Лекция 3

Тема 3. Вычислительные задачи, методы и алгоритмы

§ 3.1. Корректность вычислительной задачи

Будем считать, что постановка вычислительной задачи включает в себя задание *множества допустимых входных данных X и множества возможных решений Y*. Цель вычислительной задачи состоит в нахождении решения $y \epsilon Y$ по заданному входному данному $x \epsilon X$. Для простоты ограничимся рассмотрением задач, в которых входные данные и решения могут быть только числами, наборами чисел (векторами, матрицами, последовательностями) и функциями. Предположим, что для оценки величин погрешностей приближенных входных данных x^* и приближенного решения y^* введены абсолютные и относительные погрешности $\Delta(x^*)$, $\Delta(y^*)$, $\delta(x^*)$, $\delta(y^*)$, а также их границы $\bar{\Delta}(x^*)$, $\bar{\Delta}(x^*)$, $\bar{\delta}(x^*)$, $\bar{\delta}(y^*)$. Определения этих величин в случае, когда x и y – числа, были даны выше. В тех случаях, когда x и y не являются числами, эти характеристики погрешностей также можно ввести естественным образом, например, так, как указывалось в первом разделе курса.

Перейдем теперь к определению понятия корректности вычислительной задачи, которое было впервые сформулировано французским математиком Ж.Адамаром и развито затем российским математиком И.Г.Петровским. Вычислительная задача называется *корректной*, если выполнены следующие три требования:

1). Ее решение $y \epsilon Y$ существует при любых входных данных $x \epsilon X$;

2). Это решение единственно;

3). Решение устойчиво по отношению к малым возмущениям входных данных.

В том случае, когда хотя бы одно из этих требований не выполнено, задача называется *некорректной*.

Существование решения вычислительной задачи – естественное требование к ней. Отсутствие решения может свидетельствовать, например, о непригодности принятой математической модели либо о неправильной постановке задачи. Иногда отсутствие решения является следствием неправильного выбора множества допустимых входных данных X или множества возможных решений Y.

Пример 3.1. Рассмотрим задачу о решении квадратного уравнения

$$x^2 + bx + c = 0. \qquad (3.1)$$

Старший коэффициент $a = 1$. Если *считать входным данным пару коэффициентов* b, c и искать решение в множестве вещественных чисел, то существование решений

$$x_1 = (-b - \sqrt{b^2 - 4c})/2, \quad x_2 = (-b + \sqrt{b^2 - 4c})/2 \qquad (3.2)$$

будет гарантировано только в том случае, если ограничить множество входных данных коэффициентами, удовлетворяющими условию $b^2 - 4c \geq 0$. Если же расширить множество возможных решений и считать, что корни (3.2) могут принимать комплексные значения, то задача будет иметь решение при любых b, c.

Единственность решения для некоторых задач является естественным свойством; для других задач решение может не быть единственным. Например, квадратное уравнение (3.1) имеет два корня (3.2). Как правило, если задача имеет реальное содержание, то неединственность может быть ликвидирована введением дополнительных ограничений на решение (т.е. сужением множества Y). В некоторых случаях проблема снимается тем, что признается целесообразным найти набор всех решений, отвечающих входным данным, и тогда за решение принимается этот набор. Например, для квадратного уравнения (3.1) решением можно назвать пару (x_1, x_2).

Решение y вычислительной задачи называется ***устойчивым*** *по входным данным* x, если оно зависит от входных данных непрерывным образом. Это означает, что для любого $\varepsilon > 0$ существует $\delta = \delta(\varepsilon) > 0$ такое, что всякому исходному данному x^*, удовлетворяющему условию $\Delta(x^*) < \delta$, отвечает приближенное решение y^*, для которого $\Delta(y^*) < \varepsilon$. Таким образом, для устойчивой вычислительной задачи ее решение теоретически можно найти со сколь угодно высокой точностью ε, если обеспечена достаточно высокая точность δ входных данных.

Неустойчивость решения *y* означает, что существует такое $\varepsilon_0 > 0$, что какое было малое $\delta > 0$ ни было задано, найдутся такие исходные данные x^*, что $\Delta(x^*) < \delta$, но $\Delta(y^*) \geq \varepsilon_0$.

Приведем несколько примеров устойчивых и неустойчивых задач.

Пример 3.2. Задача вычисления корней квадратного уравнения (3.1) устойчива, т.к. корни (3.2) являются непрерывными функциями коэффициентов b и c.

Пример 3.3. Задача о вычислении ранга матрицы в общем случае неустойчива. В самом деле, для матрицы $A = \begin{matrix} 1 & 0 \\ 0 & 0 \end{matrix}$ ранг равен 1, поскольку detA = 0 и существует ненулевой элемент a_{11}=1. Однако сколь угодно малое возмущение коэффициента a_{22} на величину $\varepsilon \neq 0$ приводит к матрице $A_\varepsilon = \begin{matrix} 1 & 0 \\ 0 & \varepsilon \end{matrix}$, для которой $\det A_\varepsilon = \varepsilon \neq 0$ и следовательно ранг равен 2.

Пример 3.4. Покажем, что задача вычисления определенного интеграла $I = \int_a^b f(x)dx$ устойчива.

Пусть $f^*(x)$ – приближенно заданная интегрируемая функция и $I^* = \int_a^b f^*(x)\,dx$. Определим абсолютную погрешность функции f^* с помощью равенства $\Delta(f^*) = \max_{x\epsilon[a,b]} |f(x) - f^*(x)|$. Так как

$$\Delta(I^*) = |I - I^*| = \left|\int_a^b \left(f(x) - f^*(x)\right)dx\right| \leq (b-a)\,\Delta(f^*), \qquad (3.3)$$

то для любого $\varepsilon > 0$ неравенство $\Delta(I^*) < \varepsilon$ будет выполнено, если потребовать выполнение условия $\Delta(f^*) < \delta = \varepsilon / (b-a)$.

Пример 3.5. Покажем, что задача вычисления производной $u(x) = f'(x)$ приближенно заданной функции является неустойчивой.

Пусть $f^*(x)$ – приближенно заданная на отрезке $[a, b]$ непрерывно дифференцируемая функция и $u^*(x) = (f^*)'(x)$. Определим абсолютные погрешности с помощью равенств

$$\Delta(f^*) = \max_{x\epsilon[a,b]} |f(x) - f^*(x)| \quad \text{и}$$

$$\Delta(u^*) = \max_{x\epsilon[a,b]} |u(x) - u^*(x)|. \qquad (3.4)$$

Возьмем, например, $f^*(x) = f(x) + \alpha \sin(x / \alpha^2)$, где $0 < \alpha << 1$. Тогда $u^*(x) = u(x) + \alpha^{-1} Cos\,(x/\alpha^2)$ и $\Delta(u^*) = \alpha^{-1}$, в то время как $\Delta(f^*) = \alpha$. Таким образом, сколь угодно малой погрешности задания функции $f(x)$ может отвечать сколь угодно большая погрешность

производной $f'(x)$, т.е. задача дифференцирования является неустойчивой.

Часто требование малости абсолютной погрешности является неоправданным или трудно проверяемым. В таких случаях полезно рассмотреть *относительную устойчивость* решения, определение которой отличается от данного выше определения абсолютной устойчивости только тем, что $\Delta(x^*)$ и $\Delta(y^*)$ заменяются на $\delta(x^*)$ и $\delta(y^*)$ соответственно.

§ 3.2. Обусловленность вычислительной задачи

Определения. Пусть вычислительная задача корректна (ее решение существует, единственно и устойчиво по входным данным). Теоретически наличие у задачи устойчивости означает, что ее решение может быть найдено со сколь угодно малой погрешностью, если только гарантировать, что погрешности входных данных достаточно малы. Однако на практике погрешности входных данных не могут быть сделаны сколь угодно малыми, точность их ограничена. Даже то, что входные данные нужно ввести в ЭВМ, означает, что их точность будет заведомо ограничена конечной величиной разрядной сетки. В реальности уровень ошибок в исходной информации будет заведомо выше.

Для ответа на вопрос, как повлияют малые, но конечные погрешности входных данных на решение, введем новые понятия.

Под *обусловленностью вычислительной задачи* понимают чувствительность ее решения к малым погрешностям исходных данных.

Задачу называют *хорошо обусловленной*, если малым погрешностям исходных данных отвечают малые погрешности решения, и *плохо обусловленной*, если возможны сильные изменения решения.

Часто оказывается возможным ввести количественную меру степени обусловленности вычислительной задачи – *число обусловленности.* Эту величину можно интерпретировать как коэффициент возможного возрастания погрешностей в решении по отношению к вызвавшим их погрешностям входных данных.

Пусть между абсолютными погрешностями входных данных x и решения y установлено неравенство

$$\Delta(y^*) \le \nu_\Delta \, \Delta(x^*). \qquad (3.5)$$

Тогда величина ν_Δ называется *абсолютным числом обусловленности*. Если же установлено неравенство

$$\delta(y^*) \le \nu_\delta \, \delta(x^*) \qquad (3.6)$$

между относительными ошибками входных данных и решения, то величину ν_δ называют *относительным числом обусловленности.* В неравенствах (3.5), (3.6) вместо погрешностей $\Delta(x^*)$, $\Delta(y^*)$, $\delta(x^*)$, $\delta(y^*)$, могут фигурировать их границы $\bar{\Delta}(x^*)$, $\bar{\Delta}(x^*)$, $\bar{\delta}(x^*), \bar{\delta}(y^*)$. Обычно под числом обусловленности понимают одну из величин ν_Δ или ν_δ, причем выбор бывает ясен из смысла задачи. Чаще все же под числом обусловленности понимают относительное число обусловленности. Для плохо обусловленной задачи $\nu >> 1$. В некотором смысле неустойчивость задачи – это крайнее проявление плохой обусловленности, отвечающее значению $\nu = \infty$. Конечно, ν – это максимальный коэффициент возможного возрастания уровня ошибок, и для конкретных входных данных действительный коэффициент возрастания может оказаться существенно меньше. Однако значение $\nu >> 1$ все же свидетельствует о реальной возможности существенного роста ошибок. Грубо говоря, если $\nu_\delta \approx 10^N$, то порядок N показывает число верных цифр, которое может быть утеряно в результате по сравнению с числом верных цифр входных данных.

Каково то значение ν, при котором следует признать задачу плохо обусловленной? Ответ на этот вопрос существенно зависит, с одной стороны, от предъявляемых требований к точности решения и, с другой стороны, от уровня обеспечиваемой точности входных данных. Например, если требуется найти решение с точностью 0.1%, а входная информация задается с точностью 0.02%, то уже значение $\nu=10$ свидетельствует о плохой обусловленности. Однако (при тех же требованиях к точности результата) гарантия, что входные данные задаются с точностью не ниже 0.0001%, означает, что и при $\nu = 10^3$ задача хорошо обусловлена.

Важным примером плохо обусловленной задачи является вычитание приближенных чисел одного знака. Для нее относительное число обусловленности $\nu = |a + b| / |a – b|$.

Приведем **примеры хорошо и плохо обусловленных задач**.

1.Обусловленность задачи вычисления функции одной переменной. Пусть задача состоит в вычислении по заданному x значения $y = f(x)$ дифференцируемой функции $f(x)$.

В силу формул (2.21) и (2.22) для этой задачи имеем

$$\nu_\Delta \approx |f'(x)|, \tag{3.7}$$

$$\nu_\delta \approx \frac{|x||f'(x)|}{|f(x)|}. \tag{3.8}$$

Воспользуемся этими формулами для оценки обусловленности задачи вычисления некоторых простейших функций.

Пример 3.6. Для задачи вычисления значения функции y = e^x в силу формулы (3.8) относительное число обусловленности ν_δ=|x| и при реальных вычислениях эта величина не может быть очень большой. Например, при вычислении экспоненты на компьютере типа IBM PC всегда |x| < 88, так как в противном случае возможно переполнение или антипереполнение. Следовательно, задача вычисления значения этой функции хорошо обусловлена, однако в случае $10 < |x| < 10^2$ следует ожидать потери 1 – 2 верных значащих цифр по сравнению с числом верных цифр аргумента x. Подчеркнем, что эта потеря точности объективно обусловлена погрешностью задания аргумента и не связана с используемым алгоритмом.

Пример 3.7. Для задачи вычисления значения функции $y = \sin x$ в силу формулы (3.7) имеем $\nu_\Delta = |\cos x| \leq 1$, что говорит о хорошей абсолютной обусловленности этой задачи при всех x. Однако если важен результат с определенным числом верных знаков, то нужно исследовать относительную обусловленность. Согласно формуле (3.8) имеем

$$\nu_\delta = |x\ ctg\ x|.$$

Очевидно, что $\nu_\delta \to \infty$ при $x \to \pi\kappa$ (для $\kappa = \pm1, \pm2, \pm3, ...$), т.е. при $x \approx \pi\kappa$ задача обладает плохой относительной обусловленностью.

2.Обусловленность задачи вычисления интеграла $\int_a^b f(x)dx$. Как следует из оценки (3.3) в этом случае абсолютное число обусловленности имеет вид $\nu_\Delta = (b - a)$. Если же перейти к рассмотрению относительных погрешностей и положить $\delta(f^*) =$

$\max_{a\le x\le b} | f^*(x) - f(x)|/|f(x)|$ для тех *x*, где *f(x)* $\neq$ *0*, то используя неравенство

$\Delta(I^*) \le \int_a^b |f^*(x) - f(x)|dx \le \int_a^b |f(x)|dx \cdot \delta(f^*)$, получим оценку

$$\delta(I^*) \le \nu_\delta \, \delta(f^*), \qquad (3.9)$$

в которой $\nu_\delta \le \int_a^b |f(x)|dx / |\int_a^b f(x)\,dx|$.

Если подынтегральная функция знакопостоянна, то $\nu_\delta = 1$ и задача хорошо обусловлена. Если же функция f на $[a, b]$ принимает значения разных знаков, то $\nu_\delta > 1$. Для сильно осциллирующих (колеблющихся) около нуля функций может оказаться, что $\nu_\delta >> 1$ и тогда задача вычисления интеграла является плохо обусловленной.

§ 3.3. Вычислительные методы

Обсудив некоторые важные особенности вычислительных задач, обратим внимание на методы, которые используются в вычислительной математике для преобразования задач к виду, удобному для реализации на ЭВМ, и которые позволяют конструировать вычислительные алгоритмы. Эти методы конкретизируют тот общий метод вычислительной математики, о котором шла речь во введении. Будем называть эти методы *вычислительными.*

С некоторой степенью условности можно разбить вычислительные методы на следующие классы:

1).**Методы эквивалентных преобразований.** Эти методы позволяют заменить исходную задачу другой, имеющей то же решение. Выполнение эквивалентных преобразований оказывается полезным, если новая задача проще исходной или для нее существует известный метод решения или готовая программа. Эти методы мы будем рассматривать в дальнейшем при решении конкретных задач, например, нелинейных уравнений.

2).**Методы аппроксимации.** Эти методы позволяют аппроксимировать исходную задачу другой, решение которой в определенном смысле близко к решению исходной задачи. Погрешность, возникающая при такой замене, называется

погрешностью аппроксимации. Принято говорить, что метод сходится, если погрешность аппроксимации стремится к нулю при стремлении параметров метода к некоторому предельному значению. Методы аппроксимации будем применять, например, при численном интегрировании.

3). **Итерационные методы.** Это специальные методы построения последовательных приближений к решению задачи. Применение итерационного метода начинают с задания начальных приближений. Для получения каждого из последующих приближений выполняют однотипный набор действий с использований найденных ранее приближений – *итерацию.* Неограниченное продолжение этого *итерационного процесса* позволяет построить бесконечную последовательность приближений к решению – *итерационную последовательность*. Если эта последовательность сходится к решению задачи, то говорят, что *итерационный метод сходится.* Множество начальных приближений, для которых метод сходится, называется *областью сходимости метода.* Итерационные методы широко используются при решении самых разнообразных задач с применением ЭВМ. Мы будем их подробно рассматривать, например, при решении нелинейных уравнений.

Практическая реализация итерационных методов всегда связана с необходимостью выбора *критерия окончания итерационного процесса.* Для формирования критерия окончания итерационного процесса по достижению заданной точности используют *апостериорные оценки погрешности* – неравенства, в которых величина погрешности оценивается через величины, получаемые в ходе вычислительного процесса.

§ 3.4. Корректность вычислительных алгоритмов

Вычислительный метод, доведенный до степени детализации, позволяющей реализовать его на ЭВМ, принимает форму вычислительного алгоритма.

Определим вычислительный алгоритм как точное предписание действий над входными данными, задающее вычислительный процесс, направленный на преобразование произвольных входных данных в полностью определяемый этими входными данными результат.

К вычислительным алгоритмам предъявляется ряд весьма жестких требований. Первое из них – корректность алгоритма. Будем называть вычислительный алгоритм *корректным*, если выполнены три условия:

1). Он позволяет после выполнения конечного числа элементарных для ЭВМ операций преобразовать любое входное данное $x \in X$ в результат y;

2) результат y устойчив по отношению к малым возмущениям входных данных (это требование аналогично требованию устойчивости вычислительной задачи) при отсутствии вычислительной погрешности;

3) результат y обладает вычислительной устойчивостью. При вводе в ЭВМ входных данных и в процессе вычислений неизбежно появление вычислительной погрешности, величина которой определяется машинной точностью $\varepsilon_{\text{м}}=2^{-t}$, где t – разрядность представления мантиссы. Назовем алгоритм *вычислительно устойчивым*, если вычислительная погрешность результата стремится к нулю при $\varepsilon_{\text{м}}\to 0$.

Если хотя бы одно из перечисленных условий не выполняется, то будем называть алгоритм *некорректным.*

§ 3.5. Обусловленность вычислительных алгоритмов

По аналогии с понятием обусловленности вычислительной задачи можно ввести понятие обусловленности вычислительного алгоритма, отражающее чувствительность результата работы алгоритма к малым, но неизбежным ошибкам округления. Вычислительно устойчивый алгоритм называют *хорошо обусловленным*, если малые относительные погрешности округления (характеризуемые машинной точностью $\varepsilon_{\text{м}}$) приводят к малой относительной вычислительной погрешности $\delta(y^*)$ результата y^*, и *плохо обусловленным*, если вычислительная погрешность может быть недопустимо большой.

Если $\delta(y^*)$ и $\varepsilon_{\text{м}}$ связаны неравенством $\delta(y^*) \le \nu_{\text{A}}\varepsilon_{\text{м}}$, то число ν_{A} следует называть *числом обусловленности вычислительного алгоритма*. Для плохо обусловленных алгоритмов $\nu_{\text{A}}>>1$.