ЛР1 (вариант 7)

Содержание

0. Подготовка окружения

```
clear; clc; close all;
format short g; rng(7,'twister');
```

1. Ввод/проверка данных

```
a = [1.2, -4.1, -0.8, -0.7, -2.2, 1.7, 3.3, -6.1]; % вектор a (8 эл.)
b = [-1.5,  2.2,  1.0, -4.3, -0.0, -1.8, -1.5,  2.4]; % вектор b (8 эл.)



A = [ 0.23,  3.89, -4.23, -7.25;
      5.84,  5.13, -0.89,  3.55 ];
```

```
>> Данные (размерности):
  Name        Size            Bytes  Class     Attributes

  A           2x4                64  double
  a           1x8                64  double
  b           1x8                64  double
```

2. Операции с векторами

%.2.1 Сумма, разность, скалярное произведение

```
sum_ab  = a + b;                    % покомпонентная сумма
diff_ab = a - b;                    % покомпонентная разность
dot_ab  = dot(a, b);                % скалярное произведение (a·b)

%.2.2 Вектор c = [a, b], поиск min/max и обмен местами
c = [a, b];                         % конкатенация: c = [a b] (длина 16)
[c_max, i_max] = max(c);            % максимум и его индекс
[c_min, i_min] = min(c);            % минимум и его индекс
c_swapped = c;                      % копия для обмена
c_swapped([i_max, i_min]) = c([i_min, i_max]); % обмен min/max по индексам

% 2.3 Сортировка по возрастанию/убыванию
```

```
c_asc  = sort(c);                   % сортировка по возрастанию
c_desc = sort(c, 'descend');        % сортировка по убыванию

% 2.4 Реверс (обратный порядок)
c_rev = fliplr(c);                  % разворот вектора слева-направо

% 2.5 Векторное произведение u x v, где u=[a1,a3,a4], v=[b2,b3,b4]
u = [a(1), a(3), a(4)];             % формируем u из элементов a
v = [b(2), b(3), b(4)];             % формируем v из элементов b
cross_uv = cross(u, v);             % векторное произведение u×v (3D)
```

```
== п.2.1: сумма/разность/скалярное произведение ==
    i      a       b       a_plus_b    a_minus_b
    _    ____    ____    ________    _________

    1     1.2    -1.5      -0.3          2.7
    2    -4.1     2.2      -1.9         -6.3
    3    -0.8       1       0.2         -1.8
    4    -0.7    -4.3        -5          3.6
    5    -2.2       0      -2.2         -2.2
    6     1.7    -1.8      -0.1          3.5
    7     3.3    -1.5       1.8          4.8
    8    -6.1     2.4      -3.7         -8.5

dot(a,b) = -31.26

== п.2.2: вектор c, min/max и обмен местами ==
    i     c_original    c_after_swap_min_max
    __    __________    ____________________

     1        1.2                 1.2
     2       -4.1                -4.1
     3       -0.8                -0.8
     4       -0.7                -0.7
     5       -2.2                -2.2
     6        1.7                 1.7
     7        3.3                -6.1
     8       -6.1                 3.3
     9       -1.5                -1.5
    10        2.2                 2.2
```

```
    11          1                  1
    12       -4.3               -4.3
    13          0                  0
    14       -1.8               -1.8
    15       -1.5               -1.5
    16        2.4                2.4

c_min = -6.1 (index 8), c_max = 3.3 (index 7)

== п.2.3: сортировка c ==
    c_ascend    c_descend
    ________    _________

      -6.1         3.3
      -4.3         2.4
      -4.1         2.2
      -2.2         1.7
      -1.8         1.2
      -1.5           1
      -1.5           0
      -0.8        -0.7
      -0.7        -0.8
         0        -1.5
         1        -1.5
       1.2        -1.8
       1.7        -2.2
       2.2        -4.1
       2.4        -4.3
       3.3        -6.1

== п.2.4: реверс c ==
  Columns 1 through 6

          2.4          -1.5          -1.8             0          -4.3
1

  Columns 7 through 12

          2.2          -1.5          -6.1           3.3           1.7
-2.2

  Columns 13 through 16
```

-0.7	-0.8	-4.1	1.2

== п.2.5: векторное произведение u x v ==

u_[a1,a3,a4]			v_[b2,b3,b4]			cross_u_x_v		
1.2	-0.8	-0.7	2.2	1	-4.3	4.14	3.62	2.96

3. Вычисление f(x) на матрице A

== п.3: исходная матрица A ==

	c1	c2	c3	c4
r1	0.23	3.89	-4.23	-7.25
r2	5.84	5.13	-0.89	3.55

== п.3: результат f(A) ==

	c1	c2	c3	c4
r1	0-0.28536i	7.7106	9.6032+0i	36.122
r2	21.548+0i	15.668	0-0.024369i	6.0337

```
X  = A;                              % согласно заданию: x := A
fX = fx(X);                          % вызов функции fx(x) (см. ниже)
```

4. Аппроксимация полиномом

```
x = linspace(0, 2*pi, 10);           % 10 узлов на [0, 2π]
y = sin(x) + 0.1*randn(size(x));     % модельная зависимость + шум
(пример)

% Базовый полином для таблицы остатков (как и прежде)
n = 4;
k = polyfit(x, y, n);                % коэффициенты полинома степени n
z = polyval(k, x);                   % значения полинома в узлах x
res = y - z;                         % только разности (остатки)
```

```
% 4.Б. Мелкая сетка x1 и поведение полинома при увеличении степени
x1_min = min(x); x1_max = max(x);     % границы
h = (x1_max - x1_min) / 999;          % шаг сетки (1000 точек)
x1 = x1_min:h:x1_max;                 % мелкая равномерная сетка

deg_list = 2:5;                       % степени для сравнения

% Обзорный график поведения для разных степеней (между узлами)
figure('Name','П.4.Б: поведение полинома на x1 при росте степени');
plot(x, y, 'o', 'LineWidth', 1.2); hold on;     % исходные точки
for d = deg_list
    k_d = polyfit(x, y, d);           % полином степени d (МНК)
    z1_d = polyval(k_d, x1);          % значения полинома на x1
    plot(x1, z1_d, '-', 'LineWidth', 1.0);
end
grid on; xlabel('x'); ylabel('z_d(x)');
title('Поведение аппроксимирующего полинома на мелкой сетке x_1');
leg = [{'исходные точки y(x_i)'} arrayfun(@(d) sprintf('степень %d',
d), deg_list, 'UniformOutput', false)];
legend(leg{:}, 'Location','best');

% 4.В. Для КАЖДОГО полинома — свой график ошибки (остатков) y - z_d в
узлах
for d = deg_list
    k_d = polyfit(x, y, d);           % коэффициенты полинома степени d
    z_d = polyval(k_d, x);            % значения полинома в узлах x
    res_d = y - z_d;                  % остатки для степени d

    figure('Name', sprintf('П.4.В: остатки y - z (степень %d)', d));
    stem(x, res_d, 'filled'); grid on;
    xlabel('x_i'); ylabel('y(x_i) - z_d(x_i)');
    title(sprintf('Остатки аппроксимации (степень %d)', d));
end
```

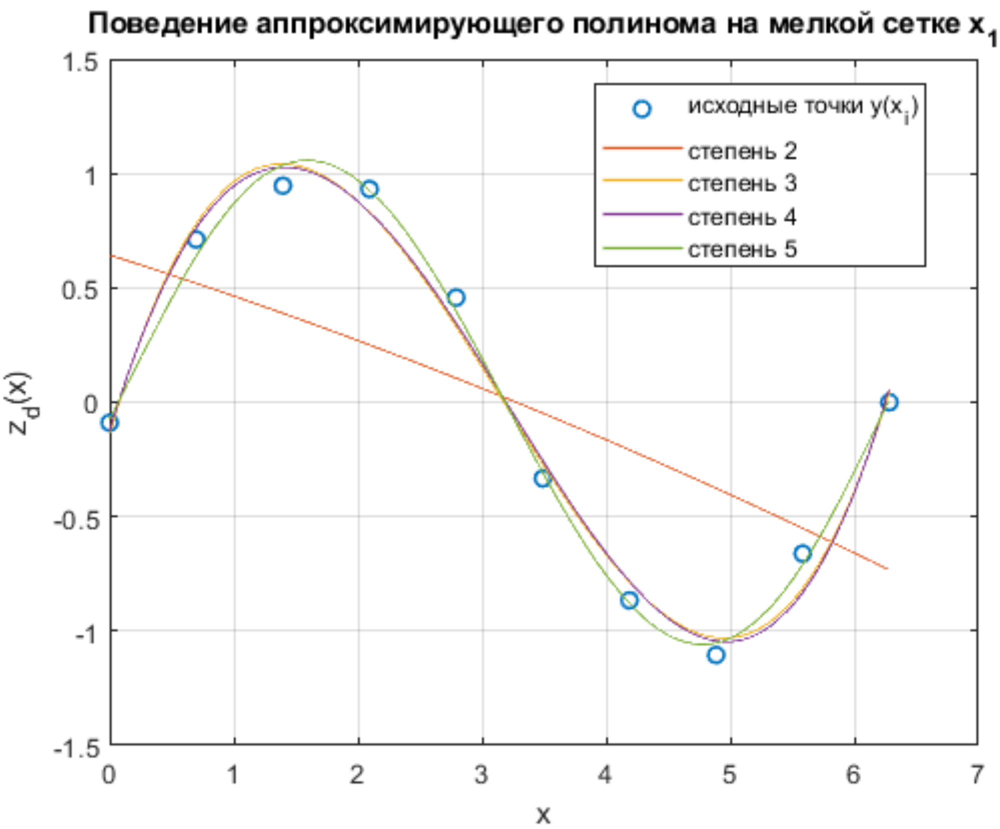
```
== П.4.А: Оценка качества приближения по разности y(x_i) - z(x_i)
(n=4) ==
      x_i          y_xi          z_xi       y_minus_z
    _______     ___________    ________    __________

          0       -0.089678    -0.12458      0.034903
    0.69813         0.71137     0.76217     -0.050801
     1.3963         0.94603       1.027     -0.080994
```

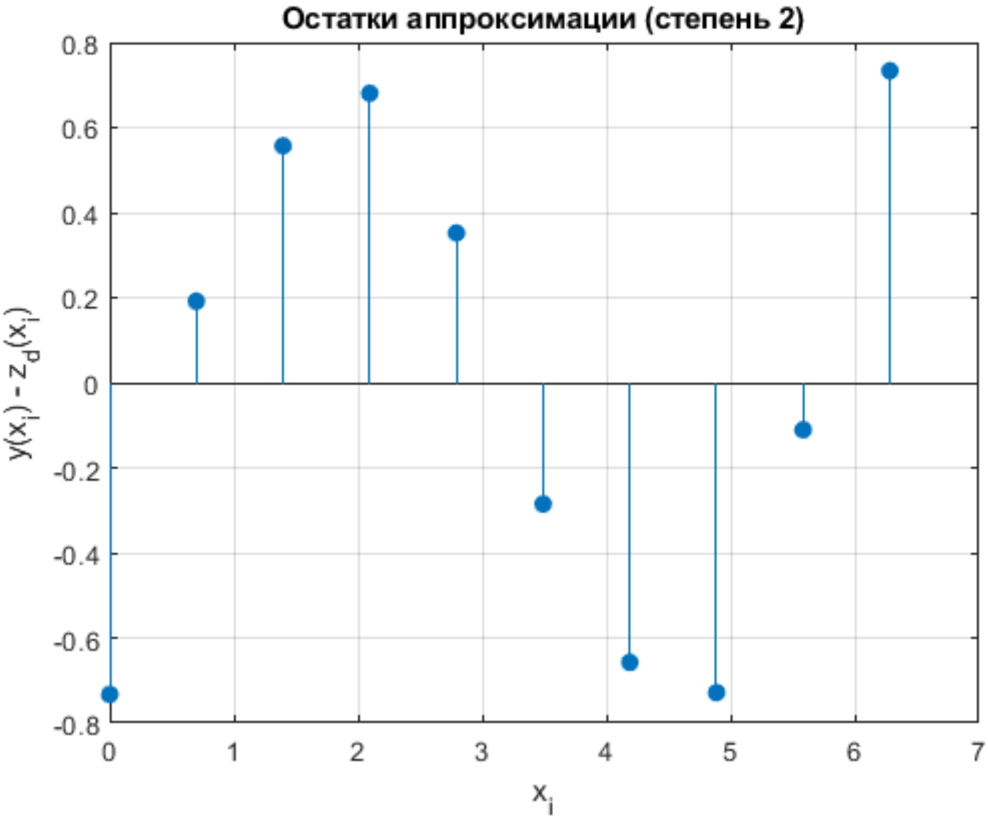
2.0944	0.93133	0.83109	0.10025
2.7925	0.45714	0.34391	0.11323
3.4907	-0.33335	-0.25651	-0.076839
4.1888	-0.86586	-0.78375	-0.082117
4.8869	-1.1047	-1.0429	-0.061822
5.5851	-0.6617	-0.83067	0.16897
6.2832	-3.7365e-05	0.064733	-0.064771

Локальные функции

```
function y = fx(x)
    y = (sqrt(x.^2 - 1).^3) ./ (abs(x) + 3);
end
```

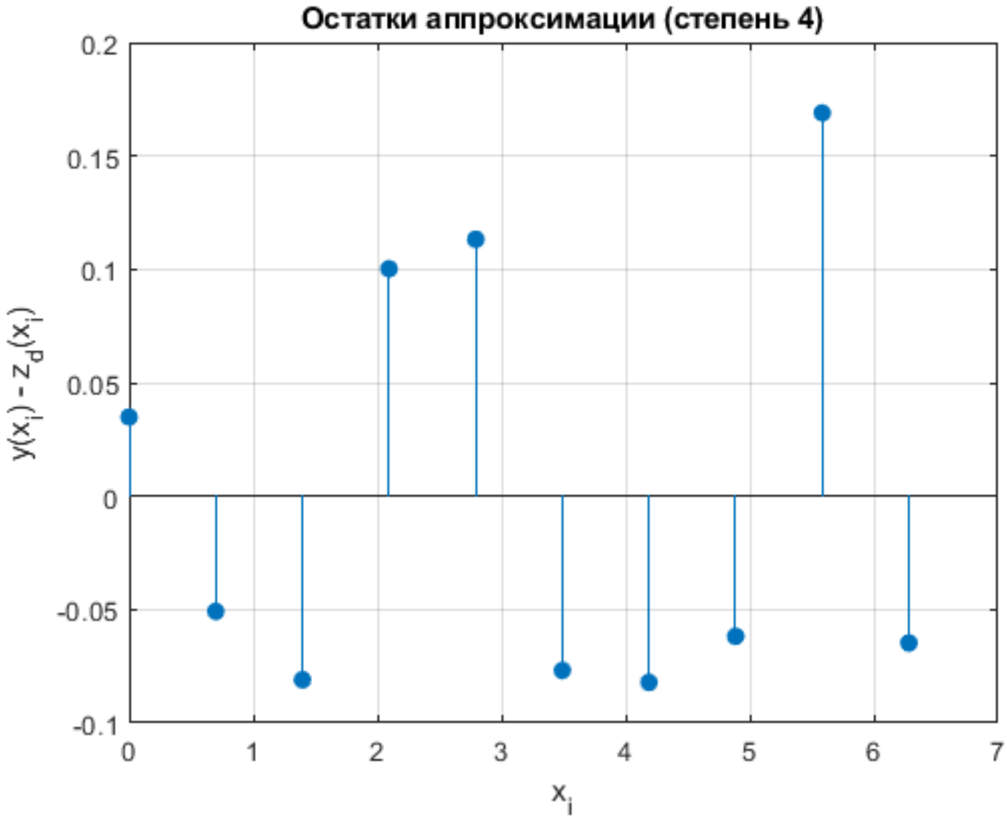
Графики

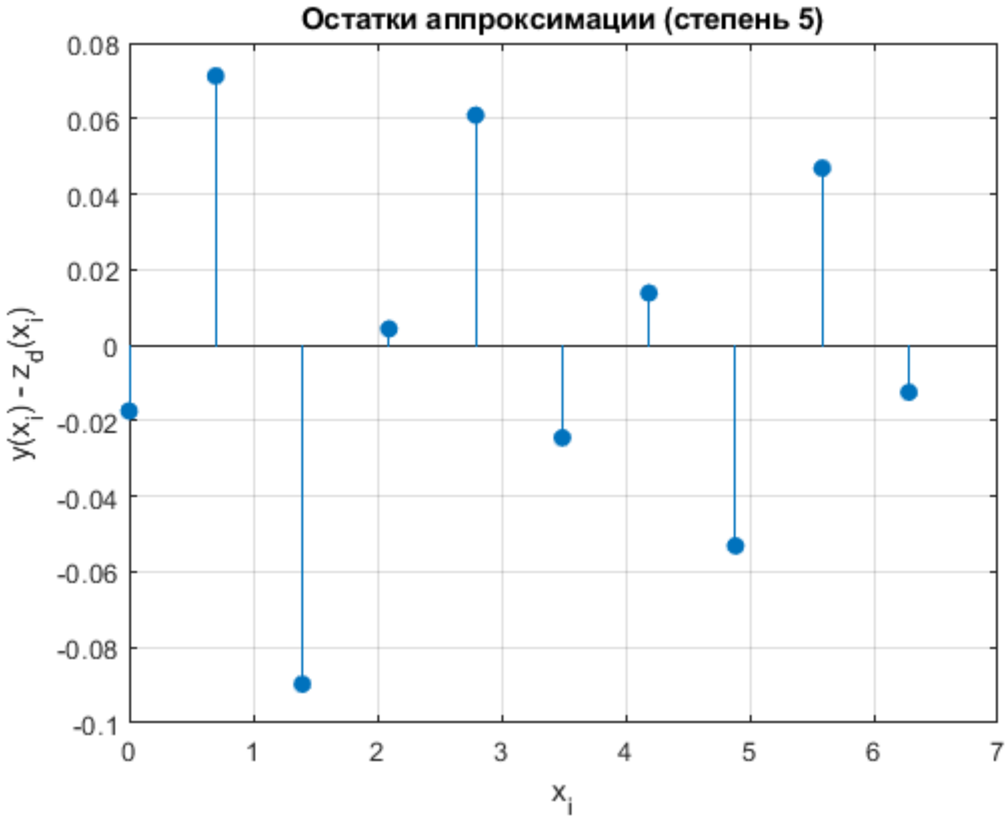




Остатки аппроксимации (степень 2)
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
1
2
3
4
5
6
7
y(x_i) - z_d(x_i)
x_i

Остатки аппроксимации (степень 3)
0.2
0.15
0.1
0.05
0
-0.05
-0.1
0
1
2
3
4
5
6
7
y(x_i) - z_d(x_i)
x_i


Остатки аппроксимации (степень 4)
0.2
0.15
0.1
0.05
0
-0.05
-0.1
0
1
2
3
4
5
6
7
$y(x_i) - z_d(x_i)$
x_i


Остатки аппроксимации (степень 5)
0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.08
-0.1
0
1
2
3
4
5
6
7
$y(x_i) - z_d(x_i)$
x_i

Вывод

В ходе лабораторной работы выполнены все предусмотренные задания для варианта 7. Были:

- подготовлено рабочее окружение MATLAB и введены исходные данные (векторы a, b и матрица A);
- реализован полный набор базовых операций с векторами:;
- вычислена заданная функция $f(x)$ на элементах матрицы A;
- проведена полиномиальная аппроксимация модельных данных, основанных на функции $\sin(x)$ с добавлением шума, с использованием полинома 4-й степени;;
- исследовано влияние степени полинома (от 2 до 5) на характер аппроксимации;
- построены наглядные графики поведения полиномов на мелкой сетке и графики остатков для каждой степени, что позволило визуально проанализировать качество приближения.

Полученные результаты подтвердили корректность реализации алгоритмов в MATLAB и продемонстрировали на практике основные приёмы работы с векторами, матрицами и полиномиальной аппроксимацией.