МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ «ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

ОТЧЕТ

по практической работе №4

по дисциплине «Вычислительная математика»

Тема: Интерполирование функций

Студент гр. 0303 _________________ Калмак Д.А.

Преподаватель _________________ Сучков А.И.

Санкт-Петербург

2021

Цель работы.

Научиться применять интерполирование функции для решения практических задач, овладеть навыками применения интерполяционных формул Лагранжа заданной степени, многочленов Ньютона. Научиться оценивать погрешности интерполяционных формул и работать в программных пакетах с целью проверки полученных результатов.

Основные теоретические положения.

Пусть значение f(x) известно в некоторых точках $X = \{xj\}_{j=0}^{n}$, и необходимо найти $f(x_i)$: $x_i \notin X$. Для этих целей, функцию $f(x)$ приближают функцией $L_n(x)$: $Ln(x) = \sum_{k=0}^{n} ak\varphi k$, где φ – произвольный базис, удобный для данной $f(x)$. Задача интерполяции – найти обобщённый многочлен. Существует несколько способов нахождения, например, метод Лагранжа. Он даёт готовый интерполяционный многочлен Лагранжа: $Ln(x) = \sum_{i=0}^{n} fili(x)$, где $f_i = f(x_i)$ – значение функции в узле x_i, а $\ell_i(x) = \prod_{\substack{k=0 \\ k \neq i}}^{n} \frac{x-xk}{xi-xk}$ – i-ый базисный полином.

Если узлы, в которых определено значение $f(x_i)$ являются равноотстоящими, т.е. $x_i = x_0+ih$, $x_0 < x_1 < \cdots < x_n$, $i = 1..n$, тогда можно воспользоваться интерполяционным многочленом Ньютона: $Nn(x) = \sum_{k=0}^{n} \frac{\Delta kf0}{k!}$, где $\Delta^k f$ – конечная разность k-го порядка, $q = (x-x0)/h$.

Многочлен Чебышёва первого рода $T_n(x)$ характеризуется как многочлен степени n со старшим коэффициентом 2^{n-1}, который меньше всего отклоняется от нуля на отрезке $[-1,1]$ $T_n(x) = cos(n*arccosx)$.

Для натурального n узлы на промежутке $x \in [-1,1]$ задаются формулой: $xk = cos(\pi \frac{2k-1}{2n})$, $k = 1..n$. Это корни многочлена Чебышёва первого рода степени n.

Для получения узлов на произвольном отрезке $[a, b]$, можно применить следующую формулу: $xk = \frac{a+b}{2} + \frac{b-a}{2} cos(\pi \frac{2k-1}{2n})$, $k = 1..n$. После нахождения интерполяционного многочлена, необходимо вычислить и оценить его погрешность. Должно выполнятся следующее неравенство: $\max_{x\in[a,b]}|Rn(x)| \leq \frac{Mn+1}{(n+1)!}\max_{x\in[a,b]}|wn+1(x)| = Qn,$ где *[a,b]* – промежуток интерполирования, $R_n(x) = f(x) - L_n(x)$, $M_{n+1} = \max_{\eta\in[a,b]}\left|f^{(n+1)}(\eta)\right|$, $w_{n+1}(x) = \prod_{j=0}^{n}(x - xj)$. Левая часть неравенства является практической погрешностью, а правая – теоретической.

Постановка задачи.

Построить интерполяционный многочлен по 2, 3, 4, 5 и 6 узлам (равноотстоящим и чебышёвским) для функции $f(x) = \frac{A}{x^2+px+q}$ на промежутке *[a,b]* по равноотстоящим и по чебышёвским узлам. Найти фактическую погрешность и сравнить её с теоретической оценкой.

Выполнение работы.

По условию:

a = -1, b = 6, A = 1000, p = -6, q = 56.

Тогда, если подставить значения, то $f(x) = \frac{1000}{x^2-6x+56}$.

Была реализована функция f(), которая вычисляет значения в функции f(x). Также была реализована функция df(), которая вычисляет n-ую производную функции *f(x)*. Функция реализована с помощью оператора switch.

Реализована функция lagrange(), которая вычисляет интерполяционный многочлен n-го порядка по методу Лагранжа.

Разработанный код см. в Приложении А.

Определим полиномы по методу Лагранжа для равноотстоящих узлов:

n = 2: L = 0.2834467x + 16.1564626

n = 3: L = $-0.3509340x^2$ *+ 2.0381168x + 18.2620667*

n = 4: L = $-0.0128554x^3 - 0.2450524x^2$ *+ 1.9072258x + 18.0124387*

n = 5: L = $0.0067326x^4 - 0.0795252x^3 - 0.0894428x^2 + 1.9504076x + 17.8266085$

n=6: $0.0003817x^5+0.0017316x^4-0.0592505x^3-0.1105098x^2+1.9283505x+17.8512756$

Были построены графики исходной функции и интерполяционного многочлена для равноотстоящих узлов. (см. рис. 1-5).

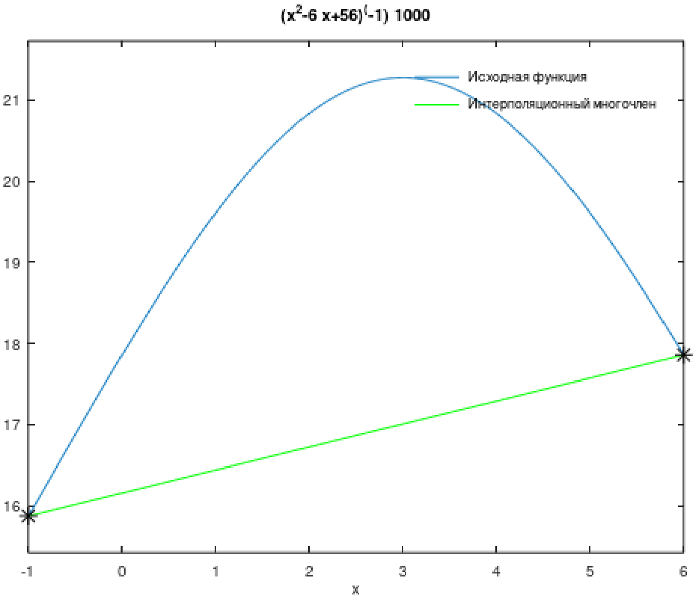

Рисунок 1 – Графики исходной функции и интерполяционного многочлена для равноотстоящих узлов при n = 2

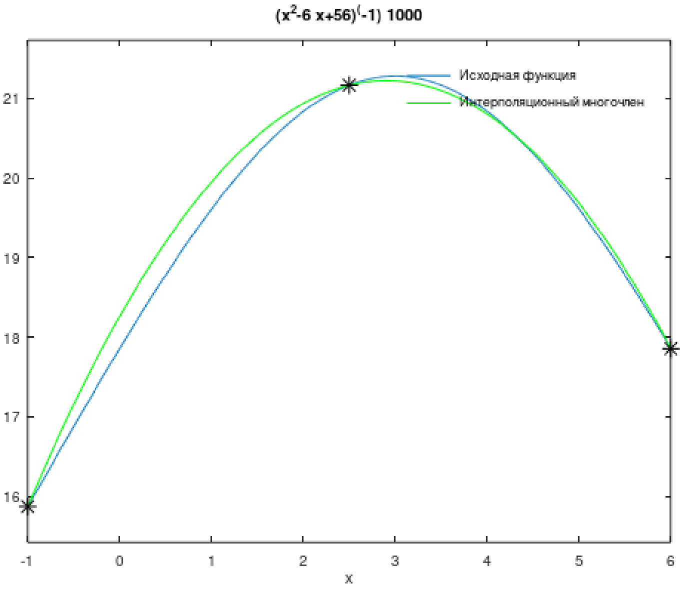

Рисунок 2 – Графики исходной функции и интерполяционного многочлена для равноотстоящих узлов при n = 3

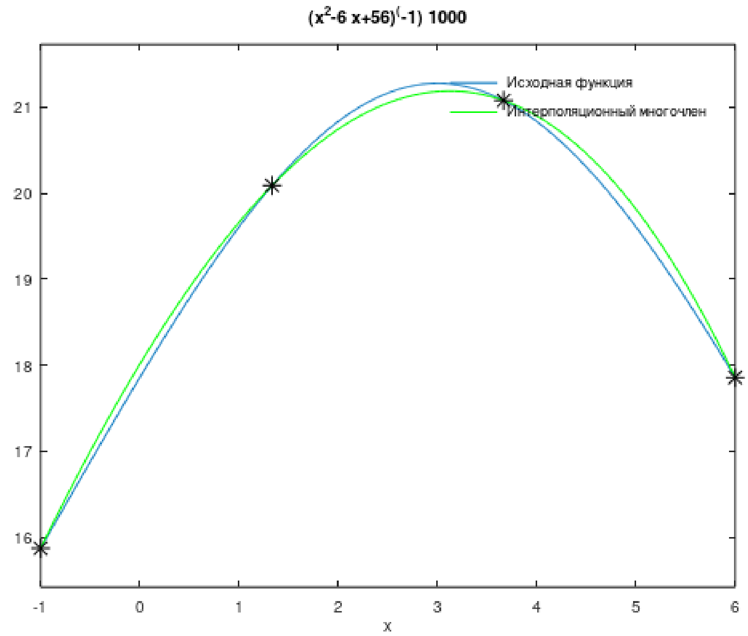

Рисунок 3 – Графики исходной функции и интерполяционного многочлена для равноотстоящих узлов при n = 4

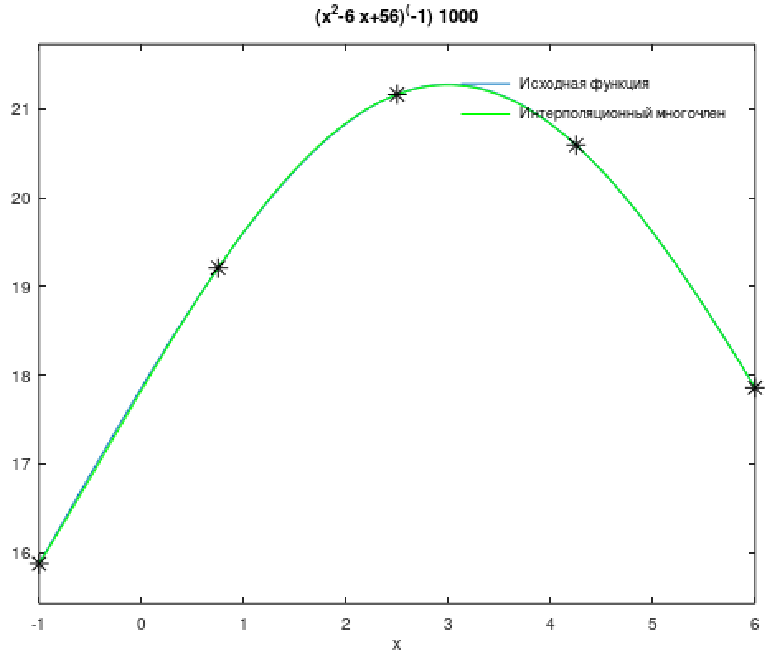

Рисунок 4 – Графики исходной функции и интерполяционного многочлена для равноотстоящих узлов при n = 5

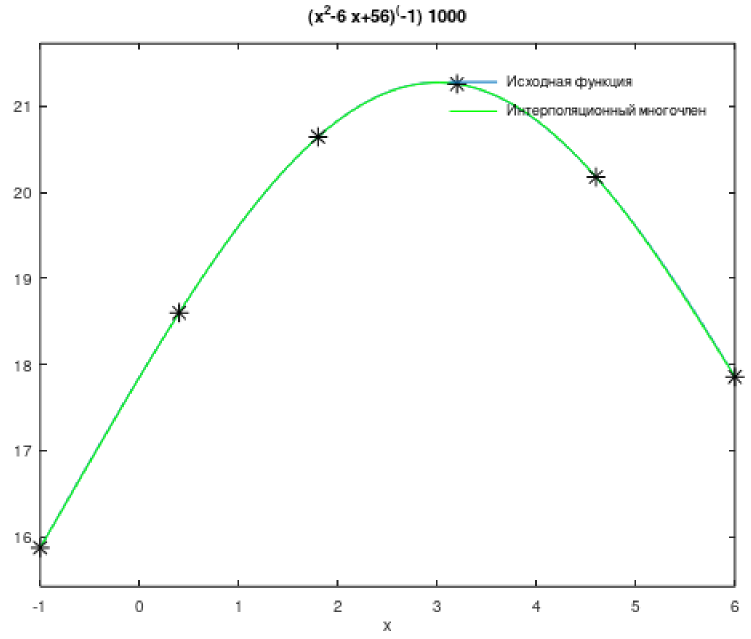

Рисунок 5 – Графики исходной функции и интерполяционного многочлена для равноотстоящих узлов при n = 6

Погрешности интерполяции с помощью равноотстоящих узлов представлены в табл. 1. С увеличением n, количества узлов, теоретическая и фактическая погрешности уменьшаются и стремятся к нулю. Происходит это по причине лучшего приближения к исходной функции при большем числе узлов.

Таблица 1 – Погрешности интерполяции для равноотстоящих узлов

Значение n	2	3	4	5	6
Значение Mn+1	0.308273	0.231095	0.141081	0.078304	0.131213
Значение max|ωn+1(x)|	0	16.502407	16.656601	59.582037	37.989616
Значение (n+1)!	6	24	120	720	5040
Значение Qn	0	0.158901	0.019583	0.006480	0.000989
Значение max|Rn(x)|	4.314220	0.069832	0.116087	0.039042	0.017332

Определим полиномы по методу Лагранжа для чебышевских узлов:

$n = 2: L = 0.3517701x + 17.8963048$

$n = 3: L = -0.3694203x^2 + 2.1619627x + 18.0679914$

$n = 4: L = -0.0132543x^3 - 0.2493885x^2 + 1.9352953x + 17.9583526$

$n = 5$: $L = 0.0067140x^4 - 0.0792649x^3 - 0.0894557x^2 + 1.9430902x + 17.8416431$

$n = 6$: $L = 0.0003688x^5 + 0.0017367x^4 - 0.0582773x^3 - 0.1132913x^2 + 1.9258893x + 17.8568255$

Были построены графики исходной функции и интерполяционного многочлена для чебышевских узлов. (см. рис. 6-10).

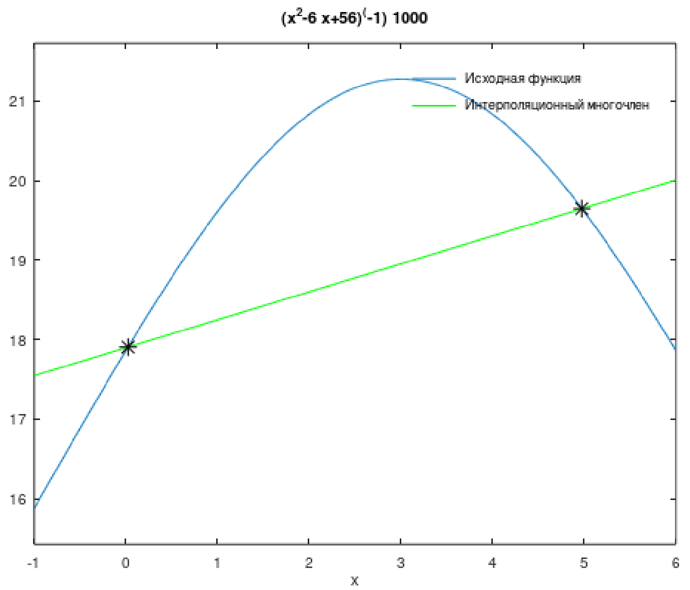

Рисунок 6 – Графики исходной функции и интерполяционного многочлена для чебышевских узлов при n = 2

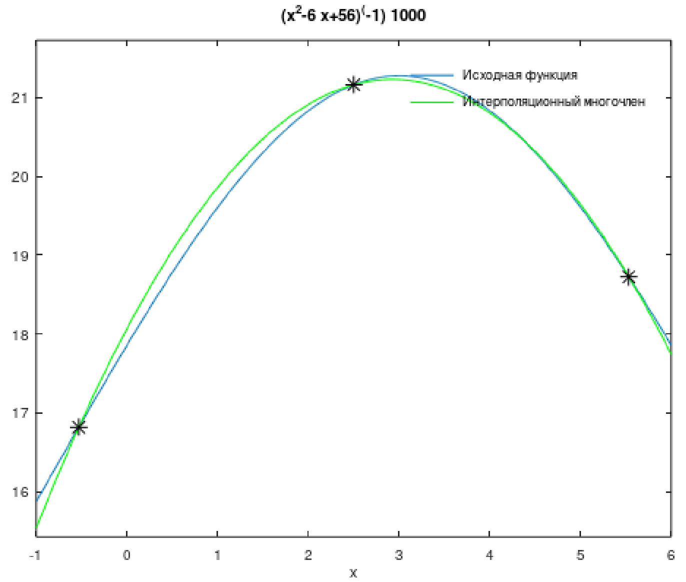

Рисунок 7 – Графики исходной функции и интерполяционного многочлена для чебышевских узлов при n = 3

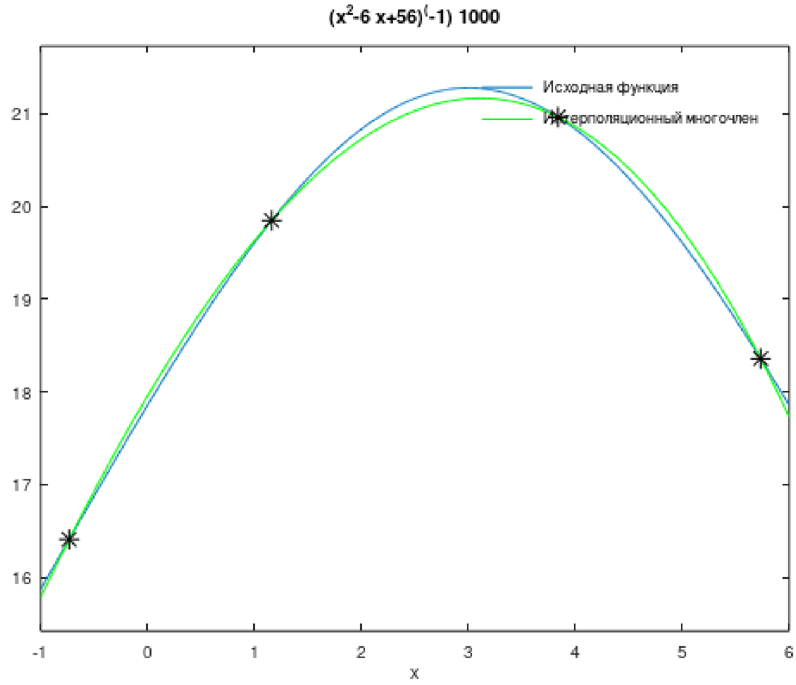

Рисунок 8 – Графики исходной функции и интерполяционного многочлена для чебышевских узлов при n = 4

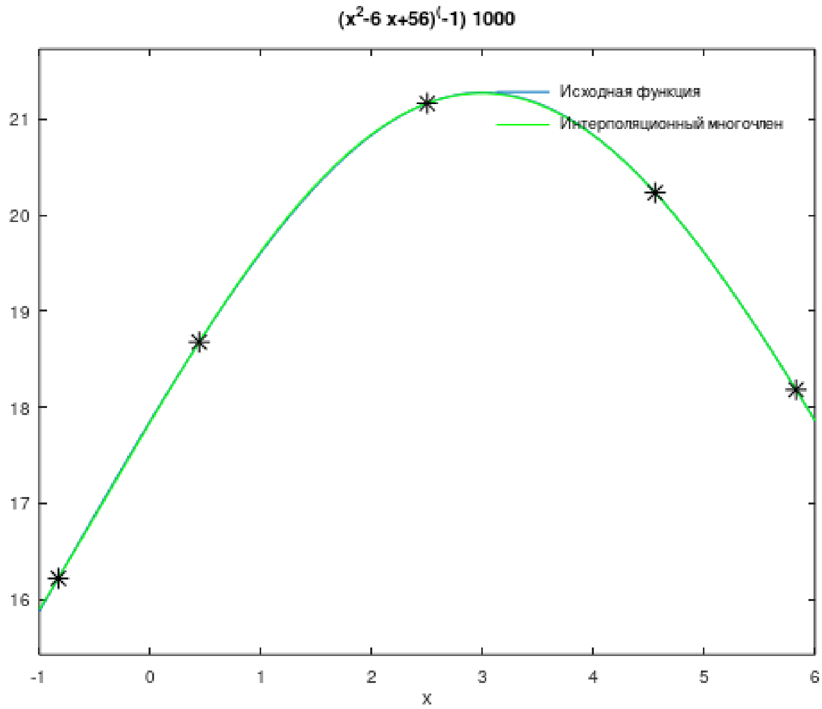

Рисунок 9 – Графики исходной функции и интерполяционного многочлена для чебышевских узлов при n = 5

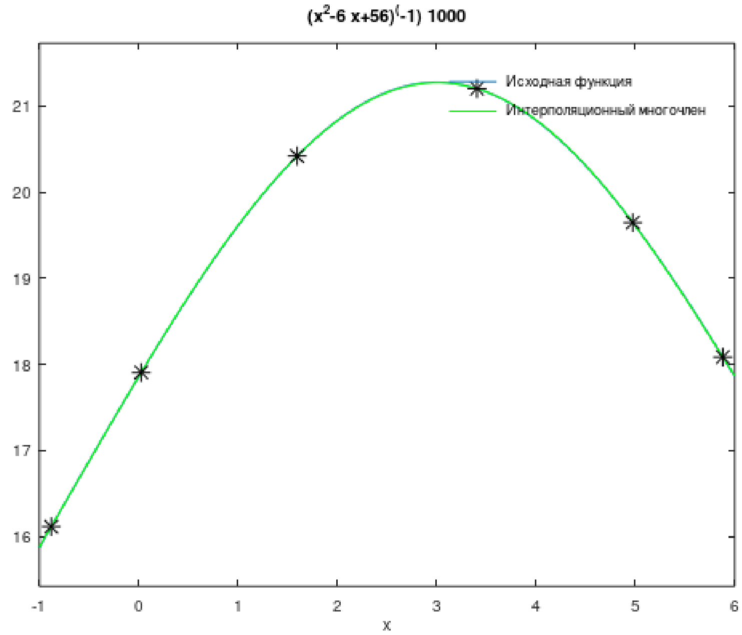

Рисунок 10 – Графики исходной функции и интерполяционного многочлена для чебышевских узлов при n = 6

Погрешности интерполяции с помощью чебышевских узлов представлены в табл. 2. Фактическая и теоретическая погрешности уменьшаются при увеличении n. Значения погрешности стремятся к нулю. Причиной является увеличение точности приближения с увеличением количества узлов.

Таблица 2 – Погрешности интерполяции для чебышевских узлов

Значение n	2	3	4	5	6
Значение Mn+1	0.308273	0.231095	0.141081	0.078304	0.131213
Значение max|ωn+1(x)|	6.125000	10.718750	18.757813	32.826172	57.445801
Значение (n+1)!	6	24	120	720	5040
Значение Qn	0.314696	0.103210	0.022053	0.003570	0.001496
Значение max|Rn(x)|	2.393535	0.336408	0.133380	0.021361	0.007612

Выводы.

Таким образом, были получены навыки применения интерполяционных формул Лагранжа заданной степени. Были получены интерполяционные многочлены для равноотстоящих и чебышевских узлов при разном количестве, а также получены фактические погрешности. Практические и теоретические погрешности уменьшаются с ростом количества узлов. Погрешности стремятся к нулю.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Файл lagrangefunction.m

```
function lagrangefunction()
  format long g
  a = -1; b = 6; n = 6;
  h = (b-a)/(n-1);
  x=a:h:b;
  y = ((x.^2-6*x+56).**(-1))*1000;
  xp = linspace(a,b,100);
  res = lagrange(x);
  yp = polyval(res,xp);
  # Исходная функция
  ezplot("(x.^2-6*x+56).^(-1)*1000", [a,b])
  hold on
  # Интерполяционный многочлен
  plot(xp,yp,'g');
  # Узлы
  plot(x,y,'k*');
  legend("Исходная функция","Интерполяционный многочлен")
  legend('boxoff')
  hold off

  x1 = linspace(a,b,100);
  r = 1:length(x1);
  r(1:length(r))=1;
  for i=1:length(x1)
    r(i) = df(n+1,x1(i));
  endfor
  mn_1 = max(r);
  #fprintf("mn_1 = %f\n", mn_1)


  x2 = linspace(a,b,100);
  r = 1:length(x2);
  r(1:length(r))=1;
  for i=1:length(x2)
    x2_1 = x2(i);
    for j=1:n
      r(i) *= (x2_1-x(j));
    endfor
  endfor
  wn_1 = max(r);
  #fprintf("wn_1 = %f\n", wn_1)

  rnx = max(f(xp)-yp);
  #fprintf("rnx = %f\n", rnx)

  qn = mn_1/factorial(n+1)*wn_1;
  #fprintf("qn = %f\n", qn)

endfunction

function res = lagrange(x)
```

```
  res = 0;
  for k = 1:length(x)
    p = 1;
    for i = 1:length(x)
      if i~=k
        p = conv(p,[1, -x(i)])/(x(k)-x(i));
      endif
    end
    res += f(x(k))*p;
  end
endfunction

function f = f(x)
  f = (x.^2-6*x+56).**(-1)*1000;
endfunction

function df = df(n,x)
  switch(n)
    case(3)
      df = -24000*(-1 + 2*(-3 + x).^2/(56 + x.^2 - 6*x))*(-3 + x)/(56
+ x.^2 - 6*x).^3;
    case(4)
      df = 24000*(1 - 12*(-3 + x).^2/(56 + x.^2 - 6*x) + 16*(-3 +
x).^4/(56 + x.^2 - 6*x).^2)/(56 + x.^2 - 6*x).^3;
    case(5)
      df = -240000*(-3 + x)*(3 - 16*(-3 + x).^2/(56 + x.^2 - 6*x) +
16*(-3 + x).^4/(56 + x.^2 - 6*x).^2)/(56 + x.^2 - 6*x).^4;
    case(6)
      df = 720000*(-1 - 80*(-3 + x).^4/(56 + x.^2 - 6*x).^2 + 24*(-3 +
x).^2/(56 + x.^2 - 6*x) + 64*(-3 + x).^6/(56 + x.^2 - 6*x)^3)/(56 +
x^2 - 6*x).^4;
    case(7)
      df = -40320000*(-3 + x)*(-1 - 24*(-3 + x).^4/(56 + x.^2 -
6*x).^2 + 10*(-3 + x).^2/(56 + x.^2 - 6*x) + 16*(-3 + x).^6/(56 + x.^2
- 6*x).^3)/(56 + x.^2 - 6*x).^5;
  endswitch
endfunction
```

Файл chebyshevfunction.m

```
function chebyshevfunction()
  format long g
  a = -1; b = 6; n = 2;
  k = 1:n;
  x = (a+b)/2 + (b-a)/2*cos(pi*(2*k-1)/(2*n));
  y = ((x.^2-6*x+56).**(-1))*1000;
  xp = linspace(a,b,100);
  res = lagrange(x);
  yp = polyval(res,xp);
  # Исходная функция
  ezplot("(x.^2-6*x+56).^(-1)*1000", [a,b])
  hold on
  # Интерполяционный многочлен
  plot(xp,yp,'g');
  # Узлы
  plot(x,y,'k*');
```

```
  legend("Исходная функция","Интерполяционный многочлен")
  legend('boxoff')
  hold off

  x1 = linspace(a,b,100);
  r = 1:length(x1);
  r(1:length(r))=1;
  for i=1:length(x1)
    r(i) = df(n+1,x1(i));
  endfor
  mn_1 = max(r);
  #fprintf("mn_1 = %f\n", mn_1)


  x2 = linspace(a,b,100);
  r = 1:length(x2);
  r(1:length(r))=1;
  for i=1:length(x2)
    x2_1 = x2(i);
    for j=1:n
      r(i) *= (x2_1-x(j));
    endfor
  endfor
  wn_1 = max(r);
  #fprintf("wn_1 = %f\n", wn_1)

  rnx = max(f(xp)-yp);
  #fprintf("rnx = %f\n", rnx)

  qn = mn_1/factorial(n+1)*wn_1;
  #fprintf("qn = %f\n", qn)

endfunction

function res = lagrange(x)
  res = 0;
  for k = 1:length(x)
    p = 1;
    for i = 1:length(x)
      if i~=k
        p = conv(p,[1, -x(i)])/(x(k)-x(i));
      endif
    end
    res += f(x(k))*p;
  end
endfunction

function f = f(x)
  f = (x.^2-6*x+56).**(-1)*1000;
endfunction

function df = df(n,x)
  switch(n)
    case(3)
      df = -24000*(-1 + 2*(-3 + x).^2/(56 + x.^2 - 6*x))*(-3 + x)/(56
+ x.^2 - 6*x).^3;
    case(4)
```

```
      df = 24000*(1 - 12*(-3 + x).^2/(56 + x.^2 - 6*x) + 16*(-3 +
x).^4/(56 + x.^2 - 6*x).^2)/(56 + x.^2 - 6*x).^3;
    case(5)
      df = -240000*(-3 + x)*(3 - 16*(-3 + x).^2/(56 + x.^2 - 6*x) +
16*(-3 + x).^4/(56 + x.^2 - 6*x).^2)/(56 + x.^2 - 6*x).^4;
    case(6)
      df = 720000*(-1 - 80*(-3 + x).^4/(56 + x.^2 - 6*x).^2 + 24*(-3 +
x).^2/(56 + x.^2 - 6*x) + 64*(-3 + x).^6/(56 + x.^2 - 6*x)^3)/(56 +
x^2 - 6*x).^4;
    case(7)
      df = -40320000*(-3 + x)*(-1 - 24*(-3 + x).^4/(56 + x.^2 -
6*x).^2 + 10*(-3 + x).^2/(56 + x.^2 - 6*x) + 16*(-3 + x).^6/(56 + x.^2
- 6*x).^3)/(56 + x.^2 - 6*x).^5;
  endswitch
endfunction
```