Численные методы

Кудряшова Татьяна Юрьевна (а. 247, 2 уч. корп)

2018 год, весенний семестр

1. Математические вычисления

1. Аналитические преобразования (символьные вычисления) 2. Численные расчеты - численные методы решения математических задач в численном виде.

- Решение систем линейных уравнений
- Решение систем нелинейных уравнений (включает, например, $3x - e^x = 0$)
-

2. Способы представления чисел в памяти вычислительных устройств

- Числа с фиксированной запятой (точкой) (стандарт $ISO/IEC\ TR$ 18037)
- Числа с плавающей запятой (точкой) (стандарт $IEEE$ 754)

Числа с фиксированной запятой

Используются, когда нужна минимальная поддержка дробных чисел на целочисленном процессоре. Плюсы:

- ускорение вычислений, когда не нужна большая точность
- устойчивость к ошибкам округления
- меньшая стоимость

Минусы:

- малый диапазон вещественных чисел

10. $1 * 10^2 + 0 * 10^1 + 3 * 10^0 + 6 * 10^{-1} + 7 * 10^{-2}$
2. 101,011 $= 1 * 2^2 + 0 * 2^1 + 1 * 2^0 + 0 * 2^{-1} + 1 * 2^{-2} + 1 * 2^{-3}$
r. $a = + - a_n r^n a_{n-1} r^{n-1} + ... a_0 r^0 + a_{-1} r^{-1} ...$, a_k - разряд, r - основание системы

x - число (103,67)
x' - целочисленное представление (10367)
z - вес младшего разряда (-2). Обычно фиксируется и не хранится.

Двоично-десятичный код Каждый десятичный разряд записывается в виде 4-битного двоичного представления. Пример: 310_{10} = 0011 0001 0000

Есть запрещенные состояния, которые невозможно перевести обратно в 10-систему.

Числа с плавающей запятой

Плюсы:

- можно использоватьбольший диапалон значений при той же относительной точности
- меньшая погрешность численного метода

Минусы:

- возможны большие ошибки округления
- вычислительные устройства существенно дороже

Мантисса (M) - целое число фиксированной длины, оперделяющее старшие разряды действительного числа. Всегда начинается с 1 (в двоичной системе счисления), причем эта 1 не пишется. (23 разряда в float). На знак выделяется 1 бит (0 - положительное).

Порядок (E) - степень базы (двойки) старшего разряда. (8 разрядов в float)

Пример: 237,342 = $\mathit{2{,}37342} * 10^2$

$(-1) * 1.M * 2^E$

$-12{,}345 = -1{,}2345 * 10^1$ порядок 1

$-12{,}345 = -1{,}2345 * 10^{-3}$ - порядок 124 (124-127=-3)

Сдвиг - возможность записывать отрицательные числа в порядок.

Пример: тип float

Знак $s = 0$ - положительное число

Порядок $E = 01111100_2 - 127_{10} = -3$

Мантисса $M = 1.01$ (первая единица неявная)

Число $F = 1.01 * E^{-3} = 0{,}00101 = 2^{-3} + 2^{-5} = 0{,}15625$

Округление в стандарте IEEE 754

Чтобы избежать накопления ошибок округления, в ситуациях типа $N.5$ округление идет до ближайшего четного числа. $4.5 \approx 4, 3.5 \approx 4$

В стандарте предусмотрены решения проблем появления большой ошибки, которые возникают при вычитании близких чисел: $x^2 - y^2$ автоматически заменяется, например, на $(x-y)(x+y)$.

Представимый диапазон вещественных чисел

M_0 - минимальное представимое число. В Matlab $2_{-1074} = 4.94...e-324, 2^{-1075} = 0$. В Mathcad $M_0 = 2^{-49}$

M_∞ - максимальное представимое число. В Matlab $2_{1023} = 8.98...E307, 2^{1024} = Inf$. В Mathcad также $M_\infty = 2^1023$

Машинный эпсилон ϵ_m - минимальное положительное число, которое при сложении с 1 дает результат, отличный от 1: если $\epsilon < \epsilon_m, 1 + \epsilon = 1$. В Matlab $1 + 2^{-52} = 1.000..., 1 + 2^{-53} = 1$.

Денормализованные числа (subnormal)

Пусть имеем нормализованное представление с длиной мантиссы |M|=2 бита (плюс 1 бит нормализаци), диапазон значений порядка -1<=E<=2. Можно записать 16 чисел. При этом около 0 будет "дырка около 0 значений будет мало.

Что делать? Для обычных больших чисел можно использовать нормализованный формат (мантисса начинается с 1), для маленьких - денормализованный формат (мантисса начинается с 0). При этом Matlab работает с денормализованными числами, а Mathcad их не воспринимает (см. различия в M_0).

Ссылка: habrahabr.ru/post/112953/

3. Влияние порядка выполнения действий

1. Умножение
   Нужно следить, чтобы результаты не выходили за $[M_0, M_\infty], M_0 = 2^{-u}, M_0 = 2^u$
   $x_1 = 2^{u/2}, x_2 = 2^{u/8}, x_3 = 2^{3u/4}, x_4 = 2^{-u/2}, x_5 = 2^{-3u/4}$

   $x_1x_2x_3 = 2^{11u/8} > M_\infty$
   $x_4x_5 = 2^{-5u/4} < M_0$

$x_1x_2x_3 = 2^{11u/8} > M_\infty$

2. Сложение
   При сложении больших чисел с малыми точность может теряться: $10^20 + 1 - 10^20 = 0$

   Поэтому сначала складываем малые значения, потом прибавляем большие. Если требуется сложить числа в массиве, его нужно сначала отсортировать.

4. Накопление погрешности округления

t - число разрядов мантиссы
a - точное число
$\tilde{a}$ - представимое число
Тогда $\frac{|a-\tilde{a}|}{|a|} \leqslant 2^{-t}$ - происходит округление.
$fl(a \bigoplus b) = (a \bigoplus b)(1+\epsilon), \epsilon \leqslant 2^{-t}$ - ошибка действия
Пример:
$z = a + b + c$

$z_1 = fl(a+b)(1+\epsilon_1)$

$\tilde{z} = fl(z_1 + c)(1+\epsilon_2) = ((y_1+y_2)(1+\epsilon_1)+y_3)(1+\epsilon_2) = (y_1+y_2)(1+\epsilon_1)(1+\epsilon_2)+y_3(1+\epsilon_2)$

При этом $\tilde{y_1} = y_1(1+\epsilon_1)(1+\epsilon_2), y_2 = y_2(1+\epsilon_1)(1+\epsilon_2), y_3 = y_3(1+\epsilon_2)$

То есть $(y_1+y_2)+y_3 \neq y_1+(y_2+y_3)$
Выводы

1. Погрешности вычислений могут накапливаться
2. Математические законы могут нарушаться
3. Нельзя рассчитывать на точное равенство