文章编号：1005－8893（2001）02－0052－02

一个反对称矩阵乘法的快速算法*

王　珂[1]，许　波[2]
（1. 苏州职工科技大学 基础课部，江苏 苏州 215004；2. 江苏石油化工学院 信息科学系，江苏 常州 213016）

摘要：矩阵乘法是数值计算中的常见问题，其运算阶的降低一直是人们关注的基本问题，而多项式求值、多项式插值及多项式求导问题迄今已出现了许多有效且稳定的快速算法。讨论了一个 n 阶反对称矩阵与 n 维列向量的乘法问题，证明了该问题与多项式求值问题的等价性，提出了一个运算阶为 $O(n(\log_2 n)^2)$ 的快速算法，并讨论了一个反对称矩阵乘法的例子，其 $O(n^2)$ 的运算阶在反对称矩阵乘法情形至少可降低到 $O(n(\log_2 n)^2)$。

关键词：多项式求值；算法；矩阵乘法；阶
中图分类号：O 241.6　　文献标识码：A

矩阵乘法是线性代数中常见的问题，由于其计算量非常大，传统算法的运算阶高达 $O(n^3)$，这直接影响到各种矩阵计算问题及包含矩阵乘法的许多数值计算问题的运行效率。自从 Strassen[1] 在 1969 年提出了一个阶为 $O(n^{\log_2 7})$ 的矩阵乘法算法以来，陆续出现了一系列新的的矩阵乘法算法。目前，串行算法的复杂性已经降到 $O(n^{2.494})$。而对于一些较为特殊的矩阵乘法，如两个 n 阶上（或下）三角形 Toeplitz 矩阵乘法及 n 阶 Toeplitz 矩阵乘 n 维列向量的算术运算次数更可以降到 $O(n(\log_2 n))$。文献 [2] 中提出了一个适用于整数矩阵乘法的算法，指出对于 n 行 m 列矩阵和 m 行 1 列矩阵的乘法，运算的次数阶为 $O(m(n+1))$，作者称之为最佳算法。并称其运算次数阶已达到理论下界。本文考察了一个 n 阶反对称矩阵与 n 维列向量的乘法问题，证明了该问题与多项式求值问题的等价性，利用运算阶为 $O(n(\log_2 n)^2)$ 的多项式求值与多项式插值的快速算法，提出了该矩阵乘法问题的快速算法，其运算次数阶不超过 $O(n(\log_2 n)^2)$，并进一步给出了一个实例，说明文献 [2] 提出的阶为 $O(n^2)$ 的算法在反对称矩阵情形运算阶可降低到 $O(n(\log_2 n)^2)$。

设 n 阶反对称矩阵 $\boldsymbol{B}=(b_{ij})$，其中 $b_{ij}=\frac{1}{c_i-c_j}$，$i\neq j$ 时；$b_{ij}=0$，$i=j$ 时，c_i，$i=1,\cdots,n$ 各不相同。$Y=(y_1y_2\cdots y_n)^{\mathrm{T}}$ 是 n 维列向量，计算 $\boldsymbol{B}Y=X=(x_1x_2\cdots x_n)^{\mathrm{T}}$。

首先，我们来考察上述问题与多项式求值的等价性。令

$$T(x)=\prod_{i=1}^{n}(x-c_i) \tag{1}$$

$$s(x)=T(x)\sum_{i=1}^{n}\frac{y_i}{x-c_i} \tag{2}$$

（2）式中令 $x=c_j$，得

$$s(c_j)=y_jT'(c_j),\quad j=1,2,\cdots,n \tag{3}$$

记 $f_j(x)=\dfrac{s(x)}{T(x)}-\dfrac{y_j}{x-c_j}=\sum\limits_{\substack{i=1\\ i\neq j}}^{n}\dfrac{y_i}{x-c_i}$，

$$t_j(x)=\prod_{\substack{i=1\\ i\neq j}}^{n}(x-c_i)=\frac{T(x)}{x-c_j},$$

$j=1,2,\cdots,n$

则 $T'(x)=(x-c_j)t'_j(x)+t_j(x)$，

$T''(x)=(x-c_j)t''_j(x)+2t'_j(x)$

所以 $t'_j(c_j)=\dfrac{1}{2}T''(c_j)$，$j=1,2,\cdots,n$

以及 $f_j(c_j)=\lim\limits_{x\to c_j}f_j(x)=$

* 收稿日期：2001－03－23
作者简介：王珂（1962－），女，江苏无锡人，讲师，主要从事计算数学方面的研究。

$$\lim_{x\to c_j}\frac{s(x)-y_j t_j(x)}{T(x)}=$$

$$\lim_{x\to c_j}\frac{s'(x)-y_j t'_j(x)}{T'(x)}=\frac{s'(c_j)-\frac{1}{2}y_j T''(c_j)}{T'(c_j)}$$

故 $\boldsymbol{B}Y=X$ 中的分量

$$x_j=\sum_{\substack{i=1\\ i\neq j}}^{n}\frac{y_i}{c_j-c_i}=f_j(c_j)=\frac{s'(c_j)-\frac{1}{2}y_j T''(c_j)}{T'(c_j)},$$

$j=1, 2, \cdots, n$ (4)

由此可见，从式 (4) 计算向量 X 等价于多项式值的计算。

下面，我们给出此问题的一个快速算法。

Step 1. 计算 $T(x)=\sum_{i=0}^{n}a_i x^i$ 的系数 a_i，运算次数为 $O(n(\log_2 n)^2)$[3]；

Step 2. 计算 $T'(x)$ 和 $T''(x)$ 的系数，运算次数为 $O(n)$；

Step 3. 计算 $T'(c_j)$ 和 $T''(c_j)$，$j=1, 2, \cdots, n$，运算次数为 $O(n(\log_2 n)^2)$[4]；

Step 4. 计算 $s(c_j)=y_j T'(c_j)$，$j=1, 2, \cdots, n$，运算次数为 $O(n)$；

Step 5. 计算多项式 $s(x)$ 的系数，用插值法计算运算次数为 $O(n(\log_2 n)^2)$[3]；

Step 6. 计算 $s'(c_j)$，$j=1, 2, \cdots, n$，运算次数为 $O(n(\log_2 n)^2)$[4]；

Step 7. 计算 $x_j=\frac{s'(c_j)-\frac{1}{2}y_j T''(c_j)}{T'(c_j)}$，$j=1, 2, \cdots, n$，运算次数为 $O(n)$。

Steps 1 ~ 7 的总运算次数不超过 $O(n(\log_2 n)^2)$。在许多情况下，我们可以利用点集 $\{c_j\}$ 的一些特殊结构来简化此计算法，进一步地降低算法的运算次数，这些尚有待进一步的探讨。

据文献 [2] 提供的算法，计算 n 阶整数矩阵 $\boldsymbol{B}$ 与 n 维整数向量 Y 的乘法，运算阶为 $O(n^2)$。下面我们考察这样一个例子：

令 $c_i=i$，$i=1, 2, \cdots, n$，$\boldsymbol{B}=(b_{ij})$ 其中 $i\neq j$ 时，$b_{ij}=\frac{1}{c_i-c_j}=\frac{1}{i-j}$；$i=j$ 时，$b_{ij}=0$，$i, j=1, 2, \cdots, n$，Y 为 n 维整数向量。

于是，$n!\boldsymbol{B}$ 为整数矩阵，从而按文献 [2] 的算法计算 $n!\boldsymbol{B}Y$，运算阶为 $O(n^2)$，计算 $\boldsymbol{B}Y$ 的运算阶也为 $O(n^2)$。然而此问题用本文提出的快速算法计算，其运算阶为 $O(n(\log_2 n)^2)$。

参考文献：

[1] Strassen. Gaussian Elimination Is Not Optimal [J]. Number Math, 1969, 4: 345－356.

[2] 蒋昌俊，吴哲辉. 矩阵乘法的一个最佳算法 [J]. 科学通报，1989, 4: 251－254.

[3] 游兆永. 线性代数与多项式的快速算法 [M]. 上海：上海科技出版社，1980. 88－89.

[4] Aho A, Hopcroft J E, Ullman J D. The Design and Analysis of Computer Algorithms [M]. New York: Addison－Wesley, 1974. 294－295.

A Fast Algorithm of Anti－symmetric Matrix Multiplication

WANG Ke[1], XU Bo[2]

(1. Department of General Courses, Suzhou Worker &Staff University of Science Technology, Suzhou 215004, China; 2. Department of Information Science, Jiangsu Institute of Petrochemical Technology, Changzhou 213016, China)

Abstract: The matrix multiplication is a common question in numerical caculation. For its massive caculation, lower order of caculation is a basic question. But finding an interpolation polynomial and polynomial evaluation have been discussed extensively, many stable fast algorithms have so far been presented. In this paper, we study a multiplication of n－order anti－symmetric matrix with vector, and prove the equivalence of the problem with polynomial evaluation, and describe a fast algorithm with $O(n(\log_2 n)^2)$ arithmetic time complexity. Furthermore, we present an anti－symmetric matrix and illustrate that for the best algorithm of matrix multiplication, its caculation order can be dropped at least to $O(n(\log_2 n)^2)$.

Key words: polynominal evaluation; algorithm; matrix multiplication; order