Metody numeryczne

Mateusz Kwolek

8. Znajdź wartości funkcji

$$f(x) = \frac{1}{1 + 5x^2} \tag{6}$$

w punktach $-7/8, -5/8, -3/8, -1/8, 1/8, 3/8, 5/8, 7/8$ a następnie skonstruuj wielomian interpolacyjny Lagrange'a oparty na tych węzłach i wartościach funkcji (6) w tych węzłach. Narysuj wykres wielomianu interpolacyjnego. w przedziale $[-1.25, 1.25]$, zaznaczając na nim węzły i wartości w węzłach. Punkt dodatkowy: znajdź *współczynniki* wielomianu interpolacyjnego.

1) Kod źródłowy

Do stworzenia programu użyłem języka Python ze względu na jego prostotę, bogactwo bibliotek oraz częste zastosowanie do wykonywania obliczeń matematycznych.
Do wykonania wykresu tempa zbieżności metod użyłem biblioteki matplotlib.
Po uruchomieniu programu zostanie wyświetlone okno z wykresem gdzie poruszając się możemy dokładnie przyjrzeć się wynikom. Te zostaną również zapisane w pliku png.

```python
import numpy as np
import matplotlib.pyplot as plt
np.set_printoptions(suppress=True)

# Zadane punkty funkcji
x = np.array([-7/8, -5/8, -3/8, -1/8, 1/8, 3/8, 5/8, 7/8])

# Formowanie funkcji
def function(x):
    return 1 / (1 + 5 * x * x)

# Obliczenie wartości funkcji w punktach x
def calculate_y_array(x_array):
    y_array = []
    for x in x_array:
        y_array.append(function(x))

    return y_array

# Algorytm interpolacji wielomianowej Lagrange'a
def lagrange_interpolation(x, y):
    n = len(x)
    coefficients = np.zeros_like(x, dtype=float)

    for i in range(n):
        numerator = np.poly1d([1])
        denominator = 1

        for j in range(n):
            if j != i:
                numerator *= np.poly1d([1, -x[j]])
                denominator *= (x[i] - x[j])

        coefficients += (numerator / denominator) * y[i]

    return coefficients


y = calculate_y_array(x)
polynomial_coefficients = lagrange_interpolation(x, y)

# Wypisanie wyniku
print("Lagrangian interpolated polynomial:")
print(np.poly1d([round(coefficient, 4) for coefficient in polynomial_coefficients]))

# Konstrukcja wykresu
x_plt = np.linspace(min(x), max(x))
y_plt = np.polyval(polynomial_coefficients, x_plt)
plt.plot(x_plt, y_plt, color = "BLUE", label="Interpolation polynomial")
plt.scatter(x, y, color = "RED", label="Calculated points")
plt.xlabel("x")
plt.ylabel("y")
plt.legend()
plt.grid()

plt.savefig('wykres.png')
plt.show()
```

2) Wynik

```
Lagrangian interpolated polynomial:
        6     5         4         2
-4.775 x - 0 x + 7.221 x - 3.745 x + 0.9843
```

3) Wykres