Sprawozdanie z ćwiczenia 6

Rozwiązywanie układów równań liniowych metodami bezpośrednimi

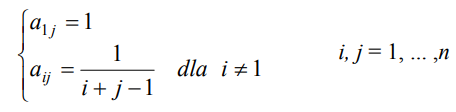
Konrad Pękala

## 1. Wstęp

### Użyte programy/biblioteki:

1. Jupyter Notebook
2. Python 3
3. NumPy, Matplotlib

## 2. Zadanie nr.1

Polecenie: rozwiązać układ równań liniowych **A**\***x** = **b** metodą Gaussa dla macierzy: 

### Wykonanie ćwiczenia:

Aby obliczyć **b** przyjąłem wektor **x** złożony tylko z jedynek.

**x** = [1,1,1,1…..] – zadana wartość

**x0** – obliczone **x** w algorytmie

Przyjąłem dwie precyzje: np.float64 (64 bity) oraz np.float32 (32 bity)

Aby pokazać różnice w eksperymentach, dokładność algorytmu będzie reprezentowana jako euklidesowa odległość obliczonego **x0** od zadanego **x**

**α** – dokładność (błąd)

### Wyniki dla **x**:

|  |  |  |
| --- | --- | --- |
|  | **Błąd** (dokładność do 7 miejsc po przecinku) | |
| **Rozmiar** | np.float32 | np.float64 |
| 2 | 0.0 | 0.0 |
| 3 | 1.1e-06 | 0.0 |
| 4 | 8.6e-06 | 0.0 |
| 5 | 0.0003825 | 0.0 |
| 6 | 0.021513 | 0.0 |
| 7 | 0.762003 | 0.0 |
| 8 | 2.2482439 | 1.1e-06 |
| 9 | 6.0443366 | 1.24e-05 |
| 10 | 5.575116 | 0.0007412 |

## 3. Zadanie nr.2

Polecenie: powtórzyć eksperyment dla macierzy zadanej wzorem:

Obraz zawierający tekst

Opis wygenerowany automatycznie

### Wyniki dla **x**:

|  |  |  |
| --- | --- | --- |
|  | **Błąd** (dokładność do 7 miejsc po przecinku) | |
| **Rozmiar** | np.float32 | np.float64 |
| 2 | 0.0 | 0.0 |
| 3 | 1e-07 | 0.0 |
| 4 | 2e-07 | 0.0 |
| 5 | 2e-07 | 0.0 |
| 6 | 4e-07 | 0.0 |
| 7 | 7e-07 | 0.0 |
| 8 | 1e-06 | 0.0 |
| 9 | 1.6e-06 | 0.0 |
| 10 | 2.7e-06 | 0.0 |

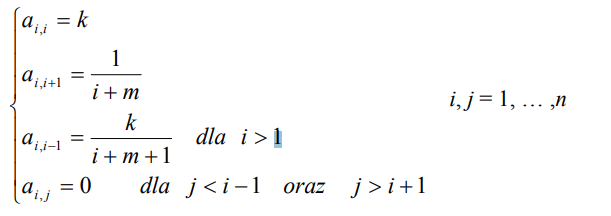
## 4. Zadanie nr.3

Polecenie: powtórzyć eksperyment dla macierzy zadanej wzorem poniżej. Następnie rozwiązać układ metodą przeznaczoną do rozwiązywania układów z macierzą trójdiagonalną.

### Wykonanie ćwiczenia:

Wykonanie ćwiczenia jest powtórzeniem eksperymentu dla nowej macierzy, z tym że skorzystamy z dwóch metod: wcześniejszej metody Gaussa oraz metody Thomasa

Macierz **A**:



Dla k = 6, m = 4

### Wyniki metody Gaussa:

|  |  |  |
| --- | --- | --- |
|  | **Błąd** (dokładność do 10 miejsc po przecinku) | |
| **Rozmiar** | np.float32 | np.float64 |
| 2 | 3.19e-08 | 0.0 |
| 3 | 4.96e-08 | 0.0 |
| 4 | 5.28e-08 | 0.0 |
| 5 | 6.1e-08 | 0.0 |
| 6 | 6.32e-08 | 0.0 |
| 7 | 6.8e-08 | 0.0 |
| 8 | 6.91e-08 | 0.0 |
| 9 | 6.97e-08 | 0.0 |
| 10 | 7.54e-08 | 0.0 |

### Wyniki metody Thomasa:

|  |  |  |
| --- | --- | --- |
|  | **Błąd** (dokładność do 10 miejsc po przecinku) | |
| **Rozmiar** | np.float32 | np.float64 |
| 2 | 5.96e-08 | 0.0 |
| 3 | 5.96e-08 | 0.0 |
| 4 | 5.96e-08 | 0.0 |
| 5 | 5.96e-08 | 0.0 |
| 6 | 5.96e-08 | 0.0 |
| 7 | 5.96e-08 | 0.0 |
| 8 | 5.96e-08 | 0.0 |
| 9 | 5.96e-08 | 0.0 |
| 10 | 5.96e-08 | 0.0 |

## Porównanie czasu działania obu metod:

|  |  |  |
| --- | --- | --- |
|  | **Czas działania** | |
| **Rozmiar** | **Metoda Gaussa** | **Algorytm Thomasa** |
| 90 | 0.109024 | 0.001 |
| 91 | 0.110025 | 0.0 |
| 92 | 0.113025 | 0.001 |
| 93 | 0.117027 | 0.001 |
| 94 | 0.120027 | 0.001 |
| 95 | 0.123029 | 0.001 |
| 96 | 0.128528 | 0.0 |
| 97 | 0.132031 | 0.0 |
| 98 | 0.132497 | 0.001 |
| 99 | 0.139009 | 0.001 |
| 100 | 0.142011 | 0.0 |