Metody numeryczne - N9

DAMIAN PORADYŁO

N9 *Zadanie numeryczne*
Znaleźć cztery najmniejsze wartości i wektory własne macierzy $N \times N$ z dokładnością 10^{-8}, której elementy są dane jako

$$A_{nm} = -\frac{\delta_{n-1,m}}{h^2} + \left(\frac{2}{h^2} + 4 - \frac{6}{\cosh^2 x_n}\right)\delta_{n,m} - \frac{\delta_{n+1,m}}{h^2}, \tag{10}$$

gdzie N duże $\sim 100 \ldots 1000$,

$$h = \frac{2L}{N-1} \qquad x_n = -L + nh, \qquad L = 10. \tag{11}$$

Cztery znalezione wektory proszę przedstawić w postaci graficznej - wykresu (x_n, y_n), gdzie y_n $n-$ta składowa wektora.

1 OPIS ZADANIA

Po wpisaniu naszych elementów do macierzy otrzymujemy typową macierz trójdiagonalną, z takimi samymi elementami nad jak i pod główną diagonalą:

główna diagonala: $b[n] = 2 * \left(\frac{N-1}{20}\right)^2 + 4 - \left(\frac{6}{\cosh^2\left(-10+\left(n*\frac{20}{N-1}\right)\right)}\right)$

górna / dola: $a[n] = b[n] = -\left(\frac{(N-1)}{20}\right)^2$

Dla macierzy o wielkości $N = 1000$ metoda potęgowa (zastosowana z zadania wcześniejszego) była nieoptymalna. Wyliczenie wartości własnych na moim komputerze zajmowało ~15minut czasu.

Jako iż w treści zadania nie było sprecyzowanej metody(metod), które mamy wykorzystać/zaimplementować, posłużyłem się tutaj specjalną funkcją z pakietu SciPy zwaną:

scipy.linalg.eigh_tridiagonal

Została specjalnie stworzona w celu znajdowania wartości jak i wektorów własnych macierzy symetrycznych, trójdiagonalnych. Także, funkcja ta idealnie nadaje się do naszej macierzy.

Uzyskane wyniki (4 najmniejsze wartości własne):

-7.6197348676182416e-05
2.9998524026464866
4.033942652976803
4.134776636606467

Uzyskane cztery najmniejsze wektory własne znajdują się w osobnym pliku.

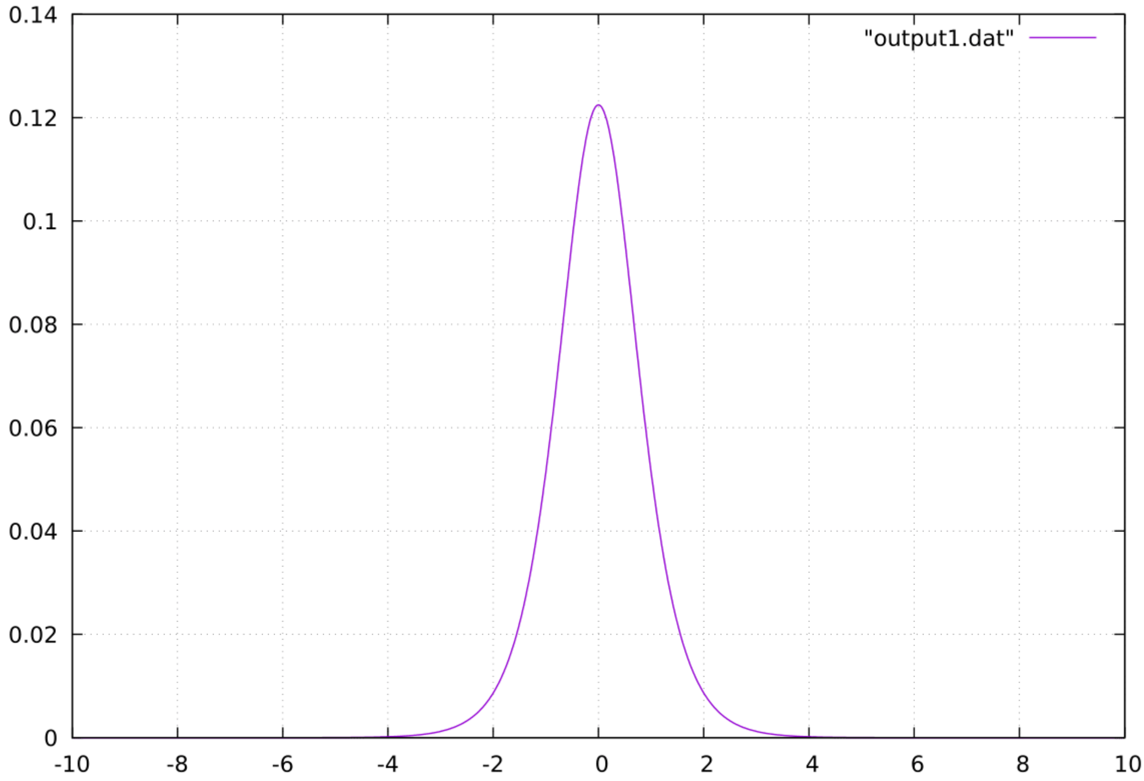
Wyniki zostały uzyskane z dokładnością: 10^{-8}

2 KOD PROGRAMU

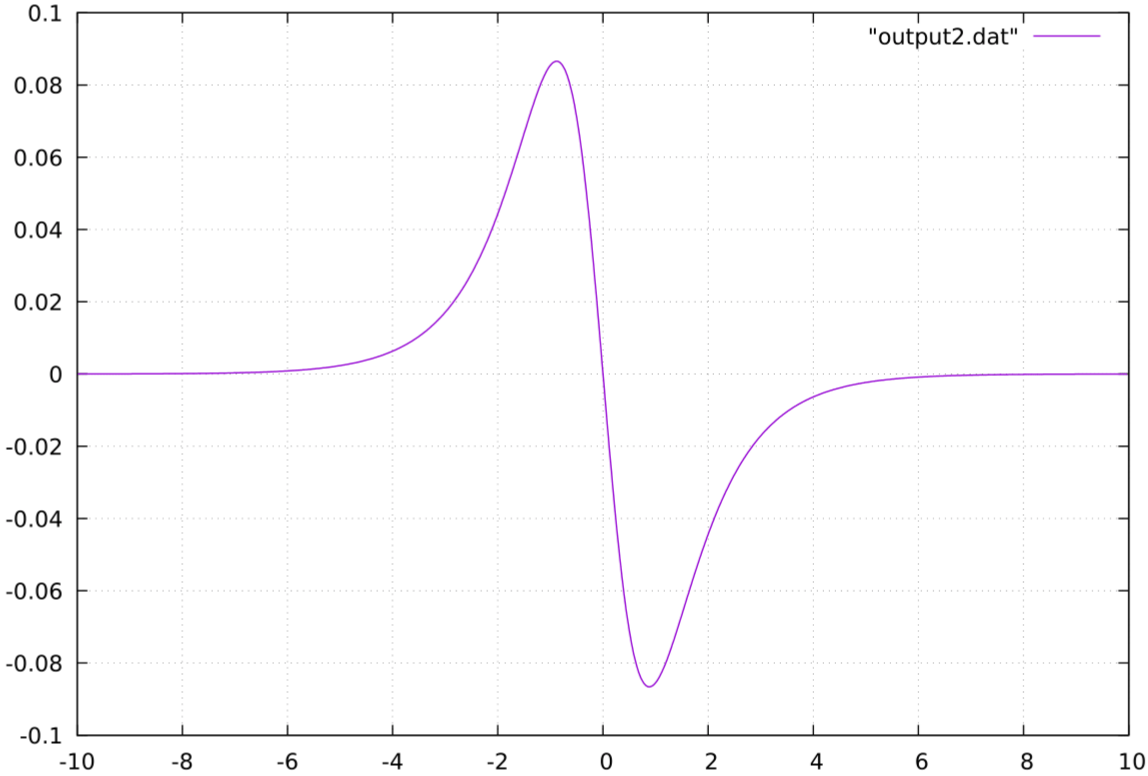
```python
import numpy as np
from scipy import linalg

e = 1e-8
L = 10.0
N = 1000
h = (2.0*L) / N
h_square = h * h

b = np.empty(N)
a = np.empty(N-1)

for i in range(0, N, 1):
    b[i] = ((2.0 / h_square) + 4 - (6.0 / np.cosh(-L + (i * h)) ** 2))

a.fill(-(1.0 / h_square))

e, v = linalg.eigh_tridiagonal(b, a, tol=e, lapack_driver='stebz')


for i in range(0, 4):
    print(e[i])

for i in range(0, 4):
    print(v[i])
```

3 WYKRESY

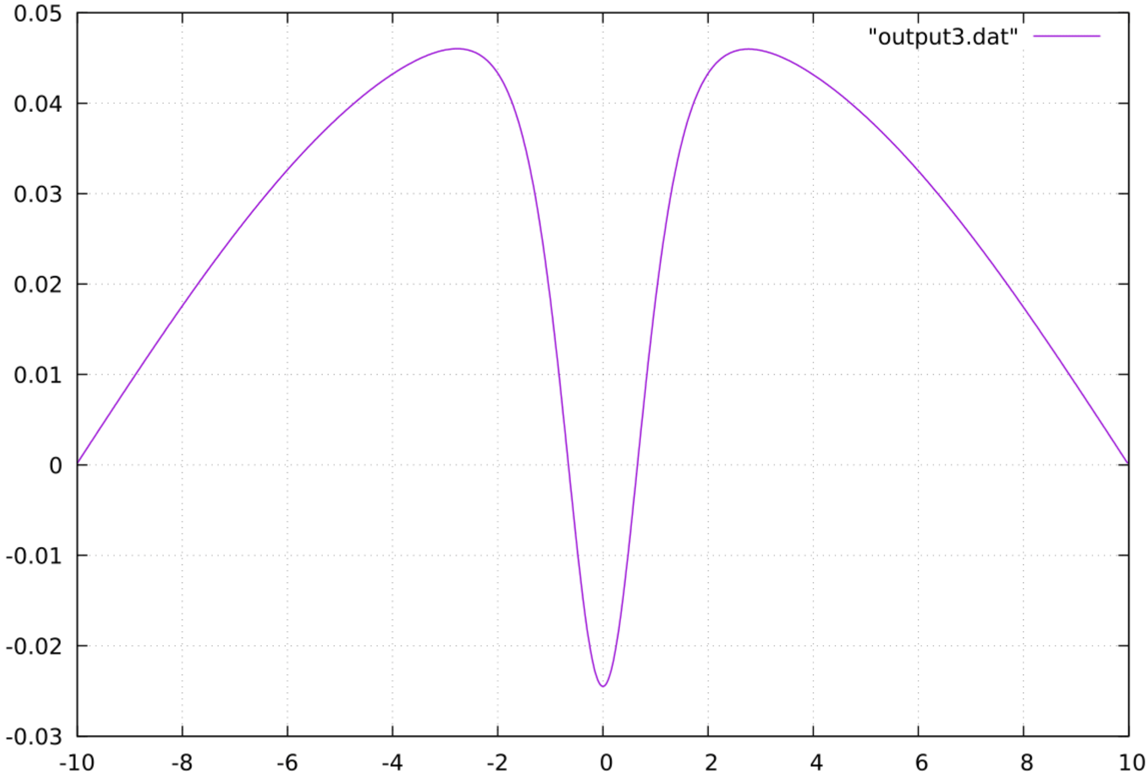
3.1 WYKRES WEKTORÓW WŁASNYCH DLA WARTOŚCI WŁASNEJ - 7.619734867618241E-05



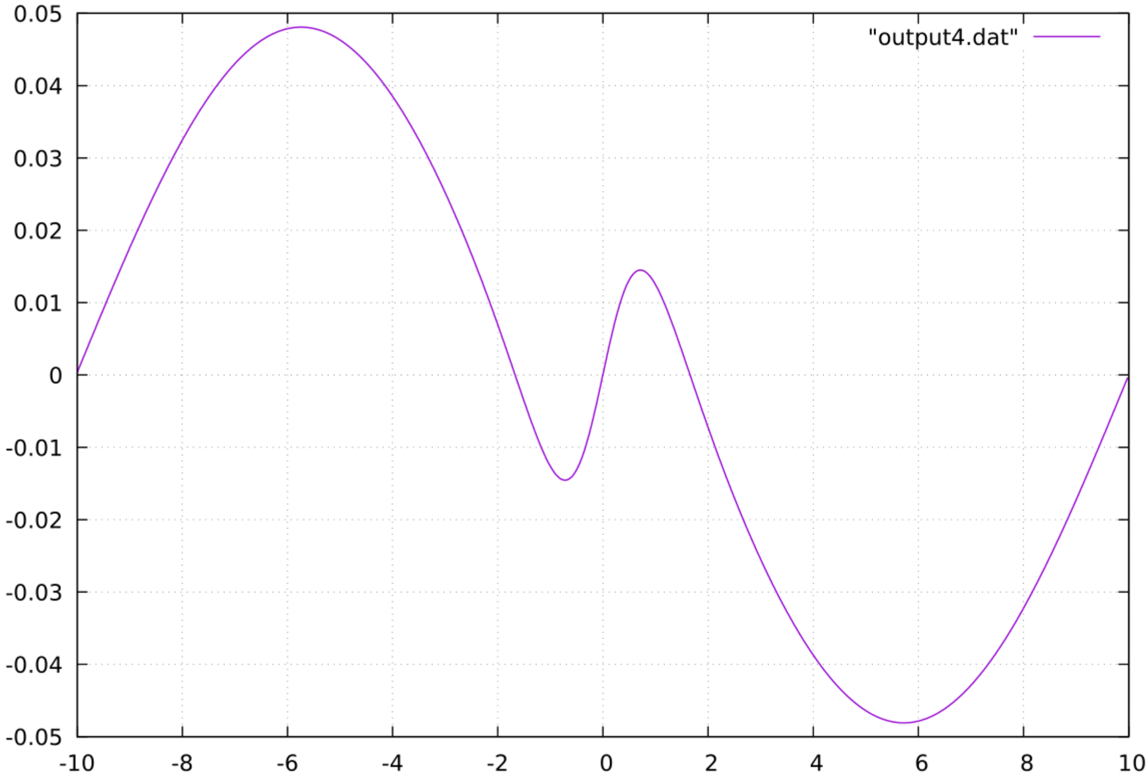
3.2 WYKRES WEKTORÓW WŁASNYCH DLA WARTOŚCI WŁASNEJ - 2.9998524026464866



3.3 WYKRES WEKTORÓW WŁASNYCH DLA WARTOŚCI WŁASNEJ - 4.033942652976803



3.4 WYKRES WEKTORÓW WŁASNYCH DLA WARTOŚCI WŁASNEJ - 4.134776636606467



4 ZAWARTOŚĆ

W zawartość paczki wchodzi:

1. Opis zadania
2. Plik programu
3. Wykresy poszczególnych wektorów własnych)
4. Plik z wektorami własnymi