

**ОПЕРАЦИЯ ДЕЛЕНИЯ ДЛЯ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ.
Ч. 1**

*МП «ПАК» Института кибернетики имени В.М. Глушкова АН УССР, Киев, Украина

Анотація. Пропонується цифровий метод виконання арифметичної операції ділення (ОД). Алгоритм забезпечує розпаралелювання обчислювального процесу, прискорення його і підвищення точності. Він заснований на виконанні етапів операцій в алгебрі матриць і реалізації методів цифрової арифметики в булевій (можливо, у багатозначній) алгебрі. Розглянуто варіанти виконання операції та приклади рішення. Передбачається розвиток аналогічних робіт у плані створення і використання комбінованих алгоритмів обробки даних шляхом застосування інших алгебр у поєднанні з методами цифрової арифметики.

Ключові слова: цифровий метод, операція ділення, паралельні обчислення, обчислювальні системи.

Аннотация. Предлагается цифровой метод выполнения арифметической операции деления (ОД). Алгоритм обеспечивает распаралеливание вычислительного процесса, ускорение его и повышение точности ОД. Он основан на выполнении этапов операций в алгебре матриц и реализации методов цифровой арифметики в булевой (возможно, многозначной) алгебре. Рассмотрены варианты выполнения ОД и примеры решения. Предполагается развитие аналогичных работ в плане создания и использования комбинированных алгоритмов обработки данных путём применения других алгебр в сочетании с методами цифровой арифметики.

Ключевые слова: цифровой метод, операция деления, параллельные вычисления, вычислительные системы.

Abstract. It is proposed the digital method of performing arithmetic division operations (DO). Algorithm provides the parallelizing of computing process, its acceleration and accuracy increase. It is based on carrying out the steps of operations in the matrix algebra and realization of digital arithmetic in Boolean (possibly multi-valued) algebra. Variants of the execution of operation and examples of solutions are regarded. The development of similar projects for the creation and use of combined data processing algorithms by applying other algebras in combination with the methods of digital arithmetic is suggested.

Keywords: digital method, division operation, parallel computing, computing systems.

1. Введение

Создание гетерогенных вычислительных систем (ГВС) высокой производительности, параллельных структур (ПС), особенно для обработки информации, представленной в виде сложных структур данных (ССД) для реализации вычислительных алгоритмов в едином технологическом (вычислительном) потоке (ЕТП) с помощью процессорных элементов (ПЭ) со скалярным умножителем (СУ), в своём составе требует нового подхода к численным и арифметическим методам, закладываемым в структуры спецпроцессора (СП) и всей вычислительной системы (ВС) [1–3].

Один из подходов в трансформации численных методов для реализации алгоритмов решения задач математической физики (МФ) с помощью метода конечных элементов (МКЭ) предложен и описан в [1, 4–7]. Такой подход предполагает обработку ССД и решение систем линейных алгебраических уравнений (СЛАУ) в режиме ЕТП. Он обеспечивает переход к построению высокопроизводительных архитектур гетерогенного типа на базе ПЭ с СУ в своём составе с HOST процессором в виде центральной интеллектуальной машины (ЦИМ).

При построении СУ в основу арифметических операций, которые он должен реализовывать, закладывают операции сложения, вычитания, умножения, деления. Это основа всех одно- и многоядерных процессоров, работающих в составе кластерных структур (КС).

Операция деления (ОД), которая часто является неотъемлемой частью алгоритма вычислительного эксперимента (ВЭ), записанного в аналитическом виде, является частью набора операций, которые впоследствии необходимо реализовывать с помощью процессора ЭВМ. Но ОД в цифровом виде является менее точной и выполняется за большее время (по отношению ко времени выполнения операции умножения). Поэтому её реализацию в АУ процессора стараются избегать, а на уровне алгоритма её выполнение в ЭВМ часто реализуют при помощи стандартных подпрограмм, основанных на операциях сложения, вычитания и умножения (или с помощью спецпроцессора, который реализует ОД). Для параллельных устройств ограничения связаны и с последовательным выполнением ОД, поскольку каждая новая цифра частного, как и новый остаток, не могут быть вычислены раньше, чем будет получен и проанализирован предыдущий остаток.

Статья состоит из двух частей. В первой части статьи излагается цель работы, суть предлагаемого метода замены ОД методом определения обратной величины (ОВ) кода числа с последующим умножением ОВ на код числа делимого. На конкретных примерах показывается возможность выполнения предлагаемой операции способом определения ОВ кода числа, опираясь на математический аппарат матричной алгебры.

Во второй части статьи показаны правомочность и целесообразность использования аппарата матричной алгебры на первом этапе решения задачи к возможности сочетания на втором этапе аппарата матричной алгебры с математикой числовой арифметики. Показаны варианты сочетания методов алгебр на разных этапах формирования обратного значения числа. Доказывается правомочность и даны условия использования числовой арифметики в сочетании с математическим аппаратом матричной алгебры. Предложены и описаны варианты вычисления ОВ кода числа. На конкретных примерах рассмотрены: возможность реализации поставленной задачи вообще и на параллельных ГВС, в частности; преимущества предлагаемого решения для замены ОД в классическом понимании операции деления.

Цель работы

Для решения задач вычислительной математики, к которым на практике сводится аппроксимация практически всех задач, а особенно для решения СЛАУ, следует разработать метод и алгоритм реализации ОД чисел или способ вычисления ОВ кода числа (1/С). В гетерогенных системах он должен заменять ОД и выполнять её аналог с помощью СУ, закладываемых в ПЭ, для реализации режима ЕТП. Метод должен позволить в параллельном режиме работы СП реализовать процедуру факторизации матрицы жесткости с помощью алгоритмов, предложенных и описанных в [2]. Взамен решения исходной матрицы с помощью выражения, свойственного методу Гаусса, связанного с делением на ведущий элемент строки (или столбца) и со сложностями метода (по выбору ведущего элемента, проверки его на ноль, перестановками строк (или столбцов) и проч.) предлагается перейти к реализации уравнения вида

$$\bar{a}_i \leftarrow \frac{a_{ij} * \bar{a}_i - a_{ij} * \bar{a}_j}{a_{j-1,j-1}}. \quad (1)$$

Запись вида

$$\bar{a}_i \leftarrow \bar{a}_i * a_{ij} - a_{ij} * \bar{a}_j \quad (2)$$

позволяет известную формулу факторизации по Гауссу для матрицы A_m с коэффициентами a_{ij} , $i, j = \overline{1, n}$:

$$\bar{a}_i \leftarrow \bar{a}_i - \frac{a_{ij}}{a_{jj}} * \bar{a}_j \quad (3)$$

заменить алгоритмом реализации выражения:

$$\bar{a}_i \leftarrow (a_{ij} * \bar{a}_i - a_{ij} * \bar{a}_j) * a_{j-1,j-1}^{-1} \quad (4)$$

Замена в выражении (4) ОД операцией умножения на ОВ $a_{j-1,j-1}^{-1}$:

$$L_{j-1,j-1} = 1/a_{j-1,j-1} = a_{j-1,j-1}^{-1} \quad (5)$$

от значения диагонального коэффициента предыдущей $a_{j-1,j-1}$, ранее вычисленной строки (при условии, что $a_{11} \equiv 1$), позволяет исключать традиционное выполнение ОД из алгоритма решения СЛАУ по ЕТП.

Для решения итерационным методом Ньютона-Рафсона нелинейных уравнений вида

$$x_{r+1} = x_r - f(x_r)/f'(x_r), \quad (6)$$

где $f(x_r)$ – значение функции $f(x_r)$ в точке $(x_r, f(x_r))$;

$f'(x_r)$ – значение производной в точке $(x_r, f(x_r))$.

В работе [8] предложен и рассмотрен метод решения уравнения (6). Согласно методу, исходное уравнение (6) предлагается записывать в виде РПМ и операцию деления функции $f(x_r)$ на производную $f'(x_r)$ от неё заменять умножением функции на обратное значение производной от нее. Для этого после вычисления $[\Phi_r']^{-1}$ необходимо выполнять операции $\delta_r = [\Phi_r] * [\Phi_r']^{-1}$ с помощью функций, записанных в виде РМП.

Постановка задачи

Разработать метод и алгоритм реализации ОД на структуре типа СУ. С этой целью операцию деления числа B на число C с получением частного от деления A

$$B : C = A \quad (7)$$

предлагается выполнять так:

$$\{B\} * \{C^{-1}\} = \{A\}. \quad (7a)$$

Для решения (7a) следует вычислить обратную от делителя $[C]$ матрицу $[C^{-1}]$ и матрицу делимого $[B]$ умножить на обратную от делителя матрицу $[C^{-1}]$, получив в результате ОВ частного $[A]$ от деления (7).

Решение задачи

Известно, что последовательность

$$C = \{c_0, c_1, c_2, \dots, c_n\} \quad (8)$$

в двоичной, например, позиционной системе счисления определяет число C :

$$C = c_0 2^0 + c_1 2^{-1} + c_2 2^{-2} + \dots + c_i 2^{-i} + \dots c_n 2^{-n}, \quad (9)$$

а полином $p(x)$ вида

$$P(x) = a_0x_0 + a_1x^1 + a_2x^2 + \dots + a_nx^n \quad (10)$$

в виде аппроксимирующего ряда

$$f(x) \approx \frac{c_1}{x} + \frac{c_2}{x_2} + \frac{c_3}{x_3} + \dots, \quad x \rightarrow \infty \quad (11)$$

можно записать, положив $x^{-1} = 2^{-1}, x^{-2} = 2^{-2}, x^{-3} = 2^{-3}, \dots$

Тогда рассмотрим двоичное представление числа в позиционной системе в виде полиномиального кода

$$C \approx \{c_02^0 \ c_12^{-1} \ c_i2^{-i} \ \dots \ c_n2^{-n}\}, \quad n \rightarrow \infty. \quad (12)$$

Запишем его с помощью РМП в матричном виде и будем рассматривать выполнение алгебраических операций по алгоритму факторизации матрицы над соответствующими полиномами, используя методы алгебры матриц. При этом число C (12) можно записать [1, 2] в виде

$$f(C) \rightarrow [C] \rightarrow \begin{bmatrix} c_0 & c_1 & c_2 & \dots & c_n \\ & c_0 & c_1 & \dots & c_{n-1} \\ & & c_0 & \dots & c_{n-2} \\ & & & \dots & \dots \\ & & & & c_0 \end{bmatrix}. \quad (13)$$

Для матрицы $[C]$ будем рассматривать $[C]^T$, транспонированную к матрице $[C]$:

$$f^T(C) \rightarrow [C]^T = \begin{bmatrix} c_0 & & & & \\ c_1 & c_0 & & & \\ c_2 & c_1 & c_0 & & \\ \dots & \dots & \dots & \dots & \\ c_n & c_{n-1} & c_{n-2} & \dots & c_0 \end{bmatrix}, \quad (14)$$

а также вектор $[C^*]$:

$$f^*(C) \rightarrow [C^*] = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \dots \\ c_n \end{bmatrix} = [c_0 \ c_1 \ c_2 \ \dots \ c_n]^T = [f^*(C)], \quad (15)$$

как это предложено в [1] и рассмотрено в [4–7] для функций с записью полиномов в виде РМП.

В (13–15) число, аналогично полиному, записано с помощью РМП по принципу циклической записи битов (разрядов) с учетом их веса в виде верхней треугольной матрицы. В записи на месте c_{11} (на главной диагонали) матрицы $[C]$ всегда стоит старший разряд $c_1 * 2^{-1}$ мантииссы числа, записанного в устройстве в виде ПЗ, а на месте c_{1n} – младший разряд числа $c_{1n} * 2^{-n}$. На главной c_{ii} диагонали матрицы $[C]$ всегда стоят «1». Это следует из того, что $c_{ii}=1, i = \overline{1, n}$ – старшая значащая цифра нормализованной мантииссы чис-

ла весом $2^{-1}c_1 = 1/2 * 1 = 1/2$. Далее, в 1-й строке матрицы $[C]$ (в сторону убывания), стоят разряды с меньшим весом ($2^{-2}c_2 \ 2^{-3}c_3 \ \dots \ 2^{-n}c_n$), то есть следующие биты числа с учетом их веса в пределах разрядной сетки.

На практике это означает, что вычисление обратной матрицы $[C]^{-1}$ по правилам алгебры матриц методом факторизации по Гауссу сводится к исключению прямого хода, на котором формируется верхне треугольная матрица. При этом заранее определено, что размерность m_c матрицы $[C]$ (делителя), матрицы $[B]$ (делимого) m_B , а также вектора столбца $[C^*]$ совпадает. Кроме того, поскольку мантиссы чисел, представленных в прямом коде в форме ПЗ, равны «1», а их вес равен $1/2$, то произведение $b_{ii} * c_{ii}^{-1} = 1/2 * 1/2 = 1/4$ даёт смещение в пределах разрядной сетки на единицу в сторону младшего разряда и в принятом формате чисел гарантирует от переполнения.

2. Получение обратной $[C]^{-1}$ матрицы от числа в РМП

Пусть матрица $[C]$ представляет число

$$C = \{.110\} = \{c_1 2^{-1} + c_2 2^{-2} + c_3 2^{-3}\} = 1 * 1/2 + 1 * 1/4 + 0 * 1/8 = 6/8 \quad (16)$$

с записью в форме РМП:

$$[C] = \begin{bmatrix} 1 & 1 & 0 \\ & 1 & 1 \\ & & 1 \end{bmatrix}. \quad (17)$$

Вычислим обратную $[C]^{-1}$ матрицу традиционным способом:

$$[C]^{-1} = \begin{array}{ccc|ccc} c_{1j} & c_{2j} & c_{3j} & e_{j1} & e_{j2} & e_{j3} \\ \hline \left[\begin{array}{ccc|ccc} 1 & 1 & 0 & 1 & 0 & 0 \\ & 1 & 1 & 0 & 1 & 0 \\ & & 1 & 0 & 0 & 1 \end{array} \right] & \rightarrow & \left[\begin{array}{ccc|ccc} 1 & -1 & 1 & 1 & 0 & 0 \\ & 1 & -1 & 0 & 1 & 0 \\ & & 1 & 0 & 0 & 1 \end{array} \right] \\ \left[\begin{array}{ccc|ccc} & 1 & & 0 & 0 & 1 \\ & 1 & & 0 & 1 & -1 \\ 1 & & & 1 & -1 & 1 \end{array} \right] \end{array} \quad (18)$$

Полученная матрица $[C]^{-1}$ записана в виде РМП, также верхне треугольная имеет целочисленные значения, но отдельные биты числа C (коэффициенты c_{ij}) имеют отрицательные значения, некоторые из них отличны от ± 1 и являются многозначными числами.

Естественно, что произведение матрицы $[C]$ на обратную $[C]^{-1}$ даёт единичную $[E]$ матрицу.

Появление отрицательных значений битов (коэффициентов матрицы) в $[C]^{-1}$ требует изменения в традиционном двухпозиционном представлении битов "0" и "1" и идентификации их как разрядов числа, а не как коэффициентов полинома.

Далее при рассмотрении способов получения ОВ для упрощения изложения предлагаемых решений используется десятичная запись отдельных разрядов ОВ при двухпозиционном представлении числа в целом.

Рассмотрим выполнение ОД традиционным методом и варианты получения ОВ кода числа предлагаемыми способами.

3. Примеры

Выполнение ОД традиционным (⊗) методом «углом»

Разделим число $B = 15/32 = .01111$ (делимое) на число $C = 3/4 = 24/32 = .11000$ (делитель), получая в результате A (частное):

$$A = 5/8 = 20/32 = .01010 \quad (19)$$

(частное) [9, 10].

Деление 1-го остатка (B) $\begin{array}{r} .01111 \\ .11110 \quad | \underline{.1100} \\ \underline{.11000} \quad 01010 \quad (A) \\ \underline{.00110} \quad \text{второй остаток} \\ \underline{.01100} \quad \text{удвоение 2-го остатка} \\ \underline{.11000} \quad \text{удвоение 3-го остатка} \\ - \underline{.11000} \quad \text{4-й остаток} \\ \underline{.00000} \quad \text{5-й остаток} \\ \underline{.00000} \quad \text{удвоение 5-го остатка} \\ - \underline{.00000} \quad \text{6-й остаток.} \end{array}$

4. Выполнение операции (вычисление ОВ кода числа) предлагаемым способом

Рассмотрим умножение делимого (B) на обратную $[C]^{-1}$ матрицу от числа [C].

Представим делимое B с помощью РМП в виде матрицы $[B_1]$, в которой число $B = \{b_0 2^0 \quad b_1 2^1 \quad b_2 2^2 \quad b_3 2^3 \dots\}$ сдвинуто влево на один разряд в сторону старшего (B) разряда (нормализовано):

$$B_1 \rightarrow [B_1] = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ & 1 & 1 & 1 & \\ & & 1 & 1 & \\ & & & 1 & 1 \\ & & & & 1 \end{bmatrix}, \quad (20)$$

а делитель C с помощью РМП представим в виде матрицы

$$C \rightarrow [C] = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ & 1 & 1 & 0 & 0 \\ & & 1 & 1 & 0 \\ & & & 1 & 1 \\ & & & & 1 \end{bmatrix} \quad (20a)$$

и записываем его без сдвига (поскольку C представлено в нормализованном виде).

Произведем вычисление обратной $[C]^{-1}$ от матрицы $[C]$ (с присоединенной к ней единичной матрицей $[E]$ той же размерности):

$$[C]^{-1} = \left[\begin{array}{ccccc|ccccc} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ & & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ & & & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ & & & & 1 & 0 & 0 & 0 & 0 & 1 \end{array} \right] \rightarrow \left[\begin{array}{ccccc} 1 & -1 & 1 & -1 & 1 \\ 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 \end{array} \right]$$

$$\left[\begin{array}{ccccc|ccccc} & & & & 1 & 0 & 0 & 0 & 0 & 1 \\ & & & & & 1 & 0 & 0 & 0 & -1 \\ & & & & & & 1 & 0 & 0 & 1 \\ & & & & & & & 1 & -1 & 1 \\ & & & & & & & & 1 & -1 \\ & & & & & & & & & 1 \end{array} \right]. \quad (21)$$

Легко проверить, что произведение матриц $[C] * [C]^{-1}$ дает единичную матрицу $[E]$:

$$[C] * [C]^{-1} = \left[\begin{array}{ccccc} 1 & 1 & 0 & 0 & 0 \\ & 1 & 1 & 0 & 0 \\ & & 1 & 1 & 0 \\ & & & 1 & 1 \\ & & & & 1 \end{array} \right] * \left[\begin{array}{ccccc} 1 & -1 & 1 & -1 & 1 \\ 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 \end{array} \right] = \left[\begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{array} \right] = [E]. \quad (22)$$

Выполним ОБ путем умножения матрицы $[B]$ ($B=0 \ 1 \ 1 \ 1 \ 1$) \rightarrow ($B_1=1 \ 1 \ 1 \ 1 \ 0$) $[B_1]$ (делимое) на обратную матрицу $[C]^{-1}$ (делитель):

$$[B_1] * [C]^{-1} = \left[\begin{array}{ccccc} 1 & 1 & 1 & 1 & 0 \\ & 1 & 1 & 1 & 1 \\ & & 1 & 1 & 1 \\ & & & 1 & 1 \\ & & & & 1 \end{array} \right] * \left[\begin{array}{ccccc} 1 & -1 & 1 & -1 & 1 \\ 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 \end{array} \right] = \left[\begin{array}{ccccc} 1 & 0 & 1 & 0 & 0 \\ & 1 & 0 & 1 & 0 \\ & & 1 & 0 & 1 \\ & & & 1 & 0 \\ & & & & 1 \end{array} \right], \quad (23)$$

Произведение даёт результат, равный числу $A * 2 = .10100$ в двоичной записи (учитывая сдвиг в сторону старшего на один разряд (20)) при записи числа B в матричном виде $[B_1]$, то есть тождественный результат. Деление B на C в матричном виде даёт (аналогично с (19)) частное A :

$$B_1 * C^{-1} \rightarrow [B_1]^T * [C^*]^{-1} = \left[\begin{array}{ccccc} 1 & & & & \\ 1 & 1 & & & \\ 1 & 1 & 1 & & \\ 1 & 1 & 1 & 1 & \\ 0 & 1 & 1 & 1 & 1 \end{array} \right] * \left[\begin{array}{c} 1 \\ -1 \\ 1 \\ -1 \\ 1 \end{array} \right] =$$

$$= \begin{bmatrix} 1*1=1 \\ 1*1+1*(-1)=0 \\ 0*1+1*(-1)+1*1=0 \\ 1*1+0*(-1)+1*1+1*(-2)=0 \\ 0*1+1*(-1)+0*1+1*(-2)+1*3=0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} x^{-1} \\ x^{-2} \\ x^{-3} \\ x^{-4} \\ x^{-5} \end{bmatrix} .=[.1 \ 0 \ 1 \ 0 \ 0]=\{A\} \rightarrow A_1, \quad (24)$$

которое после денормализации (числа A_1) будет $A_1 \rightarrow A=\{0 \ 1 \ 0 \ 1 \ 0\}$.

Полученные таким образом результаты равны

$$A \rightarrow [B] * [C]^{-1} = 01010 = 5/8 = B / C. \quad (25)$$

5. Варианты вычисления ОБ

Рассмотрение процедуры вычисления ОБ (21) по типу обратного хода метода факторизации матрицы показывает дополнительные (в вычислительном плане) возможности. Матрица $[C]^{-1}$ в форме РМП может формироваться из одного n -го (правого крайнего) столбца единичной матрицы, с помощью которого она вычисляется. Покажем это на примере.

Для числа

$$C = [1 \ 1 \ 0 \ 1 \ 0] \quad (26)$$

выполним обращение матрицы $[C]$ путем введения вектор-столбца E_{ij} , $i = \overline{n,1}$, $j = n$ для матрицы $E_{n,n}$, присоединенной к матрице $[C_{n,n}]$, с последующим вычислением значений c_{ii} , $i = \overline{n,1}$ в виде j -х вектор столбцов и с коррекцией значений $c_{i,j}$, $j = n, \overline{i, i-1, \dots, 1}$.

Запишем первый столбец (правую часть для вычисляемой обратной матрицы):

$$[C]_1^{-1} = \left[\begin{array}{cccc|cccc} 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ & & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ & & & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ & & & & 1 & 0 & 0 & 0 & 0 & 1 \end{array} \right] \rightarrow \left[\begin{array}{cccc|c} 1 & 1 & 0 & 1 & 0 \\ & 1 & 1 & 0 & 1 \\ & & 1 & 1 & 0 \\ & & & 1 & 1 \\ & & & & 1 \end{array} \right] \rightarrow [C]_2^{-1}. \quad (26a)$$

Вычислим пятый столбец матрицы $[C]$ и запишем его в качестве правой части обратной матрицы:

$$c_1 \ c_{i,2} \ \dots \ c_{i,n-1} \ c_{i,n} * e_n \ (e_{i,n}^{(j_1)} - c_{i,n} * e_n^{(j_1)}) = e_{i,n-1}^{(j_2)}, \ c_{n,n} = e_n$$

$$[C]_2^{-1} = \left[\begin{array}{cccc|c} 1 & 1 & 0 & 1 & [(0*1)=0] \Rightarrow 0 \\ & 1 & 1 & 0 & [(1*1)=1] \Rightarrow -1 \\ & & 1 & 1 & [(0*1)=0] \Rightarrow 0 \\ & & & 1 & [(1*1)=1] \Rightarrow -1 \\ & & & & 1 \Rightarrow 1 \end{array} \right] = \left[\begin{array}{cccc|c} 1 & 1 & 0 & 1 & \Rightarrow 0 \\ & 1 & 1 & 0 & \Rightarrow -1 \\ & & 1 & 1 & \Rightarrow 0 \\ & & & 1 & \Rightarrow -1 \\ & & & & \Rightarrow 1 \end{array} \right] \rightarrow [C]_3^{-1}. \quad (26b)$$

Вычислим четвертый столбец матрицы $[C]$ и запишем его в качестве правой части обратной матрицы:

$$[C]_3^{-1} = \left[\begin{array}{ccc|ccc} 1 & 1 & 0 & [(1*(-1))=-1] & |(0-(-1))=1 \Rightarrow & 1 \\ & 1 & 1 & [(0*(-1))=0] & |(-1)-0=-1 \Rightarrow & -1 \\ & & 1 & [(1*(-1))=-1] & |(0-(-1))=1 \Rightarrow & 1 \\ & & & & | \Rightarrow & -1 \\ & & & & | \Rightarrow & 1 \end{array} \right] \rightarrow [C]_4^{-1}. \quad (26\text{в})$$

Вычислим третий столбец матрицы $[C]$ и запишем его в качестве правой части обратной матрицы:

$$[C]_4^{-1} = \left[\begin{array}{ccc|ccc} 1 & 1 & (0*1)=0 & |(1-0)=1 \Rightarrow & 1 \\ & 1 & (1*1)=1 & |(-1-1)=-2 \Rightarrow & -2 \\ & & & & 1 \\ & & & & -1 \\ & & & & 1 \end{array} \right] \rightarrow [C]_5^{-1}. \quad (26\text{г})$$

Вычислим второй столбец матрицы $[C]$ и запишем его в качестве правой части обратной матрицы:

$$[C]_5^{-1} = \left[\begin{array}{ccc|ccc} 1 & 1 & 1*(-2)=-2 & |1-(-2)=3 \Rightarrow & 3 \\ & & & & -2 \\ & & & & 1 \\ & & & & -1 \\ & & & & 1 \end{array} \right]. \quad (25\text{д})$$

В итоге получим вектор-столбец, который является первой строкой вычисляемой обратной матрицы $[C]^{-1}$. В ней, по аналогии с записью полиномов в виде РМП, на месте e_{1n} – стоит младший разряд ($c_n * 2^{-n}$) обратного числа, а на месте e_{11} – старший разряд ($c_1 * 2^{-1}$) обратного числа:

$$[C]^{-1} = \left[\begin{array}{ccccc|cc} 1 & 1 & 0 & 1 & 0 & 0 \\ & 1 & 1 & 0 & 1 & 0 \\ & & 1 & 1 & 0 & 0 \\ & & & 1 & 1 & 0 \\ & & & & 1 & 1 \end{array} \right] \rightarrow \left[\begin{array}{ccccc|cc} 1 & -1 & 1 & -2 & 3 \\ & 1 & -1 & 1 & -2 \\ & & 1 & -1 & 1 \\ & & & 1 & -1 \\ & & & & 1 \end{array} \right] = \left[\begin{array}{c|c} 1 \\ -1 \\ 1 \\ -2 \\ 3 \end{array} \right] \begin{array}{l} x^{-1} \\ x^{-2} \\ x^{-3} \\ x^{-4} \\ x^{-5} \end{array} = \quad (26\text{е})$$

$$= [1 \ -1 \ 1 \ -2 \ 3]^T * \{X^{-i}\} = [C^*]^T. \quad (26\text{ж})$$

Алгоритм формально можно записать так.

1. $c_{n,n} = e_n^{(1)} = 1, .$
2. Умножив $i - e$ ($i = \overline{n,2}$) элементы $(c_{i,n})$ ведущего столбца на $e_n^{(1)}$

$$\left[\begin{array}{c} c_{i,n} \end{array} \right] * e_n^{(1)} = \left[\begin{array}{c|c} (e_n^{(1)} - c_{i,n}) \\ \hline \end{array} \right], \quad i = \overline{n,1},$$

получим вычисленный ведущий столбец.

3. Из вектор-столбца $e_{i,n}$ (уменьшаемое) вычтем вектор ведущего столбца (вычитаемое):

$$\begin{bmatrix} e_i^{(j)} \\ | \\ | \\ | \end{bmatrix} - \begin{bmatrix} c_{i,n} \\ | \\ | \\ | \end{bmatrix} * e_n^{(j)} = \begin{bmatrix} (e_i - c_{i,n} * e_n) \\ | \\ | \\ | \end{bmatrix}.$$

4. Из уравнения следующей $(n-1)$ -й строки матрицы $[C]$, где выполняется равенство

$$\begin{bmatrix} | \\ | \\ c_{n-1,n} \\ | \end{bmatrix} * e_{n-1} = \begin{bmatrix} | \\ | \\ (e_{n-1} - c_{n,n} * e_n) \\ | \end{bmatrix},$$

определяем очередное $(n-1)$ -е значение разряда ОВ кода числа $[C]^{-1}$.

$$5. c_{n-1} = (e_{n-1} - c_{n,n} * e_n).$$

В отличие от алгоритма решения уравнения $Ax = b$ правую часть уравнения (п. 5) не делим на величину ($c_{n-1,n-1} = 1$) в силу отсутствия смысла в делении на 1.

6. Делаем $(n-1)$ -й столбец ведущим. Умножаем его элементы $c_{i,n-1}$ ($i = \overline{(n-2), 1}$) на значение e_{n-1} . Получим значение соответствующего разряда ведущего $(n-1)$ -го столбца матрицы $[C]$.

$$7. e_{n-1} * \begin{bmatrix} c_{i,n-1} \\ | \\ | \\ | \end{bmatrix} = \begin{bmatrix} (c_{i,n-1} * e_{n-1}) \\ | \\ | \\ | \end{bmatrix} \quad i = \overline{(n-1), 1}.$$

Запоминаем их значения взамен соответствующих предыдущего вектор-столбца.

8. Вычисляем вектор ОВ кода числа. Для i -й строки ($i = \overline{(n-2), 1}$) матрицы $[C]$ он равен

$$\begin{bmatrix} e_i^{(j)} \\ | \\ | \\ | \end{bmatrix} = \begin{bmatrix} (e_i^{(j)} - \sum_{j=i+1}^n c_{ij} * e_j) \\ | \\ | \\ | \end{bmatrix}, \quad j = \overline{(n-1), 2}, \quad i = \overline{(n-1), 1}.$$

9. Текущее i -е значение ОВ разряда кода числа $[C]^{-1}$ для i -й строки:

$$e_i^{(j)} = (e_i^{(j-1)} - \sum_{j=i+1}^n c_{ij} * e_j), \quad j = \overline{(n-1), 2}, \quad i = \overline{(n-2), 1},$$

причем в отличие от алгоритма решения системы $Ax = b$ по методу, предложенному и описанному в [2], правую часть уравнения по п. 9 нет смысла делить на $c_{ij} = 1$ (диагональные элементы матрицы для нормализованного числа в записи РМП всегда равны "1") и т.д. То есть автоматически срабатывает приведенная в [2] коррекция алгоритма:

$$L_{ii} = 1/c_{ii} = c_{ii}^{-1} = 1,$$

$$10. c_n = e_n * L_{ii} = 1,$$

$$c_i^{(j)} = e_i^{(j)}.$$

Алгоритм вычисления обратного значения $[C]^{-1}$, таким образом, заключается в следующем:

- а) умножить вектор-столбец c_i , $i = n, n-1, \dots, 1 = \overline{n, 1}$ на значение c_i , $i = \overline{n, 1}$, вычисленное на i -м шаге;
- б) положить $c_{n,n} = e_n = 1$;
- в) приравнять с обратным знаком значения c_i -го ($i = \overline{n-1, 1}$) столбца матрицы $[C]$ к правой части матрицы $[E] = [e_n]$, то есть $-[c_{i,n}] = [e_{i,n}]$ ($i = \overline{n-1, 1}$), что равносильно вычитанию из правой части матрицы вычисленного значения текущего столбца;
- в) вычислить значения $(i-1)$ -столбца по п. а) Алгоритма и так далее до c_{11} ;
- г) вычисленный на n -м шаге вектор-столбец после транспонирования его является обратным числом $\{C\}^{-1}$ от исходного $\{C\}$, каждый из коэффициентов вектора является целочисленным коэффициентом $c_i * 2^{-i}$, $i = \overline{1, n}$ числа $\{C\}$ (10), взятым со своим весом 2^{-i} .

Замечание

Следует учитывать, что число $[C^*]^T$ получено с помощью выражения (21–26) как полином вида (10–11) в соответствии с правилами алгебры матриц с РМП. Поэтому, если операции с ним выполнять, как с вектором, в соответствии с правилами алгебры матриц, то следует помнить правило, что операнды в алгебре матриц не коммутативны. Но если его рассматривать как число, то на него будет распространяться правило коммутативности операции умножения.

Рассмотрим некоторый набор структур матрицы $[C_{n,n}]$, который образуется при вычислении матрицы $[C]^{-1}$:

$$\begin{aligned}
 [C^*]^T = [10000] &= (16/32=0,5) \rightarrow [C]^{-1} = [10000] = (16/32=0,5), \\
 [C_1]^{-1} &= [1000000000] = (512/1024 = 0,5). \quad \delta = 0,000, \\
 [C^*]^T = [10001] &= (17/32=0,53125) \rightarrow [C]^{-1} = [1000-1] = (15/32=0,46875), \\
 [C_1]^{-1} &= [1000-100010] = (482/1024 = 0,470703125). \quad \delta = 0,001953125, \\
 [C^*]^T = [10010] &= (18/32=0,5625) \rightarrow [C]^{-1} = [100-10] = (14/32=0,4375), \\
 [C_1]^{-1} &= [100-100100-1] = (455/1024 = 0,444335937). \quad \delta = 0,006835937, \\
 [C^*]^T = [10011] &= (19/32=0,59375) \rightarrow [C]^{-1} = [100-1-1] = (13/32=0,40625), \\
 [C_1]^{-1} &= [100-110101-1] = (410/1024=0,400390625), \quad \delta = -0,005859375 \\
 [C^*]^T = [10100] &= (20/32=0,625) \rightarrow [C]^{-1} = [10-101] = (13/32=0,40625), \\
 [C_1]^{-1} &= [10-1010-1010] = (489/1024=0,400390625). \quad \delta = -0,005859375, \\
 [C^*]^T = [10101] &= (21/32=0,65625) \rightarrow [C]^{-1} = [10100] = (12/32=0,375), \\
 [C_1]^{-1} &= [10-100010-10] = (390/1024 = 0,380859375). \quad \delta = 0,005859375, \\
 [C^*]^T = [10110] &= (22/32=0,6875) \rightarrow [C]^{-1} = [10-111] = (11/32=0,34375), \\
 [C_1]^{-1} &= [10-1-1120-3-23] = (371/1024 = 0,368164062), \quad \delta = +0,024414062,
 \end{aligned}$$

$$\begin{aligned}
[C^*]^T &= [10111] = (23/32 = 0,71875) \rightarrow [C]^{-1} = [10-1-10] = (10/32 = 0,3125), \\
[C_1]^{-1} &= [10-1-1022-1-4-3] = (353/1024 = 0,344726562), \quad \delta = +0,032226562, \\
[C^*]^T &= [11000] = (24/32 = 0,75) \rightarrow [C]^{-1} = [1-11-11] = (11/32 = 0,34375), \\
[C_1]^{-1} &= [1-11-11-11-11-1] = (341/1024 = 0,333007812), \quad \delta = -0,010742187, \\
[C^*]^T &= [11001] = (5/32 = 0,15625) \rightarrow [C]^{-1} = [1-11-10] = (10/32 = 0,3125), \\
[C_1]^{-1} &= [1-11-10-10-110] = (302/1024 = 0,295000000), \quad \delta = -0,01171875, \\
[C^*]^T &= [11010] = (26/32 = 0,8125) \rightarrow [C]^{-1} = [1-11-23] = (11/32 = 0,34375), \\
[C_1]^{-1} &= [1-11-23-46-913-19] = (307/1024 = 0,299804687), \quad \delta = -0,043945312, \\
[C^*]^T &= [11011] = (27/32 = 0,84375) \rightarrow [C]^{-1} = [1-11-22] = (10/32 = 0,3125), \\
[C_1]^{-1} &= [1-11-22-23-33-4] = (334/1024 = 0,294921875), \quad \delta = -0,017578125, \\
[C^*]^T &= [11100] = (28/32 = 0,875) \rightarrow [C]^{-1} = [1-101-1] = (9/32 = 0,28125), \\
[C_1]^{-1} &= [1-10-101-101] = (293/1024 = 0,286132812), \quad \delta = +0,001882812, \\
[C^*]^T &= [11101] = (29/32 = 0,90625) \rightarrow [C]^{-1} = [1-101-2] = (8/32 = 0,25), \\
[C_1]^{-1} &= [1-101-220-35-4] = (282/1024 = 0,275390625), \quad \delta = +0,025390625, \\
[C^*]^T &= [11110] = (30/32 = 0,9375) \rightarrow [C]^{-1} = [1-1001] = (8/32 = 0,25), \\
[C_1]^{-1} &= [1-101-220-35-4] = (273/1024 = 0,266601562), \quad \delta = -0,014698437, \\
[C^*]^T &= [11111] = (31/32 = 0,96875) \rightarrow [C]^{-1} = [1-1000] = (8/32 = 0,25), \\
[C_1]^{-1} &= [1-101-220-35-4] = (264/1024 = 0,2578125). \quad \delta = +0,0078125. \quad (27)
\end{aligned}$$

Анализ структур $[C^*]^T$ и $[C]^{-1}$ показывает:

– значительную погрешность при вычислении обратного значения от числа, представленного только пятью разрядами, но точность возрастает на величину δ с увеличением разрядности вдвое;

– отдельные разряды обратного числа, оставаясь в целочисленном представлении, кроме изменения знака, выходят из двухпозиционного представления его. Но вес каждого i -го разряда соответствует его весу 2^{-i} (положению) в пределах разрядной сетки исходного числа с двоичным представлением. Это указывает на возможные отклонения, которые возникнут при реализации операции умножения множимого (делимого), представленного числом с разрядами в двухпозиционном виде (“0” и “1”), на множитель (делитель), структура которого представлена также в виде двухпозиционного числа, но:

- каждый из разрядов может быть со знаком плюс или минус;
- значение каждого текущего разряда (кроме первых двух вычисляемых), оставаясь целым числом, может отличаться от “0” и “1”.

6. Выводы

1. Предлагаемый метод вычисления ОВ кода числа, выполненный в алгебре матриц, реализует операцию деления (в классическом понимании операции), используя операцию умножения кода делимого на обратное значение кода делителя.
2. Замена операции деления на умножение обеспечивает правило коммутативности исходных данных, что даёт дополнительные преимущества при реализации операции в СП.
3. Операцию выполняем над числами в позиционной (например, двоичной) системе счисления. Результат по методу определения ОВ кода числа совпадает с результатом ОД, полученным классическим методом. При этом частное от деления (число) сохраняет позиционное представление числа, в котором и заданы исходные операнды (делимое и делитель). Но каждый разряд определяемой ОВ кода числа может принимать значения "0", "±1" или другое многозначное число с разными знаками и "весом", равным номеру позиции данного разряда позиционно представленного числа.
4. Определение ОВ кода числа по предлагаемому алгоритму для замены операции деления основано на использовании методов алгебры матриц. Оно базируется на использовании процедуры, аналогичной выполнению обратного хода метода факторизации матрицы с записью числа как полинома в формате РМП.
5. Операцию можно упростить. Частично это будет показано во второй части статьи. Но дополнительные исследования и анализ предложенных решений выполнения операции (по замене традиционной ОД и отдельных этапов её) следует провести. Поиски и разработки в плане комбинирования методов алгебр матриц и цифровой арифметики могут дать новые интересные решения.

Заключение

В рамках выделяемого для статьи объёма трудно предложить и изложить метод решения, а также все нюансы отдельных этапов метода. Предполагается, что метод даст толчок для дальнейшего его продвижения специалистами в области разработки, адаптации вычислительных методов для создания высокопроизводительных параллельных структур, особенно архитектур гетерогенного типа, обеспечивающих сверхвысокую производительность при снижении затрат на обработку единицы информации.

Идея выполнения операции деления (как и метода факторизации матрицы [2]) была предложена автором во время работы в МП ПАК "Вычислительные средства, программирование и технология" Института кибернетики им. В.М. Глушкова АН УССР, организованном в конце 80-х годов прошлого столетия для создания спецпроцессора новой архитектуры, и проработана в ТОО "ФИРМА "ПАК". В силу форсмажорных обстоятельств работы по спецпроцессору были остановлены. По этой причине была задержана и публикация. Настоящая статья подготовлена к передаче в печать в год 90-летия светлой памяти академика В.М. Глушкова – талантливого ученого-кибернетика и алгебраиста, менеджера и человека, которому наука и мы во многом обязаны.

Считаю приятным долгом выразить свою признательность В.П. Клименко, который стимулировал написание и оказал содействие в издании цикла статей [2–8] в направлении работ, поддержанном в своё время В.М. Глушковым.

СПИСОК ЛИТЕРАТУРЫ

1. Ледянкин Ю.Я. Единый технологический поток в организации вычислений – способ повышения производительности параллельных структур на процессорных элементах транспьютерного типа. – Киев, 1989. – 20 с. – (Препр. /АН УССР. Ин-т кибернетики им. В.М. Глушкова; 89-57).
2. Ледянкин Ю.Я. Способы параллельного решения систем $Ax = b$ в едином технологическом потоке решения задач математической физики / Ю.Я. Ледянкин // Математичні машини і системи. – 2013. – № 1. – С. 63 – 74.

3. Клименко В.П. Вычислительные системы. Тенденции развития архитектуры гетерогенных структур / В.П. Клименко, Ю.Я. Ледянкин // Математичні машини і системи. – 2013. – № 2. – С. 19 – 31.
4. Ледянкин Ю.Я. К вопросу преобразования и параллельного ввода граничных условий при решении краевых задач в едином вычислительном потоке / Ю.Я. Ледянкин // Математичні машини і системи. – 2012. – № 1. – С. 28 – 25.
5. Ледянкин Ю.Я. Методы взвешенных невязок, коллокаций, моментов. Способ параллельной реализации в едином вычислительном потоке решения задач математической физики / Ю.Я. Ледянкин // Математичні машини і системи. – 2012. – № 2. – С. 17 – 28.
6. Ледянкин Ю.Я. Метод Галеркина. Способ параллельной реализации задач математической физики в едином вычислительном потоке / Ю.Я. Ледянкин // Математичні машини і системи. – 2012. – № 3. – С. 69 – 80.
7. Ледянкин Ю.Я. Метод наименьших квадратов. Способ параллельной реализации в едином вычислительном потоке решения задач математической физики / Ю.Я. Ледянкин // Математичні машини і системи. – 2012. – № 4. – С. 59 – 69.
8. Ледянкин Ю.Я. Способ параллельной реализации метода Ньютона-Рафсона в едином технологическом потоке на спецпроцессоре для решения задач математической физики / Ю.Я. Ледянкин // Математичні машини і системи. – 2013. – № 4. – С. 38 – 49.
9. Карцев М.А. Арифметика цифровых машин / Карцев М.А. – М.: Гл. ред. Физ.-мат. лит., Наука, 1969. – 576 с.
10. Карцев М.А. Вычислительные системы и синхронная арифметика / М.А. Карцев, В.А. Брик. – М.: Радио и связь, 1981. – 360 с.

Стаття надійшла до редакції 13.11.2013