Fast algorithms to compute matrix-vector products for Toeplitz and Hankel matrices

Abstract. The paper presents practical and effective algorithms to calculate the Toeplitz/Hankel matrix by a vector product that are recursiveless modification of Karatsuba's method. Unlike traditional algorithms, in this case using the FFT is not required. Realization of the developed algorithms involves the use of unconventional ways of choosing the elements of the initial transformation matrix during the formation of an array of data to be processed. We have called these methods respectively "7-order" technique and "mirrored 7-order" technique. This approach allows us to calculate the vector-matrix products in parallel way with a reduced number of hardware multipliers and adders.

Introduction

Matrices with special structure play a fundamental role in systems and control [1-7]. In digital signal processing, data coding and statistics is often necessary to compute the product of a Toeplitz matrix \( T_N = \mathbf{t}_1 \otimes \mathbf{t}_1 \) by arbitrary vector \( \mathbf{x}_{N \times 1} = [x_0, x_1, \ldots, x_{N-1}]^T \):

\[
(1) \quad \mathbf{y}^{(T)}_{N \times 1} = T_N \mathbf{x}_{N \times 1}
\]

where \( \mathbf{y}^{(T)}_{N \times 1} = [y_0, y_1, \ldots, y_{N-1}]^T \) is a vector which elements are the result of multiplication, and Toeplitz matrix is defined as:

\[
(2) \quad T_N = \begin{bmatrix} t_0 & t_1 & \cdots & t_{N-1} \\
 t_1 & \ddots & \ddots & \vdots \\
 \vdots & \ddots & \ddots & t_1 \\
 t_{(N-1)} & \cdots & t_{-1} & t_0 
\end{bmatrix}
\]

where \( t_i \) are scalars, or their block equivalent, where \( t_i \) are then \( k \times k \) matrices [3].

In control, one often has to deal with Hankel matrix. Then the product of a Hankel matrix \( H_N = \mathbf{h}_1 \otimes \mathbf{h}_1 \) by vector \( \mathbf{x}_{N \times 1} \) can be represented as follows:

\[
(3) \quad \mathbf{y}^{(H)}_{N \times 1} = H_N \mathbf{x}_{N \times 1}
\]

where \( \mathbf{y}^{(H)}_{N \times 1} = [y_0, y_1, \ldots, y_{N-1}]^T \) is a vector which elements are the result of multiplication, and Hankel matrix is defined as [3]:

\[
(4) \quad H_N = \begin{bmatrix} h_0 & h_1 & \cdots & h_{N-1} \\
 h_1 & \ddots & \ddots & \vdots \\
 \vdots & \ddots & \ddots & h_1 \\
 h_{N-1} & \cdots & h_1 & h_0 
\end{bmatrix}_{N \times N}
\]

Słowa kluczowe: Szybki algorytm, iloczyn macierzowo-wektorowy, macierz Toeplitza, macierz Hankel.

Keywords: Fast algorithm, matrix-vector product, Toeplitz matrix, Hankel matrix.

Synthesis of computational procedures

It is clear that Hankel matrix can be easily transformed into Toeplitz matrix and vice versa by a simple rows or columns reversal, so only one algorithm could be used. However we decided to describe two separate approaches so that additional expenses for matrix shuffle could be avoided. If there is no such a need, there is no problem to use only one of proposed solution to both matrices types.

Let \( N = 2^m \), \( m = 1, 2, \ldots, n \).

Initially, we introduce a few matrices:

\[
A^{(k)}_{2^{m-k+1} \times 2^{m-k+1}} = I_{2^{m-k+1}} \otimes P_{3 \times 2} \otimes I_{2^{m-k}}
\]
Using above defined matrices, we can write procedures for computing the Hankel matrix. By considering figure 1 (for Toeplitz matrix) and figure 2 (for Hankel matrix), we can define the diagonal matrices $S_{3^x3^t}$ and $S_{3^y3^t}$ as follows:

$$S_{3^x3^t}^{(T)} = C^{(m)}_{3^x3^t} \cdots C^{(2)}_{3^x3^t} C^{(1)}_{3^x3^t} Z^{(T)}_{3^x3^t}$$

$$S_{3^y3^t}^{(H)} = C^{(m)}_{3^y3^t} \cdots C^{(2)}_{3^y3^t} C^{(1)}_{3^y3^t} Z^{(H)}_{3^y3^t}$$

where

$$Z^{(T)}_{3^x3^t} = \begin{bmatrix} z_0, z_1, \ldots, z_{3^x-1} \end{bmatrix}^T, \quad Z^{(H)}_{3^y3^t} = \begin{bmatrix} z_0, z_1, \ldots, z_{3^y-1} \end{bmatrix}^T$$

are vectors formed from elements of the matrix $T_N$ and $H_N$ selected on the basis of the "7-order" technique (or "mirrored 7-order" technique - for Hankel matrix). The essence of the mentioned techniques (for $N = 8$) is clear by considering figure 1 (for Toeplitz matrix) and figure 2 (for Hankel matrix).

Using above defined matrices, we can write procedures for fast computation of the matrix-vector products for Toeplitz and Hankel matrices:

$$Y^{(T)}_{2^n3^x3^t} = \begin{bmatrix} A^{(1)}_{2^n3^x3^t} & A^{(2)}_{2^n3^x3^t} & \cdots & A^{(m-1)}_{2^n3^x3^t} \\ \end{bmatrix} \begin{bmatrix} A^{(m)}_{2^n3^x3^t} \end{bmatrix} \times S_{3^x3^t}^{(T)}$$

$$Y^{(H)}_{2^n3^y3^t} = \begin{bmatrix} A^{(1)}_{2^n3^y3^t} & A^{(2)}_{2^n3^y3^t} & \cdots & A^{(m-1)}_{2^n3^y3^t} \\ \end{bmatrix} \begin{bmatrix} A^{(m)}_{2^n3^y3^t} \end{bmatrix} \times S_{3^y3^t}^{(H)}$$

Let us consider the synthesis of fast algorithms for computing Toeplitz/Hankel matrix-vector products for $N = 8$. In this case the corresponding matrices take the following form:
Rys.1. Illustration of the mechanism to choice of elements of Toeplitz matrix in accordance with the "7"-ordered technique

\[ B_{27 \times 18}^{(3)} = I_9 \otimes G_{3 \times 2} = \begin{bmatrix} 1 & 0 & 0_{3 \times 2} & 0_{3 \times 2} & 0_{3 \times 2} \\ 0 & 0 & 0_{3 \times 2} & 0_{3 \times 2} & 0_{3 \times 2} \\ 1 & 1 & 0 & 0_{3 \times 2} & 0_{3 \times 2} \\ 0 & 1 & 0 & 0_{3 \times 2} & 0_{3 \times 2} \\ 1 & 1 & 1 & 0_{3 \times 2} & 0_{3 \times 2} \end{bmatrix}, \]

\[ B_{18 \times 12}^{(2)} = I_3 \otimes G_{3 \times 2} \otimes I_2 = \begin{bmatrix} I_4 & 0_4 & 0_{6 \times 4} \\ 0_4 & I_4 & 0_{6 \times 4} \\ 0_4 & 0_{6 \times 4} & I_4 \\ 0 & 0_{6 \times 4} & 0_{6 \times 4} \\ 0 & 0_{6 \times 4} & 0 \end{bmatrix}, \]
Rys. 2. Illustration of the mechanism to choice of elements of Hankel matrix in accordance with the „mirrored 7”-ordered technique

\[ S_{27\times1}^{(T)} = C_{27}^{(2)} C_{27}^{(3)} \mathbf{X}_{27\times1}^{(T)} \]

\[ S_{27\times1}^{(H)} = C_{27}^{(2)} C_{27}^{(3)} \mathbf{X}_{27\times1}^{(H)} \]

\[ C_{27}^{(1)} = \begin{bmatrix} -I_9 & I_9 & 0_9 \\ -I_9 & 0_9 & I_9 \\ I_9 & I_9 & 0_9 \end{bmatrix} \]

\[ C_{27}^{(2)} = \begin{bmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0_3 \\ 0_3 \\ -1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \]
Then the computational procedures that describe fast algorithms for Toeplitz and Hankel matrix-vector multiplication for \( N = 8 \) are as follows.

\[ \text{C}^{(3)}_{27} = I_9 \otimes \Gamma_3 = \begin{bmatrix}
  -1 & 1 & 0 \\
  -1 & 0 & 1 \\
  1 & 0 & 0 \\
  0 & 0 & 0 \\
  0 & 0 & 0 \\
  0 & 0 & 0 \\
  0 & 0 & 0 \\
  0 & 0 & 0 \\
  0 & 0 & 0
\end{bmatrix} \]

(9) \[ Y_{8x1}^{(T)} = \tilde{X}_{8x12}^{(1)} \tilde{X}_{2x12}^{(2)} \tilde{X}_{10x27}^{(3)} S_{3x8}^{(T)} A_{8x1}^{(1)} A_{12x8}^{(2)} A_{18x12}^{(3)} X_{8x1} \]

(10) \[ Y_{8x1}^{(H)} = A_{8x12}^{(1)} A_{12x18}^{(2)} A_{18x27}^{(3)} S_{3x8}^{(H)} B_{18x12}^{(1)} B_{12x8}^{(2)} X_{8x1} \]

Figure 3 shows a signal-flow graph representation of the fast Toeplitz/Hankel algorithm. It should be noted that these elements are calculated the same way both for the Toeplitz and Hankel matrix. For this reason, the figure 3 used the generalized symbols \( y_i \) instead of \( y_i \), \( y_i \), and \( z_i \) instead of \( z_i \) and \( z_i \). We assume that the exact values of the variables will be clear from the context. In this paper, signal-flow graphs are oriented from left to right. Straight lines in the figures here and farther denote the operation of data transfer. At points where lines converge, the data are summarized. (The dashed lines indicate the subtraction operation). We use the common lines, without arrows, not to clutter the picture.

Figure 4 shows a signal-flow graph of the fast algorithm of \( N = 8 \) Toeplitz matrix-vector multiplication. Figure 5 shows a signal-flow graph of the fast algorithm of \( N = 8 \) Hankel matrix-vector multiplication. Note that the circles in these figures show the operation of multiplication by a number (variable) inscribed inside a circle.

**Assessment of computational cost**

Both algorithms presented in this paper have the same computational complexity as the Karatsuba’s method. The multiplicative complexity of algorithms is \( \theta_m = 3^m \) multiplications and additive complexity of algorithms is \( \theta_a = \sum_{i=1}^{m} (3^m \cdot 2^{m-i}) \) additions. The significantly lower than the naive \( N^2 \) complexity provides savings in terms of hardware resources. Additional advantage is possibility of parallelization so let us compares computation times for usual matrix-vector multiplication parallel algorithm and presented solution.

\[ r^{(norm)} = r_s + \log_2 N r_s \]

\[ r^{(T/H)} = r_s + 2 \log_2 N r_s \]

\( r_s \) and \( r_s \) denotes respectively time of one multiplication and one addition.

Even though additions time is two times longer in presented algorithms, the most important multiplication time remains the same. Along with reduced need for multipliers and adders, and limited resources of given hardware platforms we gain great advantage especially for matrices of large order. For example, matrix of order 32 needs 4 times more multiplication than in described method. If only 256 multipliers are available the times of execution are as follows: \( r^{(norm)} = 4r_s + 5r_s \) and \( r^{(T/H)} = r_s + 10r_s \).

<table>
<thead>
<tr>
<th>( N )</th>
<th>Multiplications</th>
<th>Additions</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>9</td>
</tr>
<tr>
<td>8</td>
<td>64</td>
<td>27</td>
</tr>
<tr>
<td>16</td>
<td>256</td>
<td>81</td>
</tr>
<tr>
<td>32</td>
<td>1024</td>
<td>243</td>
</tr>
<tr>
<td>64</td>
<td>4096</td>
<td>729</td>
</tr>
<tr>
<td>128</td>
<td>16384</td>
<td>2187</td>
</tr>
<tr>
<td>256</td>
<td>65536</td>
<td>6561</td>
</tr>
<tr>
<td>512</td>
<td>262140</td>
<td>19683</td>
</tr>
<tr>
<td>1024</td>
<td>1048576</td>
<td>59049</td>
</tr>
</tbody>
</table>

Table 1. Comparison of computational complexity for proposed and naive algorithms for various dimensions of the original matrices.

Table 1 shows the number of arithmetical operations required for execution of the developed algorithms for various dimensions of the original matrices.
Conclusion

We present two new parallel algorithms for computing matrix-vector products for Toeplitz and Hankel matrices. These algorithms require fewer multiplications and additions than the naïve way of performing calculations. They require even fewer multiplications than the algorithms based on fast Fourier transforms, as in their implementation there is no need to perform complex operations. Additional advantages are parallelization possibility with no recursion requirement. On specialized hardware platform these algorithms can be significantly faster than any other known approach, however further research is needed in that aspect. Since the calculation of the matrix-vector products with Toeplitz and Hankel matrices have numerous applications, synthesized computational procedures can be used to rationalize many practical problems in various fields of science and technology.

LITERATURA


Autorzy: dr hab. inż. prof. nadzwyczajny Aleksander Cariow, Zachodniopomorski Uniwersytet Technologiczny w Szczecinie, Wydział Informatyki, ul. Żołnierka 49, 70-210 Szczecin, E-mail: atarion@wi.ps.pl, mgr inż. Marek Gliszczynski, Zachodniopomorski Uniwersytet Technologiczny w Szczecinie, Wydział Informatyki, ul. Żołnierka 49, 70-210 Szczecin, E-mail: mgliszczynski@wi.ps.pl