11. Адаптивные квадратурные формулы. Подпрограмма **QUANC8**

Андрей Бареков Ярослав Пылаев
По лекциям Устинова С.М.

February 28, 2020

Программа адаптируется к виду функции и использует переменный шаг интегрирования. Маленький там, где функция меняется быстро и обладает большими производными, и относительно большой там, где меняется медленно.
Программа **QUANC8** минимизирует объём вычислений при заданных ограничений на погрешность.

1 Особенности работы QUANC8

Рассмотрим промежуток длиной h_i внутри промежутка $[a, b]$ и введем обозначения:

- I_i - точное значение интеграла на этом промежутке
- P_i - значение интеграла, вычисленное по несоставной квадратурной формуле Ньютона-Котеса с 9 узлами
- Q_i - промежуток разделили пополам, на каждой половине применили ту же формулу, результаты сложили (по сути, это составная квадратурная формула с вдвое большим количеством узлов).

Для составных формул Ньютона-Котеса формула погрешности имеет следующий общий вид:

$$\alpha \frac{(b-a)^{p+1}}{N^p} f^{(p)}(\eta) \qquad (*)$$

Для формулы Ньютона-Котеса с 9 узлами $p = 10$. Исходя из формулы погрешности:

$$(I_i - P_i) \approx 2^p (I_i - Q_i)$$
$$I_i = \frac{2^p Q_i - P_i}{2^p - 1}$$

Тогда:

$$I_i - Q_i = \frac{2^S Q_i - P_i}{2^S - 1} - Q_i = \frac{Q_i - P_i}{2^S - 1} = \underbrace{\frac{Q_i - P_i}{1023}}_{\text{Для QUANC8}}$$

Интеграл на промежутке h_i считается вычисленным, если величина $\left|\frac{Q_i - P_i}{1023}\right|$ меньше заданной величины. Требование к погрешности в программе реализуется следующим образом:

$$\left|\frac{Q_i - P_i}{1023}\right| \leq \frac{h_i}{b-a} \max\{\varepsilon_A, \varepsilon_R \times \widetilde{I}\}$$

ε_A - абсолютная погрешность,
ε_R - относительная погрешность,

$\widetilde{I}$ - грубая оценка интеграла.
Пользователь выбирает один из трёх вариантов:

1. $\varepsilon_R = 0, \varepsilon_A \neq 0 \Rightarrow$ множитель $\frac{h_i}{b-c}$ отражает вклад в общую погрешность этого промежутка; контролируется *абсолютная* погрешность.

2. $\varepsilon_A = 0, \varepsilon_R \neq 0$ контролируется*относительная* погрешность.

3. $\varepsilon_A \neq 0, \varepsilon_R \neq 0$. Контролируется *смешанная* погрешность.

2 Работа алгоритма

1. Вычисляем P_i и Q_i для всего промежутка $[a; b]$

2. Если контроль погрешности не прошел, значение функции на правой половине промежутка запоминаем, и повторяем процедуру для левой половины промежутка.

Деление пополам продолжается до тех пор, пока крайний слева промежуток не будет принят. Затем обращаются к ближайшему правому соседу.

В программе предусмотрено два *ограничения от зацикливания*:

1. Промежуток разрешается делить пополам не более 30 раз. Если после 30-тикратного деления контроль погрешности не прошёл, результат принимается, количество промежутков с непрошедшим контролем записывается в целую часть переменной **FLAG**.

2. Задаётся максимальное значение вычислений функции $f(x)$. Если эта величина достигнута, то информация о точке x^*, в которой программа прервала свою работу, записывается в дробную часть переменной **FLAG**.

Программа имеет параметры:

$$(\underbrace{\underbrace{FUN}_{f(x)}, \underbrace{A, B}_{[a,b]}, \underbrace{EA, ER}_{\varepsilon_A, \varepsilon_R}}_{\text{входные}}, \underbrace{RESULT}_{\text{знач. интегр.}}, \underbrace{ERR}_{\text{оценка погр.}}, \underbrace{NOFUN}_{\text{кол-во выч.} f(x)}, FLAG)$$