The Journal of Chemical Physics

Direct quantum dynamics using variational Gaussian wavepackets and Gaussian process regression

Cite as: J. Chem. Phys. **150**, 041101 (2019); https://doi.org/10.1063/1.5086358
Submitted: 20 December 2018 . Accepted: 10 January 2019 . Published Online: 23 January 2019

Iakov Polyak, Gareth W. Richings, Scott Habershon, and Peter J. Knowles

The Journal of Chemical Physics COMMUNICATION scitation.org/journal/jcp

Direct quantum dynamics using variational Gaussian wavepackets and Gaussian process regression

Cite as: J. Chem. Phys. 150, 041101 (2019); doi: 10.1063/1.5086358
Submitted: 20 December 2018 • Accepted: 10 January 2019 •
Published Online: 23 January 2019

Iakov Polyak,[1,a] Gareth W. Richings,[2,b] Scott Habershon,[2] and Peter J. Knowles[1]

AFFILIATIONS

[1]School of Chemistry, Cardiff University, Main Building, Park Place, Cardiff CF10 3AT, United Kingdom
[2]Department of Chemistry and Centre for Scientific Computing, University of Warwick, Coventry CV4 7AL, United Kingdom

[a]Electronic mail: polyaki@cardiff.ac.uk
[b]Electronic mail: g.richings@warwick.ac.uk

ABSTRACT

The method of direct variational quantum nuclear dynamics in a basis of Gaussian wavepackets, combined with the potential energy surfaces fitted on-the-fly using Gaussian process regression, is described together with its implementation. Enabling exact and efficient analytic evaluation of Hamiltonian matrix elements, this approach allows for black-box quantum dynamics of multidimensional anharmonic molecular systems. Example calculations of intra-molecular proton transfer on the electronic ground state of salicylaldimine are provided, and future algorithmic improvements as well as the potential for multiple-state non-adiabatic dynamics are discussed.

Fitting molecular potential energy surfaces (PESs) with machine learning techniques is an increasingly popular approach, with the Gaussian Process Regression (GPR) method[1] recently demonstrated as a technique competitive with Artificial Neural Networks (ANNs) for general problems, due to its simplicity and robustness.[2–13] Several investigations performed for small molecular systems show that only a surprisingly modest number of data points are required to achieve high-quality PESs by means of GPR and that the scaling of their number is close to linear with the size of the system.[5,7,8,14] The two outstanding questions are whether this scaling is maintained for larger and more complex cases and developing optimal approaches for selecting training points.

A field where GPR could introduce a step change in capability, and which addresses the above two issues naturally, is direct quantum dynamics (QD), where PESs are evaluated on-the-fly, following the evolution of the time-dependent nuclear wavepacket. Until recently, the only fully variational method that exactly solved the time-dependent nuclear Schrödinger equation on-the-fly for general molecular systems was DD-vMCG (direct-dynamics variational multi-configurational Gaussian).[15] Due to the localized nature of the Gaussian wavepacket (GWP) basis functions employed in this method, a local harmonic approximation (LHA) was originally adopted to describe PESs, an approach having two drawbacks: (a) computationally expensive electronic gradients and Hessians need to be calculated at the center of each GWP at each time step of wavepacket propagation (with the overall expense being, however, reduced due to the usage of modified Shepard interpolation)[15] and (b) potential energy matrix elements are correct only to the second order, restricting the range over which interpolation can be trusted.

The first attempt to introduce machine-learned potentials into direct GWP-based QD was by Koch and Zhang,[16] who used multiplicative ANN to fit PESs on-the-fly, allowing for analytic potential energy matrix elements, and reported good performance and accuracy. Subsequently, Alborzpour *et al.* applied the Gaussian approximation potential (GAP) method to direct dynamics based on classically propagated GWPs and demonstrated improved accuracy in potential energy matrix elements compared to the LHA.[17] This method was based

on GPR with non-optimized hyper-parameters, which predictions for the PES fit can be shown to be equivalent to the kernel ridge regression (KRR) method as we also discuss below. Taking the simplicity of the GPR formalism and implementation, the authors suggested that the LHA should be abandoned altogether in the field of direct QD in its favor. Expanding on this prior work, Richings and Habershon have successfully applied the same technique to a conventional QD method [multi-configuration time-dependent Hartree (MCTDH)[18]] in a delocalized basis set defined on a grid.[19,20] This approach has been successfully demonstrated as an on-the-fly implementation of MCTDH which does not require pre-fitting of a global PES; applications to both ground-state and non-adiabatic dynamics have been presented, and further improvements to scalability and efficiency of their approach have also been recently reported.[21,22]

In this article, we fill the gap in the existing methodologies and describe the theory and implementation of the variational Gaussian-based direct QD method, DD-vMCG, using Gaussian-approximation potentials; we call the resulting approach GAP-vMCG. For full details on the DD-vMCG method, we refer the reader to Ref. 15. We would like, however, to stress here that this method follows the two coupled equations-of-motion (EOMs), one for the wavefunction expansion coefficients and the other for the parameters (position, momentum, and optionally width) of each Gaussian basis function, both derived from a variational principle without any approximations and is thus, in principle, exact.

The GAP method, used here, is an implementation of GPR using combinations of squared exponential functions, which in one dimension, q^m, are

$$k(q^m, q_k^m) = \exp\left[-\gamma(q^m - q_k^m)^2\right], \tag{1}$$

with γ defining the length-scale, as the kernel which is used to define the prior normal distribution in functional space and, effectively, to expand the PES. Matrix elements of the PES operator must be evaluated in the basis of GWPs, which in f dimensions are given by

$$g_i(\mathbf{q}, t) = \prod_{m=1}^{f} |g_i^m\rangle = \exp\left[\sum_{m=1}^{f} \varsigma_i^m (q^m)^2 + \xi_i^m q^m + \eta_i^m\right], \tag{2}$$

where $\{\boldsymbol{\varsigma}_i, \boldsymbol{\xi}_i, \boldsymbol{\eta}_i\}$ are time-dependent, complex parameters describing the exact form of the GWPs. In this work, we employ the frozen-width approximation so that only the $\boldsymbol{\xi}$ parameters are propagated in time [$\dot{\xi}_i^m = -2\varsigma_i^m q_i^m + i\dot{p}_i^m$ where ς_i^m determines the (negative) width, q_i^m determines the center, and p_i^m determines the momentum of the GWP along degree-of-freedom (DOF), m]. In order to evaluate the GAP matrix elements, we need the following three integrals along the mth DOF:

$$\langle g_i^m | g_j^m \rangle = \left(\frac{\pi}{-\varsigma_i^{*m} - \varsigma_j^m}\right)^{1/2} \exp\left[\frac{(\xi_i^{*m} + \xi_j^m)^2}{-4(\varsigma_i^{*m} + \varsigma_j^m)} + \eta_i^{*m} + \eta_j^m\right], \tag{3a}$$

$$\langle g_i^m | k(q^m, q_k^m) | g_j^m \rangle = \left(\frac{\pi}{-\varsigma_i^{*m} - \varsigma_j^m + \gamma}\right)^{1/2} \exp\left[\frac{(\xi_i^{*m} + \xi_j^m + 2\gamma q_k^m)^2}{4(-\varsigma_i^{*m} - \varsigma_j^m + \gamma)} - \gamma(q_k^m)^2 + \eta_i^{*m} + \eta_j^m\right], \tag{3b}$$

$$\langle \frac{\partial g_i^m}{\partial \xi_i^m} | k(q^m, q_k^m) | g_j^m \rangle = \frac{\xi_i^{*m} + \xi_j^m + 2\gamma q_k^m}{2(-\varsigma_i^{*m} - \varsigma_j^m + \gamma)} \langle g_i^m | k(q^m, q_k^m) | g_j^m \rangle. \tag{3c}$$

In applying GPR to fit the f-dimensional PES, the most general way to represent an f-dimensional kernel is to use a many-body expansion in terms of the one-dimensional kernels $k(q^m, q_k^m)$,

$$k^{\mathrm{G}}(\mathbf{q}, \mathbf{q}_k) = \kappa^2\Big(\sum_{m=1}^{f} k(q^m, q_k^m) + \sum_{m<n}^{f} k(q^m, q_k^m) k(q^n, q_k^n) + \cdots + \prod_{m=1}^{f} k(q^m, q_k^m)\Big), \tag{4}$$

with κ^2 being a variable prefactor. However, since the large number of terms in this expansion leads to a significant computational effort in the evaluation of the PES matrix elements, we use two truncations of the general kernel in this work (both of which are valid in terms of GPR). The first simply retains the final, f-dimensional term in Eq. (4) and is termed the full kernel, $k^{\mathrm{Full}}(\mathbf{q}, \mathbf{q}_k)$, whilst the second keeps the first two terms and is called the additive kernel, $k^{\mathrm{Add}}(\mathbf{q}, \mathbf{q}_k)$.

Whatever the exact form of the kernel [using $k(\mathbf{q}, \mathbf{q}_k)$ to represent any of those described above], it can be used in GPR as a covariance function that defines the prior joint normal distribution of the values of the function to be learned (here we use the general notation $b \sim \mathcal{N}(\mu, \sigma^2)$ to refer to a function value b distributed normally with the mean μ and variance σ^2). In other words, it defines its smoothness. Suppose we have a vector of function values $\mathbf{b}$ and another function value b_0. In GPR, they will have a prior joint normal distribution according to

$$\begin{pmatrix} \mathbf{b} \\ \mathrm{b}_0 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu} \\ \mu_0 \end{pmatrix}, \begin{pmatrix} \mathbf{A} & \mathbf{k}^{\mathrm{T}} \\ \mathbf{k} & 1 \end{pmatrix}\right), \tag{5}$$

where matrix $\mathbf{A}$ is a covariance function with elements

$$\mathrm{A}_{ij} = k(\mathbf{q}_i, \mathbf{q}_j) + \lambda^2 \delta_{ij}, \tag{6}$$

(with λ^2 being a small regularization parameter), and the elements of the vector, $\mathbf{k}$, are the values of the appropriate kernel evaluated at points $\{\mathbf{q}_0\}$ corresponding to the function value b_0: $k(\mathbf{q}_i, \mathbf{q}_0)$. The conditional (posterior) distribution of b_0, with the function values $\mathbf{b}$ known, can be shown to be

$$\mathrm{b}_0 \sim \mathcal{N}(\mu_0, \sigma_0^2), \tag{7}$$

with the mean given by

$$\mu_0(\mathbf{q}) = \mathbf{k}^{\mathrm{T}} \mathbf{A}^{-1} \mathbf{b}, \tag{8}$$

and the variance being

$$\sigma_0^2(\mathbf{q}) = (k(\mathbf{q}, \mathbf{q}) + \lambda^2) - \mathbf{k}^{\mathrm{T}} \mathbf{A}^{-1} \mathbf{k}. \tag{9}$$

The PES is defined as the mean of the posterior normal distribution [Eq. (8)], and it is expanded as a linear combination of kernel functions centered at a selection of points in configuration space,

$$\mathcal{V}(\mathbf{q}) \simeq \sum_{k=1}^{M} \omega_k k(\mathbf{q}, \mathbf{q}_k), \tag{10}$$

where the weights $\{\omega_k\}$ are found by solving

$$\mathbf{A}\boldsymbol{\omega} = \mathbf{b}, \tag{11}$$

with $\mathbf{b}$ containing the actual values of the PES at assorted points in configuration space, $b_k = \mathcal{V}(\mathbf{q}_k)$.

Given the integrals in Eq. (3) and the form of the PES in Eq. (10), it is straightforward to expand the full matrix elements needed for GAP-vMCG. For completeness, the full expressions are given in the supplementary material.

It is worth pointing out that the KRR method can be formulated as an equivalent approach to fit a function given a fixed kernel,[4] but it can only provide a prediction [such as in Eq. (8)], while GPR learns a probabilistic model of the target function, providing also a variance for each function value, knowledge of which forms an integral part of our algorithm for generating test points on-the-fly as described below. Furthermore, GPR provides a direct mean for optimizing kernel hyperparameters based on the gradient-ascent on the marginal likelihood function,[1] but the capability of doing so does not define it as a method, and we have not pursued this further in the current work.

To test the GAP-vMCG approach, we have implemented the method in a development version of the Quantics package.[23] The propagation of the vMCG wavepacket was performed as described previously,[15] except that the potential energy matrix elements are evaluated using integrals defined in Eq. (3) (with full expressions given in the supplementary material), rather than employing the LHA when constructing the EOMs. The other, new feature of this implementation concerns the sampling of configuration space so as to generate the GAP training set by evaluating the electronic energy at appropriate molecular geometries.

In the original DD-vMCG implementation, electronic energies (and gradients and Hessians) are evaluated at geometries visited by the GWPs; at each step in the propagation, the algorithm compares the location of the center of each GWP in turn to the geometries previously sampled (which are stored in a database) and if the minimum distance is larger than a pre-defined parameter, a new set of electronic data is calculated at the geometry in question, added to the database, and then used in all subsequent representations of the global PES. The GAP implementation retains this link between the path of the GWPs and the regions of configuration space sampled as it makes sense to have the PES most accurately represented in the vicinity of the GWPs but with notable differences.

The main difference between the old and new sampling methods is that the distance measure comparing GWP centers to entries in the database is no longer required; the need to add another point to the database is determined by a variance measure which is defined in the GPR theory. If the variance at a test point $\mathbf{q}$, defined in Eq. (9), is greater than a user-defined value (in this work 10^{-3} a.u.), the fit to the PES using the original training set at that point is considered to have insufficient accuracy and a new electronic energy is calculated there and added to the training set database; otherwise the point is rejected. This same approach has been used in the previously reported direct MCTDH simulations.[19–22]

Test points are chosen as follows: beginning with an empty database, the center of the initial wavepacket (along with its electronic energy) is taken as the first member in the training set. Subsequently, the first GWP is taken and the location of its center is used as a test point; the variance is evaluated, and a new energy is added to the database if required. A fixed number of points (in this work 50) are then quasi-randomly chosen around the center of this GWP following Sobol sampling in the multi-dimensional space within a pre-defined (in this work 3) multiple of the GWP width of the GWP center (see Ref. 22 for more details on sampling). Each of these points is then tested for addition to the trial set before the process is repeated for the remaining GWPs. Once all of the GWPs have been subjected to this sampling, Eq. (11) is solved and used to form the potential in Eq. (10). The potential energy matrix elements can then be formed, allowing construction of the EOMs, and the wavepacket is allowed to propagate for a given period of time (typically 1 fs) on this PES. After this period of propagation, the sampling of configuration space around the GWP centers is repeated to allow an updated PES to be constructed. By doing so the PES is most densely sampled in those regions of configuration space visited by the GWPs, while areas which they avoid are unsampled. It should be noted that whenever a new trial point is added to the database, matrix $\mathbf{A}$ is updated, together with its Cholesky decomposition which is used in the solution of Eqs. (11) and (9), thus ensuring that the subsequent sampling uses the most up to date variance measure when determining whether to add a point or not.

For test calculations, we have chosen a ground-state intra-molecular proton transfer reaction in salicylaldimine, using the PES fitted previously,[24] and the same initial conditions as in our earlier (DD)-vMCG[24] and DD-MCTDH[21] calculations. As in previous work, we use the expectation value of the flux operator defined at the barrier along the proton-transfer mode ν_1 as a highly sensitive observable and compare the GAP-vMCG results to the 4th-order vMCG results, being exact on the PES fitted to the same order. The hyperparameters have been manually adjusted to deliver approximately best performance of the GPR fit in terms of the required number of data points accumulated on-the-fly while maintaining high accuracy of the results (see supplementary material for the details). Parameters γ and λ are not expected to have different optimal values whether $k^{\mathrm{Full}}(\mathbf{q}, \mathbf{q}_k)$ or $k^{\mathrm{Add}}(\mathbf{q}, \mathbf{q}_k)$ is in use, and once adjusted values of 0.3 and 10^{-6}, correspondingly, have been used throughout all calculations. On the other hand, the optimal κ was found to depend on the kernel in use, while having an overall smaller effect on method efficiency, and was therefore set to 1.0.

The number of accumulated training points, M, is a crucial factor for any GPR implementation, which generally scales as $\mathcal{O}(M^3)$.[1] In our current implementation, due to the lack of parameter optimization and the use of updates of the Cholesky-decomposed matrix, **A**, the vector-matrix multiplication in Eq. (9) becomes the computational bottleneck of the GPR part, i.e. less than $\mathcal{O}(M^2)$. For GAP-vMCG as a whole, when using the additive kernel (see below) for the current test case, the evaluation of potential energy matrix elements becomes the bottleneck (scaling as $\mathcal{O}(Mm^2n^2)$ where m is the system dimensionality and n is the number of GWPs). The number of accumulated training points is still, however, of vital importance to the method's efficiency, especially since, in the absence of a pre-fitted PES, *ab initio* electronic energies would need to be evaluated on-the-fly at every point.

After running some preliminary test calculations, we realized that using $k^{\mathrm{Full}}(\mathbf{q}, \mathbf{q}_k)$ leads to accumulation of a very large number of training points (with a 100 fs, 6D dynamics run using adjusted κ = 0.75 resulting in more than 44 000 points and a 13D calculation becoming nearly infeasible memory- and time-wise). Introduction of $k^{\mathrm{Add}}(\mathbf{q}, \mathbf{q}_k)$ led to a dramatic reduction in the number of training points necessary to obtain a converged GPR fit for PES. This is in agreement with Ref. 21 where a poor span of the multi-dimensional space by a multi-dimensional Gaussian has been discussed in detail. Already the first (starting from an empty database) 6D run using 64 GWPs and default hyper-parameters resulted in the nearly converged flux (see Fig. 1) while accumulating only 1884 points. The same-length "from scratch" 13D run employing 96 GWPs also gave a qualitatively correct flux (see Fig. 2) with 8068 points evaluated on-the-fly. Both of the above calculations have been performed using 40 cores (Intel Xeon Gold 6148 × 2) and took 1 h 20 min and 14 h 15 min, respectively (for comparison, vMCG calculations took 8 min and 1 h 51 min, respectively). Adjustment of the κ-values to 0.3 (6D) and 0.25 (13D) led to a further reduction in the accumulated points down to 1280 and 6120, respectively, while not altering the quality of the flux significantly. Obviously, proper optimization of hyper-parameters, as commonly employed in GPR, while adding to the overall time of computation, is expected to further decrease the number of points necessary for an accurate fit.

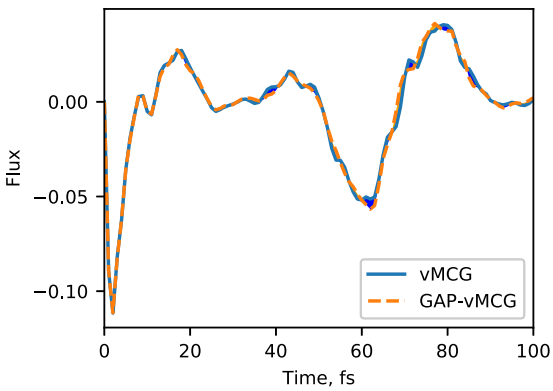

FIG. 1. Flux through the potential barrier for the proton transfer isomerisation of salicylaldimine in a 6D model using 64 GWPs, calculated with vMCG (blue solid line) and GAP-vMCG (orange dashed line) using $k^{\mathrm{Add}}(\mathbf{q}, \mathbf{q}_k)$ and κ = 1.0. The space between the two flux curves is filled with the color of the higher-lying curve.

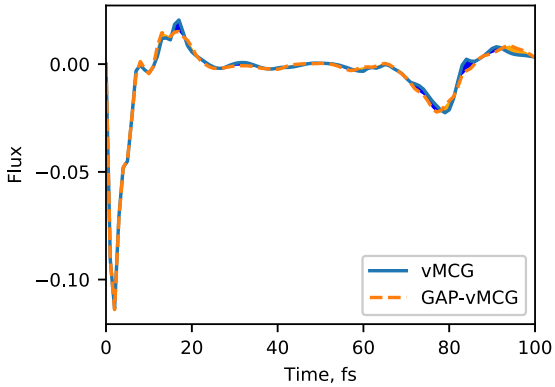

FIG. 2. Flux through the potential barrier for the proton transfer isomerisation of salicylaldimine in a 13D model using 96 GWPs, calculated with vMCG (blue solid line) and GAP-vMCG (orange dashed line) using $k^{\mathrm{Add}}(\mathbf{q}, \mathbf{q}_k)$ and κ = 1.0. The space between the two flux curves is filled with the color of the higher-lying curve.

The results presented above promise that exact (to an accuracy of at least the inherent error of the quantum chemical method used to obtain the PES) fully quantum multi-dimensional dynamics is possible for anharmonic molecular systems using the GAP-vMCG method. Even using the existing code, systems as large as 30-40 DOF should be feasible to study. One should note that the calculations performed in terms of the current work made use of the pre-fitted PES and therefore the computational time spent excluded quantum chemistry calculations themselves. Still, even tens of thousands of *ab initio* calculations are quite feasible with modern, scalable codes and ever-improving HPC hardware. Improvements are, however, sought to the GAP-vMCG algorithm to make it even more efficient, among which are the implementation of the proper automatic optimization of hyper-parameters, individual instead of unique length-scale parameters for different DOF, other covariance functions such as Matérn kernel (which might be more advantageous for some systems) and improved sampling algorithms.

The main focus of quantum dynamics calculations is usually non-adiabatic molecular systems, where different electronic states may couple to each other via nuclear distortion, thereby violating the Born-Oppenheimer approximation. Evolution of the nuclear wavepacket in a manifold of coupled electronic states is a common application both for conventional MCTDH and (DD-)vMCG methods. To achieve stable propagation and avoid discontinuities in the gradients of adiabatic surfaces and the non-adiabatic couplings at conical intersections, it is a common practice to perform a transformation to (quasi)-diabatic surfaces that cross smoothly. An

on-the-fly method of propagation diabatisation has been successfully implemented both for DD-vMCG and DD-MCTDH methods,[20,25,26] with the latter including modification allowing for the use of GPR-learned surfaces,[20,21] whilst, recently, a new projection diabatisation scheme was introduced[27] which has already been successfully coupled to the DD-MCTDH algorithm.[22] The same two approaches should be straightforward to use for GAP-vMCG, allowing for larger-scale but accurate calculations of non-adiabatic processes.

See supplementary material for the derivation and full expressions for the potential energy operator matrix elements as well as details on manual adjustment of kernel parameters used in the current work.

Iakov Polyak acknowledges the support of the Supercomputing Wales project, which is part-funded by the European Regional Development Fund (ERDF) via Welsh Government. Gareth Richings and Scott Habershon gratefully acknowledge the Leverhulme Trust for funding (No. RPG-2016-055) and the Scientific Computing Research Technology Platform at the University of Warwick for providing computational resources.

REFERENCES

[1]C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning* (The MIT Press, Cambridge, Massachusetts, 2006).
[2]A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, Phys. Rev. Lett. **104**, 136403 (2010).
[3]S. Manzhos, R. Dawes, and T. Carrington, Int. J. Quantum Chem. **115**, 1012 (2015).
[4]A. P. Bartok and G. Csanyi, Int. J. Quantum Chem. **115**, 1051 (2015).
[5]J. Cui and R. V. Krems, J. Phys. B: At., Mol. Opt. Phys. **49**, 224001 (2016).
[6]O.-P. Koistinen, E. Maras, A. Vehtari, and H. Jónsson, Nanosyst.: Phys., Chem., Math. **7**, 925 (2016); e-print arXiv:1703.10423.
[7]B. Kolb, P. Marshall, B. Zhao, B. Jiang, and H. Guo, J. Phys. Chem. A **121**, 2552 (2017).
[8]A. Kamath, R. A. Vargas-Hernández, R. V. Krems, T. Carrington, and S. Manzhos, J. Chem. Phys. **148**, 241702 (2018).
[9]T. T. Nguyen, E. Székely, G. Imbalzano, J. Behler, G. Csányi, M. Ceriotti, A. W. Götz, and F. Paesani, J. Chem. Phys. **148**, 241725 (2018).
[10]G. Schmitz and O. Christiansen, J. Chem. Phys. **148**, 241704 (2018).
[11]C. Qu, Q. Yu, B. L. Van Hoozen, J. M. Bowman, and R. A. Vargas-Hernandez, J. Chem. Theory Comput. **14**, 3381 (2018).
[12]G. Laude, D. Calderini, D. P. Tew, and J. O. Richardson, Faraday Discuss. **212**, 237 (2018); e-print arXiv:1805.02589.
[13]Y. Guan, S. Yang, and D. H. Zhang, Mol. Phys. **116**, 823 (2018).
[14]J. L. Loeppky, J. Sacks, and W. J. Welch, Technometrics **51**, 366 (2009).
[15]G. W. Richings, I. Polyak, K. E. Spinlove, G. A. Worth, I. Burghardt, and B. Lasorne, Int. Rev. Phys. Chem. **34**, 269 (2015).
[16]W. Koch and D. H. Zhang, J. Chem. Phys. **141**, 021101 (2014).
[17]J. P. Alborzpour, D. P. Tew, and S. Habershon, J. Chem. Phys. **145**, 174112 (2016).
[18]M. H. Beck, A. Jäckle, G. A. Worth, and H.-D. Meyer, Phys. Rep. **324**, 1 (2000).
[19]G. W. Richings and S. Habershon, J. Chem. Theory Comput. **13**, 4012 (2017).
[20]G. W. Richings and S. Habershon, Chem. Phys. Lett. **683**, 228 (2017).
[21]G. W. Richings and S. Habershon, J. Chem. Phys. **148**, 134116 (2018).
[22]G. W. Richings, C. Robertson, and S. Habershon, "Improved on-the-Fly MCTDH Simulations with Many-Body-Potential Tensor Decomposition and Projection Diabatization," J. Chem. Theory Comput. (to be published).
[23]G. A. Worth, K. Giri, G. W. Richings, I. Burghardt, M. H. Beck, A. Jäckle, and H.-D. Meyer, The Quantics Package, Version 1.1, 2015.
[24]I. Polyak, C. S. M. Allan, and G. A. Worth, J. Chem. Phys. **143**, 084121 (2015).
[25]G. W. Richings and G. A. Worth, J. Phys. Chem. A **119**, 12457 (2015).
[26]G. W. Richings and G. A. Worth, Chem. Phys. Lett. **683**, 606 (2017).
[27]C. Robertson, J. Gonzalez-Vazquez, S. Diaz-Tendero, I. Corral, and C. Diaz, J. Comput. Chem. **40**, 794 (2019).