What is the Method of Four Russians?

Kiran Jones*

March 2025

The "Method of Four Russians" is a technique for optimizing algorithms using precomputed lookup tables. The idea for the contemporary method was first introduced by Arlazarov, Dinic, Kronrod, and Faradzev in their 1970 paper[1] on graph theory; however, evidence has shown only one was Russian. Nevertheless, the name has stuck.

Matrix Multiplication

The Method is particularly useful in the case of binary matrix multiplication (M4RM). A binary matrix – also known as a Boolean matrix and a logical matrix – is a matrix with all entries being either 0 or 1. One common form of binary matrix is a permutation matrix. Equation 1 shows a sample binary matrix $M \in \mathbb{R}^{2\times 2}$, where $a, b, c, d \in \{0, 1\}$.

$$M = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \tag{1}$$

Consider the product of two binary matrices $AB = C$ where $A \in \mathbb{R}^{a\times b}$, $B \in \mathbb{R}^{b\times c}$, resulting in $C \in \mathbb{R}^{a\times c}$. We can choose a $k \in \mathbb{R}$ such that $k = \log_2(b)$ to partition the matrices A and B into sub-matrix sections. We divide A into vertical sections $A_1 \dots A_{b/k}$ and B into horizontal sections $B_1 \dots B_{b/k}$. Multiplying two sections, A_iB_i will yield a $a \times b/k$ by multiplication of the matrix $b/k \times c$, which ultimately produces a $a \times c$ matrix C_i. As shown in equation 2, this process can be expressed as the summation of all the C_i matrices.

$$C = AB = \sum_{i=0}^{k} C_i = \sum_{i=0}^{k} (A_iB_i) \tag{2}$$

Before any matrix multiplication is performed, we first need to create our lookup tables. For each $i = 1 \dots b/k$, we precompute all possible combinations 2^k of

*Department of Mathematics, Dartmouth College, 29 N Main St, Hanover, NH 03755 (kiran.p.jones.27@dartmouth.edu).

[1] Arlazarov et al. "On Economical Construction of the Transitive Closure of an Oriented Graph."

k rows of B_i. For example, consider $AB = C$ where $A, B, C \in \mathbb{R}^{4\times4}$ (e.g. $a, b, c = 4$). This matrix has a k value of $\log_2(4) = 2$. We can divide B into $n/k = 2$ horizontal "slices", B_1 and B_2. Now we can populate the lookup tables. There are 2^k possible binary row vectors r of length k, and we calculate the result of $r \vee B_i$ for each B_i. With $k = 2$, we have $r \in \{00, 01, 10, 11\}$. For each B_i our lookup table will be of the form:

r	$r \vee B_i$
00	$00 \vee B_i$
01	$01 \vee B_i$
10	$10 \vee B_i$
11	$11 \vee B_i$

After we have produced 2^k tables – each corresponding to a unique B_i and containing all possible r values – we can begin to compute the final product C. Using each row vector of A_i as an index to the lookup table of the corresponding B_i, we can retrieve the precomputed result $C_i = A_iB_i$ in $O(1)$ time. As we calculate all the C_i matrices, we progressively accumulate C. Note that as we are working with binary matrices, this sum is computed using boolean addition (i.e. $x, y \in R, x + y = (x + y) \bmod 2$). Once all C_i are summed, the result $C = \sum_{i=0}^{k} C_i$ is the solution to AB.

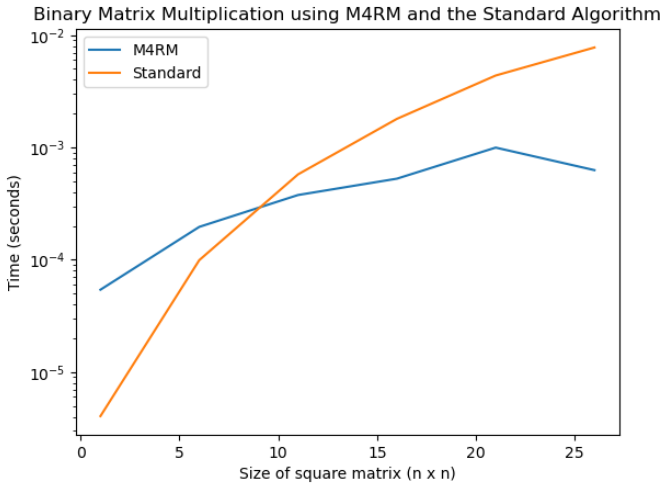

Figure 1: Performance of M4RM and standard matrix multiplication for small matrices.

As shown in Figure 1, the method is slower than traditional matrix multiplication for square matrices with less dimensions less than 10. With small matrices, any reduction in computation time is offset by the high initial cost of creating the lookup tables. As the dimension of the matrix grows the utility of the method becomes more clear.

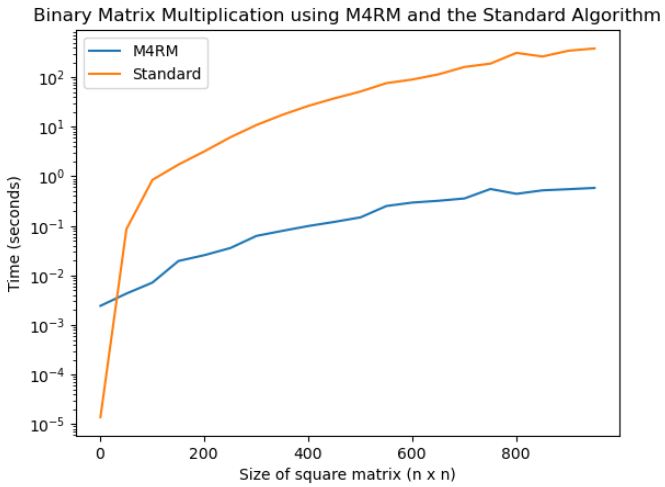

Figure 2: Performance of M4RM and standard matrix multiplication for larger matrices.

Figure 2 extends the results shown in Figure 1, testing matrices up to dimension 1000. At these scales, the method performs several orders of magnitude faster than traditional multiplication.

The time complexity of traditional matrix multiplication is $O(n^3)$. This is because – for an $n \times n$ matrix – each of the n^2 elements requires n operations. The method is able to reduce the total runtime to $O(\frac{n^3}{\log n})$. This is largely due to time savings from the preprocessing steps; by processing the matrix in blocks, the algorithm is able to minimize unneeded calculations.

References

- Bard, Gregory V. "The Method of Four Russians." Algebraic Cryptanalysis, Springer US, 2009, pp. 133–58, https://doi.org/10.1007/978-0-387-88757-9_9.
- Bard, Gregory V.. "Accelerating Cryptanalysis with the Method of Four Russians." IACR Cryptol. ePrint Arch. 2006 (2006): 251.
- Gusfield, Dan. Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology. Cambridge: Cambridge University Press, 1997. Print.

This article and its code are also available from the linked GitHub Repository.