



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ &
ΠΛΗΡΟΦΟΡΙΚΗΣ

Διπλωματική εργασία

στο πλαίσιο του μεταπτυχιακού προγράμματος

Επιστήμη και Τεχνολογία Υπολογιστών

**Σχεδιασμός και υλοποίηση
προηγμένων μαθηματικών μεθόδων
για την επίλυση προβλημάτων
πολλαπλών πεδίων σε σύγχρονες
υπολογιστικές αρχιτεκτονικές**

Κορφιάτη Αίγλη

Επιβλέπων

Σπυρίδων Λυκοθανάσης, Καθηγητής

Πάτρα, Οκτώβρης 2014



Η παρούσα έρευνα έχει συγχρηματοδοτηθεί από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο - ΕΚΤ) και από εθνικούς πόρους μέσω του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» του Εθνικού Στρατηγικού Πλαισίου Αναφοράς (ΕΣΠΑ) - Ερευνητικό Χρηματοδοτούμενο Έργο: ΘΑΛΗΣ. Επένδυση στην κοινωνία της γνώσης μέσω του Ευρωπαϊκού Κοινωνικού Ταμείου (MIS 379416).

Πρόλογος

Για την επίλυση προβλημάτων πολλαπλών πεδίων / πολλαπλών φυσικών έχουν προταθεί διάφορες τεχνικές στη βιβλιογραφία. Οι μέθοδοι χαλάρωσης στις διεπαφές είναι μια ενδιαφέρουσα προσέγγιση για την επίλυση τέτοιων προβλημάτων. Επίσης, στις μέρες μας η αρχιτεκτονική των συστοιχιών υπολογιστών (Clusters) αναδείχθηκε και καθιερώθηκε ως η πλέον κατάλληλη πλατφόρμα για την επίλυση υπολογιστικά απαιτητικών προβλημάτων. Ταυτόχρονα, η πρόσφατη ανάπτυξη και προτυποποίηση των διαδικτυακών υπηρεσιών (web services) έχει συνδράμει σημαντικά στην εύχρηστη και ομοιόμορφη διάθεση των υπολογιστικών πόρων που βρίσκονται διαθέσιμοι μέσω των υποδομών των Clouds.

Σκοπός της παρούσας διπλωματικής εργασίας είναι η υλοποίηση της μεθοδολογίας χαλάρωσης στη διεπαφή εντός ενός περιβάλλοντος επίλυσης προβλημάτων πολλαπλών πεδίων / πολλαπλών φυσικών, με παράλληλη αξιοποίηση των σύγχρονων υπολογιστικών αρχιτεκτονικών.

Σε αυτό το σημείο θα πρέπει να αναφερθεί ότι η παρούσα ερευνητική εργασία έχει δημοσιευτεί στα πρακτικά του διεθνούς συνεδρίου 6th INTERNATIONAL CONFERENCE ON NUMERICAL ANALYSIS NumAn 2014 [101].

Η παρούσα διπλωματική εργασία συγχρηματοδοτήθηκε από το πρόγραμμα ΘΑΛΗΣ.

Ευχαριστίες

Στην επιστημονική και ερευνητική δουλειά που έγινε για την ολοκλήρωση της παρούσας διπλωματικής εργασίας συντέλεσε και βοήθησε, επιστημονικά και ηθικά, ένα πλήθος ατόμων που θα ήθελα να ευχαριστήσω, καθώς χωρίς την αρωγή και τη συμπαράστασή τους δεν θα μπορούσα να βγάλω εις πέρας το απαιτητικό έργο για την εκπόνηση αυτής της εργασίας.

Αρχικά, τις θερμότερες ευχαριστίες μου απευθύνω στον καθηγητή μου, Σπύρο Λυκοθανάση, ο οποίος πρώτα μου εμπιστεύθηκε το συγκεκριμένο επιστημονικό θέμα και στη συνέχεια βοήθησε όσο ήταν δυνατό περισσότερο προσφέροντας τεχνογνωσία και πόρους για να μπορέσω να ολοκληρώσω την έρευνά μου. Επίσης είναι άξιο για αυτόν να αναφέρω ότι πάντα ήταν στον πλάι μου κατά τη διάρκεια και αυτής της εργασίας μου και παλαιότερα και πάντα με βοηθούσε και με καθοδηγούσε, ώστε να προσπεράσω τυχόν εμπόδια που εμφανίστηκαν κατά τη διάρκεια του ερευνητικού έργου μου.

Ιδιαίτερη συμβολή κατά την υλοποίηση της συγκεκριμένης διπλωματικής εργασίας είχε και ο καθηγητής Παναγιώτης Αλεφραγκής, ο οποίος με τις γνώσεις του και την καθοδήγησή του με βοήθησε ώστε να επεκτείνω με επιτυχία την έρευνά μου σε επιστημονικά πεδία τα οποία δεν ήταν αρχικά γνωστά σε μένα.

Τις ευχαριστίες μου εκφράζω και στον καθηγητή Γεώργιο Παυλίδη που δέχτηκε να είναι μέλος της τριμελούς επιτροπής αξιολόγησης της μεταπτυχιακής μου εργασίας.

Να σημειώσω σε αυτό το σημείο ότι η βοήθεια της καθηγήτριας Γιώτας Τσομπανοπούλου ήταν καθοριστική για την ολοκλήρωση της παρούσας εργασίας. Αφενός, γιατί η μέθοδος που υλοποιήθηκε είχε αναπτυχθεί από εκείνη. Αφετέρου, για τις αμέτρητες ώρες συζητήσεων και βήμα προς βήμα συνεργασίας σχετικά με την προτεινόμενη υλοποίηση.

Ένα ξεχωριστό ευχαριστώ οφείλω στην καθηγήτρια Σεφερίνα Μαυρουδή για τη στήριξη και την έμπνευση που μου έχει προσφέρει από τα φοιτητικά μου χρόνια.

Επίσης, η παρούσα εργασία δεν θα είχε επιτυχώς ολοκληρωθεί χωρίς την βοήθεια των Χρήστου Αλεξάκου, Κωνσταντίνου Θεοφιλάτου και Θωμά Αμοργιανιώτη, οι οποίοι έχουν υπάρξει άριστοι συνεργάτες κατά την διάρκεια των χρόνων που είμαι μέλος του Εργαστηρίου Αναγνώρισης Προτύπων.

Τέλος, θέλω να ευχαριστήσω τους γονείς μου Μαρίνα και Αντώνη, τον αδελφό μου Μιχαήλ-Άγγελο και τον φίλο μου Κωνσταντίνο Κυριακόπουλο, που με υπομονή και κουράγιο προσέφεραν την απαραίτητη ηθική συμπαράσταση για την ολοκλήρωση της μεταπτυχιακής μου εργασίας.

Περίληψη

Για την επίλυση προβλημάτων πολλαπλών πεδίων / πολλαπλών φυσικών έχουν προταθεί διάφορες τεχνικές στη βιβλιογραφία. Οι μέθοδοι χαλάρωσης στις διεπαφές είναι μια ενδιαφέρουσα προσέγγιση για την επίλυση αυτών των προβλημάτων. Υποθέτοντας κάποιες αρχικές τιμές στις διεπαφές του προβλήματος, οι μέθοδοι χαλάρωσης στις διεπαφές επιλύουν επαναληπτικά τα υποπροβλήματα και χαλαρώνουν τις τιμές στις διεπαφές έως ότου επιτευχθεί σύγκλιση. Τα κυριότερα πλεονεκτήματα των μεθόδων αυτών είναι ότι οι ρυθμοί σύγκλισής τους εξαρτώνται μόνο από τις παραμέτρους των ίδιων των προβλημάτων, τις παραμέτρους που σχετίζονται με την ανάλυση των προβλημάτων σε υποπροβλήματα και τις παραμέτρους που σχετίζονται με τον τελεστή που εφαρμόζεται στις διεπαφές.

Στην παρούσα μεταπτυχιακή εργασία παρουσιάζεται μια νέα υλοποίηση μιας μεθόδου χαλάρωσης στις διεπαφές, της μεθόδου GEO. Η GEO βασίζεται σε έναν απλό μηχανισμό γεωμετρικής διόρθωσης και δρα επαναληπτικά, ώστε να χαλαρώσει τις τιμές στις διεπαφές. Πιο συγκεκριμένα, προσθέτει στις παλιές τιμές των διεπαφών έναν γεωμετρικά σταθμισμένο συνδυασμό των κανονικών παραγώγων των οριακών σημείων των γειτονικών πεδίων.

Η υλοποίηση της μεθόδου GEO γίνεται στην πλατφόρμα FEniCS, η οποία είναι μια συλλογή ελεύθερου λογισμικού για την αυτοματοποιημένη, αποδοτική υλοποίηση διαφορικών εξισώσεων. Η υλοποίηση πραγματοποιείται με τρόπο τέτοιο που ο χρήστης να μπορεί εύκολα να ορίζει τις ιδιότητες των πεδίων του προβλήματος (όπως, γεωμετρία του προβλήματος, τελεστής μερικής διαφορικής εξίσωσης, οριακές συνθήκες, συνθήκες διεπαφών). Στο παρασκήνιο γίνεται η δημιουργία ή / και η βελτιστοποίηση των πλεγμάτων (τριγωνικά στοιχεία) για κάθε πεδίο του προβλήματος, η επίλυση των τοπικών προβλημάτων μερικών διαφορικών εξισώσεων και η εμφάνιση των υπολογισμένων τιμών για το πρόβλημα συνολικά και για τις διεπαφές. Η κυριότερη πρόκληση της υλοποίησης είναι η πρόσβαση στις τιμές των λύσεων στις διεπαφές και η εισαγωγή των χαλαρωμένων τιμών πίσω στα υποπροβλήματα για να αποτελέσουν τις οριακές συνθήκες των υποπροβλημάτων.

Πραγματοποιούνται πειράματα σε προβλήματα-μοντέλα, ώστε να ελεγχθεί η εφαρμοσιμότητα και η σύγκλιση της μεθόδου. Στη συνέχεια, υλοποιείται μια παράλληλη έκδοση της μεθόδου και πραγματοποιούνται συγκριτικά πειράματα απόδοσης. Η παράλληλη υλοποίησή μας αποδεικνύεται πολύ χρήσιμη, ειδικά για μεγάλα προβλήματα.

Abstract

Several approaches have been suggested in the literature for the solution of multiphysics / multidomain problems. Interface Relaxation (IR) methods are an interesting approach for the solution of these problems. Assuming initial guesses on the interfaces of the original problem, IR methods iteratively solve the subproblems and relax for new values on the interfaces until convergence is succeeded. Their main advantages are that their rates of convergence only depend on the parameters of the problem itself, the parameters related to its decomposition into subproblems and the parameters related to the operator imposed on the interfaces.

In the present master thesis a new implementation of an IR method named GEO is presented. GEO is based on a simple geometric correction mechanism and acts iteratively so as to relax the values of the solution on the interfaces. In particular, it adds to the old interface values a geometrically weighted combination of the normal boundary derivatives of the adjacent subdomains.

In this thesis GEO is implemented in FEniCS. The FEniCS project is a collection of free software for automated, efficient solution of differential equations. In order to evaluate the GEO implementation, it is applied on two different PDE problems with the same differential equation and boundary conditions and different domains. FEniCS methods are used to specify the problem's subdomains properties (i.e. geometry, PDE operator and boundary/interface conditions). They are also used to generate and/or refine meshes (triangular elements) for each subdomain, solve the local PDE problems and show the computed results in the global domain and on the interfaces. Getting values of the solutions on the interface (boundaries of the subproblems) and passing the new relaxed values back to the subproblems as updated values for the boundary conditions is the main challenge of the IR methodology implementation.

Experiments are performed for 2-dimensional elliptic partial differential model problems with partitions in multiple subdomains and the results are examined in terms of the method's applicability and convergence. A parallel implementation of the GEO method using FEniCS is also presented, as well as its performance comparison to the serial implementation. The parallel implementation proves to be really useful, especially for large problems.

Περιεχόμενα

Πρόλογος.....	4
Ευχαριστίες.....	6
Περίληψη.....	8
Abstract	10
Περιεχόμενα.....	12
Πίνακας εικόνων	14
1 Προβλήματα πολλαπλών πεδίων / πολλαπλών φυσικών	18
1.1 Εισαγωγή	18
1.2 Υβριδικές / Ασυνεχείς αριθμητικές μέθοδοι Collocation	20
1.3 Στοχαστικές / ντετερμινιστικές υβριδικές μέθοδοι	21
1.4 Μέθοδοι μετασχηματισμού Φωκά	23
1.5 Μέθοδοι χαλάρωσης στις διεπαφές	24
2 Υλοποιήσεις μεθοδολογίας χαλάρωσης στη διεπαφή	26
2.1 Εισαγωγή	26
2.2 SciAgents Framework	26
2.3 Bond Framework	28
2.4 GasTurbnLab.....	29
2.5 IRTool: Εργαλειοθήκη MATLAB	31
3 Μέθοδοι χαλάρωσης στη διεπαφή	34
3.1 Εισαγωγή	34
3.2 Διακριτοποίηση πεδίου με επαναληπτική χαλάρωση στις διεπαφές	35
3.3 Μεθοδολογίες χαλάρωσης στη διεπαφή.....	38
3.4 Επίλογος	45
4 Σύγχρονες υπολογιστικές αρχιτεκτονικές	48
4.1 Εισαγωγή	48
4.2 Clusters	49
4.2.1 Εφαρμογές των clusters	49
4.2.2 Ορισμός των clusters.....	49
4.2.3 Πλεονεκτήματα και μειονεκτήματα των clusters	51
4.2.4 Cluster of blades.....	52
4.3 Grids	52
4.4 Cloud computing	53
5 Υλοποίηση	56

5.1 Εισαγωγή	56
5.2 FEniCS	57
5.3 RabbiMQ.....	65
5.4 Σειριακή Υλοποίηση	69
5.5 Παράλληλη Υλοποίηση.....	73
6 Πειράματα	78
6.1 Εισαγωγή	78
6.2 Πειράματα	78
7 Συμπεράσματα	88
Βιβλιογραφία	90

Πίνακας εικόνων

Εικόνα 1: Ο μηχανισμός χαλάρωσης στη διεπαφή.....	37
Εικόνα 2: Ο μηχανισμός χαλάρωσης της μεθόδου GEO	40
Εικόνα 3: Η γραφική αναπαράσταση της λύσης u.....	64
Εικόνα 4: Η γραφική αναπαράσταση του πλέγματος.....	64
Εικόνα 5: Περίληψη RPC συστήματος.....	66
Εικόνα 6: Τα πεδία του ομοιόμορφου (αριστερά) και του ανομοιόμορφου (δεξιά) προβλήματος.....	78
Εικόνα 7: Η νόρμα μεγίστου της σχετικής διαφοράς διαδοχικών επαναλήψεων για τη διεπαφή $x_1=1/3$ του ομοιόμορφου προβλήματος για $h=0.1, 0.05, 0.025$	80
Εικόνα 8: 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 $h=0.05$	80
Εικόνα 9: Η ακριβής και οι υπολογισμένες λύσεις – ομοιόμορφο πρόβλημα, περίπτωση Π1, διεπαφή 1 (πάνω) και διεπαφή 2 (κάτω).....	81
Εικόνα 10: Σχετικές διαφορές διαδοχικών επαναλήψεων - ομοιόμορφο πρόβλημα, περίπτωση Π1, διεπαφή 1 (πάνω) και διεπαφή 2 (κάτω)	81
Εικόνα 11: Η ακριβής και οι υπολογισμένες λύσεις – ανομοιόμορφο πρόβλημα, περίπτωση Π1, διεπαφή 1 (πάνω) και διεπαφή 2 (κάτω).	82
Εικόνα 12: Σχετικές διαφορές διαδοχικών επαναλήψεων - ανομοιόμορφο πρόβλημα, περίπτωση Π1, διεπαφή 1 (πάνω) και διεπαφή 2 (κάτω).	82
Εικόνα 13: Η ακριβής και οι υπολογισμένες λύσεις – ομοιόμορφο πρόβλημα, περίπτωση Π2, διεπαφή 1 (πάνω) και διεπαφή 2 (κάτω).....	83
Εικόνα 14: Σχετικές διαφορές διαδοχικών επαναλήψεων - ομοιόμορφο πρόβλημα, περίπτωση Π2, διεπαφή 1 (πάνω) και διεπαφή 2 (κάτω).	83
Εικόνα 15: Η ακριβής και οι υπολογισμένες λύσεις – ανομοιόμορφο πρόβλημα, περίπτωση Π2, διεπαφή 1 (πάνω) και διεπαφή 2 (κάτω).	84
Εικόνα 16: Σχετικές διαφορές διαδοχικών επαναλήψεων - ανομοιόμορφο πρόβλημα, περίπτωση Π2, διεπαφή 1 (πάνω) και διεπαφή 2 (κάτω).	84
Εικόνα 17: Η ακριβής και οι υπολογισμένες λύσεις – ομοιόμορφο πρόβλημα, περίπτωση Π3, διεπαφή 1 (πάνω) και διεπαφή 2 (κάτω).....	85
Εικόνα 18: Σχετικές διαφορές διαδοχικών επαναλήψεων - ομοιόμορφο πρόβλημα, περίπτωση Π3, διεπαφή 1 (πάνω) και διεπαφή 2 (κάτω).	85
Εικόνα 19: Η ακριβής και οι υπολογισμένες λύσεις – ανομοιόμορφο πρόβλημα, περίπτωση Π3, διεπαφή 1 (πάνω) και διεπαφή 2 (κάτω).	86
Εικόνα 20: Σχετικές διαφορές διαδοχικών επαναλήψεων - ανομοιόμορφο πρόβλημα, περίπτωση Π3, διεπαφή 1 (πάνω) και διεπαφή 2 (κάτω)	86

Πίνακας πινάκων

Πίνακας 1: Ιδιότητες των επτά μεθόδων χαλάρωσης στη διεπαφή.....	46
Πίνακας 2: 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.....	79
Πίνακας 3: Χρόνοι συνολικής εκτέλεσης σειριακής και παράλληλης υλοποίησης.....	87

1 Προβλήματα πολλαπλών πεδίων / πολλαπλών φυσικών

1.1 Εισαγωγή

Οι Μερικές Διαφορικές Εξισώσεις (ΜΔΕ) χρησιμοποιούνται ευρύτατα σαν βασικά μαθηματικά μοντέλα για τη μελέτη μιας πληθώρας φυσικών προβλημάτων εδώ και δεκαετίες. Σήμερα, σε πολλά προβλήματα επιστημονικών και τεχνολογικών εφαρμογών απαιτείται η χρήση σύνθετων ΜΔΕ στις οποίες οι κλασικές μέθοδοι επίλυσης δεν είναι ιδιαίτερα αποτελεσματικές.

Στην παρούσα διπλωματική, αναφερόμενοι κυρίως σε προβλήματα εφαρμογών που χαρακτηρίζονται από την ύπαρξη/αλληλεπίδραση ετερογενών υλικών (multiphysics, multidomain), οι ΜΔΕ που χρησιμοποιούνται για την μοντελοποίηση τους είναι ελλειπτικού ή παραβολικού τύπου και χαρακτηρίζονται για παράδειγμα από την ύπαρξη ασυνεχών συντελεστών στα εσωτερικά σύνορα αλληλεπίδρασης των ετερογενών υλικών, δηλαδή στα σημεία διεπαφών (interfaces). Αυτό προφανώς συνεπάγεται την ασυνέχεια της λύσης της ΜΔΕ, ή/και των παραγώγων της. Παραδείγματα τέτοιων προβλημάτων προέρχονται από τις εξής δύο σημαντικής σπουδαιότητας και πολυπλοκότητας εφαρμογές από τους χώρους της Ιατρικής και της Περιβαλλοντικής Μηχανικής:

- Διάδοση πρωτογενών καρκινικών όγκων εγκεφάλου (γλοίωμα/gliomas) και
- Υφαλμύριση υδροφορέων γλυκών υδάτων λόγω υπεράντλησης.

Για την αποτελεσματική επίλυση τέτοιων προβλημάτων απαιτείται είτε η προσαρμογή γνωστών αριθμητικών ή αναλυτικών μεθόδων επίλυσης ΜΔΕ, ώστε να προσεγγίζουν ρεαλιστικά την λύση των αντίστοιχων προβλημάτων, είτε η ανάπτυξη και μελέτη καινοτόμων μεθόδων, οι οποίες να λαμβάνουν υπόψη σε υψηλότερο επίπεδο μοντελοποίησης τις ιδιαιτερότητες και πολυπλοκότητες του προβλήματος. Παράλληλα τα σύγχρονα υπολογιστικά συστήματα δίνουν νέες ευκαιρίες για την επίλυση πολύπλοκων προβλημάτων και απαιτείται ενδελεχής μελέτη για την μέγιστη αξιοποίηση των δυνατοτήτων τους.

Για την επίλυση των ανωτέρω προβλημάτων κάποιος θα έπρεπε να στοχεύει στην

1. ανάπτυξη και μελέτη νέων αναλυτικών, αριθμητικών και υπολογιστικών μεθόδων,
2. προσαρμογή κλασικών αριθμητικών μεθόδων, για την επίλυση των παραπάνω σύνθετων προβλημάτων ΜΔΕ,
3. υλοποίηση των μεθόδων αυτών σε σύγχρονα υπολογιστικά συστήματα,
4. δοκιμή, αξιολόγηση και σύγκριση των μεθόδων και του αντίστοιχου λογισμικού καθώς και την εφαρμογή και επικύρωσή τους σε προβλήματα-μοντέλα.

Στόχος δηλαδή είναι η δημιουργία ενός νέου επιπέδου συλλογιστικής η οποία θα μας επιτρέψει να αποκτήσουμε μια βαθύτερη διαίσθηση και κατανόηση των γενικών χαρακτηριστικών και του μηχανισμού συνέργειας των εμπλεκόμενων φυσικών νόμων και διαδικασιών.

Φυσικά προβλήματα, όπως η συμπεριφορά καρκινικών εγκεφαλικών όγκων (με την παρουσία ή μη θεραπευτικών μεθόδων) και η μόλυνση υπόγειων ταμειωτήρων γλυκών υδάτων με θαλάσσιο νερό, αποτελούν ευαίσθητα θέματα που απασχολούν ολόένα και περισσότερο την σύγχρονη κοινωνία. Ως εκ τούτου, η μελέτη για την αντιμετώπισή τους συνιστά σοβαρή πρόκληση για ολόκληρη την επιστημονική κοινότητα. Υπό το πρίσμα αυτό γίνεται άμεσα αντιληπτή η σημασία της ακρίβειας των υπολογιζόμενων μεγεθών αλλά και η ταχύτητα υπολογισμού τους.

Μεγάλης κλίμακας και πολυπλοκότητας προσομοιώσεις, όπως αυτές που απαιτούνται στα προαναφερόμενα προβλήματα, απαιτούν την συνεργασία και ενοποίηση φυσικών εννοιών, μαθηματικών μεθόδων επίλυσης, μεθόδων διακριτοποίησης, μεθόδων επίλυσης αλγεβρικών συστημάτων, λογισμικού και υπολογιστικών αρχιτεκτονικών. Η δημιουργία μίας τέτοιας πλατφόρμας μελέτης σύνθετων (multiphysics, multidomain) προβλημάτων αποτελεί την βασική μας επιδίωξη. Απαραίτητη προϋπόθεση για τη δημιουργία της πλατφόρμας αυτής αποτελεί η συντονισμένη διεπιστημονική ερευνητική δραστηριότητα με στόχο την ανάπτυξη υψηλής ακριβείας, πιστότητας και αποδοτικότητας μεθόδων επίλυσης των εν λόγω προβλημάτων, την υλοποίησή τους σε μορφή λογισμικού και την εφαρμογή τους σε σύγχρονα υπολογιστικά συστήματα.

Επί πολλές δεκαετίες η αντιμετώπιση των προαναφερόμενων σύνθετων προβλημάτων βασιζόταν σε μοντέλα και λογισμικά συστήματα με χαρακτηριστικά μονομέρειας ενώ οι εν γένει πρακτικές και πλατφόρμες ελάχιστα διέφεραν από αυτές που αφορούσαν συμβατικά προβλήματα (ένας διαφορικός τελεστής σε ένα ομοιογενές χωρίο). Αποτελεί πεποίθησή μας ότι τα ως άνω σύνθετα προβλήματα θα πρέπει να αντιμετωπιστούν με διαφορετικό τρόπο ο οποίος θα εγγυηθεί την:

- Ευκολότερη διαδικασία της μαθηματικής μοντελοποίησης τους.
- Αποτελεσματικότερη ανάπτυξη του λογισμικού με αντίστοιχη επαναχρησιμοποίηση ενδεχόμενου υπάρχοντος λογισμικού στο μέγιστο δυνατό βαθμό.
- Ευελιξία επιλογής του συνεπέστερου μαθηματικού μοντέλου, του καταλληλότερου σχήματος διακριτοποίησης των επιμέρους ΜΔΕ και της αποδοτικότερης μεθόδου επίλυσης του προκύπτοντος συστήματος γραμμικών αλγεβρικών εξισώσεων.
- Πλήρη αξιοποίηση σύγχρονων αλλά και αναδυόμενων υπολογιστικών αρχιτεκτονικών.

Απώτερος φυσικά στόχος μας είναι η δημιουργία ενός γενικότερου υπόβαθρου για την μελέτη σύνθετων προβλημάτων σημαντικής πολυπλοκότητας τόσο από την άποψη της φυσικής και της γεωμετρίας των προβλημάτων όσο και από την άποψη του λογισμικού και της απαιτούμενης υπολογιστικής ισχύος.

Στη βιβλιογραφία συναντά κανείς τις ακόλουθες αριθμητικές και αναλυτικές μεθόδους για ασυνεχή προβλήματα πολλαπλών πεδίων: Υβριδικές / Ασυνεχείς αριθμητικές μέθοδοι Collocation, Στοχαστικές / ντετερμινιστικές υβριδικές μέθοδοι, Μέθοδοι μετασχηματισμού-Φωκά και Μέθοδοι χαλάρωσης στις διεπαφές στις οποίες γίνεται σύντομη αναφορά στο παρόν κεφάλαιο.

1.2 Υβριδικές / Ασυνεχείς αριθμητικές μέθοδοι Collocation

Η αριθμητική επίλυση ΜΔΕ περιλαμβάνει δύο βασικά βήματα: την διακριτοποίηση της ΜΔΕ και την επίλυση του αναδυόμενου συστήματος αλγεβρικών εξισώσεων. Δύο βασικές κατηγορίες μεθόδων διακριτοποίησης για ΜΔΕ είναι οι μέθοδοι πεπερασμένων διαφορών (ΜΠΔ) και οι μέθοδοι πεπερασμένων στοιχείων (ΜΠΣ).

Η μέθοδος collocation ανήκει στην κατηγορία ΜΠΣ αλλά έχει κάποια χαρακτηριστικά παρόμοια με αυτά των ΜΠΔ. Εν συγκρίσει με τις ΜΠΔ έχει, όπως κάθε ΜΠΣ, το πλεονέκτημα ότι οι τιμές της προσεγγιστικής λύσης και των παραγώγων της υπολογίζονται εύκολα σε οποιοδήποτε σημείο του πεδίου του προβλήματος. Εν συγκρίσει με τις ΜΠΣ έχει το πλεονέκτημα ότι για την εφαρμογή της δεν χρειάζεται υπολογισμός ολοκληρωμάτων, και απαιτείται ο υπολογισμός μόνο μιας τιμής των συναρτήσεων συντελεστών ανά δεδομένο σημείο.

Δύο είναι οι βασικές κατηγορίες collocation: η ορθογώνια collocation με τμηματικά πολυώνυμα [1] και η collocation με τμηματικά πολυώνυμα μεγίστου βαθμού συνέχειας, δηλ. splines. Η ορθογώνια collocation χρησιμοποιεί τμηματικά πολυώνυμα συνέχειας C^{m-1} και βαθμού $m+k-1$, για προβλήματα m τάξεως, και k σημεία ανά υποδιάστημα (k^d σε d διαστάσεις), καταλήγοντας σε προσέγγιση τάξεως $m+k$ (σε ορισμένα σημεία η τάξη ανέρχεται σε $2k$). Αυτή η μέθοδος έχει μελετηθεί σε βάθος για προβλήματα με συνεχείς συντελεστές. Εφαρμόζεται ευρέως σε προβλήματα οποιασδήποτε τάξεως με συνεχείς συντελεστές, με τμηματικά πολυώνυμα οποιουδήποτε βαθμού, και με ανομοιόμορφα και προσαρμοστικά πλέγματα. Η collocation με splines έχει αναπτυχθεί για περιορισμένους βαθμούς splines (π.χ. 2, 3, 4 και 5), και περιορισμένη τάξη προβλημάτων συνεχών συντελεστών (π.χ. 2 και 4) [2-9]. Επίσης η εφαρμογή της ώστε να επιτευχθεί η μέγιστη τάξη προσέγγισης δεν είναι απλή: απαιτεί προσεκτική ανάπτυξη κάποιων τελεστών διατάραξης του προβλήματος. Η δε χρήση ανομοιόμορφων και προσαρμοστικών πλεγμάτων αναπτύχθηκε μόνο πρόσφατα και μόνο για

τετραγωνικές και κυβικές splines [10, 11]. Όμως η collocation με splines έχει κάποια σημαντικά πλεονεκτήματα σε σχέση με την ορθογώνια, μεταξύ των οποίων είναι ότι χρειάζεται μόνο ένα σημείο ανά υποδιάστημα τόσο στην μια διάσταση όσο και σε περισσότερες διαστάσεις, και ότι καταλήγει σε γραμμικό σύστημα μικρότερου μεγέθους, περισσότερο αραιό και με ορισμένες επιθυμητές ιδιότητες, όπως διαγώνια κυρίαρχος πίνακας, κ.α.

Πρόσφατα η collocation με τετραγωνικές splines αναπτύχθηκε για παραβολικά προβλήματα σε μία διάσταση στον χώρο [12]. Η μέθοδος αποδεικνύεται αποδοτική, δεδομένου ότι απαιτεί την επίλυση μόνο ενός τριδιαγώνιου γραμμικού συστήματος ανά χρονικό βήμα. Η μέθοδος εφαρμόστηκε μάλιστα σε προβλήματα υπολογισμού τιμής παραγώγων του χρηματιστηρίου, όπως American options, σε συνδυασμό με προσαρμοστικές μεθόδους πλέγματος και αποδείχθηκε ιδιαίτερα επιτυχής.

Επιπλέον επισημαίνουμε ότι για τα γραμμικά συστήματα που προέρχονται από διακριτοποίηση ΜΔΕ με ορθογώνια ή spline collocation έχουν αναπτυχθεί ταχείς μέθοδοι επίλυσης από τις ερευνητικές ομάδες που συμμετέχουν στην παρούσα πρόταση, όπως για παράδειγμα βέλτιστες μέθοδοι χαλάρωσης, προρυθμισμένες επαναληπτικές μέθοδοι, πολυπλεγματικές μέθοδοι, FFT μέθοδοι [13-17] και έχουν υλοποιηθεί οι αντίστοιχοι αλγόριθμοι σε σύγχρονες υπολογιστικές αρχιτεκτονικές (κοινής και διανεμημένης μνήμης παράλληλα υπολογιστικά συστήματα, Clusters, VLSI, κ.λ.π. [18-22].

Όμως, παρά του ότι έχει γίνει ενδελεχής μελέτη της μεθόδου collocation σε προβλήματα με συνεχείς συντελεστές, το αντίστοιχο δεν είναι καθόλου αληθές για προβλήματα με ασυνεχείς συντελεστές ή προβλήματα με απότομη αλλαγή συμπεριφοράς της λύσης σε ορισμένα τμήματα του πεδίου.

1.3 Στοχαστικές / ντετερμινιστικές υβριδικές μέθοδοι

Οι μέθοδοι Monte Carlo έχουν την ικανότητα, εκτελώντας απλές διαδικασίες στατιστικής δειγματοληψίας, να μας δίνουν προσεγγιστικές λύσεις σε μια πληθώρα μαθηματικών προβλημάτων ποικίλης μορφής, χαρακτηριστικών και ιδιοτήτων.

Έχει ήδη περάσει ένας αιώνας από την ανακάλυψη μεθόδων οι οποίες βασιζόμενες σε έννοιες και πρακτικές Monte Carlo υπολογίζουν προσεγγιστικές λύσεις προβλημάτων στα οποία εμπλέκονται Διαφορικές Εξισώσεις με Μερικές Παραγώγους ή Μερικές Διαφορικές Εξισώσεις (ΜΔΕ). Οι εν λόγω μέθοδοι παράγουν ακολουθίες τυχαίων αριθμών και παρατηρώντας συγκεκριμένα από τα χαρακτηριστικά και τις συμπεριφορές τους έχουν την δυνατότητα να υπολογίσουν προσεγγίσεις λύσεων των ΔΕΜΠ.

Συγκεκριμένα, ο Rayleigh [23] ήταν ο πρώτος ο οποίος μελέτησε την σχέση μεταξύ στοχαστικών διαδικασιών και παραβολικών διαφορικών εξισώσεων, ακολουθούμενος από τον Courant [24] ο οποίος ήταν ο πρώτος που πρότεινε στοχαστικές αριθμητικές διαδικασίες για την λύση ΔΕΜΠ ελλειπτικού τύπου, ενώ οι Metropolis και Ulman [25] ήταν οι πρώτοι που συνέδεσαν την στοχαστικού τύπου αυτή προσέγγιση με ένα όνομα το οποίο παραπέμπει στις εγκαταστάσεις τυχερών παιχνιδιών που υπάρχουν στην πόλη του Monte Carlo και το οποίο πρότεινε σαν έναν γενικό όρο για αριθμητικές μεθόδους που χρησιμοποιούν δειγματοληψίες τυχαίων αριθμών.

Από τότε οι μέθοδοι Monte Carlo χρησιμοποιούνται συχνότατα και σε ευρύτατη από κάθε άποψη κλίμακα. Χαρακτηριστικά αναφέρουμε ότι σύμφωνα με το U.S.A. Department of Energy οι προσομοιώσεις Monte Carlo έχουν συστηματικά καταναλώσει την μισή από την συνολική υπολογιστική δύναμη όλων των υπερυπολογιστών που διέθετε ή διαθέτει. Χρησιμοποιούνται προφανώς για την λύση μιας πληθώρας σημαντικών προβλημάτων αλλά όχι τόσο συχνά για εφαρμογές που εμπλέκουν γραμμικές ΔΕΜΠ. Θεωρούνται δε γενικώς σαν μέθοδοι ανάγκης (last resort) ιδανικά κατάλληλες μόνον για προβλήματα σε πολλές διαστάσεις ή σε χωρία με περίπλοκη γεωμετρία [26]. Ενδεδειγμένη είναι στο σημείο αυτό η εξής ρήση του πρωτοπόρου των μεθόδων Monte Carlo Mark Kac's «You use Monte Carlo methods until you understand the problem» η οποία παρόλο που διατυπώθηκε πριν πολλές δεκαετίες περιγράφει την κυρίαρχη θεώρηση πολλών ερευνητών για τις μεθόδους Monte Carlo ακόμα και σήμερα.

Προβλήματα που εμπλέκουν ΔΕΜΠ έχουν συσχετιστεί με την έννοια Monte Carlo με πολλούς τρόπους. Το [27] περιέχει μια σχετική πρόσφατη ανασκόπηση. Για παράδειγμα, η γνωστή εξίσωση Feynman-Kac καθιέρωσε έναν ενδιαφέροντα σύνδεσμο μεταξύ ΔΕΜΠ και στοχαστικών διαδικασιών. Οι μέθοδοι Monte Carlo υπήρξαν στο παρελθόν, και σε μεγάλο βαθμό παραμένουν, η μόνη υπολογιστική επιλογή για αρκετά μη-γραμμικά προβλήματα ενώ παράλληλα αναγνωρίζονται σαν μια καλή επιλογή για πολλά άλλα δύσκολα, από την άποψη των υπολογισμών, μη-γραμμικά προβλήματα. Επιπρόσθετα, φαίνεται να αποτελούν μια λογικά προφανή επιλογή για κάθε διαφορική εξίσωση της οποίας ένας ή περισσότεροι όροι είναι στοχαστικές διαδικασίες, όποτε και η συνεπαγόμενη λύση είναι και αυτή με την σειρά της μια στοχαστική διαδικασία. Το γεγονός αυτό συνάδει ξεκάθαρα με την πληθώρα των εξαιρετικών πολύ πρόσφατων ερευνητικών προσπαθειών που βασίζονται αποκλειστικά σε μεθόδους Monte Carlo για την αριθμητική επίλυση τέτοιων εξισώσεων που είναι γνωστές σαν στοχαστικές διαφορικές εξισώσεις. Τυπικά παραδείγματα τέτοιων προσπαθειών είναι τα [28, 29] για προβλήματα εξαρτώμενα από τον χρόνο και τα [28, 30-32] για ελλειπτικά προβλήματα.

Όπως έχει ήδη επισημανθεί, ακόμα και οι γραμμικές ΔΕΜΠ είναι στενά συνδεδεμένες με την στοχαστικότητα. Για παράδειγμα, είναι γεγονός ότι η διάχυση είναι στην ουσία μια μορφή κίνησης Brown σε μικροσκοπική κλίμακα. Υπήρχε λοιπόν ουσιαστικό κίνητρο το οποίο οδήγησε ερευνητές σε διαδοχικές προσπάθειες που αφορούσαν την ανάπτυξη και την διάδοση αριθμητικών επιλυτών τύπου Monte Carlo για ΔΕΜΠ εξαρτώμενες από τον χρόνο [27, 33-36]. Διάφορα γραμμικά, μη-στοχαστικά ελλειπτικά προβλήματα συννοριακών τιμών είναι επίσης στενά με την γενικότερη πιθανοθεωρητική περιοχή [24]. Τέλος αξίζει να σημειωθεί πως υπάρχουν αρκετές πρόσφατες ερευνητικές προσπάθειες που αφορούν πιθανοθεωρητικές ερμηνείες της αρμονικότητας καθώς και θεμελιωδών ΔΕΜΠ ελλειπτικού τύπου χρησιμοποιώντας κίνηση Brown και στοχαστικό λογισμό (calculus) [37].

Αξίζει τέλος να τονίσουμε ότι παρ' όλες τις σχετικά σημαντικές παλαιότερες αλλά και πρόσφατες σχετικές ερευνητικές προσπάθειες, οι συγκεκριμένες μέθοδοι δεν έχουν ακόμα ελκύσει την αναμενόμενη γενικότερη προσοχή. Σημαντικό είναι να σημειώσουμε το γεγονός ότι ελάχιστα σχετικά λογισμικά συστήματα υπάρχουν διαθέσιμα. Η αναζήτηση για παράδειγμα στην βιβλιογραφία του ACM Transaction on Mathematical Software με λέξη κλειδί τον όρο «Monte Carlo» μας επέστρεψε μόνον 10 σχετικές αναφορές.

1.4 Μέθοδοι μετασχηματισμού Φωκά

Πρόσφατα, μία νέα καινοτόμος αναλυτική μέθοδος μετασχηματισμού για την λύση γραμμικών και ολοκληρώσιμων μη-γραμμικών ΜΔΕ εισήχθη [38, 39] από τον Αθανάσιο Φωκά. Η μέθοδος αυτή, η οποία χαρακτηρίστηκε από την London Mathematical Society ως «pioneering work» κατά την απονομή του βραβείου Naylor για το 2000 στον Καθ. Α. Φωκά, για γραμμικές ΜΔΕ μπορεί να θεωρηθεί ως το ανάλογο της προσέγγισης του Green στο φασματικό (Fourier) επίπεδο και περιλαμβάνει τα εξής βήματα:

- Κατασκευή ενός ζεύγους γραμμικών εξισώσεων ιδιοτιμών (σε αναλογία με το ζευγάρι Lax για μη-γραμμικές ολοκληρώσιμες ΜΔΕ) συμβατών με την ΜΔΕ,
- Εκτέλεση φασματικής ανάλυσης στο παραπάνω ζεύγος και αφενός μεν έκφραση της λύσης της ΜΔΕ με μία παράσταση ολοκληρώματος μίας φασματικής συνάρτησης, αφετέρου δε έκφραση της φασματικής συνάρτησης με μία παράσταση ολοκληρώματος γνωστών και άγνωστων ποσοτήτων της λύσης της ΜΔΕ και των παραγώγων της, στο σύνορο του πεδίου, και

- Καθορισμός της φασματικής συνάρτησης με χρήση συνοριακών συνθηκών της λύσης της ΜΔΕ ή/και των παραγώγων της.

Επισημαίνουμε ότι ο μόνος περιορισμός που τίθεται στην όλη διαδικασία είναι αυτός της ύπαρξης λύσης της ΜΔΕ ενώ η διαδικασία είναι ανεξάρτητη της μορφής των συνοριακών δεδομένων.

Η έκφραση της φασματικής συνάρτησης στο δεύτερο στάδιο της παραπάνω διαδικασίας καλείται γενικευμένη συνθήκη και αποτελεί κλειδί της μεθόδου Φωκά. Ο καθορισμός της στο τρίτο στάδιο επιτυγχάνεται με την επίλυση μιας γενικευμένης Dirichlet-Neumann (D-N) απεικόνισης. Για μία μεγάλη κλάση συνοριακών προβλημάτων η γενικευμένη D-N απεικόνιση μπορεί να επιλυθεί σε κλειστή μορφή. Για παράδειγμα αυτό έχει επιτευχθεί για εξελικτικές γραμμικές ΜΔΕ με χωρικές παράγωγες αυθαίρετης τάξης [39, 40], για τις ΜΔΕ του Laplace, της τροποποιημένης Helmholtz και την Διαρμονική σε απλά πολυγωνικά πεδία [41-44] καθώς και για τις βασικές ολοκληρώσιμες μη-γραμμικές εξελικτικές ΜΔΕ για απλές συνοριακές συνθήκες [45, 46]. Για γενικά συνοριακά προβλήματα, για τα οποία δεν μπορούν να παραχθούν λύσεις κλειστής μορφής, έχει αναπτυχθεί με επιτυχία μία αριθμητική προσεγγιστική μέθοδος για ελλειπτικής μορφής ΜΔΕ σε γενικά κυρτά πολυγωνικά πεδία [47, 48] η οποία βασίζεται στην εφαρμογή μιας φασματικής collocation μεθόδου στο σύνορο του πεδίου. Η μέθοδος χρησιμοποιήθηκε πρόσφατα με επιτυχία για την μελέτη συνοριακών προβλημάτων τύπου Helmholtz [49].

1.5 Μέθοδοι χαλάρωσης στις διεπαφές

Για την προσομοίωση και επίλυση μεγάλων και σύνθετων επιστημονικών προβλημάτων χρησιμοποιούνται τρεις διαφορετικές μεθοδολογίες: η μέθοδος διαχωρισμού χωρίων, η μέθοδος διαχωρισμού Schwarz και η μέθοδος χαλάρωσης στη διεπαφή.

Η πρώτη μέθοδος [50-55] δημιουργεί ένα πλέγμα διακριτοποίησης του γεωμετρικού χωρίου του αρχικού προβλήματος μετατρέποντάς το από συνεχές σε διακριτό πρόβλημα. Το πλέγμα αυτό χωρίζεται σε μικρότερα πλέγματα, με τα στοιχεία τους να αποτελούν μικρότερα διακριτά υποπροβλήματα τα οποία είναι ισχυρά συνδεδεμένα μεταξύ τους καθώς τα μαθηματικά μοντέλα της κοινής διεπιφάνειας εμπλέκουν πληροφορία και από τα δύο υποχωρία.

Η δεύτερη μέθοδος [56, 57] χωρίζει το γεωμετρικό χωρίο σε μικρότερα υποχωρία με μικρή επικάλυψη, χρησιμοποιεί ανεξάρτητα μοντέλα για τα επιμέρους προβλήματα και υπολογίζει την λύση του αρχικού προβλήματος με επαναληπτική διαδικασία. Ωστόσο στη συγκεκριμένη μεθοδολογία, η επικάλυψη των υποχωρίων προσθέτει στην πολυπλοκότητα του προβλήματος.

Η τρίτη και νεότερη μέθοδος [58-61] χωρίζει το αρχικό χωρίο σε υποχωρία, χρησιμοποιεί ανεξάρτητα μαθηματικά μοντέλα διακριτοποίησης σε κάθε επιμέρους χωρίο και μέσα από μια επαναληπτική διαδικασία επιλύει τα υποπροβλήματα με σκοπό τον υπολογισμό της λύσης του αρχικού προβλήματος. Κατά μήκος της διεπαφής μεταξύ δύο γειτονικών υποχωρίων και με βάση τη φυσική του προβλήματος, ορίζονται οι συνθήκες που πρέπει να ικανοποιούνται, π.χ., συνέχεια της μάζας, της θερμότητας, διατήρηση της ορμής, κλπ. Σε κάθε βήμα της μεθόδου, επιλύονται τα επιμέρους προβλήματα ενώ το σφάλμα πάνω στη διεπαφή ελαττώνεται με διάφορες τεχνικές εξομάλυνσης με σκοπό να ικανοποιηθούν οι κατάλληλες συνθήκες και να υπολογισθεί η ολική λύση.

Τα συγκεκριμένα χαρακτηριστικά κάνουν τη μέθοδο ελκυστική καθώς απλοποιούν και αντικαθιστούν το αρχικό μεγάλο και δυσεπίλυτο πρόβλημα, με πολλά μικρά απλούστερα προβλήματα, τα οποία είναι χαλαρά συνδεδεμένα μεταξύ τους. Στις μεθόδους χαλάρωσης στη διεπαφή μπορούν να χρησιμοποιηθούν πολλά και διαφορετικά φυσικά και μαθηματικά μοντέλα για την καλύτερη προσομοίωση και επίλυση των σύνθετων πραγματικών προβλημάτων. Αυτό καθιστά δυνατή αλλά και αναγκαία τη χρήση διαφορετικού λογισμικού για την αντιμετώπιση των επιμέρους προβλημάτων. Το λογισμικό μπορεί να υπάρχει ήδη με τη μορφή πακέτου ή βιβλιοθήκης και να έχει υλοποιηθεί και ελεγχθεί για την ορθότητα και την αποδοτικότητα του. Επίσης είναι δυνατός ο πειραματισμός με νέα μοντέλα και μαθηματικές μεθόδους και κατά συνέπεια είναι δυνατή και η υλοποίηση και ο έλεγχος νέων μεθόδων. Η σύζευξη των επιμέρους γειτονικών προβλημάτων είναι χαλαρή καθώς οι τεχνικές χαλάρωσης καθορίζουν τιμές πάνω στις διεπαφές και επιτρέπουν την λύση σε κάθε βήμα της μεθόδου. Το γεγονός ότι τα επιμέρους προβλήματα προσομοιώνονται με ανεξάρτητα μοντέλα, καθιστά κάθε βήμα της επαναληπτικής διαδικασίας πλήρως παραλληλίσμο καθώς τα υποπροβλήματα μπορούν να λυθούν ανεξάρτητα και άρα ταυτόχρονα [62, 63]. Η χρήση παράλληλων υπολογιστών και ισχυρών υπολογιστικών συστημάτων, clusters, Grids και Clouds υλοποιεί πλήρως την παραλληλία και κάνει δυνατή την επίλυση πραγματικών και πολύπλοκων προβλημάτων σε πραγματικό χρόνο.

Η μεθοδολογία χαλάρωσης στη διεπαφή [61] επιτυγχάνει την επίλυση πραγματικών και πολύπλοκων προβλημάτων πολλαπλών πεδίων με γρήγορο και κομψό τρόπο, αξιοποιώντας υπάρχον λογισμικό σε μοντέρνα και ισχυρά υπολογιστικά συστήματα, σε πραγματικό χρόνο.

Στην παρούσα διπλωματική θα επικεντρωθούμε στη μεθοδολογία χαλάρωσης στη διεπαφή για την επίλυση προβλημάτων πολλαπλών πεδίων / πολλαπλών φυσικών. Πιο συγκεκριμένα θα την υλοποιήσουμε εντός ενός περιβάλλοντος επίλυσης τέτοιων προβλημάτων.

2 Υλοποιήσεις μεθοδολογίας χαλάρωσης στη διεπαφή

2.1 Εισαγωγή

Δεδομένου ότι η μεθοδολογία χαλάρωσης στη διεπαφή είναι σχετικά καινούργια, υπάρχει μικρός αριθμός υλοποιήσεων διαθέσιμος στη βιβλιογραφία. Η Matlab έχει χρησιμοποιηθεί για την υλοποίηση εφτά μεθόδων χαλάρωσης στη διεπαφή για μονοδιάστατα προβλήματα [64] και για μία για δισδιάστατα [65], χωρίς όμως αυτές οι προσπάθειες να αποτελούν ένα περιβάλλον επίλυσης προβλημάτων πολλαπλών πεδίων / πολλαπλών φυσικών.

2.2 SciAgents Framework

Μια πρώτη προσπάθεια δημιουργίας ενός ολοκληρωμένου περιβάλλοντος επίλυσης προβλημάτων πολλαπλών πεδίων/πολλαπλών φυσικών αποτελεί το SciAgents Framework [66, 67]. Η συγκεκριμένη υλοποίηση εκμεταλλεύεται τον εγγενή παραλληλισμό της μεθοδολογίας χαλάρωσης στις διεπαφές, χρησιμοποιώντας το υπολογιστικό παράδειγμα των πρακτόρων πάνω σε ένα δίκτυο από ετερογενή κατανεμημένα υπολογιστικά συστήματα. Η υλοποίηση του SciAgents αποτελείται από μία κλάση μεθόδων χαλάρωσης στις διεπαφές σε ένα πλαίσιο βασισμένο σε πράκτορες και είναι γραμμένη κυρίως σε C και JAVA.

Πιο συγκεκριμένα, το SciAgents μετατρέπει το φυσικό πρόβλημα σε ένα δίκτυο τοπικών επιλυτών ΜΔΕ και χαλαρωτών των διεπαφών. Υπάρχουν τρεις τύποι πρακτόρων: ένας PDECoordinator και διάφοροι PDESolver και PDEMediator πράκτορες. Ο PDECoordinator είναι υπεύθυνος για τον έλεγχο ολόκληρης της εφαρμογής και το συντονισμό της όλης διαδικασίας. Ένας PDEMediator δρα ως διαιτητής μεταξύ των δύο επιλυτών που μοιράζονται ένα όριο μεταξύ δύο πεδίων και ένας PDESolver είναι μια μέθοδος περιτυλίγματος μιας εφαρμογής που επιλύει το τοπικό πρόβλημα.

Όταν εκκινείται ο PDECoordinator πράκτορας, διαβάζει ένα αρχείο με την περιγραφή του προβλήματος και γράφει τις πληροφορίες στο μοντέλο του. Το αρχείο εισόδου περιέχει πληροφορίες σχετικά με τον αριθμό των επιλυτών και των μεσαζόντων, τα χαρακτηριστικά των διεπαφών, τις αρχικές μαντεψιές στις διεπαφές, τις μεθόδους χαλάρωσης στις διεπαφές και τα ονόματα των συστημάτων που θα χρησιμοποιηθούν για τη συνολική επίλυση του προβλήματος. Το επόμενο βήμα στη διαδικασία είναι η δημιουργία και η διαμόρφωση των PDESolver και PDEMediator πρακτόρων. Ο PDECoordinator χρησιμοποιεί τις διευθύνσεις τους για να ρυθμίσει την επικοινωνία μεταξύ τους. Στη συνέχεια ο συντονιστής περιμένει για μηνύματα από τους μεσάζοντες, σχετικά με την κατάσταση της σύγκλισης στη λύση του προβλήματος, ή από το χρήστη. Τα μηνύματα από το χρήστη μπορεί να

αφορούν αλλαγή στις τιμές των συγκεκριμένων μεταβλητών του αρχείου εισόδου, όπως η ανοχή σύγκλισης ή την ανάγκη η εκτέλεση να σταματήσει.

Ο PDESolver πράκτορας είναι υπεύθυνος για την επίλυση του προβλήματος τοπικά. Αρχικά, εκκινεί το Pelltool που διαβάζει το αρχείο που περιγράφει το τοπικό πρόβλημα ΜΔΕ και δημιουργεί το εκτελέσιμο που θα χρησιμοποιηθεί αργότερα από το ExecuteTool. Τόσο το Pelltool όσο και το ExecuteTool είναι κομμάτια του συστήματος PELLPACK. Στη συνέχεια, ο επιλυτής εξάγει τα σημεία των διεπαφών από το αρχείο που περιέχει τα σημεία το πλέγματος και τα γράφει σε ένα αρχείο. Όταν τα αρχεία είναι έτοιμα, ο επιλυτής ενημερώνει τους μεσάζοντες και παραμένει αδρανής έως ότου ειδοποιηθεί από κάποιον μεσάζοντα ότι η λίστα με όλα τα σημεία και τις αρχικές μαντεψιές τους έχουν αποθηκευθεί σε κάποιο αρχείο. Ο επιλυτής χρησιμοποιεί αυτά τα αρχεία για να τρέξει το ExecuteTool ώστε να λύσει το πρόβλημα. Στέλνει μήνυμα στους μεσάζοντες ότι οι τιμές έχουν υπολογιστεί και περιμένει για την απάντησή τους. Ανάλογα με το μήνυμα των μεσάζοντων, ο επιλυτής θα ξαναλύσει το πρόβλημα, θα παραμείνει αδρανής περιμένοντας άλλους επιλυτές, ή θα κάνει την γραφική παράσταση της τοπικής λύσης. Με κατάλληλο μήνυμα ο PDECoordinator τερματίζει τον PDESolver.

Ο PDEMediator πράκτορας δημιουργείται και διαμορφώνεται από τον PDECoordinator. Αυτός ο πράκτορας μεσάζοντας έχει μια πλήρη περιγραφή της διεπαφής, της μεθόδου χαλάρωσης που χρησιμοποιείται, των επιλυτών με τους οποίους θα συνεργαστεί, της θέσης των αρχείων εισόδου/εξόδου, της θέσης των προγραμμάτων που επιλύουν τα τοπικά προβλήματα, της ανοχής που χρησιμοποιείται για να αποφασιστεί η σύγκλιση και της συνάρτησης των αρχικών μαντεψιών. Αυτές οι πληροφορίες παρέχονται από τον πράκτορα συντονιστή. Αφού εκκινηθεί, ο μεσάζοντας περιμένει για τα σημεία των ορίων από τους γειτονικούς επιλυτές. Στο επόμενο στάδιο, ο μεσάζοντας συνδυάζει τις δύο λίστες σημείων και μετά χρησιμοποιεί τη συνάρτηση αρχικών μαντεψιών για να υπολογίσει τις τιμές σε αυτά τα σημεία. Στη συνέχεια, ο μεσάζοντας στέλνει μήνυμα στους δύο επιλυτές ότι τα αρχεία με τα σημεία και τις τιμές τους είναι έτοιμα. Ο μεσάζοντας παραμένει αδρανής, περιμένοντας για καινούργιες τιμές από τους δύο επιλυτές. Όταν λάβει καινούργιες τιμές, μετακινείται στο επόμενο στάδιο, όπου διαβάζει τα καινούργια δεδομένα και τα συγκρίνει με τα τρέχοντα. Χρησιμοποιεί τη μέθοδο χαλάρωσης για να υπολογίσει τις καινούργιες τιμές για τις οριακές συνθήκες. Αν επιτευχθεί σύγκλιση σε αυτή τη διεπαφή, τότε ο μεσάζοντας στέλνει μήνυμα στους επιλυτές και πληροφορεί τον PDECoordinator για την τοπική σύγκλιση, ώστε αυτός να μπορεί να αποφασίσει για την ολική. Ένα μήνυμα από το συντονιστή στέλνει στο μεσάζοντα, ο οποίος τερματίζει σε περίπτωση ολικής σύγκλισης, ή περιμένει για καινούργια δεδομένα από τους δύο επιλυτές.

Το SciAgents απαιτεί σημαντική υποστήριξη σε παρεμβολή, μια διαδικασία για τον υπολογισμό αρχικών μαντεψιών, μηχανισμούς για τον καθορισμό καλών τιμών στις παραμέτρους χαλάρωσης και κριτήρια που να ελέγχουν την επαναληπτική διαδικασία. Η παρεμβολή χρειάζεται γιατί τα πλέγματα δεν συμπίπτουν απαραίτητα στο κομμάτι της διεπαφής. Επίσης, ο υπολογισμός των αρχικών τιμών της διεπαφής είναι μια αρκετά σύνθετη διαδικασία γιατί οι Neumann και οι μικτές οριακές συνθήκες είναι ευαίσθητες στις αρχικές τους μαντεψιές και καθώς τα προβλήματα ΜΔΕ γίνονται όλο και πιο πολύπλοκα, όλο και καλύτερες αρχικές μαντεψιές θα χρειάζονται.

2.3 Bond Framework

Μια επόμενη προσπάθεια κατασκευής ενός περιβάλλοντος επίλυσης προβλημάτων πολλαπλών πεδίων/πολλαπλών φυσικών [68] χρησιμοποιεί το middleware πρακτόρων Bond [69]. Και αυτή η υλοποίηση χρησιμοποιεί την PELLPACK για την επίλυση των ΜΔΕ και οι κατηγορίες πρακτόρων, καθώς επίσης και οι λειτουργίες τους είναι ταυτόσημες με αυτές του SciAgents Framework.

Το Bond είναι ένα κατανεμημένων-αντικειμένων, προσανατολισμένο σε μηνύματα σύστημα, που παρέχει ένα δημιουργικό πλαίσιο για την κατασκευή συνεργατικών πρακτόρων δικτύου. Το Bond διατίθεται κάτω από μια άδεια ανοιχτού κώδικα LPGL και χρησιμοποιήθηκε ως μια μηχανή δημιουργίας ροών εργασιών που υποστηρίζει δυναμικές ροές εργασίας [70], για μια προσαρμοστική υπηρεσία βίντεο [71], για την ανακάλυψη των πόρων σε ένα κατανεμημένο σύστημα ευρείας περιοχής [72], για τον σχεδιασμό ενός δικτύου επιλυτών ΜΔΕ [73], και για άλλες εφαρμογές.

Το Bond χρησιμοποιεί τη KQML [74] ως μια μετα-γλώσσα για την επικοινωνία μεταξύ αντικειμένων. Υποστήριξη για βασισμένη σε XML επικοινωνία προστέθηκε στο σύστημα πρόσφατα. Η KQML προσφέρει μια ποικιλία από τύπους μηνυμάτων (performatives) που εκφράζουν μια συμπεριφορά σχετικά με το περιεχόμενο της ανταλλαγής. Τα performatives μπορούν επίσης να βοηθήσουν τους πράκτορες να βρουν άλλους πράκτορες που μπορούν να επεξεργαστούν τις αιτήσεις τους. Ένας performative εκφράζεται ως μια συμβολοσειρά ASCII, χρησιμοποιώντας τη σημειογραφία Common Lisp Polish-prefix. Η πρώτη λέξη στην ακολουθία είναι το όνομα του performative, ακολουθούμενο από παραμέτρους. Οι παράμετροι στα performatives δεικτοδοτούνται από τις λέξεις-κλειδιά και είναι, ως εκ τούτου, ανεξάρτητες της σειράς.

Η υποδομή που παρέχεται από Bond υποστηρίζει βασικό χειρισμό αντικειμένων, επικοινωνία μεταξύ αντικειμένων, υπηρεσίες τοπικού καταλόγου και τοπικής διαμόρφωσης, ένα κατανεμημένο μηχανισμό ευαισθητοποίησης, ανιχνευτές για

την ασφάλεια και τον έλεγχο λειτουργιών, γραφικές διεπαφές χρήστη, και βοηθητικά προγράμματα.

Η μεθοδολογία για τη δημιουργία ενός πράκτορα στο Bond είναι: (α) καταγραφή μιας σύντομης περιγραφής των δράσεων σε κάθε κατάσταση, (β) δημιουργία ενός διαγράμματος μετάβασης καταστάσεων για κάθε πράκτορα (γ) αναζήτηση στις βάσεις δεδομένων του Bond για στρατηγικές κατάλληλες για νέους πράκτορες (δ) δημιουργία νέων στρατηγικών όποτε είναι αναγκαίο.

Συνοπτικά, η συγκεκριμένη προσπάθεια προτείνει ότι κάποιος μπορεί να πετύχει παραλληλισμό χρησιμοποιώντας σειριακούς ήδη υλοποιημένους επιλυτές χωρίς να χρειάζεται να τους τροποποιήσει καθόλου. Μια ενδιαφέρουσα εναλλακτική στην παραλληλοποίηση κώδικα είναι η εφαρμογή μιας μεθοδολογίας διαίρει και βασίλευε βασισμένη σε διαιρέσεις των δεδομένων σε υπο-πεδία, η επίλυση των υπο-προβλημάτων ανεξάρτητα σε κάθε υπο-πεδίο και μετά η εύκολη επίλυση των προκύπτουσων συγκρούσεων μεταξύ των ανεξάρτητων εργατών. Οι πράκτορες συντονίζουν την εκτέλεση και δρουν ως μεσάζοντες στις συγκρούσεις. Το κύριο πλεονέκτημα της συγκεκριμένης υλοποίησης είναι η δραστική μείωση του χρόνου ανάπτυξης.

2.4 GasTurbnLab

Το GasTurbnLab [75, 76] είναι η τελευταία ολοκληρωμένη προσπάθεια. Είναι ένα πολλαπλών πεδίων περιβάλλον επίλυσης προβλημάτων για τον σχεδιασμό αεριοστροβίλων βασισμένο στο Grasshopper middleware [77] πρακτόρων και βιβλιοθήκες FORTRAN και C για την επίλυση των ΜΔΕ.

Το Grasshopper είναι μια πλατφόρμα ανάπτυξης πρακτόρων που υποστηρίζει κατανεμημένες εφαρμογές βασισμένες σε πράκτορες. Η πλατφόρμα παρέχει μια βάση για υπηρεσίες επικοινωνίας, κινητή υπολογιστική ισχύ και δυναμική ανάκτηση πληροφοριών. Πρόκειται ουσιαστικά για μια πλατφόρμα κινητών πρακτόρων που είναι χτισμένη πάνω σε ένα κατανεμημένο περιβάλλον επεξεργασίας, ενσωματώνοντας το παραδοσιακό μοντέλο client-server και την τεχνολογία των κινητών πρακτόρων. Το κύριο χαρακτηριστικό της πλατφόρμας Grasshopper είναι ο ανεξάρτητος της τοποθεσίας υπολογισμός, με γνώμονα την ικανότητα να μετακινεί πράκτορες μεταξύ των διαφόρων συστημάτων. Είναι ένα ισχυρό περιβάλλον που διευκολύνει τη δημιουργία των πρακτόρων, διαφανώς τοποθετώντας τους και ελέγχοντας την εκτέλεσή τους. Αυτές οι εφαρμογές βασισμένες σε πράκτορες είναι διαλειτουργικές με άλλα συστήματα πρακτόρων που είναι MASIF συμβατά. Το Grasshopper κατανεμημένων πρακτόρων περιβάλλον αποτελείται από περιοχή, φορείς και πράκτορες. Στην κορυφή της ιεραρχίας είναι η περιοχή που διαχειρίζεται τα κατανεμημένα συστατικά στο περιβάλλον

Grasshopper. Οι φορείς και οι πράκτορες σχετίζονται με μια συγκεκριμένη περιοχή. Κάθε περιοχή έχει ένα μητρώο που διατηρεί πληροφορίες σχετικά με όλα τα συστατικά που συνδέονται με αυτήν. Ένας οργανισμός είναι το περιβάλλον εκτέλεσης για κινητούς και σταθερούς πράκτορες, που παρέχουν τη λειτουργικότητα για την υποστήριξη της εκτέλεσης του πράκτορα. Ο οργανισμός είναι υπεύθυνος για μια σειρά από υπηρεσίες, όπως οι εξής: (1) υπηρεσίες επικοινωνίας για όλες τις απομακρυσμένες αλληλεπιδράσεις που λαμβάνουν χώρα μεταξύ των συστατικών του Grasshopper, τις κινήσεις τους και τις μεταφορές, (2) την υπηρεσία καταχώρησης που παρακολουθεί όλους τους τρέχοντες πράκτορες, (3) υπηρεσίες διαχείρισης που καταγράφουν τον εξωτερικό έλεγχο των πρακτόρων, (4) υπηρεσίες μεταφοράς για τη μετάβαση των πρακτόρων από έναν οργανισμό σε έναν άλλο, (5) υπηρεσίες ασφαλείας για την προστασία των απομακρυσμένων αλληλεπιδράσεων και τους πόρους του οργανισμού και (6) επίμονες υπηρεσίες για να καταστεί δυνατή η αποθήκευση των πρακτόρων για πιθανή ανάκαμψη. Οι πράκτορες είναι υπολογιστικά προγράμματα που χαρακτηρίζονται από ένα σύνολο χαρακτηριστικών. Μπορούν να είναι κινητοί ή σταθεροί. Οι κινητοί πράκτορες μετακινούνται από το ένα σημείο στο άλλο μέσα σε μια περιοχή για να επωφεληθούν από τις τοπικές αλληλεπιδράσεις, και ως εκ τούτου είναι ικανοί να μειώνουν το φορτίο του δικτύου με τη μετεγκατάσταση. Οι σταθεροί πράκτορες συνδέονται με μόνο μια συγκεκριμένη τοποθεσία, και είναι ανίκανοι μετανάστευσης.

Το περιβάλλον Grasshopper υποστηρίζει την παραδοσιακή δομή πελάτη-διακομιστή. Οι MPSE πελάτες και διακομιστές είναι Grasshopper πράκτορες που παρέχουν μοντελοποίηση (προδιαγραφή του φυσικού αντικειμένου που πρόκειται να προσομοιωθεί), έλεγχο της προσομοίωσης και υπολογιστική υπηρεσία. Το πλαίσιο MPSE αποτελείται από 6 κατηγορίες πρακτόρων: τον ModelingAgent, τον SimulationControlAgent (SCA), τον VisualizerAgent (VA), τους LegacyCodeAgents (LCA), τους MediatorAgents (MA) και τους ServiceAgents. Ο ModelingAgent και ο SCA είναι πράκτορες πελάτη, οι υπόλοιπες κατηγορίες πρακτόρων είναι διακομιστές. Οι ModelingAgent, SCA και VA έχουν γραφικές διεπαφές χρήστη που υποστηρίζουν την αλληλεπίδραση του χρήστη. Οι ServiceAgents παρέχουν υπηρεσίες αποκλειστικά για τον SCA, όπως η αναγνώριση των πόρων και η πρόσβαση σε βάσεις δεδομένων για την ανάκτηση πληροφοριών πράκτορα. Ο ModelingAgent αποτελεί το πρώτο επίπεδο διεπαφής χρήστη. Επιτρέπει στους χρήστες να καθορίσουν το αντικείμενο προσομοίωσης στόχο (πεδίο) και τη γεωμετρική διάσπασή του σε υποπεδία. Οι χρήστες μπορούν να επιλέξουν τη διαμόρφωση των υποπεδίων που θα προσομοιωθούν. Αυτά τα υποπεδία περνάνε στο πλέγμα ή σε εργαλεία διακριτοποίησης, τα οποία έχουν ενσωματωθεί στο ModelingAgent ως αυτόνομα συστήματα. Τα εργαλεία διακριτοποίησης χρησιμοποιούνται για να παράγουν το πλέγμα και καθορίζουν τις οριακές συνθήκες

για τα υποπεδία που έχουν επιλεγεί από το χρήστη. Ο ModelingAgent βοηθά τους χρήστες να εντοπίσουν τις διασυνδέσεις μεταξύ των πεδίων, να καθορίσουν πληροφορίες ανταλλαγής δεδομένων για κάθε διεπαφή, και να επιλέξουν επιλυτές για κάθε υποπεδίο. Αυτή η πληροφορία γίνεται τα δεδομένα εκκίνησης για το SCA.

Ο SCA ζητά και διαχειρίζεται τις υπηρεσίες όλων των πρακτόρων διακομιστών. Ελέγχει την όλη διαδικασία προσομοίωσης: έναρξη / τερματισμός των υπολογιστικών servers, διαχείριση των αλληλεπιδράσεων τους, και διευκόλυνση της διαδικτυακής ασύγχρονης επικοινωνίας των υπολογιστικών δεδομένων και δεδομένων ελέγχου. Ο VA είναι ένας πράκτορας διακομιστής που λαμβάνει δεδομένα λύσης και τα καθιστά γραφικά χρησιμοποιώντας το σύστημα οπτικοποίησης δεδομένων IRIS Explorer [78]. Οι LCA είναι οι υπολογιστικοί πράκτορες που προσομοιώνουν τα υποπεδία του φυσικού αντικείμενου που αποσυντέθηκε; κάθε LCA περικλείει ένα καθιερωμένο κώδικα επιλυτή που στοχεύει σε ένα συγκεκριμένο φυσικό υπο-αντικείμενο. Ο επαναληπτικός αλγόριθμος MPSE απαιτεί έναν LCA για να επικοινωνήσει τα οριακά του δεδομένα (μέσω του SCA) σε έναν ή περισσότερους MAs. Οι MAs περικλείουν τους κώδικες μεσολάβησης που είναι υπεύθυνοι για τη ρύθμιση και την επίλυση των διεπαφών (που αναπαρίστανται από τα οριακά δεδομένα) μεταξύ γειτονικών υποπεδίων, καθένα από τα οποία προσομοιώνεται με ένα LCA. Ο SCA μπορεί να χειριστεί οποιοδήποτε αριθμό υποπεδίων; δηλαδή, μπορεί να ελέγξει οποιοδήποτε αριθμό και είδος επικοινωνίας κώδικα επιλυτή και πρακτόρων μεσολαβητών.

2.5 IRTool: Εργαλειοθήκη MATLAB

Τελευταία, μια εργαλειοθήκη στη MATLAB για την επίλυση προβλημάτων πολλαπλών πεδίων / πολλαπλών φυσικών είναι υπό κατασκευή, ενώ μια πρώτη σταθερή της έκδοση παρουσιάζεται στο [79].

Το IRTool (Interface Relaxation Tool) είναι μια επέκταση της εργαλειοθήκης PDETool της MATLAB που είναι ανοιχτού κώδικα και λύνει δισδιάστατα προβλήματα ΜΔΕ. Αν και το PDETool μπορεί να χειριστεί προβλήματα πολλαπλών πεδίων στο επίπεδο της γραμμικής άλγεβρας (με τεχνικές διακριτοποίησης πεδίου), δε μπορεί να επιλύσει προβλήματα πολλαπλών πεδίων / πολλαπλών φυσικών με μεθόδους χαλάρωσης στη διεπαφή. Το κοινό όριο μεταξύ διαφορετικών πεδίων ενός προβλήματος ΜΔΕ, τη διεπαφή δηλαδή, η μεθοδολογία χαλάρωσης στη διεπαφή μπορεί να τη χειριστεί όπως απαιτεί η φυσική του προβλήματος σε ένα υψηλό επίπεδο διακριτοποίησης. Το PDETool ενσωματώθηκε στο IRTool, ώστε να είναι δυνατός ο ορισμός ΜΔΕ πολλαπλών πεδίων / πολλαπλών φυσικών, η θέσπιση κατάλληλων οριακών συνθηκών του αρχικού προβλήματος, ο ορισμός / κλήση κατάλληλων μεθόδων στις διεπαφές, η επίλυση του αρχικού προβλήματος και η οπτικοποίηση της λύσης του αρχικού προβλήματος.

Μια πρώτη διαφοροποίηση του IRTool είναι η ικανότητά του να ανοίγει ταυτόχρονα περισσότερα από ένα παράθυρα (γραφικά περιβάλλοντα χρήστη όπου σχεδιάζουμε το πρόβλημα) ώστε να μπορεί να αποσυνθέσει το αρχικό πρόβλημα στα υποπροβλήματά του. Αφού ο χρήστης σχεδιάσει τη γεωμετρία που περιγράφει το πρόβλημά του, μια συνάρτηση αναλαμβάνει να βρει τα όρια και τις διεπαφές που έχει κάθε υποπεδίο. Αυτό είναι ένα απαραίτητο βήμα πριν τη διακριτοποίηση της γεωμετρίας, γιατί το PDETool μόνο του θα αγνοούσε τα τμήματα των διεπαφών. Στη συνέχεια, δημιουργούνται τα παράθυρα των υποπεδίων μιας και τώρα η γεωμετρία κάθε υποπεδίου είναι γνωστή, και γίνεται η αρχικοποίηση των υποπροβλημάτων. Σε αυτό το σημείο πρέπει να γίνει ο ορισμός των ορίων και των διεπαφών. Και εδώ έγιναν τροποποιήσεις στο PDETool ώστε να μπορεί να αντιλαμβάνεται ότι μια διεπαφή δεν είναι ένα κοινό όριο. Τροποποιήσεις έγιναν, επίσης, και στον τρόπο που γίνεται η δημιουργία του πλέγματος, ώστε αυτή να μπορεί να γίνει σε κάθε υποπεδίο ξεχωριστά. Η υλοποίηση της ίδιας της μεθόδου χαλάρωσης στη διεπαφή έχει τα εξής χαρακτηριστικά: ο χρήστης δίνει σα παραμέτρους τον αριθμό των επαναλήψεων και το μέγιστο σφάλμα της μεθοδολογίας χαλάρωσης, δίνονται αρχικές τιμές στις διεπαφές, το κάθε υποπεδίο επιλύεται ξεχωριστά, οι τιμές της λύσης και της παραγώγου στα σημεία των διεπαφών γειτονικών υποπεδίων χρησιμοποιούνται για τον υπολογισμό των καινούργιων τιμών των διεπαφών και μετά ακολουθεί μια νέα επανάληψη.

3 Μέθοδοι χαλάρωσης στη διεπαφή

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 επιπλέον θεωρητική και πειραματική ανάλυση απαιτείται πρώτου 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 [80] και ερευνούν τις σχετιζόμενες στρατηγικές preconditioning [81, 82] 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 [56, 57, 83].

Οι μέθοδοι χαλάρωσης στη διεπαφή μας πάνε ένα βήμα παραπέρα από τις μη επικαλυπτόμενες μεθόδους διακριτοποίησης πεδίου [63]. 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 περισσότερες από τις γνωστές μεθόδους χαλάρωσης στη διεπαφή απαριθμούνται παρακάτω με την αλφαβητική σειρά των ακρωνυμίων τους:

- **AVE** Μια απλή μέθοδος που παίρνει το μέσο όρο της λύσης και της κανονικής της παραγώγου κατά μήκος των διεπαφών.
- **GEO** Μια μέθοδος που βασίζεται σε μια απλή γεωμετρική συστολή.
- **NEW** Ένα σχήμα που βασίζεται στη μέθοδο Newton για να διορθώνει τις τιμές στις διεπαφές.
- **ROB** Ένας αλγόριθμος που χρησιμοποιεί τις Robin συνθήκες στις διεπαφές για εξομάλυνση.
- **SCO** Ένα σχήμα που βασίζεται σε (αλλά δεν έχει διατυπωθεί με) μια προσέγγιση συμπληρώματος Schur.
- **SHO** Μια μέθοδος βασισμένη στη γενική ιδέα της shooting μεθόδου για την επίλυση των συνήθων διαφορικών εξισώσεων (ΣΔΕ)
- **SPO** Μια μέθοδος που προέρχεται από τη χρήση του τελεστή Steklov-Poincaré που περιλαμβάνει εναλλαγή τύπων οριακών συνθηκών.

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 [84, 85, 86, 87].

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

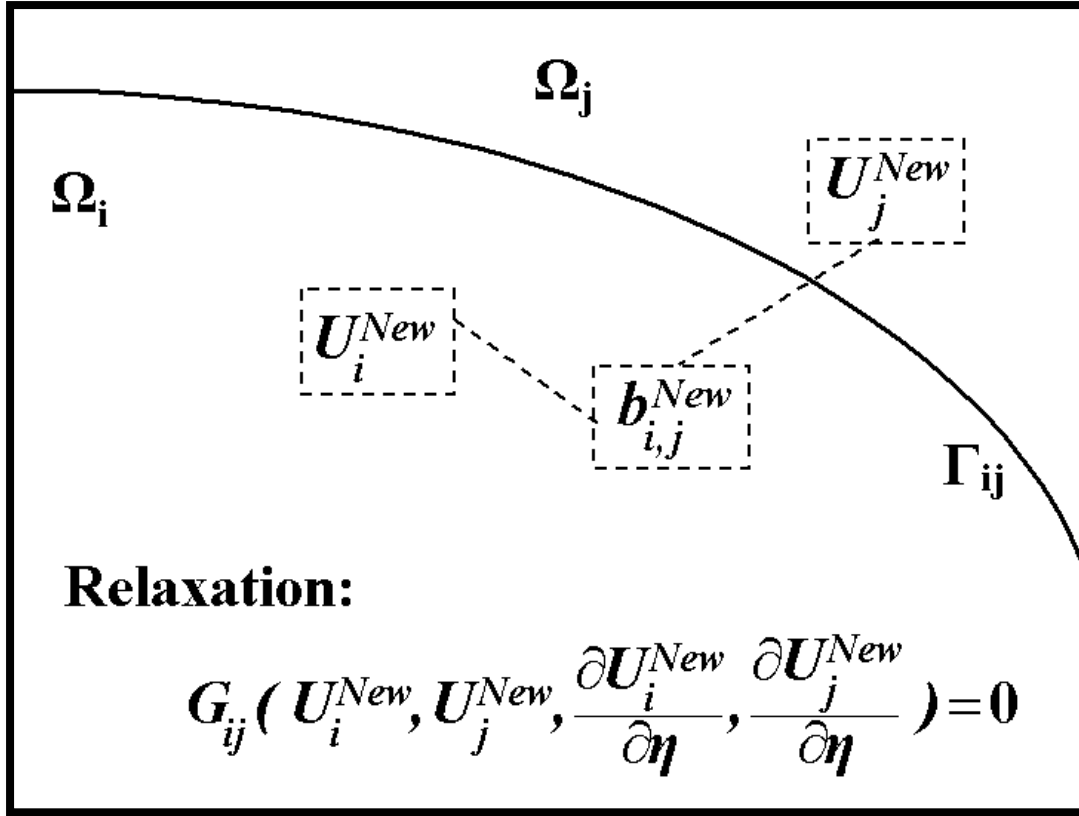
- Δεν είναι ευαίσθητες σε άλματα στους συντελεστές του τελεστή. Η συμπεριφορά σύγκλισής τους και οι υπολογισμοί θεωρητικού σφάλματός τους παραμένουν ίδια ακόμα και αν ο διαφορικός τελεστής περιλαμβάνει ασυνεχείς συντελεστές, αρκεί τα άλματα να συμβαίνουν κατά μήκος των γραμμών των διεπαφών [88].
- Έχουν 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 [66].

- Η καταγραφή/συντήρηση είναι σχετικά ευκολότερη για τη διαμέριση και το χειρισμό των σχετικών δομών δεδομένων απ' ότι στις πολύπλοκες και ακριβές επικαλυπτόμενες μεθόδους [66].

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 [89] και μια γενική διατύπωση και ανάλυση των μεθόδων χαλάρωσης στη διεπαφή στο [59]. 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 $G_{i,j}$ (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 $\Gamma_{i,j}$ 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 $\Omega = \bigcup_{i=1}^p \overline{\Omega_i} \setminus \partial\Omega$ 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

διπλωματική 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.

3.3 Μεθοδολογίες χαλάρωσης στη διεπαφή

Εξαιτίας της εγγενούς αφαίρεσης, είναι σχετικά εύκολο να περιγραφούν οι μέθοδοι λείανσης των διεπαφών τόσο στο επίπεδο της έννοιας, όσο και στο αλγοριθμικό. Στη συνέχεια της υποενότητας παρουσιάζονται επτά μέθοδοι με την υψηλού επιπέδου αλγοριθμική τους περιγραφή. Για λόγους απλότητας στην παρουσίαση των αλγορίθμων θεωρούμε μόνο μονοδιάστατο (κατά μήκος του άξονα x) διαχωρισμό του πεδίου. Οπότε κάθε υποπεδίο έχει δύο γραμμές διεπαφών με τα δύο γειτονικά υποπεδία. Το βασικό μπλοκ κώδικα των αλγορίθμων είναι η διαδικασία `u=solve_pde(ui,dui)` που υπολογίζει τη λύση u ενός τοπικού σε ένα υποπεδίο προβλήματος ΜΔΕ με Dirichlet, Neumann ή Robin οριακές συνθήκες στη διεπαφή με u_i τιμή στη διεπαφή και du_i παράγωγο στη διεπαφή. Τα R και L υποδεικνύουν δεξιά και αριστερά υποπεδία ή διεπαφές, αντίστοιχα και το u_i τη λύση του προβλήματος που σχετίζεται με το υποπεδίο Ω_i .

Η μέθοδος Dirichlet/Neumann Averaging (AVE)

Είναι ένα από τα πιο απλά σχήματα και αποτελείται από δύο περάσματα επίλυσης ΜΔΕ μαζί με δύο βήματα χαλάρωσης διεπαφών. Στο πρώτο πέραςμα το Dirichlet πρόβλημα λύνεται σε όλα τα υποπεδία. Στη συνέχεια η διαδικασία χαλάρωσης λειαίνει τις παραγώγους κατά μήκος των διεπαφών υπολογίζοντας την κανονική παράγωγο σαν ένα συνδυασμό των προηγούμενως υπολογισμένων κανονικών παραγώγων των δύο γειτονικών υποπεδίων. Αυτές οι εκτιμήσεις χρησιμοποιούνται στη συνέχεια σαν οριακές συνθήκες στο δεύτερο πέραςμα επίλυσης ΜΔΕ, όπου το πρόβλημα Neumann επιλύεται σε όλα τα υποπεδία. Το δεύτερο βήμα χαλάρωσης ακολουθεί και υπολογίζει εκτιμήσεις της άγνωστης συνάρτησης στις διεπαφές παίρνοντας ένα συνδυασμό των προηγούμενως υπολογισμένων λύσεων στα γειτονικά υποπεδία. Αυτές οι εκτιμήσεις πρόκειται να γίνουν είσοδοι στην επόμενη επανάληψη του Dirichlet περάσματος. Αυτή η μέθοδος, που μπορεί να θεωρηθεί μια μέθοδος δύο βημάτων, περιγράφεται αλγοριθμικά ως εξής:

for $k = 0, 1, 2, \dots$

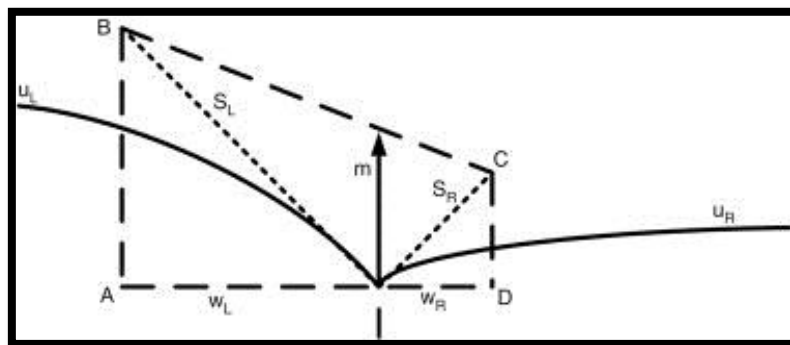
- $u^{(k+\frac{1}{2})} = \text{solve_pde}(u_i)$ σε κάθε υποπεδίο
- $du_i = \beta \frac{\partial u_R^{(k+\frac{1}{2})}}{\partial x} + (1 - \beta) \frac{\partial u_L^{(k+\frac{1}{2})}}{\partial x}$ σε κάθε διεπαφή
- $u^{(k+1)} = \text{solve_pde}(du_i)$ σε κάθε υποπεδίο
- $u_i = \alpha u_R^{(k+1)} + (1 - \alpha) u_L^{(k+1)}$ σε κάθε διεπαφή

όπου $\alpha, \beta \in (0, 1)$ είναι οι παράμετροι χαλάρωσης. Υπάρχει ένας αριθμός θεωρητικών μελετών για τη σύγκλιση του παραπάνω σχήματος [60]. Πιο συγκεκριμένα, στο [55] πραγματοποιείται η ανάλυση σύγκλισης της μεθόδου στο διαφορικό επίπεδο χρησιμοποιώντας Hilbert space τεχνικές. Στο [90] η Galerkin πεπερασμένων στοιχείων μέθοδος και η υβριδική μικτή πεπερασμένων στοιχείων μέθοδος χρησιμοποιούνται για να δώσουν διακριτές εκδοχές της μεθόδου. Ανάλυση Fourier χρησιμοποιείται στο [91] για να δώσει ακριβή αποτελέσματα σύγκλισης και να υπολογίσει βέλτιστες τιμές για τις παραμέτρους χαλάρωσης σε απλά προβλήματα μοντέλα.

Η Geometric (GEO) Contraction Based μέθοδος

Η GEO υπολογίζει τη νέα λύση για κάθε υποπεδίο επιλύοντας ένα Dirichlet πρόβλημα και κατηγοριοποιείται σαν μία μέθοδος ενός βήματος. Οι τιμές στις διεπαφές λαμβάνονται προσθέτοντας στις παλιές ένα γεωμετρικά σταθμισμένο συνδυασμό των κανονικών οριακών παραγώγων των γειτονικών υποπεδίων. Πιο

συγκεκριμένα, στην εικόνα 2 θεωρούμε μια μονοδιάστατη περίπτωση και με τα u_L και u_R τις λύσεις των διαφορικών προβλημάτων που σχετίζονται με το πεδίο αριστερά και δεξιά ενός σημείου διεπαφής I , αντίστοιχα. Ας υποθέσουμε ότι είναι ίσες στο I (αν όχι, μπορούμε να πάρουμε το μέσο όρο τους στο I) αλλά οι κλίσεις τους δεν ταιριάζουν, εκτός και αν προσθέσουμε έναν όρο διόρθωσης m , όπως αναπαρίσταται γραφικά στην εικόνα 1. Για να υπολογίσουμε το m θεωρούμε τα δύο τρίγωνα IAB και CDI των οποίων τα ύψη h_L και h_R δίνονται πολλαπλασιάζοντας την αντίστοιχη εφαπτομένη (ή ισοδύναμα την κανονική παράγωγο) με τη βάση του τριγώνου. Τα w_L και w_R είναι τα πλάτη που θεωρούμε για την εγκυρότητα των κλίσεων S_L και S_R , αντίστοιχα στο I . Μπορούν να επιλεγούν αυθαίρετα ή να παίξουν το ρόλο μιας παραμέτρου χαλάρωσης που μπορεί να αλλάζει δυναμικά από επανάληψη σε επανάληψη. Οι νέες τιμές των διεπαφών μπορούν τώρα να υπολογιστούν προσθέτοντας ένα σταθμισμένο μέσο όρο των υψών στις παλιές τιμές των διεπαφών. Αφαιρετικά, η περιγραφείσα μορφή χαλάρωσης μπορεί να ειπωθεί να τραβάμε τις συναρτήσεις u_L και u_R από το σημείο I και κατά m προς τα πάνω μέχρι να γίνουν συνεχείς οι παράγωγοί τους.



Εικόνα 2: Ο μηχανισμός χαλάρωσης της μεθόδου GEO

Η GEO μπορεί αλγοριθμικά να αποδοθεί ως εξής:

for $k = 0, 1, 2, \dots$

- $ui^{(k+1)} = \frac{u_L^{(k)} + u_R^{(k)}}{2} - \frac{w_L w_R}{w_L + w_R} \left(\frac{\partial u_L^{(k)}}{\partial x} - \frac{\partial u_R^{(k)}}{\partial x} \right)$ σε κάθε διεπαφή
- $u^{(k+1)} = solve_pde(ui^{(k+1)})$ σε κάθε υποπεδίο

Η θεωρητική ανάλυση της μεθόδου τόσο στο συνεχές (διαφορικό) όσο και στο διακριτό (γραμμικής άλγεβρας) επίπεδο για την επίλυση ελλειπτικών διαφορικών εξισώσεων παρουσιάζεται στο [65]. Εκεί αποδεικνύεται η σύγκλιση της μεθόδου για μονοδιάστατα προβλήματα. Η περιοχή σύγκλισης και οι βέλτιστες τιμές για τις

παραμέτρους χαλάρωσης καθορίζονται για προβλήματα μοντέλα. Αριθμητικά δεδομένα για μονοδιάστατα και δισδιάστατα προβλήματα που επιβεβαιώνουν τα θεωρητικά αποτελέσματα, διερευνούν την αποτελεσματικότητα της μεθόδου και διαφωτίζουν τα χαρακτηριστικά της παρουσιάζονται επίσης.

Η μέθοδος Newton (NEW)

Μια ακόμα νέα ιδέα είναι η χρήση της διακριτής μεθόδου Newton για την ανανέωση των τιμών στη διεπαφή σύμφωνα με την ακόλουθη διαδικασία:

Βήμα 0: Για $i = 1, 2$,

μάντεψε τις $u_R^{(i)}, u_L^{(i)}$ στις διεπαφές και υπολόγισε τα $\frac{\partial u_L^{(i)}}{\partial x}, \frac{\partial u_L^{(i)}}{\partial x}$

Βήμα 1: Υπολόγισε τα δ_L και δ_R ώστε σε όλες τις διεπαφές να έχουμε:

$$(u_L^{(2)} + \delta_L) - (u_R^{(2)} + \delta_R) = 0$$

$$\frac{\partial u_L^{(2)}}{\partial x} (u_L^{(2)} + \delta_L) - \frac{\partial u_R^{(2)}}{\partial x} (u_R^{(2)} + \delta_R) = 0$$

Βήμα 2: Εφάρμοσε γραμμικοποίηση για να επιλύσεις προσεγγιστικά τις εξισώσεις:

$$(u_L^{(2)} + \delta_L) - (u_R^{(2)} + \delta_R) = 0$$

$$\frac{\partial u_L^{(2)}}{\partial x} (u_L^{(2)}) [1 + \frac{\partial}{\partial u_L} (\frac{\partial u_L}{\partial x}) \delta_L] - \frac{\partial u_R^{(2)}}{\partial x} (u_R^{(2)}) [1 + \frac{\partial}{\partial u_R} (\frac{\partial u_R}{\partial x}) \delta_R] = 0$$

Βήμα 3: Προσέγγισε τις άγνωστες παραγώγους με διαφορές:

$$\frac{\partial}{\partial u_L} \left(\frac{\partial u_L}{\partial x} \right) = \frac{\frac{\partial u_L^{(2)}}{\partial x} - \frac{\partial u_L^{(1)}}{\partial x}}{u_L^{(2)} - u_L^{(1)}} = A_L$$

$$\frac{\partial}{\partial u_R} \left(\frac{\partial u_R}{\partial x} \right) = \frac{\frac{\partial u_R^{(2)}}{\partial x} - \frac{\partial u_R^{(1)}}{\partial x}}{u_R^{(2)} - u_R^{(1)}} = A_R$$

Βήμα 4: Λύσε ως προς δ_L και δ_R

$$\delta_L - \delta_R = u_R^{(2)} - u_L^{(2)} = 0$$

$$A_L \delta_L - A_R \delta_R = \frac{\partial u_R^{(2)}}{\partial x} - \frac{\partial u_L^{(2)}}{\partial x}$$

$$\text{ώστε } \delta_L = \delta_R = \delta = \left[\frac{\partial u_R^{(2)}}{\partial x} - \frac{\partial u_L^{(2)}}{\partial x} \right] / (A_L - A_R).$$

Οι τιμές των $\frac{\partial u_L}{\partial x}$ και $\frac{\partial u_R}{\partial x}$ εξαρτώνται από τις u_L και u_R για όλες τις διαφορικές εξισώσεις παραπάνω. Στο πνεύμα της παραπάνω διαδικασίας, η μέθοδος NEW μπορεί να περιγραφεί ως εξής:

for $k = 2, 3, 4 \dots$

- Λύσε τις διορθώσεις $\delta_L = \delta_R = \delta$ από το σύστημα των διεπαφών:
 - $(u_L^{(k)} + \delta_L) - (u_R^{(k)} + \delta_R) = 0$
 - $\frac{\partial u_L^{(k)}}{\partial x}(u_L^{(k)} + \delta_L) - \frac{\partial u_R^{(k)}}{\partial x}(u_R^{(k)} + \delta_R) = 0$
- $u_i^{(k+1)} = u_i^{(k)} + \delta$ σε κάθε διεπαφή
- $u^{(k+1)} = \text{solve_pde}(u_i^{(k+1)})$ σε κάθε υποπεδίο

Δεν υπάρχει γενική ανάλυση σύγκλισης για αυτό το νέο ενός βήματος σχήμα που δεν περιλαμβάνει παραμέτρους χαλάρωσης. Αλλά όπως οι περισσότερες εφαρμογές της μεθόδου Newton αναμένεται να συγκλίνει πολύ γρήγορα σε κάποια γειτονιά της πραγματικής λύσης.

Η μέθοδος Robin Relaxation (ROB)

Μια ακόμα απλούστερη μέθοδος χαλάρωσης στη διεπαφή είναι αυτή που βασίζεται στις οριακές συνθήκες Robin για να μεταδώσει πληροφορία διαμέσου των ορίων των υποπεδίων. Προτάθηκε αρχικά στο [54] και αναλύθηκε αργότερα στο [92]. Επιλύεται η τοπική ΜΔΕ στα υποπεδία χρησιμοποιώντας συνθήκες Robin στις γραμμές των διεπαφών αντιστοιχίζοντας ένα συνδυασμό Dirichlet και Neumann δεδομένων από τα γειτονικά υποπεδία.

for $k = 0, 1, 2, \dots$

Σε κάθε υποπεδίο λύσε:

- $Lu^{(k+1)} = f \in \Omega$ με
- $-\frac{\partial u^{(k+1)}}{\partial x} + \lambda u^{(k+1)} =$
 $-\frac{\partial u_L^{(k)}}{\partial x} + \lambda u_L^{(k)}$ στην αριστερή διεπαφή του υποπεδίου
- $\frac{\partial u^{(k+1)}}{\partial x} + \lambda u^{(k+1)} =$
 $\frac{\partial u_R^{(k)}}{\partial x} + \lambda u_R^{(k)}$ στη δεξιά διεπαφή του υποπεδίου

Εδώ το λ είναι μια παράμετρος χαλάρωσης. Η σύγκλιση αυτής της μεθόδου αναλύθηκε στο [54] στο διαφορικό επίπεδο υποθέτοντας αυθαίρετες αποσυνθέσεις και χρησιμοποιώντας εκτιμήσεις «ενέργειας». Ο καθορισμός αποδοτικών επιλογών για το λ είναι ένα ανοιχτό πρόβλημα. Παραλλαγές της παραπάνω μεθόδου έχουν εμφανιστεί τελευταία στη βιβλιογραφία. Πιο συγκεκριμένα στο [93] μια βασισμένη σε ADI παραλλαγή για την επιτάχυνση της σύγκλισης του σχήματος ROB προτείνεται και αναλύεται. Μια παραλλαγή της ROB, που επεκτείνει την εφαρμογή της και την απελευθερώνει από το πρόβλημα της διασταύρωσης σημείου, διατυπώνεται και αναλύεται στο [94]. Μια ακόμα παραλλαγή που χρησιμοποιεί και τις εφαπτομενικές παραγώγους εκτός από την κανονική παράγωγο για τη λείανση δίνεται στο [95] όπου βέλτιστες τιμές για τις παραμέτρους χαλάρωσης λαμβάνονται για ένα πρόβλημα μοντέλο.

Η μέθοδος Schur Complement (SCO)

Ανάμεσα στις πρώτες διαδικασίες χαλάρωσης στη διεπαφή που τράβηξαν την προσοχή των ερευνητών είναι αυτή που αναλύεται στο [50]. Εναλλάσσει Dirichlet και Neumann συνθήκες διεπαφών στο χώρο και μπορεί να περιγραφεί ως εξής:

for $k = 0, 1, 2, \dots$

- $u1_L = u, u1_R = \theta_1 u_2^{(k)} + (1 - \theta_1) u_1^{(k)}$
- $u_1^{(k+1)} = \text{solve_pde}(u1_L, u1_R)$
- for $i = 2, \dots, p - 1$
 - $dui_L = \frac{\partial u_{i-1}^{(k+1)}}{\partial x}$
 - $ui_R = \theta_i u_{i+1}^{(k)} + (1 - \theta_i) u_i^{(k)}$
 - $u_i^{(k+1)} = \text{solve_pde}(ui_R, dui_L)$
- $up_R = u, dup_L = \frac{\partial u_{p1}^{(k+1)}}{\partial x}$
- $u_p^{(k+1)} = \text{solve_pde}(up_R, dup_L)$

Εδώ το $\theta \in (0, 1)$ είναι μια παράμετρος χαλάρωσης. Η ανάλυση σύγκλισης στο διαφορικό επίπεδο για την περίπτωση της εξίσωσης Helmholtz με δύο μεταβλητές και μονοδιάστατες αποσυνθέσεις στο διαφορικό επίπεδο δίνεται στο [50] μαζί με εκφράσεις που οδηγούν σε βέλτιστες τιμές για το θ . Επίσης, δίνεται μια μέθοδος για να καθορίζονται δυναμικά, σε κάθε επανάληψη, τιμές για το θ για τη φασματική προσέγγιση συνεγκατάστασης των διαφορικών προβλημάτων. Η SCO είναι η μόνη τεχνική χαλάρωσης στη διεπαφή που έχει μέχρι τώρα επεκταθεί επιτυχώς και εφαρμοστεί σε τέταρτης τάξης ελλειπτικά προβλήματα [51].

Η μέθοδος Shooting (SHO)

Η συγκεκριμένη μέθοδος προτείνεται στο [52], όπου και διατυπώνεται αρχικά για μονοδιάστατα προβλήματα οριακών συνθηκών. Πραγματοποιείται επίσης η ανάλυση σύγκλισής της και βέλτιστες τιμές για τις παραμέτρους χαλάρωσης βρίσκονται για προβλήματα μοντέλα. Η βασική ιδέα είναι η σύνδεση των προβλημάτων στα υποπεδία επιλύοντας την ελλειπή εξίσωση $D(ui^{(k)}) \equiv \frac{\partial u_L^{(k)}}{\partial x} - \frac{\partial u_R^{(k)}}{\partial x} = 0$ στις διεπαφές χρησιμοποιώντας ένα σχήμα σταθερού σημείου (Picard) για την απόκτηση των νέων τιμών.

for $k = 0, 1, 2, \dots$

- $\alpha^{(k+1)} = \frac{a^{(k)} |D(ui^{(k+1)})|}{|D(ui^{(k)}) - D(ui^{(k+1)})|}$ σε κάθε διεπαφή
- $ui^{(k+2)} = ui^{(k)} - \alpha^{(k+1)} D(ui^{(k)})$ σε κάθε διεπαφή
- $u^{(k+2)} = solve_pde(ui^{(k+2)})$ σε κάθε υποπεδίο

Η μέθοδος του Steklov-Poincaré τελεστή (NEW)

Η συγκεκριμένη μέθοδος αναφέρθηκε αρχικά στο [53], αλλά αναλύθηκε από την οπτική του preconditioning μόνο. Χρησιμοποιεί τον τελεστή Steklov-Poincaré για να εκτελέσει την διαδικασία λείανσης των κανονικών παραγώγων στις διεπαφές. Είναι μια μέθοδος δύο βημάτων που περιγράφεται από τον ακόλουθο αλγόριθμο:

for $k = 0, 1, 2, \dots$

- $u^{(k+\frac{1}{2})} = solve_pde(ui)$ σε κάθε υποπεδίο
- $d ui = \frac{1}{2} (\frac{\partial u_R^{(k+\frac{1}{2})}}{\partial x} + \frac{\partial u_L^{(k+\frac{1}{2})}}{\partial x})$ σε κάθε διεπαφή
- $u^{(k+1)} = solve_pde(d ui)$ ($Lu = 0$) σε κάθε υποπεδίο
- $ui = ui - \frac{\rho}{2} (u_R^{(k+1)} + u_L^{(k+1)})$

Από την οπτική της χαλάρωσης στη διεπαφή δεν υπάρχουν θεωρητικά αποτελέσματα για τη μέθοδο SPO.

3.4 Επίλογος

Μία σύντομη μελέτη αξιολόγησης απόδοσης των μεθόδων μπορεί να βρεθεί στο [64] 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	Μονοτονικότητα Σύγκλισης	Αριθμός Παραμέτρων	Διαθέσιμη Θεωρία	Ιστορία	Ενός Βήματος
AVE	Μέση	Όχι	2	Λίγη	Όχι	Όχι
GEO	Μέση	Ναι	2	Λίγη	Ναι	Ναι
NEW	Μέση	Όχι	Καμία	Όχι	Ναι	Ναι
ROB	Χαμηλή	Όχι	1	Λίγη	Όχι	Ναι
SCO	Μέση	Όχι	1	Λίγη	Ναι	Όχι
SHO	Μέση	Ναι	Καμία	Λίγη	Ναι	Ναι
SPO	Υψηλή	Ναι	1	Όχι	Ναι	Όχι

Πίνακας 1: 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 διπλωματική 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 [65] 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.

4 Σύγχρονες υπολογιστικές αρχιτεκτονικές

4.1 Εισαγωγή

Κατά την τελευταία δεκαετία η αρχιτεκτονική των συστοιχιών υπολογιστών (Clusters) αναδείχθηκε και καθιερώθηκε ως η πλέον κατάλληλη πλατφόρμα για την επίλυση υπολογιστικά απαιτητικών προβλημάτων, γεγονός που καταδεικνύεται και από την παρακολούθηση της θέσης που κατέχει σε αυτό το χρονικό διάστημα η αρχιτεκτονική των Clusters ανάμεσα στα 500 ισχυρότερα υπολογιστικά συστήματα παγκοσμίως. Πρόσφατα, η σημαντική στροφή στον τομέα του σχεδιασμού των σύγχρονων επεξεργαστών, που οδήγησε στην αύξηση του εν δυνάμει παραλληλισμού στο πλαίσιο ενός επεξεργαστή, με την εισαγωγή των επεξεργαστών πολλαπλών πυρήνων, αναμένεται σύντομα να οδηγήσει στην κατασκευή συστοιχιών με επεξεργαστές της τάξης των εκατοντάδων πυρήνων (Clusters of Many-cores).

Οι εξελίξεις αυτές στην αρχιτεκτονική των συστημάτων υψηλής επίδοσης, ιδιαίτερα σημαντικές σε επίπεδο υλικού, αναμένονται ακόμη πιο καθοριστικές στο επίπεδο του λογισμικού. Το λογισμικό που θα προορίζεται για την επίλυση υπολογιστικά δύσκολων εφαρμογών στα σύγχρονα Clusters θα πρέπει να είναι σε θέση να αξιοποιεί τον παραλληλισμό που γίνεται διαθέσιμος μέσω του υλικού σε διαδοχικά επίπεδα, από το επίπεδο του επεξεργαστή έως το επίπεδο διασυνδεδεμένων συστοιχιών μέσω των Grids και των Clouds.

Στο πεδίο των υπολογιστικών πλεγμάτων (Grids) η έρευνα των τελευταίων χρόνων έχει οδηγήσει στην ανάπτυξη λογισμικού πλέγματος σημαντικά βελτιωμένου σε σχέση με τις πρώτες εκδόσεις που εμφανίστηκαν στις αρχές του 2000. Τα περιβάλλοντα Grid είναι αυτή τη στιγμή σε θέση να ενοποιούν με πιο ευέλικτο, πιο αποδοτικό και λιγότερο πολύπλοκο τρόπο τους ετερόκλητους υπολογιστικούς πόρους που τίθενται προς διάθεση σε ένα γεωγραφικά κατανεμημένο Grid.

Ταυτόχρονα, η πρόσφατη ανάπτυξη και προτυποποίηση των διαδικτυακών υπηρεσιών (web services) έχει συνδράμει σημαντικά στην εύχρηστη και ομοιόμορφη διάθεση των υπολογιστικών πόρων που βρίσκονται διαθέσιμοι στο πλαίσιο των εκάστοτε Grids, μέσω των υποδομών των Clouds. Η διάθεση της λειτουργικότητας και των υπολογιστικών πόρων των Clusters, των Grids, και των διαφόρων υποδομών δεδομένων (Data Centers) με τη μορφή υπηρεσιών στο πλαίσιο των Clouds, έχει απλοποιήσει σημαντικά την αξιοποίηση αυτών των πόρων.

4.2 Clusters

4.2.1 Εφαρμογές των clusters

Παρά τη ραγδαία εξέλιξη των υπολογιστών, εξακολουθούν να υπάρχουν υπολογιστικά προβλήματα με αυξημένες απαιτήσεις σε επεξεργαστική ισχύ, μνήμη και ταχύτητα εισόδου/εξόδου και για τον λόγο αυτό παραμένουν άλυτα ή δύσκολα επιλύσιμα. Το ερευνητικό πρόγραμμα High Performance and Communications (HPCC – Υπολογισμός και Επικοινωνίες Υψηλής Επίδοσης) προσδιόρισε ορισμένες περιοχές εφαρμογών τέτοιου τύπου με σπουδαία οικονομική και κοινωνική σημασία. Παραδείγματα τέτοιων εφαρμογών αποτελούν η τεχνολογία μαγνητικής καταγραφής, η λογική σχεδίαση φαρμάκων, οι μεταφορές υψηλής ταχύτητας, η χημική κατάλυση, η μοντελοποίηση ωκεανών, η μείωση του όζοντος, η ψηφιακή ανατομία, η ατμοσφαιρική ρύπανση, η σχεδίαση πρωτεϊνικών δομών και η κατανόηση εικόνας.

Για την επίλυση τέτοιων προβλημάτων, οι υπάρχοντες συμβατικοί υπολογιστές έχουν αποδειχθεί ανεπαρκείς. Συνεπώς, για την αντιμετώπισή τους, χρησιμοποιούνται πολυεπεξεργαστικά υπολογιστικά συστήματα, παράλληλα ή καταναμεμένα. Ωστόσο, τα τελευταία χρόνια κερδίζουν έδαφος υλοποιήσεις που βασίζονται στην ιδέα της συνεργασίας πολλών υπολογιστών για την επίλυση ενός προβλήματος. Συνεπώς, και για λόγους που παρουσιάζονται παρακάτω, παρουσιάζεται αυξημένο ενδιαφέρον για τα clusters, ως εναλλακτικά συστήματα υψηλής επίδοσης.

Εκτός από τις εφαρμογές που αναφέρθηκαν νωρίτερα και οι οποίες προέρχονται κυρίως από τον χώρο του επιστημονικού υπολογισμού, η χρήση clusters αποκτά ολοένα και περισσότερη αποδοχή στην περιοχή των ενσωματωμένων συστημάτων πραγματικού χρόνου και ιδιαίτερα σε εφαρμογές διαστημικής, επεξεργασίας σήματος και τηλεπικοινωνιών. Τέλος διαπιστώνεται ολοένα και αυξανόμενη χρήση clusters για την ανάκαμψη από σφάλματα (fail-over), την επίτευξη υψηλής διαθεσιμότητας (high availability), καθώς και την εξισορρόπηση φορτίου (load balancing), ιδιότητες ιδιαίτερα χρήσιμες σε πολυσύχναστους δικτυακούς τόπους.

4.2.2 Ορισμός των clusters

Ο συνδυασμός ιδεών από τα παράλληλα συστήματα, όπως η ομαδοποίηση πόρων υλικού (συχνά γενικού σκοπού), με έννοιες από τα καταναμεμένα συστήματα, όπως το αφαιρετικό μοντέλο περάσματος μηνυμάτων και η χρήση των υπολογιστικών δυνατοτήτων χαμηλού κόστους που προσφέρει η βιομηχανία των προσωπικών υπολογιστών και των δικτύων, οδήγησαν στην ανάπτυξη των clusters. Οι εμπορικές συστάδες σταθμών εργασίας και τα συστήματα της τάξης Beowulf αποτελούν

σήμερα μια ταχύτατα αναπτυσσόμενη κατηγορία υπολογιστικών συστημάτων υψηλής επίδοσης.

Μία εμπορική συστάδα (commodity cluster) είναι ένα τοπικό υπολογιστικό σύστημα που αποτελείται από μία ομάδα ανεξάρτητων υπολογιστών και ένα δίκτυο που τους διασυνδέει. Μια συστάδα είναι τοπική, με την έννοια ότι όλα τα επί μέρους συστήματα που την αποτελούν επιτηρούνται εντός μιας διαχειριστικής περιοχής. Οι υπολογιστικοί αυτοί κόμβοι είναι εμπορικοί και άμεσα διαθέσιμοι (commercial off the self – COTS), και είναι ικανοί να λειτουργούν ανεξάρτητα και αυτόνομα για συνήθεις λειτουργίες. Ο κάθε κόμβος μπορεί να αποτελείται είτε από έναν μικροεπεξεργαστή είτε από πολλούς σε μία συμμετρική πολυεπεξεργαστική σύνθεση (symmetric multiprocessor - SMP). Το διασυνδεδετικό δίκτυο χρησιμοποιεί εμπορικές και άμεσα διαθέσιμες τεχνολογίες δικτύων, τοπικών (local area network - LAN) ή περιοχές συστήματος (systems area network – SAN), οι οποίες απαρτίζουν πολλές ανεξάρτητες δικτυακές δομές ή μία ιεραρχία τέτοιων. Το δίκτυο του cluster ενοποιεί τους υπολογιστικούς του κόμβους και είναι διαχωρισμένο από το εξωτερικό του περιβάλλον.

Ένα cluster μπορεί να είναι ρυθμισμένο έτσι ώστε να εξυπηρετεί διάφορους σκοπούς, όπως υψηλή υπολογιστική επίδοση για την επίλυση ενός προβλήματος, υψηλή χωρητικότητα, διαμεταγωγή για ένα φορτίο διεργασιών, υψηλή διαθεσιμότητα (μέσω πλεονασμού κόμβων) ή υψηλό εύρος ζώνης (μέσω μεγάλου αριθμού δίσκων και καναλιών εισόδου/ εξόδου). Ένα σύστημα της τάξης Beowulf (Beowulf class system) είναι μία συστάδα με κόμβους οι οποίοι είναι προσωπικοί υπολογιστές ή μικροί συμμετρικοί πολυεπεξεργαστικοί προσωπικοί υπολογιστές, ενοποιημένοι με εμπορικά και άμεσα διαθέσιμα δίκτυα και διαθέτουν ένα λειτουργικό σύστημα ανά κόμβο, συνήθως ανοικτού κώδικα και τύπου Unix.

Στην πλειοψηφία των συστημάτων τέτοιου τύπου υπάρχει ένας κόμβος υπεύθυνος για την επαφή με τον υπόλοιπο κόσμο (front end node) μέσω του οποίου πραγματοποιείται και η πρόσβαση στους υπόλοιπους υπολογιστές. Μία πλειάδα (constellation) διαφέρει από μία εμπορική συστάδα στο γεγονός ότι ο αριθμός των επεξεργαστών στους συμμετρικούς πολυεπεξεργαστές των κόμβων υπερβαίνει τον αριθμό των συμμετρικών πολυεπεξεργαστών που αποτελούν το σύστημα, καθώς και στο γεγονός ότι το διασυνδεδετικό τους δίκτυο είναι πιθανά ειδικής τεχνολογίας και σχεδιασμού. Όταν οι διασυνδεδεμένοι υπολογιστές είναι σταθμοί εργασίας, τότε μιλάμε για Δίκτυο Σταθμών Εργασίας (Network Of Workstations – NOW). Τέλος, αναφερόμενοι στην χρήση υπολογιστικών πόρων διαθέσιμων μέσω του διαδικτύου μιλάμε για πλέγματα (grids) υπολογιστών.

4.2.3 Πλεονεκτήματα και μειονεκτήματα των clusters

Λαμβάνοντας υπ' όψιν τις παραδοσιακές προσεγγίσεις από τους χώρους της παράλληλης επεξεργασίας και των παράλληλων αρχιτεκτονικών μπορούμε να εντοπίσουμε αρκετά μειονεκτήματα των clusters. 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 παραμετροποιησιμότητά 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 τις 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.

4.2.4 Cluster of blades

Ένας blade server 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 στηρίζει πολλαπλούς blade servers παρέχει υπηρεσίες, όπως ρεύμα, ψύξη, δικτύωση και διαχείριση.

4.3 Grids

Grid computing είναι ο συνδυασμός των υπολογιστικών πόρων από πολλαπλούς διοικητικούς τομείς που εφαρμόζεται σε ένα κοινό καθήκον, συνήθως σε ένα επιστημονικό, τεχνικό ή επιχειρηματικό πρόβλημα που απαιτεί ένα μεγάλο αριθμό υπολογιστικών κύκλων επεξεργασίας ή την ανάγκη για την επεξεργασία μεγάλου όγκου δεδομένων. Πρόκειται για ένα είδος παράλληλου και κατανεμημένου συστήματος που επιτρέπει την κοινή χρήση, την επιλογή και τη σύνθεση γεωγραφικά κατανεμημένων αυτόνομων πόρων δυναμικά κατά το χρόνο εκτέλεσης, ανάλογα με τη διαθεσιμότητά τους, την ικανότητα, την απόδοση, το κόστος και την ποιότητα των υπηρεσιών για τους χρήστες. Είναι μια κοινή συλλογή αξιόπιστων και αναξιόπιστων πόρων και διαδραστικά επικοινωνούντων ερευνητών διαφόρων εικονικών οργανισμών (γιατροί, βιολόγοι). Αυτό το σύστημα ελέγχει και συντονίζει την ακεραιότητα του grid με την εξισορρόπηση της χρήσης αξιόπιστων και αναξιόπιστων πόρων μεταξύ των συμμετεχόντων παρέχοντας καλύτερη ποιότητα υπηρεσιών. Μπορεί επίσης να συμβολίζεται από μεγάλης κλίμακας cluster computing, καθώς και μία μορφή κατανεμημένου δικτύου παράλληλης επεξεργασίας. Πρόκειται για μια συλλογή από servers που συγκεντρώνονται μαζί για να επιτεθούν σε ένα μόνο πρόβλημα. Το grid computing έχει να κάνει με το διαμοιρασμό, τη σύνθεση, τη φιλοξενία και την προσφορά υπηρεσιών σε όλο τον κόσμο για τα οφέλη της ανθρωπότητας.

Ένα υπολογιστικό grid προσφέρει μια υποδομή από υπολογιστές, software/middleware, ειδικά όργανα, ανθρώπους και αισθητήρες. Το grid συνήθως

κατασκευάζεται σε ένα LAN, WAN ή βασισμένο στο Internet δίκτυο σε περιφερειακή, εθνική ή παγκόσμια κλίμακα. Οι επιχειρήσεις ή οι οργανισμοί παρουσιάζουν τα grids σαν ενσωματωμένους υπολογιστικούς πόρους. Μπορούμε επίσης να τα δούμε σαν εικονικές πλατφόρμες που υποστηρίζουν εικονικούς οργανισμούς. Οι υπολογιστές που χρησιμοποιούνται σε ένα grid είναι κυρίως σταθμοί εργασίας, servers, clusters και υπερυπολογιστές. Οι προσωπικοί υπολογιστές, τα laptops και τα PDAs μπορούν να χρησιμοποιηθούν σα συσκευές πρόσβασης σε ένα grid.

4.4 Cloud computing

Το cloud computing είναι ένας καινούργιος όρος στον κόσμο του υπολογισμού και σηματοδοτεί ένα νέο υπολογιστικό παράδειγμα, ταχέως αναπτυσσόμενο. Η ανάπτυξη αυτή τροφοδοτείται από τις αναδυόμενες υπολογιστικές τεχνολογίες που επιτρέπουν μείωση του κόστους και μεγάλες αποθηκευτικές δυνατότητες. Το cloud computing είναι ένα παράδειγμα υπολογισμού που περιλαμβάνει την εξωτερική ανάθεση των υπολογιστικών πόρων με τις δυνατότητες της κλιμάκωσης των αναλώσιμων πόρων και της on-demand παροχής με ελάχιστο ή καθόλου εκ των προτέρων κόστος για επενδύσεις σε υποδομές. Το cloud computing προσφέρει τα οφέλη του μέσω τριών τύπων υπηρεσιών ή μοντέλων παροχής δηλαδή Infrastructure-as-a-Service (IAAS), Platform-as-a-service (PaaS) και Software-as-a-Service (SaaS). Παρέχει επίσης τις υπηρεσίες του μέσω τεσσάρων μοντέλων ανάπτυξης, δηλαδή δημόσιο cloud, ιδιωτικό cloud, cloud κοινότητας και υβριδικό cloud.

Στα δημόσια clouds η υποδομή είναι προσβάσιμη στο κοινό και διαμοιράζεται με ένα μοντέλο πληρωμής βάσει χρήσης. Οι πόροι του cloud είναι προσβάσιμοι μέσω internet και ο πάροχος είναι υπεύθυνος για την κλιμάκωση και τη διαχείριση της υποδομής. Στο μοντέλο αυτό οι πελάτες μπορούν να επιλέγουν το επίπεδο ασφάλειας και υπηρεσιών.

Στα ιδιωτικά clouds οι πόροι δεν διαμοιράζονται με άγνωστους τρίτους και μπορούν να βρίσκονται είτε εντός των εγκαταστάσεων του οργανισμού του πελάτη ή εξωτερικά. Στο μοντέλο αυτό η ασφάλεια του πελάτη και οι απαιτήσεις συμμόρφωσης δεν επηρεάζονται και το γενικό κοινό δεν έχει πρόσβαση στο ιδιωτικό cloud.

Τα υβριδικά clouds είναι ένα μοντέλο ανάπτυξης που συνδυάζει διαφορετικά clouds, για παράδειγμα δημόσια και ιδιωτικά. Τα διαφορετικά clouds διατηρούν τις ταυτότητές τους, αλλά συνδυάζονται με προτυποποιημένη τεχνολογία. Το γενικό κοινό δεν έχει πρόσβαση στο cloud, αλλά ο οργανισμός χρησιμοποιεί την υποδομή τόσο στο δημόσιο όσο και στο ιδιωτικό cloud.

Στα clouds κοινότητας η υποδομή διαμοιράζεται από πολλαπλούς οργανισμούς και ιδρύματα που έχουν ένα κοινό ενδιαφέρον, όπως απαιτήσεις συμμόρφωσης ή ασφάλειας. Η διαχείριση γίνεται από τον οργανισμό ή από κάποιον τρίτο και εδράζεται είτε εντός των εγκαταστάσεων είτε εκτός. Τόσο το κοινό όσο και οι οργανισμοί που αποτελούν την κοινότητα έχουν πρόσβαση στις υπηρεσίες του cloud.

Clusters	Grids	Clouds
Χαρακτηριστικά: Στενά συνδεδεμένα συστήματα Εικόνα ενιαίου συστήματος Σύστημα κεντρικής διαχείρισης και προγραμματισμού εργασιών	Χαρακτηριστικά: Χαλαρά συνδεδεμένα συστήματα (Αποκέντρωση) Διαφορετικότητα και δυναμικότητα Κατανεμημένη διαχείριση και προγραμματισμός εργασιών	Χαρακτηριστικά: Δυναμική υποδομή υπολογιστικών πόρων Υψηλοκεντρική προσέγγιση Μοντέλο χρήσης βασισμένο στην αυτοεξυπηρέτηση Ελάχιστα ή αυτοδιαχειριζόμενη πλατφόρμα Τιμολόγηση βάσει κατανάλωσης
Μια ομάδα παρόμοιων (ή πανομοιότυπων) υπολογιστών είναι συνδεδεμένοι τοπικά (στην ίδια φυσική τοποθεσία, άμεσα συνδεδεμένοι με συνδέσεις πολύ υψηλής ταχύτητας) για να λειτουργούν σαν ένας ενιαίος υπολογιστής.	Οι υπολογιστές δεν χρειάζεται να είναι στην ίδια φυσική τοποθεσία και μπορούν να λειτουργούν ανεξάρτητα. Για τους άλλους υπολογιστές ο κάθε υπολογιστής στο grid είναι ξεχωριστός.	Οι υπολογιστές δεν χρειάζεται να είναι στην ίδια φυσική τοποθεσία.
Οι υπολογιστές του cluster έχουν όλοι το ίδιο hardware και λειτουργικό σύστημα.	Οι υπολογιστές μπορούν να έχουν διαφορετικά λειτουργικά συστήματα και διαφορετικό hardware.	Η μνήμη, η συσκευή αποθήκευσης και η επικοινωνία δικτύου διαχειρίζονται από το λειτουργικό σύστημα των βασικών φυσικών μονάδων του cloud.
Όλο το σύστημα (όλοι οι κόμβοι) συμπεριφέρεται σαν μια όψη ενιαίου συστήματος και οι πόροι διαχειρίζονται από κεντρικό διαχειριστή πόρων.	Κάθε κόμβος είναι αυτόνομος, δηλαδή έχει το δικό του διαχειριστή πόρων και συμπεριφέρεται σαν ανεξάρτητη οντότητα.	Κάθε κόμβος συμπεριφέρεται σαν ανεξάρτητη οντότητα.

Οι υπολογιστές στο cluster συνήθως περιλαμβάνονται σε μια μοναδική τοποθεσία ή συγκρότημα.	Είναι εκ φύσεως καταναεμημένα σε ένα LAN, μητροπολιτικό ή WAN.	Είναι κυρίως καταναεμημένα σε MAN.
Περισσότεροι από 2 υπολογιστές συνδέονται για να λύσουν ένα πρόβλημα.	Ένα μεγάλο έργο μοιράζεται σε πολλαπλού υπολογιστές για να κάνει χρήση των πόρων τους.	Επιτρέπει πολλαπλές μικρές εφαρμογές να τρέχουν ταυτόχρονα.

Table 1: Διαφορές cluster, grid και cloud computing

5 Υλοποίηση

5.1 Εισαγωγή

Η επίλυση μεγάλων και σύνθετων Μερικών Διαφορικών Εξισώσεων (ΜΔΕ) είναι ένα πρόβλημα που αντιμετωπίζεται κυρίως με τεχνικές διακριτοποίησης πεδίου [80, 81]. 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 [64,96,97]. Εδώ, το πεδίο των ΜΔΕ αποσυντίθεται σε υποπεδία, για λόγους που προέρχονται από την φυσική ή για λόγους παραλληλισμού, και την ίδια στιγμή αρχικές μαντεψιές 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 η σύγκλιση επιτευχθεί.

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 διπλωματική υλοποιούμε μια από τις μεθόδους χαλάρωσης στη διεπαφή, που παρουσιάστηκαν στο κεφάλαιο 3, τη geometric (GEO) contraction based στη FEniCS. Η FEniCS [98] 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) [99] παρουσιάζεται, επίσης. Στη συνέχεια του κεφαλαίου παρουσιάζονται σύντομα η FEniCS, το RabbiMQ και ζητήματα τόσο της σειριακής, όσο και της παράλληλης υλοποίησης.

5.2 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 MDE 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 MDE 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 [98]. 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:

$$-\nabla^2 u(x) = f(x), \quad x \text{ in } \Omega,$$

$$u(x) = u_0(x), \quad x \text{ on } \partial\Omega.$$

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 MDE 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.

Σε δύο χωρικές διαστάσεις με συντεταγμένες x και y , μπορούμε να γράψουμε την εξίσωση Poisson ως:

$$-\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = f(x, y).$$

Η άγνωστη u είναι τώρα μια συνάρτηση δύο μεταβλητών $u(x, y)$, ορισμένη σε ένα δισδιάστατο πεδίο Ω . Για να μπορέσει η FEniCS να επιλύσει το πρόβλημα πρέπει να το εκφράσουμε στην παραμετρική μορφή του. Δηλαδή, βρες τη $u \in V$ τέτοια ώστε:

$$\int_{\Omega} \nabla u \nabla v dx = \int_{\Omega} f v dx \quad \forall v \in \hat{V}.$$

Η ακόλουθη ενιαία σημειογραφία αποδεικνύεται χρήσιμη:

$$\alpha(u, v) = L(v).$$

Στο περιγραφόμενο πρόβλημα ισχύει ότι

$$\alpha(u, v) = \int_{\Omega} \nabla u \nabla v dx,$$

$$L(v) = \int_{\Omega} f v dx.$$

Μέχρι στιγμής στο πρόβλημα-μοντέλο έχουμε ένα γενικό πεδίο Ω και γενικές συναρτήσεις u_0 και f . Για την υλοποίηση στη FEniCS πρέπει να επιλέξουμε συγκεκριμένα Ω , u_0 και f . Η ακριβής λύση είναι:

$$u_e(x, y) = 1 + x^2 + 2y^2$$

σε ένα δισδιάστατο πεδίο. Τα αντίστοιχα Ω , u_0 και f είναι:

$$f(x, y) = -6, \quad u_0(x, y) = u_e(x, y) = 1 + x^2 + 2y^2, \quad \Omega = [0,1] \times [0,1].$$

Το πρόγραμμα σε FEniCS που επιλύει το παραπάνω πρόβλημα είναι το ακόλουθο:

```

"""
FEniCS tutorial demo program: Poisson equation with Dirichlet
conditions.
Simplest example of computation and visualization with FEniCS.

-Laplace(u) = f on the unit square.
u = u0 on the boundary.
u0 = u = 1 + x^2 + 2y^2, f = -6.
"""

from dolfin import *

# Create mesh and define function space
mesh = UnitSquare(6, 4)
#mesh = UnitCube(6, 4, 5)
V = FunctionSpace(mesh, 'Lagrange', 1)

# Define boundary conditions
u0 = Expression('1 + x[0]*x[0] + 2*x[1]*x[1]')

def u0_boundary(x, on_boundary):
    return on_boundary

bc = DirichletBC(V, u0, u0_boundary)

# Define variational problem
u = TrialFunction(V)
v = TestFunction(V)
f = Constant(-6.0)
a = inner(nabla_grad(u), nabla_grad(v))*dx
L = f*v*dx

# Compute solution
u = Function(V)
solve(a == L, u, bc)

# Plot solution and mesh
plot(u)
plot(mesh)

# Dump solution to file in VTK format
file = File('poisson.pvd')
file << u

# Hold plot
interactive()

```

Η πρώτη γραμμή του προγράμματος:

```
from dolfin import *
```

εισάγει τις κλάσεις UnitSquare, FunctionSpace, Function, κλπ. από τη βιβλιοθήκη DOLFIN. Όλα τα FEniCS προγράμματα που επιλύουν ΜΔΕ με τη μέθοδο πεπερασμένων στοιχείων ξεκινούν κανονικά με αυτή τη γραμμή. Το DOLFIN είναι μια βιβλιοθήκη λογισμικού με αποδοτικές και βολικές C++ κλάσεις για υπολογισμό πεπερασμένων στοιχείων και το dolfin είναι ένα πακέτο Python που παρέχει πρόσβαση σε αυτήν τη C++ βιβλιοθήκη από προγράμματα Python. Μπορεί κανείς να σκεφτεί την FEniCS σαν μια ομπρέλα για ένα σύνολο υπολογιστικών συστατικών, όπου το DOLFIN είναι ένα σημαντικό συστατικό για τη συγγραφή προγραμμάτων πεπερασμένων στοιχείων. Η πρόταση `from dolfin import *` εισάγει και άλλα συστατικά επίσης.

Η γραμμή:

```
mesh = UnitSquare(6, 4)
```

ορίζει ένα κανονικό πλέγμα πεπερασμένων στοιχείων πάνω στο μοναδιαίο τετράγωνο $[0,1] \times [0,1]$. Το πλέγμα αποτελείται από κελιά, που είναι τρίγωνα με ευθείες πλευρές. Οι παράμετροι 6 και 4 δηλώνουν ότι το τετράγωνο διαιρείται αρχικά σε 6 x 4 ορθογώνια και στη συνέχεια κάθε ορθογώνιο διαιρείται σε 2 τρίγωνα. Ο συνολικός αριθμός τριγώνων με αυτό το τρόπο γίνεται 48. Ο συνολικός αριθμός κορυφών στο πλέγμα είναι 7 x 5=35. Το DOLFIN προσφέρει κάποιες κλάσεις για την δημιουργία πλεγμάτων σε πολύ απλές γεωμετρίες. Για πεδία πιο πολύπλοκου σχήματος είναι απαραίτητο ένα εξωτερικό πρόγραμμα προεπεξεργασίας για τη δημιουργία του πλέγματος. Σε αυτήν την περίπτωση το FEniCS πρόγραμμα θα διαβάσει το πλέγμα από ένα αρχείο.

Εφόσον έχουμε το πλέγμα, ορίζουμε ένα διακριτό χώρο συνάρτησης V πάνω στο πλέγμα:

```
V = FunctionSpace(mesh, 'Lagrange', 1)
```

Το δεύτερο όρισμα αντιστοιχεί στο είδος του στοιχείου, ενώ το τρίτο όρισμα είναι ο βαθμός των συναρτήσεων βάσης στο στοιχείο.

Το είδος του στοιχείου εδώ είναι “Lagrange”, υπονοώντας τη τυπική οικογένεια στοιχείων Lagrange. Ορίζοντας βαθμό ίσο με 1 παίρνουμε το τυπικό γραμμικό στοιχείο Lagrange που είναι ένα τρίγωνο με κόμβους στις τρεις κορυφές. Η υπολογισμένη u θα είναι συνεχής και γραμμικά μεταβαλλόμενη στο x και y σε κάθε κελί του πλέγματος. Αυξάνοντας την τρίτη παράμετρο του FunctionSpace παίρνουμε μεγαλύτερου βαθμού πολυωνυμικές προσεγγίσεις σε κάθε κελί.

Στη συνέχεια ορίζουμε ένα κοινό χώρο V για τη δοκιμαστική συνάρτηση και τη συνάρτηση ελέγχου.

```
u = TrialFunction(V)
v = TestFunction(V)
```

Το επόμενο βήμα είναι να ορίσουμε την οριακή συνθήκη $u = u_0$ on $\partial\Omega$. Αυτό γίνεται με τη γραμμή:

```
bc = DirichletBC(V, u0, u0_boundary)
```

όπου το $u0$ είναι μια μεταβλητή που κρατάει τις τιμές του u_0 και το $u0_boundary$ είναι μια συνάρτηση που περιγράφει αν ένα σημείο βρίσκεται στο όριο όπου ορίζεται η u .

Οριακές συνθήκες του τύπου $u = u_0$ είναι γνωστές σαν Dirichlet συνθήκες και επίσης σαν απαραίτητες οριακές συνθήκες στο πλαίσιο των πεπερασμένων στοιχείων. Κανονικά η DOLFIN κλάση που συγκρατεί τις πληροφορίες για τις Dirichlet οριακές συνθήκες είναι η DirichletBC.

Η μεταβλητή $u0$ αναφέρεται σε ένα αντικείμενο έκφρασης που χρησιμοποιείται για να αναπαραστήσει μια μαθηματική συνάρτηση. Κατασκευάζεται ως εξής:

```
u0 = Expression(formula)
```

Εδώ το `formula` είναι μια ακολουθία που περιέχει τη μαθηματική έκφραση. Γράφεται με το συντακτικό της C++. Οι ανεξάρτητες μεταβλητές στην έκφραση της συνάρτησης θεωρείται ότι είναι διαθέσιμες σε ένα διάνυσμα σημείων x όπου το πρώτο στοιχείο $x[0]$ αντιστοιχεί στην x συντεταγμένη, το δεύτερο στοιχείο $x[1]$ στη y συντεταγμένη και το τρίτο $x[2]$ στη z . Με την επιλογή μας $u_0(x, y) = 1 + x^2 + 2y^2$, η ακολουθία της φόρμουλας πρέπει να γραφεί ως `1+x[0]*x[0]+2*x[1]*x[1]`:

```
u0 = Expression('1 + x[0]*x[0] + 2*x[1]*x[1]')
```

Η πληροφορία για το που θα εφαρμόσουμε τη συνάρτηση u_0 σαν οριακή συνθήκη κωδικοποιείται σε μια συνάρτηση `u0_boundary`:

```
def u0_boundary(x, on_boundary):  
    return on_boundary
```

Μια συνάρτηση σαν τη `u0_boundary` για να ορίσει το όριο πρέπει να επιστρέφει μια λογική τιμή: Αληθές αν το δοθέν σημείο x βρίσκεται στο Dirichlet όριο και ψευδές αλλιώς. Το όρισμα `on_boundary` είναι Αληθές αν το x είναι πάνω στο φυσικό όριο του πλέγματος, οπότε στην περίπτωση μας όπου πρέπει να επιστρέψουμε Αληθές για όλα τα σημεία στο όριο μπορούμε απλά να επιστρέψουμε την τιμή που μας παρέχει το `on_boundary`. Η συνάρτηση `u0_boundary` θα κληθεί για κάθε διακριτό σημείο του πλέγματος.

Πριν ορίσουμε τα $\alpha(u, v), L(v)$, πρέπει να ορίσουμε τη συνάρτηση f :

```
f = Expression('-6')
```

Όταν η f είναι σταθερή στο πεδίο, μπορεί πιο αποδοτικά να αναπαρασταθεί σαν ένα αντικείμενο `Constant`:

```
f = Constant(-6.0)
```

Τώρα έχουμε όλα τα αντικείμενα που χρειαζόμαστε ώστε να ορίσουμε τα $\alpha(u, v), L(v)$ του προβλήματος:

```
a = inner(nabla_grad(u), nabla_grad(v))*dx  
L = f*v*dx
```

Στην ουσία αυτές οι δύο γραμμές ορίζουν το πρόβλημα ΜΔΕ που πρέπει να λυθεί. Παρατηρούμε την πολύ στενή αντιστοιχία ανάμεσα στο συντακτικό της Python και των μαθηματικών τύπων $\nabla u \nabla v dx$ και $f v dx$. Αυτό είναι ένα πλεονέκτημα κλειδί της FEniCS.

Έχοντας ορίσει τα a και L και πληροφορίες για τις Dirichlet οριακές συνθήκες στο bc μπορούμε να υπολογίσουμε τη λύση, μια συνάρτηση πεπερασμένων στοιχείων ως εξής:

```
u = Function(V)
solve(a == L, u, bc)
```

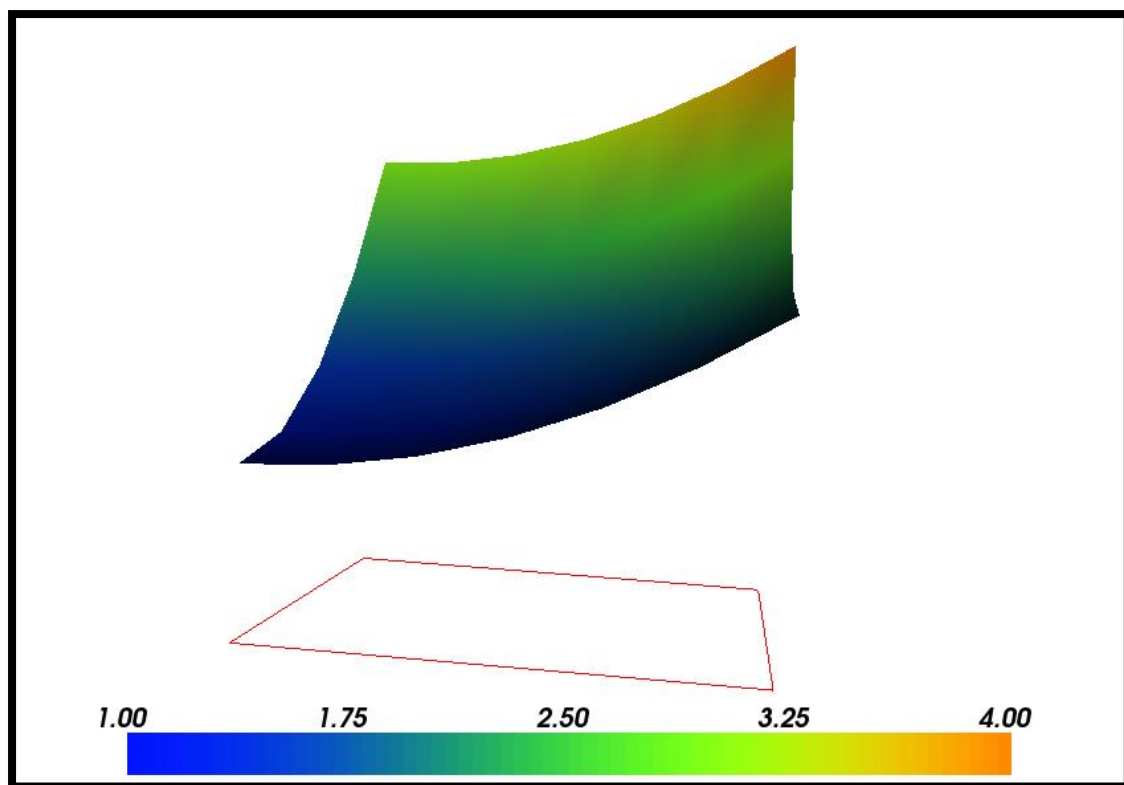
Ο πιο απλός τρόπος να δούμε γρήγορα τη u και το πλέγμα είναι:

```
plot(u)
plot(mesh)
interactive()
```

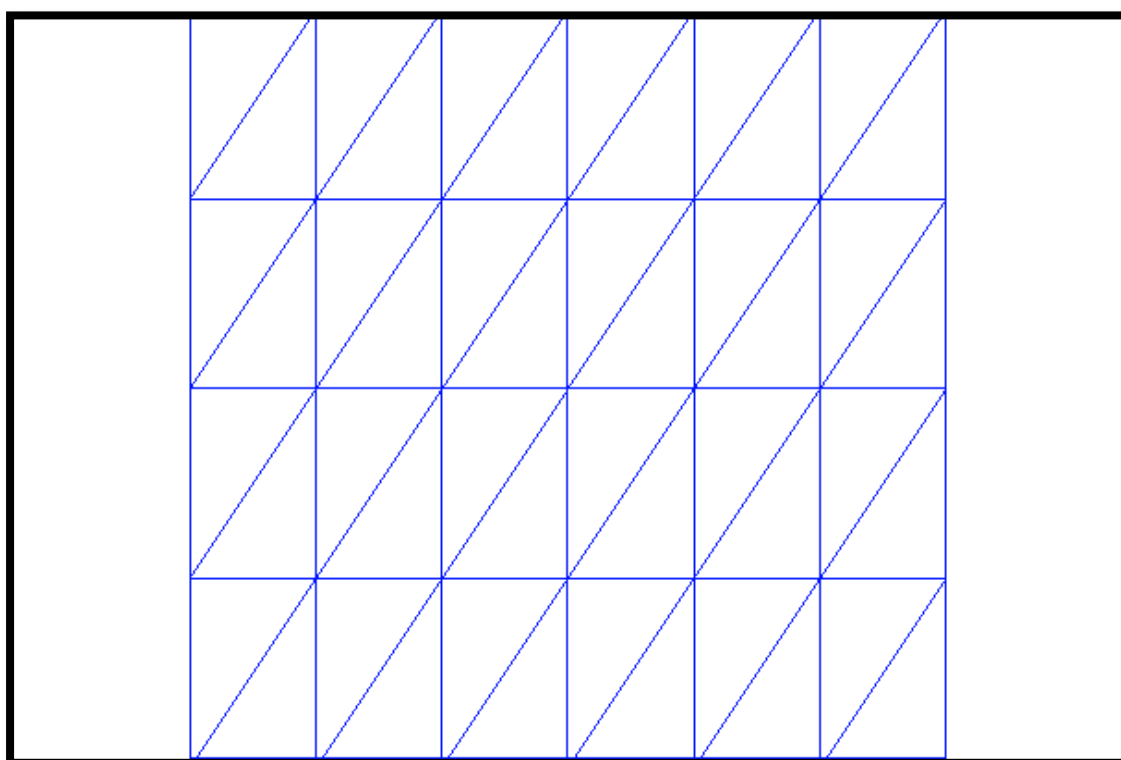
Η κλήση `interactive()` είναι απαραίτητη ώστε το γράφημα να παραμείνει στην οθόνη.

Μπορούμε επίσης να μεταφέρουμε την υπολογισμένη λύση σε ένα αρχείο, π.χ. VTK μορφής:

```
file = File('poisson.pvd')
file << u
```



Εικόνα 3: Η γραφική αναπαράσταση της λύσης u .



Εικόνα 4: Η γραφική αναπαράσταση του πλέγματος.

5.3 RabbitMQ

Για την παράλληλη υλοποίηση στην πλατφόρμα FEniCS χρησιμοποιήθηκε το RabbitMQ [99]. 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 διπλωματική 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.

Η κλάση του πελάτη έχει μια μέθοδο call που στέλνει ένα αίτημα RPC και μπλοκάρει μέχρι να λάβει την απάντηση.

```
fibonacci_rpc = FibonacciRpcClient()
result = fibonacci_rpc.call(4)
print "fib(4) is %r" % (result,)
```

Ο server απαντάει όταν λάβει ένα μήνυμα αιτήματος με ένα μήνυμα απάντησης. Για να λάβει την απάντηση ο πελάτης χρειάζεται να στείλει μία διεύθυνση ουράς επανάκλησης (callback) με το αίτημα.

```
result = channel.queue_declare(exclusive=True)
callback_queue = result.method.queue

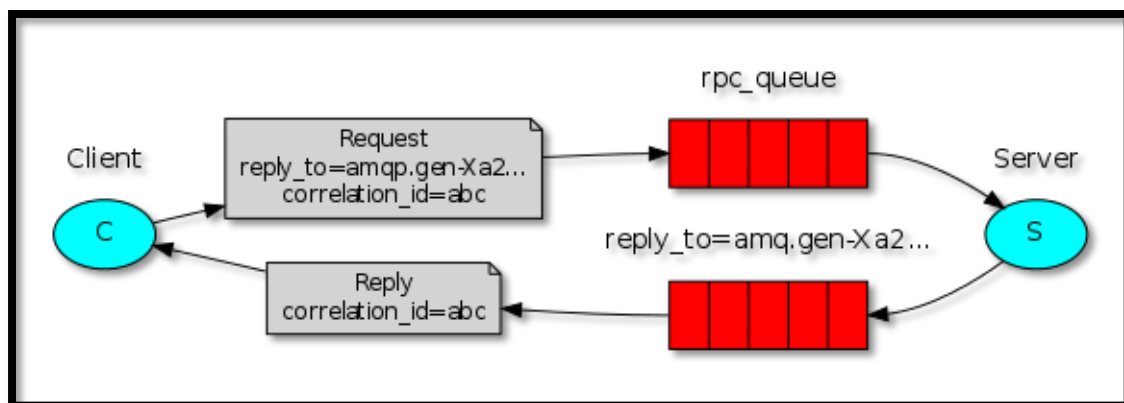
channel.basic_publish(exchange='',
                      routing_key='rpc_queue',
                      properties=pika.BasicProperties(
                          reply_to = callback_queue,
                      ),
                      body=request)

# ... and some code to read a response message from the
# callback_queue ...
```

Το πρωτόκολλο AMQP ορίζει ένα σύνολο 14 ιδιοτήτων που αποστέλλονται μαζί με το μήνυμα. Οι κυριότερες είναι

- `delivery_mode`: Χαρακτηρίζει ένα μήνυμα ως επίμονο (με την τιμή 2) ή περιοδικό (με οποιαδήποτε άλλη τιμή).
- `content_type`: Περιγράφει την κωδικοποίηση. Μια συχνή πρακτική είναι να το θέτουμε σε `application/json`.
- `reply_to`: Συνήθως χρησιμοποιείται για να ονοματίσει μια ουρά επανάκλησης.
- `correlation_id`: Χρήσιμο για τη συσχέτιση RPC απαντήσεων με αιτήματα.

Αν δημιουργήσουμε μία μόνο ουρά επανάκλησης ανά πελάτη, δεν είναι ξεκάθαρο αν λάβουμε μια απάντηση να καταλάβουμε σε ποιο αίτημα ανήκει αυτή η απάντηση. Τότε είναι που χρησιμοποιείται η ιδιότητα `correlation_id`, που παίρνει μια μοναδική τιμή για κάθε αίτημα. Οπότε, όταν ληφθεί το μήνυμα στην ουρά επανάκλησης με βάση το `correlation_id` αντιστοιχίζουμε την απάντηση με το αίτημα.



Εικόνα 5: Περίληψη RPC συστήματος.

Ο τρόπος λειτουργίας του RPC μπορεί να περιγραφεί ως εξής:

- Όταν ξεκινάει ο πελάτης, δημιουργεί μία ανώνυμη αποκλειστική ουρά επανάκλησης.
- Για ένα αίτημα RPC, ο πελάτης στέλνει ένα μήνυμα με δύο ιδιότητες: το `reply_to` που έχει την ουρά επανάκλησης και το `correlation_id` που έχει μια μοναδική τιμή για κάθε αίτημα.
- Το αίτημα στέλνεται σε μια ουρά `rpc_queue`.
- Ο server περιμένει για αιτήματα σε αυτήν την ουρά. Όταν εμφανιστεί ένα αίτημα, κάνει τη δουλειά και στέλνει ένα μήνυμα με το αποτέλεσμα πίσω στον πελάτη, χρησιμοποιώντας την ουρά από το πεδίο `reply_to`.

- Ο πελάτης περιμένει για δεδομένα στην ουρά επανάκλησης. Όταν εμφανιστεί ένα μήνυμα, ελέγχει την ιδιότητα `correlation_id`. Αν αντιστοιχεί στη τιμή από το αίτημα, επιστρέφει την απάντηση στην εφαρμογή.

Οπότε, ο κώδικας για τον server είναι:

```
1#!/usr/bin/env python
2import pika
3
4connection = pika.BlockingConnection(pika.ConnectionParameters(
5    host='localhost'))
6
7channel = connection.channel()
8
9channel.queue_declare(queue='rpc_queue')
10
11def fib(n):
12    if n == 0:
13        return 0
14    elif n == 1:
15        return 1
16    else:
17        return fib(n-1) + fib(n-2)
18
19def on_request(ch, method, props, body):
20    n = int(body)
21
22    print " [.] fib(%s)" % (n,)
23    response = fib(n)
24
25    ch.basic_publish(exchange='',
26                    routing_key=props.reply_to,
27                    properties=pika.BasicProperties(
28                        correlation_id = \
29                            props.correlation_id),
30                    body=str(response))
31    ch.basic_ack(delivery_tag = method.delivery_tag)
32
33channel.basic_qos(prefetch_count=1)
34channel.basic_consume(on_request, queue='rpc_queue')
35
36print " [x] Awaiting RPC requests"
```

και ο κώδικας για τον client:

```
1#!/usr/bin/env python
2import pika
3import uuid
4
5class FibonacciRpcClient(object):
6    def __init__(self):
7        self.connection=pika.BlockingConnection(pika.ConnectionParameters(
8            host='localhost'))
9
10        self.channel = self.connection.channel()
11
12        result = self.channel.queue_declare(exclusive=True)
13        self.callback_queue = result.method.queue
14
15        self.channel.basic_consume(self.on_response, no_ack=True,
16                                    queue=self.callback_queue)
17
18    def on_response(self, ch, method, props, body):
19        if self.corr_id == props.correlation_id:
20            self.response = body
21
22    def call(self, n):
23        self.response = None
24        self.corr_id = str(uuid.uuid4())
25        self.channel.basic_publish(exchange='',
26                                   routing_key='rpc_queue',
27                                   properties=pika.BasicProperties(
28                                       reply_to = self.callback_queue,
29                                       correlation_id = self.corr_id,
30                                   ),
31                                   body=str(n))
32
33        while self.response is None:
34            self.connection.process_data_events()
35
36        return int(self.response)
37
38fibonacci_rpc = FibonacciRpcClient()
39
40print " [x] Requesting fib(30)"
41response = fibonacci_rpc.call(30)
42print " [.] Got %r" % (response,)
```

5.4 Σειριακή Υλοποίηση

Η geometric (GEO) contraction based μέθοδος υλοποιείται σαν ένα FEniCS πρόγραμμα γραμμένο στην προγραμματιστική γλώσσα Python. Η βιβλιοθήκη DOLFIN [100] 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 μαντεψιές 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. Οι ανανεωμένες τιμές σε ένα σημείο της διεπαφής x υπολογίζονται ως εξής:

$$u^{(k+1)}(x) = u^{(k)}(x) - \rho \left(\frac{\partial u_L^{(k)}(x)}{\partial n} - \frac{\partial u_R^{(k)}(x)}{\partial n} \right), k = 1, 2, \dots$$

όπου k είναι η επανάληψη, u είναι η υπολογισμένη λύση στο σημείο της διεπαφής x , $\frac{\partial u_L^{(k)}(x)}{\partial n}$, $-\frac{\partial u_R^{(k)}(x)}{\partial n}$ είναι οι τιμές των προς τα έξω κανονικών παραγώγων στα δύο γειτονικά υποπεδία και ρ είναι η παράμετρος χαλάρωσης που χρησιμοποιείται για να επιταχύνει τη σύγκλιση. Μια νέα επανάληψη ξεκινάει με το που περαστούν οι χαλαρωμένες τιμές των διεπαφών πίσω στα υποπεδία.

Πιο αναλυτικά, τμήματα του FEniCS προγράμματος που υλοποιεί τα παραπάνω παρουσιάζεται στη συνέχεια.

Αρχικά γίνεται η δημιουργία των υποπεδίων. Ενδεικτικά παρουσιάζουμε τη δημιουργία του αριστερού υποπεδίου για το ομοιόμορφο πρόβλημα και για την περίπτωση, όπου $h = 0.5$.

```
meshdown = RectangleMesh(0, 0, 0.3333, 2, 7, 40, 'left')  
Vdown = FunctionSpace(meshdown, 'Lagrange', 1)
```

Στη συνέχεια ορίζουμε τις οριακές συνθήκες και τις συνθήκες των διεπαφών. Ενδεικτικά, η πρώτη (αριστερή) διεπαφή:

```
int1_expr = Expression('4135.5*x[1]-6077.6')
class Interface1(SubDomain):
    def inside(self, x, on_boundary):
        return on_boundary and abs(x[0] - 0.3333) < tol and
abs(x[1] - 2) > tol and abs(x[1] - 1.5) > tol
Gamma_4_left = DirichletBC(Vdown, int1_expr, Interface1())
```

Ακολουθεί ο ορισμός του παραμετρικού προβλήματος. Πάλι για το αριστερό υποπεδίο του ομοιόμορφου προβλήματος:

```
u = TrialFunction(Vdown)
v = TestFunction(Vdown)
f = interpolate(f_expr, Vdown)
a = (inner(nabla_grad(u), nabla_grad(v))+g2*inner(u, v))*dx
L = f*v*dx
```

Και στη συνέχεια ο υπολογισμός της λύσης:

```
u = Function(Vdown)
solve(a == L, u, bcsdown)
```

Και της παραγώγου:

```
V_g = VectorFunctionSpace(meshdown, 'Lagrange', 1)
v = TestFunction(V_g)
w = TrialFunction(V_g)
a = inner(w, v)*dx
L = inner(grad(u), v)*dx
grad_u = Function(V_g)
solve(a == L, grad_u)
```

Παρουσιάζεται ο τρόπος με τον οποίο λαμβάνουμε τις τιμές της λύσης και της παραγώγου στα σημεία των διεπαφών:

```
vertex_to_dof_map =
Vdown.dofmap().vertex_to_dof_map(meshdown)

dof_to_vertex_map =
Vdown.dofmap().dof_to_vertex_map(meshdown)

m = u.vector()

dofs_at_vertices = m[dof_to_vertex_map]

coor = meshdown.coordinates()

grad_u_x, grad_u_y = grad_u.split(deepcopy=True)

for vdown in vertices(meshdown):
    ldown = coor[vdown.index()]
    ldown[0], ldown[1], dofs_at_vertices[vdown.index()],
        dofs_at_vertices_g_x_down[vdown.index()],
        dofs_at_vertices_g_y_down[vdown.index()]
```

Και ακολουθεί ο τρόπος που τις ξαναθέτουμε, αφού τις επεξεργαστούμε:

```
downbound=(dofs_at_vertices[vdown.index()]+dofs_at_vertices_up[v  
up.index()])/2+omega*(-  
dofs_at_vertices_g_x_down[vdown.index()]+dofs_at_vertices_g_x_u  
p[vup.index()])  
  
upbound=(dofs_at_vertices[vdown.index()]+dofs_at_vertices_up[vup  
.index()])/2+omega*(-  
dofs_at_vertices_g_x_down[vdown.index()]+dofs_at_vertices_g_x_u  
p[vup.index()])  
  
u.vector().set_local(dofs_at_vertices.array()[vertex_to_dof_map])  
downbound=Function(Vdown)  
downbound.assign(u)
```

Γράφουμε τα αποτελέσματα σε αρχεία ως εξής:

```
import numpy as np  
  
mattxt3=file('dom1.txt', 'w')  
mattxt3.write('%%iteration x y ul ulx uly \n')  
for vdown in vertices(meshdown):  
    ldown = coor[vdown.index()]  
    mattxt3=file('dom1.txt', 'a')  
    np.savetxt(mattxt3, np.c_[iter, ldown[0], ldown[1],  
dofs_at_vertices[vdown.index()],  
dofs_at_vertices_g_x_down[vdown.index()],  
dofs_at_vertices_g_y_down[vdown.index()]], delimiter=' ', fmt='%d  
%f %f %f %f %f', newline=os.linesep)
```

Και σχεδιάζουμε γραφικές παραστάσεις:

```
import matplotlib.pyplot as plt  
plt.plot(uupsto175)  
plt.ylabel('u [0.3333,1.75]')  
plt.xlabel('iterations')  
plt.show()
```

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 RabbitMQ 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.

Άρα, το μεσαίο πεδίο το επιλύει ένας client και τα άλλα δύο ένας server το καθένα. Αυτοί οι servers χειρίζονται και τις διεπαφές. Τμήματα του κώδικα ακολουθούν ενδεικτικά.

Ο client αρχικοποιεί την επικοινωνία:

```
def __init__(self):  
    self.credentials = pika.PlainCredentials('test', 'test')  
  
    self.connection =  
pika.BlockingConnection(pika.ConnectionParameters('150.140.139.89'  
,5672,'/',self.credentials))  
  
    self.channel = self.connection.channel()  
  
    result=self.channel.queue_declare(exclusive=True)  
  
    self.callback_queue = result.method.queue  
  
    self.channel.basic_consume(self.on_response, no_ack=True,  
                                queue=self.callback_queue)  
  
    self.connection2=  
pika.BlockingConnection(pika.ConnectionParameters('150.140.139.89'  
,5672,'/',self.credentials))  
  
    self.channel2 = self.connection2.channel()  
  
    result2=self.channel2.queue_declare(exclusive=True)  
  
    self.callback_queue2 = result2.method.queue  
  
    self.channel2.basic_consume(self.on_response, no_ack=True,  
                                queue=self.callback_queue2)
```

Περιμένει τις απαντήσεις από τους servers:

```
def on_response(self, ch, method, props, body):  
    #print "on_response result ch=%s guess=%s" %  
    (ch,self.channel)  
  
    if ch == self.channel:  
        self.response = body  
  
    else:  
        self.response2 = body
```

Αφού πρώτα τους στείλει τις τιμές στις διεπαφές που υπολόγισε επιλύοντας το πρόβλημα στην εκάστοτε επανάληψη:

```
self.response = None  
  
self.corr_id = str(uuid.uuid4())  
  
self.channel.basic_publish(exchange="",  
                            routing_key='hello',  
                            properties=pika.BasicProperties(  
                                reply_to =  
self.callback_queue,  
                                correlation_id = self.corr_id,  
                                ),  
                            body=inter2)
```

Αντίστοιχα ο κάθε server επιλύει το πρόβλημά του, λαμβάνει από τον client τη λύση και την παράγωγο που υπολόγισε στη διεπαφή και υπολογίζει τις καινούργιες τιμές στη διεπαφή:

```
response=(interfaceup+interfaced)/2+omega*(-  
gradientd+gradientup)
```

Το αποτέλεσμα αυτό το χρησιμοποιεί για την επόμενη επανάληψή του. Το στέλνει, όμως και στον client για να προχωρήσει και αυτός με την επόμενη επανάληψη του.

```
ch.basic_publish(exchange="",
                  routing_key=props.reply_to,
                  properties=pika.BasicProperties(correlation_id = \
                                                    props.correlation_id),
                  body=json.dumps(response.tolist()))
ch.basic_ack(delivery_tag = method.delivery_tag)
```


6 Πειράματα

6.1 Εισαγωγή

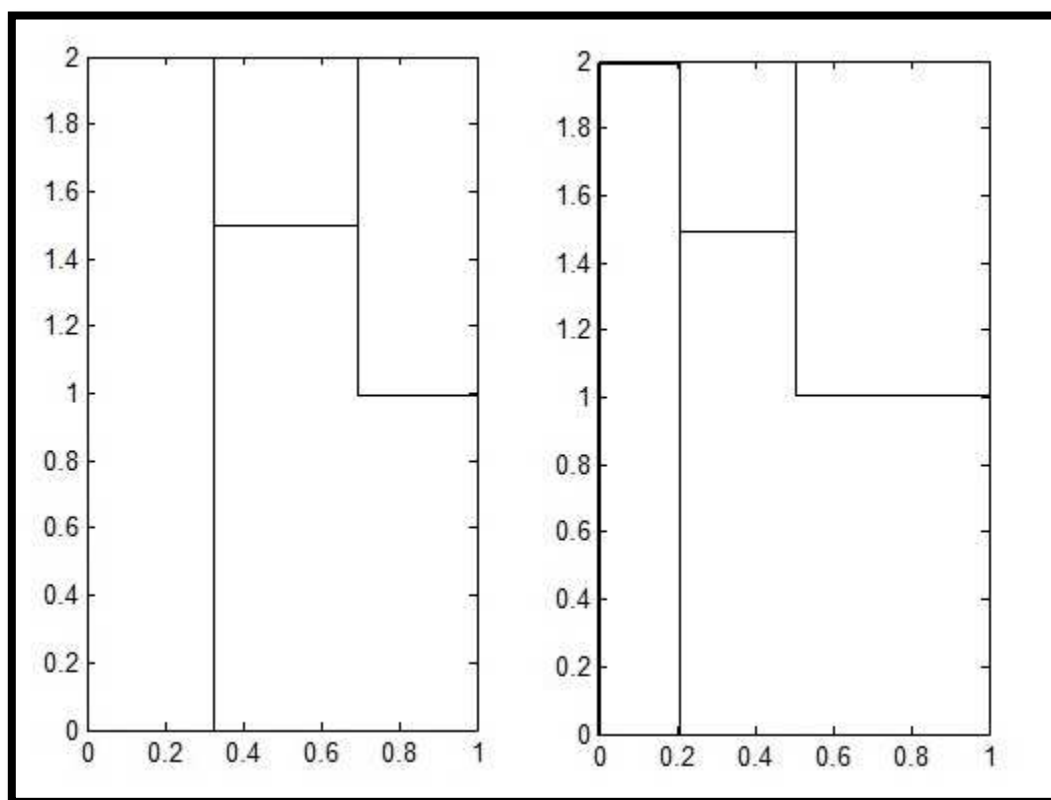
Για την εξέταση της ορθότητας και της απόδοσης της υλοποίησης, χρησιμοποιείται το ακόλουθο ελλειπτικό πρόβλημα:

$$\begin{aligned}Lu(x, y) &\equiv \nabla u(x, y) + \gamma^2 u(x, y) = f(x, y), \quad (x, y) \in \Omega \\ u(x, y) &= u^b(x, y), \quad (x, y) \in \Omega\end{aligned}$$

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 3.



Εικόνα 6: Τα πεδία του ομοιόμορφου (αριστερά) και του ανομοιόμορφου (δεξιά) προβλήματος

6.2 Πειράματα

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 2.

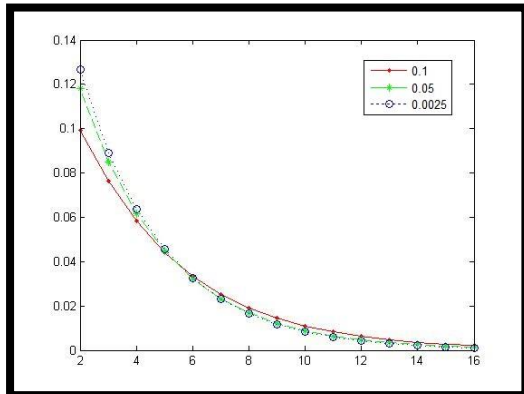
Ομοιόμορφο πρόβλημα				
Περίπτωση	h	Αριστερό υποπεδίο	Μεσαίο υποπεδίο	Δεξί υποπεδίο
Π1	0.1	4x21	4x6	4x11
Π2	0.05	8x41	8x11	8x21
Π3	0.025	14x81	14x21	14x41
Π4	0.0125	28x161	28x41	28x81
Π5	0.00625	55x321	55x81	55x161
Π6	0.003125	108x641	108x161	108x321
Π7	0.0015625	214x1281	214x321	214x641
Ανομοιόμορφο 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 2: 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

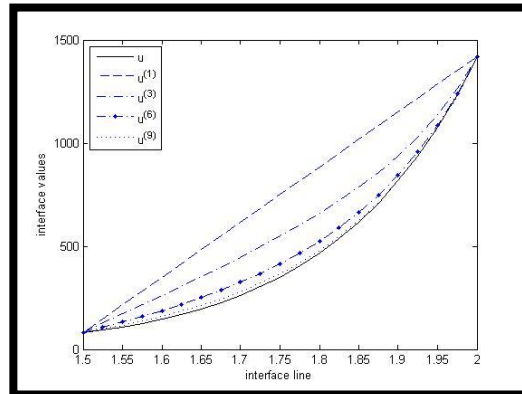
Ο αριθμός των σημείων των διεπαφών σε κάθε περίπτωση ισούται με τον αριθμό των σημείων της y διάστασης του μεσαίου υποπεδίου. Αυξάνεται, δηλαδή, από τα 6 έως τα 321 σημεία τόσο για το ομοιόμορφο, όσο και για το ανομοιόμορφο πρόβλημα. Στην ομοιόμορφη περίπτωση το αριστερό υποπεδίο είναι το μεγαλύτερο πρόβλημα από τα τρία με σημαντική διαφορά από τα υπόλοιπα. Στην ανομοιόμορφη περίπτωση το δεξί υποπεδίο είναι αυτό με το βαρύτερο έργο. Οι διεπαφές έχουν όλες ίσο αριθμό σημείων και άρα απαιτούν ίσο φόρτο εργασίας για την επεξεργασία τους.

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 $h = 0.05$.



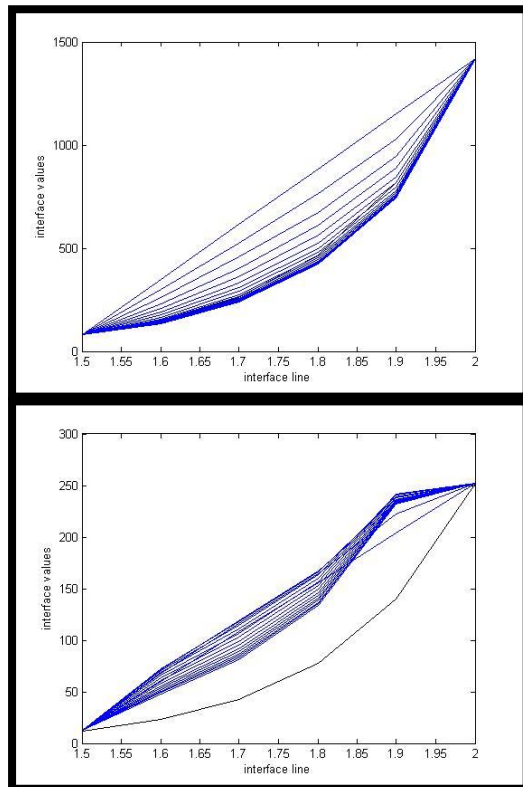
Εικόνα 7: Η νόρμα μεγίστου της σχετικής διαφοράς διαδοχικών επαναλήψεων για τη διεπαφή $x_1=1/3$ του ομοιόμορφου προβλήματος για $h=0.1, 0.05, 0.025$.



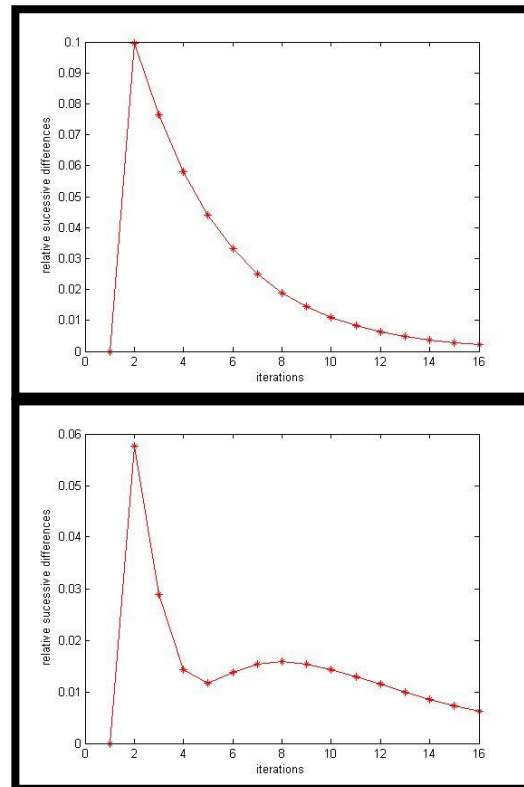
Εικόνα 8: Η ακριβής και οι υπολογισμένες λύσεις για το ομοιόμορφο πρόβλημα και για τη διεπαφή $x_1=1/3$ για τις επαναλήψεις 1, 3, 6, 10 για $h=0.05$.

Στη συνέχεια παρουσιάζονται αναλυτικά τα αποτελέσματα που προκύπτουν για όλες τις διεπαφές και για τα δύο προβλήματα για κάθε μία από τις περιπτώσεις Π1, Π2, Π3. Οι περιπτώσεις ελέγχου επιλέχτηκαν για να διερευνηθεί η συμπεριφορά της παράλληλης υλοποίησης και όχι για λόγους ακρίβειας, οπότε γι' αυτό το λόγο δεν παρουσιάζουμε εδώ τις γραφικές παραστάσεις των περιπτώσεων Π4, Π5, Π6 και Π7.

Οι γραφικές παραστάσεις για την περίπτωση Π1 και το ομοιόμορφο πρόβλημα είναι οι ακόλουθες:

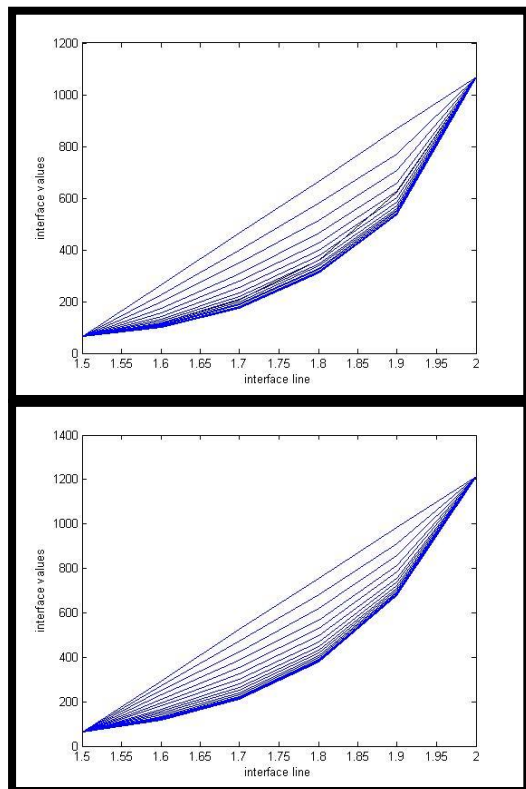


Εικόνα 9: Η ακριβής και οι υπολογισμένες λύσεις – ομοιόμορφο πρόβλημα, περίπτωση Π1, διεπαφή 1 (πάνω) και διεπαφή 2 (κάτω).

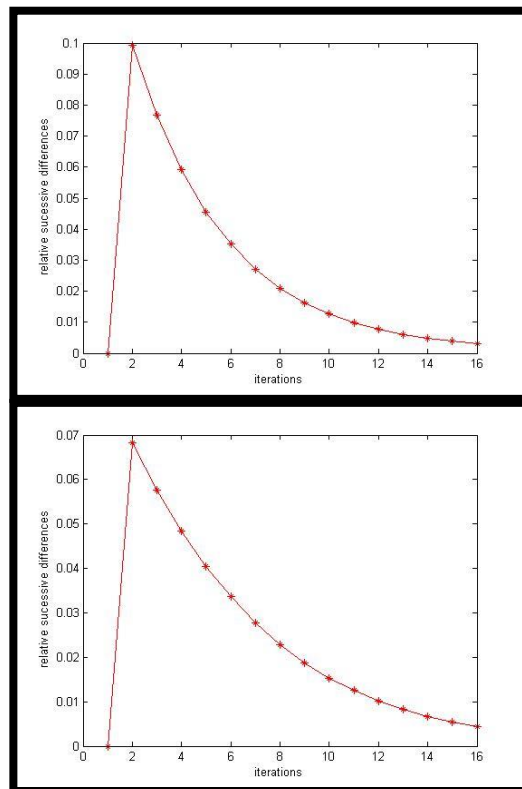


Εικόνα 10: Σχετικές διαφορές διαδοχικών επαναλήψεων - ομοιόμορφο πρόβλημα, περίπτωση Π1, διεπαφή 1 (πάνω) και διεπαφή 2 (κάτω)

Αντίστοιχα για το ανομοιόμορφο πρόβλημα:

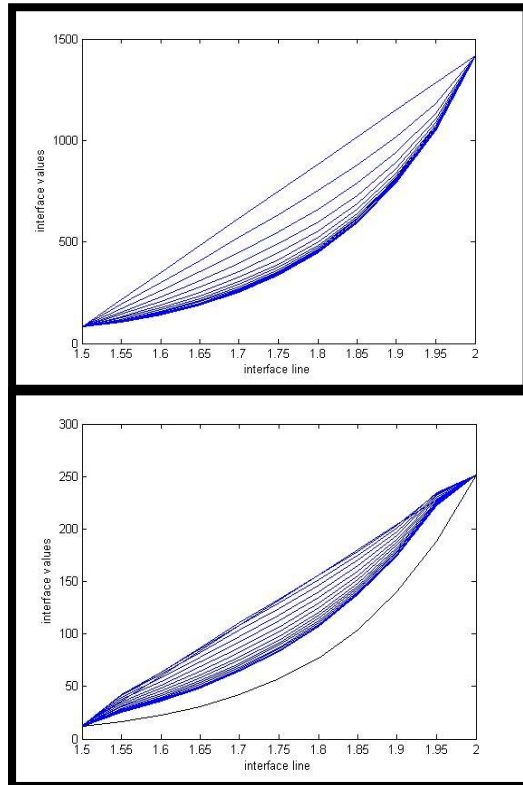


Εικόνα 11: Η ακριβής και οι υπολογισμένες λύσεις – ανομοιόμορφο πρόβλημα, περίπτωση Π1, διεπαφή 1 (πάνω) και διεπαφή 2 (κάτω).

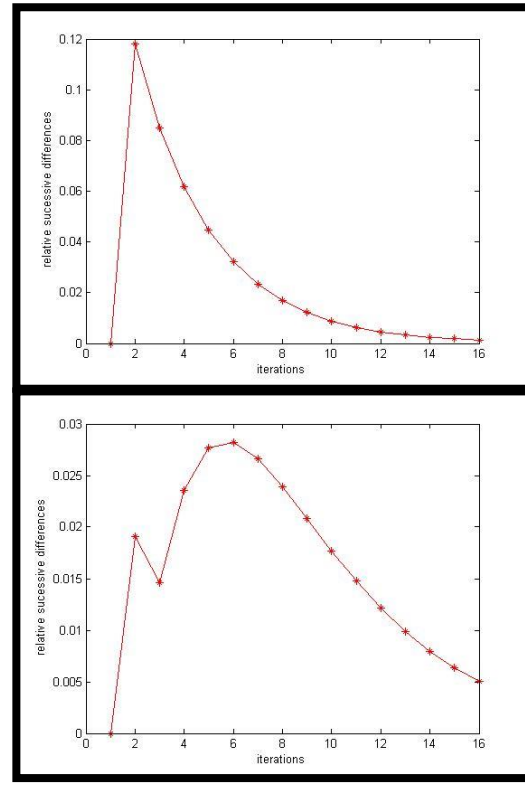


Εικόνα 12: Σχετικές διαφορές διαδοχικών επαναλήψεων - ανομοιόμορφο πρόβλημα, περίπτωση Π1, διεπαφή 1 (πάνω) και διεπαφή 2 (κάτω).

Ακολουθούν οι γραφικές παραστάσεις για την περίπτωση Π2. Αρχικά για το ομοιόμορφο πρόβλημα:

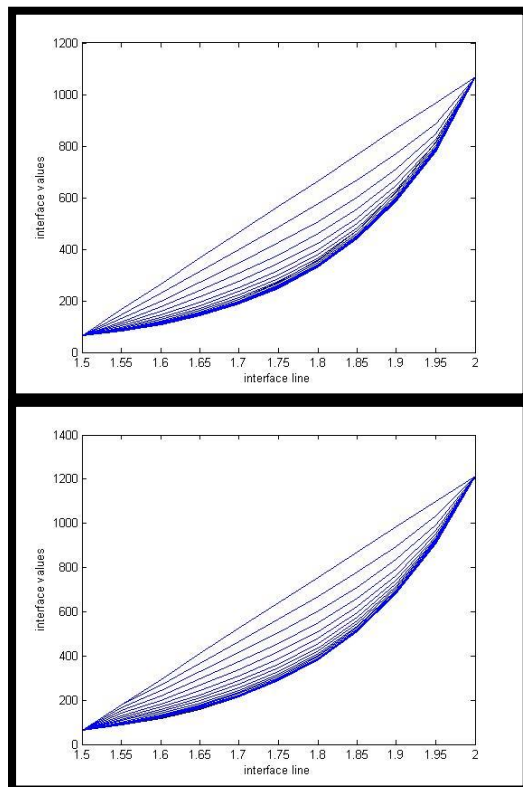


Εικόνα 13: Η ακριβής και οι υπολογισμένες λύσεις – ομοιόμορφο πρόβλημα, περίπτωση Π2, διεπαφή 1 (πάνω) και διεπαφή 2 (κάτω).

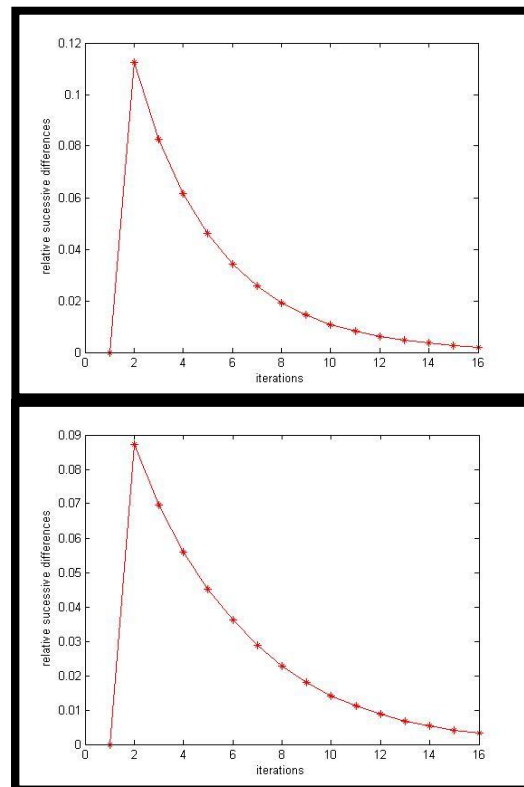


Εικόνα 14: Σχετικές διαφορές διαδοχικών επαναλήψεων - ομοιόμορφο πρόβλημα, περίπτωση Π2, διεπαφή 1 (πάνω) και διεπαφή 2 (κάτω).

Και για το ανομοιόμορφο πρόβλημα:

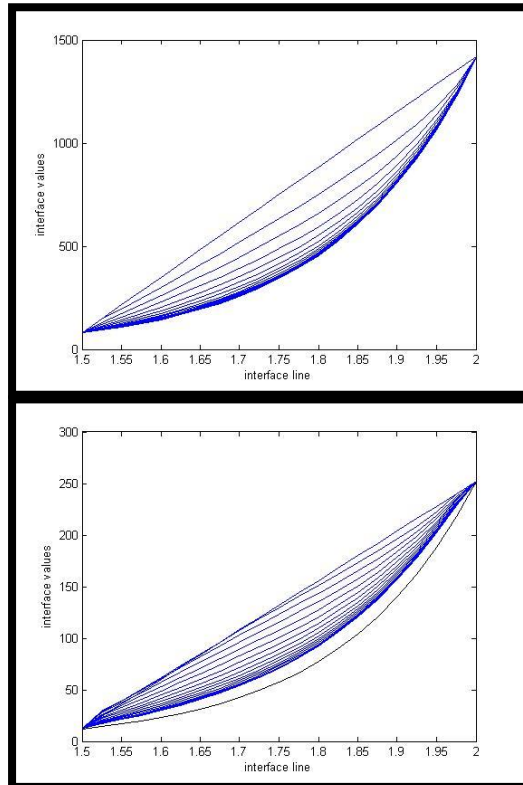


Εικόνα 15: Η ακριβής και οι υπολογισμένες λύσεις – ανομοιόμορφο πρόβλημα, περίπτωση Π2, διεπαφή 1 (πάνω) και διεπαφή 2 (κάτω).

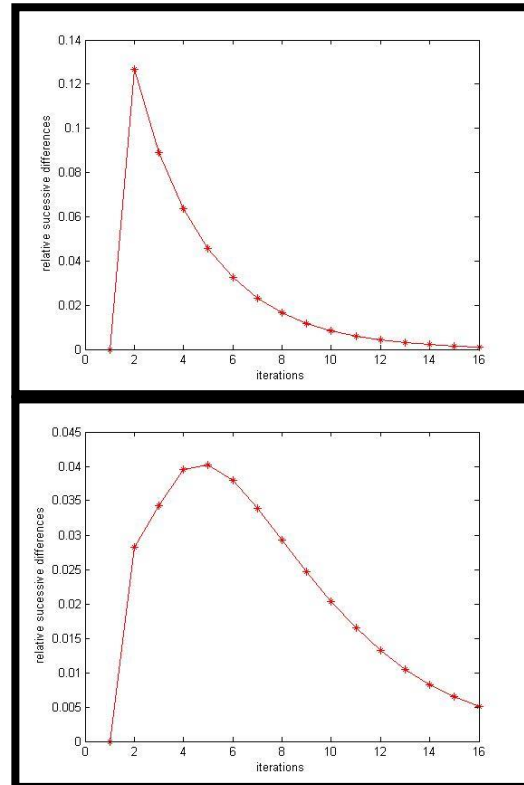


Εικόνα 16: Σχετικές διαφορές διαδοχικών επαναλήψεων - ανομοιόμορφο πρόβλημα, περίπτωση Π2, διεπαφή 1 (πάνω) και διεπαφή 2 (κάτω).

Ακολουθούν οι γραφικές παραστάσεις για την περίπτωση Π3. Αρχικά για το ομοιόμορφο πρόβλημα:

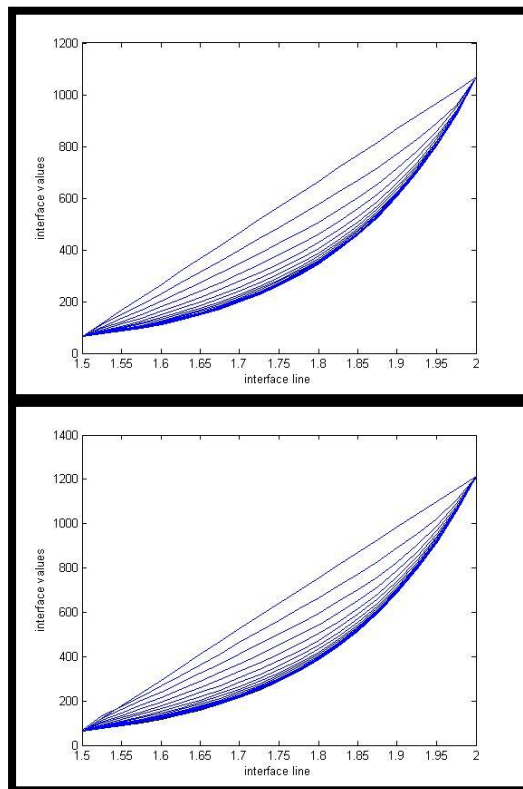


Εικόνα 17: Η ακριβής και οι υπολογισμένες λύσεις – ομοιόμορφο πρόβλημα, περίπτωση Π3, διεπαφή 1 (πάνω) και διεπαφή 2 (κάτω).

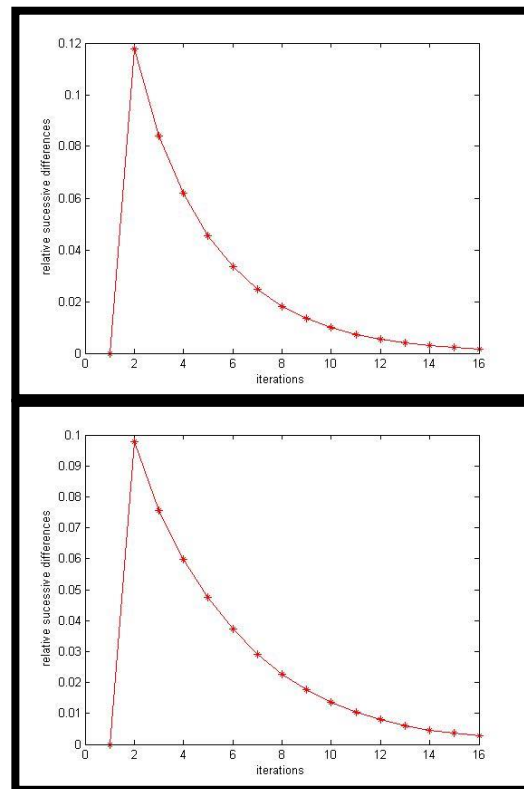


Εικόνα 18: Σχετικές διαφορές διαδοχικών επαναλήψεων - ομοιόμορφο πρόβλημα, περίπτωση Π3, διεπαφή 1 (πάνω) και διεπαφή 2 (κάτω).

Και στη συνέχεια για το ανομοιόμορφο πρόβλημα:



Εικόνα 19: Η ακριβής και οι υπολογισμένες λύσεις – ανομοιόμορφο πρόβλημα, περίπτωση Π3, διεπαφή 1 (πάνω) και διεπαφή 2 (κάτω).



Εικόνα 20: Σχετικές διαφορές διαδοχικών επαναλήψεων - ανομοιόμορφο πρόβλημα, περίπτωση Π3, διεπαφή 1 (πάνω) και διεπαφή 2 (κάτω).

Στον πίνακα 3 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 3: 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 Μηχανικών Η/Υ και Πληροφορικής, του Πανεπιστημίου Πατρών.

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.

7 Συμπεράσματα

Στην παρούσα διπλωματική παρουσιάζεται τόσο η σειριακή, όσο και η παράλληλη υλοποίηση της μεθόδου GEO στην πλατφόρμα FEniCS, ως ένα πρώτο βήμα στην προσπάθειά μας να δημιουργήσουμε ένα περιβάλλον επίλυσης προβλημάτων πολλαπλών πεδίων / πολλαπλών φυσικών. Η παράλληλη υλοποίησή μας αποδεικνύεται πολύ χρήσιμη, ειδικά για μεγάλα προβλήματα.

Συγκριτικά με ήδη υπάρχουσες υλοποιήσεις στην παρούσα διπλωματική και με την προτεινόμενη υλοποίηση προσπαθήσαμε να εκμεταλλευτούμε σε μεγαλύτερο βαθμό τον εγγενή παραλληλισμό της μεθοδολογίας χαλάρωσης στις διεπαφές, τον οποίο κάποιες από τις υπάρχουσες υλοποιήσεις τον εκμεταλλεύονται χρησιμοποιώντας το μοντέλο των πρακτόρων και κάποιες άλλες δεν τον εκμεταλλεύονται καθόλου. Επίσης, η προτεινόμενη υλοποίηση αποφεύγει την εξάρτηση από βιβλιοθήκες που περιέχουν επιλυτές, δίνοντας στο χρήστη τη δυνατότητα να επιλέξει όποιον επιλυτή επιθυμεί από όποια βιβλιοθήκη επιθυμεί ή να δημιουργήσει νέο επιλυτή κατάλληλο για το πρόβλημά του. Τέλος, η προτεινόμενη υλοποίηση θα είναι σύντομα ενταγμένη μέσα σε ένα πλήρες περιβάλλον επίλυσης προβλημάτων πολλαπλών πεδίων / πολλαπλών φυσικών, όπου ο χρήστης θα μπορεί να επιλέξει είτε τη μεθοδολογία χαλάρωσης στη διεπαφή, είτε κάποια άλλη μεθοδολογία της επιλογής του.

Στο άμεσο μέλλον σχεδιάζουμε επιπλέον πειραματισμό με πιο σύνθετες ΜΔΕ πολλαπλών πεδίων / πολλαπλών φυσικών, ακόμα λεπτότερα πλέγματα και διαφορετικούς επιλυτές σε κάθε υποπεδίο.

Σχεδιάζουμε, επίσης, την υλοποίηση επιπλέον μεθόδων χαλάρωσης στη διεπαφή, εκτός της GEO, ώστε ο χρήστης να μπορεί να χρησιμοποιεί τη μέθοδο της επιλογής του.

Ζητήματα επικοινωνίας σε διαφορετικά μηχανήματα μελετώνται, ενώ υψηλού και χαμηλού επιπέδου παραλληλισμός θα διερευνηθεί, επίσης.

Τα επόμενά μας βήματα περιλαμβάνουν, επίσης, τη δημιουργία μιας λειτουργικότητας στη FEniCS, όπου οι χρήστες θα μπορούν να:

1. ορίζουν σύνθετες ΜΔΕ πολλαπλών φυσικών,
2. επιλέγουν τους κατάλληλους επιλυτές ΜΔΕ για τα υποπεδία και μεθόδους χαλάρωσης στη διεπαφή για τις διεπαφές, και
3. οπτικοποιούν την υπολογισμένη λύση στο συνολικό πρόβλημα.

Βιβλιογραφία

- [1] B. Bialecki and G. Fairweather, Orthogonal spline collocation methods for partial differential equations, Numerical analysis 2000, Vol. VII, Partial differential equations J. Comput. Appl. Math., 128:1-2, pp. 55-82, 2001.
- [2] J. W. Daniel and B. K. Swartz, Extrapolated Collocation for Two-point Boundary-Value Problems using Cubic Splines, J. Inst. Maths Applies, 16, pp. 161-174, 1975.
- [3] C. C. Christara, Quadratic Spline Collocation Methods for Elliptic Partial Differential Equations, BIT, 34, pp. 33-61, 1994.
- [4] C. C. Christara and G. H. Liu, Quartic spline collocation for second-order boundary value problems, Proceedings of the 9th HERCMA Conference on Computer Mathematics and Applications, Athens University of Economics and Business, September 23-26, 2009, Athens, Greece, pp. 1-8.
- [5] E. N. Houstis, C. C. Christara and J. R. Rice, Quadratic-Spline Collocation Methods for Two-Point Boundary Value Problems, Internat. J. Numer. Methods Engrg., 26, pp. 935-952, 1988.
- [6] E. N. Houstis, E. A. Vavalis and J. R. Rice, Convergence of an $O(h^4)$ cubic spline collocation method for elliptic partial differential equations, SIAM J. Numer. Anal., 25:1, pp. 54-74, 1988.
- [7] M. Irodoutou-Ellina and E. N. Houstis, An $O(h^6)$ Quintic Spline Collocation Method for Fourth Order Two-Point Boundary Value Problems, BIT, 28, pp. 288-301, 1988.
- [8] Jingrui Zhang, Bi-quartic Spline Collocation Methods for Fourth-order Boundary Value Problems with an Application to the Biharmonic Dirichlet Problem, Ph.D. thesis, University of Toronto, Toronto, Ontario, Canada, 160 pages, 2008.
- [9] Y. Zhu, Optimal quartic spline collocation methods for fourth order two-point boundary value problems}, M.Sc. Thesis, University of Toronto, Toronto, Ontario, Canada, 73 pages, 2001.
- [10] Optimal Quadratic and Cubic Spline Collocation on Nonuniform Partitions, Computing, 76:3, pp. 227-257, 2006.
- [11] [ChNg06b] Adaptive Techniques for Spline Collocation, Computing, 76:3, pp. 259-277, 2006.
- [12] C. C. Christara, T. Chen and D. M. Dang, Quadratic spline collocation for one-dimensional parabolic partial differential equations, Numerical Algorithms, 43 pages, doi:10.1007/s11075-009-9317-9.
- [13] K.J. in't Hout and B.D. Welfert, Unconditional stability of second-order ADI schemes applied to multi-dimensional diffusion equations with mixed derivative terms, Appl. Numer. Math., 59, pp. 677-692, 2009.
- [14] A. Hadjidimos, T.S. Papatheodorou, Y.G. Saridakis, "Optimal Block Iterative Schemes for Certain Large, Sparse and Nonsymmetric Linear Systems", Linear Algebra & Applications **110**, pp 285-318, 1988.
- [15] M. Papadomanolaki and Y. G. Saridakis, "Hermite-Collocation for one dimensional tumor invasion model with heterogeneous diffusion", Proceedings 9th HERCMA Conference, Athens, Greece, 2009 και Hermis Journal, **11**, 2009

- [16] A.Hadjidimos, Y.G. Saridakis, “*Modified Successive Overrelaxation (MSOR) and Equivalent 2-step Iterative Methods for Collocation Matrices*”, J. of Computaitonal & Applied Math **42**, pp 375-393, 1992.
- [17] E. Mathioudakis, E.P. Papadopoulou and Y.G. Saridakis, “*Preconditioning for solving Hermite Collocation by the Bi-CGSTAB*”, Transactions on Mathematics, 5 (7), pp 811-816, 2006
- [18] E. Mathioudakis, E. P. Papadopoulou and Y.G. Saridakis, “*BiCGSTAB for Collocation systems on Distributed Memory Parallel Architectures*”, ENUMATH 2001, Ischia, Naples, 2001
- [19] E. Mathioudakis, E.P. Papadopoulou and Y.G. Saridakis, “*BiCGSTAB for Collocation systems on Shared Memory Parallel Architectures*”, International Conference on Numerical Algorithms, Marrakesh, 2001
- [20] E. Mathioudakis, E.P. Papadopoulou and Y.G. Saridakis, “*Iterative Solutions of Elliptic Collocation Systems on a Cognitive Parallel Computer*”, Computers & Mathematics with Applications, 48 (5-6), pp 951-970, 2004
- [21] E.N. Mathioudakis, E.P. Papadopoulou, “*MPI management of Hermite Collocation Computation on a Distributed-Shared Memory System*”, Transactions on Mathematics, 5(5), (520-525), 2006
- [22] E.N. Mathioudakis, E.P. Papadopoulou, “*Grid Computing for the Bi-CGSTAB applied to the solution of the Modified Helmholtz Equations*”, International J of Applied Mathematics and Computer Science, 4(3), (179-184), 2007
- [23] Rayleigh, L. 1899. On James Bernoullis theorem in probabilities. *Philos. Mag.* 47, 246–251.
- [24] Courant, R., Friedrichs, K., and Lewy, H. 1928. “U ber die partiellen differenzengleichungen der mathematischen physik. *Mathematische Annalen* 100, 32–74.
- [25] Metropolis, N. and Ulam, S. 1949. The Monte Carlo method. *Journal of the American Statistical Association* 44, 335–341.
- [26] Mascagni, M. and Hwang, C. 2003. _-Shell error analysis for “Walk On Spheres” algorithms. *Mathematics and Computers in Simulation* 63, 2, 93–104.
- [27] Lapeyre, B., Pardoux, E., and Sentis, R. 2003. *Introduction to Monte Carlo Methods for Transport and Diffusion Equations*. Oxford University Press. Translated by A. Craig and F. Craig.
- [28] Roman, L. and Sarkis, M. 2006. Stochastic Galerkin method for elliptic SPDEs: A white noise approach. *Discrete and Continuous Dynamical Systems - Series B* 6, 4, 941–955.
- [29] Alves, C. and Cruzeiro, A. 2008. Monte Carlo simulation of stochastic differential systems – a geometrical approach. *Stochastic Processes and their Applications* 118, 3, 346–367.
- [30] Babuska, I., Tempone, R., and Zouraris, G. E. 2005. Solving elliptic boundary value problems with uncertain coefficients by the finite element method: The stochastic formulation. *Computer Methods in Applied Mechanics and Engineering* 194, 12-16, 1251–1294.
- [31] Matthies, H. G. and Keese, A. 2005. Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations. *Computer Methods in Applied Mechanics and Engineering* 194, 12-16, 1295–1331.

- [32] Babuska, I., Nobile, F., and Tempone, R. 2007. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM Journal on Numerical Analysis* 45, 3, 1005–1034.
- [33] Hoshino, S. and Ichida, K. 1971. Solution of partial differential equations by a modified random walk. *Numerische Mathematik* 18, 1, 61–72.
- [34] Carlsson, J. 2000. A backward Monte Carlo method for solving parabolic partial differential equations. *arXiv:math/0010118v1*.
- [35] Griebel, M. and Schweitzer, M. A. 2001. A particle-partition of unity method for the solution of elliptic, parabolic, and hyperbolic PDEs. *SIAM Journal of Scientific Computing* 22, 3, 853–890.
- [36] Farnoosh, R. and Ebrahimi, M. 2007. Monte Carlo method via a numerical algorithm to solve a parabolic problem. *Applied Mathematics and Computation* 190, 2, 1593–1601.
- [37] Privault, N. 2008. Potential theory in classical probability. In *Quantum Potential Theory*, U. Franz and M. Schurmann, Eds. Lecture Notes in Mathematics, vol. 4310. Springer, 3–59.
- [38] A.S.Fokas, *A unified transform method for solving linear and certain nonlinear PDEs*, Proc. R. Soc. London A 53, (1997), 1411-1443.
- [39] A.S. Fokas, *A New Transform Method for Evolution PDEs*, IMA J. Appl. Math. 67 (2002), 559.
- [40] A.S. Fokas and B. Pelloni, *A Transform Method for Evolution PDEs on the Interval*, IMA J. Appl. Math. (2005), 1-24.
- [41] D. ben-Avraham and A.S. Fokas, *The Modified Helmholtz Equation in a Triangular Domain and an Application to Diffusion-limited Coalescence*, Phys. Rev. E 64 (2001), 016114-6.
- [42] Y.A. Antipov and A.S. Fokas, *A Transform Method for the Modified Helmholtz Equation in a Semi-Strip*, Math. Proc. Camb. Phil. Soc 138 (2005), 339-365.
- [43] G. Dassios and A.S. Fokas, *The Basic Elliptic Equations in an Equilateral Triangle*, Proc. R. Soc. Lond. A 461 (2005), 2721-2748.
- [44] D. Crowdy and A.S. Fokas, *Explicit Integral Solutions for the Plane Elastostatic Semi-Strip*, Proc. R. Soc. Lond. 460 (2004), 1285-1309.
- [45] A.S. Fokas, *Integrable Nonlinear Evolution Equations on the Half-Line*, Comm. Math. Phys. 230 (2002), 1-39.
- [46] A.S. Fokas, *Linearizable Initial Boundary Value Problems for the sine-Gordon Equation on the Half-Line*, Nonlinearity 17 (2004), 1521-1534.
- [47] S. Fulton, A.S. Fokas and C. Xenophontos, *An Analytical Method for Linear Elliptic PDEs and its Numerical Implementation*, J. Comp. Appl. Math. 167(2004), 465-483.
- [48] A. Sifalakis, A.S. Fokas, S. Fulton and Y.G.Saridakis, *The Generalized Dirichlet-Neumann Map for Linear Elliptic PDEs and its Numerical Implementation*, J. Comput. Appl. Math. 219 (2008), 9-34
- [49] S.A. Smitheman, E.A. Spence and A.S. Fokas, *A spectral collocation method for the Laplace and modified Helmholtz equations in a convex polygon*, IMA J. Num. Anal. (2010) 30 (4), 1184-1205
- [50] D. Funaro, A. Quarteroni, P. Zanolli, *An iterative procedure with interface relaxation for domain decomposition methods*, SIAM J. Numer. Anal. 25 (6) (1988) 1213–1236.

- [51] W. Heinrichs, Domain decomposition for fourth-order problems, *SIAM J. Numer. Anal.* 30 (2) (1993) 435–453.
- [52] C.H. Lai, On domain decomposition and shooting methods for two-point boundary value problem, in: D.E. Keyes and J. Xu (Eds.), *Seventh International Conference of Domain Decomposition Methods in Scientific and Engineering Computing*, Contemporary Mathematics, AMS, Vol. 180, 1994, pp. 257–264.
- [53] P. Le-Tallec, Y. De-Roecl, M. Vidrascu, Domain decomposition methods for large linearly elliptic 3-dimensional problems, *J. Comput. Appl. Math.* 34 (1) (1991) 93–117.
- [54] P.L. Lions, On the Schwarz alternating method III: A variant for nonoverlapping subdomains, in: R. Glowinski, G.H. Golub, G.A. Meurant and J. Periaux (Eds.), *Domain Decomposition Methods for Partial Differential Equations*, SIAM, Philadelphia, PA, 1990, pp. 202–223.
- [55] D. Yang, A parallel iterative nonoverlapping domain decomposition procedure for elliptic problems, *IMA J. Numer. Anal.* 16 (1996) 75–91.
- [56] 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.
- [57] T.F. Chan, D. Goovaerts, On the relationship between overlapping and nonoverlapping domain decomposition methods, *SIAM J. Matrix Anal. Appl.* 13 (1992) 663–670.
- [58] H.S. McFaddin, J.R. Rice, Collaborating PDE solvers, *Applied Numerical Mathematics* (1992) pp. 279–295.
- [59] M. Mu, Solving composite problems with interface relaxation, *SIAM Journal on Scientific Computing* 20 (1999) pp. 1394–1416.
- [60] M. Mu, J.R. Rice, Modeling with collaborating PDE solvers-theory and practice, *Computing Systems in Engineering* 6 (1995) pp. 87–95.
- [61] P. Tsompanopoulou, Collaborative PDEs solvers: Theory and practice, Ph.D. Thesis, Mathematics Department, University of Crete, Greece, 2000.
- [62] S. Markus, E. Houstis, A. Catlin, J. Rice, P. Tsompanopoulou, E. Vavalis, D. Gottfried, K. Su, G. Balakrishnan, An agent-based netcentric framework for multidisciplinary problem solving environments, *International Journal of Computational Engineering Science* 1 (2000) pp. 33–60.
- [63] J.R. Rice, An agent-based architecture for solving partial differential equations, *SIAM News* 31 (6) (1998).
- [64] Rice, J. R., Tsompanopoulou, P., Vavalis, E.: Interface relaxation methods for elliptic differential equations. *Applied Numerical Mathematics* 32 2, 219–245 (2000).
- [65] 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).
- [66] Drashansky T.: An Agent-Based Approach to Building Multidisciplinary Problem Solving Environments. PhD Thesis, Purdue University, Computer Science Department, (1996).
- [67] Rice, J.R., Tsompanopoulou, P., Vavalis, E.A.: *SciAgents Tool: User’s Guide*. Tech. Rpt. TR-98-043, Dept. Computer Sciences, Purdue Univ., (1998).

- [68] Bölöni, L., Marinescu, D.C., Rice, J.R., Tsompanopoulou, P., Vavalis, E.A.: Agent Based Scientific Simulation and Modelling. *Concurancy: Practice and Experience*, 12, 845–861 (2000).
- [69] Bölöni L, Marinescu DC. An object-oriented framework for building collaborative network agents. *Intelligent Systems and Interfaces*. Teodorescu NH, Mlynek D, Kandel A, Hoffmann H-J (eds.). Kluwer Publishing House, 2000; 31–65.
- [70] Palacz K, Marinescu DC. An agent-based workflow management system. *Proceedings AAAI Spring Symposium Workshop*. AAAI Press, 2000; 119–127.
- [71] Jun KK, Bölöni L, Yau DKY, Marinescu DC. Intelligent QoS support for an adaptive video service. *Proceedings International Resource Management Association*. IRMA, 2000; 1096–1099.
- [72] Jun KK, Bölöni L, Palacz K, Marinescu DC. Agent-based resource discovery. *Proceedings Heterogeneous Computing Workshop, HCW 2000*, vol. 1. IEEE Press, 2000; 43–52.
- [73] Tsompanopoulou P, Bölöni L, Marinescu DC, Rice JR. The design of software agents for a network of PDE solvers. *Proceedings Workshop on Agent Technologies for High Performance Computing*, at Agents 99, 1999; 57–68.
- [74] Finn T, Labrou Y, Mayfield J. KQML as an agent communication language. *Software Agents*. Bradshaw JM (ed.). MIT Press, 1997; 291–316.
- [75] Markus, S., Houstis, E., Catlin, A., Rice, J., Tsompanopoulou, P., Vavalis, E., Gottfried, D., Su K., Balakrishnan, G.: An Agent-Based Netcentric Framework for Multidisciplinary Problem Solving Environments. *International Journal of Computational Engineering Science*, 1, 33–60 (2000).
- [76] Houstis, E.N., Catlin, A.C., Tsompanopoulou, P., Gottfried, D., Balakrishnan, G., Su, K., Rice, J.R.: GASTURBNLAB: A Multidisciplinary Problem Solving Environment for Gas Turbine Engine Design on a Network of Non-Homogeneous Machines. *J. of Comp. and Applied Mathematics*, 149(1), 83–100 (2002).
- [77] The Grasshopper Agent Platform, IKV++ GmbH, Kurfurstendamm 173–174, D-10707 Berlin, Germany, <http://www.ikv.de>.
- [78] IRIS Explorer Toolkit, IRIS Explorer Center (North America), 1400 Opus Place, Suite 200, Downers Grove, IL 60515-5702, USA, <http://www.nag.com/IEC>.
- [79] Chalkias, C.: Implementation of a Distributed System for the Solution of MultiDomain / MultiPhysics Problems, Diploma Thesis, (2013), Dep. of Electrical and Computer Eng., Univ. of Thessaly. Online. Available: http://www.inf.uth.gr/wp-content/uploads/formidable/Chalkias_konstantinos.pdf
- [80] 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.
- [81] Tony F. Chan and Tarek P. Mathew. Domain decomposition algorithms. In *Acta Numerica 1994*, pages 61–43. Cambridge University Press, 1994.
- [82] T. F. Chan and D. C. Reasco. A survey of preconditioners for domain decomposition. Technical Report /DCS/RR-414, Yale University, 1985.
- [83] 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.

- [84] 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.
- [85] 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.
- [86] S. Kim. A parallelizable iterative procedure for the Helmholtz problem. *Appl. Numer. Math.*, 17:411-425, 1995.
- [87] 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.
- [88] D. Yang. A non-overlapping domain decomposition method for elliptic interface problems. Technical Report # 1472, University of Minnesota, 1997.
- [89] J. Xu and J. Zou. Non-overlapping domain decomposition methods. Technical report, Mathematics Department, Pennsylvania State University, University Park, PA, 1996.
- [90] D. Yang. A parallel domain decomposition algorithm for elliptic problems. *J. Comp. Math.*, 16:141-151, 1998.
- [91] 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.
- [92] J. Douglas Jr., P. J. Paes Leme, J. E. Roberts, and J. Wang. A parallel iterative procedure applicable to the approximate solution of the second order partial differential equations by mixed finite element methods. *Numer. Math.*, 65:95-108, 1993.
- [93] J. Douglas Jr. and C-S Huang. An accelerated domain decomposition procedure based on Robin transmission conditions. Technical report, Purdue University, Department of Mathematics, 1996.
- [94] Q. Deng. An analysis for a nonoverlapping domain decomposition iterative procedure. *SIAM J. Sci. Comput.*, 18:1517-1525, 1997.
- [95] H. T. M. van der Maarel and A. Platschorre. Optimization of flexible computing in domain decomposition for a system of PDEs. In *Proc. of the Ninth Int. Conf. on Domain Decomposition Methods*. 1998.
- [96] Rice, J.R., Tsompanopoulou, P. Vavalis, E.A.: Fine Tuning Interface Relaxation Methods for Elliptic Differential Equations. *Applied Numerical Mathematics*, 43(4), 459–481 (2002).
- [97] Tsompanopoulou, P., Vavalis, E.: An Experimental Study of Interface Relaxation Methods for Composite Elliptic Differential Equations. *Applied Mathematical Modelling*, 32 1620–1641 (2008).
- [98] Logg, A., Mardal, K. A., Wells, G. N. et al.: *Automated Solution of Differential Equations by the Finite Element Method*. Springer, (2012).
- [99] Rabbitmq, 2014. Online. Available: www.rabbitmq.com/documentation.html (accessed July 11, 2014).
- [100] Logg, A., Wells G. N.: DOLFIN: Automated Finite Element Computing. *ACM Transactions on Mathematical Software* 37, 2 (2010).
- [101] A. Korfiati, P. Tsompanopoulou and S. Likothanassis: Serial and Parallel Implementation of an Interface Relaxation Method. *NumAn2014 Conference*

Proceedings: Recent Approaches to Numerical Analysis: Theory, Methods and Applications, pages 167-173, (2014).