Лекция 3
Точность вычислительного эксперимента

Вопросы лекции

1. Приближенные числа.
2. Погрешности вычислений.
3. Свойства расчетных задач и численных методов.

Вопрос 1.
Приближенные числа

1. Числа с плавающей точкой. Числа могут быть представлены в памяти компьютеров различными способами. Современные компьютеры (процессоры), как правило, позволяют обрабатывать *целые* числа, а также дробные числа в форме с *плавающей точкой* [1)].

Как известно, множество целых чисел бесконечно. Однако процессор из-за ограниченности его *разрядной сетки* может оперировать лишь с некоторым конечным подмножеством этого множества. В современных компьютерах для хранения целого числа обычно отводится 4 *байта* памяти [2)], что позволяет представлять целые числа, находящиеся примерно в диапазоне от $-2 \cdot 10^9$ до $2 \cdot 10^9$.

При решении научно-технических задач в основном используются действительные (вещественные) числа. В компьютерах они представляются в форме с плавающей точкой. Десятичное число D в этой форме записи имеет вид $D = \pm m \cdot 10^n$, где m и n — соответственно *мантисса* числа и его *порядок*. Например, число -273.9 можно записать в виде: $-2739 \cdot 10^{-1}$, $-2.739 \cdot 10^2$, $-0.2739 \cdot 10^3$. Последняя запись — нормализованная форма числа с плавающей точкой. Таким образом, если представить мантиссу числа в виде $m = 0.\, d_1 d_2 \ldots d_k$, то при $d_1 \neq 0$ получим *нормализованную форму* числа с плавающей точкой. В дальнейшем, говоря о числах с плавающей точкой, будем иметь в виду именно эту форму. Обычная же запись числа в виде -273.9 называется формой записи с *фиксированной точкой*. В настоящее время такое представление используется в компьютерах, как правило, только на этапе ввода и вывода чисел.

Все сказанное выше о числах с плавающей точкой распространяется и на числа, записанные в других системах счисления. Число A в системе счисления с основанием α можно представить в виде $A = \pm 0.a_1 a_2 \ldots a_k \cdot \alpha^n$, где $a_1, a_2, \ldots, a_k$ — целые числа из диапазона $0, \ldots, \alpha - 1$. Из этой записи следует, что подмножество действительных чисел, с которым оперирует конкретный компьютер, не является бесконечным: оно конечно и определяется разрядностью k, а также границами порядка n_1, n_2 ($n_1 \leqslant n \leqslant n_2$). Можно показать, что это подмножество содержит

$$N = 2(\alpha - 1)(n_2 - n_1 + 1)\alpha^{k-1} + 1 \tag{1.1}$$

чисел, наименьшим и наибольшим по модулю являются соответственно числа

$$M_0 = (\alpha - 1)\alpha^{n_1 - 1} \quad \text{и} \quad M_\infty = (1 - \alpha^{-k})\alpha^{n_2}, \tag{1.2}$$

называемые *машинным нулем* и *машинной бесконечностью*.

Границы порядка n_1, n_2 определяют ограниченность действительных чисел по величине, а разрядность k — дискретность распределения их на отрезке числовой оси. Например, в случае десятичных чисел при четырехразрядном представлении все значения, находящиеся в промежутке $(0.28505, 0.28515)$, представляются числом 0.2851 (при выполнении округления). Если к этому числу 0.2851 прибавить число, меньшее по модулю половины единицы последнего разряда (т. е. меньшее по модулю 0.00005), в результате получится то же самое число 0.2851.

В настоящее время большинство производителей процессоров в основном придерживаются стандарта IEEE 754 [1)] для арифметических операций над двоичными числами с плавающей точкой. Данный стандарт предусматривает наличие, в частности, двух двоичных ($\alpha = 2$) форматов: с одинарной точностью и с двойной точностью. Приведем для этих форматов размер отводимой памяти, значения k, n_1, n_2 и приближенные значения M_0 и M_∞. Заметим, что стандарт IEEE 754 предусматривает обработку чисел, меньших по модулю M_0, но не меньших M_0^*, правда, с меньшей разрядностью k.

Точность	Байты	k	n_1	n_2	M_0	M_0^*	M_∞
Одинарная	4	24	-125	128	$1.2 \cdot 10^{-38}$	$1.4 \cdot 10^{-45}$	$3.4 \cdot 10^{38}$
Двойная	8	53	-1021	1024	$2.2 \cdot 10^{-308}$	$4.9 \cdot 10^{-324}$	$1.8 \cdot 10^{308}$

Поскольку для человека более удобной является десятичная система счисления, возникает вопрос о том, скольким десятичным разрядам соответствует указанная двоичная разрядность k. Можно считать, что k соответствует 6 – 9 десятичным разрядам при одинарной и 15–17 разрядам при двойной точности.

В современных языках программирования предусмотрены типы данных для представления вещественных чисел с одинарной и двойной точностью. Например, в языке Си это типы float и double, в языке Паскаль — single и double, в языке Фортран — real и double precision. Обычно эти представления соответствуют стандарту IEEE 754.

2. Понятие погрешности. Различают два вида погрешностей — абсолютную и относительную. *Абсолютная погрешность* некоторого числа равна разности между его истинным значением и приближенным значением, полученным в результате вычисления или измерения. *Относительная погрешность* — это отношение абсолютной погрешности к приближенному значению числа.

Таким образом, если a — приближенное значение числа x, то выражения для абсолютной и относительной погрешностей запишутся соответственно в виде

$$\Delta x = x - a, \quad \delta x = \Delta x / a.$$

К сожалению, истинное значение величины x обычно неизвестно. Поэтому приведенные выражения для погрешностей практически не могут быть использованы. Имеется лишь приближенное значение a и нужно найти его *предельную погрешность* Δa, являющуюся верхней оценкой модуля абсолютной погрешности, т. е. $|\Delta x| \leqslant \Delta a$. В дальнейшем значение Δa принимается в качестве абсолютной погрешности приближенного числа a. В этом случае истинное значение x находится в интервале $(a - \Delta a, a + \Delta a)$.

Для приближенного числа, полученного в результате округления, абсолютная погрешность Δa принимается равной половине единицы последнего разряда числа. Например, значение $a = 0.734$ могло быть получено округлением чисел 0.73441, 0.73353 и др. При этом $|\Delta x| \leqslant 0.0005$, и полагаем $\Delta a = 0.0005$. Если при вычислениях на компьютере округление не производится, а цифры, выходящие за разрядную сетку машины, отбрасываются, то максимально возможная погрешность результата выполнения операции в два раза больше по сравнению со случаем округления.

Приведем примеры оценки абсолютной погрешности при некоторых значениях приближенной величины a:

a	51.7	-0.0031	16	16.00
Δa	0.05	0.00005	0.5	0.005

Предельное значение относительной погрешности — отношение предельной абсолютной погрешности к абсолютной величине приближенного числа:

$$\delta a = \Delta a / |a|.$$

Например, $\delta(-2.3) = 0.05/2.3 \approx 0.022\,(2.2\,\%)$. Заметим, что погрешность округляется всегда в сторону увеличения. В данном случае $\delta(-2.3) \approx 0.03$.

Приведенные оценки погрешностей приближенных чисел справедливы, если в записи этих чисел все значащие цифры верные. Напомним, что *значащими цифрами* считаются все цифры данного числа, начиная с первой ненулевой цифры. Например, в числе 0.037 две значащие цифры: 3 и 7, а в числе 14.80 все четыре цифры значащие. Кроме того, при изменении формы записи числа (например, при записи в форме с плавающей точкой) число значащих цифр не должно меняться, т. е. нужно соблюдать равносильность преобразований. Например, записи $7500 = 0.7500 \cdot 10^4$ и $0.110 \cdot 10^2 = 11.0$ равносильные, а записи $7500 = 0.75 \cdot 10^4$ и $0.110 \cdot \cdot 10^2 = 11$ неравносильные.

3. Действия над приближенными числами. Сформулируем правила оценки предельных погрешностей при выполнении операций над приближенными числами.

При сложении или вычитании чисел их абсолютные погрешности складываются. При умножении или делении чисел друг на друга их относительные погрешности складываются. При возведении в степень приближенного числа его относительная погрешность умножается на показатель степени.

Для случая двух приближенных чисел a и b эти правила можно записать в виде формул

$$\begin{gathered}\Delta(a \pm b) = \Delta a + \Delta b, \quad \delta(a \cdot b) = \delta a + \delta b, \\ \delta(a/b) = \delta a + \delta b, \quad \delta(a^k) = k\delta a.\end{gathered} \tag{1.3}$$

Относительная погрешность суммы положительных слагаемых заключена между наибольшим и наименьшим значениями относительных погрешностей этих слагаемых. Действительно, пусть $a > 0$, $b > 0$, $m = \min(\delta a, \delta b)$, $M = \max(\delta a, \delta b)$. Тогда

$$\delta(a+b) = \frac{\Delta(a+b)}{a+b} = \frac{\Delta a + \Delta b}{a+b} = \frac{a\delta a + b\delta b}{a+b} \leqslant \frac{aM + bM}{a+b} = M.$$

Аналогично, $\delta(a+b) \geqslant m$. На практике для оценки погрешности принимается наибольшее значение M.

П р и м е р 1. Найти относительную погрешность функции

$$y = \sqrt{\frac{a+b}{x^3(1-x)}}.$$

Используя формулы (1.3), получаем

$$\delta y = \frac{1}{2}\left[\delta(a+b) + 3\delta x + \delta(1-x)\right] = \frac{1}{2}\left[\frac{\Delta a + \Delta b}{|a+b|} + 3\,\frac{\Delta x}{|x|} + \frac{\Delta x}{|1-x|}\right].$$

Полученная оценка относительной погрешности содержит в знаменателе выражение $|1-x|$. Ясно, что при $x \approx 1$ можно получить очень большую погрешность.

Запишем выражение для относительной погрешности разности двух чисел в виде

$$\delta(a-b) = \frac{\Delta(a-b)}{|a-b|} = \frac{\Delta a + \Delta b}{|a-b|}.$$

При $a \approx b$ эта погрешность может быть сколь угодно большой.

Пример 2. Пусть $a = 2520$, $b = 2518$. В этом случае имеем абсолютные погрешности исходных данных $\Delta a = \Delta b = 0.5$ и относительные погрешности $\delta a \approx \delta b = 0.5/2518 \approx 0.0002$ $(0.02\,\%)$. Относительная погрешность разности равна

$$\delta(a-b) = \frac{0.5+0.5}{2} = 0.5\ (50\,\%).$$

Следовательно, при малых погрешностях в исходных данных мы получили весьма неточный результат. Поэтому при организации вычислительных алгоритмов следует избегать вычитания близких чисел; при возможности алгоритм нужно видоизменить во избежание потери точности на некотором этапе вычислений.

Из рассмотренных правил следует, что при сложении или вычитании приближенных чисел желательно, чтобы эти числа обладали одинаковыми абсолютными погрешностями, т. е. одинаковым числом разрядов после десятичной точки.

Например, $38.723 + 4.9 = 43.6$; $425.4 - 0.047 = 425.4$.

Учет отброшенных разрядов не повысит точность результатов. При умножении и делении приближенных чисел количество значащих цифр выравнивается по наименьшему из них.

Рассмотрим функцию одной переменной $y = f(x)$. Пусть a — приближенное значение аргумента x, Δa — его абсолютная погрешность. Абсолютную погрешность функции можно считать ее приращением, которое она испытывает при изменении аргумента на Δa. Это приращение можно заменить дифференциалом: $\Delta y \approx dy$. Тогда для оценки абсолютной погрешности получим выражение $\Delta y = |f'(a)|\Delta a$.

Аналогичное выражение можно записать для функции нескольких аргументов. Например, оценка абсолютной погрешности функции $u = f(x, y, z)$, приближенные значения аргументов которой соответственно a, b, c, имеет вид

$$\Delta u = |f'_x(x, y, z)|\Delta a + |f'_y(x, y, z)|\Delta b + |f'_z(x, y, z)|\Delta c.$$

Здесь Δa, Δb, Δc — абсолютные погрешности аргументов. Относительная погрешность находится по формуле

$$\delta u = \frac{\Delta u}{|f(a, b, c)|}.$$

Полученные соотношения можно использовать для вывода оценки погрешности произвольной функции . Например, при $c = a - b$ получаем $\Delta c = |c'_a|\Delta a + |c'_b|\Delta b = \Delta a + \Delta b$.

Вопрос 2

Погрешности вычислений.

1. Источники погрешностей. На некоторых этапах решения задачи на компьютере могут возникать погрешности, искажающие результаты вычислений. Оценка степени достоверности получаемых результатов является важнейшим вопросом при организации вычислительных работ. Это особенно важно при отсутствии опытных или других данных для сравнения, которое могло бы в некоторой степени показать надежность используемого численного метода и достоверность получаемых результатов.

Рассмотрим источники погрешностей на отдельных этапах решения задачи.

Математическая модель, принятая для описания данного процесса или явления, может внести существенные погрешности, если в ней не учтены какие-либо важные черты рассматриваемой задачи. В частности, математическая модель может прекрасно работать в одних условиях и быть совершенно неприемлемой в других; поэтому важно правильно учитывать область ее применимости.

Исходные данные задачи часто являются основным источником погрешностей. Вместе с погрешностями, вносимыми математической моделью, их называют *неустранимыми погрешностями*, поскольку они не могут быть уменьшены вычислителем ни до начала решения задачи, ни в процессе ее решения. Проведенный ранее анализ оценки погрешностей при выполнении арифметических операций показывает, что следует стремиться к тому, чтобы все исходные данные были примерно одинаковой точности. Сильное уточнение одних исходных данных при наличии больших погрешностей в других, как правило, не приводит к повышению точности результатов.

Численный метод также является источником погрешностей. Это связано, например, с заменой интеграла суммой, с усечением рядов при вычислениях значений функций, с интерполированием табличных данных и т. п. Как правило, *погрешность численного метода* регулируема, т. е. теоретически она может быть уменьшена до любого значения путем изменения некоторого параметра (например, шага интегрирования, числа членов усеченного ряда и т. п.). Погрешность метода обычно стараются довести до величины, в несколько раз меньшей неустранимой погрешности. Дальнейшее снижение погрешности не приведет к повышению точности результатов, а лишь увеличит стоимость расчетов из-за необоснованного увеличения объема вычислений.

При вычислениях с помощью компьютера неизбежны *погрешности округлений*, связанные с ограниченностью разрядной сетки машины. При обычном округлении (которое, как правило, и реализуется в компьютерах) максимальная относительная погрешность есть

$$\delta_{\max} = 0.5\alpha^{1-k}, \tag{1.5}$$

где α — основание системы счисления, k — количество разрядов мантиссы числа. При простом отбрасывании лишних разрядов эта погрешность увеличивается вдвое.

Вычислим по формуле (1.5) максимальную погрешность округления $\delta_{\max}$ для чисел, представленных в форматах с одинарной и двойной точностью стандарта IEEE 754. Имеем: $\alpha = 2$ в обоих случаях, для одинарной точности $k = 24$ и $\delta_{\max} \approx 6 \cdot 10^{-8}$, для двойной точности $k = 53$ и $\delta_{\max} \approx 10^{-16}$.

Несмотря на то, что при решении больших задач выполняются миллиарды и триллионы операций, это вовсе не означает механического умножения погрешности при одном округлении на число операций, так как при отдельных действиях погрешности могут компенсировать друг друга (например, при сложении чисел разных знаков). Вместе с тем иногда погрешности округлений в сочетании с плохо организованным алгоритмом могут сильно исказить результаты.

Перевод чисел из одной системы счисления в другую также может быть источником погрешности из-за того, что основание одной системы счисления не является степенью основания другой (например, 10 и 2). Это может привести к тому, что в новой системе счисления число становится иррациональным.

Например, число 0.1 при переводе в двоичную систему счисления примет вид $0.000\,1100\,1100\ldots$ Может оказаться, что с шагом 0.1 нужно при вычислениях пройти отрезок $[0, 1]$ от $x = 1$ до $x = 0$; десять шагов не дадут точного значения $x = 0$.

2. Уменьшение погрешностей. При рассмотрении погрешностей результатов арифметических операций отмечалось, что вычитание близких чисел приводит к увеличению относительной погрешности; поэтому в алгоритмах следует избегать подобных ситуаций. Рассмотрим также некоторые другие случаи, когда можно избежать потери точности правильной организацией вычислений.

Пусть требуется найти сумму пяти четырехразрядных чисел: $S = 0.2764 + 0.3944 + 1.475 + 26.46 + 1364$. Складывая все эти числа, а затем округляя полученный результат до четырех значащих цифр, получаем $S = 1393$. Однако при вычислении на компьютере округление происходит после каждого сложения. Предполагая условно сетку четырехразрядной, проследим за вычислением на компьютере суммы чисел от наименьшего к наибольшему, т. е. в порядке их записи: $0.2764 + 0.3944 = 0.6708$, $0.6708 + 1.475 = 2.156$, $2.156 + 26.46 = 28.62$, $28.62 + 1364 = 1393$; получили $S_1 = 1393$, т. е. верный результат. Изменим теперь порядок вычислений и начнем складывать числа последовательно от последнего к первому: $1364 + 26.46 = 1390$, $1390 + 1.475 = 1391$, $1391 + 0.3944 = 1391$, $1391 + 0.2764 = 1391$; здесь окончательный результат $S_2 = 1391$, он менее точный.

Анализ процесса вычислений показывает, что потеря точности здесь происходит из-за того, что прибавления к большому числу малых чисел не происходит, поскольку они выходят за рамки разрядной сетки ($a+b=a$ при $a \gg b$) . Этих малых чисел может быть очень много, но на результат они все равно не повлияют, поскольку прибавляются по одному, что и имело место при вычислении S_2. Здесь необходимо придерживаться правила, в соответствии с которым сложение чисел нужно проводить по мере их возрастания. В машинной арифметике из-за погрешности округления существен порядок выполнения операций, и известные из алгебры законы коммутативности (и дистрибутивности) здесь не всегда выполняются.

При решении задачи на компьютере нужно использовать подобного рода маленькие хитрости для улучшения алгоритма и снижения погрешностей результатов. Например, при вычислении на компьютере значения $(a+x)^2$ величина x может оказаться такой, что результатом сложения $a+x$ получится a (при $x \ll a$); в этом случае может помочь замена $(a+x)^2 = a^2 + + 2ax + x^2 = a(a+2x) + x^2$. Действительно, теперь к a прибавляется не x, а $2x$. Если же при $x \ll a$ вычисляется величина $(a+x)^2 - a^2$, то целесообразно привести ее к виду $2ax + x^2$, избежав тем самым вычитания близких величин.

Рассмотрим еще один важный пример — использование рядов для вычисления значений функций. Запишем, например, разложение функции $\sin x$ по степеням аргумента:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

По признаку Лейбница остаток сходящегося знакочередующегося ряда, т. е. погрешность суммы конечного числа членов, не превышает значения первого из отброшенных членов (по абсолютной величине).

Вычислим значение функции $\sin x$ при $x = 0.5236$ (30°). Члены ряда, меньшие 10^{-4}, не будем учитывать. Вычисления проведем с четырьмя верными знаками. Получим

$$\sin 0.5236 = 0.5236 - 0.2392 \cdot 10^{-1} + 0.3279 \cdot 10^{-3} = 0.500.$$

Это отличный результат в рамках принятой точности.

Зная из курса высшей математики, что это разложение синуса справедливо при любом значении аргумента $(-\infty < x < +\infty)$, используем его для вычисления функции при $x = 6.807$ (390°).

Зная из курса высшей математики, что это разложение синуса справедливо при любом значении аргумента $(-\infty < x < +\infty)$, используем его для вычисления функции при $x = 6.807$ (390°).

Опуская вычисления, получаем $\sin 6.807 \approx 0.5167$.

Относительная погрешность составляет здесь около $3\,\%$ (вместо ожидаемого значения $0.01\,\%$ по признаку Лейбница). Это объясняется погрешностями округлений и способом суммирования ряда (слева направо, без учета величины членов).

Не всегда помогает и повышенная точность вычислений. В частности, для данного ряда при $x = 25.6563\ldots$ $(1470^\circ = 4 \cdot 360^\circ + 30^\circ)$ даже при учете членов ряда до 10^{-8} и вычислениях с восемью значащими цифрами в результате аналогичных вычислений (суммирование слева направо) получается результат, не имеющий смысла: $\sin x \approx 129$.

В программах, использующих степенные ряды для вычисления значений функций, могут быть приняты различные меры по предотвращению подобной потери точности. Так, влияние погрешностей округления существенно уменьшается, если $|x| < 1$. Действительно, при вычислении x^k допускается абсолютная погрешность

$$\Delta(x^k) = x^k \delta(x^k) = x^k k \delta x \ ,$$

которая при невыполнении неравенства $|x| < 1$ может стать неприемлемо большой.

Для тригонометрических функций можно использовать формулы приведения, благодаря чему аргумент будет находиться на отрезке $[0,\ 1]$. При вычислении экспоненты аргумент x можно разбить на сумму целой и дробной частей ($e^x = e^{n+a} = e^n \cdot e^a$, $0 < a < 1$) и использовать разложение в ряд только для e^a, а e^n вычислять умножением. Таким образом, при организации вычислений можно своевременно распознать «подводные камни», дающие потерю точности, и попытаться затем исправить положение.

3. О решении квадратного уравнения. Мы убедились в том, что при численном решении задач на компьютере вычислителя ожидают всякие «ловушки», которые могут привести к заметной потере точности результатов или даже к прекращению счета. Хорошей иллюстрацией к этому является анализ алгоритма решения такой простой задачи, как решение квадратного уравнения $ax^2 + bx + c = 0$. Его корни определяются соотношениями

$$x_1 = \frac{-b - \sqrt{D}}{2a}, \quad x_2 = \frac{-b + \sqrt{D}}{2a}, \quad D = b^2 - 4ac. \tag{1.6}$$

Из анализа этих формул видно, что здесь имеется ряд особенностей вычислительного характера, которые необходимо иметь в виду при составлении алгоритма.

Рассмотрим простейший случай $a = 0$. Здесь уравнение становится линейным, и его единственный корень есть $x = -c/b$, если $b \neq 0$. При $a = b = 0$ и $c \neq 0$ уравнение не имеет решения, а в случае $a = b = c = 0$ его решением будет любое число. Заметим, что в машинной арифметике редко получаются точно нулевые значения. Поэтому коэффициенты можно сравнивать не с нулем, а с некоторой малой величиной ε. Это в свою очередь порождает ряд ситуаций, зависящих от соотношения между коэффициентами.

Далее необходимо предусмотреть разветвление алгоритма в зависимости от знака дискриминанта D: $D > 0$ — корни действительные (см. (1.6)); $D = 0$ — корни равные: $x_1 = x_2 = -b/(2a)$; $D < 0$ — корни комплексные: $x_{1,2} = R \pm iI$, где $R = -b/(2a)$, $I = \sqrt{-D}/(2a)$.

Менее очевидным вопросом является возможность появления погрешностей в зависимости от соотношения между коэффициентами уравнения. Рассмотрим один из важнейших случаев, когда коэффициент b значительно превышает по абсолютной величине остальные. При этом $b^2 \gg 4ac$ и возникает опасность вычитания близких чисел в числителе одного из выражений (1.6) из-за того, что $\sqrt{D} \approx |b|$.

Положение можно исправить разными способами. Например, при $b > 0$ формулу для x_2 можно преобразовать следующим образом:

$$x_2 = \frac{\sqrt{D} - b}{2a} \frac{\sqrt{D} + b}{\sqrt{D} + b} = -\frac{2c}{\sqrt{D} + b}.$$

При $b < 0$ аналогичным способом можно записать формулу для x_1.

Более универсальным способом является использование значения $\operatorname{sign} b$ («знак величины b»):

$$\operatorname{sign} b = \begin{cases} 1, & b \geqslant 0, \\ -1, & b < 0. \end{cases} \tag{1.7}$$

Тогда один из корней может быть вычислен по формуле

$$x_1 = -\frac{b + \operatorname{sign} b \cdot \sqrt{D}}{2a}. \tag{1.8}$$

Выражение для вычисления значения второго корня можно получить с помощью теоремы Виета. Из соотношения $x_1 x_2 = c/a$ следует, что

$$x_2 = \frac{c}{a x_1}. \tag{1.9}$$

Вопрос 3

Свойства расчетных задач и численных методов

Корректность (хорошая обусловленность)

Устойчивость

Сходимость

Задача является *хорошо обусловленной*, если при небольших изменениях входных данных результаты ее решения изменяются незначительно (непрерывная зависимость решения от исходных данных) и при любых исходных данных из возможного диапазона их изменения задача однозначно разрешима.

Пример . Пусть задана система двух линейных алгебраических уравнений с двумя неизвестными:

$$300x_1 + 400x_2 = 700,$$
$$100x_1 + 133x_2 = 233.$$

Система имеет точное решение $x_* = (x_{*1}, x_{*2})^T = (1; 1)^T$.

Пусть одно из исходных данных — число $b = 233$ изменилось на доли процента, и вместо него приняли число $\tilde{b} = 232$.

Тогда получается решение $\tilde{x}_{*1} = -3;\ \tilde{x}_{*2} = 4$.

Таким образом, при изменении b на $0{,}43\%$ $\left(\dfrac{\Delta b}{233}\cdot 100\% = \dfrac{1}{233}\cdot 100\% = 0{,}43\%\right)$ компоненты решения x_{*1}, x_{*2} изменились по модулю соответственно в 3 и 4 раза.

Численный метод называется *устойчивым*, если результаты расчета непрерывно зависят от входных (исходных) данных задачи (т. е. выполняется условие хорошей обусловленности задачи) и погрешность округления, связанная с реализацией численного метода, при заданных пределах изменения параметров численного метода остается ограниченной.

Пусть в результате решения задачи по исходному значению величины x находится значение искомой величины y. Если исходная величина имеет абсолютную погрешность Δx, то решение имеет погрешность Δy. Задача называется *устойчивой* по исходному параметру x, если решение y непрерывно от него зависит, т. е. малое приращение исходной величины Δx приводит к малому приращению искомой величины Δy. Другими словами, малые погрешности в исходной величине приводят к малым погрешностям в решении.

Отсутствие устойчивости означает, что даже незначительные погрешности в исходных данных приводят к большим погрешностям в решении или даже к неверному результату. О неустойчивых задачах также говорят, что они *чувствительны* к погрешностям исходных данных.

Пример неустойчивой задачи.

Рассмотрим квадратное уравнение с параметром a

$$x^2 - 2x + \operatorname{sign} a = 0.$$

Решение этого уравнения в зависимости от значения a таково: $x_1 = x_2 = 1$ при $a \geqslant 0$; $x_{1,2} = 1 \pm \sqrt{2}$ при $a < 0$. Очевидно, что при $a = 0$ сколь угодно малая отрицательная погрешность в задании a приведет к конечной, а не сколь угодно малой погрешности в решении уравнения.

Понятие сходимости. При анализе точности вычислительного процесса одним из важнейших критериев является *сходимость* численного метода. Она означает близость получаемого численного решения задачи к истинному решению.

Рассмотрим понятие *сходимости итерационного процесса.* Этот процесс состоит в том, что для решения некоторой задачи и нахождения искомого значения определяемого параметра (например, корня нелинейного уравнения) строится метод последовательных приближений. В результате многократного повторения этого процесса (или *итераций*) получаем последовательность значений $x_1, x_2, \ldots, x_n, \ldots$ Говорят, что эта последовательность сходится к точному решению $x = a$, если при неограниченном возрастании числа итераций предел этой последовательности существует и равен a: $\lim\limits_{n\to\infty} x_n = a$. В этом случае имеем сходящийся численный метод.

Другой подход к понятию сходимости используется в методах дискретизации. Эти методы заключаются в замене задачи с непрерывными параметрами на задачу, в которой значения функций вычисляются в фиксированных точках. Это относится, в частности, к численному интегрированию, решению дифференциальных уравнений и т. п. Здесь под *сходимостью метода* понимается стремление значений решения дискретной модели задачи к соответствующим значениям решения исходной задачи при стремлении к нулю параметра дискретизации (например, шага интегрирования).