Porównanie różnych metod całkowania numerycznego w wersji sekwencyjnej oraz równoległej

Jakub Ciechowski, 186471

2014.4.25

1 Opis problemu

W poniższym raporcie postaram się porównać różne metody całkowania numerycznego, zarówno pod względem dokładności jak i szybkości. W tym celu zaimplementowałem trzy różne metody, zarówno w wersji równoległej jak i sekwencyjnej.

1.1 Metody całkowania

Według definicji całki oznaczonej Riemana, wartość całki równa jest sumie pól obszarów pod wykresem krzywej w zadanym przedziale. Na tej podstawie wykorzystałem 3 różne metody całkowania.

Metoda trapezów Polega na podzieleniu obszaru na n trapezów, których suma pól pozwala nam poznać przybliżoną wartość całki. Wzór pozwalający nam dokonać obliczeń wygląda następująco:
$\int_a^b f(x)dx \approx h(\frac{y_1}{2} + y_2 + \ldots + y_n + \frac{y_{n+1}}{2}) = \frac{h}{2} \sum_{n=1}^{k}(f(x_i) + f(x_{i+1})$ [1]

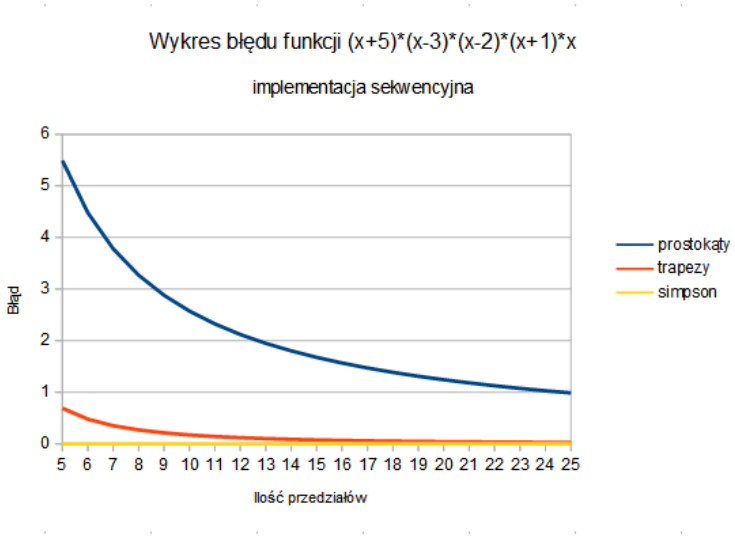
Metoda prostokątów Podobnie, jak metoda trapezów dzieli obszar na prostokąty a następnie oblicza sumę ich pól wg następującego wzoru:
$\int_{x_*}^{x_*+h} f(x)dx \approx hf(x_* + \frac{h}{2})$ [2]

Metoda Simpsona Wymaga podzielenia przedziału na parzystą liczbę podprzedziałów a następnie wykorzystuje wzór:
$\int_a^b f(x)dx \approx \frac{h}{3}[f_0 = 4(f_1 + f_3 + \ldots + f_{2n-1}) + 2(f_2 + f_4 + ... + f_{2n-2}) + f_2 n]$ [3]

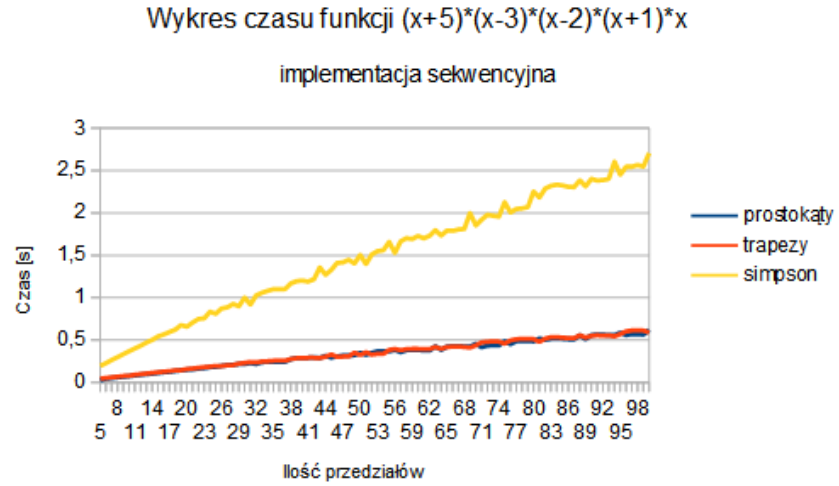
[1] źródło: Wikipedia
[2] źródło: Wikipedia
[3] źródło: Wikipedia

2 Rezultaty

2.1 Implementacja sekwencyjna



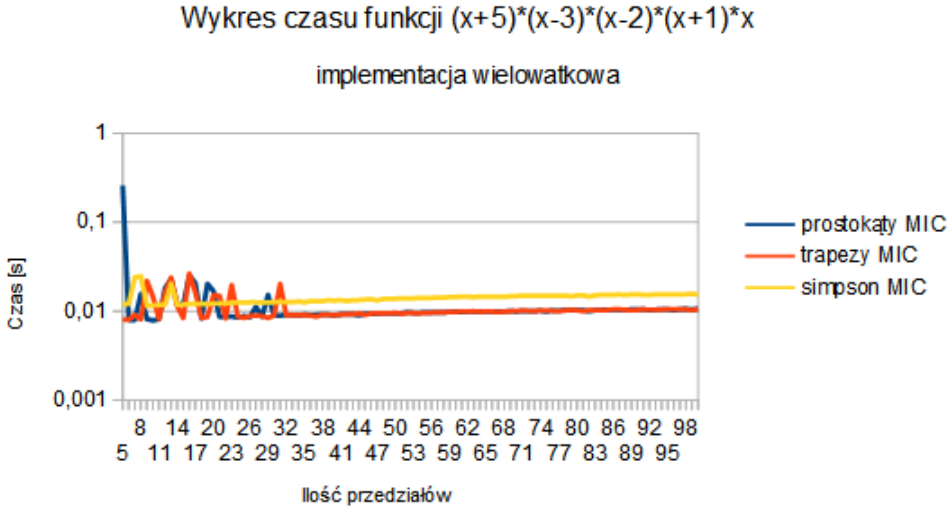
Momentalnie możemy zauważyć dokładność metody Simpsona, która już po 5 iteracji daje błąd parę rzędów wielkości mniejszy od metody trapezów i prostokątów. Można również zauważyć, że metoda prostokątów jest zdecydowanie najmniej precyzyjna. Podział pola pod wykresem na trapezy daje wyniki dokładniejsze niż podział na prostokąty ale i tak znacznie gorsze od metody Simpsona.



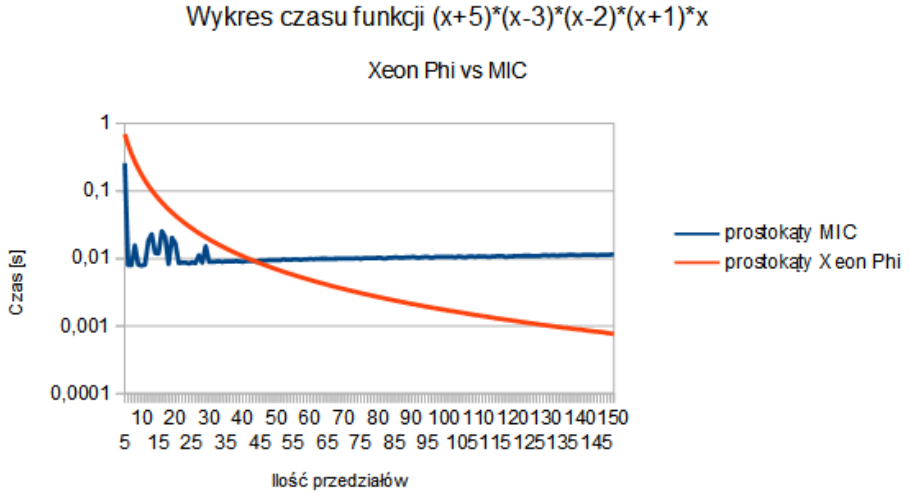
Jednak znacznie większa precyzja kosztuje również znacznie więcej czasu. Jak widać na wykresie, metoda Simpsona działa znacznie dłużej od metody trapezów i prostokątów, które działają praktycznie w tym samym czasie.

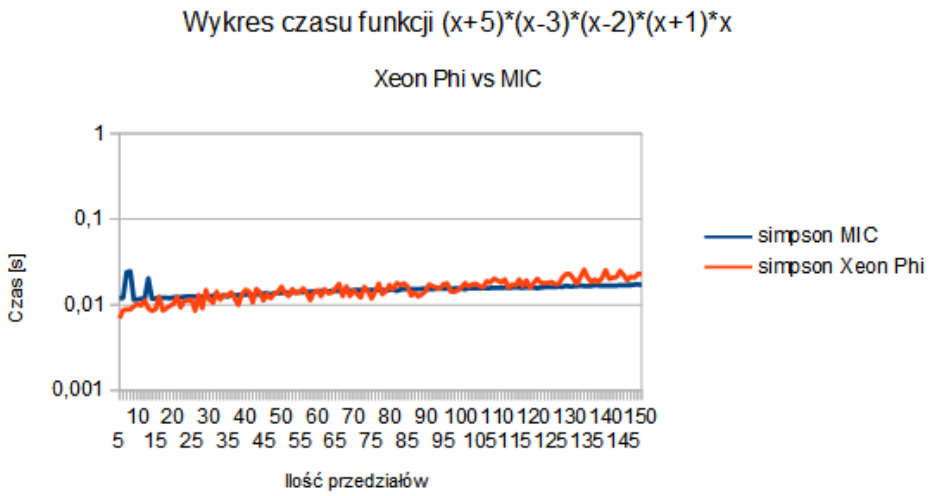
2.2 Implementacja wielowątkowa

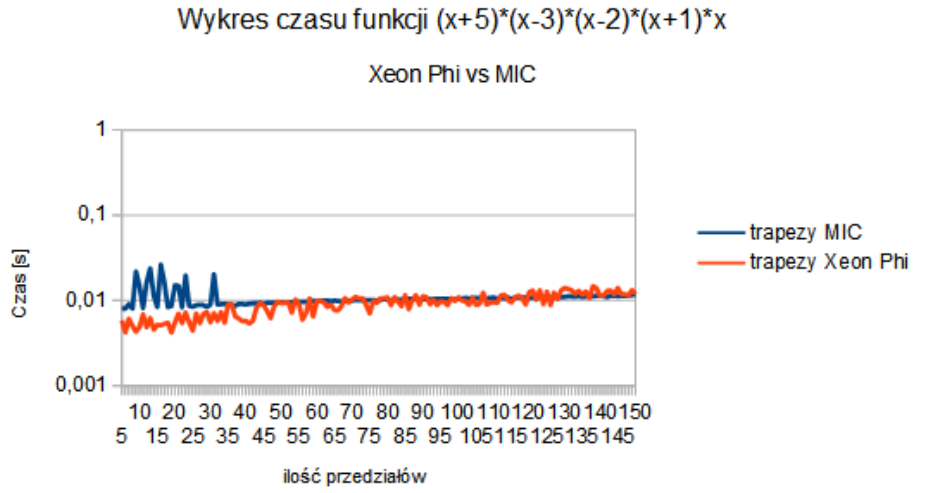
Poniżej widzimy wykres porównujący wszystkie trzy metody w wersji wielowątkowej na procesorze Xeon Phi



Powtarza się sytuacja z wersji sekwencyjnej. Chociaż różnice w czasie działania są znacznie mniejsze to wciąż metoda Simpsona jest najwolniejsza.
Porównanie czasu działania wersji równoległej dla procesora Xeon Phi z użyciem oraz bez użycia akceleratora.



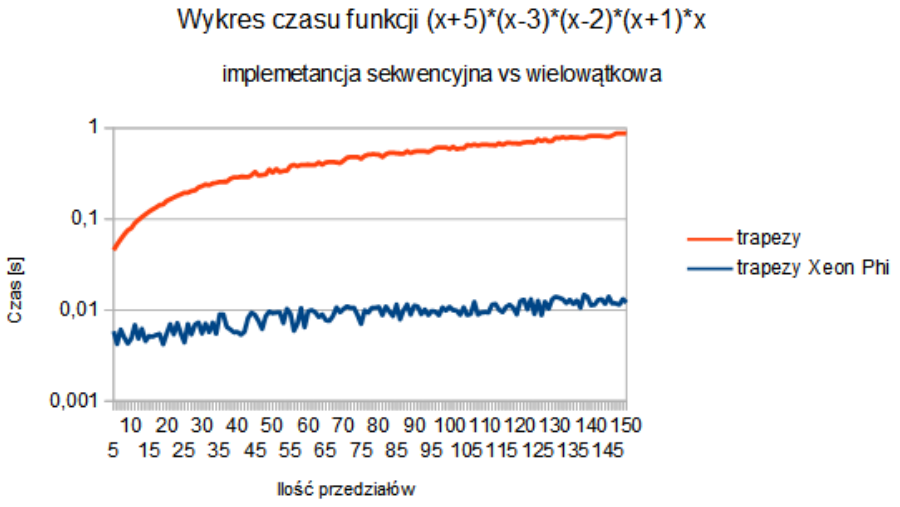




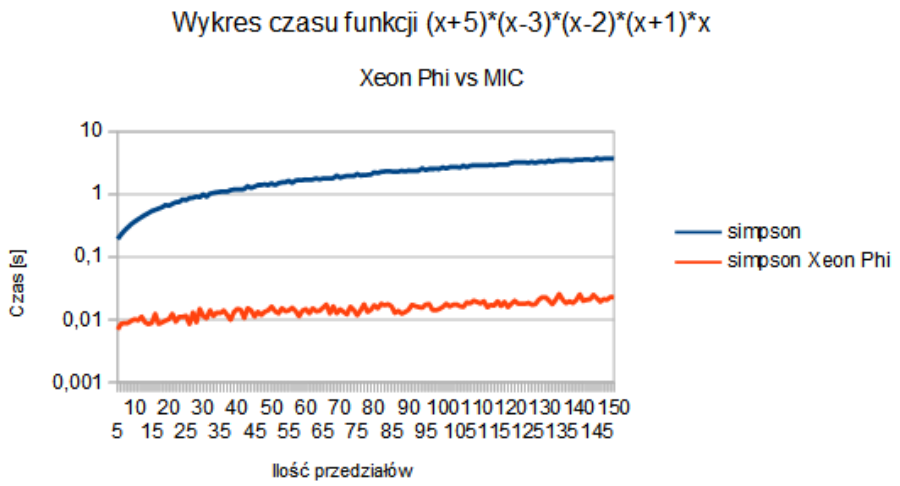
Najciekawsze rezultaty uzyskujemy dla metody prostokątów, która bez akceleracji działa znacznie szybciej.

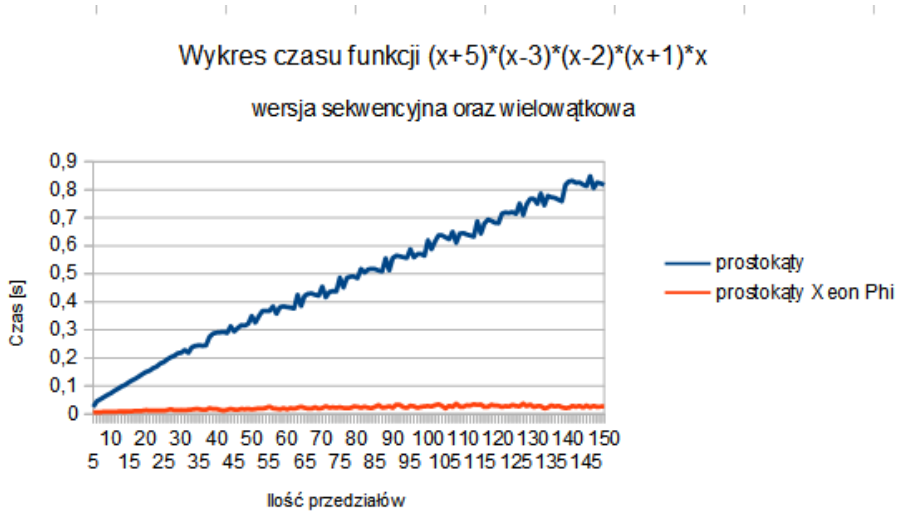
2.3 Porównanie implementacji sekwencyjnej oraz równoległej

Następnym krokiem było porównanie różnych metod w wersji sekwencyjnej oraz wielowątkowej.



Dla metody trapezów uzyskałem około dziesięciokrotne przyśpieszenie. Pierwsza wersja algorytmu korzystała z sekcji krytycznej, dzięki czemu szybkość wzrosła około dwukrotnie. Zastosowanie redukcji pozwoliło przyśpieszyć jeszcze pięciokrotnie.





Dla dwóch pozostałych metod sytuacja wygląda praktycznie tak samo, czego niestety nie oddaje wykres metody prostokątów ponieważ nie jest w skali logarytmicznej.

3 Wnioski

Jak widać, najbardziej rzuca się w oczy dokładność metody Simpsona, która daje bardzo dokładne wyniki już praktycznie po kilku iteracjach. Metoda trapezów oraz prostokątów nieznacznie się różniły od siebie ale znacznie odstawały od metody Simpsona.
Jednak, prezycja metody Simpsona ma swoją cenę którą jest czas. W niektórych przypadkach w zupełności wystarczy nam precyzja innych metod, które działają znacznie szybciej.
Jeżeli chodzi o przyśpieszenie to niestety nie udało mi się uzyskać znacznego przyśpieszenia dzięki akceleratorowi.