arXiv:2611.27896v1 [quant-ph] 20 Nov 2026

# 面向量子模拟的优化八阶 Trotter-Suzuki 方案

Marko Maleˇziˇc* 和 Johann Ostmeyer†
*Helmholtz-Institut für Strahlen- und Kernphysik, University of Bonn, Nussallee 14-16, 53115 Bonn, Germany*（日期：2026年11月23日）

在多体系统中模拟量子动力学，是量子计算机最有前景的应用之一。乘积公式是这项任务的自然选择，因为它们可直接映射为量子门；若能高效选择参数，提高公式的阶数 $n$ 便有望降低门深度。迄今为止，超过 $6^{\text{th}}$ 阶的方案仅通过二阶方案复合构造而成，而更一般的乘积公式目前只达到$6^{\text{th}}$ 阶。在这项工作中，我们将优化框架扩展至 $n = 8$阶，并找到了效率更高的复合方法以及首批 $8^{\text{th}}$ 阶一般方案。我们的公式在理论效率上优于当前最先进的方案 Morales *et al*。 [1]我们还明确推荐了两种方案。在海森堡 XZ 链的模拟中，我们推荐的其中一种方案始终优于 Morales *et al*提出的方案，不过表现最佳的方案并不总是同一种。更普遍地说，我们发现，对于拆分为多个非对易算符的哈密顿量，Trotter 分解的均匀性是判断方案性能的主要指标。为了评估这些方案在量子硬件上的应用，我们使用新的 Qiskit 插件`qiskit-omelyan`模拟含噪电路。在当前噪声水平下，蛙跳方案仍是最佳选择；但随着噪声率降低并趋近容错水平，我们推荐的方案将更具竞争力。

## I. 引言

计算多体系统的时间演化，无论在经典世界还是量子世界，都是一个极具挑战性的问题。量子多体系统的希尔伯特空间维数会随着系统规模呈指数增长，对于 $L$ 个自旋为 $2^L$，这使得在经典硬件上采用精确方法变得不可行 [2]。量子计算机通过仅以线性数量的量子比特存储状态，降低了指数级的内存成本，但要求将时间演化分解为一系列酉少量量子比特门 [3]。这种分解通常是一种近似，因此人们提出了许多方法，在保持量子门数量较少的同时实现精确构造。这些方法包括乘积公式 [4, 5], 、酉算符线性组合（LCU） [6], 、量子信号处理（QSP） [7], 以及随机化方法 [8, 9]。对于经典系统，辛积分器遵循略微扰动的哈密顿量的精确动力学，使能量误差保持有界，而不是逐渐漂移 [10]。即使对于会将任何积分误差呈指数放大的混沌动力学，这一性质仍然成立；因此，没有任何积分器能够在很长时间内跟踪单条轨迹。

乘积公式也称为 Trotter-Suzuki方案，因其无需辅助量子比特或块编码，且每个因子都可直接映射为量子门而备受青睐。对于双部分哈密顿量 $H = A + B$，其时间演化算符 $U$ 可用乘积近似

$$U(t) = e^{-itH} \approx [S_n(h)]^{t/h}, \tag{1}$$

其中 $S_n(h)$ 是一个阶数为 $n$ 、步长为 $h$ 的乘积公式，由指数算符 $e^{-\mathrm{i}a_j hA}$ 和 $e^{-\mathrm{i}b_j hB}$ 组成，其方案参数为 $a_j$ 和 $b_j$。这些因子都是幺正的，因此该近似也严格幺正。若 $A$ 和 $B$ 各自都是对易局域项之和，这些因子还可以精确分解为少量量子比特门。将同样的拆分应用于经典哈密顿流，可得到辛积分器 [10], ，因此这些方案同样适用于经典模拟。阶数 $n$ 对应于全局误差随步长 $\mathcal{O}(th^n)$ 的缩放。尽管线性组合酉算符（LCU）和量子信号处理（QSP）的渐近缩放表现更优，但对于实用规模的系统，乘积公式仍具有竞争力，因为其实际误差远小于理论界限所预示的误差 [11, 12]。因此，它们是早期容错量子计算机的理想候选方案 [13]。

在当前的量子模拟中，通常采用最低的阶数 $n = 1$, 2，因为每步所需的量子门最少。然而，要使全局误差达到目标精度 $\epsilon$ ，需要 $t/h \propto t^{1+1/n}\epsilon^{-1/n}$ Trotter步 [12]。提高阶数后，步数与 $t$几乎呈线性关系，而对精度的依赖则会减弱。更少的Trotter步可以构成更浅的量子电路，从而减少噪声累积，并支持更长时间的时间演化。另一方面，高阶公式每步包含更多指数算符，但可通过调节额外的方案参数来降低误差前因子，从而提高效率。

这类方案传统上由二阶蛙跳方案复合构造，可以递

* malezic@hiskp.uni-波恩.de
† ostmeyer@hiskp.uni-bonn.de

归地构造，正如 Suzuki 所提出的 [14, 15], ，也可以按照 Yoshida 的方法进行数值构造 [16]。目前最先进的复合方法是 Morales 等人提出的 [1] 阶方案$n = 8$，以及 10 阶方案。Omelyan 等人 [17, 18] 则直接推导乘积公式的误差项，从而摆脱了对复合方法的限制，并能够构造出更高效的方案。人们已找到越来越好的通用方案：Omelyan 等人最初提出了 6 阶方案，随后 Blanes和Moan 提出了改进方案 [19], ，而我们最近的前期工作也有所进展 [20]。然而，超过 $6^{\text{th}}$ 该阶数后，需要消除的误差项数量会迅速增加，迄今尚未构造出通用方案。关于更全面的综述，请参阅近期关于 Trotter 分解及其他时间演化方法的综述 。

本文将基于 Omelyan 方法的框架扩展到 $n = 8$阶，找到了新的复合方式以及该阶下首批通用方案，并在无噪声和含噪量子模拟中对它们进行了基准测试。第 II节介绍 $8^{\text{th}}$ 阶扩展、其解的结构，以及我们如何改进最小化程序以探索这些解。此外，我们还讨论了如何将框架调整为适用于 $n \geq 8$阶的复合方式。第 III节将我们方案的理论效率与 Morales 等人提出的方案进行比较。 [1] 此外，我们还针对海森堡 XZ 自旋链进行了数值实验，考察了两种算符排序方式，并研究了误差随计算成本的变化规律。表 III 和IV 分别给出了高效复合方案和通用方案的参数，附录 A则介绍了其他方案。最后，我们在第IV 节利用新开发的插件 `qiskit-omelyan` [22] 模拟含噪电路，以评估高阶乘积公式何时能够改善量子模拟。我们在第 V节总结全文，并讨论未来工作的设想。

## II. 框架扩展至 $8^{\text{th}}$ 阶

截至目前，阶数为 $n \geq 8$ 的乘积公式都是由低阶方案复合构造而成的，方法包括 Suzuki 提出的递归方式 [14, 15], ，或 Yoshida 方法所采用的数值优化方式 [16]。Omelyan 等人提出的另一种方法等 [18], 直接求解分裂系数的阶条件，迄今为止仅用于阶数 $n = 6$。在先前的工作中 [20, 23], ，我们基于这种方法构建了一个用于构造优化方案的框架。在此，我们将该框架扩展至阶数 $n = 8$，讨论流形结构，并说明对最小化过程所做的一些改进。

### A. 误差项的构造

要从头构造任意阶数的方案，我们使用贝克-坎贝尔-豪斯多夫（BCH）公式对其进行展开。对于推导，只需假设哈密顿量由两个不对易算子 $H = A + B$ 构成，因为推广到 $\Lambda > 2$ 个阶段的方法在 [24] 中已有现成结果。因此，我们作如下设定

$$\begin{aligned}&\exp\left[(A+B)h + \sum\nolimits_n O_{n-1}h^{n-1}\right] = \\ &= e^{\mathsf{a}_1 hA}e^{\mathsf{b}_1 hB}e^{\mathsf{a}_2 hA}\cdots e^{\mathsf{b}_q hB}e^{\mathsf{a}_{q+1} hA} = \\ &= e^{c_1 hA}e^{c_1 hB}e^{d_1 hB}e^{d_1 hA}\cdots e^{c_q hA}e^{c_q hB}e^{d_q hB}e^{d_q hA},\end{aligned} \tag{2}$$

其中，$\mathsf{a}_j$ 和$\mathsf{b}_j$ 是 Omelyan 记法中的方案参数，$h$ 是时间步长，$O_{n-1}$ 表示误差项。每个序列 $e^{\mathsf{a}_j hA}e^{\mathsf{b}_j hB}e^{\mathsf{a}_{j+1} hA}$ 构成一个周期，因此该方案由 $q$ 个（相互重叠的）周期组成。此外，我们假设乘积公式是对称的，即 $\mathsf{a}_j = \mathsf{a}_{q+2-j}$ 且 $\mathsf{b}_j = \mathsf{b}_{q+1-j}$，这使得 $q+1$ 个参数可以自由选取。此时，指数中只会出现 $h$ 的奇次幂，因此求和仅涉及偶数阶 $n = 2$，4，6，.....，且方案的阶数为偶数。我们还纳入了采用斜坡方法、参数为 $c_i$、 $d_i$的乘积公式；两种记法之间的转换见参考文献 [24]。误差项 $O_{n-1}$由系数与交换子相乘构成，这些交换子自然地出现在 BCH 展开中。唯一的例外是

$$O_1 = (\nu - 1)A + (\sigma - 1)B, \tag{3}$$

其中 $\nu = \sum_j \mathsf{a}_j = 1$ 和 $\sigma = \sum_j \mathsf{b}_j = 1$ （或等价地$\sum_j($ $c_j + d_j) = 1$）给出了前两个约束条件。

误差项的结构是该框架的基础，我们此前已在参考文献 [20] 中将其推导至 $O_7$ 。由于完整计算篇幅较长，我们现在概述 $O_9$ 误差项的推导。推导过程见我们存放于代码库 [26], 中的 Mathematica [25] 笔记本（Efficient Trotterization.nb），结果汇总在其中的笔记（Recursive formulae.pdf）中。为使推导过程能够自由获取，我们还基于开源库 SymPy [27]提供了 Python 移植版本，作为软件包 pybch。

在 $n = 8$ 阶时，误差项 $O_9$ 位于由 $A$ 和 $B$ 生成的自由李代数的 9 次齐次分量中。消除由反对称性和雅可比恒等式导致的线性相关后，还剩下 56 个线性无关的对易子。与我们先前的工作一样，我们采用 Arnal 等人讨论的右嵌套对易子基。 [28], 在交换 $A \leftrightarrow B$ 下呈对称排列，使得 $F_{57-k}$ 可由 $F_k$ 交换 $A$ 和 $B$得到。该基根据$A$项和 $B$项的数量分为 8 个子空间，其维数分别为 1、4、9、14、14、9、4、1。我们从每个子空间中列出一个元素

$$
\begin{aligned}
F_1 = [A^8B] &\longleftrightarrow [B^8A] = F_{56},\\
F_2 = [A^3BA^4B] &\longleftrightarrow [B^3AB^4A] = F_{55},\\
&\vdots\\
F_6 = [A^3B^2A^3B] &\longleftrightarrow [B^3A^2B^3A] = F_{51},\\
&\vdots\\
F_{15} = [A^2BA^2B^2AB] &\longleftrightarrow [B^2AB^2A^2BA] = F_{42},\\
&\vdots
\end{aligned}
\tag{4}
$$

为便于阅读，我们将对易子写成紧凑的右嵌套形式，例如 $[A,[A,B]] = [A^2B]$。

固定交换子基后，误差项现在可以在该基下展开为

$$
O_9 = \sum_{k=1}^{56} \epsilon_k F_k, \tag{5}
$$

其中系数 $\epsilon_k$ 是方案参数 a$j$ 和b$j$的多项式。按照参考文献 [20], 第 2.2 节中概述的方法，我们推导出关于循环数 $q$ 的递归公式，用以将 $\epsilon_k$ 表示为方案参数的函数。递归过程交替应用算子 $A$ 和算子 $B$对应的步长，而所执行的步数则由表达式 $\epsilon^{(i_A,i_B)}{}_{k,A}$ 中的指标 $i_A, i_B$ 表示。在 $k = 56$ 处，最简单的公式只为 $A$中的一个步长增加一项，

$$
\epsilon_{56,A}^{(i,i-1)} = \epsilon_{56,B}^{(i-1,i-1)} + \tilde{\epsilon}_{56} a_i \left(\sigma^{(i-1)}\right)^8. \tag{6}
$$

其中，我们用前一篇论文 [20], 中的记号表示方案参数 $a_i, b_i$；这些记号可以转换为 Omelyan 使用的记号。

分数系数 $\tilde{\epsilon}_k$ 由 BCH 展开得出。例如，最后一个交换子$F_{56}$ 的系数为

$$
\tilde{\epsilon}_{56} = -\frac{1}{1209600} \tag{7}
$$

其余基底及其前因子可在我们代码库 [26] 的笔记中找到。

系数 $\sigma^{(i-1)} = \sum_j^{i-1} b_j$ 汇总了参数 $b_j$，直至递归步长 $i-1$，最终得到 $\sigma$ 中所示的形式 $O_1$ (3)。其他表达式 $\epsilon_{i\,A}^{(i,i-1)}$，包含更多项和更高阶系数。这些表达式会变得很冗长，详情请参阅我们的笔记 [26]。

将基与系数相结合，得到误差项 $O_9$ （方程（5））, 由此我们可以将阶数为 $n = 8$ 的误差函数定义为其系数的欧几里得范数，

$$
\mathrm{Err}_8 = \sqrt{\sum_{k=1}^{56} |\epsilon_k|^2}. \tag{8}
$$

表 I. 阶数为 $n \le 12$的对称方案的阶条件计数。在阶数 $n$下，$O_{n-1}$ 的系数除较低阶的系数外也必须为零，它们共同给出约束条件的数目 $T_S$ （针对一般对称方案）。对于对称复合（9），这些约化为 $T_{SS}$ 个条件 [29]。主导误差项 $O_{n+1}$ 的系数因此见下一行。每一阶数的周期范围从最小复合 $q = 2T_{SS} - 1$开始，并在下一阶数的周期范围之前结束。一般解要求$q \ge T_S - 1$，而对于 $n \ge 10$ （灰色），该要求超出周期范围，因而只剩下复合方法。 $n \le 10$ 的系数与参考文献 [28], 一致，我们将其扩展到 $n = 12$。

| 阶数 $n$ | 系数 | $T_S$ | $T_{SS}$ | 周期 $q$ | 通用 $q$ |
|---|---|---|---|---|---|
| 2 | 2 | 2 | 1 | 1–2 | $\ge 1$ |
| 4 | 2 | 4 | 2 | 3–6 | $\ge 3$ |
| 6 | 6 | 10 | 4 | 7–14 | $\ge 9$ |
| 8 | 18 | 28 | 8 | 15–30 | $\ge 27$ |
| 10 | 56 | 84 | 16 | 31–66 | $\ge 83$ |
| 12 | 186 | 270 | 34 | 67–146 | $\ge 269$ |

因此，构造 $8^{\text{th}}$ 阶方案就成为一个优化问题：在满足阶条件 $O_1 = O_3 = O_5 = O_7 = 0$的约束下，使 $\mathrm{Err}_8$ 最小。不过，在进行优化之前，我们必须了解哪些参数区域至少存在解，这需要更加仔细地考察解流形的结构。

### B. 阶数 n = 8 流形结构

要构造 $8^{\text{th}}$ 阶的方案，在我们的框架中需要满足总计 28 个阶数约束，这通常要求 28 个方案参数。任意阶数下约束条件的总数 $T_S$ 由 Witt公式给出，正如 McLachlan 所展示的那样， [29], 我们在表 $n = 12$ 中列出了不超过该阶数的阶I。由于一个具有 $q$ 个循环的方案拥有 $q+1$ 个参数，因此通解最先出现在 $q = T_S - 1$，这意味着 $q = 27$ 在$8^{\text{th}}$ 阶时。

然而， $8^{\text{th}}$ 已知阶数方案所需的循环少得多，从 $q = 15$开始。这些是 Yoshida 类型的 [16] 复合方法，将 $q$ 不同阶数的公式 $S_2$ 以不同的步长串联成回文乘积，

$$
S_n(h) = \prod_{j=1}^{q} S_2(w_j h), \qquad w_j = w_{q+1-j}, \tag{9}
$$

其中，我们将标准 Leapfrog公式定义为：

$$
S_2(h) = e^{\frac{1}{2}hA} e^{hB} e^{\frac{1}{2}hA} \tag{10}
$$

在 $8^{\text{th}}$ 阶时，此类方案最初由 Yoshida [16], 构造，随后由 McLachlan [29], Ka- han 和Li [30], Sofroniouand Spaletta [31], 改进，最近则由 Morales 等人 [1] 提出改进。这些复合方法

表 II. $8^{\text{th}}$ 阶数解流形的维数，由雅可比矩阵的零方向数量给出 $J_{ij}$。在复合区域中，只存在复合方法（9），且其子流形每隔两个周期便增加一个自由方向 $q$，因为每对周期都会增加一个参数。在一般区域中，复合方法之外也存在解，其流形每个周期都会增加一个自由方向。复合子流形也持续增长，并始终具有更高的维数。

| 区域 | 复合区域 | | | | | | 一般区域 | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 周期数 $q$ | 15、16 | 17, 18 | 19, 20 | 21, 22 | 23, 24 | 25, 26 | 27 | 28 | 29 | 30 |
| 复合维数。 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 6 | 7 | 7 |
| 一般维数 | – | – | – | – | – | – | 0 | 1 | 2 | 3 |

自然地包含在我们的框架中。在 ramp 记法（2）中，复合方法对应于 $c_j = d_j$，与回文结构结合后便形成镜像对称性 $c_j = c_{q+1-j}$ （见表 III）。对于复合方法，阶数约束不再彼此独立。McLachlan [29]证明它们可化简为$T_{SS}$ 条件，我们也在表I中列出了这些条件。在 $8^{\text{th}}$ 阶时，仅剩下 $T_{SS} = 8$ 个条件；由于回文乘积中只有每隔一个步长才会引入一个新参数，因此复合方法需要 $q \geq 2T_{SS} - 1 = 15$。

阶数 $n = 8$ 周期范围因此分为两个区域。在 $q = 27$ 个周期以下，参数数量太少，无法一般性地满足所有约束条件，因此只有 $S_2$复合方法存在，这也是我们称其为复合区域的原因。从 27 个周期开始，除复合方法外也存在其他解；复合方法此时构成一个低维子集，我们称之为一般区域。在更高阶数下，一般区域消失，因为 $T_S$ 增长速度远快于 $T_{SS}$，而一般解只有在已经能够构造下一阶的复合方法后才会出现（见表 I）。下文将在 $8^{\text{th}}$阶数下通过数值计算验证这一图景。

上述计数还可预测解流形的维数。在复合子流形上，从 $q = 15$ 开始，每隔一个周期就会出现一个新的自由方向；而在一般区域中，从 $q = 27$ 开始，每个周期都会出现一个新的方向。为验证这一点，我们考察雅可比矩阵

$$J_{ij} = \frac{\partial C_i}{\partial c_j}, \tag{11}$$

其中 $C_i$ 表示受约束误差项的系数 $O_1$、……、$O_7$。 $J_{ij}$ 的奇异值中为零的数量，决定了解流形的局部维数，也就是我们可以进行优化的零方向数量。遍历周期数 $q$，我们在非复合点处计算雅可比矩阵，从而得到一般流形的维数。对于复合解，我们将其限制在保持镜像对称性 $c_j = c_{q+1-j}$的方案上，从而得到复合子流形的维数。我们在表 II 中总结了研究结果，相关分析见代码库 `Jacobian on manifold.nb`中的 Mathematica 笔记本（ [26]）。

正如预测的那样，复合子流形的维数会随着每个奇数 $q$增加一维，而一般流形的维数从 $q = 27$ 处的零开始，随每个周期递增。由于镜像对称性使大多数约束变得冗余，复合子流形在整个一般区域内的维数始终高于一般流形。因此，这两个流形作为解集中的独立分支共存，且至多只在孤立点相交。在实践中，这使得在一般区域内难以找到复合方法，因为随机初始参数会破坏镜像对称性，最小化器最终会收敛到一般解。另一方面，在复合区域内不存在其他解，因此每次运行收敛后都会得到复合方法。因此，我们调整了框架，从一开始就施加镜像对称性，从而能够在一般区域内找到复合方法。正如我们在第 IIC 节所示，这也改善了复合区域内搜索的条件性。

在 $q$ =17 和 18 处的一维复合子流形也可以直接观察到。如果我们只求解阶条件而不优化 $\text{Err}_8$, ，所得解会落在子流形上的任意点。我们根据它们的理论误差 $\text{Err}_8$ 以及它们与等格式参数点$c_j = 1/2q$之间的距离 $\bar{x}$ 对其进行分类，

$$\bar{x}^2 = 2\sum_{j=1}^{q}\left(c_j - \frac{1}{2q}\right)^2, \tag{12}$$

其中，因子 2 用于计入下降斜坡参数 $d_j = c_{q+1-j}$。虽然方程（12）定义了 2-范数距离，也可以采用 1-范数定义，但我们发现两者之间没有显著差异。如图1所示，解描绘出彼此不同的曲线，这与存在一个自由方向的预期相符。在 $q = 18$处找到的解较少，因此曲线更难区分。事实上，这些曲线是我们最初发现复合子流形的依据。沿自由方向优化 $\text{Err}_8$，即可得到每条曲线的最小值。在循环数更多时 $q \geq 19$，子流形的维数更高，其在这两个分类指标上的投影不再构成彼此不同的曲线。下文将说明，为了在阶数$n = 8$下高效地最小化这类曲线，我们如何调整优化程序。

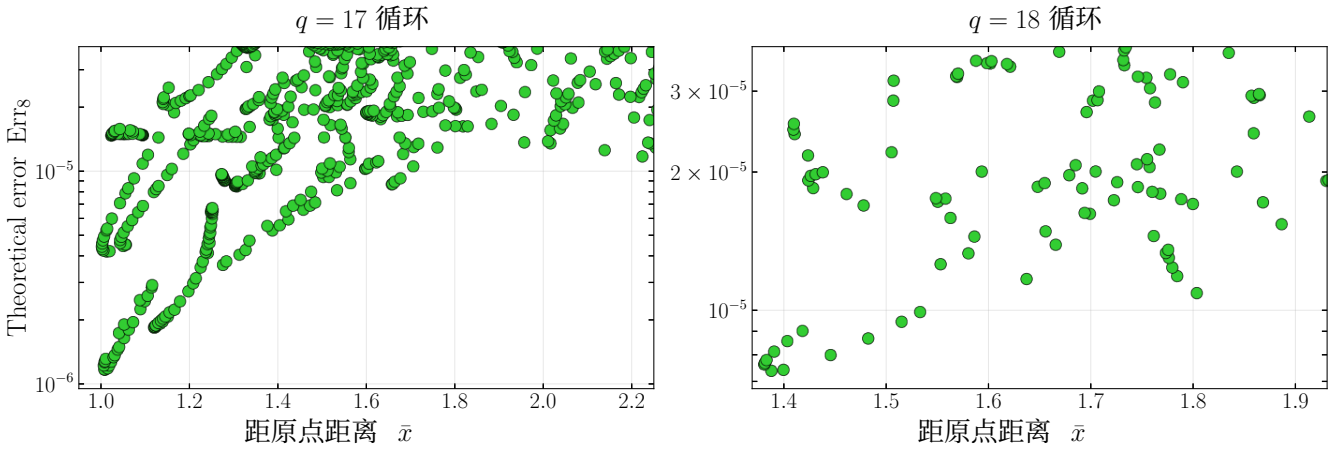



图 1. $8^{\text{th}}$ 阶条件在 $q = 17$ （左）和 $q = 18$ 循环（右）的解，按其理论误差 $\mathrm{Err}_8$ （8）以及与等格式参数点的距离 $\bar{x}$ （12）分类。这些解位于曲线上，这些曲线描绘出一维复合子流形（见表II）。

### C. 优化改进

我们首先定义 $\chi^2$ 函数，从而表述最小化问题：

$$\chi^2(\mathbf{p}) = \sum_{n\geq 2} \omega_n \mathrm{Err}_n^2(\mathbf{p}) = \mathbf{y}(\mathbf{p})^\dagger \mathbf{W} \mathbf{y}(\mathbf{p}), \qquad (13)$$

其中 $\mathrm{Err}_n$ 与 $n < 8$ 对应于阶数约束。向量 $\mathbf{p}$ 存储自由格式参数，$\mathbf{y}$ 为 $O_3$，……，$O_9$的系数，对角矩阵 $\mathbf{W}$ 为权重 $\omega_n$ [20]。与之前一样，我们最小化 $\chi^2$，采用列文伯格-马夸特算法 [32, 33]，，该算法通过阻尼参数 $\lambda$在高斯-牛顿法与梯度下降之间进行插值。每个步长 $\mathbf{h}_{\mathrm{LM}}$ 通过求解得到

$$\left[\mathbf{J}^\dagger \mathbf{W} \mathbf{J} + \lambda\, \mathrm{diag}(\mathbf{J}^\dagger \mathbf{W} \mathbf{J})\right] \mathbf{h}_{\mathrm{LM}} = -\mathbf{J}^\dagger \mathbf{W} \mathbf{y}(\mathbf{p}), \qquad (14)$$

其中 $\mathbf{J} = \partial \mathbf{y}/\partial \mathbf{p}$ 现在还包括主导误差项的系数。如图2所示，雅可比矩阵随循环次数增加而愈发病态，下文将予以说明。阻尼项使公式（14）中的矩阵保持可逆，只要 $\lambda$ 以 $\lambda_{\min}$为下界。在阶数 $n = 8$时，我们将该下界降至 $\lambda_{\min} = 10^{-12}$，这仍能对系统进行正则化，但对步长的阻尼弱于之前。

第二个挑战是计算成本，因为参数数量和误差系数数量都会增长，并且每个步长都会变得更加昂贵。由于列文伯格-马夸特算法是一种局部方法，我们必须运行许多初始条件才能找到大量极小值，因此每一个都必须快速收敛。如参考文献 [20], 所述，我们分两个阶段进行最小化。我们首先最小化完整的 $\chi^2$，以到达高效格式所在的区域，然后精确施加阶数约束通过仅最小化受约束项来实现，并从第一阶段的结果出发。对于第一阶段，我们发现以下权重效果良好，

$$\omega_2 = 1, \quad \omega_4 = 10^4, \quad \omega_6 = 10^7, \quad \omega_8 = \left[10^1, 10^3\right], \qquad (15)$$

其中，我们给出 $\omega_8$的取值范围，因为较小的权重收敛更快，但可能导致方案效率较低。在 $n \leq 6$阶时，我们在第二阶段冻结了第一阶段的近似 Hessian $\mathbf{J}^\dagger\mathbf{W}\mathbf{J}$，使解保持在高效区域附近。然而，在$n = 8$阶时，这会妨碍第二阶段的收敛，因此我们改为重新计算 Hessian，从而增加了收敛的初始条件数量。

在我们的框架中，进一步改善收敛的最后一项举措是加入测地加速度 [34]。将步长 $\mathbf{h}_{\mathrm{LM}}$ 视为沿测地路径的速度$\mathbf{v}$，我们加入加速度$\mathbf{a}$带来的二阶修正，其满足

$$\left[\mathbf{J}^\dagger \mathbf{W} \mathbf{J} + \lambda\, \mathrm{diag}(\mathbf{J}^\dagger \mathbf{W} \mathbf{J})\right] \mathbf{a} = -\mathbf{J}^\dagger \mathbf{W} \mathbf{f}_{vv}, \qquad (16)$$

使用与公式 (14) 中相同的矩阵。只有在**改进后的步长$\mathbf{h}_{\mathrm{geod.}} = \mathbf{v} + \frac{1}{2}\mathbf{a}$满足 $2|\mathbf{a}|/|\mathbf{v}| \leq 0.75$时，才接受该步长，以防加速度主导步长。方向二阶导数$\mathbf{f}_{vv}$ 通过沿归一化方向 $\hat{\mathbf{v}} = \mathbf{v}/|\mathbf{v}|$的前向有限差分，

$$\mathbf{f}_{vv} \approx \frac{2|\mathbf{v}|^2}{\delta} \left[\frac{\mathbf{y}(\mathbf{p} + \delta\, \hat{\mathbf{v}}) - \mathbf{y}(\mathbf{p})}{\delta} - \mathbf{J}\hat{\mathbf{v}}\right], \qquad (17)$$

这只需额外进行一次求值，因为$\mathbf{y(p)}$和$\mathbf{J}$ 已经计算完毕。归一化可使扰动得到良好分辨，即使在接近收敛时步长变小也是如此。我们使用相同的

有限步长 $\delta$ 作为雅可比矩阵的步长，选择为 $\delta = \epsilon_{\text{mach}}^{1/3}$，以平衡截断误差和舍入误差。加入测地加速度可适度改善收敛性，并帮助找到该阶数下的一些最高效格式。

然而，这些改进仅在循环数较少时效果良好。在较大的 $q$下，无约束最小化器难以找到解，尤其是在复合区域中；图 2 中的数量在 $q = 21$处达到峰值，随后迅速下降，在$q = 26$时仅剩三个方案。我们通过计算雅可比矩阵 $\kappa$ J 的条件数**对此下降现象进行了研究**。

$$\kappa = \frac{s_{\max}}{s_{\min}}, \tag{18}$$

它是最大 $s_{\max}$ 与最小非零奇异值之比 $s_{\min}$。图 2 的下方面板显示，雅可比矩阵的病态程度逐渐加剧，并在 $q = 26$处达到峰值。由于最小化器作用于所有 $q$ 方案参数，因此必须抑制所有破坏镜像对称性的方向，同时收敛到雅可比矩阵秩亏的一组点上。在一般区域中，无需强制施加这种对称性，因此条件数趋于平稳，方案数量则有所增加，因为最小化器可以收敛到一般流形上的任意位置。

我们试图通过施加第 $c_j = c_{q+1-j}$ 节中讨论的镜像对称性来解决条件性问题。IIB这一对框架的改编大致将参数数量减半，并将我们的搜索范围限制为仅包含复合格式。去除破坏对称性的方向应能显著改善雅可比矩阵的条件性。事实上，采用改编框架后，对于所有 $\kappa \sim 10^4$–$10^5$，中位条件数都保持在 $q$，比一般框架低最多五个数量级，并且极小值的数量在每个奇数 $q$阶。

对复合方法进行改编，使我们能够更深入地探索复合子流形。它在复合区域中，为循环数较大的情形带来了新的方案，也使我们能够找到在一般区域中持续存在的复合方法。我们将在第III节讨论这些结果。在 $n = 10$ 阶时，通用方案仅在 $q \geq 83$ 才会出现（见表I），因此，在实际可行的循环数范围内只能找到复合方法，这种改编也就至关重要。不过，这部分工作的主要难点在于推导误差项，我们将其留待未来研究。

## III. $8^{\text{th}}$ 阶 Trotter 分解结果

本节将介绍我们的最小化器找到的 $8^{\text{th}}$ 阶方案，首先从其理论效
率。随后，我们在海森堡模型上通过数值实验加以验证，并以 $n = 8$处我们推荐的方案参数作为结论。

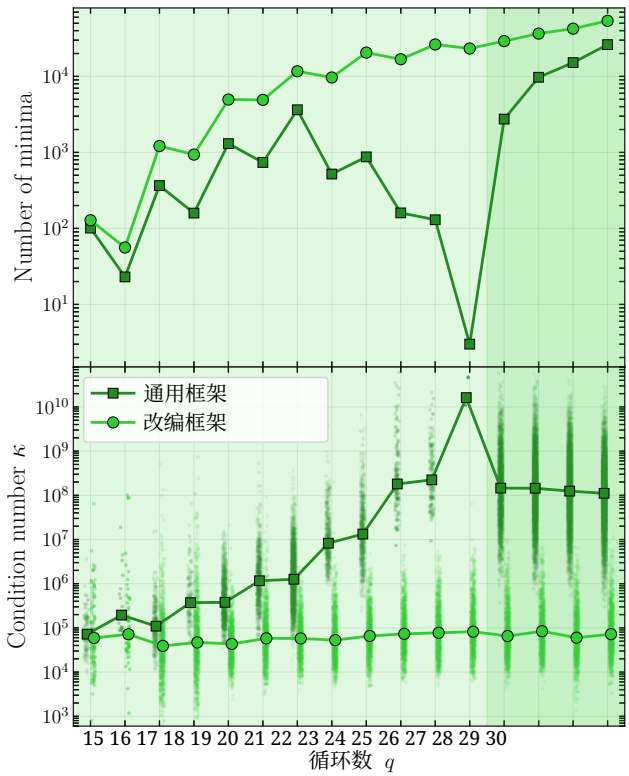



图 2. 不同极小值的数量（上图）和所发现解中雅可比矩阵的条件数 $\kappa$ (18) 及其中位数（下图），对应各个循环数 $q$ 和阶数 $n = 8$。复合区域和一般区域（见表 II）分别以浅绿色和深绿色阴影标出。图中比较了一般框架与施加镜像对称性后的复合调整方案 $c_j = c_{q+1-j}$。我们的通用优化器难以在循环数较高时找到复合方案，因为雅可比矩阵的条件数会急剧增大。调整方案通过将搜索限制在参数和阶数条件较少的复合方案中，解决了这一问题，从而稳定了条件性并增加了极小值的数量。请注意，在循环数较高时，极小值的数量几乎肯定只是一个下限，而且为计算条件数，后续仅分析了这些极小值中的一部分。

### A. 理论效率

为比较循环数不同的各方案 $q$，我们固定计算成本，如参考文献 [20]。由于每个步长由 $q$ 个循环组成，成本按 $q/h$ 缩放，我们通过选取步长 $h \propto q$ 将其固定。鉴于全局误差按 $h^n \text{Err}_n$ 缩放（见公式（8）），我们定义效率

$$\text{Eff}_n = \frac{1}{q^n \text{Err}_n}, \tag{19}$$

其中只比较阶数相同的方案 $n$，因为对于每个阶数，误差随 $h$ 的不同幂次缩放。

最小化后，我们分析找到的极小值，并按效率对其进行排序。在图3中，我们比较了我们的

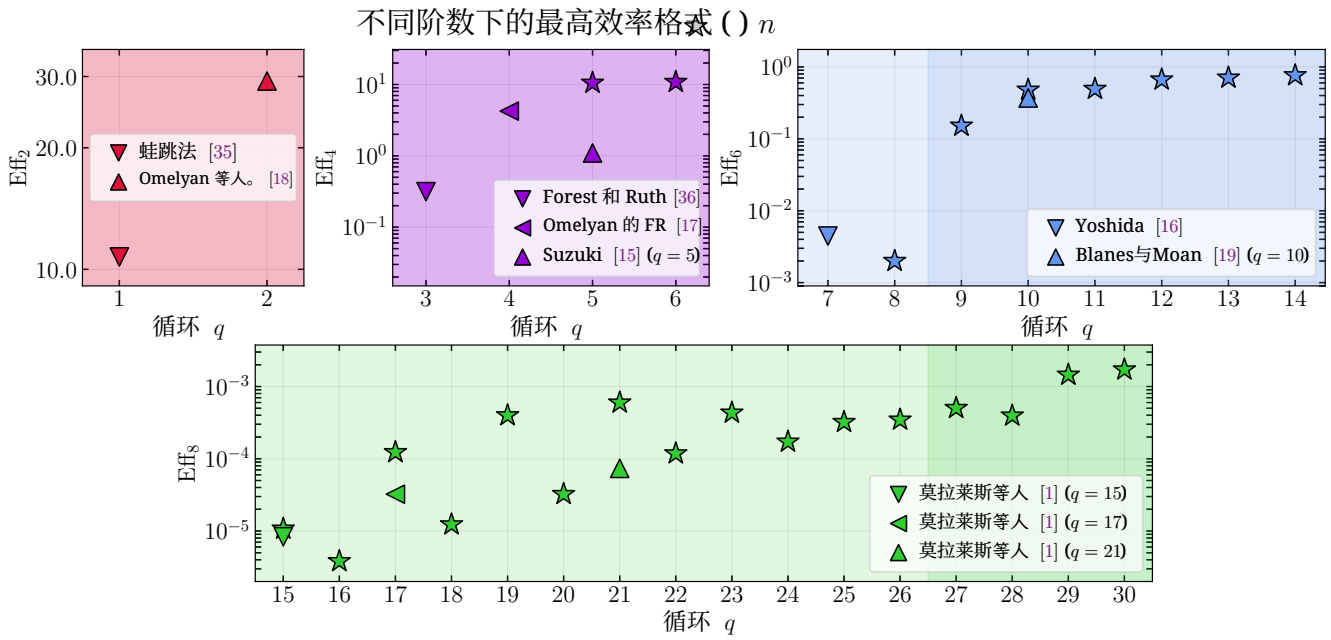



图 3. 文献中各方案（三角形）以及我们针对每个循环数提出的最高效方案（星形）的理论效率 $\mathrm{Eff}_n$ (19) $q$，适用于阶数 $n \leq 8$，扩展了我们先前的结果 [20] 至 $n = 8$阶。在 $n = 6$阶时，8，复合区域和一般区域分别以浅色和深色阴影标示。在 $n = 8$的复合区域内，我们的方案显著优于 Morales 提出的方案 *et al*。 [1], 尤其是在 $q = 21$时。对于偶数 $q$阶，其效率通常低于相邻的奇数 $q$ 阶，因为增加一个周期并不会为复合方法增加自由参数。在一般区域内，效率进一步提高；在 $q = 29$ 阶时，30 个周期的最高效方案是复合方法。

最高效格式与文献中的方案，以及参考文献 [20] 中阶数为 $n \leq 6$ 的方案。星号表示我们针对每个循环数 $q$ 找到的最高效方案，三角形表示文献中已知的最佳方案。在 $6^{\mathrm{th}}$ 和 $8^{\mathrm{th}}$ 阶时，复合区域和一般区域分别以浅色和深色阴影标示。本研究的新结果显示在底部面板中，其中所有此前已知的方案都是具有奇数 $q \leq 21$ [1, 16, 30, 31] 的复合方法；我们展示了其中由 Morales等人 [1] 提出的最新方案。我们的框架将其扩展到偶数 $q$，并进一步扩展到 $q = 30$ 之前的 $q = 21$，其中包括 $q \geq 27$ 阶的一般方案，而这些方案无法通过复合方法构造。

在 $8^{\mathrm{th}}$ 阶数下，我们的框架相比已知方案有了显著改进，尤其是在 $q = 21$方面，此时我们的最佳方案比 [1]大约高效 8 倍。在复合区域，奇数 $q$ 通常比相邻的偶数值产生更高效的方案。这一点可由第 IIB节预期得到，因为偶数 $q$ 会增加一个周期，从而增加成本，却不会为复合方法增加自由参数。在一般区域，复合方法在 $q = 27$、29 和 30 阶时仍能提供最高效的方案，因为其子流形的维数大于一般流形的维数，因而为优化提供了更大的自由度（见表 II）。只有在 $q = 28$阶时，偶数 $q$ 会对复合方法造成不利影响，此时通用方案才是最高效的。由于两个流形的维数都会随 $q$ 增加，我们的搜索在较大的 $q$阶时可能并不完整，因此那里可能存在更高效的方案。理论上，不过，数值效率并不是衡量良好性能的唯一指标，因此在推荐方案参数之前，我们还会进行数值基准测试。

### B. 数值性能

理论效率衡量方案的主导误差，且不依赖于所模拟的系统，因此无法与实际模拟中观测到的误差完全对应。误差函数 $\mathrm{Err}_n$ 对主导误差项中的所有嵌套对易子 $O_{n+1}$ 赋予相同权重，而它们的实际贡献取决于 $A$ 和 $B$ 这两个算符，它们属于所模拟的哈密顿量。此外，具有实际物理意义的哈密顿量通常由两个以上不对易的部分组成；通过参考文献 [24], 所述的推广将这些方案应用于此类哈密顿量时，会引入 $\mathrm{Err}_n$未能捕捉到的其他对易子。最后，近期研究表明，相邻步长产生的误差可能相互干涉，因此误差随时间的累积也取决于方案的结构 [37, 38]。一般而言，模型依赖性无法消除，但我们可以研究方案参数在实践中如何影响误差，这正是本节关注的重点。

在这项工作中，我们对具有 $L = 6$ 个格点和开放边界条件的海森堡 XZ 自旋链进行建模，其哈密顿量如下

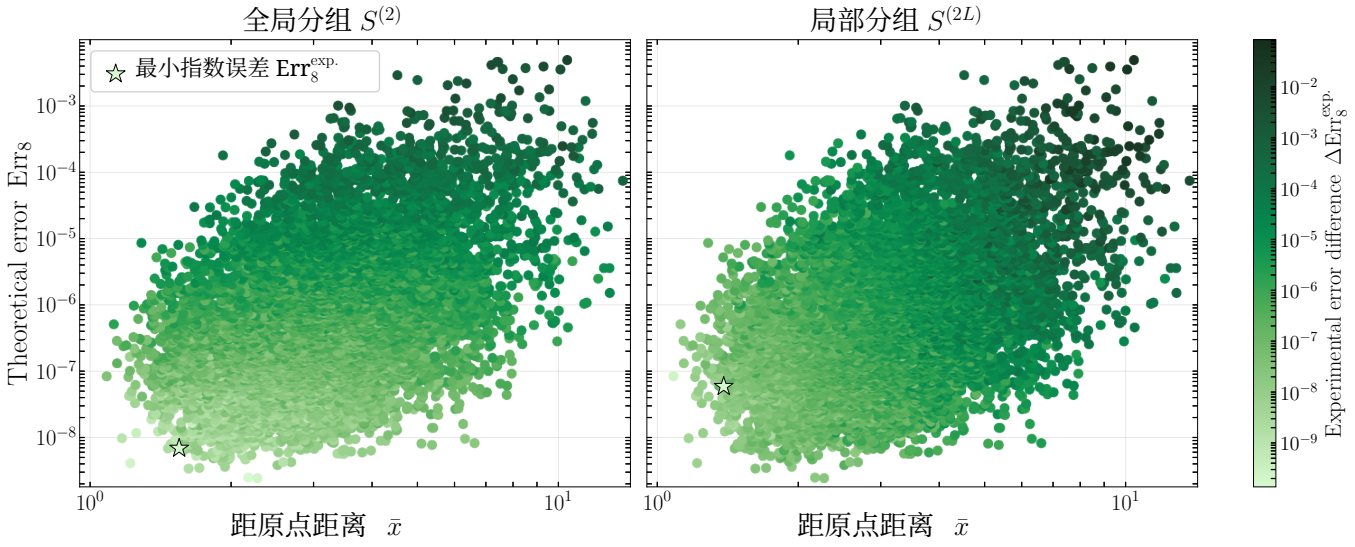



图 4. 最大循环数下的一般分解方案比较 $q = 30$，阶数为 $n = 8$。根据理论误差 $\mathrm{Err}_8$ (8)、与均匀方案参数的距离 $\bar{x}$ (12) 以及数值误差之差 $\Delta\mathrm{Err}_8^{\mathrm{exp.}}$ (24)对这些方案进行分类。后者针对具有开放边界条件的海森堡 XZ 自旋链（$L = 6$）计算，并采用两种乘积公式排序进行时间演化，直到 $t = 10$，且计算成本固定为 $q/h = 10^3$。在全局方案$S^{(2)}$（左图）中，数值误差随理论误差的变化表现更优；而在局部分组 $S^{(2L)}$（右图）中，实际性能主要与到原点的距离相关 $\bar{x}$。

$$H = \sum_{i=1}^{L-1} J\left(\sigma_i^x \sigma_{i+1}^x + \sigma_i^z \sigma_{i+1}^z\right) + \sum_{i=1}^{L} h_i \sigma_i^z, \tag{20}$$

其中，我们将格点 $i$ 处的泡利矩阵记为 $\sigma_i^\alpha$，并将耦合设为单位值 $J = 1$。为打破模型的对称性，我们沿 $z$ 方向加入随机磁场，其中 $h_i$ 从 $[-0.1, 0.1]$ 中均匀抽取，并在所有模拟中保持不变。

对于许多局部（不）对易算符，可以通过多种不同方式对哈密顿量进行拆分 [39]。由于每一项至多作用于两个相邻位置，因此可以直接将其实现为局部门。沿链键逐个分组是一种自然的分组方式（$i,\ i+1$），

$$\begin{aligned} S_n^{(2L)}(h) = {} & e^{-ic_1 h H_1^x} e^{-ic_1 h H_1^z} e^{-ic_1 h H_2^x} e^{-ic_1 h H_2^z} \cdots \\ & \times e^{-id_q h H_2^z} e^{-id_q h H_2^x} e^{-id_q h H_1^z} e^{-id_q h H_1^x}, \end{aligned} \tag{21}$$

其中 $H_i^\alpha = J\sigma_i^\alpha \sigma_{i+1}^\alpha + \delta_{\alpha z} h_i \sigma_i^\alpha$表示局部项，乘积公式采用斜坡记号，这种记号可自然推广到 $\Lambda$阶段 [24]。由于该乘积公式由 $\Lambda = 2L - 1$ 个阶段组成，我们将其称为局部排序 $S^{(2L)}$。另一种方法利用了对易性：由于所有 $H_i^x$彼此对易，所有 $H_i^z$也彼此对易，因此它们可以归并为仅有 $\Lambda = 2$个阶段，

$$\begin{aligned} S_n^{(2)}(h) = {} & \left(\prod_{i=1}^{L} e^{-ic_1 h H_i^x}\right)\left(\prod_{i=1}^{L} e^{-ic_1 h H_i^z}\right)\cdots \\ & \times \left(\prod_{i=L}^{1} e^{-id_q h H_i^z}\right)\left(\prod_{i=L}^{1} e^{-id_q h H_i^x}\right), \end{aligned} \tag{22}$$

每个阶段都贯穿整个链，因此我们将其称为全局排序 $S^{(2)}$。

我们将自己的 $8^{\mathrm{th}}$ 阶方案与精确时间演化 $U(t) = \exp(-itH)$ 进行比较。因此，Trotter 化演化 $\tilde{U}(t) = \left[S_n^{(\Lambda)}(h)\right]^{t/h}$ 由乘积公式的 $t/h$ 个步长组成。我们通过 $U - U$ 的归一化 Frobenius 范数定义实验误差˜，其计算方式为

$$\mathrm{Err}_n^{\mathrm{exp.}} = \sqrt{\frac{1}{N}\sum\nolimits_v \left\|(U - \tilde{U})|v\rangle\right\|^2}, \tag{23}$$

其中，求和遍历所有 $N = 2^L$ 个计算基态 $|v\rangle$，而 $\|\cdot\|$ 表示欧几里得范数。

我们的第一个数值实验是在固定的计算成本下模拟新的乘积公式 $q/h = 10^3$，并设定固定的总时间 $t = 10$。我们根据理论误差 $\mathrm{Err}_8$ (8) 以及与原点的距离 $\bar{x}$ (12) 对方案进行分类，如图1所示。然而，除了这些理论分类指标之外，我们还可以将实验误差（23）作为另一个维度。这样，我们便能够了解这两个量中的哪一个在实践中会影响效率。

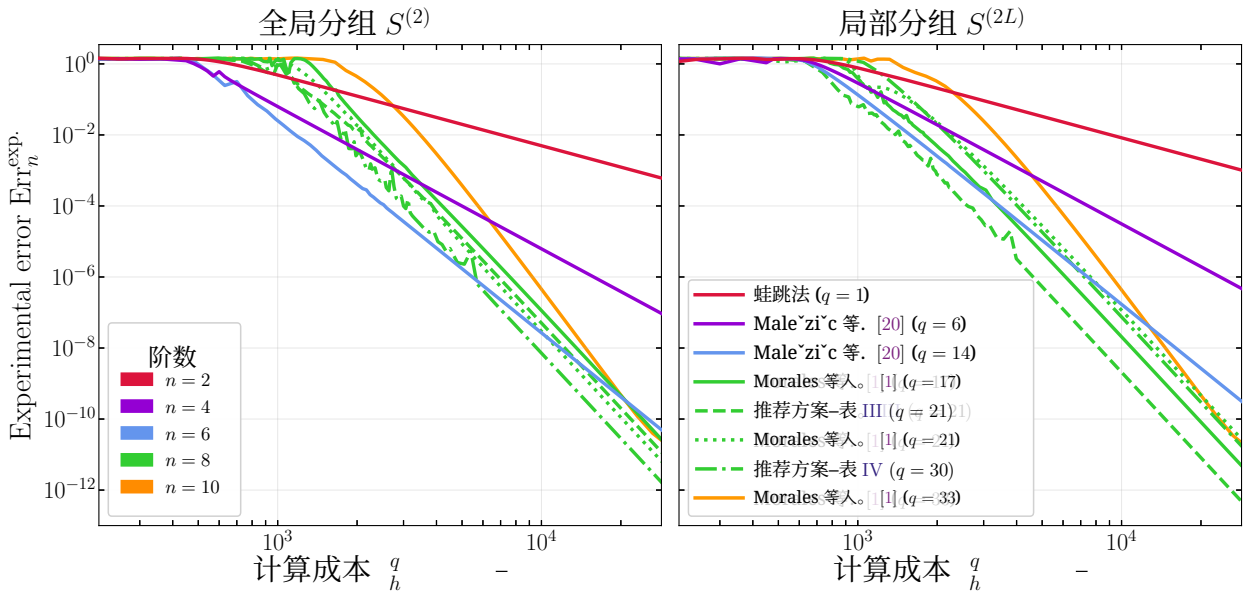



图 5。海森堡 XZ 自旋链在开放边界条件下的实验误差 $\mathrm{Err}_n^{\mathrm{exp.}}$(23)，链长为 $L=6$，绘制为计算成本 $q/h$ 与全局排序 $S^{(2)}$（左）及局部排序 $S^{(2L)}$（右）的关系。为着重考察长时间演化，我们设置总时间 $t=100$ 并改变时间步长 $h$。我们将蛙跳方案、在 $n=4$、6 [20] 时采用的推荐方案，以及在 $n=8$ 时采用的新方案（表 III 和 IV）与 Morales *et al*的方案进行比较。 [1] 在 $n=8$，我们的两种方案各自在一种排序下达到更低的误差，而在另一种排序下则保持相当的表现。

在图4 中，我们展示了最大循环次数 $q=30$ 下局部 $S^{(2L)}$和全局 $S^{(2)}$ 乘积公式排序（或分组）的一个结果。由于两种排序方式达到的绝对误差不同，我们根据每种排序方式中所有方案的最小实验误差 $\mathrm{Err}_8^{\star}$，将每个方案按相对误差着色。

$$\Delta\mathrm{Err}_8^{\mathrm{exp.}} = \mathrm{Err}_8^{\mathrm{exp.}} - \mathrm{Err}_8^{\star}, \tag{24}$$

这使两种排序都采用统一色标。在全局排序中，$S^{(2)}$实验误差主要与 $\mathrm{Err}_8$, 相关，最佳方案位于底部。实验误差随后沿 $\mathrm{Err}_8$ 增大的方向增长，并略微朝距离$\bar{x}$增大的方向增长。这是预期之中的，因为 $\mathrm{Err}_8$ 是针对恰好两个算符推导的，因此能够捕捉具有 $\Lambda=2$ 个阶段的乘积公式中的所有误差项。另一方面，在局部排序$S^{(2L)}$中，实验误差主要与距离 $\bar{x}$相关，最佳方案位于左侧，接近均匀方案参数。在有$\Lambda=2L-1$ 个阶段时，局部项之间的许多对易子无法由 $\mathrm{Err}_8$, 捕捉，其贡献可能随着方案参数大小的增加而增长，因此更有利于近乎均匀的方案。这与近期关于方案结构会影响误差累积的研究结果一致 [37, 38]。均匀性是否通常会对具有许多阶段的乘积公式发挥更大作用，仍有待进一步研究，但我们在提出推荐方案时会将其纳入考量。

在接下来的模拟中，我们将总时间增加到 $t=100$，以研究长时间误差累积，并改变计算成本 $q/h$，由此得到图5所示的标度图。作为参考，我们选取蛙跳方案、参考文献 [20] 中我们推荐的方案（$n=4$）、6，以及Morales等人在 [1] 中提出的最佳方案（$n=8$）、10。在我们新提出的 $8^{\mathrm{th}}$ 阶方案中，我们选取一个来自复合区域、效率很高的方案（$q=21$），以及一个循环数达到最大值的通用方案（$q=30$）。这两种方案兼具较高的理论效率 $\mathrm{Eff}_8$ 和相对较小的距离 $\bar{x}$，其参数分别列于表III和IV中，我们将在第IIIC节讨论选择这些方案的原因。我们没有选取时效率最高的复合方案（30），因为尽管其效率约高出三倍，但距离 $\bar{x}\geq 2$较大，导致两种排序下的误差都更大。

对于图5 所示的标度分析，我们再次考察两种分组方式 $S^{(2)}$ 和 $S^{(2L)}$。我们选取的两种方案在这两种分组方式中表现都很好，但各自在其中一种分组方式中表现尤为出色。在 $q=21$ 个循环时（见表III），高效且均匀的方案在局部 $S^{(2L)}$ 分组中表现最佳，其误差低于 Morales等人提出的公式。 [1]在这种排序方式下，该方案在标度区域内的不同计算成本下，也优于我们推荐的低阶方案$n=4$；6 [20] 而在全局排序 $S^{(2)}$中， $q=21$ 该方案仅与 Morales等人提出的方案相当，而我们在最大循环次数下的第二种方案 $q=30$（见表IV）则略胜一筹。

比较两种排序方式，$q = 21$ 局部排序中的方案 $S^{(2L)}$ 整体表现最佳，紧随其后的是全局排序中 $q = 30$ 的通用方案$S^{(2)}$。我们认为，这是因为它具有较高的理论效率和较小的距离 $\bar{x}$。另一方面，在较低阶时，我们推荐的方案在全局排序中的表现优于局部排序。我们认为，这是因为这些方案是在效率与均匀性之间折中选择的；在局部排序中，更偏重均匀性或许会更有利。这与以下观察结果一致：对于高阶乘积公式，方案参数的最佳选择可能取决于排序方式 [39]。基于这些数值实验，我们将在下一小节讨论对 $8^{\text{th}}$ 阶方案的推荐。

### C. 方案推荐

优化了方案参数并分析其在时间演化模拟中的误差累积后，我们现在可以 $8^{\text{th}}$ 根据理论和实践标准，给出阶数方案的推荐。一方面，我们寻求效率 Eff 较高的方案$_8$ (19)，即在计算成本固定时具有较小的主导误差。另一方面，我们倾向于选择距离较小的均匀方案 $\bar{x}$；在我们的模拟中，这类方案表现出良好的误差累积特性，尤其是在局部排序方式下。基于这些标准，我们针对不同的循环数选出了若干方案，这些方案可在我们的代码库中获取 [26]。由于实际表现取决于模型和排序方式（见第 IIIB节），我们还优先考虑稳健方案：这类方案的表现从不糟糕，并且理想情况下在某些设置中表现良好，正如我们在两种排序方式中观察到的那样。由于无法保证这一点适用于所有系统，我们的总体推荐主要依据上述理论标准。

在 $8^{\text{th}}$ 阶时，我们推荐两种方案。第一种是来自 $q = 21$复合区域的高效率方案，其参数见表III。除了效率高之外，它也是我们发现的均匀性最高的方案之一。其镜像对称性 $c_j = c_{q+1-j}$表明它是由蛙跳步复合而成的方案，这也是该区域中唯一的方案类型（见第IIB节）。这些特性很可能使其成为适用于多种模型和算符排序方式的良好选择。在海森堡 XZ 链中（见图5），它在局部排序下优于 Morales等人提出的方案 [1] ，在全局排序下则与之相当。

第二个推荐方案是在最大循环数 $q = 30$下效率极高的通用方案，其参数见表 IV。作为一种通用方案，它不具备第一个推荐方案的镜像对称性。尽管通用方案不受复合方法的限制，但我们在 $q = 30$ 处找到的最高效通用方案仅比表 III中的方案略高效一些，其距离 $\bar{x}$也相对较大。因此，我们更推荐一种更加统一的

表 III. 高效率方案的推荐方案参数 $c_j$，其 $q = 21$ 和 $n = 8$ 具有最小距离之一 $\bar{x}$。其镜像对称性 $c_j = c_{q+1-j}$ 表明它是 Yoshida 型复合格式（9）。凭借其效率和均匀性，它是适用于多种模型和排序方式的良好候选方案。

| $\text{Eff}_8 = 5.965 \times 10^{-4}$<br>$\bar{x} = 0.689$ | |
|---|---|
| $j$ | $c_j = d_{q+1-j}$ |
| 1 | 0.041644811759980006 |
| 2 | 0.082401316690625129 |
| 3 | −0.100053679877075558 |
| 4 | 0.056276640904364281 |
| 5 | 0.054789293427501641 |
| 6 | 0.052417975744997715 |
| 7 | 0.202143137014514557 |
| 8 | −0.156387050011859596 |
| 9 | 0.047476878333309942 |
| 10 | 0.077958103121855915 |
| 11 | −0.217334854216428036 |
| 12 | 0.077958103121855915 |
| 13 | 0.047476878333309942 |
| 14 | −0.156387050011859596 |
| 15 | 0.202143137014514557 |
| 16 | 0.052417975744997715 |
| 17 | 0.054789293427501641 |
| 18 | 0.056276640904364281 |
| 19 | −0.100053679877075558 |
| 20 | 0.082401316690625129 |
| 21 | 0.041644811759980006 |

格式方案，其效率和均匀性仍不如我们的第一个推荐方案，但在我们的基准测试中表现良好，尤其是在全局排序方面。总体而言，在一般区域中，均匀方案更难找到。我们的最小化器也无法达到那里仍然存在的复合方法（见第 IIB节），而限制在 Yoshida 型格式范围内的搜索或许能发现效率更高且更均匀的方案。

除了所给出的两种方案之外，我们还在 $n = 8$阶发现了更多具有竞争力的方案，其中一些更注重效率，另一些更注重均匀性，它们在某些情形下可能表现更好，我们将其提供于附录A。受均匀性在局部排序中所起作用的启发，我们还重新考察了 $n = 4$，6阶；在我们的部分测试中，更均匀的方案改进了我们先前的推荐，我们也将它们收录于同一附录。在 $n = 8$阶找到的所有方案，以及每个循环数 $q$ 下的若干推荐及其性质，均归档于我们的代码库 [26], ，其中大多数已最小化至四倍精度（36位有效数字）。该代码库还包含我们的程序，已更新以纳入新的方法并支持四倍精度。

表IV. 通用方案在最大循环数 $q = 30$、阶数 $n = 8$ 下的推荐方案参数 $c_j$。虽然它的效率和均匀性不如我们在 $q = 21$ 处的推荐方案，但在我们的基准测试中表现良好，尤其是在全局排序中，因为均匀性的影响较小。

| 效率$_8 = 3.685 \times 10^{-4}$<br>$\bar{x} = 1.220$ | |
|---|---|
| $j$ | $c_j = d_{q+1-j}$ |
| 1 | 0.026395999625833815 |
| 2 | −0.233684379758421046 |
| 3 | −0.347051470763567071 |
| 4 | −0.274493109455153539 |
| 5 | 0.061094592879806627 |
| 6 | 0.080452813774411269 |
| 7 | −0.063568811420699228 |
| 8 | 0.079026700770564284 |
| 9 | 0.059258569653719090 |
| 10 | 0.101028778330863239 |
| 11 | −0.164724202482292914 |
| 12 | −0.153554503905898948 |
| 13 | 0.107485850556923307 |
| 14 | −0.132867087173779791 |
| 15 | −0.074760981570922924 |
| 16 | 0.153609834123292238 |
| 17 | 0.024200829485005276 |
| 18 | −0.155771661397294425 |
| 19 | 0.089897001564395437 |
| 20 | 0.185309492367605244 |
| 21 | 0.093200666104659380 |
| 22 | 0.092333475195637504 |
| 23 | 0.050724396546212469 |
| 24 | 0.057234606768893276 |
| 25 | −0.114753172414141125 |
| 26 | 0.033620005282797918 |
| 27 | 0.217672800411770034 |
| 28 | 0.344599868646588370 |
| 29 | 0.286949675877411436 |
| 30 | 0.071133422375779604 |

## IV. 时间演化的量子电路

尽管乘积公式被广泛应用于许多领域，但它们近来因在量子计算中的应用而备受关注。量子计算机有望以仅随系统规模多项式增长的资源来模拟大型量子系统的时间演化 [3, 40]。由于乘积公式的每个因子 $e^{ic_j A_k}$ 都可以直接实现为一段较短的量子门序列，而无需辅助量子比特或块编码，因此乘积公式是完成这一任务的天然候选。然而，对于长时间动力学，Trotter步的数量以及由此产生的量子门数量会迅速增长，这使得高效的高阶方案尤其具有吸引力。因此，在本节中，我们考察新方案在量子电路中的适用性。

我们在第III B 节中的基准测试是在经典计算机上进行的，未在任何阶段加入噪声-，$e^{ic_j A_k}$。它们代表了容错量子计算机的理想情形，并让我们相信，一旦实现容错，高效的高阶乘积公式便可能派上用场。在此之前，我们必须应对退相干和含噪量子门，这正是下文的重点。

为了模拟含噪电路，我们使用 Qiskit [42]以及 `qiskit-omelyan` [22], ，这是我们开发的一个插件，用于构造基于 Omelyan 方法的乘积公式。Qiskit 核心库仅提供用于高阶方案的递归 Suzuki 构造，其阶段数随阶数呈指数增长；而我们的插件可使用用户定义的方案参数，在任意循环数下构造乘积公式 $q$。它包含文献中若干已确立的方案，以及我们针对阶数 $n = 4$，6，8 推荐的方案；该插件作为 Pauli 演化门的高级综合方法实现，可直接集成到标准 Qiskit 转译工作流中。

电路构建完成后，我们使用 Qiskit Aer 的密度矩阵方法对其进行模拟，并对每个泡利旋转门施加噪声。我们不采用特定于硬件的噪声模型，而是在每个门之后使用简单的去极化信道；对于 1 个和 2 个量子比特门，其错误概率分别为 $p_{1q}$ 和 $p_{2q}$。由于双量子比特门通常更难实现且噪声更大，我们设定$p_{1q} = p_{2q}/10$。

对于含噪声的基准测试，我们再次模拟海森堡 XZ 自旋链（$L = 6$），并采用开放边界条件。为清晰起见，我们选择总时间 $t = 10$，并将范围限制为局部排序 $S^{(2L)}$。我们不使用公式（23）中定义的实验误差，而是计算精确时间演化态与 Trotter 化（含噪声）态之间的保真度。由于精确演化态$|v\,(t)\,\rangle = e^{-iHt}|v\rangle$ 是纯态，（平方）Uhlmann 保真度可化简为

$$f_v = \langle v(t)|\rho_v|v(t)\rangle, \tag{25}$$

其中，$\rho_v$ 表示通过经过 Trotter 分解的含噪电路演化 $|v\rangle$ 得到的密度矩阵。为了获得更好的估计，我们再次对所有 $N = 2^L$ 计算基态 $|v\rangle$取平均，

$$\bar{f} = \frac{1}{N}\sum_v f_v. \tag{26}$$

在讨论含噪模拟的结果之前，我们先介绍所采用的误差缓解技术，即零噪声外推（ZNE） [43, 44]。我们通过局部幺正折叠来放大噪声 [45], ：将转译后的电路中的每个门 $G$ 替换为 $G(G^\dagger G)^k$ ，并设置$k = 0$、1、2，从而得到噪声缩放因子 $\lambda = 1$、3、5。对于每个初始态 $|v\rangle$ ，我们随后通过理查森外推将 $f_v(\lambda)$外推至$\lambda = 0$ ，即用经过这三个点的二次多项式进行外推，具体实现采用了 Mitiq [46], ；我们选择它是因为它不假设特定的噪声模型。请注意，ZNE 的目标是无噪声的

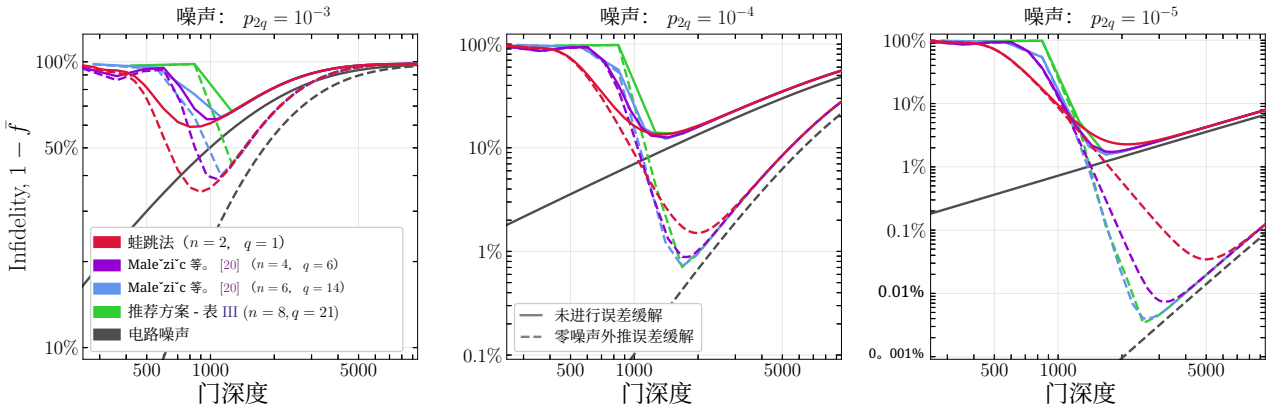



图 6. 非保真度 $1 - \bar{f}$ (26) 随海森堡 XZ 链的门深度变化（$L = 6$，开放边界，$t = 10$），采用局部排序 $S^{(2L)}$，在三种双量子比特退极化噪声水平下 $p_{2q}$，其中 $p_{1q} = p_{2q}/10$。门深度通过改变 Trotter步数来调整。实线表示未采用零噪声外推误差缓解的结果，虚线表示采用零噪声外推误差缓解的结果，灰线表示零角度蛙跳法量子电路的电路噪声。在当前最先进的含噪中等规模量子 [41] 噪声水平 $p_{2q} = 10^{-3}$（左图）下，蛙跳法仍是最佳选择。在 $p_{2q} = 10^{-4}$（中图）下，所有方案在未进行误差缓解时表现都相近，而 ZNE 显示高阶方案的效果提升了一倍。在 $p_{2q} = 10^{-5}$（右图）下，优势扩大到约一个数量级（见表 V）。

Trotter化电路，因此无法消除 Trotter 误差。此外，对于较深的电路，多项式外推会变得不可靠；在这类电路中，保真度会随着门数增加而近似呈指数衰减，折叠电路则会趋近最大混合态。需要强调的是，我们的 ZNE 设置是理想化的：每个折叠门都承受与原始门相同的退极化噪声，因此折叠会将噪声恰好放大 $\lambda$倍；而且我们的密度矩阵模拟不含散粒噪声，否则外推会进一步放大这种噪声。因此，我们的 ZNE 结果应被视为误差缓解技术在硬件上可能达到效果的乐观估计。使用我们的插件为海森堡模型构造此类时间演化电路的脚本 [22] 可在我们的代码仓库中找到 [26]。

我们在图6 中展示了三种双量子比特退极化噪声下的含噪模拟结果$p_{2q}$。为与图 5中的无噪声标度直接比较，我们绘制了非保真度 $1 - f$ 随门深度的变化曲线，其中门深度通过改变 Trotter 步数进行调整。我们将蛙跳方案（$n = 2$）与参考文献 $n = 4$ 中针对 $n = 6$ 和 [20],提出的推荐方案，以及表 $n = 8$ III中的新方案进行比较。我们还绘制了电路噪声：通过模拟将所有旋转角设为零的蛙跳量子电路得到该噪声。由于这些量子电路在理想情况下对计算基态实现恒等变换，因此可作为给定门深度下累积噪声的参考，而不受 Trotter 误差影响。不过，它们会略微低估噪声，因为$Z$型错误不会改变计算基态。除了含噪标度图，我们还在表V中列出了每条曲线的最小非保真度值。

图 6 中的三种不同噪声水平代表了 Trotter–Suzuki 方案在量子计算中适用性的潜在时间线。在当前含噪中等规模量子（NISQ）时代 [41, 47]，，$p_{2q} \gtrsim 10^{-3}$即使是最先进的设备也受到门深度限制，因此仍无法实现精确的长时间演化。在这一阶段，蛙跳方案凭借其简便性，无论是否采用误差缓解，仍是各种 Trotter 分解方法中的最佳选择，尽管非保真度依然较高。常数深度多乘积公式（MPF） [48]或（部分）随机化算法 [49–52]，可解决这一局限。关于 NISQ 设备上量子模拟前沿方法的详尽列表，见综述 [21]。

随着硬件不断改进，采用高阶分解变得越来越可行。在 $p_{2q} = 10^{-4}$ 未进行误差缓解时，所有方案的表现大致相同，但误差过大，无法获得可靠结果。采用误差缓解后，噪声得以降低，从而显现出高阶方案的优势 $n \geq 4$：与蛙跳法相比，其误差减半（见表V）。在 $p_{2q} = 10^{-5}$中，即使不进行误差缓解，高阶方案的表现也已优于蛙跳法；采用 ZNE 后，差异更加明显。我们的新型 $n = 8$ 方案取得了最佳结果，紧随其后的是推荐的 $n = 6$ 方案；二者的非保真度均比蛙跳法的最低值低约一个数量级，而门深度仅为其一半。推荐的 $n = 4$ 方案则介于两者之间，其非保真度约为蛙跳法的五分之一。因此，高阶方案不仅精度更高（见表V），成本效益也更高（见图6）。尽管图中展示的是理想化情形，但这表明，即使在容错量子计算的早期阶段，高阶乘积公式也能为高效模拟长时间动力学提供一条简便途径。

表 V. 各方案在阶数 $1-\bar{f}$ 下的最小非保真度 $n$，以及图 $p_{2q}$ 6 中存在和不存在 ZNE 误差缓解时的双量子比特退极化噪声（单位：%）。随着噪声水平 $p_{2q}$降低，高阶乘积公式的优势愈发显著，使其适用于早期容错模拟。

| $p_{2q}$ | ZNE | $n=2$ | $n=4$ | $n=6$ | $n=8$ |
|---|---|---|---|---|---|
| $10^{-3}$ | no | **59.2** | 63.0 | 64.1 | 67.1 |
| | yes | **35.0** | 38.8 | 40.0 | 44.1 |
| $10^{-4}$ | no | 13.4 | 12.6 | **12.2** | 13.7 |
| | yes | 1.50 | 0.88 | 0.72 | **0.71** |
| $10^{-5}$ | no | 2.27 | 1.72 | 1.57 | **1.56** |
| | yes | 0.034 | 0.0074 | 0.0040 | **0.0034** |

## V. 结论

在本研究中，我们推导出了新的高效 $n=8$ 阶 Suzuki-Trotter 分解，见表III和IV。对于长时间演化，高阶方案尤其可行，因为达到给定误差 $\epsilon$所需的成本按 $\mathcal{O}\left(t^{1+\frac{1}{n}}\epsilon^{-\frac{1}{n}}\right)$ 随总时间 $t$变化。在推导和基准测试新方案的过程中，我们发现，究竟是较小的主导阶误差还是尽可能均匀的方案参数能带来最佳性能，取决于哈密顿量。

我们将参考文献 [20]中用于从头构造高阶乘积公式的框架扩展到了 $n=8$ 阶，详见第IIA节。我们识别出 $8^{\text{th}}$阶基交换子 $F_k$ （4），并通过递归公式推导出其系数 $\epsilon_k$ （6），从而能够最小化 $O_9$ （5）误差项。通过考察第IIB节中的约束雅可比矩阵，我们确认了 $n=8$ 阶解的流形结构，此前已知的只有复合格式。尽管我们的框架旨在寻找通用方案，但仍能在复合区域中识别 Yoshida 型格式。然而，随着我们接近一般区域，通用框架的雅可比矩阵会逐渐变得病态，找到的极小值数量也随之减少（见图2）。通过将复合方法的镜像对称性直接施加于方案参数，我们恢复了较低的条件数，并在循环数超过 $q=16$时，对每个循环数都找到了更多极小值（见第IIC节）。

通过这些改进，我们的方法在复合区域内建立了效率更高的新方案 $\text{Eff}_8$ （19）（见图3）。此外，我们还发现了参数为 $q\geq 27$的通用方案，这是迄今尚未研究过的方案，其效率与参数为 $q=21$时的最佳复合方法相当。由于理论效率并不能完全对应实际性能，我们在第 $L=6$ IIIB节中对长度为 $L=6$ 的海森堡 XZ 自旋链进行了基准测试。作为数值实验的一部分，我们比较了两种算符排序方式：局部分组 $S^{(2L)}$ （21）和全局分组 $S^{(2)}$ （22）。

在全局排序中，我们观察到实验误差与效率 $\text{Eff}_n$相关，因为其 $\Lambda=2$个阶段都被 $\text{Err}_n$完全捕获；而在局部排序中，实验误差主要与到原点的距离 $\bar{x}$ （12）相关（见图4）。为了将我们的方案与 Morales等人 [1], 提出的方案进行比较，我们在图5中研究了累积误差 $\text{Err}_n^{\text{exp.}}$与计算成本$q/h$ 之间的标度关系。我们在参数为 $q=21$ 时的推荐方案在局部分组 $S^{(2L)}$中表现最佳，而参数为 $q=30$ 时的通用方案在全局分组 $S^{(2)}$中表现最佳；它们的参数列于表III和IV中。我们还在附录 A中讨论了其他方案，并在代码库 [26]中针对每个循环数 $q$ 给出了推荐方案。

最后，在第IV节中，我们重点研究了 Trotter-Suzuki 方案在量子电路中的适用性。我们像之前一样模拟了海森堡 XZ 链，但对实现 Trotter 步长的每个门都施加了噪声。为高效完成这项工作，我们为 Qiskit 开发了一个插件 [42], ，并将其命名为`qiskit-omelyan` [22], 以纪念 Omelyan等人提出的方法。 [18]借助该插件，我们合成了演化电路，并应用了一个针对双量子比特门、去极化水平为 $p_{2q}$ 的简单噪声模型。我们还研究了使用零噪声外推（ZNE）进行误差缓解的效果。对于三个噪声水平 $p_{2q}$，我们计算了演化态的非保真度，并将其与所需门深度的关系绘制成图（见图6）。我们发现，在当前最先进硬件的噪声水平下，蛙跳方案仍是最佳选择。然而，在较低噪声水平下，我们推荐的方案也具有竞争力，因此随着硬件的改进，这些方案可用于长时间动力学模拟。

如果要将我们的框架扩展到 $n=10$阶，通用方案才会出现于$q\geq 83$，那么将其适配到复合格式将变得至关重要。然而，其中的主要瓶颈在于系数公式的推导，我们将扩展到 $n=10$ 留待未来工作。此外，我们的框架也可适用于复数方案参数，而除了某些 $n=4$ [24]阶方案之外，这些参数尚未得到充分探索。另一个探索方向，是将通用 Trotter-Suzuki 框架专门应用于辛积分器，此时条件$[B,[B,[B,A]]]=0$ 会缩减交换子基。构造辛积分器可能有益于经典模拟，例如分子动力学或天体动力学模拟。我们目前正在开展将这一适配纳入框架的工作。最后，将高阶方案应用于其他物理系统，可能会受到某些领域的高度关注。例如，采用哈密顿形式描述的晶格场论的 Trotter 误差界，或许能够为量子硬件上的模拟提供更准确的资源成本估计。

## 代码与数据

复现本文结果所需的代码和方案参数已发布于 GitHub 和 Zenodo [26]。递归公式最初使用 Wolfram Mathematica 推导，[25], 所用例程由 FeynCalc 提供 [53, 54]。我们还将推导移植到了开源 Python 软件包 `pybch`上，该软件包基于 SymPy [27], ，并包含在同一代码库中。最小化框架使用`C++`编写，并采用线性代数库 Eigen [55]。第 IIIB 节中的无噪声时间演化使用`C++`实现，而第IV 节中的含噪电路分析则使用 Python 实现。我们使用 Qiskit [42], 及我们的插件`qiskit-omelyan` [22] 构造量子电路，以将时间演化算子合成为泡利旋转门。含噪电路使用 Qiskit Aer [42], 进行模拟，并采用零噪声外推使用 Mitiq [46] 完成了模拟。我们时间演化模拟的数据可以通过所提供的代码复现，并可根据请求从作者处获取。

## 致谢

我们感谢 Luca Dellantonio、Martina Gisti、Emanuele Mendicelli 和 Jinzhao Sun 富有洞见的讨论以及文献推荐。作者衷心感谢能够使用波恩大学的 Marvin 集群。我们感谢使用 Claude 对本手稿进行校对以及软件调试。本工作由德国科学基金会（DFG）资助，属于 CRC 1639 NuMeriQS 项目的一部分——项目编号 511713970。

[1] M. E. S. Morales、P. C. S. Costa、G. Pantaleoni、D. K. Burgarth、Y. R. Sanders 和 D. W. Berry, 《选择和改进乘积公式以实现量子模拟的最佳性能》, Quant. Inf. Comput. 25, 1 2025 2210 ( ), arXiv: .15817 [quant-ph]。[2] G. Lami、M. Collura 和 N. Ranabhat, 《量子自旋链、精确对角化与张量网络入门讲义》（ 2025 2503 03564 ）, [ ] arXiv: . cond-mat.str-el。[3] S. Lloyd, 《通用量子模拟器》, Science 273, 1073 1996( )。[4] H. F. Trotter, 《关于算子半群的乘积》, Proceedings of the American Mathematical Society 10, 545 (1959)。[5] M. Suzuki, 《指数算子与内导子的广义 Trotter 公式及系统逼近式：在多体问题中的应用》, Co mmun. Math. Phys. 51, 183 (1976)。[6] A. M. Childs 和 N. Wiebe, 《使用酉运算的线性组合进行哈密顿量模拟》, Quant. Inf. Comput. 12, 0901 (2012), arXiv:1202.5822 [quant-ph]。[7] G. H. Low 和 I. L. Chuang, 《通过量子信号处理实现最优哈密顿量模拟》, Phys. Rev. Lett. 118, 010501 (2017), arXiv:1606.02685 [quant-ph]。[8] A. M. Childs、A. Ostrander 和 Y. Su, 《通过随机化实现更快的量子模拟》, Quantum 3, 182 (2019), arXiv:1805.08385 [quant-ph]。[9] E. Campbell, 《用于快速哈密顿量模拟的随机编译器》, Phys. Rev. Lett. 123, 070503 (2019), arXiv:1811.08017 [quant-ph]。[10] E. Hairer、C. Lubich 和 G. Wanner, 《 *Geometric numerical integration* 2 》, 第 nd 版, 《计算数学丛书》第 31 卷（Springer-Verlag, 柏林, 2006 年）。[11] A. M. Childs、D. Maslov、Y. Nam、N. J. Ross 和 Y. Su, 《迈向首次具有量子加速的量子模拟》, Proceedings of the National Academy of Sciences 115, 9456 (2018), arXiv:1711.10980 [quant-ph]。[12] A. M. Childs、Y. Su、M. C. Tran、N. Wiebe 和 S. Zhu, 《具有交换子标度的 Trotter 误差理论》, Phys. Rev. X 11, 011020 (2021), arXiv:1912.08854 [quant-ph]. [13] E. T. Campbell, 《Hubbard模型的早期容错模拟》, Quantum Sci. Technol. 7, 015007 (2022), arXiv:2012.09238 [ quant-ph]. [14] M. Suzuki, 《指数算符的分形分解及其在多体理论和蒙特卡罗模拟中的应用》, Physics Letters A 146, 319 (1990). [15] N. Hatano 和 M. Suzuki, 《求解高阶指数乘积公式》, 载于 *Quantum Annealing and Other Optimization Methods* (Springer Berlin Heidelberg, 2005) pp. 37–68. [16] H. Yoshida, 《高阶辛积分器的构造》, Physics Letters A 150, 262 (1990). [17] I. Omelyan、I. Mryglod 和 R. Folk, 《用于多体系统运动积分的优化 Forest–Ruth 和 Suzuki 类算法》, Computer Physics Communications 146, 188 (2002). [18] I. Omelyan、I. Mryglod 和 R. Folk, 《辛解析可积分解算法：分类、推导及其在分子动力学、量子力学和天体力学模拟中的应用》, Computer Physics Communications 151, 272 (2003). [19] S. Blanes 和 P. Moan, 《实用的辛分割 Runge–Kutta 和 Runge–Kutta–Nystr¨om 方法》, Journal of Computational and Applied Mathematics 142, 313 (2002). [20] M. Male˘zi˘c 和 J. Ostmeyer, 《适用于长时间量子动力学的高效 Trotter–Suzuki 方案》, Journal of Physics A: Mathematical and Theoretical 59, 345301 (2026). [21]J. Ostmeyer, 《Suzuki-Trotter 分解及其他量子时间演化方法》 (2026), arXiv:2609.19184 [quant-ph]. [22] M. Male˘zi˘c, qiskit-omelyan (v1.0.0), Zenodo (2026), https://github.com/MarkoMalezic/qiskit-omelyan, doi:10.5281/zenodo.24371685. [23]M. Male˘zi˘c 和 J. Ostmeyer, 《通过高效 Trotter–Suzuki 方案减少量子门数量》, PoS LATTICE 2025 , 131 2026 2602 21145 ( ), [ ] arXiv: . hep-lat .

[24] J. Ostmeyer，《面向经典计算和量子计算的优化 Trotter 分解》，J. Phys. A 56, 285303 (2023), arXiv:2211.02691 [quant-ph]. [25] Wolfram Research, Inc.，Mathematica (2026)，美国伊利诺伊州尚佩恩，2026。可访问：https://www.wolfram.com/mathematica.[26]M. Maleˇziˇc，《高效高阶 Trotter 分解》(v1.1)，Zenodo (2026), https://github.com/MarkoMalezic/efficient-trotterizations, doi:10.5281/zenodo.24374526. [27] A. Meurer、C. P. Smith、M. Paprocki、O. ˇCert´ık、S. B. Kirpichev、M. Rocklin、A. Kumar、S. Ivanov、J. K. Moore、S. Singh、T. Rathnayake、S. Vig、B. E. Granger、R. P. Muller、F. Bonazzi、H. Gupta、S. Vats、F. Johans- son、F. Pedregosa、M. J. Curry、A. R. Terrel、ˇS. Rouˇcka、A. Saboo、I. Fernando、S. Kulal、R. Cimrman 和 A. Scopatz，《SymPy：Python 符号计算》，PeerJ Comput. Sci. 3, e103 (2017). [28] A. Arnal、F. Casas 和 C. Chiralt，《关于以右嵌套对易子表示 Baker–Campbell–Hausdorff 级数的说明》，Mediterranean J. Math. 18, 53 (2021), arXiv:2006.15869 [math-ph]. [29] R. I. McLachlan，《利用对称复合方法对常微分方程进行数值积分》，SIAM Journal on Scientific Computing 16, 151 (1995), https://doi.org/10.1137/0916010. [30] W. Kahan 和 R.-C. Li，《用于提升常微分方程非常规格式阶数的复合常数》，Mathematics of Computation 66, 1089 1997( ). [31] M. Sofroniou 和 G. Spaletta，《对称积分器的对称复合常数的推导》，Optimization Methods and Software 20, 597 (2005). [32] K. Levenberg，《求解某些非线性最小二乘问题的方法》，Quarterly of Applied Mathematics 2, 164 (1944). [33] D. W. Marquardt，《一种用于非线性参数最小二乘估计的算法》，Journal of the Society for Industrial and Applied Mathematics 11, 431 (1963), https://doi.org/10.1137/0111030. [34] M. K. Transtrum 和 J. P. Sethna，《改进 Levenberg–Marquardt 算法以进行非线性最小二乘最小化》( 2012 1201 5885 ), ar[ ]Xiv: . physics.data-an . [35] L. Verlet，《计算机上的经典流体“实验”。I. Lennard-Jones 分子的热力学性质》，Phys. Rev. 159, 98 (1967). [36] E. Forest 和 R. D. Ruth，《四阶辛积分》，Physica D: Nonlinear Phenomena 43, 105 (1990). [37] Y.-H. Chen，《通过对易子分解实现 Trotter 误差时间尺度分离》，Phys. Rev. A 111, 022612 (2025), arXiv:2409.16634 [quant-ph]. [38] B. Chen、J. Xu、X. Yuan 和 Q. Zhao，《量子模拟中误差干涉的通用框架》，Phys. Rev. Lett. 136, 200601 (2026). [39] Z. Liu、Z.-Y. Wei、M. Foss-Feig、A. V. Gorshkov 和 A. M. Childs，《考虑排序的 Trotter 误差理论》(2026), arXiv:2610.03672 [quant-ph]. [40] R. P. Feynman，《用计算机模拟物理》，International Journal of Theoretical Physics 21, 467 (1982). [41] J.-W. Yin、Z.-R. Chen、S. Peng、H.-C. Luo、C.-N. Tao、S.- W. Tan 和 L.-Q. Lu，《量子计算综述系统与软件，《计算机科学与技术杂志》41, 147 (2026)。[42] A. Javadi-Abhari、M. Treinish、K. Krsulich、C. J. Wood、J. Lishman、J. Gacon、S. Martiel、P. D. Nation、L. S. Bishop、A. W. Cross、B. R. Johnson 与 J. M. Gambetta，使用 Qiskit 进行量子计算 (2024)，arXiv:2405.08810 [quant-ph]。[43] K. Temme、S. Bravyi 与 J. M. Gambetta，短深度量子电路的误差缓解，Phys. Rev. Lett. 119, 180509 (2017)。[44] Y. Li 与 S. C. Benjamin，结合主动误差最小化的高效变分量子模拟器，Phys. Rev. X 7, 021050 (2017)。[45] T. Giurgica-Tiron、Y. Hindy、R. LaRose、A. Mari 与 W. J. Zeng，用于量子误差缓解的数字零噪声外推，载于 *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)* (2020)，第 306–316 页。[46] R. LaRose、A. Mari、S. Kaiser、P. J. Karalekas、A. A. Alves、P. Czarnik、M. El Mandouh、M. H. Gordon、Y. Hindy、A. Robertson、P. Thakre、M. Wahl、D. Samuel、R. Mistri、M. Tremblay、N. Gardner、N. T. Stemen、N. Shammah 与 W. J. Zeng，Mitiq：面向含噪量子计算机误差缓解的软件包，Quantum 6, 774 (2022)。[47] J. Preskill，NISQ时代及以后的量子计算，Quantum 2, 79 (2018)。[48] C. B. Bark、S. Lee、K. J. Ahn、S. Cheon、M. J. Park 与 Y. Kim，用于近期数字量子模拟中 Trotter 误差缓解的常数深度多乘积公式 (2026)，arXiv:2607.12225 [quant-ph]。[49] E. Granet 与 H. Dreyer，数字量子计算机上无离散化误差的哈密顿动力学，npj Quantum Inf. 10, 82 (2024)，arXiv:2308.03694 [quant-ph]。[50] T. Hayata 与 Y. Kikuchi，基于概率角度插值的连续时间演化及其应用 (2026)，arXiv:2604.02854 [quant-ph]。[51] J. G¨unther、F. Witteveen、A. Schmidhuber、M. Miller、M. Christandl 与 A. W. Harrow，采用部分随机化时间演化的相位估计，PRX Quantum 7, 020332 (2026)，arXiv:2503.05647 [quant-ph]。[52] K. Murota、Y. Kikuchi、E. Rinaldi、F. Sauvage 与 S. Todo，通过反转Trotter误差动力学实现无偏哈密顿量模拟 (2026)，arXiv:2606.29741 [quant-ph]。[53] R. Mertig、M. B¨ohm 与 A. Denner，FeynCalc——费曼振幅的计算机代数计算，Computer Physics Communications 64, 345 (1991)。[54] V. Shtabovenko、R. Mertig 与 F. Orellana，FeynCalc 10：多圈积分会梦见计算机代码吗？，Computer Physics Communications 306, 109357 (2025)。[55] G. Guennebaud、B. Jacob、 *et al.*，Eigen，https://libeigen.gitlab.io (2010)。

**附录 A：其他方案参数**

在数值测试中，我们注意到，有些方案表现良好，尽管它们并非其阶数下效率最高的方案。这些方案倾向于采用更均匀的参数，其到原点的距离是我们发现的最小值之一 $\bar{x}$ 。下面我们给出这类

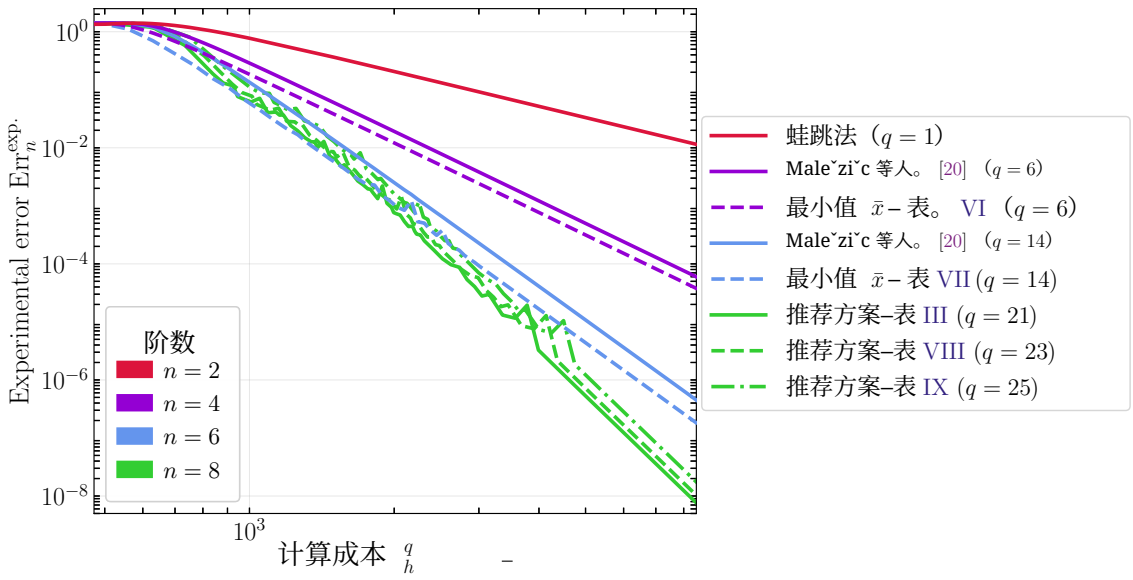



图 7. 实验误差 $\mathrm{Err}_n^{\mathrm{exp.}}$（23）相对于计算成本 $q/h$ 计算具有 $L=6$ 个格点且采用开放边界条件的海森堡 XZ 自旋链的时间演化所需。我们绘制了局部分组 $S^{(2L)}$ 中、在固定总时间 $t=100$ 下的若干额外方案，并将它们与蛙跳方案、我们在 [20] 中的推荐方案以及我们的 $q=21$ 方案（表 III）进行比较。最小的 $\bar{x}$ 方案在较低阶 $n=4$、6 时于此情形下略微降低了误差。虽然额外方案在 $8^{\mathrm{th}}$ 阶并未优于我们在 $q=21$ 个循环处的推荐格式，但它们仍具有竞争力。

阶数为 $n=4$、6、8 的方案作为补充建议，并在图 7 中比较它们在海森堡 XZ 模型上的实际表现与我们的主要建议。

首先，我们给出两个较低阶的方案 $n=4$、6。两者的循环数分别达到最大值 $q=6$ 和$q=14$，且具有最小的到原点的距离 $\bar{x}$，即在其 $q$处。其参数分别见表 VI和 VII。在局部分组中模拟海森堡模型时，我们发现它们略优于我们的推荐方案 [20] 在累积误差 $\mathrm{Err}_n^{\mathrm{exp.}}$（见图 7）。这支持了我们在第 IIIB 节中的建议，即在局部排序中更强地偏好均匀性可能是有益的。

表 VI. 在最大循环数 $q=6$ 下， $4^{\mathrm{th}}$ 阶方案的方案参数 $c_j$。该方案在此 $q$ 下到原点的距离 $\bar{x}$ 最小，但代价是效率较低。

| 效率$_4=4.829$ $\bar{x}=0.663$ | |
|---|---|
| $j$ | $c_j=d_{q+1-j}$ |
| 1 | 0.2878346136197648 12 |
| 2 | 0.151098631207585238 |
| 3 | 0.154721182891605497 |
| 4 | 0.112757707574057481 |
| 5 | 0.117761677709795246 |
| 6 | −0.324173813002808231 |

表 VII. 在最大循环数 $q=14$ 下， $6^{\mathrm{th}}$ 阶方案的方案参数 $c_j$。该方案在此 $q$ 下到原点的距离 $\bar{x}$ 最小，但代价是效率较低。

| 效率$_6=0.150$ $\bar{x}=0.637$ | |
|---|---|
| $j$ | $c_j=d_{q+1-j}$ |
| 1 | 0.0437041359488762 53 |
| 2 | 0.0859719593515434 34 |
| 3 | −0.1402109735286534 71 |
| 4 | 0.2357391697679190 05 |
| 5 | 0.0981752869008024 75 |
| 6 | 0.0945358658159630 44 |
| 7 | 0.0789812221935090 87 |
| 8 | 0.0685836659157610 11 |
| 9 | 0.0824856946456350 16 |
| 10 | −0.1654205207512137 45 |
| 11 | −0.2266754808089753 71 |
| 12 | 0.0573823517885473 23 |
| 13 | 0.1083450046513752 82 |
| 14 | 0.0784026181089106 52 |

在 $8^{\mathrm{th}}$ 阶时，我们利用改编框架找到了另外两种复合方法（见第IIC节）。第一种的 $q=23$ （见表VIII）在该 $q$ 下效率最高，且到原点的距离稍小于 $\bar{x}$ ，优于我们在 $q=21$ 个周期时的推荐方案。它的效率也略低一些，但这些特性可能使其成为整体表现良好的方案。另一种

复合方法有 $q = 25$ 个周期（见表IX），在效率和均匀性之间取得了平衡，其距离 $\bar{x}$ 与我们的推荐方案相当。我们将两种方案与推荐方案进行基准测试（见图7），发现它们的表现略逊一筹。尽管如此，它们仍具有竞争力，并且在其他模型中可能表现更好。

表 VIII. 方案参数 $c_j$，适用于 $8^{\text{th}}$ 阶复合格式，取值为 $q = 23$。该复合格式比我们在 $q = 21$ 个周期下推荐的格式略为均匀，效率约为其 70%。由于复合格式具有镜像对称性（见第IIB节），因此表格可以压缩至一半大小。

| 效率$_8 = 4.325 \times 10^{-4}$<br>$\bar{x} = 0.661$ | |
|---|---|
| $j$ | $c_j = d_{q+1-j}$ |
| 1, 23 | 0.035775813126863683 |
| 2, 22 | 0.082428855761252390 |
| 3, 21 | −0.093558098054503130 |
| 4, 20 | 0.044803882268296111 |
| 5, 19 | 0.041609234548577574 |
| 6, 18 | 0.037205238783110939 |
| 7, 17 | 0.039574307245707777 |
| 8, 16 | 0.194301912776415037 |
| 9, 15 | −0.151953430890394953 |
| 10, 14 | 0.040812082768358217 |
| 11, 13 | 0.083271171704635674 |
| 12 | −0.208541940076638649 |

表 IX. 方案参数 $c_j$，适用于 $8^{\text{th}}$ 阶复合格式，取值为 $q = 25$。该方案的效率不到我们在 $q = 21$时推荐方案的一半，距离则 $\bar{x}$。由于 $c_j = c_{q+1-j}$ 复合格式具有镜像对称性（见第IIB节），因此表格可以压缩至一半大小。

| 效率$_8 = 2.636 \times 10^{-4}$<br>$\bar{x} = 0.688$ | |
|---|---|
| $j$ | $c_j = d_{q+1-j}$ |
| 1, 25 | 0.047951181370257979 |
| 2, 24 | 0.058306285798459791 |
| 3, 23 | 0.081833806671706782 |
| 4, 22 | −0.099767039618495923 |
| 5, 21 | −0.081108207667039356 |
| 6, 20 | 0.056506301170499422 |
| 7, 19 | 0.061992066674556361 |
| 8, 18 | 0.069028295721143928 |
| 9, 17 | 0.188059120467134577 |
| 10, 16 | −0.142948183144266994 |
| 11, 15 | 0.050775495147567268 |
| 12, 14 | 0.060649773832167270 |
| 13 | −0.202666992847382155 |