АКАДЕМИЯ НАУК СССР
СИБИРСКОЕ ОТДЕЛЕНИЕ
АВТОМЕТРИЯ

№ 5 1991

ПЕРСПЕКТИВНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

УДК 519.68 : 681.32

В. А. Леус, А. И. Мишин
(Новосибирск)

ПАРАЛЛЕЛЬНЫЕ ИТЕРАЦИОННЫЕ ВЫЧИСЛЕНИЯ

Рассмотрен асинхронный подход в итерационной методологии. Представлено сравнительное моделирование параллельных вычислений по стохастическому (асинхронному) и классическому (синхронному) итерационным методам. Даны нижние и верхние оценки временной сложности последовательных и параллельных алгоритмов. Показана несостоятельность устоявшегося представления о якобы имеющихся при распараллеливании преимуществах «хаотических» итераций перед классическими.

Итерационная методика является наиболее универсальным средством приближенных вычислений, в силу чего она интенсивно развивается особенно применительно к многомерным и нелинейным задачам. В [1] содержится подробный хронологически выдержанный обзор итеративных методов. Особое место отведено в ней итерациям вероятностной природы, называемым асинхронными. Они трактуются как не имеющие равных по способности к распараллеливанию, а также по надежности реализации. Не сомневаясь в познавательной ценности «асинхронного» подхода, считаем утверждение о его параллельно-вычислительных преимуществах не соответствующим действительности и намерены показать, что асинхронные методы менее эффективны по сравнению с классическими синхронными.

1. **Современная методика итерационных вычислений.** Классическая простая итерация стала отправным пунктом для целого направления, развивающего идею однотипного циклически повторяющегося вычислительного процесса с отрицательной обратной связью. Так, систему трансцендентных уравнений можно рассматривать как частный случай уравнения

$$\mathbf{x} = F(\mathbf{x}), \qquad (1)$$

где оператор F отображает n-мерное эвклидово пространство E^n в себя.

Все многообразие итерационных методов рождается на этапе перехода от континуального операторного уравнения (1) к дискретному динамическому процессу, который в самом общем случае нестационарной итерации s-го порядка в неявной форме имеет вид:

$$\Phi_{k+1}(\mathbf{x}_{k+1}, \mathbf{x}_k, \mathbf{x}_k - 1, \ldots, \mathbf{x}_{k-s+1}) = 0.$$

Здесь Φ_{k+1} — представитель заданного однопараметрического семейства функций; k — номер итерации; $\mathbf{x}$ есть n-мерный вектор $(x^1, x^2, \ldots, x^n)$, размерность которого зависит от особенностей задачи, например, в конечно-разностных аппроксимациях уравнений математической физики она равна числу узлов пространственной сетки.

Характерной особенностью классических итерационных методов является обязательное обновление всех компонент вектора очередного приближения (итерации). Отказ от этого требования привел к созданию метода последовательно-параллельных хаотических итераций (ППХИ). Разрыв с традицией состоял во введении в рассмотрение так называемой хаотической последовательности $\{J_k\}_{k=1}^{\infty}$ непустых подмножеств множества $\{1, 2, \ldots, n\}$. Последовательность должна иметь максимальный осадок, т. е. каждый из номеров $i = \overline{1, n}$ должен встречаться в ней неограниченное число раз. В явной форме для уравнения (1) метод ППХИ записывается так:

$$x^i_{k+1} = \begin{cases} x^i_k, & \text{если } i \notin J_k; \end{cases}$$

(2) следующим образом:

$$x^i_{k+1} = \begin{cases} x^i_k, & \text{если } i \notin J_k; \\ F^i(x^1_{s1(k)}, \ldots, x^j_{sj(k)}, \ldots, x^n_{sn(k)}), & \text{если } i \in J_k. \end{cases} \tag{3}$$

Нижний индекс j-го аргумента вычисляемой функции F^i есть величина $s_j(k)$, называемая запаздыванием, которая зависит от дискретного момента k и номера компоненты очередного приближения. Запаздывание подчиняется условиям

$$s_j(k) < k, \; s_j(k) \to \infty \quad \text{для всех} \quad j = \overline{1, n}.$$

Однако метод асинхронных итераций отнюдь не является методом s-го порядка. Для методов s-го порядка обязательна строгая привязка компонент всех используемых предшествующих векторов по аргументам (sn)-местной вычисляемой функции Φ_{k+1}, тогда как в методе асинхронных итераций это распределение чисто случайное.

В [1] предложены методы, развивающие идею асинхронных итераций. Асинхронный итерационный метод сквозного счета применяется для решения задачи с цепочкой операторных уравнений, аппроксимирующей континуальное уравнение

$$\frac{\partial}{\partial t}(\mathbf{x}) = \Psi(\mathbf{x}, t), \quad \mathbf{x} \in X.$$

Здесь $\Psi : X \to X$ — оператор в банаховом пространстве, а параметр t имеет смысл физического времени. В результате дискретизации по $t = 0, 1, \ldots, T-1, T$ получается цепочка операторных уравнений

$$\begin{aligned} \mathbf{x} &= \Psi(\mathbf{x}, 0) + \mathbf{x}_0, \\ \mathbf{x} &= \Psi(\mathbf{x}, 1) + \mathbf{x}_1, \\ &\cdots \\ \mathbf{x} &= \Psi(\mathbf{x}, T-1) + \mathbf{x}_{T-1}. \end{aligned} \tag{4}$$

Исходя из известного начального вектора $\mathbf{x}_0$, требуется найти вектор $\mathbf{x}_T$ для конечного момента $t = T$. При заданной точности ε и хаотической последовательности $\{J_k\}$ в [1] предлагается искать решение

итеративным процессом, который для i-й компоненты записывается в виде

$$x^i(t^i_{k+1}) = \begin{cases} x^i(t^i_k) & \text{при} \quad i \notin J_k; \\ \Psi^i(x^1(t^1_k), \ldots, x^n(t^n_k), t^i_k) + x^i(t^i_{k-\kappa^i}) & \text{при} \quad i \in J_k. \end{cases} \tag{5}$$

По каждой компоненте предполагается запущенным свой ход времени, так что физические моменты оказываются «завязанными» сложной зависимостью с дискретными итерационными благодаря следующим условиям:

$$t^i_{k+1} = \begin{cases} t^i_k, & \text{если} \quad i \notin J_k; \\ t^i_k, & \text{если} \quad i \in J_k \quad \text{и} \quad \|x^i(t^i_k) - x^i(t^i_{k-1})\| > \varepsilon; \\ & \end{cases} \tag{6}$$

в которые для x^i достигалась заданная точность ε.

Для решения конечно-разностных аналогов задач математической физики, вместо одной фиксированной сетки, давно применяют иерархию взаимно вложенных сеток прогрессивно укрупняющегося шага. В [1] исследован вопрос о применении асинхронного мультисеточного метода к отысканию итерациями решения уравнения

$$x = \Psi(x) + \varphi \tag{7}$$

с оператором $\Psi : X \to X$ и заданным элементом $\varphi \in X$.

Пусть имеется иерархия сеток ω_q и соответственно совокупность операторных уравнений $x = \Psi_q(x) + \varphi_q$, $q = \overline{1, Q}$, каждое из которых аппроксимирует континуальное уравнение (7) в соответствующем пространстве сеточных функций H_q. Согласно асинхронному мультисеточному методу, последовательность приближений строится по правилу

$$x^i_{k+1} = \begin{cases} x^i_k & \text{при} \quad i \notin J_k; \\ \Psi_{q^i(k)}(x_{k-s(k)}) + \varphi^i_{q^i(k)} & \text{при} \quad i \in J_k. \end{cases} \tag{8}$$

Здесь $q^i(k)$ — номер сетки, $s^i(k)$ — величина запаздывания, используемые для данной компоненты в текущей итерации.

Как показано в [1], для хаотической последовательности $\{J_k\}$ с максимальным осадком и определенных условий, наложенных на совокупность операторов $\{\Psi_q\}$, процесс (8) условно сходится, так что погрешность стремится к пределу, не превосходящему заданной положительной величины.

2. Моделирование параллельных итерационных вычислений. Вычленение в задаче отдельных ветвей, выполнение которых можно поручить разным вычислительным машинам, — это только начало распараллеливания. Главное заключается в организации совместной работы машин, что и определяет в итоге эффективность распараллеливания.

Решение операторного уравнения типа (7) итерационным методом на ЭВМ непременно предполагает конечномерность оператора. Если данный оператор не таков, то он аппроксимируется оператором F, действующим в n-мерном банаховом или гильбертовом пространстве, представимом в виде конечного произведения элементарных подпространств, с векторной нормой $\|\vec{\cdot}\| \cdot \|\vec{\cdot}\|$ в нем. Используется обобщенное понятие сжимаемости

оператора — покоординатная сжимаемость, или p–сжимаемость. Условие p–сжимаемости имеет вид

$$\vec{\|}F(\mathrm{x}_1) - F(\mathrm{x}_2)\vec{\|} \le L(\vec{\|}\mathrm{x}_1 - \mathrm{x}_2\vec{\|}), \tag{9}$$

где L — так называемая матрица сжатия со спектральным радиусом $\rho(L)$, меньшим единицы.

Численный процесс здесь, как и в случае решения системы трансцендентных уравнений, естественным образом подразделяется на координатные ветви, которые могут выполняться параллельно коллективом вычислителей, содержащих каждый процессор и память. Осуществляются n циклических процессов, взаимодействующих по схеме полного графа так, что результат каждого используется всеми остальными (рис. 1).

Рассмотрим покоординатную реализацию итерационного процесса на коллективе вычислителей, называемом n-процессорной вычислительной системой (ВС), кольцевой структуры (рис. 2). Распределим работу таким образом, что вычисление i-й компоненты $x^i_{k+1} = F^i(x_k)$ вектора очередного приближения осуществляется процессором i-го вычислителя системы.

Пусть обмен между процессорами в ВС ведется словами, каждое из которых содержит одну компоненту вектора-приближения и имеет два сигнальных разряда — метки μ_1 и μ_2. Достаточна циркуляция данных по кольцу процессоров в одном направлении, для определенности — в сторону возрастания номеров i. Такая циркуляция может быть осуществлена, например, в двухфазном режиме: первая фаза — все нечетные процессоры передают, все четные — принимают; вторая фаза — все четные процессоры передают, все нечетные — принимают. Если n нечетно, то в кольцо вводится $(n + 1)$-й элемент, который не вычисляет, а лишь транслирует данные от младшего соседа к старшему. Первый процессор в системе является ведущим, через него производится обмен с внешней средой и общее управление ходом вычислений.

В исходном состоянии в систему загружен вектор нулевого приближения x_0 так, что в памяти каждого i-го вычислителя находится одна компонента. Каждый процессор передает своему старшему соседу дубликат имеющейся у него компоненты с нулевыми значениями обеих меток, принимает от младшего соседа и запоминает следующую компоненту. Затем он передаст дальше дубликат запомненной компоненты и принимает следующую и так далее n раз, после чего в каждый процессор вернется его собственная компонента x^i_0. Тем самым в память каждого i-го вычислителя будут занесены все аргументы функции F^i и его процессор одновременно с другими станет вычислять i-ю компоненту $x^i_1 = F^i(x^1_0, x^2_0, \ldots, x^n_0)$ вектора x_1, невязку $e^i = |x^i_1 - x^i_0|$, которую сравнит с заданной точностью ε. Если $e^i > \varepsilon$, разряд-метка μ_1 сохраняется в нулевом состоянии; если $e^i \le \varepsilon$, присваивается значение $\mu_1 = 1$. Затем в системе производится очередной оборот по кольцу уже вектора первого приближения и вычисление компонент вектора второго приближения и т. д. Так циклически возобновляется процедура вычисления вектора $(k + 1)$-го приближения через переданные по кольцу компоненты k-го приближения. Ведущий процессор анализирует на каждом обороте метки

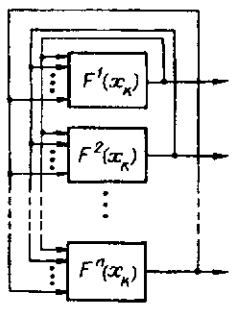

Рис. 1

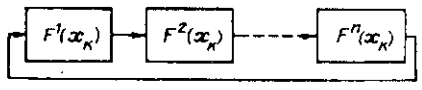

Рис. 2

всех проходящих слов. Как только все n меток приобретут на k-м обороте единичные значения, требуемая точность достигнута. Тогда ведущий процессор посылает по кольцу слово с меткой $\mu_2 = 1$, являющейся для остальных процессоров командой окончания вычислений и перехода к выдаче результата во внешнюю среду.

Время T_n^c, затрачиваемое n вычислителями системы на выполнение K итераций в синхронном параллельном процессе, складывается из времени собственно вычислений $T_в$ и времени обменов $T_о$:

$$T_n^c = K(T_в + T_о). \tag{10}$$

Время $T_о$ зависит от типа ВС (асинхронная, асинхронно-локальная, синхронная), а время $T_в$, кроме того, — и от типа операции (локальная или глобальная). Длительность глобальной, т. е. существенно скалярной, операции τ_n' пропорциональна линейному размеру вычислителя, а длительность локальной операции τ пропорциональна лишь линейному размеру процессора (без памяти). В асинхронной ВС, где в качестве вычислителей используются традиционные ЭВМ (машины с памятью произвольного доступа), время $T_о$ оборота всех n слов по кольцу из n вычислителей составляет величину $n\tau_n'$. Для асинхронно-локальной ВС время $T_о$ достигает минимально возможной величины $n\tau$ [2].

Таким образом, собственно вычислительное время $T_в$ при выполнении одной итерации (время вычисления значения n-местной функции F^i) в рассмотренном синхронном параллельном процессе определяется неравенствами

$$dn\tau \le T_в \le dn\tau_n',$$

где d — среднее число операций на один аргумент. Время межпроцессорных обменов $T_о$, необходимое для доставки компонент предшествующей итерации, определяется неравенствами $n\tau \le T_о \le n\tau_n'$. Отсюда имеем оценку

$$K(d+1)n\tau \le T_n^c \le (d+1)n\tau_n'. \tag{11}$$

Время T_n^c в случае векторизуемых вычислений равно $2Kdn\tau$ [2, 3].

Одиночный вычислитель (однопроцессорная машина) на каждой из итераций синхронного алгоритма вычисляет n значений функций F^i от n аргументов. Если структура функции F^i позволяет конвейеризовать процесс доставки (подстановки) аргументов, как это имеет место для векторизуемых вычислений, то на асинхронно-локальной однопроцессорной машине достигается минимально возможное время итерации, равное $dn^2\tau$. В случае скалярных вычислений, когда доставка аргументов функции $F^i(x)$ вообще не поддается конвейеризации, на одну итерацию затрачивается максимальное время $dn^2\tau_1'$. Время однопроцессорной реализации всех K итераций оценивается, таким образом, неравенствами

$$Kdn^2\tau \le T_1^c \le Kdn^2\tau_1'. \tag{12}$$

Из сравнения (11) и (12) получим ускорение параллельных вычислений по отношению к последовательным. Для асинхронно-локальных машин на конвейеризуемых задачах ускорение равно $dn/(d+1) \approx n$. По отношению к традиционной ЭВМ с памятью прямого доступа на асинхронно-локальной ВС может быть достигнуто максимальное ускорение $Kdn^2\tau_1'/K(d+1)n\tau \approx n\tau_1'/\tau$. При r-мерной компоновке

памяти (r = 1, 2, 3) имеем $\tau_1' \sim (n^2 \log n)^{1/r}$, $\tau_n' d \sim (n \log n)^{1/r}$, а ускорение в двумерном случае может быть порядка n^2. Асинхронная ВС из вычислителей с памятью прямого доступа по отношению к одиночному вычислителю дает ускорение $Kdn^2\tau_1'/K(d+1)n\tau_n' \sim n^{1+1/r}$.

3. Сравнение параллельных реализаций итерационных методов.

1. Во всех «хаотических» итерационных методах фундаментальную роль играет последовательность $\{J_k\}_{k=1}^{\infty}$, определяющая процедуру вычисления той или иной компоненты вектора очередного приближения. Согласно [1] для сходимости итераций нужно, чтобы хаотическая последовательность имела максимальный осадок, но это условие не является достаточным. Действительно, рассмотрим хаотическую последовательность, составленную всего из двух типов подмножеств $\{1\}$ и $\{2, 3, \ldots, n\}$. Пусть подмножество второго типа встречается в тех и только тех элементах J_k, где k есть l-я степень (l — любое целое положительное) натурального числа. Хотя в такой последовательности каждый номер компоненты встречается сколь угодно много раз (т. е. осадок максимальный), но при произвольном конечном числе итераций надлежащим образом выбранное l абсолютно исключит сходимость. Регулярность взятой лишь для простоты рассуждений последовательности несущественна, так как всегда можно потребовать, чтобы пропорция между подмножествами первого и второго типа в хаотической последовательности соблюдалась только в среднем.

Из естественного априорного постулата о равноправии всех компонент вектора приближения вытекает, что вероятность p_i встретить номер i в элементе J_k не должна зависеть от i, т. е. $p_i = p(k)$, причем $(1/n) \leq p(k) \leq 1$. Только в условиях равномерного распределения вероятности по всем компонентам будет гарантирована сходимость итерационного «хаотического» метода для сжимающих операторов.

При параллельной реализации классического итерационного метода каждый i-й процессор системы для вычисления $(k+1)$-го приближения должен получить через своих соседей все (кроме i-й) компоненты предыдущего приближения. Таким образом, в каждом итерационном цикле реализуется полная схема межпроцессорных обменов.

В методе ППХИ (2) при получении некоторых компонент очередного приближения допускается простое повторение предшествующего значения соответствующей компоненты без вычисления n-местной функции $F^i(\mathrm{x}_k)$. В методе асинхронных итераций (3), благодаря запаздывающим аргументам, допускается использование более старых, чем предыдущие, компонент также и при получении нового значения функции $F^i(\mathrm{x})$. Подобные формальные упрощения композиционно-подстановочных связей между вычисляемыми функциями в «хаотических» методах не позволяют тем не менее использовать более простую по сравнению с полной схему межпроцессорных обменов. В самом деле, конкретная структура хаотической последовательности $\{J_k\}_{k=1}^{\infty}$ априорно неизвестна, поэтому невозможно заранее фиксировать подмножество «нужных» связей, отбросив все остальные. Конечно, элементы J_k, порождаемые в ходе вычислительной работы, образуют серию ограниченных выборок, но непременно из полного ансамбля исходов. В силу этого каждому из процессоров должны доставляться результаты работы всех других, а это возможно реализовать лишь при полной схеме межпроцессорных обменов.

Как видим, в параллельной реализации итерационного метода объем обрабатываемых данных пропорционален n^2, тогда как процессоров только n. При этом каждый процессор на одной итерации выполняет

вычислительную работу объемом $\sim n$ операций за время того же порядка, что и время однократной реализации полной схемы межпроцессорных обменов. Следовательно, использование любых архитектур ВС любой размерности и с произвольным числом процессоров не позволит сделать временную сложность меньше определенного минимума, необходимого для реализации полной схемы обменов. Из всего многообразия мыслимых структур ВС особое значение имеют простейшие одномерные структуры (линейная и кольцевая) с числом процессоров порядка корня квадратного из объема данных (общего числа элементов системы). Именно такие структуры позволяют достичь минимума временной сложности не только итерационных алгоритмов, но и многих других, где время, затрачиваемое одним процессором на вычисления между двумя последовательными обменами, имеет тот же порядок, что и время однократной параллельной реализации полной схемы межпроцессорных обменов. Для некоторых алгоритмов, где собственно вычислительное время существенно больше $n\tau$ по порядку величины (например, τn^{α}, $\alpha > 1$), возможно дальнейшее распараллеливание работы процессора, что позволяет снизить временную сложность таких алгоритмов.

Рассмотрим алгоритм метода асинхронных итераций (3) в постановке на кольцевой ВС, где каждый i-й процессор вычисляет i-ю компоненту x_{k+1}^i вектора очередного приближения (рис. 3). В отличие от описанного выше синхронного варианта будем предполагать, что каждая элементарная машина разбита на трансляционную (прямоугольник) и вычислительную (квадрат) части. Транслятор Т занят приемом-передачей слов по кольцу. Вычислитель проводит собственно вычисления, иногда прерывая работу транслятора и пропуская поток слов через себя, чтобы выловить нужные ему для вычисления в очередной итерации компоненты, поглотить ранее испущенную им компоненту и запустить по кольцу новую, только что им вычисленную. Кроме признаков μ_1 и μ_2, здесь каждое слово циркулирующих данных несет в себе номер компоненты i (или, что то же самое, номер породившего процессора).

Предполагается, что каждый процессор имеет встроенный датчик псевдослучайных чисел с равномерным распределением вероятности, с помощью которого генерируется элемент хаотической последовательности J_k. Вычислитель проверяет принадлежность своего номера i к J_k. Если $i \notin J_k$, то новому x_{k+1}^i присваивается сохраняемое предыдущее значение x_k^i. Если $i \in J_k$, то вычислитель, прервав транслятор, выбирает из потока аргументы функции F^i. Затем он переключает поток слов на транслятор и приступает к вычислению величины $x_{k+1}^i = F^i(x_{s_1(i)}^1, \ldots, x_{s_n(i)}^n)$. Как только x_{k+1}^i найдена, вычислитель снова включается в поток слов, поглощает старое свое слово, запускает новое и выбирает из потока при необходимости (в зависимости от принадлежности i очередному J_{k+1}) нужные аргументы для вычисления F^i и т. д.

Оценим время, затрачиваемое вычислителем на получение одного значения компоненты x_{k+1}^i. В случае векторизуемых вычислений время счета значения функции F^i от n аргументов пропорционально n, а в случае скалярных вычислений оно существенно больше. В тех случаях, когда без вычисления F^i полагается $x_{k+1}^i = x_k^i$, все равно используется результат J_k работы генератора хаотической последовательности. Длина двоичного представления номера i равна $\log_2 n$, а средняя мощность выдаваемого генератором подмножества при равномерном распределении вероят-

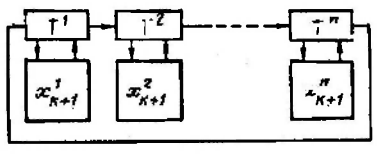

Рис. 3

ности есть $n/2$, так что время порождения одного элемента J_k пропорционально $n\log n$. Таковы же по порядку величины и средняя длительность распознавания принадлежности номера процессора элементу хаотической последовательности, и длительность идентификации n слов, взятых вычислителем из потока для подстановки в аргументы функции F^i, и время оборота слова по всему кольцу.

С другой стороны, при параллельной организации асинхронных итераций собственно вычисления идут на фоне трансляции данных по процессорному кольцу, так что время обмена T_o здесь совмещается с временем $T_в$. Вследствие необходимости идентификации компонент вектора-приближения, генерирования хаотической последовательности и распознавания принадлежности номера компоненты элементу последовательности оценка временной сложности одной асинхронной итерации возрастает на величину $\delta n\log_2 n$, где δ — усредненные временные затраты на обработку одного бита информации. Оценка временной сложности K итераций асинхронного алгоритма имеет вид

$$Kn(d\tau + \delta\log_2 n) \le T_n^a \le Kn(d\tau_n' + \delta\log_2 n). \qquad (13)$$

Отмеченные вынужденные издержки реализации асинхронного метода не только увеличивают время выполнения численных итераций, но и существенно усложняют ВС конструктивно. Здесь приходится снабжать каждый процессор генератором хаотической последовательности с равномерным распределением вероятности по всем номерам компонент вектора-приближения.

Сравнение (13) с (11) не в пользу метода асинхронных (стохастических) итераций. Его параллельная реализация сопряжена с увеличением машинного цикла, а на конвейеризуемых (векторизуемых) вычислениях неизбежен проигрыш во времени по сравнению с методом синхронных (классических) итераций в $\sim\log n$ раз из-за необходимости идентификации данных. Единственный шанс экономии времени счета для асинхронного метода — это сокращение числа итераций, требующихся для достижения заданной точности, по отношению к классическому синхронному.

Из условия (9) следует покоординатная оценка

$$\|F(\mathbf{x}_1) - F(\mathbf{x}_2)\| \le \rho\|\mathbf{x}_1 - \mathbf{x}_2\|,$$

где $\|\cdot\|$ имеет смысл любой компоненты векторной нормы $\vec{\|}\cdot\vec{\|}$. Пусть $\mathbf{x}_\infty$ — решение операторного уравнения, т. е. $x_\infty = F(\mathbf{x}_\infty)$. На k-й итерации, согласно покоординатной оценке, имеем

$$\|\mathbf{x}_\infty - F(\mathbf{x}_{k-1})\| = \|\mathbf{x}_\infty - \mathbf{x}_k\| = \varepsilon_k \le \rho\|\mathbf{x}_\infty - \mathbf{x}_{k-1}\| \le$$

$$\le \rho^2\|\mathbf{x}_\infty - \mathbf{x}_{k-2}\| \le \ldots \le \rho^k\|\mathbf{x}_\infty - \mathbf{x}_0\|.$$

Количество шагов K, нужное для достижения заданной точности, оценивается из неравенства $\varepsilon_k \le \rho^k\|\mathbf{x}_\infty - \mathbf{x}_0\| = \rho^k\varepsilon_0$, а именно $K \ge \log(\varepsilon_k/\varepsilon_0)\,(\log\rho)^{-1}$.

В [1] качественно доказывается неравенство $\rho_a < \rho_c$ для спектральных радиусов в асинхронном и синхронном методах. Однако различие между ρ_a и ρ_c если и имеется, то несущественное, и оно не может улучшить оценку (13) в сравнении с оценкой (11). Следовательно, преимущество классического синхронного метода итераций несомненно.

2. Коснемся вкратце такой важной области применения итерационных методов, как численное решение нестационарных граничных задач для уравнений в частных производных.

К цепочке операторных уравнений вида (4) сводится, в частности, конечно-разностный вариант граничной задачи для уравнения в частных производных параболического типа. Согласно классическому методу, итерации ведутся по всем n узлам пространственной сетки на t-м временном срезе до тех пор, пока заданное ε не будет достигнуто на всех компонентах некоторого приближения, после чего только и производится переход к следующему временному срезу. Именно поэтому здесь происходит моделирование перемещения плоского фронта решения во времени, т. е. всего переходного процесса от начального момента $t = 0$ до конечного $t = T$.

Асинхронный метод сквозного счета (5) допускает и отставание, и убегание вперед по отдельным компонентам, однако время получения конечного решения при $t = T$ определяется все-таки темпом достижения заданного ε на самых медленно сходящихся компонентах. Этот темп, как показано на примере кольцевых ВС, не может превосходить темпа классических синхронных итераций.

К тому же в асинхронном варианте сквозного счета из-за «отставания-убегания» фронт решения искривляется и размывается, маскируя тем самым переходный процесс. Чтобы узнать решение на каком-либо промежуточном временном срезе $T_1 < T$, нужно вначале заменить последнее из условий (6) на неравенство $t_k^i \le T_1$ и только по достижении на всех компонентах временного среза $t = T_1$ восстановить прежнее неравенство. Для получения всей картины перехода пришлось бы повторить эту процедуру последовательно на каждом временном срезе, т. е. неминуемо вернуться к синхронному методу, но отягченному всеми недостатками асинхронного.

3. Завершим сравнительный анализ традиционной и асинхронной методик рассмотрением конечно-разностной задачи с иерархией из Q взаимно вложенных сеток. В отличие от рассмотренных выше задач мультисеточные методы применяют к конечно-разностным аппроксимациям уравнений математической физики, характерной особенностью которых являются ленточные матрицы. Это ограничивает число аргументов функции Ψ^i некоторой константой, определяемой принятым шаблоном дискретизации.

Реализация классического мультисеточного метода на n-процессорной ВС состоит в следующем. Задаются убывающие положительные числа $\varepsilon_1 > \varepsilon_2 > \ldots > \varepsilon_Q$ по числу измельчающихся сеток иерархии. Итерации выполняются сначала на самой крупной сетке ω_1 с самой грубой точностью ε_1, а затем полученное в первом приближении решение последовательно уточняется итерациями по все более мелкоячеистым сеткам до получения требуемой точности ε_Q. При реализации мультисеточного метода синхронными итерациями на асинхронно-локальной ВС время решения $T_n^{\text{мс}}$ оценивается неравенствами $Q(d+1)n\tau \le T_n^{\text{мс}} \le QK(d+1)n\tau$, если на каждой из сеток выполняется не более K итераций.

Понятие степени локальности итерационного метода авторы монографии [1] ввели как минимальное число компонент предыдущей итерации, используемых для вычисления одной компоненты текущей итерации (вернее было бы назвать это число степенью глобальности метода). Итерации с ленточными матрицами характеризуются тем, что в вычислении текущей компоненты участвует небольшое количество предыдущих компонент. Однако это формальное определение совершенно не учитывает пространственного размещения процессоров. Если даже решетка процессоров топологически соответствует самой мелкоячеистой сетке конечно-разностной аппроксимации, то локальные, в смысле [1], узлы сетки ω_{Q-1}, соседствующие с некоторым фиксированным узлом, занимают в ВС процессоры, образующие окрестность удвоенного диаметра, а соседствующие с тем же узлом узлы сетки ω_{Q-2} — окрестность

учетверенного диаметра, и т. д. Диаметр окрестности, соответствующей «локальным» узлам сетки ω_1, превышает первоначальный в 2^{Q-1} раз, и соответственно возрастает время доставки данных к процессору, ведущему итерации по грубой сетке, независимо от того, синхронные или асинхронные итерации применяются в данном случае.

Заключение. Сравнение параллельных итерационных вычислений по классическому (синхронному) и хаотическому (асинхронному) методам приводит к следующим выводам. Содержащееся в [1] утверждение о том, что распараллеливание классических алгоритмов из-за коммуникационных затрат имеет предел, начиная с которого увеличение числа параллельно работающих процессоров ВС не дает (в отличие от асинхронных алгоритмов) роста производительности, несостоятельно. Еще в 70-х годах в [3] было показано, что доставка компонент вектора предыдущей итерации к процессорам кольцевой ВС может быть осуществлена в конвейерном режиме, и это позволяет достичь минимально возможного времени реализации итерационного алгоритма.

К недостаткам асинхронных вычислительных методов итераций следует отнести: 1) необходимость индивидуальной привязки компоненты вектора-приближения к вычисляющему процессору для идентификации ее другими процессорами даже в случае конвейеризуемых вычислений; 2) обязательное присутствие в каждом вычислителе генератора хаотической последовательности, использующего датчик псевдослучайных чисел с равномерным распределением вероятности.

Указанные недостатки асинхронно-хаотического подхода в итерационных методах не только усложняют вычислительную систему, но и приводят на конвейеризуемых вычислениях к замедлению вычислительного процесса в $\sim \log n$ раз по сравнению с синхронными классическими итерациями. Высказанное в [1] мнение о якобы высокой статистической надежности хаотических вычислений неверно, так как процессор с отказавшим генератором хаотической последовательности может, например, бесконтрольно выдавать в каждом итеративном цикле одно и то же число, чем будет нарушено равноправие всех компонент и, следовательно, условие сходимости итераций к решению.

Таким образом, асинхронные итерационные методы, основанные на вероятностном процессе получения компонент вектора-приближения, не имеют никаких преимуществ ни по глубине распараллеливания, а значит и производительности, ни по надежности перед классическими синхронными итерациями.

СПИСОК ЛИТЕРАТУРЫ

1. **Нестеренко Б. Б., Марчук В. А.** Основы асинхронных методов параллельных вычислений.—Киев: Наук. думка, 1989.
2. **Мишин А. И.** Параллельные вычислительные среды с локальными взаимодействиями элементов // АиТ.—1982.—№ 10.
3. **Мишин А. И., Седухин С. Г.** Вычислительные системы и параллельные вычисления с локальными взаимодействиями // Математическое обеспечение ВС (Вычислительные системы. Вып. 78).—Новосибирск: ИМ СО АН СССР, 1979.

Поступила в редакцию 21 февраля 1991 г.