Белорусский государственный университет
кафедра вычислительной математики

Б. В. Фалейчик

МЕТОДЫ ЧИСЛЕННОГО АНАЛИЗА

конспект лекций

2010–2011

Содержание

1 Приближение функций

1.1 Задача интерполяции. Общая постановка

Рассмотрим набор попарно различных точек $\{x_i\}_{i=0}^n$, $x_i \in [a,b]$. Пусть $\{y_i\}_{i=0}^n$ — значения некоторой функции $f : [a,b] \to \mathbb{R}$ в этих точках: $y_i = f(x_i)$. Рассмотрим также набор линейно независимых базисных функций $\varphi_i : [a,b] \to \mathbb{R}$, $i = \overline{0,n}$. *Задача (линейной) интерполяции* заключается в нахождении функции

$$\varphi = \sum_{i=0}^{n} \alpha_i \varphi_i, \quad \alpha_i \in \mathbb{R},$$

такой, что

$$\varphi(x_i) = y_i \quad \forall\, i = \overline{0,n}. \tag{1.1}$$

Функция f называется *интерполируемой функцией*, φ — *интерполирующей функцией*, $\{x_i\}$ — *узлами интерполяции*, точки декартовой плоскости (x_i, y_i) — *точками интерполяции*.

Таким образом, задача интерполяции сводится к нахождению неизвестных коэффициентов $\{\alpha_i\}_{i=0}^n$ из условий (1.1). По определению φ это эквивалентно решению СЛАУ

$$\sum_{j=0}^{n} \alpha_j \varphi_j(x_i) = y_i, \quad i = \overline{0,n},$$

матричный вид которой запишем как

$$\Phi\alpha = y, \tag{1.2}$$

где $\alpha = (\alpha_0, \ldots, \alpha_n)^T$, $y = (y_0, \ldots, y_n)^T$,

$$\Phi = \begin{bmatrix} \varphi_0(x_0) & \varphi_1(x_0) & \cdots & \varphi_n(x_0) \\ \varphi_0(x_1) & \varphi_1(x_1) & \cdots & \varphi_n(x_1) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_0(x_n) & \varphi_1(x_n) & \cdots & \varphi_n(x_n) \end{bmatrix}. \tag{1.3}$$

Матрицу Φ для данных узлов $\{x_i\}$ и базиса $\{\varphi_i\}$ будем называть *матрицей интерполяции*.

Очевидно, что решение задачи интерполяции существует и единственно тогда и только тогда, когда $\det \Phi \neq 0$.

Система (1.2) имеет наиболее простой вид в случае, когда $\Phi = I$. Отсюда вытекает следующее определение.

Пусть $\{x_i\}_{i=0}^n$ — узлы интерполяции. Система функций $\{\varphi_i\}_{i=0}^n$ называется *фундаментальным базисом* для данного набора узлов, если

$$\varphi_i(x_j) = \delta_{ij} = \begin{cases} 1, & i = j; \\ 0, & i \neq j; \end{cases} \qquad \forall i, j = \overline{0, n}.$$

Таким образом, если $\{\varphi_i\}$ — фундаментальный базис, то задача интерполяции решается очень просто:

$$\varphi = \sum_{i=0}^{n} y_i \varphi_i.$$

Если же базис не фундаментален, но матрица Φ невырождена, то из $\{\varphi_i\}$ можно получить фундаментальный базис $\{\hat{\varphi}_i\}$ вида

$$\hat{\varphi}_i = \sum_{j=0}^{n} \beta_{ij} \varphi_j, \quad i = \overline{0, n}.$$

$\triangleright_1$ Как в общем случае можно найти коэффициенты β_{ij}?

1.2 Полиномиальная интерполяция

Классическим способом аппроксимации (= приближения) функций является интерполяция алгебраическими многочленами:

$$\varphi(x) = P_n(x) = \sum_{i=0}^{n} \alpha_i x^i, \quad \text{т. е.} \quad \varphi_i(x) = x^i. \tag{1.4}$$

В этом случае интерполирующую функцию φ называют *интерполяционным многочленом*.

Пусть заданы узлы интерполяции. Рассмотрим для базиса из (1.4) матрицу интерполяции (1.3):

$$\Phi = \begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{bmatrix} = V(x_0, \ldots, x_n). \tag{1.5}$$

Матрица такого вида называется матрицей Вандермонда.

Лемма 1.1.

$$|V(x_0, \ldots, x_n)| = \prod_{0 \leqslant j < i \leqslant n} (x_i - x_j) \tag{1.6}$$

Доказательство.

$$|V(x_0,\dots,x_n)| = \begin{vmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{vmatrix} = [\text{элем. преобр. столбцов}] =$$

$$= \begin{vmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & x_1 - x_0 & x_1^2 - x_1x_0 & \cdots & x_1^n - x_1^{n-1}x^0 \\ 1 & x_2 - x_0 & x_2^2 - x_2x_0 & \cdots & x_2^n - x_2^{n-1}x^0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n - x_0 & x_n^2 - x_nx_0 & \cdots & x_n^n - x_n^{n-1}x^0 \end{vmatrix} =$$

$$= (x_1 - x_0)(x_2 - x_0)\dots(x_n - x_0) \begin{vmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} \end{vmatrix} =$$

$$= (x_1 - x_0)(x_2 - x_0)\dots(x_n - x_0) \cdot |V(x_1,\dots,x_n)| = [\dots] = \prod_{0 \leqslant j < i \leqslant n} (x_i - x_j).$$

■

Следствие 1.1. *Если все узлы интерполяции $\{x_i\}$ попарно различны, то при любых значениях $\{y_i\}$ интерполяционный многочлен $\varphi = P_n$ существует и единственен.*

$\triangleright_2$ Докажите.

Следствие 1.2. *Любой многочлен степени n однозначно определяется своими значениями в $n+1$ попарно различных точках.*

$\triangleright_3$ Докажите.

1.3 Интерполяционный многочлен в форме Лагранжа

Построим фундаментальный базис многочленов для узлов $x_0, \dots, x_n$. Пусть Λ_i — i-й многочлен из этого базиса. По определению

$$\Lambda_i(x_j) = 0 \quad \forall j \neq i,$$

откуда сразу имеем

$$\Lambda_i(x) = C_i \prod_{j \neq i} (x - x_j), \quad C_i \in \mathbb{R}.$$

Неизвестную константу C_i находим из оставшегося условия

$$\Lambda_i(x_i) = 1 \quad \Rightarrow \quad C_i = \left(\prod_{j \neq i} (x_i - x_j) \right)^{-1},$$

откуда окончательно получаем

$$\Lambda_i(x) = \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}. \tag{1.7}$$

Таким образом получаем знаменитую формулу интерполяционного многочлена в форме Лагранжа:

$$P_n(x) = \sum_{i=0}^{n} y_i \Lambda_i(x) = \sum_{i=0}^{n} y_i \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}. \tag{1.8}$$

Для многочленов Λ_i существует альтернативная форма записи. Рассмотрим многочлен

$$\omega_{n+1}(x) = (x - x_0)(x - x_1) \dots (x - x_n). \tag{1.9}$$

Тогда числитель в (1.7) можно записать как

$$\frac{\omega_{n+1}(x)}{x - x_i},$$

а знаменатель как

$$\omega'_{n+1}(x_i).$$

Отсюда

$$\Lambda_i(x) = \frac{\omega_{n+1}(x)}{(x - x_i)\omega'_{n+1}(x_i)}. \tag{1.10}$$

1.4 Итого

Умные мысли

1. Интерполяция — наиболее простой и популярный способ приближения функций.
2. Понимание понятия фундаментального базиса очень пригодится в дальнейшем.
3. В частности потому, что если известен фундаментальный базис, то задача интерполяции решается легко и непринуждённо.
4. *Кто не будет уметь строить интерполяционный многочлен Лагранжа — останется на второй год!*

2 Интерполяционный многочлен в форме Ньютона

2.1 Построение

Интерполяционная формула Лагранжа (1.8) становится громоздкой и неудобной для вычислений при больших n. Более удобным на практике оказывается представление интерполяционного многочлена в так называемой форме Ньютона.

Рассмотрим ещё раз СЛАУ (1.2), решение которой является решением задачи интерполяции в общем случае:

$$\Phi\alpha = \begin{bmatrix} \varphi_0(x_0) & \varphi_1(x_0) & \cdots & \varphi_n(x_0) \\ \varphi_0(x_1) & \varphi_1(x_1) & \cdots & \varphi_n(x_1) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_0(x_n) & \varphi_1(x_n) & \cdots & \varphi_n(x_n) \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}.$$

При построении формулы Лагранжа мы выбирали базис $\{\varphi_i\}$ так, чтобы матрица Φ была единичной.

Теперь же построим базис, в котором эта матрица будет **нижнетреугольной**.

$$\Phi\alpha = \begin{bmatrix} \varphi_0(x_0) & 0 & 0 & \cdots & 0 \\ \varphi_0(x_1) & \varphi_1(x_1) & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \varphi_0(x_n) & \varphi_1(x_n) & \varphi_2(x_n) & \cdots & \varphi_n(x_n) \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}. \tag{2.1}$$

Такой базис должен, очевидно, удовлетворять условиям

$$\varphi_i(x_j) = 0 \quad \forall j = \overline{0, i-1},$$

откуда с точностью до постоянного множителя получаем

$$\varphi_i(x) = (x - x_0)(x - x_1)\dots(x - x_{i-1}) = \prod_{j=0}^{i-1}(x - x_j) = \omega_i(x) \tag{2.2}$$

(сравните с формулой (1.9)). Здесь при $i = 0$ как обычно полагаем $\omega_0(x) = 1$. Следовательно, матрица СЛАУ из (2.1) будет иметь вид

$$\Phi = \Omega = \Big(\omega_j(x_i)\Big)_{i,j=0}^{n} = \\ = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & x_1 - x_0 & 0 & \cdots & 0 \\ 1 & x_2 - x_0 & (x_2 - x_0)(x_2 - x_1) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n - x_0 & (x_n - x_0)(x_n - x_1) & \cdots & (x_n - x_0)\dots(x_n - x_{n-1}) \end{bmatrix}.$$

$\triangleright_1$ При каких условиях СЛАУ с такой матрицей имеет единственное решение?

Таким образом получаем интерполяционный многочлен в виде

$$P_n(x) = \sum_{i=0}^{n} \alpha_i \omega_i(x) = \alpha_0 + \alpha_1(x - x_0) + \alpha_2(x - x_0)(x - x_1) + \dots + \\ + \alpha_n(x - x_0)(x - x_1)\dots(x - x_{n-1}), \quad (2.3)$$

где коэффициенты α_i можно найти из (2.1) по рекуррентным соотношениям:

$$\alpha_i = \Big(y_i - \sum_{j=0}^{i-1} \alpha_j \omega_j(x_i)\Big) \Big/ \omega_i(x_i), \quad i = \overline{0, n}. \quad (2.4)$$

Многочлен P_n, определяемый формулами (2.3), (2.4) называется интерполяционным многочленом в форме Ньютона.

Заметим, что использование формул (2.4) *не является общепринятым* способом вычисления α_i (см. далее).

Следует понимать также, что формулы (2.3) *и* (1.8) *— суть разные формы записи одного и того же интерполяционного многочлена!*

По построению ИМ в форме Ньютона обладает рядом полезных свойств. Во-первых, базисные функции ω_i зависят только от узлов $x_0, \dots, x_{i-1}$, а коэффициенты α_i — только от $x_0, \dots, x_i$. Это позволяет легко «обновлять» формулу ИМ при добавлении дополнительных узлов интерполяции. Другими словами, если известен ИМ P_n (2.3) по узлам $x_0, \dots, x_n$, то ИМ по узлам $x_0, \dots, x_{n+1}$ может быть найден по формуле

$$P_{n+1}(x) = P_n(x) + \alpha_{n+1}\omega_{n+1}(x). \quad (2.5)$$

Во-вторых, форма (2.3) более удобна для вычислений, чем форма Лагранжа: последовательно вынося за скобки общие множители $(x - x_i)$, можно вычислить $P_n(x)$ по т. н. схеме Горнера

$$P_n(x) = \alpha_0 + (x - x_0)\Big(\alpha_1 + (x - x_1)\big(\alpha_2 + \dots\big)\Big). \quad (2.6)$$

$\triangleright_2$ Подсчитайте количество арифметических операций, нужных для вычисления $P_n(x)$ по формуле (2.6).

2.2 Разделённые разности

Пусть $\{y_i\}_{i=0}^n$ являются значениями некоторой функции f в точках $\{x_i\}_{i=0}^n$ соответственно. Тогда каждый из коэффициентов α_i, определяемых по формулам (2.4) можно рассматривать как выражение, зависящее от функции f и узлов $x_0, \ldots, x_i$. Обозначим это выражение

$$\alpha_i = f[x_0, \ldots, x_i]. \tag{2.7}$$

Лемма 2.1. *Пусть $\{x_0, x_1, \ldots\}$ — последовательность попарно различных чисел. Обозначим P_k и P_k^+ интерполяционные многочлены для функции f по узлам $\{x_0, \ldots, x_k\}$, и $\{x_1, \ldots, x_{k+1}\}$ соответственно. Тогда*

$$P_{k+1}(x) = \frac{(x - x_0)P_k^+(x) - (x - x_{k+1})P_k(x)}{x_{k+1} - x_0}, \quad \forall\, k \geqslant 0. \tag{2.8}$$

$\triangleright_3$ Докажите лемму простой проверкой условий $P_{k+1}(x_j) = f(x_j)$, $j = \overline{0, k+1}$.

Запишем интерполяционные многочлены P_{i-1} и P_{i-1}^+ из леммы 2.1 в форме Ньютона, используя обозначение (2.7):

$$P_{i-1}(x) = \sum_{j=0}^{i-1} \alpha_j \omega_j(x) = \sum_{j=0}^{i-1} f[x_0, \ldots, x_j]\omega_j(x),$$
$$P_{i-1}^+(x) = \sum_{j=0}^{i-1} \alpha_j^+ \omega_j^+(x) = \sum_{j=0}^{i-1} f[x_1, \ldots, x_{j+1}]\omega_j^+(x).$$

Здесь $w_j^+(x) = (x - x_1)(x - x_2)\ldots(x - x_j)$. Тогда (2.8) примет вид

$$P_i(x) = \sum_{j=0}^{i} f[x_0, \ldots, x_j]\omega_j(x) =$$
$$= \frac{(x - x_0)\sum_{j=0}^{i-1} f[x_1, \ldots, x_{j+1}]\omega_j^+(x) - (x - x_i)\sum_{j=0}^{i-1} f[x_0, \ldots, x_j]\omega_j(x)}{x_i - x_0}.$$

Приравнивая коэффициенты при старшей степени в обеих частях и учитывая, что данная формула справедлива для всех $i \geqslant 0$, получаем

Разделённой разностью порядка i для функции f по попарно различным узлам $\{x_j\}_{j=0}^i$ называется выражение $f[x_0, \ldots, x_i]$, определяемое по рекуррентным соотношениям

$$f[x_0, \ldots, x_i] = \frac{f[x_1, \ldots, x_i] - f[x_0, \ldots, x_{i-1}]}{x_i - x_0}, \quad \forall i \geqslant 1; \tag{2.9a}$$
$$f[x_j] = f(x_j) \quad \forall\, j. \tag{2.9b}$$

Таким образом, коэффициенты (2.7) *ИМ в форме Ньютона традиционно вычисляются по формулам* (2.9).

$\triangleright_4$ Вычислите общий вид коэффициентов α_0, α_1 и α_2 сначала по формуле (2.4), затем по формуле (2.9).

$\triangleright_5$ Сравните вычислительную сложность вычисления α_i по формулам (2.4) и (2.9).

$\triangleright_6$ Докажите, что значение РР не зависит от порядка расположения её аргументов.

2.3 Алгоритм вычисления разделённых разностей

Коэффициенты ИМ (2.3) удобно вычислять по определению разделённых разностей (2.9) путём построения треугольной таблицы следующего вида.

$$
\begin{array}{llllll}
x_0, f[x_0] & & & & & \\
 & f[x_0,x_1] & & & & \\
x_1, f[x_1] & & f[x_0,x_1,x_2] & & & \\
 & f[x_1,x_2] & & \ddots & & \\
x_2, f[x_2] & & f[x_0,x_1,x_2] & & f[x_0,\ldots,x_{n-1}] & \\
\vdots & f[x_2,x_3] & \vdots & & & f[x_0,\ldots,x_n] \\
\vdots & \vdots & \vdots & & f[x_1,\ldots,x_n] & \\
\vdots & \vdots & \vdots & \iddots & & \\
x_{n-1}, f[x_{n-1}] & \vdots & f[x_{n-2},x_{n-1},x_n] & & & \\
 & f[x_{n-1},x_n] & & & & \\
x_n, f[x_n] & & & & &
\end{array}
$$

2.4 Остаток интерполирования

Пусть $f \in C[a,b]$ — интерполируемая функция, P_n — интерполяционный многочлен для f по узлам $\{x_i \in [a,b]\}_{i=0}^n$. *Остатком интерполирования* называют функцию

$$r_n = f - P_n. \tag{2.10}$$

Погрешностью интерполирования назовём норму остатка в пространстве $C[a,b]$:

$$\varepsilon_n = \|r_n\|_{C[a,b]} = \max_{x\in[a,b]} |r_n(x)|. \tag{2.11}$$

Теорема 2.1. *Остаток интерполирования имеет вид*

$$r_n(x) = f[x_0,\ldots,x_n,x]\,\omega_{n+1}(x) \quad \forall x \in \mathbb{R}. \tag{2.12}$$

Доказательство. Пусть P_n — ИМ по узлам $x_0,\ldots,x_n$ для f. Если $x \in \{x_i\}_{i=0}^n$, то по определению имеем $r_n(x) = 0$, что (почти) соответствует

(2.12). Пусть $x \notin \{x_i\}_{i=0}^n$. Рассмотрим для f интерполяционный многочлен P_{n+1} по узлам $x_0, \ldots, x_n, x$. По формуле (2.5) имеем

$$P_{n+1}(x) = f(x) = P_n(x) + f[x_0, \ldots, x_n, x]\omega_{n+1}(x). \qquad \blacksquare$$

Теорема 2.2 (Теорема Ролля). *Если функция g непрерывна на $[a, b]$, дифференцируема на (a, b), и $g(a) = g(b)$, то существует по крайней мере одна такая точка $\xi \in (a, b)$, что $g'(\xi) = 0$.*

Лемма 2.2. *Пусть $\underline{x} = \min\{x_i\}_{i=0}^n$, $\overline{x} = \max\{x_i\}_{i=0}^n$ и $f \in C^n[\underline{x}, \overline{x}]$. Тогда $\exists\, \xi \in (\underline{x}, \overline{x})$ такое, что*

$$f[x_0, \ldots, x_n] = \frac{f^{(n)}(\xi)}{n!}. \tag{2.13}$$

Доказательство. Построим для f ИМ P_n в форме Ньютона по узлам $x_0, \ldots, x_n$ и рассмотрим остаток интерполирования $r_n = f - P_n$. По построению имеем

$$r_n(x_0) = r_n(x_1) = \ldots = r_n(x_n) = 0,$$

значит по теореме Ролля между $\underline{x}$ и $\overline{x}$ существует как минимум n точек, в которых r_n' обращается в нуль. Продолжая аналогичные рассуждения для r_n'' и так далее, получаем, что $\exists\, \xi \in (\underline{x}, \overline{x})$ такое, что

$$r_n^{(n)}(\xi) = f^{(n)}(\xi) - P_n^{(n)}(\xi) = f^{(n)}(\xi) - n!\, f[x_0, \ldots, x_n] = 0. \qquad \blacksquare$$

Теорема 2.3. *Пусть $f \in C^{(n+1)}[a, b]$, P_n — ИМ для f по узлам $\{x_i\}_{i=0}^n \subset [a, b]$. Тогда $\forall x \in [a, b]$ $\exists \xi \in (\underline{x}, \overline{x})$, ($\underline{x} = \min\{x_0, \ldots, x_n, x\}$, $\overline{x} = \max\{x_0, \ldots, x_n, x\}$), такое, что*

$$r_n(x) = f(x) - P_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\omega_{n+1}(x). \tag{2.14}$$

Доказательство. Данная теорема является тривиальным следствием теоремы 2.1 и леммы 2.2. $\blacksquare$

Формула (2.14) является основным инструментом при оценке погрешности интерполяции. Из неё, в частности, сразу получаем

$$\varepsilon_n = \|r_n\| \leqslant \frac{\|f^{(n+1)}\|}{(n+1)!}\|\omega_{n+1}\|, \tag{2.15}$$

где $\|\cdot\| = \|\cdot\|_{C[a,b]}$ (см. (2.11)).

3 Интерполяция с кратными узлами

3.1 Постановка задачи

Интерполяция с кратными узлами отличается от обычной интерполяции тем, что в каждом узле x_i требуется совпадение не только значений f и φ, но и значений их первых m_i производных.

Рассмотрим достаточно гладкую функцию f и узлы $\{x_i\}_{i=0}^n$. *Задача интерполяции с кратными узлами* (эрмитовой интерполяции) состоит в нахождении интерполирующей функции φ, удовлетворяющей условиям

$$\varphi^{(j)}(x_i) = f^{(j)}(x_i) =: y_i^{(j)}, \quad i = \overline{0,n}, \quad j = \overline{0, m_i - 1}. \tag{3.1}$$

Натуральное число $m_i \geqslant 1$ называется *кратностью* i-го узла.

Как и ранее, φ будем искать в виде линейной комбинации некоторых базисных функций $\{\varphi_i\}$. Количество условий (3.1) равно

$$N = \sum_{i=0}^{n} m_i,$$

то есть для однозначной разрешимости задачи в общем случае необходимо иметь не менее N базисных функций φ_i:

$$\varphi = \sum_{i=0}^{N-1} \alpha_i \varphi_i, \tag{3.2}$$

где, как обычно, α_i — коэффициенты, подлежащие определению.

Аналогично лекции 1 рассмотрим СЛАУ, полученную подстановкой (3.2) в (3.1), в матричном виде

$$\Phi\alpha = y \quad \Leftrightarrow \quad \begin{bmatrix} \Phi_0 \\ \Phi_1 \\ \vdots \\ \Phi_n \end{bmatrix} \alpha = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}. \tag{3.3}$$

Здесь обобщённая матрица интерполяции Φ и вектор y состоят из $(n+1)$ блоков Φ_i и y_i, каждый их которых соответствует одному узлу интерполяции и имеет вид

$$\Phi_i = \begin{bmatrix} \varphi_0(x_i) & \varphi_1(x_i) & \cdots & \varphi_{N-1}(x_i) \\ \varphi_0'(x_i) & \varphi_1'(x_i) & \cdots & \varphi_{N-1}'(x_i) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_0^{(m_i-1)}(x_i) & \varphi_1^{(m_i-1)}(x_i) & \cdots & \varphi_{N-1}^{(m_i-1)}(x_i) \end{bmatrix}, \quad y_i = \begin{bmatrix} y_i^{(0)} \\ y_i^{(1)} \\ \vdots \\ y_i^{(m_i-1)} \end{bmatrix}. \tag{3.4}$$

3.2 Интерполяционная формула Эрмита

Положив в (3.2) $\varphi_i(x) = x^i$, мы получим многочлен степени $N-1$, который называют *интерполяционным многочленом Эрмита* и обозначают $\varphi = H_{N-1}$.

$\triangleright_1$ Выпишите аналог матрицы Вандермонда для случая кратных узлов и вычислите её определитель.

$\triangleright_2$ Что представляет собой ИМ Эрмита в случае $n = 0$, $m_0 = N$?

Интерполяционная формула Эрмита является обобщением формулы ИМ Лагранжа на случай кратных узлов. Основная задача состоит в построении обобщённого фундаментального базиса многочленов

$$\{\Lambda_{ij}\}, \quad i = \overline{0,n}, \quad j = \overline{0, m_i - 1},$$

удовлетворяющего условиям

$$\Lambda_{ij}^{(q)}(x_p) = \delta_{ip}\delta_{jq} = \begin{cases} 1, & i = p \text{ и } j = q; \\ 0, & \text{иначе}; \end{cases} \qquad \forall\, i, p = \overline{0,n} \quad j, q = \overline{0, m_i - 1}. \tag{3.5}$$

Как только такой базис построен, решение задачи (3.1), очевидно, записывается в виде

$$H_{N-1}(x) = \sum_{i=0}^{n} \sum_{j=0}^{m_i-1} y_i^{(j)} \Lambda_{ij}(x). \tag{3.6}$$

Построение искомого базиса осуществляется в два этапа. Сначала рассмотрим вспомогательные многочлены

$$\lambda_{ij}(x) = \mu_{ij}(x)\nu_i(x), \tag{3.7}$$

где

$$\mu_{ij}(x) = \frac{1}{j!}(x - x_i)^j, \quad \nu_i(x) = \prod_{\substack{k=0 \\ k\neq i}}^{n} \left(\frac{x - x_k}{x_i - x_k}\right)^{m_k}. \tag{3.8}$$

Очевидны следующие соотношения:

$$\mu_{ij}^{(q)}(x_i) = \delta_{jq} \quad \forall i = \overline{0,n}, \quad \forall j, q; \tag{3.9a}$$

$$\nu_i^{(q)}(x_p) = 0 \quad \forall p \neq i, \quad q = \overline{0, m_p - 1}. \tag{3.9b}$$

Для того, чтобы увидеть структуру $\lambda_{ij}^{(q)}$ воспользуемся формулой Лейбница

$$(fg)^{(q)} = \sum_{r=0}^{q} \frac{q!}{r!(q-r)!} f^{(r)} g^{(q-r)}.$$

Получаем

$$\lambda_{ij}^{(q)} = \sum_{r=0}^{q} \frac{q!}{r!(q-r)!} \mu_{ij}^{(r)} \nu_i^{(q-r)}.$$

Из последней формулы и (3.9) автоматически получаем следующие свойства многочленов λ_{ij}.

1. При $p \neq i$ имеем $\lambda_{ij}^{(q)}(x_p) = 0 \ \ \forall j, q = \overline{0, m_i - 1}$.
2. При $p = i$ имеем

$$\lambda_{ij}^{(q)}(x_i) = \begin{cases} 0, & q = \overline{0, j-1}; \\ 1, & q = j; \\ \chi_{ijq}, & j+1 \leqslant q \leqslant m_i - 1. \end{cases} \tag{3.10}$$

Здесь

$$\chi_{ijq} = \frac{q!}{j!(q-j)!} \nu_i^{(q-j)}(x_i).$$

Таким образом, λ_{ij} удовлетворяют почти всем условиям (3.5), которым должны удовлетворять Λ_{ij}. Более того,

$$\Lambda_{i,m_i-1} = \lambda_{i,m_i-1}, \quad \forall i = \overline{0, n}. \tag{3.11a}$$

Картину портит лишь наличие ненулевых χ_{ijq} в формуле (3.10).

Поэтому на втором этапе для каждого $i = \overline{0, n}$ нужно сгенерировать неизвестные Λ_{ij} по рекуррентным соотношениям

$$\Lambda_{ij} = \lambda_{ij} - \sum_{q=j+1}^{m_i-1} \chi_{ijq} \Lambda_{iq}, \quad j = m_i - 2, \ldots, 0. \tag{3.11b}$$

$\triangleright_3$ Обоснуйте формулу (3.11b).

Таким образом, построение ИМ в форме Эрмита осуществляется по следующей схеме.

1. Построение вспомогательных многочленов λ_{ij} по формуле (3.7).
2. Построение Λ_{ij} по рекуррентным формулам (3.11).
3. Запись ИМ Эрмита по формуле (3.6).

$\triangleright_4$ Постройте базис Λ_{ij} для случая $m_i = 2 \ \ \forall i = \overline{0, n}$.

$\triangleright_5$ Докажите единственность ИМ Эрмита.

3.3 ИМ Эрмита в форме Ньютона

3.3.1 Построение базиса

В общем случае использовать формулу Эрмита (3.6) могут лишь сильные духом. Гораздо более гуманный способ решения задачи полиномиальной интерполяции с кратными узлами состоит в построении ИМ в форме Ньютона, аналогично (2.3).

Прежде всего строим базис, в котором матрица Φ из (3.3) имеет нижнетреугольный вид. С учётом вида составляющих эту матрицу блоков (3.4), для $i = 0$ получаем условия

$$\varphi_j^{(k)}(x_0) = 0 \quad \forall\, j > k,$$

а для произвольного i должно выполняться

$$\varphi_j^{(k)}(x_i) = 0 \quad \forall\, j > k + \sum_{q=0}^{i-1} m_q.$$

Из этих условий с точностью до константы имеем

$$\begin{aligned}
&\varphi_0(x) = 1,\\
&\varphi_1(x) = (x - x_0),\\
&\dots\\
&\varphi_{m_0}(x) = (x - x_0)^{m_0};\\
&\varphi_{m_0+1}(x) = (x - x_0)^{m_0}(x - x_1),\\
&\dots\\
&\varphi_{m_0+m_1}(x) = (x - x_0)^{m_0}(x - x_1)^{m_1}\\
&\dots\\
&\dots\\
&\varphi_{N-1}(x) = (x - x_0)^{m_0}(x - x_1)^{m_1} \dots (x - x_n)^{m_n-1}.
\end{aligned}$$

Эти формулы можно записать проще, рассмотрев новую последовательность узлов $\{\xi_i\}_{i=0}^{N-1}$, полученную из $\{x_i\}_{i=0}^{n}$ повторением i-го узла m_i раз:

$$\Xi = \{\xi_i\}_{i=0}^{N-1} = \{\underbrace{x_0, \dots, x_0}_{m_0}, \underbrace{x_1, \dots, x_1}_{m_1}, \dots, \underbrace{x_n, \dots, x_n}_{m_n}\}. \tag{3.12}$$

Множество (3.12) будем называть *расширенным множеством узлов интерполяции*. С помощью него можно переформулировать постановку задачи эрмитовой интерполяции (3.1) в эквивалентной форме: для данной функции f построить многочлен H_{N-1}, удовлетворяющий условиям

$$H_{N-1}^{(j)}(\xi) = f^{(j)}(\xi), \quad \forall \xi \in \widetilde{\Xi}, \quad \forall j = \overline{0, m(\xi) - 1},$$

где $\widetilde{\Xi}$ — множество, полученное из Ξ удалением всех повторяющихся элементов, а $m(\xi)$ — количество элементов, равных ξ, в множестве Ξ.

Теперь можно аналогично (2.2) записать

$$\varphi_i(x) = \Omega_i(x) = \prod_{j=0}^{i-1}(x - \xi_j). \tag{3.13}$$

Итак, по построению в базисе (3.13) матрица Φ из (3.3) будет иметь нижнетреугольный вид. Осталось вывести формулы для выражения α_i, которые (пока что формально) снова обозначим

$$\alpha_i = f[\xi_0, \ldots, \xi_i]. \tag{3.14}$$

Основная проблема с этим обозначением состоит в том, что среди $\{\xi_i\}$ могут быть равные, а мы определяли разделённые разности только для попарно различных узлов.

3.3.2 Разделённые разности с кратными узлами

Прежде всего легко убедиться, что если все узлы $\xi_0, \ldots, \xi_i$ в (3.14) различны, то получается обычная разделённая разность (2.9).

Теперь рассмотрим противоположный случай: $\xi_0 = \xi_1 = \ldots = \xi_i = x_0$. В этом случае легко вычислить α_i как решение СЛАУ (3.3). Рассмотрим блок (3.4) для $i = 0$ (первые m_0 уравнений системы). В базисе (3.13) эти уравнения будет иметь вид

$$\left[\begin{array}{cccccccc|c} 1 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & y_0^{(0)} \\ 0 & 1 & 0 & \cdots & 0 & 0 & \cdots & 0 & y_0^{(1)} \\ 0 & 0 & 2! & \cdots & 0 & 0 & \cdots & 0 & y_0^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & (m_0-1)! & 0 & \cdots & 0 & y_0^{(m_0-1)} \end{array}\right].$$

Отсюда сразу получаем

$$\alpha_i = f[\underbrace{\xi_0, \ldots, \xi_0}_{i+1 \text{ раз}}] = y_0^{(i)}/(i!) = f^{(i)}(\xi_0)/(i!).$$

Осталось рассмотреть смешанный случай. Пусть среди $\{\xi_j\}_{j=0}^{i}$ есть хотя бы два различных: $\xi_p \neq \xi_q$. В этом случае нужно использовать два факта. Во-первых, совершенно аналогично лекции 2 имеем, что значение α_i не зависит от порядка расположения аргументов $\xi_0, \ldots, \xi_i$. Во-вторых, понадобится следующее обобщение леммы 2.1

Лемма 3.1. *Пусть H_{N-1} — ИМ Эрмита, построенный по расширенному множеству узлов Ξ (3.12), а H_{N-2} и $\tilde{H}_{N-2}$ — ИМ Эрмита, построенные соответственно по множествам $\Xi \setminus \xi_p$ и $\Xi \setminus \xi_q$, причём $\xi_p \neq \xi_q$. Тогда*

$$H_{N-1}(x) = \frac{(x - \xi_q)H_{N-2} - (x - \xi_p)\tilde{H}_{N-2}}{\xi_p - \xi_q}.$$

$\triangleright_6$ Докажите лемму.

Приравнивая старшие коэффициенты в обеих частях последней формулы (как в лекции 2), получаем следующее общее определение.

Рассмотрим достаточно гладкую функцию f и набор вещественных чисел $\Xi = \{\xi_j\}_{j=0}^{i}$, среди которых возможны совпадения. Разделённой разностью называют выражение

$$f[\xi_0, \ldots, \xi_i] = f[\Xi] = \begin{cases} \dfrac{f^{(i)}(\xi_0)}{i!}, & \text{если} \quad \xi_0 = \xi_1 = \ldots = \xi_i; \\ \dfrac{f[\Xi \setminus \xi_q] - f[\Xi \setminus \xi_p]}{\xi_p - \xi_q}, & \text{если} \quad \exists\, \xi_p \neq \xi_q. \end{cases} \tag{3.15}$$

3.3.3 Алгоритм построения ИМ Эрмита в форме Ньютона

Собирая всё вышеизложенное вместе, получаем, что

$$H_{N-1}(x) = \sum_{i=0}^{N-1} f[\xi_0, \ldots, \xi_i]\, \Omega_i(x), \tag{3.16}$$

где ξ_j — элементы расширенного множества узлов (3.12), $\Omega_i(x) = \prod_{j=0}^{i-1}(x - \xi_j)$, $f[\ldots]$ — разделённые разности с кратными узлами (3.15).

Разделённые разности вычисляются аналогично случаю однократных узлов путём построения треугольной таблицы. В первом столбце выписываются точки ξ_i и соответствующие значения $y_i^{(0)}$, после чего вычисления идут по определению (3.15). Приведём пример такой таблицы для случая $n = 2$, $m_0 = 2$, $m_1 = 3$, $m_2 = 1$.

$x_0, f(x_0)$					
	$f'(x_0)$				
$x_0, f(x_0)$		$f[x_0, x_0, x_1]$			
	$f[x_0, x_1]$		$f[x_0, x_0, x_1, x_1]$		
$x_1, f(x_1)$		$f[x_0, x_1, x_1]$		$f[x_0, x_0, x_1, x_1, x_1]$	
	$f'(x_1)$		$f[x_0, x_1, x_1, x_1]$		$f[x_0, x_0, x_1, x_1, x_1, x_2]$
$x_1, f(x_1)$		$f''(x_1)/2!$		$f[x_0, x_1, x_1, x_1, x_2]$	
	$f'(x_1)$		$f[x_1, x_1, x_1, x_2]$		
$x_1, f(x_1)$		$f[x_1, x_1, x_2]$			
	$f[x_1, x_2]$				
$x_2, f(x_2)$					

$\triangleright_7$ Обдумайте способ оптимального с точки зрения экономии памяти вычисления всех РР из (3.16).

3.4 Остаток интерполирования с кратными узлами

Теорема 3.1. *Пусть f — интерполируемая функция, H_{N-1} — ИМ Эрмита по расширенному множеству узлов* (3.12). *Тогда*

$$r_{N-1}(x) = f(x) - H_{N-1}(x) = f[\xi_0, \ldots, \xi_{N-1}, x]\Omega_N(x), \tag{3.17}$$

где $\Omega_N(x)$ вычисляется по (3.13).

Доказательство проводится аналогично теореме 2.1.

Теорема 3.2. *Пусть $f \in C^{(N)}[a,b]$, интерполируемая функция, H_{N-1} — ИМ Эрмита по множеству узлов $\{x_i\}_{i=0}^n$ с кратностями m_i, $\sum_{i=0}^n m_i = N$. Тогда $\forall x \in [a,b]$ $\exists \xi \in (\underline{x}, \overline{x})$, ($\underline{x} = \min\{x_0, \ldots, x_n, x\}$, $\overline{x} = \max\{x_0, \ldots, x_n, x\}$), такое, что*

$$r_{N-1}(x) = f(x) - H_{N-1}(x) = \frac{f^{(N)}(\xi)}{N!}\Omega_N(x). \tag{3.18}$$

4 Многочлен наилучшего равномерного приближения

4.1 Постановка задачи

Введём следующие обозначения. Множество многочленов степени не выше n обозначим $\mathbb{P}_n$, а множество всех многочленов — $\mathbb{P} = \mathbb{P}_\infty$.

Из анализа известна

Теорема 4.1 (Вейерштрасса). *Любая непрерывная на* $[a, b]$ *функция* f *может быть со сколь угодно высокой точностью приближена алгебраическим многочленом:*

$$\forall \varepsilon > 0 \quad \exists\, p \in \mathbb{P} \;:\; \|f - p\| < \varepsilon.$$

Очевидно, что чем меньше ε, тем выше должна быть степень p в общем случае. На практике же мы всегда ограничены какой-то фиксированной степенью n.

Поэтому важно уметь решать следующую задачу: для данной $f \in C[a, b]$ найти такой $p_n^* \in \mathbb{P}_n$, что

$$\|f - p_n^*\| = \inf_{p \in \mathbb{P}_n} \|f - p\| = E_n. \tag{4.1}$$

Здесь и далее $\|\cdot\| = \|\cdot\|_{C[a,b]}$.

Многочлен p_n^*, удовлетворяющий (4.1), называется *многочленом наилучшего равномерного приближения* (МНРП) степени n для функции f на отрезке $[a, b]$.

Важно, что такой многочлен всегда существует и единственен (без доказательства).

4.2 Теорема Чебышева

Пусть $f, g \in C[a, b]$ и $\|f - g\| = E$. *Альтернансом порядка* m для этих функций называется множество точек $\{\xi_i\}_{i=1}^m$,

$$a \leqslant \xi_1 < \xi_2 < \ldots < \xi_{m-1} < \xi_m \leqslant b,$$

в которых разность $f - g$ достигает максимальных по модулю значений с попарно чередующимися знаками:

$$f(\xi_i) - g(\xi_i) = \sigma(-1)^i E, \quad i = \overline{1, m},$$

где $\sigma = 1$ или -1 одновременно для всех i.

Теорема 4.2 (Чебышева, критерий МНРП). *Для того, чтобы* $p \in \mathbb{P}_n$ *был МНРП для* $f \in C[a,b]$ *необходимо и достаточно, чтобы* f *и* p *имели альтернанс порядка* $n+2$ *на* $[a,b]$.

Доказательство. см. [Бахвалов, Жидков, Кобельков, «Численные методы», 2004. Стр. 179]. ■

$\triangleright_1$ Является ли МНРП интерполяционным?

В общем случае задача построения МНРП весьма сложна. Существующие численные алгоритмы её решения, основанные на теореме Чебышева, не очень просты и редко применяются на практике. Зато с помощью теории МНРП можно красиво решить следующую не менее важную проблему.

4.3 Минимизация остатка интерполирования

Пусть P_n — интерполяционный многочлен для f по узлам $\{x_i\}_{i=0}^n$. Согласно теореме 2.3 точность приближения оценивается по формуле

$$\|f - P_n\| \leqslant \frac{\|f^{(n+1)}\|}{(n+1)!}\|\omega_{n+1}\|, \tag{4.2}$$

где

$$\omega_{n+1}(x) = (x - x_0)(x - x_1)\dots(x - x_n). \tag{4.3}$$

Наша задача — сделать правую часть (4.2) как можно меньше. И если в общем случае о величине $\|f^{(n+1)}\|$ ничего не известно, то $\|\omega_{n+1}\|$ можно минимизировать за счёт выбора узлов $\{x_i\}$. Задачу выбора оптимальных узлов интерполяции можно поставить так: найти многочлен ω_{n+1} такой, что

$$\|\omega_{n+1}\| = \inf_{p \in \mathbb{P}^*_{n+1}} \|p\|, \tag{4.4}$$

где $\mathbb{P}^*_{n+1}$ — множество многочленов степени не выше $n+1$ со *старшим коэффициентом, равным единице*. Решить эту задачу можно с помощью теоремы Чебышева.

Действительно, если ω_{n+1} — решение задачи (4.4), то, представляя его как

$$\omega_{n+1}(x) = x^{n+1} + a_n x^n + \dots + a_1 x + a_0 = x^{n+1} - q_n(x),$$

получаем, что многочлен q_n — МНРП степени n к функции $g_{n+1}(x) = x^{n+1}$.

Пусть $\|g_{n+1} - q_n\| = \|\omega_{n+1}\| = E$. По теореме Чебышева на отрезке $[-1,1]$ существует альтернанс порядка $n+2$:

$$-1 \leqslant \xi_1 < \xi_2 < \dots < \xi_{n+1} < \xi_{n+2} \leqslant 1.$$

По определению имеем $|\omega_{n+1}(\xi_i)| = E$, причем знаки $\omega_{n+1}(\xi_i)$ попарно чередуются.

Прежде всего покажем, что концы отрезка входят в альтернанс: $\xi_1 = -1$, $\xi_{n+2} = 1$. Предположим обратное. Тогда получается, что

$$\omega'_{n+1}(\xi_i) = 0 \quad \forall\, i = \overline{1, n+2}$$

и многочлен

$$Q_{2n+2} = E^2 - \omega_{n+1}^2$$

имеет двукратный корень в каждой точке альтернанса (суммарная кратность — $2n+4$), что невозможно.

Значит, концы отрезка $[-1, 1]$ должны входить в альтернанс: в этом случае $\omega'(\xi_i) = 0$ лишь для $i = \overline{2, n+1}$, а многочлен Q_{2n+2} — двукратные корни в этих точках и однократные в $\xi_1 = -1$ и $\xi_{n+2} = 1$. Следовательно, мы можем установить следующую связь между многочленами ω'_{n+1} и Q_{2n+2}:

$$Q_{2n+2}(x) = C(1 - x^2)\,\omega'_{n+1}(x)^2. \tag{4.5}$$

Приравнивая старшие коэффициенты в обеих частях, получаем $C = (n+1)^{-2}$. Обозначив $\omega_{n+1} = y$, из (4.5) получаем дифференциальное уравнение с разделяющимися переменными:

$$\frac{y'(x)}{\sqrt{E - y^2(x)}} = \frac{n+1}{\sqrt{1 - x^2}}. \tag{4.6}$$

Проинтегрировав его, получаем

$$\arccos\frac{y(x)}{E} = (n+1)\arccos x + C_0$$

или

$$y(x) = \omega_{n+1}(x) = E\cos((n+1)\arccos x + C_0).$$

Из условия $\omega_{n+1}(1) = E$ получаем $C_0 = 0$, поэтому окончательно имеем

$$\omega_{n+1}(x) = E\cos((n+1)\arccos x) = E\,T_{n+1}(x), \tag{4.7}$$

где T_{n+1} — *многочлен Чебышева*.

4.4 Многочлены Чебышева

Рассмотрим подробнее этот замечательный объект — многочлены Чебышева

$$T_n(x) = \cos(n\arccos x), \quad n = 0, 1, 2, \ldots \quad . \tag{4.8}$$

Начнём с того, что это действительно многочлены:

$$T_0(x) = 1, \quad T_1(x) = x,$$

а для произвольного n справедливо рекуррентное соотношение

$$T_{n+1}(x) = 2x\,T_n(x) - T_{n-1}(x). \tag{4.9}$$

$\triangleright_2$ Докажите это соотношение, используя тригонометрическую формулу $\cos\alpha\cos\beta = \frac{1}{2}(\cos(\alpha+\beta)+$
$\triangleright_3$ Вычислите $T_i(x)$ для $i = 2, 3, 4$.

4.5 Оптимальные узлы интерполирования

Из (4.9) имеем, что для $n \geqslant 1$ старший коэффициент T_n равен 2^{n-1}. Следовательно, в формуле (4.7) неизвестная константа E равна 2^{1-n}, и искомый многочлен ω_{n+1} имеет вид

$$\omega_{n+1}(x) = 2^{-n}T_{n+1}(x). \tag{4.10}$$

Такой многочлен, в силу понятных причин, называют *многочленом, наименее отклоняющимся от нуля на отрезке* $[-1, 1]$. Его корни по построению являются оптимальными узлами интерполяции. Они равны корням многочлена Чебышева T_{n+1}:

$$x_i = \cos\frac{\pi(2i+1)}{2n+2}, \quad i = \overline{0, n}. \tag{4.11}$$

Узлы (4.11) называются *чебышёвскими узлами*.

$\triangleright_4$ Как надо изменить формулу (4.11), чтобы узлы интерполяции расположились в порядке возрастания?

$\triangleright_5$ Используя формулу ИМ Лагранжа и (4.10) постройте формулу ИМ по чебышевским узлам на отрезке $[-1, 1]$.

Для полного решения задачи минимизации остатка интерполирования необходимо обобщить формулы (4.10) и (4.11) на случай произвольного отрезка $[a, b]$. Для этого рассмотрим замену переменных

$$x = \frac{a+b}{2} + \frac{b-a}{2}t, \quad t \in [-1, 1],\ x \in [a, b].$$

Тогда многочен Чебышева, смасштабированный на $[a, b]$, будет иметь вид

$$\widehat{T}_{n+1}(x) = T_{n+1}(t) = 2t\,T_n(t) - T_{n-1}(t) = 2t\,\widehat{T}_n(x) - \widehat{T}_{n-1}(x) = \\ = 2\frac{2x-a-b}{b-a}\widehat{T}_n(x) - \widehat{T}_{n-1}(x). \tag{4.12}$$

Кроме этого имеют место очевидные соотношения

$$\widehat{T}_0(x) = 1, \quad \widehat{T}_1(x) = \frac{2x-b-a}{b-a}.$$

Следовательно, согласно (4.12) старший коэффициент многочлена $\widehat{T}_n(x)$ при всех $n \geqslant 1$ равен

$$\hat{a}_n = \frac{2}{b-a}\left(\frac{4}{b-a}\right)^{n-1} = \frac{1}{2}\left(\frac{4}{b-a}\right)^{n},$$

откуда получаем

$$\omega_{n+1}(x) = 2\left(\frac{b-a}{4}\right)^{n+1} T_{n+1}\left(\frac{2x-b-a}{b-a}\right). \tag{4.13}$$

Корни этого многочлена (*оптимальные узлы интерполяции на отрезке* $[a, b]$), очевидно, получаются масштабированием узлов (4.11):

$$x_i = \frac{a+b}{2} + \frac{b-a}{2}\cos\frac{\pi(2i+1)}{2n+2}, \quad i = \overline{0, n}. \tag{4.14}$$

Кроме этого, из (4.13) следует, что при выборе чебышевских узлов интерполирования (4.14) имеем

$$\|\omega_{n+1}\| = 2\left(\frac{b-a}{4}\right)^{n+1}.$$

Это равенство используется при оценке погрешности интерполирования по формуле (4.2).

4.6 Сходимость полиномиальной интерполяции

В дальнейшем набор узлов интерполирования будем называть *сеткой*. На отрезке $[a, b]$ рассмотрим бесконечную последовательность сеток $\{X_n\}_{n=0}^{\infty}$ из $(n+1)$ попарно различных узлов:

$$\begin{aligned} X_0 &= \{x_0^{(0)}\}, \\ X_1 &= \{x_0^{(1)}, x_1^{(1)}\}, \\ &\dots \\ X_n &= \{x_0^{(n)}, x_1^{(n)}, \dots, x_n^{(n)}\}, \\ &\dots \end{aligned} \tag{4.15}$$

Для данной функции f эти сетки порождают последовательность интерполяционных многочленов $\{P_n\}_{n=0}^{\infty}$: каждый P_n интерполирует f по узлам X_n. Процесс построения последовательности $\{P_n\}$ называется интерполяционным процессом. Если эта последовательность сходится, то говорят, что интерполяционный процесс сходится.

Сходимость интерполяционного процесса — один из наиболее важных вопросов теории приближения функций. Мы будем говорить в основном о равномерной сходимости (т. е. о сходимости в норме $C[a, b]$):

$$\|f - P_n\| \xrightarrow[n\to\infty]{?} 0.$$

Ниже мы рассмотрим без доказательства ряд основных результатов по этому вопросу. Прежде всего, из теоремы Вейерштрасса и теоремы Чебышева следует

Теорема 4.3. *Для любой $f \in C[a, b]$ существует такая последовательность сеток, для которой интерполяционный процесс равномерно сходится к f.*

Эта теорема носит формальный характер, так как построение нужной последовательности сеток в общем случае представляет серьёзную проблему. Следующий результат говорит о том, что не существует «универсальной» последовательнсти сеток, хорошей для всех непрерывных функций.

Теорема 4.4. *Для любой последовательности сеток вида* (4.15) *существует такая $f \in C[a, b]$, для которой интерполяционный процесс* не сходится *равномерно к f.*

Таким образом, класс $C[a, b]$ слишком широк. Поэтому рассмотрим результаты о сходимости интерполяционного процесса для более узкого класса функций.

Функция f называется целой, если существует её разложение в степенной ряд вида

$$f(x) = \sum_{i=0}^{n} c_i (x - x_0)^i,$$

которое сходится при любом x.

Теорема 4.5. *Если функция f целая, то интерполяционный процесс по любой последовательности сеток вида* (4.15) *равномерно сходится на* $[a, b]$ *к* f.

Функция f называется *абсолютно непрерывной* на отрезке $[a, b]$, если для любого $\varepsilon > 0$ $\exists\, \delta > 0$ такое, что для любого конечного набора попарно непересекающихся интервалов

$$[a_k, b_k] \subset [a, b]$$

выполняется условие

$$\sum_k |a_k - b_k| < \delta \quad \Rightarrow \quad \sum_k |f(a_k) - f(b_k)| < \varepsilon.$$

Достаточными условиями абсолютной непрерывности являются (по отдельности) липшицевость и существование ограниченной производной.

Теорема 4.6. *Для любой абсолютно непрерывной на* $[a, b]$ *функции* f *интерполяционный процесс по чебышевским узлам равномерно сходится.*

Что касается равномерной сетки, то единственным классом, для которых соответствующий интерполяционный процесс будет всегда сходиться, является класс целых функций (см. теорему 4.5). Самые известные примеры функций, для которых интерполяция по равноотстоящим узлам расходится это функция Рунге $f(x) = \frac{1}{1+25x^2}$ и функция $g(x) = |x|$, обе на отрезке $[-1, 1]$. Для последней ИМ по равноотстоящим узлам степени $2n$ неограниченно растёт в любой части отрезка $[-1, 1]$.

5 Тригонометрическая интерполяция

5.1 Дискретное преобразование Фурье

Пусть $\{\sigma_k\}$ — корни степени n из единицы:

$$\sigma_k = \exp\left(\mathrm{i}\frac{2\pi}{n}k\right), \quad k \in \mathbb{Z}. \tag{5.1}$$

Рассмотрим задачу интерполирования произвольной функции $f : \mathbb{C} \to \mathbb{C}$ по её значениям в $\{\sigma_k\}$: требуется найти многочлен степени $n-1$ над полем $\mathbb{C}$:

$$p_{n-1}(z) = p(z) = \sum_{k=0}^{n-1} \alpha_k z^k, \quad z \in \mathbb{C},\ \alpha_k \in \mathbb{C}, \tag{5.2}$$

такой, что

$$p(\sigma_k) = f(\sigma_k) = y_k \in \mathbb{C}, \quad k = \overline{0, n-1}.$$

Подставляя узлы (5.1) в (5.2), как и в вещественном случае получаем СЛАУ

$$p(\sigma_k) = \sum_{j=0}^{n-1} \alpha_j \sigma_k^j = y_k, \quad k = \overline{0, n-1},$$

или

$$V\alpha = y, \tag{5.3}$$

где $V = (v_{kj})$ — матрица Вандермонда: $v_{kj} = \sigma_k^j = \sigma_{kj}$; $\alpha = (\alpha_0, \ldots, \alpha_{n-1})^T$, $y = (y_0, \ldots, y_{n-1})^T$. Вычислим элементы матрицы $W = V^*V$:

$$w_{kj} = \sum_{l=0}^{n-1} \overline{v}_{lk} v_{lj} = \sum_{l=0}^{n-1} \sigma_{-lk}\sigma_{lj} = \sum_{l=0}^{n-1} \sigma_{l(j-k)} = \sum_{l=0}^{n-1} \sigma_{j-k}^l = \begin{cases} n, & j = k, \\ \dfrac{1-\sigma_{j-k}^n}{1-\sigma_{j-k}} = 0, & j \neq k. \end{cases}$$

Отсюда $V^*V = nI$, или

$$V^{-1} = \frac{1}{n}V^*.$$

Используя это соотношение в (5.3), получаем

$$\alpha = \frac{1}{n}V^*y,$$

то есть

$$\alpha_k = \frac{1}{n}\sum_{j=0}^{n-1} y_j \exp\left(-\mathrm{i}\frac{2\pi}{n}kj\right), \quad k = \overline{0, n-1}. \tag{5.4a}$$

По построению имеем $p(\sigma_k) = y_k$, или

$$y_k = \sum_{j=0}^{n-1} \alpha_j \exp\left(\mathrm{i}\frac{2\pi}{n} kj\right), \quad k = \overline{0, n-1}. \tag{5.4b}$$

Преобразование $\{y_k\} \mapsto \{\alpha_k\}$, определяемое формулой (5.4a) называется *дискретным преобразованием Фурье* (ДПФ). *Обратное дискретное преобразование Фурье* задаётся формулой (5.4b).

$\triangleright_1$ Как нужно изменить представление многочлена p в (5.2), чтобы матрица V из (5.3) была унитарной? Как в этом случае изменятся формулы (5.4)?

Дискретное преобразование Фурье является важным инструментом во многих приложениях. В частности, оно используется при сжатии информации с потерями (форматы JPEG и MP3). Вычисление ДПФ по построенным формулам, очевидно, требует $O(n^2)$ операций. Существует также знаменитый алгоритм т. н. быстрого преобразования Фурье (алгоритм Кули–Тьюки), позволяющий вычислить ДПФ за $O(n \ln n)$ операций.

Важно, что формулы (5.4) будут справедливы также и в случае, когда функция f вместо многочлена вида (5.2) интерполируется произвольным разложением по последовательным целым степеням z:

$$p(z) = \sum_{k=N_0}^{N_0+n-1} \alpha_k z^k, \quad N_0 \in \mathbb{Z}. \tag{5.5}$$

В этом случае произойдёт лишь соответствующий сдвиг в индексации:

$$\alpha_k = \frac{1}{n} \sum_{j=N_0}^{N_0+n-1} y_j \exp\left(-\mathrm{i}\frac{2\pi}{n} kj\right), \quad k = \overline{N_0, N_0+n-1}; \tag{5.6a}$$

$$y_k = \sum_{j=N_0}^{N_0+n-1} \alpha_j \exp\left(\mathrm{i}\frac{2\pi}{n} kj\right), \quad k = \overline{N_0, N_0+n-1}. \tag{5.6b}$$

$\triangleright_2$ Докажите это.

5.2 Тригонометрическая интерполяция по равноотстоящим узлам

Положим в формуле (5.5) $n = 2N+1$, $N_0 = -N$:

$$p(z) = \sum_{k=-N}^{N} \alpha_k z^k.$$

Рассмотрим ограничение функции p на единичную окружность $S_1 \subset \mathbb{C}$,

$$S_1 = \{z \in \mathbb{C} \,:\, |z| = 1\}.$$

Для всех $z \in S_1$ имеем $z = \mathrm{e}^{\mathrm{i}\theta}$, $\theta = \operatorname{Arg} z$, и

$$p(z) = p(\mathrm{e}^{\mathrm{i}\theta}) = \sum_{k=-N}^{N} \alpha_k \mathrm{e}^{\mathrm{i}k\theta} = \alpha_0 + \sum_{k=1}^{N} \Big((\alpha_k + \alpha_{-k}) \cos k\theta + \mathrm{i}(\alpha_k - \alpha_{-k}) \sin k\theta \Big),$$

или

$$p(\mathrm{e}^{\mathrm{i}\theta}) = P(\theta) = \alpha_0 + \sum_{k=1}^{N} (a_k \cos k\theta + b_k \sin k\theta), \quad \theta \in \mathbb{R}, \tag{5.7}$$

где

$$a_k = \alpha_k + \alpha_{-k}, \quad b_k = \mathrm{i}(\alpha_k - \alpha_{-k}).$$

Таким образом из комплексной функции p мы получили *тригонометрический многочлен* P — 2π-периодическую функцию вещественной переменной θ. Покажем, что если все y_k вещественны, то $P(\theta) \in \mathbb{R} \;\; \forall\, \theta \in \mathbb{R}$. Согласно (5.6a) имеем

$$\alpha_k = \frac{1}{n} \sum_{j=-N}^{N} y_j \exp\left(-\mathrm{i} \frac{2\pi}{n} kj \right), \quad k = \overline{-N, N},$$

откуда легко видеть, что $\alpha_{-k} = \overline{\alpha}_k \;\; \forall\, k = \overline{1, N}$. Следовательно, имеем

$$a_k = \alpha_k + \overline{\alpha}_k = 2 \operatorname{Re} \alpha_k \in \mathbb{R}, \quad \text{то есть}$$

$$a_k = \frac{2}{2N+1} \sum_{j=-N}^{N} y_j \cos \frac{2\pi}{2N+1} kj. \tag{5.8a}$$

Аналогично получаем

$$b_k = \frac{2}{2N+1} \sum_{j=-N}^{N} y_j \sin \frac{2\pi}{2N+1} kj, \tag{5.8b}$$

ну и

$$\alpha_0 = \frac{1}{2N+1} \sum_{j=-N}^{N} y_j = \frac{a_0}{2}. \tag{5.8c}$$

Таким образом, если коэффициенты тригонометрического многочлена (5.7) вычислены по формулам (5.8), то он принимает вещественные значения при всех θ и по построению удовлетворяет следующим условиям интерполяции:

$$P(\theta_k + 2\pi m) = y_k, \quad \forall k = \overline{-N, N}, \quad m \in \mathbb{Z},$$

где

$$\theta_k = \frac{2\pi}{2N+1}k. \tag{5.9}$$

Такой интерполяционный тригонометрический многочлен можно также записать в виде

$$P(\theta) = \sum_{k=-N}^{N} y_k \Psi_k(\theta).$$

$\triangleright_3$ Найдите фундаментальный базис $\{\Psi_k\}$.

$\triangleright_4$ Постройте формулу тригонометрической интерполяции для случая $n = 2N$.

5.3 Масштабирование

Для того, чтобы построенные формулы тригонометрической интерполяции можно было применять для интерполяции функций по равноотстоящим узлам на произвольном отрезке $[a, b]$, ограничимся главным значением аргумента: $\theta \in [-\pi, \pi]$. На этом отрезке находится $2N+1$ точек интерполяции (5.9), расположенных на расстоянии $\frac{2\pi}{2N+1}$ друг от друга. При этом сетка узлов не содержит концы отрезка.

Поэтому для того, чтобы проинтерполировать f на отрезке $[a, b]$ по равномерной сетке

$$\left\{ x_k = \frac{a+b}{2} + \frac{b-a}{2N}k \right\}_{k=-N}^{N},$$

необходимо сделать замену переменных $x = x(\theta)$, которая осуществляет преобразование

$$[-\theta_N, \theta_N] \mapsto [a, b].$$

$\triangleright_5$ Найдите такую замену и выпишите формулы, обобщающие (5.7), (5.8) на случай отрезка $[a, b]$.

5.4 Интерполяция косинусами

Пусть $\theta \in [-\pi, \pi]$. Сетка узлов $\{\theta_k\}_{k=-N}^{N}$, очевидно, симметрична относительно нуля: $\theta_k = -\theta_{-k}$, $k = \overline{-N, N}$. Поэтому для чётных функций f мы будем иметь $y_{-k} = y_k$ и как следствие в формуле (5.8b) получим $b_k = 0$ $\forall k = \overline{-N, N}$. Аналогично для (5.8a) имеем

$$a_k = \frac{2}{2N+1}\left(y_0 + 2\sum_{j=1}^{N} y_j \cos\frac{2\pi}{2N+1}kj \right). \tag{5.10}$$

Значит, для любых $\{y_k\}_{k=0}^N$, тригонометрический многочлен

$$P(\theta) = \frac{a_0}{2} + \sum_{k=1}^{N} a_k \cos k\theta, \tag{5.11}$$

где коэффициенты a_k вычисляются по (5.10), обладает следующими интерполяционными свойствами на отрезке $[0, \pi]$:

$$P(\theta_k) = y_k, \quad k = \overline{0, N},$$

где θ_k определяются формулой (5.9).

$\triangleright_6$ Постройте аналогичные формулы интерполяции синусами и укажите на их основной недостаток.

В случае произвольного отрезка $[a, b]$, аналогично вышесказанному, нужно сделать замену переменных

$$[0, \theta_N] \mapsto [a, b].$$

5.5 Связь полиномиальной и тригонометрической интерполяции

Рассмотрим многочлен (5.11) при $\theta \in [0, \pi]$. Функция arccos непрерывно и взаимнооднозначно отображает отрезок $[-1, 1]$ на $[0, \pi]$, поэтому мы имеем право сделать замену переменных

$$\theta = \arccos t, \quad t \in [-1, 1].$$

Получим

$$P(\theta) = Q(t) = \frac{a_0}{2} + \sum_{k=1}^{N} a_k \cos(k \arccos t) = \frac{a_0}{2} + \sum_{k=1}^{N} a_k T_k(t),$$

где T_k — многочлены Чебышева, то есть Q — алгебраический многочлен степени N. Следовательно, если коэффициенты a_k алгебраического многочлена

$$Q_N(t) = \frac{a_0}{2} + \sum_{k=1}^{N} a_k T_k(t) \tag{5.12}$$

вычислены по формуле

$$a_k = \frac{2}{2N+1} \left(y_0 + 2 \sum_{j=1}^{N} y_j T_k(t_j) \right), \tag{5.13}$$

где

$$t_j = \cos \frac{2\pi}{2N+1} j, \tag{5.14}$$

то для всех j от 0 до N имеем

$$Q_N(t_j) = y_j.$$

$\triangleright_7$ Докажите это.

$\triangleright_8^\star$ Можно ли модифицировать формулы (5.12), (5.13) таким образом, чтобы узлы (5.14) были чебышевскими?

6 Кривые Безье

6.1 Приближение вектор-функций одной переменной

Рассмотрим функцию $F : [a, b] \to \mathbb{R}^m$,

$$F(x) = (f_1(x), f_2(x), \ldots, f_m(x))^T, \quad f_i : [a, b] \to \mathbb{R}.$$

Для построения приближения к этой функции можно, очевидно, использовать любой способ приближения функций одной переменной: нужно просто приблизить этим способом каждую компоненту f_i вектор-функции F.

Пусть, например, $\Pi_n : C[a, b] \to \mathbb{P}_n$ — оператор, отображающий произвольную $f \in C[a, b]$ в многочлен $P_n = \Pi_n f$, интерполирующий f по некоторому набору узлов $\{x_i\}_{i=0}^n \subset [a, b]$. Используя представление ИМ в форме Лагранжа, имеем

$$\Pi_n f(x) = \sum_{i=0}^{n} f(x_i)\Lambda_i(x). \tag{6.1}$$

$\triangleright_1$ Докажите, что оператор Π_n линейный.

$\triangleright_2$ Вычислите норму оператора Π_1 для сетки $\{x_0 = a, x_1 = b\}$.

Тогда для всех $x \in [a, b]$ будем иметь

$$F(x) \approx \big(\Pi_n f_1(x), \Pi_n f_2(x), \ldots, \Pi_n f_m(x)\big)^T =: \Pi_n F(x), \tag{6.2}$$

где

$$\Pi_n F(x) = \sum_{i=0}^{n} F(x_i)\Lambda_i(x). \tag{6.3}$$

Понятно, что в формуле (6.2) в качестве оператора Π_n может быть и оператор тригонометрической интерполяции, и вообще любой линейный оператор

$$\Pi : C[a, b] \to X \subset C[a, b],$$

обладающий свойством $\Pi f \approx f$.

Сопоставляя формулы (6.1) и (6.3) видим, что единственная разница между ними состоит в том, что во втором случае коэффициенты при Λ_i — векторы из $\mathbb{R}^m$.

6.2 Приближение пространственных кривых

Рассмотрим на декартовой плоскости кривую ℓ, заданную в параметрическом виде функцией $\gamma : \mathbb{R} \to \mathbb{R}^2$:

$$\gamma(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} \in \mathbb{R}^2, \quad t \in [a, b].$$

При аппроксимации пространственных кривых следует понимать, что одна и та же кривая ℓ может быть параметризована любым образом. То есть, если $\ell \subset \mathbb{R}^2$ — образ отрезка $[a, b]$ при отображении γ, то этот же образ даёт отображение $\gamma \circ g$ отрезка $[c, d]$, где g — любое сюръективное (как правило, конечно, биективное) отображение $g : [c, d] \to [a, b]$.

Это означает, что если для вычислителя представляет интерес приближение *геометрической формы* кривой ℓ (а не точное приближение каждой компоненты вектор-функции γ), то при построении аппроксимирующей кривой $\tilde{\ell} \approx \ell$ имеется свобода в выборе отрезка изменения параметра t.

Для случая полиномиальной интерполяции вышесказанное может быть проиллюстрировано следующим простым примером. Пусть даны $n+1$ точек $q_i = (x_i, y_i)^T \in \mathbb{R}^2$, а также *произвольный* набор попарно различных узлов $\{t_i\}_{i=0}^n \subset [a, b]$ и соответствующий ему фундаментальный базис Лагранжа $\{\Lambda_i\}_{i=0}^n$. Тогда образ вектор-функции

$$\tilde{\gamma}(t) = \sum_{i=0}^{n} q_i \Lambda_i(t), \tag{6.4}$$

на отрезке $[a, b]$, очевидно, содержит все точки q_i.

Все вышесказанное, естественно, справедливо и для случая однопараметрической кривой в трёхмерном пространстве.

6.3 Кривые Безье

6.3.1 Интерактивный дизайн кривой

Рассмотрим задачу *интерактивного дизайна* кривой: требуется построить кривую, обладающую требуемой *формой*. Эта задача существенно отличается от рассматриваемой ранее задачи аппроксимации кривой тем, что здесь нету строго определённого объекта для приближения. Нужно путём визуального подбора построить кривую, обладающую определёнными свойствами (например, имеющую форму кузова автомобиля).

Интерполяция многочленами плохо подходит для решения такой задачи. Это связано с тем, что базисные функции Лагранжа Λ_i хоть и обладают свойством $\Lambda_i(t_j) = \delta_{ij}$, но на всём остальном отрезке интерполяции $[a, b]$ могут принимать значения, намного бóльшие единицы. Это приводит к тому, что небольшое изменение одной точки q_i в формуле (6.4) может сильно изменить значение $\tilde{\gamma}$ в точке t, расположенной достаточно далеко от узла t_i. Этот недостаток можно выразить так: i-ая базисная функция Лагранжа Λ_i *не локализована вблизи* i*-го узла интерполяции*.

Гораздо более удобным с этой точки зрения является базис из многочленов Бернштейна.

6.3.2 Базисные многочлены Бернштейна

Многочлен
$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i} \tag{6.5}$$
называют i-м (базисным) многочленом Бернштейна степени n, $0 \leqslant i \leqslant n$. Здесь $\binom{n}{i} = \frac{n!}{i!(n-i!)}$ — биномиальные коэффициенты. При $i < 0$ и $i > n$ полагаем $B_i^n(t) = 0$.

Множество $\{B_i^n\}_{i=0}^n$ образует базис в пространстве $\mathbb{P}_n$. Многочлены Бернштейна обладают множеством полезных свойств и достаточно широко используются в численном анализе. Наиболее интересны для нас следующие свойства этих многочленов.

1. $B_i^n(t) \geqslant 0 \quad \forall\, t \in [0,1], \quad \forall\, n, i$;
2. $B_0^n(0) = B_n^n(1) = 1$;
   $B_i^n(0) = B_i^n(1) = 0 \quad \forall i = \overline{1, n-1}$;
3. $B_i^n(t) = B_{n-i}^n(1-t)$;
4. $\frac{d}{dt} B_i^n(t) = n(B_{i-1}^{n-1}(t) - B_i^{n-1}(t))$;
5. B_i^n имеет на отрезке $[0,1]$ единственный локальный максимум, который достигается в точке i/n (B_i^n локализована в точке i/n);
6. $\sum_{i=0}^n B_i^n(t) = 1 \quad \forall\, t \in \mathbb{R}, \quad n \geqslant 0$;
7. $B_i^n(t) = (1-t)B_i^{n-1}(t) + tB_{i-1}^{n-1}(t)$.

$\triangleright_3$ Докажите свойства 4-7.

6.3.3 Кривые Безье

В двух словах, кривая Безье — это образ единичного отрезка под действием линейной комбинации базисных функций Бернштейна с векторными коэффициентами. Кривые Безье широко применяются в компьютерной графике, с их помощью работают практически все компьютерные векторные шрифты и много чего ещё.

Рассмотрим набор точек плоскости $Q = \{q_i \in \mathbb{R}^2\}_{i=0}^n$, которые в дальнейшем будем называть *контрольными точками*. Кривой Безье называется множество

$$\beta(Q) = \{B(t) \mid t \in [0,1]\},$$

где

$$B(t) = \sum_{i=0}^{n} q_i B_i^n(t). \tag{6.6}$$

Интересно, что одно из доказательств теоремы Вейерштрасса 4.1 тоже использует многочлены Бернштейна в качестве базиса: если для любой $f \in C[a,b]$ определить

$$B_n = \sum_{i=0}^{n} f\left(\frac{i}{n}\right) B_i^n,$$

то

$$\|f - B_n\| \xrightarrow[n\to\infty]{} 0.$$

К сожалению, скорость сходимости последовательности $\{B_n\}$ слишком мала, чтобы иметь практическую ценность для приближения функций.

Перед тем, как обсудить свойства кривой Безье, рассмотрим вид кривых для $n = 2, 3, 4$:

Рисунки наглядно иллюстрируют следующие свойства кривых Безье:

1. $B(0) = q_0$, $B(1) = q_n$.
2. Касательные к кривой в точках $t = 0$ и $t = 1$ коллинеарны векторам $q_1 - q_0$ и $q_n - q_{n-1}$ соответственно.
3. $\beta(Q) \subset H(Q)$, где $H(Q) \subset \mathbb{R}^2$ — выпуклая оболочка множества контрольных точек Q.

Свойство 2 можно сформулировать более конкретно. Согласно свойству 4 многочленов Бернштейна, имеем

$$\frac{d}{dt}B(t) = \sum_{i=0}^{n} q_i \frac{d}{dt}B_i^n(t) = n\sum_{i=0}^{n} q_i(B_{i-1}^{n-1}(t) - B_i^{n-1}(t)) = \\ = n\sum_{i=0}^{n-1}(q_{i+1} - q_i)B_i^{n-1}(t).$$

Отсюда с учётом свойства 2 многочленов Бернштейна получаем

$$B'(0) = n(q_1 - q_0), \qquad B'(1) = n(q_n - q_{n-1}).$$

6.4 Алгоритмы построения кривых Безье

Рассмотрим задачу построения кривой Безье, задаваемой уравнением (6.6), которая сводится к вычислению значений B в точках $t \in [0, 1]$. Оба способа, которые мы рассмотрим, основываются на рекуррентных соотношениях для многочленов Бернштейна (свойство 7).

6.4.1 Прямой алгоритм

Вычисление $B(t)$ для данного t, очевидно, можно разбить на два этапа:

1. Вычисление $b_i = B_i^n(t)$ для всех i от 0 до n;
2. Вычисление $B(t) = \sum_{i=0}^{n} b_i q_i$.

Основная работа выполняется на первом этапе, поэтому рассмотрим его. Понятно, что грубое вычисление b_i по определению (6.5) — не наш метод. Согласно свойству 7 имеем

$$B_i^n(t) = (1 - t)B_i^{n-1}(t) + tB_{i-1}^{n-1}(t). \tag{6.7}$$

Для того, чтобы вычислить все b_i по такой схеме, необходимо вычислить все $B_j^k(t)$ для $k = \overline{0, n}$, $j = \overline{0, k}$. Делается это последовательным (слева направо) заполнением следующей треугольной таблицы:

$$
\begin{array}{ccccccc}
B_0^0 & B_0^1 & B_0^2 & \cdots & B_0^{n-1} & B_0^n \\
 & B_1^1 & B_1^2 & \cdots & B_1^{n-1} & B_1^n \\
 & & B_2^2 & \cdots & \vdots & \vdots \\
 & & & & B_{n-1}^{n-1} & B_{n-1}^n \\
 & & & & & B_n^n
\end{array}
$$

Хранить всю таблицу не обязательно. Достаточно завести вектор b длины $n+1$ и начиная с

$$b = (B_0^0, 0, 0, \ldots, 0) = (1, 0, 0, \ldots, 0)$$

последовательно заполнить все его компоненты, в итоге получив

$$b = (B_0^n, B_1^n, \ldots, B_n^n).$$

$\triangleright_4$ Запишите соответствующий алгоритм.

При построении графика кривой $\beta(Q)$ можно в два раза сократить объём вычислений, если учесть симметрию базисных многочленов Бернштейна (свойство 3): один раз вычислив b_i в точке t можно вычислить не только $B(t)$, но и

$$B(1-t) = \sum_{i=0}^{n} b_{n-i} q_i.$$

6.4.2 Алгоритм de Casteljau

Формулу (6.7) можно применить непосредственно к многочлену (6.6). Если обозначить

$$B(t) = B(q_0, \ldots, q_n, t),$$

то из указанных двух формул получаем

$$B(q_0, \ldots, q_n, t) = (1-t)\sum_{i=0}^{n} q_i B_i^{n-1} + t\sum_{i=0}^{n} q_i B_{i-1}^{n-1} =$$
$$= (1-t)B(q_0, \ldots, q_{n-1}, t) + t\,B(q_1, \ldots, q_n, t). \quad (6.8)$$

Таким образом, по рекуррентной схеме (6.8) мы можем напрямую вычислить $B(t)$, заполняя треугольную таблицу практически аналогично схеме вычисления разделённых разностей. Начинаем с массива

$$\big(B(q_0,t),B(q_1,t),\ldots,B(q_n,t)\big)=(q_0,q_1,\ldots,q_n)$$

постепенно вычисляем $B(q_i,\ldots,q_{i+k},t)$, складывая соседние элементы массива с весами $(1-t)$ и t. Схема получается такая (точку t в обозначении $B(\ldots)$ опускаем):

$$\begin{array}{lllll}
q_0 & & & & \\
 & B(q_0,q_1) & & & \\
q_1 & & B(q_0,q_1,q_2) & & \\
 & B(q_1,q_2) & & \ddots & \\
q_2 & & B(q_0,q_1,q_2) & B(q_0,\ldots,q_{n-1}) & \\
\vdots & B(q_2,q_3) & \vdots & & B(q_0,\ldots,q_n) \\
\vdots & \vdots & \vdots & B(q_1,\ldots,q_n) & \\
\vdots & \vdots & \vdots & \iddots & \\
q_{n-1} & \vdots & B(q_{n-2},q_{n-1},q_n) & & \\
 & B(q_{n-1},q_n) & & & \\
q_n & & & &
\end{array}$$

$\triangleright_5$ Запишите алгоритм.

$\triangleright_6$ Сравните трудоёмкость обоих алгоритмов.

7 Сплайны

7.1 Определение

Рассмотрим сетку $\{x_i\}_{i=0}^n$,

$$a = x_0 < x_1 < \ldots < x_{n-1} < x_n = b$$

и соответствующее ей разбиение отрезка $[a, b]$ на n частей:

$$\Delta = \{\Delta_i\}_{i=1}^n, \quad \Delta_i = [x_{i-1}, x_i].$$

Говоря простым языком, сплайном называется функция, задаваемая на каждом из отрезков Δ_i алгебраическим многочленом, и при этом обладающая определённым количеством непрерывных производных на $[a, b]$. Строгое определение выглядит так.

Для $m \geqslant 0$ и разбиения Δ рассмотрим функцию s, определённую на $[a, b]$ и обладающую следующими свойствами:

1. $s(x)|_{x\in\Delta_i} = s_i(x), \quad s_i \in \mathbb{P}_m$.
2. $s \in C^{m-1}[a, b]$, что эквивалентно условиям

$$s_i^{(j)}(x_i - 0) = s_{i+1}^{(j)}(x_i + 0), \quad i = \overline{1, n-1}, \quad j = \overline{0, m-1}. \tag{7.1}$$

Такая функция называется *полиномиальным сплайном порядка (степени) m*. Множество всех таких s будем обозначать S_Δ^m.

Легко убедиться, что множество S_Δ^m образует линейное подпространство в $C^{m-1}[a, b]$.

$\triangleright_1$ Чему равна размерность S_Δ^m?

7.2 Интерполяционные сплайны

Сплайн $s \in S_\Delta^m$ называется интерполяционным для функции f, если

$$s(x_i) = f(x_i) = y_i \quad \forall i = \overline{0, n}. \tag{7.2}$$

Понятно, что интерполяционный сплайн $s \in S_\Delta^1$ представляет собой кусочно-линейную функцию (график — ломаная линия), построенную по точкам $\{(x_i, y_i)\}_{i=0}^n$. Больший интерес, конечно, представляют сплайны высших степеней. Наиболее распространёнными являются кубические интерполяционные сплайны.

7.2.1 Построение кубического интерполяционного сплайна

Итак, рассмотрим задачу нахождения $s \in S^3_\Delta$, удовлетворяющего условиям (7.2). Каждый «кусок» сплайна будем искать в виде

$$s_i(x) = \alpha_i + \beta_i(x - x_i) + \frac{\gamma_i}{2}(x - x_i)^2 + \frac{\delta_i}{6}(x - x_i)^3, \quad x \in \Delta_i = [x_{i-1}, x_i].$$

Таким образом, нам нужно определить $4n$ неизвестных $\{\alpha_i, \beta_i, \gamma_i, \delta_i\}_{i=1}^n$. Обозначим

$$\alpha_0 = s_1(x_0), \quad \beta_0 = s_1'(x_0), \quad \gamma_0 = s_1''(x_0), \quad \delta_0 = s_1'''(x_0), \tag{7.3}$$

после чего с полным правом можем записать

$$s(x_i) = \alpha_i, \quad s'(x_i) = \beta_i, \quad s''(x_i) = \gamma_i, \quad \forall\, i = \overline{0, n}.$$

Величины γ_i называются *моментами* сплайна. Наиболее популярный в литературе способ построения кубического сплайна заключается в следующем: рассматривается $s'' \in S^1_\Delta$, «склеенная» из s_i'' — линейных функций, проходящих через точки (x_{i-1}, γ_{i-1}) и (x_i, γ_i). Записывая уравнение s_i'' и дважды интегрируя его, из условий (7.1) находятся неизвестные коэффициенты многочленов s_i. Мы рассмотрим другой, более простой на наш взгляд, способ вывода расчётных формул: вместо интегрирования будем дифференцировать [Самарский, Гулин, «Численные методы», 1989].

Обозначим $h_i = x_i - x_{i-1}$. Прежде всего используем условия интерполяции (7.2). Имеем $s_i(x_i) = y_i$, откуда

$$\alpha_i = y_i, \quad i = \overline{1, n}, \tag{7.4a}$$

и $s_i(x_{i-1}) = y_{i-1}$, или

$$\beta_i = \frac{y_i - y_{i-1}}{h_i} + \frac{\gamma_i}{2}h_i - \frac{\delta_i}{6}h_i^2, \quad i = \overline{1, n}. \tag{7.4b}$$

Отметим, что выполнение этих двух условий сразу гарантирует непрерывность s. Теперь используем условия непрерывности s': $s_i'(x_{i-1}) = s_{i-1}'(x_{i-1})$, или

$$\gamma_i h_i - \frac{\delta_i}{2}h_i^2 = \beta_i - \beta_{i-1}, \quad i = \overline{2, n}. \tag{7.4c}$$

Ну и, наконец, непрерывность s'' аналогично даёт

$$\delta_i = \frac{\gamma_i - \gamma_{i-1}}{h_i}, \quad i = \overline{2, n}. \tag{7.4d}$$

Важно: если в последних двух формулах положить $i = 1$, то мы получим просто определение величин β_0 и γ_0 согласно (7.3).

Формулы (7.4) дают $4n-2$ уравнений для нахождения $4n$ неизвестных, поэтому без двух дополнительных условий сплайн однозначно не определить. Как правило эти условия задаются на концах отрезка $[a, b]$ и поэтому называются *граничными*.

Прежде, чем рассматривать различные варианты задания дополнительных условий для определения интерполяционного кубического сплайна, заметим, что если известны значения моментов $\{\gamma_i\}_{i=0}^n$, то остальные коэффициенты сразу находятся по формулам (7.4a), (7.4b) и (7.4d) (в последней нужно взять ещё $i = 1$). В частности, (7.4b) даёт

$$\beta_i = \frac{y_i - y_{i-1}}{h_i} + \frac{2\gamma_i + \gamma_{i-1}}{6} h_i. \tag{7.4b'}$$

Для нахождения $\{\gamma_i\}$ подставим (7.4b') и (7.4d) в (7.4c). После сдвига в нумерации получаем

$$h_i\gamma_{i-1} + 2(h_i + h_{i+1})\gamma_i + h_{i+1}\gamma_{i+1} = 6\left(\frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i}\right)$$

что можно записать как

$$c_i\gamma_{i-1} + 2\gamma_i + e_i\gamma_{i+1} = b_i, \quad i = \overline{1, n-1}, \tag{7.5}$$

где

$$c_i = \frac{h_i}{h_i + h_{i+1}}, \quad e_i = \frac{h_{i+1}}{h_i + h_{i+1}}, \quad b_i = 6f[x_{i-1}, x_i, x_{i+1}].$$

Формулы (7.5) дают $(n-1)$ линейных уравнений для нахождения $n+1$ неизвестных $\{\gamma_i\}_{i=0}^n$.

Таким образом, **общая схема вычисления интерполяционного кубического сплайна** такова:

1. Вычисляются моменты $\{\gamma_i\}_{i=0}^n$ как решение СЛАУ (7.5), дополненной двумя дополнительными уравнениями — граничными условиями.
2. Находятся остальные коэффициенты по формулам (7.4a), (7.4b') и (7.4d).

7.2.2 Виды граничных условий

Естественные граничные условия. Самые удобные граничные условия

$$s''(a) = s''(b) = 0,$$

как и соответствующий интерполяционный сплайн, называются *естественными*. В этом случае имеем $\gamma_0 = \gamma_n = 0$, а остальные $\{\gamma_i\}_{i=1}^{n-1}$ легко найти из (7.5), которая имеет вид

$$\begin{bmatrix} 2 & e_1 & & & & \\ c_2 & 2 & e_2 & & & \\ & c_3 & 2 & e_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & c_{n-2} & 2 & e_{n-2} \\ & & & & c_{n-1} & 2 \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \vdots \\ \gamma_{n-2} \\ \gamma_{n-1} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_{n-2} \\ b_{n-1} \end{bmatrix},$$

Данную СЛАУ решаем, естественно, методом прогонки.

$\triangleright_2$ Докажите, что метод прогонки в данном случае применим.

$\triangleright_3$ Запишите вид аналогичной СЛАУ для граничных условий вида $s''(x_0) = f''(x_0)$, $s''(x_n) = f''(x_n)$.

Неестественные граничные условия. Пусть в концах отрезка сплайн s должен удовлетворять дополнительным условиям

$$s'(a) = f'(a), \quad s'(b) = f'(b),$$

что равносильно $\beta_0 = f'(a)$, $\beta_n = f'(b)$. Рассмотрим (7.4c) при $i = 1$ с учётом (7.4b'):

$$\gamma_1 h_1 - \frac{\gamma_1 - \gamma_0}{2} h_1 = \frac{y_1 - y_0}{h_1} + \frac{2\gamma_1 + \gamma_0}{6} h_1 - f'(a),$$

откуда получаем

$$2\gamma_0 + \gamma_1 = \frac{6}{h_1}\left(\frac{y_1 - y_0}{h_1} - f'(a)\right) = 6f[x_0, x_0, x_1].$$

Аналогично для $i = n - 1$ имеем

$$\gamma_{n-1} + 2\gamma_n = \frac{6}{h_n}\left(f'(b) - \frac{y_n - y_{n-1}}{h_1}\right) = 6f[x_{n-1}, x_n, x_n].$$

$\triangleright_4$ Запишите соответствующую СЛАУ для нахождения $\{\gamma_i\}_{i=0}^{n}$ и докажите, что к ней применим метод прогонки.

Сверхъестественные граничные условия. На практике используется ещё несколько видов граничных условий. Например, периодические:

$$s'(a) = s'(b), \quad s''(a) = s''(b).$$

Или так называемые условия «отсутствия узлов», накладывающие требования непрерывности s''' в узлах x_1 и x_{n-1}.

$\triangleright_5$ Почему в этом случае говорят об отсутствии узлов?

Интересен также вариант

$$s'(a) = f'(a), \quad s''(a) = f''(a).$$

В этом случае вообще не нужно решать никакой СЛАУ — все s_i вычисляются явно по очереди от 1 до n.

7.3 Экстремальное свойство кубического сплайна

Рассмотрим функционал $\Phi : C^2[a, b] \to \mathbb{R}$,

$$\Phi(f) = \int_a^b (f''(x))^2 dx.$$

Экстремальное свойство интерполяционного сплайна $s \in S_\Delta^3$ состоит в том, что он доставляет минимум Φ среди всех функций из $C^2[a, b]$, интерполирующих f и удовлетворяющих требуемым граничным условиям. Сформулируем это утверждение более строго для случая естественных граничных условий.

Теорема 7.1. *Пусть $F_0 \subset C^2[a, b]$ — множество функций f таких, что*

$$f(x_i) = y_i, \quad i = \overline{0, n}, \quad \text{и} \quad f''(a) = f''(b) = 0.$$

Если кубический сплайн $s \in S_\Delta^3$ обладает свойством $s \in F_0$, то

$$\Phi(s) \leqslant \Phi(f) \quad \forall f \in F_0.$$

Доказательство. Для произвольной $f \in F_0$ рассмотрим функцию $g = f - s$. По условию имеем $g \in F_0$ и

$$g(x_i) = 0 \quad \forall i = \overline{0, n}.$$

Расписывая $\Phi(f) = \Phi(s + g)$, получаем

$$\Phi(f) = \Phi(s) + \Phi(g) + 2 \int_a^b s''g''dx.$$

Для доказательства теоремы теперь достаточно показать, что

$$\int_a^b s''g''dx \geqslant 0.$$

Для этого воспользуемся интегрированием по частям, а также тем фактом, что s''' на интервале (x_{i-1}, x_i) равна константе, которую обозначим δ_i. Вычисляем:

$$\int_a^b s''g''dx = \sum_{i=1}^n \int_{x_{i-1}}^{x_i} s''g''dx = \sum_{i=1}^n \left(s''g' \Big|_{x_{i-1}}^{x_i} - \int_{x_{i-1}}^{x_i} s'''g'dx \right) =$$

$$= -\sum_{i=1}^n \int_{x_{i-1}}^{x_i} s'''g'dx = -\sum_{i=1}^n \delta_i \int_{x_{i-1}}^{x_i} g'dx = -\sum_{i=1}^n \delta_i \Big(g(x_i) - g(x_{i-1}) \Big) dx = 0.$$

■

Отметим, что в аналитической геометрии кривизна кривой-графика $y = f(x)$ вычисляется по формуле

$$\frac{|f''(x)|}{(1 + f'(x)^2)^{3/2}},$$

так что величину $\Phi(f)$ можно характеризовать как приближённое значение для усреднённой кривизны графика. Поэтому часто говорят (пусть и не совсем корректно), что интерполяционный сплайн обладает минимальной кривизной среди всех интерполирующих функций с указанными граничными условиями.

Заметим также, что аналогичные теореме 7.1 утверждения справедливы и для других типов граничных условий.

7.4 Сходимость интерполяции кубическим сплайном

Приведём без доказательства следующий результат.

Теорема 7.2. *Рассмотрим $f \in C^4[a, b]$ и разбиение Δ отрезка $[a, b]$ такое, что $\max_i h_i = h$. Пусть $s \in S_\Delta^3$ — естественный интерполяционный сплайн для f. Тогда*

$$\|f - s\| \leqslant \frac{5}{384} h^4 \|f^{(4)}\|.$$

Здесь $\|\cdot\| = \|\cdot\|_{C[a,b]}$.

Таким образом, для достаточно гладких f погрешность интерполирования кубическим сплайном $\varepsilon(h)$ равна $O(h^4)$.

7.5 Кубический эрмитов сплайн

Кусочно-полиномиальная на разбиении Δ функция s называется *эрмитовым сплайном* третьей степени для $f \in C^1[a, b]$, если

$$s|_{x\in\Delta_i} = s_i \in \mathbb{P}_3,$$

и

$$s(x_i) = f(x_i) \quad \text{и} \quad s'(x_i) = f'(x_i) \quad \forall\, i = \overline{0, n}.$$

Отметим, что эрмитов сплайн s формально не является кубическим сплайном, так как $s \notin C^2[a, b]$. Его можно назвать кубическим сплайном дефекта 2 (обычный сплайн имеет дефект 1).

$\triangleright_6$ Докажите, что кубический эрмитов сплайн существует и единственен.

$\triangleright_7$ Выведите формулы для вычисления такого сплайна.

Для погрешности эрмитова сплайна также имеет место хорошая оценка

$$\|f - s\| \leqslant \frac{1}{384} h^4 \|f^{(4)}\|.$$

8 Базисы в пространстве сплайнов

Во многих приложениях полезно иметь базис для линейного пространства сплайн-функций S_Δ^m. Так как $\dim S_\Delta^m = n + m$, то базис образует, вообще говоря, любое множество $n + m$ линейно независимых функций из S_Δ^m. Если $\{\varphi_i\}_{i=0}^{n+m-1}$ — такое множество, то любой сплайн $s \in S_\Delta^m$, очевидно, может единственным образом быть представлен в виде

$$s = \sum_{i=0}^{n+m-1} \alpha_i \varphi_i, \quad \alpha_i \in \mathbb{R}.$$

Для разных задач, понятное дело, удобны различные наборы базисных сплайнов.

8.1 Фундаментальные базисы сплайнов

Сначала построим для S_Δ^m базис, удобный для решения задачи интерполяции — фундаментальный базис сплайнов, который также иногда называют *кардинальным* (*cardinal spline basis*). Тут будет иметь место практически полная аналогия с встречавшимися ранее фундаментальными базисами: множество сплайнов

$$\{\psi_i^m\}_{i=0}^n \subset S_\Delta^m$$

будем называть фундаментальным сплайн-базисом m-го порядка, если

$$\psi_i^m(x_j) = \delta_{ij} \quad \forall i, j = \overline{0, n}. \tag{8.1}$$

Главный нюанс заключается в том, что в общем случае фундаментальных сплайнов не достаточно, чтобы описать всё пространство S_Δ^m: всего их $n + 1$ штука, а надо $n + m$.

8.1.1 Фундаментальные сплайны первого порядка

При $m = 1$ условий (8.1) как раз хватает, чтобы найти все $\{\psi_i^1\}$. Так как ψ_i^1 являются кусочно-линейными функциями на разбиении Δ, легко выписать следующие явные формулы:

$$\psi_0^1(x) = \begin{cases} \dfrac{x_1 - x}{h_1}, & x \in \Delta_1, \\ 0, & x \notin \Delta_1; \end{cases} \tag{8.2a}$$

$$\psi_i^1(x) = \begin{cases} \dfrac{x - x_{i-1}}{h_i}, & x \in \Delta_i, \\ \dfrac{x_{i+1} - x}{h_{i+1}}, & x \in \Delta_{i+1}, \\ 0, & x \notin \Delta_i \cup \Delta_{i+1}, \end{cases} \qquad i = \overline{1, n-1}; \tag{8.2b}$$

$$\psi_n^1(x) = \begin{cases} \dfrac{x - x_{n-1}}{h_n}, & x \in \Delta_n, \\ 0, & x \notin \Delta_n. \end{cases} \tag{8.2c}$$

На протяжении всего курса мы будем достаточно плотно работать с этими базисными функциями, поэтому формулы (8.2) нужно усвоить очень хорошо.

По построению очевидно, что любой сплайн первого порядка раскладывается по базису $\{\psi_i^1\}_{i=0}^n$ следующим образом:

$$s(x) = \sum_{i=0}^{n} s(x_i)\psi_i^1(x).$$

8.1.2 Фундаментальные сплайны третьего порядка

В кубическом случае для определения фундаментального базисного сплайна (как и в случае произвольного интерполяционного кубического сплайна) не хватает двух условий. Поэтому для каждого типа граничных условий будут свои фундаментальные сплайны.

Естественные фундаментальные сплайны. Прежде всего заметим, что множество всех естественных кубических сплайнов (т. е., удовлетворяющих условиям $s''(a) = s''(b) = 0$) образует линейное подпространство $S_0 \subset S_\Delta^3$. Размерность этого подпространства равна $n+1$, а его фундаментальный базис $\{\psi_i^3\}_{i=0}^n$ строится очевидным образом по общей схеме, рассмотренной в предыдущей лекции, по условиям

$$\psi_i^3(x_j) = \delta_{ij}, \quad j = \overline{0,n}, \quad (\psi_i^3)''(x_0) = (\psi_i^3)''(x_n) = 0$$

для всех $i = \overline{0,n}$. Вычисление базиса по такой схеме требует решения $n+1$ СЛАУ с одной и той же трёхдиагональной матрицей, поэтому вместо метода прогонки в данном случае уместно применить метод LU-разложения, оптимизированный для трёхдиагональных матриц.

Как только построен базис $\{\psi_i^3\}_{i=0}^n$, любой интерполяционный кубический сплайн с естественными граничными условиями мгновенно строится по формуле

$$s(x) = \sum_{i=0}^{n} y_i \psi_i^3(x).$$

Заданные значения второй производной. Рассмотрим теперь граничные условия вида

$$s''(a) = \gamma_0, \quad s''(b) = \gamma_n, \tag{8.3}$$

где γ_0 и γ_n — произвольные (не известные заранее) числа. В этом случае нам необходимо $n+3$ базисных функций, так как множество сплайнов, удовлетворяющих граничным условиям вида (8.3), совпадает с S_Δ^3. Нетрудно заметить, что каждый такой сплайн можно представить в виде

$$s = s_0 + s_1,$$

где $s_0 \in S_0$, $s_0(x_i) = s(x_i) \ \ \forall i = \overline{1,n}$, а сплайн s_1 удовлетворяет условиям

$$s_1(x_i) = 0 \quad \forall\, i = \overline{0,n},$$

и

$$s_1''(a) = \gamma_0, \quad s_1''(b) = \gamma_n.$$

Следовательно, искомый интерполяционный сплайн можно записать в виде

$$s(x) = \sum_{i=0}^{n} y_i \psi_i^3(x) + \gamma_0 \psi_{n+1}^3(x) + \gamma_n \psi_{n+2}^3(x),$$

где $\{\psi_i^3\}_{j=0}^n$ — построенные ранее естественные фундаментальные сплайны, а ψ_{n+1}^3 и ψ_{n+2}^3 — базисные кубические сплайны, удовлетворяющие условиям

$$\psi_{n+1}^3(x_i) = \psi_{n+2}^3(x_i) = 0 \quad \forall\, i = \overline{0,n},$$

и

$$(\psi^3_{n+1})''(x_0) = 1, \quad (\psi^3_{n+1})''(x_n) = 0,$$
$$(\psi^3_{n+2})''(x_0) = 0, \quad (\psi^3_{n+1})''(x_n) = 1.$$

Аналогичным образом строится фундаментальный базис кубических сплайнов для остальных типов граничных условий, описанных в предыдущей лекции.

$\triangleright_1$ По аналогии выпишите условия, которые определяют фундаментальные сплайны второго порядка для какого-нибудь типа граничных условий.

8.2 B-сплайны

Если фундаментальные базисные сплайны это аналог базисных многочленов Лагранжа $\{\Lambda_i\}$, то B-сплайны (bell splines) в некотором роде являются родственниками базисных многочленов Бернштейна. Существует много эквивалентных и не очень способов определения B-сплайнов. Опишем один из них.

8.2.1 Построение B-сплайнов.

Прежде всего, расширим отрезок $[a, b]$ до размеров всей вещественной оси, а сетку узлов $\{x_i\}$ будем считать бесконечной:

$$\ldots < x_{-1} < x_0 < x_1 < \ldots$$

Определение сплайна порядка m из лекции 7, очевидно, легко переносится на этот случай, поэтому сохраним и обозначение пространства сплайнов на таком разбиении — S^m_Δ. При $m = 0$ мы получаем пространство S^0_Δ, элементами которого являются кусочно-постоянные функции с разрывами в точках x_i. Общепринято считать, что сплайны нулевого порядка непрерывны справа.

B-сплайны порядка m будем обозначать N^m_i, $i \in \mathbb{Z}$, обозначим также

$$\mathbf{N}^m = \{N^m_i\}_{i\in\mathbb{Z}}.$$

Порядок 0. B-сплайны нулевого порядка — это фундаментальные базисные функции для пространства S^0_Δ.

B-сплайнами порядка 0 называются функции

$$N_i^0(x) = \begin{cases} 1, & x \in [x_i, x_{i+1}), \\ 0, & \text{иначе;} \end{cases} \qquad \forall\, i \in \mathbb{Z}. \tag{8.4}$$

Порядок 1. Разница между B-сплайнами и фундаментальными сплайнами первого порядка состоит исключительно в способе нумерации:

$$N_i^1(x) = \psi_{i+1}^1(x) = \begin{cases} \dfrac{x - x_i}{x_{i+1} - x_i}, & x \in [x_i, x_{i+1}), \\ \dfrac{x_{i+2} - x}{x_{i+2} - x_{i+1}}, & x \in [x_{i+1}, x_{i+2}), \\ 0, & x \notin [x_i, x_{i+2}); \end{cases} \qquad \forall i \in \mathbb{Z}. \tag{8.5}$$

Перечислим наиболее важные свойства функций N_i^0 и N_i^1.

1. Эти функции имеют компактный носитель (см. ниже).
2. $N(x) \geqslant 0 \;\; \forall x \in \mathbb{R}, \; \forall N \in \mathbf{N}^0 \cup \mathbf{N}^1$.
3. $\sum_{i=-\infty}^{\infty} N_i^m(x) = 1 \;\; \forall x \in \mathbb{R}, \; m = 0, 1$.

Носителем функции f называется наименьшее замкнутое множество, вне которого f тождественно равна нулю:

$$\operatorname{supp} f = \overline{\{x \in \mathbb{R} \mid f(x) \neq 0\}}.$$

Если носитель функции f — компактное множество (отрезок в случае $\mathbb{R}$), то такая f называется *финитной*, или просто *функцией с компактным носителем*.

Перечисленные выше свойства, особенно первые два, весьма полезны в различных приложениях. Поэтому B-сплайны высоких порядков строятся таким образом, чтобы и они тоже обладали этими свойствами.

B-сплайны произвольного порядка. Для определения B-сплайна произвольного порядка установим связь между множествами $\mathbf{N}^0$ и $\mathbf{N}^1$. Во-первых, имеем

$$\operatorname{supp} N_i^1 = [x_i, x_{i+2}] = [x_i, x_{i+1}] \cup [x_{i+1}, x_{i+2}] = (\operatorname{supp} N_i^0) \cup (\operatorname{supp} N_{i+1}^0),$$

так что N_i^1 можно представить в виде линейной комбинации с переменными коэффициентами от функций N_i^0 и N_{i+1}^0. Эти переменные коэффициенты должны удовлетворять двум условиям: во-первых, они должны быть, очевидно, многочленами первой степени, и, во-вторых, они должны гарантировать непрерывность N_i^1. Непосредственной проверкой убеждаемся в справедливости соотношения

$$N_i^1(x) = \frac{x - x_i}{x_{i+1} - x_i} N_i^0(x) + \frac{x_{i+2} - x}{x_{i+2} - x_{i+1}} N_{i+1}^0(x),$$

которое удобно записать в компактной форме

$$N_i^1(x) = V_i^1 N_i^0 + (1 - V_{i+1}^1) N_{i+1}^0, \tag{8.6}$$

где

$$V_i^1(x) = \frac{x - x_i}{x_{i+1} - x_i}.$$

Теперь подумаем, какими свойствами должны обладать квадратичные B-сплайны. Носитель сплайна N_i^2 по аналогии должен быть $[x_i, x_{i+3}]$, и порождаться этот сплайн будет B-сплайнами первого порядка N_i^1 и N_{i+1}^1 по формуле, аналогичной (8.6).

Таким образом, в общем случае мы будем иметь следующее определение.

B-сплайном порядка $m \geqslant 1$ называется функция $N_i^m \in S_\Delta^m$, определяемая соотношением

$$N_i^m = V_i^m N_i^{m-1} + (1 - V_{i+1}^m) N_{i+1}^{m-1}, \quad \forall i \in \mathbb{Z}, \tag{8.7}$$

где

$$V_i^m(x) = \frac{x - x_i}{x_{i+m} - x_i}. \tag{8.8}$$

B-сплайны N_i^0 определяются формулой (8.4).

$\triangleright_2$ Постройте явные формулы для вычисления N_i^2.

Без доказательства приведём следующие важные свойства сплайнов N_i^m.

1. $N_i^m \in C^{m-1}(\mathbb{R})$;
2. $\{N_i^m\}_{i \in \mathbb{Z}}$ линейно независимы;

3. $\operatorname{supp} N_i^m = [x_i, x_{i+m+1}]$ (следует из определения);
4. $N_i^m(x) \geqslant 0 \quad \forall i \in \mathbb{Z}\,,\ x \in \mathbb{R}$ (следует оттуда же);
5. $\sum\limits_{i\in\mathbb{Z}} N_i^m(x) = 1 \quad \forall m \geqslant 0\,,\ \forall x \in \mathbb{R}\,.$

   $\triangleright_3$ Докажите это свойство по индукции.

Из первых двух свойств следует, что множество $\mathbf{N}^m$ действительно образует базис пространства S_Δ^m.

8.2.2 Вычислительный алгоритм

На практике наиболее часто встречается задача вычисления линейной комбинации B-сплайнов

$$N(x) = \sum_{i\in\mathbb{Z}} c_i N_i^m(x), \tag{8.9}$$

где c_i — скалярные или векторные коэффициенты.

Ввиду свойства 3 имеем, что в сумме (8.9) ненулевыми будут лишь те слагаемые, для которых $x \in \operatorname{supp} N_i^m$. Пусть $x \in [x_k, x_{k+1})$, тогда

$$N(x) = \sum_{i=k-m}^{k} c_i N_i^m(x).$$

Задача свелась таким образом к вычислению значение N_i^m для $i = \overline{k-m, k}$. Решается она путём прямого использования рекуррентных соотношений (8.7) аналогично тому, как в лекции 6 вычислялись значения многочленов Бернштейна. Слева направо вычисляется треугольная таблица вида

$$\begin{array}{ccccccc}
N_k^0 & N_k^1 & N_k^2 & \cdots & N_k^{m-1} & N_k^m \\
 & N_{k-1}^1 & N_{k-1}^2 & \cdots & N_{k-1}^{m-1} & N_{k-1}^m \\
 & & N_{k-2}^2 & \cdots & \vdots & \vdots \\
 & & & & N_{k-m+1}^{m-1} & N_{k-m+1}^m \\
 & & & & & N_{k-m}^m
\end{array}$$

Совершенно аналогично лекции 6, вычисления начинаются с вектора

$$(N_k^0(x), 0, 0, \ldots, 0) = (1, 0, 0, \ldots, 0),$$

который постепенно превращается в

$$(N_k^m, N_{k-1}^m, \ldots, N_{k-m}^m).$$

Кроме рассмотренного алгоритма существует и аналог алгоритма de Casteljau — алгоритм де Бура (de Boor). Его достаточно легко построить по аналогии.

8.2.3 B-сплайны на конечном отрезке

Вернёмся теперь снова на конечный отрезок $[a, b]$. Для этого нам понадобится запись

$$f|_{[a,b]},$$

которая обозначает *ограничение* функции $f : \mathbb{R} \to \mathbb{R}$ на отрезок $[a, b]$.

Пусть построен базис сплайнов $\mathbf{N}^m$ для бесконечного набора точек $\{x_i\}_{i\in\mathbb{Z}}$. Тогда базис пространства сплайнов порядка m на отрезке $[a, b] = [x_0, x_n]$ с узлами $\{x_i\}_{i=0}^n$ образует множество $\mathbf{N}_n^m \subset \mathbf{N}^m$, в которое входят ограничения на $[a, b]$ всех сплайнов N_i^m, которые тождественно не равны 0 на $[a, b]$[1]:

$$\mathbf{N}_n^m = \{N_i^m|_{[a,b]} \ : \ \operatorname{supp} N_i^m \cap (a, b) \neq \emptyset\} = \{N_i^m|_{[a,b]}\}_{i=-m}^{n-1}.$$

Заметьте, что мы получили ровно столько функций, сколько нужно — $n+m$ штук.

Теперь — внимание, начинается волшебство. Базисные функции из полученного множества $\mathbf{N}_n^m$ однозначно определяются набором узлов

$$X_n^m = \{x_i\}_{i=-m}^{n+m}.$$

Это означает, что в стандартной ситуации, когда у нас есть только множество $\{x_i\}_{i=0}^n$, для построения базиса B-сплайнов $\mathbf{N}_n^m$ по определению (8.7) нужно доопределить $2m$ узлов x_i для $i = \overline{-m, -1}$ и $i = \overline{n+1, n+m}$ (назовём их *виртуальными узлами*). Причем сделать это, вообще говоря, можно *произвольным образом*, лишь бы выполнялось условие $x_i < x_{i+1}$. Но и это условие (!) можно ослабить: взять $x_i = x_0$ для всех $i = \overline{-m, -1}$ и $x_i = x_n$ для всех $i = \overline{-m, -1}$. Чтобы это имело смысл, нужно обобщить определение B-сплайна (8.7) на случай совпадающих узлов.

[1] Вообще говоря, это утверждение нужно доказывать, хотя оно достаточно очевидно.

Пусть последовательность узлов $\{x_i\}_{i\in\mathbb{Z}}$ удовлетворяет условию

$$x_i \leqslant x_{i+1}.$$

B-сплайном порядка $m \geqslant 1$ для таких узлов называется функция $N_i^m \in S_\Delta^m$, определяемая соотношением

$$N_i^m(x) = \frac{x - x_i}{x_{i+m} - x_i} N_i^{m-1}(x) + \frac{x_{i+m+1} - x}{x_{i+m+1} - x_{i+1}} N_{i+1}^{m-1}(x), \quad \forall i \in \mathbb{Z}. \qquad (8.7')$$

В случае, когда знаменатель в каком-либо слагаемом из этой формулы обращается в 0, само это слагаемое полагается равным 0.
B-сплайны N_i^0 по-прежнему определяются формулой (8.4). В противоречивом случае $x_i = x_{i+1}$ берём $N_i^0 = 0$.

Заметим, что формулы (8.7) и (8.7′) идентичны.

Возникает закономерный вопрос: зачем нужны эти фокусы с совпадающими узлами на концах отрезка? Во-первых, это несколько уменьшает объём вычислений. Во-вторых, мы получаем полезные свойства, которыми обладает базис многочленов Бернштейна: все функции базиса $\mathbf{N}_n^m$, кроме первой и последней, обратятся в нуль на концах отрезка, а сами крайние функции на концах отрезка будут равны единице (убедитесь в этом самостоятельно).

Кроме этого, важен следующий факт: в общем случае увеличение кратности i-го узла на единицу на столько же уменьшает гладкость всех базисных B-сплайнов в этом узле и, как следствие, всех сплайнов, порождённых их линейными комбинациями.

8.3 B-сплайновые кривые

Рассмотрим приложение B-сплайнов к задаче интерактивного дизайна кривой.

Пусть $\{q_i\}_{i=0}^M$ — контрольные точки на плоскости. Построим обобщение кривой Безье для этих контрольных точек, используя в качестве базиса B-сплайны. Как обычно, в качестве отрезка параметризации возьмём $[a, b] = [0, 1]$. Нам нужно $M + 1$ линейно независимых на этом отрезке базисных сплайнов порядка m, желательно локализованных в точках $\left\{\frac{i}{M}\right\}_{i=0}^M$. Искомый базис сплайнов обозначим по традиции $\{N_i^m\}_{i=0}^M$.

Первый вопрос, который нужно решить, это сколько внутренних узлов нужно брать между 0 и 1 для корректного определения базисных сплайнов. Ответ: пусть n — число отрезков, на которое надо разбить $[0, 1]$. Согласно теории, базисных сплайнов должно быть $n + m = M + 1$, откуда

$$n = M - m + 1. \qquad (8.10)$$

Таким образом, для построения базиса сплайнов отрезок $[0,1]$ нужно разбить на $M-m+1$ частей.

Второй вопрос — как выбирать внутренние узлы. Ответ: самый простой и удобный способ в общем случае — равномерно расположенные узлы. Таким образом получаем «базовую» сетку

$$\left\{0, \frac{1}{n}, \frac{2}{n}, \ldots, \frac{n-1}{n}, 1\right\}, \quad n = M - m + 1. \tag{8.11}$$

Третий вопрос — выбор виртуальных узлов: нужно доопределить по m штук в начале и в конце последовательности (8.11). Ответ: согласно вышеизложенному, берём нули в начале и единицы в конце. Получаем таким образом набор узлов

$$X = \left\{\underbrace{0, \ldots, 0}_{m+1}, \frac{1}{n}, \frac{2}{n}, \ldots, \frac{n-1}{n}, \underbrace{1, \ldots, 1}_{m+1}\right\} = \{x_i\}_{i=0}^{M+m+1}, \tag{8.12}$$

итого $M+m+2$ штуки — ровно столько, сколько нужно для определения B-сплайнов $\{N_i^m\}_{i=0}^{M}$ по формуле (8.7′).

Вот, собственно, и всё. Базис построен, осталось нарисовать образ вектор-функции

$$N^m(t) = \sum_{i=0}^{M} q_i N_i^m(t)$$

для $t \in [0,1]$ — это и есть искомая B-сплайновая кривая.

В заключение отметим, что согласно (8.10) максимально допустимая степень сплайна для M контрольных точек равна $m = M$. В этом случае будем иметь $n = 1$ и

$$N_i^M(t) = B_i^M(t) \quad \forall i = \overline{0, M}, \quad \forall t \in [0,1],$$

(B_i^M — многочлены Бернштейна), то есть B-сплайновая кривая превращается в кривую Безье!

9 Среднеквадратичные приближения

В этой лекции мы познакомимся с другим важным способом приближения функций — среднеквадратичным приближением, или, говоря умными словами, наилучшим приближением в гильбертовом пространстве. Если попытаться в двух словах описать суть этого подхода, то можно сказать, что найти наилучшее среднеквадратичное приближение — это практически то же самое, что найти проекцию точки евклидова пространства $\mathbb{R}^n$ на какое-то m-мерное подпространство (гиперплоскость).

9.1 Геометрические основы

В трёхмерном евклидовом пространстве $\mathbb{R}^3$ рассмотрим точку (вектор) p и два линейно независимых вектора v_1 и v_2, линейная оболочка которых образует плоскость $L \subset \mathbb{R}^3$. Наша задача — среди всех точек $q \in L$ найти самую близкую к p точку q^*:

$$\|q^* - p\| = \inf_{q \in L} \|q - p\|.$$

Здесь $\|\cdot\|$ — естественно, евклидова норма. Из геометрии мы знаем, что искомая точка q^* — *проекция* точки p на L.

Определяющее свойство проекции заключается в том, что любой вектор $q \in L$ *ортогонален* вектору $p - q^*$. А так как q раскладывается по базису $\{v_1, v_2\}$, то для нахождения q^* достаточно потребовать, чтобы $p - q^*$ был ортогонален обоим базисным векторам:

$$\begin{cases} (p - q^*, v_1) = 0, \\ (p - q^*, v_2) = 0. \end{cases}$$

Так как $q^* \in L$, представляем его в виде $q^* = \alpha_1 v_1 + \alpha_2 v_2$, подставляем в

последнюю формулу и получаем СЛАУ для нахождения неизвестных коэффициентов α_1 и α_2:

$$\begin{cases} \alpha_1(v_1, v_1) + \alpha_2(v_2, v_1) = (p, v_1), \\ \alpha_1(v_1, v_2) + \alpha_2(v_2, v_2) = (p, v_2). \end{cases}$$

Задача решена. Совершенно аналогично решается подобная задача в случае, когда $p \in \mathbb{R}^n$, а $L = \text{span}\{v_1, \ldots, v_m\}$: коэффициенты $\alpha = (\alpha_1, \ldots, \alpha_m)^T$ проекции

$$q^* = \sum_{i=1}^{m} \alpha_i v_i$$

находятся как решение СЛАУ вида

$$\Gamma\alpha = b,$$

где $\Gamma = \Big((v_j, v_i)\Big)_{i,j=1}^{m}$ — матрица Грамма, $b_i = (p, v_i)$, $i = \overline{1, m}$.

9.2 Наилучшее приближение в гильбертовом пространстве

Оказывается, что приведённые выкладки справедливы не только для векторов из $\mathbb{R}^n$, но и для элементов всех *гильбертовых* пространств. Гильбертово пространство — это обобщение понятия евклидова пространства. Ключевое отличие гильбертовых пространств от остальных функциональных пространств ровно то же, что отличает евклидово пространство от простого линейного пространства: наличие операции *скалярного произведения*. Как только определено скалярное произведение, сразу же обретают смысл слова «ортогональность» и «проекция» — термины, с помощью которых решается задача о поиске наилучшего приближения.

Гильбертовым пространством (ГП) называется линейное векторное пространство со скалярным произведением $(\cdot, \cdot)$, *полное* относительно нормы, порождённой этим скалярным произведением ($\|u\| = (u, u)^{\frac{1}{2}}$).

$\triangleright_1$ Что такое скалярное произведение?

Основное пространство, с которым мы будем работать — пространство вещественнозначных функций, интегрируемых с квадратом — $L_2[a, b]$. Стандартное скалярное произведение в нём определено как

$$(u, v) = \int_a^b u(x)v(x)dx,$$

соответственно порождаемая им норма имеет вид

$$\|u\| = \left(\int_a^b u(x)^2 dx \right)^{\frac{1}{2}}. \tag{9.1}$$

Пусть L — *замкнутое* подпространство в ГП H. Элементом наилучшего приближения (ЭНП) из подпространства L для произвольного $f \in H$ называется такое $\varphi^* \in L$, что

$$\|f - \varphi^*\| = \inf_{\varphi \in L} \|f - \varphi\|.$$

Задача поиска ЭНП в гильбертовом пространстве заключается в нахождении φ^* для данных f и подпространства L с базисом $\{\varphi_i\}_{i=0}^n$.

Норма (9.1) представляет собой усреднённое значение функции в квадрате по всему отрезку $[a, b]$, поэтому наилучшее приближение в этой норме называют также *среднеквадратичным приближением*.

В курсе функционального анализа доказываются следующие важные факты.

- ЭНП существует и единственен.
- ЭНП равен проекции f на L.

Проекцией вектора f из ГП H на подпространство $L \subset H$ называется такой вектор $q^* \in L$, что

$$(f - q^*) \perp L,$$

то есть $(f - q^*, q) = 0 \ \ \forall q \in L$.

Таким образом, алгоритм нахождения ЭНП в произвольном ГП может отличаться от рассмотренного выше случая нахождения проекции вектора на плоскость только способом вычисления скалярного произведения.

9.2.1 Правило вычисления ЭНП в произвольном ГП

Элемент наилучшего приближения φ^* из $(n+1)$-мерного подпространства $L \subset H$ для элемента $f \in H$ имеет вид

$$\varphi^* = \sum_{i=0}^{n} \alpha_i \varphi_i,$$

где $\{\varphi_i\}_{i=0}^n$ — любой базис в L, а коэффициенты $\alpha_i = (\alpha_0, \ldots, \alpha_n)^T$ вычисляются как решение СЛАУ

$$\Gamma \alpha = b, \tag{9.2}$$

$$\Gamma = \begin{bmatrix} (\varphi_0, \varphi_0) & (\varphi_1, \varphi_0) & \dots & (\varphi_n, \varphi_0) \\ (\varphi_0, \varphi_1) & (\varphi_1, \varphi_1) & \dots & (\varphi_n, \varphi_1) \\ \vdots & \vdots & \ddots & \vdots \\ (\varphi_0, \varphi_n) & (\varphi_1, \varphi_n) & \dots & (\varphi_n, \varphi_n) \end{bmatrix}, \quad b = \begin{bmatrix} (f, \varphi_0) \\ (f, \varphi_1) \\ \vdots \\ (f, \varphi_n) \end{bmatrix}. \tag{9.3}$$

Заметим, что при любом базисе матрица Γ симметрична, поэтому для решения СЛАУ с этой матрицей хорошо подойдут методы квадратного корня и сопряжённых градиентов.

9.3 Примеры среднеквадратичных приближений

9.3.1 Среднеквадратичное приближение полиномами

Итак, пускай $L = \mathbb{P}_n$ — подпространство многочленов степени не выше n, $[a, b] = [0, 1]$. Возьмём самый простой базис в этом пространстве

$$\varphi_i(x) = x^i, \quad i = \overline{0, n}, \tag{9.4}$$

и рассмотрим задачу поиска ЭНП в L для $f \in L_2[0, 1]$.

Матрица Грамма Γ для базиса (9.4) вычисляется легко:

$$(\varphi_j, \varphi_i) = \int_0^1 x^{i+j} dx = \frac{1}{i + j + 1},$$

то есть

$$\Gamma = H_{n+1} = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \dots & \frac{1}{n+1} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \dots & \frac{1}{n+2} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \dots & \frac{1}{n+3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{n+1} & \frac{1}{n+2} & \frac{1}{n+3} & \dots & \frac{1}{2n+1} \end{bmatrix}.$$

Это — знаменитая матрица Гильберта, классический пример плохо обусловленной матрицы. Её число обусловленности очень быстро растёт с ростом n, поэтому на практике базис (9.4) применяется крайне редко. В этом

матрица Гильберта схожа со своим аналогом для случая полиномиальной интерполяции — матрицей Вандермонда.

Вместо базиса (9.4) для численно устойчивого вычисления среднеквадратичного приближения полиномами используются различные системы *ортогональных многочленов*, с которыми мы близко познакомимся в следующей лекции.

Формулу для правой части СЛАУ (9.2) мы не выписываем специально — она слишком очевидна (см. (9.3)). Отметим лишь одно: в общем случае входящие в неё интегралы необходимо вычислять приближённо.

9.3.2 Среднеквадратичное приближение сплайнами

Как обычно, для определения пространства сплайнов рассмотрим на отрезке $[a, b]$ разбиение Δ, определяемое набором точек

$$\{a = x_0, x_1, \ldots, x_{M-1}, x_M = b\}.$$

Пусть $L = S_\Delta^m$ — пространство сплайнов порядка m на этом разбиении, размерность этого пространства равна $n = M + m$. Для решения задачи среднеквадратичного приближения в этом пространстве нам очень пригодятся базисные сплайны из предыдущей лекции.

Сплайны порядка 0. Если в качестве базиса взять

$$\varphi_i = N_i^0, \quad i = \overline{0, n-1},$$

то получим, очевидно, $\Gamma = I$, $b_i = \int_{x_i}^{x_{i+1}} f(x)dx$, то есть

$$\varphi^*(x) = \int_{x_i}^{x_{i+1}} f(x)dx, \quad \text{при} \quad x \in [x_i, x_{i+1}).$$

Отметим, что хотя формально мы должны включить в базис ещё и функцию N_n^0, мы этого не делаем, так как она равна нулю на $[a, b]$ везде, кроме $x = b$. Вместо этого мы будем считать, что $N_{n-1}^0(b) = 1$.

Несмотря на примитивность, такой способ приближения вполне имеет право на жизнь и применяется, например, при численном решении интегральных уравнений и уравнений в частных производных.

Сплайны порядка 1 представляют собой уже более совершенный способ приближения. Берём фундаментальный кусочно-линейный базис (8.2):

$$\varphi_i = \psi_i^1, \quad i = \overline{1, n}.$$

Интересно вычислить матрицу Грамма — она будет трёхдиагональной, так как $\operatorname{supp}\psi_i^1 = [x_{i-1}, x_{i+1}] \cap [x_0, x_n]$. Имеем

$$\Gamma = \begin{bmatrix} d_0 & c_1 & 0 & \cdots & 0 \\ c_1 & d_1 & c_2 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & c_{n-1} & d_{n-1} & c_n \\ 0 & 0 & 0 & c_n & d_n \end{bmatrix}, \tag{9.5}$$

где

$$\begin{aligned} &d_i = \frac{h_i + h_{i+1}}{3}, \quad i = \overline{1, n-1}, \quad d_0 = \frac{h_1}{3}, \quad d_n = \frac{h_n}{3}, \\ &c_i = \frac{h_i}{6}, \quad i = \overline{1, n}. \end{aligned} \tag{9.6}$$

Здесь, как и ранее, $h_i = x_i - x_{i-1}$. Построенные формулы важны, их мы будем использовать не только в этом семестре. Для решения СЛАУ (9.2) с матрицей (9.6) можно использовать специально оптимизированный для трёхдиагональных матриц метод квадратного корня.

Вектор правой части b вычисляется согласно (9.3). Понятно, что в силу компактности носителя пределами интегрирования будут соответствующие точки сетки $\{x_i\}$.

Сплайны высших порядков. Среднеквадратичное приближение сплайнами порядка $m \geqslant 2$ осуществляется по аналогии. В качестве базиса можно, конечно, брать любые базисы пространства S_Δ^m, но особенно удобными будут B-сплайны, так как в силу компактности их носителя матрица Грамма будет иметь K ненулевых диагоналей. Для $m = 1$ мы уже получили $K = 3$, для $m = 2$ будет $K = 5$, ну а в общем случае будем иметь $K = 2m + 1$.

Вычисление матрицы Грамма (как и вычисление вектора b) при больших m лучше доверить любому из многочисленных «математических пакетов».

$\triangleright_2$ Вычислите матрицу Грамма для $m = 2$, $x_i = i$, $i = \overline{0, n}$.

9.4 Метод наименьших квадратов

Метод наименьших квадратов, который повсеместно используется для построения непрерывного приближения для дискретного набора точек, является частным случаем задачи поиска ЭНП в ГП.

Постановка задачи

Рассмотрим набор точек $\{(x_i, y_i)\}_{i=0}^N$ и набор линейно независимых базисных функций $\{\varphi_i\}_{i=0}^n$. Метод наименьших квадратов состоит в построении функции φ вида

$$\varphi = \sum_{i=0}^{n} \alpha_i \varphi_i, \quad \alpha_i \in \mathbb{R}, \tag{9.7}$$

которая минимизирует значение функционала

$$\Psi(\varphi) = \sum_{i=0}^{N} (\varphi(x_i) - y_i)^2 \to \min. \tag{9.8}$$

Стандартный способ вывода расчётных формул. Этот способ вывода формул можно найти в практически любом учебнике, поэтому опишем его кратко. Для решения задачи нужно найти $\{\alpha_i\}_{i=0}^n$. После подстановки (9.7) в (9.8) получаем, что

$$\Psi(\varphi) = \Phi(\alpha_0, \ldots, \alpha_n).$$

Минимум этого функционала достигается в стационарной точке $\alpha^* \in \mathbb{R}^{n+1}$, в которой $\dfrac{d\Phi}{d\alpha_k}(\alpha^*) = 0$, $k = \overline{0, n}$. Аккуратно вычисляя частные производные, получаем СЛАУ для нахождения α^*. Всё.

Мгновенный способ вывода расчётных формул. Прежде всего стоит сказать, что мгновенным этот способ становится тогда, когда известна теория, изложенная в пункте 9.2.

Пусть $a = \min\{x_i\}$, $b = \max\{x_i\}$. В пространстве $L_2[a, b]$ определим скалярное произведение как

$$(u, v) = \sum_{i=0}^{N} u(x_i) v(x_i). \tag{9.9}$$

Строго говоря, для того, чтобы формула (9.9) действительно определяла скалярное произведение, необходимо отождествить все функции из $L_2[a, b]$, которые принимают в точках $\{x_i\}_{i=0}^N$ одинаковые значения.

$\triangleright_3$ Почему?

Полученное пространство обозначим $\tilde{L}_2[a, b]$. В такой постановке ординаты $\{y_i\}_{i=0}^N$ однозначно определяют какой-то элемент $f \in \tilde{L}_2[a, b]$. Теперь заметим, что (9.9) порождает норму

$$\|u\| = \left(\sum_{i=0}^{n} u(x_i)^2 \right)^{\frac{1}{2}},$$

так что задача (9.8) становится задачей поиска ЭНП в ГП $\tilde{L}_2[a,b]$ со скалярным произведением (9.9).

Таким образом, решение задачи наименьших квадратов (9.8) *находится по формулам* (9.2), (9.3), *где скалярное произведение определяется формулой* (9.9).

Заметим в заключение, что формально такой подход не определён в случае, когда среди $\{x_i\}$ есть совпадения.

$\triangleright_4$ Что является решением задачи наименьших квадратов в случаях $n = N$ и $n > N$?

10 Ортогональные базисы

10.1 Введение

Задача поиска ЭНП в ГП решается проще всего, когда базис подпространства L выбран таким образом, что матрица Γ в (9.2) диагональна (или просто единична). Этот случай возникает, когда

$$(\varphi_i, \varphi_j) = 0 \quad \text{при} \quad i \neq j,$$

то есть если базис $\{\varphi_i\}$ является *ортогональным*. Напомним, что если

$$(\varphi_i, \varphi_j) = \delta_{ij},$$

то базис называется *ортонормированным*.

В случае ортогонального базиса для ЭНП φ^* имеем очевидное выражение

$$\varphi^* = \varphi_n^* = \sum_{i=0}^{n} \frac{(f, \varphi_i)}{\mu_i} \varphi_i, \tag{10.1}$$

где

$$\mu_i = (\varphi_i, \varphi_i) = \|\varphi_i\|^2. \tag{10.2}$$

Заметим, что если $f \in \operatorname{span}\{\varphi_i\}_{i=0}^n$, то, очевидно, $\varphi^* = f$, или

$$f = \sum_{i=0}^{n} \frac{(f, \varphi_i)}{\mu_i} \varphi_i. \tag{10.3}$$

10.2 Погрешность среднеквадратичного приближения

Пусть для данной $f \in H$ построен ЭНП (10.1) по ортогональной системе $\{\varphi_i\}$. Вычислим величину погрешности

$$e_n = \|f - \varphi_n^*\|. \tag{10.4}$$

Для начала докажем следующее простое утверждение.

Лемма 10.1. *Пусть* $\varphi = \sum_{i=0}^{n} \alpha_i \varphi_i$, *где* $\{\varphi_i\}$ — *ортогональная система векторов,* $\{\alpha_i\}$ — *произвольные коэффициенты. Тогда*

$$\|\varphi\|^2 = \sum_{i=0}^{n} \mu_i \alpha_i^2. \tag{10.5}$$

Доказательство.

$$\|\varphi\|^2 = (\varphi,\varphi) = \Big(\sum_{i=0}^{n}\alpha_i\varphi_i, \sum_{i=0}^{n}\alpha_i\varphi_i\Big) = \sum_{i=0}^{n}\alpha_i\sum_{j=0}^{n}\alpha_j(\varphi_i,\varphi_j) =$$
$$= \sum_{i=0}^{n}\alpha_i^2(\varphi_i,\varphi_i) = \sum_{i=0}^{n}\mu_i\alpha_i^2. \quad \blacksquare$$

Теорема 10.1. *Пусть* $\{\varphi_i\}$ *— ортогональная система векторов,* φ_n^* *— ЭНП для* $f \in H$, *определяемый формулой* (10.1) . *Тогда*

$$e_n^2 = \|f-\varphi_n^*\|^2 = \|f\|^2 - \sum_{i=0}^{n}(f,\varphi_i)^2/\mu_i. \tag{10.6}$$

Доказательство. Обозначим

$$\alpha_i = (f,\varphi_i)/\mu_i, \tag{10.7}$$

тогда $\varphi_n^* = \sum_{i=0}^{n}\alpha_i\varphi_i$ и по лемме 10.1 имеем

$$\|\varphi_n^*\|^2 = \sum_{i=0}^{n}\mu_i\alpha_i^2 = \sum_{i=0}^{n}(f,\varphi_i)^2/\mu_i. \tag{10.8}$$

Распишем e_n:

$$\|f-\varphi_n^*\|^2 = (f-\varphi_n^*, f-\varphi_n^*) = \|f\|^2 - 2(f,\varphi_n^*) + \|\varphi_n^*\|^2 =$$
$$= \|f\|^2 - 2\sum_{i=0}^{n}\alpha_i(f,\varphi_i) + \|\varphi_n^*\|^2 = \big[(10.7),(10.8)\big] = \|f\|^2 - \|\varphi_n^*\|^2 =$$
$$= \|f\|^2 - \sum_{i=0}^{n}(f,\varphi_i)^2/\mu_i. \quad \blacksquare$$

Замечание 10.1. Если φ_n^* сразу задана в виде $\varphi_n^* = \sum\alpha_i\varphi_i$, то вместо (10.6) удобнее использовать эквивалентную формулу

$$\|f-\varphi_n^*\|^2 = \|f\|^2 - \sum_{i=0}^{n}\mu_i\alpha_i^2. \tag{10.6′}$$

10.3 Полные системы векторов

Обычно на практике требуется приблизить функцию $f \in H$ с какой-то заданной погрешностью ε, которая, вообще говоря, может быть сколь угодно

малой. Поэтому наибольший интерес представляют счётные системы базисных функций

$$\{\varphi_i\}_{i=0}^{\infty},$$

которые позволяют сколь угодно точно приблизить любой элемент $f \in H$ их конечной линейной комбинацией. Такие системы векторов называются *полными*.

Система векторов $\{\varphi_i\}_{i=0}^{\infty}$ называется *полной* в H, если множество её конечных линейных комбинаций всюду плотно в H:

$$\forall f \in H \quad \forall \varepsilon > 0 \quad \exists \{\alpha_i\}_{i=0}^{n}, \quad n < \infty \; : \; \left\| f - \sum_{i=0}^{n} \alpha_i \varphi_i \right\| \leqslant \varepsilon.$$

Достаточно очевидно, что если $\{\varphi_i\}_{i=0}^{\infty}$ — полная (не обязательно ортогональная) система в ГП H, то

$$\|f - \varphi_n^*\| \xrightarrow[n\to\infty]{} 0,$$

где φ_n^* — ЭНП для f по системе базисных функций $\{\varphi_i\}_{i=0}^{n}$.

Полные *ортогональные* системы векторов удобны не только тем, что коэффициенты α_i легко вычисляются. Важно ещё и то, что для уточнения приближения φ_n^* нужно вычислить только один дополнительный коэффициент:

$$\varphi_{n+1}^* = \varphi_n^* + \alpha_{n+1}\varphi_{n+1}.$$

В случае ортонормированной системы наилучшие приближения φ_n^* являются частными суммами ряда Фурье .

Таким образом, в дальнейшем нас будут интересовать полные ортогональные (ортонормированные) системы функций в ГП $L_2[a, b]$.

10.4 Классическая тригонометрическая система

Классическую ортогональную систему на отрезке $[-\pi, \pi]$ образуют функции

$$\{1, \cos\cdot, \sin\cdot, \cos 2\cdot, \sin 2\cdot, \ldots\},$$

то есть

$$\varphi_0(x) = 1, \quad \varphi_{2k-1}(x) = \cos kx, \quad \varphi_{2k}(x) = \sin kx, \quad k = \overline{1, \infty}. \qquad (10.9)$$

Тригонометрическая система не является нормированной: $\mu_0 = 2\pi$, $\mu_i = \pi$, $i = \overline{1, \infty}$. Поиск ЭНП для данной системы по формуле (10.1) эквивалентен вычислению частной суммы ряда Фурье по классическим формулам:

$$\varphi_n^*(x) = \frac{\alpha_0}{2} + \sum_{i=1}^{n} \alpha_i \varphi_i(x),$$

где

$$\alpha_0 = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)dx,$$
$$\alpha_{2k-1} = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\cos kx\,dx, \quad \alpha_{2k} = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\sin kx\,dx, \quad k = \overline{1,\infty}.$$

10.5 Ортогональные многочлены

Система многочленов

$$\psi_i(x) = x^i, \quad i = \overline{0,\infty}, \tag{10.10}$$

является полной в $L_2[a,b]$, но, как мы знаем из прошлой лекции, вычисление ЭНП по этой системе является вычислительно неустойчивым.

Поэтому для решения задачи среднеквадратичного приближения многочленами используются ортогональные базисы многочленов. В пространстве $L_2[a,b]$ определим скалярное произведение как

$$(u,v) = \int_a^b u(x)v(x)\rho(x)dx, \tag{10.11}$$

где для корректности необходимо потребовать $\rho(x) \geqslant 0 \ \ \forall x \in [a,b]$ и $\rho(x) = 0$ лишь на множестве меры нуль. Тогда каждая весовая функция ρ будет порождать свою систему ортогональных многочленов. О том, как строятся такие системы, мы сейчас узнаем.

10.5.1 Ортогонализация Грамма–Шмидта

Первый способ построения ортогонального базиса многочленов состоит в ортогонализации системы (10.10) методом Грамма–Шмидта. Этот известный процесс ортогонализации системы векторов из $\mathbb{R}^n$ один-в-один переносится на случай произвольного гильбертова пространства.

Пусть имеется набор линейно независимых векторов $\{\psi_i\}$ и $L = \operatorname{span}\{\psi_i\}$. Наша задача из $\{\psi_i\}$ получить ортонормированный базис для пространства L. Обозначим его $\{g_i\}$.

1. Полагаем $g_0 = \psi_0/\|\psi_0\|$.
2. Строим g_1. Это нужно сделать так, чтобы

   $$\operatorname{span}\{g_0, g_1\} = \operatorname{span}\{\psi_0, \psi_1\},$$

   поэтому найдём вектор $g_1' = cg_0 + \psi_1$ такой, что $(g_0, g_1') = 0$, откуда $c = -(\psi_1, g_0)$. Осталось пронормировать: $g_1 = g_1'/\|g_1'\|$.

3. Предположим, что уже построена ортонормированная система $\{g_0, \ldots, g_{n-1}\}$ такая, что $\operatorname{span}\{g_i\}_0^{n-1} = \operatorname{span}\{\psi_i\}_0^{n-1}$. Построение g_n начинаем с того, что строим

$$g'_n \in \operatorname{span}\{g_0, \ldots, g_{n-1}, \psi_n\} = \sum_{i=0}^{n-1} c_i g_i + \psi_n$$

такой, что $(g'_n, g_i) = 0 \ \ \forall i = \overline{0, n-1}$. Эти условия ортогональности дают

$$c_i = -(\psi_n, g_i), \quad i = \overline{0, n-1}.$$

Наконец, нормируем g'_n: $g_n = g'_n / \|g'_n\|$.

$\triangleright_1$ Постройте таким образом первые три ортогональных многочлена на отрезке $[-1, 1]$ для $\rho \equiv 1$.

Как видим, для построения g_n нужно вычислить $(n+1)$ скалярных произведений (интегралов вида (10.11)), что в общем случае весьма накладно.

10.5.2 Рекуррентные соотношения

В дальнейшем многочлены с единичным старшим коэффициентом условимся называть *унитарными*[2]. Пусть $\{p_i\}_{i=0}^{\infty}$, $\deg p_i = i$, — система многочленов, ортогональных в смысле (10.11). Такая система обладает важным свойством:

$$(p_n, q) = 0 \quad \forall q \in \mathbb{P}_{n-1}. \tag{10.12}$$

$\triangleright_2$ Докажите это очевидное свойство.

Обозначим буквой χ многочлен

$$\chi(x) = x$$

и рассмотрим χp_n,

$$(\chi p_n)(x) = x p_n(x).$$

Так как $\chi p_n \in \mathbb{P}_{n+1}$, имеет место разложение

$$\chi p_n = \sum_{i=0}^{n+1} \alpha_i p_i,$$

где, согласно (10.3),

$$\alpha_i = (\chi p_n, p_i)/\mu_i. \tag{10.13}$$

Здесь, как и ранее, $\mu_i = (p_i, p_i)$. Оказывается, среди коэффициентов α_i будет всего три ненулевых. Покажем это:

$$(\chi p_n, p_i) = \int_a^b x p_n(x) p_i(x) \rho(x) dx = (p_n, \chi p_i) = \big[(10.12)\big] = 0 \quad \forall i \leqslant n-2.$$

[2] Некоторые называют их «моническими», от английского monic, но уж больно коряво звучит эта калька.

Таким образом, любая система ортогональных многочленов последовательно возрастающих степеней удовлетворяет соотношению

$$\chi p_n = \alpha_{n-1} p_{n-1} + \alpha_n p_n + \alpha_{n+1} p_{n+1}. \tag{10.14}$$

Если же предположить, что все $\{p_i\}$ унитарны, то получим $\alpha_{n+1} = 1$, и

$$p_{n+1} = (\chi - a_n) p_n - b_n p_{n-1}, \tag{10.15a}$$

$$a_n = (\chi p_n, p_n)/\mu_n, \quad b_n = (\chi p_n, p_{n-1})/\mu_{n-1}. \tag{10.15b}$$

Очевидно, что $a_n = 0 \ \forall n$ в случае пространства $L_2[-A, A]$ и чётной весовой функции ρ.

Формулы (10.15) позволяют рекуррентно вычислить все $\{p_i\}$, начиная с $p_{-1} \equiv 0$, $p_0 \equiv 1$. Нахождение каждого p_n для любых n требует вычисления всего трёх скалярных произведений.

$\triangleright_3$ Выполните упражнение 10.1, используя формулы (10.15).

10.5.3 Классические ортогональные многочлены

Многочлены Лежандра L_n ортогональны с весом $\rho \equiv 1$ на отрезке $[-1, 1]$. Для них существует явная формула

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

Для такой нормировки справедливо

$$\mu_n = (L_n, L_n) = \frac{2}{2n+1},$$

так что в ЭНП по многочленам Лежандра для $f \in L_2[-1, 1]$ имеет вид

$$\varphi_n^* = \sum_{i=0}^{n} \alpha_i L_i,$$

где

$$\alpha_i = \frac{2i+1}{2} \int_{-1}^{1} f(x) L_i(x) dx.$$

Рекуррентное соотношение для данных многочленов имеет вид

$$(n+1) L_{n+1} - (2n+1) \chi L_n + n L_{n-1} = 0.$$

Многочлены Чебышёва $T_n = \cos n \arccos x$**,** уже знакомые нам, как оказывается, ортогональны с весом

$$\rho(x) = \frac{1}{\sqrt{1-x^2}}$$

на отрезке $[-1, 1]$.

$\triangleright_4$ Докажите это, используя замену $x = \cos\theta$. Вычислите μ_n и запишите явные формулы для ЭНП по многочленам Чебышева.

С остальными классическими ортогональными многочленами мы познакомимся чуть позже.

11 Аппроксимация функций нескольких переменных

11.1 Введение

В этой лекции мы рассмотрим вопрос приближения функций нескольких переменных $f : \mathbb{R}^n \to \mathbb{R}$, который в дальнейшем будем называть также *многомерной аппроксимацией*. И хотя с формальной точки зрения эта задача мало чем отличается от одномерного случая, технически и теоретически она намного сложнее.

Следующая ниже формулировка задачи линейной аппроксимации обобщает практически все рассмотренные нами ранее способы приближения функций одной переменной и естественным образом включает в себя многомерный случай.

Пусть

- $\mathbb{R}^n \supset X$ — компакт,
- F — пространство функций, отображающих X в $\mathbb{R}$ (например, $C(X, \mathbb{R})$ или $L_2(X, \mathbb{R})$),
- $\{\varphi_i\}_{i=0}^n$ — набор линейно независимых функций из F,
- $\mathrm{F} \supset \mathrm{U} = \operatorname{span}\{\varphi_i\}_{i=0}^n$,
- $\{\lambda_i\}_{i=0}^n$ — набор линейно независимых линейных функционалов, определяющих способ приближения, $\lambda_i : \mathrm{F} \to \mathbb{R}$.

Линейной аппроксимацией элемента $f \in \mathrm{F}$ будем называть элемент $\varphi \in \mathrm{U}$, удовлетворяющий *условиям аппроксимации*

$$\lambda_i(\varphi) = \lambda_i(f), \quad i = \overline{0, n}. \tag{11.1}$$

Условия (11.1) определяют систему уравнений для нахождения $\{\alpha_i\}_{i=0}^n$ — коэффициентов разложения φ по базису подпространства U: подставляя
$\varphi = \sum_i \alpha_i \varphi_i$ в (11.1), получаем СЛАУ

$$\sum_{j=0}^{n} \alpha_j \lambda_i(\varphi_j) = \lambda_i(f), \quad i = \overline{0, n}, \tag{11.2}$$

или

$$\Phi\alpha = g, \tag{11.3}$$

где $\Phi = \Big(\lambda_i(\varphi_j)\Big)_{i,j=0}^{n}$, $g_i = \lambda_i(f)$.

Для того, чтобы быстрее подружиться с этим определением, покажем как из него получаются уже знакомые нам методы приближения. Пусть $X = [a, b] \subset \mathbb{R}$, $\{x_i\}_{i=0}^n$ — набор попарно различных узлов интерполяции. Если определить

$$\lambda_i(f) = f(x_i),$$

то задача линейной аппроксимации становится задачей интерполяции по узлам $\{x_i\}$: (11.2) в точности даст систему (1.2). Если же взять $\mathrm{F} = L_2[a, b]$ и положить

$$\lambda_i(f) = (f, \varphi_i),$$

то получим задачу среднеквадратичного приближения (9.2), (9.3)!

Из вышесказанного само собой напрашивается следующее обобщение понятия фундаментального базиса.

Базис $\{\varphi_i\}_{i=0}^n$ будем называть фундаментальным для системы линейных функционалов $\{\lambda_i\}_{i=0}^n$, если

$$\lambda_i(\varphi_j) = \delta_{ij}, \quad i, j = \overline{0, n}.$$

В частности имеем, что ортонормированный базис — фундаментальный базис для задачи среднеквадратичного приближения. Введём ещё одно полезное понятие.

Пусть F — векторное пространство, $\mathrm{F} \supset \mathrm{U}$ — подпространство. Отображение $\Pi : \mathrm{F} \to \mathrm{U}$ называется *проектором* на U, если

$$\Pi(\varphi) = \varphi \quad \forall \varphi \in \mathrm{U},$$

или, что то же самое, $\Pi^2 = \Pi$.

Любой способ линейной аппроксимации порождает проектор

$$\Pi \ : \ \mathrm{F} \ni f \mapsto \varphi \in \mathrm{U},$$

который каждой функции f ставит в соответствие её приближение $\varphi = \Pi f$ согласно данному выше определению:

$$\Pi f = \sum_{i=0}^{n} \alpha_i \varphi_i,$$

где $\{\alpha_i\}$ находится как решение СЛАУ (11.2). Для однозначного определения проектора нужно задать подпространство U и функционалы $\{\lambda_i\}$. Тот факт, что базис в U при этом может быть любым, наверное, легко доказывается.

Таким образом, как в одномерном, так и в многомерном случае линейная аппроксимация функций может быть сведена к решению СЛАУ вида

(11.2). Особенность многомерного случая заключается в том, что 1) размерность системы как правило велика (выше вычислительные затраты) и 2) труднее контролировать невырожденность матрицы Φ. О сходимости мы даже не говорим — сходимость многомерной аппроксимации доказывать намного сложнее, чем одномерной.

Основное внимание в дальнейшем мы уделим наиболее простому случаю, когда многомерная аппроксимация сводится к одномерной — случай аппроксимации тензорными произведениями.

11.2 Линейная аппроксимация тензорными произведениями

Мы рассмотрим лишь случай приближения функций двух переменных, однако этот подход естественным образом может быть обобщён и на случай бóльших размерностей.

11.2.1 Тензорные произведения функций и пространств

Рассмотрим два пространства функций одной переменной: $\mathsf{F}_1 = \mathsf{F}_1(X, \mathbb{R})$ и $\mathsf{F}_2 = \mathsf{F}_2(Y, \mathbb{R})$, $X, Y \subset \mathbb{R}$. Пусть $f_1 \in \mathsf{F}_1$, $f_2 \in \mathsf{F}_2$. *Тензорным произведением* этих функций называется функция $f = f_1 \otimes f_2$,

$$f : X \times Y \to \mathbb{R}, \quad f(x, y) = f_1(x) f_2(y).$$

Соответственно, тензорное произведение пространств F_1 и F_2 это пространство функций двух переменных

$$\mathsf{F}_1 \otimes \mathsf{F}_2 = \operatorname{span}\Big\{ f : X \times Y \to \mathbb{R} \quad | \quad f = f_1 \otimes f_2, \quad f_1 \in \mathsf{F}_1, f_2 \in \mathsf{F}_2 \Big\}.$$

11.2.2 Построение приближения

Пусть в пространстве F_1 задан проектор Π_1, осуществляющий линейную аппроксимацию элементами подпространства

$$\mathsf{U}_1 = \operatorname{span}\{\varphi_i^1\}_{i=0}^{n_1} \subset \mathsf{F}_1$$

по набору функционалов $\{\lambda_i^1\}_{i=0}^{n_1}$. Аналогично в F_2 рассмотрим

$$\Pi_2 : \mathsf{F}_2 \to \mathsf{U}_2 = \operatorname{span}\{\varphi_j^2\}_{j=0}^{n_2}$$

по функционалам $\{\lambda_j^2\}_{j=0}^{n_2}$.

Рассмотрим задачу приближения функций двух переменных из пространства

$$\mathsf{F} = \mathsf{F}(X \times Y, \mathbb{R}).$$

Идея построения приближения для $f \in \mathsf{F}$ с помощью проекторов Π_1 и Π_2 проста: по первой переменной приближаем с помощью Π_1, по второй — с помощью Π_2. Это выглядит следующим образом.

Фиксируем $y \in Y$ и рассматриваем $f_y = f(\cdot, y) \in \mathsf{F}_1$. Тогда для любого $x \in X$

$$f(x, y) \approx \Pi_1 f_y(x) = \sum_{i=0}^{n_1} \alpha_i(y)\varphi_i^1(x) = \tilde{f}(x, y).$$

По определению для любого $y \in Y$ имеем

$$\lambda_i^1(f(\cdot, y)) = \lambda_i^1(\tilde{f}(\cdot, y)), \quad i = \overline{0, n_1}. \tag{11.4}$$

Особо отметим также, что тут все коэффициенты $\alpha_i \in \mathsf{F}_2$.

Теперь фиксируем x: $\tilde{f}_x = \tilde{f}(x, \cdot) \in \mathsf{F}_2$,

$$\begin{aligned}\tilde{f}(x, y) \approx \Pi_2 \tilde{f}_x(y) = \Big(\Pi_2 \sum_{i=0}^{n_1} \alpha_i \varphi_i^1(x)\Big)(y) = \Big(\sum_{i=0}^{n_1} \varphi_i^1(x)\Pi_2\alpha_i\Big)(y) = \\ = \sum_{i=0}^{n_1}\sum_{j=0}^{n_2} \alpha_{ij}\varphi_i^1(x)\varphi_j^2(y) = \sum_{i=0}^{n_1}\sum_{j=0}^{n_2} \alpha_{ij}\varphi_{ij}(x, y) = \varphi(x, y).\end{aligned}$$

Здесь уже $\{\alpha_{ij}\}$ — нормальные скаляры, и при любом $x \in X$ по аналогии имеем

$$\lambda_j^2(\tilde{f}(x, \cdot)) = \lambda_j^2(\varphi(x, \cdot)), \quad j = \overline{0, n_2}. \tag{11.5}$$

Полученная функция φ принадлежит пространству

$$\mathsf{U} = \mathsf{U}_1 \otimes \mathsf{U}_2,$$

а функции

$$\{\varphi_{ij}\}_{i=0,j=0}^{n_1,n_2}, \quad \varphi_{ij} = \varphi_i^1 \otimes \varphi_j^2, \tag{11.6}$$

образуют базис этого пространства.

$\triangleright_1$ Докажите эти два утверждения.

Таким образом мы получили

$$\mathsf{F} \ni f \approx \varphi \in \mathsf{U}.$$

Это была общая идея аппроксимации тензорными произведениями. Для того, чтобы вложить этот способ приближения в общую схему линейной аппроксимации, нужно определить вид функционалов $\{\lambda_{ij}\}$, $\lambda_{ij} : \mathsf{F} \to \mathbb{R}$ на которых выполняется

$$\lambda_{ij}(\varphi) = \lambda_{ij}(f).$$

Нетрудно, конечно, догадаться, что $\lambda_{ij} = \lambda_i^1 \otimes \lambda_j^2$, только что это значит? Сопоставляя формулы (11.4) и (11.5) получаем

$$(\lambda_i^1 \otimes \lambda_j^2)(f) = \lambda_j^2(f_i^2) = \lambda_i^1(f_j^1), \tag{11.7a}$$

где

$$f_i^2 \in \mathsf{F}_2, \quad f_i^2(y) = \lambda_i^1(f(\cdot, y)), \qquad f_j^1 \in \mathsf{F}_1, \quad f_j^1(x) = \lambda_j^2(f(x, \cdot)). \tag{11.7b}$$

Наконец мы готовы дать строгое определение.

Пусть объекты $\mathsf{F} = \mathsf{F}_1 \otimes \mathsf{F}_2$, $\mathsf{U} = \mathsf{U}_1 \otimes \mathsf{U}_2$, $\{\varphi_i^k\}$, $\{\lambda_i^k\}$, $k = 1, 2$, определены как и ранее. Тогда двумерная *линейная аппроксимация тензорными произведениями (ЛАТП)* для $f \in \mathsf{F}$ определяется как обычная линейная аппроксимация элементами подпространства $\mathsf{U} \subset \mathsf{F}$ с базисом $\{\varphi_{ij}\}$ (11.6) и функционалами $\{\lambda_{ij} = \lambda_i^1 \otimes \lambda_j^2\}$ (11.7).

Заметим, что если решать задачу ЛАТП как общую задачу линейной аппроксимации, «в лоб», то условия

$$\lambda_{ij}(\varphi) = \lambda_{ij}(f)$$

дадут СЛАУ с $(n_1+1)(n_2+1)$ неизвестными $\{\alpha_{ij}\}$. Для её решения, скажем, методом Гаусса, при $n_1 = n_2 = n$ понадобится $O(n^6)$ операций. К счастью, если учесть специфику задачи (структуру базисных функций), то вычислительную нагрузку можно существенно снизить.

11.3 Вычисление аппроксимации тензорными произведениями

11.3.1 Тензорное произведение фундаментальных базисов

Самый простой случай, как всегда, — случай фундаментального базиса.

Лемма 11.1. *Тензорное произведение фундаментальных базисов образует фундаментальный базис.*

Доказательство. По условию имеем

$$\lambda_i^1(\varphi_k^1) = \delta_{ik}, \quad \lambda_j^2(\varphi_l^1) = \delta_{jl}.$$

Тогда

$$\lambda_{ij}(\varphi_{kl}) = (\lambda_i^1 \otimes \lambda_j^2)(\varphi_k^1 \otimes \varphi_j^2) = \lambda_i^1(\varphi_k^1)\lambda_j^2(\varphi_l^2) = \delta_{ik}\delta_{jl}.$$

■

Как следствие, получаем чудесную формулу

$$\varphi = \Pi f = \sum_{i=0}^{n_1} \sum_{j=0}^{n_2} \lambda_{ij}(f)\varphi_{ij}. \tag{11.8}$$

Таким образом, в случае фундаментального базиса для построения ЛАТП нужно вычислить только *матрицу значений*

$$F = \Big(\lambda_{ij}(f)\Big)_{i=0,j=0}^{n_1,n_2}. \tag{11.9}$$

11.3.2 Общий случай

Здесь мы продолжим использовать все обозначения, введённые выше. Решение задачи ЛАТП в общем случае сводится к нахождению матрицы коэффициентов

$$A = \big(\alpha_{ij}\big)_{i=0,j=0}^{n_1,n_2}$$

по значениям $\lambda_{ij}(f)$, которые образуют матрицу значений F (11.9).

Для каждого из одномерных способов аппроксимации Π_1 и Π_2 рассмотрим матрицы СЛАУ (11.3)

$$\begin{aligned} \Phi_1 &= \big(\phi^1_{ik}\big)_{i,k=0}^{n_1}, \quad \phi^1_{ik} = \lambda^1_i(\varphi^1_k), \\ \Phi_2 &= \big(\phi^2_{jl}\big)_{j,l=0}^{n_2}, \quad \phi^2_{jl} = \lambda^2_j(\varphi^2_l). \end{aligned} \tag{11.10}$$

Будем считать, что Π_1 и Π_2 определены корректно, то есть Φ_1 и Φ_2 невырождены.

Теорема 11.1.

$$A = \Phi_1^{-1} F (\Phi_2^{-1})^T. \tag{11.11}$$

Доказательство.

$$\begin{aligned} F_{ij} = \lambda_{ij}(f) = \lambda_{ij}(\varphi) = \lambda_{ij}\Big(\sum_k \sum_l \alpha_{kl}\varphi_{kl}\Big) = \sum_k \sum_l \alpha_{kl}\lambda_{ij}(\varphi_{kl}) = \\ = \sum_k \sum_l \alpha_{kl}(\lambda^1_i \otimes \lambda^2_j)(\varphi^1_k \otimes \varphi^2_l) = \sum_k \sum_l \alpha_{kl}\lambda^1_i(\varphi^1_k)\lambda^2_j(\varphi^2_l) = \Big[(11.10)\Big] = \\ = \sum_k \sum_l \alpha_{kl}\phi^1_{ik}\phi^2_{jl} = \sum_l \phi^2_{jl} \underbrace{\sum_k \phi^1_{ik}\alpha_{kl}} = \Big[\Psi = \Phi_1 A\Big] = \sum_l \psi_{il}\phi^2_{jl} = \omega_{ij}, \end{aligned}$$

где ω_{ij} — элемент матрицы $\Omega = \Psi\Phi_2^T = \Phi_1 A \Phi_2^T$. Следовательно,

$$F = \Phi_1 A \Phi_2^T,$$

откуда получаем (11.11). ■

Что нам даёт на практике эта теорема? Во-первых, прямое вычисление A по формуле (11.11) требует обращения двух матриц (при $n_1 = n_2 = n$ это $O(n^3)$ операций), плюс две операции умножения матриц — ещё $O(n^3)$ операций. Итого, сложность вычисления равна $O(n^3)$.

Во-вторых, мы получили удобный алгоритм, позволяющий решать задачу ЛАТП при условии, что мы умеем решать одномерные задачи вычисления приближения $\Pi_1 f$ и $\Pi_2 f$. Согласно формуле (11.3), линейный оператор Φ^{-1} осуществляет преобразование вектора значений функционалов g в вектор коэффициентов α, то есть по сути осуществляет решение одномерной задачи линейной аппроксимации. Это наблюдение позволяет использовать формулу (11.11) даже в тех случаях, когда приближения $\Pi_i f$ вычисляются без прямого построения матриц Φ_i.

Вычислительный алгоритм. Итак, пусть функция `GetAlpha1` находит значения коэффициентов $\{\alpha_i\}_{i=0}^{n_1}$ для приближения

$$\Pi_1 f = \sum_{i=0}^{n_1} \alpha_i \varphi_i^1$$

по входному вектору $g = (\lambda_0^1(f), \ldots, \lambda_{n_1}^1(f))$. Аналогичную функцию для оператора Π_2 назовём `GetAlpha2`. Тогда алгоритм вычисления матрицы коэффициентов A для ЛАТП по формуле (11.11) может быть следующим.

1. Вычисляем матрицу F (11.9), полагаем $A \leftarrow F$.
2. Каждый столбец a_j матрицы A заменяем на `GetAlpha1(` a_j `)`, $\quad j = \overline{0, n_2}$.
3. Каждую строку $\underline{a_i}$ матрицы A заменяем на `GetAlpha2(` $\underline{a_i}$ `)`, $\quad i = \overline{0, n_1}$.

11.4 Полиномиальная интерполяция функций двух переменных на прямоугольнике

Теперь посмотрим как работает всё вышесказанное в случае полиномиальной интерполяции.

На отрезке $X = [a, b]$ как обычно рассматриваем сетку узлов $\{x_i\}_{i=0}^{n_1}$, на отрезке $Y = [c, d]$ — сетку $\{y_j\}_{j=0}^{n_2}$. Наша задача — для функции $f : X \times Y \to \mathbb{R}$ построить многочлен от двух переменных $\varphi = P$ такой, что

$$P(x_i, y_j) = f(x_i, y_j), \qquad i = \overline{0, n_1}, \quad j = \overline{0, n_2}.$$

11.4.1 Интерполяционный многочлен в форме Лагранжа

Так как ИМ в форме Лагранжа для случая одной переменной представляет собой разложение по фундаментальному базису, тут будет работать лекция 1 и формула (11.8).

Имеем $\lambda_i^1(f) = f(x_i)$, $\lambda_j^2(f) = f(y_j)$, то есть

$$\lambda_{ij}(f) = f(x_i, y_j).$$

Теперь базис: $\varphi_i^1(x) = \Lambda_i^1(x) = \prod_{k\neq i} \frac{x-x_k}{x_i-x_k}$, $\varphi_j^2(y) = \Lambda_j^2(y) = \prod_{l\neq j} \frac{y-y_l}{y_j-y_l}$, и

$$\varphi_{ij}(x,y) = \Lambda_{ij}(x,y) = \prod_{k\neq i} \frac{x-x_k}{x_i-x_k} \prod_{l\neq j} \frac{y-y_l}{y_j-y_l}.$$

Собирая всё вместе в формуле (11.8), получаем

$$P(x,y) = \sum_{i=0}^{n_1} \sum_{j=0}^{n_2} f(x_i,y_j)\Lambda_{ij}(x,y). \tag{11.12}$$

11.4.2 Интерполяционный многочлен в форме Ньютона

Здесь уже работает общий случай ЛАТП. Функционалы остаются прежними, а базис другой: $\varphi_i^1(x) = \omega_i^1(x) = \prod_{k=0}^{i-1}(x-x_k)$, $\varphi_j^2(y) = \omega_j^2(y) = \prod_{l=0}^{j-1}(y-y_l)$,

$$\varphi_{ij}(x,y) = \omega_{ij}(x,y) = \prod_{k=0}^{i-1}(x-x_k) \prod_{l=0}^{j-1}(y-y_l).$$

Сам ИМ в форме Ньютона будет иметь вид

$$P(x,y) = \sum_{i=0}^{n_1} \sum_{j=0}^{n_2} \alpha_{ij}\omega_{ij}(x,y). \tag{11.13}$$

Как найти $\{\alpha_{ij}\}$? В одномерном случае (см. лекцию 2) коэффициенты ИМ в форме Ньютона являются разделёнными разностями:

$$\alpha_i = f[x_0,\ldots,x_i].$$

Согласно формуле (11.11) и соответствующему ей вычислительному алгоритму этой информации достаточно, чтобы вычислить неизвестные коэффициенты в формуле (11.13).

$\triangleright_2$ Запишите алгоритм вычисления $\{\alpha_{ij}\}$ в формуле (11.13).

12 Многомерная аппроксимация сплайнами

На прямоугольнике $X \times Y = [x_0, x_{n_1}] \times [y_0, y_{n_2}]$ рассмотрим множество попарно различных узлов

$$\{(x_i, y_j)\}_{i=0,j=0}^{n_1,n_2} = \{x_i\}_{i=0}^{n_1} \times \{y_j\}_{j=0}^{n_2}. \tag{12.1}$$

Отрезок $[x_{i-1}, x_i]$ по традиции обозначаем Δ_i^1, $\Delta_j^2 = [y_{j-1}, y_j]$,

$$\Delta^k = \{\Delta_i^k\}_{i=0}^{n_k}, \quad k = 1, 2.$$

Обозначим также

$$\Delta = \Delta^1 \times \Delta^2 = \{\Delta_{ij} = \Delta_i^1 \times \Delta_j^2\}_{i=0,j=0}^{n_1,n_2}.$$

Пусть на Δ^1 и Δ^2 для некоторого порядка m определены пространства сплайнов $S_{\Delta^1}^m$ и $S_{\Delta^2}^m$ (вообще говоря, порядок не обязан совпадать, но мы для простоты будем считать, что порядок одинаковый).

Рассмотрим, что из себя будет представлять пространство[3]

$$S_{\Delta^1}^m \otimes S_{\Delta^2}^m = S_\Delta^m$$

Его элементами по определению являются всевозможные линейные комбинации функций вида

$$u(x, y) = s^1(x)s^2(y), \qquad s^1 \in S_{\Delta^1}^m, \quad s^2 \in S_{\Delta^2}^m.$$

Какими свойствами обладают такие функции? По построению для $s \in S_\Delta^m$ имеем

$$s|_{\Delta_{ij}} = s_{ij} \in \mathbb{P}_m \otimes \mathbb{P}_m, \tag{12.2}$$

то есть при фиксированном x (или y) функция $s_{ij}(x, \cdot)$ (соответственно $s_{ij}(\cdot, y)$) является многочленом степени m. Что касается гладкости, то так как $s^1 \in C^{m-1}(X)$, $s^2 \in C^{m-1}(Y)$, то все частные производные функции s порядка от 0 до $m-1$ (включая смешанные) будут непрерывными, то есть

$$s \in C^{(m-1)}(X \times Y).$$

Основное внимание в дальнейшем мы уделим задаче интерполяции функций двух переменных элементами пространства S_Δ^m по узлам (12.1)

Сплайн $s \in S_\Delta^m$ называется интерполяционным для $f : X \times Y \to \mathbb{R}$, если

$$s(x_i, y_j) = f(x_i, y_j) := z_{ij}, \qquad i = \overline{0, n_1}, \quad j = \overline{0, n_2}, \tag{12.3}$$

[3] Здесь пространство многомерных сплайнов мы обозначаем S_Δ^m — так же, как и пространство одномерных в лекции 7. По смыслу тут просто другое Δ (не набор отрезков, а набор прямоугольников).

12.1 Билинейный интерполяционный сплайн

Рассмотрим случай $m = 1$.

Функция вида
$$\varphi(x, y) = \alpha_{00} + \alpha_{10}x + \alpha_{01}y + \alpha_{11}xy \tag{12.4}$$
называется *билинейной*.

Сплайн $s \in S^1_\Delta$ представляет собой непрерывную функцию, ограничение которой на Δ_{ij} является билинейной функцией.

Существует два эквивалентных способа построения интерполяционного сплайна $s \in S^1_\Delta$ для функции $f : X \times Y \to \mathbb{R}$.

Способ 1. По определению понятно, что $s|_{\Delta_{ij}} = s_{ij}$ является многочленом вида (12.4), интерполирующим f в точках (x_{i-1}, y_{i-1}), (x_{i-1}, y_i), (x_i, y_{i-1}) и (x_i, y_i). Следовательно, для вычисления s_{ij} можно использовать формулы (11.12), (11.13).

Способ 2. Можно определить s сразу целиком с помощью фундаментального базиса в пространстве S^1_Δ. Такой базис составляют функции ψ^1_{ij},
$$\psi^1_{ij}(x, y) = \psi^1_i(x)\tilde{\psi}^1_j(y),$$
где ψ^1_i и $\tilde{\psi}^1_j$ — фундаментальные сплайны в пространствах $S^1_{\Delta^1}$ и $S^1_{\Delta^2}$ соответственно (формулы (8.2)). Следовательно, согласно общей формуле (11.8), имеем
$$s = \sum_i \sum_j z_{ij}\psi^1_{ij}. \tag{12.5}$$

12.2 Бикубический интерполяционный сплайн

Пусть теперь $m = 3$.

Элементы пространства $S^3_\Delta = S^3_{\Delta^1} \otimes S^3_{\Delta^2}$ называются *бикубическими сплайнами*.

Таким образом, бикубический сплайн s это функция двух переменных из класса $C^2(X \times Y)$ такая, что $s|_{\Delta_{ij}}$ является многочленом третьей степени по каждой переменной.

Совершенно аналогично одномерному случаю, условий интерполяции (12.3) недостаточно для однозначного определения бикубического сплайна.

$\triangleright_1$ Подсчитайте количество недостающих условий.

Поэтому сначала необходимо задать граничные условия. Эти условия как правило порождаются соответствующими одномерными аналогами. Например, естественный бикубический сплайн будет удовлетворять условию
$$\left.\frac{\partial^2 s}{\partial \nu^2}\right|_\Gamma = 0,$$

где Γ — граница прямоугольника $X \times Y$, а ν — внешняя нормаль к Γ.

Опишем два способа построения интерполяционного бикубического сплайна.

Способ 1. Зафиксируем произвольное $y \in Y$. Тогда $s(\cdot, y)$ представляет собой обычный кубический сплайн, который можно записать, как это мы делали в пункте 7.2.1:

$$s(x,y)|_{x\in\Delta_i^1} = \alpha_i(y) + \beta_i(y)(x-x_i) + \frac{\gamma_i(y)}{2}(x-x_i)^2 + \frac{\delta_i(y)}{6}(x-x_i)^3. \quad (12.6)$$

Здесь $\alpha_i, \beta_i, \gamma_i, \delta_i \in S^3_{\Delta^2}$, $i = \overline{1, n_1}$. Таким образом, задача сводится к нахождению $4n_1$ одномерных сплайнов. Необходимые для этого условия получаются из (12.6) подстановкой $y = y_j$, $j = \overline{1, n_2}$. Так мы получим n_2 задач вычисления кубических сплайнов $u_j = s(\cdot, y_j) \in S^3_{\Delta^1}$ для $j = \overline{1, n_2}$:

$$u_j(x)|_{x\in\Delta_i^1} = \alpha_{ij} + \beta_{ij}(x-x_i) + \frac{\gamma_{ij}}{2}(x-x_i)^2 + \frac{\delta_{ij}}{6}(x-x_i)^3, \quad i = \overline{1, n_1}.$$

Условия интерполяции, по которым вычисляются $\{u_j\}_{j=1}^{n_2}$, здесь очевидны:

$$u_j(x_i) = z_{ij}.$$

Граничные условия при этом, конечно, считаются заданными. После этого остаётся вычислить искомые сплайны $\alpha_i, \beta_i, \gamma_i, \delta_i$ по условиям

$$\varkappa_i(y_j) = \varkappa_{ij}, \qquad \varkappa \in \{\alpha, \beta, \gamma, \delta\}, \quad j = \overline{1, n_2}, \quad i = \overline{1, n_1},$$

а также по соответствующим граничным условиям.

$\triangleright_2$ Укажите количество и размерность СЛАУ, которых нужно решить для построения бикубического сплайна таким способом.

Способ 2 является более прямолинейным: строим фундаментальные базисы $\{\psi_i^3\}$ и $\{\tilde\psi_j^3\}$ для пространств $S^3_{\Delta^1}$ и $S^3_{\Delta^2}$ соответственно (см. пункт 8.1.2). После этого бикубический сплайн сразу записывается в виде

$$s = \sum_i \sum_j z_{ij} \psi_i^3(x) \tilde\psi_j^3(y).$$

$\triangleright_3$ Укажите достоинства и недостатки этого способа.

12.3 Бикубический эрмитов сплайн

Как нетрудно догадаться, сейчас мы будем строить обобщение кубического эрмитова сплайна (пункт 7.5) на случай функции двух переменных. Как мы

помним, в одномерном случае построение кубического эрмитова сплайна s осуществляется следующим образом: каждый кусок $s_i \in \mathbb{P}_3$ находится как решение задачи интерполяции с кратными узлами

$$\begin{aligned} s_i(x_{i-1}) &= f(x_{i-1}), \qquad & s_i(x_i) &= f(x_i), \\ s_i'(x_{i-1}) &= f'(x_{i-1}), \qquad & s_i'(x_i) &= f'(x_i). \end{aligned} \tag{12.7}$$

Поэтому нам достаточно понять, как обобщается приближение типа (12.7) на двумерный случай. Для этого нужно вложить эту задачу в общую схему линейной аппроксимации.

12.3.1 Бикубическая эрмитова интерполяция

Случай одной переменной. Рассмотрим оператор

$$\Pi_{a,b} : f \mapsto \varphi \in \mathbb{P}_3,$$

который осуществляет линейную аппроксимацию (11.1) по функционалам

$$\begin{aligned} \lambda_0(f) &= f(a), \qquad & \lambda_2(f) &= f(b), \\ \lambda_1(f) &= f'(a), \qquad & \lambda_3(f) &= f'(b). \end{aligned} \tag{12.8}$$

Тогда, очевидно, (12.7), записывается как $s_i = \Pi_{x_{i-1},x_i} f$. В качестве базиса в $\mathbb{P}_3$ удобно взять базисные функции для эрмитовой интерполяции в форме Ньютона:

$$\begin{aligned} \varphi_0(x) &= 1, \qquad & \varphi_2(x) &= (x-a)^2, \\ \varphi_1(x) &= x-a, \qquad & \varphi_3(x) &= (x-a)^2(x-b). \end{aligned} \tag{12.9}$$

Тогда

$$\Pi_{a,b} f = \sum_{i=0}^{3} \alpha_i \varphi_i,$$

где

$$\alpha = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} f(a) \\ f'(a) \\ f[a,a,b] \\ f[a,a,b,b] \end{bmatrix}.$$

Отображение $\{\lambda_i(f)\}_{i=0}^3 \mapsto \alpha$ обозначим $\Psi_{a,b}$:

$$\Psi_{a,b} \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \end{bmatrix} = \begin{bmatrix} g_0 \\ g_1 \\ (g_2 - g_0 - g_1 h)/h^2 \\ (2(g_0 - g_2) + (g_1 + g_3)h)/h^3 \end{bmatrix}, \tag{12.10}$$

где $h = b - a$. Заметим, что отображение $\Psi_{a,b}$ это не что иное, как `GetAlpha` из вычислительного алгоритма в пункте 11.3.2.

Случай нескольких переменных. Теперь начинаем обобщать: пусть теперь f — функция двух переменных, определённая на $[a,b]\times[c,d]$. Для приближения по первой переменной используем $\Pi_1 = \Pi_{a,b}$, по второй — $\Pi_2 = \Pi_{c,d}$. Соответствующий оператор обозначим

$$\Pi_{[a,b]\times[c,d]} : f \mapsto \varphi.$$

Базисы: $\{\varphi_i^1\} = \{\varphi_i\}$ (по (12.9)), $\{\varphi_j^2\}$ определены так же, только $a \leftarrow c$, $b \leftarrow d$.

Функционалы: $\{\lambda_i^1\} = \{\lambda_i\}$ (по (12.8)), $\{\lambda_j^2\}$ определены так же, только $a \leftarrow c$, $b \leftarrow d$. Теперь самое интересное — вычисляем тензорные произведения функционалов $\lambda_{ij} = \lambda_i^1 \otimes \lambda_j^2$ по формуле (11.7). Имеем

$$\begin{bmatrix} \lambda_{00}(f) & \lambda_{01}(f) & \lambda_{02}(f) & \lambda_{03}(f) \\ \lambda_{10}(f) & \lambda_{11}(f) & \lambda_{12}(f) & \lambda_{13}(f) \\ \lambda_{20}(f) & \lambda_{21}(f) & \lambda_{22}(f) & \lambda_{23}(f) \\ \lambda_{30}(f) & \lambda_{31}(f) & \lambda_{32}(f) & \lambda_{33}(f) \end{bmatrix} = \begin{bmatrix} f(a,c) & \frac{\partial f}{\partial y}(a,c) & f(a,d) & \frac{\partial f}{\partial y}(a,d) \\ \frac{\partial f}{\partial x}(a,c) & \frac{\partial^2 f}{\partial x \partial y}(a,c) & \frac{\partial f}{\partial x}(a,d) & \frac{\partial^2 f}{\partial x \partial y}(a,d) \\ f(b,c) & \frac{\partial f}{\partial y}(b,c) & f(b,d) & \frac{\partial f}{\partial y}(b,d) \\ \frac{\partial f}{\partial x}(b,c) & \frac{\partial^2 f}{\partial x \partial y}(b,c) & \frac{\partial f}{\partial x}(b,d) & \frac{\partial^2 f}{\partial x \partial y}(b,d) \end{bmatrix}. \tag{12.11}$$

Таким образом, алгоритм вычисления приближающей функции

$$\varphi = \Pi_{[a,b]\times[c,d]} f = \sum_{i,j} \alpha_{ij} (\varphi_i^1 \otimes \varphi_j^2)$$

выглядит следующим образом.

1. Вычисляем матрицу F по формуле (12.11). Полагаем $A = F$.
2. Каждый столбец a_j матрицы A заменяем на $\Psi_{a,b}\, a_j$. Это преобразование изменит в матрице только последние две строки.
3. Каждую строку $\underline{a}_i$ у получившейся матрицы заменяем на $(\Psi_{c,d}\, \underline{a}_i^T)^T$. Это преобразование изменит только последние два столбца.

После выполнения этого алгоритма получаем $A = (\alpha_{ij})_{i,j=0}^3$.

12.3.2 Собираем кусочки

Бикубическим эрмитовым сплайном для функции $f : X \times Y \to \mathbb{R}$ на разбиении (12.1) называется функция $s \in C^1(X \times Y)$ такая, что

$$s|_{\Delta_{ij}} = s_{ij} = \Pi_{[x_{i-1},x_i]\times[y_{j-1},y_j]} f.$$

$\triangleright_4$ Оцените сложность построения бикубического эрмитова сплайна в общем случае.

Прежде, чем приступить к изучению нового раздела, хочется сказать пару напутственных слов. Все темы, абсолютно все, с которыми мы будем сталкиваться в дальнейшем, так или иначе будут опираться на аппроксимацию функций, то есть на материал предыдущих 12-ти лекций. Если вы усвоили его — дальше всё будет относительно легко и просто. Если нет — пощады не ждите.

13 Приближённое вычисление интегралов

13.1 Постановка задачи

Рассмотрим задачу вычисления интеграла

$$S(f) = \int_a^b f(x)dx,$$

где f — интегрируемая по Риману функция на отрезке $[a, b]$. Отметим, что отображение S — линейный функционал (отображает функцию в число). Как известно, далеко не всегда $S(f)$ может быть выражен через элементарные функции, поэтому необходимы методы приближённого вычисления интегралов. Вообще говоря, можно воспользоваться определением интеграла как предела интегральных сумм вида

$$\sum_i f(x_i)\Delta x_i,$$

но такой способ как правило сходится очень медленно. Гораздо эффективнее *приблизить подынтегральную функцию некоторой функцией φ, интеграл от которой вычисляется легко, и положить $S(f) \approx S(\varphi)$.*

13.2 Общая схема приближённого вычисления интегралов

Для формулировки общей схемы запишем интеграл в более общем виде

$$S_\rho(f) = \int_a^b f(x)\rho(x)dx, \tag{13.1}$$

где ρ — весовая функция, неотрицательная и ненулевая на $[a, b]$ кроме, может быть, множества меры нуль. В дальнейшем для краткости будем опускать индекс ρ, то есть положим $S = S_\rho$.

Выберем любой способ линейной аппроксимации

$$\Pi : f \mapsto \varphi = \sum_{i=0}^{n} \alpha_i \varphi_i$$

и приблизим с помощью него подынтегральную функцию:

$$S(f) \approx S(\varphi) = S\Big(\sum_{i=0}^{n} \alpha_i \varphi_i\Big) = \sum_{i=0}^{n} \alpha_i S(\varphi_i). \tag{13.2}$$

Интегралы $S(\varphi_i)$, конечно, можно вычислить заранее. Таким образом, *каждый способ линейной аппроксимации функций порождает метод приближённого интегрирования* вида (13.2).

13.3 Квадратурные формулы

Понятно, что основная вычислительная нагрузка при использовании формулы (13.2) заключается в вычислении коэффициентов α_i. Наиболее простой в этом смысле случай имеет место, когда способ приближения П является интерполяцией по точкам $\{x_i\}$, а $\{\varphi_i\}$ — фундаментальный базис. Тогда (13.2) даёт

$$S(f) \approx \sum_{i=0}^{n} f(x_i) S(\varphi_i).$$

В связи с этим введём следующее традиционное определение.

13.3.1 Определение

Функционал Q_n вида

$$Q_n(f) = \sum_{i=0}^{n} A_i f(x_i) \tag{13.3}$$

называется *квадратурной суммой*. Формулы вида

$$S(f) \approx Q_n(f)$$

называются *квадратурными формулами* (КФ). Вещественные числа $\{A_i\}$ называются *коэффициентами*, $\{x_i\}$ — *узлами* квадратурной формулы.

Замечание 13.1. Отметим, что в данном определении ничего не говорится о способе выбора коэффициентов и узлов, то ест формально их значения могут быть любыми и не связанными с каким бы то ни было способом аппроксимации. Таким образом, вид квадратурной формулы в общем случае определяется $2n+2$ параметрами $\{A_i\}_{i=0}^{n}$ и $\{x_i\}_{i=0}^{n}$.

Остатком квадратурной формулы назовём функционал

$$R_n(f) = S(f) - Q_n(f).$$

Модуль остатка будем называть *погрешностью* квадратурной формулы.

13.3.2 Показатели качества КФ

Порядок точности. Интуитивно понятно, что чем меньше длина интервала интегрирования $b-a$, тем точнее будет приближённое значение интеграла $Q_n(f)$. При этом важна скорость убывания погрешности при $b \to a$. Эту скорость характеризует величина, называемая порядком точности.

Говорят, что квадратурная формула имеет *порядок точности*, равный p, если для всех достаточно гладких функций f имеет место разложение

$$R_n(f) = C(b-a)^{p+1} + O((b-a)^{p+q}), \quad q \geqslant 2.$$

Здесь C — константа, зависящая от f.

Степень точности относительно базиса. Другой подход к определению точности квадратурной формулы заключается в следующем. Пусть $\{\psi_i\}_{i=0}^{\infty}$ — полная система функций из пространства F, в котором живут подынтегральные функции f. Для того, чтобы квадратурная формула *точно* работала на всех функциях вида

$$f = \sum_{i=0}^{m} c_i \psi_i,$$

в силу линейности достаточно выполнения условий

$$Q_n(\psi_i) = S(\psi_i), \quad \forall\, i = \overline{0, m}. \tag{13.4a}$$

$\triangleright_1$ Докажите это.

Если квадратурная формула удовлетворяет условиям (13.4a) и

$$Q_n(\psi_{m+1}) \neq S(\psi_{m+1}), \tag{13.4b}$$

то говорят, что её *степень точности относительно базиса* $\{\psi_i\}$ равна m.

Наиболее часто степень точности КФ исследуют относительно полиномиального базиса. Такую степень точности называют алгебраической.

Если квадратурная сумма формула точна для всякого многочлена степени не выше m и не точна хотя бы для одного многочлена степени $m+1$, то говорят, что её *алгебраическая степень точности (АСТ)* равна m.
Это эквивалентно выполнению условий (13.4), где $\{\psi_i\}$ — произвольная система базисных многочленов такая, что $\deg \psi_i = i$. Обычно берут $\psi_i(x) = x^i$.

13.3.3 Обобщённые интерполяционные КФ

Пусть $\{x_i\}_{i=0}^n$ — набор попарно различных узлов на отрезке $[a,b]$, $\{\varphi_i\}_{i=0}^n$ — произвольный фундаментальный базис для этого набора, $\varphi_i(x_j)=\delta_{ij}$. Квадратурную формулу

$$S(f)=\int_a^b f(x)\rho(x)dx \approx \sum_{i=0}^n A_i f(x_i) = Q_n(f). \tag{13.5}$$

где

$$A_i = S(\varphi_i) = \int_a^b \varphi_i(x)\rho(x)dx$$

будем называть *обобщённой интерполяционной квадратурной формулой*.

Наибольшую популярность (в силу исторических причин, в также в силу простоты) получили квадратурные формулы, основанные на полиномиальной интерполяции — именно они гордо называются «интерполяционные квадратурные формулы», то есть так, как будто других способов интерполяции не существует.

13.4 Интерполяционные квадратурные формулы

Пусть $\{x_i\}_{i=0}^n$ — набор попарно различных узлов на отрезке $[a,b]$, $\{\varphi_i\}_{i=0}^n$ — полиномиальный базис Лагранжа: $\varphi_i(x)=\Lambda_i(x)=\prod_{j\neq i}\frac{x-x_j}{x_i-x_j}$. Квадратурные формулы вида (13.5), где

$$A_i = S(\Lambda_i) = \int_a^b \prod_{j\neq i}\frac{x-x_j}{x_i-x_j}\rho(x)dx \tag{13.6}$$

называются *интерполяционными* КФ.

Таким образом, любая интерполяционная КФ однозначно определяется набором узлов $\{x_i\}_{i=0}^n$. Существует критерий, которому удовлетворяют все интерполяционные КФ.

Теорема 13.1. *Квадратурная формула с $(n+1)$ узлом является интерполяционной если и только если она имеет алгебраическую степень точности не менее n.*

Доказательство. $\boxed{\Rightarrow}$ Если КФ является интерполяционной, то $Q_n(f)=S(P_n)$, где P_n — интерполяционный многочлен степени n для f. При $f\in\mathbb{P}_n$ имеем $P_n=f$, то есть $Q_n(f)=S(f)$ — формула точна для всех многочленов степени n и ниже.

$\boxed{\Leftarrow}$ Пусть $Q_n(f)=S(f)\ \ \forall f\in\mathbb{P}_n$. Нужно доказать, что КФ интерполяционная, то есть что её коэффициенты находятся как $A_i=S(\Lambda_i)$, где

Λ_i — фундаментальная базисная функция Лагранжа (см. (13.6)). Нет ничего проще: так как $\Lambda_i \in \mathbb{P}_n$, имеем

$$S(\Lambda_i) = Q_n(\Lambda_i) = \sum_{j=0}^{n} A_j \Lambda_i(x_j) = A_i.$$

■

13.4.1 Остаток интерполяционных КФ

Имея выражение для остатка полиномиального интерполирования (например, формулу (2.14)), легко получить выражение для остатка интерполяционных КФ. Действительно, если нам известна функция r_n,

$$r_n = f - P_n,$$

где P_n — интерполяционный многочлен для f, то

$$R_n(f) = S(f) - Q_n(f) = S(f) - S(P_n) = S(f - P_n) = S(r_n).$$

В частности, используя формулу (2.14)

$$r_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\omega_{n+1}(x),$$

где $\omega_{n+1}(x) = (x - x_0)\dots(x - x_n)$, для достаточно гладких f получаем

$$R_n(f) = \frac{1}{(n+1)!}\int_a^b f^{(n+1)}(\xi)\omega_{n+1}(x)\rho(x)dx. \tag{13.7}$$

Напомним, что здесь величина ξ зависит от x, поэтому острое желание вынести $f^{(n+1)}(\xi)$ за знак интеграла следует, вообще говоря, подавлять. Однако в некоторых случаях это желание осуществимо. Например, при оценке погрешности $|R_n(f)|$ с применением теоремы о среднем получаем

$$|R_n(f)| \leqslant \frac{\|f^{(n+1)}\|}{(n+1)!}\int_a^b |\omega_{n+1}(x)\rho(x)|dx. \tag{13.8}$$

Полученные формулы будут нами активно использоваться в дальнейшем.

13.5 Простейшие интерполяционные КФ

Рассмотрим ряд простейших интерполяционных КФ, получаемых при $\rho \equiv 1$. Как уж упоминалось, для их определения достаточно задать узлы $\{x_i\}$.

13.5.1 Формулы прямоугольников.

Начнём со случая одного узла: $n = 0$. Это соответствует замене подынтегральной функции f на $P_0 \equiv f(x_0)$, то есть интеграл $S(f)$ приближается площадью прямоугольника со сторонами $b - a$ и $f(x_0)$. Соответственно квадратурные формулы будут иметь вид

$$S(f) \approx Q_0(f) = (b - a)f(x_0).$$

Остаток, согласно (13.7), примет вид

$$R_0(f) = \int_a^b f'(\xi)(x - x_0)dx. \tag{13.9}$$

Посмотрим что получится при различных значениях x_0.

Левые прямоугольники. Пусть $x_0 = a$. Соответствующая КФ *левых прямоугольников* имеет вид

$$\int_a^b f(x)dx \approx f(a)(b - a). \tag{13.10}$$

Что касается остатка, то в силу знакопостоянства $(x - x_0)$ при $x_0 = a$ в формуле (13.9) можно применить теорему о среднем. Получается вот что:

$$R_0(f) = f'(\eta)\frac{(b - a)^2}{2}, \quad \eta \in [a, b], \tag{13.11}$$

то есть формула имеет первый порядок точности. Понятно, что её АСТ равна 0.

Правые прямоугольники получаются при $x_0 = b$ и выводятся совершенно аналогично.

$\triangleright_2$ Постройте соответствующие формулы.

Средние прямоугольники — самые интересные. При $x_0 = (a + b)/2$ получаем КФ *средних прямоугольников*

$$\int_a^b f(x)dx \approx f\Big(\frac{a + b}{2}\Big)(b - a). \tag{13.12}$$

Тот факт, что узел расположен посередине отрезка, даёт массу полезных свойств. Во-первых, АСТ этой формулы равна 1 (все многочлены первой степени интегрируются точно) — это легко увидеть геометрически.

Во-вторых, несмотря на то что, что $(x - x_0)$ уже не знакопостоянно, хорошее представление для остатка можно получить следующим образом. Предположим, что f'' существует и непрерывна. Тогда по формуле Тейлора имеем

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(\xi)}{2}(x - x_0)^2, \quad \xi \in [a, b].$$

Вычислим $R_0(f) = S(f - P_0)$, где $P_0 \equiv f(x_0)$, $x_0 = (a + b)/2$:

$$S(f - P_0) = \int_a^b \left(f'(x_0)(x - x_0) + \frac{f''(\xi)}{2}(x - x_0)^2 \right) dx = \\ = \int_a^b \frac{f''(\xi)}{2}(x - x_0)^2 dx = \frac{f''(\eta)}{2} \int_a^b (x - x_0)^2 dx,$$

откуда

$$R_0(f) = f''(\eta)\frac{(b - a)^3}{24}. \tag{13.13}$$

Таким образом, формула имеет второй порядок точности.

13.5.2 Формула трапеций

Перейдём к случаю двух узлов: $n = 1$. Наиболее известная КФ такого типа получается при $x_0 = a$ и $x_1 = b$ — это *формула трапеций*. Геометрический смысл прозрачен: интеграл $S(f)$ приближается площадью трапеции с высотой $b - a$ и основаниями, равными $f(a)$ и $f(b)$. Отсюда сразу имеем

$$\int_a^b f(x)dx \approx \frac{b - a}{2}(f(a) + f(b)). \tag{13.14}$$

Заметим, что то же самое можно получить по общей формуле (13.6).

Запишем формулу для остатка КФ трапеций. Из (13.7) имеем

$$R_1(f) = \frac{1}{2}\int_b^a f''(\xi)(x - a)(x - b)dx.$$

Многочлен $(x - a)(x - b)$ знакопостоянен на $[a, b]$, поэтому снова имеем право применить теорему о среднем:

$$R_1(f) = \frac{f''(\eta)}{2}\int_a^b (x - a)(x - b)dx = -\frac{(b - a)^3}{12}f''(\eta), \quad \eta \in [a, b]. \tag{13.15}$$

Замечание 13.2. При взятии интеграла в данной формуле удобно сделать замену переменных $x = a + (b - a)t$.

Итак, порядок точности формулы трапеций равен двум, а АСТ равна 1. Заметим, что сравнение формул (13.13), (13.15) показывает, что при меньшей трудоёмкости КФ средних прямоугольников в два раза точнее КФ трапеций.

14 Симметричные квадратурные формулы

При построении и анализе формулы средних прямоугольников мы уже могли убедиться в том, что симметричное расположение узлов интегрирования может давать преимущество в виде повышенния АСТ и порядка КФ. Оказывается, в общем случае такие эффекты тоже имеют место.

14.1 Общий случай

Квадратурную формулу

$$S(f) \approx \sum_{i=0}^{n} A_i f(x_i) = Q_n(f)$$

будем называть *симметричной*, если она имеет симметричные узлы относительно середины отрезка $[a, b]$,

$$x_i - a = b - x_{n-i}, \quad \forall i = \overline{0, n},$$

а также симметричные коэффициенты:

$$A_i = A_{n-i}, \quad \forall i = \overline{0, n}.$$

Лемма 14.1. *Пусть весовая функция ρ является чётной относительно середины $[a, b]$, а КФ $S(f) \approx Q_n(f)$ является симметричной. Тогда для всякой нечётной относительно $(a+b)/2$ функции f,*

$$f(x) = -f(a + b - x),$$

справедливо

$$S(f) = Q_n(f).$$

Доказательство. В силу нечётности f имеем $S(f) = 0$, а симметрия узлов даёт $f(x_i) = -f(x_{n-i})$. Покажем, что $Q_n(f) = 0$:

$$Q_n(f) = \sum_{i=0}^{n} A_i f(x_i) = -\sum_{i=0}^{n} A_{n-i} f(x_{n-i}) = -Q_n(f),$$

откуда сразу следует утверждение леммы. ■

Теорема 14.1 (Повышение АСТ для симметричных КФ). *Пусть весовая функция ρ является чётной относительно середины $[a, b]$, а квадратурная формула $S(f) \approx Q_n(f)$ точна для всех многочленов степени $2M$ и ниже,*

$M \in \mathbb{Z}$. Если к тому же эта КФ симметрична, то её алгебраическая степень точности равна как минимум $2M+1$.

Доказательство. Нам достаточно показать, что $Q_n(P) = S(P)$ для любого многочлена P степени $2M+1$. Итак, пусть $\deg P = 2M+1$. Его старший коэффициент обозначим α и рассмотрим многочлен

$$U(x) = \alpha\Big(x - \frac{a+b}{2}\Big)^{2M+1}.$$

В силу нечётности этого многочлена $S(U) = 0$, а также в силу симметричности по лемме 14.1 имеем $Q_n(U) = 0$.

Осталось представить P в виде

$$P = U + \tilde{P},$$

где $\tilde{P} = (P - U) \in \mathbb{P}_{2M}$. Тогда по условию $S(\tilde{P}) = Q_n(\tilde{P})$ и

$$S(P) = S(U) + S(\tilde{P}) = S(\tilde{P}) = Q_n(\tilde{P}) = Q_n(\tilde{P}) + Q_n(U) = Q_n(P). \qquad \blacksquare$$

$\triangleright_1$ Укажите в каком месте доказательства используется чётность ρ.

14.2 Симметрия интерполяционных КФ

14.2.1 Достаточное условие симметричности

Теорема 14.2. *Пусть функция ρ является чётной относительно середины отрезка $[a,b]$.: Тогда если узлы $\{x_i\}$ симметричны, то соответствующая им интерполяционная квадратурная формула с весом ρ является симметричной.*

Доказательство. Другими словами, нам нужно доказать, что если узлы удовлетворяют свойству $x_i - a = b - x_{n-i}$, то коэффициенты соответствующей КФ, которые вычисляются по формуле (13.6),

$$A_i = \int_a^b \Lambda_i(x)\rho(x)dx,$$

обладают свойством $A_i = A_{n-i}$. Доказательство основывается на факте попарной симметричности базисных функций Λ_i:

$$\Lambda_i(x) = \Lambda_{n-i}(a+b-x).$$

Докажем это.

$$\Lambda_{n-i}(a+b-x) = \prod_{j\neq n-i} \frac{a+b-x-x_j}{x_{n-i}-x_j} = \prod_{j\neq n-i} \frac{x_{n-j}-x}{(a+b-x_i)-(a+b-x_{n-j})} =$$
$$= \prod_{j\neq n-i} \frac{x-x_{n-j}}{x_i-x_{n-j}} = \prod_{j\neq i} \frac{x-x_j}{x_i-x_j} = \Lambda_i(x).$$

Осталось сделать замену переменных в интеграле:

$$A_i = \int_a^b \Lambda_i(x)\rho(x)dx = \int_a^b \Lambda_{n-i}(a+b-x)\rho(x)dx = [z = a+b-x] =$$
$$= -\int_b^a \Lambda(z)\rho(a+b-z)dz = \int_a^b \Lambda_{n-i}(z)\rho(z)dz = A_{n-i}. \qquad \blacksquare$$

14.2.2 Случай нечётного числа узлов

Напомним, что по построению интерполяционная КФ с $n+1$ узлом имеет АСТ не ниже чем n. Поэтому для того, чтобы извлечь выгоду из теоремы (14.1), симметричная интерполяционная КФ должна иметь нечётное количество узлов. Понятно, что в этом случае множество узлов содержит середину отрезка $[a, b]$.

Теорема 14.3. *Пусть интерполяционная квадратурная формула имеет нечётное число симметрично расположенных узлов* $\{x_i\}_{i=0}^n$, $n = 2N$, $N \in \mathbb{Z}$, *а весовая функция* ρ *является чётной относительно середины отрезка* $[a, b]$. *Тогда для остатка КФ справедливо представление*

$$R_n(f) = \frac{1}{(n+2)!}\int_a^b f^{(n+2)}(\xi)\Omega_{n+2}(x)\rho(x)dx, \quad \xi \in [a,b], \qquad (14.1)$$

где

$$\Omega_{n+2}(x) = (x - x_N)\prod_{i=0}^{n}(x - x_i), \quad x_N = (a+b)/2.$$

Доказательство. Пусть P_n — интерполяционный многочлен для f по узлам $\{x_i\}_{i=0}^n$, а P_{n+1} — многочлен, интерполирующий f по тем же узлам, но имеющий кратность 2 в центральной точке $x_N = (a+b)/2$:

$$P_{n+1}(x_i) = f(x_i), \quad i = \overline{0,n}, \qquad P'_{n+1}(x_N) = f'(x_N).$$

Тогда $P_{n+1} = P_n + \alpha\,\omega$, где $\omega(x) = \omega_{n+1}(x) = \prod_{i=0}^n (x - x_i)$.

$\triangleright_2$ Чему равно α?

По построению имеем $Q_n(f) = S(P_n)$. Покажем, что $S(P_n) = S(P_{n+1})$. Для этого рассмотрим многочлен ω:

$$\omega(x) = \prod_{i=0}^{n}(x - x_i) = \prod_{i=0}^{n}(x - (a+b-x_{n-i})) = \prod_{i=0}^{n}(x_{n-i} - (a+b-x)) =$$
$$= -\prod_{i=0}^{n}((a+b-x) - x_{n-i}) = -\prod_{i=0}^{n}((a+b-x) - x_i) = -\omega(a+b-x).$$

Таким образом, функция ω является нечётной относительно середины $[a, b]$ — точки x_N. Следовательно, $S(\omega) = 0$ и

$$S(P_{n+1}) = S(P_n + \alpha\,\omega) = S(P_n).$$

Значит,

$$R_n(f) = S(f - P_n) = S(f - P_{n+1}) = S(r_{n+1}),$$

где r_{n+1} — остаток интерполирования с кратными узлами (формула (3.18)):

$$r_{n+1}(x) = \frac{f^{(n+2)}(\xi)}{(n+2)!}\Omega_{n+2}(x) = \frac{f^{(n+2)}(\xi)}{(n+2)!}\omega(x)(x - x_N).$$

Для получения (14.1) осталось проинтегрировать r_{n+1} с весом ρ. ■

$\triangleright_3$ Проверьте этот результат, применив его к КФ средних прямоугольников.

14.3 Квадратурные формулы Ньютона–Котеса

Интерполяционные КФ для весовой функции $\rho \equiv 1$ по равноотстоящим узлам на отрезке $[a, b]$ называются КФ Ньютона–Котеса.

Итак, по определению узлами КФ Ньютона–Котеса являются точки

$$x_i = a + ih, \quad i = \overline{0, n}, \quad h = \frac{b-a}{n}.$$

Значит, все такие КФ симметричны. С одной формулой Ньютона–Котеса мы уже знакомы: при $n = 1$ получается формула трапеций.

14.3.1 Вывод коэффициентов

Теперь выведем общую формулу для коэффициентов КФ Ньютона–Котеса. Согласно общей теории интерполяционных КФ имеем

$$A_i = \int_a^b \prod_{j\neq i} \frac{x - x_j}{x_i - x_j} dx.$$

Сделаем замену переменной: $x = a + th$. Получим

$$A_i = h\int_0^n \prod_{j\neq i} \frac{th - jh}{ih - jh} dt = h\int_0^n \prod_{j\neq i} \frac{t - j}{i - j} dt.$$

Рассмотрим отдельно знаменатель:

$$\prod_{j\neq i}(i - j) = \prod_{j=0}^{i-1}(i - j) \prod_{j=i+1}^{n}(i - j) = (-1)^{n-i}\, i!(n - i)!.$$

Собирая всё вместе и подставляя $h = (b-a)/n$, получим

$$A_i = \frac{(-1)^{n-i}}{i!(n-i)!} \cdot \frac{b-a}{n} \int_0^n \prod_{j \neq i} (t-j)dt, \quad i = \overline{0,n}. \tag{14.2}$$

Итак, коэффициенты КФ Ньютона–Котеса вычисляются по формуле (14.2).

14.3.2 Формула Симпсона

Квадратурная формула Симпсона (формула парабол) это КФ Ньютона–Котеса с тремя узлами: $x_0 = a$, $x_1 = (a+b)/2$, $x_2 = b$. Вычислим коэффициенты этой формулы по формуле (14.2), учитывая, что в силу симметрии $A_0 = A_2$.

$$A_0 = \frac{b-a}{2! \cdot 1! \cdot 2} \int_0^2 (t-1)(t-2)dt = \frac{b-a}{6} = A_2.$$

$$A_1 = -\frac{b-a}{1! \cdot 1! \cdot 2} \int_0^2 t(t-2)dt = \frac{2}{3}(b-a).$$

Таким образом, КФ Симпсона имеет вид

$$\int_a^b f(x)dx \approx \frac{b-a}{6} \left(f(a) + 4f(\tfrac{a+b}{2}) + f(b) \right). \tag{14.3}$$

Исследуем теперь её остаток. Формула имеет нечётное число узлов, поэтому применима теорема 14.3. Получаем

$$R_2(f) = \frac{1}{4!} \int_a^b f^{(4)}(\xi)(x-a)(x-\tfrac{a+b}{2})^2(x-b)dx.$$

По счастливому стечению обстоятельств здесь снова есть возможность применить теорему о среднем, что в итоге даёт

$$R_2(f) = -\frac{f^{(4)}(\eta)}{2880}(b-a)^5. \tag{14.4}$$

Резюмируем: КФ Симпсона имеет АСТ 3 и порядок 4.

$\triangleright_4$ Постройте КФ Ньютона–Котеса с четырьмя узлами.

15 Сходимость квадратурного процесса. Составные квадратурные формулы

15.1 Сходимость квадратурного процесса

При приближённом вычислении интегралов на практике как правило идёт речь о вычислении $S(f)$ с заданной точностью, которая, вообще говоря, может быть любой. Глядя на оценку для остатка интерполяционых КФ (13.8), да и просто из здравого смысла понятно, что *по идее* точность вычисления интеграла должна расти при увеличении количества узлов квадратурной формулы. Сформулируем эти мысли строже по аналогии с пунктом 4.6, где мы рассматривали сходимость интерполяционных процессов.

На отрезке $[a, b]$ рассмотрим бесконечную последовательность сеток из $(n+1)$ попарно различных узлов:

$$\{x_0^{(0)}\},$$
$$\{x_0^{(1)}, x_1^{(1)}\},$$
$$\dots$$
$$\{x_0^{(n)}, x_1^{(n)}, \dots, x_n^{(n)}\},$$
$$\dots$$

Эта последовательность порождает последовательность интерполяционных квадратурных сумм

$$Q_n(f) = \sum_{i=0}^{n} A_i^{(n)} f(x_i^{(n)}),$$

которая для каждой f порождает *квадратурный процесс* — последовательность приближений $\{Q_n(f)\}_{i=0}^{n}$. Если эта последовательность сходится к точному значению интеграла,

$$|S(f) - Q_n(f)| \xrightarrow[n\to\infty]{} 0,$$

то будем говорить, что *квадратурный процесс для функции f сходится*.

Далее мы будем выяснять, каким условиям должны удовлетворять квадратурные суммы $Q_n(f)$ для того, чтобы такая сходимость имела место. Главная мысль, которую следует запомнить, заключается в следующем.

Если все коэффициенты квадратурной формулы положительные, то формула хорошая. Если есть $A_i < 0$ — формула плохая.

Сейчас мы это покажем, с двух разных сторон.

15.1.1 Вычислительная устойчивость квадратурных формул

Рассмотрим сначала вопрос о чувствительности квадратурных формул к погрешностям. Рассмотрим квадратурную сумму

$$Q_n(f) = \sum_{i=0}^{n} A_i f(x_i).$$

Так как коэффициенты $\{A_i\}$ как правило известны заранее, главное влияние на точность результата оказывают погрешности в значениях $\{f(x_i)\}$ — их и будем считать входными параметрами. Возмущённые значения функции обозначим $\{\tilde{f}(x_i)\}$.

Пусть абсолютная погрешность входных параметров ограничена величиной ε:

$$\max_i |f(x_i) - \tilde{f}(x_i)| = \varepsilon.$$

Рассмотрим абсолютную погрешность результата:

$$|Q_n(f) - Q_n(\tilde{f})| = \left|\sum_i A_i (f(x_i) - \tilde{f}(x_i))\right| \leqslant \varepsilon \sum_i |A_i|,$$

причём эта оценка является достижимой. Таким образом, *чем больше сумма модулей коэффициентов квадратурной формулы, тем сильнее её чувствительность к погрешности во входных данных*. Другими словами, при больших значениях $\sum |A_i|$ квадратурная формула является вычислительно неустойчивой.

Теперь предположим, что квадратурная формула является интерполяционной. Значит, её АСТ как минимум равна 0, то есть

$$S(1) = \int_a^b \rho(x)dx = \sum_{i=0}^{n} A_i = \text{Const} \quad \forall n.$$

Мы видим, что *сумма коэффициентов интерполяционной квадратурной формулы постоянна при любом* n. Учитывая очевидное неравенство

$$\sum A_i \leqslant \sum |A_i|,$$

а также что $S(1) > 0$, нетрудно сделать вывод, что наименьшее значение $\sum |A_i|$ равно $S(1)$ и достигается оно в случае, когда все A_i неотрицательны.

Исходя из вышеизложенного,

> квадратурную формулу (сумму) будем называть *вычислительно устойчивой*, если все её коэффициенты A_i положительны.

Теперь перейдём непосредственно к вопросу о сходимости квадратурного процесса.

15.1.2 Банах и Штейнгауз спешат на помощь

Мы хотим узнать, каким условиям должна удовлетворять бесконечная последовательность функционалов $\{Q_n\}_{n=0}^{\infty}$, чтобы соответствующий квадратурный процесс сходился для любой функции из определённого класса. Пусть это будет класс $C[a,b]$.

При каждом фиксированном n функционал Q_n представляет собой линейный оператор, отображающий $C[a,b]$ в $\mathbb{R}$. Условие сходимости

$$|S(f) - Q_n(f)| \xrightarrow[n\to\infty]{} 0, \quad \forall f \in C[a,b], \tag{15.1}$$

представляет собой не что иное, как определение сильной сходимости последовательности линейных операторов.

Теорема Банаха–Штейнгауза — важный результат из функционального анализа, который даёт необходимые условия сильной сходимости. Этих условий два:

1. Оператор S должен быть ограничен, то есть $\|S\| < \infty$.
2. Операторы Q_n должны быть ограничены в совокупности, то есть должна существовать константа $M < \infty$ такая, что $\|Q_n\| < M \quad \forall n \geqslant 0$.

Исходя из этого, найдём нормы операторов S и Q_n.

Начнём с оператора интегрирования S. По определению норма линейного оператора вычисляется как

$$\|S\| = \sup_{\|f\|=1} \|S(f)\|.$$

Нам достаточно показать её ограниченность (это очевидно, но ради строгости посчитаем честно). Пусть $\|f\| = \max\limits_{x\in[a,b]} |f(x)| = 1$. Тогда, учитывая неотрицательность ρ, имеем

$$\|S(f)\| = \left|\int_a^b f(x)\rho(x)dx\right| \leqslant \int_a^b |f(x)|\,\rho(x)dx \leqslant \int_a^b \rho(x)dx < \infty,$$

то есть оператор S ограничен — первое условие теоремы Банаха–Штейнгауза выполнено всегда. Перейдём ко второму условию.

Найдём норму Q_n. Как и ранее, пусть $\|f\| = 1$. Тогда

$$\|Q_n(f)\| = \left|\sum_{i=0}^{n} A_i f(x_i)\right| \leqslant \sum_{i=0}^{n} |A_i|\,|f(x_i)| \leqslant \sum_{i=0}^{n} |A_i|.$$

Здесь для краткости мы опускали верхний индекс в $A_i^{(n)}$. Не правда ли, что-то знакомое? Так как данная оценка является достижимой, получаем

$$\|Q_n\| = \sum_{i=0}^{n} |A_i^{(n)}|.$$

Таким образом, согласно теореме Банаха–Штейнгауза, необходимое условие сходимости квадратурного процесса (15.1) имеет вид

$$\sum_{i=0}^{n} |A_i^{(n)}| \leqslant M < \infty \quad \forall n \geqslant 0. \tag{15.2}$$

На самом деле, в случае интерполяционных квадратурных формул это условие будет и достаточным.

Осталось увязать условие (15.2) с условием устойчивости квадратурных формул из предыдущего пункта. Это легко: если предположить, что все Q_n являются интерполяционными и устойчивыми, то для них верно

$$\sum_{i=0}^{n} |A_i^{(n)}| = \sum_{i=0}^{n} A_i^{(n)} = S(1) < \infty \quad \forall n \geqslant 0.$$

То есть, в указанном случае квадратурный процесс будет сходящимся.

Закрепим результат: *свойство положительности всех коэффициентов интерполяционных квадратурных формул крайне важно как для вычислительной устойчивости, так и для сходимости квадратурного процесса*.

15.1.3 Сходимость формул Ньютона–Котеса

Исходя из всего вышесказанного, проверим качество единственного семейства интерполяционных квадратурных формул, которое мы пока знаем, — формул Ньютона–Котеса. До сих пор все известные нам представители этого класса (формулы трапеций и Симпсона) удовлетворяли условию устойчивости. Однако при больших n доказаны следующие неутешительные результаты:

1. При всех $n \geqslant 10$ среди коэффициентов формул Ньютона–Котеса существуют отрицательные A_i.
2. При $n \to \infty$ среди $\{A_i\}$ будут как положительные, так и отрицательные коэффициенты, по модулю превосходящие любое наперёд заданное число.

Таким образом, при больших n формулы Ньютона–Котеса а) вычислительно неустойчивы и б) вообще могут не сходиться к искомому интегралу.

$\triangleright_1$ Обоснуйте последнее утверждение.

Значит, мы не можем добиться высокой точности численного интегрирования за счёт увеличения числа узлов этих квадратурных формул.

Поэтому широко используется другой способ повышения точности — уменьшение интервалов интегрирования, который приводит к так называемым *составным квадратурным формулам*.

15.2 Составные квадратурные формулы

15.2.1 Построение составных квадратурных формул

Пусть $\rho \equiv 1$. Для вычисления интеграла

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

с требуемой точностью разобьём отрезок $[a, b]$ на N равных частей точками $\{\xi_k\}_{k=0}^N$,

$$\xi_k = a + kh, \quad k = \overline{0, N}, \quad h = \frac{b-a}{N},$$

и на каждом отрезке $[\xi_{k-1}, \xi_k]$ приблизим интеграл какой-нибудь простой квадратурной формулой. Понятно, что таким образом точность приближения интеграла $S(f)$ можно повышать за счёт увеличения N.

Для того, чтобы придать описанному способу общность и математическую строгость, на отрезке $[0, 1]$ рассмотрим *базовую квадратурную формулу*

$$\int_0^1 f(t)dt \approx \sum_{i=0}^n A_i f(t_i) = Q_n(f), \quad t_i \in [0, 1]. \tag{15.3}$$

Тогда на произвольном отрезке $[\alpha, \beta]$ длины h будем иметь

$$\int_\alpha^\beta f(x)dx = h\int_0^1 f(\alpha + ht)dt \approx h\sum_{i=0}^n A_i f(\alpha + ht_i). \tag{15.4}$$

Теперь применим полученную формулу для приближения интеграла от f на отрезке $[\xi_{k-1}, \xi_k]$:

$$S_k(f) = \int_{\xi_{k-1}}^{\xi_k} f(x)dx \approx h\sum_{i=0}^n A_i f(\xi_{k-1} + ht_i) =: Q_{n,k}(f). \tag{15.5}$$

Собирая вместе все такие кусочки, получаем

$$S(f) = \sum_{k=1}^N S_k(f) \approx h\sum_{k=1}^N\sum_{i=0}^n A_i f(\xi_{k-1} + ht_i) = \sum_{k=1}^N Q_{n,k}(f) =: Q_n^N(f). \tag{15.6}$$

Полученное выражение для Q_n^N можно переписать в виде

$$h\sum_{k=1}^N\sum_{i=0}^n A_i f(\xi_{k-1} + t_i h) = h\sum_{i=0}^n A_i \sum_{k=1}^N f(\xi_{k-1} + t_i h) =$$

$$h\sum_{i=0}^n A_i \sum_{k=1}^N f(a + (k-1)h + t_i h) = h\sum_{i=0}^n A_i \sum_{k=0}^{N-1} f\big(a + (t_i + k)h\big).$$

Таким образом, мы построили *составную квадратурную формулу*

$$S(f) \approx Q_n^N(f),$$

или

$$\int_a^b f(x)dx \approx h\sum_{i=0}^{n} A_i \sum_{k=0}^{N-1} f\big(a + (t_i + k)h\big), \tag{15.7}$$

где $h = (b-a)/N$, а $\{A_i\}$ и $\{t_i\}$ — коэффициенты и узлы базовой квадратурной формулы $\int_0^1 f(t)dt \approx \sum_{i=0}^{n} A_i f(t_i)$.

Замечание 15.1. Из полученного представления следует, что составная формула будет устойчивой если и только если базовая КФ устойчива.

Замечание 15.2. Вообще говоря, сетка узлов $\{\xi_i\}$ не обязана быть равномерной. В случае неравномерной сетки будут работать формулы (15.6) с небольшим изменением.

$\triangleright_2$ Сделайте это изменение.

$\triangleright_3$ Объясните, почему нельзя использовать построенные формулы для вычисления интегралов с непостоянным весом ρ?

15.2.2 Остаток составных квадратурных формул

Теперь наша задача — вычислить остаток составных КФ, который обозначим

$$R_n^N(f) = S(f) - Q_n^N(f).$$

Согласно формулам (15.5), (15.6) имеем

$$R_n^N(f) = S(f) - Q_n^N(f) = \sum_{k=1}^{N} S_k(f) - \sum_{k=1}^{N} Q_{n,k}(f) = \\ = \sum_{k=1}^{N} \Big(S_k(f) - Q_{n,k}(f)\Big) = \sum_{k=1}^{N} R_{n,k}(f).$$

Таким образом, как нетрудно было догадаться сразу, *остаток составной квадратурной формулы равен сумме остатков её составных частей*.

Предположим, что для каждой квадратурной суммы $Q_{n,k}$ имеет место представление остатка в виде

$$R_{n,k}(f) = Ch^l f^{(m)}(\eta_k),$$

где $\eta_k \in [\xi_{k-1}, \xi_k]$, C — некоторая константа, l и m — фиксированные натуральные числа. Как мы помним, такая форма остатка справедлива, в частности, для всех уже рассмотренных нами интерполяционных КФ. Тогда для всего остатка получаем очевидное представление

$$R_n^N(f) = \sum_{k=1}^{N} R_{n,k}(f) = Ch^l \sum_{k=1}^{N} f^{(m)}(\eta_k).$$

Эту формулу можно упростить, если предположить, что $f^{(m)}$ непрерывна на $[a, b]$. В этом случае на этом отрезке она достигает своих экстремальных значений $y_{\min}$ и $y_{\max}$. Тогда

$$y_{\min} \leqslant \frac{1}{N} \sum_{k=1}^{N} f^{(m)}(\eta_k) \leqslant y_{\max}.$$

А это означает, что по теореме о промежуточном значении существует такая точка $\eta \in [a, b]$, что

$$f^{(m)}(\eta) = \frac{1}{N} \sum_{k=1}^{N} f^{(m)}(\eta_k).$$

Следовательно, формула для остатка составной КФ примет вид

$$R_n^N(f) = Ch^l N f^{(m)}(\eta), \tag{15.8a}$$

или, с учётом $h = (b-a)/N$,

$$R_n^N(f) = C\frac{(b-a)^l}{N^{l-1}} f^{(m)}(\eta), \tag{15.8b}$$

или, с учётом $N = (b-a)/h$,

$$R_n^N(f) = C(b-a)h^{l-1} f^{(m)}(\eta). \tag{15.8c}$$

Здесь $\eta \in [a, b]$. На практике предпочтительнее использовать последние две формы остатка.

15.2.3 Пример построения составной КФ

Построим составную квадратурную формулу трапеций. Основная формула при этом — (15.7). Предварительно выпишем, чему в нашем случае равны параметры этой формулы. Базовая квадратурная сумма имеет вид

$$Q_1(f) = \frac{1}{2}(f(0) + f(1)),$$

то есть $n = 1$, $t_0 = 0$, $t_1 = 1$, $A_0 = A_1 = \frac{1}{2}$. Подставляем!

$$\int_a^b f(x)dx \approx h \sum_{i=0}^{n} A_i \sum_{k=0}^{N-1} f\big(a + (t_i + k)h\big) =$$

$$= h\Big(A_0 \sum_{k=0}^{N-1} f\big(a + (t_0 + k)h\big) + A_1 \sum_{k=0}^{N-1} f\big(a + (t_1 + k)h\big)\Big) =$$

$$= \frac{h}{2}\Big(\sum_{k=0}^{N-1} f(a+kh) + \sum_{k=0}^{N-1} f\big(a+(k+1)h\big)\Big) = \frac{h}{2}\Big(f(a) + 2\sum_{k=1}^{N-1} f(a+kh) + f(b)\Big).$$

Готово. Итак, составная КФ трапеций имеет вид

$$\int_a^b f(x)dx \approx \frac{h}{2}\Big(f(a) + 2\sum_{k=1}^{N-1} f(a+kh) + f(b)\Big). \tag{15.9}$$

Теперь построим формулу для остатка согласно (15.8). Для этого вспомним, что простая формула трапеций имеет остаток вида

$$R_1(f) = -\frac{(b-a)^3}{12} f''(\eta),$$

то есть $C = -1/12$, $l = 3$, $m = 2$. Тогда из (15.8c) получаем

$$R_1^N(f) = -\frac{b-a}{12} h^2 f''(\eta), \quad \eta \in [a,b]. \tag{15.10}$$

$\triangleright_4$ Постройте составные КФ левых, средних, правых прямоугольников и Симпсона. Запишите формулы для остатка.

15.2.4 Практическая оценка погрешности

Составные квадратурные формулы по построению представляют собой инструмент, позволяющий регулировать точность вычисления интеграла за счёт увеличения числа отрезков разбиения N (уменьшения шага h). Понятно, что для вычислений на практике нужно уметь выбирать такое N, которое бы позволило достичь требуемой точности ε, и в то же время не приводило к избыточным вычислениям (не было слишком велико).

Простейший способ такой оценки — выражения для погрешности (15.8). В частности, из (15.8b) для достаточно гладких f имеем

$$N \geqslant \left(\frac{C(b-a)^l}{\varepsilon} \|f^{(m)}\|_{C[a,b]}\right)^{\frac{1}{l-1}} \quad \Rightarrow \quad |R_n^N| \leqslant \varepsilon.$$

Отметим, что данная оценка является априорной. Этот способ имеет два очевидных недостатка. Во-первых, вычисление $f^{(m)}$ часто является проблематичным, не говоря уже о нахождении нормы этой функции. Во-вторых, даже если $\|f^{(m)}\|_{C[a,b]}$ известна, то оценка такого вида может быть сильно завышенной.

Поэтому чаще применяют другой, апостериорный, способ оценки погрешности, называемый *правилом Рунге*, или *методом двойного пересчёта*.

Правило Рунге. Пусть $Q_n = Q$ — функционал базовой квадратурной формулы. Вычислим приближённое значение интеграла $S(f)$ по соответствующей составной формуле *дважды*, на двух разных разбиениях с числом

отрезков N_1 и N_2, $N_2 > N_1$. Полученные приближения обозначим соответственно

$$q_1 = Q^{N_1}(f) \quad \text{и} \quad q_2 = Q^{N_2}(f).$$

Рассмотрим величину

$$\delta = q_2 - q_1.$$

Идею метода интуитивно можно описать следующим образом. При достаточно больших N_1 и N_2 квадратурная сумма $Q^{N_2}(f)$ *по идее* приближает $S(f)$ «лучше», чем $Q^{N_1}(f)$. Тогда

$$\delta = q_2 - q_1 \approx S(f) - q_1 = R^{N_1}(f).$$

То есть, δ можно рассматривать как своеобразный показатель точности квадратурной суммы $Q^{N_1}(f)$.

Теперь сформулируем метод более строго. Предположим, что для остатка нашей составной КФ справедливо представление

$$R^N(f) = S(f) - Q^N(f) = Kh^p + o(h^p), \quad h = \frac{b-a}{N}, \tag{15.11}$$

где *константа* K *не зависит от* h. Обратите внимание, что такое представление в корне отличается от формул (15.8), в которых величина $f^{(m)}(\eta)$ зависит от h. Таким образом, Kh^p представляет собой *главную часть погрешности*, которую можно найти, если известно значение константы K.

Обозначив $h_i = (b-a)/N_i$, согласно (15.11) можно записать приближённые равенства

$$S(f) - q_1 \approx Kh_1^p,$$
$$S(f) - q_2 \approx Kh_2^p.$$

Отсюда, исключая $S(f)$, имеем

$$K \approx \frac{q_2 - q_1}{h_1^p - h_2^p} = \frac{\delta}{h_1^p - h_2^p} =: \tilde{K},$$

а это очень важная информация. Во-первых, с помощью нее получаем оценку погрешности для обеих квадратурных сумм:

$$R^{N_i}(f) \approx Kh_i^p \approx \tilde{K}h_i^p = \tilde{R}_i,$$

где

$$\tilde{R}_i = \frac{\delta h_i^p}{h_1^p - h_2^p}. \tag{15.12}$$

Во-вторых, имея $\tilde{R}_i$, мы можем уточнить значения q_i:

$$S(f) = q_i + R^{N_i}(f) \approx q_i + \tilde{R}_i. \tag{15.13}$$

Понятно, что для получения максимальной точности лучше всего уточнять q_2.

В-третьих, с помощью $\tilde{K}$ мы имеем возможность оценить оптимальную величину шага h^* для достижения точности ε. Делается это с помощью очевидного требования

$$|R^N(f)| \approx |\tilde{K}|h^p \leqslant \varepsilon,$$

из которого получаем

$$h \leqslant \left(\frac{\varepsilon}{|\tilde{K}|}\right)^{1/p} = \left(\frac{\varepsilon}{|\tilde{R}_i|}\right)^{1/p} h_i = h^*,$$

или

$$N \geqslant N^* = \frac{b-a}{h^*} = \left(\frac{|\tilde{R}_i|}{\varepsilon}\right)^{1/p} N_i. \tag{15.14}$$

Практические рекомендации. Теперь поговорим о том, как всё это реализовывается на практике. Простейший алгоритм выглядит следующим образом.

1. Выбираем базовую квадратурную сумму $Q_n = Q$, а также N_1 и N_2, точность ε.
2. $q_1 \leftarrow Q^{N_1}(f)$, $\quad q_2 \leftarrow Q^{N_2}(f)$.
3. По формуле (15.12) вычисляем оценку погрешности для «более точного» приближения q_2:

   $$\tilde{R} \leftarrow \frac{(q_2 - q_1)h_2^p}{h_1^p - h_2^p}.$$

   Как мы помним, здесь p — показатель точности КФ, определяемый (15.11), $h_i = (b-a)/N_i$.
4. Если $|\tilde{R}| \leqslant \varepsilon$, полагаем $q \leftarrow q_2$ и завершаем алгоритм. В противном случае переходим к следующему шагу.
5. Вычисляем «оптимальное» количество разбиений N^* по формуле (15.14):

   $$N^* \leftarrow \left[\left(\frac{|\tilde{R}|}{\varepsilon}\right)^{1/p} N_2\right] + 1.$$

   Здесь $[\cdot]$ — целая часть числа, а плюс единица — для надёжности.
6. Полагаем $N_1 \leftarrow N_2$, $q_1 \leftarrow q_2$, $N_2 \leftarrow N^*$, $q_2 \leftarrow Q^{N^*}(f)$.
7. Переходим к шагу 3.

Результатом работы алгоритма является значение $q \approx S(f)$. Этот алгоритм можно совершенствовать в следующих направлениях.

- **Экономия вычислений f:** в зависимости от используемой базовой КФ подбирать N_1 и N_2 так, чтобы в соответствующих сетках для вычисления q_1 и q_2 было как можно больше одинаковых узлов. Например, в случае составной формулы трапеций (15.9), обычно берут $N_{i+1} = 2N_i$. Тогда получается

$$Q_1^N(f) = \frac{h}{2}\Big(f(a) + 2\sum_{k=1}^{N-1} f(a+kh) + f(b)\Big),$$

$$Q_1^{2N}(f) = \frac{h}{4}\Big(f(a) + 2\sum_{k=1}^{2N-1} f(a+k\frac{h}{2}) + f(b)\Big) =$$

$$= \frac{1}{2}\left(Q_1^N(f) + h\sum_{k=1}^{N} f(a+(2k-1)\frac{h}{2})\right).$$

  $\triangleright_5$ Постройте соответствующую модификацию алгоритма.

- **Использование уточнённого приближения:** на шаге 4 можно возвращать не q_2, а его уточнённое по формуле (15.13) значение $q_2 + \tilde{R}$.

- **Использование неравномерных сеток** — наиболее радикальное изменение, которое по сути представляет собой совершенно иной алгоритм. Мотивация тут следующая: численное интегрирование по равномерно расположенным узлам является неоптимальным, если функция f на отрезке интегрирования имеет участки с различным поведением (например, сначала сильно осциллирует, а потом изменяется очень медленно). Поэтому современные программы численного интегрирования используют алгоритмы с неравномерно расположенными узлами. При этом принцип оценки погрешности остаётся прежним.

16 Квадратурные формулы наивысшей АСТ

16.1 Постановка задачи

Рассмотрим задачу построения квадратурных формул

$$S(f) = \int_a^b f(x)\rho(x)dx \approx \sum_{i=0}^{n} A_i f(x_i) = Q_n(f),$$

имеющих максимально возможную алебраическую степень точности. Как мы помним, по определению формула имеет АСТ, равную m, если

$$R_n(p) = S(p) - Q_n(p) = 0 \quad \forall p \in \mathbb{P}_m \tag{16.1a}$$

и

$$\exists\, p^* \in \mathbb{P}_{m+1} : \quad R_n(p^*) \neq 0. \tag{16.1b}$$

Мы помним также, что в силу линейности для выполнения условия (16.1a) необходимо и достаточно потребовать, чтобы квадратурная формула работала точно на произвольном базисе пространства $\mathbb{P}_m$. Проще всего выбрать степенной базис, поэтому условия (16.1) эквивалентны (в частности) следующим:

$$\sum_{i=0}^{n} A_i x_i^j = \int_a^b x^j \rho(x)dx, \quad j = \overline{0, m}; \tag{16.2a}$$

$$\sum_{i=0}^{n} A_i x_i^{m+1} \neq \int_a^b x^{m+1} \rho(x)dx. \tag{16.2b}$$

В дальнейшем мы будем рассматривать только условия (16.2a).

Какова же будет максимально возможная АСТ для данного n? Условия (16.2) представляют собой систему из $(m+1)$ нелинейных уравнений относительно $(2n+2)$ неизвестных $\{x_i\}_{i=0}^n$, $\{A_i\}_{i=0}^n$. Поэтому можно *по крайней мере надеяться*, что эта система будет иметь решение когда количество уравнений и неизвестных совпадает, то есть когда

$$m = 2n + 1.$$

Как мы увидим далее, это и есть максимально возможная АСТ (*наивысшая АСТ*, или *НАСТ*).

16.2 Построение квадратурных формул НАСТ

Итак, построение квадратурных формул НАСТ, чаще называемых КФ Гаусса, заключается в нахождении узлов $\{x_i\}$ и коэффициентов $\{A_i\}$, удовлетворяющих условиям (16.2a). Поэтому самый тривиальный способ их нахождения это решение указанной системы «в лоб». Для каждого n мы

будем иметь $2n+2$ нелинейных уравнений. Случай $n=0$ прост, при $n=1$ систему можно решить относительно легко, а при бóльших значениях n задача становится практически неподъёмной. То есть, для нас это не способ.

Следующее тривиальное следствие теоремы 13.1 существенно упрощает нашу задачу: *если квадратурная формула имеет наивысшую АСТ, то она является интерполяционной, то есть её коэффициенты имеют вид*

$$A_i = \int_a^b \prod_{j\neq i} \frac{x-x_j}{x_i-x_j}\rho(x)dx, \quad i=\overline{0,n}.$$

$\triangleright_1$ Докажите.

Таким образом задача *упростилась в два раза*: достаточно найти узлы КФ НАСТ, а коэффициенты можно определить по знакомой нам формуле, приведённой выше.

А вот о том, какими должны быть узлы формулы, говорит следующая теорема. Она является наиболее важным результатом всей лекции.

Теорема 16.1 (критерий КФ НАСТ, К. Ф. Гаусс). *Для того, чтобы квадратурная формула с $(n+1)$ узлами была точной для любых алгебраических многочленов степени $(2n+1)$ и ниже, необходимо и достаточно, чтобы*

1) квадратурная формула была интерполяционной и

2) многочлен $\omega_{n+1}(x)=(x-x_0)\dots(x-x_n)$ был ортогонален по весу ρ на отрезке $[a,b]$ ко всем многочленам степени n и ниже:

$$(\omega_{n+1},p)=\int_a^b \omega_{n+1}(x)p(x)\rho(x)dx=0 \quad \forall p\in\mathbb{P}_n. \tag{16.3}$$

Доказательство. $\boxed{\Rightarrow}$ Пусть формула имеет АСТ $2n+1$. Значит, она будет точна для всех многочленов вида $q(x)=\omega_{n+1}(x)p(x)$, $p\in\mathbb{P}_n$, так как $\deg q\leqslant 2n+1$. С учётом того, что $\omega_{n+1}(x_i)=0$ при всех i, получаем

$$S(q)=(\omega_{n+1},p)=Q_n(q)=\sum_{i=0}^{n} A_i\omega_{n+1}(x_i)p(x_i)=0.$$

То, что КФ интерполяционная, мы уже доказывали.

$\boxed{\Leftarrow}$ Пусть верно (16.3). Возьмём любой $q\in\mathbb{P}_{2n+1}$ и разделим его на ω_{n+1} с остатком:

$$q(x)=\omega_{n+1}(x)p(x)+r(x),$$

где $\deg p\leqslant n$, $\deg r\leqslant n$. Тогда с учётом $q(x_i)=r(x_i)$ имеем

$$S(q)=(\omega_{n+1},p)+S(r)=S(r)=\sum_{i=0}^{n} A_i r(x_i)=\sum_{i=0}^{n} A_i q(x_i),$$

то есть формула точна для любого $q \in \mathbb{P}_{2n+1}$. ■

Таким образом, *квадратурная формула НАСТ это интерполяционная КФ с узлами — корнями многочлена, ортогонального по весу ρ всем многочленам степени n и ниже.* Это прекрасный результат, но для строгости его доказательства нужно а) показать, что нельзя достичь АСТ, большей $2n+1$ и б) доказать существование узлов КФ НАСТ на отрезке $[a,b]$ (а вдруг они вообще комплексные?). Этим и займёмся.

Теорема 16.2. *Если весовая функция ρ знакопостоянна на отрезке $[a,b]$, то не существует квадратурных формул с $(n+1)$ узлами, имеющих АСТ $2n+2$.*

Доказательство. Предположим существование такой формулы. Пусть $\{x_i\}_{i=0}^n$ — её узлы. Рассмотрим многочлен

$$\nu(x) = \omega_{n+1}(x)^2 = (x-x_0)^2 \dots (x-x_n)^2.$$

Для него, очевидно, имеем $S(\nu) \neq 0$, так как подынтегральное выражение знакопостоянно и не равно нулю тождественно. С другой стороны имеем $Q_n(f) = \sum_{i=0}^n A_i \nu(x_i) = 0$. Полученное противоречие доказывает теорему. ■

Теорема 16.3. *Если весовая функция ρ сохраняет знак на отрезке $[a,b]$, то многочлен ω_{n+1} степени $(n+1)$, ортогональный на данном отрезке по весу ρ ко всем многочленам степени n и ниже, существует и единственен для любого фиксированного n. При этом все его корни действительны, различны и лежат внутри $[a,b]$.*

Доказательство. Существование многочлена ω_{n+1} следует из пункта 10.5, в котором мы рассматривали ортогональные многочлены. Понятно, что ω_{n+1} это $(n+1)$-й элемент системы ортогональных многочленов на отрезке $[a,b]$ с весом ρ, пронормированный так, чтобы старший коэффициент был равен 1. Значит, его всегда можно построить либо методом Грамма–Шмидта, либо по рекуррентным соотношениям (10.15а).

Докажем единственность, для краткости обозначив $\omega_{n+1} = \omega$. Пусть существует многочлен с единичным старшим коэффициентом $\omega^* \neq \omega$, $\deg \omega^* = n+1$, такой что

$$(\omega^*, p) = 0 \quad \forall p \in \mathbb{P}_n.$$

Тогда

$$(\omega - \omega^*, p) = 0 \quad \forall p \in \mathbb{P}_n.$$

А так как $\omega - \omega^* \in \mathbb{P}_n$, получаем $\omega - \omega^* = 0$, что и требовалось.

$\triangleright_2$ Обоснуйте последнее утверждение.

Докажем теперь, что все корни многочлена ω вещественны, различны и лежат в $[a,b]$. Для начала покажем, что на $[a,b]$ существует по крайней

мере один корень ξ_0, имеющий нечётную кратность: по условию $(\omega, 1) = 0$, то есть

$$\int_a^b \omega(x)\rho(x)dx = 0,$$

значит на $[a, b]$ обязана быть как минимум одна точка ξ_0, в которой ω меняет знак, то есть корень нечётной кратности.

Пусть теперь $\{\xi_0, \dots \xi_m\}$ — множество всех корней ω, принадлежащих $[a, b]$ и имеющих нечётную кратность. Если мы покажем, что $m = n$, то теорема будет доказана. Предположим, что $m < n$. Тогда для многочлена

$$q(x) = (x - \xi_0) \dots (x - \xi_m), \quad \deg q = m \leqslant n,$$

по условию имеем

$$(\omega, q) = \int_a^b \omega(x)q(x)\rho(x)dx = 0.$$

А это невозможно, потому что по предположению все корни многочлена ωq, принадлежащие отрезку $[a, b]$, имеют чётную кратность, что означает его знакопостоянство на $[a, b]$. ■

16.3 Снова ортогональные многочлены

Вычисление коэффициентов формул НАСТ «обычным» способом

$$A_i = \int_a^b \prod_{j \neq i} \frac{x - x_j}{x_i - x_j} \rho(x) dx, \quad i = \overline{0, n}.$$

при больших n может представлять определённые сложности. Сейчас мы выведем другую формулу, не требующую интегрирования. Но для этого нужно *вспомнить всё* об ортогональных многочленах.

16.3.1 Flashback

Итак (см. пункт 10.5.2), пусть $\{p_i\}_{i=0}^{\infty}$ — система ортогональных многочленов на отрезке $[a, b]$ с весом ρ, причём $\deg p_i = i$. Попытаемся ещё раз разобраться с рекуррентными соотношениями (10.14), (10.15a).

Важно понимать, что p_n ортогонален всем многочленам степени меньше n:

$$(p_n, q) = 0 \quad \forall\, q \in \mathbb{P}_{n-1}, \tag{16.4}$$

так как такой q всегда можно разложить по базису $\{p_i\}_{i=0}^{n-1}$, а $(p_n, p_i) = 0$ при $i \neq n$. Без осознания этого факта нельзя идти дальше.

Обозначим теперь $P_i = p_i / \|p_i\|$, то есть система $\{P_i\}_{i=0}^{\infty}$ ортонормирована: $(P_i, P_j) = \delta_{ij}$. Зафиксируем n и рассмотрим многочлен $V_{n+1}(x) = xP_n(x)$. Так как $V_{n+1} \in \mathbb{P}_{n+1}$, он раскладывается по базису $\{P_i\}_{i=0}^{n+1}$:

$$V_{n+1}(x) = \sum_{i=0}^{n+1} \alpha_i^n P_i(x), \quad \alpha_i^n = (V_{n+1}, P_i).$$

Рассмотрим коэффициент α_i^n (индекс вверху обязан присутствовать, так как для разных n значения i-го коэффициента, естественно, не совпадают):

$$\alpha_i^n = (V_{n+1}, P_i) = \int_a^b xP_n(x)P_i(x)\rho(x)dx = \int_a^b P_n(x)(xP_i(x))\rho(x)dx = (P_n, V_{i+1}),$$

$V_{i+1}(x) = xP_i(x)$. В силу (16.4) имеем

$$\alpha_i^n = 0 \quad \text{если} \quad i + 1 < n,$$

то есть

$$xP_n(x) = \alpha_{n-1}^n P_{n-1}(x) + \alpha_n^n P_n(x) + \alpha_{n+1}^n P_{n+1}(x), \tag{16.5a}$$

$$\alpha_i^n = \int_a^b xP_n(x)P_i(x)dx \tag{16.5b}$$

(сравните с (10.14)). Заметим, что эти соотношения нельзя непосредственно использовать для последовательного нахождения ортогональных многочленов, так как коэффициент α_{n+1}^n выражается через неизвестный многочлен P_{n+1}. А вот если взять другую нормировку, как мы делали в пункте (10.5.2), то будем иметь $\alpha_{n+1}^n = 1 \ \forall\, n$, и соответственно получим формулы (10.15).

16.3.2 Тождество Кристоффеля–Дарбу

Заменим в (16.5a) n на i и умножим на $P_i(y)$:

$$xP_i(x)P_i(y) = \alpha_{i-1}^i P_{i-1}(x)P_i(y) + \alpha_i^i P_i(x)P_i(y) + \alpha_{i+1}^i P_{i+1}(x)P_i(y).$$

Теперь перепишем эту формулу, поменяв местами x и y:

$$yP_i(x)P_i(y) = \alpha_{i-1}^i P_i(x)P_{i-1}(y) + \alpha_i^i P_i(x)P_i(y) + \alpha_{i+1}^i P_i(x)P_{i+1}(y).$$

Вычитая одну формулу из другой, получаем

$$\begin{aligned}(x-y)P_i(x)P_i(y) = \alpha_{i-1}^i \Big(\underbrace{P_{i-1}(x)P_i(y) - P_i(x)P_{i-1}(y)}_{U_i(x,y)}\Big) + \\ + \alpha_{i+1}^i \Big(P_{i+1}(x)P_i(y) - P_i(x)P_{i+1}(y)\Big) = \\ = \Big[\alpha_{i+1}^i = \alpha_i^{i+1}\Big] = \alpha_{i-1}^i U_i(x,y) - \alpha_i^{i+1} U_{i+1}(x,y).\end{aligned}$$

Вот. Теперь просуммируем это по всем i от 0 до n, вспомним, что $P_{-1} \equiv 0$ и получим важное *тождество Кристоффеля–Дарбу*, справедливое для любых систем ортонормированных многочленов:

$$\boxed{\sum_{i=0}^{n} P_i(x)P_i(y) = \alpha_{n+1}^n \frac{P_{n+1}(x)P_n(y) - P_n(x)P_{n+1}(y)}{x-y},} \tag{16.6}$$

где $\alpha_{n+1}^n = \int_a^b xP_n(x)P_{n+1}(x)\rho(x)dx$.

16.3.3 Коэффициенты КФ НАСТ

Итак, по определению имеем

$$A_i = \int_a^b \Lambda_i(x)\rho(x)dx, \quad i = \overline{0,n},$$

где Λ_i — базисный многочлен Лагранжа, который имеет вид (см. (1.7), (1.10))

$$\Lambda_i = \prod_{j\neq i} \frac{x-x_j}{x_i-x_j} = \frac{\omega_{n+1}(x)}{(x-x_i)\omega'_{n+1}(x_i)}.$$

Так как узлы $\{x_i\}_{i=0}^n$ КФ НАСТ являются корнями многочлена P_{n+1}, то

$$\omega_{n+1} = \frac{1}{c_{n+1}} P_{n+1}, \tag{16.7}$$

где c_{n+1} — старший коэффициент многочлена P_{n+1}. Возьмём в (16.6) $y = x_i$ и проинтегрируем с весом:

$$\sum_{k=0}^{n} \int_a^b P_k(x_i) P_k(x) \rho(x) dx = \alpha_{n+1}^n \int_a^b \frac{P_{n+1}(x) P_n(x_i) - P_n(x) P_{n+1}(x_i)}{x - x_i} \rho(x) dx.$$

Выражение слева равно $(P_0, P_0) = 1$ при $k = 0$, и нулю в остальных случаях. В интеграле справа по построению имеем $P_{n+1}(x_i) = 0$. Поэтому формула принимает вид

$$1 = \alpha_{n+1}^n P_n(x_i) \int_a^b \frac{P_{n+1}(x)}{x - x_i} \rho(x) dx. \tag{16.8}$$

Собираем всё вместе:

$$A_i = \int_a^b \frac{\omega_{n+1}(x)}{(x - x_i)\omega'_{n+1}(x_i)} \rho(x) dx \overset{(16.7)}{=} \frac{1}{P'_{n+1}(x_i)} \int_a^b \frac{P_{n+1}(x)}{x - x_i} \rho(x) dx \overset{(16.8)}{=}$$
$$= \frac{1}{\alpha_{n+1}^n P_n(x_i) P'_{n+1}(x_i)} = \frac{c_{n+1}}{c_n} \frac{1}{P_n(x_i) P'_{n+1}(x_i)}.$$

В последнем равенстве из этой цепочки мы использовали равенство

$$\alpha_{n+1}^n = \frac{c_n}{c_{n+1}},$$

которое получается из (16.5a) приравниванием коэффициентов при старшей степени.

Таким образом, *если известны n-й и $(n+1)$-й элементы ортонормированной системы многочленов $\{P_i\}_{i=0}^\infty$ с весом ρ, то узлы $\{x_i\}_{i=0}^n$ соответствующей КФ НАСТ являются корнями многочлена P_{n+1}, а коэффициенты можно вычислить по формуле*

$$\boxed{A_i = \frac{c_{n+1}}{c_n} \frac{1}{P_n(x_i) P'_{n+1}(x_i)}, \quad i = \overline{0, n},} \tag{16.9}$$

где c_i — старший коэффициент многочлена P_i.

Для использования формулы (16.9) нужно иметь ортонормированные многочлены P_i, которые, в принципе, можно найти ортогонализацией Грамма–Шмидта. Но при больших n это не эффективно, и лучше воспользоваться рекуррентными соотношениями (10.15), которые генерируют ортогональные многочлены $\{p_i\}$ с единичным старшим коэффициентом (да-да, $p_i = \omega_i$).

То есть, если нам известны p_n и p_{n+1} (они же ω_n и ω_{n+1} соответственно), то (16.9) с учётом $P_i = c_i p_i$ превращается в

$$A_i = \frac{1}{c_n^2 p_n(x_i) p'_{n+1}(x_i)},$$

откуда с учётом $1 = \|P_i\|^2 = c_i^2 \|p_i\|^2$ окончательно получаем

$$\boxed{A_i = \frac{\|p_n\|^2}{p_n(x_i) p'_{n+1}(x_i)}, \quad i = \overline{0, n}.} \tag{16.10}$$

16.3.4 Третий способ построения ортогональных многочленов

До сих пор мы рассматривали два способа построения ортогональных многочленов: ортогонализация Грамма-Шмидта и рекуррентные соотношения. Эти способы имеют одну общую черту: для нахождения многочлена степени n необходимо сначала найти все предыдущие.

При малых n можно использовать ещё один подход, самый бесхитростный из всех. Он позволяет для данного n найти непосредственно унитарный многочлен p_n в явном виде

$$p_n(x) = x^n + a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \ldots + a_1 x + a_0.$$

Для этого просто воспользуемся условиями (16.4), последовательно положив $q(x) = x^i$:

$$\int_a^b p_n(x) x^i \rho(x) dx = 0, \quad i = \overline{0, n-1}.$$

Эти условия дают СЛАУ вида

$$Ha = -g,$$

для нахождения вектора коэффициентов $a = (a_0, a_1, \ldots, a_{n-1})^T$, где $H = (h_{ij})_{i,j=0}^{n-1}$ — обобщённая матрица Гильберта, $h_{ij} = \int_a^b x^{i+j}\rho(x)dx$, $g = (g_0, g_1, \ldots g_{n-1})^T$, $g_i = \int_a^b x^{n+i}\rho(x)dx$.

16.3.5 Классические ортогональные многочлены

Для некоторых весовых функций известны явные формулы, определяющие ортогональные многочлены, а также явные рекуррентные соотношения. Для этих многочленов давным-давно составлены таблицы коэффициентов и узлов соответствующих КФ НАСТ. Соответствующие формулы выводить слишком долго, а запоминать бессмысленно. Но нужно о них знать, чтобы при необходимости найти в литературе.

$\rho(x)$	Отрезок	Название системы многочленов
1	$[-1,1]$	Лежандра
$\dfrac{1}{\sqrt{1-x^2}}$	$[-1,1]$	Чебышева
$x^\alpha e^{-x}$	$[0,+\infty)$	Лагерра
e^{-x^2}	$(-\infty,+\infty)$	Эрмита
$(1-x)^\alpha(1+x)^\beta$	$[-1,1]$	Якоби

16.4 Остаток КФ НАСТ

Теорема 16.4. *Для достаточно гладких функций f остаток КФ НАСТ с узлами $\{x_i\}_{i=0}^n$ и весом ρ имеет вид*

$$\boxed{S(f)-Q_n(f)=R_n(f)=\frac{f^{(2n+2)}(\eta)}{(2n+2)!}\,\|\omega_{n+1}\|^2,} \qquad (16.11)$$

где как обычно $\omega_{n+1}=(x-x_0)\dots(x-x_n)$, η — какая-то точка из $[a,b]$, $\|\cdot\|$ — норма в $L_2[a,b]$.

Доказательство. Строим для f интерполяционный многочлен H_{2n+1} с двукратными узлами $\{x_i\}_{i=0}^n$. Для погрешности согласно (3.18) имеем

$$r_{2n+1}(x)=f(x)-H_{2n+1}(x)=\frac{f^{(2n+2)}(\xi)}{(2n+2)!}(x-x_0)^2\dots(x-x_n)^2,\quad \xi\in[a,b].$$

Так как рассматриваемая КФ имеет наивысшую АСТ, имеем $Q_n(H_{2n+1})=S(H_{2n+1})$. Поэтому

$$S(f)-S(H_{2n+1})=S(f)-Q_n(H_{2n+1})=S(f)-Q_n(f)=S(r_{2n+1}),$$

то есть

$$R_n(f)=S(f)-Q_n(f)=\int_a^b\frac{f^{(2n+2)}(\eta)}{(2n+2)!}\,\omega_{n+1}(x)^2\rho(x)dx,$$

откуда применяя теорему о среднем получаем (16.11). ■

17 Вычисление кратных интегралов

17.1 Введение

Рассмотрим задачу приближённого вычисления m-кратного интеграла

$$S(f) = \int_\Omega f(x)\rho(x)dx,$$

где $\mathbb{R}^m \supset \Omega$ — некоторая область m-мерного пространства, $f : \Omega \to \mathbb{R}$, $\rho : \Omega \to \mathbb{R}$ — весовая функция, удовлетворяющая условиям, аналогичным одномерному случаю, $x = (x_1, \ldots, x_m)$.

Существует два основных способа приближённого вычисления таких интегралов.

1. Сведение интеграла по области Ω к повторному интегралу. Это приводит к цепочке однократных интегралов, которые вычисляются с помощью квадратурных формул.
2. Построение специализированных формул для многомерного случая.

Оба эти способа приводят к так называемым *кубатурным* формулам, которые отличаются от квадратурных лишь множеством определения функции f:

$$S(f) \approx Q(f) = \sum_{i=0}^{n} A_i f(x^i), \tag{17.1}$$

где $\mathbb{R} \ni A_i$ — коэффициенты, $\mathbb{R}^m \ni x^i$ — узлы кубатурной формулы, $i = \overline{0, n}$.

Построение кубатурных формул можно осуществлять следующими способами:

1. Замена функции f интерполирующей функцией $\varphi = \sum_i f(x^i)\varphi_i$, тогда $A_i = \int_\Omega \varphi_i(x)\rho(x)dx$.
2. Использование условий, определяющих степень точности относительно выбранного базиса $\{\varphi_i\}$.

Сейчас мы рассмотрим указанные способы подробнее на примере двумерного случая, $\Omega \subset \mathbb{R}^2$, и единичной весовой функции $\rho(x, y) = 1$.

17.2 Сведение кратного интеграла к повторному

17.2.1 Интеграл по прямоугольнику

Простейший случай возникает когда $\Omega = \Pi = [a, b] \times [c, d]$:

$$S(f) = \iint_\Pi f(x, y)dxdy = \int_c^d dy \int_a^b f(x, y)dx.$$

Приближая внутренний интеграл с помощью какой-нибудь квадратурной суммы $Q_{n_1}(f) = \sum_{i=0}^{n_1} A_i^1 f(x_i)$, считая при этом y фиксированным, получим

$$\int_c^d dy \int_a^b f(x,y)dx \approx \int_c^d \sum_{i=0}^{n_1} A_i^1 f(x_i,y)dy.$$

Этот однократный интеграл приближаем (другой) квадратурной суммой $Q_{n_2}(f) = \sum_{j=0}^{n_2} A_j^2 f(y_j)$, получая в итоге

$$\boxed{S(f) \approx \sum_{i=0}^{n_1}\sum_{j=0}^{n_2} A_i^1 A_j^2 f(x_i,y_j)} \qquad (17.2)$$

Рассмотрим теперь несколько частных случаев этой формулы.

Формула средних прямоугольников. Если для численного интегрирования по обеим переменным взять формулу средних прямоугольников,

$$\int_a^b f(x)dx \approx (b-a) f\left(\frac{a+b}{2}\right),$$

получим

$$\iint\limits_{\Pi} f(x,y)dxdy \approx S(1)\cdot f\left(\frac{a+b}{2},\frac{c+d}{2}\right), \qquad (17.3)$$

где $S(1) = (b-a)(d-c)$ — площадь прямоугольника Π.

Кубатурная формула трапеций. Обозначим $h_1 = b-a$, $h_2 = d-c$. Для формулы трапеций имеем $n_1 = n_2 = 1$, $A_0^k = A_1^k = h_k/2$, $k = 1,2$, $x_0 = a$, $x_1 = b$, $y_0 = c$, $y_1 = d$. Подставляем всё это в (17.2):

$$S(f) = \frac{h_2}{2}\left(\frac{h_1}{2}f(a,c) + \frac{h_1}{2}f(b,c)\right) + \frac{h_2}{2}\left(\frac{h_1}{2}f(a,d) + \frac{h_1}{2}f(b,d)\right),$$

откуда

$$\iint\limits_{\Pi} f(x,y)dxdy \approx \frac{S(1)}{4}\Big(f(a,c) + f(b,c) + f(a,d) + f(b,d)\Big). \qquad (17.4)$$

Кубатурная формула Симпсона. В этом случае $n_1 = n_2 = 2$, $A_0^k = A_2^k = h_k/6$, $A_1^k = 2h_k/3$, $x_0 = a$, $x_1 = (a+b)/2 = \xi$, $x_2 = b$, $y_0 = c$, $y_1 = (d-c)/2 = \eta$, $y_2 = d$. Подставляем это в (17.2) и получаем

$$\iint\limits_{\Pi} f(x,y)dxdy \approx \frac{4}{9}S(1)f(\xi,\eta) + \tag{17.5}$$

$$+\frac{1}{9}S(1)\Big(f(\xi,c)+f(\xi,d)+f(a,\eta)+f(b,\eta)\Big) + \tag{17.6}$$

$$+\frac{1}{36}S(1)\Big(f(a,c)+f(a,d)+f(b,c)+f(b,d)\Big). \tag{17.7}$$

17.2.2 Интеграл по криволинейной трапеции

Пусть теперь область интегрирования Ω является криволинейной трапецией, то есть ограничена прямыми $x = a$ и $x = b$, а также кривыми $y = y_1(x)$ и $y = y_2(x)$. Тогда

$$S(f) = \iint_{\Omega} f(x,y)dxdy = \int_a^b dx \int\limits_{y_1(x)}^{y_2(x)} f(x,y)dy = \int_a^b F(x)dx.$$

Этот случай сводится к предыдущему следующим образом. Приведём интеграл

$$F(x) = \int\limits_{y_1(x)}^{y_2(x)} f(x,y)dy$$

к интегралу с фиксированными пределами заменой

$$y = y_1(x) + \big(y_2(x) - y_1(x)\big)t, \quad t \in [0,1].$$

Тогда

$$F(x) = (y_2(x) - y_1(x)) \int_0^1 f\Big(x, y_1(x) + \big(y_2(x) - y_1(x)\big)t\Big)dt.$$

Теперь можно последовательно применить квадратурные формулы:

$$S(f) = \int_a^b F(x)dx \approx \sum_{i=0}^{n_1} A_i^1 F(x_i) =$$

$$= \sum_{i=0}^{n_1} A_i^1 (y_2(x_i) - y_1(x_i)) \int_0^1 f\Big(x_i, y_1(x_i) + \big(y_2(x_i) - y_1(x_i)\big)t\Big)dt \approx$$

$$\approx \sum_{i=0}^{n_1} A_i^1 (y_2(x_i) - y_1(x_i)) \sum_{j=0}^{n_2} A_j^2 f\Big(x_i, y_1(x_i) + \big(y_2(x_i) - y_1(x_i)\big)t_j\Big).$$

$\triangleright_1$ Постройте аналог кубатурных формул средних прямоугольников и трапеций для этого случая.

17.3 Интерполяционные кубатурные формулы

17.3.1 Общее определение

Кубатурную формулу

$$\iint_\Omega f(x,y)dxdy \approx \sum_{i=0}^{n} A_i f(x_i,y_i)$$

будем называть *интерполяционной*, если она получена заменой

$$f(x,y) \approx \varphi(x,y) = \sum_{i=0}^{n} f(x_i,y_i)\varphi_i(x,y),$$

где φ — интерполяционный многочлен, $f(x_i,y_i) = \varphi(x_i,y_i)$.

Понятно, что коэффициенты интерполяционных кубатурных формул выражаются формулой

$$A_i = \iint_\Omega \varphi_i(x,y)dxdy.$$

17.3.2 Прямоугольная сетка

Рассмотрим снова интеграл по прямоугольнику

$$S(f) = \iint_\Pi f(x,y)dxdy.$$

Очевидно, что если рассмотреть на $\Pi = [a,b]\times[c,d]$ прямоугольную сетку узлов

$$\{(x_i,y_j)\} = \{x_0,x_1,\ldots,x_{n_1}\}\times\{y_0,y_1,\ldots,y_{n_2}\}$$

и проинтерполировать по ней подынтегральную функцию многочленом (11.12), то в результате получатся в точности квадратурные формулы (17.2):

$$S(f) \approx \iint_\Pi \sum_{i,j} f(x_i,y_j)\Lambda_i^1(x)\Lambda_j^2(y)dxdy =$$

$$= \sum_{i,j} f(x_i,y_j)\int_a^b \Lambda_i^1(x)dx \int_c^d \Lambda_j^2(y)dy = \sum_{i,j} A_i^1 A_j^2 f(x_i,y_j).$$

Другие формулы получатся, если область интегрирования Ω не совпадает с Π. Учитывая, что в этом случае выбор подходящей прямоугольной сетки представляется проблематичным, рассмотрение таких формул не имеет большого смысла.

17.3.3 Интеграл по треугольнику

Достаточно большой практический интерес представляют кубатурные формулы по треугольной области, так как их можно использовать для численного интегрирования по произвольной области, предварительно выполнив её триангуляцию.

Итак, пусть $\Omega = \Delta$, где Δ — невырожденный треугольник с вершинами $p_0 = (x_0, y_0)$, $p_1 = (x_1, y_1)$, $p_2 = (x_2, y_2)$. Для приближённого вычисления

$$S(f) = \iint_{\Delta} f(x, y) dx dy$$

проинтерполируем f по вершинам треугольника. Получится многочлен первой степени

$$f(x, y) \approx f(p_0)\varphi_0(x, y) + f(p_1)\varphi_1(x, y) + f(p_2)\varphi_2(x, y)$$

и соответственная кубатурная формула

$$S(f) \approx A_0 f(p_0) + A_1 f(p_1) + A_2 f(p_2), \quad A_i = \iint_{\Delta} \varphi_i(x, y) dx dy.$$

Самое приятное здесь заключается в том, что для нахождения A_i нам не нужен явный вид φ_i. На лишь достаточно знать, что эти функции а) являются многочленами первой степени по каждой переменной и б) образуют фундаментальный базис:

$$\varphi_i(p_j) = \delta_{ij}.$$

Это означает, что $\iint_{\Delta} \varphi_i(x, y) dx dy$ равен объёму треугольной пирамиды (тетраэдра) $P_0 P_1 P_2 Q_i$, где $P_i = (x_i, y_i, 0)$, $Q_i = (x_i, y_i, 1)$. Как мы помним из курса стереометрии, объём любой пирамиды равен

$$V = \frac{1}{3} s h,$$

где s — площадь основания, h — высота. Значит,

$$A_i = \frac{1}{3} S_{\Delta}, \quad i = \overline{0, 2}.$$

где $S_{\Delta} = S(1)$ — площадь треугольника Δ, которую можно вычислить по формуле

$$S_{\Delta} = \frac{1}{2}|G|, \quad G = \det \begin{bmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{bmatrix}. \tag{17.8}$$

Таким образом, окончательно получаем кубатурную формулу

$$\boxed{\iint_{\Delta} f(x,y)dxdy \approx \frac{S_\Delta}{3}\Big(f(p_0)+f(p_1)+f(p_2)\Big),} \tag{17.9}$$

где $p_i = (x_i, y_i)$ — вершины треугольника Δ, а его площадь S_Δ вычисляется по формуле (17.8).

17.4 Кубатурные формулы наивысшей АСТ

Для начала вспомним, что многочленом двух переменных степени m называется функция вида

$$P_m = \sum_{i=0}^{m}\sum_{j=0}^{m-i} a_{ij}\psi_{ij}, \quad \psi_{ij}(x,y) = x^i y^j.$$

Таким образом, базис пространства всех таких многочленов состоит из $\frac{(m+1)(m+2)}{2}$ функций

$$\Psi_m = \{\psi_{ij}\}_{i+j\leqslant m}.$$

Будем говорить, что кубатурная формула

$$S(f) = \iint_{\Omega} f(x)\rho(x)dxdy \approx Q(f) = \sum_{i=0}^{n} A_i f(x_i, y_i) \tag{17.10}$$

имеет АСТ, равную m, если

$$S(\psi) = Q(\psi) \quad \forall \psi \in \Psi_m \tag{17.11}$$

и существует по крайней мере одна функция $\hat{\psi} \in \Psi_{m+1}$ такая, что

$$S(\hat{\psi}) \neq Q(\hat{\psi})$$

Условия (17.11) представляют собой $(m+1)(m+2)/2$ уравнений (нелинейных при $m > 1$). Число неизвестных параметров в кубатурной формуле (17.10) равно $3n+3$, так что для достижения АСТ m кубатурная формула должна иметь как минимум

$$n+1 = \left\lceil \frac{(m+1)(m+2)}{6} \right\rceil$$

узлов. Здесь $\lceil x \rceil$ — ближайшее к x целое число, больше либо равное x. В частности, для достижения АСТ 1 достаточно взять 1 узел, для АСТ 2 — 2 узла, для АСТ 3 — 4 узла.

17.4.1 Кубатурная формула средних

Построим квадратурную формулу наивысшей АСТ с одним узлом для произвольной области интерирования $\Omega \in \mathbb{R}^2$. Формула будет иметь вид

$$S(f) \approx A_0 f(x_0, y_0).$$

Применяя непосредственно условия (17.11), для $\psi(x,y) = \psi_0(x,y) \equiv 1$ получаем

$$A_0 = S(1) = \iint_\Omega \rho(x,y)dxdy,$$

для $\psi_{10}(x,y) = x$ получаем

$$x_0 = S(\psi_{10})/S(1) = \frac{1}{S(1)} \iint_\Omega x\rho(x,y)dxdy,$$

и наконец для $\psi_{01}(x,y) = y$

$$y_0 = \frac{1}{S(1)} \iint_\Omega y\rho(x,y)dxdy.$$

В частности, при $\rho \equiv 1$ получаем, что A_0 равен площади области Ω, а точка (x_0, y_0) это центр масс данной области.

Совершенно аналогично доказывается общий многомерный случай: если $\Omega \subset \mathbb{R}^N$, $x = (x_1, \ldots, x_N)$, то

$$S(f) = \int_\Omega f(x)\rho(x)dx \approx S(1)f(\xi),$$

где $\mathbb{R}^N \ni \xi$ — обобщённый центр масс, вычисляемый по формуле

$$\xi = \frac{1}{S(1)} \int_\Omega x\rho(x)dx.$$

18 Численное решение интегральных уравнений. Метод механических квадратур

18.1 Введение

Мы начинаем новый раздел — численное решение интегральных уравнений. Начнём с рассмотрения интегральных уравнений Фредгольма второго рода

$$u(x) - \lambda \int_a^b k(x,s)u(s)ds = f(x) \tag{18.1}$$

Здесь $U \ni u$ — искомая функция одной переменной, U — пространство функций, например $L_2[a,b]$, $f \in U$ — заданная функция, $k : \mathbb{R}\times\mathbb{R} \to \mathbb{R}$ — функция двух переменных, $\lambda \in \mathbb{R}$. Уравнение (18.1) в некоторых случаях удобно записать в операторной форме

$$(\mathcal{I} - \lambda\mathcal{K})u = f, \tag{18.1′}$$

где $\mathcal{I}u = u$ — тождественный оператор, $\mathcal{K}$ — линейный интегральный оператор,

$$\mathcal{K}u(x) = \int_a^b k(x,s)u(s)ds.$$

Функцию k называют ядром интегрального оператора $\mathcal{K}$.

Все методы приближённого решения линейных интегральных уравнений (по крайней мере все, что мы будем рассматривать) в конечном итоге будут сводиться к решению СЛАУ.

18.2 Метод механических квадратур для ИУФ-II

18.2.1 Вывод расчётных формул

Первый метод численного решения (18.1), который мы рассмотрим, основывается на приближении интеграла в уравнении некоторой квадратурной суммой. Называется такой метод *методом механических квадратур*. Рассмотрим квадратурную формулу с остатком

$$\int_a^b f(x)dx = \sum_{j=0}^{n} A_j f(x_j) + R,$$

$R = R_n(f)$, и применим её к интегралу в (18.1):

$$\int_a^b k(x,s)u(s)ds = \sum_{i=0}^{n} A_i k(x,x_i)u(x_i) + R(x), \tag{18.2}$$

откуда

$$u(x) - \lambda \sum_{j=0}^{n} A_j k(x, x_j) u(x_j) - \lambda R(x) = f(x). \tag{18.3}$$

Подставляя сюда $x = x_i$, получаем

$$u(x_i) - \lambda \sum_{j=0}^{n} A_j k(x_i, x_j) u(x_j) = f(x_i) + \lambda R(x_i).$$

Значит, точное решение в точках $\{x_i\}$ удовлетворяет системе линейных уравнений

$$(I - \lambda K) v = g + \lambda r, \tag{18.4}$$

где

$$\begin{array}{l} v = (u(x_0), \ldots, u(x_n))^T, \\ g = (f(x_0), \ldots, f(x_n))^T, \\ r = (R(x_0), \ldots, R(x_n))^T, \\ K = \Big(A_j k(x_i, x_j)\Big)_{i,j=0}^{n}, \\ I \text{ — единичная матрица.} \end{array} \tag{18.5}$$

$\triangleright_1$ При каких λ система (18.4) имеет единственное решение?

Отметим, что до сих пор все соотношения у нас были точными. Найти $u(x_i)$ из (18.4) нам мешает вектор r, соответствующий остатку квадратурной формулы. Отбрасывая этот остаток в (18.3), получаем уравнение, которому удовлетворяет приближённое решение $\tilde{u}(x) \approx u(x)$:

$$\tilde{u}(x) - \lambda \sum_{j=0}^{n} A_j k(x, x_j) \tilde{u}(x_j) = f(x). \tag{18.6}$$

Обозначив $\tilde{u}(x_i) = y_i$, аналогичным образом получим соответствующую СЛАУ для нахождения вектора $y = (y_0, \ldots, y_n)^T$:

$$\boxed{(I - \lambda K) y = g.} \tag{18.7}$$

Решив данную систему, найдём приближенные значения решения во всех точках $\{x_i\}$. Кроме этого, из (18.6) автоматически получаем выражение для $\tilde{u}(x)$ $\forall\, x \in [a, b]$:

$$\boxed{\tilde{u}(x) = f(x) + \lambda \sum_{j=0}^{n} A_j k(x, x_j) y_j.} \tag{18.8}$$

Таким образом, **алгоритм метода механических квадратур** прост до неприличия:

1. Выбираем квадратурную формулу.
2. Вычисляем матрицу $I - \lambda K$ и вектор правой части g из (18.5).
3. Решая СЛАУ (18.7), находим значения $y_i \approx u(x_i)$.
4. Записываем приближённое решение для всех $x \in [a, b]$ по формуле (18.8).

Надо заметить, что алгоритмы всех рассматриваемых в дальнейшем методов решения ИУ Фредгольма II рода будут иметь практически такой же вид.

18.2.2 Анализ погрешности метода

Наша цель теперь — рассмотреть и оценить норму вектора погрешностей

$$\varepsilon = v - y, \quad \varepsilon_i = u(x_i) - \tilde{u}(x_i).$$

Начнём с того, что запишем вид отбрасываемого в (18.3) квадратурного остатка $R(x)$. Так как длина отрезка $[a, b]$ может быть достаточно большой, как правило в методе механических квадратур используются составные квадратурные формулы. Как мы помним, для этих формул справедливо представление остатка (15.8c):

$$R_n^N(f) = C(b-a)h^{l-1}f^{(m)}(\eta)$$

(пожалуйста, не путайте f в этой формуле и правую часть уравнения (18.1)). Напомним, что здесь h — длина шага разбиения в составной КФ.

Согласно (18.2) роль интегрируемой функции в нашем случае выполняет функция $k(x, \cdot)u(\cdot)$, и здесь x — параметр, а не аргумент! Значит, для любого $x \in [a, b]$ имеем

$$R(x) = C(b-a)h^{l-1}\frac{\partial^m}{\partial s^m}\Big(k(x,s)u(s)\Big)\Big|_{s=\eta}, \quad \eta \in [a, b]. \tag{18.9}$$

Теперь установим связь между R и ε. Вычитая из (18.4) уравнение (18.7), получаем

$$(I - \lambda K)\varepsilon = \lambda r,$$

откуда, обозначив

$$B = I - \lambda K,$$

имеем

$$\|\varepsilon\| = \|\lambda B^{-1} r\| \leqslant |\lambda| \cdot \|B^{-1}\| \|r\|.$$

Рассматривая векторную максимум-норму, из (18.9) получаем

$$\|r\| = \max_i |R(x_i)| \leqslant |C|(b-a)Mh^{l-1},$$

где

$$M = \max_{x,s\in[a,b]} \left| \frac{\partial^m}{\partial s^m}\Big(k(x,s)u(s)\Big) \right|.$$

Понятно, что эта оценка справедлива лишь для достаточно гладких k и u. Ну а учитывая, что про u мы мало что знаем, ценность данной оценки не очень велика. Тем не менее, доведём дело до конца:

$$\|\varepsilon\| \leqslant |\lambda\, C| \cdot (b-a) \cdot ||B^{-1}|| \cdot M \cdot h^{l-1}, \tag{18.10}$$

то есть, проще говоря,

$$\|\varepsilon\| = O(h^{l-1}).$$

Теперь посмотрим что получится, скажем, для составной КФ средних прямоугольников. Остаток простой формулы имеет вид (13.13):

$$R_0(f) = \frac{h^3}{24} f''(\eta),$$

значит остаток составной формулы имеет вид

$$R_0^N(f) = \frac{b-a}{24} h^2 f''(\eta),$$

и (18.10) примет вид

$$\|\varepsilon\| \leqslant \frac{|\lambda|}{24}(b-a)||B^{-1}||Mh^2.$$

$\triangleright_2$ В каких случаях метод механических квадратур будет давать точное решение?

19 Метод замены ядра на вырожденное

19.1 ИУ Фредгольма II с вырожденным ядром

Продолжаем рассматривать ИУ Фредгольма второго рода

$$u(x) - \lambda \int_a^b k(x,s)u(s)ds = f(x).$$

Как мы помним из курса функционального анализа, ядро k называется *вырожденным*, если оно имеет вид

$$k(x,s) = \sum_{i=0}^{n} a_i(x)b_i(s), \tag{19.1}$$

где a_i и b_i — функции одной переменной. Подставляя такое ядро в уравнение, получим

$$u(x) - \lambda \int_a^b \sum_{i=0}^{n} a_i(x)b_i(s)u(s)ds = u(x) - \lambda \sum_{i=0}^{n} a_i(x) \underbrace{\int_a^b b_i(s)u(s)ds}_{c_i} = f(x),$$

или

$$\boxed{u = f + \lambda \sum_{i=0}^{n} c_i a_i.} \tag{19.2}$$

Здесь u, f, $\{a_i\}$ — функции, $\{c_i\}$ — неизвестные скалярные коэффициенты, которые запишем в виде

$$c_i = \int_a^b b_i(s)u(s)ds = (b_i, u). \tag{19.3}$$

Для их нахождения подставим (19.2) в (19.3):

$$c_i = (b_i, u) = (b_i, f + \lambda \sum_{j=0}^{n} c_j a_j) = (b_i, f) + \lambda \sum_{j=0}^{n} c_j (b_i, a_j),$$

или

$$c_i - \lambda \sum_{j=0}^{n} c_j (b_i, a_j) = (b_i, f), \quad i = \overline{0,n}.$$

Мы получили СЛАУ для нахождения вектора $c = (c_0, \ldots, c_n)^T$. Запишем её в матричном виде:

$$\boxed{(I - \lambda B)c = d,} \tag{19.4}$$

где

$$B = \big((b_i, a_j)\big)_{i,j=0}^{n} = \left(\int_a^b b_i(x)a_j(x)dx\right)_{i,j=0}^{n} \qquad (19.5)$$
$$d = \big((b_0, f), (b_1, f), \ldots, (b_n, f)\big)^T.$$

Ясно, что в общем случае все интегралы в этих формулах приходится вычислять приближённо с помощью квадратурных формул.

Итак, **алгоритм решения ИУФ-II с вырожденным ядром вида** (19.1) выглядит следующим образом:

1. Вычисляем матрицы $I - \lambda B$ и вектор d по формулам (19.5).
2. Решая СЛАУ (19.4), находим вектор c.
3. Записываем решение в виде (19.2).

Понятное дело, что если ядро интегрального уравнения не вырождено, то его всегда можно приблизить вырожденным и воспользоваться описанным выше методом для нахождения приближённого решения. Существует две стратегии приближения ядра вырожденным: первая основана на использовании одномерной аппроксимации, вторая — на двумерной, то есть на линейной аппроксимации тензорными произведениями.

19.2 Замена ядра путём одномерной аппроксимации

Рассмотрим любой способ линейной аппроксимации функций одной переменной:

$$f(x) \approx (\Pi f)(x) = \sum_i \alpha_i \varphi_i(x), \quad \alpha_i \in \mathbb{R}.$$

Воспользуемся этим способом для приближения ядра k по переменной x:

$$k(x, s) \approx (\Pi k(\cdot, s))(x) = \sum_i \alpha_i(s)\varphi_i(x). \qquad (19.6a)$$

Ничто не мешает, конечно, сделать аппроксимацию по другой переменной:

$$k(x, s) \approx (\Pi k(x, \cdot))(s) = \sum_i \alpha_i(x)\varphi_i(s). \qquad (19.6b)$$

Необходимо понимать, что функции-коэффициенты α_i зависят от ядра k, тогда как базисные функции φ_i — нет. Поэтому если стоит вопрос о выборе одного из двух способов (19.6), то нужно руководствоваться формулой (19.2): в случае приближения типа (19.6a) приближённое решение будет выражаться через функции φ_i, а в случае (19.6b) — через α_i, то есть через ядро k.

Рассмотрим теперь подробно как это работает на примерах двух наиболее распространённых типов приближений: интерполяции и среднеквадратичного приближения.

19.2.1 Интерполяция

Общий случай. Рассмотрим сразу случай интерполяции по узлам $\{x_i\}_{i=0}^n$ с использованием фундаментального базиса $\{\varphi_i\}_{i=0}^n$:

$$\Pi f = \sum_i f(x_i)\varphi_i.$$

Применим этот способ к ядру k по переменной x согласно (19.6a):

$$k(x,s) \approx \sum_i k(x_i,s)\varphi_i(x),$$

то есть

$$a_i(x) = \varphi_i(x), \quad b_i(s) = k(x_i,s).$$

Тогда общая схема решения уравнений с вырожденным ядром согласно (19.2) даёт приближённое решение в виде

$$\tilde{u}(x) = f(x) + \lambda \sum_{i=0}^n c_i\varphi_i(x), \tag{19.7}$$

где вектор коэффициентов c вычисляется по формулам (19.4), (19.5).

Теперь посмотрим что будет, если интерполировать ядро по второй переменной:

$$k(x,s) \approx \sum_i k(x,x_i)\varphi_i(s),$$

и, соответственно

$$\tilde{u}(x) = f(x) + \lambda \sum_{i=0}^n c_i k(x,x_i). \tag{19.8}$$

$\triangleright_1$ Как будут связаны между собой матрицы СЛАУ для нахождения коэффициентов $\{c_i\}$ в (19.7) и (19.8)?

Из имеющегося у нас арсенала способов интерполяции теоретически можно использовать любой. Кратко обсудим их достоинства и недостатки в контексте нашей задачи.

Полиномиальная интерполяция. Честно говоря, эти формулы уже надоели:

$$\varphi_i(x) = \Lambda_i(x) = \prod_{j\neq i} \frac{x-x_j}{x_i-x_j} = \frac{\omega(x)}{(x-x_i)\omega'(x_i)}.$$

Понятно, что если уж использовать этот способ, то узлы $\{x_i\}$ лучше брать чебышевскими. У этого способа два основных недостатка: 1) при больших n базисные функции становятся очень громоздкими и 2) носитель базисных функций содержит весь отрезок $[a,b]$, то есть приближённое вычисление интегралов (19.5) будет достаточно трудоёмким.

Тригонометрическая интерполяция Обладает всеми недостатками своей полиномиальной подружки.

Интерполяция сплайнами — пожалуй, лучше всего подходит. Причём сплайны нужно брать порядка 0 (8.4) или 1 (8.2), чтобы носитель базисных сплайнов был компактным[4]. А интерполировать следует по второй переменной (формула (19.6b)), чтобы приближённое решение было гладким (хотя тогда при больших n будет трудно вычислять $\tilde{u}(x)$).

$\triangleright_2$ Постройте соответствующие вычислительные формулы для сплайнов нулевого и первого порядка на равномерной сетке.

19.2.2 Среднеквадратичное приближение

Пусть на отрезке $[a, b]$ определена система ортонормированных функций $\{\varphi_i\}_{i=0}^n$. Тогда оператор среднеквадратичного приближения Π работает так:

$$\Pi f = \sum_i (f, \varphi_i)\varphi_i.$$

Применяя этот оператор для приближения k по x, получаем

$$k(x, s) \approx \sum_i (k(\cdot, s), \varphi_i)\varphi_i(x),$$

то есть

$$a_i(x) = \varphi_i(x), \quad b_i(s) = \int_a^b K(x, s)\varphi_i(x)dx.$$

И здесь уже видны проблемы: для вычисления $b_i(s)$ в общем случае нужно использовать квадратурные формулы. А с учётом последующей необходимости вычисления матрицы B (19.5), использование этого подхода на практике вряд ли стоит рекомендовать. Тем не менее среднеквадратичное приближение вполне может использоваться для приближения ядра тензорными произведениями.

19.3 Приближение ядра тензорными произведениями

Теперь мы будем приближать ядро по обеим переменным тензорными произведениями вида

$$k(x, s) \approx \tilde{k}(x, s) = \sum_{i=0}^{n} \sum_{j=0}^{n} \alpha_{ij}\varphi_i(x)\varphi_j(s). \tag{19.9}$$

[4] B-сплайны высоких порядков имеют компактный носитель — скажете вы. Но, к сожалению, интерполировать этими сплайнами при $n > 1$ мы не умеем.

Для простоты мы используем один и тот же базис по каждой переменной. Понятно, что в общем случае базисы могут быть разными.

Самое главное сейчас — заметить, что ядро $\tilde{k}$, хоть и является вырожденным, несколько отличается от (19.1): здесь в сумме $(n+1)^2$ функций, а не $(n+1)$. Но его можно привести к общему виду:

$$\sum_{j=0}^{n} \alpha_{ij}\varphi_i(x)\varphi_j(s) = \sum_i \varphi_i(x) \sum_j \alpha_{ij}\varphi_j(s) = \sum_i a_i(x)b_i(s),$$

где

$$a_i(x) = \varphi_i(x), \quad b_i(s) = \sum_j \alpha_{ij}\varphi_j(s).$$

В связи с этим изменятся формулы (19.5). Обозначим матрицу коэффициентов

$$A = (\alpha_{ij})_{i,j=0}^{n} \tag{19.10a}$$

и рассмотрим вектор d:

$$d_i = (b_i, f) = \sum_j \alpha_{ij}(\varphi_j, f),$$

откуда

$$\begin{gathered} d = Ae, \\ e = \big((\varphi_0, f), (\varphi_1, f), \ldots, (\varphi_n, f)\big)^T. \end{gathered} \tag{19.10b}$$

Теперь разберёмся с матрицей B:

$$B_{ij} = (b_i, a_j) = (\sum_k \alpha_{ik}\varphi_k, \varphi_j) = \sum_k \alpha_{ik}(\varphi_k, \varphi_j),$$

то есть

$$\begin{gathered} B = A\Gamma, \\ \Gamma = \big((\varphi_i, \varphi_j)\big)_{i,j=0}^{n}. \end{gathered} \tag{19.10c}$$

Несмотря на то, что полученные формулы вкладываются в общую схему, описанную в начале лекции, в силу некоторой специфики переформулируем алгоритм. Итак, **алгоритм метода приближения ядра тензорными произведениями**:

1. Приближаем ядро тензорными произведениями (19.9), то есть находим коэффициенты α_{ij}.
2. Вычисляем матрицы $I - \lambda B$ и вектор d по формулам (19.10).
3. Решаем СЛАУ (19.4).

4. Записываем приближённое решение в виде

$$\tilde{u}(x) = f(x) + \lambda \sum_{i=0}^{n} c_i \varphi_i.$$

Сделаем несколько замечаний. Во-первых, способ приближения тензорными произведениями может быть любой. Подойдёт и интерполяция, и среднеквадратичное приближение. В качестве базиса, как и ранее, лучше выбирать функции с компактным носителем. Фундаментальные сплайны первого порядка — очень хороший вариант.

Во-вторых, матрица Грамма Γ не зависит от ядра и может быть вычислена для конкретного базиса раз и навсегда. Следовательно, нет необходимости в многократном вычислении интегралов для получения матрицы B, как в случае (19.5).

$\triangleright_3$ Вывести расчётные формулы для случая а) интерполяции б) среднеквадратичного приближения тензорными произведениями фундаментальных сплайнов первого порядка.

20 Проекционные методы

20.1 Проекционный метод для операторных уравнений общего вида

Рассмотрим линейное уравнение в некотором нормированном векторном пространстве U:

$$\mathcal{L}u = f. \tag{20.1}$$

Здесь $U \ni u$ — неизвестный элемент, $f \in U$, $\mathcal{L} : U \to U$ — обратимый линейный оператор.

Общая идея *проекционных методов* решения операторных уравнений вида (20.1) состоит в следующем. Приближённое решение $\tilde{u}$ ищется в виде

$$\tilde{u} = \sum_{i=0}^{n} \alpha_i \varphi_i,$$

где $\{\alpha_i\}$ — неизвестные коэффициенты, $\{\varphi_i\}$ — линейно независимые базисные функции, определяющие подпространство

$$U_n = \operatorname{span}\{\varphi_i\}_{i=0}^{n} \subset U.$$

Таким образом, $\tilde{u} \in U_n$. Рассмотрим невязку

$$\tilde{r} = \mathcal{L}\tilde{u} - f.$$

Понятно, что в общем случае $\tilde{r} \neq 0$, однако мы можем потребовать, чтобы *проекция* этой невязки на подпространство U_n была нулевой. Для этого нам понадобится проектор

$$\Pi : U \to U_n,$$

$\Pi v = v \ \ \forall v \in U_n$. Тогда условие, которому должно удовлетворять приближённое решение, записывается в виде

$$\Pi\tilde{r} = \Pi(\mathcal{L}\tilde{u} - f) = 0. \tag{20.2}$$

Лемма 20.1. *Пусть $\Pi : U \to U_n$ является оператором линейной аппроксимации, то есть*

$$\lambda_i(\Pi v) = \lambda_i(v) \quad \forall\, i = \overline{0, n},$$

где $\{\lambda_i\}$ — набор линейно независимых линейных функционалов. Тогда если матрица

$$\Phi = \Big(\lambda_i(\varphi_j)\Big)_{i,j=0}^{n} \tag{20.3}$$

не вырождена, то

$$\boxed{\Pi v = 0 \quad \Leftrightarrow \quad \lambda_i(v) = 0 \quad \forall\, i = \overline{0, n}.}$$

Доказательство. По определению (см. (11.3)) имеем

$$\Pi v = \sum_i a_i \varphi_i,$$

где $a = (a_0, a_1, \ldots, a_n)^T$ — решение СЛАУ

$$\Phi a = \big(\lambda_0(v), \ldots, \lambda_n(v)\big)^T.$$

Так как по условию теоремы матрица Φ не вырождена, данная система будет иметь нулевое решение тогда и только тогда, когда её правая часть — нулевой вектор. Для полноты доказательства остаётся воспользоваться линейной независимостью функций $\{\varphi_i\}$. ■

Согласно данной лемме, требование (20.2) эквивалентно условиям

$$\boxed{\lambda_i(\mathcal{L}\tilde{u} - f) = 0, \quad i = \overline{0, n},} \tag{20.4}$$

или

$$\lambda_i(\mathcal{L}\tilde{u} - f) = \left[\tilde{u} = \sum_j \alpha_j \varphi_j\right] = \sum_j \alpha_j \lambda_i(\mathcal{L}\varphi_j) - \lambda_i(f) = 0, \quad i = \overline{0, n}.$$

Таким образом, для нахождения приближённого решения задачи $\mathcal{L}u = f$ в виде $\tilde{u} = \sum_i \alpha_i \varphi_i$ проекционным методом необходимо решить СЛАУ

$$\boxed{\Psi\alpha = g,} \tag{20.5}$$

где

$$\Psi = \Big(\lambda_i(\mathcal{L}\varphi_j)\Big)_{i,j=0}^{n}, \tag{20.6a}$$

$$g = \big(\lambda_0(f), \lambda_1(f), \ldots, \lambda_n(f)\big)^T, \tag{20.6b}$$

$\{\lambda_i\}$ — линейные функционалы, определяющие проектор $\Pi : U \to \operatorname{span}\{\varphi_i\}_{i=0}^n$.

20.2 Проекционные методы для ИУ Фредгольма II рода

Запишем интегральное уравнение Фредгольма II рода в операторной форме (18.1′):

$$(\mathcal{I} - \lambda\mathcal{K})u = f, \tag{20.7}$$

$$\mathcal{K}v(x) = \int_a^b k(x, s)v(s)ds \quad \forall v \in U.$$

Нет ничего проще, чем применить к решению этого уравнения проекционный метод, описанный выше. Выбираем любой способ линейной аппроксимации, то есть базис $\{\varphi_i\}$ и функционалы $\{\lambda_i\}$ (не путать эти функционалы с константой λ).

Далее, согласно (20.7) имеем

$$\mathcal{L} = \mathcal{I} - \lambda\mathcal{K},$$

значит элементы матрицы Ψ (20.6a) будут иметь вид

$$\lambda_i(\mathcal{L}\varphi_j) = \lambda_i(\varphi_j - \lambda\mathcal{K}\varphi_j) = \lambda_i(\varphi_j) - \lambda \cdot \lambda_i(\mathcal{K}\varphi_j),$$

то есть

$$\Psi = \Phi - \lambda M, \tag{20.8}$$

где

$$\Phi \text{ — матрица (20.3)}, \quad M = \Big(\lambda_i(\mathcal{K}\varphi_j)\Big)_{i,j=0}^{n}, \tag{20.9}$$

Таким образом, **алгоритм проекционного метода для решения ИУ Фредгольма II рода** выглядит следующим образом:

1. Выбираем способ аппроксимации.
2. Строим матрицу Ψ (20.8) и вектор g (20.6b).
3. Решаем СЛАУ (20.5), и записываем приближённое решение в виде $\tilde{u} = \sum_{i=0}^{n} \alpha_i \varphi_i$.

Рассмотрим теперь наиболее популярные частные случаи проекционных методов.

20.3 Коллокационный метод

Проекционный метод, основанный на интерполяции, называется *коллокационным методом*. Другими словами, коллокационный метод получается из общей схемы проекционных методов, если в качестве определяющих функционалов взять

$$\lambda_i(v) = v(x_i), \quad i = \overline{0, n},$$

где $\{x_i\}_{i=0}^{n}$ — попарно различные точки на $[a, b]$, называемые *узлами коллокации*.

По определению коллокационного метода получаем следующий вид составных частей матрицы Ψ (20.8) и вектора g (20.6b):

$$\Phi = \Big(\varphi_j(x_i)\Big)_{i,j=0}^{n}, \quad M = \Big(\int_a^b k(x_i,s)\varphi_j(s)ds\Big)_{i,j=0}^{n}, \tag{20.10a}$$

$$g = \big(f(x_0), f(x_1), \ldots, f(x_n)\big)^T. \tag{20.10b}$$

После этого решаем СЛАУ (20.5)

$$(\Phi - \lambda M)\alpha = g$$

и записываем решение: $\tilde{u} = \sum_{i=0}^{n} \alpha_i \varphi_i$.

Понятно, что основная вычислительная нагрузка при использовании метода приходится на нахождение $(n+1)^2$ интегралов – элементов матрицы M. Поэтому удобнее всего, как и прежде, брать в качестве базиса $\{\varphi_i\}$ функции с компактным носителем — фундаментальные сплайны нулевого и первого порядка. В этом случае матрица Φ, понятное дело, будет единичной.

$\triangleright_1$ Запишите формулы, соответствующие сплайнам порядка 0 и 1. На что похож полученный метод?

$\triangleright_2$ Запишите вид коллокационного метода, соответствующего интерполяционному многочлену в форме а) Лагранжа, б) Ньютона.

Смысл коллокационного метода состоит в следующем. Так как этот метод является проекционным, в соответствии с общими требованиями (20.4) система (20.5), (20.10a) соответствует уравнениям

$$\lambda_i(\tilde{r}) = \lambda_i(\tilde{u} - \lambda\mathcal{K}\tilde{u} - f) = 0,$$

то есть

$$\tilde{u}(x_i) - \lambda\int_a^b k(x_i,s)\tilde{u}(s)ds = f(x_i), \quad i = \overline{0,n}.$$

Такие условия иногда называют *условиями коллокации*: требуется, чтобы приближённое решение удовлетворяло исходному уравнению лишь в конечном наборе узлов $\{x_i\}_{i=0}^{n}$.

20.3.1 Метод Галёркина

Метод Галёркина — проекционный метод с использованием среднеквадратичного приближения:

$$\lambda_i(v) = (v, \varphi_i) = \int_a^b v(x)\varphi_i(x)dx.$$

Отсюда согласно накатанной схеме получаем

$$\Phi = \Gamma = \Big((\varphi_i, \varphi_j)\Big)_{i,j=0}^{n}, \tag{20.11a}$$

$$M_{ij} = (\varphi_i, \mathcal{K}\varphi_j) = \int_a^b \int_a^b \varphi_i(x) k(x,s) \varphi_j(s) ds dx, \tag{20.11b}$$

$$g_i = (f, \varphi_i) = \int_a^b f(x)\varphi_i(x) dx. \tag{20.11c}$$

Очень хочется верить, что вы знаете, что делать с этими данными дальше.

Нетрудно заметить, что метод Галёркина гораздо более трудоёмок, чем метод коллокации: для вычисления матрицы M требуется вычисление двойных интегралов, с использованием кубатурных формул в общем случае. Рекомендации по выбору базиса остаются прежними: оптимальными являются функции с конечным носителем.

Стоит заметить, что вообще метод Галёркина — один из наиболее популярных методов решения не только интегральных уравнений, но и многих других задач. Поэтому мы с ним ещё не раз встретимся.

21 Численное решение интегральных уравнений Вольтерры

21.1 Введение

Как мы помним из курса функционального анализа, (линейное) интегральное уравнение Вольтерры II рода имеет вид

$$u(x) - \lambda \int_a^x k(x,s)u(s)ds = f(x). \tag{21.1}$$

Теоретически можно свести это уравнение к уравнению типа Фредгольма вида

$$u(x) - \lambda \int_a^b k_0(x,s)u(s)ds = f(x),$$

где

$$k_0(x,s) = \begin{cases} k(x,s), & s \leqslant x, \\ 0, & s > x, \end{cases}$$

и после этого применить любой из рассмотренных в предыдущих лекциях методов. Однако полученное ядро является разрывной функцией, и это существенно осложняет дело. Поэтому уравнения Вольтерры требуют построения специальных методов.

Сделаем ещё одно замечание. Если ядро k не зависит от x, а решение u дифференцируемо, то такое уравнение Вольтерры эквивалентно задаче Коши

$$u'(x) = f'(x) + \lambda k(x)u(x), \quad u(a) = f(a).$$

Поэтому методы численного решения уравнений Вольтерры и обыкновенных дифференциальных уравнений (которые мы начнём рассматривать со следующей лекции) имеют много общего.

21.2 Метод механических квадратур

21.2.1 Общая схема

По аналогии с лекцией 18 можно применить к уравнению (21.1) метод механических квадратур. Введём на $[a,b]$ сетку узлов

$$a \leqslant x_0 < x_1 < \ldots x_n \leqslant b$$

и рассмотрим уравнение в i-ом узле:

$$u(x_i) - \lambda \int_a^{x_i} k(x_i,s)u(s)ds = f(x_i). \tag{21.2}$$

Приблизим интеграл какой-нибудь квадратурной формулой. При этом важно, чтобы узлами этой формулы были $\{x_j\}_{j=0}^{i}$ (почему?):

$$u(x_i) - \lambda \sum_{j=0}^{i} A_j^{(i)} k(x_i, x_j) u(x_j) - \lambda R_i(x_i) = f(x_i).$$

Здесь, $R_i(x_i)$ — остаток квадратурной формулы. Понятно, что для каждого i коэффициенты $\{A_j^{(i)}\}_{j=0}^{i}$ могут быть различными. Отбрасывая остаток, получаем уравнения для нахождения приближённых значений решения в узлах:

$$\tilde{u}(x_i) - \lambda \sum_{j=0}^{i} A_j^{(i)} k(x_i, x_j) \tilde{u}(x_j) = f(x_i), \quad i = \overline{0, n}. \tag{21.3}$$

Обозначая $\tilde{u}(x_i) = y_i$ получаем СЛАУ, аналогичную (18.7):

$$\boxed{(I - \lambda K) y = g,} \tag{21.4}$$

и отличающуюся только видом матрицы K (напомним, что $g_i = f(x_i)$):

$$K = \left(A_j^{(i)} k(x_i, x_j) \right)_{i,j=0}^{n},$$

причём здесь мы положили $A_j^{(i)} = 0$ при $i < j$, то есть матрица K — нижнетреугольная. Таким образом, (21.4) — система с нижнетреугольной матрицей, которая легко решается обратным ходом метода Гаусса. Можно даже сразу выписать явные формулы из (21.3), для краткости обозначая $k(x_i, x_j) = k_{ij}$:

$$y_i - \lambda \sum_{j=0}^{i} A_j^{(i)} k_{ij} y_j = f(x_i),$$

откуда

$$\boxed{y_i = \frac{1}{1 - \lambda A_{ii}^{(i)} k_{ii}} \left(f(x_i) + \lambda \sum_{j=0}^{i-1} A_j^{(i)} k_{ij} y_j \right), \quad i = \overline{0, n}.} \tag{21.5}$$

Заметим, что в случае уравнения Вольтерры мы не имеем формулы для вычисления приближённого решения $\tilde{u}(x)$ в произвольной точке отрезка $[a, b]$. Приходится довольствоваться только значениями на сетке.

21.2.2 Пример для равномерной сетки

Рассмотрим подробно что получится для равномерной сетки

$$x_i = a + ih, \quad h = \frac{b - a}{n}.$$

Так как $x_0 = a$, сразу получаем $y_0 = g_0$. Для приближения интеграла возьмём квадратурную формулу трапеций:

$$\int_{x_0}^{x_i} F(x)dx \approx h\left(\frac{F(x_0)}{2} + F(x_1) + F(x_2) + \ldots + F(x_{i-1}) + \frac{F(x_i)}{2}\right),$$

то есть

$$A_0^{(i)} = A_i^{(i)} = \frac{h}{2}, \quad A_j^{(i)} = h, \quad j = \overline{1, i-1}.$$

В результате СЛАУ (21.4) примет вид

21.3 Нелинейное уравнение

Метод механических квадратур легко обобщается на случай нелинейного уравнения Вольтерры

$$u(x) - \lambda \int_a^x k(x, s, u(s))ds = f(x). \tag{21.6}$$

Для такого уравнения (21.3) примет вид

$$\tilde{u}(x_i) - \lambda \sum_{j=0}^{i} A_j^{(i)} k(x_i, x_j, \tilde{u}(x_j)) = f(x_i), \quad i = \overline{0, n}.$$

Обозначая $\tilde{u}(x_i) = y_i$ и предполагая известными значения $y_0, \ldots, y_{i-1}$, для нахождения y_i получаем уравнение

$$y_i = f(x_i) + \lambda F_i(y_i),$$

где $F : \mathbb{R} \to \mathbb{R}$,

$$F = \sum_{j=0}^{i-1} A_j^{(i)} k(x_i, x_j, y_j) + A_i^{(i)} k(x_i, x_i, \cdot).$$

$\triangleright_1$ Запишите алгоритм метода Ньютона для нахождения y_i.

22 Численное решение задачи Коши. Одношаговые методы

22.1 Постановка задачи

Рассмотрим систему обыкновенных дифференциальных уравнений (ОДУ)

$$y'(x) = f(x, y(x)), \tag{22.1a}$$

и начальное условие

$$y(x_0) = y_0, \quad x_0 \in \mathbb{R},\ y_0 \in \mathbb{R}^n. \tag{22.1b}$$

Здесь $y : \mathbb{R} \to \mathbb{R}^n$ — искомая вектор-функция, $f : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$ — функция, задающая поле направлений (наклонов). Формулы (22.1) вместе определяют *задачу Коши* для системы ОДУ. Во многих случаях эти формулы удобно объединить в одну путём перехода к интегральному уравнению

$$y(x) = y_0 + \int_{x_0}^{x} f(z, y(z))dz. \tag{22.2}$$

Пусть нам необходимо найти значение решения в некоторой точке $x_1 > x_0$. На практике редко удаётся вычислить $y(x_1)$ точно, поэтому приходится прибегать к приближённым (численным) методам.

22.2 Методы Эйлера

22.2.1 Явный метод Эйлера

Рассмотрим скалярный случай $n = 1$. Начальное условие (22.1b) позволяет составить уравнение касательной к точному решению y в точке x_0: если (x_1, y_1) — точка на касательной, то справедливо соотношение

$$\frac{y_1 - y_0}{x_1 - x_0} = y'(x_0) = f(x_0, y_0).$$

Полагая $y(x_1) \approx y_1$, получаем самый простой и самый известный метод — *явный метод Эйлера*

$$\boxed{y_1 = y_0 + (x_1 - x_0)f(x_0, y_0).} \tag{22.3}$$

22.2.2 Неявный метод Эйлера

Рассмотрим соотношение (22.2) для $x = x_1$:

$$y(x_1) = y_0 + \int_{x_0}^{x_1} f(z, y(z))dz.$$

Приблизим интеграл по правилу правых прямоугольников[5]

$$\int_a^b f(x)dx \approx (b-a)f(b)$$

и получим неявное соотношение для вычисления $y_1 \approx y(x_1)$:

$$\boxed{y_1 = y_0 + (x_1 - x_0)f(x_1, y_1).} \tag{22.4}$$

Это *неявный* метод Эйлера. Очевидно, что для вычисления y_1 необходимо решать нелинейное уравнение (систему из n нелинейных уравнений в общем случае).

$\triangleright_1$ Примените для аппроксимации интеграла (22.2) квадратурную формулу трапеций и запишите формулу для соответствующего метода.

22.3 Одношаговые методы: терминология

Итак, методы Эйлера позволяют вычислить приближённое значение решения в точке x_1 по известному значению y_0 в точке x_0. Методы такого типа называются *одношаговыми*.

Одношаговым методом численного интегрирования задачи (22.1) называется отображение

$$\Phi : \{x_0, y_0, x_1\} \mapsto y_1,$$

которое данному начальному условию (x_0, y_0) ставит в соответствие значение приближённого решения задачи (22.1) в данной точке $x_1 \in \mathbb{R}$:

$$\Phi(x_0, y_0, x_1) = y_1 \approx y(x_1).$$

Главное различие между двумя методами Эйлера состоит в том, что в одном случае y_1 задаётся явной формулой, а в другом оно определяется как решение нелинейного уравнения. Аналогично все методы численного решения ОДУ разделяются на явные и неявные.

Если отображение Φ является неявной функцией, то есть задано уравнением вида

$$\Psi(x_0, y_0, x_1, y_1) = 0,$$

то соответствующий одношаговый метод называется *неявным*. В противном случае, то есть если y_1 явно выражается через x_0, y_0 и x_1, метод называется *явным*.

[5] Заметим, что если применить формулу левых прямоугольников, получится явный метод Эйлера (22.3).

Как правило, отображение Φ определено для всех x_1, лежащих в некоторой окрестности x_0, что позволяет рассматривать функцию непрерывного аргумента $\Phi(x_0, y_0, \cdot)$.

Пусть дан одношаговый метод Φ и зафиксировано начальное условие (x_0, y_0). *Функцией метода* Φ будем называть отображение $\varphi : \mathbb{R} \to \mathbb{R}^n$, определённое по правилу

$$\varphi(x) = \Phi(x_0, y_0, x).$$

22.4 Порядок метода

Классическим показателем точности методов численного интегрирования ОДУ является порядок метода. Это понятие связано с разложением точного и приближённого решений в ряд Тейлора.

Рассмотрим сначала y — точное решение уравнения (22.1) в скалярном случае и предположим, что оно достаточно гладкое. Введём следующие традиционные обозначения:

$$f = f(x_0, y_0), \quad f_x = \frac{\partial f}{\partial x}(x_0, y_0), \quad f_y = \frac{\partial f}{\partial y}(x_0, y_0), \tag{22.5}$$

$$f_{\underbrace{x \ldots x}_{M} \underbrace{y \ldots y}_{N}} = \frac{\partial^{M+N} f}{\partial x^M \partial y^N}(x_0, y_0). \tag{22.6}$$

Разложение Тейлора для $y(x_0 + h)$ в точке x_0 имеет вид

$$y(x) = y_0 + hy'(x_0) + \frac{h^2}{2!}y''(x_0) + \frac{h^3}{3!}y'''(x_0) + \ldots \tag{22.7}$$

Тот факт, что y удовлетворяет (22.1), позволяет вычислить неизвестные коэффициенты $y^{(k)}(x_0)$ для любого k. Действительно, мы имеем

$$y'(x_0) = f, \tag{22.8a}$$

$$y''(x_0) = \frac{d}{dx} f(x, y(x))\bigg|_{x=x_0} = f_x + f_y y'(x_0) = f_x + f_y f. \tag{22.8b}$$

$$y'''(x_0) = \frac{d^2}{dx^2} f(x, y(x))\bigg|_{x=x_0} = f_{xx} + 2f_{xy} f + f_{yy} f^2 + f_y(f_x + f_y f), \tag{22.8c}$$

и так далее. Таким образом, мы имеем теоретическую возможность *точно* вычислить разложение Тейлора (22.7).

Теперь зафиксируем начальное условие (x_0, y_0), рассмотрим произвольный одношаговый метод Φ и φ — функцию этого метода. Традиционный способ оценки точности такого приближения заключается в сравнении разложений Тейлора для $\varphi(x)$ с разложением (22.7). Чем больше членов в этих разложениях совпадают, тем выше порядок метода.

Локальной погрешностью одношагового метода Φ называется функция

$$r(x) = \varphi(x) - y(x) = \Phi(x_0, y_0, x) - y(x).$$

Одношаговый метод имеет порядок p, если для всех достаточно гладких задач его локальная погрешность имеет вид

$$r(x_0 + h) = Ch^{p+1} + O(h^{p+2}), \quad C \neq 0, \tag{22.9}$$

то есть разложения Тейлора в точке x_0 для точного решения $y(x_0 + h)$ и приближённого $\varphi(x_0 + h)$ совпадают до члена h^p включительно:

$$y^{(k)}(x_0) = \varphi^{(k)}(x_0) \quad \forall k = \overline{0, p}.$$

Слагаемое Ch^{p+1} называют *главным членом локальной погрешности*, а константу C — *константой погрешности* метода.

Замечание 22.1. Следует понимать, что высокий порядок метода не всегда гарантирует высокую точность приближения.

22.4.1 Примеры вычисления порядка

Порядок явного метода Эйлера. Согласно определению (22.3) имеем

$$\varphi(x) = \Phi(x_0, y_0, x) = y_0 + (x - x_0)f(x_0, y_0).$$

Отсюда получаем

$$\begin{aligned}
&\varphi(x_0) = y_0, \\
&\varphi'(x_0) = f(x_0, y_0) = y'(x_0), \\
&\varphi^{(k)}(x_0) = 0 \neq y^{(k)}(x_0) \quad \forall k \geqslant 2.
\end{aligned}$$

Следовательно,

$$y(x_0 + h) - \varphi(x_0 + h) = \frac{h^2}{2!}y''(x_0) + O(h^3).$$

Таким образом, порядок явного метода Эйлера равен единице, а константа погрешности, согласно (22.8b), равна

$$C = \frac{1}{2}(f_x + f_y f). \tag{22.10}$$

Порядок неявного метода Эйлера. Рассмотрим (22.4). Для этого метода приближённое решение $\varphi(x)$ определяется неявным соотношением

$$\varphi(x) = y_0 + (x - x_0)f(x, \varphi(x)).$$

Вычислим производные $\varphi(x)$:

$$\varphi'(x) = f(x, \varphi(x)) + (x - x_0)\frac{d}{dx}f(x, \varphi(x)),$$
$$\varphi''(x) = 2\frac{d}{dx}f(x, \varphi(x)) + (x - x_0)\frac{d^2}{dx^2}f(x, \varphi(x)).$$

Отсюда с учётом того, что $\varphi(x_0) = y_0$, получаем

$$\varphi'(x_0) = f(x_0, y_0) = y'(x_0),$$
$$\varphi''(x_0) = 2\frac{d}{dx}f(x_0, y_0) = 2(f_x + f_y f) \neq y''(x_0).$$

Следовательно, разложение локальной погрешности в ряд Тейлора имеет вид

$$y(x_0 + h) - \varphi(x_0 + h) = -\frac{h^2}{2!}y''(x_0) + O(h^3).$$

Итак, неявный метод Эйлера имеет первый порядок, а его константа погрешности равна

$$C = -\frac{1}{2}(f_x + f_y f). \tag{22.11}$$

Замечание 22.2. На рассмотренных примерах видно, что в общем случае константа погрешности C представляет собой вектор, который выражается через значения частных производных f в точке (x_0, y_0).

$\triangleright_2$ Найдите порядок метода, построенного в упражнении 22.1, а также главный член его локальной погрешности.

23 Методы Рунге–Кутты

Методы типа Рунге–Кутты (РК) являются наиболее популярными одношаговыми методами численного решения обыкновенных дифференциальных уравнений в настоящее время. Название этих методов связано с именами немецких математиков Карла Рунге (1856-1927) и Мартина Кутты (1867-1944). Рунге был первым, кто построил частные методы данного типа, а Кутта впоследствии дал общую форму (явного) метода практически в том же виде, который мы имеем сейчас.

23.1 Простейшие методы Рунге–Кутты

До сих пор мы рассматривали лишь два простейших метода численного решения ОДУ — явный и неявный методы Эйлера. Настало время построить чуть более совершенные одношаговые методы.

Рассмотрим задачу Коши в форме интегрального уравнения (22.2) для $x = x_0 + h$:

$$y(x_0 + h) = y_0 + \int_{x_0}^{x_0+h} f(z, y(z))dz.$$

В интеграле удобно сделать замену переменных $z = x_0 + th$:

$$y(x_0 + h) = y_0 + h\int_0^1 f\big(x_0 + th, y(x_0 + th)\big)dt. \tag{23.1}$$

Приблизим интеграл квадратурной формулой средних прямоугольников:

$$y(x_0 + h) \approx y_0 + hf\Big(x_0 + \frac{h}{2}, y\big(x_0 + \frac{h}{2}\big)\Big).$$

Использовать эту формулу для вычислений пока что нельзя, так как неизвестно значение $y(x_0 + \frac{h}{2})$. Но его можно приблизить любым из методов Эйлера (22.3), (22.4). Для начала возьмём явный метод:

$$y\Big(x_0 + \frac{h}{2}\Big) \approx y_0 + \frac{h}{2}f(x_0, y_0),$$

и получим численный метод вида

$$Y_2 = y_0 + \frac{h}{2}f(x_0, y_0), \tag{23.2a}$$

$$y_1 = y_0 + hf\Big(x_0 + \frac{h}{2}, Y_2\Big). \tag{23.2b}$$

Этот метод будем называть *явным методом средних прямоугольников (средней точки)*.

Если же приблизить $y(x_0 + \frac{h}{2})$ неявным методом Эйлера

$$y\left(x_0 + \frac{h}{2}\right) \approx Y_1 = y_0 + \frac{h}{2} f\left(x_0 + \frac{h}{2}, Y_1\right),$$

получим *неявный метод средних прямоугольников*

$$Y_1 = y_0 + \frac{h}{2} f\left(x_0 + \frac{h}{2}, Y_1\right), \tag{23.3a}$$

$$y_1 = y_0 + h f\left(x_0 + \frac{h}{2}, Y_1\right). \tag{23.3b}$$

Оба полученных метода, а также и сами методы Эйлера являются частными случаями методов Рунге–Кутты.

23.2 Общий случай

В общем случае приблизим интеграл в (23.1) квадратурной формулой с узлами $(c_1, \ldots, c_s)$ и весами $(b_1, \ldots, b_s)$:

$$\int_0^1 f\big(x_0 + th, y(x_0 + th)\big)\,dt \approx \sum_{i=1}^{s} b_i f\big(x_0 + c_i h, y(x_0 + c_i h)\big).$$

Для нахождения приближений к неизвестным величинам $y(x_0 + c_i h)$ используем такой же подход:

$$y(x_0 + c_i h) = y_0 + \int_{x_0}^{x_0 + c_i h} f(z, y(z))dz = y_0 + h \int_0^{c_i} f(x_0 + th, y(x_0 + th))dt,$$

а интеграл приближаем квадратурной формулой *по тем же самым узлам* $(c_1, \ldots, c_s)$, *но с другими коэффициентами* $(a_{i1}, \ldots, a_{is})$:

$$y(x_0 + c_i h) \approx y_0 + h \sum_{j=1}^{s} a_{ij} f(x_0 + c_j h, y(x_0 + c_j h)).$$

Обозначая $y(x_0 + c_i h) \approx Y_i$, в итоге получаем **s-стадийный метод Рунге–Кутты общего вида**:

$$Y_i = y_0 + h \sum_{j=1}^{s} a_{ij} f(x_0 + c_j h, Y_j), \quad i = \overline{1, s}, \tag{23.4a}$$

$$y_1 = y_0 + h \sum_{i=1}^{s} b_i f\big(x_0 + c_i h, Y_i\big). \tag{23.4b}$$

Запись метода РК в виде (23.4) называется *симметричной*. Более распространённой является другая, эквивалентная форма записи, которая получается заменой переменных

$$\kappa_i = f(x_0 + c_i h, Y_i), \tag{23.5}$$

или

$$Y_i = y_0 + h\sum_{j=1}^{s} a_{ij}\kappa_j. \tag{23.6}$$

В результате имеем

$$\kappa_i = f\left(x_0 + c_i h, y_0 + h\sum_{j=1}^{s} a_{ij}\kappa_j\right), \quad i = \overline{1,s}, \tag{23.7a}$$

$$y_1 = y_0 + h\sum_{i=1}^{s} b_i\kappa_i. \tag{23.7b}$$

Таким образом, s-стадийный метод РК определяется $s^2 + 2s$ параметрами $(a_{ij})_{i,j=1}^{s}$, $(b_1, \ldots, b_s)$ и $(c_1, \ldots, c_s)$. Традиционно эти параметры размещают в таблицу вида

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array} = \begin{array}{c|ccc} c_1 & a_{11} & \cdots & a_{1s} \\ \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & \cdots & a_{ss} \\ \hline & b_1 & \cdots & b_s \end{array}. \tag{23.8}$$

Матрицу $A = (a_{ij})$ называют *матрицей Бутчера*[6], а всю таблицу (23.8) — *таблицей Бутчера*.

$\triangleright_1$ Постройте таблицу Бутчера для методов Эйлера, а также для явного и неявного метода средних прямоугольников.

Заметим, что формулы (23.7a), как и их симметричный аналог (23.4a), задают систему нелинейных уравнений относительно неизвестных $\{\kappa_i\}_{i=1}^{s}$ ($\{Y_i\}_{i=1}^{s}$ соответственно). То есть методы РК являются в общем случае неявными. Однако если коэффициенты метода удовлетворяют условиям

$$a_{ij} = 0 \quad \forall i \leqslant j$$

и

$$c_1 = 0,$$

то такой метод РК, очевидно, является явным.

[6] Джон Бутчер (John Butcher), университет Окленда,— новозеландский математик, внесший большой вклад в развитие и популяризацию методов РК.

23.3 Условия порядка для методов РК

Неизвестные коэффициенты методов типа Рунге–Кутты (23.8) традиционно выбираются таким образом, чтобы получившийся метод имел как можно более высокий порядок точности.

Найдём условия, которым должен удовлетворять произвольный метод РК второго порядка. Рассмотрим приближённое решение y_1 как функцию переменной h. Так как функция метода имеет вид $\varphi(x_0 + h) = y_1$, имеем

$$\varphi^{(k)}(x_0) = y_1^{(k)}\Big|_{h=0}.$$

Тогда для того, чтобы метод имел порядок 2, по определению необходимо, чтобы выполнялись условия

$$y_1|_{h=0} = y_0, \tag{23.9a}$$
$$y_1'|_{h=0} = y'(x_0) = f, \tag{23.9b}$$
$$y_1''|_{h=0} = y''(x_0) = f_x + f_y f. \tag{23.9c}$$

Условие (23.9a), очевидно, выполняется всегда. Дифференцируя (23.7b) по h, имеем

$$\begin{aligned} y_1' &= \textstyle\sum_i b_i \kappa_i + h \sum_i b_i \kappa_i', \\ y_1'' &= 2\textstyle\sum_i b_i \kappa_i' + h \sum_i b_i \kappa_i''. \end{aligned} \tag{23.10}$$

Во всех суммах здесь и далее суммирование идёт от 1 до s. Вычислим

$$\kappa_i' = \frac{d}{dh} f(x_0 + c_i h, Y_i) = c_i f_x'() + f_y'() Y_i' = \big[\text{используем (23.6)}\big] = \\ = c_i f_x'() + f_y'() \left(\sum_j a_{ij} \kappa_j + h \sum_j a_{ij} \kappa_j' \right). \tag{23.11}$$

Здесь $() = (x_0 + c_i h, Y_i)$. Используя (23.11) из (23.10) получаем

$$y_1'|_{h=0} = \textstyle\sum_i b_i \kappa_i|_{h=0} = \sum_i b_i f,$$
$$y_1''|_{h=0} = 2\textstyle\sum_i b_i \kappa_i'\big|_{h=0} = 2\sum_i b_i \left[c_i f_x + f_y f \sum_j a_{ij} \right].$$

Сопоставляя эти формулы с (23.9b), (23.9c), получаем следующие условия второго порядка для методов РК:

$$\begin{aligned} &\textstyle\sum_i b_i = 1, \\ &\textstyle\sum_i b_i c_i = \frac{1}{2}, \\ &\textstyle\sum_i b_i \sum_j a_{ij} = \frac{1}{2}. \end{aligned} \tag{23.12}$$

$\triangleright_2$ Выведите условия третьего порядка.

Замечание 23.1. В подавляющем большинстве случаев при построении методов РК априори полагают $c_i = \sum_j a_{ij}$, так что последние два условия в (23.12) эквивалентны.

Замечание 23.2. При выводе условий (23.12) мы рассматривали случай скалярного ОДУ. Следует понимать, что в общем случае условия порядка для систем ОДУ не будут совпадать со скалярными.

23.4 Явные методы Рунге–Кутты

23.4.1 Общий вид

Именно явные методы Рунге–Кутты методы можно назвать классическими, так как именно такого рода формулы строили в своё время Рунге и Кутта.

Методы Рунге–Кутты с таблицей вида

$$\begin{array}{c|cccc} 0 & & & & \\ c_2 & a_{21} & & & \\ c_3 & a_{31} & a_{32} & & \\ \vdots & \vdots & \vdots & \ddots & \\ c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} \\ \hline & b_1 & b_2 & \cdots & b_{s-1} \quad b_s \end{array} \tag{23.13}$$

называются *явными s-стадийными методами Рунге–Кутты*.

Общая схема вычисления $y_1 \approx y(x_0 + h)$ по формулам (23.7) для явных методов РК имеет простой вид:

$$\begin{aligned} \kappa_1 &= f(x_0, y_0) \\ \kappa_2 &= f(x_0 + c_2 h, y_0 + h a_{21} \kappa_1), \\ \kappa_3 &= f(x_0 + c_3 h, y_0 + h(a_{31}\kappa_1 + a_{32}\kappa_2)), \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \kappa_s &= f(x_0 + c_s h, y_0 + h(a_{s1}\kappa_1 + \ldots + a_{s,s-1}\kappa_{s-1})); \\ y_1 &= y_0 + h \textstyle\sum_{i=1}^{s} b_i \kappa_i. \end{aligned} \tag{23.14}$$

Явные методы являются наиболее простыми в реализации методами типа РК: вспомогательные значения κ_i вычисляются последовательно, каждое последующее из них явно выражается через уже найденные. Такой процесс, очевидно, невозможен, если в верхнем треугольнике матрицы A (включая диагональ) есть ненулевые элементы.

23.4.2 Примеры явных методов Рунге–Кутты

Второй порядок. Используя условия (23.12) легко получить общий вид явных двустадийных методов второго порядка. Так как $c_1 = a_{11} = a_{12} = a_{22} = 0$, для оставшихся коэффициентов метода имеем условия

$$b_1 + b_2 = 1,$$

$$b_2 c_2 = b_2 a_{21} = \tfrac{1}{2},$$

откуда получаем следующий общий вид таблицы Бутчера:

$$\begin{array}{c|cc} 0 & & \\ \frac{1}{2}\beta^{-1} & \frac{1}{2}\beta^{-1} & \\ \hline & 1-\beta & \beta \end{array} \qquad (23.15)$$

Здесь $b_2 = \beta$ — параметр, как правило лежащий в полуинтервале $(0, 1]$.

Третий порядок. Приведём без вывода методы порядка 3.

$$\begin{array}{c|ccc} 0 & & & \\ \frac{1}{3} & \frac{1}{3} & & \\ \frac{2}{3} & 0 & \frac{2}{3} & \\ \hline & \frac{1}{4} & 0 & \frac{3}{4} \end{array} \qquad \begin{array}{c|ccc} 0 & & & \\ \frac{1}{2} & \frac{1}{2} & & \\ 1 & 0 & 1 & \\ 1 & 0 & 0 & 1 \\ \hline & \frac{1}{6} & \frac{2}{3} & 0 \quad \frac{1}{6} \end{array} \qquad (23.16)$$

▷$_3$ Постройте какой-нибудь трёхстадийный явный метод РК третьего порядка.

Четвёртый порядок. Наиболее популярный «классический» метод четвёртого порядка определяется таблицей

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{2}{6} & \frac{2}{6} & \frac{1}{6} \end{array} \quad . \qquad (23.17)$$

23.4.3 Явные методы высших порядков

Построение явных методов РК высоких порядков является достаточно сложной задачей, для эффективного решения которой Дж. Бутчеру пришлось

разработать целую теорию. Не вдаваясь в подробности, мы приведём здесь лишь основные «отрицательные» результаты, называемые барьерами Бутчера.

Первый барьер. *При* $p \geqslant 5$ *не существует явных методов РК порядка* p *с числом стадий* $s = p$.

Второй барьер. *При* $p \geqslant 7$ *не существует явных методов РК порядка* p *с числом стадий* $s = p + 1$.

Третий барьер. *При* $p \geqslant 8$ *не существует явных методов РК порядка* p *с числом стадий* $s = p + 2$.

Один из наиболее известных методов высокого порядка (достаточно редко применяемый на практике) — метод десятого порядка точности с числом стадий $s = 17$. Чуть позже мы увидим, что существуют алгоритмы, позволяющие построить явный метод РК сколь угодно высокого порядка точности.

23.5 Неявные методы

Как мы уже говорили, в общем случае метод РК (23.7), (23.4) является неявным. Для его реализации необходимо решить систему нелинейных уравнений (23.7а) или (23.4а).

23.5.1 Диагонально-неявные методы

Данные методы более сложны в реализации, чем явные, но проще, чем неявные методы РК общего вида.

Методы Рунге-Кутты с матрицей A такой, что $a_{ij} = 0$ при всех $i < j$, называются *диагонально-неявными* (ДНРК). Если при этом все диагональные элементы a_{ii} равны между собой, метод называют *однократно диагонально-неявным* (ОДНРК).

Причина, по которой такие методы выделяют в отдельный класс, заключается в относительной простоте вычислений на стадиях (23.7а):

$$\kappa_i = f\Big(x_0 + c_i h, y_0 + h \sum_{j=1}^{i-1} a_{ij}\kappa_j + h a_{ii}\kappa_i\Big). \tag{23.18}$$

На i-й стадии мы имеем нелинейное уравнение для нахождения κ_i (в случае $n = 1$) или систему из n нелинейных уравнений в общем случае.

$\triangleright_4$ Почему однократно диагонально-неявные методы РК так называются?

$\triangleright_5$ Найдите общий вид двухстадийных методов ДНРК третьего порядка. Постройте соответствующий метод ОДНРК.

23.5.2 Неявные методы общего вида

Сейчас мы займемся записью системы нелинейных уравнений, которую нужно решить для применения метода РК, в виде, удобном для машинной реализации. Будем рассматривать симметричную форму записи метода (23.4a). Для того, чтобы прочувствовать структуру этой системы, настоятельно рекомендуется выполнить следующее упражнение.

$\triangleright_6$ Выпишите явный вид уравнений (23.4a) для случая $s = n = 2$.

Прямым (кронекеровским) произведением матриц A и B называется блочная матрица

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1s}B \\ a_{21}B & a_{22}B & \cdots & a_{2s}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{s1}B & a_{s2}B & \cdots & a_{ss}B \end{pmatrix}. \tag{23.19}$$

Итак, возьмём симметричную форму записи метода РК (23.4) и составим из неизвестных векторов Y_i один большой вектор размерности ns (напомним, что $Y_i \in \mathbb{R}^n$):

$$Y = (Y_1, \ldots, Y_s)^T.$$

Аналогичным образом определим вектор-функцию $F : \mathbb{R} \times \mathbb{R}^{ns} \to \mathbb{R}^{ns}$ по правилу

$$F(x, Y) = \begin{pmatrix} f(x + c_1 h, Y_1) \\ f(x + c_2 h, Y_2) \\ \vdots \\ f(x + c_s h, Y_s) \end{pmatrix}.$$

Теперь уравнения

$$Y_i = y_0 + h \sum_{j=1}^{s} a_{ij} f(x_0 + c_j h, Y_j), \quad i = \overline{1, s},$$

могут быть записаны в компактной векторной форме

$$Y = e \otimes y_0 + h(A \otimes I)F(x_0, Y). \tag{23.20}$$

Здесь I — единичная матрица размерности n, $e = (\underbrace{1, \ldots, 1}_{s})^T$.

23.5.3 Реализация неявных методов Рунге–Кутты

Обсудим проблему эффективного решения системы уравнений (23.20). Прежде всего для минимизации ошибок округления[7] вместо Y_i рассмотрим

$$Z_i = Y_i - y_0 \quad \text{и соответственно} \quad Z = Y - e \otimes y_0.$$

Тогда по построению имеем

$$Z_i = h \sum_{j=1}^{s} a_{ij} f(x_0 + c_j h, y_0 + Z_j), \quad i = \overline{1, s},$$

или, по аналогии с (23.20)

$$Z = h(A \otimes I) F(x_0, e \otimes y_0 + Z). \tag{23.21}$$

Традиционно при реализации неявных методов Рунге–Кутты используют метод Ньютона, для применения которого систему (23.21) нужно сначала представить в виде

$$\mathbf{F}(Z) = 0,$$

где

$$\mathbf{F}(Z) = Z - h(A \otimes I) F(x_0, e \otimes y_0 + Z). \tag{23.22}$$

Теперь метод Ньютона можно записать в общей форме

$$Z^{k+1} = Z^k + \Delta Z^k, \quad k = 0, 1, 2, \ldots, \tag{23.23a}$$

где на каждой итерации метода ΔZ^k вычисляется путём решения системы линейных уравнений

$$\frac{\partial \mathbf{F}}{\partial Z}(Z^k) \Delta Z^k = -\mathbf{F}(Z^k). \tag{23.23b}$$

Важный момент теперь состоит в том, чтобы по определению (23.22) выразить матрицу Якоби $\frac{\partial \mathbf{F}}{\partial Z}$ через $\frac{\partial f}{\partial y}$:

$$\frac{\partial \mathbf{F}}{\partial Z}(Z) = I - h \begin{pmatrix} a_{11} \frac{\partial f}{\partial y}(x_0 + c_1 h, y_0 + z_1) & \cdots & a_{1s} \frac{\partial f}{\partial y}(x_0 + c_s h, y_0 + z_s) \\ \vdots & \ddots & \vdots \\ a_{s1} \frac{\partial f}{\partial y}(x_0 + c_1 h, y_0 + z_1) & \cdots & a_{ss} \frac{\partial f}{\partial y}(x_0 + c_s h, y_0 + z_s) \end{pmatrix}.$$

$\triangleright_7$ Выведите эту формулу.

Таким образом, прямая реализация итерационного процесса (23.23) является весьма трудоёмкой: на каждой итерации приходится s раз вычислять матрицу Якоби $\frac{\partial f}{\partial y}$, а также решать СЛАУ (23.23b), что требует $O(N^3)$

[7] Напомним, что в машинной арифметике абсолютная погрешность округления возрастает с ростом модуля числа. Поэтому на практике вычислители стараются работать с меньшими по модулю величинами.

операций, где $N = ns$. Значительно снизить вычислительные затраты позволяет использование «замороженной» матрицы Якоби:

$$\frac{\partial f}{\partial y}(x_0 + c_i h, y_0 + z_i) \approx J = \frac{\partial f}{\partial y}(x_0, y_0),$$

откуда получаем

$$\frac{\partial \mathbf{F}}{\partial Z}(Z^k) \approx I - hA \otimes J.$$

Это приводит к тому, что вместо (23.23b) на каждой итерации метода Ньютона нужно решать СЛАУ с постоянной матрицей:

$$(I - hA \otimes J)\Delta Z^k = -\mathbf{F}(Z^k). \tag{23.24}$$

Для решения этих систем естественно применить метод LU-разложения. Однократное построение разложения $I - hA \otimes J = LU$ требует $O(N^3)$ операций (столько же, сколько и метод Гаусса), зато сложность последующего решения систем вида (23.24) составляет $O(N^2)$. Описанный подход называют *упрощёнными ньютоновскими итерациями*.

Число условий порядка для методов Рунге–Кутты приведено на следующей таблице.

Порядок	1	2	3	4	5	6	7	8	9	10
Число условий	1	2	4	8	17	37	85	200	486	1205

24 Коллокационные методы

24.1 Общая схема построения

Как и ранее, рассмотрим задачу Коши

$$y'(x) = f(x, y(x)), \quad y(x_0) = y_0. \tag{24.1}$$

Для простоты будем рассматривать пока лишь случай одного скалярного уравнения. Пусть необходимо найти приближённое решение этой задачи на отрезке $[x_0, x_0 + h]$.

Для нахождения приближённого решения воспользуемся идеей метода коллокации по аналогии со случаем ИУ Фредгольма. Напомним, что суть метода такова: приближённое решение ищется в виде разложения по некоторому базису функций $\{\varphi_i\}$ и выбирается таким образом, чтобы при его подстановке исходное уравнение обращалось в тождество в заранее заданных точках $\{\xi_i\}$. Мы будем рассматривать полиномиальный базис, хотя понятно, что можно использовать любой другой.

Рассмотрим s различных точек $\{\xi_i\}_{i=1}^s$:

$$x_0 \leqslant \xi_1 < \xi_2 < \ldots < \xi_s \leqslant x_0 + h.$$

Коллокационным многочленом называется многочлен u степени s, удовлетворяющий начальному условию $u(x_0) = y_0$ и дифференциальному уравнению (24.1) в каждой точке ξ_i :

$$u'(\xi_i) = f(\xi_i, u(\xi_i)), \quad i = \overline{1, s}. \tag{24.2}$$

Точки $\{\xi_i\}$ будем называть *узлами коллокации*.

Нахождение коллокационного многочлена осуществляется следующим образом. Рассмотрим его производную u', которая, очевидно, является многочленом степени не выше $s-1$. Обозначим $u'(\xi_i) = \kappa_i$. Тогда u' может быть представлена в виде

$$u'(x) = \sum_{i=1}^{s} \kappa_i \Lambda_i(x), \tag{24.3}$$

где $\{\Lambda_i\}$ — базисные многочлены Лагранжа для узлов коллокации,

$$\Lambda_i(x) = \prod_{j \neq i}^{s} \frac{x - \xi_j}{\xi_i - \xi_j}. \tag{24.4}$$

Интегрируя (24.3) с учётом $u(x_0) = y_0$, получаем

$$u(x) = y_0 + \sum_{i=1}^{s} \kappa_i \int_{x_0}^{x} \Lambda_i(z)dz. \tag{24.5}$$

Подставляя это выражение в условия коллокации (24.2), получаем систему уравнений для нахождения неизвестных κ_i :

$$\kappa_i = f\left(\xi_i,\, y_0 + \sum_{j=1}^{s} \kappa_j \int_{x_0}^{\xi_i} \Lambda_j(z)dz\right). \tag{24.6}$$

Покажем, что построенный метод является представителем семейства методов Рунге–Кутты. Сделаем замену переменных: $x = x_0 + th$. Тогда узлы коллокации можно представить в виде

$$\xi_i = x_0 + c_i h, \quad 0 \leqslant c_i \leqslant 1,$$

и

$$\Lambda_i(x) = \Lambda_i(x_0 + th) = \prod_{j \neq i} \frac{t - c_j}{c_i - c_j} = \tilde{\Lambda}_i(t). \tag{$\widetilde{24.4}$}$$

Подставляя ($\widetilde{24.4}$) в (24.6), получаем

$$\kappa_i = f\Big(x_0 + c_i h,\, y_0 + h\sum_{j=1}^{s} a_{ij}\kappa_j\Big), \quad i = \overline{1,s}, \tag{24.7}$$

где

$$\boxed{a_{ij} = \psi_j(c_i),} \tag{24.8}$$

$$\boxed{\psi_j(t) = \int_0^t \tilde{\Lambda}_j(\tau)d\tau = \int_0^t \prod_{k \neq j} \frac{\tau - c_k}{c_j - c_k} d\tau.} \tag{24.9}$$

Для приближённого решения $y_1 = u(x_0 + h) \approx y(x_0 + h)$ из (24.5) имеем

$$y_1 = y_0 + h\sum_{i=1}^{s} b_i \kappa_i, \tag{24.10}$$

$$\boxed{b_i = \int_0^1 \tilde{\Lambda}_i(\tau)d\tau = \psi_i(1).} \tag{24.11}$$

Как видим, (24.7), (24.10) являются классическими формулами метода Рунге-Кутты!

Таким образом, **алгоритм построения коллокационного метода** состоит из следующих этапов:

1. Выбираем узлы $\{c_i\}_{i=1}^s$.
2. Строим базисные многочлены Лагранжа $\{\tilde{\Lambda}_i\}_{i=1}^s$ ($\widetilde{24.4}$).
3. Интегрируя их, находим $\{\psi_i\}_{i=1}^s$ (24.9).
4. Вычисляем матрицу Бутчера по формуле (24.8).
5. Вычисляем коэффициенты $\{b_i\}_{i=1}^s$ по формуле (24.11).

Преимущество коллокационных методов по сравнению с большинством других методов РК, заключается в том, что они дают не только приближённое решение в одной точке $x_0 + h$, но и непрерывное приближение к решению на всём отрезке $[x_0, x_0 + h]$ — коллокационный многочлен (24.5), который также можно выразить через полиномы ψ_i (24.9) в виде

$$\boxed{u(x_0 + th) = y_0 + h\sum_{i=1}^{s} \kappa_i \psi_i(t).} \tag{24.12}$$

Также по построению имеем

$$u(x_0 + c_i h) = y_0 + h\sum_{j=1}^{s} a_{ij}\kappa_j = Y_i.$$

Понятно, что в случае системы из n обыкновенных дифференциальных уравнений будем иметь $\kappa_i \in \mathbb{R}^n$, а коллокационный многочлен u будет вектор-функцией с полиномиальными компонентами.

24.1.1 Примеры коллокационных методов

На самом деле, практически все простейшие методы РК, рассмотренные нами в предыдущей лекции, являются коллокационными. Чтобы в этом убедиться, получим общий вид одностадийных коллокационных методов. Пусть $c_1 = \theta \in [0, 1]$. При $s = 1$ имеем

$$\tilde{\Lambda}_1(t) = 1.$$

Формулы (24.8), (24.11) дают

$$a_{11} = \theta, \quad b_1 = 1.$$

Из (24.7), (24.10) получаем

$$y_1 = y_0 + h\kappa_1, \qquad (*)$$
$$\kappa_1 = f(x_0 + \theta h, y_0 + h\theta\kappa_1). \qquad (**)$$

Выразим из $(*)$ $\kappa_1 = h^{-1}(y_1 - y_0)$, подставим $(**)$ в $(*)$ и получим в итоге

$$y_1 = y_0 + hf(x_0 + \theta h, y_0 + \theta(y_1 - y_0)). \qquad (24.14)$$

Мы получили так называемый θ-метод. При $\theta = 0$ мы имеем явный, а при $\theta = 1$ — неявный метод Эйлера, при $\theta = 0.5$ — правило средней точки (неявный метод средних прямоугольников).

$\triangleright_1$ Найдите общий вид коллокационных методов РК для $s = 2$.

24.2 Классические коллокационные методы

24.2.1 Порядок коллокационного метода

Особенности конструкции коллокационных методов позволяют легко (в отличие от методов РК общего вида) исследовать их порядок для произвольного числа стадий s.

Теорема 24.1. *Порядок коллокационного метода с узлами $\{c_i\}$ равен $\sigma + 1$, где σ — алгебраическая степень точности интерполяционной квадратурной формулы с этими узлами.*

$\triangleright_2$ Какой наивысший возможный порядок может иметь одностадийный коллокационный метод (24.14)? При каком θ достигается этот порядок?

$\triangleright_3$ Чему равен наивысший порядок коллокационного метода в общем случае?

Теорема 24.1 указывает прямой путь построения коллокационных методов высокого порядка: необходимо в качестве $\{c_i\}$ выбрать узлы квадратурных формул с высокой АСТ, затем вычислить коэффициенты метода по формулам (24.8), (24.11). Получаемые таким образом методы являются в настоящее время наиболее популярными, и сейчас мы с ними познакомимся.

24.2.2 Гауссовы методы

Эти методы строятся на основе узлов квадратурных формул наивысшей алгебраической степени точности, или квадратурных формул Гаусса. Напомним, что узлы этих квадратурных формул являются корнями ортогональных (с единичным весом) на отрезке $[0, 1]$ многочленов — смещённых многочленов Лежандра

$$\frac{d^s}{dx^s}\left(x^s(x-1)^s\right). \qquad (24.15)$$

Согласно теореме 24.1, порядок коллокационных методов Гаусса равен $2s$.

Приведём таблицу коэффициентов для трёхстадийного гауссова метода (порядок 6):

$$\begin{array}{c|ccc} \frac{1}{2}-\frac{\sqrt{15}}{10} & \frac{5}{36} & \frac{2}{9}-\frac{\sqrt{15}}{15} & \frac{5}{36}-\frac{\sqrt{15}}{30} \\ \frac{1}{2} & \frac{5}{36}+\frac{\sqrt{15}}{24} & \frac{2}{9} & \frac{5}{36}-\frac{\sqrt{15}}{24} \\ \frac{1}{2}+\frac{\sqrt{15}}{10} & \frac{5}{36}+\frac{\sqrt{15}}{30} & \frac{2}{9}+\frac{\sqrt{15}}{15} & \frac{5}{36} \\ \hline & \frac{5}{18} & \frac{4}{9} & \frac{5}{18} \end{array} \tag{24.16}$$

24.2.3 Методы Радо

Если в качестве узлов коллокации выбрать корни так называемого правого многочлена Радо

$$\frac{d^{s-1}}{dx^{s-1}}\left(x^{s-1}(x-1)^s\right), \tag{24.17}$$

то получатся методы PK, называемые методами Радо IIA. За счёт фиксированного узла $c_s = 1$ АСТ соответствующей квадратурной формулы на единицу меньше максимальной и, следовательно, порядок методов такого типа равен $2s-1$. Коэффициенты трёхстадийного метода Радо IIA пятого порядка приведены ниже.

$$\begin{array}{c|ccc} \frac{4-\sqrt{6}}{10} & \frac{88-7\sqrt{6}}{360} & \frac{296-169\sqrt{6}}{1800} & \frac{-2+3\sqrt{6}}{255} \\ \frac{4+\sqrt{6}}{10} & \frac{296+169\sqrt{6}}{1800} & \frac{88+7\sqrt{6}}{360} & \frac{-2-3\sqrt{6}}{255} \\ 1 & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \\ \hline & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \end{array} \tag{24.18}$$

24.2.4 Методы Лобатто

Для этих методов узлы c_i являются корнями многочлена

$$\frac{d^{s-2}}{dx^{s-2}}\left(x^{s-1}(x-1)^{s-1}\right). \tag{24.19}$$

Построенные по этим узлам коллокационные методы называеются методами Лобатто IIIA. Порядок точности таких методов равен $2s-2$.

$\triangleright_4$ Постройте двухстадийные методы Радо IIA и Лобатто IIIA.

24.3 Особенности машинной реализации

Коллокационные методы, как мы уже успели заметить, занимают особое место среди неявных методов Рунге–Кутты. Помимо прочего их конструкция позволяет эффективно реализовать выбор начального приближения при решении систем нелинейных уравнений (23.21) (см. раздел 23.5.3).

Рассмотрим процесс пошагового численного интегрирования

$$y_{m+1} = \Phi(x_m, y_m, x_{m+1}), \quad m = 0, 1, 2, \ldots,$$

где Φ — произвольный s-стадийный коллокационный метод. Пусть нами уже вычислено приближенное решение y_k в текущей точке x_k. Так как метод коллокационный, это автоматически означает, что нам известен многочлен u_k степени s, удовлетворяющий условиям коллокации в точках $x_{k-1} + c_i h$, $i = \overline{1, s}$, где $h = x_k - x_{k-1}$. Из (24.12) имеем

$$u_k(x) = y_{k-1} + h \sum_{i=1}^{s} \kappa_i \psi_i \left(\frac{x - x_0}{h} \right). \tag{24.20}$$

Многочлен u_k, очевидно, определён на всей числовой прямой, и если решение задачи Коши меняется достаточно медленно, то u_k будет хорошим приближением к нему не только на отрезке $[x_{k-1}, x_k]$, но и при $x > x_k$.

Наша задача теперь — выбрать с помощью имеющихся данных начальное приближение

$$Z^0 = (z_1^0, \ldots, z_s^0)^T = (Y_1^0 - y_k, \ldots, Y_s^0 - y_k)^T$$

для для ньютоновских итераций (23.23a) на следующем отрезке $[x_k, x_k + h_{new}]$. Это следует сделать таким образом, чтобы соответствующее начальное приближение u_{k+1}^0 к искомому многочлену u_{k+1} совпадало с u_k в точках $x_k + c_i h_{new}$:

$$u_{k+1}^0(x_k + c_i h_{new}) = Y_i^0 = u_k(x_k + c_i h_{new}),$$

откуда

$$z_i^0 = y_{k-1} - y_k + h \sum_{j=1}^{s} \kappa_j \psi_j (1 + \delta c_i), \tag{24.21}$$

где $\delta = h_{new}/h$.

25 Выбор шага численного интегрирования ОДУ

Обычно исследователю необходимо получить приближённое решение задачи Коши (22.1) на «большом» отрезке $[x_0, x_0 + H]$. Так как функция метода φ даёт приемлемое приближение лишь в достаточно малой окрестности точки x_0, традиционный подход к численному интегрированию ОДУ заключается в следующем. Отрезок интегрирования разбивается на N частей сеткой узлов

$$x_0 < x_1 < x_2 \ldots < x_N = x_0 + H \tag{25.1}$$

и строится набор приближённых значений $y_k \approx y(x_k)$ по правилу

$$y_{k+1} = \Phi(x_k, y_k, x_{k+1}), \quad k = \overline{0, N-1}. \tag{25.2}$$

Очевидно, что точность приближённого решения зависит от «частоты» сетки, а именно от величины

$$\tilde{h} = \max_{k=1,\ldots,N} (x_k - x_{k-1}).$$

Как правило требуется найти приближённое решение с какой-то точностью. «Идеальный» критерий точности можно сформулировать, например так:

$$\sum_{k=1}^{N} |y_k - y(x_k)| < \varepsilon. \tag{25.3}$$

К сожалению, такая постановка задачи в общем случае оказывается слишком сложной, и на практике вычислитель требует лишь, чтобы на каждом шаге *главная часть локальной погрешности не превышала заданной величины*, которую обозначают[8] tol:

$$|C_k h_{k+1}^{p+1}| \leqslant tol, \quad \forall\, k = \overline{0, N-1}, \tag{25.4}$$

где $h_k = x_k - x_{k-1}$. Напомним, что константа погрешности C_k выражается через значения частных производных f в точке (x_k, y_k) (см. пункт 22.4).

25.1 Равномерная сетка

Самое простое разбиение вида (25.1) — равномерная сетка

$$\{x_k\} = \{x_0 + kh\}_{k=0}^{N}, \quad h = H/N. \tag{25.5}$$

Такой выбор сетки имеет очевидные удобства при программной реализации и на практике им достаточно часто пользуются, если нужен быстрый результат. Однако такой подход, во-первых, избыточен и, во-вторых, крайне неэффективен.

[8] От английского tolerance.

Предположим, что необходимо найти приближённое решение, удовлетворяющее (25.4) (будем рассматривать скалярный случай $n = 1$). Для этого, очевидно, необходимо располагать априорной информацией о величине

$$C_{\max} = \max_k |C_k|.$$

Получение такой информации в общем случае — достаточно серьёзная проблема. Если всё же $C_{\max}$ нам известно, то из (25.4) следует, что шаг равномерной сетки (25.5) следует выбирать по формуле

$$h = \left(\frac{tol}{C_{\max}} \right)^{\frac{1}{p+1}}. \tag{25.6}$$

Понятно, что в случае, если главный член погрешности существенно изменяется на отрезке интегрирования, такая сетка, мягко говоря, не оптимальна: мы вынуждены делать очень маленькие шаги там, где, может быть, достаточно гораздо бóльшего шага для достижения нужной точности.

25.2 Адаптивный выбор шага

Теперь нам должно быть понятно, что разумный процесс пошагового интегрирования задачи Коши с ограничением на точность вида (25.4) должен использовать величину шага, зависящую от величины константы погрешности C_k. Чем больше константа, тем меньше должен быть шаг, и наоборот. Это означает, что нам необходим механизм оценки величины C_k на каждом шаге.

Существует два широко применяемых способа получения такой оценки. Первый — это уже знакомый нам метод двойного пересчёта, или *правило Рунге*.

25.2.1 Правило Рунге

Правило Рунге для численного решения ОДУ практически полностью аналогично случаю приближённого вычисления определённого интеграла, который мы рассматривали в разделе **??**. Рассмотрим первый шаг процесса численного интегрирования методом Φ порядка p. Выберем какую-то величину начального шага h и сделаем сначала один «большой» шаг длины $2h$

$$\tilde{y}_2 = \Phi(x_0, y_0, x_0 + 2h),$$

а также два шага длины h:

$$y_1 = \Phi(x_0, y_0, x_0 + h), \quad y_2 = \Phi(x_0 + h, y_1, x_0 + 2h).$$

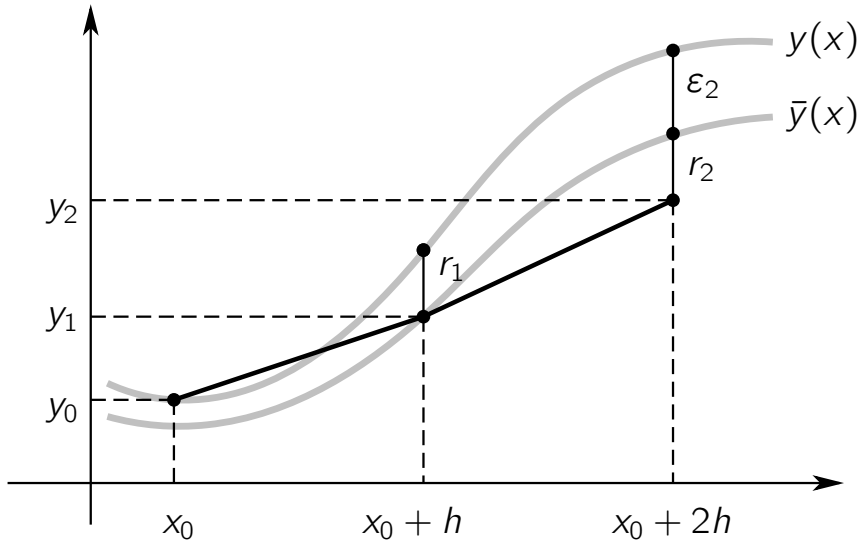


Рис. 1: Распространение ошибки на двух шагах численного метода.

Наша цель — сравнить погрешности приближений y_2 и $\tilde{y}_2$. По определению локальной погрешности (22.9) имеем

$$y(x_0 + 2h) - \tilde{y}_2 = C_0(2h)^{p+1} + O(h^{p+2}). \tag{25.7}$$

Выражение погрешности для y_2 имеет более сложный вид (см. рисунок). Оно состоит из двух частей:

$$e_2 = y(x_0 + 2h) - y_2 = r_2 + \varepsilon_2.$$

Здесь r_2 — локальная погрешность второго шага, по определению имеющая вид

$$r_2 = \bar{y}(x_0 + 2h) - y_2 = C_1 h^{p+1} + O(h^{p+2}), \tag{25.8}$$

а

$$\varepsilon_2 = y(x_0 + 2h) - \bar{y}(x_0 + 2h),$$

где $\bar{y}$ — функция, удовлетворяющая исходному уравнению

$$\bar{y}'(x) = f(x, \bar{y}(x)),$$

но с начальным условием $\bar{y}(x_0 + h) = y_1$.

Можно показать, что для величины ε_2 справедлива оценка

$$\varepsilon_2 = \big(1 + O(h)\big) r_1,$$

где r_1 — локальная ошибка первого шага

$$r_1 = y(x_0 + h) - y_1 = C_0 h^{p+1} + O(h^{p+2}),$$

а константа погрешности C_1 в (25.8) имеет вид

$$C_1 = C_0 + O(h).$$

Таким образом,

$$e_2 = \varepsilon_2 + r_2 = \Big(r_1 + O(h^{p+2})\Big) + \Big((C_0 + O(h))h^{p+1} + O(h^{p+2})\Big),$$

откуда

$$e_2 = y(x_0 + 2h) - y_2 = 2C_0 h^{p+1} + O(h^{p+2}). \tag{25.9}$$

Вычитая из (25.7) выражение (25.9), получаем

$$y_2 - \tilde{y}_2 = 2C_0(2^p - 1)h^{p+1} + O(h^{p+2}).$$

Это равенство позволяет вычислить оценку главной части погрешности для y_2 (см. (25.9)):

$$y(x_0 + 2h) = y_2 + err + O(h^{p+2}), \tag{25.10}$$

где

$$\boxed{err = \frac{y_2 - \tilde{y}_2}{2^p - 1},} \tag{25.11}$$

а также приближённое значение константы погрешности

$$C_0 = \frac{err}{2h^{p+1}} + O(h). \tag{25.12}$$

Две последние формулы позволяют нам, во-первых, оценить погрешность e_2, во-вторых, уточнить приближённое решение y_2, и, в-третьих, выбрать оптимальный шаг интегрирования для дальнейших вычислений. Остановимся на каждом из этих пунктов подробнее.

Величина err, вычисляемая по формуле (25.11), по построению характеризует точность приближённого решения $y_2 \approx y(x_0 + 2h)$:

$$e_2 = err + O(h^{p+2}).$$

Если $|err| < tol$, то данное приближение приемлемо, в противном случае необходимо повторить вычисления с меньшим шагом (см. ниже).

Далее, согласно (25.10), величина

$$\hat{y}_2 = y_2 + err \tag{25.13}$$

является приближением к $y(x_0 + 2h)$ порядка $p + 1$:

$$y(x_0 + 2h) - \hat{y}_2 = O(h^{p+2}).$$

Таким образом, мы фактически получили способ повышения порядка метода на единицу. Величину $\hat{y}_2$ иногда называют *экстраполированным* приближением.

$\triangleright_1$ Пользуясь формулами (25.13), (25.11) постройте общую формулу, по которой из любого одношагового метода Φ порядка p можно получить метод $\widehat{\Phi}$ порядка $p+1$.

$\triangleright_2$ Запишите методы, полученные таким образом из явного и неявного методов Эйлера.

После того, как вычислена оценка погрешности err, возможны два варианта развития событий в зависимости от выполнения неравенства

$$|err| < tol. \tag{25.14}$$

Предположим, что неравенство не выполняется, то есть приближённое решение y_2 не является достаточно точным. Стандартный способ повышения точности в этом случае таков: y_2 «отбрасывается» и вычисления повторяются с меньшим шагом h_{new}. Выбор величины h_{new} упрощается тем, что нам известна оценка константы погрешности (25.12). Согласно (25.9) мы должны выбрать h_{new} из условия

$$|2C_0 h_{new}^{p+1}| < tol,$$

откуда сразу же получаем

$$h_{new} < \delta\, h, \tag{25.15}$$

где

$$\delta = \left(\frac{tol}{|err|}\right)^{\frac{1}{p+1}}. \tag{25.16}$$

Если же неравенство (25.14) выполняется, мы принимаем приближение y_2 (или его уточнённое значение $\hat{y}_2$) и продолжаем процесс численного интегрирования из точки $x_0 + 2h$ с новым значением шага h_{new}. Выбирают этот шаг из следующих соображений. Предположим, что константа погрешности на следующем шаге $\tilde{C}_0$ будет незначительно отличаться от C_0 (это будет верно, если шаг достаточно мал, а функция f достаточно гладкая). Тогда для того, чтобы на новом шаге выполнилась оценка (25.14), мы совершенно аналогично предыдущему случаю должны выбрать h_{new} из условий (25.15), (25.16). Только теперь величина δ будет больше единицы, то есть шаг численного интегрирования *увеличится*.

Таким образом, **алгоритм автоматического (адаптивного) выбора шага численного интегрирования** имеет следующий общий вид.

1. $x \leftarrow x_0,\ \ y \leftarrow y_0,\ \ h \leftarrow h_0,\ \ X \leftarrow x_0 + H.$
2. Если $x = X,$ то завершаем алгоритм.
3. Вычисляем $y_2 \leftarrow \Phi(x+h, \Phi(x, y, x+h), x+2h),$
   $\tilde{y}_2 \leftarrow \Phi(x, y, x+2h).$
4. Находим оценку погрешности err (25.11) и коэффициент δ (25.16).
5. Вычисляем $h_{new} \leftarrow \alpha\delta h$, где $\alpha < 1$ — «страховочный множитель». Как правило α выбирают в пределах от 0.7 до 0.9.

6. Если $\delta < 1$, то полагаем $h \leftarrow h_{new}$ и возвращаемся к пункту 3.
7. Если же $\delta \geqslant 1$, то принимаем шаг: запоминаем пару значений $(x + 2h, y_2)$. Вместо y_2 здесь можно взять $\hat{y}_2$.
8. Полагаем $x \leftarrow x + 2h$, $y \leftarrow y_2\ (\hat{y}_2)$, $h \leftarrow \min\{h_{new}, \frac{X-x}{2}\}$.
9. Возвращаемся к пункту 2.

Замечание 25.1. Понятно, что в силу погрешностей округления условие окончания алгоритма в пункте 2 следует использовать с осторожностью.

Замечание 25.2. В случае системы ОДУ оценка погрешности err, очевидно, будет векторной величиной. Поэтому в формулах вида (25.16) следует использовать какую-либо векторную норму. Кроме этого, для некоторых задач величина относительной погрешности может быть более информативной, поэтому используют также формулы типа

$$err = \frac{1}{2^p - 1}\|D^{-1}(y_2 - \tilde{y}_2)\|,$$

где $D = \mathrm{diag}(\hat{y}_2)$. Если используется абсолютная погрешность, то $D = I$ — единичная матрица.

Замечание 25.3. Общепринято накладывать ограничения на величину шага h: он не должен быть слишком мал. Если же такой случай возникает, программы численного интегрирования обычно выдают сообщение об ошибке и прекращают работу. Кроме этого, накладываются ограничения на скорость увеличения шага, то есть на величину δ:

$$\delta = \min\left(\delta_{\max}, \left(\frac{tol}{|err|}\right)^{\frac{1}{p+1}}\right).$$

25.2.2 Оценка погрешности с помощью вложенных методов.

Данный способ оценки главной части погрешности более прост, но менее универсален, чем правило Рунге. Он требует наличия двух методов: метода Φ порядка p и метода $\widehat{\Phi}$ порядка $q > p$. Обозначим

$$y_1 = \Phi(x_0, y_0, x_0 + h), \quad \hat{y}_1 = \widehat{\Phi}(x_0, y_0, x_0 + h).$$

По определению имеем

$$y(x_0 + h) - y_1 = Ch^{p+1} + O(h^{p+2}),$$
$$y(x_0 + h) - \hat{y}_1 = \widehat{C}h^{q+1} + O(h^{q+2}).$$

Отсюда с учётом того, что $q > p$, получаем

$$\hat{y}_1 - y_1 = Ch^{p+1} + O(h^{p+2}).$$

Таким образом, величина

$$\boxed{err = \hat{y}_1 - y_1} \tag{25.17}$$

с точностью до $O(h^{p+2})$ равна главной части локальной погрешности $y(x_0+h)-y_1$. Дальнейшие рассуждения полностью аналогичны правилу Рунге. В частности, правило выбора оптимального шага (25.15), (25.16), а также общая схема алгоритма адаптивного выбора шага остаётся без изменений. Единственное отличие заключается в способе оценки погрешности: вместо (25.11) имеем (25.17).

В заключение раздела приведём один из наиболее успешных вложенных методов Рунге–Кутты — метод Дорма́на–Принса порядка 5(4).

$$
\begin{array}{c|ccccccc}
0 & & & & & & & \\
\frac{1}{5} & \frac{1}{5} & & & & & & \\
\frac{3}{10} & \frac{3}{40} & \frac{9}{40} & & & & & \\
\frac{4}{5} & \frac{44}{45} & -\frac{56}{15} & \frac{32}{9} & & & & \\
\frac{8}{9} & \frac{19372}{6561} & -\frac{25360}{2187} & \frac{64448}{6561} & -\frac{212}{729} & & & \\
1 & \frac{9017}{3168} & -\frac{355}{33} & \frac{46732}{5247} & \frac{49}{176} & -\frac{5103}{18656} & & \\
1 & \frac{35}{384} & 0 & \frac{500}{1113} & \frac{125}{192} & -\frac{2187}{6784} & \frac{11}{84} & \\
\hline
\hat{b}_i & \frac{35}{384} & 0 & \frac{500}{1113} & \frac{125}{192} & -\frac{2187}{6784} & \frac{11}{84} & 0 \\
\hline
b_i & \frac{5179}{57600} & 0 & \frac{7571}{16695} & \frac{393}{640} & -\frac{92097}{339200} & \frac{187}{2100} & \frac{1}{40}
\end{array}
\qquad (25.18)
$$

Удобство таких методов состоит в том, что для получения оценки погрешности требуется минимальное количество дополнительных вычислений: первая строчка коэффициентов $\hat{b}_i$ в этой таблице определяет приближение $\hat{y}_1$ порядка 5,

$$\hat{y}_1 = y_0 + h\sum_i \hat{b}_i \kappa_i,$$

а вторая — y_1 порядка 4 для оценки погрешности, то есть

$$err = h\sum_i (\hat{b}_i - b_i)\kappa_i.$$

26 Экстраполяционные методы

26.1 Глобальная погрешность одношаговых методов

Глобальной погрешностью одношагового метода решения задачи Коши называется погрешность, накопленная после нескольки последовательных шагов.

Рассмотрим пошаговый процесс численного интегрирования задачи (22.1):

$$y_{k+1} = \Phi(x_k, y_k, x_{k+1}), \quad k = 0, 1, \ldots, N-1.$$

Обозначим $x_N = x_0 + H = X$. Глобальной погрешностью метода Φ называют величину

$$y(X) - y_N.$$

Нас интересует зависимость величины глобальной погрешности от частоты сетки $\{x_k\}$. Для этого удобно рассмотреть равномерную сетку

$$\{x_k = x_0 + kh\}_{k=0}^{N}, \quad h = H/N.$$

Тогда величина h является естественным рычагом управления точностью, поэтому рассмотрим зависимость глобальной погрешности от h. Приближённое решение $y_N \approx y(X)$ обозначим $\tilde{y}(X, h)$ и, не вдаваясь в технические детали, приведём следующий важный результат.

Теорема 26.1 (Асимптотическое разложение глобальной погрешности). *Пусть метод Φ обладает свойством $\Phi(x_0, y_0, x_0) = y_0$ и имеет порядок p, а его функция φ является достаточно гладкой. Тогда глобальная погрешность может быть представлена в виде*

$$\tilde{y}(X, h) - y(X) = \\ = c_p(X)h^p + c_{p+1}(X)h^{p+1} \ldots + c_{p+q}(X)h^{p+q} + O(h^{p+q+1}), \quad (26.1)$$

где величины $c_k(X)$ не зависят от h.

Таким образом, с точностью до слагаемого порядка $O(h^{p+q+1})$ величина $\tilde{y}(X, h)$ представляет собой многочлен степени $p+q$ по степеням h:

$$\tilde{y}(X, h) = U(h) + O(h^{p+q+1}), \quad (26.2)$$

где

$$U(h) = y(X) + a_p h^p + a_{p+1}h^{p+1} + \ldots + a_{p+q}h^{p+q}, \quad a_k = c_k(X). \quad (26.3)$$

Если бы нам был известен многочлен U, было бы легко найти точное решение:

$$y(X) = U(0).$$

26.2 Общая схема экстраполяционных методов

Отбрасывая остаточный член в (26.2), получаем возможность приблизить неизвестный многочлен U путём вычисления $\tilde{y}(X,h)$ при различных h. Точнее, рассмотрим набор целых чисел

$$N_0 < N_1 < N_2 < \ldots < N_q$$

и соответствующие им длины шага

$$h_0 > h_1 > \ldots > h_q, \quad h_i = H/N_i.$$

Выполняя $q+1$ интегрирований с шагами $\{h_i\}$, вычислим

$$\tilde{y}(X,h_i), \quad i = \overline{0,q},$$

и построим многочлен

$$P_q(h) = \alpha_0 + \alpha_p h^p + \alpha_{p+1} h^{p+1} + \ldots + \alpha_{p+q-1} h^{p+q-1} \tag{26.3′}$$

такой, что

$$P_q(h_i) = \tilde{y}(X,h_i), \quad i = \overline{0,q}. \tag{26.4}$$

Понятно, что в общем случае для нахождения $\{\alpha_i\}$ нужно решить СЛАУ. Теперь в качестве приближения к $y(X)$ естественно взять значение

$$\alpha_0 = P_q(0).$$

Замечание 26.1. Вообще говоря, экстраполяцией называется вычисление значения интерполяционного многочлена в точке, находящейся за пределами отрезка интерполяции (у нас это $[h_q, h_0]$). Именно эта ситуация имеет место в данном случае, отсюда и название метода.

$\triangleright_1$ Проверьте, что при $q=1$, $N_0=1$, $N_1=2$ экстраполяционный метод эквивалентен процедуре повышения порядка точности из прошлой лекции.

Теорема 26.2. *Точность приближения решения задачи Коши величиной* $P_q(0)$ *соответствует методу порядка* $p+q$*:*

$$y(X) - P_q(0) = O(H^{p+q+1}).$$

Доказательство. Сопоставляя формулы (26.2), (26.3′) и (26.4), получаем

$$P_q(h_i) = U(h_i) + O(h_i^{p+q+1}), \quad i = \overline{0,q}.$$

Подставляя сюда явный вид многочленов P_q и U, получаем СЛАУ

$$\alpha_0 - y(X) + \sum_{j=0}^{q-1} (\alpha_{p+j} - a_{p+j}) h_i^{p+j} = a_{p+q} h_i^{p+q} + O(h_i^{p+q+1}), \quad i = \overline{0,q}.$$

Делая замену $h_i = H/N_i$, эту систему можно записать в виде

$$A\varepsilon = \Delta,$$

где

$$A = \begin{bmatrix} 1 & N_0^{-p} & \cdots & N_0^{-p-q+1} \\ 1 & N_1^{-p} & \cdots & N_1^{-p-q+1} \\ \vdots & \vdots & & \vdots \\ 1 & N_q^{-p} & \cdots & N_q^{-p-q+1} \end{bmatrix}, \quad \varepsilon = \begin{bmatrix} \alpha_0 - y(X) \\ (\alpha_p - a_p)H^p \\ \vdots \\ (\alpha_{p+q-1} - a_{p+q-1})H^{p+q-1} \end{bmatrix}.$$

$\triangleright_2$ Докажите, что матрица A невырождена при любых попарно различных N_i.

Компоненты вектора Δ имеют вид

$$\Delta_i = a_{p+q}h_i^{p+q} + O(h_i^{p+q+1}) = c_{p+q}(x_0 + H)h_i^{p+q} + O(h_i^{p+q+1})$$

Так как по определению $c_{p+q}(x_0) = 0$, имеем $c_{p+q}(x_0 + H) = O(H)$, откуда

$$\|\Delta\|_\infty = \max_i |\Delta_i| = O(H^{p+q+1}). \tag{26.5}$$

В итоге получаем

$$|y(X) - \alpha_0| \leqslant \|\varepsilon\|_\infty \leqslant \|A^{-1}\|_\infty \|\Delta\|_\infty = O(H^{p+q+1}).$$

■

26.3 Алгоритм Эйткена–Невилла

При $p = 1$ задача поиска многочлена P_q (26.3′) по условиям (26.4) становится классической задачей алгебраической интерполяции. В этом случае возможно эффективно вычислить последовательность

$$P_0(0), P_1(0), \ldots, P_q(0), \ldots$$

с помощью так называемого алгоритма Эйткена–Невилла, который является прародителем алгоритма вычисления разделённых разностей. Основывается этот алгоритм на лемме 2.1, которая в наших обозначениях примет следующий вид.

Лемма 26.1. *Пусть* $\{h_0, h_1, h_2, \ldots\}$ *— последовательность попарно различных узлов,* $\{\tilde{y}_0, \tilde{y}_1, \tilde{y}_2, \ldots\}$ *— соответствующая последовательность значений. Обозначим* $P_{j,k}$ *многочлен степени* j*, обладающий свойством*

$$P_{j,k}(h_i) = \tilde{y}_i, \quad \forall i = \overline{k, k+j},$$

Тогда

$$P_{j+1,k}(h) = \frac{(h-h_k)P_{j,k+1}(h) - (h-h_{j+k+1})P_{j,k}(h)}{h_{j+k+1}-h_k}. \tag{26.6}$$

Понятно, что по смыслу у нас $\tilde{y}_i = \tilde{y}(X, h_i)$. При $h = 0$ из (26.6) получаем

$$P_{j+1,k}(0) = \frac{h_{j+k+1}P_{j,k}(0) - h_k P_{j,k+1}(0)}{h_{j+k+1}-h_k}.$$

Обозначая $P_{j,k}(0) = T_{j,k}$ и учитывая $h_i = H/N_i$, получаем основное рекуррентное соотношение

$$\boxed{T_{j+1,k} = T_{j,k} + \frac{T_{j,k} - T_{j,k+1}}{(N_k/N_{j+k+1}) - 1}.} \tag{26.7}$$

Таким образом, начиная с $T_{0,k} = \tilde{y}(X, h_k)$ и применяя (26.7) можно вычислить $P_q(0) = T_{q,0}$ путём заполнения таблицы

$$\begin{array}{lllll} T_{0,0} & & & & \\ T_{0,1} & T_{1,0} & & & \\ \vdots & \vdots & \ddots & & \\ T_{0,q-1} & T_{1,q-2} & \cdots & T_{q-1,0} & \\ T_{0,q} & T_{1,q-1} & \cdots & T_{q-1,1} & T_{q,0} \end{array} \tag{26.8}$$

Аналогично теореме 26.2 нетрудно доказать, что *каждый* элемент j-го столбца построенной таблицы является приближением порядка $j+1$ к точному решению $y(X)$.

На практике таблица (26.8) вычисляется построчно: значение q заранее не фиксируется, а новые строки, соответствующие пересчёту численного решения с более мелким шагом, добавляются по мере необходимости. Для этого, аналогично случаю вложенных методов, используется оценка погрешности вида

$$err_j = \|T_{j,0} - T_{j-1,0}\|.$$

На основе этой оценки можно также выбирать длину шага H.

Последовательность $\{N_i\}$ может выбираться по-разному, но наиболее экономичной в общем случае считается «гармоническая последовательность»

$$N_i = i + 1.$$

$\triangleright_3$ Вычислите $T_{2,0}$ и $T_{3,0}$ для явного и неявного методов Эйлера. Постройте соответствующие таблицы Бутчера.

26.4 Метод Грэгга–Булирша–Штёра (ГБШ)

Экстраполяционный подход наиболее эффективен в случае, когда разложение глобальной погрешности вида (26.1) содержит *только чётные степени* h:

$$y(X,h) - y(X) = c_2(X)h^2 + c_4(X)h^4 \ldots + c_{2m}(X)h^{2m} + O(h^{2m+2}). \quad (26.9)$$

Сделав формальную замену $h^2 = \eta$, видим, что экстраполяционные приближения можно по-прежнему находить по схеме (26.8), причём при этом *каждый этап экстраполяции будет повышать порядок метода на два*. Слегка изменится лишь формула (26.7):

$$\boxed{T_{j+1,k} = T_{j,k} + \frac{T_{j,k} - T_{j,k+1}}{(N_k/N_{j+k+1})^2 - 1}.} \quad (26.7')$$

Таким образом, величина $T_{j,0}$ в данном случае будет иметь порядок $2j+2$.

Методов, обладающих разложением вида (26.9), не так уж и много. Среди них можно выделить неявный метод средней точки, а из явных методов наиболее хорошо себя зарекомендовал метод ГБШ, который имеет следующий вид:

$$\begin{gathered} y_1 = y_0 + hf(x_0, y_0), \\ y_{k+1} = y_{k-1} + 2hf(x_k, y_k), \quad k = \overline{1, N-1}, \\ \tilde{y}(X,h) = S(X,h) = \tfrac{1}{2}(y_N + y_{N-1} + hf(x_N, y_N)). \end{gathered} \quad (26.10)$$

При этом в качестве последовательности $\{N_i\}$ следует брать

$$\{2, 4, 6, 8, 12, 16, \ldots\}, \quad N_{i+1} = 2N_{i-1} \quad \text{при} \quad i \geqslant 2.$$

26.5 Методы Рунге–Кутты произвольного порядка

Нетрудно заметить, что каждое экстраполяционное приближение $T_{j,0}$ порождает некоторый метод Рунге–Кутты. На этом пути получаются следующие результаты:

- Методы ГБШ порождают методы РК чётного порядка p с количеством стадий $s = p^2/4 + 1$.
- Экстраполяция методов Эйлера даёт метод РК порядка p с количеством стадий $s = \frac{p(p-1)}{2} + 1$.

27 Многошаговые методы

27.1 Введение

По-прежнему рассматриваем задачу Коши (22.1):

$$y'(x) = f(x, y(x)), \quad y(x_0) = y_0.$$

Предположим, что нам известны значения

$$y_i \approx y(x_i), \quad \text{для} \quad i = \overline{0, k-1}, \quad k \geqslant 1, \quad x_{k-1} < x_k.$$

Многошаговым (k-шаговым) методом решения задачи Коши называется отображение

$$\Phi : \left\{ \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}, \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, \ldots, \begin{bmatrix} x_{k-1} \\ y_{k-1} \end{bmatrix}, x_k \right\} \mapsto y_k \approx y(x_k).$$

При $k = 1$, очевидно, данное определение превращается в определение одношагового метода.

Понятно, что в общем случае задачи Коши нам известно только y_0, следовательно многошаговые методы при численной реализации нуждаются в процедуре вычисления «стартовых» значений $y_1, \ldots, y_{k-1}$. Обычно для этого используются одношаговые методы. После этого уже возможно пошаговое применение метода Φ:

$$y_{n+1} = \Phi\left(\begin{bmatrix} x_{n-k+1} \\ y_{n-k+1} \end{bmatrix}, \ldots, \begin{bmatrix} x_n \\ y_n \end{bmatrix}, x_{n+1} \right), \quad n = k-1, k, \ldots \tag{27.1}$$

27.2 Многошаговые методы квадратурного типа

27.2.1 Общий вид методов

Большой объём доступной информации о решении, по сравнению с одношаговым случаем, порождает широкий ассортимент многошаговых методов. Мы начнём с рассмотрения класса методов, основанных на приближении интегрального соотношения[9] вида

$$y(x_k) = y(x_{k-\ell}) + \int_{x_{k-\ell}}^{x_k} f(x, y(x))dx, \quad 0 \leqslant \ell \leqslant k, \tag{27.2}$$

[9] Это соотношение эквивалентно исходному дифференциальному уравнению.

путём полиномиальной интерполяции подынтегральной функции. По понятным причинам такие методы будем называть методами квадратурного типа[10]. В дальнейшем будем использовать обозначение

$$f_i = f(x_i, y_i).$$

Введём множество индексов

$$J \subset \{0, 1, \ldots, k\}$$

и проинтерполируем подынтегральную функцию в (27.2) по точкам $\{x_i\}_{i \in J}$. Как результат получаем методы вида

$$y_k = y_{k-\ell} + \int\limits_{x_{k-\ell}}^{x_k} P(x)dx, \tag{27.3}$$

где P — интерполяционный многочлен, определяемый соотношениями

$$P(x_i) = f_i, \quad i \in J, \tag{27.4}$$

откуда, используя запись в форме Лагранжа, имеем

$$P(x) = \sum_{i \in J} f_i \Lambda_i(x), \tag{27.5}$$

$$\boxed{\Lambda_i = \prod_{j \in J \setminus \{i\}} \frac{x - x_j}{x_i - x_j}.} \tag{27.6}$$

Подставляя всё это в (27.3), получаем общий вид многошаговых методов квадратурного типа:

$$\boxed{y_k = y_{k-\ell} + \sum_{i \in J} f_i \int\limits_{x_{k-\ell}}^{x_k} \Lambda_i(x)dx.} \tag{27.7}$$

Таким образом, *для однозначного определения метода нужно задать параметры ℓ и J*. Заметим также, что построенные методы будут неявными в случае $k \in J$ (f_k зависит от y_k) и явными во всех остальных случаях.

Замечание 27.1. При реализации неявных многошаговых методов требуется решить систему из $N = n$ нелинейных уравнений (n — размерность интегрируемой системы ОДУ). Это выгодно отличает их от неявных методов РК, для которых, как мы помним, $N = sn$.

[10] Учтите, что это не общепринятый термин.

27.2.2 Явные методы Адамса

Исторически первыми многошаговыми методами считаются явные методы Адамса, которые представляют собой методы квадратурного типа с

$$\ell = 1 \quad \text{и} \quad J = \{0, 1, \ldots, k-1\}.$$

Таким образом из (27.7), (27.6) получаем

$$y_k = y_{k-1} + \sum_{i=0}^{k-1} f_i \int_{x_{k-1}}^{x_k} \Lambda_i(x)dx, \tag{27.8}$$

$$\Lambda_i(x) = \prod_{\substack{j=0 \\ j\neq i}}^{k-1} \frac{x - x_j}{x_i - x_j}. \tag{27.9}$$

$\triangleright_1$ Постройте двухшаговый явный метод Адамса для неравномерной сетки, обозначив $x_{i+1} - x_i = h_i$.

Классические методы Адамса строятся для равномерной сетки:

$$x_i = x_0 + ih.$$

В этом случае коэффициенты метода вычисляются полностью аналогично коэффициентам квадратурных формул Ньютона–Котеса. Различие состоит только в отрезке интегрирования. Напомним, что в пункте 14.3.1 для равноотстоящих узлов мы выводили формулу

$$S(\Lambda_i) = \int_{x_0}^{x_n} \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j} dx = \frac{(-1)^{n-i}}{i!(n-i)!} h \int_0^n \prod_{\substack{j=0 \\ j\neq i}}^{n} (t - j)dt,$$

которая получена заменой $x = x_0 + th$. Следовательно, явные методы Адамса (27.8) примут вид

$$\boxed{y_k = y_{k-1} + h\sum_{i=0}^{k-1} \beta_i f_i,} \tag{27.10}$$

где коэффициенты β_i вычисляются по формуле

$$\beta_i = \frac{(-1)^{k-i-1}}{i!(k-i-1)!} \int_{k-1}^{k} \prod_{\substack{j=0 \\ j\neq i}}^{k-1} (t - j)dt. \tag{27.11}$$

При $k = 1$, очевидно, получаем явный метод Эйлера. Для остальных значений $k \leqslant 4$ коэффициенты β_i приведены в следующей таблице.

k	β_0	β_1	β_2	β_3
2	$-\frac{1}{2}$	$\frac{3}{2}$	$\times$	$\times$
3	$\frac{5}{12}$	$-\frac{16}{12}$	$\frac{23}{12}$	$\times$
4	$-\frac{9}{24}$	$\frac{37}{24}$	$-\frac{59}{24}$	$\frac{55}{24}$

$\triangleright_2$ Вычислите недостающие коэффициенты в таблице.

$\triangleright_3$ Вычислите сумму всех β_i для каждого k и объясните результат.

27.2.3 Неявные методы Адамса

Неявные методы Адамса получаются при

$$\ell = 1 \quad \text{и} \quad J = \{0, 1, \ldots, k\}.$$

В точности повторяя предыдущие рассуждения, получаем общий вид этих методов:

$$\boxed{y_k = y_{k-1} + h \sum_{i=0}^{k} \beta_i f_i,} \tag{27.12}$$

где

$$\beta_i = \frac{(-1)^{k-i}}{i!(k-i)!} \int_{k-1}^{k} \prod_{\substack{j=0 \\ j \neq i}}^{k} (t-j)\,dt. \tag{27.13}$$

Соответствующая таблица значений приведена ниже.

k	β_0	β_1	β_2	β_3
1	$\frac{1}{2}$	$\frac{1}{2}$	$\times$	$\times$
2	$-\frac{1}{12}$	$\frac{8}{12}$	$\frac{5}{12}$	$\times$
3	$\frac{1}{24}$	$-\frac{5}{24}$	$\frac{19}{24}$	$\frac{9}{24}$

Заметим, что при каждом k за счёт привлечения точки x_k мы имеем на один коэффициент больше, чем в явных методах Адамса.

$\triangleright_4$ Вычислите недостающие коэффициенты в таблице.

$\triangleright_5$ Почему явные методы Адамса иногда называют экстраполяционными, а неявные — интерполяционными?

27.2.4 Методы Нюстрёма и Милна–Симпсона

Следующие два семейства методов, которые мы вкратце рассмотрим, получаются при $\ell = 2$. В остальном они аналогичны явным и неявным методам Адамса. То есть, их общий вид это

$$\boxed{y_k = y_{k-2} + h\sum_{i=0}^{K}\beta_i f_i.} \tag{27.14}$$

При $K = k-1$ получаются методы Нюстрёма (явные), а при $K = k$ — методы Милна–Симпсона (неявные). Для вычисления коэффициентов этих методов используются формулы (27.11), (27.13) соответственно, с заменой нижнего предела интегрирования на $k-2$.

$\triangleright_6$ Постройте двухшаговый метод Милна–Симпсона.

27.3 Методы интерполяционного типа

Другой класс многошаговых методов основан на интерполяции приближенного решения по точкам $\{(x_i, y_i)\}_{i=0}^{k}$. Полученный многочлен обозначим Q,

$$Q(x_i) = y_i, \quad i = \overline{0,k}.$$

Для определения y_k на Q накладывается условие коллокации в узле y_k:

$$Q'(x_k) = f_k. \tag{27.15}$$

Замечание 27.2. В принципе, вместо x_k здесь можно взять любой другой доступный узел, но полученные таким образом явные методы не заслуживают внимания.

Описанные методы называются *формулами дифференцирования назад (ФДН)*, а также *методами Гира*. Займёмся выводом расчётных формул для случая равномерной сетки.

Для записи многочлена Q удобнее воспользоваться формой Ньютона, причём начиная с конца:

$$Q(x) = y_k + a_1(x-x_k) + a_2(x-x_k)(x-x_{k-1}) + \ldots + a_k(x-x_k)\ldots(x-x_0).$$

Здесь $a_i = f[x_k, \ldots, x_{k-i}]$ — разделённые разности:

$$a_1 = (y_k - y_{k-1})/h, \quad a_2 = (y_k - 2y_{k-1} + y_{k-2})/(2h^2), \quad \ldots$$

Для компактной записи этих величин нам потребуется следующее определение.

Пусть дана последовательность чисел $\{y_k\}_{k\in\mathbb{Z}}$. *Конечной разностью* порядка i для элемента y_k называется величина $\Delta^i y_k$, определяемая как

$$\begin{aligned}&\Delta^0 y_k = y_k,\\ &\Delta^i y_k = \Delta^{i-1}y_k - \Delta^{i-1}y_{k-1}, \quad i \geqslant 1.\end{aligned}$$

Таким образом, разделённые разности a_i по равноотстоящим узлам $x_i = x_0 + ih$ можно выразить через конечные разности:

$$a_i = \frac{\Delta^i y_k}{i! h^i}.$$

То есть Q принимает вид

$$Q(x) = \sum_{i=0}^{k} \frac{\Delta^i y_k}{i! h^i} \omega_i(x), \quad \omega_i(x) = (x - x_k) \dots (x - x_{k-i+1}).$$

Подставляя полученное представление Q в условие коллокации (27.15), получаем

$$\sum_{i=0}^{k} \frac{\Delta^i y_k}{i! h^i} \omega_i'(x_k) = f_k.$$

Осталось вычислить $\omega_i'(x_k)$:

$$\omega_i'(x_k) = (x_k - x_{k-1}) \dots (x_k - x_{k-i+1}) = (i-1)! h^{i-1}.$$

В итоге имеем следующий общий вид методов ФДН:

$$\boxed{\sum_{i=1}^{k} \frac{1}{i} \Delta^i y_k = h f_k.} \tag{27.16}$$

$\triangleright_7$ Постройте методы ФДН для $k = 1, 2, 3$.

27.4 Важное замечание о форме записи методов

Рассмотрим, к примеру, явный метод Адамса (27.10):

$$y_k = y_{k-1} + h \sum_{i=0}^{k-1} \beta_i f_i.$$

Такая форма представления удобна при выводе формул и при анализе методов. Но для практической реализации удобнее переписать метод как правило вычисления y_{n+1} в виде (27.1), то есть в нашем случае как

$$y_{n+1} = y_n + h \sum_{i=0}^{k-1} \beta_{k-i} f_{n-i}.$$

$\triangleright_8$ Перепишите в аналогичном виде остальные методы, рассмотренные выше.

28 Порядок и устойчивость многошаговых методов

28.1 Общий вид линейных многошаговых методов

Рассмотрим трёхшаговый явный метод Адамса

$$y_3 = y_2 + h\left(\frac{23}{12}f_2 - \frac{16}{12}f_1 + \frac{5}{12}f_0\right)$$

и, скажем, трехшаговый неявный метод ФДН:

$$\frac{11}{6}y_3 - 3y_2 + \frac{3}{2}y_1 - \frac{1}{3}y_0 = hf_3.$$

Оба этих метода, равно как и остальные линейные многошаговые методы (с постоянным шагом) подпадают под следующее определение.

Линейными многошаговыми (k-шаговыми) методами называются методы вида

$$\alpha_k y_k + \alpha_{k-1}y_{k-1} + \ldots + \alpha_0 y_0 = h(\beta_k f_k + \beta_{k-1}f_{k-1} + \ldots + \beta_0 f_0). \qquad (28.1)$$

При этом считается, что выполнены условия

$$\alpha_k \neq 0, \quad |\alpha_0| + |\beta_0| \neq 0. \qquad (28.2)$$

Напоминаем, что из соотношения (28.1) при реализации метода выражается величина y_k. Первое из накладываемых условий (28.2) означает, что это возможно, по крайней мере при достаточно маленьких h.

$\triangleright_1$ Зачем нужно второе условие?

Заметим также, что при $\beta_k = 0$ многошаговый метод будет явным, при остальных значениях — неявным.

28.2 Порядок точности многошаговых методов

28.2.1 Два способа определения погрешности

Для многошаговых методов существует два способа определения погрешности. Первое определение аналогично одношаговому случаю.

Локальной погрешностью многошагового метода называется величина

$$y(x_k) - y_k,$$

где y — точное решение задачи Коши $y'(x) = f(x, y(x))$, $y(x_0) = y_0$, а y_k — приближённое решение, полученное по формуле (28.1) при точных стартовых значениях $y_i = y(x_i)$, $i = \overline{0, k-1}$.

Вычисление определённой таким образом погрешности становится проблематичным в случае неявных методов (почему?), поэтому вместо локальной погрешности часто рассматривают более простую величину, называемую погрешностью аппроксимации.

Пусть функция y — точное решение рассматриваемой задачи Коши. Величину

$$\psi(y, x_0, h) = \sum_{i=0}^{k} \Big(\alpha_i y(x_0 + ih) - h\beta_i y'(x_0 + ih) \Big), \tag{28.3}$$

принято называть *погрешностью аппроксимации* метода (28.1). Как видно, она представляет собой невязку, полученную при подстановке точного решения задачи в уравнение (28.1).

Помимо (28.3) можно также рассмотреть $\psi(\cdot\,, x_0, h)$ — линейный оператор (функционал), в качестве аргумента принимающий дифференцируемые функции, определённые на интервале $[x_0, x_0 + kh]$.

Покажем связь между локальной погрешностью и погрешностью аппроксимации. Для простоты рассмотрим лишь скалярный случай.

Лемма 28.1. *Рассмотрим задачу Коши для скалярного ОДУ*

$$y'(x) = f(x, y(x)), \quad y(x_0) = y_0.$$

Пусть функции f и y непрерывно дифференцируемы. Тогда локальная погрешность линейного многошагового метода представима в виде

$$y(x_k) - y_k = \Big(\alpha_k - h\beta_k \tfrac{\partial}{\partial y} f(x_k, \eta) \Big)^{-1} \psi(y, x_0, h),$$

где η — некоторая точка между $y(x_k)$ и y_k.

Доказательство. Так как при определении локальной погрешности считается, что $y_i = y(x_i)$, $i = \overline{0, k-1}$, метод (28.1) можно записать в виде

$$\sum_{i=0}^{k-1} \Big(\alpha_i y(x_i) - h\beta_i f(x_i, y(x_i)) \Big) = -\alpha_k y_k + h\beta_k f(x_k, y_k).$$

Сумму в левой части формулы выразим через $\psi(y, x_0, h)$ и получим

$$\psi(y, x_0, h) = \alpha_k (y(x_k) - y_k) - h\beta_k \Big(f(x_k, y(x_k)) - f(x_k, y_k) \Big), \tag{28.4}$$

или

$$\psi(y, x_0, h) = F(y(x_k)) - F(y_k).$$

Применяя теорему о среднем к

$$F(u) = \alpha_k u - h\beta_k f(x_k, u),$$

получаем утверждение теоремы. ■

Замечание 28.1. В случае системы ОДУ справедлива практически идентичная формула (см. теорему о среднем для вектор-функций из последней лекции по ВМА).

Помимо основного результата теоремы интерес представляет также соотношение (28.4), которое можно записать в виде

$$\boxed{y(x_k) - y_k = \alpha_k^{-1}\psi(y, x_0, h) + O(h).}$$

Это означает, что величина $\alpha_k^{-1}\psi(y, x_0, h)$ является *главной частью локальной погрешности* метода. Заметим также, что для явных методов эта величина *совпадает* с локальной погрешностью.

28.2.2 Порядок точности многошаговых методов

В силу рассмотренной выше связи между локальной погрешностью и погрешностью аппроксимации порядок точности линейных многошаговых методов можно определить двумя эквивалентными способами.

Говорят, что многошаговый метод имеет порядок p, если выполнено одно из следующих условий:

- $y(x_k) - y_k = O(h^{p+1})$ для всех достаточно гладких ОДУ.
- Для всех достаточно гладких функций u выполняется

$$\psi(u, x, h) = O(h^{p+1}).$$

Ещё раз подчеркнём, что оба условия эквивалентны в силу леммы 28.1. Проще всего проверить второе, которое в свою очередь можно переписать в виде

$$\frac{\partial^q}{\partial h^q}\psi(u, x, h)\bigg|_{h=0} = 0 \quad \forall q = \overline{0, p}. \tag{28.5}$$

Исходя из этого, нетрудно получить искомые условия на коэффициенты метода.

Теорема 28.1. *Многошаговый метод* (28.1) *имеет порядок* p *тогда и только тогда, когда*

$$\boxed{\sum_{i=0}^{k} \alpha_k = 0 \quad \text{и} \quad \sum_{i=0}^{k} \alpha_i i^q = q \sum_{i=0}^{k} \beta_i i^{q-1} \quad \forall\, q = \overline{1, p}.} \tag{28.6}$$

Доказательство. Доказательство основано на непосредственном использовании определения (28.5), то есть на соотношениях

$$\frac{\partial^q}{\partial h^q} \sum_{i=0}^{k} \Big(\alpha_i u(x + ih) - h\beta_i u'(x + ih)\Big)\bigg|_{h=0} = 0.$$

Для $q = 0$, очевидно, получаем условие

$$\sum_{i=0}^{k} \alpha_k = 0.$$

Для $q > 1$:

$$\frac{\partial^q}{\partial h^q} \sum_{i=0}^{k} \Big(\alpha_i u(x + ih) - h\beta_i u'(x + ih) \Big) \Big|_{h=0} = \\ = \sum_{i=0}^{k} \Big(\alpha_i i^q u^{(q)}(x) - \beta_i q i^{q-1} u^{(q)}(x) \Big) = 0,$$

откуда сразу следуют остальные условия порядка (28.6). ■

28.2.3 Порядок методов Адамса

Покажем применение полученных условий порядка на примере неявных методов Адамса. Напомним, что эти методы имеют общий вид

$$y_k = y_{k-1} + h \sum_{i=0}^{k} \beta_i f_i,$$

Отсюда

$$\alpha_k = 1, \quad \alpha_{k-1} = -1, \quad \alpha_{k-2} = \ldots = \alpha_0 = 0.$$

Условие нулевого порядка выполнено ($\alpha_k + \alpha_{k-1} = 0$), а для остальных q получаем

$$\sum_{i=0}^{k} \beta_i i^{q-1} = \frac{1}{q}(k^q - (k-1)^q). \tag{28.7}$$

Теперь нужно просто вспомнить о том, что коэффициенты β_i (27.13) по построению являются коэффициентами интерполяционной квадратурной формулы

$$\sum_{i=0}^{k} \beta_i F(i) \approx \int_{k-1}^{k} F(x)dx.$$

Так как эта формула интерполяционная, она имеет АСТ не менее k. Полагая $F(x) = x^{q-1}$, получаем, что (28.7) выполняется как минимум для всех q от 1 до $k+1$. Следовательно, *неявный k-шаговый метод Адамса имеет порядок точности не менее $k+1$*. Не очень трудно показать, что более высокого порядка достичь нельзя.

▷$_2$ Определите порядок явных методов Адамса.

▷$_3$ Постройте явный двухшаговый метод максимально возможного порядка.

28.3 Устойчивость многошаговых методов

Оказывается, высокого порядка точности не достаточно для того, чтобы метод был пригоден для практических вычислений. Некоторые, на первый взгляд хорошие методы, на практике дают «расходящееся» (уходящее на бесконечность) численное решение, не имеющее ничего общего с точным. Такое поведение — классическое проявление *неустойчивости* метода.

Рассмотрим задачу Коши на отрезке $[x_0, x_0 + H]$ и какой-нибудь многошаговый метод с постоянным шагом $h = H/N$. Понятно, что для достижения высокой точности необходимо брать $N \to \infty$, то есть $h \to 0$. При этом необходимым условием сходимости численного решения к точному будет ограниченность:

$$|y_N| \leqslant C \quad \text{при} \quad N \to \infty.$$

Понятно, что здесь $y_N \approx y(x_0 + H)$ — приближенное решение, полученное нашим методом при соответствующем $h = H/N$. При $h \to 0$ многошаговый метод общего вида

$$\alpha_k y_k + \alpha_{k-1} y_{k-1} + \ldots + \alpha_0 y_0 = h(\beta_k f_k + \beta_{k-1} f_{k-1} + \ldots + \beta_0 f_0)$$

сводится, очевидно, к соотношению

$$\alpha_k y_k + \alpha_{k-1} y_{k-1} + \ldots + \alpha_0 y_0 = 0. \tag{28.8}$$

Эту формулу можно также интерпретировать как применение метода к уравнению $y' = 0$.

Многошаговый метод называется *нуль-устойчивым* (*D-устойчивым*), если для любых начальных значений y_0, y_1, …, y_{k-1} последовательность $\{y_i\}_{i=0}^{\infty}$, определяемая рекуррентным соотношением

$$\alpha_k y_{n+k} + \alpha_{k-1} y_{n+k-1} + \ldots + \alpha_0 y_n = 0, \quad n = 0, 1, 2, \ldots \tag{28.9}$$

является ограниченной.

28.3.1 Критерий нуль-устойчивости

Исследование рекуррентных последовательностей вида (28.9), называемых также разностными уравнениями, — классическая математическая задача. В частности, явный вид i-го члена последовательности $\{y_i\}$ даёт следующая теорема.

Теорема 28.2. *Пусть многочлен*

$$\rho(\xi) = \sum_{j=0}^{k} \alpha_j \xi^j$$

имеет корни $\xi_1, \ldots, \xi_\ell$ *кратностей* $m_1, \ldots, m_\ell$ *соответственно. Тогда общее решение задачи* (28.9) *определяется формулой*

$$y_i = p_1(i)\xi_1^i + \ldots + p_\ell(i)\xi_\ell^i,$$

где p_j *— многочлены степеней* $m_j - 1$. ■

Следовательно, элемент y_i будет ограниченным при $i \to \infty$ лишь при выполнении следующих условий.

Многошаговый метод является нуль-устойчивым если и только если все корни многочлена $\rho(\xi) = \sum_{j=0}^{k} \alpha_j \xi^j$ принадлежат единичному кругу комплексной плоскости, причём на границе этого круга нет кратных корней.

Нетрудно показать, что все методы Адамса устойчивы. Трудно показать, что методы ФДН устойчивы при $k \leqslant 6$ и неустойчивы при всех остальных k.

Прежде, чем приступить к новой теме, напомним, что любое обыкновенное дифференциальное уравнение (система уравнений) порядка m,

$$y^{(m)}(x) = f\Big(x, y^{(m-1)}(x), \dots, y'(x), y(x)\Big),$$

может быть сведено к системе первого порядка введением дополнительных переменных

$$u_j(x) = y^{(j)}(x), \quad j = \overline{1, m-1}.$$

Полученная эквивалентная система первого порядка будет, очевидно, иметь вид

$$\begin{cases} y'(x) = u_1(x), \\ u_1'(x) = u_2(x), \\ \dots, \\ u_{m-2}'(x) = u_{m-1}(x), \\ u_{m-1}'(x) = f\Big(x, u_{m-1}(x), \dots, u_1(x), y(x)\Big). \end{cases}$$

29 Численное решение краевых задач. Метод стрельбы

29.1 Двухточечные краевые задачи

29.1.1 Примеры

Знакомство с краевыми задачами для ОДУ начнём с рассмотрения пары примеров.

Уравнение полёта снаряда. В плоскости xOy рассмотрим полёт снаряда, выпущенного из точки $(0, 0)$ с начальной абсолютной скоростью v_0 под углом α к оси Ox. Траектория полёта описывается системой ОДУ

$$\begin{cases} y'(x) = \operatorname{tg}\theta(x), \\ v'(x) = -g\dfrac{\operatorname{tg}\theta(x)}{v(x)} - k\dfrac{v(x)}{\cos\theta(x)}, \\ \theta'(x) = -\dfrac{g}{v(x)^2}. \end{cases} \tag{29.1}$$

с начальными условиями

$$y(0) = 0, \quad v(0) = v_0, \quad \theta(0) = \alpha. \tag{29.2}$$

Здесь $(x, y(x))$ — координаты снаряда, $v(x)$ — скорость, $\theta(x)$ — угол между вектором скорости и осью Ox в точке $(x, y(x))$, g — гравитационная

постоянная (смасштабированная должным образом), k — коэффициент сопротивления воздуха.

Понятно, что решить данную задачу Коши можно с помощью любого из рассмотренных нами ранее численных методов. Однако с практической точки зрения более интересна другая задача: *определить траекторию полёта снаряда, при которой он поражает цель, находящуюся в точке* (X, Y). Соответствующее решение уравнения (29.1) вместо начальных условий (29.2) должно, очевидно, удовлетворять условиям

$$y(0) = 0, \quad y(X) = Y, \quad v(0) = v_0, \tag{29.3}$$

которые называются *краевыми условиями*. Уравнения (29.1) вместе с условиями (29.3) представляют собой типичный пример *двухточечной краевой задачи*.

Одномерное стационарное уравнение теплопроводности. На оси Ox рассмотрим теплопроводящий стержень с концами в точках a и b. Пусть $u(x)$ — температура стержня в точке $x \in [a, b]$. Из курса математической физики известно, что функция u удовлетворяет следующему ОДУ второго порядка:

$$-\Big(k(x)u'(x)\Big)' + q(x)u(x) = f(x), \quad a \leqslant x \leqslant b. \tag{29.4}$$

Здесь $k(x)$ — коэффициент теплопроводности, $q(x)$ — коэффициент теплоотдачи, $f(x)$ — плотность источников тепла.

Так как это уравнение второго порядка, для однозначного определения решения нужно задать два дополнительных условия. Простейший вариант — зафиксировать температуру на концах отрезка:

$$u(a) = u_a, \quad u(b) = u_b. \tag{29.5}$$

Такие условия называются *краевыми условиями первого рода*. В *условиях второго рода* на концах отрезка задаётся плотность потока тепла:

$$-k(a)u'(a) = w_a, \quad -k(b)u'(b) = w_b. \tag{29.6}$$

Понятно, что на разных концах могут быть заданы условия различного рода. Таким образом, (29.4)+(29.5) и (29.4)+(29.6) являются двухточечными краевыми задачами.

29.1.2 Общий вид двухточечных краевых задач

Теперь мы готовы к общей постановке двухточечных краевых задач.

Рассмотрим систему ОДУ

$$y'(x) = f(x, y(x)), \quad x \in [a, b], \tag{29.7a}$$

где y и f — вектор-функции:

$$y(x) = \begin{bmatrix} y_1(x) \\ \vdots \\ y_n(x) \end{bmatrix}, \quad f(x, y(x)) = \begin{bmatrix} f_1(x, y(x)) \\ \vdots \\ f_n(x, y(x)) \end{bmatrix}.$$

Зададим n условий (связей), которым должно удовлетворять решение системы (29.7a):

$$G_i\Big(y(a), y(b)\Big) = 0, \quad i = \overline{1, n}. \tag{29.7b}$$

Уравнения (29.7a), (29.7b) определяют *двухточечную краевую задачу для ОДУ* общего вида.

$\triangleright_1$ Приведите рассмотренные выше краевые задачи для уравнения теплопроводности (29.4) к общему виду.

Замечание 29.1. Понятно, что помимо (вместо) краёв отрезка $[a, b]$ в условиях (29.7b) могут фигурировать и другие точки из этого отрезка. В этом случае получим более общую задачу, которую называют *многоточечной задачей*.

В случае, если функции f и G_i линейны, соответствующую краевую задачу также называют линейной. В частности, обе задачи для уравнения теплопроводности — линейны. Такие задачи, как правило, решаются существенно проще нелинейных.

29.2 Метод стрельбы

Приступим к рассмотрению первого метода решения граничных задач вида (29.7) — метода стрельбы. Это один из методов, основанных на сведении граничной задачи (которую мы не умеем решать) к последовательности задач Коши (которые мы решать уже научились).

29.2.1 Пример

Вспомним артиллерийскую краевую задачу (29.1), (29.3). Эту задачу можно переформулировать так: *найти такой угол наклона пушки* $\alpha = \alpha^*$, *при котором решение задачи Коши с условиями* (29.2) *проходит через точку* (X, Y). Решать такую задачу можно так же, как это происходит на войне. Обозначим $y(X, \alpha)$ значение $y(X)$, полученное при начальном условии $\theta(0) = \alpha$. Тогда алгоритм действий артиллериста прост:

1. Выбираем два значения α_- и α_+, для которых заведомо выполняется

$$y(X, \alpha_-) \leqslant Y \leqslant y(X, \alpha_+).$$

2. Полагаем $\alpha \leftarrow (\alpha_- + \alpha_+)/2$ и делаем пробный выстрел: решаем задачу Коши, находя $Y_\alpha = y(X, \alpha)$.
3. Если $Y_\alpha = Y$ — победа!
4. Если перелёт ($Y_\alpha > Y$), полагаем $\alpha_+ \leftarrow \alpha$, в противном случае — $\alpha_- \leftarrow \alpha$.
5. Переходим к шагу 2.

Это и есть метод стрельбы.Описанная схема, очевидно, есть не что иное как алгоритм метода бисекции для решения уравнения

$$F(\alpha) = y(X, \alpha) - Y = 0.$$

Понятно, что для ускорения сходимости можно применить более совершенные методы решения нелинейных уравнений. Например, метод секущих или Ньютона. Применение последнего, однако, затруднено необходимостью вычисления $F'(\alpha) = \frac{\partial}{\partial \alpha} y(X, \alpha)$.

Подведём локальный итог: *метод стрельбы заключается в последовательном поиске начальных условий, при которых решение задачи Коши будет удовлетворять поставленным краевым условиям*.

29.2.2 Общая схема метода стрельбы

В общем случае ситуация осложняется тем, что неизвестных начальных условий может быть несколько. В этом случае метод бисекции практически бесполезен и приходится использовать методы решения систем нелинейных уравнений: различные модификации метода Ньютона, метод Бройдена и т. п. Итак, приступим.

Пусть дана система ОДУ (29.7a), которую запишем в покомпонентной форме

$$y_i'(x) = f_i(x, y_1(x), \ldots, y_n(x)), \quad i = \overline{1, n}, \tag{29.8a}$$

и краевые условия, первые m из которых заданы в точке $x = b$, а остальные — в точке $x = a$, то есть являются начальными:

$$\begin{cases} y_i(b) = \beta_i, & i = \overline{1, m}, \\ y_i(a) = \alpha_i, & i = \overline{m+1, n}. \end{cases} \tag{29.8b}$$

Для решения данной краевой задачи определим вектор-функцию

$$F = (F_1, \ldots, F_m)^T : \mathbb{R}^m \to \mathbb{R}^m$$

следующим образом:

$$F_i(\alpha_1, \ldots, \alpha_m) = y_i(b, \alpha_1, \ldots, \alpha_m) - \beta_i, \quad i = \overline{1, m},$$

где $y_i(x, \alpha_1, \ldots, \alpha_m)$ — решение исходной системы ОДУ (29.8a) с *начальными условиями*

$$y_i(a) = \alpha_i, \quad i = \overline{\mathbf{1}, \mathbf{n}}. \tag{29.9}$$

Метод стрельбы решения краевой задачи (29.8) заключается в нахождении вектора $\alpha^* = (\alpha_1^*, \ldots, \alpha_m^*)$, такого, что

$$F(\alpha^*) = 0.$$

Полученное решение $y(x, \alpha_1^*, \ldots, \alpha_m^*)$ по построению будет являтся решением краевой задачи (29.8).

Обратим особое внимание, что каждое вычисление $F(\alpha)$ требует решения задачи Коши (29.8a), (29.9), что, вообще говоря, весьма трудоёмко. Более того, для реализации метода Ньютона, который считается наиболее эффективным методом решения систем уравнений, необходимо вычислять матрицу Якоби $\frac{\partial F}{\partial \alpha}$, которой у нас просто нет. Эту матрицу приходится приближать с помощью конечных разностей, что, во-первых, ненадёжно, и, во-вторых, очень накладно. Поэтому следует помнить, что метод стрельбы для краевых задач большой размерности а) очень трудоёмок и б) ненадёжен.

29.2.3 Совсем общая схема метода стрельбы

Метод можно применять не только в случае простейших краевых условий (29.8b), но и в общем случае (29.7b). Итак, в случае краевой задачи

$$y_i'(x) = f_i(x, y_1(x), \ldots, y_n(x)), \quad i = \overline{1, n}, \tag{29.10a}$$

$$G_i\Big(y(a), y(b)\Big) = 0, \quad i = \overline{1, n}, \tag{29.10b}$$

метод стрельбы выглядит следующим образом. Так как у нас по сути не задано ни одного начального условия ($m = n$), рассматриваем задачу Коши

$$y_i'(x) = f_i(x, y_1(x), \ldots, y_n(x)), \quad i = \overline{1, n}, \tag{29.11a}$$

$$y_i(a) = \alpha_i, \quad i = \overline{1, n}. \tag{29.11b}$$

Здесь $\alpha = (\alpha_1, \ldots, \alpha_n)^T$ — вектор начальных условий, который нужно выбрать так, чтобы решение задачи (29.11), которое обозначаем $y(x, \alpha)$, совпадало с решением (29.10). То есть, всё делаем по аналогии с рассмотренным выше случаем: находим α^* как решение системы нелинейных уравнений $F(\alpha) = 0$, где компоненты вектор-функции F имеют вид

$$F_i(\alpha_1, \ldots, \alpha_n) = F_i(\alpha) = G_i(\alpha, y(b, \alpha)), \quad i = \overline{1, n}.$$

Решение задачи (29.11) с $\alpha = \alpha^*$ и будет искомым решением краевой задачи (29.10).

30 Линейные краевые задачи

30.1 Общие сведения

Линейной краевой задачей называется задача вида

$$y_i'(x) = \sum_{j=1}^{n} a_{ij}(x) y_j(x) + f_i(x), \quad i = \overline{1, n}, \tag{30.1a}$$

$$\sum_{j=1}^{n} \Big(p_{ij}\, y_j(a) + q_{ij}\, y_j(b) \Big) = g_i, \quad i = \overline{1, n}, \tag{30.1b}$$

которую можно также кратко записать в матричном виде как

$$y'(x) = A(x)y(x) + f(x), \tag{30.2a}$$

$$Py(a) + Qy(b) = g. \tag{30.2b}$$

Понятно, что при $P = I$, $R = 0$, краевые условия (30.2b) превращаются в начальные.

Важным свойством линейных задач является известный результат из теории дифференциальных уравнений: каждое решение линейной неоднородной системы ОДУ (30.2a) может быть представлено в виде

$$y(x) = y^0(x) + \sum_{i=1}^{n} c_i y^i(x), \tag{30.3}$$

где[11] y^0 — частное (т. е. любое) решение неоднородного уравнения (30.2a), $\{y^i\}_{i=1}^n$ — фундаментальная (линейно независимая) система решений однородного уравнения

$$y'(x) = A(x)y(x),$$

c_i — некоторые вещественные коэффициенты. Следующий метод напрямую использует данное свойство.

30.2 Метод редукции

30.3 Общая схема

Предположим, что нам известны функции y^i, $i = \overline{0, n}$ из (30.3). Тогда решение краевой задачи (30.2) можно получить, просто подставив (30.3)

[11] Здесь все y^i являются вектор-функциями, поэтому индекс ставим вверху.

в граничные условия (30.2b) и решив полученную систему линейных уравнений для нахождения $\{c_i\}_{i=1}^n$:

$$P\Big(y^0(a) + \sum_{i=1}^n c_i y^i(a)\Big) + Q\Big(y^0(b) + \sum_{i=1}^n c_i y^i(b)\Big) = g,$$

откуда

$$\sum_{i=1}^n c_i \Big(Py^i(a) + Qy^i(b)\Big) = g - Py^0(a) - Qy^0(b),$$

или совсем кратко

$$Rc = d, \tag{30.4}$$

где $c = (c_1, \ldots, c_n)^T$,

$$R = PY(a) + QY(b), \quad d = g - Py^0(a) - Qy^0(b),$$

а $Y(x)$ — матрица, составленная из столбцов $y^i(x)$, $i = \overline{1,n}$.

Таким образом, дело стало за нахождением базисных функций y^0 и $\{y^i\}_{i=1}^n$. Найти эти функции можно просто решив соответствующие им задачи Коши. Начнём с y^0.

Как мы помним, y^0 может быть *любым* решением неоднородного уравнения (30.2a), поэтому зададим простейшие (нулевые) начальные условия, то есть найдём y_0 как решение задачи Коши

$$\begin{aligned} &(y^0)'(x) = A(x)y^0(x) + f(x), \\ &y^0(a) = 0. \end{aligned} \tag{30.5}$$

Далее, каждое y^i является решением однородного уравнения

$$y'(x) = A(x)y(x).$$

Причём начальные условия мы можем выбирать как угодно, главное чтобы полученные решения были линейно независимыми. Для этого достаточно потребовать, чтобы соответствующая матрица $Y(a)$ была невырожденной, а проще всего — единичной. Поэтому базисные функции y^i определяем как решения задач Коши вида

$$\begin{aligned} &(y^i)'(x) = A(x)y^i(x), \\ &y^i_j(a) = \delta_{ij}, \quad j = \overline{1,n}. \end{aligned} \tag{30.6}$$

Вот, собственно, и весь метод. Соберём всё вместе: **алгоритм метода редукции** для решения задачи (30.2) выглядит следующим образом.

1. Решаем задачу Коши (30.5).

2. Решаем n задач Коши (30.6) для $i = \overline{1, n}$.

3. Решаем СЛАУ (30.4), находим $\{c_i\}$.

4. Записываем решение в виде (30.3).

Таким образом, метод редукции сводит решение линейной краевой задачи к решению $(n+1)$ начальных задач (задач Коши).

Замечание 30.1. Метод редукции применим также в случае нелинейных краевых условий. В этом случае, естественно, вместо СЛАУ (30.4) для нахождения $\{c_i\}$ получим систему нелинейных уравнений.

Замечание 30.2. При численном решении задач Коши (30.5), (30.6), сетки узлов должны совпадать или по крайней мере иметь достаточное количество общих точек. В противном случае вычисление решения по формуле (30.3) будет затруднено.

30.4 Случай присутствия начальных условий

Метод редукции можно упростить в случае, когда часть краевых условий является начальными, то есть если условия (30.1b) можно привести к виду

$$\sum_{j=1}^{n} \Big(p_{ij}\, y_j(a) + q_{ij}\, y_j(b) \Big) = g_i, \quad i = \overline{1, m}, \tag{30.7}$$

$$y_i(a) = g_i, \quad i = m+1, n. \tag{30.8}$$

В такой ситуации решение можно представить в виде

$$y(x) = y^0(x) + \sum_{i=1}^{m} c_i y^i(x), \tag{30.9}$$

где y^0 определено как решение задачи Коши

$$\begin{aligned} &(y^0)'(x) = A(x) y^0(x) + f(x), \\ &y^0_j(a) = 0, \quad j = \overline{1, m} \\ &y^0_j(a) = g_j, \quad j = \overline{m+1, n}, \end{aligned} \tag{30.10}$$

а y^i — как решения задач

$$\begin{aligned} &(y^i)'(x) = A(x) y^i(x), \\ &y^i_j(a) = \delta_{ij}, \quad j = \overline{1, m}, \\ &y^i_j(a) = 0, \quad j = \overline{m+1, n}. \end{aligned} \tag{30.11}$$

Понятно, что при таком выборе базиса решение, определяемое (30.9), всегда будет удовлетворять начальным условиям (30.8). За счёт выбора параметров $\{c_i\}$ можно добиться выполнения и остальных (граничных) условий. Для этого подставляем (30.9) в (30.7) и решаем полученную СЛАУ.

$\triangleright_1$ Запишите вид этой СЛАУ.

Видим, что в данной ситуации нам необходимо решать только $m+1$ задачу Коши вместо $n+1$.

31 Проекционные методы решения граничных задач

31.1 Введение

В этом разделе мы будем в основном рассматривать линейную краевую задачу для ОДУ второго порядка с условиями первого рода:

$$u''(x) + p(x)u'(x) + q(x)u(x) = f(x), \tag{31.1a}$$
$$u(a) = A, \quad u(b) = B. \tag{31.1b}$$

Здесь p, q и f — некоторые непрерывные функции. Считаем, что решение задачи существует и единственно.

А вот метод решения этой задачи будем формулировать в общем виде. Для этого рассмотрим соответствующий уравнению (31.1a) линейный дифференциальный оператор

$$\mathcal{L} \,:\, u \mapsto u'' + pu' + qu. \tag{31.2}$$

Рассмотрим также линейные функционалы

$$\mu_x \,:\, u \mapsto u(x). \tag{31.3}$$

В таких обозначениях задача (31.1) принимает вид

$$\mathcal{L}u = f, \tag{31.4a}$$
$$\mu_a(u) = A, \quad \mu_b(u) = B. \tag{31.4b}$$

Понятно, что в общем случае оператор $\mathcal{L}$ и функционалы μ_a, μ_b могут быть и другими. Как нетрудно догадаться, для численного решения этой задачи мы будем использовать материал из лекции 20.

31.2 Общая схема метода

31.2.1 Flashback: проекционный метод

Напомним, что общая схема применения проекционного метода к решению уравнения $\mathcal{L}u = f$ выглядит так: приближенное решение ищется в виде

$$\tilde{u} = \sum_{i=0}^{n} \alpha_i \varphi_i, \quad \tilde{u} \in U_n = \operatorname{span}\{\varphi_i\},$$

где $\{\varphi_i\}$ — базисные функции, а неизвестные коэффициенты $\{\alpha_i\}$ находятся как решение СЛАУ

$$\lambda_i(\mathcal{L}\tilde{u} - f) = 0 \quad \Leftrightarrow \quad \sum_j \alpha_j \lambda_i(\mathcal{L}\varphi_j) = \lambda_i(f), \quad i = \overline{0, n}. \tag{31.5}$$

см. формулу (20.4).Здесь, как мы помним, $\{\lambda_i\}_{i=0}^n$ — линейные функционалы, определяющие Π (оператор проектирования на подпространство U_n) следующим образом: $\lambda_i(\Pi v) = \lambda_i(v)$, $i = \overline{0,n}$.

31.2.2 Обработка граничных условий

Единственным препятствием на пути непосредственного применения описанной схемы проекционного метода к решению задачи (31.4) является наличие граничных условий (31.4b). В этой ситуации на приближенное решение $\tilde{u}$ в дополнение к условиям (31.5) накладываются условия

$$\mu_a(\tilde{u}) = A, \quad \mu_b(\tilde{u}) = B, \tag{31.6}$$

которые, очевидно, представляют собой два линейных уравнения

$$\sum_{i=0}^{n} \alpha_i \mu_a(\varphi_i) = A, \qquad \sum_{i=0}^{n} \alpha_i \mu_b(\varphi_i) = B. \tag{31.7}$$

Понятно, что для того, чтобы система (31.5), (31.7) была совместна, необходимо уравнять число уравнений с числом неизвестных. Для этого просто уберём два условия из (31.5). В итоге получим СЛАУ

$$\begin{cases} \sum_{j=0}^{n} \alpha_j \lambda_i(\mathcal{L}\varphi_j) = \lambda_i(f), \quad i = \overline{0,n-2}, \\ \sum_{i=0}^{n} \alpha_i \mu_a(\varphi_i) = A, \\ \sum_{i=0}^{n} \alpha_i \mu_b(\varphi_i) = B. \end{cases} \tag{31.8}$$

Замечание 31.1. В первых $n-1$ уравнениях этой системы индекс i не обязан всегда принимать значения от 0 до $n-2$. Часто берут $i = \overline{1,n-1}$, что, конечно, никак не влияет на способ решения.

Таким образом, *приближенное решение задачи* (31.4) *проекционным методом сводится к решению СЛАУ* (31.8).

31.3 Метод Галеркина

Рассмотрим, во что превращается описанная общая схема для в случае использования среднеквадратичного приближения: $\lambda_i(v) = (v, \varphi_i)$. Непосредственно из (31.8) получаем знакомые формулы

$$\begin{cases} \sum_{j=0}^{n} \alpha_j (\mathcal{L}\varphi_j, \varphi_i) = (f, \varphi_i), \quad i = \overline{0,n-2}, \\ \sum_{i=0}^{n} \alpha_i \mu_a(\varphi_i) = A, \\ \sum_{i=0}^{n} \alpha_i \mu_b(\varphi_i) = B. \end{cases} \tag{31.9}$$

31.3.1 Метод конечных элементов

Этот метод в настоящее время стал таким же брендом, как и методы Рунге–Кутты (может даже и более дорогим). Как это обычно бывает, за этими словами стоит нечто достаточно абстрактное. Так вот, «метод конечных элементов» — это просто метод Галёркина, где в качестве базиса взяты функции с компактным носителем. Этот метод очень хорошо себя зарекомендовал в огромном количестве прикладных задач за счёт того, что СЛАУ, которые получаются при его реализации, как правило достаточно хорошо обусловлены и разрежены.

Рассмотрим простейший пример этого метода на задаче (31.1). В качестве базиса возьмём фундаментальные сплайны первого порядка на равномерной сетке с шагом $h = (b-a)/n$:

$$x_i = a + ih, \quad \varphi_i(x) = \varphi(x - x_i), \quad \varphi(x) = \begin{cases} 1 - |x|/h, & |x| \leqslant h, \\ 0, & |x| > h. \end{cases} \tag{31.10}$$

В этом случае (31.9) будет иметь вид

$$\begin{cases} \sum\limits_{j=0}^{n} \alpha_j (\varphi_j'' + p\varphi_j' + q\varphi_j, \varphi_i) = (f, \varphi_i), \quad i = \overline{1, n-1}, \\ \sum_{i=0}^{n} \alpha_i \varphi_i(a) = A, \quad \sum_{i=0}^{n} \alpha_i \varphi_i(b) = B. \end{cases} \tag{31.11}$$

Причины, по которым здесь мы взяли $i = \overline{1, n-1}$, вскоре станут очевидными. С учётом $\varphi_i(x_j) = \delta_{ij}$ последние два уравнения, соответствующие граничным условиям, принимают простой вид

$$\alpha_0 = A, \quad \alpha_n = B. \tag{31.12}$$

Главная проблема при составлении остальных уравений состоит в том, что φ_i недифференцируемы. Ситуацию спасает хитрость и факт наличия операции интегрирования:

$$(\varphi_j'' + p\varphi_j' + q\varphi_j, \varphi_i) = \int_a^b \Big(\varphi_j''(x) + p(x)\varphi_j'(x) + q(x)\varphi_j(x)\Big)\varphi_i(x)\,dx. \tag{31.13}$$

Производная φ_j' не определена лишь в точках x_{j-1}, x_j и x_j. В остальных случаях имеем

$$\varphi_j'(x) = \begin{cases} h^{-1}, & x \in (x_{j-1}, x_j), \\ -h^{-1}, & x \in (x_j, x_j + 1), \\ 0 & \text{иначе}. \end{cases} \tag{31.14}$$

Поэтому коэффициент $\int_a^b p(x)\varphi_j'(x)\varphi_i(x)\,dx$ легко вычисляется, используя при необходимости квадратурные формулы.

Со второй производной всё не так просто. Если просто положить её равной нулю, ничего хорошего не выйдет. Поэтому, сугубо формально, воспользуемся интегрированием по частям:

$$\int_a^b \varphi_j'' \varphi_i dx = \varphi_j' \varphi_i \Big|_a^b - \int_a^b \varphi_j' \varphi_i' dx = -\int_a^b \varphi_j' \varphi_i' dx. \tag{31.15}$$

Здесь мы использовали очевидный факт: $\varphi_i(a) = \varphi_i(b) = 0$ для $i = \overline{1, n-1}$. Таким образом, с учетом компактности носителя базисных функций, формула (31.13) может быть записана в виде

$$(\varphi_j'' + p\varphi_j' + q\varphi_j, \varphi_i) = \int\limits_{\max\{x_{i-1}, x_{j-1}, x_0\}}^{\min\{x_{i+1}, x_{j+1}, x_n\}} (-\varphi_j' \varphi_i' + p\varphi_j' \varphi_i + q\varphi_j \varphi_i) dx. \tag{31.16}$$

Отсюда как итог получаем общий вид СЛАУ (31.11)

$$\begin{cases} d_0\alpha_0 + e_0\alpha_1 = b_1, \\ c_i\alpha_{i-1} + d_i\alpha_i + e_i\alpha_{i+1} = b_i, \quad i = \overline{1, n-1}, \\ c_n\alpha_{n-1} + d_n\alpha_n = b_n, \end{cases} \tag{31.17}$$

Эта система, естественно, решается методом прогонки.

$\triangleright_1$ Запишите формулы для вычисления коэффициентов b_i, c_i, d_i и e_i.

Замечание 31.2. Для того, чтобы сделать вывод формулы (31.16) правильным, нужно заменить дифференциальную формулировку исходной задачи интегральной: вместо решения уравнения $\mathcal{L}u = f$ нужно найти такое u, что

$$(\mathcal{L}u, \phi) = (f, \phi)$$

для всех ϕ, таких, что $\phi(a) = \phi(b) = 0$. Тогда интегрирование по частям будет обосновано.

31.4 Коллокационный метод

Как мы помним, коллокационный метод это проекционный метод, основанный на интерполяции: $\lambda_i(v) = v(x_i)$. В этом случае для задачи (31.1) получим СЛАУ

$$\begin{cases} \sum\limits_{j=0}^{n} \alpha_j \Big(\varphi_j''(x_i) + p(x_i)\varphi_j'(x_i) + q(x_i)\varphi_j(x_i) \Big) = f(x_i), \\ \sum_{i=0}^{n} \alpha_i \varphi_i(a) = A, \quad \sum_{i=0}^{n} \alpha_i \varphi_i(b) = B. \end{cases} \tag{31.18}$$

31.4.1 Коллокация с использованием кубических сплайнов

Необходимость честного вычисления производных φ_j приводит к тому, что базисные функции должны быть как минимум дважды дифференцируемы в узлах x_i Поэтому в качестве базисных функций возьмём кубические сплайны. Сетку будем по-прежнему считать равномерной: $x_i = a + ih$, $i = \overline{0,n}$.

В литературе как правило рекомендуется следовать общей схеме (31.18), в качестве базисных функций используя кубические B-сплайны N_i^3. При этом нужно будет вычислять производные от этих функций.

Мы же рассмотрим другую, более простую для вывода вычислительную схему. Она основана на непосредственном вычислении решения в явном виде:

$$\tilde{u}(x) = s(x) = s_i(x), \quad \text{при} \quad x \in \Delta_i = [x_{i-1}, x_i],$$

где

$$s_i(x) = \alpha_i + \beta_i(x - x_i) + \frac{\gamma_i}{2}(x - x_i)^2 + \frac{\delta_i}{6}(x - x_i)^3, \quad i = \overline{1,n}. \qquad (31.19)$$

Как обычно полагаем $\alpha_0 = s(x_0)$, $\beta_0 = s'(x_0)$, $\gamma_0 = s''(x_0)$. Таким образом у нас $\alpha_i = s(x_i)$, $\beta_i = s'(x_i)$, $\gamma_i = s''(x_i)$

Неизвестные коэффициенты будем находить точно так же, как и при построении интерполяционного кубического сплайна. Единственная разница состоит в том, что вместо условий интерполяции у нас будут условия коллокации

$$\mathcal{L}s(x_i) = f(x_i), \quad i = \overline{0,n}.$$

Добавляя сюда граничные условия $s(x_0) = A$, $s(x_n) = B$, получаем требуемые $n+3$ условия, достаточные для однозначного определения сплайна.

$\triangleright_2$ Вывести уравнения для определения неизвестных коэффициентов сплайна и предложить метод решения полученной СЛАУ.

32 Сеточный метод решения краевых задач

32.1 Общий нелинейный случай

Сейчас мы начнем рассмотрение одного из наиболее популярных (особенно на постсоветском пространстве) методов решения задач математической физики — метод сеток. Применять этот метод будем к нелинейной краевой задаче для ОДУ второго порядка:

$$u''(x) = g(x, u(x), u'(x)), \quad u(a) = A, \quad u(b) = B. \tag{32.1}$$

Здесь $u : \mathbb{R} \to \mathbb{R}$, соответственно $g : \mathbb{R}^3 \to \mathbb{R}$. Согласно методу сеток, приближенное решение этой задачи будем искать лишь в точках сетки $\{x_i\}_{i=0}^n$, которую чаще всего берут равномерной с шагом $h = (b-a)/n$:

$$x_i = a + ih, \quad i = \overline{0, n}. \tag{32.2}$$

Приближенное решение в этих узлах традиционно обозначают

$$y_i \approx u(x_i).$$

Так как по условию можно сразу положить $y_0 = A$, $y_n = B$, для нахождения $\{y_i\}_{i=1}^{n-1}$ необходимо из (32.1) получить $n-1$ уравнений, которые бы связывали между собой эти неизвестные. Для этого обычно используются следующие соотношения:

$$u'(x) \approx \frac{1}{2h}\Big(u(x+h) - u(x-h)\Big), \tag{32.3a}$$

$$u''(x) \approx \frac{1}{h^2}\Big(u(x+h) - 2u(x) + u(x-h)\Big). \tag{32.3b}$$

$\triangleright_1$ Получите данные формулу путём полиномиальной интерполяции с последующим дифференцированием.

Подставляя соотношения (32.3) в исходное уравнение (32.1), для $x = x_i$ получаем следующую систему нелинейных уравнений:

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} = g\Big(x_i, y_i, \frac{y_{i+1} - y_{i-1}}{2h}\Big), \quad i = \overline{1, n-1}, \tag{32.4a}$$

$$y_0 = A, \quad y_n = B. \tag{32.4b}$$

Решая эту систему любым из известных нам численных методов, получим искомое приближенное решение.

32.2 Линейный случай

Наиболее просто метод сеток реализуется в случае, когда ОДУ линейно:

$$u''(x) + p(x)u'(x) + q(x)u(x) = f(x), \quad u(a) = A, \quad u(b) = B, \tag{32.5}$$

то есть $g(x, u(x), u'(x)) = f(x) - p(x)u'(x) - q(x)u(x)$. Уравнения (32.4a) при этом превращаются в

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} = f_i - p_i \frac{y_{i+1} - y_{i-1}}{2h} - q_i y_i, \tag{32.6}$$

где $f_i = f(x_i)$, $p_i = p(x_i)$, $q_i = q(x_i)$. Полученные уравнения можно переписать в виде

$$\left(1 - \frac{p_i}{2}h\right)y_{i-1} + \left(-2 + q_i h^2\right)y_i + \left(1 + \frac{p_i}{2}h\right)y_{i+1} = h^2 f_i. \tag{32.7}$$

Обозначая

$$c_i = 1 - \frac{p_i}{2}h, \quad d_i = -2 + q_i h^2, \quad e_i = 1 + \frac{p_i}{2}h, \tag{32.8}$$

а также учитывая краевые условия, окончательно получаем следующую СЛАУ:

$$L_h y_h = f_h, \quad y_0 = A, \quad y_n = B, \tag{32.9}$$

где

$$L_h = \frac{1}{h^2}\begin{bmatrix} c_1 & d_1 & e_1 & & & \\ & c_2 & d_2 & e_2 & & \\ & & \ddots & \ddots & \ddots & \\ & & & c_{n-2} & d_{n-2} & e_{n-2} \\ & & & & c_{n-1} & d_{n-1} & e_{n-1} \end{bmatrix}, \tag{32.10}$$

$$y_h = (y_0, y_1, \ldots, y_{n-1}, y_n)^T, \quad f_h = (f_1, \ldots, f_{n-1})^T.$$

Система (32.9) представляет собой простейший пример того, что в дальнейшем мы будем называть *разностной схемой*. Понятно, что она сводится к трехдиагональной СЛАУ

$$\underbrace{\frac{1}{h^2}\begin{bmatrix} h^2 & 0 & & & & & \\ c_1 & d_1 & e_1 & & & & \\ & c_2 & d_2 & e_2 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & c_{n-2} & d_{n-2} & e_{n-2} & \\ & & & & c_{n-1} & d_{n-1} & e_{n-1} \\ & & & & & 0 & h^2 \end{bmatrix}}_{\tilde{L}_h} \underbrace{\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{n-2} \\ y_{n-1} \\ y_n \end{bmatrix}}_{y_h} = \underbrace{\begin{bmatrix} A \\ f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_{n-2} \\ f_{n-1} \\ B \end{bmatrix}}_{\tilde{f}_h}. \tag{32.11}$$

32.3 Устойчивость и сходимость метода сеток

Исследуем сходимость метода сеток для частного случая задачи (32.5) — задачи теплопроводности (29.4) с постоянным коэффициентом $k(x) = 1$:

$$\mathcal{L}u(x) = -u''(x) + q(x)u(x) = f(x), \quad u(a) = A, \quad u(b) = B. \tag{32.12}$$

В этом случае метод сеток приведет к СЛАУ вида (32.9), где коэффициенты матрицы L_h (из-за знака минус перед $u''(x)$) будут иметь немного отличный от (32.8) вид:

$$c_i = -1, \quad d_i = 2 + q_i h^2, \quad e_i = -1, \tag{32.13}$$

Таким образом, матрица полной системы (32.17) имеет вид

$$\tilde{L}_h = \frac{1}{h^2} \begin{bmatrix} h^2 & 0 & & & & & \\ -1 & d_1 & -1 & & & & \\ & -1 & d_2 & -1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & -1 & d_{n-2} & -1 & \\ & & & & -1 & d_{n-1} & -1 \\ & & & & & 0 & h^2 \end{bmatrix}. \tag{32.14}$$

В дальнейшем будем считать $q(x) \geqslant 0 \ \ \forall x \in [a, b]$, так что имеем

$$d_i \geqslant 2, \quad c_i + d_i + e_i \geqslant 0, \quad |d_i| \geqslant |c_i| + |e_i|, \qquad \forall i = \overline{1, n-1}. \tag{32.15}$$

Последнее свойство (диагонального преобладания) гарантирует существование и единственность решения СЛАУ (32.9), а также применимость метода прогонки для ее решения.

Наша задача — доказать, что метод сеток сходится, то есть что

$$\boxed{\max_{0 \leqslant i \leqslant n} |u(x_i) - y_i| = \|u_h - y_h\| \xrightarrow[h \to 0]{} 0.} \tag{32.16}$$

Здесь и далее $u_h = (u(x_0), u(x_1), \ldots, u(x_{n-1}), u(x_n))^T$ — вектор точных значений решения в узлах сетки, $\|\cdot\| = \|\cdot\|_\infty$.

Рассмотрим невязку, полученную при подстановке u_h в систему (32.17):

$$\psi_h = \tilde{L}_h u_h - \tilde{f}_h.$$

Такой вектор в дальнейшем будем называть *погрешностью аппроксимации*. Так как по условию

$$\tilde{L}_h y_h = \tilde{f}_h,$$

имеем

$$\tilde{L}_h(u_h - y_h) = \psi_h,$$

откуда

$$\|u_h - y_h\| \leqslant \|\tilde{L}_h^{-1}\| \, \|\psi_h\|.$$

Из последнего соотношения становятся очевидными **достаточные условия сходимости метода**: *для того, чтобы выполнялось* (32.16), *достаточно выполнения следующих двух условий:*

1. $\|\psi_h\| \xrightarrow[h\to 0]{} 0$ *(условие аппроксимации),*
2. $\|\tilde{L}_h^{-1}\| \leqslant M \leqslant \infty \;\; \forall h \leqslant h_0$, *где* M *не зависит от* h *(условие устойчивости).*

32.3.1 Аппроксимация

Докажем выполнение первого условия сходимости. Рассмотрим ψ_i — i-ю компоненту вектора $\psi_h = \tilde{L}_h u_h - \tilde{f}_h$. Пусть $i = \overline{1, n-1}$, тогда

$$\begin{aligned}
\psi_i &= -\frac{u_{i-1} + 2u_i - u_{i+1}}{h^2} + q_i u_i - f_i = \\
&= -\frac{u(x_i - h) + 2u(x_i) - u(x_i + h)}{h^2} + q(x_i)u(x_i) - f(x_i) = \\
&= \left[f(x_i) = -u''(x_i) + q(x_i)u(x_i)\right] = u''(x_i) - \frac{u(x_i - h) + 2u(x_i) - u(x_i + h)}{h^2}.
\end{aligned}$$

Раскладывая в ряд Тейлора величины $u(x_i \pm h)$, в итоге получаем отсюда

$$\psi_i = O(h^2), \quad i = \overline{1, n-1}.$$

Для $i = 0$ и $i = n$ имеем, очевидно, $\psi_i = 0$, так что окончательно

$$\|\psi_h\| = \max_i |\psi_i| = O(h^2) \xrightarrow[h\to 0]{} 0.$$

В таких случаях говорят, что *метод (схема) аппроксимирует дифференциальное уравнение со вторым порядком*.

32.3.2 Устойчивость

Для доказательства устойчивости, то есть равномерной ограниченности нормы матрицы $\tilde{L}_h^{-1}$, достаточно показать, что если

$$\tilde{L}_h y_h = \tilde{f}_h,$$

то

$$\|y_h\| \leqslant M\|\tilde{f}_h\|, \quad M < \infty,$$

для любых h.

Для этого нам понадобятся следующие свойства матрицы $\tilde{L}_h$.

Лемма 32.1 (Принцип максимума для трехдиагональных матриц). *Пусть вектор y_h является решением следующей СЛАУ (32.9), которую запишем в виде*

$$\begin{aligned} &c_i y_{i-1} + d_i y_i + e_i y_{i+1} = f_i, \quad i = \overline{1, n-1} \\ &y_0 = A, \quad y_n = B. \end{aligned}$$

для коэффициентов которой справедливы соотношения

$$c_i < 0, \quad d_i \geqslant 0, \quad e_i < 0, \quad c_i + d_i + e_i \geqslant 0.$$

Тогда если $A < 0$, $B < 0$ и $f_i < 0$ для всех $i = \overline{1, n-1}$, то

$$y_i \leqslant 0 \text{ для всех } i = \overline{1, n-1}.$$

Доказательство. Докажем лемму от потивного. Предположим, что существует

$$y_{\max} = \max_{0 \leqslant i \leqslant n} y_i > 0$$

и пусть j — максимальный индекс, при котором $y_j = y_{\max}$. Тогда $y_{j-1} \leqslant y_j$, $y_{j+1} < y_j$, следовательно

$$0 \leqslant c_j y_j + d_j y_j + e_j y_j \quad < \quad c_j y_{j-1} + d_j y_j + e_j y_{j+1} = f_j \leqslant 0.$$

Полученное противоречие ($0 < 0$) доказывает лемму. ■

Прямым следствием из этой леммы является следующая важная теорема.

Теорема 32.1. *Пусть векторы y_h и v_h удовлетворяют следующим свойствам:*

$$L_h y_h = f_h, \quad L_h v_h = g_h,$$

где коэффициенты матрицы L_h удовлетворяют условиям леммы 32.1. Пусть также

$$\begin{aligned} &|f_i| \leqslant g_i, \quad \forall i = \overline{0, n}, \\ &|y_0| \leqslant v_0, \quad |y_n| \leqslant v_n. \end{aligned}$$

Тогда

$$|y_i| \leqslant v_i \quad \forall i = \overline{0, n}.$$

$\triangleright_2$ Докажите теорему.

Теперь можно приступать к доказательству устойчивости. Итак, мы рассматриваем СЛАУ

$$\underbrace{\frac{1}{h^2}\begin{bmatrix} h^2 & 0 & & & & & \\ -1 & d_1 & -1 & & & & \\ & -1 & d_2 & -1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & -1 & d_{n-2} & -1 & \\ & & & & -1 & d_{n-1} & -1 \\ & & & & & 0 & h^2 \end{bmatrix}}_{\tilde{L}_h} \underbrace{\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{n-2} \\ y_{n-1} \\ y_n \end{bmatrix}}_{y_h} = \underbrace{\begin{bmatrix} A \\ f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_{n-2} \\ f_{n-1} \\ B \end{bmatrix}}_{\tilde{f}_h}, \tag{32.17}$$

$$d_i = 2 + q_i h^2.$$

Строим мажоранту для вектора $\tilde{f}_h$: нужно подобрать такой вектор $v_h = (v_0, \ldots, v_n)^T$, чтобы для него выполнялось $|\tilde{f}_i| \leqslant (\tilde{L}_h v_h)_i$. Тогда по теореме 32.1 можно будет установить соотношение $|y_i| \leqslant v_i$. Напомним, что наша конечная цель — получить оценку вида $\|y_h\| \leqslant M\|\tilde{f}_h\|$, Поэтому компоненты v_i нужно выражать через $|A|$, $|B|$ и $\|f_h\|$:

$$v_i = C + w_i, \quad i = \overline{0, n},$$

где

$$C = \max\{|A|, |B|\},$$
$$w_i = \frac{\|f_h\|}{2}(x_i - a)(b - x_i),$$
$$\|f_h\| = \max_{1 \leqslant i \leqslant n-1} |f_i|.$$

Вычислим компоненты вектора $g_h = \tilde{L}_h v_h$. Для $i = \overline{1, n-1}$ имеем

$$g_i = \frac{1}{h^2}\Big(-(C + w_{i-1}) + (2 + q_i h^2)(C + w_i) - (C + w_{i+1})\Big) = \\ = \frac{1}{h^2}\Big(-w_{i-1} + 2w_i - w_{i+1} + q_i h^2 (C + w_i)\Big).$$

Теперь заметим, что

$$-w_{i-1} + 2w_i - w_{i+1} = \|f_h\| h^2,$$

отсюда

$$g_i = \|f_h\| + q_i v_i, \quad i = \overline{1, n-1}.$$

Для $i = 0$, $i = n$, очевидно, получаем

$$g_0 = g_i = C.$$

Таким образом, так как $v_i \geqslant 0$, $c_i \geqslant 0$, из последних двух соотношений получаем долгожданное

$$|\tilde{f}_i| \leqslant g_i, \quad i = \overline{0, n},$$

откуда по теореме 32.1 имеем оценку

$$|y_i| \leqslant v_i = \max\{|A|, |B|\} + \frac{\|f_h\|}{2}(x_i - a)(b - x_i). \tag{32.18}$$

Остаётся заметить, что $\tilde{f}_0 = A$, $\tilde{f}_n = B$, а также

$$\max_{x \in [a,b]} (x - a)(b - x) = (b - a)^2/4.$$

Следовательно, из (32.18) имеем

$$\|y_h\| = \max_{0 \leqslant i \leqslant n} |y_i| \leqslant \|v_h\| \leqslant \max\{|\tilde{f}_0|, |\tilde{f}_n|\} + \|f_h\| \frac{(b-a)^2}{8},$$

откуда окончательно получаем

$$\boxed{\|y_h\| \leqslant \left(1 + \frac{(b-a)^2}{8}\right) \|\tilde{f}_h\|,} \tag{32.19}$$

причем эта оценка выполняется *при любых* h. Следовательно, мы доказали устойчивость метода.

Итак, метод сеток для задачи (32.12) имеет второй порядок аппроксимации, устойчив, и, следовательно, сходится.

33 Основные понятия теории разностных схем

33.1 Сетки и сеточные функции

В предыдущей лекции мы уже познакомились с основными принципами теории разностных схем (сеточного метода). Настало время их формализовать.

Линейной дифференциальной задачей в дальнейшем будем называть граничную (краевую) задачу для дифференциальных уравнений вида

$$\mathcal{L}u(x) = f(x), \quad x \in \Omega, \tag{33.1a}$$
$$\ell u(x) = \mu(x), \quad x \in \Gamma. \tag{33.1b}$$

Здесь Ω — некоторая область n-мерного евклидова пространства,
Γ — граница Ω,
$u \in H$ — неизвестная функция, $u : \bar{\Omega} \to \mathbb{R}$,
$\bar{\Omega} = \Omega \cup \Gamma$,
$\mathcal{L} : H \to H^f$ — линейный дифференциальный оператор,
$\ell : H \to H^\mu$ — линейный оператор, задающий граничные условия.

На множестве $\bar{\Omega}$ введём сетку узлов (множество точек), которую традиционно обозначают $\bar{\omega}_h$:

$$\bar{\omega}_h = \{x_i\}_{i=0}^N \subset \bar{\Omega}.$$

Сетку $\bar{\omega}_h$ естественно разбить на две части — на граничные и внутренние узлы:

$$\bar{\omega}_h = \omega_h \cup \gamma_h,$$
$$\omega_h = \bar{\omega}_h \cap \Omega, \quad \gamma_h = \bar{\omega}_h \cap \Gamma.$$

Здесь везде индекс h характеризует «плотность» сетки (в полной аналогии с h для равномерной сетки на отрезке). В дальнейшем будет удобно разделить множество индексов

$$J = \{0, 1, \ldots, N\}$$

на индексы внутренних и граничных узлов:

$$J = J_\omega \cup J_\gamma,$$

$$\omega_h = \{x_i\}_{i \in J_\omega}, \quad \gamma_h = \{x_i\}_{i \in J_\gamma}.$$

Мощность множеств J_ω и J_γ обозначим соответственно N_ω и N_γ.

Сеточной функцией, определённой на некоторой сетке ω, будем называть любое отображение

$$y_h : \omega \to \mathbb{R}.$$

Множество сеточных функций на сетке ω_h будем обозначать H_h, на сетке $\bar{\omega}_h$ — $\tilde{H}_h$.

Понятно, что $\tilde{H}_h \sim \mathbb{R}^{N+1}$, $H_h \sim \mathbb{R}^{N_\omega}$. Таким образом, по сути сеточная функция представляет собой просто упорядоченный набор вещественных чисел (вектор), в котором i-я компонента соответствует значению в узле x_i.

Основной принцип метода сеток — поиск приближенного решения задачи (33.1) в виде сеточной функции $y_h \in \tilde{H}_h$,

$$y_h = (y_0, y_1, \ldots, y_N)^T, \quad y_i \approx u(x_i).$$

Чтобы найти y_h, от дифференциальной задачи (33.1) переходят к так называемой *разностной задаче*, которая представляет собой систему линейных уравнений относительно компонент сеточной функции. Сейчас мы разберемся каким образом строится эта система.

33.2 Разностные схемы: общая формулировка

Рассмотрим дифференциальное уравнение (33.1a) в произвольном внутреннем узле сетки $\bar{\omega}_h$:

$$\mathcal{L}u(x_i) = f(x_i), \quad i \in J_\omega. \tag{33.2}$$

Нам нужно приблизить значение $\mathcal{L}u(x_i)$, используя только значения функции u в узлах сетки. Так как оператор $\mathcal{L}$ линеен, естественно использовать линейную комбинацию $u(x_j)$:

$$\boxed{\mathcal{L}u(x_i) \approx \sum_{j \in J} a_{ij} u(x_j), \quad i \in J_\omega.} \tag{33.3}$$

Мы получили общий вид того, что в литературе называется *разностной аппроксимацией дифференциального оператора*. Теперь обратимся к правой части уравнения (33.2). В принципе, её можно оставить без изменений, так как обычно значения функции f в точках сетки известны. Но можно также вместо точных значений $f_i = f(x_i)$ использовать приближенные:

$$\varphi_i \approx f_i.$$

Мы имеем на это право, так как и левую часть тождества мы представляем приближенно. В результате, подставляя вместо $u(x_i)$ приближенные значения y_i, для всех внутренних узлов мы получаем СЛАУ (неполную)

$$L_h y_h = \varphi_h, \tag{33.4a}$$

где L_h — матрица (линейный оператор), состоящая из N_ω строк и $N+1$ столбцов (N_ω — мощность множества внутренних узлов), φ_h — сеточная функция, определённая на внутренних узлах, то есть вектор размерности N_ω:

$$\varphi_h = (\varphi_i)_{i \in J_\omega}.$$

Совершенно аналогично для граничных узлов из тождества

$$\ell u(x_i) = \mu(x_i) \quad i \in J_\gamma,$$

получаем приближенные линейные уравнения

$$\ell_h y_h = \nu_h. \tag{33.4b}$$

Здесь ℓ_h — матрица (линейный оператор), состоящая из N_γ строк и $N+1$ столбцов (N_γ — мощность множества граничных узлов), ν_h — сеточная функция, приближающая μ в граничных узлах, то есть вектор размерности N_γ, $\nu_i \approx \mu(x_i)$. Собирая вместе уравнения (33.4), получим полную СЛАУ

$$\begin{cases} \sum\limits_{j=0}^{N} a_{ij} y_j = \varphi_i, & i \in J_\omega, \\ \sum\limits_{j=0}^{N} a_{ij} y_j = \nu_i, & i \in J_\gamma \end{cases} \quad \Leftrightarrow \quad \boxed{\begin{cases} L_h y_h = \varphi_h, \\ \ell_h y_h = \nu_h. \end{cases}} \tag{33.5}$$

Итак, мы имеем систему линейных алгебраических уравнений для нахождения y_h. Её можно записать совсем коротко:

$$\boxed{\tilde{L}_h y_h = \tilde{\varphi}_h,} \tag{33.5′}$$

где

$$\tilde{L}_h = A = (a_{ij})_{i,j=0}^{N}, \qquad \tilde{\varphi}_i = \begin{cases} \varphi_i, & i \in J_\omega, \\ \nu_i, & i \in J_\gamma. \end{cases}$$

Система (33.5), (33.5′) называется разностной схемой.

> *Разностной схемой* называется семейство систем линейных уравнений, зависящих от параметра h (или, что то же самое, от N), решение которых является сеточной функцией, аппроксимирующей решение исходной дифференциальной задачи.

33.3 Точность и сходимость разностных схем

В дальнейшем нам пригодится отображение (его называют *проектором*, хотя по смыслу оно не очень похоже на проекторы, с которыми мы работали до сих пор), которое переводит непрерывную функцию в сеточную:

$$\tilde{P}_h : H \to \tilde{H}_h,$$
$$\tilde{P}_h(u) = u_h = (u_0, u_1, \ldots, u_n)^T, \quad u_i = u(x_i).$$

Также введем проектор на внутренние узлы сетки:

$$P_h : H \to H_h.$$

33.3.1 Погрешность аппроксимации оператора

Пусть $\mathcal{L}$ — некоторый дифференциальный оператор, L_h — аппроксимирующий его разностный оператор, u — произвольная функция из H, $u_h = P_h(u)$. Погрешностью аппроксимации оператора $\mathcal{L}$ оператором L_h называется сеточная функция

$$\psi_h = L_h u_h - P_h(\mathcal{L}u). \tag{33.6}$$

Следует понимать, что ψ_h определена лишь на множестве внутренних узлов ω_h. Значение ψ_h в точке $x_i \in \omega_h$, которое обозначим ψ_i, называют *погрешностью аппроксимации в узле* x_i.

Если для достаточно гладких функций u имеет место оценка

$$\psi_i = \Big(L_h u_h - P_h(\mathcal{L}u)\Big)_i = O(|h|^p),$$

то говорят, что разностный оператор L_h аппроксимирует дифференциальный оператор $\mathcal{L}$ в точке x_i с порядком p.
Если же

$$\|\psi_h\| = O(|h|^p),$$

то говорят о p-ом порядке аппроксимации *на сетке*.

Замечание 33.1. Обозначение $|h|$ использовано здесь для того, чтобы определение было пригодно в многомерном случае, когда h представляет собой набор параметров сетки для каждой координаты, а также в случае неравномерной сетки. В первом случае $|h| = \|h\|$, во втором — $|h| = \max_i h_i$.

Замечание 33.2. Совершенно аналогично вводятся определения погрешности и порядка аппроксимации для граничных условий.

33.3.2 Погрешность аппроксимации и устойчивость схемы

Рассмотрим снова разностную схему (33.5)

$$\begin{cases} L_h y_h = \varphi_h, \\ \ell_h y_h = \nu_h. \end{cases} \qquad \Leftrightarrow \qquad \tilde{L}_h y_h = \tilde{\varphi}_h,$$

полученную из задачи (33.1)

$$\mathcal{L}u(x) = f(x), \quad x \in \Omega, \qquad \ell u(x) = \mu(x), \quad x \in \Gamma,$$

путем замены непрерывных функций сеточными и дифференциальных операторов разностными. Естественно поставить вопрос о близости решения

разностной задачи (сеточной функции $y_h \in \tilde{H}_h$) и точного решения дифференциальной задачи (функции $u \in H$).

Погрешностью разностной схемы $\tilde{L}_h y_h = \tilde{\varphi}_h$ называется сеточная функция

$$z_h = y_h - u_h \ \in \ \tilde{H}_h,$$

где $u_h = \tilde{P}_h(u)$ — точное решение исходной дифференциальной задачи в узлах сетки $\bar{\omega}_h$.
Если $\|z_h\| \xrightarrow[h\to 0]{} 0$, то говорят, что *разностная схема сходится*.
Если $\|z_h\| = O(|h|^p)$, то говорят, что *разностная схема имеет порядок точности* p.

Основной инструмент исследования погрешности разностных схем — невязка, получаемая подстановкой точного решения u_h в разностную схему. Такая невязка, как мы уже упоминали в предыдущих лекциях, называется погрешностью аппроксимации.

Погрешностью аппроксимации разностной схемы $\tilde{L}_h y_h = \tilde{\varphi}_h$ называется сеточная функция

$$\psi_h = \tilde{\varphi}_h - \tilde{L}_h u_h \ \in \ \tilde{H}_h,$$

где $u_h = \tilde{P}_h(u)$ — точное решение исходной дифференциальной задачи в узлах сетки $\bar{\omega}_h$.
Если $\|\psi_h\| \xrightarrow[h\to 0]{} 0$, то говорят, что *разностная схема аппроксимирует дифференциальную задачу*.
Если $\|\psi_h\| = O(|h|^p)$, то говорят, что *разностная схема имеет порядок аппроксимации* p.

Установим связь между z_h и ψ_h:

$$\begin{cases} 0 = \tilde{\varphi}_h - \tilde{L}_h y_h, \\ \psi_h = \tilde{\varphi}_h - \tilde{L}_h u_h \end{cases} \qquad \Rightarrow \quad \tilde{L}_h(y_h - u_h) = \psi_h,$$

или просто

$$\boxed{\tilde{L}_h z_h = \psi_h.} \tag{33.7}$$

Отсюда получаем, что если матрица $\tilde{L}_h$ невырождена, то

$$z_h = \tilde{L}_h^{-1} \psi_h.$$

Следовательно, если $\psi_h \to 0$ при $h \to 0$, то для сходимости схемы достаточно потребовать, чтобы при всех $h < h_0$ обратная матрица к $\tilde{L}_h$ была равномерно ограничена по норме. Это свойство называют устойчивостью разностной схемы.

Разностная схема $\tilde{L}_h y_h = \tilde{\varphi}_h$ называется *устойчивой*, если при всех $h < h_0$, где h_0 — некоторое фиксированное положительное число, справедлива оценка

$$\|\tilde{L}_h^{-1}\| \leqslant M < \infty,$$

где M не зависит от h.

Таким образом, сказанное выше можно сформулировать как теорему.

Теорема 33.1 (Лакс). *Если разностная схема а) имеет порядок аппроксимации $p > 0$ и б) устойчива, то она сходится с порядком p.*

Замечание 33.3. Приведенное определение устойчивости допускает также следующие две эквивалентные формулировки.

1. Разностная схема (33.5′) называется устойчивой, если для любых $h < h_0$ и векторов (сеточных функций) y_h, $\tilde{\varphi}_h$ таких, что $\tilde{L}_h y_h = \tilde{\varphi}_h$, справедливо $\|y_h\| \leqslant M\|\tilde{\varphi}_h\|$.
2. Разностная схема (33.5) называется устойчивой, если для любых $h < h_0$ и векторов (сеточных функций) y_h, φ_h, ν_h таких, что

   $$L_h y_h = \varphi_h, \quad \ell_h y_h = \nu_h,$$

   справедливо

   $$\|y_h\| \leqslant M(\|\varphi_h\| + \|\nu_h\|).$$

   Последняя формулировка является «классической».

33.4 Разностные аппроксимации основных дифференциальных операторов

Теперь рассмотрим один из главных вопросов теории разностных схем — вопрос построения разностных аппроксимаций для дифференциальных операторов. Как мы помним (см. (33.3)), разностные операторы имеют следующий общий вид (покомпонентный):

$$(L_h u_h)_i = \sum_{j \in J} a_{ij} u_j \approx \mathcal{L}u(x_i), \quad i \in J_\omega.$$

Здесь J — множество всех индексов $\{0, 1, \ldots, N\}$, a_{ij} — некоторые коэффициенты, которые определяются исходя из следующих соображений.

Во-первых, для практики важно, чтобы матрица оператора L_h была разреженной, поэтому желательно в линейной комбинации оставить лишь небольшое число слагаемых. Множество индексов, которые будут участвовать в суммировании, назовём *шаблоном* и обозначим $\text{Ш}(i)$:

$$(L_h u_h)_i = \sum_{j \in \text{Ш}(i)} a_{ij} u_j. \tag{33.8}$$

Во-вторых, за счет определения a_{ij} необходимо обеспечить должный порядок аппроксимации. Понятно, что чем меньше множество $\text{Ш}(i)$, тем

меньше свободы и тем меньший порядок аппроксимации можно получить. С другой стороны, использование слишком большого количества точек шаблона очень часто приводит к неустойчивости разностной схемы. С учетом всего этого приступим к построению разностных операторов.

33.4.1 Первые разностные производные

Итак, пусть

$$\mathcal{L}u(x) = u'(x).$$

Правая разностная производная. Производную в точке x, можно приблизить следующим очевидным способом:

$$u'(x) \approx \frac{u(x+h) - u(x)}{h} =: u_x. \tag{33.9}$$

Приближённое значение u_x называется *правой разностной производной*. Перепишем это соотношение в терминах сеточных функций. Рассмотрим равномерную сетку

$$x_i = x_0 + ih, \quad i = \overline{0,n}, \quad \omega_h = \{x_1, x_2, \ldots, x_{n-1}\}, \quad \gamma_h = \{x_0, x_n\}. \tag{33.10}$$

Пусть как обычно $u_h = P_h(u)$. Тогда формула (33.9) запишется как

$$u'(x_i) \approx (L_h u_h)_i = \frac{u_{i+1} - u_i}{h}.$$

Сопоставляя это с общей формой (33.8), видим, что

$$\text{Ш}(i) = \{i, i+1\}, \quad a_{ii} = -\frac{1}{h}, \quad a_{i,i+1} = \frac{1}{h}.$$

Таким образом, матрица оператора правой разностной производной на сетке (33.10) имеет ленточный вид

$$L_h = D_h^+ = \frac{1}{h}\begin{bmatrix} 0 & -1 & 1 & & \\ & 0 & -1 & 1 & \\ & & \ddots & \ddots & \ddots \\ & & & 0 & -1 \quad 1 \end{bmatrix}$$

(напомним, что $(L_h y_h)_i$ задают приближение к $\mathcal{L}u(x_i)$ только во внутренних узлах, то есть в нашем случае L_h имеет $n-1$ строк и $n+1$ столбцов).

Чему равен порядок аппроксимации оператора правой разностной производной? По определению для этого нужно найти погрешность аппроксимации в точке сетки $x_i = x$:

$$\psi_i = (L_h u_h)_i - u'(x) = \frac{u(x+h) - u(x)}{h} - u'(x) =$$
$$= \frac{u(x) + hu'(x) + \frac{h^2}{2}u''(x) + O(h^3) - u(x)}{h} - u'(x) = \frac{h}{2}u''(x) + O(h^2) = O(h).$$

Итак, *оператор правой разностной производной имеет первый порядок аппроксимации*:

$$\boxed{u_x = \frac{u(x+h) - u(x)}{h} = u'(x) + \frac{h}{2}u''(x) + O(h^2).} \qquad (33.11)$$

Левая разностная производная определена, как нетрудно догадаться, следующим образом:

$$u'(x) \approx \frac{u(x) - u(x-h)}{h} =: u_{\bar{x}}.$$

Действуя в полной аналогии со случаем правой производной, имеем $\text{Ш}(i) = \{i-1, i\}$

$$L_h = D_h^- = \frac{1}{h}\begin{bmatrix} -1 & 1 & 0 & & \\ & -1 & 1 & 0 & \\ & & \ddots & \ddots & \ddots \\ & & & -1 & 1 & 0 \end{bmatrix}.$$

Кроме этого, *оператор левой разностной производной имеет первый порядок аппроксимации*:

$$\boxed{u_{\bar{x}} = \frac{u(x) - u(x-h)}{h} = u'(x) - \frac{h}{2}u''(x) + O(h^2).} \qquad (33.12)$$

$\triangleright_1$ Докажите.

Центральная разностная производная. Для достижения более высокого порядка аппроксимации оператора первой производной необходимо расширить шаблон:

$$W(i) = \{i-1, i, i+1\}.$$

Можно догадаться, что если взять среднее арифметическое от (33.11) и (33.12), то члены $O(h)$ сократятся. Таким образом, центральная разностная производная имеет вид

$$u(x) \approx \frac{u(x+h) - u(x-h)}{2h} =: u_{\mathring{x}}.$$

По накатанной схеме для этого разностного оператора получаем

$$L_h = D_h^\circ = \frac{1}{2h}\begin{bmatrix} -1 & 0 & 1 & & \\ & -1 & 0 & 1 & \\ & & \ddots & \ddots & \ddots \\ & & & -1 & 0 & 1 \end{bmatrix},$$

$$\boxed{u_{\overset{\circ}{x}} = \frac{u(x+h)-u(x-h)}{2h} = u'(x) + \frac{h^2}{6}u(x) + O(h^4).} \qquad (33.13)$$

$\triangleright_2$ Докажите.

$\triangleright_3$ Запишите матрицы всех рассмотренных операторов разностного дифференцирования для случая неравномерной сетки.

33.4.2 Вторая разностная производная

Теперь рассмотрим аппроксимацию дифференциального оператора

$$\mathcal{L}u(x) = u''(x)$$

на равномерной сетке (33.10). Для его аппроксимации в точке $x = x_i$ нужен как минимум трехточечный шаблон:

$$\text{Ш}(i) = \{i-1, i, i+1\}.$$

Соответствующие коэффициенты a, b, c разностного оператора L_h,

$$(L_h u_h)_i = au_{i-1} + bu_i + cu_{i+1} \approx u''(x_i),$$

можно получить разными способами. Один способ был указан в упражнении 32.1. Здесь же мы воспользуемся другим общим методом — **методом неопределенных коэффициентов**. Рассмотрим разложение погрешности аппроксимации в точке $x = x_i$ по формуле Тейлора:

$$\begin{aligned}\psi_i = (L_h u_h)_i - u''(x) = au(x-h) + bu(x) + cu(x+h) - u''(x) = \\ = (a+b+c)u(x) + (c-a)hu'(x) + (c+a)\frac{h^2}{2!}u''(x) + (c-a)\frac{h^3}{3!}u'''(x) + \\ + (c+a)\frac{h^4}{4!}u^{(4)}(\xi) - u''(x).\end{aligned}$$

Для достижения максимально возможного порядка аппроксимации приравниваем к нулю коэффициенты при $u^{(j)}(x)$:

$$\begin{cases} a+b+c = 0, \\ c-a = 0, \\ (c+a)\dfrac{h^2}{2} - 1 = 0, \end{cases}$$

откуда

$$a = c = \frac{1}{h^2}, \quad b = -\frac{2}{h^2}.$$

Таким образом, мы получили искомую аппроксимацию

$$u''(x) \approx \frac{u(x-h) - 2u(x) + u(x+h)}{h^2} =: u_{\bar{x}x}.$$

Заметим, что здесь $u_{\bar{x}x}$ — не просто обозначение, а вполне конкретное указание на последовательное применение разностных операторов:

$$u_{\bar{x}x} = (u_{\bar{x}})_x = \Big(\frac{u(x) - u(x-h)}{h}\Big)_x =$$
$$= \frac{\frac{1}{h}(u(x+h) - u(x)) - \frac{1}{h}(u(x) - u(x-h))}{h} = \frac{u(x-h) - 2u(x) + u(x+h)}{h^2}.$$

С учетом всего вышеизложенного, имеем

$$\boxed{u_{\bar{x}x} = \frac{u(x-h) - 2u(x) + u(x+h)}{h^2} = u''(x) + \frac{h^2}{12}u^{(4)}(x) + O(h^4).} \quad (33.14)$$

$\triangleright_4$ Постройте матрицу оператора второй разностной производной.

Заключение лекции 33

Замечание 33.4. Следует понимать, что в случае неравномерной сетки операторы центральной разностной производной и второй разностной производной будут иметь только первый порядок аппроксимации.

34 Сеточный метод для нестационарного уравнения теплопроводности

Рассмотрим применение метода сеток к одномерному нестационарному уравнению теплопроводности

$$\mathcal{L}u(x,t) = \frac{\partial}{\partial t}u(x,t) - \frac{\partial^2}{\partial x^2}u(x,t) = f(x,t), \qquad \begin{array}{l} x \in [0,1], \\ t \geqslant 0. \end{array} \tag{34.1a}$$

Здесь $u(x,t)$ — неизвестная функция, определяющая температуру стержня, находящегося на отрезке $[0,1]$, в точке x и момент времени t. Для однозначного определения решения необходимо задать температуру стержня в начальный момент времени, то есть начальное условие вида

$$u(x,0) = u_0(x). \tag{34.1b}$$

Кроме этого нужны ещё граничные условия. Мы начнем с простейших условий первого рода:

$$u(0,t) = \mu_0(t), \quad u(1,t) = \mu_1(t). \tag{34.1c}$$

Таким образом, областью определения функции u является множество

$$\bar{\Omega} = \Omega \cup \Gamma,$$
$$\Omega = (0,1) \times (0,+\infty), \quad \Gamma = \{(x,0),(0,t),(1,t) \mid x \in [0,1], t \in [0,+\infty)\}.$$

Понятно, что на практике нас интересует решение на ограниченном временном интервале $t \in [0,T]$.

На области $\bar{\Omega}$ введём равномерную сетку узлов:

$$\bar{\omega}_{h\tau} = \bar{\omega}_h \times \bar{\omega}_\tau = \{(x_i,t_j)\}, \quad i = \overline{0,n}, \quad j = \overline{0,m},$$

$x_i = ih$, $h = 1/n$ — шаг по пространству, $t_j = j\tau$, $\tau = T/m$ — шаг по времени. Значения сеточной функции $y : \bar{\omega}_{h\tau} \to \mathbb{R}$ в узле (x_i,t_j) будем обозначать y_i^j.

Для построения разностной схемы нужно аппроксимировать дифференциальный оператор

$$\mathcal{L} = \frac{\partial}{\partial t} - \frac{\partial^2}{\partial x^2}$$

на сетке $\bar{\omega}_{h\tau}$. Сделать это можно несколькими способами.

34.1 Явная схема

Наиболее простая разностная схема получится, если использовать следующую аппроксимацию:

$$\mathcal{L}u(x,t) = \dot{u}(x,t) - u''(x,t) \approx u_t - u_{\bar{x}x} = \\ = \frac{1}{\tau}\Big(u(x,t+\tau) - u(x,t)\Big) - \frac{1}{h^2}\Big(u(x-h,t) - 2u(x,t) + u(x+h,t)\Big).$$

Подставляя сюда $x = x_i$, $t = t_j$ и заменяя $u(x_i, t_j)$ их приближенными значениями в узлах сетки,

$$y_i^j \approx u(x_i, t_j),$$

получаем следующий вид разностного оператора L:

$$(Ly)_i^j = \frac{1}{\tau}\Big(y_i^{j+1} - y_i^j\Big) - \frac{1}{h^2}\Big(y_{i-1}^j - 2y_i^j + y_{i+1}^j\Big). \tag{34.2}$$

Шаблон этого оператора имеет вид

$$\text{Ш}(i,j) = \left\{ \begin{matrix} & (i,j+1) & \\ (i-1,j) & (i,j) & (i+1,j) \end{matrix} \right\}.$$

Используя построенный оператор и учитывая граничные и начальные условия, получаем разностную схему

$$\frac{1}{\tau}\Big(y_i^{j+1} - y_i^j\Big) - \frac{1}{h^2}\Big(y_{i-1}^j - 2y_i^j + y_{i+1}^j\Big) = \varphi_i^j, \quad \begin{matrix} i = \overline{1, n-1}, \\ j = \overline{0, m-1}, \end{matrix} \tag{34.3a}$$

$$y_i^0 = u_0(x_i), \quad i = \overline{0, n}, \tag{34.3b}$$

$$y_0^j = \mu_0(t_j), \quad y_n^j = \mu_1(t_j), \quad j = \overline{0, m}. \tag{34.3c}$$

Здесь, как обычно, φ — сеточная функция, приближающая f. В простейшем случае $\varphi_i^j = f(x_i, t_j)$. Сеточное уравнение в (34.3a) можно переписать в так называемой безындексной форме:

$$y_t - y_{\bar{x}x} = \varphi, \quad \text{или} \quad \frac{\hat{y} - y}{\tau} - y_{\bar{x}x} = \varphi. \tag{34.3a'}$$

Здесь $y = y_i^j$ — значение сеточной функции в «текущей точке», $\hat{y} = y_i^{j+1}$ — значение на «верхнем слое». Под слоем подразумевается множество значений сеточной функции y при фиксированном значении индекса j.

Заметим, что структура уравнений (34.3a') позволяет легко находить решение «послойно» (так как все значения на нулевом слое известны):

$$\hat{y} = y + \tau(\varphi + y_{\bar{x}x}),$$

или

$$y_i^{j+1} = y_i^j + \tau\left(\varphi_i^j + \frac{1}{h^2}(y_{i-1}^j - 2y_i^j + y_{i+1}^j)\right), \qquad \begin{array}{l} i = \overline{1, n-1}, \\ j = 0, 1, 2, \ldots. \end{array}$$

Поэтому схема и называется явной (по аналогии с явными методами решения задачи Коши).

Найдём порядок аппроксимации построенной разностной схемы. Так как все краевые условия и правая часть уравнения приближаются точно, достаточно рассмотреть погрешность аппроксимации только во внутренних узлах (напомним, что разностный оператор $L \approx \mathcal{L}$ определяется формулой (34.2), а P — проектор на сетку $\omega_{h\tau}$):

$$\psi_i^j = \varphi_i^j - (LP(u))_i^j = \varphi_i^j - \frac{1}{\tau}\left(u_i^{j+1} - u_i^j\right) + \frac{1}{h^2}\left(u_{i-1}^j - 2u_i^j + u_{i+1}^j\right) =$$
$$= f(x_i, t_j) - \frac{1}{\tau}\Big(u(x_i, t_{j+1}) - u(x_i, t_j)\Big) + \frac{1}{h^2}\Big(u(x_{i-1}, t_j) - 2u(x_i, t_j) + u(x_{i+1}, t_j)\Big) =$$
$$= \Big[f(x,t) = \dot{u}(x,t) - u''(x,t), (33.11), (33.14)\Big] = O(h^2) + O(\tau) = O(h^2 + \tau).$$

Отсюда

$$\|\psi\| = \|P(\mathcal{L}u) - LP(u)\| = O(h^2) + O(\tau) = O(h^2 + \tau).$$

Напомним еще раз, что здесь u — точное решение задачи (34.1a). Таким образом, явная схема (34.3) имеет второй порядок аппроксимации по пространству и первый по времени.

34.2 Неявная схема

Воспользуемся теперь для аппроксимации производной по времени левой разностной производной:

$$\mathcal{L}u \approx u_{\bar{t}} - u_{\bar{x}x}.$$

В результате вместо (34.3a′) будем иметь

$$Ly = y_{\bar{t}} - y_{\bar{x}x} = \varphi, \quad \text{или} \quad \frac{y - \check{y}}{\tau} - y_{\bar{x}x} = \varphi.$$

$\triangleright_1$ Запишите вид шаблона для данного разностного оператора.

Здесь, понятное дело $\check{y} = y_i^{j-1}$. Для реализации удобнее сместить в этой формуле индексацию по слоям:

$$\frac{\hat{y} - y}{\tau} - \hat{y}_{\bar{x}x} = \hat{\varphi}. \tag{34.4′}$$

В индексной форме соответствующие разностные уравнения примут вид

$$\frac{1}{\tau}\left(y_i^{j+1} - y_i^j\right) - \frac{1}{h^2}\left(y_{i-1}^{j+1} - 2y_i^{j+1} + y_{i+1}^{j+1}\right) = \varphi_i^{j+1}, \quad \begin{matrix} i = \overline{1, n-1}, \\ j = \overline{0, m-1}. \end{matrix} \tag{34.4}$$

Значения в граничных точках, очевидно, определяются так же, как и в явной схеме (34.3).

Реализация решения данной схемы также имеет послойную структуру. Только для перехода на следующий слой необходимо каждый раз решать трехдиагональную СЛАУ вида

$$\begin{cases} -\frac{1}{h^2}\hat{y}_{i-1} + \left(\frac{1}{\tau} + \frac{2}{h^2}\right)\hat{y}_i - \frac{1}{h^2}\hat{y}_{i+1} = \hat{\varphi}_i + \frac{1}{\tau}y_i, \quad i = \overline{1, n-1}, \\ \hat{y}_0 = \mu_0(t_j), \quad \hat{y}_n = \mu_1(t_j). \end{cases} \tag{34.5}$$

Здесь $\hat{y}_i = y_i^{j+1}$, $y_i = y_i^j$. Отметим, что данная СЛАУ всегда разрешима в силу строгого диагонального преобладания. *Таким образом, решение СЛАУ* (34.4) *размерности* $(n-1) \times m$ *сводится к решению* m *СЛАУ размерности* $n-1$ *с трехдиагональной матрицей.*

Аналогично явной, неявная схема имеет имеет второй порядок аппроксимации по пространству и первый по времени, $\psi = O(h^2 + \tau)$.

34.3 Шеститочечная схема с весами

34.3.1 Построение

Теперь мы рассмотрим обобщение построенных выше явной и неявной разностных схем, полученное путем их «смешивания». Для начала введем еще одно обозначение для разностного оператора второй производной:

$$\Lambda y = y_{\bar{x}x}, \quad \Lambda y_i^j = \frac{1}{h^2}(y_{i-1}^j - 2y_i^j + y_{i+1}^j).$$

Тогда уравнения явной схемы (34.3) запишутся в виде

$$\frac{\hat{y}-y}{\tau} = \Lambda y + \varphi,$$

а неявной (34.4′) —

$$\frac{\hat{y}-y}{\tau} = \Lambda \hat{y} + \hat{\varphi}.$$

Взвешивая эти два выражения с весами $1-\sigma$ и σ, $\sigma \in [0,1]$, получим

$$\boxed{\frac{\hat{y}-y}{\tau} = \Lambda\big(\sigma\hat{y} + (1-\sigma)y\big) + \phi,} \tag{34.6}$$

где ϕ — некоторая сеточная функция, приближающая f, например

$$\phi_i = f(x_i, t_j + \tau/2).$$

Дополнительные краевые условия тут ничем не отличаются от рассмотренных выше схем. Очевидно, что при $\sigma = 0$ из (34.6) получаем явную схему (34.3), при остальных значениях σ схема будет неявной (при $\sigma = 1$ имеем «чисто неявную» схему (34.4′)). Для $\sigma \in (0,1)$ шаблон разностной схемы состоит, очевидно, из шести точек:

$$\text{Ш}(i,j) = \left\{ \begin{matrix} (i-1,j+1) & (i,j+1) & (i+1,j+1) \\ (i-1,j) & (i,j) & (i+1,j) \end{matrix} \right\}.$$

В индексной форме (34.6) записывается как

$$\frac{\hat{y}_i - y_i}{\tau} = \frac{1}{h^2}\Big(\sigma(\hat{y}_{i-1} - 2\hat{y}_i + \hat{y}_{i+1}) + (1-\sigma)(y_{i-1} - 2y_i + y_{i+1})\Big) + \phi_i.$$

Это — система линейных уравнений, которая позволяет по известным значениям $y_i = y_i^j$ на j-ом слое найти значения $\hat{y}_i = y_i^{j+1} \approx u(x_i, t_{j+1})$, $i = \overline{0,n}$, на следующем, $(j+1)$-ом, слое. Перепишем уравнения системы в виде

$$\begin{aligned} -\frac{\sigma\tau}{h^2}\hat{y}_{i-1} + \Big(1 + \frac{2\sigma\tau}{h^2}\Big)\hat{y}_i - \frac{\sigma\tau}{h^2}\hat{y}_{i+1} = \\ = \frac{\sigma'\tau}{h^2}y_{i-1} + \Big(1 - \frac{2\sigma'\tau}{h^2}\Big)y_i + \frac{\sigma'\tau}{h^2}y_{i+1} + \tau\phi_i, \quad i = \overline{1,n-1}, \end{aligned} \tag{34.7a}$$

(здесь $\sigma' = 1-\sigma$), причем граничные условия (34.1c) дают

$$y_0 = \mu_0(t_j), \quad y_n = \mu_1(t_j), \quad \hat{y}_0 = \mu_0(t_{j+1}), \quad \hat{y}_n = \mu_1(t_{j+1}). \tag{34.7b}$$

Запишем эту СЛАУ в векторном виде. Введем обозначения

$$v = (y_1, y_2, \ldots, y_{n-1})^T, \quad \hat{v} = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{n-1})^T,$$

и для (34.7) получим представление

$$\boxed{A\hat{v} = Bv + g,} \tag{34.8}$$

$$A = \begin{bmatrix} 1+\frac{2\sigma\tau}{h^2} & -\frac{\sigma\tau}{h^2} & & & \\ -\frac{\sigma\tau}{h^2} & 1+\frac{2\sigma\tau}{h^2} & -\frac{\sigma\tau}{h^2} & & \\ & \ddots & \ddots & \ddots & \\ & & -\frac{\sigma\tau}{h^2} & 1+\frac{2\sigma\tau}{h^2} & -\frac{\sigma\tau}{h^2} \\ & & & -\frac{\sigma\tau}{h^2} & 1+\frac{2\sigma\tau}{h^2} \end{bmatrix}, \tag{34.9a}$$

$$B = \begin{bmatrix} 1-\frac{2\sigma'\tau}{h^2} & \frac{\sigma'\tau}{h^2} & & & \\ \frac{\sigma'\tau}{h^2} & 1-\frac{2\sigma'\tau}{h^2} & \frac{\sigma'\tau}{h^2} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{\sigma'\tau}{h^2} & 1-\frac{2\sigma'\tau}{h^2} & \frac{\sigma'\tau}{h^2} \\ & & & \frac{\sigma'\tau}{h^2} & 1-\frac{2\sigma'\tau}{h^2} \end{bmatrix}, \tag{34.9b}$$

$$g = \begin{bmatrix} \tau\tilde{\phi}_1 \\ \tau\phi_2 \\ \vdots \\ \tau\phi_{n-2} \\ \tau\tilde{\phi}_{n-1} \end{bmatrix}, \quad \begin{aligned} \tilde{\phi}_1 &= \phi_1 + \tfrac{1}{h^2}(\sigma\mu_0(t_{j+1}) + \sigma'\mu_0(t_j)), \\ \tilde{\phi}_{n-1} &= \phi_{n-1} + \tfrac{1}{h^2}(\sigma\mu_1(t_{j+1}) + \sigma'\mu_1(t_j)). \end{aligned} \tag{34.9c}$$

Таким образом, система (34.8) всегда разрешима и метод прогонки для нее всегда выполним (в силу диагонального преобладания матрицы A).

Далее мы займемся исследованием свойств полученной схемы в зависимости от значения параметра σ.

34.3.2 Порядок аппроксимации

Согласно определению, для исследования порядка аппроксимации разностной схемы (34.6), мы должны вычислить погрешность аппроксимации ψ, которая в нашем случае равна

$$\psi = \phi + \Lambda\big(\sigma\hat{u} + (1-\sigma)u\big) - u_t.$$

Следует понимать, что здесь для краткости опущены индексы:

$$\psi = \psi_i^j, \quad \phi = \phi_i^j, \quad u = u_i^j = u(x_i, t_j), \quad \hat{u} = u_i^{j+1} = u(x_i, t_{j+1}), \quad u_t = (\hat{u}-u)/\tau,$$

а $u(x,t)$ — точное решение исходной задачи (34.1a). Воспользуемся тождествами

$$\begin{aligned}\hat{u} = 0.5(\hat{u}+u) + 0.5(\hat{u}-u) &= 0.5(\hat{u}+u) + 0.5\tau u_t,\\ u &= 0.5(\hat{u}+u) - 0.5\tau u_t,\\ \sigma\hat{u} + (1-\sigma)u &= 0.5(\hat{u}+u) + (\sigma-0.5)\tau u_t,\end{aligned}$$

откуда

$$\psi = \phi + 0.5\Lambda(\hat{u}+u) + (\sigma-0.5)\tau\Lambda u_t - u_t, \tag{34.10}$$

Вычислим разложение ψ в ряд Тейлора в точке $(x_i, t_j+\tau/2)$ — центре шаблона. Чтобы избежать громоздких выкладок, введем обозначения

$$\bar{u} = u(x_i,\bar{t}), \quad \bar{t} = t_j + \tau/2, \quad \dot{u} = \frac{\partial}{\partial t}u(x_i,t_j), \quad u'' = \frac{\partial^2}{\partial x^2}u(x_i,t_j).$$

Тогда формулы (33.11)–(33.14) дают следующие разложения в указанной точке:

$$\begin{gathered}\Lambda u = u'' + \frac{h^2}{12}u^{(4)} + O(h^4),\\ \hat{u} = \bar{u} + \frac{\tau}{2}\bar{\dot{u}} + \frac{\tau^2}{8}\bar{\ddot{u}} + O(\tau^3),\\ u = \bar{u} - \frac{\tau}{2}\bar{\dot{u}} + \frac{\tau^2}{8}\bar{\ddot{u}} + O(\tau^3),\\ 0.5(\hat{u}+u) = \bar{u} + \frac{\tau^2}{8}\bar{\ddot{u}} + O(\tau^3),\\ u_t = \bar{\dot{u}} + O(\tau^2).\end{gathered}$$

Подставляя все это в (34.10), получаем

$$\psi = (\bar{u}'' - \bar{\dot{u}} + \phi) + (\sigma-0.5)\tau\bar{\dot{u}}'' + \frac{h^2}{12}\bar{u}^{(4)} + O(\tau^2+h^4). \tag{34.11}$$

Напомним, что u — проекция точного решения на сетку, то есть $\dot{u} - u'' = f$. Следовательно, если взять

$$\phi = \bar{f} = f(x_i, t+\tau/2) \quad \text{и} \quad \sigma = 0.5,$$

то разностная схема будет иметь второй порядок аппроксимации по обеим переменным, то есть

$$\psi = O(h^2+\tau^2).$$

Такая схема называется *симметричной шеститочечной схемой*, или *схемой Кранка-Николсона*.

Но самое интересное только начинается! Можно еще «убить» член разложения с h^2 в (34.11):

$$\bar{u}^{(4)} = (\bar{u}'')'' = (\bar{\dot{u}} - \bar{f})'' = \bar{\dot{u}}'' - \bar{f}''.$$

Подставляя это выражение в (34.11), получаем

$$\psi = (\phi - \bar{f}) + \left[(\sigma - 0.5)\tau + \frac{h^2}{12}\right]\bar{\dot{u}}'' - \frac{h^2}{12}\bar{f}'' + O(\tau^2 + h^4). \qquad (34.12)$$

Приравнивая выражение в квадратных скобках к нулю, получаем

$$\sigma = \frac{1}{2} - \frac{h^2}{12\tau} = \sigma_*.$$

Следовательно, если $\sigma = \sigma_*$ и

$$\phi = \bar{f} + \frac{h^2}{12}\bar{f}'',$$

то будем иметь второй порядок по времени и четвертый по пространству: $\psi = O(h^4 + \tau^2)$. Кроме того, порядок аппроксимации не нарушится, если приблизить $\bar{f}''$ второй разностной производной:

$$\phi = \bar{f} + \frac{h^2}{12}\Lambda\bar{f},$$

или

$$\phi_i^j = \frac{5}{6}f(x_i, t_j + 0.5\tau) + \frac{1}{12}\Big(f(x_{i-1}, t_j + 0.5\tau) + f(x_{i+1}, t_j + 0.5\tau)\Big).$$

Такую схему называют *схемой повышенного порядка точности*.

$\triangleright_2$ Пользуясь формулами (34.11), (34.12) выпишите порядок аппроксимации шеститочечной разностной схемы для всех возможных случаев.

34.3.3 Устойчивость

Устойчивость мы будем исследовать для случая однородной задачи и нулевых граничных условий:

$$f(x, t) = 0, \quad \mu_0(t) = \mu_1(t) = 0.$$

В данном случае разностная схема будет устойчива (по начальному условию), если для приближенного решения $\{y_i^j\}$ справедлива оценка

$$\max_{i,j} |y_i^j| \leqslant M\|u_0\|, \quad \forall h < h_0,\ \tau < \tau_0.$$

Здесь мы учли, что начальное условие (34.1b) в нашей схеме выполняется точно.

Вспомним, что на каждых двух соседних слоях решение разностной схемы удовлетворяет СЛАУ (34.8), которую в нашем случае ($g = 0$) можно записать как

$$v^{(j+1)} = A^{-1}Bv^{(j)} = Cv^{(j)},$$

где

$$v^{(j)} = (y_1^j, y_2^j, \ldots, y_{n-1}^j)^T.$$

Таким образом, для решения на последнем слое имеем соотношение

$$v^{(m)} = Cv^{(m-1)} = \ldots = C^m v^{(0)}.$$

Здесь, понятное дело,

$$v^{(0)} = (u_0(x_1), u_0(x_2), \ldots, u_0(x_{n-1}))^T.$$

Видно, что критерием ограниченности $v^{(m)}$ при $\tau \to 0$ (что в свою очередь является критерием условием устойчивости разностной схемы) является равномерная ограниченность по норме матрицы C^m при $m \to \infty$. Из курса вычислительных методов алгебры мы, конечно, помним, что для этого необходимо и достаточно потребовать

$$\rho(C) \leqslant 1,$$

где $\rho(C)$ — абсолютная величина максимального по модулю собственного значения матрицы C.

Таким образом, *для устойчивости шеститочечной схемы по начальным данным необходимо и достаточно, чтобы все собственные значения матрицы $C = A^{-1}B$ по модулю не превышали единицу при всех $h < h_0$ и $\tau < \tau_0$.*

Проверкой этого условия мы сейчас и займемся. Для этого нам понадобятся следующие леммы.

Лемма 34.1. *Собственные значения λ_k и собственные векторы ξ^k матрицы*

$$D = \frac{1}{h^2}\begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{bmatrix}$$

размерности $(n-1)\times(n-1)$, *имеют вид*

$$\lambda_k = -\frac{4}{h^2}\sin^2\frac{\pi k}{2n},$$
$$\xi^k = \left(\sin\frac{\pi k}{n}, \sin\frac{2\pi k}{n}, \ldots, \sin\frac{(n-1)\pi k}{n}\right)^T, \quad k = \overline{1, n-1}. \tag{34.13}$$

Доказательство. Начнем со следующего наблюдения. Задача нахождения собственных значений и векторов матрицы D,

$$Dv = \lambda v \tag{34.14}$$

прямым образом связана с непрерывной задачей Штурма–Лиувилля для уравнения теплопроводности:

$$u''(x) - \lambda u(x) = 0, \quad x \in [0,1], \quad u(0) = u(1) = 0, \tag{34.15}$$

решения которой легко найти:

$$u_k(x) = \sin\pi k x, \quad \lambda_k = -\pi^2 k^2, \quad k = 0, 1, 2, \ldots. \tag{34.16}$$

Теперь разобьем отрезок $[0,1]$ на n равных частей и на полученной сетке $\bar{\omega}_h = \{x_i = ih\}_{i=0}^n$, $h = 1/n$, аппроксимируем задачу (34.15). Получим СЛАУ

$$\frac{1}{h^2}(y_{i-1} - 2y_i + y_{i+1}) - \lambda y_i = 0, \quad i = \overline{0, n}, \quad y_0 = y_n = 0, \tag{34.17}$$

которая в точности совпадает с (34.14), где $v = (y_1, \ldots, y_{n-1})^T$. Так как вектор v представляет собой сеточный аналог функции u из (34.15), можно предположить, что собственные векторы ξ^k дискретной задачи (34.14) имеют вид, аналогичный (34.16):

$$\xi^k = (\sin\pi k x_1, \sin\pi k x_2, \ldots, \sin\pi k x_{n-1})^T.$$

Прямая подстановка $v = \xi^k$, то есть $y_i = \sin\pi k x_i$, в (34.17) дает

$$\frac{1}{h^2}\Big(\sin\pi k(x_i - h) - 2\sin\pi k x_i + \sin\pi k(x_i + h)\Big) - \lambda\sin\pi k x_i = 0.$$

Так как

$$\sin\pi k(x_i - h) + \sin\pi k(x_i + h) = 2\cos\pi k h\,\sin\pi k x_i,$$

отсюда получаем

$$\frac{2}{h^2}\Big(\cos\pi k h - 1\Big)\sin\pi k x_i - \lambda\sin\pi k x_i = 0,$$

откуда

$$\lambda = \lambda_k = \frac{2}{h^2}\Big(\cos \pi k h - 1\Big) = -\frac{4}{h^2}\sin^2\frac{\pi k h}{2} = -\frac{4}{h^2}\sin^2\frac{\pi k}{2n}.$$

■

Лемма 34.2. *Пусть матрицы A и B имеют одинаковые собственные векторы $\{\xi^k\}$, но различные собственные значения — $\{\lambda_k\}$ и $\{\mu_k\}$ соответственно, $\lambda_k \neq 0$. Тогда матрица $C = A^{-1}B$ имеет собственные векторы ξ^k и собственные значения*

$$\nu_k = \frac{\mu_k}{\lambda_k}.$$

Доказательство. Везде далее для краткости будем опускать индекс k. По условию имеем

$$Av = \lambda v, \quad Bv = \mu v.$$

Отсюда $A^{-1}v = \lambda^{-1}v$, и $Cv = A^{-1}Bv = \mu A^{-1}v = \mu\lambda^{-1}v$. ■

Вернемся к нашей задаче: нам надо найти собственные значения матрицы $C = A^{-1}B$, где A и B — трехдиагональные симметричные матрицы, имеющие вид (34.9a), (34.9b). Нетрудно заметить, что их можно представить в виде

$$A = I + \sigma\tau D, \quad B = I - \sigma'\tau D.$$

Эти матрицы имеют собственные векторы $\{\xi^k\}$ (34.13), а также собственные значения

$$1 + \sigma\tau\lambda_k \quad \text{и} \quad 1 - \sigma'\tau\lambda_k$$

соответственно, где $\lambda_k = -\frac{4}{h^2}\sin^2\frac{\pi k}{2n}$.

$\triangleright_3$ Докажите это.

Следовательно, по лемме 34.2, собственные значения матрицы C равны

$$\nu_k = \frac{1 - \sigma'\tau\lambda_k}{1 + \sigma\tau\lambda_k}.$$

Условие устойчивости $|\nu_k| \leqslant 1$ с учетом $\sigma' = 1 - \sigma$ приводит к неравенствам

$$-1 - \sigma\tau\lambda_k \leqslant 1 - (1-\sigma)\tau\lambda_k \leqslant 1 + \sigma\tau\lambda_k,$$

откуда

$$\sigma \geqslant \frac{1}{2} + \frac{1}{\tau\lambda_k} \geqslant \left[-\frac{4}{h^2} < \lambda_k < 0\right] \geqslant \frac{1}{2} - \frac{h^2}{4\tau}.$$

Итак, условие устойчивости шеститочечной разностной схемы имеет вид

$$\boxed{\sigma \geqslant \sigma_0 = \frac{1}{2} - \frac{h^2}{4\tau}.} \tag{34.18}$$

Отсюда следуют следующие свойства.

1. Явная схема ($\sigma = 0$) устойчива лишь при

$$\tau \leqslant \frac{h^2}{2}.$$

   Это очень серьезное препятствие на пути применения этой схемы для маленьких h.

2. При $\sigma \geqslant \frac{1}{2}$ схема устойчива при любых h и τ (безусловно устойчива).

$\triangleright_4$ Докажите, что шеститочечная схема повышенного порядка безусловно устойчива.

35 Разностные схемы для волнового уравнения

35.1 Постановка задачи

Волновое уравнение (уравнение колебаний струны):

$$\ddot{u}(x,t) = u''(x,t) + f(x,t), \quad x \in [0,1], \quad 0 \leqslant t \leqslant T, \tag{35.1a}$$
$$u(x,0) = u_0(x), \quad \dot{u}(x,0) = v_0(x), \tag{35.1b}$$
$$u(0,t) = \mu_0(t), \quad u(1,t) = \mu_1(t). \tag{35.1c}$$

Область определения u — прежняя:

$$\bar{\Omega} = \Omega \cup \Gamma,$$
$$\Omega = (0,1) \times (0,+\infty), \quad \Gamma = \{(x,0),(0,t),(1,t) \mid x \in [0,1], t \in [0,+\infty)\}.$$

35.2 Девятиточечная параметрическая схема

Здесь все будет проходить аналогично случаю шеститочечной схемы для уравнения теплопроводности из прошлой лекции с небольшими нюансами.

Сетка:

$$\bar{\omega}_{h\tau} = \bar{\omega}_h \times \bar{\omega}_\tau = \{(x_i,t_j)\} = \{(ih,j\tau)\}, \quad i = \overline{0,n}, \quad j = \overline{0,m}. \tag{35.2}$$

Стандартные обозначения:

$$u(x_i,t_j) \approx y_i^j = y, \quad \hat{y} = y_i^{j+1}, \quad \check{y} = y_i^{j-1}. \tag{35.3}$$

Апроксимация операторов:

$$\ddot{u}(x_i,t_j) \approx y_{\bar{t}t} = \frac{1}{\tau^2}(\hat{y} - 2y + \check{y}),$$
$$u''(x_i,t_j) \approx y_{\bar{x}x} = \Lambda y = \frac{1}{h^2}(y_{i-1}^j - 2y_i^j + y_{i+1}^j),$$
$$f(x_i,t_j) =: \varphi_i^j.$$

Семейство схем с весами:

$$y_{\bar{t}t} = \Lambda(\sigma\hat{y} + (1-2\sigma)y + \sigma\check{y}) + \varphi, \tag{35.4a}$$
$$y_i^0 = u_0(x_i), \quad \frac{1}{\tau}(y_i^1 - y_i^0) = \tilde{v}_0(x_i), \tag{35.4b}$$
$$y_0^j = \mu_0(t_j), \quad y_n^j = \mu_1(t_j). \tag{35.4c}$$

Шаблон — 9 точек. Схема — *трехслойная*.

Основная проблема — вычисление решения с достаточной точностью на первом слое. Другими словами, нужно выбрать $\tilde{v}_0$ в (35.4b) так, чтобы

второе начальное условие $\dot{u}(x,0) = v_0(x)$ аппроксимировалось со вторым порядком по t. Для этого используем (33.11) и (35.1a).

$$u_t(x,0) = \frac{1}{\tau}(u(x,\tau) - u(x,0)) = \dot{u}(x,0) + \frac{1}{2}\tau\ddot{u}(x,0) + O(\tau^2) =$$
$$= \left[\ddot{u} = u'' + f\right] = v_0(x) + \frac{1}{2}(u_0''(x) + f(x,0)) + O(\tau^2), \quad (35.5)$$

следовательно,

$$\tilde{v}_0(x) = v_0(x) + \frac{1}{2}\Big(u_0''(x) + f(x,0)\Big). \quad (35.6)$$

Распишем уравнения (35.4a):

$$\hat{y}_i - \frac{\tau^2}{h^2}\sigma(\hat{y}_{i-1} - 2\hat{y}_i + \hat{y}_{i+1}) =$$
$$= 2y_i + \frac{\tau^2}{h^2}\sigma'(y_{i-1} - 2y_i + y_{i+1}) - \Big(\check{y}_i - \frac{\tau^2}{h^2}\sigma(\check{y}_{i-1} - 2\check{y}_i + \check{y}_{i+1})\Big) + \tau^2\varphi_i,$$

$$\sigma' = 1 - 2\sigma.$$

С учетом граничных условий эти уравнения записываются в следующем виде:

$$\boxed{A\hat{v} + Bv + A\check{v} = \phi,} \quad (35.7)$$

$$A = I - \tau^2\sigma D, \quad B = -2(I + \tfrac{1}{2}\tau^2\sigma' D), \quad (35.8)$$

$$D = \frac{1}{h^2}\begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{bmatrix},$$

$$v = (y_1^j, y_2^j, \ldots, y_{n-1}^j)^T.$$

$\triangleright_1$ Запишите вид вектора ϕ.

Реализация схемы заключается в послойном вычислении $\hat{v}$ из системы (35.7). Для применимости метода прогонки достаточно выполнения условия $\sigma \geqslant 0$.

35.2.1 Порядок аппроксимации

Погрешность аппроксимации во внутренних узлах:

$$\psi = \varphi + \Lambda(\sigma\hat{u} + (1-2\sigma)u + \sigma\check{u} - u_{\bar{t}t},$$

где $u = u(x_i, t_j)$ — точное решение: $\ddot{u} = u'' + f$. Избавляемся от крышек:

$$\begin{aligned} \hat{u} &= u + \tau u_t, \\ \check{u} &= u - \tau u_{\bar{t}} \end{aligned} \quad \Rightarrow \quad \sigma\hat{u} + (1-2\sigma)u + \sigma\check{u} = u + \sigma\tau^2 u_{\bar{t}t},$$

откуда

$$\psi = \varphi + \Lambda u + \sigma\tau^2 \Lambda u_{\bar{t}t} - u_{\bar{t}t} = f + u'' + \sigma\tau^2 \ddot{u}'' - \ddot{u} + O(\tau^2 + h^2),$$

то есть

$$\psi = O(\tau^2 + h^2).$$

Таким образом, при любых σ схема имеет второй порядок аппроксимации по обеим переменным (во внутренних узлах).

С учетом того, что начальные условия аппроксимируются с погрешностью $O(\tau^2)$, окончательно получаем, что порядок аппроксимации разностной схемы равен двум по обеим переменным.

Замечание 35.1. За счет выбора σ и φ можно повысить порядок аппроксимации по x до четвертого.

35.2.2 Устойчивость

Исследуем устойчивость по начальным данным, то есть полагаем

$$\mu_0(t) = \mu_1(t) = 0, \quad f(x,t) = 0.$$

Из (35.7) имеем

$$v^{(j+1)} = -A^{-1}Bv^{(j)} - v^{(j-1)}, \quad j = 1, 2, 3, \ldots \tag{35.9}$$

Критерий устойчивости:

$$\|v^{(j)}\| \leqslant C \quad \text{при всех} \quad j \geqslant 1, \quad h < h_0, \quad \tau < \tau_0.$$

Рассмотрим большие векторы размерности $2(n-1)$:

$$w^{(j)} = \begin{bmatrix} v^{(j)} \\ v^{(j-1)} \end{bmatrix}.$$

Тогда (35.9) можно записать как

$$w^{(j+1)} = Mw^{(j)} = M^2 w^{(j-1)} = \ldots = M^j w^{(1)},$$

где

$$M = \begin{bmatrix} -A^{-1}B & -I \\ I & 0 \end{bmatrix}, \tag{35.10}$$

I — единичная матрица.

Таким образом, для устойчивости схемы необходимо, чтобы все собственные значения матрицы M не превосходили единицы по модулю. Найдем их.

Пусть μ и ζ — собственное значение и вектор матрицы $A^{-1}B$ соответственно:

$$A^{-1}B\zeta = \mu\zeta.$$

Покажем, что собственные векторы матрицы M имеют вид

$$\eta = \begin{bmatrix} \nu\zeta \\ \zeta, \end{bmatrix}.$$

где ν — соответствующее собственное значение M. Действительно,

$$M\eta = \begin{bmatrix} -A^{-1}B & -I \\ I & 0 \end{bmatrix} \begin{bmatrix} \nu\zeta \\ \zeta \end{bmatrix} = \begin{bmatrix} -\nu A^{-1}B\zeta - \zeta \\ \nu\zeta \end{bmatrix} = \begin{bmatrix} -(\mu\nu+1)\zeta \\ \nu\zeta \end{bmatrix} = \nu\eta.$$

Последнее равенство будет достигаться, если мы потребуем выполнения тождества $\nu^2 = -(\mu\nu + 1)$, или

$$\nu^2 + \mu\nu + 1 = 0. \tag{35.11}$$

Итак, все собственные значения матрицы M удовлетворяют уравнению (35.11), где μ — собственные значения матрицы $A^{-1}B$.

По определению (35.8) имеем

$$A = I - \tau^2\sigma D, \quad B = -2(I + \tfrac{1}{2}\tau^2\sigma' D),$$

откуда согласно леммам 34.1, 34.2 получаем

$$\mu = -2\frac{1 + \frac{1}{2}\tau^2\sigma'\lambda}{1 - \tau^2\sigma\lambda} = -2\frac{1 + \frac{1}{2}\tau^2(1-2\sigma)\lambda}{1 - \tau^2\sigma\lambda} = -2(1+\alpha),$$

где

$$\alpha = \frac{\frac{1}{2}\tau^2\lambda}{1 - \tau^2\sigma\lambda}, \tag{35.12}$$

а

$$\lambda = \lambda_k = -\frac{4}{h^2}\sin^2\frac{\pi k}{2n}.$$

Таким образом, (35.11) принимает вид

$$\nu^2 - 2(1+\alpha)\nu + 1 = 0,$$

откуда

$$\nu = 1 + \alpha \pm \sqrt{\alpha(\alpha+2)}.$$

Несложный анализ показывает, что при $\alpha \notin [-2, 0]$ имеем $|\nu| > 1$, а при

$$\alpha \in [-2, 0] \quad \Rightarrow \quad \nu \in \mathbb{C}, \quad |\nu| = 1.$$

Таким образом критерий устойчивости по начальным данным имеет вид

$$-2 \leqslant \alpha \leqslant 0$$

или

$$-2 \leqslant \frac{\frac{1}{2}\tau^2\lambda}{1 - \tau^2\sigma\lambda} \leqslant 0.$$

Отсюда с учетом $\lambda \in (-\frac{4}{h^2}, 0)$ получаем следующее ограничение на величину σ:

$$\boxed{\sigma > \frac{1}{4} - \frac{h^2}{4\tau^2}.} \tag{35.13}$$

Ура, товарищи!

$\triangleright_2$ Запишите условие устойчивости соответствующей явной схемы.

36 Численное решение задачи Дирихле для уравнения Пуассона

36.1 Постановка задачи

Рассмотрим двумерную задачу Дирихле для уравнения Пуассона, которая описывает стационарное распределение тепла в области Ω с границей Γ:

$$\frac{\partial^2}{\partial x_1^2}u(x_1,x_2)+\frac{\partial^2}{\partial x_2^2}u(x_1,x_2)=f(x_1,x_2),\quad (x_1,x_2)\in\Omega\subset\mathbb{R}^2, \tag{36.1a}$$

$$u(x_1,x_2)|_\Gamma=\mu(x_1,x_2), \tag{36.1b}$$

или кратко

$$\Delta u=f,\quad x\in\Omega,\qquad u|_\Gamma=\mu. \tag{36.2}$$

Как обычно, здесь Γ — граница области Ω.

36.2 Разностные схемы

36.2.1 Прямоугольная область

Так как отличия прямоугольной области от квадратной не существенны, рассмотрим задачу Дирихле в единичном квадрате:

$$\Omega=(0,1)\times(0,1).$$

Введем на области $\bar\Omega=\Omega\cup\Gamma$ равномерную сетку с шагом $h=1/n$:

$$\bar\omega=\Big\{(ih,jh)\Big\}_{i,j=0}^{n}=\omega\cup\gamma,$$

$\omega=\bar\omega\cap\Omega$, $\gamma=\bar\omega\cap\Gamma$.

Для аппроксимации оператора Лапласа $\Delta=\frac{\partial^2}{\partial x_1^2}+\frac{\partial^2}{\partial x_1^2}$ естественно использовать вторые разностные производные по каждой переменной:

$$\Delta u(x_1,x_2)\approx u_{\bar x_1 x_1}+u_{\bar x_2 x_2}=\Lambda_1 u+\Lambda_2 u=$$
$$=\frac{1}{h^2}(u_{i-1,j}-2u_{i,j}+u_{i+1,j})+\frac{1}{h^2}(u_{i,j-1}-2u_{i,j}+u_{i,j+1}). \tag{36.3}$$

Здесь $x_1 = ih$, $x_2 = jh$, $u_{i,j} = u(ih, jh)$, шаблон пятиточечный, его называют «крест».

Заменяя $u_{i,j}$ их приближенными значениями $y_{i,j}$ и учитывая граничные условия, получаем общий вид СЛАУ (разностной схемы):

$$\frac{1}{h^2}(y_{i-1,j} + y_{i+1,j} + y_{i,j-1} + y_{i,j+1} - 4y_{i,j}) = \varphi_{i,j}, \quad i,j = \overline{1, n-1}, \tag{36.4a}$$

$$y_{i,j} = \mu_{i,j}, \quad (i,j) \in J_\gamma. \tag{36.4b}$$

Здесь $\varphi_{i,j} = f(ih, jh)$, $\mu_{i,j} = \mu(ih, jh)$, J_γ — множество индексов граничных узлов. Заметим, что четыре угловые точки $(0,0)$, $(1,0)$, $(0,1)$, $(1,1)$ нет смысла включать в сетку γ, так как значения в этих точках в разностной схеме нигде не используются.

Порядок аппроксимации данной разностной схемы, очевидно, равен двум по обеим переменным.

Для записи схемы в матричном виде, упорядочим компоненты сеточной функции y естественным образом:

$$y = (y_{1,1}, \ldots, y_{1,n-1}, y_{2,1}, \ldots, y_{2,n-1}, \ldots\ldots, y_{n-1,1}, \ldots, y_{n-1,n-1})^T.$$

Тогда СЛАУ (36.4) запишется как

$$Ay = \phi, \tag{36.5}$$

где матрица A размерности $(n-1)^2$ имеет трехдиагональный блочный вид

$$A = \frac{1}{h^2}\begin{bmatrix} B & I & & & \\ I & B & I & & \\ & \ddots & \ddots & \ddots & \\ & & I & B & I \\ & & & I & B \end{bmatrix}, \quad B = \begin{bmatrix} -4 & 1 & & & \\ 1 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -4 & 1 \\ & & & 1 & -4 \end{bmatrix},$$

I — единичная матрица. Размерность каждого блока равна $n-1$. Матрица $-A$ является положительно определенной и симметричной.

$\triangleright_1$ Запишите вид вектора ϕ.

$\triangleright_2$ Запишите вид матрицы A для случая прямоугольной сетки с шагами h_1 и h_2.

36.2.2 О реализации решения СЛАУ

Как правило для решения системы (36.5) матрица A в явном виде не строится. Это связано с тем, что данная система в силу ее большой размерности и разреженности решается итерационными методами (см. далее), для

реализации которых нужно лишь уметь вычислять произведение матрицы A на вектор.

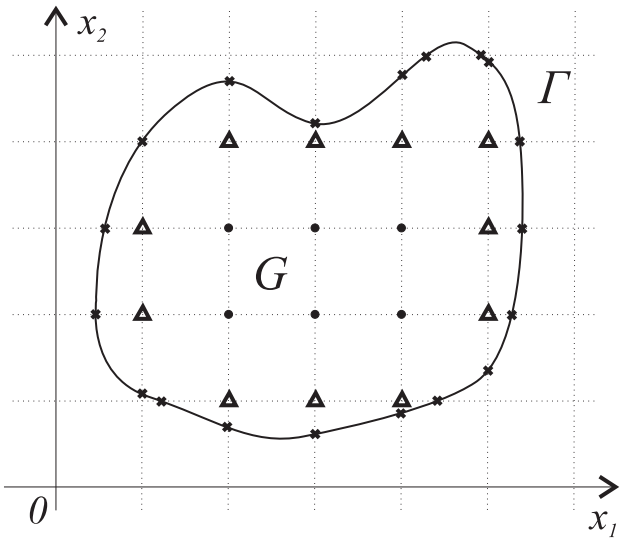
В частности, итерация метода Зейделя описывается следующим простым алгоритмом:

$$y_{i,j} \leftarrow \frac{1}{4}\Big(y_{i-1,j} + y_{i+1,j} + y_{i,j-1} + y_{i,j+1} - h^2\varphi_{i,j}\Big), \quad i,j = \overline{1,n-1}. \qquad (36.6)$$

Перед началом итераций, естественно, нужно заполнить значения $y_{i,j}$ в граничных узлах точными значениями $\mu_{i,j}$, а во внутренних — произвольными начальными приближениями, как правило нулевыми.

Можно показать, что метод Гаусса-Зейделя, равно как и другие простейшие методы, рассмотренные нами в курсе ВМА, будут сходиться на данной задаче. Однако с ростом числа узлов в сетке скорость их сходимости будет ощутимо снижаться. ———

36.2.3 Область сложной формы



В случае, когда область Ω имеет сложную форму, сеточный метод можно реализовать двумя способами: 1) используя неравномерную сетку в приграничных узлах, или 2) сдвигая граничные узлы таким образом, чтобы получить равномерную сетку.

Понятно, что второй способ имеет смысл применять только когда шаг сетки достаточно мал, так как в противном случае искажение границы будет слишком велико. Далее мы будем говорить о первом способе построения сетки.

Предположим, что область $\bar{\Omega}$ целиком содержится в единичном квадрате $[0,1] \times [0,1]$. Сетку строим следующим образом. Выбираем $h = 1/n$ и в качестве граничных узлов выбираем все точки пересечения границы Γ с линиями $x_1 = ih$ и $x_2 = jh$, $i,j = \overline{0,n}$. Внутренние узлы, понятное дело, определяются как

$$\omega = \Big\{(ih, jh)\Big\}_{i,j=0}^{n} \cap \Omega.$$

Множество внутренних узлов в этом случае разделяется на *регулярные* — такие точки (x_1, x_2), что $(x_1 \pm h, x_2 \pm h) \in \omega$, и *нерегулярные* — которые не обладают таким свойством. Понятно, что в регулярных узлах оператор Лапласа можно приблизить на равномерном шаблоне со вторым порядком аппроксимации по формуле (36.3). А в нерегулярных узлах приходится использовать аппроксимацию вторых производных на неравномерном шаблоне вида

$$\Delta u(x_1, x_2) \approx \frac{1}{\hbar_1}\left(\frac{u_1^+ - u}{h_1^+} - \frac{u - u_1^-}{h_1^-}\right) + \frac{1}{\hbar_2}\left(\frac{u_2^+ - u}{h_2^+} - \frac{u - u_2^-}{h_2^-}\right), \qquad (36.7)$$

где

$$u = u(x_1, x_2), \quad u_1^\pm = u(x_1 \pm h_1^\pm, x_2), \quad u_2^\pm = u(x_1, x_2 \pm h_2^\pm), \quad \hbar_i = \frac{h_i^+ + h_i^-}{2}.$$

$\triangleright_3$ Покажите, что порядок аппроксимации такого разностного оператора в общем случае равен единице.

36.3 Метод конечных элементов

36.3.1 Слабая постановка задачи

Переформулируем исходную задачу (36.1) следующим образом. Для начала обозначим Ψ_0 множество всех кусочно-гладких функций

$$\psi : \bar{\Omega} \to \mathbb{R}$$

таких, что

$$\psi(x_1, x_2) = 0 \quad \forall (x_1, x_2) \in \Gamma.$$

Умножим обе части дифференциального уравнения (36.1a) на произвольную функцию $\psi \in \Psi_0$ и проинтегрируем их по области Ω:

$$\int_\Omega \psi \Delta u \, d\Omega = \int_\Omega \psi f \, d\Omega.$$

Воспользуемся в левой части данного тождества формулой интегирования по частям, которая в многомерном случае имеет вид

$$\int_\Omega u \frac{\partial v}{\partial x_i} \, d\Omega = \oint_\Gamma uv\nu_i \, d\Gamma - \int_\Omega v \frac{\partial u}{\partial x_i} \, d\Omega,$$

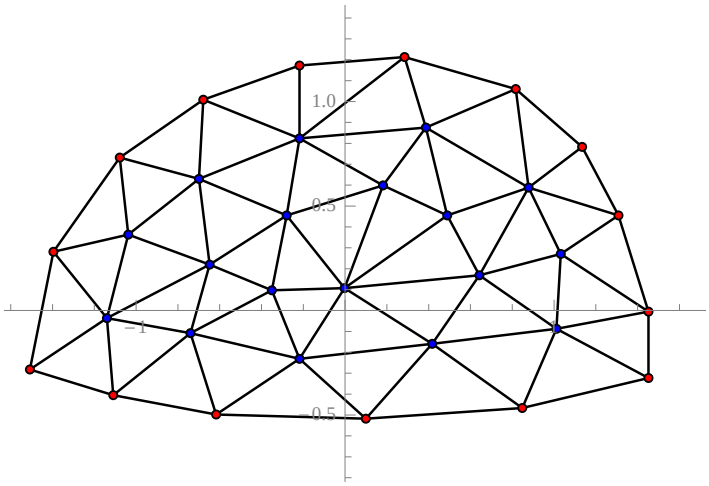
где ν_i — i-я компонента единичного вектора нормали к кривой Γ. У нас интеграл по границе будет равен нулю, и мы получим

$$\boxed{-\int_\Omega \frac{\partial u}{\partial x_1}\frac{\partial \psi}{\partial x_1} + \frac{\partial u}{\partial x_2}\frac{\partial \psi}{\partial x_2} \, d\Omega = \int_\Omega \psi f \, d\Omega.} \qquad (36.8)$$

Функция u, удовлетворяющая тождеству (36.8) при любых $\psi \in \Psi_0$, а также граничному условию $u|_\Gamma = \mu$, называется *слабым решением* задачи Дирихле (36.1).

Переход к слабой постановке задачи нам необходим для того, чтобы ослабить требование к гладкости решения: слабое решение не обязано быть дважды дифференцируемой функцией.

36.3.2 Сетка и базис



Зададим на области $\bar{\Omega}$ *треугольную* сетку, которая по сути является (неориентированным) графом и определяется следующими двумя составляющими.

1. Множество узлов (вершин)

$$\bar{\omega} = \{x_i = (x_{1,i}, x_{2,i})\}_{i \in J}.$$

   Множество всех индексов J традиционно разобьём на индексы внутренних

$$J_\omega = \{1, 2, \ldots, n\}$$

   и граничных узлов

$$J_\gamma = \{n+1, \ldots, n+m\}.$$

2. *Триангуляция* — множество отрезков (дуг), соединяющих узлы сетки.

К треугольным сеткам как правило предъявляются следующие требования:

- каждый узел сетки должен быть вершиной всех треугольников, которым он принадлежит;
- чем больше треугольники похожи на равносторонние, тем лучше (углы всех треугольников по возможности должны быть острыми).

Заметим также, что использование треугольной сетки влечет замену границы Γ на кусочно-линейную замкнутую кривую.

Чтобы не запутаться в множестве треугольников и их вершин, введем следующие обозначения.

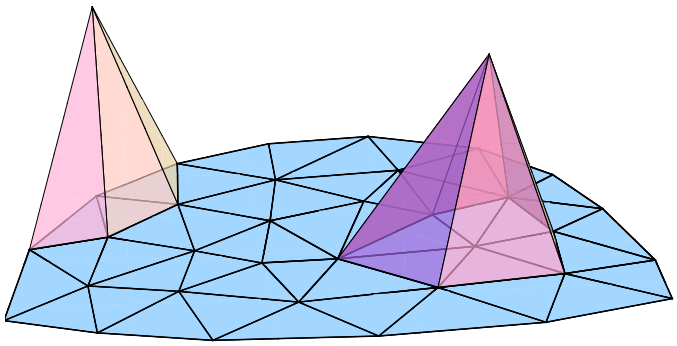
Пусть J_i — множество индексов всех точек, соединенных с i-ым узлом (упорядоченное по часовой стрелке для определенности);

D_i — многоугольник с вершинами $\{x_j\}$, где $\begin{cases} j \in J_i, & i \in J_\omega, \\ j \in J_i \cup \{i\}, & i \in J_\gamma, \end{cases}$

Δ_{ij} — треугольники с вершиной в x_i, из которых состоит D_i, $j = \overline{1, n_i}$.

Для построения приближенного решения построим фундаментальный кусочно-линейный базис для точек сетки $\bar{\omega}$:

$$\varphi_i(x_j) = \delta_{ij} = \begin{cases} 1, & i = j, \\ 0, & i \neq j, \end{cases} \qquad \operatorname{supp} \varphi_i = D_i, \quad i \in J. \tag{36.9}$$



Таким образом, график φ_i представляет собой пирамиду с вершиной в точке $(x_{1,i}, x_{2,i}, 1)$ и основанием D_i. На каждом треугольнике Δ_{ij} функция φ_i является многочленом первой степени по обеим переменным:

$$\varphi_i(x_1, x_2)|_{\Delta_{ij}} = a_{ij}x_1 + b_{ij}x_2 + c_{ij}, \quad j = \overline{1, n_i}. \tag{36.10}$$

Такие функции называют *пирамидальными*.

36.3.3 Метод Галеркина

Приближенное решение задачи Дирихле будем искать в виде разложения по построенному базису (36.9):

$$\varphi = \sum_{i \in J} \alpha_i \varphi_i = \overbrace{\sum_{i \in J_\gamma} \alpha_i \varphi_i}^{\varphi^\gamma} + \overbrace{\sum_{i \in J_\omega} \alpha_i \varphi_i}^{\varphi^\omega}. \tag{36.11}$$

Так как на границе должно выполнятся условие

$$\varphi|_\Gamma = \mu,$$

сразу имеем

$$\alpha_i = \mu(x_i), \quad i \in J_\gamma,$$

то есть

$$\varphi^\gamma = \sum_{i \in J_\gamma} \mu(x_i)\varphi_i. \tag{36.12}$$

Осталось найти коэффициенты α_i, $i \in J_\omega$. Сделаем это методом Галеркина, который применительно к слабой формулировке задачи сводится к требованию выполнения (36.8) при подстановке

$$u \leftarrow \varphi, \quad \psi \leftarrow \varphi_i, \quad \forall i \in J_\omega.$$

Введем обозначение

$$\langle u, \psi \rangle = -\int_\Omega \frac{\partial u}{\partial x_1}\frac{\partial \psi}{\partial x_1} + \frac{\partial u}{\partial x_2}\frac{\partial \psi}{\partial x_2}\, d\Omega.$$

Тогда указанная подстановка с учетом линейности дает

$$\langle \varphi, \varphi_i \rangle = \left\langle \varphi^\gamma + \sum_{j \in J_\omega} \alpha_j \varphi_j, \varphi_i \right\rangle = \langle \varphi^\gamma, \varphi_i \rangle + \sum_{j \in J_\omega} \alpha_j \langle \varphi_j, \varphi_i \rangle,$$

и из (36.8) получаем СЛАУ

$$\boxed{\Phi\alpha = g,} \tag{36.13}$$

где

$$\alpha = (\alpha_1, \ldots, \alpha_n)^T,$$

(напомним, что внутренние узлы имеют индексы от 1 до n), элементы матрицы Φ имеют вид

$$\boxed{\phi_{ij} = \phi_{ji} = \langle \varphi_j, \varphi_i \rangle,} \tag{36.14}$$

а элементы вектора g с учетом (36.12) имеют вид

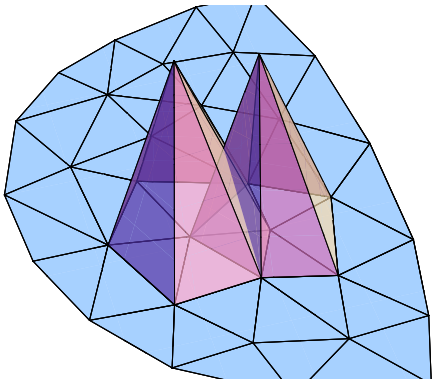
$$\boxed{g_i = \int_\Omega f\varphi_i d\Omega - \sum_{j \in J_\gamma} \mu(x_j)\langle \varphi_j, \varphi_i \rangle.} \tag{36.15}$$

Теперь аккуратно вычислим коэффициенты $\langle \varphi_j, \varphi_i \rangle$. Понятно, что в силу компактности носителя базисных функций, эти коэффициенты будут отличны от нуля только в случае, когда

$$\operatorname{mes} G_{ij} \neq 0, \quad \text{где} \quad G_{ij} = D_i \cap D_j.$$

Если это условие выполняется при $i \neq j$, то G_{ij} является объединением двух треугольников,

$$G_{ij} = \Delta_{ik} \cup \Delta_{i\ell} = \Delta_{jr} \cup \Delta_{js}.$$



Таким образом, вспоминая определение (36.10), получаем

$$\langle \varphi_j, \varphi_i \rangle = -\int_{G_{ij}} \frac{\partial \varphi_i}{\partial x_1}\frac{\partial \varphi_j}{\partial x_1} + \frac{\partial \varphi_i}{\partial x_2}\frac{\partial \varphi_j}{\partial x_2}\, dx = \\ = -S(\Delta_{ik})(a_{ik}a_{jr} + b_{ik}b_{jr}) - S(\Delta_{i\ell})(a_{i\ell}a_{js} + b_{i\ell}b_{js}), \quad (36.16a)$$

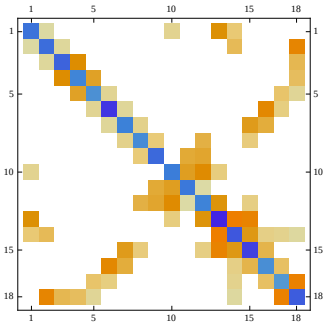
где $S(\Delta_{..})$ — площадь треугольника, которую можно вычислить по формуле (17.8).

$\triangleright_4$ Запишите формулы для вычисления коэффициентов a_{ij} и b_{ij}.

В случае же $i = j$ имеем

$$\langle \varphi_i, \varphi_i \rangle = -\sum_{j=1}^{n_i} S(\Delta_{ij})(a_{ij}^2 + b_{ij}^2). \quad (36.16b)$$

Заметим, что получаемая таким образом матрица $\Phi = \Big(\langle \varphi_j, \varphi_i \rangle\Big)_{i,j=1}^{n}$ является а) симметричной, б) положительно определенной, в) разреженной, а также г) обладает диагональным преобладанием (в случае «хорошей» триангуляции). Эти свойства благоприятно влияют на сходимость итерационных методов, которые как правило применяются для решения СЛАУ (36.13).



Осталось поговорить о вычислении интеграла

$$\int_{\Omega} f\varphi_i d\Omega = \int_{D_i} f(x)\varphi_i(x)dx$$

в формуле (36.15). Здесь $x = (x_1, x_2)$. Понятно, что в общем случае нужно использовать кубатурную формулу, а так как сетка у нас треугольная, очень хорошо подойдет вариант (17.9):

$$\iint_{\Delta} f(x,y)dxdy \approx \frac{S_\Delta}{3}\Big(f(p_0) + f(p_1) + f(p_2)\Big),$$

где p_i — вершины треугольника Δ. Учитывая специфический вид функции φ_i, данная формула дает нам в итоге

$$\boxed{\int_{D_i} f(x)\varphi_i(x)dx \approx \frac{f(x_i)}{3}\sum_{j=1}^{n_i} S(\Delta_{ij}).} \tag{36.17}$$

$\triangleright_5$ Перечислите преимущества и недостатки МКЭ по сравнению с сеточным методом

37 Численное решение двумерного нестационарного уравнения теплопроводности

Задача:

$$\dot u(x,t) = \Delta u(x,t) + f(x,t), \quad x \in G \subset \mathbb{R}^2, \quad t > 0, \tag{37.1a}$$
$$u(x,0) = u_0(x), \quad x \in \bar G, \tag{37.1b}$$
$$u(x,t) = \mu(x,t), \quad x \in \partial G. \tag{37.1c}$$

Здесь и далее $x = (x_1, x_2)$, $\dot u = \frac{\partial u}{\partial t}$, $\Delta = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2}$.

Будем для краткости рассматривать квадратную область $\bar G = [0,1]\times[0,1]$, пусть также $t \in [0,T]$, тогда область Ω — прямоугольный параллелепипед. Как обычно, строим в Ω равномерную сетку:

$$\bar\omega = \bar\omega_h \times \bar\omega_\tau, \quad \bar\omega_h = \left\{ x_{ij} = (ih, jh) \right\}_{i,j=0}^{n}, \quad \bar\omega_\tau = \left\{ k\tau \right\}_{k=0}^{m},$$

$h = 1/n$, $\tau = T/m$. Стандартные обозначения:

$$y = y_{ij}^k \approx u_{ij}^k = u(x_{ij}, t_k), \quad \hat y = y_{ij}^{k+1},$$

J_ω — индексы внутренних, J_γ — индексы граничных узлов.

37.1 Явная разностная схема

37.1.1 Построение

Полагаем $\dot u \approx u_t$, $\Delta u \approx \Lambda u$, $u \leftarrow y$:

$$y_t = \frac{\hat y - y}{\tau} = \Lambda y + \varphi, \quad (i,j,k) \in J_\omega, \tag{37.2a}$$
$$y_{ij}^0 = u_0(ih, jh), \quad i,j = \overline{1, n-1}, \tag{37.2b}$$
$$y = \mu, \quad (i,j,k) \in J_\gamma. \tag{37.2c}$$

Это — двумерный аналог схемы (34.3). Здесь

$$\varphi = \varphi_{ij}^k = f_{ij}^k, \quad \Lambda y = \Lambda_1 y + \Lambda_2 y,$$
$$\Lambda_1 y = \frac{1}{h^2}(y_{i-1,j}^k - 2y_{i,j}^k + y_{i+1,j}^k), \quad \Lambda_2 y = \frac{1}{h^2}(y_{i,j-1}^k - 2y_{i,j}^k + y_{i,j+1}^k). \tag{37.3}$$

Перепишем (37.2a) в виде

$$\hat y = y + \tau(\Lambda y + \varphi), \tag{37.4}$$

что в индексной форме дает явный алгоритм вычисления решения на новом слое:

$$y_{ij}^{k+1} = \left(1 - 4\frac{\tau}{h^2}\right) y_{ij}^k + \frac{\tau}{h^2}\left(y_{i-1,j}^k + y_{i+1,j}^k + y_{i,j-1}^k + y_{i,j+1}^k\right) + \tau\varphi_{ij}^k,$$
$$i,j = \overline{1,n-1}. \quad (37.5)$$

Погрешность аппроксимации схемы, очевидно, равна $O(h^2+\tau)$.

37.1.2 Устойчивость

Устойчивость (по начальному условию) исследуем аналогично одномерному случаю. Представим приближенное решение на k-ом слое в виде вектора $v^{(k)}$ размерности $(n-1)^2$:

$$v^{(k)} = \left(y_{11}^k, \ldots, y_{n-1,n-1}^k\right)^T \quad (37.6)$$

и предположим, что $\varphi = \mu = 0$. Тогда (37.4) примет вид

$$v^{(k+1)} = (I + \tau A)v^{(k)}, \quad (37.7)$$

где A — блочная матрица из формулы (36.5):

$$A = \frac{1}{h^2}\begin{bmatrix} B & I & & & \\ I & B & I & & \\ & \ddots & \ddots & \ddots & \\ & & I & B & I \\ & & & I & B \end{bmatrix}, \quad B = \begin{bmatrix} -4 & 1 & & & \\ 1 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -4 & 1 \\ & & & 1 & -4 \end{bmatrix}, \quad (37.8)$$

Лемма 37.1. *Собственные значения матрицы A (37.8) равны*

$$\lambda_{k\ell} = -\frac{4}{h^2}\left(\sin^2\frac{\pi k}{2n} + \sin^2\frac{\pi \ell}{2n}\right), \quad k,\ell = \overline{1,n-1}. \quad (37.9)$$

Доказательство. Для нумерации компонент собственных векторов матрицы A, который обозначим $\xi \in \mathbb{R}^{(n-1)^2}$, аналогично (37.6) будем использовать двойные индексы:

$$\xi = (\xi_{11}, \ldots, \xi_{n-1,n-1})^T.$$

Из райкома нам поступила информация, что эти компоненты имеют вид

$$\xi_{ij} = \eta_i\zeta_j \neq 0, \quad i,j = \overline{1,n-1}, \quad \eta_0 = \eta_n = \zeta_0 = \zeta_n = 0.$$

Подставляя эти значения в

$$A\xi = \lambda\xi,$$

получаем

$$\zeta_j(\eta_{i-1} - 2\eta_i + \eta_{i+1}) + \eta_j(\zeta_{j-1} - 2\zeta_j + \zeta_{j+1}) = \lambda h^2 \eta_i \zeta_j,$$

или

$$\eta_{i-1} - 2\eta_i + \eta_{i+1} = \left(\lambda h^2 - \frac{\zeta_{j-1} - 2\zeta_j + \zeta_{j+1}}{\zeta_j}\right)\eta_i, \quad i = \overline{1, n-1}.$$

А это — известная нам задача на собственные значения (34.14), то есть

$$\lambda h^2 - \frac{\zeta_{j-1} - 2\zeta_j + \zeta_{j+1}}{\zeta_j} = -4\sin^2\frac{\pi k}{2n}, \quad j = \overline{1, n-1}.$$

Повторяя проделанную процедуру, получаем (37.9). ■

Возвращаемся к доказательству устойчивости. Из (37.7) получаем критерий: спектральный радиус матрицы $I + \tau A$ не должен превосходить единицы, то есть

$$|1 + \tau\lambda_{k\ell}| \leqslant 1.$$

Отсюда с учетом $-\frac{8}{h^2} < \lambda_{k\ell} < 0$ (лемма 37.1) получаем жестокое ограничение на величину τ:

$$\boxed{\tau \leqslant \frac{h^2}{4}.} \tag{37.10}$$

Из-за этого ограничения на устойчивость явная схема малопригодна для практических вычислений.

37.2 Метод переменных направлений

В качестве альтернативы явной схеме на ум сразу приходит неявная:

$$\hat{y} = y + \tau(\Lambda\hat{y} + \hat{\varphi}). \tag{37.11}$$

Действительно, эта схема имеет такой же порядок точности и при этом устойчива при любых τ и h.

$\triangleright_1$ Докажите это.

Однако тут встает другое препятствие: для реализации неявной схемы на каждом слое необходимо решать СЛАУ большой размерности, а это $O(n^4)$ операций в лучшем случае.

Поэтому сейчас мы рассмотрим более изощренную схему — *схему переменных направлений*, которую можно назвать «полуявной» ___. Эта схема имеет второй порядок аппроксимации по t и безусловно устойчива.

37.3 Построение схемы

Фокус заключается во введении промежуточного $(k+\frac{1}{2})$-го слоя $\bar{y}$. Вычисление $\hat{y}$ осуществляется в два этапа. Сначала находят значения $\bar{y}$ из

$$\frac{\bar{y}-y}{0.5\tau} = \Lambda_1\bar{y} + \Lambda_2 y + \varphi. \tag{37.12a}$$

Заметьте, что эта «подсхема» неявна только по переменной x_1 (см. (37.3)), поэтому значения $\bar{y}$ во всех узлах можно найти за $O(n^2)$ операций ($n-1$ методов прогонки для систем размерности $n-1$).

После этого уже находятся интересующие нас значения $\hat{y}$ по схеме

$$\frac{\hat{y}-\bar{y}}{0.5\tau} = \Lambda_1\bar{y} + \Lambda_2\hat{y} + \varphi, \tag{37.12b}$$

которая неявна уже по переменной x_2 и решается аналогично. Таким образом, вычисление решения на каждом слое требует $O(n^2)$ операций.

В (37.12) везде полагается

$$\varphi = \varphi_{ij}^k = f(x_{ij}, t_k + 0.5\tau) = \bar{f}.$$

Кроме этого нужно определить граничные условия. Для (37.12b), очевидно, нужно положить

$$\hat{y} = \hat{\mu} = \mu_{ij}^{k+1} \quad \text{при} \quad j=0,\ j=n, \quad i=\overline{1,n-1}. \tag{37.13}$$

А вот что делать с граничными условиями на среднем слое, то есть чему полагать $\bar{y}_{0j}$ и $\bar{y}_{nj}$, $j=\overline{1,n-1}$, в формуле (37.12a), сходу сказать трудно. Между тем, не зная этих значений, $\bar{y}$ определить вообще невозможно.

Для решения вопроса райком рекомендует выразить $\bar{y}$ через y и $\hat{y}$ путем вычитания (37.12a) из (37.12b):

$$\bar{y} = \frac{\hat{y}+y}{2} - \frac{\tau}{4}\Lambda_2(\hat{y}-y). \tag{37.14}$$

Эти соотношения справедливы для внутренних узлов, а мы потребуем их выполнения и на граничных узлах, в которых определен оператор Λ_2, то есть как раз там, где нам нужно. То есть, мы полагаем

$$\bar{y} = \bar{\mu} = \frac{\hat{\mu}+\mu}{2} - \frac{\tau}{4}\Lambda_2(\hat{\mu}-\mu), \quad i=0,\ i=n, \quad j=\overline{1,n-1}. \tag{37.15}$$

Таким образом, схема переменных направлений определяется формулами (37.12), (37.13), (37.15).

$\triangleright_2$ Запишите данную схему в индексной форме.

37.3.1 Порядок аппроксимации

Для исследования порядка аппроксимации метода переменных направлений, исключим $\bar{y}$ из схемы: подставим (37.14) в (37.12a) и сделаем замену $\hat{y} = y + \tau y_t$. Если все сделать правильно, получится

$$\boxed{\left(I - \frac{\tau}{2}\Lambda_1 - \frac{\tau}{2}\Lambda_2 + \frac{\tau^2}{4}\Lambda_1\Lambda_2\right) y_t = \Lambda y + \varphi.} \tag{37.16}$$

Теперь вычисляем погрешность аппроксимации — подставляем в эту формулу $y = u$ и строим невязку:

$$\psi = \Lambda u + \varphi - \left(I - \frac{\tau}{2}\Lambda_1 - \frac{\tau}{2}\Lambda_2 + \frac{\tau^2}{4}\Lambda_1\Lambda_2\right) u_t.$$

Используя очевидные соотношения

$$\varphi = \bar{f} = f(x_{ij}, t_k + \tau/2) = \bar{\ddot{u}} - \Delta\bar{u} = u_t - \Lambda\bar{u} + O(h^2 + \tau^2),$$
$$\frac{1}{2}(\hat{u} + u) = \bar{u} + O(\tau^2),$$

для достаточно гладких u получаем

$$\psi = \Lambda u + u_t - \Lambda\bar{u} - \left(I - \frac{\tau}{2}\Lambda_1 - \frac{\tau}{2}\Lambda_2 + \frac{\tau^2}{4}\Lambda_1\Lambda_2\right) u_t + O(h^2 + \tau^2) =$$
$$= \Lambda u - \Lambda\bar{u} + \frac{1}{2}\Lambda(\hat{u} - u) + O(h^2 + \tau^2) = -\Lambda\bar{u} + \frac{1}{2}\Lambda(\hat{u} + u) + O(h^2 + \tau^2) =$$
$$= O(h^2 + \tau^2).$$

Итак, как и было обещано, схема имеет второй порядок аппроксимации по всем переменным.

Что же касается безусловной (при любых h и τ) устойчивости, то ее мы доказывать не будем.

★ Многосеточный метод

★.1 Введение

Как мы уже знаем, численное решение линейных дифференциальных уравнений в частных производных на практике как правило сводится к решению СЛАУ большой размерности с разреженной матрицей. Сейчас мы рассмотрим наиболее эффективный из известных методов решения таких систем — многосеточный метод. Он применим также к интегральным уравнениям и другим задачам.

Опишем многосеточный метод для задачи Дирихле (36.1) с нулевыми граничными условиями и соответствующей простейшей разностной схемы (36.4):

$$\frac{1}{h^2}(y_{i-1,j} + y_{i+1,j} + y_{i,j-1} + y_{i,j+1} - 4y_{i,j}) = \varphi_{i,j}, \quad i,j = \overline{1,n-1}, \tag{1a}$$

$$y_{i,j} = 0, \quad (i,j) \in J_\gamma. \tag{1b}$$

Напомним, что если искомую сеточную функцию представить в виде вектора

$$y = (y_{1,1}, \ldots, y_{1,n-1}, y_{2,1}, \ldots, y_{2,n-1}, \ldots\ldots, y_{n-1,1}, \ldots, y_{n-1,n-1})^T,$$

то в матричной форме эта система записывается как

$$Ay = \phi, \tag{2}$$

где A — матрица разностного оператора Лапласа, которая имеет блочный вид (37.8). Согласно лемме (37.1) собственные значения этой матрицы равны

$$\lambda_{k\ell} = -\frac{4}{h^2}\left(\sin^2\frac{\pi k}{2n} + \sin^2\frac{\pi \ell}{2n}\right), \quad k,\ell = \overline{1,n-1}. \tag{3}$$

Из схемы доказательства указанной леммы также нетрудно видеть (см. (34.13)), что компоненты соответствующих собственных векторов $\xi_{k\ell}$ имеют вид

$$(\xi_{k\ell})_{ij} = \sin\frac{\pi k i}{n}\sin\frac{\pi \ell j}{n}, \quad i,j = \overline{1,n-1}. \tag{4}$$

Для построения многосеточного метода вспомним сначала простейший итерационный метод решения СЛАУ – метод простой итерации.

★.2 Метод простой итерации

Для системы (2) стандартный метод простой итерации выглядит как

$$y^{p+1} = (I + A)y^p - \phi. \tag{5}$$

Из курса вычислительных методов алгебры мы помним, что этот метод будет сходиться тогда и только тогда, когда все собственные значения матрицы $I + A$, равные $1 + \lambda_{k\ell}$, будут меньше единицы по модулю, то есть когда

$$-2 < \lambda_{k\ell} < 0.$$

Согласно (3) мы имеем

$$M = -\frac{8}{h^2} < \lambda_{k\ell} \lessapprox -2\pi^2 = m, \tag{6}$$

поэтому в чистом виде метод простой итерации не пригоден для решения нашей задачи. Можно, однако, модифицировать этот метод введением параметра $\tau \in \mathbb{R}$:

$$y^{p+1} = (I + \tau A)y^p - \tau\phi. \tag{7}$$

Так как все $\lambda_{k\ell}$ отицательны, за счет выбора τ можно добиться сходимости итерационного процесса (7), то есть выполнения свойства

$$\rho(I + \tau A) = \max_{k,\ell} |1 + \tau\lambda_{k\ell}| < 1,$$

при любых h. Здесь $\rho(\cdot)$ — спектральный радиус матрицы (величина максимального по модулю собственного значения).

Итерационные процессы вида $y^{p+1} = By^p + g$ сходятся со скоростью геометрической прогрессии со знаменателем, равным $\rho(B)$. Поэтому оптимальный параметр τ следует выбирать путем решения задачи минимизации

$$\max_{k,\ell} |1 + \tau\lambda_{k\ell}| \to \min.$$

Зная границы (6) нетрудно решить эту задачу:

$$\tau_{\text{опт}} = -\frac{2}{M + m}, \tag{8}$$

причем соответствующий показатель сходимости равен

$$q_{\text{опт}} = \frac{M - m}{M + m}. \tag{9}$$

Заметим, что $q_{\text{опт}} \to 1$ при $h \to 0$. Например, при $h = 0.05$ (сетка 20×20) имеем $q_{\text{опт}} \approx 0.99$. Таким образом, при уменьшении шага сетки не только растет размерность СЛАУ, но еще и замедляется сходимость итерационного процесса. То есть даже модифицированный метод простой итерации не годится для решения нашей задачи. Чтобы построить на его основе более совершенный метод, нужно провести дополнительный анализ.

★.3 Сглаживающее свойство метода простой итерации

Введем следующие обозначения. Точное решение системы (2) обозначим y^*, $y^p := y$, $y^{p+1} := \hat{y}$. Рассмотрим также невязку

$$r = Ay - \phi, \quad \hat{r} = A\hat{y} - \phi,$$

а также погрешность

$$e = y - y^*.$$

В этих обозначениях метод простой итерации (7) примет вид

$$\hat{y} = y + \tau r. \tag{7'}$$

Наша локальная цель — исследовать изменение невязки r в ходе итерационного процесса (7'). По определению имеем

$$\hat{r} = A\hat{y} - \phi = A(y + \tau r) - \phi = (I + \tau A)r. \tag{10}$$

Разложим теперь невязку по базису из собственных векторов матрицы A (4):

$$r = \sum_{k,\ell} \alpha_{k\ell}\xi_{k\ell}, \quad \hat{r} = \sum_{k,\ell} \hat{\alpha}_{k\ell}\xi_{k\ell}. \tag{11}$$

Тогда из (10) получаем

$$\hat{\alpha}_{k\ell} = (1 + \tau\lambda_{k\ell})\alpha_{k\ell}. \tag{12}$$

Это ключевое соотношение для понимания ситуации. Оно показывает, что с ростом числа итераций p в разложении невязки (11) быстрее всего убывают компоненты, для которых

$$|1 + \tau\lambda_{k\ell}| \ll 1.$$

В частности, при $\tau = \tau_{\text{опт}}$ (8) отсюда следует, что при $p \to \infty$ в разложении невязки будут преобладать собственные векторы ξ_{11} и $\xi_{n-1,n-1}$ (так как именно для $k = \ell = 1$ и $k = \ell = n - 1$ выражение $|1 + \tau_{\text{опт}}\lambda_{k\ell}|$ достигает наибольшего значения). Учитывая вид собственных векторов (4) (с ростом $k\ell$ увеличивается «частота колебаний» компонент) можно сказать, что при выборе τ по формуле (8) каждая итерация метода (7) «гасит средние частоты» в невязке.

Если же нужно «сгладить» невязку, то есть нейтрализовать в ней высокочастотные компоненты $\xi_{k\ell}$ ($k, \ell \geqslant \frac{n}{2}$), то необходимо взять другое значение весового коэффициента τ, а именно

$$\tau^* = -\frac{2}{\frac{8}{h^2} + \frac{2}{h^2}} = \frac{h^2}{5}. \tag{13}$$

$\triangleright_1$ Обоснуйте эту формулу.

★.4 Двухсеточный метод

Итак, рассмотрим метод простой итерации (7) с параметром $\tau = \tau^*$ (13). Согласно вышеизложенному, после нескольких итераций невязка $r = Ay - \phi$ (y — текущее приближение) будет гладкой сеточной функцией. Заметим, что +

$$A(y - y^*) = Ay - b = r,$$

то есть погрешность $e = y - y^*$ является решением СЛАУ

$$Ae = r. \tag{14}$$

Основное отличие этой системы от исходной (2) заключается в том, что разложение правой части (14) по векторам $\xi_{k\ell}$ содержит (в основном) только низкочастотные компоненты ($k, \ell \leqslant \frac{n}{2}$), то есть сеточную функцию r можно без существенной потери точности приблизить на более крупной (с меньшим числом узлов) сетке. Удобнее всего уменьшить n в два раза (считаем n четным), то есть попросту выбросить из $r = (r_{ij})_{i,j=1}^{n-1}$ все элементы с нечетными индексами i или j.

Соответствующее отображение

$$R : \mathbb{R}^{(n-1)^2} \to \mathbb{R}^{(\frac{n}{2}-1)^2} \tag{15}$$

называют *ограничением*. Обозначим

$$\tilde{r} = Rr$$

и запишем СЛАУ

$$\tilde{A}\tilde{e} = \tilde{r}, \tag{16}$$

где $\tilde{A}$ — матрица разностного оператора Лапласа соответствующей размерности. Решив эту систему (а решается она проще хотя бы потому, что ее размерность в два раза меньше исходной), находим $\tilde{e}$ — приближенное значение ошибки. Для уточнения решения y на мелкой сетке необходимо проинтерполировать, или продолжить, сеточную функцию $\tilde{e}$ на мелкую сетку. Реализуется это простой линейной интерполяцией по каждой пере-

менной.

Отображение

$$P : \mathbb{R}^{(\frac{n}{2}-1)^2} \to \mathbb{R}^{(n-1)^2}, \tag{17}$$

которое осуществляет такое преобразование, называется *продолжением*. Таким образом, уточненное значение $\hat{y}$ вычисляется по формуле

$$\hat{y} = y - P\tilde{e}. \tag{18}$$

Вышеописанный процесс представляет собой одну итерацию *двухсеточного метода*.

★.5 Многосеточный метод

Понятно, при решении уравнения для ошибки на грубой сетке (16) можно рекурсивно использовать двухсеточный метод, введя в рассмотрение еще более грубую сетку с числом узлов $n/4 - 1$ по каждой переменной, и так далее. В результате получим *многосеточный метод*. Опишем его более строго.

Зафиксируем некоторое $N \in \mathbb{Z}$ и рассмотрим на квадрате $[0, 1] \times [0, 1]$ множество квадратных сеток

$$\{\omega_s\}_{s=1}^{N},$$

построенных по соответствующим шагам

$$h_s = \frac{1}{n_s}, \quad n_s = 2^s.$$

Рассмотрим также A_s — матрицы разностного оператора Лапласа на этих сетках. Наша цель — решить задачу на самой мелкой сетке ω_N, то есть решить СЛАУ

$$A_N y = \phi_N. \tag{19}$$

Функцию, осуществляющую одну итерацию многосеточного метода для решения СЛАУ вида $A_s y = b$, начиная с начального приближения y^0, обозначим

$$MGM(s, y^0, b).$$

Таким образом, итерационный процесс решения (19) будет иметь вид

$$y^{p+1} = MGM(N, y^p, \phi_N), \quad p = 1, 2, \ldots$$

Опишем алгоритм работы функции MGM.

$MGM(s, y^0, b)$
{

1. Если $s = 1$, возвращаем $y = A^{-1}b$, иначе идем дальше.
2. Делаем M итераций метода простой итерации для сглаживания невязки:
$$v^0 = y^0, \quad v^{p+1} = v^p + \tau_s^*(A_s v^p - b), \quad p = \overline{0, M-1}.$$
Здесь τ_s^* вычисляется по формуле (13).
3. Ограничиваем невязку на сетку ω_{s-1}: $r = R(A_s v^M - b)$.
4. Для нахождения поправки рекурсивно делаем K итераций многосеточного метода:
$$e^0 = 0, \quad e^{p+1} = MGM(s-1, e^p, r), \quad p = \overline{0, K-1}.$$
5. Возвращаем $y = v^M - Pe^K$.

}

Здесь использованы операторы ограничения и продолжения (15), (17). Что касается параметров M и K то их обычно берут равными 1 или 2. Если K не брать слишком большим, то в большинстве случаев сложность одного вызова функции MGM составляет $O(N_u)$ операций, где N_u — количество неизвестных в решаемой СЛАУ.

Что касается скорости сходимости, то для решения СЛАУ (19) с относительной точностью ε требуется всего

$$O(N_u |\log_{10} \varepsilon|),$$

операций, что является оптимальной асимптотической сложностью для итерационных методов.