Weight-sharing Transformer quantum states with Suzuki–Trotter decompositions

Kimihiro Yamazaki
The University of Osaka
k-yamazaki@ist.osaka-u.ac.jp

Itsushi Sakata
RIKEN
itsushi.sakata@riken.jp

Takuya Konishi
The University of Osaka
RIKEN
konishi@ist.osaka-u.ac.jp

Yoshinobu Kawahara
The University of Osaka
RIKEN
kawahara@ist.osaka-u.ac.jp

Abstract

Transformer quantum states (TQS) achieve competitive accuracy on frustrated spin systems, yet their reliance on increasing the parameter count offers limited systematic control of the accuracy–efficiency trade-off in variational Monte Carlo. We propose a weight-sharing TQS that treats a single encoder block as a short imaginary-time propagator for discrete imaginary-time evolution. Within each block, we embed Suzuki–Trotter decompositions to increase the local approximation order, thereby improving accuracy without adding trainable parameters. In this framework, depth sets the total imaginary time and acts as a learned low-energy projector, providing a tunable accuracy control at fixed parameter count. On the square-lattice J_1–J_2 Heisenberg model, it attains accuracy comparable to conventional non-weight-sharing TQS while using fewer parameters.

1 Introduction

Neural-network quantum states (NQS) provide accurate variational ansätze for ground states within variational Monte Carlo (VMC) [1, 2]. They have demonstrated effectiveness across a wide range of quantum lattice models [3–5]. Recent work has introduced Transformer quantum states (TQS) [6, 7], a class of NQS based on Transformer architectures, which achieve state-of-the-art performance on challenging frustrated quantum spin models [5]. Notably, this accuracy is achieved without introducing explicit physical priors, relying instead on the Transformer's expressive power.

Despite strong empirical performance, these TQS treat the Transformer as a black box: accuracy is often sought to be improved by blindly scaling up model size (e.g., depth and hidden dimension) rather than systematic architectural controls. This practice often yields overparameterized models and redundant capacity for target quantum systems [5]. Such overparameterization also raises both sampling and optimization costs in VMC, where higher accuracy also requires more samples.

To balance the efficiency–accuracy trade-off in a transparent manner, we integrate two insights into a TQS architecture: i) *weight sharing* to improve parameter efficiency [8–12], and ii) *a dynamical system view on Transformer encoders* to enhance accuracy [13, 14]. Building on these insights, we implement a weight-sharing TQS in which a single shared encoder block is treated as a short imaginary-time propagator as part of discretized *imaginary-time evolution*. This view offers a clear physical interpretation that the encoder depth sets the total imaginary time while the propagator acts as a learned low-energy projector. Motivated by this view, we embed the *Suzuki–Trotter decomposition schemes* [15–19] within each block, improving accuracy without adding trainable parameters. We

Machine Learning and the Physical Sciences Workshop, NeurIPS 2025.

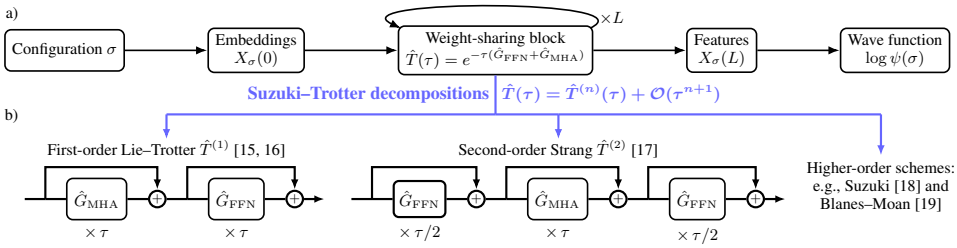

Figure 1: (a) Our weight-sharing TQS architecture. (b) Examples of the weight-sharing block derived from Suzuki–Trotter decompositions.

demonstrate that our proposed method matches conventional non-weight-sharing TQS baselines with fewer parameters on the square-lattice J_1–J_2 Heisenberg model.

2 Preliminary

Frustrated quantum spin model. We take the target Hamiltonian $\hat{H}$ to be the J_1–J_2 Heisenberg model on the square lattice,

$$\hat{H} = J_1 \sum_{\langle i,j\rangle} \hat{\mathbf{S}}_i \cdot \hat{\mathbf{S}}_j + J_2 \sum_{\langle\langle i,j\rangle\rangle} \hat{\mathbf{S}}_i \cdot \hat{\mathbf{S}}_j, \tag{1}$$

where $\hat{\mathbf{S}}_i = (\hat{S}_i^x, \hat{S}_i^y, \hat{S}_i^z)$ are spin-1/2 operators, $\langle i,j\rangle$ denotes nearest neighbors, and $\langle\langle i,j\rangle\rangle$ next-nearest neighbors. Here J_1 and J_2 denote the exchange coupling constants of the nearest-neighbor and next-nearest-neighbor terms, respectively. On the square lattice, the coupling ratio $J_2/J_1 \approx 0.5$ sits in a strongly frustrated regime where the ground-state sign structure remains unsettled, posing a long-standing challenge. Recent TQS models have achieved state-of-the-art variational energies on the standard 10×10 benchmark in this regime [5].

Variational Monte Carlo optimization. We consider a variational wave function $\psi(\sigma)$, where σ denotes a spin configuration. The variational energy E is estimated from samples $\sigma \sim p(\sigma)$, with $p(\sigma) \propto |\psi(\sigma)|^2$, as $E = \mathbb{E}_{\sigma\sim p(\sigma)}[E_{\mathrm{loc}}(\sigma)]$, where $E_{\mathrm{loc}}(\sigma) = (\hat{H}\psi)(\sigma)/\psi(\sigma)$. In the TQS setting, $\psi(\sigma)$ is parameterized by Transformers; we minimize E using stochastic reconfiguration [20], specifically its efficient minimum-step stochastic reconfiguration (minSR) methods [4, 5].

3 Weight-sharing Transformer quantum states

Architecture overview. We propose a weight-sharing TQS (Fig. 1a) that substantially improves parameter efficiency. Given a spin configuration σ, we partition the lattice into N patches and embed each patch as a vector $x_i \in \mathbb{R}^d$ $(i = 1, \cdots, N)$, yielding the embeddings $X_\sigma(0) \in \mathbb{R}^{N\times d}$. We model a single Transformer encoder block, consisting of a *factored* multi-head attention (MHA) sublayer [5, 21] and a position-wise feed-forward network (FFN), as an operator $\hat{T}(\tau)$ characterized by a step size $\tau > 0$. We then apply it L times with parameters shared across depth, producing $X_\sigma(L) = [\hat{T}(\tau)]^L X_\sigma(0)$ (see Physically interpretable encoder block). Finally, we pool $X_\sigma(L)$ to a global latent $z_\sigma(L) = \mathrm{Pool}(X_\sigma(L)) \in \mathbb{R}^d$ and parameterize the variational wave function as $\log\psi(\sigma) = \sum_{a=1}^d \log\cosh\big(b_a + \mathbf{w}_a^\top z_\sigma(L)\big) \in \mathbb{C}$, with complex-valued parameters $\{b_a, \mathbf{w}_a\}_{a=1}^d$.

Physically interpretable encoder block. We interpret repeating the weight-sharing block as a dynamical system in imaginary time. With weight sharing, a single encoder application realizes an operator $\hat{T}(\tau)$, which acts as a short imaginary-time propagator and induces the recursion $X_\sigma(\ell+1) = \hat{T}(\tau)\, X_\sigma(\ell)$, hence $X_\sigma(L) = [\hat{T}(\tau)]^L X_\sigma(0)$ with the total imaginary time $\beta = L\tau$; in the limit $\tau \to 0$ at fixed β, $[\hat{T}(\tau)]^L \to e^{-\beta\hat{G}}$ for a learned generator $\hat{G}$. Because VMC drives the variational wave function toward the ground state, the learned generator behaves as a low-energy

filter. When convergence is achieved, the repeated application of this filter $e^{-\beta\hat{G}}$ suppresses excited-state components, a process analogous to imaginary-time evolution. To expose $\hat{G}$ from the block sub-operations, we define a single shared block generator as $\hat{G} \equiv \hat{G}_{\mathrm{MHA}} + \hat{G}_{\mathrm{FFN}}$, where $\hat{G}_{\mathrm{MHA}}$ and $\hat{G}_{\mathrm{FFN}}$ are the formal generators of the MHA and FFN sublayers, which implicitly include the pre-applied layer normalization. Their actions can be described by $\hat{G}_{\mathrm{MHA}}$ and $\hat{G}_{\mathrm{FFN}}$, such that the updates are written as $X_\sigma \leftarrow X_\sigma + \tau\,\hat{G}_{\mathrm{MHA}} X_\sigma$ and $X_\sigma \leftarrow X_\sigma + \tau\,\hat{G}_{\mathrm{FFN}} X_\sigma$. Furthermore, the generators $\hat{G}_{\mathrm{MHA}}$ and $\hat{G}_{\mathrm{FFN}}$, can be identified with an effective interaction Hamiltonian $\hat{H}_{\mathrm{int}}$ and an onsite Hamiltonian $\hat{H}_{\mathrm{loc}}$, respectively. Specifically, for the factored MHA, the update term defines $\hat{H}_{\mathrm{int}} = -U$, where the interaction of two patches (i, j) is $U_{ij} = \sum_h B^{(h)} \alpha_{ij}^{(h)} V^{(h)}$. Here $\alpha_{ij}^{(h)}$ denotes the attention weight of head h, $V^{(h)}$ is the value projection, and $B^{(h)}$ is the h-th slice of the output projection W_o. For the FFN, the update term defines $\hat{H}_{\mathrm{loc}} = -\sum_i k_i$, where k_i is an effective operator for the local action at patch i. Accordingly, the full block generator corresponds to an effective many-body Hamiltonian, $\hat{G} = \hat{H}_{\mathrm{int}} + \hat{H}_{\mathrm{loc}}$.

Suzuki–Trotter decompositions for TQS. The central challenge is that the ideal propagator $\hat{T}(\tau) = e^{-\tau(\hat{G}_{\mathrm{MHA}} + \hat{G}_{\mathrm{FFN}})}$ cannot be implemented directly because its generators do not commute. Our approach is to approximate $\hat{T}(\tau)$ by systematically composing the block's elementary operations, $e^{-\tau\hat{G}_{\mathrm{MHA}}}$ and $e^{-\tau\hat{G}_{\mathrm{FFN}}}$, using the Suzuki–Trotter decompositions (Fig. 1b). For an n-th order decomposition, the implemented propagator $\hat{T}^{(n)}(\tau)$ approximates the ideal one with an error of $\mathcal{O}(\tau^{n+1})$. The standard encoder block (MHA followed by FFN) [6, 7] realizes the first-order Lie–Trotter scheme [15, 16], where the local Trotter error is $\mathcal{O}(\tau^2)$, and the corresponding propagator is given by $\hat{T}^{(1)} = e^{-\tau\hat{G}_{\mathrm{MHA}}} e^{-\tau\hat{G}_{\mathrm{FFN}}}$. This error accumulates with depth L, reflecting the generic noncommutativity between interaction and onsite terms in many-body Hamiltonians ($[\hat{G}_{\mathrm{MHA}}, \hat{G}_{\mathrm{FFN}}] \neq 0$). To reduce this error, we replace the single block with a higher-order product:

$$\hat{T}^{(n)}(\tau) = \prod_{i=1}^{k} e^{-a_i^{(n)}\tau\,\hat{G}_{\mathrm{FFN}}}\, e^{-b_i^{(n)}\tau\,\hat{G}_{\mathrm{MHA}}}, \tag{2}$$

where k is the number of stages, and $a_i^{(n)}, b_i^{(n)}$ are scheme-dependent coefficients satisfying $\sum_i a_i^{(n)} = \sum_i b_i^{(n)} = 1$. In practice, we instantiate the second-order scheme by Strang [17] and the fourth-order schemes by Suzuki [18] and Blanes–Moan [19]. The key advantage of this design is that it allows us to increase the decomposition order, and thus the accuracy, *without adding any trainable parameters*. This is achieved by constructing higher-order schemes simply by reordering and rescaling the same underlying MHA and FFN sub-operations within the block. This provides a powerful mechanism to trade a modest increase in computational cost for a systematic reduction in Trotter error, all within a fixed parameter budget. This parameter-efficient strategy for error reduction stands in contrast to methods like MacaronNet [13], which also embed a second-order decomposition but do not share weights, thereby increasing the parameter count with depth.

4 Results

We evaluate the proposed weight-sharing TQS on the frustrated square-lattice J_1–J_2 Heisenberg model at $J_2/J_1 = 0.5$ on a 10×10 lattice. For all experiments, we use a patch size of $N = 2 \times 2$. Implementations are built on NetKet [22, 23], and we use minSR [4, 5] for optimization, with learning rate set to 0.0075 and diagonal shift set to 10^{-4}. We set the short imaginary-time step to $\tau = 0.5$, so the controlled knobs are the decomposition order n and the number of shared blocks L.

Decomposition order. To study how the decomposition order n impacts accuracy, we perform a comparative analysis of different schemes at a fixed encoder depth of $L = 4$ (i.e., $\beta = 2.0$). For our weight-sharing TQS, we use a configuration with $d = 60$ and $n_h = 10$ attention heads, resulting in $44{,}890$ trainable parameters. We also evaluate a non-weight-sharing TQS baseline [5], which would require $155{,}620$ parameters (about $3.5\times$ more) under the same hyperparameter configuration as our models. Running the optimization on two NVIDIA A100 40GB GPUs for 800 iterations with $4{,}096$ samples, we find a clear trend (Fig. 2a): the higher-order schemes monotonically improve the mean energy per site relative to the first-order Lie–Trotter scheme. In particular, among all schemes,

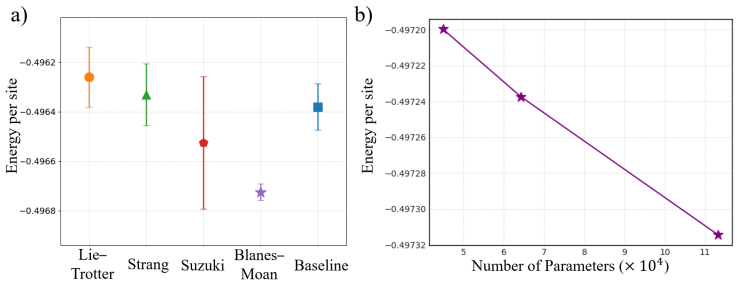

Figure 2: (a) Energies per site of four Suzuki–Trotter decomposition schemes and the non-weight-sharing TQS baseline. Each point shows the mean ± s.e.m. of a method over five seeds. (b) Energies per site of three models with different model sizes using the Blanes–Moan scheme.

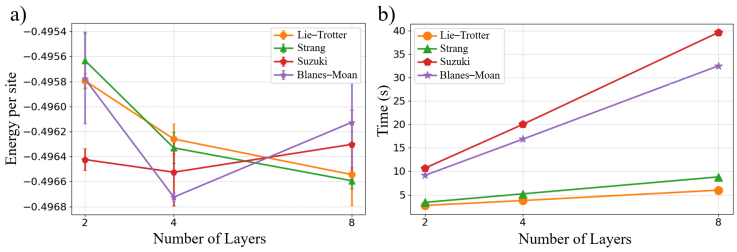

Figure 3: (a) Energies per site and (b) computation time per optimization step of four Suzuki–Trotter decomposition schemes with different encoder depths. A marker in (a) indicates the mean ± s.e.m. of a method over five seeds.

the fourth-order Blanes–Moan ansatz achieves the lowest energy per site, indicating that raising n effectively reduces the Trotter error while maintaining robust optimization. The best energies per site across seeds for the different schemes are non-weight-sharing baseline $-0.49666(9)$, Lie–Trotter $-0.49669(9)$, Strang $-0.49671(9)$, Suzuki $-0.49697(9)$, and Blanes–Moan $-0.49683(7)$, where parentheses indicate statistical (Monte Carlo) errors. Both fourth-order schemes thus reach lower energies than the Lie–Trotter, Strang, and non-weight-sharing baseline. While the Suzuki scheme attains the lowest best energy, the Blanes–Moan variant yields the lowest energy across seeds and the smallest seed-to-seed variance (as reflected in its standard deviation), making it the most robust fourth-order choice in practice. Notably, these schemes achieve lower variational energies than the non-weight-sharing baseline, highlighting the superior accuracy of our parameter-efficient approach. Furthermore, when we restrict the non-weight-sharing baseline to the same number of trainable parameters as our weight-sharing TQS (corresponding to a single encoder layer), the best energy per site across seeds is $-0.49335(21)$, indicating that the baseline fails to achieve a comparable level of accuracy under an equal parameter budget.

Parameter scaling. We next investigate the scaling behavior of our approach with $8{,}192$ samples per iteration and $3{,}000$ optimization iterations, using two NVIDIA H100 80GB GPUs. We compare three models of the Blanes–Moan scheme and $L = 4$, which have $44{,}890$, $64{,}236$, and $113{,}296$ parameters where (d, n_h) correspond to $(60, 10)$, $(72, 12)$, and $(96, 16)$, respectively. Fig. 2b shows that the larger models obtain the better per-site energies from -0.49725 to -0.49730. The results also match the accuracy reported in the previous study [5]. Note that our models have fewer parameters than the non-weight-sharing ones in [5], which typically require $267{,}720$ to $994{,}700$ parameters.

Impact of total imaginary time. Finally, we analyze how the total imaginary time $\beta = L\tau$ affects the accuracy and efficiency of our approach. We fix the time step at $\tau = 0.5$ and change the encoder depth L to 2, 4, and 8. All other settings are identical to those in Decomposition order. As shown in Fig. 3a, the lower-order schemes exhibit a nearly monotonic decrease in energy as L increases. This is consistent with the interpretation of greater depth acting as a stronger low-energy projector as the total imaginary time $\beta = L\tau$ grows. In contrast, the higher-order Suzuki and Blanes–Moan schemes show limited improvement or even slight degradation at the largest depth of $L = 8$. Interestingly, when we

inspect individual runs for the Blanes–Moan scheme, the best seed at $L = 8$ attains an energy per site of $-0.49681(8)$, which is essentially as good as the best seed at $L = 4$, $-0.49683(7)$. At the same time, across seeds the standard deviation of the energy per site increases from 3.22×10^{-5} at $L = 4$ to about 3.60×10^{-4} at $L = 8$, i.e., by roughly an order of magnitude. This indicates that the underlying high-order approximation does not fundamentally deteriorate with depth and can in principle reach equally accurate energies, but optimization instabilities and stochastic noise become much more pronounced as L increases, effectively limiting the practical benefit of going beyond $L = 4$ at the fixed step size $\tau = 0.5$. Fig. 3b shows that the lower-order schemes are significantly faster than the fourth-order schemes. These results support that controlling the encoder depth L provides a physically interpretable trade-off between accuracy and computation time. From the results, we suggest the Strang scheme with $L = 8$ and the Blanes–Moan scheme with $L = 4$ as reasonable choices to balance accuracy and computational efficiency.

5 Conclusion

We introduced a weight-sharing TQS that models the encoder block as a short imaginary-time propagator and improves accuracy at a fixed parameter budget via Suzuki–Trotter decompositions. Our results on the square-lattice J_1–J_2 Heisenberg model showed that our approach achieved lower energy and matched recent non-weight-sharing baselines with fewer parameters. Future directions include reducing the computational overhead of higher-order schemes and applying other quantum many-body systems, e.g., fermion systems.

References

[1] Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017.

[2] Yusuke Nomura, Andrew S. Darmawan, Youhei Yamaji, and Masatoshi Imada. Restricted Boltzmann machine learning for solving strongly correlated quantum systems. *Phys. Rev. B*, 96: 205152, 2017.

[3] Yusuke Nomura and Masatoshi Imada. Dirac-Type Nodal Spin Liquid Revealed by Refined Quantum Many-Body Solver Using Neural-Network Wave Function, Correlation Ratio, and Level Spectroscopy. *Phys. Rev. X*, 11:031034, 2021.

[4] Ao Chen and Markus Heyl. Empowering deep neural quantum states through efficient optimization. *Nature Physics*, 20(9):1476–1481, 2024.

[5] Riccardo Rende, Luciano Loris Viteritti, Lorenzo Bardone, Federico Becca, and Sebastian Goldt. A simple linear algebra identity to optimize large-scale neural network quantum states. *Communications Physics*, 7(1):260, 2024.

[6] Luciano Loris Viteritti, Riccardo Rende, and Federico Becca. Transformer Variational Wave functions for Frustrated Quantum Spin Systems. *Phys. Rev. Lett.*, 130:236401, 2023.

[7] Luciano Loris Viteritti, Riccardo Rende, Alberto Parola, Sebastian Goldt, and Federico Becca. Transformer wave function for two dimensional frustrated magnets: Emergence of a spin-liquid phase in the Shastry-Sutherland model. *Phys. Rev. B*, 111:134411, 2025.

[8] Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. Universal Transformers. In *International Conference on Learning Representations*, 2019.

[9] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A Lite BERT for Self-supervised Learning of Language Representations. In *International Conference on Learning Representations*, 2020.

[10] Angeliki Giannou, Shashank Rajput, Jy-Yong Sohn, Kangwook Lee, Jason D. Lee, and Dimitris Papailiopoulos. Looped Transformers as Programmable Computers. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 11398–11442, 2023.

[11] Sangmin Bae, Adam Fisch, Hrayr Harutyunyan, Ziwei Ji, Seungyeon Kim, and Tal Schuster. Relaxed Recursive Transformers: Effective Parameter Sharing with Layer-wise LoRA. In *International Conference on Learning Representations*, 2025.

[12] Ying Fan, Yilun Du, Kannan Ramchandran, and Kangwook Lee. Looped Transformers for Length Generalization. In *International Conference on Learning Representations*, 2025.

[13] Yiping Lu, Zhuohan Li, Di He, Zhiqing Sun, Bin Dong, Tao Qin, Liwei Wang, and Tie yan Liu. Understanding and Improving Transformer From a Multi-Particle Dynamic System Point of View. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*, 2019.

[14] Borjan Geshkovski, Cyril Letrouit, Yury Polyanskiy, and Philippe Rigollet. A mathematical perspective on transformers. *Bulletin of the American Mathematical Society*, 62(3):427–479, 2025.

[15] Sophus Lie. *Theorie der Transformationsgruppen*, volume 1. BG Teubner, 1888.

[16] Hale Freeman Trotter. On the Product of Semi-Groups of Operators. *Proceedings of the American Mathematical Society*, 10(4):545–551, 1959.

[17] Gilbert Strang. On the Construction and Comparison of Difference Schemes. *SIAM Journal on Numerical Analysis*, 5(3):506–517, 1968.

[18] Masuo Suzuki. Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations. *Physics Letters A*, 146(6):319–323, 1990.

[19] Sergio Blanes and Per Christian Moan. Practical symplectic partitioned Runge–Kutta and Runge–Kutta–Nyström methods. *Journal of Computational and Applied Mathematics*, 142(2): 313–330, 2002.

[20] Sandro Sorella. Generalized Lanczos algorithm for variational quantum Monte Carlo. *Phys. Rev. B*, 64:024512, 2001.

[21] Riccardo Rende and Luciano Loris Viteritti. Are queries and keys always relevant? A case study on transformer wave functions. *Machine Learning: Science and Technology*, 6(1):010501, 2025.

[22] Giuseppe Carleo, Kenny Choo, Damian Hofmann, James E. T. Smith, Tom Westerhout, Fabien Alet, Emily J. Davis, Stavros Efthymiou, Ivan Glasser, Sheng-Hsuan Lin, Marta Mauri, Guglielmo Mazzola, Christian B. Mendl, Evert van Nieuwenburg, Ossian O'Reilly, Hugo Théveniaut, Giacomo Torlai, Filippo Vicentini, and Alexander Wietek. Netket: A Machine Learning Toolkit for Many-Body Quantum Systems. *SoftwareX*, page 100311, 2019.

[23] Filippo Vicentini, Damian Hofmann, Attila Szabó, Dian Wu, Christopher Roth, Clemens Giuliani, Gabriel Pescia, Jannes Nys, Vladimir Vargas-Calderón, Nikita Astrakhantsev, and Giuseppe Carleo. NetKet 3: Machine Learning Toolbox for Many-Body Quantum Systems. *SciPost Phys. Codebases*, page 7, 2022.