ВЫЧИСЛЕНИЕ ИНТЕГРАЛОВ ПО МНОГОМЕРНЫМ ОБЛАСТЯМ НА МНОГОПРОЦЕССОРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ*

Д. Я. Рахматуллин
Институт математики с ВЦ Уфимского НЦ РАН, Россия
e-mail: nicknamej@rambler.ru

An algorithm for approximate integration of functions is considered. The domain is a hyper prism with one smooth curved boundary. Results of the computations using the proposed parallel algorithm are analyzed.

Введение

Вычисление интегралов по областям больших размерностей сдерживается отсутствием достаточных вычислительных мощностей. Это NP-полная задача, и даже самые быстрые персональные компьютеры вкупе с лучшими методами не могут справиться с этой задачей за разумное время при удовлетворительной точности вычислений. Применение суперкомпьютеров — мощных многопроцессорных вычислительных систем — хотя и не устранило полностью, но существенно ослабило ограничения на входные параметры задачи — размерность пространства и количество верных знаков в ответе. Если вычислительный алгоритм достаточно хорошо распараллеливается, то предельная размерность пространства, в котором требуется найти интеграл, ограничивается лишь количеством процессоров конкретного суперкомпьютера, а оно на данный момент может измеряться тысячами. Мы используем алгоритм, основанный на применении решетчатых кубатурных формул с ограниченным пограничным слоем. Он почти идеально распараллеливается, т. е. исходная задача почти поровну разбивается на ряд подзадач, независимо решаемых на каждом процессоре. При этом обмен данными между процессорами сведен к минимуму, так что итоговое время вычислений уменьшается с увеличением числа процессоров практически пропорционально. Суперкомпьютер МВС-15 000, на котором тестировалась программа, установлен в Межведомственном суперкомпьютерном центре РАН. Для вычислений предоставлялось 922 процессора с распределенной, т. е. отдельной для каждого двухпроцессорного вычислительного модуля, памятью.

*Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (грант № 03-07-90077-в) и Программы президиума РАН (№ 17).

1. Алгоритм

Рассмотрим пространство Соболева $W_p^m(\Omega)$, $\Omega \subset \mathbb{R}^n$, функций, интегрируемых с p-й степенью ($p > 1$) вместе с производными до m-го порядка включительно. Здесь $m \in \mathbb{N}$ — гладкость функций, а Ω — ограниченная область с гладкой границей Γ, лежащая вместе со своим замыканием в некоторой гиперпризме (многомерном прямоугольном параллелепипеде) Q [1].

Мы берем это пространство в одной из эквивалентных норм:

$$\|f\|_{\widetilde{W}_p^m(\Omega)} = \inf_{\substack{g\in\widetilde{W}_p^m(Q),\\ g|_\Omega=f|_\Omega}} \|g\|_{\widetilde{W}_p^m(Q)}, \quad \|g\|_{\widetilde{W}_p^m(Q)} = \left(|g_0|^p + \int\limits_Q dx \left| \sum_{k\neq 0} g_k (1+|k|^2)^{m/2} e^{2\pi i x k} \right|^p \right)^{1/p},$$

где

$$g_k = \int\limits_Q dx\, g(x)\, e^{-2\pi i x k}.$$

Будем использовать последовательность кубических решеток $\{hk\}_{k\in\mathbb{Z}^n}$ при $h \to 0$. Наша задача состоит в возможно более точном приближении интеграла

$$I^\Omega(f) := \int\limits_\Omega dx f(x)$$

решетчатыми кубатурными формулами вида

$$K_h^\Omega(f) := h^n \sum_{\substack{hk\in\Omega,\\ k\in Z^n}} c_k(h) f(hk), \quad h \to 0.$$

Введем несколько определений.

Определение 1. Последовательность кубатурных формул (ПКФ) называется *оптимальной*, если каждый ее элемент минимизирует норму его разности с точным значением интеграла по множеству коэффициентов этого элемента:

$$\forall h \quad K_h^{\Omega,\mathrm{opt}} := \arg\min_{\{c_k(h)\}} \left\| I^\Omega - K_h^\Omega \right\|_{(W_p^M(\Omega))^*}.$$

Определение 2. Последовательность кубатурных формул $K_h^{\Omega,\mathrm{as}}$ называется *асимптотически оптимальной*, если она имеет оптимальные свойства в пределе, т. е. выполняется следующее равенство:

$$\lim_{h\to 0} \frac{\left\| I^\Omega - K_h^{\Omega,\mathrm{as}} \right\|_{(W_p^m(\Omega))^*}}{\left\| I^\Omega - K_h^{\Omega,\mathrm{opt}} \right\|_{(W_p^m(\Omega))^*}} = 1.$$

Определение 3. Последовательность кубатурных формул называется *оптимальной по порядку*, если существует такая, не зависящая от h константа C, что верна оценка

$$\overline{\lim_{h\to 0}} \frac{\left\| I^\Omega - K_h^{\Omega,\mathrm{ord}} \right\|_{(W_p^m(\Omega))^*}}{\left\| I^\Omega - K_h^{\Omega,\mathrm{opt}} \right\|_{(W_p^m(\Omega))^*}} \leqslant C.$$

Определение 4. Последовательностью кубатурных формул с ограниченным пограничным слоем (ОПС) называется такая ПКФ, для которой выполняются два условия

1. Все ее коэффициенты равномерно ограничены по h и k:

$$\sup_{h,\ k} |c_k(h)| \leqslant L_1.$$

2. Всем узлам решетки, содержащимся в области Ω, за исключением *пограничного слоя* толщины $L_2 h$, соответствуют коэффициенты, равные единице:

$$\forall k,\ h: \quad \rho(kh,\ \mathbb{R}^n \backslash \Omega) \geqslant L_2 h \Rightarrow c_k(h) = 1.$$

Мы будем рассматривать именно ОПС-формулы. Верна следующая теорема:

Теорема 1. *Пусть* $1 \leqslant p_1 \leqslant p_2 < \infty$ *и* $\dfrac{n}{p_1} < m_1 < m_2$. *Последовательность кубатурных формул* $K_h^{\Omega}(f)$ *с ОПС асимптотически оптимальна на пространствах*

$$\left\{ \widetilde{W}_p^m(\Omega) \right\}_{m \in (m_1,\ m_2),\ p \in (p_1,\ p_2)}$$

тогда и только тогда, когда она оптимальна на них по порядку [2].

Определение 5. Последовательностью функционалов погрешности (ПФП) $l_h^{\omega}(f)$, соответствующей ПКФ $K_h^{\omega}(f)$, называется последовательность, каждый элемент которой представляет собой разность точного интеграла $I^{\omega}(f)$ от функции f по области ω и соответствующего элемента ПКФ:

$$l_h^{\omega}(f) := I^{\omega}(f) - K_h^{\omega}(f), \quad h \to 0,$$

или в виде обобщенной функции:

$$l_h^{\omega}(x) := \chi_{\omega}(x) - h^n \sum_{\substack{hk \in \omega, \\ k \in Z^n}} c_k(h) \delta(x - hk), \quad h \to 0,$$

где $\chi_{\omega}(x)$ — характеристическая функция области ω.

Приведем теперь алгоритм построения ПКФ с нужными нам свойствами.

Для каждой точки $x \in \overline{\Omega}$ можно указать окрестность $U(x)$ такую, что попавшая в нее часть границы $\Gamma \bigcap U(x)$ может быть гладко спроектирована на одну из координатных плоскостей. Так как $\overline{\Omega}$ ограничена, существует ее конечное покрытие $\{U_t\}_{t=1}^T$, $U_t = U(x^{(t)})$. Пусть разбиение единицы $\{\alpha_t\}_{t=1}^T$ подчинено этому покрытию, т. е.

$$\forall \alpha_t \in C_0^{\infty} \quad \operatorname{supp} \alpha_t \subset U_t, \quad \sum_{t=1}^T \alpha_t(x) = 1.$$

Введем обозначение: $V_t = \Omega \bigcap U_t$. Последовательность функционалов погрешности для всей области Ω получим суммированием по областям V_t локальных ПФП, помноженных на функции разбиения единицы:

$$l_h^{\Omega}(x) = \sum_{t=1}^T \alpha_t(x)\ l_h^{V_t}(x),$$

где

$$l_h^{V_t}(x) := \chi_{V_t}(x) - h^n \sum_{\substack{hk\in V_t,\\ k\in Z^n}} c_k^t(h)\delta(x-hk), \quad h\to 0. \tag{1}$$

Тогда $l_h^{\Omega}(x)$ будет иметь коэффициенты $c_k^a(h)$, вычисляемые по формуле

$$c_k^a(h) = \sum_{t=1}^{T} \alpha_t(hk)\ c_k^t(h). \tag{2}$$

Таким образом, если уметь считать коэффициенты локальных ПФП $l_h^{V_t}(x)$ для любого t, то задача будет решена. Далее мы приведем формулы, дающие явный вид коэффициентов локальных ПФП.

Рассмотрим любую из локальных ПФП $l_h^{V_t}(x)$. Будем обозначать штрихом $(n-1)$-мерный вариант того или иного обозначения, например $(x_1,...,x_{n-1},x_n) = (x',x_n)$; $(\delta'(x'),\varphi(x)) = \varphi(0',x_n)$. Пусть для определенности $\Gamma_t \equiv \overline{\Gamma \bigcap U_t} = \left\{x : x_n = \gamma(x'), \gamma\in C^M\right\}$, т. е. последняя координата любого вектора из множества Γ_t явно выражается через остальные координаты. Будем предполагать, что $x_n > \gamma(x')$. Тогда для коэффициентов ПФП $l_h^{V_t}(x)$ можно записать явное выражение [1]:

$$c_k^t = \begin{cases} \sum\limits_{p=1}^{M+1} \dfrac{1}{p} \sum\limits_{q=1}^{M+1} \eta^{q-1} \sum\limits_{s=1}^{\min\{k_n-\xi-1,M+1\}} w_{sq} \sum\limits_{r=1}^{\min\{k_n-\xi-s,M+1\}} w_{rp}, & k_n > 1+\xi, \\ 0, & k_n \leqslant 1+\xi. \end{cases}$$

Здесь $\xi = \left[\dfrac{\gamma(hk')}{h}\right]$, $\eta = \left\{\dfrac{\gamma(hk')}{h}\right\}$ — целая и дробная части числа $\dfrac{\gamma(hk')}{h}$, а $\{w_{i,j}\}_{i,j=1}^{M+1}$ — элементы матрицы, обратной к матрице Вандермонда $(M+1)\times(M+1)$:

$$\begin{pmatrix} 1 & 1 & ... & 1 \\ 1 & 2 & ... & M+1 \\ ... & ... & ... & ... \\ 1 & 2^M & ... & (M+1)^M \end{pmatrix}.$$

Построенная таким образом ПФП $l_h^{\Omega}(x)$ является оптимальной по порядку на любом пространстве $W_p^m(\Omega)$ с $m < M$ [1], а значит, по теореме 1, и асимптотически оптимальной. В частности, для любого $m < M$ выполняется оценка

$$\left\|l_h^{\Omega}\right\|_{\left(W_p^m(\Omega)\right)^*} \leqslant C\ h^m, \quad h\to 0.$$

Так как $\left(l_h^{\Omega}(x), f(x)\right) \leqslant \left\|l_h^{\Omega}\right\|_{\left(W_p^m(\Omega)\right)^*} \|f\|_{W_p^m(\Omega)}$, рассчитанный программой ответ может разниться с точным на величину произведения константы, нормы функции f и величины h^m.

Составим программу приближенного вычисления интеграла $\int\limits_{V_t} dx f(x)\alpha(x)$ с использованием формул (1) и (2).

2. Программа

Для вычислений по изложенному алгоритму написана программа на языке C++ с использованием библиотеки параллельных функций MPI. На данный момент она реализует основную часть алгоритма: при заданной области V_t с явно определенной границей Γ_t она вычисляет интеграл от произведения функции $f(x)$ и срезывающей функции $\alpha_t(x)$, вычисляя коэффициенты c_k^t локальной ПФП $l_h^{V_t}(x)$.

Мы предполагаем, что область V_t вложена в два соединенных в направлении n-й координаты единичных гиперкуба, в одном из которых находится гладкая граница Γ_t. Таким образом, V_t — вытянутая гиперпризма, в которой одна из плоских границ заменена произвольной гладкой поверхностью.

В качестве параметров программы следует указывать:

— размерность пространства n;

— подынтегральную функцию $f(x)$;

— срезывающую функцию $\alpha_t(x)$;

— $\gamma(x')$ — функцию, задающую поверхность Γ_t;

— параметр гладкости M;

— параметр разбиения h; Задается как N^{-1}, где N — количество точек решетки на ребре единичного куба. Общее количество узлов в гиперпризме Q вычисляется как $\widehat{N} := |Q|N^n$;

— коэффициенты растяжения области V_t по всем координатам;

— количество процессоров P.

Следует отметить, что используемые нами кубатурные формулы являются условно-ненасыщаемыми. Хотя параметр гладкости M задается заранее, наш алгоритм применим и для функций, реальная гладкость m которых меньше или больше M. В первом случае теоретическая погрешность вычислений имеет порядок h^m, во-втором — h^M. Так как наши эксперименты проводились с бесконечно гладкими функциями f, мы имели второй случай, т. е. гладкость таких функций искусственно понижалась до порядка гладкости M, фактически учитываемого в программе. Ясно, что теоретически в этом случае лучше брать возможно большее M. Однако, так как от параметра M зависит размер матрицы Вандермонда, с обратной к которой мы имеем дело, при увеличении M усиливаются ее плохие свойства. Кроме того, поскольку параметр M лежит в основе нескольких вложенных циклов нашей программы, при его увеличении, как будет показано позже, значительно возрастает время вычислений. В связи с этим величина M варьировалась в не очень широком диапазоне — от 2 до 6.

3. Вычислительные эксперименты

Программа тестировалась при следующих параметрах:

1) n от 2 до 10;

2) $f(x) = \sin\left(\sum\limits_{i=1}^{n} a_i x_i^{b_i}\right)$, $a = (1, 2, 3, 1, 1..., 1)$, $b = (1, 2, 3, 1, 1..., 1)$;

3) $\alpha(x_1, ..., x_n) = \prod\limits_{i=1}^{n-1} \varphi(x_i)\, \psi(x_n)$, где $\varphi(t) = \xi(2\varepsilon t)\xi(2\varepsilon(1-t))$, $\psi(t) = \xi(1+\varepsilon-2t)$,

а $\varepsilon = 1$ и

$$\xi(t) = \begin{cases} 0, \ t < 0 \\ \int\limits_0^t z(t)dt / \int\limits_0^\varepsilon z(t)dt, \quad z(t) = (t(\varepsilon - t))^M, \quad 0 \leqslant t < \varepsilon, \\ 1, \ \varepsilon \leqslant t; \end{cases}$$

4) $\gamma(x') = \dfrac{1}{4}\sin\left(\sum\limits_{i=1}^{n-1} c_i x_i^{d_i}\right) + \dfrac{1}{2}$, где $c = (1, 2, 3, 1, 1..., 1)$, $d = (1, 2, 3, 1, 1..., 1)$;

5) M от 2 до 6.

6) N от 5 до 100000.

7) Коэффициенты растяжения области V_t: $(1, ...1, 2)$ — n-мерный вектор.

8) P от 1 до 900.

Заметим, что независимыми параметрами являются лишь n, M, N и P, поэтому в процессе тестирования следует варьировать лишь их.

Точность вычислений рассчитывалась по устойчивости десятичных знаков в ответе при использовании последовательности уменьшающегося параметра h. Верными знаками при этом считались совпадающие десятичные знаки в ответах при текущем и следующем — меньшем значении h (см., например, [3]). В результате экспериментов установлено, что при фиксированных значениях параметров n, M, N и изменении P точность вычислений не изменяется, т. е. она не зависит от числа процессоров. Далее, если фиксировать n, M и увеличивать N, то точность повышается. То же самое происходит при постоянных параметрах n, N и возрастающем M. Например, при $n = 2$ получаем табл. 1.

Сравнивая при каждом M вышестоящие ответы с нижестоящими, мы нашли верные (устойчивые) знаки и убрали несовпадающие (округление учитывалось). Количество верных знаков для каждого результата показано в табл. 2

Теоретически ожидаемые порядки точности представлены в табл. 3. Как видно, при теоретической точности до 16 знаков практические результаты не менее точны, тогда как далее сказываются ошибки округления. В программе используются числа типа long double с 15 (в ОС UNIX) значащими цифрами после точки.

Согласно теории, при увеличении размерности n и сохранении величин N и M постоянными погрешность ПФП имеет порядок $h^m = N^{-m}$, т. е. не зависит от размерности пространства. Однако не нужно забывать, что на практике мы должны учитывать важный дополнительный параметр — предоставленное для вычислений время. Следует подбирать число N так, чтобы счет не оказался слишком долгим. Для этого нужно поддерживать

Т а б л и ц а 1. Результаты экспериментов, $n = 2$

$N \backslash M$	2	3	4	5	6
100	0.03539	0.032350	0.03034590	0.028926864	0.0278703404
1000	0.03539291	0.0323504440	0.0303459075339	0.02892686406822	0.027870340463228
10 000	0.03539291153	0.0323504440474	0.030345907533967	0.028926864068224	0.0278703404632276
100 000	0.035392911532	0.03235044404743	0.0303459075339675	0.0289268640682242	0.0278703404632276

Т а б л и ц а 2. Количество верных знаков

$N \backslash M$	2	3	4	5	6
100	5	6	8	9	10
1000	8	10	13	14	15
10 000	11	13	15	15	16

Т а б л и ц а 3. Теоретическая точность

$N \backslash M$	2	3	4	5	6
100	4	6	8	10	12
1000	6	9	12	15	18
10 000	8	12	16	20	24

постоянным общее количество узлов в гиперпризме $Q : \widehat{N} := |Q|N^n$, а значит, при увеличении n уменьшать N.

Практика показала, что при максимальном на МВС-15000 количестве процессоров оптимальное по затратам времени общее количество узлов — 10^{10} точек. Поэтому, если для размерности $n = 2$ мы можем взять до 10^5 точек в одном измерении, то для $n = 10$ вынуждены ограничиться значением $N = 10$. В этом случае пограничный слой кубатурной формулы, т. е. приграничная часть области, где пересчитываются коэффициенты, сравним по объему со всей областью, что плохо сказывается на точности вычислений. В самом деле, счет с параметрами $n = 10,\ N = 10,\ M = 2$ дает лишь четыре верных знака.

Теперь проанализируем практическую значимость применения большого количества процессоров в нашей задаче. Время вычислений при варьировании параметров $n,\ M,\ N$ и числа процессоров P изменялось следующим образом.

При фиксированных $n,\ M,\ N$ и увеличивающемся числе процессоров P время вычислений уменьшалось почти пропорционально возрастанию P. Пусть, например, $n = 7$, $M = 5$, $N = 15$.

Т а б л и ц а 4. Реальный случай ($n = 7,\ M = 5,\ N = 15$)

P	100	200	300	400	500	600	700	800	900
T_P	262.0	135.0	89.0	68.0	53.0	47.0	39.0	34.0	30.0
S_P	100.0	194.07	294.38	385.29	494.34	557.45	671.79	770.59	873.33
E_P	1.0	0.97	0.98	0.96	0.99	0.93	0.96	0.96	0.97

Т а б л и ц а 5. Идеальный случай

P	100	200	300	400	500	600	700	800	900
T_P	262.0	131.0	87.33	65.50	52.40	43.67	37.43	32.75	29.11
S_P	100.0	200.0	300.0	400.0	500.0	600.0	700.0	800.0	900.0
E_P	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

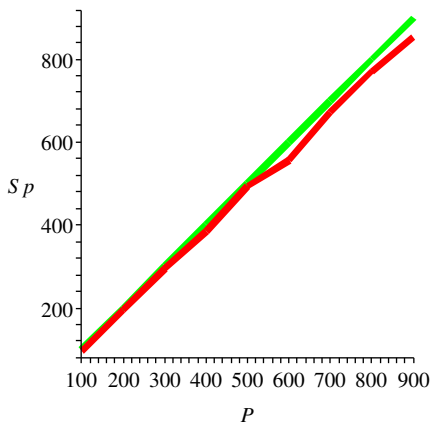

Рис. 1. Зависимость ускорения от числа процессоров.

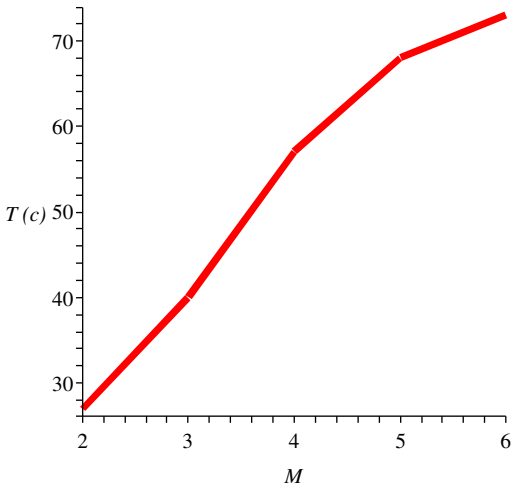

Рис. 2. Зависимость времени вычислений от гладкости.

Для лучшего анализа введем понятия ускорения и эффективности распараллеливания программы:

$$S_P = \frac{T_1}{T_P} \quad \text{(ускорение)},$$

$$E_P = \frac{S_P}{P} \quad \text{(эффективность)},$$

где T_P — время, за которое задача выполняется на P процессорах. Будем варьировать P от 100 до 900 с шагом 100. При этом положим $T_1 = 100\ T_{100}$. Дело в том, что реальное время T_1 слишком велико, поэтому фактически идеальным для нас будет случай, когда $P = 100$. Имеем табл. 4.

В идеальном (пропорциональном) случае было бы так, как в табл. 5.

На графике (рис. 1) наглядно показано отклонение экспериментального ускорения S_P (ломаная линия) от идеального ускорения (прямая).

Зависимость времени вычислений от параметра гладкости M при прочих равных условиях продемонстрируем на примере (рис. 2). Пусть $n = 10,\ N = 5,\ P = 20$. Как видно, при увеличении M время счета возрастает. Это связано с тем, что в алгоритме активно используется матрица, обратная к матрице Вандермонда размера $(M+1) \times (M+1)$.

Список литературы

[1] Соболев С.Л., Васкевич В.Л. Кубатурные формулы. Новосибирск: Изд-во ИМ СО РАН, 1996. 484 с.

[2] Ramazanov M.D. To the L_p-Theory of sobolev formulas // Siberian Advances in Mathematics. 1999. Vol. 9, N 1. P. 99–125.

[3] Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы: Учеб. пособие. М.: Наука. Гл. ред. физ.-мат. лит., 1987. 600 с.

Поступила в редакцию 23 декабря 2005 г.,
в переработанном виде — 7 февраля 2006 г.