Quantum-mechanical wavepacket propagation in a sparse, adaptive basis of interpolating Gaussians with collocation

J. Sielk,[a] H. F. von Horsten,[a] F. Krüger,[b] R. Schneider[b] and B. Hartke*[a]

Received 19th August 2008, Accepted 30th September 2008
First published as an Advance Article on the web 17th November 2008
DOI: 10.1039/b814315c

We present an extension of our earlier work on adaptive quantum wavepacket dynamics [B. Hartke, *Phys. Chem. Chem. Phys.*, 2006, **8**, 3627]. In this dynamically pruned basis representation the wavepacket is only stored at places where it has non-negligible contributions. Here we enhance the former 1D proof-of-principle implementation to higher dimensions and optimize it by a new basis set, interpolating Gaussians with collocation. As a further improvement the TNUM approach from Lauvergnat and Nauts [*J. Chem. Phys.*, 2002, **116**, 8560] was implemented, which in combination with our adaptive representation offers the possibility of calculating the whole Hamiltonian on-the-fly. For a two-dimensional artificial benchmark and a three-dimensional real-life test case, we show that a sparse matrix implementation of this approach saves memory compared to traditional basis representations and comes even close to the efficiency of the fast Fourier transform method. Thus we arrive at a quantum wavepacket dynamics implementation featuring several important black-box characteristics: it can treat arbitrary systems without code changes, it calculates the kinetic and potential part of the Hamiltonian on-the-fly, and it employs a basis that is automatically optimized for the ongoing wavepacket dynamics.

I. Introduction

Exact quantum wavepacket dynamics remains an important tool in chemical reaction dynamics. Quantum corrections to classical methods are being pursued but are not possible for all non-classical phenomena yet; in fact, many semiclassical approaches are often good only as short-time approximations. Hence quantum dynamics are still indispensable where quantum effects are important. Also, quantum wavepacket dynamics offers a natural description without approximations for dynamics on multiple coupled electronic states (photochemistry).

In traditional approaches, however, quantum wavepacket dynamics is hit very quickly by the curse of dimensionality: every standard representation of the wavepacket scales exponentially with the number of degrees of freedom. For this reason, fully exact quantum wavepacket dynamics of chemical reactions has been essentially limited to 4-atomic systems for more than 10 years now, despite all progress in computer hardware in the meantime.

A prominent remedy in this situation has been the multi-configuration time-dependent Hartree (MCTDH) method.[1–3] It expands the wavefunction into a series of dynamically adapted product representations (boiling down the propagation to many 1D ones), which can be cut off early for weakly coupled degrees of freedom. This makes it possible to do quantum wavepacket dynamics for a few dozen degrees of freedom[4–6] (with cascading or multilayer MCTDH[3,7,8] even for many dozens). However, the approach is not without problems. Both, the kinetic and the potential part of the Hamiltonian have to be separable into 1D terms, which is not necessarily possible in both cases. In its original form, the algorithm does not beat the exponential scaling;[3] cascading MCTDH offers some potential to partially alleviate it, but it is hard to implement in generality. Finally for strongly coupled degrees of freedom it can be more efficient to do parts of the full problem in more than 1D (sometimes significantly more).

Therefore, both for other authors and for us, it still is a legitimate purpose to develop wavepacket propagation methods for many strongly coupled degrees of freedom, not to supplant MCTDH but as alternative approaches, and possibly even for use within it. This is not the place to give a comprehensive overview of all developments in this direction, therefore we mention just a few characteristic works. Grid representations "actively" following the moving wavepacket around have been suggested already at early stages of the development of the fast Fourier transform (FFT) method.[9] More recently, such moving grids have been implemented by Wyatt and co-workers,[10,11] borrowing ideas from computational fluid dynamics. Instead of letting grid points move, it is also possible to let traditional basis functions follow the wavepacket around, as for example realized in the multiple spawning method[12] or in the Gaussian wavepacket propagation scheme.[13] While these methods offer efficient propagations through their classical-like equations of motion for the Gaussian parameters, they carry the danger of under/overcompleteness of these moving Gaussians. Also, with arbitrary and non-predetermined basis function locations, it is challenging to establish efficient interpolation schemes for on-the-fly dynamics to reduce the number of *ab initio* calculations. Both aspects become very easy in the scheme we propose

[a] *Institut für Physikalische Chemie, Christian-Albrechts-Universität, Olshausenstraße 40, 24098 Kiel, Germany. E-mail: hartke@phc.uni-kiel.de*
[b] *Institut für Mathematik, Technische Universität Berlin, Sekretariat MA 8-1, Straße des 17. Juni 136, 10623 Berlin, Germany*

here, due to a static basis-function grid at a fixed, optimal resolution. Given the tight relations between basis and grid representations *via* the discrete variable representation (DVR), it is not surprising that there is also a time-dependent DVR,[14,15] which additionally offers a "smooth" interpolation between (semi-)classical dynamics and the fully quantum case. Also related to moving grids are "quantum trajectories",[16–18] as modern realizations of Bohmian mechanics. The aspect of non-standard basis functions or grids in quantum wavepacket dynamics, of course, also has been an issue since many years; recently, for example, distributed approximating functionals[19] have been used, as well as sparse grids.[20]

Two other central ideas of reducing the size of a basis or grid representation are contraction and pruning. In both cases, a large direct-product basis is set up at first. In pruning, this primitive set is then made smaller by discarding basis functions or grid points which will not contribute significantly to the desired solution wavefunction(s). Obviously, the difficulty is to decide *a priori* what can be discarded, before having done the full calculation. A simple pruning strategy discards grid points in regions of high potential energy.[21] However, while standard DVR basis functions are strictly local on the DVR grid points, they are not local in the corresponding finite-basis representation (FBR); therefore, from the FBR viewpoint one cannot discard any single function by the potential energy criterion. Indeed, it has been pointed out that the non-local "tails" of DVR functions can seriously damage this simple pruning concept (see the discussion and references in ref. 22). Another serious problem for application of this strategy in higher-dimensional examples is that it needs pre-calculation of the potential over the whole region of interest, explicitly including the areas that can then be discarded as being too high in potential energy. This is clearly unfeasible in strongly coupled high-dimensional cases. Therefore, pruning a primitive basis is less straightforward than it may seem at first. Contraction schemes exploit the fact that pruning is comparatively trivial after the primitive basis has been applied to solve lower-dimensional subproblems of the full-dimensional problem, since by construction the resulting lower-dimensional eigenfunctions are focused on the region of interest after a simple total energy cutoff of the eigenvalue spectrum.[23] The downside of contraction approaches is that the quality of the pruned lower-dimensional eigenfunctions as basis for the full-dimensional problem is strongly system-dependent (choice of coordinates, coupling strengths, *etc.*), hence it is hard to turn this approach into a black-box method. Recently, Dawes and Carrington[22,24,25] have used simultaneous diagonalization to construct DVRs corresponding to pruned, contracted basis functions; this appears to be a promising strategy for time-independent problems.

In a previous publication,[26] we presented our proDG approach for time-dependent problems, which follows a different but very simple idea: a primitive basis of spatially local basis functions is pruned down to a minimal size that automatically adapts to the movements of the wavepacket, by simple fixed threshold criterions and without any user-intervention or preparative steps. The basic strategy is as follows: a fixed grid is defined at the start. On its nodes, localized basis functions (*e.g.* Gaussians) are positioned. However, these basis functions exist only where they contribute non-negligibly to the wavefunction. Two fixed threshold values allow one to delete basis functions with negligible contributions, and to introduce new ones next to basis functions whose contributions increase, as the wavepacket is propagated. This establishes a compact representation that "moves" with the wavepacket, although all basis functions are stationary. Thus, we gain the advantages of a moving grid without its disadvantages (*i.e.* additional equations of motion for the basis functions or grid points, danger of overcompleteness or "holes" in the representation, *etc.*). Additionally, the locality of the primitive basis minimizes pruning errors (in contrast to standard DVRs, as mentioned above). Also, there is no need for system-dependent preparative steps or any other user intervention, nor a need for costly precalculations of potential energy values, since in a very direct sense the moving wavepacket itself determines the presence or absence of basis functions at each point in time.

In ref. 26, a first, simple implementation of this idea was shown to be working in one dimension (1D) in all typical situations to be encountered in quantum wavepacket dynamics, *i.e.*, compact and spreading wavepackets, splitting of the wavepacket in several pieces, re-joining of such wavepacket pieces, and even tunneling through a barrier. In all cases, our automatically adjusting representation remained compact and followed the moving wavepacket location, governed by a minimal set of simple parameters, without case-specific adjustments. We should point out that at the same time but independent of our work McCormack[27] demonstrated possible savings of similar ways of grid/basis pruning, but apparently without having an actually working implementation.

An additional important feature of this approach is that it allows for a simple realization of on-the-fly calculation of the potential in quantum wavepacket dynamics. With non-local basis functions, or with local basis functions (or grids) but without any dynamic adaptivity, there is no way to avoid precalculation of the potential everywhere in the whole region covered by the basis. With moving (rectangular) grids one has to add many points at each enlargement operation, many of them *a posteriori* superfluous. Only basis functions that move around come closer to allowing economic on the fly potential calculations, but then it is still unclear when and where to do how many new potential evaluations. With our scheme, this is always perfectly clear, and it is always just at a few well-defined points at pre-defined, equidistant locations, which is ideal for combining it with efficient interpolation mechanisms.

With the work presented here, we are not only extending our 1D proof-of-principle implementation of ref. 26 to an arbitrary number of dimensions, showing a 2D example and a 3D real-life example, but we have also included further improvements: a highly efficient basis set of interpolating Gaussians, collocation evaluation of the integrals, and a sparse-matrix implementation. As we will show, all this leads to considerable additional savings, beyond the basic idea of a dynamically adaptive representation, offering perspectives for extension of full quantum wavepacket dynamics to higher-dimensional cases. Furthermore, we take advantage of a numerical method by Lauvergnat *et al.*[13,28] to calculate the kinetic part of the Hamiltonian. This software (Tnum) is perfectly suited for our adaptive approach since the sometimes

awkward derivation of analytical expressions of the kinetic energy operator in curvilinear coordinates becomes superfluous. Instead, the values of all necessary kinetic functions are computed numerically but exactly for a given (arbitrary) molecular geometry, and this is just what is needed if a new (localized) basis function is introduced. In essence, the combination of TNUM and our adaptive approach provides the possibility to calculate not only the potential but also the kinetic part of the Hamiltonian on-the-fly.

The remainder of this article is organized as follows: section II presents the relevant theoretical ingredients in the necessary details. In section III, representative results for 2D and 3D cases are shown, including a discussion of the savings incurred by the various features of our approach. Section IV summarizes the most important points of this work.

II. Theoretical background

In this work we enhance our adaptive finite basis wavepacket representation (FBR) approach, called proDG,[26] to higher dimensionality and merge it with the collocation method, using a basis of interpolating Gaussians. To explain our extension to higher dimensionality, we first give a brief review of the one-dimensional approach. Later we introduce the terms collocation and interpolating Gaussians.

A Adaptive basis

In a regular FBR approach the wavefunction $\Psi(t)$ at a given time t is expanded in a set of non-orthogonal, time-independent basis functions $\{\varphi_i\}$,

$$|\Psi(t)\rangle = \sum_i c_i(t)|\varphi_i\rangle. \quad (1)$$

Insertion of this expansion into the time-dependent Schrödinger equation and operating from the left with $\langle\varphi_i|$ yields a matrix–vector equation

$$\mathrm{i}\hbar \boldsymbol{S}\frac{\partial}{\partial t}\boldsymbol{c}(t) = \boldsymbol{H}\boldsymbol{c}(t), \quad (2)$$

with the usual overlap and Hamilton matrices, from which standard propagation schemes can be derived (see ref. 29).

In the standard case there are no general restrictions for the choice of the basis functions. While they usually are normalized, they do not necessarily need to be orthogonal ($\langle\varphi_i|\varphi_j\rangle \neq \delta_{ij}$) or localized. So a variety of basis functions, such as sines, polynomials or Gaussians can be utilized. For our proDG approach some restrictions have to be applied. The basic idea of the algorithm is valid only if the basis functions are local to some extent and not delocalized over the whole grid. This inhibits the use of sine functions or polynomials. As shown in ref. 26, a distributed Gaussian basis (DGB), as popularized by Hamilton and Light,[30] is a good first choice.

To achieve a minimal representation of the wavefunction in the proDG approach, expansion coefficients [eqn (1)] and the corresponding basis functions are only stored at places where they have a non-negligible contribution to the total wavefunction (these are the "active" coefficients or basis functions). All possible places for the centers of basis functions are predefined due to the equal spacing of a fictitious grid, which is in principle infinite in all dimensions. This grid will be referred to as basis function grid. Before each time step, the magnitudes of the coefficients are checked. According to the results of this check, the set of basis functions is dynamically adjusted to the present form of the wavepacket, using these two simple rules:

1. If the absolute value of an active coefficient raises above a predefined threshold value θ_a, additional basis functions are introduced at the immediately neighboring basis function grid point(s). Their coefficient values are initialized as zero.

2. If the absolute value of an active coefficient decreases below another predefined threshold value θ_d, this basis function and the corresponding Hamiltonian matrix elements are deleted (with certain restrictions, which are described below).

Due to the continuous checking of the coefficients, an apparently moving grid with spatially fixed basis functions is established. Because the deleted coefficients are not just passive but physically "not there", *i.e.* the Hamiltonian matrix (and possibly other arrays containing basis functions) are resized accordingly, we save considerable amounts of memory compared to the FBR direct product approach. It should also be possible to treat periodic coordinates (*e.g.* dihedral angles) in this fashion, by adding the periodicity requirement directly to the extension/pruning operations.

Extending the approach described above to higher dimensions is straightforward: The basis function grid is equally spaced in all dimensions (although allowing different spacing per dimension), but again it is never fully filled but only predefines basis function locations. For a coefficient raising above θ_a, now a maximum of two basis functions *per dimension* is added to the basis function grid, provided they did not exist before. To decrease the computational overhead, a few constraints were set up in addition to the two basic rules given above. For a coefficient which already has two active neighbors in each dimension (a "bulk" coefficient), no basis enlargement can be induced anymore. Therefore, we store this information and skip the coefficient check for these bulk coefficients. Moreover, when a coefficient drops below θ_d, before deleting it from the set of active basis functions the magnitude of all direct neighbors is checked. If one of the neighbors is larger than θ_a, we skip the deletion, since this coefficient would be re-introduced immediately by this neighbor as the coefficient check progresses further.

Further advantages arise from the fact that due to the sparse wavepacket representation the Hamiltonian and therefore also the potential energy surface only have to be calculated at the active points. This saves computational time, and it also opens up a simple but efficient route towards combining quantum wavepacket propagation with on the fly calculation of the potential. In classical mechanics, at each time step the trajectory dictates in a simple way at which point to calculate the potential for the next time step. In quantum mechanics, position is not a function of time but also an independent variable. Hence, in a very basic sense, it is quite unclear where to calculate the potential for the next time step. Our present scheme, however, provides just this, *i.e.* an unambiguous prescription of which new points will become "active" at the next time step, and hence exactly where the potential is needed. For the application examples shown in this work, the potential

energy surfaces were given analytically (either directly or as fits to calculated data) and evaluated pointwise, as the coefficients to the according coordinates become active. This already is the simplest possible realization of an on-the-fly potential calculation. For the case of on-the-fly *ab initio* calculations, we expect this simple scheme of calculating the potential at every point to be too expensive. Hence, we are currently working towards implementing efficient interpolation strategies.

The obvious price to pay for all these advantages is the overhead due to the inspection of the coefficients, but this is tolerable compared to the computational effort needed for the propagation itself: Whatever the propagation algorithm, each coefficient has to be handled at least once in each time step anyway, to calculate the "new" coefficients from the "old" ones.

A less obvious but more costly disadvantage is that the ability to resize the Hamiltonian matrix (and possibly also other matrices and arrays numbered by basis functions) to the actually needed size at each time step sounds more attractive in theory than in practical applications: in standard computer languages, simple matrix formats are efficient for the needed matrix–vector multiplications but adding/deleting rows and columns from them incurs an overhead in computer time and memory. The alternative of linked lists (or similar data formats) decreases the latter overhead but renders matrix–vector operations more inefficient. In current work, we are trying to establish an implementation that is a best compromise between these two extremes.

B Variants of Gaussian basis functions

The initial implementation of our proDG approach employed a regular FBR scheme. As pointed out earlier, the local character of the method itself limits the choice of basis functions to localized ones, for example standard Gaussians. A disadvantage of these basis functions is that they are non-orthogonal. Hence the overlap matrix $\boldsymbol{S}$ in eqn (2) has to be carried through the propagation and needs to be re-inverted after each basis modification, which is a big numerical effort. Additionally, a numerical integration grid is needed to evaluate the Hamiltonian matrix elements. It has to be much denser than the basis function grid to ensure numerical stability of the propagation. Adaption of this numerical integration grid to the changing basis incurs further computational overhead.

As we show in this work, it is possible to overcome these disadvantages. Distributed Gaussian basis functions can be orthonormalized, which yields an overlap matrix $\boldsymbol{S}$ that is just the unity matrix. Additionally, we employ so-called interpolating Gaussians that are optimized for usage with the collocation method (see below). They are forced to be zero on all (collocation) grid points, except where their center is located.

Let us first remind the reader of the usual distributed Gaussians in one (spatial) dimension x:

$$\tilde{\phi}_i^h(x) = \mathrm{e}^{-\frac{(h^{-1}x-i)^2}{D}}. \tag{3}$$

Furthermore, let V_h denote the space of linear combinations of $\tilde{\phi}_i^h$:

$$V_h = \left\{ \sum_k a_k \tilde{\phi}_i^h \,|\, \sum_k |a_k|^2 < \infty \right\}. \tag{4}$$

$h > 0$ is a grid space parameter and $D \in \mathbb{R} \geq 1$ is a decay parameter. $\tilde{\phi}_i^h$ is centered at the grid point $h{\cdot}i$. For fixed D we cannot represent polynomials exactly with Gaussians. But we can choose D such that the local approximation error does not exceed a given bound. The larger D is chosen, the smaller is the approximation error (*cf.* ref. 31 and 32 for further details).

Gaussians can also be used for multiresolution analyses[33,32] which allow in principle even better adaptivity, *i.e.* localized resolution refinements in coordinate space (or, equivalently, a variable representation interval in momentum space). While this may also be desirable, we do not employ this feature in the present paper. In situations typical for quantum wavepacket dynamics of chemical reactions, there are huge differences in coordinate ranges for different degrees of freedom. For instance dissociative ones cover a large (in principle infinite) range while others (spectator modes) only show small deviations from their equilibrium values. In contrast the associated momenta do not differ nearly as much. Therefore, we expect significantly larger savings from adapting the representation interval in coordinate space. Extensions to multiresolution analysis are the topic of future work.

The problem with large D lies in the large width of the Gaussians which results in a very dense and very badly conditioned overlap matrix $\boldsymbol{S}$. With linear combination of the Gaussians we can get an orthonormal basis set (*e.g.* ref. 34) *via* the Fourier transform:

$$\mathscr{F}(\phi)(\omega) = \frac{\mathscr{F}(\tilde{\phi}_0^1)(\omega)}{\left(\sum_{k\in\mathbb{Z}} \mathscr{F}(\tilde{\phi}_0^1)(\omega - 2k\pi)^2\right)^{1/2}}. \tag{5}$$

The shifted and dilated $\phi_i^h(x) = h^{-1/2}\phi(h^{-1}x - i)$ form an orthonormal basis of V_h. From these functions we easily get the interpolating functions mentioned above. A set $\{\varphi_i\}$ is said to be interpolating if it fulfils

$$\varphi_i(j) = \delta_{ij} \text{ for all } i, j \in \mathbb{Z}. \tag{6}$$

We get an interpolating function $\tilde{\varphi}$ by folding ϕ with itself [eqn (7)].

$$\tilde{\varphi} = \phi * \phi, \;\; i.e.\; \tilde{\varphi}(x) = \int \phi(y)\phi(x - y)dy \tag{7}$$

and $\tilde{\varphi}_i(x) = \tilde{\varphi}(x - i)$. The interpolating condition is satisfied since

$$\begin{aligned}\tilde{\varphi}_i(j) &= \int \phi(y)\phi(j - i - y)dy \\ &= \int \phi(y)\phi(y - (j - i))dy = \left\langle \phi_0 \phi_{j-i} \right\rangle = \delta_{ij}.\end{aligned}$$

To achieve the interpolating condition at the grid points $x_i = h{\cdot}i$, $i \in \mathbb{Z}$

$$\varphi_i(x_j) = \delta_{ij} \tag{8}$$

we set $\varphi_i(x) = \tilde{\varphi}(h^{-1}x - i)$ for all $i \in \mathbb{Z}$.

Since the orthonormal functions ϕ_i^h are in V_h we have a representation

$$\phi_0^h = \sum_{k\in\mathbb{Z}} a_k \tilde{\phi}_k^h. \tag{9}$$

Folding a Gaussian with itself is again a Gaussian but with a decay of $2D$:

$$\tilde{\phi}_0^1 * \tilde{\phi}_0^1 = \int \mathrm{e}^{-\frac{y^2}{D}} \mathrm{e}^{-\frac{(x-y)^2}{D}} \mathrm{d}y = \sqrt{\frac{\pi D}{2}}\, \mathrm{e}^{-\frac{1}{2D}x^2}. \tag{10}$$

Therefore the interpolating function $\tilde{\varphi}$ is given by

$$\tilde{\varphi}(x) = \sqrt{\frac{\pi D}{2}} \sum_{k\in\mathbb{Z}} \tilde{b}_k\, \mathrm{e}^{-\frac{(x-k)^2}{2D}} \tag{11}$$

with

$$\{\tilde{b}_k\}_l = (\{a_k\}^* \{a_k\})_l = \sum_{k\in\mathbb{Z}} a_k a_{k-l}. \tag{12}$$

It turns out that these coefficients are very alternating, hence they are not quite useful for numerical calculations. It is better to represent $\tilde{\varphi}$ approximatively in $V_{1/2}$, *i.e.*

$$\tilde{\varphi}(x) \approx \sum_{k\in\mathbb{Z}} b_k \tilde{\phi}_k^{1/2}(x) = \sum_{k\in\mathbb{Z}} b_k \mathrm{e}^{-\frac{(2x-k)^2}{D}}. \tag{13}$$

The additional error made by this approximation is negligible for $D \geq 3$. (Exact error estimates can be derived following the papers of Maz'ya and Schmidt.[31–33]) As shown below, with $D = 4$ we indeed obtain sufficient accuracy in practical applications. With eqn (14) the unknown coefficients b_k can be determined *via* Fourier transform.

$$\mathscr{F}(\tilde{\varphi}) = \mathscr{F}(\phi)^2. \tag{14}$$

We take the ansatz

$$\tilde{\varphi} = \sum_{k\in\mathbb{Z}} b_k\, \tilde{\phi}_k^{1/2} \tag{15}$$

which in Fourier space becomes

$$\mathscr{F}(\tilde{\varphi})(\omega) = b(\omega)\mathscr{F}(\tilde{\varphi}_0^{1/2})(\omega) \tag{16}$$

where $b(\omega)$ is given by

$$b(\omega) = \sum_{k\in\mathbb{Z}} b_k \mathrm{e}^{-\mathrm{i}k\omega}. \tag{17}$$

With eqn (16), (14) and (5) we obtain the analytical expression

$$b(\omega) = \frac{\mathscr{F}(\tilde{\phi}_0^1)(\omega)^2}{\mathscr{F}(\tilde{\phi}_0^{1/2})(\omega) \sum_{k\in\mathbb{Z}} \mathscr{F}(\tilde{\phi}_0^1)(\omega - 2k\pi)}. \tag{18}$$

Since a Gaussian in Fourier space is again a Gaussian, we have the explicit expression

$$b(\omega) = \frac{2}{\sqrt{\pi D}} \mathrm{e}^{-\frac{7}{4}D\omega^2} \left(\sum_k \mathrm{e}^{-\frac{D}{2}(2\omega - 2k\pi)^2} \right)^{-1}. \tag{19}$$

Hence, our new basis functions are

$$\varphi_i(x) = \sum_{k\in\mathbb{Z}} b_k \mathrm{e}^{-\frac{(2h^{-1}x - 2i - k)^2}{D}}. \tag{20}$$

Practically, discrete inverse Fourier transformation (FT) of the analytically given $b(\omega)$ [eqn (19)] yields the expansion coefficients b_k for the calculation of the new basis functions φ_i in eqn (20). Derivatives of those interpolating basis functions are analytically available due to the easy derivation of the Gaussian basis functions:

$$\varphi_i'(x) = -\frac{4}{hD} \sum_{k\in\mathbb{Z}} b_k \left(\frac{2x}{h} - 2i - k \right) \mathrm{e}^{-\frac{(2h^{-1}x - 2i - k)^2}{D}}, \tag{21}$$

$$\varphi_i''(x) = \frac{8}{h^2 D^2} \sum_{k\in\mathbb{Z}} b_k \left[2\left(\frac{2x}{h} - 2i - k \right) - D \right] \mathrm{e}^{-\frac{(2h^{-1}x - 2i - k)^2}{D}}. \tag{22}$$

For the multivariate case [eqn (23)], *i.e.* with d dimensions, we take as basis function just the tensor products of the one dimensional interpolating functions.

$$\varphi_{\boldsymbol{k}}(x_1,\ldots,x_d) = \varphi_{k_1}(x_1)\ldots\varphi_{k_d}(x_d), \text{ for all } \boldsymbol{k} \in \mathbb{Z}^d, \quad x_1,\ldots,x_\mathrm{d} \in \mathbb{R}. \tag{23}$$

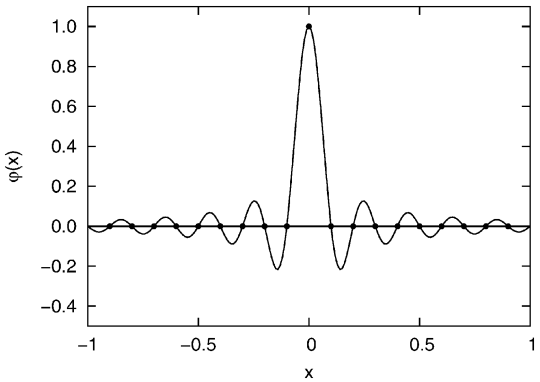

Fig. 1 1D interpolating Gaussian ($D = 4$) centered at $x_i = 0.0$ on the basis function grid (grid parameter $h = 0.1$). Solid dots mark grid points. The function value is zero on all grid points except for $x = x_i$.

Fig. 1 shows the values of an interpolating Gaussian basis function with $D = 4$, centered at $x_i = 0$ with a grid parameter of $h = 0.1$. As seen from the plot, the nodes of the function are exactly located on the collocation grid points (black dots).

C Collocation

Collocation methods for the time-independent Schrödinger equation were popularized by Friesner[35] and by Yang and Peet.[36] Combining the collocation method with interpolating Gaussians offers two big advantages for our proDG approach: Firstly, as shown in Table 1, all integrals needed to calculate the matrix elements of the Hamiltonian (*e.g.* the kinetic energy entry T_{ij} and the potential energy entry V_{ij}) are replaced by evaluation of the basis functions (or their derivatives) and of the potential on the grid given by the basis function centers. In contrast, in the usual "Galerkin" method, an additional (finer) grid is needed to perform numerical integration. In line with

Table 1 Comparison of FBR and collocation method

	FBR	Collocation
Propagation scheme	$\mathrm{i}\hbar \boldsymbol{S} \frac{\partial}{\partial t}\boldsymbol{c} = \boldsymbol{H}\boldsymbol{c}$	$\mathrm{i}\hbar \boldsymbol{R} \frac{\partial}{\partial t}\boldsymbol{c} = \boldsymbol{H}\boldsymbol{c}$
T_{ij}	$\langle\varphi_i\vert\hat{T}\vert\varphi_j\rangle$	$\hat{T}\varphi_j(x_i)$
V_{ij}	$\langle\varphi_i\vert\hat{V}\vert\varphi_j\rangle$	$V(x_i)\delta_{ij}$
S_{ij} resp. R_{ij}	$\langle\varphi_i\vert\varphi_j\rangle$	$\varphi_j(x_i)$

our proDG idea, this second grid would have to be pruned and extended, as basis functions are deleted and added, respectively. Clearly, this incurs a substantial administrative overhead in any implementation of this approach, in particular for higher dimensions. Collocation allows one to eliminate this secondary grid entirely. This also shrinks the representation range further, since in traditional FBR the integration grid must continue at least until the last basis function has decayed to zero.

Secondly, for interpolating Gaussians, the $\boldsymbol{R}$ matrix mentioned in Table 1 is just the identity matrix. In general collocation, the $\boldsymbol{R}$ matrix contains the values of all basis functions on all grid points, replacing the overlap matrix in non-orthogonal FBR. In this sense, collocation with interpolating Gaussians offers a very simple numerical representation of the time-dependent Schrödinger equation

$$\mathrm{i}\hbar\frac{\partial}{\partial t}\boldsymbol{c} = \boldsymbol{H}\boldsymbol{c} \tag{24}$$

with the expansion coefficients being identical to the wavefunction values at the collocation grid points and the matrix $\boldsymbol{H}$ containing the matrix elements T_{ij} and V_{ij} of Table 1. All standard propagation schemes can be easily applied to eqn (24).

The simple representation of eqn (24) is effectively the same as in standard DVR approaches. Compared to polynomial DVRs we are missing the implicit quadrature of Gaussian quality, which is replaced by a more general collocation here. However, this is recognized as a non-essential feature of DVRs in the literature;[22] the diagonality of the potential matrix is clearly more important. In fact, the application examples presented below demonstrate that we can easily achieve sufficient accuracy with our present scheme. Using a standard polynomial DVR with unequally distributed grid points would make arbitrary extensions and contractions of grids very awkward; this would presumably render our approach unfeasible. In addition, as mentioned in the introduction, our wavepacket-based automatic pruning/extension scheme relies upon spatial locality of our basis functions, also from the FBR perspective. This also rules out straightforward adaption of the sine-based DVR of Colbert and Miller,[21] which does feature equally spaced grid points but for the price of delocalized FBR functions. Actually, in our application examples, we demonstrate that our approach is far more efficient than a Colbert–Miller DVR without pruning.

D Sparse matrix implementation

A major disadvantage of the FBR method is the amount of memory needed, as the size of the Hamiltonian matrix scales quadratically with the number of basis functions. The proDG approach cannot reduce this inherent scaling, but it does reduce its prefactor significantly, due to the minimal number of coefficients representing the wavefunction. Using the collocation method combined with interpolating Gaussian basis functions changes the structure of the Hamiltonian matrix such that it contains many zero values due to the fact that the potential part of the Hamiltonian is only on its diagonal. The sparsity is even more pronounced for higher dimensions and therefore suggests the use of a sparse matrix representation. For simplicity, in this first implementation the standard coordinate storage scheme (COO)[37] was utilized. Exemplary data on the sparsity will be mentioned with the applications in section III. Ongoing work focuses on implementing more refined sparse matrix methods that can be combined efficiently with a parallelized, distributed computing approach.[38]

E Numerical kinetic energy operator

It is a well-known fact that the derivation of analytical expressions of the kinetic energy operator (KEO) in curvilinear coordinates for a molecular system becomes very complicated with increasing number N of atoms. Hence, already for $N \gtrsim 5$ computer algebra programs like Mathematica have to be employed to obtain numerically usable expressions of the Laplace–Beltrami operator [eqn (25)]. This, in turn, leads to the phenomenon that most programs in this field of research are not universal but need to be re-written for each new system under consideration. However, even for programs which can handle arbitrary systems (like MCTDH) normally extensive *a priori* calculations have to be carried out to obtain a numerically applicable expression of the KEO. In order to circumvent this problem we use a numerical but exact approach of Lauvergnat *et al.*, which is called Tnum.[13,28] Note that this program has also recently been used to check the correctness of analytically derived KEOs in MCTDH calculations.[5,39] In the following we give a brief review of the basic equations needed in context of KEOs within curvilinear coordinates, where, for the sake of simplicity, we restrict ourselves to the case of zero total angular momentum ($\mathscr{J} = 0$).

Assuming that the configuration of an N-atomic molecule is appropriately described in terms of $3N - 6$ internal coordinates $\boldsymbol{q} = (q^1,\ldots,q^{3N-6})^{\mathrm{T}}$, the general expression of the kinetic energy operator can be written in a very compact form:[40–43]

$$\hat{T}(\boldsymbol{q},\hat{\boldsymbol{p}}) = -\frac{\hbar^2}{2}\Delta = \frac{1}{2}\sum_{i,j=1}^{3N-6}\hat{p}_i^{\dagger}\, g^{ij}(\boldsymbol{q})\,\hat{p}_j\,. \tag{25}$$

In eqn (25) $\hat{p}_i = -\mathrm{i}\hbar\partial/\partial q^i$ are the conjugate momentum operators, $\hat{p}_i^{\dagger} = J^{-1}(\boldsymbol{q})\hat{p}_i J(\boldsymbol{q})$ their adjoints, and g^{ij} the contravariant components of the deformational part of the metric tensor. J denotes the Jacobian determinant of the transformation from cartesian to curvilinear coordinates.

Expanding eqn (25) leads to an expression of the kinetic energy operator which is much better suited for numerical calculations:

$$\hat{T}(\boldsymbol{q},\partial_{\boldsymbol{q}}) = \sum_{i,j=1}^{3N-6} f_2^{ij}(\boldsymbol{q})\frac{\partial^2}{\partial q^i \partial q^j} + \sum_{i=1}^{3N-6} f_1^{i}(\boldsymbol{q})\frac{\partial}{\partial q^i} \tag{26}$$

with

$$f_2^{ij}(\boldsymbol{q}) = -\frac{\hbar^2}{2}g^{ij}(\boldsymbol{q}), \tag{27}$$

$$f_1^{i}(\boldsymbol{q}) = -\frac{\hbar^2}{2}\sum_{j=1}^{3N-6}\left[J^{-1}(\boldsymbol{q})\frac{\partial}{\partial q^j}J(\boldsymbol{q})\right]g^{ij}(\boldsymbol{q}) + \left[\frac{\partial}{\partial q^j}g^{ij}(\boldsymbol{q})\right]. \tag{28}$$

It should be noted here that all the above equations are valid only if the standard Euclidean volume element $\mathrm{d}\tau = J(\boldsymbol{q})\mathrm{d}\boldsymbol{q}$ has been

used to normalize the wavefunction. In case an arbitrary (positive) weighting function ρ is used (*i.e.* $\int\psi^*(\boldsymbol{q})\psi(\boldsymbol{q})\rho(\boldsymbol{q})\mathrm{d}\boldsymbol{q} = 1$), the kinetic energy operator in eqn (25) has to be modified according to $\hat{T} \rightarrow \hat{T}\rho = J^{-1/2}\rho^{1/2}\hat{T}J^{1/2}\rho^{-1/2}$. As a consequence, the substitution $J \rightarrow \rho$ is necessary in eqn (28) and an additional purely multiplicative term V_{ep} appears in eqn (26). The latter, being a function of the coordinates only, is often referred to as extra-potential term, and analytical expressions can be found, *e.g.* in ref. 43.

The TNUM software provides the exact numerical values of the kinetic functions f_2^{ij} and f_1^i (and V_{ep}) for essentially arbitrary choices of coordinates, where the molecular geometry can be specified conveniently by means of a Z-matrix. Furthermore, it offers several options to treat spectator coordinates approximately (rigid or flexible models) without the need of rewriting any part of the program. In our present multidimensional implementation, we have combined the adaptive PRODG strategy with the TNUM approach. Thus, also the kinetic part, and hence the whole Hamiltonian, can be calculated on-the-fly in this setup.

III. Numerical results

The methods described in section II were implemented in our own quantum-dynamics program package MRPROPA. This is a general-purpose quantum wavepacket propagation program, written from scratch[44] to cover systems of arbitrary dimensionality, and has been used in earlier calculations in conjunction with traditional non-adaptive basis sets of direct-product form.[45,46]

In the following sections, numerical expenses such as computational time and memory requirements are presented. These should not be understood as the best possible absolute measures. Instead, we have performed all calculations shown here within the same MRPROPA program framework, thus ensuring direct comparability of these calculations with each other. The program MRPROPA itself, however, is being developed to treat *general* molecular systems. Therefore, the computational expenses are by construction somewhat higher than for programs which are designed and optimized for specific problems. Also, further optimization may be possible within MRPROPA in several locations. Hence, the expenses of our reference calculations are probably not optimal, compared to similar calculations done with other programs. Also, needless to say, timings are highly machine-dependent, and exact memory requirements are not easy to determine.

A 2D-model example: double-well potential

As a proof for the reliability of the PRODG approach with interpolating Gaussians in more than one dimension, we first present calculations with a model potential which has a double minimum in one dimension, denoted as q_1 ($V(q_1) = \frac{c}{(\Delta-a)^2(\Delta-b)^2}(q_1-a)^2(q_1-b)^2$, with the potential minima coordinates $a = 1.0$ Å, $b = 2.0$ Å, $\Delta \equiv \frac{a+b}{2}$, and the barrier height $c = 80.0$ kJ mol^{-1}), combined with a harmonic potential in the second coordinate q_2 ($V(q_2) = \frac{k}{2}(q_2-q_2^0)^2$, with the potential minimum coordinate $q_2^0 = 2.0$ Å and a force constant of $k = 10\,630$ kJ mol^{-1} Å^{-2}).

The calculations are compared to a converged reference, which is performed with the FFT method[47,48] on a direct-product grid in both degrees of freedom (DOFs).

This 2D-model case has been chosen with some deliberation: due to the harmonic potential in q_2 and the absence of couplings the wavepacket remains compact in this coordinate during the propagation. This is an optimal test case for the PRODG method since the mechanisms for basis set pruning and enlargement have to be constantly active in different locations, leading to periodic increases and decreases of the number of basis functions that can easily be checked. At the same time, the wavepacket expands in coordinate q_1 and tunnels through the double-well potential barrier, which is challenging for any adaptive approach. The separability of this model is not exploited in the program, therefore this is a real 2D test case. This two-dimensional model also easily allows to compare and test different parameters. The kinetic part of the Hamiltonian is kept quite simple, with constant effective masses that correspond to those of hydrogen for both DOFs.

The initial wavepacket for the PRODG and reference calculation is chosen as a 2D-Gaussian function, which is situated only near one minimum of the double well potential. In particular no parts of the wavepacket are located on top of the barrier or in the second minimum at $t = 0$ fs (*cf.* Fig. 2). Time propagation was performed using symplectic integrators[49] of order 4. The time step is 0.04 fs and the total propagation time is 100 fs. The interpolating Gaussian basis functions (with $D = 4$) were arranged equally spaced in q_1 and q_2 with a grid parameter h of 0.025 Å [eqn (3) and (20)] for both DOFs. For the FFT reference, the propagation grid extends from 0.0 to 3.0 Å in q_1 and from 1.0 to 3.0 Å in q_2, also with an equal grid spacing of 0.025 Å in both dimensions. The total energy of the system is 74.4 kJ mol^{-1}, which is well below the barrier height of the potential of 80.0 kJ mol^{-1}.

Calculation times and maximum memory used on our test system (a 2.2 GHz AMD Opteron with 4 GB RAM) are 5:04 min and 9.5 MB RAM for the PRODG calculation (with PRODG thresholds set to $\theta_{\mathrm{a}} = 7 \times 10^{-5}$ and $\theta_{\mathrm{d}} = 4 \times 10^{-5}$). This is to be compared to 1:33 min and 9.1 MB RAM for the FFT reference. These total times and memory usages are within the same order of magnitude for both calculations, with the reference being 3.3 times faster than our method. This is surprising since one would not expect an FBR-based calculation to come close to a FFT calculation for a low-dimensional example. Due to its enforced direct-product form the FFT representation is expected to suffer from decreasing efficiency when the dimensionality of the problems is increased further.[50] As a side-note we mention that a standard Colbert–Miller DVR[21] without any pruning (neither statically nor dynamically) takes 16:41 min with the same parameter settings. Also, we would like to re-emphasize that the relative timings have to be taken more seriously than the absolute ones, since our code is not fully optimized in every respect.

As described in the theory section, in the PRODG case basis functions exist only in the spatial region where the contributions to the wavepacket are non-negligible. Note that in our computational approach also the potential is stored only at points where the active basis functions exist; nevertheless, for the sake of clarity the global potential as used in the reference

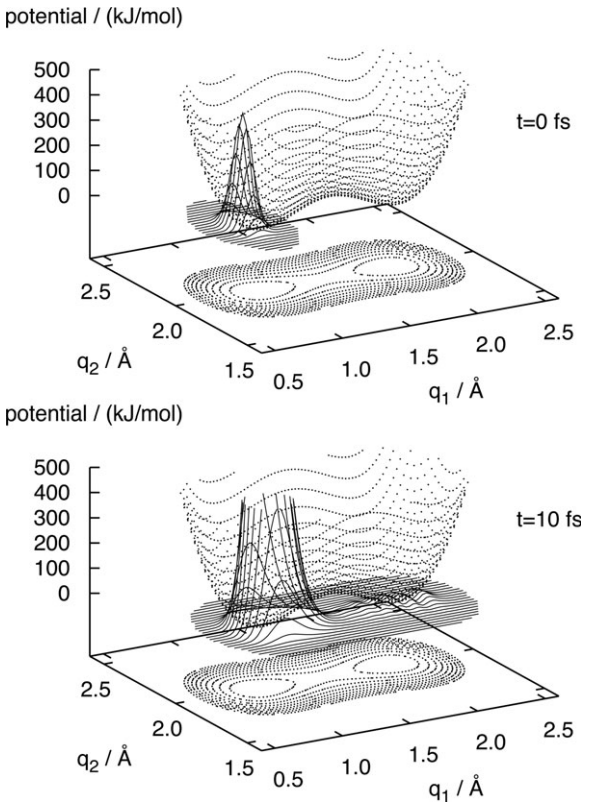

Fig. 2 2D snapshots of the propagating wavepacket at $t = 0$ fs (top) and $t = 10$ fs (bottom). Note that the zoom on the wavepacket is ten times larger at $t = 10$ fs (this is unrelated to the vertical axis which displays the potential energy scale). The wavepacket movement from the "left" to the "right" minimum of the potential is a superposition of barrier crossing and tunneling contributions, since the total energy of the wavepacket is about 74 kJ mol^{-1} compared to a barrier height of 80 kJ mol^{-1}.

calculation is shown in Fig. 2. During the propagation the wavepacket spatially expands, so the number of active basis functions (n_{bf}) increases accordingly (Fig. 3, solid line). Due to a combination of tunneling and barrier crossing the wavepacket spreads out to the second potential minimum, and the number

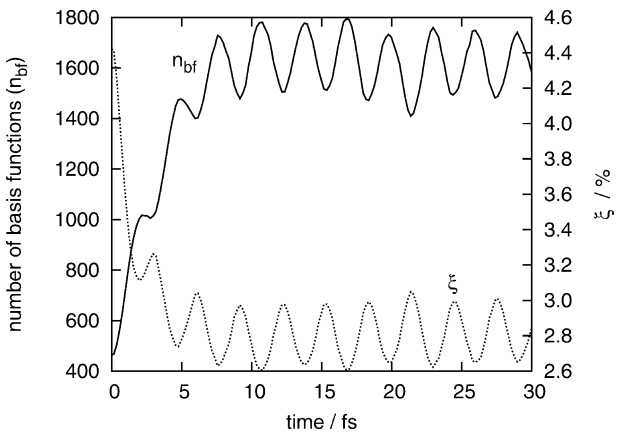

Fig. 3 Number of active basis functions (solid line), and fraction ξ of non-zero elements (dotted line) in the sparse matrix representation in percent of a full matrix representation.

of basis functions rises to a maximum. Because of the harmonic potential in the second dimension, oscillations occur in this degree of freedom. This can also be seen in Fig. 3, where the grid size decreases and increases periodically with half the vibrational period of the harmonic oscillator (which is about 6.1 fs). The new spatial extent is also visualized in the second wavefunction plot, Fig. 2, at $t = 10$ fs.

Fig. 3 also shows the percentage ξ of non-zero elements in the sparse matrix representation (dotted line) with respect to the full dense Hamiltonian matrix representation. As seen from the plot, ξ decreases from an initial value of about 4.4% rapidly to oscillations around 2.8%. Interestingly, the fraction ξ decreases as the number of active basis functions increases. This is an important issue for higher dimensional calculations as we now know that more basis functions yield a lower percentage of non-zero elements in the Hamiltonian matrix.

In order to compare the FFT and proDG propagations in a condensed manner, we monitor the quantum mechanical flux, F, through the dividing surface $s(\boldsymbol{q}) = q_1 - q_1^* = 0$, where q_1^* denotes the location of the transition state at 1.5 Å. In general, the flux through the boundaries $\partial\Omega$ of an area Ω can be written as

$$F(t) = \oint_{\partial\Omega} \boldsymbol{J}\cdot\boldsymbol{n} \quad \mathrm{d}S = \int_{\Omega} \nabla\cdot\boldsymbol{J}\mathrm{d}\boldsymbol{q} \qquad \text{with} \qquad (29)$$

$$\nabla\cdot\boldsymbol{J} = 2\hbar\ \mathrm{Im}(\Psi^*(\boldsymbol{q},t)\hat{T}\,\Psi(\boldsymbol{q},t)). \qquad (30)$$

In eqn (29) $\boldsymbol{J}$, $\boldsymbol{n}$ and $\mathrm{d}S$ denote the flux density, the outward pointing unit normal, and the surface element, respectively. The latter is in the present 2D case a line element, of course, and the area Ω is given by $\Omega = \{\boldsymbol{q} \in \mathbb{R}^2 | q_1 \leq q_1^*\}$. In this context it is also worth noting that an explicit derivation of the flux operator in curvilinear coordinates, which can be used to calculate cumulative reaction probabilities, is given in ref. 51 and 52.

The fluxes obtained with the 2D proDG approach and the FFT method as reference are shown in Fig. 4. Obviously there are only very small differences between the fluxes calculated by the two methods. Only after substantial enlargements (Fig. 5) any differences can be noticed at all. As problems often occur at long propagation times due to error accumulation, it is important to notice that the differences do not increase after long times.

The error made by our approach can be tuned by changing the size of the thresholds θ_a and θ_d. Fig. 6 shows that lower thresholds—meaning less pruning—result in a smaller error between the reference and the proDG calculation. If no pruning is used at all on a basis covering all relevant space initially, the error consistently vanishes (not shown). If the adaptive basis initially covers only the starting wavepacket, it becomes more difficult in this example to reach arbitrary accuracy (the graph in Fig. 6 levels off) by only changing the thresholds, since the deep tunneling situation modeled intentionally here magnifies all numerical errors. Properly adjusting more parameters (*e.g.* the frequency of pruning/extension) to this situation should lead to further reduction of the error, but we have not followed this line of enquiry further here.

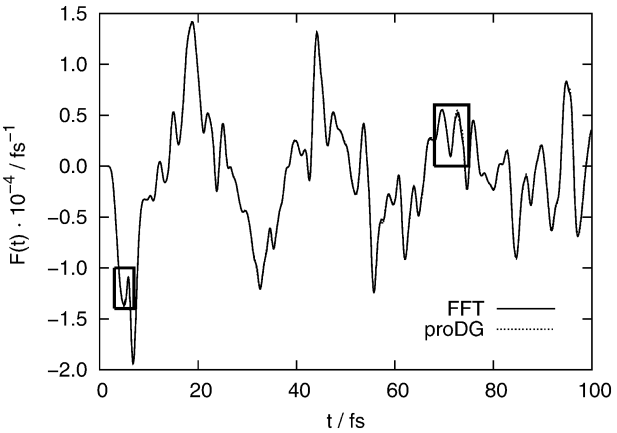

Fig. 4 Flux through the dividing surface between the two minima, for the situation depicted in Fig. 2. Data for our proDG method (thresholds set to $\theta_a = 7 \times 10^{-5}$ and $\theta_d = 4 \times 10^{-5}$) are directly superimposed with the corresponding data for the FFT reference calculation, showing perfect agreement.

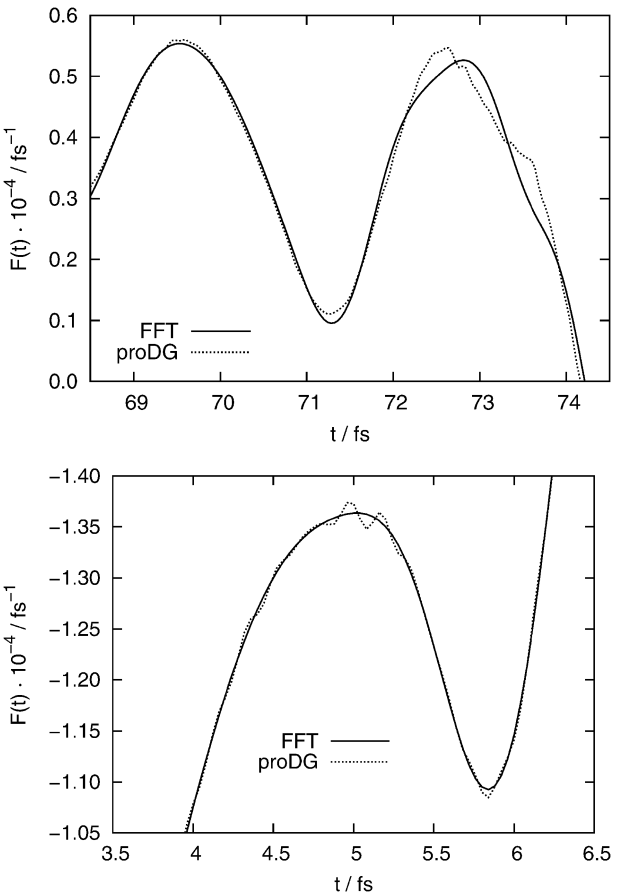

Fig. 5 Small deviations between the proDG calculation and the reference calculation, for the data shown in Fig. 4, become visible only upon magnification; as examples, the two regions marked with rectangles in Fig. 4 are shown enlarged here, in the upper and lower panel, respectively.

Independent of the application, the prize to pay for increased exactness is the computational time, which increases exponentially with smaller exponents of the threshold, as expected.

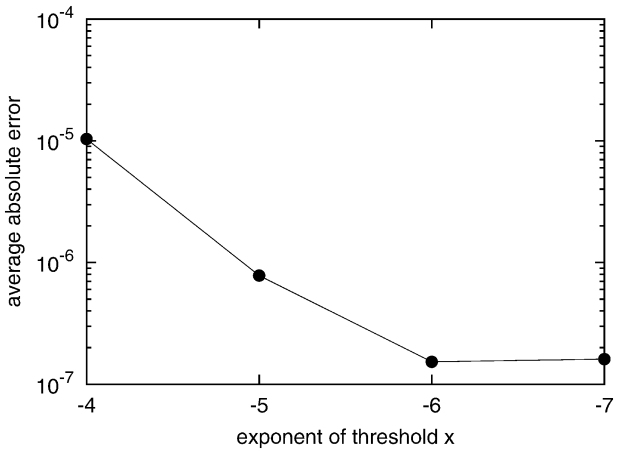

Fig. 6 Absolute error, averaged over complete propagation time, of 2D-proDG calculation compared to FFT reference depending on thresholds for addition and deletion of basis functions. Thresholds shown on the abscissa are $\theta_a = 7 \times 10^x$ and $\theta_d = 4 \times 10^x$, respectively.

Conversely, if an increased error is still acceptable, computational time decreases exponentially.

The 2D model example of this section suggests that qualitative and quantitative agreement can be extended to higher-imensional cases, even in the presence of strong tunneling effects. For the one-dimensional case, quantitative agreement between the proDG approach and traditional reference calculations was already shown in ref. 26.

B 3D-example: NOCl in the T_1-state

Here we present a 3D example that is fundamentally different in several respects from the 2D example of the previous section: it is not artificial but a real system, it features non-constant kinetic functions, and the dynamical behavior of the wavepacket is very different (in a way, it is simpler for our adaptive approach, since the dynamics is dominated by a simple spreading-out of the packet, without barrier tunneling). Also, of course, this example demonstrates that further extensions to higher-dimensional cases are possible with equal ease and without deterioration of the quantitative reliability demonstrated in the previous 2D example.

For this 3D case, we have chosen the quantum dynamics of NOCl in the T_1 state, as studied by Sölter *et al.*[53] Using the original potential energy surface of those authors, and the same initial wavepacket and grid parameters, we first performed a reference calculation using our own traditional FFT/DVR implementation, where the two distances in Jacobi coordinates are treated by a 2D-FFT representation with 56 × 96 grid points, while the angular coordinate is represented by a Gauss–Legendre DVR with $j_{max} = 45$ for the rotational basis. As expected, these calculations reproduce the original data of ref. 53 perfectly. We then repeated the dynamics calculations, using our adaptive FBR-representation with interpolating Gaussians, comparing energy, norm and the autocorrelation function with the FFT/DVR reference calculation.

proDG calculations were performed incorporating different parameter sets, varying the distance between the basis functions and the thresholds. Two different sets will be presented in the

Table 2 Numerical parameters used in proDG calculations. For the two stretching coordinates the same grid spacing h is employed

	proDG1	proDG2
h	0.016 Å/3.0°	0.020 Å/4.0°
θ_a	2×10^{-4}	7×10^{-4}
θ_d	9×10^{-5}	5×10^{-4}

following section and are summarized in Table 2. In the first set (from now on referenced as proDG1), the FFT/DVR refecence parameters are matched as closely as possible. The interpolating Gaussian basis functions with $D = 4$ have a spacing h of 0.016 Å for the two distances and 3.0° for the angular coordinate, respectively. The time step used throughout is 0.1 fs at a total propagation time of 150 fs. Time propagation is again performed with simplectic integrators of order 4. The thresholds are relatively small, *i.e.* $\theta_a = 2 \times 10^{-4}$ and $\theta_d = 9 \times 10^{-5}$, which means that even small contributions of the wavepacket are kept. The second parameter set, labeled proDG2, has wider grid spacings (0.02 Å and 4.0°) and larger thresholds ($\theta_a = 7 \times 10^{-4}$ and $\theta_d = 5 \times 10^{-4}$). These parameters result in reduced computational costs. The two different proDG calculations presented here should be understood as two representative points on a broader scale of calculations that all yield results very similar in quality to the FFT/DVR reference, as we demonstrate below.

Complex absorbing potentials (CAPs) can also easily be used with the proDG approach. For all following calculations the form given by Manolopoulos and Zhang[54,55] is applied:

$$V_{\mathrm{cap}}(\boldsymbol{Q}) = \sum_j V_{R,j} \exp(-\alpha_{R,j}\kappa_j) + \mathrm{i}V_{I,j} \exp(-\alpha_{I,j}\kappa_j), \quad (31)$$

where

$$\kappa_j = \frac{Q_j^{\max} - Q_j}{Q_j - Q_j^0}. \quad (32)$$

In eqn (32) $Q_j^{\max}$ denotes the end of the grid in the jth coordinate. The amplitudes of the real and imaginary parts are $\alpha_R = 0.739$ and $\alpha_I = 3.071$, respectively, while the corresponding coefficients are set to $V_R = -1.27E_T^{\max}$ and $V_I = -0.994E_T^{\max}$ for a maximal kinetic energy $E_T^{\max}$. The starting values of the CAP, Q_j^0, were chosen to be about 1.36 and 3.11 Å, respectively, for the two stretch coordinates. For the angular coordinate no CAP is necessary in this particular case.

As seen qualitatively in Fig. 7 and quantitatively in Fig. 8, both proDG1 and proDG2 almost perfectly reproduce the reference autocorrelation function. The autocorrelation function is given by the amplitude of the initial wavepacket with the wavepacket at time t, *i.e.* $A(t) = \langle \Psi(0) | \Psi(t) \rangle$. Even in the enlarged region, where visual deviations of the two lines are noticeable, the qualitative shape of the reference autocorrelation function is reproduced quite well. The absolute maximum deviation is less than 0.001 for the first parameter set and only slightly larger (0.004) for the second one. Even at longer propagation times the deviations do not systematically increase. The reader should also note that the absorption spectra (calculated by Fourier transformation of the autocorrelation function) for all three presented cases do not differ at all from those shown in Fig. 8b of the original paper.[53]

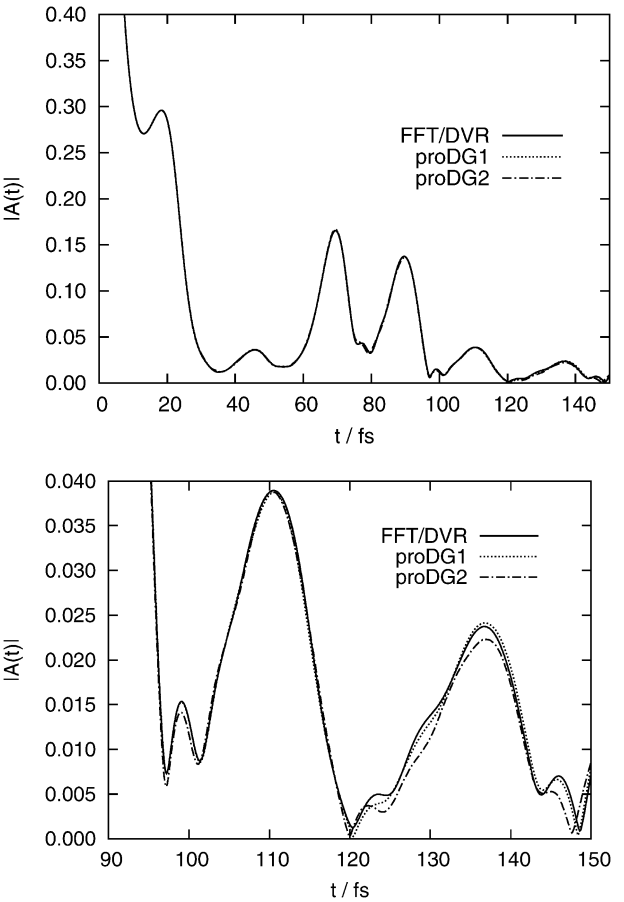

Fig. 7 Modulus of autocorrelation functions for the 3D dynamics of NOCl in the T_1 state, directly comparing both proDG runs (proDG1 and proDG2) with the FFT/DVR reference (upper panel). Again, differences become visible only upon magnification (lower panel).

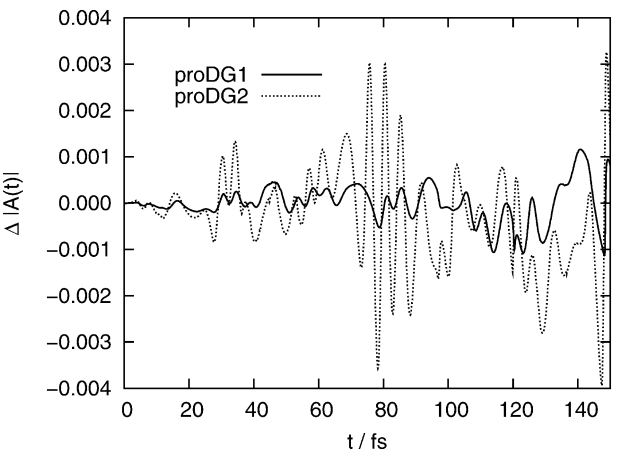

Fig. 8 To further magnify the small deviations in Fig. 7, here only the differences between the FFT reference and proDG1 (solid line) and between the FFT reference and proDG2 (dotted line) are shown.

On our test machine, the 2D-FFT/1D-DVR reference calculation takes 0:55 h (61 MB RAM), the proDG1 calculation needs 13:09 h (93 MB RAM) whereas the proDG2 parameter set

yields a calculation time of only 1:47 h (29 MB RAM). Compared to the FFT/DVR reference both proDG calculations use about the same order of magnitude in memory but are slower roughly by factors of 14 and 2, respectively. This shows that with the appropriate choice of thresholds a good balance between numerical costs and accuracy can be achieved. Of course, optimal thresholds are not easily determined, but from this case and also from the previous 2D example values between 10^{-4} and 10^{-5} turn out to be a very good first choice. Another possibility would be to use relative thresholds, *i.e.* only coefficients whose magnitude is of a certain percentage of the maximal coefficient value are kept, which leads to time-dependent thresholds (as suggested in ref. 27).

Fig. 9 shows the number of active basis functions as function of the propagation time. It continuously increases during the first 50 fs due to the spread of the wavepacket into the flat regions of the potential energy surface. Because at later times the complex absorbing potential deletes parts of the wavepacket, the active basis function region does not further increase and approximately stays at a constant level of about 35 000 or 11 000 basis functions for the proDG1 and proDG2 calculations, respectively. A traditional, non-adaptive FBR representation would not be used here, given the better alternatives. However, since our approach can be understood as being conceptually FBR-based, it is interesting to mention that a fully dense Hamiltonian matrix with 35 000 basis functions would require at least 9.1 GB if stored completely. This drastic reduction shows the capabilities of the proDG approach. As already seen in the 2D example, the percentage ξ of the non-zero elements in the sparse matrix representation is less for the proDG1 case, where the number of basis functions is larger. This parameter set requires about half the percentage of the full Hamiltonian matrix compared to the proDG2 calculation. With starting values from 0.45% to a final value of about 0.14% the percentage is lowered by a factor of more than 10 on going from a 2D to a 3D calculation.

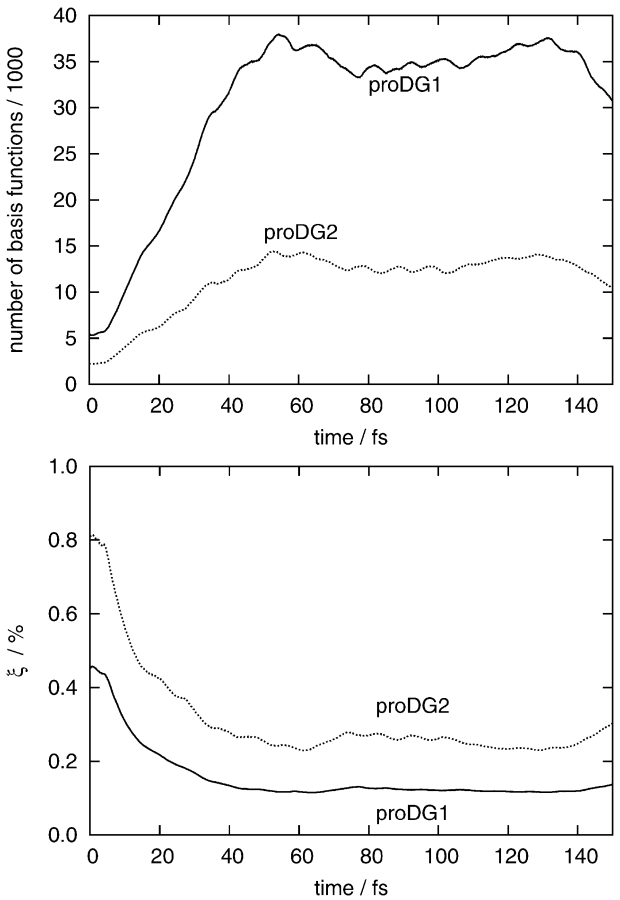

Fig. 9 Number of active basis functions (top panel), and percentage of non-zero elements in the sparse matrix representation (bottom panel), for 3D proDG dynamics calculations of NOCl in the T_1 state, with different setups as specified in the text (proDG1: solid line, proDG2: dotted line).

A point worth mentioning is where the computational time is spent during a proDG calculation. This is illustrated in comparison to the FFT/DVR reference in Fig. 10. The solid lines show the total real time as function of the propagation time. As expected, for the FFT/DVR case the computational time increases constantly with the propagation time and most of the numerical expense is spent on the action of the Hamiltonian on the wavefunction (dash–dotted lines). For the proDG case the situation is quite different. Only a tiny fraction of the total time, in fact much less than in the FFT/DVR case, is used for the sparse matrix–vector multiplication (Hamiltonian matrix × wavefunction). The major amount of time is spent to compute the matrix elements of the new rows and columns of the Hamiltonian after new basis functions are introduced (dotted line). However, as Fig. 11 shows, the Hamiltonian matrix has a very ordered structure. Hence, it appears possible that the negligible elements can be "anticipated" and thus would not need to be calculated. Closer analysis of the structure reveals a fractal buildup. The innermost rectangles are comparatively dense and correspond

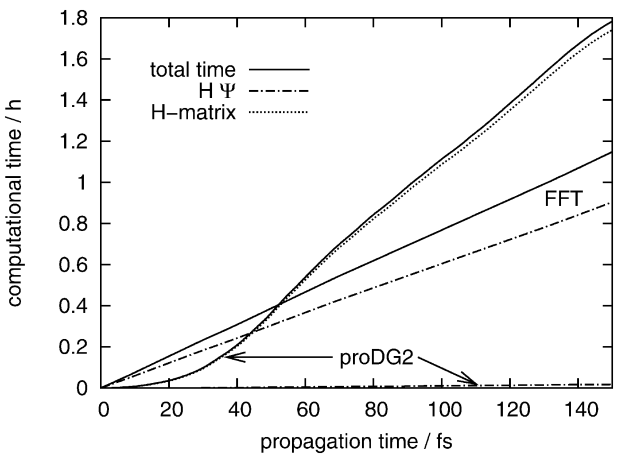

Fig. 10 Computational time in hours *versus* propagation time in femtoseconds. The solid lines show the total time for the FFT/DVR and proDG2 calculations while the dashed-dotted lines show the computational time need for the action of the Hamiltonian on the wavefunction for both cases. The dotted line refers only to the proDG calculation and shows the time necessary for the calculation of the new matrix elements of the Hamiltonian matrix after a new basis function is introduced.

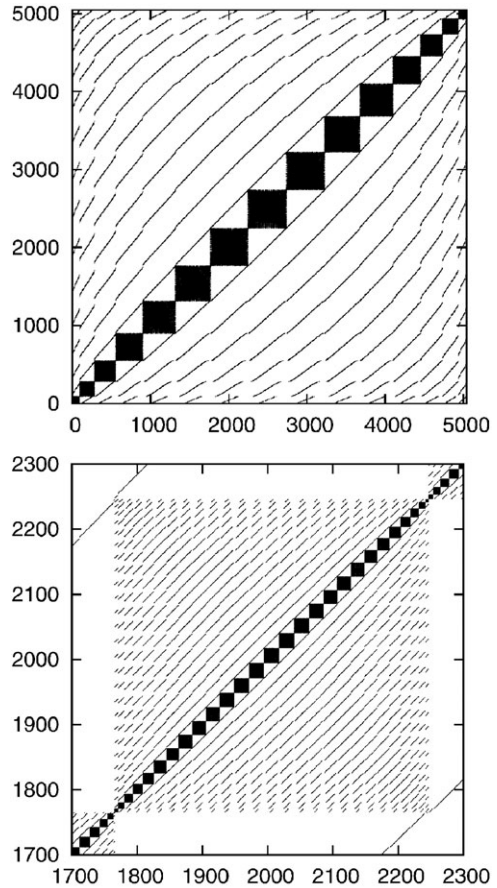

Fig. 11 Contour plot of the non-zero elements of the Hamiltonian matrix. The lower panel shows a closer view of one of the black squares from the upper panel.

to 1D calculations. For each further dimension, building blocks as depicted in the lower panel of Fig. 11 are added. For the 3D example presented here, this results in a complete Hamiltonian matrix as shown on the top panel of Fig. 11. Thus, the fraction of non-zero elements will decrease significantly with the dimensionality of the problem (as already observed above when going from 2D to 3D) and will tend towards insignificant deviations from diagonality. In addition, we anticipate that also the non-zero matrix elements will exhibit only relatively few repeating values (disregarding the possibly geometry-dependent factors f_1 and f_2 of eqn (26)), so that recalculating all of them can also be avoided. In short, there still is potential for further development of this approach.

IV. Conclusions

With this work, we have extended our adaptive basis proDG approach from a naive proof-of-principle implementation in one dimension[26] to a production code in many dimensions. At the same time, we have optimized the underlying basis representation from standard non-orthogonal distributed Gaussians to orthogonalized, interpolating Gaussians, eliminating both the overlap matrix and the additional numerical integration grid. The whole implementation is cast in a sparse matrix representation. All these methods have been implemented within the framework of our MrPropa propagation suite which we had combined with the Tnum approach of Lauvergnat *et al.*[28] already in earlier work. Thus, this package can also treat arbitrary molecular systems in Z-matrix representations without needing to change the code.

With application examples in 2D and 3D, we have demonstrated that this new proDG code saves huge amounts of memory and computer time compared to the traditional FBR approach from which it is derived. When using appropriate parameter settings, we can even get close to the highly efficient FFT or mixed FFT/DVR approaches, without significant loss in accuracy. Given the decreasing efficiency of necessarily rectangular FFT grids, this encourages us that proDG might even beat the FFT approach for higher-dimensional examples. Of course, other approaches like non-direct-product DVRs or MCTDH variants have been shown to be efficient roads towards higher-dimensional applications. Nevertheless, we believe to have captured additional aspects here that are worthy of some attention. Clearly, this confidence rests upon the wavepacket not spreading out equally in all dimensions. However, physical intuition tells us that exactly this is to be expected in chemical reactions. After all, chemical reactions are not completely "messy" but so orderly that organic and inorganic chemists could come up with large sets of empirical rules that predict a few active degrees of freedom and many spectators.

As additional aspect, we should mention that in this work all potential energy surfaces were available analytically, which minimizes the calculation time of the potential energy matrix. In real life applications, however, the PES is not known beforehand. With the proDG approach applied on-the-fly, the region where the potential is needed is automatically minimized and solely determined by the dynamics of the wavepacket. Thus, computational times needed for PES determination by quantum chemical programs are reduced compared to standard approaches in which a full rectangular grid is needed, and also compared to more refined approaches where wavefunction spread is estimated *a priori*. At the same time, quantum chemical calculations are automatically avoided in regions of unphysically high energies, where difficult SCF convergence behavior is to be expected.

To exploit this feature in practical applications, we are currently implementing on-the-fly calculations of the potential with suitable interpolation schemes, and more advanced sparse matrix representations which take advantage the structure of the Hamiltonian in combination with parallelized, distributed computing.

Acknowledgements

This work was inspired and supported by the trans-disciplinary Computational Sciences Center at the University of Kiel. Financial support by the German Science Foundation (DFG) *via* grant Ha2498/6 and the Collaborative Research Center SFB 677 is gratefully acknowledged.

References

1 H. D. Meyer, U. Manthe and L. S. Cederbaum, *Chem. Phys. Lett.*, 1990, **165**, 73.
2 A. D. Hammerich, R. Kosloff and M. A. Ratner, *Chem. Phys. Lett.*, 1990, **171**, 97.
3 H.-D. Meyer and G. A. Worth, *Theor. Chem. Acc.*, 2003, **109**, 251.
4 F. Huarte-Larrañaga and U. Manthe, *J. Chem. Phys.*, 2000, **113**, 5115.
5 O. Vendrell, F. Gatti, D. Lauvergnat and H.-D. Meyer, *J. Chem. Phys.*, 2007, **127**, 184302.
6 O. Vendrell, F. Gatti and H.-D. Meyer, *J. Chem. Phys.*, 2007, **127**, 184303.
7 H. Wang and M. Thoss, *J. Chem. Phys.*, 2003, **119**, 1289.
8 M. Thoss and H. Wang, *Chem. Phys.*, 2006, **322**, 210.
9 R. Kosloff, *Dynamics of molecules and chemical reactions*, Marcel Dekker Inc., New York, 1996, ch. 5, p. 185.
10 L. R. Pettey and R. E. Wyatt, *Chem. Phys. Lett.*, 2006, **424**, 443.
11 L. R. Pettey and R. E. Wyatt, *Int. J. Quantum Chem.*, 2007, **107**, 1566.
12 M. Ben-Nun and T. J. Martinez, *J. Chem. Phys.*, 2000, **112**, 6113.
13 D. Lauvergnat, E. Baloïtcha, G. Dive and M. Desouter-Lecomte, *Chem. Phys.*, 2006, **306**, 500.
14 G. D. Billing, *The Quantum Classical Theory*, Oxford University Press, Oxford, 2003.
15 P. Puzari, B. Sarkar and S. Adhikari, *J. Chem. Phys.*, 2004, **121**, 707.
16 R. E. Wyatt, *J. Chem. Phys.*, 2002, **117**, 9569.
17 Y. Goldfarb, I. Degani and D. J. Tannor, *J. Chem. Phys.*, 2006, **125**, 231103.
18 Y. Goldfarb and D. J. Tannor, *J. Chem. Phys.*, 2007, **127**, 161101.
19 S. S. Iyengar and J. Jakowski, *J. Chem. Phys.*, 2005, **122**, 114105.
20 V. Gradinaru, *Computing*, 2007, **80**, 1.
21 D. T. Colbert and W. H. Miller, *J. Chem. Phys.*, 1992, **96**, 1982.
22 R. Dawes and T. Carrington, Jr, *J. Chem. Phys.*, 2004, **121**, 726.
23 Z. Bacic and J. C. Light, *J. Chem. Phys.*, 1986, **85**, 4594.
24 R. Dawes and T. Carrington, Jr, *J. Chem. Phys.*, 2005, **122**, 134101.
25 R. Dawes and T. Carrington, Jr, *J. Chem. Phys.*, 2006, **124**, 054102.
26 B. Hartke, *Phys. Chem. Chem. Phys.*, 2006, **8**, 3627.
27 D. A. McCormack, *J. Chem. Phys.*, 2006, **124**, 204101.
28 D. Lauvergnat and A. Nauts, *J. Chem. Phys.*, 2002, **116**, 8560.
29 C. Leforestier, R. H. Bisseling, C. Cerjan, M. D. Feit, R. Friesner, A. Guldberg, A. Hammerich, G. Jolicard, W. Karrlein, H.-D. Meyer, N. Lipkin, O. Roncero and R. Kosloff, *J. Comput. Phys.*, 1991, **94**, 59.
30 I. P. Hamilton and J. C. Light, *J. Chem. Phys.*, 1986, **84**, 306.
31 V. Maz'ya and G. Schmidt, *IMA J. Num. Anal.*, 1996, **16**, 13.
32 V. Maz'ya and G. Schmidt, *Approximate Approximations*, AMS, 2007.
33 V. Maz'ya and G. Schmidt, *Appl. Comp. Harm. Anal.*, 1999, **6**, 287.
34 S. Mallat, *A wavelet tour of signal processing*, Academic Press, London, 2nd edn, 1999.
35 R. A. Friesner, *J. Chem. Phys.*, 1986, **85**, 1462.
36 W. Yang and A. C. Peet, *Chem. Phys. Lett.*, 1988, **153**, 98.
37 S. Pissanetsky, *Sparse Matrix Technology*, Academic Press, London, 1984.
38 E. Steffen, Master's thesis, University of Kiel, 2008.
39 M. R. Brill, F. Gatti, D. Lauvergnat and H.-D. Meyer, *Chem. Phys.*, 2007, **338**, 186.
40 B. Podolsky, *Phys. Rev.*, 1928, **32**, 812.
41 H. Margenau and G. M. Murphy, *The Mathematics of Physics and Chemistry*, Van Nostrand, Princeton, NJ, 1956.
42 A. Nauts and X. Chapuisat, *Mol. Phys.*, 1985, **55**, 1287.
43 X. Chapuisat, A. Belafhal and A. Nauts, *J. Mol. Spectrosc.*, 1991, **149**, 274.
44 H. F. von Horsten, PhD thesis, University of Kiel, 2008.
45 H. F. von Horsten and B. Hartke, *Chem. Phys.*, 2007, **338**, 160.
46 H. F. von Horsten, G. Rauhut and B. Hartke, *J. Phys. Chem. A*, 2006, **110**, 13014.
47 D. Kosloff and R. Kosloff, *J. Comput. Phys.*, 1983, **52**, 35.
48 M. D. Feit, J. A. Fleck, Jr and A. Steiger, *J. Comput. Phys.*, 1982, **47**, 412.
49 S. K. Gray and D. E. Manolopoulos, *J. Chem. Phys.*, 1996, **104**, 7099.
50 R. Kosloff, *Numerical grid methods and their applications to Schrödinger's equation*, Kluwer, Dordrecht, 1993, vol. 412 of *NATO ASI Series C*, chapter titled The Fourier Method, p. 175.
51 E. Baloïtcha, B. Lasorne, D. Lauvergnat, G. Dive, Y. Justum and M. Desouter-Lecomte, *J. Chem. Phys.*, 2002, **117**, 727.
52 B. Lasorne, F. Gatti, E. Baloitcha, H.-D. Meyer and M. Desouter-Lecomte, *J. Chem. Phys.*, 2004, **121**, 644.
53 D. Sölter, H.-J. Werner, M. von Dirke, A. Untch, A. Vegiri and R. Schinke, *J. Chem. Phys.*, 1992, **97**, 3357.
54 D. Manolopoulos, Talk at the "*Charles Coulson Summer School in Theoretical Chemistry*", Oxford, 1996.
55 J.-Y. Ge and J. Z. H. Zhang, *J. Chem. Phys.*, 1998, **108**, 1429.