Artur Gęsiarz
Marzec 23, 2024

Laboratorium nr 4
MOwNiT – Efekt Rungego

1. Treść zadania

1.1. Wyznaczę wielomiany interpolujące funkcje

$f_1 = \frac{1}{1+25x^2}$, na przedziale [-1, 1]

$f_2 = \exp\,(\cos(x))$, na przedziale [0, 2π],

używając:

- wielomianów Lagrange'a z równoodległymi węzłami xj = x0 + jh, j = 0, 1, . . . , n, gdzie h = (xn − x0)/n

- kubicznych funkcji sklejanych z równoodległymi węzłami xj = x0+jh, j = 0, 1, . . . , n, gdzie h = (xn − x0)/n

- wielomianów Lagrange'a z węzłami Czebyszewa:

$$x_j = \cos(\theta_j), \qquad \theta_j = \frac{2j+1}{2(n+1)}\pi, \qquad 0 \le j \le n$$

(a) Dla funkcji Rungego, f1(x), z n = 12 węzłami interpolacji przedstawie na wspólnym wykresie funkcję f1(x) oraz wyznaczone wielomiany interpolacyjne i funkcję sklejaną. W celu stworzenia wykresu wykonam próbkowanie funkcji f1(x) i wielomianów interpolacyjnych na 10 razy gęstszym zbiorze (próbkowanie jednostajne w x dla węzłów równoodległych, jednostajne w θ dla węzłów Czebyszewa).

(b) Wykonam interpolację funkcji f1(x) i f2(x) z n = 4, 5, ..., 50 węzłami interpolacji, używając każdej z powyższych trzech metod interpolacji. Ewaluację funkcji, wielomianów interpolacyjnych oraz funkcji sklejanych przeprowadze na zbiorze 500 losowo wybranych punktów z dziedziny funkcji. Stwórze dwa rysunki, jeden dla f1(x), drugi dla f2(x). Na każdym rysunku przedstawie razem wykresy normy wektora błędów (czyli długości wektora) na tym zbiorze punktów w zależności od liczby węzłów interpolacji, n, dla każdej z trzech metod interpolacji.

2. Rozwiązanie zadania

2.1. Implementacja funkcji Rungego

```
def f1(x):
    return 1 / (1 + 25 * x ** 2)

def f2(x):
    return np.exp(np.cos(x))
```

2.2. Węzły interpolacji równoodległe

```
def equidistant_nodes(n, a, b):
    return np.linspace(a, b, n + 1)
```

2.3. Węzły interpolacji Czebyszewa

```
def chebyshev_nodes(n, a, b):
    theta = np.pi * (2 * np.arange(n + 1) + 1) / (2 * (n + 1))
    # return (a + b) / 2 + (b - a) / 2 * np.cos(theta)
    return a + (b - a) * (theta + 1) / 2
```

2.4. Metoda Lagrange'a

```
def lagrange_interpolation(x, nodes, values):
    result = 0
    n = len(nodes)
    for i in range(n):
        term = values[i]
        for j in range(n):
            if i != j:
                term *= (x - nodes[j]) / (nodes[i] - nodes[j])
        result += term
    return result
```

2.5. Próbkowanie funkcji

```
x_dense = np.linspace(-1, 1, 1000)
y_dense = f1(x_dense)
```

2.6. Liczba węzłów interpolacji

```
n = 12
```

2.7. Węzły interpolacji

```
nodes_equidistant = equidistant_nodes(n, -1, 1)
nodes_chebyshev = chebyshev_nodes(n, -1, 1)
```

2.8. Wartościami funkcji w węzłach interpolacji

```
values_equidistant = f1(nodes_equidistant)
values_chebyshev = f1(nodes_chebyshev)
```

2.9. Interpolacja Lagrange'a

```
y_interpolated_equidistant = lagrange_interpolation(x_dense, nodes_equidistant, values_equidistant)
y_interpolated_chebyshev = lagrange_interpolation(x_dense, nodes_chebyshev, values_chebyshev)
```

2.10. Sortowanie węzłów

```
sorted_indices_equidistant = np.argsort(nodes_equidistant)
sorted_indices_chebyshev = np.argsort(nodes_chebyshev)

nodes_equidistant_sorted = nodes_equidistant[sorted_indices_equidistant]
nodes_chebyshev_sorted = nodes_chebyshev[sorted_indices_chebyshev]

values_equidistant_sorted = values_equidistant[sorted_indices_equidistant]
values_chebyshev_sorted = values_chebyshev[sorted_indices_chebyshev]
```

2.11. Interpolacja funkcji sklejanych

```
cs_equidistant = CubicSpline(nodes_equidistant_sorted, values_equidistant_sorted)
cs_chebyshev = CubicSpline(nodes_chebyshev_sorted, values_chebyshev_sorted)
```

2.12. Wykres ze wszystkim

```
plt.figure(figsize=(10,6))
plt.plot(x_dense, y_dense, label='f_1(x)')
plt.plot(x_dense, y_interpolated_equidistant, label="Lagrange equidistant nodes")
plt.plot(x_dense, cs_equidistant(x_dense), label="Parallel glued node functions")
plt.plot(x_dense, y_interpolated_chebyshev, label="Lagrange Chebyshev nodes")
plt.plot(x_dense, cs_chebyshev(x_dense), label="Parallel glued node Chebyshev")
plt.xlabel('x')
plt.ylabel('y')
plt.title('Interpolation Rungeg function (f_1(x))')
plt.legend()
plt.grid(True)
plt.show()
```

2.13. Wykres porównujący f_1(x) z wielomianem interpolacyjnym Lagrange'a z równoodległymi węzłami

```
plt.figure(figsize=(10,6))
plt.plot(x_dense, y_dense, label='f_1(x)')
plt.plot(x_dense, y_interpolated_equidistant, label="Lagrange equidistant nodes")
plt.xlabel('x')
plt.ylabel('y')
plt.title('Interpolation Rungeg function (f_1(x)) and Lagrange with equidistant nodes')
plt.legend()
plt.grid(True)
plt.show()
```

2.14. Wykres porównujący f_1(x) z funkcją sklejoną z równoległymi węzłami

```
plt.figure(figsize=(10,6))
plt.plot(x_dense, y_dense, label='f_1(x)')
plt.plot(x_dense, cs_equidistant(x_dense), label="Parallel glued node functions")
plt.xlabel('x')
plt.ylabel('y')
plt.title('Interpolation Rungeg function (f_1(x)) with a glued function with parallel nodes')
plt.legend()
plt.grid(True)
plt.show()
```

2.15. Wykres porównujący f_1(x) z wielomianem interpolacyjnym Lagrange'a z węzłami Czebyszywa

```
plt.figure(figsize=(10,6))
plt.plot(x_dense, y_dense, label='f_1(x)')
plt.plot(x_dense, y_interpolated_chebyshev, label="Lagrange Chebyshev nodes")
plt.xlabel('x')
plt.ylabel('y')
plt.title('Interpolation Rungeg function (f_1(x)) with Lagrange interpolating polynomial with Chebyshev knots')
plt.legend()
plt.grid(True)
plt.show()
```

2.16. Wykres porównujący f_1(x) z funkcją sklejoną z węzłami Czebyszywa

```
plt.figure(figsize=(10,6))
plt.plot(x_dense, y_dense, label='f_1(x)')
plt.plot(x_dense, cs_chebyshev(x_dense), label="Parallel glued node Chebyshev")
plt.xlabel('x')
plt.ylabel('y')
plt.title('Interpolation Rungeg function (f_1(x)) with a glued Chebyshev function')
plt.legend()
plt.grid(True)
plt.show()
```

2.17. Funkcja obliczająca normę błędu

```
def error_norm(f_true, f_approx):
    return np.linalg.norm(f_true - f_approx)
```

2.18. Interpolacja i obliczanie błędów

```
def interpolate_and_cacluate_errors(f, nodes_generator, interpolation_method, num_nodes_range):
    errors = []
    for n in num_nodes_range:
        a, b = -1, 1
        nodes = nodes_generator(n, a, b)
        values = f(nodes)
        random_points = np.random.uniform(a, b, 500)
        true_values = f(random_points)
        interpolated_values = interpolation_method(random_points, nodes, values)
        errors.append(error_norm(true_values, interpolation_method(random_points, nodes, values)))
    return errors
```

2.19. Zakres liczby węzłow interpolacji

```
num_nodes_range = range(4,51)
```

2.20. Obliczanie błędów dla f_1(x)

```
errors_equidistant_f1 = interpolate_and_cacluate_errors(f1, equidistant_nodes, lagrange_interpolation,
 num_nodes_range)
errors_chebyshev_f1 = interpolate_and_cacluate_errors(f1, chebyshev_nodes, lagrange_interpolation,
 num_nodes_range)
```

2.21. Obliczanie błędow dla f_2(x)

```
errors_equidistant_f2 = interpolate_and_cacluate_errors(f2, equidistant_nodes, lagrange_interpolation,
 num_nodes_range)
errors_chebyshev_f2 = interpolate_and_cacluate_errors(f2, chebyshev_nodes, lagrange_interpolation,
 num_nodes_range)
```

2.22. Wykresy błędów

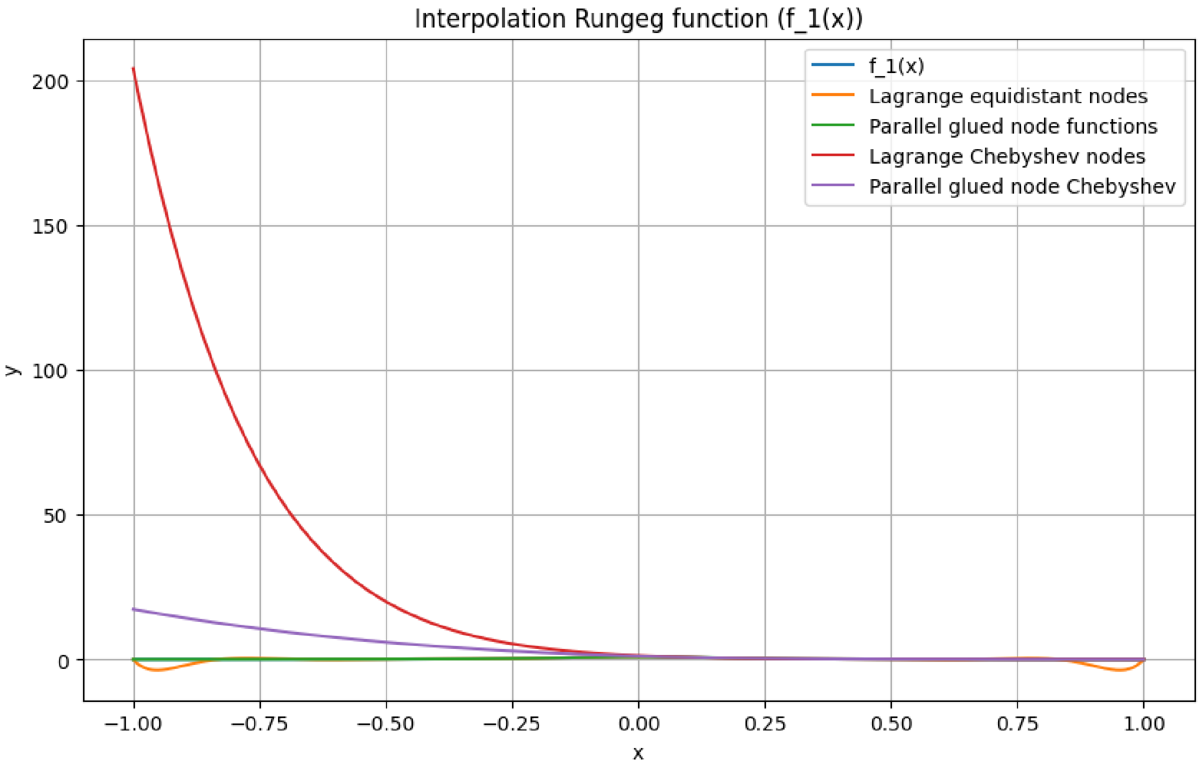
```
plt.figure(figsize=(10, 6))
plt.subplots_adjust(hspace=0.5)
plt.subplot(2, 1, 1)
plt.plot(num_nodes_range, errors_equidistant_f1, label="Parallel nodes", marker='o')
plt.plot(num_nodes_range, errors_chebyshev_f1, label="Chebyshev nodes", marker='o')
plt.title('Interpolation erros for f_1(x)')
plt.xlabel('Number interpolation nodes')
plt.ylabel('Error standard')
plt.legend()
plt.grid(True)

plt.subplot(2, 1, 2)
plt.plot(num_nodes_range, errors_equidistant_f2, label="Parallel nodes", marker='o')
plt.plot(num_nodes_range, errors_chebyshev_f2, label="Chebyshev nodes", marker='o')
plt.title('Interpolation erros for f_2(x)')
plt.xlabel('Number interpolation nodes')
plt.ylabel('Error standard')
plt.legend()
plt.grid(True)

plt.show()
```

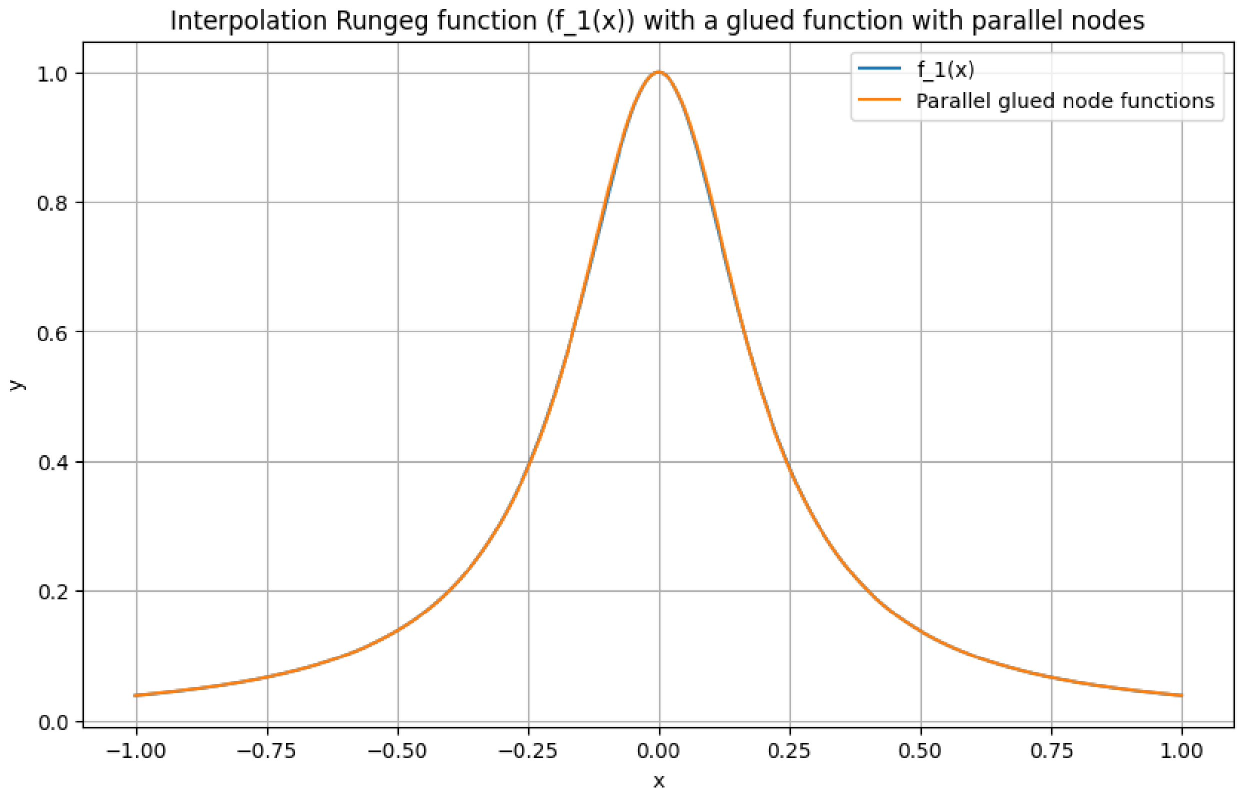
3. Wykresy

3.1. Funkcja Rungego razem z wielomianami interpolacji i funkcją sklejaną



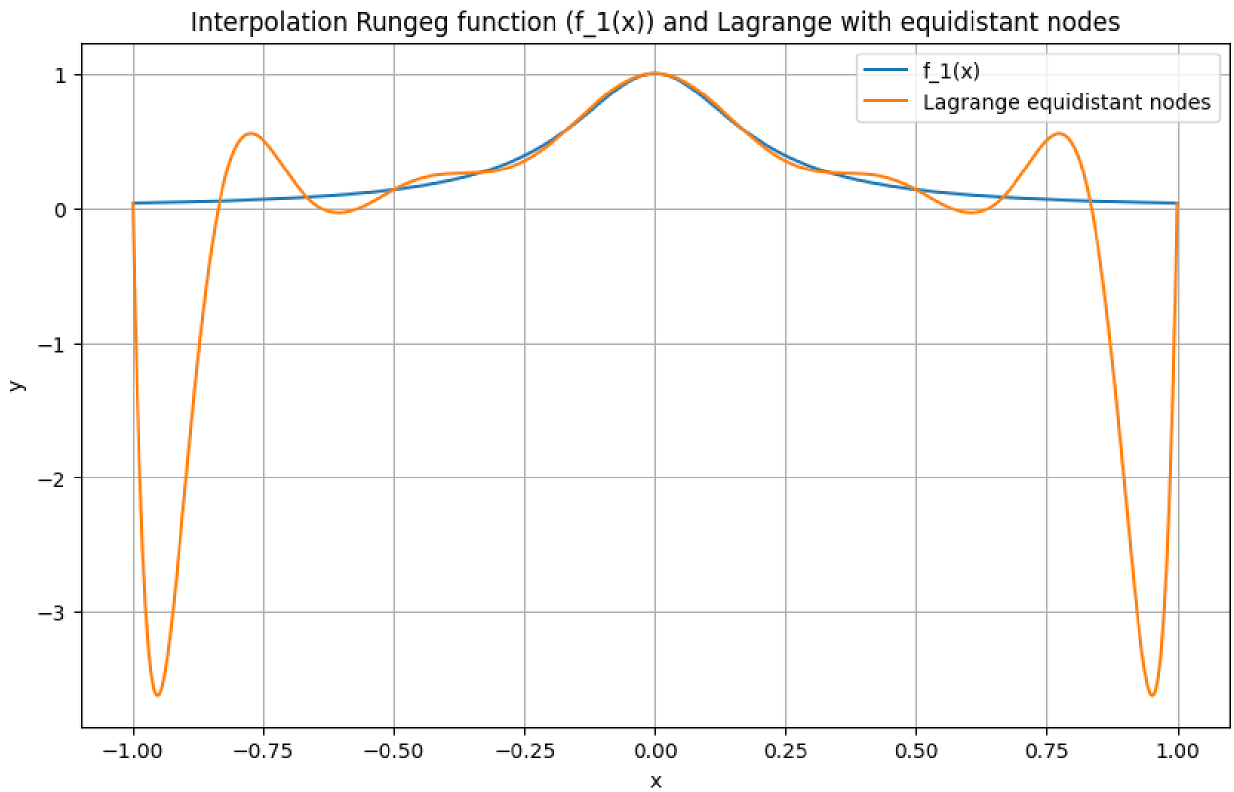
Wykres 1. Wykres przedstawia porównanie funkcji Rungego razem ze wszystkimi wielomianami

3.2. Funkcja Rungego razem z funkcja sklejaną z równoodległymi węzłami



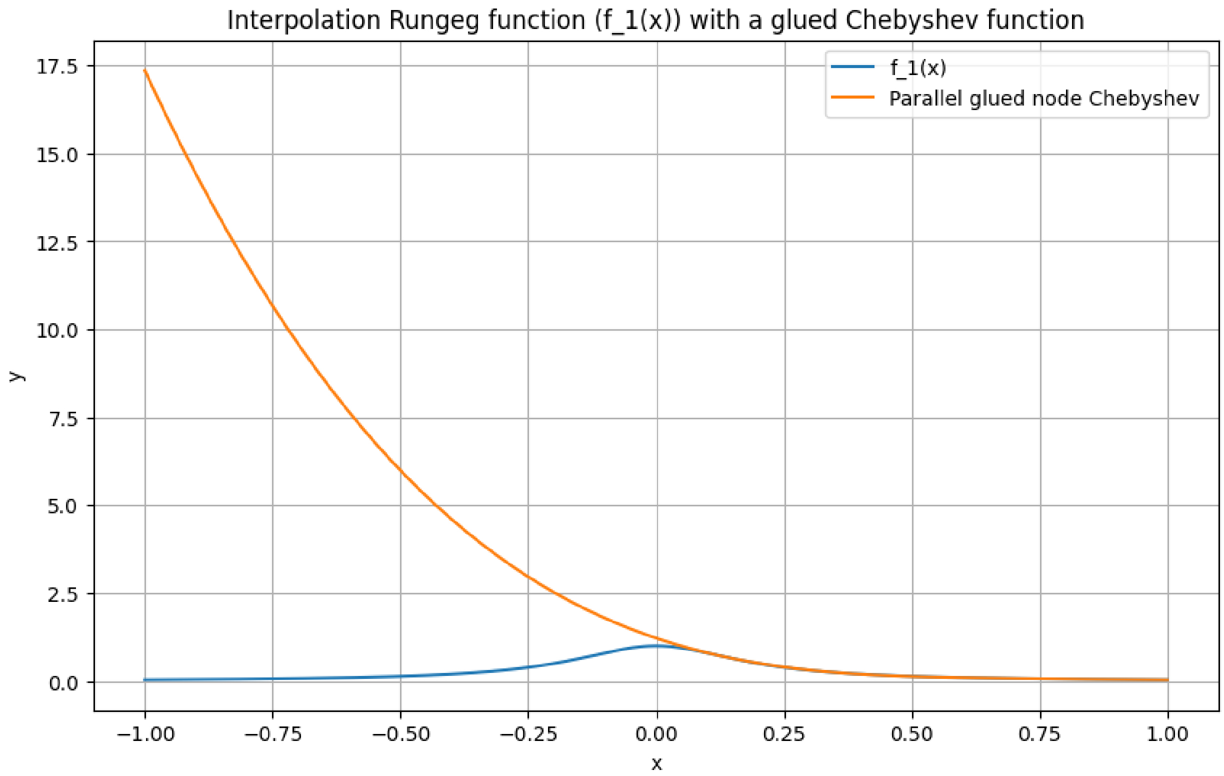
Wykres 2. Wykres przedstawia porównanie funkcji Rungego razem z funkcją sklejaną z równoodległymi węzłami

3.3. Funkcja Rungego razem z wielomianem Lagrange'a z węzłami równoodległymi



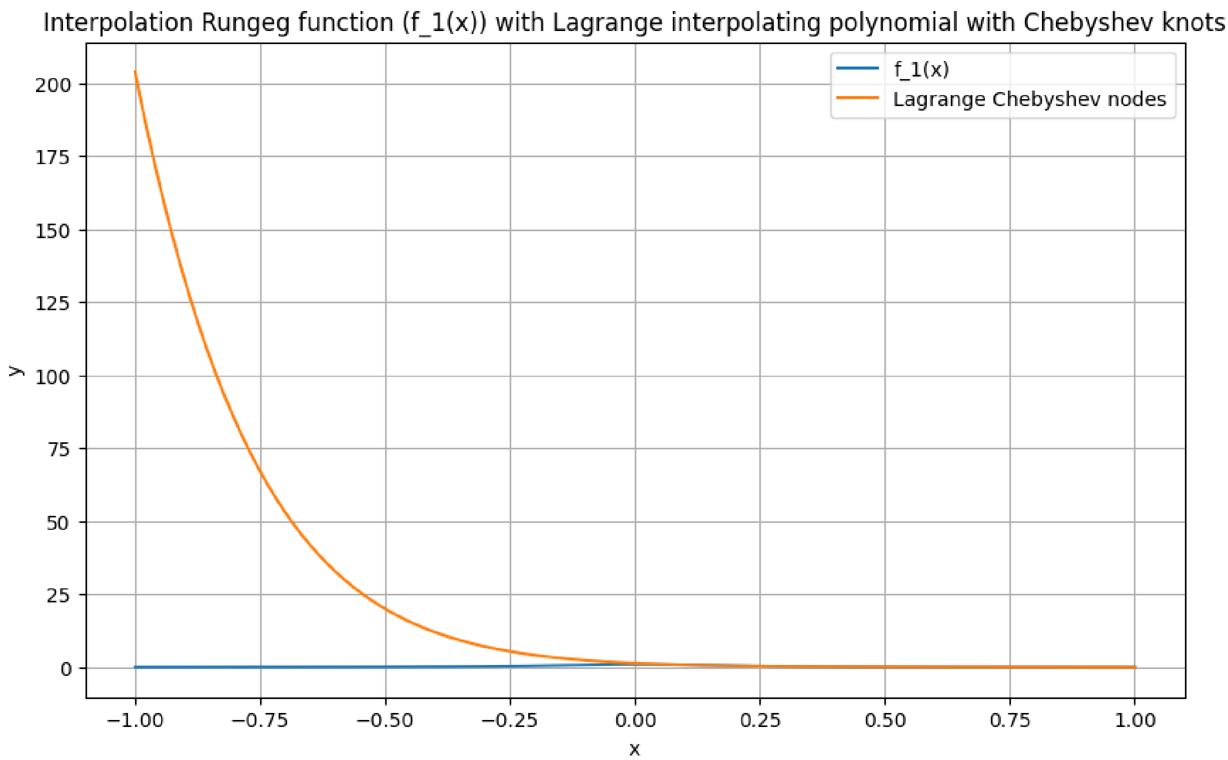
Wykres 3. Wykres przedstawia prównanie funkcji Rungego razem z wielomianem Lagrange'a z węzłami równodległymi

3.4. Funkcja Rungego razem z funkcja sklejaną z węzłami Czebyszewa



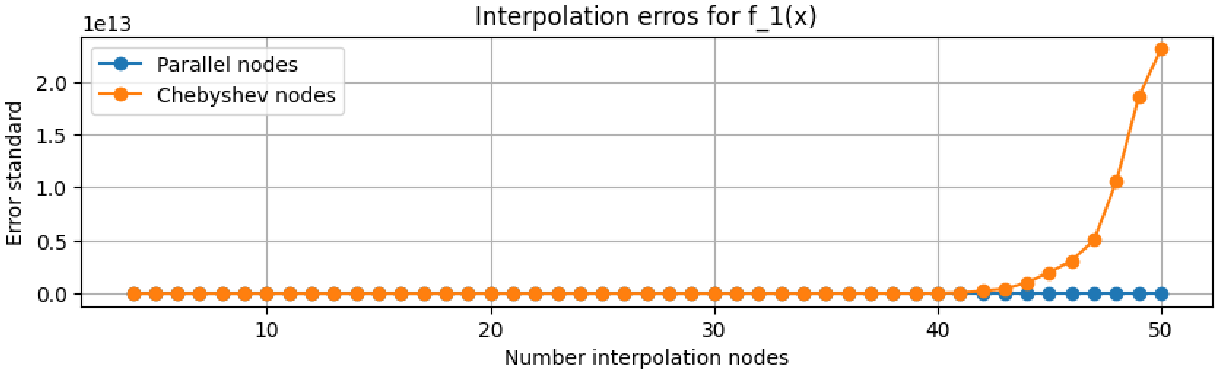
Wykres 4. Wykres przedstawia porównanie funkcji Rungego razem z funkcją sklejaną z węzłami Czebyszewa

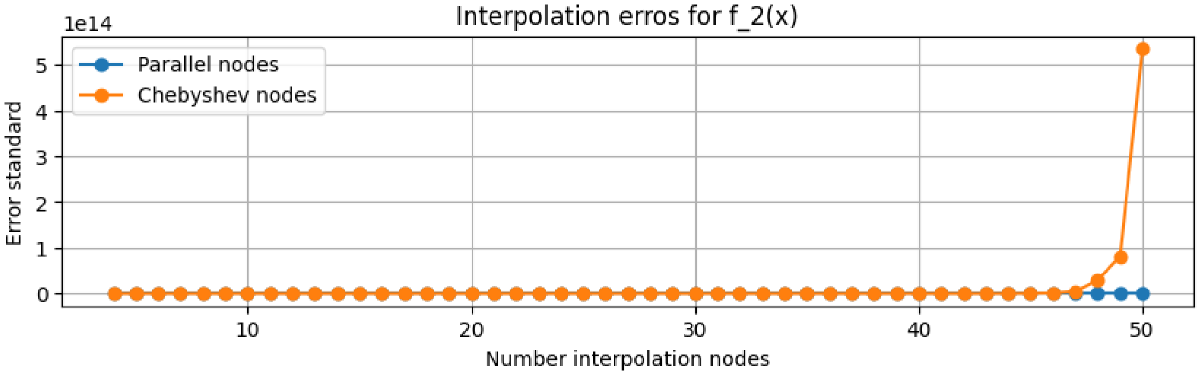
3.5. Funkcja Rungego razem z wielomianem Lagrange'a z węzłami Czebyszewa



Wykres 5. Wykres przedstawia prównanie funkcji Rungego razem z wielomianem Lagrange'a z węzłami Czebyszewa

3.6. Wykres normy wektora błędów





Wykres 6. Wykresy norma wektora błędów

4. Wnioski

Dla obu funkcji $f_1(x)\ i\ f_2(x)$, metoda funkcji sklejanych wydaje się dawać lepsze wyniki niż metoda interpolacji Lagrange'a. Wynika to z tego, że funkcje sklejane są w stanie dostosować się do bardziej złożonych kształtów funkcji, podczas gdy interpolacja Lagrange'a może prowadzić do efektu Rungego, szczególnie gdy używamy równoodległych węzłów interpolacji.

W przypadku obu funkcji $f_1(x)\ i\ f_2(x)$, równoodległe węzły interpolacji wydają się być równie skuteczne, a nawet lepsze niż węzły Czebyszewa. Równoodległe węzły pozwalają na równomierne próbkowanie funkcji, co prowadzi do dobrze zrównoważonych wyników interpolacji. Należy jednak zauważyć, że dla tych funkcji, wzrost błędu może wystąpić po przekroczeniu pewnej liczby węzłów interpolacji, szczególnie dla węzłów Czebyszewa w obu funkcjach. Dla $f_2(x)$ wzrost ten jest widoczny od około 48 węzła znacznie. Dla $f_1(x)$, także jest widoczny wzrost znacznie łagodniejszy od $f_2(x)$, ale za to zaczynający się do 40 węzła. Ogólnie rzecz biorąc do 40 węzła obje funkcje radzą sobie świetnie oraz obie metody.

Interpolacja z równoodległymi węzłami interpolacji może być równie skuteczna, a nawet lepsza niż interpolacja z węzłami Czebyszewa. Jest to szczególnie prawdziwe w przypadku funkcji o złożonym kształcie, gdzie równomierne próbkowanie funkcji jest kluczowe dla uzyskania dokładnych wyników interpolacji.

5. Bibliografia

Wykład MOwNiT - prowadzony przez dr. Inż. K. Rycerz
Prezentacje – dr. Inż. M. Kuta

6. Dodatkowe informacje

Rozwiązanie zadania znajduje się w pliku ex1.pynb