Отчет по лабораторной работе №5: Аппроксимация функции

Чигарев Дмитрий 381807-1

1. Исходные данные

x	0.351	0.664	0.978	1.291	1.605	1.918	2.232	2.546	2.859
y	0.605	0.265	0.064	0.116	0.415	0.728	1.673	3.138	5.092

Задана табличная функция $y = f(x)$, требуется найти функцию $\mathrm{F}(x,p)$ наилучшим образом её аппроксимирующую:

$$\min_{p \in R^k} ||f(x) - F(x,p)|| \quad (1)$$

В качестве функций $\mathrm{F}(x,p)$ предложено использовать следующие варианты:

1) $y = ax^2 + bx + c$

2) $y = \dfrac{a}{x^2} + \dfrac{b}{x} + c$

3) $y = bx^a + c$

4) $y = be^{ax} + c$

5) $y = \dfrac{b}{x+a} + c$

6) $y = ax + be^{-x} + c$

7) $y = \dfrac{a}{x} + be^x + c$

8) $y = a\ln x + be^x + c$

9) $y = b\exp(-a(x+c)^2) + c$

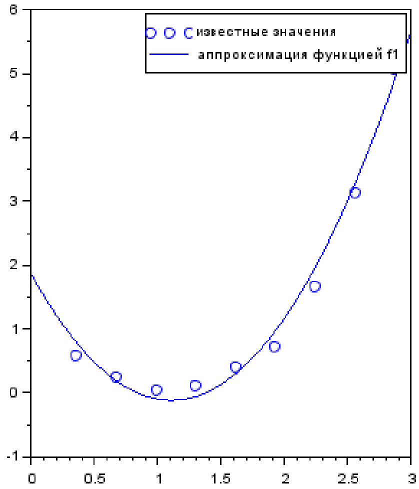
10) $y = a\sqrt{x} + b\sin x + c$

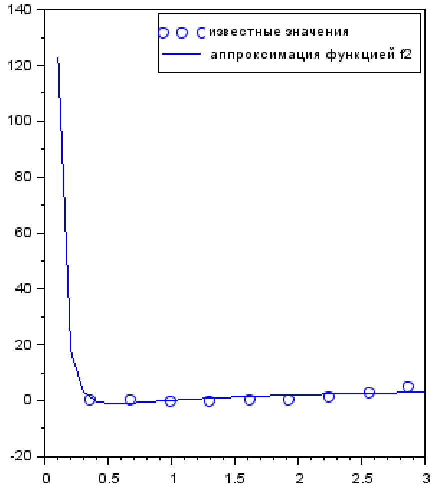
2. SciLab datafit

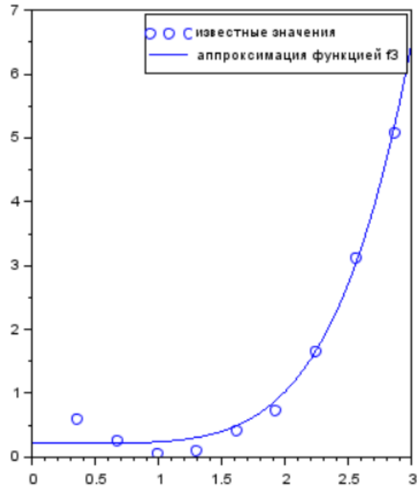
SciLab имеет встроенную функцию datafit, которую можно использовать для аппроксимации функций. Метод принимает на вход функцию отклонения $G(p,x,y) \coloneqq y - F(x,p)$, таблично заданную функцию $[x;y]$ и отыскивает вектор p путем минимизации следующей суммы неким не задокументированным способом:

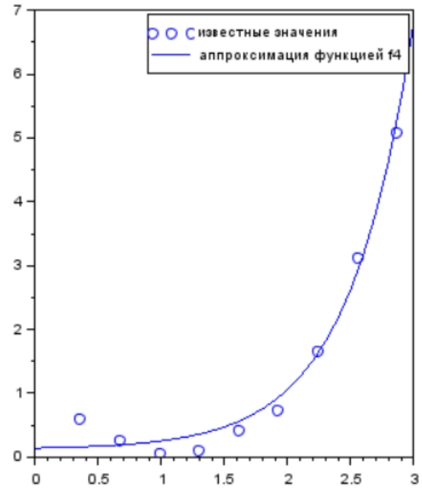
$$\sum_{i=1}^{n} G(p,x_i,y_i)^T * G(p,x_i,y_i)$$

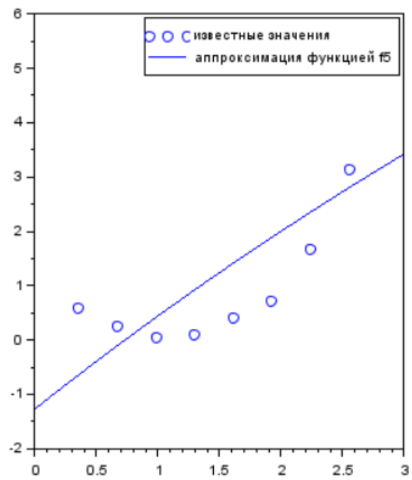
Запустим метод для 10 представленных функций и нарисуем, какие функции $F(x,p)$ мы получили:

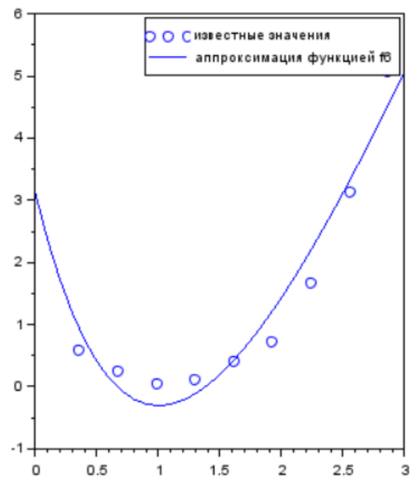

известные значения
аппроксимация функцией f1

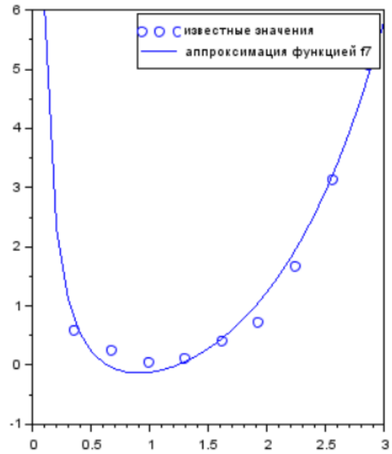

известные значения
аппроксимация функцией f2

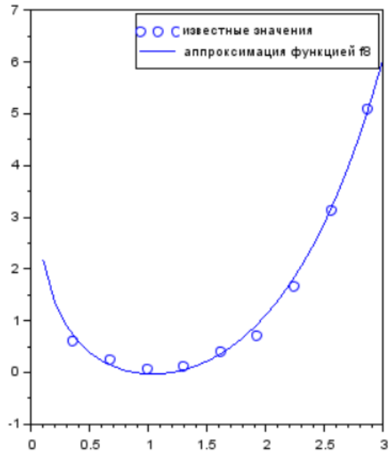

известные значения
аппроксимация функцией f3

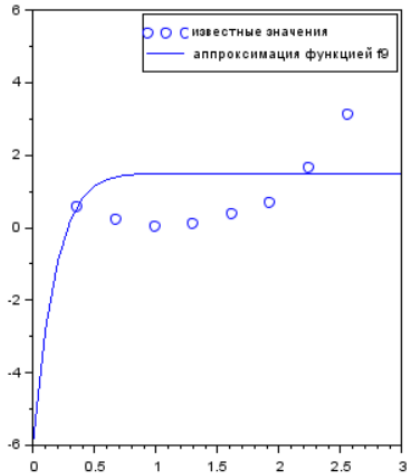

известные значения
аппроксимация функцией f4

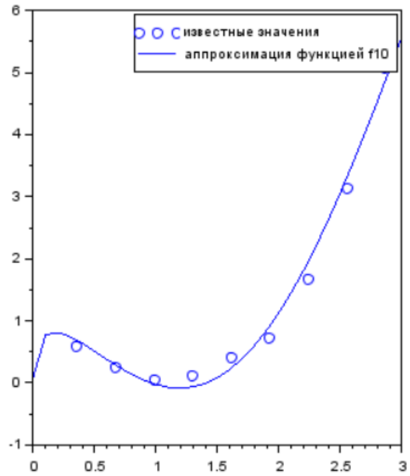

известные значения
аппроксимация функцией f5


известные значения
аппроксимация функцией f6









Помимо найденного вектора p метод возвращает среднее отклонение $F(x, p)$ от $f(x)$

Функция	Невязка	std невязки	Datafit error	Коэффициенты
1	0.3001609	0.1937011	1.826231e-01	1.6159313 -3.5713519 1.8543995
2	10.589244	1.1505023	1.084704e+00	1.8548665 -6.7461345 4.9918623
3	0.2803654	0.1872049	1.764985e-01	5.0417897 0.0248264 0.2121952
4	0.4422643	0.2351234	2.216765e-01	1.9659115 0.0184186 0.1207107
5	8.7990524	1.0486536	9.887732e-01	20.087452 -725.50576 34.849521
6	1.2280867	0.3918046	3.693969e-01	4.7467913 12.913721 -9.8047619
7	0.3685086	0.2146243	2.023497e-01	0.7468319 0.3725364 -1.8836816

8	0.1070554	0.1156803	1.090644e-01	-1.2607352 0.432039 -1.2057716
9	22.617770	1.6808880	1.585271e+00	1.7350857 -370.54299 1.5030636
10	0.2665145	0.1825221	1.720835e-01	3.5257753 -4.3356736 0.0928693

Из полученных данных видно, что лучшее приближение дает 8-я по счету модель:

$$F(x, a, b, c) = alog(x) + be^x + c$$

Её можно выразить как линейную комбинацию некоторых базисных функций:

$$F(x, p) = \sum_{i=1}^{3} p_i g_i(x)$$
$$g_1(x) = \log(x)$$
$$g_2(x) = e^x$$
$$g_3(x) = 1$$

Оптимальное значения для коэффициентов можно получить, решив СЛАУ, задающую ограничение (1). Полученная СЛАУ будет выглядеть следующим образом:

$$\begin{cases} a\langle\log(x_1)|\log(x_1)\rangle + b\langle e^{x_1}|\log(x_1)\rangle + c\langle 1|\log(x_1)\rangle = \langle\log(x_1)|y_1\rangle \\ a\langle\log(x_2)|e^{x_2}\rangle + b\langle e^{x_2}|e^{x_2}\rangle + c\langle 1|e^{x_2}\rangle = \langle e^{x_2}|y_2\rangle \\ a\langle\log(x_3)|1\rangle + b\langle e^{x_3}|1\rangle + c\langle 1|1\rangle = \langle 1|y_3\rangle \end{cases}$$

Сформируем систему и решим её встроенным в SciLab методом linsolve:

```
>>> [A, b] = build_system(bias_fn, X, Y)
>>> p = linsolve(A, -b)
>>> disp("Оптимальные параметры для модели 8:", p)

  "Оптимальные параметры для модели 8:"

  -1.260735569618991
   0.43203908137157
  -1.205771896320093
```

Посчитаем ошибку аппроксимации для полученной модели:

```
>>> predicted_vals = error_fn(p, [X; Y])
>>> disp("Значение невязок в исходных точках:", predicted_vals' * predicted_vals)
>>> disp("Средне-квадратичное отклонение ошибки:", stdev(predicted_vals))

  "Значение невязок в исходных точках:"

   0.1070554

  "Средне-квадратичное отклонение ошибки:"

   0.1156803
```

Приложения

1. Ссылка на исходный код: https://github.com/proxodilka/numerical-analysis-labs/blob/master/lab5_approximation/lab5.sce