Simple Ways to improve Discrete Time Evolution

Johann Ostmeyer[a,*]

[a]*Department of Mathematical Sciences, University of Liverpool, United Kingdom*

E-mail: J.Ostmeyer@liverpool.ac.uk

Suzuki-Trotter decompositions of exponential operators like $\exp(Ht)$ are required in almost every branch of numerical physics. Often the exponent under consideration has to be split into more than two operators, for instance as local gates on quantum computers. In this work, we demonstrate how highly optimised schemes originally derived for exactly two operators can be applied to such generic Suzuki-Trotter decompositions. After this first trick, we explain what makes an efficient decomposition and how to choose from the large variety available. Furthermore we demonstrate that many problems for which a Suzuki-Trotter decomposition might appear to be the canonical ansatz, are better approached with different methods like Taylor or Chebyshev expansions. In particular, we derive an efficient and numerically stable method to implement truncated polynomial expansions based on a linear factorisation using their complex zeros.

---

*Speaker

1. Introduction

(Suzuki-)Trotter decomposition schemes, or **Trotterizations**, also referred to as **splitting methods**, are approximations to operator exponentials of the form

$$e^{(A+B)h+O(h^{n+1})} = e^{Aa_1h}e^{Bb_1h}e^{Aa_2h}\cdots e^{Bb_qh}e^{Aa_{q+1}h},\tag{1}$$

where A, B are some non-commuting operators, h is a (small) time step and the a_i, b_i denote some coefficients guaranteeing correctness to order n.

Deriving precise Trotter decompositions is not only a challenging mathematical problem, their applications are practically ubiquitous. Just to give a few examples from numerical physics, Trotterization is needed from condensed matter to high energy physics, allowing real and imaginary time evolution [1], it is a corner stone in hybrid Monte Carlo [2], tensor networks [3] as well as quantum computing [4].

We remark that sometimes "symplectic integrator" is used interchangeably with Trotterization, but this can be very misleading. For more details on the distinctive properties of symplectic integrators see Section 3.

This proceeding expands on the results in our previous work [5] where all the technical background can be found that has been omitted here. In the next Section 2 we will reiterate the highly useful method to construct decompositions featuring arbitrarily many operators starting from a two-operator decomposition as in equation (1). Closely following Ref. [6], we will introduce a notion of efficiency for Trotterizations in Section 3. Section 4 contains the major novelty of this work, namely an efficient way to implement truncated polynomial (non-Trotter-type) approximations of the matrix exponential that is numerically stable. The approach is based on a factorisation of the polynomial using its zeros in the complex plane. The various methods will be benchmarked in Section 5 and a summary of the "simple ways to improve discrete time evolution" will be provided in the concluding Section 6.

2. Adapting 2-stage decompositions to an arbitrary number of stages

Decompositions with two stages, i.e. with a Hamiltonian $H = A + B$ consisting of exactly 2 operators as in equation (1), have been studied and optimised extensively in literature (e.g. [7, 8] and many more). However, this is only a special case. Often a Hamiltonian consists of three or more non-commuting operators (an obvious example is the Heisenberg model) and in many cases it is necessary to split it into $O(L)$ local contributions, e.g. to apply gates to tensor network or quantum simulations. In these cases only less optimised schemes like those by Hatano and Suzuki [9] or Yoshida [10] have been applicable so far. Now recently, we have shown that *every scheme of order n applicable to two stages defines an order n scheme for an arbitrary number of stages* [5]. This allows to adapt the valuable work done on 2-stage decompositions so far and it furthermore provides a simple means to derive efficient new methods in the future since it is sufficient to analyse the method's properties in the simple case of two stages. The idea behind the transformation required for this adaptation has been visualised in figure 1.

Figure 1: Visualisation of the transformation allowing to rewrite a 2-stage decomposition for arbitrarily many stages. In this depiction every dot corresponds to a single operator evaluation or *stage*, a straight line corresponds to a *ramp* and two adjacent ramps form a *cycle*. We start in the top row with a 2-stage decomposition as in eq. (1) and transform the coefficients a_i, b_i into the equivalent c_i, d_i as in eq. (2) using eqs. (3) through (5). Every ramp can now be populated with more stages in the last row without changing the validity of the decomposition.

More rigorously, the theorem proved in [5] reads as follows. Let the coefficients a_i and b_i as in equation (1) define a 2-stage decomposition of order n. Then the decomposition

$$
e^{h \sum_{k=1}^{\Lambda} A_k + O(h^{n+1})} = \left(\prod_{k=1}^{\Lambda} e^{A_k c_1 h} \right) \left(\prod_{k=\Lambda}^{1} e^{A_k d_1 h} \right) \cdots \left(\prod_{k=1}^{\Lambda} e^{A_k c_q h} \right) \left(\prod_{k=\Lambda}^{1} e^{A_k d_q h} \right) \tag{2}
$$

of Λ non-commuting operators A_k into ordered products $\prod_{k=1}^{\Lambda} X_k \equiv X_1 X_2 \cdots X_\Lambda$ is exact to order n as well and the coefficients c_i, d_i are given by

$$
c_1 = a_1, \qquad\qquad\qquad d_1 = b_1 - c_1, \tag{3}
$$

$$
c_2 = a_2 - d_1, \qquad\qquad\qquad d_2 = b_2 - c_2, \tag{4}
$$

$$
\vdots \qquad\qquad\qquad\qquad \vdots
$$

$$
c_q = a_q - d_{q-1}, \qquad\qquad\qquad d_q = b_q - c_q. \tag{5}
$$

3. Analysing the efficiency of decomposition schemes

For the rest of this work we will consider symmetric decomposition schemes only. Besides their often useful reversibility, symmetric schemes are simply more efficient for evolutions over long times t, i.e. when $t/h \gg 1$ which is the physically interesting and computationally challenging case. The reason is that any non-symmetric and therefore odd order $n-1$ scheme can be elevated to even order n without any drawbacks by reverting the sequence of coefficients a_i, b_i (or c_i, d_i) in every second step. The established symmetry guarantees an even order n. In case the integer t/h was an even number to start with, the computational effort remains the same. If t/h was odd,

it has to be incremented by one at the relative additional computational cost of $h/t \ll 1$. In both cases the induced symmetric even order n decomposition is strictly more efficient than the initial non-symmetric one.

Omelyan, Mryglod and Folk [7] as well as Blanes and Moan [8] derived a number of highly optimised schemes. Closely following the approach by Omelyan et al. we will now define what requirements exactly a decomposition scheme has to fulfil to be of order n and what it means for such a scheme to be efficient.

Following the derivations in Refs. [6, 7], we can write down a basis for the leading order errors of a decomposition

$$e^{(A+B)h+O_1 h+O_3 h^3+O_5 h^5+\cdots} = e^{Aa_1 h}e^{Bb_1 h}\cdots e^{Bb_q h}e^{Aa_{q+1}h} \tag{6}$$

in terms of commutators

$$O_1 = (\nu - 1)A + (\sigma - 1)B\,, \tag{7}$$

$$O_3 = \alpha[A,[A,B]] + \beta[B,[A,B]]\,, \tag{8}$$

$$O_5 = \gamma_1[A,[A,[A,[A,B]]]] + \gamma_2[A,[A,[B,[A,B]]]] + \gamma_3[B,[A,[A,[A,B]]]]$$
$$+ \gamma_4[B,[B,[B,[A,B]]]] + \gamma_5[B,[B,[A,[A,B]]]] + \gamma_6[A,[B,[B,[A,B]]]]\,. \tag{9}$$

A decomposition is valid if $\nu = \sigma = 1$, that is $O_1 = 0$. This can easily be guaranteed by the intuitive condition $\sum_i a_i = \sum_i b_i = 1$ (or equivalently $\sum_i(c_i + d_i) = 1$). In order to construct a 4th order decomposition, $\alpha = \beta = 0$ has to be satisfied additionally. For a 6th order scheme all the γ_j have to vanish, and so on.

We remark that symplectic integrators (or Runge–Kutta–Nyström methods) deal with the special case of $[B,[B,[B,A]]] = 0$ leading to significantly less contributions to all the operators O_k with $k \geq 5$ than in the general case discussed here. This allows for more efficient but specialised methods, in particular higher order symplectic decompositions $n \geq 6$ require less cycles q than general Trotterizations. This distinction has been discussed in detail by Blanes and Moan [8]. A comprehensive list of efficient symplectic integrators up to order $n \leq 6$ has been provided by Omelyan [6], we especially recommend Table 2 therein.

A decomposition is efficient if its leading order errors are small compared to the number q of cycles[1] it requires. We assume all the commutator products in the equations (7-9) to be mutually orthogonal so that the euclidean norm over the spanned vector space estimates the total errors. Note that orthogonality is not guaranteed, but for high enough dimensional operators the overlap will be negligible in general. In particular we define the efficiencies following Omelyan [6]

$$\text{Eff}_2 = \frac{1}{q^2\sqrt{|\alpha|^2 + |\beta|^2}}\,, \qquad\qquad \text{Eff}_4 = \frac{1}{q^4\sqrt{\sum_{j=1}^6 |\gamma_j|^2}}\,. \tag{10}$$

Higher order efficiencies Eff_n are defined analogously via the leading order error and the respective number of cycles. Within an order the efficiency is directly proportional to the precision obtained at any given computational effort and a higher efficiency is generally better. More details on comparisons between orders and a detailed list of Suzuki-Trotter decompositions can be found in Ref. [5].

---

[1]q is also often called 'stages' or 'number of force calculations' in the literature.

4. Truncated polynomial expansion

We now present a totally different, not strictly unitary, approach based on the truncated Taylor expansion

$$\mathrm{e}^{Hh} = \lim_{k\to\infty} \sum_{i=0}^{k} \frac{(Hh)^i}{i!}, \tag{11}$$

or Chebyshev expansion

$$\mathrm{e}^{Hh} = \lim_{k\to\infty} \sum_{i=0}^{k} \mu_i T_i\left(\frac{H}{\Gamma}\right), \quad \mu_0 = I_0(\Gamma h), \ \mu_i = 2I_i(\Gamma h), \tag{12}$$

where the T_i denote the Chebyshev polynomials, I_i the modified Bessel functions of first kind and $\Gamma > 0$ large enough (as defined below). Note that the coefficients μ_i in equation (12) are different for real- and imaginary time evolution (use $I_i(\mathrm{i}\,\Gamma h) = \mathrm{i}^{-i} J_i(\Gamma h)$ with the (unmodified) Bessel functions J_i in the former case).

The cutoff k has to be chosen so that all eigenvalues λ of H are sufficiently suppressed

$$\left|\frac{(\lambda_{\max}(H)h)^k}{(k+1)!}\right| < \varepsilon \qquad \text{(Taylor)}, \tag{13}$$

$$|\mu_{k+1}| < \varepsilon \qquad \text{(Chebyshev)}, \tag{14}$$

where ε is the desired relative precision. Technically a cutoff k large enough guarantees super-exponential convergence independently of the step size h for both, Taylor and Chabyshev expansions. In practice, however, long time evolutions have to be subdivided into factors of limited step size h. In particular, the Taylor addends (11) can span many orders of magnitude so that finite precision arithmetics leads to significant errors. In Ref. [5] we presented more details on the derivation of the optimal step size h and cutoff k assuming the direct summation of the terms in the Taylor series (11).

Here, we put forward a fundamentally different method for the evaluation of the truncated series. Since we are always dealing with finite order k polynomials, they have k zeros $z_i^{(k)}$ in the complex plane and the exponential can be approximated as

$$\mathrm{e}^{Hh} = \prod_{i=1}^{k}\left(1 + \frac{\gamma_i^{(k)}}{k} Hh\right) + O\left(h^{k+1}\right) \tag{15}$$

$$\propto \prod_{i=1}^{k}\left(Hh - z_i^{(k)}\right) + O\left(h^{k+1}\right), \tag{16}$$

$$\gamma_i^{(k)} \equiv -\frac{k}{z_i^{(k)}}. \tag{17}$$

In particular the zeros of the truncated Taylor series are very well behaved falling on the Szegő curve [11]

$$\lim_{k\to\infty} \frac{1}{k} z^{(k)} = \left\{ z : |z\mathrm{e}^{1-z}| = 1, |z| \le 1 \right\}. \tag{18}$$
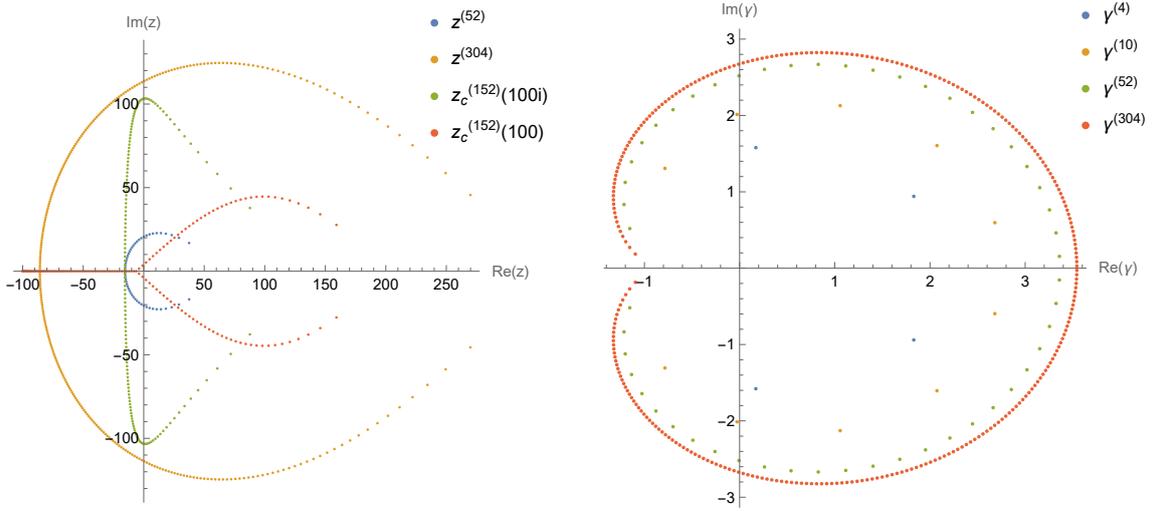
Figure 2: Distribution of Taylor and Chebyshev expansion coefficients for the exponential function in the complex plane. A factorisation (15) is exact up to double machine precision within $|z| \leq 10$ when using $z^{(52)}$, within $|z| \leq 100$ using $z^{(304)}$, in the interval $z \in [-100\,\mathrm{i}, 100\,\mathrm{i}]$ using $z_c^{(152)}(100\,\mathrm{i})$, and in $z \in [-100, 100]$ using $z_c^{(152)}(100)$. Left: Zeros $z^{(k)}$ of the truncated Taylor series (11) and zeros $z_c^{(k)}(\Gamma h)$ of the Chebyshev polynomial (12) best describing the interval $[-\Gamma h, \Gamma h]$ truncated at k-th order, respectively. Right: Coefficients $\gamma^{(k)} = -k/z^{(k)}$ in the factorisation of the Taylor series.

Both, the zeros (left) and their corresponding coefficients (right) have been visualised in figure 2. The distribution of the zeros intuitively forms a boundary for the region in which the exponential function is well approximated by the respective truncation since $\exp(z)$ itself is never zero and a zero of the polynomial marks a breakdown of the approximation. Clearly, the Taylor series provides a universal approximation in the complex plane within a given radius (asymptotically k/e) while the Chebyshev polynomials only yield a good approximation on either the real or the imaginary axis. However, on this one axis the Chebyshev polynomials can approximate an interval that is asymptotically as large as $[-k, k]$ which makes them a factor $e \approx 2.7$ more efficient than the Taylor expansion in the limit of large k.

In practice, an implementation using the factorisation (15) is always preferable to a direct summation (11) because it does not suffer the aforementioned numerical instability. Less obviously, the factorised version is also more memory efficient than either Taylor or Chebyshev summation. Furthermore it has a reduced runtime compared to the stable iterative construction

$$T_0(H) = \mathbb{1}, \tag{19}$$

$$T_1(H) = H, \tag{20}$$

$$T_{i+1}(H) = 2H \cdot T_i(H) - T_{i-1}(H) \tag{21}$$

of Chebyshev polynomials. For high accuracy, however, it if of crucial importance to choose a good order for the coefficients $\gamma_i^{(k)}$, even though theoretically all the factors in the product (15) commute. We find empirically that best results are achieved when coefficients are always grouped together in tuples, so that the sum over the coefficients within each tuple is roughly the same. In this manner,

the $\gamma_i^{(k)}$ can be thought of as individual sub-steps that should be arranged in a way maintaining a roughly constant progress.

Since the coefficients of the factorised polynomials are real, all the zeros $z_i^{(k)}$ and coefficients $\gamma_i^{(k)}$ come in complex conjugate pairs. These pairs play a special role in the aforementioned ordering of coefficients. They should never be separated. In fact, numerical accuracy is increased when these pairs are jointly evaluated in a quadratic term with real coefficients

$$\left(1 + \frac{\gamma_i^{(k)}}{k}Hh\right)\left(1 + \frac{\bar{\gamma}_i^{(k)}}{k}Hh\right) = 1 + 2\frac{\mathrm{Re}\,\gamma_i^{(k)}}{k}Hh + \frac{\left|\gamma_i^{(k)}\right|^2}{k^2}(Hh)^2 \ . \tag{22}$$

In most classical implementations this quadratic version will be favourable over the fully factorised one, especially if the complex numbers can be completely avoided this way. The only drawback is the increased memory footprint that can become relevant in some cases, especially if the squared hamiltonian H^2 has to be constructed explicitly.

Note that the number k is not equivalent with the number of cycles. This relation strongly depends on the implementation. For instance in the case we are discussing below (see Section 5) the implementation with cutoff k has approximately the runtime of $q \approx k/6$ cycles.

5. Numerical experiments

We use the Heisenberg XXZ-model [12] as a test for the decomposition schemes. Details on the model and the error estimation can be found in [5]. In short, we compute the exact time evolution operator $\mathrm{e}^{-\mathrm{i}Ht}$ using exact diagonalisation and calculate its difference to the respective approximation in Frobenius norm. The results are visualised in figure 3. All the errors have been obtained after the same evolution time of $t = 100$ and the axis labelled q/h is equivalent to the de-facto computational costs. For the Trotter-type schemes the minimal number of 3 stages has been used.

The fourth order schemes benchmarked here (fig. 3, left) have been catalogued in [5] including their theoretical efficiencies (10). In accordance with said efficiencies, they form a clear hierarchy that is mirrored by our numerical experiments. The decomposition by Forest and Ruth [13] is clearly worst by a margin. We quote it here because it is still popular in the literature for its simplicity, but we emphasise that it should never be used. Next comes Suzuki's standard scheme [9] which can be considered a baseline to which every other decomposition has to be compared. Applying our transformation from equations (3) through (5) to the highly optimised scheme by Blanes and Moan [8], allows to significantly outperform Suzuki's decomposition. This makes Blanes and Moan's scheme the most efficient strictly unitary fourth order decomposition to date. Our non-unitary method [5] employing complex-valued coefficients as well as a fourth order Taylor expansion both easily outperform the unitary methods. It is interesting to observe that Taylor's method gives relatively good results even for extremely low computations cost, even though it is asymptotically inferior to the non-unitary Trotter-type decomposition.

We remark that this reasonably good performance at low cost is a characteristic feature of low order Taylor series setting them apart from Chebyshev expansions. In this regime a Chebyshev expansion is uniformly mediocre, while the Taylor series still describes most of the eigenvalues of
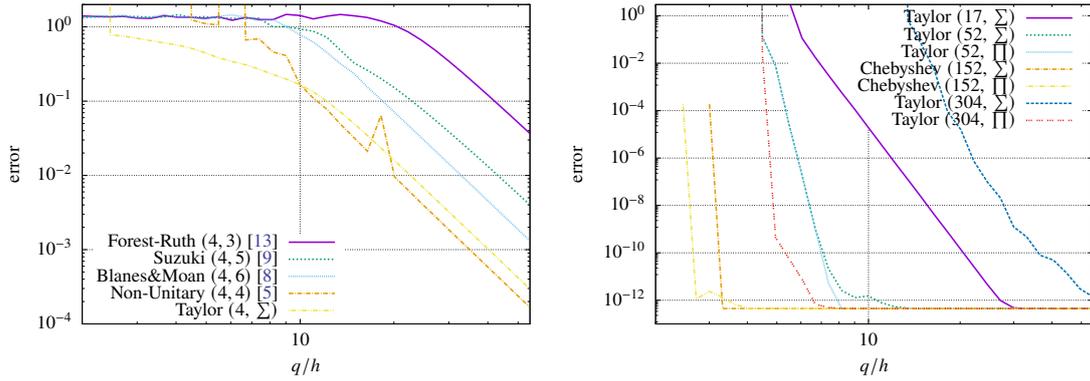
Figure 3: Collection of order $n = 4$ schemes (left) and truncated polynomial expansions (right). Error, i.e. the difference between the exact time evolution operator and the respective decomposition in Frobenius norm, as a function of computational cost, i.e. number of cycles q (assuming $q = k/6$ for non-Trotter type methods) divided by the time step h. Following the name of the scheme we write the order / cycles tuple (n, q) or the cutoff / method tuple $(k, \cdot)$, respectively. Here the method can be either '$\sum$' as in eqs. (11) and (12) or '$\prod$' as in eq. (15)

H very well because they tend to be clustered around zero. This makes Taylor the ideal choice when the target precision is not too high, Chebyshev on the other hand is preferable in the regime of high accuracy and high orders.

The same conclusion is supported in the right panel of figure 3 where results from different truncated polynomials are summarised. As expected, we find that the efficiency of the Taylor series increases with higher orders / cutoffs k and that they in turn yield to Chebyshev polynomials in terms of performance. It is absolutely essential for this trend that the factorised version '$\prod$' (eq. (15)) is used rather than the sum '$\sum$' (eqs. (11), (12)). While in the case of Chebyshev the factorisation merely reduces computational costs, for a Taylor series it can mean the difference between a correct result and utter nonsense. In double precision, the difference is bound to occur for every cutoff $k > 17$, it is clearly visible for $k = 52$, and it becomes a dominant feature for $k = 304$.

On a side note, the identical error plateau reached by all Taylor and Chebyshev methods strongly indicates that the limiting factor in this case is no longer the respective quoted method but the exact diagonalisation which is also performed to finite precision.

6. Conclusion

In this work we have presented a variety of methods for discrete time evolution. The focus lies on representative fourth order Trotterizations and truncated polynomial expansions. It is not the aim of this work to provide a comprehensive overview over different Trotter decomposition and the interested reader is referred to our previous work [5] for this purpose as well as for a flow chart providing a simple guideline to choose a close to optimal method in a given scenario.

The main novelty setting this work apart from [5] is the introduction and discussion of factorised formulae (15) for the efficient and numerically stable implementation of truncated polynomial series in Section 4. We showed how, based on the complex zeros of the polynomial, a product featuring

only coefficients of order one can be constructed and demonstrated that this construction is always preferable to the naive summation of the different order contributions.

In summary, efficient discrete time evolution can be boosted rather easily if the following points are considered prior to deciding on an implementation: Use a Chebyshev or Taylor expansion (in this order) whenever possible and implement it in the factorised form (15). Do not be afraid to use complex-valued coefficients in Trotter decompositions as they can significantly improve precision at the price of marginally violating unitarity. Adapt highly optimised 2-operator decompositions like that by Blanes and Moan [8] or our non-unitary schemes for any required number of operators using the transformation detailed in equations (3) through (5) and visualised in figure 1.

Finally, we remark that the theoretical error prediction for a given time evolution method is still anything but accurate and subject to ongoing research. So long, the only way to guarantee that the optimal solution to a given problem has been chosen, remains numerical testing of that particular problem.

Acknowledgements

The entire code (implemented in C with a front-end in R) and data required to reproduce the results and plots of this paper have been published under open access and can be found in [14]. The coefficients $\gamma^{(k)}$ have been computed in Mathematica and a notebook allowing to calculate these coefficients to arbitrary precision is included in the code.

References

[1] S.-H. Lin, R. Dilip, A.G. Green, A. Smith and F. Pollmann, *Real- and Imaginary-Time Evolution with Compressed Quantum Circuits*, *PRX Quantum* **2** (2021) 010342.

[2] S. Duane, A.D. Kennedy, B.J. Pendleton and D. Roweth, *Hybrid Monte Carlo*, *Phys. Lett.* **B195** (1987) 216.

[3] M. Schneider, J. Ostmeyer, K. Jansen, T. Luu and C. Urbach, *Simulating both parity sectors of the Hubbard Model with Tensor Networks*, *Phys. Rev. B* **104** (2021) 155118 [2106.13583].

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

[5] J. Ostmeyer, *Optimised Trotter decompositions for classical and quantum computing*, *J. Phys. A* **56** (2023) 285303 [2211.02691].

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

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

[8] S. Blanes and P. Moan, *Practical symplectic partitioned Runge–Kutta and Runge–Kutta–Nyström methods*, *Journal of Computational and Applied Mathematics* **142** (2002) 313.

[9] N. Hatano and M. Suzuki, *Finding Exponential Product Formulas of Higher Orders*, in *Quantum Annealing and Other Optimization Methods*, pp. 37–68, Springer Berlin Heidelberg (2005), DOI.

[10] H. Yoshida, *Construction of higher order symplectic integrators*, *Physics Letters A* **150** (1990) 262.

[11] R.S. Varga and A.J. Carpenter, *Zeros of the partial sums of cos(z) and sin(z). III*, *Applied Numerical Mathematics* **60** (2010) 298.

[12] P. Buividovich and J. Ostmeyer, *Real-time simulations of quantum spin chains: Density of states and reweighting approaches*, *Phys. Rev. B* **107** (2023) 024302.

[13] E. Forest and R.D. Ruth, *Fourth-order symplectic integration*, *Physica D: Nonlinear Phenomena* **43** (1990) 105.

[14] J. Ostmeyer, *j-ostmeyer/efficient-trotter: Optimised Trotter Decompositions*, Sept., 2023. 10.5281/zenodo.8322984.