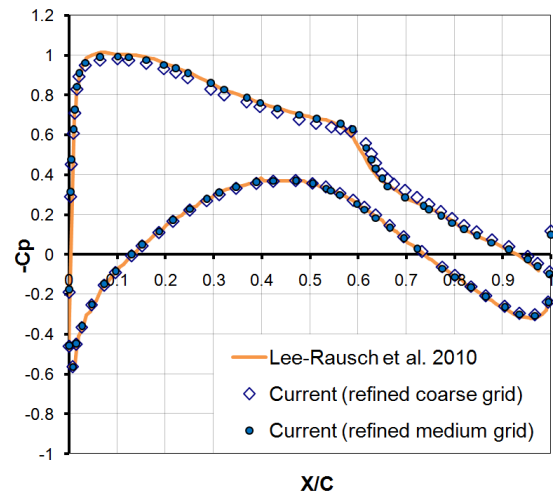


Figure 7.89: Pressure coefficient distribution at wing span-wise section 28.28% (turbulent flow over the CRM WBHT aircraft).

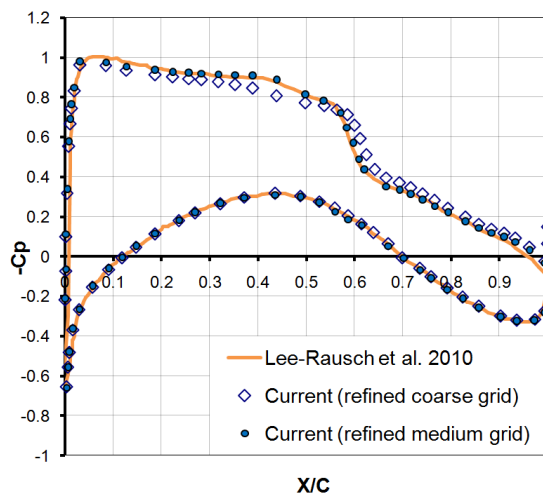


Figure 7.90: Pressure coefficient distribution at wing span-wise section 39.71% (turbulent flow over the CRM WBHT aircraft).

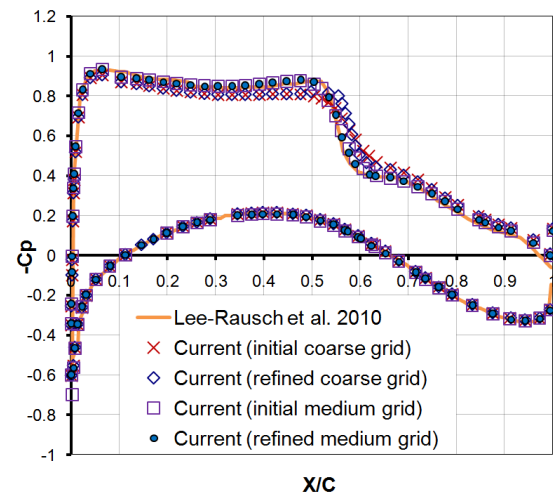


Figure 7.91: Pressure coefficient distribution at wing span-wise section 50.24% (turbulent flow over the CRM WBHT aircraft).

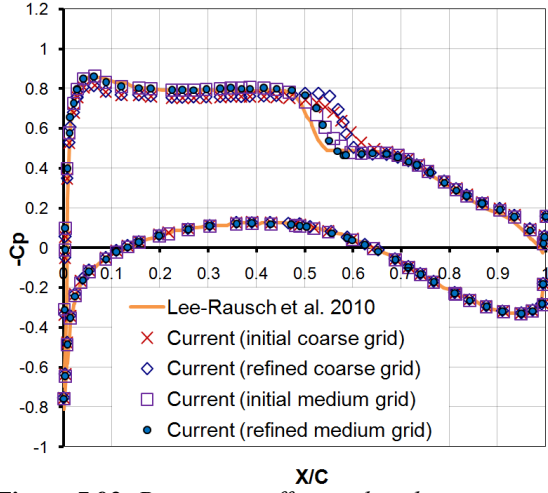


Figure 7.92: Pressure coefficient distribution at wing span-wise section 72.68% (turbulent flow over the CRM WBHT aircraft).

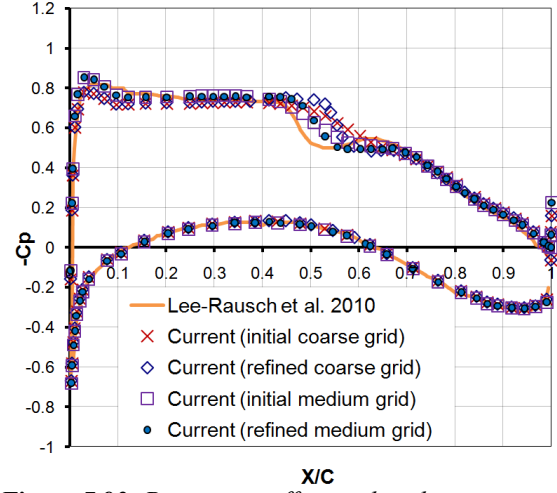


Figure 7.93: Pressure coefficient distribution at wing span-wise section 84.56% (turbulent flow over the CRM WBHT aircraft).

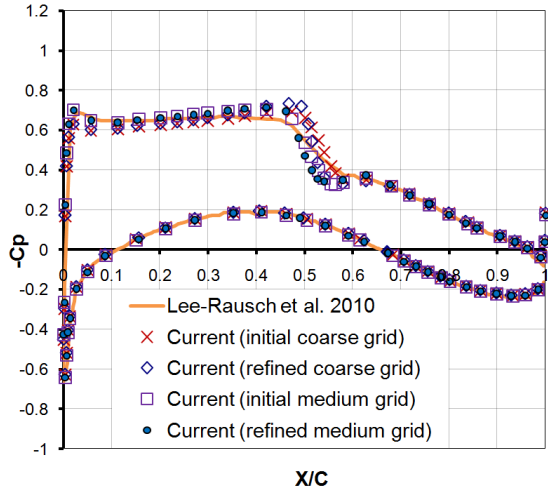


Figure 7.94: Pressure coefficient distribution at wing span-wise section 95.00% (turbulent flow over the CRM WBHT aircraft).

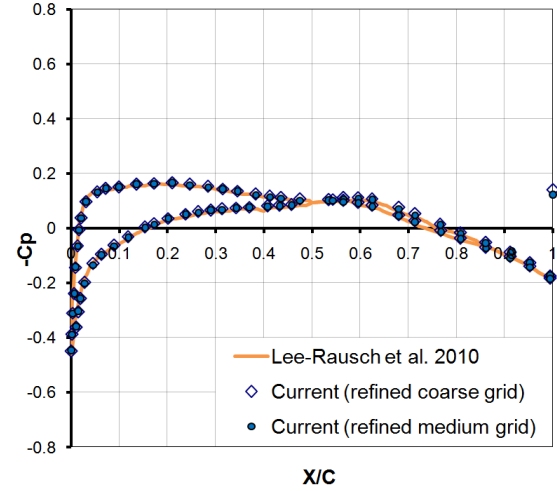


Figure 7.95: Pressure coefficient distribution at horizontal-tail span-wise section 18.00% (turbulent flow over the CRM WBHT aircraft).

Table 7.12: Lift and idealized drag coefficients ($C_L \sim 0.5$, turbulent flow over the CRM WBHT aircraft).

	C_L	$C_{D_{idealized}}$
Current (initial coarse grid)	0.5027	0.02161
Current (refined coarse grid)	0.5031	0.02145
Current (initial medium grid)	0.5092	0.01955
Current (refined medium grid)	0.5088	0.01955
Vos and Sanchi 2010 [Vos10]	~ 0.5	~ 0.02210
NTF Run 97 [Vas10]	~ 0.5	~ 0.01877

A downwash study, considering different angles of attack (0° , 1° , 3° and 4°), was performed similarly to the WB aircraft test case, employing the refined grid. Figure 7.102 includes the distribution of lift coefficient for the aforementioned angles of attack, while in Figure 7.103 the same distribution is illustrated for different values of idealized drag coefficient. The obtained results compare sufficiently with the reference computed ones [Lee10, Vos10], while a break in the lift curve is also identified at about 3° .

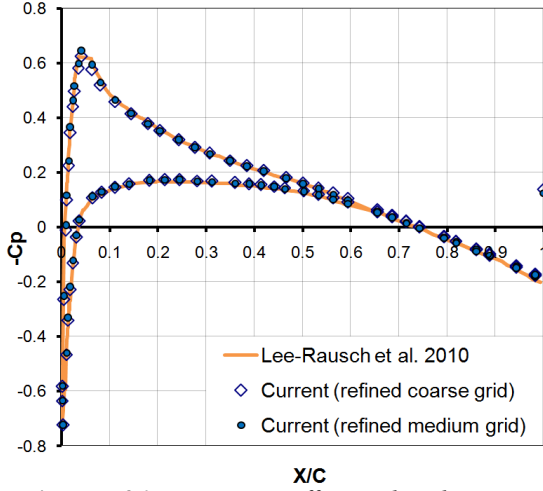


Figure 7.96: Pressure coefficient distribution at horizontal-tail span-wise section 30.00% (turbulent flow over the CRM WBHT aircraft).

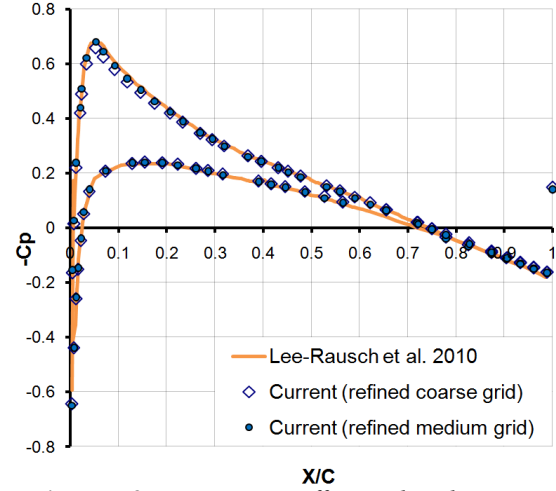


Figure 7.97: Pressure coefficient distribution at horizontal-tail span-wise section 50.00% (turbulent flow over the CRM WBHT aircraft).

A deviation of the idealized drag coefficient curve is identified for greater values probably due to the insufficient number of DoF's; nevertheless, considering the scattering of the corresponding distributions, included in the summary paper of the fourth DPW [Vas10], the current one is still located among of them. In addition, Figure 7.104 includes close-up views of the wing and tail surfaces along with the obtained dimensionless pressure contours in case of a 4° angle of attack; a shock-including flow on the main wing and a fully attached one on the tail are observed. Their comparison with the corresponding ones of Mavriplis and Long [Mav10] confirms once more the capability of the developed methodology for such simulations.

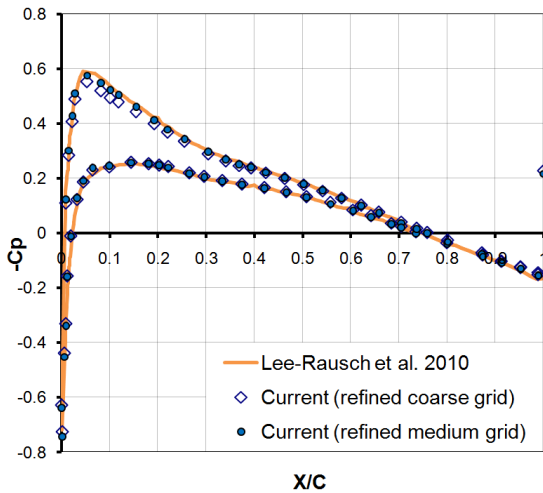


Figure 7.98: Pressure coefficient distribution at horizontal-tail span-wise section 70.00% (turbulent flow over the CRM WBHT aircraft).

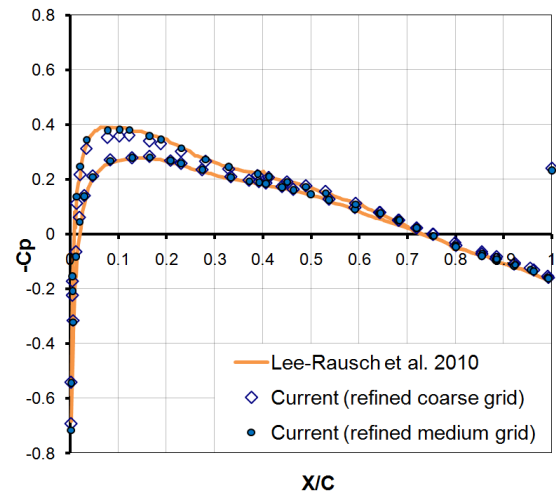


Figure 7.99: Pressure coefficient distribution at horizontal-tail span-wise section 90.00% (turbulent flow over the CRM WBHT aircraft).

Finally, the refined grid (6,334,399 nodes, 10,920,171 tetrahedra, 4,571,468 prisms and 55,697 pyramids), examined in this test case, was used in order to evaluate the incorporated in the proposed algorithm acceleration techniques, namely the parallelization strategy and the agglomeration multigrid scheme. Therefore, single-grid and multigrid (three-level full-coarsening fusion, only FAS) simulations employing the explicit Runge-Kutta method (CFL=1.0), but different numbers of partitions (2, 4, 8, 16

and 32) were run on a DELLTM R815 Poweredge server with four AMD OpteronTM 6380 sixteen-core processors at 2.50 Ghz. In Figure 7.105 far views of the initial and agglomerated control volumes grids are illustrated, while Figure 7.106 includes the corresponding close-up views of the symmetry/WBHT aircraft surface (16 partitions). Inviscid fluxes were obtained with a second-order accurate spatial scheme, coupled with the Min-mod limiter, while the computation of the viscous ones was based on the nodal-averaging method. Figure 7.107 illustrates the density (left) and turbulent kinetic energy (right) convergence history for the performed single-grid runs, while Figure 7.108 includes the corresponding speed-up/number of processors distribution. A sufficient agreement between the ideal and the current computation is observed up to 16 processors, while for more of them a deviation is identified due to the increase of communication load with the increase of used cores. It should be noted here that the parallelization strategy has not been optimized yet for larger number of processes.

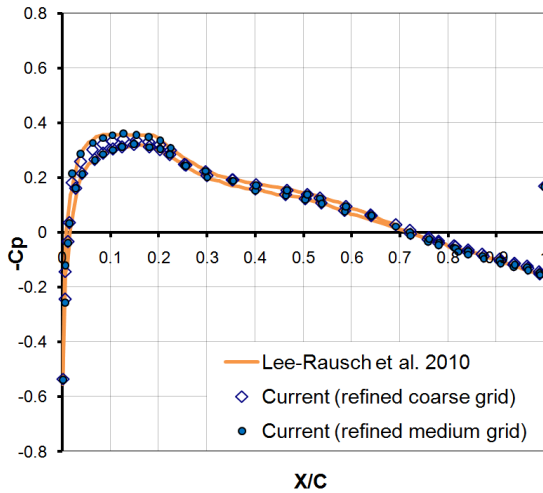


Figure 7.100: Pressure coefficient distribution at horizontal-tail span-wise section 95.00% (turbulent flow over the CRM WBHT aircraft).

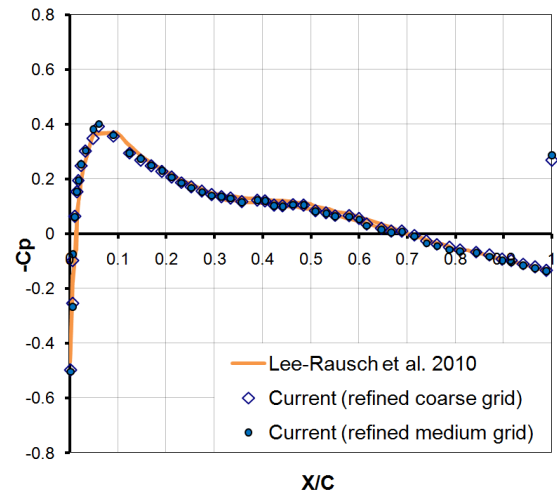


Figure 7.101: Pressure coefficient distribution at horizontal-tail span-wise section 99.00% (turbulent flow over the CRM WBHT aircraft).

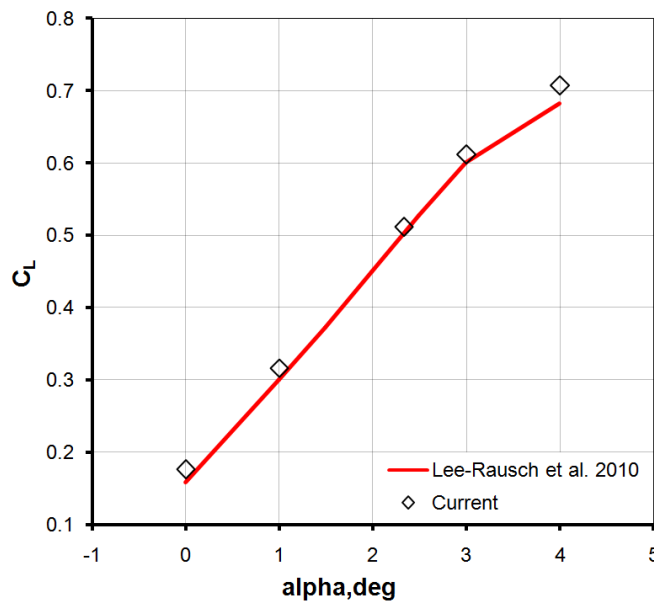


Figure 7.102: Lift coefficient for different values of angle of attack (turbulent flow over the CRM WBHT aircraft).

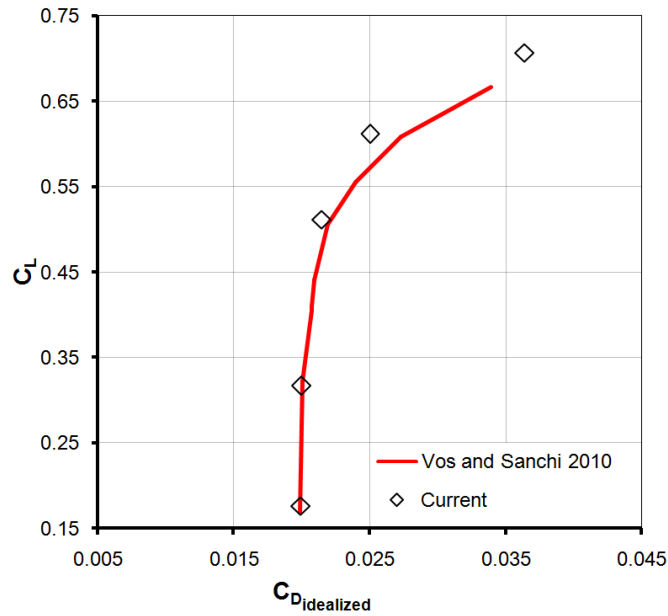


Figure 7.103: Lift coefficient - idealized drag coefficient curve (turbulent flow over the CRM WBHT aircraft).

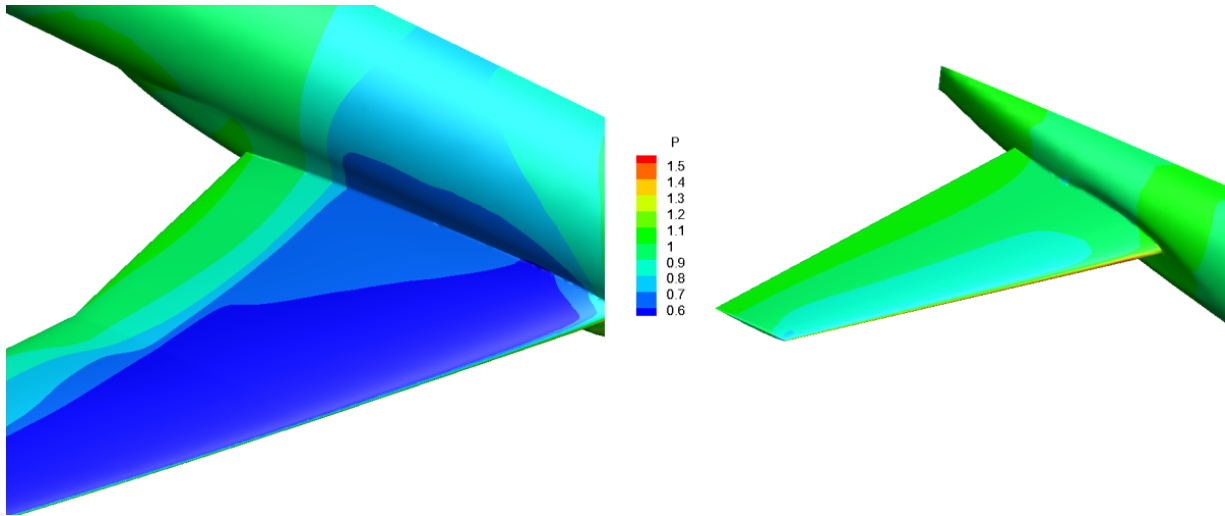


Figure 7.104: Shock-including flow on the main wing and fully attached one on the horizontal tail (4° angle of attack, turbulent flow over the CRM WBHT aircraft).

In Figures 7.109 and 7.110 the corresponding convergence results for multigrid simulations are presented. As expected, the deviation between the ideal and the current computation is increased, comparing to the single-grid distribution; nevertheless, the needed simulation time is considerably reduced, as the run combining multigrid scheme and parallel processing in 32 cores required less than 20 hours to decrease density residual more than three orders of magnitude.

Finally, Figure 7.111 includes the density convergence time history for the single-grid and multigrid simulations distributed in 2 and 32 processes. The considerable improvement of algorithm's computational performance by the incorporated acceleration techniques is clearly demonstrated; in simulation combining parallel computation in 32 processes and multigrid method a temporal speed-up equal to ~ 18.0 was succeeded, comparing to the single-grid run distributed to only two processors.

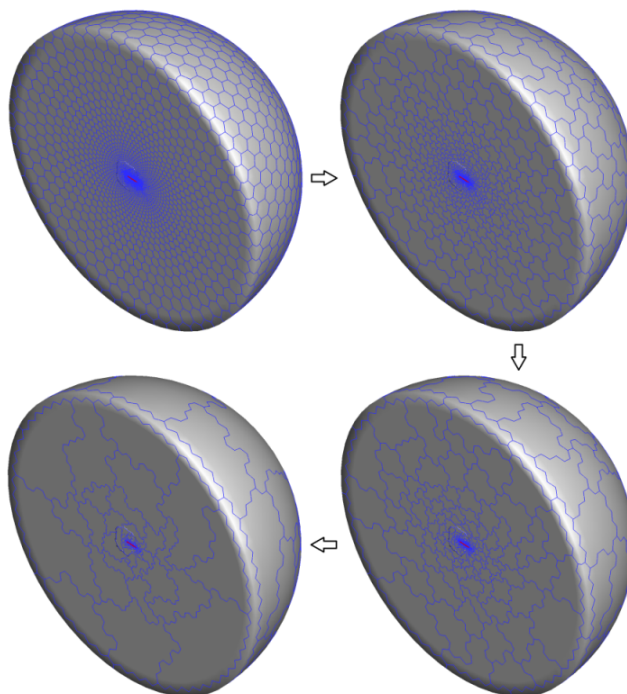


Figure 7.105: Initial and agglomerated control volume grids (turbulent flow over the CRM WBHT aircraft).

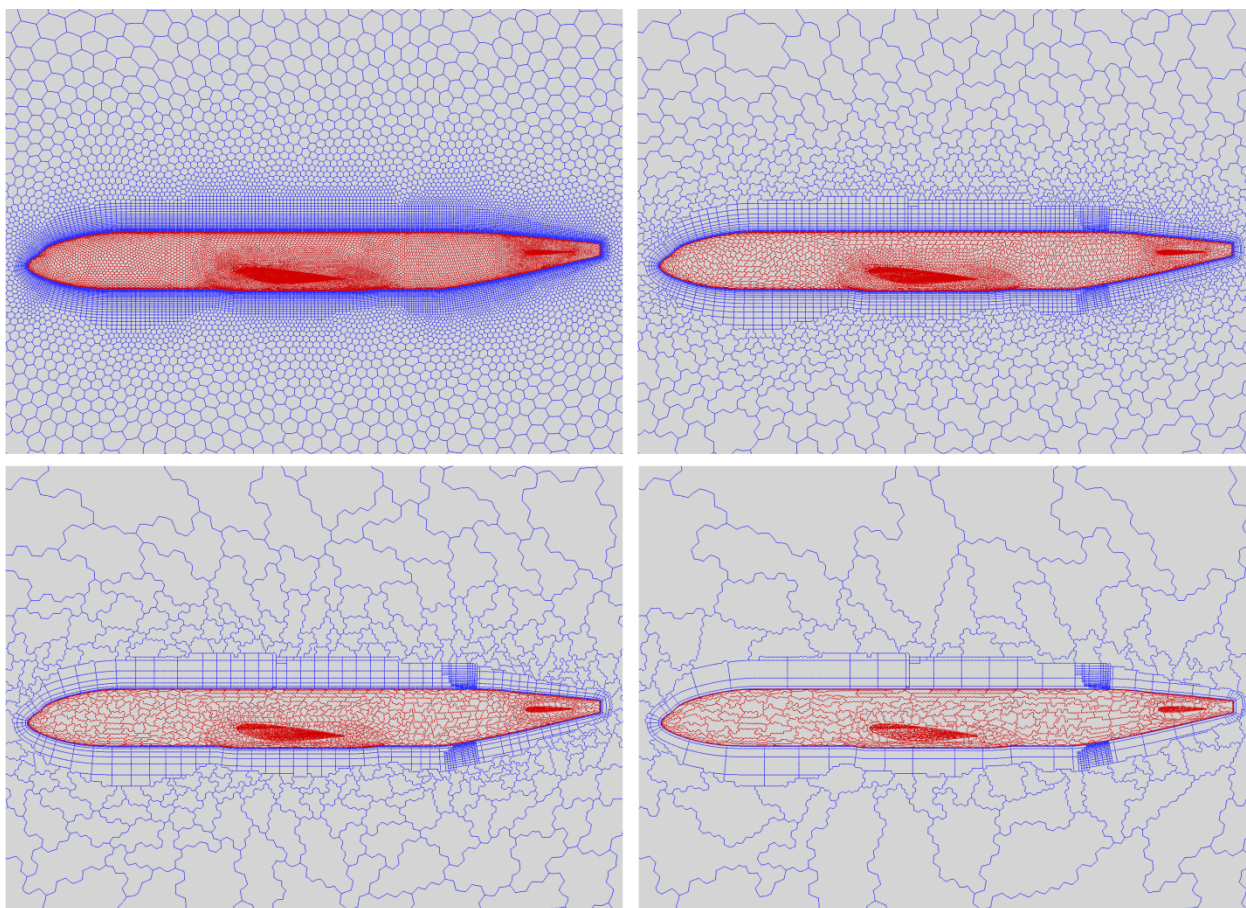


Figure 7.106: Mesh density on the symmetry surface of the initial and agglomerated control volume grids (turbulent flow over the CRM WBHT aircraft).

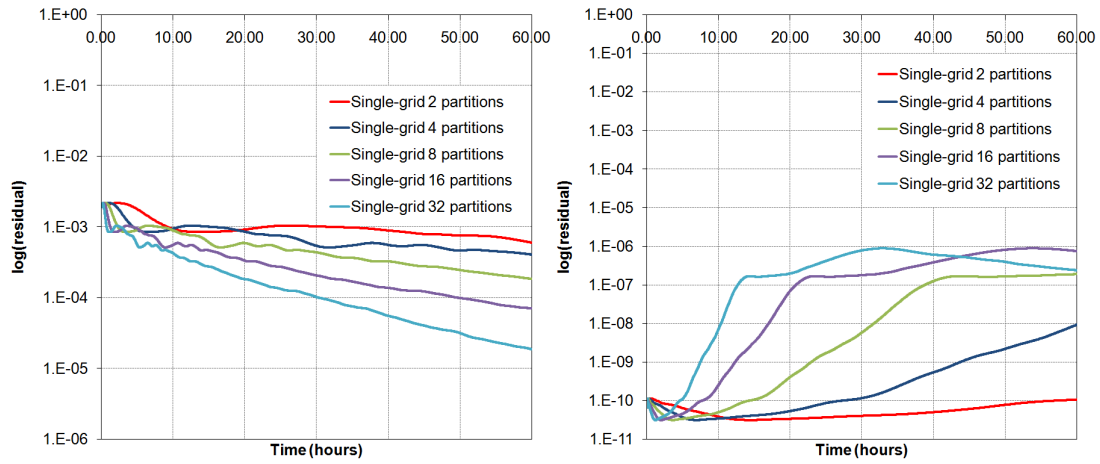


Figure 7.107: Density and turbulent kinetic energy convergence history per time (single-grid runs, turbulent flow over the CRM WBHT aircraft).

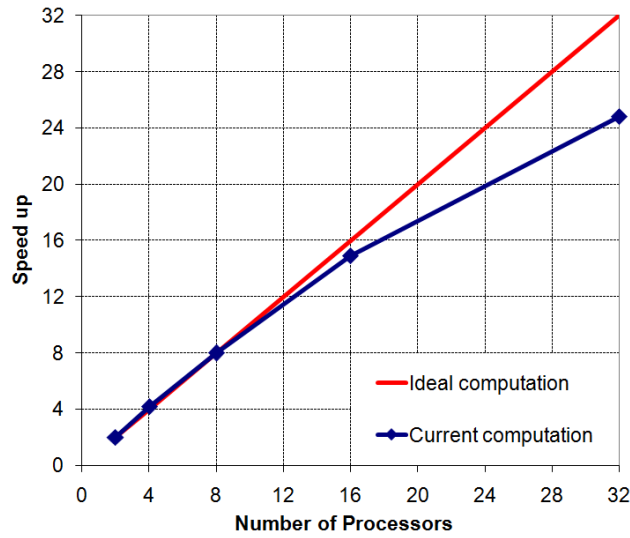


Figure 7.108: Scalability of parallel implementation (single-grid runs, turbulent flow over the CRM WBHT aircraft).

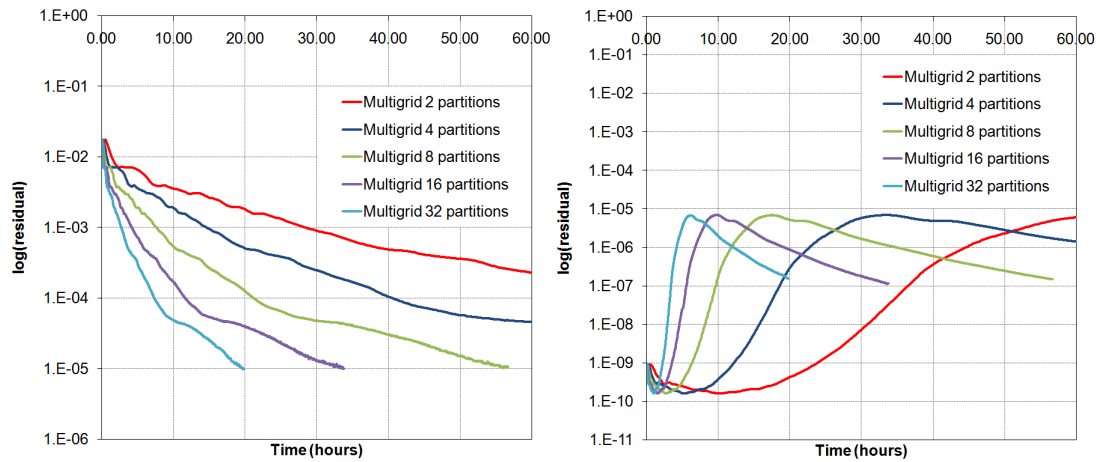


Figure 7.109: Density and turbulent kinetic energy convergence history per time (multigrid runs, turbulent flow over the CRM WBHT aircraft).

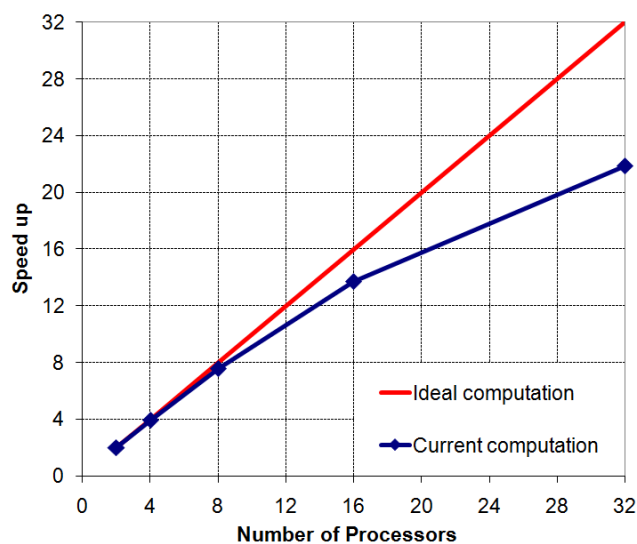


Figure 7.110: Scalability of parallel implementation (multigrid runs, turbulent flow over the CRM WBHT aircraft).

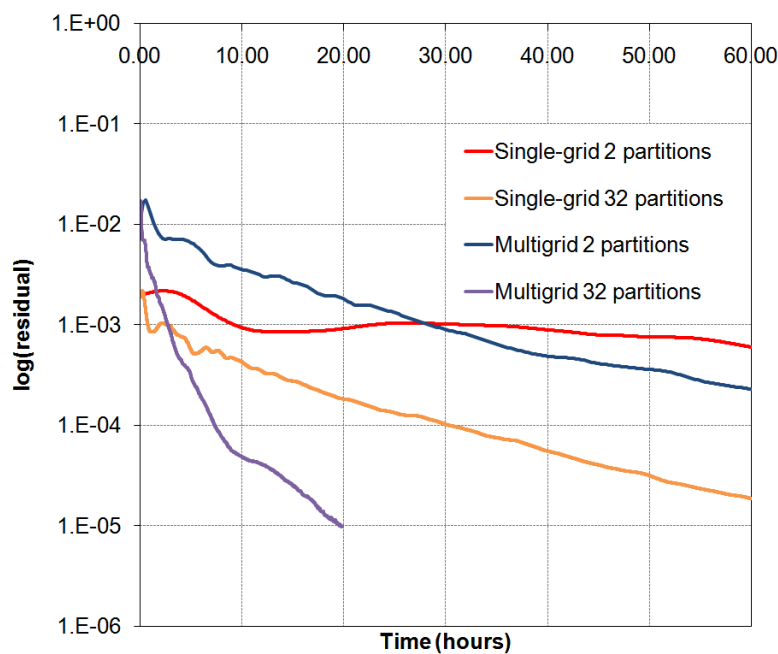


Figure 7.111: Density convergence history per time (single-grid and multigrid runs with 2 and 32 partitions, turbulent flow over the CRM WBHT aircraft).

8. Radiative heat transfer numerical results

8.1. Definitions

Similarly to the fluid flow methodology validation, the radiative heat transfer solver has been evaluated, especially in terms of accuracy and efficiency, against three- and quasi-three-dimensional benchmark test cases, while its results have been compared with those of reference solvers. This comparison is mainly based on the distribution of dimensionless incident radiative heat flux $Q^*(\vec{r})$ along a line on the enclosure under examination; the aforementioned heat flux is defined as [Kim88]

$$Q^*(\vec{r}) = \frac{Q(\vec{r})}{E} = \frac{\int_{4\pi} I(\vec{r}, \hat{s}) \cdot (\hat{s} \cdot \hat{n}_i) d\Omega}{\sigma T^4} = \frac{\sum_{mn}^{N_\theta \times N_\phi} I_p^{mn} D_{ci}^{mn}}{\sigma T^4} \quad (8.1)$$

where $Q(\vec{r})$ denotes the incident radiative heat flux, while E is the initial radiative energy generating the heat transfer, e.g., in case that the medium has the highest temperature E_m is used, while in case that boundary walls have the highest temperature E_w is applied instead. Considering that in most of the cases the radiative flux in a specific direction (x, y, z) is required, the appropriate directional weight $(D_{ci,x}^{mn}, D_{ci,y}^{mn}, D_{ci,z}^{mn})$, based on the corresponding unit normal vector $(\hat{e}_x, \hat{e}_y, \hat{e}_z)$, is utilized in equation (8.1). In addition, the average radiative heat flux G is computed, mainly for the visualization of the extracted solutions; it is defined as follows [Kim88]

$$G(\vec{r}) = \int_{4\pi} I(\vec{r}, \hat{s}) d\Omega = \sum_{mn}^{N_\theta \times N_\phi} I_p^{mn} \Delta\Omega^{mn} \quad (8.2)$$

while in case its dimensionless formulation is required, it is divided (similarly to the dimensionless incident radiative heat flux) with $4E$.

In order to validate the computational performance of the developed methodology, the radiative intensity residual is computed, similarly to density residual, at each cycle (of multigrid runs) or at each iteration (of single-grid runs) $l+1$ as

$$residual = \frac{\sum_{p=1}^{N_p} \sum_{m=1}^{N_\theta} \sum_{n=1}^{N_\phi} |I_p^{mn,l+1} - I_p^{mn,l}|}{N_p N_\theta N_\phi} \quad (8.3)$$

where N_p denotes the total number of nodes of the non-agglomerated mesh, while N_θ and N_ϕ represent the numbers of the initially defined polar and azimuthal control angles respectively; a decrease of residual about two or three orders of magnitude is usually enough to extract the desired radiative heat flux distributions. Additionally, throughout the test cases in which spatial and/or angular agglomeration multigrid method is employed, the following notation "SxAy" is used; "x" corresponds to the defined number of Spatial levels, while "y" to the defined number of Angular ones. For example, S1A1 denotes a single-grid run, S4A1 a four-level spatial agglomeration multigrid one and S4A3 a combined four-level spatial/three-level angular agglomeration multigrid simulation. It should be noted that the solver has not been further optimized for computational performance.

8.2. A hexahedral enclosure with quadratic bases

The first (quasi-3D) benchmark test case concerns radiative heat transfer in a hexahedral enclosure with quadratic bases; its dimensions ($1, 1, 5$) m are illustrated in the following Figure 8.1 [Kim88, Lyg12b]. The included medium is a purely scattering one ($k_a=0 \text{ m}^{-1}$, $\sigma_s=1 \text{ m}^{-1}$) while all the walls are assumed cold ($T=0 \text{ K}$) and black ($\varepsilon_w=1$), except for the bottom face ($y=0 \text{ m}$), at which a constant heating energy is imposed ($E=1 \text{ W/m}^2$, corresponding to 64.80435 K). Both isotropic and anisotropic forward (F2) scattering was considered in this problem. The utilized tetrahedral mesh consists of $196,732$ nodes and $1,053,402$ tetrahedra, while for parallel computation on a DELL T7500 workstation with two Intel^(R) Xeon^(R)-X5650 six-core processors at 2.67 GHz , it was divided in twelve sub-domains; in Figure 8.2 the employed grid, representing the examined enclosure, as well as a close-up view of its quadratic bases' density is presented. The angular domain was discretized by sixteen azimuthal and eight polar control angles, while in order the overhang problem to be alleviated the bold approximation was employed. Radiative intensity at walls was explicitly imposed with Dirichlet boundary conditions, while the steady (non time-dependent) RTE was solved with iterative approximations along with a first-order accurate spatial scheme; it required about 1.7 s per iteration. The parameters of the simulation are summarized in the following Table 8.1.

Table 8.1: Parameters of simulation (hexahedral enclosure with quadratic bases).

Parameters	
Absorption coefficient	0 m^{-1}
Scattering coefficient	1 m^{-1}
Wall emissivity	1
Grid density	$196,732$ nodes and $1,053,402$ tetrahedra
Number of partitions	12
Angular resolution	16 azimuthal and 8 polar control angles
Iterative scheme	Iterative approximations
Computational system	DELL T7500 workstation with two Intel ^(R) Xeon ^(R) -X5650 six-core processors at 2.67 GHz

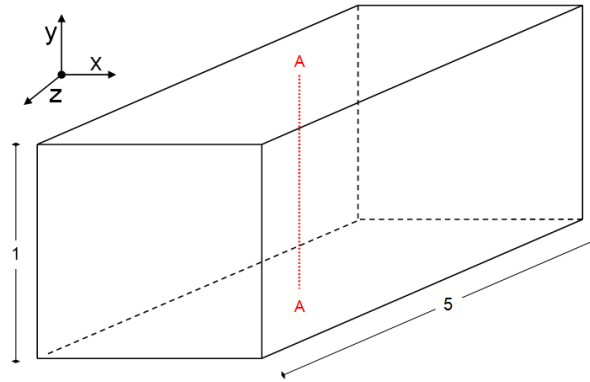


Figure 8.1: Dimensions of hexahedral enclosure with quadratic bases.

In Figures 8.3 and 8.4 the dimensionless incident (left) and average (right), respectively, radiative heat flux distributions along the A-A line, which corresponds to the centerline of the hexahedral enclosure in y -direction ($x=0.5 \text{ m}$ and $z=2.5 \text{ m}$), are illustrated for an isotropically and anisotropically scattering medium; they compare with the corresponding distributions extracted by the two-dimensional, cell-centered, DOM solver of Kim and Lee [Kim88], succeeding a satisfactory agreement. As expected, anisotropic scattering with respect to isotropic one derives greater values of radiative intensity, reaching

almost the value of imposed heating power at the bottom side. In general, with forward scattering functions (based on Legendre polynomials), i.e., F1 and F2, larger values of radiation are obtained than the isotropic and backward (B1 and B2) ones; the latter leads to even smaller values comparing to the isotropic one [Lyg12b]. Nevertheless, in dimensionless average heat flux curves, an alternation is observed between higher values of anisotropic and isotropic distributions; the isotropic function leads to higher values near the heating source surface, while the anisotropic one predicts more radiation near the upper boundary.

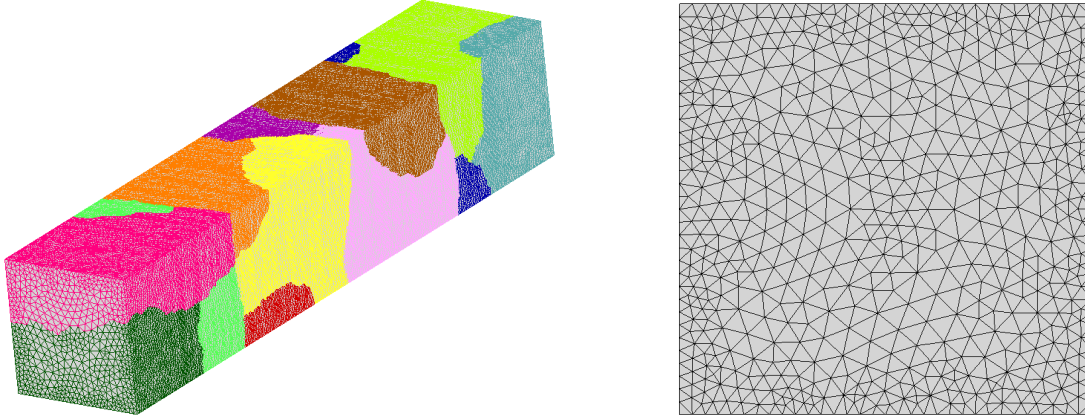


Figure 8.2: Employed grid and its density on one of its quadratic bases (hexahedral enclosure with quadratic bases).

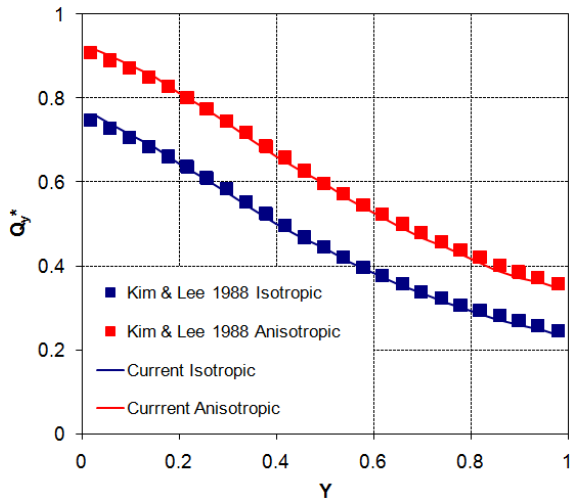


Figure 8.3: Distributions of dimensionless incident radiative heat flux along the A-A line (hexahedral enclosure with quadratic bases).

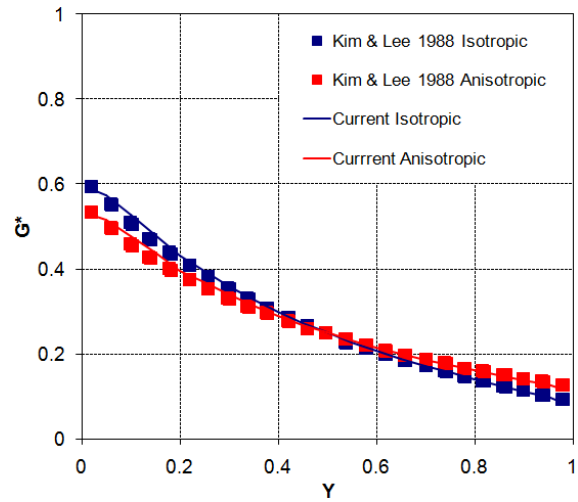


Figure 8.4: Distributions of dimensionless average radiative heat flux along the A-A line (hexahedral enclosure with quadratic bases).

Additional runs with different values of scattering albedo ($\omega = \sigma_s/\beta$) and wall emissivity ε_w were performed, in order to evaluate their contribution to the predicted radiative heat transfer. Figure 8.5 includes the distributions of dimensionless incident radiative heat flux along the A-A line for different scattering albedos, compared well with the corresponding ones obtained by the two-dimensional DOM algorithm of Kim and Lee [Kim88]; the values of extinction coefficient ($\beta = k_a + \sigma_s$) and wall emissivity ε_w remained equal to unity. As scattering albedo increases higher values of radiation are obtained, due to the forward effect of the employed function F2 [Lyg12b]. Similarly, Figure 8.6 illustrates the distributions of dimensionless incident radiative heat flux along the A-A line for different values of wall emissivity, compared well with those of the aforementioned reference solver [Kim88]; the values of extinction coefficient β and scattering albedo ω remained equal to unity. As expected, the emissivity coefficient

reduction leads to less produced radiation at the same distance from the heating source surface, because of the diffusive effect of the walls. The previous distributions (per values of scattering albedo and wall emissivity) appear to be consistent with the reference two-dimensional ones [Kim88], despite the fact that a three-dimensional method usually adds larger amounts of false scattering.

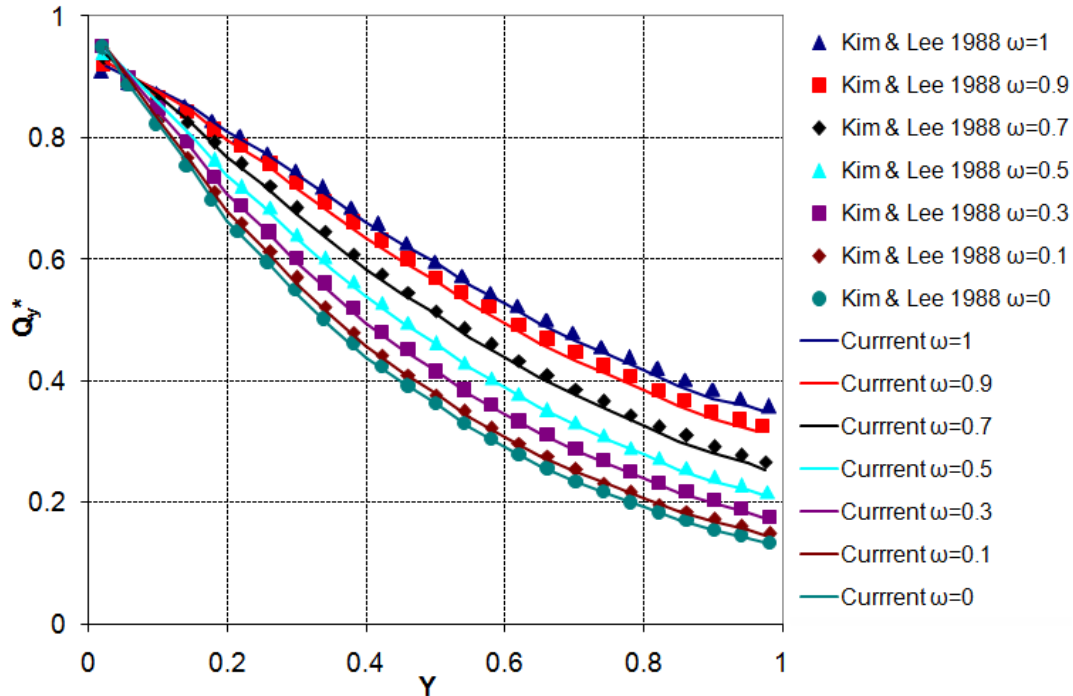


Figure 8.5: Distributions of dimensionless incident radiative heat flux along the A-A line for different values of scattering albedo (hexahedral enclosure with quadratic bases).

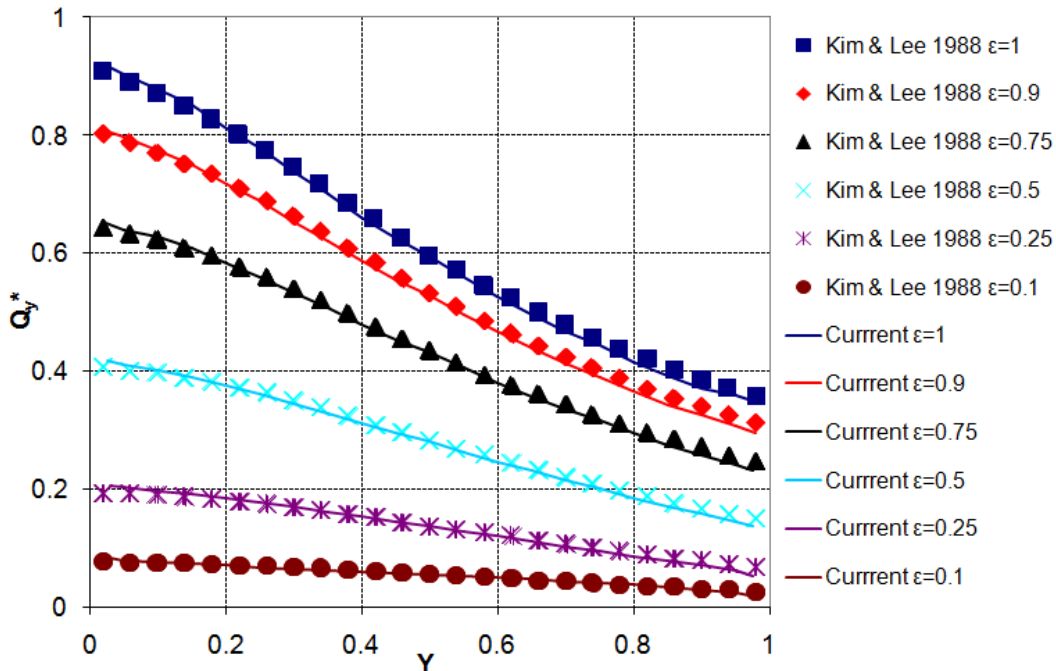


Figure 8.6: Distributions of dimensionless incident radiative heat flux along the A-A line for different values of wall emissivity (hexahedral enclosure with quadratic bases).

8.3. A cubic enclosure

The second radiative heat transfer problem examined in this study concerns a cubic enclosure with edges' length equal to unity, as presented in Figure 8.7 [Kim88, Tri08, Cap10, Lyg13c, Lyg14b, Lyg14g]. Three sub-cases were encountered with this enclosure, the first considering an absorbing and emitting medium, while the second and the third one an isotropically and anisotropically scattering medium respectively.

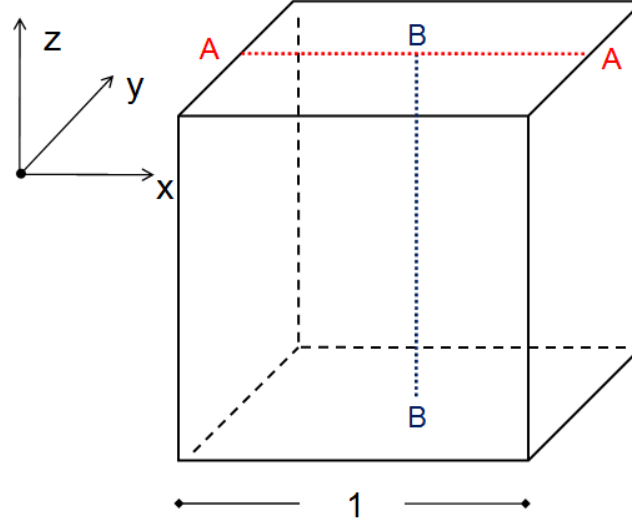


Figure 8.7: Dimensions of the cubic enclosure.

For the first sub-problem, the enclosure walls are assumed radiatively black ($\epsilon_w=1$) and cold ($T_w=0$ K), while the included medium is considered absorbing and emitting ($k_a=1 \text{ m}^{-1}$, $\sigma_s=0 \text{ m}^{-1}$) with a constant heating energy equal to unity ($E=1 \text{ W/m}^3$, corresponding to 64.80435 K). The employed mesh consists of 5,914 nodes, 15,399 tetrahedra and 4,900 prisms, while for parallel computation on a DELL^(R) Laptop with a single Intel^(R) Core^(TM) 2 Duo T6400 two-core processor at 2.00 GHz, it was divided in two sub-domains; the utilized mesh is illustrated in Figure 8.8. For angular discretization eight azimuthal and four polar angles are used along with the pixelation method to allow for the alleviation of the overhang problem, derived from the combination of the unstructured grid and the division of the angular domain in finite solid angles. Radiative intensity at wall surfaces is obtained implicitly. The final steady-state solution of the time-dependent RTE is approximated with the incorporated second-order accurate in time Runge-Kutta method. The parameters of the simulation are summarized in Table 8.2.

Table 8.2: Parameters of simulation (cubic enclosure - first sub-case).

Parameters	
Absorption coefficient	1 m^{-1}
Scattering coefficient	0 m^{-1}
Wall emissivity	1
Grid density	5,914 nodes, 15,399 tetrahedra and 4,900 prisms
Number of partitions	2
Angular resolution	8 azimuthal and 4 polar control angles
Iterative scheme	Runge-Kutta method
Computational system	DELL ^(R) Laptop with a single Intel ^(R) Core ^(TM) 2 Duo T6400 two-core processor at 2.00 GHz

In order to evaluate additionally the contribution of the developed grid adaptation methodology in the improvement of accuracy, the initial mesh was h-refined, employing a criterion targeting the region of the extracted results (upper surface); more specifically the edges with end-points with coordinate z greater than 0.95 m were marked for division. The new mesh is composed of $14,179$ nodes, $16,827$ tetrahedra and $19,600$ prisms; in Figure 8.9 the mesh density on the face of the cube with coordinate y equal to zero is illustrated prior (left) and after (right) h-refinement. Figure 8.10 includes the distributions of dimensionless incident radiative heat flux along the A-A line of the initial mesh for a first-order accurate spatial scheme, a simple second-order one, a second-order jointed with the Van Albada-Van Leer limiter (VAVL) and a second-order with the Min-mod one, compared with the exact distribution, obtained from the study of Capdevilla et al. [Cap10]; in Figure 8.11 the corresponding distributions are presented for the adapted finer grid. The solutions were extracted in approximately twenty four minutes, achieving a seven-order decrease of radiative intensity residual. The superiority of second-order accurate solutions, compared to the first-order one, is clearly observed, while among them this employing the Min-mod limiter seems to be more accurate as the other two appear to lead to overestimations. Furthermore, the contribution of the h-refinement methodology to the improvement of the extracted results is identified, especially in case of the first-order accurate scheme; for the rest of the simulations improvement is marginally noticed near the cold surfaces.

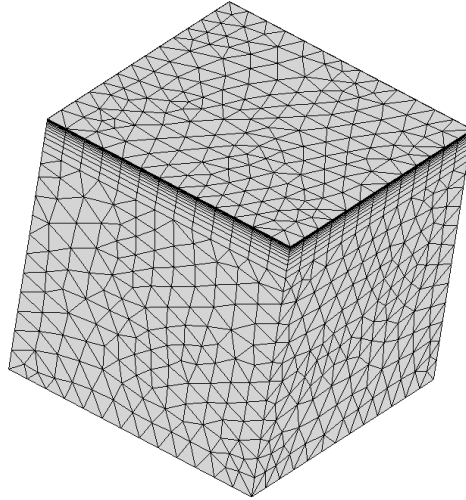


Figure 8.8: Initial mesh representing the cubic enclosure (first sub-case).

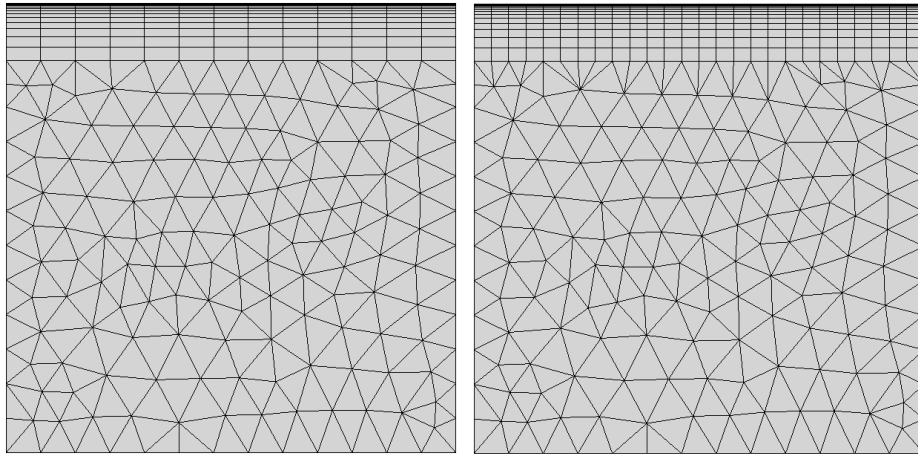


Figure 8.9: Mesh density on the face of the cube with coordinate y equal to zero prior (left) and after (right) h-refinement (cubic enclosure - first sub-case).

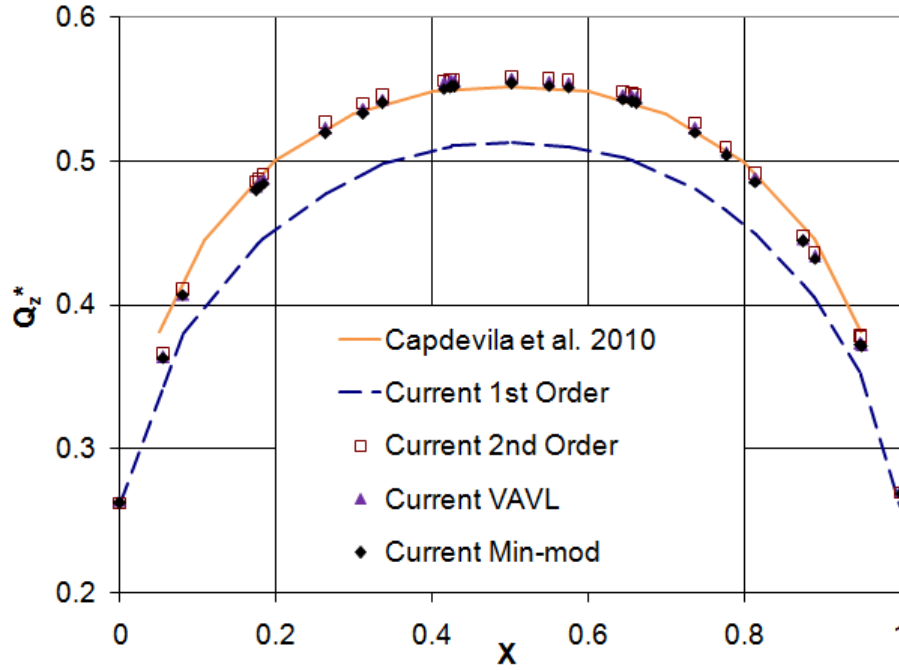


Figure 8.10: Distributions of dimensionless incident radiative heat flux along the A-A line of the initial mesh (cubic enclosure - first sub-case).

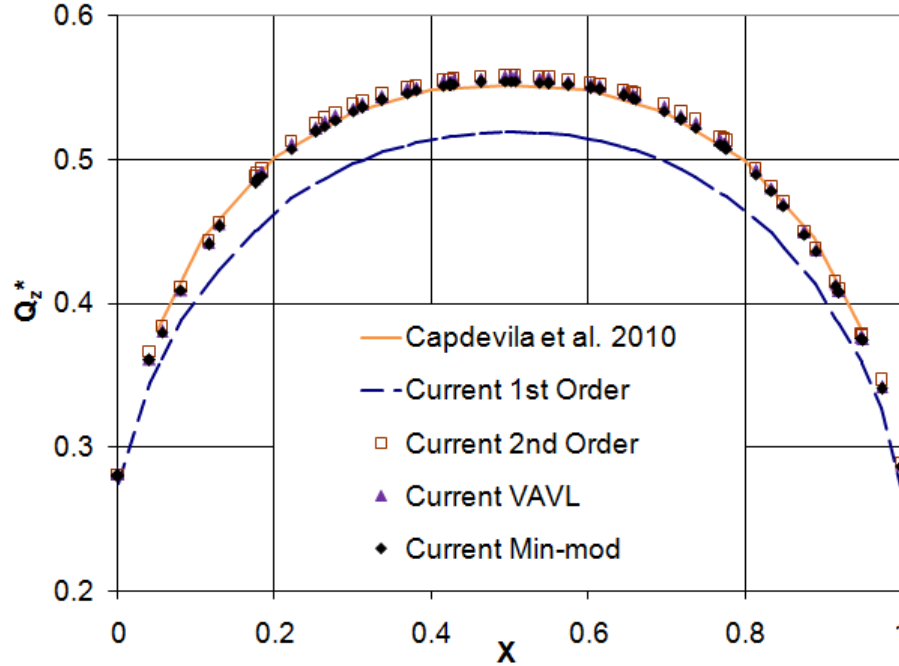


Figure 8.11: Distributions of dimensionless incident radiative heat flux along the A-A line of the refined mesh (cubic enclosure - first sub-case).

In the second sub-case the adapted grid from the previous simulation is employed on the same computational system, including a purely (isotropically) scattering medium ($k_a=0 \text{ m}^{-1}$, $\sigma_s=1 \text{ m}^{-1}$) instead. The radiatively black walls ($\epsilon_w=1$) as well as the medium are assumed cold ($T_w=0 \text{ K}$, $T_m=0 \text{ K}$), except for the lower surface ($z=0 \text{ m}$) at which a constant heating power is imposed ($E=1 \text{ W/m}^2$). The time-dependent RTE is approximated with the Runge-Kutta method, while implicit boundary conditions are implemented.

The angular domain was divided by twenty four azimuthal and twelve polar angles due to the significant ray effect encountered in this sub-case; the pixelation method was applied for the mitigation of the overhang problem. The parameters of the simulation are summarized in the following Table 8.3.

Table 8.3: Parameters of simulation (cubic enclosure - second sub-case).

Parameters	
Absorption coefficient	0 m^{-1}
Scattering coefficient	1 m^{-1}
Wall emissivity	1
Grid density	14,179 nodes, 16,827 tetrahedra and 19,600 prisms
Number of partitions	2
Angular resolution	24 azimuthal and 12 polar control angles
Iterative scheme	Runge-Kutta method
Computational system	DELL ^(R) Laptop with a single Intel ^(R) Core ^(TM) 2 Duo T6400 two-core processor at 2.00 GHz

In Figure 8.12 the obtained distributions of dimensionless incident radiative heat flux along the half A-A line for all the available schemes are depicted, compared with the computed ones by the ZM-MC (Zone Method - Monte Carlo Method) and FV-ME (Finite-Volume method - Mie Equations) methods of Trivic et al. [Tri08]. The results were extracted after the radiative intensity residual was decreased about four orders of magnitude. Although the first-order accurate distribution almost coincides with the corresponding FV-ME one, it overestimates the incident radiation [Tri08]. On the other hand, the solutions obtained by the second-order methods are sufficiently accurate compared with the more accurate ZM-ME one, while no considerable differentiation is identified among them; the improvement in accuracy of the final solution, derived by the second-order accurate schemes, is demonstrated once more.

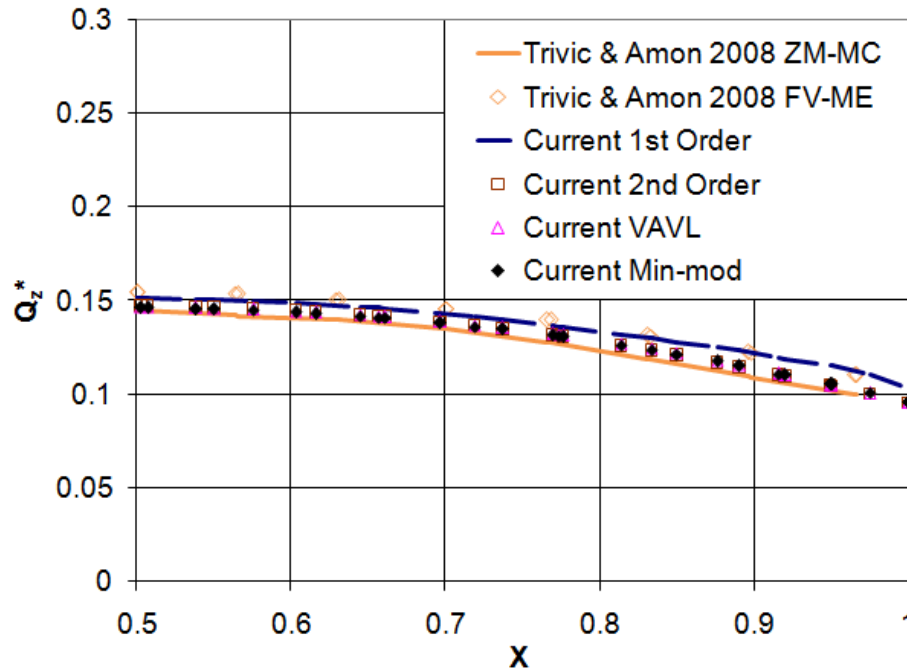
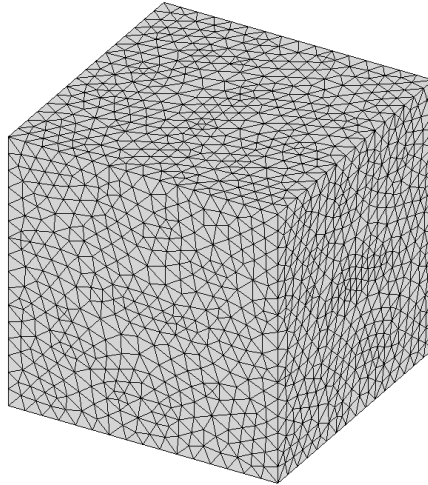


Figure 8.12: Distributions of dimensionless incident radiative heat flux along the half A-A line (cubic enclosure - second sub-case).

Table 8.4: Parameters of simulation (cubic enclosure - third sub-case).

Parameters	
Absorption coefficient	$0-1 \text{ m}^{-1}$
Scattering coefficient	$0-1 \text{ m}^{-1}$
Wall emissivity	$0-1$
Grid density	13,433 nodes and 71,589 tetrahedra
Number of partitions	2
Number of spatial multigrid levels	1-4
Angular resolution	16 azimuthal and 8 polar control angles
Number of angular multigrid levels	1-3
Iterative scheme	Runge-Kutta method
Computational system	Workstation with an AMD FX ^(tm) -8350 eight-core processor at 4.0 GHz

The third (quasi-3D) sub-case considering radiative heat transfer in a cubic enclosure, was encountered mainly for the evaluation of the developed spatial/angular agglomeration multigrid methodology and more specifically of its contribution to the solution's convergence rate for various values of scattering albedo (forward anisotropic, F2, $\omega=0-1$, $\beta=1 \text{ m}^{-1}$) and wall emissivity ($\varepsilon_w=0.1-1$). It is similar to the case of section 8.2, concerning a cold medium ($T_m=0 \text{ K}$) as well as cold and black walls ($T_w=0 \text{ K}$, $\varepsilon_w=1$), except for the bottom face ($z_w=0 \text{ m}$) and the bases normal to x -direction, where a constant heating energy ($E_w=1 \text{ W/m}^2$ corresponding to 64.80435 K) and symmetry boundary conditions are applied respectively; implicit treatment of boundary conditions was employed. The utilized tetrahedral grid, divided into two partitions for parallelized solution of time-dependent RTE with the Runge-Kutta method on a workstation with an AMD FX^(tm)-8350 eight-core processor at 4.0 GHz, consists of 13,433 nodes and 71,589 tetrahedra; it is illustrated in Figure 8.13. For angular discretization sixteen azimuthal and eight polar angles were applied, while the pixelation method subdued the overhang problem. For all the single-grid and multigrid runs a second-order accurate spatial scheme, with the Min-mod slope limiter, was implemented unless otherwise stated. The parameters of the simulation are summarized in Table 8.4.

*Figure 8.13: Initial mesh representing the cubic enclosure (third sub-case).*

As mentioned above, runs with different values of scattering albedo ω and wall emissivity ε_w were performed. The implementation of the developed multigrid method influences only the number of required iterations and wall-clock computation time; the extracted solution remains independent of the

acceleration technique. Figures 8.14 and 8.15 include the finally obtained distributions of dimensionless incident radiative heat flux along the B-B line (centerline of the grid in z -direction, $x=0.5\text{ m}$, $y=0.5\text{ m}$) for various values of scattering albedo and wall emissivity respectively. They are close enough with the corresponding ones obtained by the two-dimensional DOM algorithm of Kim and Lee [Kim88]. Comparing to the similar simulation of section 8.2, another significant conclusion is extracted; equally accurate results were obtained with a much coarser grid. Employing a second-order accurate spatial scheme along with the implicit treatment of boundary conditions as well as the implementation of symmetry ones, decrease the excessive need for very fine spatial resolution.

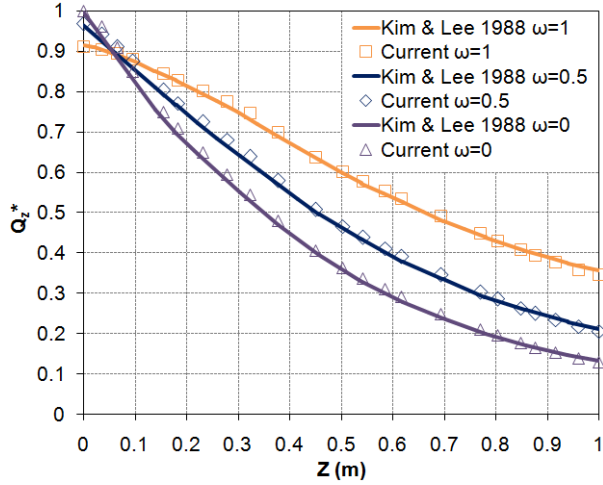


Figure 8.14: Distributions of dimensionless incident radiative heat flux along the B-B line for different values of scattering albedo (cubic enclosure - third sub-case).

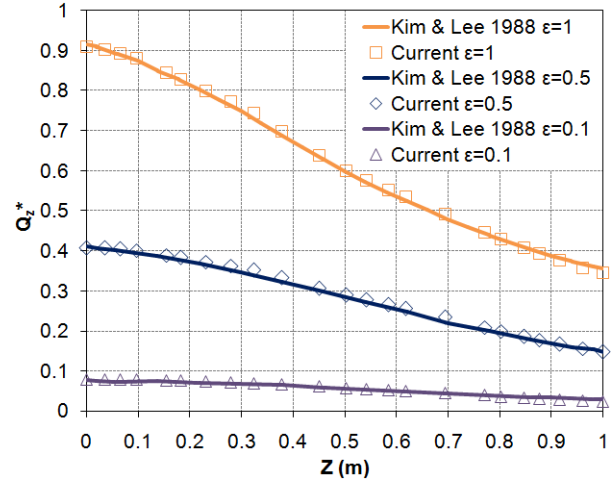


Figure 8.15: Distributions of dimensionless incident radiative heat flux along the B-B line for different values of wall emissivity (cubic enclosure - third sub-case).

Table 8.5: Grid coarsening by agglomeration of two, four and six sub-grids (cubic enclosure - third sub-case).

	Level 1 (DoF's)	Level 2 (DoF's)	Level 3 (DoF's)	Level 4 (DoF's)
2 Partitions				
First Sub-domain	7,363	1,517	403	230
Second Sub-domain	7,316	1,486	414	237
4 Partitions				
First Sub-domain	3,959	878	267	155
Second Sub-domain	3,898	881	268	156
Third Sub-domain	4,009	900	268	154
Fourth Sub-domain	4,033	892	262	151
6 Partitions				
First Sub-domain	2,827	662	211	132
Second Sub-domain	2,775	651	194	99
Third Sub-domain	2,746	643	191	107
Fourth Sub-domain	2,787	652	186	103
Fifth Sub-domain	2,803	647	192	98
Sixth Sub-domain	2,788	656	217	139

Three coarser grids were generated via isotropic agglomeration in order to validate the spatial and the combined spatial/angular agglomeration multigrid scheme. Figure 8.16 illustrates their boundary mesh density; the blue coloured faces correspond to mirroring surfaces, employed to simulate the two-dimensional problem with a three-dimensional grid. Table 8.5 includes the number of DoF's (nodes) per sub-domain of the initial and coarser grids, considering a two-, four- and six-partitions division. The *ghost*

nodes at overlapping regions are also included in this Table; as such, the total number of DoF's differs among different number of sub-grids of the initial mesh. Similarly, angular agglomeration was performed, deriving two coarser angular resolutions, in which the initial discretization of 16 azimuthal and 8 polar angles was reduced to 8×4 (second level) and 4×2 (third level) respectively.

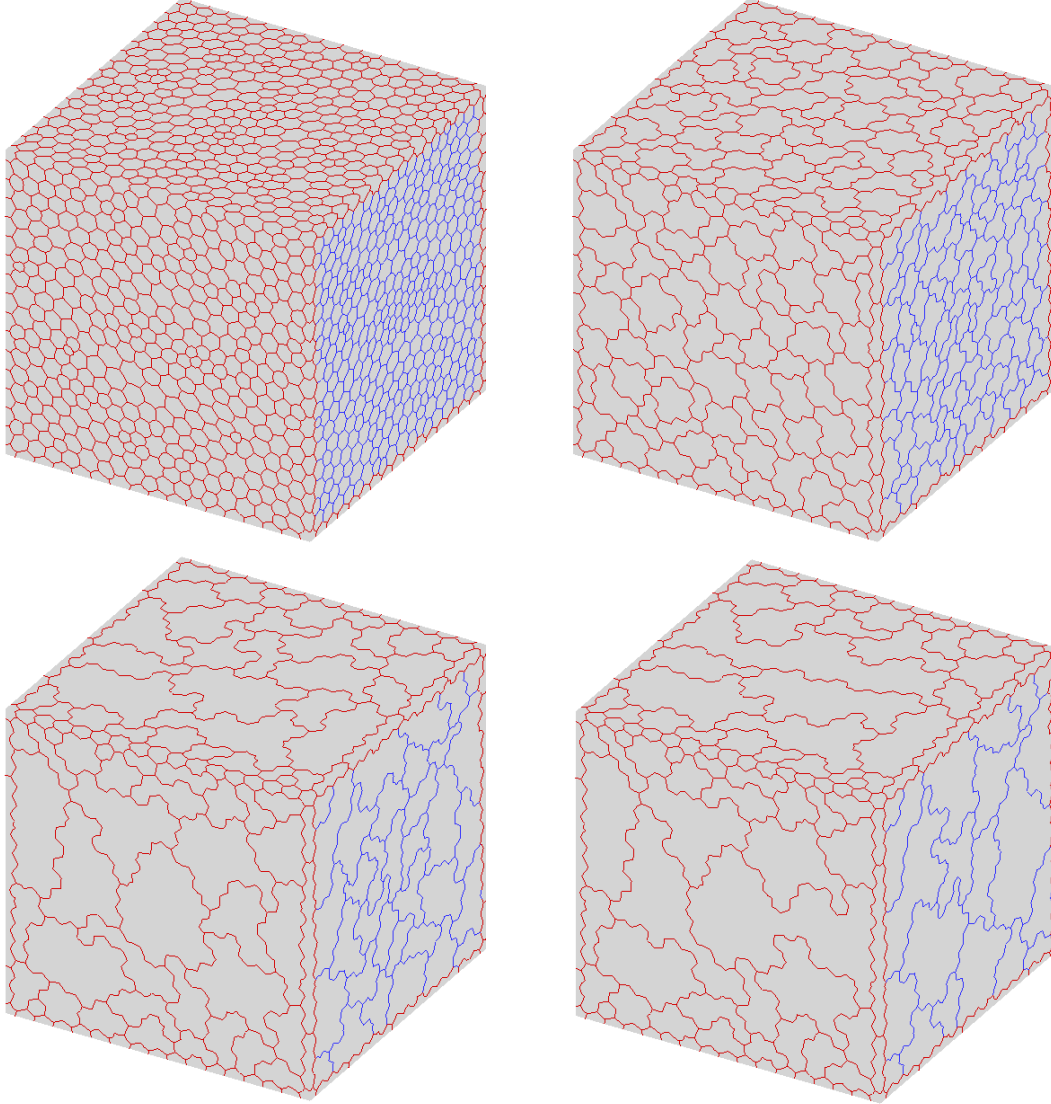


Figure 8.16: Initial and agglomerated control volume grids (cubic enclosure - third sub-case).

The evaluation of the developed multigrid methodology began with the study of an absorbing but no scattering medium ($k_a=1 \text{ m}^{-1}$, $\sigma_s=0 \text{ m}^{-1}$) along with black walls ($\epsilon_w=1$). In Figure 8.17 its convergence history per number of iterations/cycles and wall-clock computation time is illustrated for the single-grid (S1A1) and different combinations of the multigrid schemes (spatial-S4A1, angular-S1A2 and nested spatial/angular-S4A2), while in Table 8.6 the corresponding speed-up coefficients to the final residual value of $1.0E-15$ are included (bold style denotes the higher time speed-up coefficient for each set of runs). Due to the absence of scattering and walls' reflection, the acceleration obtained by the spatial and the nested spatial/angular scheme isn't so significant. The speed-up gained by the angular scheme (compared to the single-grid one), as well as the one derived by the nested scheme (compared to the spatial one), although not expected, is attributed to the back and forth travel of information imposed by the symmetry surfaces and the second-order accurate spatial scheme.

Table 8.6: Speed-up coefficients per number of iterations/cycles and time, considering runs with different values of k_a , σ_s and ε_w (cubic enclosure - third sub-case).

	$k_a (m^{-1})$	$\sigma_s (m^{-1})$	ε_w	Iter. Speed-up	Time Speed-up
S1A2	1	0	1	1.1155	1.075
S4A1	1	0	1	1.6937	1.6433
S4A2	1	0	1	3.5944	1.6765
S1A3	0.5	0.5	1	1.3109	1.2881
S4A1	0.5	0.5	1	2.2051	1.9287
S4A2	0.5	0.5	1	4.5989	1.9917
S1A3	0	1	1	1.4331	1.3213
S4A1	0	1	1	2.8015	2.3625
S4A2	0	1	1	5.8072	2.3719
S1A3	0	1	0.5	2.6473	2.4474
S4A1	0	1	0.5	5.1482	4.2825
S4A2	0	1	0.5	14.2349	4.7256
S1A3	0	1	0.1	2.8813	2.1678
S4A1	0	1	0.1	4.9996	4.1377
S4A3	0	1	0.1	19.2667	7.5545

Another test case, against which the single-grid (S1A1) and various multigrid schemes were applied (S1A3, S4A1 and S4A2), concerned a half absorbing and half scattering ($k_a=0.5 m^{-1}$, $\sigma_s=0.5 m^{-1}$) medium. As illustrated in Figure 8.18 the employed multigrid schemes succeed an increased acceleration, up to a ~ 2.0 temporal speed-up coefficient for the nested version (Table 8.6). The same attitude is identified in case a purely scattering medium ($k_a=0 m^{-1}$, $\sigma_s=1 m^{-1}$) is considered instead; the corresponding convergence history per number of iterations/cycles and computational time is included in Figure 8.19, while Table 8.6 contains the finally derived speed-up coefficients (S1A3, S4A1 and S4A2). Although the gained acceleration by the spatial scheme in the previous cases was quite encouraging, the one achieved by the angular scheme (compared to this of the single-grid one), as well as this of the nested scheme (compared to this of the spatial one) is relatively negligible; the small number of DoF's along with the absence of reflecting walls (excepting the mirroring ones) resulted in a reduced radiation exchange between the defined solid control angles.

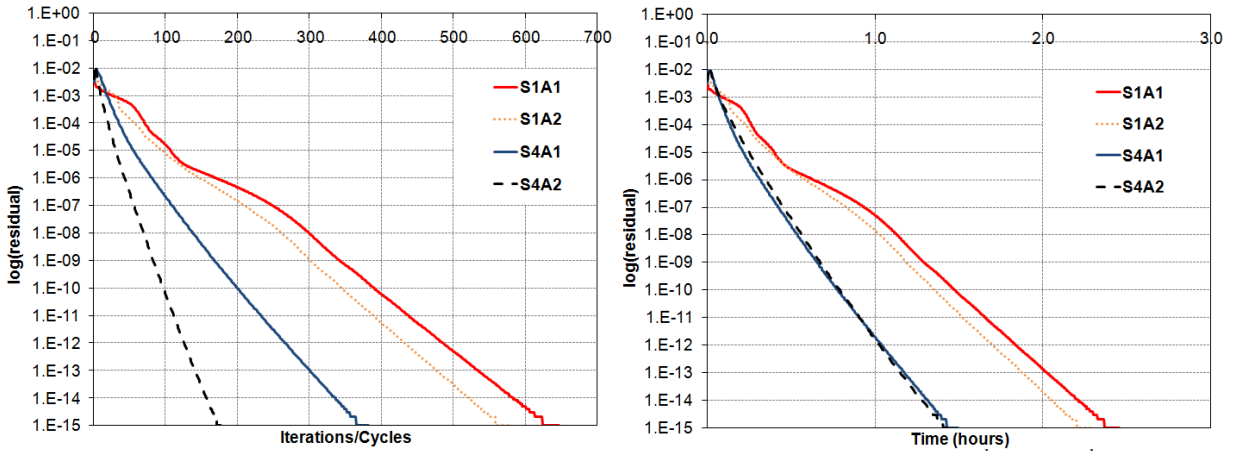


Figure 8.17: Radiative intensity convergence history per iterations/cycles and time ($k_a=1 m^{-1}$, $\sigma_s=0 m^{-1}$, $\varepsilon_w=1$, cubic enclosure - third sub-case).

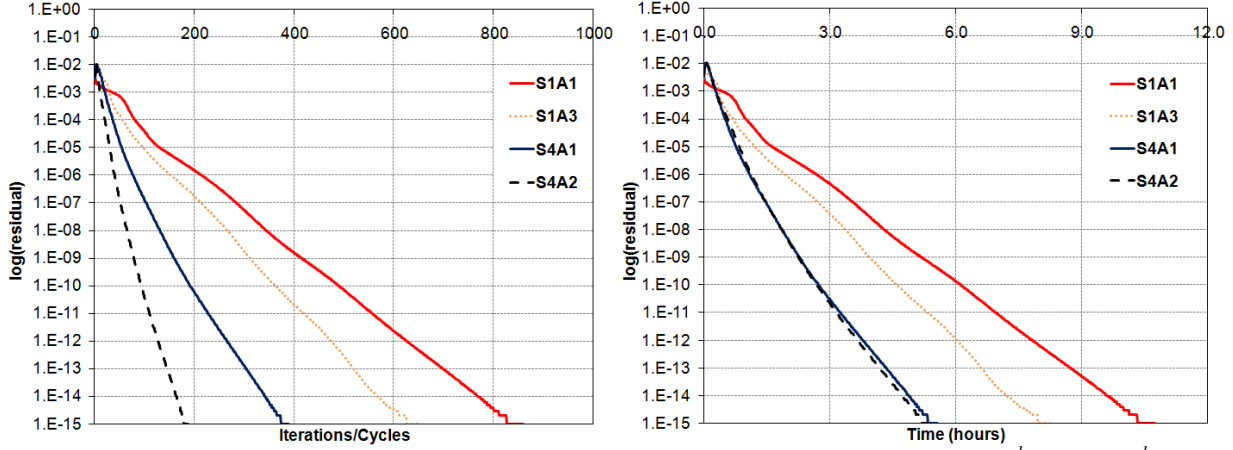


Figure 8.18: Radiative intensity convergence history per iterations/cycles and time ($k_a=0.5 \text{ m}^{-1}$, $\sigma_s=0.5 \text{ m}^{-1}$, $\varepsilon_w=1$, cubic enclosure - third sub-case).

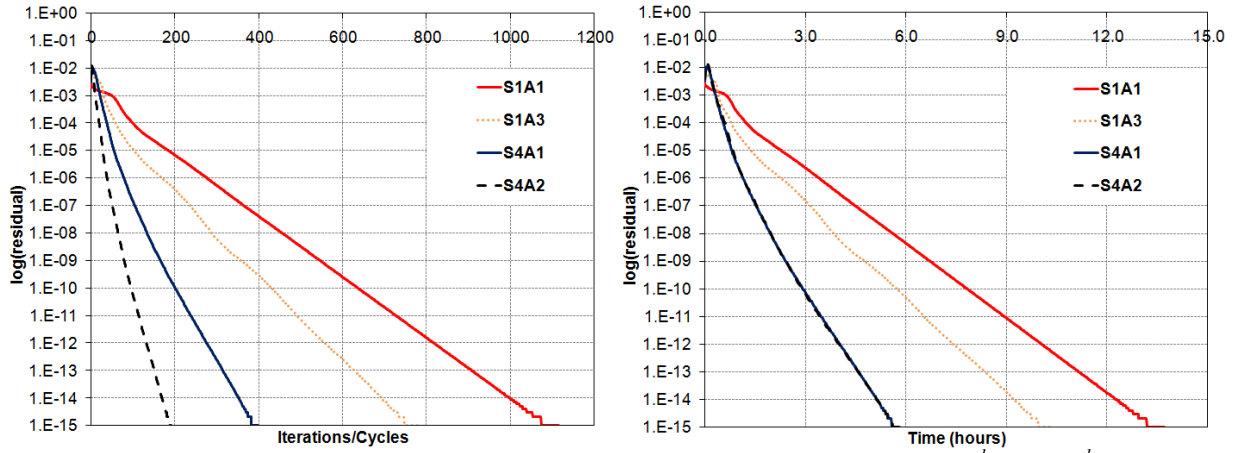


Figure 8.19: Radiative intensity convergence history per iterations/cycles and time ($k_a=0 \text{ m}^{-1}$, $\sigma_s=1 \text{ m}^{-1}$, $\varepsilon_w=1$, cubic enclosure - third sub-case).

Nevertheless, the situation is changing for a purely scattering medium ($k_a=0 \text{ m}^{-1}$, $\sigma_s=1 \text{ m}^{-1}$) along with reflecting walls ($\varepsilon_w=0.5$). According to the results included in Figure 8.20 and Table 8.6, the obtained acceleration coefficients for the final residual $1.0E-15$ increase almost two times for all the multigrid schemes (S1A3, S4A1 and S4A2), compared to the corresponding ones of the previous simulations; the maximum ~ 4.73 speed-up coefficient in time is derived by the nested methodology. The contribution of angular agglomeration (both for the angular and the combined scheme) appears to be quite significant, compared to all of the previous runs (with black boundaries).

In order to study the effect of grid size to the gained acceleration by the multigrid algorithm, a finer mesh was employed, composed of 105,856 nodes and 589,283 tetrahedra. Figure 8.21 contains the convergence history per number of iterations/cycles and wall-clock computation time for the single-grid (S1A1) and all the applied multigrid schemes (S1A2, S4A1 and S4A2) on the finer grid, which was divided in two sub-domains for parallel processing on a DELL workstation T7400 with two Intel Xeon E5410 four-core processors at 2.33 GHz; an acceleration increase is obvious, compared to the corresponding results on the coarser mesh. Table 8.7 contains the speed-up coefficients for residual's value equal to $1.0E-9$ (bold style denotes the higher time speed-up coefficient for each set of runs), obtained by the multigrid schemes on both grids (coarse and fine), confirming the previous state; the S4A2 scheme achieved a temporal speed-up coefficient equal to ~ 4.57 and ~ 6.17 for the coarse and fine grid respectively.

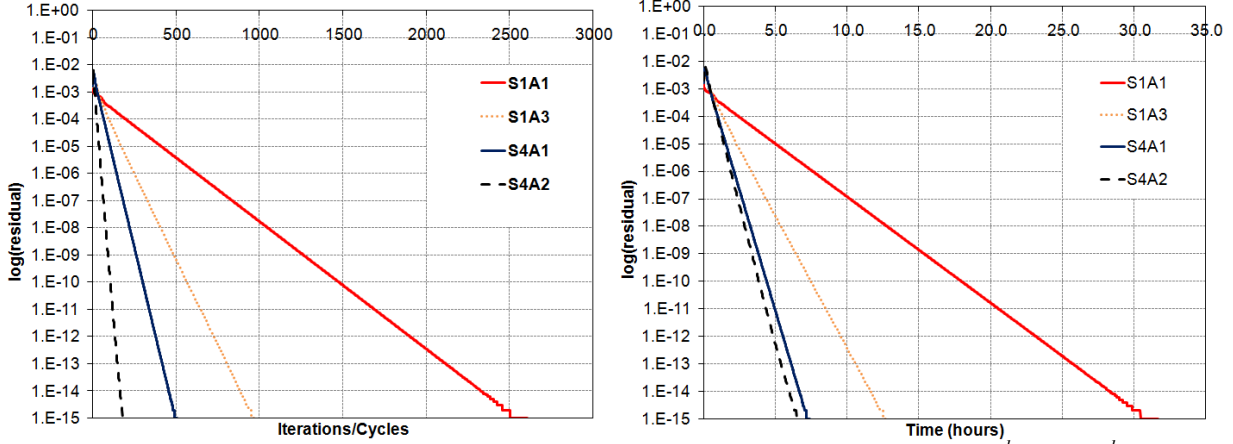


Figure 8.20: Radiative intensity convergence history per iterations/cycles and time ($k_a=0 \text{ m}^{-1}$, $\sigma_s=1 \text{ m}^{-1}$, $\varepsilon_w=0.5$, (cubic enclosure - third sub-case).

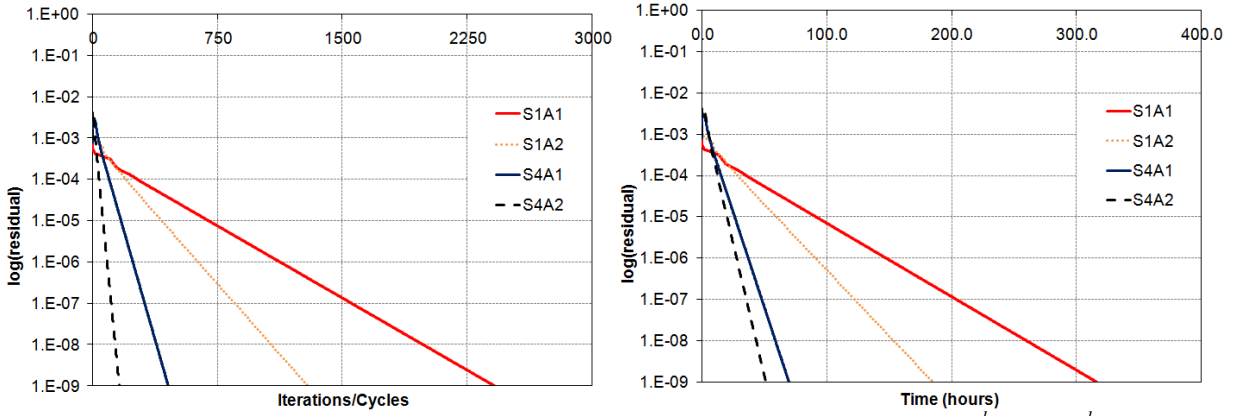


Figure 8.21: Radiative intensity convergence history per iterations/cycles and time ($k_a=0 \text{ m}^{-1}$, $\sigma_s=1 \text{ m}^{-1}$, $\varepsilon_w=0.5$, fine grid, cubic enclosure - third sub-case).

Table 8.7: Speed-up coefficients per number of iterations/cycles and time, considering runs on different spatial resolutions ($k_a=0 \text{ m}^{-1}$, $\sigma_s=1 \text{ m}^{-1}$, $\varepsilon_w=0.5$, cubic enclosure - third sub-case).

	Number of DoF's	Iter. Speed-up	Time Speed-up
S1A3	13,433	2.5975	2.3931
S4A1	13,433	4.8282	4.0063
S4A2	13,433	13.7500	4.5659
S1A2	105,856	1.8699	1.7073
S4A1	105,856	5.3334	4.5336
S4A2	105,856	15.2911	6.1690

The contribution of the developed methodology is revealed even more valuable, in case a purely scattering medium ($k_a=0 \text{ m}^{-1}$, $\sigma_s=1 \text{ m}^{-1}$) along with almost fully reflecting walls ($\varepsilon_w=0.1$) is assumed. The single-grid (S1A1) and various multigrid schemes were employed (S1A3, S4A1 and S4A3), while the convergence rates per iterations/cycles and time are illustrated in Figure 8.22. The increased radiation exchange between different solid control angles, generated by the combination of the purely scattering behaviour of the medium, the almost fully reflecting walls and the imposed symmetry boundary conditions, resulted in higher acceleration rates (Table 8.6); the maximum temporal coefficient ~ 7.55 was succeeded by the nested four-level spatial/three-level angular scheme (S4A3). The same simulation was performed employing a first-order accurate spatial scheme instead, in order to assess the effect of spatial

accuracy (used for the computation of fluxes) to the gained multigrid acceleration; Figure 8.23 and Table 8.8 (bold style denotes the higher time speed-up coefficient for each set of runs) contain the corresponding results. Although the second-order scheme is implemented only to the initial non-agglomerated level, it causes a non-trivial delay (depending of course on the examined case), compared to the fully first-order one; the first-order multigrid solutions were obtained in less computation time but with slight differentiations in acceleration compared to the second-order ones.

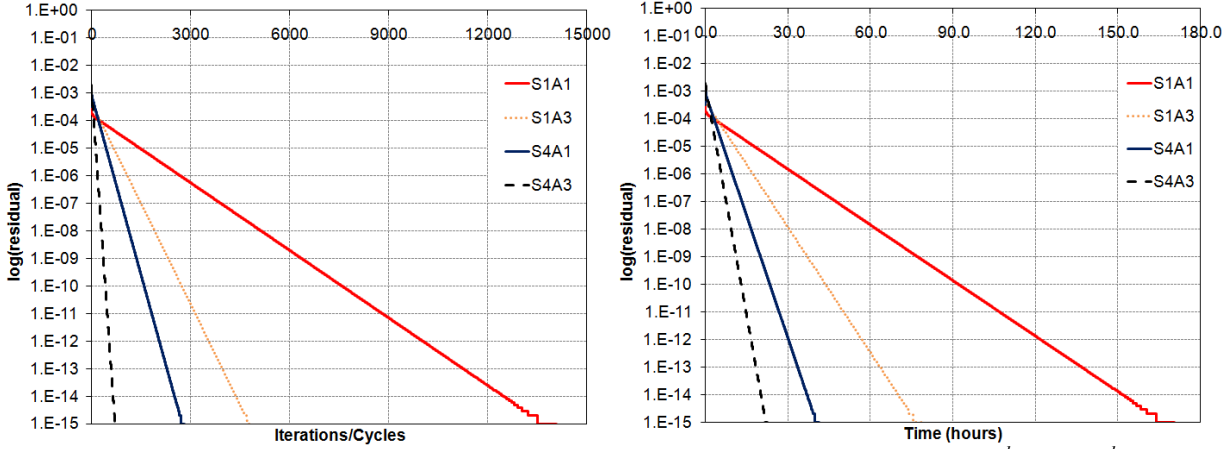


Figure 8.22: Radiative intensity convergence history per iterations/cycles and time ($k_a=0 \text{ m}^{-1}$, $\sigma_s=1 \text{ m}^{-1}$, $\varepsilon_w=0.1$, cubic enclosure - third sub-case).

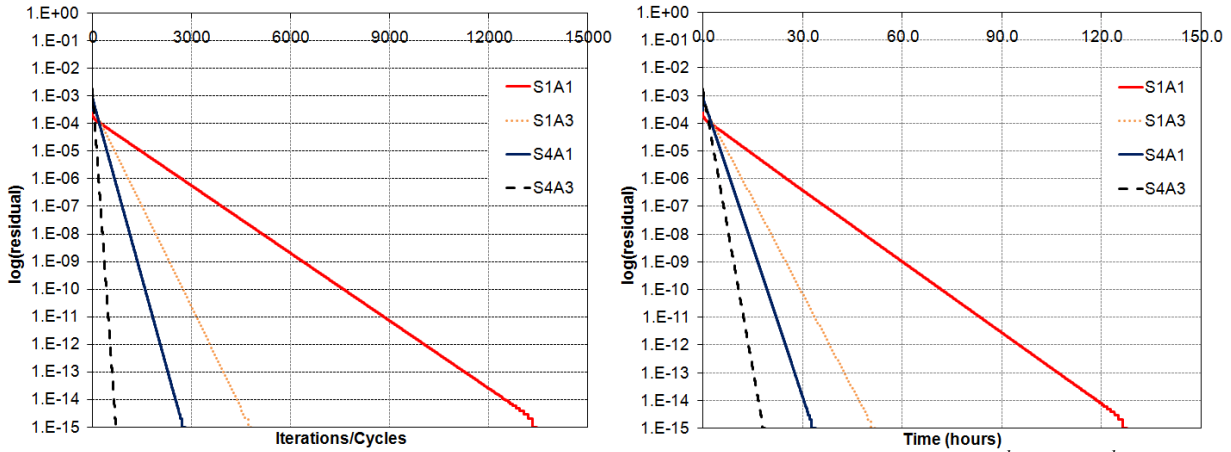


Figure 8.23: Radiative intensity convergence history per iterations/cycles and time ($k_a=0 \text{ m}^{-1}$, $\sigma_s=1 \text{ m}^{-1}$, $\varepsilon_w=0.1$, first-order accurate scheme, cubic enclosure - third sub-case).

In order to further evaluate the contribution of the developed multigrid methodology in accelerating the numerical solution of RTE, three additional runs were performed with the same medium and walls features ($k_a=0 \text{ m}^{-1}$, $\sigma_s=1 \text{ m}^{-1}$, $\varepsilon_w=0.1$), but with different multigrid schemes, namely S1A2, S2A1 and S3A1. Figure 8.24 illustrates the convergence history per iterations/cycles and computation time, derived by the single-grid and three spatial agglomeration multigrid schemes, whose differentiation regards the number of employed multigrid levels (S2A1, S3A1 and S4A1); the distributions, along with the corresponding acceleration coefficients (Table 8.8), confirm that additional acceleration is gained for any additional spatial multigrid level. Similarly, in Figure 8.25 and Table 8.8 the convergence rate of the single-grid run is compared to those of the angular agglomeration multigrid ones, employing two different numbers of successive angular levels (S1A2 and S1A3); the contribution of each extra angular level is revealed as the three-level simulation led to a faster solution than the two-level one. Furthermore, considering that the acceleration derived by the three-level angular multigrid scheme is almost equal to

the one obtained by the three-level spatial one, the potential of the first for increased computational performance is reinforced. Finally, another nested scheme (S4A2) with only two angular levels was employed, to examine whether the increase of the angular discretizations in the combined scheme causes a delay to the solution; the results in Table 8.8 confirm the superiority of S4A3 scheme comparing to S4A2 one.

Table 8.8: Speed-up coefficients per number of iterations/cycles and time, considering low and high-order accurate spatial schemes as well as different numbers of sub-grids ($k_a=0 \text{ m}^{-1}$, $\sigma_s=1 \text{ m}^{-1}$, $\varepsilon_w=0.1$, cubic enclosure - third sub-case).

	Number of Partitions	Order Scheme	Iter. Speed-up	Time Speed-up
S1A3	2	1st	2.7508	2.4294
S4A1	2	1st	4.7696	3.7588
S4A3	2	1st	18.4890	6.8036
S1A2	2	2nd-Minmod	1.9523	1.51745
S1A3	2	2nd-Minmod	2.8813	2.1678
S2A1	2	2nd-Minmod	1.8231	1.3326
S3A1	2	2nd-Minmod	3.1514	2.1991
S4A1	2	2nd-Minmod	4.9996	4.1377
S4A2	2	2nd-Minmod	14.6555	5.8690
S4A3	2	2nd-Minmod	19.2667	7.5545
S1A3	4	2nd-Minmod	2.8813	2.6856
S4A1	4	2nd-Minmod	4.9158	4.0878
S4A3	4	2nd-Minmod	18.9301	7.4375
S1A3	6	2nd-Minmod	2.8811	3.2992
S4A1	6	2nd-Minmod	4.9224	4.1117
S4A3	6	2nd-Minmod	18.9542	8.9062

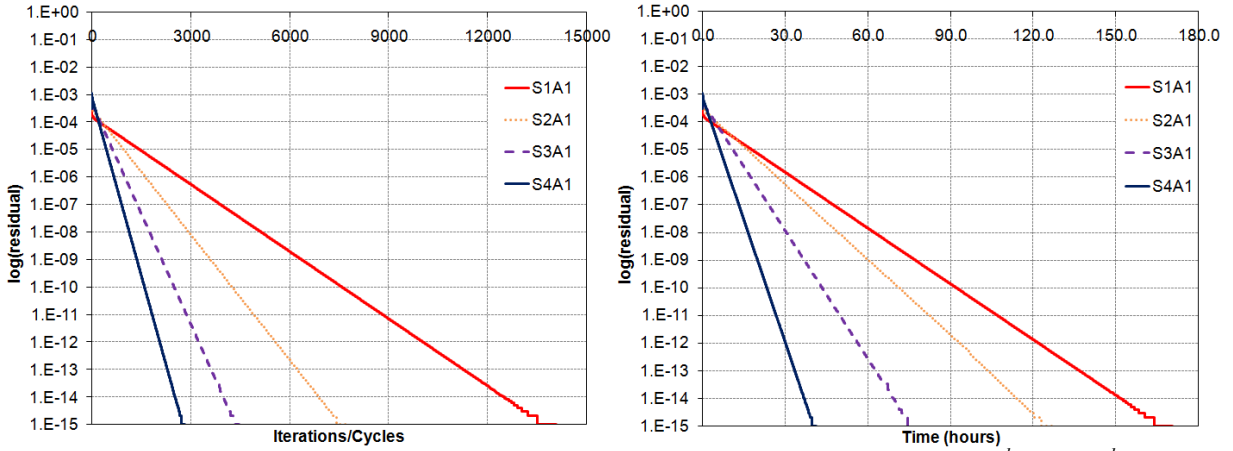


Figure 8.24: Radiative intensity convergence history per iterations/cycles and time ($k_a=0 \text{ m}^{-1}$, $\sigma_s=1 \text{ m}^{-1}$, $\varepsilon_w=0.1$, various spatial multigrid levels, cubic enclosure - third sub-case).

The effect of the number of sub-grids, dividing the initial mesh for parallel processing, was examined with the last simulations on the cubic enclosure. Thus, the initial cubic mesh was divided in four and six partitions respectively, for each of which three coarser sub-grids were generated via isotropic agglomeration; Figure 8.26 illustrates their boundary mesh density, while Table 8.5 contains the corresponding number of DoF's (nodes) for each sub-domain of the initial and the coarser meshes. As the *ghost* nodes at overlapping regions are also included in this Table, the number of total nodes differs among different partitionings of the initial grid. The convergence histories and coefficients per number of

iterations/cycles and time of the simulations with four sub-domains (S1A1, S1A3, S4A1 and S4A3) are presented in Figure 8.27 and Table 8.8 respectively; although the total wall-clock computation time is reduced comparing to this obtained with two partitions due to utilization of two more processors, no significant difference is identified as far as the convergence rates are concerned. However, the corresponding numerical solutions with six sub-domains appear to provide increased temporal speed-up coefficients (Figure 8.28 and Table 8.8), especially in case of the angular (~ 3.3) and the nested spatial/angular scheme (~ 8.9). This acceleration derives probably from the decreased number of transferred variables between the neighbouring sub-grids, due to angular coarsening (for the second angular level only $N_{p_{ghost}} \times (N_\theta/2) \times (N_\phi/2)$ variables are exchanged) as well as due to the less DoF's at the coarsest mesh (fourth multigrid level) in case of six sub-domains compared to those at the coarsest grids in case of two or four partitions (Table 8.5); a more extended agglomeration is achieved, contributing consequently to an elevated acceleration for the same computational system.

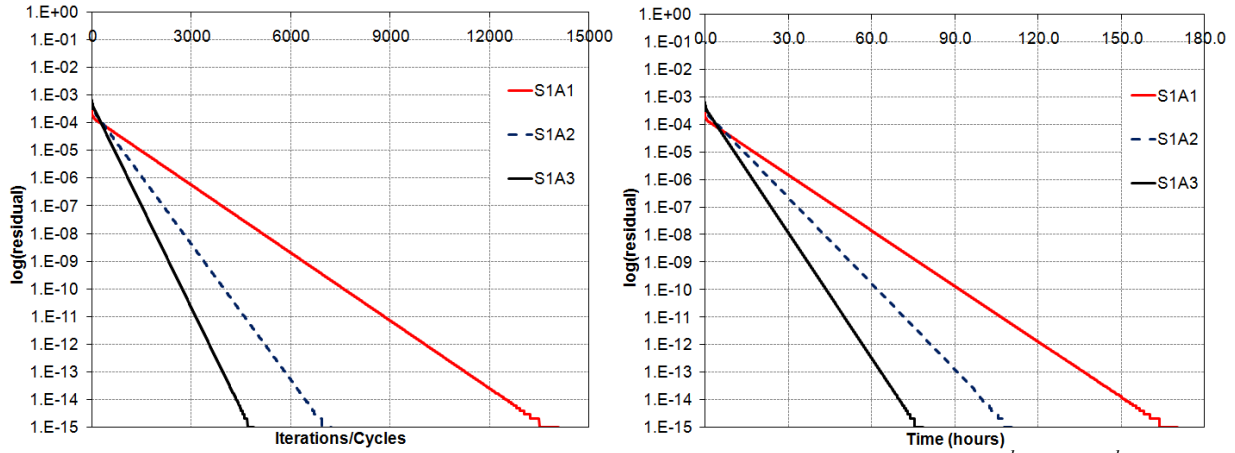


Figure 8.25: Radiative intensity convergence history per iterations/cycles and time ($k_a=0 \text{ m}^{-1}$, $\sigma_s=1 \text{ m}^{-1}$, $\varepsilon_w=0.1$, various angular multigrid levels, cubic enclosure - third sub-case).

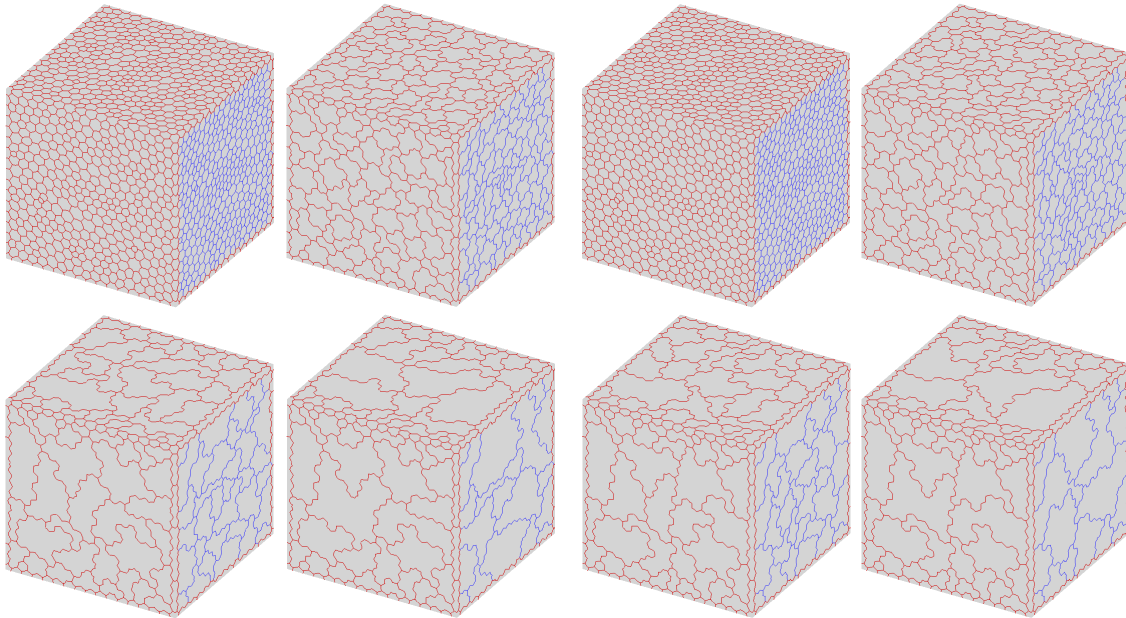


Figure 8.26: Initial and coarser control volume grids, divided in four (left) and six (right) partitions (cubic enclosure - third sub-case).

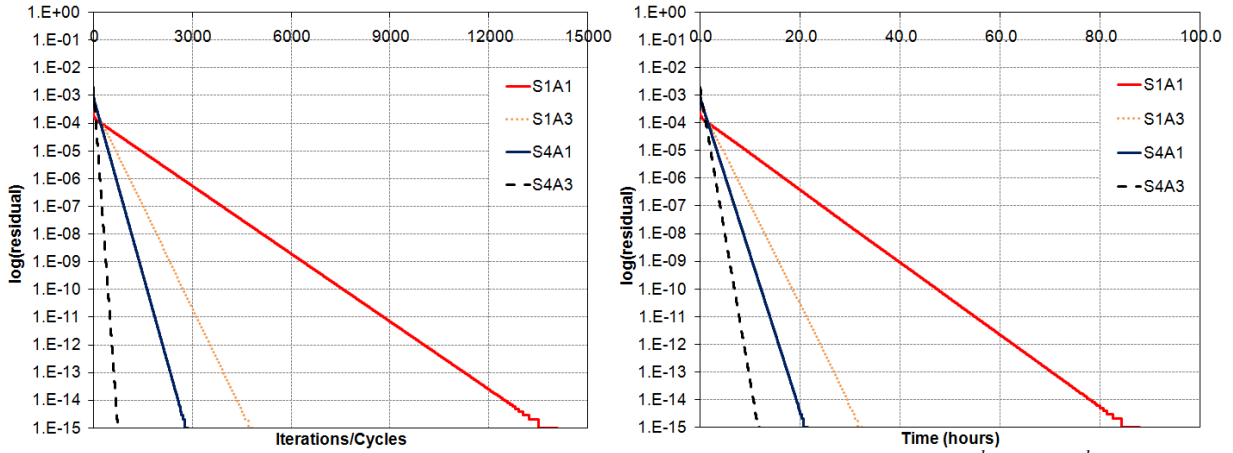


Figure 8.27: Radiative intensity convergence history per iterations/cycles and time ($k_a=0 \text{ m}^{-1}$, $\sigma_s=1 \text{ m}^{-1}$, $\varepsilon_w=0.1$, four partitions, cubic enclosure - third sub-case).

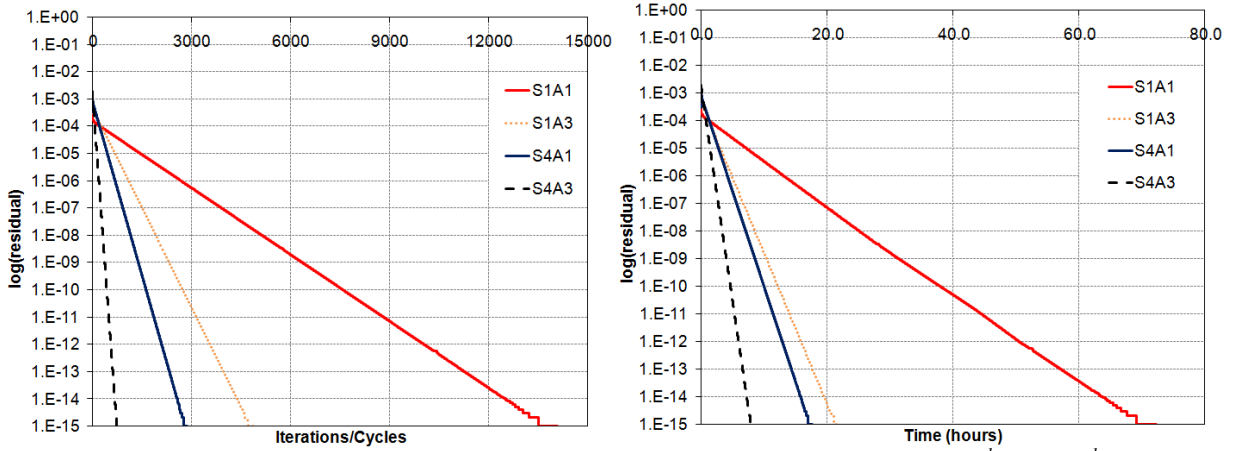


Figure 8.28: Radiative intensity convergence history per iterations/cycles and time ($k_a=0 \text{ m}^{-1}$, $\sigma_s=1 \text{ m}^{-1}$, $\varepsilon_w=0.1$, six partitions, cubic enclosure - third sub-case).

8.4. A prismatic enclosure

The third test case concerns radiative heat transfer in a prismatic enclosure with equilateral triangular bases; its dimensions are illustrated in Figure 8.29 [Kim01, Gri10, Lyg12a, Lyg13c]. Two sub-cases were simulated for this enclosure, employing the time-dependent RTE along with the Runge-Kutta method on a DELL T7400 workstation with two Intel^(R) Xeon^(R)-E5410 four-core processors at 2.33 GHz.

For the first (quasi-3D) sub-case, the walls are considered radiatively black ($\varepsilon_w=1$) and cold ($T_w=0 \text{ K}$), except for the two equilateral triangular bases assumed to behave as symmetry planes; all types of boundary conditions were imposed in an implicit way. The included medium is considered absorbing and emitting, but no scattering ($\sigma_s=0 \text{ m}^{-1}$), maintained at constant temperature ($T_m=1000 \text{ K}$); an absorption coefficient equal to unity ($k_a=1 \text{ m}^{-1}$) was selected. For spatial discretization a mesh, composed of 10,989 nodes, 8,159 tetrahedra, 16,658 prisms and 15 pyramids, was employed, divided in two partitions for parallel processing (Figure 8.30). Two different angular resolutions were defined, resulting in the decomposition of the angular sphere in eight azimuthal - four polar angles and in sixteen azimuthal - four polar angles respectively. Pixelation method was selected instead of bold approximation for the mitigation of the overhang problem. The parameters of the simulation are included in Table 8.9.

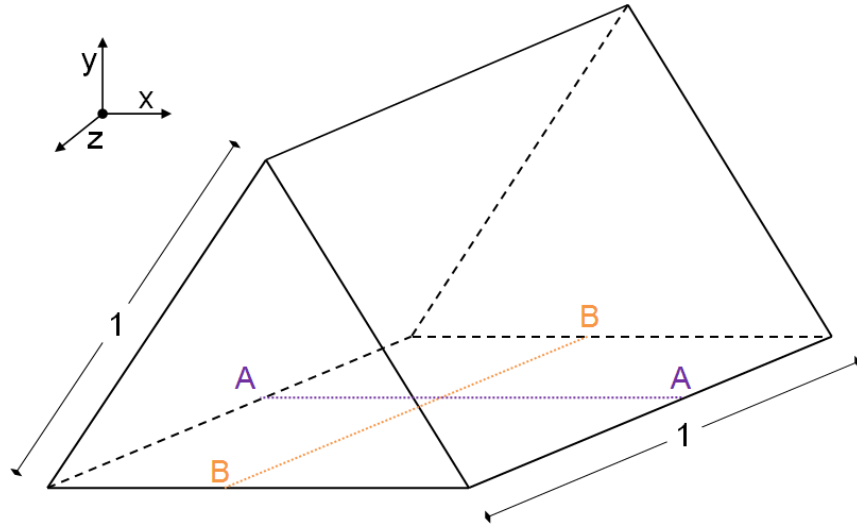


Figure 8.29: Dimensions of the prismatic enclosure.

Table 8.9: Parameters of simulation (prismatic enclosure - first sub-case).

Parameters	
Absorption coefficient	1 m^{-1}
Scattering coefficient	0 m^{-1}
Wall emissivity	1
Grid density	10,989 nodes, 8,159 tetrahedra, 16,658 prisms and 15 pyramids
Number of partitions	2
Angular resolution	8 azimuthal - 4 polar and 16 azimuthal - 4 polar control angles
Iterative scheme	Runge-Kutta method
Computational system	DELL T7400 workstation with two Intel ^(R) Xeon ^(R) -E5410 four-core processors at 2.33 GHz

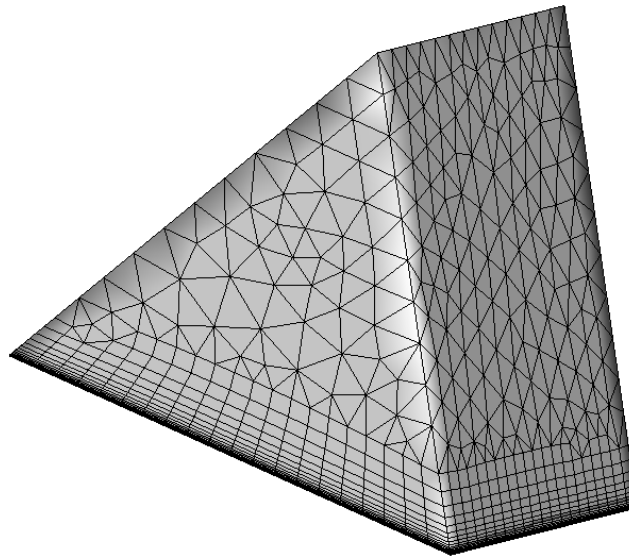


Figure 8.30: Employed grid (prismatic enclosure - first sub-case).

Additionally, in order to evaluate the contribution of the developed h-refinement methodology in accuracy improvement of the final steady-state solution, the initial grid was adapted employing a criterion, which targeted the nodes at the region of the bottom face; more specifically, the edges with endpoints with coordinate y less than 0.05 m were marked. The derived adapted mesh consisted of 38,285 nodes, 11,522 tetrahedra, 66,976 prisms and 154 pyramids. Figure 8.31 illustrates the mesh density on the equilateral triangular base with coordinate z equal to zero ($z=0\text{ m}$) prior (left) and after (right) h-refinement; the division of the lower elements (prismatic and tetrahedral) into smaller ones can be identified.

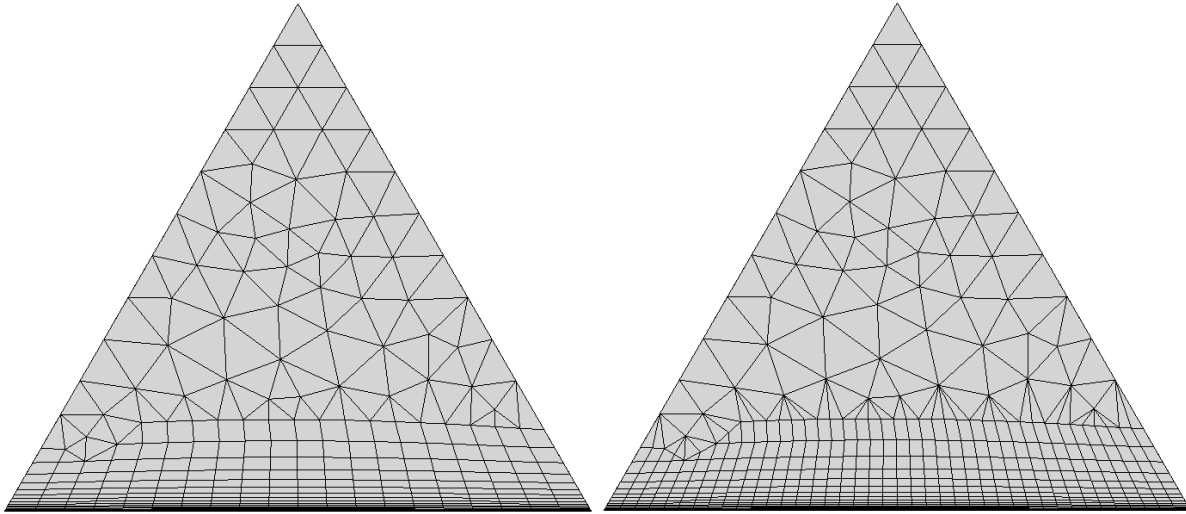


Figure 8.31: Mesh density on an equilateral triangular base ($z=0$) prior (left) and after (right) h-refinement (prismatic enclosure - first sub-case).

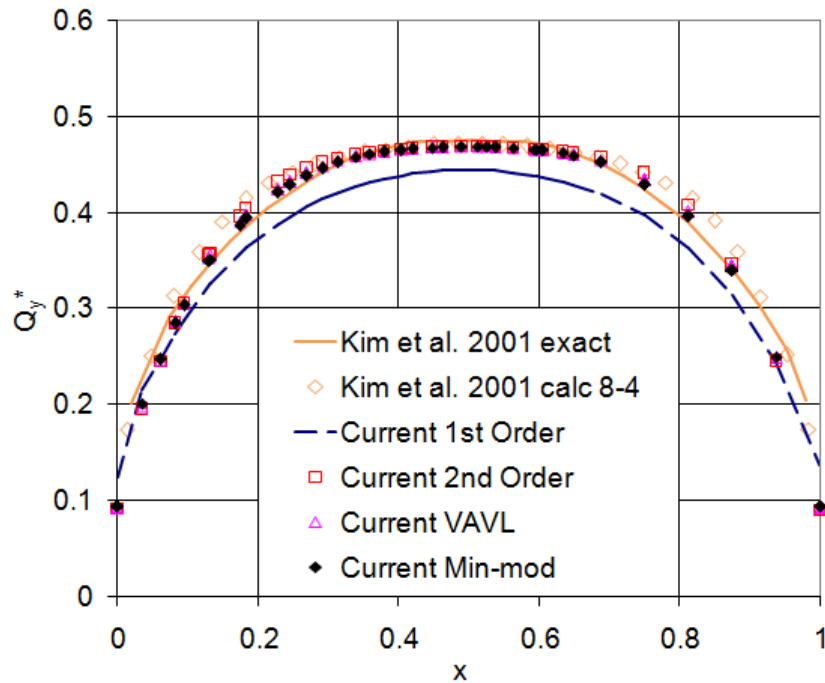


Figure 8.32: Distributions of dimensionless incident radiative heat flux along the A-A line (initial grid, 8-4 angular resolution, prismatic enclosure - first sub-case).

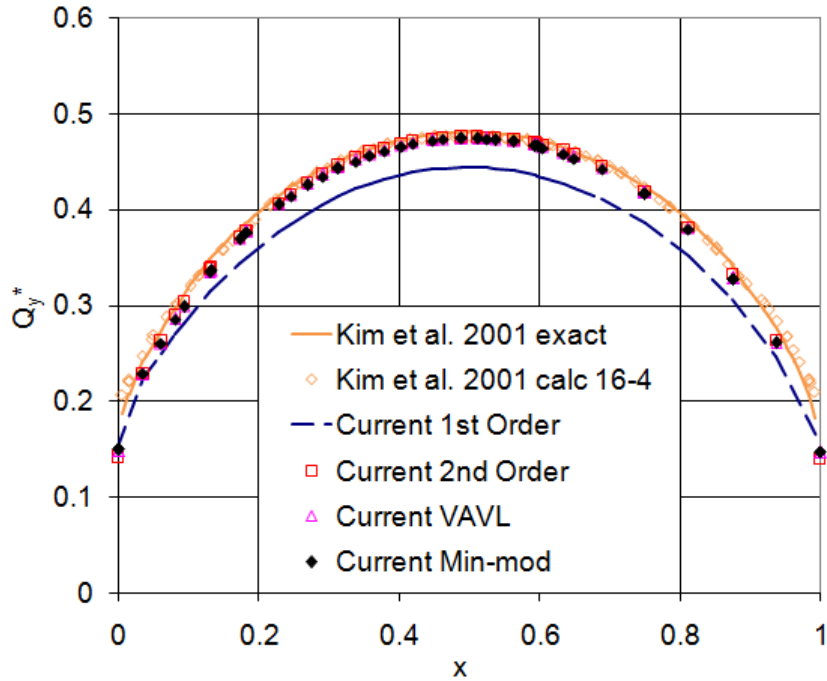


Figure 8.33: Distributions of dimensionless incident radiative heat flux along the A-A line (initial grid, 16-4 angular resolution, prismatic enclosure - first sub-case).

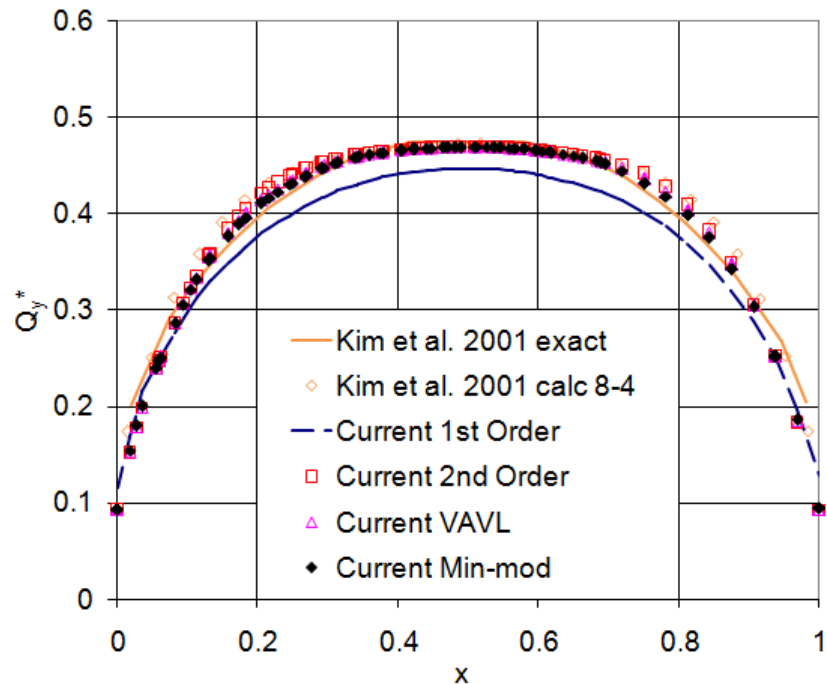


Figure 8.34: Distributions of dimensionless incident radiative heat flux along the A-A line (adapted grid, 8-4 angular resolution, prismatic enclosure - first sub-case).

Figures 8.32 - 8.35 include the distributions of dimensionless incident radiative heat flux along the A-A line of the initial (8.32, 8.33) and the adapted grid (8.34, 8.35), derived by the 8-4 (left) and 16-4 (right) angular discretization as well as by all the available spatial schemes, compared with the exact and the computed ones of Man Young Kim et al. [Kim01]. As expected the denser angular resolution (16-4) produced more accurate results; nevertheless, implementing the incorporated higher-order accurate spatial

schemes reduced considerably this differentiation. In general, the second-order accurate schemes derived better results than the first-order one, while among them this accompanied by the Min-mod limiter appears to approximate more precisely the exact solution. Although the accuracy improvement gained by the h-refinement method is more obvious in the solution considering an 8-4 angular resolution and a first-order accurate spatial scheme, it contributed more or less to all the employed schemes. The aforementioned solutions were obtained after radiative intensity residual was decreased at least six orders of magnitude, while they required approximately two seconds per iteration for the first-order scheme and five seconds per iteration for the second-order ones, considering the initial mesh and the denser angular discretization. In case of the coarser angular resolution the temporal difference per iteration between the first- and the second-order scheme is decreased. Although the first-order scheme appears to be twice faster of the second-order ones, the radiative intensity convergence history per number of iterations, illustrated in Figure 8.36, reveals that the latter ones succeeded a greater convergence rate, as well as a more accurate solution.

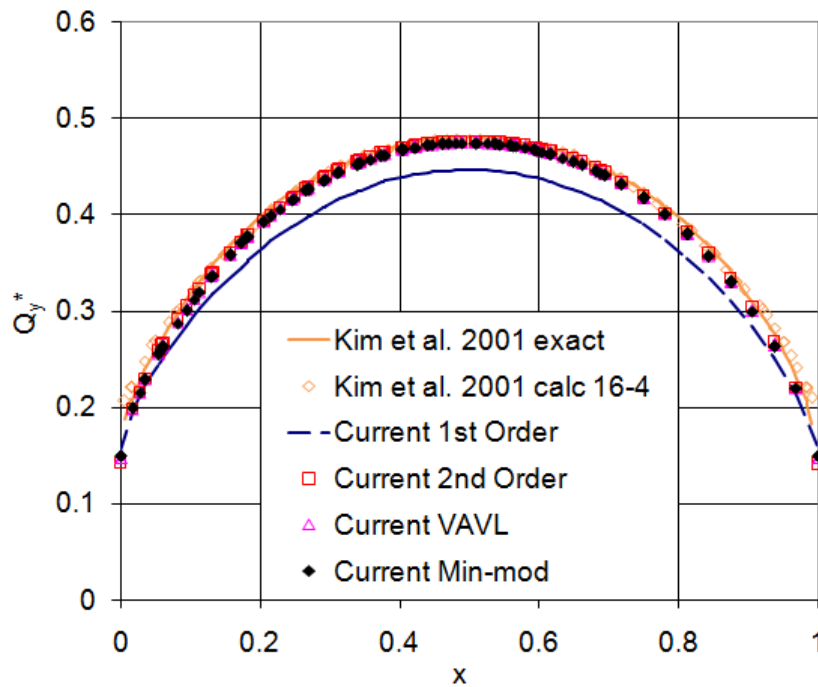


Figure 8.35: Distributions of dimensionless incident radiative heat flux along the A-A line (adapted grid, 16-4 angular resolution, prismatic enclosure - first sub-case).

In order to assure that the derived differentiation between the previously encountered simulations, originates indeed from the upgrading of the spatial resolution with the implementation of the developed higher-order accurate spatial schemes, a very fine mesh was utilized, consisting of 225,151 nodes, 665,561 tetrahedra and 207,780 prisms. In Figure 8.37 this grid as well as the density on one of its equilateral triangular bases is illustrated.

The first- and the second-order accurate scheme coupled with the Min-mod limiter were implemented, along with both 8-4 and 16-4 angular resolutions. In Figures 8.38 and 8.39 the distributions of dimensionless incident radiative heat flux along the A-A line, derived by the aforementioned spatial accurate schemes and angular discretizations are presented, compared with the exact and the computed ones of Man Young Kim et al. [Kim01]. As expected, the differentiation between the results of the employed schemes is reduced considerably, especially for the 16-4 angular resolution, confirming in that way the enhancement of the proposed algorithm by the higher-order accurate spatial schemes.

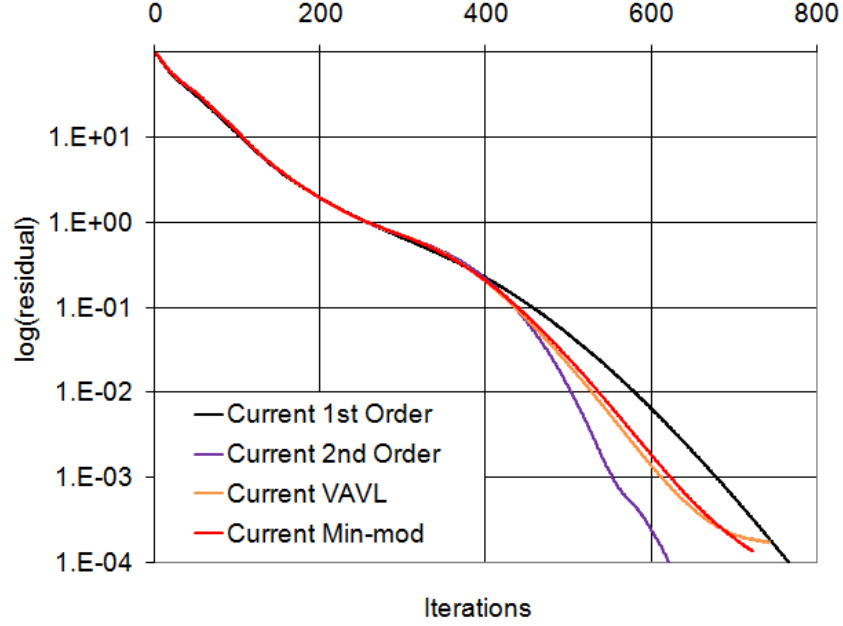


Figure 8.36: Radiative intensity convergence history per iterations (initial grid, 16-4 angular resolution, prismatic enclosure - first sub-case).

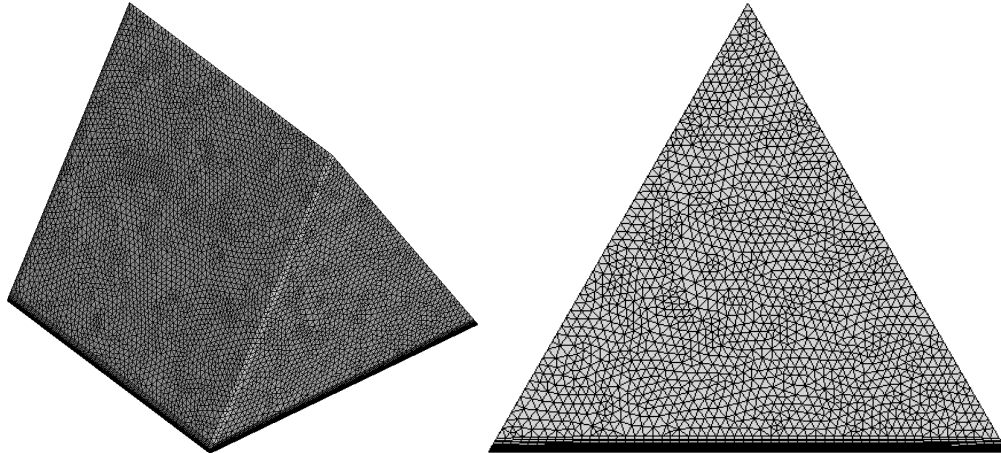


Figure 8.37: Employed grid and its density on one of its equilateral triangular bases (prismatic enclosure - first sub-case).

The second three-dimensional sub-case concerns an absorbing, but no scattering ($\sigma_s=0 \text{ m}^{-1}$), medium assigned with an emissive radiative power of unity ($E=1 \text{ W/m}^3$), which corresponds to 64.80435 K ; three different absorption coefficients k_a were employed, namely 0.1 , 1.0 and 10.0 m^{-1} . All the walls are assumed black ($\varepsilon_w=1$) and cold ($T_w=0 \text{ K}$), while the respective boundary conditions are imposed implicitly. The utilized mesh consists of 29,170 nodes, 81,217 tetrahedra and 26,100 prisms, while for parallel computation it was decomposed in four partitions (Figure 8.40). Angular discretization is achieved using 8 azimuthal and 8 polar angles, while pixelation method alleviates any arisen overhang problem. The parameters of this simulation are summarized in Table 8.10.

This sub-case is focused on the evaluation of the accuracy improvement, obtained by the incorporated h-refinement methodology. Therefore, the initial mesh was locally refined, employing the same to the previous sub-case criterion (bottom face region); the generated grid is composed of 78,822 nodes, 128,604 tetrahedra and 104,400 prisms. In Figure 8.41 the mesh density on one of its equilateral triangular bases is illustrated prior (left) and after (right) adaptation.

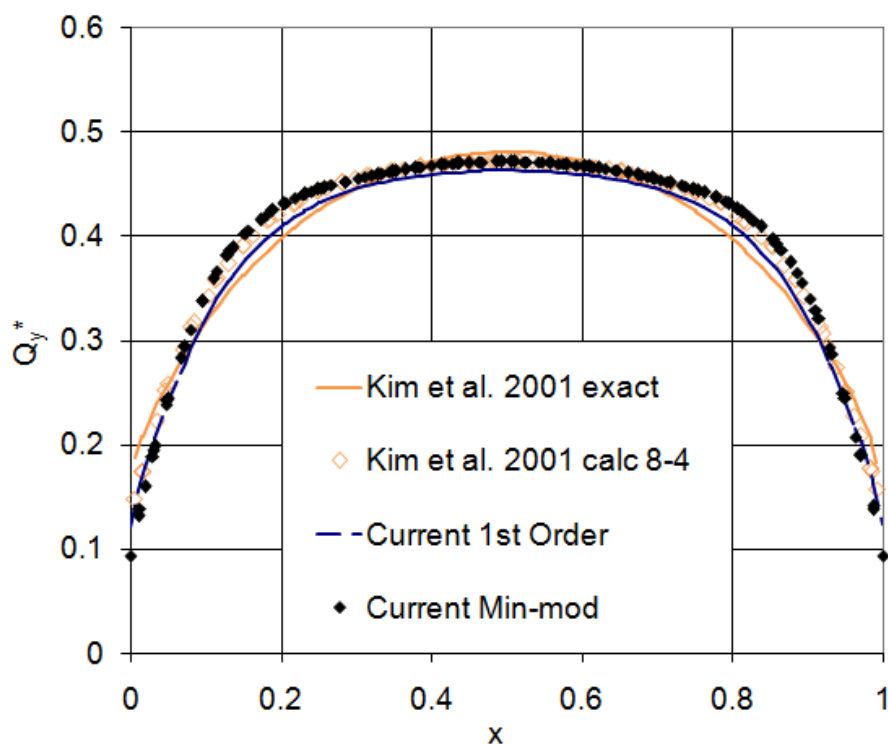


Figure 8.38: Distributions of dimensionless incident radiative heat flux along the A-A line (very fine grid, 8-4 angular resolution, prismatic enclosure - first sub-case).

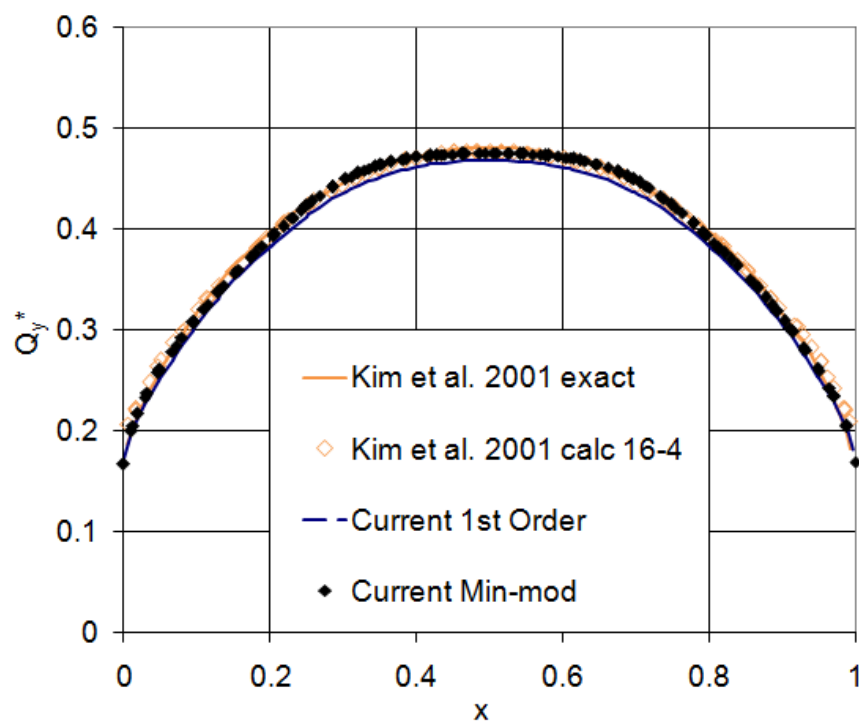


Figure 8.39: Distributions of dimensionless incident radiative heat flux along the A-A line (very fine grid, 16-4 angular resolution, prismatic enclosure - first sub-case).

Table 8.10: Parameters of simulation (prismatic enclosure - second sub-case).

Parameters	
Absorption coefficient	$0.1, 1.0, 10.0 \text{ m}^{-1}$
Scattering coefficient	0 m^{-1}
Wall emissivity	1
Grid density	29,170 nodes, 81,217 tetrahedra and 26,100 prisms
Number of partitions	4
Angular resolution	8 azimuthal - 8 polar control angles
Iterative scheme	Runge-Kutta method
Computational system	DELL T7400 workstation with two Intel ^(R) Xeon ^(R) -E5410 four-core processors at 2.33 GHz

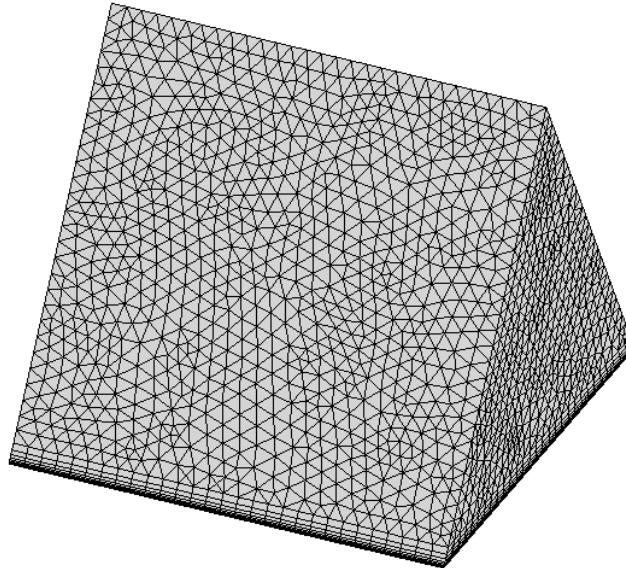
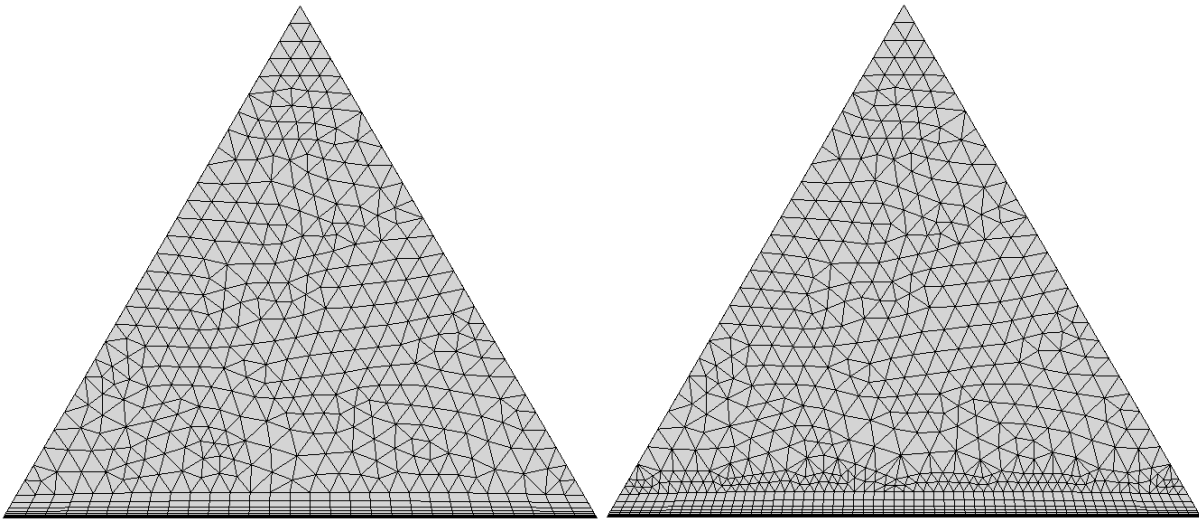
*Figure 8.40: Utilized mesh (prismatic enclosure - second sub-case).**Figure 8.41: Mesh density on an equilateral triangular base ($z=0$) prior (left) and after (right) h -refinement (prismatic enclosure - second sub-case).*

Figure 8.42 includes the distributions of dimensionless incident radiative heat flux along the half B-B line of the initial and refined mesh for the pre-mentioned values of absorption coefficient ($k_a=0.1, 1.0, 10.0 \text{ m}^{-1}$), compared with the corresponding exact ones of Grissa et al. [Gri10]. The encountered simulations succeeded at least a four-order of residual decrease, requiring approximately two seconds per iteration for the initial mesh and five seconds for the refined one. As far as the runs with absorption coefficient values equal to 0.1 and 1.0 m^{-1} are concerned, no significant accuracy improvement is identified, except that values of intensity can be retrieved from the area very close to the cold walls. The situation is changing in case of absorption coefficient equal to 10.0 m^{-1} ; a difference in the same area between the two grids is clearly observed, revealing the potential of h-refinement for more accurate solutions, without the need to start from scratch the construction of a new finer grid. Although the accuracy improvement, obtained by grid adaptation, isn't so significant, compared to the one provided by the implementation of second-order accurate schemes, it can be proved helpful depending on the examined problem.

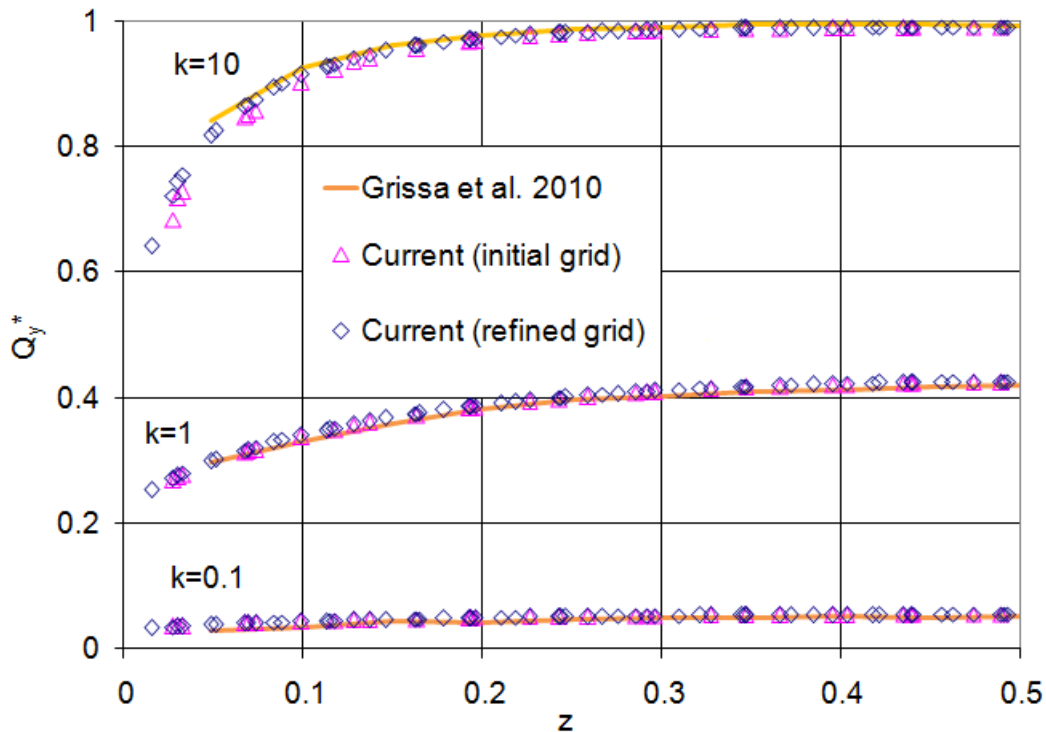


Figure 8.42: Distributions of dimensionless incident radiative heat flux along the half B-B line for different values of absorption coefficient (prismatic enclosure - second sub-case).

8.5. A cylindrical enclosure

This test case concerns radiative heat transfer in a cylindrical enclosure with the radius of each base equal to unity and height equal to 2 m , as this illustrated in Figure 8.43 [Sal04, Hun11, Lyg12b, Lyg14g]. Three different sub-cases were encountered on this enclosure, whose differences are focused on the type of medium (absorbing or scattering) as well as on the source surface generating heat transfer (medium or wall surface).

In the first sub-case, the included medium is considered absorbing and emitting, but no scattering, ($\sigma_s=0$) held at constant temperature ($T_m=100 \text{ K}$); three different absorption coefficients k_a were employed, namely $0.1, 1.0$ and 5.0 m^{-1} . All the walls are assumed radiatively black ($\epsilon_w=1$) and cold ($T_w=0 \text{ K}$), while the corresponding boundary conditions were implemented explicitly (Dirichlet conditions). The utilized grid consists of $95,716$ nodes, $447,436$ tetrahedra and $30,352$ prisms, while for parallel processing it was decomposed in twelve partitions; its boundary mesh density is presented in Figure 8.44. The angular

domain was divided by twelve azimuthal and twelve polar angles, while bold approximation was utilized to alleviate any possible overhang problem. The solution of non-time-dependent RTE with iterative approximations and a first-order accurate spatial scheme was performed on a DELL T7500 workstation with two Intel^(R) Xeon^(R)-X5650 six-core processors at 2.67 GHz, succeeding a final radiative intensity residual equal to $1.0E-9$. The parameters of these runs are included in Table 8.11. Figure 8.45 illustrates the extracted distributions of dimensionless incident radiative heat flux along the A-A line (cylinder lateral surface), compared with those obtained by the corresponding three-dimensional cell-centered finite-volume algorithm of Salah et al. [Sal04]. A satisfactory agreement between the computed and reference results for all the employed absorption coefficients can be identified; as expected an increase to the value of this coefficient leads to an increased blackbody term of RTE and consequently to an increased incident radiative heat flux. Independently of the absorption coefficient value, a rapid decrease of the exiting flux is observed near the circular bases, due to the neighbouring cold walls.

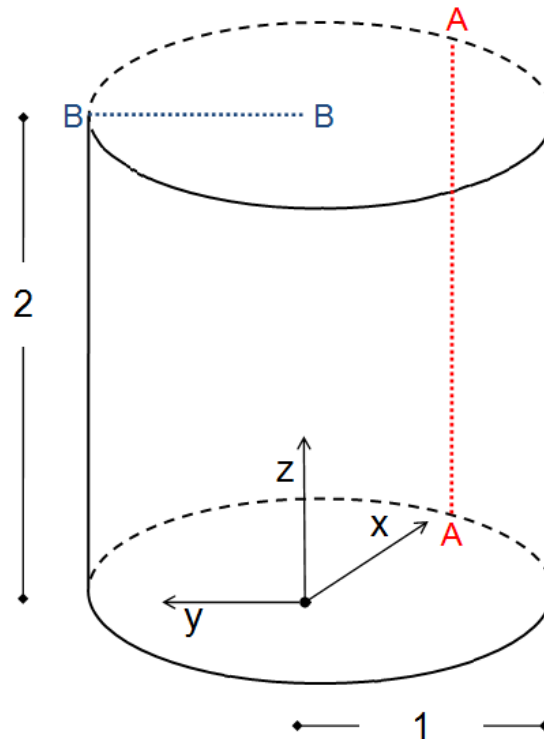


Figure 8.43: Dimensions of cylindrical enclosure.

Table 8.11: Parameters of simulation (cylindrical enclosure - first sub-case).

Parameters	
Absorption coefficient	$0.1, 1.0, 5.0 \text{ m}^{-1}$
Scattering coefficient	0 m^{-1}
Wall emissivity	1
Grid density	95,716 nodes, 447,436 tetrahedra and 30,352 prisms
Number of partitions	12
Angular resolution	12 azimuthal - 12 polar control angles
Iterative scheme	Iterative approximations
Computational system	DELL T7500 workstation with two Intel ^(R) Xeon ^(R) -X5650 six-core processors at 2.67 GHz

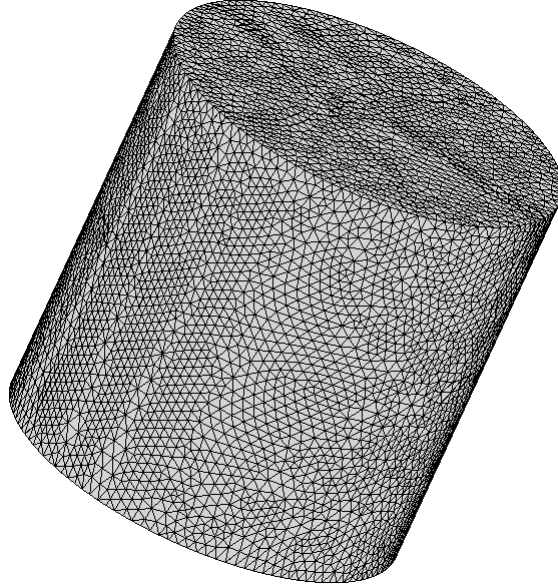


Figure 8.44: Utilized grid (cylindrical enclosure - first and second sub-case).

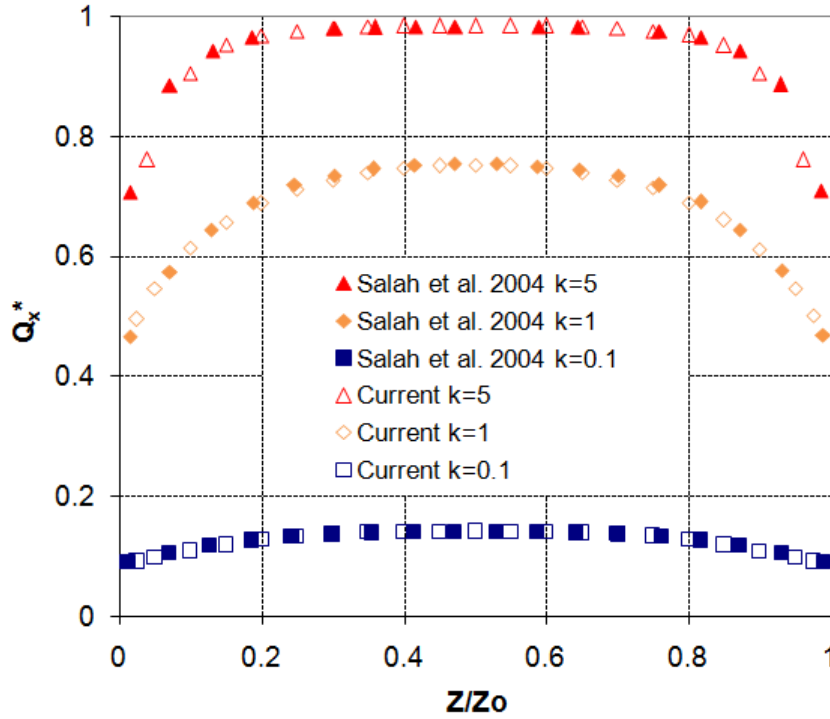


Figure 8.45: Distributions of dimensionless incident radiative heat flux along the A-A line for different values of absorption coefficient (cylindrical enclosure - first sub-case).

For the second sub-case, the enclosure is filled with a purely scattering medium ($k_a = 0 \text{ m}^{-1}$, $\sigma_s = 1 \text{ m}^{-1}$) instead, maintained cold ($T_m = 0 \text{ K}$); an isotropic and a forward anisotropic (F2) scattering function was employed for the validation of scattering capability of the developed methodology. All the walls are considered black ($\epsilon_w = 1$); the circular bases are assumed cold ($T_{wb} = 0 \text{ K}$), while the lateral surface at constant temperature ($T_{wl} = 100 \text{ K}$). Radiative intensity at walls was explicitly imposed (Dirichlet boundary conditions), while the steady (non time-dependent) RTE was solved with iterative approximations along with a first-order accurate spatial scheme. The same to the previous sub-case spatial (grid) and angular

(12 azimuthal - 12 polar control angles along with bold approximation) resolution was employed on the same computational system, deriving a radiative intensity residual equal to $1.0E-6$ in approximately $\sim 6.5E-5$ s per iteration and number of DoF's (anisotropic scattering run). The parameters of this simulation are summarized in Table 8.12.

Table 8.12: Parameters of simulation (cylindrical enclosure - second sub-case).

Parameters	
Absorption coefficient	0 m^{-1}
Scattering coefficient	1 m^{-1}
Wall emissivity	1
Grid density	95,716 nodes, 447,436 tetrahedra and 30,352 prisms
Number of partitions	12
Angular resolution	12 azimuthal - 12 polar control angles
Iterative scheme	Iterative approximations
Computational system	DELL T7500 workstation with two Intel ^(R) Xeon ^(R) -X5650 six-core processors at 2.67 GHz

Figure 8.46 includes the obtained distributions of dimensionless incident radiative heat flux along the B-B line (half base surface) for an isotropically and anisotropically scattering medium, compared with the corresponding ones provided by the study of Salah et al. [Sal04]. As expected, the forward anisotropic scattering function (F2) produces higher values of exiting intensity, comparing to the isotropic one, as it enables greater diffusion. Computed and reference incident radiation fluxes compare satisfactorily, indicating the good potential of the developed algorithm for such simulations.

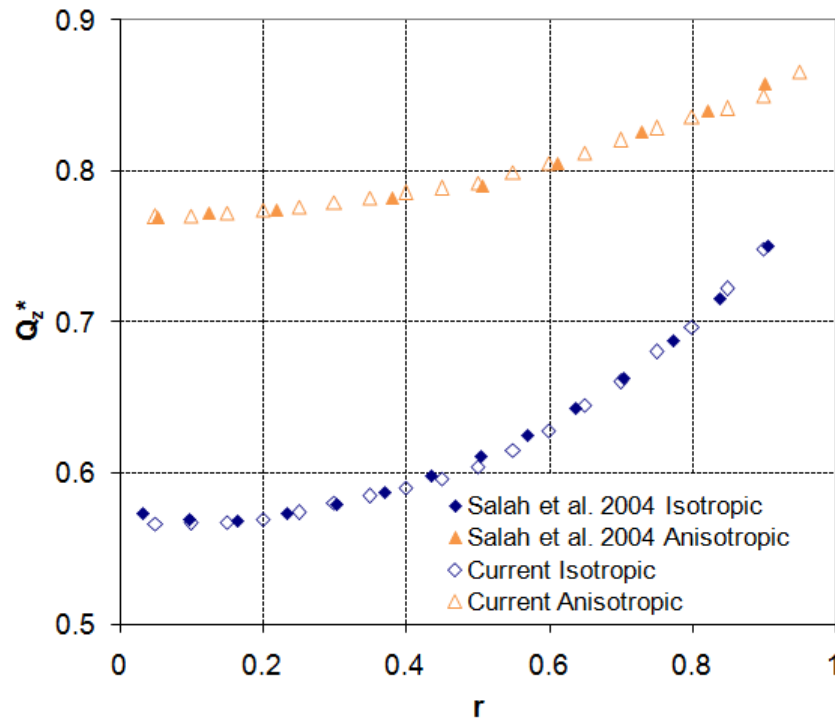


Figure 8.46: Distributions of dimensionless incident radiative heat flux along the B-B line for isotropically and anisotropically scattering medium (cylindrical enclosure - second sub-case).

Additionally, in order to assess the parallelization strategy incorporated in the present solver, similar to the previous case runs (concerning an anisotropically scattering medium) were performed on the same computational system, utilizing two, four, and eight processors; in Figure 8.47 the derived speed-up/number of processors distribution is illustrated. As the number of employed processors increases a deviation between the ideal and the achieved performance is observed, deriving from the increase of communication load. Nevertheless, it is an acceptable deviation, which can be further improved if the proposed methodology is applied to larger scale problems with more DoF's.

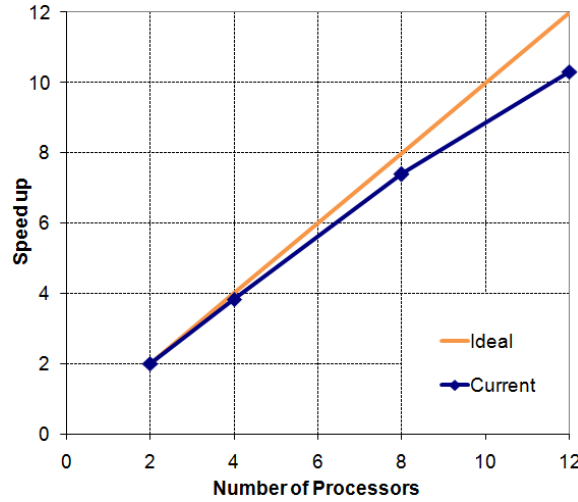


Figure 8.47: Scalability of parallel implementation (anisotropic scattering, cylindrical enclosure - second sub-case).

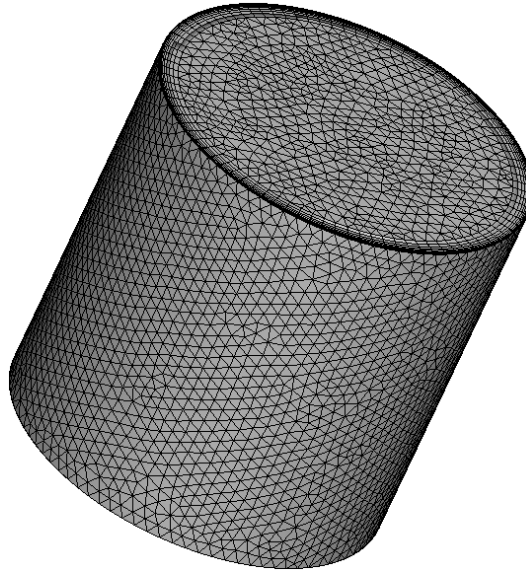


Figure 8.48: Employed mesh (cylindrical enclosure - third sub-case).

The third sub-case concerning radiative heat transfer in a cylindrical enclosure, was encountered mainly for the evaluation of the developed spatial/angular agglomeration multigrid methodology. The included absorbing and isotropically scattering medium is maintained cold ($T_m=0$ K), while runs with different values of scattering albedo were performed ($\beta=1$ m⁻¹, $\omega=0.1-1.0$). All the walls are considered black ($\epsilon_w=1$), while their radiative intensity is computed implicitly; the circular bases are assumed cold ($T_{wb}=0$ K), while the lateral surface exhibits a constant unity heating power ($E_{lateral}=1$ W/m² corresponding to 64.80435 K). The employed grid consists of 55,572 nodes, 131,731 tetrahedra and 61,040 prisms, while

for parallel processing, it was decomposed in four partitions; its boundary mesh density is presented in Figure 8.48. The angular *sphere* was divided by sixteen azimuthal and eight polar angles, while the pixelation method was used to mitigate any overhang problem. The solution of the time-dependent RTE with the Runge-Kutta method, applying a first-order accurate spatial scheme, was performed on a workstation equipped with an AMD FX^(tm)-8350 eight-core processor at 4.00 GHz. The parameters of this simulation are included in Table 8.13.

Table 8.13: Parameters of simulation (cylindrical enclosure - third sub-case).

Parameters	
Absorption coefficient	$0-1 \text{ m}^{-1}$
Scattering coefficient	$0-1 \text{ m}^{-1}$
Wall emissivity	1
Grid density	55,572 nodes, 131,731 tetrahedra and 61,040 prisms
Number of partitions	4
Angular resolution	16 azimuthal - 8 polar control angles
Iterative scheme	Runge-Kutta method
Computational system	Workstation with an AMD FX ^(tm) -8350 eight-core processor at 4.0 GHz

Figure 8.49 includes the obtained distributions of dimensionless incident radiative heat flux along the A-A line for scattering albedo ω equal to 0.1 and 1.0. The value of normalized heat flux has a minimum at the midplane ($z=1 \text{ m}$), where the effects of cold bases are minimized. Although a first-order accurate spatial scheme was employed, the derived results are close enough with the corresponding ones obtained by the finite-volume algorithm of Hunter and Guo [Hun11].

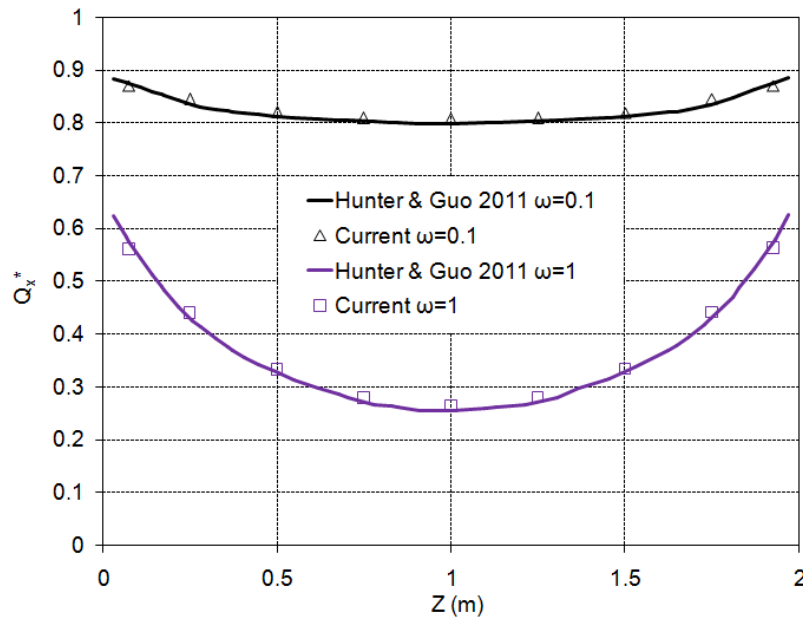


Figure 8.49: Distributions of dimensionless incident radiative heat flux along the A-A line for values 0.1 and 1.0 of scattering coefficient (cylindrical enclosure - third sub-case).

In order to implement the developed multigrid methodology, three coarser meshes were generated via isotropic agglomeration, at each of which the included DoF's were reduced about four times; in Figure

8.50 the boundary density of the initial and agglomerated control volume grids is illustrated. Similarly, angular agglomeration was performed, deriving a coarser angular resolution, including eight azimuthal and four polar angles.

Only the spatial agglomeration multigrid scheme was implemented in the simulation considering the medium with 0.1 scattering albedo, due to the reduced radiation exchange between different solid control angles. Figure 8.51 includes the radiative intensity convergence history, as well as the corresponding convergence rate, per number of iterations/cycles and time of the multigrid scheme (S4A1), compared to those of the single-grid one (S1A1). Considering the depicted temporal speed-up coefficient, computed for a residual value of $1.0E-8$, the multigrid run converged more than twice faster than the single-grid one; because of the absence of symmetry and reflecting surfaces as well as of the small value of the scattering coefficient, the produced rate is assumed relatively satisfactory.

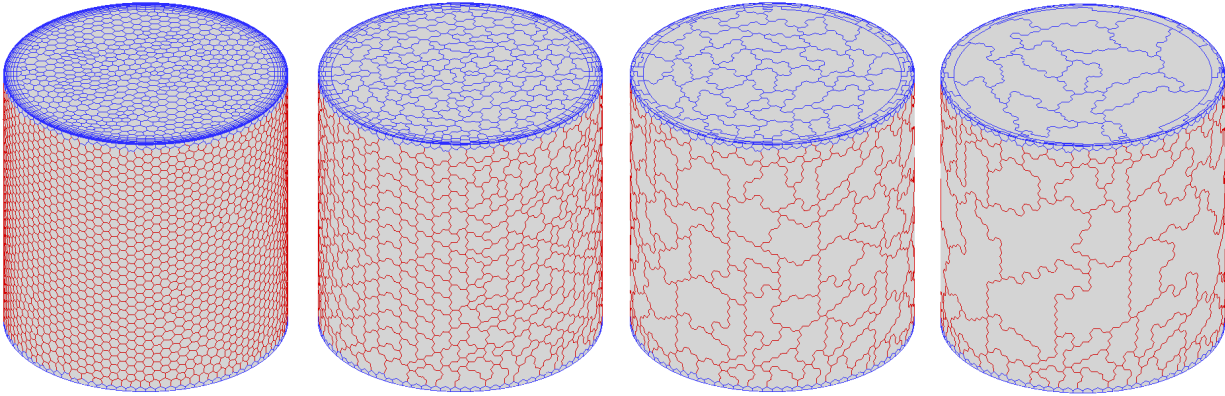


Figure 8.50: Initial and agglomerated control volume grids (cylindrical enclosure - third sub-case).

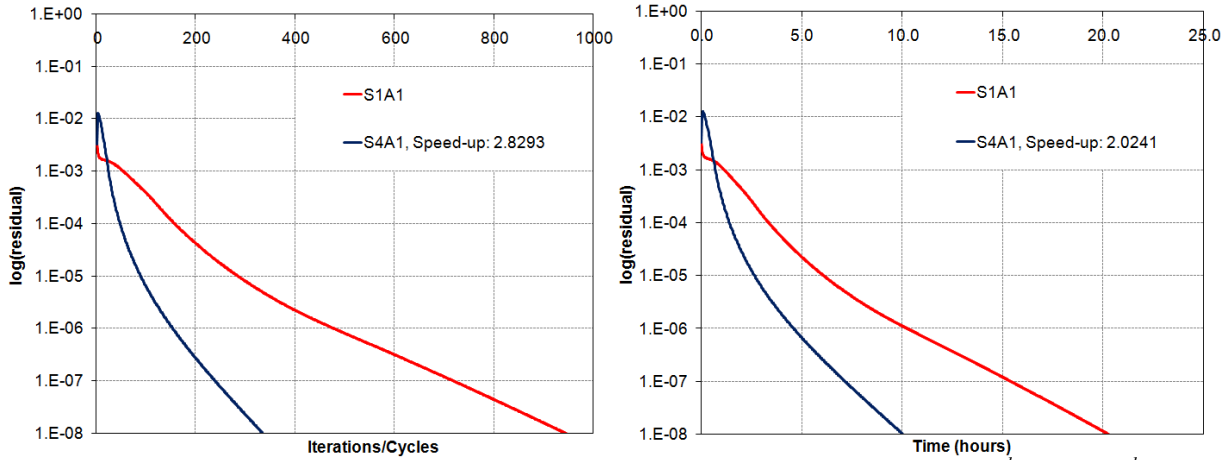


Figure 8.51: Radiative intensity convergence history per iterations/cycles and time ($k_a=0.9 \text{ m}^{-1}$, $\sigma_s=0.1 \text{ m}^{-1}$, $\omega=0.1$, $\varepsilon_w=1$, cylindrical enclosure - third sub-case).

For the second simulation regarding a medium with a unit scattering albedo, various multigrid schemes (S1A3, S4A1 and S4A2) were implemented besides the single-grid one (S1A1). Figure 8.52 includes the corresponding convergence histories (per iterations/cycles and computation time) and speed-up coefficients, the latter computed for final residual equal to $1.0E-8$; the significant contribution of spatial agglomeration as well as the negligible one of angular fusion is revealed. However, if these rates are re-computed for a value of residual equal to $1.0E-6$, a considerable increase is observed; more specifically this re-calculation results in ~ 1.37 for S1A3, ~ 3.24 for S4A1 and ~ 3.41 for S4A2. Considering that a decrease in residual of three orders of magnitude is usually enough for the extraction

of the required incident radiative heat flux distributions, the contribution of angular agglomeration appears to be non-trivial, especially in case the combined spatial/angular scheme is selected.

Finally, dummy runs, regarding semi-reflecting surfaces with wall emissivity ε_w equal to 0.5, were performed with the single-grid (S1A1) and various multigrid schemes (S1A3, S4A1 and S4A2), in order to examine the effect of reflectivity. In Figure 8.53 the multigrid convergence histories and acceleration coefficients are illustrated, compared to this of the non-multigrid one; the maximum obtained speed-up coefficient in time for a residual equal to $1.0E-8$ is ~ 3.3 , derived by the nested spatial/angular scheme (S4A2). Re-computing this rate for a higher but simultaneously sufficient residual equal to $1.0E-6$, provides an even higher value of ~ 4.3 , indicating the enhancement of the present solver by the incorporated angular agglomeration multigrid methodology.

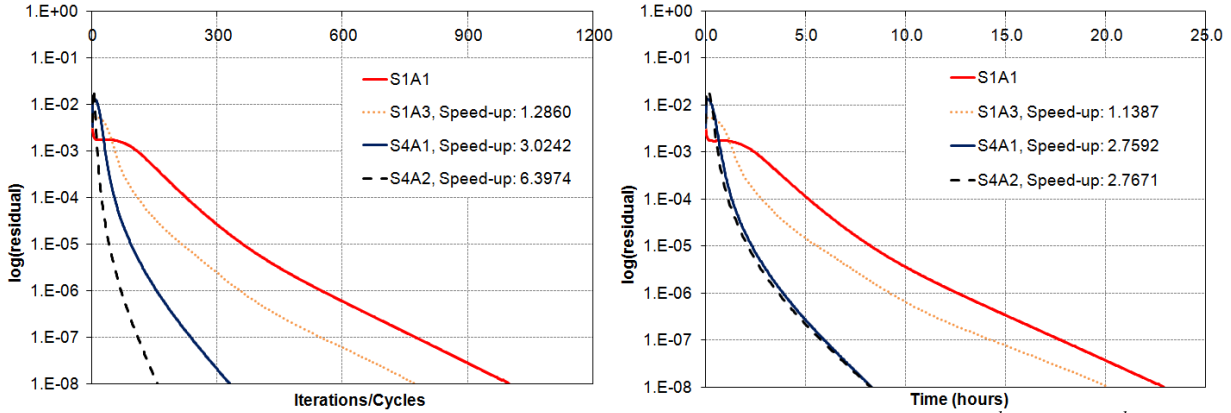


Figure 8.52: Radiative intensity convergence history per iterations/cycles and time ($k_a=0 \text{ m}^{-1}$, $\sigma_s=1.0 \text{ m}^{-1}$, $\omega=1.0$, $\varepsilon_w=1$, cylindrical enclosure - third sub-case).

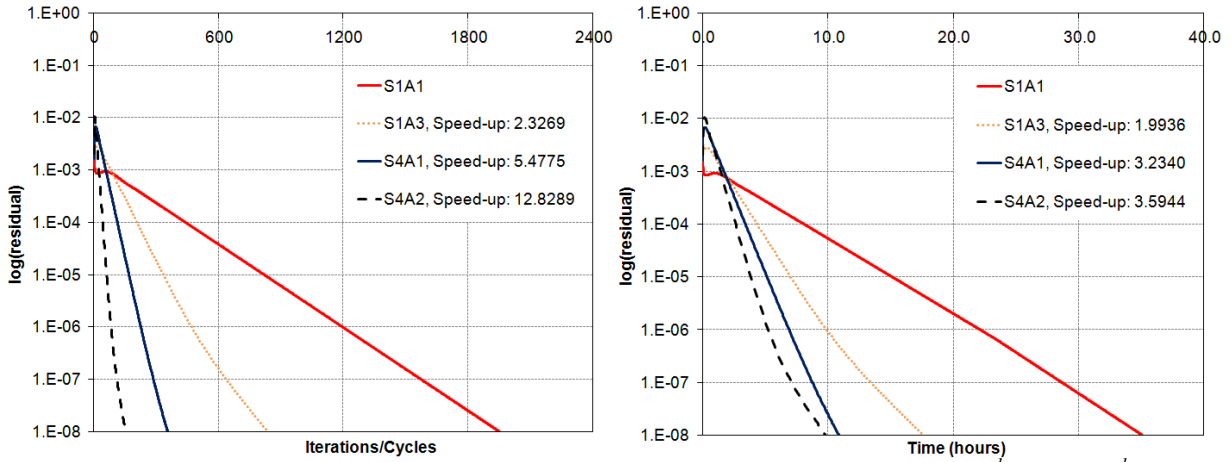


Figure 8.53: Radiative intensity convergence history per iterations/cycles and time ($k_a=0 \text{ m}^{-1}$, $\sigma_s=1.0 \text{ m}^{-1}$, $\omega=1.0$, $\varepsilon_w=0.5$, cylindrical enclosure - third sub-case).

8.6. A hemi-cylindrical enclosure

In this test case radiative heat transfer is predicted in a hemi-cylindrical enclosure, as this illustrated in Figure 8.54 [Kim08, Lyg13c]. All the walls are assumed black ($\varepsilon_w=1$) and cold ($T_w=0 \text{ K}$), except for the two half-circular bases in which symmetry boundary conditions are imposed; for the computation of radiative intensity on these surfaces the implicit treatment was selected. The included medium is considered absorbing and emitting, but no scattering ($\sigma_s=0 \text{ m}^{-1}$), while it is maintained at constant temperature ($T_m=1000 \text{ K}$); three different absorption coefficients k_a were employed, namely 0.1, 1.0 and 10.0 m^{-1} . The used mesh consists of 18,217 nodes, 53,658 tetrahedra, and 14,424 prisms, while for

parallel processing it was decomposed in two sub-domains; Figure 8.55 includes the utilized grid as well as the boundary density on one of its half-circular bases. Angular discretization is succeeded by dividing the directional domain in sixteen azimuthal and four polar angles, while the effects of overhang problem are subdued by the pixelation approach.

Table 8.14: Parameters of simulation (hemi-cylindrical enclosure).

Parameters	
Absorption coefficient	$0.1, 1.0, 10.0 \text{ m}^{-1}$
Scattering coefficient	0 m^{-1}
Wall emissivity	1
Grid density	18,217 nodes, 53,658 tetrahedra and 14,424 prisms
Number of partitions	2
Angular resolution	16 azimuthal - 4 polar control angles
Iterative scheme	Runge-Kutta method
Computational system	DELL T7400 workstation with two Intel ^(R) Xeon ^(R) -E5410 four-core processors at 2.33 GHz

The solution of the time-dependent RTE with the Runge-Kutta method, applying all the available second-order accurate spatial schemes besides the first-order one, was performed on a DELL T7400 workstation with two Intel^(R) Xeon^(R)-E5410 four-core processors at 2.33 GHz. The parameters of this test case are summarized in Table 8.14. Figure 8.56 illustrates the obtained distributions of dimensionless incident radiative heat flux along the A-B line (Figure 8.54) by all the performed simulations (various spatial schemes and absorption coefficients), compared with the corresponding computed ones provided in the work of Man Young Kim et al. [Kim08]. Between the results of runs with absorption coefficient equal to 0.1 m^{-1} but different spatial schemes, no difference can be identified. However for the rest simulations a deviation between the first- and the second-order accurate spatial schemes can be clearly distinguished. In all simulations, the results were extracted after radiative intensity residual was decreased at least seven orders of magnitude, requiring approximately twenty three minutes for the first-order accurate scheme and thirty four minutes for the second-order ones.

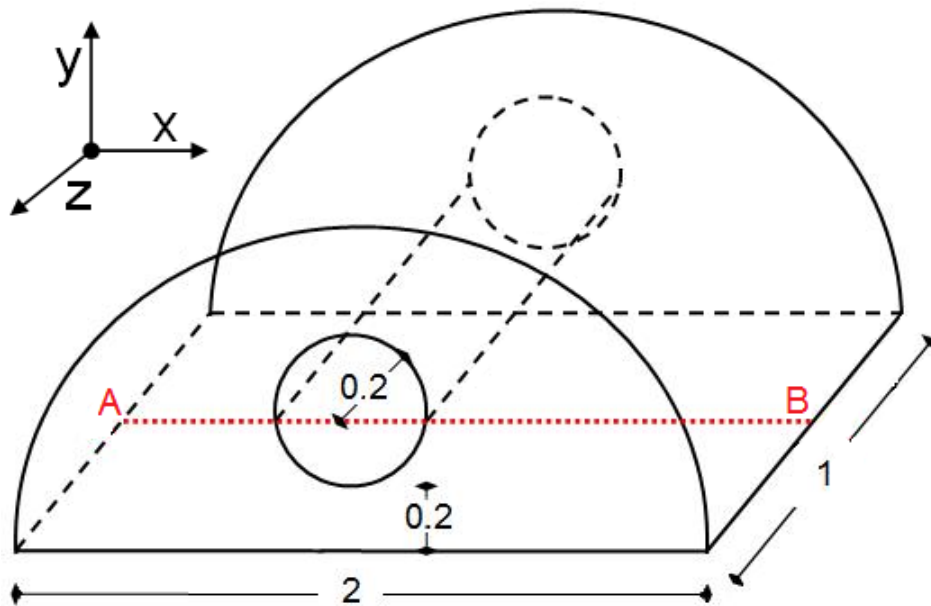


Figure 8.54: Dimensions of hemi-cylindrical enclosure.

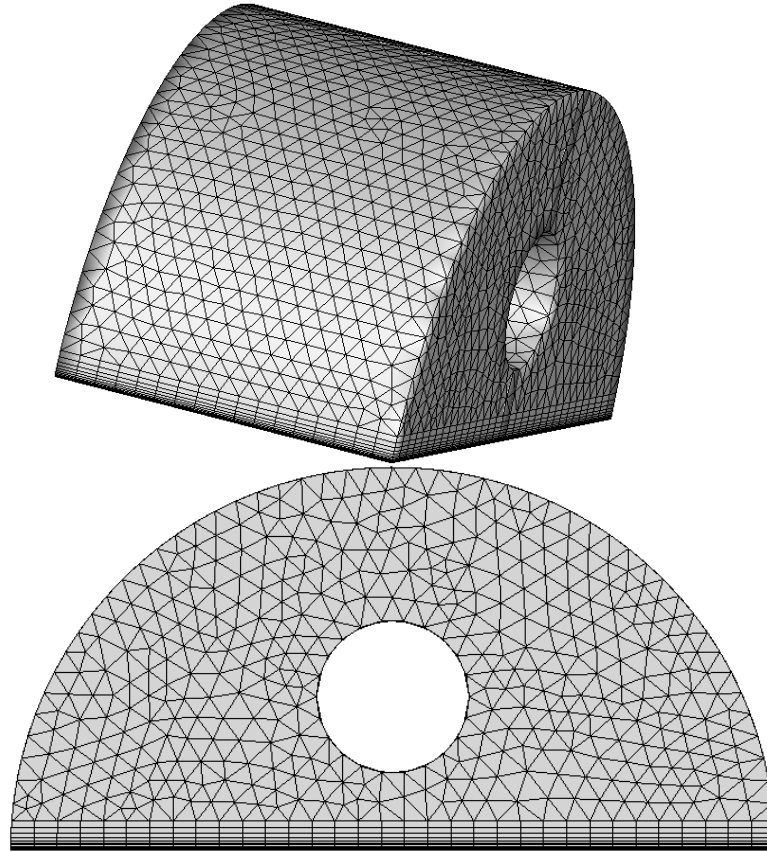


Figure 8.55: Employed grid and its density on symmetry surface (hemi-cylindrical enclosure).

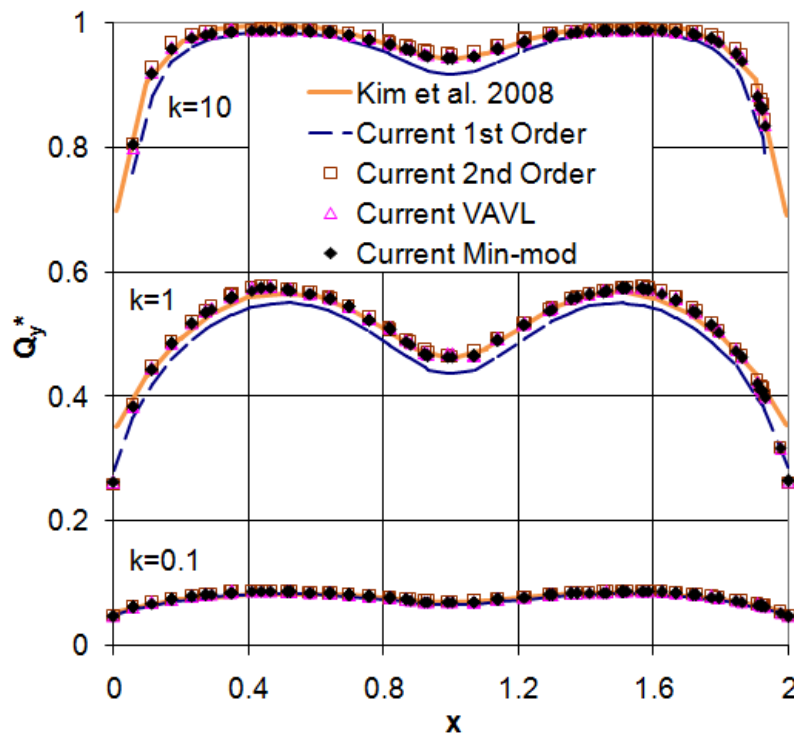


Figure 8.56: Distributions of dimensionless incident radiative heat flux along the A-B line for various values of absorption coefficient (hemi-cylindrical enclosure).

8.7. An L-shaped enclosure

An L-shaped enclosure, being illustrated in Figure 8.57, including an absorbing and emitting, but no scattering medium ($\sigma_s=0 \text{ m}^{-1}$), held at a constant temperature ($T_m=1000 \text{ K}$), is studied in this section [Tal05, Lyg13c]; three different absorption coefficients k_a were assumed, namely 0.5 , 1.0 and 2.0 m^{-1} . All the walls are considered black ($\varepsilon_w=1$) with a constant temperature ($T_w=500 \text{ K}$); for the computation of radiative intensity on these surfaces the implicit treatment was selected. The utilized tetrahedral mesh is composed of $18,130$ nodes and $96,010$ tetrahedra, while for parallel solution of the time-dependent RTE with the Runge-Kutta method it was divided in two partitions; Figure 8.58 includes the employed grid as well as the boundary density on the surface with A-B line. The directional *sphere* was divided by eight azimuthal and four polar angles, while pixelation method was implemented to alleviate the effects of overhang problem. Simulations for all the incorporated spatial schemes were performed on a DELL T7400 workstation with two Intel^(R) Xeon^(R)-E5410 four-core processors at 2.33 GHz , requiring approximately five minutes for the first-order accurate scheme and fifteen minutes for the second-order ones to decrease the radiative intensity residual at least nine orders of magnitude. The parameters of these runs are included in Table 8.15.

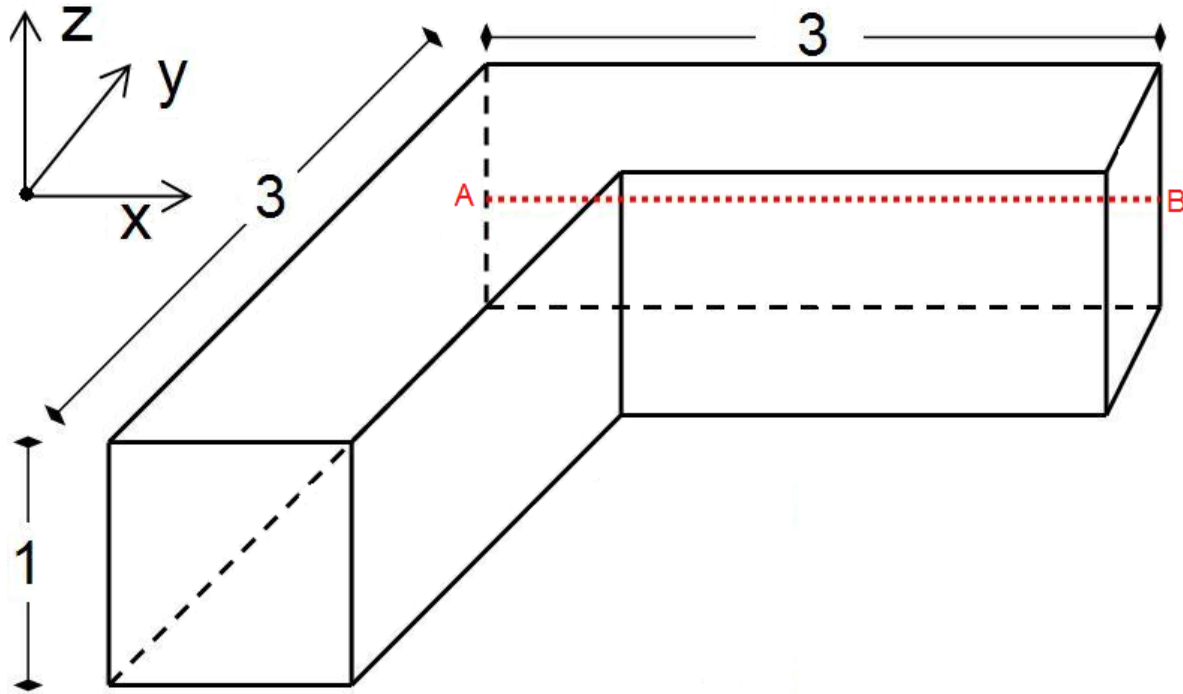
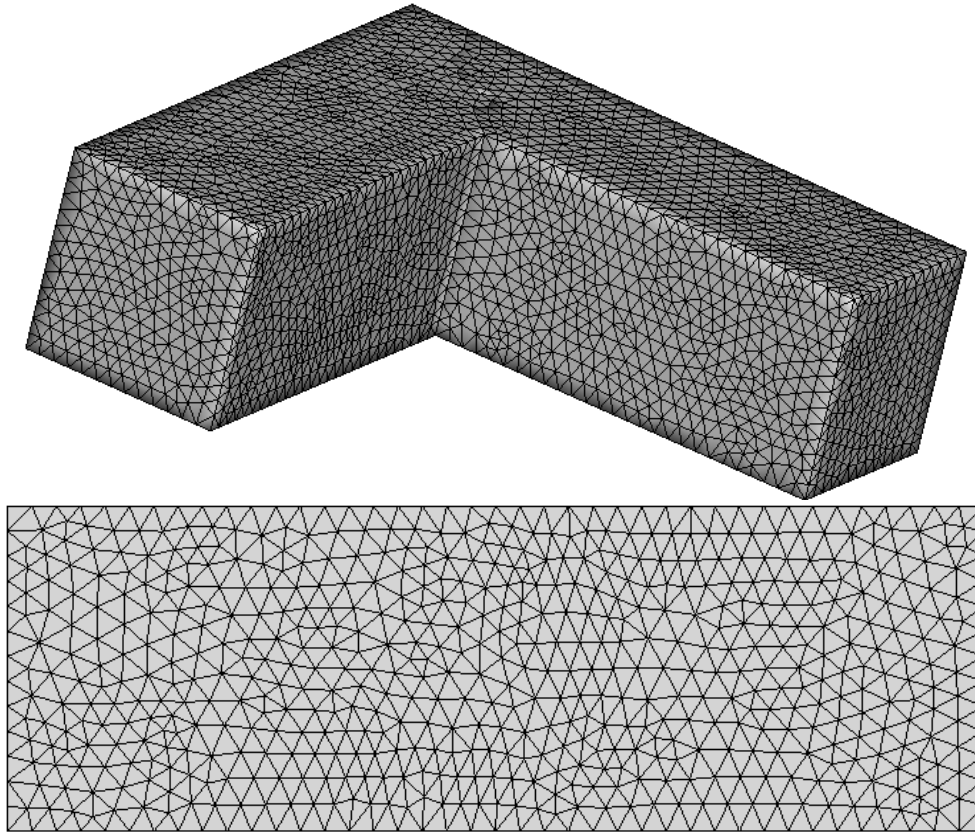


Figure 8.57: Dimensions of L-shaped enclosure.

In Figure 8.59 the extracted distributions of incident radiative heat flux (Q_y in kW/m^2) along the A-B line are presented for all the employed spatial schemes and values of absorption coefficient, compared to the corresponding computed ones provided in the paper of Talukdar et al. [Tal05]; a satisfactory agreement is achieved in general. Independently of the value of absorption coefficient, a significant improvement in accuracy of the exiting fluxes is obvious for the runs utilizing the second-order accurate spatial approaches; the deviation from the distribution of the first-order scheme increases with the increase of absorption coefficient. A slight flux overestimation can be identified in the distribution of the simple second-order scheme (without limiting) unlike the schemes coupled with the slope limiters (VAVL and Min-mod), due to the unbounded reconstructed values of radiative intensity. Nevertheless, the potential of all the developed higher-order accurate schemes for improved accuracy, even in very coarse spatial resolutions, such as in this test case, is demonstrated.

Table 8.15: *Parameters of simulation (L-shaped enclosure).*

Parameters	
Absorption coefficient	$0.5, 1.0, 2.0 \text{ m}^{-1}$
Scattering coefficient	0 m^{-1}
Wall emissivity	1
Grid density	18,130 nodes and 96,010 tetrahedra
Number of partitions	2
Angular resolution	8 azimuthal - 4 polar control angles
Iterative scheme	Runge-Kutta method
Computational system	DELL T7400 workstation with two Intel ^(R) Xeon ^(R) -E5410 four-core processors at 2.33 GHz

**Figure 8.58:** *Employed grid and its density on the surface with A-B line (L-shaped enclosure).*

8.8. A J-shaped enclosure

Radiative heat transfer in a J-shaped enclosure, being illustrated in Figure 8.60, is examined in this section [Kim08, Lyg13b]. It is filled with an absorbing and emitting ($k_a=0.01 \text{ m}^{-1}$), but no scattering ($\sigma_s=0 \text{ m}^{-1}$), cold medium ($T_m=0 \text{ K}$), while its walls are assumed black ($\varepsilon_w=1$) and cold ($T_w=0 \text{ K}$), except for the upper surface at $y=2.4 \text{ m}$ held at constant temperature of 100 K and the two bases at $z=0 \text{ m}$ and $z=1 \text{ m}$ for which symmetry boundary conditions are considered; boundary conditions are implemented implicitly. The employed grid includes 4,040 nodes, 11,132 tetrahedra and 2,800 prisms, while for parallel processing it was decomposed in two sub-domains; Figure 8.61 presents the utilized grid as well as the boundary density on one of its mirroring surfaces.

For angular discretization, a relatively fine resolution of thirty azimuthal and four polar angles was selected, due to the significant ray effect encountered; pixelation method was used to subdue the effects

of the overhang problem. Iterative solution of the time-dependent RTE for all the available spatial schemes was obtained with the implementation of Runge-Kutta method on a DELL T7400 workstation with two Intel^(R) Xeon^(R)-E5410 four-core processors at 2.33 GHz; it required approximately five minutes for the first-order scheme and fourteen minutes for the second-order ones, to achieve a four-order magnitude decrease of radiative intensity residual. The parameters of this test case are summarized in Table 8.16.

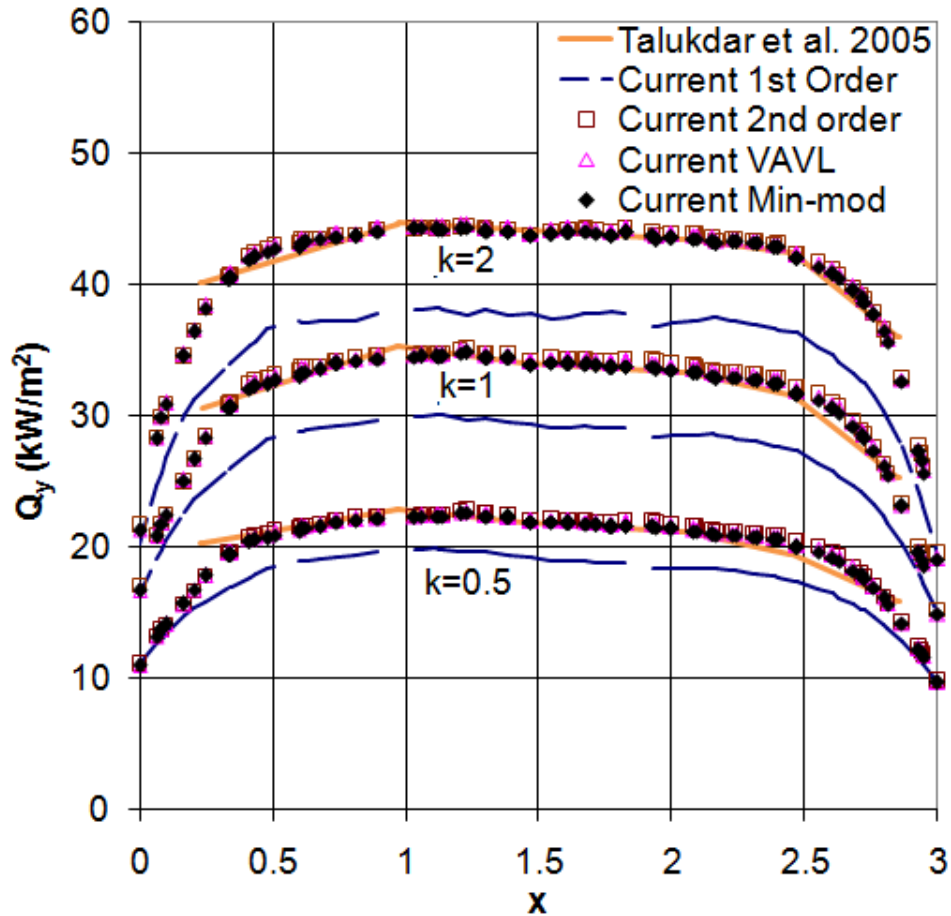


Figure 8.59: Distributions of incident radiative heat flux along the A-B line for various values of absorption coefficient (L-shaped enclosure).

Table 8.16: Parameters of simulation (J-shaped enclosure).

Parameters	
Absorption coefficient	0.01 m^{-1}
Scattering coefficient	0 m^{-1}
Wall emissivity	1
Grid density	4,040 nodes, 11,132 tetrahedra and 2,800 prisms
Number of partitions	2
Angular resolution	30 azimuthal - 4 polar control angles
Iterative scheme	Runge-Kutta method
Computational system	DELL T7400 workstation with two Intel ^(R) Xeon ^(R) -E5410 four-core processors at 2.33 GHz

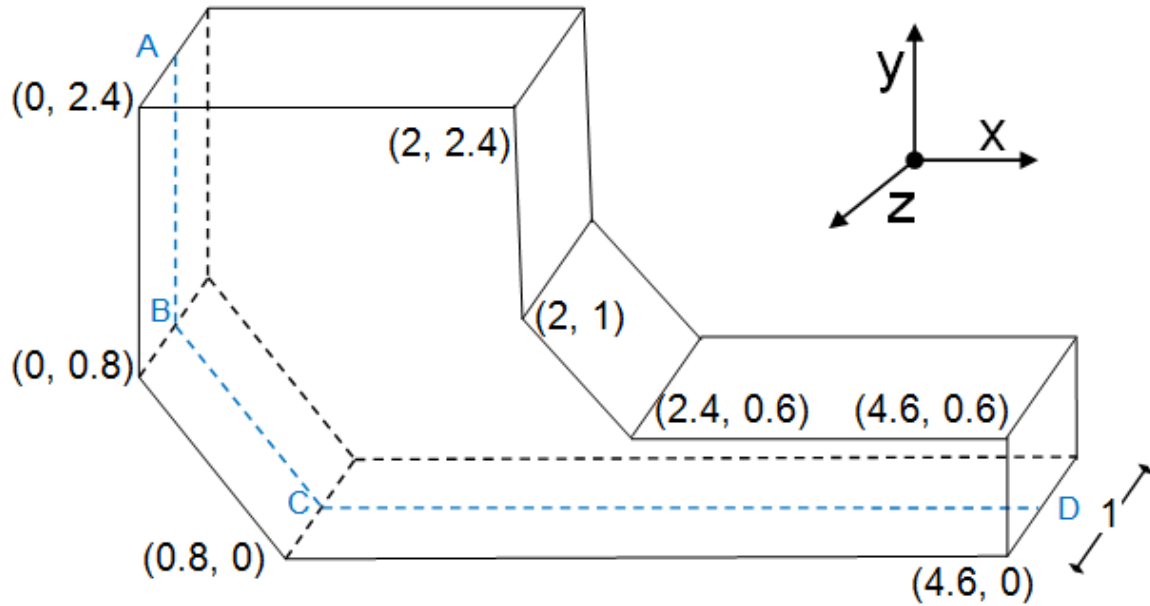


Figure 8.60: Dimensions of J-shaped enclosure.

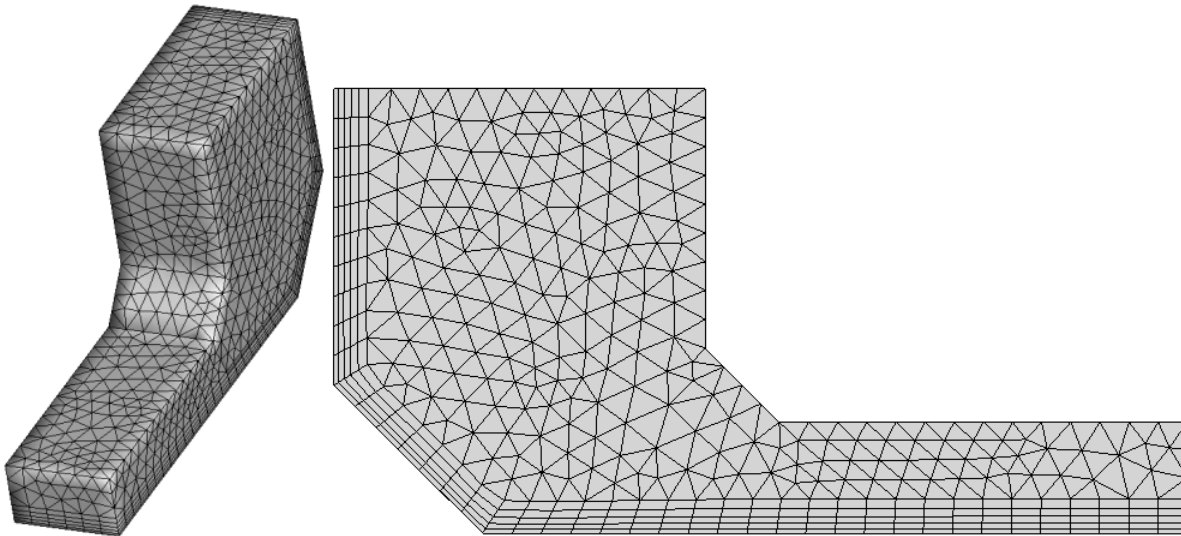


Figure 8.61: Utilized grid and its density on the symmetry surface (J-shaped enclosure).

Figure 8.62 illustrates the distribution of dimensionless incident radiative heat flux along the ABCD line ($z=0.5\text{ m}$) (Figure 8.60) for all the employed spatial schemes, compared with the computed ones of Man Young Kim et al. using two-dimensional triangular and polygonal meshes [Kim08]. Similarly to previous test cases, a deviation between the results of the first- and second-order runs is identified; this difference is more clearly distinguished in the areas near the points A, B and C. Moreover, the simple second-order scheme appears to mispredict slightly the exited fluxes in specific regions (e.g., near point B), comparing its results with those of the schemes enhanced with limiters. Nevertheless, the potential of all the developed higher-order schemes for improved accuracy of the final steady-state solution is revealed in this test case, despite the relatively coarse density of the utilized grid.

8.9. A hexahedral enclosure with trapezoidal bases

Another quasi-3D test case concerns radiative heat transfer in a hexahedral enclosure with trapezoidal bases; its dimensions are illustrated in Figure 8.63 [Mur98c, Lyg12a, Lyg13b]. Two sub-

problems were modelled for this enclosure, employing the time-dependent RTE along with the Runge-Kutta method on a DELL T7400 workstation with two Intel^(R) Xeon^(R)-E5410 four-core processors at 2.33 GHz.

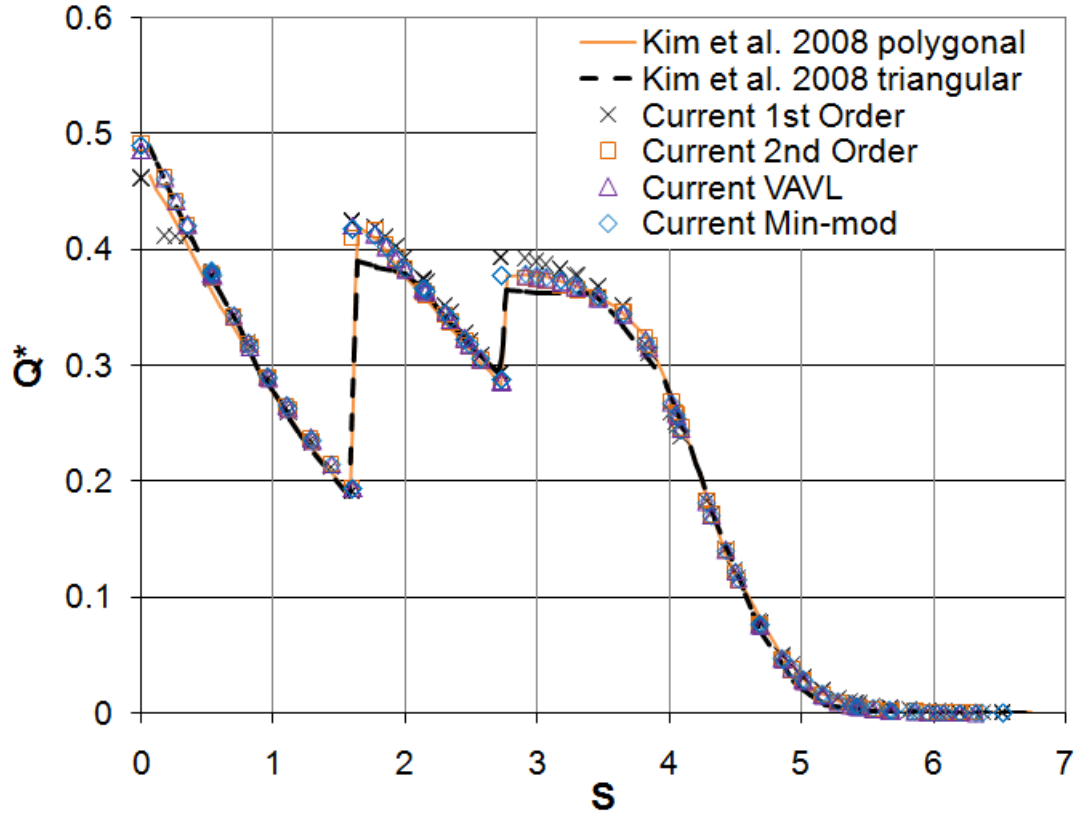


Figure 8.62: Distributions of dimensionless incident radiative heat flux along the ABCD line for various spatial schemes (J-shaped enclosure).

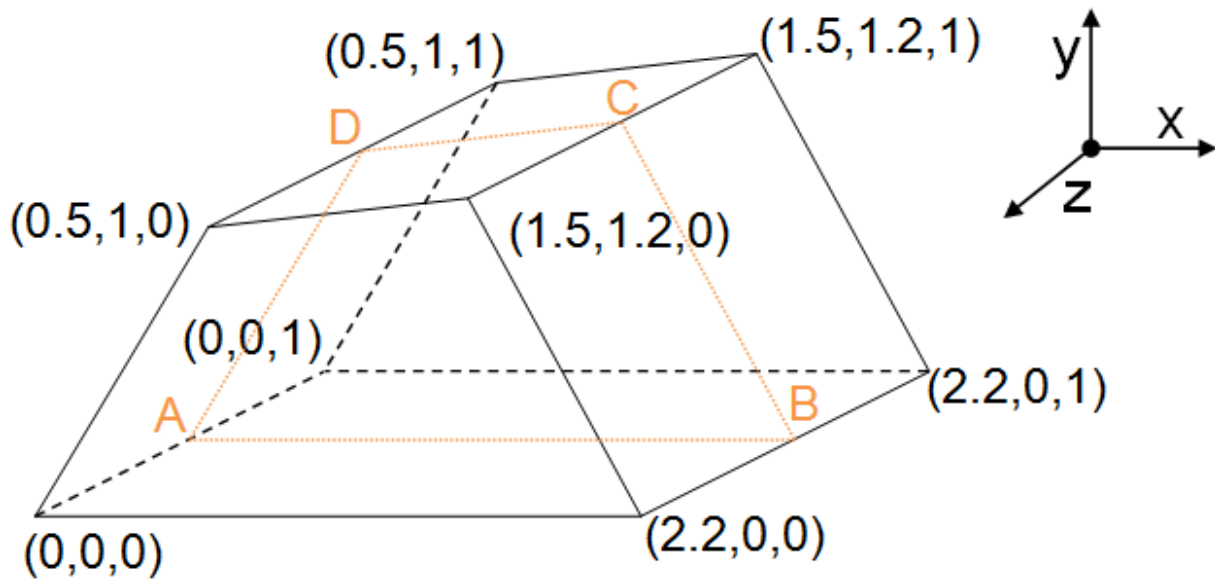


Figure 8.63: Dimensions of hexahedral enclosure with trapezoidal bases.

In the first sub-problem, the walls are assumed radiatively black ($\epsilon_w=1$) and cold ($T_w=0$ K), except for the two trapezoidal bases at $z=0$ m and $z=1$ m for which symmetry boundary conditions are imposed; radiative intensity on all the surfaces is implicitly obtained. The included medium is considered absorbing and emitting ($k_a=1$ m⁻¹), but no scattering ($\sigma_s=0$ m⁻¹), held at constant temperature of 100 K. The spatial field is discretized with a mesh composed of 8,507 nodes, 12,608 tetrahedra 10,754 prisms and 28 pyramids, while for parallel processing it was divided in two partitions; Figure 8.64 presents the utilized grid as well as the boundary density on one of its mirroring surfaces. Angular discretization is obtained with sixteen azimuthal and four polar angles, while pixelation method is employed to alleviate overhang problem and increase accuracy of the final steady-state solution. The parameters of this sub-case are contained in Table 8.17.

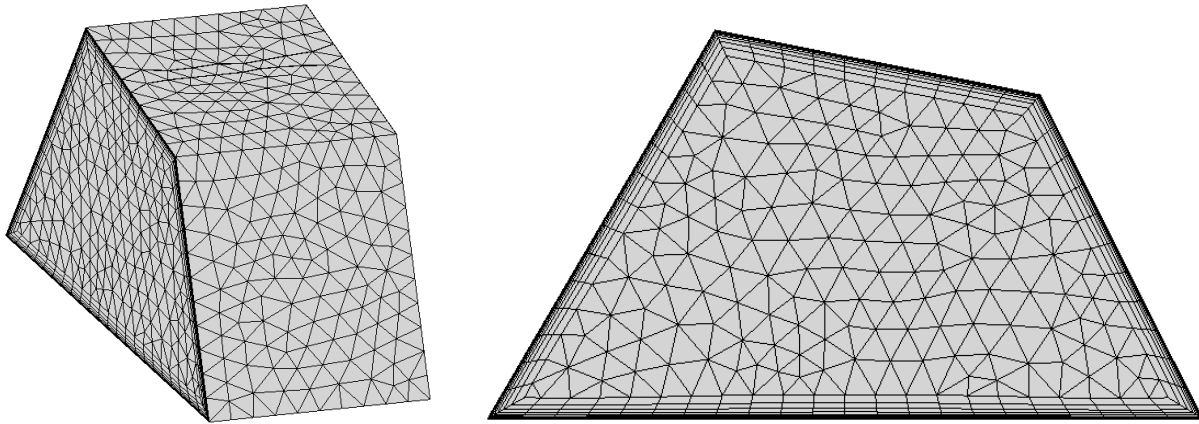


Figure 8.64: Utilized grid and its density on the symmetry surface (hexahedral enclosure with trapezoidal bases - first sub-case).

Table 8.17: Parameters of simulation (hexahedral enclosure with trapezoidal bases - first sub-case).

Parameters	
Absorption coefficient	1 m ⁻¹
Scattering coefficient	0 m ⁻¹
Wall emissivity	1
Grid density	8,507 nodes, 12,608 tetrahedra 10,754 prisms and 28 pyramids
Number of partitions	2
Angular resolution	16 azimuthal - 4 polar control angles
Iterative scheme	Runge-Kutta method
Computational system	DELL T7400 workstation with two Intel ^(R) Xeon ^(R) -E5410 four-core processors at 2.33 GHz

In Figures 8.65 to 8.68 the distributions of extracted dimensionless incident radiative heat flux along the AB, BC, CD and DA lines ($z=0.5$ m) (Figure 8.63) for all the available spatial schemes are illustrated, compared with the corresponding exact ones of Murthy and Mathur [Mur98c]. They were extracted after radiative intensity residual was decreased at least four orders of magnitude; approximately two and five seconds per iteration were required for the first- and second-order schemes respectively. As expected the higher-order runs derived more accurate results than the first-order one; moreover, those ones which are coupled with limiting functions managed to avoid the flux under-prediction near the cold wall boundaries, produced by the unlimited second-order scheme.

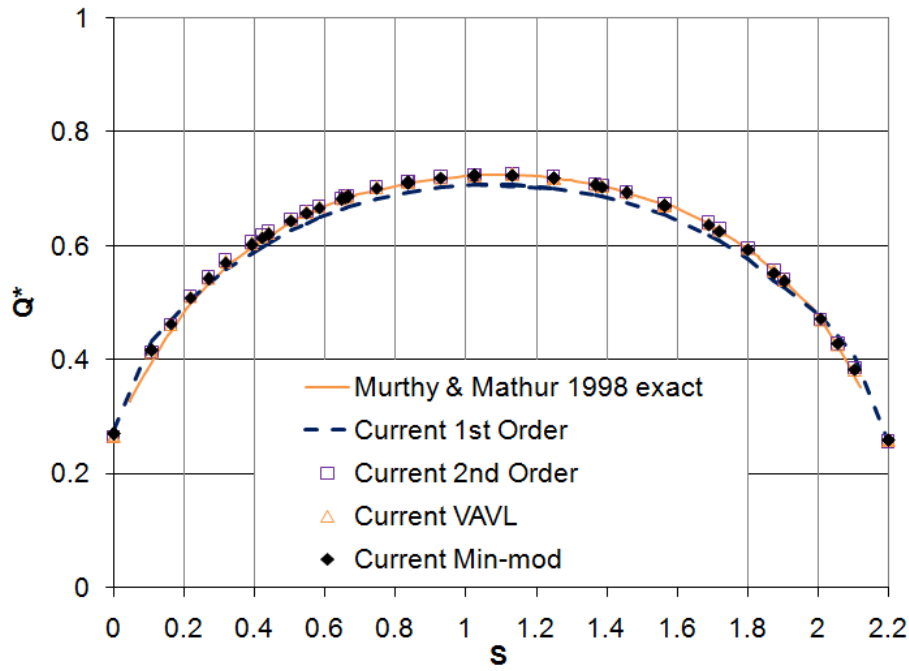


Figure 8.65: Distributions of dimensionless incident radiative heat flux along the AB line (hexahedral enclosure with trapezoidal bases - first sub-case).

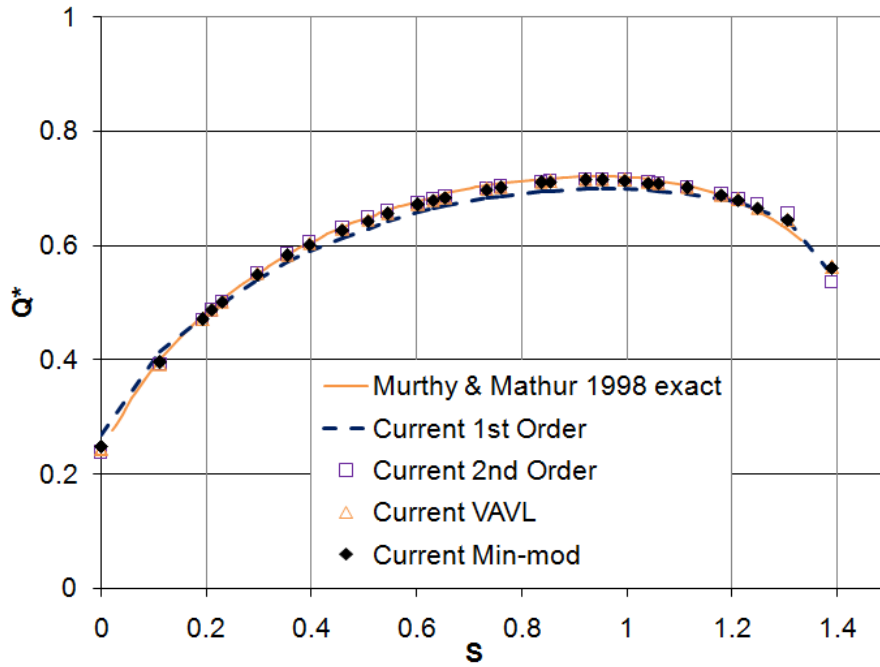


Figure 8.66: Distributions of dimensionless incident radiative heat flux along the BC line (hexahedral enclosure with trapezoidal bases - first sub-case).

In the second sub-problem the same behavior is assumed for the medium and the walls, except for the value of absorption coefficient; three different values are imposed, namely, $k_a = 0.1$, 1.0 and 10.0 m^{-1} . Similarly to the first sub-case boundary conditions are implicitly treated, but only a first-order accurate scheme is applied. The employed mesh includes 22,788 nodes, 53,692 tetrahedra and 23,184 prisms, while for parallel computation it was divided in two sub-domains; its boundary density is presented in

Figure 8.69. The division of angular field was achieved using twenty azimuthal and eight polar angles, while pixelation method was selected to manage the overhang problem. The parameters of this sub-case are summarized in Table 8.18.

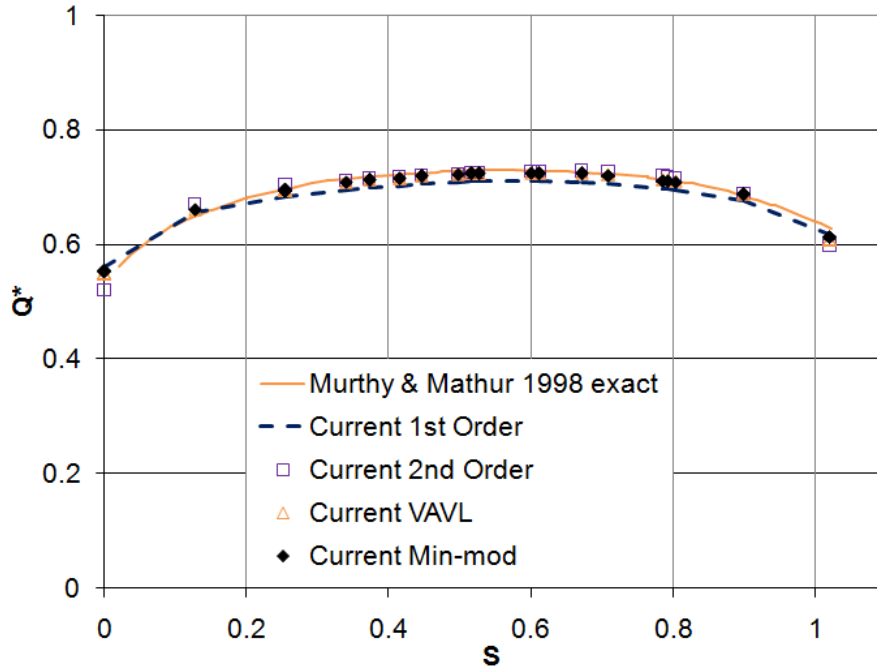


Figure 8.67: Distributions of dimensionless incident radiative heat flux along the CD line (hexahedral enclosure with trapezoidal bases - first sub-case).

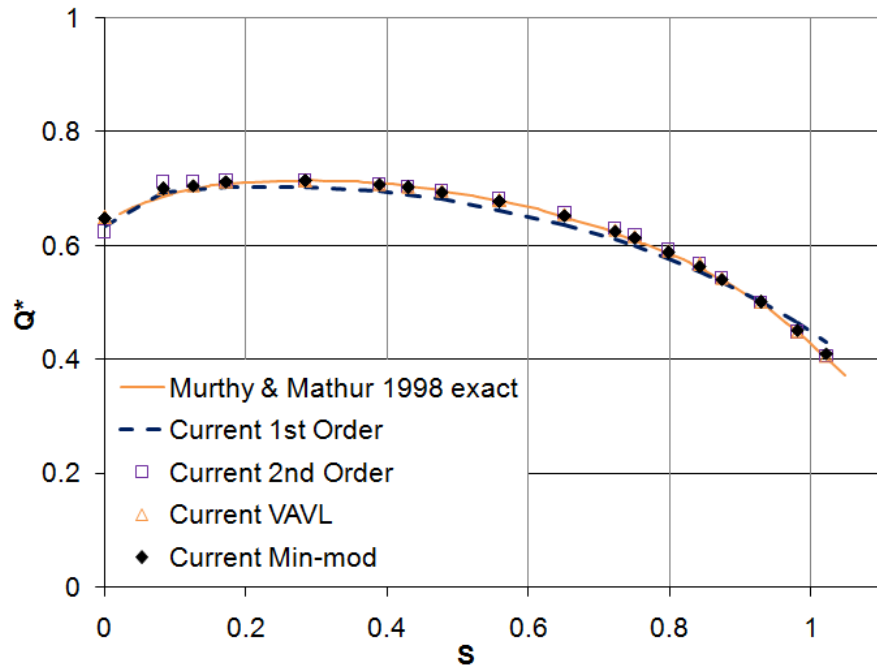


Figure 8.68: Distributions of dimensionless incident radiative heat flux along the DA line (hexahedral enclosure with trapezoidal bases - first sub-case).

The validation of the contribution of the developed grid adaptation methodology in accuracy improvement was the main goal of this sub-case. Therefore, the initial mesh was refined implementing a

criterion, which targeted the region up to 0.1 m away from the cold boundaries; as a result a new grid was derived, composed of 83,305 nodes, 184,522 tetrahedra and 92,736 prisms. However, as the solution, extracted by the run combining the refined mesh and a value of absorption coefficient equal to 10.0 m^{-1} wasn't satisfactory, especially at the areas near the lower cold corners, an additional h-refinement was applied, producing a mesh with 190,771 nodes, 428,159 tetrahedra and 217,812 prisms. Figure 8.70 illustrates the mesh density on the mirroring surface ($z=0\text{ m}$) prior and after the first and second h-refinement.

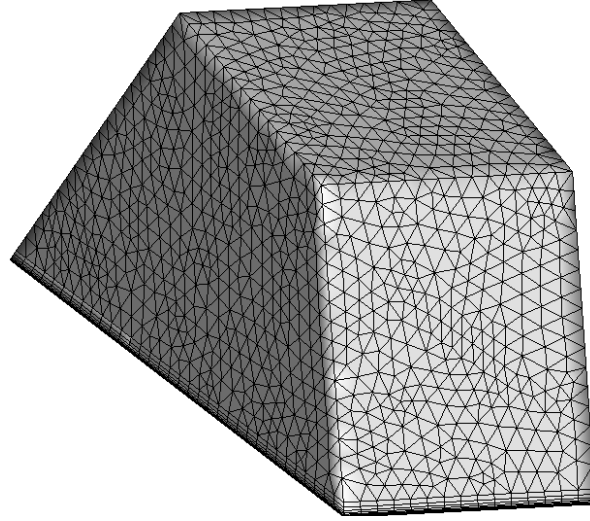


Figure 8.69: Boundary density of utilized grid (hexahedral enclosure with trapezoidal bases - second sub-case).

Table 8.18: Parameters of simulation (hexahedral enclosure with trapezoidal bases - second sub-case).

Parameters	
Absorption coefficient	$0.1, 1.0$ and 10.0 m^{-1}
Scattering coefficient	0 m^{-1}
Wall emissivity	1
Grid density	22,788 nodes, 53,692 tetrahedra and 23,184 prisms
Number of partitions	2
Angular resolution	20 azimuthal - 8 polar control angles
Iterative scheme	Runge-Kutta method
Computational system	DELL T7400 workstation with two Intel ^(R) Xeon ^(R) -E5410 four-core processors at 2.33 GHz

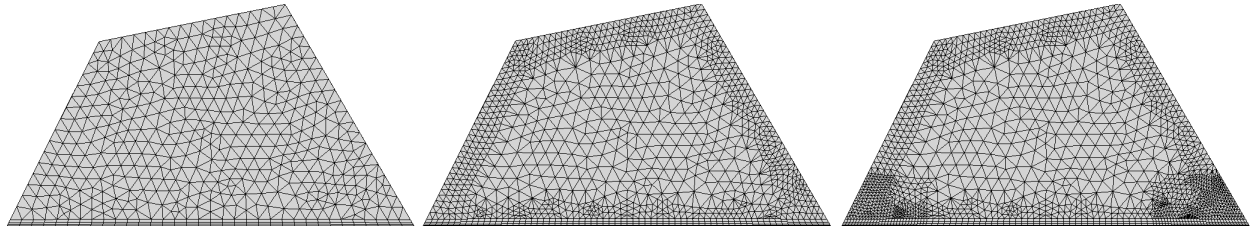


Figure 8.70: Mesh density on the mirroring surface ($z=0\text{ m}$) prior and after the first and second h-refinement (hexahedral enclosure with trapezoidal bases - second sub-case).

Figure 8.71 contains the distributions of dimensionless incident radiative heat flux along the A-B line (Figure 8.63) for the three different values of absorption coefficient with the initial and the refined

meshes, compared with the corresponding exact ones of Murthy and Mathur [Mur98c]. They were extracted after radiative intensity residual was decreased at least five orders of magnitude; approximately twelve seconds per iteration were required for the initial mesh, while after the first and second grid adaptation each iteration lasted about thirty and sixty seconds respectively. As far as the accuracy improvement is concerned, no obvious difference can be identified between the runs with absorption coefficient equal to 0.1 m^{-1} . For the remaining ones the situation is different; the difference between the initial and the refined grids is evident, revealing h-refinement potential to produce more accurate solutions, without the need to start from the very beginning the construction of a new finer grid. Although the improvement, provided by grid adaptation, isn't so important, compared to the one obtained by the second-order accurate schemes, it can be proved a viable tool, depending on the examined problem.

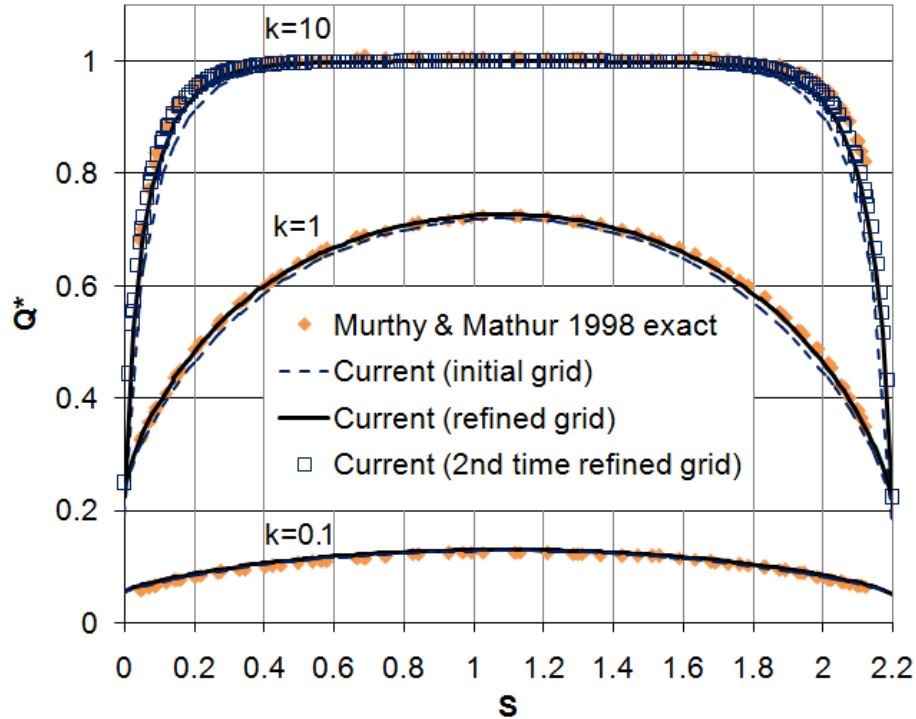


Figure 8.71: Distributions of dimensionless incident radiative heat flux along the A-B line for three different values of absorption coefficient, employing the initial and the refined meshes (hexahedral enclosure with trapezoidal bases - second sub-case).

8.10. A cubic enclosure with three baffles

In this test case radiative heat transfer in a cubic enclosure with three baffles is considered [Coe98, Lyg13b, Lyg14g]; its dimensions are illustrated in Figure 8.72. It is filled with an absorbing and emitting but non-scattering medium ($k_a=0.1 \text{ m}^{-1}$, $\sigma_s=0 \text{ m}^{-1}$) with a constant heating power E_m equal to 10 W/m^3 . Its walls as well as the baffles are assumed black ($\varepsilon_w=1$) with a constant heating energy equal to unity ($E_w=1 \text{ W/m}^2$), except for the faces normal to the z -direction, at which symmetry boundary conditions are implemented; radiative intensity on all the surfaces is implicitly obtained.

The utilized mesh consists of 46,198 nodes, 58,624 tetrahedra and 66,024 prisms, while for parallel processing it was decomposed in two sub-domains; Figure 8.73 illustrates the employed grid as well as the boundary density on one of its mirroring surfaces. Angular discretization was obtained with twenty four azimuthal and four polar angles, while the effects, derived by the overlapping of control angles and faces of control volumes, were subdued by the pixelation method. In order to implement the developed multigrid methodology, three coarser meshes were generated via isotropic agglomeration, at each of which the included DoF's were reduced approximately four times; in Figure 8.74 the boundary density of the initial and agglomerated control volumes is presented. Similarly, angular agglomeration was

performed, deriving a coarser directional resolution, including twelve azimuthal and two polar angles. The solution of time-dependent RTE with the Runge-Kutta method, applying a first- and second-order accurate spatial scheme coupled with Min-mod limiter, was performed on a workstation equipped with an AMD FX^(tm)-8120 eight-core processor at 3.1 GHz; besides multigrid simulations, single-grid ones were performed for the evaluation of the corresponding scheme. The parameters of these runs are included in Table 8.19.

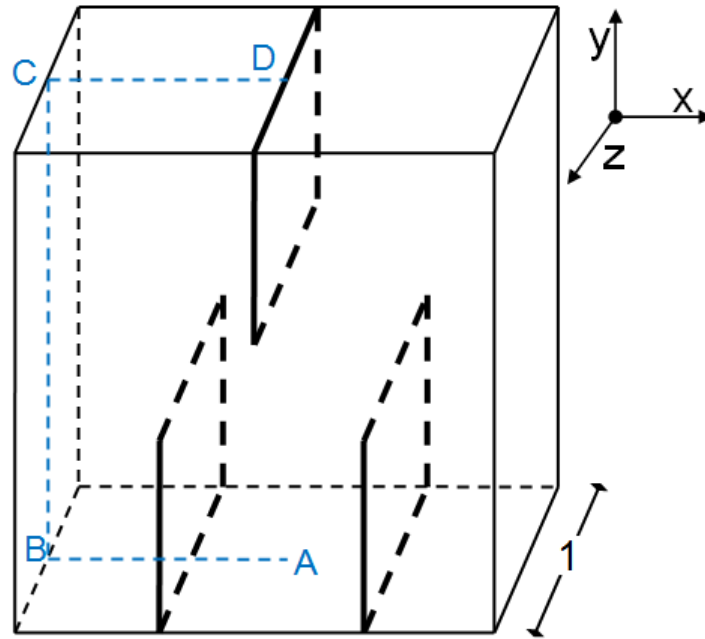


Figure 8.72: Dimensions of cubic enclosure with three baffles.

Figure 8.75 illustrates the contours of average radiative heat flux G derived by the run with the second-order scheme, while Figure 8.76 includes the obtained distributions of incident radiative heat flux along the ABCD line (Figure 8.72) for runs with both schemes, compared with the corresponding computed with Z-M (Zone-Method) ones of Coelho et al. [Coe98]. As expected, the results extracted with the higher-order method are more close to the reference ones, especially along the line AB, comparing to those obtained by the first-order accurate simulation.

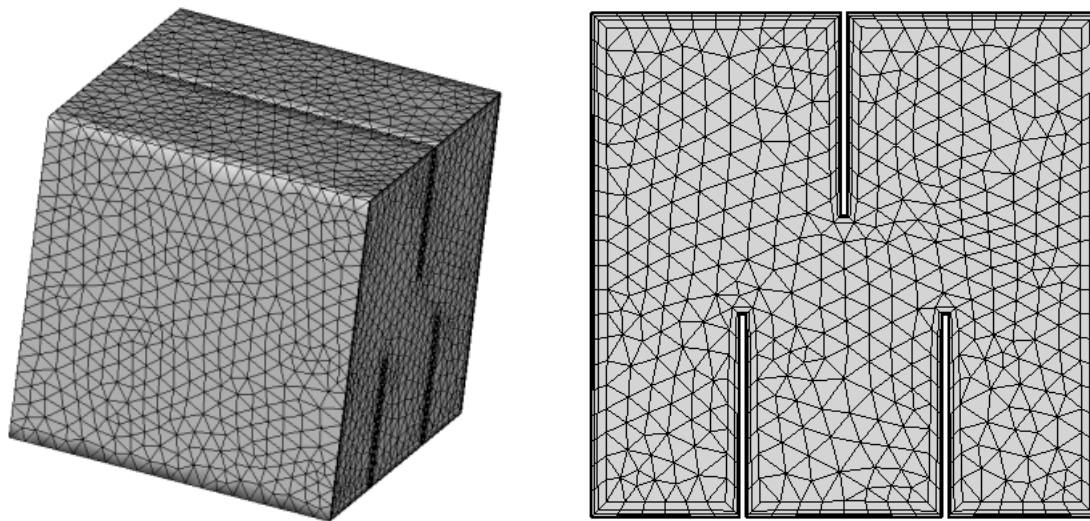


Figure 8.73: Employed mesh and its density on one of its symmetry surfaces (cubic enclosure with three baffles).

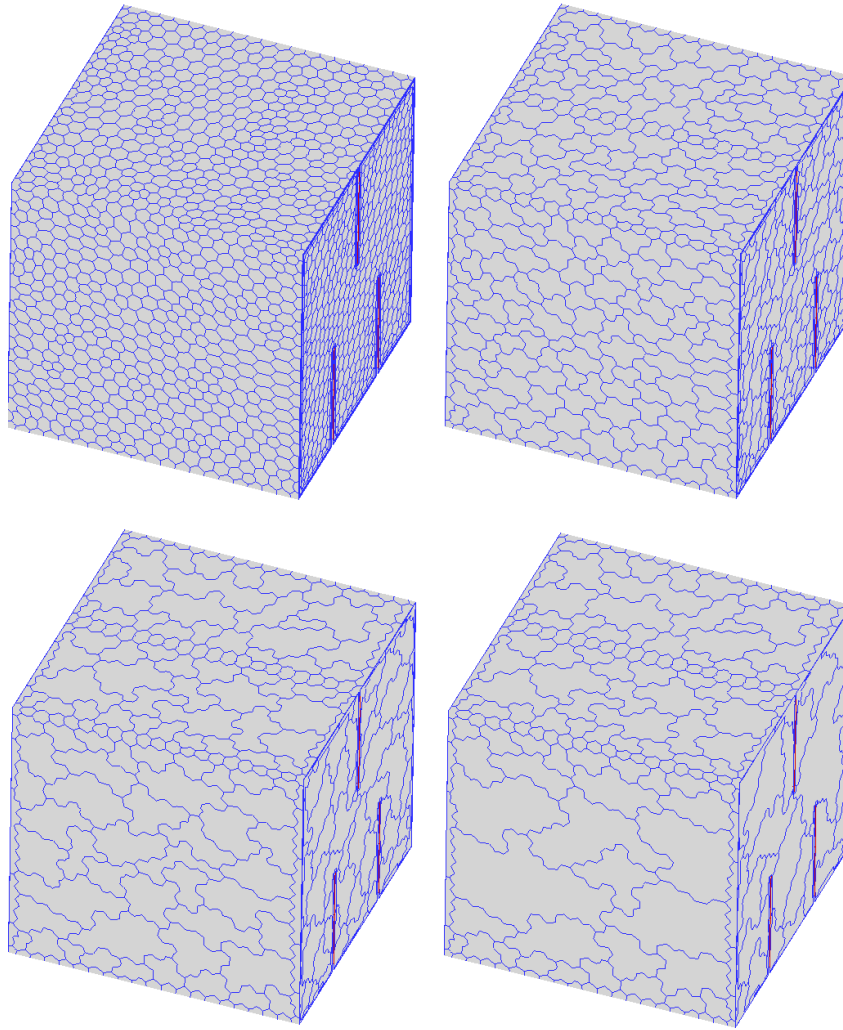


Figure 8.74: Initial and agglomerated control volume grids (cubic enclosure with three baffles).

Table 8.19: Parameters of simulation (cubic enclosure with three baffles).

Parameters	
Absorption coefficient	0.1 m^{-1}
Scattering coefficient	0 m^{-1}
Wall emissivity	1
Grid density	46,198 nodes, 58,624 tetrahedra and 66,024 prisms
Number of partitions	2
Number of spatial multigrid levels	1-4
Angular resolution	24 azimuthal and 4 polar control angles
Number of angular multigrid levels	1-2
Iterative scheme	Runge-Kutta method
Computational system	Workstation with an AMD FX ^(tm) -8120 eight-core processor at 3.1 GHz

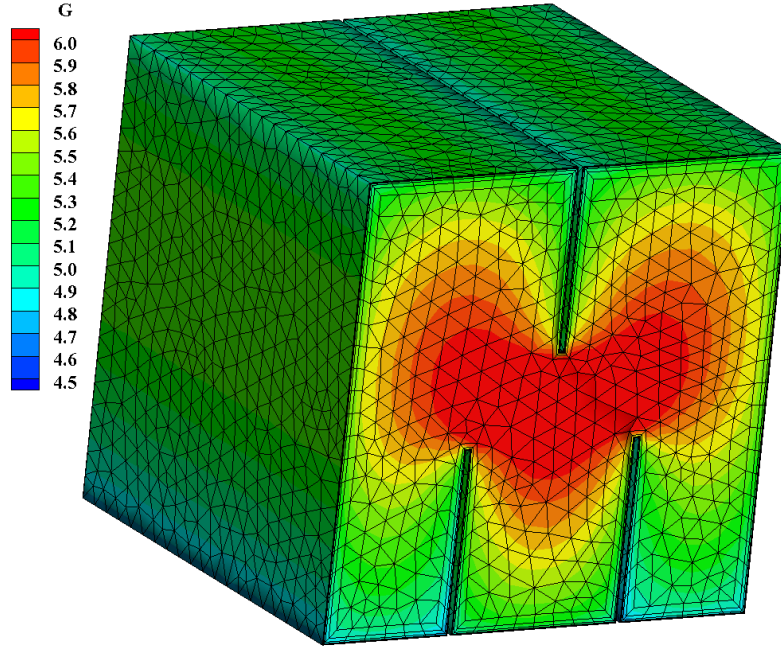


Figure 8.75: Contours of average radiative heat flux G , obtained by the second-order accurate run (cubic enclosure with three baffles).

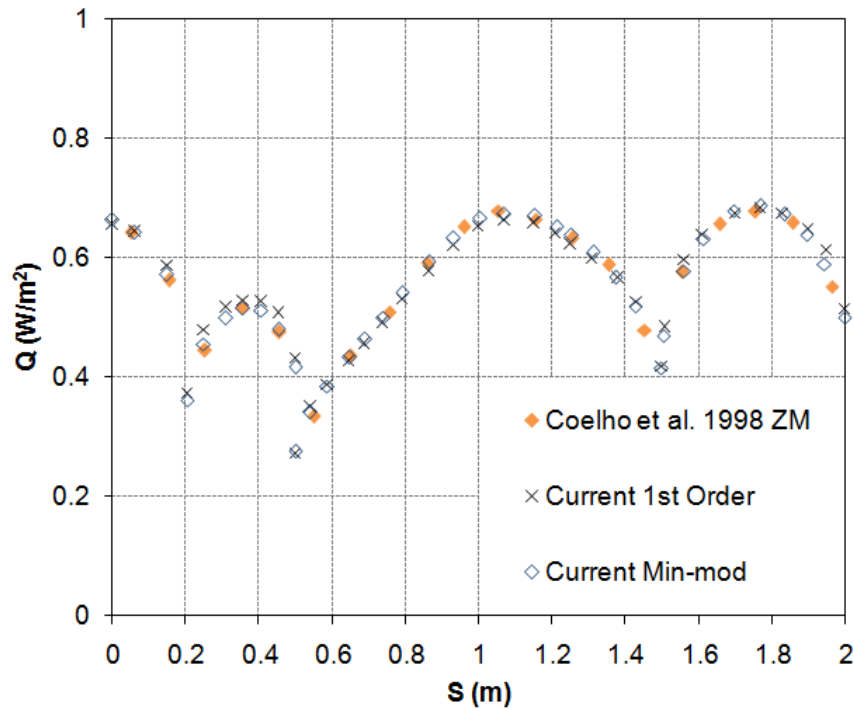


Figure 8.76: Distributions of incident radiative heat flux along the ABCD line for different spatial schemes (cubic enclosure with three baffles).

Only spatial agglomeration multigrid method (S4A1) was employed for the acceleration of the sub-problem with first-order spatial accuracy as the angular one was revealed to contribute negatively. Figure 8.77 includes the radiative intensity convergence history per number of iterations/cycles and computation time, compared with those of a single-grid run (S1A1); an iteration and temporal speed-up coefficient equal to ~ 2.65 and ~ 1.8 respectively was achieved for a final residual equal to $1.0E-6$. Different multigrid

schemes (S1A2, S4A1 and S4A2) were implemented along with the higher-order accurate spatial method, resulting in the corresponding convergence histories per iterations and time, presented in Figure 8.78; the nested spatial/angular scheme results in a slightly faster solution than the simple spatial one, while similarly the angular method in a slightly faster run than the single-grid one. Thus, the small contribution of the angular multigrid method is confirmed once more for test cases with no medium scattering and wall reflection.

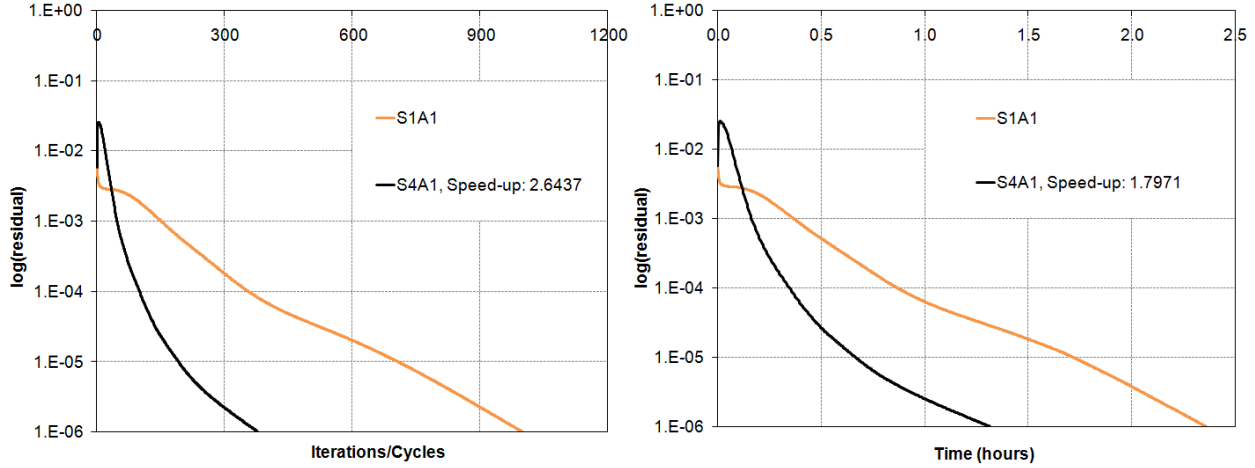


Figure 8.77: Radiative intensity convergence history per iterations/cycles and time (first-order scheme, cubic enclosure with three baffles).

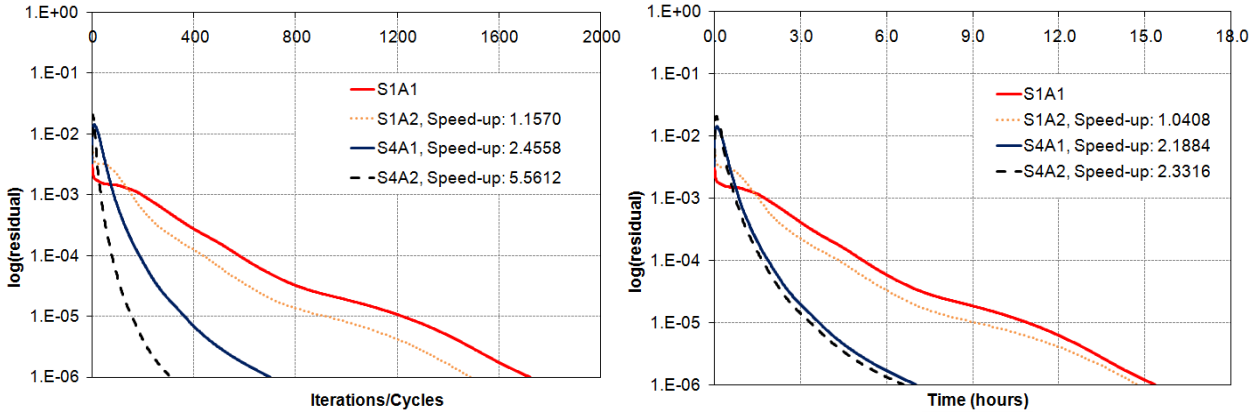


Figure 8.78: Radiative intensity convergence history per iterations/cycles and time (second-order scheme, cubic enclosure with three baffles).

An additional (multigrid and single-grid) simulation was made with the aforementioned enclosure, assuming the same parameters, except for the value of wall emissivity of walls ($\varepsilon_w=0.8$) and baffles ($\varepsilon_{wb}=0.6$) (gray walls). Figure 8.79 contains the obtained distributions of incident radiative heat flux along the ABCD line for runs with both schemes (first- and second-order accurate with Min-mod limiter), compared to the corresponding ones computed with Z-M (Zone-Method) of Coelho et al. [Coe98]; similarly to previous runs (with black walls), the results derived by the higher-order method agree more with the reference ones. Only the spatial multigrid method (S4A1) was employed again along with the first-order accurate scheme due to the absence of medium scattering and wall reflection. Figure 8.80 illustrates the radiative intensity convergence history per number of iterations/cycles and wall-clock computation time, compared to those of a single-grid run (S1A1); an increased time speed-up coefficient (~ 2.34) for a final residual equal to $1.0E-6$ was obtained, compared to the corresponding one of the sub-case with black walls (~ 1.80), due to the increased radiation transfer between the solid control angles,

imposed by the increased values of wall reflectivity. Different multigrid schemes (S1A2, S4A1 and S4A2) were applied along with the higher-order method, deriving the corresponding convergence histories per iterations and time, presented in Figure 8.81. Similarly to the runs with black walls, the contribution of angular acceleration method appears to be relatively unimportant; nevertheless, an increased time speed-up coefficient is succeeded for all multigrid schemes, with a maximum value of ~ 3.02 obtained by the nested four-level spatial/two-level angular scheme (S4A2).

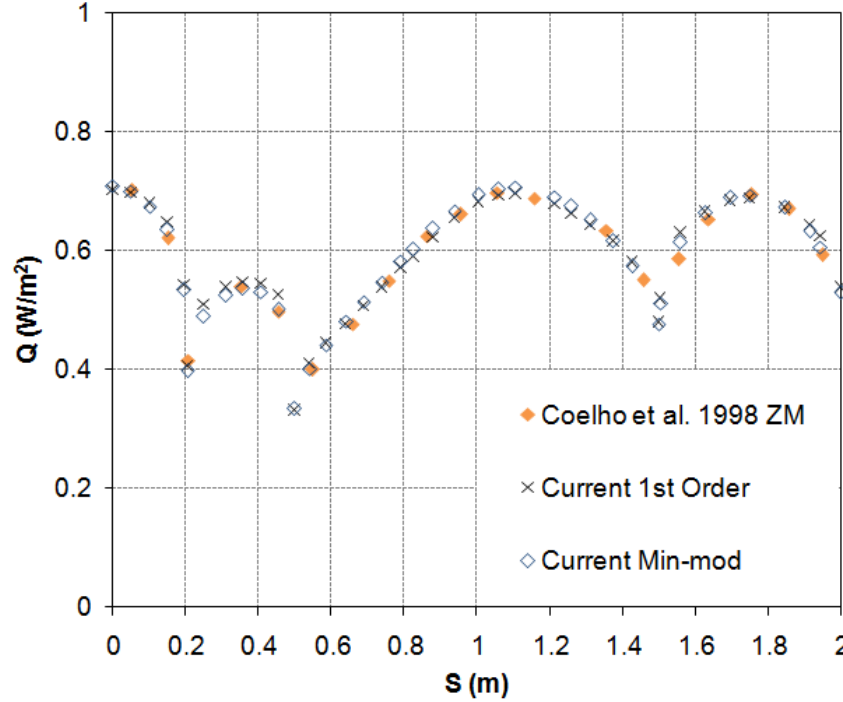


Figure 8.79: Distributions of incident radiative heat flux along the ABCD line for different spatial schemes (gray walls, cubic enclosure with three baffles).

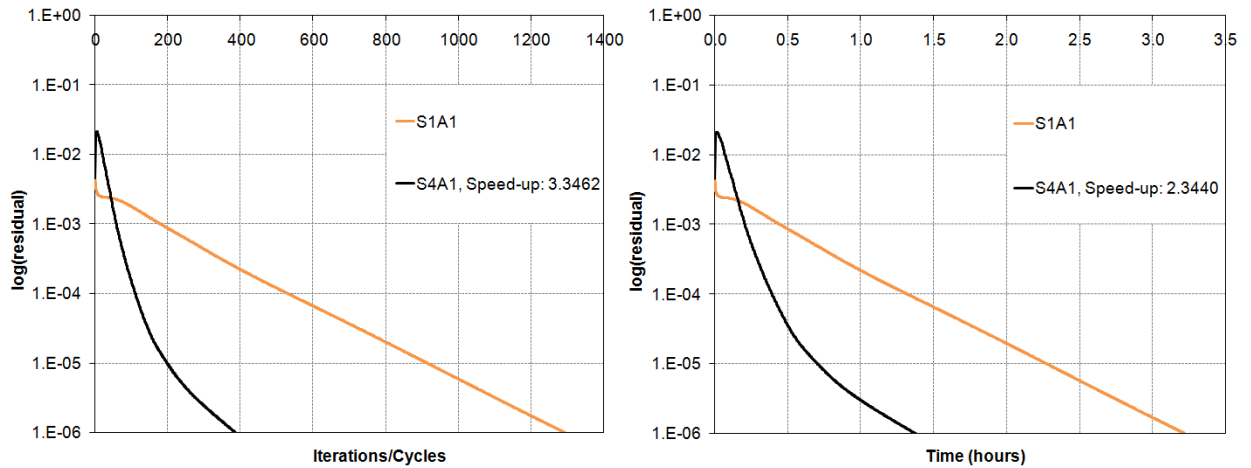


Figure 8.80: Radiative intensity convergence history per iterations/cycles and time (first-order scheme, gray walls, cubic enclosure with three baffles).

8.11. An annular sector

Another problem examined mainly for the validation of the developed multigrid methodology, concerns radiative heat transfer in a black ($\epsilon_w=1$) annular sector with thickness of 0.5 m , length equal to unity ($L=1\text{ m}$) and angular extent ϕ of 60° , as illustrated in Figure 8.82 [Bae98b, Lyg12a, Lyg14g]. The

medium as well as all the walls are considered cold ($T_m=0$ K, $T_w=0$ K), except for the inner surface, maintained at constant temperature of 100 K; boundary conditions are implemented implicitly. Two sub-cases were examined regarding different medium features; for the first one the medium is assumed absorbing and emitting, but non-scattering ($k_a=1$ m⁻¹, $\sigma_s=0$ m⁻¹, $\omega=0$), while for the second one it is assumed to be isotropically scattering ($k_a=0.5$ m⁻¹, $\sigma_s=0.5$ m⁻¹, $\omega=0.5$). The employed grid consists of 49,925 nodes, 216,595 tetrahedra and 20,460 prisms, the latter positioned at the outer surface of the annular sector; Figure 8.83 illustrates the utilized grid as well as the boundary density on one of its bases. For parallel solution of the time-dependent RTE with the Runge-Kutta method and a first-order accurate spatial scheme on a workstation with an AMD FX^(tm)-8120 eight-core processor at 3.1 GHz, it was divided in two partitions. The angular discretization was obtained with sixteen azimuthal and eight polar angles, while pixelation approach was applied for the alleviation of the overhang problem. Besides single-grid simulations, multigrid ones were performed for the evaluation of the corresponding scheme. Therefore, two coarser meshes were generated via isotropic agglomeration; the number of DoF's was decreased approximately five times for the second agglomerated level and four additional times for the final one. In Figure 8.84 the initial and agglomerated control volume grids are presented. Similarly, angular agglomeration was performed, deriving a coarser directional resolution, including eight azimuthal and four polar angles. The parameters of this case are contained in Table 8.20.

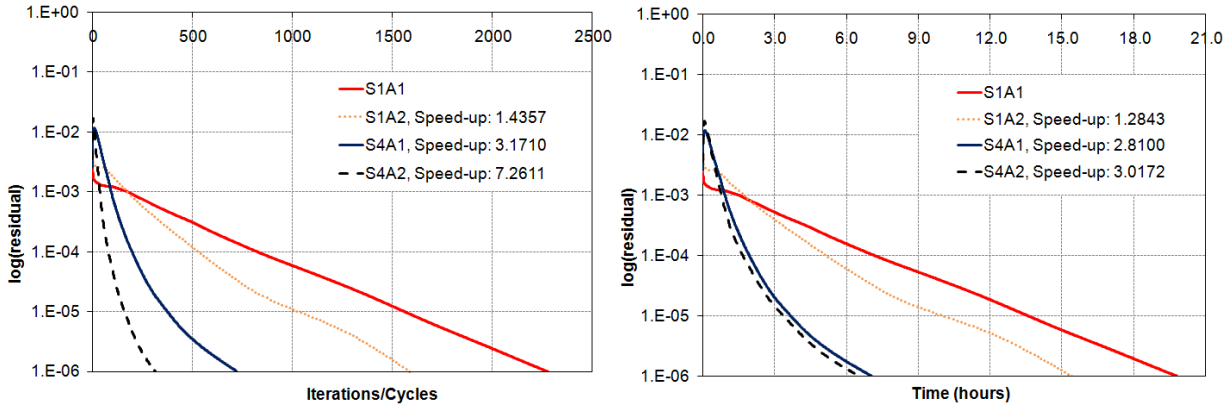


Figure 8.81: Radiative intensity convergence history per iterations/cycles and time (second-order scheme, gray walls, cubic enclosure with three baffles).

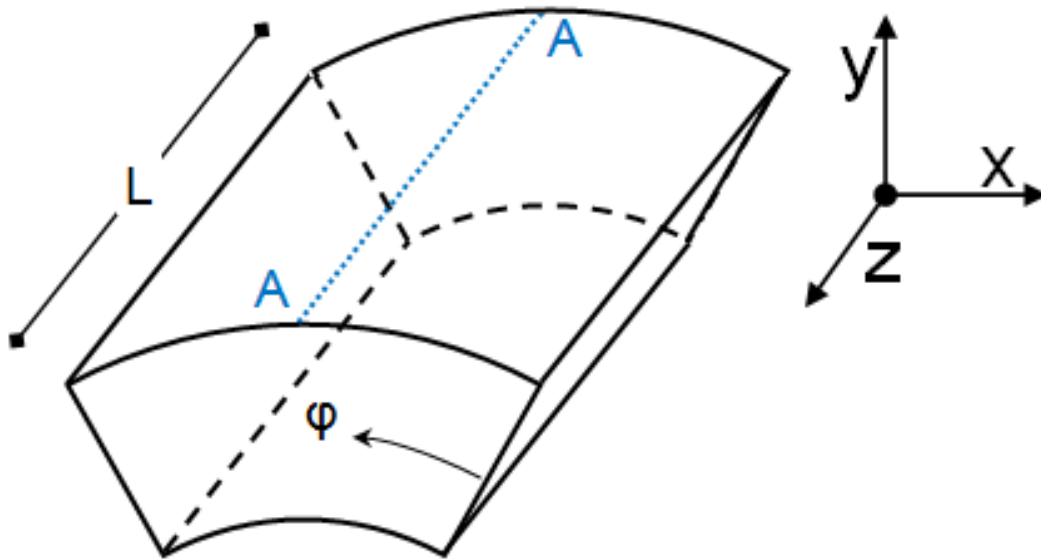


Figure 8.82: Dimensions of the annular sector.

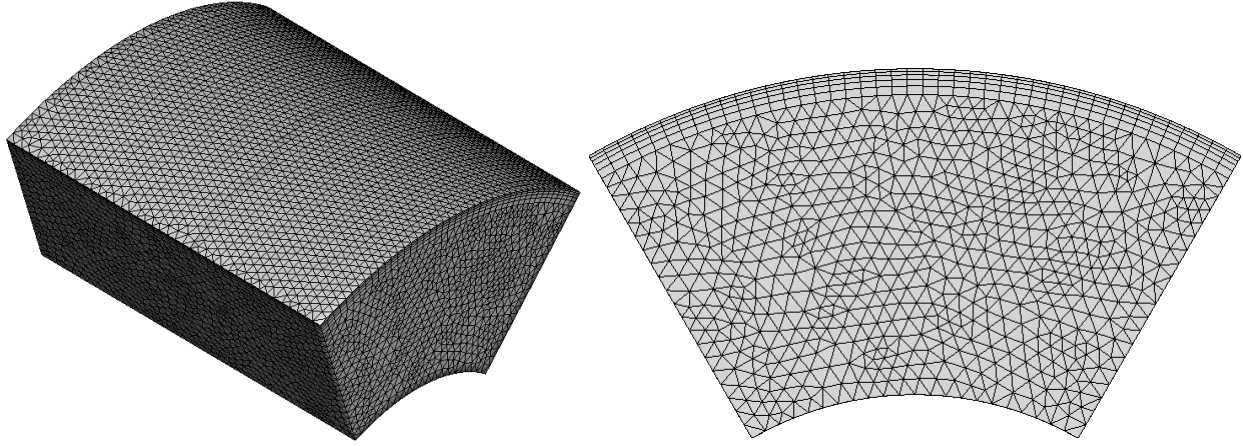


Figure 8.83: Utilized mesh and its density on one of its bases (annular sector).

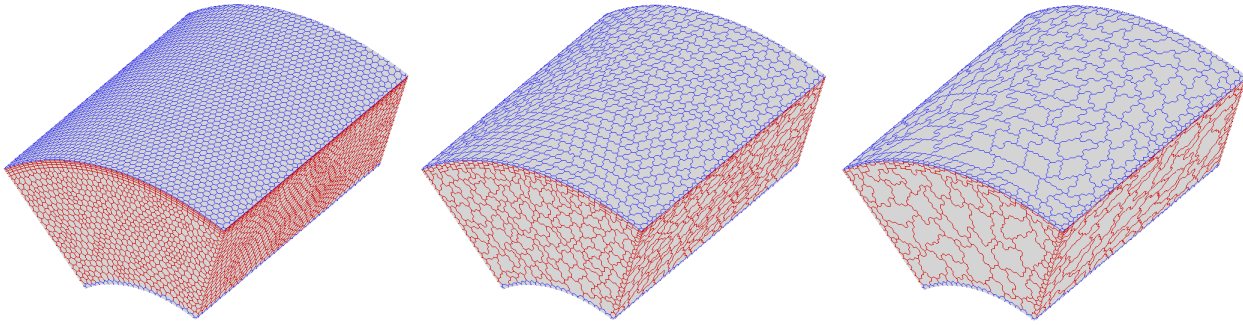


Figure 8.84: Initial and agglomerated control volume grids (annular sector).

Table 8.20: Parameters of simulation (annular sector).

Parameters	
Absorption coefficient	$0.5-1 \text{ m}^{-1}$
Scattering coefficient	$0-0.5 \text{ m}^{-1}$
Wall emissivity	1
Grid density	49,925 nodes, 216,595 tetrahedra and 20,460 prisms
Number of partitions	2
Number of spatial multigrid levels	1-3
Angular resolution	16 azimuthal and 8 polar control angles
Number of angular multigrid levels	1-2
Iterative scheme	Runge-Kutta method
Computational system	Workstation with an AMD FX ^(tm) -8120 eight-core processor at 3.1 GHz

Figure 8.85 contains the obtained distributions of dimensionless incident radiative heat flux along the half A-A line (Figure 8.82) for scattering albedo equal to 0 and 0.5. Although only a first-order accurate spatial scheme was applied, the extracted results agree sufficiently with the corresponding ones of the cell-centered finite-volume solver of Baek et al. [Bae98b]. In Figure 8.86 the contours of average radiative heat flux G at the middle of the annular sector ($z=0.5$) are illustrated for the sub-case concerning a medium with scattering albedo 0.5.

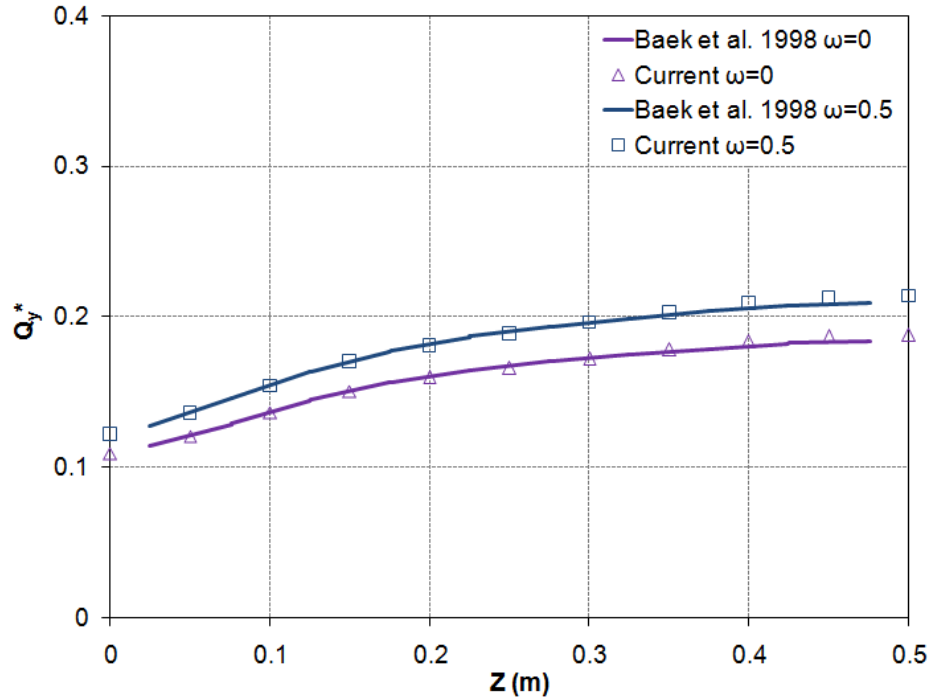


Figure 8.85: Distributions of dimensionless incident radiative heat flux along the half A-A line for different values of scattering albedo (annular sector).

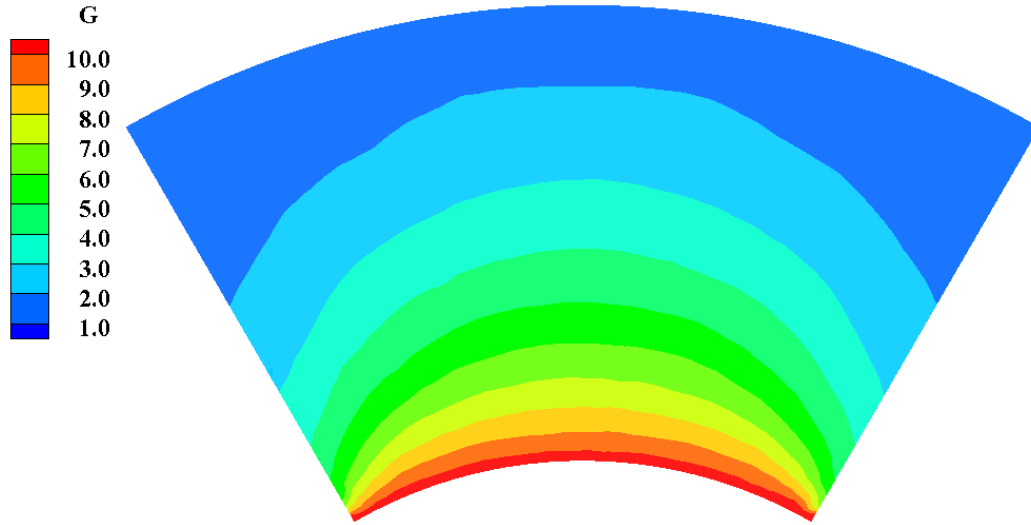


Figure 8.86: Contours of average radiative heat flux G at the middle of the sector ($z=0.5$) for the sub-case considering a medium with scattering albedo 0.5 (annular sector).

Considering the absence of scattering behaviour of the medium and reflection of the walls, for the first sub-case, only the spatial agglomeration multigrid method was employed with different number of levels (S2A1 and S3A1). Figure 8.87 includes their convergence history per number of iterations/cycles and computation time, compared with the corresponding one of the single-grid simulation (S1A1). As the S2A1 scheme derived a temporal speed-up coefficient equal to ~ 1.68 and the S3A1 a corresponding coefficient equal to ~ 2.35 , it is confirmed that extra acceleration is gained with addition of an extra coarser level to the multigrid procedure.

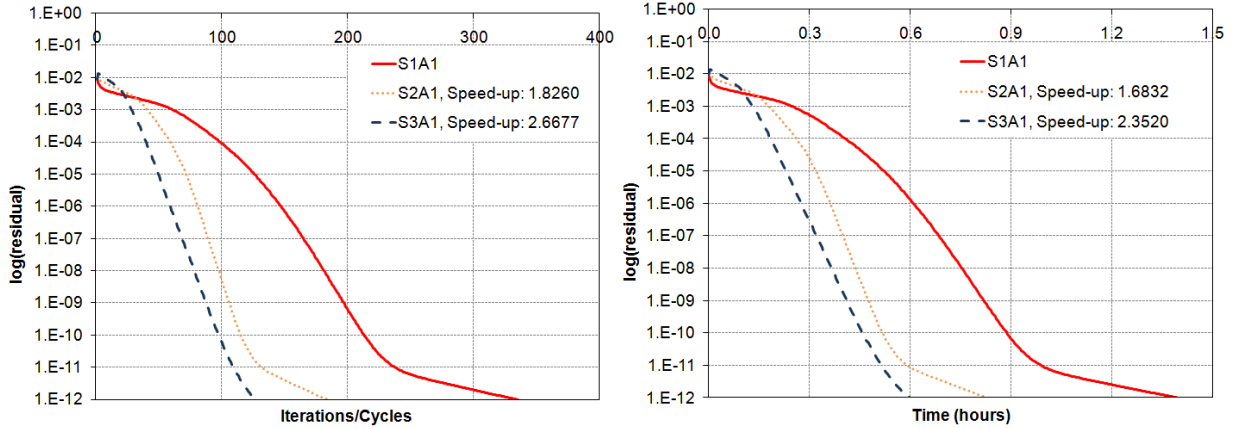


Figure 8.87: Radiative intensity convergence history per iterations/cycles and time (only absorbing and emitting medium, annular sector).

Unlike the previous sub-case, different multigrid schemes, namely, S1A2, S3A1 and S3A2, were employed for the second sub-problem with the isotropically scattering medium. Figure 8.88 includes their convergence history per number of iterations/cycles and computation time, compared with the corresponding one of the single-grid simulation (S1A1). According to the extracted temporal speed-up coefficients for the final residual $1.0E-12$, the nested scheme (~ 2.28) led to a slightly faster solution than the only spatial one (~ 2.24); similarly the solution with the only angular method (~ 1.09) is slightly accelerated, compared to the single-grid one. However, for a residual value $1.0E-6$, which is more than sufficient for practical radiative heat transfer applications, the temporal speed-up coefficients are re-computed as ~ 1.31 for S1A2, ~ 2.35 for S3A1 and ~ 2.72 for S3A2, demonstrating a satisfactory acceleration obtained by the angular agglomeration methodology for both the angular and the nested multigrid scheme.

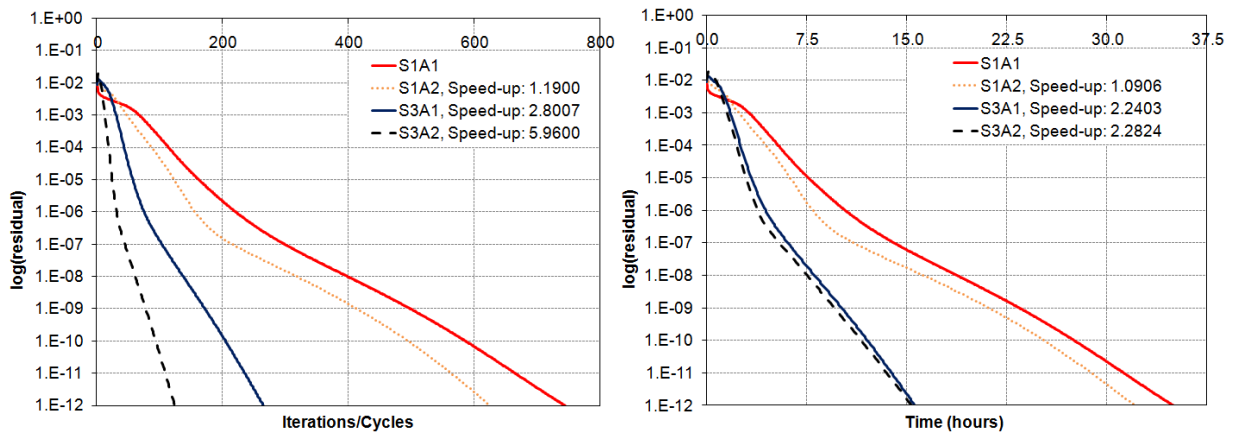


Figure 8.88: Radiative intensity convergence history per iterations/cycles and time (isotropically scattering medium, annular sector).

8.12. A tetrahedral enclosure

The last test case considers radiative heat transfer in a tetrahedral enclosure with black ($\varepsilon_w=1$) and cold ($T_w=0$ K) walls, as this is illustrated in Figure 8.89 [Mur98c, Lyg12a, Lyg14b]. It is filled with an absorbing and emitting, but non-scattering, ($k_a=1$ m⁻¹, $\sigma_s=0$ m⁻¹) medium, held at constant temperature 100 K. The utilized grid includes 196,847 nodes and 1,119,456 tetrahedra, while for parallel processing it was divided in two partitions; Figure 8.90 presents the employed grid as well as its density on the surface with A-A line. Although such a fine grid wasn't necessary as far as the solution accuracy is concerned, it was employed in order to amplify the effect of the developed multigrid method and evaluate its

performance in large resolutions. Angular discretization was obtained with sixteen azimuthal and eight polar angles, while pixelation method subdued overhang problem effects. Radiative intensity on all the walls was obtained implicitly. The solution of time-dependent RTE with the Runge-Kutta method, applying a first-order accurate spatial scheme, was performed on a workstation equipped with an AMD FX^(tm)-8120 eight-core processor at 3.1 GHz.

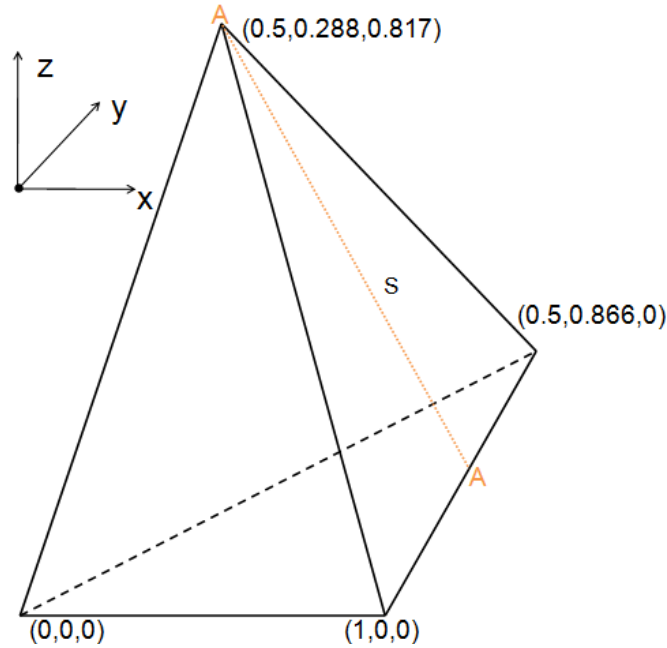


Figure 8.89: Dimensions of tetrahedral enclosure.

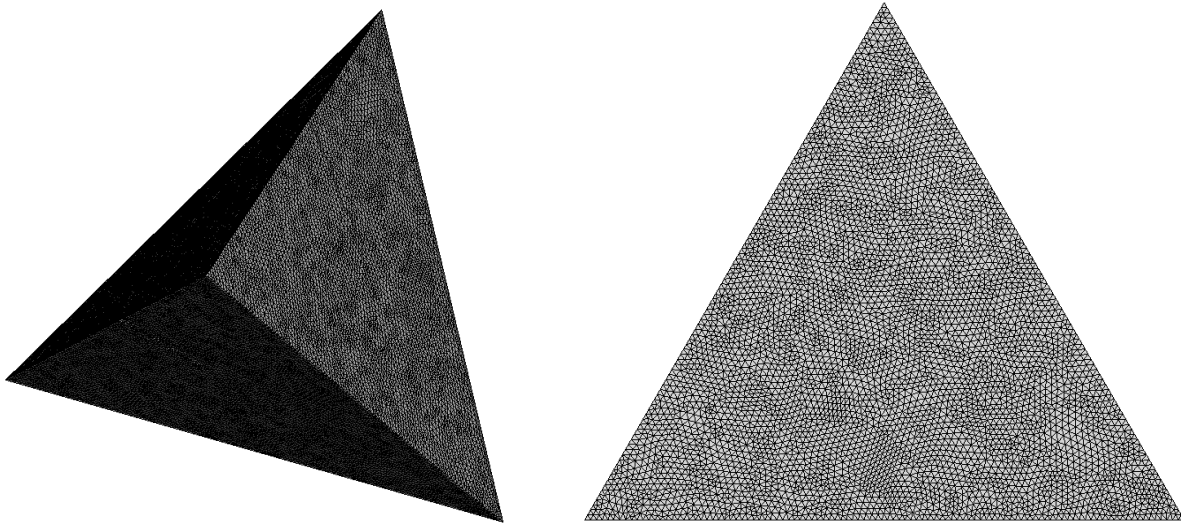


Figure 8.90: Employed grid and its density on the surface with A-A line (tetrahedral enclosure).

In order to implement the developed multigrid methodology, two coarser meshes were generated via isotropic agglomeration; in Figure 8.91 the boundary density of the initial and agglomerated control volumes is presented. Besides multigrid simulations, a single-grid one was performed for the evaluation of the corresponding scheme. The parameters of those runs are summarized in Table 8.21.

In Figure 8.92 the distribution of dimensionless incident radiative heat flux along the A-A line (Figure 8.89) is illustrated, compared with the corresponding one of Murthy and Mathur [Mur98c]; a

satisfactory agreement is identified between the obtained and reference results. Figure 8.93 contains the convergence history per iterations/cycles and time for all the employed spatial multigrid schemes, compared with this of the single-grid simulation; a maximum temporal speed-up coefficient equal just to 1.95 was obtained, due to the absence of medium's scattering behaviour and walls' reflection.

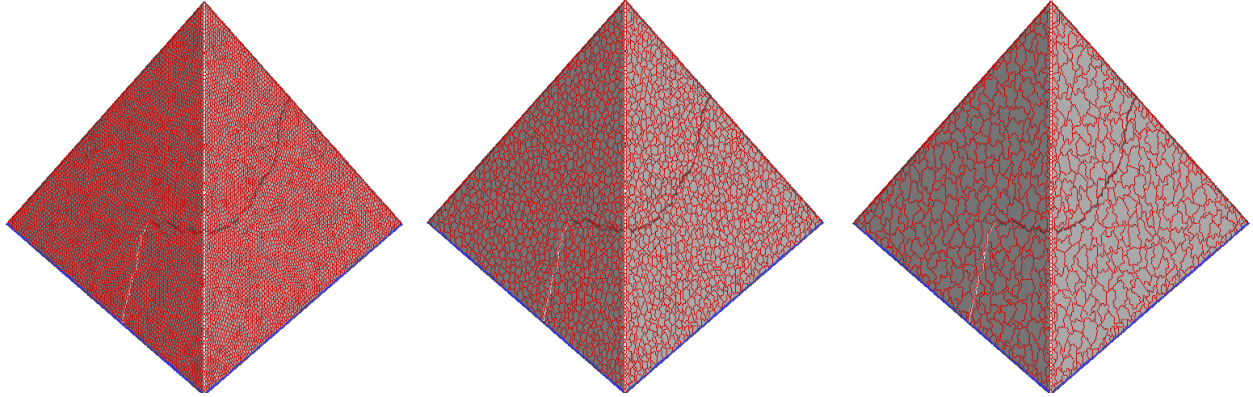


Figure 8.91: Initial and agglomerated control volume grids (tetrahedral enclosure).

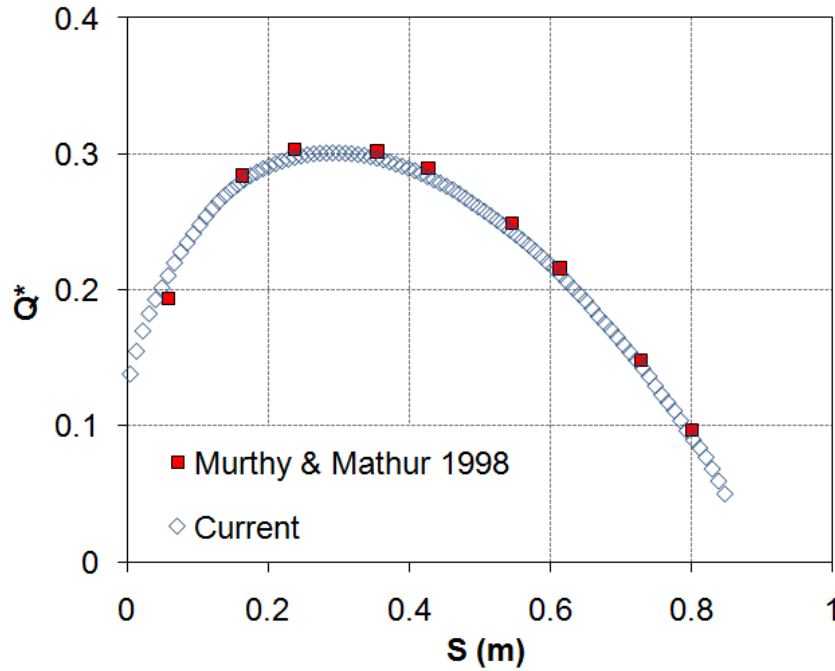


Figure 8.92: Distributions of dimensionless incident radiative heat flux along the A-A line (tetrahedral enclosure).

In order to further evaluate the incorporated multigrid methodology, additional (multigrid and single-grid) simulations were performed for the aforementioned enclosure, assuming though a purely anisotropically scattering ($k_a=0 \text{ m}^{-1}$, $\sigma_s=1 \text{ m}^{-1}$) and cold ($T_m=0 \text{ K}$) medium as well as black ($\varepsilon_w=1$) and cold ($T_w=0 \text{ K}$) walls, except for the face including the A-A line, which is held at constant temperature of 100 K . Angular agglomeration was also implemented, deriving a coarser directional resolution with eight azimuthal and four polar angles. Figure 8.94 illustrates the extracted convergence histories along with the corresponding speed-up rates per number of iterations/cycles and computation time for all the employed multigrid runs, namely, S1A2, S3A1 and S3A2, compared with those of the single-grid simulation. The maximum temporal coefficient of ~ 3.07 was derived by the nested S3A2 scheme, revealing once more the

capability of the algorithm for enhanced acceleration of the solution procedure when a scattering medium is considered.

Table 8.21: Parameters of simulation (tetrahedral enclosure).

Parameters	
Absorption coefficient	1 m^{-1}
Scattering coefficient	0 m^{-1}
Wall emissivity	1
Grid density	196,847 nodes and 1,119,456 tetrahedra
Number of partitions	2
Number of spatial multigrid levels	1-3
Angular resolution	16 azimuthal and 8 polar control angles
Number of angular multigrid levels	1
Iterative scheme	Runge-Kutta method
Computational system	Workstation with an AMD FX ^(tm) -8120 eight-core processor at 3.1 GHz

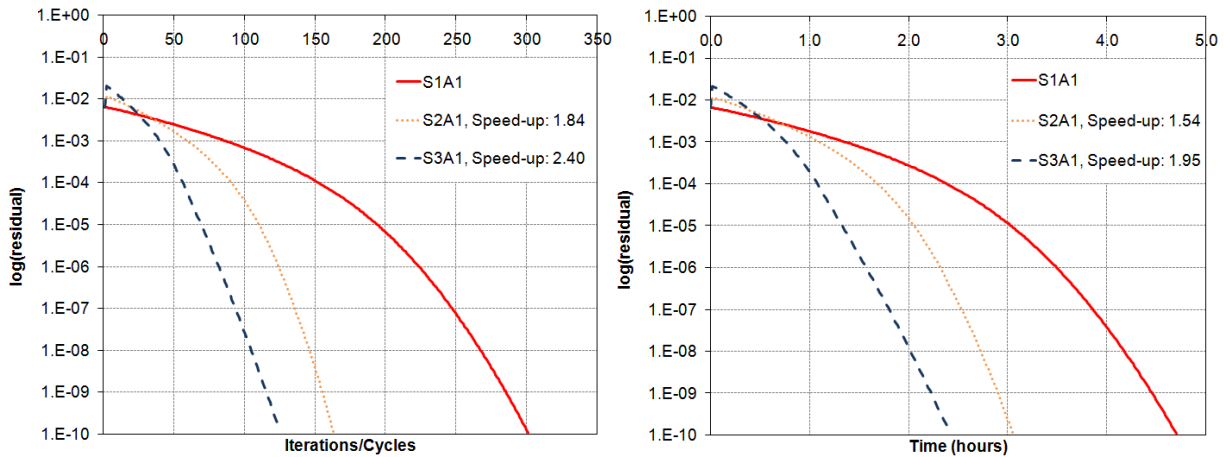


Figure 8.93: Radiative intensity convergence history per iterations/cycles and time (only absorbing medium, tetrahedral enclosure).

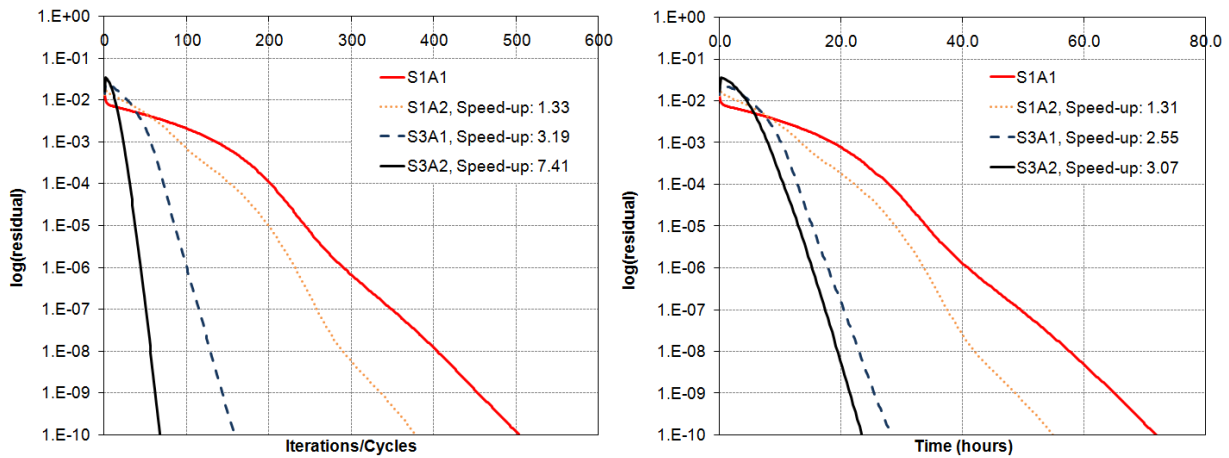


Figure 8.94: Radiative intensity convergence history per iterations/cycles and time (anisotropically scattering medium, tetrahedral enclosure).

"Intentionally left blank"

9. Conclusions

9.1. Summary

In this thesis the development and validation of methodologies allowing for the numerical solution of steady-state compressible fluid flow, and radiative heat transfer problems was reported. For the representation of computational fields, three-dimensional unstructured hybrid grids were employed, composed of tetrahedral, prismatic and pyramidal elements, mainly to resolve effectively the severe anisotropy of boundary layer in viscous fluid flows [Sor03, Lyg14a, Lyg15]. Unlike flow simulations, this type of mesh discretized with a node-centered finite-volume scheme is rarely used in radiative heat transfer predictions comparing to tetrahedral grids along with a cell-centered method [Kim88, Bae98b, Mur98c, Kim01, Sal04, Kim10]; nevertheless, the examined in this study numerical runs with such meshes derived equally accurate solutions [Lyg12b, Lyg13c]. Moreover, their capability to include highly stretched elements at selected areas of the examined grid, e.g., at boundary regions with large radiative intensity gradients, resulted in utilization of relatively coarser grids and consequently in considerable computational savings, maintaining though the desired accuracy [Lyg12b].

Fluid flow numerical results, obtained by the implementation of RANS PDE's along with appropriate turbulence models, namely, $k-\varepsilon$ (in three versions), $k-\omega$ and SST [Lau74, Saf74, Koo00, Men03a, Lyg11], compare sufficiently well with the corresponding experimental data and computational results of reference solvers, indicating the potential of the proposed methodology for such simulations. The previous statement derives mainly from the satisfactory solutions extracted for the more complicated test cases performed, concerning fully turbulent flow over aircrafts (DLR-F6 and CRM) with the presence of considerable flow discontinuities, although coarse meshes were utilized [Lyg13a, Lyg14a, Lyg14e, Lyg15]. The incorporated high-order accurate spatial scheme coupled with the slope limiters of Van Albada-Van Leer, Min-mod and Barth-Jespersen [VanA82, Swe84, Bar89, Bar92, Bla01], appears to enhance the present algorithm with improved accuracy by alleviating sufficiently numerical diffusion as well as with the capability to utilize relatively coarse grids along with significant computational economy; depending on the test case, the appropriate limiting function should be selected, e.g. min-mod limiter is not appropriate for problems involving sharp shock waves, due to its smoothing attitude. Although the nodal-averaging scheme [Bla01], employed for the computation of flow and turbulence model variables' gradients at control volumes' interfaces, is assumed generally less accurate than the element-based one, using edge-dual volumes [Kal96, Kal05a], it derived equally accurate but much more computationally efficient solutions, utilizing the edge-based data structure of the algorithm [Lyg14c]. It is this structure that allows nodal-averaging scheme, unlike the element-based method, to be applied to the agglomerated grids also, composed of polyhedral elements for the implementation of the developed multigrid methodology [Lyg14c]. Implicit schemes, such as those incorporated in the proposed algorithm employing the Jacobi or the Gauss-Seidel algorithm [Ven95, Kou03], are considered more effective in terms of efficiency, comparing to the corresponding explicit ones, as they allow for the utilization of larger CFL numbers; nevertheless, considering the available finite computing resources, the included second-order temporal accurate four-stage Runge-Kutta (RK(4)) method [Lal88a] revealed to be a valuable tool, due to its reduced requirements for memory storage and communication load in a parallel computational environment.

Similarly to fluid flow, radiative heat transfer numerical results, considering their prediction in general enclosures through absorbing, emitting and isotropically or anisotropically scattering gray media, obtained either with the time-dependent or steady (non time-dependent) RTE, agree sufficiently well with the exact solutions or computed ones of reference solvers. The incorporated second-order accurate spatial scheme with or without limiting function (Van Albada-Van Leer or Min-mod) [VanA82, Swe84, Lyg13c], was proved to be an important enhancement of the algorithm, alleviating sufficiently false scattering [Cha93, Coe02, Cap10], especially at boundary surfaces' regions with large radiative intensity gradients, and consequently improving the accuracy of the final steady-state solution even in very coarse spatial resolutions [Lyg13b, Lyg13c]. Among the developed second-order accurate schemes (simple or

together with Van Albada-Van Leer or Min-mod limiter), the unlimited former one appeared to produce a flux over- or under-estimation (depending on the test case), due to the unbounded reconstructed values of radiative intensity, while no significant difference was distinguished between the other two limited ones [Lyg13b, Lyg13c]. The main drawback of these schemes is the requirement for increased computational effort comparing to the first-order one; it can be assumed relatively trivial, considering the resulting accuracy improvement, especially for coarse grids. The non time-dependent RTE, solved with iterative approximations and a second-order scheme appeared to lead to spurious oscillations in some test cases [Lyg13c]; the remedy to this shortcoming revealed to be the employment of the time-dependent RTE [Hun11] along with the second-order accurate in time four-stage Runge-Kutta (RK(4)) method [Lal88a, Lyg13c]. Any overhang conditions, caused by the utilization of unstructured grids and the division of directional domain in finite solid control angles [Cha94a, Kim01], were sufficiently mitigated by the incorporated methods, namely, the bold approximation and the pixelation method [Lyg13b]. The latter approach, being of course more accurate, is implemented by most of the researchers only at solid control angles-control volume faces, for which significant overhang is exhibited [Mur98a, Kim01, Kim05b, Lyg13c]. Unlike this common practice, in this study it is employed to the total number of nodes and control angles in a pre-computation stage, considering that the values of method's coefficients, as well as those of directional weights, do not change during the solution of the RTE; in that way a reduced computational cost per iteration is succeeded [Lyg13b, Lyg13c]. Implicit treatment of boundary conditions is another feature of the proposed methodology; besides being a technique more close to those employed in CFD than the explicit approach, e.g. in free-slip, inlet and outlet surfaces, it avoids the requirement of the latter one for *a priori* finer computational grids at these regions, thus resulting in significant computational savings [Lyg13c]. In addition, it allows for the implementation of mirroring boundary conditions, resembling specularly reflecting walls, and consequently for the utilization of smaller grids with significantly reduced numbers of DoF's. A drawback of this methodology is that the mirroring direction has to be selected to coincide with a discrete one; in order to implement it on more directions, an interpolation technique should be employed [Liu00]. Nevertheless, as the symmetry planes are usually designed parallel to the axes' planes (as in this work), this shortcoming appears to be relatively unimportant [Lyg13c].

Significant contribution to the accuracy of the final steady-state solutions of both flow and radiative heat transfer problems was obtained from the implementation of the developed h-refinement methodology, enriching with more DoF's specific grid regions, selected either automatically or by the user [Lyg13c, Lyg14a, Lyg15]. For example, although relatively coarse initial grids were used in test cases concerning flow over the CRM aircraft and radiative heat transfer in a hexahedral enclosure with trapezoidal bases, the applied grid adaptation method resulted in accurate predictions of the surface pressure and dimensionless incident radiative heat flux distributions, respectively [Lyg14a, Lyg15]. As such, h-refinement was demonstrated to be a viable tool in test cases where important amounts of numerical diffusion and false scattering in flow and radiative heat transfer predictions, respectively, are important, or in problems faced first time for which no previous knowledge exists about the required spatial resolution [Loh92, Kal96, Lyg13c, Lyg14a, Lyg15]. Besides the accuracy improvement it entails, it results in significant computational savings, as it avoids the construction of a locally or globally finer mesh from the very beginning, being implemented during the solution procedure; it requires a few minutes to refine locally a grid with approximately 4,500,000 nodes, producing a new mesh with about 6,500,000 nodes (CRM WBHT aircraft) [Lyg14a, Lyg15].

Besides the effectiveness of the present methodology in terms of accuracy, the obtained numerical results demonstrated its satisfactory efficiency, derived by the incorporated acceleration techniques, namely, the edge-based data structure along with the local time-stepping scheme, the parallelization strategy based on the domain decomposition approach and MPI library functions, and the agglomeration multigrid method, employed in isotropic or directional (semi- or full-coarsening) spatial formulation for the flow solver and in spatial, angular or nested spatial/angular one for radiative heat transfer algorithm [Lyg14b, Lyg14c, Lyg14d, Lyg14f, Lyg14g]. Although unstructured grids entail significant geometric flexibility, they call for increased computational effort and memory requirements, comparing to structured

ones; therefore the utilization of a more sophisticated data structure with indirect addressing, such as the edge-based one employed with the present algorithm, becomes a prerequisite [Lyg12b]. In addition, parallel processing reduced considerably the required wall-clock computation time, especially in large-scale simulations including grids with millions of DoF's, such as in the test cases of DLR-F6 and CRM aircrafts [Lyg13a, Lyg14a, Lyg14e, Lyg15]. Considering the scalability results of the developed parallelization strategy for both the flow and radiative heat transfer solver, a linear relation between speed-up and number of employed processors was identified up to approximately sixteen processors; a deviation between the ideal and the current computations began at this point, caused by the increase of communication load with the increase of utilized processors [Lyg12b]. Nevertheless, it is assumed an acceptable deviation, which can be further improved if the proposed methodology is applied to larger scale problems with more DoF's [Lyg12b].

The last method, which contributed significantly to the improvement of the proposed algorithm's computational performance, is the parallel agglomeration multigrid scheme [Nis11, Lyg14b, Lyg14c, Lyg14d, Lyg14f, Lyg14g]. For the flow solver, it was employed in an isotropic or directional (semi- or full-coarsening) formulation, depending on the type of flow (inviscid or viscous) and consequently on the type of the examined grid (tetrahedral or hybrid); a maximum temporal speed-up coefficient equal to ~ 9.0 was obtained for an inviscid flow (isotropic fusion), while for laminar and turbulent ones ~ 7.5 and ~ 5.0 rates were achieved respectively (directional full-coarsening agglomeration), indicating the potential of this methodology for considerable acceleration of solution procedure [Lyg14c]. Based on speed-up results extracted from the DLR-F6 WB test case, the superiority of full-coarsening directional agglomeration [Nis11, Lyg14a, Lyg15] was proved, comparing to this of semi-coarsening one [Mav97, Mav98]; moreover, the higher contribution by the combined FMG-FAS strategy was revealed, comparing to only FAS one, although the first one remains susceptible to be confined in local minima or even lead solution to failure. For radiative heat transfer simulations, the aforementioned methodology was extended to an angular and a combined (nested) spatial/angular version, enhancing the corresponding solver with further acceleration capabilities [Lyg14b, Lyg14f, Lyg14g]; a maximum temporal speed-up coefficient equal to ~ 4.3 was succeeded for the spatial scheme, while for angular and nested ones ~ 3.3 and ~ 8.9 rates were obtained respectively (all of them extracted from test cases considering purely scattering media). Based on the examined results, the acceleration, derived by the developed multigrid schemes (spatial, angular and combined spatial/angular), was revealed to become higher with the increase of radiation exchange between different solid control angles; this interaction is usually a result of the combination of purely scattering media, fully reflecting surfaces, and symmetry boundary conditions [Lyg14g]. Unlike them, for simulations concerning only absorbing media and black walls, the spatial agglomeration multigrid scheme should be the preferred choice. Independently of the employed solver (flow or radiative heat transfer), the acceleration obtained by the developed multigrid methodology increases with the corresponding increase to the mesh size; the greater the number of DoF's, the greater is the acceleration gained [Lyg14g]. As the spatial agglomeration is performed separately in each partition (in which the initial grid is divided for parallel processing), the acceleration coefficients derived by the proposed multigrid scheme vary for different numbers of sub-domains, for the same test case. Based on the radiative heat transfer cases encountered in this study, an acceleration was observed with the increase of the number of sub-grids, instead of an expected deceleration due to a processor load imbalance; this improvement is assumed to be a result of the effective treatment of *ghost* nodes at overlapping regions during fusion procedure, deriving actually virtual *ghost supernodes* [Lyg14f].

To conclude, the obtained results by the developed numerical approach (*Galatea* code) against three- and quasi-three-dimensional benchmark test cases revealed its capability for both compressible fluid flow and radiative heat transfer simulations in terms of accuracy, geometric flexibility and computational efficiency.

9.2. Contributions

Considering similar studies available in open literature, the following original contributions have been introduced in this thesis:

- **A combination of a parallel node-centered finite-volume method with three-dimensional unstructured hybrid grids for radiative heat transfer prediction.** In three-dimensional runs, *a priori* greater numerical diffusion exists, comparing to two-dimensional ones; a remedy to this shortcoming appears to be the utilization of highly stretched elements at the region of interest, e.g., at boundary surfaces from which the desired distribution of incident radiative heat flux is obtained [Lyg12b]. Unlike CFD, cell-centered discretized tetrahedral grids are usually employed in a serial way for radiation simulations [Kim88, Bae98b, Mur98c, Kim01, Sal04, Kim10]; nevertheless, the proposed methodology derives equally accurate results in an acceptable computational time due to parallelization [Lyg12a, Lyg12b].
- **A second-order accurate spatial scheme jointed or not with the slope limiter of Van Albada-Van Leer or Min-mod, improving accuracy of radiative heat transfer finite-volume solutions, especially at boundary surfaces' regions with large radiative intensity gradients.** Unlike CFD, such methods aren't frequently employed in radiation simulations, except for a few similar studies concerning though unlimited second-order schemes with a cell-centered finite-volume method [Cap10] or bounded ones with DOM [Coe02]. Among the developed higher-order schemes (simple or coupled with Van Albada-Van Leer or Min-mod limiter), the unlimited first one appears to produce a flux over- or under-estimation (depending on the test case) due to the unbounded reconstructed values of radiative intensity, while no significant differences are observed between the other two more accurate ones incorporating limiting functions [Lyg13b, Lyg13c].
- **Implicit treatment of diffusively and specularly reflecting surfaces' boundary conditions.** According to the common practice, the wall boundary conditions of a general enclosure are imposed explicitly, setting radiative intensity of boundary nodes equal to the sum of their blackbody intensity, due to their temperature and the reflected one by the incoming rays [Bae98b, Rai99, Kim01, Kim05]; nevertheless, implicit approach developed in this study, besides resembling corresponding CFD techniques for free-slip, inlet and outlet surfaces, contributing to flux balances of boundary nodes, it avoids the requirement of the explicit one for *a priori* finer computational grids at these areas, resulting in significant computational savings [Lyg13b, Lyg13c]. Based on this approach, mirroring boundary conditions, resembling specularly reflecting walls, can be implemented as well. Unlike CFD, such boundaries aren't commonly applied in radiative heat transfer computations, except for a few studies, such as this of Liu et al. [Liu00], although in that way even coarser spatial resolutions can be used. For example, in axisymmetric test cases only the half computational field can be discretized [Lyg13c].
- **A grid adaptation method, employing h-refinement technique to improve accuracy of radiative heat transfer solution at specific regions of the examined mesh.** Although such methodologies, adapting grids with the addition of more DoF's [Kal96, Lyg13c, Lyg14a, Lyg15] or with their redistribution [Loh92, Kha00], are commonly applied in flow computations, they aren't frequently met in radiation ones. However, h-refinement approach developed in this study is revealed to be a viable tool alleviating significantly false scattering and consequently increasing accuracy of the final radiation prediction; it enriches automatically selected or user defined regions of the grid with more DoF's (nodes). Besides accuracy improvement, it results in significant computational savings, as it is implemented during the solution procedure, avoiding the construction of a locally or globally finer mesh from the beginning [Lyg13a].
- **A parallel spatial agglomeration multigrid method enhancing both flow and radiative heat transfer algorithms with improved computational performance.** Although such a methodology is frequently incorporated in CFD algorithms [Mav94, Mav96, Mav97, Mav98, Car00, Bla01, Sor03, Nis10, Nis11, Nis13, Lyg14b, Lyg14c, Lyg14d, Lyg14f, Lyg14g], this is not the case for radiative heat transfer ones; according to the presented results in Chapters 7 and 8 their efficiency can be considerably improved with its implementation. The proposed scheme considers the implementation of FAS (nested in FMG procedure) on successively coarser spatial resolutions,

derived from the finest initial sub-grids (in which the whole initial mesh has been decomposed for parallel processing) through the isotropic or directional (full- or semi-coarsening) fusion of their adjacent control volumes in a way similar to advancing front technique [Lyg14f]. The merging strategy is based on the methodology reported by Nishikawa et al. [Nis11], which allows for full-coarsening directional agglomeration of adjacent control cells on hybrid grids, preserving simultaneously the topology of the initial finest mesh; therefore, it is assumed to be more computationally efficient than the semi-coarsening approach introduced by Mavriplis [Mav97], which limits the fusion of implicit lines bottom nodes [Mav97, Lam04]. The previous state is confirmed by the speed-up results obtained for the CRM test case, encountered in this study. Further features have been included to the proposed algorithm, differentiating it from this of Nishikawa et al. [Nis10, Nis11, Nis13] as follows [Lyg14c, Lyg14f]: a) Agglomeration is extended across partition boundaries creating virtual *ghost supernodes* at overlapping regions according to the fusion of their corresponding *core* nodes at neighbouring sub-domains. b) Considering the previous characteristic, an implicit line can belong to different sub-grids, allowing for the utilization of a general partitioning application, such as METIS. c) Another limitation is imposed to preserve the topology of the initial mesh up to the coarsest generated one; after the first agglomeration process the nodes of the lowest prismatic layer are not allowed to be merged with the nodes of the next layer, while the same constraint is applied to the nodes of the second prismatic layer after the second process, and so forth. d) Since the line-agglomeration is accomplished, the isotropic fusion is performed similarly to the advancing front technique concept, assuming as initial *seed* nodes not the rest boundary ones, but those touched by the agglomeration front, which are actually the adjacent to the prismatic region tetrahedral nodes. e) In case of laminar or turbulent viscous flow, a distance-based scheme is employed for the prolongation of variables' corrections from the coarser to the finer grid; unlike other sophisticated types of prolongation (not simple point injected), it requires no more topological information than the edge-based data structure employed for flux computation on coarse resolutions.

- **A parallel angular agglomeration multigrid scheme for radiative heat transfer simulations.** A noticeable contribution of this thesis concerns the extension of spatial agglomeration multigrid method in angular direction for radiation computations. In the same to spatial scheme way, the RTE is approximated with FAS on successively coarser angular resolutions, derived from the initial finest one via the fusion of neighbouring solid control angles [Lyg14b, Lyg14f, Lyg14g]. Besides similarities, angular agglomeration appears to be much simpler, mainly due to the absence of the large number of limitations of the spatial scheme. Based on the performed simulations' results, the acceleration derived by this method is revealed to become higher with the increase of radiation exchange between different solid control angles; this interaction is usually a result of the combination of purely scattering media, fully reflecting surfaces, and symmetry boundary conditions [Lyg14f]. Although this methodology doesn't contribute in general so significantly to algorithm's efficiency, depending on the examined problem it can derive equally accelerated to the spatial one solutions; in the encountered test case considering radiative heat transfer in a cubic enclosure with reflecting walls ($\epsilon_w=0.1$), and an absorbing, emitting and anisotropically scattering gray medium ($k_a=0 \text{ m}^{-1}$, $\sigma_s=1 \text{ m}^{-1}$) almost the same speed-up coefficient was obtained by a three-level angular and a three-level spatial scheme [Lyg14g].
- **An extension of previous methods in a combined (nested) spatial/angular agglomeration multigrid scheme for radiation problems.** The last novelty developed in this study concerns a nested combination of the aforementioned (spatial and angular) agglomeration multigrid schemes, according to which a complete angular FAS cycle is accomplished at each level of the spatial multigrid methodology [Lyg14b, Lyg14f, Lyg14g]. Similarly to the angular multigrid version, it achieves better acceleration rates in test cases with increased radiation exchange between different solid control angles. Considering the simulations' results, it is revealed to be capable to succeed

even two or three times higher acceleration, comparing to only spatial or only angular multigrid method respectively [Lyg14g].

9.3. Ongoing work

Ongoing work concerns the incorporation of the developed spatial agglomeration procedure (slightly modified) in a mesh deformation algorithm using Radial Basis Functions (RBF's), in order to improve its computational performance by a reduced surface point selection. Although RBF-based mesh motion is assumed in general to be a very robust method producing high quality deformed grids, it appears to call for excessive computational effort and memory storage, especially in large-scale problems including meshes with millions or tens of millions nodes, as it requires the solution of an equation system with dimensions $N_s \times N_v$, where N_s denotes the number of surface nodes, while N_v the number of volume mesh nodes to be moved [Ren10]. A remedy to this shortcoming is revealed to be the proposed agglomeration method, employed though only to boundary surfaces, in order to reduce successively the number of their nodes; in the work of Rendall and Allen [Ren10] an error function is utilized for this point selection instead. In that way the dimensions of the aforementioned equation system are reduced and the volume mesh motion depends on a smaller number of surface points; as a result, efficiency of the algorithm is considerably improved. The new positions of the surface nodes of the initial (non-agglomerated) grid are finally obtained implementing RBF's interpolation. Figure 9.1 includes an example of the described methodology, concerning parabolic movement of the CRM (WB) aircraft wing in upper direction; more specifically the agglomerated surface nodes (yellow coloured) and aircraft/symmetry surface grid (black coloured) prior and after mesh deformation respectively are illustrated.

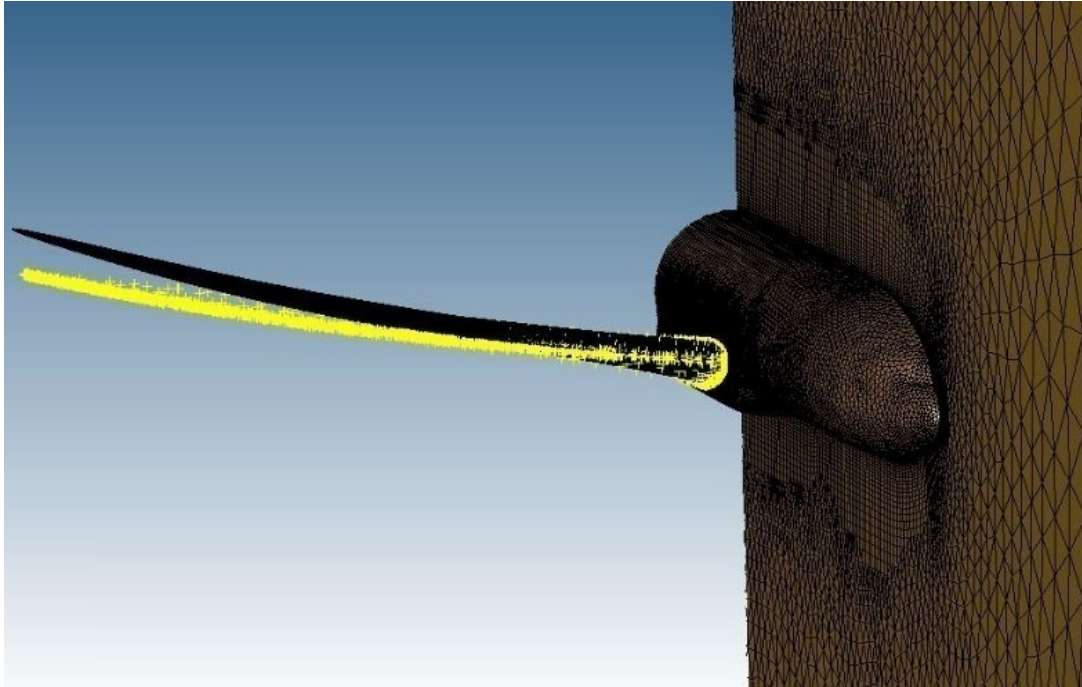


Figure 9.1: Agglomerated surface nodes (coloured yellow) and CRM aircraft/symmetry surface grid (coloured black) prior and after mesh deformation respectively.

9.4. Future developments

Besides the previously described ongoing work, future extensions of the presented numerical methodology include the following:

- Incorporation of the developed radiative heat transfer algorithm to the corresponding compressible fluid flow one, in order coupled compressible fluid flow/radiative heat transfer simulations to be encountered [Kim96, Ko08].
- Implementation of Arbitrary Lagrangian-Eulerian (ALE) approach to allow for the solution of fluid dynamics problems with moving mesh and boundaries [Koo00, Zha03, Ahn06]; combining this method with the aforementioned mesh deformation algorithm using RBF's, aeroelasticity analysis and Fluid-Structure Interaction (FSI) simulations can be performed [Bra09].
- Modelling of turbulence with the LES approach; according to this method the contributions of large, energy-carrying, eddies are computed straightforward with the numerical scheme, while the small ones are obtained with a Sub-Grid Scale (SGS) model [Sma63, Dea70, Bla01].
- Implementation of preconditioning methodology for test cases involving low Mach number flows, in order the stiff attitude of convective terms to be overcome [Bla01].
- Enhancement of flow and radiative heat transfer solvers with the capability of unsteady/transient simulations [Koo00, Tan02, Cha03, Tai03].
- Coupling of the developed radiative heat transfer algorithm with an appropriate model, e.g., WSGGM (Weighted Sum of Gray Gases Model) or SLW (Spectral Line-based Weighted sum of gray gases model), to account for non-gray media [Tri04b].
- Incorporation of a more sophisticated and accurate scheme than Legendre polynomials to compute scattering phase functions, e.g., Mie theory [Tri04a, Tri08].
- Examination of different combinations of the proposed spatial and angular agglomeration multigrid schemes, besides the nested one presented in this study [Lyg14b, Lyg14f, Lyg14g].

"Intentionally left blank"

10. References

- [Abb49] I.H. Abbott and A.E. Doenhoff, Theory of Wing Sections, *Dover Publications, Inc.*, New York, 1949.
- [Ahn06] H.T. Ahn and Y. Kallinderis, Strongly Coupled Flow/Structure Interactions with a Geometrically Conservative ALE Scheme on General Hybrid Meshes, *Journal of Computational Physics*, vol. 219, pp. 671-696, 2006.
- [Ahn07] H.T. Ahn and G.F. Carey, An Enhanced Polygonal Finite-Volume Method for Unstructured Hybrid Meshes, *International Journal for Numerical Methods in Fluids*, vol. 54, pp. 29-46, 2007.
- [And94] W.K. Anderson and D.L. Bonhaus, An Implicit Algorithm for Computing Turbulent Flows on Unstructured Grids, *Computers & Fluids*, vol. 23, pp. 1-21, 1994.
- [ANSYS06] ANSYS CFX-Solver Theory Guide, ANSYS CFX Release 11.0, December 2006.
- [Ara10] G. Araya, B. Evans, O. Hassan and K. Morgan, Assessment of Several Turbulence Models in a Supersonic Car, *Proceedings of the V European Conference on Computational Fluid Dynamics, ECCOMAS CFD 2010*, Lisbon, Portugal, 14-17 June 2010.
- [Bae97] S.W. Baek and M.Y. Kim, Analysis of Radiative Heating of a Rocket Plume Base with the Finite Volume Method, *International Journal of Heat and Mass Transfer*, vol. 40, pp. 1501-1508, 1997.
- [Bae98a] S.W. Baek and M.Y. Kim, Modification of the Discrete Ordinates Method in an Axisymmetric Cylindrical Geometry, *Numerical Heat Transfer Part B: Fundamentals*, vol. 31, pp. 313-326, 1998.
- [Bae98b] S.W. Baek, M.Y. Kim and J.S. Kim, Nonorthogonal Finite-Volume Solutions of Radiative Heat Transfer in a Three-Dimensional Enclosure, *Numerical Heat Transfer Part B: Fundamentals*, vol. 34, pp. 419-437, 1998.
- [Bae09] A. Baeza, C. Castro, F. Palacios and E. Zuazua, 2D Euler Shape Design on Non-Regular Flows Using Adjoint Rankine-Hugoniot Relations, *AIAA*, vol. 47, pp. 552-562, 2009.
- [Bar89] T.J. Barth and D.C. Jespersen, The Design and Application of Upwind Schemes on Unstructured Meshes, *AIAA, Proceedings of the 27th Aerospace Sciences Meeting and Exhibit, AIAA*, Reno, NV, Jan 9-12 1989, pp. 1-12, AIAA-89-0366.
- [Bar92] T.J. Barth, Aspects of Unstructured Grids and Finite-Volume Solvers for the Euler and Navier-Stokes Equations, *Proceedings of the AGARD-FDP-VKI special course at VKI*, Rhode-Saint-Genese 2-6 March 1992, AGARD-R-787, pp. 6.1-6.61.
- [Bla01] J. Blazek, Computational Fluid Dynamics: Principles and Applications, *Kidlington: Elsevier Science*, 2001.
- [Bor13] M.H. Bordbar and T. Hyppanen, Multiscale Numerical Simulation of Radiation Heat Transfer in Participating Media, *Heat Transfer Engineering*, vol. 34, pp. 54-69, 2013.
- [Bra84] A. Brandt, Multigrid Techniques with Applications to Fluid Dynamics: 1984 Guide, *VKI Lecture Series*, pp. 1-176, 1984.

- [Bra96] M.E. Braaten and S.D. Connell, Three-Dimensional Unstructured Adaptive Multigrid Scheme for the Navier-Stokes Equations, *AIAA*, vol. 34, pp. 281-290, 1996.
- [Bra09] A.L. Braun and A.M. Awruch, Aerodynamic and Aeroelastic Analyses on the CAARC Standard Tall Building Model Using Numerical Simulation, *Computers and Structures*, vol. 87, pp. 564-581, 2009.
- [Bre02] J. Bredberg, S. Peng and L. Davidson, An Improved $k-\omega$ Turbulence Model Applied to Recirculating Flows, *International Journal of Heat and Fluid Flow*, vol 23, pp. 731-743, 2002.
- [Bro01] O. Brodersen and A. Sturner, Drag Prediction of Engine-Airframe Interference Effects Using Unstructured Navier-Stokes Calculations, *Proceedings of the 19th AIAA Applied Aerodynamics Conference*, Anaheim, California, 11-14 June 2001.
- [Cap10] R. Capdevila, C.D. Perez-Segarra and A. Oliva, Development and Comparison of Different Spatial Numerical Schemes for the Radiative Transfer Equation Resolution Using Three-Dimensional Unstructured Meshes, *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 111, pp. 264-273, 2010.
- [Car68] B.G. Carlson and K.D. Lathrop, Transport Theory: The Method of Discrete-Ordinates, in H. Greenspan, C.N. Kelber and D. Okrent, Computing methods in reactor physics, *Gordon & Breach*, New York. 1968.
- [Car00] G. Carre, L. Fournier and S. Lanteri, Parallel Linear Multigrid Algorithms for the Acceleration of Compressible Flow Calculations, *Computer Methods in Applied Mechanics and Engineering*, vol. 184, pp. 427-448, 2000.
- [Cez13] M. Ceze and K.J. Fidkowski, Drag Prediction Using Adaptive Discontinuous Finite Elements, *Proceedings of the 51st AIAA Aerospace Sciences Meeting*, Grapevine, Texas, USA, 7 - 10 January 2013, AIAA 2013-0051.
- [Cha93] J.C. Chai, H.S. Lee and S.H. Patankar, Ray Effect and False Scattering in the Discrete Ordinates Method, *Numerical Heat Transfer Part B: Fundamentals*, vol. 24, pp. 373-389, 1993.
- [Cha94a] J.C. Chai, H.C. Lee and S.V. Patankar, Finite Volume Method for Radiation Heat Transfer, *Journal of Thermophysics and Heat Transfer*, vol. 8, pp. 419-425, 1994.
- [Cha94b] J.C. Chai, H.C. Lee and S.V. Patankar, Treatment of Irregular Geometries Using a Cartesian Coordinates Finite-Volume Radiation Heat Transfer Procedure, *Numerical Heat Transfer Part B: Fundamentals*, vol. 26, pp. 225-235, 1994.
- [Cha03] J.C. Chai, One-Dimensional Transient Radiation Heat Transfer Modelling Using a Finite-Volume Method, *Numerical Heat Transfer Part B: Fundamentals*, vol. 44, pp. 187-208, 2003.
- [Chi96] R.V. Chima, A $k-\omega$ Turbulence Model for Quasi-Three-Dimensional Turbomachinery Flows, *Proceedings of the 34th Aerospace Sciences Meeting and Exhibit, AIAA*, Reno, NV, Jan 15-18 1996, pp. 541-546, AIAA-96-0248.
- [Cho76] A Chorin, A Numerical Method for Solving Incompressible Viscous Flow Problems, *Journal of Computational Physics*, vol. 2, pp. 12-26, 1976.

- [Chu92] E.H. Chui, G.D. Raithby and P.M.J. Hughes, Prediction of Radiative Transfer in Cylindrical Enclosures with the Finite Volume Method, *Journal of Thermophysics and Heat Transfer*, vol. 6, pp. 605-611, 1992.
- [Chu93a] E.H. Chui, P.M.J. Hughes and G.D. Raithby, Implementation of the Finite Volume Method for Calculating Radiative Transfer in a Pulverized Fuel Flame, *Combustion Science and Technology*, vol. 92, pp. 225-242, 1993.
- [Chu93b] E.H. Chui and G.D. Raithby, Computation of Radiant Heat Transfer on a Nonorthogonal Mesh Using Finite Volume Method, *Numerical Heat Transfer Part B: Fundamentals*, vol. 23, pp. 269-288, 1993.
- [Cla57] G.C. Clark, C.M. Chu and S.W. Churchill, Angular Distribution Coefficients for Radiation Scattered by a Spherical Particle, *Journal of Optical Society of America*, vol. 47, pp. 81-84, 1957.
- [Coe98] P.J. Coelho, J.M. Goncalves and M.G. Carvalho, Modelling of Radiative Heat Transfer in Enclosures with Obstacles, *International Journal of Heat and Mass Transfer*, vol. 41, pp. 745-756, 1998.
- [Coe02] P.J. Coelho, Bounded Skew High-Order Resolution Schemes for the Discrete Ordinates Method, *Journal of Computational Physics*, vol. 175, pp. 412-437, 2002.
- [Coe14] P.J. Coelho, Advances in the Discrete Ordinates and Finite Volume Methods for the Solution of Radiative Heat Transfer Problems in Participating Media, *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 145, pp. 121-146, 2014.
- [Coo79] P.H. Cook, M.A. McDonald and M.C.P. Firmin, Aerofoil RAE 2822 – Pressure Distributions, and Boundary Layer and Wake Measurements, *Proceedings of the Fluid Dynamics Panel Working Group 04*, Neuilly sur Seine, France, May 1979, AGARD-AR-138, pp. A8.1-A8.25.
- [Dar06] M.S. Darwish, T. Saad and Z. Hamdan, A High Scalability Parallel Algebraic Multigrid Solver, *Proceedings of the European Conference on Computational Fluid Dynamics (ECCOMAS CFD 2006)*, Egmond aan Zee, The Netherlands, 5-8 September, 2006.
- [Das08] R. Das, S.C. Mishra and R. Uppaluri, Multiparameter Estimation in a Transient Conduction-Radiation Problem Using the Lattice Boltzmann Method and the Finite-Volume Method Coupled with the Genetic Algorithms, *Numerical Heat Transfer Part A: Applications*, vol. 53, pp. 1321-1338, 2008.
- [Dea70] J.W. Deardoff, A Numerical Study Three-Dimensional Turbulent Channel Flow at Large Reynolds Numbers, *Journal of Fluid Mechanics*, vol. 41, pp. 453-480, 1970.
- [Dec92] H. Deconick, R. Struijs, G. Bourgois, H. Paillere and P.L. Roe, Multidimensional Upwind Methods for Unstructured Grids, *Proceedings of the AGARD-FDP-VKI special course at VKI*, Rhode-Saint-Genese, 2-6 March 1992, AGARD-R-787, pp. 4.1-4.17.
- [Del11] A.I. Delis, I.K. Nikolos and M. Kazolea, Performance and Comparison of Cell-Centered and Node-Centered Unstructured Finite Volume Discretizations for Shallow Water Free Surface Flows, *Archives of Computational Methods in Engineering*, vol. 18, pp. 57-118, 2011.
- [Del13] A.I. Delis and I.K. Nikolos, A Novel Multidimensional Solution Reconstruction and Edge-Based Limiting Procedure for Unstructured Cell-Centered Finite Volumes with Application to Shallow Water Dynamics, *International Journal for Numerical Methods in Fluids*, vol. 71, pp. 584-633, 2013.

- [Du05] T. Du and Z. Wu, Mixed Analytical/Numerical Method Applied to the High Reynolds Number k - ϵ Turbulence Model, *Computers & Fluids*, vol. 34, pp. 97-119, 2005.
- [Dur96] P.A. Durbin, On the k -3 Stagnation Point Anomaly, *International Journal of Heat and Fluid Flow*, vol. 17, pp. 89-90, 1996.
- [Eva92] A. Evans, M.J. Marchant, N.P. Weatherill, J. Szmelter and D.K. Natakusumah, A Point Enrichment Strategy for Polygonal Volume Meshes, *Applied Mathematical Modelling*, vol. 16, pp. 562-575, 1992.
- [Fel03] J. Felcman, Grid Refinement/Alignment in 3D Flow Computation, *Mathematics and Computers in Simulation*, vol. 61, pp. 317-331, 2003.
- [Fer02] J.H. Ferziger and M. Peric, Computational Methods for Fluid Dynamics. 3rd Edition, *Springer*, 2002.
- [Fit99] A. Fiterman, R. Ben-Zvi and A. Kribus, DOTS: Pseudo-Time-Stepping Solution of the Discrete Ordinate Equations, *Numerical Heat Transfer Part B: Fundamentals*, vol. 35, pp. 163-183, 1999.
- [Fiv84] W.A. Fiveland, Discrete Ordinates Solutions of Transport Equations for Rectangular Enclosures, *Journal of Heat Transfer*, vol. 106, pp. 699-706, 1984.
- [Fiv87] W.A. Fiveland, Discrete Ordinate Methods for Radiative Heat Transfer in Isotropically and Anisotropically Scattering Media, *Journal of Heat Transfer*, vol. 109, pp. 809-812, 1987.
- [Fiv88] W.A. Fiveland, Three-Dimensional Radiative Heat Transfer Solutions by Discrete-Ordinates Method, *Journal of Thermophysics and Heat Transfer*, vol. 2, pp. 309-316, 1988.
- [Fiv94] W.A. Fiveland, Finite Element Formulation of the Discrete-Ordinates Method for Multidimensional Geometries, *Journal of Thermophysics and Heat Transfer*, vol. 8, pp. 426-433, 1994.
- [Fly98] M.J. Flynn, Computer Architecture: Pipelined and Parallel Processor Design, *Jones and Bartlett*, 1998.
- [Fum04] P. Furmanski and J. Bannaszek, Finite Element Analysis of Concurrent Radiation and Conduction in Participating Media, *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 84, pp. 563-573, 2004.
- [Gri98] M.J. Grismer, W.Z. Strang, R.F. Tomaro and F.C. Witzeman, Cobalt: a Parallel, Implicit, Unstructured Euler/Navier-Stokes Solver, *Advances in Engineering Software*, vol. 29, pp. 365-373, 1998.
- [Gri08] H. Grissa, F. Askri, M. Ben Salah and S. Ben Nasrallah, Nonaxisymmetric Radiative Transfer in Inhomogenous Cylindrical Media with Anisotropic Scattering, *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 109, pp. 494-513, 2008.
- [Gri10] H. Grissa, F. Askri, M. Ben Salah and S. Ben Nasrallah, Prediction of Radiative Heat Transfer in 3D Complex Geometries Using the Unstructured Control Volume Finite Element Method, *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 111, pp. 144-154, 2010.

- [Gue09] K. Guedri, M. Ammar Abbassi, M. Naceur Borjini, K. Halouani and R. Said, Application of the Finite-Volume Method to Study the Effects of Baffles on Radiative Heat Transfer in Complex Enclosures, *Numerical Heat Transfer Part A: Applications*, vol. 55, pp. 780-806, 2009.
- [Han02] V. Hannemann, Structured Multigrid Agglomeration on a Data Structure for Unstructured Meshes, *International Journal for Numerical Methods in Fluids*, vol. 40, pp. 361-368, 2002.
- [Has08] P. Hassanzadeh and G.D. Raithby, Finite-Volume Solution of the Second Order Radiative Transfer Equation: Accuracy and Solution Cost, *Numerical Heat Transfer Part B: Fundamentals*, vol. 53, pp. 374-382, 2008.
- [Hir90] C. Hirsch, Numerical Computation of Internal and External Flows. Vol. 2: Computational Methods for Inviscid and Viscous Flows, *John Wiley and Sons*, New York, 1990.
- [Hof88] N. Hoffmann and N.C. Markatos, Thermal Radiation Effects on Fires in Enclosures, *Applied Mathematical Modelling*, vol. 12, pp. 129-140, 1988.
- [Hos06] S. Hostika and K. McGrattan, Numerical Modelling of Radiative Heat Transfer in Water Sprays, *Journal of Fire Safety*, vol. 41, pp. 76-86, 2006.
- [Hot67] H.C. Hottel and A.F. Sarofim, Radiative Transfer, *McGraw Hill*, New York, 1967.
- [Hun70] J.E. Hunt, The Generation of Angular Distribution Coefficients for Radiation Scattered by a Spherical Particle, *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 10, pp. 857-864, 1970.
- [Hun11] B. Hunter and Z. Guo, Comparison of the Discrete-Ordinates Method and the Finite-Volume Method for Steady-State and Ultrafast Radiative Transfer Analysis in Cylindrical Coordinates, *Numerical Heat Transfer Part B: Fundamentals*, vol. 59, pp. 339-359, 2011.
- [Ili99] T. Iliescu, A 3D Flow-Aligning Algorithm for Convection-Diffusion Problems, *Applied Mathematics Letters*, vol. 12, pp. 67-70, 1999.
- [Jam81] A. Jameson, W. Schmidt and E. Turkel, Numerical Simulation of the Euler Equations by Finite Volume Methods Using Runge-Kutta Timestepping Schemes, *AIAA*, vol. 81, pp. 1259-1268, 1981.
- [Jam86] A. Jameson, T.J. Baker and N.P. Weatherill, Calculation of Inviscid Transonic Flow Over a Complete Aircraft, *Proceedings of the 24th AIAA Aerospace Sciences Meeting*, Nevada, USA, 6 - 9 January, 1986, AIAA 86-0103.
- [Jam92] A.S. Jamaluddin and P.J. Smith, Discrete-Ordinates Solution of Radiative Transfer Equation in Nonaxisymmetric Cylindrical Enclosures, *Journal of Thermophysics and Heat Transfer*, vol. 6, pp. 242-245, 1992.
- [Jim03] H. Jimbo, R. Liming and T. Heping, Effect of Anisotropic Scattering on Radiative Heat Transfer in Two-Dimensional Rectangular Media, *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 78, pp. 151-161, 2003.
- [Joh04] H. Johnston and J. Liu, Accurate, Stable and Efficient Navier-Stokes Solvers Based on Explicit Treatment of the Pressure Term, *Journal of Computational Physics*, vol. 199, pp. 221-259, 2004.

- [Jon97] T. Jongen and Y.P. Marx, Design of an Unconditionally Stable, Positive Scheme for the k- ϵ and Two-Layer Turbulence Models, *Computers & Fluids*, vol. 26, pp. 469-487, 1997.
- [Kal92] Y. Kallinderis and J.R. Baron, A New Adaptive Algorithm for Turbulent Flows, *Computers & Fluids*, vol. 21, pp. 77-96, 1992.
- [Kal96] Y. Kallinderis, A 3-D Finite Volume Method for the Navier Stokes Equations with Adaptive Hybrid Grids, *Applied Numerical Mathematics*, vol. 20, pp. 387-406, 1996.
- [Kal05a] Y. Kallinderis and H.T. Ahn, Incompressible Navier-Stokes Method with General Hybrid Meshes, *Journal of Computational Physics*, vol. 210, pp. 75-108, 2005.
- [Kal05b] Y. Kallinderis and C. Kavouklis, A Dynamic Adaptation Scheme for General 3-D Hybrid Meshes, *Computer Methods in Applied Mechanics and Engineering*, vol. 194, pp. 5019-5050, 2005.
- [Kar99] G. Karypis and V. Kumar, A Fast and Highly Quality Multilevel Scheme for Partitioning Irregular Graphs, *SIAM Journal on Scientific Computing*, vol. 20, pp. 359-392, 1999.
- [Kat09] A. Katz and A. Jameson, Multicloud: Multigrid Convergence with a Meshless Operator, *Journal of Computational Physics*, vol. 228, pp. 5237-5250, 2009.
- [Kav10] C. Kavouklis and Y. Kallinderis, Parallel Adaptation of General Three-Dimensional Hybrid Meshes, *Journal of Computational Physics*, vol. 229, pp. 3454-3473, 2010.
- [Kaz12] M. Kazolea, A.I. Delis, I.K. Nikolos and C.E. Synolakis, An Unstructured Finite Volume Numerical Scheme for Extended 2D Bousinesq-Type Equations, *Coastal Engineering*, vol. 69, pp. 42-66, 2012.
- [Ker00] E.P. Keramida, H.H. Liakos, M.A. Founti, A.G. Boudouvis and N.C. Markatos, Radiative Heat Transfer in Natural Gas-Fired Furnaces, *International Journal of Heat and Mass Transfer*, vol. 43, pp. 1801-1809, 2000.
- [Kha00] A. Khawaja, T. Minyard and Y. Kallinderis, Adaptive Hybrid Grid Methods, *Computer Methods in Applied Mechanics and Engineering*, vol. 180, pp. 1231-1245, 2000.
- [Kim88] T.K. Kim and H. Lee, Effect of Anisotropic Scattering on Radiative Heat Transfer in Two-Dimensional Rectangular Enclosures, *International Journal of Heat and Mass Transfer*, vol. 31, pp. 1711-1721, 1988.
- [Kim96] S.S. Kim and S.W. Baek, Radiation Affected Compressible Turbulent Flow Over a Backward Facing Step, *International Journal of Heat and Mass Transfer*, vol. 39, pp. 3325-3332, 1996.
- [Kim98] M.Y. Kim and S.W. Baek, Radiative Heat Transfer in a Body-Fitted Axisymmetric Cylindrical Enclosure, *Journal of Thermophysics and Heat Transfer*, vol. 12, pp. 596-599, 1998.
- [Kim01] M.Y. Kim, S.W. Baek and J.H. Park, Unstructured Finite-Volume Method for Radiative Heat Transfer in a Complex Two-Dimensional Geometry with Obstacles, *Numerical Heat Transfer Part B: Fundamentals*, vol. 39, pp. 617-635, 2001.
- [Kim03] K. Kim, Three-Dimensional Hybrid Grid Generator and Unstructured Flow Solver for Compressors and Turbines, PhD thesis, *Texas A&M University*, 2003.

- [Kim05a] G. Kim, S. Kim and Y. Kim, Parallelized Unstructured-Grid Finite Volume Method for Modelling Radiative Heat Transfer, *Journal of Mechanical Science and Technology*, vol. 19, pp. 1006-1017, 2005.
- [Kim05b] M.Y. Kim and S.W. Baek, Modelling of Radiative Heat Transfer in an Axisymmetric Cylindrical Enclosure with Participating Medium, *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 90, pp. 377-388, 2005.
- [Kim08] M.Y. Kim, S.W. Baek and S.I. Park, Evaluation of the Finite-Volume Solutions of Radiative Heat Transfer in a Complex Two-Dimensional Enclosure with Unstructured Polygonal Meshes, *Numerical Heat Transfer Part B: Fundamentals*, vol. 54, pp. 116-137, 2008.
- [Kim10] C. Kim, M.Y. Kim, M.Y. Yu and S.C. Mishra, Unstructured Polygonal Finite-Volume Solutions of Radiative Heat Transfer in a Complex Axisymmetric Enclosure, *Numerical Heat Transfer Part B: Fundamentals*, vol. 57, pp. 227-239, 2010.
- [Ko08] M. Ko and N.K. Anand, Three-Dimensional Combined Convective-Radiative Heat Transfer over a Horizontal Backward-Facing Step - A Finite-Volume Method, *Numerical Heat Transfer Part A: Applications*, vol. 54, pp. 109-129, 2008.
- [Koo00] B. Koobus, C. Farhat and H. Tran, Computation of Unsteady Viscous Flows around Moving Bodies Using the k- ϵ Turbulence Model on Unstructured Dynamic Grids, *Computer Methods in Applied Mechanics and Engineering*, vol. 190, pp. 1441-1466, 2000.
- [Kou03] D.G. Koubogiannis, A.N. Athanasiadis and K.C. Giannakoglou, One- and Two-Equation Turbulence Models for the Prediction of Complex Cascade Flows Using Unstructured Grids, *Computer and Fluids*, vol. 32, pp. 403-430, 2003.
- [Kun92] R.F. Kunz and B. Lakshminarayana, Stability of Explicit Navier Stokes Procedures Using k- ϵ and k- ϵ Algebraic Reynolds Stress Turbulence Models, *Journal of Computational Physics*, vol. 103, pp. 141-59, 1992.
- [Laf05] K.R. Laflin, S.M. Klausmeyer, T. Zickuhr, J.C. Vassberg, R.A. Wahls, J.H. Morrison, O.P. Brodersen, M.E. Rakowitz, E.N. Tinoco and J. Godard, Data Summary from Second AIAA Computational Fluid Dynamics Drag Prediction Workshop, *Journal of Aircraft*, vol. 42, pp. 1165-1178, 2005.
- [Lal88a] M.H. Lallemand, Etude de Schemas Runge-Kutta a 4 pas pour la Resolution Multigrille des Equations d' Euler 2D, *Raport de Recherche*, INRIA, 1988.
- [Lal88b] M.H. Lallemand, Schemas decentres multigrilles pour la resolution des equations d' Euler en elements finis, PhD Thesis, *Universite de Provence*, France, 1988.
- [Lam04] N.K. Lambropoulos, D.G. Koubogiannis and K.C. Giannakoglou, Acceleration of a Navier-Stokes Equation Solver for Unstructured Grids Using Agglomeration Multigrid and Parallel Processing, *Computer Methods in Applied Mechanics and Engineering*, vol. 193, pp. 781-803, 2004.
- [Lan96] S. Lanteri, Parallel Solutions of Compressible Flows Using Overlapping and Nonoverlapping Mesh Partitioning Strategies, *Parallel Computing*, vol. 22, pp. 943-968, 1996.

- [Lan98] C.B. Laney, Computational Gasdynamics, *Cambridge University Press*, 1998.
- [Lan05] R.B. Langtry, M. Kuntz and F.R. Menter, Drag Prediction of Engine-Airframe Interference Effects with CFX-5, *Journal of Aircraft*, vol. 42, pp. 1523-1529, 2005.
- [Lar91] B. Larrouturou, How to Preserve Mass Fractions when Computing Compressible Multi-component Flows, *Journal of Computational Physics*, vol. 95, pp. 59-84, 1991.
- [Lau74] B.E. Launder and D.B. Spalding, The Numerical Computation of Turbulent Flows, *Computer Methods in Applied Mechanics and Engineering*, vol. 3, pp. 269-289, 1974.
- [Lee10] E.M. Lee-Rausch, D.P. Hammond, E.J. Nielsen, S.Z. Pirzadeh and C.L. Rumsey, Application of the FUN3D Unstructured-Grid Navier-Stokes Solver to the 4th AIAA Drag Prediction Workshop Cases, *Proceedings of the 28th AIAA Applied Aerodynamics Conference*, Chicago, Illinois, USA, 28 June - 1 July 2010, AIAA 2010-4551.
- [Lev13] D.W. Levy, K.R. Laflin, E.N. Tinoco, J.C. Vassberg, M. Mani, B. Rider, C. Rumsey, R.A. Wahls, J.H. Morrison, O.P. Brodersen, S. Crippa, D.J. Mavriplis and M. Murayama, Summary of Data from the Fifth AIAA CFD Drag Prediction Workshop, *Proceedings of the 51st AIAA Aerospace Sciences Meeting*, Grapevine, Texas, USA, 7 - 10 January 2013, AIAA 2013-0046.
- [Lie02] J.H. Lienhard IV and J.H. Lienhard V, A Heat Transfer Textbook. 3rd Edition, *Phlogiston Press*, Cambridge, Massachusetts, 2002.
- [Liu96] F. Liu and X. Zheng, A Strongly Coupled Time-Marching Method for Solving the Navier-Stokes and $k-\omega$ Turbulence Model Equations with Multigrid, *Journal of Computational Physics*, vol. 128, pp. 289-300, 1996.
- [Liu00] J. Liu, H.M. Shang and Y.S. Chen, Development of an Unstructured Radiation Model Applicable for Two-Dimensional Planar, Axisymmetric, and Three-Dimensional Geometries, *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 66, pp. 17-33, 2000.
- [Liu04] L.H. Liu, Finite Element Simulation of Radiative Heat Transfer in Absorbing and Scattering Media, *Journal of Thermophysics and Heat Transfer*, vol. 18, pp. 555-557, 2004.
- [Liu05] L.H. Liu, Finite Element Solution of Radiative Transfer Across a Slab with Variable Spatial Refractive Index, *International Journal of Heat and Mass Transfer*, vol. 48, pp. 2260-2265, 2005.
- [Liu06] L.H. Liu, L. Zhang and H.P. Tan, Finite Element Method for Radiation Heat Transfer in Multi-dimensional Graded Index Medium, *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 97, pp. 436-445, 2006.
- [Loc81] F.C. Lockwood and N.G. Shah, A New Radiation Solution Method for Incorporation in General Combustion Prediction Procedures, *Proceedings of the 18th International Symposium on Combustion*, The Combustion Institute, Pittsburgh, 1981.
- [Loh92] R. Lohner, Finite Element Methods in CFD: Grid generation, Adaptivity and Parallelization, *Proceedings of the AGARD-FDP-VKI special course at VKI*, Rhode-Saint-Genese, 2-6 March 1992, AGARD-R-787: 8.1-8.58.

- [Luo05] H. Luo, J.D. Baum and R. Lohner, High-Reynolds Number Viscous Flow Computations Using an Unstructured-Grid Method, *Journal of Aircraft*, vol. 42, pp. 483-492, 2005.
- [Lyg11] G.N. Lygidakis and I.K. Nikolos, An Unstructured Node-Centered Finite Volume Method for Computing 3D Viscous Compressible Flows on Hybrid Grids, *Proceedings of the 7th GRACM (Greek Association of Computational Mechanics) International Congress on Computational Mechanics*, Athens, 30 June - 2 July 2011.
- [Lyg12a] G.N. Lygidakis and I.K. Nikolos, A Parallelized Node-Centered Finite Volume Method for Computing Radiative Heat Transfer on 3D Unstructured Hybrid Grids, *Proceedings of the 11th Biennial Conference on Engineering Systems Design and Analysis, ASME-ESDA2012*, Nantes, France, 2-4 July 2012, vol.1, pp. 177-186, ESDA2012-82331.
- [Lyg12b] G.N. Lygidakis and I.K. Nikolos, Using the Finite-Volume Method and Hybrid Unstructured Meshes to Compute Radiative Heat Transfer in 3-D Geometries, *Numerical Heat Transfer Part B: Fundamentals*, vol. 62, pp. 289-314, 2012.
- [Lyg13a] G.N. Lygidakis and I.K. Nikolos, Evaluating a Parallel Node-Centered Finite-Volume Algorithm, named Galatea, in simulation of 3D Compressible Flows, *Proceedings of the 10th HSTAM (Hellenic Society for Theoretical and Applied Mechanics) International Congress on Mechanics*, Chania, Crete, Greece, 25-27 May 2013.
- [Lyg13b] G.N. Lygidakis and I.K. Nikolos, Improving the Accuracy of a Finite-Volume Method for Computing Radiative Heat Transfer in Three-Dimensional Unstructured Meshes, *Proceedings of the ECCOMAS Special Interest Conference - SEECCM 2013: 3rd South-East European Conference on Computational Mechanics - An IACM Special Interest Conference 2013*, Kos, Greece, 12-14 June 2013, pp. 599-620.
- [Lyg13c] G.N. Lygidakis and I.K. Nikolos, Using a High-Order Spatial/Temporal Scheme and Grid Adaptation with a Finite-Volume Method for Radiative Heat Transfer, *Numerical Heat Transfer Part B: Fundamentals*, vol. 64, pp. 89-117, 2013.
- [Lyg14a] G.N. Lygidakis and I.K. Nikolos, Assessment of the Academic CFD Code "Galatea" Using the NASA Common Research Model (CRM), *Proceedings of the 12th Biennial Conference on Engineering Systems Design and Analysis, ASME-ESDA2014*, Copenhagen, Denmark, 25-27 June 2014, ESDA2014-20265.
- [Lyg14b] G.N. Lygidakis and I.K. Nikolos, A Parallel Finite-Volume Spatial/Angular Agglomeration Multigrid Method for Radiative Heat Transfer Computation, *Proceedings of the HEAT TRANSFER 2014, 13th International Conference on Simulation and Experiments in Heat Transfer and its Applications (WIT (Wessex Institute of Technology) Transactions on Engineering Sciences)*, A Coruna, Spain, 2-4 July 2014, vol. 83, pp. 275-286.
- [Lyg14c] G.N. Lygidakis and I.K. Nikolos, A Parallel Agglomeration Multigrid Method for the Acceleration of Compressible Flow Computations on 3D Hybrid Unstructured Grids, *Proceedings of the 11th World Congress on Computational Mechanics (WCCM XI), 5th European Conference on Computational Mechanics (ECCM V), 6th European Conference on Computational Fluid Dynamics (ECFD VI), IACM-ECCOMAS*, Barcelona, Spain, 20-25 July 2014, pp. 4852-4863.

- [Lyg14d] G.N. Lygidakis, S.S. Sarakinos and I.K. Nikolos, A Parallel Agglomeration Multigrid Method Incompressible Flow Simulations, *Proceedings of the 9th International Conference on Engineering Computational Technology, ECT 2014*, Naples, Italy, 2-5 September 2014.
- [Lyg14e] G.N. Lygidakis and I.K. Nikolos, Using the DLR-F6 Aircraft Model for the Evaluation of the Academic CFD Code "Galatea", *Proceedings of the International Mechanical Engineering Congress and Exposition, ASME-IMECE2014*, Montreal, Canada, 14-20 November 2014, IMECE2014-39756.
- [Lyg14f] G.N. Lygidakis and I.K. Nikolos, Using a Parallel Spatial/Angular Agglomeration Multigrid Scheme to Accelerate the FVM Radiative Heat Transfer Computation - Part I: Methodology, *Numerical Heat Transfer Part B: Fundamentals*, vol. 66, pp. 471-497, 2014.
- [Lyg14g] G.N. Lygidakis and I.K. Nikolos, Using a Parallel Spatial/Angular Agglomeration Multigrid Scheme to Accelerate the FVM Radiative Heat Transfer Computation - Part II: Numerical Results, *Numerical Heat Transfer Part B: Fundamentals*, vol. 66, pp. 498-525, 2014.
- [Lyg15] G.N. Lygidakis and I.K. Nikolos, Numerical Analysis of Flow over the NASA Common Research Model Using the Academic Computational Fluid Dynamics Code *Galatea*, *ASME Journal of Fluids Engineering*, vol. 137, FE-14-1602, DOI: 10.1115/1.4029730.
- [Lyr02] P.R.M. Lyra, R.C.F. Lima, C.S.C. Guimaraes and D.K.E. Carvalho, An Edge-Based Unstructured Finite Volume Method for the Solution of Potential Problems, *Mecanica Computacional*, vol. XXI, pp. 1213-31, 2002.
- [Mar07] O. Marxen, G. Iaccarino and E.S.G. Shaqfeh, Numerical Simulation of Hypersonic Boundary-Layer Instability Using Different Gas Models, *Center of Turbulence Research, Annual Research Briefs 2007*, pp. 15-27, 2007.
- [Mat99] S.R. Mathur and J.Y. Murthy, Radiative Heat Transfer in Periodic Geometries Using a Finite Volume Scheme, *Journal of Heat Transfer*, vol. 121, pp. 357-364, 1999.
- [Mav94] D.J. Mavriplis and L. Martinelli, Multigrid Solution of Compressible Turbulent Flow on Unstructured Meshes Using a Two-Equation Model, *International Journal for Numerical Methods in Fluids*, vol. 18, pp. 887-914, 1994.
- [Mav96] D.J. Mavriplis and V. Venkatakrishnan, A 3D Agglomeration Multigrid Solver for the Reynolds-Averaged Navier-Stokes Equations on Unstructured Meshes, *International Journal for Numerical Methods in Fluids*, vol. 23, pp. 527-544, 1996.
- [Mav97] D.J. Mavriplis, Directional Coarsening and Smoothing for Anisotropic Navier-Stokes Problems, *Electronic Transactions on Numerical Analysis*, vol. 6, pp. 182-197, 1997.
- [Mav98] D.J. Mavriplis, Multigrid Strategies for Viscous Flow Solvers on Anisotropic Unstructured Meshes, *Journal of Computational Physics*, vol. 145, pp. 141-165, 1998.
- [Mav99] D.J. Mavriplis and S. Pirzadeh, Large-Scale Parallel Unstructured Mesh Computations for 3D High-Lift Analysis, *NASA ICASE Report*, No. 99-9, 1999.
- [Mav10] D. Mavriplis and M. Long, NSU3D Results for the Fourth AIAA Drag Prediction Workshop, *Proceedings of the 28th AIAA Applied Aerodynamics Conference*, Chicago, Illinois, USA, 28 June - 1 July 2010, AIAA 2010-4550.

- [Men94] F.R. Menter, Two-Equation Eddy Viscosity Turbulence Models for Engineering Applications, *AIAA*, vol. 32, pp. 1598-1605, 1994.
- [Men03a] F. Menter, F.J. Carregal, T. Esch and B. Konno, The SST Turbulence Model with Improved Wall Treatment for Heat Transfer Predictions in Gas Turbines, *Proceedings of the International Gas Turbine Congress*, Tokyo, 2-7 November 2003, IGTC2003-TS-059: 1-7.
- [Men03b] F.R. Menter, M. Kunz and R. Langtry, Ten Years of Industrial Experience with the SST Turbulence Model, *Turbulence, Heat and Mass Transfer 4*, pp.625-632, 2003.
- [METIS] METIS, Family of multilevel partitioning algorithms, <http://glaros.dtc.umn.edu>.
- [Mit98] S. Mittal, Finite Element Computation of Unsteady Viscous Compressible Flows, *Computer Methods in Applied Mechanics and Engineering*, vol. 157, pp. 151-175, 1998.
- [MPI] MPI - Message Passing Interface, <http://en.wikipedia.org>.
- [Mun98] B.R. Munson, D.F. Young and T.I. Okiishi, Fundamentals of Fluid Mechanics, *John Wiley & Sons, Inc.*, New York, 1998.
- [Mur98a] J.Y. Murthy and S.R. Mathur, Radiative Heat Transfer in Axisymmetric Geometries Using an Unstructured Finite-Volume Method, *Numerical Heat Transfer Part B: Fundamentals*, vol. 33, pp. 397-416, 1998.
- [Mur98b] J.Y. Murthy and S.R. Mathur, Finite Volume Method for Radiative Heat Transfer Using Unstructured Meshes, *Journal of Thermophysics and Heat Transfer*, vol. 12, pp. 313-321, 1998.
- [Mur98c] J.Y. Murthy and S.R. Mathur, A Finite Volume Method for Radiative Heat Transfer Using Unstructured Meshes, *Proceedings of the 36th Aerospace Sciences Meeting and Exhibit*, Reno, NV, USA, 12 - 15 January, AIAA 98-0860.
- [Mur13] M. Murayama, K. Yamamoto, A. Hashimoto, T. Ishida, M. Ueno, K. Tanaka and Y. Ito, Summary of JAXA Studies for the Fifth AIAA CFD Drag Prediction Workshop Using UPACS and FaSTAR, *Proceedings of the 51st AIAA Aerospace Sciences Meeting*, Grapevine, Texas, USA, 7 - 10 January 2013, AIAA 2013-0049.
- [NASA] NASA/Langley Research Center, www.larc.nasa.gov.
- [Nis10] H. Nishikawa, B. Diskin and J.L. Thomas, Critical Study of Agglomerated Multigrid Methods for Diffusion, *AIAA*, vol. 48, pp. 839-847, 2010.
- [Nis11] H. Nishikawa and B. Diskin, Development and Application of Parallel Agglomerated Multigrid Methods for Complex Geometries, *Proceedings of the 20th AIAA Computational Fluid Dynamics Conference*, Honolulu, Hawaii, USA, 27 - 30 June, 2011, AIAA 2011-3232.
- [Nis13] H. Nishikawa, B. Diskin, J.L. Thomas, and D.P. Hammond, Recent Advances in Agglomerated Multigrid, *Proceedings of the 51st AIAA Aerospace Sciences Meeting*, Grapevine, Texas, USA, 7 - 10 January, 2013, AIAA 2013-0863.

- [Osh84] S. Osher, Riemann Solvers, the Entropy Condition and Difference Approximations, *SIAM Journal on Numerical Analysis*, vol. 21, pp. 217-235, 1984.
- [PAB3D] PAB3D, A 3-D Navier-Stokes CFD System, ONERA M6 Wing: Study # 1, <http://www.asmc-usa.com/PAB3D>.
- [Pai01] C.C. Pain, A.P. Umpleby, C.R.E. de Oliveira and A.J.H. Goddard, Tetrahedral Mesh Optimisation and Adaptivity for Steady-State and Transient Finite Element Calculations, *Computer Methods in Applied Mechanics and Engineering*, vol. 190, pp. 3771-3796, 2001.
- [PARCOMP] Introduction to Parallel Computing, https://computing.llnl.gov/tutorials/parallel_comp.
- [Par13] M.A. Park, K.R. Laflin, M.S. Chaffin, N. Powell and D.W. Levy, CFL3D, FUN3D and NSU3D Contributions to the Fifth Drag Prediction Workshop, *Proceedings of the 51st AIAA Aerospace Sciences Meeting*, Grapevine, Texas, USA, 7 - 10 January 2013, AIAA 2013-0050.
- [Pen99] S. Peng and L. Davidson, Computation of Turbulent Buoyant Flows in Enclosures with Low-Reynolds-Number $k-\omega$ Models, *International Journal of Heat and Fluid Flow*, vol. 20, pp. 172-84, 1999.
- [Per07] P. Persson, Viscous Flow Simulations Using the Compact Discontinuous Galerkin (CDG) Method and Efficient Iterative Solvers, *Proceedings of the 14th International Conference on Finite Elements in Flow Problems*, Santa Fe, NM, 26 March 2007.
- [Pet07] J. Peter and F. Drullion, Large Stencil Viscous Flux Linearization for the Simulation of 3D Compressible Turbulent Flows with Backward-Euler Schemes, *Computers & Fluids*, vol. 36, pp. 1005-1027, 2007.
- [Rai90] G.D. Raithby and E.H. Chui, A Finite-Volume Method for Predicting a Radiant Heat Transfer in Enclosures with Participating Media, *Journal of Heat Transfer*, vol. 112, pp. 415-423, 1990.
- [Rai99] G.D. Raithby, Discussion of the Finite Volume Method for Radiation, and its Application Using 3D Unstructured Meshes, *Numerical Heat Transfer Part B: Fundamentals*, vol. 35, pp. 389-405, 1999.
- [Raz83] M.M. Razzaque, D.E. Klein and J.R. Howell, Finite Element Solution of Radiative Heat Transfer in Two-Dimensional Rectangular Enclosure with Gray Participating Media, *Journal of Heat Transfer*, vol. 105, pp. 933-936, 1983.
- [Ren10] T.C.S. Rendall and C.B. Allen, Reduced Surface Point Selection Options for Efficient Mesh Deformation Using Radial Basis Functions, *Journal of Computational Physics*, vol. 229, pp. 2810-2820, 2010.
- [Roe81] P. Roe, Approximate Riemann Solvers, Parameter Vectors and Difference Schemes, *Journal of Computational Physics*, vol. 43, pp. 357-371, 1981.
- [Ryb96] G.B. Rybicki, Radiative transfer, *Journal of Astrophysics and Astronomy*, vol. 17, pp. 95-112, 1996.
- [Saf74] P.G. Saffman and D.C. Wilcox, Turbulence-Model Predictions for Turbulent Boundary Layers, *AIAA*, vol. 12, pp. 541-546, 1974.

- [Sal04] M.B. Salah, F. Askri, K. Slimi and S.B. Nasrallah, Numerical Resolution of the Radiative Transfer Equation in a Cylindrical Enclosure with the Finite Volume Method, *International Journal of Heat and Mass Transfer*, vol. 47, pp. 2501-2509, 2004.
- [Sar14] S.S. Sarakinos, G.N. Lygidakis and I.K. Nikolos, Evaluation of a Parallel Agglomeration Multigrid Finite-Volume Algorithm, named Galatea-I, for the Simulation of Incompressible Flows on 3D Hybrid Unstructured Grids, *Proceedings of the International Mechanical Engineering Congress and Exposition, ASME-IMECE2014*, Montreal, Canada, 14-20 November 2014, IMECE2014-39759.
- [Sar15] S.S. Sarakinos, G.N. Lygidakis and I.K. Nikolos, Acceleration Strategies for Simulating Compressible and Incompressible Flows, *Proceedings of the 4th International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering, PARENG 2015*, Dubrovnik, Croatia, 24-27 March 2015.
- [Sas95] C. Sasse, R. Koenigsdorf and S. Frank, Evaluation of an Improved Hybrid Six-Flux/Zone Model for Radiative Transfer in Rectangular Enclosures, *International Journal of Heat and Mass Transfer*, vol. 38, pp. 3423-3431, 1995.
- [Sca13] L.C. Scalabrin and R.F. de Souza, Grid Assessment Using the NASA Common Research Model (CRM) Wind Tunnel Data, *Proceedings of the 51st AIAA Aerospace Sciences Meeting*, Grapevine, Texas, USA, 7 - 10 January 2013, AIAA 2013-0052.
- [Sch79] V. Schmitt and F. Charpin, Pressure Distributions on the ONERA-M6 Wing at Transonic Mach Numbers, *Office National d'etudes et de recherches aerospaciales*, Chatillon, France, AGARD-AR-138: B1.1-B1.44.
- [Scl13] A.J. Sclafani, J.C. Vassberg, C. Winkler, A.J. Dorgan, M. Mani, M.E. Olsen and J.C. Coder, DPW-5 Analysis of the CRM in a Wing-Body Configuration Using Structured and Unstructured Meshes, *Proceedings of the 51st AIAA Aerospace Sciences Meeting*, Grapevine, Texas, USA, 7 - 10 January 2013, AIAA 2013-0048.
- [Sma63] J. Smagorinsky, General Circulation Experiments with the Primitive Equations, *Monthly Weather Review*, vol. 91, pp. 99-165, 1963.
- [Smi96] B. Smith, P. Bjorstad and W. Gropp, Domain Decomposition. Parallel Multilevel Methods for Elliptic Partial Differential Equations, *Cambridge University Press*, 1996.
- [Smi07] T.M. Smith et al., Comparison of Reconstruction Techniques for Unstructured Mesh Vertex Centered Finite Volume Schemes, *Proceedings of the 18th AIAA Computational Fluid Dynamics Conference*, Miami, Florida, 25-28 June 2007.
- [Sor03] K.A. Sorensen, O. Hassan, K. Morgan and N.P. Weatherill, A Multigrid Accelerated Hybrid Unstructured Mesh Method for 3D Compressible Turbulent Flow, *Computational Mechanics*, vol. 31, pp. 101-114, 2003.
- [Ste71] F.R. Steward and P. Cannon, The Calculation of Radiative Heat Flux in a Cylindrical Furnace Using the Monte Carlo method, *International Journal of Heat and Mass Transfer*, vol. 14, pp. 245-262, 1971.
- [Ste81] J.L. Steger and R.F. Warming, Flux Vector Splitting of the Inviscid Gasdynamic Equations with Application to Finite Difference Methods, *Journal of Computational Physics*, vol. 40, pp. 263-293, 1981.

- [Ste94] J. Stellant and E. Dick, A Multigrid Method for the Compressible Navier-Stokes Equations Coupled to the k- ϵ Turbulence Equations, *International Journal of Numerical Methods for Heat and Fluid Flow*, vol. 4, pp. 99-113, 1994.
- [Ste07] G.D. Stefanidis, B. Merci, G.J. Heyndericx and G.B. Marin, Gray/Nongray Gas Radiation Modelling in Steam Cracker CFD Calculations, *AIChE Journal*, vol. 53, pp. 1658-1669, 2007.
- [Swe84] P.K. Sweby, High Resolution Schemes Using Flux Limiters for Hyperbolic Conservation Laws, *SIAM Journal on Numerical Analysis*, vol. 21, pp. 995-1011, 1984.
- [Tai03] C.H. Tai and Y. Zhao, Parallel Unsteady Incompressible Viscous Flow Computations Using an Unstructured Multigrid Method, *Journal of Computational Physics*, vol. 192, pp. 277-311, 2003.
- [Tal05] P. Talukdar, M. Steven, F.V. Issendorff and D. Trimis, Finite Volume Method in 3-D Curvilinear Coordinates with Multiblocking Procedure for Radiative Transport Problems, *International Journal of Heat and Mass Transfer*, vol. 48, pp. 4657-4666, 2005.
- [Tal11] P. Talukdar, Mixed Convection and Non-Gray Radiation in a Horizontal Rectangular Duct, *Numerical Heat Transfer Part A: Applications*, vol. 59, pp. 185-208, 2011.
- [Tan02] Z.M. Tan and P.F. Hsu, Transient Radiative Heat Transfer in Three-Dimensional Homogenous and non Homogenous Participating Media, *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 73, pp. 181-194, 2002.
- [Tor97] E.F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics. A Practical Application. 2nd Edition, *Springer*, 1997.
- [Tri04a] D.N. Trivic, T.J. O'Brien and C.H. Amon, Modelling the Radiation of Anisotropically Scattering Media by Coupling Mie Theory with Finite Volume Method, *International Journal of Heat and Mass Transfer*, vol. 47, pp. 5675-5780, 2004.
- [Tri04b] D.N. Trivic, Modelling of 3-D Non-Gray Gases Radiation by Coupling the Finite Volume Method with weighted sum of Gray Gases Model, *International Journal of Heat and Mass Transfer*, vol. 47, pp. 1367-1382, 2004.
- [Tri08] D.N. Trivic and C.H. Amon, Modelling the 3-D Radiation of Anisotropically Scattering Media by Two Different Numerical Methods, *International Journal of Heat and Mass Transfer*, vol. 51, pp. 2711-2732, 2008.
- [Ull06] A. Ullah and K. Harada, Continuity Analysis of Non-Uniform Subdivision Surfaces, *International Journal of Computer Science and Network Security*, vol. 6, pp. 52-56, 2006.
- [Vak09] S. Vakili and M. Darbandi, Recommendations on Enhancing the Efficiency of Algebraic Multigrid Preconditioned GMRES in Solving Coupled Fluid Flow Equations, *Numerical Heat Transfer Part B: Fundamentals*, vol. 55, pp. 232-256, 2009.
- [VanA82] G.D. Van Albada, B. Van Leer and W.W. Roberts, A Comparative Study of Computational Methods in Cosmic Gas Dynamics, *Astronomy and Astrophysics*, vol. 108, pp.46-84, 1982.

- [VanL85] B. Van Leer, On the Relation Between the Upwind Differencing Schemes of Godunov, Enguist-Osher and Roe, *SIAM Journal on Scientific and Statistical Computing*, vol. 5, pp. 1-20, 1985.
- [Vas10] J.C. Vassberg, E.N. Tinoco, M. Mani, B. Rider, T. Zickuhr, D.W. Levy, O.P. Brodersen, B. Eisfeld, S. Crippa, R.A. Wahls, J.H. Morrison, D.J. Mavriplis and M. Murayama, Summary of the Fourth AIAA CFD Drag Prediction Workshop, *Proceedings of the 28th AIAA Applied Aerodynamics Conference*, Chicago, Illinois, USA, 28 June - 1 July 2010, AIAA 2010-4547.
- [Ven91] V. Venkatakrishnan and D. Mavriplis, Implicit Solvers for Unstructured Meshes, *Institute for Computer Applications in Science and Engineering*, NASA Langley Research Center, Hampton, Virginia, NASA Contractor Report 187564 1991, ICASE Report No. 91-40.
- [Ven95] V. Venkatakrishnan, Implicit Schemes and Parallel Computing in Unstructured Grid CFD, *Proceedings of the 26th Computational Fluid Dynamics Lecture Series Program*, Von Karman Institute for Fluid Dynamics, Rhode, Saint-Genese, Belgium, 13-17 March 1995.
- [Vos10] J.B. Vos and S. Sanchi, DPW4 Results Using Different Grids Including Near-Field/Far-Field Drag Analysis, *Proceedings of the 28th AIAA Applied Aerodynamics Conference*, Chicago, Illinois, USA, 28 June - 1 July 2010, AIAA 2010-4552.
- [Vos13] J.B. Vos, S. Sanchi and A. Gehri, Drag Prediction Workshop 4 Results Using Different Grids Including Near-Field/Far-Field Drag Analysis, *Journal of Aircraft*, vol. 50, pp. 1615-1627, 2013.
- [WIND] WIND code, J.W. Slater, ONERA M6 Wing: Study # 1, <http://www.grc.nasa.gov/www/wind/valid/m6wing/m6wing01/m6wing01.html>.
- [Wolfram] Wolfram MathWorld, <http://mathworld.wolfram.com>.
- [Yod99] D.A. Yoder and N.J. Georgiadis, Implementation and Validation of the Chien k- ϵ Turbulence Model in the Wind Navier-Stokes Code, *Proceedings of the 37th Aerospace Sciences Meeting and Exhibit*, Reno, NV, Jan 11-14, 1999, AIAA-99-0745.
- [Zha03] Y. Zhao and A. Forhad, A General Method for Simulation of Fluid Flows with Moving and Compliant boundaries on Unstructured Grids, *Computer Methods in Applied Mechanics and Engineering*, vol. 197, pp. 4439-4466, 2003.
- [Zor98] D.N. Zorin, Stationary Subdivision and Multiresolution Surface Representations, Pasadena, California: PhD thesis, *California Institute of Technology*, 1998.

"Intentionally left blank"

Appendix A: Jacobian matrix decomposition

For the implementation of the Roe's approximate Riemann problem solver the Jacobian matrix of the convective flux vector \vec{H}^{inv} has to be computed; it is analyzed via the eigenvalue decomposition as follows [Hir90, Lan98]

$$\underline{A} = \underline{T} \underline{\Lambda} \underline{T}^{-1} \quad (\text{A.1})$$

where $\underline{\Lambda}$ is a 5x5 diagonal matrix, whose entries are the eigenvalues of the Jacobian matrix \underline{A} , defined as [Hir90]

$$\underline{\Lambda} = d \text{ ai } \left\{ \hat{V}_n |\vec{n}|, \hat{V}_n |\vec{n}|, \hat{V}_n |\vec{n}|, (\hat{V}_n + c) |\vec{n}|, (\hat{V}_n - c) |\vec{n}| \right\} \quad (\text{A.2})$$

while \underline{T} is a matrix, including the eigenvectors of the Jacobian matrix \underline{A} [Hir90]

$$\underline{T} = \begin{bmatrix} \hat{n}_x & \hat{n}_y & \hat{n}_z & C & C \\ \hat{n}_x u & \hat{n}_y u - \hat{n}_z \rho & \hat{n}_z u + \hat{n}_y \rho & C(u + c \hat{n}_x) & C(u - c \hat{n}_x) \\ \hat{n}_x v + \hat{n}_z \rho & \hat{n}_y v & \hat{n}_z v - \hat{n}_x \rho & C(v + c \hat{n}_y) & C(v - c \hat{n}_y) \\ \hat{n}_x w - \hat{n}_y \rho & \hat{n}_y w + \hat{n}_x \rho & \hat{n}_z w & C(w + c \hat{n}_z) & C(w - c \hat{n}_z) \\ \frac{\vec{V}^2}{2} \hat{n}_x + \rho X & \frac{\vec{V}^2}{2} \hat{n}_y + \rho Y & \frac{\vec{V}^2}{2} \hat{n}_z + \rho Z & C(H + c \hat{V}_n) & C(H - c \hat{V}_n) \end{bmatrix} \quad (\text{A.3})$$

and \underline{T}^{-1} is its inverse matrix [Hir90]

$$\underline{T}^{-1} = \begin{bmatrix} B \hat{n}_x - \frac{X}{\rho} & \frac{(\gamma-1) \hat{n}_x u}{c^2} & \frac{(\gamma-1) \hat{n}_x v}{c^2} + \frac{\hat{n}_z}{\rho} & \frac{(\gamma-1) \hat{n}_x w}{c^2} - \frac{\hat{n}_y}{\rho} & -\frac{(\gamma-1) \hat{n}_x}{c^2} \\ B \hat{n}_y - \frac{Y}{\rho} & \frac{(\gamma-1) \hat{n}_y u}{c^2} - \frac{\hat{n}_z}{\rho} & \frac{(\gamma-1) \hat{n}_y v}{c^2} & \frac{(\gamma-1) \hat{n}_y w}{c^2} + \frac{\hat{n}_x}{\rho} & -\frac{(\gamma-1) \hat{n}_y}{c^2} \\ B \hat{n}_z - \frac{Z}{\rho} & \frac{(\gamma-1) \hat{n}_z u}{c^2} + \frac{\hat{n}_y}{\rho} & \frac{(\gamma-1) \hat{n}_z v}{c^2} - \frac{\hat{n}_x}{\rho} & \frac{(\gamma-1) \hat{n}_z w}{c^2} & -\frac{(\gamma-1) \hat{n}_z}{c^2} \\ \frac{c}{\rho} \left(\frac{(\gamma-1) \vec{V}^2}{2c^2} - \frac{\hat{V}_n}{c} \right) & \left(\hat{n}_x - \frac{(\gamma-1)u}{c} \right) \frac{1}{\rho} & \left(\hat{n}_y - \frac{(\gamma-1)v}{c} \right) \frac{1}{\rho} & \left(\hat{n}_z - \frac{(\gamma-1)w}{c} \right) \frac{1}{\rho} & \frac{(\gamma-1)}{\rho c} \\ \frac{c}{\rho} \left(\frac{(\gamma-1) \vec{V}^2}{2c^2} + \frac{\hat{V}_n}{c} \right) & \left(-\hat{n}_x - \frac{(\gamma-1)u}{c} \right) \frac{1}{\rho} & \left(-\hat{n}_y - \frac{(\gamma-1)v}{c} \right) \frac{1}{\rho} & \left(-\hat{n}_z - \frac{(\gamma-1)w}{c} \right) \frac{1}{\rho} & \frac{(\gamma-1)}{\rho c} \end{bmatrix} \quad (\text{A.4})$$

where u , v and w are the components of velocity \vec{V} , $\vec{n} = (\hat{n}_x, \hat{n}_y, \hat{n}_z)$ is the unit normal vector and $\hat{V}_n = \vec{V} \cdot \vec{n}$ is the value of corresponding velocity. The terms C , H , X , Y , Z and B are auxiliary values defined as:

$$\begin{aligned}
 C &= \frac{\rho}{2c} \\
 H &= \frac{\vec{V}^2}{2} + \frac{c^2}{\gamma - 1} \\
 X &= \hat{n}_z v - \hat{n}_y w \\
 Y &= \hat{n}_x w - \hat{n}_z u \\
 Z &= \hat{n}_y u - \hat{n}_x v \\
 B &= 1 - \frac{(\gamma - 1)\vec{V}^2}{2c^2}
 \end{aligned} \tag{A.5}$$

Depending on the eigenvalue λ , the Jacobian matrix for the Roe's approximate Riemann solver is calculated as [Hir90]

$$\begin{aligned}
 \underline{A}^{\pm} &= \underline{T} \underline{\Lambda}^{\pm} \underline{T}^{-1}, \quad \underline{\Lambda}^{\pm} = \text{diag} \{ \lambda_i^{\pm} \} \\
 |\underline{A}| &= \underline{T} |\underline{\Lambda}| \underline{T}^{-1}, \quad |\underline{\Lambda}| = \text{diag} \{ |\lambda_i| \}
 \end{aligned} \tag{A.6}$$

while the eigenvalues λ as:

$$\begin{aligned}
 \lambda_i^+ &= \max(\lambda_i, 0), \quad i = 1, \dots, 5 \\
 \lambda_i^- &= \min(\lambda_i, 0), \quad i = 1, \dots, 5
 \end{aligned} \tag{A.7}$$

Appendix B: Newton linearization of the viscous fluxes

B.1. Flow model

The flow model's viscous fluxes at the midpoint of an edge PQ with a vector $\vec{n}_{PQ} = (n_{PQ,x}, n_{PQ,y}, n_{PQ,z})$ of the interface area of adjacent control volumes of P and Q are defined as

$$\vec{\Phi}_{PQ}^{vis} = n_{PQ,x} \vec{\Phi}_{PQ,x}^{vis} + n_{PQ,y} \vec{\Phi}_{PQ,y}^{vis} + n_{PQ,z} \vec{\Phi}_{PQ,z}^{vis} = (\Phi_1, \Phi_2, \Phi_3, \Phi_4, \Phi_5)^T \quad (B.1)$$

where $\Phi_1, \Phi_2, \Phi_3, \Phi_4, \Phi_5$ are described as follows

$$\Phi_1 = 0 \quad (B.2)$$

$$\begin{aligned} \Phi_2 &= n_{PQ,x} \tau_{xx} + n_{PQ,y} \tau_{xy} + n_{PQ,z} \tau_{xz} \\ &= n_{PQ,x} \left[\frac{2}{3} \frac{\mu}{\text{Re}} \left(2 \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} - \frac{\partial w}{\partial z} \right) \right] + n_{PQ,y} \frac{\mu}{\text{Re}} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) + n_{PQ,z} \frac{\mu}{\text{Re}} \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) \\ &= a_{11} \frac{\partial u}{\partial x} + a_{12} \frac{\partial u}{\partial y} + a_{13} \frac{\partial u}{\partial z} + a_{21} \frac{\partial v}{\partial x} + a_{22} \frac{\partial v}{\partial y} + a_{23} \frac{\partial v}{\partial z} + a_{31} \frac{\partial w}{\partial x} + a_{32} \frac{\partial w}{\partial y} + a_{33} \frac{\partial w}{\partial z} \end{aligned} \quad (B.3)$$

$$\begin{aligned} \Phi_3 &= n_{PQ,x} \tau_{xy} + n_{PQ,y} \tau_{yy} + n_{PQ,z} \tau_{yz} \\ &= b_{11} \frac{\partial u}{\partial x} + b_{12} \frac{\partial u}{\partial y} + b_{13} \frac{\partial u}{\partial z} + b_{21} \frac{\partial v}{\partial x} + b_{22} \frac{\partial v}{\partial y} + b_{23} \frac{\partial v}{\partial z} + b_{31} \frac{\partial w}{\partial x} + b_{32} \frac{\partial w}{\partial y} + b_{33} \frac{\partial w}{\partial z} \end{aligned} \quad (B.4)$$

$$\begin{aligned} \Phi_4 &= n_{PQ,x} \tau_{xz} + n_{PQ,y} \tau_{yz} + n_{PQ,z} \tau_{zz} \\ &= c_{11} \frac{\partial u}{\partial x} + c_{12} \frac{\partial u}{\partial y} + c_{13} \frac{\partial u}{\partial z} + c_{21} \frac{\partial v}{\partial x} + c_{22} \frac{\partial v}{\partial y} + c_{23} \frac{\partial v}{\partial z} + c_{31} \frac{\partial w}{\partial x} + c_{32} \frac{\partial w}{\partial y} + c_{33} \frac{\partial w}{\partial z} \end{aligned} \quad (B.5)$$

$$\begin{aligned} \Phi_5 &= n_{PQ,x} (u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + q_x) + n_{PQ,y} (u\tau_{xy} + v\tau_{yy} + w\tau_{yz} + q_y) \\ &\quad + n_{PQ,z} (u\tau_{xz} + v\tau_{yz} + w\tau_{zz} + q_z) \\ &= d_{11} \frac{\partial u}{\partial x} + d_{12} \frac{\partial u}{\partial y} + d_{13} \frac{\partial u}{\partial z} + d_{21} \frac{\partial v}{\partial x} + d_{22} \frac{\partial v}{\partial y} + d_{23} \frac{\partial v}{\partial z} \\ &\quad + d_{31} \frac{\partial w}{\partial x} + d_{32} \frac{\partial w}{\partial y} + d_{33} \frac{\partial w}{\partial z} + e_1 \frac{\partial T}{\partial x} + e_2 \frac{\partial T}{\partial y} + e_3 \frac{\partial T}{\partial z} \end{aligned} \quad (B.6)$$

while their coefficients a_{ij} , b_{ij} , c_{ij} , d_{ij} and e_i are computed as:

$$\begin{aligned} a_{11} &= \frac{4}{3} \frac{\mu}{\text{Re}} n_{PQ,x} & a_{12} &= \frac{\mu}{\text{Re}} n_{PQ,y} & a_{13} &= \frac{\mu}{\text{Re}} n_{PQ,z} & a_{21} &= \frac{\mu}{\text{Re}} n_{PQ,y} \\ a_{22} &= -\frac{2}{3} \frac{\mu}{\text{Re}} n_{PQ,x} & a_{23} &= 0 & a_{31} &= \frac{\mu}{\text{Re}} n_{PQ,z} & a_{32} &= 0 & a_{33} &= -\frac{2}{3} \frac{\mu}{\text{Re}} n_{PQ,x} \end{aligned}$$

$$\begin{aligned}
b_{11} &= -\frac{2}{3} \frac{\mu}{\text{Re}} n_{PQ,y} & b_{12} &= \frac{\mu}{\text{Re}} n_{PQ,x} & b_{13} &= 0 & b_{21} &= \frac{\mu}{\text{Re}} n_{PQ,x} \\
b_{22} &= \frac{4}{3} \frac{\mu}{\text{Re}} n_{PQ,y} & b_{23} &= \frac{\mu}{\text{Re}} n_{PQ,z} & b_{31} &= 0 & b_{32} &= \frac{\mu}{\text{Re}} n_{PQ,z} & b_{33} &= -\frac{2}{3} \frac{\mu}{\text{Re}} n_{PQ,y} \\
c_{11} &= -\frac{2}{3} \frac{\mu}{\text{Re}} n_{PQ,z} & c_{12} &= 0 & c_{13} &= \frac{\mu}{\text{Re}} n_{PQ,x} & c_{21} &= 0 \\
c_{22} &= -\frac{2}{3} \frac{\mu}{\text{Re}} n_{PQ,z} & c_{23} &= \frac{\mu}{\text{Re}} n_{PQ,y} & c_{31} &= \frac{\mu}{\text{Re}} n_{PQ,x} & c_{32} &= \frac{\mu}{\text{Re}} n_{PQ,y} & c_{33} &= \frac{4}{3} \frac{\mu}{\text{Re}} n_{PQ,z} \\
d_{11} &= \frac{2}{3} \frac{\mu}{\text{Re}} (2n_{PQ,x}u - n_{PQ,y}v - n_{PQ,z}w) & d_{12} &= \frac{\mu}{\text{Re}} (n_{PQ,x}v + n_{PQ,y}u) & d_{13} &= \frac{\mu}{\text{Re}} (n_{PQ,x}w + n_{PQ,z}u) \\
d_{21} &= \frac{\mu}{\text{Re}} (n_{PQ,x}v + n_{PQ,y}u) & d_{22} &= \frac{2}{3} \frac{\mu}{\text{Re}} (2n_{PQ,y}v - n_{PQ,x}u - n_{PQ,z}w) & d_{23} &= \frac{\mu}{\text{Re}} (n_{PQ,z}v + n_{PQ,y}w) \\
d_{31} &= \frac{\mu}{\text{Re}} (n_{PQ,x}w + n_{PQ,z}u) & d_{32} &= \frac{\mu}{\text{Re}} (n_{PQ,z}v + n_{PQ,y}w) & d_{33} &= \frac{2}{3} \frac{\mu}{\text{Re}} (2n_{PQ,z}w - n_{PQ,x}u - n_{PQ,y}v) \\
e_1 &= n_{PQ,x} \frac{\gamma\mu}{\text{Pr Re}} & e_2 &= n_{PQ,y} \frac{\gamma\mu}{\text{Pr Re}} & e_3 &= n_{PQ,z} \frac{\gamma\mu}{\text{Pr Re}}
\end{aligned} \tag{B.7}$$

B.1.1. Linearization based on the edge-dual volume method

In case of employing the edge-dual volume-based methodology for the computation of the required derivatives, equations (B.3) - (B.6) are transformed as [Kal96, Lyg14a, Lyg15]

$$\begin{aligned}
\Phi_2 &= a_{11} \frac{1}{V_e} \int u \hat{n}_x ds + a_{12} \frac{1}{V_e} \int u \hat{n}_y ds + a_{13} \frac{1}{V_e} \int u \hat{n}_z ds \\
&+ a_{21} \frac{1}{V_e} \int v \hat{n}_x ds + a_{22} \frac{1}{V_e} \int v \hat{n}_y ds + a_{23} \frac{1}{V_e} \int v \hat{n}_z ds \\
&+ a_{31} \frac{1}{V_e} \int w \hat{n}_x ds + a_{32} \frac{1}{V_e} \int w \hat{n}_y ds + a_{33} \frac{1}{V_e} \int w \hat{n}_z ds \\
&= \frac{1}{V_e} \int \sum_{j=1}^3 (a_{1j} \hat{n}_{x_j}) ds + \frac{1}{V_e} \int \sum_{j=1}^3 (a_{2j} \hat{n}_{y_j}) ds + \frac{1}{V_e} \int \sum_{j=1}^3 (a_{3j} \hat{n}_{z_j}) ds \\
&= \frac{1}{V_e} \sum_{j=1}^3 a_{1j} \sum_{k=1}^m n_{x_j,k} u_k + \frac{1}{V_e} \sum_{j=1}^3 a_{2j} \sum_{k=1}^m n_{y_j,k} v_k + \frac{1}{V_e} \sum_{j=1}^3 a_{3j} \sum_{k=1}^m n_{z_j,k} w_k \\
&= \frac{1}{V_e} \sum_{j=1}^3 a_{1j} \sum_{k=1}^m n_{x_j,k} \sum_{l=1}^{L_k} \frac{u_l}{L_k} + \frac{1}{V_e} \sum_{j=1}^3 a_{2j} \sum_{k=1}^m n_{y_j,k} \sum_{l=1}^{L_k} \frac{v_l}{L_k} + \frac{1}{V_e} \sum_{j=1}^3 a_{3j} \sum_{k=1}^m n_{z_j,k} \sum_{l=1}^{L_k} \frac{w_l}{L_k}
\end{aligned} \tag{B.8}$$

$$\begin{aligned}
\Phi_3 &= \frac{1}{V_e} \sum_{j=1}^3 b_{1j} \sum_{k=1}^m n_{x_j,k} \sum_{l=1}^{L_k} \frac{u_l}{L_k} + \frac{1}{V_e} \sum_{j=1}^3 b_{2j} \sum_{k=1}^m n_{y_j,k} \sum_{l=1}^{L_k} \frac{v_l}{L_k} \\
&+ \frac{1}{V_e} \sum_{j=1}^3 b_{3j} \sum_{k=1}^m n_{z_j,k} \sum_{l=1}^{L_k} \frac{w_l}{L_k}
\end{aligned} \tag{B.9}$$

$$\begin{aligned}\Phi_4 &= \frac{1}{V_e} \sum_{j=1}^3 c_{1j} \sum_{k=1}^m n_{x_j,k} \sum_{l=1}^{L_k} \frac{u_l}{L_k} + \frac{1}{V_e} \sum_{j=1}^3 c_{2j} \sum_{k=1}^m n_{x_j,k} \sum_{l=1}^{L_k} \frac{v_l}{L_k} \\ &+ \frac{1}{V_e} \sum_{j=1}^3 c_{3j} \sum_{k=1}^m n_{x_j,k} \sum_{l=1}^{L_k} \frac{w_l}{L_k}\end{aligned}\quad (\text{B.10})$$

$$\begin{aligned}\Phi_5 &= \frac{1}{V_e} \sum_{j=1}^3 d_{1j} \sum_{k=1}^m n_{x_j,k} \sum_{l=1}^{L_k} \frac{u_l}{L_k} + \frac{1}{V_e} \sum_{j=1}^3 d_{2j} \sum_{k=1}^m n_{x_j,k} \sum_{l=1}^{L_k} \frac{v_l}{L_k} \\ &+ \frac{1}{V_e} \sum_{j=1}^3 d_{3j} \sum_{k=1}^m n_{x_j,k} \sum_{l=1}^{L_k} \frac{w_l}{L_k} + \frac{1}{V_e} \sum_{j=1}^3 e_j \sum_{k=1}^m n_{x_j,k} \sum_{l=1}^{L_k} \frac{T_l}{L_k}\end{aligned}\quad (\text{B.11})$$

where n_{x_j} accounts for the components of the normal vector to each face k with nodes L_k of the PQ edge-dual volume. Based on these transformations, equation (B.1) is redefined as:

$$\vec{\Phi}_{PQ}^{vis} = \frac{1}{V_e} \sum_{k=1}^m \sum_{l=1}^{L_k} \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \sum_{j=1}^3 \frac{\alpha_{1j} n_{x_j,k}}{L_k} & \sum_{j=1}^3 \frac{\alpha_{2j} n_{x_j,k}}{L_k} & \sum_{j=1}^3 \frac{\alpha_{3j} n_{x_j,k}}{L_k} & 0 \\ 0 & \sum_{j=1}^3 \frac{b_{1j} n_{x_j,k}}{L_k} & \sum_{j=1}^3 \frac{b_{2j} n_{x_j,k}}{L_k} & \sum_{j=1}^3 \frac{b_{3j} n_{x_j,k}}{L_k} & 0 \\ 0 & \sum_{j=1}^3 \frac{c_{1j} n_{x_j,k}}{L_k} & \sum_{j=1}^3 \frac{c_{2j} n_{x_j,k}}{L_k} & \sum_{j=1}^3 \frac{c_{3j} n_{x_j,k}}{L_k} & 0 \\ 0 & \sum_{j=1}^3 \frac{d_{1j} n_{x_j,k}}{L_k} & \sum_{j=1}^3 \frac{d_{2j} n_{x_j,k}}{L_k} & \sum_{j=1}^3 \frac{d_{3j} n_{x_j,k}}{L_k} & \sum_{j=1}^3 \frac{e_j n_{x_j,k}}{L_k} \end{bmatrix}}_{\vec{M}_{kl}} \underbrace{\begin{bmatrix} \rho_l \\ u_l \\ v_l \\ w_l \\ T_l \end{bmatrix}}_{\vec{V}_l} = \frac{1}{V_e} \sum_{k=1}^m \sum_{l=1}^{L_k} \vec{M}_{kl} \vec{V}_l \quad (\text{B.12})$$

Considering the new variables' vector \vec{V} , the viscous flux at the midpoint of an edge PQ at the time step $n+1$ is written as:

$$\vec{\Phi}_{PQ}^{vis,n+1} = \vec{\Phi}_{PQ}^{vis,n} + \frac{1}{V_e} \sum_{k=1}^m \sum_{l=1}^{L_k} \vec{M}_{k,l} \delta \vec{V}_l \quad (\text{B.13})$$

The last equation has to be further analyzed to include the conservative variables' vector of the flow equations \vec{W} ; utilizing the following equations (B.14) and (B.15), which derive the appropriate transformation vector \vec{Q} , the final equation (B.16) is obtained:

$$\begin{aligned}
V_1 &= W_1, & V_2 &= \frac{W_2}{W_1}, & V_3 &= \frac{W_3}{W_1}, \\
V_4 &= \frac{W_4}{W_1}, & V_5 &= \frac{W_5}{W_1} - \frac{W_2^2 + W_3^2 + W_4^2}{2W_1^2}
\end{aligned} \tag{B.14}$$

$$\delta \vec{V}^{n+1} = \left(\frac{\partial \vec{V}}{\partial \vec{W}} \right) \delta \vec{W} = \vec{Q} \delta \vec{W} \tag{B.15}$$

$$\vec{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -\frac{W_2}{W_1^2} & \frac{1}{W_1} & 0 & 0 & 0 \\ -\frac{W_3}{W_1^2} & 0 & \frac{1}{W_1} & 0 & 0 \\ -\frac{W_4}{W_1^2} & 0 & 0 & \frac{1}{W_1} & 0 \\ -\frac{W_5}{W_1^2} + \frac{W_2^2 + W_3^2 + W_4^2}{W_1^3} & -\frac{W_2}{W_1^2} & -\frac{W_3}{W_1^2} & -\frac{W_4}{W_1^2} & \frac{1}{W_1} \end{bmatrix} \tag{B.16}$$

Thus, equation (B.13) for the examined edge PQ is written in the following formulation, deriving the required per Newton linearized matrices \underline{C}_d^n , \underline{C}_o^n , \underline{C}_{adj}^n , based on the values of conservative variables at the end-points P and Q , and the rest nodes R of the corresponding edge-dual volume as well:

$$\vec{\Phi}_{PQ}^{vis,n+1} = \vec{\Phi}_{PQ}^{vis,n} + \frac{1}{V_e} \sum_{k=1}^m \sum_{l=1}^{L_k} \vec{M}_{k,l} \vec{Q}_l \delta \vec{W}_l \tag{B.17}$$

B.1.2. Linearization based on the nodal-averaging scheme

If the nodal-averaging method (equations (2.104)-(2.106)) is selected, the gradients at the middle of an edge PQ of the grid are evaluated as follows [Bla01, Lyg14a, Lyg15]

$$(\nabla U)_{PQ} = (\nabla \bar{U})_{PQ} - \left[(\nabla \bar{U})_{PQ} \cdot \vec{r}_{PQ} - \left(\frac{\partial U}{\partial l} \right)_{PQ} \right] \vec{r}_{PQ} \tag{B.18}$$

while omitting the directional derivative, the previous equation becomes:

$$(\nabla U)_{PQ} \approx (\nabla \bar{U})_{PQ} = \frac{1}{2} \left((\nabla U)_P + (\nabla U)_Q \right) \tag{B.19}$$

Utilizing equation (2.91), according to which the gradients of the flow variables are obtained at each computational node, the previous relation for a variable u_i is further transformed as

$$\begin{aligned}
 \frac{\partial u_{iPQ}}{\partial x_j} &= \frac{1}{2} \left(\frac{\partial u_{iP}}{\partial x_j} + \frac{\partial u_{iQ}}{\partial x_j} \right) = \frac{1}{2} \left[\frac{1}{V_P} \left(\sum_{R \in K_N(P)} \frac{1}{2} (u_{iP} + u_{iR}) n_{PR_j} + \sum_{(R_{out} \in \partial C_P \cap \Gamma)} u_{iP} n_{out j} \right) \right] + \\
 &\frac{1}{2} \left[\frac{1}{V_Q} \left(\sum_{T \in K_N(Q)} \frac{1}{2} (u_{iQ} + u_{iT}) n_{QT_j} + \sum_{(T_{out} \in \partial C_P \cap \Gamma)} u_{iQ} n_{out j} \right) \right] = \\
 &\underbrace{\left(\frac{1}{4V_P} \sum_{R \in K_N(P)} n_{PR_j} + \frac{1}{2V_P} \sum_{(R_{out} \in \partial C_P \cap \Gamma)} n_{out j} \right)}_{D_{Pj}} u_{iP} + \underbrace{\left(\frac{1}{4V_Q} \sum_{T \in K_N(Q)} n_{QT_j} + \frac{1}{2V_Q} \sum_{(T_{out} \in \partial C_P \cap \Gamma)} n_{out j} \right)}_{D_{Qj}} u_{iQ} + \\
 &\underbrace{\frac{1}{4V_P} \sum_{R \in K_N(P)} n_{PR_j} u_{iR} + \frac{1}{4V_Q} \sum_{T \in K_N(Q)} n_{QT_j} u_{iT}}_{E_{PQw_j}}
 \end{aligned} \tag{B.20}$$

where P and Q are the end-points of the examined edge PQ , while R and T are their neighbouring nodes, connected with an edge, respectively. Considering the previous expression, equations (B.3) - (B.6) are described as

$$\begin{aligned}
 \Phi_2 &= a_{11} \frac{\partial u}{\partial x} + a_{12} \frac{\partial u}{\partial y} + a_{13} \frac{\partial u}{\partial z} + a_{21} \frac{\partial v}{\partial x} + a_{22} \frac{\partial v}{\partial y} + a_{23} \frac{\partial v}{\partial z} + a_{31} \frac{\partial w}{\partial x} + a_{32} \frac{\partial w}{\partial y} + a_{33} \frac{\partial w}{\partial z} = \\
 &\sum_{j=1}^3 a_{1j} \frac{\partial u_{PQ}}{\partial x_j} + \sum_{j=1}^3 a_{2j} \frac{\partial v_{PQ}}{\partial x_j} + \sum_{j=1}^3 a_{3j} \frac{\partial w_{PQ}}{\partial x_j} = \\
 &\sum_{j=1}^3 a_{1j} (D_{Pj} u_P + D_{Qj} u_Q + E_{PQw_j}) + \sum_{j=1}^3 a_{2j} (D_{Pj} v_P + D_{Qj} v_Q + E_{PQv_j}) + \\
 &\sum_{j=1}^3 a_{3j} (D_{Pj} w_P + D_{Qj} w_Q + E_{PQw_j})
 \end{aligned} \tag{B.21}$$

$$\begin{aligned}
 \Phi_3 &= \sum_{j=1}^3 b_{1j} (D_{Pj} u_P + D_{Qj} u_Q + E_{PQw_j}) + \sum_{j=1}^3 b_{2j} (D_{Pj} v_P + D_{Qj} v_Q + E_{PQv_j}) + \\
 &\sum_{j=1}^3 b_{3j} (D_{Pj} w_P + D_{Qj} w_Q + E_{PQw_j})
 \end{aligned} \tag{B.22}$$

$$\begin{aligned}
 \Phi_4 &= \sum_{j=1}^3 c_{1j} (D_{Pj} u_P + D_{Qj} u_Q + E_{PQw_j}) + \sum_{j=1}^3 c_{2j} (D_{Pj} v_P + D_{Qj} v_Q + E_{PQv_j}) + \\
 &\sum_{j=1}^3 c_{3j} (D_{Pj} w_P + D_{Qj} w_Q + E_{PQw_j})
 \end{aligned} \tag{B.23}$$

$$\begin{aligned}
 \Phi_5 &= \sum_{j=1}^3 d_{1j} (D_{Pj} u_P + D_{Qj} u_Q + E_{PQw_j}) + \sum_{j=1}^3 d_{2j} (D_{Pj} v_P + D_{Qj} v_Q + E_{PQv_j}) + \\
 &\sum_{j=1}^3 d_{3j} (D_{Pj} w_P + D_{Qj} w_Q + E_{PQw_j}) + \sum_{j=1}^3 e_j (D_{Pj} T_P + D_{Qj} T_Q + E_{PQT_j})
 \end{aligned} \tag{B.24}$$

while equation (B.1) is per Newton linearized as:

$$\begin{aligned} \vec{\Phi}_{PQ}^{vis} = & \sum_{j=1}^3 \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & a_{1j}D_{Pj} & a_{2j}D_{Pj} & a_{3j}D_{Pj} & 0 \\ 0 & b_{1j}D_{Pj} & b_{2j}D_{Pj} & b_{3j}D_{Pj} & 0 \\ 0 & c_{1j}D_{Pj} & c_{2j}D_{Pj} & c_{3j}D_{Pj} & 0 \\ 0 & d_{1j}D_{Pj} & d_{2j}D_{Pj} & d_{3j}D_{Pj} & e_jD_{Pj} \end{bmatrix}}_{\vec{M}_P} \begin{bmatrix} \rho_P \\ u_P \\ v_P \\ w_P \\ T_P \end{bmatrix} + \\ & \sum_{j=1}^3 \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & a_{1j}D_{Qj} & a_{2j}D_{Qj} & a_{3j}D_{Qj} & 0 \\ 0 & b_{1j}D_{Qj} & b_{2j}D_{Qj} & b_{3j}D_{Qj} & 0 \\ 0 & c_{1j}D_{Qj} & c_{2j}D_{Qj} & c_{3j}D_{Qj} & 0 \\ 0 & d_{1j}D_{Qj} & d_{2j}D_{Qj} & d_{3j}D_{Qj} & e_jD_{Qj} \end{bmatrix}}_{\vec{M}_Q} \begin{bmatrix} \rho_Q \\ u_Q \\ v_Q \\ w_Q \\ T_Q \end{bmatrix} + \sum_{j=1}^3 \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & a_{1j}E_{PQuj} & a_{2j}E_{PQvj} & a_{3j}E_{PQwj} & 0 \\ 0 & b_{1j}E_{PQuj} & b_{2j}E_{PQvj} & b_{3j}E_{PQwj} & 0 \\ 0 & c_{1j}E_{PQuj} & c_{2j}E_{PQvj} & c_{3j}E_{PQwj} & 0 \\ 0 & d_{1j}E_{PQuj} & d_{2j}E_{PQvj} & d_{3j}E_{PQwj} & e_jE_{PQTj} \end{bmatrix}}_{\vec{E}'_{PQ}} \end{aligned} \quad (B.25)$$

Finally, the transformation matrix \vec{Q} is utilized similarly to the previous section, in order equation (B.25) to include conservative variables' vector as

$$\vec{\Phi}_{PQ}^{vis,n+1} = \vec{\Phi}_{PQ}^{vis,n} + \sum_{j=1}^3 \vec{M}_P \vec{Q}_P \delta \vec{W}_P + \sum_{j=1}^3 \vec{M}_Q \vec{Q}_Q \delta \vec{W}_Q + \sum_{j=1}^3 \vec{E}'_{PQ\delta} \quad (B.26)$$

where the matrix $\vec{E}'_{PQ\delta}$ includes also the transformation matrix \vec{Q} .

B.2. Turbulence models

For the diffusive terms of the incorporated turbulence models, the same procedure is followed; the viscous fluxes at the midpoint of an edge PQ are defined similarly to these of the flow PDE's (equations (B.2)-(B.6)) as

$$\begin{aligned} \Phi_1 &= a_x \frac{\partial k}{\partial x} + a_y \frac{\partial k}{\partial y} + a_z \frac{\partial k}{\partial z} \\ \Phi_2 &= b_x \frac{\partial \varphi}{\partial x} + b_y \frac{\partial \varphi}{\partial y} + b_z \frac{\partial \varphi}{\partial z} \end{aligned} \quad (B.27)$$

where the variable φ corresponds to ε if k - ε model is employed and to ω if k - ω or SST model is implemented instead. The coefficients in the previous equation a_{x_j} and b_{x_j} are computed for k - ε model as

$$\begin{aligned} a_{x_j} &= \left(\frac{\mu}{\text{Re}} + \frac{\mu_t}{\sigma_k} \right) n_{PQ,x_j} \\ b_{x_j} &= \left(\frac{\mu}{\text{Re}} + \frac{\mu_t}{\sigma_\varepsilon} \right) n_{PQ,x_j} \end{aligned} \quad (B.28)$$

while for k - ω and SST models as:

$$\begin{aligned} a_{x_j} &= \left(\frac{\mu}{\text{Re}} + \mu_t \sigma_k \right) n_{PQ, x_j} \\ b_{x_j} &= \left(\frac{\mu}{\text{Re}} + \mu_t \sigma_\omega \right) n_{PQ, x_j} \end{aligned} \quad (\text{B.29})$$

B.2.1. Linearization based on the edge-dual volume method

If the edge-dual volume-based methodology is implemented for the computation of the required derivatives, equation (B.27) is defined similarly to equations (B.8)-(B.11) as

$$\begin{aligned} \Phi_1 &= \sum_{j=1}^3 a_{x_j} \frac{\partial k}{\partial x_j} = \sum_{j=1}^3 a_{x_j} \frac{1}{V_e} \int k \hat{n}_{x_j} ds = \frac{1}{V_e} \sum_{k=1}^m \sum_{l=1}^{L_k} \sum_{j=1}^3 \frac{a_{x_j} n_{x_j, k}}{L_k} k \\ \Phi_2 &= \sum_{j=1}^3 b_{x_j} \frac{\partial \varphi}{\partial x_j} = \sum_{j=1}^3 b_{x_j} \frac{1}{V_e} \int \varphi \hat{n}_{x_j} ds = \frac{1}{V_e} \sum_{k=1}^m \sum_{l=1}^{L_k} \sum_{j=1}^3 \frac{b_{x_j} n_{x_j, k}}{L_k} \varphi \end{aligned} \quad (\text{B.30})$$

while considering additionally the corresponding transformation vector, its final linearized formulation is the following:

$$\bar{\Phi}_{PQ}^{vis} = \frac{1}{V_e} \sum_{k=1}^m \sum_{l=1}^{L_k} \begin{bmatrix} \sum_{j=1}^3 \frac{a_{x_j} n_{x_j, k}}{L_k} & 0 \\ 0 & \sum_{j=1}^3 \frac{b_{x_j} n_{x_j, k}}{L_k} \end{bmatrix} \begin{bmatrix} \frac{1}{\rho_l} & 0 \\ 0 & \frac{1}{\rho_l} \end{bmatrix} \begin{bmatrix} \rho_l k_l \\ \rho_l \varphi_l \end{bmatrix} \quad (\text{B.31})$$

B.2.2. Linearization based on the nodal-averaging scheme

In case the nodal-averaging method is used for the evaluation of gradients, the diffusive fluxes of the incorporated turbulence models are linearized in the same way to equation (B.25) as:

$$\begin{aligned} \bar{\Phi}_{PQ}^{vis} &= \sum_{j=1}^3 \begin{bmatrix} a_{x_j} D_{Pj} & 0 \\ 0 & b_{x_j} D_{Pj} \end{bmatrix} \begin{bmatrix} \frac{1}{\rho_p} & 0 \\ 0 & \frac{1}{\rho_p} \end{bmatrix} \begin{bmatrix} \rho_p k_p \\ \rho_p \varphi_p \end{bmatrix} + \\ &\sum_{j=1}^3 \begin{bmatrix} a_{x_j} D_{Qj} & 0 \\ 0 & b_{x_j} D_{Qj} \end{bmatrix} \begin{bmatrix} \frac{1}{\rho_q} & 0 \\ 0 & \frac{1}{\rho_q} \end{bmatrix} \begin{bmatrix} \rho_q k_q \\ \rho_q \varphi_q \end{bmatrix} + \sum_{j=1}^3 \begin{bmatrix} a_{x_j} E_{t, PQij} & 0 \\ 0 & b_{x_j} E_{t, PQij} \end{bmatrix} \end{aligned} \quad (\text{B.32})$$

"Intentionally left blank"

Appendix C: Newton linearization of the free-slip convective fluxes

The free-slip convective flux vector, required at inviscid solid wall or symmetry boundary surfaces, is computed as follows

$$\vec{H}_{freeslip} = \begin{pmatrix} 0 & pn_{out,x} & pn_{out,y} & pn_{out,z} & 0 \end{pmatrix}^T \quad (C.1)$$

while its diagonal contribution for the point-implicit scheme is defined by implementing the chain rule as

$$\frac{\partial \vec{H}_{freeslip}}{\partial \vec{W}} = \frac{\partial \vec{H}_{freeslip}}{\partial \vec{U}} \cdot \frac{\partial \vec{U}}{\partial \vec{W}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & n_{out,x} \\ 0 & 0 & 0 & 0 & n_{out,y} \\ 0 & 0 & 0 & 0 & n_{out,z} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial \rho}{\partial \rho} & \frac{\partial \rho}{\partial \rho u} & \frac{\partial \rho}{\partial \rho v} & \frac{\partial \rho}{\partial \rho w} & \frac{\partial \rho}{\partial \rho E} \\ \frac{\partial u}{\partial \rho} & \frac{\partial u}{\partial \rho u} & \frac{\partial u}{\partial \rho v} & \frac{\partial u}{\partial \rho w} & \frac{\partial u}{\partial \rho E} \\ \frac{\partial v}{\partial \rho} & \frac{\partial v}{\partial \rho u} & \frac{\partial v}{\partial \rho v} & \frac{\partial v}{\partial \rho w} & \frac{\partial v}{\partial \rho E} \\ \frac{\partial w}{\partial \rho} & \frac{\partial w}{\partial \rho u} & \frac{\partial w}{\partial \rho v} & \frac{\partial w}{\partial \rho w} & \frac{\partial w}{\partial \rho E} \\ \frac{\partial p}{\partial \rho} & \frac{\partial p}{\partial \rho u} & \frac{\partial p}{\partial \rho v} & \frac{\partial p}{\partial \rho w} & \frac{\partial p}{\partial \rho E} \end{bmatrix} \quad (C.2)$$

where \vec{W} is the conservative variables' vector and \vec{U} is the primitive variables' vector. Thus, the free-slip contribution becomes as

$$\frac{\partial \vec{H}_{freeslip}}{\partial \vec{W}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \frac{\partial p}{\partial \rho} n_{out,x} & \frac{\partial p}{\partial \rho u} n_{out,x} & \frac{\partial p}{\partial \rho v} n_{out,x} & \frac{\partial p}{\partial \rho w} n_{out,x} & \frac{\partial p}{\partial \rho E} n_{out,x} \\ \frac{\partial p}{\partial \rho} n_{out,y} & \frac{\partial p}{\partial \rho u} n_{out,y} & \frac{\partial p}{\partial \rho v} n_{out,y} & \frac{\partial p}{\partial \rho w} n_{out,y} & \frac{\partial p}{\partial \rho E} n_{out,y} \\ \frac{\partial p}{\partial \rho} n_{out,z} & \frac{\partial p}{\partial \rho u} n_{out,z} & \frac{\partial p}{\partial \rho v} n_{out,z} & \frac{\partial p}{\partial \rho w} n_{out,z} & \frac{\partial p}{\partial \rho E} n_{out,z} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (C.3)$$

where $\vec{n}_{out} = (n_{out,x}, n_{out,y}, n_{out,z})$ is the normal to the boundary face vector. Considering equation (2.10) transformed in the following formulation (C.4)

$$p = (\gamma - 1) \rho E - \frac{(\gamma - 1)}{2\rho} [(\rho u)^2 + (\rho v)^2 + (\rho w)^2] \quad (C.4)$$

the final free-slip diagonal contribution can be calculated as:

$$\frac{\partial \vec{H}_{\text{freeslip}}}{\partial \vec{W}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \frac{(\gamma-1)}{2}(u^2 + v^2 + w^2)n_{out,x} & -(\gamma-1)un_{out,x} & -(\gamma-1)vn_{out,x} & -(\gamma-1)wn_{out,x} & (\gamma-1)n_{out,x} \\ \frac{(\gamma-1)}{2}(u^2 + v^2 + w^2)n_{out,y} & -(\gamma-1)un_{out,y} & -(\gamma-1)vn_{out,y} & -(\gamma-1)wn_{out,y} & (\gamma-1)n_{out,y} \\ \frac{(\gamma-1)}{2}(u^2 + v^2 + w^2)n_{out,z} & -(\gamma-1)un_{out,z} & -(\gamma-1)vn_{out,z} & -(\gamma-1)wn_{out,z} & (\gamma-1)n_{out,z} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{C.5})$$

