

Тема 3. Аппроксимация функций и смежные вопросы (продолжение).

Простейшие квадратурные формулы.

Квадратурные формулы – это формулы для вычисления значения определенного интеграла. Их получение – одно из многочисленных возможных приложений интерполяционных полиномов. Пусть требуется вычислить следующий интеграл

$$I = \int_a^b f(x)dx, \quad (1)$$

точное определение которого весьма затруднено или невозможно. Тогда исходная функция может быть аппроксимирована интерполяционным полиномом

$$f(x) = Q_m(x) + R_m(x),$$

и интеграл от интерполяционного полинома порождает некоторую квадратурную формулу

$$I = \int_a^b f(x)dx \approx \int_a^b Q_m(x)dx, \quad (2)$$

а интеграл от остаточного члена полинома определяет ее погрешность

$$\varepsilon = \int_a^b R_m(x)dx \quad (3)$$

Будем последовательно подставлять в (2) полиномы различных степеней, начиная с нулевой ($Q_0(x) = f(x_0)$)

$$I \approx (b-a)f(x_0).$$

Наиболее популярными являются следующие три варианта выбора узла x_0

$$x_0 = a, \quad I \approx (b-a)f(a), \quad (4)$$

$$x_0 = b, \quad I \approx (b-a)f(b) \quad (5)$$

$$x_0 = \frac{a+b}{2}, \quad I \approx (b-a)f\left(\frac{a+b}{2}\right) \quad (6)$$

Формула (4) называется квадратурной формулой *левых прямоугольников*, (5) – формулой *правых прямоугольников*, (6) – формулой *средних прямоугольников*. Причины таких названий легко понять из геометрических иллюстраций рис. 1.а,б,в, откуда видно, что площадь под заданной функцией аппроксимируется площадью соответствующего прямоугольника.

Интегрирование полинома первой степени с узлами интерполирования $x_0 = a$ и $x_1 = b$

$$Q_1(x) = \frac{x-b}{a-b}f(a) + \frac{x-a}{b-a}f(b)$$

порождает квадратурную формулу *трапеций* (рис. 1г)

$$I \approx \frac{b-a}{2}(f(a) + f(b)), \quad (7)$$

а интегрирование полинома второй степени с узлами $x_0 = a$, $x_1 = \frac{a+b}{2}$ и $x_2 = b$ приводит к квадратурной формуле Симпсона

$$I \approx \frac{b-a}{6} \left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right). \quad (8)$$

Аналогичные формулы могут быть получены и для полиномов более высоких степеней. Прежде, чем будут получены соответствующие выражения для погрешности, проверьте свою интуицию и попытайтесь ответить на вопрос, руководствуясь формулами (6), (7) и соответствующими рисунками.

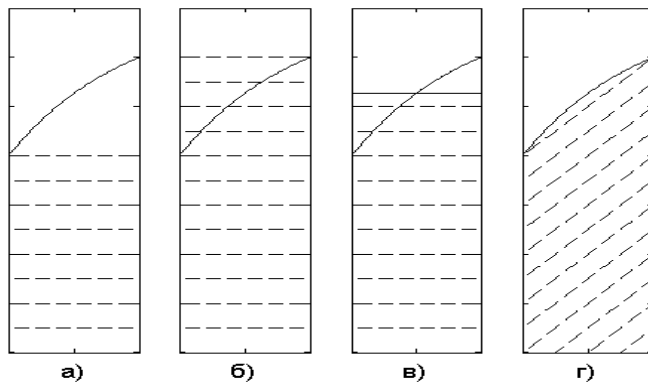


Рис 1. Квадратурные формулы (4) – (7).

Вопрос 8. Многократно для самых различных функций считаются интегралы по формуле средних прямоугольников и по формуле трапеций. Для какой из них «в среднем» значение интеграла будет получаться точнее?

Оценку погрешности квадратурных формул будем выполнять на основе двух известных в математике теорем о среднем.

Теорема. Пусть $f(x)$ и $g(x)$ непрерывны на $[a, b]$, а $g(x)$ в дополнение к этому знакопостоянна. Тогда найдется точка $c \in [a, b]$ такая, что

$$\int_a^b g(x)f(x)dx = f(c) \int_a^b g(x)dx. \quad (9)$$

Теорема. Пусть $f(x)$ непрерывна на $[a, b]$, и заданы N точек $x_k \in [a, b]$. Найдется точка $\eta \in [a, b]$ такая, что

$$\frac{1}{N} \sum_{k=1}^N f(x_k) = f(\eta). \quad (10)$$

Остаточный член интерполяционного полинома нулевой степени имеет вид

$$R_0(x) = \frac{x - x_0}{1!} f'(\eta).$$

Полезно заметить, что точка η зависит от x , т.е. $\eta = \eta(x)$, как это уже отмечалось при выводе остаточного члена интерполяционного полинома. Последовательно подставляя $x_0 = a$, b и $\frac{a+b}{2}$, вычислим интеграл (3).

Формула левых прямоугольников:

$$\varepsilon_{лев.нр.} = \int_a^b (x-a) f'(\eta) dx = f'(\eta^*) \int_a^b (x-a) dx = \frac{(b-a)^2}{2} f'(\eta^*). \quad (11)$$

Формула правых прямоугольников:

$$\varepsilon_{прав.нр.} = \int_a^b (x-b) f'(\eta) dx = f'(\eta^*) \int_a^b (x-b) dx = -\frac{(b-a)^2}{2} f'(\eta^*). \quad (12)$$

Формула средних прямоугольников

$$\varepsilon_{средн.нр.} = \int_a^b \left(x - \frac{a+b}{2} \right) f'(\eta) dx.$$

В выражениях (11) и (12) была использована теорема о среднем (9), так как функции $x-a$ и $x-b$ являются знакопостоянными на $[a, b]$. Воспользоваться этой теоремой для средних прямоугольников не удастся, так как $\left(x - \frac{a+b}{2} \right)$ меняет знак. Для оценки погрешности в этом случае воспользуемся разложением $f(x)$ в ряд в точке $\frac{a+b}{2}$

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

интегрируя его на $[a, b]$. Интеграл от первого слагаемого дает формулу средних прямоугольников, интеграл от второго слагаемого равен нулю, а интеграл от последнего слагаемого дает оценку погрешности. Таким образом, для формулы средних прямоугольников имеем

$$\varepsilon_{ср.нр.} = \frac{1}{2} \int_a^b \left(x - \frac{a+b}{2} \right)^2 f''(\eta) dx = \frac{f''(\eta^*)}{2} \int_a^b \left(x - \frac{a+b}{2} \right)^2 dx = \frac{(b-a)^3}{24} f''(\eta^*). \quad (13)$$

Использование теоремы о среднем в последнем случае уже вполне закономерно. Формулы (11) и (12) отличаются знаком. Возникает следующий вопрос.

Вопрос 9. Дважды вычисляем интеграл по формулам сначала левых, а затем правых прямоугольников. Определяем полусумму результатов. Погрешности сокращаются, и результат получается точным?

Для оценки погрешности формулы трапеций проинтегрируем остаточный член полинома первой степени

$$\varepsilon_{\text{тран}} = \int_a^b \frac{(x-a)(x-b)}{2!} f''(\eta) dx = f''(\eta^*) \int_a^b \frac{(x-a)(x-b)}{2!} dx = -\frac{(b-a)^3}{12} f''(\eta^*). \quad (14)$$

Здесь также применена теорема о среднем. Наконец, для интеграла от остаточного члена интерполяционного полинома второй степени

$$R_2(x) = \frac{(x-a)\left(x - \frac{a+b}{2}\right)(x-b)}{3!} f'''(\eta)$$

условия теоремы о среднем не выполняются, и погрешность формулы Симпсона определяется по-другому. По этим же трем узлам строится полином Эрмита уже третьей степени с двумя условиями в центральной точке, погрешность которого и интегрируется. Результат приведем без вывода.

$$\varepsilon_{\text{Симпс}} = -\frac{(b-a)^5}{2880} f^{(4)}(\eta) \quad (15)$$

В большинстве случаев подынтегральная функция не описывается удовлетворительно полиномами первой или второй степени. Поэтому, для достижения необходимой точности исходный промежуток разбивается на такие малые промежутки, где указанная аппроксимация удачна, на каждом из этих промежутков применяется выбранная квадратурная формула, а результаты складываются. Такие формулы получили название составных (или больших) квадратурных формул.

Составные квадратурные формулы.

Разобьем исходный промежуток на N равных промежутков $[x_k, x_{k+1}]$

$$h = \frac{b-a}{N}, \quad x_k = x_0 + kh, \quad x_0 = a, \quad x_N = b.$$

На каждом таком промежутке применим формулу левых прямоугольников и результаты сложим

$$I_k = \int_{x_k}^{x_{k+1}} f(x) dx \approx (x_{k+1} - x_k) f(x_k) = \frac{b-a}{N} f(x_k);$$

$$I_{\text{лев.пр}} = \sum_{k=0}^{N-1} I_k \approx \frac{b-a}{N} \sum_{k=0}^{N-1} f(x_k) \quad (16)$$

Аналогичные преобразования для формул правых и средних прямоугольников, а также для формулы трапеций приводят к следующим результатам

правые прямоугольники:

$$I_k = \int_{x_k}^{x_{k+1}} f(x) dx \approx (x_{k+1} - x_k) f(x_{k+1}) = \frac{b-a}{N} f(x_{k+1});$$

$$I_{\text{прав.пр}} = \sum_{k=0}^{N-1} I_k \approx \frac{b-a}{N} \sum_{k=0}^{N-1} f(x_{k+1}) = \frac{b-a}{N} \sum_{k=1}^N f(x_k) \quad (17)$$

средние прямоугольники:

$$I_k = \int_{x_k}^{x_{k+1}} f(x)dx \approx (x_{k+1} - x_k) f(x_k + h/2) = \frac{b-a}{N} f(x_k + h/2);$$

$$I_{\text{средн.пр}} = \sum_{k=0}^{N-1} I_k \approx \frac{b-a}{N} \sum_{k=0}^{N-1} f(x_k + h/2) = \frac{b-a}{N} \sum_{k=0}^{N-1} f\left(x_k + \frac{b-a}{2N}\right) \quad (18)$$

трапеции: $I_k = \int_{x_k}^{x_{k+1}} f(x)dx \approx \frac{x_{k+1} - x_k}{2} (f(x_{k+1}) + f(x_k)) = \frac{b-a}{2N} (f(x_{k+1}) + f(x_k));$

$$I_{\text{trap}} = \sum_{k=0}^{N-1} I_k \approx \frac{b-a}{2N} \sum_{k=0}^{N-1} (f(x_{k+1}) + f(x_k)) = \frac{b-a}{2N} \left(f(a) + 2 \sum_{k=1}^{N-1} f(x_k) + f(b) \right). \quad (19)$$

Формулы (16) – 19) называются *составными квадратурными формулами* левых, правых, средних прямоугольников и трапеций соответственно. Для получения составной формулы Симпсона будем выбирать N всегда четным, а исходный промежуток разобьем на $N/2$ равных промежутков $[x_k, x_{k+2}]$ длиной $2(b-a)/N$. На каждом из них интеграл равен

$$I_k = \int_{x_k}^{x_{k+2}} f(x)dx \approx \frac{x_{k+2} - x_k}{6} (f(x_{k+2}) + 4f(x_{k+1}) + f(x_k)) = \frac{b-a}{3N} (f(x_{k+2}) + 4f(x_{k+1}) + f(x_k))$$

Суммируя по всем промежуткам, получаем *составную формулу Симпсона*

$$I_{\text{Симпс}} = \frac{b-a}{3N} [f(a) + 4(f_1 + f_3 + f_5 + \dots + f_{N-1}) + 2(f_2 + f_4 + \dots + f_{N-2}) + f(b)]. \quad (20)$$

Погрешности составных формул.

Для оценки погрешности составных формул вычисляем погрешность каждого малого участка и результаты складываем. Так, для формулы левых прямоугольников на основе (11), применяя дискретный вариант теоремы о среднем (10) к выражению в квадратных скобках, имеем

$$\varepsilon_k = \frac{(x_{k+1} - x_k)^2}{2} f'(\eta_k) = \frac{(b-a)^2}{2N^2} f'(\eta_k),$$

$$\varepsilon_{\text{лев.пр}} = \sum_{k=1}^N \varepsilon_k = \frac{(b-a)^2}{2N} \left[\frac{1}{N} \sum_{k=1}^N f'(\eta_k) \right] = \frac{(b-a)^2}{2N} f'(\eta). \quad (21)$$

Повторяя аналогичные вычисления с формулой (12) для правых прямоугольников получаем

$$\varepsilon_{\text{прав.пр}} = -\frac{(b-a)^2}{2N} f'(\eta). \quad (22)$$

Выражение (13) позволяет получить составную формулу средних прямоугольников

$$\varepsilon_k = \frac{(x_{k+1} - x_k)^3}{24} f''(\eta_k) = \frac{(b-a)^3}{24N^3} f''(\eta_k),$$

$$\varepsilon_{\text{средн.пр}} = \sum_{k=1}^N \varepsilon_k = \frac{(b-a)^3}{24N^2} \left[\frac{1}{N} \sum_{k=1}^N f''(\eta_k) \right] = \frac{(b-a)^3}{24N^2} f''(\eta), \quad (23)$$

а выражение (14) – аналогичную формулу трапеций

$$\begin{aligned}\varepsilon_k &= -\frac{(x_{k+1} - x_k)^3}{12} f''(\eta_k) = -\frac{(b-a)^3}{12N^3} f''(\eta_k), \\ \varepsilon_{\text{trap}} &= \sum_{k=1}^N \varepsilon_k = -\frac{(b-a)^3}{12N^2} \left[\frac{1}{N} \sum_{k=1}^N f''(\eta_k) \right] = -\frac{(b-a)^3}{12N^2} f''(\eta).\end{aligned}\quad (24)$$

Теперь сравнение формул (23) и (24) позволяет вернуться к вопросу 8.

При использовании выражения (15) для оценки погрешности составной формулы Симпсона следует учитывать, что длина каждого малого участка в два раза больше, чем ранее, а число таких участков в два раза меньше – $N/2$.

$$\begin{aligned}\varepsilon_k &= -\frac{(x_{k+2} - x_k)^5}{2880} f^{(4)}(\eta_k) = -\frac{(b-a)^5}{90N^5} f^{(4)}(\eta_k), \\ \varepsilon_{\text{Симпс}} &= \sum_{k=1}^{N/2} \varepsilon_k = -\frac{(b-a)^5}{180N^4} \left[\frac{1}{N/2} \sum_{k=1}^{N/2} f^{(4)}(\eta_k) \right] = -\frac{(b-a)^5}{180N^4} f^{(4)}(\eta).\end{aligned}\quad (25)$$

При сравнении формул (21) – (25) наибольший интерес вызывает зависимость погрешности от N , позволяющая оценить дополнительный объем вычислений, связанный с увеличением N для уменьшения погрешности до требуемой величины. Нетрудно также заметить, что все эти формулы описываются общей зависимостью

$$\varepsilon = \alpha \frac{(b-a)^{p+1}}{N^p} f^{(p)}(\eta), \quad (26)$$

где α – некоторое число. Так для формул левых и правых прямоугольников $p = 1$, для средних прямоугольников и трапеций $p = 2$, а для формулы Симпсона $p = 4$.

Следует заметить, что все квадратурные формулы были получены в предположении, что пределы интегрирования конечны, а подынтегральная функция не имеет особенностей. При вычислении значений несобственных интегралов обычно предварительно с помощью специальных приемов интеграл приводится к специальному виду, позволяющему сочетать использование квадратурной формулы и некоторого аналитического приема. При этом успех часто определяется квалификацией пользователя и его умением учесть специфику решаемой задачи.

В заключение настоящего параграфа остановимся на том, как погрешность квадратурных формул оценивается на практике. Целесообразно назвать следующие способы.

Способ 1. Непосредственная оценка погрешности по формуле (26). Применяется крайне редко из-за известных трудностей, связанных с оценкой производной, особенно для табличной функции, а также из-за отсутствия информации о расположении точки η .

Способ 2. Сравнение результатов, полученных по выбранной составной квадратурной формуле для N и $2N$. В случае неудовлетворительного совпадения значение N удваивается до тех пор, пока требуемая точность не будет достигнута.

Способ 3. Если по каким-либо причинам есть ограничение на рост величины N и способ 2 применен быть не может, сравнивают результаты для одного и того же значения N , но для различных квадратурных формул.

Способ 2 наиболее популярен, и построенная на его основе упрощенная версия алгоритма легко укладывается в несколько элементарных шагов.

Шаг 1. Задаемся начальным значением N и вычисляем интеграл по любой составной квадратурной формуле, присваивая результат переменной I_{old} .

Шаг 2. Удваиваем величину N и вновь вычисляем интеграл, присваивая результат переменной I_{new} .

Шаг 3. Если значения I_{old} и I_{new} совпали с заданной точностью, то прекращаем работу. В противном случае присваиваем переменной I_{old} значение I_{new} и возвращаемся к шагу 2.

Несмотря на весьма простой вид, этот алгоритм позволяет достичь требуемой точности при вычислении интеграла от любой «разумной» функции, не имеющей особенностей на промежутке интегрирования. Более того, надежность написанной на его основе программы вполне сравнима с надежностью программного обеспечения, обсуждаемого ниже (программа QUANC8).

Общий подход к построению квадратурных формул. Метод неопределенных коэффициентов.

Все полученные выше простейшие квадратурные формулы (не составные!) имеют следующий вид

$$\int_a^b f(x)dx \approx \sum_{k=1}^S A_k f(x_k) \quad (27)$$

Узлы x_k и веса A_k квадратурной формулы в предыдущем параграфе получались на основе интегрирования соответствующих интерполяционных полиномов. Поставим задачу несколько иначе. Требуется выбрать x_k и A_k так, чтобы формула (27) была бы точной для полиномов заданной степени. Логика таких требований очевидна. Если подынтегральная функция хорошо аппроксимируется этим полиномом (не обязательно интерполяционным!), то и формула (27) обеспечит требуемую погрешность решения задачи. В противном случае промежуток $[a, b]$ всегда можно разбить на достаточно малые промежутки, применить составные квадратурные формулы и добиться желаемой точности.

Потребуем, чтобы формула (27) была бы точна для полинома нулевой степени $f(x) = \alpha = const$. Вынося константу α из-под знаков интеграла и суммы и сокращая на нее, имеем

$$\sum_{k=1}^S A_k = b - a.$$

Второе уравнение получим, требуя точности (27) для полинома первой степени и поставив с этой целью $f(x) = x$

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

Это же требование для $f(x) = x^2$ выглядит следующим образом

$$\sum_{k=1}^S A_k x_k^2 = \frac{b^3 - a^3}{3},$$

а в общем случае для $f(x) = x^N$ условие с номером $N+1$ приобретает вид

$$\sum_{k=1}^S A_k x_k^N = \frac{b^{N+1} - a^{N+1}}{N+1}.$$

Объединяя все уравнения, получим следующую систему в общем случае нелинейных уравнений

$$\begin{aligned} 1) \quad & \sum_{k=1}^S A_k = b - a, \\ 2) \quad & \sum_{k=1}^S A_k x_k = \frac{b^2 - a^2}{2}, \\ 3) \quad & \sum_{k=1}^S A_k x_k^2 = \frac{b^3 - a^3}{3}, \\ & \dots \dots \dots \\ N+1) \quad & \sum_{k=1}^S A_k x_k^N = \frac{b^{N+1} - a^{N+1}}{N+1} \end{aligned} \quad (28)$$

относительно $2S$ неизвестных x_k и A_k . В связи с нелинейностью возникает целый ряд вопросов, требующих разрешения. *Существует ли решение? Единственно ли оно? Получаются ли x_k в этом случае не только вещественными и различными, но и лежащими на $[a, b]$?*

Система (28) является общей для многих семейств квадратурных формул, отличающихся друг от друга дополнительными условиями, накладываемыми на x_k и A_k . Рассмотрим три таких семейства.

Квадратурные формулы Ньютона-Котеса. Узлы квадратурной формулы x_k здесь выбираются равноотстоящими

$$h = \frac{b-a}{S-1}, \quad x_k = a + (k-1)h, \quad x_1 = a, \quad x_S = b.$$

Система (28) в данном случае является линейной относительно S неизвестных A_k и легко разрешима. Ее определитель является уже известным нам определителем Вандермонда, что обеспечивает единственность решения. Кроме того, равноотстоящие узлы дают некоторые удобства при программировании этих формул. Но не эти моменты являются решающими. Для составных квадратурных формул при удвоении N (числа внутренних промежутков) в половине возникающих узлов x_k значения функций $f(x_k)$ уже вычислялись ранее и могли быть сохранены, что позволяет сократить объем вычислений вдвое.

Как результат, имеем систему (28) из S уравнений с S неизвестными A_k , и эти квадратурные формулы оказываются гарантированно точными для полиномов степени $N = S - 1$. Нетрудно заметить, что, решая систему последовательно для $S = 1, 2$ и 3 , приходим к уже хорошо знакомым формулам прямоугольников, трапеций и Симпсона, получавшимся ранее другим способом.

Квадратурные формулы Чебышева. Для целого ряда приложений, особенно когда значения $f(x_k)$ определены с заметной погрешностью, П.Л.Чебышевым были предложены квадратурные формулы с равными весами ($A_k = A = const$)

$$\int_a^b f(x)dx \approx A \sum_{k=1}^S f(x_k). \quad (29)$$

Система (28) оказывается системой из $S+1$ уравнения относительно такого же числа неизвестных x_k и A , что позволяет в случае успешного решения получить формулы, гарантированно точные для полиномов степени $N = S$. Однако, учитывая нелинейность системы, вопросы существования и единственности решения выходят на первый план. Оказывается, что система (28) благополучно решается единственным образом для S от 1 до 7 и для $S = 9$. С.Н.Бернштейном было показано, что для других значений S формулы Чебышева не существуют. Для практических целей это не является серьезным препятствием, так как необходимая точность обеспечивается использованием составных квадратурных формул.

Квадратурные формулы Гаусса. В этих формулах на узлы и веса не накладываются никакие дополнительные условия, и все свободные $2S$ параметров используются при решении системы (28) из $2S$ уравнений. В отличие от формул Чебышева формулы Гаусса существуют для любого числа узлов. Они гарантированно точны для полиномов степени $N = 2S - 1$ и называются формулами *наивысшей алгебраической степени точности*. Дополнительно можно отметить, что узлы x_k являются нулями так называемых ортогональных полиномов Лежандра, которые будут рассмотрены позднее.

В заключение укажем, что на практике для получения A_k и x_k нет никакой необходимости каждый раз обращаться к системе (28). Результаты ее решения для стандартного промежутка $[-1, 1]$ или $[0, 1]$ и для каждого конкретного семейства формул приведены в многочисленных справочниках и учебниках. Пользователь лишь ограничивается заменой переменных в (27). Так для промежутка $[-1, 1]$ такая замена переменных выглядит следующим образом:

$x = \frac{a+b}{2} + \frac{b-a}{2}t$. При изменении t от -1 до 1 , переменная x пробегает значения от a до b . С учетом того, что $dx = \frac{b-a}{2}dt$, формула (27) имеет вид

$$\int_a^b f(x)dx = \frac{b-a}{2} \int_{-1}^1 f\left(\frac{a+b}{2} + \frac{b-a}{2}t\right)dt \approx \frac{b-a}{2} \sum_{k=1}^S A_k f\left(\frac{a+b}{2} + \frac{b-a}{2}t_k\right), \quad (30)$$

где веса A_k и узлы t_k квадратурной формулы берутся из справочника для $[-1, 1]$. Рассмотренные семейства квадратурных формул не исчерпывают всех возможностей формулы (27). Как уже отмечалось, другие семейства отличаются выбором дополнительных требований к A_k и x_k .

Адаптивные квадратурные формулы. Программа QUANC8.

Рассмотренный ранее простой алгоритм вычисления интеграла на основе составных квадратурных формул со сравнением результатов для N и $2N$, отличается

достаточной надежностью, но его быстродействие может быть заметно повышено. Например, для функции, представленной на рис. 2, на участке с быстрым изменением функции требуется относительно малый шаг. В то же время, так как в составной формуле шаг постоянен для всего промежутка, то общий объем вычислений окажется неоправданно велик.

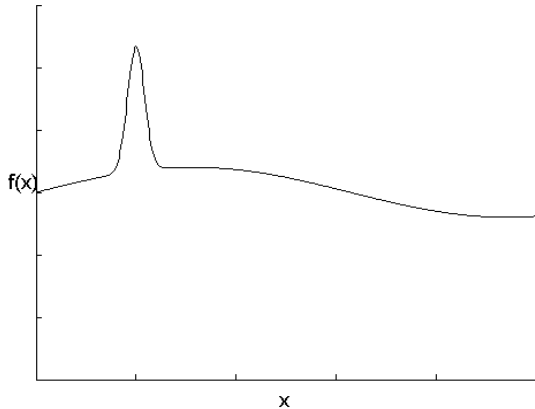


Рис 2. Пример функции с различным поведением

Целесообразным представляется построение алгоритма, который был бы способен *адаптироваться* к виду функции и выбирать достаточно малый шаг там, где функция меняется быстро и характеризуется большими производными, и относительно большой шаг там, где функция меняется медленно. На этом пути возможны два варианта: минимизировать погрешность при заданном объеме вычислений или минимизировать объем вычислений при заданных требованиях к погрешности. В рассматриваемой программе реализован второй подход.

В основу положена квадратурная формула Ньютона-Котеса с девятью узлами, т.е. восемью промежутками между ними, что и оправдывает название программы. Ее составная формула имеет погрешность вида (26) для $p = 10$.

Рассмотрим промежуток длиной h_k внутри $[a, b]$ и введем для него следующие обозначения:

I_k – точное значение интеграла на этом промежутке;

P_k – значение интеграла, вычисленное по квадратурной формуле с девятью узлами;

Q_k – значение интеграла, вычисленное по той же формуле, примененной к двум половинам этого промежутка (по сути используется составная формула с вдвое большим значением N).

Учитывая вид (26) для $p = 10$, для погрешностей P_k и Q_k последовательно имеем

$$I_k - P_k \approx 2^p (I_k - Q_k); \quad I_k \approx \frac{2^p Q_k - P_k}{2^p - 1}; \quad I_k - Q_k \approx \frac{Q_k - P_k}{2^p - 1} = \frac{Q_k - P_k}{1023}.$$

Обозначая требуемую абсолютную погрешность вычисления интеграла на всем промежутке $[a, b]$ за ε_A , считаем промежуток h_k «принятым», а интеграл на нем вычисленным, если выполняется неравенство

$$\left| \frac{Q_k - P_k}{1023} \right| \leq \frac{h_k}{b - a} \varepsilon_A, \quad (31)$$

Множитель $\frac{h_k}{b-a}$ является весовым коэффициентом и отражает вклад погрешности на промежутке h_k в общую погрешность для всего промежутка. Возможно также использование и относительной погрешности ε_R

$$\left| \frac{Q_k - P_k}{1023} \right| \leq \frac{h_k}{b-a} \varepsilon_R |\tilde{I}_k|, \quad (32)$$

где \tilde{I}_k – оценка вычисления интеграла по всему промежутку. Следует, однако, помнить, что использование критерия относительной погрешности усложняется, если значение \tilde{I}_k оказывается нулевым или близким к нулю. В программе **QUANC8** пользователю предоставляется возможность использовать один из трех вариантов контроля погрешности на основе объединения формул (31) и (32)

$$\left| \frac{Q_k - P_k}{1023} \right| \leq \frac{h_k}{b-a} \max(\varepsilon_A; \varepsilon_R |\tilde{I}_k|), \quad (33)$$

вариант 1. $\varepsilon_R = 0$, $\varepsilon_A \neq 0$ – контроль абсолютной погрешности;

вариант 2. $\varepsilon_R \neq 0$, $\varepsilon_A = 0$ – контроль относительной погрешности;

вариант 3. $\varepsilon_R \neq 0$, $\varepsilon_A \neq 0$ – контроль «смешанной» погрешности.

В последнем случае делается попытка избежать упомянутых неприятных ситуаций со значениями \tilde{I}_k , близкими к нулю.

Адаптация программы к виду функции и реализация переменного шага интегрирования реализуются в соответствии со следующим алгоритмом. Вычисляются P_k и Q_k применительно ко всему промежутку. Если погрешность еще достаточно велика, промежуток делится пополам, значения подынтегральной функции $f(x)$, вычисленные на правой половине промежутка, запоминаются, и все повторяется для левой половины промежутка. Такое обращение каждый раз к левой половине текущего промежутка продолжается до тех пор, пока крайний слева промежуток не будет принят. После этого обрабатывается ближайший к нему правый промежуток. Запоминание значений $f(x)$ повышает быстродействие алгоритма. Как уже отмечалось ранее, для составных квадратурных формулах Ньютона-Котеса при удвоении N (числа внутренних промежутков) в половине возникающих узлов x_k значения функций $f(x_k)$ уже вычислялись ранее и могли быть сохранены, что позволяет сократить объем вычислений вдвое.

В программе реализовано два ограничения сверху на объем вычислений. Во-первых, деление промежутка пополам продолжается не более 30 раз. По достижении этой величины соответствующий интеграл на нем считается вычисленным, а промежуток «принятым», независимо от условия (33). Число таких промежутков, принятых с нарушением условия (33), содержится в целой части выходного значения переменной **FLAG**. Следует признать, что длина каждого такого промежутка крайне мала $(b-a)/2^{30} \approx 10^{-9}(b-a)$, и подобная ситуация, как правило, связана с разрывами подынтегральной функции или ее «зашумлением» вычислительной погрешностью. Во-вторых, вводится ограничение сверху на количество вычислений подынтегральной функции $f(x)$. Если этот предел достигнут, то информация о точке

x^* , где возникла трудность, отражена в дробной части выходного значения переменной *FLAG* (там записана величина $(b - x^*) / (b - a)$).

В [14] приводится пример использования программы для вычисления интеграла

$$I = \int_0^2 \frac{\operatorname{tg}(x)}{x} dx.$$

QUANC8 выдает значение переменной *FLAG* равное 91.21, что не только обращает внимание на 91 (!!) промежуток, принятый с нарушением условия (33), но и указывает на точку x^* , где встретились затруднения. Определив x^* ,

$$(b - x^*) / (b - a) = 0.21; \quad x^* = b - 0.21(b - a) = 1.58,$$

пользователь вынужден обратить внимание на неинтегрируемую особенность в точке

$$x = \pi / 2 \approx 1.57.$$

Программа имеет следующие параметры

QUANC8 (*FUN*, *A*, *B*, *ABSERR*, *RELERR*, *RESULT*, *ERREST*, *NOFUN*, *FLAG*),

FUN – имя подпрограммы-функции, вычисляющей значение подынтегральной функции $f(x)$;

A, *B* – нижний и верхний пределы интегрирования;

ABSERR и *RELERR* – границы абсолютной ε_A и относительной ε_R погрешностей.

Остальные параметры – выходные со следующим смыслом:

RESULT – значение интеграла; определенное программой;

ERREST – оценка погрешности, выполненная программой и удовлетворяющая (33);

NOFUN – количество вычислений подынтегральной функции $f(x)$, использованных для получения результата;

FLAG – индикатор надежности результата. Нулевое значение этой переменной отвечает относительной надежности результата, а ненулевое, как уже отмечалось, свидетельствует об отклонениях от нормального хода выполнения программы.

В заключение два вопроса.

Вопрос 10. Предложенная программа **QUANC8** построена на основе формулы Ньютона-Котеса с девятью узлами. Возможно ли построение аналогичной адаптивной программы на основе формулы Симпсона? Если «нет», то почему? Если «да», то в каких частях программы произойдут изменения?

Вопрос 11. Та же группа вопросов, но уже применительно к формуле Гаусса, например, с девятью узлами.

Численное дифференцирование.

Предлагаемая задача ставится следующим образом. Для таблично заданной функции

x	x_0	x_1	x_2	...	x_m
$f(x)$	$f(x_0)$	$f(x_1)$	$f(x_2)$...	$f(x_m)$

требуется оценить значения производной функции $f(x)$ в узлах таблицы.

Книга известного американского ученого Р.В.Хемминга «Численные методы»[15] начинается с весьма полезного совета читателю: «Прежде, чем решать задачу, подумай, что делать с ее решением». Последуем этой рекомендации.

Пусть для функции $f(x)$ выбрана аппроксимирующая функция $g(x)$, которая признана достаточно близкой к исходной по какому-либо заданному критерию, например, по максимуму модуля отклонения

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

Будут ли близки интегралы от этих функций? Положительный ответ на этот вопрос очевиден. А будут ли близки их производные? Рис 3 отражает вид двух функций: $f(x)$ и ее «зашумленного» варианта $g(x)$. Сами функции и интегралы от них (т.е. площади под обеими кривыми) весьма близки, в то время как производные совпадают только в нескольких точках, а часто отличаются даже по знаку!

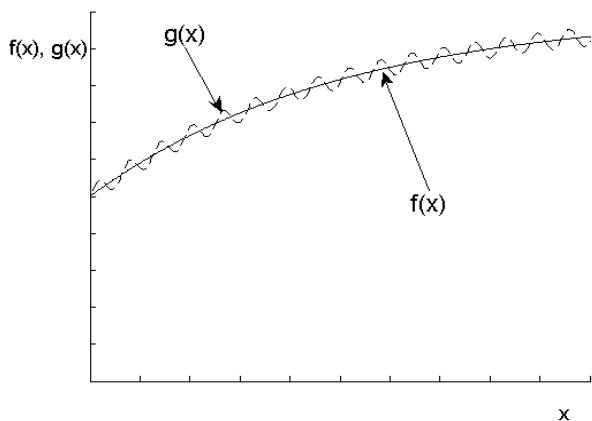


Рис. 3. Функция $f(x)$ и ее «зашумленный» вариант

Еще более убедительным является следующий пример. Рассмотрим следующие две функции $f(x)$ и $g(x)$ и их производные

$$f(x); \quad g(x) = f(x) + \frac{1}{N} \sin(N^2 x);$$

$$\frac{df(x)}{dx}; \quad \frac{dg(x)}{dx} = \frac{df(x)}{dx} + N \cos(N^2 x).$$

С ростом величины N обе функции становятся по отношению друг к другу все ближе и ближе, а их производные ... все дальше и дальше! Таким образом, близость $f(x)$ и $g(x)$ еще не гарантирует близости их производных.

Для того, чтобы повысить уверенность в надежном решении задачи численного дифференцирования, на практике делают целый ряд предположений о характере дифференцируемой функции. Шаг таблицы должен быть не слишком велик и согласован с быстротой изменения функции. Сама функция не должна быть измерена со слишком большими погрешностями (т.е. быть «зашумленной»), не должна слишком резко изменяться и т.п. Примеры функций, подобных $g(x)$ на рис 3, должны быть исключены. Таким образом, численное дифференцирование должно выполняться с максимальной осторожностью и сопровождаться предварительным анализом характера изменения функции и погрешности исходных

данных. В силу указанных причин с повышением порядка получаемой производной значительно возрастает риск получения отрицательного результата.

Идея, лежащая в основе численного дифференцирования, крайне проста и уже использовалась при получении квадратурных формул. (см. (2) – (3)). Исходная функция аппроксимируется интерполяционным полиномом

$$f(x) = Q_m(x) + R_m(x),$$

и производная от полинома дает формулу численного дифференцирования

$$\frac{df(x_k)}{dx} \approx \frac{dQ_m(x_k)}{dx},$$

а производная от остаточного члена позволяет оценить погрешность этой операции

$$\varepsilon = \frac{dR_m(x_k)}{dx}.$$

Ограничимся случаем, когда узлы таблицы будут равноотстоящими с шагом $h = x_{k+1} - x_k$. Начнем с полинома первой степени, построенного по двум узлам x_k и x_{k+1}

$$Q_1(x) = \frac{x - x_{k+1}}{x_k - x_{k+1}} f(x_k) + \frac{x - x_k}{x_{k+1} - x_k} f(x_{k+1}).$$

Дифференцируя его и полагая последовательно $x = x_k$ и $x = x_{k+1}$, получаем

$$\frac{df(x_k)}{dx} \approx \frac{f_{k+1} - f_k}{h}, \quad (34)$$

$$\frac{df(x_{k+1})}{dx} \approx \frac{f_{k+1} - f_k}{h}. \quad (35)$$

Хотя правые части обоих выражений равны, формулы получились принципиально различными. Выполним аналогичные операции для полинома второй степени

$$Q_2(x) = \frac{(x - x_{k+1})(x - x_{k+2})}{(x_k - x_{k+1})(x_k - x_{k+2})} f(x_k) + \frac{(x - x_k)(x - x_{k+2})}{(x_{k+1} - x_k)(x_{k+1} - x_{k+2})} f(x_{k+1}) + \\ + \frac{(x - x_k)(x - x_{k+1})}{(x_{k+2} - x_k)(x_{k+2} - x_{k+1})} f(x_{k+2}).$$

Последовательно полагая $x = x_k$, $x = x_{k+1}$ и $x = x_{k+2}$, получаем

$$\frac{df(x_k)}{dx} \approx \frac{-3f_k + 4f_{k+1} - f_{k+2}}{2h}, \quad (36)$$

$$\frac{df(x_{k+1})}{dx} \approx \frac{f_{k+2} - f_k}{2h}, \quad (37)$$

$$\frac{df(x_{k+2})}{dx} \approx \frac{3f_{k+2} - 4f_{k+1} + f_k}{2h}. \quad (38)$$

Для оценки погрешности всех формул необходимо продифференцировать остаточный член $R_m(x)$. Для полинома первой степени он имеет вид

$$R_1(x) = \frac{(x - x_k)(x - x_{k+1})}{2!} f''(\eta).$$

Уместно напомнить, что при выводе остаточного члена интерполяционного полинома отмечалась зависимость величины η от точки x , где оценивается погрешность, т.е. $\eta = \eta(x)$, и этот факт нужно учитывать при дифференцировании

$$\frac{dR_1(x)}{dx} = \frac{x - x_{k+1}}{2!} f''(\eta) + \frac{x - x_k}{2!} f''(\eta) + \frac{(x - x_k)(x - x_{k+1})}{2!} f'''(\eta) \eta'(x).$$

Спасает ситуацию то, что погрешность нужно оценивать в узлах интерполирования. Так при $x = x_k$ и $x = x_{k+1}$ два слагаемых из трех в этой формуле обращаются в нуль

$$\varepsilon_1(x_k) = \frac{dR_1(x_k)}{dx} = -\frac{h}{2} f''(\eta), \quad (39)$$

$$\varepsilon_1(x_{k+1}) = \frac{dR_1(x_{k+1})}{dx} = \frac{h}{2} f''(\eta), \quad (40)$$

Выражения (39) и (40) задают погрешность численного дифференцирования для формул (34) и (35) соответственно. Аналогично продифференцируем погрешность $R_2(x)$

$$R_2(x) = \frac{(x - x_k)(x - x_{k+1})(x - x_{k+2})}{3!} f'''(\eta),$$

последовательно подставляя в результат $x = x_k$, $x = x_{k+1}$ и $x = x_{k+2}$

$$\varepsilon_2(x_k) = \frac{dR_1(x_k)}{dx} = \frac{h^2}{3} f'''(\eta), \quad (41)$$

$$\varepsilon_2(x_{k+1}) = \frac{dR_1(x_{k+1})}{dx} = -\frac{h^2}{6} f'''(\eta), \quad (42)$$

$$\varepsilon_2(x_{k+2}) = \frac{dR_1(x_{k+2})}{dx} = \frac{h^2}{3} f'''(\eta), \quad (43)$$

и определяя, таким образом, погрешность численного дифференцирования для формул (36) – (38) соответственно. Легко заметить, что на меньшую погрешность можно рассчитывать, используя формулу (37), которая и является наиболее популярной на практике. Формулы же (36) и (38) используются для дифференцирования в начале и в конце таблицы соответственно.

Если интерполяционный полином второй степени продифференцировать дважды, то получается простейшая формула для второй производной

$$\frac{d^2 f(x_{k+1})}{dx^2} \approx \frac{f_{k+2} - 2f_{k+1} + f_k}{h^2}. \quad (44)$$

Предложенный подход можно было бы развивать и дальше, привлекая новые узлы интерполирования и дифференцируя полиномы более высоких степеней, но мы ограничимся рассмотренными формулами.

Практически важным является вопрос о выборе шага h для формул численного дифференцирования. Ограничение сверху накладывается величиной погрешности (39) – (43), а снизу – точностью задания табличных данных для $f(x)$, что, в свою очередь, определяется, например, результатами эксперимента или ошибками округления (разрядной сеткой компьютера).

Влияние погрешности задания функции на точность.

В качестве примера вновь обратимся к простейшей формуле для первой производной (34). Пусть в ней значение f_{k+1} определено с погрешностью Δ_{k+1} , а значение f_k – с погрешностью Δ_k . Тогда общая погрешность $\varepsilon(h)$ складывается из двух погрешностей

$$\varepsilon(h) = \varepsilon_1(h) + \varepsilon_2(h),$$

первая из которых $\varepsilon_1(h)$ задается формулой (39) и примерно линейно убывает с уменьшением шага h , а вторая зависит от Δ_k и Δ_{k+1} . Оценивая полную погрешность сверху, получаем

$$|\varepsilon(h)| \leq |\varepsilon_1(h)| + |\varepsilon_2(h)| = \frac{h}{2} |f''(\eta)| + \frac{|\Delta_{k+1} - \Delta_k|}{h}. \quad (45)$$

Налицо две противоречивые тенденции: с уменьшением h роль первого слагаемого, определяемого погрешностью формулы дифференцирования, уменьшается, а роль второго слагаемого возрастает. Таким образом, недопустимо использование как слишком большого шага h , так и слишком малого. Оптимальное значение шага h_{opt} отвечает ситуации, когда оба слагаемых равны друг другу. Все сказанное и отражает, на первый взгляд, парадоксальный рис. 4 оценки сверху для $\varepsilon(h)$.

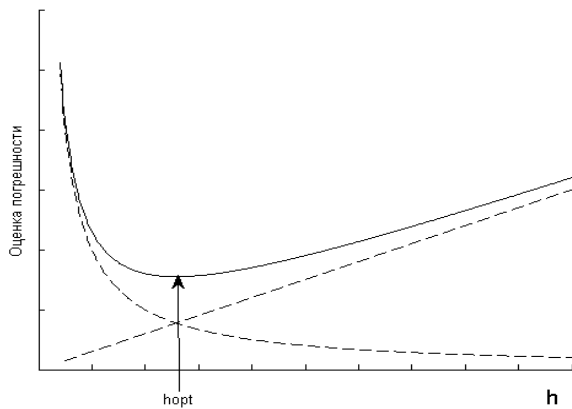


Рис. 4. Оценка погрешности (45)

На практике в точном определении h_{opt} нет необходимости, важно лишь знать о характере зависимости (45). В заключение напомним, что величины Δ_k , Δ_{k+1} и их вклад в общую погрешность численного дифференцирования могут быть как относительно малы, если определяются разрядной сеткой компьютера, так и весьма велики, если являются следствием погрешности измерения экспериментальных данных.