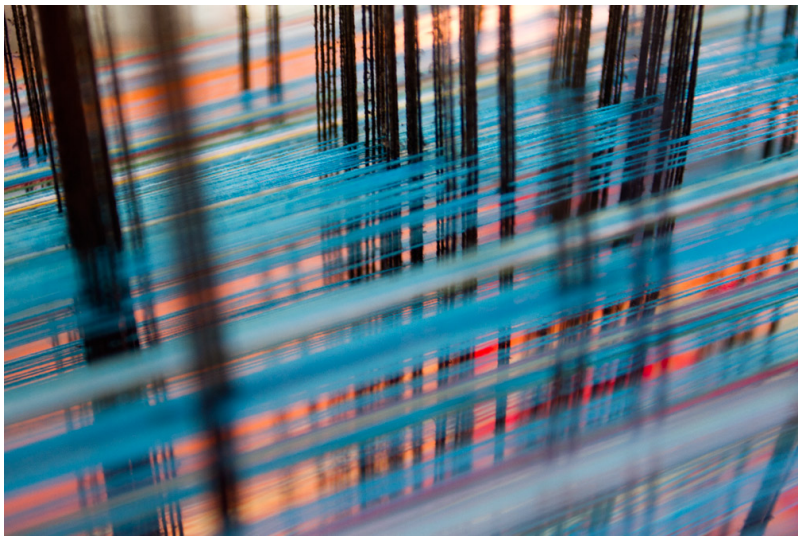
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Σχολή Θετικών Επιστημών, Τμήμα Φυσικής

Εργασία στην OpenMP

Κίμων Ζαγκούρης
A.E.M. 4353
Email: kzagko@gmail.com

16 Μαρτίου 2019



1 Ανάλυση

Στην εργασία αυτή βρήκαμε την λύση του ελλειπτικού προβλήματος συνοριακών τιμών που καθορίζεται από τη διαφορική εξίσωση,

$$u_{xx} + u_{yy} = -10\left(x^2 + y^2 + 5\right),$$

με την χρήση των επαναληπτικών μεθόδων Liebmann και SOR.
η ανάλυση εκτελέστηκε σε επεξεργαστή τα χαρακτηριστικά του οποίου αναφέρονται στον Πίνακα 1.1. Ο συγκεκριμένος επεξεργαστής έχει 32 φυσικούς πυρήνες και άλλους 32 εικονικούς (multithreading).

Processor name	Intel(R) Xeon(R) Gold 6140
Packages(sockets)	2
Cores	36
Processors(CPUs)	72
Cores per package	18
Threads per core	2

Πίνακας 1.1: Χαρακτηριστικά επεξεργαστή.

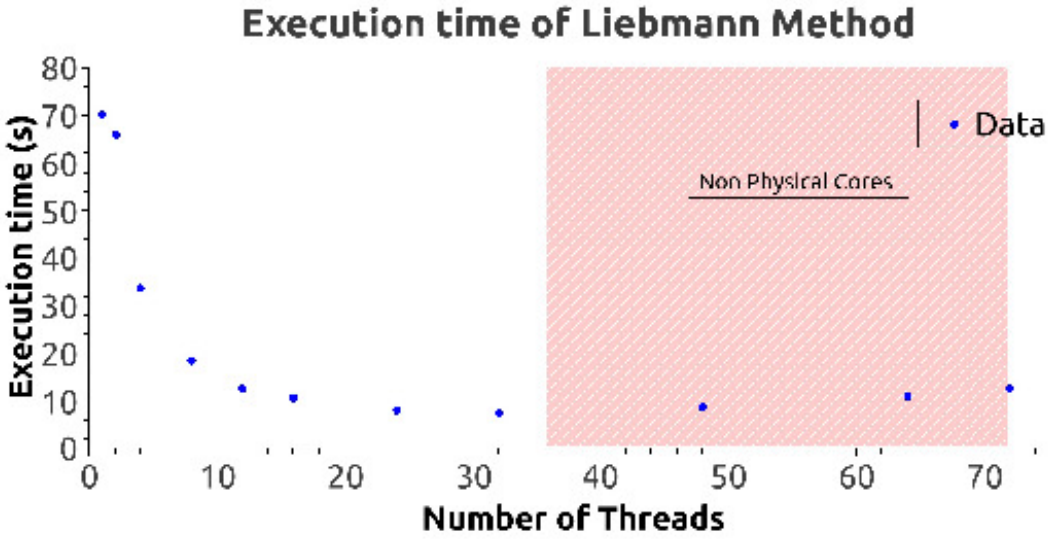
Παρακάτω είναι τα αποτελέσματα της ανάλυσής μας.
Στον Πίνακα 1.2 έχουμε τα αποτελέσματα από την μέθοδο Liebmann.

Time(s)	Threads	Tolerance	Itterations	Central Value
70.3677	1	9.99991e-08	208010	4.35178
66.0042	2	9.99991e-08	208010	4.35178
33.781	4	9.99991e-08	208010	4.35178
18.4393	8	9.99991e-08	208010	4.35178
12.7887	12	9.99991e-08	208010	4.35178
10.7612	16	9.99991e-08	208010	4.35178
8.21691	24	9.99991e-08	208010	4.35178
7.62344	32	9.99991e-08	208010	4.35178
8.92122	48	9.99991e-08	208010	4.35178
11.0658	64	9.99991e-08	208010	4.35178
12.6623	72	9.99991e-08	208010	4.35178

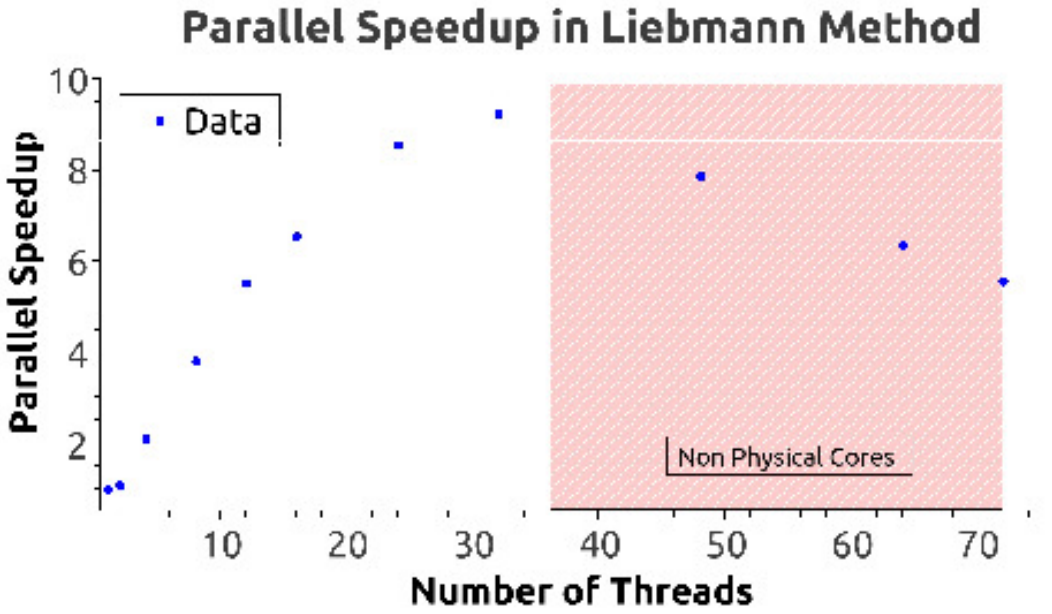
Πίνακας 1.2: Τιμές Ανάλυσης με την μέθοδο Liebmann.

Στα σχήματα 1.1,1.2 και 1.3 έχουμε τον χρόνο εκτέλεσης, την επιτάχυνση παράλληλης επεξεργασίας και την απόδοση παράλληλης επεξεργασίας αντίστοιχα.
Από το Σχήμα 1.1 βλέπουμε ότι ο χρόνος επίλυσης για **έναν** πυρήνα, αν και μεγαλύτερος, είναι παρόμοιος με αυτόν για **δύο**. Αυτό οφείλεται πιθανότατα στο γεγονός ότι ο compiler αυτόματα βελτιστοποιεί τον κώδικα για έναν πυρήνα, πιθανότατα μέσω vectorization των for loops με αποτέλεσμα να χρειάζεται λιγότερο χρόνο εκτέλεσης.
Στο Σχήμα 1.2 βλέπουμε την επιτάχυνση της παράλληλης επεξεργασίας. πως παρατηρούμε το μέγιστο εμφανίζεται περίπου στους 36 πυρήνες που είναι και ο μέγιστος αριθμός φυσικών πυρήνων σε αυτόν τον επεξεργαστή. Η επιτάχυνση βέβαια δεν είναι ίση με τον αριθμ'ο των πυρήνων κυρίως λόγω του ότι υπάρχει Overhead αλλά και γραμμίκο κομμάτι στον κώδικα που δεν μπορεί να παραλληλιστεί. Στο συγκεκριμένο παράδειγμα επίσης, ο φόρτος εργασίας μέσα στα for loops που παραλληλίζουμε είναι σχετικά μικρός και συγκρίσιμος με το κόστος της δημιουργίας των threads. Αυτό φαίνεται πιο αναλυτικά στο Σχήμα 1.4 όπου παρατηρούμε ότι

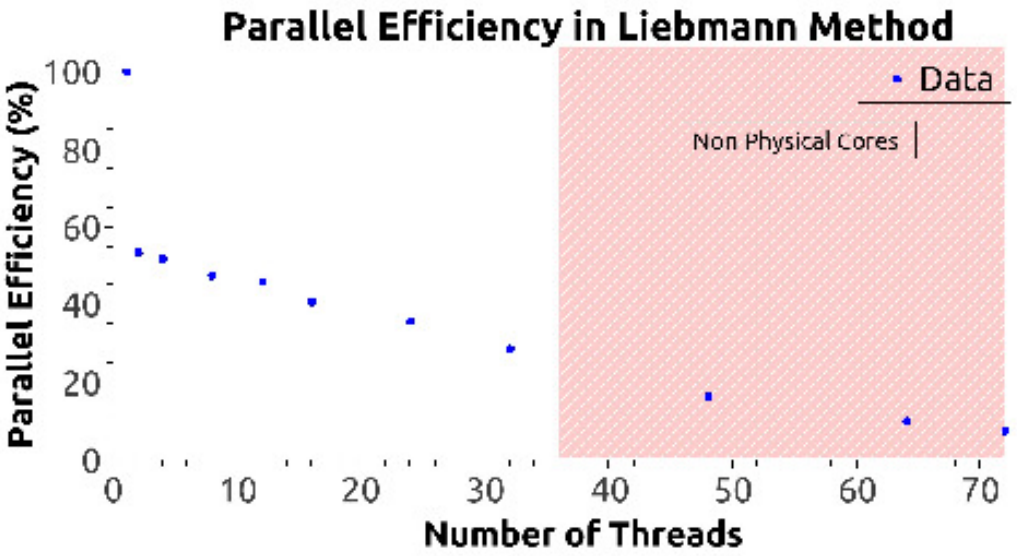
το πρόγραμμά μας δεν τρέχει συνέχεια σε 24 πυρήνες όπως ζητήσαμε αλλά έχει την κατανομή που βλέπουμε με μέσο όρο τους 17 πυρήνες. Με επιπλέον ανάλυση έχουμε το Σχήμα 1.5 στο οποίο παρατηρούμε ότι σημαντικό ποσοστό του χρόνου στα threads δαπανάτε είτε σε αναμονή (spin) είτε σε overhead. Αυτό είναι κυρίως αποτέλεσμα το χαμηλού φόρτου υπολογισμού μέσα στα threads.



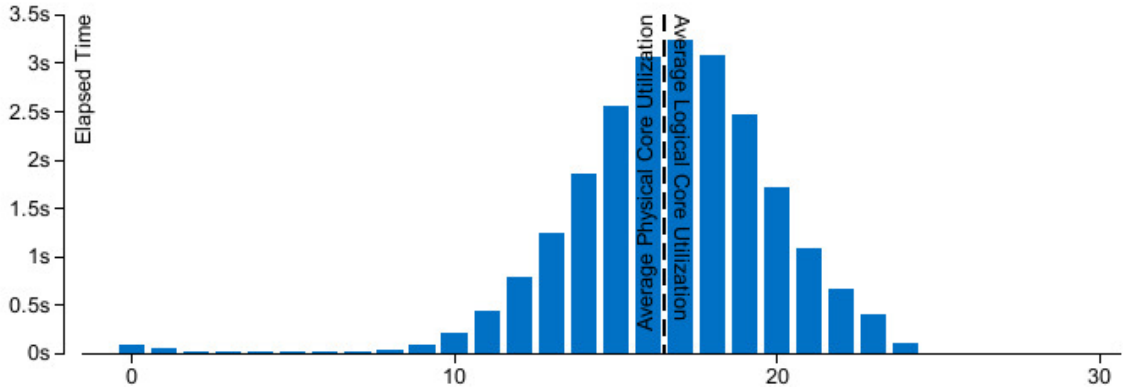
Σχήμα 1.1: Χρόνος εκτέλεσης για διάφορο αριθμό threads.



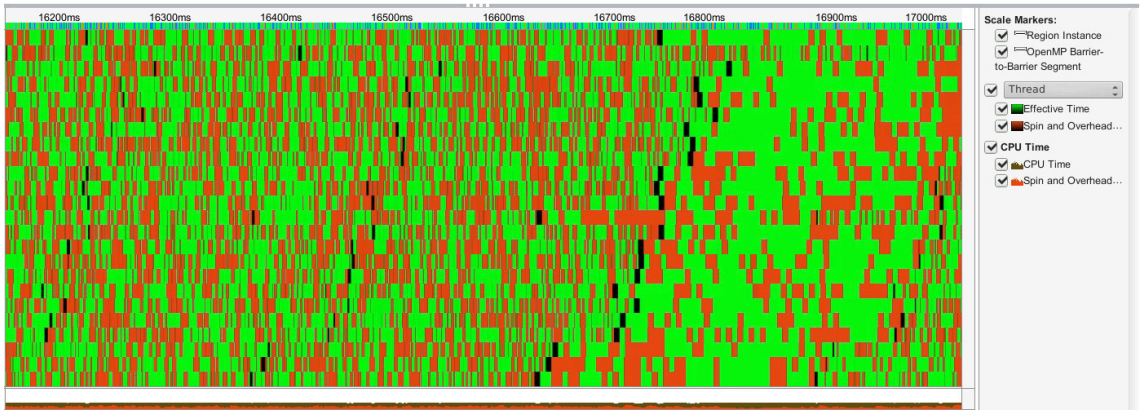
Σχήμα 1.2: Επιτάχυνση παράλληλης επεξεργασίας για διάφορο αριθμό threads.



Σχήμα 1.3: Απόδοση παράλληλης επεξεργασίας για διάφορο αριθμό threads.

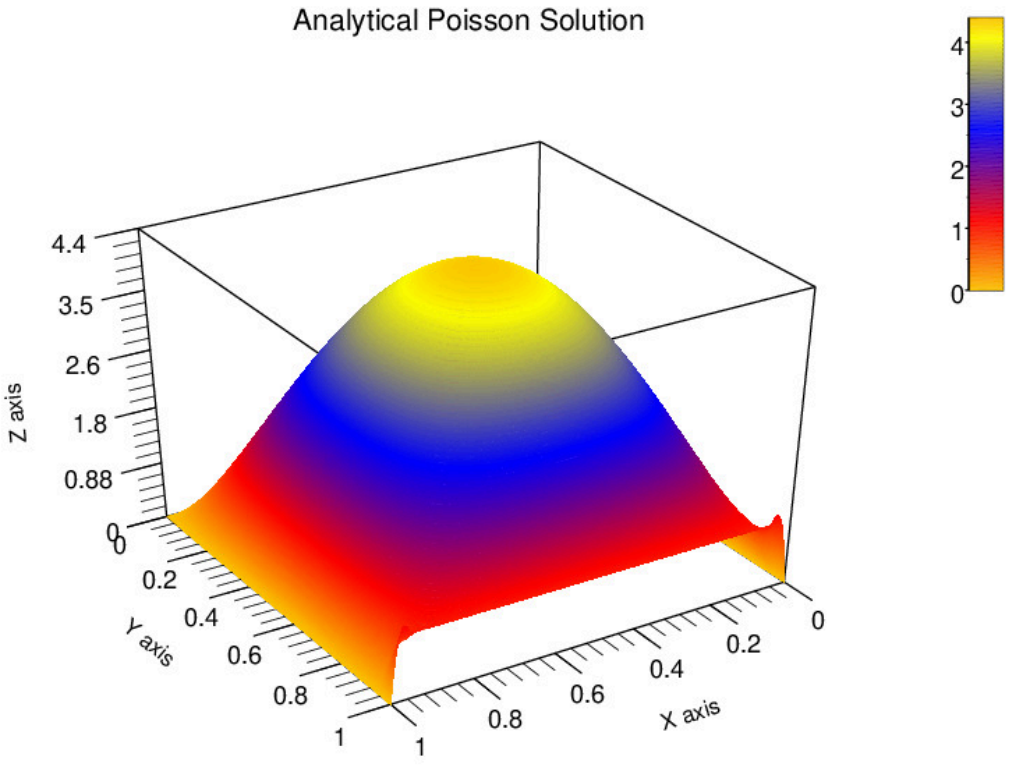


Σχήμα 1.4: Μέση χρήση πυρήνων για εκτέλεση με 24 threads. Στον οριζόντιο άξονα ο αριθμός πυρήνων που δουλεύουν ταυτόχρονα και στον κάθετο άξονα ο χρόνος για τον οποίο δουλεύουν.

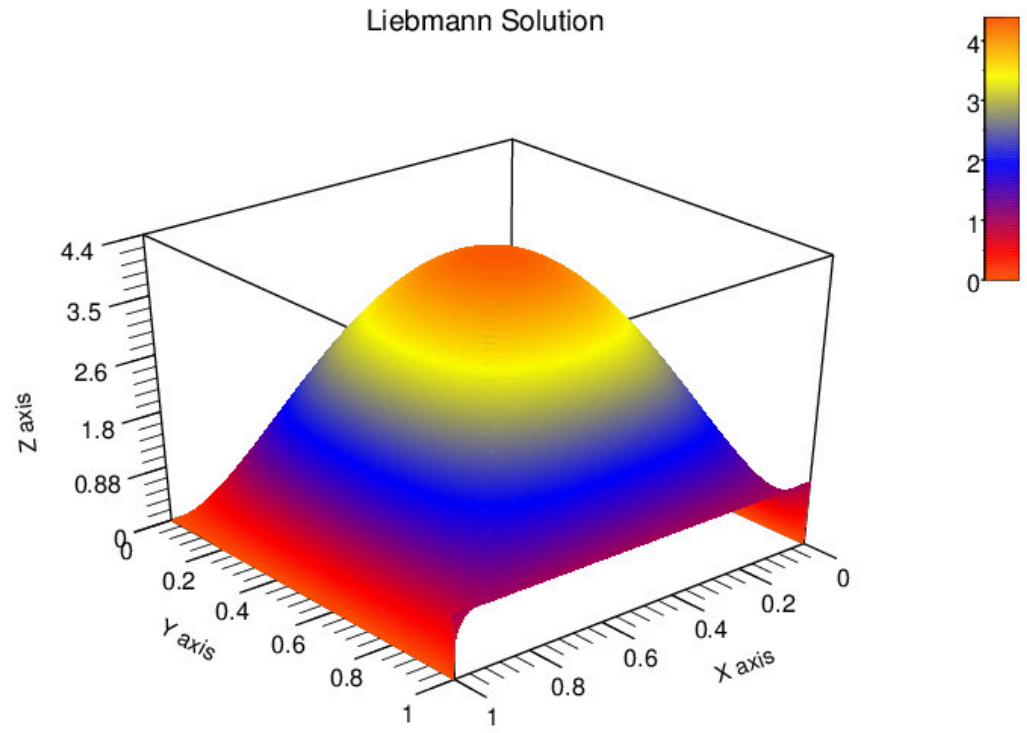


Σχήμα 1.5: Ανάλυση του χρόνου εργασίας 24 threads. Στον οριζόντιο άξονα ο χρόνος εκτέλεσης από 16-17 δευτερόλεπτα και στον κάθετο άξονα τα 24 threads. Με πράσινο χρώμα ο χρόνος που τα threads εκτελούν εργασία και με πορτοκαλί χρώμα ο χρόνος που τα threads αναμένουν ή έχουν overhead.

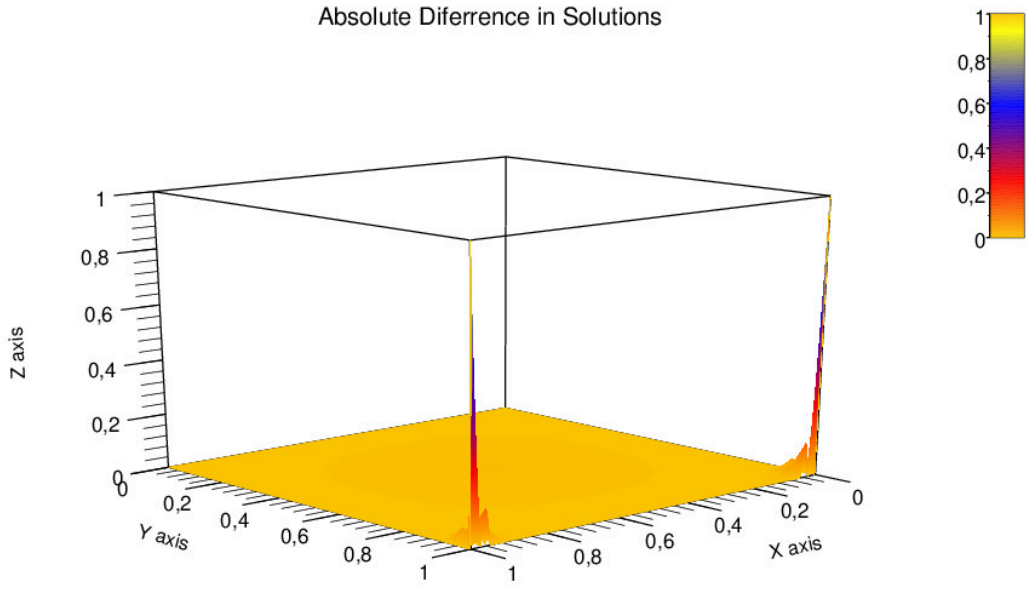
Στα Σχήματα 1.6 και 1.7, έχουμε την λύση της εξίσωσης Poisson. Ενώ στα Σχήματα 1.8 και 1.9 έχουμε τις διαφορές των λύσεων όπου βλέπουμε ότι υπάρχουν μεγάλες διαφορές στα άκρα της αριθμητικής λύσης καθώς και μικρότερες στο κέντρο.



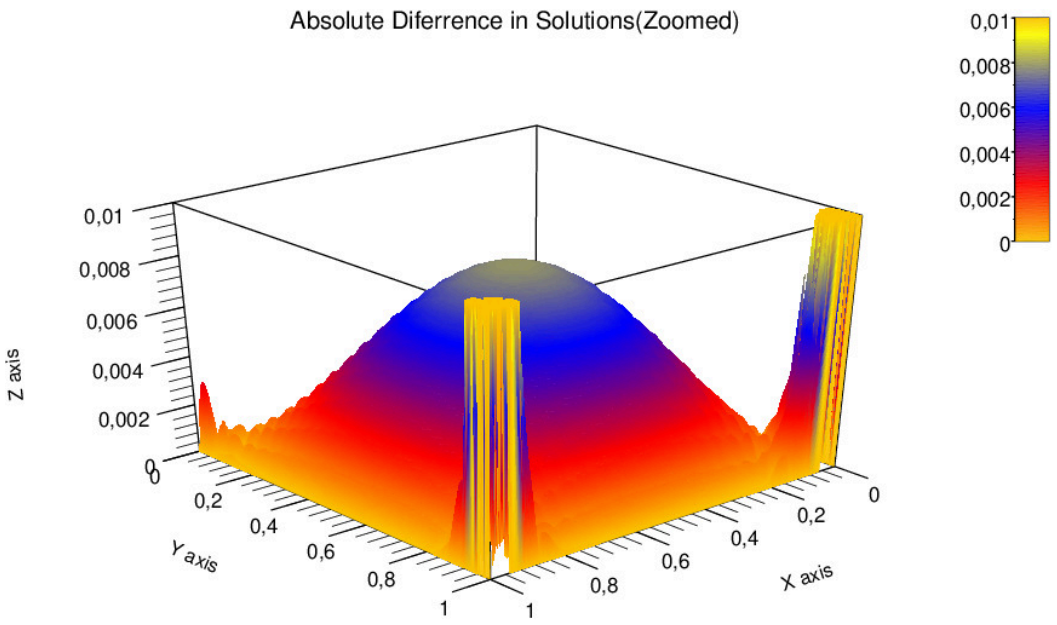
Σχήμα 1.6: Αναλυτική λύση της εξίσωσης Poisson.



Σχήμα 1.7: Αριθμητική λύση της εξίσωσης Poisson.

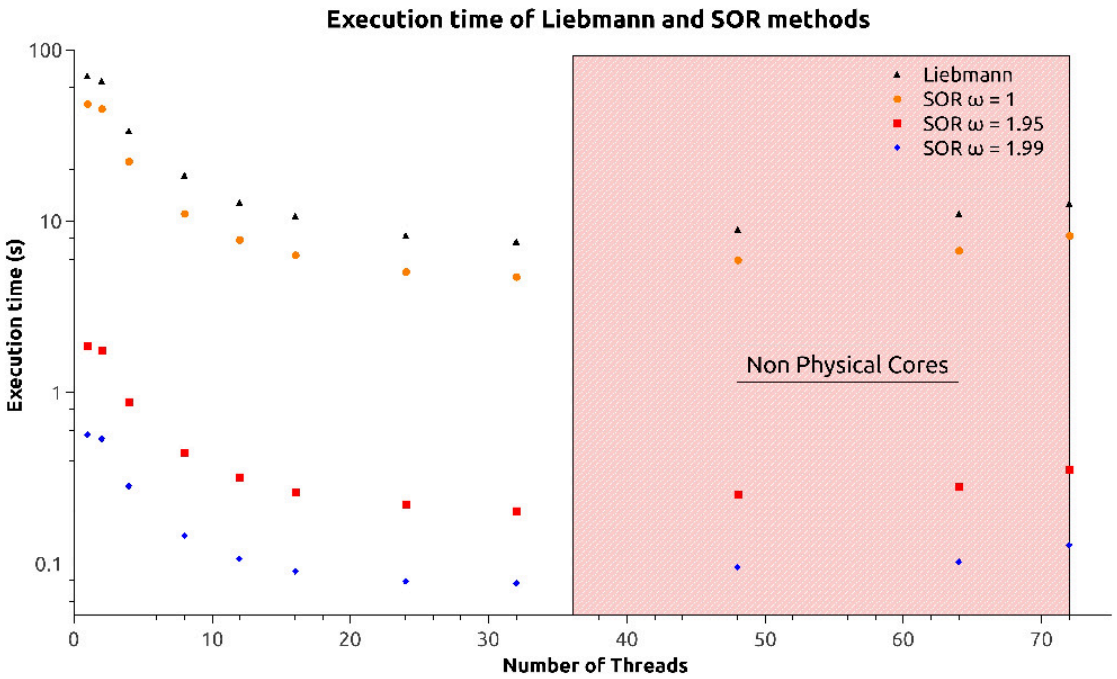


Σχήμα 1.8: Απόλυτη διαφορά των λύσεων.

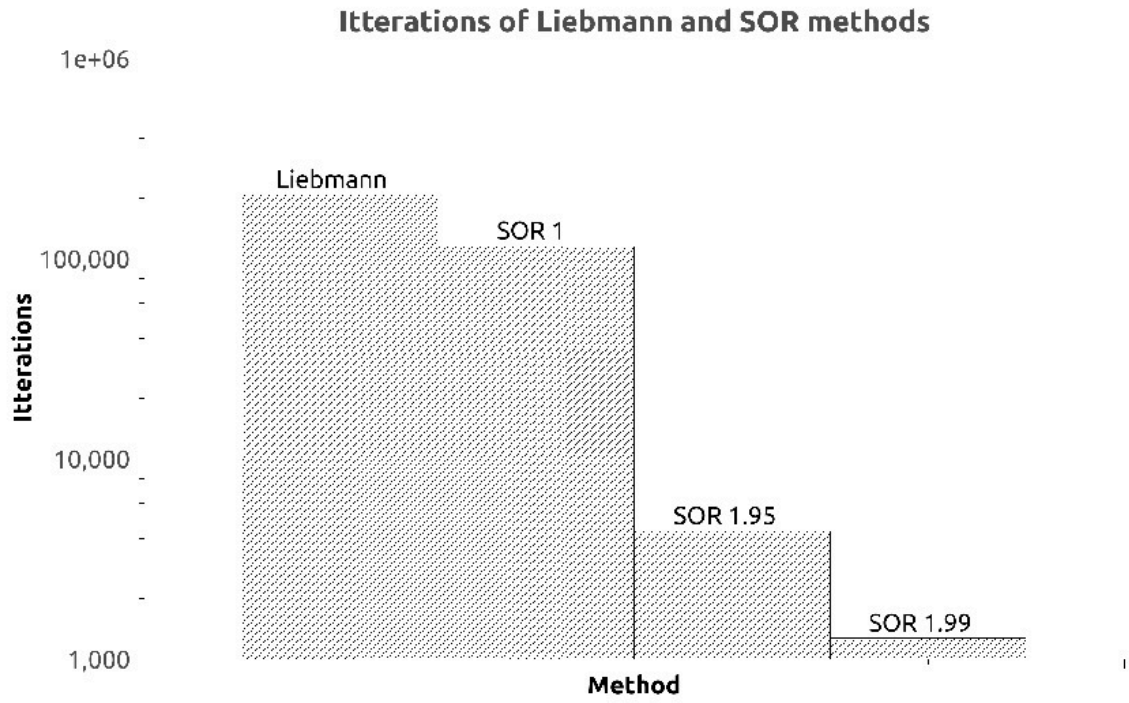


Σχήμα 1.9: Μεγέθυνση της απόλυτης διαφοράς των λύσεων.

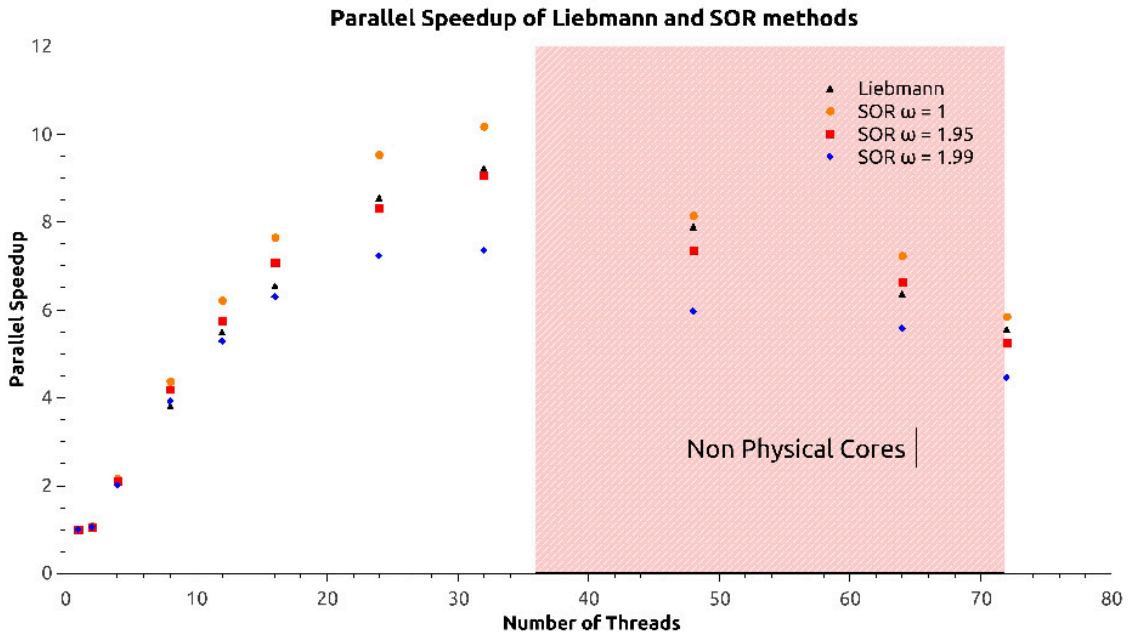
Επαναλάβαμε την παραπάνω ανάλυση αλλά αυτή την φορά με την μέθοδο SOR. Στον Πίνακα 1.3 έχουμε τα αποτελέσματα από την μέθοδο SOR για διάφορες τιμές της μεταβλητής ω.
Στο Σχήμα 1.10 παρατηρούμε ότι ο χρόνος εκτέλεσης της μεθόδου SORί είναι γενικά μικρότερος από την μέθοδο Liebmann. Στο Σχήμα 1.11 παρατηρούμε ότι τα βήματα που απαιτούνται από την μέθοδο SOR για την επίλυση αυτού του προβλήματος είναι σημαντικά μικρότερα από αυτά της μεθόδου Liebmann. Για παράδειγμα ακόμα και με $\omega = 1$ η SOR χρειάζεται περίπου τα μισά βήματα από την Liebmann και αυτό είναι λογικό αφού στην SOR στο ίδιο βήμα ανανεώνουμε τις μισές τιμές του πλέγματος για μια χρονική στιγμή και τις άλλες μισές για την επόμενη ουσιαστικά εκτελώντας δύο χρονικά βήματα με το μισό πλέγμα. Αλλά η πραγματική δύναμη της μεθόδου SOR εμφανίζεται όταν το $omega > 1$ όπου η μέθοδος επιταχύνει σημαντικά και χρειάζεται τάξεις μεγέθους λιγότερα βήματα για την επίλυση του ίδιου προβλήματος.
Στα σχήματα 1.12 και 1.13 παρατηρούμε συγκριτικά την Παράλληλη επιτάχυνση και απόδοση των μεθόδων. Όπως αναφέραμε και προηγουμένως για αυτόν τον επεξεργαστή το μέγιστο παρατηρείτε κοντά στο 36 που είναι και ο αριθμός των πραγματικών πυρήνων του. Παρατηρούμε ότι ενώ ο κώδικας εν γένει τρέχει πιο γρήγορα όσο αυξάνουμε τα threads μέχρι το 36 η απόδοση δεν είναι σημαντικά διαφορετική από ένα σημείο και μετά και ίσως να μην αξίζει η χρήση όλων των δυνατών πυρήνων.
Στο Σχήμα 1.13 παρατηρούμε την σύγκλιση της tolerance σε σχέση με τα βήματα της μεθόδου. Για $\omega = 1$ η σύγκλιση είναι παρόμοια της Liebmann αλλά ελαφρώς μεγαλύτερη λόγω του μικρού παράγοντα Overrelaxation. Παρατηρούμε ότι όσο αυξάνει ο παράγοντας ω και άρα το Overrelaxation η μέθοδος SOR συγκλίνει με γρηγορότερο ρυθμό σε σχέση με την μέθοδο Liebmann.



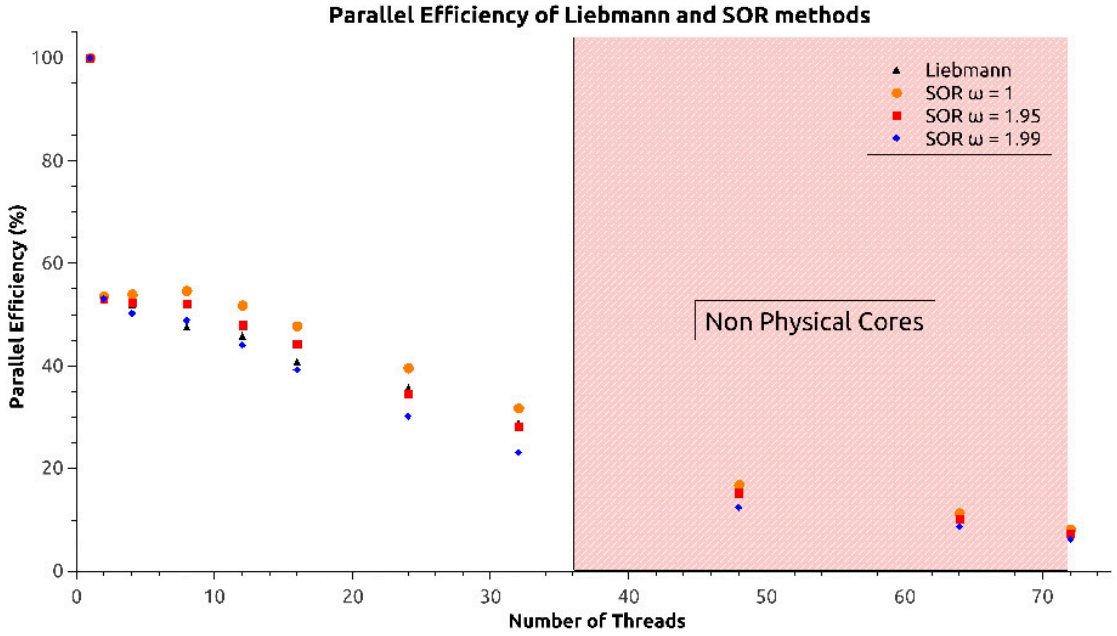
Σχήμα 1.10: Χρόνος εκτέλεσης των μεθόδων.



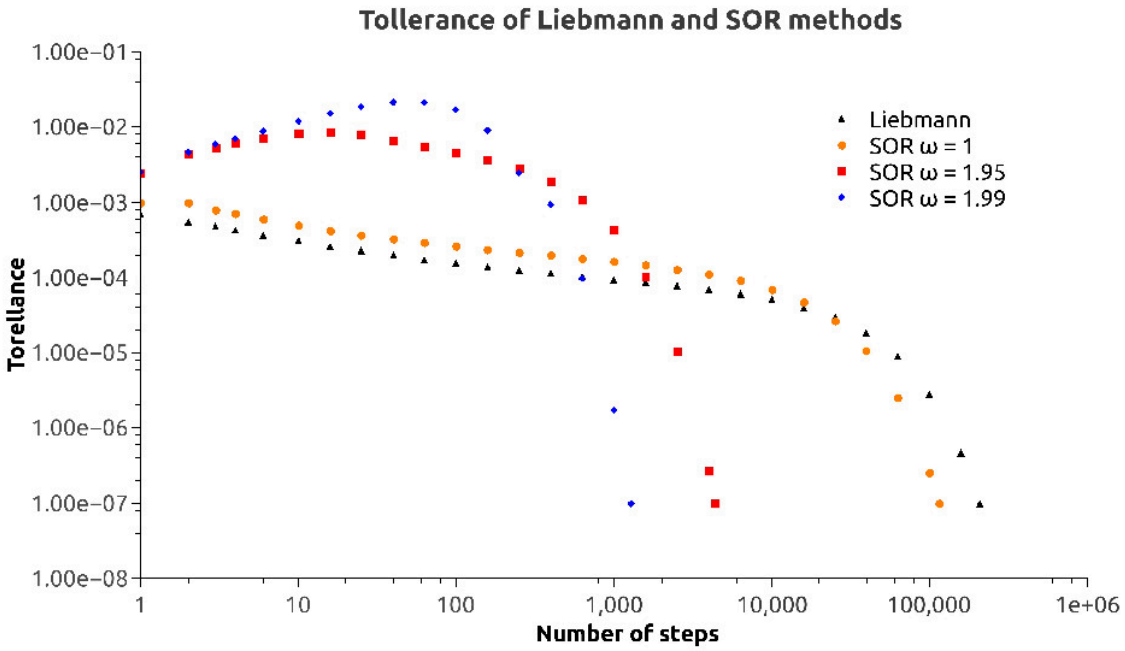
Σχήμα 1.11: Συνολικές επαναλήψεις των μεθόδων.



Σχήμα 1.12: Επιτάχυνση παράλληλης επεξεργασίας των μεθόδων.



Σχήμα 1.13: Απόδοση παράλληλης επεξεργασίας των μεθόδων.



Σχήμα 1.14: Μεταβολή της tolerance με τον αριθμό βημάτων των μεθόδων.

Time(s)	Threads	Tolerance	Itterations	Central Value
$\omega = \mathbf{1}$				
48.3543	1	9.99938e-08	115187	4.35574
45.0474	2	9.99938e-08	115187	4.35574
22.4003	4	9.99938e-08	115187	4.35574
11.0445	8	9.99938e-08	115187	4.35574
7.76593	12	9.99938e-08	115187	4.35574
6.31984	16	9.99938e-08	115187	4.35574
5.07433	24	9.99938e-08	115187	4.35574
4.75437	32	9.99938e-08	115187	4.35574
5.9415	48	9.99938e-08	115187	4.35574
6.6835	64	9.99938e-08	115187	4.35574
8.26442	72	9.99938e-08	115187	4.35574
$\omega = \mathbf{1.95}$				
1.8593	1	9.98276e-08	4382	4.3596
1.75654	2	9.98276e-08	4382	4.3596
0.886944	4	9.98276e-08	4382	4.3596
0.444763	8	9.98276e-08	4382	4.3596
0.322664	12	9.98276e-08	4382	4.3596
0.262419	16	9.98276e-08	4382	4.3596
0.223713	24	9.98276e-08	4382	4.3596
0.205245	32	9.98276e-08	4382	4.3596
0.252401	48	9.98276e-08	4382	4.3596
0.280584	64	9.98276e-08	4382	4.3596
0.354445	72	9.98276e-08	4382	4.3596
$\omega = \mathbf{1.99}$				
0.568381	1	9.81067e-08	1277	4.35971
0.534328	2	9.81067e-08	1277	4.35971
0.282344	4	9.81067e-08	1277	4.35971
0.145155	8	9.81067e-08	1277	4.35971
0.107497	12	9.81067e-08	1277	4.35971
0.0902891	16	9.81067e-08	1277	4.35971
0.0785352	24	9.81067e-08	1277	4.35971
0.0771618	32	9.81067e-08	1277	4.35971
0.0952247	48	9.81067e-08	1277	4.35971
0.102064	64	9.81067e-08	1277	4.35971
0.127787	72	9.81067e-08	1277	4.35971

Πίνακας 1.3: Τιμές ανάλυσης με την μέθοδο SOR για τιμές του ω=1, 1.95 και 1.99