

Учреждение образования
«Гомельский государственный университет
имени Франциска Скорины»

Е. М. БЕРЕЗОВСКАЯ

**МЕТОДЫ ВЫЧИСЛЕНИЙ:
ПРИБЛИЖЕНИЕ ФУНКЦИЙ**

Практическое пособие

для студентов специальности
6-05-0533-11 «Прикладная информатика»

Гомель
ГГУ им. Ф. Скорины
2025

УДК 519.651(076)
ББК 22.192.24я73
Б484

Рецензенты:

кандидат физико-математических наук В. И. Мироненко,
кандидат физико-математических наук С. П. Новиков

Рекомендовано к изданию научно-методическим советом
учреждения образования «Гомельский государственный
университет имени Франциска Скорины»

Березовская, Е. М.

Б484 Методы вычислений: приближение функций : практическое
пособие / Е. М. Березовская ; Гомельский гос. ун-т им.
Ф. Скорины. – Гомель : ГГУ им. Ф. Скорины, 2025. – 41 с.
ISBN 978-985-32-0126-0

Практическое пособие содержит теоретический материал по вопросам теории интерполирования функций, равномерного и среднеквадратичного приближения функций, примеры программной реализации задач, задания для индивидуального выполнения, вопросы для самоконтроля. Издание призвано оказать помощь студентам в овладении и закреплении базовых знаний в области численных методов.

Адресовано студентам специальности 6-05-0533-11 «Прикладная информатика» (профилизация Программное обеспечение компьютерных систем).

УДК 519.651(076)
ББК 22.192.24я73

ISBN 978-985-32-0126-0

© Березовская Е. М., 2025
© Учреждение образования
«Гомельский государственный университет
имени Франциска Скорины», 2025

ОГЛАВЛЕНИЕ

Предисловие.....	4
Тема 1. Приближение функций.....	5
1.1 Интерполирование функций.....	5
1.1.1 Пример программной реализации многочлена Лагранжа..	6
1.2 Интерполирование для равноотстоящих узлов.....	9
1.2.1 Пример программной реализации многочлена Ньютона..	11
1.3 Интерполирование для неравноотстоящих узлов.....	13
1.4 Многочлены Чебышева.....	14
1.5 Обратное интерполирование.....	15
1.6 Интерполирование функций многих переменных.....	15
1.6.1 О проблеме интерполирования функций многих пе-	
ременных.....	15
1.6.2 Интерполяционная формула Ньютона.....	17
1.7 Численное дифференцирование.....	17
1.7.1 Погрешность формул численного дифференцирования..	18
1.8 Приближение функций сплайнами.....	19
1.8.1 Пример программной реализации кубического сплайна..	23
1.9 Задания по интерполированию функций.....	25
Тема 2. Метод наименьших квадратов.....	29
2.1 Основные понятия.....	29
2.2 Класс алгебраических многочленов.....	31
2.3 Точечная среднеквадратичная аппроксимация функций на	
отрезке.....	33
2.3.1 Пример программной реализации точечного метода..	34
2.4 Задания по среднеквадратическому приближению	
функций.....	38
Литература.....	41

ПРЕДИСЛОВИЕ

Характерной особенностью современного этапа образования и развития производства следует считать резкое повышение требований к квалификации работников производства, управления и образования. Содержание образовательного процесса в вузах достаточно жестко определяется государственными образовательными стандартами высшего профессионального образования, которые предусматривают выпуск конкурентноспособных на рынке труда специалистов. Последние должны формировать у себя прочные теоретические и практические навыки самообразования, что возможно в современных условиях только при наличии должного уровня информационной культуры.

Изучение учебной дисциплины «Методы вычислений» предусмотрено учебным планом подготовки специалистов специальности 6-05-0533-11 Прикладная информатика (профилизация программное обеспечение компьютерных систем). Изучение курса базируется на дисциплинах: «Алгебра и теория чисел», «Математический анализ», «Дифференциальные уравнения», «Технологии программирования». Данное практическое пособие призвано облегчить изучение необходимого тематического теоретического материала, а также поспособствовать приобретению практических навыков в области вычислительной математики.

В практическом пособии рассматриваются вопросы, связанные с элементами теории и практики интерполирования функций, равномерного и среднеквадратичного приближения функций. Приводятся примеры программной реализации задач, поясняющие теоретический материал.

Практическое пособие может быть использовано преподавателями при проведении практических занятий и студентами при их самостоятельной работе над учебной дисциплиной.

ТЕМА 1. ПРИБЛИЖЕНИЕ ФУНКЦИЙ

1.1 Интерполирование функций

Пусть на отрезке $[a, b]$ задана некоторая функция $f(x)$ так, что в точках $x_0, x_1, \dots, x_n$ известны её значения $y_0, y_1, \dots, y_n$. Задача интерполирования заключается в построении функции $\varphi(x)$ из некоторого класса $\Phi(x)$ такой, что выполняется условие

$$\varphi(x_i) = y_i, \quad i = \overline{1, n}. \quad (1)$$

Пусть $\Phi(x)$ класс алгебраических многочленов степени n . Тогда справедлива следующая теорема.

Теорема 1. Существует и при том единственный многочлен, степени не выше n , для которого выполняется условие (1) на любой системе попарно различных точек $x_0, x_1, \dots, x_n$.

Пусть $\varphi(x) = P_n(x) = a_0 x^n + a_1 x^{n-1} + \dots + a_n$. Тогда из условия (1) получим систему из $n+1$ линейных алгебраических уравнений

$$a_0 x_i^n + a_1 x_i^{n-1} + \dots + a_n = y_i, \quad i = \overline{0, n}. \quad (2)$$

Определяя коэффициенты $\{a_i\}$, получим интерполяционный многочлен $P_n(x)$. Заменяя функцию интерполяционным многочленом, можно определить значения $\bar{y}$ при не табличном значении аргумента $\bar{x}$. Если $\bar{x} \in [a, b]$, то говорят, что решается задача интерполирования; если $\bar{x} \notin [a, b]$, то речь идет об экстраполировании функции $f(x)$.

Будем искать интерполяционный многочлен в виде

$$L_n(x) = \sum_{i=0}^n L_{n,i}(x) \cdot y_i, \quad \text{где } L_{n,i}(x_j) = \begin{cases} 1, & \text{если } i = j \\ 0, & \text{если } i \neq j \end{cases}.$$

Очевидно, что это условие будет выполняться, если

$$L_{n,i}(x) = \frac{(x - x_0) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n)}{(x_i - x_0) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)}.$$

Тогда

$$L_n(x) = \sum_{i=0}^n \frac{(x-x_0)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_n)}{(x_i-x_0)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_n)} y_i. \quad (3)$$

Интерполяционный многочлен, записанный в форме (3), называется *многочленом Лагранжа*. Обозначая $\omega(x) = \prod_{i=0}^n (x-x_i)$, многочлен $L_n(x)$

можно записать в виде $L_n(x) = \omega(x) \sum_{i=0}^n \frac{y_i}{(x-x_i)\omega'(x_i)}$.

Если $f(x)$ $n+1$ раз непрерывно дифференцируемая функция, то для погрешности интерполирования справедлива формула

$$R_n(f) = L_n(x) - f(x) = \frac{\omega(x)}{(n+1)!} f^{(n+1)}(\xi), \quad (4)$$

где ξ некоторая точка из отрезка $[a, b]$.

На практике пользуются оценкой

$$|R_n(f)| \leq \frac{|\omega(x)|}{(n+1)!} \max_{x \in [a,b]} |f^{(n+1)}(x)|. \quad (5)$$

1.1.1 Пример программной реализации многочлена Лагранжа

Для таблицы значений с неравностоящими узлами построить интерполяционный многочлен Лагранжа третьей степени.

x	1,0	1,3	1,8	2,2
y	1,683	0,500	6,403	29,113

Ниже приводится возможная программа с фиксированным вводом на языке паскаль ABC, реализующая поставленную задачу.

```

Program Lagranj;
type mas1=array[1..10] of real;
      mas2=array[1..10,1..10] of real;
var X,A:mas2;
      n,i,j,k,p,q,l,m: integer;
      x0,y0,b,c: mas1 ;
      x0x,s,s1,s2 : real;
function Lag(t:real):real;
      var i,j: integer;

```

```

        s,p : real;
begin  s:=0;
    for i:=1 to n do
        begin
            p:=1;
            for j:=1 to n do
                if j<>i then p:=p*(t-x0[j])/(x0[i]-x0[j]);
            s:=s+p*y0[i];
            end;
        Result:=s;
    end;
begin
    n:=4;
    x0[1]:=1; x0[2]:=1.3; x0[3]:=1.8; x0[4]:=2.2;
    y0[1]:=1.683;y0[2]:=0.50;y0[3]:=6.403;y0[4]:=29.113;
    writeln('Исходные данные 1<n<5.    n=',n);
    write('x0[i]= ');
    for i:=1 to n do
        write(x0[i]:8:3); writeln;
        write('y0[i]= ');
        for i:=1 to n do
            write(y0[i]:8:3);
            writeln;
            x0x:=1.8;
            write('Значение      в      данной      точке:      Lag(' ,x0x,')=
',Lag(x0x):8:4);
//  writeln('Формирование массива X с исходными данными');
    for i:=1 to n do
        begin
            p:=1;
            for j:=1 to n do
                if j<>i then begin X[i,p]:=x0[j];p:=p+1; end;
            end;
            writeln;
            writeln('Сформированный массив X с исходными данными');
        for i:=1 to n do
            begin
                for j:=1 to n-1 do
                    write(' ',X[i,j]:6:1);
                    writeln;
                end;
                writeln('Массив b знаменателей');
            for i:=1 to n do
                begin
                    b[i]:=1;
                    for j:=1 to n-1 do
                        b[i]:=b[i]*(x0[i]-X[i,j]);
                    end;
                for i:=1 to n do
                    write(' ',b[i]:8:3);
                    writeln;
//  writeln('Коэффициенты A[i,j] при степенях x в числителе');
                for q:=1 to n do
                    begin
                        if q=1 then
                            for p:=1 to n do A[p,q]:=1;

```

```

    if q=2 then
        for p:=1 to n do
            begin A[p,q]:=0;
                for i:=1 to n-1 do A[p,q]:=A[p,q]-X[p,i];
            end;
        if q=3 then
            for p:=1 to n do
                begin A[p,q]:=0;
                    for i:=1 to n-1 do
                        begin s:=0;
                            for j:=i+1 to n-1 do s:=s+X[p,j];
                                A[p,q]:=A[p,q]+X[p,i]*s;
                            end;
                        end;
                    end;
                if q=n then
                    for p:=1 to n do
                        begin A[p,q]:=1;
                            for i:=1 to n-1 do A[p,q]:=-A[p,q]*X[p,i];
                        end;
                    end;
                writeln('Матрица A коэффициентов при степенях x');
            for i:=1 to n do
                begin
                    for j:=1 to n do
                        write(' ',A[i,j]:8:3);
                    writeln;
                end;
                writeln('Коэффициенты полинома Лагранжа при степенях x');
            for i:=1 to n do
                begin
                    c[i]:=0;
                    for j:=1 to n do
                        c[i]:=c[i]+y0[j]*A[j,i]/b[j];
                    end;
                    for i:=1 to n do
                        write(' ',c[i]:6:3);
                    writeln;
                end;
                writeln('Полином Лагранжа');
                write('Lagranj(x)= ');
                for i:=1 to n-1 do
                    if i=1 then write(' ',c[i]:3:3,'*x^',n-i)
                    else if c[i]>=0.0 then write(' + ',c[i]:3:3,'*x^',n-i)
                        else write(' ',c[i]:3:3,'*x^',n-i);
                    if c[n]>=0.0 then write(' + ',c[n]:3:3) else write('
',c[n]:3:3);
                    writeln;
                end;
                writeln('Проверка'); x0x:=1.8;
                s:=c[1]*x0x;
                for i:=2 to n-1 do
                    s:=(s+c[i])*x0x;
                    s:=s+c[n];
                end;
                write('Значение полинома Лагранжа в точке x= ',x0x,' равно
',s:8:4);
            end.

```

Результат выполнения программы

```

Окно вывода
Исходные данные 1<n<5.    n=4
x0[i]=    1.000    1.300    1.800    2.200
y0[i]=    1.683    0.500    6.403    29.113
Значение в данной точке: Lag(1.8)=    6.4030
Сформированный массив X с исходными данными
    1.3    1.8    2.2
    1.0    1.8    2.2
    1.0    1.3    2.2
    1.0    1.3    1.8
Массив b знаменателей
    -0.288    0.135    -0.160    0.432
Матрица A коэффициентов при степенях x
    1.000    -5.300    9.160    -5.148
    1.000    -5.000    7.960    -3.960
    1.000    -4.500    6.360    -2.860
    1.000    -4.100    5.440    -2.340
Коэффициенты полинома Лагранжа при степенях x
    25.232 -83.766 88.042 -27.825
Полином Лагранжа
Lagranj(x)= 25.232*x^3 -83.766*x^2 + 88.042*x^1 -27.825
Проверка
Значение полинома Лагранжа в точке x= 1.8 равно    6.4030|

```

1.2 Интерполирование для равноотстоящих узлов

Пусть все узлы интерполяции равностоящие, так что $x_{i+1} = x_i + h$, $h = \frac{b-a}{n}$. Величина $\Delta y(x) = y(x+h) - y(x)$ называется *первой конечной разностью* или *разностью первого порядка функции $y(x)$ в точке x* . Конечной разностью порядка k называется величина $\Delta^k y(x) = \Delta(\Delta^{k-1} y(x))$. Из определения конечных разностей вытекают следующие свойства:

- 1) $\Delta(u + v) = \Delta u + \Delta v$;
- 2) $\Delta(cu) = c\Delta u$, где c – некоторая постоянная;
- 3) $\Delta^m(\Delta^n u) = \Delta^{m+n} u$.

Так как конечные разности определяются значениями в точках $x_0, x_1, \dots, x_n$ функции $y(x)$, то из условия интерполирования следует, что

$$\Delta^m P_n(x_0) = \Delta^m y_0, \quad m = \overline{1, n}. \quad (6)$$

Используя равенство (6), интерполяционный многочлен можно записать в виде

$$P_n(x) = y_0 + \frac{\Delta y_0}{1!h}(x-x_0) + \frac{\Delta^2 y_0}{2!h^2}(x-x_0)(x-x_1) + \dots + \frac{\Delta^n y_0}{n!h^n} \times \\ \times (x-x_0)(x-x_1)\dots(x-x_{n-1}). \quad (7)$$

Если сделать замену $q = \frac{x-x_0}{h}$, то формулу (7) можно записать в более компактной форме

$$P_n(q(x)) = y_0 + q\Delta y_0 + \frac{q(q-1)}{2!}\Delta^2 y_0 + \dots + \frac{q(q-1)\dots(q-n+1)}{n!}\Delta^n y_0. \quad (8)$$

Формула (8) называется *первой интерполяционной формулой Ньютона для равноотстоящих узлов*. Аналогичным образом можно построить *вторую интерполяционную формулу Ньютона для равноотстоящих узлов*

$$P_n(q(x)) = y_n + q\Delta y_{n-1} + \frac{q(q+1)}{2!}\Delta^2 y_{n-2} + \dots + \frac{q(q+1)\dots(q+n-1)}{n!}\Delta^n y_0, \quad (9)$$

$$\text{где } q = \frac{x-x_n}{h}.$$

Формулу (7) следует использовать для интерполирования или экстраполирования вблизи начала таблицы значений, а формулу (9) вблизи конца таблицы значений.

Остаточные члены в формулах (8) и (9) соответственно имеют вид

$$R_n(x) = h^{n+1} \frac{q(q-1)\dots(q-n)}{(n+1)!} f^{(n+1)}(\xi),$$

$$R_n(x) = h^{n+1} \frac{q(q+1)\dots(q+n)}{(n+1)!} f^{(n+1)}(\xi).$$

В том случае, когда точка $\bar{x}$ лежит в середине таблицы, используют формулы с центральными разностями, построенные для системы узлов $x_{-n}, x_{-(n-1)}, \dots, x_{-1}, x_0, x_1, \dots, x_{2n}$, например, *первую формулу Гаусса*

$$P_{2n}(x) = y_0 + q\Delta y_0 + \frac{q(q-1)}{2!}\Delta^2 y_{-1} + \frac{(q+1)q(q-1)}{3!}\Delta^3 y_{-1} + \dots + \frac{(q+n-1)\dots(q+1)q(q-1)\dots(q-n)}{(2n)!}\Delta^{2n} y_{-n} \quad (10)$$

или вторую формулу Гаусса

$$P_{2n}(x) = y_0 + q\Delta y_{-1} + \frac{(q+1)q}{2!}\Delta^2 y_{-1} + \frac{(q+1)q(q-1)}{3!}\Delta^3 y_{-2} + \dots + \frac{(q+n)\dots((q+1)q(q-1)\dots(q-n+1))}{(2n)!}\Delta^{2n} y_{-n}. \quad (11)$$

При этом первая формула Гаусса (10) применяется при $x > x_0$, а вторая формула (11) – при $x < x_0$.

1.2.1 Пример программной реализации многочлена Ньютона

Для таблицы значений с равностоящими узлами построить интерполяционный многочлен Ньютона третьей степени и найти его значение в заданной точке $\bar{x} = 2,2$.

x	1,0	2,0	3,0	4,0
y	1,0	5,0	14,0	25,0

Ниже приводится возможная программа на языке Паскаль ABC, реализующая поставленную задачу с фиксированным вводом.

```

Program Newton1;
  var i,j,n: integer;
  a,x,y,dy,p: array [1..4] of real ;
  k,x0,h : real;
  function PN(x:real):real;
    begin Result:= ((p[4]*x+p[3])*x+p[2])*x+p[1];    end;
begin
  k:=1; n:=4; h:=1;
  x[1]:=1.0; x[2]:=2.0; x[3]:=3.0; x[4]:=4.0;
  y[1]:=1.0; y[2]:=5.0; y[3]:=14.0; y[4]:=25.0;
  writeln('Исходные данные');
  write('x[i]= ');
  for i:=1 to n do

```

```

    write(x[i]:6:1); writeln;
    write('y[i]= ');
    for i:=1 to n do
        write(y[i]:6:1); writeln;
writeln(' Коэффициенты для полинома Ньютона');
for i:=1 to n do
    dy[i]:=y[i];
    for i:=1 to n-1 do
        begin
            for j:=1 to n-i do
                dy[j]:=dy[j+1]-dy[j];
            k:=k*i*h;
            a[i]:=dy[1]/k;
            write(' a[' ,i, ']= ', a[i]);
        end;
    writeln;
{ Коэффициенты для формулы Ньютона по степеням x}
    p[4]:=a[3];
    p[3]:=a[2]-a[3]*(x[1]+x[2]+x[3]);
    p[2]:=a[1]-
a[2]*(x[1]+x[2])+a[3]*(x[1]*x[2]+x[1]*x[3]+x[2]*x[3]);
    p[1]:=y[1]-a[1]*x[1]+a[2]*x[1]*x[2]-a[3]*x[1]*x[2]*x[3];
    writeln;
    writeln('1 формула Ньютона для равноотстоящих узлов');
    writeln('PN=',p[4],'x^3 ',p[3],'x^2 ',p[2],'x ',p[1]);
    writeln;
    x0:=4.0;
    write('Проверка: PN(',x0,')= ',PN(x0),' y[2]= ',y[2]);
    writeln;
    x0:=2.2;
    write('Значение в данной точке: PN(',x0,')= ',PN(x0));
end.

```

Результат выполнения программы

Окно вывода

```

Исходные данные
x[i]=    1.0    2.0    3.0    4.0
y[i]=    1.0    5.0   14.0   25.0
Коэффициенты для полинома Ньютона
a[1]=4   a[2]=2.5   a[3]=-0.5|

1 формула Ньютона для равноотстоящих узлов
PN=-0.5x^3 5.5x^2 -9x 5

Проверка: PN(4)= 25 y[4]= 25
Значение в данной точке: PN(2.2)= 6.496

```

Программная реализация является читаемой и содержит всю информацию необходимую для понимания решения задачи.

1.3 Интерполирование для неравноотстоящих узлов

Для неравноотстоящих узлов интерполяции понятие конечной разности обобщается в понятие разделенной разности. Величина

$f(x_0, x_1) = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$ называется *разделенной разностью* первого

порядка $f(x)$ в точке x_0 . Разделенной разностью порядка n в точке x_i называется величина

$$f(x_i, x_{i+1}, \dots, x_{i+n}) = \frac{f(x_{i+1}, \dots, x_{i+n}) - f(x_i, \dots, x_{i+n-1})}{x_{i+n} - x_i}.$$

Из определения следует, что разделенная разность является симметрической функцией своих аргументов, т. е.

$$f(x_0, x_1, \dots, x_i, x_{i+1}, \dots, x_n) = f(x_n, x_{n-1}, \dots, x_{i+1}, x_i, \dots, x_0).$$

Используя понятие разделенной разности, формулу Лагранжа можно записать в виде *первой интерполяционной формулы Ньютона для неравноотстоящих узлов*

$$\begin{aligned} P_n(x) = & f(x_0) + f(x_0, x_1)(x - x_0) + f(x_0, x_1, x_2)(x - x_0)(x - x_1) + \\ & + \dots + f(x_0, x_1, \dots, x_n)(x - x_0)(x - x_1) \dots (x - x_{n-1}) \end{aligned} \quad (12)$$

или в виде *второй интерполяционной формулы Ньютона для неравноотстоящих узлов*

$$\begin{aligned} P_n(x) = & f(x_n) + f(x_n, x_{n-1})(x - x_n) + \\ & + f(x_n, x_{n-1}, x_{n-2})(x - x_n)(x - x_{n-1}) + \dots + f(x_n, \dots, x_0)(x - x_n) \dots (x - x_1). \end{aligned} \quad (13)$$

Формулы (12) и (13) следует использовать для интерполирования или экстраполирования в начале и конце таблицы значений соответственно.

1.4 Многочлены Чебышева

Из выражения для погрешности следует, что ошибка интерполирования зависит от $|\omega(x)|$, т. е. от распределения узлов интерполирования $x_0, x_1, \dots, x_n$ и от величины $f^{(n+1)}(x)$ на отрезке $[a, b]$. Допустим, что $f^{(n+1)}(x)$ мало меняется на отрезке $[a, b]$. В этом случае величина погрешности зависит от величины $|\omega(x)|$ – модуля отклонения $\omega(x)$ от нуля на отрезке $[a, b]$. Эта задача вплотную примыкает к задаче, решенной русским математиком П. Л. Чебышевым: среди всех многочленов степени n с коэффициентом равным единице при старшей степени найти многочлен наименее уклоняющийся от нуля на отрезке $[-1, +1]$.

Теорема 2. Среди всех многочленов степени n со старшим коэффициентом равным единице наименьшее значение максимума модуля на отрезке $[-1, +1]$ имеет многочлен

$$\bar{T}_n(x) = 2^{1-n} T_n(x), \quad n \geq 1,$$

где $T_n(x) = \cos(n \arccos x)$ – многочлен Чебышева.

Отрезок $[a, b]$ заменой переменных $t = \frac{2x}{b-a} - \frac{b+a}{b-a}$ переводится в отрезок $[-1, +1]$. Поэтому для минимизации $|\omega(x)|$ в качестве узлов интерполирования следует взять точки $x_k = \frac{1}{2}(t_k(b-a) + b+a)$, где

$t_k = \cos \frac{(2k+1)}{2(n+1)}\pi$ – корни многочлена Чебышева степени $n+1$. Тогда

$x_k - x = \frac{(b-a)(t_k - t)}{2}$ и для величины отклонения $\omega(x)$ получим

$$\omega(x) = \frac{(b-a)^{n+1}}{2^{n+1}} \omega(t) = \frac{(b-a)^{n+1}}{2^{n+1}} \bar{T}_{n+1}(t) \leq \frac{(b-a)^{n+1}}{2^{2n+1}}.$$

В этом случае для погрешности справедлива оценка

$$|R_n(x)| \leq \frac{|f^{(n+1)}(\xi)|}{(n+1)!} \frac{(b-a)^{n+1}}{2^{2n+1}}.$$

1.5 Обратное интерполирование

Задача обратного интерполирования заключается в том, чтобы по заданному значению $\bar{y}$ определить аргумент $\bar{x}$, соответствующий этому значению. Решение задачи будет единственным, если $f(x)$ монотонна на отрезке, содержащем узлы интерполирования. Если узлы равноотстоящие, то для решения этой задачи можно использовать метод последовательных приближений. Пусть $\bar{y}$ содержится между y_0 и y_1 . Запишем первую интерполяционную формулу Ньютона в виде

$$q = \varphi(q) = \frac{y - y_0}{\Delta y_0} - \frac{\Delta^2 y_0}{2! \Delta y_0} q(q-1) - \dots - \frac{\Delta^n y_0}{n! \Delta y_0} q(q-1) \dots (q-n+1). \quad (14)$$

Для определения q можно использовать итерационный процесс

$$q_m = \varphi(q_{m-1}), \quad m = 1, 2, \dots \quad (15)$$

В качестве нулевого приближения можно положить $q_0 = \frac{y - y_0}{\Delta y_0}$. Если итерационный процесс сходится, то значение искомого аргумента определяется по формуле $\bar{x} = x_0 + qh$.

В случае неравноотстоящих узлов значение аргумента можно определить по формуле Лагранжа для обратной функции

$$\bar{x} = \sum_{i=0}^n \frac{(\bar{y} - y_0) \dots (\bar{y} - y_{i-1})(\bar{y} - y_{i+1}) \dots (\bar{y} - y_n)}{(y_i - y_0) \dots (y_i - y_{i-1})(y_i - y_{i+1}) \dots (y_i - y_n)} x_i. \quad (16)$$

1.6 Интерполирование функций многих переменных

1.6.1 О проблеме интерполирования функций многих переменных

Интерполирование функций многих переменных значительно сложнее, чем для одной переменной. Ограничимся случаем двух переменных. Пусть на плоскости (x, y) даны $n+1$ точка $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$.

Будем искать многочлен $P(x, y)$ относительно (x, y) возможно низшей степени, который в этих точках принимал соответственно значения $z_0, z_1, \dots, z_n$. Если его записать в виде

$$P(x, y) = a_{00} + a_{10}x + a_{01}y + a_{20}x^2 + a_{11}xy + a_{02}y^2,$$

то, подставляя данные координаты точек, и приравнявая левую часть соответствующему значению z_i , получим систему $n+1$ линейных независимых алгебраических уравнений относительно

$1 + 2 + 3 + \dots + m + 1 = \frac{(m+1)(m+2)}{2}$ неизвестных a_{ij} . Если не накладывать

на $P(x, y)$ никаких дополнительных условий, то $n+1 = \frac{(m+1)(m+2)}{2}$.

То есть, мы не можем решать задачу при любом количестве узлов интерполирования – это *во-первых*.

Рассмотрим определитель получившейся системы. Например, при $n = 2$ и $n = 5$

$$\begin{vmatrix} 1 & x_0 & y_0 \\ 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \end{vmatrix} (n=2), \quad \begin{vmatrix} 1 & x_0 & y_0 & x_0^2 & x_0y_0 & y_0^2 \\ 1 & x_1 & y_1 & x_1^2 & x_1y_1 & y_1^2 \\ 1 & x_2 & y_2 & x_2^2 & x_2y_2 & y_2^2 \\ 1 & x_3 & y_3 & x_3^2 & x_3y_3 & y_3^2 \\ 1 & x_4 & y_4 & x_4^2 & x_4y_4 & y_4^2 \\ 1 & x_5 & y_5 & x_5^2 & x_5y_5 & y_5^2 \end{vmatrix} (n=5).$$

Первый определитель обращается в нуль, если все три точки $(x_0, y_0), (x_1, y_1), (x_2, y_2)$ лежат на одной прямой; второй равен нулю, если все шесть точек лежат на одной кривой второго порядка. То есть, узлы интерполирования не могут быть расположены произвольно – это *во-вторых*, а проверка того, что определители не равны нулю довольно затруднительна. *Третье* принципиальное затруднение возникает при оценке остаточных членов, так как теорема Ролля в этом случае неприменима. Отметим так же и то, что более громоздкими становятся рассуждения, связанные с большим числом переменных.

1.6.2 Интерполяционная формула Ньютона

Приведем интерполяционную формулу Ньютона для случая двух переменных. Возьмем $\frac{(n+1)(n+2)}{2}$ узлов, расположенных следующим образом

$$\begin{array}{ccccccc} (x_0, y_0) & (x_1, y_0) & \dots & (x_{n-1}, y_0) & (x_n, y_0) & & \\ (x_0, y_1) & (x_1, y_1) & \dots & (x_{n-1}, y_1) & & & \\ \dots & \dots & & & & & \\ (x_0, y_{n-1}) & (x_1, y_{n-1}) & & & & & \\ (x_0, y_n) & & & & & & \end{array}$$

Причем, $x_i \neq x_j$, $y_i \neq y_j$ при $i \neq j$. Значения x_i и y_i могут быть произвольными, так, что взаимное расположение узлов может быть довольно общим. Обобщение интерполяционной формулы Ньютона (12) для неравных промежутков на случай интерполирования функций двух переменных имеет вид

$$P_n(x, y) = \sum_{k=0}^n \sum_{i+j=k}^n (x-x_0) \dots (x-x_{i-1})(y-y_0) \dots \\ \dots (y-y_{j-1}) f(x_0, x_1, \dots, x_i, y_0, y_1, \dots, y_j).$$

1.7 Численное дифференцирование

Заменяя $f(x)$ интерполяционным приближением $\varphi(x)$, получим $f(x) = \varphi(x) + R(f)$ и $f^{(k)}(x) = \varphi^{(k)}(x) + R^{(k)}(f(x))$. Если погрешность $R^{(k)}(f(x))$ достаточно малая, то можно положить $f^{(k)}(x) \approx \varphi^{(k)}(x)$.

Пусть узлы $x_0, x_1, \dots, x_n$ – равноотстоящие. Тогда для вычисления производной в точке x вблизи x_0 удобно использовать первую интерполяционную формулу Ньютона

$$P_n(q) = y_0 + q\Delta y_0 + \frac{\Delta^2 y_0}{2!} q(q-1) + \dots + \frac{\Delta^n y_0}{n!} q(q-1) \dots (q-n+1).$$

Учитывая, что $\frac{dP}{dx} = \frac{dP_n}{dq} \frac{dq}{dx} = \frac{1}{h} \frac{dP_n}{dq}$, получим

$$P_n'(x) = \frac{1}{h} \left(\Delta y_0 + \frac{2q-1}{2} \Delta^2 y_0 + \frac{3q^2-6q+2}{6} \Delta^3 y_0 + \frac{2q^3-9q^2+11q-3}{12} \Delta^4 y_0 + \dots \right),$$

$$P_n''(x) = \frac{1}{h^2} \left(\Delta^2 y_0 + \Delta^3 y_0 (q-1) + \frac{6q^2-18q+11}{12} \Delta^4 y_0 + \dots \right).$$

Тогда

$$R_n'(x) = \frac{h^n}{(n+1)!} \left\{ f^{(n+1)}(\xi) \frac{d}{dq} [q(q-1)\dots(q-n)] + q(q-1)\dots(q-n) \frac{d}{dq} [f^{(n+1)}(\xi)] \right\}.$$

В узлах интерполирования формула последняя примет вид

$$R_n'(x) = (-1)^{n-i} \frac{h^n i! (n-i)!}{(n+1)!} f^{(n+1)}(\xi).$$

При достаточно малом шаге $h \ll 1$, можно считать, что $f^{(n+1)}(\xi) \approx \frac{\Delta^{n+1} f_0}{h^{n+1}}$.

Если узлы неравноотстоящие, то для построения функции численного дифференцирования можно использовать формулы Ньютона для неравноотстоящих узлов

$$P_n'(x) = f(x_0, x_1) + f(x_0, x_1, x_2)(\alpha_0 + \alpha_1) + \dots + f(x_0, \dots, x_i) \times$$

$$\times \sum_{k=0}^{i-1} \alpha_0 \alpha_1 \dots \alpha_{k-1} \alpha_{k+1} \dots \alpha_{i-1} + \dots + f(x_0, x_1, \dots, x_n) \sum_{k=0}^{n-1} \alpha_0 \alpha_1 \dots \alpha_{k-1} \alpha_{k+1} \dots \alpha_{n-1},$$

где $\alpha_i = x - x_i$.

1.7.1 Погрешность формул численного дифференцирования

Для погрешности численного дифференцирования справедливой формулы $R_n^{(k)}(f(x)) = \frac{\omega^{(k)}(x)}{(n+1)!} f^{(n+1)}(\xi)$, если $x \notin [a, b]$ и

$$R_n^{(k)}(f(x)) = \frac{\bar{\omega}(x)}{(n-k+1)!} f^{(n+1)}(\xi), \quad \text{если} \quad x \in [a, b], \quad \text{где}$$

$\bar{\omega}(x) = (x - \varepsilon_0) \dots (x - \varepsilon_{n-k})$, а $\varepsilon_0, \dots, \varepsilon_{n-k}$ — нули $f^{(k)}(x) - L_n^{(k)}(x)$ на отрезке $[a, b]$. При численном дифференцировании может произойти существенная потеря точности. Так при использовании формулы $f'(x_0) \approx \frac{f(x_1) - f(x_0)}{h}$ значение производной вычисляется в лучшем

случае с половиной верных знаков. Поэтому на практике редко применяют формулы численного дифференцирования для производных выше второго порядка.

1.8 Приближение функций сплайнами

Сплайном называется функция, которая вместе с несколькими производными непрерывна на всем заданном отрезке $[a, b]$, а на каждом частичном отрезке $[x_i, x_{i+1}]$ является некоторым алгебраическим многочленом. Максимальная по всем отрезкам степень многочленов называется *степенью сплайна*, а разность между степенью сплайна и порядком наивысшей непрерывной производной на $[a, b]$ — *дефектом*. На практике широко используются сплайны третьей степени $S_3(x)$. Такие сплайны называются *кубическими*. Сплайн, который принимает в узлах те же значения, что и функция $f(x)$ называется *интерполяционным*.

Пусть $k_i = S_3^{(1)}(x_i)$ — наклоны сплайна в точке x_i . Если заданы значения f_i, f_{i+1} и k_i, k_{i+1} на $[x_i, x_{i+1}]$ и $h_i = x_{i+1} - x_i$, то интерполяционный сплайн имеет вид

$$\begin{aligned} S_3(x) = & \frac{(x_{i+1} - x)^2 (2(x - x_i) + h_{i+1})}{h_{i+1}^3} f_i + \\ & + \frac{(x - x_i)^2 (2(x_{i+1} - x) + h_{i+1})}{h_{i+1}^3} f_{i+1} + \\ & + \frac{(x_{i+1} - x)^2 (x - x_i)}{h_{i+1}^2} k_i + \frac{(x - x_i)^2 (x - x_{i+1})}{h_{i+1}^2} k_{i+1}. \end{aligned} \quad (17)$$

О способах задания сплайнов. Существует несколько способов задания интерполяционных сплайнов. Пусть узлы интерполирования – равноотстоящие.

Способ 1. Положим, что $k_i = S_3'(x_i)$. Тогда определяя наклоны k_i по формуле $k_i = \frac{f_{i+1} - f_{i-1}}{2h}$, $i = 1, \dots, N-1$, $k_0 = \frac{4f_1 - f_2 - 3f_0}{2h}$, $k_N = \frac{3f_N + f_{N-2} - 4f_{N-1}}{2h}$, где $h = \frac{b-a}{N}$, кубический сплайн можно записать в виде (17), причем $h_{i+1} = h$.

Способ 2. Если известны значения f_i' , то полагаем $k_i = f_i'$. Затем сплайн записывается в форме (17).

Способы 1 и 2 называются *локальными*. При их использовании гарантируется непрерывность $S_3'(x)$ в узлах x_i .

Способ 3. Будем строить сплайн $S_3(x)$ так, чтобы выполнялись условия:

- 1) $S_3(x) \in C^2(a, b)$;

- 2) на каждом из отрезков сплайн является многочленом третьей степени;

- 3) в узлах $S_3(x_k) = f(x_k)$, $k = \overline{0, n}$.

Будем искать сплайн на отрезке $[x_{i-1}, x_i]$ в виде

$$S_i(x) = a_i + b_i(x - x_i) + \frac{c_i}{2}(x - x_i)^2 + \frac{d_i}{6}(x - x_i)^3. \quad (18)$$

Из условия 3) следует, что $a_i = S_i(x_i)$; $b_i = S_i'(x_i)$, $c_i = S_i''(x_i)$, $d_i = S_i^{(3)}(x_i)$. Условие 3) обеспечивают равенства $a_0 = f_0$, $a_i = f_i$. Из условия непрерывности 1) функции получим $S_i(x_i) = S_{i+1}(x_i)$, что приводит к уравнению

$$f_i - f_{i-1} = h_i b_i - \frac{h_i^2}{2} c_i + \frac{h_i^3}{6} d_i, \quad i = \overline{1, N}. \quad (19)$$

Из условия непрерывности первой производной $S_i'(x_i) = S_{i+1}'(x_i)$ имеем

$$c_i h_i - \frac{d_i}{2} h_i^2 = b_i - b_{i-1}, \quad i = \overline{2, N}. \quad (20)$$

Из условия непрерывности второй производной $S_i''(x_i) = S_{i+1}''(x_i)$ вытекает

$$d_i h_i = c_i - c_{i-1}, \quad i = \overline{2, N}. \quad (21)$$

Следовательно, мы имеем систему (19)–(21) из $3N - 2$ уравнений относительно $b_1, b_2, \dots, b_N, c_1, c_2, \dots, c_N, d_1, d_2, \dots, d_N - 3N$ неизвестных. Дополняя эту систему условиями на границе для $S_3(x)$, получим систему из $3N$ уравнений.

Рассмотрим различные типы краевых условий.

1 Пусть заданы значения $f_0'' = c_0$ и $f_N'' = c_N$. Из уравнений (19)–(21) можно получить систему

$$\begin{cases} c_0 = f_0^{(2)} \\ h_i c_{i-1} + 2(h_i + h_{i+1})c_i + h_{i+1}c_{i+1} = 6 \left(\frac{f_{i+1} - f_i}{h_{i+1}} - \frac{f_i - f_{i-1}}{h_i} \right), \quad i = 1, 2, \dots, n-1 \\ c_n = f_n^{(2)} \end{cases} \quad (22)$$

Система (22) имеет диагональное преобладание и может быть эффективно решена методом трехточечной прогонки. Затем можно определить остальные неизвестные

$$d_i = \frac{c_i - c_{i-1}}{h_i}; \quad b_i = \frac{h_i}{2}c_i - \frac{h_i^2}{6}d_i + \frac{f_i - f_{i-1}}{h_i}.$$

2 Если на границе заданы значения первых производных b_0 и b_N , $f_a^{(1)} = f_0^{(1)} = b_0$ и $f_b^{(1)} = f_n^{(1)} = b_n$, то систему (19)–(21) можно привести к виду

$$\begin{cases} b_0 = f_0^{(1)} \\ 2h_{i+1}b_{i-1} + 4(h_i + h_{i+1})b_i + 2h_i b_{i+1} = 6 \left(\frac{h_i}{h_{i+1}}(f_{i+1} - f_i) + \frac{h_{i+1}}{h_i}(f_i - f_{i-1}) \right), \quad i = 1, 2, \dots, n-1 \\ b_n = f_n^{(1)} \end{cases} \quad (23)$$

Очевидно, что система (23) также может быть эффективно решена методом прогонки. Если эти граничные условия использовать при решении системы (22), то ее нужно дополнить уравнениями

$$\frac{2}{3}c_0 + \frac{1}{3}c_1 = \frac{2}{h_1} \left(\frac{f_1 - f_0}{h_1} - f_0^{(1)} \right),$$

$$\frac{1}{3}c_{n-1} + \frac{2}{3}c_n = \frac{2}{h_n} \left(f_n^{(1)} - \frac{f_n - f_{n-1}}{h_n} \right).$$

3 Если на границе заданы значения только функции f_0 и f_N , то можно решать систему (23) с дополнительными условиями

$$b_0 = \frac{1}{6h}(-11f_0 + 18f_1 - 9f_2 + 2f_3),$$

$$b_N = \frac{1}{6h}(11f_N - 18f_{N-1} + 9f_{N-2} - 2f_{N-3}).$$
(24)

4 Пусть $f(x)$ периодическая функция с периодом $b-a$, тогда следует считать, что $f_0 = f_N$, $S_3'(a) = S_3'(b)$, $S_3''(a) = S_3''(b)$. Если использовать систему (23), то ее нужно дополнить условиями

$$b_0 = b_N,$$
(25)

$$\frac{f_1 - f_0}{h_1^2} - \frac{f_{N-1} - f_0}{h_N^2} = \frac{1}{3h_1}(2b_0 + b_1) + \frac{1}{3h_N}(2b_0 + b_{N-1}).$$

Для равномерной сетки из (25) получим уравнение

$$\frac{3(f_1 - f_{N-1})}{h} = b_1 + 4b_0 + b_{N-1}.$$

Если использовать систему (22), то ее можно дополнить краевыми условиями

$$c_0 = c_N,$$
(26)

$$c_1 h_1 + 2(h_N + h_1)c_N + c_{N-1} h_N = 6 \left[\frac{f_1 - f_N}{h_1} - \frac{f_N - f_{N-1}}{h_N} \right].$$

Для равномерной сетки из (26) получим уравнение

$$c_1 + 4c_N + c_{N-1} = \frac{6}{h^2} (f_1 - 2f_N + f_{N-1}).$$

После решения системы (22) или (23) кубический сплайн можно сразу записать в явном виде (17) или в виде на отрезке $[x_{i-1}, x_i]$

$$\begin{aligned} S_3(x) = & c_{i-1} \frac{(x_i - x)^3}{6h_i} + c_i \frac{(x - x_{i-1})^3}{6h_i} + \\ & + \left(f_{i-1} - \frac{c_{i-1}h_i^2}{6} \right) \frac{x_i - x}{h_i} + \left(f_i - \frac{c_i h_i^2}{6} \right) \frac{x - x_{i-1}}{h_i}. \end{aligned} \quad (27)$$

Погрешность приближения сплайнами

Теорема 3. Если $f(x) \in C_{k+1}[a, b]$, $0 \leq k \leq 3$, то интерполяционный сплайн $S_3(x)$, построенный способами 2 или 3, удовлетворяет неравенству

$$\max_{[x_i, x_{i+1}]} |f^{(m)}(x) - S_3^{(m)}| \leq ch^{k+1-m} \max_{[a, b]} |f^{(k+1)}(x)| \quad (28)$$

где $i = \overline{0, N-1}$;

$m = \overline{0, k}$;

c – не зависящая от h, i, f постоянная.

Если $f \in C_4[a, b]$, то можно положить $c = 1$. Если $S_3(x)$ построен способом 1, то теорема справедлива при $0 \leq k \leq 2$.

1.8.1 Пример программной реализации кубического сплайна

Для таблицы значений с равностоящими узлами построить кубический сплайн, используя наклоны и найти значение функции в заданной точке $\bar{x} = 1,6$.

x	1,1	1,3	1,5	1,7	1,9
y	1,683	0,50	2,403	9,113	16,353

Ниже приводится возможная программа на языке Паскаль ABC, реализующая поставленную задачу с фиксированным вводом исходных данных.

```

Program Splain1;
var i,j,n: integer;
a,x,y,k,p: array [1..5] of real ;
x0,h,s : real;
function SpN(x:real):real;
begin Result:= ((p[4]*x+p[3])*x+p[2])*x+p[1])/h/h/h;
end;
begin
n:=5; h:=0.2;
x[1]:=1.1; x[2]:=1.3; x[3]:=1.5; x[4]:=1.7; x[5]:=1.9;
y[1]:=1.683; y[2]:=0.50; y[3]:=2.403; y[4]:=9.113; y[5]:=16.353;
k[1]:=0.5*(4*y[2]-y[3]-3*y[1])/h; k[5]:=0.5*(3*y[5]+y[3]-
4*y[4])/h;
for j:=2 to n-1 do
k[j]:=0.5*(y[j+1]-y[j-1])/h;
// коэффициенты сплайна
i:=2; //локальный сплайн, для каждого i свой
p[4]:=2*y[i]-2*y[i+1]+h*k[i]+h*k[i+1];
p[3]:=y[i+1]*(h+4*x[i]+2*x[i+1])-
h*(k[i]*x[i]+2*k[i+1]*x[i]+2*k[i]*x[i+1]+k[i+1]*x[i+1]);
p[3]:=p[3]+y[i]*(h-2*(x[i]+2*x[i+1]));
p[2]:=2*y[i]*x[i+1]*(2*x[i]+x[i+1])-
2*y[i+1]*x[i]*(h+x[i]+2*x[i+1]);
p[2]:=p[2]+h*(k[i+1]*x[i]*(x[i]+2*x[i+1])+x[i+1]*(-
2*y[i]+2*k[i]*x[i]+k[i]*x[i+1]));
p[1]:=y[i+1]*x[i]*x[i]*(h+2*x[i+1]);
p[1]:=p[1]-x[i+1]*(2*y[i]*x[i]*x[i+1]+h*(k[i+1]*x[i]*x[i]-
y[i]*x[i+1]+k[i]*x[i]*x[i+1]));
writeln('Исходные данные');
write('x[j]= ');
for j:=1 to n do
write(x[j]:8:5); writeln;
write('y[j]= ');
for j:=1 to n do
write(y[j]:8:5); writeln;
writeln('Коэффициенты кубического сплайна');
for j:=1 to n-1 do
write(p[j]:8:5); writeln;
writeln('Кубический сплайн');
write('PN3(x)= ');
for j:=1 to n-2 do
if j=1 then write(' ',p[n-j]:3:5,'*x^',n-j-1)
else if p[n-j]>=0.0 then write(' + ',p[n-j]:3:5,'*x^',n-
j-1)
else write(' ',p[n-j]:3:5,'*x^',n-j-1);
if p[1]>=0.0 then write(' + ',p[1]:3:5)

```

```

else write(' ',p[1]:3:5);
writeln;
writeln('Проверка для i= ',i); x0:=1.3;
write(' SpN(',x0,')= ',SpN(x0):8:5,' y[' ,i,']=
',y[i]:8:5);
writeln; x0:=1.6;
write('Значение в данной точке: SpN(',x0,')=
',SpN(x0):8:5);
writeln;
end.

```

Программная реализация имеет вид

```

Окно вывода
Исходные данные
x[j]= 1.10000 1.30000 1.50000 1.70000 1.90000
y[j]= 1.68300 0.50000 2.40300 9.11300 16.35300
Коэффициенты кубического сплайна
-1.67455 4.02223 -3.21945 0.86050
Кубический сплайн
PN3(x)= 0.86050*x^3 -3.21945*x^2 + 4.02223*x^1 -1.67455
Проверка для i= 2
SpN(1.3)= 0.50000 y[2]= 0.50000
Значение в данной точке: SpN(1.6)= 5.47981

```

1.9 Задания по интерполированию функций

1 Просуммировать конечные ряды.

Таблица 1

Вариант	Ряд
1	2
1	$1+3+5+\dots+(2n-1)$
2	$1^2+2^2+3^2+\dots+n^2$
3	$1^3+2^3+3^3+\dots+n^3$
4	$1^2+3^2+5^2+\dots+(2n-1)^2$
5	$1^3+3^3+5^3+\dots+(2n-1)^3$
6	$1+4+7+\dots+(3n-2)$
7	$1^2+4^2+7^2+\dots+(3n-2)^2$
8	$1^3+4^3+7^3+\dots+(3n-2)^3$
9	$1+5+9+\dots+(4n-3)$
10	$1^2+5^2+9^2+\dots+(4n-3)^2$

Окончание таблицы 1

1	2
11	$1^3 + 5^3 + 9^3 + \dots + (4n-3)^3$
12	$2^2 + 4^2 + 6^2 + \dots + (2n)^2$
13	$2^3 + 4^3 + 6^3 + \dots + (2n)^3$
14	$3^2 + 7^2 + 11^2 + \dots + (4n-1)^2$
15	$2^2 + 6^2 + 10^2 + \dots + (4n-2)^2$

2 Для функции, заданной таблично, построить интерполяционный многочлен Лагранжа и вычислить значение функции в точке $\bar{x}$.

Таблица 2

Вариант	x_i	2.0	2.3	2.5	3.0	$\bar{x}$
1	y_i	5,84	6,13	6,30	6,69	2,02
2	y_i	11,38	12,80	14,70	17,07	2,09
3	y_i	3,14	4,15	5,65	6,91	2,91
4	y_i	6,87	6,41	4,42	3,91	2,85
5	y_i	7,19	6,21	5,12	3,98	2,44
6	y_i	1,50	1,34	1,23	1,16	2,13
7	y_i	1,10	1,05	0,97	0,79	2,25
8	y_i	1,54	1,61	1,66	1,71	2,32
9	y_i	0,78	0,39	0,26	0,19	2,79
10	y_i	0,69	0,35	0,23	0,17	2,94
11	y_i	1,05	1,21	1,57	2,42	2,48
12	y_i	6,28	5,62	5,14	4,91	2,59
13	y_i	1,57	1,21	1,11	1,05	2,64
14	y_i	1,11	0,74	0,56	0,44	2,71
15	y_i	1,88	1,54	1,39	1,30	2,12

3 Для табличной функции из задания 2 построить интерполяционный многочлен Ньютона и вычислить значение функции в заданной точке $\bar{x}$. Сравнить результаты.

4 Для заданной функции $f(x)$ построить таблицу значений функции на отрезке $[a, b]$, разбив отрезок на n равных частей, а затем с помощью интерполирования в форме Ньютона найти значение функции в заданной точке $\bar{x}$. Определить погрешность интерполирования.

Таблица 3

Вариант	$f(x)$	a	b	n	$\bar{x}$
1	$\ln x$	1,0	1,6	3	1,55
2	2^x	-0,2	0,4	3	-0,11
3	$\frac{x}{(x^2 + 1)}$	0,6	1,2	3	0,65
4	$\bar{x}$	0,0	0,6	3	0,54
5	$\sin x$	0,1	0,4	3	0,13
6	$\cos x$	-0,1	0,2	3	0,18
7	$\sin x + \cos x$	0,0	0,3	3	0,27
8	$\sqrt{x}$	1,2	1,8	3	1,24
9	$\sqrt[3]{x}$	0,4	1,0	3	0,45
10	$\sqrt[3]{x^2}$	0,5	0,8	3	0,77
11	$\sqrt{x+2}$	-0,9	-0,6	3	-0,65
12	$3^{\sqrt{x}}$	0,1	0,4	3	0,37
13	$2^x \cos x$	0,0	0,3	3	0,12
14	$2^x \sin x$	-0,1	0,2	3	-0,05
15	$x2^x$	0,4	0,7	3	0,63

5. Для функции, заданной таблицей 4, найти приближенное значение первой и второй производной в точке $\bar{x}$.

Таблица 4

x	0,98	1,00	1,02	1,04
y	0,7825	0,7739	0,7651	0,7473

$\bar{x} = 0,98 + 0,001k$, k – номер варианта.

6. По заданной таблице 5 значений функции определить значения аргументов x , соответствующих заданным значениям y .

Таблица 5

x	4	6	9	11	15	17	19
y	16	36	81	121	225	289	361

1) $y = 25$; 2) $y = 49$; 3) $y = 100$; 4) $y = 64$; 5) $y = 144$; 6) $y = 196$; 7) $y = 169$;
8) $y = 55$; 9) $y = 105$; 10) $y = 200$; 11) $y = 300$; 12) $y = 180$; 13) $y = 150$;
14) $y = 130$; 15) $y = 200$.

7. Для функции, заданной таблицей из задания 4, построить кубический сплайн и вычислить значение функции в указанной точке $\bar{x}$.

Вопросы для самоконтроля

- 1 Постановка задачи интерполирования и ее геометрический смысл.
- 2 Теорема о существовании и единственности интерполяционного многочлена.
- 3 Интерполирование в форме Лагранжа. Остаточный член интерполяционной формулы Лагранжа.
- 4 Определение конечных разностей k -го порядка и их свойства. Таблица конечных разностей.
- 5 Интерполяционные формулы Ньютона для случая равноотстоящих узлов интерполяции.
- 6 Определение разделенных разностей k -го порядка и их свойства. Таблица разделенных разностей.
- 7 Интерполяционные формулы Ньютона для случая неравноотстоящих узлов интерполяции.
- 8 Обратная задача теории интерполирования и методы ее решения.
- 9 Численное дифференцирование.
- 10 Приближение функций сплайнами. Интерполяционный сплайн. Степень и дефект сплайна.
- 11 Кубический сплайн. Способы задания наклона сплайна. Погрешность приближения сплайном.

ТЕМА 2. МЕТОД НАИМЕНЬШИХ КВАДРАТОВ

2.1 Основные понятия

Пусть R – пространство функций $f(x)$ интегрируемых с квадратом на отрезке $[a, b]$. Скалярным произведением функций f и g в R называется выражение $(f, g) = \int_a^b f(x)g(x)dx$. Нормой элемента f называется число $\|f\|^2 = \int_a^b f^2(x)dx$. Если для функций f и g выполняется условие $(f, g) = 0$, то говорят, что элементы f и g ортогональны. Полученное таким образом пространство обозначается $L_2[a, b]$. Рассмотрим подпространство H состоящее из линейных комбинаций $n+1$ линейно независимых функций $\varphi_0, \dots, \varphi_n$

$$\varphi = c_0\varphi_0 + c_1\varphi_1 + \dots + c_n\varphi_n. \quad (29)$$

Это подпространство называется *подпространством обобщенных многочленов* по системе функций $\{\varphi_i(x)\}$. Среднеквадратичным отклонением элемента $\varphi \in H$ от элемента $f \in L_2$ называется величина

$$S^2 = \int_a^b (f(x) - \varphi(x))^2 dx.$$

В методе наименьших квадратов приближение элемента $f \in R$ элементом $\bar{\varphi} \in H$ осуществляется таким образом, чтобы величина среднеквадратичного отклонения была минимальна в L_2

$$S^2(c_0, \dots, c_n) = \|f - \bar{\varphi}\| = \inf_{\varphi \in H[a, b]} \|f - \varphi\|. \quad (30)$$

Если такой элемент $\bar{\varphi}$ существует, он называется *элементом наилучшего среднеквадратичного приближения функции $f \in R$ в подпространстве H* .

Теорема 4. Для того, чтобы элемент $\bar{\varphi} \in H$ был элементом наилучшего среднеквадратичного приближения, необходимо и достаточно, чтобы выполнялось условие

$$(f - \bar{\varphi}, \varphi) = 0, \quad (31)$$

для любого $\varphi \in H$.

Доказательство: Необходимость. Пусть существует такой элемент φ_1 , что $(f - \bar{\varphi}, \varphi_1) = \alpha \neq 0$. Без нарушения общности можно считать, что $\|\varphi_1\| = 1$. Возьмем элемент $\varphi_2 = \bar{\varphi} + \alpha\varphi_1$. Тогда имеем

$$\begin{aligned} \|f - \varphi_2\|^2 &= (f - \varphi_2, f - \varphi_2) = (f - \bar{\varphi} - \alpha\varphi_1, f - \bar{\varphi} - \alpha\varphi_1) = \\ &= (f - \bar{\varphi}, f - \bar{\varphi}) - \alpha(\varphi_1, f - \bar{\varphi}) - \alpha(f - \bar{\varphi}, \varphi_1) + \alpha^2(\varphi_1, \varphi_1) = \\ &= \|f - \bar{\varphi}\|^2 - 2\alpha^2 + \alpha^2 = \|f - \bar{\varphi}\|^2 - \alpha^2. \end{aligned}$$

Отсюда следует $\|f - \varphi_2\|^2 < \|f - \bar{\varphi}\|^2$, что невозможно, т. к. $\bar{\varphi}$ — элемент наилучшего приближения.

Достаточность. Пусть $\varphi \in H$. Тогда

$$\begin{aligned} \|f - \varphi\|^2 &= (f - \bar{\varphi} + \bar{\varphi} - \varphi, f - \bar{\varphi} + \bar{\varphi} - \varphi) = \|f - \bar{\varphi}\|^2 + (\bar{\varphi} - \varphi, f - \bar{\varphi}) + \\ &+ (f - \bar{\varphi}, \bar{\varphi} - \varphi) + \|\bar{\varphi} - \varphi\|^2 = \|f - \bar{\varphi}\|^2 + \|\bar{\varphi} - \varphi\|^2. \end{aligned}$$

Если $\varphi \neq \bar{\varphi}$, то получим $\|f - \varphi\|^2 > \|f - \bar{\varphi}\|^2$ для произвольного $\varphi \in H$, т. е. $\bar{\varphi}$ — элемент наилучшего среднеквадратичного приближения в H .

Из теоремы 4, в частности, следует, что если взять в качестве элементов из H последовательно $\varphi_0(x), \varphi_1(x), \dots, \varphi_n(x)$, то из условия ортогональности (31) получим

$$(f - \bar{\varphi}, \varphi_i(x)) = 0, \quad (32)$$

для любого $i = \overline{0, n}$. Систему (32) можно переписать в виде

$$c_0(\varphi_0, \varphi_i) + c_1(\varphi_1, \varphi_i) + \dots + c_n(\varphi_n, \varphi_i) = (f, \varphi_i), \quad i = \overline{0, n}. \quad (33)$$

Нетрудно показать, что матрица системы (33) $A = \{(\varphi_i, \varphi_j)\}$ симметрична, положительно определена и ее определитель, называемый определителем Грамма, для системы линейно независимых функций $\{\varphi_i(x)\}$ отличен от нуля. Следовательно, система (32) имеет единственное решение.

2.2 Класс алгебраических многочленов

Возьмем в качестве H класс алгебраических многочленов

$$P_n(x) = c_0 + c_1x + \dots + c_nx^n.$$

Тогда $(\varphi_i, \varphi_j) = \int_a^b x^i x^j dx$, $(f, \varphi_i) = \int_a^b f(x) x^i dx$. Обозначим $s_i = \int_a^b x^i dx$, $m_i = \int_a^b f(x) x^i dx$. Перепишем систему (33) в виде

$$\begin{cases} c_0 s_0 + c_1 s_1 + \dots + c_n s_n = m_0 \\ c_0 s_1 + c_1 s_2 + \dots + c_n s_{n+1} = m_1 \\ \dots \\ c_0 s_n + c_1 s_{n+1} + \dots + c_n s_{2n} = m_n. \end{cases} \quad (34)$$

Определитель Грамма этой системы всегда положителен и система имеет единственное решение.

Если $\varphi = \bar{\varphi}$, то из (31) следует

$$\|f - \bar{\varphi}\|^2 = \|f\|^2 - \|\bar{\varphi}\|^2. \quad (35)$$

В этом случае, если система функций $\{\varphi_i(x)\}$ ортонормирована, т. е.

$$(\varphi_k, \varphi_l) = \begin{cases} 0, & \text{если } k \neq l \\ 1, & \text{если } k = l \end{cases}, \text{ то система (33) решается в явном виде}$$

$$c_k = (f, \varphi_k), \quad k = \overline{0, n}. \quad (36)$$

При этом погрешность можно вычислить по формуле

$$\|f - \overline{\varphi}\|^2 = \|f\|^2 - \sum_{k=0}^n c_k^2. \quad (37)$$

Соответствующий обобщенный многочлен называется *многочленом Фурье*, а C_k – коэффициентами Фурье элемента $f(x) \in R$ по ортонормированной системе функций $\{\varphi_i(x)\}$.

Примером ортогональной системы функций являются многочлены Лежандра

$$L_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} (x^2 - 1)^n, \quad (38)$$

которые ортогональны на отрезке $[-1, +1]$ и $\|L_n(x)\| = \sqrt{\frac{2}{2n+1}}$. Эти многочлены можно вычислить по рекуррентной формуле

$$(n+1)L_{n+1}(x) - (2n+1)xL_n(x) + nL_{n-1}(x) = 0. \quad (39)$$

Многочленом наилучшего среднеквадратичного приближения по отрезку $[-1, +1]$ будет многочлен

$$P_n(x) = \sum_{k=0}^n c_k L_k(x), \quad (40)$$

$$\text{где } c_k = \frac{2n+1}{2} \int_{-1}^{+1} f(x) L_k(x) dx.$$

Отметим, что ряд Фурье по многочленам Лежандра для любой непрерывной функции $f(x)$ на отрезке $[-1, +1]$ сходится равномерно.

Для приближения на интервале $(-l, l)$ периодической функции с периодом $2l$ удобно использовать ортогональную систему функций $\left\{ \frac{1}{2}, \cos \frac{\pi k x}{l}, \sin \frac{\pi k x}{l} \right\}$. Обобщенным многочленом наилучшего приближения тогда будет тригонометрический ряд Фурье

$$\Phi(x) = \frac{a_0}{2} + \sum_{k=0}^N \left(a_k \cos \frac{\pi k x}{l} + b_k \sin \frac{\pi k x}{l} \right), \quad (41)$$

$$\text{где } a_k = \frac{1}{l} \int_{-l}^l f(x) \cos \frac{\pi k x}{l} dx, \quad b_k = \frac{1}{l} \int_{-l}^l f(x) \sin \frac{\pi k x}{l} dx.$$

Если функция $f(x)$ задана на интервале $(0, l)$, то коэффициенты дискретного ряда Фурье (41) вычисляются по формулам

$$a_k = \frac{2}{l} \int_0^l f(x) \cos \frac{\pi k x}{l} dx, \quad b_k = \frac{2}{l} \int_0^l f(x) \sin \frac{\pi k x}{l} dx.$$

2.3 Точечная среднеквадратичная аппроксимация функций на отрезке

Пусть на отрезке $[a, b]$ для функции $f(x)$ известны только ее значения в $n+1$ точках $x_0, x_1, \dots, x_n$. Будем искать в классе обобщенных многочленов такой, чтобы величина квадратичного отклонения

$$S^2(c_0, \dots, c_m) = \sum_{i=0}^n (f(x_i) - \varphi(x_i))^2 \quad (42)$$

принимала наименьшее возможное значение. Если

$$\varphi(x) = c_0 \varphi_0 + c_1 \varphi_1 + \dots + c_m \varphi_m,$$

то из необходимого условия минимума $\frac{\partial S}{\partial c_i} = 0$ получим систему вида (33)

$$\begin{cases} c_0 s_0 + c_1 s_1 + \dots + c_m s_m = t_0 \\ c_0 s_1 + c_1 s_2 + \dots + c_m s_{m+1} = t_1 \\ \dots \\ c_0 s_m + c_1 s_{m+1} + \dots + c_m s_{2m} = t_n \end{cases}, \quad (43)$$

$$\text{где } s_k = (\varphi_i, \varphi_j) = \sum_{l=0}^n \varphi_i(x_l) \varphi_j(x_l),$$

$$i + j = k,$$

$$t_k = (f, \varphi_k) = \sum_{l=0}^n f(x_l) \varphi_k(x_l).$$

Если в качестве системы $\{\varphi_i(x)\}$ берется система $1, x, \dots, x^m$, то приближение ищется в классе алгебраических многочленов степени m

$$P_n(x) = c_0 + c_1 x + \dots + c_n x^n.$$

В этом случае коэффициенты системы (43) определяются по формулам

$$s_k = \sum_{l=0}^n x_l^k, \quad t_k = \sum_{l=0}^n x_l^k f(x_l).$$

Определитель Грамма системы (43) положителен и, следовательно, система имеет единственное решение.

Замечание. В том случае, когда $m=n$ и приближение ищется в классе алгебраических многочленов, приближение дает интерполяционный многочлен Лагранжа $L_n(x)$, т. к. для него величина среднеквадратичного отклонения равна нулю: $S^2 = \sum_{i=0}^n (f(x_i) - L_n(x_i))^2 \equiv 0$.

2.3.1 Пример программной реализации точечного метода

Для таблицы значений с неравностоящими узлами построить многочлен третьей степени методом наименьших квадратов и найти значение функции в заданной точке $\bar{x} = 2,2$.

x	1,0	1,3	1,8	2,2
y	1,683	0,500	6,403	29,113

Ниже приводится возможная программа на языке Паскаль ABC, реализующая поставленную задачу с фиксированным вводом для четырех точек.

```

Program МНКточ;
Const  n=4; m=4;
type mas1=array[1..m] of real;
      mas2=array[1..m,1..m] of real;
var i,j,k,kod, ll: integer;
x,y: array [1..n] of real ;
p,r: mas1 ;
s: mas2 ;
x0,det,s1 : real;
function stepx(a:real;ll:integer):real; //возведение в степень
var i: integer;
    q : real;
begin  q:=1;
      for i:=1 to ll do
        q:=q*a;
      Result:=q;
end;
Procedure GAUSS(a:mas2;b:mas1; n:integer; var p:real;var
x:mas1);
var
  nom:array[1..10] of integer;
  c,s,p1,max:real;
  i,j,k,m,l:integer;
begin
  writeln ('    Процедура GAUSS');
  for l:=1 to n do nom[l]:=l; //Исходная нумерация неизвестных
  for i:=1 to n-1 do {ведущий элемент ставится на диагональ}
    begin {и запоминается нумерация перемещенных столбцов}
      max:=abs(a[i,i]); m:=i; k:=i;
      for j:=i to n do
        if max<abs(a[i,j]) then
          begin max:=abs(a[i,j]); k:=j; end;
      if m<k then
        begin
          for l:=1 to n do
            begin c:=A[l,m]; A[l,m]:=A[l,k]; A[l,k]:=c; end;
          kod:=nom[m]; nom[m]:=nom[k]; nom[k]:=kod ;
        end;
      end;
  p:=1; {начальное значение определителя}
  {прямой ход метода Гаусса}
  for k:=1 to n do
    begin
      c:=a[k,k]; p1:=c; {p1 - текущий диагональный элемент }
      if abs(c)=0 then
        begin writeln ('Метод некорректен, ведущий элемент =0');
          exit
        end;
    end;

```

```

    for j:=k to n do           {деление на ведущий элемент}
        a[k,j]:=a[k,j]/c;
        b[k]:=b[k]/c;
    for i:=k+1 to n do        {преобразование остальных строк мат-
рицы}
        begin
            c:=a[i,k];
            for j:=k to n do
                a[i,j]:=a[i,j]-a[k,j]*c;
            b[i]:=b[i]-b[k]*c
        end;
        p:=p*p1;               {нахождение определителя}
    end;
{Обратный ход}
x[n]:=b[n];
    for i:=n-1 downto 1 do
        begin
            s:=0;
            for j:=i+1 to n do
                s:=s+a[i,j]*x[j];
            x[i]:=b[i]-s;
        end;
    for i:=1 to n do b[nom[i]]:=x[i];
    for i:=1 to n do x[i]:=b[i];
end;
begin
x[1]:=1; x[2]:=1.3; x[3]:=1.8; x[4]:=2.2;
y[1]:=1.683; y[2]:=0.50; y[3]:=6.403; y[4]:=29.113;
    for j:=1 to m do begin    // формирование системы
        for k:=1 to m do begin
            s[j,k]:=0;
            for i:=1 to n do begin
                ll:=j+k-2;
                s[j,k]:=s[j,k]+stepx(x[i],ll);
            end
        end; r[j]:=0;
        for i:=1 to n do
            r[j]:=r[j]+y[i]*stepx(x[i],j-1);
        end;
        writeln('Исходные данные');
        write('x[j]= ');
    for j:=1 to n do
        write(x[j]:8:4); writeln;
        write('y[j]= ');
    for j:=1 to n do
        write(y[j]:8:4); writeln;
        writeln('Система уравнений');
        for j:=1 to m do begin
            write('j=',j,' ');
            for k:=1 to m do write(s[j,k]:10:3);
            write(' ',r[j]:10:4); writeln;
        end;
        GAUSS(s,r,m,det,p);

```

```

writeln ('Решение исходной системы p:');
for i:=1 to m do write (p[i]:5:3, ' ');
writeln('    det= ',det:7:4);
writeln ('Проверка решения исходной системы');
  for j:=1 to m do begin
    s1:=0;
    for k:=1 to m do
      s1:=s1+s[j,k]*p[k];
      writeln('j=',j,s1:10:3, ' = ',r[j]:8:3);
    end;

    writeln;
    writeln('Полином точечного МНК ');
    x0:=2.2;
    write('FI(x)= ');
    for j:=1 to m-1 do
      if j=1 then write(' ',p[m-j+1]:3:5, '*x^',m-j)
      else if p[m-j+1]>=0.0 then write(' + ',p[m-j+1]:3:5, '*x^',m-
j)
        else write(' ',p[m-j+1]:3:5, '*x^',m-j);
      if p[1]>=0.0 then write(' + ',p[1]:3:5)
        else write(' ',p[1]:3:5);
      writeln;
      s1:=p[m]*x0;
      for i:=m-1 downto 2 do
        s1:=(s1+p[i])*x0;
        s1:=s1+p[1];
      write('Значение полинома FI(',x0:5:3,')= ',s1:8:5);
    end.

```

Протокол решения задачи имеет вид

Окно вывода					
Исходные данные					
x[j]=	1.0000	1.3000	1.8000	2.2000	
y[j]=	1.6830	0.5000	6.4030	29.1130	
Система уравнений					
j=1	4.000	6.300	10.770	19.677	37.6990
j=2	6.300	10.770	19.677	37.779	77.9070
j=3	10.770	19.677	37.779	75.145	164.1806
j=4	19.677	37.779	75.145	153.219	350.1190
Процедура GAUSS					
Решение исходной системы p:					
-27.825	88.042	-83.766	25.232	det=	0.0027
Проверка решения исходной системы					
j=1	37.699	=	37.699		
j=2	77.907	=	77.907		
j=3	164.181	=	164.181		
j=4	350.119	=	350.119		
Полином точечного МНК					
FI(x)= 25.23241*x^3 -83.76620*x^2 + 88.04163*x^1 -27.82483					
Значение полинома FI(2.200)= 29.11300					

2.4 Задания по среднеквадратическому приближению функций

1 Для заданной функции $y = f(x)$ построить полином второй степени, приближающий $f(x)$ с наименьшим квадратичным отклонением на отрезке $[a, b]$.

Таблица 6

Вариант	$f(x)$	a	b
1	e^x	0	1
2	$\frac{1}{(x+1)}$	1	2
3	$\frac{1}{(1+x^2)}$	0	1
4	$\sin x$	0	π
5	$\frac{x}{(1+x)}$	1	2
6	$\ln x$	1	2
7	$\cos x$	0	π
8	$x^3 - 2x^2$	0	2
9	$\frac{x}{(1+x^2)}$	0	1
10	$x^2 - 4x$	-1	1
11	2^x	0	1
12	$3x^3 + x$	1	2
13	$\frac{2}{(x+3)}$	0	1
14	$\frac{x^2}{(1+x^2)}$	0	1
15	$2x + 5x^2$	-1	2

2 Для функции $f(x)$, заданной таблично, с помощью метода наименьших квадратов построить аппроксимирующий многочлен второй степени.

Таблица 7

Вариант	x_i	1	2	3	4
1	y_i	-1	0	2	4
2	y_i	1	9	10	16
3	y_i	-4	0	2	7
4	y_i	4	1	0	2
5	y_i	1	3	3	1
6	y_i	-1	-2	0	5
7	y_i	1	1	0	-1
8	y_i	-3	3	-3	3
9	y_i	0	3	8	15
10	y_i	-2	0	2	15
11	y_i	5	-1	1	2
12	y_i	1	3	-2	-3
13	y_i	-3	0	1	4
14	y_i	2	3	4	-1
15	y_i	-2	-1	0	2

3 Для функции $y = f(x)$, заданной на отрезке $[-\pi, \pi]$, с помощью метода наименьших квадратов построить обобщенный аппроксимирующий многочлен, взяв в качестве ортогональной системы систему тригонометрических функций $1, \sin x, \cos x, \sin 2x, \cos 2x$.

Таблица 8

Вариант	$f(x)$
1	2
1	$y = x$
2	$y = x $
3	$y = e^x$
4	$y = e^{-x}$
5	$y = 1 - x$
6	$y = x^2$
7	$y = x + 1$
8	$y = 3x + 1$
9	$y = 1 + \sin x$
10	$y = 1 + \cos x$

Окончание таблицы 8

1	2
11	$y = 1 + \sin 2x$
12	$y = 1 + \cos 2x$
13	$y = \sin x + \cos 2x$
14	$y = \sin x + \cos x$
15	$y = \cos 2x + \sin x + 1$

Вопросы для самоконтроля

1 Евклидовы пространства. Примеры.

2 Нормированные пространства. Примеры.

3 Линейно зависимые и линейно независимые системы функций.
Определитель Грамма. Ортогональные системы функций.

4 Обобщенный многочлен. Многочлен наилучшего среднеекватра-
тичного приближения. Теорема о существовании и единственности
многочлена наилучшего среднеекватратичного приближения.

5 Интегральная квадратичная аппроксимация функций на от-
резке $[a, b]$.

6 Точечная квадратичная аппроксимация функций.

7 Приближение функций дискретным рядом Фурье.

ЛИТЕРАТУРА

1 Амосов, А. А. Вычислительные методы для инженеров / А. А. Амосов, Ю. А. Дубинский, Н. В. Копченова // Научная библиотека. – Текст : электронный. – 2023. – URL: https://scask.ru/i_book_clm.php?id=1 (дата обращения: 03.05.2025).

2 Бахвалов, Н. С. Численные методы : учебник / Н. С. Бахвалов, Н. П. Жидков, Г. М. Кобельков. – 12-е изд. – М. : Лаборатория знаний, 2024. – 639 с.

3 Бахвалов, Н. С. Численные методы. Решения задач и упражнения : учебное пособие для вузов / Н. С. Бахвалов, А. А. Корнев, Е. В. Чижонков. – М. : Лаборатория знаний, 2016. – 355 с.

4 Березин, И. С. Методы вычислений / И. С. Березин, Н. П. Жидков // Научная библиотека. – 2023. – Текст : электронный. – Т. 1. – URL: https://scask.ru/i_book_calc1.php?id=1 (дата обращения: 03.05.2025).

5 Березин, И. С. Методы вычислений / И. С. Березин, Н. П. Жидков // Научная библиотека. – Текст : электронный. – 2023. – Т. 2. – URL: https://scask.ru/i_book_calc2.php?id=1 (дата обращения: 03.05.2025).

6 Березовская, Е. М. Методы вычислений : тексты лекций : в 2 ч. / Е. М. Березовская, М. И. Жадан. – Текст : электронный. – Гомель : ГГУ им. Ф. Скорины, 2010. – Ч. 1 : Интерполирование и нелинейные уравнения. – 80 с. – URL: <http://elib.gsu.by/handle/123456789/1589> (дата обращения: 03.05.2025).

7 Березовская, Е. М. Методы численного анализа : тексты лекций : в 2 ч. / Е. М. Березовская. – Текст : электронный. – Гомель : ГГУ им. Ф. Скорины, 2007. – Ч. 1 : Интерполяция и интегрирование. – 131 с. – URL <http://elib.gsu.by/handle/123456789/3523> (дата обращения: 03.05.2025).

8 Калиткин, Н. Н. Численные методы / Н. Н. Калиткин // Научная библиотека. – Текст : электронный. – 2023. – URL: https://scask.ru/q_book_dig_m.php?id=1 (дата обращения: 03.05.2025).

9 Киреев, В. И. Численные методы в примерах и задачах : учебное пособие / В. И. Киреев, А. В. Пантелеев. – М. : Лань, 2015. – 448 с.

10 Колдаев, В. Д. Численные методы и программирование : учебное пособие / В. Д. Колдаев. – М. : Форум, 2018. – 336 с.

11 Копченова, Н. В. Вычислительная математика в примерах и задачах : учебное пособие / В. Н. Копченова, И. А. Марон. – 2-е изд., стер. – СПб ; М. ; Краснодар : Лань, 2008. – 368 с.

12 Поршнев, С. В. Численные методы на базе Mathcad / С. В. Поршнев, И. В. Беленкова. – СПб. : БХВ-Петербург, 2005. – 464 с.

Производственно-практическое издание

Березовская Елена Михайловна

**МЕТОДЫ ВЫЧИСЛЕНИЙ:
ПРИБЛИЖЕНИЕ ФУНКЦИЙ**

Практическое пособие

Редактор Е. С. Балашова
Корректор В. В. Калугина

Подписано в печать 16.06.2025. Формат 60x84 1/16.
Бумага офсетная. Ризография.
Усл. печ. л. 2,56. Уч.-изд. л. 2,80.
Тираж 10 экз. Заказ 343.

Издатель и полиграфическое исполнение:
учреждение образования
«Гомельский государственный университет имени Франциска Скорины».
Специальное разрешение (лицензия) № 02330 / 450 от 18.12.2013 г.
Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий в качестве:
издателя печатных изданий № 1/87 от 18.11.2013 г.;
распространителя печатных изданий № 3/1452 от 17.04.2017 г.
Ул. Советская, 104, 246028, Гомель.

