УДК 004.00

НОВЫЕ ЧИСЛЕННЫЕ МЕТОДЫ И ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ НА АРХИТЕКТУРЕ ФУНКЦИОНАЛЬНО-ОПЕРАТОРНОГО ТИПА

Гливенко Е.В., д.т.н., проф., E-mail: deevagal@yandex.ru
Петрова Г.Н., к.ф.-м.н
НИИ вычислительных комплексов им. М.А. Карцева, Москва, Россия

Аннотация. В статье предлагается ряд новых численных методов, опирающихся на фундаментальные достижения современной геометрии. Эти методы обладают естественным параллелизмом и могут эффективно использоваться с применением многопроцессорных компьютеров.
Ключевые слова: многопроцессорный компьютер, параллелизм, естественный параллелизм, параллельные структуры.

NEW NUMERICAL METHODS AND PARALLEL COMPUTING BASED ON ARCHITECTURE OF FUNCTIONAL-OPERATION TYPE

Glivenko E., D.ofSci., Prof., E-mail: deevagal@yandex.ru
Petrova G., Ph.D., Associate Professor
NIIVK, Moscow, Russia

Abstract. A number of new numerical methods based on fundamental successes of the modern geometry are proposed. These methods possess the natural parallelism and can be effectively used with the help of multiprocessor computers.
Keywords: multiprocessor computer, parallelism, natural parallelism, parallel structures.

Введение

Проблема распараллеливания алгоритмов при реализации на параллельных структурах является одной из важных современных задач. Использование параллельных структур не только решает вопрос выигрыша во времени, но и помогает избавиться от накопления ошибок вычислений. Например, идея о том, что обеспечение компьютера быстродействующей элементной базой может снять все проблемы «хорошего» решения задачи, упирается в проблему накопления ошибок вычислений, которая может стать определяющей из-за ограниченности разрядной сетки. Поэтому при создании вычислительных методов, предполагающих большое количество вычислений, естественный параллелизм этих методов имеет существенное значение. Под естественным параллелизмом данных подразумевается их представление в виде больших массивов однотипной информации, обрабатываемой по одному алгоритму.

Это обеспечивает возможность перейти от последовательной обработки одной точки функции к одновременной обработке фрагмента функции [1].

В нашем случае речь пойдет о решении систем уравнений с использованием неподвижных точек непрерывного преобразования пространства, исследуемых в современной геометрии [2].

Параллельные вычисления на архитектуре функционально-операторного типа

Примером такого использования может служить известный метод решения систем линейных уравнений, записанных в виде:

$$\begin{cases} x_1 = a_{11}x_1 + a_{12}x_2 + \dots a_{1n}x_n \\ x_2 = a_{21}x_1 + a_{22}x_2 + \dots a_{2n}x_n \\ \dots \\ x_n = a_{n1}x_1 + a_{n2}x_2 + \dots a_{nn}x_n \end{cases}$$

Эту запись можно рассматривать как преобразование n-мерного пространства, если координаты в правых частях считать координатами прообраза, а координаторы левых частей координатами образца. Неподвижная точка этого преобразования соответствует случаю, когда правые координаты совпадают с левыми, т.е. это будут координаты решения системы. Известно, что, если матрица A={aij} осуществляет сжатое отображение, т.е. если ее норма меньше единицы, то последовательным приближением можно найти решение. Предлагаемый нами метод рассматривает то же преобразование, то уже в случае произвольной матрицы A={aij}. Правда, задача ставится несколько иначе. Рассматривается некоторая область пространства и задача сводится к определению того факта, что решение системы лежит в этой области. Уменьшая размер области, мы можем уточнять решение.

Конкретно, в наших алгоритмах границей области считается n-мерный симплекс. Вершины этого симплекса преобразуются с помощью векторов. Оказывается, что если эти векторы лежат все вместе в некотором полупространстве, то это означает, что симплекс не содержит неподвижной точки, т.е. решения системы. Если же такого полупространства не найдется, значит, решение системы лежит внутри симплекса. Это обстоятельство выявляется в предлагаемом алгоритме почти без счета. Предлагаемый метод работает, начиная с $n > 7$, т.е. для систем большого размера. Причем, чем больше n, тем метод эффективнее.

То же непрерывное преобразование мы используем для решения систем нелинейных уравнений [3], а именно, систем вида:

$$\begin{cases} F_1(x_1, x_2, ..., x_n) = 0 \\ F_2(x_1, x_2, ..., x_n) = 0 \\ ... \\ F_n(x_1, x_2, ..., x_n) = 0 \end{cases}$$

В этом случае преобразование имеет вид:

$$\begin{cases} x_{1\text{нов}} = F_1(x_1, x_2, ..., x_n) + x_{1\text{стар}} \\ x_{2\text{нов}} = F_2(x_1, x_2, ..., x_n) + x_{2\text{стар}} \\ ... \\ x_{n\text{нов}} = F_n(x_1, x_2, ..., x_n) + x_{n\text{стар}} \end{cases}$$

Вместо симплекса в этом случае рассматривается сфера, ограничивающая область. Задача о нахождении решения внутри области решается методом компьютерного моделирования с применением топологических теорем о степени отображения.

Дело в том, что в случае нелинейных систем уравнений имеется не одно решение, а несколько. В предлагаемых методах речь идет о нахождении всех решений. Сейчас инженеры ищут только одно решение, пользуясь методом Ньютона, основанным на линеаризации системы, причем начальное приближение для этого ищется из сторонних соображений.

Нами рассмотрен еще один метод решения нелинейных систем. Мы называем его методом выхода в (n+1)мерное пространство. Здесь мы уже не пользуемся непрерывным отображением, а рассматриваем функции:

Z1 = F1 (x1, x2, …, xn),

Z2 = F1 (x1, x2, …, xn),

…

Zn = Fn (x1, x2, …, xn),

для которых решение самой системы ищется как пересечение нулевых линий уровня этих функций.

Все предлагаемые методы обладают естественным параллелизмом, поэтому их использование на параллельных структурах должно оказаться эффективным.

Авторы много лет занимались разработкой алгоритмов и программ для параллельной архитектуры функционально-операторного типа, реализованной в СуперЭВМ М10.

В основе создания машины М10 лежит реализация идей, взятых из математической области, называемой функциональный анализ [4]. В отличие от математического анализа (хорошо нам знакомого), где вводились понятия непрерывности функции и дифференцируемости в отдельной точке, в функциональном анализе роль отдельной точки играет уже целая функция одного и нескольких переменных, преобразования которых осуществляет оператор.

Если в однопроцессорной машине роль аргумента играет число, задаваемое конечным числом разрядов (приближенно), то в машине М-10 роль аргумента играет функция одного, как это было в реализации, или нескольких переменных. Преобразователем чисел в однопроцессорной машине является одно из математических действий (сложение и т.д.). Преобразование в машине М-10 названо оператором по аналогии с функциональным анализом, который преобразует одну функцию в другую.

Как в однопроцессорной машине нельзя реализовать число бесконечным числом разрядов, так и в машине М-10 функция представлена не в бесконечном множестве точек, а лишь в конечном их числе.

$\boldsymbol{f \quad (агм)_n}$ $\rightarrow$ функция от агм x в n точках.

Реализацию машины М-10 можно поделить на так называемую «многопроцессорную часть» - арифметику, которая работает с любыми функциями и «символьную часть» - арифметику, которая работает с функциями, задаваемыми в тех же точках, что и функции «многопроцессорной арифметики», но здесь уже каждая функция может принимать только два значения – ноль или единицу. Это соответствует работе с так называемыми характеристическими функциями в рамках функционального анализа. С ее помощью можно было выделять отдельные области задания функций многопроцессорной арифметики. Состояния характеристической (символьной) функции, где фиксируются признаки вычислений (такие, как переполнение, знак результата, равенство нулю результата и другие), обрабатываются по правилам булевой алгебры и могут быть использованы в качестве сигнала управления вычислением в отдельной точке функции – включить (+), выключить (–), сохранить ($\perp$).

Первый и главный вопрос при создании машины М-10 заключался в том, чтобы на машине М-10 мог бы быть реализован любой алгоритм, т.е. чтобы комплект базовых операторов, реализуемых в машине, мог реализовать любое преобразование функций в виде последовательного применения базовых операторов. Работы по поиску таких “полных систем” базовых операторов велись в рамках теоретических исследований [5]. Введение операторов, реализуемых под маской (маской являлась функция характеристической арифметики) решило этот вопрос, т.к. с помощью маскирования можно многопроцессорную М-10 превратить в однопроцессорную, замаскировав все точки задания функций, кроме одной.

Под руководством главного конструктора М.А. Карцева идеи функциональной многопроцессорной архитектуры были реализованы в машине М-10 (1971г.).

Основными техническими новациями явились:

– многопроцессорное арифметико-логическое устройство (АЛУ), объединяющее две функциональные линейки на 16 точек функции каждая (слоги 1А, 2А команды), программно-перестраиваемое в 16, 32, 64, 128-разрядные модули с суммарным быстродействием 5,1·106 оп/с;

– символьная (R1÷R3) и адресная (АМ1÷АМ3) арифметики, контролирующие состояние системы в целом (слог У), работающие синхронно с АЛУ;

– двухуровневая память прямого доступа (ОП) объемом 5 Мбайт, КЭШ память 1-го уровня объемом 8×64К слов.

Связь между АЛУ и ОП осуществляют два программируемых информационных канала с выборкой из памяти 512 разрядов каждый (слоги 1П, 2П команды). При приеме информации в регистры АЛУ, возможно программное управление сдвигом информации по кольцу с шагом кратным значению одной или нескольких точек функции. Это обеспечивает обмен информацией между процессами без дополнительной затраты времени. Пример фрагмента функциональной программы приведен на рисунке 1.

```
У[1П, 2П]
1П [Чт, Z, AM₁, + ∞]      (адрес ОП)
2П [Чт, Z, AM₂, + ∞]

У[1А, 2А, К4;  Ц:=К1;  R1=Ц]       ( формирование маски)
1А[А:=1П0;  С:= 2П0; В:=f*(А,С),  Ф=1, М=R1]   ( вычисление функции)
2А[А:=1П3;  С:= 2П3; В:=f*(А,С),  Ф=1, М=R1]

У[1П, 0Б]
1П [Зп, Z, AM₁, +   ∞]      ( запись результата 16 точек)
0Б [1П:=1В0÷2Д3]
```

Рис.1. **Пример программы вычисления функции f * в 16-ти точках.**

Разработана операционная система (ОС), обеспечивающая распределение ресурсов при многозадачном режиме с параллельной обработкой в сочетании с режимом реального времени.

Для программирования пользователю предлагались:

а) язык Автокод-1, компилятор и система отладки. Распараллеливание алгоритма входит в функции компилятора Автокод-1;

б) модификации языков Алгол-60 и Паскаль;

в) язык ОПЕМА для программирования ОС.

Наиболее эффективное программирование осуществлялось на языке Автокод-1.

Заключение

Большой опыт в эксплуатации программного продукта, созданного для комплексов на базе М-10, показал высокую производительность и надежность выбранной архитектуры. Машина хорошо зарекомендовала себя при обработке радиолокационной информации, при фильтрации снимков пузырьковых камер физического эксперимента [6], а также в задачах численного моделирования в физике и геофизике. Опыт программирования на архитектуре функционально-операторного типа в настоящее время позволяет развивать новые численные методы, используя достижения фундаментальной математики.

Список литературы:

1. Карцев М.А. Архитектура цифровых вычислительных машин. М., Наука, 1978.
2. Тихонов А.Н. Численные методы решения некорректных задач. М., Наука, 1990.
3. Гливенко Е.В., Прядко С.А., Фомочкина А.С. Решение систем нелинейных алгебраических уравнений с помощью степени отображения. – «Информационные технологии», 2013, № 7, с. 7-10.
4. Гливенко Е.В. О функциях k-значной логики. //Вопросы радиоэлектроники. – Серия ЭВТ, вып. 9. М., 1970.
5. Саконтикова Н.Н. О полных системах операторов. //Вопросы радиоэлектроники. – Серия ЭВТ, вып.9. М., 1970.
6. Петрова Г.Н. Метод распараллеливания вычислительных алгоритмов применительно к анализу изображений снимков пузырьковых камер. //Вопросы радиоэлектроники. – Серия ЭВТ, вып.8. М., 1974.