Reviewing Time Evolution Methods for Quantum Systems

Johann Ostmeyer

Helmholtz-Institut für Strahlen- und Kernphysik, University of Bonn, Germany

Queen Mary University of London, QuantHEP, July 15, 2026

Gefördert durch

Deutsche Forschungsgemeinschaft

Suzuki-Trotter decompositions, aka splitting methods or (Lie) product formulae

[Suzuki *CommunMathPhys* **51** (1976); Trotter *ProcAMS* **4** (1959)]

$$\mathrm{i}\,\frac{\mathrm{d}}{\mathrm{d}t}\,|\psi(t)\rangle = H\,|\psi(t)\rangle$$

Suzuki-Trotter decompositions, aka splitting methods or (Lie) product formulae

[Suzuki *CommunMathPhys* **51** (1976); Trotter *ProcAMS* **4** (1959)]

$$\mathrm{i}\,\frac{\mathrm{d}}{\mathrm{d}t}\,|\psi(t)\rangle = H\,|\psi(t)\rangle$$

$$\Rightarrow |\psi(t)\rangle = \mathrm{e}^{-\mathrm{i}\,Ht}\,|\psi(0)\rangle$$

Suzuki-Trotter decompositions, aka splitting methods or (Lie) product formulae

[Suzuki *CommunMathPhys* **51** (1976); Trotter *ProcAMS* **4** (1959)]

$$
\begin{aligned}
&\mathrm{i}\,\frac{\mathrm{d}}{\mathrm{d}t}\,|\psi(t)\rangle = H\,|\psi(t)\rangle \\
&\Rightarrow |\psi(t)\rangle = \mathrm{e}^{-\mathrm{i}\,Ht}\,|\psi(0)\rangle \equiv U(-\mathrm{i}\,t)\,|\psi(0)\rangle
\end{aligned}
$$

Suzuki-Trotter decompositions, aka splitting methods or (Lie) product formulae

[Suzuki *CommunMathPhys* **51** (1976); Trotter *ProcAMS* **4** (1959)]

$$\mathrm{i}\,\frac{\mathrm{d}}{\mathrm{d}t}\,|\psi(t)\rangle = H\,|\psi(t)\rangle$$

$$\Rightarrow |\psi(t)\rangle = \mathrm{e}^{-\mathrm{i}\,Ht}\,|\psi(0)\rangle \equiv U(-\mathrm{i}\,t)\,|\psi(0)\rangle$$

$$H = \sum_i^{\Lambda} A_i\,, \quad [A_i, A_j] \neq 0$$

Suzuki-Trotter decompositions,
aka splitting methods or (Lie) product formulae

[Suzuki *CommunMathPhys* **51** (1976); Trotter *ProcAMS* **4** (1959)]

$$\mathrm{i}\,\frac{\mathrm{d}}{\mathrm{d}t}\,|\psi(t)\rangle = H\,|\psi(t)\rangle$$

$$\Rightarrow |\psi(t)\rangle = \mathrm{e}^{-\mathrm{i}\,Ht}\,|\psi(0)\rangle \equiv U(-\mathrm{i}\,t)\,|\psi(0)\rangle$$

$$H = \sum_{i}^{\Lambda} A_i\,, \quad [A_i,A_j] \neq 0$$

$$U(h) \equiv \mathrm{e}^{Hh}$$

Suzuki-Trotter decompositions, aka splitting methods or (Lie) product formulae

[Suzuki *CommunMathPhys* **51** (1976); Trotter *ProcAMS* **4** (1959)]

$$\mathrm{i}\,\frac{\mathrm{d}}{\mathrm{d}t}\,|\psi(t)\rangle = H\,|\psi(t)\rangle$$

$$\Rightarrow |\psi(t)\rangle = \mathrm{e}^{-\mathrm{i}\,Ht}\,|\psi(0)\rangle \equiv U(-\mathrm{i}\,t)\,|\psi(0)\rangle$$

$$H = \sum_i^{\Lambda} A_i\,, \quad [A_i, A_j] \neq 0$$

$$U(h) \equiv \mathrm{e}^{Hh}$$

$$U(h) = \mathrm{e}^{A_1 h}\,\mathrm{e}^{A_2 h}\cdots\mathrm{e}^{A_\Lambda h} + \mathcal{O}\left(h^2\right)$$

Suzuki-Trotter decompositions, aka splitting methods or (Lie) product formulae

[Suzuki *CommunMathPhys* **51** (1976); Trotter *ProcAMS* **4** (1959)]

$$\mathrm{i}\,\frac{\mathrm{d}}{\mathrm{d}t}\,|\psi(t)\rangle = H\,|\psi(t)\rangle$$

$$\Rightarrow |\psi(t)\rangle = \mathrm{e}^{-\mathrm{i}\,Ht}\,|\psi(0)\rangle \equiv U(-\mathrm{i}\,t)\,|\psi(0)\rangle$$

$$H = \sum_{i}^{\Lambda} A_i\,, \quad [A_i, A_j] \neq 0$$

$$U(h) \equiv \mathrm{e}^{Hh}$$

$$U(h) = \mathrm{e}^{A_1 h}\,\mathrm{e}^{A_2 h}\cdots\mathrm{e}^{A_\Lambda h} + \mathcal{O}\left(h^2\right)$$

Never use!

Suzuki-Trotter decompositions,
aka splitting methods or (Lie) product formulae

[Suzuki *CommunMathPhys* **51** (1976); Trotter *ProcAMS* **4** (1959)]

$$\mathrm{i}\,\frac{\mathrm{d}}{\mathrm{d}t}\,|\psi(t)\rangle = H\,|\psi(t)\rangle$$

$$\Rightarrow |\psi(t)\rangle = \mathrm{e}^{-\mathrm{i}\,Ht}\,|\psi(0)\rangle \equiv U(-\mathrm{i}\,t)\,|\psi(0)\rangle$$

$$H = \sum_{i}^{\Lambda} A_i\,, \quad [A_i, A_j] \neq 0$$

$$U(h) \equiv \mathrm{e}^{Hh}$$

$$U(h) = \mathrm{e}^{A_1 h}\,\mathrm{e}^{A_2 h}\cdots\mathrm{e}^{A_\Lambda h} + \mathcal{O}\left(h^2\right)$$ Never use!

$$U(h) = \mathrm{e}^{A_1 h/2}\,\mathrm{e}^{A_2 h/2}\ldots\mathrm{e}^{A_\Lambda h/2}\,\mathrm{e}^{A_\Lambda h/2}\ldots\mathrm{e}^{A_2 h/2}\,\mathrm{e}^{A_1 h/2} + \mathcal{O}\left(h^3\right)$$

Suzuki-Trotter decompositions, aka splitting methods or (Lie) product formulae

[Suzuki *CommunMathPhys* **51** (1976); Trotter *ProcAMS* **4** (1959)]

$$\mathrm{i}\,\frac{\mathrm{d}}{\mathrm{d}t}\,|\psi(t)\rangle = H\,|\psi(t)\rangle$$

$$\Rightarrow |\psi(t)\rangle = \mathrm{e}^{-\mathrm{i}\,Ht}\,|\psi(0)\rangle \equiv U(-\mathrm{i}\,t)\,|\psi(0)\rangle$$

$$H = \sum_i^{\Lambda} A_i\,, \quad [A_i, A_j] \neq 0$$

$$U(h) \equiv \mathrm{e}^{Hh}$$

$$U(h) = \mathrm{e}^{A_1 h}\,\mathrm{e}^{A_2 h} \cdots \mathrm{e}^{A_\Lambda h} + \mathcal{O}\left(h^2\right)$$ Never use!

$$U(h) = \mathrm{e}^{A_1 h/2}\,\mathrm{e}^{A_2 h/2} \cdots \mathrm{e}^{A_\Lambda h/2}\,\mathrm{e}^{A_\Lambda h/2} \cdots \mathrm{e}^{A_2 h/2}\,\mathrm{e}^{A_1 h/2} + \mathcal{O}\left(h^3\right)$$

$$\vdots$$

Error estimation and efficiency [Omelyan + *CPC* **146** (2002), *CPC* **151** (2003)]

$$\mathrm{e}^{(A+B)h+\mathcal{O}_1\,h+\mathcal{O}_3\,h^3+\mathcal{O}_5\,h^5+\cdots} = \mathrm{e}^{Aa_1h}\,\mathrm{e}^{Bb_1h}\cdots\mathrm{e}^{Bb_qh}\,\mathrm{e}^{Aa_{q+1}h}$$

Error estimation and efficiency [Omelyan+ *CPC* **146** (2002), *CPC* **151** (2003)]

$$\mathrm{e}^{(A+B)h+\mathcal{O}_1\,h+\mathcal{O}_3\,h^3+\mathcal{O}_5\,h^5+\cdots} = \mathrm{e}^{Aa_1h}\,\mathrm{e}^{Bb_1h}\cdots\mathrm{e}^{Bb_qh}\,\mathrm{e}^{Aa_{q+1}h}$$

$$\mathcal{O}_1 = (\nu-1)A + (\sigma-1)B, \quad \nu = \sum_i a_i, \sigma = \sum_i b_i$$

Error estimation and efficiency [Omelyan + *CPC* **146** (2002), *CPC* **151** (2003)]

$$\mathrm{e}^{(A+B)h+\mathcal{O}_1\,h+\mathcal{O}_3\,h^3+\mathcal{O}_5\,h^5+\cdots} = \mathrm{e}^{Aa_1h}\,\mathrm{e}^{Bb_1h}\cdots\mathrm{e}^{Bb_qh}\,\mathrm{e}^{Aa_{q+1}h}$$

$$\mathcal{O}_1 = (\nu-1)A + (\sigma-1)B, \quad \nu = \textstyle\sum_i a_i, \sigma = \sum_i b_i$$

$$\mathcal{O}_3 = \alpha[A,[A,B]] + \beta[B,[A,B]]$$

Error estimation and efficiency [Omelyan+ *CPC* **146** (2002), *CPC* **151** (2003)]

$$e^{(A+B)h+\mathcal{O}_1 h+\mathcal{O}_3 h^3+\mathcal{O}_5 h^5+\cdots} = e^{Aa_1 h}\, e^{Bb_1 h} \cdots e^{Bb_q h}\, e^{Aa_{q+1} h}$$

$$\begin{aligned}
\mathcal{O}_1 &= (\nu-1)A + (\sigma-1)B, \quad \nu = \textstyle\sum_i a_i,\, \sigma = \textstyle\sum_i b_i \\
\mathcal{O}_3 &= \alpha[A,[A,B]] + \beta[B,[A,B]] \\
\mathcal{O}_5 &= \gamma_1[A,[A,[A,[A,B]]]] + \gamma_2[A,[A,[B,[A,B]]]] \\
&\quad + \gamma_3[B,[A,[A,[A,B]]]] + \gamma_4[B,[B,[B,[A,B]]]] \\
&\quad + \gamma_5[B,[B,[A,[A,B]]]] + \gamma_6[A,[B,[B,[A,B]]]]
\end{aligned}$$

Error estimation and efficiency [Omelyan+ *CPC* **146** (2002), *CPC* **151** (2003)]

$$\mathrm{e}^{(A+B)h+\mathcal{O}_1\,h+\mathcal{O}_3\,h^3+\mathcal{O}_5\,h^5+\cdots} = \mathrm{e}^{Aa_1h}\,\mathrm{e}^{Bb_1h}\cdots\mathrm{e}^{Bb_qh}\,\mathrm{e}^{Aa_{q+1}h}$$

$$\begin{aligned}
\mathcal{O}_1 &= (\nu-1)A + (\sigma-1)B, \quad \nu = \textstyle\sum_i a_i,\, \sigma = \sum_i b_i \\
\mathcal{O}_3 &= \alpha[A,[A,B]] + \beta[B,[A,B]] \\
\mathcal{O}_5 &= \gamma_1[A,[A,[A,[A,B]]]] + \gamma_2[A,[A,[B,[A,B]]]] \\
&\quad + \gamma_3[B,[A,[A,[A,B]]]] + \gamma_4[B,[B,[B,[A,B]]]] \\
&\quad + \gamma_5[B,[B,[A,[A,B]]]] + \gamma_6[A,[B,[B,[A,B]]]]
\end{aligned}$$

$$\mathrm{Eff}_2 = \frac{1}{q^2\sqrt{|\alpha|^2+|\beta|^2}}$$

Error estimation and efficiency [Omelyan+ *CPC* **146** (2002), *CPC* **151** (2003)]

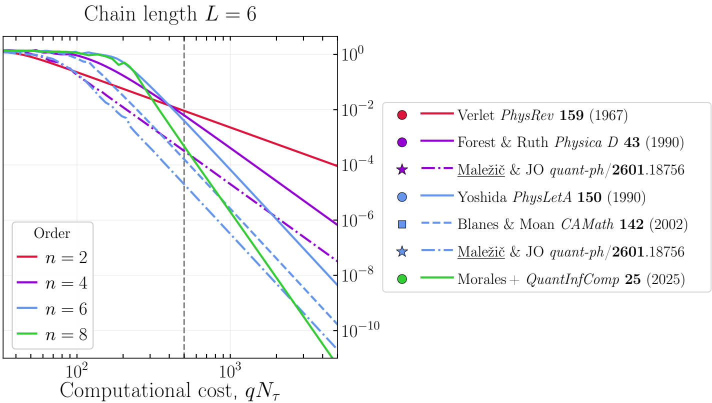
$$e^{(A+B)h+\mathcal{O}_1\,h+\mathcal{O}_3\,h^3+\mathcal{O}_5\,h^5+\cdots} = e^{Aa_1h}\,e^{Bb_1h}\cdots e^{Bb_qh}\,e^{Aa_{q+1}h}$$

$$\begin{aligned}
\mathcal{O}_1 &= (\nu-1)A+(\sigma-1)B, \quad \nu=\textstyle\sum_i a_i,\, \sigma=\sum_i b_i \\
\mathcal{O}_3 &= \alpha[A,[A,B]]+\beta[B,[A,B]] \\
\mathcal{O}_5 &= \gamma_1[A,[A,[A,[A,B]]]]+\gamma_2[A,[A,[B,[A,B]]]] \\
&\quad+\gamma_3[B,[A,[A,[A,B]]]]+\gamma_4[B,[B,[B,[A,B]]]] \\
&\quad+\gamma_5[B,[B,[A,[A,B]]]]+\gamma_6[A,[B,[B,[A,B]]]] \\
\mathrm{Eff}_2 &= \frac{1}{q^2\sqrt{|\alpha|^2+|\beta|^2}} \\
\mathrm{Eff}_4 &= \frac{1}{q^4\sqrt{\sum_{j=1}^{6}|\gamma_j|^2}}
\end{aligned}$$

Scheme construction and efficiency verification

[Maležič & JO *quant-ph*/**2601**.18756; Maležič *Qiskit* (2026)]

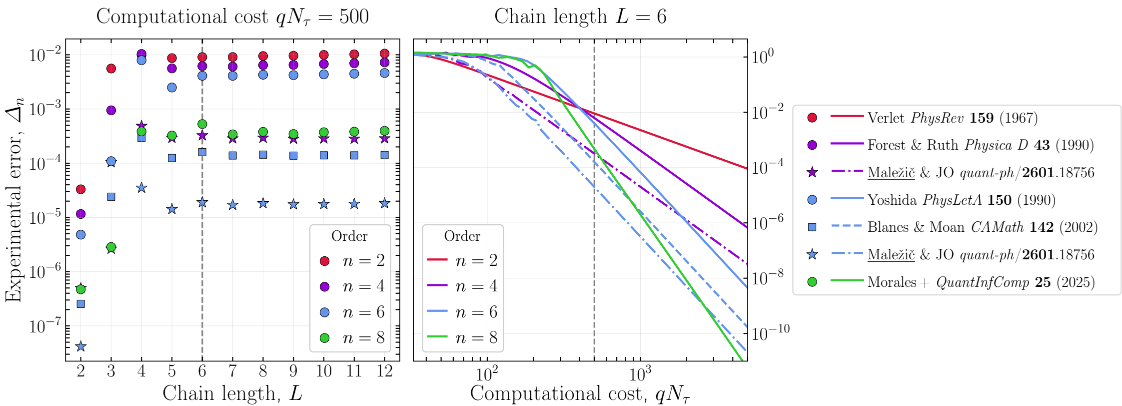
[Maležič & JO *PoS* **LATTICE** (2025)]



Scheme construction and efficiency verification

[Maležič & JO *quant-ph*/**2601**.18756; Maležič *Qiskit* (2026)]

[Maležič & JO *PoS* **LATTICE** (2025)]



General error bounds [Childs+ *PRX* **11** (2021); Heyl, Hauke+ *SciAdv* **5** (2019)]

Theorem (Generally applicable error bounds [Schubert & Mendl *PRB* **108** (2023)])

General error bounds [Childs + *PRX* **11** (2021); Heyl, Hauke + *SciAdv* **5** (2019)]

Theorem (Generally applicable error bounds [Schubert & Mendl *PRB* **108** (2023)])

Let $S_n(h) = \mathrm{e}^{A_K h} \cdots \mathrm{e}^{A_1 h}$ *be a product formula of order* n.

General error bounds [Childs + *PRX* **11** (2021); Heyl, Hauke + *SciAdv* **5** (2019)]

Theorem (Generally applicable error bounds [Schubert & Mendl *PRB* **108** (2023)])

Let $S_n(h) = \mathrm{e}^{A_K h} \cdots \mathrm{e}^{A_1 h}$ *be a product formula of order* n.
Then for every $s \in \{1, \ldots, K\}$ *the error is bounded:*

General error bounds [Childs+ *PRX* **11** (2021); Heyl, Hauke+ *SciAdv* **5** (2019)]

Theorem (Generally applicable error bounds [Schubert & Mendl *PRB* **108 (2023)])**

Let $S_n(h) = \mathrm{e}^{A_K h} \cdots \mathrm{e}^{A_1 h}$ be a product formula of order n. Then for every $s \in \{1, \ldots, K\}$ the error is bounded:

$$\left\| S_n(h) - \mathrm{e}^{Hh} \right\|_2 \leq \frac{|h|^{n+1}}{(n+1)!}$$

General error bounds [Childs+ *PRX* **11** (2021); Heyl, Hauke+ *SciAdv* **5** (2019)]

Theorem (Generally applicable error bounds [Schubert & Mendl *PRB* **108 (2023)])**

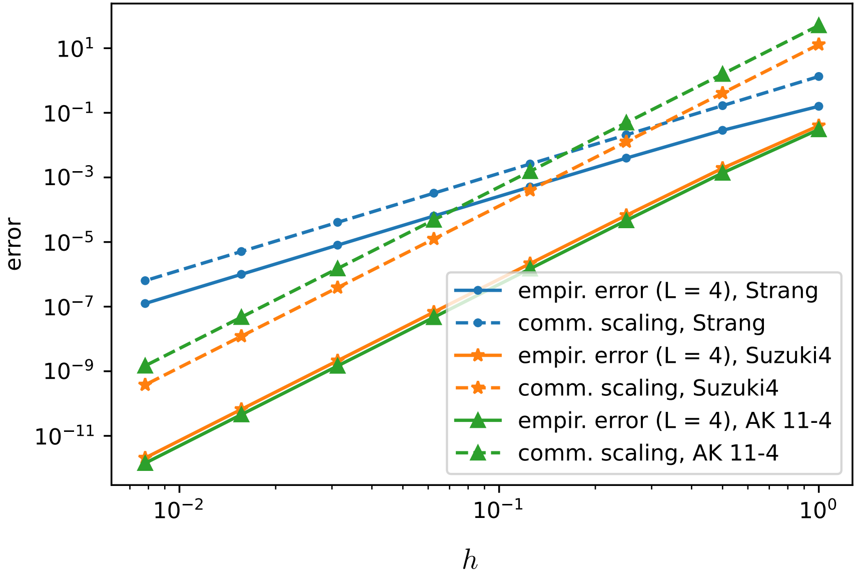
Let $S_n(h) = \mathrm{e}^{A_K h} \cdots \mathrm{e}^{A_1 h}$ be a product formula of order n.
Then for every $s \in \{1, \ldots, K\}$ the error is bounded:

$$\left\| S_n(h) - \mathrm{e}^{Hh} \right\|_2 \leq \frac{|h|^{n+1}}{(n+1)!} \left(\sum_{j=2}^{s} \sum_{\substack{p_j + \cdots + p_s = n \\ p_j \neq 0}} \binom{n}{p_j, \ldots, p_s} \left\| \mathrm{ad}_{A_s}^{p_s} \cdots \mathrm{ad}_{A_j}^{p_j} B_j \right\|_2 \right.$$

$$\left. + \sum_{j=s+1}^{K} \sum_{\substack{p_{s+1} + \cdots + p_j = n \\ p_j \neq 0}} \binom{n}{p_{s+1}, \ldots, p_j} \left\| \mathrm{ad}_{A_{s+1}}^{p_{s+1}} \cdots \mathrm{ad}_{A_j}^{p_j} B_j \right\|_2 \right),$$

$$\mathrm{ad}_C^p D := \underbrace{[C, \cdots [C}_{p\text{-times}}, D] \cdots] \quad \textit{and} \quad B_j := \sum_{\ell=1}^{j-1} A_\ell, \quad j = 2, \ldots, K\,.$$

Bound application [Schubert & Mendl *GitHub* (2023), *PRB* **108** (2023)]



Error accumulation [Chen *PRA* **111** (2025); Poster: Marko Maležič]

- Total time $t = Nh$

Error accumulation [Chen *PRA* **111** (2025); Poster: Marko Maležič]

- Total time $t = Nh$

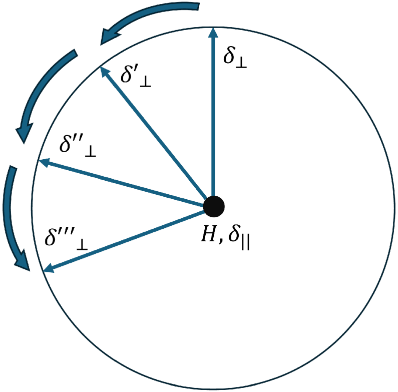
$\Rightarrow$ $\text{error}(t) \leq N\text{error}(h) = \mathcal{O}\left(Nh^{n+1}\right) = \mathcal{O}\left(th^{n}\right)$

Error accumulation [Chen *PRA* **111** (2025); Poster: Marko Maležič]

- Total time $t = Nh$

$\Rightarrow$ $\text{error}(t) \leq N\text{error}(h) = \mathcal{O}\left(Nh^{n+1}\right) = \mathcal{O}\left(th^{n}\right)$

- Error splits: 'parallel' $\delta_{\|} = \mathcal{O}\left(th^{n}\right)$ + 'orthogonal' $\delta_{\perp} = \mathcal{O}\left(h^{n}\right)$ parts.

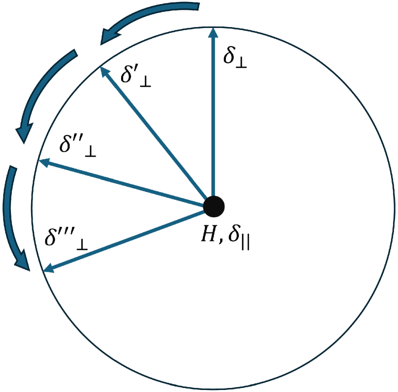


Error accumulation [Chen *PRA* **111** (2025); Poster: Marko Maležič]

- Total time $t = Nh$

$\Rightarrow$ error$(t) \leq N$error$(h) = \mathcal{O}\left(Nh^{n+1}\right) = \mathcal{O}\left(th^{n}\right)$

- Error splits: 'parallel' $\delta_{\|} = \mathcal{O}\left(th^{n}\right)$ + 'orthogonal' $\delta_{\perp} = \mathcal{O}\left(h^{n}\right)$ parts.
- Typically smallest 'parallel' error for uniform Trotter coefficients.

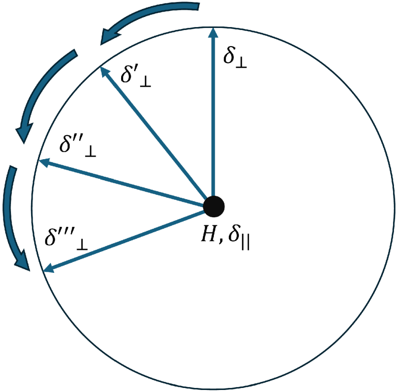


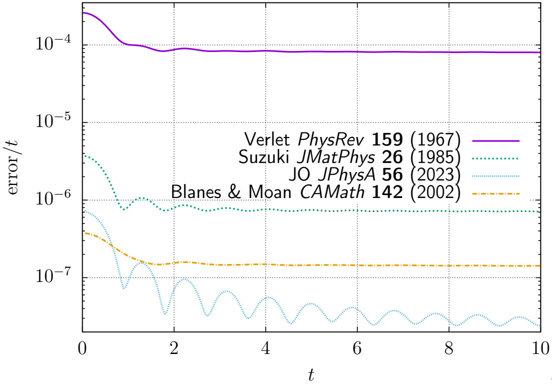
Error accumulation [Chen *PRA* **111** (2025); Poster: Marko Maležič]

- Total time $t = Nh$

$\Rightarrow$ $\text{error}(t) \leq N\text{error}(h) = \mathcal{O}\left(Nh^{n+1}\right) = \mathcal{O}\left(th^{n}\right)$

- Error splits: 'parallel' $\delta_{\|} = \mathcal{O}\left(th^{n}\right)$ + 'orthogonal' $\delta_{\perp} = \mathcal{O}\left(h^{n}\right)$ parts.
- Typically smallest 'parallel' error for uniform Trotter coefficients.





Worst case errors [Burgarth + *PRR* **6** (2024)]

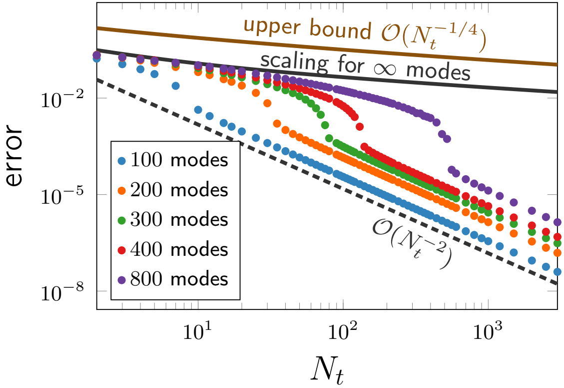
- Usual Trotter error valid for smooth bounded operators only!

Worst case errors [Burgarth+ *PRR* **6** (2024)]

- Usual Trotter error valid for smooth bounded operators only!
- E.g. Coulomb potential $\text{error}(h) = \mathcal{O}\left(h^{\frac{1}{4}}\right)$ for every scheme.

Worst case errors [Burgarth+ *PRR* **6** (2024)]

- Usual Trotter error valid for smooth bounded operators only!
- E.g. Coulomb potential $\mathrm{error}(h) = \mathcal{O}\left(h^{\frac{1}{4}}\right)$ for every scheme.
- Finite modes (discrete space) act as regulator.



Lower error bounds [Hahn+ *PRA* **111** (2025)]

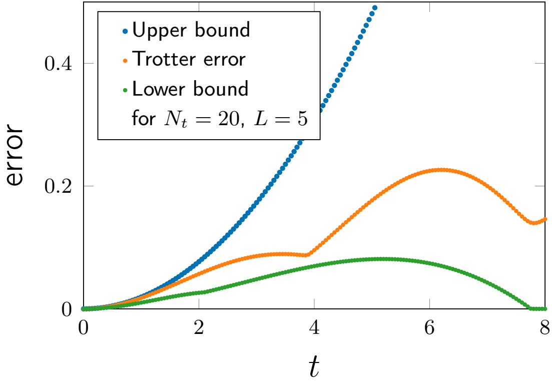
$$\left\| \left(\mathrm{e}^{-\mathrm{i}\,\frac{t}{n}A} \mathrm{e}^{-\mathrm{i}\,\frac{t}{n}B} \right)^n - \mathrm{e}^{-\mathrm{i}\,t(A+B)} \right\| \geq \frac{t^2}{2n} \left(1 - \frac{t}{2n} \|A\| \right)$$

Lower error bounds [Hahn+ *PRA* **111** (2025)]

$$\left\|\left(\mathrm{e}^{-\mathrm{i}\,\frac{t}{n}A}\mathrm{e}^{-\mathrm{i}\,\frac{t}{n}B}\right)^{n}-\mathrm{e}^{-\mathrm{i}\,t(A+B)}\right\| \geq \frac{t^2}{2n}\left(1-\frac{t}{2n}\|A\|\right)$$
$$\times\left(\|[A,B]\|-\frac{t}{2}\|[A+B,[A,B]]\|\right)-\frac{t^3}{24n^2}\|[A,[A,B]]\|-\frac{t^3}{12n^2}\|[B,[B,A]]\|$$

Lower error bounds [Hahn+ *PRA* **111** (2025)]

$$\left\|\left(\mathrm{e}^{-\mathrm{i}\,\frac{t}{n}A}\mathrm{e}^{-\mathrm{i}\,\frac{t}{n}B}\right)^n - \mathrm{e}^{-\mathrm{i}\,t(A+B)}\right\| \geq \frac{t^2}{2n}\left(1 - \frac{t}{2n}\|A\|\right)$$
$$\times\left(\|[A,B]\| - \frac{t}{2}\|[A+B,[A,B]]\|\right) - \frac{t^3}{24n^2}\|[A,[A,B]]\| - \frac{t^3}{12n^2}\|[B,[B,A]]\|$$



Decompositions into 2 operators [JO *JPhysA* **56** (2023); McLachlan *JSciComp* **16** (1995)]

$$\mathrm{e}^{(A+B)h+\mathcal{O}\left(h^{n+1}\right)} = \mathrm{e}^{Aa_1 h}\,\mathrm{e}^{Bb_1 h}\,\mathrm{e}^{Aa_2 h}\cdots\mathrm{e}^{Bb_q h}\,\mathrm{e}^{Aa_{q+1} h}$$

Decompositions into 2 operators [JO *JPhysA* **56** (2023); McLachlan *JSciComp* **16** (1995)]

$$
\begin{aligned}
\mathrm{e}^{(A+B)h+\mathcal{O}\left(h^{n+1}\right)} &= \mathrm{e}^{Aa_1h}\,\mathrm{e}^{Bb_1h}\,\mathrm{e}^{Aa_2h}\cdots\mathrm{e}^{Bb_qh}\,\mathrm{e}^{Aa_{q+1}h} \\
&= \mathrm{e}^{Ac_1h}\,\mathrm{e}^{Bc_1h}\,\mathrm{e}^{Bd_1h}\,\mathrm{e}^{Ad_1h}\cdots\mathrm{e}^{Bd_qh}\,\mathrm{e}^{Ad_qh}
\end{aligned}
$$

Decompositions into $\mathit{\Lambda}$ operators [JO *JPhysA* **56** (2023); McLachlan *JSciComp* **16** (1995)]

$$
\begin{aligned}
\mathrm{e}^{(A+B)h+\mathcal{O}\left(h^{n+1}\right)} &= \mathrm{e}^{Aa_1h}\,\mathrm{e}^{Bb_1h}\,\mathrm{e}^{Aa_2h}\cdots\mathrm{e}^{Bb_qh}\,\mathrm{e}^{Aa_{q+1}h} \\
&= \mathrm{e}^{Ac_1h}\,\mathrm{e}^{Bc_1h}\,\mathrm{e}^{Bd_1h}\,\mathrm{e}^{Ad_1h}\cdots\mathrm{e}^{Bd_qh}\,\mathrm{e}^{Ad_qh}
\end{aligned}
$$

$$
\Rightarrow \mathrm{e}^{h\sum\limits_{k=1}^{\mathit{\Lambda}} A_k+\mathcal{O}\left(h^{n+1}\right)}
$$

$$
= \left(\prod_{k=1}^{\mathit{\Lambda}} \mathrm{e}^{A_kc_1h}\right)\left(\prod_{k=\mathit{\Lambda}}^{1} \mathrm{e}^{A_kd_1h}\right)\cdots\left(\prod_{k=1}^{\mathit{\Lambda}} \mathrm{e}^{A_kc_qh}\right)\left(\prod_{k=\mathit{\Lambda}}^{1} \mathrm{e}^{A_kd_qh}\right)
$$

Time-dependent Hamiltonian [Ikeda + *Quantum* **7** (2023); Suzuki *ProcJapAcB* **69** (1993)]

$$\mathrm{i}\,\frac{\mathrm{d}}{\mathrm{d}t}\,|\psi(t)\rangle = H(t)\,|\psi(t)\rangle$$

$$\Rightarrow |\psi(t_1)\rangle = U(t_1,t_0)\,|\psi(t_0)\rangle$$

Time-dependent Hamiltonian [Ikeda + *Quantum* **7** (2023); Suzuki *ProcJapAcB* **69** (1993)]

$$\mathrm{i}\,\frac{\mathrm{d}}{\mathrm{d}t}\,|\psi(t)\rangle = H(t)\,|\psi(t)\rangle$$

$$\Rightarrow |\psi(t_1)\rangle = U(t_1,t_0)\,|\psi(t_0)\rangle$$

$$U(t_1,t_0) \equiv \mathcal{T}\left\{\mathrm{e}^{-\mathrm{i}\int_{t_0}^{t_1}\mathrm{d}t H(t)}\right\}$$

Time-dependent Hamiltonian [Ikeda+ *Quantum* **7** (2023); Suzuki *ProcJapAcB* **69** (1993)]

$$
\begin{aligned}
\mathrm{i}\,\frac{\mathrm{d}}{\mathrm{d}t}\,|\psi(t)\rangle &= H(t)\,|\psi(t)\rangle \\
\Rightarrow |\psi(t_1)\rangle &= U(t_1,t_0)\,|\psi(t_0)\rangle \\
U(t_1,t_0) &\equiv \mathcal{T}\left\{\mathrm{e}^{-\mathrm{i}\int_{t_0}^{t_1}\mathrm{d}t H(t)}\right\} \\
&= \lim_{N_t\to\infty}\left(\mathrm{e}^{-\mathrm{i}\,H(t_1-h)h}\cdots\mathrm{e}^{-\mathrm{i}\,H(t_0+h)h}\mathrm{e}^{-\mathrm{i}\,H(t_0)h}\right)
\end{aligned}
$$

Time-dependent Hamiltonian [Ikeda + *Quantum* **7** (2023); Suzuki *ProcJapAcB* **69** (1993)]

$$
\begin{aligned}
\mathrm{i}\,\frac{\mathrm{d}}{\mathrm{d}t}\,|\psi(t)\rangle &= H(t)\,|\psi(t)\rangle \\
\Rightarrow |\psi(t_1)\rangle &= U(t_1,t_0)\,|\psi(t_0)\rangle \\
U(t_1,t_0) &\equiv \mathcal{T}\left\{\mathrm{e}^{-\mathrm{i}\int_{t_0}^{t_1}\mathrm{d}t H(t)}\right\} \\
&= \lim_{N_t\to\infty}\left(\mathrm{e}^{-\mathrm{i}\,H(t_1-h)h}\cdots\mathrm{e}^{-\mathrm{i}\,H(t_0+h)h}\mathrm{e}^{-\mathrm{i}\,H(t_0)h}\right)
\end{aligned}
$$

- Introduce symbolic time-shift operator:

$$
\mathrm{e}^{-\mathrm{i}\,A_k(t)}\mathrm{e}^{-\mathrm{i}\,\mathcal{T}h} := \mathrm{e}^{-\mathrm{i}\,\mathcal{T}h}\mathrm{e}^{-\mathrm{i}\,A_k(t+h)}
$$

Time-dependent Hamiltonian [Ikeda+ *Quantum* **7** (2023); Suzuki *ProcJapAcB* **69** (1993)]

$$
\begin{aligned}
\mathrm{i}\frac{\mathrm{d}}{\mathrm{d}t}|\psi(t)\rangle &= H(t)|\psi(t)\rangle \\
\Rightarrow |\psi(t_1)\rangle &= U(t_1,t_0)|\psi(t_0)\rangle \\
U(t_1,t_0) &\equiv \mathcal{T}\left\{\mathrm{e}^{-\mathrm{i}\int_{t_0}^{t_1}\mathrm{d}t H(t)}\right\} \\
&= \lim_{N_t\to\infty}\left(\mathrm{e}^{-\mathrm{i}H(t_1-h)h}\cdots\mathrm{e}^{-\mathrm{i}H(t_0+h)h}\mathrm{e}^{-\mathrm{i}H(t_0)h}\right)
\end{aligned}
$$

- Introduce symbolic time-shift operator:

$$
\mathrm{e}^{-\mathrm{i}A_k(t)}\mathrm{e}^{-\mathrm{i}\mathcal{T}h} := \mathrm{e}^{-\mathrm{i}\mathcal{T}h}\mathrm{e}^{-\mathrm{i}A_k(t+h)}
$$

$$
\Rightarrow \mathcal{T}\left\{\mathrm{e}^{-\mathrm{i}\int_t^{t+h}\mathrm{d}s H(s)}\right\} = \mathrm{e}^{-\mathrm{i}(H(t)+\mathcal{T})h}
$$

Generalised Trotter formula for $H(t) = \sum_{k=1}^{\Lambda} A_k(t)$

[Suzuki *ProcJapAcB* **69** (1993); Hatano & Suzuki *math-ph*/**0506007**]

$$\mathcal{T}\left\{\mathrm{e}^{-\mathrm{i}\int_t^{t+h}\mathrm{d}s H(s)}\right\} = \mathrm{e}^{-\mathrm{i}(H(t)+\mathcal{T})h}$$
$$= S_n(t,h) + \mathcal{O}\left(h^{n+1}\right)$$

Generalised Trotter formula for $H(t) = \sum_{k=1}^{\Lambda} A_k(t)$

[Suzuki *ProcJapAcB* **69** (1993); Hatano & Suzuki *math-ph*/**0506007**]

$$\mathcal{T}\left\{\mathrm{e}^{-\mathrm{i}\int_t^{t+h}\mathrm{d}s H(s)}\right\} = \mathrm{e}^{-\mathrm{i}(H(t)+\mathcal{T})h}$$

$$= S_n(t,h) + \mathcal{O}\left(h^{n+1}\right)$$

$$\Rightarrow S_n(t,h) = \prod_{i=1}^{q}\left[\left(\prod_{k=1}^{\Lambda}\mathrm{e}^{c_i A_k(t+e_i h)h}\right)\left(\prod_{k=\Lambda}^{1}\mathrm{e}^{d_i A_k(t+e_i h)h}\right)\right],$$

$$e_i := \sum_{j=i}^{q} d_j + \sum_{j=i+1}^{q} c_q$$

Asymmetric Trotterization

$$H = H_0 + \varepsilon H_1 \,, \quad \varepsilon \ll 1$$

Asymmetric Trotterization

$$H = H_0 + \varepsilon H_1\,, \quad \varepsilon \ll 1$$

- Remove dominant error [Omelyan + *CPC* **146** (2002)]:

$$\mathrm{e}^{ah\varepsilon H_1}\mathrm{e}^{\frac{1}{2}hH_0}\mathrm{e}^{(1-2a)h\varepsilon H_1}\mathrm{e}^{\frac{1}{2}hH_0}\mathrm{e}^{ah\varepsilon H_1} = \mathrm{e}^{h(H_0+\varepsilon H_1)+h^3\left(\alpha\varepsilon[H_0,[H_0,H_1]]+\beta\varepsilon^2[H_1,[H_0,H_1]]\right)+\mathcal{O}\left(h^5\right)}\,,$$

$$\alpha = \frac{1-6a}{24} \quad \Rightarrow \text{ set } a = \frac{1}{6}$$

Asymmetric Trotterization

$$H = H_0 + \varepsilon H_1\,, \quad \varepsilon \ll 1$$

- Remove dominant error [Omelyan + *CPC* **146** (2002)]:

$$\mathrm{e}^{ah\varepsilon H_1}\mathrm{e}^{\frac{1}{2}hH_0}\mathrm{e}^{(1-2a)h\varepsilon H_1}\mathrm{e}^{\frac{1}{2}hH_0}\mathrm{e}^{ah\varepsilon H_1} = \mathrm{e}^{h(H_0+\varepsilon H_1)+h^3\left(\alpha\varepsilon[H_0,[H_0,H_1]]+\beta\varepsilon^2[H_1,[H_0,H_1]]\right)+\mathcal{O}\left(h^5\right)}\,,$$

$$\alpha = \frac{1-6a}{24} \quad \Rightarrow \ \text{set } a = \frac{1}{6}$$

- Interaction picture [Low & Wiebe *quant-ph*/**1805**.00675; Rajput, Roggero + *Quantum* **6** (2022); Ciavarella & Bauer *PRL* **133** (2024); Bosse + *NatCommun* **16** (2025)]:

$$H_1 = \sum_{\gamma=1}^{\Gamma} H_1^{\gamma}\,,$$

$$\mathrm{e}^{-ihH} = \mathrm{e}^{-ihH_0}\prod_{\gamma}\left(\mathrm{e}^{ihH_0}\mathrm{e}^{-ih(H_0+\varepsilon H_1^{\gamma})}\right) + \mathcal{O}\left(\varepsilon^2 h^2\right)$$

Choose your own adventure!

1. Processed methods
2. Multi-product formulae
3. Randomised methods
4. Symplectic integrators
5. Polynomial expansions
6. Quantum circuit optimisation
7. TDVP for tensor networks
8. Integration on manifolds

Processed methods

[Butcher *ConfNumDiffEq* (1969); Blanes + *AplNumMat* **204** (2024); Morales + *QuantInfComp* **25** (2025)]

$$S(h) = P(h)\,\hat{S}(h)\,P(h)^{-1}$$
$$\Rightarrow S(h)^N = P(h)\,\hat{S}(h)^N\,P(h)^{-1}$$

- Kernel $\hat{S}$ cheap, processor P rarely needed

Processed methods

[Butcher *ConfNumDiffEq* (1969); Blanes+ *AplNumMat* **204** (2024); Morales+ *QuantInfComp* **25** (2025)]

$$S(h) = P(h)\,\hat{S}(h)\,P(h)^{-1}$$
$$\Rightarrow S(h)^N = P(h)\,\hat{S}(h)^N\,P(h)^{-1}$$

- Kernel $\hat{S}$ cheap, processor P rarely needed
- Can help with efficient long-time evolution

Processed methods

[Butcher *ConfNumDiffEq* (1969); Blanes + *AplNumMat* **204** (2024); Morales + *QuantInfComp* **25** (2025)]

$$
\begin{aligned}
S(h) &= P(h)\,\hat{S}(h)\,P(h)^{-1} \\
\Rightarrow S(h)^N &= P(h)\,\hat{S}(h)^N\,P(h)^{-1}
\end{aligned}
$$

- Kernel $\hat{S}$ cheap, processor P rarely needed
- Can help with efficient long-time evolution
- Useless if intermediate measurements are required

Choose your own adventure!

1. Processed methods
2. Multi-product formulae
3. Randomised methods
4. Symplectic integrators
5. Polynomial expansions
6. Quantum circuit optimisation
7. TDVP for tensor networks
8. Integration on manifolds

Multi-product formulae

[Blanes + *CelMechDynAst* **75** (1999); Chin *CelMechDynAst* **106** (2008); Low + *quant-ph*/**1907**.11679]

- Linear combinations of order n_0 product formula S_{n_0}

$$S^{(s)}_{n,n_0}(h) = \sum_{j=1}^{m} a_j \, S_{n_0}\left(\frac{h}{k_j}\right)^{k_j}$$

Multi-product formulae

[Blanes + *CelMechDynAst* **75** (1999); Chin *CelMechDynAst* **106** (2008); Low + *quant-ph*/**1907**.11679]

- Linear combinations of order n_0 product formula S_{n_0}
- Effective order $n = n_0 + 2(m-1)$

$$S^{(s)}_{n,n_0}(h) = \sum_{j=1}^{m} a_j \, S_{n_0}\left(\frac{h}{k_j}\right)^{k_j}$$

Multi-product formulae

[Blanes + *CelMechDynAst* **75** (1999); Chin *CelMechDynAst* **106** (2008); Low + *quant-ph*/**1907**.11679]

- Linear combinations of order n_0 product formula S_{n_0}
- Effective order $n = n_0 + 2(m-1)$
- Unitary to order $s = \max(2n_0 + 1, n + 3)$

$$S_{n,n_0}^{(s)}(h) = \sum_{j=1}^{m} a_j \, S_{n_0}\left(\frac{h}{k_j}\right)^{k_j}$$

Multi-product formulae

[Blanes + *CelMechDynAst* **75** (1999); Chin *CelMechDynAst* **106** (2008); Low + *quant-ph*/**1907**.11679]

- Linear combinations of order n_0 product formula S_{n_0}
- Effective order $n = n_0 + 2(m-1)$
- Unitary to order $s = \max\left(2n_0+1, n+3\right)$

$$S^{(s)}_{n,n_0}(h) = \sum_{j=1}^{m} a_j \, S_{n_0}\left(\frac{h}{k_j}\right)^{k_j}$$

$$a_j = \prod_{l \neq j} \frac{k_j^2}{k_j^2 - k_l^2} \quad \text{for } n_0 = 2$$

Multi-product formulae

[Blanes + *CelMechDynAst* **75** (1999); Chin *CelMechDynAst* **106** (2008); Low + *quant-ph*/**1907**.11679]

- Linear combinations of order n_0 product formula S_{n_0}
- Effective order $n = n_0 + 2(m-1)$
- Unitary to order $s = \max(2n_0 + 1, n + 3)$

$$S^{(s)}_{n,n_0}(h) = \sum_{j=1}^{m} a_j \, S_{n_0}\left(\frac{h}{k_j}\right)^{k_j}$$

$$a_j = \prod_{l \neq j} \frac{k_j^2}{k_j^2 - k_l^2} \quad \text{for } n_0 = 2$$

Quantum computing post-processing:

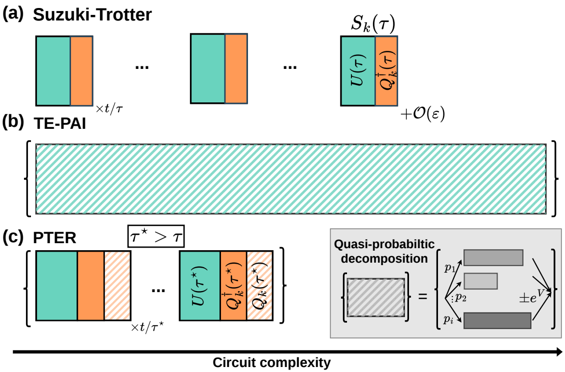
[Carrera Vazquez + *Quantum* **7** (2023); Childs & Wiebe *QantInfComp* **12** (2012); Park + *quant-ph*/**2602**.01713]

Choose your own adventure!

1. Processed methods
2. Multi-product formulae
3. Randomised methods
4. Symplectic integrators
5. Polynomial expansions
6. Quantum circuit optimisation
7. TDVP for tensor networks
8. Integration on manifolds

Randomised methods

[Campbell *PRL* **123** (2019); Granet & Dreyer *QuantInf* **10** (2024); Kiss, Roggero + *Quantum* **7** (2023); Kiumi & Koczor *QuantSciTech* **10** (2025); Hayata & Kikuchi *quant-ph*/**2604**.02854; Wang, Sun + *quant-ph*/**2607**.11856] [Murota, Rinaldi + *quant-ph*/**2606**.29741]

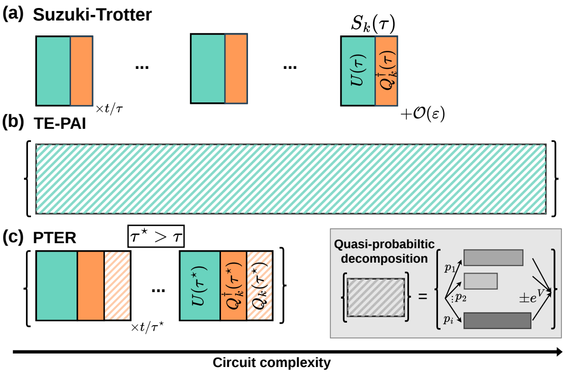


- Sample few gates randomly, repeat many times

Randomised methods

[Campbell *PRL* **123** (2019); Granet & Dreyer *QuantInf* **10** (2024); Kiss, Roggero + *Quantum* **7** (2023); Kiumi & Koczor *QuantSciTech* **10** (2025); Hayata & Kikuchi *quant-ph*/**2604**.02854; Wang, Sun + *quant-ph*/**2607**.11856]

[Murota, Rinaldi + *quant-ph*/**2606**.29741]

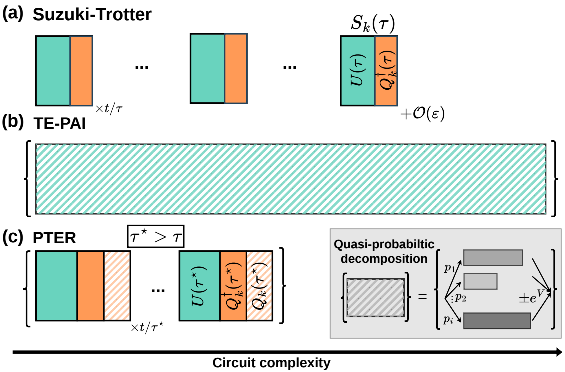


- Sample few gates randomly, repeat many times
- Reduces circuit depth

Randomised methods

[Campbell *PRL* **123** (2019); Granet & Dreyer *QuantInf* **10** (2024); Kiss, Roggero + *Quantum* **7** (2023); Kiumi & Koczor *QuantSciTech* **10** (2025); Hayata & Kikuchi *quant-ph*/**2604**.02854; Wang, Sun + *quant-ph*/**2607**.11856]

[Murota, Rinaldi + *quant-ph*/**2606**.29741]

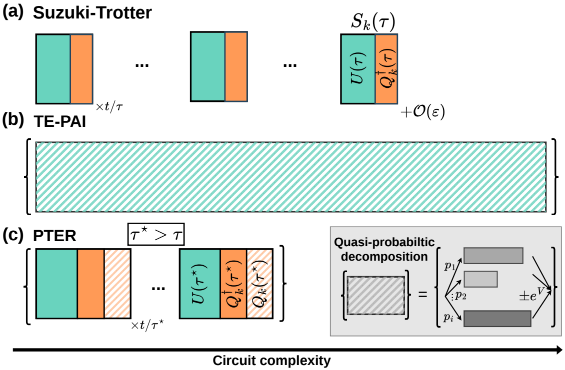


- Sample few gates randomly, repeat many times
- Reduces circuit depth
- Can eliminate Trotter error (continuous formulation)

  [Granet & Dreyer *QuantInf* **10** (2024); Murota, Rinaldi + *quant-ph*/**2606**.29741; Hayata & Kikuchi *quant-ph*/**2604**.02854]

Randomised methods

[Campbell *PRL* **123** (2019); Granet & Dreyer *QuantInf* **10** (2024); Kiss, Roggero + *Quantum* **7** (2023); Kiumi & Koczor *QuantSciTech* **10** (2025); Hayata & Kikuchi *quant-ph*/**2604**.02854; Wang, Sun + *quant-ph*/**2607**.11856] [Murota, Rinaldi + *quant-ph*/**2606**.29741]



- Sample few gates randomly, repeat many times
- Reduces circuit depth
- Can eliminate Trotter error (continuous formulation)

  [Granet & Dreyer *QuantInf* **10** (2024); Murota, Rinaldi + *quant-ph*/**2606**.29741; Hayata & Kikuchi *quant-ph*/**2604**.02854]
- Introduces stochastic error

Choose your own adventure!

1. Processed methods
2. Multi-product formulae
3. Randomised methods
4. Symplectic integrators
5. Polynomial expansions
6. Quantum circuit optimisation
7. TDVP for tensor networks
8. Integration on manifolds

Symplectic integrators [Omelyan + *CPC* **151** (2003)]

- Numerically solve equations of motion

$$
\begin{aligned}
\dot{x} &= p \\
\dot{p} &= -V'(x)
\end{aligned}
$$

Symplectic integrators [Omelyan+ *CPC* **151** (2003)]

- Numerically solve equations of motion

$$\dot{x} = p$$
$$\dot{p} = -V'(x)$$

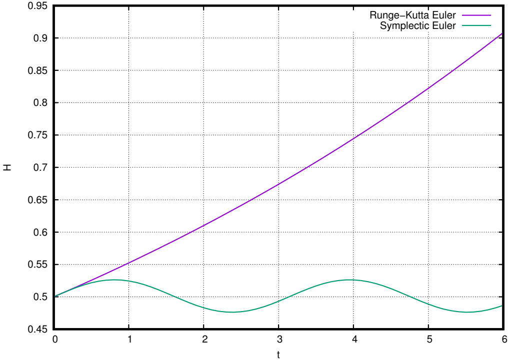
- Preserve phase space $\Rightarrow$ error of $H = \frac{1}{2}p^2 + V(x)$ bounded.

Symplectic integrators [Omelyan + *CPC* **151** (2003)]

- Numerically solve equations of motion

$$\dot{x} = p$$
$$\dot{p} = -V'(x)$$

- Preserve phase space $\Rightarrow$ error of $H = \frac{1}{2}p^2 + V(x)$ bounded.

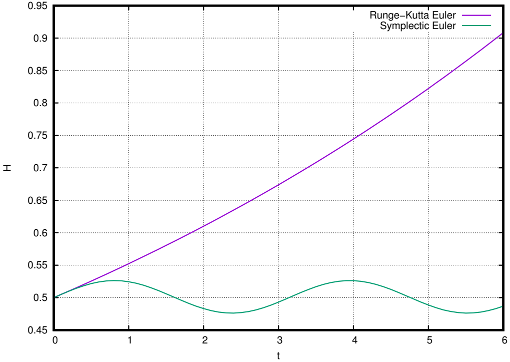


Symplectic integrators [Omelyan+ *CPC* **151** (2003)]

- Numerically solve equations of motion

$$\dot{x} = p$$
$$\dot{p} = -V'(x)$$

- Preserve phase space $\Rightarrow$ error of $H = \frac{1}{2}p^2 + V(x)$ bounded.



$$x_1 = x_0 + a_1 h p_0$$
$$p_1 = p_0 - b_1 h V'(x_1)$$
$$x_2 = x_1 + a_2 h p_1$$
$$\vdots$$

Comparison to Runge-Kutta

$$\begin{aligned} H &= T(p) + V(x)\,, \\ \dot{x} &= T'(p) \\ \dot{p} &= -V'(x) \end{aligned}$$

Explicit Runge-Kutta	Symplectic	Implicit Runge-Kutta
$x_{n+1} = x_n + hT'(p_n)$	$x_{n+1} = x_n + hT'(p_n)$	$x_{n+1} = x_n + hT'(p_{n+1})$
$p_{n+1} = p_n - hV'(x_n)$	$p_{n+1} = p_n - hV'(x_{n+1})$	$p_{n+1} = p_n - hV'(x_{n+1})$

Special case of Trotterization [Omelyan+ *CPC* **151** (2003)]

$$\begin{pmatrix} x(t) \\ p(t) \end{pmatrix} = \mathrm{e}^{t\left(p\frac{\partial}{\partial x} - V'(x)\frac{\partial}{\partial p}\right)} \begin{pmatrix} x(0) \\ p(0) \end{pmatrix}$$

Special case of Trotterization [Omelyan + *CPC* **151** (2003)]

$$\begin{pmatrix} x(t) \\ p(t) \end{pmatrix} = \mathrm{e}^{t\left(p\frac{\partial}{\partial x} - V'(x)\frac{\partial}{\partial p}\right)} \begin{pmatrix} x(0) \\ p(0) \end{pmatrix}$$

$$=: \mathrm{e}^{t(A+B)} \begin{pmatrix} x(0) \\ p(0) \end{pmatrix}$$

Special case of Trotterization [Omelyan + *CPC* **151** (2003)]

$$\begin{pmatrix} x(t) \\ p(t) \end{pmatrix} = \mathrm{e}^{t\left(p\frac{\partial}{\partial x} - V'(x)\frac{\partial}{\partial p}\right)} \begin{pmatrix} x(0) \\ p(0) \end{pmatrix}$$

$$=: \mathrm{e}^{t(A+B)} \begin{pmatrix} x(0) \\ p(0) \end{pmatrix}$$

$$\Rightarrow [B,[B,[A,B]]] = 0$$

Special case of Trotterization [Omelyan + *CPC* **151** (2003)]

$$\begin{pmatrix} x(t) \\ p(t) \end{pmatrix} = \mathrm{e}^{t\left(p\frac{\partial}{\partial x} - V'(x)\frac{\partial}{\partial p}\right)} \begin{pmatrix} x(0) \\ p(0) \end{pmatrix}$$

$$=: \mathrm{e}^{t(A+B)} \begin{pmatrix} x(0) \\ p(0) \end{pmatrix}$$

$$\Rightarrow [B,[B,[A,B]]] = 0$$

$$\mathcal{O}_3 = \alpha[A,[A,B]] + \beta[B,[A,B]]$$

Special case of Trotterization [Omelyan+ *CPC* **151** (2003)]

$$\begin{pmatrix} x(t) \\ p(t) \end{pmatrix} = \mathrm{e}^{t\left(p\frac{\partial}{\partial x} - V'(x)\frac{\partial}{\partial p}\right)} \begin{pmatrix} x(0) \\ p(0) \end{pmatrix}$$

$$=: \mathrm{e}^{t(A+B)} \begin{pmatrix} x(0) \\ p(0) \end{pmatrix}$$

$$\Rightarrow [B,[B,[A,B]]] = 0$$

$$\mathcal{O}_3 = \alpha[A,[A,B]] + \beta[B,[A,B]]$$

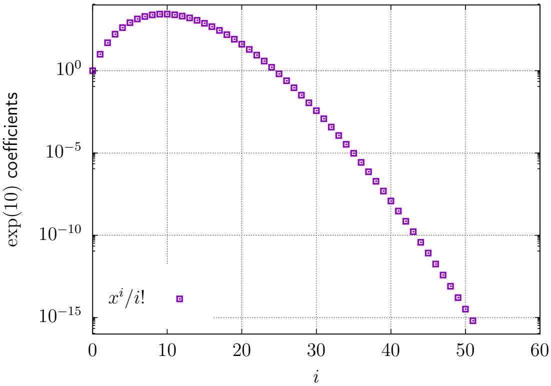
$$\Rightarrow \mathcal{O}_5 = \gamma_1[A,[A,[A,[A,B]]]] + \gamma_2[A,[A,[B,[A,B]]]] + \gamma_3[B,[A,[A,[A,B]]]] + \cancel{\gamma_4[B,[B,[B,[A,B]]]]} + \gamma_5[B,[B,[A,[A,B]]]] + \cancel{\gamma_6[A,[B,[B,[A,B]]]]}$$

Choose your own adventure!

1. Processed methods
2. Multi-product formulae
3. Randomised methods
4. Symplectic integrators
5. Polynomial expansions
6. Quantum circuit optimisation
7. TDVP for tensor networks
8. Integration on manifolds

Polynomial expansions

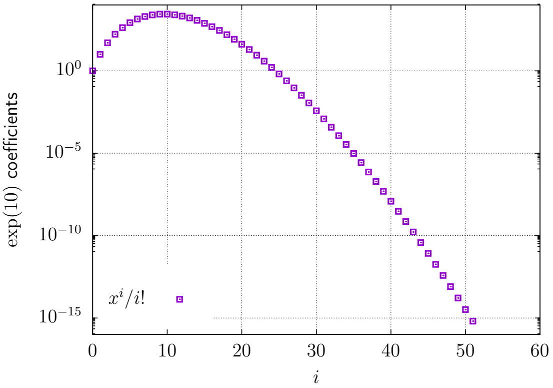
$$\mathrm{e}^{Hh} = \sum_{i=0}^{k} \frac{(Hh)^i}{i!} + \mathcal{O}\left(h^{k+1}\right)$$



Polynomial expansions

$$\mathrm{e}^{Hh} = \sum_{i=0}^{k} \frac{(Hh)^i}{i!} + \mathcal{O}\left(h^{k+1}\right)$$

$$\Rightarrow \mathrm{e}^{Hh}|\psi\rangle = |\psi\rangle + \left(hH\left(|\psi\rangle + \cdots + \frac{h}{n-1}H\left(|\psi\rangle + \frac{h}{n}H|\psi\rangle\right)\cdots\right)\right) + \mathcal{O}\left(h^{k+1}\right) \quad \textbf{(Horner)}$$



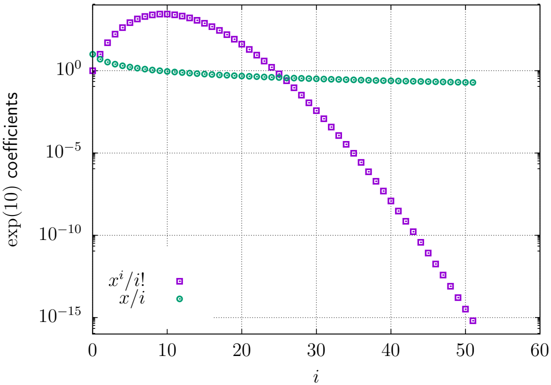
Polynomial expansions

$$e^{Hh} = \sum_{i=0}^{k} \frac{(Hh)^i}{i!} + \mathcal{O}\left(h^{k+1}\right)$$

$$\Rightarrow e^{Hh}|\psi\rangle = |\psi\rangle + \left(hH\left(|\psi\rangle + \cdots \right.\right.$$

$$\left.\left. + \frac{h}{n-1}H\left(|\psi\rangle + \frac{h}{n}H|\psi\rangle\right)\cdots\right)\right)$$

$$+\mathcal{O}\left(h^{k+1}\right) \quad \textbf{(Horner)}$$

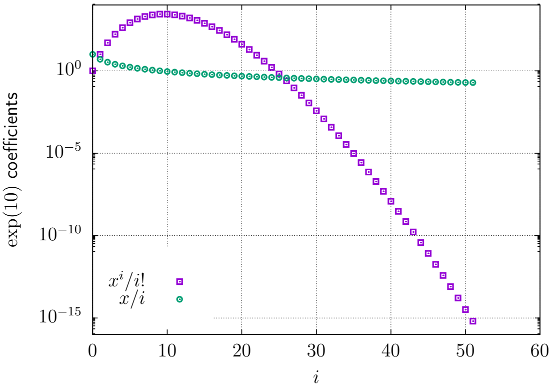


Polynomial expansions

$$e^{Hh} = \sum_{i=0}^{k} \frac{(Hh)^i}{i!} + \mathcal{O}\left(h^{k+1}\right)$$

$$\Rightarrow e^{Hh}|\psi\rangle = |\psi\rangle + (hH(|\psi\rangle + \cdots$$
$$+ \frac{h}{n-1} H \left(|\psi\rangle + \frac{h}{n} H |\psi\rangle \right) \cdots))$$
$$+ \mathcal{O}\left(h^{k+1}\right) \qquad \textbf{(Horner)}$$

$$\sum_{i=0}^{k} \frac{(Hh)^i}{i!} = \prod_{i=1}^{k} \left(1 + a_i^k Hh\right)$$

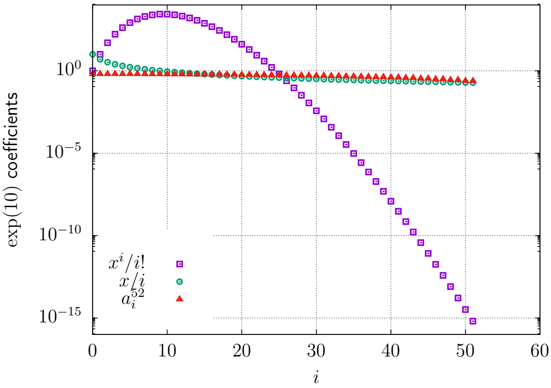


Polynomial expansions

$$e^{Hh} = \sum_{i=0}^{k} \frac{(Hh)^i}{i!} + \mathcal{O}\left(h^{k+1}\right)$$

$$\Rightarrow e^{Hh}|\psi\rangle = |\psi\rangle + \left(hH\left(|\psi\rangle + \cdots + \frac{h}{n-1}H\left(|\psi\rangle + \frac{h}{n}H|\psi\rangle\right)\cdots\right)\right) + \mathcal{O}\left(h^{k+1}\right) \quad \textbf{(Horner)}$$

$$\sum_{i=0}^{k} \frac{(Hh)^i}{i!} = \prod_{i=1}^{k}\left(1 + a_i^k Hh\right)$$



Taylor vs. Chebyshev expansion [JO *PoS* **LATTICE** (2023)]

$$
\mathrm{e}^{Hh} = \lim_{k\to\infty} \sum_{i=0}^{k} \mu_i T_i \left(\frac{H}{\Gamma}\right) ,
$$

$$
\mu_0 = I_0(\Gamma h)\,,\; \mu_i = 2I_i(\Gamma h)\,,
$$

Taylor vs. Chebyshev expansion [JO *PoS* **LATTICE** (2023)]

$$
\begin{aligned}
\mathrm{e}^{Hh} &= \lim_{k\to\infty} \sum_{i=0}^{k} \mu_i T_i\left(\frac{H}{\Gamma}\right) \,, \\
\mu_0 &= I_0(\Gamma h)\,,\; \mu_i = 2I_i(\Gamma h)\,, \\
T_0(H) &= \mathbb{1}\,,
\end{aligned}
$$

Taylor vs. Chebyshev expansion [JO *PoS* **LATTICE** (2023)]

$$
\begin{aligned}
\mathrm{e}^{Hh} &= \lim_{k\to\infty} \sum_{i=0}^{k} \mu_i T_i \left(\frac{H}{\Gamma}\right) \,, \\
\mu_0 &= I_0(\Gamma h) \,, \; \mu_i = 2 I_i(\Gamma h) \,, \\
T_0(H) &= \mathbb{1} \,, \\
T_1(H) &= H \,,
\end{aligned}
$$

Taylor vs. Chebyshev expansion [JO *PoS* **LATTICE** (2023)]

$$
\begin{aligned}
\mathrm{e}^{Hh} &= \lim_{k\to\infty} \sum_{i=0}^{k} \mu_i T_i \left(\frac{H}{\Gamma} \right) , \\
\mu_0 &= I_0(\Gamma h) \, , \, \mu_i = 2 I_i(\Gamma h) \, , \\
T_0(H) &= \mathbb{1} \, , \\
T_1(H) &= H \, , \\
T_{i+1}(H) &= 2H \cdot T_i(H) - T_{i-1}(H)
\end{aligned}
$$

Taylor vs. Chebyshev expansion [JO *PoS* **LATTICE** (2023)]

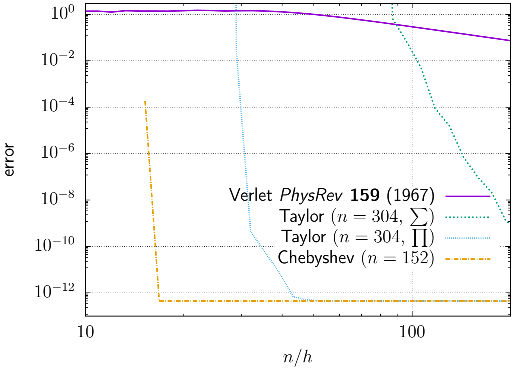
$$\mathrm{e}^{Hh} = \lim_{k\to\infty} \sum_{i=0}^{k} \mu_i T_i \left(\frac{H}{\Gamma}\right) \,,$$

$$\mu_0 = I_0(\Gamma h) \,, \; \mu_i = 2I_i(\Gamma h) \,,$$

$$T_0(H) = \mathbb{1} \,,$$

$$T_1(H) = H \,,$$

$$T_{i+1}(H) = 2H \cdot T_i(H) - T_{i-1}(H)$$



Choose your own adventure!

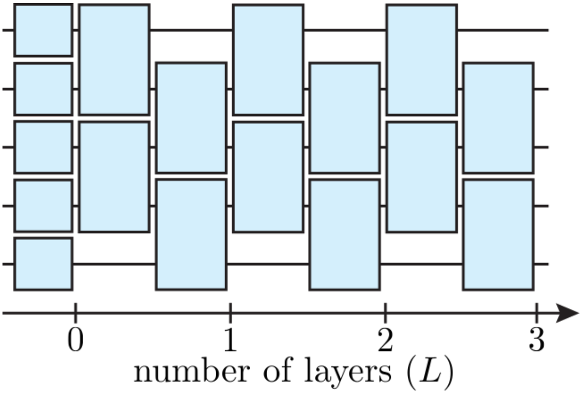
1. Processed methods
2. Multi-product formulae
3. Randomised methods
4. Symplectic integrators
5. Polynomial expansions
6. Quantum circuit optimisation
7. TDVP for tensor networks
8. Integration on manifolds

Quantum circuit optimisation

[Le + *Quantum* **9** (2025); Mansuroglu + *QuantSciTech* **8** (2023); Tepaske + *SciPostPhys* **14** (2023)]

[Mc Keever & Lubasch *PRR* **5** (2023)]

Ansatz: brickwall circuit

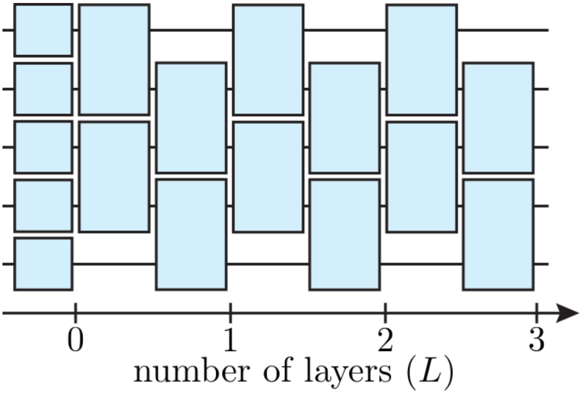


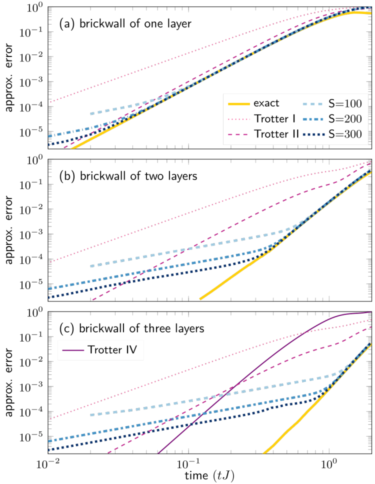
Quantum circuit optimisation

[Le + *Quantum* **9** (2025); Mansuroglu + *QuantSciTech* **8** (2023); Tepaske + *SciPostPhys* **14** (2023)]

[Mc Keever & Lubasch *PRR* **5** (2023)]

Ansatz: brickwall circuit





Choose your own adventure!

1. Processed methods
2. Multi-product formulae
3. Randomised methods
4. Symplectic integrators
5. Polynomial expansions
6. Quantum circuit optimisation
7. TDVP for tensor networks
8. Integration on manifolds

Time-dependent variational principle (TDVP)

[Haegeman + *PRL* **107** (2011), *PRB* **94** (2016); Paeckel + *AnnalsPhys* **411** (2019)]

- Tensor network state $\psi(X)$ parametrised by X of limited dimension D

Time-dependent variational principle (TDVP)

[Haegeman + *PRL* **107** (2011), *PRB* **94** (2016); Paeckel + *AnnalsPhys* **411** (2019)]

- Tensor network state $\psi(X)$ parametrised by X of limited dimension D
- Trotterization "time-evolving block decimation (TEBD)" increases D with every step

$$-\,\mathrm{i}\,H\,|\psi(X)\rangle = \frac{\mathrm{d}}{\mathrm{d}t}\,|\psi(X)\rangle$$

Time-dependent variational principle (TDVP)

[Haegeman + *PRL* **107** (2011), *PRB* **94** (2016); Paeckel + *AnnalsPhys* **411** (2019)]

- Tensor network state $\psi(X)$ parametrised by X of limited dimension D
- Trotterization "time-evolving block decimation (TEBD)" increases D with every step

$\Rightarrow$ Information is lost when truncating back to D

$$-\,\mathrm{i}\,H\,|\psi(X)\rangle = \frac{\mathrm{d}}{\mathrm{d}t}\,|\psi(X)\rangle$$

Time-dependent variational principle (TDVP)

[Haegeman + *PRL* **107** (2011), *PRB* **94** (2016); Paeckel + *AnnalsPhys* **411** (2019)]

- Tensor network state $\psi(X)$ parametrised by X of limited dimension D
- Trotterization "time-evolving block decimation (TEBD)" increases D with every step

$\Rightarrow$ Information is lost when truncating back to D

- TDVP directly evolves X via projection

$$
\begin{aligned}
-\mathrm{i}\, H \,|\psi(X)\rangle &= \frac{\mathrm{d}}{\mathrm{d}t}\,|\psi(X)\rangle \\
&= |\nabla_X \psi(X)\rangle \cdot \dot{X}
\end{aligned}
$$

Time-dependent variational principle (TDVP)

[Haegeman + *PRL* **107** (2011), *PRB* **94** (2016); Paeckel + *AnnalsPhys* **411** (2019)]

- Tensor network state $\psi(X)$ parametrised by X of limited dimension D
- Trotterization "time-evolving block decimation (TEBD)" increases D with every step

$\Rightarrow$ Information is lost when truncating back to D

- TDVP directly evolves X via projection

$\Rightarrow$ No additional truncation required

$$\begin{aligned} -\mathrm{i}\,H\,|\psi(X)\rangle &= \frac{\mathrm{d}}{\mathrm{d}t}\,|\psi(X)\rangle \\ &= |\nabla_X \psi(X)\rangle \cdot \dot{X} \end{aligned}$$

Time-dependent variational principle (TDVP)

[Haegeman + *PRL* **107** (2011), *PRB* **94** (2016); Paeckel + *AnnalsPhys* **411** (2019)]

- Tensor network state $\psi(X)$ parametrised by X of limited dimension D
- Trotterization "time-evolving block decimation (TEBD)" increases D with every step

$\Rightarrow$ Information is lost when truncating back to D

- TDVP directly evolves X via projection

$\Rightarrow$ No additional truncation required

$$
\begin{aligned}
-\mathrm{i}\, H \, |\psi(X)\rangle &= \frac{\mathrm{d}}{\mathrm{d}t} \, |\psi(X)\rangle \\
&= |\nabla_X \psi(X)\rangle \cdot \dot{X}
\end{aligned}
$$

$$
\Rightarrow \left\langle \nabla_{\bar{X}} \psi(\bar{X}) \,\middle|\, \nabla_X \psi(X) \right\rangle \cdot \dot{X} = -\mathrm{i} \left\langle \nabla_{\bar{X}} \psi(\bar{X}) \right| H \left| \psi(X) \right\rangle
$$

Choose your own adventure!

1. Processed methods
2. Multi-product formulae
3. Randomised methods
4. Symplectic integrators
5. Polynomial expansions
6. Quantum circuit optimisation
7. TDVP for tensor networks
8. Integration on manifolds

Numerical integration on manifolds

[Crouch & Grossman *JNonLinSci* **3** (1993); Munthe-Kaas *AplNumMat* **29** (1999)]

	Runge-Kutta	Crouch-Grossman
ODE	$\dot{z} = f(t,z)$	$\dot{\psi} = H(t,\psi) \cdot \psi$

Numerical integration on manifolds

[Crouch & Grossman *JNonLinSci* **3** (1993); Munthe-Kaas *AplNumMat* **29** (1999)]

	Runge-Kutta	Crouch-Grossman
ODE	$\dot{z} = f(t,z)$	$\dot{\psi} = H(t,\psi) \cdot \psi$
step	$z(t+h) = z(t) + h \sum_{i=1}^{s} b_i k_i$	$\psi(t+h) = \mathrm{e}^{h \sum_{i=1}^{s} \tilde{b}_i k_i} \cdot \psi(t)$

Numerical integration on manifolds

[Crouch & Grossman *JNonLinSci* **3** (1993); Munthe-Kaas *AplNumMat* **29** (1999)]

	Runge-Kutta	Crouch-Grossman
ODE	$\dot{z} = f(t,z)$	$\dot{\psi} = H(t,\psi) \cdot \psi$
step	$z(t+h) = z(t) + h \sum_{i=1}^{s} b_i k_i$	$\psi(t+h) = \mathrm{e}^{h \sum_{i=1}^{s} \tilde{b}_i k_i} \cdot \psi(t)$
	$u_i = h \sum_{j=1}^{s} a_{ij} k_j$	$u_i = h \sum_{j=1}^{s} \tilde{a}_{ij} k_j$

Numerical integration on manifolds

[Crouch & Grossman *JNonLinSci* **3** (1993); Munthe-Kaas *AplNumMat* **29** (1999)]

	Runge-Kutta	Crouch-Grossman
ODE	$\dot{z} = f(t,z)$	$\dot{\psi} = H(t,\psi) \cdot \psi$
step	$z(t+h) = z(t) + h\sum_{i=1}^{s} b_i k_i$	$\psi(t+h) = \mathrm{e}^{h\sum_{i=1}^{s} \tilde{b}_i k_i} \cdot \psi(t)$
	$u_i = h\sum_{j=1}^{s} a_{ij} k_j$	$u_i = h\sum_{j=1}^{s} \tilde{a}_{ij} k_j$
	$k_i = f\left(t + c_i h, z(t) + u_i\right)$	$k_i = H\left(t + \tilde{c}_i h, \mathrm{e}^{u_i} \cdot \psi(t)\right)$

Numerical integration on manifolds

[Crouch & Grossman *JNonLinSci* **3** (1993); Munthe-Kaas *AplNumMat* **29** (1999)]

	Runge-Kutta	Munthe-Kaas
ODE	$\dot{z} = f(t,z)$	$\dot{\psi} = H(t,\psi) \cdot \psi$
step	$z(t+h) = z(t) + h\sum_{i=1}^{s} b_i k_i$	$\psi(t+h) = \mathrm{e}^{h\sum_{i=1}^{s} b_i \tilde{k}_i} \cdot \psi(t)$
	$u_i = h\sum_{j=1}^{s} a_{ij} k_j$	$u_i = h\sum_{j=1}^{s} a_{ij} k_j$
	$k_i = f\left(t + c_i h, z(t) + u_i\right)$	$k_i = H\left(t + c_i h, \mathrm{e}^{u_i} \cdot \psi(t)\right)$
		$\tilde{k}_i := \sum_{l=1}^{n-1} \frac{\mathcal{B}_l}{l!} \mathrm{ad}_{u_i}^{l} k_i$

Choose your own adventure!

1. Processed methods
2. Multi-product formulae
3. Randomised methods
4. Symplectic integrators
5. Polynomial expansions
6. Quantum circuit optimisation
7. TDVP for tensor networks
8. Integration on manifolds

Bibliography I

[1] Blanes & Moan, “Practical symplectic partitioned Runge–Kutta and Runge–Kutta–Nyström methods”, Journal of Computational and Applied Mathematics **142**, 313–330 (*CAMath* **142** (2002)).

[2] S. Blanes, F. Casas, A. Escorihuela-Tomàs, “Families of efficient low order processed composition methods”, Applied Numerical Mathematics **204**, 86–100 (*AplNumMat* **204** (2024)).

[3] S. Blanes, F. Casas, J. Ros, “Extrapolation of Symplectic Integrators”, Celestial Mechanics and Dynamical Astronomy **75**, 149–161 (*CelMechDynAst* **75** (1999)).

[4] S. Blanes, F. Casas, M. Thalhammer, “Splitting and composition methods with embedded error estimators”, Applied Numerical Mathematics **146**, 400–415 (*AplNumMat* **146** (2019)).

[5] J. L. Bosse, A. M. Childs, C. Derby, F. M. Gambetta, A. Montanaro, R. A. Santos, “Efficient and practical Hamiltonian simulation from time-dependent product formulas”, Nature Commun. **16**, 2673 (*NatCommun* **16** (2025)).

Bibliography II

[6] D. Burgarth, P. Facchi, A. Hahn, M. Johnsson, K. Yuasa, "Strong error bounds for Trotter and strang-splittings and their implications for quantum chemistry", Phys. Rev. Res. **6**, 043155 (*PRR* **6** (2024)).

[7] J. C. Butcher, "The effective order of Runge-Kutta methods", in Conference on the Numerical Solution of Differential Equations, edited by J. L. Morris (*ConfNumDiffEq* (1969)), pp. 133–139.

[8] E. Campbell, "Random Compiler for Fast Hamiltonian Simulation", Phys. Rev. Lett. **123**, 070503 (*PRL* **123** (2019)).

[9] A. Carrera Vazquez, D. J. Egger, D. Ochsner, S. Woerner, "Well-conditioned multi-product formulas for hardware-friendly Hamiltonian simulation", Quantum **7**, 1067 (*Quantum* **7** (2023)).

[10] Y.-H. Chen, "Trotter error timescaling separation via commutant decomposition", Phys. Rev. A **111**, 022612 (*PRA* **111** (2025)).

Bibliography III

[11] A. M. Childs, Y. Su, M. C. Tran, N. Wiebe, S. Zhu, "Theory of Trotter Error with Commutator Scaling", Phys. Rev. X **11**, 011020 (*PRX* **11** (2021)).

[12] A. M. Childs, N. Wiebe, "Hamiltonian Simulation Using Linear Combinations of Unitary Operations", Quant. Inf. Comput. **12**, 0901–0924 (*QantInfComp* **12** (2012)).

[13] S. Chin, "Multi-product splitting and Runge-Kutta-Nyström integrators", Celestial Mechanics and Dynamical Astronomy **106**, 391–406 (*CelMechDynAst* **106** (2008)).

[14] A. N. Ciavarella, C. W. Bauer, "Quantum Simulation of SU(3) Lattice Yang-Mills Theory at Leading Order in Large-Nc Expansion", Phys. Rev. Lett. **133**, 111901 (*PRL* **133** (2024)).

[15] P. E. Crouch, R. Grossman, "Numerical integration of ordinary differential equations on manifolds", Journal of NonLinear Science **3**, 1–33 (*JNonLinSci* **3** (1993)).

[16] Forest & Ruth, "Fourth-order symplectic integration", Physica D: Nonlinear Phenomena **43**, 105–117 (*Physica D* **43** (1990)).

[17] E. Granet, H. Dreyer, "Hamiltonian dynamics on digital quantum computers without discretization error", npj Quantum Inf. **10**, 82 (*QuantInf* **10** (2024)).

Bibliography IV

[18] J. Haegeman, J. I. Cirac, T. J. Osborne, I. Pizorn, H. Verschelde, F. Verstraete, "Time-Dependent Variational Principle for Quantum Lattices", Phys. Rev. Lett. **107**, 070601 (*PRL* **107** (2011)).

[19] J. Haegeman, C. Lubich, I. Oseledets, B. Vandereycken, F. Verstraete, "Unifying time evolution and optimization with matrix product states", Phys. Rev. B **94**, 165116 (*PRB* **94** (2016)).

[20] A. Hahn, P. Hartung, D. Burgarth, P. Facchi, K. Yuasa, "Lower bounds for the Trotter error", Phys. Rev. A **111**, 022417 (*PRA* **111** (2025)).

[21] N. Hatano, M. Suzuki, "Finding Exponential Product Formulas of Higher Orders", in *Quantum annealing and other optimization methods* (Springer Berlin Heidelberg, *math-ph*/**0506007**), pp. 37–68.

[22] M. Heyl, Hauke, P. Zoller, "Quantum localization bounds Trotter errors in digital quantum simulation", Science Advances **5**, `10.1126/sciadv.aau8342` (*SciAdv* **5** (2019)).

Bibliography V

[23] T. N. Ikeda, A. Abrar, I. L. Chuang, S. Sugiura, "Minimum Trotterization Formulas for a Time-Dependent Hamiltonian", Quantum **7**, 1168 (*Quantum* **7** (2023)).

[24] JO, *j-ostmeyer/efficient-trotter: Optimised Trotter Decompositions*, version v1.0.0, Nov. 2022.

[25] JO, "Optimised Trotter decompositions for classical and quantum computing", J. Phys. A **56**, 285303 (*JPhysA* **56** (2023)).

[26] JO, "Simple Ways to improve Discrete Time Evolution", in 40th International Symposium on Lattice Field Theory — PoS(LATTICE2023) (Sept. *PoS* **LATTICE** (2023)), p. 025.

[27] O. Kiss, Roggero, M. Grossi, "Importance sampling for stochastic quantum simulations", Quantum **7**, 977 (*Quantum* **7** (2023)).

[28] C. Kiumi, B. Koczor, "TE-PAI: Exact Time Evolution by Sampling Random Circuits", Quantum Sci. Technol. **10**, 045071 (*QuantSciTech* **10** (2025)).

[29] I. N. M. Le, S. Sun, C. B. Mendl, "Riemannian quantum circuit optimization based on matrix product operators", Quantum **9**, 1833 (*Quantum* **9** (2025)).

Bibliography VI

[30] G. H. Low, V. Kliuchnikov, N. Wiebe, "Well-conditioned multiproduct Hamiltonian simulation", (*quant-ph*/**1907**.11679).

[31] G. H. Low, N. Wiebe, "Hamiltonian Simulation in the Interaction Picture", (*quant-ph*/**1805**.00675).

[32] R. Mansuroglu, T. Eckstein, L. Nützel, S. A. Wilkinson, M. J. Hartmann, "Variational Hamiltonian simulation for translational invariant systems via classical pre-processing", Quantum Science and Technology **8**, 025006 (*QuantSciTech* **8** (2023)).

[33] C. Mc Keever, M. Lubasch, "Classically optimized hamiltonian simulation", Physical Review Research **5**, `10.1103/physrevresearch.5.023146` (*PRR* **5** (2023)).

[34] R. I. McLachlan, "On the Numerical Integration of Ordinary Differential Equations by Symmetric Composition Methods", SIAM Journal on Scientific Computing **16**, 151–168 (*JSciComp* **16** (1995)).

Bibliography VII

[35]M. E. S. Morales, P. C. S. Costa, G. Pantaleoni, D. K. Burgarth, Y. R. Sanders, D. W. Berry, "Selection and Improvement of Product Formulae for Best Performance of Quantum Simulation", Quant. Inf. Comput. **25**, 1–35 (*QuantInfComp* **25** (2025)).

[36]H. Munthe-Kaas, "High order Runge-Kutta methods on manifolds", Applied Numerical Mathematics **29**, Proceedings of the NSF/CBMS Regional Conference on Numerical Analysis of Hamiltonian Differential Equations, 115–127 (*AplNumMat* **29** (1999)).

[37]K. Murota, Rinaldi, Y. Kikuchi, F. Sauvage, S. Todo, "Unbiased Hamiltonian Simulation by Reversing Trotter Error Dynamics", (*quant-ph*/**2606**.29741).

[38]I. Omelyan, I. Mryglod, R. Folk, "Optimized Forest–Ruth- and Suzuki-like algorithms for integration of motion in many-body systems", Computer Physics Communications **146**, 188–202 (*CPC* **146** (2002)).

Bibliography VIII

[39] I. Omelyan, I. Mryglod, R. Folk, "Symplectic analytically integrable decomposition algorithms: classification, derivation, and application to molecular dynamics, quantum and celestial mechanics simulations", Computer Physics Communications **151**, 272–314 (*CPC* **151** (2003)).

[40] S. Paeckel, T. Köhler, A. Swoboda, S. R. Manmana, U. Schollwöck, C. Hubig, "Time-evolution methods for matrix-product states", Annals Phys. **411**, 167998 (*AnnalsPhys* **411** (2019)).

[41] S. Park, S. Lee, K. Baek, "Dual channel multi-product formulas", (*quant-ph*/**2602**.01713).

[42] A. Rajput, Roggero, N. Wiebe, "Hybridized Methods for Quantum Simulation in the Interaction Picture", Quantum **6**, 780 (*Quantum* **6** (2022)).

[43] A. Schubert, C. B. Mendl, "Trotter error with commutator scaling for the Fermi-Hubbard model", GitHub repository, https://github.com/qc-tum/fermi_hubbard_commutators.git (*GitHub* (2023)).

Bibliography IX

[44] A. Schubert, C. B. Mendl, "Trotter error with commutator scaling for the fermi-hubbard model", Physical Review B **108**, 10.1103/physrevb.108.195105 (*PRB* **108** (2023)).

[45] M. Suzuki, "Generalized Trotter's Formula and Systematic Approximants of Exponential Operators and Inner Derivations with Applications to Many Body Problems", Commun. Math. Phys. **51**, 183–190 (*CommunMathPhys* **51** (1976)).

[46] M. Suzuki, "Decomposition formulas of exponential operators and Lie exponentials with some applications to quantum mechanics and statistical physics", Journal of Mathematical Physics **26**, 601–612 (*JMatPhys* **26** (1985)).

[47] M. Suzuki, "Fractal decomposition of exponential operators with applications to many-body theories and monte carlo simulations", Physics Letters A **146**, 319–323 (*PhysLetA* **146** (1990)).

[48] M. Suzuki, "General Decomposition Theory of Ordered Exponentials", Proceedings of the Japan Academy, Series B **69**, 161–166 (*ProcJapAcB* **69** (1993)).

Bibliography X

[49]M. S. J. Tepaske, D. Hahn, D. J. Luitz, "Optimal compression of quantum many-body time evolution operators into brickwall circuits", SciPost Physics **14**, 10.21468/scipostphys.14.4.073 (*SciPostPhys* **14** (2023)).

[50]H. F. Trotter, "On the Product of Semi-Groups of Operators", Proceedings of the American Mathematical Society **10**, 545–551 (*ProcAMS* **4** (1959)).

[51]T. Hayata, Y. Kikuchi, "Continuous-time evolution via probabilistic angle interpolation and its applications", (*quant-ph*/**2604**.02854).

[52]M. Maležič, JO, "Efficient Trotter-Suzuki Schemes for Long-time Quantum Dynamics", 10.48550/arXiv.2601.18756 (*quant-ph*/**2601**.18756), Staff Pick by Wolfram Community.

[53]M. Maležič, JO, "Reducing the Gate Count with Efficient Trotter-Suzuki Schemes", PoS, 10.22323/1.518.0131 (*PoS* **LATTICE** (2025)).

[54]M. Maležič, "qiskit-omelyan", version v0.2.1, GitHub repository, https://github.com/MarkoMalezic/qiskit-omelyan (*Qiskit* (2026)), part of the Qiskit Ecosystem.

Bibliography XI

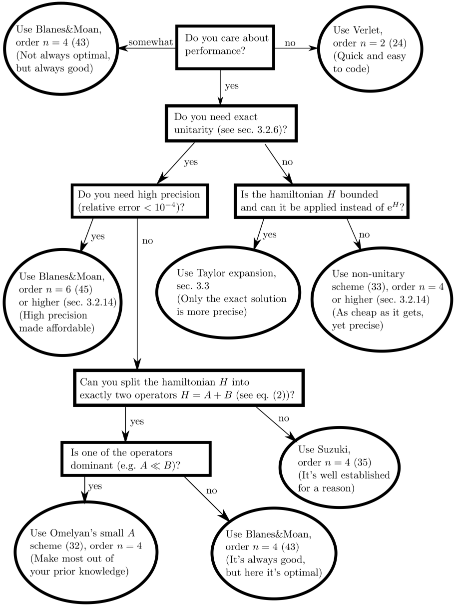
[55] L. Verlet, “Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules”, Phys. Rev. **159**, 98–103 (*PhysRev* **159** (1967)).

[56] X. Wang, Sun, S. Zhou, Z. Wang, P. Zeng, Q. Zhao, T. Gur, T. Li, “Trotter error compensation with polylogarithmic precision and nested-commutator scaling without ancillas”, (*quant-ph*/**2607**.11856).

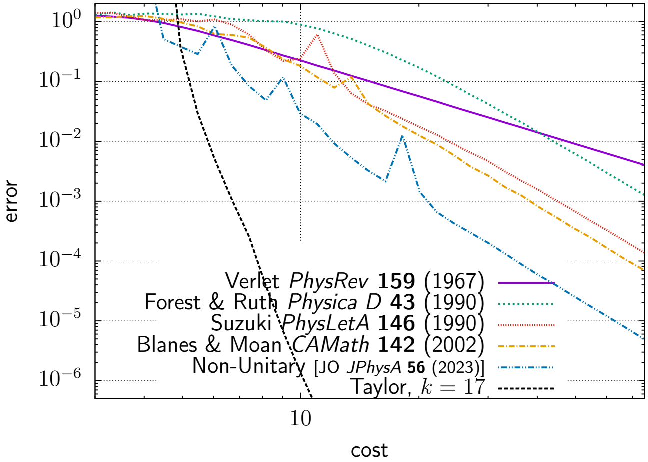
[57] H. Yoshida, “Construction of higher order symplectic integrators”, Physics Letters A **150**, 262–268 (*PhysLetA* **150** (1990)).

[58] H. Zhao, M. Bukov, M. Heyl, R. Moessner, “Making Trotterization Adaptive and Energy-Self-Correcting for NISQ Devices and Beyond”, PRX Quantum **4**, 030319 (*PRXQ* **4** (2023)).

How to choose... [JO *JPhysA* **56** (2023)]



Three simple tricks for better Trotterization



0. Don't use Forest & Ruth.
1. Use 2-operator methods for Λ operators.
2. Use complex coefficients.
3. Consider alternatives.

Heisenberg model and error estimation

[Childs + *PRX* **11** (2021); Heyl, Hauke + *SciAdv* **5** (2019); Schubert & Mendl *PRB* **108** (2023)]

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

$$\begin{aligned} \mathrm{e}^{\mathrm{i}\,Ht} &\equiv U(t) = U(h)^{t/h} \\ &= S(h)^{t/h} + \mathcal{O}\left(h^n\right) , \end{aligned}$$

$$S(h) = \mathrm{e}^{\mathrm{i}\,H_1^x c_1 h} \mathrm{e}^{\mathrm{i}\,H_1^y c_1 h} \mathrm{e}^{\mathrm{i}\,H_1^z c_1 h} \mathrm{e}^{\mathrm{i}\,H_2^x c_1 h} \mathrm{e}^{\mathrm{i}\,H_2^y c_1 h} \mathrm{e}^{\mathrm{i}\,H_2^z c_1 h} \cdots \mathrm{e}^{\mathrm{i}\,H_1^z d_q h} \mathrm{e}^{\mathrm{i}\,H_1^y d_q h} \mathrm{e}^{\mathrm{i}\,H_1^x d_q h} ,$$

$$\begin{aligned} \text{error} &= \frac{1}{\sqrt{N}} \left\| U(t) - S(h)^{t/h} \right\|_{\mathsf{F}} \\ &= \frac{1}{\sqrt{N}} \sqrt{\sum_v \left| U(t) \cdot v - S(h)^{t/h} \cdot v \right|^2} \end{aligned}$$

Proof of validity of Λ-operator decomposition I

For every $i \in \{1, \ldots, q\}$ (setting $d_0 = 0$)

$$\begin{aligned} c_i + d_{i-1} &= a_i - d_{i-1} + d_{i-1} \\ &= a_i \,, \\ d_i + c_i &= b_i - c_i + c_i \\ &= b_i \,, \end{aligned}$$

i.e. decomposition works trivially for $\Lambda = 2$.

Proof of validity of Λ-operator decomposition II

Use the Baker–Campbell–Hausdorff formula:

$$\left(\prod_{k=1}^{\Lambda} \mathrm{e}^{A_k c_i h}\right) = \mathrm{e}^{C_1 c_i h + C_2 (c_i h)^2 + C_3 (c_i h)^3 + \cdots}$$

$$\left(\prod_{k=\Lambda}^{1} \mathrm{e}^{A_k d_i h}\right) = \mathrm{e}^{D_1 d_i h + D_2 (d_i h)^2 + D_3 (d_i h)^3 + \cdots}$$

where $C_1 = D_1 = \sum_{k=1}^{\Lambda} A_k$ and the remaining operators C_l, D_l are some linear combinations of lth order commutators $[A_{k_1}, [A_{k_2}, \cdots [A_{k_{l-1}}, A_{k_l}] \cdots]]$.

Proof of validity of Λ-operator decomposition III

C_l, D_l are independent of i and for every $\Lambda \geq 2$ all the C_l (D_l) are mutually non-commuting and linearly independent in general. Thus, the right hand side is completely independent of Λ.
For $\Lambda = 2$:

$$\mathrm{e}^{h \sum\limits_{k=1}^{\Lambda} A_k + \mathcal{O}\left(h^{n+1}\right)} = \mathrm{e}^{C_1 c_1 h + C_2 (c_1 h)^2 + \cdots} \mathrm{e}^{D_1 d_1 h + D_2 (d_1 h)^2 + \cdots} \\ \dots \mathrm{e}^{C_1 c_q h + C_2 (c_q h)^2 + \cdots} \mathrm{e}^{D_1 d_q h + D_2 (d_q h)^2 + \cdots}$$

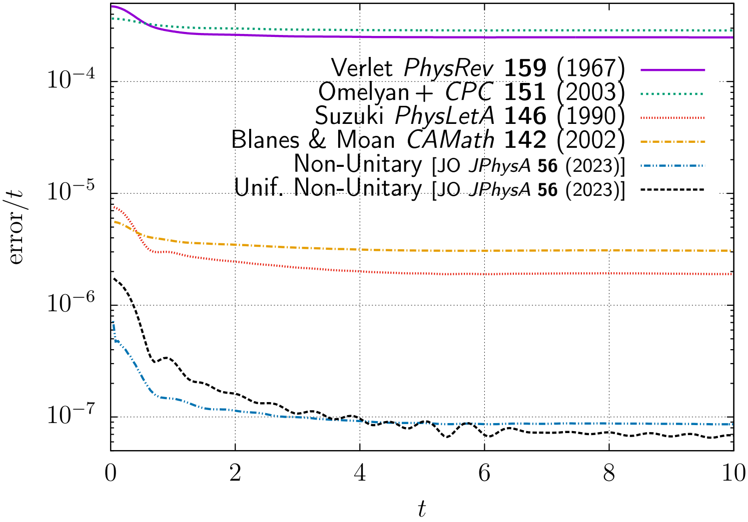
Since the c_i, d_i fulfil this condition for every C_i, D_i (they solve the corresponding system of polynomial equations), they automatically fulfil it for every Λ.

Derivation time-dependent Hamiltonian evolution

Effective Hamiltonian $H_{\mathsf{eff}}(t) \equiv \sum_k A_k(t) + \mathcal{T}$, Trotterize:

$$
\begin{aligned}
S_n(t,h) = {} & \mathrm{e}^{c_1 \mathcal{T} h} \left(\prod_{k=1}^{\Lambda} \mathrm{e}^{c_1 A_k(t) h} \right) \left(\prod_{k=\Lambda}^{1} \mathrm{e}^{d_1 A_k(t) h} \right) \mathrm{e}^{d_1 \mathcal{T} h} \cdots \\
& \left(\prod_{k=\Lambda}^{1} \mathrm{e}^{d_{q-1} A_k(t) h} \right) \mathrm{e}^{d_{q-1} \mathcal{T} h} \, \mathrm{e}^{c_q \mathcal{T} h} \left(\prod_{k=1}^{\Lambda} \mathrm{e}^{c_q A_k(t) h} \right) \left(\prod_{k=\Lambda}^{1} \mathrm{e}^{d_q A_k(t) h} \right) \mathrm{e}^{d_q \mathcal{T} h} \\
= {} & \mathrm{e}^{\mathcal{T} h} \left(\prod_{k=1}^{\Lambda} \mathrm{e}^{c_1 A_k(t+(1-c_1)h) h} \right) \left(\prod_{k=\Lambda}^{1} \mathrm{e}^{d_1 A_k(t+(1-c_1)h) h} \right) \cdots \\
& \left(\prod_{k=\Lambda}^{1} \mathrm{e}^{d_{q-1} A_k(t+(d_q+c_q+d_{q-1})h) h} \right) \left(\prod_{k=1}^{\Lambda} \mathrm{e}^{c_q A_k(t+d_q h) h} \right) \left(\prod_{k=\Lambda}^{1} \mathrm{e}^{d_q A_k(t+d_q h) h} \right)
\end{aligned}
$$

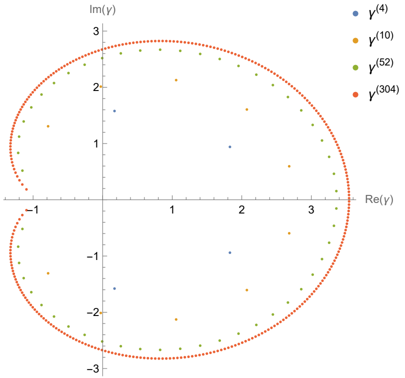
Error accumulation II

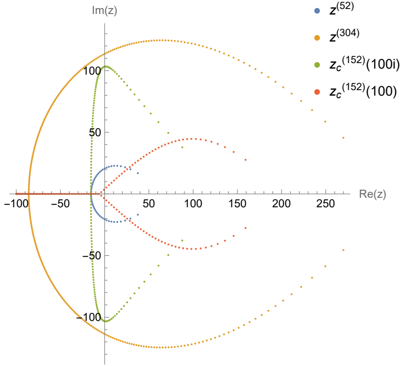


$$c_1 = 0.1 + 0.025\,\mathrm{i}$$
$$d_1 = 0.1 + 0.025\,\mathrm{i}$$
$$c_2 = 0.1 - 0.066\,\mathrm{i}$$
$$d_2 = 0.1 - 0.066\,\mathrm{i}$$
$$c_3 = 0.1 + 0.082\,\mathrm{i}$$
$$\text{order} = 4\,,$$
$$\text{cycles} = 5\,,$$
$$\mathrm{Eff}_4 = 6.38$$

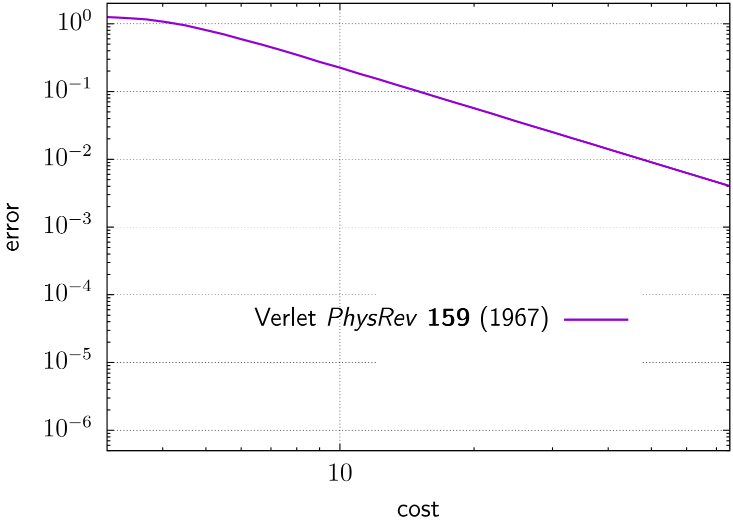
Convergence regions [JO *PoS* **LATTICE** (2023)]

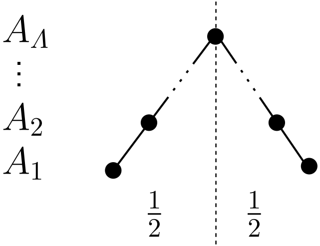
$$\mathrm{e}^{Hh} = \prod_{i=1}^{k} \left(1 + \frac{\gamma_i^{(k)}}{k} Hh \right) + \mathcal{O}\left(h^{k+1}\right) \qquad \propto \prod_{i=1}^{k} \left(Hh - z_i^{(k)} \right) + \mathcal{O}\left(h^{k+1}\right)$$





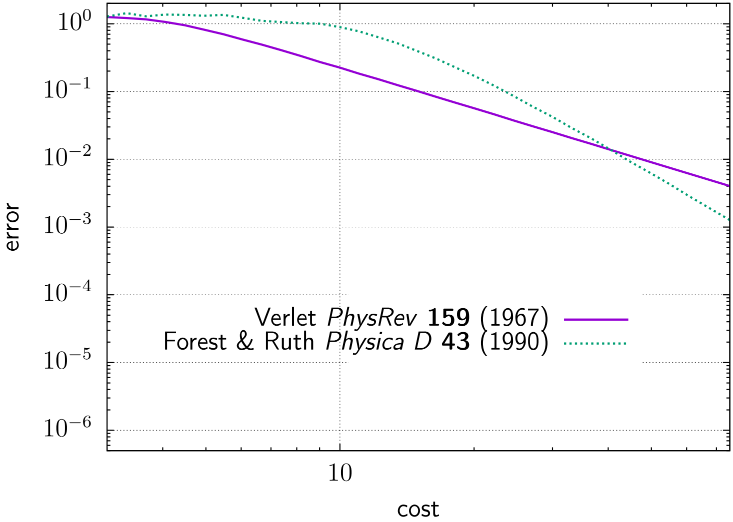
Benchmarking the Heisenberg model

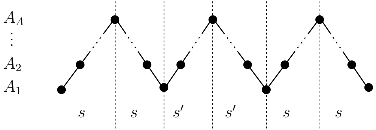




$$\text{order} = 2\,,$$
$$\text{cycles} = 1$$

Benchmarking the Heisenberg model





$$s = \frac{1}{2(2-\sqrt[3]{2})}$$

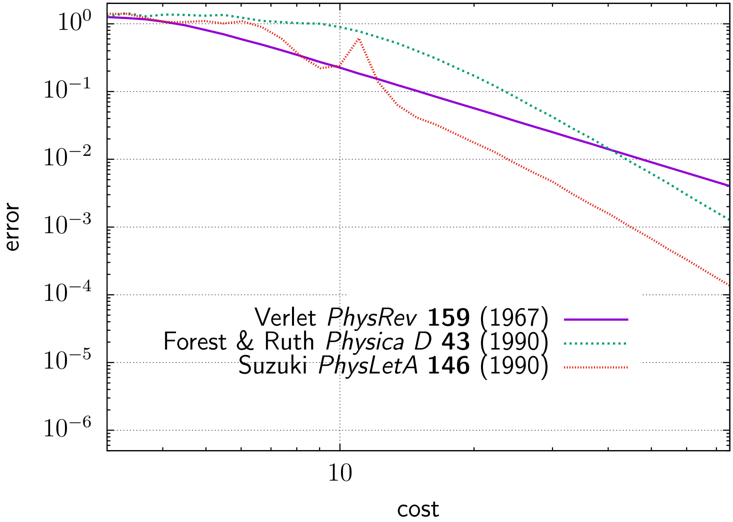
$$s' = \frac{1}{2} - 2s$$

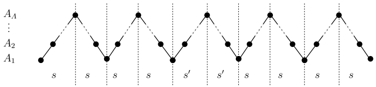
$$\text{order} = 4\,,$$

$$\text{cycles} = 3\,,$$

$$\text{Eff}_4 = 0.315$$

Benchmarking the Heisenberg model





$$s = \frac{1}{2(4-\sqrt[3]{4})}$$

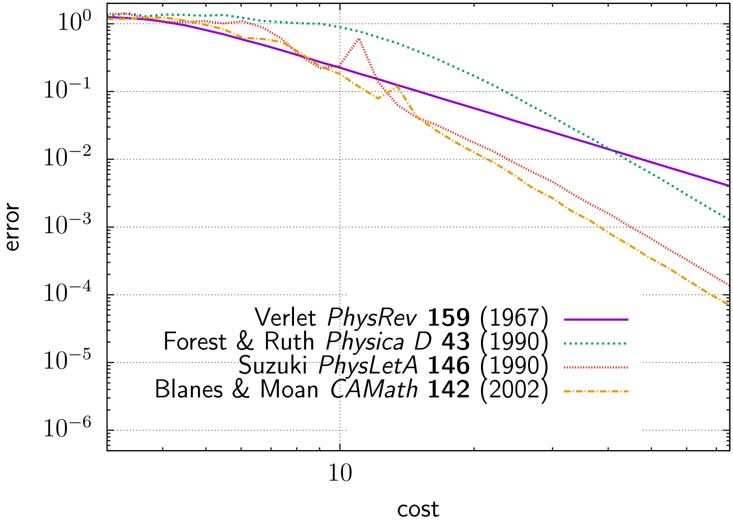
$$s' = \frac{1}{2} - 4s$$

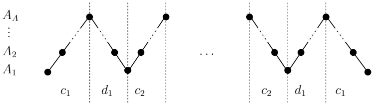
$$\text{order} = 4\,,$$

$$\text{cycles} = 5\,,$$

$$\mathrm{Eff}_4 = 1.10$$

Benchmarking the Heisenberg model





$c_1 \approx 0.08 \quad d_1 \approx 0.13$

$c_2 \approx 0.22 \quad d_2 \approx -0.36$

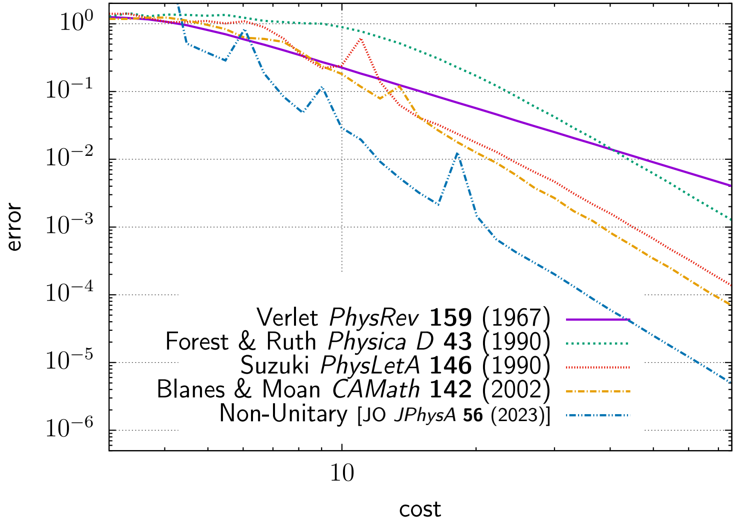
$c_3 \approx 0.32 \quad d_3 \approx 0.11$

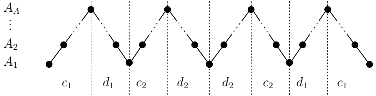
$$\text{order} = 4\,,$$

$$\text{cycles} = 6\,,$$

$$\mathrm{Eff}_4 = 10.2$$

Benchmarking the Heisenberg model





$$c_1 \approx 0.10 + 0.02\,\mathrm{i}$$

$$d_1 \approx 0.15 + 0.07\,\mathrm{i}$$

$$c_2 \approx 0.14 + 0.07\,\mathrm{i}$$

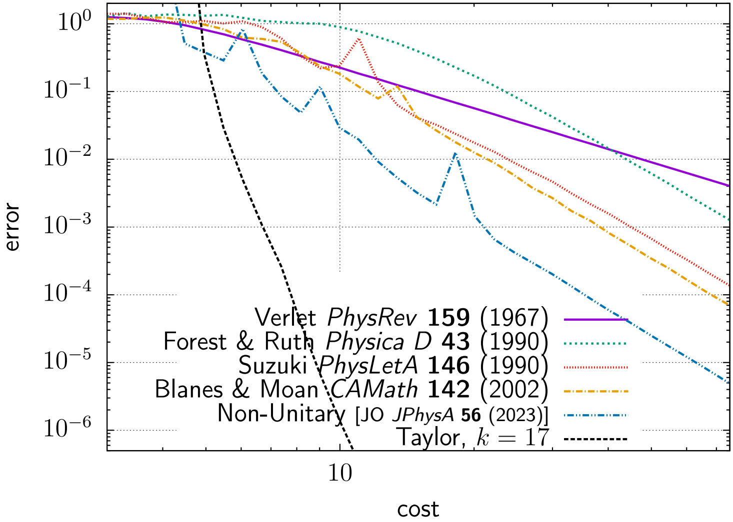
$$d_2 \approx 0.11 - 0.16\,\mathrm{i}$$

$$\text{order} = 4\,,$$

$$\text{cycles} = 4\,,$$

$$\mathrm{Eff}_4 = 29.9$$

Benchmarking the Heisenberg model



$$e^{Hh} = \lim_{k\to\infty} \sum_{i=0}^{k} \frac{(Hh)^i}{i!}$$

$$\left| \frac{(\lambda_{\mathsf{max}}(H)h)^k}{(k+1)!} \right| < \varepsilon$$

Adaptive step size [Blanes + *AplNumMat* **146** (2019); Zhao + *PRXQ* **4** (2023)]