Προγραμματιστικά εργαλεία-Ασκήσεις OpenMP

Δημήτριος Λούπας

Φεβρουάριος 2021

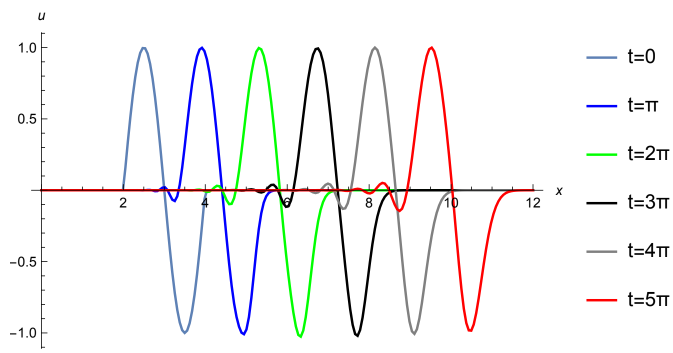
1 Άσκηση 1

Να λυθεί η γραμμική κυματική εξίσωση $u_{tt} - \alpha^2 u_{xx} = 0$ με $\alpha^2 = 2/\pi^2$ στο διάστημα $0 \leq x \leq 12$ και με αρχικές συνθήκες $u(x, 0) = \sin(\pi x)$ στο διάστημα $2 \leq x \leq 4$ και $u(x, 0) = 0$ στα διαστήματα $0 \leq x < 2$ και $4 < x \leq 12$ και με συνοριακές συνθήκες $u(0, t) = 0$ και $u(12, t) = 0$. Χρησιμοποιείστε την μέθοδο Lax-Wendroff

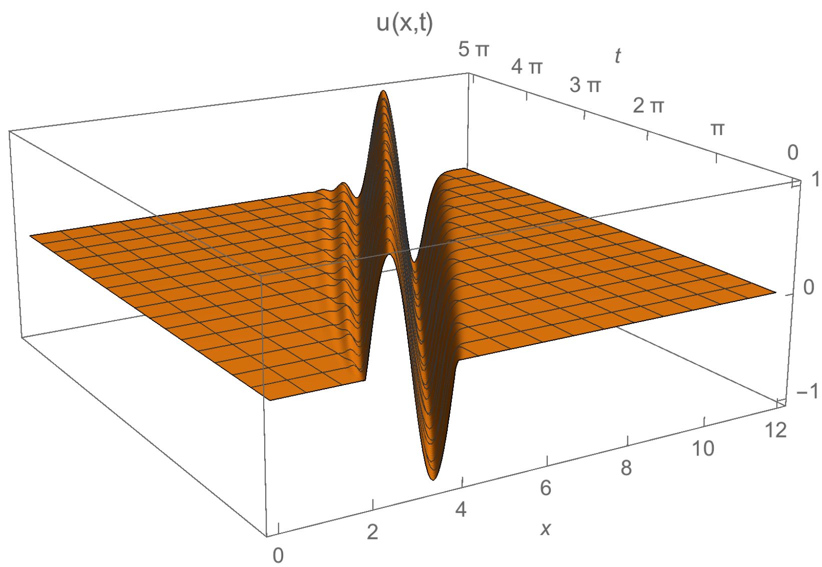
$$u_i^{n+1} = u_i^n - \frac{c}{2}(u_{i+1}^n - u_{i-1}^n) + \frac{c^2}{2}(u_{i+1}^n - 2u_i^n + u_{i-1}^n)$$

όπου $c = \alpha\Delta t/\Delta x$. Ως χρονικό βήμα χρησιμοποιείστε το $\Delta t = 0.5\Delta x/\alpha$ και βρείτε την λύση $u(x, t)$ για $0 \leq t \leq 5\pi$.

Παρακάτω παρουσιάζεται η λύση $u(x, t)$ για $t = 0, \pi, 2\pi, 3\pi, 4\pi, 5\pi$ καθώς και η επιφάνεια $u(x, t)$ για $0 < x < 12$ και $0 \leq t \leq 5\pi$ για $N = 200$ πλεγματικά σημεία.



Σχήμα 1: Λύση $u(x, t)$ για $t = 0, \pi, 2\pi, 3\pi, 4\pi, 5\pi$.



Σχήμα 2: Επιφάνεια $u(x, t)$ για $0 < x < 12$ και $0 \leq t \leq 5\pi$.

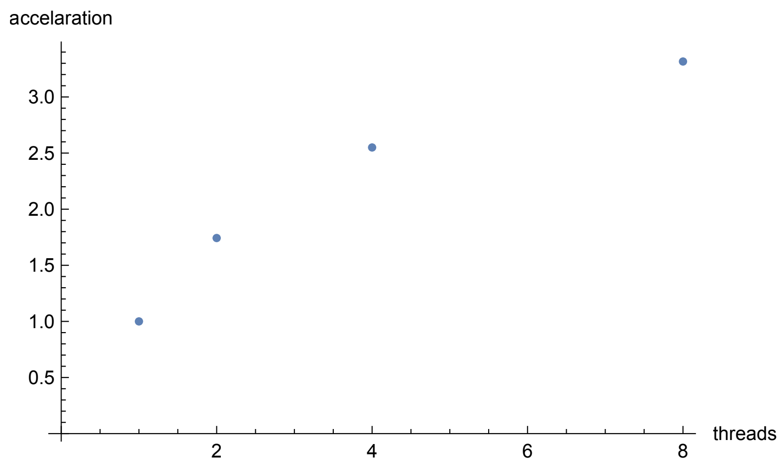
Για $N = 2000$ σημεία συγκρίνουμε τους χρόνους εκτέλεσης για 1, 2, 4, 8 threads και υπολογίζουμε την επιτάχυνση, η οποία ορίζεται ώς

$$a = \frac{\text{time in 1 core}}{\text{time in many cores}}$$

Οι χρόνοι για κάθε που προέκυψαν φαίνονται στον παρακάτω πίνακα

threads	χρόνος εκτέλεσης (s)	επιτάχυνση (a)
1	0.171	1
2	0.098	1.743
4	0.067	2.55
8	0.051	3.316

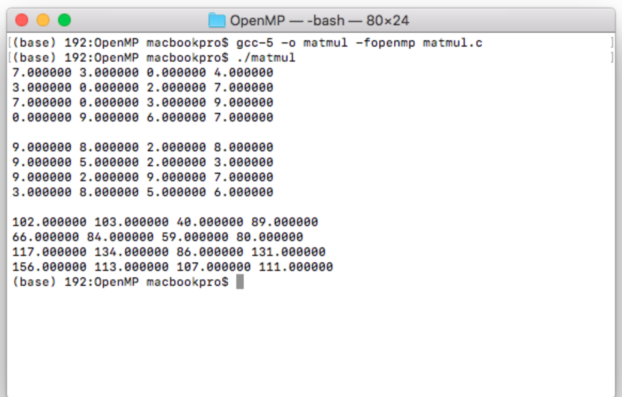
Το αντίστοιχο διάγραμμα φαίνεται παρακάτω,



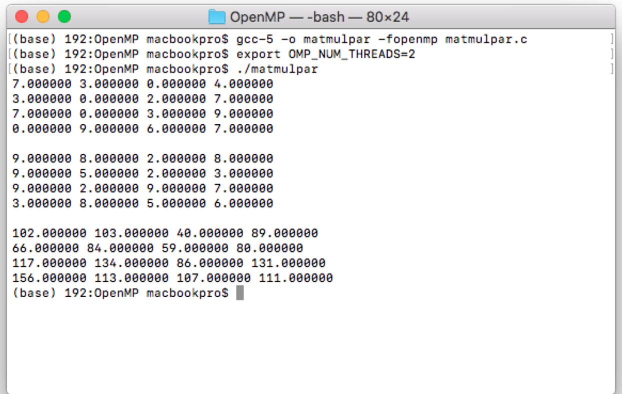
Σχήμα 3: Διάγραμμα επιτάχυνσης προς αριθμό πυρήνων.

2 Άσκηση 2

Μετατρέψτε το πρόγραμμα πολλαπλασιασμού πίνακα επί πίνακα matmul.c σε OpenMP. Στις παρακάτω εικόνες φαίνεται ότι τα αποτελέσματα του matmul.c και του αρχείου matmulpar.c συμφωνούν. Οι πίνακες δημιουργήθηκαν με τυχαίους ακέραιους αριθμούς απο $0 - 10$ με srand(1). Επίσης το αποτέλεσμα ελέγχθηκε αρκετές φορές ώστε να επαληθευτεί ότι οι δύο αλγόριθμοι δίνουν τα ίδια αποτελέσματα.



Σχήμα 4: Αποτέλεσμα για πινακα 4×4 του αρχείου matmul.c.



Σχήμα 5: Αποτέλεσμα για πινακα 4×4 του αρχείου matmul.c μετα απο παραλληλοποίηση σε 2 πυρήνες.

Στη συνέχεια επιλέξαμε να πραγματοποιήσουμε πολλαπλασιασμό πινάκων διαστάσεων 2000×2000 και συγκρίναμε τους χρόνους εκτέλεσης για 1, 2, 4, 8 threads. Επίσης υπολογίζουμε την επιτάχυνση a, η οποία ορίζεται ως

$$a = \frac{\text{time in 1 core}}{\text{time in many cores}}$$

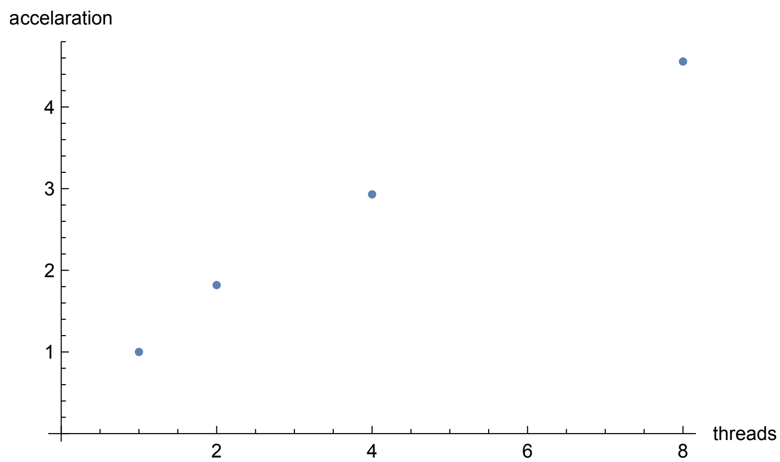
καθώς και το βαθμό παράλληλης απόδοσης που ορίζεται ως

$$\eta = \frac{a}{\text{number of cores}} 100\%$$

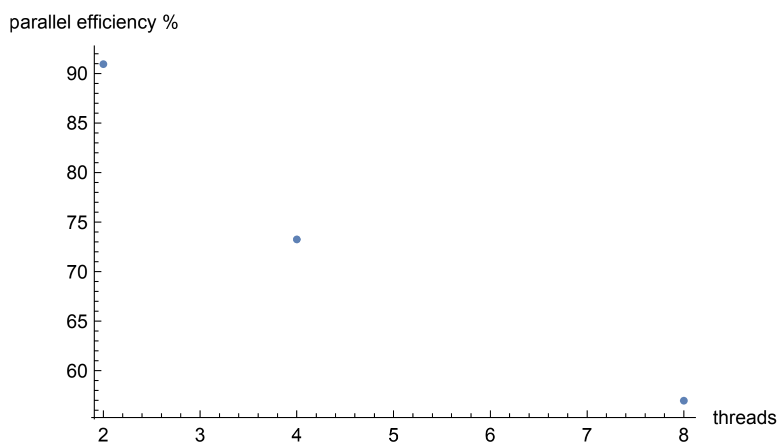
Οι χρόνοι που προέκυψαν για κάθε αριθμό threads, καθώς και η επιτάχυνση μαζί με την παράλληλη απόδοση φαίνονται στον παρακάτω πίνακα

threads	χρόνος εκτέλεσης (s)	επιτάχυνση (a)	παράλληλη απόδοση (η)
1	53.098	1	
2	29.19	1.819	90.95%
4	18.118	2.93	73.25%
8	11.65	4.557	56.96%

Τα αντίστοιχα διαγράμματα φαίνονται παρακάτω



Σχήμα 6: Διάγραμμα επιτάχυνσης προς αριθμό πυρήνων.



Σχήμα 7: Διάγραμμα παράλληλης απόδοσης προς αριθμό πυρήνων.

3 Άσκηση 3

Για την άσκηση αυτή θα γραφεί σε OpenMP ο αλγόριθμος του κανόνα του Simpson h/3 ο οποίος χρησιμοποιείται για αριθμητική ολοκλήρωση. Πιο συγκεκριμένα θα παρουσιάσουμε τον πολλαπλό κανόνα του Simpson h/3. Ο κανόνας του Simpson h/3 προκύπτει αν στο ολοκλήρωμα, που θελουμε να λύσουμε, αντικαταστήσουμε την συνάρτηση $f(x)$ με ένα πολυώνυμο $f_2(x)$ δευτέρου βαθμού,

$$I = \int_a^b f(x)dx \approx \int_a^b f_2(x)dx$$

αντικαθιστώντας την $f(x)$ απο ένα πολυώνυμο παρεμβολής με την μέθοδο του Lagrange, προκύπτει ο γνωστός κανόνας του Simpson h/3

$$I \approx \frac{h}{3}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right)$$

όπου $h = \frac{b-a}{2}$.

Ο πολλαπλός κανόνας του Simpson h/3 προκύπτει απο την διαμέριση του διαστήματος $[a, b]$ σε n ίσα μέρη, $h = \frac{b-a}{n}$, με αποτέλεσμα να προκύπτει για το ολοκλήρωμα

$$I = \int_{x_0}^{x_2} f(x)dx + \int_{x_2}^{x_4} f(x)dx + \cdots + \int_{x_{n-2}}^{x_n} f(x)dx$$

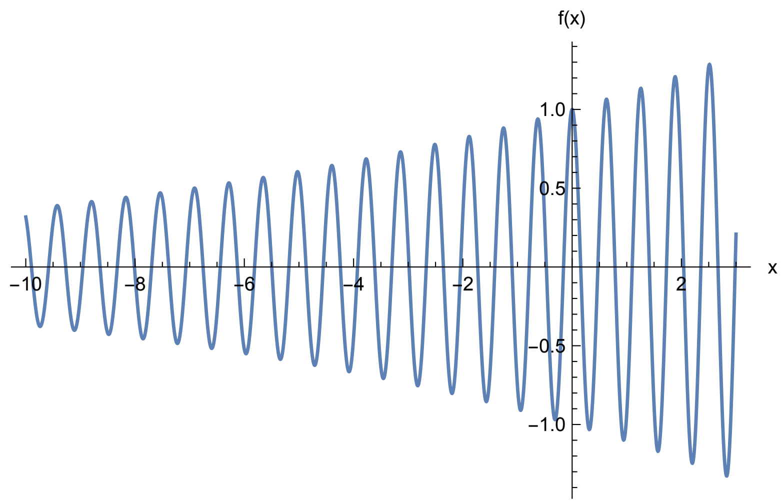
με $a = x_0$ και $b = x_n$. Αντικαθιστώντας το κάθε ολοκλήρωμα με τον κανόνα του Simpson h/3 προκύπτει

$$I \approx \frac{h}{3}\left(f(x_0) + 4\sum_{i=1,3,5}^{n-1} f(x_i) + 2\sum_{j=2,4,6}^{n-2} f(x_j) + f(x_n)\right)$$

Η συνάρτηση για την οποία εφαρμόσαμε τον κανόνα του Simpson h/3 είναι η

$$f(x) = e^{x/10}\cos(kx)$$

η οποία, για $k = 10$, στο διάστημα $[-10, 3]$ έχει την μορφή

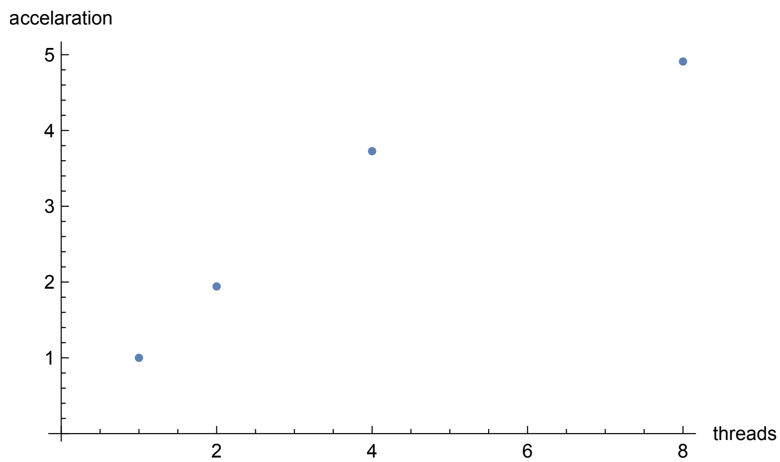


Σχήμα 8: Συνάρτηση $f(x)$.

Ολοκληρώσαμε την παραπάνω συνάρτηση για $k = 10000$ με τον κανόνα του Simpson h/3 για $n = 50000000$. Το αποτέλεσμα βρέθηκε να είναι 9.8197169723815, με σφάλμα $3.0020431 \cdot 10^{-13}$. Επίσης παρατηρήθηκε ότι το αποτέλεσμα διαφέρει στο 13^o δεκαδικό όταν το τρέχουμε σε πιο πολλούς πυρήνες. Ψάχνοντας για το λόγο που συμβαίνει αυτό, βρήκα ότι οφείλεται στο ότι η πρόσθεση των floating point αριθμών δεν έχει την προσεταιριστική ιδιότητα. Οι χρόνοι που προέκυψαν για κάθε αριθμό threads καθώς και η επιτάχυνση φαίνονται στον παρακάτω πίνακα

threads	χρόνος εκτέλεσης (s)	επιτάχυνση (a)
1	1.689	1
2	0.87	1.942
4	0.453	3.727
8	0.344	4.91

Το αντίστοιχο διάγραμμα φαίνεται παρακάτω,



Σχήμα 9: Διάγραμμα επιτάχυνσης προς αριθμό πυρήνων.