Содержание

1 Конечные разности и их свойства

Рассмотрим дискретные функции, т.е функции, заданные таблицей.

x	$f(x)$
x_0	f_0
x_1	f_1
$\vdots$	$\vdots$
x_m	f_m

Как анализировать такие функции? Предположим, что значение аргумента изменяется с постоянным шагом h, т.е. $x_m = x_0 + mh$ (h — шаг таблицы). Введём обозначение $f(x_m) = f_m$. Введём $\Delta_h f(x_m) = f(x_{m+1}) - f(x_m) = f(x_0 + (m+1)h) - f(x_0 + mh)$ (аналог дифференциала). Так как x_0 и h — константы, то $f(m)$ — функция целого аргумента. В дальнейшем будем рассматривать только такие функции, так как в любой момент можно перейти к x.

Введём понятие первой конечной разности функции f в точке m:

$$\Delta f(m) \stackrel{\text{def}}{=} \Delta f(m+1) - \Delta f(m) = f_{m+1} - f_m$$

Аналогично можно ввести конечные разности более высокого порядка:

$$\Delta^2 f(m) \stackrel{\text{def}}{=} \Delta f(m+1) - \Delta f(m) = f_{m+2} - f_{m+1} - f_{m+1} + f_m = f_{m+2} - 2f_{m+1} - f_m$$

$$\Delta^3 f(m) \stackrel{\text{def}}{=} \Delta^2 f(m+1) - \Delta^2 f(m) = \cdots = f_{m+3} - 3f_{m+2} + 3f_{m+1} - f_m$$

В общем случае имеем:

$$\Delta^k f(m) = \Delta^{k-1} f(m+1) - \Delta^{k-1} f(m)$$

Свойства конечных разностей

1. $\Delta\alpha = \alpha - \alpha = 0$, где $\alpha = f(k, m) = const$
2. $\Delta(\alpha f_m) = \alpha f_{m+1} - \alpha f_m = \alpha \Delta f_m$
3. $\Delta(f_m + g_m) = \Delta f_m + \Delta g_m$
4. $\Delta(f_m \cdot g_m) = f_{m+1} g_{m+1} - f_m g_m \pm f_{m+1} g_m = f_{m+1} \Delta g_m + g_m \Delta f_m \equiv f_m \Delta g_m + g_{m+1} \Delta f_m$
   $f_{m+1} = f(x_m + h)$, при $h \to 0$ переходит в аналог дифференциального случая.
5. $\Delta m^s = (m+1)^s - m^s = m^s + s m^{s-1} + \frac{s(s-1)}{2} m^{s-2} + \cdots - m^s$
   Конечная разность от многочлена степени m есть многочлен степени $m-1$. В этом виде определение годится и для дифференциального случая.

Таблица конечных разностей

1. $\Delta\alpha^m = \alpha^{m+1} - \alpha^m = \alpha^m(\alpha - 1)$
2. $\Delta \sin k = \sin(k+1) - \sin k = 2 \sin \frac{1}{2} \cos(k + \frac{1}{2})$
3. $\Delta \cos k = \cos(k+1) - \cos k = 2 \sin \frac{1}{2} \sin(k + \frac{1}{2})$

2 Суммирование функций

Ранее осуществлялся поиск $\varphi(k)$. Рассмотрим обратную задачу — поиск $F(k)$ по заданной $\varphi(k)$.

$$\Delta F(k) = \varphi(k)$$

Запиишем это равенство при различных значениях k:

$$\begin{gathered} F_1 - F_0 = \varphi_0 \\ F_2 - F_1 = \varphi_1 \\ F_3 - F_2 = \varphi_2 \\ \dots \\ F_{n+1} - F_n = \varphi_n \end{gathered}$$

Суммируя, получим дискретный аналог формулы Ньютона-Лейбница:

$$\sum_{k=0}^{n} \varphi(k) = F_{n+1} - F_0 \tag{1}$$

Найдем суммы различных функций:

1. $\sum_{k=0}^{n} \alpha^k = \frac{\alpha^{n+1} - \alpha^0}{\alpha - 1}$
2. $\sum_{k=0}^{n} \cos(k + \frac{1}{2}) = \frac{\sin(n+1)}{2 \sin \frac{1}{2}} - \sin 0$
3. $\sum_{k=0}^{n} \cos(k) \stackrel{\text{def}}{=} \sin(an + b)$
   a и b находятся методом неопределенных коэффициентов.
4. $\sum_{k=0}^{n} k^2 = an^3 + bn^2 + cn + d$
   Коэффициенты ищутся аналогично предыдущему случаю.
5. $\sum_{k=0}^{n} ka^k = ?$
   В непрерывном случае следовало бы интегрировать по частям.

Формула Абеля суммирования по частям

Рассмотрим непрерывный случай: $u(x), v(x), U(x) = \int_0^x u(t)\,\mathrm{d}t$. Тогда:

$$\int_a^b u \cdot v\,\mathrm{d}x = \int_a^b v\,\mathrm{d}U(x) = U(x) \cdot v(x)\Big|_a^b - \int_a^b U(x)\frac{\mathrm{d}v}{\mathrm{d}x}\,\mathrm{d}x$$

В дискретном случае введем:

$$u(k),\ v(k),\ U(k) = \sum_{i=0}^{k} u_i \tag{2}$$

Посчитаем конечную разность произведения исходных функций:
$\Delta(v_k \cdot U_k) = v_{k+1} \cdot \Delta U_k + \Delta v_k \cdot U_k$
$\Delta U_k \stackrel{\text{def}}{=} U_{k+1} - U_k = u_{k+1}$, так как все осталные k слагаемых сокращаются при вычитании сумм (2).

$u_{k+1}v_{k+1} = \Delta(v_k \cdot U_k) - U_k \cdot \Delta v_k$
После суммирования с помощью формулы (1) получим формулу Абеля:

$$\sum_{k=m}^{n} u_{k+1} \cdot v_{k+1} = v_k \cdot U_k \Big|_m^{n+1} - \sum_{k=m}^{n} U_k \Delta v_k$$

Теперь можно легко вычислить сумму из пункта 5.

3 Разностные уравнения и их порядок

Рассмотрим аналогию в непрерывном случае — дифференциальные уравнения:

$$F(y^{(m)}, y^{(m-1)}, \ldots, y'', y', y, t) = 0, \text{ где } y = y(t)$$

Порядок дифференциального уравнения определяется порядком старшей производной (в данном случае m). Эта величина задает количество начальных условий, необходимых для однозначного решения задачи, либо число ЛНЗ решений решений в линейных уравнениях.

Аналогично рассмотрим разностные уравнения:

$$F(\Delta^m z, \Delta^{m-1} z, \ldots, \Delta^2 z, \Delta z, z, n) = 0, \text{ т.е. } z = z(n)$$

Порядок равен разности между конечным и начальным индексами (но не более m), например:

$$2\Delta^3 z_n + 3\Delta^2 z_n - z_n = 0$$

Преобразуем конечные разности:

$$\Delta^2 z_n = z_{n+2} - 2z_{n+1} - z_n$$
$$\Delta^3 z_n = z_{n+3} - 3z_{n+2} + 3z_{n+1} - z_n$$

Домножив на соответствующие коэффициенты эти два равенства и сложив их, получим:

$$2z_{n+3} - 3z_{n+2} + 0 + z_n - z_n = 0$$

Т.е. получим уравнение $n + 3 - n - 2 = 1$ порядка.

Для определения порядка разностного уравнения необходимо выразить все конечные разности через значения функции в различных точках, тогда порядком разностного уравнения является разность между наибольшим и наименьшим значением аргумента.

В общем случае разностное уравнение порядка s имеет следующий вид:

$$z_{n+s} = F(z_{n+s-1}, z_{n+s-2}, \ldots, z_{n+1}, z_n, n) \tag{3}$$

Для его решения необходимо s начальных условий: $z_0, z_1, \ldots, z_{s-1}$. Положим в (3) $n = 0$. Тогда аргументами F будут начальные условия и можно вычислить z_s.
Положим $n = 1$. Аргументами F будут $s - 1$ начальных условий и z_s, посчитанное на предыдущем шаге.
Это — пошаговый метод решения разностного уравнения. Его достоинство заключается в легкости реализации на ЭВМ, недостаток — в необходимости обсчета всех шагов.

В некоторых случаях можно получить аналитическое решение z_n, позволяющее вычислить z_n непосредственно, не определяя все предыдущие точки пошаговым методом.

Рассмотрим линейные разностные уравнения первого порядка:

$$z_{n+1} = \alpha z_n + \varphi(n),\ z_0 \text{ — начальное условие} \tag{4}$$

Начнем его решать пошаговым методом:

$$\begin{aligned} z_1 &= \alpha z_0 + \varphi_0 \\ z_2 &= \alpha z_1 + \varphi_1 = \alpha^2 z_0 + \alpha\varphi_0 + \varphi_1 \\ z_3 &= \alpha z_2 + \varphi_2 = \alpha^3 z_0 + \alpha^2\varphi_0 + \alpha\varphi_1 + \varphi_2 \end{aligned}$$

В общем виде:

$$z_n = \alpha^n z_0 + \sum_{k=0}^{n-1} \alpha^k \varphi_{n-1-k} \tag{5}$$

Предположим, что $\varphi_n = \beta =$const. Тогда $z_n = \alpha^n z_0 + \beta \sum_{k=0}^{n-1} \alpha^k$.
Свернем сумму:

$$z_n = \alpha^n z_0 + \frac{1-\alpha^n}{1-\alpha}\beta$$

Линейные разностные уравнения высших порядков с постоянными коэффициентами

$$\alpha_0 z_{n+s} + \alpha_1 z_{n+s-1} + \cdots + \alpha_s z_n = \varphi(n),\ z_0, \ldots, z_{s-1} \text{ — начальные условия} \tag{6}$$

Это уравнение линейное, т.к. величины z входят линейно; с постоянными коэффициентами, т.к. $\alpha_k \neq f(n)$; неоднородное, т.к. $\varphi_n \neq 0$.

По аналогии с дифференциальными уравнениями, начнем решать однородное уравнение:

$$\alpha_0 z_{n+s} + \alpha_1 z_{n+s-1} + \cdots + \alpha_s z_n = 0,\ z_0, z_1, \ldots, z_{s-1} \text{ — начальные условия} \tag{7}$$

Будем искать решение в виде $z_n = C\gamma^n$. Подставим его в (7) и сократим на C и γ^n:

$$\alpha_0\gamma^s + \alpha_1\gamma^{s-1} + \cdots + \alpha_s = 0 \tag{8}$$

Уравнение (8) называется характеристическим для уравнения (7), как, впрочем, и для исхдного уравнения. $\gamma_1, \gamma_2, \ldots, \gamma_s$ — корни характеристического уравнения.

Любая комбинация линейнонезависимых решений уравнения (7) также является его решением. Таким образом, общее решение уравнения (7) имеет вид:

$$z_n = \sum_{k=1}^{s} C_k \gamma_k^n$$

Если корень γ_1 имеет кратность q, то соответствующая группа решений имеет следующий вид: $P_{q-1}(n)\gamma_1^n$, где $P_{q-1}(n)$ — полином $q-1$ степени с произвольными коэффициентами.

Общее решение неоднородного уравнения (6) равняется сумме общего решения однородного уравнения и любого частного решения неоднородного уравнения:

$$z_{\text{неоднор}}(n) = z_{\text{однор}}(n) + z^*(n)$$

Если $\varphi(n)$ — линейная комбинация полиномов и показательных функций, то z^* подбирается в том же виде с помощью метода неопределенных коэффициентов, а в общем случае используется метод вариаций произвольных постоянных Лагранжа.

4 Разделенные разности и их связь с конечными

Если табличная функция задана с неравноотстоящими аргументами, то для характеристики быстроты изменений функции вместо конечной разности вводится понятие разделенной разности (аналог производной для непрерывных функций):

$$f(x_s, x_{s+1}) = f(x_{s+1}, x_s) = \frac{f_{s+1} - f_s}{x_{s+1} - x_s} \text{ — разделенная разность 1 порядка}$$

Разделенная разность является симметричной функцией своих аргументов. Можно показать, что и разделенные разности более высокого порядка обладают этим свойством.

$$f(x_{s+2}, x_s) = \frac{f(x_{s+2}, x_{s+1}) - f(x_{s+1}, x_s)}{x_{s+2} - x_s}$$

Разделенная разность n порядка выглядит так:

$$f(x_{s+n}, \ldots, x_s) = \frac{f(x_{s+n}, \ldots, x_{s+1}) - f(x_{s+n-1}, \ldots, x_s)}{x_{s+n} - x_s}$$

Если узлы равноотстоящие, то можно использовать как конечные так и разделенные разности:

$$f(x_{s+1}, x_s) = \frac{f_{s+1} - f_s}{x_{s+1} - x_s} = \frac{f(x_s + h) - f(x_s)}{h} = \frac{\Delta f(x_s)}{h}$$

$$f(x_{s+2}, x_{s+1}, x_s) = \frac{\frac{\Delta f(x_{s+1})}{h} - \frac{\Delta f(x_s)}{h}}{2h} = \frac{\Delta^2 f(x_s)}{2! h^2}$$

По индукции легко показать, что разделенная разность n порядка выглядит так:

$$f(x_{s+n}, x_{s+n-1}, \ldots, x_s) = \frac{\Delta^n f(x_s)}{n! h^n}$$

5 Аппроксимация функций

$f(x)$ может быть представлена различными способами:

1. Аналитически
2. Таблично

3. Графически

4. Алгоритмически (отличается тем, что можем взять $\forall x$)

На практике часто возникает потребность описать исходную функцию, заданную одним из четырех способов, другой, более простой (с точки зрения решаемой задачи), представленной аналитически (например для того, чтобы можно было посчитать интеграл). Аналитически заданную функцию тоже иногда удобно на каком-либо участке заменить на более простую (полином, тригонометрисекую или показательную).

Для сравнения функций друг с другом необходимо ввести критерий близости между функциями.

Рассмотрим исходную функцию $f(x), x \in [a,b]$ и $g(x)$ — аппроксимирующую функцию. Можно использовать два критерия:

1. $\delta = \max\limits_{x\in[a,b]} |f(x) - g(x)|$

2. $\rho^2 = \int_a^b (f(x) - g(x))^2 \,\mathrm{d}x$ — среднеквадратичный критерий.
   Вместо 2 можно использовать любую четную степень или модуль(но это неудобно, так как модуль не является гладкой функцией).

Если функция задана таблично, то дискретный аналог среднеквадратичного критерия выглядит так:

$$\rho^2 = \sum_{k=1}^{n} (f(x_k) - g(x_k))^2$$

Выбор критерия зависит от характера решаемой задачи.

Задача интерполирования

Пусть дана исходная функция:

x	$f(x)$
x_0	f_0
x_1	f_1
$\vdots$	$\vdots$
x_m	f_m

Рассмотрим набор простых, линейнонезависимых функций $\{\varphi_k(x)\}$ и их линейную комбинацию $Q_m(x) = \sum_{k=0}^{m} a_k \varphi_k(x)$. Базисные функции $\varphi_k(x)$ заданы, a_k надо найти, т.е. подобрать их так, чтобы аппроксимирующая функция была ближе.$\varphi_k(x)$ задается исходя из вида исходной функции.

Потребуем чтобы во всех узлах таблицы исходная функция совпадала с аппроксимирующей:

$$Q_m(x_i) = f(x_i) \quad \forall i \in [0,m] \tag{9}$$

Это — $m+1$ уравнение с $m+1$ неизвестными.

$$\triangle = \begin{vmatrix} \varphi_0(x_0) & \varphi_1(x_0) & \dots & \varphi_m(x_0) \\ \vdots & & & \vdots \\ \varphi_0(x_m) & \varphi_1(x_m) & \dots & \varphi_m(x_m) \end{vmatrix} \neq 0$$

Этот определитель не должен быть равен 0 для того чтобы задача имела единственное решение.

Наиболее часто в качестве аппроксимирующих функций выбирают многочлены, т.е. $\varphi_k(x) = x^k$. В этом случае опредедитель системы принимает следующий вид:

$$\triangle = \begin{vmatrix} 1 & x_0 & x_0^2 & \dots & x_0^m \\ \vdots & & & & \vdots \\ 1 & x_m & x_m^2 & \dots & x_m^m \end{vmatrix} \neq 0 \tag{10}$$

(10) называется определителем Ван-дер-Монда. Он не равен 0, если узлы не одинаковы, поэтому система всегда имеет единственное решение.

6 Интерполяционный полином Лагранжа

Если условие (9) выполняется, то $Q_m(x)$ называется интерполяционным многочленом, а x_k — узлами интерполирования.

Критерий близости в данном случае — равенство в узлах интерполирования.

Если узлы расположены близко друг к другу, то определитель оказывается близким к 0 и решение системы становится очень чувствительным к погрешности в исходных данных.

Систему (9) можно не решать и построить интерполяцонный полином в готовом виде. Введем несколько функций:

$$\omega(x) = (x - x_0)(x - x_1)\cdots(x - x_m) \text{ — многочлен степени } m+1$$

Его корни $(x_0, x_1, \dots, x_m)$ — узлы интерполирования.

$$\omega_k(x) = \frac{\omega(x)}{x - x_k} \text{ — многочлен степени } m$$

$$Q_m(x) = \sum_{k=0}^{m} \frac{\omega_k(x)}{\omega_k(x_k)} f(x_k) \tag{11}$$

Причем условия (9) выполняются, т.к.:

$$Q_m(x_i) = \left[\omega_k(x_i) = \begin{cases} 0 & k \neq i \\ \omega_i(x_i) & k = i \end{cases} \right] = f(x_i)$$

Т.е. в $m+1$ узле полином равен значению функции, следовательно это — интерполяционный полином по определению. (11) называется интерполяционным полиномом в форме Лагранжа, т.к. из (10) следует, что все интерполяционные полиномы равны.

Остаточный член интерполяционного полинома

$f(x) = Q_m(x) + R_m(x)$, где $R_m(x)$ — остаточный член интерполяционного полинома, т.е. его погрешность.

Введем $\varphi(z) = f(z) - Q_m(z) - k\omega(x)$. В узлах интерполирования $\varphi(z) = 0$,

т.е. $x_0, x_1, \ldots, x_m$ — нули этой функции. Пусть x — точка, где надо оценить погрешность. Выберем k таким образом, чтобы $\varphi(x) = 0$, т.е:

$$k = \frac{f(x) - Q_m(x)}{\omega(x)} \tag{12}$$

Тогда $\varphi(z)$ будет иметь по крайней мере $m+2$ нуля. По теореме Ролля первая производная будет иметь по крайней мере $m+1$ нуль, вторая — m, $(m+1)$ производная будет иметь 1 нуль. Обозначим эту точку на η:

$$\varphi^{(m+1)}(\eta) = 0$$

Расположение точки η зависит от вида $f(x)$, расположения узлов интерполирования, а также от точки x, где оценивается погрешность.

Возьмем $(m+1)$ производную от φ: $\varphi^{(m+1)}(\eta) = f^{(m+1)}(\eta) - 0 - k(m+1)!$ Следовательно

$$k = \frac{f^{(m+1)}(\eta)}{(m+1)!}$$

Подставляя в (12) получим:

$$f(x) = Q_m(x) + \underbrace{\frac{\omega(x)}{(m+1)!} f^{(m+1)}(\eta)}_{R_m(x)}$$

7 Выбор узлов интерполирования

$R_m(x) = \frac{\omega(x)}{(m+1)!} f^{(m+1)}(\eta).\omega(x)$ в узлах интерполирования обращается в 0. Если f — полином степени m, то погрешность равна 0, т.е. любой полином степени m воспроизводится интерполяционным полиномом по $m+1$ точке точно.

Рассмотрим задачу минимизации погрешности: а η, f и $(m+1)!$ влиять не можем, поэтому минимизируем $|\omega(x)|$:

1. m — const и есть таблица с числом точек $\geq (m+1)$. Точка x^*, в которой вычисляем значение $Q_m(x)$ известна заранее. Тогда для $\min|\omega(x)|$ будем брать $m+1$ ближайших узлов.

2. m — const, таблицы нет, но задан промежуток интерполирования $[a, b]$. Задача состоит в том, чтобы выбрать $m+1$ узел так, чтобы минимизировать максимуму погрешности: $\max\limits_{x \in [a,b]} |\omega(x)| \to \min$. Если x^* неизвестно, то узлы не следует выбирать равномерно, т.к. в этом случае высота "пиков"функции $\omega(x)$ не будет одинаковой в любом случае. Для оптимального выбора узлы надо сдвинуть к краям, т.е. оптимальный выбор узлов отвечает нулям ортогональных полиномов Чебышёва.

Как оценить погрешность интерполяционного полинома? На практике её редко оценивают по остаточному члену в связи с трудностями по оценке производной.
Наиболее популярна следующая процедура: по таблице строят $Q_m(x)$ и вычисляют значение в точке x^*. Потом добавляют ещё одну точку и вычисляют

$Q_{m+1}(x^*)$. Продолжают до тех пор, пока разность между соседними полиномами не будет меньше заданной. Интерполяционный полином Лагранжа мало пригоден для этой процедуры, т.к. при добавлении ещё одной точки необходимо производить все вычисления заново. Поэтому вводят интерполяционный полином в форме Ньютона.

Интерполяционный полином Ньютона для равно- и неравноотстоящих узлов

$x_0, x_1, \ldots, x_m$ — узлы. В точке x надо вычислить значение полинома.

$$f(x, x_0) \stackrel{\text{def}}{=} \frac{f(x) - f(x_0)}{x - x_0} \text{ — разделенная разность первого порядка}$$

Тогда $f(x) = \underbrace{f(x_0)}_{\text{и.п 0 степени}} + \underbrace{(x - x_0) f(x, x_0)}_{\text{погрешность}}$

$$f(x, x_0, x_1) \stackrel{\text{def}}{=} \frac{f(x, x_0) - f(x_0, x_1)}{x - x_1}$$

Выразим $f(x, x_0)$ через вторую разделенную разность и подставим в $f(x)$:

$$f(x) = \underbrace{\underbrace{f(x_0)}_{Q_0} + (x - x_0) f(x_0, x_1)}_{Q_1} + (x - x_0)(x - x_1) f(x, x_0, x_1)$$

Продолжая аналогично, получим:

$$\begin{aligned} f(x) \quad = \quad & f(x_0) + (x - x_0) f(x_0, x_1) + (x - x_0)(x - x_1) f(x, x_0, x_1) + \cdots + \\ & + (x - x_0)(x - x_1) \cdots (x - x_{m-1}) f(x_0, x_1, \ldots, x_m) + \\ & + (x - x_0) \ldots (x - x_m) f(x, x_0, x_1, \ldots, x_m) \end{aligned} \tag{13}$$

(13) — интерполяционный полином Ньютона в общем виде. Последнее слагаемое — его погрешность.

Легко заметить, что $Q_m(x) = Q_{m-1}(x) + (x - x_0)(x - x_1) \ldots (x - x_{m-1}) \times$ $\times f(x_0, \ldots, x_m)$. Тперь легко провести вышеописанную процедуру для подсчета значения в x: построим Q_0 и посчитаем $Q_0(x)$. Достаточно ли? Найдем $Q_1(x)$ и сравним и т.д. Для каждой степени полинома надо знать очередную конечную разность, т.е. нужно дополнительно произвести одно вычитание и одно деление.

В случае, когда узлы таблицы равноотстоящие, разделеннные разности могут быть выражены через конечные: $x_k = x_0 + h t_k, t_k = 0, 1, 2, 3, \ldots$, т.е. считаем в узлах. $f(x_0, x_1, \ldots, x_m) \stackrel{\text{def}}{=} \frac{\Delta^m f(x_0)}{m! h^m}$. Подставим в (13):

$$Q_m(x) = f(x_0) + \frac{t}{1!} \Delta f(x_0) + \cdots + \frac{t(t-1) \cdots (t-m+1)}{m!} \Delta^m f(x_0) \tag{14}$$

(14) — интерполяционный полином Ньютона для равноотстоящих узлов.

Интерполяционные полиномы высших степеней на практике строятся относительно редко, т.к. они очень чувствительны к погрешности в исходных данных. Поэтому часто разбивают таблицу на отдельные участки, на каждом

из которых строят и.п. относительно невысокой степени. На практике такой подход часто используется, однако, в ряде применений это неудобно, т.к. в точках стыка соседних полиномов функция не дифференцируема. В такой ситуации используют интерполяцию сплайнами.

8 Сплайн-интерполяция

Рассмотрим промежуток $[a, b]$ на котром заданы n точек (не обязательно равноотстоящих). Разобьем на $n-1$ промежутков: $[x_1, x_2], \ldots, [x_{n-1}, x_n]$. На каждом промежутке построим по полиному 3 степени — получим всего $n-1$ полиномов.
$[x_k, x_{k+1}] \to S_k(x) = a_k + b_k(x - x_k) + c_k(x - x_k)^2 + d_k(x - x_k)^3$. Возникшая степень свободы (стороим полином 3 степени по 2 точкам) используется для получения гладкой кривой. Всего получаем $4(n-1)$ параметров. Требуется совпадение внутренних точек соседних полиномов, а также их 1 и 2 производных.

$$S_k(x_{k+1}) = S_{k+1}(x_{k+1}) \tag{15}$$
$$S'_k(x_{k+1}) = S'_{k+1}(x_{k+1}) \tag{16}$$
$$S''_k(x_{k+1}) = S''_{k+1}(x_{k+1}) \tag{17}$$

где $k = 1 \ldots n-2$ — рассматриваем внутренние точки. Получили $3n-6$ уравнений. Ещё нужно n условий совпадения с табличной функцией, поэтому всего имеем $4n-6$ условий. Оставшиеся 2 условия могут быть наложены относительно произвольно, например, для они могут быть использованы для обеспечения условий на концах. Из двух наборов условий выбираем те, что наиболее точно подходят под физику задачи.

Различные условия на границах:

1. $S''_1(x_1) = 0$
   $S''_{n-1}(x_n) = 0$
   Это — естественный сплайн. Это термин из механики: $S'' = 0$ — условие того, что линейка (сплайн) находится в свободном состоянии.

2. По первым четырем точкам строится $Q_3(x)$ — и.п. Лагранжа. Аналогично на другом конце($\tilde{Q}_3(x)$). Условия:
   $Q'''_3(x_1) = S'''_1(x_1)$
   $\tilde{Q}'''_3(x_n) = S'''_{n-1}(x_n)$

Возможно построение сплайнов разных степеней, но сплайны третьей степени легче всего считать и они достаточно "красиво" выглядят, поэтому и получили наибольшую популярность.

Программы SPLINE и SEVAL

Интерфейс программы таков: SPLINE(N,X,F,B,C,D).

N количество точек

X массив узлов

F массив значений функции

B,C,D одномерные массивы, элементами которых являются b_k, c_k и d_k.

a_k не находим: $a_k = f_k$, т.к. S_k в узлах с одной стороны равна a_k, с другой — совпадает со значением функции.

Другая функция вычисляет значение сплайна (Spline Evaluation).
SEVAL(N,U,F,B,C,D)
В точке **U** происходит вычисление значения сплайна

Интерполирование по Эрмиту

При построении интерполяционных полиномов Лагранжа и Ньютона использовались значения функции. Если даны значения производной функции в соответствующих точках, то их можно использовать при построении интерполяционного полинома Эрмита.

Предположим, что функция задана следующей таблицей:

x	$f(x)$	$f'(x)$	$f''(x)$
x_0	f_0	f'_0	$-$
x_1	f_1	f'_1	f''_1
x_2	f_2	$-$	$-$

Степень полинома будет равна 5, т.к. имеем 6 "точек".

$$\begin{cases} H_5(x_k) = f_k & k = 0, 1, 2 \\ H'_5(x_k) = f'_k & k = 0, 1 \\ H''_5(x_k) = f''_k & k = 1 \end{cases}$$

Аналогично интерполяционному полиному Лагранжа полином Эрмита тоже можно записать в готовом виде, не решая систему. Пример — разложение функции в степенной ряд:

$$f(x) = \underbrace{f(x_0) + \frac{x - x_0}{1!} f'(x_0) + \frac{(x - x_0)^2}{2!} f''(x_0)}_{P_2(x)} + \underbrace{\frac{(x - x_0)^3}{3!} f'''(x_0)}_{\text{остаточный член}}$$

$P_2(x)$ — частичная сумма разложения в стапенной ряд. Причем $P_2(x_0) = f(x_0), P'_2(x_0) = f'(x_0), P''_2(x_0) = f''(x_0)$, т.е. $P_2(x)$ имеет один узел интерполирования и совпадает в этой точке со значениями функции, следовательно это интерполяционный полином Эрмита.

Обратная задача интерполирования

Ранее требовалось в заданной точке x^* найти значение таблично заданной функции. Рассмотрим обратную задачу — нахождение x^* по заданной f^*. Это можно осуществить двумя способами:

1. Посчитать: $Q_m(x) = f^*$, однако это удобно при малых степенях Q_m (до 5).
2. Поменять аргумент и функцию местами — построить интерполяционный полином от обратной функции. Это возможно, если функция является строго монотонной. Если это не так, то разбиваем на части область определения функции.

9 Квадратурные формулы

$I = \int_a^b f(x)\,\mathrm{d}x$ и $f(x) = Q_m(x) + R_m(x)$. Тогда:

$$\int_a^b f(x)\,\mathrm{d}x \approx \int_a^b Q_m(x)\,\mathrm{d}x \text{ — квадратурная формула}$$

$$\varepsilon = \int_a^b R_m(x)\,\mathrm{d}x \text{ — погрешность квадратурной формулы}$$

Рассмотрим $Q_k(x)$ при различных k:

- $k = 0 \quad Q_0(x) = f(x_0)$, а $x_0 = \begin{cases} a \\ b \\ \frac{a+b}{2} \end{cases}$

$$I \approx (b-a)f(a) \tag{18}$$
$$I \approx (b-a)f(b) \tag{19}$$
$$I \approx (b-a)f\left(\frac{a+b}{2}\right) \tag{20}$$

  Где (18) — квадратурная формула левых, (19) — правых, (20) — средних прямоугольников.

- $Q_1(x) = \frac{x-x_0}{x_1-x_0}f_1 + \frac{x-x_1}{x_0-x_1}f_0$, $x_0 = a$, $x_1 = b$

$$I \approx \frac{b-a}{2}\Big(f(a) + f(b)\Big), \text{ квадратурная формула трапеций}$$

- $Q_2(x)$ строим по трем точкам: $x_0 = a, x_2 = b, x_1 = \frac{a+b}{2}$

$$I \approx \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right], \text{ квадратурная формула Симпсона}$$

- Для четырех узлов формула носит название формулы Ньютона 3/8.

Если результат получается неудовлетворительным, то следует разбить исходный промежуток на ряд промежутков так, чтобы на каждом из них функция описывалась полиномом заданной степени. Затем применяется одна из квадратурных формул и результаты складываются. Такие формулы называются составными.

Составные квадратурные формулы

Разделим $[a, b]$ на n равных промежутков: $[x_0, x_1], [x_1, x_2], \ldots, [x_{n-1}, x_n]$. $x_0 = a, x_n = b, x_{k+1} - x_k = \frac{b-a}{n}$. На каждом промежутке применяем формулу

и складываем результат:

$$I \approx \frac{b-a}{n} \sum_{k=0}^{n-1} f(x_k) \tag{21}$$

$$I \approx \frac{b-a}{n} \sum_{k=0}^{n-1} f(x_{k+1}) \tag{22}$$

$$I \approx \frac{b-a}{n} \sum_{k=0}^{n-1} f\left(x_k + \frac{b-a}{2n}\right) \tag{23}$$

$$I \approx \frac{b-a}{2n} \left(2\sum_{k=1}^{n} f(x_k) + f(a) + f(b)\right) \tag{24}$$

Для составной формулы Симпсона n удобно взять четным:$[x_0, x_2], \ldots,$ $[x_{n-2}, x_n]$. Тогда $x_{k+2} - x_k = \frac{2(b-a)}{n}$ и составная формула Симпсона приобретает следующий вид:

$$I \approx \frac{b-a}{3n} \left(f(a) + \sum_{k=1}^{n/2} f(x_{2k-1}) + \sum_{k=1}^{(n-2)/2} f(x_{2k}) + f(b)\right)$$

Погрешности квадратурных формул

$\varepsilon = \int_a^b R_m(x)\,\mathrm{d}x$. Погрешность будем искать, используя полином Эрмита. Остаточный член и.п. Эрмита нулевой степени для 1 узла выглядит так: $R_0(x) = \frac{x-x_0}{1!} f'(\eta)$. Погрешности формул прямоугольников:

$$\varepsilon_{\text{лпр}} = \int_a^b (x-a) f'(\eta)\,\mathrm{d}x = \frac{(b-a)^2}{2} f'(\eta^*) \tag{25}$$

$$\varepsilon_{\text{ппр}} = \int_a^b (x-b) f'(\eta)\,\mathrm{d}x = -\frac{(b-a)^2}{2} f'(\eta^*) \tag{26}$$

$$\varepsilon_{\text{спр}} = \int_a^b (x - \frac{a+b}{2}) f'(\eta)\,\mathrm{d}x = ? \tag{27}$$

При вычислении используется теорема о среднем: $\exists c \in [a,b] : \int_a^b f(x)g(x)\,\mathrm{d}x = f(c)\int_a^b g(x)\,\mathrm{d}x$, если $g(x)$ знакопостоянна на $[a,b]$. Из-за того, что $\left(x - \frac{a+b}{2}\right)$ меняет знак при переходе через точку $\frac{a+b}{2}$ итеграл (27) по этой теореме посчитать нельзя. Построим интерполяционный полином Эрмита 1 степени:

$$f(x) = f\left(\frac{a+b}{2}\right) + \frac{\left(x - \frac{a+b}{2}\right)}{1!} f'\left(\frac{a+b}{2}\right) + \frac{\left(x - \frac{a+b}{2}\right)^2}{2!} f''(\eta)$$

Тогда при интергировании на $[a,b]$ получим:

$$\underbrace{\int_a^b f(x)\,\mathrm{d}x}_{\text{точное значение}} = \underbrace{(b-a) f\left(\frac{a+b}{2}\right)}_{\text{формула средних прямоугольников}} + 0 + \underbrace{\varepsilon}_{\text{погрешность}}$$

Где $\varepsilon = \int_a^b \frac{\left(x-\frac{a+b}{2}\right)^2}{2!} f''(\eta)\,\mathrm{d}x$. Тогда, по теореме о среднем:

$$\varepsilon_{\text{спр}} = \frac{(b-a)^3}{24} f''(\eta^*)$$

Для формумлы трапеций используется остаточный член и.п. Эрмита, построенного по двум узлам:

$$R_1(x) = \frac{(x-a)(x-b)}{2} f''(\eta)$$

$$\varepsilon_{\text{тр}} = \int_a^b \frac{(x-a)(x-b)}{2} f''(\eta)\,\mathrm{d}x = -\frac{(b-a)^3}{12} f''(\eta^*)$$

То есть формула средних прямоугольников в среднем в два раза точнее формулы трапеций.

Посчитаем погрешность формулы Симпсона:

$$\varepsilon_{\text{Симпсона}} = \int_a^b \frac{(x-a)(x-\frac{a+b}{2})(x-b)}{3!} f'''(\eta)\,\mathrm{d}x = ?$$

Теоремой о среднем воспользоваться нельзя, т.к. функция $x - \frac{a+b}{2}$ не знакопостоянна на $[a,b]$. Аналогично, использовав полином Эрмита 3 степени, получим:

$$\varepsilon_{\text{Симпсона}} = -\frac{(b-a)^5}{2880} f^{(4)}(\eta^*)$$

Погрешности составных квадратурных формул

Рассмотрим разбиение $[x_0,x_1],[x_1,x_2],\ldots,[x_{n-1},x_n]$. Длина каждого промежутка равна $\frac{b-a}{n}$. Тогда для промежутка $[x_k,x_{k+1}]$ остаточный член интерполяционного полинома Эрмита равен $\frac{(x_{k+1}-x_k)^2}{2} f'(\eta_k^*) = \frac{(b-a)2}{2n^2} f'(\eta_k^*)$ (для формулы левых прямоугольников). Общая погрешность равна сумме погрешностей на каждом промежутке:

$$\varepsilon_{\text{лпр}} = \sum_{k=1}^{n} \frac{(b-a)^2}{2n^2} f'(\eta_k^*) = \frac{(b-a)^2}{2n} \frac{1}{n} \sum_{k=1}^{n} f'(\eta_k^*) = \frac{(b-a)^2}{2n} f'(\eta_k^{**})$$

При этом использовался тот факт, что $\exists \eta \in [x_1,x_n] : \frac{1}{n}\sum_{k=1}^{n} f(x_k) = f(\eta)$, т.е. найдется точка, где значение функции равно среднему арифмктическому её значений во всех узлах (дискретный аналог теоремы о среднем).

Получилось, что $\varepsilon \sim \frac{1}{n}$. При этом выбором n можно обеспечить нужную точность. Недостаток заключается в том, что для получения ещё одного верного разряда надо увеличить n в 10 раз. Аналогично получаем для формулы правых прямоугольников:

$$\varepsilon_{\text{ппр}} = -\frac{(b-a)^2}{2n} f'(\eta_k^{**})$$

Для формулы средних прямоугольников:

$$\varepsilon_{\text{спр}} = \sum_{k=1}^{n} \frac{(b-a)^3}{24n^3} f''(\eta_k^*) = \frac{(b-a)^3}{24n^2} f''(\eta_k^{**}) \sim \frac{1}{n^2},$$ что значительно лучше

Погрешность формулы трапеций:

$$\varepsilon_{\text{тр}} = -\frac{(b-a)^3}{12n^2} f''(\eta^{**})$$

Для формуля Симпсона имеем на в 2 раза меньшее число промежутков, поэтому длина будет равна $2\frac{b-a}{n}$. Тогда погрешность составит:

$$\varepsilon_{\text{Симпсона}} = \sum_{k=1}^{n/2} -\frac{\left(\frac{2(b-a)}{n}\right)^5}{2880} f^{(4)}(\eta^*) = -\frac{(b-a)^5}{90n^2} \sum_{k=1}^{n/2} f^{(4)}(\eta_k^*) =$$

$$-\frac{(b-a)^5}{180n^4} \frac{1}{n/2} \sum_{k=1}^{n/2} f^{(4)}(\eta_k^*) =$$

$$-\frac{(b-a)^5}{180n^4} f^{(4)}(\eta^{**}) \sim \frac{1}{n^4} \text{ — вообще прекрасно}$$

Вообще, любая погрешность может быть представлена в следующем виде:

$$\varepsilon_{\forall} = \alpha \frac{(b-a)^{p+1}}{n^p} f^{(p)}(\eta)$$

Для оценки погрешности можно:

1. оценить погрешность по выведенной формуле (редко используется, т.к. надо считать производную высшего порядка в неизвестной точке η — надо оценивать сверху значение производной).
2. использовать какую-либо составную квадратурную формулу для n и $2n$ и сравнить полученные результаты. n удваивается до тех пор, пока результат не будет получен с необходимой точностью.
3. уменьшать n в два раза и сравнивать с результатом для n в случае, если удваивать n нельзя (например, задана фиксированная таблица).
4. для одного и того же значения n сравнивать результаты двух различных квадратурных формул.

10 Общий подход к построению квадратурных формул

Все простые квадратурные формулы укладываются в следующую формулу:

$$\int_a^b f(x)\,\mathrm{d}x \approx \sum_{k=1}^{s} A_k f(x_k) \tag{28}$$

При $s = 1, 2, 3$ получаем общий вид квадратурных формул прямоугольников, трапеций и Симпсона соответственно. x_k — узлы квадратурной формулы, A_k — веса, поэтому всего имеем $2s$ параметров. Надо выбрать их так, чтобы интеграл в (28) считался как можно более точно.

Выберем A_k и x_k так, чтобы (28) была точной для полинома заданной степени. Если подынтегральная функция хорошо описывается таким полиномом, то и интеграл будет хорошо вычисляться, в противном случае следует использовать составные квадратурные формулы.

- $f(x) = \alpha$ — const. Подставим в (28):

$$\sum_{k=1}^{s} A_k = b - a \tag{29}$$

- $f(x) = \alpha x + \beta = x$

$$\sum_{k=1}^{s} A_k x_k = \frac{b^2 - a^2}{2}$$

- $f(x) = x^n$

$$\sum_{k=1}^{s} A_k x_k^n = \frac{b^{n+1} - a^{n+1}}{n+1}$$

Получили систему относительно A_k и x_k. Далее разные авторы по-разному развивали полученную идею.

Квадратурные формулы Ньютона-Котеса

Эти авторы рассмотрели случай равноотстоящих узлов x_k, при этом потеряв половину параметров, но пробретя следующие преимущества:

1. Легкость программирования
2. Легкость решения линейной относительно A_k системы
3. При использовании составной формулы и удвоении n — знание половины значений x_k (главное достоинство)

При $n = s - 1$ формула точна для полиномов степени $s - 1$. При $s = 1$ получаем формулу прямоугольников, $s = 2$ — трапеций и т.д.

Квадратурные формулы Чебышёва

У него был заказ — исходные данные были с большой погрешностью. Чебышёв положил $A_k = A = const$, тогда из (29) следует, что $A = \frac{b-a}{s}$. $n = s$, т.к. 1 уравнение не надо решать. Относительно x_k система является нелинейной. Надо чтобы имелось вещественное решение на $[a, b]$. Оказалось, что для $s = 1, 2 \ldots, 7, 9$ решение имеется, а для $s = 8$ — нет. Позже Бернштейн доказал, что для $s > 9$ этих формул нет (система не имеет решения).

Квадратурные формулы Гаусса (формулы наивысшей алгебраической степени точности)

Формулы носят такое название, т.к. больше "выжать" нельзя. Гаусс использовал все $2s$ параметров, поэтому полученные формулы верны для полиномов степени $n = 2s - 1$. Формулы Гаусса существуют для любого значения s.

Для всех семейств квадратурных формул найдены узлы x_k и веса A_k для стандартных промежутков ($[0, 1]$ или $[-1, 1]$). Для произвольного промежутка

пользуются следующими формулами: $x = \frac{a+b}{2} + \frac{b-a}{2}t$, где $t \in [-1, 1]$. Тогда при $t = -1$ $x = a$, при $t = 1$ $x = b$. Получаем:

$$I = \int_a^b f(x)\,\mathrm{d}x = \frac{b-a}{2}\int_{-1}^{1} f\left(\frac{a+b}{2} + \frac{b-a}{2}t\right)\,\mathrm{d}t \approx$$
$$\approx \frac{b-a}{2}\sum_{k=1}^{s} A_k f\left(\frac{a+b}{2} + \frac{b-a}{2}t_k\right)$$

11 Адаптивные квадратурные формулы

Предположим, что надо посчитать интеграл от функции, содержащей высокий и узкий "пик". Для его описания надо использовать маленький шаг на всем промежутке. Адаптивные формулы изменяют шаг, адаптируясь в виду функции. Шаг выбирается малым там, где функция изменяется быстро (т.е. обладает большой производной), и большим там, где функция меняется медленно.

Добьемся заданной точности при минимальных затратах.

$$\varepsilon_\forall = \alpha\frac{(b-a)^{p+1}}{n^p} f^{(p)}(\eta)$$

Для квадратурной формулы Ньютона-Котеса с 9 узлами $p = 10$.

Предположим, что на $[a, b]$ имеется промежуток h_i, I_i — точное значение интеграла на этом промежутке, P_i — значение интеграла, посчитанное по квадратурной формуле, Q_i — посчитанное по квадратурной формуле при разбиении на 2 участка (т.е. с исползовнием 17 точек). При построении Q_i число точек $n' = 2n$ числа точек, использованых при построении P_i. Тогда:

$$I_i - P_i \approx 2^p(I_i - Q_i)$$

Равенство приближенное, т.к. средние точки для обоих формул не одинаковы. Выразим I_i:

$$I_i = \frac{2^p Q_i - P_i}{2^p - 1} \quad I_i - Q_i = \frac{Q_i - P_i}{2^p - 1} = \frac{Q_i - P_i}{1023}$$

Промежуток h_i считается принятым (а интеграл на нем — вычисленным), если $\left|\frac{Q_i - P_i}{1023}\right|$ меньше заданной величины. Если пользоваться погрешностью, то неравенство имеет следующий вид:

$$\left|\frac{Q_i - P_i}{1023}\right| < \frac{h_i}{b-a}\varepsilon_A$$

Где ε_A — требование ко всему промежутку, а $\frac{h_i}{b-a}\varepsilon_A$ — требование к одному промежутку. При использовании относительной погрешности формула имеет следующий вид:

$$\left|\frac{Q_i - P_i}{1023}\right| < \frac{h_i}{b-a}\varepsilon_R|Q_i|$$

Единственный недостаток — при вычислении интеграла, заначение которого близко к 0 такой контроль погрешность пиведет к вычилению с максимальной точностью. В программе **QUANC8** используется смешанный контроль

погрешности:

$$\left|\frac{Q_i - P_i}{1023}\right| < \frac{h_i}{b-a}\max\Big\{\varepsilon_R|Q_i|, \varepsilon_A\Big\}$$

При приближении $|Q_i|$ к нулю происходит переключение на контроль абсолютной погрешности.

Вычисление интеграла происходит следующим образом:
Высчитывается P_i и Q_i. Результат сравнивается. Если контроль погрешности не прошел, то значение интеграла на правой половине запоминается и снова считается интеграл на левой половине. Деление продолжается до тех пор, пока крайний слева промежуток не будет принят. Потом обрабатывается ближайший справа промежуток. Такое деление пополам происходит не более 30 раз, после интеграл считается вычисленным, а число таких промежутков располагается в целой части переменной FLAG. Интерйес **QUANC8**: QUANC8(FUN,A,B,EA,ER,RES,EPS,NOFUN,FLAG)

FUN имя интегрируемой функции

A,B пределы интегрирования

EA,EB абсолютная и относительная погрешности

EPS оценка точности, выполненная программно

NOFUN количество вычислений подынтегральной функции

FLAG целая часть: количество промежутков, принятых с нарушением контроля погрешности. Дробная чать: номер промежутка, на котором "застряли".

12 Задача численного дифференцирования

Рассмотрим задачу нахождения поизводной функции:

x	$f(x)$	$f'(x)$
x_0	f_0	?
x_1	f_1	?
$\vdots$	$\vdots$	$\vdots$
x_m	f_m	?

Если две функции почти совпадают, то об их производных такого сказать нельзя, например, рассмотрим $f(x)$ и $g(x) = f(x) + \frac{1}{n}\sin(n^2x)$. Тогда $g'(x) = f'(x) + n\cos(n^2x)$, т.е. при $n \to \infty$ функция $g(x) \to f(x)$, а её производная – нет. Поэтому надеемся на то, что функция не зашумлена — является относительно гладкой и задана с не слишком большой погрешностью. Из-за этого уже вторую производную функции считают очень редко.

Общая идея метода — дифференцирование интерполяционного полинома:

$$f'(x) \approx Q'_m(x) \qquad \varepsilon = R'_m(x)$$

Начнем с Q_1 и рассмотрим случай равноотстоящих узлов:

$$Q_1(x) = \frac{x - x_1}{x_0 - x_1}f_0 + \frac{x - x_0}{x_1 - x_0}f_1 \qquad x_k = x_0 + hk$$

$Q_1'(x) = const$, поэтому получаем:

$$f'(x_0) \approx \frac{f_1 - f_0}{h} \tag{30}$$

$$f'(x_1) \approx \frac{f_1 - f_0}{h} \tag{31}$$

Это — разные формулы, т.к. в одном случае берется следующая точка, в другом — предыдущая.

Погрешность посчитаем с помощью остаточного члена полинома Эрмита:

$$R_1(x) = \frac{(x - x_0)(x - x_1)}{2!} f''(\eta)$$

Получается, что погрешность формул (30) и (31) одинакова:

$$R_1'(x_0) = -\frac{h}{2} f''(\eta)$$

$$R_1'(x_1) = \frac{h}{2} f''(\eta)$$

Для $Q_2(x)$ получим:

$$f'(x_0) = \frac{1}{2h}\Big(-3f(x_0) + 4f(x_1) - f(x_2)\Big)$$

$$f'(x_1) = \frac{1}{2h}\Big(f(x_2) - f(x_0)\Big)$$

$$f'(x_2) = \frac{1}{2h}\Big(f(x_0) - 4f(x_1) + 3f(x_2)\Big)$$

Погрешность (из $R_2(x) = \frac{(x-x0)(x-x_1)(x-x_2)}{3!} f'''(\eta)$):

$$R_2'(x_0) = \frac{h^2}{3} f'''(\eta) \tag{32}$$

$$R_2'(x_1) = -\frac{h^2}{6} f'''(\eta) \tag{33}$$

$$R_2'(x_2) = \frac{h^2}{3} f'''(\eta) \tag{34}$$

Получается, что (33) в два раза точнее чем две другие. Однако, их приходится использовать в крайних точках (в начале и в конце).

Простейшая формула для втрой производной:

$$f''(x_1) \approx \frac{f_0 - 2f_1 + f_2}{h^2} = \frac{\Delta^2 f(x_0)}{h^2}$$

Влияние вычислительной погрешности на точность

Снова обратимся к простейшей формуле:

$$f'(x_0) \approx \frac{f_1 - f_0}{h} \qquad \varepsilon = -\frac{h}{2} f''(\eta)$$

При проверке этой формулы (построении $\varepsilon(h)$) для $f(x) = \sin(x)$, т.е. построении $\varepsilon = \sin''(x) - \frac{f_1 - f_0}{h}$ (реальной погрешности), получается что при

достаточно малых h переполняется разрядная сетка и погрешность резко увеличивается.

Оптимальный шаг — такое h, при котором эти погрешности равны, т.к.:

$$|\Delta| \leq \left|\frac{h}{2} f''(\eta)\right| + \left|\frac{\varepsilon_1 - \varepsilon_0}{h}\right|$$

13 Среднеквадратичная аппроксимация (дискретный случай)

Рассмотрим некую функцию (например, экспериментальный данные), заданную таблицей.

x	$f(x)$
x_1	f_1
x_2	f_2
$\vdots$	$\vdots$
x_n	f_n

Как узнать, что это — полином или прямая? Аппроксимируем с использованием среднеквадратичного критерия близости (35):

$$Q_m(x) = \sum_{k=0}^{m} a_k \varphi_k(x), \text{ где } \varphi_k \text{ — заданный набор ЛНЗ функций}$$

$$\rho^2 = \sum_{i=1}^{n} (Q_m(x_i) - f(x_i))^2 \tag{35}$$

Будем выбирать a_k так, чтобы величина ρ^2 была минимальной, тогда получим систему уравнений:

$$\frac{\partial \rho^2}{\partial a_k} = 0 \qquad k = 0, 1, 2, \ldots, m$$

Тут $m+1$ коэффициентов и n экспериментальных точек. Возможны следующие варианты:

1. $n = m + 1$. Решением задачи является интерполяционный полином, решение единственное, $\min(\rho^2) = 0$ в узлах.
2. $n < m + 1$. $\min(\rho^2) = 0$, решений бесконечно много.
3. $n > m + 1$. Очень часто бывает, что $n \gg m + 1$, тогда снова получаем единственное решение, но $\min(\rho^2) \neq 0$ в общем случае.

Часто важно учесть то, что различные точки измерены с различной степенью точности (использовался различный диапазон, например), поэтому удобно ввести весовые коэффициенты:

$$\rho^2 = \sum_{i=1}^{n} P_i (Q_m(x_i) - f(x_i))^2, \quad P_i > 0$$

Те точки, к которым доверия больше, имеют большее значение P_i

Найдем a_k при которых ρ^2 минимально:

$$\frac{\partial \rho^2}{\partial a_k} = 0 = 2\sum_{i=1}^{n} P_i(Q_m(x_i) - f(x_i)) \cdot \varphi_k(x_i)$$

Т.к. $Q_m(x) = \sum_{k=0}^{m} a_k\varphi_k(x)$, то:

$$a_0\sum_{i=1}^{n} P_i\varphi_0(x_i)\varphi_k(x_i) + \cdots + a_m\sum_{i=1}^{n} P_i\varphi_m(x_i)\varphi_k(x_i) = \sum_{i=1}^{n} P_i f(x_i)\varphi_k(x_i)$$

При $k = 0, 1, \ldots, m$ получаем $m+1$ уравнение с $m+1$ неизвестным.

14 Среднеквадратичная аппроксимация (непрерывный случай)

Рассмотрим непрерывный случай:

$$f(x), \quad x \in [a,b] \quad Q_m(x) = \sum_{k=0}^{m} a_k\varphi_k(x) \quad \rho^2 = \int_a^b p(x)(Q_m(x) - f(x))^2$$

Где ρ^2 — функця штрафа, если все точки эквивалентны по достоверности, то $\rho^2 \equiv 0$. Задача прежняя — минимизация функции штрафа. Необходимое условие экстремума:

$$\frac{\partial \rho^2}{\partial a_k} = 0 = 2\int_a^b p(x)\Big(Q_m(x) - f(x)\Big)\varphi_k(x)\,\mathrm{d}x$$

Для $k = 0, 1, \ldots, m$ получаем $m+1$ уравнение с $m+1$ неизвестным:

$$\begin{aligned} a_0\int_a^b p(x)\varphi_0(x)\varphi_k(x)\,\mathrm{d}x + \cdots + a_m\int_a^b p(x)\varphi_m(x)\varphi_k(x)\,\mathrm{d}x = \\ = \int_a^b p(x)f(x)\varphi_k(x)\,\mathrm{d}x \end{aligned} \tag{36}$$

Но можно получить $a_0, a_1, \ldots, a_m$ не решая систему, если функции $\varphi_k(x)$ являютя ортогональными.

Последовательность функций $\{\varphi_k(x)\}$ называется ортогональной на $[a,b]$ с весом $p(x)$ если:

$$\int_a^b p(x)\varphi_k(x)\varphi_i(x)\,\mathrm{d}x = \begin{cases} 0 & k \neq i \\ A > 0 & k = i \end{cases}$$

Ортогональная последовательность называется нормированной, если $A = 1$. Тогда, если $\{\varphi_k(x)\}$ ортонормирована, то в левой части (36) все коэффициенты кроме одного обращаются в 0. Тогда легко найти a_k:

$$a_k = \frac{\int_a^b p(x)f(x)\varphi_k(x)\,\mathrm{d}x}{\int_a^b p(x)\varphi_k^2(x)\,\mathrm{d}x}$$

Рассмотрим промежуток $[0,1]$ и $\varphi_k(x) = x^k = 1, x, x^2, x^3, \ldots$. Эта система не является ортогональной, но её можно сделать таковой, испольую процедуру ортогонализации Грамма-Шмидта.

15 Ортогонализация по Шмидту

Зафиксируем $[a, b]$ и $p(x)$. Рассмотрим систему ЛНЗ функций $\{\varphi_k(x)\}$, где $k \in 0 : m$. Надо получить систему ортогональных функций $g_0(x), \ldots, g_m(x)$, которые бы являлись линейной комбинацией функций исходной системы.

На k шаге $g_k(x)$ должна быть построена так, чтобы быть ортогональной всем функциям, построенным до неё. $\tilde{g}_k(x)$ — ортогональные, но не нормированные функции.

Возьмем $\tilde{g}_0(x) = \varphi_0(x)$ — уже ортогональная. Нормируем:

$$\int_a^b p(x) g_0^2(x)\,\mathrm{d}x = \alpha_0^2 > 0$$

Теперь возьмем $g_0(x) = \frac{\tilde{g}_0(x)}{\alpha_0}$.

По индукции легко показать, что на m шаге имеем:

$$\tilde{g}_m(x) = \varphi_m(x) - \sum_{k=1}^{m-1} C_{mk} g_k(x)$$

Домножим на $g_i(x) \cdot p(x)$ для получения ортогональности и проинтегрируем:

$$\int_a^b p(x)\tilde{g}_m(x) g_i(x)\,\mathrm{d}x \overset{\text{д.б.}}{=} 0 = \int_a^b p(x)\varphi_m(x) g_i(x)\,\mathrm{d}x - C_{mi}$$

C_{mi} — всё, что останется от суммы (при $k = i$). Выразим его:

$$C_{mi} = \int_a^b p(x)\varphi_m(x) g_i(x)\,\mathrm{d}x$$

Теперь необходимо нормировать:

$$\int_a^b p(x)\tilde{g}_m^2(x)\,\mathrm{d}x = \alpha_m^2 \text{ и } g_m = \frac{\tilde{g}_m(x)}{\alpha_m}$$

Примеры ортогональных полиномов

Характеристика полинома — $p(x)$. Промежуток можно выбрать любой с помощью замены переменной.

1. Ортогональные полиномы Лежандра: $[-1, 1]$, $p(x) \equiv 1$. Для них справедливо следующие реккурентное соотношение:

$$(m+1)L_{m+1}(x) - (2m+1)xL_m(x) + mL_{m-1}(x) = 0$$
$$L_0(x) = 1$$
$$L_1(x) = x$$

   При $m = 1$ получаем $L_2(x) = \frac{3x^2-1}{2}$. Все чётные полиномы содержат четные степени x, нечётные — соответственно нечетные.

   Эти полиномы являются ортогональными, но не номированными.

2. Ортогональные полиномы Чебышёва: $[-1,1]$, а $p(x)=\frac{1}{\sqrt{1-x^2}}$, т.е. крайние точки бесконечно "надежны", а середина имеет $p(0)=1$. Полином Чебышёва выглядит так:

$$T_m(x)=\cos(m\arccos(x))$$
$$T_0(x)=1$$
$$T_1(x)=x$$
$$T_2(x)=?$$

Получим рекуррентное соотношение для полиномов Чебышёва:

$$\cos(m+1)\varphi+\cos(m-1)\varphi=2\cos\varphi\cos m\varphi$$

Подставим $\varphi=\arccos(x)$:

$T_{m+1}(x)+T_{m-1}(x)=2xT_m(x)$ — снова разностное уравнение 2 порядка

Видно, что дальше ничего кроме полиномов ничего не получим:

$$T_{m+1}(x)=2xT_m(x)-T_{m-1}(x)$$

Далее получается аналогично полиномам Лежандра — полиномы чётной степени содержат чётные степени x:

$$T_2(x)=2x^2-1$$
$$T_3(x)=4x^3-3x$$

Если построить $|T_m(x)|$ на $[-1,1]$ то видно, что оптимальным выбором узлов интерполирования является выбор в нулях полинома Чебышёва.

16 Матрицы. Собственные значения и собственные векторы

Матрицей называется таблица $m\times n$:

$$A=\begin{pmatrix}a_{11} & a_{12} & \dots & a_{1n}\\ a_{21} & a_{22} & \dots & a_{2n}\\ \vdots & \vdots & \ddots & \vdots\\ a_{m1} & a_{m2} & \dots & a_{mn}\end{pmatrix}$$

Матрица $X=A^{-1}$ (т.е. является обратной), если $AX=E$, где E — единичная матрица.

Собственные значения и собственные векторы

λ и X, удовлетворяющие следующему уравнению, называются собственными значениями и собственными векторами матрицы A:

$$Ax=\lambda x$$

Нулевые собственные векторы нас не интересуют (при $|A-\lambda E|\neq 0$), находим λ и X и характеристического уравнения:

$$|A-\lambda E|=0$$

Задачи на матрицы

1. $(A \cdot B)^T = B^T \cdot A^T$
2. $(AB)^{-1} = B^{-1}A^{-1}$
3. $(A^T)^{-1} = (A^{-1})^T$
4. Умножение диагональной матрицы слева на матрицу A даёт умножение каждой строки A на соответствующий диагональный элемент диагональной матрицы.
5. Умножение диагональной матрицы справа на матрицу A даёт умножение каждого столбца A на соответствующий диагональный элемент диагональной матрицы.
6. При перемножении левых или правых треугольных матриц получается матрица того же вида.
7. Обратная матрица для треугольной есть треугольная матрица того же вида.
8. Собственные значения диагональной матрицы равны её диагональным элементам.
9. Собственные значения треугольной матрицы также равны её диагональным элементам.
10. Сумма собственных значений равна сумме диагональных элементов, а произведение собственных значений равно определителю матрицы.

Норма матрицы

Нормой матрицы называется число, удовлетворяющее следующим аксиомам:

1. $||A|| \geq 0$ (т.к. $||A|| = 0 \leftrightarrow A = 0$)
2. $||\alpha A|| = |\alpha| \cdot ||A||$
3. $||A + B|| \leq ||A|| + ||B||$
4. $||AB|| \leq ||A|| \cdot ||B||$

Норма называется канонической, если дополнительно выполняются следующие условия:

1. $||A|| \geq |a_{ij}|$
2. $|a_{ij}| > |b_{ij}|$, то $||A|| > ||B||$

Наиболее популярными являются следующие канонический формы:

1. $||A||_I = \max_i \sum k = 1^n |a_{ik}|$
2. $||A||_{II} = \max_i \sum k = 1^n |a_{ki}|$
3. $||A||_{III} = \sqrt{\sum_{i=1}^n \sum_{k=1}^n |a_{ik}|^2}$

Отношение между нормами зависит от конкретной матрицы и в общем случае сказать, что какая-то норма больше другой нельзя.

Матричный ряд и его сходимость

$P_m(A) = \alpha_0 E + \alpha_1 A + \cdots + \alpha_m A^m = \sum_{k=0}^{m} \alpha_k A^k$ (где α_k — скалярные величины). При $m \to \infty$ получаем степенной матричный ряд:

$$P(A) = \sum_{k=0}^{\infty} \alpha_k A^k \tag{37}$$

Матричный ряд сходится, если сходятся все n^2 скалярных рядов для элементов матрицы $P(A)$. Уже для $n = 10$ тяжело считать, поэтому находят мажорантный ряд (оценку сверху). Его сходимость является достаточным условием сходимости исходного ряда.

Элемент ряда (37) будем обозначать так:

$$U_{ij}^{(k)} = \alpha_k A_{ij}^k$$

Тогда о сходимости ряда (37) можно судить по $||A||$:

$$|P_{ij}(A)| = |\sum_{k=0}^{\infty} U_{ij}^{(k)}| \le \sum_{k=0}^{\infty} |U_{ij}^{(k)}| \overset{\text{доп.A1}}{\le} \sum_{k=0}^{\infty} ||\alpha_k A^k|| \overset{\text{A2}}{=}$$
$$\overset{\text{A2}}{=} \sum_{k=0}^{\infty} |\alpha_k| ||A^k|| \overset{\text{A4}}{\le} \sum_{k=0}^{\infty} |\alpha_k| ||A||^k$$

Т.е. достаточный ряд сходится при $||A|| < R$. Если матричный ряд сходится, то его сумма является матричной функией:

$$e^A = \sum_{k=0}^{\infty} \frac{A^k}{k!} \quad R = \infty$$
$$\cos(A) = \sum_{k=0}^{\infty} \frac{(-1)^k A^{2k}}{(2k)!} \quad R = \infty$$
$$(E - A)^{-1} = \sum_{k=0}^{\infty} A^k \quad R = 1 \ \to ||A|| \overset{\text{д.б.}}{<} 1 \text{ для сходимости}$$

17 7 теорем о матричных функциях

Матрицы A и B называются подобными, если найдется такая неособенная S: $B = SAS^{-}1$.

1. Подобные матрицы имеют одинаковые собственные значения.

$$\det(B - \lambda E) = \det(SAS^{-1} - \lambda SS^{-1}) = \det(S(A - \lambda E)S^{-1}) =$$
$$= \det S \det(A - \lambda E) \det S^{-1} = det(A - \lambda E)$$

2. Матричные функции подобных матриц подобны, т.е. если $B = SAS^{-1}$, то $f(B) = Sf(A)S^{-1}$.

$$f(B) = \sum_{k=0}^{\infty} \alpha_k B^k = \sum_{k=0}^{\infty} SA^k S^{-1} = S \left(\sum_{k=0}^{\infty} \alpha_k A^k \right) S^{-1} = Sf(A)S^{-1}$$

3. Если собственные значения матрицы A различны, то она подобна диагональной матрице, на диагонале которой стоят собственные значения этой матрицы.
   Матрица A имеет собственные значения λ_k и собственные вектора U_k. Тогда $U = (\,{}_{U_1}\ {}_{U_2}\ ...{}_{U_n}\,)$ — матрица собственных значений.

$$\Lambda = \begin{pmatrix} \lambda_1 & & & & 0 \\ & \lambda_2 & & & \\ & & \lambda_3 & & \\ & & & \ddots & \\ 0 & & & & \lambda_n \end{pmatrix}$$

   Т.к. $A \cdot U_k \overset{\text{def}}{=} \lambda_k U_k$, то $A \cdot U = (\,{}_{\lambda_1 U_1}\ {}_{\lambda_2 U_2}\ ...\ {}_{\lambda_n U_n}\,)$. Домножая на U^{-1} справа и слева соответственно, получим:

$$A = U\Lambda U^{-1} \text{ — что и требовалось доказать}$$
$$U^{-1}AU = \Lambda$$

4. Матричные функции от одной и той же матрицы комутируют: $f(A)g(A) = g(A)f(A)$. $A = U\Lambda U^{-1}$. Тогда $f(A) \overset{\text{T2}}{=} Uf(\Lambda)U^{-1}$, $g(A) = Ug(\Lambda)U^{-1}$
   Перемножим f и g:

$$f(A)g(A) = Uf(\Lambda)U^{-1}Ug(\Lambda)U^{-1} = Uf(\Lambda)g(\Lambda)U^{-1}$$

   Произведение диагональных матриц — произведение их диагональных элементов, поэтому поменяем местами $f(\Lambda)$ и $g(\Lambda)$:

$$Uf(\Lambda)g(\Lambda)U^{-1} = Ug(\Lambda)f(\Lambda)U^{-1} = g(A)f(A)$$

5. Если у матрицы A собственные числа λ_k, то у $f(A)$ — $f(\lambda_k)$.
   $A \overset{\text{T3}}{=} U\Lambda U^{-1}$. По T2 имеем $f(A) = Uf(\lambda)U^{-1}$, где

$$f(\Lambda) = \sum_{k=0}^{\infty} \alpha_k \Lambda^k = \begin{pmatrix} \sum_{k=0}^{\infty} \alpha_k \lambda_1^k & & & 0 \\ & \sum_{k=0}^{\infty} \alpha_k \lambda_2^k & & \\ & & \ddots & \\ 0 & & & \sum_{k=0}^{\infty} \alpha_k \lambda_n^k \end{pmatrix} =$$
$$= \begin{pmatrix} f(\lambda_1) & & & 0 \\ & f(\lambda_2) & & \\ & & \ddots & \\ 0 & & & f(\lambda_n) \end{pmatrix}$$

   Тогда $f(A) = U(\,{}^{f(\lambda 1)}_{0}\ {}^{0}_{f(\lambda_n)}\,)U^{-1}$ и по T1 они имеют одинаковые собственные значения.

   Из этой теоремы следует:

   а Матричная функция $f(A)$ существует тогда и только тогда, когда существуют скалярные ряды на диагонале матрицы $f(\lambda)$. А эти ряды существуют тогда и только тогда, когда все собственные числа матрицы A меньше R, т.е. $|\lambda_k| < R$. Это необходимое и достаточное условие сходимости матричного ряда.

б

$$\left.\begin{array}{r} ||A|| < R \text{ — достаточное условие} \\ |\lambda_k| < R \text{ — необходимое и достаточное условие} \end{array}\right\} \Rightarrow |\lambda_k| \le ||A||$$

6. Теорема Гамильтона-Кэли:
   Любая матрица удовлетворяет своему характеристическому уравнению.

$$\det(A - \lambda E) = Q(\lambda) = \alpha_0\lambda^n + \alpha_1\lambda^{n-1} + \cdots + \alpha_n = 0$$

   Если вместо λ подставить A, то тоже получим 0 (но уже не скалярный). По Т3 имеем $A = U\Lambda U^{-1}$, по Т2 — $Q(A) = UQ(\lambda)U^{-1}$,

$$Q(\lambda) = \alpha_0\Lambda^n + \cdots + \alpha_n E = \begin{pmatrix} Q(\lambda_1) & & & 0 \\ & Q(\lambda_2) & & \\ & & \ddots & \\ 0 & & & Q(\lambda_n) \end{pmatrix} = 0$$

   Отсюда следут, что $Q(A) = 0$.

7. Формула Лагранжа-Сильвестра.
   Рассмотрим матрицу A и её собственные числа $\lambda_1, \lambda_2, \ldots, \lambda_n$. Для $f(x)$ построим интерполяционный полином Лагранжа ($Q_{n-1}(x)$) с узлами интерполирования в точках, равных собственным значениям матрицы A. Тогда:

$$R_{n-1}(x) = \frac{\omega(x)}{n!} f^{(n)}(\eta), \quad \omega(x) = (x - \lambda_1)(x - \lambda_2)\cdots(x - \lambda_n)$$

   Т.е. $\omega(x)$ — характеристический полином для матрицы A. Тогда, по Т6 $\omega(A) = 0$ и $R_{n-1}(A) = 0$. Получается, что $f(A) = Q_{n-1}(A)$. Выразим $f(A)$:

$$f(A) = \sum_{k=1}^{n} \frac{(A - \lambda_1 E)\cdots(A - \lambda_{k-1}E)(A - \lambda_{k+1}E)\cdots(A - \lambda_n E)}{(\lambda_k - \lambda_1)\cdots(\lambda_k - \lambda_{k-1})(\lambda_k - \lambda_{k+1})\cdots(\lambda_k - \lambda_n)} f(\lambda_k)$$

18 Решение систем линейных дифференциальных с постоянной матрицей

$$\frac{\mathrm{d}x}{\mathrm{d}t} = Ax + f(t), \quad x(0) = x_0 \tag{38}$$

Здесь x — вектор решения, A — постоянная квадратная матрица, $f(t)$ — заданная вектор-функция.

1. $f(t) \equiv 0$, т.е. $x' = Ax$. Тогда, по аналогии ($x(t) = ce^{\alpha t}$ для случая, когда A — скаляр) получаем, что $x(t) = Ce^{At}$. В этом легко убедиться, продифференцировав e^{At}.

2. $f(t) \neq 0$. Тогда решение будем искать в виде

$$x(t) = C(t)e^{At} \tag{39}$$

Подставим его в (38):

$$Ae^{At}C(t) + e^{At}C'(t) = Ae^{At}C(t) + f(t)$$
$$C'(t) = e^{-At}f(t)$$

Подставим полученное выражение в (39):

$$C(t) - C(0) = \int_0^t e^{A(t-\tau)}f(\tau)\,\mathrm{d}\tau$$

Если $t = 0$, то $x_0 = E \cdot C(0)$ и $C(0) = x_0$. Тогда (39) приобретает следующий вид:

$$x(t) = e^{At}x_0 + \int_0^t e^{A(t-\tau)}f(\tau)\,\mathrm{d}\tau$$

Решение систем линейных разностных уравнений с постоянной матрицей

$$x_{n+1} = Bx_n + \varphi(n), \quad x(0) = x_0 \tag{40}$$

Здесь x — вектор решения, B — постоянная квадратная матрица, $\varphi(n)$ — заданная вектор-функция.

Начнем решать пошаговым методом:

$$x_1 = Bx_0 + \varphi_0$$
$$x_2 = Bx_1 + \varphi_1 = Bx^2 + B\varphi_0 + \varphi_1$$
$$x_3 = Bx_2 + \varphi_2 = B^3x_0 + B^2\varphi_0 + B\varphi_1 + \varphi_2$$

По индукции получаем, что:

$$x_n = B^nx_0 + \sum_{k=0}^{n-1} B^k\varphi_{n-k-1} \text{ — точное решение (40)}$$

Рассмотрим важный случай: $\varphi_n = d = const$ Тогда по Т4 имеем:

$$\sum_{k=0}^{n-1} B_k = (E - B^n)(E - B)^{-1}$$

19 Устойчивость решений дифференциальных и разностных уравнений

Рассмотрим систему нелинейных уравнений ($f(t, x)$ — уже не произвольная функция):

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(t, x), \quad x(t_0) = x_0 \tag{41}$$

Будем рассматривать устойчивость решений (41) по отношению к изменениям в начальных условиях. За $x^{(k)}(t)$ обозначим k компоненту вектора x, за $y(t)$ — другое решение (41), отличающееся от $x(t)$ начальными условиями: $y(t_0) = y_0$

Решение $x(t)$ системы (41) называется устойчивым по Ляпунову, если

$$(\forall \varepsilon > 0)(\exists \delta > 0)(\forall y(t))(||x_0 - y_0|| < \delta) \Rightarrow (||x(t) - y(t)|| < \varepsilon)$$

Если t принадлежит конечному промежутку и решение $x(t)$ на этом промежутке ограничено, то это условие выполняется практически всегда, поэтому наибольший интерес представляет поведение решения при $t \to \infty$.

Решение $x(t)$ системы (41) называется ассимптотически устойчивым по Ляпунову, если

$$\lim_{t\to\infty}(y(t) - x(t)) = 0$$

Аналогичные определения можно ввести и для систем разностных уравнений первого порядка: $x_{n+1} = F(n, x_1)$. Все определения будут иметь место, только $t \to n$.

Устойчивость решений СЛДУ с постоянной матрицей

$$\frac{\mathrm{d}x}{\mathrm{d}t} = Ax + f(t) \tag{42}$$

Точное решение (42) имеет вид:

$$x(t) = e^{At}x_0 + \int_0^t e^{A(t-\tau)}f(\tau)\,\mathrm{d}\tau \tag{43}$$

$$y(t) = e^{At}y_0 + \int_0^t e^{A(t-\tau)}f(\tau)\,\mathrm{d}\tau \tag{44}$$

Вычтем из (43) (44):

$$x(t) - y(t) = e^{At}(x_0 - y_0)$$

Для выполнения условия устойчивости элементы матрицы e^{At} при $t \to \infty$ должны быть ограниченными, а для выполнения условия ассимптотической устойчивости эти элементы должны стремиться к нулю, т.е $e^{At} \underset{t\to\infty}{\longrightarrow} 0$.

Рассмотрим ситуацию, когда все собственные значения различны. Тогда:

$$e^{At} \overset{\text{T7}}{=} \sum_{k=1}^{n} T_k e^{\lambda_k t}$$

T_k не имеет отношения к виду функции, поэтому $e^{\lambda_k t}$ определяет поведение при $t \to \infty$. Элементы e^{At} будут ограниченными, если $\Re\lambda_k \leq 0$. Условие ассимптотичекой устойчивости $\Re\lambda_k < 0$.

В случае, если λ_1 имеет кратность s, то в решении появляется составляющая $P_{s-1}(t)e^{\lambda_1 t}$, где P_{s-1} — полином степени $(s-1)$. На ассимптотическую устойчивость это никак не повлияет, а о просто устойчивости такого сказать нельзя.

Таким образом, необходимое и достаточное условие ассимпотической устойчивости выглядит так:

$$\Re\lambda_k < 0 \; \forall k$$

Решение будет устойчиво, если $\Re\lambda_k \leq 0$ и для собственных значений с нулевой вещественной частью нет кратных.

Устойчивость решений СЛРУ с постоянной матрицей

Рассмотрим систему разностностных уравнений:

$$x_{n+1} = Bx_n + \varphi(n),\ x(0) = x_0$$

и её решения:

$$x_n = B^n x_0 + \sum_{k=0}^{n-1} B^k \varphi_{n-k-1} \tag{45}$$

$$y_n = B^n y_0 + \sum_{k=0}^{n-1} B^k \varphi_{n-k-1} \tag{46}$$

Вычтем из (45) (46):

$$x_n - y_n = B^n(x_0 - y_0)$$

Для устойчивости необходимо, чтобы элементы B^n при $n \to \infty$ были бы ограничены, а для ассимптотической устойчивости они должны стремиться к нулю.

$$B^n = \sum k = 1^n T_k \lambda_k^n$$

Необходиомое и достаточное условие ассимптотической устойчивости: $|\lambda_k| < 1\ \forall k$. Для устойчивости надо, чтобы $|\lambda_k| \leq 1$ и для собственных значений с $|\lambda_i| = 1$ не было кратных.

Следует заметить, что из устойчивости ограниченность не следует.

20 Метод Гаусса и явление плохой обусловленности

Рассмотрим линейную систему

$$Ax = B \tag{47}$$

Если $\det A \neq 0$, то $x^* = A^{-1}B$. Существует две группы методов решения таких систем:

1. точные: в отсутствие ошибок округления за конечное число арифметических операций дают точное решение. Пример: метод Гаусса.

2. итерационные: в ходе их применения рождается последовательность векторов, сходящихся к решению.

Явление плохой обусловленности

Изменим немного A и B. Если малое изменение исходных данных приводит к сильному изменению решения, то такая матрица называется плохо обусловленной.

1. Рассмотрим возмущения в векторе B, т.е. $b + \Delta b$,и $x + \Delta x$:

$$A(x + \Delta x) = B + \Delta B$$

Вычтем из неё (47):

$$\begin{aligned} A\Delta x &= \Delta b \\ \Delta x &= A^{-1}\Delta B \end{aligned} \tag{48}$$

Перемножим $||47||$ и $||48||$ и разделим результат на $||x|| \cdot ||B||$:

$$\underbrace{\frac{||\Delta x||}{||x||}}_{\text{погр. результата}} \leq \text{cond}(A) \underbrace{\frac{||\Delta B||}{||B||}}_{\text{погр. исх. данных}}$$

Где $\text{cond}(A) = ||A|| \cdot ||A^{-1}||$ — число обусловленности (коэффициент увеличения погрешности).

Вводят и другие числа обусловленности, например :

$$k(A) = \frac{|\lambda_k|_{\max}}{|\lambda_k|_{\min}}$$

Если $k(A) \gg 1$, то A — плохо обусловленная матрица (с большим разбросом модулей собственных значений).

Легко показать, что $k(A) \leq \text{cond}(A)$ (по следствию из Т6): $|\lambda_k|_{\max} \leq ||A||$. Тогда по Т5 матрица A^{-1} имеет собственные числа $-\frac{1}{|\lambda_k|}$. При этом $\frac{1}{|\lambda_k|_{\min}} \leq ||A^{-1}||$. Перемножая, получаем:

$$\frac{|\lambda_k|_{\max}}{|\lambda_k|_{\min}} \leq ||A|| \cdot ||A^{-1}||$$

2. Рассмотрим погрешность задания исходной матрицы A: $A+\Delta A$: Вычтем из $(A + \Delta A)(x + \Delta x) = B$ (47):

$$\begin{aligned} A\Delta x + \Delta A(x + \Delta x) &= 0 \\ \Delta x &= -A^{-1}\Delta A(x + \Delta x) \end{aligned}$$

Взяв норму, получим:

$$\frac{||\Delta x||}{||x + \Delta x||} \leq \text{cond}(A)\frac{||\Delta A||}{||A||}$$

Положение 1: Максимальные по модулю элементы матрицы имеют величину порядка $|\lambda_k|_{\max}$ или превышают эту величину (по следствию из Т5): $|\lambda_k|_{\max} \leq ||A||$.

Положение 2: Собственные значения матрицы могут меняться на величину порядка элементов матрицы возмущения ($A : \lambda_k,\ A + \Delta A : \lambda_k + \lambda$). На практике это положение очень часто имеет место.

Матрицы с большим разбросом плохо обусловлены, т.к.:

1. Элементы матрицы A заданы с погрешностью δ. Тогда максимальный элемент матрицы ΔA имеет величину порядка $\delta|A_{ij}|_{\max}$, т.е. величину порядка $\delta|\lambda_k|_{\max}$ или больше (по положению 1).

2. В соответствии с положением 2 на величину $\delta|\lambda_k|_{\max}$ могут измениться все собственные числа матрицы A: $\lambda_k \pm \delta|\lambda_k|_{\max}$. Максимальные по модуль собственные значения матрицы практически не изменятся, а а минимальные могут измениться достаточно сильно.

3. Матрице A^{-1} соответствует максимальное собственное число $\frac{1}{|\lambda_k|_{\min}}$, поэтому элементы матрицы A^{-1} тоже могут сильно измениться, и следовательно, сильно может измениться решение x^*.

Описанная ситуация будет проявляться тем острее, чем больше разброс между собственными значениями матрицы A. Работать с матрицей, у которой $\mathrm{cond}(A) > \frac{1}{\delta}$ нельзя.

Метод Гаусса

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n \end{cases}$$

В примитивной реалзации метода Гаусса (последовтельном исключением) можем стлкнуться с делением на 0 или близкий к нулю элемент, поэтому реализуется метод ведущего элемента. 2 способа:

1. На k шаге в оставшейся(необработанной) матрице находят самый большой по модулю элемент. Затем строки и столбцы переставляют так, чтобы он поменялся местами с a_{kk}. Затем на этот элемент делят k строку. Метод хорош с точки зрения надежности, но при перестановке столбцов происходит перенумерация компонент вектора x.

2. Но чаще на практике используют другой способ. На k шаге максимальный по модулю элемент разыскивают не во всей матрице, а только в k столбце и переставляют местами только строчки.

Хорошая программа, реализующая метод Гаусса, должна удовлетворять следующим условиям:

- выбор ведущего элемента
- эффективность решения нескольких СЛАУ с одинаковыми A и различными B. В частности, такая потребность может возникнуть при нхождении элементов обратной матрицы $Ax_k = b_k$
- выполнение оценки числа обусловленности

Второе условие реализуется с помощью LU-разложения.

LU-разложение

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_4 = b_1 \\ \vdots \\ a_{41}x_1 + a_{42}x_2 + \ldots + a_{4n}x_4 = b_4 \end{cases}$$

Исключим x_1 из всех столбцов крому первого(домножим слева на левую треугольную матрицу M_1):

$$M_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{a_{21}}{a_{11}} & 1 & 0 & 0 \\ -\frac{a_{31}}{a_{11}} & 0 & 1 & 0 \\ -\frac{a_{41}}{a_{11}} & 0 & 0 & 1 \end{pmatrix}$$

Тогда $M_1 Ax = M_1 B$. Дальше аналогично со вторым столбцом:

$$M_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -\frac{a^*_{32}}{a_{22}} & 1 & 0 \\ 0 & -\frac{a^*_{41}}{a_{22}} & 0 & 1 \end{pmatrix}$$

В итоге получим:

$$\underbrace{M_3 M_2 M_1}_{M} A = U$$

Где U — верхняя треугольная матрица, а M — левая треугольная (по построению). $A = M^{-1}U$, т.е.:

$$A = LU$$

Причем матрица L имеет единичную диагональ.

Если LU-разложение известно ($LUx = B$), то решение (47) сводится к решению двух систем с треугольными матрицами:

$$\begin{cases} Ly = B \\ Ux = y \end{cases} \tag{49}$$

Второе требование реализуется следующим образом: однократно находят LU разложение матрицы, а затем необходимое число раз решают системы (49).

Программы DECOMP и SOLVE

DECOMP(NDIM,N,A,COND,IPVT,WORK)

NDIM размерность матрицы A.

A исходная матрица. На выходе там находится LU-разложение этой матрицы.

COND оценка числа обусловленности.

IPVT вектор ведущих элементов. k компонента указывает, какое уравнение использовалось на k шаге.

WORK рабочий массив рамерности N (из-за убогости фортрана).

SOLVE(NDIM,N,A,B,IPVT)

B вектор правых частей, туда записывается ответ.

IPVT указывает где какие элементы в B (столбцы меняем местами).

21 Метод последовательных приближений для решения линейных систем

Систему (47) эквивалентными преобразованиями приводят в виду:

$$x = Cx + d \tag{50}$$

Точное решение (50) выглядит так:

$$x^* = (E - C)^{-1}d \tag{51}$$

Запишем вместо (50) систему разностных уравнений (52), которую решим пошаговым методом:

$$x_{n+1} = Cx_n + d \tag{52}$$

3 вопроса:

1. Сходится ли (52)?
2. Если сходится, то к чему?
3. Как быстро сходится?

Точное решение (52) имеет вид:

$$x_n = C^n x_0 + (E - C^n)(E - C)^{-1}d \tag{53}$$

Вычтем из (50) (53):

$$x^* - x_n = C^n(E - C)^{-1}d - C^n x_0 = C^n(x^* - x_0)$$

Обозначим $\varepsilon_n = x^* - x_n$ Тогда

$$x^* - x_n = C^n \varepsilon_0$$

$$C^n = \sum_{k=1}^{m} T_k \lambda_k^n$$

Нобходимое и достаточное условие сходимости метода: $|\lambda_k| < 1 \ \forall k$. На практике собственые значения не считают, а ипользуют достаточное условие $||C|| < 1$, т.к. по второму следствию из Т5 $|\lambda_k| \leq ||C||$.

Итерационный процесс будет сходится тем быстрее, чем меньше собственные значения матрицы C или её норма. Поэтому выбор x_0 влияет не на сходимость, а на её скорость, и все искусство заключается в выборе таких эквивалентных преобразований, которые делают $||C||$ как можно меньше.

22 Решение нелинейных уравнений и систем

$f(x) = 0$ в общем случае может иметь сколько угодно решений (и не иметь вовсе).

Задаче поиска нуля предшествует этап локализации промежутка с 0. Рассмотрим вариант, когда x — скаляр, а $f(x)$ — скалярная функция.

1. Дихотомия (метод бисекции). Суть метода заключается в делении отрезка на 2 части и выбора того отрезка, где $f(a)f(b) < 0$, т.е. на концах которого функция имеет разные знаки. После n шагов промежуток сокращается в 2^n раз независимо от вида функции, поэтому метод гарантированно сходится. Но при этом будет найден только один 0 из имеющихся (если есть).

2. Метод секущих. Точкой деления отрезка на две части выбирается не середина, как в методе бисекций, а точка пересечения прямой, проведенной через $f(a)$ и $f(b)$, с осью абсцисс.

   Метод хорд (модификация). Через две точки проведём интерполяционный полином 1 степении и возьмем его корень в качестве точки деления отрезка.

   Оба этих метода также гарантированно сходятся.

3. Метод обратной параболической интерполяции. Начиная со второго шага можно по 3 точкам строить $Q_2(x)$, решать квадратное уравнение $Q_2(x) = 0$, получать четвертую точку и одну из старых точек отбрасывать.

   Если функция хорошо описывается прямой или параболой, то ответ полчаем очень быстро. В случае замедления сходимости (приближения к краю промежутка) на 1–2 шага используетя метод бисекций.

Программа ZEROIN

ZEROIN(A,B,F,EPS)

A,B концы промежутка ($F(A)F(B) < 0$ должно быть).

EPS требуемая точность.

Внутри программы **ZEROIN** реализованы два метода — метод бисекций и метод обратной параболической интерполяции. Переключение на первый метод происходит при замедлении сходимости.

23 Метод последовательных приближений

$$f(x) = 0 \tag{54}$$

Эквивалентными преобразованиями (54) приводят к виду $x = \varphi(x)$, корень которого обозначают за x^*, т.е. $x^* = \varphi(x^*)$. При этом $x^* = x_n + \varepsilon_n$ Вместо этого уравнения решается разностное уравение (55) пошаговым методом:

$$x_{n+1} = \varphi(x_n) \tag{55}$$

Снова имеем три вопроса:

1. Сходится ли (55)?
2. Если сходится, то к чему?
3. Как быстро сходится?

$$x^* - x_n = \varphi(x^*) - \varphi(x_n)$$
$$\varepsilon_{n+1} = \varphi(x_n + \varepsilon_n) - \varphi(x_n) = \varphi(x_n) + \frac{\varepsilon_n}{1!}\varphi'(\eta) - \varphi(\eta) = \varepsilon_n \varphi'(\eta)$$

Условием сходимости является $|\varphi'(\eta)| < 1$, причем процесс сходится тем быстрее, чем меньше эта производная, следовательно, всё искусство состоит в переходе от (54) к виду $x = \varphi(x)$.

Формула (55) сохраняет свой вид и при решении систем уравнений (т.е. когда x, φ — вектора). Можно показать, что при этом достаточным условием сходимости будет $||\frac{\partial \varphi}{\partial x}|| < 1$, т.е. норма матрицы Якоби меньше 1. Необходимое условие $|\lambda_k| < 1$. Матрица Якоби выглядит так:

$$\frac{\partial \varphi}{\partial x} = \begin{pmatrix} \frac{\partial \varphi^{(1)}}{\partial x^{(1)}} & \frac{\partial \varphi^{(1)}}{\partial x^{(2)}} & \cdots & \frac{\partial \varphi^{(1)}}{\partial x^{(m)}} \\ \vdots & & & \vdots \\ \frac{\partial \varphi^{(m)}}{\partial x^{(1)}} & \frac{\partial \varphi^{(m)}}{\partial x^{(2)}} & \cdots & \frac{\partial \varphi^{(m)}}{\partial x^{(m)}} \end{pmatrix}$$

Метод Ньютона (метод касательных)

$f(x) = 0$, $x^* = x_n + \varepsilon_n$

$$f(x^*) = 0 = f(x_n + \varepsilon_n) = f(x_n) + \frac{\varepsilon_n}{1!} f'(x_n) + \frac{\varepsilon_n^2}{2!} f''(\eta) \tag{56}$$

Отбрасывая последнее слагаемое (погрешность), получим:

$$\varepsilon_n = -\frac{f(x_n)}{f'(x_n)}$$

Тогда очередное приближение выглядит так:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{57}$$

Оценим скорость сходимости метода Ньютона. Для этого вычтем из (57) x^* из обоих частей:

$$\varepsilon_{n+1} = \varepsilon_n + \frac{f(x_n)}{f'(x_n)} = \frac{f(x_n) + \varepsilon_n f'(x_n)}{f'(x_n)}$$

Выразим числитель через вторую производную по формуле (56):

$$\varepsilon_{n+1} = -\frac{\varepsilon_n^2}{2} \frac{f''(\eta)}{f'(x_n)}$$

Тогда, если $|\frac{f''(\eta)}{2f'(x_n)}| < c$, т.е. ограничена, то $|\varepsilon_{n+1}| < c\varepsilon_n^2$ и метод Ньютона сходится. Говорят, что метод Ньютона имеет квадратичную скорость сходимость, т.к. при $c = 1$ будем получать по два верных знака на каждом шаге.

Метод Ньютона имеет высокую скорость сходимости, если вообще сходится, т.е. он карпизен к выбору начального приближения. Поэтому одно из часто накладываемых условий — знакопостоянство производной.

На практике часто используют модифицированный метод Ньютона:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_0)}$$

Производную считают однократно или достаточно редко. Скорость сходимости при этом падает, но каждый шаг обходится "дешевле".

Расширить область сходимости часто позволяет метод Ньютона с регулировкой шага:

$$x_{n+1} = x_n - \alpha\frac{f(x_n)}{f'(x_n)}, \qquad 0 < \alpha \leq 1$$

На первых шагах выбирают $\alpha \approx 1/3$ (например). По мере приближения к 0 выбирают $\alpha \approx 1$ и формула переходит в обычный метод Ньютона.

Решение систем нелинейных уравнений

Рассмотрим систему

$$F(X) = 0 \tag{58}$$

(т.е. F и X — вектора). Точное решение этой системы обзоначим так:

$$X_* = \begin{pmatrix} X_*^{(1)} \\ X_*^{(2)} \end{pmatrix}$$

$X_* = X_n + \varepsilon_n$. Приближение на n шаге:

$$X_n = \begin{pmatrix} X_n^{(1)} \\ X_n^{(2)} \end{pmatrix}$$

В первое уравнение (58) подставим точное решение:

$$0 = f^{(1)}\left(X_*^{(1)}\ X_*^{(2)}\right) = f^{(1)}\left(X_n^{(1)} + \varepsilon_n^{(1)}\ X_n^{(2)} + \varepsilon_n^{(2)}\right)$$

Разложим в ряд:

$$f^{(1)}\left(X_*^{(1)}\ X_*^{(2)}\right) = f^{(1)}\left(X_n^{(1)}\ X_n^{(2)}\right) + \frac{\partial f^{(1)}}{\partial x^{(1)}}\left(X_n^{(1)}\ X_n^{(2)}\right)\varepsilon_n^{(1)} + \frac{\partial f^{(1)}}{\partial x^{(2)}}\left(X_n^{(1)}\ X_n^{(2)}\right)\varepsilon_n^{(2)} + \cdots$$

Проделав аналогичную операцию для второго уравнения в выразив $f(X_n)$, получим:

$$\begin{cases} \frac{\partial f^{(1)}}{\partial x^{(1)}}\varepsilon_n^{(1)} + \frac{\partial f^{(1)}}{\partial x^{(2)}}\varepsilon_n^{(2)} = -f^{(1)}\left(X_n^{(1)}\ X_n^{(2)}\right) \\ \frac{\partial f^{(2)}}{\partial x^{(1)}}\varepsilon_n^{(1)} + \frac{\partial f^{(2)}}{\partial x^{(2)}}\varepsilon_n^{(2)} = -f^{(2)}\left(X_n^{(1)}\ X_n^{(2)}\right) \end{cases}$$

Метод Ньютона для решения СУ:

$$\begin{cases} X_{n+1} = X_n + \varepsilon_n \\ \frac{\partial f}{\partial x}(X_n)\varepsilon_n = -f(X_n) \end{cases}$$

Каждый шаг — решение СЛАУ относитеьлно ε_n. Этот метод также чувствителен к выбору начального приближения.

На практике часто используют модифицированный метод Ньютона:

$$\begin{cases} X_{n+1} = X_n + \varepsilon_n \\ \frac{\partial f}{\partial x}(X_0)\varepsilon_n = -f(X_n) \end{cases}$$

Матрицу Якоби вычисляют в точке X_0 и производят её LU-разложение с помощью программы **DECOMP**. На всех последующих шагах при решении используется программа **SOLVE**. Матрица Якоби вычисляется и раскладывается повторно только в случае замедления сходимости.

24 Решение обыкновенных дифференциальных уравнений и задача Коши

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(t,x), \quad x(t_0) = x_0 \tag{59}$$

$x(t)$ — решение (59). Рассмотрим скалярный случай, однако все полученные методы годятся и для решения систем уравнений.

Основная идея большинства численных методов: исходное дифференциальное уравнение приближенно сводится к некоторому разностному уравнению, которое затем решается пошаговым методом.

Для этого введем дискретный набор значений $t_n = t_0 + nh$ (если необходимо между ними решение посчитать, то всегда можно построить интерполяционный полином), где h — шаг интергирования. Для краткости обозначим $x(t_n) = x_n$ и $f(t_n, x_n) = f_n$.

Перейдем к разностному уравнению, для этого проинтергируем (59) на $[t_n, t_{n+1}]$:

$$x_{n+1} = x_n + \int_{t_n}^{t_{n+1}} f(\tau, x(\tau))\,\mathrm{d}\tau \tag{60}$$

Различные методы отличаются друг от друга способом вычисления интеграла в формуле (60).

Посчитаем по формуле прямоугольников:

$$x_{n+1} = x_n + hf(t_n, x_n) \tag{61}$$

$$x_{n+1} = x_n + hf(t_{n+1}, x_{n+1}) \tag{62}$$

$$x_{n+1} = x_n + \tfrac{h}{2}\left(f(t_{n+1}, x_{n+1}) + f(t_n, x_n)\right) \tag{63}$$

Формула (61) — явный метод ломаных Эйлера, (62) — неявный метод ломаных Эйлера, (63) — неявный метод трапеций. Неявными они называются потому, что являются нелинейными уравнениями от x_{n+1}.

Попробуем воспользоваться квадратурной формулой Симпсона:

$$\frac{b-a}{6}\left(f(a) + f(b) + f\left(\frac{a+b}{2}\right)\right)$$

Для её применения надо вычислять $f(x_{n+\frac{1}{2}})$ — неизвестно что. Все остальный квадратурные формулы не подходят по этим же соображениям.

25 Методы Адамса

В середине XIX века Адамс предложил использовать предыдущие точки $(t_n, t_{n-1}, t_{n-2}, \ldots)$ — строить по ним интерполяционный полином и интегрировать его в (60). Например, по t_n, t_{n-1} строим:

$$f(\tau, x(\tau)) = \frac{\tau - t_{n-1}}{t_n - t_{n-1}} f_n + \frac{\tau - t_n}{t_{n-1} - t_n} f_{n-1}$$

Интегрируя, получим:

$$x_{n+1} = x_n + \frac{h}{2}(3f_n - f_{n-1})$$

Если взять 4 предыдущие точки и построить Q_3, а затем его проинтегрировать, то получим более точный метод Адамса:

$$x_{n+1} = x_n + \frac{h}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}) \tag{64}$$

Методы Адамса не "самостартующие" и для начала своей работы требуют расчета дополнительных начальных условий.

Трудоёмкость методов состоит в количестве вычислений f на одном шаге. И у метода Адамса и у прочих надо считать всего один раз. В методе Адамса используются результаты, полученные на предыдущем шаге — в этом заключается его преимущество.

Локальная и глобальная погрешность. Порядок точности метода

Для простоты рассмотрим явный метод ломаных Эйлера:

$$x_{n+1} = x_n + hf(t_n, x_n)$$

1. $f(t_n, x_n) = f(t_n)$ — не зависит от x. Метод превращается в квадратурную формулу левых прямоугольников. ε_0 — погрешность вычисления x_1. ε_1 — погрешность вычисления x_2. На n шаге $\varepsilon_n = \sum_{k=0}^{n-1} \varepsilon_k$
2. $f = f(t_n, x_n)$ — зависит от x. x_1 получаем с погрешностью ε_0. Но $x_2 \neq x_1 + \varepsilon_0 + \varepsilon_1$, т.к. в формуле левых прямоугольников $(hf(t_1, x_1))$ f зависит от x, полученного с погрешностью. Поэтому на n шаге может вообще не быть верных разрядов.

Погрешность зависит от h, f, формулы метода и устойчивости методв. Поэтому все методы делят на 2 класса — устойчивые и неустойчивые.

В общем случае погршность n шага является сложной функцией от всех погрешностей, допущенных на предыдущих шагах. Если погрешность, допущенная на одном шаге экспоненциально растет на последних шагах, то говорят о неустойчивых алгоритмах. В противном случае, если её удается удержать в управляемых границах, то говорят об устойчивых алгоритмах.

На практике вводят 2 вида погрешности — локальную и глобальную:
Локальная погрешность — погрешность, допущенная на одном шаге при условии, что все предыдущие значения x_n были посчитаны точно.

Глобальная погрешность — разность между точным и приближенным решением в данной точке.

Малая локальная погрешность не всегда приводит к малой величине глобальной погрешности. Если устойчивость метода обеспечена, то малой величины локальной погрешности будет следовать малая величина глобальной погрешности.

В качестве характеристики локальной погрешности выстуает степень метода или порядок точности метода.

Все ранее рассматриваемые методы укладывались в следующую схему:

$$x_{n+1} = x_n + hF(t_n, h, x_n, x_{n-1}, \dots, x_{n-s}) \tag{65}$$

С другой стороны $x_{n+1} = x(t_n + h)$. Разложим в ряд по степеням h:

$$x(t_n + h) = x_n + \sum_{k=1}^{\infty} \frac{h^k x^{(k)}(t_n)}{k!} \tag{66}$$

Разложим (65) в ряд в точке t_n:

$$x_{n+1} = x_n + \sum_{k=1}^{\infty} \alpha_k h^k x^{(k)}(t_n) \tag{67}$$

Говорят, что метод (65) имеет степень точности (порядок) p, если разложение метода (67) совпадает с точным разложением (66) до h^p включительно.

Установим точность ранее рассмотренных методов:

- Явный метод ломаных Эйлера:

  $$x_{n+1} = x_n + hf(t_n, x_n) = x_n + hx'(t_n)$$

  т.к. $x' = f(t, x)$. Сравним с (66): совпадает только первый член разложения — это метод первой степени точности.

- Неявный метод ломаных Эйлера:

  $x_{n+1} = x_n + hf(t_{n+1}, x_{n+1}) = x_n + hx'(t_n + h) = x_n + hx'(t_n) + h^2 x''(t_n) + \cdots$

  Это тоже метод первой степени точности.

- Формула трапеций

  $$x_{n+1} = x_n + \frac{h}{2}\left(x'(t_n) + x'(t_n + h)\right) =$$
  $$x_n + \frac{h}{2}\left(x'(t_n) + x'(t_n) + hx''(t_n) + \frac{h^2}{2}x'''(t_n) + \cdots\right)$$

  Этот метод имеет вторую степень точности.

- Метод Адамса

  $$x_{n+1} = x_n + \frac{h}{2}(3f_n - f_{n-1}) = x_n + \frac{h}{2}(3x'(t_n) - x'(t_n - h)) =$$
  $$= x_n + \frac{h}{2}(3x'(t_n) - x'(t_n) + hx''(t_n) - h^2 x'''(t_n) + \cdots)$$

  Этот метод тоже имеет вторую степень точности.

  Легко убедиться в том, что (64) — метод Адамса четвертой степени точности.

26 Методы Рунге-Кутты

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(t,x)$$

Сделаем из него разностное уравнение 1 порядка.

$$\begin{cases} x^*_{n+1} = x_n + hf(t_n, x_n) \text{ — явный метод ломаных Эйлера} \\ x_{n+1} = x_n + \dfrac{h}{2}\big(f(t_n,x_n) + f(t_{n-1},x_{n-1})\big) \text{ — неявный метод трапеций} \end{cases} \tag{68}$$

Это — метод Эйлера-Коши.

$$\begin{cases} x^*_{n+1/2} = x_n + \dfrac{h}{2} f(t_n, x_n) \\ x_{n+1} = x_n + h\left(f\left(t_n + \dfrac{h}{2}, x_{n+1/2}\right)\right) \end{cases} \tag{69}$$

Этот метод носит название усовершенствованного метода ломаных Эйлера.

Степень точности была равна двум, но сохранится ли она при приближенном вычислении x^*_{n+1} или станет 1? Если нет, то метод ломаных Эйлера дает ту же точность при меньших затратах.

Обобщим:

$$\begin{cases} k_1 = hf(t_n, x_n) \\ k_2 = hf(t_{n+\alpha h}, x_{n+\beta k_1}) \\ x_{n+1} = x_n + p_1 k_1 + p_2 k_2, \text{ где } p_1 \text{ и } p_2 \text{ — неопределённые коэффициенты} \end{cases}$$

Параметры α, β, p_1, p_2 будем выбирать так, чтобы метод имел максимальную степень точности.

$$x_{n+1} = x(t_n + h) = x_n + \sum_{k=1}^{\infty} \frac{h^k x^{(k)}(t_n)}{k!} \text{ — точное разложение} \tag{70}$$

$x' = f(t,x)$. Тогда

$$x'' = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial x}\frac{\partial x}{\partial t} = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial x} f$$

Подставим в формулу метода k_1 и k_2:

$$x_{n+1} = x_n + p_1 hf(t_n, x_n) + p_2 hf(t_n + \alpha h, x_n + \beta h f_n)$$

Разложим в ряд по степеням h в точке t_n:

$$x_{n+1} = x_n + p_1 h f_n + p_2 h\left(f(t_n,x_n) + \frac{h}{1!}\left(\frac{\partial f}{\partial t}\alpha + \frac{\partial f}{\partial x}\beta f_n\right)\right) \tag{71}$$

Выбирать параметры будет так, чтобы (71) максимально совпадало с (70).

$$\begin{cases} p_1 + p_2 = 1 \\ p_2\alpha = 1/2 \\ p_2\beta = 1/2 \end{cases}$$

Получили 3 уравнения с тремя неизвестными, поэтому существует бесконечно много методов Рунге-Кутты со степенью точности 2. Наиболее популярны следующие методы:

- Метод Эйлера-Коши. При этом $p_1 = p_2 = 1/2$ и $\alpha = \beta = 1$
- Усовершенствованный метод ломанных Эйлера: $p_1 = 0$, $p_2 = 1$, $\alpha = \beta = 1/2$

Оба этих метода имеют вторую степень точности.

Методы 3 степени точности требуют, чтобы f вычислялась на одном шаге 3 раза, 4 степени — 4 раза, метод 5 степени требует вычисления f 6 раз. Поэтому метод Рунге-Кутта 4 степени получил наибольшую популярность:

$$\begin{cases} k_1 = hf(t_n, x_n) \\ k_2 = hf(t_n + \dfrac{h}{2}, x_n + \dfrac{1}{2}k_1) \\ k_3 = hf(t_n + \dfrac{h}{2}, x_n + \dfrac{1}{2}k_2) \\ k_4 = hf(t_n + h, x_n + k_3) \\ x_{n+1} = x_n + \dfrac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \end{cases}$$

Если $f \neq f(x)$, то можно привести слагаемые и получить квадратурную формулу Симпсона.

Программа RKF45

Используются методы Рунге-Кутта и Фельберга 4 и 5 степени точности. Методы построены таким образом, что и метод 4 степени и метод 5 степени использует одни и те же значения (залуга Фельберга).

$$x_{n+1}^{(5)} = x_n + \sum_{i=1}^{6} P_i k_i$$

$$x_{n+1}^{(4)} = x_n + \sum_{i=1}^{6} P_i^* k_i$$

Используется для контроля погрешности:

$$x_{n+1}^{(5)} - x_{n+1}^{(4)} = \sum_{i=1}^{6} (P_i - P_i^*) k_i$$

Если эта разность мала, то увеличиваем шаг, если мала — уменьшаем.

RKF45(F,N,X,T,TOUT,RE,AE,INDEX,WORK,IWORK)

F процедура, вычисляющая $f(t, x)$.

N порядок системы уравнений.

X вектор решения в точке T.

T,TOUT начальные и конечные точки.

RE,AE относительная и абсолютная погрешности.

INDEX переменная управления расчетом. На входе при первом обращении INDEX=1, при последнем INDEX=2. Нормальное выходное значение INDEX=2.

WORK,IWORK рабочие массивы.

27 Глобальая погрешность. Ограничение на шаг интегрирования

На практике глобальную погрешность тяжело оценивать, поэтому оценивают локальную, обеспечивая при этом устойчивость метода.

Будем изучать поведение глобальной погрешности на примере решения системы ДУ:

$$\frac{\mathrm{d}x}{\mathrm{d}x} = Ax \tag{72}$$

Решение любой нелинейной системы в окрестности любой точки может быть аппроксимировано решением системы (72), поэтому если метод плохо себя ведёт в при решении (72), то он почти наверняка будет плохо себя вести и при решении нелинейной системы.

Предположим, что $\Re\lambda_k < 0 \ \forall k$. Тогда точное решение (72) будет ассимптотически устойчивым и ограниченным. Для качественного соответствия точного и приближенного решения следует потребовать, чтобы решение разностного уравнения выбранного численного метода также было АУ и ограниченным.

Рассмотрим явных метод ломаных Эйлера:

$$x_{n+1} = x_n + hf(t_n, x_n) = x_n + hAx_n = (E + hA)x_n$$

Необходимо, чтобы $|1 + h\lambda_k| < 1$. Если $\lambda_k = \alpha + i\omega$, то это эквивалентно $|(1 + h\alpha) + ih\omega| < 1$ или:

$$(1 + h\alpha)^2 + h^2\omega^2 < 1$$

Множество значений $h\lambda$, удовлетворяющее устойчивости метода называется областью устойчивости данного метода. В частности, если λ_k вещественные, то получаем следующее ограничение на шаг:

$$h < \frac{2}{|\lambda_k|_{\max}}$$

Но чаще оценивают не с помощью λ_k, а используют достаточное условие $h < \frac{2}{||A||}$

Шаг придётся выбирать тем меньше, чем больше разброс между λ_k, т.е. чем хуже обусловлена A. Такие ДУ (с большим разбросом λ_k) название жестких систем ДУ. Для их решений характерны два участка:

- начальный участок очень маленькой продолжительности (пограничный слой). Характеризуется большими производными.
- остальной участок с относительно малыми производными.

Такие же ограничения на шаг имеют и другие явные методы. Так, например, для методов Рунге-Кутта 2 степени $h|\lambda_k| < 2$, 4 степени $h|\lambda_k| < 2,78$, а для метода Адамса 4 степени $h|\lambda_k| < 0,3$.

Для решения жестких систем целесообразно применять методы, область устойчивости которых включает в себя всю или почти всю левую полуплоскость.

Рассмотрим неявный метод ломаных Эйлера:

$$\begin{aligned} x_{n+1} &= x_n + hf(t_{n+1}, x_{n+1}) = x_n + hAx_{n+1} \\ (E - hA)x_{n+1} &= x_n \\ x_{n+1} &= (E - hA)^{-1}x_n \end{aligned}$$

Для устойчивости необходимо:

$$\frac{1}{|1 - h\lambda_k| < 1}, \text{ т.е. } |1 - h\lambda_k| > 1$$

Для неявного метода трапеций:

$$x_{n+1} = x_n + \frac{h}{2}(f_n + f_{n+1}) = x_n + \frac{h}{2}(Ax_n + Ax_{n+1})$$

$$x_{n+1} = \left(E - \frac{h}{2}A\right)^{-1}\left(E + \frac{h}{2}A\right)x_n$$

Тогда:

$$\left|\frac{1 + \frac{h}{2}\lambda_k}{1 - \frac{h}{2}\lambda_k}\right| < 1$$

Если $\lambda_k = \alpha + i\omega$, то получаем:

$$\left(1 + \frac{h}{2}\alpha\right)^2 + \frac{h^2\omega^2}{4} < \left(1 - \frac{h}{2}\alpha\right)^2 + \frac{h^2\omega^2}{4}$$

Это соответствует левой полуплоскости: $2h\alpha < 0$.

Почти все методы для решения жестких систем являются неявными. Для нахождения x_{n+1} на каждом шаге используется метод Ньютона решения нелинейных систем.

28 Метод Ньютона в неявных алгоритмах решения дифференциальных уравнений

Для примера рассмотрим неявный метод ломаных Эйлера. Нахождение x_{n+1} сводится к решению системы

$$F(z) = z - x_n - hf(t_{n+1}, z) = 0$$

методом Ньютона.

$$\frac{\partial F}{\partial z}(z^{(k)})(z^{(k+1)} - z^{(k)}) = -F(z^{(k)}) \quad \frac{\partial F}{\partial z} = E - h\frac{\partial f}{\partial z}$$

Где $z^{(k)}$ - k приближение к значению x_{n+1}. При этом весьма эффективен модифицированный метод Ньютона, когда матрица $\frac{\partial F}{\partial z}$ вычисляется в точке x_0, раскладывается программой **DECOMP** и на последующих шагах используется только программа **SOLVE**. С обращением матрицы $\frac{\partial F}{\partial z}$ нет проблем, т.к. она имеет не слишком большое число обусловленности, даже если матрица $\frac{\partial f}{\partial z}$ плохо обусловлена.

В данном случае с выбором начального приближения, к которому так капризен метод, проблем нет — в качестве $z^{(0)}$ может быть выбранозначение x_n или выполнен шаг явным методом ломаных Эйлера.

В итоге применение метода Ньютона в неявных алгоритмах может быть описано следующей схемой:

1. В некоторой точке производится вычисление матрицы Якоби, а затем производится её разложение с помощью программы **DECOMP**.

2. По начальному условию $z^{(0)}$, рассчитанному с помощью явного метода ломаных Эйлера, выполняется итерация метода Ньютона для получения x_{n+1}.

3. После одной-двух итераций по методу Ньютона при достижении сходимости осуществляется переход к шагу 2. Возврат к шагу 1 производится лишь в том случае, если метод Ньютона перестает сходится за три итерации.