Лекция 1.

Литература:

1. Самарский А.А. Введение в численные методы
2. Бахвалов Н.С. Численные методы
3. Форсайт Дж. Машинные методы математических вычислений

Начнем с простого и в тоже время (поучительного, предостерегающего) настораживающего примера для того чтобы уяснить те проблемы, которые появляются при применении вычислительных машин для решения задач математической физики

Пример:

$$\begin{aligned} x_1 - x_2 - x_3 \cdots - x_n &= -1 \\ x_2 - x_3 \cdots - x_n &= -1 \\ \cdots \\ x_{n-1} - x_n &= -1 \\ x_n &= 1 \end{aligned}$$

Точное решение этой системы линейных уравнений $x = (x_1, x_2, \cdots, x_n) = (0, 0, \cdots 0, 1)$ нетрудно получить методом исключения, начиная с последнего уравнения.

Допустим, что при в n-компоненте правой части или, что то же самое при вычислении x_n допущена ошибка $\epsilon = 10^{-15}$!!, достаточно маленькая, т.е. $x_n - 1 + \epsilon$. Иными словами, решаем систему

$$\begin{aligned} \overline{x}_1 - \overline{x}_2 - \overline{x}_3 \cdots - \overline{x}_n &= -1 \\ \overline{x}_2 - \overline{x}_3 \cdots - \overline{x}_n &= -1 \\ \cdots \\ \overline{x}_{n-1} - \overline{x}_n &= -1 \\ \overline{x}_n &= 1 + \epsilon \end{aligned}$$

Вопрос, как различаются решения этих двух систем. Заметим, что исходя из жизненного опыта, мы ожидаем небольшое различие.

Итак $|x - \overline{x}| =?$

Введем обозначения $\overline{x} = x + \delta$, $\delta = (\delta_1, \delta_2, \cdots, \delta_n) =?$.

Для нахождения δ составим систему, полученную вычитанием первой из второй

$$\begin{aligned} \delta_1 - \delta_2 - \delta_3 \cdots - \delta_n &= 0 \\ \delta_2 - \delta_3 \cdots - \delta_n &= 0 \\ \cdots \\ \delta_{n-1} - \delta_n &= 0 \\ \delta_n &= \epsilon \end{aligned}$$

Решая ее методом исключения, начиная с последнего получаем $\delta_n = \epsilon$,$\delta_{n-1} = \delta_n = \epsilon$,$\delta_{n-2} = \delta_{n-1} + \delta_n = 2\epsilon$ и т.д. $\delta_1 = 2^{n-2}\epsilon$.

Если $\epsilon = 10^{-15}$, n=102,

$\Longrightarrow \delta_1 = 2^{100}10^{-15} \simeq 10^{30}10^{-15} = 10^{15}$!!!!

Заметим - погрешности разного рода присутствуют во всех задачах и следовательно основной вопрос вычислений что мы находим в результате ???

В любой вычислительной задаче требуется по некоторым входным данным найти ответ на поставленный вопрос (решить систему линейных или нелинейных уравнений, либо дифференциальных уравнений и т.д.). Если ответ дается точно, то погрешность отсутствует. Но как правило, ответ удается найти лишь с некоторой погрешностью, которые обусловлены тремя основными причинами:

1. Неопределенностью задания входных данных, которые приводят к неопределенности в ответе, т.е. ответ указывается с некоторой погрешностью. Этот тип погрешности называют неустранимой.

2. Фиксируя входные данные можно забыть о неустранимой погрешности, но как правило решение поставленной задачи находится приближенным методом и следовательно имеем погрешность, как результат приближенности метода. Такую погрешность называют погрешностью метода.

3. Метод решения (даже точный) в ЭВМ реализуется всегда неточно из-за погрешности, возникающей в результате округления чисел на реальном компьютере - погрешность округления.

Погрешность результата - погрешность из-за неустранимой погрешности, погрешности метода и погрешности округлений.

Разберем эти понятия более подробно.

1. Неустранимая погрешность. Пусть вычислительная задача состоит в вычислении некоторой функции $y = f(x)$ при некотором заданном $x = t$. Число t и отображение f в этом случае служат входными данными, y(t) - решением.

Пусть отображение f заданно приближенно $f(x) \simeq sin(x)$, причем

$$|f(x) - sin(x)| < \epsilon \qquad x \subset [0, \pi/2]$$

Значение аргумента t получается приближенным измерением в некоторой точке t^*, причем

$$|t - t^*| < \delta, \qquad 0 < t < \pi/2$$

Из рисунка видно, что величина $y = f(x)$ может оказаться в любой точке отрезка [a,b], где

$$a = sin(t^* - \delta) - \epsilon$$

$$b = sin(t^* + \delta) + \epsilon$$

и, следовательно, неустранимая погрешность может быть оценена неравенством

$$|y - y^*| < |b - a|$$

2. Погрешность приближенного метода - будет разбираться отдельно для каждого метода в курсе на пример $\int \to \sum$

3. Погрешность округлений - источником погрешности является способ представления чисел в ЭВМ. В математике при вычислениях мы работаем либо с N-множеством натуральных чисел, либо с R-множеством вещественных чисел.

Вычислительная машина все вычисления проводит во множестве целых чисел $M_{N_0} \in N_0$,где N_0-конечное число, при этом необходимо помнить, что если $I_1, I_2 \in M_{N_0}$, то $I_1 + I_2$ вообще говоря может не принадлежать множеству M_{N_0}.

Вычисления с вещественными числами в ЭВМ проводятся во множестве M_R, которое не только ограничено по модулю сверху, но и снизу, а также любое вещественное число аппроксимируется.

Для аппроксимации используются различные системы. Разберем подробнее - так называемую нормализованную систему с плавающей запятой.

В этой системе некоторое число b выбирается в качестве основания системы счисления. Удобно $b = 2, 8, 16$ можно и 10. t-разрядов из чисел a_i, $i = \overline{1, t}$ по основанию b образуют дробную часть или мантиссу числа, кроме того, задается порядок числа $e < E \sim 200$.

Число x представляется в форме

$x = \pm 0.a_1 a_2 \dots a_t b^{\pm e} = f b^{\pm e}$, где e-порядок числа, а f удовлетворяет неравенству $b^{-1} \leq f \leq 1$ - признак представления в нормализованной форме.

Основные операции которые выполняет ЭВМ при вычислениях $= \pm, \times, /$, которые будем обозначать в машинном множестве через $\oplus, \otimes, \oslash$ соответственно.

Очевидно, что в результате выполнения арифметических операций над числами x, y из M_R как правило $\notin M_R$ и подлежит округлению. В основе операций округления следующее

Утверждение 1.

Пусть $A = max_{x \in M_R}|x|, B = min_{x \in M_R}|x|$. Пусть далее $y \in R$ и $B < |y| < A$, тогда существует $\overline{y} \in M_R$ такое, что $\overline{y} = y(1 + \tau)$, где $\tau < b_{1-t}/2 = \epsilon$-машинное эрсилон. Величину τ называют погрешностью округлений.

Легко видеть, что во множестве M_R не выполняются законы ассоциативности т.е.

$(x \bigoplus y) \bigoplus z \neq x \bigoplus (y \bigoplus z)$

$(x \bigotimes y) \bigotimes z \neq x \bigotimes (y \bigotimes z)$

Если порядок результата операции $x \bigotimes y$ больше E -наступает явление переполнения.

Если порядок результата операции $x \bigotimes y$ меньше ϵ -наступает явление исчезновения порядка !!!.

Наиболее часто выполняемая операция при вычислениях - операция вычисления $\sum_{i=1}^{n} a_i b_i$. Грубая оценка погрешности округления при вычислении этого выражения содержится в

Утверждение 2. Пусть $n\epsilon < 2h(1+h)$,h- заданное число, тогда

$|\sum_{i=1}^{n} a_i b_i - \sum_{i=1}^{n} \bigoplus a_i \bigotimes b_i| \leq$

$(n+1)\epsilon(1+h)\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} b_i^2}$

Примеры:

1. Исчезновение порядка

Пусть требуется вычислить $\prod_{i=1}^{5} q_i, q_1 = b^{-\frac{E}{2}} = q_2 = q_3, q_4 = b^{\frac{E}{2}} = q_5$

$\prod_{i=1}^{5} \bigotimes q_i = q_1 \bigotimes q_2 \bigotimes q_3 \bigotimes q_4 \bigotimes q_5 = \underbrace{b^{-\frac{E}{2}} \bigotimes b^{-\frac{E}{2}}}_{=0} \bigotimes b^{-\frac{E}{2}} \bigotimes b^{\frac{E}{2}} \bigotimes b^{\frac{E}{2}} = 0$

С другой стороны, если поменять порядок следования, то имеем

$\prod_{i=1}^{5} \bigotimes q_i = (q_1 \bigotimes q_4) \bigotimes (q_2 \bigotimes q_5) \bigotimes q_3 = b^{\frac{E}{2}}$

2. Вычислить $\sum_{i=1}^{n=10^7+1} a_i, a_1 = 1, a_2 = a_3 = \ldots = a_n = 10^{-7}, b = 10, t = 7$

В нормализованной системе счисления при сложении чисел с разными порядками предварительно происходит их выравнивание т.е. сумма вычисляется по правилу

$\sum_{i=1}^{n} a_i = 0.110^1 + \underbrace{0.0000000110^1}_{=0} + \ldots = 1$ С другой стороны, меняя порядок вычислений, будем иметь

$\sum_{i=1}^{n} a_i = 1 + (\sum_{i=2}^{n} a_i = 2$

Правило: при умножении числа с различными порядками сортируют и порядок умножения меняют так, чтобы умножать маленькое на большое; при сложении сортируют и меняют порядок действий так чтобы складывать числа одного порядка.

Приведенный предварительный анализ показывает, что любой вычислительный процесс должен удовлетворять требованию не накопления ошибок при вычислениях.

На вычислительные алгоритмы накладывают требование

Определение Вычислительный процесс называют устойчивым, если ошибки округления при выполнении вычислений не накапливаются, в противном случае вычислительный процесс называют неустойчивым.

Примеры:

1. Требуется вычислить y_n по рекуррентной формуле $y_{i+1} = y_i + d$, где y_0, d - заданные числа. Пусть при вычислении y_i допустили погрешность δ_i, $\overline{y_i} = y_i + \delta_i$.

$\overline{y_{i+1}} = \overline{y_i} + d = y_i + \delta_i + d = y_{i+1} + \delta_i = y_{i+1} + \delta_{i+1}$

или

$\delta_{i+1} = \delta_i$, следовательно, процесс устойчив.

2. Требуется вычислить y_n по рекуррентной формуле $y_{i+1} = qy_i$, где y_0, q - заданные числа. Пусть при вычислении y_i допустили погрешность δ_i, $\overline{y_i} = y_i + \delta_i$.

$\overline{y_{i+1}} = q\overline{y_i} = q(y_i + \delta_i) = y_{i+1} + q\delta_i = y_{i+1} + \delta_{i+1}$

или $\delta_{i+1} = q\delta_i$, следовательно, процесс устойчив, если $|q| < 1$ и неустойчив если $|q| > 1$.