

Całkowanie metodą trapezową

Jarosław Niesyto
Języki Asemblerowe - Projekt
2019/2020

Założenia

Celem projektu było wykonanie aplikacji która będzie liczyć całkę metodą trapezową. Aplikacja miała wykorzystywać bibliotekę C++ oraz ASM. Dane miały być wczytywane z pliku tekstowego. Maksymalny rozmiar danych to 10000 punktów. Dane mają być typu float. Aplikacja ma działać w trybie konsolowym, z wykorzystaniem wielu wątków. Wybór ilość wątków oraz rodzaju biblioteki miał należeć do użytkownika.

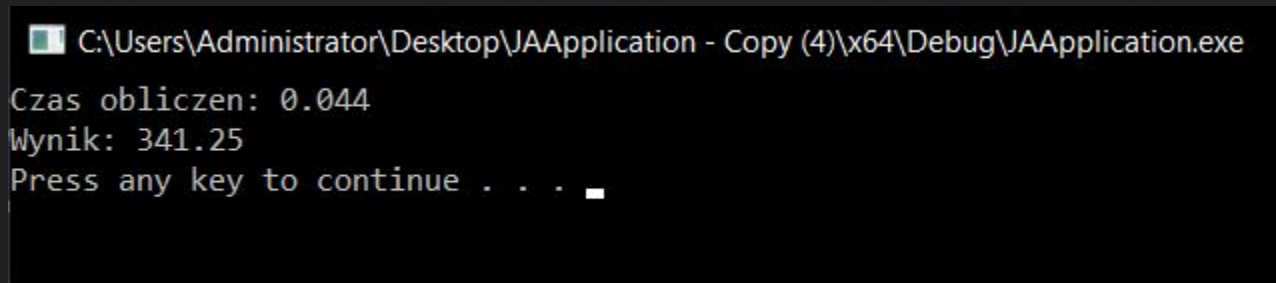
Dane wejściowe

Dane wejściowe przechowywane są w pliku tekstowym. Kolejne liczby oddzielone są białymi znakami. Każda para liczb tworzy jeden punkt.

1	5.5
2	3
3	-3
4	333
5	11
6	0
7	89
8	11
9	-6
10	6.6

Przykładowe dane testowe. W lewej kolumnie znajdują się współrzędne X, w prawej Y

Wynik działania programu. Widoczny jest czas obliczeń oraz wynik
(wyliczona całka)



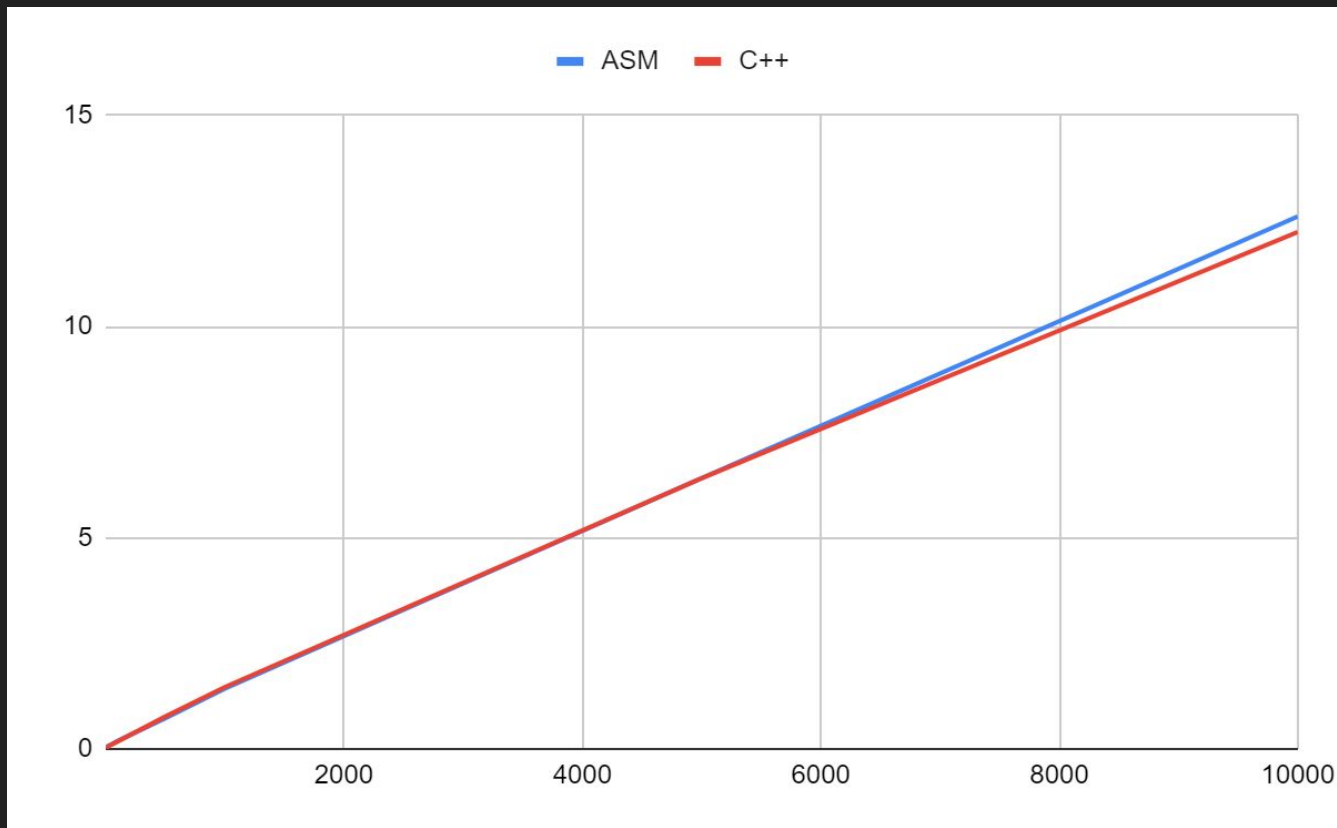
```
C:\Users\Administrator\Desktop\JAApplication - Copy (4)\x64\Debug\JAApplication.exe
Czas obliczen: 0.044
Wynik: 341.25
Press any key to continue . . .
```

Parametry

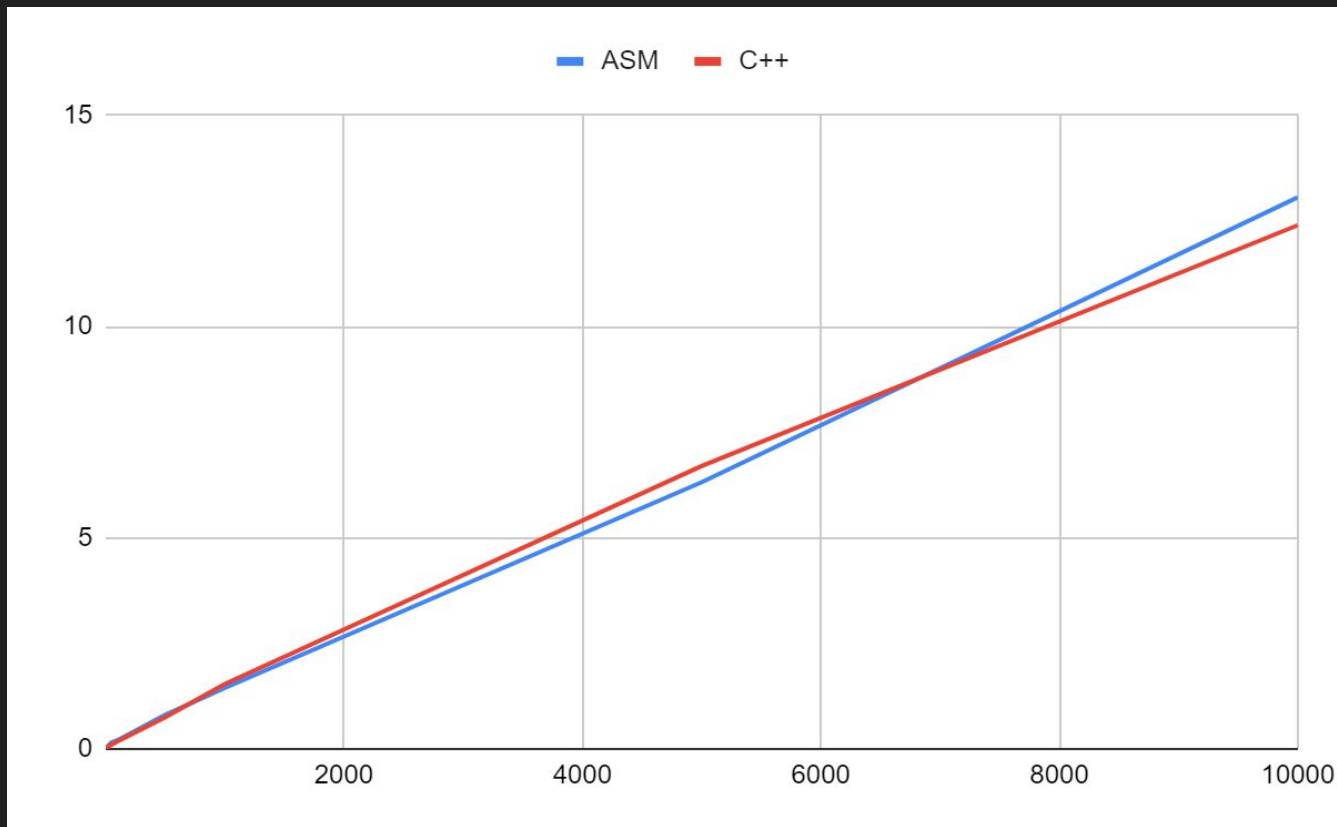
Użycie programu wymaga podania trzech parametrów

- i Ścieżka do pliku wejściowego
- t Ilość wątków
- d Wybór biblioteki, 0 to C++, 1 to ASM

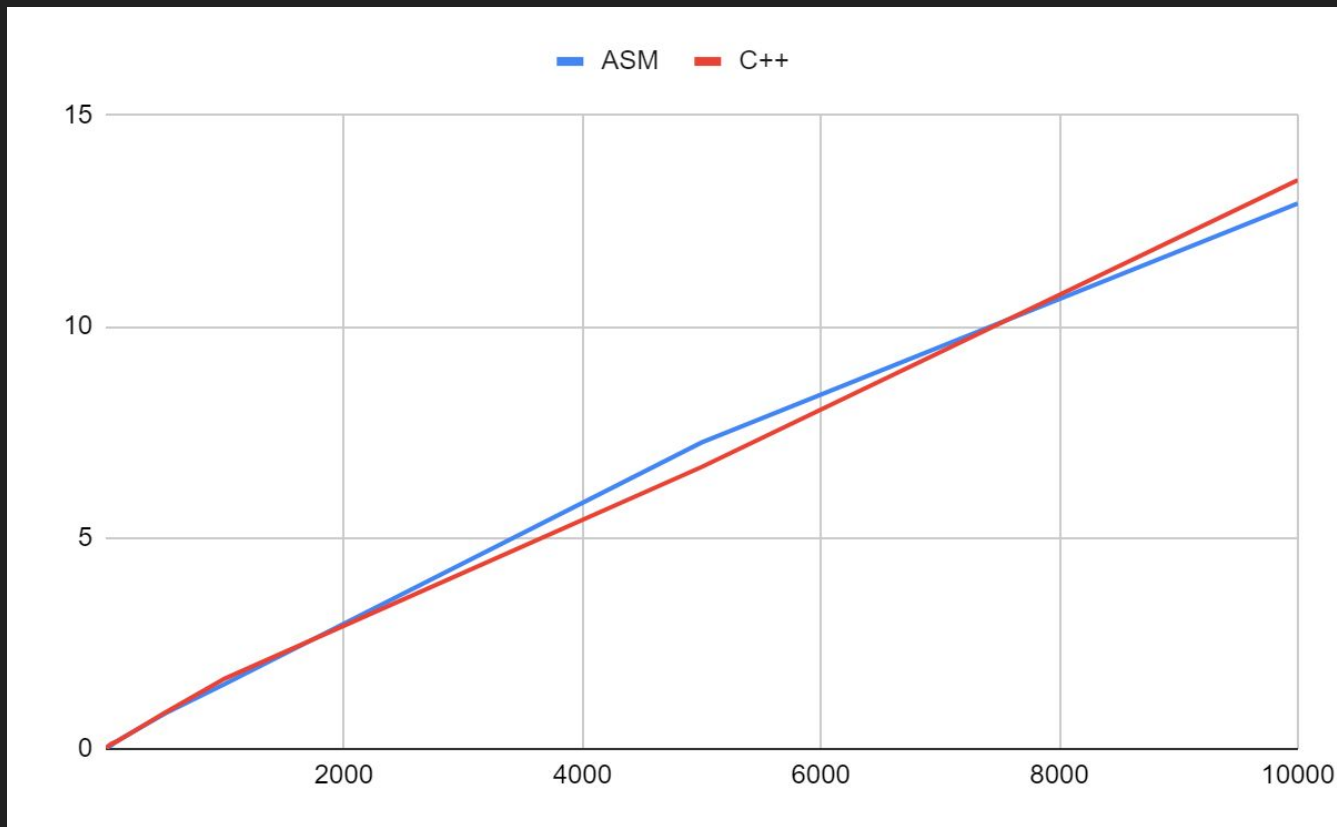
Porównanie wyników dla 8 wątków



Porównanie wyników dla 4 wątków

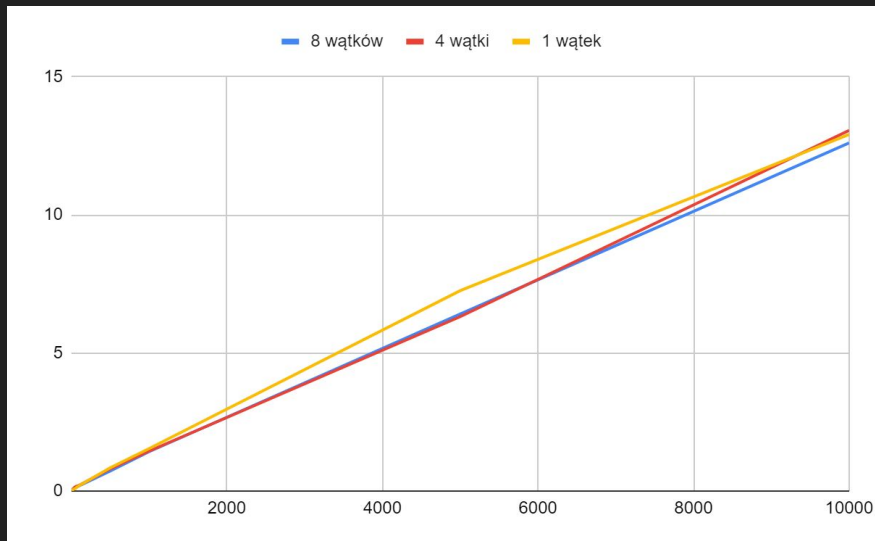


Porównanie wyników dla 1 wątku

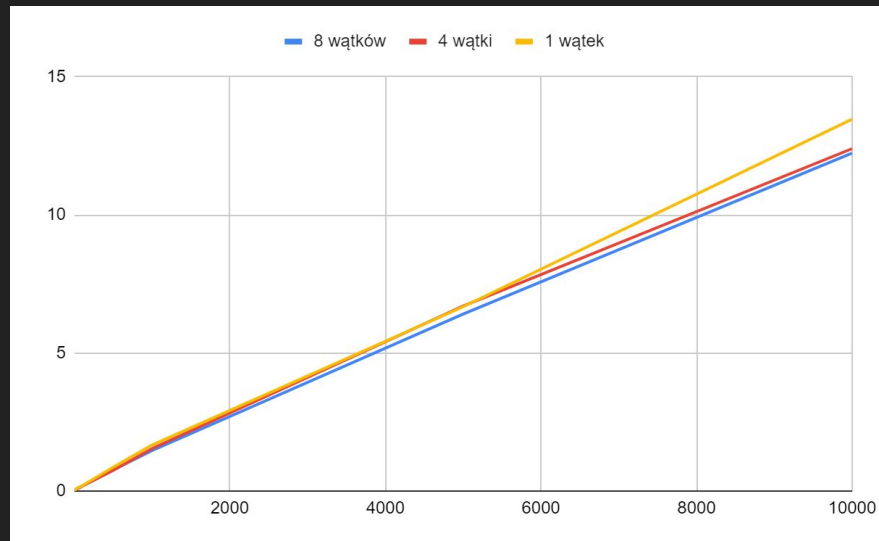


Porównanie wyników dla biblioteki ASM i C++

ASM



C++



Wnioski

Po przeanalizowaniu wykresów widać że czas działania aplikacji dla ośmiu wątków jest mniejszy niż dla jednego wątku. Różnica uwydatnia się przy większej ilości danych. Można też zauważyć że czas działania biblioteki ASM jest większy niż dla biblioteki C++ dla ośmiu i czterech wątków, natomiast odwrotna zależność jest do zaobserwowania dla jednego wątku.

Największym problemem w wykonaniu zadania okazało się połączenie biblioteki ASM z resztą programu. Problemów przysporzyło też zrozumienie gdzie znajdują się kolejne parametry.

Maksymalna ilość punktów danych dla których projekt był testowany to 1000000