Задание 3. Улучшать степень полинома, путем подбора коэффициентов полиномов.

$$f(n) = a_n x^n + \cdots + a_1 x + a_0$$

Входные данные:

- множество упорядоченных точек (x_k, y_k) по оси абцисса;
- точность.

Выходные данные:

- оптимальный степень полинома;
- ошибка приближенного полинома.

Решение

❖ **<u>Алгоритм</u>**

Обычным методом находим коэффициенты $A = [a_0 \;\; a_1 \dots a_n]$.

Используя метод наименьших квадратов:

$$\phi(A) = \sum_i \big(y_i - f(x_i)\big)^2 \to \; min$$

$$\frac{\partial \phi}{\partial a_k} = -2 \sum_i \big(y_i - f(x_i)\big) x_i^k = 0, \qquad \forall k = \overline{0, n}$$

Полученная система уравнений можно записывается в следующим матричном виде

$$MA = B$$

$$M = \begin{pmatrix} n & \cdots & \sum_i x_i^n \\ \vdots & \ddots & \vdots \\ \sum_i x_i^n & \cdots & \sum_i x_i^{2n} \end{pmatrix}$$

$$B = \left(\sum_i y_i \quad \sum_i y_i x_i \quad \dots \sum_i y_i x_i^n \right)^T$$

Решением этого уравнения является: $A = M^{-1} B$

❖ **Ошибка приближенного полинома**

Ошибка приближенного полинома вычисляется по следующим формулам:

Формула №1	**Формула №2**
$err(n) = \frac{\sum_i \lvert y_i - f(z_i, A) \rvert}{n \cdot \sigma_Y}$	$err(n) = \frac{\sqrt{\frac{1}{n}\sum_i \left(y_i - f(x_i)\right)^2}}{\sigma_Y}$

Здесь σ_Y − среднее квадратическое Y.

$$\sigma_Y = \sqrt{\frac{\sum_i y_i^2}{n}}$$

❖ **Код программы** (*Python*)

В данной программе были реализованы две библиотеки функций:

1. *ols.py* – библиотека, реализующая метод наименьших квадратов и вычислить ошибку;
2. *bplotlib.py* – библиотека для реализации графиков.

- **Эксперименты**

1) Входные данные (21 точки)
   - $X = 0{,}0.05..1$
   - $Y -$ изменяется
2) Замечание на рисуках:
   - Линия синяя – данная;
   - Линия оранжевая – аппроксимированная;
3) **Выводы:**

- Чем выше точность, тем ближе графики, т.е. качество аппроксимации высоче;
- Однако при повышении точности оптимальный степень полином также повысит;
- Всегда можно улучшать степень полином (целые), для обычных видов функций степень приближенного полинома снижается почти $2 - 2.5$ раза меньше у исходной!

4) **Замечание**

В дальнейших работах будем рассматривать на задаче улучшения степень полином при вещественных показателях.

1) **<u>Экс. №1</u>**

Y = [30000, 29989, 29540, 29205, 28956, 28403, 27899, 27181, 26000, 23305, 18990, 14392, 9530, 5900, 3593, 2999, 2512, 2311, 2012, 2005, 2000]

Точность	Оптимальный степень полинома	<u>Ошибка</u>
0.1	3	0.070871
0.01	9	0.008108

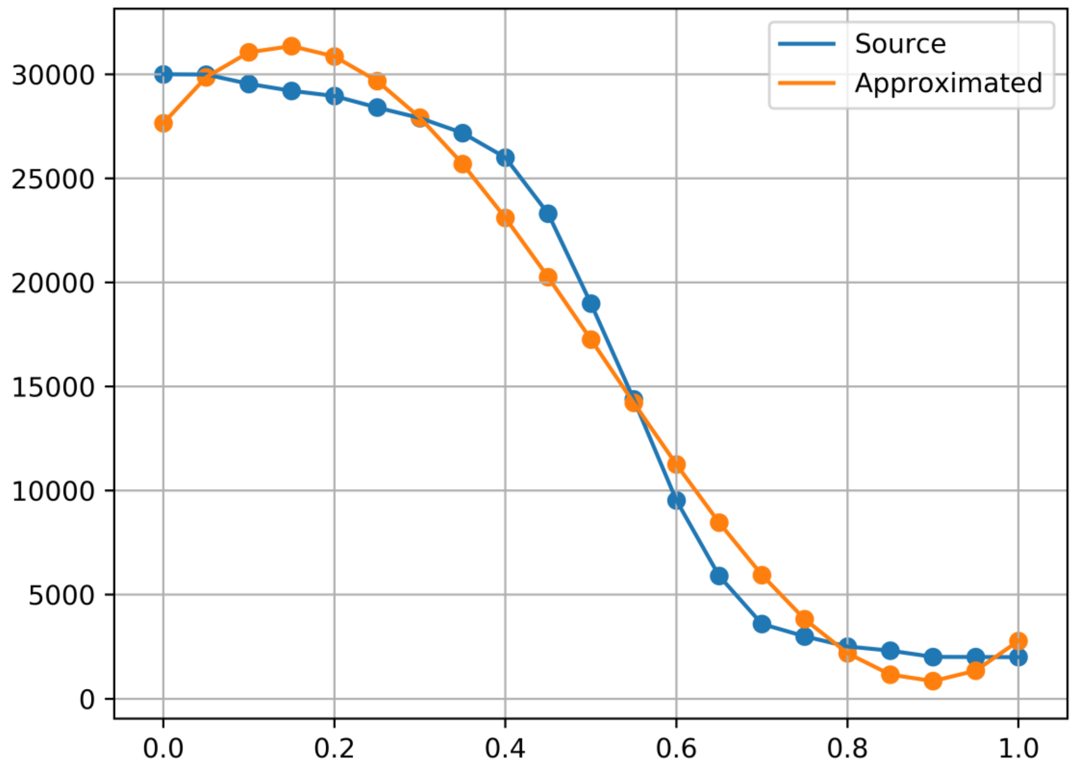

Рис. 1.1 Приближенный полином при точности 0.1

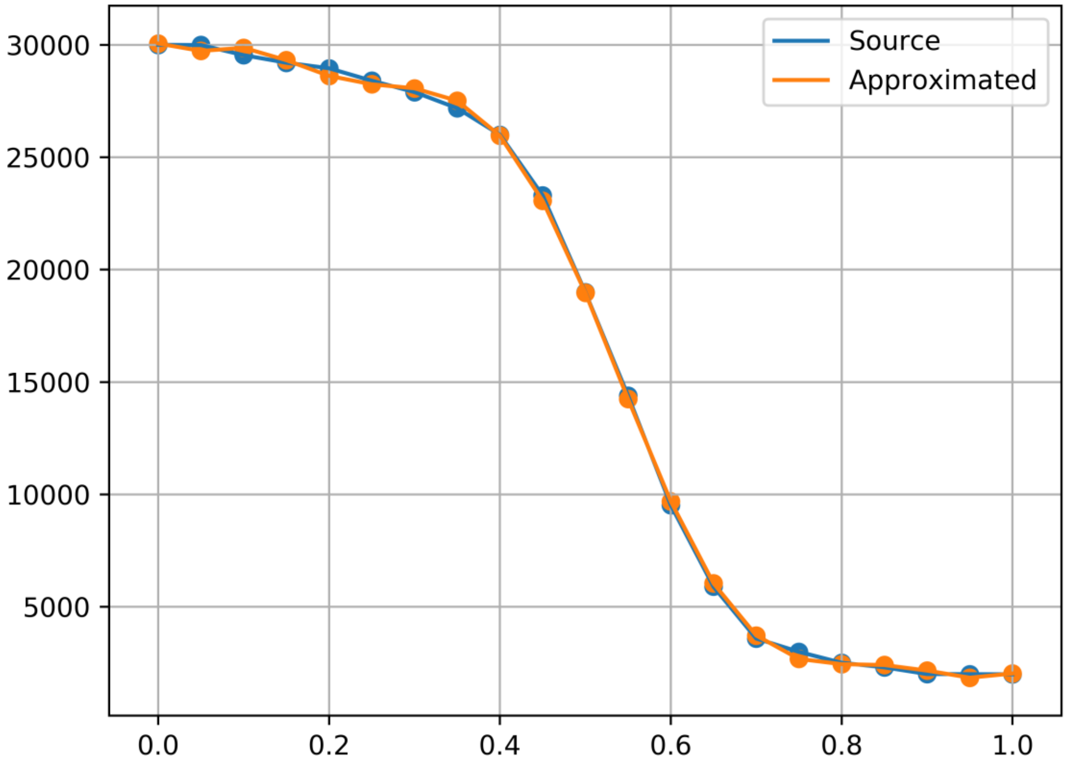

Рис. 1.2 Приближенный полином при точности 0.01

Вывод: качество аппроксимации очень хорошо получается при точности до 2 знака!

2) <u>Экс. №2</u>

Y = [30000, 29999, 29996, 29989, 29981, 29970, 29957, 29939, 29917, 29875, 29839, 29777, 29403, 29001, 28101, 25112, 21052, 13917, 7912, 4371, 3000]

Точность	Оптимальный степень полинома	<u>Ошибка</u>
0.1	2	0.083734
0.01	7	0.009463

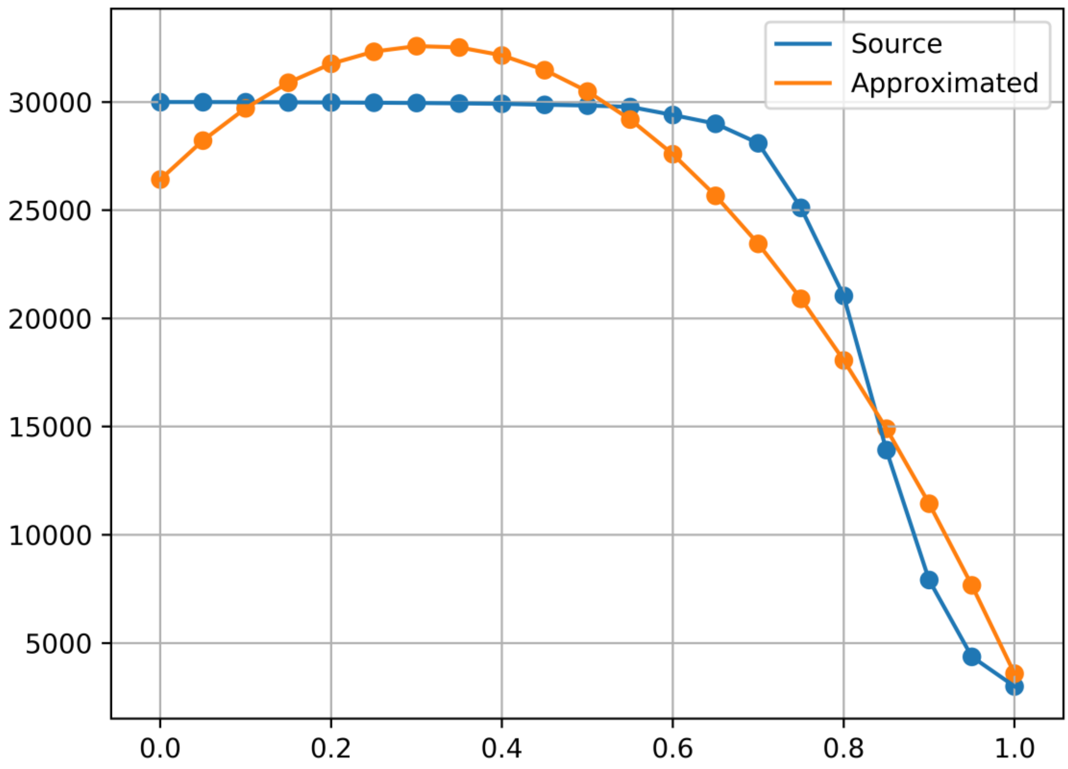

Рис. 2.1 Приближенный полином при точности 0.1

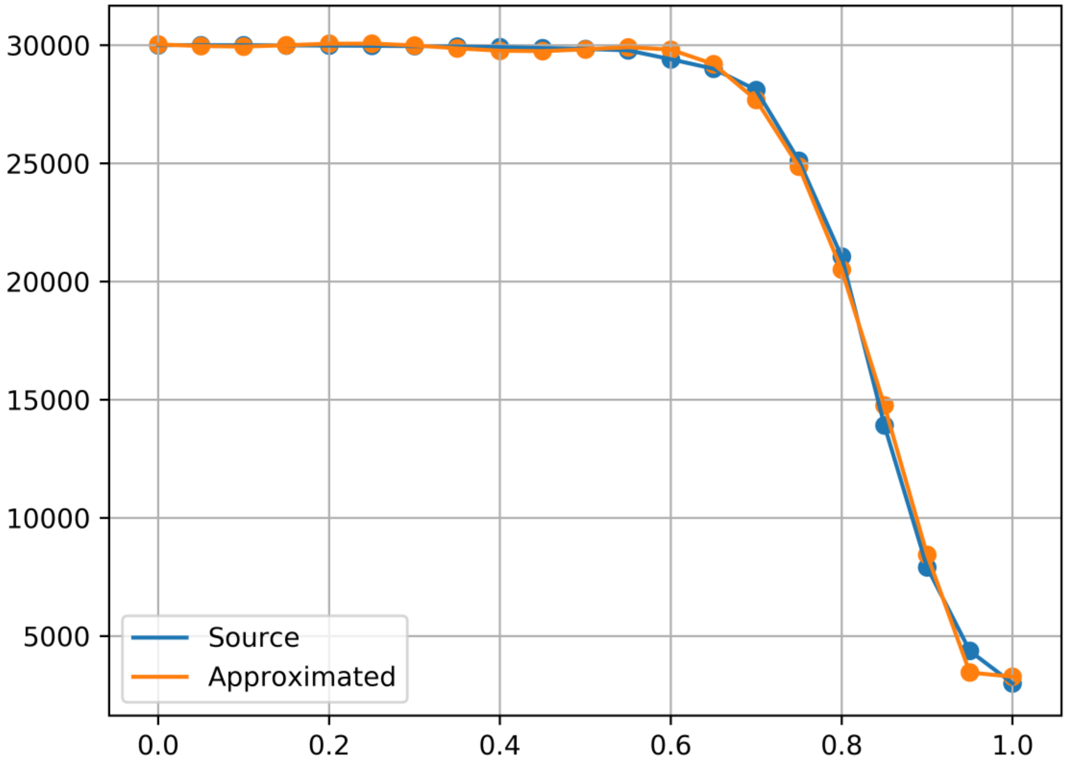

Рис. 2.2 Приближенный полином при точности 0.01

Вывод: качество аппроксимации очень хорошо получается при точности до 2 знака!

3) <u>Экс. №3</u>

Данный эксперимент реализуется по точной формуле вида:

$$y = b_0 + b_1 x^{10}$$

Точность	Оптимальный степень полинома	<u>Ошибка</u>
0.1	2	0.0907197
0.01	5	0.00624241

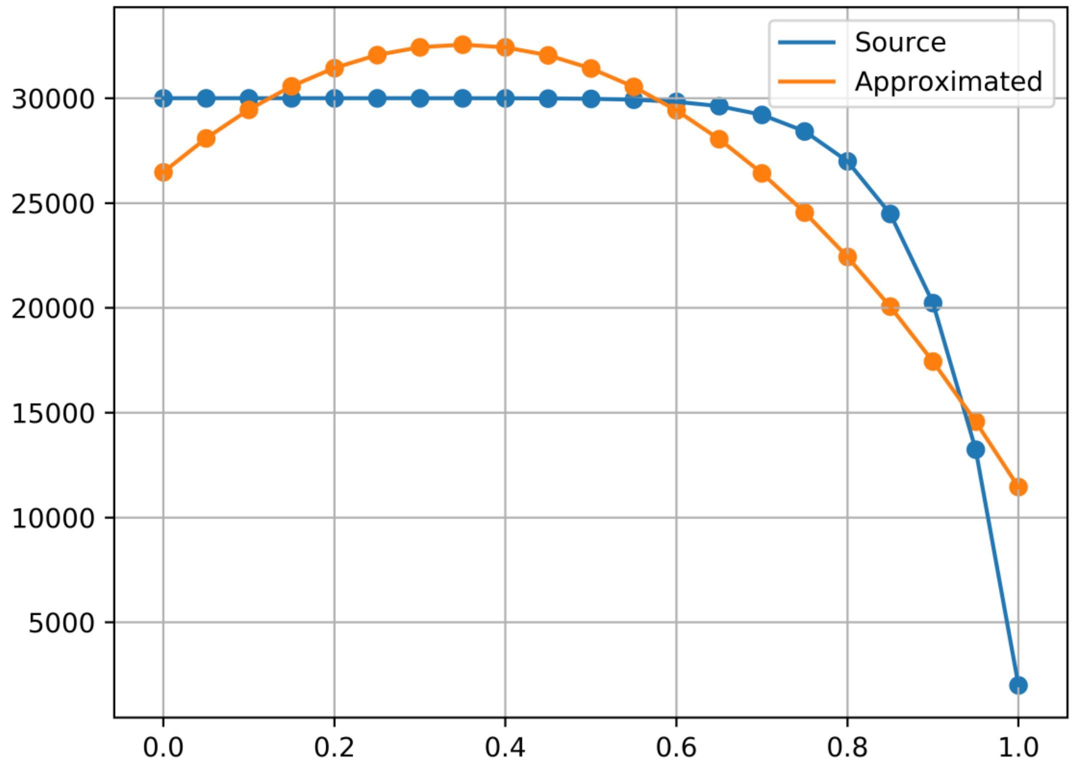

Рис. 3.1 Приближенный полином при точности 0.1

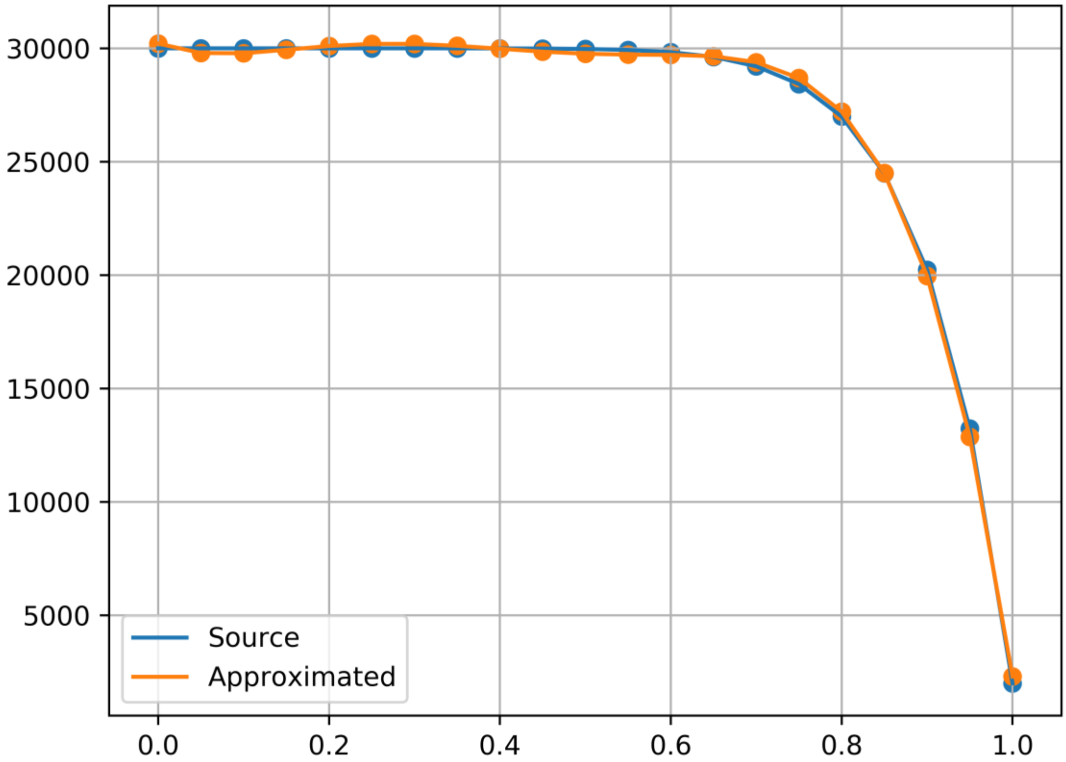

Рис. 3.2 Приближенный полином при точности 0.01

Вывод: качество аппроксимации очень хорошо получается при точности до 2 знака!

Оптимальный степень равен 2 раза меньше, чем в исходной формуле.

4) Экс. №4

Y = [30000, 29999, 29996, 29989, 29981, 29970, 29957, 29939, 26917, 20875, 20039, 19777, 19603, 19425, 19223, 19032, 18752, 18031, 13934, 4912, 3000]

Точность	Оптимальный степень полинома	Ошибка
0.1	2	0.0911172
0.01	20	0.0130245

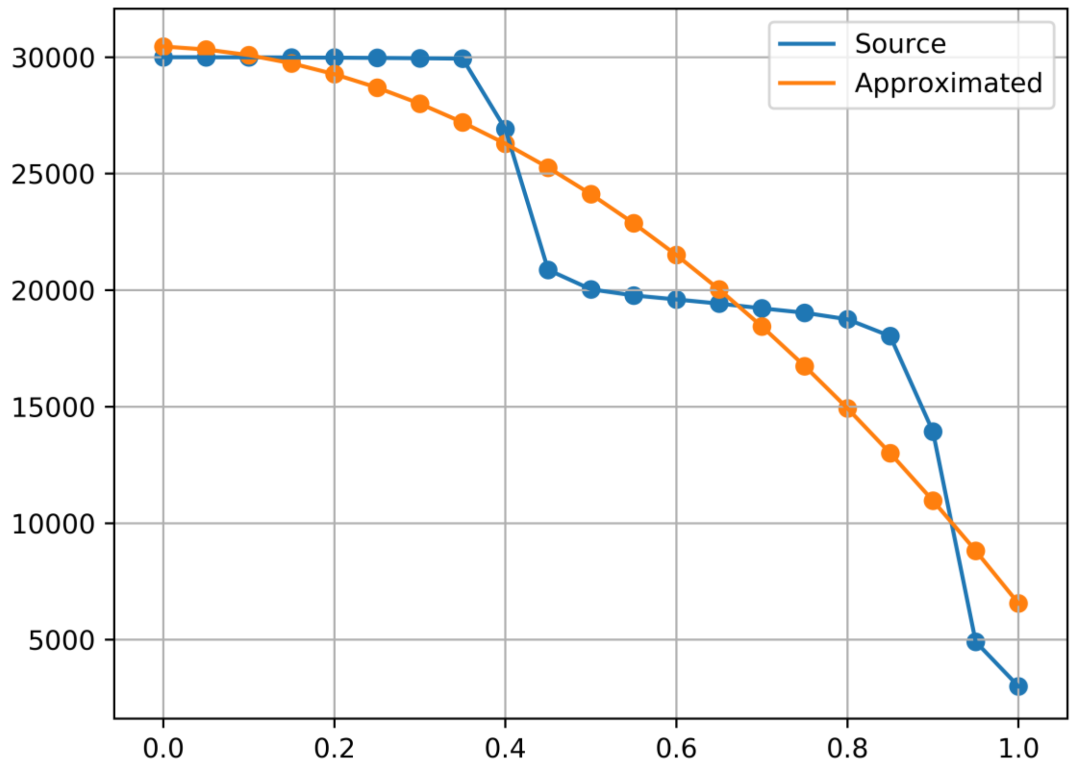

Рис. 4.1 Приближенный полином при точности 0.1

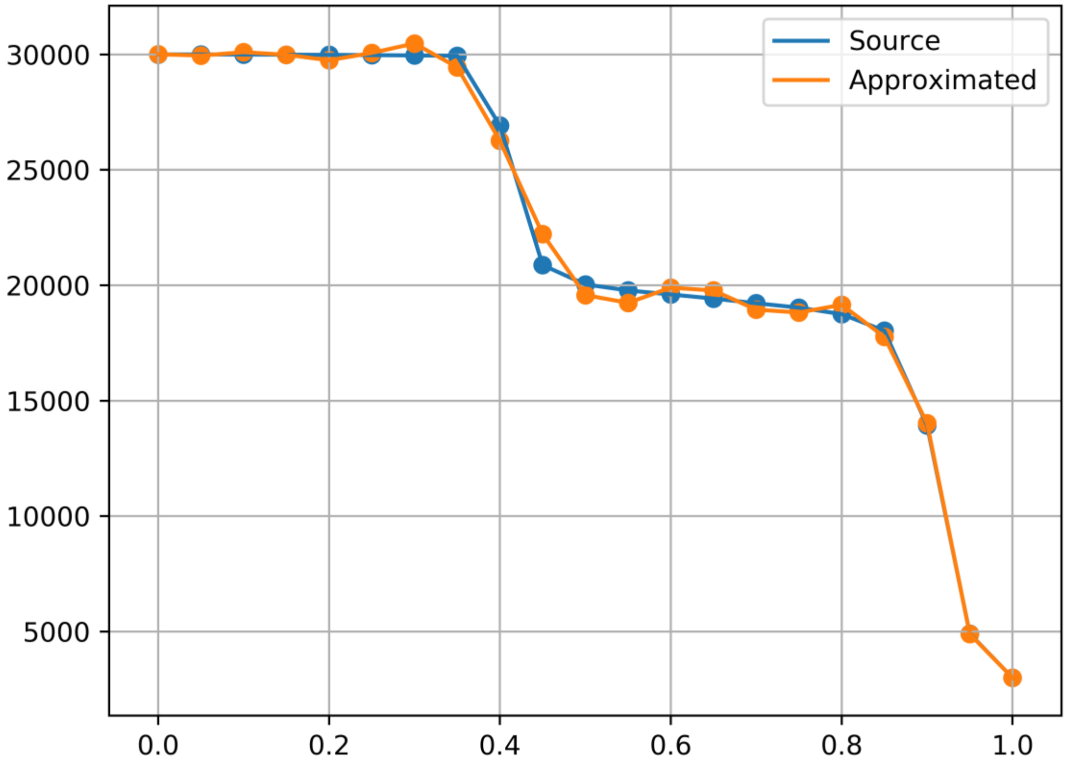

Рис. 4.2 Приближенный полином при точности 0.01

Вывод: качество аппроксимации хорошо получается при точности до 2 знака!

5) **Экс. №5**

Y определяется по равномерному закону распределения со следующим видом:

```
# 5
Y = np.random.uniform(low=2000, high=30000, size=(X.size,))
```

Точность	Оптимальный степень полинома	Ошибка
0.1	20	0.154872171
0.01	20	0.125793564

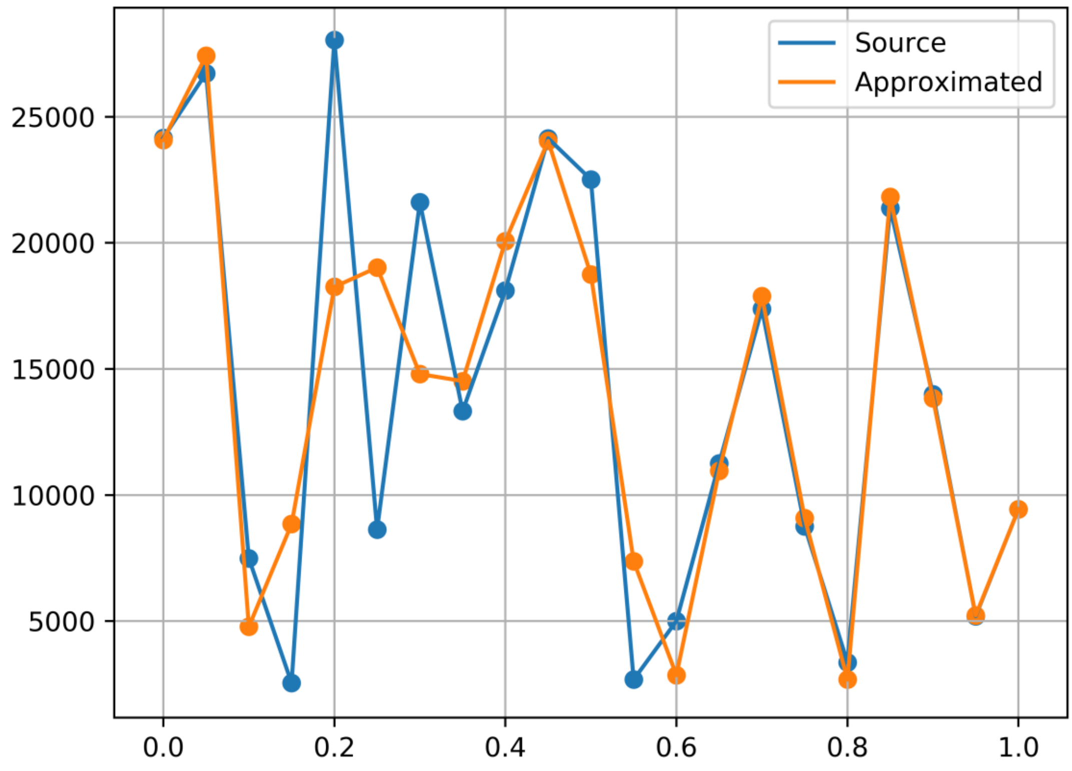

Рис. 5.1 Приближенный полином при точности 0.1

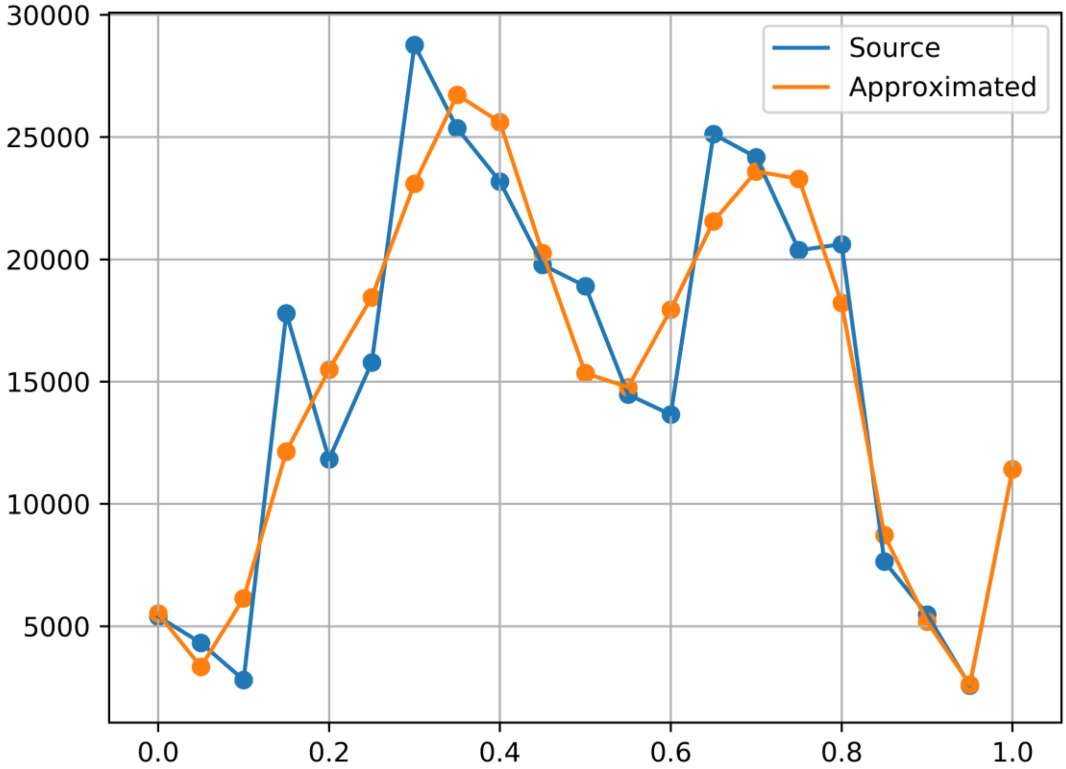

Рис. 5.2 Приближенный полином при точности 0.01

Вывод: качество аппроксимации улучше получается при точности до 2 знака!