

Functional Tensor-Train Chebyshev Method for Multidimensional Quantum Dynamics Simulations

Micheline B. Soley,^{†,‡} Paul Bergold,[¶] Alex A. Gorodetsky,[§] and Victor S.
Batista^{*,†,‡,¶}

[†] *Yale Quantum Institute, Yale University,
P.O. Box 208334, New Haven, CT, 06520-8263, USA*

[‡] *Department of Chemistry, Yale University,
P.O. Box 208107, New Haven, CT, 06520, USA*

[¶] *Zentrum Mathematik, Technical University of Munich,
Boltzmannstr. 3, 85748 Garching, Germany*

[§] *Department of Aerospace Engineering, University of Michigan,
1320 Beal Avenue Ann Arbor, MI 48109-2140, USA*

^{||} *Energy Sciences Institute, Yale University,
P.O. Box 27394, West Haven, CT, 06516-7394, USA*

E-mail: victor.batista@yale.edu

Abstract

Methods for efficient simulations of multidimensional quantum dynamics are essential for theoretical studies of chemical systems where quantum effects are important, such as those involving rearrangements of protons or electronic configurations. Here, we introduce the *functional tensor-train Chebyshev* (FTTC) method for rigorous nuclear quantum dynamics simulations. FTTC is essentially the Chebyshev propagation scheme applied to the initial state represented in a continuous analogue tensor-train format. We demonstrate the capabilities of FTTC as applied to simulations of proton quantum dynamics in a 50-dimensional model of hydrogen-bonded DNA base pairs.

1 Introduction

Quantum dynamics simulations are essential for rigorous theoretical studies of quantum reaction dynamics, including applications to structural and dynamical problems with critical rearrangements of protons or electronic configurations. A variety of approaches have been developed, including time-dependent Hartree methods^{1–11} and other methods based on short-time approximations of the time-evolution operator, such as the Trotter expansion and finite difference methods.^{12–24} Here, we focus on the Chebyshev method for simulation of quantum wavepacket dynamics, which enables computation of the time-evolved quantum state at the final time without having to compute intermediate states at earlier times.¹² Thus, contrary to methods based on short-time propagators, the Chebyshev propagation scheme can be implemented without error accumulation. Chebyshev propagation is currently one of the foremost approaches for simulations of quantum dynamics in low dimensionality,^{12,25–27} as demonstrated for nuclear quantum dynamics simulations of molecular systems with up to six dimensions.^{28,29} However, applications to higher-dimensional systems have been hindered by the exponential scaling of memory and computational cost with dimensionality, due to its reliance on full-grid representations. Here, we introduce a viable solution to the exponential scaling with dimensionality by applying the Chebyshev propagation scheme to the initial state represented in functional tensor-train format (FT) – i.e., the continuous analogue of the tensor-train/matrix product state decomposition. The resulting functional tensor-train Chebyshev (FTTC) method is demonstrated as applied to simulations of proton dynamics in a high-dimensional (50-dimensional) model of hydrogen-bonded adenine-thymine DNA base pairs, where photo-induced proton transfer has long been thought to have important biological implications (*e.g.*, photoinduced mutations).³⁰

Tensor networks have been successfully applied to a wide range of studies, including applications to solving partial differential equations,^{31–34} quantum dynamics methods,^{35–55} machine learning,⁵⁶ electronic structure calculations,^{57–62} and calculations of vibrational states.^{63–65}

Here, we build upon the tensor-train split-operator Fourier transform (TT-SOFT) method,⁴⁰ and we develop the FTTC method which is essentially a functional tensor-train implementation of the Chebyshev propagation scheme, popularized by Kosloff and co-workers.²⁵ FTTC expands the initial state as a functional tensor train and evolves it by applying the Chebyshev expansion of the time-evolution operator,

$$e^{-it\hat{H}} \approx \sum_{k=0}^{N-1} (2 - \delta_{k,0}) (-i)^k J_k(t) T_k(\hat{\mathcal{H}}) \quad (1)$$

where $T_k(\hat{\mathcal{H}})$ are Chebyshev polynomials of the Hamiltonian \hat{H} in functional tensor-train format, $J_k(t)$ are the Bessel functions of the first kind, and t is the final propagation time. In practice, a finite number $N \geq 1$ of polynomials is employed and the expansion is applied iteratively in time. Important advantages of the proposed FTTC algorithm when compared to other propagation methods based on matrix product states are: (i) the error need not accumulate with time since the state at time t can be obtained directly without having to compute earlier intermediate states and (ii) the uniform character of the Chebyshev expansion that decreases the error exponentially with N .

We focus on functional tensor trains (FT),^{66–69} which are continuous analogues of tensor trains/matrix product states and have yet to be demonstrated as applied to simulations of quantum nuclear dynamics. Functional tensor-train representations of time-dependent states allow for efficient computations of gradients of multidimensional tensors, so they are expected to be particularly valuable for a variety of applications, including studies of the quantum control of molecular systems.⁷⁰ Therefore, we can implement Eq. (1) as applied to the time-evolving state directly by computing functional tensor-train decompositions of the Chebyshev polynomials applied to the time-evolved wave function without having to pre-compute the Chebyshev polynomials of the Hamiltonian. The FTTC method can also be implemented using discrete tensor trains,^{71,72} instead of their continuous analogues, allowing for efficient representation and manipulation of matrix product states.⁷³

The article is organized as follows. Section 2 describes the Chebyshev methodology. Section 2.1 introduces the Chebyshev polynomials. Section 2.2 describes how to generate Chebyshev expansions of complex-valued functions, and Section 2.3 describes Chebyshev propagation based on discrete space representations. Section 3 describes the functional tensor-train decomposition, as a continuous analogue tensor-train format for multilinear algebra manipulations of high-dimensional tensors. Section 4 describes our functional tensor-train Chebyshev (FTTC) propagation method as implemented for numerical integration of the time-dependent Schrödinger equation. Section 6 demonstrates the capabilities of FTTC as applied to simulations of proton quantum dynamics in a 50-dimensional model of DNA base pairs with highly anharmonic modes. The Supporting Information demonstrates that tensor-train Chebyshev dynamics can entail lower computational cost relative to the state-of-the-art split operator Fourier transform method for long time steps. The results show that FTTC enables simulations of molecular systems far beyond the capabilities of the standard grid-based Chebyshev method.

2 Chebyshev Approach

The Chebyshev propagation method²⁵ integrates numerically the time-dependent Schrödinger equation

$$i \frac{\partial \Psi}{\partial t} = \hat{H} \Psi \quad (2)$$

where we have used atomic units ($\hbar = 1$). For simplicity, we consider a system described by the Hamiltonian

$$\hat{H} = \frac{\hat{p} \cdot \hat{p}}{2m} + \hat{V} = -\frac{1}{2m} \Delta_x \Psi + \hat{V} \quad (3)$$

where $m > 0$ is the mass of the system, $\hat{p} = -i\nabla_x$ is the momentum operator, and $V: \mathbb{R}^d \rightarrow \mathbb{R}$ is a given potential energy surface (PES) describing interactions that rule the underlying

dynamics of the system. Using the unitary evolution propagator $\hat{U}(t) = e^{-it\hat{H}}$, the solution of Eq. (2), $\Psi(t)$, corresponding to the initial state Ψ_0 is

$$\Psi(t) = \hat{U}(t)\Psi_0. \quad (4)$$

The Chebyshev propagation method approximates the propagator $\hat{U}(t) = e^{-it\hat{H}}$ for a fixed final time t in terms of a linear combination of the first $N \geq 1$ Chebyshev polynomials of the Hamiltonian $T_0(\hat{H}), \dots, T_{N-1}(\hat{H})$, as discussed in Section 2.1.

2.1 Chebyshev Polynomials

For all integers $k \geq 0$ and all $x \in [-1, 1]$, the k th Chebyshev polynomial is defined as follows

$$T_k(x) = \cos(k \arccos(x)) \quad (5)$$

where \arccos is the inverse of the cosine (*i.e.*, $\cos(\arccos(x)) = \arccos(\cos(x)) = x$). We note that the Chebyshev polynomials are defined only for input values $x \in [-1, 1]$, since the cosine function attains values only in that limited range, and satisfy the following recurrence relation

$$T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x) \quad (6)$$

with $T_0(x) = 1$ and $T_1(x) = x$ defining the subsequent Chebyshev polynomials, so the first four polynomials are

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_2(x) = 2x^2 - 1, \quad T_3(x) = 4x^3 - 3x. \quad (7)$$

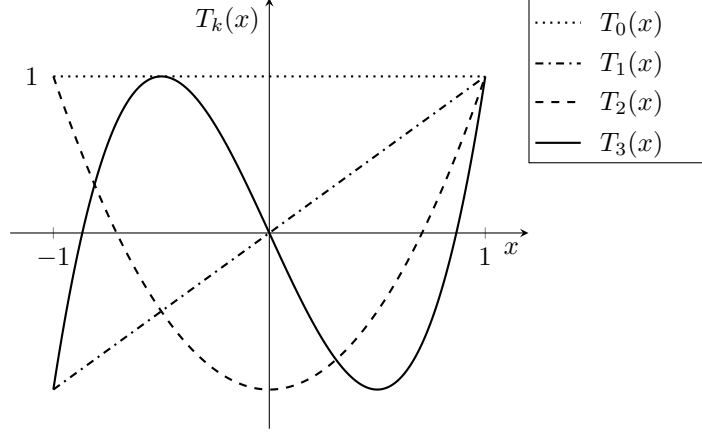


Figure 1: Plot of the first four Chebyshev polynomials, defined only in the limited interval $[-1, 1]$, since the possible values of the cosine function are limited to that interval.

We note that Chebyshev polynomials have a number of remarkable properties and are therefore an important tool in the field of approximation theory.^{74,75} For instance, let us remark that they satisfy the following orthogonality relation for all $j, k \geq 1$, $j \neq k$,

$$\int_{-1}^1 \frac{dx}{\sqrt{1-x^2}} T_j(x) T_k(x) = \frac{\pi}{2} \delta_{j,k} \quad (8)$$

showing that the Chebyshev polynomials are orthogonal with respect to the weighted inner product defined by the left hand side of Eq. (8).

2.2 Chebyshev Expansion of Complex-Valued Functions

Chebyshev polynomials can be used to approximate a given complex-valued function f via its Fourier series representation. To show that, we introduce the 2π -periodic function⁷⁵

$$g(x) = f(\cos(x)) \quad (9)$$

which can be represented in the interval (*e.g.*, $-\pi < x < \pi$) in terms of its Fourier series as follows

$$g(x) = \sum_{k=0}^{\infty} (2 - \delta_{k,0}) a_k \cos(kx), \quad a_k = \frac{1}{\pi} \int_0^{\pi} g(x) \cos(kx) dx. \quad (10)$$

Therefore

$$f(y) = g(\arccos(y)) \quad (11)$$

can be represented in terms of the Chebyshev polynomials as follows

$$f(y) = \sum_{k=0}^{\infty} (2 - \delta_{k,0}) c_k T_k(y), \quad c_k = \frac{1}{\pi} \int_{-1}^1 \frac{dy}{\sqrt{1-y^2}} f(y) T_k(y), \quad (12)$$

for $y \in [-1, 1]$. Equation (12) is called the Chebyshev expansion of f and it can be used to approximate f as the linear combination of the first N Chebyshev polynomials as follows

$$f(y) \approx S_N f(y) = \sum_{k=0}^{N-1} (2 - \delta_{k,0}) c_k T_k(y). \quad (13)$$

The coefficients c_k , defined by Eq. (12), are essentially the Fourier coefficients of the function g that decay exponentially with N for analytic functions⁷⁶ (*i.e.*, smooth in the complex domain) and thus provide fast convergence of the partial sums $S_N f$. The resulting Chebyshev approximant $S_N f$ is a polynomial of degree N , which is known to be close to the polynomial of the same degree with minimal error in the interval $[-1, 1]$.⁷⁷

2.3 Chebyshev Propagation in Discrete Representations

We obtain an approximation of the operator $\hat{U} = e^{-it\hat{H}}$ at time t by considering the function $f(y) = e^{-ity}$ for which the coefficients c_k defined according to Eq. (12) can be expressed in

terms of the Bessel functions J_k (of the first kind) as follows⁷⁸

$$c_k = (-i)^k J_k(t) \quad (14)$$

yielding the following approximation

$$e^{-ity} \approx \sum_{k=0}^{N-1} (2 - \delta_{k,0}) (-i)^k J_k(t) T_k(y), \quad (15)$$

with $y \in [-1, 1]$. Using a linear transformation of the argument y , we can restate Eq. (15) for an arbitrary Hermitian matrix $H \in \mathbb{C}^{D \times D}$ (where $D > 1$ is a positive integer) with eigenvalues contained in a finite interval $[a, b]$ as follows

$$e^{-itH} \approx e^{-it^+} \sum_{k=0}^{N-1} (2 - \delta_{k,0}) (-i)^k J_k(t^-) T_k(H_0) \quad (16)$$

where we have introduced the rescaled variables $t^-, t^+ \in \mathbb{R}$, and the matrix $H_0 \in \mathbb{C}^{D \times D}$ with eigenvalues in $[-1, 1]$ defined as follows

$$t^\pm = \frac{t}{2}(b \pm a) \quad \text{and} \quad H_0 = \frac{2}{b-a} \left(H - \frac{b+a}{2} I_D \right) \quad (17)$$

where I_D is the $D \times D$ identity matrix.

Fast convergence is typically obtained for e^{-ity} since it is a smooth function, although the number of required polynomials increases with t since e^{-ity} is oscillatory. Thus, a sufficiently large number N of Chebyshev polynomials is needed to resolve the oscillations. In fact, it has been shown that the error falls like the N th order in $|t^-|/(2N)$ for sufficiently large N .⁷⁹

It is important to note that Eq. (16) can be used more generally than in the current implementation to approximate the solution to any linear system of the form $i\dot{u} = Hu$. Such linear systems typically arise in space discretization methods, including the Fourier collocation method, the Fourier Galerkin method, or the Hermite Galerkin method.⁷⁹ So,

we anticipate that the FTTC method should also be valuable for solving high-dimensional linear systems in a wide range of applications beyond the solution of the time-dependent Schrödinger equation.

2.4 Discrete Tensor-Train Implementation

Discrete tensor-train approximations of $e^{-it\hat{H}}$ are obtained by discretizing the d -dimensional space with a uniform grid of size $\Delta x_j > 0$ for the j th nuclear coordinate, spanning the range $x_{j,\min}$ to $x_{j,\max}$, with $n_j > 1$ points for each dimension $j = 1, \dots, d$. Analogously, discrete tensor-train representations of wave functions are obtained as low-rank d -dimensional complex-valued tensor trains approximating

$$\mathcal{W}[k_1, \dots, k_d] = \Psi(x_k) \quad (18)$$

where $k = (k_1, \dots, k_d)$ are the indices of tensor-train entries corresponding to nuclear coordinate values $x_k = (x_{1,k_1}, \dots, x_{d,k_d})$. The discrete tensor-train representation of $\mathcal{W}[k_1, \dots, k_d]$ is defined as follows

$$\mathcal{W}[k_1, k_2, \dots, k_d] = \mathcal{W}_1[k_1] \mathcal{W}_2[k_2] \cdots \mathcal{W}_d[k_d], \quad 1 \leq k_j \leq n_j \text{ for all } j, \quad (19)$$

where $\mathcal{W}_j[k_j] \in \mathbb{R}^{r_{j-1} \times r_j}$ are matrices and n_j are the number of grid points in the j th coordinate direction.

The action of the Hamiltonian \hat{H} on a wave function Ψ is represented by the Hermitian operator $\hat{\mathcal{H}} = \hat{\mathcal{T}} + \mathcal{V}$. The real-valued “potential energy tensor”

$$\mathcal{V}[k_1, \dots, k_d] = V(x_{1,k_1}, \dots, x_{d,k_d}) \quad (20)$$

acts on $\mathcal{W}[k_1, \dots, k_d]$ as an element-wise multiplication operator (Hadamard product). The

discrete kinetic energy operator

$$(\hat{\mathcal{T}}\mathcal{W})[k_1, \dots, k_d] \approx -\frac{1}{2m}\Delta_x\Psi(x_k), \quad (21)$$

is defined by the Laplacian Δ_x that acts as a multiplication operator in momentum space. Therefore, we apply the kinetic energy operator in momentum space by exploiting the highly efficient (linear scaling with dimensionality) implementation of multidimensional discrete Fourier transform of tensor trains to switch between position and momentum space. With the help of the fast Fourier transform (FFT), we therefore obtain a very efficient and accurate implementation of the discretized kinetic energy operator.

The discrete Hamiltonian is rescaled, according to Eq. (17), as follows

$$\hat{\mathcal{H}}_0 = \frac{2}{E_{\max} - E_{\min}} \left(\hat{\mathcal{H}} - \frac{E_{\max} + E_{\min}}{2} \hat{\mathcal{I}} \right) \quad (22)$$

where $\hat{\mathcal{I}}$ denotes the identity on the tensor space. The bounds for the eigenvalues E_{\min} and E_{\max} depend on the extension of the grid and are given by

$$E_{\min} = \min_k \mathcal{V}[k_1, \dots, k_d], \quad E_{\max} = \max_k \mathcal{V}[k_1, \dots, k_d] + \frac{\pi^2}{2m} \left(\frac{1}{\Delta x_1^2} + \dots + \frac{1}{\Delta x_d^2} \right) \quad (23)$$

where we used $\Delta p_j = 2\pi/(x_{j,\max} - x_{j,\min})$ for the grid size in momentum space of the j th coordinate, giving the maximum kinetic energy

$$\frac{1}{2m}p_{j,\max}^2 = \frac{1}{2m} \frac{\pi^2}{\Delta x_j^2}. \quad (24)$$

Consequently, the solution $\Psi(t)$ is approximated with N Chebyshev polynomials as follows

$$\Psi(t) = e^{-it\hat{H}}\Psi(0) \approx e^{-it^\pm} \sum_{k=0}^{N-1} (2 - \delta_{k,0}) (-i)^k J_k(t^\pm) T_k(\hat{\mathcal{H}}_0) \mathcal{W}_0 \quad (25)$$

where $t^\pm = tE^\pm/2$, $E^\pm = E_{\max} \pm E_{\min}$ and \mathcal{W}_0 samples the initial wave function.

We implement Eq. (25) as a one-step propagator to compute $\Psi(t)$ directly from the initial data by using the Clenshaw algorithm⁸⁰ (see Appendix A). Alternatively, we can obtain the time-dependent states $T_k(\hat{\mathcal{H}}_0)\mathcal{W}_0$ according to the recurrence relation Eq. (6)

$$\begin{aligned} T_0(\hat{\mathcal{H}}_0)\mathcal{W}_0 &= \mathcal{W}_0, \\ T_1(\hat{\mathcal{H}}_0)\mathcal{W}_0 &= \hat{\mathcal{H}}_0\mathcal{W}_0, \\ T_{k+1}(\hat{\mathcal{H}}_0)\mathcal{W}_0 &= 2\hat{\mathcal{H}}_0T_k(\hat{\mathcal{H}}_0)\mathcal{W}_0 - T_{k-1}(\hat{\mathcal{H}}_0)\mathcal{W}_0, \quad \text{for } k \geq 1. \end{aligned} \tag{26}$$

The same Chebyshev propagation scheme described in this section for discrete tensor-train (TT) decompositions⁷² can be readily implemented using the continuous analogue functional tensor-train decomposition,⁶⁶ as described in the following section.

3 Functional Tensor-Train Decomposition

3.1 Continuous Analogue of the Tensor-Train Decomposition

Following refs 68 and 69, here we give a brief overview of the functional tensor-train (FT) format, which offers an efficient data compression scheme^{66,67}

$$f(x_1, x_2, \dots, x_d) = \sum_{i_0=1}^{r_0} \sum_{i_1=1}^{r_1} \cdots \sum_{i_d=1}^{r_d} f_1^{(i_0 i_1)}(x_1) f_2^{(i_1 i_2)}(x_2) \cdots f_d^{(i_{d-1} i_d)}(x_d), \tag{27}$$

where $f_k^{(ij)} : \mathcal{X}_k \rightarrow \mathbb{R}$, \mathcal{X}_k denotes the domain of the k th physical dimension, with $r_0 = r_d = 1$ for single-output functions such as polynomials or linear elements. The FT decomposition, introduced by Eq. (27), is a low-rank decomposition of multivariate functions in an analogous way that the TT decomposition, Eq. (19), is a low-rank decomposition of multivariate arrays. A more compact expression for the FT analogue is obtained by viewing a function value as a set of products of matrix-valued functions,

$$f(x_1, x_2, \dots, x_d) = \mathcal{F}_1(x_1)\mathcal{F}_2(x_2) \cdots \mathcal{F}_d(x_d), \tag{28}$$

where each matrix-valued function $\mathcal{F}_k : \mathcal{X}_k \rightarrow \mathbb{R}^{r_{k-1} \times r_k}$ is called a *core* and can be visualized as an array of the univariate functions

$$\mathcal{F}_k(x_k) = \begin{bmatrix} f_k^{(11)}(x_k) & \cdots & f_k^{(1r_k)}(x_k) \\ \vdots & \ddots & \vdots \\ f_k^{(r_{k-1}1)}(x_k) & \cdots & f_k^{(r_{k-1}r_k)}(x_k) \end{bmatrix}. \quad (29)$$

If each univariate function is represented with p parameters (for example, coefficients of a polynomial) and $r_k \leq r$ for all k , then the storage complexity scales as $\mathcal{O}(dpr^2)$. Comparing this representation with Eq. (19), we see a very close resemblance between the TT cores and the FT cores. Indeed, they are both matrices when indexed by a discrete index i_k for the TT or a continuous index x_k for the FT.

3.2 Parameterizations of Low-Rank Functions

The finer structure of FT format is described by the FT cores comprised of d sets of univariate functions $(\mathcal{F}_k)_{k=1}^d$. Each set could be different for different dimensions (*e.g.*, 2π -periodic functions could represent physical dimensions corresponding to torsional angles, while Hermite polynomials could represent stretching modes). As a result, the full FT is parameterized through the parameterization of the set of univariate functions of each dimension, as chosen for the optimal representation of physical coordinates. Considering that $p_{kij} \in \mathbb{Z}_+$ denotes the number of parameters describing $f_k^{(ij)}$, and $\boldsymbol{\theta} \in \mathbb{R}^{p_t}$ the vector of parameters of all of the univariate functions, then there are a total of $p_t \equiv \sum_{k=1}^d \sum_{i=1}^{r_{k-1}} \sum_{j=1}^{r_k} p_{kij}$ parameters describing the FT representation.

The parameter vector $\boldsymbol{\theta}$ is indexed by a multi-index $\boldsymbol{\alpha} = (k, i, j, \ell)$ where $k = 1, \dots, d$ corresponds to an input variable, $i = 1, \dots, r_{k-1}$ and $j = 1, \dots, r_k$ correspond to a univariate function within the k th core, and $\ell = 1, \dots, p_{kij}$ corresponds to a specific parameter within that univariate function. In other words, we adopt the convention that $\boldsymbol{\theta}_{\boldsymbol{\alpha}} = \boldsymbol{\theta}_{kij\ell}$ refers to the ℓ th parameter of the univariate function in the i th row and j th column of the k th core.

The additional flexibility of the representation allows both linear and nonlinear parameterizations of univariate functions. In particular, the linear parameterization represents a univariate function as an expansion of basis functions $\left(\phi_{k\ell}^{(ij)} : \mathcal{X}_k \rightarrow \mathbb{R}\right)_{\ell=1}^{p_{kij}}$ according to

$$f_k^{(ij)}(x_k; \boldsymbol{\theta}) = \sum_{\ell=1}^{p_{kij}} \theta_{kij\ell} \phi_{k\ell}^{(ij)}(x_k). \quad (30)$$

3.3 Low-Rank Functions vs Low-Rank Coefficients

For greater versatility, the FT can be used by independently parameterizing the univariate functions of each core, and both linear and nonlinear parameterizations are possible. As described below, the advantage of this representation includes a naturally sparse storage scheme for the cores.

An advantage of this type of structure is that it readily enables adaptivity when performing common multilinear algebraic operations with functions in low-rank format.⁶⁸ For example, taking the product of two functions in low-rank format requires computing products between univariate functions in corresponding cores of the two functions. In particular, it requires computing the product between every combination of the functions. Because we store each univariate function separately, this product can accurately account for the complexity of the resulting univariate function. For instance, if the product of two third-order polynomials is considered, then a sixth-order polynomial will be stored. However, if a third-order polynomial is multiplied by a first-order polynomial, then only a fourth-order polynomial needs to be stored. In contrast, traditional tensor-based storage schemes would require storing each univariate function with the same number of polynomials. These advantages arise because we consider Chebfun-style continuous computation.⁸¹ The salient point is that we consider univariate functions, rather than matrices or lower-order arrays, as the building blocks of low-rank representations. Another advantage is the availability of efficient computational algorithms for multilinear algebra that can adapt the representation of each univariate function individually as needed in the spirit of continuous computation pioneered

by Chebfun.⁸¹

The TT/MPS format is a particular case of the general FT decomposition, naturally arising when two simplifying assumptions are made^{82,83}

1. linear parameterization of each $f_k^{(ij)}$;
2. identical basis for the functions within each FT core, i.e., $p_{kij} = p_k$ and $\phi_{k\ell}^{(ij)} = \phi_{k\ell}$ for all $i = 1, \dots, r_{k-1}$, $j = 1, \dots, r_k$, and $\ell = 1, \dots, p_k$;

These assumptions transform the problem of storing low-rank functions to the problem of storing low-rank coefficients, allowing the use of discrete TT algorithms and theory. Both representations store the coefficients of a tensor-product basis $\phi_{k\ell}$ for all k and ℓ .

Function evaluations can be obtained from the coefficients defining the tensor $\mathcal{F}_k \in \mathbb{R}^{r_{k-1} \times p_k \times r_k}$ of the following form

$$\mathcal{F}_k[:, \ell, :] = \begin{bmatrix} \theta_{k11\ell} & \cdots & \theta_{k1r_k\ell} \\ \vdots & \ddots & \vdots \\ \theta_{kr_{k-1}1\ell} & \cdots & \theta_{kr_{k-1}r_k\ell} \end{bmatrix}, \quad (31)$$

for $\ell = 1, \dots, p_k$, by performing the following summation

$$f(x_1, \dots, x_d) = \sum_{\ell_1=1}^{p_1} \cdots \sum_{\ell_d=1}^{p_d} \mathcal{F}_1[:, \ell_1, :] \cdots \mathcal{F}_d[:, \ell_d, :] \phi_{1\ell_1}(x_1) \cdots \phi_{d\ell_d}(x_d). \quad (32)$$

From Eqs. (28), (31), and (32) we can see that the relationship between the TT cores \mathcal{F}_k and the FT cores \mathcal{F}_k is

$$\mathcal{F}_k(x_k) = \sum_{\ell=1}^{p_k} \mathcal{F}_k[:, \ell, :] \phi_{k\ell}(x_k), \quad (33)$$

where the basis function multiplies every element of the tensor. In other words, the FT cores represent a TT decomposition of the $p_1 \times p_2 \times \cdots \times p_d$ coefficient tensor of a tensor-product basis and inherit the properties of the TT decomposition.

3.4 Operations in the FT Format

Performing continuous multilinear algebra is one of the main advantages of the continuous framework. The operations of addition, multiplication, differentiation, integration, and inner products are easily performed for functions in the FT format as follows.⁶⁸ Addition and multiplication of two functions are performed similarly to addition and multiplication of tensors in the TT format. For addition, the cores of $g(x) = f(x) + h(x)$ are

$$\mathcal{G}_1(x) = [\mathcal{F}_1(x) \quad \mathcal{H}_1(x)], \quad \mathcal{G}_k(x) = \begin{bmatrix} \mathcal{F}_k(x) & \mathbf{0} \\ \mathbf{0} & \mathcal{H}_k(x) \end{bmatrix}, \quad \mathcal{G}_d(x) = \begin{bmatrix} \mathcal{F}_d(x) \\ \mathcal{H}_d(x) \end{bmatrix}, \quad (34)$$

for $k = 2, \dots, d$. For multiplication, $g(x) = f(x)h(x)$, we have

$$\mathcal{G}_k(x) = \mathcal{F}_k(x) \otimes \mathcal{H}_k(x) \quad \text{for } k = 1, \dots, d. \quad (35)$$

For both of these operations, the continuous functional decomposition has an important advantage compared to operations based on the discretized representation. Primarily, the advantage comes from the ability to add functions of differing discretization levels, e.g., functions represented with bases of different orders. In the discrete case, one can only add functions with identical discretizations.

The continuous nature of the FT also allows us to perform differentiation, as necessary to the implementation of the Laplacian, by differentiating scalar-valued functions that make up the corresponding core. For example, consider the partial derivative of a d -dimensional function f

$$\frac{\partial f}{\partial x_k} = \mathcal{F}_1 \dots \mathcal{F}_{k-1} \begin{bmatrix} \frac{df_k^{(11)}}{dx_k} & \dots & \frac{df_k^{(1r_k)}}{dx_k} \\ \vdots & \ddots & \vdots \\ \frac{df_k^{(r_{k-1}1)}}{dx_k} & \dots & \frac{df_k^{(r_{k-1}r_k)}}{dx_k} \end{bmatrix} \mathcal{F}_{k+1} \dots \mathcal{F}_d. \quad (36)$$

When the univariate functions are expressed in, for example, a basis of orthonormal polyno-

mials, then this operation is unique, well-defined, and computationally inexpensive.

Integration is widely used in Section 6 to compute expectation values. Integrating the multivariate functions scales linearly with dimensionality since it requires integrating over the one-dimensional functions in each core and then performing matrix-vector multiplication $d - 1$ times as follows

$$\begin{aligned}\int f(x)dx &= \int \mathcal{F}_1(x_1)\mathcal{F}_2(x_2)\dots\mathcal{F}_d(x_d)dx_1\dots dx_d \\ &= \left(\int \mathcal{F}_1(x_1)dx_1\right)\left(\int \mathcal{F}_2(x_2)dx_2\right)\dots\left(\int \mathcal{F}_d(x_d)dx_d\right) \\ &= \mathbf{\Gamma}_1\mathbf{\Gamma}_2\dots\mathbf{\Gamma}_d\end{aligned}\tag{37}$$

where $\mathbf{\Gamma}_k = \int \mathcal{F}_k(x_k)dx_k$ contains entries $\mathbf{\Gamma}_k[i, j] = \int f_k^{(ij)}(x_k)dx_k$ and the integral stands for an integral over an arbitrary domain. Furthermore, since each of the univariate functions is typically represented on a known basis, the integral is well defined, unique, and computationally inexpensive to obtain.

The inner product between two functions is another important operation essential for quantum dynamics simulations and computations of correlation functions. Naively, the inner product can be implemented by first computing the product $g(x) = f(x)h(x)$ and then integrating $g(x)$, requiring $\mathcal{O}(dr^4)$ operations. However, this operation can be made more efficient by combining the operations needed for integration and multiplication. For example, Algorithm 1 uses an efficient computation of $v^T(\mathbf{A} \otimes \mathbf{B})$ to perform the inner product in $\mathcal{O}(dr^3)$ where v is a vector and \mathbf{A} and \mathbf{B} are matrices.

Algorithm 1 ft-inner: Inner product between two functions in FT format

Require: Functions f with ranks $r_k^{(f)}$ and g with ranks $r_k^{(g)}$ in FT format

Ensure: $y = \int f(x)g(x)dx$

```

1:  $\mathbf{y} = \int \mathcal{G}_1(x_1) \otimes \mathcal{F}_1(x_1) dx_1$ 
2: for  $k = 2$  to  $d$  do
3:    $\mathbf{Y} = \text{reshape}(\mathbf{y}, r_{k-1}^{(f)}, r_{k-1}^{(g)})$ 
4:    $\mathcal{T} = \mathcal{F}_k(x)^T \mathbf{Y}$ 
5:    $\mathcal{A} = \mathcal{G}_k(x)^T \mathcal{T}^T(x)$ 
6:    $\mathbf{Y} = \int \mathcal{A}(x_k) dx_k$ 
7:    $\mathbf{y} = \text{reshape}(\mathbf{Y}, 1, r_k^{(f)} r_k^{(g)})$ 
8: end for
9:  $y = \mathbf{y}[1]$ 

```

Furthermore, once in FT format, many other familiar operators may be applied to a function with relative ease. Consider the Laplacian $\Delta f(x) = g(x) = \sum_{k=1}^d \frac{\partial^2 f(x)}{\partial x_k^2}$, necessary for implementation of the kinetic energy operator without having to rely on the Fourier transform. Written in this form, one can consider the Laplacian as the summation of d functions $g_k(x)$ in function-train format, where

$$g_k(x) = \frac{\partial^2 f(x)}{\partial x_k^2}. \quad (38)$$

The second derivative is implemented core-by-core in the space of univariate functions. The second derivatives of univariate functions are computed only once, which exploits the benefits of the continuous representation and avoids the need for explicit calculation of the Fourier transform required by grid-based methods.

4 Functional Tensor-Train Chebyshev Propagation

Wavepackets and operators are efficiently represented in terms of low-rank functional tensor trains (FT) or discrete tensor trains (TT). The decompositions are constructed analytically or interpolated with the cross approximation as implemented in the Compressed Continuous Computation (C^3) library⁸⁴ in terms of linear element expansions or the TT-Toolbox.⁸⁵

Operations are computed in the position-space representation, including the kinetic energy operator in the FT representation, which is computed analytically from the Laplacian. In the discrete TT representation, the kinetic energy operator is computed numerically in momentum space. Algebraic manipulations are followed by rounding schemes to avoid an artificial growth of the rank.

The functional tensor-train algebra discussed in Section 3 is then employed to express the individual Chebyshev polynomials of the Hamiltonian as applied to the initial wavepacket, as discussed in Section 2. The codes are available in public domain.⁸⁶ The minimal and maximal potential energy surface values required for rescaling the Hamiltonian in the Chebyshev scheme are determined either analytically or through constrained nonlinear optimization to avoid calculation of the multidimensional potential energy surface at all position-space grid points considered. Individual Chebyshev polynomials are determined as either tensor trains or function trains via the recurrence relation Eq. (6) or the action of the propagator on the wavefunction is determined directly from the Clenshaw algorithm, see Appendix A. The resultant dynamics is analyzed via calculation of survival amplitudes and wavepackets.

5 Chemical Model

We simulate the dynamics of protons in a 50-dimensional model of hydrogen-bonded DNA adenine-thymine base pairs, described by the model potential energy surface^{87,88}

$$V(x_1, x_2, \dots, x_d) = \sum_{i=1}^D \alpha (0.429 x_i - 1.126 x_i^2 - 0.143 x_i^3 + 0.563 x_i^4) + \sum_{i>1}^D \alpha \beta (x_i x_{i-1}) \quad (39)$$

where $\alpha = 0.1$ au determines the energy scaling of the model potential and β is the hydrogen-bond coupling parameter (see Fig. 2). Each x_i describes the coordinate of proton motion in an individual adenine-thymine (A-T) pair as it tautomerizes from the energetically favored amino-keto A-T form to the isomeric imino-enol A*-T* form. The coupling term

parameterized by β provides a model of interaction between base pairs.

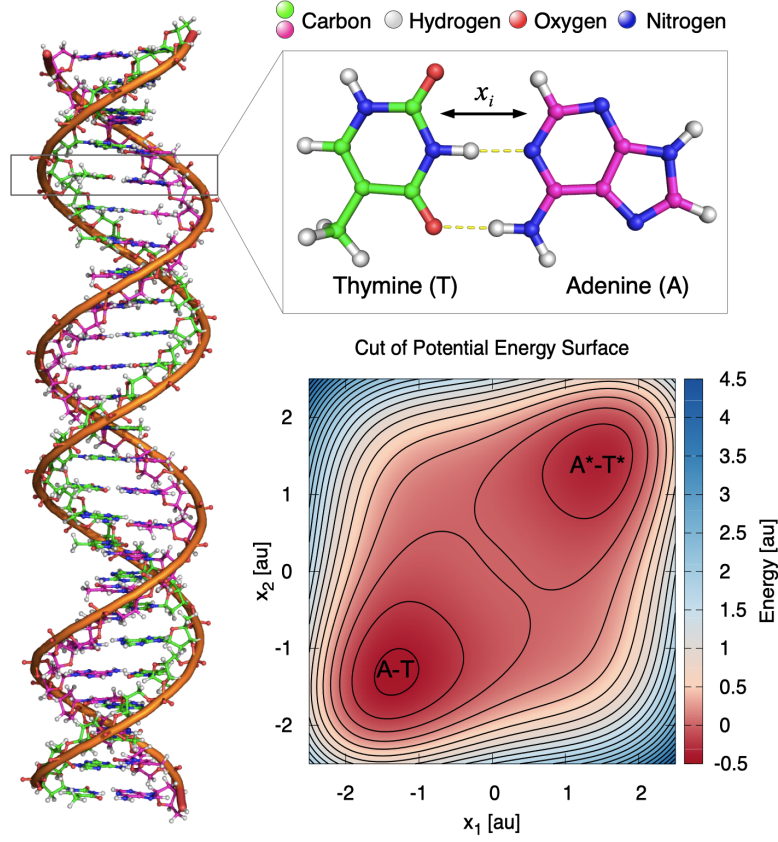


Figure 2: (Left) DNA strand of adenine-thymine base pairs (top right) with a two-dimensional slice of the model potential energy surface Eq. (39) (bottom right).

The resulting 50-dimensional model potential involves strongly anharmonic modes, which are beyond the reach of the grid-based Chebyshev approach or other quantum dynamic methods based on full-grid representations. The molecular system also provides a challenging test case for low-rank tensor-train-based dynamics, as the potential energy surface becomes increasingly demanding as the coupling parameter is dialed up to $\beta = -2$ au. Therefore, the resulting wavepackets can reach maximal ranks of over $r_{\max} = 100$ without truncation in the FT representation.

We examine the ability of the Chebyshev method to simulate isomerization processes by considering the initial state, introduced by Eq. (40), that represents the excited A*-T* tautomer with width $\alpha = 1$ au, position $x_{0,i} = 1$ au, momentum $p_{0,i} = 0$ au, and mass

$m = 1$ au. A position-space vector of grid length $L = 10$ a.u. in the TT format and a position-space region of $x \in [-5, 5]$ au in the FT format (with $N_x = N_p = 2^5$ equal divisions in position space) is used to capture the full extent of the reactive coordinate oscillation between the two isomers. The wavepacket is computed at intermediate times (with a time step of $\tau = 0.01$ au) by defining each intermediate time as an endpoint. A basis set of $N_{\text{poly}} = 50$ polynomial terms is used in the Chebyshev expansion to accurately represent the dynamics in both the TT and FT formats for the DNA system.

6 Results

Figures 3 and 4 show benchmark calculations of FTTC simulations for the 50-dimensional tautomerization of uncoupled DNA base pairs, as compared to the discrete TT implementation and TT-SOFT simulations. The corresponding simulations for coupled DNA base pairs $\beta = -2$ au are compared in Figs. 5 and 6. Comparison slices of the time-dependent wavepacket along two of the 50 dimensions and survival amplitudes show excellent agreement between the methodologies and efficient performance even without relying on high-performance computing facilities.

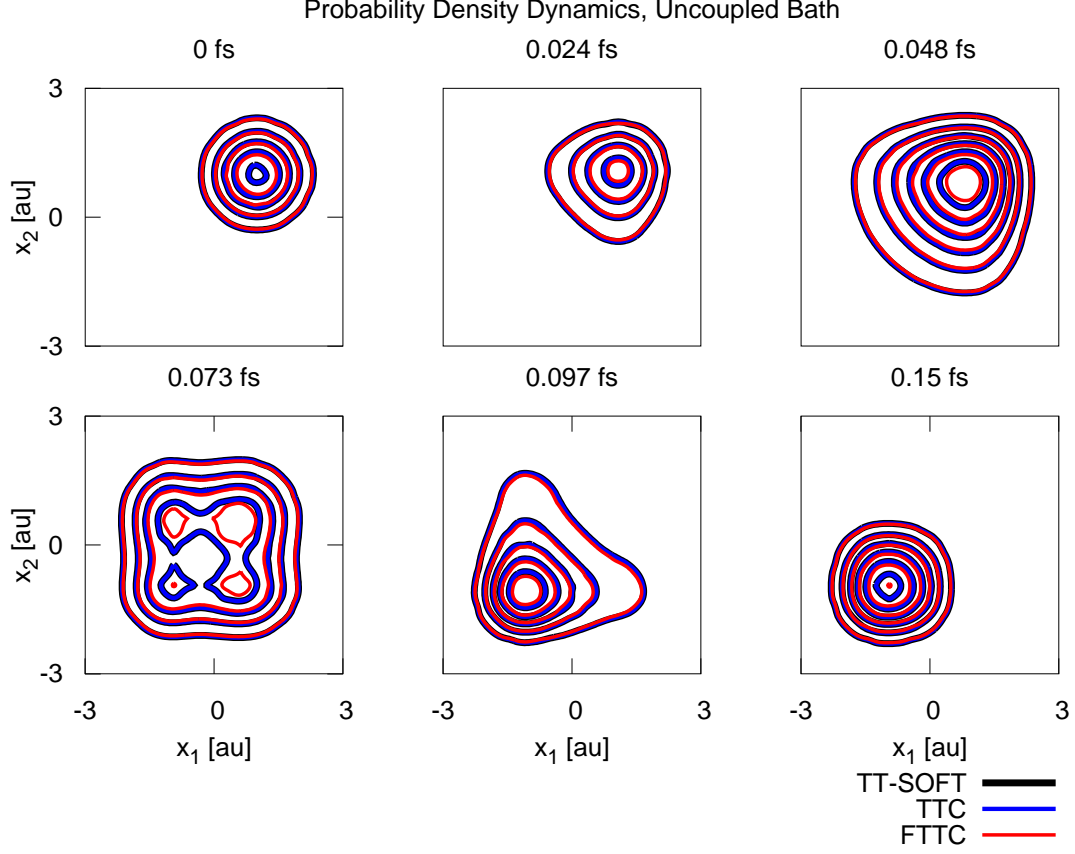


Figure 3: Comparison of two-dimensional slices of the 50-dimensional time-dependent wavepacket obtained from FTTC simulations (red line) and its discrete TT implementation (blue line) as compared to benchmark TT-SOFT (black) simulations of tautomerization quantum dynamics for uncoupled ($\beta = 0$ au) DNA base pairs.

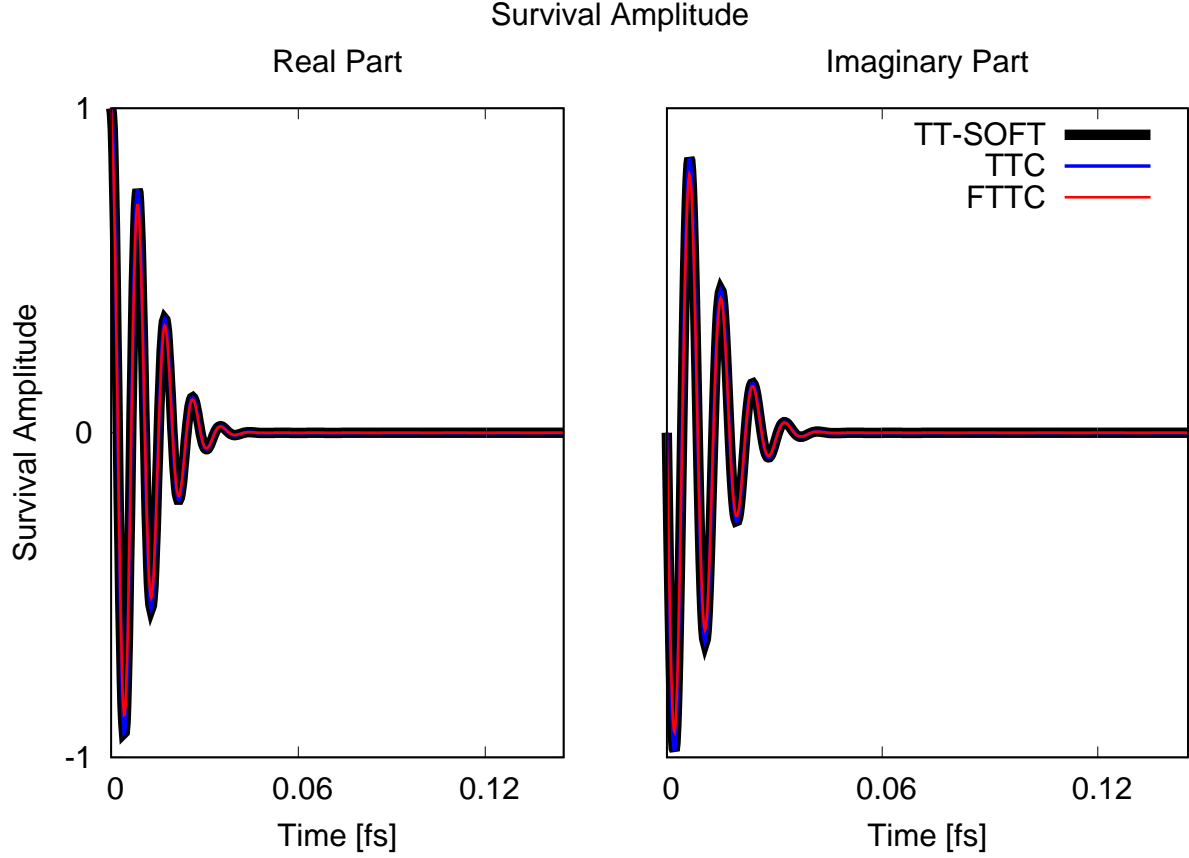


Figure 4: Comparison of survival amplitudes from simulations of the time-dependent wavepacket evolving on a 50-dimensional potential energy surface for the tautomerization dynamics of uncoupled ($\beta = 0$ au) DNA base pairs, including the real (left) and imaginary (right) parts, obtained with FTTC (red line) and its discrete TT implementation (blue line) as compared to benchmark TT-SOFT (black).

The simulations are initialized by a Gaussian,

$$\Psi_0(x) = \prod_{i=1}^D \sqrt{\frac{\alpha}{\pi}} \exp\left(-\frac{\alpha}{2}(x_i - x_{0,i})^2 + ip_{0,i}(x_i - x_{0,i})\right), \quad (40)$$

with $x_{0,i} = 1$ au and $p_{0,i} = 0$ au, corresponding to a displaced tautomeric form along the double-well potential energy surface characterizing the energy change as a function of the proton displacement. The resulting dynamics leads to the motion of the wavepacket into the well of the energetically favored tautomer as the isomerization due to proton dynamics proceeds in the 50-dimensional space of the model system.

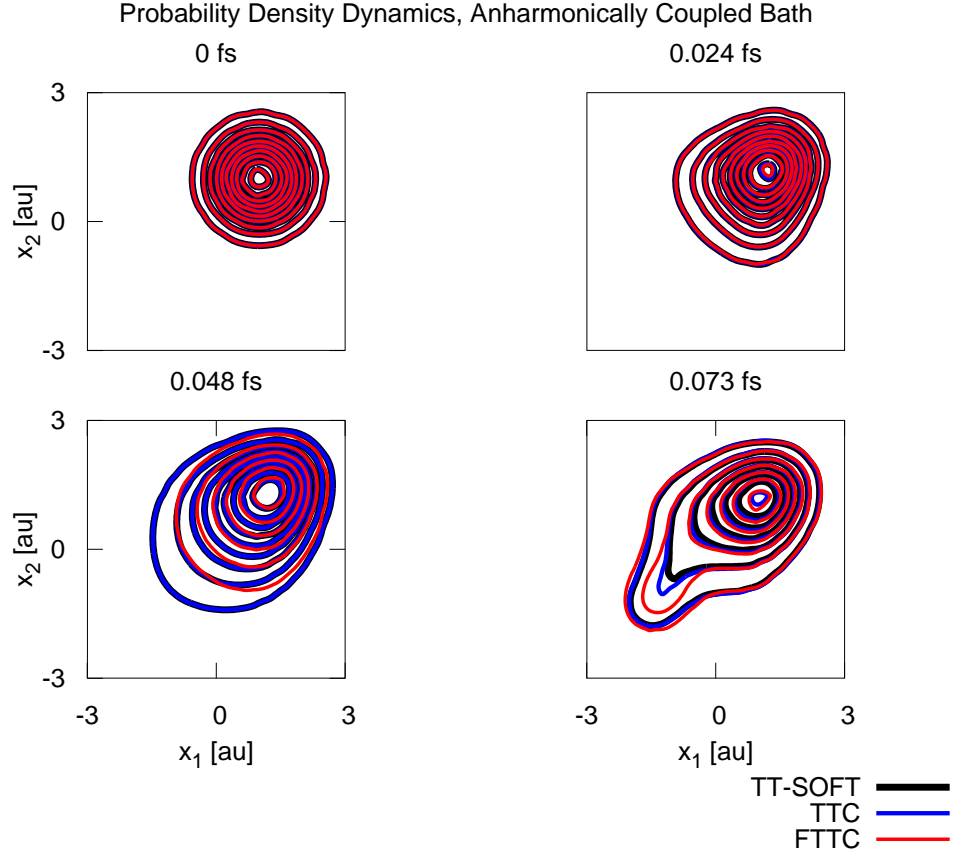


Figure 5: Comparison of two-dimensional slices of the 50-dimensional time-dependent wavepacket obtained from FTTC simulations (red line) and its discrete TT implementation (blue line) as compared to benchmark TT-SOFT (black) simulations of tautomerization quantum dynamics for coupled ($\beta = -2$ au) DNA base pairs.

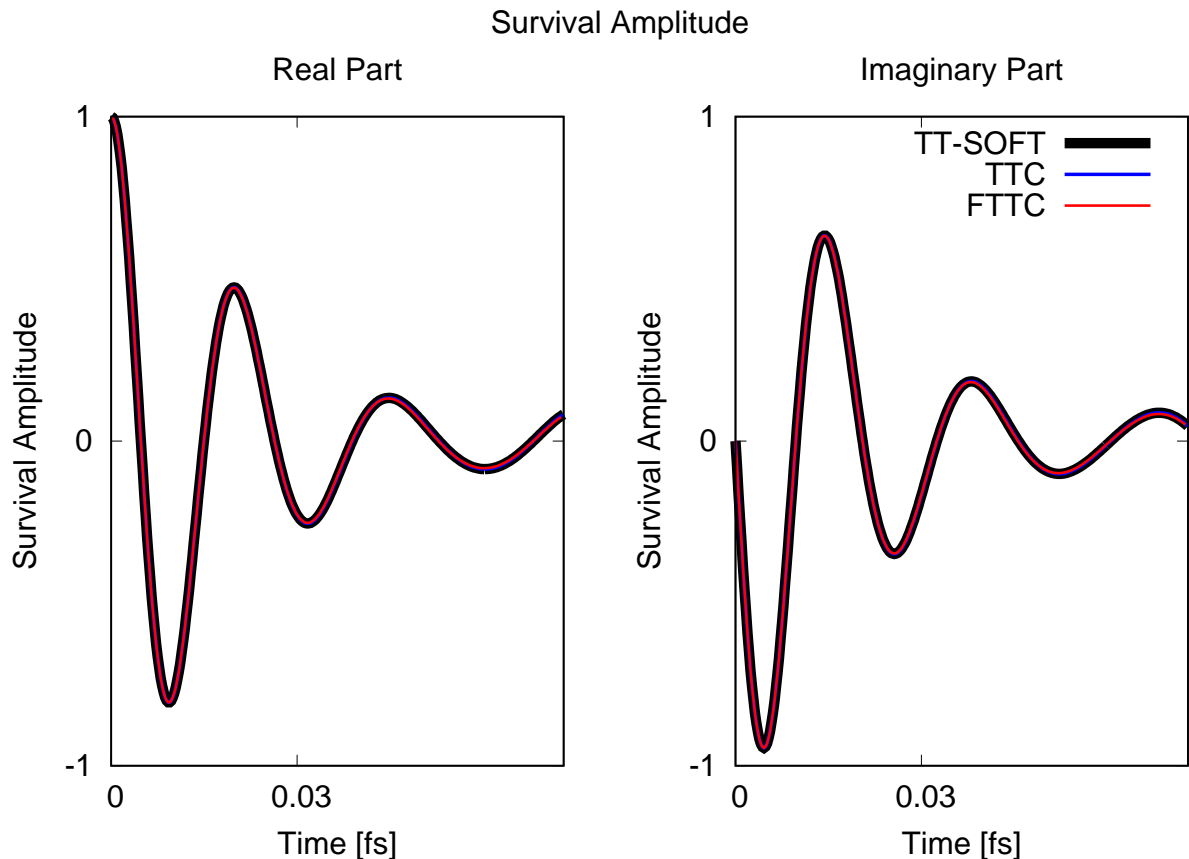


Figure 6: Comparison of survival amplitudes from simulations of the time-dependent wavepacket evolving on a 50-dimensional potential energy surface for the tautomerization dynamics of coupled ($\beta = -2$ au) DNA base pairs, including the real (left) and imaginary (right) parts, obtained with FTTC (red line), its discrete TT implementation (blue line), and benchmark TT-SOFT (black).

7 Discussion

Numerically exact quantum dynamical methods that rely on full-grid representations are not applicable to high-dimensional model systems since they require computational resources that scale exponentially with dimensionality. Even the standard grid-based implementation of the Chebyshev method, renowned for its ability to achieve accuracy within machine precision, has been limited in applications to nuclear quantum dynamics to model systems with no more than four atoms. Here, we have shown how to extend the capabilities of the Cheby-

shev propagation scheme to high-dimensional systems in terms of the FTTC algorithm. We anticipate that the resulting FTTC methodology will be useful not only for simulations of quantum reaction dynamics in general but also as a general method to obtain numerical solutions of linear systems in high dimensionality, typically arising from space discretization in many other types of applications. Furthermore, the functional train decomposition should also find wide applicability in studies requiring computations of gradients, integrals, and correlation functions of systems with high dimensionality.

With regards to the basis functions, we note that the functional tensor-train representation can implement suitable choices of univariate basis functions that could be ideal for data compression in chemistry, for example, waveforms or Gaussian functions, which are common to both wavepacket propagation methods and electronic structure calculations alike. In general, representations that require $O(n^d)$ data points in a d -dimensional grid with n points for each direction would require at most $O(dnr^2)$ data points for a maximum rank r in a discrete tensor-train representation and only $O(dpr^2)$ data points in functional tensor-train format where p is the number of parameters, which represents a significant reduction in computational cost ideal for modeling molecular systems.

8 Acknowledgments

M. B. S. acknowledges financial support from the Yale Quantum Institute Postdoctoral Fellowship. V. S. B. acknowledges support from the NSF Grant no. CHE-1900160 and high-performance computing time from NERSC and the Yale High-Performance Computing Center. A. A. G. was supported by the AFOSR Computational Mathematics Program under the Young Investigator Program.

Appendix

A Clenshaw Algorithm

The direct computation of the Chebyshev expansion Eq. (25) based on the usual summation algorithm has two disadvantages: (1) all summands have to be kept in the memory of the computer, which can be very expensive in practical applications since the tensors $T_k(\hat{\mathcal{H}}_0)\mathcal{W}_0$ (and also their low-rank approximations) are typically large objects, and (2) it is known that the worst-case error generated by the floating point operations grows proportionally to the number N of summands.⁸⁹ We therefore use the Clenshaw algorithm,⁸⁰ which offers a stable alternative to evaluate linear combinations of polynomials that satisfy a linear recurrence relation such as the Chebyshev polynomials.⁷⁵

Assuming that for given coefficients $c_0, c_1, \dots, c_{N-1} \in \mathbb{C}$ we are interested in the value of the partial Chebyshev sum Eq. (13), the Clenshaw algorithm replaces the summation by the evaluation of the following backward recurrence system

$$\begin{cases} B_r(y) = 2yB_{r+1}(y) - B_{r+2}(y) + c_r, & r = N-1, \dots, 0; \\ B_N(y) = 0, & B_{N+1}(y) = 0; \end{cases} \quad (41)$$

and then expresses the partial Chebyshev sum as

$$\sum_{k=0}^{N-1} (2 - \delta_{k,0}) c_k T_k(y) = B_0(y) - B_2(y). \quad (42)$$

To obtain the approximation of the wavefunction $\Psi(t)$, we adapted the Clenshaw algorithm

by first solving the backward recurrence system

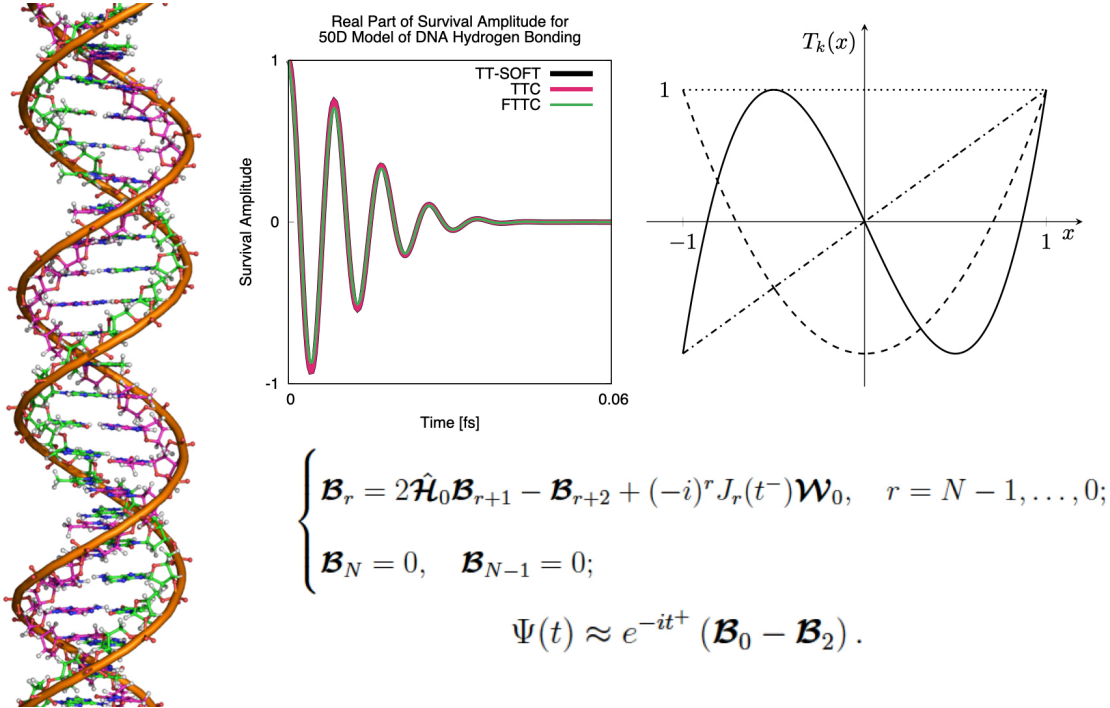
$$\begin{cases} \mathcal{B}_r = 2\hat{\mathcal{H}}_0\mathcal{B}_{r+1} - \mathcal{B}_{r+2} + (-i)^r J_r(t^-)\mathcal{W}_0, & r = N-1, \dots, 0; \\ \mathcal{B}_N = 0, \quad \mathcal{B}_{N-1} = 0; \end{cases} \quad (43)$$

and then computing the approximant

$$\Psi(t) \approx e^{-it^+} (\mathcal{B}_0 - \mathcal{B}_2). \quad (44)$$

We note that this numerically stable procedure needs to keep only three tensors in memory.

B TOC Graphic



References

- (1) Dirac, P. A. M. Note on Exchange Phenomena in the Thomas Atom. *Math. Proc. Cambridge Philos. Soc.* **1930**, *26*, 376–385.
- (2) McLachlan, A. D.; Ball, M. A. Time-Dependent Hartree-Fock Theory for Molecules. *Rev. Mod. Phys.* **1964**, *36*, 844–855.
- (3) Gerber, R. B.; Buch, V.; Ratner, M. A. Time-dependent self-consistent field approximation for intramolecular energy transfer. I. Formulation and application to dissociation of van der Waals molecules. *J. Chem. Phys.* **1982**, *77*, 3022–3030.
- (4) Flores, S. C.; Batista, V. S. Model Study of Coherent-Control of the Femtosecond Primary Event of Vision. *J. Phys. Chem. B* **2004**, *108*, 6745–6749.
- (5) Meyer, H.-D.; Manthe, U.; Cederbaum, L. S. The Multi-Configurational Time-Dependent Hartree Approach. *Chem. Phys. Lett.* **1990**, *165*, 73–78.
- (6) Meyer, H.-D.; Manthe, U.; Cederbaum, L. S. In *Numerical Grid Methods and Their Application to Schrödinger’s Equation*; Cerjan, C., Ed.; Kluwer Academic Publishers, 1993; Chapter The Multi-Configuration Hartree Approach.
- (7) Beck, M. H.; Jäckle, A.; Worth, G. A.; Meyer, H.-D. The multi-configuration time-dependent Hartree (MCTDH) method: a highly efficient algorithm for propagating wavepackets. *Phys. Rep.* **2000**, *324*, 1–105.
- (8) Wang, H.; Thoss, M. Multilayer-formulation of the multiconfiguration time-dependent Hartree theory. *J. Chem. Phys.* **2003**, *119*, 1289–1299.
- (9) Schulze, J.; Shibl, M. F.; Al-Marri, M. J.; Kühn, O. Multi-layer multi-configuration time-dependent Hartree (ML-MCTDH) approach to the correlated exciton-vibrational dynamics in the FMO complex. *J. Chem. Phys.* **2016**, *144*, 185101.

- (10) Burghardt, I.; Meyer, H.-D.; Cederbaum, L. S. Approaches to the approximate treatment of complex molecular systems by the multi-configuration time-dependent Hartree method. *J. Chem. Phys.* **1999**, *111*, 2927–2939.
- (11) Worth, G. A.; Burghardt, I. Full quantum mechanical molecular dynamics using Gaussian wavepackets. *Chem. Phys. Lett.* **2003**, *368*, 502–508.
- (12) Kosloff, R. Propagation Methods for Quantum Molecular Dynamics. *Annu. Rev. Phys. Chem.* **1994**, *45*, 145–178.
- (13) Feit, M. D.; Fleck Jr., J. A.; Steiger, A. Solution of the Schrödinger equation by a spectral method. *J. Comput. Phys.* **1982**, *47*, 412–433.
- (14) Gray, S. K.; Verosky, J. M. Classical Hamiltonian structures in wave packet dynamics. *J. Chem. Phys.* **1994**, *100*, 5011–5022.
- (15) Takahashi, K.; Ikeda, K. Applicability of symplectic integrator to classically unstable quantum dynamics. *J. Chem. Phys.* **1993**, *99*, 8680–8694.
- (16) Mazur, J.; Rubin, R. J. Quantum-Mechanical Calculation of the Probability of an Exchange Reaction for Constrained Linear Encounters. *J. Chem. Phys.* **1959**, *31*, 1395–1412.
- (17) McCullough, E. A. Jr. and Robert E. Wyatt, Quantum Dynamics of the Collinear (H, H₂) Reaction. *J. Chem. Phys.* **1969**, *51*, 1253–1254.
- (18) McCullough, E. A. Jr. and Robert E. Wyatt, Dynamics of the Collinear H+H₂ Reaction. I. Probability Density and Flux. *J. Chem. Phys.* **1971**, *54*, 3578–3591.
- (19) Lanczos, C. An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators. *J. Res. Natl. Bur. Stand.* **1950**, *45*, 255–282.
- (20) Leforestier, C.; Bisseling, R. H.; Cerjan, C.; Feit, M. D.; Friesner, R.; Guldberg, A.; Hammerlich, A.; Jolicard, G.; Karrlein, W.; Meyer, H.-D.; Lipkin, N.; Roncero, O.;

- Kosloff, R. A Comparison of Different Propagation Schemes for the Time Dependent Schrödinger Equation. *J. Comput. Phys.* **1991**, *94*, 59–80.
- (21) Kong, X.; Markmann, A.; Batista, V. S. Time-Sliced Thawed Gaussian Propagation Method for Simulations of Quantum Dynamics. *J. Phys. Chem. A* **2016**, *120*, 3260–3269.
- (22) Wu, Y.; Batista, V. S. Matching-pursuit for simulations of quantum processes. *J. Chem. Phys.* **2003**, *118*, 6720–6724.
- (23) Wu, Y.; Batista, V. S. Quantum tunneling dynamics in multidimensional systems: A matching-pursuit description. *J. Chem. Phys.* **2004**, *121*, 1676–1680.
- (24) Chen, X.; Batista, V. S. Matching-pursuit/split-operator-Fourier-transform simulations of excited-state nonadiabatic quantum dynamics in pyrazine. *J. Chem. Phys.* **2006**, *125*, 124313.
- (25) Tal-Ezer, H.; Kosloff, R. An accurate and efficient scheme for propagating the time dependent Schrödinger equation. *J. Chem. Phys.* **1984**, *81*, 3967–3971.
- (26) Ndong, M.; Tal-Ezer, H.; Kosloff, R.; Koch, C. P. A Chebychev propagator with iterative time ordering for explicitly time-dependent Hamiltonians. *J. Chem. Phys.* **2010**, *132*, 064105.
- (27) Schaefer, I.; Tal-Ezer, H.; Kosloff, R. Semi-global approach for propagation of the time-dependent Schrödinger equation for time-dependent and nonlinear problems. *J. Comput. Phys.* **2017**, *343*, 368–413.
- (28) Goldfield, E. M.; Gray, S. K. A quantum dynamics study of $\text{H}_2 + \text{OH} \rightarrow \text{H}_2\text{O} + \text{H}$ employing the Wu-Schatz-Lendvay-Fang-Harding potential function and a four-atom implementation of the real wave packet method. *J. Chem. Phys.* **2002**, *117*, 1604–1613.

- (29) Cvitaš, M. T.; Althorpe, S. C. A Chebyshev method for state-to-state reactive scattering using reactant-product decoupling: $\text{OH} + \text{H}_2 \rightarrow \text{H}_2\text{O} + \text{H}$. *J. Chem. Phys.* **2013**, *139*, 064307.
- (30) Löwdin, P.-O. Proton Tunneling in DNA and its Biological Implications. *Rev. Mod. Phys.* **1963**, *35*, 724–732.
- (31) Khoromskij, B. N. $O(d\log N)$ -Quantics Approximation of $N - d$ Tensors in High-Dimensional Numerical Modeling. *Constr. Approximation* **2011**, *34*, 257–280.
- (32) Townsend, A.; Over, S. The automatic solution of partial differential equations using a global spectral method. *J. Comput. Phys.* **2015**, *299*, 106–123.
- (33) Dolgov, S.; Kressner, D.; Strössner, C. Functional Tucker approximation using Chebyshev interpolation. *SIAM J. Sci. Comput.* **2021**, *43*, A2190–A2210.
- (34) Halimeh, J. C.; Kolley, F.; McCulloch, I. P. Chebyshev matrix product state approach for time evolution. *Phys. Rev. B* **2015**, *92*, 115130.
- (35) Buser, M.; Cerrillo, J.; Schaller, G.; Cao, J. Initial system-environment correlations via the transfer-tensor method. *Phys. Rev. A* **2017**, *96*, 062122.
- (36) Kananenka, A. A.; Hsieh, C.-Y.; Cao, J.; Geva, E. Accurate Long-Time Mixed Quantum-Classical Liouville Dynamics via the Transfer Tensor Method. *J. Phys. Chem. Lett.* **2016**, *7*, 4809–4814.
- (37) Chenu, A.; Cao, J. Construction of Multichromophoric Spectra from Monomer Data: Applications to Resonant Energy Transfer. *Phys. Rev. Lett.* **2017**, *118*, 013001.
- (38) Wang, H.; Thoss, M. Multilayer formulation of the multiconfiguration time-dependent Hartree theory. *J. Chem. Phys.* **2003**, *119*, 1289–1299.

- (39) Binder, R.; Wahl, J.; Romer, S.; Burghardt, I. Coherent exciton transport driven by torsional dynamics: a quantum dynamical study of phenylene-vinylene type conjugated systems. *Faraday Discuss.* **2013**, *163*, 205–222.
- (40) Greene, S. M.; Batista, V. S. Tensor-Train Split-Operator Fourier Transform (TT-SOFT) Method: Multidimensional Nonadiabatic Quantum Dynamics. *J. Chem. Theory Comput.* **2017**, *13*, 4034–4042.
- (41) Haegeman, J.; Dirac, J. I.; Osborne, T. J.; Pižorn, I.; Verschelde, H.; Verstraete, F. Time-Dependent Variational Principle for Quantum Lattices. *Phys. Rev. Lett.* **2011**, *107*, 070601.
- (42) Lubich, C.; Rohwedder, T.; Schneider, R.; Vandereycken, B. Dynamical approximation by hierarchical Tucker and tensor-train tensors. *SIAM J. Matrix Anal. Appl.* **2013**, *34*, 470–494.
- (43) Lubich, C.; Oseledets, I. V.; Vandereycken, B. Time Integration of Tensor Trains. *SIAM J. Numer. Anal.* **2015**, *53*, 917–941.
- (44) Haegeman, J.; Lubich, C.; Oseledets, I.; Vandereycken, B.; Verstraete, F. Unifying time evolution and optimization with matrix product states. *Phys. Rev. B* **2016**, *94*, 165116.
- (45) Frahm, L.-H.; Pfannkuche, D. Ultrafast ab Initio Quantum Chemistry Using Matrix Product States. *J. Chem. Theory Comput.* **2019**, *15*, 2154–2165.
- (46) Cazalilla, M. A.; Marston, J. B. Time-Dependent Density-Matrix Renormalization Group: A Systematic Method for the Study of Quantum Many-Body Out-of-Equilibrium Systems. *Phys. Rev. Lett.* **2002**, *88*, 256403.
- (47) Vidal, G. Efficient Classical Simulation of Slightly Entangled Quantum Computations. *Phys. Rev. Lett.* **2003**, *91*, 147902.

- (48) Luo, H. G.; Xiang, T.; Wang, X. Q. Comment on “Time-Dependent Density-Matrix Renormalization Group: A Systematic Method for the Study of Quantum Many-Body Out-of-Equilibrium Systems”. *Phys. Rev. Lett.* **2003**, *91*, 049701.
- (49) White, S. R.; Feiguin, A. E. Real-Time Evolution Using the Density Matrix Renormalization Group. *Phys. Rev. Lett.* **2004**, *93*, 076401.
- (50) Vidal, G. Efficient Simulation of One-Dimensional Quantum Many-Body Systems. *Phys. Rev. Lett.* **2004**, *93*, 040502.
- (51) Verstraete, F.; García-Ripoll, J. J.; Cirac, J. I. Matrix Product Density Operators: Simulation of Finite-Temperature and Dissipative Systems. *Phys. Rev. Lett.* **2004**, *93*, 207204.
- (52) Feiguin, A. E.; White, S. R. Time-step targeting methods for real-time dynamics using the density matrix renormalization group. *Phys. Rev. B* **2005**, *72*, 020404(R).
- (53) García-Ripoll, J. J. Time evolution of Matrix Product States. *New J. Phys.* **2006**, *8*, 305.
- (54) Baiardi, A.; Reiher, M. Large-Scale Quantum Dynamics with Matrix Product States. *J. Chem. Theory Comput.* **2019**, *15*, 3481–3498.
- (55) Paeckel, S.; Köhler, T.; Swoboda, A.; Manmana, S. R.; Schollwöck, U.; Hubig, C. Time-evolution methods for matrix-product states. *Ann. Phys.* **2019**, *411*, 167998.
- (56) Chertkov, A.; Oseledets, I. Solution of the Fokker-Planck Equation By Cross Approximation Method In The Tensor Train Format. **2021**, arXiv:2102.08143. arXiv.org e-Print archive. <https://arxiv.org/abs/2102.08143>.
- (57) Niklasson, A. M. N.; Mniszewski, S. M.; Negre, C. F. A.; Cawkwell, M. J.; Swart, P. J.; Mohd-Yusof, J.; Germann, T. C.; Wall, M. E.; Bock, N.; Rubensson, E. H.; Djid-

- jev, H. Graph-based linear scaling electronic structure theory. *J. Chem. Phys.* **2016**, *144*, 234101.
- (58) Chan, G. K.-L.; Keselman, A.; Nakatani, N.; Li, Z.; White, S. R. Matrix product operators, matrix product states, and ab initio density matrix renormalization group algorithms. *J. Chem. Phys.* **2016**, *145*, 014102.
- (59) Chan, G. K.-L.; Kállay, M.; Gauss, J. State-of-the-art density matrix renormalization group and coupled cluster theory studies of the nitrogen binding curve. *J. Chem. Phys.* **2004**, *121*, 6110–6116.
- (60) Roemelt, M.; Guo, S.; Chan, G. K.-L. A projected approximation to strongly contracted N-electron valence perturbation theory for DMRG wavefunctions. *J. Chem. Phys.* **2016**, *144*, 204113.
- (61) Sharma, P.; Bernales, V.; Knecht, S.; Truhlar, D. G.; Gagliardi, L. Density matrix renormalization group pair-density functional theory (DMRG-PDFT): singlet-triplet gaps in polyacenes and polyacetylenes. *Chem. Sci.* **2019**, *10*, 1716–1723.
- (62) Zhou, C.; Gagliardi, L.; Truhlar, D. G. Multiconfiguration Pair-Density Functional Theory for Iron Porphyrin with CAS, RAS, and DMRG Active Spaces. *J. Phys. Chem. A* **2019**, *123*, 3389–3394.
- (63) Wang, X.-G.; Carrington, T. A contracted basis-Lanczos calculation of vibrational levels of methane: Solving the Schrödinger equation in nine dimensions. *J. Chem. Phys.* **2003**, *119*, 101–117.
- (64) Bramley, M. J.; Tromp, J. W.; Carrington, T.; Corey, G. C. Efficient calculation of highly excited vibrational energy levels of floppy molecules: The band origins of H_3^+ up to 35000 cm^{-1} . *J. Chem. Phys.* **1994**, *100*, 6175–6194.

- (65) Bramley, M. J.; Carrington, T. Calculation of triatomic vibrational eigenstates: Product or contracted basis sets, Lanczos or conventional eigensolvers? What is the most efficient combination? *J. Chem. Phys.* **1994**, *101*, 8494–8507.
- (66) Gorodetsky, A. A.; Karaman, S.; Marzouk, Y. M. A continuous analogue of the tensor-train decomposition. *Comput. Methods Appl. Mech. Eng.* **2019**, *347*, 59–84.
- (67) Oseledets, I. V. Constructive representation of functions in low-rank tensor formats. *Constr. Approx.* **2013**, *37*, 1–18.
- (68) Gorodetsky, A. A. Continuous Low-rank Tensor Decompositions, with Applications to Stochastic Optimal Control and Data Assimilation. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 2017.
- (69) Gorodetsky, A.; Jakeman, J. D. Gradient-based Optimization for Regression in the Functional Tensor-Train Format. *J. Comput. Phys.* **2018**, *374*, 1219–1238.
- (70) Rego, L. G.; Santos, L. F.; Batista, V. S. Coherent control of quantum dynamics with sequences of unitary phase-kick pulses. *Annu. Rev. Phys. Chem.* **2009**, *60*, 293–320.
- (71) Oseledets, I.; Tyrtysnikov, E. T. TT-cross approximation for multidimensional arrays. *Linear Algebra Appl.* **2010**, *432*, 70–88.
- (72) Oseledets, I. V. Tensor-Train Decomposition. *SIAM J. Sci. Comput.* **2011**, *33*, 2295–2317.
- (73) Östlund, S.; Rommer, S. Thermodynamic Limit of Density Matrix Renormalization. *Phys. Rev. Lett.* **1995**, *75*, 3537–3540.
- (74) Trefethen, L. N. *Approximation Theory and Approximation Practice*; Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2013.
- (75) Fox, L.; Parker, I. B. *Chebyshev polynomials in numerical analysis*; Oxford University Press, London-New York-Toronto, Ont., 1968.

- (76) Bergold, P.; Lasser, C. Fourier Series Windowed by a Bump Function. *J. Anal. Appl.* **2020**, *26*, 65.
- (77) Tal-Ezer, H. Polynomial approximation of functions of matrices and applications. *J. Sci. Comput.* **1989**, *4*, 25–60.
- (78) Abramowitz, M.; Stegun, I. A. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*; National Bureau of Standards Applied Mathematics Series; 1964; Vol. 55.
- (79) Lubich, C. *From Quantum to Classical Molecular Dynamics: Reduced Models and Numerical Analysis*; Zurich Lectures in Advanced Mathematics; European Mathematical Society (EMS), Zürich, 2008.
- (80) Clenshaw, C. W. A note on the summation of Chebyshev series. *Math. Comput.* **1955**, *9*, 118–120.
- (81) T. A. Driscoll, N. Hale, and L. N. Trefethen, editors, *Chebfun Guide*. Pafnuty Publications, Oxford, 2014.
- (82) Mathelin, L. Quantification of uncertainty from high-dimensional scattered data via polynomial approximation. *Int. J. Uncertainty Quantif.* **2014**, *4*, 243–271.
- (83) Chevreuil, M.; Lebrun, R.; Nouy, A.; Rai, P. A least-squares method for sparse low rank approximation of multivariate functions. *SIAM/ASA J. Uncertainty Quantif.* **2015**, *3*, 897–921.
- (84) Gorodetsky, A. A. Compressed Continuous Computation (C3) Library. <https://github.com/goroda/Compressed-Continuous-Computation>.
- (85) Oseledets, I. oseledets/TT-Toolbox. 2020; <https://www.github.com/oseledets/TT-Toolbox>.

- (86) Soley, M. B.; Bergold, P.; Gorodetsky, A.; Batista, V. S. Functional Tensor-Train Chebyshev Method for Multidimensional Quantum Dynamics Simulations. <https://github.com/michelinesoley/FTTC>.
- (87) Godbeer, A. D.; Al-Khalili, J. S.; Stevenson, P. D. Modelling proton tunneling in the adenine-thymine base pair. *Phys. Chem. Chem. Phys.* **2015**, *17*, 13034.
- (88) Soley, M. B.; Bergold, P.; Batista, V. S. Iterative Power Algorithm for Global Optimization with Quantics Tensor Trains. *J. Chem. Theory Comput.* **2021**, *17*, 3280–3291.
- (89) Higham, N. J. *Accuracy and Stability of Numerical Algorithms*, 2nd ed.; Society for Industrial and Applied Mathematics, 2002.

Supporting Information: Functional Tensor-Train Chebyshev Method for Multidimensional Quantum Dynamics Simulations

Micheline B. Soley,^{†,‡} Paul Bergold,[¶] Alex A. Gorodetsky,[§] and Victor S.
Batista^{*,†,‡,||}

[†] *Yale Quantum Institute, Yale University,
P.O. Box 208334, New Haven, CT, 06520-8263, USA*

[‡] *Department of Chemistry, Yale University,
P.O. Box 208107, New Haven, CT, 06520, USA*

[¶] *Zentrum Mathematik, Technical University of Munich,
Boltzmannstr. 3, 85748 Garching, Germany*

[§] *Department of Aerospace Engineering, University of Michigan,
1320 Beal Avenue Ann Arbor, MI 48109-2140, USA*

^{||} *Energy Sciences Institute, Yale University,
P.O. Box 27394, West Haven, CT, 06516-7394, USA*

E-mail: victor.batista@yale.edu

1 Computational Performance

The favorable computational performance of the tensor-train Chebyshev (TTC) method is demonstrated through analysis of a system whose time evolution is known exactly, a Gaussian coherent state in the harmonic oscillator potential

$$V(x) = \sum_{i=1}^D \frac{1}{2} m \omega^2 x_i^2 \quad (1)$$

of dimension $D = 2$, mass $m = 1$ au, and frequency $\omega = 1$ au.

The benchmark analytic time evolution of the Gaussian state is determined recognizing that a coherent state $|\alpha\rangle$ can be expanded in terms of the harmonic oscillator eigenstates $|n\rangle$ of energy $E_n = (n + 1/2)\hbar\omega$ as follows

$$|\alpha\rangle = e^{-\frac{1}{2}|\alpha|^2} \sum_{n=0}^{\infty} \frac{\alpha^n}{\sqrt{n!}} |n\rangle \quad (2)$$

where α is a complex number that indicates the displacement of the coherent state from the origin in phase space, such that the coherent state is an eigenstate of the annihilation operator

$$\hat{a} |\alpha\rangle = \alpha |\alpha\rangle \quad (3)$$

where

$$\hat{a} |n\rangle = \sqrt{n} |n-1\rangle \quad (4)$$

and an eigenstate of the time-evolved annihilation operator in the Heisenberg picture

$$\hat{a}(t) |\alpha\rangle = e^{\frac{i}{\hbar} \hat{H} t} \hat{a} e^{-\frac{i}{\hbar} \hat{H} t} |\alpha\rangle \quad (5)$$

$$= \alpha e^{-\frac{1}{2} |\alpha|^2} \sum_{n=1}^{\infty} \frac{\alpha^{n-1}}{\sqrt{n-1}!} e^{-\frac{i}{\hbar} (E_n - E_{n-1}) t} |n-1\rangle \quad (6)$$

$$= \alpha e^{-i\omega t} e^{-\frac{1}{2} |\alpha|^2} \sum_{n=0}^{\infty} \frac{\alpha^n}{\sqrt{n}!} |n\rangle \quad (7)$$

$$= \alpha e^{-i\omega t} |\alpha\rangle \quad (8)$$

The time evolution of the coherent state in the Schrödinger picture is then determined by the action of the annihilation operator

$$\hat{a} |\alpha(t)\rangle = \alpha(t) |\alpha(t)\rangle \quad (9)$$

with the time-dependent displacement of the coherent state

$$\alpha(t) = e^{-i\omega t} \alpha(0) \quad (10)$$

which yields the position-space representation of the time-evolved coherent state as a coherent state of the form

$$\Psi(x, t) = \langle x | \alpha \rangle = \left(\frac{\omega}{\pi \hbar} \right)^{1/4} \exp \left(-\frac{1}{2} |\alpha(0)|^2 + \frac{\omega}{2\hbar} x^2 - \left(\sqrt{\frac{\omega}{\hbar}} x - \frac{e^{-i\omega t} \alpha(0)}{\sqrt{2}} \right)^2 \right) \quad (11)$$

We find the wavepacket determined by tensor-train Chebyshev (TTC) quantum dynamics is significantly more accurate than the short-time Tensor-Train Split Operator Fourier Transform (TT-SOFT) approach for long time steps, as shown in Fig. 1. Whereas the error of the L^2 -norm of the wavefunction is nearly equivalent for both methods for short time steps, TTC produces the wavepacket with several orders of magnitude lower L^2 -norm error for time steps on the order of 100 to 1000 times longer than those required by TT-SOFT

given a sufficient number of Chebyshev polynomials in the expansion of the propagator. This suggests TTC can require fewer costly Fourier transform computations than TT-SOFT for long time steps.

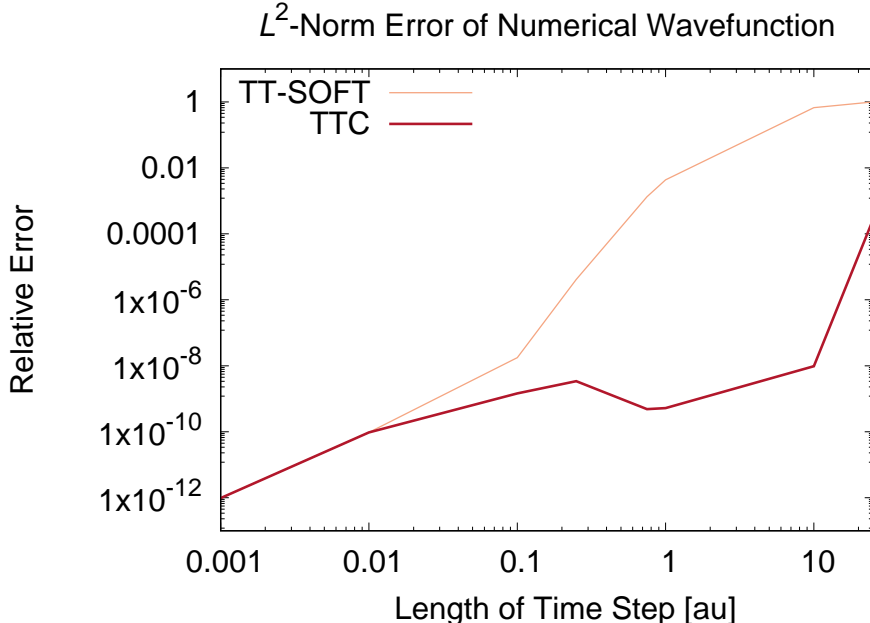


Figure 1: The L^2 -norm error of the numerical wavefunction is significantly lower for TTC (red line, $N = 750$ Chebyshev polynomials) than TT-SOFT (orange line) for long time steps.

Likewise, TTC molecular dynamics is found to be more accurate relative to TT-SOFT for computation of the autocorrelation function over long time steps (see Fig. 2). For short time steps, TTC and TT-SOFT closely agree, and TTC reduces the relative error of the autocorrelation function by orders of magnitude at the longest time steps considered. We find TTC yields the autocorrelation function with only one percent error with a time step 100 times longer than the maximum time step that can be used to accurately simulate the dynamics with TT-SOFT. The TTC molecular dynamics method therefore successfully maintains accuracy and, where desired, avoids calculation of the wavefunction at intermediate time steps for calculation of the autocorrelation function at the final time.

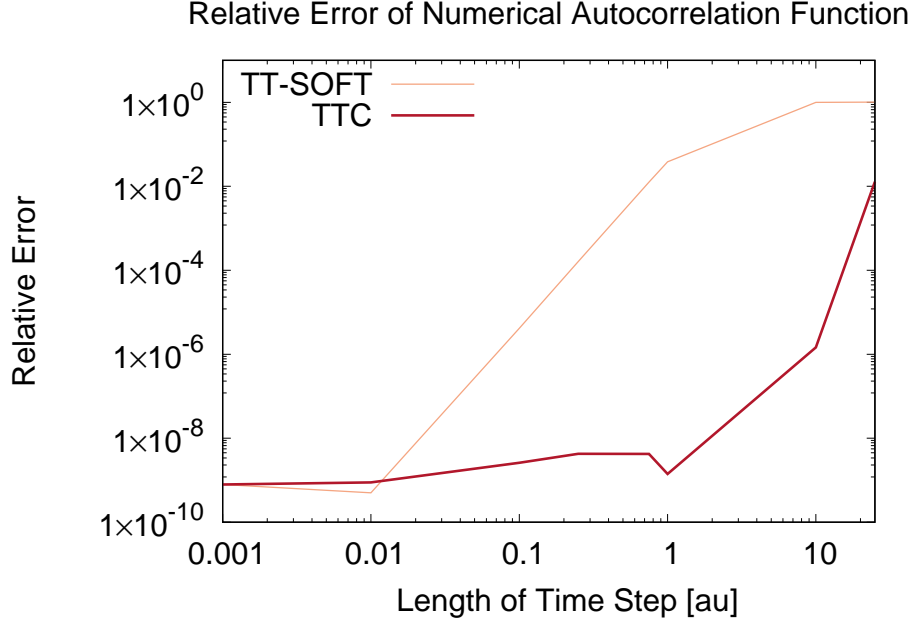


Figure 2: TTC (red line, $N = 750$ Chebyshev polynomials) significantly reduces the relative error associated with determination of the autocorrelation function of the benchmark system relative to the state-of-the-art short-time TT-SOFT method (orange line) for long time steps.

Examination of the relative error of the L^2 -norm of the wavefunction as a function of the number of terms in the Chebyshev expansion confirms the ability of the TTC algorithm to achieve high accuracy solutions for long time steps given a sufficient number of Chebyshev polynomials, as shown in Fig. 3. The TTC approach requires fewer than 200 Chebyshev polynomials to accurately determine the propagator for a final time of 1 au, such that the method is efficient for short-time propagation, which is beneficial for computation of the wavefunction at intermediate time steps. As expected, the number of polynomials required for accurate simulation of the benchmark system's dynamics increases for direct propagation of the initial wavepacket for larger final times. The expansion is shown to converge for final times up to 6 au for fewer than 500 polynomials, which demonstrates the robustness of TTC for simulation of long-time dynamics in agreement with the measure of Chebyshev accuracy previously observed in the one-dimensional case.

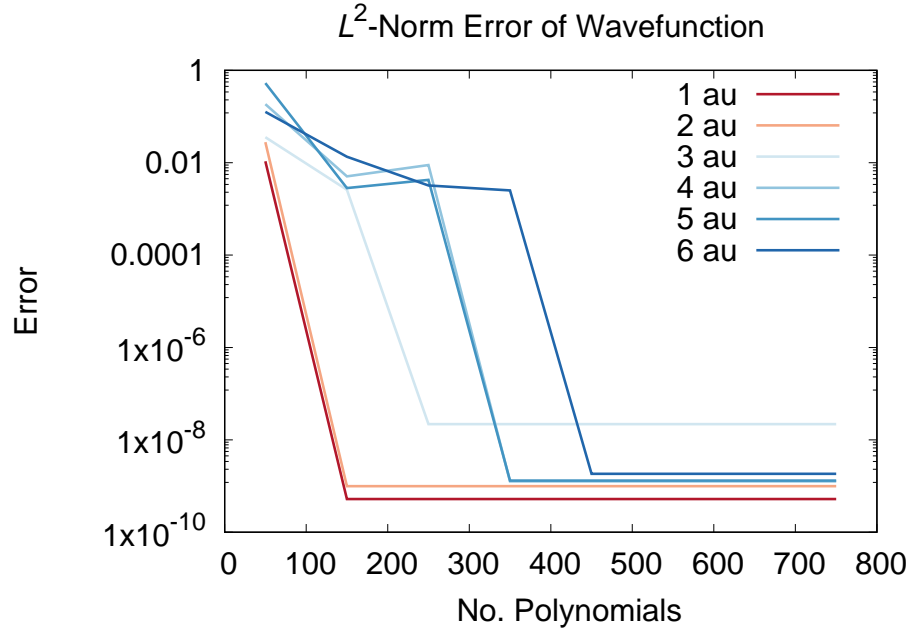


Figure 3: L^2 -norm error of the wavefunction in TTC as a function of the number of Chebyshev polynomials for varying final times.