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

Входные данные:

* множество упорядоченных точек по оси абцисса;
* точность.

Выходные данные:

* оптимальный степень полинома;
* ошибка приближенного полинома.

**Решение**

* **Алгоритм**

Обычным методом находим коэффициенты .

Используя метод наименьших квадратов:

Полученная система уравнений можно записывается в следующим матричном виде

Решением этого уравнения является:

* **Ошибка приближенного полинома**

Ошибка приближенного полинома вычисляется по следующим формулам:

|  |  |
| --- | --- |
| **Формула №1** | **Формула №2** |
|  |  |

Здесь среднее квадратическое Y.

* **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)

* изменяется

1. Замечание на рисуках:

* Линия синяя – данная;
* Линия оранжевая – аппроксимированная;

1. **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 раза меньше у исходной!

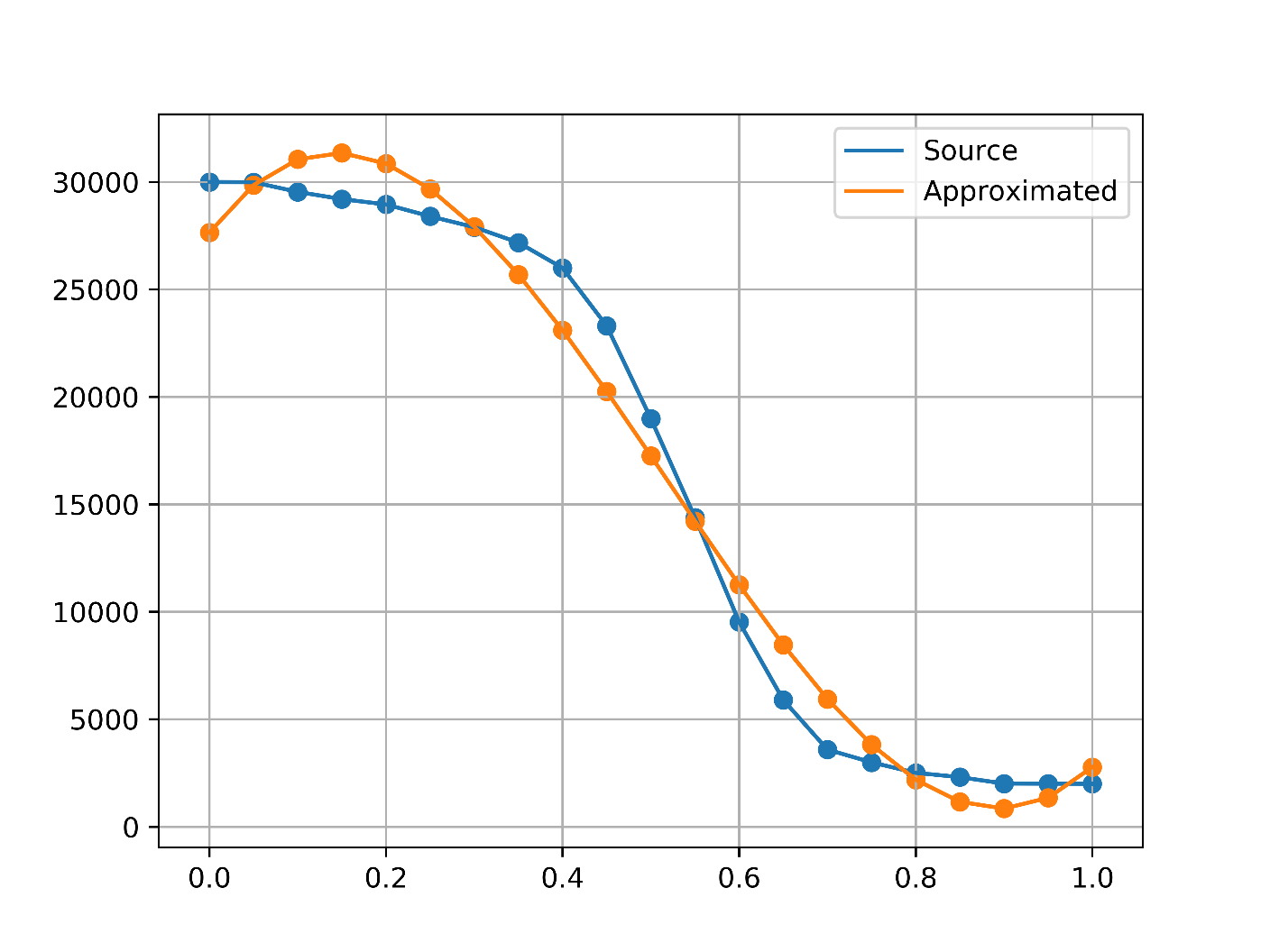
1. **Замечание**

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

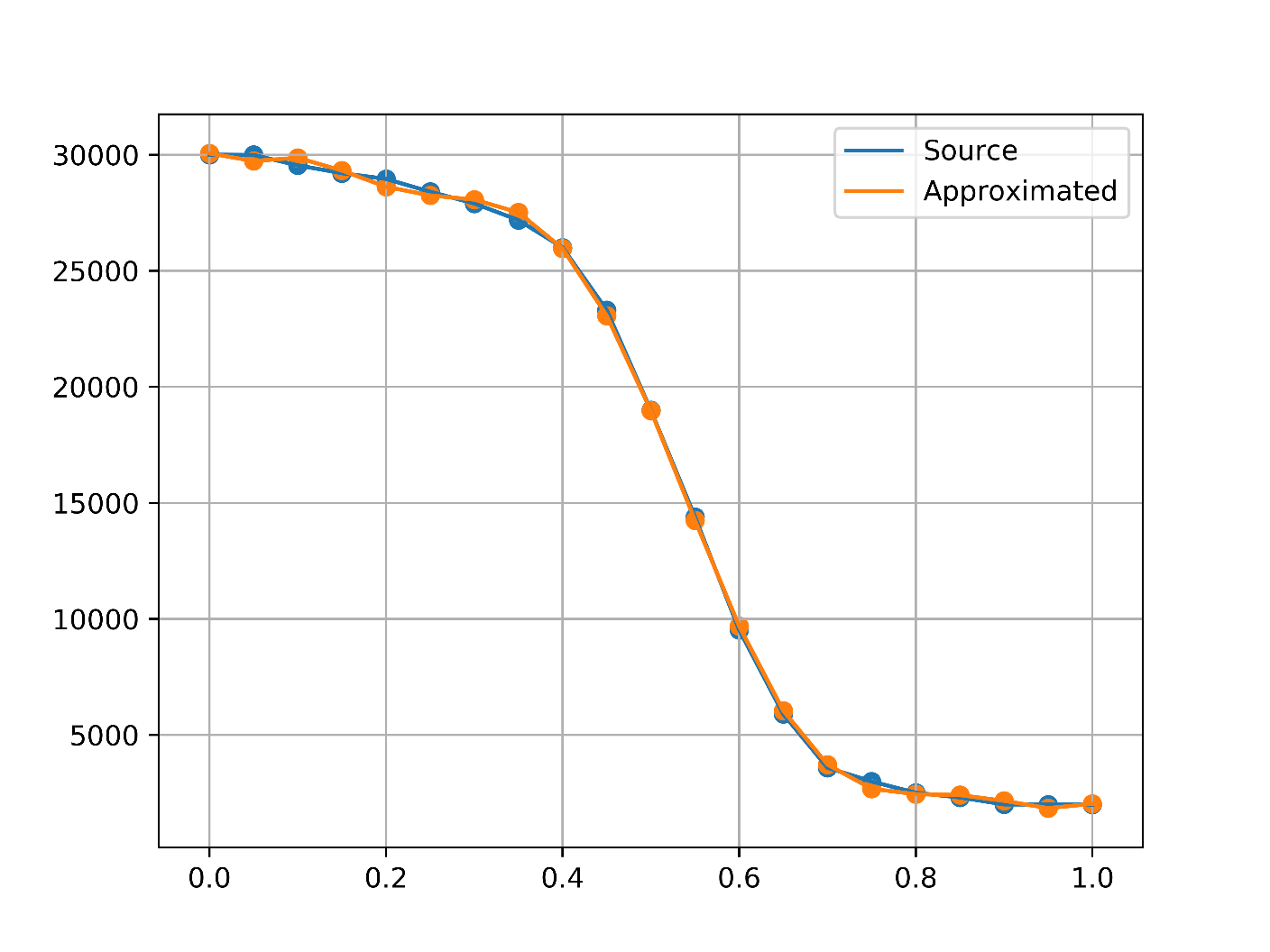
1. **Экс. №1**

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 | Ошибка |
|  | 3 | 0.070871 |
| 0.01 | 9 | 0.008108 |



*Рис. 1.1 Приближенный полином при точности 0.1*



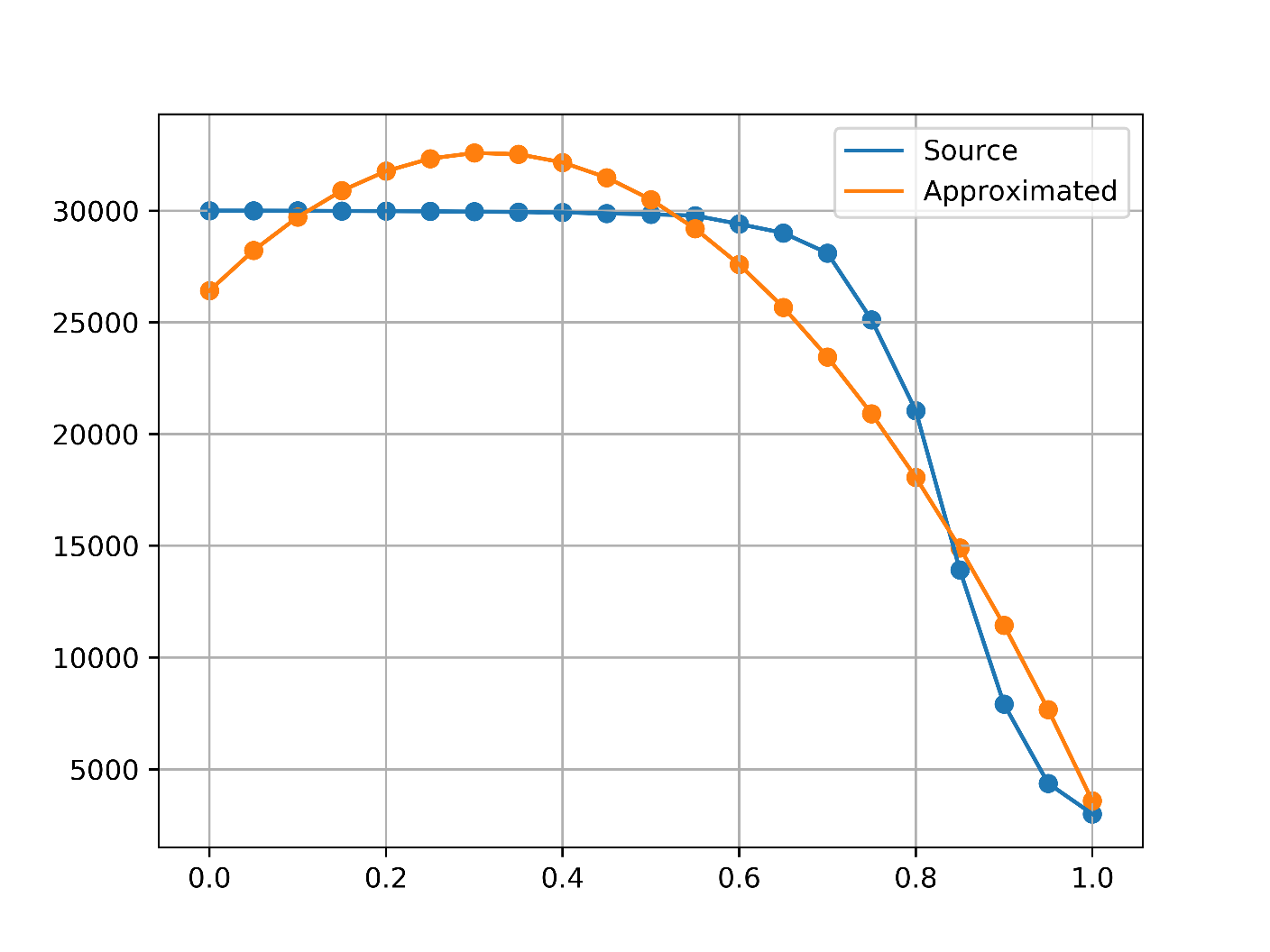
*Рис. 1.2 Приближенный полином при точности 0.01*

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

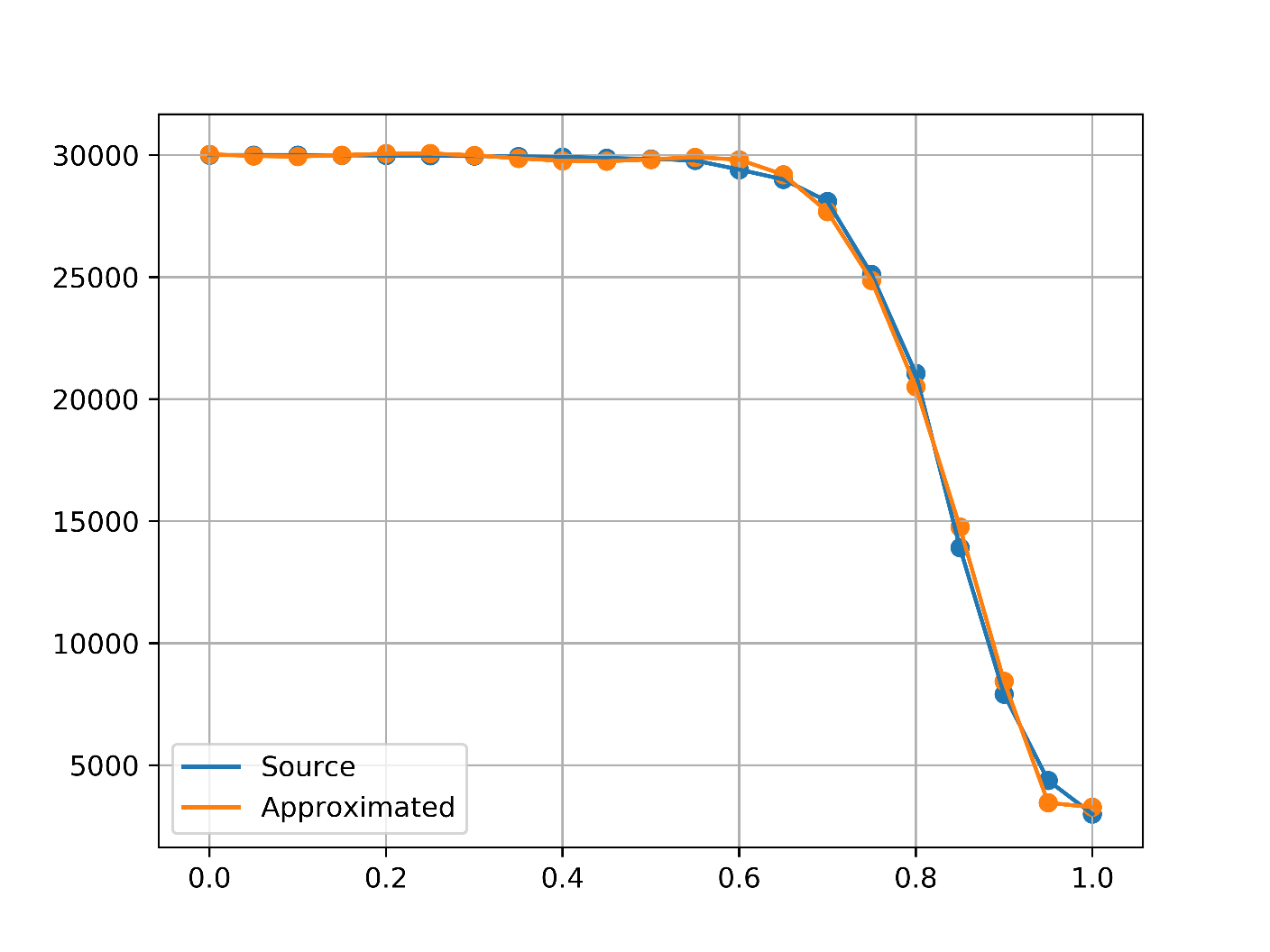
1. **Экс. №2**

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 | Ошибка |
|  | 2 | 0.083734 |
| 0.01 | 7 | 0.009463 |



*Рис. 2.1 Приближенный полином при точности 0.1*



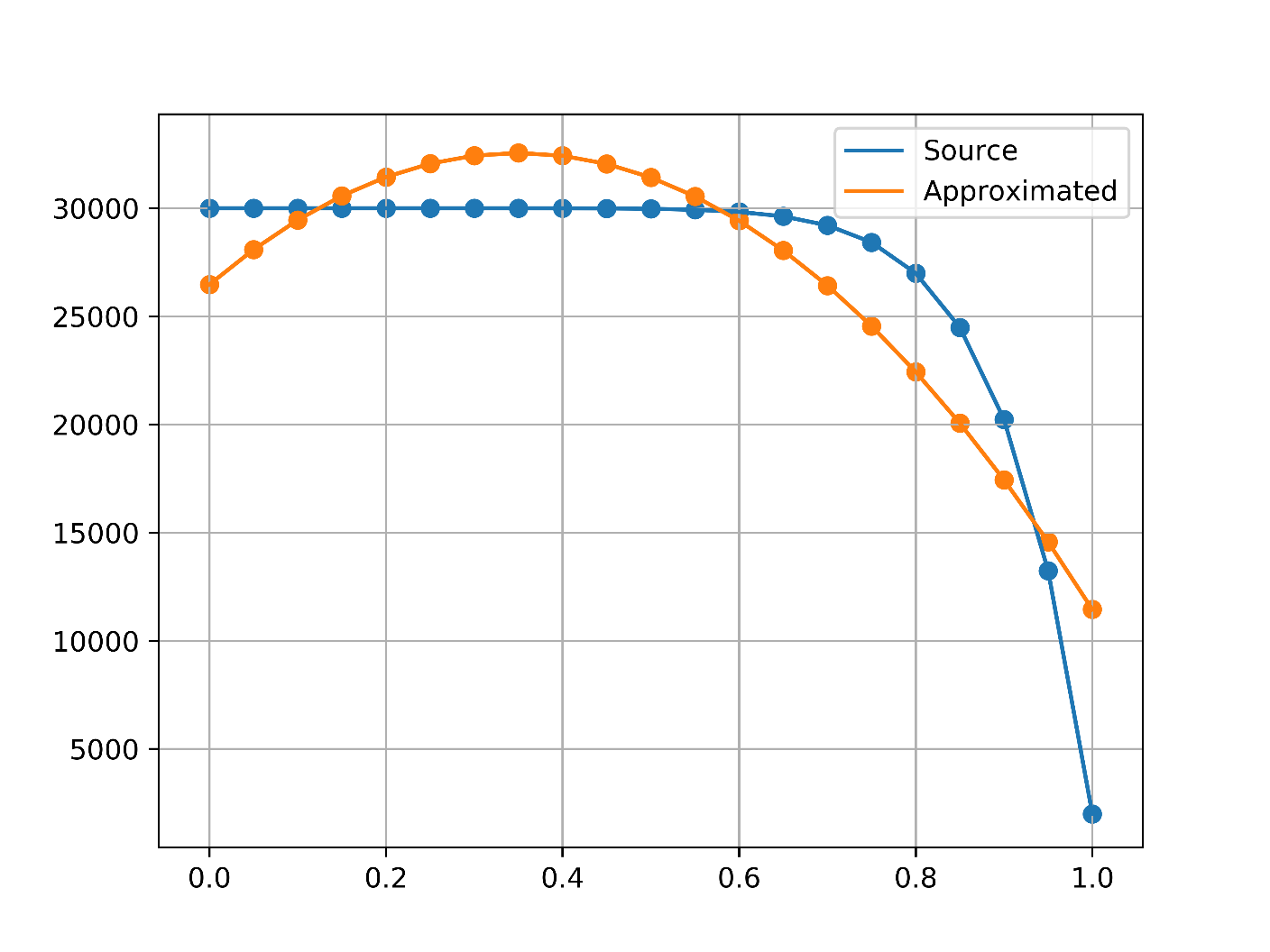
*Рис. 2.2 Приближенный полином при точности 0.01*

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

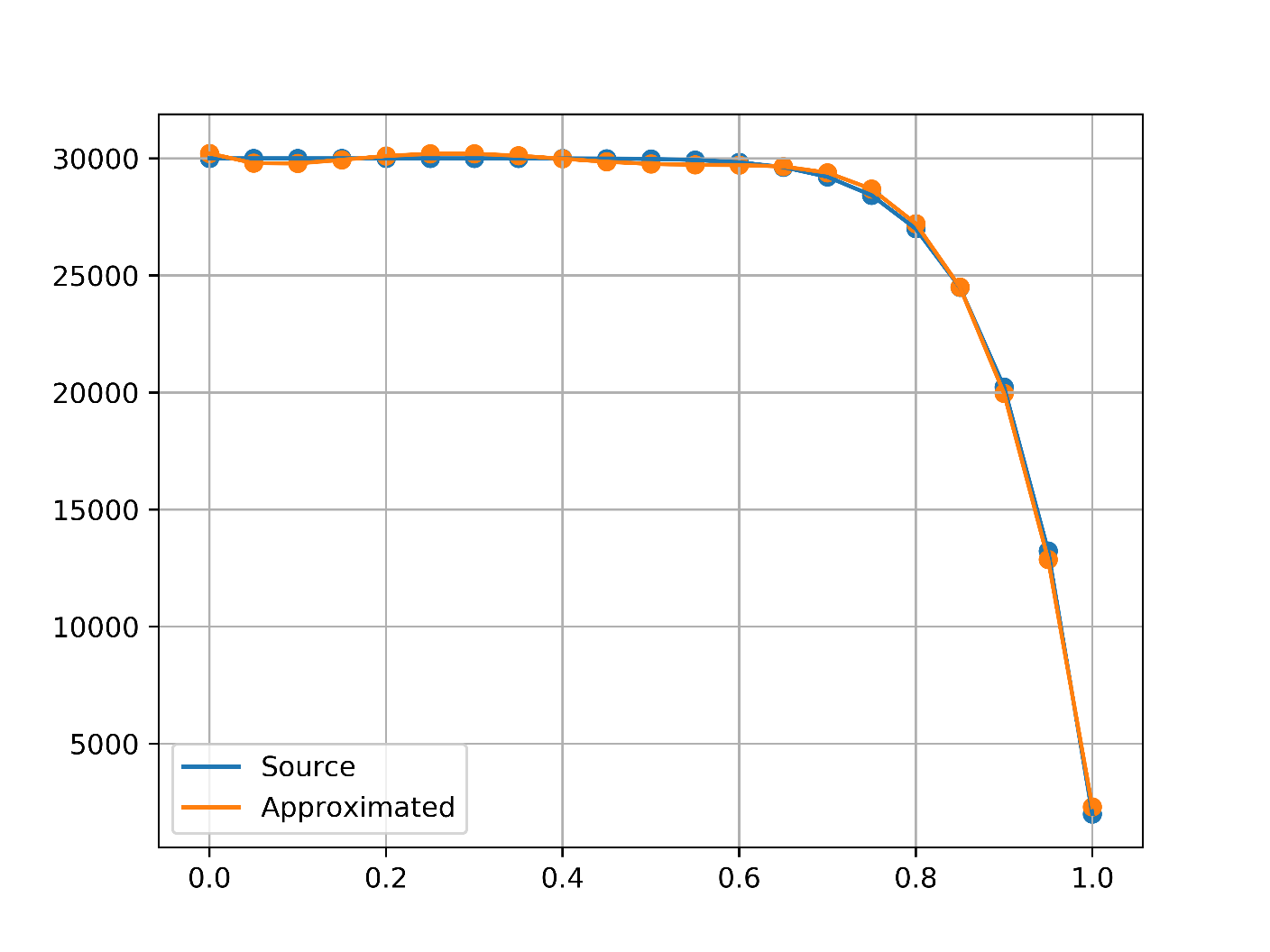
1. **Экс. №3**

Данный эксперимент реализуется по точной формуле вида:

|  |  |  |
| --- | --- | --- |
| Точность | Оптимальный степень полинома | Ошибка |
|  | 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*

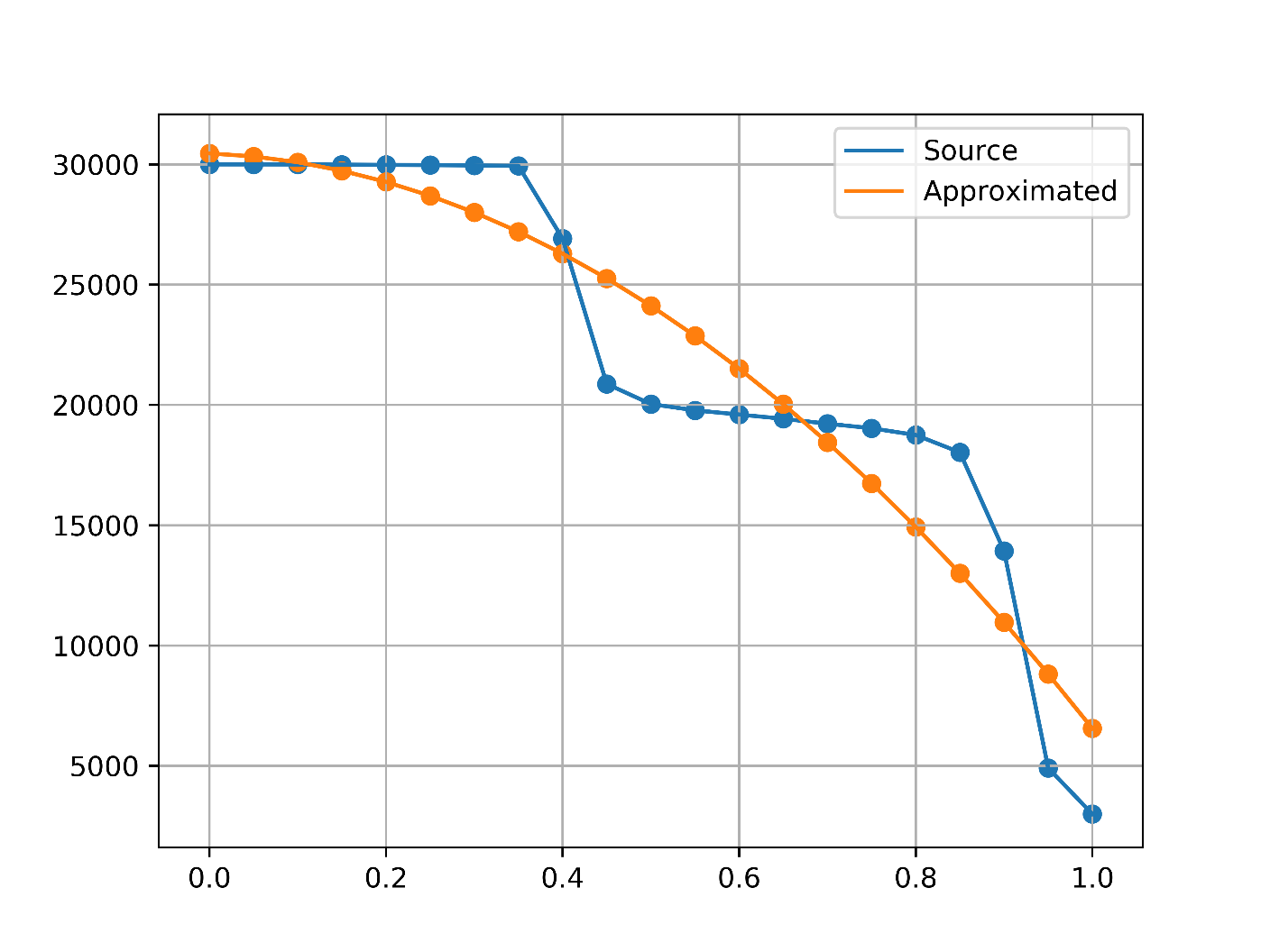
**Вывод**: качество аппроксимации очень хорошо получается при точности до 2 знака!

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

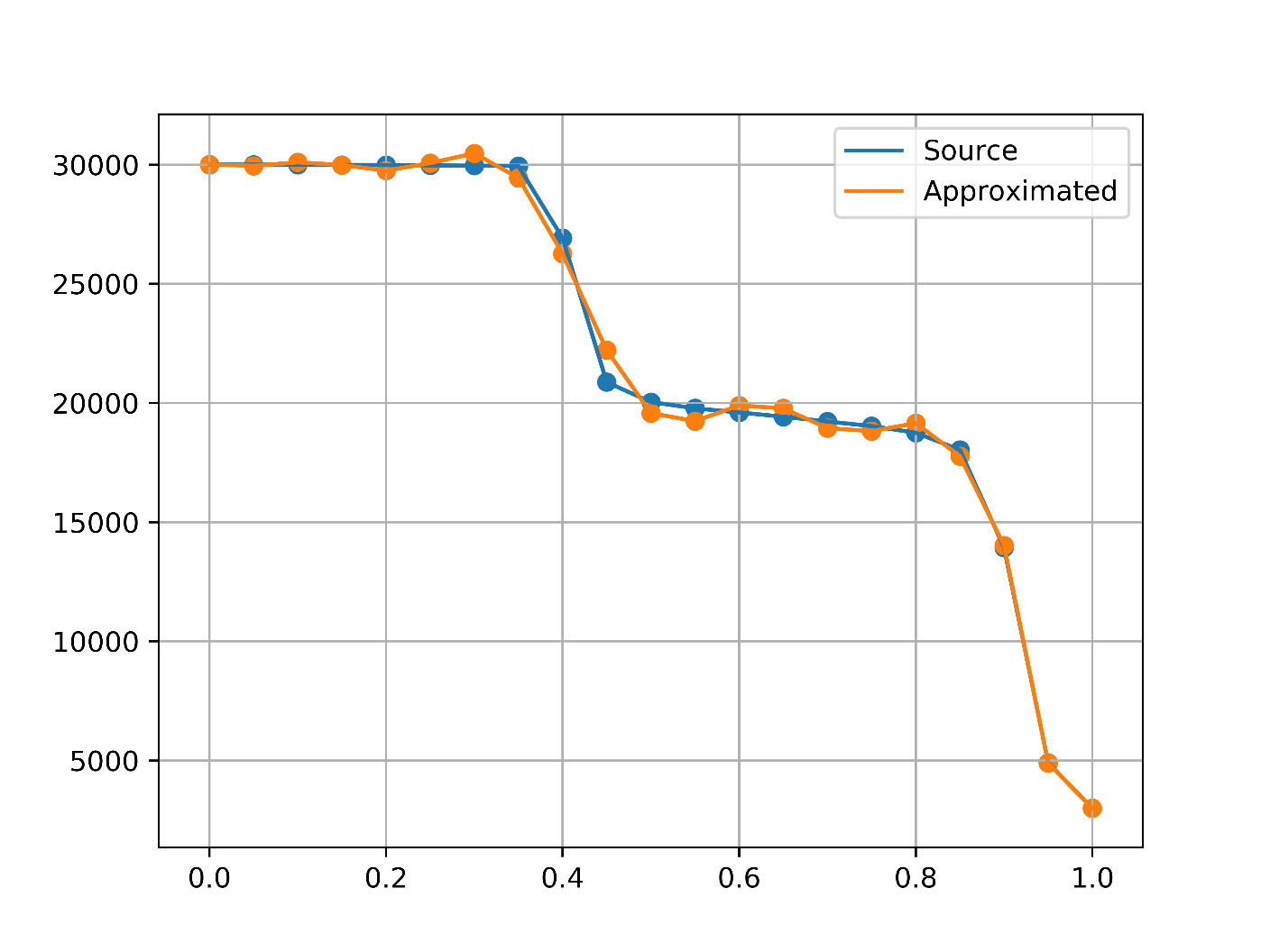
1. **Экс. №4**

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 |
|  | 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!

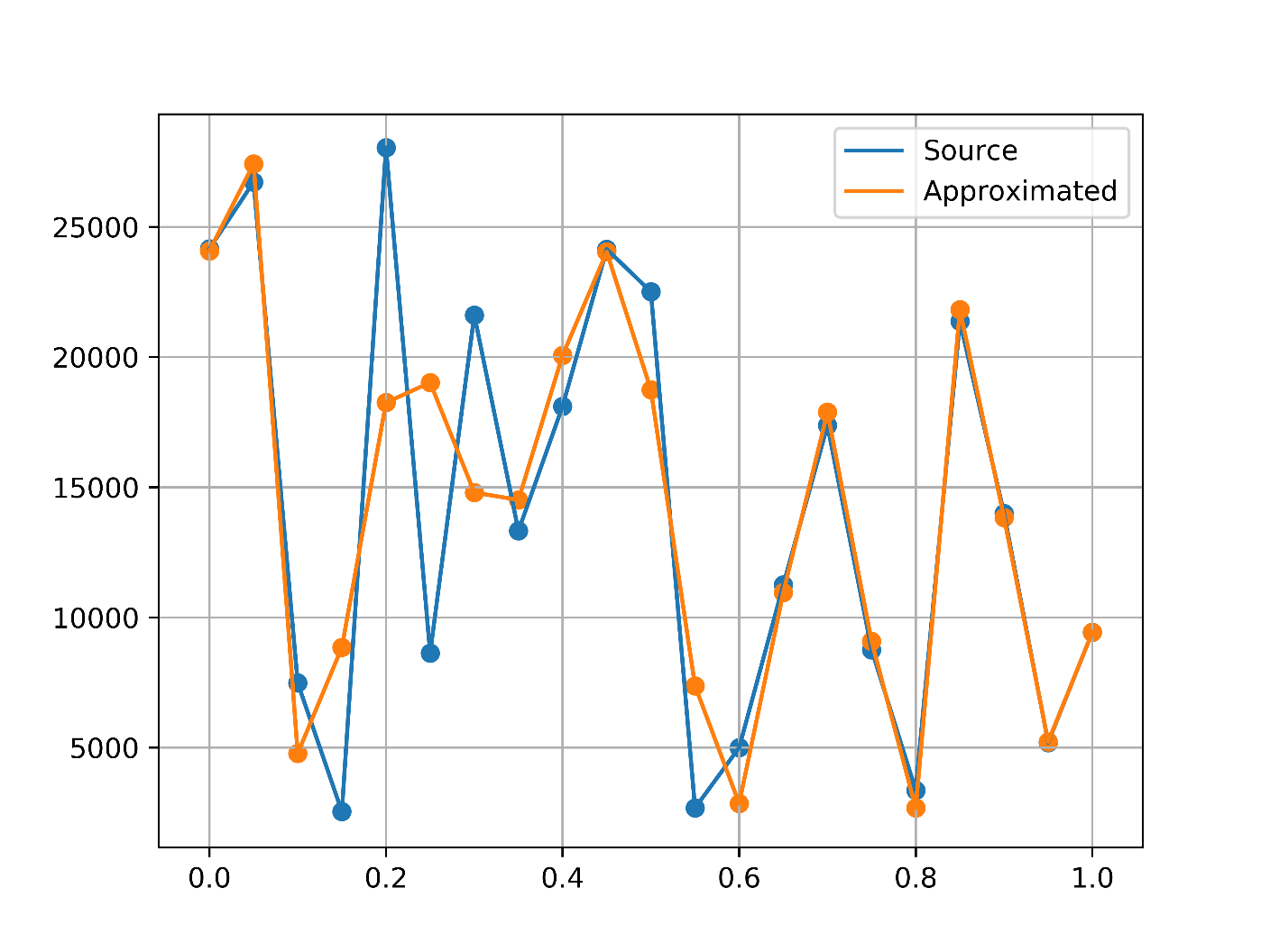
1. **Экс. №5**

Y определяется по равномерному закону распределения со следующим видом:

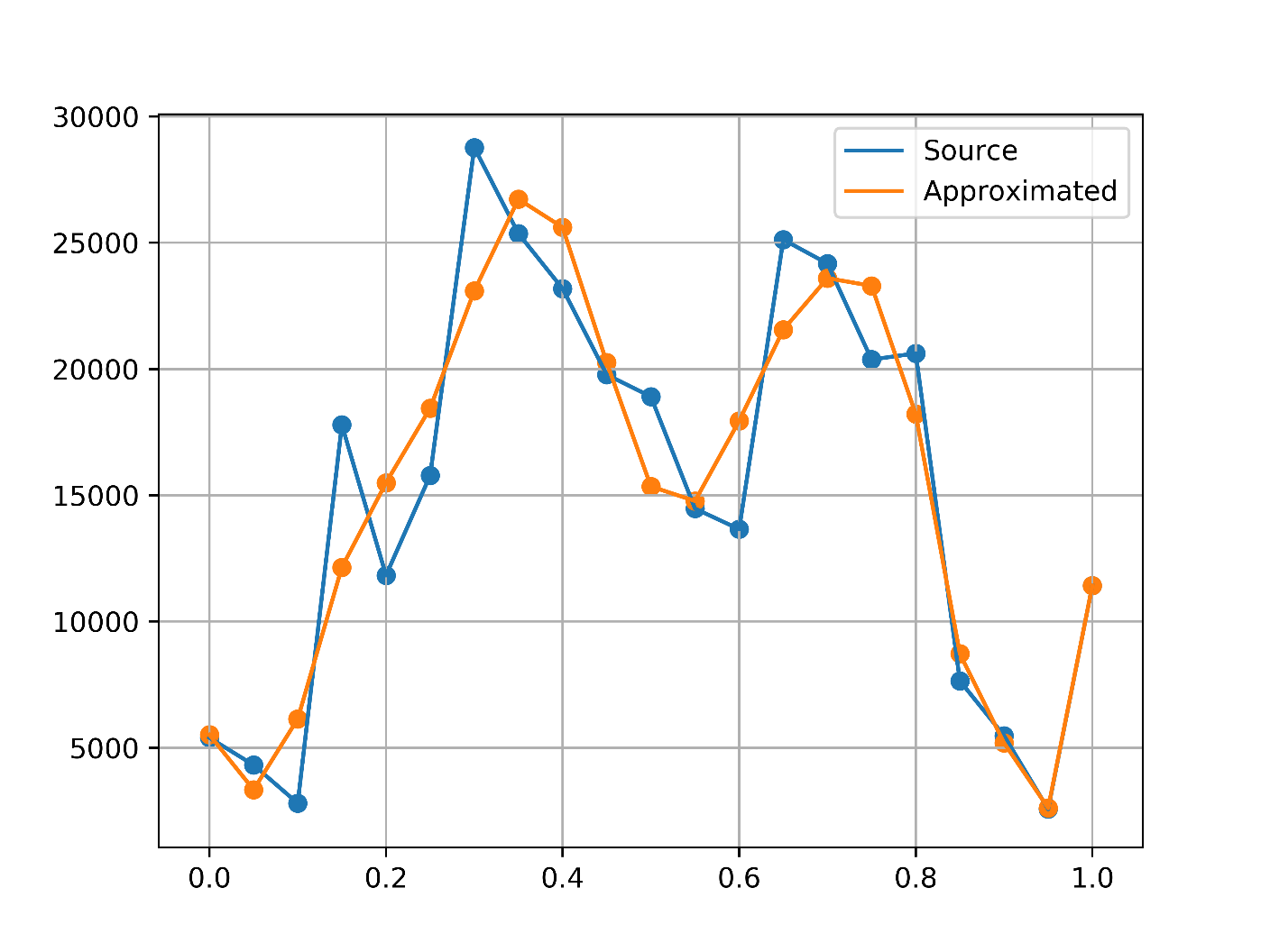
# 5

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

|  |  |  |
| --- | --- | --- |
| Точность | Оптимальный степень полинома | Ошибка |
|  | 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!