Emiliya Rybak, Fizyka Medyczna, II rok

**Dane:**

Изображение выглядит как текст, снимок экрана, программное обеспечение, компьютер

Автоматически созданное описание

Użyj interpolacji wielomianowej, aby obliczyć i przy , używając danych

Изображение выглядит как текст, снимок экрана, программное обеспечение, компьютер

Автоматически созданное описание

Biorąc pod uwagę, że, zmierz dokładność wyniku.

**Rozwiązanie:**

1. **Najpierw użyjemy danych do interpolacji Lagrange'a i uzyskiwania wielomianu P(x).**

# Dane

x\_values = np.array([-2.2, -0.3, 0.8, 1.9])

f\_values = np.array([15.180, 10.962, 1.920, -2.040])

# Interpolacja Lagrange'a

def lagrange\_interpolation(x, x\_values, f\_values):

result = 0.0

n = len(x\_values)

for i in range(n):

term = f\_values[i]

for j in range(n):

if j != i:

term = term \* (x - x\_values[j]) / (x\_values[i] - x\_values[j])

result += term

return result

# Obliczanie wielomianu P(x)

f\_interpolated = lambda x: lagrange\_interpolation(x, x\_values, f\_values)

# Wyświetlanie wielomianu P(x)

print("Wielomian P(x) =", f\_interpolated)

1. **Teraz, aby obliczyć pochodne F′ F", skorzystajmy z funkcji np.polyderz biblioteki NumPy**

# Wielomian P(x)

coefficients = np.polyfit(x\_values, f\_values, len(x\_values) - 1)

P\_x = np.poly1d(coefficients)

# Obliczanie pochodnych

f\_prime = np.polyder(P\_x) # Pochodna pierwsza

f\_double\_prime = np.polyder(f\_prime) # Pochodna druga

# Wyświetlanie pochodnych

print("Pochodna pierwsza P'(x) =", f\_prime)

print("Pochodna druga P''(x) =", f\_double\_prime)

1. **Teraz, kiedy mamy wielomiany P(x), P′(x), P"(x), możemy ocenić ich wartości w x = 0.**

# Obliczanie pochodnych w punkcie x=0

f\_prime\_at\_0 = f\_prime(0)

f\_double\_prime\_at\_0 = f\_double\_prime(0)

# Wyświetlanie wyników

print("Pochodna pierwsza P'(0) =", f\_prime\_at\_0)

print("Pochodna druga P''(0) =", f\_double\_prime\_at\_0)

1. **Co do wniosków końcowych, można uzyskać uzyskane wartości pochodnych z teoretycznych pochodnych funkcji f ( x ). Obliczamy pochodne teorytyczne funkcji**

**i porównujemy wyniki**

# Definicja funkcji f(x)

def f(x):

return x\*\*3 - 0.3\*x\*\*2 - 8.56\*x + 8.448

# Pochodne analityczne

f\_prime\_analytical = lambda x: 3\*x\*\*2 - 0.6\*x - 8.56

f\_double\_prime\_analytical = lambda x: 6\*x - 0.6

# Obliczanie pochodnych analitycznych w punkcie x=0

f\_prime\_analytical\_at\_0 = f\_prime\_analytical(0)

f\_double\_prime\_analytical\_at\_0 = f\_double\_prime\_analytical(0)

# Wyświetlanie wyników

print("Analityczna pochodna pierwsza f'(0) =", f\_prime\_analytical\_at\_0)

print("Analityczna pochodna druga f''(0) =", f\_double\_prime\_analytical\_at\_0)

Wyniki uzyskane za pomocą interpolacji z wynikami analitycznymi muszą być zbliżone.   
Interpolacja wielomianowa Lagrange'a jest techniką używaną do wyszukiwania wielomianu, która przechodzi przez zbiór punktów danych. Jest to metoda wielomianowych konstrukcji interpolacyjnych, która idealnie nadaje się do elementów punktów danych.

**Taki to wynik uzyskujemy w tym zadaniu:**

Изображение выглядит как текст, программное обеспечение, снимок экрана, Мультимедийное программное обеспечение

Автоматически созданное описание