УДК 681.3

Ю.Я. ЛЕДЯНКИН*

СПОСОБ ПАРАЛЛЕЛЬНОЙ РЕАЛИЗАЦИИ МЕТОДА НЬЮТОНА-РАФСОНА НА СПЕЦПРОЦЕССОРЕ В ЕДИНОМ ВЫЧИСЛИТЕЛЬНОМ ПОТОКЕ

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

Анотація. *Розвивається ідея створення єдиного обчислювального (технологічного) потоку в організації паралельних обчислень на рівні складних структур даних як способу підвищення продуктивності пристроїв, що включають скалярний помножувач. Передбачається виключити операції ділення на кожній ітерації.*
Ключові слова: *задачі математичної фізики, єдиний технологічний потік, поліноми, складні структури даних, паралельні обчислення, спецпроцесор, скалярний помножувач.*

Аннотация. *Развивается идея создания единого вычислительного (технологического) потока в организации параллельных вычислений на уровне сложных структур данных как способа повышения производительности устройств, включающих скалярный умножитель. Предполагается исключение деления на каждой итерации.*
Ключевые слова*: задачи математической физики, единый технологический поток, полиномы, сложные структуры данных, параллельные вычисления, спецпроцессор, скалярный умножитель.*

Abstract. *It is developed the idea of creating a single computational (technological) thread to organize parallel calculations at the level of complex data structures as a way to improve the performance of devices which include a scalar multiplier. It is expected to exclude the division operations at each iteration.*
Keywords: *problems of mathematical physics, single technological thread, polynomials, complex data structures, parallel calculations, specialized processor, scalar multiplier.*

1. Введение

Поскольку современная вычислительная математика в основном ориентирована на последовательное выполнение арифметических операций, а возникает потребность в переходе к обработке сложных структур данных (ССД), необходимо разрабатывать новые методы, способы реализации существующих с их трансформацией под аппаратную реализацию в параллельной структуре.

В [1, 2] автором рассмотрены различные задачи математической физики (МФ) как стационарные, так и нестационарные, с различными краевыми условиями, с аппроксимацией явными и неявными схемами на различных сетках с методами конечно-разностной аппроксимации (МКР) с реализацией их на спецпроцессоре параллельного типа.

В [3] предложен и описан метод решения задач МФ методом конечных элементов (МКЭ) с аппроксимацией кусочно-гладкими полиномиальными функциями (сплайнами) в едином технологическом потоке (ЕТП) в математическом плане с последующим отображением структуры решаемой задачи на структуру устройства.

Основная цель, которая ставилась в этих работах, определить основной набор арифметико-логических операций, которые следует заложить в спецпроцессор (СП) параллельного типа, чтобы он смог решать такие задачи в ЕТП (в математическом плане). Отсюда следует, что и оборудование СП, реализующее такой ЕТП (в техническом плане), будет максимально технологически однотипным и будет его отражать. А это определит и экономическую составляющую проекта. Желательно, чтобы операнды в виде ССД обрабатывались за время, соизмеримое со временем выполнения арифметической операции. Тогда повышение производительности спецпроцессора будет гарантировано.

Такого рода ЕТП был предложен в [3]. В основу ЕТП заложена идея аппроксимации исходных аналитических уравнений с помощью сплайнов, записываемых в виде регуляр-

© Ледянкин Ю.Я., 2013

ного матричного представления (РМП). За время, соизмеримое со временем выполнения одной арифметической операции, скалярный умножитель (СУ) в составе процессорного элемента (ПЭ) СП параллельного типа вычисляет сумму парных произведений [2] коэффициентов полиномов (обрабатывается ССД).

Разработке и созданию архитектур на базе спецпроцессоров, ориентированных на класс конкретных задач, всегда сопутствовало естественное желание разработчиков расширить круг решаемых задач. При решении практических задач часто возникает потребность в решении нелинейных уравнений, нахождении корней полиномов, поиске общего решения двух уравнений, которые в общем случае, являясь нелинейными, описываются полиномами. Тем более, что точные формулы, как правило, не позволяют находить корни полиномов высших порядков.

С этой целью часто используют итерационный метод Ньютона-Рафсона (МНР), который эффективен в плане точности и скорости нахождения решения.

Но в том виде, в котором метод представлен и описан в [4, 5], он больше рассчитан на использование в машинах с последовательной обработкой данных. Это плохо подходит для СП параллельного типа, предложенного в [3], в котором предполагается обработка информации на уровне ССД.

2. Постановка задачи

Трансформировать метод Ньютона-Рафсона таким образом, чтобы его можно было использовать в структурах параллельного типа.

В общем виде итерационный процесс по методу Ньютона-Рафсона записывают так:

$$x_{r+1} = x_r - f(x_r)/f'(x_r). \tag{1}$$

Обозначим:

$$f(x_r) \tag{2}$$

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

$$f'(x_r) \tag{2а}$$

– значение производной $[f'(x_r)]$ от функции в точке $(x_r, f'(x_r))$.

Обозначим:

$$f(x) \rightarrow [\Phi_\gamma], \quad f'(x) \rightarrow [\Phi'_r] \tag{2б}$$

– значение функции и производной от нее в виде РМП;

$$[\Phi'_r]^{-1} \tag{2в}$$

– обратная матрица от $[\Phi'_r]$ такая, что

$$[\Phi'_r] * [\Phi'_r]^{-1} = E. \tag{3}$$

Рассмотрим метод на примере, взятом из [4]. Будем искать точку пересечения прямой

$$y = x + 2 \tag{4}$$

с кубической кривой

$$y = x^3 - x^2 + 3x + 1. \tag{5}$$

Точка пересечения этих двух линий имеет x-координату, определяемую из соотношения

$$x+2=x^3-x^2+3x+1, \tag{6}$$

$$p(x)=x^3-x^2+2x-1=0,\ \ p(0)=-1,\ \ p(1)=1. \tag{7}$$

3. Запись полиномов в виде РМП

Запишем полином $f(x)$

$$f(x)=a_0x^0+a_1x^1+a_2x^2+\ldots+a_kx^k \tag{8}$$

в виде [3] циклической матрицы $[f(x)]$ с РМП:

$$f(x)\rightarrow[f(x)]=\begin{bmatrix} a_0 & a_1 & a_2 & \ldots & a_k \\ & a_0 & a_1 & \ldots & a_{k-1} \\ & & a_0 & \ldots & a_{k-2} \\ & & & \ldots & \ldots \\ & & & & a_0 \end{bmatrix}. \tag{9}$$

Тогда матрица, транспонированная к $[f(x)]$, будет

$$[f(x)]^T=\begin{bmatrix} a_0 & & & \ldots & \\ a_1 & a_0 & & \ldots & \\ a_2 & a_1 & a_0 & \ldots & \\ \ldots & \ldots & \ldots & \ldots & \\ a_k & a_{1-1} & a_1 & \ldots & a_0 \end{bmatrix} \tag{10}$$

с записью в виде вектора-столбца $[f^*(x)]$ от функции, представленной в матричном виде:

$$[f^*(x)]=[a_0\,a_1\,a_2\ldots a_k]^T. \tag{11}$$

Произведение $f_1(x)$ и $f_2(x)$ с коэффициентами b равно

$$f_1(x)*f_2(x)=\begin{bmatrix} a_0 & a_1 & a_2 & \ldots \\ & a_0 & a_1 & \ldots \\ & & a_0 & \ldots \end{bmatrix}*\begin{bmatrix} b_0 & b_1 & b_2 & \ldots \\ & b_0 & b_1 & \ldots \\ & & b_0 & \ldots \end{bmatrix}=\begin{bmatrix} g_0 & g_1 & g_2 & \ldots \\ & g_0 & g_1 & \ldots \\ & & g_0 & \ldots \end{bmatrix}=$$

$$=\begin{bmatrix} (a_0b_0) & (a_1b_0+a_0b_1) & (a_0b_2+a_1b_1+a_2b_0) & \ldots \\ & (a_0b_0) & (a_1b_0+a_0b_1) & \ldots \\ & & (a_0b_0) & \ldots \end{bmatrix} \tag{12}$$

или

$$[f_{1_1}(x)]^T*[f_2^*(x)]=\begin{bmatrix} a_0 & & & & \\ a_1 & a_0 & & & \\ a_2 & a_1 & a_0 & & \\ . & . & . & & \\ a_k & a_{k-1} & a_{k-2} & \ldots & a_0 \end{bmatrix}*\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \ldots \\ b_k \end{bmatrix}=\begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ \ldots \\ g_k \end{bmatrix}\begin{matrix} x^0 \\ x^1 \\ x^2 \\ \ldots \\ x^k \end{matrix}. \tag{12a}$$

Зададим оператор дифференцирования функции в матричном виде $[A]$ так:

$$[A]=\begin{bmatrix} 0 & 1 & 0 & 0 \\ & 0 & 2 & 0 \\ & & 0 & 3 \\ \dots & \dots & \dots & \dots \end{bmatrix} dx. \tag{13}$$

Применительно к рассматриваемым уравнениям (4), (5), с учетом (2) и (3), запишем функцию $p(x)$ (8) в виде РМП:

$$f\left(x_0\right) \rightarrow [\Phi_0]=\begin{bmatrix} 1 & 3 & -1 & 1 \\ & 1 & 3 & -1 \\ & & 1 & 3 \\ & & & 1 \end{bmatrix}. \tag{14}$$

4. Решение поставленной задачи

Рассмотрим выражение (1) в такой записи:

$$x_{r+1}=(x_r f^{|}(x)-f(x_r))/\ f^{|}(x_r). \tag{15}$$

Выражение (15) в общем виде с применением РПМ можно записать:

$$x_{r+1}[\Phi_{\text{г}}^{'}]=x_r[\Phi_{\text{г}}^{'}]-[\Phi_{\text{г}}). \tag{15a}$$

Умножив все члены левой и правой частей полученного выражения справа на обратную матрицу $[\Phi_{\text{г}}^{'}]^{-1}$, получим

$$x_{r+1}\ [\Phi_{\text{г}}^{'}]*[\Phi_{\text{г}}^{'}]^{-1}=x_r[\Phi_{\text{г}}^{'}]*[\Phi_{\text{г}}^{'}]^{-1}-[\Phi_{\text{г}}^{'}]*[\Phi_{\text{г}}^{'}]^{-1} \tag{16}$$

или

$$x_{r+1}*E=x_r*E-\left[\Phi_r\right]*[\Phi_r^{'}]^{-1}. \tag{17}$$

Рассмотрим пример (4), (5).

Точка пересечения этих двух линий имеет x координату, определяемую из соотношения

$$x+2=x^3-x^2+3x+1 \tag{18}$$

или

$$p\left(x\right)=x^3-x^2+3x+1=0,\quad p(0)=-1,\quad p(1)=1. \tag{19}$$

Применительно к рассматриваемым выражениям, с учетом (9), положив $p(x)\equiv f(x)$, будем рассматривать функцию $p(x)$ в виде РМП:

$$f(x_0)\rightarrow[\Phi_0]=\begin{bmatrix} 1 & 3 & -1 & 1 \\ & 1 & 3 & -1 \\ & & 1 & 3 \\ & & & 1 \end{bmatrix}=[\Phi_0^{*}]\rightarrow f^{*}(x). \tag{20}$$

Продифференцируем функцию $f(x_r)$, получив $f^{'}(x_r)$ в виде РМП по правилу [12а] с учетом, что оператор дифференцирования определен матрицей (13):

$$f^{'}(x_r)\rightarrow[\Phi_{\text{г}}^{'}]=[A]*[\Phi_r]dx=$$

$$=[\mathrm{A}]*\begin{bmatrix}1 & 3 & -1 & 1\\ & 1 & 3 & -1\\ & & 1 & 3\\ & & & 1\end{bmatrix}=[\mathrm{A}]*\begin{bmatrix}-1\\2\\-1\\1\end{bmatrix}dx=\left.\begin{bmatrix}2\\-2\\3\\0\end{bmatrix}\right|\begin{matrix}x^0\\x^1\\x^2\\x^3\end{matrix}\Rightarrow[\Phi_{\mathrm{r}}^{'}]=$$

$$=[2\text{-}2\ 3\ 0]^{T}\{x\}. \tag{21}$$

После этого уравнение (17) можно записать таким образом:

$$x_1 = x_0 * E - [\Phi_0] * [\Phi_0^{'}]^{-1}. \tag{22}$$

Обозначим величину приращения $\delta(x_r)$, $r = 0,1,...,n$ на каждой итерации:

$$\delta(x_r) = [\Phi_r] * [\Phi_r^{'}]^{-1}. \tag{23}$$

Замечание

В структурах вычислительной техники из набора арифметических операций (сложение, вычитание, умножение и деление) наиболее тяжелой (по длительности выполнения) и по наименьшей точности (по погрешности округления) является операция деления. От нее в спецпроцессорах стремятся (по возможности) избавляться (в крайнем случае, не делать ее базовой).

Заменив $[\Phi_{\mathrm{r}}^{'}]$, находящуюся в (1, 13) в знаменателе (делитель на каждой итерации), на обратную $[\Phi_{\mathrm{r}}^{'}]^{-1}$ к производной функции (сделав ее таким образом сомножителем) в уравнении (22), мы на каждой итерации заменим выполнение операции деления на умножение.

Далее вся процедура вычисления нового значения x_{r+1} сводится к выполнению итерационного процесса

$$x_{r+1} = x_r - \delta(x_r) \tag{24}$$

над ССД, то есть к процедуре вычисления сумм парных произведений. К ним сводится решение задач МФ [3] с помощью МКЭ или метода конечных разностей (МКР). Это практически отвечает общей идеологии построения параллельного спецпроцессора из ПЭ с СУ в его основе.

Вычислим матрицу $[\Phi_{\mathrm{r}}^{'}]^{-1}$, обратную к матрице от производной $[\Phi_{\mathrm{r}}^{'}]$ для рассматриваемого выше примера (с помощью метода факторизации без деления на прямом ходе [10]):

$$[\Phi_{\mathrm{r}}^{'}]^{-1}=\left[\begin{array}{cccc|cccc}2 & -2 & 3 & 0 & 1 & 0 & 0 & 0\\ & 2 & -2 & 3 & 0 & 1 & 0 & 0\\ & & 2 & -2 & 0 & 0 & 1 & 0\\ & & & 2 & 0 & 0 & 0 & 1\end{array}\right]=\begin{bmatrix}1/2 & 1/2 & -1/4 & -1\\ 0 & 1/2 & 1/2 & -1/4\\ 0 & 0 & 1/2 & 1/2\\ 0 & 0 & 0 & 1/2\end{bmatrix}$$

$$\left.\begin{matrix}x_{1j}\\x_{2j}\\x_{3j}\\x_{4j}\end{matrix}\right|\begin{matrix}\overline{0} & \overline{0} & \overline{0} & \overline{1/2}\\ 0 & 0 & 1/2 & 1/2\\ 0 & 1/2 & 1/2 & -1/4\\ 1/2 & 1/2 & -1/4 & -1\end{matrix}\Bigg|. \tag{25}$$

(Перемножение матриц дает единичную матрицу $[\Phi_r^{'}] * [\Phi_r^{'}]^{-1} = E$).

Произведем вычисления на основе изложенного способа и примера (4), (5).

Зададим начальное приближение $x_0 = 0,5$ и вычислим раздельно значения $[\Phi_r]$ и $[\Phi_r^{'}]$:

$$[\Phi_0] = \begin{bmatrix} -1 & 2 & -1 & 1 \\ & -1 & 2 & -1 \\ & & -1 & 2 \\ & & & -1 \end{bmatrix} \begin{bmatrix} x_0^0 = 1 \\ x_0^1 = 1/2 \\ x_0^2 = 1/4 \\ x_0^3 = 1/8 \end{bmatrix} = ((-1*1)+(2*1/2)-(1*1/4)+(1*1/8)) =$$

$$= -1/8 = -0,125 = [\Phi_0]^T. \qquad (26)$$

$$[\Phi_r^{'}]^{-1} = \begin{bmatrix} 1/2 & 1/2 & -1/4 & -1 \\ 0 & 1/2 & 1/2 & -1/4 \\ 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 \end{bmatrix} * \begin{bmatrix} x_0^0 = 1 \\ x_0^1 = 1/2 \\ x_0^2 = 1/4 \\ x_0^3 = 1/8 \end{bmatrix} =$$

$$= ((1/2 * \cdot 1)+(1/2*1/2)-(1/4*1/4)-(1*1/8) = -9/16 = -0,5625 = [\Phi_0^{|}]^{-1}. \qquad (27)$$

После вычисления $[\Phi_0]$ и $[\Phi_0^{'}]^{-1}$ по их произведению определим приращение на первой итерации:

$$\delta(x_1) = [\Phi_0] * [\Phi_0^{'}]^{-1} = -0,125 * 0,5625 = -0,0703125 \qquad (28)$$

и коррекцию (24) начального значения $x_0 = 0,5$.

После первой итерации первое приближенное решение x_1 равно:

$$x_1 = x_0 - \delta(x_1) = 0,5 - (-0,0703125) = 0,5703125, \qquad (29)$$

что практически близко к значениям, приведенным в [4].

Следующие итерации вычисляют по аналогии, используя на второй итерации значение x_1, полученное в (29) на первой итерации, и т.д.

В результате имеем:

$x_2 = 0,5698634$;

$x_3 = 56984141$;

$x_4 = 569878258$;

$x_5 = 0,569842131$;

$x_6 = 0,56984038$;

$x_7 = 0,569840295$,

что совпадает со значением, вычисленным в исходном примере [4].

Замечания

1. Вычисления, проведенные по уравнениям (21) – (29), сведены к раздельному получению на каждой итерации значения функции $[\Phi_r]$ и значения производной от нее $[\Phi_r^{'}]$.

Значения $x_r^0, x_r^1, x_r^2, x_r^3$ (которые использовались при вычислении в (26), 27)), вычисляются с помощью СУ путем умножения соответствующих компонент $(x_r * x_r) = x_r^2$,

$\left(x_r * x_r^2\right) = x_r^3$, … в двоичном дополнительном, например, коде методом умножения младшими разрядами обоих сомножителей вперед. Это позволит получить младшие разряды на первом микротакте умножения с тем, чтобы использовать их на следующем микротакте в качестве исходных данных в других СУ при получении $x_r^2 * x_r$ или $x_r^2 * x_r^2$.

2. Вычисление обратной матрицы $[\Phi_r^{'}]^{-1}$, которая определяет полином в записи РМП проще. Оно предполагает циклическую запись коэффициентов полинома, что в результате дает верхнюю треугольную матрицу. На главной диагонали ее стоит число, которое равно значению коэффициента при неизвестном полиноме в нулевой степени. За счет этого мы, применяя для обращения матрицы метод исключения, на практике исключаем прямой ход процедуры факторизации матрицы, на котором формируется нулевая нижняя треугольная матрица. Выполняется только обратный ход, на котором надо выполнить единственное деление (или взятие обратной величины от коэффициента на главной диагонали $l_{mm} = 1 / k_{mm}$, m – показатель степени полинома $f(x)$). Но при РМП диагональный элемент для $k_{11} = k_{ii} = k_{mm}$ (одинаков). Далее на l_{mm} следует умножать столбцы матрицы $[\Phi_r^{'}]^{-1}$.Таким образом, значения коэффициентов обратной матрицы $[\Phi^{'}]^{-1}$ будут вычислены.

Отметим, что вычисление обратной матрицы $[\Phi_r^{'}]^{-1}$ в нашем случае оправдано многократным использованием ее в ходе итерационного процесса. Обычно считают, что проще получить решение $Ax = y$ сразу, без обращения матрицы.

3. Раздельное вычисление $[\Phi_r]$, $[\Phi_r^{'}]$, как было показано ранее в (26) – (27), обеспечивает понижение порядка результирующей m_R матрицы и матриц сомножителей m_1 и m_2:

$$m_R = m_1 + m_2 + 1,$$

где m_1 и m_2 – порядки полиномов функции и производной от нее (порядок результирующего полинома на единицу больше суммы порядков перемножаемых полиномов).
Предлагается и другой вариант вычисления приращения (x_r) итерационного процесса по методу Ньютона-Рафсона.

Второй вариант вычислительного процесса

Положим начальное значение x_0 =0,5 и запишем уравнение (23) по выражению (21), когда вычисление величины (x_1) выполняется сразу путем перемножения матриц $[\Phi_0]$ и $[\Phi_0^{|}]^{-1}$:

$$\delta\,(x_1) \rightarrow [\Phi_0]^T * [\Phi_0^{|}]^{-1} =$$

$$= \begin{bmatrix} -1 & & & & & & \\ 2 & -1 & & & & & \\ -1 & 2 & -1 & & & & \\ 1 & -1 & 2 & -1 & & & \\ 0 & 1 & -1 & 2 & -1 & & \\ 0 & 0 & 1 & -1 & 2 & -1 & \\ 0 & 0 & 0 & 1 & -1 & 2 & -1 \end{bmatrix} * \begin{bmatrix} 1/2 \\ 1/2 \\ -1/4 \\ -1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{vmatrix} x_0^0 = 1 \\ x_0^1 = 1/2 = 0,5 \\ x_0^2 = 1/4 = 0,25 \\ x_0^3 = 1/8 = 0,125 \\ x_0^4 = 1/16 = 0,0625 \\ x_0^5 = 1/32 = 0,03125 \\ x_0^6 = 1/64 = 0,015625 \end{vmatrix} =$$

$$=(-0,5+0,25+0,1875+0,0625–0,078125+0,0234375–0,015625)=-0,0703125. \qquad (31)$$

$$x_1 = 0{,}5 + 0{,}0703125 = 0{,}5703125. \quad (32)$$

Следующая итерация начинается с возведения с помощью СУ значения x_1 =0,5703125 в степени n=2, 3, 4, 5, 6, на которые следует умножить вектор $\delta(x)$=[-1/2+1/2+3/4+1/2-5/4 3/4-1], чтобы получить новое значение x_2, то есть вычислить $\delta(x_2)$ и в целом по (29) – значение x_2.

Расчеты, проведенные по обоим вариантам, совпадают. Меняются порядок вычисления поправки и размерность матриц. Их размерность определяется величиной показателя степени полинома.

Вариант вычислений, который следует применить, зависит от разработчика спецпроцессора или от программиста, который применяет уже готовое устройство.

К вопросу о вычислении функции и производной от неё применительно к методу Ньютона-Рафсона с РМП полиномов заметим следующее.

Следуя Д. Мак-Кракену и Х. Дорну [5, с. 104], для вычисления полиномов по оптимальному правилу Горнера требуется n умножений и n сложений. Число умножений при вычислении по формуле (1) составляет $n(n+1)/2$, если каждая степень получена последовательным способом.

В нашем случае производится параллельная обработка сомножителей. Это позволяет перевести вычислительный процесс с уровня обработки значений чисел на уровень обработки ССД, которые представлены числами.

Затем, если в СУ заложено выполнение операции умножения чисел, начиная с младших разрядов, вперед обоих сомножителей, то после первого микротакта будут получены значения младших разрядов результата произведения. Вычисление значений $x^2 = x * x$, $x^4 = x^2 * x^2$, $x^8 = x^4 * x^4$, $x^{10} = x^8 * x^2$, а также получение в параллель нечетных значений степеней $x^3 = x^2 * x^1$, $x^5 = x^3 * x^{2\,5}$, $x^7 = x^5 * x^2$, $x^9 = x^5 * x^4$, полагает задержку всего на четыре микротакта. Для чисел, представленных в форме фиксированной запятой с разрядностью, например, n=128 и при обработке чисел с удвоенным числом разрядов, дополнительные задержки при умножении еще на 4 микротакта составят около ≤ 2 %, что можно считать несущественным. При одновременном сложении сомножителей, которое выполняется параллельно с умножением как одно сложение, такое решение дает неоспоримый выигрыш.

Отметим также, что если процедура выполняется на оборудовании, предназначенном для решения задач в ЕТП, то она вписывается в процесс, обеспечивая дополнительные преимущества.

Особо следует сказать, что в течение всего итерационного процесса необходимо выполнить только одно деление при вычислении обратной величины от диагонального элемента при обращении матрицы (производной от функции, записанной в циклическом виде). Традиционные методы требуют выполнения тяжелой и всегда менее точной операции деления на каждой итерации.

5. Выводы

1. Предлагаемый способ решения уравнений (или вычисления корней полиномов) по методу Ньютона-Рафсона позволит использовать спецпроцессоры параллельного типа и выполнять вычислительный процесс с использованием ССД на высокопроизводительных структурах.
2. Эти структуры разработаны для решения уравнений, которыми описывают задачи МФ с применением аппроксимаций, используемых в методах МКЭ, МКР и др. [3] с обработкой в ЕТП.

3. Предложенная выше процедура замены операции деления полиномиальных функций на операцию умножения на матрицу, обратную к исходной матрице, может быть успешно применима и в других случаях.

Для этого следует их описать в виде РМП, вычислить обратную матрицу и осуществить в таком виде операцию перемножения полиномов.

4. Применение предлагаемого подхода возможно и без использования в спецтехнике. В таком случае выигрыш в повышении скорости вычислений можно ожидать меньше. Но поскольку погрешность округления при умножении всегда меньше, чем при делении, следует ожидать повышения точности или сокращения количества итераций.

5. Предлагаемый способ еще раз напоминает о необходимости создания параллельных методов вычисления уравнений и решения практических задач. Следует создавать новые методы, искать способы преобразования существующих методов, что позволит реально от машин с последовательной организацией вычислений при обработке чисел перейти к параллельным структурам, с обработкой ССД. В процессе обсуждения статьи возникли вопросы и замечания, на которые следует ответить.

1. Область применения МНР. Она уже определена и описана в научно-технической и учебной литературе. Автор предложил и показал, как можно МНР реализовать на параллельном спецпроцессоре для решения задач МФ в режиме ЕТП с обработкой информации на уровне ССД. Это, естественно, расширит круг задач, решаемых на спецпроцессоре. Хорошим примером могут служить архитектуры, которые развиваются в последнее время фирмами nVidea Tesla, AMD, и др. на базе графических процессоров (GPU) и кластерных систем классической структуры (CPU) [6, 7].

2. Вопрос: «Зачем применять МНР, если есть подпрограмма (ПП) ZEROIN [8]?». Из машинных алгоритмов, предназначенных для нахождения действительного корня функции, подпрограмма ZEROIN является одной из лучших. Алгоритм сочетает безотказность метода бисекции с асимптотической скоростью метода секущих (в случае гладких функций). ПП алгоритма ZEROIN хорошая, но она не может быть прямо поставлена на СП, который реализует ЕТП с ПЭ с СУ в своей основе. Кроме того, иногда следует находить корни в комплексной области и не всегда функции бывают гладкие.

Тем не менее уравнения, заложенные в метод секущих ПП ZEROIN:

$$x_{k+1} = x_k - \delta(x_k), \tag{33}$$

$$\delta(x_k) = \frac{f_k * (x_{k-1} - x_k)}{(f_{k-1} - f_k)}, \tag{34}$$

где $f_k = f(x_k)$ – значение функции $f(x_k)$ в точке x_k, по своей структуре записи мало отличается от выражения (24), рассмотренного ранее для МНР, где в качестве поправки $\delta(x_k)$ вычисляется отношение значения функции к производной от неё с помощью выражения (23) в записи РМП:

$$\delta(x_k) = [\Phi_k] * [\Phi_k]^{-1}, \quad f(x_k) \rightarrow [\Phi_k]. \tag{35}$$

Поскольку x_k и x_{k-1} – это числовые значения абсцисс, а f_k и f_{k-1} – значения функций $f(x_k)$ и $f(x_{k-1})$ в этих точках и представлены многочленом (полиномом), то возможность применения РМП не вызывает сомнений. Вычислению подлежит значение $f(x_k)$ функции в точке x_k, поскольку предыдущее значение в точке x_{k-1} ранее вычислено.

Для исключения на каждой итерации операции деления следует (по аналогии с (25)) один раз вычислить обратную (с единственным делением) матрицу от функции

$f(x) \rightarrow [\Phi_0]^{-1}$. Она представлена верхней треугольной матрицей, на главной диагонали которой стоит число, равное свободному члену многочлена (полинома).

Учитывая, что в конкретном случае пользователь исследует один многочлен, и свойство ассоциативности многочленов, по которому их сложение (вычитание) сводится к сложению (вычитанию) коэффициентов при неизвестных с одинаковой степенью, операция вычитания $[\Phi_k] - [\Phi_{k-1}]$ упрощается.

Замечание

Следует помнить, что методы МНР и секущих можно использовать в комплексной плоскости [9]. На практике используют комбинированный метод хорд и касательных (МНР) в разных сочетаниях значений произведения функции и второй производной относительно нуля

$$f(x) * f^{11}(x) < 0, \tag{36}$$

$$f(x) * f^{11}(x) > 0, \tag{37}$$

когда для случая (36) метод хорд даёт приближение корня с недостатком, а метод касательных – с избытком. Или наоборот – для случая (37).

Выражения, которые применяют при вычислениях, по своей структуре аналогичные с ранее приведенными (для МНР и ZEROIN). Так, для условия (36) они имеют вид [9]:

$$a_{n+1} = a_n - \delta(a_n), \tag{38а}$$

$$\delta(a_n) = f(a_n) * (b_n - a_n)/[f(b_n) - f(a_n)], \tag{38б}$$

$$b_{n+1} = b_n - \delta(b_n), \tag{38в}$$

$$\delta(b_n) = f(b_n)/[f^1(b_n)]. \tag{38г}$$

В этом случае приближения корня лежат со стороны b и получают по методу касательных, а со стороны a – значения, полученные по методу хорд.

Для условия (37) выражения аналогичные:

$$a_{n+1} = a_n - \delta(a_n), \tag{39а}$$

$$\delta(a_n) = f(a_n)/[f^1(a_n)],$$

$$b_{n+1} = b_n - \delta(b_n), \tag{39б}$$

$$\delta(b_n) = f(b_n) * (b_n - a_n)/[f(b_n) - f(a_n)],$$

но приближения значения корня со стороны a получены по методу касательных, а со стороны b – значения, полученные по методу хорд.

При этом для исключения операции деления необходимо от матрицы (с помощью которой в знаменателе выражения (39б) записан полином) вычислить обратную (от $[\Phi_0]^{-1}$ или $[\Phi_0^1]^{-1}$), а затем по выражению вида (35) вычислить поправку.

Вычисления прекращают, когда по методу бисекции достигается требуемая точность:

$$\xi = ½ (x_н - x_{из}), \tag{40}$$

где $x_н, x_{из}$ – приближенные значения корня с недостатком и избытком соответственно.

Поскольку вычисление значений (знаков) функции производной (первой от неё и второй) с помощью РМП сводится к выполнению обычной операции умножения матрицы (для дифференцирования $[A]$) (13) на вектор-столбец функции (или её первой производной), можно легко определить исходную точку. В качестве её следует выбирать тот конец отрезка $[a,b]$, где знаки функции и второй производной совпадают. А по времени такое произведение в базисе РМП выполняется как обычная операция умножения, в то время как для вычислительных машин классической структуры это сложная процедура.

Пример.

Значение функции (для $x_0 = 0,5$, например) равно

$$f(x) = x^3 - x^2 + 3x + 1 \rightarrow [13\text{-}11] * [x^0 x^1 x^2 x^3]^T =$$

$$= [1*1 + 3*0,5 - 1*0,25 + 1*0,0625] = 2,3125 > 0. \qquad (41)$$

В соответствии с (14) первая производная от исходной функции (8) равна произведению матрицы $[A]$ (13) на вектор $[f^*(x)]$ (20) от функции $f(x)$.

Для $x_0 = 0,5$ значение первой производной $f^1(x) \rightarrow [\Phi_0^1]$ равно

$$f^1(x) \rightarrow [\Phi_0^1] = [A] * [\Phi_0]\, dx = [A] * [1\ 3\ \text{-}1\ 1]^T\, dx = \begin{bmatrix} 3 \\ -2 \\ 3 \\ 0 \end{bmatrix} \Bigg| \begin{matrix} x^0 \\ x^1 \\ x^2 \\ x^3 \end{matrix} =$$

$$= [3*1 - 2*0,5 \quad 3*0,25] = 2,75 > 0, \qquad (42)$$

и значение второй производной от функции $f(x)$ (8) равно

$$f^{11}(x) \rightarrow [\Phi_0^{11}] = [A] * [\Phi_0^1]\, dx = [A] * [3\text{-}2\ 3\ 0]^T\, dx = \begin{bmatrix} -2 \\ 6 \\ 0 \end{bmatrix} \Bigg| \begin{matrix} x^0 = 1 \\ x^1 = 0,5 \\ x^2 = 0,25 \end{matrix} =$$

$$= (-2*1 + 6*0,5) = 1 > 0. \qquad (43)$$

Произведение значений функции $f(x)$ и второй $f^{11}(x)$ производной от неё (8) равно

$$f(x) * f^{11}(x) = 2,3125 * 1 = 2,3125 > 0. \qquad (44)$$

Выполняется условие (37), которое и определяет исходную точку.

Рассмотренный выше материал по нахождению нулевых значений полиномов даёт основание предполагать, что в такой постановке способ распараллеливания и обработки исходных данных в виде ССД может быть применен при решении общих задач и на существующих гетерогенных кластерных структурах типа GPU – CPU, выполняющих векторные операции. Требуется лишь доработка ПП.

СПИСОК ЛИТЕРАТУРЫ

1. Ледянкин Ю.Я. Вопросы создания спецпроцессоров, проблемно-ориентированных на решение задач математической физики / Ледянкин Ю.Я. – Киев: ИК АН УССР, 1980. – 54 с. – (Препринт / АН УССР, Ин-т кибернетики; 80-15).

2. Ледянкин Ю.Я. Принципы построения и структуры спецпроцессоров, ориентированных на решение задач математической физики / Ледянкин Ю.Я. – Киев: ИК АН УССР, 1981. – 45 с. – (Препринт / АН УССР, Ин-т кибернетики; 81-41).

3. Единый технологический поток в организации вычислений – способ повышения производительности параллельных структур на процессорных элементах транспьютерного типа / Ледянкин Ю.Я. – Киев, 1989. – 20 с. – (Препринт / АН УССР, Ин-т кибернетики им. В.М. Глушкова; 89-57).
4. Фокс А. Вычислительная геометрия. Применение в проектировании и на производстве / А. Фокс, М. Пратт; пер. с англ. – Л.: Мир, 1982. – 304 с.
5. Мак-Кракен Д. Численные методы и программирование на Фортране / Д. Мак-Кракен, У. Дорн; пер. с англ. – М.: Мир, 1977. – 585 с.
6. Адинец А. Графический вызов суперкомпьютерам / А. Адинец, В. Воеводин // Открытые системы. – 2008. – № 4. – С. 35 – 41.
7. Губайдуллин Д.А. Об особенностях использования архитектуры гетерогенного кластера для решения задач механики сплошных сред / Д.А. Губайдуллин, А.И. Никифоров, Р.В. Садовников // Вычислительные методы и программирование. – 2011. – № 12. – Р. 450 – 460.
8. Форсайт Дж. Машинные методы математических вичислений / Форсайт Дж., Малькольм М., Моулер К.; пер. с англ. – М.: Мир, 1980. – 280 с.
9. Численные методы / Н.И. Данилина, Н.С. Дубровская, О.П. Кваша [и др.]. – М.: Высшая школа, 1976. – 368 с.
10. 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.

Стаття надійшла до редакції 13.05.2013