

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Кафедра радиофизики, полупроводниковой микро- и наноэлектроники

**В.В. Зайцев**

**ЧИСЛЕННЫЕ МЕТОДЫ ДЛЯ ФИЗИКОВ.  
ПРИБЛИЖЕНИЕ ФУНКЦИЙ И ОБРАБОТКА ДАННЫХ**

*Утверждено редакционно-издательским советом университета  
в качестве учебного пособия*

Издательство «Самарский университет»

2014

УДК 519.6  
ББК 22.3  
317

Рецензенты: докт. физ.-мат. наук, проф. О.В. Осипов,  
докт. физ.-мат. наук, проф. А.Н. Степанов.

**Зайцев В.В.**

317 Численные методы для физиков. Приближение функций и обработка данных: учебное пособие / В. В. Зайцев. – Самара : Изд-во «Самарский университет», 2014. – 64 с.

Учебное пособие посвящено одному из разделов курса численных методов – интерполяции и аппроксимации функций, численному интегрированию и дифференцированию. Изложение проведено на «физическом» уровне строгости. Основное внимание уделено описанию численных алгоритмов и ограничениям и проблемам, возникающим при их применении.

Приведены примеры реализации численных алгоритмов с использованием пакета MathCAD.

Предназначено для студентов физического факультета СамГУ, изучающих курс «Численные методы и математическое моделирование».

УДК 519.6  
ББК 22.3

© Зайцев В.В., 2014

© Самарский государственный  
университет, 2014

© Оформление. Изд-во «Самарский университет», 2014

## ПРЕДИСЛОВИЕ

Учебное пособие написано на основе лекций, читаемых автором студентам-физикам Самарского государственного университета, и посвящено одному из разделов курса численных методов – приближению функций и обработке экспериментальных данных.

В пособии приведены основные сведения о приближении функций методами интерполяции и наименьших квадратов, необходимые для их практического использования при анализе математических моделей систем различной физической природы. Рассмотрены как традиционные, классические численные методы, так и методы, вошедшие в вычислительную практику сравнительно недавно.

Изложение проводится на «физическом» уровне строгости. Математические обоснования большинства методов даны на основе элементарных результатов математического анализа и должны быть понятны студентам младших курсов. Основное внимание уделено практической стороне использования численных методов, а также ограничениям и проблемам, возникающим при их применении. Там, где это возможно, описание математических результатов сопровождается физической интерпретацией процессов, лежащих в основе численных алгоритмов.

Приведены многочисленные примеры реализации рассмотренных численных алгоритмов с использованием пакета MathCAD. Предполагается, что студенты прослушали курс программирования и имеют навыки составления программ. Детальный разбор программных модулей MathCAD, имеющих «прозрачную» структуру, часто способных заменить стандартные блок-схемы алгоритмов, рекомендуется для наиболее полного понимания сути изучаемых методов. Кроме того, использование MathCAD дает возможность студентам совершенствоваться в научном программировании.

# ГЛАВА 1. ИНТЕРПОЛЯЦИЯ

## 1.1. Введение

Теория приближения функций рассматривает методы замены исходных, часто достаточно сложных зависимостей более простыми функциями с хорошо изученными свойствами. При этом приближаемая функция может быть задана как в аналитическом виде, так и в форме таблицы. Различны также критерии, которыми руководствуются при построении приближения.

Интерполяция – один из наиболее распространенных способов приближения функций. Задача интерполяции состоит в следующем. Пусть заданы последовательности значений аргумента  $x_0, x_1, \dots, x_N$  и соответствующих им значений функции  $y = f(x)$ :  $y_0, y_1, \dots, y_N$ . На плоскости эти последовательности отображаются совокупностью точек  $(x_n, y_n)$ . Требуется построить такую функцию  $g(x)$ , график которой проходит через все заданные точки. Функция  $g(x)$  носит название интерполяционной функции (интерполянта). Ее вид задается, либо исходя из физических соображений о природе зависимости  $y = f(x)$ , либо – из формальных соображений, например, о простоте вычисления.

После построения интерполяционной функции  $g(x)$  появляется возможность вычисления значений функции  $y = f(x)$  при значениях аргумента, не входящих в таблицу. Что касается самой таблицы, то она может содержать:

1. результаты эксперимента;
2. результаты вычислений по некоторому алгоритму;
3. выборку значений аналитически заданной функции.

Среди множества типов интерполяционных функций на практике наибольшее распространение получили полиномы. Однако полиномиальная интерполяция не единственный возможный способ интерполяции. Часто в качестве интерполянтов целесообразно использовать тригонометрические функции и тригонометрические полиномы. В области цифровой обработки сигналов успешно применяются дробно-рациональные функции.

## 1.2. Полиномиальная интерполяция. Метод неопределенных коэффициентов

Интерполяционный полином  $P_N(x)$  степени  $N$ , описывающий кривую, проходящую через  $N+1$  задающую точку  $(x_n, y_n)$ , имеет вид

$$P_N(x) = a_N x^N + a_{N-1} x^{N-1} + \dots + a_1 x + a_0 = \sum_{m=0}^N a_m x^m. \quad (1.1)$$

Во всех узлах интерполяции  $x_n$  он удовлетворяет условиям  $P_N(x_n) = y_n$ , что можно записать в виде совокупности равенств

$$\sum_{m=0}^N a_m x_n^m = y_n, \quad n = 0, 1, 2, \dots, N, \quad (1.2)$$

представляющих собой систему линейных алгебраических уравнений (СЛАУ) относительно коэффициентов полинома. Матрица системы (1.2)

$$V[x_n^N] = \begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^N \\ 1 & x_1 & x_1^2 & \dots & x_1^N \\ 1 & x_2 & x_2^2 & \dots & x_2^N \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_N & x_N^2 & \dots & x_N^N \end{bmatrix}$$

носит название матрицы Вандермонда. Если все  $x_n$  различны между собой, то ее определитель отличен от нуля. Следовательно, СЛАУ (1.2) всегда имеет единственное решение. На основании этого можно утверждать, что всегда существует единственный интерполяционный полином вида (1.1).

Метод решения задачи интерполяции, в котором коэффициенты  $a_n$  вычисляются непосредственным решением СЛАУ (1.2), называется методом неопределенных коэффициентов. Метод полезен для теоретических рассуждений, но при практических вычислениях он используется редко, т.к. не обладает высокой вычислительной эффективностью. Кроме того, даже при не слишком высоких порядках полинома  $P_N(x)$  определитель Вандермонда имеет малую величину. Например, в случае эквидистантного расположения точек  $x_n$  на отрезке  $[0,1]$  определитель принимает значения:  $\det(V[x_n^4]) = 2.75 \cdot 10^{-4}$ ,  $\det(V[x_n^5]) = 1.13 \cdot 10^{-6}$ ,  $\det(V[x_n^6]) = 1.13 \cdot 10^{-9}$ . Малая величина определителя является признаком плохой обусловленности матрицы Вандермонда. При этом вычисление коэффициентов  $a_n$  путем решения СЛАУ (1.2) сопровождается большой вычислительной погрешностью.

Кратко остановимся на вопросе о точности полиномиальной интерполяции. Пусть полином  $P_N(x)$  интерполирует функцию  $f(x)$ , которая опре-

делена на всем интересующем нас отрезке  $[x_0, x_N]$ , но ее значения известны только на дискретном множестве точек  $x_n - y_n = f(x_n)$ . Тогда, если функция  $f(x)$  имеет  $N+1$  непрерывную производную и удовлетворяются условия (1.1), то для любого  $x$  из указанного интервала

$$f(x) - P_N(x) = \frac{(x - x_0)(x - x_1)\dots(x - x_N)}{(N + 1)!} f^{(N+1)}(z),$$

где  $z$  – некоторая точка интервала интерполяции. Модуль представленной разности можно принять в качестве оценки погрешности интерполяции. Нетрудно показать, что для случая, когда узлы интерполяции расположены равномерно с шагом  $h = x_{n+1} - x_n$ ,

$$|f(x) - P_N(x)| \leq Mh^{N+1}, \quad (1.3)$$

где  $M = \max_{x_0 \leq z \leq x_N} |f^{(N+1)}(z)| / (N + 1)!$ .

Хотя на практике оценку (1.3) сложно использовать из-за входящей в нее неизвестной величины  $M$ , в теории указание на порядок погрешности по шагу  $h$  часто является весьма полезным.

Методы построения полинома  $P_N(x)$  можно разделить на три группы: метод Лагранжа, разностные и итерационные методы. Но еще раз следует обратить внимание на то, что независимо от способа построения интерполяционный полином является единственным полиномом степени  $N$ , удовлетворяющим условиям (1.2).

### 1.3. Интерполяция по Лагранжу

В методе Лагранжа интерполяционный полином представляется в виде

$$P_N(x) = y_0 L_0(x) + y_1 L_1(x) + \dots + y_N L_N(x),$$

где  $L_n(x)$  также есть полиномы степени  $N$ . Они носят название полиномов Лагранжа и подбираются так, чтобы

$$L_n(x_m) = \delta_{n,m}.$$

При этом условия (1.1) выполняются автоматически.

Полиномы Лагранжа  $L_n(x)$  в общем случае имеют вид

$$L_n(x) = C_n (x - x_0)(x - x_1)\dots(x - x_{n-1})(x - x_{n+1})\dots(x - x_N).$$

Коэффициенты  $C_n$  определяются из условий  $L_n(x_n) = 1$ :

$$C_n = 1 / (x_n - x_0)(x_n - x_1)\dots(x_n - x_{n-1})(x_n - x_{n+1})\dots(x_n - x_N).$$

С учетом этих выражений полиномы Лагранжа записываются как

$$L_n(x) = \prod_{\substack{m=0 \\ m \neq n}}^N \frac{x - x_m}{x_n - x_m},$$

а интерполяционный полином в форме Лагранжа – как

$$P_N(x) = \sum_{n=0}^N y_n \prod_{\substack{m=0 \\ m \neq n}}^N \frac{x - x_m}{x_n - x_m}. \quad (1.4)$$

Вычисления интерполяционного полинома Лагранжа можно реализовать с помощью следующей MathCAD-программы.

$$\text{PolLag}(x, X, Y) := \left| \begin{array}{l} N \leftarrow \text{length}(X) - 1 \\ \text{for } n \in 0..N \\ \quad \left| \begin{array}{l} L_n \leftarrow 1 \\ \text{for } m \in 0..N \\ \quad \left| \begin{array}{l} L_n \leftarrow L_n \cdot \frac{x - X_m}{X_n - X_m} \text{ if } m \neq n \\ L_n \text{ otherwise} \end{array} \right. \end{array} \right. \\ P \leftarrow \sum_{n=0}^N Y_n \cdot L_n \end{array} \right.$$

Существенным недостатком интерполяции по Лагранжу является невысокая вычислительная эффективность алгоритма (1.4). Кроме того, вычислительные сложности возникают при изменении количества узловых точек. Если, например, мы хотим добавить к набору данных узловую точку  $(x_{N+1}, y_{N+1})$  и составить полином степени  $N+1$ , интерполирующий эти данные, то нам придется пересчитать все полиномы Лагранжа.

#### 1.4. Итерационный метод интерполяции

Итерационные методы интерполяции основаны на многократном применении некоторой сравнительно простой интерполяционной схемы. Наиболее известным из них является метод Эйткена, в котором на каждом шаге итерации используется линейная интерполяция.

Линейная интерполяция между точками  $(x_0, y_0)$  и  $(x_m, y_m)$  проводится по формуле

$$P_{1m}(x) = \frac{1}{x_m - x_0} [y_0(x_m - x) - y_m(x_0 - x)], \quad m = 1, 2, \dots, N. \quad (1.5)$$

С ее помощью можно получить  $N$  полиномов первой степени  $P_{1m}(x)$ .

На втором шаге итерации функции (1.5) используются для построения  $N-1$  полинома второй степени

$$P_{2m}(x) = \frac{1}{x_m - x_1} [P_{11}(x)(x_m - x) - P_{1m}(x)(x_1 - x)], \quad m = 2, 3, \dots, N, \quad (1.6)$$

которые описывают кривые, проходящие через точки  $(x_0, y_0), (x_1, y_1), (x_m, y_m)$ , в чем нетрудно убедиться прямой подстановкой.

Далее, на третьем шаге, с использованием функций (1.6) получим  $N-2$  полинома третьей степени

$$P_{3m}(x) = \frac{1}{x_m - x_2} [P_{22}(x)(x_m - x) - P_{2m}(x)(x_2 - x)], \quad m = 3, 4, \dots, N,$$

задающих кривые, проходящие через точки  $(x_0, y_0), (x_1, y_1), (x_2, y_2), (x_m, y_m)$ .

Продолжая итерационный процесс, можно в конечном итоге построить полином  $P_n(x)$ , проходящий через все узлы интерполяции.

Следующая MathCAD-программа реализует алгоритм метода Эйткена.

$$\text{PolAit}(x, X, Y) := \left| \begin{array}{l} N \leftarrow \text{length}(X) - 1 \\ \text{for } m \in 0..N \\ \quad P_{0,m} \leftarrow Y_m \\ \text{for } n \in 1..N \\ \quad \text{for } m \in n..N \\ \quad \quad P_{n,m} \leftarrow \frac{P_{n-1,n-1} \cdot (X_m - x) - P_{n-1,m} \cdot (X_{n-1} - x)}{X_m - X_{n-1}} \\ \quad P_{N,N} \end{array} \right.$$

Следует отметить, что метод Эйткена, как и метод Лагранжа, не требует того, чтобы узлы интерполяции располагались с постоянным шагом по аргументу  $x$ .

## 1.5. Разностный метод Ньютона

Среди разностных методов интерполяции наиболее широко распространен метод Ньютона интерполирования вперед, известный также как метод Ньютона – Грегори.

В этом методе интерполяционный полином записывается в виде

$$P_N(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + \dots + c_N(x - x_0)(x - x_1)\dots(x - x_{N-1}).$$

Тогда система уравнений (1.2) для нахождения коэффициентов  $c_i$  выглядит следующим образом:

$$\begin{aligned}
 c_0 &= y_0, \\
 c_0 + c_1(x_1 - x_0) &= y_1, \\
 c_0 + c_1(x_1 - x_0) + c_2(x_2 - x_0)(x_2 - x_1) &= y_2, \\
 &\dots\dots\dots \\
 c_0 + c_1(x_1 - x_0) + \dots + c_N(x_N - x_0)(x_N - x_1)\dots(x_N - x_{N-1}) &= y_N.
 \end{aligned}
 \tag{1.7}$$

Система (1.7) имеет треугольную матрицу коэффициентов и поэтому расчет значений  $c_i$  не вызывает затруднений и производится последовательной подстановкой. Однако существует и более формальный способ вычисления  $c_i$ , основанный на использовании правых разностей.

Пусть значения аргументов заданы с постоянным шагом  $h = x_{i+1} - x_i$ . Тогда система уравнений (1.7) записывается в виде

$$\begin{aligned}
 y_0 &= c_0, \\
 y_1 &= c_0 + c_1h, \\
 y_2 &= c_0 + 2c_1h + 2c_2h^2, \\
 &\dots\dots\dots \\
 y_n &= c_0 + Nc_1h + N(N-1)c_2h^2 + \dots + N!c_Nh^N.
 \end{aligned}
 \tag{1.8}$$

Из первых двух уравнений системы (1.8) находим

$$\begin{aligned}
 c_0 &= y_0, \\
 c_1 &= \frac{y_1 - c_0}{h} = \frac{y_1 - y_0}{h} = \frac{\Delta y_0}{h}.
 \end{aligned}$$

Здесь  $\Delta y_0$  называется первой правой разностью. Продолжая вычисления, получим

$$c_2 = \frac{1}{2h^2}(y_2 - c_0 - 2hc_1) = \frac{1}{2h^2}[(y_2 - y_1) - (y_1 - y_0)] = \frac{1}{2h^2}\Delta\Delta y_0 = \frac{\Delta^2 y_0}{2h^2}.$$

Разность  $\Delta^2 y_0$  носит название второй правой разности. В результате последовательных вычислений приходим к выражению

$$c_n = \frac{\Delta^n y_0}{n!h^n}.$$

Для вычисления разностей высоких порядков нетрудно получить следующее рекуррентное соотношение:

$$\Delta^i y_j = \Delta^{i-1} y_{j+1} - \Delta^{i-1} y_j \text{ при } j = 0, 1, \dots, N - i.
 \tag{1.9}$$

С помощью правых разностей интерполяционный полином в представлении Ньютона записывается в виде

$$P_n(x) = y_0 + \frac{(x-x_0)}{h} \Delta y_0 + \frac{(x-x_0)(x-x_1)}{2h^2} \Delta^2 y_0 + \dots + \frac{(x-x_0)(x-x_1)\dots(x-x_{N-1})}{N!h^N} \Delta^N y_0. \quad (1.10)$$

Отметим, что это представление (1.10) имеет определенное сходство с первыми  $N+1$  слагаемыми в ряде Тейлора для функций  $y(x)$  в точке  $x_0$ .

Предположим теперь, что к набору данных необходимо добавить новую точку  $(x_{N+1}, y_{N+1})$ . Тогда интерполирующий новую совокупность данных полином Ньютона примет вид

$$P_{N+1}(x) = P_N(x) + \frac{(x-x_0)(x-x_1)\dots(x-x_N)}{(N+1)!h^{N+1}} \Delta^{N+1} y_0.$$

Выражаемое этой формулой свойство представления Ньютона делает его во многих случаях особенно полезным на практике.

Следующая MathCAD-программа реализует вычисления значений полинома (1.10).

$$\text{PolNew}(x, X, Y) := \left| \begin{array}{l} N \leftarrow \text{length}(X) - 1 \\ h \leftarrow X_1 - X_0 \\ \text{for } j \in 0..N \\ \quad \Delta_{0,j} \leftarrow Y_j \\ \text{for } i \in 1..N \\ \quad \text{for } j \in 0..N-i \\ \quad \quad \Delta_{i,j} \leftarrow \Delta_{i-1,j+1} - \Delta_{i-1,j} \\ \text{for } n \in 1..N \\ \quad C_n \leftarrow \frac{\Delta_{n,0}}{n! \cdot h^n} \\ \text{PN} \leftarrow Y_0 + \sum_{n=1}^N C_n \cdot \prod_{k=0}^{n-1} (x - X_k) \end{array} \right.$$

Если новая узловая точка добавляется в начало последовательности данных, то для интерполяционного полинома можно воспользоваться представлением Ньютона с левыми разностями. В этом случае

$$P_N(x) = b_0 + b_1(x_N - x) + b_2(x_N - x)(x_{N-1} - x) + \dots + b_N(x_N - x)(x_{N-1} - x)\dots(x_1 - x).$$

Для поиска коэффициентов  $b_n$  нетрудно получить систему уравнений, аналогичную системе (1.8):

$$\begin{aligned} y_N &= b_0, \\ y_{N-1} &= b_0 + b_1h, \\ y_{N-2} &= b_0 + 2b_1h + 2b_2h^2, \\ &\dots \\ y_0 &= b_0 + Nb_1h + N(N-1)b_2h^2 + \dots + N!b_Nh^N. \end{aligned}$$

Первые два уравнения дают

$$\begin{aligned} b_0 &= y_N, \\ b_1 &= -\frac{y_N - y_{N-1}}{h} = -\frac{\nabla y_N}{h}. \end{aligned}$$

Здесь  $\nabla y_N$  является первой левой разностью. Затем находим выражение

$$\begin{aligned} b_2 &= \frac{1}{2h^2}(y_{N-2} - b_0 - 2hb_1) = \frac{1}{2h^2}[(y_N - y_{N-1}) - (y_{N-1} - y_{N-2})] = \\ &= \frac{1}{2h^2}\nabla\nabla y_N = \frac{\nabla^2 y_N}{2h^2} \end{aligned}$$

со второй левой разностью  $\nabla^2 y_N$ . Дальнейшие последовательные вычисления приводят к общей формуле

$$b_n = \frac{(-1)^n \nabla^n y_n}{n!h^n}.$$

При этом устанавливается, что левые разности различных порядков связаны рекуррентным соотношением

$$\nabla^i y_j = \nabla^{i-1} y_j - \nabla^{i-1} y_{j-1} \text{ при } j = N, N-1, \dots, N-i.$$

С использованием левых разностей интерполяционный полином Ньютона записывается в форме

$$\begin{aligned} P_N(x) &= y_N - \frac{(x_N - x)}{h}\nabla y_N + \frac{(x_N - x)(x_{N-1} - x)}{2h^2}\nabla^2 y_N + \dots + \\ &+ \frac{(-1)^N (x_N - x)(x_{N-1} - x)\dots(x_1 - x)}{N!h^N}\nabla^N y_N. \end{aligned}$$

При добавлении новой точки в начало последовательности данных можно построить интерполяционный полином

$$P_{N+1}(x) = P_N(x) + \frac{(-1)^{N+1}(x_N - x)(x_{N-1} - x)\dots(x_1 - x)(x_0 - x)}{(N+1)!h^{N+1}} \nabla^{N+1} y_N.$$

Помимо непосредственного решения задач интерполяции полиномы в представлении Ньютона оказываются также полезными при построении алгоритмов численного дифференцирования и решения систем обыкновенных дифференциальных уравнений.

## 1.6. Кубические сплайны

В рассмотренных до сих пор методах интерполяционный полином имел один и тот же вид на всем отрезке  $[x_0, x_N]$ . Однако такой подход в теории интерполяции не является единственным. Не менее распространен способ, при котором интерполяция осуществляется посредством кусочных полиномов, т.е. полиномов, различающихся между собой на некотором количестве подобластей, в сумме перекрывающих всю область интерполяции. В качестве таких элементарных подобластей можно взять все отрезки  $I_n = [x_n, x_{n+1}]$  и потребовать, чтобы в точках сопряжения кусочные полиномы имели не только одинаковые значения  $y_n$ , но и равные первую и вторую производные. Нетрудно показать, что этим условиям удовлетворяет совокупность полиномов степени не ниже третьей.

Группа сопряженных кубических полиномов, в точках сопряжения которых первая и вторая производные непрерывны, называется кубическим сплайном. На каждом отрезке  $I_n$  кубический сплайн  $C(x)$  представляется в виде

$$C(x) = C_n(x) = a_{3n}x^3 + a_{2n}x^2 + a_{1n}x + a_{0n}, \quad n = 0, 1, \dots, N-1. \quad (1.11)$$

Для задания сплайна (1.11) необходимо определить  $4N$  коэффициентов. Это можно сделать путем решения системы из  $4N$  линейных алгебраических уравнений. Из них  $3N-3$  уравнения дают условия непрерывности:

$$C_{n-1}(x_n) = C_n(x_n), \quad C'_{n-1}(x_n) = C'_n(x_n), \quad C''_{n-1}(x_n) = C''_n(x_n); \quad n = 1, 2, \dots, N-1. \quad (1.12)$$

Еще  $N+1$  уравнение получается из требования равенства функции  $C(x)$  в узлах интерполяции заданным значениям:

$$\begin{aligned} C_n(x_n) &= y_n, \quad n = 0, 1, \dots, N-1 \\ C_{N-1}(x_N) &= y_N. \end{aligned} \quad (1.13)$$

Таким образом, для однозначного задания кубического сплайна необходимо еще два дополнительных условия. Обычно используются равенства

$$C''_0(x_0) = 0, \quad C''_{N-1}(x_N) = 0. \quad (1.14)$$

Полученный таким образом сплайн называется естественным кубическим сплайном.

Хотя непосредственное решение системы уравнений (1.12)-(1.14) не связано с какими-либо принципиальными затруднениями, этот метод построения кубического сплайна нельзя признать эффективным. Путем специального выбора вида кубических многочленов (1.11) можно значительно упростить задачу.

Для этого запишем полиномы (1.11) в форме

$$C_n(\xi_n) = y_n(1 - \xi_n) + y_{n+1}\xi_n + h_n[(A_n - d_n)\xi_n(1 - \xi_n)^2 - (A_{n+1} - d_n)\xi_n^2(1 - \xi_n)], \quad (1.15)$$

где

$$\xi_n = \frac{x - x_n}{h_n}, \quad d_n = \frac{y_{n+1} - y_n}{h_n}, \quad h_n = x_{n+1} - x_n; \quad n = 0, 1, \dots, N - 1.$$

При такой записи, как нетрудно убедиться прямой подстановкой, в (1.11) условия непрерывности функции  $C(x)$  и ее первой производной и условия (1.12) выполняются автоматически. Любое из равенств (1.15) содержит только два неизвестных коэффициента:  $A_n$  и  $A_{n+1}$ . При переходе к рассмотрению каждого следующего равенства добавляется лишь один новый неизвестный коэффициент. Поэтому в совокупности система равенств (1.15) содержит  $N+1$  неизвестный коэффициент. Для их определения следует сформировать систему уравнений, исходя из требований выполнения условий непрерывности вторых производных в (1.15) и условий (1.14).

Непрерывность вторых производных дает уравнения

$$A_{n-1}h_n + 2A_n(h_{n-1} + h_n) + A_{n+1}h_{n-1} = 3(d_{n-1}h_n + d_n h_{n-1}), \quad n = 1, 2, \dots, N - 1, \quad (1.16)$$

а из условий (1.13) имеем

$$2A_0 + A_1 = 3d_0, \quad A_{N-1} + 2A_N = 3d_{N-1}. \quad (1.17)$$

Полученная система линейных алгебраических уравнений (1.16) и (1.17) имеет трехдиагональную матрицу в левой части. Для решения таких уравнений разработан весьма эффективный метод прогонки.

Для проведения кубической сплайн-интерполяции можно воспользоваться следующей MathCAD-программой.

```

Spline3(x, X, Y) :=
  N ← length(X) - 1
  for i ∈ 0..N - 1
    | hi ← Xi+1 - Xi
    | Di ←  $\frac{Y_{i+1} - Y_i}{h_i}$ 
  A0,0 ← 2
  A0,1 ← 1
  B0 ← 3·D0
  for i ∈ 1..N - 1
    | Ai,i ← 2·(hi-1 + hi)
    | Ai,i-1 ← hi
    | Ai,i+1 ← hi-1
    | Bi ← 3·(Di-1·hi + Di·hi-1)
  AN,N-1 ← 1
  AN,N ← 2
  BN ← 3·DN-1
  a ← Progon(A, B)
  for i ∈ 0..N - 1
    n ← i if Xi ≤ x < Xi+1
    ξ ←  $\frac{x - X_n}{h_n}$ 
  C ← Yn·(1 - ξ) + Yn+1·ξ + hn· $\left[ (a_n - D_n)·ξ·(1 - ξ)^2 - (a_{n+1} - D_n)·ξ^2·(1 - ξ) \right]$ 

```

Часто оказывается, что кубические сплайны интерполируют функции лучше, чем полиномы высоких степеней, особенно вблизи границ области интерполяции. Благодаря свойству двукратной дифференцируемости кубические сплайны широко применяются при численном интегрировании дифференциальных уравнений. Существуют и другие сплайны, определяемые при отличных от (1.13) условиях на границах или с использованием многочленов более высоких степеней.

## 1.7. Примеры полиномиальной интерполяции

При измерении амплитудно-частотной характеристики колебательно-го контура были получены результаты, сведенные в таблицу 1.1. Ее первая строка содержит номер эксперимента, вторая – значение частоты внешнего воздействия в килогерцах, а третья – значение отношения амплитуды вынужденных колебаний к амплитуде возбуждения.

Таблица 1.1

$n$	0	1	2	3	4	5	6	7	8	9	10
$f$ , кГц	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5
$A$	1.33	1.55	1.92	2.65	4.36	7.02	3.81	2.12	1.39	1.02	0.79

График таблично заданной амплитудной характеристики контура показан точками на рис. 1.1. На этом же рисунке непрерывной линией построен график интерполяционного полинома  $P_{10}(f)$ .

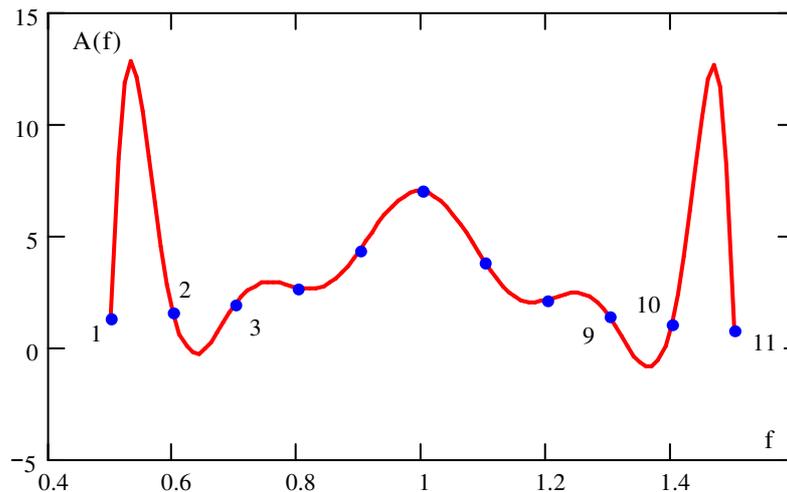


Рис. 1.1

Как видно из рисунка, график полинома, действительно, проходит через все узлы интерполяции. Но это, пожалуй, его единственное достоинство. Интерполяционный полином  $P_{10}(f)$  не соответствует нашим представлениям об амплитудно-частотной характеристике, как функции с одним максимумом (на резонансной частоте, расположенной вблизи 1 кГц). В противоположность этому,  $P_{10}(f)$ , осциллируя, достигает побочных максимумов, причем, размах осцилляций увеличивается при приближении к границам интервала интерполяции.

Впервые осцилляции интерполяционного полинома в сходной ситуации обнаружил Карл Рунге. В 1901 году он пытался интерполировать по-

линомами  $P_N(x)$  сравнительно простую функцию

$$f(x) = \frac{1}{1 + 25x^2},$$

равномерно расположив абсциссы узлов интерполяции на отрезке  $[-1,1]$ . Было установлено, что осцилляции интерполянта (явление Рунге) имеют наибольший размах вблизи границ интервала интерполяции и он возрастает при увеличении степени полинома.

Влияние явления Рунге можно ослабить неравномерным расположением абсцисс узлов интерполяции, в частности, концентрируя их ближе к границам интервала. Однако более простым решением на практике является кусочно-полиномиальная интерполяция – метод, основанный на разбиении отрезка  $[x_0, x_N]$  на ряд подинтервалов и использовании на каждом из них интерполяционных полиномов невысокой степени. Метод кусочно-полиномиальной интерполяции иллюстрирует рис. 1.2, на котором показаны графики трех интерполяционных полиномов. Один из них – полином четвертой степени – построен по узлам таблицы 1 с номерами  $n = 0 \div 4$ . Другой – полином второй степени – по узлам с номерами  $n = 4 \div 6$ . И, наконец, узлы с номерами  $n = 6 \div 9$  интерполируются полиномом третьей степени.

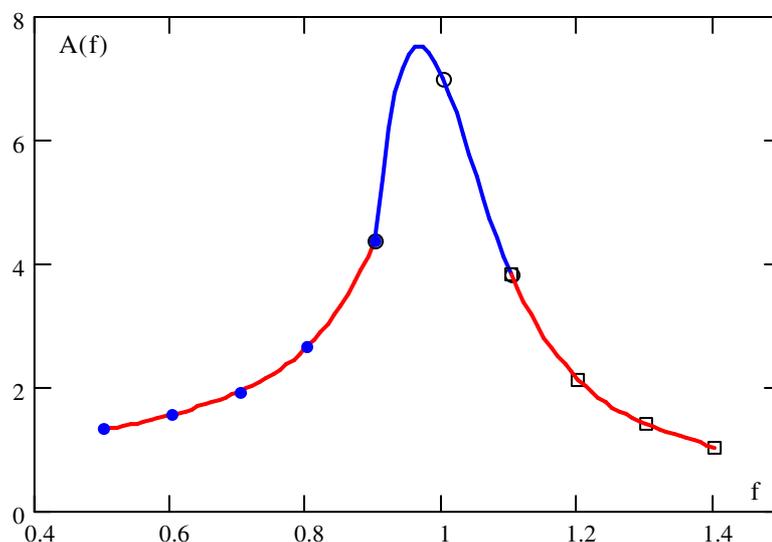


Рис. 1.2

Из рис. 1.2 видно, что осцилляций интерполянта удалось избежать и его график в значительно большей степени, чем предыдущий, соответствует истинной амплитудно-частотной характеристике колебательного контура. Тем не менее, слишком значительная асимметрия кривой и отчетливо прослеживаемый разрыв производной на частоте  $f = 0.9$  кГц не позволяют признать результат интерполяции удовлетворительным.

Обратимся теперь к интерполяции кубическими сплайнами. Ее результат в виде графика зависимости  $A = C_3(f)$ , рассчитанной с помощью приведенной в п. 1.6 программы, представлен на рис. 1.3. Этот унимодальный график вполне соответствует физическим представлениям о частотной характеристике линейного колебательного контура и может быть признан вполне удовлетворительным.

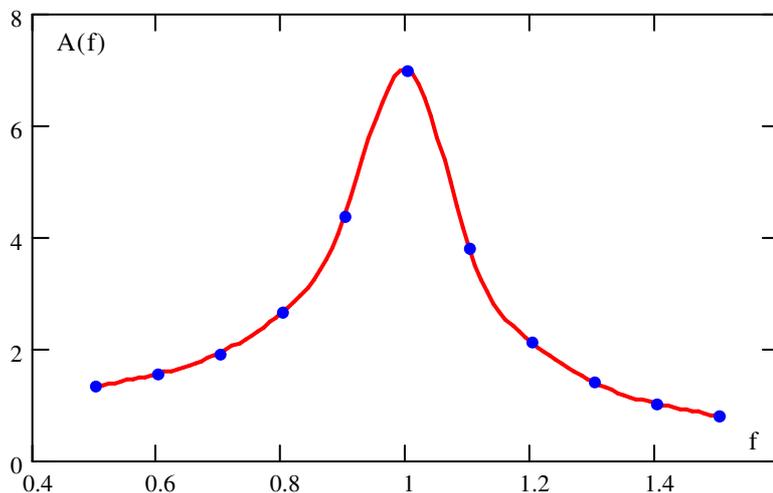


Рис. 1.3

В заключение приведем результаты численного эксперимента, косвенно подтверждающего единственность интерполяционного полинома и указывающего на влияние арифметических погрешностей на результат его вычислений по различающимся алгоритмам. На рис. 1.4 показан модуль разности значений полиномов  $P_{10}(f)$ , интерполирующих таблицу 1 по алгоритму Лагранжа и алгоритму Ньютона с правыми разностями. А на рис. 1.5 – модуль разности значений полиномов в формах Лагранжа и Эйткена.

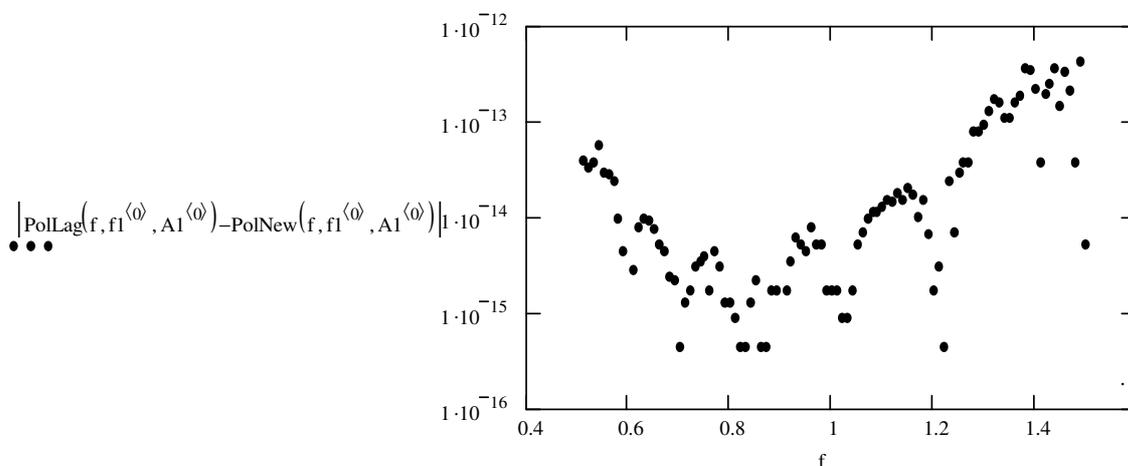


Рис. 1.4

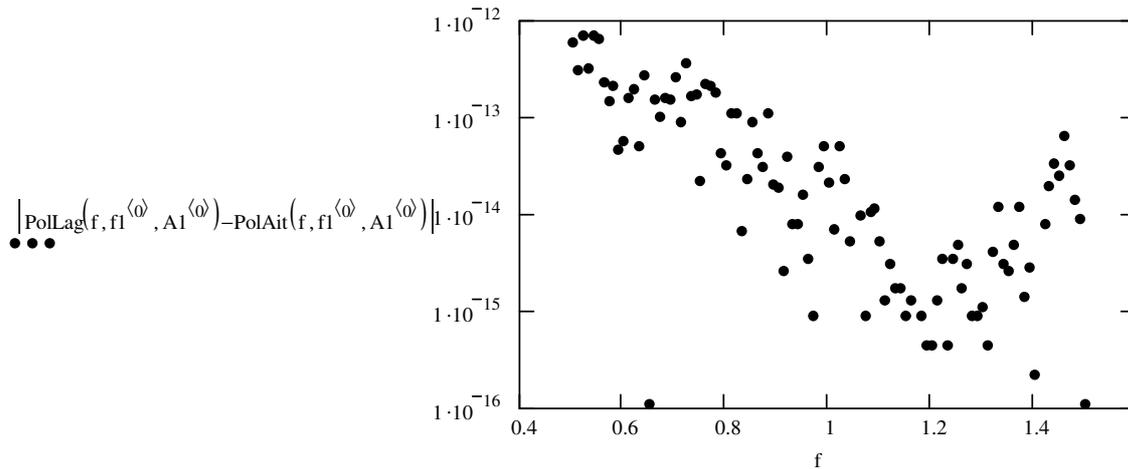


Рис. 1.5

Малая величина разности – на уровне «шума вычислительного алгоритма» – является подтверждением исходного тезиса о единственности интерполяционного полинома.

### 1.8. Интерполяция В-сплайнами

В ряде случаев для интерполяции полезно иметь возможность использовать набор функций, каждая из которых локализована на соответствующем интервале, т.е. принимает нулевые значения, если аргумент не принадлежит интервалу локализации. В частности, при решении граничных задач для дифференциальных уравнений такой способ интерполяции решений позволяет свести задачи к системам алгебраических уравнений с ленточными матрицами, а для них разработаны весьма эффективные численные методы. Кубические базисные сплайны – сокращенно В-сплайны – дают такую возможность.

Кубическим В-сплайном называется полином третьей степени, принимающий на сетке узлов  $x_0, x_1, \dots, x_N$  с шагом  $h$  значения

$$B_n(x_n) = 1, B_n(x_{n\pm 1}) = 0.25$$

и равный нулю в других узлах. Явная запись этого сплайна с аргументом  $x/h$ , локализованного на отрезке  $[-2, 2]$ , в форме MathCAD-функции имеет вид

$$B(x) := \begin{cases} x \leftarrow -x & \text{if } x < 0 \\ B \leftarrow \frac{1}{4} \cdot \left[ 1 + 3 \cdot (1 - x) + 3 \cdot (1 - x)^2 - 3 \cdot (1 - x)^3 \right] & \text{if } 0 \leq x \leq 1 \\ B \leftarrow \frac{1}{4} \cdot (2 - x)^3 & \text{if } 1 < x \leq 2 \\ B \leftarrow 0 & \text{otherwise} \\ B \end{cases}$$

Его график показан на рис. 1.6.

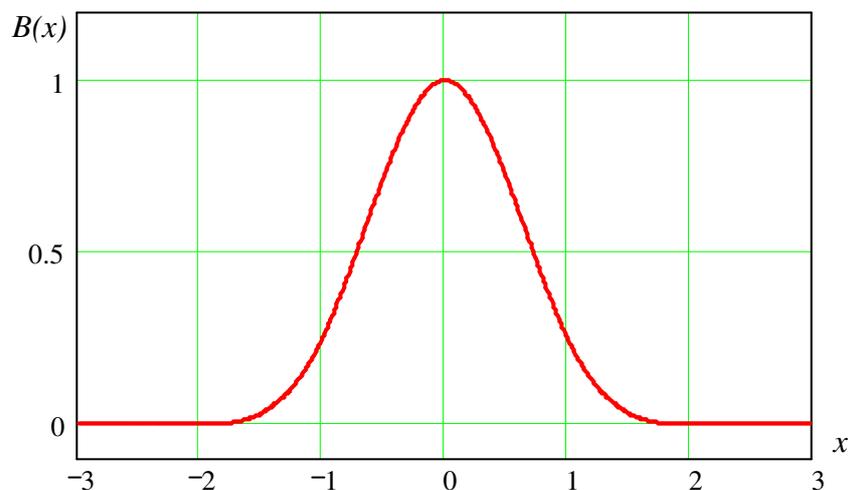


Рис. 1.6

Нетрудно выяснить, что кубический В-сплайн удовлетворяет условиям

$$B_n''(x_{n\pm 2}) = B_n'(x_{n\pm 2}) = B_n(x_{n\pm 2}) = 0.$$

Интерполянт для таблицы исходных данных  $(x_0, y_0), (x_1, y_1), \dots, (x_N, y_N)$  на основе В-сплайнов запишем в виде

$$g(x) = \sum_{n=0}^N a_n B_n(x - x_n).$$

Коэффициенты  $a_n$ , как нетрудно понять, определяются решением СЛАУ

$$a_0 + 0.25a_1 = y_0,$$

$$0.25a_0 + a_1 + 0.25a_2 = y_1,$$

...

$$0.25a_{N-2} + a_{N-1} + 0.25a_N = y_{N-1},$$

$$a_{N-1} + 0.25a_N = y_N.$$

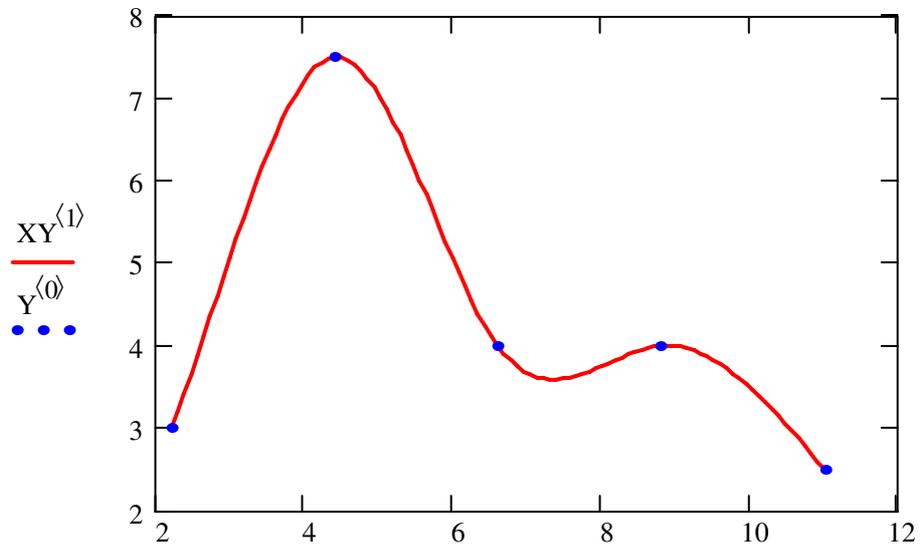
Для интерполяции В-сплайнами можно воспользоваться следующей MathCAD-программой. Программа возвращает значения интерполянта на сетке из  $Nx$  значений аргумента.

```

BSpline(Nx, X, Y) :=
  N ← length(X) - 1
  a ← min(X)
  b ← max(X)
  A ← identity(N + 1)
  for n ∈ 0..N - 1
    |
    | An, n+1 ← 1/4
    | An+1, n ← An, n+1
  β ← Progon(A, Y)
  h ← X1 - X0
  Ni ← Nx
  Ni ← 0 if a = b
  Δx ← (b - a) / Nx
  for i ∈ 0..Ni
    |
    | xi ← a + i · Δx
    | I ← floor((xi - X0) / h)
    | L ← max(I - 1, 0)
    | H ← min(I + 2, N)
    | yi ← ∑_{n=L}^H βn · B((xi - Xn) / h)
  augment(x, y)

```

$XY := \text{BSpline}(100, X, Y)$



*Рис. 1.7*

Пример интерполяции табличной функции В-сплайнами приведен на рис. 1.7.

## ГЛАВА 2. АППРОКСИМАЦИЯ МЕТОДОМ НАИМЕНЬШИХ КВАДРАТОВ

### 2.1. Введение

Рассмотренные ранее методы интерполяции решали задачу о нахождении полинома, график которого проходит через заданные узловые точки. При этом системе равенств (1.1) удовлетворяет единственный полином степени  $N$ .

Предположим теперь, что значения  $y_m$  получены в результате эксперимента и могут содержать ошибки измерений. Задача аппроксимации состоит в подборе достаточно простой функции, применимой во всем диапазоне табличных значений, но не обязательно проходящей через все заданные точки. Естественно, что при этом должен быть определен критерий близости значений аппроксимирующей функции в узловых точках к исходным данным. Обычно в качестве такого критерия выбирают минимум суммы квадратов разностей между значениями функции и соответствующими табличными значениями. Этот метод аппроксимации называется методом наименьших квадратов.

Метод наименьших квадратов широко применяется из статистических соображений. Дело в том, что при наличии погрешностей обычно проводят гораздо больше измерений, чем необходимо, например, для построения интерполяционного полинома степени  $N$ , надеясь, что в результате аппроксимации произойдет усреднение ошибок измерений. Методами теории вероятностей можно показать, что при нормальном законе распределения ошибок метод наименьших квадратов действительно обладает усредняющими свойствами. Кроме того, этот метод весьма прост и элегантен с математической точки зрения.

### 2.2. Общая формулировка метода наименьших квадратов

Пусть в таблице задана  $M+1$  точка  $(x_0, y_0), (x_1, y_1), \dots, (x_M, y_M)$  и в диапазоне значений аргумента  $x_0 \leq x \leq x_M$  требуется найти такую аппроксимирующую функцию  $g(x; a_0, a_1, \dots, a_M)$ , что сумма квадратов погрешностей (целевая функция), определяемая выражением

$$E(a_0, a_1, \dots, a_N) = \sum_{m=0}^M [g(x_m; a_0, a_1, \dots, a_N) - y_m]^2, \quad (2.1)$$

принимает минимальное значение в пространстве параметров  $a_0, a_1, \dots, a_N$ .

Если аппроксимирующая функция  $g(x; a_0, a_1, \dots, a_N)$  нелинейно зависит от параметров аппроксимации  $a_0, a_1, \dots, a_N$ , то такой, наиболее общий вари-

ант метода, носит название нелинейного метода наименьших квадратов (НМНК). Расчеты по НМНК требуют применения компьютерных программ минимизации функций многих переменных или решения систем нелинейных уравнений. Как правило, это технически сложная задача. Но для небольшого числа  $N$  – она вполне разрешима.

В качестве примера применим НМНК для аппроксимации амплитудно-частотной характеристики колебательного контура по данным, представленным в таблице 1.1. При этом, исходя из физических соображений, в качестве аппроксимирующей функции  $g(x; a_0, a_1, \dots, a_N)$  выберем теоретическую зависимость

$$A(f) = \frac{f_0^2}{\sqrt{(f^2 - f_0^2)^2 + f^2 f_0^2 / Q^2}}. \quad (2.2)$$

Здесь  $f_0$  и  $Q$  – собственная частота и добротность колебательного контура.

Вариант MathCAD-программы вычислений по НМНК выглядит следующим образом.

$$\begin{aligned} \text{EF}(a, b, x, y, F) &:= \left| \begin{array}{l} M \leftarrow \text{length}(x) - 1 \\ E \leftarrow \sum_{m=0}^M (F(x_m, a, b) - y_m)^2 \end{array} \right. \\ g(x, a, b) &:= a^2 \cdot \left[ \sqrt{(x^2 - a^2)^2 + \frac{a^2 \cdot x^2}{b^2}} \right]^{-1} \quad E(a, b) := \text{EF}(a, b, f, A, g) \\ a &:= 2 \quad b := 4 \quad ab := \text{Minimize}(E, a, b) \\ ab^T &= (1 \quad 7) \quad E(ab_0, ab_1) = 6.209 \times 10^{-11} \end{aligned}$$

В результате вычислений получены значения собственной частоты  $f_0 = 1$  и добротности  $Q = 7$  (с точностью трех знаков после запятой). Минимальное значение целевой функции  $E(1, 7) = 6.209 \times 10^{-11}$  указывает на то, что данные таблицы 1.1 имеют малую погрешность измерений и адекватно представляются моделью (2.2). Этот вывод подтверждается также показанным на рис. 2.1 непрерывной линией графиком функции (2.2) с вычисленными значениями  $f_0$  и  $Q$ . Точками на рисунке показаны узлы аппроксимации из таблицы 1.1.

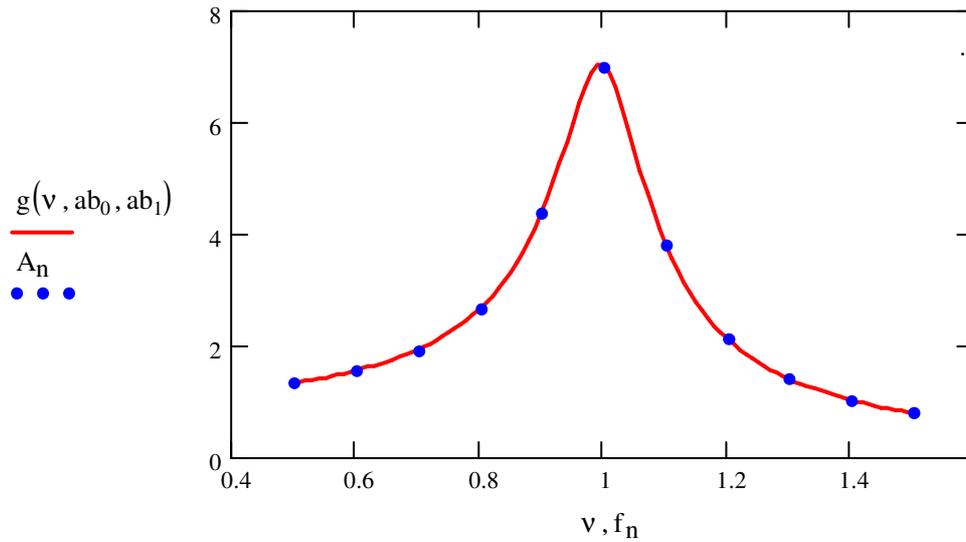


Рис. 2.1

Теперь искусственно понизим точность измерений, добавив к значениям амплитуды в третьей строке таблицы 11. случайную погрешность  $\xi_m$ , равномерно распределенную в интервале  $[-0.3, 0.3]$ . График сформированной таким образом «экспериментальной» амплитудно-частотной характеристики показан на рис. 2.2 точками. Погрешность эксперимента заметна визуально. Расчеты по НМНК в этом случае дают значения  $f_0 = 0.998$  и  $Q = 6.885$  с минимальным значением целевой функции  $E(0.998, 6.885) = 0.405$ . Соответствующий данным значениям параметров аппроксимации график функции (2.2) изображен на рис. 2.2 непрерывной линией. Усредняющее свойство метода наименьших квадратов проявляется здесь вполне отчетливо: несмотря на значительную погрешность данных, погрешность параметров невелика.

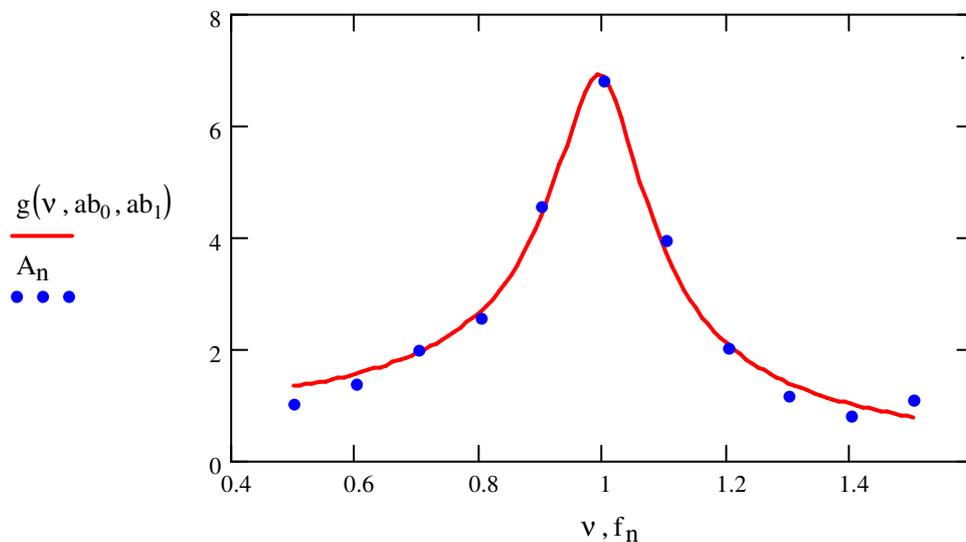


Рис. 2.2

Отметим, что в тех случаях, когда функция  $g(x; a_0, a_1, \dots, a_N)$  выбирается на основе физической модели, расчет значений параметров аппроксимации является основной целью НМНК. Так, в приведенном примере мы установили, что колебательный контур имеет собственную частоту, близкую к значению  $f_0 = 1$ , и добротность – к значению  $Q = 7$ .

Качественно оценить результаты аппроксимации и интерполяции амплитудно-частотной характеристики колебательного контура можно, сопоставив графики аппроксиманта и интерполянта, как это сделано на рис. 2.3. Видно, что в данной ситуации аппроксимация дает более плавный график, чем интерполяция. Этот вывод сохраняет справедливость и в общем случае.

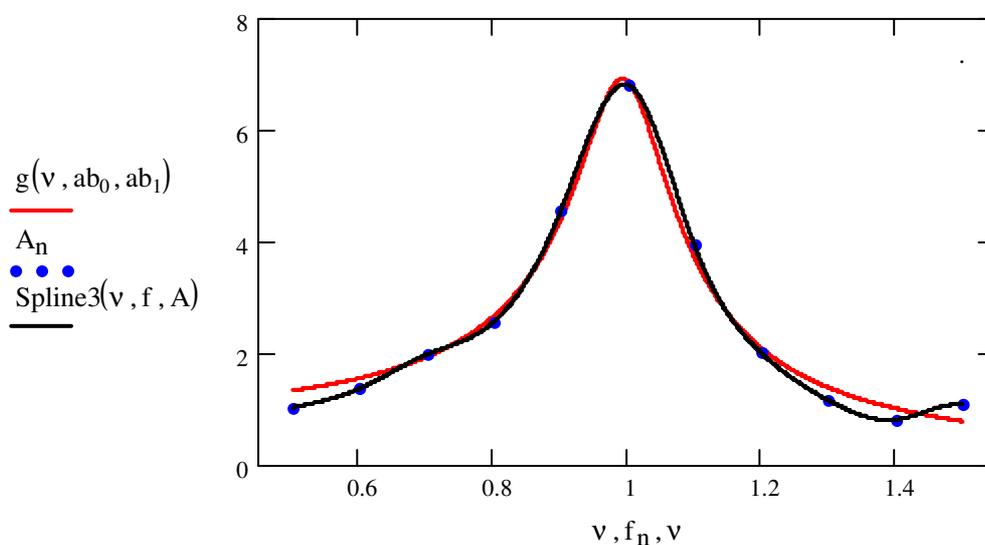


Рис. 2.3

Таким образом, при решении физических задач нужно стремиться применять методы аппроксимации с использованием моделей исследуемых объектов. К сожалению, следовать этому правилу не всегда удастся.

### 2.3. Линейный метод наименьших квадратов

В противоположность нелинейному, линейный метод наименьших квадратов (ЛМНК) оперирует с представлением функции  $g(x; a_0, a_1, \dots, a_N)$  в виде линейной комбинации базисных функций:

$$g(x; a_0, a_1, \dots, a_N) = a_0 g_0(x) + a_1 g_1(x) + \dots + a_N g_N(x). \quad (2.3)$$

При этом подбор функций  $g_n(x)$  осуществляется с учетом либо физических соображений о происхождении табличных значений, либо такой их специфики, как периодичность, релаксация, наличие симметрии или асимптотики и прочее.

Условия минимума целевой функции определяются равенствами

$$\frac{\partial E}{\partial a_0} = \frac{\partial E}{\partial a_1} = \dots = \frac{\partial E}{\partial a_N} = 0.$$

С учетом выражений (2.1) и (2.3) они эквивалентны системе уравнений относительно параметров аппроксимации

$$\sum_{m=0}^M g_n(x_m) [a_0 g_0(x_m) + a_1 g_1(x_m) + \dots + a_N g_N(x_m) - y_m] = 0, \quad n = 0, 1, \dots, N.$$

Эту систему линейных алгебраических уравнений можно записать также в векторно-матричной форме:

$$\mathbf{G}\mathbf{a} = \mathbf{b}, \quad (2.4)$$

где  $\mathbf{a} = (a_0, a_1, \dots, a_N)^T$  – вектор-столбец неизвестных параметров аппроксимации, матрица Грама  $\mathbf{G}$  составлена из базисных функций:

$$\mathbf{G} = \begin{pmatrix} \sum g_1^2(x_i) & \sum g_1(x_m)g_2(x_m) & \dots & \sum g_1(x_m)g_N(x_m) \\ \dots & \dots & \dots & \dots \\ \sum g_1(x_m)g_N(x_m) & \dots & \dots & \sum g_N^2(x_m) \end{pmatrix},$$

а вектор-столбец  $\mathbf{b}$  помимо базисных функций содержит аппроксимируемые данные:

$$\mathbf{b} = \begin{pmatrix} \sum g_1(x_m)y_m \\ \dots \\ \sum g_N(x_m)y_m \end{pmatrix}.$$

Здесь все суммы вычисляются в пределах от 0 до  $M$ .

СЛАУ (2.4) носит название системы нормальных уравнений. Нормальные уравнения весьма полезны для теоретических исследований. В частности, установлено, что, если функции  $g_n(x)$  в (2.3) образуют линейно независимый базис, то матрица Грама невырождена и система нормальных уравнений имеет единственное решение. Однако при практических вычислениях нормальные уравнения рекомендуется использовать только для малых  $N$ , поскольку уже для  $N > 5$  матрица Грама в левой части системы (2.4) становится плохо обусловленной, что приводит к большой погрешности в решении системы.

Иногда исходную таблицу значений разбивают на несколько частей и подбирают отдельную аппроксимирующую функцию для каждой части. Однако такой подход оправдан в лишь тех случаях, когда есть основания полагать, что различные части аппроксимируемых данных соответствуют разным физическим состояниям системы.

## 2.4. Полиномиальная аппроксимация методом наименьших квадратов

Если в качестве базисных функций в выражении (2.2) взять степенные функции:

$$g_n(x) = x^n,$$

то аппроксимирующая функция становится полиномом степени  $N$ :

$$P_N(x) = g(x; a_0, a_1, \dots, a_N) = a_0 + a_1x + \dots + a_Nx^N. \quad (2.5)$$

Система нормальных уравнений (2.4) для случая полиномиальной аппроксимации принимает вид

$$\begin{pmatrix} M+1 & \sum x_m & \dots & \sum x_m^N \\ \sum x_m & \sum x_m^2 & \dots & \sum x_m^{N+1} \\ \dots & \dots & \dots & \dots \\ \sum x_m^N & \sum x_m^{N+1} & \dots & \sum x_m^{2N} \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ \dots \\ a_N \end{pmatrix} = \begin{pmatrix} \sum y_m \\ \sum x_m y_m \\ \dots \\ \sum x_m^N y_m \end{pmatrix}. \quad (2.6)$$

Здесь все суммы вычисляются в пределах от 0 до  $M$ . Можно показать, что если  $M \geq N$  и среди точек  $x_m$  нет совпадающих, то решение системы (2.6) определяет единственный, минимизирующий сумму (2.1), набор коэффициентов полинома (2.5). Таким образом, задача полиномиальной аппроксимации по методу наименьших квадратов имеет единственное решение.

Приведем пример расчета коэффициентов аппроксимирующего поли-

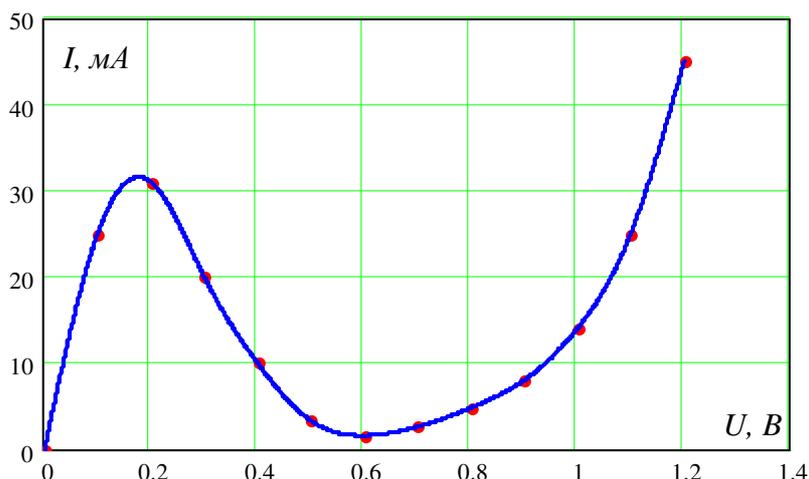


Рис. 2.4

нома на основе решения системы нормальных уравнений (2.6). На рис. 2.4 показана взятая из справочника вольт-амперная характеристика (ВАХ) туннельного диода типа АИ202И. Точками на графике отмечены узлы ап-

проксимации. Соответствующие им значения напряжения и тока сведены в таблицу 2.1.

Таблица 2.1

$n$	0	1	2	3	4	5
$U, В$	0.0	0.1	0.2	0.3	0.4	0.5
$I, мА$	0.0	25.0	31.0	20.0	10.0	3.3

$n$	6	7	8	9	10	11	12
$U, В$	0.6	0.7	0.8	0.9	1.0	1.1	1.2
$I, мА$	1.6	2.6	4.8	8.0.0	14.0	25.0	45.0

Расчет коэффициентов аппроксимирующего полинома можно проводить с помощью следующей MathCAD-программы.

```

ApproPol(N, X, Y) :=
  M ← length(X) - 1
  for n ∈ 0..N
    for k ∈ n..N
      Gn,k ← ∑m=0M (Xm)n+k
      Gk,n ← Gn,k if k ≠ n
    bn ← ∑m=0M (Xm)n · Ym
  a ← lsolve(G, b)
  
```

Результаты аппроксимации нормированной ВАХ полиномом пятой степени отражает рис. 2.5. Отметим, что после того, как определены коэффициенты полинома, его значения целесообразно вычислять по схеме Горнера. Она состоит в организации вычислений в соответствии с записью

$$P_N(x) = a_0 + x(a_1 + x(a_2 + \dots x(a_{N-1} + xa_N)))$$

Схема Горнера реализуется MathCAD-программой.

```

Horner(x, a) :=
  N ← length(a) - 1
  P ← aN
  for n ∈ 1..N
    P ← aN-n + P · x
  P
  
```

$$a := \text{ApproPol}(N, X, Y) \quad \text{Pa}(U) := \text{Horner}(U, a)$$

$$a^T = \left( -6.39 \times 10^{-3} \quad 9.08 \quad -44.66 \quad 81 \quad -64.26 \quad 19.13 \right)$$

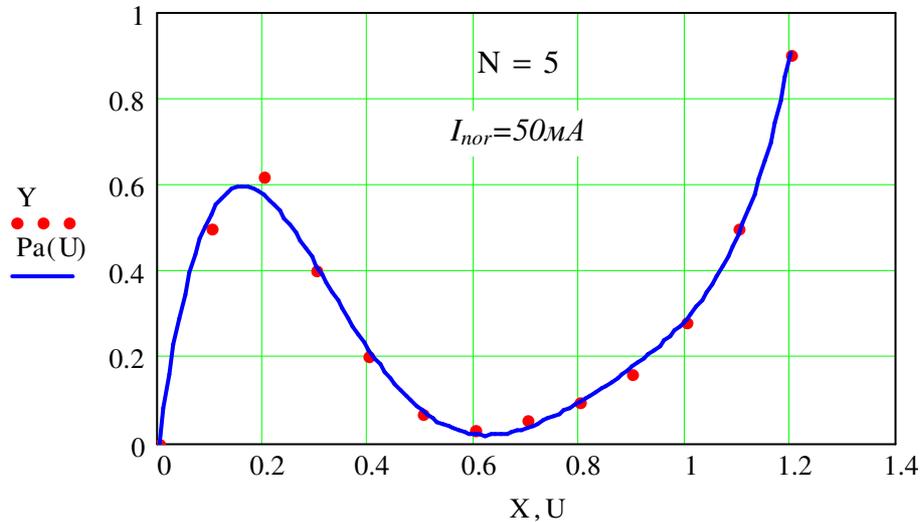


Рис. 2.5

Как видно из графиков на рис. 2.5, результаты аппроксимации анализируемой ВАХ полиномом пятой степени качественно правильно отражают исходную зависимость и вполне удовлетворительны количественно.

## 2.5. Аппроксимация ортогональными полиномами

Рассмотрим ряд полиномов степеней, изменяющихся от  $n=0$  до  $n=N$ :  $p_0(x), p_1(x), \dots, p_N(x)$ . На множестве точек  $x_0, x_1, \dots, x_M$  полиномы являются взаимно ортогональными, если

$$\sum_{m=0}^M p_n(x_m) p_k(x_m) = 0, \quad n, k = 0, 1, \dots, N; k \neq n. \quad (2.7)$$

Используем эти ортогональные полиномы в качестве базисных функций в выражении (2.3). Получим

$$Q_N(x) = a_0 p_0(x) + a_1 p_1(x) + \dots + a_N p_N(x). \quad (2.8)$$

В силу единственности полинома степени  $N$ , аппроксимирующего по методу наименьших квадратов исходный набор данных, выражение (2.8) дает еще одно представление полинома (2.5) с коэффициентами, определяемыми решением системы нормальных уравнений (2.6).

В представлении (2.8) коэффициенты  $a_n$  легко находятся из системы нормальных уравнений (2.4). Действительно, в этом случае матрица в левой части системы диагональна и мы приходим к выражениям

$$a_n = \sum_{m=0}^M p_n(x_m) y_m / \sum_{m=0}^M p_n^2(x_m), \quad n = 0, 1, \dots, N. \quad (2.9)$$

Они значительно облегчают задачу расчета коэффициентов аппроксимации, т.к. позволяют избежать необходимости решения плохо обусловленных систем уравнений. Все трудности теперь связаны с построением системы ортогональных полиномов  $p_n(x)$ .

Рассмотрим один из наиболее распространенных способов построения ортогональных полиномов. Начнем с

$$p_0(x) = 1, \quad p_1(x) = x - \alpha_0,$$

где константа  $\alpha_0$  должна быть определена из условия ортогональности (2.7) для  $k=0, n=1$ :

$$\sum_{m=0}^M (x_m - \alpha_0) = \sum_{m=0}^M x_m - (M + 1)\alpha_0 = 0.$$

Отсюда находим

$$\alpha_0 = \frac{1}{M + 1} \sum_{m=0}^M x_m.$$

Пусть теперь

$$p_2(x) = (x - \alpha_1)p_1(x) - \beta_1,$$

где константы  $\alpha_1$  и  $\beta_1$  определяются из условий ортогональности (2.7) для  $k=0, 1; n=2$ , т.е. из условий

$$\begin{aligned} \sum_{m=0}^M [(x_m - \alpha_1)p_1(x_m) - \beta_1] &= \sum_{m=0}^M x_m p_1(x_m) - (M + 1)\beta_1 = 0, \\ \sum_{m=0}^M p_1(x_m)[(x_m - \alpha_1)p_1(x_m) - \beta_1] &= \sum_{m=0}^M x_m p_1^2(x_m) - \alpha_2 \sum_{m=0}^M p_1^2(x_m) = 0. \end{aligned}$$

Отсюда нетрудно получить

$$\begin{aligned} \beta_1 &= \frac{1}{M + 1} \sum_{m=0}^M x_m p_1(x_m), \\ \alpha_1 &= \sum_{i=0}^m x_m p_1^2(x_m) / \sum_{m=0}^M p_1^2(x_m). \end{aligned}$$

Построение полиномов последующих степеней проводится аналогичным образом. Считая, что полиномы  $p_0(x), p_1(x), \dots, p_n(x)$  уже определены, положим

$$p_{n+1}(x) = (x - \alpha_n)p_n(x) - \beta_n p_{n-1}(x). \quad (2.10)$$

Применив условия ортогональности, находим формулы для расчета неизвестных констант:

$$\begin{aligned} \beta_n &= \frac{\sum_{m=0}^M x_m p_{n-1}(x_m) p_n(x_m)}{\sum_{m=0}^M p_{n-1}^2(x_m)}, \\ \alpha_n &= \frac{\sum_{m=0}^M x_m p_n^2(x_m)}{\sum_{m=0}^M p_n^2(x_m)}. \end{aligned} \quad (2.11)$$

Теперь алгоритм построения ортогональных полиномов и полинома метода наименьших квадратов формулируется следующим образом:

1. Положить  $p_0(x) = 1$ ,  $p_1(x) = x - \frac{1}{M+1} \sum_{m=0}^M x_m$ .
2. Для  $n = 1, 2, \dots, N-1$  применить рекуррентное соотношение (2.10), в котором константы определить по формулам (2.11).
3. По формулам (2.9) вычислить коэффициенты и рассчитать значения полинома (2.8).

Такой подход является привлекательным с вычислительной точки зрения, т.к. позволяет избежать решения системы нормальных уравнений, которая может оказаться плохо обусловленной. Еще одним преимуществом является возможность строить полиномы метода наименьших квадратов последовательно, степень за степенью. Например, если заранее неизвестна степень аппроксимирующего полинома, то можно начать с полинома первой степени, затем построить полином второй степени и т.д., пока не получим полином, который можно считать подходящим.

Описанный алгоритм реализует представленная ниже MathCAD-программа *OrthoPol*. Результаты аппроксимации нормированной ВАХ туннельного диода последовательностью полиномов степени  $N = 3, 5, 9$  показаны на рис. 2.6.

```

OrthoPol(x, N, X, Y) :=
  M ← length(X) - 1
  Xa ← mean(X)
  a0 ← mean(Y)
  S0 ← M + 1
  for m ∈ 0..M
    P0,m ← 1
    P1,m ← Xm - Xa
  for n ∈ 1..N
    Sn ← ∑m=0M (Pn,m)2
    an ← (Sn)-1 · ∑m=0M Ym · Pn,m
    αn ← (Sn-1)-1 · ∑m=0M Xm · (Pn-1,m)2
    βn ← (Sn-1)-1 · ∑m=0M Xm · Pn-1,m · Pn,m
    for m ∈ 0..M
      Pn+1,m ← (Xm - αn) · Pn,m - βn · Pn-1,m
  p0 ← 1
  p1 ← x - Xa
  Q1 ← a0 + a1 · p1
  for n ∈ 2..N
    pn ← (x - αn-1) · pn-1 - βn-1 · pn-2
    Qn ← Qn-1 + an · pn
  QN

```

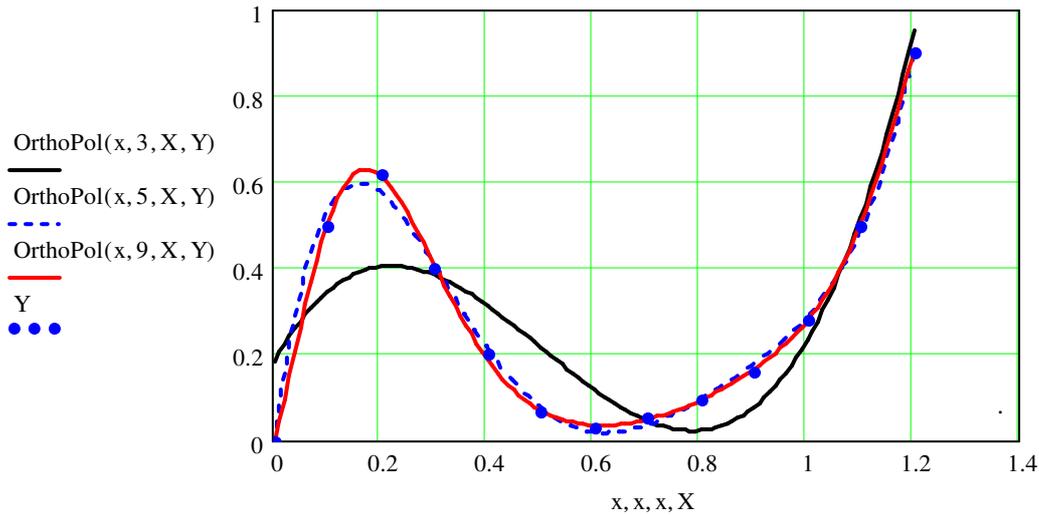


Рис. 2.6

## 2.6. Метод наименьших квадратов и линейная регрессия

На практике нередко встречаются ситуации, когда экспериментальные данные связаны линейной зависимостью, на которую накладываются шумы и ошибки измерений. Возможно также, что линейная зависимость между данными получается после преобразования переменных.

Естественно, что для аппроксимации такого рода зависимостей целесообразно применять полиномы первой степени. Для их построения воспользуемся подходом, основанным на использовании ортогональных полиномов. В этом случае в соответствии с выражением (2.8) аппроксимирующий полином запишем в виде

$$Q_1(x) = a_0 p_0(x) + a_1 p_1(x),$$

где  $p_0(x) = 1$ ,  $p_1(x) = x - \langle x \rangle$  и использовано обозначение

$$\langle x \rangle = \alpha_0 = \frac{1}{M+1} \sum_{m=0}^M x_m. \quad (2.12)$$

Вычисляя по формулам (2.9) коэффициенты  $a_0$  и  $a_1$ , получим

$$a_0 = \langle y \rangle, \quad a_1 = \frac{\langle xy \rangle - \langle x \rangle \langle y \rangle}{\langle x^2 \rangle - \langle x \rangle^2},$$

где

$$\langle y \rangle = \frac{1}{M+1} \sum_{m=0}^M y_m, \quad \langle xy \rangle = \frac{1}{M+1} \sum_{m=0}^M x_m y_m, \quad \langle x^2 \rangle = \frac{1}{M+1} \sum_{m=0}^M x_m^2. \quad (2.13)$$

В конечном итоге мы приходим к следующему выражению

$$y = Q_1(x) = \langle y \rangle + \frac{\langle xy \rangle - \langle x \rangle \langle y \rangle}{\langle x^2 \rangle - \langle x \rangle^2} (x - \langle x \rangle), \quad (2.14)$$

в котором средние значения соответствующих величин вычисляются по формулам (2.12) и (2.13).

Линия, задаваемая уравнением (2.14), называется прямой линейной регрессии  $y$  на  $x$ . Она широко используется на практике для линейного прогноза результатов эксперимента и оценок параметров линейных систем. В ряде случаев исходные нелинейные зависимости между величинами путем замен переменных можно преобразовать к линейным, а затем применять метод линейной регрессии.

Часто линия регрессии записывается как

$$y = ax + b, \quad (2.15)$$

причем коэффициенты  $a$  и  $b$  являются искомыми параметрами, оцениваемыми по результатам эксперимента. Преобразовав выражение (2.14) к форме (2.15), получим

$$a = \frac{\langle xy \rangle - \langle x \rangle \langle y \rangle}{\langle x^2 \rangle - \langle x \rangle^2}, \quad b = \frac{\langle y \rangle \langle x^2 \rangle - \langle x \rangle \langle xy \rangle}{\langle x^2 \rangle - \langle x \rangle^2}. \quad (2.16)$$

Конечно, для того, чтобы вычислить коэффициенты  $a$  и  $b$ , не обязательно использовать ортогональные полиномы. Это можно сделать также, решив систему нормальных уравнений (2.6), которая в данном случае принимает вид

$$\begin{aligned} a + b \langle x \rangle &= \langle y \rangle, \\ a \langle x \rangle + b \langle x^2 \rangle &= \langle xy \rangle. \end{aligned}$$

Для ее решения воспользуемся пакетом MathCAD в режиме символических вычислений. Соответствующая программа выглядит следующим образом.

$$\text{Given } a_0 + a_1 \cdot Xa = Ya \quad a_0 \cdot Xa + a_1 \cdot XXa = XYa$$

$$\text{Find}(a_0, a_1) \rightarrow \left[ \begin{array}{c} \frac{(Xa \cdot XYa - Ya \cdot XXa)}{(Xa^2 - XXa)} \\ \frac{(Xa \cdot Ya - XYa)}{(Xa^2 - XXa)} \end{array} \right]$$

С учетом обозначений результат действия оператора *Find* полностью совпадает с выражениями (2.16). Коэффициенты линии регрессии (2.15) при значении параметра  $Par = 0$  вычисляет следующая MathCAD-программа. Если  $Par \neq 0$ , то программа возвращает значение  $y$  в текущей точке  $x$ .

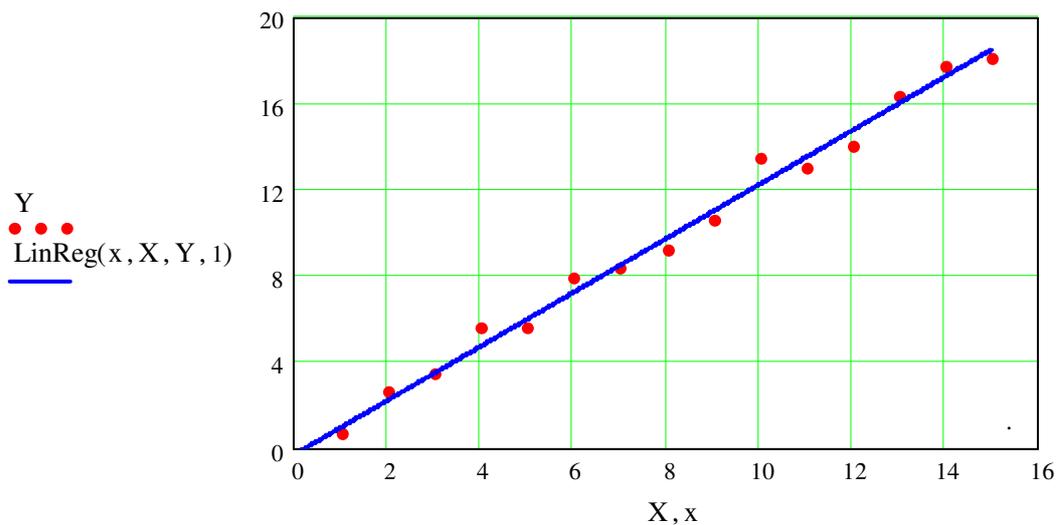
```

LinReg(x, X, Y, Par) :=
  M ← length(X) - 1
  Xa ← mean(X)
  Ya ← mean(Y)
  XYa ←  $\frac{(X^T \cdot Y)_0}{M + 1}$ 
  XXa ←  $\frac{(X^T \cdot X)_0}{M + 1}$ 
  a ←  $\frac{XYa - Xa \cdot Ya}{XXa - Xa^2}$ 
  b ←  $\frac{Ya \cdot XXa - Xa \cdot XYa}{XXa - Xa^2}$ 
  y ← a · x + b
  return stack(a, b) if Par = 0

```

На рис. 2.7 в качестве примера приведен график линии регрессии для набора данных, отображенных на рисунке точками.

$ab := \text{LinReg}(1, X, Y, 0) \quad ab^T = (1.254 \quad -0.29)$



Помимо аппроксимации в линейных физических моделях метод линейной регрессии удастся применять также тогда, когда двухпараметрическая нелинейная модель заменой переменных сводится к линейной. Продемонстрируем это на конкретном примере.

Пусть проводится обработка данных эксперимента по изучению процесса разрядки емкости через сопротивление. Схема эксперимента приведена на рис. 2.8.

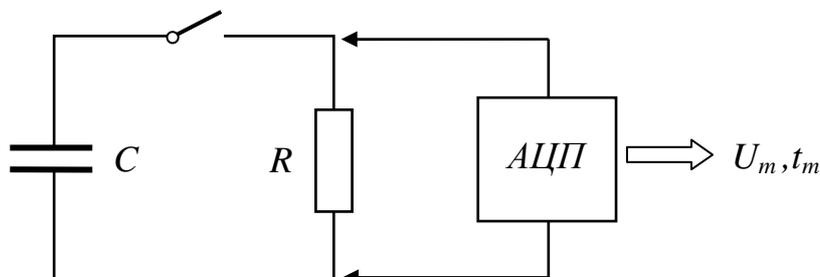


Рис. 2.8

С помощью аналого-цифрового преобразователя (АЦП) удастся записать в память компьютера последовательность отсчетов напряжения  $U_m$  в дискретные моменты времени  $t_m$ . «Зашумленные» экспериментальные данные (малоразрядный АЦП) представлены на рис. 2.9, *а* точками.

Нетрудно установить теоретический закон изменения со временем напряжения на емкости:

$$U(t) = U(0) \exp\left(-\frac{t}{\tau}\right). \quad (2.17)$$

Данную теоретическую зависимость целесообразно принять в качестве аппроксимирующей функции для таблицы экспериментальных данных. Подлежат определению параметры  $U(0)$  и  $\tau$  – начальное напряжение и постоянная времени  $RC$ -цепи. Это стандартная задача нелинейного МНК.

Вычислив экспоненту от обеих частей выражения (2.17), получим

$$\ln U = \ln U(0) - \frac{t}{\tau}.$$

Теперь, введя обозначения  $\ln U = y$ ,  $t = x$ ,  $\ln U(0) = a$  и  $-\tau^{-1} = b$ , придем к стандартной записи линейной регрессии (2.15). Экспериментальные данные для переменных  $x$  и  $y$  точками показаны на рис. 2.9, *б*.

Построим линейную регрессию для имеющегося набора данных по MathCAD-программе.

$$\begin{pmatrix} a \\ b \end{pmatrix} := \text{LinReg}(1, x, y, 0) \quad a = -0.1 \quad b = 4.6$$

$$\tau := -\frac{1}{a} \quad \tau = 10.17 \quad U_0 := \exp(b) \quad U_0 = 99.53$$

Непрерывными линиями на рис. 2.9 изображены зависимость (2.17) с рассчитанными выше параметрами и линия регрессии.

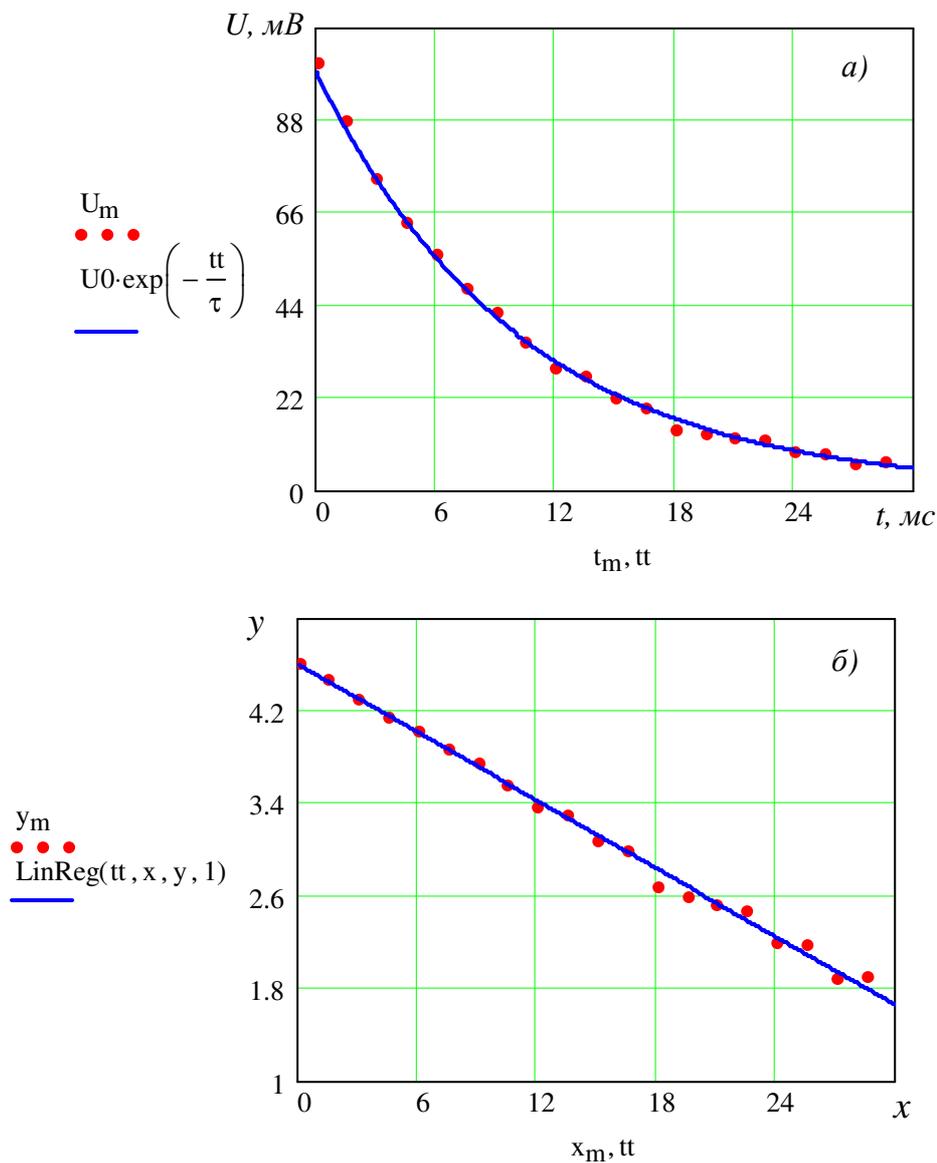


Рис. 2.9

Таким образом, в приведенном примере удалось свести задачу нелинейного метода наименьших квадратов к более простой задаче линейной регрессии.

## ГЛАВА 3. ЧИСЛЕННОЕ ДИФФЕРЕНЦИРОВАНИЕ И АППРОКСИМАЦИЯ ПРОИЗВОДНЫХ

### 3.1. Введение

В вычислительной практике формулы численного дифференцирования наиболее часто используются:

- а) для определения производных функций, заданных таблично;
- б) для аппроксимации производных при решении дифференциальных уравнений конечно-разностными методами.

Если формулы численного дифференцирования применяются для функций, заданных таблично по экспериментальным данным, то следует иметь в виду, что регистрируемая в эксперименте полезная информация сопровождается более или менее интенсивным шумом. При наличии интенсивной шумовой компоненты в исходных данных результаты численного дифференцирования могут оказаться совершенно бессмысленными. В этом случае последовательность данных следует подвергнуть предварительной фильтрации (сглаживанию), подавляющей шумы эксперимента.

Широко распространенный способ получения формул численного дифференцирования – это метод неопределенных коэффициентов. Источниками формул могут служить также различные формы интерполяционных полиномов и разложения функций в ряды Тейлора.

### 3.2. Метод неопределенных коэффициентов

Производную  $y^{(k)} = f^{(k)}(x)$  функции  $y = f(x)$  представим в виде линейной комбинации ее значений в узлах сетки аргумента  $x$ :  $x_0, x_1, \dots, x_N$ :

$$f^{(k)}(\xi) \approx \sum_{n=0}^N C_n f(x_n). \quad (3.1)$$

Здесь точка  $\xi$  принадлежит отрезку  $[x_0, x_N]$ . При этом коэффициенты  $C_n$  формулы дифференцирования (3.1) определяются, исходя из некоторых дополнительных условий. Пусть таким условием является требование о том, чтобы формула была точна для полинома максимально высокой степени  $M$ . То есть, для полинома

$$P_M(x) = \sum_{m=0}^M a_m x^m \quad (3.2)$$

приближенное равенство (3.1) должно обратиться в точное.

Подставив (3.2) в (3.1), получим

$$\sum_{m=0}^M a_m (\xi^m)^{(k)} = \sum_{n=0}^N C_n \sum_{m=0}^M a_m x_n^m .$$

Для выполнения этого равенства при любом полиноме степени  $M$  необходимо и достаточно, чтобы множители при всех коэффициентах  $a_m$  в правой и левой частях были равны. Это дает нам систему из  $M+1$  линейного уравнения:

$$\sum_{n=0}^N C_n x_n^m = (\xi^m)^{(k)}, \quad m = 0, 1, \dots, M. \quad (3.3)$$

Если  $M = N$ , то в системе (3.3) число уравнений равно числу неизвестных. В таком случае определитель системы, определитель Вандермонда, отличен от нуля. Поэтому система (3.3) при  $M = N$  всегда имеет единственное решение. Значит всегда можно построить формулу численного дифференцирования вида (3.1) с  $N + 1$  узлом, точную для полинома степени  $N$ .

### 3.3. Формулы численного дифференцирования

Определимся с терминологией и построим ряд наиболее известных формул. Если все узлы формулы (3.1), кроме одного, расположены правее точки  $\xi$ , что автоматически выполняется при  $\xi = x_0$ , то формула аппроксимирует производную правыми разностями. Альтернативой является формула аппроксимации левыми разностями ( $\xi = x_N$ ). При симметричном относительно точки  $\xi$  расположении узлов имеет место аппроксимация центральными разностями. Можно показать, что такая аппроксимация точна для полинома степени  $N + 1$ . Кроме того, в зависимости от числа узлов, входящих в (3.1), различают двухточечные, трехточечные и т.д. формулы дифференцирования.

При расчете коэффициентов  $C_n$  в формуле (3.1) для первой и второй производных полезна MathCAD-программа *DifMUC*, решающая СЛАУ (3.3) в режиме аналитических преобразований

**Program DifMUC**     $x := (0 \ h \ 2 \cdot h \ 3 \cdot h \ 4 \cdot h \ 5 \cdot h \ 6 \cdot h)^T$

$$\begin{array}{l} \mathbb{W}(x, N) := \left| \begin{array}{l} \text{for } m \in 0..N-1 \\ \quad W_{0,m} \leftarrow 1 \\ \quad \text{for } n \in 1..N-1 \\ \quad \quad W_{n,m} \leftarrow (x_n)^n \\ \quad W \end{array} \right. \end{array} \quad \begin{array}{l} b(\xi, N) := \left| \begin{array}{l} b_0 \leftarrow 0 \\ \text{for } n \in 1..N-1 \\ \quad b_n \leftarrow n \cdot \xi^{n-1} \\ b \end{array} \right. \end{array} \quad \begin{array}{l} b2(\xi, N) := \left| \begin{array}{l} b_0 \leftarrow 0 \\ b_1 \leftarrow 0 \\ b_2 \leftarrow 2 \\ \text{for } n \in 3..N-1 \quad \text{if } N \geq 4 \\ \quad b_n \leftarrow n \cdot (n-1) \cdot \xi^{n-2} \\ b \end{array} \right. \end{array}$$

$$\text{lsolve}(W(x, 2), b(0, 2))^T \rightarrow \left( -\frac{1}{h} \quad \frac{1}{h} \right) \quad \text{lsolve}(W(x, 3), b2(h, 3))^T \rightarrow \left( \frac{1}{h^2} \quad -\frac{2}{h^2} \quad \frac{1}{h^2} \right)$$

Результаты расчета коэффициентов двух-, трех-, четырех и пятиточечной аппроксимаций первой производной правыми разностями выглядят следующим образом.

$$C^{(0)} := \text{lsolve}(W(x,2), b(0,2)) \quad C^{(1)} := \text{lsolve}(W(x,3), b(0,3))$$

$$C^{(2)} := \text{lsolve}(W(x,4), b(0,4)) \quad C^{(3)} := \text{lsolve}(W(x,5), b(0,5))$$

$$C^T \rightarrow \begin{pmatrix} -\frac{1}{h} & \frac{1}{h} & 0 & 0 & 0 \\ -\frac{3}{2 \cdot h} & \frac{2}{h} & -\frac{1}{2 \cdot h} & 0 & 0 \\ -\frac{11}{6 \cdot h} & \frac{3}{h} & -\frac{3}{2 \cdot h} & \frac{1}{3 \cdot h} & 0 \\ -\frac{25}{12 \cdot h} & \frac{4}{h} & -\frac{3}{h} & \frac{4}{3 \cdot h} & -\frac{1}{4 \cdot h} \end{pmatrix}$$

Аналогичные результаты для трех-, четырех- и пятиточечной аппроксимаций второй производной имеют вид.

$$C^{(0)} := \text{lsolve}(W(x,3), b2(0,3)) \quad C^{(1)} := \text{lsolve}(W(x,4), b2(0,4))$$

$$C^{(2)} := \text{lsolve}(W(x,5), b2(0,5))$$

$$C^T \rightarrow \begin{pmatrix} \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} & 0 & 0 \\ \frac{2}{h^2} & -\frac{5}{h^2} & \frac{4}{h^2} & -\frac{1}{h^2} & 0 \\ \frac{35}{12 \cdot h^2} & -\frac{26}{3 \cdot h^2} & \frac{19}{2 \cdot h^2} & -\frac{14}{3 \cdot h^2} & \frac{11}{12 \cdot h^2} \end{pmatrix}$$

Формулы численного дифференцирования с помощью правых разностей сведены в таблицы 3.1 и 3.2.

Изменим в выражении (3.1) нумерацию узлов так, что  $x_N \rightarrow x_0$ ,  $x_{N-1} \rightarrow x_{-1}$ , ...,  $x_0 \rightarrow x_{-N}$ . Нетрудно показать, что, если при этом в формулах дифференцирования с помощью правых разностей изменить знаки на противоположные так, как это сделано в таблицах 3.1 и 3.2, то мы получим формулы дифференцирования с помощью левых разностей (нижние знаки в арифметических операциях и индексах).

Таблица 3.1

Аппроксимация	$y^{(1)}(x_0)$	Погрешность
Двухточечная	$y_0^{(1)} = \pm \left( \frac{y_{\pm 1} - y_0}{h} \right)$	$\frac{h}{2} y_0^{(2)}$
Трехточечная	$y_0^{(1)} = \mp \left( \frac{3y_0 - 4y_{\pm 1} + y_{\pm 2}}{2h} \right)$	$\frac{h^2}{3} y_0^{(3)}$
Четырехточечная	$y_0^{(1)} = \mp \left( \frac{11y_0 - 18y_{\pm 1} + 9y_{\pm 2} - 2y_{\pm 3}}{6h} \right)$	$\frac{h^3}{4} y_0^{(4)}$
Пятиточечная	$y_0^{(1)} = \mp \left( \frac{25y_0 - 48y_{\pm 1} + 36y_{\pm 2} - 16y_{\pm 3} + 3y_{\pm 4}}{12h} \right)$	$\frac{h^4}{5} y_0^{(5)}$

Таблица 3.2

Аппроксимация	$y^{(2)}(x_0)$	Погрешность
Трехточечная	$y_0^{(1)} = \frac{y_0 - 2y_{\pm 1} + y_{\pm 2}}{h^2}$	$hy_0^{(3)}$
Четырехточечная	$y_0^{(1)} = \frac{2y_0 - 5y_{\pm 1} + 4y_{\pm 2} - y_{\pm 3}}{h^2}$	$\frac{11h^2}{12} y_0^{(4)}$
Пятиточечная	$y_0^{(1)} = \frac{35y_0 - 104y_{\pm 1} + 114y_{\pm 2} - 56y_{\pm 3} + 11y_{\pm 4}}{12h^2}$	$\frac{5h^3}{6} y_0^{(5)}$

При вычислениях по программе *DifMUC* коэффициентов аппроксимаций производных центральными разностями получим результаты:

– для первой производной

$$C^{(0)} := \text{lsolve}(W(x, 3), b(h, 3)) \quad C^{(1)} := \text{lsolve}(W(x, 5), b(2 \cdot h, 5))$$

$$C^{(2)} := \text{lsolve}(W(x, 7), b(3 \cdot h, 7))$$

$$C^T \rightarrow \begin{pmatrix} -\frac{1}{2 \cdot h} & 0 & \frac{1}{2 \cdot h} & 0 & 0 & 0 & 0 \\ \frac{1}{12 \cdot h} & -\frac{2}{3 \cdot h} & 0 & \frac{2}{3 \cdot h} & -\frac{1}{12 \cdot h} & 0 & 0 \\ -\frac{1}{60 \cdot h} & \frac{3}{20 \cdot h} & -\frac{3}{4 \cdot h} & 0 & \frac{3}{4 \cdot h} & -\frac{3}{20 \cdot h} & \frac{1}{60 \cdot h} \end{pmatrix}$$

– для второй производной

$$C^{(0)} := \text{lsolve}(W(x, 3), b2(h, 3)) \quad C^{(1)} := \text{lsolve}(W(x, 5), b2(2 \cdot h, 5))$$

$$C^{(2)} := \text{lsolve}(W(x, 7), b2(3 \cdot h, 7))$$

$$C^T \rightarrow \begin{pmatrix} \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} & 0 & 0 & 0 & 0 \\ -\frac{1}{12 \cdot h^2} & \frac{4}{3 \cdot h^2} & -\frac{5}{2 \cdot h^2} & \frac{4}{3 \cdot h^2} & -\frac{1}{12 \cdot h^2} & 0 & 0 \\ \frac{1}{90 \cdot h^2} & -\frac{3}{20 \cdot h^2} & \frac{3}{2 \cdot h^2} & -\frac{49}{18 \cdot h^2} & \frac{3}{2 \cdot h^2} & -\frac{3}{20 \cdot h^2} & \frac{1}{90 \cdot h^2} \end{pmatrix}$$

Формулы численного дифференцирования с помощью центральных разностей сведены в таблицы 3.3 и 3.4.

Таблица 3.3

Аппроксимация	$y^{(1)}(x_0)$	Погрешность
Трехточечная	$y_0^{(1)} = \frac{y_1 - y_{-1}}{2h}$	$\frac{h^2}{6} y_0^{(2)}$
Пятиточечная	$y_0^{(1)} = \frac{-y_2 + 8y_1 - 8y_{-1} + y_{-2}}{12h}$	$\frac{h^4}{30} y_0^{(5)}$
Семиточечная	$y_0^{(1)} = \frac{y_3 - 9y_2 + 45y_1 - 45y_{-1} + 9y_{-2} - y_{-3}}{60h}$	$\frac{h^6}{140} y_0^{(7)}$

Таблица 3.4

Аппроксимация	$y^{(2)}(x_0)$	Погрешность
Трехточечная	$y_0^{(2)} = \frac{y_1 - 2y_0 + y_{-1}}{h^2}$	$\frac{h^2}{12} y_0^{(4)}$
Пятиточечная	$y_0^{(2)} = \frac{-y_2 + 16y_1 - 30y_0 + 16y_{-1} - y_{-2}}{12h^2}$	$\frac{h^4}{90} y_0^{(6)}$
Семиточечная	$y_0^{(2)} = \frac{2y_3 - 27y_2 + 270y_1 - 490y_0}{180h^2} + \frac{270y_{-1} - 27y_{-2} + 2y_{-3}}{180h^2}$	$\frac{h^6}{560} y_0^{(8)}$

### 3.4. Погрешность формул

Для оценки погрешностей формул численного дифференцирования естественным представляется использование разложений функции  $y = f(x)$  в ряды Тейлора. Например, для исследования двухточечной аппроксимации

$$y_0^{(1)} = \pm \left( \frac{y_{\pm 1} - y_0}{h} \right) \quad (3.4)$$

воспользуемся разложениями

$$y_{\pm 1} = f(x_0 \pm h) = f(x_0) \pm hf'(x_0) + \frac{h^2}{2} f''(x_0) \pm O(h^3).$$

Подставив их в правую часть формулы (3.4), получим

$$y_0^{(1)} = f'(x_0) \pm \frac{h}{2} f''(x_0) + O(h^2).$$

Отсюда следует, что формула (3.4) аппроксимирует первую производную  $f'(x_0)$  с погрешностью первого порядка малости по шагу  $h$ :

$$e_a(h) = Ch + O(h^2). \quad (3.5)$$

Здесь первое слагаемое носит название главного члена погрешности. Именно оно занесено в третий столбец таблицы 3.1.

Погрешность любой формулы из таблиц 3.1 – 3.4 можно представить в общем виде как

$$e_a(h) = Ch^p + O(h^k). \quad (3.6)$$

Здесь параметр  $p$  называется порядком точности формулы. В справедливости записи (3.6) нетрудно убедиться, воспользовавшись возможностями аналитических преобразований пакета MathCAD. В частности, для правых разностей из таблицы 3.1 имеют место следующие разложения:

$$\frac{-3 \cdot f(0) + 4 \cdot f(h) - f(2 \cdot h)}{2 \cdot h} \text{ series, h, 5} \rightarrow \left| \begin{array}{l} t \leftarrow 0 \\ \frac{d}{dt} f(t) \end{array} \right| - \frac{1}{3} \cdot \left| \begin{array}{l} t \leftarrow 0 \\ \frac{d^3}{dt^3} f(t) \end{array} \right| \cdot h^2 - \frac{1}{4} \cdot \left| \begin{array}{l} t \leftarrow 0 \\ \frac{d^4}{dt^4} f(t) \end{array} \right| \cdot h^3,$$

$$D(h) := \frac{-11 \cdot f(0) + 18 \cdot f(h) - 9 \cdot f(2 \cdot h) + 2 \cdot f(3 \cdot h)}{6 \cdot h}$$

$$D(h) \text{ series, h, 6} \rightarrow \left| \begin{array}{l} t \leftarrow 0 \\ \frac{d}{dt} f(t) \end{array} \right| + \frac{1}{4} \cdot \left| \begin{array}{l} t \leftarrow 0 \\ \frac{d^4}{dt^4} f(t) \end{array} \right| \cdot h^3 + \frac{3}{10} \cdot \left| \begin{array}{l} t \leftarrow 0 \\ \frac{d^5}{dt^5} f(t) \end{array} \right| \cdot h^4,$$

$$D(h) := \frac{-25 \cdot f(0) + 48 \cdot f(h) - 36 \cdot f(2 \cdot h) + 16 \cdot f(3 \cdot h) - 3 \cdot f(4 \cdot h)}{12 \cdot h}$$

$$D(h) \text{ series , h , 7 } \rightarrow \left| \begin{array}{l} t \leftarrow 0 \\ \frac{d}{dt} f(t) \end{array} \right. - \frac{1}{5} \cdot \left| \begin{array}{l} t \leftarrow 0 \\ \frac{d^5}{dt^5} f(t) \end{array} \right. \cdot h^4 - \frac{1}{3} \cdot \left| \begin{array}{l} t \leftarrow 0 \\ \frac{d^6}{dt^6} f(t) \end{array} \right. \cdot h^5$$

Главные члены этих разложений занесены в третий столбец таблицы 3.1.

Центральные разности из таблицы 3.3 имеют разложения вида

$$\frac{f(h) - f(-h)}{2 \cdot h} \text{ series , h , 6 } \rightarrow \left| \begin{array}{l} t \leftarrow 0 \\ \frac{d}{dt} f(t) \end{array} \right. + \frac{1}{6} \cdot \left| \begin{array}{l} t \leftarrow 0 \\ \frac{d^3}{dt^3} f(t) \end{array} \right. \cdot h^2 + \frac{1}{120} \cdot \left| \begin{array}{l} t \leftarrow 0 \\ \frac{d^5}{dt^5} f(t) \end{array} \right. \cdot h^4$$

$$D(h) := \frac{-f(2 \cdot h) + 8 \cdot f(h) - 8 \cdot f(-h) + f(-2 \cdot h)}{12 \cdot h}$$

$$D(h) \text{ series , h , 8 } \rightarrow \left| \begin{array}{l} t \leftarrow 0 \\ \frac{d}{dt} f(t) \end{array} \right. - \frac{1}{30} \cdot \left| \begin{array}{l} t \leftarrow 0 \\ \frac{d^5}{dt^5} f(t) \end{array} \right. \cdot h^4 - \frac{1}{252} \cdot \left| \begin{array}{l} t \leftarrow 0 \\ \frac{d^7}{dt^7} f(t) \end{array} \right. \cdot h^6$$

$$D(h) := \frac{f(3 \cdot h) - 9 \cdot f(2 \cdot h) + 45 \cdot f(h) - 45 \cdot f(-h) + 9 \cdot f(-2 \cdot h) - f(-3 \cdot h)}{60 \cdot h}$$

$$D(h) \text{ series , h , 10 } \rightarrow \left| \begin{array}{l} t \leftarrow 0 \\ \frac{d}{dt} f(t) \end{array} \right. + \frac{1}{140} \cdot \left| \begin{array}{l} t \leftarrow 0 \\ \frac{d^7}{dt^7} f(t) \end{array} \right. \cdot h^6 + \frac{1}{720} \cdot \left| \begin{array}{l} t \leftarrow 0 \\ \frac{d^9}{dt^9} f(t) \end{array} \right. \cdot h^8$$

Их главные члены внесены в третий столбец таблицы 3.3.

Аналогичные вычисления позволили заполнить ячейки погрешностей в таблицах 3.2 и 3.4.

### 3.5. Метод Рунге – Ромберга – Ричардсона

Приведенные выше формулы численного дифференцирования позволяют получить тем более точный результат, чем большее число узлов в них используется. Однако формулы становятся довольно громоздкими и их практическое применение не всегда оправдано. Тем более, что существует способ повышения точности без значительного усложнения вычислений. Этот способ известен под общим названием метода Рунге – Ромберга – Ричардсона (используются также названия метод Рунге – Ромберга, экстраполяция по Ричардсону) и состоит в следующем.

Пусть для некоторой величины  $D(x)$  имеется приближенная оценка  $\hat{D}(x, h)$ , вычисляемая на равномерной сетке узлов  $x_n$  с шагом  $h$ . Тогда погрешность оценки записывается как

$$D(x) - \hat{D}(x, h) = Ch^p + O(h^k). \quad (3.7)$$

Если провести вычисления на сетке с шагом  $\alpha h$ , то погрешность примет вид

$$D(x) - \hat{D}(x, \alpha h) = C\alpha^p h^p + O(\alpha^k h^k). \quad (3.8)$$

Составив разность погрешностей (3.8) и (3.7), найдем выражение для главного члена погрешности (первую формулу Рунге):

$$Ch^p = \frac{\hat{D}(x, h) - \hat{D}(x, \alpha h)}{\alpha^p - 1} + O(h^k). \quad (3.9)$$

Формула позволяет оценить погрешность расчета на сетке с шагом  $h$ , используя результат расчета на сетке с шагом  $\alpha h$ .

Теперь можно подставить (3.9) в (3.7) и найти уточненную оценку величины  $D(x)$ :

$$D(x) = \frac{\alpha^p \hat{D}(x, h) - \hat{D}(x, \alpha h)}{\alpha^p - 1} + O(h^k). \quad (3.10)$$

Эта оценка (вторая формула Рунге) имеет порядок точности  $k > p$ .

Для примера применим метод к уточнению аппроксимации первой производной центральными разностями. В качестве исходной аппроксимации возьмем

$$\hat{D}(x, h) = y_0^{(1)} = \frac{y_1 - y_{-1}}{2h}.$$

Она имеет второй порядок точности:  $p = 2$ ,  $k = 2$ . Аппроксимация при удвоенном шаге ( $\alpha = 2$ ) вычисляется как

$$\hat{D}(x, 2h) = \frac{y_2 - y_{-2}}{4h}.$$

По формуле (3.10) вычислим

$$y_0^{(1)} = \frac{1}{3} \left( 4 \frac{y_1 - y_{-1}}{2h} - \frac{y_2 - y_{-2}}{4h} \right) + O(h^4) = \frac{-y_2 + 8y_1 - 8y_{-1} + y_{-2}}{12h} + O(h^4),$$

что соответствует формуле дифференцирования четвертого порядка точности (см. вторую строку в таблице 3.3).

Метод Рунге – Ромберга – Ричардсона находит применение при численном интегрировании, решении дифференциальных уравнений.

### 3.6. Некорректность численного дифференцирования

В численном анализе встречаются задачи, для которых характерно то, что малые изменения входных данных приводят к немалым изменениям решения. Такие задачи объединяются под общим названием некорректных. Конечно, можно дать строгое математическое определение некорректности, но оно не меняет сути дела. Задача численного дифференцирования относится именно к этому классу задач.

Типичной ситуацией, возникающей при численном дифференцировании, является то, что табличные значения  $y_n$  дифференцируемой функции заданы с некоторыми погрешностями. Это погрешности измерений, если таблица формируется по результатам эксперимента, или погрешности вычислений при формировании таблицы по результатам предварительных расчетов. Оценим влияние погрешностей исходных данных на результат дифференцирования.

Операцию численного дифференцирования будем рассматривать как результат  $y[k]$  преобразования входного сигнала  $x[k]$  дифференцирующей системой, функционирующей в дискретном времени. Система, соответствующая двухточечной аппроксимации производной левой разностью, определяется уравнением вида

$$y[k] = \frac{1}{h}(x[k] - x[k-1]). \quad (3.11)$$

Погрешности исходных данных представим как дискретный белый шум  $\xi[k]$  с дисперсией  $\sigma_\xi^2 = \langle \xi^2[k] \rangle$ . При этом на выходе системы будет действовать сигнал

$$Y[k] = \frac{1}{h}(x[k] - x[k-1]) + \frac{1}{h}(\xi[k] - \xi[k-1]). \quad (3.12)$$

Разность выражений (3.12) и (3.11) определяет сигнал погрешности  $e[k] = Y[k] - y[k]$ :

$$e[k] = \frac{1}{h}(\xi[k] - \xi[k-1]). \quad (3.13)$$

Статистическое усреднение в уравнении (3.13) с учетом нулевой корреляции  $\langle \xi[k]\xi[k-1] \rangle = 0$  дает дисперсию и среднеквадратическое отклонение погрешности дифференцирования

$$\sigma_e^2 = 2 \frac{\sigma_\xi^2}{h^2} \text{ и } e_{sq} = \sigma_e = \sqrt{2} \frac{\sigma_\xi}{h}. \quad (3.14)$$

Если на интервале анализа  $[a, b]$  увеличивать число узловых точек  $N$ , то, как следует из полученного выражения для дисперсии погрешности,

$$\sigma_e^2 = N^2 \frac{2\sigma_\xi^2}{(b-a)^2},$$

т. е. с ростом объема выборки дисперсия не стремится к нулю. Такая оценка считается несостоятельной в смысле математической статистики. Несостоятельность оценок производных аналогичным образом можно продемонстрировать и для всех остальных формул из таблиц 3.1 – 3.4.

Рассмотрим теперь соотношение между погрешностью алгоритма (регулярной погрешностью) и погрешностью данных (статистической погрешностью) в результате дифференцирования. Сумма главного члена регулярной погрешности (3.5) и статистической погрешности (3.14) образует полную погрешность:

$$E(h) = Ch + \sqrt{2} \frac{\sigma_\xi}{h}. \quad (3.15)$$

Схематически зависимость  $E(h)$  изображена на рис. 3.1 непрерывной линией, пунктирными показаны слагаемые в (3.15). При уменьшении шага  $h$  уменьшается первое слагаемое (регулярная погрешность), но растет второе (статистическая погрешность). Минимальное значение погрешности

$$E_{\min} = 2\sqrt{\sqrt{2}\sigma_\xi C} \quad (3.16)$$

достигается при шаге

$$h_0 = \sqrt{\sqrt{2} \frac{\sigma_\xi}{C}}. \quad (3.17)$$

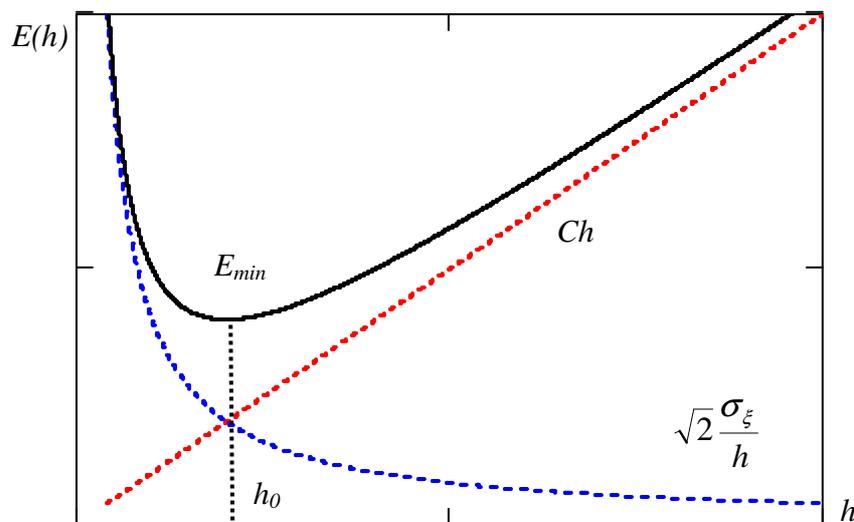


Рис. 3.1

Однако заметим, что не следует рассчитывать на достижение минимальной погрешности дифференцирования путем подбора оптимального шага. Так как значения  $E_{\min}$  и  $h_0$  существенным образом зависят от дифференцируемых функций и статистики случайных погрешностей данных, их подбор – весьма трудоемкая задача. Выражение (3.16) важно как теоретическая оценка.

В частности, можно показать, что при повышении порядка точности формулы дифференцирования величина оптимального шага увеличивается, а минимальное значение погрешности уменьшается. Например, для трехточечных формул аппроксимации первой производной левыми (правыми) разностями

$$E_{\min} = 9 \left( \frac{13}{8} \sigma_{\xi}^2 C \right)^{1/3}, \quad h_0 = \left( \sqrt{\frac{13}{8}} \frac{\sigma_{\xi}}{C} \right)^{1/3}.$$

При малых  $\sigma_{\xi}$  первая из этих величин меньше, чем величина (3.16), а вторая – больше, чем величина (3.17). Таким образом, и при наличии погрешностей данных повышение порядка точности формулы улучшает результат дифференцирования.

Но более перспективный подход к дифференцированию «зашумленных» данных состоит в их предварительном сглаживании. Сглаживать можно, например, путем аппроксимации по методу наименьших квадратов. В этом случае результат получается дифференцированием аппроксимирующей функции.

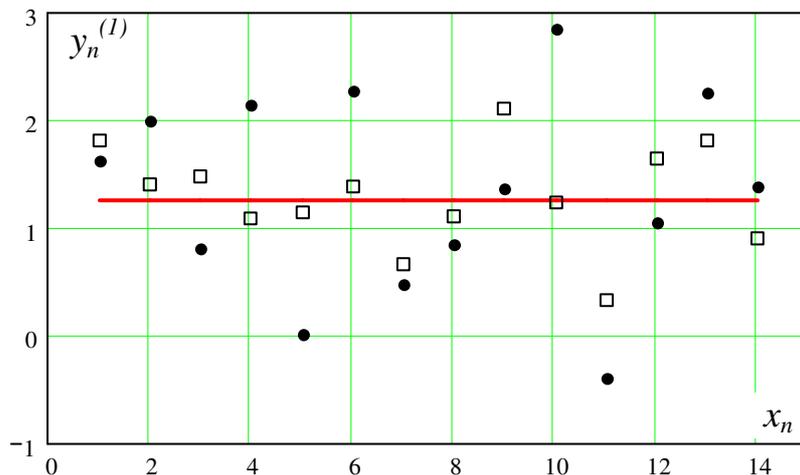


Рис. 3.2

На рис. 3.2 для примера приведены результаты дифференцирования табличной функции, график которой показан на рис. 2.7. Точки – это вычисления по двухточечной формуле с левыми разностями, квадраты – по трехточечной формуле с центральными разностями. Как и следовало ожи-

дать, результаты вычислений представляют собой последовательности случайных чисел. Между тем регулярная («незашумленная») часть зависимости  $y = f(x)$  – линейная функция, аппроксимируемая линией регрессии  $y = ax + b$ . Поэтому следует ожидать, что значение  $y_*^{(1)} = a = 1.25$  (сплошная линия на рис. 3.2) дает оценку производной, близкую к ее истинному значению. Заметим, что численное дифференцирование с максимальным на анализируемом отрезке шагом  $h = 15$  дает значение  $y^{(1)} = (y_{15} - y_0)/h = 1.36$ , близкое к  $y_*^{(1)}$ .

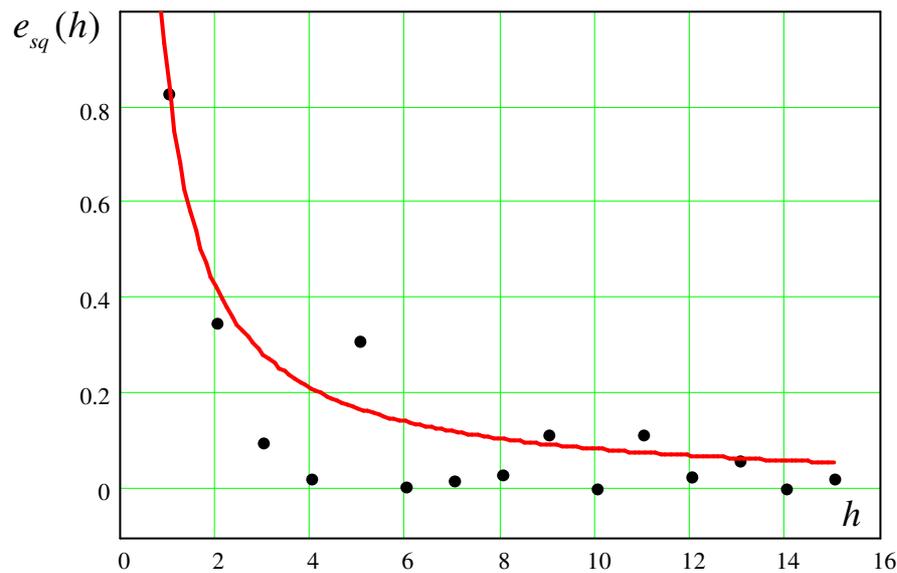


Рис. 3.3

Рассматриваемый пример позволяет также методом численного эксперимента исследовать зависимость статистической погрешности  $e_{sq}$  от величины шага  $h$ . Теоретическая кривая (3.14) изображена на рис. 3.3 непрерывной линией. Точками показаны значения погрешности, рассчитанные по формуле

$$\hat{e}_{sq}(h) = \left| \frac{y_{15} - y_{15-h}}{h} - y_*^{(1)} \right|.$$

Формула в данном случае дает именно статистическую погрешность, т.к. регулярная погрешность отсутствует (двухточечная аппроксимация производной точна для линейной функции). Группировка при больших шагах экспериментальных точек около теоретической кривой является аргументом в пользу справедливости последней.

## ГЛАВА 4. ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ

### 4.1. Введение

С необходимостью вычисления определенных интегралов вида

$$I = \int_a^b f(x)dx$$

приходится сталкиваться при решении многих физических задач. При этом подынтегральная функция  $f(x)$  обычно определяется одним из трех способов:

1. Задается явная формула для  $f(x)$ .
2. Функция  $f(x)$  явно не задана, но ее значения могут быть рассчитаны для любых значений аргумента из отрезка  $[a, b]$ . Это можно сделать, например, путем реализации некоторого вычислительного алгоритма.
3. Функция  $f(x)$  задается таблично, часто по результатам эксперимента.

В первом случае интегралы иногда удается вычислить аналитически, а во втором и третьем задача интегрирования решается лишь численными методами. К численным методам прибегают и тогда, когда интеграл от явно заданной функции не выражается через элементарные или специальные функции.

Пусть  $f(x)$  – ограниченная на отрезке  $[a, b]$  функция. Тогда наиболее часто применяемый метод численного интегрирования состоит в том, что интеграл заменяется некоторой линейной комбинацией значений  $f(x_n)$ , вычисленных в  $N+1$  точке отрезка  $[a, b]$ :

$$I = \int_a^b f(x)dx \approx \sum_{n=0}^N A_n f(x_n). \quad (4.1)$$

Приближенное равенство (4.1) называется квадратурной формулой, коэффициенты  $A_n$  – квадратурными коэффициентами, а точки  $x_n$  – узлами квадратурной формулы.

Большинство квадратурных формул получается путем замены подынтегральной функции на всем отрезке интегрирования или на его частях на более простую функцию  $g(x)$ , интегрируемую аналитически и принимающую в узлах квадратурной формулы значения  $g(x_n) = f(x_n)$ . В качестве  $g(x)$  можно взять полиномы, тригонометрические, экспоненциальные и другие функции. Конкретный выбор аппроксимирующей функции  $g(x)$  может зависеть от вида подынтегральной функции  $f(x)$  или определяться физическим содержанием решаемой задачи.

Важной особенностью квадратурной формулы является положение ее узлов. В ряде формул узлы следуют друг за другом с постоянным шагом, в других – выбор узлов основан, например, на соображениях максимальной точности.

Рассмотрим некоторые из наиболее распространенных квадратурных формул.

## 4.2. Формулы Ньютона – Котеса

Выберем на отрезке  $[a, b]$  с постоянным шагом  $N+1$  узел интерполяции  $x_0 = a, x_1 = a + h, \dots, x_n = b$  и построим интерполяционный полином  $P_N(x)$ . Положив  $f(x) = P_N(x)$  и проведя интегрирование, получим группу формул Ньютона – Котеса. Формулы, принадлежащие этой группе, различаются числом квадратурных узлов.

Для определения вида квадратурных коэффициентов Ньютона – Котеса полином  $P_N(x)$  удобно представить в форме Лагранжа:

$$P_N(x) = \sum_{n=0}^N f(x_n)L_n(x).$$

Тогда формальная процедура интегрирования приводит к выражению

$$I \approx \int_a^b P_N(x)dx = \sum_{n=0}^N f(x_n) \int_a^b L_n(x)dx.$$

Сравнивая его с выражением (4.1), находим общий вид квадратурных коэффициентов в формулах Ньютона – Котеса:

$$A_n = \int_a^b L_n(x)dx. \quad (4.2)$$

Простейшей из формул Ньютона – Котеса является формула прямоугольников, соответствующая интерполяции подынтегральной функции полиномом нулевой степени – константой. Для вычисления единственного квадратурного коэффициента этой формулы в выражении (4.2) следует положить  $L_0(x) = 1$ . после чего находим  $A_0 = b - a$  и формула прямоугольников записывается в виде

$$I_R = (b - a)f(a).$$

Далее в группе формул Ньютона-Котеса следует формула трапеций, являющаяся результатом интерполяции функции  $f(x)$  полиномом первой степени по узловым точкам  $x_0 = a, x_1 = a + h = b$ . В этом случае полиномы Лагранжа имеют вид

$$L_0(x) = -\frac{x-x_1}{h}, \quad L_1(x) = \frac{x-x_0}{h}.$$

Вычисления интегралов (4.2) дают значения  $A_0 = A_1 = h/2$ , и формула трапеций записывается как

$$I_T = \frac{b-a}{2}[f(a) + f(b)]. \quad (4.3)$$

Пусть теперь подынтегральная функция интерполируется по узлам  $x_0 = a$ ,  $x_1 = a+h$ ,  $x_2 = a+2h = b$ . Квадратичный интерполяционный полином  $P_2(x)$  выражается через три полинома Лагранжа

$$L_0(x) = \frac{1}{2h^2}(x-x_1)(x-x_2), \quad L_1(x) = -\frac{1}{h^2}(x-x_0)(x-x_2),$$

$$L_2(x) = \frac{1}{2h^2}(x-x_0)(x-x_1).$$

Использование этих выражений в интегралах (4.2) позволяет получить следующие значения квадратурных коэффициентов

$$A_0 = A_2 = \frac{1}{3}h, \quad A_1 = \frac{4}{3}h$$

и записать формулу

$$I_S = \frac{b-a}{3} \left[ f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right]. \quad (4.4)$$

Она называется формулой Симпсона или формулой парабол.

Продолжая и далее эту процедуру построения квадратурных формул, можно получить, в частности, формулу «трех восьмых»

$$I_V = \frac{3(b-a)}{8} [f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)]$$

и формулу Боде

$$I_B = \frac{2(b-a)}{45} [7f(x_0) + 32f(x_1) + 12f(x_2) + 32f(x_3) + 7f(x_4)].$$

### 4.3. Погрешности формул Ньютона – Котеса

Оценим теперь погрешности, возникающие при использовании описанных выше квадратурных формул. Так как в них во всех в качестве приближенного значения интеграла берется интеграл от интерполяционного полинома, то допускаемая при этом ошибка есть

$$E = \int_a^b [f(x) - P_n(x)] dx.$$

С учетом выражения (1.3) для погрешности интерполяции это равенство принимает вид

$$E = \frac{1}{(N+1)!} \int_a^b (x-x_0)(x-x_1)\dots(x-x_n) f^{(N+1)}(z(x)) dx, \quad (4.5)$$

где  $z(x)$  – зависящая от  $x$  точка отрезка  $[a, b]$ .

Применим оценку (4.5) к конкретным квадратурным формулам. При этом будем считать, что производные ограничены значениями  $M_n = \max_{a \leq x \leq b} |f^{(n)}(x)|$ .

В случае формулы прямоугольников  $N=0$ ,  $x_0 = a$ ,  $b = a + h$  и оценка погрешности преобразуется к виду

$$|E_R| = \left| \int_a^{a+h} (x-a) f''(z(x)) dx \right| \leq \frac{M_1}{2} (b-a)h. \quad (4.6)$$

Для формулы трапеций (4.3) при  $N=1$ ,  $x_0 = a$ ,  $x_1 = b = a + h$  из (4.5) находим

$$|E_T| = \frac{1}{2} \left| \int_a^{a+h} (x-a)(x-a-h) f''(z(x)) dx \right| \leq \frac{M_2}{12} (b-a)h^2. \quad (4.7)$$

Наконец, для формулы «трех восьмых» аналогичным образом находим следующую оценку погрешности

$$|E_V| \leq \frac{3M_4}{80} (b-a)h^4. \quad (4.8)$$

Для центрированных квадратурных формул, к которым относится формула Симпсона, соотношение (4.5) дает заниженную оценку погрешности. Для них погрешность удобнее оценивать, используя разложения подынтегральных функций в ряды Тейлора. Расчеты удобно проводить в математическом пакете MathCAD:

$$I(h) := \int_{-h}^h f(x) dx \quad I_s(h) := \frac{h}{3} \cdot (f(-h) + 4 \cdot f(0) + f(h))$$

$$(I(h) - I_s(h)) \text{ series, } h, 8 \rightarrow \frac{-1}{90} \cdot \left. \begin{array}{l} t \leftarrow 0 \\ \frac{d^4}{dt^4} f(t) \end{array} \right| \cdot h^5 - \frac{1}{1890} \cdot \left. \begin{array}{l} t \leftarrow 0 \\ \frac{d^6}{dt^6} f(t) \end{array} \right| \cdot h^7$$

Таким образом, главный член погрешности формулы Симпсона имеет вид

$$|E_s| \leq \frac{M_4}{180} (b-a)h^4. \quad (4.9)$$

Метод разложения в ряды Тейлора позволяет получить еще одну полезную формулу интегрирования, известную под названием формулы средней точки. Вычислим интеграл на отрезке  $[a, b = a + h]$ , используя в качестве опорного значение функции в середине этого отрезка. С точностью до слагаемых порядка  $h^3$  вычисления дают следующий результат:

$$\int_{-\frac{h}{2}}^{\frac{h}{2}} f\left(x + \frac{a+b}{2}\right) dx \approx f\left(\frac{1}{2} \cdot a + \frac{1}{2} \cdot b\right) \cdot h + \frac{1}{24} \cdot \left. \frac{d^2}{dt^2} f(t) \right|_{t = \frac{1}{2} \cdot a + \frac{1}{2} \cdot b} \cdot h^3$$

Первое слагаемое в этом разложении определяет приближенное значение интеграла:

$$I_{CP} = (b-a) f\left(\frac{a+b}{2}\right),$$

а второе – главный член погрешности:

$$|E_T| \leq \frac{M_2}{24} (b-a)h^2.$$

Формула средней точки широко используется при разработке численных алгоритмов различного назначения.

#### 4.4. Составные квадратурные формулы

Во все приведенные оценки (4.6)-(4.9) входят степени величины шага  $h$ . Если эта величина не мала, то не следует ожидать значительной точности вычисления интеграла. На практике малость шага  $h$  можно обеспечить двумя способами. Один из них состоит в увеличении числа узлов  $N+1$  квадратурной формулы. Тогда при фиксированной длине отрезка  $[a, b]$  шаг уменьшается, как  $(b-a)/N$ , и можно ожидать достижения приемлемой точности интегрирования. Однако при этом приходится применять формулу Ньютона – Котеса, основанную на высокой степени интерполяционного полинома. Такие полиномы подвержены осцилляциям (явление Рунге). Поэтому формулы с большим числом квадратурных узлов на практике применяют редко. Предпочтение отдают формулам низкого порядка.

Второй способ уменьшения шага заключается в том, что отрезок интегрирования  $[a, b]$  делится на несколько частей. Затем на каждой отдельно

взятой части применяется избранная квадратурная формула и результаты суммируются. Этим способом в вычислительную практику вводятся так называемые составные формулы. Как нетрудно показать, составные формулы прямоугольников, трапеций и Симпсона при фиксированном шаге  $h$  имеют вид

$$\begin{aligned} I_R &= h \sum_{n=0}^{N-1} f(x_n), \\ I_T &= \frac{h}{2} \sum_{n=0}^{N-1} [f(x_n) + f(x_{n+1})], \\ I_S &= \frac{h}{3} \sum_{n=0}^{N-1} [f(x_n) + 4f((x_n + x_{n+1})/2) + f(x_{n+1})]. \end{aligned} \quad (4.10)$$

Для погрешностей составных формул (4.10) справедливы приведенные выше оценки (но разность  $b - a$  при этом не является малой величиной). Таким образом, из оценок (4.6)-(4.8) следует, что составные формулы прямоугольников и трапеций имеют первый и второй порядок точности, в то время как составная формула Симпсона – четвертый порядок. В силу этой относительно высокой точности составная формула Симпсона особенно часто используется в практике численного интегрирования. Однако следует иметь в виду, что составная формула Симпсона оперирует с нечетным числом квадратурных узлов. Если по каким-либо причинам это условие выполнить не удастся, то можно применять имеющую тот же порядок точности составную формулу «трех восьмых»

#### 4.5. Метод Ромберга

Этот метод численного интегрирования основан на том, что, составляя определенные линейные комбинации формул Ньютона – Котеса при различных шагах сетки, можно получить квадратурные формулы более высокого порядка. Убедимся в этом на примере формулы трапеций.

При  $b - a = h_0$  по формуле трапеций имеем следующее приближенное значение интеграла

$$T_0^{(0)} = \frac{h_0}{2} [f(a) + f(b)].$$

Если же отрезок интегрирования разделить на две части длины  $h_1 = h_0 / 2$ , то, применяя составную формулу трапеций, получим

$$T_0^{(1)} = \frac{h_1}{2} [f(a) + 2f(a + h_1) + f(b)].$$

Составим теперь линейную комбинацию

$$T_1^{(0)} = \frac{4T_0^{(1)} - T_0^{(0)}}{3}.$$

Простые преобразования сводят ее к выражению

$$T_1^{(0)} = \frac{h_1}{3} [f(a) + 4f(a + h_1) + f(b)], \quad (4.11)$$

которое представляет собой не что иное, как формулу Симпсона. Она имеет четвертый порядок точности по шагу сетки, в то время как исходная формула трапеций – второй. Таким образом, налицо возрастание порядка точности аппроксимации интеграла без существенного увеличения объема вычислений.

Продолжим деление отрезка  $[a, b]$  на части. Для четырех частей длины  $h_2 = h_0 / 4$  составная формула трапеций имеет вид

$$T_0^{(2)} = \frac{h_2}{2} [f(a) + 2f(a + h_2) + 2f(a + 2h_2) + 2f(a + 3h_2) + f(b)].$$

Линейная комбинация

$$T_1^{(1)} = \frac{4T_0^{(2)} - T_0^{(1)}}{3}$$

дает составную формулу Симпсона

$$T_1^{(1)} = \frac{h_2}{3} [f(a) + 4f(a + h_2) + 2f(a + 2h_2) + 4f(a + 3h_2) + f(b)]. \quad (4.12)$$

Комбинируя же формулы (4.11) и (4.12) по правилу

$$T_2^{(0)} = \frac{16T_1^{(1)} - T_1^{(0)}}{15},$$

получим формулу Боде

$$T_2^{(0)} = \frac{2h_2}{45} [7f(a) + 32f(a + h_2) + 12f(a + 2h_2) + 32f(a + 3h_2) + 7f(b)].$$

Ее точность на два порядка выше, чем точность формулы Симпсона.

В продолжение описанного вычислительного процесса Ромберг предложил строить  $T$ -схему

$$\begin{array}{cccccc}
T_0^{(0)} & & & & & \\
T_0^{(1)} & T_1^{(0)} & & & & \\
T_0^{(2)} & T_1^{(1)} & T_2^{(0)} & & & \\
T_0^{(3)} & T_1^{(2)} & T_2^{(1)} & T_3^{(0)} & & \\
\dots & \dots & \dots & \dots & \dots & \dots,
\end{array}$$

которая задается общей формулой

$$T_n^{(m)} = \frac{4^n T_{n-1}^{(m+1)} - T_{n-1}^{(m)}}{4^n - 1}. \quad (4.13)$$

Все столбцы и все диагонали  $T$ -схемы сходятся к искомому значению интеграла, причем наибольшую скорость сходимости обеспечивает самая верхняя диагональ.

При интегрировании по методу Ромберга, сравнивая результаты каждого шага итерационного процесса (4.13), достаточно просто контролировать точность вычислений.

Алгоритм реализован в MathCAD-программе *IntRomb*. Программа интегрирует таблично заданную функцию по  $N = 2^v + 1$  узлам с абсолютной точностью не менее  $\varepsilon$ . Вариант тестовой программы интегрирования выглядит следующим образом.

$$\int_0^1 \sin(\pi \cdot x) dx \rightarrow \frac{2}{\pi} = 0.6366198$$

$$v := 4 \quad n := 0..2^v \quad f_n := \sin\left(\pi \cdot \frac{n}{2^v}\right) \quad \text{IntRomb}(0, 1, f, 0.0001) = 0.6366198$$

$$\text{IntRomb}(0, 1, f, 0) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0.6666667 & 0 & 0 & 0 \\ 0.6035534 & 0.6380712 & 0.6361648 & 0 & 0 \\ 0.6284174 & 0.6367055 & 0.6366144 & 0.6366215 & 0 \\ 0.6345731 & 0.6366251 & 0.6366197 & 0.6366198 & 0.6366198 \end{pmatrix}$$

```

IntRomb(a, b, f, ε) :=
  N ← length(f) - 1
  v ← log(N, 2)
  h ← b - a
  T0,0 ← 0.5·h·(f0 + fN)
  Int ← T0,0
  for n ∈ 1..v
    h ← 0.5·h
    M ← 2n
    Tn,0 ←  $\frac{h}{2} \cdot \left( f_0 + 2 \cdot \sum_{k=1}^{M-1} f_{k \cdot \frac{N}{M}} + f_N \right)$ 
    for m ∈ 1..n
      Tn,m ←  $\frac{4^m \cdot T_{n,m-1} - T_{n-1,m-1}}{4^m - 1}$ 
    return Tn,n if |Int - Tn,n| ≤ ε
  Int ← Tn,n
T

```

#### 4.6. Квадратурные формулы Гаусса

Во всех рассмотренных до сих пор квадратурных формулах используются равноотстоящие узлы. Но рассматриваются и другие способы расположения узлов. Например, узлы в формулах Гаусса выбираются таким образом, чтобы при их наименьшем числе можно было точно интегрировать полиномы наивысшей степени. Рассмотрим это подробнее.

При  $N+1$  гауссовом узле формула (4.1) содержит  $2N+2$  неизвестных величин: квадратурные коэффициенты и координаты квадратурных узлов. Столько же неизвестных коэффициентов имеет и полином степени  $2N+1$ . Поэтому, потребовав, чтобы квадратурная формула была точна для всех полиномов той же или меньшей степени, получим систему из  $2N+2$  уравнений с  $2N+2$  неизвестными  $A_0, A_1, A_2, \dots, A_N, x_0, x_1, x_2, \dots, x_N$ . Несмотря на нелинейность, эта система имеет решение, и, следовательно, можно построить квадратурную формулу, точно интегрирующую полином степени  $2N+1$  по  $N+1$  узлу.

Так как благодаря замене переменной

$$x = \frac{2X - (a+b)}{b-a}, \quad F(X) = f(x)$$

интеграл с произвольными пределами интегрирования приводится к виду

$$I = \int_a^b F(X) dX = \frac{b-a}{2} \int_{-1}^1 f(x) dx,$$

то при выводе квадратурных формул Гаусса без ограничения общности можно считать  $[a, b] = [-1, 1]$ .

Пусть квадратурная формула (4.1) точно интегрирует любой полином вида

$$P_{2N+1}(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_{2N+1} x^{2N+1}.$$

Тогда должно иметь место равенство

$$a_0 \mu_0 + a_1 \mu_1 + a_2 \mu_2 + \dots + a_{2N+1} \mu_{2N+1} = \sum_{n=0}^N A_n P_{2N+1}(x_n),$$

где  $\mu_m = (1 - (-1)^{m+1}) / (m+1)$ . Из требования о сохранении этого равенства при любых значениях коэффициентов полинома  $a_m$  следует система нелинейных алгебраических уравнений для коэффициентов и узлов квадратурной формулы:

$$\sum_{n=0}^N A_n x_n^m = \mu_m, \quad m = 0, 1, 2, \dots, 2N+1. \quad (4.14)$$

В частном случае  $N=1$  система уравнений (4.14) принимает вид

$$\begin{aligned} A_0 + A_1 &= 2, \\ A_0 x_0 + A_1 x_1 &= 0, \\ A_0 x_0^2 + A_1 x_1^2 &= 2/3, \\ A_0 x_0^3 + A_1 x_1^3 &= 0. \end{aligned}$$

Ее решением, как нетрудно проверить, являются числа  $A_0 = A_1 = 1$ ,  $x_0 = -\sqrt{3}/3$ ,  $x_1 = \sqrt{3}/3$ . В результате мы получаем широко известную формулу Гаусса с двумя узлами:

$$\int_{-1}^1 f(x) dx \approx G_2(f) = f(-\sqrt{3}/3) + f(\sqrt{3}/3).$$

Для произвольного отрезка интегрирования  $[a, b]$  с учетом проведенной выше замены переменной имеем

$$G_2(F) = \frac{b-a}{2} \left[ F\left(\frac{a+b}{2} - \frac{\sqrt{3}}{3} \frac{b-a}{2}\right) + F\left(\frac{a+b}{2} + \frac{\sqrt{3}}{3} \frac{b-a}{2}\right) \right]. \quad (4.15)$$

Формула (4.15), так же как и формула Симпсона (4.4), точно интегрирует полином третьей степени. Поэтому их погрешности при вычислении интеграла от произвольной функции одного порядка. Но при использовании квадратурной формулы Гаусса требуется меньше вычислений функции, что убыстряет вычислительный процесс.

Для произвольного  $N$  решение системы уравнений (4.14) можно получить, находя нули полинома Лежандра степени  $N+1$ :

$$P_{N+1}(x_n) = 0, \quad n = 0, 1, 2, \dots, N, \quad (4.16)$$

и с помощью формул

$$A_n = \frac{2(1-x_n^2)}{(N+1)^2 P_N^2(x_n)}. \quad (4.17)$$

Два первых полинома Лежандра имеют вид

$$P_0(x) = 1, \quad P_1(x) = x,$$

а все полиномы высших порядков могут быть найдены из рекуррентного соотношения

$$(n+1)P_{n+1}(x) - (2n+1)xP_n(x) + nP_{n-1}(x) = 0.$$

В частности, при  $n=2$  имеем

$$P_2(x) = \frac{1}{2}(3x^2 - 1), \quad P_3(x) = \frac{1}{2}(5x^3 - 3x).$$

Нули полинома  $P_3(x)$  дают следующие узлы квадратурной формулы:

$$x_0 = -\sqrt{\frac{3}{5}}, \quad x_1 = 0, \quad x_2 = \sqrt{\frac{3}{5}},$$

а из (4.17) находим квадратурные коэффициенты

$$A_0 = A_2 = \frac{5}{9}, \quad A_1 = \frac{8}{9}.$$

Таким образом, формула Гаусса с тремя узлами имеет вид

$$\int_{-1}^1 f(x) dx \approx G_3(f) = \frac{1}{9} [5f(-\sqrt{3/5}) + 8f(0) + 5f(\sqrt{3/5})].$$

В основе популярности квадратурных формул Гаусса лежит их высокий порядок точности при меньшем, чем в формулах Ньютона – Котеса, объеме вычислений.

## 4.7. Интегрирование методом Монте-Карло

Метод Монте-Карло дает эффективный способ вычисления многомерных интегралов. В отличие от рассмотренных ранее регулярных методов интегрирования, он основан на результатах случайных испытаний, в связи с чем от названия известного казино и возникло его название. Основная идея метода состоит в том, чтобы вычислять подынтегральную функцию не во всех квадратурных узлах, число которых может быть очень велико, а только на некотором их подмножестве, являющемся репрезентативной случайной выборкой из множества узлов. Ситуация здесь аналогична, например, предсказанию результатов выборов путем опроса небольшого числа избирателей.

Хотя метод Монте-Карло эффективен лишь при вычислении интегралов большой размерности, его основная идея наиболее наглядно иллюстрируется решением одномерной задачи. Будем считать, что требуется вычислить интеграл

$$I = \int_0^1 f(x) dx. \quad (4.18)$$

Выбор отрезка интегрирования  $[0, 1]$  не ограничивает общности рассмотрения, так как соответствующей заменой переменной интегрирования к нему всегда можно привести произвольный отрезок  $[a, b]$ .

Сравнивая (4.18) с общим выражением для статистического среднего функции  $f(x)$

$$\langle f \rangle = \int_{-\infty}^{\infty} f(x) W(x) dx, \quad (4.19)$$

нетрудно сделать вывод о том, что искомый интеграл совпадает со средним значением функции случайной величины с плотностью вероятности

$$W_U(x) = \begin{cases} 0 & \text{при } x < 0 \text{ и } x > 1; \\ 1 & \text{при } 0 \leq x \leq 1, \end{cases}$$

то есть равномерно распределенной на отрезке  $[0, 1]$  случайной величины.

Используем выборочное среднее для оценки статистического среднего (4.19):

$$\hat{f} = \frac{1}{N} \sum_{n=0}^{N-1} f(x_n).$$

Тогда, приняв за приближенное значение интеграла (4.19) его оценку способом выборочного среднего, получим квадратурную формулу метода Монте-Карло:

$$I \approx \hat{I} = \frac{1}{N} \sum_{n=0}^{N-1} f(x_n). \quad (4.20)$$

В этой формуле подынтегральная функция вычисляется при значениях аргумента из множества равномерно распределенных на отрезке  $[0, 1]$  случайных чисел.

На практике вычисления по формуле (4.20) реализуются с использованием последовательностей программно генерируемых случайных чисел. При этом оценка интеграла является величиной, принимающей случайные значения на каждой из сгенерированных последовательностей. Однако можно показать, что математическое ожидание случайной величины  $\hat{I}$  равно  $I$ , то есть формула (4.20) дает несмещенную оценку интеграла. Кроме того, оценка (4.20) состоятельна, так как ее дисперсия

$$\sigma_I^2 = \frac{1}{N} \sigma_f^2$$

стремится к нулю с ростом  $N$  – размера выборки случайных значений аргумента. Здесь  $\sigma_f^2$  – дисперсия случайной последовательности значений подынтегральной функции.

Погрешностью в оценке значения интеграла следует считать среднеквадратичное отклонение

$$\sigma_I = \frac{1}{\sqrt{N}} \sigma_f. \quad (4.21)$$

Данное выражение показывает, что хотя ошибка интегрирования методом Монте-Карло с ростом  $N$  и уменьшается, это уменьшение происходит очень медленно. Поэтому для одномерных интегралов метод Монте-Карло по вычислительной эффективности существенно уступает рассмотренным ранее методам. Например, в методе трапеций ошибка убывает как  $N^{-2}$ . Однако, если мы применим многомерный аналог метода трапеций к вычислению  $p$ -мерного интеграла, то ошибка в этом случае будет убывать уже как  $N^{-2/p}$ . Следовательно, при  $p > 4$  метод Монте-Карло обеспечивает вычисления с ошибкой, уменьшающейся быстрее, чем в методе трапеций. Этот качественный вывод о возрастании относительной эффективности метода Монте-Карло при повышении кратности интеграла справедлив и в отношении других регулярных методов численного интегрирования.

Обратим теперь внимание на то, что, согласно выражению (4.21), ошибка интегрирования методом Монте-Карло пропорциональна среднеквадратичному отклонению подынтегральной функции  $\sigma_f$ . Очевидно, что при равномерном распределении аргумента дисперсия функции тем больше, чем больше интервал изменения ее значений. В частности, при

$f(x) = \text{const}$  дисперсия  $\sigma_f^2 = 0$  и формула (4.20) дает точное значение интеграла. Поэтому можно предложить следующий способ уменьшения статистической погрешности метода. Путем тождественных преобразований с участием функции  $W(x)$ , не принимающей на отрезке интегрирования нулевых и отрицательных значений, исходный интеграл приводится к виду

$$I = \int_0^1 F(x)W(x)dx, \quad (4.22)$$

где  $F(x) = f(x)/W(x)$ . Если неотрицательная функция  $W(x)$  удовлетворяет условию нормировки

$$\int_0^1 W(x)dx = 1,$$

то ее можно рассматривать как плотность вероятности некоторой случайной величины. Тогда интеграл (4.22) есть статистическое среднее функции  $F(x)$  по ансамблю значений аргумента с плотностью вероятностного распределения  $W(x)$ . Приняв в качестве его оценки выборочное среднее, получим квадратурную формулу

$$I \approx \hat{I} = \frac{1}{N} \sum_{n=0}^{N-1} F(x_n) = \frac{1}{N} \sum_{n=0}^{N-1} \frac{f(x_n)}{W(x_n)}. \quad (4.23)$$

Особо отметим, что здесь, в отличие от формулы (4.20), выборочное среднее вычисляется не по равномерно распределенной случайной величине, а по случайной величине, характеризуемой плотностью вероятности  $W(x)$ . Метод интегрирования, основанный на формуле (4.23), часто называют методом выборки по значимости.

При выборе функции  $W(x)$  следует учитывать вид подынтегральной функции  $f(x)$  и возможность программной генерации случайных чисел с заданным вероятностным распределением. Например, для функций  $f(x)$  с одним максимумом в области интегрирования вычисления по формуле (4.21) можно проводить на основе усеченного гауссовского распределения.

Для генерации случайных чисел с заданной плотностью вероятности  $W(x)$  можно воспользоваться следующим методом обратного преобразования случайных величин. Пусть  $x = X(y)$  – монотонно возрастающая функция, а случайные числа  $y$  имеют равномерное распределение с плотностью вероятности  $W_U(y)$ . Тогда имеет место вполне очевидное соотношение

$$W_U(y)dy = W(x)dx,$$

из которого следует, что

$$y = Y(x) = \int_0^x W(x') dx'.$$

Если на основе этого выражения удастся найти обратную функцию  $x = Y^{-1}(y) = X(y)$ , то полученное преобразование эффективно решает поставленную задачу, т.е. последовательность равномерно распределенных случайных чисел  $y_n$  преобразуется в последовательность случайных чисел  $x_n$  с требуемой плотностью вероятности  $W(x)$ .

#### 4.8. Интегралы от функций с особенностями и несобственные интегралы

Если в интервале интегрирования подынтегральная функция или ее производная обращаются в бесконечность, то прямое применение квадратурных формул сопровождается большими погрешностями или вообще приводит к бессмысленным результатам. В таком случае путем замены переменной интегрирования следует попытаться устранить особые точки. Зачастую этот путь приводит к положительным результатам. Например, в выражении

$$p(z) = \frac{1}{\sigma\sqrt{2\pi^3}} \int_{-A}^A \exp[-(x-z)^2/2\sigma^2] \sqrt{A^2-x^2} dx,$$

описывающем плотность вероятности смеси гармонического сигнала и нормального шума, интеграл обращается в бесконечность на границах отрезка: при  $x = \pm A$ . Но после замены переменной  $x = A \cos y$  он преобразуется к виду

$$p(z) = \frac{1}{\sigma\sqrt{2\pi^3}} \int_0^\pi \exp[-(A \cos y - z)^2/2\sigma^2] dy,$$

где подынтегральная функция не имеет особенностей во всей области интегрирования.

Можно также выделить особенности, разделив интервал интегрирования на части. Например, пусть функция  $f(x)$  вблизи нуля возрастает, как  $Cx^{-\nu}$  при  $0 < \nu < 1$ . Тогда интеграл от  $f(x)$  на отрезке  $[0, b]$  следует представить в виде суммы

$$\int_0^b f(x) dx = \int_0^h f(x) dx + \int_h^b f(x) dx.$$

Подынтегральная функция не имеет особенностей на отрезке  $[h, b]$ , и второй интеграл здесь может быть вычислен по квадратурным формулам. Первый же интеграл приближенно вычисляется аналитически

$$\int_0^h f(x)dx \approx \frac{C}{1-\nu} h^{1-\nu} = h \frac{f(h)}{1-\nu} .$$

Аналогичным образом, выделяя малую область в окрестности особой точки, можно устранить и другие интегрируемые особенности.

Вычисление несобственных интегралов вида

$$I = \int_a^{\infty} f(x)dx$$

заменой переменной  $y = 1/x$  можно свести к вычислению собственных интегралов

$$I = \int_0^{1/a} (F(y)/y^2)dy, \quad F(y) = f(1/y) .$$

Такая замена бесполезна, если нижний предел интегрирования равен нулю. Однако это затруднение часто удается устранить, разбивая отрезок интегрирования на две части:

$$I = \int_0^{\infty} f(x)dx = \int_0^a f(x)dx + \int_0^{1/a} (F(y)/y^2)dy .$$

Если несобственный интеграл не удается свести к собственному, то он представляется в виде суммы парциальных интегралов

$$I = \int_a^b f(x)dx + \int_b^{2b} f(x)dx + \int_{2b}^{4b} f(x)dx + \dots ,$$

вычисляемых на отрезках с удвоением верхнего предела до тех пор, пока величина очередного слагаемого не станет меньше заданной погрешности интегрирования.

## Библиографический список

1. Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы. – 3-е изд., доп. и перераб. – М.: БИНОМ. Лаборатория знаний, 2011. 636 с.
2. Зализняк В.Е. Численные методы. Основы научных вычислений. – 2-е изд., перераб. и доп. – М.: Юрайт, 2012. 356 с.
3. Турчак Л.И., Плотников П.В. Основы численных методов. – М.: ФИЗМАТЛИТ, 2002. 304 с.
4. Пирумов У.Г. Численные методы. – 2-е изд., перераб. и доп. – М.: Дрофа, 2003. 224 с.
5. Формалев В.Ф., Ревизников Д.Л. Численные методы. – М.: ФИЗМАТЛИТ, 2004. 400 с.
6. Мэтьюз Д., Финк К. Численные методы. Использование MATLAB. – М.: Издательский дом «Вильямс», 2001. 720 с.
7. Махмутов М.М. Лекции по численным методам. – М. – Ижевск: НИЦ РХД, Институт компьютерных исследований, 2007. 238 с.
8. Численные методы. Сборник задач / Под ред. У.Г. Пирумова – М.: Дрофа, 2007. 144 с.

## ОГЛАВЛЕНИЕ

<b>Предисловие</b> .....	3
<b>Глава 1. ИНТЕРПОЛЯЦИЯ</b> .....	4
1.1. Введение .....	4
1.2. Полиномиальная интерполяция. Метод неопределенных коэффициентов .....	5
1.3. Интерполяция по Лагранжу .....	6
1.4. Итерационный метод интерполяции.....	7
1.5. Разностный метод Ньютона .....	8
1.6. Кубические сплайны .....	12
1.7. Примеры полиномиальной интерполяции .....	15
1.8. Интерполяция В-сплайнами .....	18
<b>Глава 2. АППРОКСИМАЦИЯ МЕТОДОМ НАИМЕНЬШИХ КВАДРАТОВ</b> .....	22
2.1. Введение .....	22
2.2. Общая формулировка метода наименьших квадратов .....	22
2.3. Линейный метод наименьших квадратов .....	25
2.4. Полиномиальная аппроксимация методом НК .....	27
2.5. Аппроксимация ортогональными полиномами .....	29
2.6. Метод НК и линейная регрессия .....	33
<b>Глава 3. ЧИСЛЕННОЕ ДИФФЕРЕНЦИРОВАНИЕ И АППРОКСИМАЦИЯ ПРОИЗВОДНЫХ</b> .....	38
3.1. Введение .....	38
3.2. Метод неопределенных коэффициентов .....	38
3.3. Формулы численного дифференцирования .....	39
3.4. Погрешность формул .....	43
3.5. Метод Рунге – Ромберга – Ричардсона .....	45
3.6. Некорректность численного дифференцирования .....	46
<b>Глава 4. ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ</b> .....	50
4.1. Введение .....	50
4.2. Формулы Ньютона – Котеса .....	51
4.3. Погрешность формул Ньютона – Котеса .....	52
4.4. Составные квадратурные формулы .....	54
4.5. Метод Ромберга .....	55
4.6. Квадратурные формулы Гаусса.....	57
4.7. Интегрирование методом Монте-Карло .....	61
4.8. Интегралы от функций с особенностями и несобственные интегралы .....	64
Библиографический список .....	66