**Projekt Zaliczeniowy Pracowni Problemów Numerycznych**

Zrównoleglenie Algorytmów Runge-Kutty Rzędu Pierwszego, Drugiego oraz Czwartego w języku C#

*Kevin Springer*

Cel projektu był, aby zrównoleglić obliczenia z algorytmów Runge-Kutty pierwszego, drugiego oraz czwartego rzędu za pomocą języka C# oraz w nim zawartej biblioteki ThreadPool. Jest ona bardzo przydatna, ponieważ można dynamicznie dodawać obliczenia do puli wątków, w której obliczenia są równolegle wykonywane. Aby poprawnie zrównoleglić obliczenia w naszym przypadku, musimy zrównoleglić pętle, które się znajdują w funkcjach OdeInt\_X, ponieważ w tych obliczamy po kolei każde równania. Zwykle zastosowanie wątków lub puli wątków jest po to, aby przyspieszyć działanie programu tak szybko jak możliwe. Niestety w naszym przypadku używając puli wątków, wszystkie obliczenia, które są wykonywane w puli wątków, są wykonywane asynchronicznie. Ma to duży wpływ na wyniki, ponieważ mogą wyjść chaotyczne wyniki lub mogą być po prostu do pierwotnych obliczeń (bez puli wątków) niepodobne. Dlatego niezbędne jest korzystanie z blokad, które wstrzymują obliczenia innych wątków do momentu aż pierwszy w kolejce wykona swoje zadanie. Po implementacji blokady, wciąż może się zdarzyć, że wyniki nie będą podobne do oryginalnych wyników, jednakże różnica jest niewielka. Natomiast główny problem tkwi w czasie obliczeń. Mając mały krok całkowania czyli h=0.01 , obojętnie czy z wątkami czy bez program się wykonuje w mniej niż sekunda. Natomiast po zwiększeniu kroku całkowania na np. h=0.00001 prędkość się gwałtownie zmienia w przypadku korzystania z puli wątków. Bez puli wątków program obliczający MidPoint obliczy się w 2 sekundy, natomiast z wątkami już 24 sekund. Niestety nawet zmiana ilości wątków za pomocą ThreadPool.SetMaxThreads lub ThreadPool.SetMinThreads nie zmienia czas obliczenia na naszą korzyść. Na rysunku przedstawiłem przykład dla N = 100 dla od jednego do czterech wątków. Z jednym wątkiem wykonuje zadanie w pół sekundy a mając 4 wątki trwają obliczenia dwa razy dłużej.

