

Sprawozdanie z projektu - Opracowanie algorytmu równoległej integracji numerycznej przy użyciu reguły trapezów

Przedmiot: Obliczenia inżynierskie w chmurze

Autor: Marek Perko 336486

Data: 19.01.2025

1. Wprowadzenie

Celem projektu było opracowanie aplikacji wykonującej równoległą integrację numeryczną przy użyciu reguły trapezów. Program pozwala użytkownikowi wybrać funkcję matematyczną, zakres całkowania, liczbę podprzedziałów oraz liczbę procesorów. Wyniki obliczeń są wyświetlane w konsoli.

Dodatkowo aplikacja została wdrożona na platformie Microsoft Azure z wykorzystaniem Dockera oraz FileZilla co umożliwia jej uruchamianie w środowisku wirtualnym.

2. Opis aplikacji

2.1. Działanie programu

Aplikacja została napisana w języku Python i działa w konsoli. Wykorzystuje:

- Biblioteki: `numpy`, `multiprocessing`, `matplotlib`.
- Funkcjonalności:

Wybór jednej z czterech funkcji matematycznych:

1. $f(x)=\sin(x)+x^2$
2. $f(x)=e^{-x^2}$
3. $f(x)=\ln(x+10)$
4. $f(x)=x^3-2x$

Ustawianie zakresu całkowania $[a,b][a, b][a,b]$.

Wybór liczby podprzedziałów (nnn) oraz liczby procesorów do równoległych obliczeń.

Wyświetlanie wyników w konsoli:

- Wzór funkcji.
- Zakres całkowania.
- Liczba podprzedziałów i procesorów.
- Wynik całki.

Generowanie wykresu funkcji na zadanym zakresie z zaznaczeniem obszaru całkowania.

2.2. Algorytm reguły trapezów

Aplikacja wykorzystuje regułę trapezów, dzieląc zakres całkowania na n równych przedziałów. Obliczenia są przyspieszane przez zastosowanie równoległości z użyciem modułu `multiprocessing`.

3. Wdrożenie aplikacji na platformie Azure

3.1. Tworzenie maszyny wirtualnej

1. Utworzono maszynę wirtualną z systemem **Ubuntu Server 20.04 LTS**.
2. Skonfigurowano maszynę z użyciem:
 - Rozmiar: **Standard_D2s_v3** (2 CPU, 8 GB RAM).
 - Login oraz hasło do logowania.
 - Otwarty port 22 (dla SSH).

^ Podstawowe elementy	
Grupa zasobów (przenies) : Dis_group	System operacyjny : Linux (ubuntu 20.04)
Stan : Uruchomione	Rozmiar : Standard D2s v3 (2 procesory wirtualne vcpu, 8 GiB pamięci)
Lokalizacja : West Europe (Strefa 1)	Publiczny adres IP : 74.234.196.81
Subskrypcja (przenies) : Azure for Students	Sieć/podsieć wirtualna : Dis-vnet/default
Identyfikator subskrypcji : 09d55b76-e0b7-417b-9b9b-249944a9d140	Nazwa DNS : Nieskonfigurowano
Strefa dostępności : 1	Stan kondycji : —
Tagi (edytuj) : Dodaj tagi	Godzina utworzenia : 18.01.2025, 22:51 UTC

Rys. 1. Parametry maszyny wirtualnej

3.2. Przesyłanie plików za pomocą FileZilla

1. Połączono się z maszyną wirtualną za pomocą FileZilla (protokół SFTP).
2. Przesłano pliki:
 - `Projekt.py` – kod aplikacji.
 - `Dockerfile` – plik konfiguracyjny Dockera.

3.3. Konfiguracja Dockera

1. Na maszynie zainstalowano Dockera:

```
sudo apt-get update
sudo apt-get install docker.io
```

2. Zbudowano obraz Dockera z plikiem `Dockerfile`:

```
sudo docker build -t projekt_integral .
```

3. Uruchomiono kontener Dockera:

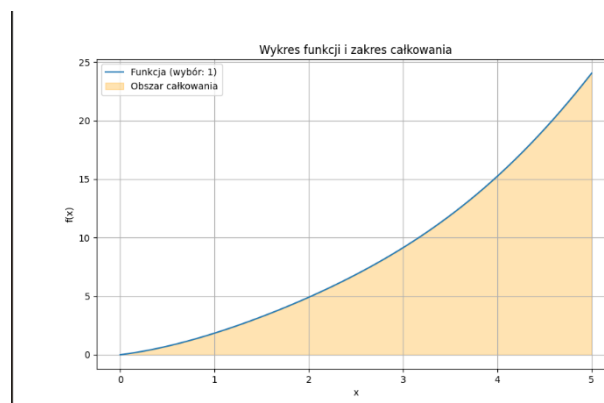
```
sudo docker run --name integral_container -it projekt_integral
```

4. Wyniki testów

5. 1. Funkcja $f(x)=\sin(x)+x^2$

```
Projekt@Projekt:~$ sudo docker run --name integral_container -it projekt_integral
Wybierz funkcję:
1. f(x) = sin(x) + x^2
2. f(x) = exp(-x^2)
3. f(x) = log(x + 10)
4. f(x) = x^3 - 2 * x
Wybór funkcji (1-4): 1
Podaj początek zakresu całkowania (a): 0
Podaj koniec zakresu całkowania (b): 5
Podaj liczbę podprzedziałów (n): 1000
Podaj liczbę procesorów: 4
--- Wyniki ---
Wybrana funkcja: 1
Zakres całkowania: [0.0, 5.0]
Liczba podprzedziałów: 1000
Liczba procesorów: 4
Wynik całki: 42.383024
```

Rys 2. Wyniki obliczeń

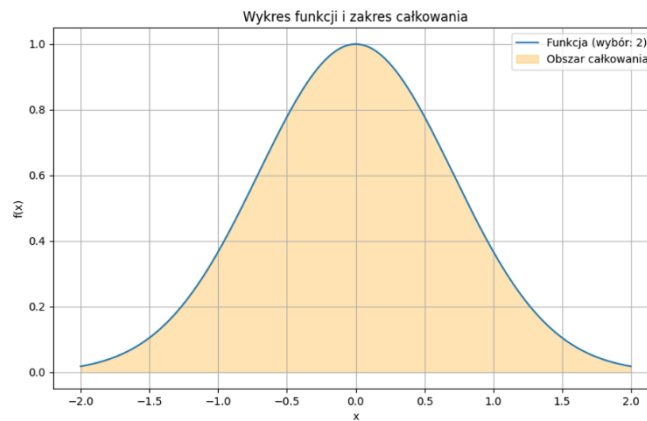


Wykres 1. Wykres funkcji wygenerowany przez program

6. 2. Funkcja $f(x)=e^{-x^2}$

```
Projekt@Projekt:~$ sudo docker start -i integral_container
Wybierz funkcję:
1. f(x) = sin(x) + x^2
2. f(x) = exp(-x^2)
3. f(x) = log(x + 10)
4. f(x) = x^3 - 2 * x
Wybór funkcji (1-4): 2
Podaj początek zakresu całkowania (a): -2
Podaj koniec zakresu całkowania (b): 2
Podaj liczbę podprzedziałów (n): 500
Podaj liczbę procesorów: 2
--- Wyniki ---
Wybrana funkcja: 2
Zakres całkowania: [-2.0, 2.0]
Liczba podprzedziałów: 500
Liczba procesorów: 2
Wynik całki: 1.764162
```

Rys 3. Wyniki obliczeń dla drugiej próby



Wykres 2. Wykres funkcji dla drugiej próby wygenerowany przez program

5. Wnioski

1. Funkcjonalność programu:

- Aplikacja działa poprawnie, umożliwiając wybór funkcji, zakresu całkowania, liczby podprzedziałów i procesorów.
- Wykorzystanie reguły trapezów w połączeniu z równoległością pozwala na szybkie obliczenia nawet dla dużej liczby podprzedziałów.

2. Zastosowanie Dockera:

- Użycie Dockera zapewnia przenośność aplikacji i łatwość wdrażania na różnych platformach.

3. Platforma Azure:

- Wykorzystanie maszyny wirtualnej w chmurze pozwala na uruchamianie aplikacji w środowisku niezależnym od lokalnych zasobów.

4. Możliwe ulepszenia:

- Dostosowanie programu do automatycznego doboru liczby podprzedziałów w zależności od tolerancji błędu.
- Wdrożenie graficznego interfejsu użytkownika (GUI) w oparciu o technologie webowe.