Zadanie numeryczne nr 4

Kamila Dmowska

December 2021

1 Wstęp

Celem zadania jest rozwiązanie układu równań $(N + 1) \times (N + 1)$.

$$\begin{cases} y_0 & = 1 \\ (D_2 y)_n + y_n & = 0\,, \quad n = 1 \ldots (N-1) \\ y_{N-1} - 2y_N + y_0 & = 0\,, \end{cases}$$

gdzie $N = 1000$, $h = 0.01$, a

$$(D_2 y)_n = \frac{y_{n-1} - 2y_n + y_{n+1}}{h^2}.$$

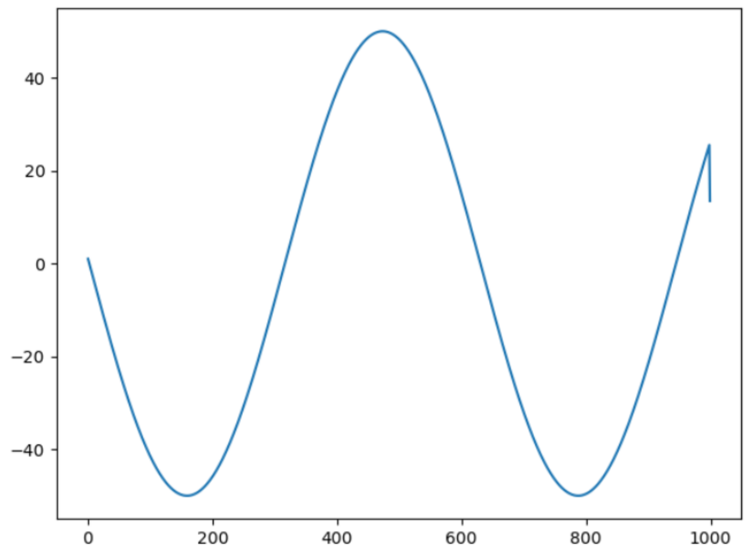
2 Implementacja

W porównaniu do zadania N4, ten układ równań trzeba będzie rozwiązać w sposób rekurencyjny. Poniżej znajduje się kod. Przekształcając równania otrzymujemy $y_{n-1} = 2y_n - y_{n+1} - h^2 y_n$ oraz $y_n = \dfrac{-y_{n-1} - y_{n1}}{-2 + h^2}$

```
import numpy as np
import matplotlib.pyplot as plt
N = 1000
h = 0.01
y = np.zeros(N, dtype=np.float64)
y[0] = 1

def Y(n):

    if n == 0:
        return y[0]
    if n >= 2:
        y_n1 = Y(n-1)
        y_n2 = y[n-2]
```

```
        y[n] = 2 * y_n1 - y_n2 - h * h * y_n1

        return y[n]

    if n == 1:
        y_n_minus1 = Y(n-1)
        y_n_plus1 = y[n+1]

        y[n] = (-y_n_minus1 - y_n_plus1) / (-2 + h**2)

        return y[n]

a = Y(N-1)

y[N-1] = (a + y[0]) / 2
plt.plot(y)
plt.show()
```

Na początku, inincjalizuję tablicę y za pomocą np.zeros, czyli n-elentowa tablicę w ktorej wszystkie elementy sa zerami. Następnie, przypisuje pierwszy elemement tablicy równy 1, który był podany w treści zadania. W kolejnym kroku, tworzę funkcję rekurencujną. Warunek stopu jest wtedy, kiedy n == 0. Funkcja zaczyna się od przedostantniego elementu i kończy się na zerowym. W każdym wywołaniu funkcji, oprócz części, z warunkiem stopu, przypisuję w tablicy y elementowi n odpowiednia wartość z przekształconego wzoru podanego w treści zadania. Na końcu, po przejściu przez funkcję rekurencyjną przez wszystkie elementy tablicy y wyliczam ostatnią wartość N z przekształconego wzoru $y_N = \frac{y_{N-1} + y_0}{2}$.



Rysunek 1: Wykres (nh, y_n).

3 Wnioski

Suma złożoności wynosi $O(n+n)$, wynika to z tego, że na początku inicjalizujemy wektor y wartościami 0 za pomocą funkcji np.zeros() z biblioteki numpy oraz poźniej wykonujemy obliczenia iteracyjne w pętli. Jednak zgodnie z definicją, upraszczamy ten wynik do $O(n)$.