

ARTICLE

Low-Storage Runge-Kutta Method for Simulating Time-Dependent Quantum Dynamics

Yun-an Yan*

Guizhou Provincial Key Laboratory of Computational Nano-Material Science, Guizhou Education University, Guizhou 550018, China
Beijing Computational Science Research Center, Beijing 100094, China

(Dated: Received on March 3, 2017; Accepted on March 17, 2017)

A wide range of quantum systems are time-invariant and the corresponding dynamics is dictated by linear differential equations with constant coefficients. Although simple in mathematical concept, the integration of these equations is usually complicated in practice for complex systems, where both the computational time and the memory storage become limiting factors. For this reason, low-storage Runge-Kutta methods become increasingly popular for the time integration. This work suggests a series of s -stage s th-order explicit Runge-Kutta methods specific for autonomous linear equations, which only requires two times of the memory storage for the state vector. We also introduce a 13-stage eighth-order scheme for autonomous linear equations, which has optimized stability region and is reduced to a fifth-order method for general equations. These methods exhibit significant performance improvements over the previous general-purpose low-stage schemes. As an example, we apply the integrator to simulate the non-Markovian exciton dynamics in a 15-site linear chain consisting of perylene-bisimide derivatives.

Key words: Low-storage Runge-Kutta, Autonomous linear differential equation, Time-dependent dynamics, Time-invariant Hamiltonian

I. INTRODUCTION

The physics laws of quantum dynamics, such as the Schrödinger's wave equation [1] and the Heisenberg matrix equation [2] for isolated systems, take the form of linear differential equations with constant coefficients. In reality, a quantum system is inevitably coupled to an environment and thus becomes open. When the coupling is weak and the memory time of the environment is short compared to the intrinsic time scale of the system, a Markovian approximation ignoring the memory time leads to a good description [3–5]. Otherwise, the hierarchical approach is a powerful tool to simulate the non-Markovian dynamics by embedding the non-Markovian system in a larger, possibly infinite Markovian system [6–9]. Once again, the equations of motion for the open systems take a linear differential form with constant coefficients if external driving fields are not applied. Formally, the dynamics for all of these cases are dictated by the ordinary differential equation

$$\dot{\mathbf{x}} = \mathcal{M}\mathbf{x} \quad (1)$$

together with an initial condition $\mathbf{x}(t_0)=\mathbf{x}_0$ at time t_0 . Here, \mathbf{x} is a state vector of length N and \mathcal{M} is an $N \times N$

matrix which does not depend on either the time t or the state vector \mathbf{x} .

The numerical integration of such an equation is simple in concept from the mathematical perspective. For systems with small to moderate dimension, one can benefit from the extensive studies on the linear differential equations [10–14]. The accumulated results include the Lanczos iteration in Krylov subspace [15–17], the Chebyshev scheme [18, 19], matrix exponential method [20–22], and the classical Runge-Kutta method [23].

In practice, however, the computations are complicated by the big size of the state vector in applying the quantum principles for complex systems. As the computer memory required to store the state vector grows exponentially with the size of the system, the length N of the state vector can easily reach 10^9 and even higher in various simulations. For large systems, the quantum dynamical simulation becomes expansive or even prohibited, which is called the curse of dimensionality [24]. This curse with respect to the memory issue is enhanced by the numerical integration because the integrator itself needs additional memory to store the state vector.

For this reason, low-storage methods draw a lot of attention. The split operator method, which is a symplectic integrator not requiring additional memory location for the wavefunction, is efficient for quantum mechanical systems without space-momentum mixed terms in the Hamiltonian [25]. For other systems, low-storage

* Author to whom correspondence should be addressed. E-mail: yunan@gznc.edu.cn

Runge-Kutta (LSRK) methods, which use less memory locations than the classical Runge-Kutta schemes, are useful in reducing the memory use [26–29]. Actually, the LSRK scheme was proposed at the early time of computer simulations [30–32], when the memory storage was small. Later, different groups made schemes using only two memory locations for any stage number (the number of derivative evaluations in each time step of the numerical integration) [27, 29, 33]. Kennedy and co-workers systematically studied the low-storage schemes and made the low-storage idea popular [34, 35].

Regardless of how powerful the high performance computing becomes and how big the memory storage is available, we always ask more to simulate bigger systems to reveal new mechanism, to seek new phenomena and to design new materials. A partial, alternative remedy for this problem is to develop efficient algorithm with not only shorter computational time but also less memory demand so that we can simulate bigger systems with available computational resources. Thus, the low-storage methods became increasingly popular during the past two decades although the available computer memory was dramatically improved.

The LSRK scheme assumes the same accuracy and robustness as the traditional explicit Runge-Kutta methods but only needs two memory locations for the state vector regardless of the order. By contrast, the classical explicit s -stage Runge-Kutta method normally requires $s+1$ memory locations for the state vector. But, the available LSRK methods were designed for general purpose problems, which suffers from the order barrier condition, namely, there is no s -stage s th-order explicit Runge-Kutta method for $s>4$ and no $(s+2)$ -stage s th-order method for $s>7$ [36]. The order barrier is severer for LSRK schemes because there are additional conditions in order to achieve the low-storage integration. For example, for general nonlinear multidimensional differential equation, we have to use five stages to get a fourth-order LSRK method [35]. To have an optimal stability region, a 12-stage fourth-order LSRK scheme is also widely used [37]. The larger stage number will significantly increase the computational time for the simulations of the dynamics.

Actually, we can develop more efficient LSRK schemes if we take advantage of the linearity of Eq.(1). Zingg and Chisholm [38] suggested a four-stage fourth-order LSRK method when adding an inhomogeneous time-dependent term to Eq.(1). Indeed, more efficient low-storage algorithm exists for linear differential equations with constant coefficients.

In this report, we show that s -stage s th-order LSRK method exists for Eq.(1), so that the time-dependent dynamics of time-invariant systems can be propagated very efficiently in the use of both computational time and computer memory. The developed methods are applied to the hierarchical simulations of the exciton diffusion in a 15-site linear aggregate.

In this work, we show that s -stage s -order LSRK

methods exist for any positive s and suggest the schemes for $s=4, 6, 8, 10$ and 12 . A 13-stage method with optimized stability region, which is of eighth-order for linear equation with constant coefficients and of fifth-order for general systems, is also suggested. We also presents a hierarchical simulation of the exciton diffusion in a chain of a J -type linear aggregate with the 12-stage 12th-order LSRK method.

II. LOW-STORAGE RUNGE-KUTTA METHOD FOR AUTONOMOUS LINEAR

A. Classical explicit Runge-Kutta methods

The Runge-Kutta methods are derivative-free schemes to find the numerical solution for the initial value problem of the ordinary differential equations

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}) \quad (2)$$

$$\mathbf{x}(t_0) = \mathbf{x}_0 \quad (3)$$

for a given function \mathbf{f} . A general s -stage Runge-Kutta method discretizes the time with a grid $\{t_0, t_1, \dots, t_n, \dots\}$ and approximates the solution of Eq.(2) step-by-step with s iterations in each step,

$$\mathbf{k}_1 = \mathbf{f}(t_n, \mathbf{x}_n) \quad (4)$$

$$\mathbf{k}_j = \mathbf{f}\left(t_n + c_j \Delta t, \mathbf{x}_n + \Delta t \sum_{l=1}^{j-1} a_{jl} \mathbf{k}_l\right), \quad j=2, \dots, s \quad (5)$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta t \sum_{j=1}^s b_j \mathbf{k}_j \quad (6)$$

where $\Delta t = t_{n+1} - t_n$ is the size of the time step, and \mathbf{x}_n is an approximation to $\mathbf{x}(t_n)$. Here, the coefficients a_{jk} , b_j and c_j are the elements of the Butcher tableau [23]. To have a compact presentation, we will treat \mathbf{a} as a matrix which only has non-zero matrix elements a_{jk} in the strict lower triangular part. The consistency condition of the Runge-Kutta method imposes the constraints

$$c_j = \sum_{l=1}^{j-1} a_{j,l}, \quad j = 2, \dots, s \quad (7)$$

Therefore, only a_{jk} and b_j are free parameters, which are chosen so that \mathbf{x}_{n+1} is the p -order Taylor expansion of $\mathbf{x}(t)$ at time t_n ,

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \sum_{j=1}^p \frac{\Delta t^j}{j!} \frac{d^j}{dt^j} \mathbf{f}(t, \mathbf{x}) \Big|_{\mathbf{x}=\mathbf{x}_n} + O(\Delta t^{p+1}) \quad (8)$$

If this condition holds, the Runge-Kutta scheme is called a p th-order method.

B. Low-storage Runge-Kutta methods

Eq.(3) clearly shows that the s -stage explicit Runge-Kutta scheme normally requires $s + 1$ memory locations for integration, which is not favorable for large systems. By putting further constraints on the coefficients a_{jk} and b_j , we may achieve a low-storage method which requires less memory locations, even two memory locations regardless of the stage number. In the literature, there exist different schemes to achieve the two-register method [27, 29, 33]. Here, we will be stuck to the formulation originally proposed by Williamson [29], which takes the following iteration in each time step

$$\begin{aligned} &\text{for } j = 1, \dots, s, \\ &\text{do the iteration} \\ &\quad \Delta \mathbf{y} = A[j] \Delta \mathbf{y} + \Delta t \mathbf{f}(t_n + c_j \Delta t, \mathbf{y}) \quad (9) \\ &\quad \mathbf{y} = \mathbf{y} + B_j \Delta \mathbf{y} \quad (10) \\ &\text{end of the iteration} \end{aligned}$$

Here the register \mathbf{y} points to the the memory location storing the state vector \mathbf{x} , which contains the initial condition \mathbf{x}_n before the iteration and directly gives \mathbf{x}_{n+1} after the iteration. An explicit scheme implies $A_1=0$, so there are $2s-1$ independent parameters for an s -stage scheme, based on which the classical Butcher tableau can be constructed [37]

$$a_{jk} = \begin{cases} A_{k+1}a_{j,k+1} + B_k, & \text{if } k < j - 1 \\ B_k, & \text{if } k = j - 1 \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

$$b_j = \begin{cases} A_{j+1}b_{j+1} + B_j, & \text{if } j < s \\ B_j, & \text{if } j = s \end{cases} \quad (12)$$

For the autonomous linear equation Eq.(1), all second-order partial derivatives vanish, namely,

$$\frac{\partial^2 \mathbf{f}}{\partial t^2} = \frac{\partial^2 \mathbf{f}}{\partial t \partial \mathbf{x}} = \frac{\partial^2 \mathbf{f}}{\partial \mathbf{x} \partial \mathbf{x}} = 0 \quad (13)$$

As a consequence, the order condition Eq.(8) is greatly simplified,

$$\begin{aligned} \sum_{j=1}^s b_j &= 1 \\ \sum_{j,l=1}^s b_j [\mathbf{a}^{k-1}]_{jl} c_l &= \frac{1}{k!}, \quad k = 1, \dots, p - 1 \end{aligned} \quad (14)$$

Therefore, there are p constraints in order to have a p th-order method, which can normally be satisfied because there are $2p-1$ parameters in a p -stage scheme. This implies that s -stage s -order LSRK methods exist for the autonomous linear differential equations. Although there are multiple solutions of Eq.(14), particular interest is in the choice $A_j=-1$ for $j \geq 2$. With such a choice, these algebraic equations can be conveniently

TABLE I The coefficients B_j of the low-storage Runge-Kutta methods for $s=4, 6, 8, 10$ and 12 for the homogeneous linear differential equations with constant coefficients.

j	B_j				
	$s=4$	$s=6$	$s=8$	$s=10$	$s=12$
1	$\frac{1}{3}$	$\frac{7}{15}$	$\frac{3923}{9765}$	$-\frac{8549}{19215}$	$\frac{580674203}{2261068425}$
2	$\frac{3}{4}$	$\frac{15}{14}$	$\frac{181629}{407992}$	$-\frac{1172115}{25424726}$	$\frac{42155682725475}{139531365587276}$
3	$\frac{2}{3}$	$-\frac{1}{15}$	$\frac{3380}{13671}$	$\frac{2211169}{2171295}$	$\frac{7217530658}{19832800185}$
4	$\frac{1}{4}$	$-\frac{5}{12}$	$\frac{343}{936}$	$\frac{446915}{844616}$	$\frac{181429325}{105488188}$
5		$\frac{3}{5}$	$-\frac{54}{245}$	$\frac{10082}{43505}$	$-\frac{192721}{51245975}$
6		$\frac{1}{6}$	$-\frac{7}{72}$	$\frac{847}{7100}$	$-\frac{368449}{298520}$
7			$\frac{4}{7}$	$-\frac{250}{693}$	$\frac{12716}{38241}$
8			$\frac{1}{8}$	$-\frac{9}{200}$	$\frac{45}{952}$
9				$\frac{5}{9}$	$-\frac{49}{99}$
10				$\frac{1}{10}$	$-\frac{11}{420}$
11					$\frac{6}{11}$
12					$\frac{1}{12}$

and analytically solved with a Gröbner basis decomposition [39]. The coefficients B_j for the schemes with 4, 6, 8, 10 and 12 stages are given in Table I. Hereafter, the method will be named as LSRK s - p (q), where s refers to the number of stages, p denotes the order of the method for autonomous linear equations. When it applies, q means the order of the method for general differential equations.

In numerical applications, another important issue is the stability of the solver, which requires that small errors in the initial state do not drive the Runge-Kutta approximation off from the real value. The stability analysis of the solver is complicated and case-dependent for general systems. Thus a paradigm for the stability analysis is to use the one-dimensional equation with a complex coefficient λ

$$\dot{x} = \lambda x \quad (15)$$

TABLE II The coefficients of the 13-stage eighth-order low-storage Runge-Kutta method for autonomous linear differential equations. This method is of the fifth-order for time-variant or nonlinear differential equations.

j	γ_j	A_j	B_j	c_j
1	1	0	0.069632640247059393	0
2	1/2	-0.33672143119427413	0.088918462778092020	0.069632640247059393
3	1/6	-1.2018205782908164	1.0461490123426779	0.12861035097891748
4	1/24	-2.6261919625495068	0.42761794305080487	0.34083022189561149
5	1/120	-1.5418507843260567	0.20975844551667144	0.54063706308495402
6	1/720	-0.2845614242371758	-0.11457151862012136	0.59927749518613931
7	1/5040	-0.1700096844304301	-0.01392019988507068	0.49382042519248519
8	1/40320	-1.0839412680446804	4.0330655626956709	0.48207852767699775
9	$2.6145987936825117 \times 10^{-6}$	-11.61787957751822	0.35106846752457162	0.82762865209834452
10	$2.1032335098184808 \times 10^{-7}$	-4.5205208057464192	-0.16066651367556576	0.82923953914857933
11	$1.1468372886291829 \times 10^{-8}$	-35.86177355832474	-0.0058633163225038929	0.67190565554748019
12	$3.6778983665015162 \times 10^{-10}$	-0.00002134089996007288	0.077296133865151863	0.87194975193167848
13	$5.1875143617757772 \times 10^{-12}$	-0.066311516687861348	0.054301254676908338	0.94930216564503562

to obtain useful guidance. The coefficient λ corresponds to the eigenvalues in multi-dimensional problems. For this one-dimensional case, the explicit Runge-Kutta method approximates $x(t_0+n\Delta t)$ with $x_n=G(\lambda\Delta t)x_{n-1}$, where $G(z)$ is the polynomial

$$G(z) = 1 + \sum_{j=1}^s \gamma_j z^j \quad (16)$$

with the coefficients γ_j being

$$\gamma_j = \begin{cases} \sum_{k=1}^s b_k, & \text{if } j = 1 \\ \sum_{k,l=1}^s b_k [\mathbf{a}^{j-2}]_{kl} c_l, & \text{if } j > 1 \end{cases} \quad (17)$$

The iteration Eqs.(4–6) is robust against small perturbations if the characteristic polynomial $G(\lambda\Delta t)$ fulfills the criterion $|G(\lambda\Delta t)| \leq 1$. The stability region of the Runge-Kutta method is then defined as the set of all points z in the complex plane satisfying $|G(z)| \leq 1$.

For schemes with $s=p$, all γ_j are determined by the order condition Eq.(14) so that there is no freedom to manipulate the stability region. Therefore, in order to optimize the stability, we have to sacrifice the order of the method to have free parameters.

Following Ref.[37], we adopt a two-stage process to find the LSRK method with optimized stability. For a given stage number s and a desired order p , we first find γ_j ($p < j \leq s$) to optimize the area within the contour $|G(z)|=1$. We then determine the coefficients A_j and B_j satisfying the order conditions with the γ_j .

In principle, we have to optimize the stability with the exact area within the contour $|G(z)|=1$. Here, we will adopt a much simpler, yet reasonable, approxima-

tion to this area, namely,

$$S[G] \approx \frac{x_{\max}}{9} \sum_{j=1}^9 y_j \quad (18)$$

where x_{\max} is the maximum positive real number satisfying $|G(-x_{\max})|=1$, and y_j is the minimum positive number solving $|G(iy_j - jx_{\max}/10)|=1$, with $i=\sqrt{-1}$ being the imaginary unit.

Once γ_j are obtained, we proceed to find the LSRK parameters A_j and B_j . In this work, we are interested in the 13-stage 8-order method to have balance between the order and the stability. With this choice, there are five free parameters γ_j ($9 \leq j \leq 13$) to optimize the stability at the first step. The optimization is carried out by a random search with the Nelder-Mead method [40] to find the local maximum as implemented in the NLOpt library [41]. The optimized results for γ_j are listed in Table II. At the second step, there are 25 A_j and B_j parameters satisfying 13 order conditions. In order to fix the rest 12 parameters, we require the scheme to be a fifth-order method for general differential equations. Besides the order conditions for linear equations with constant coefficients, there are 12 additional constraints

$$\sum_{j=1}^s b_j c_j^2 = \frac{1}{3} \quad (19)$$

$$\sum_{j=1}^s b_j c_j^3 = \frac{1}{4} \quad (20)$$

$$\sum_{j=1}^s b_j c_j^4 = \frac{1}{5} \quad (21)$$

$$\sum_{j,k=1}^s b_j c_j a_{jk} c_k = \frac{1}{8} \quad (22)$$

$$\sum_{j,k=1}^s b_j c_j^2 a_{jk} c_k = \frac{1}{10} \quad (23)$$

$$\sum_{j,k=1}^s b_j c_j a_{jk} c_k^2 = \frac{1}{15} \quad (24)$$

$$\sum_{j,k=1}^s b_j c_j [\mathbf{a}^2]_{jk} c_k = \frac{1}{30} \quad (25)$$

$$\sum_{j,k=1}^s b_j a_{jk} c_k^2 = \frac{1}{12} \quad (26)$$

$$\sum_{j,k,l=1}^s b_j a_{jk} c_k a_{kl} c_l = \frac{1}{40} \quad (27)$$

$$\sum_{j,k=1}^s b_j [\mathbf{a}^2]_{jk} c_k^2 = \frac{1}{60} \quad (28)$$

$$\sum_{j,k,l=1}^s b_j a_{jk} c_k a_{kl} c_l = \frac{1}{20} \quad (29)$$

$$\sum_{j,k=1}^s b_j a_{jk} c_k^3 = \frac{1}{20} \quad (30)$$

The nonlinear Eq.(14) and Eqs.(19–30) are solved with the Levenberg-Marquardt algorithm [42, 43] as implemented in the levmar package [44] with a modified long double precision. Multiple solutions are found in determining A_j and B_j . The results for A_j , B_j and c_j with smallest sum of the absolute values of A_j are given in Table II, showing 17 effective digits.

C. Stability contours

FIG. 1 presents the stability contours normalized by the inverse of the stage number for all the methods listed in Tables I and II. As the number of stages coincides the order of the methods, the stability contours of LSRK4-4, LSRK6-6, LSRK8-8, LSRK10-10 and LSRK12-12 schemes are uniquely defined and do not depend on how the parameters A_j and B_j are chosen. For these methods, the stability region is larger when the stage number increases. However, FIG. 1 clearly shows that the renormalized region shrinks for larger stage numbers. When the stability is the limiting factor in the numerical integration, the maximum step size is controlled by the stability region. Then, the efficiency of the un-optimized high order methods is lower because the increase of the stability region is not sufficient to compensate the cost in performing the calculations for the additional stages. In these cases, it is appreciated to sacrifice the order of the method to increase the stability region. As shown, normalized stability region of the LSRK13-8(5) scheme is significantly bigger than that of the un-optimized methods. A dominant feature is that its contour is elongated along the x -axis. Correspondingly, LSRK13-8(5) is suitable to sim-

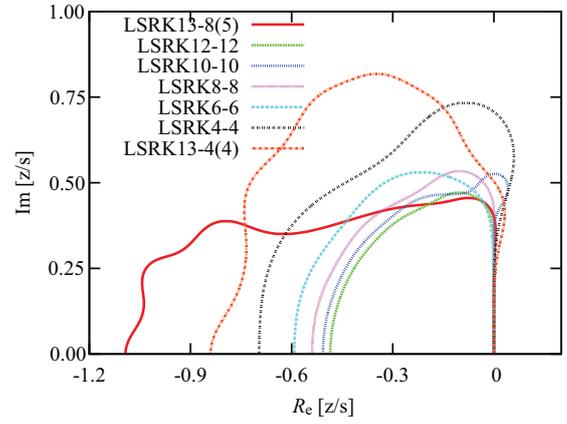


FIG. 1 Stability contours for the methods suggested in Tables I and II. The contours are symmetric with respect to the x -axis, and the lower parts thus are not displayed. For comparison, also shown is the 13-stage fourth-order scheme [LSRK13-4(4)] for general differential equations proposed by Niegemann *et al.* [37].

ulate the over-damped motion, in which the real part of the eigenvalues is larger than the imaginary part. To have a comparison, we also plot the stability contour for the LSRK13-4(4) method suggested by Niegemann *et al.* [37]. This method have a stability region elongated along y -axis and is suitable to simulate the under-damped motion, in which the imaginary part of the eigenvalues is dominant.

D. Accuracy verification

We check the efficiency of all the methods with a symmetric matrix \mathcal{M} . Here, the adopted eigenvalues λ_j are

$$\lambda_j = \frac{5i(j-1)}{N} - \frac{4(j-1)}{N} \exp \left[1 - \frac{4(j-1)}{N} \right] \quad 1 \leq j \leq N \quad (31)$$

where $N=256$ is the dimension of the matrix \mathcal{M} . With this settings, $\lambda_1=0$ is the only zero of the eigenvalues, which implies that there is a unique stationary state of the systems.

For the simplicity of computation, we choose the discrete Fourier transform matrix $F_{jk}=\exp[\pi i(j-1)(k-1)/N]$ as the eigen-matrix. Then the exact solution at time t is calculated with discrete Fourier transform, namely,

$$\mathbf{x}^{(\text{ex})}(t) = F\Lambda(t)F^\dagger \mathbf{x}(0) \quad (32)$$

where $\Lambda(t)$ refers to the diagonal matrix taking the eigenvalues $\exp(-\lambda_j t)$, and the initial vector $\mathbf{x}(0)$ at $t=0$ is filled with standard normal Gaussian numbers. The errors of the numerical integration are defined as

the Euclidean distance between the exact result and the Runge-Kutta approximation $\mathbf{x}^{(\text{appr})}(t)$

$$\text{Err}(t) = \left| \sum_{j=1}^N \left[\mathbf{x}_j^{(\text{ex})}(t) - \mathbf{x}_j^{(\text{appr})}(t) \right]^2 \right|^{1/2} \quad (33)$$

In FIG. 2, we present the error of all the methods with the step size of $2^n \times 10^{-3}$ ($0 \leq n \leq 12$). The results clearly show that there exists a region that the accuracy will be comparable with the machine round-off errors and they cannot be reduced further simply by decreasing the step size. There also exists an unstable region for the each integrator, when the step size is bigger than a critical value, the error increases explosively. The error corresponding to this critical step size is about one, which implies that the step size in this problem is controlled by the accuracy.

Not shown in FIG. 2, we also carry out the integration with the classical fourth-order Runge-Kutta method, the Fehlberg's 4(5) method [45], the Calvo 6(5) method [46], and the Prince-Dormand 8(7) method [47]. The 4th-order LSRK is identical to the classical fourth-order Runge-Kutta method in errors as well as the stability. Higher-order LSRK schemes are faster than the same order classical RK ones, but no more than a factor of two. However, a high-order LSRK ($p > 6$) scheme can be two or three times faster than the corresponding classical Runge-Kutta method which assumes the same number of stages.

Provided with a pre-selected error tolerance, higher order methods normally can take a bigger step size. For example, with the error tolerance of 1×10^{-5} , the step size can be as large as 0.31 for LSRK12-12, but only 0.0083 for LSRK4-4. If the error tolerance decreases by two orders, the corresponding step size becomes 0.21 for LSRK12-12 and 0.0026 for LSRK4-4. Roughly speaking, the speedup will be 9 to 21 times if we use a 12-stage 12-order method instead of a fourth-order Runge-Kutta scheme. Other higher s -stage s -order methods also depict an increase of 3–20 times in efficiency, depending on the order and the error tolerance. With the same step size in the stable region, the stability-optimized method LSRK13-8(5) always produces smaller errors than the same order method LSRK8-8. When the additional stages of the LSRK13-8(5) scheme are taken into account, the LSRK13-8(5) scheme has the same performance as the LSRK8-8 method.

Again, to have comparison, we consider the LSRK13-4(4) scheme which is designed for general differential equations. With a given step size, it systematically produces smaller errors than that of the LSRK4-4 method. However, provided the same accuracy (1×10^{-5} or 1×10^{-7}), the increase in the step size is less than three times. We thus conclude that the LSRK13-4(4) scheme is less efficient than the LSRK4-4 method.

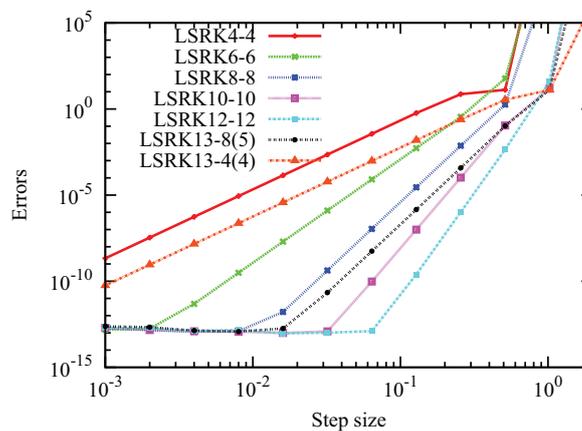


FIG. 2 (Color online) Errors of the methods suggested in Tables I and II for the linear problem in Section II D at time $t=8.192$. For comparison, also shown is the 13-stage fourth-order scheme [LSRK13-4(4)] for general differential equations proposed by Niegemann [37]. Dots are calculated results and lines are guide for the eye.

III. APPLICATIONS TO SIMULATION OF QUANTUM COHERENCE IN BIOLOGICAL-RELEVANT SYSTEMS

We will apply the high-order low-storage methods outlined in Section II to a more realistic simulation, that is, the non-Markovian exciton diffusion in a linear aggregate. The exciton dynamics plays a crucial role in organic photovoltaic cell [48], photosynthesis [49], organic light-emitting diodes [50] and many others [51, 52]. In such systems, energy transfers to the reaction center along with the exciton diffusion. The natural photosynthesis systems exhibit surprisingly high efficiency in the exciton energy transfer [49]. The theoretical understanding of the design features and the underline mechanism will greatly help the artificial solar energy application [53]. We are still at the very beginning to understand the process and now expect that dissipation caused by the environment is the key for efficient energy transport [49, 53].

Here, we study the noise-assisted exciton diffusion in an artificial chain consisting of N, N' -Di(N -(2-aminoethyl)-benzamide)-1,6,7,12-tetra(4-tert-butylphenoxy)-3,4:9,10-tetracarboxylic acid bisimide (PBI-1). Perylene bismide systems are very flexible in tuning the geometric configuration as well as the electronic properties in a large extent upon proper substitution or different solvation [54–56]. Therefore, they serve as idea test systems for theories to investigate the effects of various factors on exciton transfer and have been extensively studied in the literature. For example, Wolter and co-workers investigated the size-dependent exciton dynamics in one-dimensional PBI aggregates [57]. A maximal diffusion length was estimated to be 96 nm, which is about 10 times larger than that in disordered polymers [58]. The photophysical properties of PBI monomer and oligomers have been calculated

in Refs.[59, 60].

As the time scale of the exciton dynamics is of the order of picosecond and longer, it is comparable to that of the environment. Thus, the dynamics is essentially non-Markovian. Here, the non-Markovian dynamics is investigated with the hierarchical approach, which introduces auxiliary density matrices to characterize the effect of the bath and is one of a few available methods to study non-Markovianity [6–9, 61–64]. Usually, the number of auxiliary matrices are large and the requirement for the memory storage is high at low temperatures. The low-storage scheme thus is ideal to integrate the hierarchical equations.

A. Hierarchical approach for dissipative exciton dynamics

In order to focus on the exciton diffusion, the exciton is supposed to be stable and its dissociation as well as the decay is not taken into account. Moreover, we only investigate the dynamics in the one-exciton manifold, that is, only the diabatic electronic state corresponding to the S_0 state is included. With this restriction, there are only N_{agg} one-exciton states $|m\rangle$ (only site m is in S_1) for an N_{agg} -site aggregate.

The Hamiltonian for the exciton diffusion under dissipation can be approximated with the system-plus-bath model,

$$\hat{H}_{\text{tot}} = \hat{H}_{\text{ex}} + \sum_m \hat{H}_m + \hat{H}_b \quad (34)$$

where \hat{H}_{ex} is the Hamiltonian of the exciton, and \hat{H}_b is the Hamiltonian of the linear bath consisting of independent harmonic oscillators. We will investigate the homogeneous aggregate, so we only include the site-site coupling leading to single exciton transfer,

$$\hat{H}_{\text{ex}} = \sum_{mn} J_{mn} |m\rangle \langle n| \quad (35)$$

where J_{mn} is the coupling constant.

When the size of the monomer in the aggregate is big, each site may have a set of localized vibrations coupled to the exciton at this site only. Operator \hat{H}_m in Eq.(34) stands for the exciton-vibration coupling at site m and a linear form is accepted [65]

$$\hat{H}_m = \hat{h}_m \sum_{\xi} \hat{q}_{m,\xi} g_{m,\xi}, \quad \hat{h}_m \equiv |m\rangle \langle m| \quad (36)$$

For linear dissipation, the effect of the vibrations on the exciton dynamics is fully captured by the bath response function,

$$\alpha(t) = \frac{1}{\pi} \int_0^{\infty} d\omega J(\omega) \left[\coth\left(\frac{\beta\hbar\omega}{2}\right) \cos(\omega t) - i \sin(\omega t) \right] \quad (37)$$

which in turn is determined by the spectral density function $J(\omega)$ [62, 64]. Here, $\beta=1/k_B T$ is the inverse temperature rescaled by the Boltzmann constant

k_B . Generally, the spectral density function can be expressed in terms of the coupling constant, reduced mass, and frequency of each oscillator in the bath. But most often theoretical simulations adopt special form to model the bath. The localized vibrational modes normally have high harmonic frequencies and can be described by a damped Brownian oscillator bath model [66]

$$J(\omega) = \frac{\eta\omega_0^3\omega\Lambda}{[(\omega^2 - \omega_0^2)^2 + \Lambda^2\omega^2]} \quad (38)$$

where η is the dissipation strength, ω_0 the central mode frequency, and Λ the cutoff. Below we will refer to the under-damped limit, where the central mode frequency ω_0 is much larger than the cutoff Λ .

For this spectral density the response function can be expanded into a sum of exponentials as follows [62]

$$\begin{aligned} \alpha(t) &= \frac{\eta\hbar\omega_0^3}{4\xi} \left\{ -e^{\Omega_1 t} \left[\coth\left(\frac{i}{2}\hbar\beta\Omega_1\right) - 1 \right] \right. \\ &\quad \left. + e^{\Omega_2 t} \left[\coth\left(\frac{i}{2}\hbar\beta\Omega_2\right) - 1 \right] \right\} \\ &\quad - 2e\omega_0^3\Lambda^2\hbar\beta \sum_{k=1}^{\infty} \frac{\nu_k e^{-\nu_k t}}{(\nu_k^2 + \omega_0^2)^2 - \Lambda^2\nu_k^2} \\ &\equiv \sum_{j=1}^{N_e} c_j \exp(-\Omega_j t) + \alpha^{(\text{short})}(t) \end{aligned} \quad (39)$$

where $\Omega_{1,2} = \Lambda/2 \pm i\xi$ with $\xi = \sqrt{\omega_0^2 - \Lambda^2/4}$, and $\Omega_{k+2} = 2\pi k/(\hbar\beta)$ ($k>0$) is the k th Matsubara frequency of the bath. In writing this equation, we have separated the short memory contribution of the response function with the smallest integer N_e satisfying $\Omega_{N_e} \gg \Lambda$.

With the exponential decomposition of the response functions, the hierarchical approach is efficient to propagate the reduced density matrix for the description of the non-Markovian exciton dynamics. The hierarchical approach is a scheme to attack the non-perturbative non-Markovian dynamics for the open quantum system by converting the memory of the bath into a series of auxiliary density matrices. Here we will follow the stochastic decoupling procedure of Shao and coworkers [9, 67]. Derivation based on fundamental Itô calculus can be also found somewhere else [68, 69]. The resulting equations are as follows

$$\begin{aligned} i\hbar \frac{\partial \tilde{\rho}_A}{\partial t} &= -i\hbar \left(\sum_{m=1}^{N_{\text{agg}}} \sum_{j=1}^{N_e} A_{m,j} \Omega_j + \mathcal{R} \right) \tilde{\rho}_A \\ &\quad + \left[\hat{H}_{\text{ex}}, \tilde{\rho}_A \right] + \sum_{m=1}^{N_{\text{agg}}} \sum_{j=1}^{N_e} \left[\hat{h}_m, \tilde{\rho}_{A_{m,j}^+} \right] \\ &\quad + \sum_{m=1}^{N_{\text{agg}}} \sum_{j=1}^{N_e} A_{m,j} \left(\alpha_{j,+} \left[\hat{h}_m, \tilde{\rho}_{A_{m,j}^-} \right] + \right. \\ &\quad \left. \alpha_{j,-} \left\{ \hat{h}_m, \tilde{\rho}_{A_{m,j}^-} \right\} \right) \end{aligned} \quad (40)$$

$\alpha_{j;\pm}$ are defined as:

$$\alpha_{j;+} = \frac{c_j + c_{j'}^*}{2}, \quad \alpha_{j;-} = \frac{-i(c_j - c_{j'}^*)}{2} \quad (41)$$

with j' being the index satisfying $\gamma_{j'}^* = \gamma_j$. The matrix A has the dimension of $N_{\text{agg}} \times N_e$ and its matrix element $A_{m,j}$ refers to the j -th exponential in the response function of the bath for the m -th site. The operator “+/-” is defined as $[A_{m,j}^{\pm}]_{n,k} = A_{m,j} \pm \delta_{mn} \delta_{jk}$. In Eq.(40), \mathcal{R} is the Redfield superoperator which approximates the effect of the short memory part of the response function in Eq.(39) [62].

In the above equation $\tilde{\rho}_0$ is the reduced density matrix for the exciton with the initial condition $\tilde{\rho}_0(0) = \hat{\rho}(0)$. And $\tilde{\rho}_A (A \neq 0)$ are the auxiliary density matrices which characterize the memory effect caused by the system-bath interaction with the initial condition $\tilde{\rho}_A(0) = 0$.

Eq.(40) is an infinite-dimensional differential equation with constant coefficients. In numerical simulations, it is an unattainable yet unnecessary task to explicitly treat an infinite system. In practice, one can safely truncate the hierarchy equation to the L th layer with a sufficiently large integer L to yield a preselected accuracy. That is, all terms $\rho_A(t)$ with $\sum_{m=1}^{N_{\text{agg}}} \sum_{j=1}^{N_e} A_{m,j} > L$ are set to zero [69].

B. Results and discussions

As far as the present model is concerned, we will adopt the same parameters as those in Ref.[62]. Specifically, in the excitonic part we will only consider the nearest neighbor coupling and use the calculated value $J_{mn} = -514 \text{ cm}^{-1}$ [60]. For the bath, we assume all of them are independent and identical and take the theoretical calculated values $\omega_0 = 1415 \text{ cm}^{-1}$ and $\eta = 0.44$ in the spectral density function Eq.(32) [59]. The simulations are carried out at a temperature of 300 K with an initial condition $\rho(0) = |1\rangle\langle 1|$.

There are two key factors for the hierarchical approach to obtain converged results. One is the number of exponentials N_e to approximate the bath response function. The other is the level of truncation L of the hierarchy. The number of auxiliary density matrices depends exponentially on N_e and L . To reduce the number of auxiliary density matrices and yet keep the accuracy, all matrices $\tilde{\rho}_A$ with the factor $\sum_{j=1}^{N_e} A_{m,j} \Omega_j > 4.8\pi\hbar/\beta$ are not included in the hierarchy. For the present model, choice with $N_e = 3$ for all bathes and $L = 6$ yields converged results for the dynamics at temperature of 300 K.

With these settings, there are totally 1,674,992 auxiliary density matrices in the hierarchy and it takes

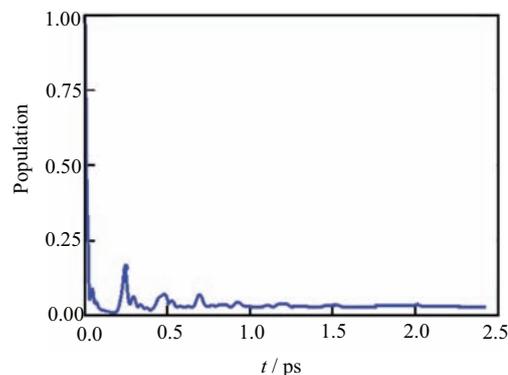


FIG. 3 Exciton population evolution at site one.

about 5.6 GB memory to store all complex matrices. A LSRK method with any order will need 11.2 GB memory storage for the numerical propagation. By contrast, the classical fourth-order Runge-Kutta method requires 16.8 GB space; the fifth-order Runge-Kutta-Fehlberg method needs 39.2 GB [45].

The numerical simulations are done with the Hyshe package [70], which employs a shared memory parallelization. Specifically, we use a 24-core Intel(R) Xeon(R) E5-2650v3 CPU with a clock speed of 2.30 GHz for the calculations. The efficiency is estimated as about 0.75 for this simulation. With the LSRK12-12 scheme, the dynamics can be propagated using a step size of 100 a.u. (~ 2.42 fs). As such, the numerical integration takes 6.5 h for 1000 steps.

The exciton population at site one is shown in FIG. 3. The population follows a steady equilibration after an initial quick drop. Note that there exist three clear spikes around 0.24, 0.47, 0.69 ps, and minor ones around 0.92, 1.2, and 1.5 ps. The peak heights decrease as time elapses, but the peak separation is almost equal in time. One may ask why this quasi-periodic phenomenon emerges?

To answer this question, we scrutinize the exciton population at all sites and present the data in FIG.4. The results reveal that the exciton transfers in the aggregate with a constant rate and is bounced by the two ends of the aggregate. Together with the dissipation induced by the vibrations, these two factors lead to an under-damped exciton motion. Based on this observation, the exciton hopping rate is about 123.4 ps^{-1} , which is significantly larger than $7.8 \pm 2.7 \text{ ps}^{-1}$, a value measured with ultrafast transient absorption spectroscopy [56]. The reason of this gap is because we are using the theoretically calculated bare Coulomb coupling J_{mn} of -514 cm^{-1} , which is larger than the experimental value (120 cm^{-1}) [56]. A closer examination of the population at each time reveals that the exciton population spread in the aggregate and thus the exciton transfer as a wave. Of course, the environment causes dissipation and the exciton diffuses along the aggregate irreversibly.

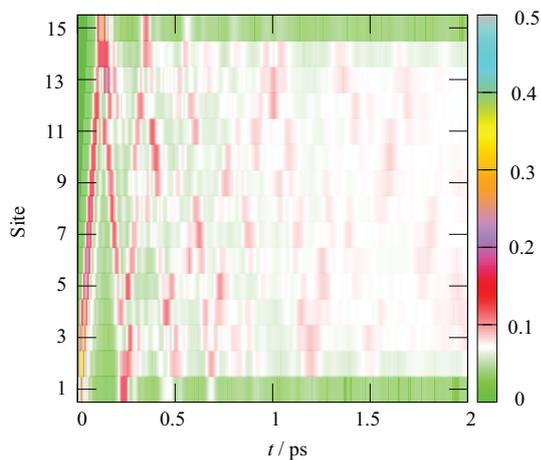


FIG. 4 Exciton population evolution of the aggregate. The region $[n - 1/2, n + 1/2]$ along the y -axis refers to the population at Site n .

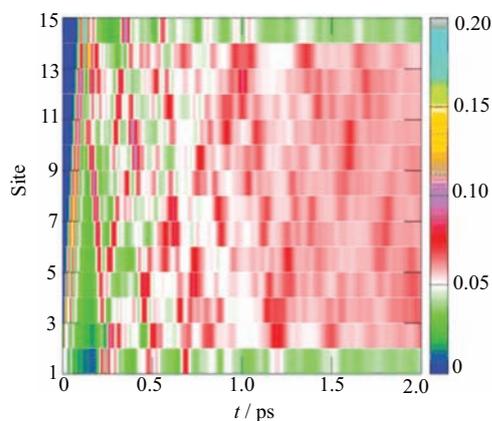


FIG. 5 Coherence beating in the aggregate. The region $[n, n + 1]$ along the y -axis refers to the absolute value of the off-diagonal matrix elements $\rho_{n,n+1}(t)$ of the reduced density matrix.

The wave-like motion of the exciton transfer can also be manifested by the coherence beating. FIG. 5 plots the absolute value of the off-diagonal matrix elements $\rho_{n,n+1}(t)$ ($1 \leq n \leq 14$) of the reduced density matrix at different times. The coherence exhibits a similar pattern as the population. As time elapses, the coherence does not vanish and approaches a constant instead, which reflects the exciton entanglement in equilibrium.

IV. SUMMARY

Previously, low-storage Runge-Kutta methods were designed for integration of general nonlinear or time-variant systems, which is efficient in the use of the memory storage. To generate a low-storage scheme for general equations, we have to take additional stages because there are more restrictions for the method. However, the dynamics of many quantum systems is gov-

erned by linear differential equations with constant coefficients. For these autonomous linear equations, there are only p restrictions versus $2p-1$ parameters for a p -stage low-storage method. Thus, the order barrier does not apply for Runge-Kutta methods in solving autonomous linear equations. The system of equations for the parameters in the current scheme can be solved analytically and it is straightforward to design a new LSRK method. This allows us to obtain p -stage p th-order low-storage methods which require less stages than the classical p th-order Runge-Kutta schemes when p is larger than four. While the general LSRK method consumes longer computational time than the same order classical Runge-Kutta scheme, the current method takes less computational time for autonomous linear equations. In this work, we have introduced a series of low-storage schemes specific for autonomous linear equations.

Numerics shows that the 12-stage 12th-order LSRK method could be an order of magnitude faster than the fourth-order Runge-Kutta method.

As an example, we have applied the suggested 12-stage 12th-order method for the hierarchical simulations of the exciton diffusion in a 15-site linear aggregate, where it takes 5.6 GB memory to store the reduced and auxiliary density matrices. Thanks to the efficient low-stage scheme, we can propagate the non-Markovian dynamics up to 2.4 ps within 6.5 h with a 24-core Intel(R) Xeon(R) E5-2650v3 CPU.

V. ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (No.21373064), the Program for Innovative Research Team of Guizhou Province (No.QKTD[2014]4021), and the Natural Scientific Foundation from Guizhou Provincial Department of Education (No.ZDXK[2014]18). All the calculations were performed at Guizhou Provincial High-Performance Computing Center of Condensed Materials and Molecular Simulation in Guizhou Education University.

- [1] E. Schrödinger, Ann. Phys. **384**, 361 (1926).
- [2] M. Born, W. Heisenberg, and P. Jordan, Zeit. Phys. **35**, 557 (1925).
- [3] F. Bloch, Phys. Rev. **70**, 460 (1946).
- [4] A. G. Redfield, IBM J. Res. Dev. **1**, 19 (1957).
- [5] G. Lindblad, Commun. Math. Phys. **48**, 119 (1976).
- [6] V. Shapiro and V. Loginov, Physica A **91**, 563 (1978).
- [7] Y. Tanimura and R. Kubo, J. Phys. Soc. Jpn. **58**, 101 (1989).
- [8] Y. Tanimura, Phys. Rev. A **41**, 6676 (1990).
- [9] Y. A. Yan, F. Yang, Y. Liu, and J. Shao, Chem. Phys. Lett. **395**, 216 (2004).

- [10] L. Adrianova, *Introduction to Linear Systems of Differential Equations, Translations of Mathematical Monographs*, Vol.146, Providence, Rhode Island: AMS, (1995).
- [11] A. Greenbaum, *Iterative Methods for Solving Linear Systems, Frontiers in Applied Mathematics, Vol.17*, Philadelphia: SIAM, (1997).
- [12] E. Coddington and R. Carlson, *Linear Ordinary Differential Equations*. Philadelphia: SIAM, (1997).
- [13] A. Forsyth, Part 3. *Ordinary Linear Equations, Theory of Differential Equations Vol.4*, Cambridge: Cambridge University Press (1902).
- [14] Y. Saad, *Iterative Methods for Sparse Linear Systems 2nd Edn.*, Philadelphia: SIAM, (2003).
- [15] C. Lanczos, *J. Res. Nat. Bur. Standards* **45**, 255 (1950).
- [16] R. Haydock, *Comput. Phys. Commun.* **20**, 11 (1980).
- [17] V. Druskin and L. Knizhnerman, *J. Comput. Appl. Math.* **50**, 255 (1994).
- [18] H. Tal-Ezer and R. Kosloff, *J. Chem. Phys.* **81**, 3967 (1984).
- [19] C. Leforestier, R. Bisseling, C. Cerjan, M. Feit, R. Friesner, A. Guldborg, A. Hammerich, G. Jolicard, W. Karlein, H. Meyer, N. Lipkin, O. Roncero, and R. Kosloff, *J. Comput. Phys.* **94**, 59 (1991).
- [20] R. C. Ward, *SIAM J. Numer. Anal.* **14**, 600 (1977).
- [21] M. Hochbruck, C. Lubich, and H. Selhofer, *SIAM J. Sci. Comput.* **19**, 1552 (1998).
- [22] R. B. Sidje, *ACM Trans. Math. Softw.* **24**, 130 (1998).
- [23] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*, 2nd Edn. Chichester: Wiley, (2008).
- [24] R. E. Bellman, *Dynamic Programming*, Princeton: Princeton University Press, (1957).
- [25] M. Feit, J. Fleck, and A. Steiger, *J. Comput. Phys.* **47**, 412 (1982).
- [26] P. Houwen, *Numer. Math.* **20**, 149 (1972).
- [27] P. J. Van Der Houwen, *Construction of Integration Formulas for Initial Value Problems, Vol.19*, Amsterdam: North-Holland, (1977).
- [28] A. Wray, *NASA Ames Research Center*, California: Moffett Field, 202 (1990).
- [29] J. Williamson, *J. Comput. Phys.* **35**, 48 (1980).
- [30] S. Gill, *Math. Proc. Cambridge Phil. Soc.* **47**, 96 (1951).
- [31] E. K. Blum, *Math. Comput.* **16**, 176 (1962).
- [32] D. J. Fyfe, *Math. Comput.* **20**, 392 (1966).
- [33] D. I. Ketcheson, *SIAM J. Sci. Comput.* **30**, 2113 (2008).
- [34] M. H. Carpenter and C. A. Kennedy, *Fourth-order 2Nstorage Runge-Kutta schemes, Tech. Rep. NASA-TM-109112*, VA: National Aeronautics and Space Administration, NASA Langley Research Center, (1994).
- [35] C. A. Kennedy, M. H. Carpenter, and R. Lewis, *Appl. Numer. Math.* **35**, 177 (2000).
- [36] J. Butcher, *BIT Numer. Math.* **25**, 521 (1985).
- [37] J. Niegemann, R. Diehl, and K. Busch, *J. Comput. Phys.* **231**, 364 (2012).
- [38] T. Becker, H. Kredel, and V. Weispfenning, *Gröbner Bases: a Computational Approach to Commutative Algebra, Corrected Edn., Graduate Texts in Mathematics, Vol.141*, London, UK: Springer-Verlag, (1993).
- [39] D. Zingg and T. Chisholm, *Appl. Numer. Math.* **31**, 227 (1999).
- [40] J. A. Nelder and R. Mead, *Comput. J.* **7**, 308 (1965).
- [41] S. Johnson, *The Nlopt Nonlinear-Optimization Package*.
- [42] K. Levenberg, *Quart. J. Appl. Math.* **2**, 164 (1944).
- [43] D. W. Marquardt, *J. Soc. Indust. Appl. Math.* **11**, 431 (1963).
- [44] M. Lourakis, *Levmar: Levenberg-Marquardt Nonlinear Least Squares Algorithms in C/C++*, <http://www.ics.forth.gr/~lourakis/levmar/> July (2004).
- [45] E. Fehlberg, *Computing* **6**, 61 (1970).
- [46] M. Calvo, J. Montijano, and L. Randez, *Comput. Math. Appl.* **20**, 15 (1990).
- [47] P. Prince and J. Dormand, *J. Comput. Appl. Math.* **7**, 67 (1981).
- [48] C. W. Tang, *Appl. Phys. Lett.* **48**, 183 (1986).
- [49] H. Lee, Y. C. Cheng, and G. R. Fleming, *Science* **316**, 1462 (2007).
- [50] J. H. Burroughes, D. D. C. Bradley, A. R. Brown, R. N. Marks, K. Mackay, R. H. Friend, P. L. Burns, and A. B. Holmes, *Nature* **347**, 539 (1990).
- [51] G. Trinkunas, J. L. Herek, T. Polívka, V. Sundström, and T. Pullerits, *Phys. Rev. Lett.* **86**, 4167 (2001).
- [52] V. May and O. Kühn, *Charge and Energy Transfer Dynamics in Molecular Systems, 3rd Edn.* Weinheim: WILEY-VCH, (2010).
- [53] M. del Rey, A. W. Chin, S. F. Huelga, and M. B. Plenio, *J. Phys. Chem. Lett.* **4**, 903 (2013).
- [54] Z. Chen, V. Stepanenko, V. Dehm, P. Prins, L. Siebbeles, J. Seibt, P. Marquetand, V. Engel, and F. Würthner, *Chem. Eur. J.* **13**, 436 (2007).
- [55] X. Q. Li, X. Zhang, S. Ghosh, and F. Würthner, *Chem. Eur. J.* **14**, 8074 (2008).
- [56] M. R. Wasielewski, *Acc. Chem. Res.* **42**, 1910 (2009).
- [57] S. Wolter, J. Aizezers, F. Fennel, M. Seidel, F. Würthner, O. Kühn, and S. Lochbrunner, *New J. Phys.* **14**, 105027 (2012).
- [58] H. Marciniak, X. Q. Li, F. Würthner, and S. Lochbrunner, *J. Phys. Chem. A* **115**, 648 (2011).
- [59] D. Ambrosek, H. Marciniak, S. Lochbrunner, J. Tatchen, X. Li, F. Würthner, and O. Kühn, *Phys. Chem. Chem. Phys.* **13**, 17649 (2011).
- [60] D. Ambrosek, A. Köhn, J. Schulze, and O. Kühn, *J. Phys. Chem. A* **116**, 11451 (2012).
- [61] Q. Shi, L. Chen, G. Nan, R. X. Xu, and Y. Yan, *J. Chem. Phys.* **130**, 084105 (2009).
- [62] Y. A. Yan and O. Kühn, *New J. Phys.* **14**, 105004 (2012).
- [63] Y. A. Yan and S. Cai, *J. Chem. Phys.* **141**, 054105 (2014).
- [64] Y. Yan, *J. Chem. Phys.* **144**, 024305 (2016).
- [65] O. Kühn and V. Sundström, *J. Chem. Phys.* **107**, 4154 (1997).
- [66] H. Grabert, P. Schramm, and G. L. Ingold, *Phys. Rep.* **168**, 115 (1988).
- [67] J. Shao, *J. Chem. Phys.* **120**, 5053 (2004).
- [68] M. Schröter, S. Ivanov, J. Schulze, S. Polyutov, Y. Yan, T. Pullerits, and O. Kühn, *Phys. Rep.* **567**, 1 (2015).
- [69] Y. Yan and J. Shao, *Front. Phys.* **11**, 110309 (2016).
- [70] Y. A. Yan and Y. Zhou, *Hybrid Stochastic-Hierarchical Equations*. <http://nano.gznc.edu.cn/~yunan/hyshe.html> (2012).