ЛЕКЦИЯ 1.2 ВЫЧИСЛИТЕЛЬНЫЕ ЗАДАЧИ И МЕТОДЫ

1. Вычислительные задачи, их корректность и обусловленность

Постановка вычислительной задачи подразумевает задание исходных числовых данных (входные данные) и формулировку требуемого числового результата (решения). Интуитивно понятно, что для задачи желательно наличие свойств, которые делают её актуальной и решаемой, например, наличие решение, его нечувствительность к погрешностям исходных данных. Сейчас мы строго определим эти свойства.

Будем называть решение вычислительной задачи *устойчивым по отношению к малым возмущениям исходных данных***,** если последним соответствуют малые погрешности решения, не выше порядка погрешностей исходных данных. Таким образом, небольшие «сдвиги» исходных данных приводят к небольшим «сдвигам» устойчивого решения. Решение вычислительной задачи в этом случае *непрерывным образом* зависит от входных данных. В самом простом скалярном случае это означает, что для любого сколь угодно малого положительного числа ε существует положительное число $\delta(\varepsilon)$, что для любого входного значения x^* с $\Delta(x^*) < \delta(\varepsilon)$ соответствующее решение y^* удовлетворяет условию $\Delta(y^*) < \varepsilon$. Другими словами, для любой наперёд заданной погрешности решения можно подобрать такую точность исходного значения, что при любых исходных значениях такой точности погрешность решения будет меньше заданного уровня.

Вычислительная задача называется *корректной,* если:

1) Для любого допустимого набора исходных данных *существует единственное ее решение*.

2) Это решение *устойчиво* по отношению к малым возмущениям исходных данных.
При невыполнении хотя бы одного из этих условий задача называется *некорректной*.

Как уже пояснялось, устойчивость решения означает, что любую наперед заданную ее точность можно достичь, обеспечив надлежащую точность исходных данных. Численной мерой степени устойчивости решения является *число обусловленности***,** равное коэффициенту *возможного* роста погрешности решения по отношению к погрешностям исходных данных. В самом простом скалярном случае его можно определить так. Пусть $\overline{\Delta}x^*$,

$\overline{\delta}x^*$ — оценки абсолютной и относительной погрешностей исходной величины, $\overline{\Delta}(y^*)$, $\overline{\delta}(y^*)$ — оценки погрешностей полученного по x^* решения y^*. Тогда, если известно, что

$$\overline{\Delta}y^* \leq \vartheta_\Delta \overline{\Delta}x^*, \overline{\delta}y^* \leq \vartheta_\delta \overline{\delta}x^*,$$

то ϑ_Δ — абсолютное число обусловленности, ϑ_δ — относительное число обусловленности. Например, из приведённых в лекции 1.1 оценок погрешностей функции следует, что для задачи приближенного вычисления функции одного аргумента $y = f(x)$

$$\vartheta_\Delta = |f'(x^*)|, \vartheta_\delta = \left|\frac{x^* f'(x^*)}{y^*}\right|.$$

Если числа обусловленности вычислительной задачи малы (близки к нулю), то ее называют *хорошо обусловленной,* если близки к единице или больше единицы, то *плохо обусловленной*.

Пример. Рассмотрим задачу решения систему линейных уравнений

$$\begin{cases} x_1 + x_2 = 0{,}5, \\ (1+\varepsilon)x_1 + x_2 = 1{,}5. \end{cases}$$

Её нетрудно решить аналитически:

$$x_1 = \frac{1}{\varepsilon}, x_2 = 0{,}5 - \frac{1}{\varepsilon}.$$

Коэффициент при x_1 – это $1 + \varepsilon$. Малое изменение параметра ε матрицу системы меняет незначительно, но приводит к большим изменениям в решении. Если $\varepsilon = 0$, то система не имеет решения, а если $\varepsilon \neq 0$, то имеет. Небольшое изменение параметра приводит к большим изменениям: наличию или отсутствию решения. Если считать параметр ε входным данным, то решение системы неустойчиво.

Можно рассмотреть пример другой системы:

$$\begin{cases} 1{,}1x_1 + x_2 = 1{,}1, \\ x_1 + x_2 = 1, \end{cases}$$

имеющей решение $x_1 = 1, x_2 = 0$. Немного изменим второе уравнение:

$$(1+\varepsilon)x_1 + x_2 = 1.$$

Тогда система имеет такое решение:

$$x_1 = \frac{1}{1 - 10\varepsilon}, x_2 = \frac{-11\varepsilon}{1 - 10\varepsilon}.$$

Если $\varepsilon = 0{,}001$, то $x_1 = 1{,}01, x_2 = -0{,}011$; если $\varepsilon = 0{,}002$, то $x_1 = 1{,}02$, $x_2 = -0{,}022$. Если считать параметр ε входным данным, $\varepsilon = 0{,}001$, $\varepsilon^* = 0{,}002$, то $\Delta\varepsilon^* = 0{,}001$, $\Delta x_2^* = 0{,}011$,

$\vartheta_\Delta = 11$. Абсолютное число обусловленности компоненты решения x_2 велико, вычислительная задача плохо обусловлена.

Из приведенных примеров можно сделать вывод, что даже решение систем линейных уравнений может быть неустойчивой задачей. Приведенные небольшие изменения параметров могли быть либо ошибками входных данных, либо погрешностями округления или машинного представления чисел. Задаче обусловленности систем линейных уравнений уделяется большое внимание. На практике системы могут иметь десятки и сотни неизвестных.

Многие практические задачи являются некорректными. Разработаны методы решения различных классов некорректных задач. Некорректные задачи могут быть из самых разных отраслей, не только из прикладной математики или механики. Существуют *методы регуляризации* некорректных задач. Существенный вклад в изучение некорректных задач внесли советские математики. Например, выделяется класс некорректных по А.Н. Тихонову задач. Один из подходов к регуляризации некорректных задач – это учет априорной информации. Такой учет позволяет сузить множество решений задачи Y до некоторого множества M, в котором решение устойчиво.

2. Классификация вычислительных методов

Методы численного решения задач делятся на следующие классы:

1) методы эквивалентных преобразований;

2) методы аппроксимации;

3) прямые методы;

4) итерационные методы.

Кратко охарактеризуем каждый класс.

В *методах эквивалентных преобразований* исходная задача заменяется ей эквивалентной, имеющей то же решение. Это делается в тех случаях, когда последняя проще или для нее имеется готовый эффективный метод решения. Например, от задачи приближенного решения уравнения $f(x) = 0$ можно перейти к эквивалентной задаче поиска абсолютного минимума функции $g(x) = (f(x))^2$, которая может решаться более простыми методами.

В *методах аппроксимации* исходная задача заменяется другой, решение которой близко к решению исходной в некотором смысле. При этом большое значение имеет оценка погрешности аппроксимации, т.е. различия между решениями исходной и аппроксимирующей задач. Например, для приближенного вычисления интеграла переходят к задаче вычисления интеграла от интерполяционного многочлена, построенного приближающего подынтегральную функцию. Эта задача имеет простые решения, причем подбором многочлена можно добиться любой степени точности.

В *прямых методах* теоретически точное решение получается за конечное число шагов. Эти методы составляют весьма немногочисленную группу. Например, точные формулы корней некоторых видов нелинейных уравнений, методы Гаусса, прогонки решения систем линейных алгебраических уравнений.

В *итерационных методах* строится последовательность приближений - *итераций* - к точному решению - *итерационная последовательность*. Каждая последующая итерация получается применением к одной или нескольким предыдущим однотипного набора операций - *итерационный шаг*. Понятно, что для запуска итерационного процесса нужно задать начальные приближения. Если итерационный шаг использует одно приближение, вычисленное на предыдущем, то метод называется *одношаговым*, если несколько, то *многошаговым*. Для запуска одношагового метода задается одна начальная итерация, многошагового — несколько. Если это невозможно, то сначала с помощью какого-либо одношагового метода вычисляется нужное количество итераций, а потом запускается многошаговый метод.

Если итерационная последовательность сходится к точному решению, то говорят, что метод *сходится*, в противном случае — *расходится*. Множество начальных приближений, для которых метод сходится, называется его *областью сходимости*. Очень важно правильно определить область сходимости.

Итерационный процесс прекращается, когда достигается заданная точность решения, например, две последовательно вычисленных итерации совпадают в пределах какой-то величины, определяемой по наперед заданной точности. Таким образом, для итерационного метода необходимы:

1. Точное конструктивное описание итерационного шага, например, расчётная формула очередной итерации;
2. Описание области сходимости или определение условий сходимости;
3. Оценка погрешности текущей итерации для критерия останова вычислений.

Примерами итерационных методов являются методы простой итерации, Ньютона решения нелинейных уравнений и систем, Зейделя, Якоби решения систем линейных уравнений, методы решения дифференциальных уравнений.

Пример. Рассмотрим задачу извлечения квадратного корня $a = \sqrt{x}$, что равносильно нахождению такого числа, a, что $a^2 = x$. Задача заменяется эквивалентной – решением нелинейного уравнения. Приближенно вычислять корень уравнения можно по итерационной схеме

$$a = \frac{1}{2}\left(a + \frac{x}{a}\right).$$

Берем начальное приближение к корню, подставляем его в правую часть формулы и получаем первое приближение, подставив первое приближение туда же, получим второе. Для $x = 2$ имеем последовательность приближений 1 (начальное), $1{,}5$, $1{,}41667$, $1{,}41421$. Все значащие цифры верные. Итерационная последовательность сходится при любом положительном начальном приближении. Данный метод извлечения корня был известен еще в Древней Греции. Его называют методом Герона извлечения корней. Надо отметить, что понятие итераций и сходимости известны и разрабатываются с древних времен. И до сих пор бурно развиваются. Компьютер – хороший инструмент для проведения итерационных вычислений.

3. Корректность алгоритмов

Вычислительный алгоритм – это точное предписание действий над входными данными, задающее вычислительный процесс, направленный на преобразование произвольных входных данных в полностью определенный этими входными данными результат.

Вычислительный алгоритм называется *корректным*, если:

1. Он позволяет после выполнения конечного числа элементарных для вычислительной машины операций преобразовать любое входное допустимое данное

$x \in X$ в единственный результат $y \in Y$ (X – множество допустимых входных данных, Y - множество решений).

2. Полученное с его помощью решение устойчиво по отношению к малым возмущениям входных данных.
3. Результат обладает вычислительной устойчивостью.

Устойчивость по входным данным означает то, что результат непрерывным образом зависит от входных данных при условии, что отсутствует вычислительная погрешность. *Вычислительная устойчивость* означает стремление к нулю вычислительной погрешности (т.е. устранимой, обусловленной приближённостью реализуемого метода и машинной погрешностью) при стремлении к нулю машинной погрешности.

Алгоритм называется *вычислительно устойчивым*, если таков результат его работы при любых допустимых входных данных. Алгоритм *устойчив по входным данным*, если результат его работы устойчив по входным данным. Если алгоритм устойчив по входным данным и вычислительно устойчив, то он называется *устойчивым*. Таким образом, алгоритм корректен, если он даёт единственное решение за конечное число шагов при любых допустимых исходных данных и устойчив.

Вычислительный алгоритм называют *хорошо обусловленным*, если малые относительные погрешности округления приводят к малой относительной вычислительной погрешности результата. Численной характеристикой обусловленности алгоритма служит *число обусловленности*. В скалярном случае оно определяется так: если $\delta y^* \leq \vartheta_A \cdot \varepsilon_M$, где δy^* - вычислительная относительная погрешность решения y^*, ε_M – относительная погрешность округления, то ϑ_A - число обусловленности алгоритма. Если ϑ_A близко к нулю, то алгоритм хорошо обусловлен, если $\vartheta_A \gg 1$, то плохо обусловлен.