arXiv:2601.05272v1 [cs.SC] 18 Dec 2025

一种秩为23的算法：以算术复杂度59乘以 3×3矩阵

Erik M̈artensson[1,2],Paul Stankovski Wagner[1], 和 Joshua Stapleton[3]

[1]隆德大学，瑞典隆德{erik.martensson,paul.stankovski wagner}@eit.lth.se[2]Advenica AB，马尔̇默̈，瑞典[3]帝国理工学院数学系，英国伦敦jbs123@ic.ac.uk

摘要

1969年，施特拉森出人意料地证明，两个 2×2 矩阵可以用七次乘法和18次加法相乘，而不必采用朴素方法所需的八次乘法和四次加法。后来，加法次数减少到了15次。Karstadt 和 Schwartz 通过基变换方法进一步将加法次数减少到12次。研究表明，对于 2×2 情形，乘法次数和加法次数都已达到最优。

对于 3×3 矩阵乘法，目前发现的最低乘法次数为 23 次。Schwart 等人通过基变换方法，展示了如何在使用 23 次乘法的情况下将加法次数降至 61 次。M̈artensson 和 Stankovski Wagner 展示了如何在不改变基的情况下实现 62 次加法。使用 M̈artensson 和 Stankovski Wagner 的优化方法，Stapleton 找到了一种只需 60 次加法的算法。在这项工作中，我们继续结合 M̈artensson、Stankovski Wagner 和 Stapleton 的方法，找到了一种只需 59 次加法的算法，且仍然不需要基变换。关于用于寻找该方案的方法和工具的技术细节，以及对这一发现影响的讨论，将在后续发表的论文中介绍。

1 引言

直接根据定义计算 2×2 矩阵需要进行八次乘法和四次加法。1969 年，施特拉森 [Str69] 展示了如何将乘法次数减少到七次，但代价是将加法次数增加到 18[1]。乘法次数也称为该算法的秩

[1]提到加法时，也默认包括加法和减法。

。在 [Win71] 中，秩 7 被证明是最优的。

对于秩 7 算法而言，计算 2×2 矩阵乘法所需的加法次数在 [Bsh95]中降至 15。所需的加法次数也称为算法的算术复杂度。Karstadt 和 Schwartz 通过基变换在 [KS20]中将这一数字改进为 12，并证明这是最优值。从计算复杂度的角度来看，执行基变换会引入低阶项，与执行一次加法的成本相比，这些项很小。

现在考虑一个 $n\times m$ 矩阵与一个 $m\times k$ 矩阵相乘。对于大多数维度大于 2×2 的矩阵，矩阵乘法的最优秩和最优算术复杂度都尚不明确。有关针对大量不同维度发现的方案汇总，请参见 [Sed25]。

对于 3×3 矩阵乘法，Laderman 在 1976 [Lad76]中通过手工找到了一个秩为 23 的算法。经过近半个世纪不断采用日益复杂的搜索方法，这一秩仍未得到改进。在本文中，我们为计算 3×3 矩阵乘法的秩为 23 的算法创下了加法次数的新纪录。

就秩为23的矩阵乘法算法所需的加法次数而言，Smirnov 找到了一种朴素实现需要84次加法的算法 [Smi13]。Smirnov 指出，他的算法所需的加法次数还可以进一步降低，但并未进行这项优化。Karstadt 和 Schwartz 将加法次数降至75次 [KS20]。Beniamini 等人进一步将这一数字降至61次 [BCH+20]。在不进行基变换的情况下，M˚artensson 和 Stankovski Wagner 展示了如何仅用62次加法实现 Laderman 的原始算法，几乎达到了 Beniamini 等人的结果 [MS25]。

最近，Stapleton 开发了一种利用神经网络生成低秩矩阵乘法方案的方法 [Sta25]。通过生成许多秩为23的方案，并使用 M˚artensson 和 Stankovski Wagner 的实现来减少加法次数，他找到了一种只需60次加法的算法。

我们进一步结合 Stapleton 生成快速矩阵乘法方案的方法，以及 M˚artensson 和 Stankovski Wagner 减少加法次数的方法，找到了一种只需 59 次加法的方案。该方案的朴素（未化简）形式需要 110 次加法，详见表 1。我们优化后的方案详见表 2，只需 59 次加法。我们采用紧凑索引记号

$$\begin{pmatrix} C_0 & C_1 & C_2 \\ C_3 & C_4 & C_5 \\ C_6 & C_7 & C_8 \end{pmatrix} = \mathbf{C} = \mathbf{AB} = \begin{pmatrix} A_0 & A_1 & A_2 \\ A_3 & A_4 & A_5 \\ A_6 & A_7 & A_8 \end{pmatrix} \begin{pmatrix} B_0 & B_1 & B_2 \\ B_3 & B_4 & B_5 \\ B_6 & B_7 & B_8 \end{pmatrix}.$$

最后，为方便起见，我们在附录中以便于 M˚artensson [M˚a24] 和 Stankovski Wagner [Sta24] 软件实现读取的形式，给出了表 3 中的 59-算法。

关于该方案的发现过程、所用工具和方法，以及这一发现带来的影响的更多技术细节，将在即将发表的论文中介绍。

$$
\begin{aligned}
M_0 &= A_1B_4M_1 = A_0B_1M_2 = (\\
&A_3 - A_4 - A_5 + A_6 - A_7 - A_8)B_8M_3 = (A_2 - A_8)\\
&B_8M_4 = (A_3 - A_5 + A_6 - A_8)B_6M_5 = A_2B_7\\
&M_6 = A_0B_0M_7 = (A_1 - A_7)(B_2 + B_5)\\
&M_8 = A_5(B_4 + B_5 - B_6 - B_7 - B_8)\\
&M_9 = A_4B_3M_{10} = (A_0 - A_1 - A_6 + A_7)B_2\\
&M_{11} = (A_3 + A_6 - A_8)(B_1 + B_2 + B_4 + B_5 - B_6)\\
&M_{12} = (A_1 + A_6 - A_7)(B_2 + B_4 + B_5)\\
&M_{13} = A_1B_3M_{14} = (A_1 + A_3 + A_6 - A_7)(\\
&B_2 - B_3 + B_5)M_{15} = (A_3 - A_4 + A_6 - A_7)(\\
&B_3 - B_5 + B_8)M_{16} = (A_3 - A_4 - A_5 + A_6 - A_7)(\\
&B_4 + B_5 - B_8)M_{17} = A_3(\\
&B_0 + B_1 + B_2 + B_4 + B_5)M_{18} = (A_3 - A_4 - A_5)(\\
&B_4 + B_5)M_{19} = A_8(B_1 + B_2 + B_4 + B_5 + B_7)\\
&M_{20} = A_2B_6M_{21} = (A_6 - A_8)(\\
&B_1 + B_2 + B_4 + B_5)M_{22} = (A_3 + A_6)(\\
&B_0 - B_2 + B_3 - B_5 + B_6)
\end{aligned}
$$

$$
\begin{aligned}
C_0 &= M_6 + M_{13} + M_{20}\\
C_1 &= M_0 + M_1 + M_5\\
C_2 &= -M_0 - M_2 + M_3 + M_{10} + M_{12} - M_{16} + M_{18}\\
C_3 &= -M_4 + M_9 - M_{11} + M_{17} + M_{21}\\
C_4 &= -M_0 + M_4 - M_8 - M_9 + M_{11} + M_{12} - M_{13} - M_{14} - M_{15} - M_{16} - M_{21}\\
C_5 &= M_0 + M_9 - M_{12} + M_{13} + M_{14} + M_{15} + M_{16} - M_{18}\\
C_6 &= -M_7 + M_{11} + M_{13} + M_{14} - M_{17} - M_{21} + M_{22}\\
C_7 &= M_0 + M_7 - M_{12} + M_{19} + M_{21}\\
C_8 &= -M_0 - M_2 - M_7 + M_{12} - M_{16} + M_{18}
\end{aligned}
$$

表 1：包含 110 次加法的快速矩阵乘法算法（约化前）。此处采用紧凑索引记号。该算法按实现顺序指定。请注意，通过重新排列各项，实际实现中完全避免了取负（负首项）。

$$
\begin{aligned}
t_0 &= A_3 + A_6 \\
t_1 &= A_1 - A_7 \\
t_2 &= A_4 + A_5 \\
t_3 &= A_7 - t_0 \\
t_4 &= A_6 + t_1 \\
t_5 &= A_8 - t_0 \\
t_6 &= t_2 + t_3 \\
\\
u_0 &= B_2 + B_5 \\
u_1 &= B_4 + u_0 \\
u_2 &= B_1 + u_1 \\
u_3 &= B_4 + B_5 \\
u_4 &= B_3 - u_0 \\
u_5 &= B_8 - u_3 \\
\\
M_0 &= A_1 B_4 \\
M_1 &= A_0 B_1 \\
M_2 &= (A_8 + t_6) B_8 \\
M_3 &= (A_2 - A_8) B_8 \\
M_4 &= (A_5 + t_5) B_6 \\
M_5 &= A_2 B_7 \\
M_6 &= A_0 B_0 \\
M_7 &= t_1 u_0 \\
M_8 &= A_5 (B_6 + B_7 + u_5) \\
M_9 &= A_4 B_3 \\
M_{10} &= (A_0 - t_4) B_2 \\
M_{11} &= t_5 (B_6 - u_2) \\
M_{12} &= t_4 u_1 \\
M_{13} &= A_1 B_3
\end{aligned}
\qquad
\begin{aligned}
M_{14} &= (t_0 + t_1) u_4 \\
M_{15} &= (A_4 + t_3)(B_3 - B_5 + B_8) \\
M_{16} &= t_6 u_5 \\
M_{17} &= A_3 (B_0 + u_2) \\
M_{18} &= (A_3 - t_2) u_3 \\
M_{19} &= A_8 (B_7 + u_2) \\
M_{20} &= A_2 B_6 \\
M_{21} &= (A_6 - A_8) u_2 \\
M_{22} &= t_0 (B_0 + B_6 + u_4) \\
\\
v_0 &= M_0 - M_{12} \\
v_1 &= M_{16} + v_0 \\
v_2 &= M_{11} - M_{21} \\
v_3 &= M_{14} - M_{13} \\
v_4 &= M_{18} - v_1 \\
v_5 &= M_2 + v_4 \\
v_6 &= M_4 + M_9 \\
v_7 &= M_{15} + v_3 \\
v_8 &= M_{17} - v_2 \\
\\
C_0 &= M_6 + M_{13} + M_{20} \\
C_1 &= M_0 + M_1 + M_5 \\
C_2 &= M_3 + M_{10} + v_5 \\
C_3 &= v_6 + v_8 \\
C_4 &= M_8 - v_1 + v_2 - v_6 + v_7 \\
C_5 &= M_9 - v_4 - v_7 \\
C_6 &= -M_7 + M_{22} - v_3 - v_8 \\
C_7 &= M_7 + M_{19} + M_{21} + v_0 \\
C_8 &= -M_7 + v_5
\end{aligned}
$$

表 2：使用 Greedy Vanilla 将快速矩阵乘法算法的加法次数降至 59 次，源自 [MS25]。Greedy Potential 未能带来改进。这里采用紧凑索引记号。算法按实现顺序指定。请注意，在实现中，通过重新排列各项可完全避免取负（负首项）。

参考文献

[BCH+20]Gal Beniamini、Nathan Cheng、Olga Holtz、Elaye Karstadt 和 Oded Schwartz。《快速矩阵乘法算法算子的稀疏化》，2020年。[Bsh95] Nader H. Bshouty。关于 2×2 矩阵乘法的加法复杂度。《信息处理快报》，56(6):329–335，1995年。[KS20] Elaye Karstadt 和 Oded Schwartz。《矩阵乘法：再快一点》。《J. ACM》，67(1)，2020年1月。[Lad76] Julian D. Laderman。一种使用23次乘法计算3×3 矩阵的非交换算法。《美国数学会通报》，82(1):126–128，1976年。[MS25] Erik M˚artensson 和 Paul Stankovski Wagner。《野兽之数：将维度不超过666的快速矩阵乘法算法中的加法次数降下来》，第47–60页。SIAM，2025年。[M˚a24] Erik M˚artensson。GitHub——fmm add reduction 的 Python 代码仓库。可在 github.com/ErikMaartensson/fmm add reduction 获取，最后访问于2026–01–12，2024年。[Sed25] Alexandre Sedoglavic。又一份快速矩阵乘法算法目录，2025年。https://fmm.univ-lille.fr/。[Smi13] A.V Smirnov。矩阵乘法的双线性复杂度与实用算法。《计算数学与数学物理》，53:1781–1795，2013年。[Sta24] Paul Stankovski Wagner。GitHub——fmm add reduction 的 C 代码仓库。可在 github.com/werekorren/fmm add reduction 获取，最后访问于2026–01–12，2024年。[Sta25] Joshua Stapleton。一个用于精确3x3矩阵乘法的60次加法、秩为23的方案，2025年。[Str69] Volker Strassen。高斯消元并非最优。《数值数学》，13(4):354–356，1969年8月。[Win71] S. Winograd。关于 2×2 矩阵的乘法。《线性代数及其应用》，4(4):381–388，1971年。

A 59-算法的文件格式

为方便使用，我们也提供了文件格式的59-算法。

```
 0  1  0  0  0  0  1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0
 1  0  0  0  0  0  0 -1  0  0 -1  0  1 -1  1  0  0  0  0  0  0  0  0
 0  0  0  1  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0 -1  0  0
 0  0  1  0  1  0  0  0  0  0  0 -1  0  0  1  1  1 -1  1  0  0  0 -1
 0  0 -1  0  0  0  0  0  0  1  0  0  0  0  0 -1 -1  0 -1  0  0  0  0
 0  0 -1  0 -1  0  0  0  1  0  0  0  0  0  0  0 -1  0 -1  0  0  0  0
 0  0  1  0  1  0  0  0  0  0 -1 -1  1  0  1  1  1  0  0  0  0 -1 -1
 0  0 -1  0  0  0  0  1  0  0  1  0 -1  0 -1 -1 -1  0  0  0  0  0  0
 0  0 -1 -1 -1  0  0  0  0  0  0  1  0  0  0  0  0  0  0 -1  0  1  0
#
 0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  1
 0  1  0  0  0  0  0  0  0  0  0 -1  0  0  0  0  0  1  0 -1  0 -1  0
 0  0  0  0  0  0  0 -1  0  0 -1 -1  1  0  1  0  0  1  0 -1  0 -1 -1
 0  0  0  0  0  0  0  0  0  1  0  0  0  1 -1 -1  0  0  0  0  0  0  1
 1  0  0  0  0  0  0  0  1  0  0 -1  1  0  0  0  1  1 -1 -1  0 -1  0
 0  0  0  0  0  0  0 -1  1  0  0 -1  1  0  1  1  1  1 -1 -1  0 -1 -1
 0  0  0  0  1  0  0  0 -1  0  0  1  0  0  0  0  0  0  0 -1  0  0  1
 0  0  0  0  0  1  0  0 -1  0  0  0  0  0  0  0  0  0  0 -1  0  0  0
 0  0  1  1  0  0  0  0 -1  0  0  0  0  0  0 -1 -1  0  0  0  0  0  0
#
 0  0  0  0  0  0  1  0  0  0  0  0  0 -1  0  0  0  0  0  0  1  0  0
 1  1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
-1  0 -1  1  0  0  0  0  0  0 -1  0  1  0  0  0 -1  0 -1  0  0  0  0
 0  0  0  0 -1  0  0  0  0  1  0 -1  0  0  0  0  0 -1  0  0  0  1  0
-1  0  0  0  1  0  0  0 -1 -1  0  1  1  1 -1  1 -1  0  0  0  0 -1  0
 1  0  0  0  0  0  0  0  0  1  0  0 -1 -1  1 -1  1  0  1  0  0  0  0
 0  0  0  0  0  0  0 -1  0  0  0  1  0 -1  1  0  0  1  0  0  0 -1 -1
 1  0  0  0  0  0  0  1  0  0  0  0 -1  0  0  0  0  0  0  1  0  1  0
-1  0 -1  0  0  0  0 -1  0  0  0  0  1  0  0  0 -1  0 -1  0  0  0  0
```

表 3：以 M˚artensson [M˚a24] 和 Stankovski Wagner [Sta24] 实现可读取的格式编写的 59-算法。