Интерполяция и числовые ряды

Задача интерполяции формулируется следующим образом: по значениям некоторой функции $f_0, f_1, \dots f_n$ в дискретных точках $x_0, x_1, \dots x_n$ (узлах интерполяции) восстановить ее значения при любых других значениях x. Очевидно, что эта задача не имеет единственного решения и, кроме того, требует знания таких свойств интерполируемой функции, как ее гладкость, т.е. существование и ограниченность ее производных до некоторого порядка включительно. В дальнейшем будем предполагать, что $f(x) \in C_n[x_0, x_n]$, т.е. $f(x)$ непрерывна вместе со своими производными до порядка n включительно.

Многочлен Лагранжа

Среди многих способов интерполяции наиболее распространен способ линейной интерполяции, когда приближение отыскивается в виде непрерывной функции, являющейся линейной комбинацией некоторых известных, заранее выбранных m непрерывных функций

$$g_n(x; \alpha_0, \alpha_1, \dots, \alpha_m) = \sum_{i=0}^{m} \alpha_i \varphi_i(x),$$

где $\varphi_i(x)$ - известные функции, называемые базисными; α_i - коэффициенты, значения которых определяются из условия совпадения интерполирующей функции g_n с исходной в узлах интерполяции:

$$\sum_{i=0}^{m} \alpha_i \varphi_i(x_j) = f(x_j), \quad j = 0,1,\dots,n \qquad (1)$$

Чтобы решение системы полученной системы линейных алгебраических уравнений (1) существовало и было единственным, необходимо и достаточно, чтобы ее числовая матрица $\left\| \varphi_i(x_j) \right\|$ была квадратной и невырожденной. Следовательно, необходимо, чтобы выполнялось условие $m = n$, а выбранная совокупность функций $\varphi_i(x)$ была линейно независимой, т.е. образовывала базис. Метод определения коэффициентов α_i путем непосредственного решения системы (1) и называется методом неопределенных коэффициентов. Наиболее часто в качестве базисных функций $\varphi_i(x)$ используются многочлены. Тогда $\varphi_i(x) = x^i$, и система уравнений (1) принимает вид

$$\sum_{i=0}^{n}\alpha_i x_j^i = f(x_j), \quad j = 0,1,\ldots,n \tag{2}$$

Если при этом все x_i различны, то определитель этой системы (определитель Вандермонда) не обращается в нуль и система (2) всегда имеет единственное решение. Таким образом, интерполяционный многочлен существует и единственный. Однако при численном решении системы (2) возникают большие затруднения, связанные с тем, что при больших значениях n она оказывается плохо обусловленной, т.е. становится чувствительной к ошибкам округления и чтобы обойти эту трудность, можно поступить следующим образом. Представим $g_n(x)$ в виде

$$g_n(x) = \sum_{i=0}^{n} f(x_i)\Phi_i(x),$$

где $\Phi_i(x)$ - многочлен степени n, причем такой, что $\Phi_i(x_j) = \begin{cases} 0, \text{ при } i \neq j, \\ 1, \text{ при } i = j. \end{cases}$

Очевидно, что его корнями являются все узлы таблицы $x_j \neq x_i$, а потому будем строить его в виде

$$\Phi_i(x) = A(x - x_0)(x - x_1)\ldots(x - x_{i-1})(x - x_{i+1})\ldots(x - x_n),$$

где A - не определенная пока константа. Определяя значение A из условия $\Phi_i(x_i) = 1$, окончательно получим

$$L_n(x) = \sum_{i=0}^{n} f(x_i) \prod_{j=0,\, i\neq j}^{n} \frac{x - x_j}{x_i - x_j}. \tag{3}$$

Построенный интерполяционный полином называется интерполяционным многочленом Лагранжа. Если $f(x) \in C_{n+1}[x_0, x_n]$, то можно дать оценку погрешности интерполяции (3).

$$\varepsilon(x) = |f(x) - L_n(x)| \leq \frac{M_{n+1}}{(n+1)!} |(x - x_0)(x - x_1)\ldots(x - x_n)|, \tag{4}$$

где $M_{n+1} = \underbrace{\sup}_{x_0 \leq x \leq x_n} \left| f^{(n+1)}(x) \right|$.

Как показывает анализ (4), по мере роста числа узлов интерполяции n погрешность интерполяции будет уменьшаться только при условии, если числитель растет не быстрее знаменателя, что далеко не всегда может иметь место даже для гладких функций. Поэтому на практике сравнительно редко строят много-

член Лагранжа с использованием всех узлов интерполяции. Чтобы избежать больших значений величины M_n чаще всего поступают следующим образом. Выбирают несколько узлов, ближайших к точке x (обычно не более трех - четырех), и для них строят многочлен Лагранжа (3). В этом случае $i_н \le i \le i_k$, где $i_н, i_k$ - номера узлов начала и конца интервала интерполяции. Основным недостатком такого подхода является разрыв производных функции Лагранжа при переходе от одного интервала к другому.

Сплайны

Для преодоления основного недостатка многочлена Лагранжа - зависимости точности интерполяций от числа узлов интерполяции был предложен подход, основанный на построении интерполирующих функций, представляющих собой многозвенники, каждое звено которых есть полином фиксированной (заданной) степени. Такие функции получили название сплайн-функций, или просто сплайнов, которые широко применяются при интерполяции таблично заданных функций. Пусть узлы таблично заданной фикции расположены (для определенности) в порядке возрастания, и образованный ими отрезок $[x_0, x_N]$ разобьем на интервалы (звенья) $[x_0, x_1], [x_1, x_2], \ldots, [x_{N-1}, x_N]$. Назовем сплайном порядка m функцию- многозвенник, являющуюся на каждом из интервалов $[x_{n-1}, x_n]$ многочленом степени m, т.е.
удовлетворяющую условиям непрерывности производных до порядка m – 1

$$\begin{aligned} &P_{nm}(x) = a_{n0} + a_{n1}x + a_{n2}x^2 + \ldots + a_{nm}x^m, \\ &x_{n-1} \le x \le x_n, \quad n = 1,2,3,\ldots N \end{aligned} \tag{5}$$

включительно во всех точках на границах интервалов $x_1, x_2, \ldots x_{N-1}$:

$$P_{nm}^{(k)}(x) = P_{n+1,m}^{(k)}(x), \quad k = 0,1,\ldots,m-1; \quad n = 1,2,\ldots N \tag{6}$$

Всего имеется $Q = (m+1)N$ неизвестных коэффициентов и соотношения (6), называемые условиями гладкого сопряжения, образуют систему из $(N-1)m$ линейных алгебраических уравнений. Для замыкания этой системы должны быть сформулированы дополнительные условия. В качестве таких условий можно использовать условия равенства сплайна значениям интерполируемой функции в узлах интерполяции (условие интерполяции)

$$P_{nm}(x_n) = f(x_n), \quad n = 0,1,\ldots,N \tag{7}$$

или какие-либо условия близости сплайна в них этой функции (условия аппроксимации). Эти дополнительные условия (интерполяции или аппроксимации), накладываемые в узлах на сплайн-функцию, дают еще N + 1 условие. Оставшиеся m - 1 условий можно задавать только на границах отрезка, т.е. в точках x_0 и x_N, и только для значений производных сплайн -функции.

Рассмотрим процесс построения интерполирующих сплайнов (в качестве дополнительных условий используются условия интерполяции) последовательно для случаев m = 1, 2, 3.

1. При m = 1 Q = 2N и на каждом из интервалов $[x_{n-1}, x_n]$ имеем

$$P_{n1}(x) = a_{n0} + a_{n1}x.$$

Условия гладкого сопряжения (6) и условия интерполяции (7) дают замкнутую попарно - независимую систему уравнений

$$a_{n0} + a_{n1}x_{n-1} = f(x_{n-1}),$$
$$a_{n0} + a_{n1}x_n = f(x_n),$$

откуда

$$a_{n1} = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}, \quad a_{n0} = f(x_{n-1}) - a_{n1}x_{n-1}$$

2. При $m = 2$ число неизвестных $Q = 3N$, число уравнений из (6), (7) равно $3N - 1$. Недостающее одно условие можно определить, задав $P_{12}^{(1)}(x_0)$ или $P_{N2}^{(1)}(x_N)$. Пусть, для определенности, задано значение $P_{12}^{(1)}(x_0) = y_0'$. Тогда из системы трех уравнений

$$a_{10} + a_{11}x_0 + a_{12}x_0^2 = f_0,$$
$$a_{11} + 2a_{12}x_0 = y_0',$$
$$a_{10} + a_{11}x_1 + a_{12}x_1^2 = f_1$$

находятся значения коэффициентов a_{10}, a_{11}, a_{12}, а затем и значение производной $P_{12}^{(1)}(x_1)$, которое можно использовать в качестве дополнительного (граничного) условия для следующего интервала, что позволяет аналогично определить значения коэффициентов a_{20}, a_{21}, a_{22} и т.д. Однако счет по такому алгоритму оказывается неустойчивым для больших значений N, и в таком виде сплайн второго порядка обычно не строят.

Кубический сплайн

Случай m = 3 (кубический сплайн) наиболее распространен, и поэтому остановимся на нем особо. На каждом из N интервалов $[x_{n-1}, x_n]$ сплайн представляет собой кубический полином:

$$S_n(x) = \sum_{m=0}^{3} a_{mn}(x_n - x)^m, \quad n = 1,2,\ldots,N, \tag{8}$$

В узлах интерполяции выполняются равенства $S(x_n) = f_n$, $n = 0,1,\ldots,N$, которые дают N + 1 условие. Требуя непрерывности самой сплайн-функции, ее первых и вторых производных в соответствии с (6) во внутренних узлах отрезка $[x_0, x_N]$, получаем еще $3(N-1)$ условий. Всего неизвестных $4N$, уравнений $4N-2$. В качестве дополнительных условий на концах отрезка можно задать значения либо первых производных сплайн-функции, либо вторых, либо их комбинации. Чаще всего, когда нет информации о поведении сплайн-функции на концах отрезка, здесь задают нулевые значения ее вторых производных. Это соответствует предположению, что в вблизи границ отрезка сплайн-функция близка к линейной. Итак, пусть $S_1^{(2)} = S_N^{(2)} = 0$. Тогда для определения коэффициентов сплайна имеет место следующая система уравнений

$$\begin{aligned}
&a_{0n} = f_n, \quad n = 1,2,\ldots,N \\
&\quad f_n + a_{1n}h_n + a_{2n}h_n^2 + a_{3n}h_n^3 = f_{n-1}, \quad n = 1,2,\ldots,N \\
&a_{1n} + 2a_{2n}h_n + 3a_{3n}h_n^2 = a_{1,n-1}, \quad n = 2,3,\ldots,N \\
&2a_{2n} + 6a_{3n}h_n = 2a_{2,n-1}, \quad n = 2,3,\ldots,N,
\end{aligned} \tag{9}$$

где $h = x_n - x_{n-1}$. Замыкают (9) граничные условия

$$\begin{aligned}
S_1^{(2)}(x_0) = 2a_{21} + 6a_{31}h_1 = 0 = 2a_{20} \\
S_N^{(2)}(x_N) = 2a_{2N} = 0
\end{aligned} \tag{10}$$

где коэффициент $a_{20} = 0$ введен для единообразия записи. Из последнего уравнения (10) выразим a_{3n}:

$$a_{3n} = \frac{a_{2,n-1} - a_{2n}}{3h_n}, \quad n = 1,2,\ldots,N$$

Из второго уравнения (9) выразим a_{1n}:

$$a_{1n} = -\frac{h_n}{3}(a_{2,n-1} + 2a_{2n}) + \frac{f_{n-1} - f_n}{h_n}, \quad n = 1,2,\ldots,N$$

Подставляя выражения для a_{3n} и a_{1n} в третье уравнение системы (9) и сдвигая индекс n на единицу, получим систему линейных алгебраических уравнений для определения коэффициентов a_{2n}

$$h_n a_{2,n-1} + 2(h_n + h_{n+1})a_{2n} + h_{n+1}a_{2,n+1} = F_n, \quad n = 1,2,\dots,N-1 \qquad (11)$$

где $F_n = 3\left[\frac{f_{n-1} - f_n}{h_n} - \frac{f_n - f_{n+1}}{h_{n+1}}\right]$. Нетрудно видеть, что граничные условия на концах интервала (10) замыкают систему и ее можно записать в следующем матрично-векторном виде с трехдиагональной матрицей

$$A\vec{a}_2 = \vec{F},$$

$$A = \begin{pmatrix} 1 & 0 & 0\dots & & 0 \\ h_1 & 2(h_1+h_2) & h_2 \dots & & 0 \\ \dots & \dots & \dots & \dots & \\ 0 & \dots & h_n & 2(h_n+h_{n+1}) & h_{n+1} \quad \dots 0 \\ \dots & & \dots & \dots & \\ 0 & 0 & \dots & \dots & 1 \end{pmatrix}, \quad \vec{a}_2 = \begin{pmatrix} a_{20} \\ a_{21} \\ \dots \\ a_{2n} \\ \dots \\ a_{2N} \end{pmatrix}, \quad \vec{F} = \begin{pmatrix} 0 \\ F_1 \\ \dots \\ F_n \\ \dots \\ 0 \end{pmatrix}$$

Для решения такой системы используется метод прогонки, который рассмотрим в дальнейшем.

Если отрезок разбит на равные интервалы длинной h, то для оценки погрешности кубического сплайна имеет место следующая теорема.

Теорема. *Если* $f(x) \in C_4[x_0, x_N]$*, то интерполяционный сплайн* $S_3(x)$ *удовлетворяет неравенству*

$$\max |f^{(m)}(x) - S_3^{(m)}(x)| \le \frac{C \cdot h^{4-m}}{(k+1)!} \underbrace{\max}_{[x_0, x_N]} |f^{(4)}(x)|,$$

где $n = 1,2,\dots,N; \quad m = 0,1,2$*, C – постоянная, не зависящая от длины интервала.*

Из теоремы следует, что максимальная погрешность для кубического сплайна и его первой и второй производных есть $O(h^4), O(h^3), O(h^2)$ соответственно.

Широкое распространение интерполяции с помощью кубических сплайнов объясняется тем, что у него, в отличие от всех остальных интерполирующих функций оказывается минимальным интеграл от квадратов вторых производных, т.е.

$$\int_{x_0}^{x_N} \left(\frac{d^2 S_3(x)}{dx^2}\right)^2 dx = \min,$$

что придает ему определенные сглаживающие свойства.

Локальные сплайны (B-сплайны).

Описанные выше сплайны часто неудобны из-за своей нелокальности, т.е. значение сплайна в точке x зависит от значений $f(x_n)$ во всех узлах интерполяции. Особенно это приводит к затруднениям при приближении функций многих переменных. Кроме того, в случае когда значения функции в узлах таблицы заданы с погрешностью, интерполяция сплайнами может приводить к значительным погрешностям, причиной которых является требование интерполяции, т.е. совпадение сплайна в узлах таблицы со значениями интерполируемой функции. Поэтому часто задачу интерполяции решают с помощью так называемых локальных (аппроксимационных) сплайнов, при построении которых требования интерполяции заменяются на требования аппроксимации, т.е. некоторой близости значения сплайна в узлах таблицы к заданным значениям функции в этих узлах. Среди таких аппроксимирующих сплайнов наибольшее распространение получили т.н. B-сплайны.

Рассмотрим построение B- сплайна третьей степени для случая постоянного шага между узлами интерполяции на единичном отрезке (заметим, что любой отрезок легко приводится к единичному путем замены переменных). Основу этого сплайна составляет стандартный B- сплайн третьей степени от аргумента ξ, представляющий собой следующую комбинацию полиномов третьей степени:

$$B(\xi)=\begin{cases}\frac{2}{3}-\xi^2+\frac{1}{2}|\xi|^2, & |\xi|\le 1,\\ \frac{1}{6}(2-|\xi|)^3, & 1\le|\xi|\le 2,\\ 0, & |\xi|\ge 2\end{cases}$$

Локальный сплайн второго порядка точности $B_2(x)$ записывается в виде

$$B_2(x)=\sum_{n=-1}^{N+1} f_n B\left(\frac{x-n\cdot h}{h}\right),$$

где $f_{-1}=2f_0-f_1,\quad f_{N+1}=2f_N-f_{N-1}$, т.е. значения интерполируемой функции в дополнительных узлах доопределяются линейной экстраполяцией. Легко проверить, что при такой экстраполяции имеют место равенства

$$B_2(x_0)=f_0,\ B_2(x_N)=f_N.$$

Чтобы понять свойства сплайна, вычислим его значение в некотором внутреннем узле таблицы $x_k = k \cdot h$. Тогда

$$B_2(x_k) = \frac{2}{3} f_k + \frac{1}{6}[f_{k-1} + f_{k+1}] = f_k + \frac{h^2}{6} \frac{f_{k-1} - 2f_k + f_{k-1}}{h^2} = f_k + \frac{h^2}{6} f''(x_k) + O(h^4),$$

т.е. действительно имеет место сглаживание таблично заданной функции и при этом величина погрешности интерполяции имеет порядок $O(h^2)$. В общем же случае эта погрешность определяется следующим образом:

$$\underbrace{\max(x)}_{[0,1]} | B_2(x)^{(k)} - f^{(k)}(x) | \le C_2 \cdot h^{2-k}, \quad k \le 1, \; C_2 \text{- константа.}$$

Более точный B – сплайн четвертого порядка точности $B_4(x)$ записывается в виде

$$B_4(x) = \sum_{n=-1}^{N+1} \frac{8f_n - f_{n-1} - f_{n+1}}{6} B\left(\frac{x - n \cdot h}{h}\right),$$

$$\begin{aligned}
f_{-1} &= 4f_0 - 6f_1 + 4f_2 - f_3, \\
f_{-2} &= 10f_0 - 20f_1 + 15f_2 - 4f_3, \\
f_{N+1} &= 4f_N - 6f_{N-1} + 4f_{N-2} - f_{N-3}, \\
f_{N+2} &= 10f_N - 20f_{N-1} + 15f_{N-2} - 4f_{N-3},
\end{aligned}$$

При этом $B_4(x_0) = f_0, B_4(x_1) = f_1, \; B_4(x_N) = f_N, B_4(x_{N-1}) = f_{N-1}$. Погрешность интерполяции

$$\underbrace{\max(x)}_{[0,1]} | B_4(x)^{(k)} - f^{(k)}(x) | \le C_4 \cdot h^{4-k} \underbrace{\sup}_{[0,1]} | f^{(4)}(x) |, \quad k \le 2, \; C_4 \text{- константа.}$$

Построение эмпирических формул с использованием метода наименьших квадратов

Часто в прикладных задачах встречается ситуация, когда таблично заданную функцию $y^*(x)$, заданную в N узлах требуется аппроксимировать некоторой априорно заданной функцией

$$y = f(x, \alpha_1, \alpha_2, \ldots, \alpha_m),$$

содержащей ряд параметров $\alpha_1, \alpha_2, \ldots, \alpha_m$. Вид функциональной зависимости и

число параметров m известны из каких-либо дополнительных предположений, и требуется определить их численные значения. Если m > N, то в решении системы уравнений

$$y_i^* = f(x_i, \alpha_1, \alpha_2, \ldots, \alpha_m), \quad i = 1,2,\ldots,N$$

имеется m - N константный произвол, что дает бесконечное множество решений и интереса не представляет. В случае m = N это задача интерполяции, различные способы решения которой рассматривались выше. Если же m < N, то для определения $\alpha_1, \alpha_2, \ldots, \alpha_m$ имеем дело с переопределенной системой уравнений, одним из эффективных способов решения которой является метод наименьших квадратов.

Суть его состоит в следующем. Составляется функция

$$S_2(\alpha_1, \alpha_2, \ldots, \alpha_m) = \sum_{i=1}^{N} \left[y_i^* - f(x_i, \alpha_1, \alpha_2, \ldots, \alpha_m) \right]^2, \qquad (12)$$

являющаяся суммой квадратов невязок, и значения искомых параметров $\alpha_1, \alpha_2, \ldots, \alpha_m$ определяются исходя из требования минимума этой функции. Отсюда и название метода - *метод наименьших квадратов*.

Так как функция S_2 достигает абсолютного минимума внутри области изменения параметров $\alpha_1, \alpha_2, \ldots, \alpha_m$, то необходимым условием ее минимума является обращение в нуль частных производных первого порядка от S_2 по $\alpha_1, \alpha_2, \ldots, \alpha_m$:

$$\frac{\partial S_2}{\partial \alpha_k} = -2 \sum_{i=1}^{N} \left[y_i^* - f(x_i, \alpha_1, \alpha_2, \ldots, \alpha_m) \right] \frac{\partial f}{\partial \alpha_k} = 0. \qquad (13)$$

Таким образом, для определения $\alpha_1, \alpha_2, \ldots, \alpha_m$ получаем систему из m уравнений с m неизвестными.

В общем случае полученная система уравнений является нелинейной, если функция $f(x_i, \alpha_1, \alpha_2, \ldots, \alpha_m)$ являлась нелинейной по $\alpha_1, \alpha_2, \ldots, \alpha_m$ и ее решение представляет серьезную проблему. Поэтому чаще всего в качестве аппроксимирующей функции выбирают линейную по $\alpha_1, \alpha_2, \ldots, \alpha_m$ функцию вида:

$$f(x, \alpha_1, \alpha_2, \ldots, \alpha_m) = \sum_{j=1}^{m} \alpha_j \varphi_j(x),$$

где $\varphi_j(x)$ - некоторые выбранные базисные (линейно независимые) функции. В этом случае система уравнений (13) является системой линейных алгебраических уравнений относительно искомых коэффициентов $\alpha_1, \alpha_2, \ldots, \alpha_m$:

$$\sum_{i=1}^{N}\left[y_i^* - \sum_{j=1}^{m}\alpha_j\varphi_j(x_i)\right]\cdot\varphi_k(x_i) = 0. \qquad (14)$$

Матрица полученной системы является симметричной и положительно определенной.

Пример. Выберем в качестве базисных функций мономы. Тогда

$$f(x,\alpha_1,\alpha_2,\ldots,\alpha_m) = \sum_{j=1}^{m}\alpha_j x_j^j$$

и система (14) принимает вид

$$\sum_{i=1}^{N}\left[y_i^* - \sum_{j=1}^{m}\alpha_j x_j^j\right]\cdot x_i^k = 0$$

или

$$\begin{aligned}
&\alpha_1 N + \alpha_2\sum_{i=1}^{N}x_i + \alpha_3\sum_{i=1}^{N}x_i^2 + \ldots + \alpha_m\sum_{i=1}^{N}x_i^m = \sum_{i=1}^{N}y_i^*\\
&\alpha_1\sum_{i=1}^{N}x_i + \alpha_2\sum_{i=1}^{N}x_i^2 + \alpha_3\sum_{i=1}^{N}x_i^3 + \ldots + \alpha_m\sum_{i=1}^{N}x_i^{m+1} = \sum_{i=1}^{N}y_i^*x_i\\
&\alpha_1\sum_{i=1}^{N}x_i^2 + \alpha_2\sum_{i=1}^{N}x_i^3 + \alpha_3\sum_{i=1}^{N}x_i^4 + \ldots + \alpha_m\sum_{i=1}^{N}x_i^{m+2} = \sum_{i=1}^{N}y_i^*x_i^2\\
&\ldots\quad\ldots\quad\ldots\quad\ldots\quad\ldots\quad\ldots\quad\ldots\quad\ldots\quad\ldots\quad\ldots\quad\ldots\quad\ldots\\
&\alpha_1\sum_{i=1}^{N}x_i^m + \alpha_2\sum_{i=1}^{N}x_i^{m+1} + \alpha_3\sum_{i=1}^{N}x_i^{m+2} + \ldots + \alpha_m\sum_{i=1}^{N}x_i^{2m} = \sum_{i=1}^{N}y_i^*x_i^m
\end{aligned}$$

Определитель такой системы является определителем Грамма, не обращается в нуль и, следовательно, система всегда имеет единственное решение.

Численное интегрирование

Вычисление определенных интегралов с помощью численных методов сводится к построению т.н. квадратурных формул вида

$$\int_a^b f(x)dx \approx \sum_{j=0}^{j=N} q_j f(x_j),$$

где $f(x)$ - непрерывная на [a,b] функция, q_j, x_j - веса и узлы квадратурной формулы, причем $a \le x_j \le b$, $j = 0,1,\ldots,N$. Достаточно общий подход к построению квадратурных формул заключается в разбиении отрезка интегрирования [a,b] на N интервалов $[x_j, x_{j+1}]$ в общем случае переменной длины и в замене на каждом из них подинтегральной функции $f(x)$ какой-либо близкой к ней функцией $\varphi(x)$, интеграл от которой вычисляется аналитически. После этого, используя свойство аддитивности, интеграл на всем отрезке [a,b] вычисляется как сумма интегралов по всем этим интервалам. Очевидно, что погрешность построенной таким образом квадратурной формулы будет тем меньше, чем больше используется число интервалов N и чем точнее функция $\varphi(x)$ приближает подинтегральную функцию $f(x)$. Рассмотрим ряд наиболее употребительных формул численного интегрирования, основанных на изложенном выше подходе, полагая для простоты, что отрезок $[a,b]$ разбит на N равных интервалов длины h, где $h = \dfrac{b-a}{N}$.

Построение формул численного интегрирования базируется на обобщенной теореме о среднем.

Теорема. Пусть $f, g \in [a,b]$, причем $g(x) \ge 0$ на [a, b]. Тогда существует такая точка $\xi \in [a,b]$, что

$$\int_a^b f(x)g(x)dx = f(\xi)\int_a^b g(x)dx$$

Первая формула прямоугольников

Пусть $f(x) \in C_1[0,h]$, $h > 0$. Положим приближенно

$$\int_0^h f(x)dx \approx h f_0,$$

где $f_0 = f(0)$. Для всего отрезка интегрирования формула приобретает вид

$$\int_a^b f(x)dx \approx h\sum_{i=0}^{N-1} f(x_i). \qquad (1)$$

Найдем остаточный член, т.е. погрешность этой формулы. Пусть

$$F(x) = \int_0^x f(t)dt,\ F'(x) = f(x),\ F''(x) = f'(x).$$

Так как $F(0)=0,\ F'(0)=f_0$, то по формуле Тейлора с остаточным членом в форме Лагранжа имеем

$$F(h)=F(0)+hF'(0)+\frac{h^2}{2}F''(\xi),$$

где ξ - некоторая точка [0, h], $0<\xi<h$. Тогда

$$F(h)=\int_0^h f(x)dx=hf_0+\frac{h^2}{2}f'(\xi).$$

Для всего интервала интегрирования формула запишется:

$$\int_a^b f(x)dx=h\sum_{i=0}^{N-1}f_i+\frac{b-a}{2}hf'(\xi),\quad a<\xi<b,$$

т.е. на всем отрезке интегрирования погрешность первой формулы прямоугольников (1) имеет порядок O(h).

Вторая формула прямоугольников

Отличие этой формулы от предыдущей состоит лишь в том, что подынтегральное выражение вычисляется в правой точке интервала. Поэтому вывод ее полностью аналогичен предыдущему и можно окончательно записать

$$\int_a^b f(x)dx\approx h\sum_{i=1}^{N}f_i. \qquad (2)$$

Полученная формула, как и (1), имеет также первый порядок точности, т.е. O(h).

Третья формула прямоугольников

В этой формуле подынтегральная функция вычисляется в точке, расположенной в центре интервала. Поместим в эту точку начало координат, и пусть $f(x)\in C_2[-h/2,h/2]$. Положим приближенно

$$\int_{-h/2}^{h/2}f(x)dx\approx hf_0,$$

где $f_0=f(0)$. Для всего отрезка формула приобретает вид

$$\int_a^b f(x)dx\approx h\sum_{i=0}^{N-1}f(x_{i+1/2}). \qquad (3)$$

Найдем погрешность этой формулы. Положим, как и раньше,

$$F(x)=\int_0^x f(t)dt,\ F_{\pm1/2}=F(\pm h/2)$$

Тогда

$$\int_{-h/2}^{h/2} f(x)dx = F(h/2) - F(-h/2).$$

Разлагая F(x) в ряд Тейлора с остаточным членом в форме Лагранжа, имеем

$$F(\pm h/2) = \pm\frac{h}{2}f_0 + \frac{h^2}{8}f_0' \pm \frac{h^3}{48}f''(\xi_\pm),$$

где $\xi_+ \in (0, h/2)$, $\xi_- \in (-h/2, 0)$. Тогда

$$\int_{-h/2}^{h/2} f(x)dx = h f_0 + \frac{h^3}{24}f''(\xi), \quad |\xi| < \frac{h}{2}.$$

Для всего интервала интегрирования [a, b] формула запишется

$$\int_a^b f(x)dx = h\sum_{i=0}^{N-1} f_{i+1/2} + \frac{b-a}{24}h^2 f''(\xi), \quad a < \xi < b \ ,$$

где $f_{i+1/2} = f(x_i + h/2)$. Таким образом, формула (3) имеет уже второй порядок точности $O(h^2)$.

Формула трапеций

Пусть $f(x) \in C_2[0,h]$. Заменим подынтегральную функцию на этом интервале прямой, проходящей через точки $(0, f_0)$ и (h, f_1). Интегрируя ее, получим

$$\int_0^h f(x)dx \approx h\frac{f_0 + f_1}{2},$$

где $f_0 = f(0)$, $f_1 = f(h)$. Заметим, что полученная формула есть площадь трапеции, откуда и ее название. Для всего отрезка формула приобретает вид

$$\int_a^b f(x)dx \approx h\sum_{i=0}^{N-1}\frac{f_i + f_{i+1}}{2}. \qquad (4)$$

Определим погрешность этой формулы на шаге, используя в разложении Тейлора представление остаточного члена в точке $(0, f_0)$ в интегральной форме $R_n(x) = \frac{1}{n!}\int_0^x (x-t)^n f^{(n+1)}(t)dt$:

$$F(h) = F_1 = F(0) + h F'(0) + \frac{h^2}{2}F''(0) + \frac{1}{2}\int_0^h (h-t)^2 F'''(t)dt;$$

или

$$F_1 = hf_0 + \frac{h^2}{2}f_0' + \frac{1}{2}\int_0^h (h-t)^2 f''(t)dt; \qquad (4a)$$

Запишем аналогичное разложение для подинтегральной функции

$$f_1 = f_0 + h f_0' + \int_0^h (h - t) f''(t) dt; \qquad (5)$$

Исключая из (4a) и (5) величину f_0' и учитывая, что $F_1 = \int_0^h f(x)dx$, получим

$$\int_0^h f(x)dx = h\frac{f_0 + f_1}{2} - \frac{1}{2}\int_0^h (h - t)\, t\, f''(t)dt.$$

Далее, принимая во внимание, что (h-t)t > 0, по теореме о среднем имеем

$$\int_0^h (h - t)\, t\, f''(t)dt = f''(\xi)\int_0^h (h - t)\, t dt = \frac{h^3}{6} f''(\xi), \quad \xi \in (0, h).$$

Тогда на интервале [0, h] формула трапеций запишется

$$\int_0^h f(x)dx = h\frac{f_0 + f_1}{2} - \frac{h^3}{6} f''(\xi)$$

Производя суммирование по всем интервалам, для всего отрезка [a, b] получим

$$\int_a^b f(x)dx = h\sum_{i=0}^{N-1} \frac{f_i + f_{i+1}}{2} - \frac{b - a}{12} h^2 f''(\xi), \quad a < \xi < b.$$

Следовательно, формула трапеций (4) имеет второй порядок точности $O(h^2)$ и менее точна, чем формула (3). Заметим, что (4) может быть получена как полусумма (1) и (2) (докажите самостоятельно).

Формула Симпсона

Разобьем весь отрезок интегрирования [a, b] на интервалы длиной 2h каждый и поместим начало координат в середину такого интервала. Пусть $f(x) \in C_4[-h, h]$. Заменим подынтегральную функцию параболой, проходящей через точки $(-h, f_{-1}), (0, f_0), (h, f_1)$.

$$y = f_0 + \frac{f_1 - f_0}{2h} x + \frac{f_{-1} - 2f_0 + f_1}{2h^2} x^2$$

Произведя ее интегрирование (как интегрирование полинома второй степени), запишем формулу Симпсона

$$\int_{-h}^{h} y\,dx = \frac{h}{3}(f_{-1} + 4f_0 + f_1).$$

Для всего интервала

$$\int_a^b f(x)\,dx \approx \frac{h}{3}\sum_{i=1(2)}^{N-1}(f_{i-1} + 4f_i + f_{i+1}), \qquad (6)$$

где суммирование по индексу i производится с шагом 2.

Оценка погрешности полученной формулы проводится аналогично тому, как это было проведено для метода трапеций. Результат здесь следующий:

$$\int_a^b f(x)\,dx = \frac{h}{3}\sum_{i=1(2)}^{N-1}(f_{i-1} + 4f_i + f_{i+1}) - \frac{(b-a)}{180}h^4 f^{(4)}(\zeta), \quad \zeta \in (a,b),$$

т.е. формула (6) имеет четвертый порядок точности $O(h^4)$.

Метод Монте-Карло

Статистический метод вычисления интегралов основан на вычислении математических ожиданий некоторых случайных величин, поскольку эти математические ожидания представляют собой обычные интегралы. Пусть $z = f(\xi)$ неслучайная функция от непрерывного случайного аргумента ξ определена на отрезке $[0,1]$. Тогда ее математическое ожидание $M[z]$ есть

$$M[z] = I = \int_0^1 f(\xi)p(\xi)d\xi,$$

где $p(\xi)$ - плотность вероятности распределения случайной величины на данном отрезке. Если случайная величина ξ распределена равномерно, то $p(\xi) \equiv 1$ и тогда выражение для $M[z]$ есть обыкновенный интеграл. Если ξ является дискретной равномерно распределенной случайной величиной, т.е. может принимать значения $\xi_1, \xi_2, \ldots \xi_n$, то в этом случае $M[z] = \bar{z}_n = \frac{1}{n}\sum_{i=1}^{n} f(\xi_i)$ и тогда приближенно

$$\int_0^1 f(\xi)d\xi \approx \frac{1}{n}\sum_{i=1}^{n} f(\xi_i).$$

Согласно центральной предельной теореме разность между левой и правой частями этого приближенного равенства стремится к нулю по вероятности при $n \to \infty$.

Если дисперсия $D[z] = \sigma^2 = M[(M[z] - z)^2]$ конечная, то при достаточно большом числе испытаний $n\ (n > 10)$ имеет место правило «трех сигм»

$$P(|I - \bar{z}_n| < \frac{3\sigma}{\sqrt{n}}) \approx 0.997,$$

т.е. неравенство $|I - \bar{z}_n| < \frac{3\sigma}{\sqrt{n}}$ выполняется с вероятностью 0.997. Если σ неизвестно, то при $n>10$ ее можно приближенно оценить по формуле

$$\sigma \approx \sqrt{\frac{1}{n}\sum_{i=1}^{n}(z_i - \bar{z}_n)^2}\ .$$

Таким образом, если ξ - равномерно распределенная на [0,1] дискретная случайная величина, то

$$\int_0^1 f(x)dx \approx \bar{z}_n = \frac{1}{n}\sum_{i=1}^{n} f(\xi_i),$$

где ξ_i независимые реализации случайной величины ξ.

Кратные интегралы методом Монте-Карло вычисляются аналогично

$$\int_0^1\int_0^1 f(x,y)dxdy \approx \frac{1}{n}\sum_{i=1}^{n} f(\eta_i,\mu_i),$$

где η_i, μ_i - независимые реализации случайных величин, равномерно распределенных на $[0,1]$. Заметим, что произвольный отрезок интегрирования $[a, b]$ приводится к единичному отрезку простым линейным преобразованием $\zeta = (x - a)/(b - a)$.

Если погрешность квадратурной формулы оценить как $\varepsilon \approx C_1 N^{-m}$, где m - точность формулы, N – число узлов, то для заданной точности ε требуется $N \approx C_2\varepsilon^{-1/m}$ итераций, а для k –мерного интеграла $N \approx C_2\varepsilon^{-k/m}$. В методе же Монте-Карло $\varepsilon \approx \frac{3\sigma}{\sqrt{n}}$, т.е. требуемое число испытаний $n \approx C_3\varepsilon^{-2}$ и не зависит от кратности интеграла. Поэтому, несмотря на более низкую точность, методы Монте-Карло широко применяются при вычислении интегралов высокой кратности.

Основным недостатком метода Монте-Карло является вероятностный характер результата, т.е. отсутствие строгих, стремящихся к нулю при $n \to \infty$ оценок погрешности.

Вычисления кратных интегралов

Для вычисления кратных интегралов используются те же методы, что и были изложены выше для вычисления определенных интегралов. Для простоты рассмотрим только случай, когда областью интегрирования является прямоугольник или параллелепипед и вся область разбивается на аналогичные подобласти одинаковых размеров. Так, метод прямоугольников для двойных интегралов может быть записан в виде:

$$\int_a^b\int_c^d f(x,y)dxdy \approx h_x h_y \sum_{i=0}^{N_x-1}\sum_{j=0}^{N_y-1} f(x_i, y_j),$$

где $h_x = (b-a)/N_x; h_y = (d-c)/N_y;\ x_i = i\cdot h_x; y_j = j\cdot h_y$. Совершенно аналогично запишется формула для тройного интеграла

$$\int_a^b\int_c^d\int_m^n f(x,y,z)dxdydz \approx h_x h_y h_z \sum_{i=0}^{N_x-1}\sum_{j=0}^{N_y-1}\sum_{k=0}^{N_z-1} f(x_i, y_j, z_k)$$

где $h_x = (b-a)/N_x; h_y = (d-c)/N_y; h_z = (m-n)/N_z;$

$x_i = i\cdot h_x; y_j = j\cdot h_y; z_k = k\cdot h_z$.

Формула трапеций может быть представлена в виде:

$$\int_a^b\int_c^d f(x,y)dxdy \approx h_x h_y \sum_{i=0}^{N_x-1}\sum_{j=0}^{N_y-1}[f(x_i, y_j) + f(x_{i+1}, y_j) + f(x_i, y_{j+1}) + f(x_{i+1}, y_{j+1})]/4$$

и

$$\int_a^b\int_c^d\int_m^n f(x,y,z)dxdydz \approx h_x h_y h_z \sum_{i=0}^{N_x-1}\sum_{j=0}^{N_y-1}\sum_{k=0}^{N_z-1}[f(x_i, y_j, z_k) + f(x_{i+1}, y_j, z_k) + f(x_i, y_{j+1}, z_k) +$$
$$+ f(x_i, y_j, z_{k+1}) + f(x_{i+1}, y_{j+1}, z_k) + f(x_{i+1}, y_j, z_{k+1}) + f(x_i, y_{j+1}, z_{k+1}) + f(x_{i+1}, y_{j+1}, z_{k+1})]/8$$

Параллельное вычисление интегралов

Для понимания схемы вычисления определенного интеграла с помощью кластера следует воспользоваться свойством его аддитивности

$$\int_a^b f(x)dx = \sum_{i=0}^{p-1}\int_{a_i}^{b_i} f(x)dx$$

где $b_1 = a_i + d, d = (b-a)/p,$ p - число процессоров (не считая корневого). Тогда с учетом сказанного схема вычисления будет иметь вид, представленный на рис. 6.

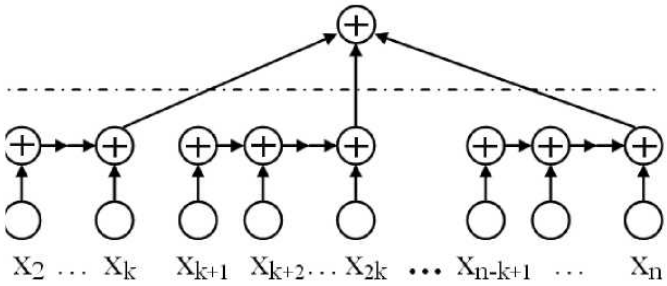

Рис. 6. Схема вычисления определенного интеграла

По данной схеме суммируется n значений с помощью $p + 1$ узла кластера. Каждому процессору, кроме корневого, достается по $k = n/p$ слагаемых. После нахождения всех последовательных сумм (ниже пунктирной линии) корневой процесс выполняет редукцию по сумме этих значений. Если корневой узел кластера является многопроцессорным, то он может вместо редукции выполнить сложение по каскадной схеме. При вычислении кратных интегралов свойство аддитивности следует применять к самому внешнему интегралу. В этом случае увеличивается вычислительная нагрузка на узлы кластера, что приводит к уменьшению относительного времени «накладных» расходов, связанных с операциями обмена, т.е. отношение времени операций обмена к времени счета.

Числовые ряды

К этому разделу относятся задачи определения различных констант (числа π, числа и др.), представляемых в виде числового ряда, нахождение определенных интегралов, суммирование большого количества констант и подобные этому задачи, определение всех частичных сумм, поиска глобального максимума, минимума, вычисления значений логических выражений над длинными последовательностями величин и т. п.

Алгоритмы, позволяющие выполнять параллельное суммирование с помощью многопроцессорных систем с общей памятью, могут выполняться с помощью каскадной схемы суммирования, модифицированной каскадной схемы.

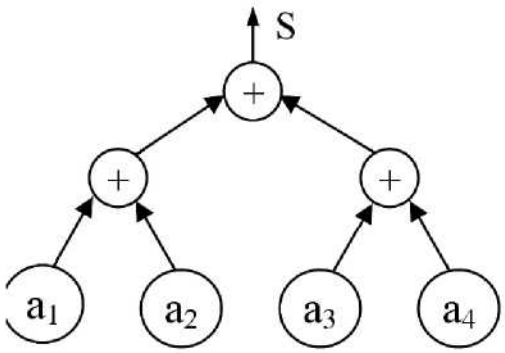

Рис. 1. Каскадная схема

Каскадная схема суммирования имеет вид бинарного дерева, в листьях которого записаны члены ряда, а во всех остальных узлах - операции сложения.

Например, чтобы найти сумму $S = a1 + A2 + a3 + a4$, потребуется схема, представленная на рис. 1.

Эффективность рассмотренной схемы определяется как

$$S_p = T_1 / T_p = (n-1)/\log_2 n,$$

где T_1 - трудоемкость алгоритма для одного процессора, T_p - для сети из p процессоров.

Эффективность использования процессоров вычисляется по формуле

$$E_p = T_1/pT_p = (n-1)/((n/2)\log_2 n) .$$

Для данной схемы $E\,p \to 0$ при $n \to \infty$.

Если допустить, что каскадная схема вычисления выполняется с помощью кластера, каждый узел которого делает одну операцию сложения, то время на пересылку промежуточных результатов будет неизмеримо больше времени вычисления. Применение кластера по такой схеме оказывается нецелесообразным. Однако при использовании многопроцессорного суперкомпьютера с общей памятью каскадная схема является очень эффективной. Ограничением будет являться максимально допустимое на современном этапе количеством процессоров в одном суперкомпьютере - примерно 130 тыс. И то это в предположении, что все процессоры компьютера можно задействовать для вычислений по каскадной схеме. Решение задачи суммирования большого количества слагаемых, измеряемого сотнями тысяч и миллионами чисел, потребует использовать другие вычислительные схемы.

В этом случае для решения задачи с помощью многопроцессорной вычислительной сети с распределенной памятью (кластера) можно предложить, например, использовать каскадные схемы в комбинации с операцией редукции по сумме (см. рис. 2), если многопроцессорные узлы кластера реализуют кас-

кадную схему, а редукцию по сумме выполнит обычный последовательный процессор.

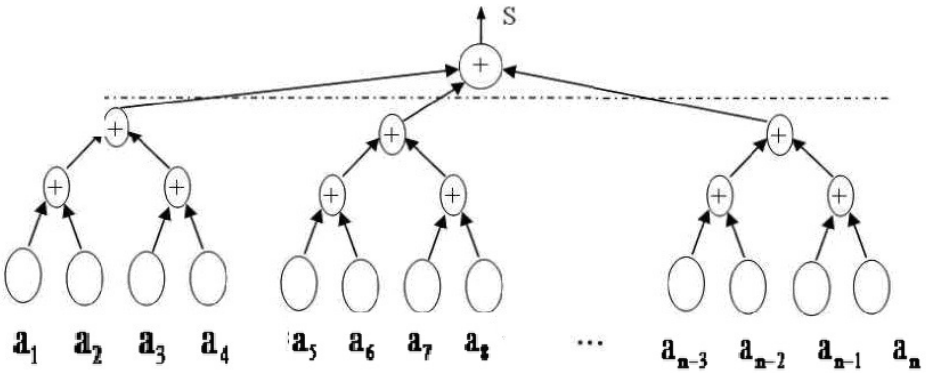

Рис.2. Каскадная схема с редукцией по сумме

Схема обработки для такой реализации получается, если лес, состоящий из бинарных деревьев, соединить связями от корней образующих этот лес деревьев с новым узлом - корневым узлом сформированного дерева. Фрагменты части рисунка, расположенные ниже штрих-пунктирной линии, выполняются на многопроцессорных узлах кластера. Фрагмент, который расположен выше штрих-пунктирной линии, соответствует однопроцессорному корневому узлу.

Аналогом этой схемы является схема, в которой вместо редукции выполняется обычное последовательное суммирование (рис. 3).

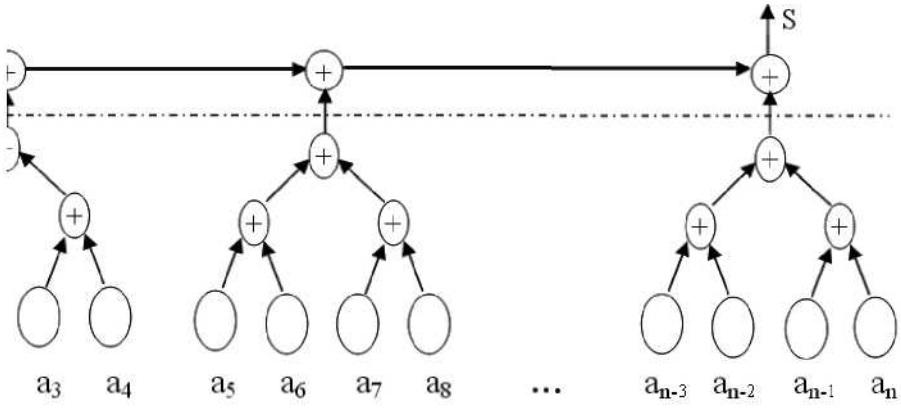

Рис. 3. Каскадно-последовательная схема

На этом рисунке фрагменты части рисунка, расположенные ниже штрих-пунктирной линии, также выполняются на многопроцессорных узлах кластера. Фрагмент, который расположен выше штрих-пунктирной линии, соответствует однопроцессорному корневому узлу. Однако вычислительная трудо-

емкость данной схемы примерно такая же, как и у схемы на рис. 2, так как редукция по сумме выполняется на обычном последовательном компьютере.

Выбор одной из двух схем зависит от программного обеспечения, используемого для последовательного суммирования, от эффективности его реализации в конкретной программной системе. Если же в распоряжении имеется кластер, позволяющий лишь на одном узле реализовать каскадную схему, а на других - только последовательное вычисление, то можно использовать другую схему (см. рис. 4).

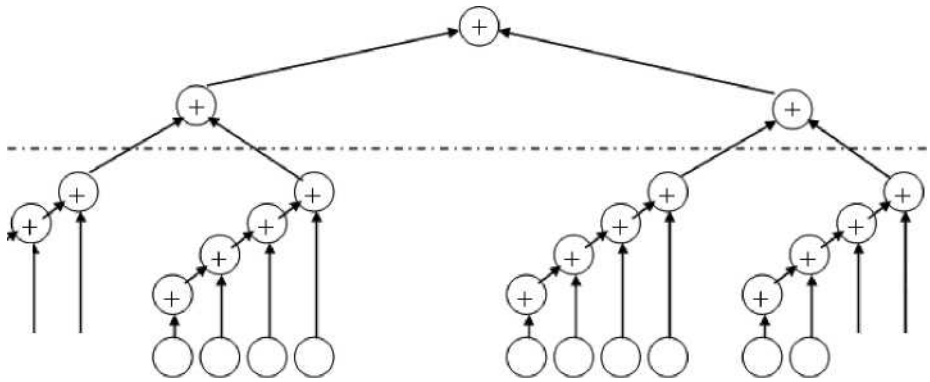

Рис. 4. Последовательно-каскадная схема

Фрагменты части рисунка, расположенные ниже штрих-пунктирной линии, выполняются на однопроцессорных узлах кластера. Фрагмент, который расположен выше штрих-пунктирной линии, соответствует многопроцессорному корневому узлу.

Последний вариант (рис. 5) соответствует случаю, когда все узлы кластера, включая корневой, являются многопроцессорными. Он позволяет ускорить вычисление результата на заключительном этапе объединения частичных результатов, а также использовать узлы с ограниченным количеством процессоров в каждом из них.

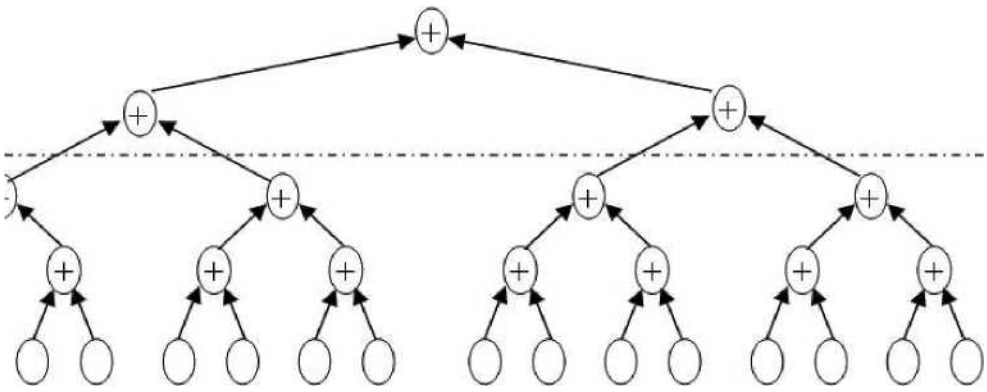

Рис. 5. Каскадно-каскадная схема

Для вычисления суммы числового ряда, например числа π

$$\pi \approx \sum_{i=1}^{n} x_i, \quad x_i = \frac{4}{n + n[(i - 0.5)/n]^2}$$

можно воспользоваться схемой, аналогичной той, которая представлена на рис. 6

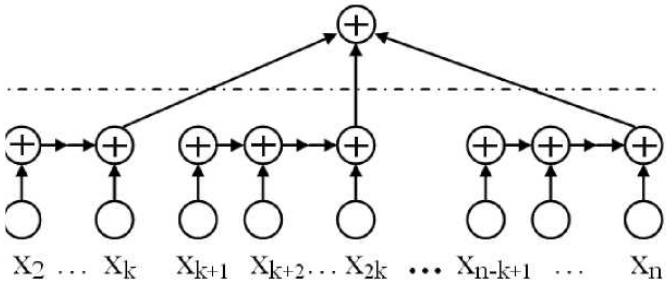

Рис. 6. Схема вычисления определенного интеграла

организовав в ней порядок группирования слагаемых следующим образом. Первый узел кластера будет вычислять сумму $S_1 = x_1 + x_{k+1} + \cdots + x_{n-k+1}$, второй процессор найдет сумму $S_2 = x_2 + x_{k+2} + \cdots + x_{n-k+2}$, . . , p-й процессор найдет $S_p = x_k + x_{2k} + \cdots + x_n$. После этого корневой процессор выполнит редукцию и вычислит $\pi = S = S_1 + S_2 + \cdots + S_p$. Такой порядок группировки слагаемых легче программируется, позволяет более равномерно загрузить вычислительной работой процессоры в том случае, если сложность вычисления последних членов ряда (если они представлены в виде некоторого выражения) существенно отличается от сложности вычисления начальных.

Таким образом, идея распараллеливания вычислений при подсчете значений числовых рядов заключается в том, что все слагаемые делятся некоторым образом в зависимости от метода на определенное количество промежуточных групп, после чего подсчет сумм в группах выполняется одновременно разными процессорами вычислительной системы. Затем значения частичных сумм складываются для получения конечного результата. Способы вычисления сумм в промежуточных группах и в конечной группе зависят от используемого метода.

Эту же идею можно использовать при поиске максимальных, минимальных значений среди последовательности величин, подсчете значений логических выражений с операциями И, ИЛИ и другими бинарными функциями, как и с функциями от большего числа аргументов.

Комбинаторные задачи

Среди задач, относящихся к данной группе, можно обратить внимание, в первую очередь, на параллельные методы сортировки и поиска данных.

Суть многих параллельных алгоритмов сортировки состоит в том, что упорядочиваемые данные разбиваются на какие-либо части в зависимости от метода, каждая из которых сортируется по отдельности разными процессорами. Затем упорядоченные части объединяются некоторым образом, опять же в зависимости от метода. Эти два этапа могут повторяться необходимое число раз.

Из методов параллельной сортировки в качестве примеров рассмотрим наиболее типичных представителей - метод пузырька.

Для параллельной реализации метода пузырьковой сортировки используется модификация, называемая методом чет-нечетной перестановки. В простейшем случае сортировка заключается в чередовании двух этапов. На первом этапе массив разбивается на пары (a0, a1), (a2, a3), (a4, a5), . . ., (a_{n-2}, a_{n-1}), начинающиеся с четных индексов. В каждой паре слева помещается наименьший элемент из двух, справа - наибольший из них. На втором этапе аналогичная обработка выполняется в парах (a1, a_2), (a_3, a_4), (a_5, a_6),..., (a_{n-3}, a_{n-2}), начинающихся с нечетных индексов. Максимум через n повторений массив оказывается упорядоченным. Иллюстрация приведена на рис. 7. Стрелками обозначены пары элементов, которые могут поменяться местами.

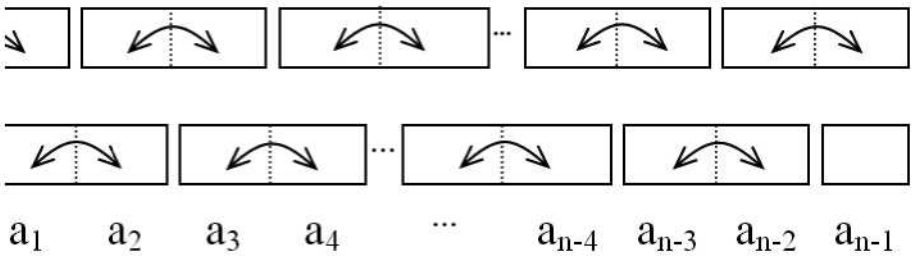

Рис.7. Чет-нечетная перестановка

В рассмотренном методе подразумевается, что число процессоров равно числу *n* элементов массива *A*. Такая реализация может быть использована в многопроцессорном компьютере, но для кластера рабочих станций она не годится, так как затраты на пересылку соседних элементов будут превышать время на выполнение операций по сортировке.

Чтобы приспособить метод к реализации в распределенной вычислительной системе, состоящей из *p* процессоров, все *n* сортируемых элементов делятся на части, содержащие по *n/p* элементов. Каждая из частей рассылается в локальную память соответствующего процессора и предварительно сортируется в нем каким-либо быстрым методом. Затем чередуются два этапа:

1) Обмен фрагментами, состоящими из *n/p* элементов, между соседними процессорами (рис. 8). Этот обмен выполняется по очереди:

1а) внутри четных пар процессоров с номерами (0,1), (2,3), (4,5) ...,
1б) внутри нечетных пар процессоров с номерами (1,2), (3,4), (5,6) ...

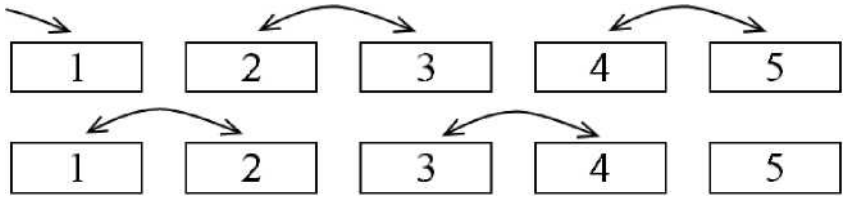

Рис. 8. Порядок обмена фрагментами в чет-нечетной сортировке

В каждом процессоре выполняется операция «сравнить и разделить» (compare-split). При ее выполнении процессор объединяет два упорядоченных фрагмента в один упорядоченный удвоенной длины *2n/p.* Для этого можно использовать метод сортировки слиянием («слить» два в один). Затем левый процессор пары оставляет у себя левую половину результата, а правый процессор - правую половину (рис. 9).

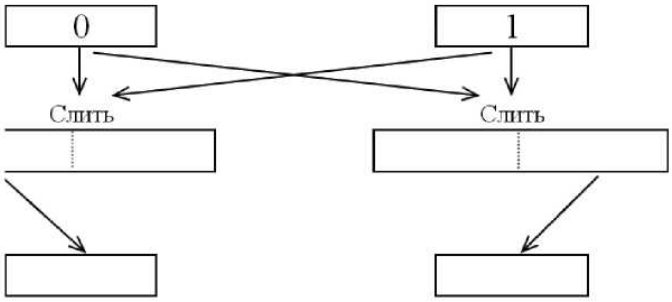

Рис. 9. Операция «сравнить и разделить»

Решение нелинейных алгебраических уравнений и их систем

В этом разделе нам понадобятся некоторые сведения и понятия из функционального анализа.

1. Метрическое пространство. Пусть M - множество элементов f, g, r, произвольной природы. Множество M называется метрическим пространством, если любой паре его элементов поставлено в соответствие неотрицательное

число $\rho(f,g)$, называемое расстоянием (метрикой) между элементами f и g, которое удовлетворяет аксиомам расстояния:

1) $\rho(f,g)=0$ тогда и только тогда, когда $f=g$;

2) $\rho(f,g)=\rho(g,f), \quad \forall f,g\in M$;

3) $\rho(f,g)\le\rho(f,b)+\rho(b,g), \quad \forall f,g\in M$ (неравенство треугольника).

Примером метрического пространства является евклидово пространство n – мерных векторов R^n с расстоянием

$$\rho(\vec{x},\vec{y})=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}\ ,\ \vec{x}=(x_1,x_2,\dots,x_n),\quad \vec{y}=(y_1,y_2,\dots,y_n)$$

Множество M называется линейным пространством, если в нем определены операции сложения и умножения на действительные числа, результатами которых являются новые элементы, не выходящие за пределы M и удовлетворяющие условиям:

1) $(f+g)+b=f+(g+b)$ - ассоциативность сложения;

2) $f+g=g+f$ - коммутативность сложения;

3) существует нулевой элемент $\Theta\in M$, т.е. $f+\Theta=f,\ \forall f\in M$;

4) $f\Theta=\Theta,\ \forall f\in M$;

5) $(\alpha+\beta)f=\alpha f+\beta f$;

6) $\alpha(f+g)=\alpha f+\alpha g$;

7) $\alpha(\beta f)=(\alpha\beta)f$;

8) $1\cdot f=f,\quad \alpha,\beta$ - действительные числа.

Система элементов $f_1,f_2,\dots,f_n$ линейного пространства M называется линейно зависимой, если существуют такие числа $\alpha_1,\alpha_2,\dots,\alpha_n$, не равные одновременно нулю, что

$$\alpha_1f_1+\alpha_2f_2+\dots+\alpha_nf_n=\Theta$$

Если данное равенство возможно только при $\alpha_1=\alpha_2=\dots=\alpha_n=0$, то система элементов $f_1,f_2,\dots,f_n$ называется линейно зависимой. Множество M называется линейным нормированным пространством, если оно линейно и каждому элементу $f\in M$ поставлено в соответствие действительное число $\|f\|\ge 0$, называемое нормой и удовлетворяющее аксиомам нормы:

1) $\|f\|\ge 0$, причем $\|f\|=0$ тогда и только тогда, когда $f=\Theta$;

2) $\|\alpha f\|=|\alpha|\cdot\|f\|$ для любого действительного α;

3) $\|f+g\|\le\|f\|+\|g\|, \quad \forall f,g\in M$ (неравенство треугольника).

Любое нормированное линейное пространство является одновременно метрическим пространством с расстоянием (метрикой) $\rho(f,g)=\|f-g\|$. Действительно, $\rho(f,g)=\|f-g\|\ge 0$;

$$\rho(f,g)=\|f-g\|=\|(f-g)+(g-b)\|\le\|f-g\|+\|g-b\|=\rho(f,g)+\rho(g,b).$$

В одном и том же линейном пространстве норму можно вводить различными способами:

$$\|f\|_C = \underbrace{\max}_{[a,b]} |f(x)| \; ; \; \|f\|_{C_n} = \underbrace{\max}_{[a,b]} \{|f(x)|, |f'(x), |f''(x)|, \ldots, |f^{(n)}(x)|\};$$

$$\|f\|_{L_2} = \left(\int_a^b f^2(x)dx \right)^{1/2}$$

Соответственно изменяется и метрика:

$$\rho(f,g) = \|f-g\|_C = \underbrace{\max}_{[a,b]} |f(x)-g(x)| ; \; \rho(f,g) = \|f-g\|_{L_2} = \left(\int_a^b [f(x)-g(x)]^2 dx \right)^{1/2}.$$

В дальнейшем будем пользоваться в основном нормами, введенными следующим образом

$$\|\vec{x}\|_1 = \underbrace{\max}_{1 \le j \le n} |x_j| \qquad \text{(первая норма);}$$

$$\|\vec{x}\|_2 = \left(\sum_{j=1}^n x_j^2 \right)^{1/2} \qquad \text{(вторая норма),}$$

которые, как легко проверить, удовлетворяют всем трем аксиомам норм. Соотношение между этими нормами

$$\|\vec{x}\|_1 \le \|\vec{x}\|_2 \le \sqrt{n}\|\vec{x}\|_1 .$$

Пусть имеется два пространства R_1, R_2 и каждому элементу $x \in R_1$ поставлен в соответствие по какому-то закону элемент $y \in R_2$. Тогда говорят, что задан оператор $y = A(x)$ с областью определения R_1 и областью значений, принадлежащей R_2. Если $R_1 = R_2$ и тогда оператор $A(x)$ отображает пространство R_1 в себя.

В метрических и нормированных пространствах определено понятие сходимости последовательности. Последовательность элементов $\{f_n\}$ метрического (нормированного) пространства сходится к его элементу f, если $\rho(f, f_n) \to 0$ при $n \to \infty$. Такое пространство называется полным. Последовательность $\{f_n\}$ называется фундаментальной, если $\rho(f_i, f_j) \le \varepsilon(N)$, где $i, j \ge N$ и $\varepsilon(N) \to 0$ при $N \to \infty$ или $\|f_i - f_j\| \le \varepsilon(N)$.

Теоретические основы методов

Будем рассматривать некоторые численные методы приближенного решения уравнений вида f(x) = 0 и систем уравнений вида

$$f_i(x_1, x_2, \ldots, x_n) = 0, \quad i = 1,2,\ldots,n .$$

Большинство методов отыскания корней уравнений предполагает, что заранее известны некоторые достаточно малые окрестности, в каждой из которых имеемся только один корень уравнения. Принимая за начальное приближение корня любую точку этой окрестности, с помощью этих методов можно вычислить искомый корень с заданной точностью.

Таким образом, при численном решении нелинейных уравнений возникает две задачи:

- **задача отделения корней**, т.е. отыскания достаточно малых областей, в каждой из которых содержится один и только один корень;

- **вычисление корня с заданной точностью**, если известно его начальное приближение в этой области.

Первая задача значительно сложнее второй, и до настоящего времени не существует достаточно эффективного метода отделения корней в общем случае. Особенно большие трудности здесь возникают для систем уравнений. Обычно приближенное представление о положении корней дает физический аспект решаемой задачи. Часто полезным оказывается графическое представление поведения функции y = f(x). В дальнейшем рассматривается только вторая задача.

Для решения нелинейного уравнения $f(x) = 0$ разработано много итерационных методов. Их сущность заключается в следующем. Пусть известна достаточно малая область, содержащая единственный корень $x = \alpha$ уравнения f(x) = 0. В этой области выбирается точка x_0 - начальное приближение корня, и с помощью некоторого рекуррентного соотношения $x_{n+1} = \varphi_n(x_n)$ строится последовательность точек $x_1, x_2, \ldots, x_n, \ldots$, сходящаяся к $x = \alpha$. Сходимость обеспечивается соответствующим выбором функции φ_n и начального приближения x_0. Чаще всего по функции f(x) строят функцию $\varphi(x)$ такую, что искомый корень $x = \alpha$ является и корнем уравнения $x = \varphi(x)$, и затем строят последовательность $x_{n+1} = \varphi(x_n)$ $n = 0,1,2,\ldots$ исходя из некоторого начального приближения x_0.

Для исследования сходимости итерационных процессов широко применяется принцип сжатых отображений, который может быть сформулирован следующим образом. Пусть R - некоторое метрическое пространство, а Ax - некоторый оператор, определенный на этом пространстве. Будем говорить, что этот

оператор осуществляет сжатое отображение R в себя, если существует такое положительное число. $0 < \alpha < 1$, что имеет место неравенство

$$\rho(Ax, Ay) \le \alpha \rho(x, y),$$

т.е. оператор А сближает элементы.
Рассмотрим применение этого принципа к исследованию сходимости итерационного метода решения уравнения $x = \varphi(x)$ при некоторых ограничениях на функцию $\varphi(x)$. Предположим, что это уравнение имеет корень $x = \beta$ и в круге $R\{|x - \beta| \le r\}$ радиуса r функция $\varphi(x)$ удовлетворяет условию Липшица с константой К (константа Липшица):

$$|\varphi(x_1) - \varphi(x_2)| \le K|x_1 - x_2|, \quad \forall x_1, x_2 \in R$$

Теорема. *Каково бы ни было* $x_0 \in R$, *последовательность* $x_{n+1} = \varphi(x_n), \quad n = 0,1,2,\ldots$ *сходится к* β, *если только* $\varphi(x)$ *в круге* R *удовлетворяет условию Липшица с константой* $K < 1$, *причем скорость сходимости характеризуется неравенством* $|x_{n+1} - \beta| \le K^n |x_0 - \beta|$.

Доказательство. Совокупность точек круга R образует полное метрическое пространство, если ввести метрику $\rho(x, y) = |x - y|$. Если $x \in R$, то $y = \varphi(x)$ тоже принадлежит R, так как $\rho(y, \beta) = |y - \beta| = |\varphi(x) - \varphi(\beta)| \le K|x - \beta| \le r$.
Отображение, определяемое функцией $\varphi(x)$ есть сжатое отображение R в себя, так как для любых $x, y \in R$

$$\rho(\varphi(x), \varphi(y)) = |\varphi(x) - \varphi(y)| \le K|x - y| = K\rho(x, y), \quad K < 1.$$

Поэтому по принципу сжатых отображений в R существует одна и только одна неподвижная точка, но эта точка $x = \beta$. Она может быть получена как предел последовательности $x_{n+1} = \varphi(x_n)$ при любом $x_0 \in R$. Используя условие Липшица, имеем

$$|x_{n+1} - \beta| = |\varphi(x_n) - \varphi(\beta)| \le K|x_n - \beta|,$$

т.е.

$$|x_{n+1} - \beta| \le K|x_n - \beta| \le K^2|x_{n-1} - \beta| \le \ldots \le K^n|x_0 - \beta|. \qquad (3.1)$$

Таким образом, $\{x_n\}$ сходится к β со скоростью геометрической прогрессии со знаменателем K.

Чаще всего функция $\varphi(x)$ имеет производные, и условие Липшица с константой $K < 1$ будет иметь место, если в некоторой окрестности корня $x = \beta$

функция $\varphi(x)$ имеет производную $|\varphi'(x)| \le K < 1$. Это неравенство и может служить условием выбора области начального x_0 приближения итерационного процесса. На рис. 10.1 – 10.4 показано, что последовательность $x_{n+1} = \varphi(x_n)$ сходится, если начальное приближение x_0 выбирается из области ее сходимости, где $|\varphi'(x)| < 1$, т.е. $-1 < \varphi'(x) < 1$ и расходится, если $|\varphi'(x)| > 1$. Здесь итерационный процесс представлен в виде системы двух уравнений $\begin{cases} y_n = \varphi(x_n) \\ x_{n+1} = y_n \end{cases}$, которая полностью эквивалентна исходной последовательности $x_{n+1} = \varphi(x_n)$. Стрелками указаны последовательности вычислений очередных приближений к корню уравнения $x = \beta$.

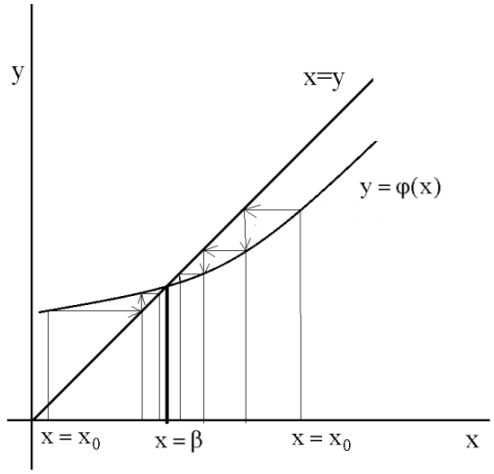

Рис.10.1 Сходимость последовательности $x_{n+1} = \varphi(x_n)$ при $\varphi'(x) < 1$.

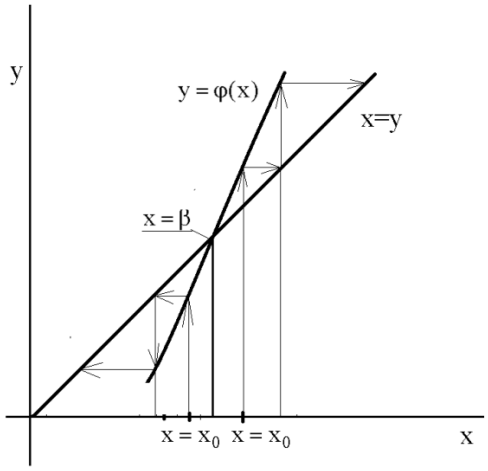

Рис.10.2. Расходимость последовательности $x_{n+1} = \varphi(x_n)$ при $\varphi'(x) > 1$.

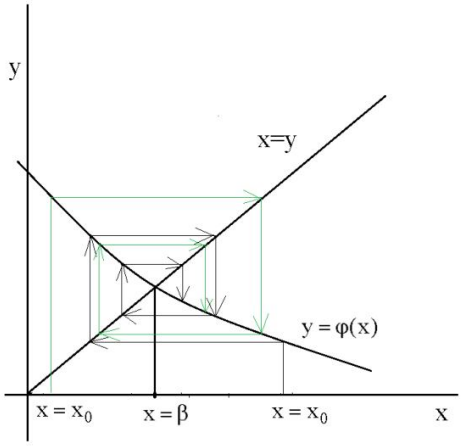

Рис.10.3. Сходимость последовательности $x_{n+1} = \varphi(x_n)$ при $\varphi'(x) > -1$.

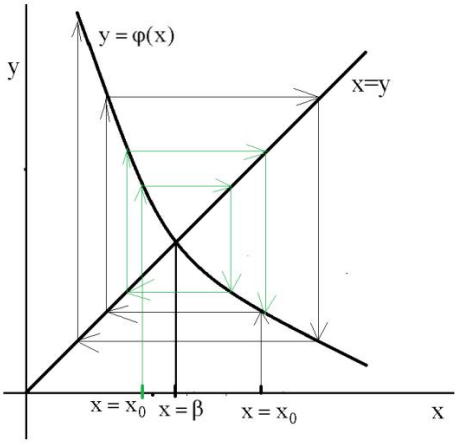

Рис.10.4. Расходимость последовательности $x_{n+1} = \varphi(x_n)$ при $\varphi'(x) < -1$.

Для характеристики методов решения нелинейных уравнений вводится понятие порядка метода.

Говорят, что метод имеет k - й порядок, если существуют $C_1 > 0, C > \infty$ *такие, что*

$$\rho(x_{n+1},\beta) \le C_2[\rho(x_n,\beta)]^k$$

при условии, что $\rho(x_n,\beta) \le C_1$.

Чем больше k, тем быстрее сходится данный метод. Заметим, что данная оценка сходимости метода справедлива лишь в достаточно малой окрестности корня, где $\rho(x_n,\beta) < 1$. Рассмотрим ряд конкретных методов.

Метод деления отрезка пополам

Дано уравнение f(x) = 0. Пусть $f(x) \in C[a,b]$ и известно, что на [a,b] имеется единственный корень этого уравнения. Этот корень может быть найден с помощью следующего алгоритма:

1. Полагаем $a_0 = a, b_0 = b$ и вычисляем $c_0 = (a_0 + b_0)/2$.
2. Вычисляем $f(c_0)$ и произведения $f(a_0)f(c_0)$ и $f(b_0)f(c_0)$.
3. Если $f(a_0)f(c_0) < 0$, то b_0 заменяем на c_0 и повторяем пункт 1. Если $f(b_0)f(c_0) < 0$, то a_0 заменяем на c_0 и повторяем пункт 1. Процесс заканчивается, если $f(c_k) < \varepsilon$. Тогда корень уравнения $\beta = c_k$.

Принцип сжатых отображений очевиден. Справедлива оценка погрешности метода

$$|\beta - c_n| \le \frac{b-a}{2^{n+1}},$$

из которой следует, что это метод первого порядка по скорости сходимости

$$\rho(\beta,c_{n+1}) \le \frac{1}{2}\rho(\beta,c_n).$$

Метод релаксации

Пусть в некоторой окрестности корня уравнения $f(x) = 0$ производная $f'(x)$ не меняет знака. Тогда итерационный метод решения такого уравнения, называемый *методом релаксации* записывается в следующем виде

$$x_{n+1} = x_n - \tau \cdot f(x_n), \quad n = 0,1,2,\dots, \qquad (3.2)$$

где $\tau = \text{const}$ - параметр релаксации. Нетрудно видеть, что для этого метода $\varphi(x) = x - \tau \cdot f(x)$ и условие сходимости $|\varphi'(x)| < 1$ принимает вид

$$-2 < -\tau \cdot f'(x) < 0.$$

Если в окрестности корня имеет место оценка $0 < m < |f'(x)| < M$, то из этого неравенства следует, что $\tau \in (0, 2/M)$, причем знак параметра релаксации определяется из условия $\tau \cdot f'(x) > 0$. Понятно, что скорость сходимости метода (3.2) будет наименьшей при $\tau \to 0$, а также при $\tau \to 2/M$, когда константа Липшица близка к единице (3.1), и поэтому существует оптимальное значение $\tau = \tau_{opt}$, при котором скорость сходимости будет максимальной.

Обозначим $q = |\varphi'(x)| = |1 - \tau \cdot f'(x)|$ и условие сходимости (3.2) есть $q < 1$. В плоскости $(q, f'(x))$ функция $q(f'(x))$ есть уравнение прямой, а параметр τ есть тангенс угла ее наклона к оси $f'(x)$. Тогда наибольшее значение величины q достигается на концах интервала $f'(x) \in [m, M]$, т.е.

$$q(\tau) = \max\{|1 - \tau \cdot m|, |1 - \tau \cdot M|\}.$$

Тогда значение параметра $\tau = \tau_{opt}$, при котором величина $q(\tau)$ достигает наименьшего значения, а, следовательно, скорость сходимости итерационного процесса будет максимальной, определится из равенства

$$1 - \tau_{opt} \cdot m = -(1 - \tau_{opt} \cdot M),$$

откуда

$$\tau_{opt} = \frac{2}{m + M}.$$

Метод Ньютона

Если уравнение $f(x) = 0$ имеет корень $x = \beta$, а функция $\varphi(x)$ непрерывна в окрестности $x = \beta$, то уравнение

$$x = \varphi(x) \equiv x - \psi(x) f(x)$$

также имеет корень $x = \beta$. Функцию $\psi(x)$ можно подобрать так, чтобы итерационный процесс оказался сходящимся.

В методе Ньютона, который также называют методом касательных, в качестве функции $\psi(x)$ выбирается $\psi(x) = \frac{1}{f'(x)}$. Тогда имеем:

$$x = \varphi(x) \equiv x - \frac{f(x)}{f'(x)}.$$

В этом случае

$$\varphi'(x) = 1 - \frac{[f'(x)]^2 - f(x)f''(x)}{[f'(x)]^2} = \frac{f(x)f''(x)}{[f'(x)]^2}$$

и $\varphi'(\beta) = 0$. Это означает, что существует такая окрестность точки β, в которой $|\varphi'(x)| \le K < 1$, и если x_0 взята из этой окрестности, то последовательность

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n = 0,1,2,\ldots \tag{3.3}$$

будет сходиться к $x = \beta$. Геометрический смысл метода заключается в следующем. Уравнение касательной, проведенной к кривой $y = f(x)$ из точки с координатами (x_n, y_n) до пересечения касательной с осью абсцисс в точке с координатами $(x_{n+1}, y_{n+1} = 0)$ имеет вид $(y_{n+1} - y_n) = f'(x_n)(x_{n+1} - x_n)$, откуда и следует формула (3.3).

Оценим скорость сходимости этого метода. Разлагая f(x) в ряд Тейлора в точке x_n и вычисляя ее значение в точке $x = \beta$, имеем

$$f(\beta) = 0 = f(x_n) + f'(x_n)(\beta - x_n) + \frac{(\beta - x_n)^2}{2} f''(\xi).$$

Отсюда

$$\frac{f(x_n)}{f'(x_n)} = x_n - \beta - \frac{1}{2}\frac{f''(\xi)}{f'(x_n)}(\beta - x_n)^2.$$

Следовательно,

$$x_{n+1} - \beta = x_n - \beta - \frac{f(x_n)}{f'(x_n)} = \frac{1}{2}\frac{f''(\xi)}{f'(x_n)}(\beta - x_n)^2.$$

Если $m = \underbrace{\min}_{[a,b]} |f'(x)|$, $M = \underbrace{\max}_{[a,b]} |f''(x)|$, то

$$|x_{n+1} - \beta| \le \frac{M}{2m} |x_n - \beta|^2$$

и метод Ньютона является методом второго порядка.

Решение систем нелинейных уравнений

Рассмотрим систему нелинейных уравнений

$$
\begin{aligned}
&f_1(x_1, x_2, \ldots, x_n) = 0,\\
&f_2(x_1, x_2, \ldots, x_n) = 0,\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&f_n(x_1, x_2, \ldots, x_n) = 0,
\end{aligned}
\tag{3.4}
$$

где $f_i, i = 1,2,\ldots,n$ - функции действительных переменных $x_1, x_2, \ldots, x_n$. В векторном виде она имеет вид

$$\vec{f}(\vec{x}) = 0, \quad \vec{x} = (x_1, x_2, \ldots, x_n), \vec{f} = (f_1, f_2, \ldots, f_n)^T$$

Для ее решения также используются итерационные методы, основанные на построении сходящихся последовательностей векторов $\{\vec{x}^n\}$, (номер итерации будем теперь писать вверху) однако, в отличие от скалярного уравнения, сходимость уже следует рассматривать по некоторой норме векторов.
Далее, по аналогии со скалярным случаем, запишем общий вид стационарного итерационного метода

$$\vec{x}^{n+1} = \vec{\Phi}(\vec{x}^n), \tag{3.5}$$

для сходимости которого необходимо, чтобы вектор-функция $\vec{\Phi}$ обеспечивала сжатое отображение n- мерного линейного пространства R^n векторов $\vec{x}$ в себя. Выясним условия, при которых это имеет место. Пусть $\vec{\Phi}$ непрерывна вместо со своими производными в некоторой области D и пусть $\vec{x}, \vec{y} \in D$. Применяя формулу Тейлора для функции многих переменных, разложим ее в ряд в точке $\vec{x}$ и вычислим в точке $\vec{y}$

$$\vec{\Phi}(\vec{y}) = \vec{\Phi}(\vec{x}) + \vec{\Phi}'(\vec{\xi})(\vec{y} - \vec{x}), \tag{3.6}$$

где $\vec{\xi} = \vec{x} + \Theta(\vec{y} - \vec{x}), 0 < \Theta < 1, \vec{\Phi}'(\vec{\xi})$ - матрица Якоби, имеющая следующий вид

$$\vec{\Phi}'(\vec{\xi}) = \begin{pmatrix}
\frac{\partial \Phi_1}{\partial x_1} & \frac{\partial \Phi_1}{\partial x_2} & \cdots & \frac{\partial \Phi_1}{\partial x_n}\\
\frac{\partial \Phi_2}{\partial x_1} & \frac{\partial \Phi_2}{\partial x_2} & \cdots & \frac{\partial \Phi_2}{\partial x_n}\\
\ldots\ldots & \ldots\ldots & \cdots & \\
\frac{\partial \Phi_n}{\partial x_1} & \frac{\partial \Phi_n}{\partial x_2} & \cdots & \frac{\partial \Phi_n}{\partial x_n}
\end{pmatrix}$$

Из (5.13) следует, что

$$\left\|\vec{\Phi}(\vec{y})-\vec{\Phi}(\vec{x})\right\| \le \left\|\vec{\Phi}'(\vec{\xi})\right\| \cdot \left\|\vec{y}-\vec{x}\right\|.$$

Таким образом, если существует некоторая константа q, такая, что $0 < q < 1$, что для любого $\vec{\xi} \in D$ выполняется неравенство

$$\left\|\vec{\Phi}'(\vec{\xi})\right\| \le q, \tag{3.7}$$

то отображение $\vec{\Phi}$ является сжимающим с постоянной q на множестве D. Конечно, выполнение условие (3.7) зависит от выбора нормы, но если с его помощью удается показать сходимость последовательности (3.5) в некоторой норме, то, вследствие эквивалентности норм в конечномерных пространствах, последовательность (3.5) будет сходиться в любой норме. Поэтому достаточно установить справедливость выполнения условия (3.7) хотя бы в какой-то одной норме.

Метод релаксации

Этот метод является простым обобщением метода (3.2) на случай системы нелинейных уравнений

$$\vec{x}^{n+1} = \vec{x}^{n} - \tau \cdot \vec{f}(\vec{x}^{n}), \quad n = 0,1,2,\ldots \tag{3.8}$$

Для него $\vec{\Phi}(\vec{x}) = \vec{x} - \tau \cdot \vec{f}(\vec{x})$ и метод сходится с первым порядком, если в некоторой области D выполнено условие сходимости

$$\left\|E - \tau \cdot \vec{f}'(\vec{x})\right\| < 1,$$

где $\vec{f}'(\vec{x})$ есть матрица Якоби и $\vec{x}_0 \in D$.

Метод Ньютона

Запишем метод Ньютона для скалярного случая (3.3) в следующем виде

$$f'(x_n) \cdot (x_{n+1} - x_n) = -f(x_n).$$

Тогда его обобщение на случай системы нелинейных уравнений достаточно очевидно

$$\vec{f}'(\vec{x}^{n}) \cdot (\vec{x}^{n+1} - \vec{x}^{n}) = -\vec{f}(\vec{x}^{n}),$$

где $\vec{f}'(\vec{x}^n)$ есть матрицы Якоби, вычисляемая в точке $\vec{x}^n$. Или

$$\vec{x}^{n+1} = \vec{x}^n - [\vec{f}'(\vec{x}^n)]^{-1}\vec{f}(\vec{x}^n), \quad n = 0,1, \ldots , \tag{3.9}$$

где

$$[\vec{f}'(\vec{x}^n)]^{-1} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \ldots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \ldots & \frac{\partial f_2}{\partial x_n} \\ \ldots\ldots & \ldots\ldots & & \ldots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \ldots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}^{-1}$$

есть матрица, обратная к матрице Якоби, также вычисляемая в точке $\vec{x}^n$.
Условия сходимости метода (3.9) определяются следующей теоремой

Теорема *(*Л.В.Канторович*). Если функции* $f_i(x_1, x_2, \ldots, x_n)$ *непрерывны вместе со своими частными производными первого и второго порядков в некоторой выпуклой области* G, *содержащей точку* $\vec{x}^0$ *вместе с ее окрестностью* $\left\|\vec{x} - \vec{x}^0\right\|_1 \le \rho$ *и выполнены следующие условия:*

1) в точке $\vec{x}^0$ *существует матрица* $[\vec{f}'(\vec{x}^0)]^{-1}$, *обратная к* $\vec{f}'(\vec{x}^0)$ *и* $\left\|[\vec{f}'(\vec{x}^0)]^{-1}\right\|_1 \le M_0$;

2) $\left\|\vec{f}(\vec{x}_0^n)\right\|_1 \le \delta,\ M_0\delta = r_0 \le \rho/2$;

3) $\sum_{k=1}^{n}\left|\frac{\partial^2 f_i}{\partial x_j \partial x_k}\right| \le L_0$ при $i, j = 1,2, \ldots, n$;

4) $h_0 = 2n\, r_0 M_0 L_0 \le 1$,

то последовательность (5.12) *сходится к* $\vec{x}^*$. *При этом* $\vec{x}^*$ *является единственным решением системы* $\vec{f}(\vec{x}) = 0$ *в области* $\left\|\vec{x} - \vec{x}^0\right\|_1 \le 2r_0$ *и имеет место оценка*

$$\left\|\vec{x}^n - \vec{x}^*\right\|_1 \le \left(\frac{1}{2}\right)^{(n-1)} h_0^{2^n - 1} r_0.$$

К сожалению, воспользоваться этой теоремой при решении практических задач удается в довольно редких случаях.

Методы Якоби и Зейделя

В этих методах новое значение вектора $\vec{x}^{n+1}$ находится последовательно для всех его координат.

Метод Якоби для решения системы (3.4) имеет следующий вид

$$f_i(x_1^n, x_2^n, \ldots, x_{i-1}^n, x_i^{n+1}, x_{i+1}^n, \ldots, x_i^n) = 0, \quad i = 1,2, \ldots, n \ .$$

Таким образом, в этом методе для нахождения нового значения каждой компоненты x_i^{n+1} вектора $\vec{x}^{n+1}$ нужно решить одно скалярное нелинейное уравнение каким-либо из рассмотренных ранее методов. Такая независимость вычислений позволяет эффективно реализовывать данный алгоритм на параллельных вычислительных системах.

Метод Зейделя для (3.4) записывается следующим образом

$$f_i(x_1^{n+1}, x_2^{n+1}, \ldots, x_{i-1}^{n+1}, x_i^{n+1}, x_{i+1}^n, \ldots, x_i^n) = 0, \quad i = 1,2, \ldots, n \ ,$$

и сводится к последовательному решению уравнений, входящих в систему, для нахождения значения x_i^{n+1}. В отличие от метода Якоби, каждое новое значение x_i^{n+1} подставляется во все последующие уравнения системы. Поэтому построить эффективный параллельный алгоритм для этого метода не удается.

Простых и конструктивных условий сходимости для этих двух методов, к сожалению, не получено и области сходимости приходиться определять эмпирически для каждой конкретной системы нелинейных уравнений.