

ВОПРОСЫ МОДУЛЬНОГО АНАЛИЗА И ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ В ЗАДАЧАХ МАТЕМАТИЧЕСКОЙ ФИЗИКИ*

К настоящему времени при решении на ЭВМ задач математической физики сложилась вполне определенная технологическая цепочка [1]: объект исследования → математическая модель → физическая модель → численный алгоритм → программа → расчет на ЭВМ. Объектом математической технологии здесь является вычислительная часть этой цепочки: математическая модель → численный алгоритм → программа → расчет на ЭВМ, которую в дальнейшем будем называть просто вычислительной цепочкой. Структура ее такова, что допускается цикличность как для всей цепочки в целом, так и между ее отдельными звеньями, т.е. является нелинейной.

В силу этого основная задача математической технологии - оптимизация вычислительной цепочки - должна рассматриваться как задача глобальной оптимизации. Примером существенной нелинейной цепочки может служить влияние на ее структуру применения многопроцессорных ЭВМ. Действительно, их специфика приводит не только к определенному пересмотру численных алгоритмов, но и вносит в вычислительную цепочку новое звено - структуру ЭВМ, под которой здесь следует понимать совокупность технических и системных средств, допускающих взаимодействие с остальными звеньями цепочки. Тогда вычислительная цепочка приобретает вид: математическая модель → численный алгоритм → программа → структура ЭВМ → расчет. Практика вычислений позволяет утверждать, что структура алгоритма и программы выдвигает ряд требований, предъявляемых к структуре ЭВМ. Как следствие, может оказаться целесообразным создание специализированных вычислительных комплексов для решения задач математической физики. Так, например, в настоящее время уже назрела проблема создания процессоров и соответствующих структур оперативной и внешней памяти, специально приспособленных для работы с большими разреженными матрицами, столь часто встречающихся в задачах математической физики, поскольку эффективность здесь универсальных ЭВМ крайне низка.

Появление многопроцессорных ЭВМ позволяет решить очень важную проблему - проблему сокращения существующего разрыва между требованиями по быстродействию, предъявляемыми к ЭВМ этими задачами, и техническими возможностями существующих (однопроцессорных) ЭВМ. Так, по оценкам американских специалистов (см. также [2]) для решения задач пространственного обтекания тел в широком диапазоне изменения числа Маха набегающего потока для детального учета структуры течения (ламинарный, переходный, турбулентный участки) требуется ЭВМ с быстродействием порядка 10^{10} оп/с. Очевидно, что вплотную подойти к решению таких задач можно только с помощью ЭВМ параллельного действия, имеющих достаточно большое число процессоров. В связи с этим вопрос организации параллельных вычислений, затрагивающий все звенья вычислительной цепочки, приобретает сейчас принципиальное значение.

Построение пакета прикладных программ для класса задач математической физики начинается с модульного анализа вычислительного алгоритма [1,3 - 7]. Если в основу алгоритма положены методы расщепления по физическим процессам и пространственным переменным [8], то решение исходной задачи из рассматриваемого класса сводится к последовательному решению более простых задач [3]. Рассмотрение и учет условий, накладываемых на программную реализацию таких задач конкретной структурой ЭВМ, проведенное в работе [9] для задач аэродинамики, привело к формированию понятия "простая задача". Она была определена как краевая задача с однородной физико-математической моделью, однородной разностной сеткой и целиком вкладываемая в однородную (оперативную) память ЭВМ. Проводя сегментацию области решения исходной задачи на ряд подобластей, в каждой из которых определена простая задача, мы решаем одновременно проблему перераспределения памяти ЭВМ между ее неоднородными компонентами и получаем возможность организации параллельного решения задачи в целом. Для стыковки решения на сложных границах подобластей используется введенное в [7] понятие обменных краевых условий. Еще одно преимущество такого подхода заключается в том, что в рамках каждой простой задачи существенно облегчается задача построения хорошо обусловленной сетки, правда, появляется задача стыковки решения на разных сетках, но это уже отдельный, впрочем вполне разрешимый, вопрос.

Определенная выше простая задача может рассматриваться как определение модуля в смысле [3], сведение же конкретной задачи к последовательности таких "простых задач" является модульной декомпозицией задачи. Дать определение "простой задачи" вообще (как, впрочем, и модуля) пока не представляется возможным и поэтому мы попытаемся указать лишь общий способ ее определения для различных классов задач математической физики. Опыт решения больших задач показывает, что конечной устойчивостью (устойчивость при конкретных расчетах, в отличие от теоретической) чаще всего обладают программы, реализующие однородные алгоритмы и математи-

ческие модели в однородной информационной среде. Тогда наложением ряда независимых требований по однородности на рассматриваемый класс задач (каких именно определяется как самим классом задач, так и методами их решения и способами реализации) можно определить "простую задачу". Все конкретные задачи из этого класса будут представлены конкретной же совокупностью этих "простых задач", определенная последовательность решения которых и даст решение исходной задачи. Так, для задач механики сплошной среды "простая задача" должна удовлетворять следующим требованиям.

1. Однородность математической модели. Под этим требованием понимается единство структуры математического описания физической модели исследуемого явления. Если, например, для описания используется система дифференциальных уравнений, то она не должна менять своей структуры, т.е. набора определяющих членов в области определения "простой задачи". Тип ее может и меняться на решении или от решения к решению, лишь бы алгоритм решения был единым. Примером здесь может служить применение метода установления для решения стационарных уравнений Эйлера в газовой динамике. Однородность математической модели позволяет применять апробированные, конечно-устойчивые алгоритмы, существенно упрощает тестирование задачи, в целом. Практически однородность модели достигается чаще всего применением метода расщепления по физическим процессам.

2. Однородность постановки задачи. Включает в себя: а) однородность математической модели, б) алгоритмическую однородность начальных и краевых условий. Последнее означает, что заданный набор краевых условий не приводит к изменению алгоритма решения задачи. Сам этот набор, естественно, зависит от алгоритма. Например, для уравнения Лапласа $\Delta u = 0$ в случае задач Дирихле и Неймана алгоритмы решения различны. Тогда в случае разнородных условий, когда на одной части границы области (γ) заданы условия $u|_{\gamma_1} = a(\gamma_1)$, а на другой $(\partial u/\partial n)|_{\gamma_2} = b(\gamma_2)$, требованиям однородности можно удовлетворить, если применять алгоритм, допускающий единую реализацию условий вида $a(\gamma)u + \beta(\gamma)\partial u/\partial n = g(\gamma)$, где $a(\gamma)$, $\beta(\gamma)$ могут обращаться в нуль на γ или любой ее части.

3. Однородность алгоритма. Здесь наряду с обычными, традиционными требованиями однородности, предъявляемыми к алгоритмам, само понятие однородности трактуется в более широком аспекте. Прежде всего сюда включено требование модельности алгоритма. Модельность алгоритма есть требование аппроксимации корректно поставленной континуальной задачи. Это означает, что алгоритм не должен содержать наборов различного рода малых параметров, не связанных с существом задачи и с дискретизацией континуальной модели, влияние которых на само решение трудно контролировать. Так, например, для нелинейных неявных разностных схем использование различного рода итераций по нелинейности может привести к потере модельности. В целом этот вопрос в настоящее время почти не исследован и конкретных рекомендаций практически нет.

Другим важным требованием, связанным с модельностью и применением методов аналитического расщепления алгоритма, является требование одноступенчатости алгоритма. Это означает, что сам алгоритм решения "простой задачи" должен представлять собой последовательность простых, одноступенчатых алгоритмов. В качестве примера можно указать на двухшаговые схемы типа Рунге-Кутты и ряд других. Такой одноступенчатый алгоритм уже не будет содержать, например, итераций по нелинейности. По отношению к нему они оказываются внешними, увеличивая лишь пошаговую дробность всего алгоритма в целом.

Расчетная схема тогда будет следующей. В каждой "простой задаче" одноступенчатый алгоритм продвигает решение на один шаг. Далее, если нужно, рассчитываются краевые условия и осуществляется стыковка решений на границах областей определения "простых задач", и процесс повторяется.

Таким образом, алгоритм решения "простой задачи" может быть только алгоритмом одного глобального шага.

4. Однородность информационной среды. Под информационной средой в задачах математической физики здесь понимается дискретное представление области определения задачи вместе с формальной структурой аналитического представления решения (способ дискретизации, шаблон, аппроксимация на этом шаблоне и т.д.). Информационная среда неотделима от метода решения и определяется самим методом. Так, в случае применения конечно-разностных методов это будет разностная сетка, для конечного элемента - упорядоченное множество, определяемое способом триангуляции области решения и т.д. Требование однородности информационной среды заключается в том, чтобы в пределах "простой задачи" она не вносила элементов неоднородности в алгоритм решения. Например, это означает, что разностная сетка должна быть регулярной, т.е. переход от обработки одного ее узла к другому не должен приводить к перестройке алгоритма решения. Стыковка различных сеток, не согласованных между собой, должна осуществляться только на границах областей решения "простых задач".

Аналитическое представление решения также должно быть однородным, т.е. должно

быть единство способа и порядка аппроксимации решения при переходе от континуального его представления к дискретному.

5. Однородность памяти. Это есть требование вложения простой задачи в оперативную память ЭВМ. Тогда вопрос о перераспределении памяти ЭВМ между внешней и оперативной для всей задачи в целом решается путем сегментации области решения на подобласти таким образом, чтобы каждая из них была областью определения "простой задачи". Для стыковки решения между ними на границах этих подобластей используются обменные краевые условия [7].

6. Хорошая обусловленность. Под этим требованием понимается, кроме обычного требования хорошей обусловленности краевой задачи, еще и требование определенности областей изменения параметров однородного алгоритма, при которых гарантируется его надежная реализация на ЭВМ.

7. Функциональная и алгоритмическая полнота. Это требование означает, что алгоритм, реализующий простую задачу, должен быть достаточно универсальным и должен включать в себя максимально возможное число произвольных параметров и функций, которые соответствуют свободным параметрам и функциям в исходной системе уравнений, начальных и краевых условиях, описаниях области решения. В частности, алгоритм должен быть рассчитан на его применение в достаточно произвольных областях и допускать достаточно произвольную сетку, основным требованием, предъявляемым к которой, является требование ее реальной невырожденности.

8. Автономность, неавтономность. "Простая задача" называется автономной, если ее область определения совпадает с областью решения исходной задачи. Краевые условия для нее заданы, а ее решение не зависит от решения других "простых задач". Для больших задач математической физики эта ситуация является исключительной.

Как правило, область решения исходной задачи требуется сегментировать (см. п. 4), и уже тогда для каждой "простой задачи", определенной на сегменте, на каждой определяющей границе должны быть заданы краевые условия, обеспечивающие стыковку решений этих "простых задач". Такие краевые условия называются обменными [7,9], а "простая задача" становится неавтономной, так как ее решение зависит от решения других "простых задач". Реализация обменных краевых условий определяется самим алгоритмом решения. Так, для конечно-разностных методов в случае явных схем это может быть просто переброска массивов, в случае неявных приходится рассматривать параметрические краевые условия и неавтономная "простая задача" должна уже решаться как параметрическая.

Рассмотрим теперь какую роль играет "простая задача" в организации параллельных вычислений в задачах математической физики. Прежде всего заметим, что при редукции исходной задачи к последовательности "простых задач" (модульная декомпозиция), как правило, возникает иерархическая структура. В этой структуре сегментация области решения приводит к стыковкам "простых задач" по пространству только через обменные краевые условия, что образует горизонтальный уровень. Требование однородности математической модели (обычно расщепление по физическим процессам) образует вертикальный уровень, когда строго последовательно решение одной "простой задачи" передается следующей в качестве исходных значений. Нетрудно видеть, что горизонтальный уровень этой структуры естественным образом приспособлен для распараллеливания вычислений. Таким образом, модульная декомпозиция задачи по существу решает проблему распараллеливания вычислений. В качестве примера рассмотрим смешанную задачу Коши для уравнения теплопроводности

$$\frac{\partial u}{\partial t} = \sum_{m=1}^2 \frac{\partial}{\partial x_m} \left(a_m(\mu) \frac{\partial u}{\partial x_m} \right); \quad a_m(\mu) > 0,$$

$$u(\mu, 0) = \varphi(\mu), \quad \mu \in D, \quad u(\mu', t) = \psi(\mu', t), \quad \mu' \in \gamma \quad (1)$$

Для простоты положим $D = \{0 \leq x_m \leq 1\}$, тогда $\gamma = \bar{D} - D$. Пусть далее $\tau > 0$, $N_m h_m = 1$ (обозначения см. в [10]). В случае применения здесь явной разностной схемы можно, например, провести сегментацию области D на прямоугольные под-области $\bar{D}_i (\cup_i \bar{D}_i = \bar{D})$ и определить на \bar{D}_i простую задачу:

$$u_{i,j}^{n+1} = (E + \tau(\Lambda_1 + \Lambda_2)) u_{i,j}^n$$

причем на границах γ_i будут определены обычные и (или) обменные краевые условия. Для реализации последних достаточно ввести примыкающие к границам фиктивные узлы разностной сетки и организовать переброску в них значений искомой функции из соседних подобластей. Однако гораздо целесообразнее воспользоваться расщеплением по пространственным переменным

$$u_{i,j}^{n+1/2} = (E + \tau\Lambda_1) u_{i,j}^n, \quad (2)$$

$$u_{i,j}^{n+1} = (E + \tau\Lambda_2) u_{i,j}^{n+1/2}. \quad (3)$$

Тогда простой задачей в \bar{D}_i будет решение одномерного уравнения теплопроводности. Реализация обменных краевых условий здесь существенно проще, а при переходе от (2) к (3) достаточно выполнить только операцию транспонирования. Распараллеливание здесь очевидно: в (2) счет по индексу i можно выполнять одновременно для всех j , а в (3) наоборот. Кроме того, внутри простой задачи также допускается распараллеливание, дробность которого ограничена только шаблоном разностной схемы. Такая приспособленность явных схем к распараллеливанию породила ряд необоснованных высказываний об усилении их роли в связи с появлением многопроцессорных ЭВМ. Однако это совершенно не так. Еще в работе 1965 г. Г.И. Марчука и Н.Н. Яненко [8] было указано на возможность параллельной реализации неявных схем расщепления. Это высказывание позднее было повторено Д. Дугласом в дискуссии по большим компьютерам. Покажем, как для (1) организовать распараллеливание в случае применения неявных схем.

Будем использовать неявную схему расщепления с весами

$$(E - \tau\alpha\Lambda_1) u_{i,i}^{n+1/2} = (E + \tau\beta\Lambda_1) u_{i,i}^n,$$

$$u^{n+1/2}|_{x=0,1} = (E - \tau\Lambda_2) \psi^{n+1}, \quad j = 1, 2, \dots, N_2 - 1,$$

$$(E - \tau\gamma\Lambda_2) u_{i,i}^{n+1} = (E - \tau\delta\Lambda_2) u_{i,i}^{n+1/2},$$

$$u^{n+1}|_{x=0,1} = \psi^{n+1}, \quad i = 1, 2, \dots, N_1 - 1$$

Здесь, как и прежде, (4), (5) есть простые задачи, допускающие параллельный счет по j в (4) и по i в (5). При сегментации на \bar{D}_i реализация обменных краевых условий осуществляется с помощью параметрической прогонки [11, 12]. Кроме того, имеется возможность распараллеливания вычислений и в самой простой задаче путем распараллеливания прогонки [11].

Ряд других примеров, подтверждающих эффективность применения неявных схем при распараллеливании вычислений, приведен в работе [13]. В заключение отметим, что модульный анализ и параллельные вычисления органически связаны между собой и дальнейшее совместное развитие этих вопросов позволит существенно повысить эффективность решения задач математической физики на ЭВМ.

ЛИТЕРАТУРА

1. Яненко Н.Н., Коновалов А.Н. Технологические аспекты численных методов математической физики // Acta Univ. carol. math. Phys. 1974. N 1/2. P. 47-53.
2. Софронов И.Д. Оценка параметров вычислительной машины, предназначенной для решения задач механики сплошной среды // Числ. методы механики сплош. среды. 1975. Т. 6, № 3. С. 98-147.
3. Коновалов А.Н., Яненко Н.Н. Модульный принцип построения программ как основа создания пакета прикладных программ решения задач механики сплошной среды // Комплексы программ математической физики. Новосибирск, 1972. С. 48-54.
4. Коновалов А.Н. Модульный анализ вычислительного алгоритма в задаче планового вытеснения нефти водой // Тр. III семинара по комплексам программ мат. физики. Новосибирск, 1973. С. 81-94.
5. Коновалов А.Н. О принципах построения пакета программ для решения задач математической физики // Acta Politechn. Pr. CVUT. 1975. Vol. 4. P. 37-49.
6. Коновалов А.Н., Карначук В.И., О пакете прикладных программ математической физики // Структура и организация пакетов программ: Междунар. конф.: (Тез. докл.). Тбилиси: Мецниереба, 1976. С. 51-52.
7. Яненко Н.Н., Карначук В.И., Коновалов А.Н. Проблемы математической технологии // Числ. методы механики сплош. среды. 1977. Т. 8, № 3. С. 129-157.
8. Яненко Н.Н. Метод дробных шагов решения многомерных задач математической физики. Новосибирск: Наука, 1967. 197 с.
9. Ковеня В.М., Яненко Н.Н. Некоторые проблемы развития пакетов программ для решения задач аэродинамики // Числ. методы механики сплош. среды. 1979. Т. 10, № 3. С. 89-99.
10. Самарский А.А. Введение в теорию разностных схем. М.: Наука, 1971.
11. Яненко Н.Н., Коновалов А.Н., Бугров А.Н., Шустов Г.В. Об организации параллельных вычислений и "распараллеливание" прогонки // Числ. методы механики сплош. среды. 1978. Т. 9. №7. С. 139-146.
12. Воеводин А.Ф. Метод прогонки для разностных уравнений, определенных на комплексе // Журн. вычисл. математики и мат. физики. 1973. Т. 13, № 2. С. 494-497.

Яненко Н.Н., Коновалов А.Н. Некоторые вопросы теории модульного анализа и параллельного программирования для задач математической физики и механики сплошной среды // Современные проблемы математической физики и вычислительной математики. М.: Наука, 1982.

* Комплексы программ математической физики. (Матер. VI Всесоюз. семинара по комплексам

программ мат. физики). Новосибирск; 1980. С. 3-12.