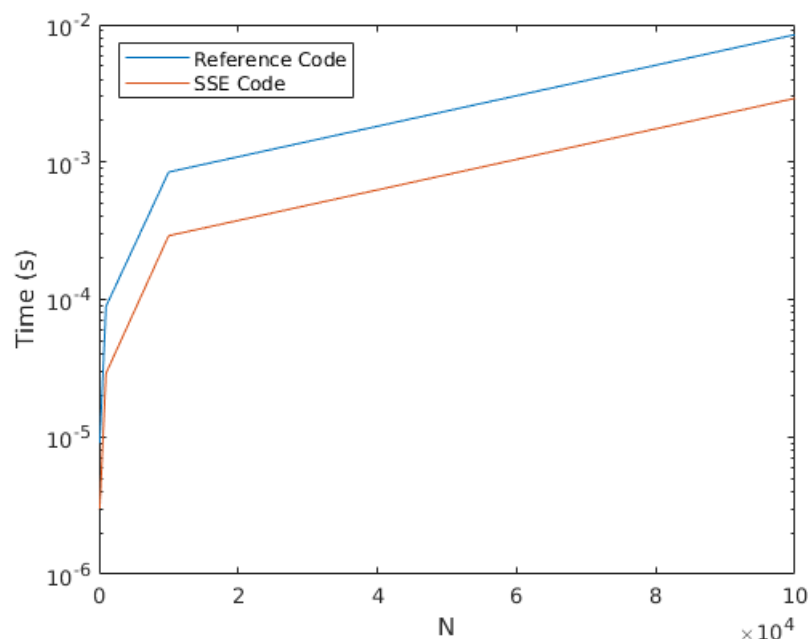


Στην 2η άσκηση μελετήσαμε τις εντολές SSE, τη βιβλιοθήκη MPI καθώς επίσης και τον συνδυασμό τους για ακόμα καλύτερα αποτελέσματα όπως θα δούμε στη συνέχεια. Αρχικά, αντιγράψαμε τον κώδικα που δινόταν στην εκφώνηση και τον εκτελέσαμε για να επαληθεύσουμε ότι δουλεύει. Αφότου σιγουρευτήκαμε ότι όλα ήταν εντάξει προχωρήσαμε στην τροποποίηση του κώδικα με τη χρήση εντολών SSE και ακόμη μετέπειτα με την προσθήκη των εντολών της βιβλιοθήκης MPI. Αναλυτικότερα περιγράφουμε τους κώδικες και τις αλλαγές που κάναμε παρακάτω.

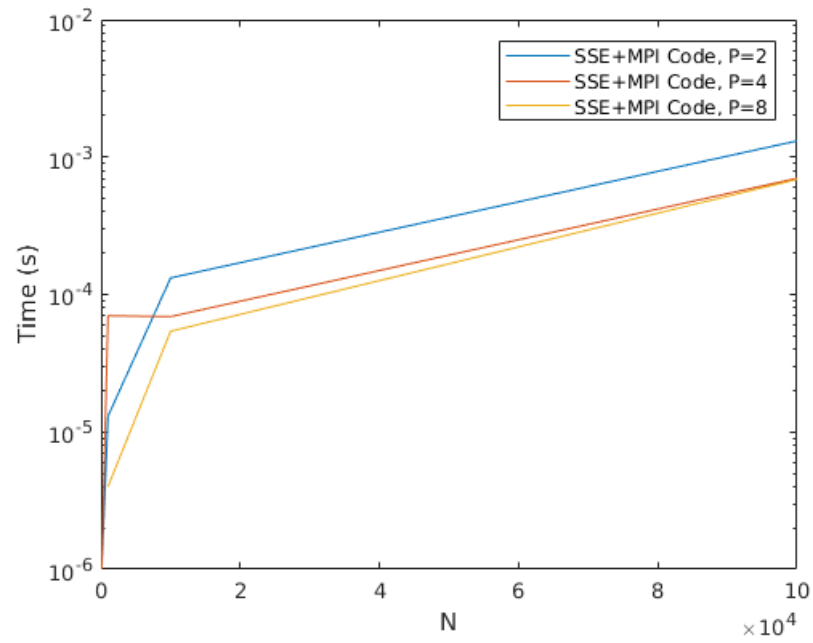
1 SSE

Στον κώδικα με τις εντολές SSE χρησιμοποιήσαμε όπως μας υποδείχθηκε μόνο εντολές τύπου `_m128` καθώς όλα τα διανύσματα που χρησιμοποιούσαμε στις πράξεις που χρειαζόταν να κάνουμε ήταν τύπου `float`. Πιο συγκεκριμένα, χρησιμοποιήσαμε τις εντολές/συναρτήσεις, `add`, `sub`, `div`, `mul`, `max`, `load`, `store`, `set` και `setzero` για single-precision (32-bit) floating-point εισόδους. Δε χρειαστήκαμε παραπάνω εντολές καθώς δε θέλαμε να κάνουμε κάτι πιο περίπλοκο όπως επίσης και δεν κάναμε καμία άλλη αλλαγή στον κώδικα. Εφόσον οι εντολές είναι 128-bit και το κάθε στοιχείο έχει ακρίβεια 32-bit αντιληφθήκαμε μετά από δοκιμές που κάναμε και για να το επαληθεύσουμε, ότι οι εντολές αυτές 'έπαιρναν' κάθε φορά 4 στοιχεία από το εκάστοτε διάνυσμα και εκτελούσαν την πράξη που έλεγε το όνομα τους. Έτσι, προσαρμόσαμε την `for-loop` ώστε να προχωράει μέχρι το N με βήμα 4 στοιχεία κάθε φορά. Παρακάτω παρουσιάζουμε ένα διάγραμμα σε λογαριθμική κλίμακα για να είναι πιο αισθητή η διαφορά στους χρόνους εκτέλεσης ανάμεσα στον κώδικα που μας είχε δοθεί ως `reference` και στην τροποποίηση που του κάναμε με τις εντολές SSE.



Σχήμα 1: Σύγκριση χρόνων εκτέλεσης του `reference` κώδικα και του κώδικα με εντολές SSE.

2 SSE & MPI



Σχήμα 2: Σύγκριση χρόνων εκτέλεσης του κώδικα με SSE+MPI εντολές για διαφορετικές τιμές της παραμέτρου P.