Неудачный выбор базиса

Неудачный выбор базиса может вызвать большие вычислительные трудности при фактическом построении элемента наилучшего приближения. Приведем соответствующий пример.

.

В линейном пространстве $C[0,1]$ введем скалярное произведение:

$$(f,g) = \int\limits_0^1 f(x)g(x)dx.$$

Рассмотрим в $C[0,1]$ пятимерное подпространство, натянутое на базис, образованный функциями

$$\varphi_0(x) \equiv 1, \quad \varphi_1(x) = x, \quad \varphi_2(x) = x^2, \quad \varphi_3(x) = x^3, \quad \varphi_4(x) = x^4,$$

и найдем наилучшее приближение к функции

$$\varphi(x) = x^5.$$

.

Матрица Грама в этом случае вычисляется элементарно:

$$\int_0^1 \varphi_k(x)\varphi_l(x)dx = \int_0^1 x^{k+l}dx = \frac{1}{k+l+1}, \quad k,\ l = 0,\ 1,\ 2,\ 3,\ 4.$$

Столбец правой части системы также вычисляется элементарно:

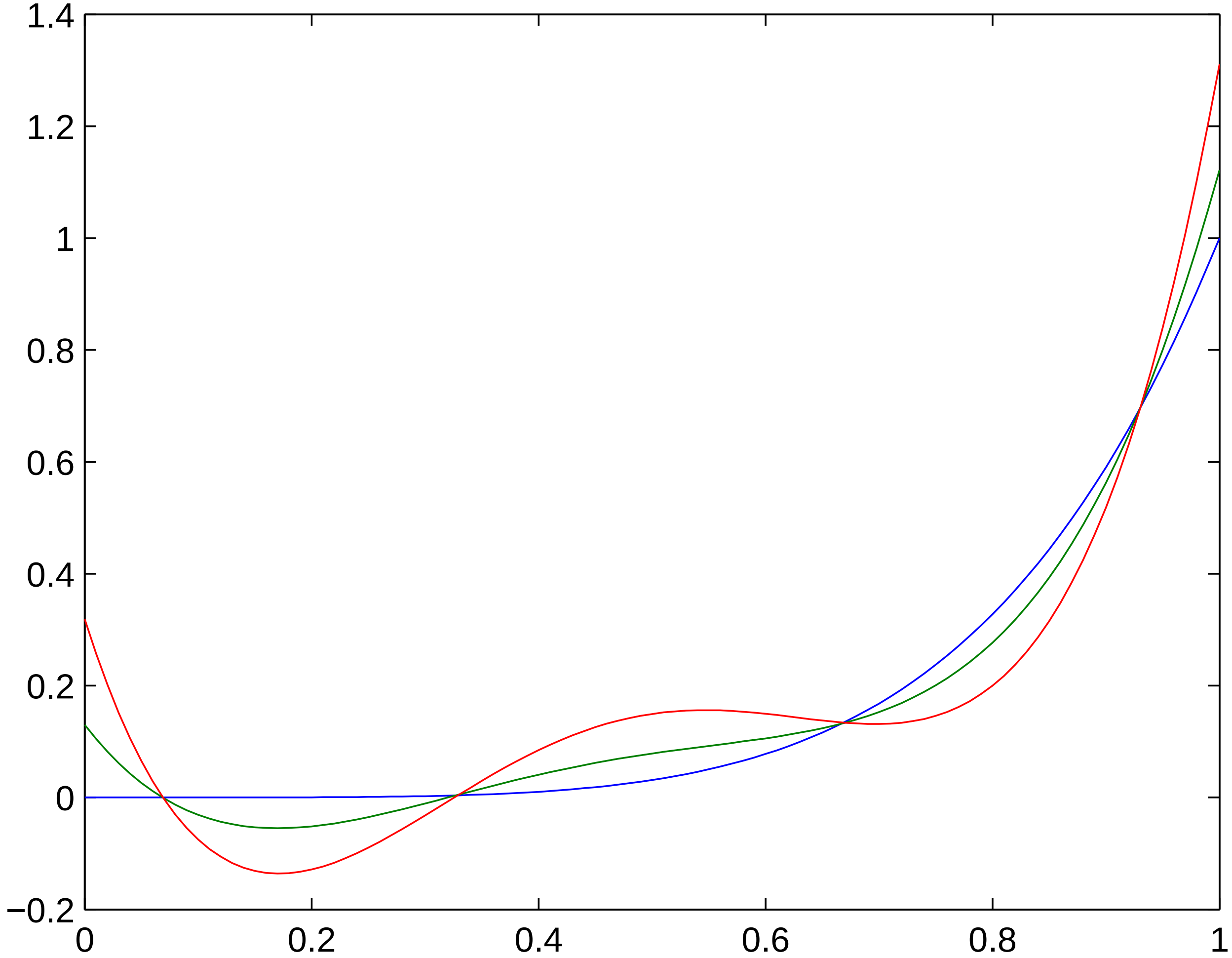
$$\int\limits_0^1 \varphi(x)\varphi_k(x)dx = \int\limits_0^1 x^{5+k}dx = \frac{1}{5+k+1}, \quad k = 0,\ 1,\ 2,\ 3,\ 4.$$

Он, очевидно, состоит из чисел

$$\frac{1}{6},\quad \frac{1}{7},\quad \frac{1}{8},\quad \frac{1}{9},\quad \frac{1}{10}.$$

Будем считать, что при вычислении последнего элемента столбца правой части допущена ошибка, и заменим число

$$\frac{1}{10} \quad \textbf{на} \quad \frac{1}{10} + \varepsilon.$$



blue — функция $\varphi(x) = x^5$,

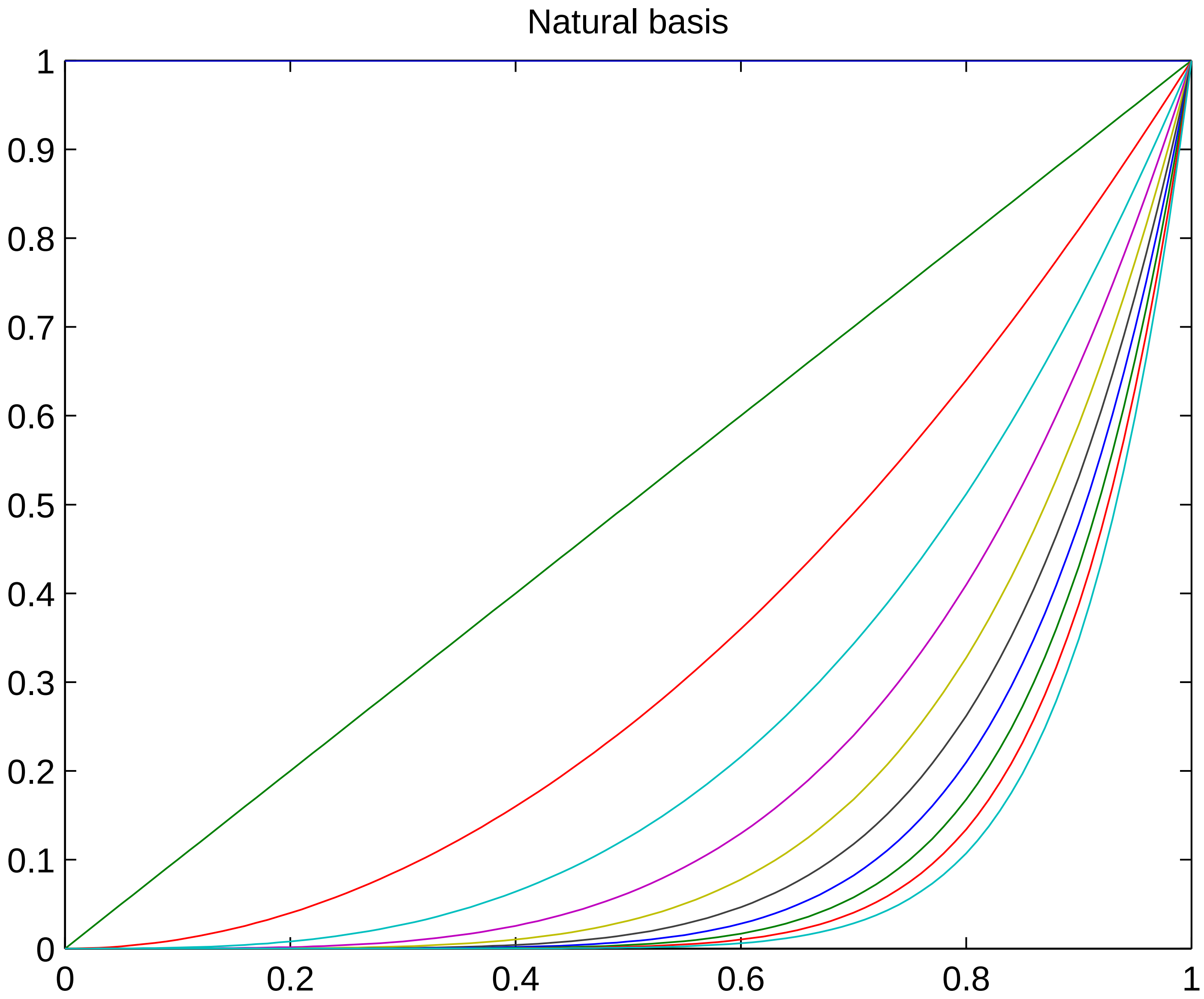
green — график приближающего полинома при $\varepsilon = 2 \cdot 10^{-4}$,

red — график приближающего полинома при $\varepsilon = 5 \cdot 10^{-4}$.

Малым погрешностям, допущенным при вычислении правой части (неизбежным на практике), соответствуют значительные погрешности приближения функции φ.

.

Причина в том, что выбранный нами базис степеней независимой переменной состоит из функций, почти линейно зависимых.



Функции x^p, $p = 0,\ 1,\ 2,\ \ldots,\ 9$, **на отрезке** $[0, 1]$.

.

Матрица системы оказалась в данном случае близкой к вырожденной или, как говорят, *плохо обусловленной*. Матрица

$$H_n = \left\{ \frac{1}{i+j-1} \right\}_{i,j=1}^{n}$$

называется *матрицей Гильберта*. Это классический пример плохо обусловленной матрицы.

.

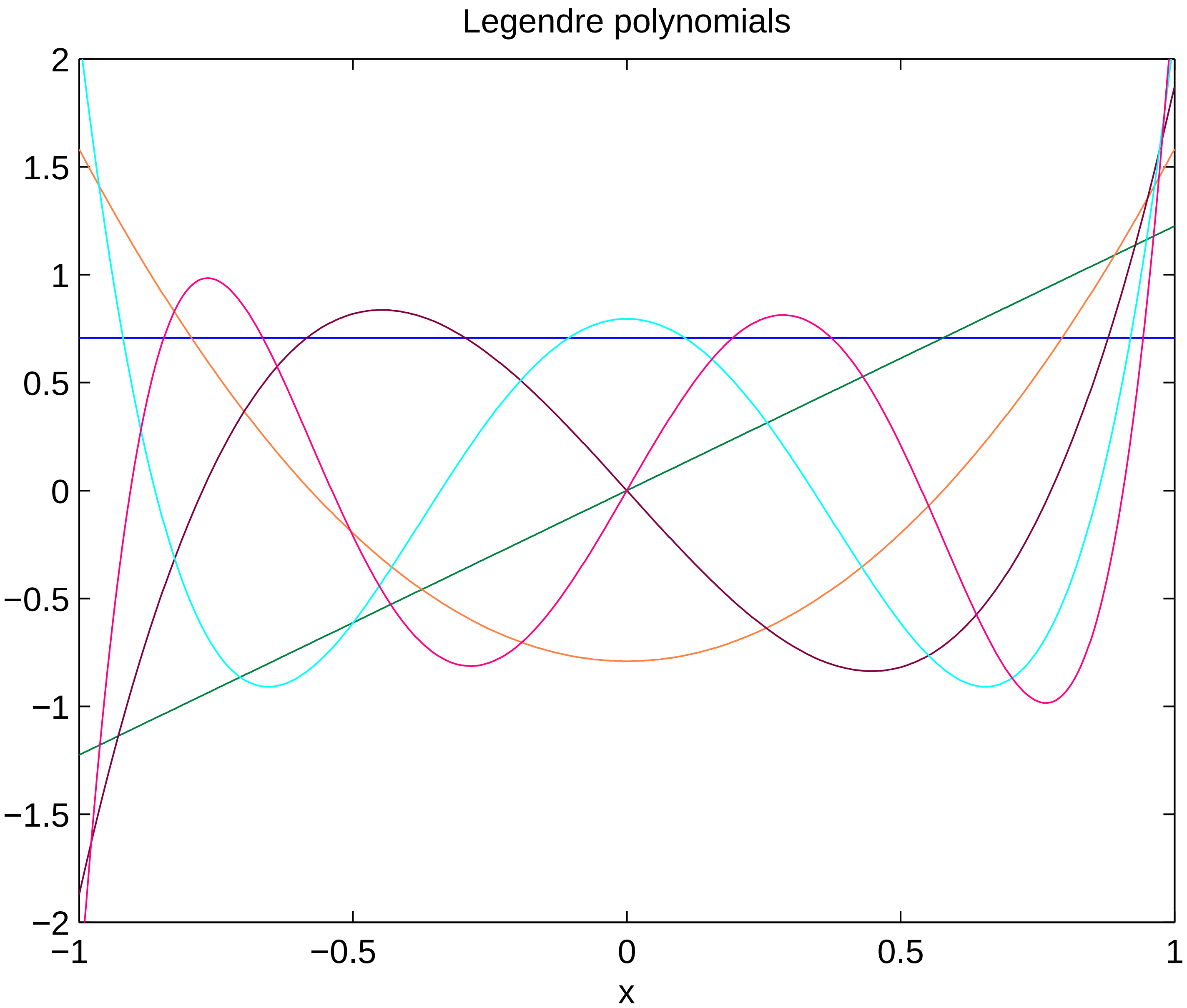
Давид Гильберт (David Hilbert; 1862 — 1943) — немецкий математик.

.

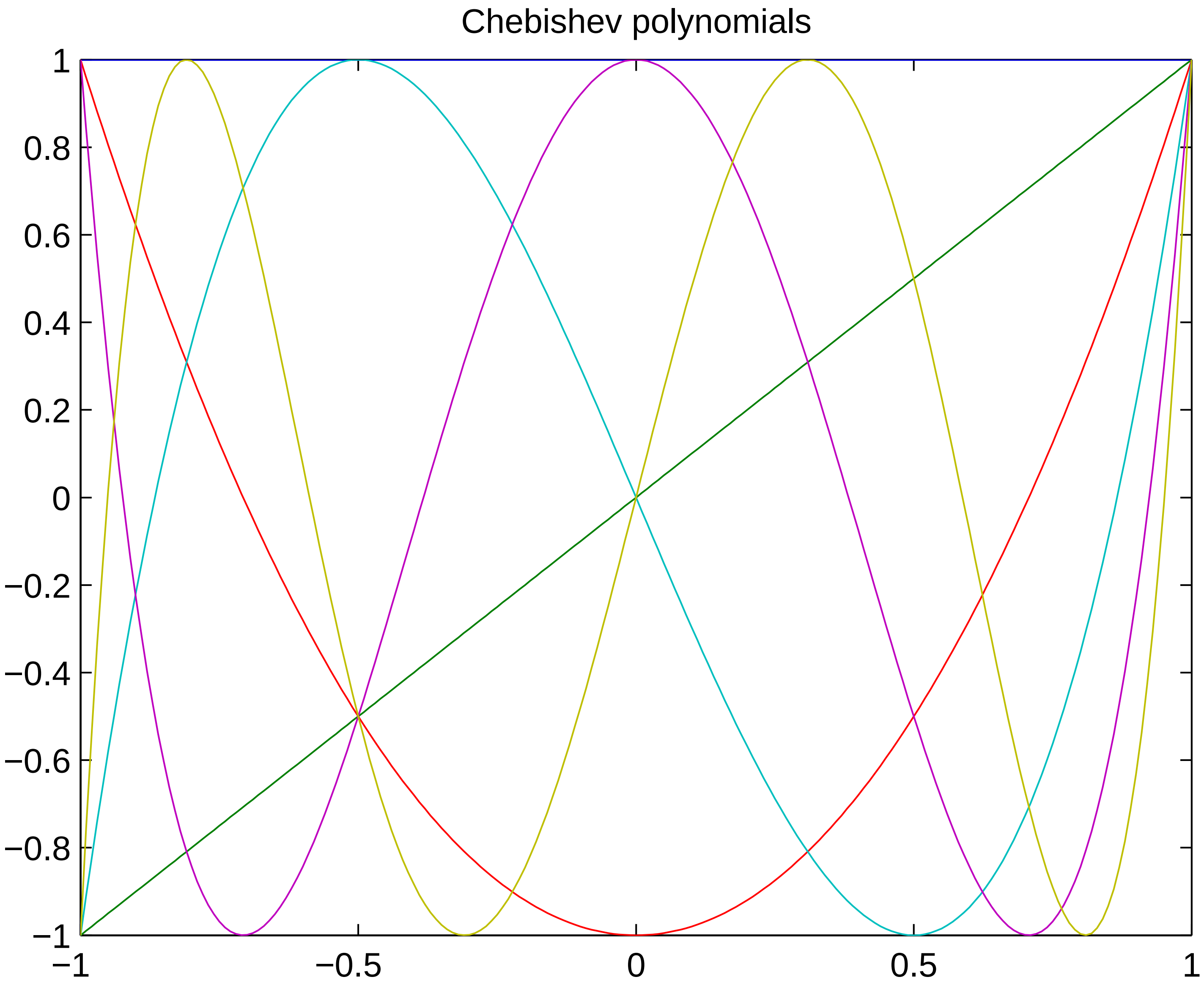
Упражнение. Написать программу и повторить численный эксперимент, описанный выше. Поэкспериментировать с разными n и ε. Написать отчет о результатах численных экспериментов.

.

<u>Замечание.</u> Обычно, приближая функции полиномами, используют ортогональные базисы, например, полиномы Лежандра или Чебышева. В этом случае матрица Грама становится диагональной.



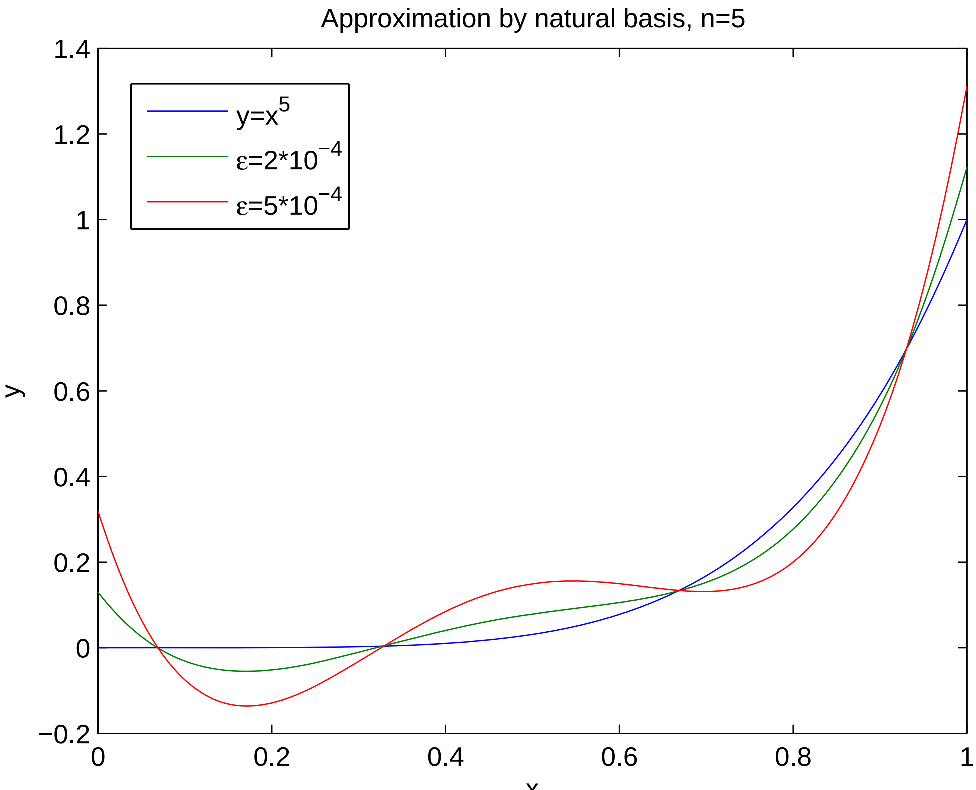
Полиномы Лежандра.



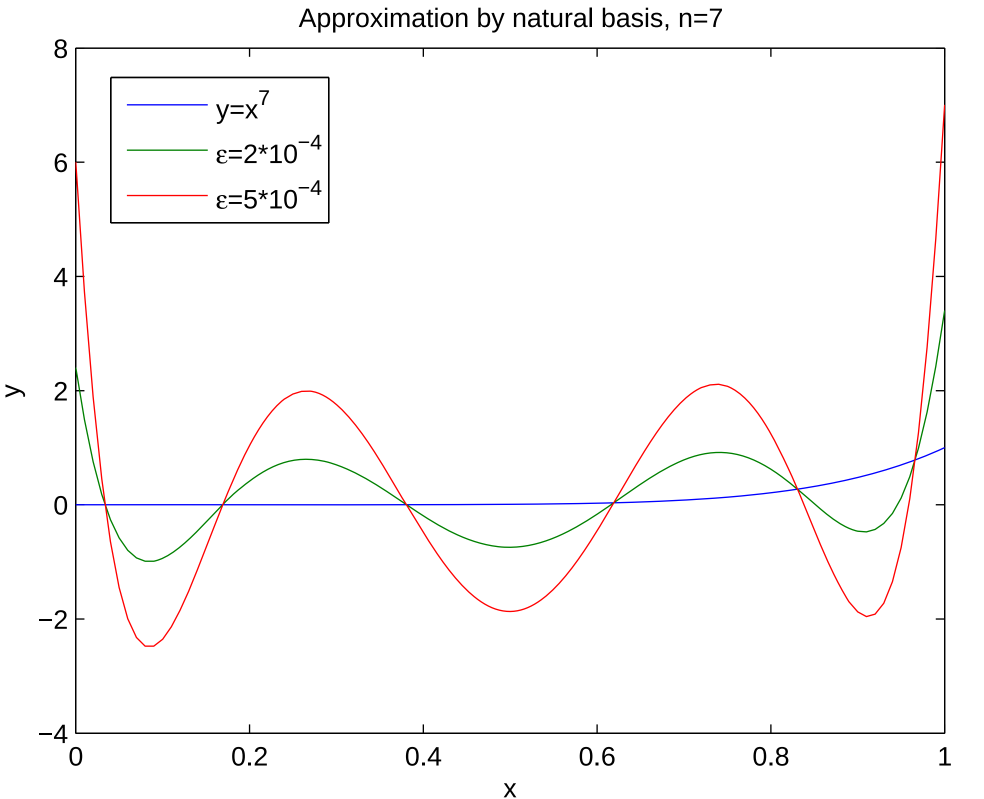
Полиномы Чебышева.

.

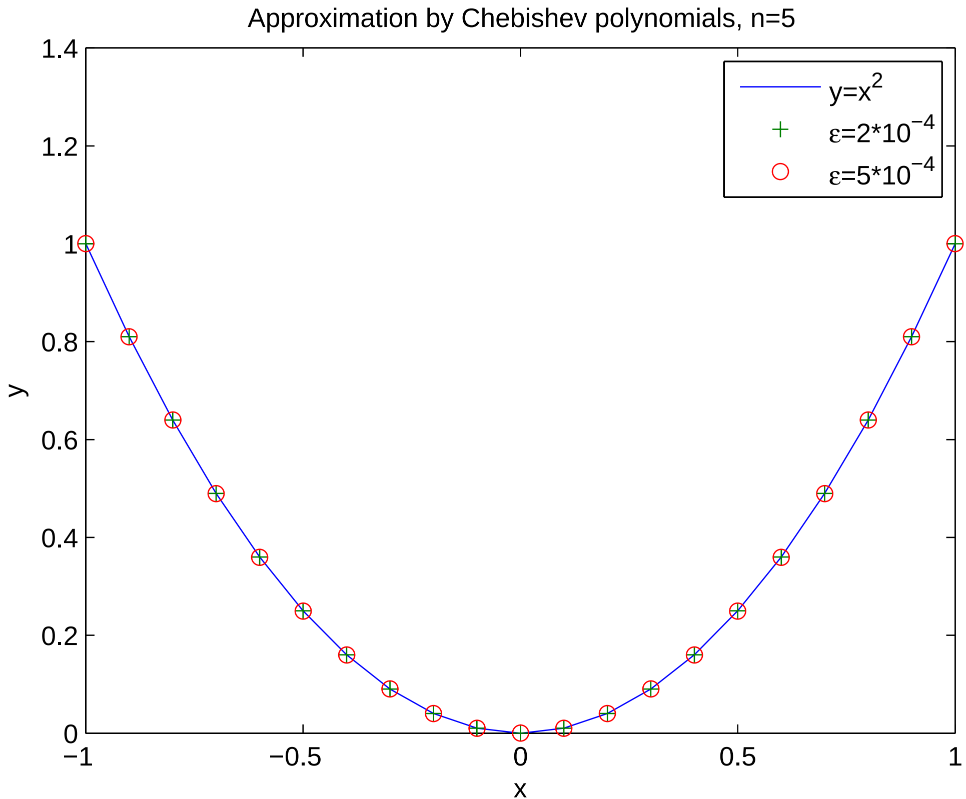
Упражнение. Написать программу для приближения функции полиномами Чебышева. Поэкспериментировать с разными n и ε. Написать отчет о результатах численных экспериментов.



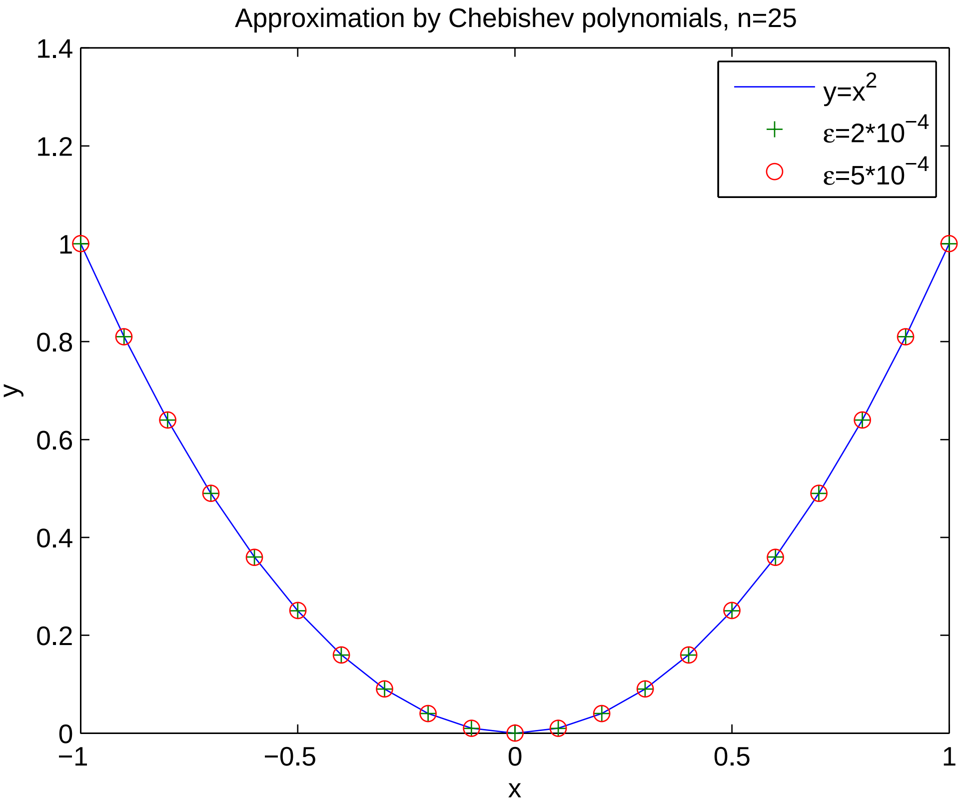
Approximation by natural basis, n=5
y=x^5
ε=2*10^−4
ε=5*10^−4
y
x
1.4
1.2
1
0.8
0.6
0.4
0.2
0
−0.2
0
0.2
0.4
0.6
0.8
1



Approximation by natural basis, n=7
y=x^7
ε=2*10^−4
ε=5*10^−4
8
6
4
2
0
−2
−4
0
0.2
0.4
0.6
0.8
1
x
y



Approximation by Chebishev polynomials, n=5
y=x^2
ε=2*10^−4
ε=5*10^−4
1.4
1.2
1
0.8
0.6
0.4
0.2
0
−1
−0.5
0
0.5
1
x
y



Approximation by Chebishev polynomials, n=25
1.4
1.2
1
0.8
0.6
0.4
0.2
0
y
−1
−0.5
0
0.5
1
x
y=x^2
ε=2*10^−4
ε=5*10^−4

```
clc
clear
n = 7;
A = zeros(n,n);    % Preallocate matrix A
b = zeros(n,1);    % Preallocate vector b
eta1 = zeros(n,1);  % Preallocate vector eta1
eta2 = zeros(n,1);  % Preallocate vector eta2
for k = 1:n
    for l = 1:n
        A(k,l) = 1/(k+l-1);
    end
end
for k = 1:n
    b(k) = 1/(n+k);
end
b(n) = b(n) + 2*10^(-4); % First error of calculations
eta1 = A\b;
b(n) = b(n) + 3*10^(-4); % Second error of calculations
eta2 = A\b;
x = 0:0.01:1;
varphi = x.^n;
approx1 = 0;
approx2 = 0;
for k = 1:n
    approx1 = approx1 + eta1(k)*x.^(k-1);
    approx2 = approx2 + eta2(k)*x.^(k-1);
end
plot(x,varphi,x,approx1,x,approx2)
legend('y=x^7','\epsilon=2*10^{-4}','\epsilon=5*10^{-4}')
xlabel('x')
ylabel('y')
title 'Approximation by natural basis, n=7'
%axis([0 1 -0.4 1])
```

```
clc
clear
global k
k = 0;
n = 24;
A = zeros(n,1);    % Preallocate vector A
b = zeros(n,1);    % Preallocate vector b
eta1 = zeros(n,1);  % Preallocate vector eta1
eta2 = zeros(n,1);  % Preallocate vector eta2
for k = 0:n
    k
    A(k+1) = quad8('T_basis_k(x).*T_basis_k(x).*(sqrt(1-x.^2)).^(-1)',-1,1,1e-5);
    b(k+1) = quad8('T_basis_k(x).*(sqrt(1-x.^2)).^(-1).*x.^2',-1,1,1e-5);
end
b(n+1) = b(n+1) + 2*10^(-4);  % First error of calculations
eta1 = b./A;
b(n+1) = b(n+1) + 3*10^(-4);  % Second error of calculations
eta2 = b./A;
x = -1:0.1:1;
varphi = x.^2;
approx1 = 0;
approx2 = 0;
for k = 0:n
    approx1 = approx1 + eta1(k+1)*T_basis_k(x);
    approx2 = approx2 + eta2(k+1)*T_basis_k(x);
end
plot(x,varphi,x,approx1,'+',x,approx2,'o')
legend('y=x^2','\epsilon=2*10^{-4}','\epsilon=5*10^{-4}')
xlabel('x')
ylabel('y')
title 'Approximation by Chebishev polynomials, n=25'
axis([-1 1 0 1.4])
```

```
function z=T_basis_k(x)
global k
z=cos(k*acos(x));
```