

ИНТЕРПОЛИРОВАНИЕ ФУНКЦИЙ

In [1]:

```
import numpy as np
import math
import matplotlib.pyplot as plt
```

задаем функцию $f(x) = x - \sin(x) - 0.25$

In [2]:

```
def f(x):
    return (x - math.sin(x) - 0.25)
```

функция вычисляющая полином Лагранжа

$$L_n(x) = \sum_{i=0}^n \frac{\omega_{n+1}(x)}{(x-x_i)\omega'_{n+1}(x_i)} f(x_i)$$

In [3]:

```
def lagrange(x):
    def l(k):
        w = 1
        for i in range(n):
            if i != k:
                w *= (x - xx[i])
        w_ = 1
        for i in range(n):
            if i != k:
                w_ *= (xx[k] - xx[i])
        return w / w_

    res = 0
    for i in range(n):
        res += yy[i] * l(i)
    return res
```

Берем равностоящие узлы на промежутке $[-5, 5]$

In [4]:

```
a = -5
b = 5
n = 10
xx = np.linspace(a, b, n)
yy = np.array(list(map(f, xx)))
xch = np.zeros(n)
```

Вычисляем узлы по формуле

$$x_i = \frac{1}{2} \left[(b - a) \cos \frac{(2i+1)}{2(n+1)} \pi + (b + a) \right], i = \overline{1, n}$$

In [5]:

```
for i in range(n):
    xch[i] = 0.5 * ((b - a) * math.cos(
        (2 * i + 1) / (2 * n) * math.pi) + (b + a))
```

Методическая погрешность

In [6]:

```
xg = np.arange(a, b + 0.1, 0.1)
maxr = 0
for cur in xch:
    maxr = max(maxr, abs(f(cur) - lagrange(cur)))
print(maxr)
```

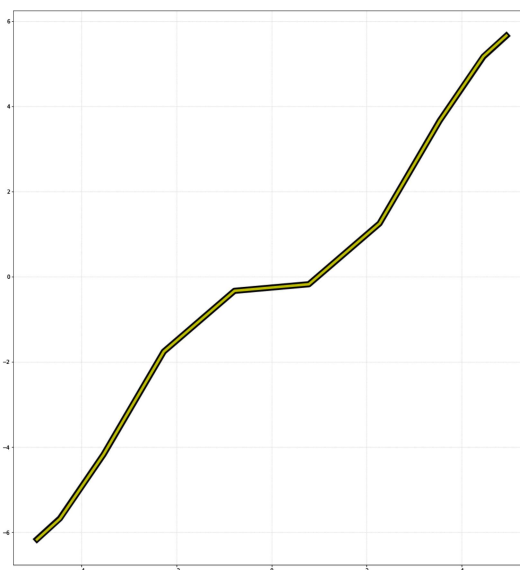
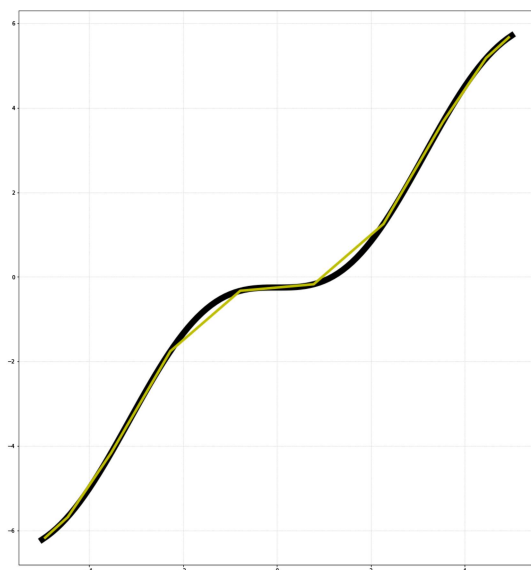
0.006936596977038967

Рисуем график исходного полинома поверх полученного полинома Лагранжа

слева график построенный по равноотстоящим узлам, справа по определенных по формуле

In [7]:

```
plt.figure(figsize=(40, 20))
ych = np.array(list(map(f, xch)))
plt.subplot (1, 2, 2)
plt.plot(xch, ych, linewidth=12, color='k')
ych = np.array(list(map(lagrange, xch)))
plt.grid(ls=':')
plt.plot(xch, ych, linewidth=5, color='y')
yg = np.array(list(map(f, xg)))
plt.subplot (1, 2, 1)
plt.plot(xg, yg, linewidth=12, color='k')
yg = np.array(list(map(lagrange, xg)))
plt.grid(ls=':')
plt.plot(xch, ych, linewidth=5, color='y')
plt.show()
```



Увеличиваем количество узлов

In [8]:

```
a = -5
b = 5
n = 24
xx = np.linspace(a, b, n)
yy = np.array(list(map(f, xx)))
xch = np.zeros(n)
```

Вычисляем узлы по формуле

$$x_i = \frac{1}{2} \left[(b - a) \cos \frac{(2i+1)}{2(n+1)} \pi + (b + a) \right], i = \overline{1, n}$$

In [9]:

```
for i in range(n):
    xch[i] = 0.5 * ((b - a) * math.cos(
        (2 * i + 1) / (2 * n) * math.pi) + (b + a))
```

Методическая погрешность

In [10]:

```
xg = np.arange(a, b + 0.1, 0.1)
maxr = 0
for cur in xch:
    maxr = max(maxr, abs(f(cur) - lagrange(cur)))
print(maxr)
```

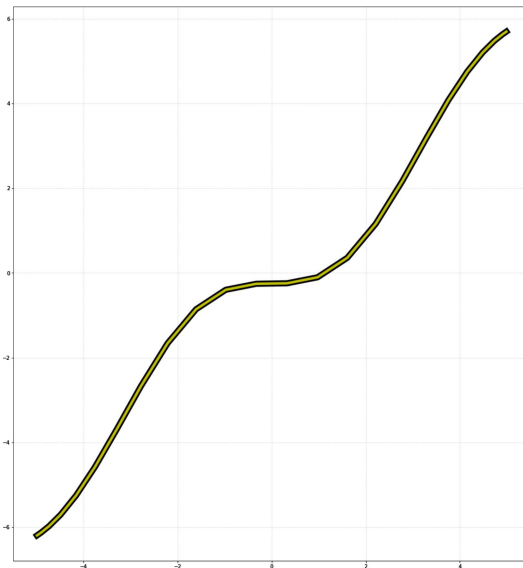
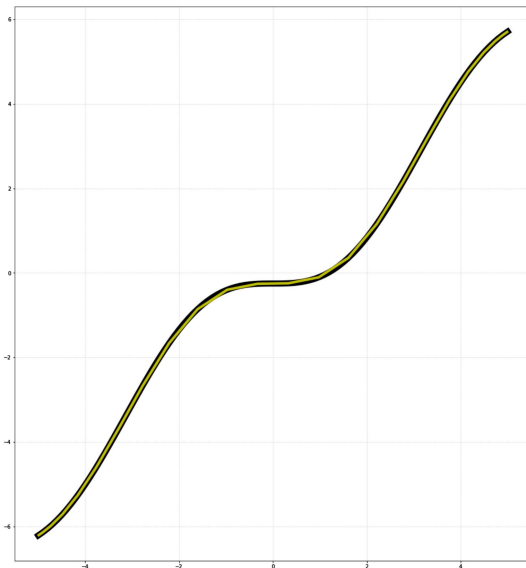
7.561951065326866e-12

Рисуем график исходного полинома поверх полученного полинома Лагранжа

слева график построенный по равноотстоящим узлам, справа по определенных по формуле

In [11]:

```
plt.figure(figsize=(40, 20))
ych = np.array(list(map(f, xch)))
plt.subplot (1, 2, 2)
plt.plot(xch, ych, linewidth=12, color='k')
ych = np.array(list(map(lagrange, xch)))
plt.grid(ls=':')
plt.plot(xch, ych, linewidth=5, color='y')
yg = np.array(list(map(f, xg)))
plt.subplot (1, 2, 1)
plt.plot(xg, yg, linewidth=12, color='k')
yg = np.array(list(map(lagrange, xg)))
plt.grid(ls=':')
plt.plot(xch, ych, linewidth=5, color='y')
plt.show()
```



Повторяем тоже самое, но для функции $h(x)$

задаем функцию $h(x) = |x| * (x - \sin(x) - 0.25)$

In [12]:

```
def h(x):
    return abs(x) * (x - math.sin(x) - 0.25)
```

Берем равностоящие узлы на промежутке $[-5, 5]$

In [13]:

```
a = -5
b = 5
n = 10
xx = np.linspace(a, b, n)
yy = np.array(list(map(h, xx)))
xch = np.zeros(n)
```

Вычисляем узлы по формуле

$$x_i = \frac{1}{2} \left[(b - a) \cos \frac{(2i+1)}{2(n+1)} \pi + (b + a) \right], i = \overline{1, n}$$

In [14]:

```
for i in range(n):
    xch[i] = 0.5 * ((b - a) * math.cos(
        (2 * i + 1) / (2 * n) * math.pi) + (b + a))
```

Методическая погрешность

In [15]:

```
xg = np.arange(a, b + 0.1, 0.1)
maxr = 0
for cur in xch:
    maxr = max(maxr, abs(h(cur) - lagrange(cur)))
print(maxr)
```

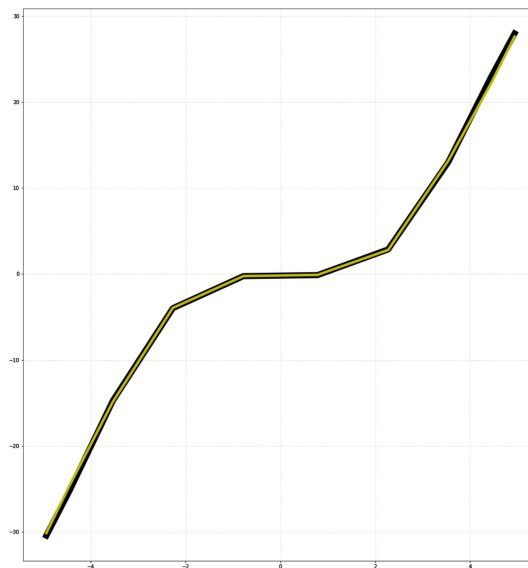
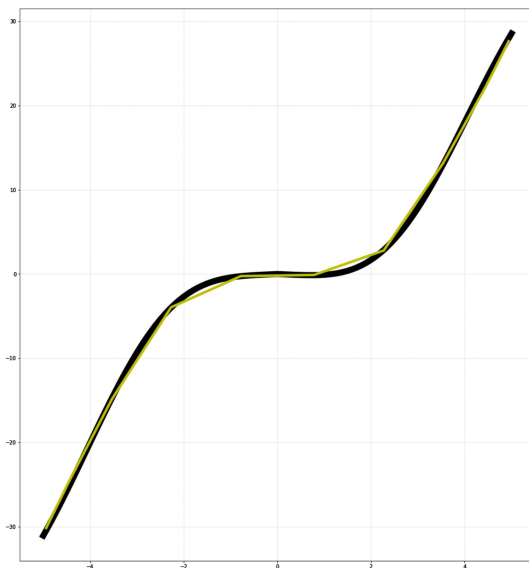
0.7342042694681936

Рисуем график исходного полинома поверх полученного полинома Лагранжа

слева график построенный по равноотстоящим узлам, справа по определенных по формуле

In [16]:

```
plt.figure(figsize=(40, 20))
ych = np.array(list(map(h, xch)))
plt.subplot(1, 2, 2)
plt.plot(xch, ych, linewidth=12, color='k')
ych = np.array(list(map(lagrange, xch)))
plt.grid(ls=':')
plt.plot(xch, ych, linewidth=5, color='y')
yg = np.array(list(map(h, xg)))
plt.subplot(1, 2, 1)
plt.plot(xg, yg, linewidth=12, color='k')
yg = np.array(list(map(lagrange, xg)))
plt.grid(ls=':')
plt.plot(xch, ych, linewidth=5, color='y')
plt.show()
```



Увеличиваем количество узлов

In [17]:

```
a = -5
b = 5
n = 24
xx = np.linspace(a, b, n)
yy = np.array(list(map(h, xx)))
xch = np.zeros(n)
```

Вычисляем узлы по формуле

$$x_i = \frac{1}{2} \left[(b - a) \cos \frac{(2i+1)}{2(n+1)} \pi + (b + a) \right], i = \overline{1, n}$$

In [18]:

```
for i in range(n):
    xch[i] = 0.5 * ((b - a) * math.cos(
        (2 * i + 1) / (2 * n) * math.pi) + (b + a))
```

Методическая погрешность

In [19]:

```
xg = np.arange(a, b + 0.1, 0.1)
maxr = 0
for cur in xch:
    maxr = max(maxr, abs(f(cur) - lagrange(cur)))
print(maxr)
```

80.52926308099192

Рисуем график исходного полинома поверх полученного полинома Лагранжа

слева график построенный по равноотстоящим узлам, справа по определенным по формуле

In [20]:

```
plt.figure(figsize=(40, 20))
ych = np.array(list(map(h, xch)))
plt.subplot(1, 2, 2)
plt.plot(xch, ych, linewidth=12, color='k')
ych = np.array(list(map(lagrange, xch)))
plt.grid(ls=':')
plt.plot(xch, ych, linewidth=5, color='y')
yg = np.array(list(map(h, xg)))
plt.subplot(1, 2, 1)
plt.plot(xg, yg, linewidth=12, color='k')
yg = np.array(list(map(lagrange, xg)))
plt.grid(ls=':')
plt.plot(xch, ych, linewidth=5, color='y')
plt.show()
```

