ՀԱՅԱՍՏԱՆԻ ԳԻՏՈՒԹՅՈՒՆՆԵՐԻ ԱԿԱԴԵՄԻԱՅԻ ԶԵԿՈՒՅՑՆԵՐ
ДОКЛАДЫ АКАДЕМИИ НАУК АРМЕНИИ
Том 93 1992 № 1

ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА

УДК 519.61

Член-корреспондент АН Армении А. Б. Нерсесян, Г. В. Агекян

Векторизованные алгоритмы решения систем с матрицами теплицева типа

(Представлено 9/IV 1991)

Применение многопроцессорных систем является одним из главных направлений повышения производительности ЭВМ. В связи с этим особое значение придается векторизации (распараллеливанию) известных последовательных алгоритмов решения конкретных задач. В качестве первого шага естественно осуществить «абстрактное» распараллеливание, когда не ограничивается число процессоров, синхронизируется счет и не учитываются потери на пересылку данных.

Условимся считать, что алгоритм решения данной задачи, определяемой входными данными в виде $(n\times n)$-матрицы, допускает эффективную векторизацию (линейное ускорение счета) при применении $p=p(n)$ процессоров, если независимо от n реализуема оценка $pt_p\leqslant \text{const}\, t_1$, где через t_k обозначено время обсчета (число тактов) на k процессорах. Для определенности в дальнейшем учитываются только мультипликативные операции.

Классические алгоритмы решения задач линейной алгебры, как правило, эффективно векторизуемы (см. [1, 2]). Так, алгоритмы решения системы с невырожденной $(n\times n)$-матрицей $A_n=\|a_{ij}\|$, $i, j=1, 2, \ldots, n$ по методу Гаусса требует выполнения $n^3/3+O(n^2)$, $(n\to\infty)$ умножений, а его векторизованный вариант (метод Гаусса—Жордана) может быть обсчитан на $p=n^2$ процессорах за $2n$ тактов. Решение той же системы методом ортогональных разложений распараллеливается уже непросто и здесь при счете на $3/2\, n^2$-n процессорах требуется $7n-10$ тактов (см. [3]).

В случае теплицевой матрицы $A_n=T_n=\|t_{i-j}\|$, $i, j=1, 2, \ldots, n$ алгоритм Левинсона—Тренча (см. [4, 5]) позволяет решить систему посредством $3n^2+O(n)$ умножений, а применение $6n$ процессоров ускоряет счет в $3n$ раз (n тактов). К сожалению, эти отличные характеристики получены при условии невырожденности всех ведущих подматриц T_k, $k=1, 2, \ldots, n$. В то же время неудобны для распараллеливания модификации этого подхода, основанные на «прыжках» через нулевые миноры $\{T_k\}$, $k\leqslant n-1$ и реагирующие на их расположение изменением структуры операций (см. [5, 6]).

В последнее время были разработаны несколько менее быстрые, но пригодные для обращения теплицевых матриц общего вида методы. Алгоритмы работ [7, 8] требуют выполнения соответственно порядка $25n^2$ и $19n^2$ операций, однако они оказались неустойчивыми [9, 10]. Алгоритм работы [11] ($19{,}5n^2$ операций) устойчив, но дополнительно требует невырожденности T_{n-1}. Хорошими вычислительными качествами обладает метод обращения ортогонального разложения, предложенный Г. Цыбенко ($23n^2$ операций [12, 13]).

В случае блочно-теплицевой матрицы и более общей матрицы малого теплицева ранга $T^l_{n,k} = \| a_{ij} \|^n_{i,j=1}$, состоящей из $(l \times l)$ блоков a_{ij}, удовлетворяющих соотношению

$$a_{i+1,j+1} - a_{ij} = \sum_{m=1}^{k} c_{mi} d_{mj}, \quad i, j = 1, 2, \ldots, n-1, \tag{1}$$

алгоритм Г. Цыбенко был обобщен в работе [14] на основе блочно-ортогонального разложения. Вычислительный эксперимент на ряде тестовых матриц подтвердил хорошие характеристики этого метода.

Как будет показано ниже, алгоритмы [12, 14] эффективно распараллеливаются.

1°. Остановимся на случае невырожденной блочно-теплицевой матрицы $T^l_{n,0} = T^l_n = \| t_{i-j} \|$, $i, j = 1, \ldots, n$, состоящей из $(l \times l)$-блоков t_k, $|k| \leqslant n-1$. Отметим, что (см. [4]) уравнение с матрицей $T^l_{n,k}$ сводится к уравнению с $T^{l_1}_n$, где $l_1 = l(k+1)$, однако это сведение не всегда рационально (см. также ниже, п. 3°).

Решение уравнения $T^l_n x = b$ может быть получено посредством операций следующего последовательного алгоритма [14]:

: INITIALIZATIONS :

$q_1 = t, \quad f_1^s = 0, \quad \beta_1^s = 0 \quad (s = 2, 3)$	$\beta_1^1 = \rho_1, \quad u_1 = [q_1, q_1]^{-1}$
$f_1^1 = q_1, \quad \rho_1 = 0, \quad \rho_{11} = E_l$	$a_1 = [q_1, b], \quad x_1 = \rho_1 u_1 a_1$

: MAIN LOOP :

For $j = 2$ to n do

$y_j^1 = -u_1 [q_1, z q_{j-1}]$	$x_j = x_{j-1} + \rho_j u_j a_j$
$y_j^2 = q_{2n-1, j-1}$	$v_j^1 = -[z^{j-1} t, f_{j-1}^1]$
$y_j^3 = -q_{n-1, j-1}$	$v_j^2 = q^*_{2n, j}$
$q_j = z q_{j-1} + \sum_{s=1}^{3} f_{j-1}^s y_j^s$	$v_j^3 = (q_{nj} - t_0 \rho_{1j})^*$
$\rho_j = z \rho_{j-1} + \sum_{s=1}^{3} \beta_{j-1}^s y_j^s$	$f_j^s = f_{j-1}^s + q_j u_j v_j^s, \quad s = 1, 2, 3$
$u_j = [q_j, q_j]^{-1}$	$\beta_j^s = \beta_{j-1}^s + \rho_j u_j v_j^s, \quad s = 1, 2, 3$
$a_j = [q_j, b]$	END

Здесь приняты следующие обозначения

$$t = [t_{-n+1}^{\prime\prime}, t_{-n+2}^{\prime\prime}, \ldots, t_{-1}^{\prime\prime}, t_0^{\prime\prime}, \ldots, t_{n-1}^{\prime\prime}, \overbrace{0, 0, \ldots, 0}^{(n-1)l}]^{\prime\prime}$$

$$z = \begin{bmatrix} 0 & 0 & \ldots & 0 & E_l \\ E_l & 0 & \ldots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \ldots & E_l & 0 \end{bmatrix}; \qquad W = \operatorname{diag}[\overbrace{0, \ldots, 0}^{(n-1)l}, \overbrace{1, 1, \ldots, 1}^{nl}, \overbrace{0, \ldots, 0}^{(n-1)l}]$$

$$E_l = \operatorname{diag}[\overbrace{1, 1, \ldots, 1}^{l}]; \quad [x, y] = x^* W y$$

Отметим, что матрицы t, q_j, f_j^s ($j = 1, 2, \ldots, n$; $s = 1, 2, 3$) имеют размерность $(3n - 2)\, l \times l$, а ρ_j, β_j^s ($j = 1, 2, \ldots, n$; $s = 1, 2, 3$) — $nl \times l$.

2°. Главным препятствием для эффективной векторизации является наличие в этом алгоритме „длинных" матричных (при $l = 1$ — скалярных) произведений при $j \geqslant 2$. Что же касается произведений $v_j^1 = -[z^{j-1} t, f_{j-1}^1]$, то их можно исключить, исходя из соотношений $v_j^1 = \rho_{1j}^* [t, t]$.

Обозначим теперь

$$\begin{aligned} &\nu_{ij}^s = [q_1, z^i f_j^s]; \quad \omega_{ij}^s = [z^i q_1, f_j^s]; \quad \tau_{ij}^s = [z^i f_j^s, b]. \quad s = 1, 2, 3 \\ &\nu_{ij}^4 = [q_1, z^i q_j], \quad \omega_{ij}^4 = [z^i q_1, q_j]. \quad \tau_{ij}^4 = [z^i q_j, b]. \end{aligned} \qquad (2)$$

Имеем

$$y_j^1 = -u_1 \nu_{1, j-1}^4, \quad [q_j, q_j] = \rho_{jj}^* \omega_{j-1, j}^4, \quad a_j = \tau_{0j}^4.$$

Векторизация основана на следующих формулах:

$$\begin{aligned}
&\nu_{ij}^4 = \nu_{i+1, j-1}^4 + \sum_{s=1}^{3} \nu_{i, j-1}^s y_j^s, \qquad i = 1, 2, \ldots, n - j + 1; \\
&\omega_{ij}^4 = \omega_{i-1, j-1}^4 + q_{n-i, 1}^* q_{n-1, j-1} - q_{2n-i, 1}^* q_{2n-1, j-1} + \sum_{s=1}^{3} \omega_{i, j-1}^s y_j^s, \\
&\qquad\qquad i = j - 1, j, \ldots, n - 1; \\
&\tau_{ij}^4 = \tau_{i+1, j-1}^4 + \sum_{s=1}^{3} y_j^{s*} \tau_{i, j-1}^s, \qquad i = 0, 1, \ldots, n - j; \\
&\nu_{ij}^s = \nu_{i, j-1}^s + \nu_{ij}^4 u_j v_j^s, \qquad s = 1, 2, 3; \quad i = 1, 2, \ldots, n + 1 - j; \\
&\omega_{ij}^s = \omega_{i, j-1}^s + \omega_{ij}^4 u_j v_j, \qquad s = 1, 2, 3; \quad i = j, j + 1, \ldots, n; \\
&\tau_{ij}^s = \tau_{i, j-1}^s + v_j^s u_j^* \tau_{ij}^4, \qquad s = 1, 2, 3; \quad i = 0, 1, \ldots, n - j.
\end{aligned} \qquad (3)$$

На первом этапе необходимо (при $j = 1$) вычислить $2n + 1$ матричных произведений ω_{i1}^4 ($i = 0, 1, \ldots, n$) и τ_{i1}^4 ($i = 0, 1, \ldots, n - 1$) и для этого можно задействовать максимальное количество процессоров на соответствующую серию тактов. Поскольку матрицы u_j положительно определены, для вычисления u_j^{-1} можно применить схему Гаусса—Жордана и при наличии l^2 процессоров произвести $2l$ тактов.

3°. Приводимая таблица содержит основные характеристики обсуждаемых алгоритмов.

Задача	Число процессоров	Память	Число тактов
$Tx = b$	1	$17n$	$24,5n^2$
	n	$29n$	$55n$
$T^l x = b$	1	$17nl^2$	$20l^3n^3 + 1.5l^2n^2$
	l^2n	$29nl^2$	$42nl + 12n$
$T^l_k x = b$, а)	1	$17l^2(k+1)^2 n$	$20l^3(k+1)^3 n^2 + 1,5l^2(k+1)^2 n^2$
	$l^2(k+1)^2 n$	$29l^2(k+1)^2 n$	$42l(k+1)n + 12n$
$T^l_k x = b$, б)	1	$(4k^2 + 12k + 12)l^2n$	$(8k^2 + 24k + 12)l^3n^2 + 1,5l^2n^2$
	l^2n	$(1 + 2k)l^2n^2$	$(13k^2 + 42k + 25)ln + 17n$

В первой графе через T обозначена скалярная теплицева $(n \times n)$-матрица, далее через T^l $(l \geqslant 2)$ — блочно-теплицева $(nl \times nl)$-матрица с $(l \times l)$-блоками, а через T^l_k — $(nl \times nl)$-матрица, удовлетворяющая условию (1). Решению уравнения с невырожденной матрицей последнего типа сведением T^l_k к блочно-теплицевому виду T^{l_1}, $l_1 = (k+1)l$ (см. п. 1°) соответствует предпоследняя строка а), а последняя строка б) относится к случаю непосредственной векторизации алгоритма работы [14] для матриц типа [1] общего вида. Эта векторизация может быть проведена совершенно аналогично п. 2°, и здесь мы на изложении этого не остановились не только из-за громоздкости формул, но и (главным образом) потому, что (см. таблицу) нам пока что не удалось освободиться от слишком (по сравнению с а)) большого объема памяти, хотя остальные характеристики в этом случае предпочтительней.

Отметим, что векторизация в изучаемых случаях не могла быть проведена по схеме T-алгоритмов, предложенной в [5], поскольку нам не удалось преобразовать исходные алгоритмы к требуемому для этого виду.

В таблице приведены результаты распараллеливания на оптимальном, в определенном смысле, количестве процессоров (если иметь в виду линейное ускорение и устойчивый счет), однако нетрудно использовать меньшее (большее) количество процессоров, соответственно усложнив (упростив) их функции. Это может оказаться полезным при практической реализации схемы на систолических массивах, когда необходимо найти разумный компромисс между сложностью функций каждого процессора и сложностью межпроцессорных связей (см. [2]).

4°. Остановимся теперь на использовании результатов с целью быстрого решения интегральных уравнений со скалярным разностным ядром $K = K(x-t)$

$$(\mu \mathbf{J} - \mathbf{K})\, y \overset{\text{def}}{=} \mu y(x) - \int_0^1 K(x-t)\, y(t)\, dt = f(x), \tag{4}$$

где $K \in C^p([0,1] \times [0,1])$, $f \in C^p([0,1])$, $p \geq 1$, $\mu = \text{const}$.

В случае уравнения второго рода ($\mu = 1$), если оператор $\mathbf{J} - \mathbf{K}$ обратим в $C([0,1])$, квадратурные формулы порядка p ($p \geq 1$) сводят задачу приближенного решения уравнения (4) к решению системы с матрицей T_l ($l \geq p$) при ошибке порядка $O(h^p)$ (h — длина максимального интервала дискретизации). Таким образом, при достаточно гладком ядре K эффективен именно алгоритм п. 1° или его приведенный в п. 2° распараллеленный вариант.

В случае уравнения первого рода ($\mu = 0$) необходимо, как известно, провести регуляризацию. Пусть, например, существование и единственность решения $y(x)$ обеспечены. Тогда при достаточно малом $\varepsilon > 0$ приближенным можно считать решение уравнения (см. [15])

$$\varepsilon y + \mathbf{K}^* \mathbf{K} y = \mathbf{K}^* f, \tag{5}$$

где $\mathbf{K}^*$ — сопряженный к $\mathbf{K}$ оператор. Заметим, что ядро оператора $\mathbf{K}\mathbf{K}^*$ (обозначим его через $K_1(x, t)$) удовлетворяет соотношению

$$\left(\frac{\partial}{\partial x} + \frac{\partial}{\partial t}\right) K_1(x, t) + \left\{\overline{K(s-x)}\, K(s-t)\right\}_{s=0}^{s=1} = 0. \tag{6}$$

Отсюда следует, что дискретизация уравнения (5) сводит его решение к решению алгебраической системы с матрицей T_k^l вида (1). Таким образом, и в этом случае работают последовательные или параллельные алгоритмы, изученные выше (см. также [14]).

В случае матричного ядра в уравнении (4) ситуация вполне аналогична.

Институт математики Академии наук Армении

Հայաստանի ԳԱ թղթակից անդամ Ա. Բ. ՆԵՐՍԵՍՅԱՆ, Գ. Ռ. ԱՂԻԿՅԱՆ

Տյոպլիցյան տիպի մատրիցներով համակարգերի լուծման վեկտորացված ալգորիթմներ

Լուծվում է ([13, 14]) աշխատանքներում մշակված ալգորիթմների վեկտորացման (զուգահեռացման) խնդիրը, որը ապահովվում է արագ հաշվարկ բազմապրոցեսորային հաշվիչ մեքենաների օգնությամբ։

Աշխատանքում բերված են համապատասխան ալգորիթմների բնութագրիչները։

Արդյունքները օգտակար են նաև ինտեգրալ հավասարումների մոտավոր լուծման համար։

ЛИТЕРАТУРА—ԳՐԱԿԱՆՈՒԹՅՈՒՆ

[1] *В. Н. Фаддеева, Д. К. Фаддеев*, Кибернетика, № 6, с. 28—40, 1977, № 3, с. 18—31, 1982. [2] *В. В. Воеводин*, Математические модели и методы в параллельных процессах. Наука, М., 1986. [3] *A. H. Sameh, D. J. Kuck*, J. Assoc. Comput. Mach., v. 25, № 1, p. 81—91 (1978). [4] *В. В. Воеводин, Е. Е. Тыртышников*, в кн. Вычислительные процессы и системы, вып. 1, Наука, М., с. 124—267, 1983. [5] *В. В. Воеводин, Е. Е. Тыртышников*, Вычислительные процессы с теплицевыми матрицами. Наука, М., 1987. [6] *G. Heinig*, Wissensch. Zeitschrift d. TH Karl-Marx-Stadt, v. 25, № 3, p. 326—333, (1983). [7] *D. R. Sweet*, Numer. Math., v. 43, p. 1—21 (1984). [8] *A. W. Bojanczyk, R. P. Brent, F. R. De Hoog*, Numer. Math., v. 49, p. 81—94 (1986). [9] *F. T. Luk, S. Qiao*, J. Linear Algebra Appl., v. 88/89, p. 495—506 (1987). [10] *G. W. Stewart*, J. Inst. Math. Appl., v. 23, p. 203—213 (1979). [11] *Sanzheng Qiao*, Numer. Math., v. 53, p. 351—366 (1988). [12] *G. Cybenko*, Siam. J. Sci. Stat. Comput., v. 8, № 5, p. 734—740 (1987). [13] *G. Cybenko*, BIT, 24, p. 441—455 (1984). [14] *Г. В. Агекян, А. Б. Нерсесян*, Деп. в Арм. НИИНТИ, № 64—Ар 89, 1989. [15] *А. Н. Тихонов, В. Я. Арсенин*, Методы решения некорректных задач. Наука, М., 1974.