

Ετήσια Τεχνική Έκθεση

Έτος 2012



ΘΑΛΗΣ – Πολυτεχνείο Κρήτης

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

Δράση 2.1

Υβριδικές/Ασυνεχείς Μέθοδοι Collocation



Περιεχόμενα

1	Σκοπός	3
1.1	Παράλληλη επαναληπτική επίλυση των Collocation εξισώσεων σε γραφικά υποσυστήματα GPUs	3
2	Μεθοδολογία	4
2.1	Red Black Collocation γραμμικά συστήματα	4
2.1.1	Παράλληλος αλγόριθμος για τη μέθοδο Collocation	9
2.2	Η ασυνεχής Collocation σε 3 περιοχές	16
3	Αριθμητικά Αποτελέσματα Υλοποιήσεων	23
3.1	Αποτελέσματα υλοποίησης σε ένα GPU Shared-memory υπολογιστικό σύστημα	23
3.1.1	Συμπεράσματα υλοποίησης	25
3.2	Η ασυνεχής Collocation σε 3 περιοχές	25
4	Παραδοτέα	26
5	Συνεργασίες	26
6	Μελλοντικές Δράσεις	26

1 Σκοπός

Σκοπός της παρούσας δράσης είναι η ανάπτυξη και μελέτη ορθογώνιων ή spline collocation μεθόδων για την επίλυση μη-κλασικών ΜΔΕ πολλαπλών πεδίων (multiphysics, multidomain) και η αντιμετώπιση ασυνεχειών στους συντελεστές, αλλά και η κατανόηση της επίδρασης των ασυνεχειών αυτών στην συμπεριφορά της μεθόδου collocation όσον αφορά τον βαθμό σύγκλισης και την ευστάθεια της μεθόδου, καθώς και της κατάστασης των αντίστοιχων γραμμικών συστημάτων. Τα αριθμητικά σχήματα που θα αναπτυχθούν θα υλοποιηθούν σε σειριακά υπολογιστικά περιβάλλοντα αλλά και σύγχρονα παράλληλα για προβλήματα εφαρμογών (στα πλαίσια δράσεων 3 και 4). Την τρέχουσα περίοδο τα παραγόμενα κύρια ερευνητικά αποτελέσματα αφορούν την:

- Παράλληλη επαναληπτική επίλυση των εξισώσεων Hermite Collocation σε γραφικά υποσυστήματα υπολογισμών (GPUs),
- Εισαγωγή των ασυνεχών κυβικών Hermite πολυώνυμων και τη αντίστοιχη μέθοδο Collocation στις 1+1 διαστάσεις και 3 περιοχές ασυνέχειας,

και αναπτύσσονται στις παραγράφους που ακολουθούν.

1.1 Παράλληλη επαναληπτική επίλυση των Collocation εξισώσεων σε γραφικά υποσυστήματα GPUs

Αντικείμενο αυτής της ερευνητικής δραστηριότητας είναι η κατασκευή και υλοποίηση παράλληλων αλγορίθμων της αριθμητικής μεθόδου collocation, ώστε να είναι εφικτή η επίλυση multiphysics, multidomain ρεαλιστικών προβλημάτων σε σύγχρονες αρχιτεκτονικές υπολογισμών. Η συνολική ερευνητική διαδικασία περιγράφηκε σε εργασία, η οποία υποβλήθηκε για παρουσίαση στο διεθνές συνέδριο World Congress on Engineering 2013, το οποίο θα διεξαχθεί στο Imperial College του Λονδίνου τον Ιούλιο του 2013.

Ειδικότερα σε αυτή την ερευνητική δραστηριότητα μελετάται η υπολογιστικά απαιτητική αριθμητική επίλυση του μεγάλου, αραιού και μη-συμμετρικού συστήματος εξισώσεων που προκύπτει από τη διακριτοποίηση ελλειπτικών Προβλημάτων Συνοριακών Τιμών (ΠΣΤ) σύμφωνα με τη μέθοδο πεπερασμένων στοιχείων Collocation χρησιμοποιώντας ως βάση τα πολυώνυμα Hermite bicubic. Ο τύπος του γραμμικού συστήματος μας υποδεικνύει τη χρήση παράλληλων επαναληπτικών μεθόδων για την αποδοτική επίλυσή του. Έτσι εξετάζεται η εφαρμογή της μη στατικής επαναληπτικής μεθόδου Bi-Conjugate Gradient Stabilized (BiCGSTAB) με προρρύθμιση, σε πολυεπεξεργαστικές παράλληλες αρχιτεκτονικές κοινής μνήμης με την ταυτόχρονη χρήση γραφικών υποσυστημάτων (GPUs) για την επιπρόσθετη διεξαγωγή υπολογισμών. Για την αρτιότερη



υλοποίηση του όσο αφορά το υπολογιστικό κόστος των αριθμητικών διαδικασιών, κατασκευάζουμε τον Collocation πίνακα χρησιμοποιώντας μια διχρωματική αρίθμηση αγνώστων και εξισώσεων τύπου red-black, η οποία οδηγεί στην κατασκευή ενός καλά δομημένου επαναληπτικού παράλληλου αλγορίθμου. Η εφαρμογή της μεθόδου με την υλοποίηση του αλγορίθμου πραγματοποιήθηκε σε πολυεπεξεργαστικό μηχάνημα τύπου HP SL390 το οποίο διαθέτει γραφικά υποσυστήματα της κατηγορίας Tesla M2070. Οι μετρήσεις των χρόνων εκτέλεσης των υπολογισμών είναι ενδεικτικές της απόδοσης της παράλληλης αυτής υλοποίησης.

2 Μεθοδολογία

Η μέθοδος Πεπερασμένων Στοιχείων Collocation είναι μια αριθμητική διαδικασία διακριτοποίησης υψηλής τάξης ακρίβειας για εφαρμογές που προκύπτουν από ΠΣΤ σε πολλούς τομείς της μηχανικής και της επιστήμης γενικότερα (π.χ. [11]). Η μέθοδος προσεγγίζει τη λύση του προβλήματος αποφεύγοντας τη χρήση διαδικασιών αριθμητικής ολοκλήρωσης παρέχοντας άμεσα τις προσεγγιστικές τιμές της συνάρτησης καθώς και της πρώτης τάξης παραγώγων της σε όλους τους κόμβους του πλέγματος διακριτοποίησης. Έτσι το παραγόμενο γραμμικό σύστημα είναι αραιάς μορφής, χωρίς σημαντικές ιδιότητες και μεγάλου μεγέθους, υποδεικνύοντας τη χρήση επαναληπτικών επιλυτών [27, 28, 21] για την αποδοτική προσέγγιση της λύσης του. Σε ρεαλιστικές εφαρμογές, όπου είναι αναγκαία η χρήση πολύ μικρής διακριτοποίησης, η αποδοτική υλοποίηση της μεθόδου απαιτεί τη διεξαγωγή των υπολογισμών σε υψηλής απόδοσης αρχιτεκτονικές. Σε ένα τέτοιο επιστημονικό υπολογιστικό περιβάλλον μπορούν να συνδυαστούν πολυεπεξεργαστικά μηχανήματα με μονάδες γραφικών υποσυστημάτων (GPUs) για την επιτάχυνση της απόδοσης των αριθμητικών πράξεων. Η εκμετάλλευση των αυξημένων υπολογιστικών δυνατοτήτων των GPUs, απαιτεί την εφαρμογή ενός κατάλληλου, αποδοτικού παράλληλου αλγορίθμου για αρχιτεκτονικές κοινής μνήμης.

Επιστημονικά ζητήματα τέτοιας μορφής προσέλκυσαν το ενδιαφέρον ερευνητών στο παρελθόν και από τότε μέχρι σήμερα έχει πραγματοποιηθεί σημαντική πρόοδος στην παράλληλη επαναληπτική επίλυση με χρήση της μεθόδου πεπερασμένων στοιχείων Collocation (π.χ. [8], [5], [12, 13, 14, 16]).

2.1 Red Black Collocation γραμμικά συστήματα

Ας επιλέξουμε το ΠΣΤ Modified Helmholtz ως πρόβλημα μοντέλο

$$\begin{cases} \nabla^2 u(x, y) - \lambda u(x, y) = f(x, y) , & (x, y) \in \Omega \\ u(x, y) = g(x, y) , & (x, y) \in \partial\Omega \end{cases} \quad (1)$$



με $\lambda \geq 0$ στο ορθογώνιο χωρίο $\Omega \equiv (0, 1) \times (0, 1)$ ως ένα πρόβλημα δοκιμής. Θεωρούμε ομοιόμορφη διαμέριση των διαστημάτων $I^x = I^y = [0, 1]$ σε n_s υποδιαστήματα $I_m^x = I_m^y$, $m = 1, \dots, n_s$, τα οποία παράγουν ένα ομοιόμορφα καταναμημένο πλέγμα με βήμα διακριτοποίησης $h = \frac{1}{n_s}$ και συντεταγμένες κόμβων τα σημεία (x_i, y_j) , όπου $x_i = (i-1)h$ και $y_j = (j-1)h$, $i, j = 1, \dots, (n_s + 1)$. Η μέθοδος πεπερασμένων στοιχείων Hermite Collocation θα παράξει μια προσεγγιστική λύση $\tilde{u}(x, y)$ της μορφής

$$u(x, y) \sim \tilde{u}(x, y) = \sum_{i=1}^{\tilde{n}} \sum_{j=1}^{\tilde{n}} \alpha_{i,j} \phi_i(x) \phi_j(y), \quad (2)$$

όπου $\tilde{n} = 2(n_s + 1)$. Οι συναρτήσεις βάσης $\phi_i(x)$ και $\phi_j(y)$ είναι τα μονοδιάστατα, τμηματικά κυβικά πολυώνυμα Hermite [20]. Βασιζόμενος κάποιος στις ιδιότητες των συναρτήσεων βάσης Hermite, μπορεί εύκολα να επαληθεύσει ότι οι παρακάτω τέσσερις άγνωστοι

$$\begin{cases} a_{2i-1, 2j-1} &= \tilde{u}(x_i, y_j) \\ a_{2i-1, 2j} &= h \frac{\partial}{\partial y} \tilde{u}(x_i, y_j) \\ a_{2i, 2j-1} &= h \frac{\partial}{\partial x} \tilde{u}(x_i, y_j) \\ a_{2i, 2j} &= h^2 \frac{\partial^2}{\partial x \partial y} \tilde{u}(x_i, y_j) \end{cases} \quad (3)$$

αναφέρονται στο σημείο του πλέγματος (x_i, y_i) . Με την εισαγωγή των συνοριακών συνθηκών οι $8n_s + 4$ στο πλήθος άγνωστοι, οι οποίοι σχετίζονται με κόμβους του συνόρου $\partial\Omega$, μπορούν να προσδιοριστούν εκ των προτέρων. Για το λόγο αυτό, οι collocation εξισώσεις που χρειάζονται για τον προσδιορισμό των υπολοίπων $n = 4n_s^2$ αγνώστων κατασκευάζονται με την απαίτηση η προσέγγιση $\tilde{u}(x, y)$ να ικανοποιεί το ΠΣΤ σε n εσωτερικά collocation σημεία. Αυτά είναι τα 4 σημεία Gauss για καθένα από τα n_s^2 πεπερασμένα στοιχεία I_{ij} . Με δεδομένο ότι υπάρχει μια προς μια αντιστοιχία μεταξύ των σημείων αυτών και των εξισώσεων, η αρίθμηση των εξισώσεων θα προκύψει μόλις αριθμήσουμε αυτά τα collocation σημεία, ενώ η αρίθμηση των αγνώστων προκύπτει άμεσα μετά τον προσδιορισμό της αρίθμησης των αγνώστων για κάθε κόμβο. Η διαδικασία αυτή οδηγεί στη κατασκευή ενός γραμμικού συστήματος

$$Ax = b, \quad (4)$$

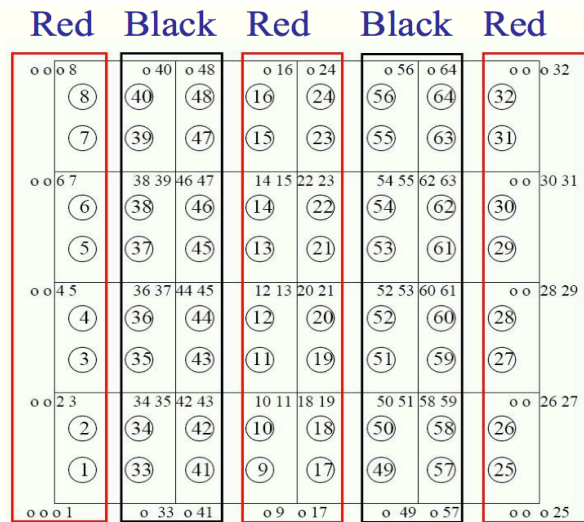
όπου A θα είναι ο $n \times n$ Collocation πίνακας συντελεστών των αγνώστων και

$$x = [x_1 \ x_2 \ \dots \ x_n]^T \equiv [\alpha_{1,1} \ \dots \ \alpha_{\tilde{n},\tilde{n}}]^T$$

θα είναι το διάνυσμα των αγνώστων διάστασης $n = 4n_s^2$.

Με στόχο την αύξηση των παράλληλων ιδιοτήτων του γραμμικού συστήματος, χρησιμοποιείται η διχρωματική αρίθμηση μορφής red-black (βλ. [12]) όπως





Σχήμα 1: Red-Black αρίθμηση των collocation αγνώστων και εξισώσεων για $n_s = 4$.

απεικονίζεται στο Σχ. 1 για μέγεθος διακριτοποίησης $n_s = 4$. Οι μικροί σε μέγεθος αριθμοί απεικονίζουν την αρίθμηση των αγνώστων σε κάθε κόμβο, ενώ οι αριθμοί εντός κύκλων την αρίθμηση των εξισώσεων ανά πεπερασμένο στοιχείο. Οι μικροί κύκλοι στο σύνορο υποδεικνύουν την ύπαρξη αγνώστων που υπολογίστηκαν με χρήση των συνοριακών συνθηκών στη θέση αυτή. Στο Σχ. 2 παρουσιάζεται σχηματικά η δομή του παραγόμενου Collocation πίνακα μετά την εφαρμογή κατάλληλου μετασχηματισμού ομοιότητας [12, 13]. Εύκολα καταλήγει κανείς στο συμπέρασμα ότι ο Collocation πίνακας θα έχει τη παρακάτω μορφή

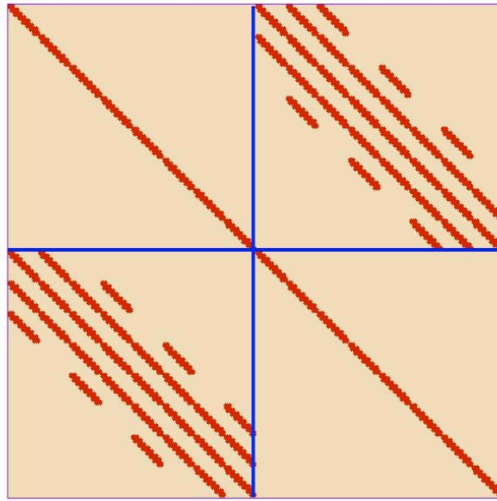
$$A = \begin{bmatrix} D_R & H_B \\ H_R & D_B \end{bmatrix}, \quad (5)$$

όπου D_R και D_B είναι αντιστρέψιμοι block διαγώνιοι πίνακες. Στην περίπτωση όπου $n_s = 2p$ αυτοί θα έχουν τη μορφή

$$D_R = \text{diag}[\underbrace{A_2 \ 2A_1 \ 2A_2 \ \cdots \ 2A_1 \ 2A_2 \ -A_2}_{2p\text{-blocks}}], \quad (6)$$

$$D_B = 2 \text{diag}[\underbrace{A_1 \ A_2 \ \cdots \ A_1 \ A_2}_{2p\text{-blocks}}] \quad (7)$$

$$H_R = \begin{bmatrix} R_1 & R_2 & & & & \\ R_3 & R_1 & R_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & R_3 & R_1 & R_2 \\ & & & & R_3 & \hat{R}_1 \end{bmatrix} \quad (8)$$



Σχήμα 2: Η δομή του Collocation πίνακα για $n_s = 4$.

$$H_B = \begin{bmatrix} B_1 & B_2 & & & & \\ B_3 & B_1 & B_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & & & \ddots & \\ & & & & & B_3 & B_1 & B_2 \\ & & & & & & B_3 & B_1 \end{bmatrix} \tag{9}$$

όπου

$$R_1 = \begin{bmatrix} A_4 & A_3 \\ -A_4 & A_3 \end{bmatrix}, \hat{R}_1 = \begin{bmatrix} A_4 & -A_4 \\ -A_4 & -A_4 \end{bmatrix},$$

$$R_2 = - \begin{bmatrix} A_4 & 0 \\ A_4 & 0 \end{bmatrix}, R_3 = \begin{bmatrix} 0 & A_3 \\ 0 & -A_3 \end{bmatrix},$$

και

$$B_1 = \begin{bmatrix} A_3 & -A_4 \\ A_3 & A_4 \end{bmatrix},$$

$$B_2 = \begin{bmatrix} 0 & 0 \\ A_3 & -A_4 \end{bmatrix}, B_3 = - \begin{bmatrix} A_3 & A_4 \\ 0 & 0 \end{bmatrix}.$$

Η block μορφή των παραπάνω πινάκων εμπλέκει τέσσερις πενταδιαγώνιους βασικούς πίνακες A_i για $i = 1, \dots, 4$, διάστασης $2n_s \times 2n_s$ [16].

Σε προηγούμενες εργασίες μας [13, 14, 15, 16] έχουμε παρουσιάσει την αποδοτική επίλυση του red-black collocation γραμμικού συστήματος, με χρήση της

επαναληπτικής μεθόδου SOR και επαναληπτικών μεθόδων υποχώρων Krylov, στηριζόμενοι στην ακόλουθη διάσπαση του πίνακα

$$A = D_A - L_A - U_A, \quad (10)$$

όπου

$$D_A = \begin{bmatrix} D_R & O \\ O & D_B \end{bmatrix}, \quad L_A = \begin{bmatrix} O & O \\ -H_R & O \end{bmatrix},$$

$$U_A = \begin{bmatrix} O & -H_B \\ O & O \end{bmatrix}, \quad (11)$$

και για τα υπόλοιπα μέρη του γραμμικού συστήματος υποθέσαμε την ανάλογη διαμέριση των διανυσμάτων x και b σε

$$x = \begin{bmatrix} x_R \\ x_B \end{bmatrix} \quad \text{και} \quad b = \begin{bmatrix} b_R \\ b_B \end{bmatrix}. \quad (12)$$

Σε αυτές τις εργασίες, όπως επίσης και στην [5], η επαναληπτική μέθοδος BiCGSTAB, προρρυθμισμένη είτε με το Symmetric Gauss-Seidel (SGS) είτε από το Gauss-Seidel (GS) επαναληπτικό σχήμα, επιτύγχανε ταχύτερα σύγκλιση σε σχέση με τη κλασική μέθοδο SOR ή από οποιαδήποτε άλλη τύπου Krylov επαναληπτική μέθοδο.

Με σκοπό τη μείωση του χρόνου επίλυσης, αλλά και την αύξηση του βαθμού παραλληλοποίησης της μεθόδου BiCGSTAB, θα εφαρμοστεί η παρακάτω προρύθμιση

$$M_1^{-1} A M_2^{-1} M_2 x = M_1^{-1} b, \quad (13)$$

όπου M_1 είναι ο επαναληπτικός πίνακας της μεθόδου Gauss Seidel, βασισμένος στη διάσπαση (10)

$$M_1 = D_A - L_A = D_A(I - D_A^{-1}L_A) \quad (14)$$

και

$$M_2 = I - D_A^{-1}U_A. \quad (15)$$

Οπότε το collocation γραμμικό σύστημα παίρνει τη παρακάτω μορφή

$$\begin{bmatrix} I & O \\ O & S \end{bmatrix} \begin{bmatrix} x_R + D_R^{-1}H_R x_B \\ x_B \end{bmatrix} = \begin{bmatrix} \hat{b}_R \\ \hat{b}_B \end{bmatrix}, \quad (16)$$

όπου

$$S = D_B - H_R D_R^{-1} H_B \quad (17)$$

είναι το συμπλήρωμα του Schur του collocation πίνακα όσο αφορά τον πίνακα D_B και

$$\hat{b}_R = D_R^{-1} b_R \quad \text{και} \quad \hat{b}_B = b_B - H_R \hat{b}_R. \quad (18)$$

Ο ακόλουθος αλγόριθμος περιγράφει αναλυτικά τις διαδοχικές φάσεις υπολογισμών για την επίλυση του γραμμικού συστήματος :

Αλγόριθμος για τις *collocation* εξισώσεις του συμπληρώματος *Schur*

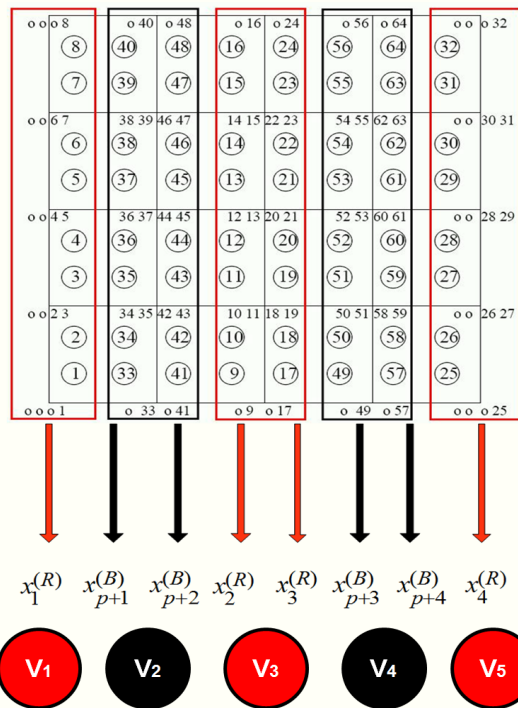
- S1: Επίλυση $D_R \hat{\mathbf{b}}_R = \mathbf{b}_R$
 S2: Υπολογισμός $\hat{\mathbf{b}}_B = \mathbf{b}_B - H_R \hat{\mathbf{b}}_R$
 S3: Επίλυση με BiCGSTAB $S \mathbf{x}_B = \hat{\mathbf{b}}_B$
 S4: Υπολογισμός $\hat{\mathbf{x}}_B = H_B \mathbf{x}_B$
 S5: Επίλυση $D_R \hat{\mathbf{x}}_R = \hat{\mathbf{x}}_B$
 S6: Υπολογισμός $\mathbf{x}_R = \hat{\mathbf{b}}_R - \hat{\mathbf{x}}_R$

Είναι προφανές ότι ο πολλαπλασιασμός πίνακα με διάνυσμα, με τους block διαγώνιους πίνακες D_R , D_B , H_B και H_R , είναι η διεργασία που κυριαρχεί στον αλγόριθμο, ακολουθούμενη από την άμεση επίλυση των γραμμικών συστημάτων που εμπλέκουν τους πίνακες D_R και D_B . Οι παραπάνω διεργασίες, στους *red* και *black* κύκλους υπολογισμών, απαιτούν τον σχεδιασμό αποδοτικών αλγορίθμων για υπολογιστικές αρχιτεκτονικές κοινής μνήμης.

2.1.1 Παράλληλος αλγόριθμος για τη μέθοδο *Collocation*

Η αρχιτεκτονική του διαθέσιμου παράλληλου μηχανήματος και ο αριθμός των επεξεργαστών του, είναι οι σημαντικότεροι παράγοντες που λαμβάνονται υπ' όψη στο σχεδιασμό και την οργάνωση των υπολογισμών ενός παράλληλου αλγορίθμου. Το υπολογιστικό σύστημα στο οποίο πραγματοποιήθηκαν οι δοκιμές, είναι κοινής μνήμης, αποτελούμενο από λίγους σε πλήθος αλλά ισχυρούς επεξεργαστικούς πυρήνες. Αντίθετα το γραφικό υποσύστημα GPU διαθέτει μερικές εκατοντάδες υπολογιστικών πυρήνων με δυνατότητα εκτέλεσης βασικών αριθμητικών πράξεων. Οι υπολογιστικοί αυτοί πυρήνες έχουν τη δική τους μνήμη κι έτσι θα πρέπει να ληφθεί υπ' όψη η συγκεκριμένη αρχιτεκτονική στο διαχωρισμό δεδομένων και υπολογισμών, ενώ ταυτόχρονα όλοι οι πυρήνες να είναι απασχολημένοι κατά τη διάρκεια των υπολογισμών. Αυτό το *no idle core* υπολογιστικό μοντέλο απαιτεί ισοκατανομή και ισορροπία μεταξύ φόρτου επικοινωνίας και υπολογισμών. Ένας αποδοτικός τρόπος εκτέλεσης αυτής της απαιτητικής διαδικασίας, δηλαδή της ανάθεσης των υπολογιστικών νημάτων (threads) σε κατάλληλους υπολογιστικούς πυρήνες, είναι να θεωρηθεί αρχικά μια εικονική αρχιτεκτονική με απεριόριστο αριθμό επεξεργαστών. Λαμβάνοντας υπ' όψη την αναγκαιότητα να έχουν τα threads ανεξάρτητα, μεταξύ τους, δεδομένα με κατά το δυνατόν ελαχιστοποίηση του κόστους επικοινωνίας με τη μνήμη, και ότι ο





Σχήμα 3: Αντιστοίχιση των collocation αγνώστων σε threads για $n_s = 4$.

αριθμός των υποδιαστημάτων $n_s = 2p$ της διακριτοποίησης και προς τις δύο κατευθύνσεις x και y είναι άρτιος, γίνεται αντιληπτό ότι η κατάλληλη κατανομή επιτυγχάνεται με τη αντιστοίχιση ενός thread για κάθε μια από τις $2p + 1$ κάθετες γραμμές του πλέγματος διακριτοποίησης. Αυτό παρουσιάζεται σχηματικά στο Σχ.3 για τη περίπτωση $n_s = 4$ πεπερασμένων στοιχείων σε κάθε κατεύθυνση.

Επιπλέον, τα περιττής αρίθμησης threads αντιστοιχούν σε red κάθετες γραμμές πλέγματος, ενώ τα άρτια threads αντιπροσωπεύουν υπολογισμούς των black γραμμών πλέγματος. Εξαιτίας της $2p$ -block διαμέρισης όλων των διανυσμάτων στους υπολογισμούς, είναι ξεκάθαρο ότι σε καθένα από τα περιττα threads V_{2i-1} , $i = 1, \dots, p + 1$, έχει ανατεθεί ο υπολογισμός των διανυσμάτων t_{2i-2} και t_{2i-1} της λύσης, ενώ κάθε άρτιο thread V_{2i} , $i = 1, \dots, p$, θα πρέπει να υπολογίσει τα διανύσματα $t_{2p+2i-1}$ και t_{2p+2i} της λύσης. Η ιδιαιτερότητα που παρουσιάζεται στα thread V_1 και V_{2p+1} οφείλεται στις συνοριακές συνθήκες.

Στον παράλληλο αλγόριθμο, όλες οι βασικές πράξεις διανυσμάτων γραμμικής άλγεβρας όπως εσωτερικά γινόμενα, πρόσθεση διανυσμάτων και πολλαπλασιασμοί τους με βαθμωτά μεγέθη, μπορούν να εκτελεστούν παράλληλα, βάσει της παραπάνω $2p$ -block διαμέρισης όλων των διανυσμάτων. Ο λόγος που μπορούν να εκτελεστούν παράλληλα οι παραπάνω διαδικασίες είναι η ανεξαρτησία μεταξύ

των δεδομένων που έχει το κάθε thread.

Σύμφωνα με τον παραπάνω τρόπο αντιστοίχισης δεδομένων σε threads, οι παράλληλες διαδικασίες είναι ανεξάρτητες ως προς τα δεδομένα για όλους τους πολλαπλασιασμούς πίνακα με διάνυσμα και για τις πράξεις άμεσης επίλυσης κατά τους red και black κύκλους υπολογισμών. Οι παρακάτω παράλληλοι αλγόριθμοι ενσωματώνουν όλες τις παραπάνω ιδιότητες για τον υπολογισμό ενός τυχαίου διανύσματος t διάστασης $4n_s^2$.

<p><u>Red Computation Cycle</u></p> <pre>C\$ OMP PARALLEL DO DEFAULT(SHARED) do i = 0 to p V_{2i+1} computes t_{2i}, t_{2i+1} enddo C\$ OMP END PARALLEL DO</pre>
<p><u>Black Computation Cycle</u></p> <pre>C\$ OMP PARALLEL DO DEFAULT(SHARED) do i = 1 to p V_{2i} computes $t_{2p+2i}, t_{2p+2i-1}$ enddo C\$ OMP END PARALLEL DO</pre>

Η υλοποίηση των παραπάνω αλγορίθμων με CPU threads μπορεί να πραγματοποιηθεί με χρήση αποδοτικών διαδικασιών από υπάρχουσες αριθμητικές βιβλιοθήκες [10]. Για παράδειγμα, στην περίπτωση της εμπρός και πίσω αντικατάστασης κατά τη διάρκεια της block άμεσης επίλυσης του γραμμικού συστήματος με τους πίνακες D_R και D_B , μπορεί να επιλεγεί η κατάλληλη διαδικασία από την βιβλιοθήκη Lapack, ενώ για τον πολλαπλασιασμό πίνακα με διάνυσμα, που εμπλέκει τους πίνακες H_B και H_R , μπορεί να χρησιμοποιηθεί η κατάλληλη διαδικασία από τη βιβλιοθήκη BLAS. Ωστόσο, για την υλοποίηση με GPU threads κρίνεται αναγκαία η σχεδίαση κατάλληλων αλγορίθμων, έτσι ώστε να είναι εφικτή η εκμετάλλευση της ιδιαίτερης δομής των πινάκων που εμπλέκονται στους υπολογισμούς.

Η σειριακότητα των υπολογισμών κατά τη διάρκεια της εμπρός και πίσω αντικατάστασης στη φάση επίλυσης του γραμμικού συστήματος, σε συνδυασμό με τους περιορισμούς που υπάρχουν λόγω μεγέθους μνήμης των GPUs, άμεσα μας υποδεικνύει ότι οι επιλύσεις κατά τους red και black κύκλους υπολογισμών, με τους πίνακες συντελεστών D_R και D_B αντίστοιχα, πρέπει να εκτελεστούν από CPU threads.

Από την άλλη πλευρά η ανεξαρτησία των δεδομένων που εμπλέκονται στις βασικές πράξεις μεταξύ πινάκων και διανυσμάτων, σε συνδυασμό με το γεγονός ότι οι εκατοντάδες υπολογιστικοί πυρήνες των GPUs είναι οργανωμένοι σε

υπολογιστικές ομάδες και, μέσω της εκτέλεσης εφαρμογών κατά SIMD, μπορούν εκτελέσουν ταυτόχρονα βασικές αριθμητικές πράξεις ταχύτερα από ότι οι πυρήνες του κεντρικού επεξεργαστή του μηχανήματος. Κατά συνέπεια, κρίνεται κατάλληλη η εκτέλεση των διαδικασιών που εμπλέκουν πολλαπλασιασμό πίνακα-διανύσματος κατά τη διάρκεια των red και black κύκλων, με τους block πίνακες H_R και H_B , στη GPU. Περαιτέρω βελτίωση της απόδοσης μπορεί να επιτευχθεί με την εκμετάλλευση της block δομής των H_R και H_B , από τις σχέσεις (8) και (9) αντίστοιχα, επειδή εμπλέκονται αποκλειστικά δύο πενταδιαγώνιοι πίνακες A_3 και A_4 τάξης $2n_s$.

Ο παρακάτω αλγόριθμος περιγράφει λεπτομερώς ένα black κύκλο υπολογισμών του $t = H_B z$ πολλαπλασιασμού πίνακα-διανύσματος στη GPU.

```

Black GPU Cycle
!$ACC DATA COPYIN(z) CREATE(temp) COPYOUT(t)
!$ACC KERNELS
!$ACC LOOP INDEPENDENT
  do k = 1 to n_s - 3 with step 2
    k_1 = (k - 1)2n_s , k_2 = k2n_s
    k_3 = (k + 1)2n_s , k_4 = (k + 2)2n_s
!$ACC LOOP INDEPENDENT
    do i = 1 to 2n_s
      t(k_2 + i) = A_3(3, i)z(k_1 + i) + A_4(3, i)z(k_2 + i)
    enddo
!$ACC LOOP INDEPENDENT
    do i = 2 to 2n_s
      t(k_2 + i - 1) = t(k_2 + i - 1) +
        A_3(2, i)z(k_1 + i - 1) + A_4(2, i)z(k_2 + i)
    enddo
!$ACC LOOP INDEPENDENT
    do i = 1 to 2n_s - 1
      t(k_2 + i + 1) = t(k_2 + i + 1) +
        A_3(4, i)z(k_1 + i) + A_4(4, i)z(k_2 + i)
    enddo
!$ACC LOOP INDEPENDENT
    do i = 1 to 2n_s - 2
      t(k_2 + i) = t(k_2 + i) +
        A_3(1, i + 2)z(k_1 + i + 2) +
        A_4(1, i + 2)z(k_2 + i + 2)
    enddo
!$ACC LOOP INDEPENDENT
    do i = 1 to 2n_s - 2
      t(k_2 + i + 2) = t(k_2 + i + 2) +
        A_3(5, i)z(k_1 + i) + A_4(5, i)z(k_2 + i)
    enddo
  enddo

```



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

```

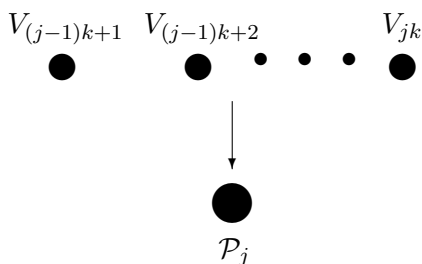
!$ACC LOOP INDEPENDENT
  do   i = 1 to 2ns
     $t(k_3 + i) = A_3(3, i)z(k_3 + i) - A_4(3, i)z(k_4 + i)$ 
  enddo
!$ACC LOOP INDEPENDENT
  do   i = 2 to 2ns
     $t(k_3 + i - 1) = t(k_3 + i - 1) +$ 
       $A_3(2, i)z(k_3 + i - 1) - A_4(2, i)z(k_4 + i)$ 
  enddo
!$ACC LOOP INDEPENDENT
  do   i = 1 to 2ns - 1
     $t(k_3 + i + 1) = t(k_3 + i + 1) +$ 
       $A_3(4, i)z(k_3 + i) - A_4(4, i)z(k_4 + i)$ 
  enddo
!$ACC LOOP INDEPENDENT
  do   i = 1 to 2ns - 2
     $t(k_3 + i) = t(k_3 + i) +$ 
       $A_3(1, i + 2)z(k_3 + i + 2) -$ 
       $A_4(1, i + 2)z(k_4 + i + 2)$ 
  enddo
!$ACC LOOP INDEPENDENT
  do   i = 1 to 2ns - 2
     $t(k_3 + i + 2) = t(k_3 + i + 2) +$ 
       $A_3(5, i)z(k_3 + i) - A_4(5, i)z(k_4 + i)$ 
  enddo
!$ACC LOOP INDEPENDENT
  do   i = 1 to 2ns
     $temp(i) = t(k_3 + i)$ 
  enddo
!$ACC LOOP INDEPENDENT
  do   i = 1 to 2ns
     $t(k_3 + i) = t(k_3 + i) - t(k_2 + i)$ 
  enddo
!$ACC LOOP INDEPENDENT
  do   i = 1 to 2ns
     $t(k_2 + i) = t(k_2 + i) + temp(i)$ 
  enddo
!$ACC END KERNELS
!$ACC END DATA

```

Σημειώνεται ότι το τμήμα του αλγορίθμου που αφορά τη κατασκευή του πρώτου και του τελευταίου block γραμμών του πίνακα H_B δεν συμπεριλαμβάνεται, αλλά έχει παρόμοια μορφή. Η πραγματοποίηση των υπολογισμών του αντίστοιχου red κύκλου για GPU μπορεί να περιγραφεί με παρόμοιο αλγόριθμο.

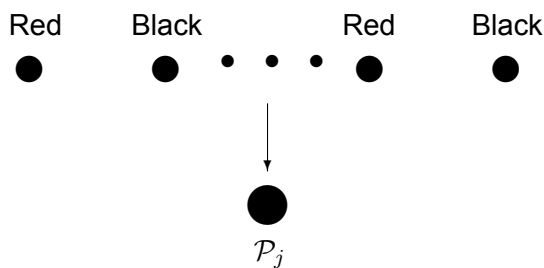
Απεικόνιση υπολογισμών σε αρχιτεκτονικές σταθερής διάστασης

Για την εκτέλεση του αλγορίθμου σε ένα παράλληλο υπολογιστικό σύστημα, αποτελούμενου από P πυρήνες, ομάδες των threads πρέπει να αντιστοιχούν σε πυρήνες της GPU και σε αυτούς του κεντρικού επεξεργαστή. Αυτή η ενότητα περιγράφει τη διαδικασία αντιστοίχισης για τη περίπτωση όπου $n_s = kP$, καθώς για τις υπόλοιπες περιπτώσεις υπάρχει όμοια αντιμετώπιση. Στη συγκεκριμένη περίπτωση, ωστόσο, το υπολογιστικό κόστος είναι το ίδιο για όλους τους πυρήνες υπολογισμών. Σύμφωνα με τον τρόπο αντιστοίχισης που ακολουθήθηκε ως τώρα, κάθε k διαδοχικά threads αντιστοιχίζονται σε κάθε έναν από τους \mathcal{P}_j , ($j = 1, \dots, P$) πυρήνες, με βάση την αρχιτεκτονική του συστήματος.



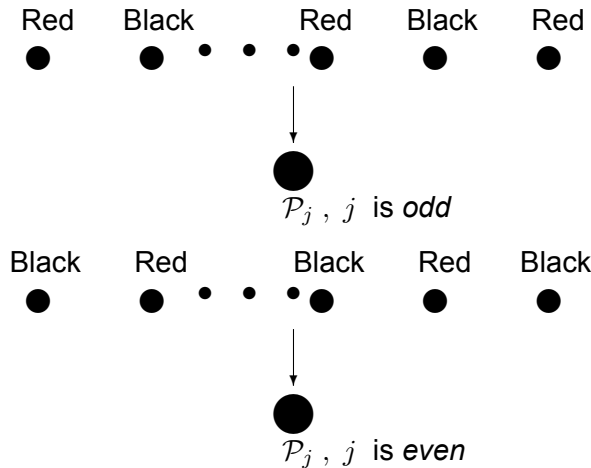
Οπότε γίνεται σαφές, ακολουθώντας την παραπάνω διαδικασία, ότι σε κάθε \mathcal{P}_j πυρήνα μπορούμε να αντιστοιχίσουμε k threads $V_{(j-1)k+1}, \dots, V_{jk}$. Παρατηρούμε ότι:

- Όταν το k είναι *άρτιος* οι δείκτες $(j - 1)k + 1$ και jk που ικανοποιούν τη συνθήκη $(j - 1)k + 1$ είναι *περιττοί*, ενώ οι jk είναι *άρτιοι*. Οπότε τα υπολογιστικά threads $V_{(j-1)k+1}$ και V_{jk} είναι αντίστοιχα *red (περιττά)* και *black (άρτια)* threads κι έτσι σχηματικά θα ισχύει ότι:



Οπότε, στους πυρήνες \mathcal{P}_j αντιστοιχίζονται τα k *red* διανύσματα t_l , $l = (j - 1)k, \dots, jk - 1$ και τα k *black* διανύσματα t_{2p+l} , $l = (j - 1)k + 1, \dots, jk$.

- Όταν ο δείκτης k είναι *περιττός* οι δείκτες $(j - 1)k + 1$ και jk που ικανοποιούν τη συνθήκη $(j - 1)k + 1$ και jk είναι *περιττοί* όταν j είναι *περιττός*, ενώ οι $(j - 1)k + 1$ και jk είναι *άρτιοι* όταν το j είναι *άρτιος*. Έτσι τα threads $V_{(j-1)k+1}$ και V_{jk} είναι μαζί *red (περιττά)* όταν το j είναι *περιττός* ενώ και τα δυο είναι *black (άρτια)* όταν το j είναι *άρτιος* κι έτσι θα έχουμε το σχήμα:



Οπότε, όταν το j είναι *περιττός*, στους πυρήνες \mathcal{P}_j αντιστοιχίζονται τα $k+1$ *red* διανύσματα t_l , $l = (j-1)k, \dots, jk$ και τα $k-1$ *black* διανύσματα t_{2p+l} , $l = (j-1)k+1, \dots, jk-1$, ενώ, όταν το j είναι *άρτιος*, στους πυρήνες \mathcal{P}_j αντιστοιχίζονται τα $k-1$ *red* διανύσματα t_l , $l = (j-1)k+1, \dots, jk-1$ και τα $k-1$ *black* διανύσματα t_{2p+l} , $l = (j-1)k, \dots, jk$.

Όλα τα παραπάνω μπορούν να περιγραφούν με τον παρακάτω παράλληλο αλγόριθμο για τον υπολογισμό κάθε επαναληπτικού βήματος του Schur complement:

Παράλληλος αλγόριθμος για Schur comp. collocation εξισώσεις

- S1: Solve in parallel on host $D_R \hat{\mathbf{b}}_R = \mathbf{b}_R$
- S2: Send matrices A_3 and A_4 to GPU
- S3: Evaluate in parallel on GPU $\hat{\mathbf{b}}_B = \mathbf{b}_B - H_R \hat{\mathbf{b}}_R$
- S4: Solve in parallel with BiCGSTAB $S \mathbf{x}_B = \hat{\mathbf{b}}_B$
- S5: Evaluate in parallel on GPU $\hat{\mathbf{x}}_B = H_B \mathbf{x}_B$
- S6: Solve in parallel on host $D_R \hat{\mathbf{x}}_R = \hat{\mathbf{x}}_B$
- S7: Evaluate in parallel on host $\mathbf{x}_R = \hat{\mathbf{b}}_R - \hat{\mathbf{x}}_R$

Οι υπολογισμοί του βήματος S4 της μεθόδου BiCGSTAB πραγματοποιούνται από τους επεξεργαστές του υπολογιστικού συστήματος εκτός από τους υπολογισμούς των δυο πολλαπλασιασμών πινάκων-διανυσμάτων σε κάθε επαναληπτικό βήμα όπου εμπλέκεται ο Schur complement πίνακας S . Πιο αναλυτικά, οι υπολογισμοί του βήματος S4, που εμπλέκουν τους πίνακες H_R και H_B ως πολλαπλασιαστές, θα διεξάγονται στη GPU σύμφωνα με τον ακόλουθο αλγόριθμο:

Υπολογισμός του $t = Sp$

S1: Send p from host to GPU

S2: Evaluate in parallel on GPU $t = H_B p$

S3: Send t from GPU to host

S4: Solve in parallel on host $D_R s = t$

S5: Send s from host to GPU

S6: Evaluate in parallel on GPU $q = H_R s$

S7: Send q from GPU to host

S8: Evaluate in parallel on host $t = D_B p - q$

Το κόστος επικοινωνίας για τη μεταφορά δεδομένων μεταξύ της κύριας μνήμης του υπολογιστικού συστήματος και του γραφικού υποσυστήματος είναι αυτό της μεταφοράς δυο διανυσμάτων μεγέθους $2n_s^2$ ανά κατεύθυνση. Δηλαδή, το κόστος επικοινωνίας κάθε επαναληπτικού βήματος της BiCGSTAB είναι το κόστος μεταφοράς οκτώ διανυσμάτων μεγέθους $2n_s^2$ προς κάθε κατεύθυνση, αφού οι πίνακες A_3 και A_4 θα μεταφερθούν και θα αποθηκευτούν στη μνήμη της GPU κατά την έναρξη της διαδικασίας επίλυσης μία και μόνο φορά.

2.2 Η ασυνεχής Collocation σε 3 περιοχές

Ορμώμενοι απο το μοντέλο διάχυσης καρικινικών όγκων στον εγκέφαλο (βλ. Τεχνική Έκθεση Δράση 4.2 Έτος 2012 και σχετικές αναφορές), αναπτύξαμε [18]-[19] την ασυνεχή μέθοδο Hermite Collocation (dDHC) για το αδιάστατο μοντέλο της μορφής:

$$\begin{cases} c_t = (Dc_x)_x + c, & x \in [a, b], t \geq 0 \\ c_x(a, t) = 0 \text{ και } c_x(b, t) = 0 \\ c(x, 0) = f(x) \end{cases} \quad (19)$$

ή, ισοδύναμα, αντικαθιστώντας $c(x, t) = e^t u(x, t)$, της μορφής:

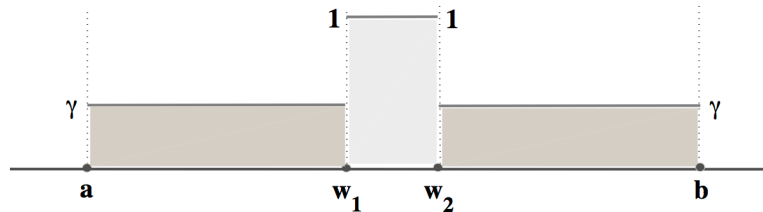
$$\begin{cases} u_t = (Du_x)_x, & x \in [a, b], t \geq 0 \\ u_x(a, t) = 0 \text{ και } u_x(b, t) = 0 \\ u(x, 0) = f(x) \end{cases} \quad (20)$$

Χαρακτηριστική ιδιότητα στην εξίσωση του ανωτέρου μοντέλου είναι ο ασυνεχής



συντελεστής διάχυσης $D = D(x)$ της μορφής:

$$D(x) = \begin{cases} \gamma & , a \leq x < w_1 \\ 1 & , w_1 \leq x < w_2 \\ \gamma & , w_2 \leq x \leq b \end{cases} \quad (21)$$



με $\gamma < 1$ να είναι η αδιάστατη σταθερά διάχυσης. Επίσης, προς χάριν γενικότητας, θεωρούμε ότι η συνάρτηση $f(x)$, που χαρακτηρίζει την αρχική συνθήκη, είναι μία τυχαία συνεχής συνάρτηση.

Η παραβολική φύση της διαφορικής συνεπάγεται την συνέχεια της u , καθώς και των u_t , (Du_x) . Και, καθώς ο συντελεστής D είναι ασυνεχής, η u_x πρέπει να είναι επίσης κατάλληλα ασυνεχής, ώστε να προκύπτει συνέχεια της (Du_x) . Επομένως, μπορούμε να θεωρήσουμε ότι

$$[u] := u^+ - u^- = 0 \quad , \quad \text{για } x = w_k \quad , \quad k = 1, 2 \quad , \quad (22)$$

$$u^+ := \lim_{x \rightarrow w_k^+} u(x) \quad \text{και} \quad u^- := \lim_{x \rightarrow w_k^-} u(x) \quad ,$$

$$[Du_x] := D^+ u_x^+ - D^- u_x^- = 0 \quad , \quad \text{για } x = w_k \quad , \quad k = 1, 2 \quad . \quad (23)$$

Οπότε χρησιμοποιώντας τους παραπάνω περιορισμούς συνέχειας στα σημεία διασύνδεσης το μοντέλο καταλήγει στην παρακάτω μορφή :

$$\begin{cases} u_t = Du_{xx} \quad , \quad x \in \mathcal{R}_\ell \quad , \quad \ell = 1, 2, 3 \quad , \quad t \geq 0 \\ u_x(a, t) = 0 \quad \text{και} \quad u_x(b, t) = 0 \\ [u] = 0 \quad \text{και} \quad [Du_x] = 0 \quad \text{για } x = w_k \quad , \quad k = 1, 2 \\ u(x, 0) = f(x) \end{cases} \quad (24)$$

με \mathcal{R}_ℓ να δηλώνει την κάθε περιοχή

$$\mathcal{R}_1 := [a, w_1) \quad , \quad \mathcal{R}_2 := (w_1, w_2) \quad , \quad \mathcal{R}_3 := (w_2, b] \quad . \quad (25)$$

Μεταξύ άλλων προκαταρτικών μεθόδων που αναπτύξαμε (βλ. [19] και τις σχετικές αναφορές) για να αντιμετωπίσουμε τις ασυνεχείς στα σημεία διεπαφής, εισαγάγαμε και την νέα ασυνεχή μέθοδο Hermite Collocation (derivative Discontinuous Hermite Collocation - dDHC). Για μία σύντομη περιγραφή αλλά και για να αναπτύξουμε τον απαιτούμενο φορμαλισμό, ας θεωρήσουμε κατ' αρχήν για κάθε περιοχή \mathcal{R}_ℓ ένα ομοιόμορφο πλέγμα $t \times \mathcal{R}_\ell(\epsilon)$ που οι πλευρές είναι παράλληλες στους άξονες x και t και το βήμα διακριτοποίησης είναι h_ℓ και τ στην x και t κατεύθυνση αντίστοιχα. Συγκεκριμένα

$$\begin{cases} h_1 := (w_1 - a)/N_1, \\ h_2 := (w_2 - w_1)/N_2, \\ h_3 := (b - w_2)/N_3, \end{cases} \quad (26)$$

όπου N_ℓ δηλώνει τον αριθμό των υποδιαστημάτων της κάθε περιοχής $\mathcal{R}_\ell(\epsilon)$ αντίστοιχα, καθώς και οι συντεταγμένες των κόμβων (x_{ℓ, m_ℓ}, t_n) δίνονται από τις σχέσεις :

$$\begin{cases} x_{1, m_1} := a + m_1 h_1, & m_1 = 0, \dots, N_1 \\ x_{2, m_2} := w_1 + m_2 h_2, & m_2 = 0, \dots, N_2 \\ x_{3, m_3} := w_2 + m_3 h_3, & m_3 = 0, \dots, N_3 \end{cases} \quad (27)$$

και

$$t_n = n\tau, \quad n = 0, 1, \dots \quad (28)$$

Για ομαλές λύσεις u της διαφορικής εξίσωσης (24) σε κάθε υποδιάστημα $\mathcal{R}_\ell(\epsilon)$, $\ell = 1, 2, 3$ η προσεγγιστική λύση $U_\ell(x, t)$ χρησιμοποιώντας πεπερασμένα στοιχεία με συναρτήσεις βάσης τα πολυώνυμα Hermite θα είναι της μορφής:

$$U_\ell(x, t) = \sum_{j_\ell=1}^{N_\ell+1} [\alpha_{\ell, 2j_\ell-1}(t)\phi_{\ell, 2j_\ell-1}(x) + \alpha_{\ell, 2j_\ell}(t)\phi_{\ell, 2j_\ell}(x)] \quad (29)$$

όπου $\phi_{\ell, 2j_\ell-1}(x)$ και $\phi_{\ell, 2j_\ell}(x)$ είναι τα Hermite κυβικά πολυώνυμα.

Σε κάθε κόμβο $x_{\ell, j_\ell-1}$ ορίζονται δύο κυβικά πολυώνυμα Hermite :

$$\phi_{\ell, 2j_\ell-1}(x) = \begin{cases} \phi\left(\frac{x_{\ell, j_\ell-1}-x}{h_\ell}\right), & x \in I_{\ell, j_\ell-1} \\ \phi\left(\frac{x-x_{\ell, j_\ell-1}}{h_\ell}\right), & x \in I_{\ell, j_\ell} \\ 0, & \text{διαφορετικά} \end{cases} \quad (30)$$

και

$$\phi_{\ell,2j_\ell}(x) = \begin{cases} -h_\ell \psi\left(\frac{x_{\ell,j_\ell-1}-x}{h_\ell}\right) & , x \in I_{\ell,j_\ell-1} \\ h_\ell \psi\left(\frac{x-x_{\ell,j_\ell-1}}{h_\ell}\right) & , x \in I_{\ell,j_\ell} \\ 0 & , \text{διαφορετικά} \end{cases} \quad (31)$$

με $I_{\ell,j_\ell} := [x_{\ell,j_\ell-1}, x_{\ell,j_\ell}]$, $j_\ell = 1, \dots, N_\ell$. Οι συναρτήσεις $\phi(s)$ και $\psi(s)$ προέρχονται συμβολίζουν τα Hermite κυβικά πολυώνυμα στο $[0, 1]$ δηλαδή:

$$\phi(s) = (1-s)^2(1+2s) \quad , \quad \psi(s) = s(1-s)^2 \quad , \quad (32)$$

με $s \in [0, 1]$.

Για την λύση u του μαθηματικού προβλήματος (24), αφού υπάρχουν ασυνέχειες στα σημεία διασύνδεσης $w_1 \equiv x_{N_1+1}$ και $w_2 \equiv x_{N_1+2+1}$ είναι προφανές ότι θα πρέπει να εξαναγκάσουμε την προσεγγιστική λύση $U_\ell(x, t)$ να ικανοποιεί τις συνθήκες

$$[Du_x] := D^+u_x^+ - D^-u_x^- = 0 \quad , \quad \text{για } x = w_k \quad , \quad k = 1, 2 \quad . \quad (33)$$

ισοδύναμα:

$$\gamma \phi_{2i}(x_i^-) = \phi_{2i}(x_i^+) \quad , \quad i = N_1 + 1 \quad (34)$$

και

$$\gamma \phi_{2i}(x_i^-) = \phi_{2i}(x_i^+) \quad , \quad i = N_1 + 2 + 1 \quad (35)$$

Οπότε στα σημεία διεπαφής, και προκειμένου να ικανοποιήσουμε τις ασυνέχειες, ορίζουμε ως νέα γενικευμένα ασυνεχή Hermite τις συναρτήσεις βάσης $\phi_{\ell,2j_\ell}(x)$ ως εξής:

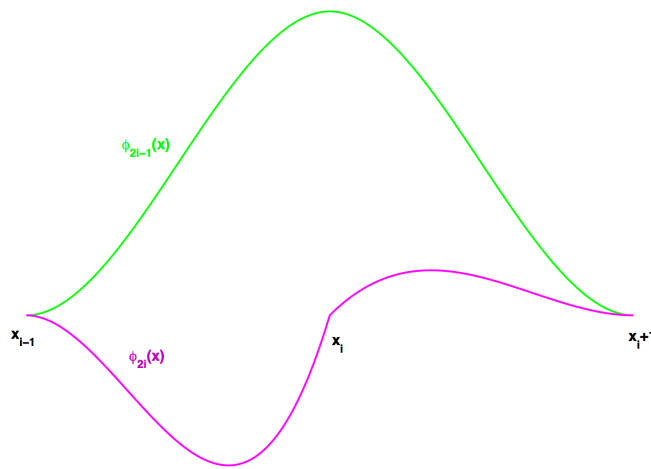
1η -2η περιοχή :

$$\phi_{2i}(x) = \begin{cases} -\frac{h}{\gamma} \psi\left(\frac{x_i-x}{h}\right) & , x \in I_{\ell,j_\ell-1} \\ h \psi\left(\frac{x-x_i}{h}\right) & , x \in I_{\ell,j_\ell} \\ 0 & , \text{διαφορετικά} \end{cases} \quad i = N_1 + 2 \quad (36)$$

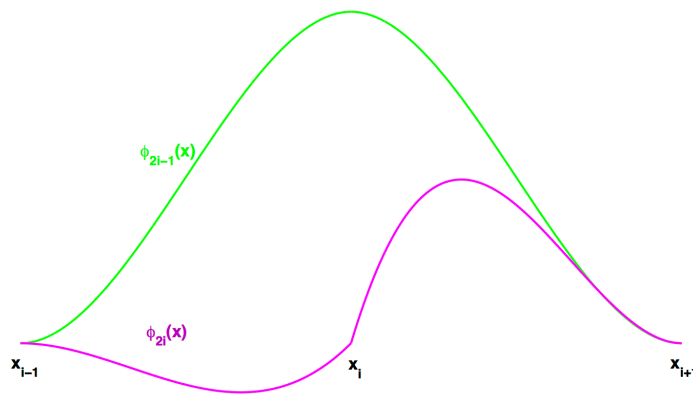
2η -3η περιοχή :

$$\phi_{2i}(x) = \begin{cases} -h_\ell \psi \left(\frac{x_{\ell, j_\ell-1} - x}{h_\ell} \right) & , x \in I_{\ell, j_\ell-1} \\ \frac{h_\ell}{\gamma} \psi \left(\frac{x - x_{\ell, j_\ell-1}}{h_\ell} \right) & , x \in I_{\ell, j_\ell}, \quad j_\ell = N_1 + N_2 + 2 \\ 0 & , \text{διαφορετικά} \end{cases} \quad (37)$$

Μία γραφική αναπαράσταση, για $\gamma = 0.3$, των παραπάνω συναρτήσεων παρατίθεται στα σχήματα που ακολουθούν:



Σχήμα 4: Κυβικά πολυώνυμα Hermite



Σχήμα 5: Κυβικά πολυώνυμα Hermite

Χρησιμοποιώντας τώρα τη νέα βάση Hermite, και αντικαθιστώντας την προσεγγιστική λύση (29) στη διαφορική εξίσωση προκύπτει ότι:

	<p>ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ <i>επένδυση στην κοινωνία της γνώσης</i></p>	
<p>Ευρωπαϊκή Ένωση Ευρωπαϊκό Κοινωνικό Ταμείο</p>	<p>ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ</p>	<p>πρόγραμμα για την ανάπτυξη ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ</p>
<p>Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης</p>		

$$\sum_{j=1}^{N+1} [\dot{\alpha}_{2j-1}(t)\phi_{2j-1}(x) + \dot{\alpha}_{2j}(t)\phi_{2j}(x)] = D \sum_{j=1}^{N+1} [\alpha_{2j-1}(t)\phi_{2j-1}''(x) + \alpha_{2j}(t)\phi_{2j}''(x)], \quad (38)$$

όπου με τη τελεία δηλώνουμε τη πρώτη παράγωγο ως προς το χρόνο. Εάν τώρα απαιτήσουμε η παραπάνω σχέση να επαληθεύεται ακριβώς στα δύο εσωτερικά σημεία Gauss του κάθε υποδιαστήματος, οι στοιχειώδεις collocation εξισώσεις που αντιστοιχούν στο στοιχείο I_j , $j = 1, \dots, N$ παίρνουν τη μορφή πίνακα:

$$C_j^{(0)} \begin{bmatrix} \dot{\alpha}_{2j-1} \\ \dot{\alpha}_{2j} \\ \dot{\alpha}_{2j+1} \\ \dot{\alpha}_{2j+2} \end{bmatrix} = \frac{D}{h^2} C_j^{(2)} \begin{bmatrix} \alpha_{2j-1} \\ \alpha_{2j} \\ \alpha_{2j+1} \\ \alpha_{2j+2} \end{bmatrix} \quad (39)$$

όπου τα h και D έχουν οριστεί στις εξισώσεις (26) και (21) και

$$C_j^{(0)} := \begin{bmatrix} s_1^{(0)} & \frac{h}{\alpha_i} s_2^{(0)} & s_3^{(0)} & -\frac{h}{\beta_i} s_4^{(0)} \\ s_3^{(0)} & \frac{h}{\alpha_i} s_4^{(0)} & s_1^{(0)} & -\frac{h}{\beta_i} s_2^{(0)} \end{bmatrix} \quad (40)$$

όπου $s_1^{(0)} = \frac{9+4\sqrt{3}}{18}$, $s_2^{(0)} = \frac{3+\sqrt{3}}{36}$, $s_3^{(0)} = \frac{9-4\sqrt{3}}{18}$, $s_4^{(0)} = \frac{3-\sqrt{3}}{36}$, ενώ

$$C_j^{(2)} := \begin{bmatrix} s_1^{(2)} & \frac{h}{\alpha_i} s_2^{(2)} & s_3^{(2)} & -\frac{h}{\beta_i} s_4^{(2)} \\ s_3^{(2)} & \frac{h}{\alpha_i} s_4^{(2)} & s_1^{(2)} & -\frac{h}{\beta_i} s_2^{(2)} \end{bmatrix} \quad (41)$$

με $s_1^{(2)} = -2\sqrt{3}$, $s_2^{(2)} = -1 - \sqrt{3}$, $s_3^{(2)} = 2\sqrt{3}$ και $s_4^{(2)} = -1 + \sqrt{3}$.

Οι σταθερές α_i και β_i χρησιμοποιούνται για να ξεχωρίζουμε τους στοιχειώδεις πίνακες για τα στοιχεία I_{N_1} και $I_{N_1+N_2+1}$, αφού ως συναρτήσεις βάσης για τα στοιχεία αυτά χρησιμοποιούνται αντίστοιχα οι εξισώσεις (36)-(37). Δηλαδή

$$\alpha_i = \begin{cases} 1 & , \quad i \neq N_1 + N_2 + 1 \\ \gamma & , \quad i = N_1 + N_2 + 1 \end{cases} \quad (42)$$

και

$$\beta_i = \begin{cases} 1, & i \neq N_1 \\ \gamma, & i = N_1 \end{cases} \quad (43)$$

Για να παράξουμε τις συνοριακές collocation εξισώσεις απαιτούμε η προσεγγιστική λύση $U(x,t)$ να ικανοποιεί τις συνοριακές συνθήκες σε κάθε χρονικό βήμα, και ως εκ τούτου

$$\dot{\alpha}_2 = 0, \quad \alpha_2 = 0, \quad \dot{\alpha}_{2N+2} = 0, \quad \alpha_{2N+2} = 0. \quad (44)$$

Συγκεντρώνοντας τις εσωτερικές στοιχειώδεις και συνοριακές collocation εξισώσεις καταλήγουμε στην επίλυση του collocation συστήματος:

$$A\dot{\mathbf{a}} = B\mathbf{a} \quad (45)$$

όπου $\dot{\mathbf{a}} = [\dot{\alpha}_1 \quad \dot{\alpha}_3 \cdots \dot{\alpha}_{2N+1}]^T$, $\mathbf{a} = [\alpha_1 \quad \alpha_3 \cdots \alpha_{2N+1}]^T$ και

$$A = \begin{bmatrix} \tilde{A}_1 & B_1 & & & & \\ & A_2 & B_2 & & & \\ & & & \backslash & & \\ & & & A_{N-1} & B_{N-1} & \\ & & & & A_N & \tilde{B}_N \end{bmatrix}, \quad (46)$$

$$B = \begin{bmatrix} \tilde{F}_1 & G_1 & & & & \\ & F_2 & G_2 & & & \\ & & & \backslash & & \\ & & & F_{N-1} & G_{N-1} & \\ & & & & F_N & \tilde{G}_N \end{bmatrix}. \quad (47)$$

με A_j, B_j, F_j και G_j 2×2 πίνακες οι οποίοι ορίζονται απο τις σχέσεις (40) και

$$[A_j \quad B_j] = C_j^{(0)} \quad \text{και} \quad [F_j \quad G_j] = \frac{D}{h^2} C_j^{(2)}. \quad (48)$$

Οι 2×1 πίνακες \tilde{A}_1, \tilde{F}_1 προκύπτουν από τους αντίστοιχούς πίνακες A_1 και F_1 αν παραλείψουμε τη δεύτερη στήλη. Αντίστοιχα οι πίνακες \tilde{B}_N, \tilde{G}_N προκύπτουν από τους αντίστοιχούς πίνακες B_N και G_N αν παραλείψουμε τη πρώτη στήλη.

Το σύστημα (45) μπορεί να θεωρηθεί και ως σύστημα συνήθων διαφορικών εξισώσεων της μορφής:

$$\dot{\mathbf{a}} = C(t, \mathbf{a}) \quad (49)$$

όπου $C(t, \mathbf{a}) = A^{-1}B\mathbf{a}$.



3 Αριθμητικά Αποτελέσματα Υλοποιήσεων

3.1 Αποτελέσματα υλοποίησης σε ένα GPU Shared-memory υπολογιστικό σύστημα

Το HP SL390s G7 είναι ένα υπολογιστικό σύστημα αρχιτεκτονικής κοινής μνήμης, που αποτελείται από 6-πυρήνους τύπου Xeon X5660@2.8GHz επεξεργαστές με 12 MB Level 3 μνήμης cache. Η συνολική μνήμη είναι 24 GB και το λειτουργικό σύστημα είναι Oracle Linux, έκδοσης 6.2. Επίσης, διαθέτει γραφικό υποσύστημα τύπου GPU Tesla M2070 της σειράς Fermi [30], συνδεδεμένο μέσω διαύλου PCI-gen2. Η GPU έχει 6GB μνήμης και 448 πυρήνες σε 14 πολυεπεξεργαστές. Η εφαρμογή αναπτύχθηκε σε γλώσσα προγραμματισμού Fortran, διπλής ακρίβειας, χρησιμοποιώντας τα πρότυπα OpenMP [31, 32] και OpenACC [33] με μεταγλωτιστές PGI έκδοσης 12.9 [34]. Για τις βασικές πράξεις γραμμικής άλγεβρας, χρησιμοποιούνται υλοποιήσεις διαδικασιών από τις επιστημονικές βιβλιοθήκες BLAS και LAPACK [35].

Στην υλοποίηση του παραπάνω παράλληλου αλγόριθμου, επιλύεται το πρόβλημα δοκιμής Dirichlet Modified Helmholtz με $\lambda = 1$, το οποίο έχει την παρακάτω αναλυτική λύση

$$u(x, y) = 10 \phi(x) \phi(y), \quad \phi(x) = e^{-100(x-0.1)^2} (x^2 - x),$$

Ο ακόλουθος πίνακας παρουσιάζει την συμπεριφορά της μεθόδου για διακριτοποίηση μεγέθους έως και 2048 πεπερασμένων στοιχείων ανά κατεύθυνση, με αναφορά στα απαιτούμενα βήματα σύγκλισης και την Ευκλείδεια νόρμα του σφάλματος του γραμμικού συστήματος.

n_s	Επαναληπτικά βήματα	$\ b - Ax^{(m)}\ _2$
256	294	6.06e-11
512	589	2.85e-11
1024	1161	1.39e-11
2048	3726	9.59e-12

Στη μελέτη της συμπεριφοράς εκτέλεσης του παράλληλου αλγορίθμου για την υλοποίηση στο συγκεκριμένο υπολογιστικό περιβάλλον, έγινε συλλογή μετρήσεων των χρόνων εκτέλεσης για διάφορες παραμέτρους του προβλήματος, όπως η χρήση του γραφικού υποσυστήματος υπολογισμών και του πληθους των CPU πυρήνων. Οι παρακάτω πίνακες εμφανίζουν αυτές τις μετρήσεις σε δευτερόλεπτα για διαφορετικού μεγέθους προβλήματα με διακριτοποιήσεις από $n_s = 256$ έως και $n_s = 2048$.



$n_s = 256$		CPU	GPU + CPU
CPU cores	Total Time	Total Time	Total Time
1	12.24		11.18
2	8.12		7.53
4	4.87		5.63
$n_s = 512$		CPU	GPU + CPU
CPU cores	Total Time	Total Time	Total Time
1	88.83		71.25
2	52.94		46.59
4	34.25		32.63
$n_s = 1024$		CPU	GPU + CPU
CPU cores	Total Time	Total Time	Total Time
1	750.35		549.82
2	448.76		352.95
4	283.14		250.64
$n_s = 2048$		CPU	GPU + CPU
CPU cores	Total Time	Total Time	Total Time
1	9176		6770
2	5001		4387
4	2999		2839

Στο παρακάτω πίνακα εμφανίζονται λεπτομερώς οι μετρήσεις κάθε χρόνου εκτέλεσης για τις περιπτώσεις χρήσης της GPU. Οι χρόνοι μεταφοράς δεδομένων μεταξύ μνήμης των CPU και GPU και αντίστροφα παρουσιάζονται για όλα τα προβλήματα διακριτοποιήσεων. Επίσης μετρήθηκαν οι χρόνοι εκτέλεσης για κάθε διαθέσιμο αριθμό CPU υπολογιστικών πυρήνων.

n_s	GPU - CPU Comm. Time	Computation Time		
		1 Core	2 Cores	4 Cores
256	1.52	9.66	6.01	4.11
512	6.81	64.44	39.78	25.82
1024	44.3	505.5	308.6	206.3
2048	541	6229	3846	2298

Πρέπει να αναφερθεί ότι ο χρόνος επικοινωνίας μεταξύ των CPU και GPU είναι ανεξάρτητος του αριθμού των πυρήνων της CPU, διότι στον αλγόριθμο η διαδικασία μεταφοράς των δεδομένων μεταξύ μνήμης CPU) και GPU πραγματοποιείται από ένα μόνο υπολογιστικό thread για λόγους ισοκατανομής του υπολογιστικού φορτίου. Έτσι αποφεύγονται φαινόμενα deadlock των δεδομένων κατά τη κίνηση τους μεταξύ πολλαπλών CPU threads προς τη μνήμη του γραφικού υποσυστήματος διαμέσου του διαύλου PCI του μηχανήματος.



3.1.1 Συμπεράσματα υλοποίησης

Σε αυτή την ερευνητική δραστηριότητα αναπτύχθηκε ένας νέος παράλληλος αλγόριθμος για την υλοποίηση της επαναληπτικής μεθόδου BiCGSTAB, με σκοπό την επίλυση των εξισώσεων Hermite Collocation που προκύπτουν από ελλειπτικού τύπου μερικές διαφορικές εξισώσεις. Ο αλγόριθμος υλοποιήθηκε σε πολυεπεξεργαστικά συστήματα με γραφικά υποσυστήματα. Η απόδοση του αλγορίθμου εξαρτάται από το μέγεθος του προβλήματος και τον αριθμό των διαθέσιμων επεξεργαστών. Παρατηρήθηκε βελτίωση της απόδοσης έως και περίπου 30%.

3.2 Η ασυνεχής Collocation σε 3 περιοχές

Τα αριθμητικά αποτελέσματα των μεθόδων που αναπτύχθηκαν στην παρούσα Τεχνική Έκθεση και αφορούν ιατρικές εφαρμογές, παρουσιάζονται αναλυτικά στην Τεχνική Έκθεση 4.2 Έτος 2012. Για να επιβεβαιώσουμε όμως την τέταρτη τάξη σύγκλισης της dDHC παραθέτουμε τα παρακάτω αποτελέσματα που αναφέρονται στο πρόβλημα (19) με τιμές των παραμέτρων ως ακολούθως:

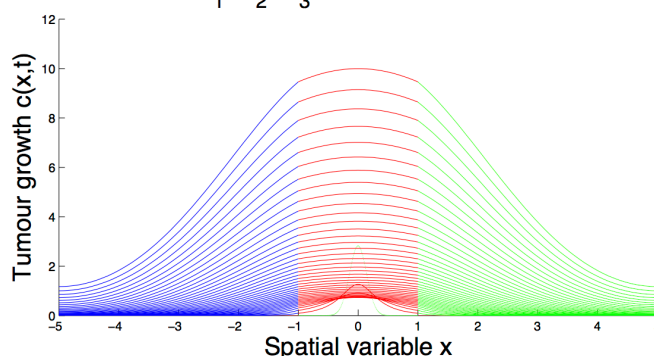
$$a = -5, w_1 = -1, w_2 = 1, b = 5, \gamma = 0.5$$

και

$$f(x) = \frac{1}{\eta\sqrt{\pi}} e^{-x^2/\eta^2}.$$

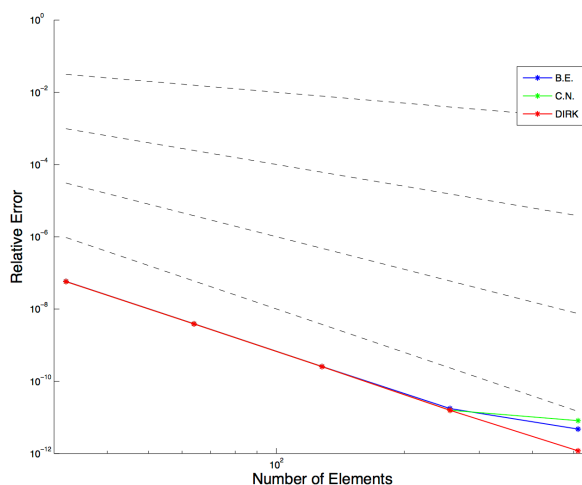
Στο σχήμα 6 απεικονίζεται το προφίλ της λύσης, ενώ στο σχήμα 6 επιβεβαιώνε-

$$N_1=N_2=N_3=256 \quad \tau=0.1 \quad t_{max}=4$$



Σχήμα 6: Η απόδοση της μεθόδου (dDHC)/(BE)

ται η τέταρτη τάξη σύγκλισης της μεθόδου με διάφορα σχήματα διακριτοποίησης του χρόνου.



Σχήμα 7: Τάξη σύγκλισης της dDHC μεθόδου.

4 Παραδοτέα

- Παπαδομανωλάκη Μαρία ” Η μέθοδος Collocation για παραβολικές μερικές διαφορικές εξισώσεις με ασυνεχή συντελεστή διάχυσης: στην κατεύθυνση προσομοίωσης καρκινικών όγκων εγκεφάλου.” Διδακτορική Διατριβή 2012.
- Ανάπτυξη λογισμικού σε προγραμματιστικό περιβάλλον MATLAB και FORTRAN
- Η παρούσα Τεχνική Έκθεση

5 Συνεργασίες

Η παρούσα έρευνα πραγματοποιήθηκε από η ερευνητική ομάδα του Πολυτεχνείου Κρήτης (ΚΕΟ1) αποτελούμενη από τους καθ. Ι. Σαριδάκη και καθ. Ε. Παπαδοπούλου, επ.καθ. Ε. Μαθιουδάκη, Δρ. Μ. Παπαδομανωλάκη και τον υποψήφιο διδάκτορα Ι. Αθανασάκη.

6 Μελλοντικές Δράσεις

- Επέκταση της μεθόδου collocation με ασυνεχή στοιχεία Hermite σε N περιοχές και αρχικές πηγές για τις $1 + 1$ διαστάσεις.
- Ανάπτυξη της dDHC/HC για μη γραμμικά προβλήματα $1 + 1$ διαστάσεων



- Ανάπτυξη της dDHC για προβλήματα στις $2 + 1$ διαστάσεις

Αναφορές

- [1] R. Alexander “Diagonally Implicit Runge-Kutta Methods for stiff ODE’s,” *SIAM Num. Anal.*, vol. 14 no. 6, pp. 1006-1021, 1977.
- [2] C. de Boor and B. Swartz “Collocation at Gaussian points,” *SIAM Num. Anal.*, vol.10, pp. 582-606, 1973.
- [3] P.K.Burgess,P.M.Kulesa,J.D.Murray,and E.C.Alvord Jr. “The interaction of growth rates and diffusion coefficients in a three-dimensional mathematical model of gliomas,” *Journal of Neuropathology and Experimental Neurology* , vol.56, no. 6, pp.704-713, 1997.
- [4] J.C.Butcher “Implicit Runge-Kutta processes ,” *Math.Comp.*, vol.18, pp.50-64, 1964.
- [5] S.H.Brill and G.F.Pinder, “Parallel implementation of the bi-cgstab method with block red-black gauss-seidel preconditioner applied to the hermite collocation discretization of partial differential equations,” *Parallel Computing*, vol. 28, pp. 399–414, 2002.
- [6] M. Crouzeix“ Sur l’approximation des equations differentielles operationnelles lineaires par des methodes de Runge Kutta,” *PHD Thesis, Univercity Paris VI, Paris,1975.*
- [7] J.C.Butcher “The numerical analysis of ordinary differential equations ,” *John Wiley* , 1987.
- [8] C.C.Christara, “Parallel solvers for spline collocation equations,” *Advances in Engineering Software*, vol. 27, pp. 71–89, 1996.
- [9] G.C.Cruywagen,D.E.Woodward,P.Tracqui,G.T.Bartoo,J.D.Murray,and E.C.Alvord Jr. “The modeling of diffusive tumours,” *Journal of Biological Systems* , vol.3, pp.937-945, 1995.
- [10] J. Dongarra, I. Duff, D. Sorensen, and H. van der Vorst, *Numerical Linear Algebra for high-performance computers*. Phil.: SIAM, 1998.
- [11] C. E. Houstis, E. N. Houstis, and J. Rice, “Pde computations: Methods and performance evaluation,” *Par. Comp.*, vol. 5, pp. 141–163, 1997.



- [12] E. Mathioudakis, E. Papadopoulou, and Y. Saridakis, "Iterative solution of elliptic collocation systems on a cognitive parallel computer," *Computers and Maths with Appl.*, vol. 48, pp. 951–970, 2004.
- [13] —, "Mapping parallel iterative algorithms for pde computations on a distributed memory computers," *Parallel Algorithms and Applications*, vol. 8, pp. 141–154, 1996.
- [14] —, "Bi-cgstab for collocation equations on distributed memory parallel computers," *Numerical Mathematics and advanced applications - ENUMATH 2001*, Springer, pp. 957–966, 2003.
- [15] E. Mathioudakis, E. Papadopoulou, and Y. Saridakis, "Preconditioning for solving hermite collocation by the bi-cgstab," *WSEAS Trans. on Mathematics*, vol. (5),7, pp. 811–816, 2006.
- [16] E. Mathioudakis and E. Papadopoulou, "Grid computing for the bi-cgstab applied to the solution of the modified helmholtz equation," *Int. J. App. Maths and comp. sciences*, vol. (4),3, pp. 179–184, 2007.
- [17] A. R. Mitchell, D.F. Griffiths "The Finite Difference Method in Partial Differential Equations," *John Willey & Sons*, 1980.
- [18] M.G. Papadomanolaki and Y.G. Saridakis "Collocation with discontinuous Hermite elements for a tumour invasion model with heterogeneous diffusion in 1+1 dimensions," in *Proceedings of Numan2010 conference "Conference in Numerical Analysis"*, pp. 238-245, 2010.
- [19] M.G. Papadomanolaki "The collocation method for parabolic differential equations with discontinuous diffusion coefficient: in the direction of brain tumour simulations", *PhD Thesis*, Technical University of Crete, 2012 (in Greek)
- [20] T. Papatheodorou, "Block aor iteration for nonsymmetric matrices," *Math. Comp.*, vol. (41),164, pp. 511–525, 1983.
- [21] Y. Saad, *Iterative methods for sparse linear systems*. SIAM, 2003.
- [22] G.D. Smith "Numerical solution of partial equations:finite difference methods(third edition)," *Oxford University Press*, 1985.
- [23] K.R.Swanson "Mathematical modelling of the growth and control of tumour," *PHD Thesis*, University of Washington, 1999.



- [24] K.R.Swanson, E.C.Alvord Jr and J.D.Murray “A quantitative model for differential motility of gliomas in grey and white matter,” *Cell Proliferation*, vol.33, pp.317-329, 2000.
- [25] K.R.Swanson,C.Bridge,J.D.Murray and E.C.Alvord Jr “ Virtual and real brain tumours:using mathematical modeling to quantify glioma growth and invasion,” *J.Neurol.Sci*, vol.216, pp.1-10, 2003.
- [26] P.Tracqui,G.C.CruywagenG,D.E.Woodward,T.Bartoo, J.D.Murray and E.C.Alvord Jr. “A mathematical model of glioma growth:The effect of chemotherapy on spatio-temporal growth,” *Cell Proliferation*, vol.28, pp.17-31, 1995.
- [27] R. Varga, *Matrix Iterative Analysis*, New York: Springer Verlag, 2000.
- [28] H. der Vorst, “Bi-CGSTAB : A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems,” *SIAM J. Sci.Stat.Comp.*, vol. 13, pp. 631–644, 1992.
- [29] D.E.Woodward,J.Cook,P.Tracqui,G.C.Cruywagen,J.D.Murray,and E.C.Alvord Jr.“A mathematical model of glioma growth: the effect of extent of surgical resection,” *Cell Proliferation*, vol.29, pp.269-288, 1996.
- [30] <http://www.nvidia.com/object/tesla-servers.html>.
- [31] C. Rohit, *Parallel programming with OpenMP*. M. K., 2001.
- [32] <http://www.openmp.org>.
- [33] <http://www.openacc.org>.
- [34] <http://www.pgroup.com>.
- [35] <http://www.netlib.org>.