

Optymalizacja kodu na różne architektury

Zadanie 1

Oscar Teeninga

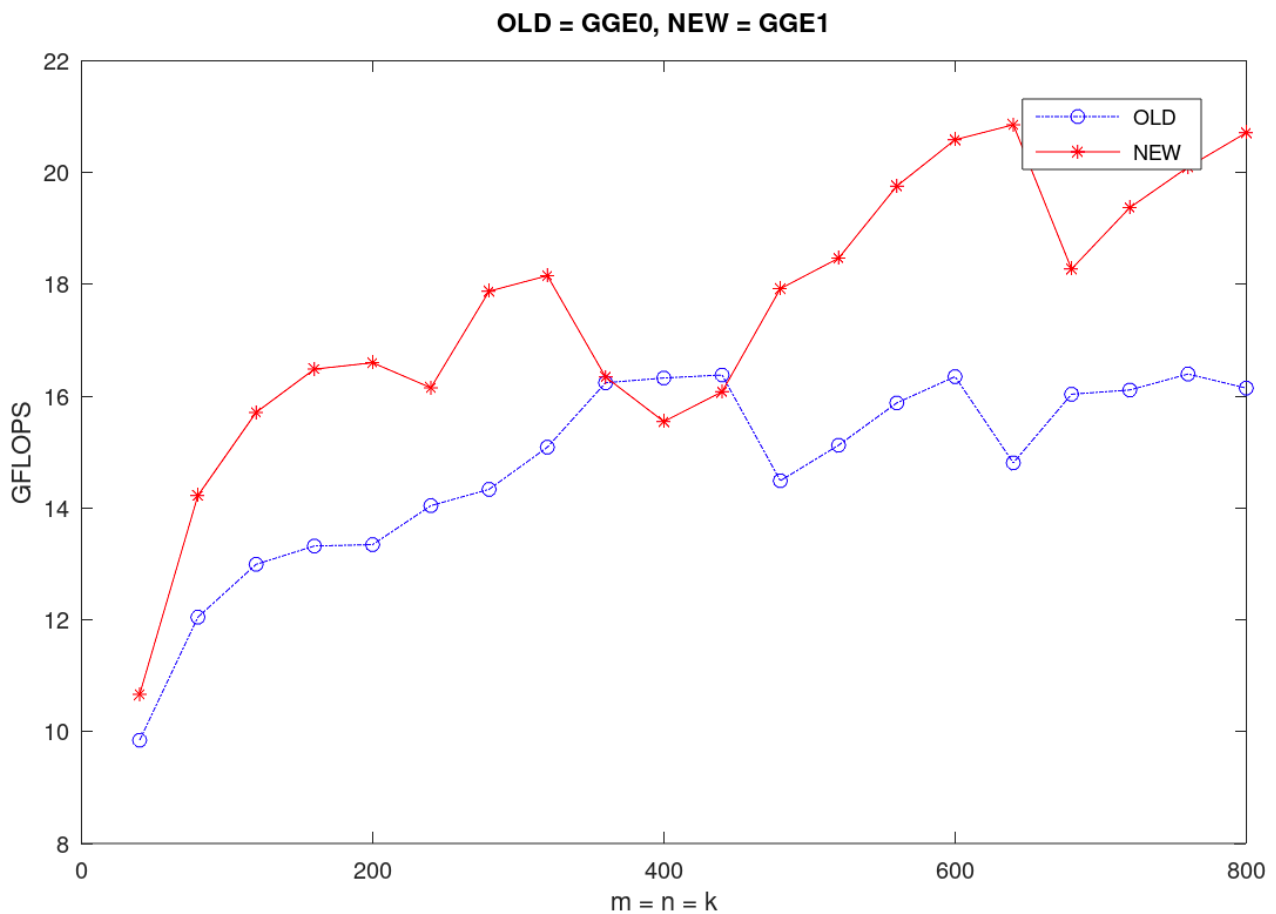
1. Platforma testowa

Testy były wykonywane pod systemem MacOS X Catalina. Parametry procesora:

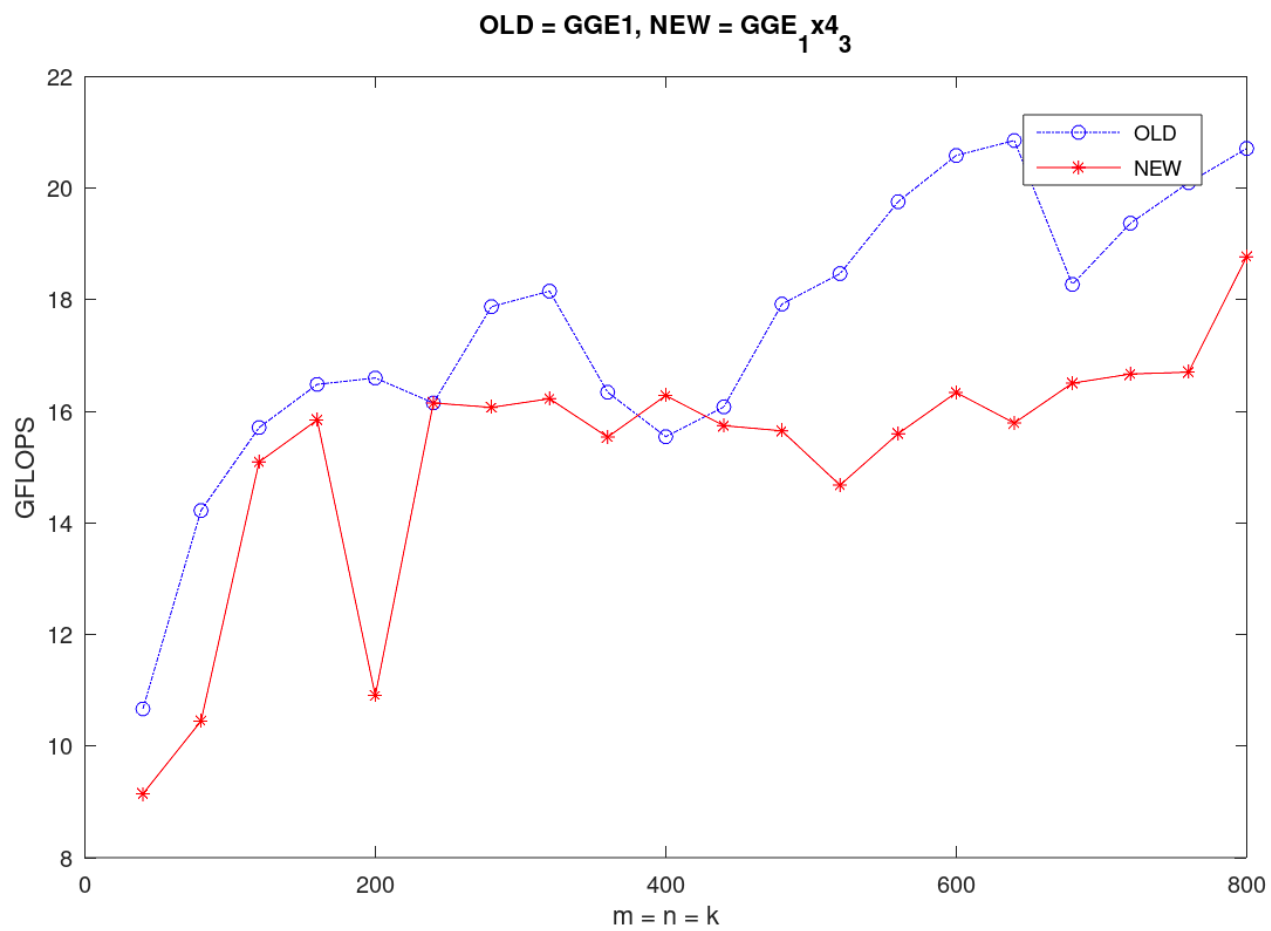
| | |
|----------------------|----------|
| Producent | Intel |
| Model | i9 9880H |
| Rdzenie | 8 |
| Wątki | 16 |
| Częstotliwość bazowa | 2,30 GHz |
| Turbo | 4,80 GHz |
| Cache L3 | 16 MB |

Autor ćwiczenia posiadał procesor o tej samej architekturze co ja oraz ten sam system operacyjny, więc zmieniłem jedynie taktowanie procesora w pliku *proc_parameters.m* na 4,80 GHz.

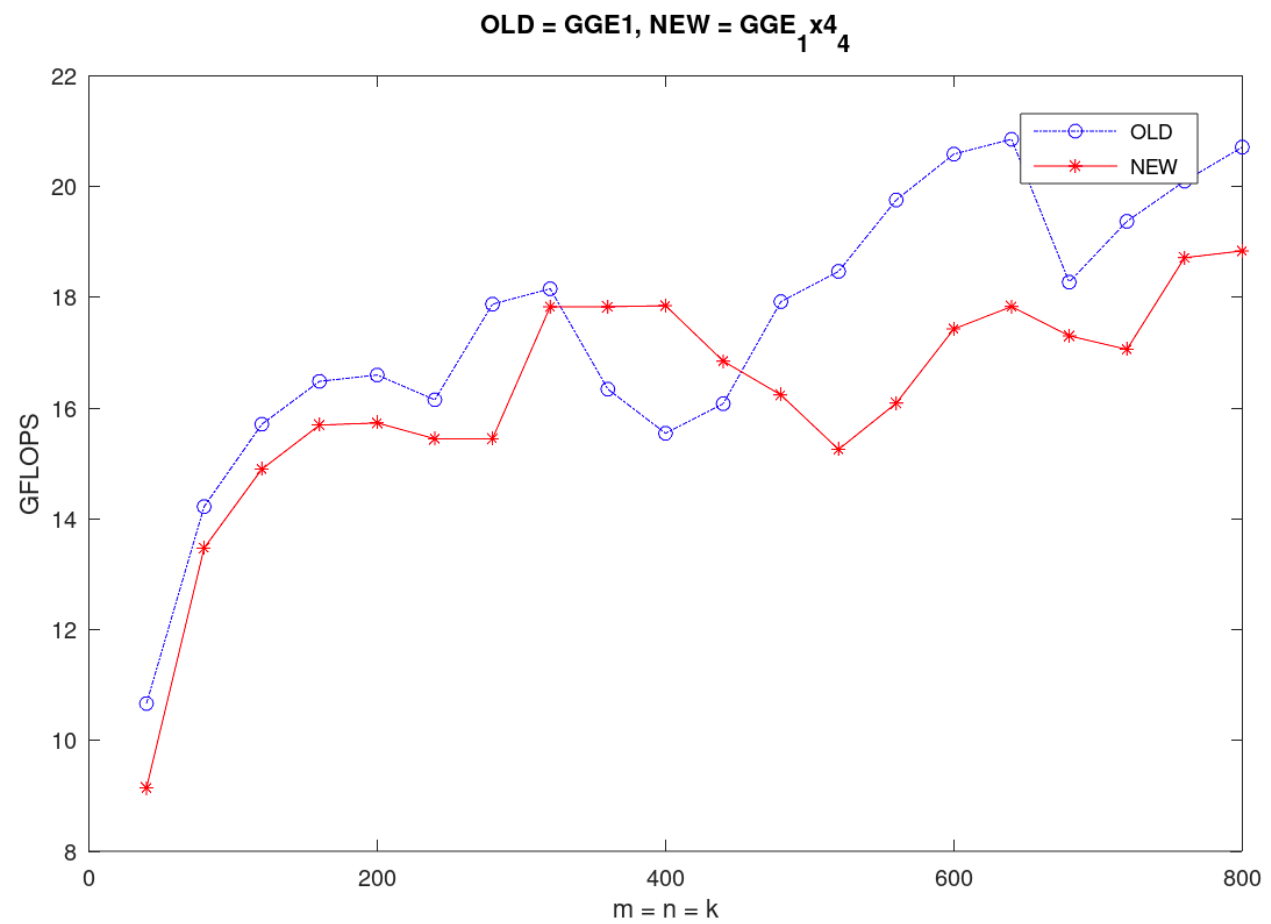
2. Optymalizacja GGE1



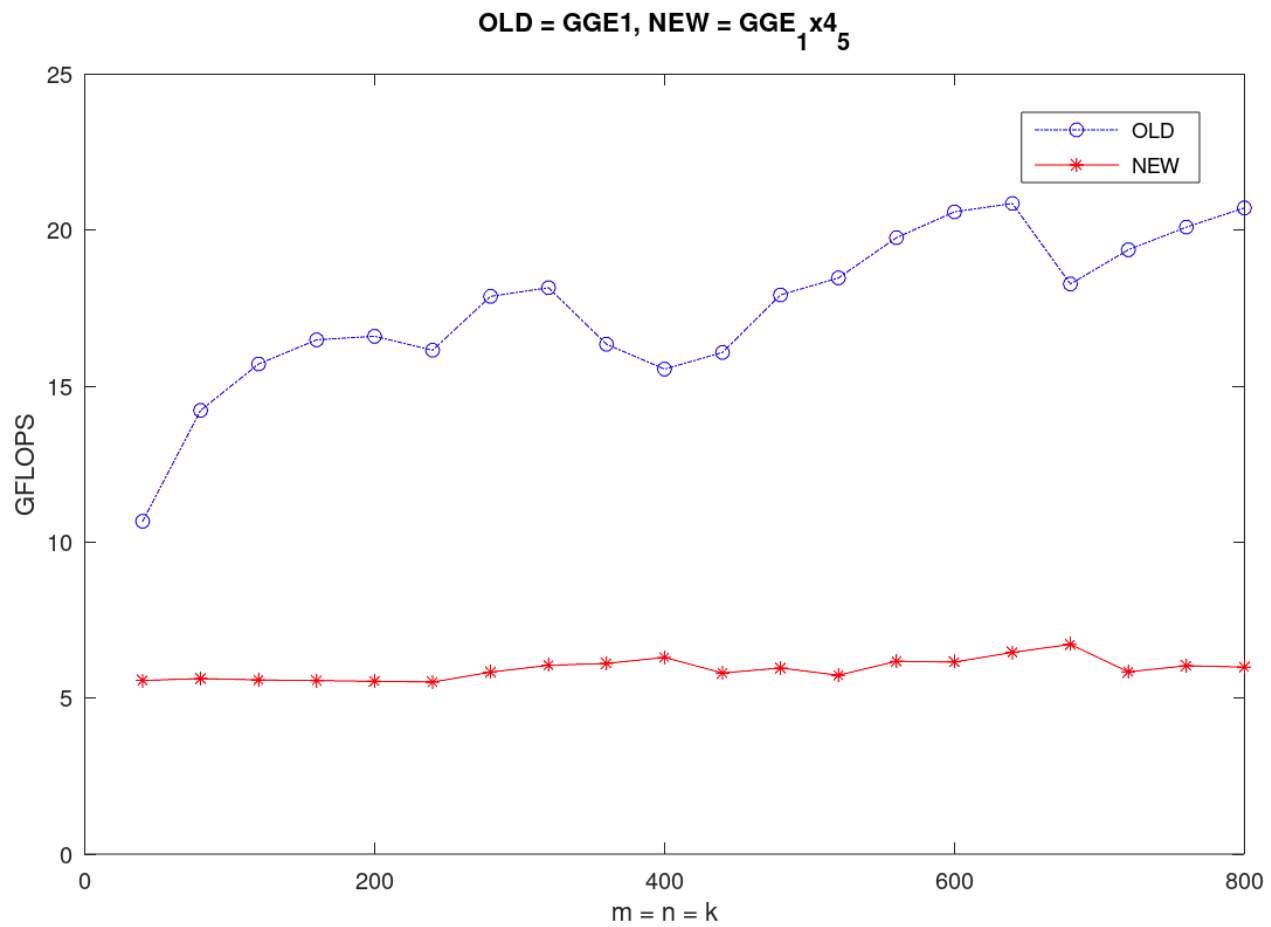
2. Optymalizacja GGE_1x4_3



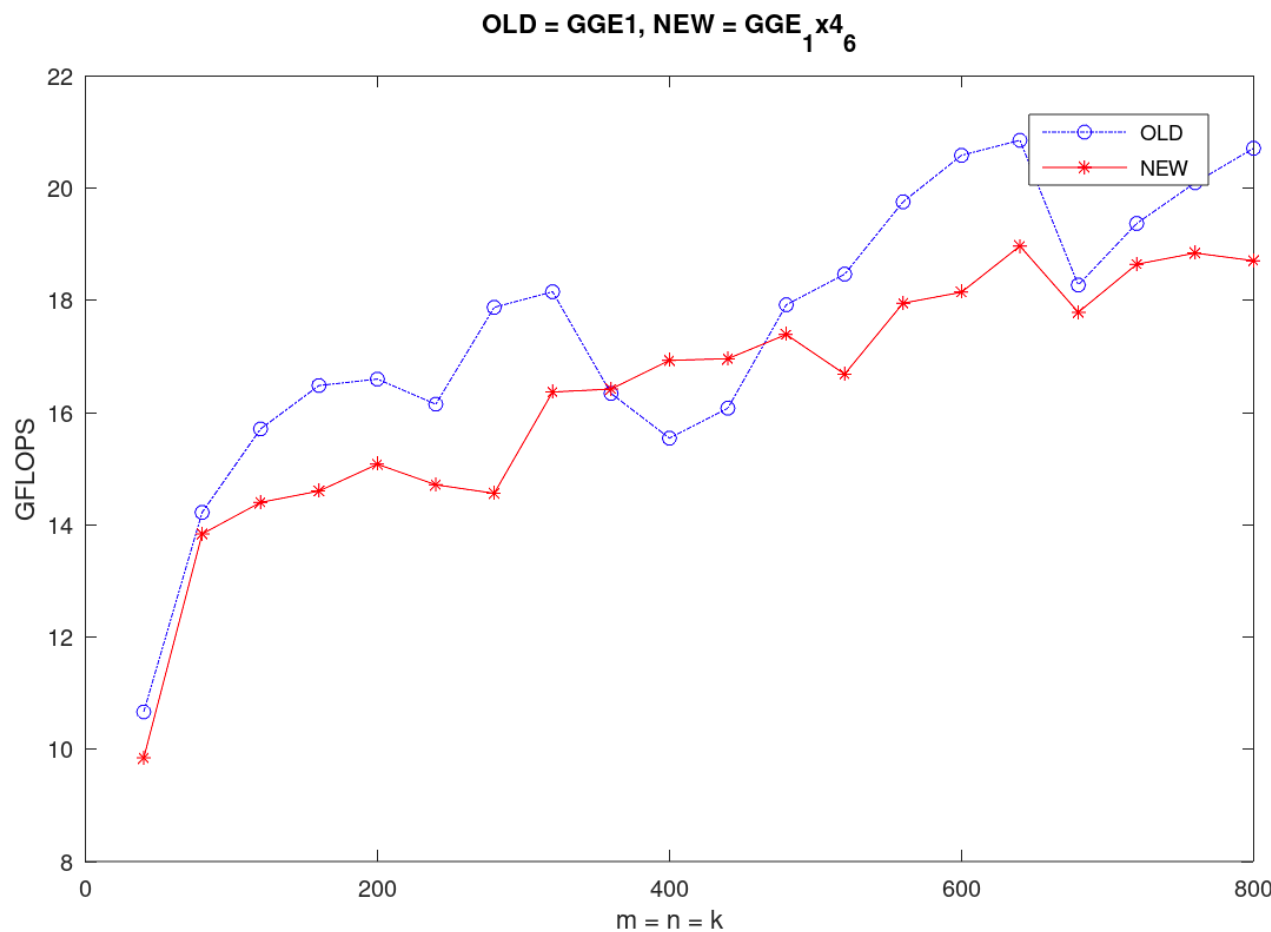
3. Optymalizacja GGE_1x4_4



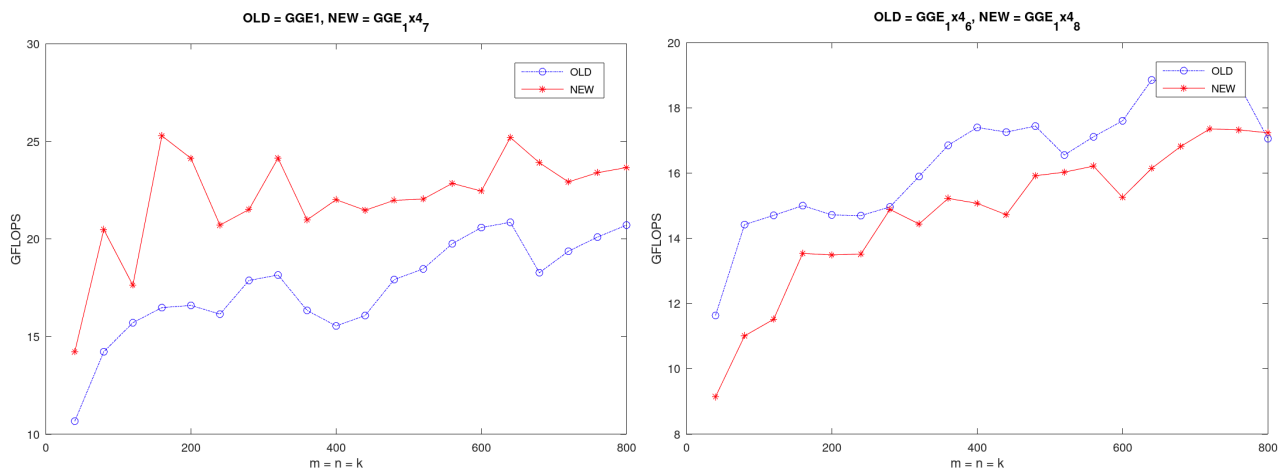
4. Optymalizacja GGE_1x4_5



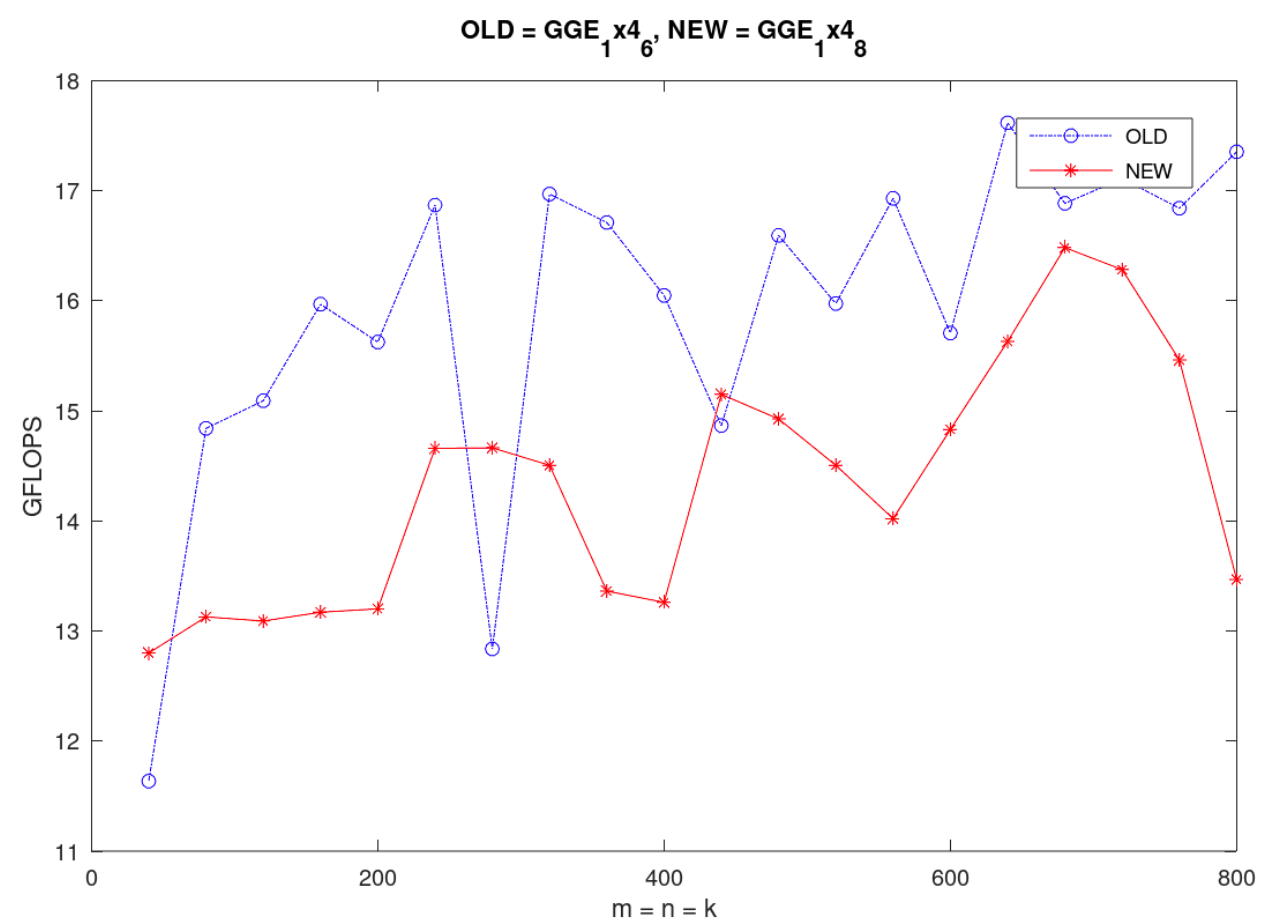
5. Optymalizacja GGE_1x4_6



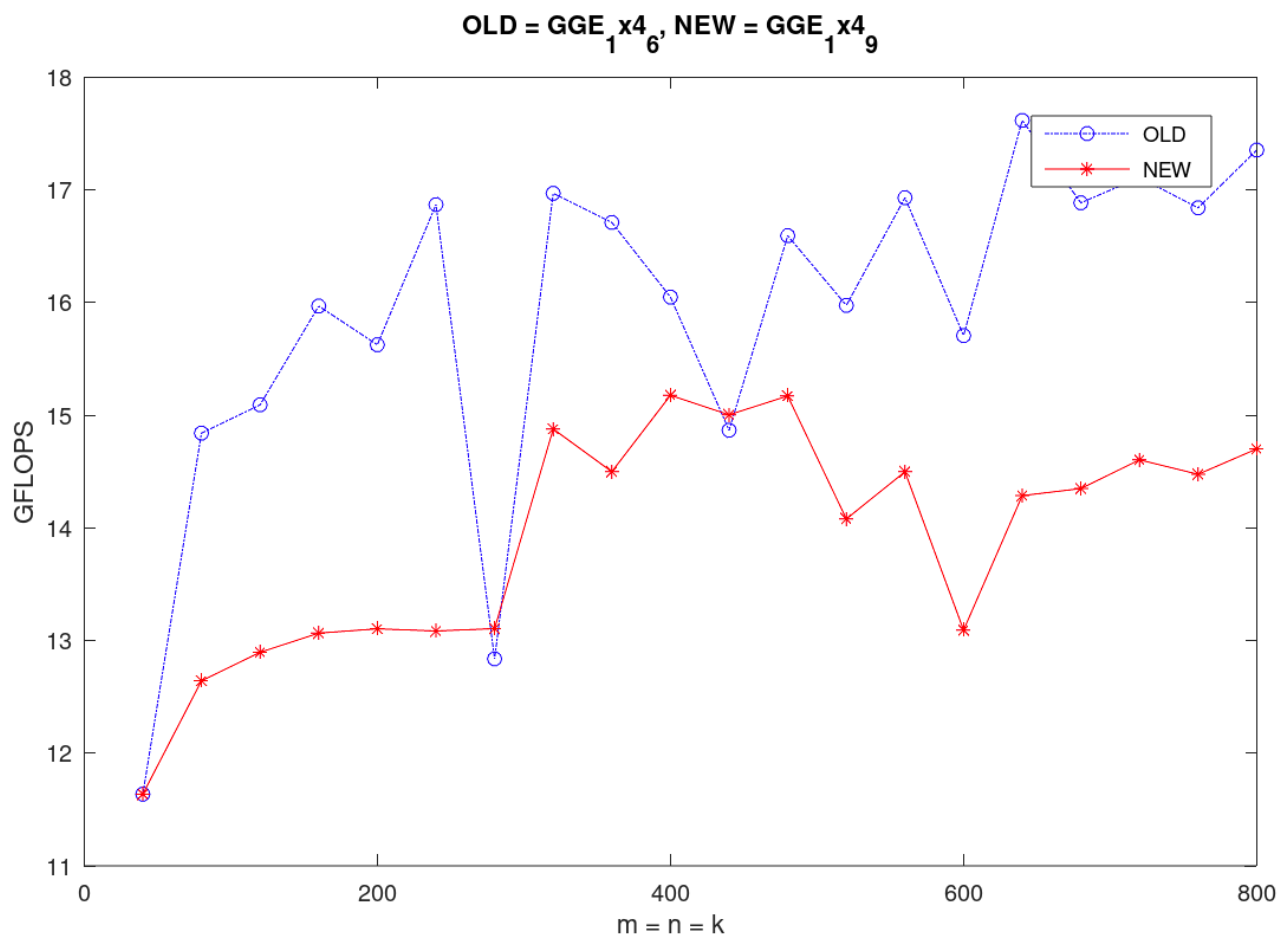
6. Optymalizacja GGE_1x4_7



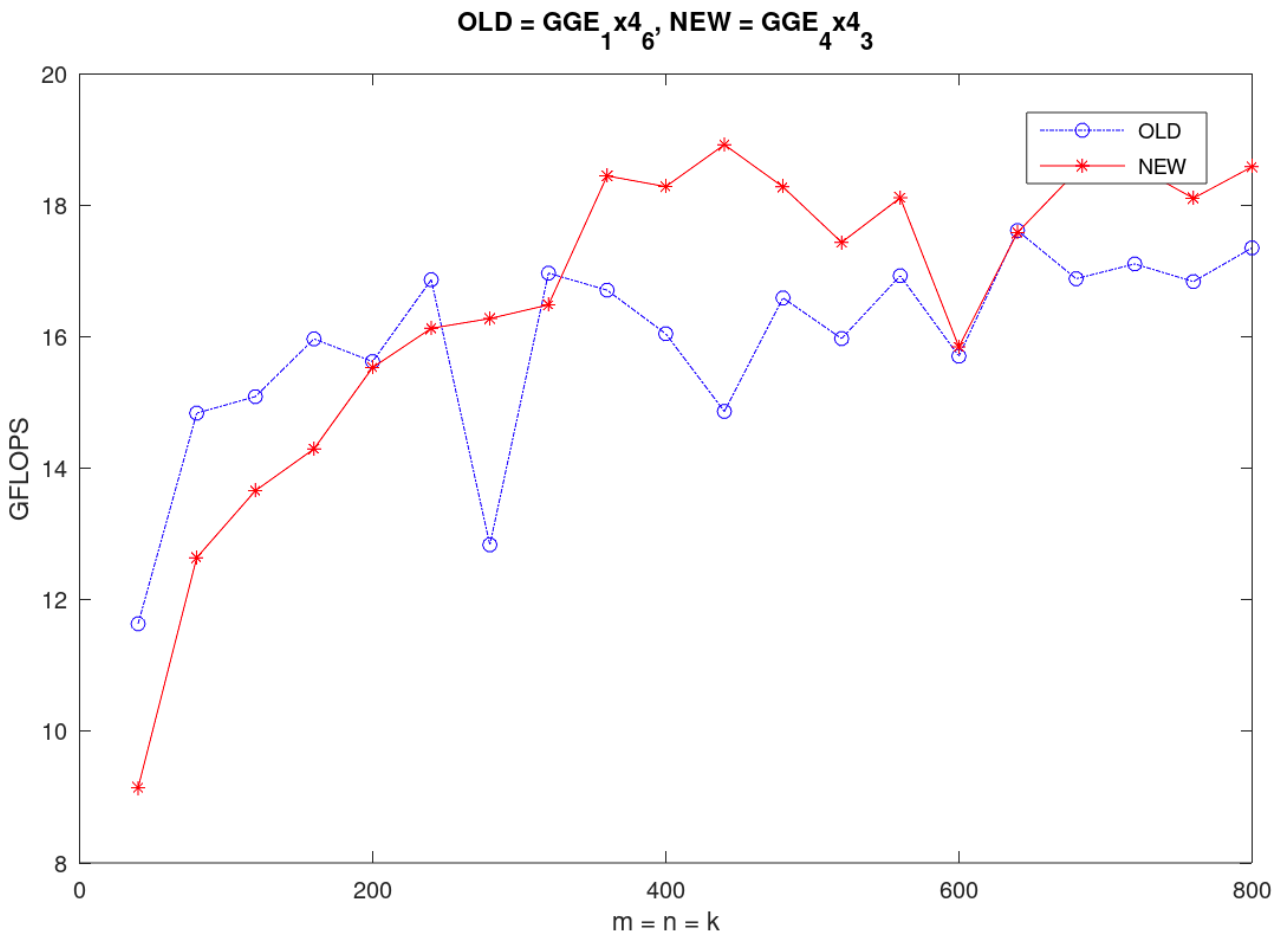
7. Optymalizacja GGE_1x4_8



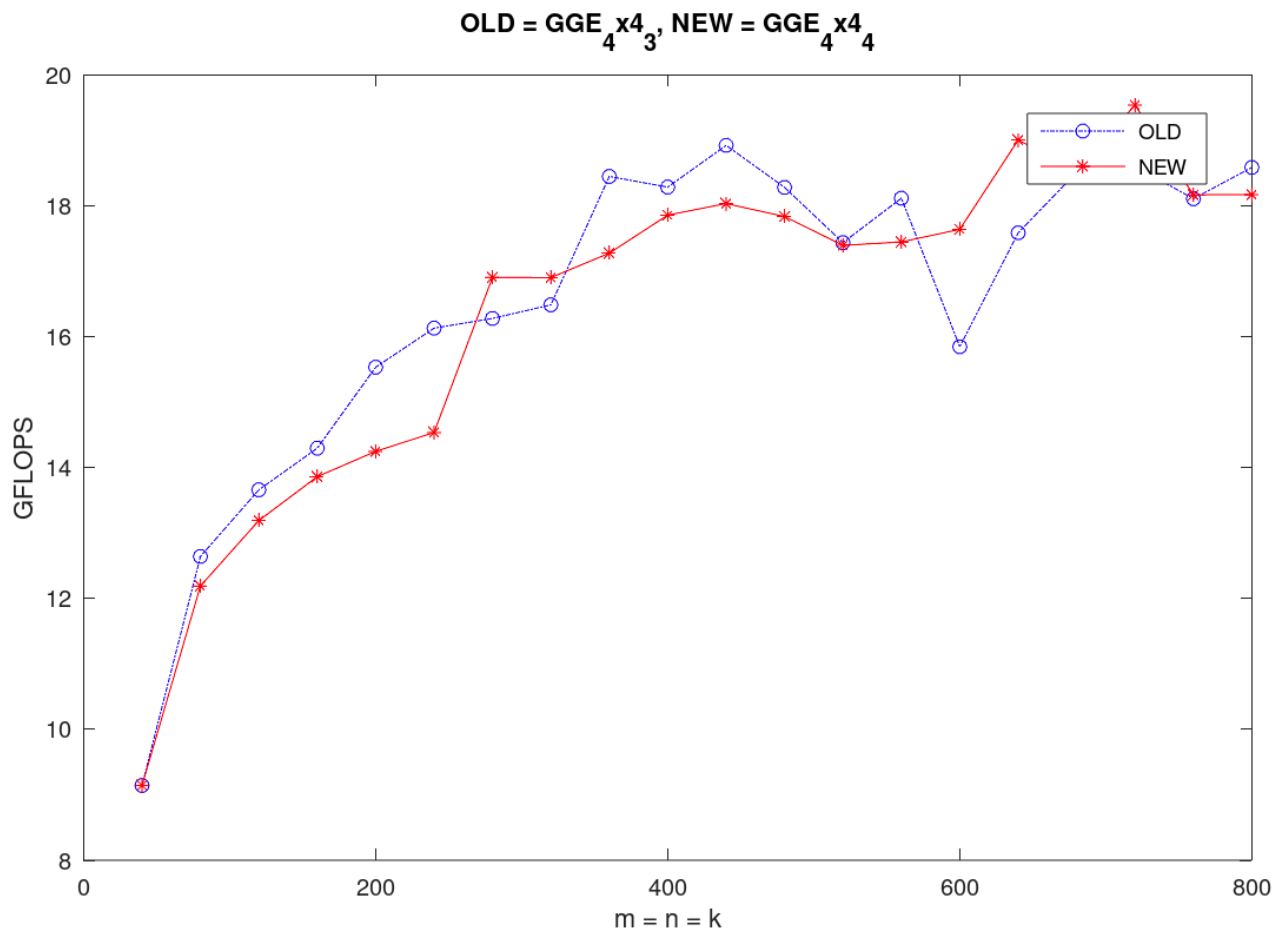
8. Optymalizacja GGE_1x4_9



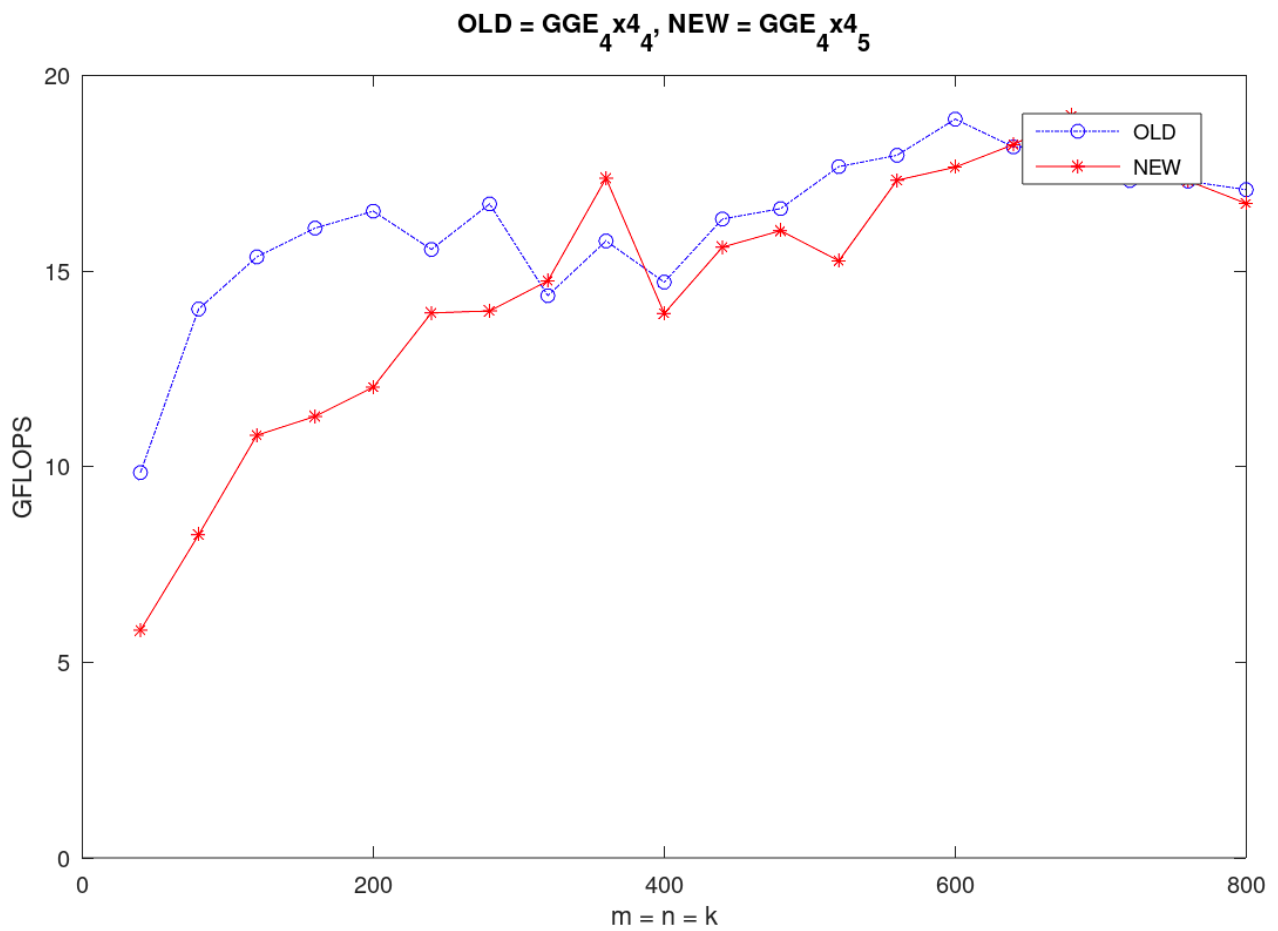
9. Optymalizacja GGE_4x4_3



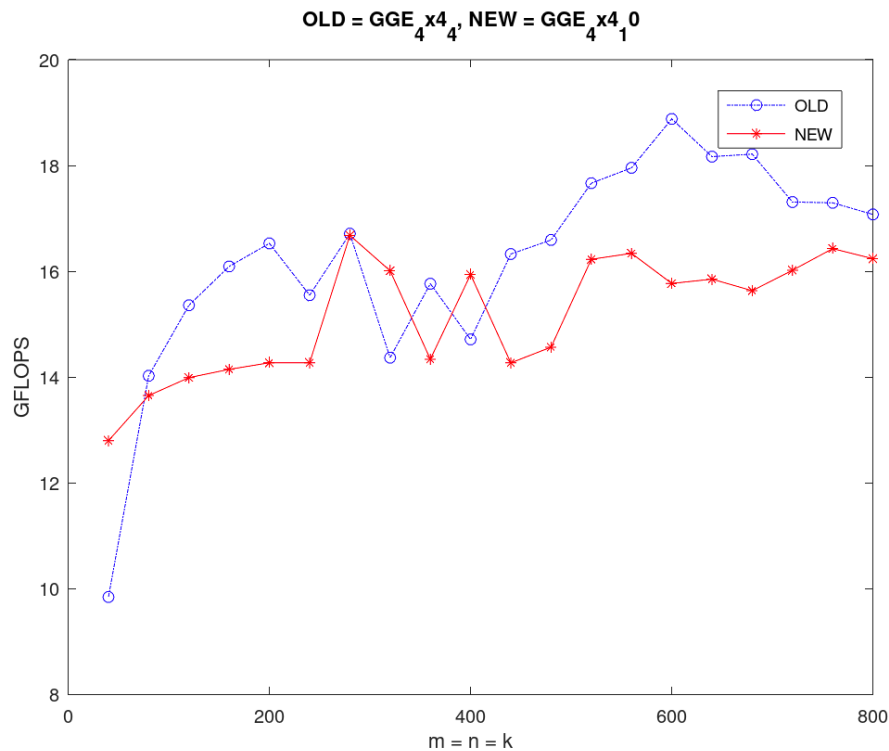
10. Optymalizacja GGE_4x4_4



11. Optymalizacja GGE_4x4_5



12. Optymalizacja GGE_4x4_10



13. Podsumowanie

Nie udało się osiągnąć dużej poprawy wydajności, różnica jest, jednak nie jest to bardzo znacząca różnica ani stała. Wynik wydajności w przypadku eliminacji Gaussa mocno zależała od “podejścia”.

