Тема: «Неустойчивость алгоритмов и чувствительность некоторых задач».

1. Для некоторых задач «хороший» ответ невозможно получит никаким алгоритмом, потому что задача чувствительна к малым ошибкам, допущенными при представлении данных и в арифметике.

Рассмотрим пример неустойчивости алгоритма:

$E_n = \int_0^1 x^n e^{x-1} dx$, $n = 1, 2...$

Интегрируя по частям, получаем:

$$E_n = \int_0^1 x^n e^{x-1} dx = x^n e^{x-1}\Big|_0^1 - \int_0^1 n x^{n-1} e^{x-1} dx,$$

$E_n = 1 - nE_{n-1}, n = 2, 3, \ldots ,$ где $E_1 = \frac{1}{e}$. Пусть $\beta = 10$ и $t = 6$,тогда

$E_1 \approx 0.367879, E_2 \approx 0.264242, E_3 \approx 0.207274,$
$E_4 \approx 0.170904, E_5 \approx 0.145480, E_6 \approx 0.127120,$
$E_7 \approx 0.110160, E_8 \approx 0.118720, E_9 \approx -0.0684800 \prec 0.$

Заметим, что единственная ошибка округления — это ошибка в округлении E_1 . Ошибка накапливается и в E_9 примерно равна $9! \times 4.412 \times 10^{-7} \approx 0.1601$. Подлинное значение E_9 0. 0916. Если переписать рекуррентное соотношение в виде: $E_n = \int_0^1 x^n e$,то теперь вычисляя в обратном порядке, ошибки округления будут уменьшаться на каждом шагу. Это и называется устойчивым алгоритмом.

Замечая, что $E_n = \int_0^1 x^n e^{x-1} dx \le \int_0^1 x^n dx = \frac{x^{n+1}}{n+1}\Big|_0^1 = \frac{1}{n+1}$.

Итак, $E_n \to 0$ при $n \to \infty$.Например, если мы аппроксимируем нулем число E_{20} и возьмем этот ноль стартовым значением, то сделаем начальную ошибку не превосходящую 1/21.Ко времени вычисления E_{15} начальная ошибка уменьшится до величины , меньшей 4×10^{-8},что меньше единственной ошибки округления. Когда мы дойдем до E_{15}, начальная ошибка в E_{20} будет совершенно подавлена устойчивостью алгоритма.

2. Решение квадратных уравнений

Рассмотрим, как работают формулы для нахождения корней квадратного уравнения через формулу дискриминанта.

Возьмем плавающую систему с β=10, t=8, L= - 50, она намного точнее многих широко используемых систем.

Случай1. $a = 1, b = -10^3, c = 1$.

Точные корни соответствующего квадратного уравнения, правильно округленные до 11 значащих десятичных цифр:

$$x_1 = 99999.999990$$
$$x_2 = 0.000010000000001.$$

Если мы воспользуемся формулами теоремы, то вычислим

$x_1 = 100000.00$ (очень хорошо)

$x_2 = 0$ (совершенно неверно).

При вычислении x_2 мы пострадали от катастрофической потери верных знаков. Существуют различные альтернативные способы вычисления корней квадратного полинома. Одним из них является способ использования знака при коэффициенте b для определения какая формула несет меньшую потерю знака, и затем вычисление одного из корней по этой формуле.

Таким образом, $x_1 = -\dfrac{b + sign(b)\sqrt{b^2 - 4ac}}{2a}$.

Поскольку из равенства $ax^2 + bx + c = a\left(x - x_1\right)\left(x - x_2\right)$ следует, $ax_1x_2 = c$, то второй корень можно вычислить по формуле $x_2 = \dfrac{c}{ax_1}$.

Для данных случая1 такой способ дает $x_1 = 100000.00$, $x_2 = 1.0000000 / 100000.00 = 0.000010000000$; оба результата вполне приемлемы.

Рассмотрим мощный критерий для реализации машинного алгоритма решения квадратных уравнений. Мы скажем, что комплексное число z находится строго в пределах множества F, если либо z=0 , либо $\beta^{L+2} \le \operatorname{Re} z \le \beta^{U-2}$ и $\beta^{L+2} \le \operatorname{Im} z \le \beta^{U-2}$.Это означает, что мнимая и действительная части этого числа находятся строго внутри области чисел, которые могут быть хорошо приближены элементами F.

Предположим, что числа a,b,c находятся строго в пределах F.Тогда они должны быть допустимы в качестве входных данных для алгоритма решения квадратных уравнений. Если $a=b=c=0$,то алгоритм должен заканчивать свою работу сообщением, что все комплексные числа удовлетворяют квадратному уравнению $ax^2+bx+c=0$.Если $a=b=0, c\neq 0$, то на выходе алгоритма должна быть информация, что ни одно комплексное число не удовлетворяет этому уравнению.

В противном случае пусть z_1, z_2 - точные корни уравнения, занумерованные так, что $|z_1|\leq|z_2|$.Во всех случаях, когда z_1 находится строго в пределах F, алгоритм должен находить хорошее приближение к z_1 (приближение должно отличаться не более, чем на одну единицу предпоследнего знака корня).То же самое должно выполняться и для z_2.

Если один или два корня не находятся строго в пределах F, то должно выдаваться соответствующее сообщение и тот корень, который лежит строго в пределах, должен быть определен с указанной точностью.

Рассмотрим типичные примеры.

Случай2. $a=6, b=5, c=-4$.

Здесь нет никаких сложностей при вычислении

$$x_1 = 0.50000000$$
$$x_2 = -1.3333333$$

или приближений к этим значениям, какой бы формулой ни пользоваться.

Случай3. $a=6\times10^{30}, b=5\times10^{30}, c=-4\times10^{30}$.

Поскольку, с точностью до множителя 10^{30},это те же коэффициенты, что и в случае 2, то корни не изменились. Однако применение формул для x_1, x_2 вызовет переполнение. Это скорее всего будет обнаружено до входа в алгоритм, и все три числа будут поделены на масштабирующий множитель типа 10^{30}.и все сведется к случаю 2.

Случай4. $a=10^{-30}, b=-10^{30}, c=10^{30}$.

Здесь x_1 примерно равен 1, в то время как x_2 примерно равен 10^{60}.Значит наш алгоритм должен находить x_1 находится очень точно, несмотря на то, что x_2 за пределами F.

Случай5. $a = 1.0000000, b = -4.0000000, c = 3.9999999$.

Здесь корнями уравнения являются $x_1 = 1.999683772, x_2 = 2.000316228$.Однако применение квадратных формул дает $x_1 = x_2 = 2.0000000$, это приближение не удовлетворяет выдвинутому выше критерию. Это случай чувствительного уравнения. Однако же чувствительные уравнения все же могут решаться так точно, чтобы удовлетворять нашему критерию, если часть вычислений проводить с повышенной точностью. Для нашего примера это $t = 16$. Этот способ учета ошибок называется обратным анализом. $t = 16$. В определенном смысле корни вычисленные в случае 5($x_1 = x_2 = 2.0000000$) являются хорошими решениями данного уравнения. Этот способ учета ошибок называется обратным анализом. Такой способ характеризуется следующим подходом: насколько малые изменения в исходных данных задачи необходимы, чтобы представить вычисленные результаты как точное решение представленной задачи. Тогда как в прямом анализе выясняется просто насколько ошибочны полученные результаты в качестве решения задачи с ее собственными исходными данными.

Упражнение

Напишите программу для решения квадратного уравнения $ax^2 + bx + c = 0$,
где a, b, c -числа с плавающей точкой.

1) Нужно считаться с возможным переполнением и машинными нулями.
2) Удовлетворительную точность вычисления можно получить с точностью превосходящей точность рабочей арифметики.