Семестр 2 (2020), занятие 5. Приближенное вычисление значения функции

Требуется вычислить значение функции f в точке x с точностью $\varepsilon > 0$ путем суммирования ряда Тейлора

$$f(x) \;\approx\; \sum_{i=1}^{k} a_i, \quad |a_k| < \varepsilon.$$

Вычисление $\sin x$ и $\cos x$

С использованием формул приведения задача вычисления $\sin x$ ($\cos x$) может быть сведена к вычислению $\sin y$ ($\cos y$), где $y \in [0, \pi/4]$.

Вычисление очередных слагаемых ряда Тейлора для $\sin y$ проводить по следующим рекуррентным соотношениям

$$\begin{aligned} a_1 &= y, \\ a_{i+1} &= -a_i \tfrac{y^2}{2i(2i+1)}. \end{aligned}$$

Вычисление очередных слагаемых ряда Тейлора для $\cos y$ проводить по следующим рекуррентным соотношениям

$$\begin{aligned} a_1 &= 1, \\ a_{i+1} &= -a_i \tfrac{y^2}{2i(2i-1)}. \end{aligned}$$

Вычисление e^x

При $x < 0$ необходимо воспользоваться соотношением

$$e^x \;=\; \frac{1}{e^{-x}},$$

которое позволяет свести задачу к случаю положительного аргумента.

Пусть $x > 0$. Представим аргумент в виде суммы целой и дробной частей

$$x \;=\; [x] + \{x\}.$$

Тогда

$$e^x \;=\; e^{[x]}\, e^{\{x\}}.$$

Каждый множитель данного произведения вычисляется отдельно. Значение $e^{[x]}$ вычисляется при помощи алгоритма быстрого возведения в степень. Заголовочный файл `math.h` содержит константу `M_E`, которую можно использовать в качестве приближенного значения числа e.

Величина $e^{\{x\}}$ вычисляется путем суммирования ряда Тейлора с использованием рекуррентного соотношения

$$a_{i+1} \;=\; a_i \,\frac{\{x\}}{i}.$$

Вычисление $\ln x$

При $0 < x < 1$, используя соотношение

$$\ln\frac{1}{x} \;=\; -\ln x,$$

сведем задачу к случаю $x > 1$.

Пусть $x > 1$. Подберем натуральное число n таким образом, чтобы имело место представление

$$x \;=\; (\sqrt{e})^n\,(1+y), \quad |y| < 1.$$

Тогда искомое значение вычисляется как

$$\ln x \;=\; \frac{n}{2} + \ln(1+y).$$

Далее необходимо воспользоваться разложением

$$\ln(1+y) \;=\; y - \frac{y^2}{2} + \frac{y^3}{3} - \frac{y^4}{4} + \ldots$$

Реализации вычислительной процедуры

Вычислительная процедура должна иметь следующие входные и выходные параметры.

Входные параметры.

1. `x` (тип `double`) – точка, в которой вычисляется значение функции.
2. `e` (тип `double`) – требуемая точность.
3. `N` (тип `int`) – максимальное количество шагов.
4. `tr` (тип `int`) – ненулевое значение предписывает выводить информационные сообщения о состоянии шага вычислительного процесса.

Выходные параметры.

1. `st` (тип `int`) – статус выполнения процедуры. Возможные значения:

– 0 – решение было успешно найдено;

- -1 – некорректные входные параметры;

- -2 – превышен лимит шагов.

2. `fx` (тип `double`) – значение функции в точке `x`.

3. `n` (тип `int`) – количество выполненных шагов.

Все выходные параметры должны быть оформлены в виде струтуры. Ниже приведен возможный вариант прототипа функции.

```
typedef double (*fun_t)(double);

typedef struct {
    int    st;
    double fx;
    int    n;
} res_t;

void root(double x,
          double e,
          int    N,
          int    tr,
          res_t *p);
```

Интерфейс программы

При запуске программы все входные параметры передаются через аргументы командной строки. Запуск программы без аргументов командной строки приводит к выводу на экран инструкции по использованию программы.

Программа должна выводить на экран значения выходных параметров вычислительной процедуры.

В случае успешного вычисления также должны быть напечатаны:

- значение функции, вычисленное с помощью процедуры из стандартной библиотеки;

- модуль разности между значением функции, вычисленным с помощью реализованной процедуры, и значением функции, вычисленным с помощью процедуры из стандартной библиотеки.