Параллельные итерационные альтернирующие методы для решения трехмерного диффузионно-конвективного уравнения. *

Д.В. Перевозкин, Н.В. Панченко

Целью данной работы была разработка, реализация и экспериментальное исследование параллельных алгоритмов для итерационного решения симметричных и не симметричных систем линейных алгебраических уравнений (СЛАУ) с разреженными матрицами высокого порядка возникающими при сеточных аппроксимациях трехмерных задач теплопроводности.

1. Введение

В работе рассматриваются итерационные параллельные алгоритмы решения сеточных систем линейных алгебраических уравнений (СЛАУ), возникающих при аппроксимации трехмерных краевых задач. Пусть требуется найти функцию $u(x, y, z)$, которая является решением задачи Дирихле для диффузионно-конвективного уравнения

$$\begin{cases} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} + p\frac{\partial u}{\partial x} + q\frac{\partial u}{\partial y} + r\frac{\partial u}{\partial z} = f(x, y, z), \quad (x, y, z) \in \Omega, \\ u|_\Gamma = g(x, y, z), \end{cases} \tag{1}$$

в области $\Omega = [0, 1]^3$ с границей Γ.

Предполагается существование, единственность и достаточная гладкость решения, необходимая для использования далее описанной аппроксимации.

Исходная задача (1) аппроксимируется методом конечных объемов на параллелепипедальной сетке Ω^h, формируемой при помощи пересечения плоскостей, перпендикулярных координатным линиям

$$\Omega^h = [x_0, x_{N_x+1}] \times \left[y_0, x_{N_y+1}\right] \times [z_0, z_{N_z+1}],$$

$$x_{i+1} = x_i + h_i^x,\ i = 0, 1, \ldots, N_x, \quad y_{j+1} = y_j + h_j^y,\ j = 0, 1, \ldots, N_y,$$

$$z_{k+1} = z_k + h_k^z,\ k = 0, 1, \ldots, N_z,$$

где h_i^x, h_j^y, h_k^z — шаги сетки, $x_0 = y_0 = z_0 = 0$, $x_{N_x} = y_{N_y} = z_{N_z} = 1$. В результате получим семиточечную систему линейных алгебраических уравнений (СЛАУ):

$$\begin{gathered} p^0_{i,j,k}u_{i,j,k} - p^1_{i,j,k}u_{i-1,j,k} - p^2_{i,j,k}u_{i,j-1,k} - p^3_{i,j,k}u_{i+1,j,k} - p^4_{i,j,k}u_{i,j+1,k} - \\ -p^5_{i,j,k}u_{i,j,k-1} - p^6_{i,j,k}u_{i,j,k+1} = f_{i,j,k}, \end{gathered} \tag{2}$$

где $u_{i,j,k}$ — приближенное решение задачи (1) в узлах сетки, $p^n_{i,j,k},\ f_{i,j,k},\ n = 0, 1, \ldots, 6$ — известные величины. Причем вследствие учета граничных условий в (2) для $i = 1, N_x,\ j = 1, N_y,\ k = 1, N_z$ занулены коэффициенты $p^1, p^3, p^2, p^4, p^5, p^6$ соответственно, а их значения, умноженные на значение функции u на границе, перенесены в правую часть уравнения.

Сеточные коэффициенты $p^0, p^1, p^2, p^3, p^4, p^5, p^6$ вычисляются следующим образом ([1],

*Работа поддержана грантом РФФИ №08-01-00526 и грантом Президиума РАН №2.5.

с.57):

$$p^0_{i,j,k} = h_x^{-2}\left(e^{-ph_x/2} + e^{ph_x/2}\right) + h_y^{-2}\left(e^{-qh_y/2} + e^{qh_y/2}\right) + h_z^{-2}\left(e^{-rh_z/2} + e^{rh_z/2}\right),$$

$$p^1_{i,j,k} = h_x^{-2}e^{-ph_x/2}, \qquad p^2_{i,j,k} = h_y^{-2}e^{-qh_y/2}, \qquad p^5_{i,j,k} = h_z^{-2}e^{-rh_z/2},$$

$$p^3_{i,j,k} = h_x^{-2}e^{ph_x/2}, \qquad p^4_{i,j,k} = h_y^{-2}e^{qh_y/2}, \qquad p^6_{i,j,k} = h_z^{-2}e^{rh_z/2},$$

Преимуществом данной аппроксимации является порядок $O(h^2)$, в отличие от схем с использованием разности вперед или разности назад.

Запишем (2) в матричном виде

$$Au = f, \tag{3}$$

где $u = \{u_{i,j,k}\}$, $f = \{f_{i,j,k}\}$ - искомый вектор и вектор правой части, $A = \{p_{i,j}\} \in \mathbb{R}^{N,N}$, где $N = N_xN_yN_z$, — вещественная, квадратная, невырожденная матрица, которую считаем положительно определенной в смысле

$$(Au, u) > \delta(u, u),\ \delta > 0, \forall u \in \mathbb{R}^{N_x \cdot N_y \cdot N_z},$$

хотя, вообще говоря, вопрос о положительной определенности матрицы A в общем случае остается открытым.

Данная работа построена следующим образом: в пункте 2 предлагается новый метод решения задачи (3), основанный на методе симметричной последовательной верхней релаксации, в пункте 3 рассматривается использование полученного метода в качестве предобуславливателя для методов сопряженных направлений, и последний пункт посвящен обсуждению результатов численных экспериментов.

2. Альтернирующий метод симметричной последовательной верхней релаксации

Рассмотрим метод симметричной последовательной верхней релаксации (SSOR) [1] для решения СЛАУ (3). Представим матрицу A из (3) в виде $A = D - L - U$, где D — диагональная или блочно-диагональная матрица, U, L - верхняя и нижняя треугольные матрицы, которые зависят от упорядоченности узлов в области Ω^h. Тогда метод SSOR имеет следующий вид:

$$\begin{cases} (D - L)\hat{u}^{n+1/2} = Uu^n + f, \\ u^{n+1/2} = \omega\hat{u}^{n+1/2} + (1 - \omega)u^n, \\ (D - U)\hat{u}^{n+1} = Lu^{n+1/2} + f, \\ u^{n+1} = \omega\hat{u}^{n+1} + (1 - \omega)u^{n+1/2}, \end{cases} \tag{4}$$

где ω — числовой параметр, принадлежащий интервалу $(0, 2)$ ([1], с.284).

Представим метод симметричной последовательной верхней релаксации в каноническом виде, для чего из уравнений (4) исключим $u^{n+1/2}$:

$$B(u^{n+1} - u^n) = f - Au^n,$$

$$u^{n+1} = (I - B^{-1}A)u^n + B^{-1}f,$$

где предобуславливающая матрица B имеет вид

$$B = (G - L)G^{-1}(G - U), \quad G = \frac{1}{\omega}D.$$

При выборе оптимального ω будем использовать эмпирический подход, заключающийся в выборе такого ω_{opt}, чтобы выполнялось условие $(Ae, e) = (Be, e)$, где $e = (1, 1, \ldots, 1)$. При этом формула для ω_{opt} имеет вид

$$\omega_{opt} = \frac{a - \sqrt{a^2 - 4ab}}{2b} = \frac{a}{2b}\left(1 - \sqrt{1 - \frac{4b}{a}}\right) = \frac{2}{1 + \sqrt{1 - \frac{4b}{a}}}, \tag{5}$$

где $b = \left(LD^{-1}Ue, e\right)$, $a = (De, e)$. Если $1 - \frac{4b}{a} < 0$, полагаем $\omega_{opt} = 1$.

Положим $n_z = [N_z/2]$ и представим сеточную область Ω^h в виде

$$\Omega^h = \Omega_1^h \cup \Omega_2^h \cup \Omega_0^h,$$

$$\Omega_1^h = \left\{(i, j, k) | k < n_z\right\}, \ \Omega_2^h = \left\{(i, j, k) | k > n_z\right\}, \ \Omega_0^h = \left\{(i, j, k) | k = n_z\right\}.$$

Упорядочим сеточные узлы и соответствующие компоненты вектора u следующим образом: сначала пронумеруем узлы из Ω_1, используя такую упорядоченность, при которой индекс k меняется медленнее всего, затем узлы из Ω_2 с использованием упорядоченности, обратной принятой в Ω_1, и, наконец, узлы из Ω_0. При такой упорядоченности узлов матрица A имеет вид

$$A = \begin{pmatrix} A_{11} & 0 & A_{10} \\ 0 & A_{22} & A_{20} \\ A_{01} & A_{02} & A_{00} \end{pmatrix},$$

где $A_{11} = D_1 - L_1 - U_1$, $A_{22} = D_2 - L_2 - U_2$, а алгоритм (4) перепишется следующим образом:

$$\begin{cases} D_1 \hat{u}_1^{n+1/2} = f_1 + L_1 u_1^{n+1/2} + U_1 u_1^n - A_{10} u_0^n, \\ u_1^{n+1/2} = \omega \hat{u}_1^{n+1/2} + (1-\omega) u_1^n, \\ D_2 \hat{u}_2^{n+1/2} = f_2 + L_2 u_2^{n+1/2} + U_2 u_2^n - A_{20} u_0^n, \\ u_2^{n+1/2} = \omega \hat{u}_2^{n+1/2} + (1-\omega) u_2^n, \\ A_{00} \hat{u}_0^{n+1/2} = f_0 - A_{01} u_1^{n+1/2} - A_{02} u_2^{n+1/2}, \\ u_0^{n+1} = u_0^{n+1/2} = \omega \hat{u}_0^{n+1/2} + (1-\omega) u_0^n, \\ D_1 \hat{u}_1^{n+1} = f_1 + L_1 u_1^{n+1/2} + U_1 u_1^{n+1} - A_{10} u_0^{n+1}, \\ u_1^{n+1} = \omega \hat{u}_1^{n+1} + (1-\omega) u_1^{n+1/2}, \\ D_2 \hat{u}_2^{n+1} = f_2 + L_2 u_2^{n+1/2} + U_2 u_2^{n+1} - A_{20} u_0^{n+1}, \\ u_2^{n+1} = \omega \hat{u}_2^{n+1} + (1-\omega) u_2^{n+1/2}, \end{cases} \tag{6}$$

Метод (6) назовем альтернирующим методом симметричной последовательной верхней релаксации (ASSOR). При использовании такого переупорядочивания метод (4) удается распараллелить на два потока.

Для вычисления $u_0^{n+1/2}$ используется соотношение

$$A_{00} \hat{u}_0^{n+1/2} = f_0 - A_{01} u_1^{n+1/2} - A_{02} u_2^{n+1/2}, \tag{7}$$

которое предлагается решать либо с помощью метода ASSOR, либо каким-либо прямым или итерационным методом.

Данный итерационный метод можно представить в форме

$$u^{n+1} = Tu^n + g, \qquad n = 1, 2, \ldots \tag{8}$$

где T и g - некоторые матрица перехода и вектор. Если $u^n \to u$ при $n \to \infty$, то

$$\bar{A}u = (I - T)u = g. \tag{9}$$

К полученной системе (9) можно применить метод полусопряженных градиентов, описываемый в следующей главе формулами (10), (11).

3. Методы сопряженных и полусопряженных градиентов

Если матрица $\bar{A}$ — симметричная, то для решения (9) можно применить метод сопряженных градиентов (СГ) [2]:

$$\begin{gathered} r^0 = g - \bar{A}u^0, \ \ p^0 = r^0 \\ u^{n+1} = u^n + \alpha_n p^n, \ \ r^{n+1} = r^n - \alpha_n \bar{A}p^n, \ \ \alpha_n = \tfrac{(r^n, r^n)}{(\bar{A}p^n, p^n)}, \end{gathered} \tag{10}$$

для которого направляющие векторы находятся по формулам

$$p^{n+1} = r^{n+1} + \beta_n p^n, \ \ \beta_n = \frac{(r^n, r^n)}{(r^{n-1}, r^{n-1})}.$$

Когда же матрица $\bar{A}$ несимметрична, для решения СЛАУ нельзя применять метод сопряженных градиентов, т.к. направляющие вектора не будут обладать свойством $\bar{A}$-ортогональности. Но можно применить обобщенный метод сопряженных градиентов — метод полусопряженных градиентов (ПСГ) [2], реализуемый с помощью формул (10) и следующих рекуррентных соотношений для направляющих векторов p^k:

$$\begin{gathered} p^{n+1} = r^{n+1} + \sum_{k=0}^{n} \beta_{n,k} p^k = p^{n+1,1} + \sum_{k=0}^{n} \beta_{n,k} p^k, \ l = 0, 1, \ldots, n, \\ p^{n+1,0} = r^{n+1}, \ p^{n+1,l} = p^{n+1,l-1} + \beta_{n,l-1} p^{l-1}. \end{gathered} \tag{11}$$

Векторы p_0, r_0 при использовании в качестве предобуславливателя метода вида (8) вычисляются по формулам:

$$p^0 = r^0 = g - \bar{A}u^0 = Tu^0 + g - u^0.$$

4. Численные эксперименты

Ниже представлены результаты численных экспериментов. Расчеты проводились на четырехпроцессорном сервере Itanium2 с частотой 1.5 ГГц и 64 ГБ памяти. Для тестов использовалась задача (1) с постоянными коэффициентами p, q, r, и решением, равным единице во всей области. Аппроксимация задачи производилась на кубической равномерной сетке. В качестве критерия остановки было выбрано соотношение

$$\|r^n\| < \varepsilon \|f\|, \ \ \varepsilon = 10^{-7}.$$

Для решения уравнения (7) использовался метод ASSOR или прямой решатель PARDISO из пакета Intel MKL, что отмечено в таблицах.

Таблица 1. Сравнение методов SSOR и ASSOR, $p = q = r = 16$

N		32			64			128		
ω		Iter	Time		Iter	Time		Iter	Time	
1.1	A	123	0.97	0.85	487	33.8	29.1	1889	1087	944
	AP	120	1.01	0.89	482	34.7	30.1	1879	1102	963
	-	122	0.96		485	33.5		1886	1085	
1.5	A	45	0.36	0.31	198	13.7	11.8	772	448	402
	AP	40	0.34	0.30	191	13.8	11.9	760	463	414
	-	41	0.32		122	13.3		763	446	
1.7	A	34	0.27	0.24	107	7.48	6.46	416	240	208
	AP	38	0.32	0.28	88	6.38	5.56	394	232	202
	-	35	0.28		88	6.13		396	228	
1.9	A	119	0.93	0.82	116	8.11	6.99	294	184	164
	AP	164	1.38	1.21	144	10.4	9.0	111	66.9	57.3
	-	120	0.94		115	8.00		100	58.1	
ω_{opt}	A	33	0.28	0.25	91	6.85	5.68	269	157.5	136.7
	AP	23	0.21	0.18	45	3.42	2.95	105	63.17	55.01
	-	27	0.23		102	7.35		285	167	

Таблица 2. ASSOR+SCG и ASSOR+PARDISO+SCG ($p = q = r = 16$)

N	32				64			128		
ω		Iter	Time		Iter	Time		Iter	Time	
1.1	AS	33	0.20	0.15	68	10.36	6.36	178	311	303
	APS	34	0.26	0.20	70	11.63	6.77	189	276.3	263.5
1.5	AS	20	0.10	0.06	41	4.59	2.83	87	150.2	140.6
	APS	23	0.15	0.10	49	6.37	3.77	102	189.8	87.5
1.7	AS	21	0.09	0.06	31	2.38	1.36	61	62.9	58.4
	APS	25	0.13	0.09	36	3.0	1.81	76	92.9	87.6
1.9	AS	30	0.17	0.12	35	3.57	2.34	53	62.9	59.4
	APS	38	0.27	0.23	43	5.05	2.85	50	58.1	54.3
ω_{opt}	AS	19	0.09	0.06	30	2.30	1.41	51	46.8	43.1
	APS	21	0.11	0.08	32	2.88	1.78	48	42.8	38.8

В таблице 1 проводится сравнение методов SSOR и ASSOR. В этой и последующих таблицах буква A означает метод ASSOR, прочерк — SSOR, P указывает на использование PARDISO. Время представлено для одного и двух потоков, соответственно. ω_{opt} вычислено с помощью формулы (5) и равно 1.59, 1.77 и 1.87 для сеток размерностей 32^3, 64^3, 128^3, соответственно.

В таблице 2 сравниваются методы ASSOR с ускорением методом полусопряженных градиентов.

Из таблиц 1, 2 видно, что использование прямого метода для решения вспомогательного уравнения (7) не дает существенного выигрыша, за исключением весьма узкого множества значений параметра ω.

Заключение.

Был построен параллельный метод, основанный на методе симметричной последовательной верхней релаксации. По скорости сходимости полученный метод оказался сравним с исходным. Параллельные версии алгоритма, однако, не показали существенного прироста производительности, что связано, скорее всего, с интенсивным использованием процессорами шины памяти. Поэтому представляется целесообразным проводить распараллеливание метода либо на вычислительных системах с распределенной памятью, либо на системах, имеющих более высокую скорость обмена между процессорами и памятью.

Литература

1. *Ильин В.П.* Методы конечных разностей и конечных объемов для эллиптических уравнений. // Новосибирск: Изд. ИМ СО РАН, 2000.

2. *Ильин В.П.* Методы и технологии конечных элементов. // Новосибирск: Изд. ИВМиМГ СО РАН, 2007.

Таблица 3. Количество итераций для всех методов, $p = q = r = 16$

N	32				64				128			
ω	A	AP	AS	APS	A	AP	AS	APS	A	AP	AS	APS
1.0	151	148	36	36	596	590	74	74	2308	2297	178	190
1.1	123	120	32	33	487	482	67	69	1889	1879	180	187
1.2	99	97	28	30	397	392	59	64	1540	1530	160	178
1.3	79	76	25	27	320	315	53	59	1244	1234	114	159
1.4	61	58	22	24	255	249	47	54	991	981	98	133
1.5	45	40	19	22	198	191	41	48	772	760	86	101
1.6	29	24	18	22	148	139	36	42	582	567	74	90
1.7	34	38	21	25	107	88	31	36	416	394	61	76
1.8	56	69	24	30	85	54	31	34	289	235	53	61
1.9	119	164	30	38	116	144	34	43	294	111	53	50