ASC – Tema 2

Trașcă Andrei-Cătalin – 335CA

1. Varianta **BLAS**

Pentru implementarea folosind BLAS am folosit funcția cblas_dtrmm. Aceasta funcție înmulțește 2 matrici, A și B, A putând fi triunghiulară. Rezultatul final se pune în B. Deoarece matricile inițiale sunt suprascrise, am folosit 2 matrici auxiliare astfel:

- A2 – copiez inițial în el matricea A, iar apoi înmulțesc A cu A2, la final A2 conținând matricea A^2;
- Următoare înmulțire este A^2 x B, rezultatul final fiind pus în B;
- La început am copiat în matricea B2 matricea B, pentru a o putea înmulți la final cu A^t;
- La final mai trebuie sa adun cele 2 matrici rezultate, B și B2, rezultatul final fiind pus în B2.

2. Varianta **neopt**

Pentru implementarea variantei neoptime am creat o funcție – multiplyMatrix, care primește ca parametrii 3 matrici de dimensiune N x N, și un întreg care specifică tipul primei matrice, A – inferior, superior sau non triunghiulară, și face operația C = A x B.

3. Varianta **opt_m**

Plecând de la funcția variantei neopt și de la explicațiile din laboratorul 5 ale algoritmului BMM, am făcut funcția multiplyMatrix2. De asemenea, am înlocuit și accesul la memorie folosind vectori cu pointeri.

Am rulat 5 teste pe cele 2 variante, neopt și opt_m, iar rezultatele au fost următoarele:

```
[andrei.trasca1702@fep7-1 tema2-ASC]$ cat *.o* | grep 'N=1200'
Run=./tema2_neopt: N=1200: Time=46.200024
Run=./tema2_neopt: N=1200: Time=46.430740
Run=./tema2_neopt: N=1200: Time=46.157246
Run=./tema2_neopt: N=1200: Time=46.688515
Run=./tema2_neopt: N=1200: Time=42.474140
Run=./tema2_opt_m: N=1200: Time=12.895268
Run=./tema2_opt_m: N=1200: Time=12.893928
Run=./tema2_opt_m: N=1200: Time=12.885343
Run=./tema2_opt_m: N=1200: Time=12.893934
Run=./tema2_opt_m: N=1200: Time=12.875225
```

Se poate observa un timp mediu pentru neopt de 45.58 secundem iar pentru opt_m de 12.88 secunde. Astfel am obținut un speedup mediu de 32.7 secunde, adică peste 70%.

4. Varianta opt_f_extra

Flagurile folosite au fost următoarele:

```
-funroll-loops -ffast-math -mfpmath=sse -funsafe-math-optimizations
```

- -funroll-loops: acest flag desface loop-urile al căror număr de pași poate fi determinat la compilare. Singurul dezavantaj al său îl reprezintă faptul că sursa va fi la final mai mare, chiar daca îmbunătățește sau nu rezultatul. În cazul de față eu nu am loop-uri al căror număr de pași poate fi determinat înainte de citire, dar neștiind cum este implementat main-ul l-am inclus oricum, neavând nimic de pierdut.
- -mfpmath=sse: procesorul va folosi setul de instrucțiuni de tip sse – single instruction, multiple data, schimbând resursele utilizate de acesta. Este un set de instrucțiuni ce este folosit în aplicații ce fac intens operații cu matrice(cum ar fi GPU și cum este și cazul de față).
- -funsafe-math-optimizations: acest flag determină programul să presupună că întotdeauna argumentele și rezultatele sunt valide. Deși nu este recomandat pentru că poate duce la rezultate greșite, în cazul de față se poate garanta întotdeauna corectitudinea datelor, iar în urma testelor am observat că rezultatele sunt în continuare corecte.
- --ffast-math: asemănător cu funsafe-math-optimizations, acesta activează mai multe flaguri ce pot duce la un speedup al programului, dar nu mai garantează rezultate valide la final.

Am rulat 5 teste pe cele 2 variante, opt_f și opt_f_extra, iar rezultatele au fost următoarele:

```
[andrei.trasca1702@fep7-1 tema2-ASC]$ cat *.o* | grep 'N=1200'
Run=./tema2_opt_f_extra: N=1200: Time=5.499411
Run=./tema2_opt_f_extra: N=1200: Time=5.750024
Run=./tema2_opt_f_extra: N=1200: Time=5.975331
Run=./tema2_opt_f_extra: N=1200: Time=5.661085
Run=./tema2_opt_f_extra: N=1200: Time=5.785672
Run=./tema2_opt_f: N=1200: Time=5.830511
Run=./tema2_opt_f: N=1200: Time=6.140317
Run=./tema2_opt_f: N=1200: Time=6.278801
Run=./tema2_opt_f: N=1200: Time=6.349958
Run=./tema2_opt_f: N=1200: Time=6.114664
[andrei.trasca1702@fep7-1 tema2-ASC]$
```

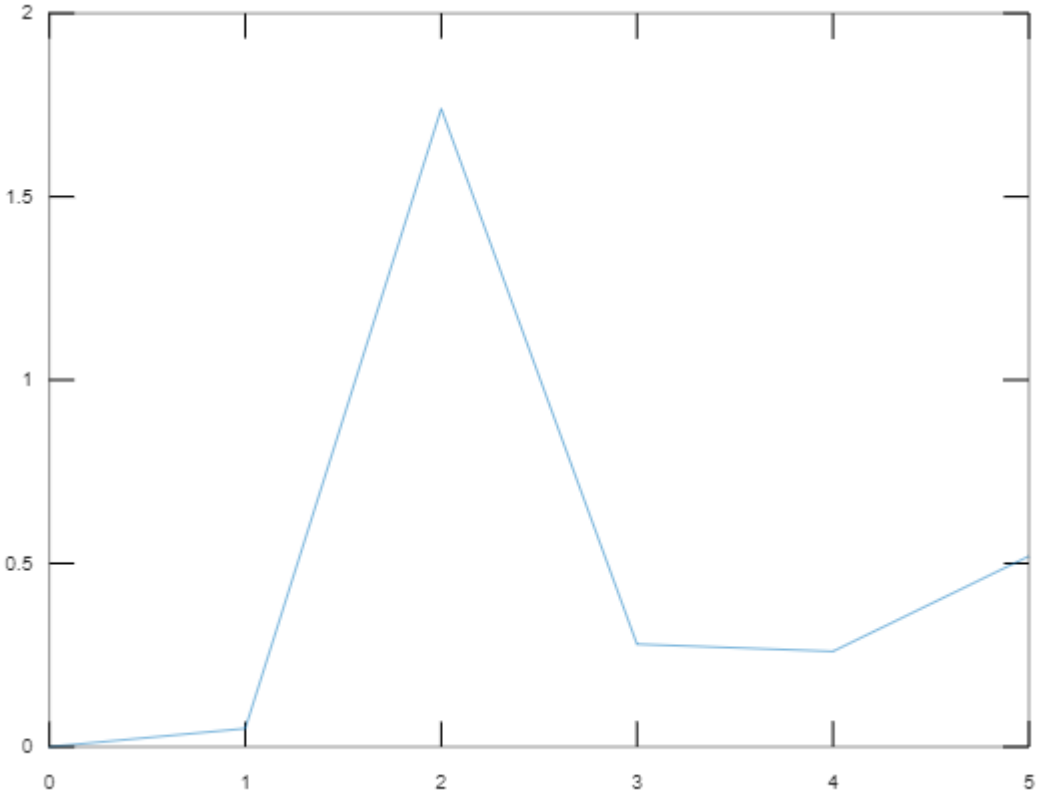
Se poate observa un timp mediu pentru opt_f de 6.14 secunde iar pentru opt_f_extra de 5.73 secunde. Astfel am obținut un speedup mediu de 0.41 secunde, adică peste 6%.

5. Analiza performanței

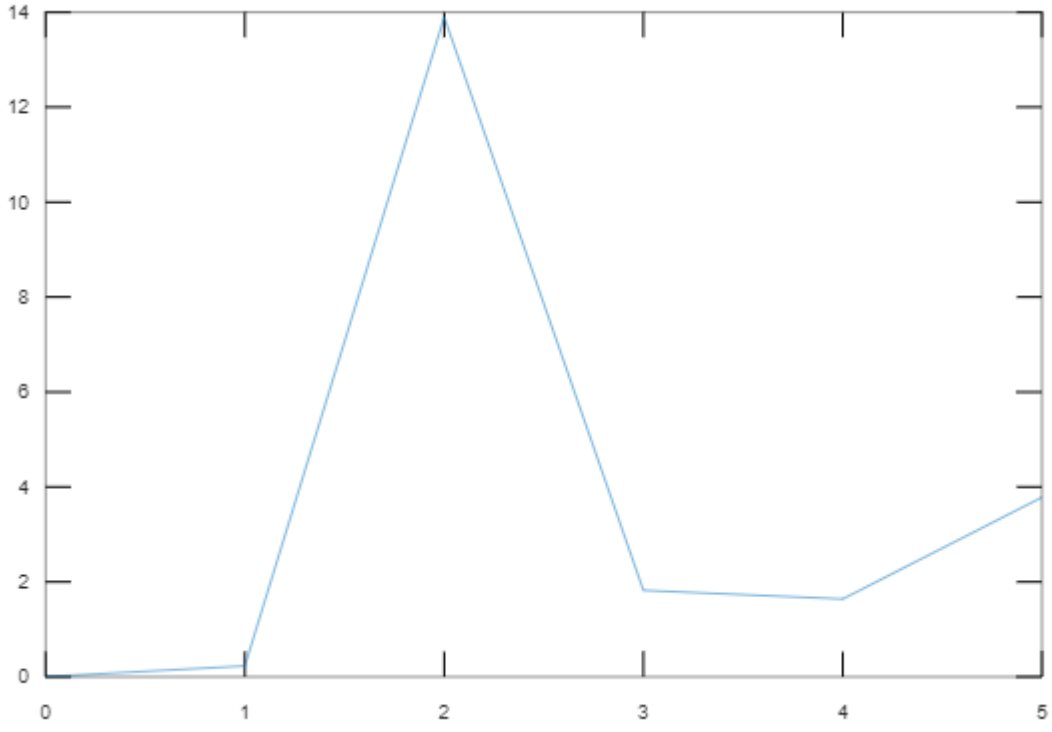
Cele 5 grafice care urmează a fi prezentate au următoarea legendă:

- Pe axa O_x se află cei 5 algoritmi, numerotați de la 1 la 5 astfel:
  - 1. BLAS
  - 2. Neopt
  - 3. Opt_f
  - 4. Opt_f_extra
  - 5. Opt_m
- Pe axa O_y se află timpii obținuți de fiecare algoritm
- Fiecare grafic reprezintă rezultatul pentru un anume N, N = [400, 800, 1000, 1200, 1600]

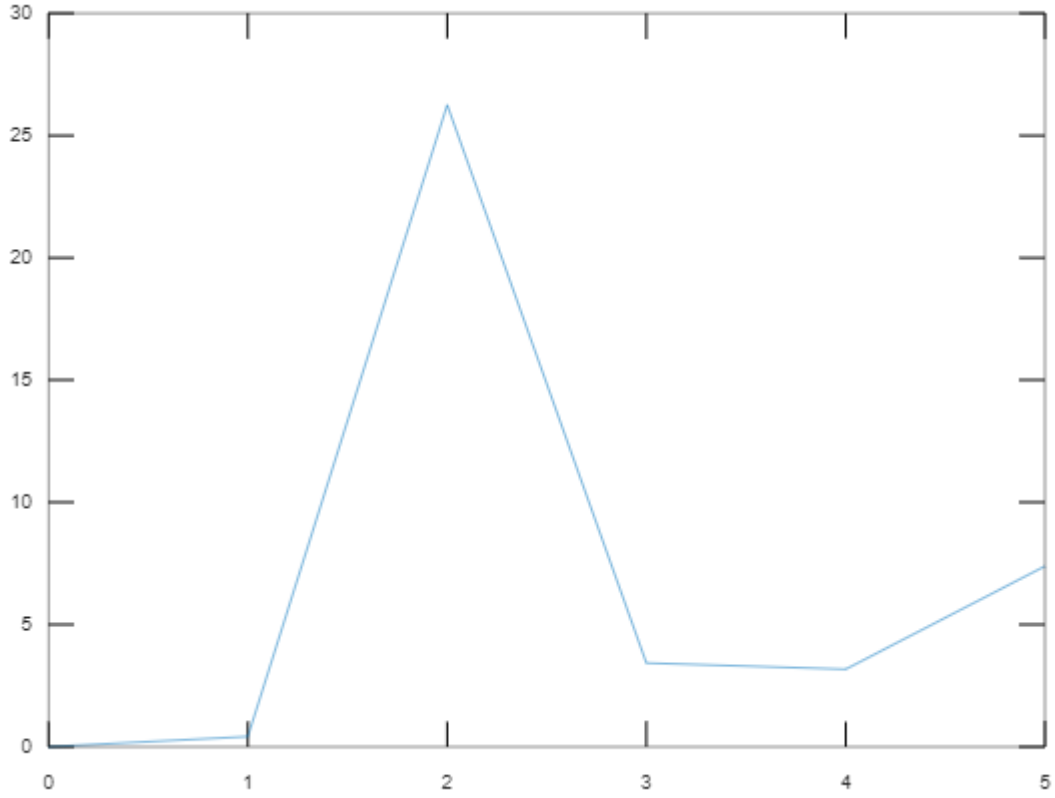
1. N = 400



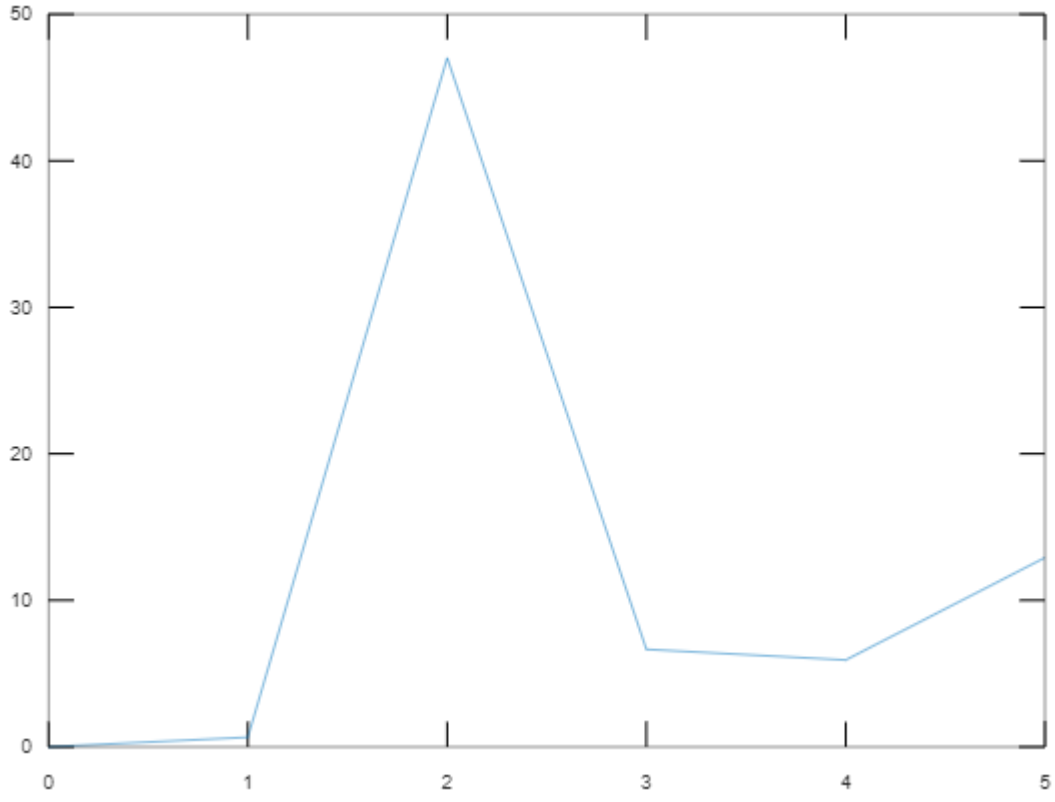
2. N = 800



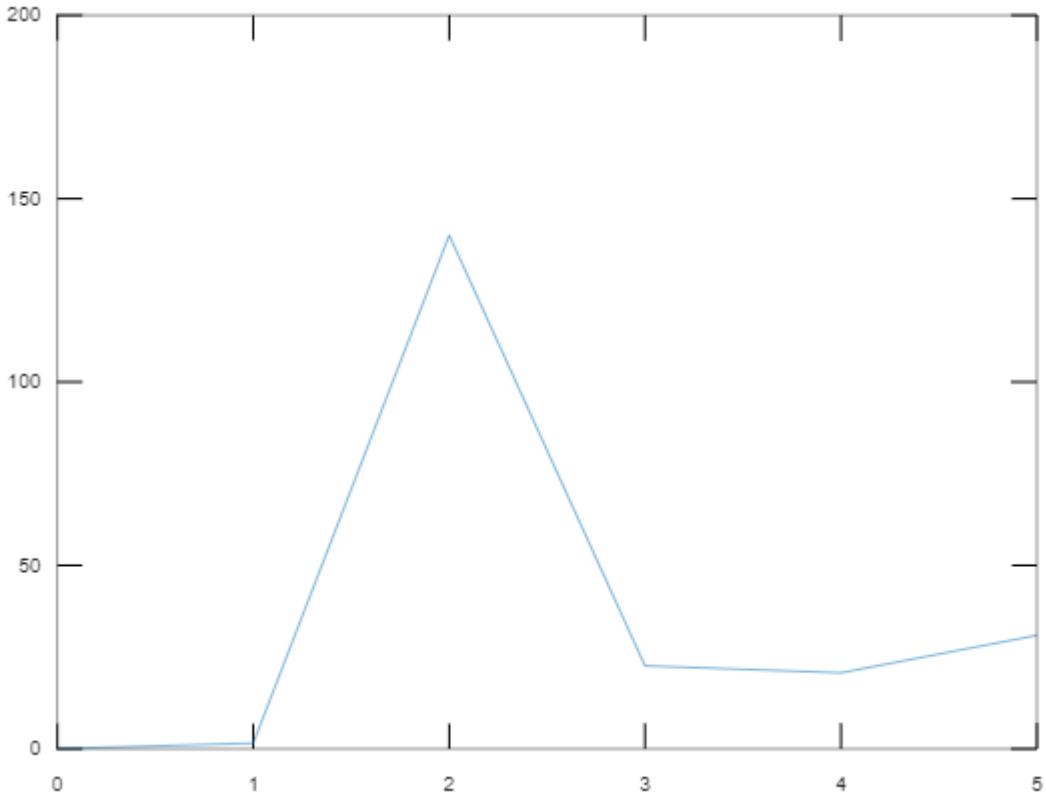
3. N = 1000



4. N = 1200



5. N = 1600



Se poate observa din cele 5 grafice că varianta BLAS este fără discuție cea mai optimă, având rezultate net superioare oricărei alte variante. Varianta neoptimă are cel mai rău timp, ajungând la un rezultat de 160 de ori mai mare decât varianta BLAS. Între variantele opt_f și opt_f_extra se poate observa în fiecare caz un speed-up micuț. Varianta optimizată are un timp mult mai bun fața de variantă neoptimizată(de 10 ori mai mic), însă în niciun punct nu poate să se apropie măcar putin de variantele compilate cu O3, dovedind astfel faptul că în acest moment compilatoarele au ajuns la un nivel foarte înalt de optimizare. Deși nici ele nu sunt apropiate ca timp de varianta BLAS, reușesc o variantă de 2 ori mai bună pe un cod neoptimizat, față de un cod optimizat ce este compilat cu flagul O3.

Datele pe care au fost făcute graficele sunt următoarele:

```
[andrei.trasca1702@fep7-1 tema2-ASC]$ cat *.o*
Run=./tema2_blas: N=400: Time=0.051757
Run=./tema2_blas: N=800: Time=0.233646
Run=./tema2_blas: N=1000: Time=0.422197
Run=./tema2_blas: N=1200: Time=0.656858
Run=./tema2_blas: N=1600: Time=1.528427
Run=./tema2_neopt: N=400: Time=1.746676
Run=./tema2_neopt: N=800: Time=13.906057
Run=./tema2_neopt: N=1000: Time=26.254597
Run=./tema2_neopt: N=1200: Time=47.028236
Run=./tema2_neopt: N=1600: Time=140.893723
Run=./tema2_opt_f_extra: N=400: Time=0.260502
Run=./tema2_opt_f_extra: N=800: Time=1.649732
Run=./tema2_opt_f_extra: N=1000: Time=3.177583
Run=./tema2_opt_f_extra: N=1200: Time=5.937847
Run=./tema2_opt_f_extra: N=1600: Time=20.899639
Run=./tema2_opt_f: N=400: Time=0.283189
Run=./tema2_opt_f: N=800: Time=1.821481
Run=./tema2_opt_f: N=1000: Time=3.437074
Run=./tema2_opt_f: N=1200: Time=6.654273
Run=./tema2_opt_f: N=1600: Time=22.113411
Run=./tema2_opt_m: N=400: Time=0.521916
Run=./tema2_opt_m: N=800: Time=3.788756
Run=./tema2_opt_m: N=1000: Time=7.397385
Run=./tema2_opt_m: N=1200: Time=12.921011
Run=./tema2_opt_m: N=1600: Time=30.764929
<<< Bonus=10p >>>
```