

Τελική Τεχνική Έκθεση Έτος 2015



ΘΑΛΗΣ – Πολυτεχνείο Κρήτης

Πλατφόρμα προηγμένων μαθηματικών μεθόδων και λογισμικού για την επίλυση προβλημάτων πολλαπλών πεδίων (multi physics, multidomain) σε σύγχρονες υπολογιστικές αρχιτεκτονικές: Εφαρμογή σε προβλήματα Περιβαλλοντικής Μηχανικής και Ιατρικής (MATENVMED- MIS 379416)

Δράση 3.1

Υλοποίηση σε Clusters, Grids και Cloud



Πίνακας Περιεχομένων

1. ΣΚΟΠΟΣ	4
1.1. ΔΡΑΣΗ 3.1: ΥΛΟΠΟΙΗΣΗ ΣΕ CLUSTERS, GRIDS ΚΑΙ CLOUD	4
2. ΜΕΘΟΔΟΛΟΓΙΑ	5
2.1. ΈΡΕΥΝΑ ΠΕΔΙΟΥ ΣΕ ΠΑΡΑΛΛΗΛΕΣ ΑΡΧΙΤΕΚΤΟΝΙΚΕΣ	5
2.1.1. Εισαγωγή	5
2.2. ΔΟΚΙΜΑΣΤΙΚΗ ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΛΕΙΤΟΥΡΓΙΑ ΠΛΑΤΦΟΡΜΑΣ FENICS	7
2.3. ΜΕΘΟΔΟΙ ΧΑΛΑΡΩΣΗΣ ΣΤΗ ΔΙΕΠΑΦΗ	7
2.3.1. Εισαγωγή	7
2.3.2. Διακριτοποίηση πεδίου με επαναληπτική χαλάρωση στις διεπαφές	8
2.4. ΕΦΑΡΜΟΓΗ ΕΝΟΣ ΥΒΡΙΔΙΚΟΥ ΠΑΡΑΛΛΗΛΟΥ MONTE CARLO ΕΠΙΛΥΤΗ ΜΔΕ ΣΕ ΟΡΘΟΓΩΝΙΑ ΠΟΛΛΑΠΛΑ ΠΕΔΙΑ	10
2.4.1. Εισαγωγή	10
2.4.2. Ανάλυση βασικού αλγορίθμου	11
2.5. ΥΛΟΠΟΙΗΣΗ ΜΕΘΟΔΩΝ IR ΣΤΗ FENICS	13
2.5.1. Εισαγωγή	13
2.5.2. FEniCS	14
2.5.3. RabbitMQ	15
2.5.4. Σειριακή Υλοποίηση	15
2.5.5. Παράλληλη Υλοποίηση	16
2.6. ΑΣΥΓΧΡΟΝΗ ΜΕΘΟΔΟΣ ΧΑΛΑΡΩΣΗΣ ΔΙΕΠΑΦΩΝ	16
2.6.1. Εισαγωγή	16
2.6.2. Σειριακός αλγόριθμος	16
2.6.3. Σύγχρονος παράλληλος αλγόριθμος	17
2.6.4. Ασύγχρονος παράλληλος αλγόριθμος	17
2.7. ΠΑΡΑΛΛΗΛΟΠΟΙΗΣΗ ALOPEX	17
3. ΑΠΟΤΕΛΕΣΜΑΤΑ ΠΑΡΑΛΛΗΛΗΣ ΥΛΟΠΟΙΗΣΗΣ ΑΛΓΟΡΙΘΜΩΝ	18
3.1. ΜΕΘΟΔΟΙ ΧΑΛΑΡΩΣΗΣ ΣΤΗ ΔΙΕΠΑΦΗ	18
3.1.1. Πειράματα	19
3.2. ΕΦΑΡΜΟΓΗ ΥΒΡΙΔΙΚΟΥ ΠΑΡΑΛΛΗΛΟΥ MONTE CARLO ΕΠΙΛΥΤΗ Μ.Δ.Ε.	22
3.2.1. Πειραματικά αποτελέσματα	22
3.2.2. Συμπεράσματα	25
3.3. ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΣΥΓΧΡΟΝΗΣ ΥΛΟΠΟΙΗΣΗΣ IR ΜΕΘΟΔΩΝ	26
4. ΠΑΡΑΔΟΤΕΑ	28
Α. ΥΛΟΠΟΙΗΣΗ ΣΕ CLUSTERS, GRIDS ΚΑΙ CLOUD	29
5. ΣΥΝΕΡΓΑΣΙΕΣ	30
6. ΜΕΛΛΟΝΤΙΚΕΣ ΔΡΑΣΕΙΣ	30
7. ΒΙΒΛΙΟΓΡΑΦΙΑ	30

1. Σκοπός

Η Δράση 3 αναφέρεται γενικά στη χρήση σύγχρονων υπολογιστικών αρχιτεκτονικών αφενός μεν στην υλοποίηση των αριθμητικών μεθόδων και του αντίστοιχου επιστημονικού λογισμικού, που θα αναπτυχθούν στα πλαίσια της Δράσης 2 (ιδιαίτερα 2.2 και 2.3), αφετέρου δε στην ανάπτυξη της πλατφόρμας λογισμικού της Δράσης 4. Η υλοποίησή του πραγματοποιείται από δύο διαφορετικές δράσεις (3.1 και 3.2), ανάλογα με την κατηγορία αρχιτεκτονικών που χρησιμοποιούνται. Η ΚΕΟ3 έχει αναλάβει την δράση 3.1: Υλοποίηση σε Clusters, Grids και Cloud και η ΚΕΟ2 την δράση 3.2: Υλοποίηση σε FPGAs.

1.1. Δράση 3.1: Υλοποίηση σε Clusters, Grids και Cloud

Αντικείμενο της συγκεκριμένης δράσης αποτελεί η αποδοτική υλοποίηση των αριθμητικών μεθόδων για ασυνεχή προβλήματα πολλαπλών πεδίων σε σύγχρονες παράλληλες αρχιτεκτονικές.

Αρχικά η υλοποίηση έλαβε χώρα σε επίπεδο συστοιχιών υπολογιστών (Clusters). Η υλοποίηση των μεθόδων αφορά τόσο σε ομοιογενή, συμμετρικά σχήματα όπου γίνεται χρήση ενός συγκεκριμένου προγραμματιστικού μοντέλου ανταλλαγής μηνυμάτων (MPI), όσο και σε ετερογενή, μη συμμετρικά σχήματα, όπου ο παραλληλισμός των μεθόδων επιτελείται σε πολλαπλά επίπεδα. Σύμφωνα με αυτά τα συνεργατικά σχήματα εφαρμόστηκε συνδυασμός μοντέλων ανταλλαγής μηνυμάτων (MPI) με μοντέλα προγραμματισμού κοινής μνήμης (OpenMP, Pthreads) για την περισσότερο ευέλικτη αξιοποίηση των επεξεργαστών πολλαπλών πυρήνων που περιέχονται σε κάθε κόμβο του Cluster.

Κατά το επόμενο στάδιο της συγκεκριμένης δράσης η υλοποίηση έγινε σε περιβάλλον Cloud μέσω χρήσης υπηρεσιών διαδικτύου (web services). Στη περίπτωση αυτή το παρεχόμενο λογισμικό είναι σε θέση να αξιοποιεί με καλύτερο τρόπο τους διαθέσιμους υπολογιστικούς πόρους, ιδιαίτερα κατά την περίπτωση όπου υποβάλλονται μέσω του Cloud αιτήσεις για παράλληλη εκτέλεση από περισσότερους του ενός χρήστες. Για το λόγο αυτό στη δράση 4.1 υλοποιήθηκε το μοντέλο Λογισμικό σαν Υπηρεσία (SaaS – Software as a Service), που δίνει τη δυνατότητα ενός φιλικού προς το χρήστη διαδικτυακού γραφικού περιβάλλοντος που μπορεί να χρησιμοποιηθεί όχι μόνο από σταθμούς εργασίας αλλά και από φορητές συσκευές. Επιπλέον, για να ληφθεί υπόψη η ποικιλομορφία των υποπροβλημάτων για ένα δοθέν πρόβλημα πολλαπλών πεδίων / πολλαπλής φυσικής, η προτεινόμενη εφαρμογή έχει τη δυνατότητα να υπολογίσει και να παρέχει τα βέλτιστα VMs (σε πλήθος και δυνατότητες) σύμφωνα με τους διαθέσιμους πόρους, το μέγεθος του συνολικού προβλήματος και το μέγεθος των υποπροβλημάτων του.

Σημείωση: Στο Τεχνικό Δελτίο Έργου στη συγκεκριμένη Δράση προβλεπόταν η υλοποίηση σε Grids ή εναλλακτικά σε περιβάλλον Cloud, εφόσον αυτό είχε ωριμάσει. Επειδή κατά την περίοδο υλοποίησης του έργου η Cloud τεχνολογία είχε ήδη εφαρμοστεί με επιτυχία, αποφασίστηκε η υλοποίηση σε αυτό το υπολογιστικό περιβάλλον.

2. Μεθοδολογία

2.1. Έρευνα Πεδίου σε Παράλληλες Αρχιτεκτονικές

2.1.1. Εισαγωγή

Με την έναρξη του έργου, έγινε ανασκόπηση της διεθνούς βιβλιογραφίας και μελέτη των μεθόδων επίλυσης προβλημάτων πολλαπλών πεδίων /πολλαπλής φυσικής (Multi Domain / Multi Physics) προκειμένου τα μέλη της ΚΕΟ3 να αποκτήσουν το απαραίτητο θεωρητικό υπόβαθρο κατανόησης των αριθμητικών μεθόδων επίλυσης, οι οποίες θα υλοποιηθούν στα παραπάνω περιβάλλοντα. Επίσης, μελετήθηκαν οι μέθοδοι Collocation και Μετασχηματισμού Φωκά.

Ακολούθησε η ανασκόπηση λογισμικού ανοικτού (ελεύθερου) κώδικα (Open Source) αφενός για γλώσσες προγραμματισμού για παράλληλη υλοποίηση και αφετέρου για πλατφόρμες λογισμικού, που διαθέτουν υλοποιημένα εργαλεία για επίλυση PDEs και άλλες αριθμητικές μεθόδους.

Η ανασκόπηση της βιβλιογραφίας έδειξε ότι έχουν υλοποιηθεί δύο βασικές εκδόσεις στην παράλληλη εκτέλεση προγραμμάτων: Fork-Join Style (που επιτυγχάνει και Nested Parallelism) και SPMD style (Single program, multiple data).

Τα βασικά μοντέλα παράλληλου προγραμματισμού, για τα οποία υπάρχουν διαθέσιμες υλοποιήσεις ελεύθερου λογισμικού, είναι τα ακόλουθα [1]: Parallel Random Access Machine, Unrestricted Message Passing, Asynchronous Shared Memory and Partitioned Global Address Space, Data Parallel Models, Task Parallel Models and Task Graphs.

Τέλος έχει αναπτυχθεί και υλοποιηθεί, μια πληθώρα από γλώσσες προγραμματισμού, για αρχιτεκτονικές παράλληλης επεξεργασίας, όπως η C*, Occam, PVM, HPF, Linda, αλλά και πολλές ερευνητικές προτάσεις όπως Orca, PCN, CC++, P3L, BSP language. Μεταξύ των γλωσσών ελεύθερου λογισμικού, είναι η OpenCL, η οποία έχει καθιερωθεί σαν το Open Standard για παράλληλο υπολογισμό σε ετερογενή συστήματα. Επίσης η OpenCL μπορεί να χρησιμοποιηθεί σε μια μεγάλη ποικιλία από πολύ-πύρνα συστήματα και έχει τη δυνατότητα κλιμάκωσης (scaling). Μια νέα γλώσσα παράλληλου προγραμματισμού είναι η ParaSail, που μοιάζει με μια τροποποίηση της Java ή της C#. Η διαφορά είναι ότι διαμερίζει αυτόματα ένα πρόγραμμα σε χιλιάδες άλλες εργασίες, οι οποίες μπορούν τότε να διαχυθούν σε διάφορους πυρήνες, ένα τέχνασμα που λέγεται pico-threading, το οποίο μεγιστοποιεί τον αριθμό των εργασιών που μπορούν να εκτελεστούν παράλληλα, ανεξάρτητα από τον αριθμό των πυρήνων. Η ParaSail κάνει επίσης αυτόματη αποσφαλμάτωση (debugging), που κάνει τον κώδικα πιο ασφαλή. Παρόλα αυτά είναι σχετικά νέα έκδοση και στην ανασκόπηση εντοπίστηκε ότι έχει ακόμη κάποια bugs.

Συνοψίζοντας, το μέλλον του υπολογισμού σε παράλληλα περιβάλλοντα υπαγορεύεται από τις φυσικές και τεχνικές ανάγκες. Οι παράλληλοι υπολογιστές γίνονται περισσότερο υβριδικόι, συνδυάζοντας πολυνηματικό υλικό, πολλαπλούς πυρήνες, μονάδες SIMD, επιταχυντές και on-chip συστήματα επικοινωνίας.

Η τελική επιλογή του μοντέλου θα γίνει σε συνδυασμό με την πολυπλοκότητα των λύσεων του προβλήματος πολλαπλών πεδίων, αλλά και τη μεταφερσιμότητα σε περιβάλλοντα Clusters, GRIDS και Cloud Computing.

Λαμβάνοντας υπ' όψιν τις παραδοσιακές προσεγγίσεις από τους χώρους της παράλληλης επεξεργασίας και των παράλληλων αρχιτεκτονικών μπορούμε να εντοπίσουμε αρκετά μειονεκτήματα των clusters [2]. Από την εμπειρία μας, έχουμε διαπιστώσει πόσο κρίσιμοι παράγοντες είναι η καθυστέρηση και το εύρος ζώνης του διασυνδεδετικού δικτύου, πόσο χρήσιμη είναι η κοινόχρηστη μνήμη και πόσο αναγκαίο το ελαφρύ λογισμικό ελέγχου. Σε αυτά τα σημεία, τα clusters υστερούν συγκριτικά με τα συμβατικά παράλληλα συστήματα, όπου το διασυνδεδετικό δίκτυο είναι υψηλότερων επιδόσεων, η κοινή μνήμη, ο καθολικός συγχρονισμός και η διατήρηση της συνοχής της κρυφής μνήμης υλοποιούνται με υλικό, υπερτερώντας σε σχέση με την αντίστοιχη υλοποίηση με λογισμικό, ενώ τέλος το ειδικού τύπου λογισμικό που μπορεί να ελέγχει κάθε κόμβο καταναλώνει λιγότερη μνήμη και αντιδρά πιο γρήγορα από το πλήρες και αυτόνομο λειτουργικό σύστημα που ελέγχει κάθε κόμβο του cluster. Λόγω των παραπάνω μειονεκτημάτων, τα clusters αποδεικνύονται ακατάλληλα για ορισμένες εφαρμογές. Σε πολλές ωστόσο περιπτώσεις, χρησιμοποιώντας νέες αλγοριθμικές τεχνικές που δεν εξαρτώνται τόσο από την γρήγορη επικοινωνία των κόμβων, τα clusters μπορούν να ανταγωνιστούν σε επιδόσεις παράλληλα συστήματα πολύ υψηλότερου κόστους.

Το σημαντικότερο πλεονέκτημα των clusters είναι ότι χρησιμοποιούνται τεχνολογίες υλικού και λογισμικού ευρείας χρήσης. Τόσο τα δίκτυα σταθμών εργασίας, όσο και οι συστάδες προσωπικών υπολογιστών της τάξης Beowulf άνθισαν διότι δεν απαιτούσαν δαπανηρά και μακροπρόθεσμα σχέδια μέχρι να αρχίσουν να αποδίδουν. Το χαρακτηριστικό αυτό οδηγεί προφανώς και σε ένα σαφές πλεονέκτημα κόστους. Συνεπώς ο λόγος κόστους προς απόδοση των clusters μπορεί να υπερτερεί του αντίστοιχου των παραδοσιακών υπερυπολογιστών για εφαρμογές που δεν απαιτούν επικοινωνία υψηλού εύρους ζώνης και χαμηλής καθυστέρησης.

Ένα άλλο σημείο υπεροχής των εμπορικών συστάδων υπολογιστών σε σχέση με τις κλασικές παράλληλες μηχανές, είναι η ευελιξία και η παραμετροποισιμότητά τους. Ο αριθμός των κόμβων, η χωρητικότητα μνήμης ανά κόμβο, ο αριθμός των επεξεργαστών ανά κόμβο και η τοπολογία του διασυνδεδετικού δικτύου είναι δομικές παράμετροι που μπορούν να καθοριστούν με μεγάλη λεπτομέρεια και για κάθε cluster ξεχωριστά. Επιπλέον, η δομή του συστήματος μπορεί να αλλάζει ή να επεκτείνεται αξιοποιώντας την ήδη υπάρχουσα υποδομή. Ο εκτεταμένος αυτός έλεγχος στη δομή του συστήματος ωφελεί τόσο τους χρήστες, που μπορούν να έχουν κάθε στιγμή αυτό που επιθυμούν, όσο και τους κατασκευαστές, που μπορούν να καλύπτουν ένα ευρύ φάσμα απαιτήσεων επιδόσεων ή κόστους.

Τέλος, οι εμπορικές συστάδες υπολογιστών μπορούν να ενσωματώνουν γρήγορα τεχνολογικές εξελίξεις που χρησιμοποιούνται μαζικά. Νέοι τύποι επεξεργαστών, μνημών, δίσκων και δικτύων κατασκευάζονται διαρκώς για τους προσωπικούς υπολογιστές, επιτρέποντας και στα clusters να επωφελούνται από τις προόδους αυτές. Τα clusters επιπλέον επωφελούνται από πτώσεις τιμών λόγω της μαζικής παραγωγής των εξαρτημάτων προσωπικών υπολογιστών. Συνολικά θα λέγαμε ότι αν και τα clusters υπολογιστών δεν επιτυγχάνουν πάντα επιδόσεις αντάξιες κορυφαίων παράλληλων μηχανών, το χαμηλό τους κόστος και η ευελιξία που προσφέρουν τα καθιστά σαφώς ανταγωνιστικά.

Ένας blade server [3],[4] είναι ένας απογυμνωμένος server με τμηματική σχεδίαση βελτιστοποιημένος για να ελαχιστοποιήσει τη χρήση φυσικού χώρου και ενέργειας. Ενώ ένας τυπικός rack-mount server μπορεί να λειτουργήσει τουλάχιστον με ένα καλώδιο ρεύματος και ένα καλώδιο ισχύος, στους blade servers έχουν αφαιρεθεί πολλά συστατικά για να εξοικονομήσουν χώρο, να ελαχιστοποιήσουν την κατανάλωση ισχύος, αλλά εξακολουθούν να έχουν όλα εκείνα τα λειτουργικά

συστατικά για να θεωρούνται υπολογιστές. Ένα blade enclosure που μπορεί να στηρίξει πολλούς blade servers παρέχει υπηρεσίες, όπως ρεύμα, ψύξη, δικτύωση και διαχείριση.

2.2. Δοκιμαστική εγκατάσταση και λειτουργία πλατφόρμας FEniCS

Το project FEniCS [5] είναι μια συλλογή από δωρεάν, ανοιχτού κώδικα, στοιχεία λογισμικού με κοινό στόχο να καταστεί δυνατή αυτοματοποιημένη επίλυση των διαφορικών εξισώσεων. Τα στοιχεία που το απαρτίζουν παρέχουν εργαλεία επιστημονικού υπολογισμού για εργασία με υπολογιστικά πλέγματα, πεπερασμένων στοιχείων μεταβολικές διατυπώσεις για διαφορικές εξισώσεις και μερικές διαφορικές εξισώσεις καθώς επίσης για αριθμητική γραμμική άλγεβρα. Η τρέχουσα σταθερή έκδοση του FEniCS Project είναι η 1.1 η οποία και κυκλοφόρησε τον 7 Ιανουαρίου του 2013. Για τις ανάγκες του έργου, η παραπάνω έκδοση εγκαταστάθηκε σε server του Εργαστηρίου Αναγνώρισης Προτύπων, όπου έγινε η δοκιμαστική λειτουργία της, με τα έτοιμα demo/παραδείγματα που περιέχει.

2.3. Μέθοδοι Χαλάρωσης στη Διεπαφή

2.3.1. Εισαγωγή

Οι διάφορες μέθοδοι διακριτοποίησης πεδίου που έχουν πρόσφατα αναπτυχθεί για την αποδοτική επίλυση ελλειπτικών διαφορικών εξισώσεων μπορούν εύκολα να ταξινομηθούν σε δύο κατηγορίες: επικαλυπτόμενες και μη επικαλυπτόμενες. Και οι δύο προσεγγίσεις έχουν ήδη χρησιμοποιηθεί για να μοντελοποιήσουν αποτελεσματικά μεγάλης κλίμακας, βιομηχανικά, προβληματικών συνθηκών προβλήματα. Παρόλα αυτά θεωρείται ότι απαιτείται επιπλέον θεωρητική και πειραματική ανάλυση ώστε να γίνουν πρακτικές αυτές οι μέθοδοι και χρήσιμα εργαλεία για τους μη ειδήμονες.

Τα επικαλυπτόμενα (Schwartz) σχήματα έχουν στο παρελθόν λάβει μεγάλη προσοχή. Άρθρα που κάνουν ανασκόπηση και συγκρίνουν διάφορα τέτοια σχήματα [1] και ερευνούν τις σχετιζόμενες στρατηγικές preconditioning [2, 3] υπάρχουν ήδη στη βιβλιογραφία. Σχετικά πρόσφατα ένας αριθμός μελετών έχουν δείξει ότι τα μη επικαλυπτόμενα σχήματα μπορούν να συμπεριφερθούν καλά και μπορούν πιθανά να ελευθερώσουν τους χρήστες από συγκεκριμένες πολυπλοκότητες στη διατύπωση και στην υλοποίησή τους. Η σύγκριση των κυριότερων χαρακτηριστικών αυτών των δύο κατηγοριών μεθόδων και η ύπαρξη σχέσεων ισοδυναμίας ανάμεσά τους έχει ήδη μελετηθεί αρκετά [4, 5, 6].

Οι μέθοδοι χαλάρωσης στη διεπαφή μας πάνε ένα βήμα παραπέρα από τις μη επικαλυπτόμενες μεθόδους διακριτοποίησης πεδίου [7]. Σε μια προσπάθεια να μιμηθούν τη φυσική του προβλήματος στον πραγματικό κόσμο, σπάνε μια πολύπλοκη μερική διαφορική εξίσωση (ΜΔΕ) που δρα σε ένα μεγάλο ή/και πολύπλοκο πεδίο σε ένα σύνολο προβλημάτων ΜΔΕ με διαφορετικούς αλλά απλούς τελεστές που δρουν σε διαφορετικά μικρότερα και εύκολα υποπεδία. Το πολλαπλών πεδίων, πολλαπλών ΜΔΕ σύστημα είναι σωστά συζευγμένο χρησιμοποιώντας τελεστές εξομάλυνσης στα όρια μεταξύ των υποπεδίων.

Από την οπτική της χαλάρωσης στη διεπαφή αυτές οι μέθοδοι αποτελούνται από τη διαμέριση του πεδίου σε ένα σύνολο από μη επικαλυπτόμενα υποπεδία και

από την επιβολή κάποιων οριακών συνθηκών στα όρια των διεπαφών που ορίζονται από αυτήν τη διαμέριση. Στη συνέχεια χρησιμοποιώντας αρχικές μαντεψιές στις διεπαφές επιλύεται το σύνολο των συνεπαγόμενων προβλημάτων ΜΔΕ. Οι λύσεις που προκύπτουν δεν ικανοποιούν τις οριακές συνθήκες των διεπαφών και εφαρμόζεται χαλάρωση στις διεπαφές για να αποκτήσουμε νέες τιμές στα όρια των διεπαφών, που να ικανοποιούν καλύτερα τις συνθήκες και λύνουμε τις ΜΔΕ με αυτές τις νέες τιμές. Τα παραπάνω βήματα επαναλαμβάνονται μέχρι να επιτευχθεί σύγκλιση.

Οι δύο μέθοδοι χαλάρωσης στη διεπαφή που επελέγησαν ως πλέον κατάλληλες για τις ανάγκες του έργου είναι οι δύο παρακάτω:

- **GEO** Μια μέθοδος που βασίζεται σε μια απλή γεωμετρική συστολή.
- **ROB** Ένας αλγόριθμος που χρησιμοποιεί τις Robin συνθήκες στις διεπαφές για εξομάλυνση.

2.3.2. Διακριτοποίηση πεδίου με επαναληπτική χαλάρωση στις διεπαφές

Επί του παρόντος, ο χώρος της διακριτοποίησης πεδίου αποτελείται από δύο μέρη - τις επικαλυπτόμενες και τις μη επικαλυπτόμενες μεθόδους. Οι επικαλυπτόμενες – επίσης γνωστές και ως Schwartz – ήταν οι πρώτες που χρησιμοποιήθηκαν και έχουν αποδείξει ήδη ότι είναι πολύ αποδοτικές αριθμητικές διαδικασίες που χαίρουν συγκεκριμένων πολύ επιθυμητών ιδιοτήτων σύγκλισης. Παρ' όλα αυτά, έχει επίσης παρατηρηθεί ότι έχουν διάφορα πολύ σημαντικά μειονεκτήματα που απαγορεύουν τη χρήση τους για συγκεκριμένες εφαρμογές. Για παράδειγμα, σχεδόν όλες εκ των πολλών προτεινόμενων μεθόδων διακριτοποίησης πεδίου για την επίλυση μοντέλων διάδοσης κυμάτων είναι είτε μη επικαλυπτόμενες, είτε χαλάρωσης στη διεπαφή [8, 9, 10, 11].

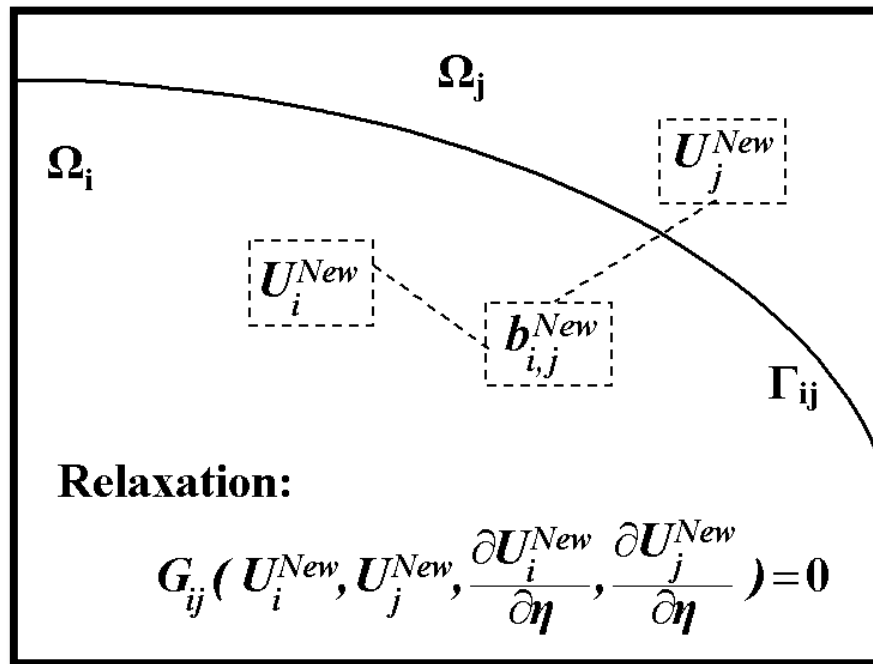
Οι μη επικαλυπτόμενες μέθοδοι παρουσιάζουν συγκεκριμένα πλεονεκτήματα σε σχέση με τις επικαλυπτόμενες. Πιο συγκεκριμένα:

- Δεν είναι ευαίσθητες σε άλματα στους συντελεστές του τελεστή. Η συμπεριφορά σύγκλισής τους και οι υπολογισμοί θεωρητικού σφάλματός τους παραμένουν ίδια ακόμα και αν ο διαφορικός τελεστής περιλαμβάνει ασυνεχείς συντελεστές, αρκεί τα άλματα να συμβαίνουν κατά μήκος των γραμμών των διεπαφών [12].
- Έχουν μικρότερο κόστος (επιβάρυνση) επικοινωνίας σε μια παράλληλη υλοποίηση σε κατανεμημένης μνήμης πολυεπεξεργαστικά συστήματα. Το κόστος της επικοινωνίας είναι αναλογικό με το μήκος των γραμμών των διεπαφών, τη στιγμή που στις επικαλυπτόμενες μεθόδους είναι αναλογικό με την επικαλυπτόμενη περιοχή [13].
- Η καταγραφή/συντήρηση είναι σχετικά ευκολότερη για τη διαμέριση και το χειρισμό των σχετικών δομών δεδομένων απ' ότι στις πολύπλοκες και ακριβές επικαλυπτόμενες μεθόδους [13].

Υπάρχουν δύο κύριες οπτικές για τις μη επικαλυπτόμενες μεθόδους, το preconditioning και η χαλάρωση στη διεπαφή. Μια εις βάθος έρευνα για τις μη επικαλυπτόμενες μεθόδους διακριτοποίησης πεδίου αναλυμένες από την οπτική του preconditioning μπορεί να βρεθεί στο [14] και μια γενική διατύπωση και ανάλυση

των μεθόδων χαλάρωσης στη διεπαφή στο [15]. Για την αναγνώριση των κύριων χαρακτηριστικών τους, ακολουθεί μια σύντομη περιγραφή των μεθόδων χαλάρωσης στη διεπαφή.

Η χαλάρωση στη διεπαφή είναι ένα βήμα πέρα από τις μη επικαλυπτόμενες μεθόδους (ακολουθώντας τη χαλάρωση του Southwell του 1930), αλλά στη ΜΔΕ αντί για το επίπεδο της γραμμικής άλγεβρας, για να διατυπώσει τη χαλάρωση σαν επαναλαμβανόμενες διαδικασίες εξομάλυνσης της διεπαφής. Ένα σύνθετο φυσικό φαινόμενο αποτελείται από μια συλλογή απλών μερών με καθένα να υπακούει έναν μεμονωμένο φυσικό νόμο τοπικά και να μοιράζεται τις συνθήκες των διεπαφών του με γείτονες. Η χαλάρωση στη διεπαφή χωρίζει το πεδίο σε ένα σύνολο μη επικαλυπτόμενων πεδίων και επιβάλλει κάποιες οριακές συνθήκες στη διεπαφή ανάμεσα στις γραμμές των υποπεδίων. Δοθείσης μιας αρχικής μαντεψιάς, μιμείται τη φυσική του πραγματικού κόσμου επιλύοντας τα τοπικά προβλήματα ακριβώς σε κάθε υποπεδίο και χαλαρώνοντας τις οριακές τιμές για να πάρει καλύτερους υπολογισμούς των σωστών συνθηκών στη διεπαφή. Αυτή η διαδικασία παρουσιάζεται στην εικόνα 1, όπου ο γενικός τύπος χαλάρωσης G_{ij} (που βασίζεται στις τρέχουσες τοπικές λύσεις U_i^{New} και U_j^{New} των δύο γειτονικών υποπεδίων Ω_i και Ω_j) υπολογίζει διαδοχικές προσεγγίσεις της λύσης $b_{i,j}^{New}$ στη διεπαφή Γ_{ij} ανάμεσά τους.



Εικόνα 1: Ο μηχανισμός χαλάρωσης στη διεπαφή

Για την επίσημη περιγραφή της μεθόδου θεωρούμε το ακόλουθο πρόβλημα:

$$Du = f \text{ in } \Omega, \quad Bu = c \text{ on } \partial\Omega \quad (1)$$

όπου το D είναι ένας ελλειπτικός, μη γραμμικός γενικά, διαφορικός τελεστής και το B είναι ένας τελεστής συνθήκης που ορίζεται στο όριο $\partial\Omega$ ενός ανοιχτού πεδίου $\Omega \in R^d, d = 1, 2, \dots$. Αυτό το πεδίο διαμερίζεται σε p ανοιχτά υποπεδία $\Omega_i, i = 1, \dots, p$ τέτοια ώστε $\Omega = \bigcup_{i=1}^p \bar{\Omega}_i \setminus \partial\Omega$ και $\bigcap_{i=1}^p \Omega_i = \emptyset$. Για λόγους που σχετίζονται είτε με τα φυσικά χαρακτηριστικά του προβλήματος, είτε με τους

διαθέσιμους υπολογιστικούς πόρους, κάποιος θα ήθελε να αντικαταστήσει την (1) με το ακόλουθο σύστημα χαλαρά συνδεδεμένων διαφορικών προβλημάτων:

$$\begin{aligned} D_i u &= f_i \text{ in } \Omega_i, \\ G_{i,j} u &= 0 \text{ on } \partial\Omega_i \cap \partial\Omega_j \setminus \partial\Omega \quad \forall j \neq i, \quad B_i u = c_i \text{ on } \partial\Omega_i \cap \partial\Omega \end{aligned} \quad (2)$$

όπου $i = 1, \dots, p$. Αυτά τα διαφορικά προβλήματα συνδέονται μέσω των συνθηκών των διεπαφών $G_{i,j} u = 0$ και περιλαμβάνουν τους περιορισμούς D_i και B_i των ολικών διαφορικού και συνθήκης τελεστών D και B , αντίστοιχα σε κάθε υποπεδίο, με κάποιους από αυτούς γραμμικούς και άλλους μη γραμμικούς. Οι συναρτήσεις f_i και c_i είναι ανάλογοι περιορισμοί των συναρτήσεων f και c . Ο τοπικός τελεστής διεπαφής $G_{i,j}$ σχετίζεται με τη μέθοδο χαλάρωσης στη διεπαφή και διαφορετικές επιλογές για τα $G_{i,j}$ οδηγούν σε διαφορετικά σχήματα χαλάρωσης. Στην παρούσα μελέτη θα αναφερθούμε σε διάφορες μεθόδους χαλάρωσης στη διεπαφή που έχουν τα ακόλουθα χαρακτηριστικά:

- Αποσυνθέτουν αρχικά το πρόβλημα (1) στο διαφορικό επίπεδο και στη συνέχεια διακριτοποιούν τα προκύπτοντα διαφορικά υποπροβλήματα (2).
- Έχουν την ευελιξία να χρησιμοποιούν το καταλληλότερο σχήμα διακριτοποίησης για κάθε υποπρόβλημα.
- Δεν επικαλύπτουν τα υποπεδία Ω_i .
- Με χρήση καλών παραμέτρων χαλάρωσης στο $G_{i,j}$ είναι αρκετά γρήγορες, ώστε να μη χρειάζεται preconditioning.
- Απλοποιούν τη γεωμετρία και τη φυσική του υπολογισμού λαμβάνοντας υπόψη τα υποπροβλήματα (2) και όχι το συνολικό διαφορικό πρόβλημα (1).
- Μπορούν να χρησιμοποιούν τμήματα λογισμικού ξαναχρησιμοποιώντας τμήματα λογισμικού επιλυτών για την επίλυση των μεμονωμένων υποπροβλημάτων (2).
- Είναι γενικοί και ισχυροί.

Υπάρχουν διάφορα ερωτήματα με πρόκληση που αφορούν πρακτικές εφαρμογές τέτοιων μεθόδων, όπως το να βρίσκεις τον καταλληλότερο χαλαρωτή για ένα συγκεκριμένο πρόβλημα, να καθορίζεις το πεδίο εφαρμογής καθενός, να εξηγείς την αλληλεπίδραση ανάμεσα στη μαθηματική επανάληψη και τη μέθοδο αριθμητικής επίλυσης, να επιλέγεις καλές ή βέλτιστες τιμές για τις παραμέτρους χαλάρωσης, κλπ. Αξίζει να σημειωθεί ότι εφόσον όλες οι μέθοδοι αποσυνθέτουν και χαλαρώνουν τις τιμές των διεπαφών στο συνεχές επίπεδο, η ανάλυση σύγκλισης αυτών των μεθόδων χρειάζεται να γίνει επίπεδο των ΜΔΕ (συνεχές) και άρα είναι ένα πρόβλημα μαθηματικής ανάλυσης και όχι αριθμητικής. Μια λεπτομερής περιγραφή και σύγκριση των δύο μεθοδολογιών χαλάρωσης δίνεται στην αναφορά [16].

2.4 Εφαρμογή ενός υβριδικού παράλληλου Monte Carlo επιλυτή ΜΔΕ σε ορθογώνια πολλαπλά πεδία

2.4.1. Εισαγωγή

Η μέθοδος Monte Carlo είναι γνωστή εδώ και πολλά χρόνια, αλλά μόνο μετά την εμφάνιση των πρώτων ηλεκτρονικών υπολογιστών, η μέθοδος έχει αποκτήσει ένα αυξανόμενο ενδιαφέρον ως ένα εργαλείο για την επίλυση σημαντικών φυσικών προβλημάτων. Ο αλγόριθμος Monte Carlo παράγει ένα τυχαίο περίπατο,

χρησιμοποιώντας μια προτεινόμενη πυκνότητα και περιλαμβάνει μια μέθοδο για την απόρριψη/αποδοχή των προτεινόμενων κινήσεων. Σήμερα χρησιμοποιείται σχεδόν σε κάθε πτυχή της επιστημονικής έρευνας. Οι Monte Carlo μέθοδοι προσέλκυαν πάντα τους ανθρώπους που εργάζονται στον τομέα της αριθμητικής επίλυσης ΜΔΕ. Αυτή η έλξη έχει εξελιχθεί με το πέρασμα των χρόνων σε συνδυασμό με τις ραγδαίες αλλαγές στην τεχνολογία, όπως ο παράλληλος υπολογισμός, τα clusters υπολογιστών κ.λπ. [1] [2] [3].

2.4.2. Ανάλυση βασικού αλγορίθμου

2.4.2.1. Περιγραφή βασικού αλγορίθμου

Το αρχικό βήμα προς την κατεύθυνση της αξιοποίησης της απόδοσης ενός cluster πολλαπλών πυρήνων ήταν η εκτέλεση ενός profiling της ακολουθιακής υλοποίησης για τον εντοπισμό hotspots στον κώδικα που θα μπορούσαν να οδηγήσουν σε μείωση του χρόνου εκτέλεσης, αν γίνει παράλληλη υλοποίηση αποτελεσματικά. Η ανάλυση απόδοσης βασίστηκε στην υλοποίηση των αλγορίθμων που προτείνονται στο Vavalis et al [4]. Η εν λόγω υλοποίηση είναι σε θέση να λύσει την εξίσωση Poisson με Dirichlet συνοριακές συνθήκες για 2D και 3D υπέρ-ορθογώνια πεδία, αλλά μπορεί εύκολα να επεκταθεί για να χρησιμοποιεί άλλες μεθόδους και πεδία. Ο αλγόριθμος είναι ένας υβριδικός αλγόριθμος με την έννοια ότι τόσο ντετερμινιστικές όσο και μη ντετερμινιστικές μέθοδοι μπορούν να χρησιμοποιηθούν για την επίλυση των πεδίων. Αρχικά, εφαρμόζεται μια πιθανοτική αποσύνθεση πεδίου για τον υπολογισμό εκτιμήσεων των τοπικών λύσεων κατά μήκος των ορίων του υποπεδίου με τη μέθοδο Monte Carlo [5] και στη συνέχεια εφαρμόζεται μια παρεμβολή στις εκτιμήσεις των τοπικών λύσεων [6]. Ειδικότερα, για διδιάστατα πεδία χρησιμοποιείται η Burkardt's splines βιβλιοθήκη [7], ενώ η Multilevel B-spline (MBA) [8] βιβλιοθήκη χρησιμοποιείται ώστε τα τρισδιάστατα πεδία να σχηματίζουν τα όρια των νέων υποπεδίων. Τέλος, καθώς τα όρια των υποπεδίων είναι γνωστά ένας ντετερμινιστικός Laplace επιλυτής που χρησιμοποιεί τη Deal.II [9] χρησιμοποιείται για να βρει τη λύση του προβλήματος για κάθε υποπεδίο. Η βασική υλοποίηση μπορεί να περιγραφεί με τον ακόλουθο ψευδοκώδικα.

```

Προσδιόρισε τις συντεταγμένες των κόμβων όλων των
υποπεδίων
  Για κάθε κόμβο
    Δημιούργησε μια δουλειά
    Ανάθεσε τη δουλειά σε ένα νήμα
      Εκτέλεσε Monte Carlo για να υπολογίσεις τις τιμές
      των κόμβων
    Χρησιμοποίησε μια (2|3) διαστάσεων συνάρτηση παρεμβολής
για να βρεις νέα όρια
  Για κάθε υποπεδίο
    Δημιούργησε μια δουλειά
    Ανάθεσε τη δουλειά σε ένα νήμα
      Λύσε με Laplace επιλυτή
    Πάραξε τα αποτελέσματα

```

Μια ενδιαφέρουσα λεπτομέρεια υλοποίησης είναι ότι παραλληλοποίηση διαμοιραζόμενης μνήμης ήδη υποστηρίζεται χρησιμοποιώντας την Pthreads βιβλιοθήκη.

2.4.2.2. Profiling βασικού κώδικα

Πριν την παραλληλοποίηση κώδικα σε μια αρχιτεκτονική κατανεμημένης μνήμης, ένα βήμα profiling είναι σημαντικό προκειμένου να γίνουν αντιληπτά τα σημεία όπου ο κώδικας σπαταλά χρόνο εκτέλεσης και να καθοριστεί η ακολουθία εκτέλεσης. Με αυτόν τον τρόπο, ο κώδικας μπορεί να υποδιαιρεθεί σε πυρήνες που μπορούν να αναλυθούν για να προσδιορισθεί αν είναι δυνατή η αξιοποίηση των υπολογιστικών πόρων παράλληλα και εάν η παραλληλοποίηση κάποιου πυρήνα θα έχει σημαντική επίπτωση στη συνολική εκτέλεση του προγράμματος. Μια εκτενής ανάλυση απόδοσης έχει πραγματοποιηθεί, χρησιμοποιώντας μια σειρά από διαφορετικές διαμορφώσεις παραμέτρων για το χρησιμοποιηθέν πρόβλημα (Πίνακας 1) και έχουν εντοπιστεί τρεις πυρήνες που θα μπορούσαν ενδεχομένως να βελτιώσουν τη συνολική απόδοση αν υπήρχε μια κατανεμημένη υλοποίηση. Οι τρεις πυρήνες είναι ο Monte Carlo αλγόριθμος, ο αλγόριθμος Παρεμβολής και ο Laplace Επιλυτής. Καθώς για τα περισσότερα προβλήματα εισόδου ο Monte Carlo αλγόριθμος έπαιρνε από 50% έως 94% του χρόνου εκτέλεσης, οι προσπάθειές μας επικεντρώθηκαν σε μια αποτελεσματική παραλληλοποίηση του αλγορίθμου Monte Carlo με χρήση MPI.

Ο Πίνακας 1 παρουσιάζει τα κύρια χαρακτηριστικά των προβλημάτων εισόδου. Το *dim* αντιπροσωπεύει αν το πρόβλημα είναι διδιάστατο ή τρισδιάστατο, το *dlen* το μήκος του πεδίου κατά μήκος κάθε διάστασης, το *subd* τον αριθμό των υποπεδίων κατά μήκος κάθε διάστασης, το *dect* τον τύπο αποσύνθεσης για κάθε διάσταση με *u* να αντιπροσωπεύει την ομοιόμορφη και *n* την ανομοιόμορφη, το *decc* τις συντεταγμένες των διεπαφών που αντιστοιχούν σε κάθε διάσταση κατά μήκος της οποίας το πεδίο αποσυντίθεται ανομοιόμορφα, το *nrpp* τον αριθμό των κόμβων σε μια διεπαφή κατά μήκος μιας διάστασης. Για διδιάστατα προβλήματα παίρνει 2 ορίσματα που αντιστοιχούν στις γραμμές αποσύνθεσης κατά μήκος της διάστασης *Y* και *X* αντίστοιχα, ενώ για τρισδιάστατα παίρνει 6 ορίσματα, με κάθε ζευγάρι να αντιστοιχεί στα επίπεδα αποσύνθεσης *YZ*, *XZ* και *XY* αντίστοιχα. Τέλος, το *btol* αντιπροσωπεύει την ανοχή στο όριο.

Πρόβλημα	Dim	Dlen	Subd	Dect	Decc	nrpp	Btol
1	2D	1. 1.	2 4	n u	.3	12 12	1e-15
2	2D	1. 1.	2 4	n n	.3 .25 .5 .7	12 12	1e-15
3	2D	1. 1.	2 4	n n	.3 .25 .5 .7	12 12	1e-15
4	3D	1. 1. 1.5	3 3 2	u u u	.5 .5 .5 .5	12 12	1e-15
5	3D	1. 1. 1.5	3 3 2	u n u	.5 .55 .5 .5 .5 .5	5 5 5 5 7 7	1e-15
6	3D	1. 1. 1.5	3 3 2	n n n	.5 .4 .5 .55 1.2 5 5 5 5	5 5 5 5 7 7	1e-15
7	3D	1. 1. 1.5	3 3 2	n n n	.5 .4 .5 .55 1.2 5 5 5 5	5 5 5 5 7 7	1e-15

Πίνακας 1: Χαρακτηριστικά προβλημάτων εισόδου

Ο Πίνακας 2 παρουσιάζει το ποσοστό του ακολουθιακού χρόνου εκτέλεσης για διάφορα μεγέθη περιπάτων των προβλημάτων εισόδου για τους προαναφερθέντες πυρήνες, με το MC να σημαίνει Monte Carlo, το I παρεμβολή και το LS επιλυτής Laplace.

Περίπατοι	5K			20K			80K		
Πρόβλημα	C		S	C		S	C		S
1									

	0%		%	9%		%	5%		%
2	0%		0%	6%		%	0%		%
3	4%		%	8%		%	2%		%
4	0%	3%	%	5%	1%	%	8%	%	%
5	0%	0%	0%	3%	%	%	0%	%	%
6	8%	2%	0%	4%	0%	%	4%	%	%
7	4%	0%	%	1%	1%	%	0%	%	%

Πίνακας 2: Ανάλυση απόδοσης

2.4.2.4. Υβριδική προσέγγιση παραλληλοποίησης

Καθώς η βασική υλοποίηση [4] ήδη υποστηρίζει παραλληλοποίηση διαμοιραζόμενης μνήμης του Monte Carlo και του Laplace επιλυτή με χρήση Pthreads, εστιάσαμε στην άμεση υλοποίηση μιας υβριδικής παράλληλης υλοποίησης με χρήση OpenMPI [6] για επικοινωνία μεταξύ διεργασιών. Βασισμένοι στα πειραματικά αποτελέσματα της ανάλυσης απόδοσης, συμπεράναμε ότι:

1. ένα απλό master/worker σχήμα ήταν αρκετό για να καταναίμει τον υπολογιστικό φόρτο του Monte Carlo πυρήνα ανάμεσα στις διεργασίες. Αυτό οφείλεται στο ότι η δουλειά κάθε Monte Carlo νήματος καθορίζεται κυρίως από τον αριθμό των περιπάτων, οπότε μια δίκαιη κατανομή κόμβων μεταξύ των worker διεργασιών και άρα και η τοπική κατανομή κόμβων μεταξύ των νημάτων δεν θα δημιουργήσει κάποια ιδιαίτερη ανισορροπία φόρτου.
2. η master διεργασία μπορούσε αποδοτικά να υλοποιήσει τους πυρήνες της παρεμβολής και του Laplace επιλυτή για κάθε υποπεδίο τοπικά με χρήση threads, αφού είχε πρώτα λάβει την απαιτούμενη εκτίμηση των σχετικών κόμβων από τις άλλες διεργασίες.

Η λογική του βασικού κώδικα [4] μένει σχετικά απaráλλακτη. Προστέθηκε μόνο ένα επίπεδο που καταναίμει τον υπολογιστικό φόρτο και συλλέγει τα αποτελέσματα εκτίμησης από κάθε worker.

2.5 Υλοποίηση Μεθόδων IR στη FEniCS

2.5.1. Εισαγωγή

Η επίλυση μεγάλων και σύνθετων Μερικών Διαφορικών Εξισώσεων (ΜΔΕ) είναι ένα πρόβλημα που αντιμετωπίζεται κυρίως με τεχνικές διακριτοποίησης πεδίου [2, 3]. Αυτή η προσέγγιση περιλαμβάνει την αποσύνθεση στο επίπεδο της γραμμικής άλγεβρας μετά την διακριτοποίηση του πεδίου και της εξίσωσης με την επιθυμητή μέθοδο, δηλαδή: Πεπερασμένων Διαφορών (ΠΔ) ή πεπερασμένων στοιχείων (ΠΣ). Το κύριο χαρακτηριστικό αυτών των μεθόδων είναι η μη ευελιξία για την επιλογή διαφορετικών μεθόδων για κάθε υποπεδίο του αρχικού προβλήματος. Η μεθοδολογία χαλάρωσης στη διεπαφή είναι μια ενδιαφέρουσα εναλλακτική λύση [17], [18]. Εδώ, το πεδίο των ΜΔΕ αποσυντίθεται σε υποπεδία, για λόγους που προέρχονται από την

φυσική ή για λόγους παραλληλισμού, και την ίδια στιγμή αρχικές μαντενιές καθορίζονται στις διεπαφές μεταξύ των υποπεδίων. Τα υποπροβλήματα επιλύονται και νέες τιμές στις διεπαφές υπολογίζονται με συγκεκριμένες μεθόδους χαλάρωσης στη διεπαφή (επιβάλλοντας τις σωστές συνθήκες για το πρόβλημα) επαναληπτικά έως ότου επιτευχθεί η σύγκλιση.

Περιβάλλοντα επίλυσης προβλημάτων πολλαπλών πεδίων / πολλαπλών φυσικών που υλοποιούν τη μεθοδολογία χαλάρωσης στη διεπαφή θα πρέπει να είναι ικανά να φιλοξενούν και να ενσωματώνουν μια ποικιλία υφιστάμενων επιλυτών ΜΔΕ και μεθόδων χαλάρωσης στη διεπαφή. Αυτοί οι επιλυτές πρέπει να παρέχουν μια ελάχιστη λειτουργικότητα συμπεριλαμβανομένων των εξής: ορισμό πεδίου και ΜΔΕ, δημιουργία πλέγματος, σχήμα διακριτοποίησης, εκτίμηση της λύσης και της παραγώγου σε οποιοδήποτε σημείο του πεδίου συμπεριλαμβανομένων των ορίων / διεπαφών. Οι υφιστάμενες υλοποιήσεις, που παρουσιάστηκαν στο κεφάλαιο 2, έχουν διάφορα μειονεκτήματα, μεταξύ των οποίων το ότι εξαρτώνται σε μεγάλο βαθμό από πλατφόρμες πρακτόρων και την βιβλιοθήκη PELLPACK, αποκαλύπτοντας έτσι την ανάγκη μιας νέας υλοποίησης απαλλαγμένης από αυτούς τους περιορισμούς.

Στην παρούσα μελέτη υλοποιούμε μια από τις μεθόδους χαλάρωσης στη διεπαφή, που παρουσιάστηκαν 2^η Ετήσια Έκθεση Προόδου, τη *geometric (GEO) contraction based* στη FEniCS. Η FEniCS [19] είναι μια συλλογή ελεύθερου λογισμικού για αυτοματοποιημένη, αποδοτική επίλυση διαφορικών εξισώσεων. Οι λόγοι για την επιλογή της FEniCS είναι, μεταξύ άλλων, ότι είναι αξιόπιστη, αποδοτική, ελεύθερη και υποσχόμενη να υποστηρίζεται και στο μέλλον. Μια παράλληλη υλοποίηση της GEO στη FEniCS με τη βοήθεια του *RabbiMQ* (προσανατολισμένο σε μηνύματα *middleware*) [20] παρουσιάζεται, επίσης. Στη συνέχεια του κεφαλαίου παρουσιάζονται σύντομα η FEniCS, το *RabbiMQ* και ζητήματα τόσο της σειριακής, όσο και της παράλληλης υλοποίησης.

2.5.2. Υλοποίηση στη FEniCS

Σκοπεύοντας στην ενσωμάτωση της μεθόδου GEO, μαζί με άλλες μεθόδους χαλάρωσης στη διεπαφή, σε ένα ολοκληρωμένο περιβάλλον επίλυσης προβλημάτων που να αφορά προβλήματα πολλαπλών πεδίων / πολλαπλών φυσικών την υλοποιήσαμε στη FEniCS. Η FEniCS παρέχει κλάσεις και μεθόδους με τις οποίες είναι δυνατός ο καθορισμός των ιδιοτήτων των υποπεδίων του προβλήματος (γεωμετρία πεδίου, τελεστής ΜΔΕ και οριακές συνθήκες ή/και συνθήκες διεπαφών). Οι μέθοδοί της μπορούν επίσης να δημιουργήσουν ή/και να βελτιώσουν πλέγματα (τριγωνικά στοιχεία) για κάθε υποπεδίο, να επιλύσουν τα τοπικά προβλήματα ΜΔΕ και να απεικονίσουν τα υπολογισμένα αποτελέσματα για ολόκληρο το πεδίο και για τις διεπαφές. Ο προεπιλεγμένος επιλυτής στα προγράμματα της FEniCS είναι η αραιή LU αποσύνθεση, καθώς είναι ισχυρή για κάποιες χιλιάδες αγνώστους στο σύστημα εξισώσεων. Παρόλα αυτά, η αραιή LU αποσύνθεση γίνεται αργή και πολύ απαιτητική σε μνήμη για μεγάλα προβλήματα. Για αυτό το λόγο η FEniCS παρέχει εναλλακτικά και επαναληπτικές μεθόδους, όπως οι *preconditioned Krylov* επιλυτές, οι οποίοι είναι γρηγορότεροι και απαιτούν σημαντικά λιγότερη μνήμη. Μια πλήρης λίστα των διαθέσιμων *Krylov* επιλυτών και *preconditioners* μπορεί να βρεθεί στην τεκμηρίωση της FEniCS [19]. Στην πραγματικότητα, οι υλοποιήσεις των επιλυτών που τίθενται σε δράση εξαρτώνται από την επιλογή του πακέτου γραμμικής άλγεβρας. Η FEniCS υποστηρίζει διάφορα πακέτα γραμμικής άλγεβρας, που αποκαλούνται *backends* στην ορολογία της FEniCS. Το PETSc είναι το προεπιλεγμένο backend και εναλλακτικά *backends* που υποστηρίζονται είναι τα uBLAS, Epetra (Trilinos) και MTL4.

Για την καλύτερη κατανόηση του τρόπου με τον οποίο ένα πρόβλημα επιλύεται στη FEniCS πειραματιστήκαμε με το ακόλουθο πρόβλημα-μοντέλο, που είναι το Poisson πρόβλημα:

$$\begin{aligned} -\nabla^2 u(x) &= f(x), \quad x \text{ in } \Omega, \\ u(x) &= u_0(x), \quad x \text{ on } \partial\Omega. \end{aligned}$$

Εδώ, η $u(x)$ είναι η άγνωστη συνάρτηση, η $f(x)$ είναι μια ορισμένη συνάρτηση, το ∇^2 είναι ο τελεστής Laplace, το Ω είναι το χωρικό πεδίο και το $\partial\Omega$ είναι το όριο του Ω . Μια ΜΔΕ σαν αυτή, μαζί με ένα πλήρες σύνολο από οριακές συνθήκες αποτελούν ένα πρόβλημα οριακών τιμών, που πρέπει να οριστεί πλήρως πριν την επίλυσή του στη FEniCS.

2.5.3. RabbitMQ

Για την παράλληλη υλοποίηση στην πλατφόρμα FEniCS χρησιμοποιήθηκε το RabbitMQ [20]. Το RabbitMQ είναι ένας ελαφρύς, αξιόπιστος, επεκτάσιμος και φορητός μεσολαβητής μηνυμάτων. Δίνει στις εφαρμογές μια κοινή πλατφόρμα για αποστολή και λήψη μηνυμάτων. Το RabbitMQ τρέχει σε όλα τα κύρια λειτουργικά συστήματα και είναι εύκολο στη χρήση. Υποστηρίζει έναν τεράστιο αριθμό από πλατφόρμες ανάπτυξης, μεταξύ των οποίων και την Python, που είναι η πλατφόρμα ανάπτυξης της FEniCS. Το RabbitMQ βασίζεται στο Advanced Message Queuing Protocol (AMQP). Το AMQP είναι ένα πρωτόκολλο μηνυμάτων που ασχολείται με εκδότες και καταναλωτές. Οι εκδότες παράγουν τα μηνύματα, οι καταναλωτές τα λαμβάνουν και τα επεξεργάζονται.

Στην παρούσα μελέτη δημιουργήθηκε ένα σύστημα κλήσης απομακρυσμένης διαδικασίας (Remote Procedure Call - RPC), βασισμένο στο RabbitMQ καθότι υπάρχει μεγάλη ανάγκη για επικοινωνία κατά τη διάρκεια της παράλληλης επίλυσης των προβλημάτων ΜΔΕ. Στη συνέχεια θα περιγράψουμε τη διαδικασία δημιουργίας ενός RPC συστήματος (που επιστρέφει Fibonacci αριθμούς): ένας πελάτης και ένας server χρησιμοποιώντας το RabbitMQ.

2.5.4. Σειριακή Υλοποίηση

Η geometric (GEO) contraction based μέθοδος υλοποιείται σαν ένα FEniCS πρόγραμμα γραμμένο στην προγραμματιστική γλώσσα Python. Η βιβλιοθήκη DOLFIN χρησιμοποιείται για την εισαγωγή κλάσεων χρήσιμων για τη δημιουργία των υποπεδίων του προβλήματος και τη δημιουργία πλεγμάτων (τριγωνικά στοιχεία) σε αυτά τα υποπεδία. Στη συνέχεια δηλώνουμε και εφαρμόζουμε τις οριακές συνθήκες, καθώς επίσης και τις αρχικές μαντεπιές στις διεπαφές των υποπεδίων. Το πρόβλημα ΜΔΕ πρέπει να εκφραστεί σαν ένα παραμετρικό (variational) πρόβλημα και στη συνέχεια να οριστεί στο πρόγραμμα. Μετά τον υπολογισμό της λύσης, πραγματοποιείται και ο υπολογισμός της παραγώγου.

Η δημιουργία κατάλληλων συναρτήσεων για την λήψη των τιμών της λύσης και της παραγώγου στα σημεία των διεπαφών (όρια των υποπροβλημάτων), ο υπολογισμός των νέων χαλαρωμένων τιμών και το πέρασμά τους πίσω στα υποπροβλήματα σαν ανανεωμένες τιμές για τις διεπαφές ήταν οι κυριότερες προκλήσεις της υλοποίησης της GEO.

2.5.5. Παράλληλη Υλοποίηση

Δεδομένου ότι η μέθοδος GEO είναι εγγενώς παραλληλίσιμη, στην παράλληλη υλοποίησή της κάθε κόμβος επιλύει ένα υποπεδίο. Οι υπολογισμένες τιμές της λύσης και της παραγώγου στα σημεία των διεπαφών από κάθε κόμβο (που επιλύει ένα υποπεδίο που έχει παραπάνω από ένα γειτονικά υποπεδία) στέλνονται σαν ένα RabbiMQ μήνυμα, έτσι ώστε η νέα επανάληψη να ξεκινήσει και στον άλλο κόμβο επίσης. Αυτό το σχήμα, που δεν εμπεριέχει ξεχωριστούς κόμβους για να διαχειρίζονται τις διεπαφές εξυπηρετεί στη μείωση του αριθμού των μηνυμάτων που ανταλλάσσονται, σε μια προσπάθεια να ελαχιστοποιηθεί ο συνολικός χρόνος επικοινωνίας.

Αρα, το μεσαίο πεδίο το επιλύει ένας client και τα άλλα δύο ένας server το καθένα. Αυτοί οι servers χειρίζονται και τις διεπαφές.

2.6 Ασύγχρονη Μέθοδος Χαλάρωσης Διεπαφών

2.6.1. Εισαγωγή

Η επίλυση των σύνθετων μερικών διαφορικών εξισώσεων (PDEs) είναι ένα αναπόφευκτο βήμα στις πολυάριθμες επιστημονικές εφαρμογές. Εντούτοις, αυτό είναι μια διαδικασία υπολογιστικά απαιτητική και με μεγάλες απαιτήσεις μνήμης για τις μεγάλης κλίμακας διαφορικές εξισώσεις. Αυτό ισχύει ιδιαίτερα για τα προβλήματα πολλαπλών-περιοχών και πολλαπλής-φυσικής όπου διαφορετικοί τελεστές PDE εφαρμόζονται στα διαφορετικά subdomains. Οι τεχνικές αποσύνθεσης περιοχών [1], [2] χρησιμοποιήθηκαν πρώτιστα για να αντιμετωπίσουν τέτοια προβλήματα. Περιλαμβάνουν την αποσύνθεση στο γραμμικό επίπεδο άλγεβρας με διακριτοποίηση της περιοχής και της εξίσωσης με την επιθυμητή μέθοδο. Αν και, το κύριο μειονέκτημά τους είναι η μη ευελιξία στη χρήση της διαφορετικής μεθόδου ανά subdomain του αρχικού προβλήματος. Η μεθοδολογία χαλάρωσης διεπαφών είναι μια ενδιαφέρουσα εναλλακτική λύση [3]– [5].

Το κύριο πλεονέκτημα των μεθόδων IR είναι ότι μεταχειρίζονται ένα πρόβλημα πολλαπλών-περιοχών /πολλαπλής φυσικής ως αόριστα συνδεδεμένο σύστημα των υπο-προβλημάτων που αποτελούνται από τα πολύ απλούστερα προβλήματα PDE και σχετικά με τη γεωμετρία και το διαφορικό τελεστή. Πιο συγκεκριμένα, στις μεθόδους IR η περιοχή PDE αποσυντίθεται στα subdomains, που παράγονται από την ελλοχεύουσα φυσική ή από τις υψηλές υπολογιστικές ανάγκες.

Σε αυτή τη δράση έγινε μια ασύγχρονη παράλληλη εφαρμογή της μεθόδου GEO/IR [6]. Η προτεινόμενη μεθοδολογία κατορθώνει να υπερνικήσει την παραδοσιακή δυσχέρεια των παράλληλων μεθόδων IR και να εκμεταλλευτεί τον ελλοχεύοντα έμφυτο παραλληλισμό της λύσης των subdomains, που καθιστούν τη μέθοδο εξελικτική στη μεγάλη κατανεμημένη παράλληλη εγκατάσταση. Μια προκαταρκτική πειραματική αξιολόγηση του ασύγχρονου GEO καταδεικνύει την εξελικτικότητα και την αποτελεσματικότητά της για τη λύση των αρκετά μεγάλων προβλημάτων πολλαπλών-περιοχών και πολλαπλής-φυσικής.

2.6.2. Σειριακός αλγόριθμος

Το πρόβλημα PDE διαιρείται στα υπο-προβλήματα, κάθε ένα από τα οποία διακριτοποιείται με την κατάλληλη μέθοδο. Οι οριακές συνθήκες πρέπει να

διευκρινιστούν και να εφαρμοστούν σε κάθε υπο-πρόβλημα, καθώς επίσης και οι αρχικές συνθήκες στις διεπαφές μεταξύ των υποπροβλημάτων. Κατόπιν παρουσιάζονται η λύση και η κλίση κάθε υπο-προβλήματος. Αφότου εκτελείται το βήμα λύσης, πρέπει να πάρουμε τις τιμές της λύσης και της κλίσης στα σημεία διεπαφών και να υπολογίσουμε τις νέες χαλαρωμένες τιμές. Με τη μέθοδο GEO / IR [6] οι νέες χαλαρωμένες τιμές στα σημεία διεπαφών λαμβάνονται με την προσθήκη στα παλαιά ενός γεωμετρικά σταθμισμένου μέσου όρου των κανονικών παραγώγων ορίου των παρακείμενων subdomains.

2.6.3. Σύγχρονος παράλληλος αλγόριθμος

Η GEO έχει μια βαθιά στατική παράλληλη εφαρμογή, όπου κάθε κόμβος αντιπροσωπεύει έναν υπολογιστικό πυρήνα που λύνει ένα subdomain. Οι υπολογισμένες λύσεις PDE και οι κλίσεις στα σημεία διεπαφών από κάθε κόμβο στέλνονται ως μήνυμα στον κόμβο που χειρίζεται το παρακείμενο subdomain. Όταν κάθε κόμβος λαμβάνει όλες τις παρακείμενες πληροφορίες σημείων διεπαφών, υπολογίζει τις νέες χαλαρωμένες τιμές σημείων διεπαφών, που εξυπηρετούν τόσο τις νέες εκτιμήσεις για την επόμενη επανάληψη των παρακείμενων subdomains και τις στέλνει πίσω σαν μηνύματα. Όταν οι νέες τιμές διεπαφών σταλούν σε όλα τα subdomains, η επόμενη επανάληψη μπορεί να αρχίσει σε όλους τους κόμβους.

2.6.4. Ασύγχρονος παράλληλος αλγόριθμος

Η κύρια δυσχέρεια του παράλληλου αλγορίθμου GEO προκύπτει όταν τα υπο-προβλήματα έχουν πολλούς γείτονες, τα οποία είναι άνισου μεγέθους, και έτσι άνισου υπολογιστικού φόρτου εργασίας. Κατά συνέπεια, κάθε υπο-πρόβλημα πρέπει να περιμένει έως ότου τελειώνουν οι όλοι γείτονές του το βήμα λύσης τους και οι νέες χαλαρωμένες τιμές έχουν υπολογιστεί για όλες τις διεπαφές.

Στην ασύγχρονη προσέγγιση παραλληλισμού, το βήμα συγχρονισμού είναι χρονικά χαλαρωμένο και κάθε subdomain υπολογίζεται επαναληπτικά βασισμένο στις πρόσφατες διαθέσιμες τιμές διεπαφών που παρέχονται από τις γειτονικές υπο-περιοχές. Το παραπάνω σχήμα δημιουργεί μια σειρά αναμονής μηνυμάτων για κάθε subdomain, με τη σειρά αναμονής που γεμίζουν με τα μηνύματα για τις νέες τιμές διεπαφών από τα γειτονικά subdomains. Στην περίπτωση που δεν υπάρχει κανένα νέο μήνυμα, ο επιλυτής του subdomain περιμένει. Εάν η σειρά αναμονής δεν είναι κενή, όλα τα μηνύματα καταναλώνονται σε μια διάταξη LIFO για κάθε διεπαφή και τα παλαιότερα μηνύματα απορρίπτονται. Οι νέες τιμές για κάθε διεπαφή υπολογίζονται τοπικά και αρχίζει μια νέα επανάληψη των επιλυτών PDE για το αντίστοιχο subdomain. Στο τέλος κάθε τοπικής επανάληψης, οι νέες τοπικές υπολογισμένες τιμές σε όλες τις διεπαφές στέλνονται ασύγχρονα στους αντίστοιχους γείτονες. Η διαδικασία τελειώνει όταν για όλες τις διεπαφές αποτυγχάνουν να βρουν όλα τα subdomains τη σημαντική διαφορά στις νέες τιμές που υπολογίζονται.

2.7 Παραλληλοποίηση Alorex

Στην παρούσα δράση παραλληλοποιήθηκε από την ΚΕΟ 3 η μέθοδος Alorex στην εφαρμογή της για το πρόβλημα της υφαλμύρισης υδροφορέων γλυκών υδάτων λόγω υπεράντλησης στα πηγάδια στο Βαθύ της Καλύμνου που έχει υλοποιηθεί από την ΚΕΟ 1. Στο συγκεκριμένο πρόβλημα η Alorex υπολογίζει επαναληπτικά νέες

τιμές αντλήσεων των πηγαδιών σε μια προσπάθεια να βρεθεί εκείνος ο συνδυασμός αντλήσεων που δίνει το μέγιστο κέρδος χωρίς ταυτόχρονα να υφαιμυριστεί κάποιο πηγάδι.

Στην παράλληλη υλοποίηση της ΚΕΟ3 ο υπολογισμός νέων τιμών αντλήσεων των πηγαδιών διανέμεται σε μεταβλητό αριθμό εικονικών μηχανών που μπορεί να ορίζει ο χρήστης ή το σύστημα ανάλογα με τους διαθέσιμους πόρους. Αφού υπολογιστούν διάφοροι συνδυασμοί αντλήσεων επιλέγεται ένας συνδυασμός με έναν από τους παρακάτω τρόπους είτε βάσει επιλογής του χρήστη είτε τυχαία:

- η υψηλότερη συνολική άντληση,
- τυχαία επιλογή όπου ο κάθε συνδυασμός είναι ισοπίθανος
- τυχαία επιλογή όπου ο κάθε συνδυασμός έχει πιθανότητα επιλογής ανάλογη της συνολικής άντλησης

Επίσης δίνεται η δυνατότητα βάσει επιλογής του χρήστη οι τιμές των αντλήσεων στις πρώτες επαναλήψεις να μεταβάλλονται εντονότερα από ότι στις επόμενες.

Τα αποτελέσματα της εφαρμογής της παράλληλης υλοποίησης της Αlopex στο πρόβλημα της υφαιμύρισης υδροφορέων γλυκών υδάτων λόγω υπεράντλησης στα πηγάδια στο Βαθύ της Καλύμνου είναι ενθαρρυντικά καθώς παρατηρείται σύγκλιση σε μικρότερο αριθμό επαναλήψεων.

Συνδυασμός Αlopex με Interface Relaxation μεθόδους

Το πρόβλημα των πηγαδιών στο Βαθύ της Καλύμνου είναι ένα πρόβλημα πολλαπλών πεδίων / πολλαπλών φυσικών, οπότε στην παρούσα δράση εξετάζεται η δυνατότητα επίλυσής του με Interface Relaxation μεθόδους σε κάθε Αlopex επανάληψη. Αρχικά εφαρμόστηκε η GEO IR μέθοδος, η οποία δεν έδωσε καλά αποτελέσματα λόγω της φύσης του προβλήματος. Οπότε, στη συνέχεια εφαρμόστηκε η ROB IR μέθοδος η οποία δίνει ενθαρρυντικά αποτελέσματα που χρήζουν όμως περαιτέρω μελέτης.

3. Αποτελέσματα Παράλληλης Υλοποίησης Αλγορίθμων

Σε αυτό το κεφάλαιο παρουσιάζονται τα αποτελέσματα της παράλληλης υλοποίησης των αλγορίθμων που παρουσιάστηκαν στο κεφάλαιο 2. Αρχικά ο έλεγχος έγινε με απλά παραδείγματα μοντέλων πολλαπλών πεδίων, ακολούθησαν πιο σύνθετα προβλήματα, σε συνεργασία με μέλη της ΚΕΟ 2. Τέλος, σε συνεργασία με την ΚΕΟ 1, η παραπάνω μεθοδολογία εφαρμόστηκε στο πραγματικό πρόβλημα υφαιμύρισης των πηγαδιών στο Βαθύ της Καλύμνου.

3.1. Μέθοδοι χαλάρωσης στη Διεπαφή

Μία σύντομη μελέτη αξιολόγησης απόδοσης των μεθόδων μπορεί να βρεθεί στο [21] (του κεφ. 2.3) όπου παρουσιάζονται και συζητούνται αριθμητικά δεδομένα για μονοδιάστατα προβλήματα. Τα προβλήματα αυτά μπορεί να θεωρηθούν πολύ απλά για να έχουν πρακτική αξία, αλλά ο πειραματισμός μαζί τους ανέδειξε διάφορες πλευρές της φύσης των μεθόδων χαλάρωσης στη διεπαφή. Τα ποιοτικά τους

χαρακτηριστικά φαίνονται στον πίνακα 1. Η ταχύτητα σύγκλισης, όπως φαίνεται μπορεί να είναι χαμηλή, μέση ή υψηλή. Οι επαναλήψεις μπορούν να προσεγγίζουν την πραγματική λύση μονότονα ή όχι και μπορεί να υπάρχουν μία, δύο ή καμία παράμετροι χαλάρωσης για να επιταχύνουν τη σύγκλιση. Κάποιες μέθοδοι είναι ενός βήματος, άλλες όχι και κάποιες χρησιμοποιούν ιστορία (η καινούργια τιμή στη διεπαφή ισούται με την παλιά συν έναν όρο διόρθωσης), ενώ άλλες όχι.

	Ταχύτητα	Μονοτ/κότητα Σύγκλισης	Αριθμός Παραμέτρων	Διαθέσιμη Θεωρία	Ιστορία	Ενός Βήματος
VE	Μέση	Όχι	2	Λίγη	Όχι	Όχι
EO	Μέση	Ναι	2	Λίγη	Ναι	Ναι
EW	Μέση	Όχι	Καμία	Όχι	Ναι	Ναι
OB	Χαμηλή	Όχι	1	Λίγη	Όχι	Ναι
CO	Μέση	Όχι	1	Λίγη	Ναι	Όχι
HO	Μέση	Ναι	Καμία	Λίγη	Ναι	Ναι
PO	Υψηλή	Ναι	1	Όχι	Ναι	Όχι

Πίνακας 3: Ιδιότητες των επτά μεθόδων χαλάρωσης στη διεπαφή

Τα βασικά συμπεράσματα που προκύπτουν από αυτή τη συγκριτική ανάλυση είναι ότι υπάρχουν αρκετές μέθοδοι χαλάρωσης στη διεπαφή που δείχνουν ότι έχουν τις δυνατότητες να δουλέψουν αποδοτικά και ότι μένουν ακόμα πολλά να μάθουμε για τη συμπεριφορά τους και για το πώς να διαλέγουμε ανάμεσά τους την κατάλληλη, αλλά και το πώς να διαλέγουμε τις παραμέτρους τους.

Στην παρούσα μελέτη η μέθοδος που θα υλοποιήσουμε είναι η GEO γιατί είναι μία από αυτές που έχουν πολύ καλά χαρακτηριστικά, όπως μονοτονικότητα στη σύγκλιση, χρήση ιστορίας (η καινούργια τιμή στη διεπαφή ισούται με την παλιά συν έναν όρο διόρθωσης) και το ότι είναι ενός βήματος. Επίσης γιατί για τη GEO υπάρχει μια πρόσφατη υλοποίηση [18] (κεφ. 2.3) με την οποία μπορούμε να συγκρίνουμε την προτεινόμενη υλοποίηση. Η υλοποίηση βέβαια και των υπόλοιπων έξι μεθόδων στο περιβάλλον επίλυσης προβλημάτων πολλαπλών πεδίων/ πολλαπλών φυσικών που υλοποιούμε είναι στα άμεσα μελλοντικά μας σχέδια.

3.1.1. Πειράματα

Για την εξέταση της ορθότητας και της απόδοσης της υλοποίησης, χρησιμοποιείται το ακόλουθο ελλειπτικό πρόβλημα:

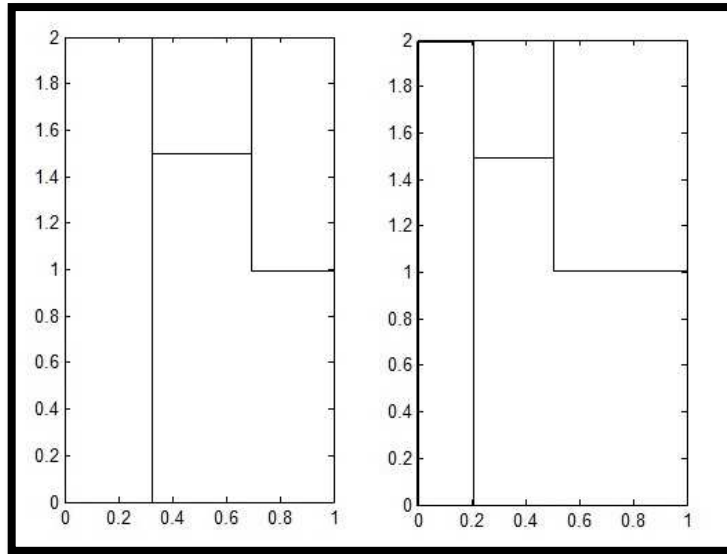
$$Lu(x,y) \equiv \nabla u(x,y) + \gamma^2 u(x,y) = f(x,y), \quad (x,y) \in \Omega$$

$$u(x,y) = u^b(x,y), \quad (x,y) \in \Omega$$

με $f(x,y)$ και $u^b(x,y)$ επιλεγμένα έτσι ώστε η πραγματική λύση να είναι:

$$u(x,y) = e^{y(x=4)} x(x-1)(x-0.7)y(y-0.5) \quad (3)$$

Πιο συγκεκριμένα μελετώνται δύο διαφορετικά προβλήματα ΜΔΕ που αποτελούνται από την προηγούμενη διαφορική εξίσωση και οριακές συνθήκες και τα δύο διαφορετικά πεδία που αποτυπώνονται στην εικόνα 6.



Εικόνα 2: Τα πεδία του ομοιόμορφου (αριστερά) και του ανομοιόμορφου (δεξιά) προβλήματος

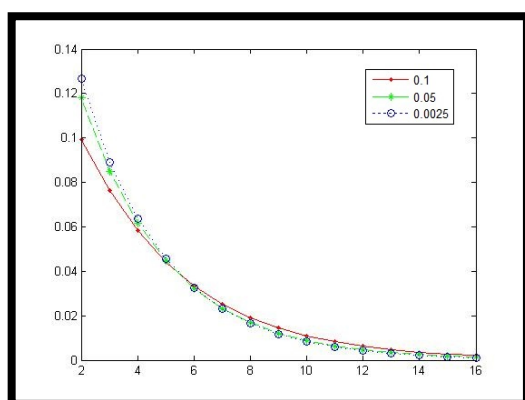
Για το ομοιόμορφο πρόβλημα τα σημεία των διεπαφών είναι στα $x_1 = \frac{1}{3}$ και $x_2 = \frac{2}{3}$ και για το ανομοιόμορφο πρόβλημα στα $x_1 = \frac{1}{5}$ και $x_2 = \frac{1}{2}$ και το $\gamma^2 = 2$. Τα πειράματα εκτελούνται για διάφορες τιμές της παραμέτρου διακριτοποίησης h , η οποία έχει θεωρηθεί ίση τόσο στην κατεύθυνση x όσο και στην κατεύθυνση y . Τα προκύπτοντα μεγέθη του πλέγματος παρουσιάζονται στον πίνακα 4.

Ομοιόμορφο πρόβλημα				
Περίπτωση	H	Αριστερό υποπεδίο	Μεσαίο υποπεδίο	Δεξί υποπεδίο
Π1	0.1	4x21	4x6	4x11
Π2	0.05	8x41	8x11	8x21
Π3	0.025	14x81	14x21	14x41
Π4	0.0125	28x161	28x41	28x81
Π5	0.00625	55x321	55x81	55x161
Π6	0.003125	108x641	108x161	108x321
Π7	0.0015625	214x1281	214x321	214x641
Ανομοιόμορφο πρόβλημα				
Περίπτωση	H	Αριστερό υποπεδίο	Μεσαίο υποπεδίο	Δεξί υποπεδίο
Π1	0.1	3x21	4x6	6x11
Π2	0.05	5x41	7x11	11x21
Π3	0.025	9x81	13x21	21x41
Π4	0.0125	17x161	25x41	41x81

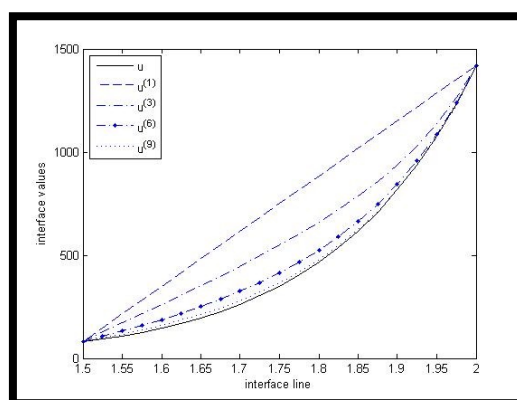
Π5	0.00625	33x321	49x81	81x161
Π6	0.003125	65x641	97x161	161x321
Π7	0.0015625	129x1281	193x321	321x641

Πίνακας 4: Περιπτώσεις ελέγχου που εξετάστηκαν μαζί με το βήμα διακριτοποίησης και το μέγεθος πλέγματος για το αριστερό, μεσαίο και δεξί υποπεδίο του ομοιόμορφου και του ανομοιόμορφου προβλήματος

Το ιστορικό της σύγκλισης αποτυπώνεται στις επόμενες εικόνες. Πιο συγκεκριμένα, στο αριστερό γράφημα παρουσιάζεται η νόρμα μεγίστου της σχετικής διαφοράς διαδοχικών επαναλήψεων για τη διεπαφή $x_1 = \frac{1}{3}$ του ομοιόμορφου προβλήματος για $h = 0.1, 0.05, 0.025$. Όπως άλλωστε υπόσχεται η μεθοδολογία χαλάρωσης στη διεπαφή, παρατηρούμε ότι ο ρυθμός σύγκλισης είναι ανεξάρτητος του τοπικού βήματος διακριτοποίησης h . Για το ομοιόμορφο πρόβλημα και για τη διεπαφή $x_1 = \frac{1}{3}$ στο δεξί γράφημα παρουσιάζονται η ακριβής και οι υπολογισμένες λύσεις για τις επαναλήψεις 1, 3, 6, 10 για $h = 0.05$.



Εικόνα 3: Η νόρμα μεγίστου της σχετικής διαφοράς διαδοχικών επαναλήψεων για τη διεπαφή $x_1=1/3$ του ομοιόμορφου προβλήματος για $h=0.1,0.05,0.025$.



Εικόνα 4: Η ακριβής και οι υπολογισμένες λύσεις για το ομοιόμορφο πρόβλημα και για τη διεπαφή $x_1=1/3$ για τις επαναλήψεις 1, 3, 6, 10 για $h=0.05$.

Στον πίνακα 5 παρουσιάζονται οι συνολικοί χρόνοι εκτέλεσης της σειριακής και της παράλληλης υλοποίησης.

	Π1	Π2	Π3	Π4	Π5	Π6	Π7
Ομοιόμορφο πρόβλημα							
Σειριακή Υλοποίηση	1.900	2.664	5.216	12.883	40.883	178.217	996.348
Παράλληλη Υλοποίηση	3.495	4.068	6.140	13.595	39.808	154.633	723.731
Ανομοιόμορφο πρόβλημα							

Σειριακή Υλοποίηση	2.294	2.854	5.111	13.251	37.266	179.561	1057.644
Παράλληλη Υλοποίηση	3.720	4.264	5.732	10.633	34.172	126.607	918.838

Πίνακας 5: Χρόνοι συνολικής εκτέλεσης σειριακής και παράλληλης υλοποίησης

Για το ομοιόμορφο πρόβλημα η παράλληλη υλοποίηση αρχίζει να γίνεται γρηγορότερη από την περίπτωση Π5 και μετά, ενώ για το ανομοιόμορφο από την περίπτωση Π4 και μετά. Αυτό είναι λογικό μιας και στο ομοιόμορφο πρόβλημα ένα από τα 3 υποπροβλήματα είναι σημαντικά μεγαλύτερο από τα υπόλοιπα και άρα είναι το κυρίαρχο στο χρόνο εκτέλεσης και της σειριακής και της παράλληλης υλοποίησης.

Η σειριακή εκτέλεση πραγματοποιείται σε έναν κόμβο με 4 Intel(R) Xeon(R) CPU E5-2620, 2.00GHz επεξεργαστές και 2GB RAM και η παράλληλη σε 3 κόμβους με την ίδια διαμόρφωση, στην Cloud Υποδομή του Εργαστηρίου Αναγνώρισης Προτύπων, του τμήματος Μηχ/κών Η/Υ και Πληροφορικής.

Ο χρόνος των σειριακών εκτελέσεων περιλαμβάνει το συνολικό χρόνο των υπολογισμών για τα τρία πεδία και τις δύο διεπαφές για 16 γενιές. Στις παράλληλες εκτελέσεις, ο συνολικός χρόνος αντιστοιχεί στο χρόνο που απαιτείται για την επίλυση του μεγαλύτερου από τα τρία υποπεδία μαζί με τους υπολογισμούς για τις διεπαφές και την απαραίτητη επικοινωνία, για 16 γενιές. Οι περιπτώσεις ελέγχου επιλέχθηκαν για να διερευνηθεί η συμπεριφορά της παράλληλης υλοποίησης και όχι για λόγους ακρίβειας. Όσο τα πλέγματα γίνονται λεπτότερα, ο φόρτος εργασίας αυξάνεται σημαντικά, ενώ ο χρόνος επικοινωνίας αυξάνεται με μια τάξη μεγέθους μικρότερη. Αυτός είναι ο κύριος λόγος που κέρδος παρατηρείται για λεπτά πλέγματα.

3.2. Εφαρμογή Υβριδικού Παράλληλου Monte Carlo Επιλυτή Μ.Δ.Ε.

3.2.1. Πειραματικά αποτελέσματα

Έχουμε εκτελέσει έναν εκτενή αριθμό πειραμάτων σε ένα εικονικό cluster 8 μηχανημάτων που τρέχει σε μια OpenStack Cloud υποδομή του υπολογιστικού κέντρου του Τμήματος Υπολογιστών και Πληροφορικής του ΤΕΙ Δυτικής Ελλάδας. Η cloud υποδομή χρησιμοποιεί ένα Dell Blade Enclosure με 16 dual processor Xeon servers, με κάθε επεξεργαστή να έχει 6 hyper-threading πυρήνες και 48 Gbytes κύριας μνήμης. Τα blades είναι διασυνδεδεμένα με 1 GBit Ethernet. Κάθε εικονικό μηχανήμα έχει 4 εικονικές CPUs και 8 GBytes RAM.

Για όλα τα προβλήματα έχουμε δημιουργήσει όλους τους πιθανούς συνδυασμούς εκτέλεσης για το μέγιστο αριθμό νημάτων $nt=\{1, 2, 4, 6, 8\}$ ανά διεργασία $np=\{1, 2, 4, 8, 16, 32, 64\}$ και τον αριθμό των τυχαίων περιπάτων που πραγματοποιούνται ανά κόμβο από τον Monte Carlo αλγόριθμο $walks = \{20K, 40K, 80K, 160K\}$, ώστε να μπορέσουμε να καθορίσουμε την επιτάχυνση και την κλιμάκωση της προτεινόμενης προσέγγισης. Για το profiling της υβριδικής υλοποίησης έχουμε χρησιμοποιήσει τον mpiP profiler [10] που είναι μια ελαφριά βιβλιοθήκη profiling για MPI εφαρμογές. Επειδή απλά συλλέγει στατιστική πληροφορία για τις MPI συναρτήσεις, το mpiP δημιουργεί σημαντικά λιγότερη επιβάρυνση και πολύ λιγότερα δεδομένα από εργαλεία ανίχνευσης. Όλη η συλλεγόμενη πληροφορία από το mpiP είναι τοπική ανά διεργασία.

Οι Πίνακες 3-10 παρουσιάζουν τους χρόνους εκτέλεσης και ο Πίνακας 11 την επιτάχυνση που επιτεύχθηκε για όλα τα προβλήματα εισόδου.

Πρόβλημα	1						
Np	1	2	4	8	6	2	64
Nt	1	2	4	6	6	8	8
40K	6	8	6	3	3	4	3
80K	9	14	11	5	4	6	5
160K	8	4	19	9	6	6	8

Πίνακας 4: Υβριδικοί χρόνοι εκτέλεσης – πρόβλημα 1

Πρόβλημα	2						
Np	1	2	4	8	6	2	4
Nt	1	2	4	6	6	8	8
40K	26	9	7	4	3	3	4
80K	50	14	10	6	4	5	5
160K	99	25	19	9	6	7	9

Πίνακας 5: Υβριδικοί χρόνοι εκτέλεσης – πρόβλημα 2

Πρόβλημα	3						
Np	1	2	4	8	6	2	4
Nt	1	2	4	6	6	8	8
40K	25	0	5	3	3	2	3
80K	0	6	10	4	3	4	4
160K	9	8	8	8	5	6	6

Πίνακας 6: Υβριδικοί χρόνοι εκτέλεσης – πρόβλημα 3

Πρόβλημα	4						
Np	1	2	4	8	16	32	64
Nt	1	2	4	6	6	8	8
40K	538	183	120	20	62	68	77
80K	931	298	170	88	80	86	90
160K	1777	511	295	124	109	122	129

Πίνακας 7: Υβριδικοί χρόνοι εκτέλεσης – πρόβλημα 4

Πρόβλημα	5						
Np	1	2	4	8	16	32	64
Nt	1	2	4	6	6	8	8
40K	534	183	120	70	62	68	74
80K	946	298	170	88	180	86	90
160K	1757	511	295	124	109	122	129

Πίνακας 8: Υβριδικοί χρόνοι εκτέλεσης – πρόβλημα 5

Πρόβλημα	6						
Np	1	2	4	8	16	32	64
Nt	1	2	4	6	6	8	8
40K	494	176	423	78	72	75	77
80K	847	273	164	97	82	92	94
160K	1550	464	292	135	106	121	30

Πίνακας 9: Υβριδικοί χρόνοι εκτέλεσης – πρόβλημα 6

Πρόβλημα	7						
Np	1	2	4	8	16	32	64
Nt	1	2	4	6	6	8	8
40K	365	176	109	78	72	75	77
80K	713	273	164	97	82	92	94
160K	1416	464	292	135	105	121	130

Πίνακας 10: Υβριδικοί χρόνοι εκτέλεσης – πρόβλημα 7

Προβλήματα εισόδου							
Np	1	2	4	8	16	32	64
Nt	1	2	4	6	6	8	8
40K	2008	745	610	306	277	295	312
80K	3576	1196	699	385	335	371	382
160K	6796	2037	1230	544	446	505	541

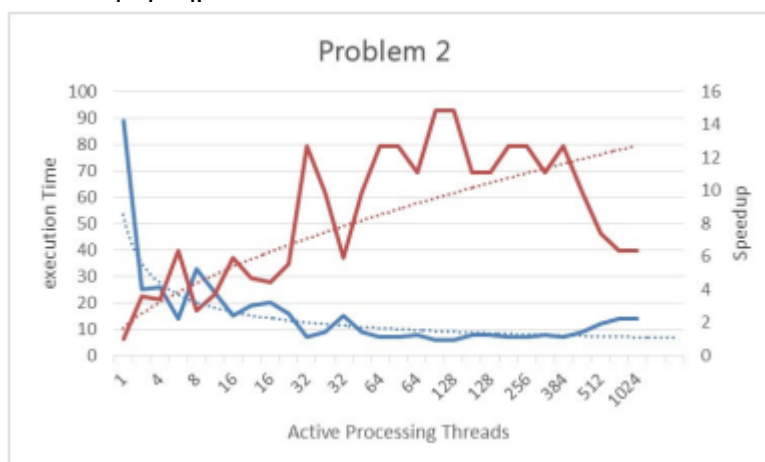
Πίνακας 11: Υβριδικοί χρόνοι εκτέλεσης – άθροισμα

Επιτάχυνση προβλημάτων εισόδου						
Np	2	4	8	16	32	64
Nt	2	4	6	6	8	8
Min	2.81	2.03	6.33	6.86	6.59	6.42
Avg	3.08	4.17	8.85	10.60	9.53	9.12
Max	3.34	5.13	11.48	14.62	12.81	11.92

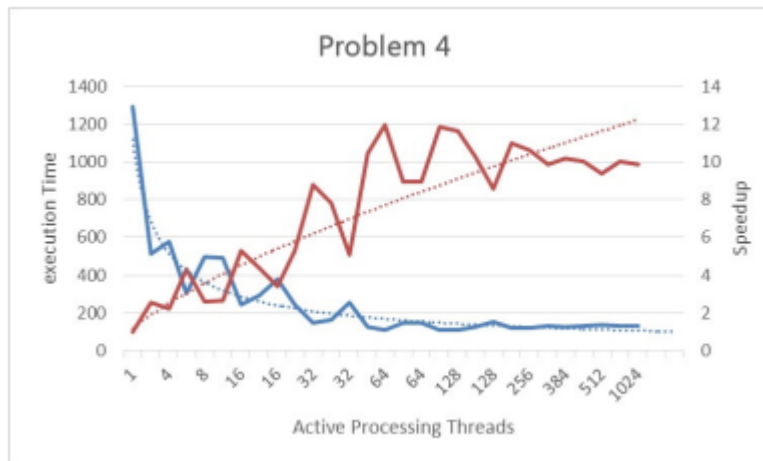
Πίνακας 12: Επιτάχυνση

Η λεπτομερής ανάλυση των εκτελέσεων δείχνει ότι ο αλγόριθμος κλιμακώνεται σχεδόν γραμμικά σε σχέση με τον αριθμό των ενεργών νημάτων επεξεργασίας, δηλαδή τον αριθμό των διεργασιών πολλαπλασιασμένο με τον αριθμό των νημάτων, έως ότου ο χρόνος εκτέλεσης του Monte Carlo μειωθεί σημαντικά λόγω της παραλληλοποίησης και οι υπόλοιποι πυρήνες τώρα κυριαρχούν στο χρόνο εκτέλεσης.

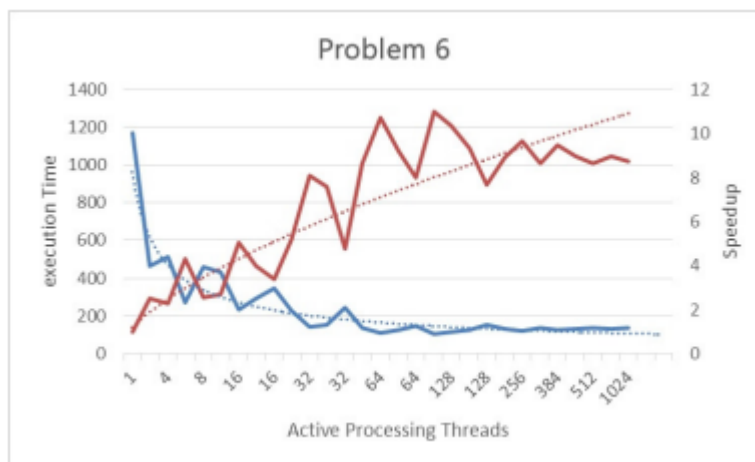
Οι Εικόνες 21-24 παρουσιάζουν την επιτάχυνση και το χρόνο εκτέλεσης για κάποια από τα προβλήματα εισόδου.



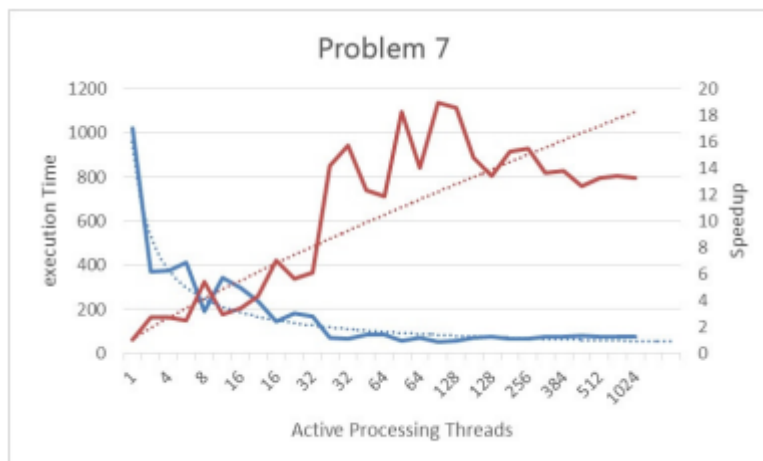
Εικόνα 21: Χρόνος εκτέλεσης και επιτάχυνση - Πρόβλημα 2



Εικόνα 22: Χρόνος εκτέλεσης και επιτάχυνση - Πρόβλημα 4



Εικόνα 23: Χρόνος εκτέλεσης και επιτάχυνση - Πρόβλημα 6



Εικόνα 24: Χρόνος εκτέλεσης και επιτάχυνση - Πρόβλημα 7

3.2.2. Συμπεράσματα

Στα προηγούμενα κεφάλαια παρουσιάστηκε η ακολουθιακή απόδοση ενός επιλυτή ΜΔΕ που εφαρμόζεται σε ελλειπτικές ΜΔΕ (στη Poisson συγκεκριμένα) σε ορθογώνια πολλαπλά πεδία τόσο σε δύο όσο και σε τρεις διαστάσεις. Επίσης παρουσιάστηκαν λεπτομέρειες υλοποίησης και επεκτάσεις στον ακολουθιακό αλγόριθμο, ώστε να επιτρέψει την αποδοτική υλοποίηση σε ένα περιβάλλον

κατανεμημένης μνήμης. Τα πειραματικά αποτελέσματα που παρουσιάζονται δείχνουν ότι μια αποδοτική υλοποίηση του βασικού αλγορίθμου μπορεί να επιτευχθεί σε αρχιτεκτονικές κατανεμημένης μνήμης. Η υβριδική υλοποίηση καταφέρνει να επιδείξει σημαντική επιτάχυνση, ενώ διατηρεί την ποιότητα της λύσης σε δείγματα συνόλων δεδομένων εισόδου. Τα μελλοντικά μας σχέδια περιλαμβάνουν την παραλληλοποίηση του Laplace επιλυτή σε μηχανήματα κατανεμημένης μνήμης και την ενσωμάτωση του υβριδικού αλγορίθμου στην πλατφόρμα FEniCS.

Τα αποτελέσματα αυτής της μελέτης, παρουσιάστηκαν στο διεθνές συνέδριο NUMAN 2014 (Chania, Greece 9/2014) στην εργασία: P. Alefragis, K.A. Spirou and S. Likothanassis: “*Application of a hybrid parallel Monte Carlo PDE Solver on rectangular multi-domains*”.

3.3. Αποτελέσματα Ασύγχρονης Υλοποίησης IR Μεθόδων

Προκειμένου να αξιολογηθούν τα αποτελέσματα απόδοσης του προτεινόμενου ασύγχρονου αλγορίθμου, έχουμε υιοθετήσει ένα απλουστευμένο πρόβλημα που αποτελείται από τρία υπο-πρόβλήματα και δύο διεπαφές.

Περιγράφεται ως εξής:

$$Lu \equiv \nabla^2 u(x, y) + \gamma^2 u(x, y) = f(x, y), (x, y) \in \Omega(2)$$

$$u(x, y) = u^b(x, y), (x, y) \in \partial\Omega$$

$$\text{πού } \Omega \text{ είναι } \left[0, \frac{1}{3}\right] \times [0, 2] \cup \left[\frac{1}{3}, \frac{2}{3}\right] \times [1, 2] \cup \left[\frac{2}{3}, 1\right] \times \left[\frac{1}{2}, 2\right], \text{ και } f(x, y) \text{ και}$$

$u^b(x, y)$ επιλέγεται έτσι ώστε η αληθινή λύση είναι:

$$u(x, y) = e^{y^{(x+4)}} x(x-1)(x-0.7)y(y-0.5)(3)$$

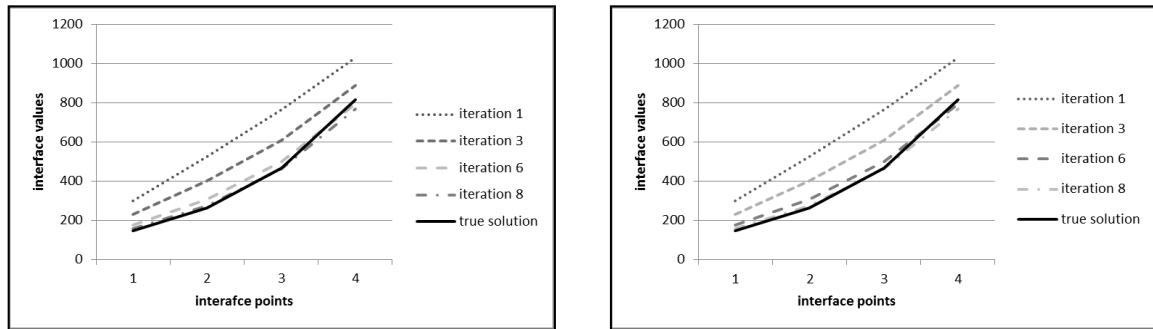
Οι διεπαφές είναι $x_1 = \frac{1}{3}, x_2 = \frac{2}{3}$ ενώ $\gamma^2 = 2$. Επτά διαφορετικές περιπτώσεις

διαμορφώνονται εξετάζοντας τα διαφορετικά μεγέθη πλέγματος, σύμφωνα με επτά διαφορετικές τιμές της διακριτοποιημένης παραμέτρου h , η οποία θεωρείται ίση και στη x και την y κατεύθυνση. Κάθε διαφορετικό μέγεθος πλέγματος αντιστοιχεί σε έναν διαφορετικό αριθμό σημείων διεπαφών, που αυξάνονται από 6 σε 321. Το αριστερό υπο-πρόβλημα είναι περίπου τέσσερις φορές μεγαλύτερο από το μεσαίο, ενώ το δεξιό είναι περίπου δύο φορές μεγαλύτερο από το μεσαίο υπο-πρόβλημα. Οι δύο διεπαφές έχουν τον ίδιο αριθμό σημείων και επομένως είναι ίσου φόρτου εργασίας.

Τα πειράματα εκτελούνται σε 3 διαφορετικά προκαθορισμένες εικονικές μηχανές. Οι εικονικές μηχανές που χρησιμοποιούνται για τα πειράματα, έχουν 4 εικονικούς πυρήνες και 2GB RAM και τρέχουν σε ένα virtualization XenServer περιβάλλον που εγκαθίσταται σε έναν κεντρικό υπολογιστή με 2 X Intel (P) Xeon (P) KME E5-2620, 2.00GHz.

Η ασύγχρονη εφαρμογή GEO συγκρίνεται με μια σύγχρονη παράλληλη εφαρμογή GEO που παρουσιάστηκε πρόσφατα στην [7] (κεφ. 2.3), από την άποψη της ταχύτητας σύγκλισης και του χρόνου εκτέλεσης.

Τα παρακάτω σχήματα απεικονίζουν την πορεία σύγκλισης της ασύγχρονης εφαρμογής GEO στα εσωτερικά σημεία των διεπαφών. Το ΣΧΗΜΑ 1 (α) απεικονίζει τις ακριβείς και υπολογισμένες λύσεις στις επαναλήψεις 1, 3, 6, 8 στη διεπαφή 1 για την περίπτωση c1. Το ΣΧΗΜΑ 1 (β) παρουσιάζει πορεία σύγκλισης της διεπαφής 1 και της περίπτωσης c1 για το σύγχρονο αλγόριθμο. Τα αποτελέσματα μπορούν να συγκριθούν και κάποιος μπορεί να δει την ομοιότητα των υπολογισμένων λύσεων ανά επανάληψη IR και για τους σύγχρονους και ασύγχρονους αλγόριθμους. Παρόμοια αποτελέσματα ισχύουν για όλες τις περιπτώσεις δοκιμής και των δύο διεπαφών, αποδεικνύοντας τη γρήγορη σύγκλιση GEO.



ΣΧΗΜΑ 1. Πραγματική λύση και υπολογισμένες λύσεις στις επαναλήψεις 1, 3, 6, 8 στη διεπαφή 1 για την περίπτωση c1 σε ασύγχρονη (α) και σύγχρονη (β) GEO.

Στο ΣΧΗΜΑ 2 παρουσιάζεται η max norm της σχετικής διαφοράς των διαδοχικών λύσεων στα σημεία διεπαφών σαν συνάρτηση του αριθμού επαναλήψεων

$$r_i = \frac{\|u_{i+1} - u_i\|_{\infty}}{\|u_{i+1}\|_{\infty}}. \text{ Αφορά τη διεπαφή 1 για την περίπτωση c1 και για το προτεινόμενο}$$

ασύγχρονο GEO και για το σύγχρονο GEO. Αν και για τους δύο αλγορίθμους οι αρχικές τιμές στις διεπαφές να είναι οι ίδιες, αυτό δεν απεικονίζεται στο σχήμα επειδή

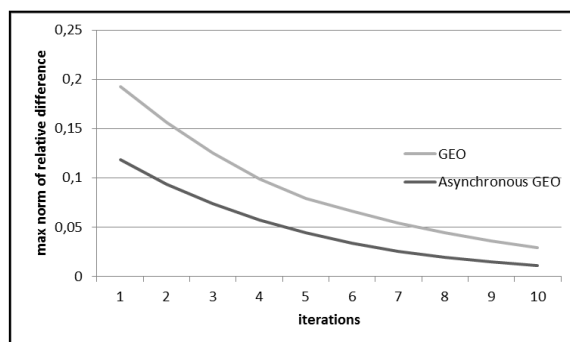
$$\text{η πρώτη τιμή είναι } r_1 = \frac{\|u_2 - u_1\|_{\infty}}{\|u_2\|_{\infty}}. \text{ Πάλι, τα παρόμοια αποτελέσματα ισχύουν και για}$$

όλα τα προβλήματα δοκιμής και τις δύο διεπαφές.

Ο ΠΙΝΑΚΑΣ 1 παρουσιάζει τη σύγκριση των χρόνων εκτέλεσης των δύο εφαρμογών GEO. Για τη σύγχρονη εφαρμογή, ο συνολικός χρόνος αντιστοιχεί στο χρόνο για τη λύση των τριών subdomains μαζί με το χρόνο των υπολογισμών διεπαφών και την απαραίτητη επικοινωνία, για 15 επαναλήψεις, ενώ στην ασύγχρονη εφαρμογή μόνο το μεσαίο subdomain εκτελεί 15 επαναλήψεις. Η σύγκριση των χρόνων εκτέλεσης αποδεικνύει την υψηλή σημασία της ασύγχρονης εφαρμογής, ειδικά καθώς τα πλέγματα γίνονται λεπτότερα και έτσι, τα προβλήματα γίνονται μεγαλύτερα.

Προκειμένου να μελετηθεί και να διευκρινιστεί η συμβολή του προτεινόμενου αλγορίθμου, ως στιγμιαία βήματα για το υιοθετημένο πρόβλημα δοκιμής, προγραμματίζουμε να αναλύσουμε τα timestamps κάθε επανάληψης IR, για κάθε διεπαφή, μέσω της εκτέλεσης του σφαιρικού προβλήματος, και για το σύγχρονο και ασύγχρονο GEO. Θα θεωρήσουμε το σχετικό κανόνα λάθους στις διεπαφές ανά επανάληψη προκειμένου να απεικονιστούν οι διαφορές στη συμπεριφορά των δύο GEO. Τέλος, έχουμε την πρόθεση για τη μελέτη πώς ο φόρτος εργασίας κάθε κόμβου συνδέεται με το χρόνο εκτέλεσης του σφαιρικού προβλήματος, με το χωρισμό και την

ένδειξη του χρόνου για solver PDE, τη μέθοδο IR, το χρόνο επικοινωνίας και το χρόνο μη απασχόλησης και για τους σύγχρονους και τους ασύγχρονους αλγορίθμους.



ΣΧΗΜΑ 2. Ο max norm της σχετικής διαφοράς των διαδοχικών λύσεων στη διαπαφή 1 σαν συνάρτηση του αριθμού επαναλήψεων για την περίπτωση c1 για GEO και ασύγχρονο GEO.

ΠΙΝΑΚΑΣ 1. ΧΡΟΝΟΙ ΕΚΤΕΛΕΣΗΣ ΑΣΥΓΧΡΟΝΟΥ ΚΑΙ ΣΥΓΧΡΟΝΟΥ GEO

	c1	c2	c3	c4	c5	c6	c7
ASYNCHRONOUS GEO	12.12	12.361	11.81	7.727	14.805	38.022	256.084
SYNCHRONOUS GEO	3.495	4.068	6.14	13.595	39.808	154.633	723.731

Τα πειραματικά αποτελέσματα δείχνουν ότι η προσωρινή χαλάρωση του βήματος συγχρονισμού μεταξύ των subdomains δεν υποβιβάζει τις ιδιότητες σύγκλισης της διαδικασίας λύσεων και μειώνει σημαντικά το χρόνο εκτέλεσης ρολογιού του αλγορίθμου. Κατά τη διάρκεια των πειραμάτων, περισσότερο από 300% επιταχύνονται επιτεύχθηκαν έναντι της σύγχρονης παράλληλης εφαρμογής. Η μελλοντική εργασία θα περιλάβει την ολοκλήρωση ενός σχήματος υποκλοπής (stealing) φορτίου για την καλύτερη χρησιμοποίηση των υπολογιστικών πόρων για τα προβλήματα όπου τα subdomains είναι αριθμητικά περισσότερα τους υπολογιστικούς πυρήνες και τις εκτενείς πειραματικές δοκιμές με τα πολλαπλά σενάρια στη χρησιμοποίηση των υπολογιστικών πόρων, έναν μεγαλύτερο αριθμό περιοχών που ποικίλλουν στο μέγεθος και τη δυσκολία και τα διαφορετικά σενάρια στη σύνθεση του αρχικού προβλήματος πολλαπλών-περιοχών.

4. Παραδοτέα

Σύμφωνα με το τεχνικό Δελτίο Προτεινόμενης Πράξης, η ΚΕΟ3 για τη Δράση 3.1 είχε τη συμβατική υποχρέωση για τα παρακάτω παραδοτέα.

- 1 Τεχνική Έκθεση
- 3 δημοσιεύσεις
- Το Software παράλληλης υλοποίησης των αλγορίθμων.

Τελικά. Η ΚΕΟ3 σε συνεργασία με τις άλλες 2 ΚΕΟ έχει ολοκληρώσει τα παρακάτω παραδοτέα.

α. Υλοποίηση σε Clusters, Grids και Cloud

1. 3 Ετήσιες Τεχνικές Εκθέσεις (επισυνάπτονται)
2. 1 Τελική Έκθεση Προόδου
3. 3 επιστημονικά άρθρα σε πρακτικά διεθνών συνεδρίων σχετικά με το αντικείμενο του έργου. Παρακάτω δίνονται οι αναφορές των άρθρων (το πλήρες κείμενο επισυνάπτεται):
 - **A. Korfiati, P. Tsompanopoulou, S. Likothnassis, “Serial and Parallel Implementation of the Interface Relaxation Method GEO”, Numerical Analysis Conference (NumAn 2014), September 2-5, Chania, Greece, 2014.**
 - **P. Alefragis, A. Spyrou and S. Likothanassis, “Application of a hybrid parallel Monte Carlo PDE Solver on rectangular multi-domains” Numerical Analysis Conference (NumAn 2014), September 2-5, Chania, Greece, 2014.**
 - **A.Korfiati, N. Sfika, K. Daloukas, C. Alexakos, P. Tsompanopoulou and S. Likothanassis, “IRaaS: A Cloud Implementation of an Interface Relaxation Method for the Solution of PDEs”, Proceedings of the World Congress on Engineering 2015 Vol I, pp541-546, 2015 (Awarded with a Certificate of Merit)**
 - **A. Korfiati, K. Daloukas, P. Alefragis, P. Tsompanopoulou and S. Likothanassis, “An Asynchronous Interface Relaxation Method for Multi-domain / Multi-Physics Problems”, International Conference on Numerical Analysis Applications and Methods (ICNAAM), Rhodes, Greece, Sept. 23-29, 2015 (accepted).**
4. Στα πλαίσια του έργου ολοκληρώθηκε μια μεταπτυχιακή διπλωματική εργασία, από την εξωτερική συνεργάτιδα κα Αίγλη Κορφιάτη, μεταπτυχιακής φοιτήτριας στο μεταπτυχιακό πρόγραμμα ειδίκευσης “Επιστήμη και τεχνολογία Υπολογιστών”, του Τμήματος Μηχανικών Η/Υ και Πληροφορικής του Πανεπιστημίου Πατρών. Η επίβλεψη της Μ.Δ.Ε. έγινε από τον Επιστημονικό Υπεύθυνο της ΚΕΟ3 καθηγητή Σ. Λυκοθανάση. Το θέμα της εργασίας είναι:

«Σχεδιασμός και υλοποίηση προηγμένων μαθηματικών μεθόδων για την επίλυση προβλημάτων πολλαπλών πεδίων σε σύγχρονες υπολογιστικές αρχιτεκτονικές».
5. Είναι υπό συγγραφή 1 επιστημονικό άρθρο για υποβολή σε επιστημονικό περιοδικό. Θα αφορά την παραλληλοποίηση των 2 μεθόδων της Interface Relaxation (GEO, ROB) στην Πλατφόρμα FEniCS.
6. Ο κώδικας των παράλληλων υλοποιήσεων των μεθόδων Interface Relaxation και Monte Carlo.

5. Συνεργασίες

Κατά τη διάρκεια του έργου η ΚΕΟ 3, ανέπτυξε τις παρακάτω συνεργασίες:

1. Με την ΚΕΟ 2, σε θέματα υλοποίησης της μεθόδου Interface Relaxation στην πλατφόρμα FEniCS (με την Αν. Καθηγήτρια Γ. Τσομπανοπούλου) και της μεθόδου Monte Carlo σε pThreads (με τον Καθηγητή Μ. Βάβαλη και τον υποψήφιο διδάκτορα Μ. Μαρούδα).
2. Με την ΚΕΟ 1, σε θέματα παράλληλης υλοποίησης της μεθόδου ALOPEX, για την επίλυση του προβλήματος Περιβαλλοντικής Μηχανικής (πρόβλημα υφαλμύρισης).

6. Μελλοντικές Δράσεις

Παρόλο που το έργο ολοκληρώθηκε η ΚΕΟ3 θα συνεχίσει την συνεργασία της με τις άλλες δύο ΚΕΟ. Εκτός από τις δράσεις που αναφέρθηκαν στα προηγούμενα (όπως παραλληλοποίηση και άλλων IR μεθόδων και συνδυασμός της μεθόδου Alopex, με IR μεθόδους), στα πλαίσια του έργου είναι σε εξέλιξη η διδακτορική διατριβή της εξωτερικής συνεργάτιδας κας Αίγλης Κορφιάτη. Η κα Κορφιάτη είναι υποψήφια διδάκτωρ στο Μ.Δ.Ε. “Επιστήμη και Τεχνολογία Υπολογιστών”, του Τμήματος Μηχανικών Η/Υ και Πληροφορικής του Πανεπιστημίου Πατρών. Η διατριβή βρίσκεται στο 2^ο έτος, υπό την επίβλεψη του Επιστημονικού Υπεύθυνου της ΚΕΟ 3 καθηγητή Σ. Λυκοθανάση και έχει θέμα:

«Σχεδιασμός και υλοποίηση ολοκληρωμένου περιβάλλοντος επίλυσης προβλημάτων σε αρχιτεκτονικές υψηλής επίδοσης και εφαρμογές»

7. Βιβλιογραφία

Κεφάλαιο 2.1- 2.2:

[1] C. Kessler and J. Keller, “Models for Parallel Computing: Review and Perspectives”, PARS – Mitteilungen 24:Dec.2007, ISSN 0177-0454, GI/ITG PARS.

[2] M. Baker, A. Apon, R. Bayyaa nd H. Jin “ Cluster Computing and Applications”, 2000, <http://www.buuya.com/papers/encyclopedia.pdf>.

[3] <http://searchdatacenter.techtarget.com/definition/server-blade>.

[4] <http://searchitchannel.techtarget.com/feature/Using-blade-servers-for-virtualization-clustering>.

[5] <http://fenicsproject.org/about/>.

Κεφάλαιο 2.3:

[1] D. Keyes and W. Gropp. A comparison of domain decomposition techniques for elliptic partial differential equations and their parallel implementation. SIAM J. Sci. Stat. Comput., 8:s166-s202, 1987.

[2] Tony F. Chan and Tarek P. Mathew. Domain decomposition algorithms. In Acta Numerica 1994, pages 61-43. Cambridge University Press, 1994.

- [3] T. F. Chan and D. C. Reasco. A survey of preconditioners for domain decomposition. Technical Report /DCS/RR-414, Yale University, 1985.
- [4] P.E. Bjorstad, O. Widlund. To overlap or not overlap: A note on a domain decomposition method for elliptic problems, SIAM J. Sci. Statist Comput. 10 (2) (1989) 1053–1061.
- [5] T.F. Chan, D. Goovaerts. On the relationship between overlapping and nonoverlapping domain decomposition methods, SIAM J. Matrix Anal. Appl. 13 (1992) 663–670.
- [6] T. F. Chan, T. Y. Hou, and P. L. Lions. Geometry related convergence results for domain decomposition algorithms. SIAM J. Numer. Anal., 28(2):378-391, 1991.
- [7] J.R. Rice, An agent-based architecture for solving partial differential equations, SIAM News 31 (6) (1998).
- [8] A. Bamberger, R. Glowinski, and Q. H. Tran. A domain decomposition method for the acoustic wave equation with discontinuous coefficients and grid change. SIAM J. Numer. Anal., 34(2):603-639, 1997.
- [9] B. Despres. Domain decomposition method and the Helmholtz problem. In L. Halpern, G. Cohen and P. Joly, editors, Mathematical and Numerical Aspects of Wave Propagation Phenomena, pages 44-52. SIAM, 1991.
- [10] S. Kim. A parallelizable iterative procedure for the Helmholtz problem. Appl. Numer. Math., 17:411-425, 1995.
- [11] J. R. Rice, P. Tsompanopoulou, E. Vavalis, and D. Yang. Domain decomposition methods for underwater acoustics problems. Technical report, Computer Science Department, Purdue University, W. Lafayette.
- [12] D. Yang. A non-overlapping domain decomposition method for elliptic interface problems. Technical Report # 1472, University of Minnesota, 1997.
- [13] Drashansky T.: An Agent-Based Approach to Building Multidisciplinary Problem Solving Environments. PhD Thesis, Purdue University, Computer Science Department, (1996).
- [14] J. Xu and J. Zou. Non-overlapping domain decomposition methods. Technical report, Mathematics Department, Pennsylvania State University, University Park, PA, 1996.
- [15] M. Mu, J.R. Rice, Modeling with collaborating PDE solvers-theory and practice, Computing Systems in Engineering 6 (1995) pp. 87-95.
- [16] Αίγλη Κορφιάτη, «Σχεδιασμός και υλοποίηση προηγμένων μαθηματικών μεθόδων για την επίλυση προβλημάτων πολλαπλών πεδίων σε σύγχρονες υπολογιστικές αρχιτεκτονικές», Μ.Δ.Ε., Μ.Π.Σ. “Επιστήμη και Τεχνολογία Υπολογιστών”, τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πάτρα, 2014.
- [17] J.R. Rice, E. Vavalis, and D. Yang. Analysis of a non-overlapping domain decomposition method for elliptic PDEs. J. Comput. Appl. Math., 87:11-19, 1997.
- [18] Tsompanopoulou, P., Vavalis, E.: Analysis of an interface relaxation method for composite elliptic differential equations. Journal of Computational and Applied Mathematics 226 2, 370–387 (2009).
- [19] Logg, A., Mardal, K. A., Wells, G. N. et al.: Automated Solution of Differential Equations by the Finite Element Method. Springer, (2012).
- [20] Rabbitmq, 2014. Online. Available: www.rabbitmq.com/documentation.html(accessed July 11, 2014).

- [21] Rice, J. R., Tsompanopoulou, P., Vavalis, E.: Interface relaxation methods for elliptic differential equations. *Applied Numerical Mathematics* 32 2, 219–245 (2000).

Κεφάλαιο 2.4

- [1] K. P. Esler, Jeongnim Kim, L. Shulenburg, and D.M. Ceperley “Fully accelerating quantum Monte Carlo simulations of real materials on GPU clusters”.
- [2] John Strikwerda, “Finite Difference Schemes and Partial Differential Equations”, SIAM, 2nd edition, 2004.
- [3] Norma Alias, “Some Parallel Numerical Methods in Solving Partial Differential Equations”, submitted 2nd International Conference on Computer Engineering and Technology, 2010.
- [4] G. Sarailidis and M. Vavalis, “Hybrid Solvers for Elliptic PDEs on rectangular multi-domains in 2 and 3 dimensions”, submitted to *Scientific Computing*, 2013.
- [5] J.M Delaurentis, L.A Romero, ”A Monte Carlo method for poisson’s equation”, *Journal of Computational Physics*, Volume 90, Issue 1”, September 1990, Pages 123-140, ISSN 0021-9991, [http://dx.doi.org/10.1016/0021-9991\(90\)90199-B](http://dx.doi.org/10.1016/0021-9991(90)90199-B).
- [6] Edgar Gabriel et al. “Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation”, In *Proceedings, 11th European PVM/MPI Users’ Group Meeting*, Budapest, Hungary, September 2004.
- [7] SPLINE, a C++ library which constructs and evaluates spline functions. http://people.sc.fsu.edu/~jburkardt/cpp_src/spline/spline.html
- [8] MBA - Multilevel B-Spline Approximation Library: <http://www.sintef.no/Projectweb/Geometry-Toolkits/MBA/>
- [9] deal.II - an open source finite element library <http://www.dealii.org/>
- [10] mpiP Profiler - a lightweight profiling library for MPI applications. <http://mpip.sourceforge.net/>
- [11] Fenics - The FEniCS Project is a collection of free software with an extensive list of features for automated, efficient solution of differential equations. <http://fenicsproject.org/>

Κεφάλαιο 2.5

Όπως στο κεφάλαιο 2.3.

Κεφάλαιο 2.6

- [1] Chan, T.F., Mathew, T.P., “Domain decomposition algorithms”, *Acta Numerica* 1994, Cambridge University Press, Cambridge, 61-143 (1994)
- [2] Keyes, D., Gropp, W., “A comparison of domain decomposition techniques for elliptic partial differential equations and their parallel implementation”, *SIAM J. Sci. Statist. Comput.* 8 s166-s202 (1987)
- [3] Rice, J. R., Tsompanopoulou, P., Vavalis, E., “Interface relaxation methods for elliptic differential equations”, *Applied Numerical Mathematics* 32 2, 219–245 (2000)

- [4]Rice, J.R., Tsompanopoulou, P. Vavalis, E.A., “Fine Tunning Interface Relaxation Methods for Elliptic Differential Equations”, Applied Numerical Mathematics, 43(4), 459–481 (2002)
- [5]Tsompanopoulou, P., Vavalis, E., “An Experimental Study of Interface Relaxation Methods for Composite Elliptic Differential Equations”, Applied Mathematical Modelling, 32 1620–1641 (2008)
- [6]Tsompanopoulou, P., Vavalis, E., “Analysis of an interface relaxation method for composite elliptic differential equations”, Journal of Computational and Applied Mathematics 226 2, 370– 387 (2009)
- [7]Korfiati, A., Tsompanopoulou, P., Likothanassis, S., “Serial and Parallel Implementation of an Interface Relaxation Method”, Proceedings of the 6th International Conference on Numerical Analysis, pp 167-173 (2014)