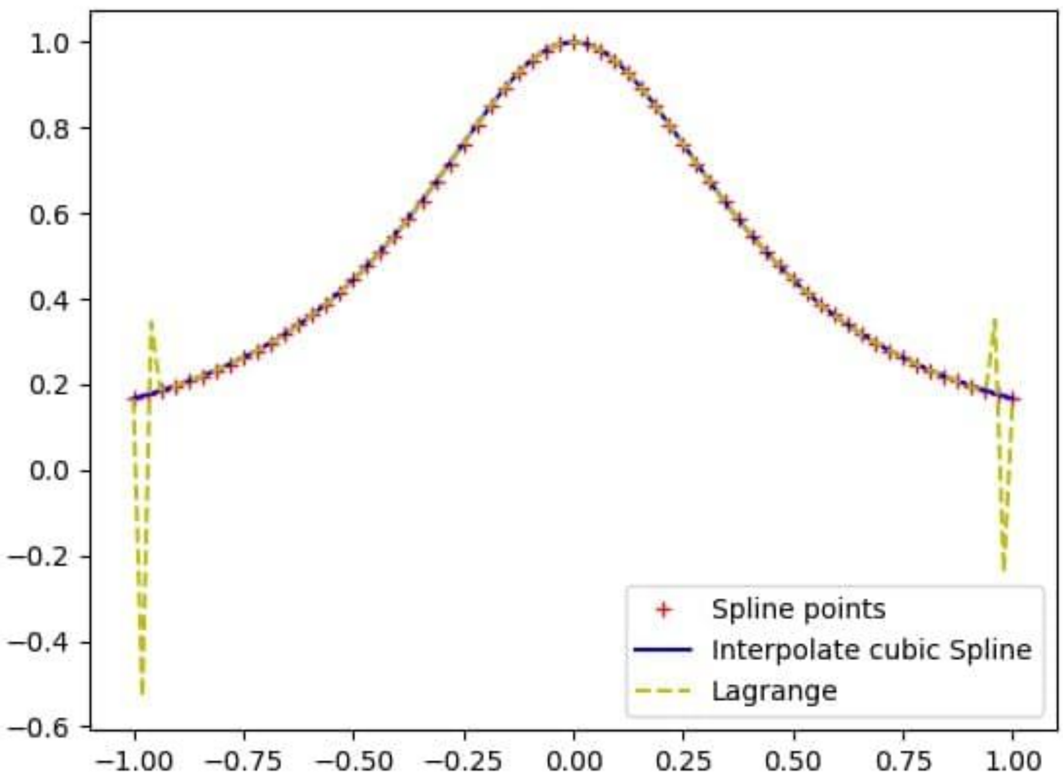
Program do zadania Nr.8:

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy import interpolate


def f(X):
    return 1 / (1 + 5 * (X ** 2))


def values():
    Yvalues = np.zeros(len(x))
    for k in range(len(x)):
        Yvalues[k] = f(x[k])
    return Yvalues


if '__main__' in __name__:
    x = np.arange(-1, 1.0001, 1 / 32)
    y = values()
    tck = interpolate.splrep(x, y, s=0)
    xnew = np.linspace(-1, 1, 65)
    ynew = interpolate.splev(xnew, tck, der=0)
    yplt = np.array([])

    for xp in xnew:
        yp = 0
        for xi, yi in zip(x, y):
            p = np.prod((xp - x[x != xi]) / (xi - x[x != xi]))
            yp += yi * p
        yplt = np.append(yplt, yp)

    plt.figure()
    plt.plot(x, y, 'r+', xnew, ynew, 'b-', xnew, yplt, 'y--')
    plt.legend(['Spline points', 'Interpolate cubic Spline', 'Lagrange'])
    plt.show()
    print(y)

```

Wyniki do zadania Nr.8: Wartości funkcji:

```
[0.16666667 0.17567336 0.18537292 0.19583094 0.20711974 0.21931891
 0.23251589 0.24680646 0.26229508 0.27909512 0.29732869 0.31712605
 0.33862434 0.36196536 0.38729198 0.41474281 0.44444444 0.4765007
 0.51097804 0.54788657 0.58715596 0.62860651 0.67191601 0.71658502
 0.76190476 0.80693459 0.85049834 0.89120975 0.92753623 0.95790458
 0.98084291 0.99514091 1.         0.99514091 0.98084291 0.95790458
 0.92753623 0.89120975 0.85049834 0.80693459 0.76190476 0.71658502
 0.67191601 0.62860651 0.58715596 0.54788657 0.51097804 0.4765007
 0.44444444 0.41474281 0.38729198 0.36196536 0.33862434 0.31712605
 0.29732869 0.27909512 0.26229508 0.24680646 0.23251589 0.21931891
 0.20711974 0.19583094 0.18537292 0.17567336 0.16666667]
```

Wykres:



Tutaj możemy zobaczyć Efekt Rungego, czyli pogorszenie jakości interpolacji wielomianowej, mimo zwiększenia liczby jej węzłów. (Wystarczy zmniejszyć ilosć danych do obliczeń)

Komentarze do zadania Nr.8:

W tym zadaniu było użyto interpolację Lagrange'a oraz naturalnego splajnu kubicznego.

Interpolacja Lagrange’a, zwaną także interpolacją wielomianową - metoda numeryczna przybliżania funkcji tzw. wielomianem Lagrange’a o stopnia n przyjmującym w n+1 punktach, zwanych węzłami interpolacji, wartości takie same jak przybliżana funkcja.

Wzór interpolacyjny Lagrange'a

$$N_k(x) = \prod_{\substack{i=0 \\ i \neq k}}^{n} \frac{(x - x_i)}{(x_k - x_i)} = \frac{(x - x_0)(x - x_1)\dots(x - x_{k-1})(x - x_{k+1})\dots(x - x_n)}{(x_k - x_0)(x_k - x_1)\dots(x_k - x_{k-1})(x_k - x_{k+1})\dots(x_k - x_n)} \quad (1)$$

Splajn naturalny.

Jest to taki splajn kubicznym który spełnia warunki postaci:

$$C''(x_0) = 0 \ , \ C''(x_{n-1}) = 0$$

Splajn naturalny posiada interesującą własność. Okazuje się, że spośród wszystkich funkcji ciągłych wraz z dwiema pierwszymi pochodnymi i interpolujących zadany układ węzłów splajn naturalny ma najmniejszą wartość całki z kwadratu drugiej pochodnej na przedziale interpolacji [a,b], tj.

$$\int_a^b \left[C''(x)\right]^2 dx = \min$$