CPC 50th anniversary article

Quantics: A general purpose package for Quantum molecular dynamics simulations☆

G.A. Worth

Department of Chemistry, University College London, 20, Gordon St., WC1H 0AJ, UK

ABSTRACT

Quantics is a general purpose program package to simulate the time-evolution of a molecular system by solving the time-dependent Schrödinger equation. The main code is based on the multi-configurational time-dependent Hartree (MCTDH) algorithm in various variants, including the powerful multilayer-MCTDH algorithm that has been used to propagate a wavefunction for up to 1000 degrees of freedom. MCTDH uses a contraction of traditional discrete basis set representations of the Hamiltonian and wavefunction, and Quantics includes a range of variable representation (DVR) grid basis sets and collocation methods. Input is via ascii text files and for molecules with analytical potential functions no programming is required. A library of potential functions is included to treat more complicated cases, and more functions can be added as required by the user. The code also includes the variational multi-configurational Gaussian (vMCG) method that is based on a Gaussian wavepacket expansion of the wavefunction. vMCG can be run in a "direct" manner (DD-vMCG), calculating the potential energy surfaces on-the-fly using a number of quantum chemistry programs. In addition to wavepacket propagation, Quantics can solve the time-independent Schrödinger equation for small systems and can solve the Liouville–von-Neumann equation to propagate density matrices. The Package includes auxiliary programs to help set up calculations and analyse the output. Quantics is a community code of the UK Collaborative Computational Project for Quantum Dynamics (CCPQ) and the European E-CAM project, an e-infrastructure for software development run by the Centre Européen de Calcul Atomique et Moléculaire (CECAM). Through this it has become a framework for general dynamics codes, for example enabling an external surface hopping code to use the Quantics input and operator interfaces.

Program summary

Program Title: Quantics
Program Files doi: http://dx.doi.org/10.17632/x9dcpc2r5c.1
Licensing provisions: LGPLv3
Programming language: Fortran90. Some Fortran77, Fortran2003, C and python.
Nature of problem: Solving the time-dependent Schrödinger equation for a set of nuclei allows a range of physical processes to be studied including all quantum effects. This allows an experimental signal to be given a molecular interpretation. Typical applications are scattering cross-sections or time-resolved spectra, but also rate constants and other transport properties are possible. The exact problem to be solved is defined by the Hamiltonian, which must be provided by the user, and the initial wavepacket, again defined by the user. The final analysis of the evolving wavepacket then provides the experimental signal or molecular property.
Solution method: A range of methods are possible for solving the time-evolution of a wavepacket (see main text). These can be broadly described as basis-set methods, in which the wavepacket and Hamiltonian are expanded in a set of functions. Various functions are possible, including grid-based sets (DVRs and collocation), and Gaussian wavepackets. The wavepacket can then be propagated using a variety of algorithms depending on the representation chosen. These include the full numerically-exact solution, various versions of the multi-configurational time-dependent Hartree method and approximate methods such as trajectory surface hopping. Full details are given in the documentation provided with the package and in a book and a number of review articles [1,2,3].
Additional comments including restrictions and unusual features: The code has been tested on a number of linux distributions and compilers. It works best with a bash environment and a gnu gcc / gfortran

---

compiler greater than version 4.8. The code is parallelised in parts using either OpenMP or MPI. There is a suite of test calculations to test an installation.

References

[1] M. H. Beck, A. Jäckle, G. A. Worth, H.-D. Meyer, The multiconfiguration time-dependent Hartree method: A highly efficient algorithm for propagating wavepackets., Phys. Rep. 324 (2000) 1–105.

[2] H.-D. Meyer, G. A. Worth, Quantum molecular dynamics: propagating wavepackets and density operators using the multiconfiguration time-dependent Hartree method, Theo. Chem. Acc. 109 (2003) 251–267.

[3] H.-D. Meyer, F. Gatti, G. A. Worth, Multidimensional Quantum Dynamics: MCTDH Theory and Applications, Wiley-VCH, Weinheim, Germany, 2009.

1. Introduction

The QUANTICS Package is a suite of programs for simulating the non-equilibrium time-evolution of a molecular system including all quantum effects. The key aim is to solve the time-dependent Schrödinger equation as completely and as accurately as possible for a wide range of physical processes. The time-dependent Schrödinger equation (TDSE) is written as

$$i\frac{\partial}{\partial t}\Psi(\mathbf{q}, t) = H\Psi(\mathbf{q}, t) \tag{1}$$

$$= (T + V(\mathbf{q}))\Psi(\mathbf{q}, t) \tag{2}$$

where the molecular wavefunction, Ψ, is a function of the nuclear coordinates, $\mathbf{q}$, and the Hamiltonian operator, H, is composed of the nuclear kinetic energy T and potential function V. More generally, and especially for the simulation of photo-excited molecules, the system is described by a set of coupled potential surfaces and the TDSE is

$$i\frac{\partial}{\partial t}\boldsymbol{\Psi}(\mathbf{q}, t) = (T\mathbf{1} + \mathbf{W}(\mathbf{q}))\boldsymbol{\Psi}(\mathbf{q}, t) \tag{3}$$

where $\mathbf{W}$ is a matrix of potentials and couplings in a diabatic picture and the nuclear wavefunction Ψ is a vector with a component associated with each state.

The TDSE is a first-order differential equation and the physical system is specified by the coordinates and the potential function. It is an initial value problem and so the process to be simulated is defined by the chosen initial conditions. Typical examples of processes are branching ratios and cross-sections for molecular scattering, photo-dissociation and photo-excitation. The latter leads to an absorption or emission spectrum. By inclusion of a time-dependent field in the Hamiltonian operator to simulate a light pulse, time-resolved spectra can be simulated.

The TDSE describes a single *pure* state formed as a coherent superposition of eigenstates. Statistical mixtures (thermalised states) are better described using a density matrix representation and the Liouville–von Neumann equation (LvN)

$$i\frac{\partial}{\partial t}\rho(\mathbf{q}, \mathbf{q}', t) = \mathcal{L}(\rho) \tag{4}$$

where $\rho(\mathbf{q}, \mathbf{q}', t) = |\Psi(\mathbf{q}, t)\rangle\langle\Psi(\mathbf{q}', t)|$ is the density matrix and $\mathcal{L}(\rho)$ the Liouvillian superoperator. For a closed system, the Liouvillian is the commutator of the system Hamiltonian with the density matrix,

$$\mathcal{L}(\rho) = [H, \rho] \quad . \tag{5}$$

For open systems, the "system" is coupled to an implicit environment. There are a number of ways to treat this coupling, the most common being a Lindblad operator, which changes the Liouvillian to read

$$\mathcal{L}(\rho) = [H, \rho] + i\left(V\rho V^\dagger - \tfrac{1}{2}\rho V^\dagger V - \tfrac{1}{2}V^\dagger V\rho\right) \quad . \tag{6}$$

In the following, the focus is on solving the TDSE, but extensions that can be used to solve the LvN are also available in QUANTICS.

The most direct and accurate way to solve the TDSE is to use a time-independent basis set to represent the wavefunction and Hamiltonian operator, effectively discretising the problem onto a multi-dimensional grid. The initial wavefunction is setup as a non-stationary state, a wavepacket, that is propagated in time using an integration scheme. The evolving wavepacket can then be analysed to obtain the results: a property that can be related to an experimental signal, or an expectation value that can be related to the molecular evolution.

A numerically exact solution to the TDSE is obtained by representing the wavepacket directly as a set of time-dependent coefficients of the time-independent basis functions, which can be seen as amplitudes on the grid points. This is referred to in the following as "standard" wavepacket dynamics, or the *standard WP method*. If there are N basis functions for each coordinate (degree of freedom), and f degrees of freedom, the multi-dimensional grid has N^f points, and the computational effort scales exponentially with system size. This restricts the solution in general to 4 or 5 degrees of freedom. To go to larger systems, the *multi-configurational time-dependent Hartree* (MCTDH) method is used. In this, the wavepacket is expressed in a set of time-dependent basis functions, known as *single-particle functions* (SPFs), which in turn are described using the time-independent *primitive basis functions*. MCTDH is thus a contraction scheme and in its most powerful form, *multi-layer MCTDH* (ML-MCTDH), allows the time-evolution of wavepackets to be simulated for over 100 degrees of freedom. The key to the MCTDH method is that the equations of motion are variational and so convergence of the basis set results in the numerically exact solution.

The MCTDH method in a number of variants is at the heart of the QUANTICS package. The package aims to be a general purpose code, with a user friendly interface: for most calculations the input is entirely given in ascii text files. It is portable, being written for the most part in standard Fortran and tested with readily available compilers. It comes with an extensive set of documentation and a user manual.

In this article, an overview of the code will be given. The theory behind the algorithms has been published elsewhere and only a brief description of the methods available is given, with references to the literature. After a listing of the code capabilities, a brief history of its development is given. Many people over the years have been involved in the code development, and here the major developments are described and credited. Unfortunately there is not space here to credit all involved, but this is done in the code documentation. The relationship of the package to the Heidelberg MCTDH Package [1], from which it evolved is also made clear. After this, in Section 4 the basic equations being solved are given. The final sections describe the code structure (Section 5) and the basic usage (Section 6) to give a flavour of how the code is built up and how it can be used.

3

Table 1
Main methods available in the QUANTICS Program.

Method	Equation solved	Basis sets
Standard WP	TDSE	DVRs or FFT
MCTDH	TDSE	DVRs or FFT
ML-MCTDH	TDSE	DVRs or FFT
GMCTDH	TDSE	DVRs or FFT or GWPs
ML-GMCTDH	TDSE	DVRs or FFT or GWPs
ρMCTDH(I)	LvN	DVRs or FFT
ρMCTDH(II)	LvN	DVRs or FFT
ML-ρMCTDH(I)	LvN	DVRs or FFT
ML-ρMCTDH(II)	LvN	DVRs or FFT
ρGMCTDH(II)	LvN	DVRs or FFT or GWPs
ML-ρGMCTDH(II)	LvN	DVRs or FFT or GWPs
Standard WP	TISE	DVRs or FFT
vMCG	TDSE	GWPs
clMCG	TDSE	GWPs
iMCG	TDSE	GWPs
DD-vMCG	TDSE	GWPs
TSH	TDSE	trajectories
DD-TSH	TDSE	trajectories

Integration schemes	Method
Chebyshev	Standard WP
Second Order Differencing	Standard WP
Split Operator	Standard WP
Constant Mean Field (CMF)	MCTDH (not ML-)
Adams–Bashforth–Moulton	all MCTDH and MCG
Burlisch–Stoer	all MCTDH and MCG
Runge–Kutta	all MCTDH and MCG

DVRs:
Harmonic Oscillator, Sine, Cosine, Exponential, Legendre, Laguerre

2D spherical harmonic basis sets:
Spherical Harmonic FBR, Extended Legendre, 2D Legendre

2. Capabilities

The methods available in the QUANTICS Package to simulate the non-equilibrium time-evolution of distinguishable particles are listed in Table 1. In addition, to propagation in real time, propagation in imaginary time is also available for most methods. This procedure, known as *energy relaxation*, can be used to either obtain a single eigenfunction, or a set of eigenfunctions using a *block relaxation* scheme. There is also the possibility of the direct diagonalisation of the Hamiltonian matrix in a grid-basis to directly obtain the eigenvalues. For a full list of programs available in the QUANTICS Package and their capabilities, see the documentation that is provided with the code. This is in the form of html files describing the codes and options, and a user guide with a short tutorial that is provided as a pdf file.

3. History of the program

The QUANTICS Package started in the group of Lorenz Cederbaum at the University of Heidelberg. The initial code was written by Uwe Manthe as part of his PhD to run the very first MCTDH calculations [2,3]. This was a set of routines in Fortran77 that required compilation for each calculation to select the relevant potential surface, integration scheme and other options. Under the supervision of Hans-Dieter Meyer, together with Andreas Jäckle and Michael Beck, I put these routines together as a single program that could select the different options and operators using ascii input files. The program then grew into the Heidelberg MCTDH Package [1] able to solve the time-dependent Schrödinger equation for distinguishable particles using both the grid-based MCTDH algorithm as well as numerically exact calculations. A key design feature from the start was treating molecules of any size (from diatomic to 10 atoms or more), modularity, and ease of use.

A suite of stand-alone programs to analyse the output was also started including automatic plotting using the GNUPLOT program.

The initial code included innovations for scattering calculations such as an adiabatic correction to shorten the asymptotic grid [4] and the use of complex absorbing potentials (CAPs) to calculate the reactive flux [5]. The *filter diagonalisation* time-dependent approach to obtain the eigenvalues of a Hamiltonian was added from the work of Michael Beck, helped by Fabien Gatti [6,7], along with time-independent calculations diagonalising the Hamiltonian matrix. The ability to propagate a density matrix rather than a wavepacket, including thermalisation or dissipative environments, was added from the work of Andreas Raab and Irene Burghardt [8,9].

One of the drawbacks of the MCTDH method is that it requires a particular form for the Hamiltonian. For efficiency, this operator must be written as a sum of products of one-dimensional operators. This is not generally the case for molecular potentials and Andreas Jäckle wrote the POTFIT program that automatically fits a function into the desired form [10].

Early applications focused on scattering in the gas-phase [11] and on surfaces [12,13]. A novel two-dimensional discrete variable representation (DVR) for the rotational operator allowed systems with total angular momentum greater than zero to be studied [14]. The method was also used to study the other great interest in Heidelberg: studying non-adiabatic systems with the vibronic coupling Hamiltonian [15].

A significant breakthrough was my implementation of general, multi-dimensional basis functions. The resulting saving in computational effort allowed us to converge a calculation of the absorption spectrum of pyrazine including all 24 degrees of freedom [15,16]. Given the exponential scaling of quantum dynamics methods, this was a huge step forward from the 3–4 dimensional systems that had been studied using these methods up to this point. To ease the fitting of vibronic coupling models, the VCHAM set of programs were added to the package by Christopher Cattarius, Andreas Markmann and myself [17,18]. VCHAM was updated more recently to include a range of diabatic potentials and fitting procedures by Simon Neville and Christopher Robertson.

The next developments to be added to the package were the use of Gaussian wavepacket basis sets in place of DVRs — known as the GMCTDH approach. Developed mostly by Irene Burghardt [19], this aims to break the scaling problem further. This method, however, proved to be numerically unstable and it took a number of years before we could get it to work reliably. The initial interest in GMCTDH was due to the possibility of greater efficiency in system-bath problems in which the bath is described by a set of harmonic oscillators for which the Gaussian functions are perfectly adapted. Later interest came from the possibility of using it as a direct dynamics algorithm, in which the potential surfaces are calculated on-the-fly using a quantum chemistry code. This has been a significant part of the development, working together with Mike Robb, Benjamin Lasorne and Mike Bearpark first at King's College London and then Imperial College London [20–22].

At this point, the code was showing its age, being written in standard Fortran77. At Birmingham, Kousik Giri then worked with me on converting the code to Fortran90. This allowed a better use of dynamic memory allocation (the older code uses C-routines to allocate large blocks of memory). This code became the QUANTICS PACKAGE.

The next big step forward for the capabilities of the Heidelberg MCTDH package was the implementation of ML-MCTDH by Oriel Vendrell [23]. This was written in Fortran95 using derived types to elegantly enable the recursive algorithm required to code this highly efficient method. It has been included into QUANTICS, running faster than in the Fortran77 code due to a simpler interface.

Parallelisation of the MCTDH algorithm was first introduced in the Heidelberg package by Michael Brill using pthreads [24] and later extended to using MPI. In Quantics, we have chosen to use the more readily available OpenMP in place of pthreads, implemented by Gareth Richings. The MCTDH algorithm, however, is not easy to parallelise due to the interconnectivity of the data. Only parallelisation over the terms in the Hamiltonian has been implemented and this is effective when the potential expansion is very long, for example in the benchmark calculations on the Zundel cation [25]. Further improvements are required to improve general efficiency.

Recent developments on the Quantics package have mostly been to the Gaussian wavepacket direct dynamics methods. Gareth Richings, Iakov Polyak and Eryn Spinlove were particularly involved in the implementation of features such as propagation diabatisation [26] and Hessian updating [27]. Complementary developments have been made by Terry Frankcombe to use the Grow methodology to produce potential surfaces from a set of data points [28]. Recently, we have written an interface to a surface hopping code from the group of Nadia Doslic [29] to provide direct comparisons with this popular semi-classical method.

Over the years, other capabilities have been added. Time-dependent fields can be a part of the operator to simulate directly time-resolved spectroscopic signals such as photo-electron spectra [30]. Using this, Alex Brown and Markus Schröder implemented an optimal control scheme [31], and Tom Penfold local control [32]. A library of potential functions has been built up and automatic spline fitting to data added by Cristina Sanz Sanz. Most recently, we have returned to developing the density matrix approach to include thermalisation and environmental effects, and this has led to an implementation of multi-layer versions of the original Raab type II density matrix scheme. This should get around the scaling problem of density matrix methods and open new avenues for simulations. As a Gaussian wavepacket basis can be used, direct dynamics density matrix propagation will also be possible.

Development has continued in Heidelberg on the original MCTDH package led by Hans-Dieter Meyer, who has been actively involved throughout the code development in an inspirational as well as practical capacity. Frank Otto and Markus Schröder were, in particular, involved in the parallelisation and general improvement of this package. There are three versions. Version 8.3 is the standard serial code for grid-based MCTDH calculations. Version 8.4 is the parallel version, and version 8.5 includes ML-MCTDH. These are now mature, well tested codes that are available on request. Quantics includes the functionality of all these versions and the user interface is the same. The structures of the binary files with output from a calculation have been changed to provide greater flexibility, but the reading of files is backward compatible.

Throughout the development of this code, we have used revision control, first PRCS (introduced to us by Stefan Wefing), then SVN, and now Git. A set of automatic tests (known as the Elk Test after the car industry standard) have also been developed to ensure that new additions do not break the code. The package also has extensive documentation, and a user guide, the original version of which was written by Michael Beck.

4. Solving the TDSE: Methods

The details of the algorithms implemented in Quantics are contained in a book [33] and a number of review articles [27,34, 35]. Only a brief overview is given here of the different methods in terms of the ansatz used along with their main properties and citations to original papers. Equations of motion are not given.

4.1. The time-dependent Schrödinger equation

The TDSE is solved using the Dirac–Frenkel variational principle:

$$\langle \delta \Psi | H - i \frac{\partial}{\partial t} | \Psi \rangle = 0 \quad . \tag{7}$$

The different forms of the trial wavefunction define the methods. The variational principle ensures that the total energy and norm of the wavefunction should be conserved.

Standard WP Method. To solve the TDSE, the wavefunction can be expressed as a direct product expansion in a set of (usually one-dimensional) time-independent basis functions [36,37]:

$$\Psi(\mathbf{q}, t) = \sum_{j_1 \dots j_f} A_{j_1 \dots j_f}(t) \chi_{j_1}^{(1)}(q_1) \dots \chi_{j_f}^{(f)}(q_f) \quad , \tag{8}$$

where $\{\chi_{j_\kappa}^{(\kappa)}(q_\kappa)\}$ are the set of functions for the κth degree of freedom and $A_{j_1 \dots j_f}(t)$ the time-dependent expansion coefficients.

The basis functions are either a *discrete variable representation* (DVR) [34,38], or a *collocation* [36]. DVRs are constructed by diagonalising the position operator in an analytic set of functions, such as harmonic oscillator or Legendre functions. This leads to a set of localised functions in coordinate space in which the potential operator is diagonal, while the kinetic energy operator can be expressed analytically in the conjugate basis function space. A number of DVRs are included in Quantics with properties suitable for different coordinates.

A collocation method exactly represents the wavefunction at pre-defined grid points. The potential energy operator is again taken as diagonal on the grid points and the kinetic energy operator is evaluated by making use of a Fourier Transform to take the wavefunction onto the conjugate momentum grid, where the momentum is diagonal, followed by a reverse Fourier Transform. For this reason it is referred to as an FFT basis.

The main effort in a wavepacket propagation calculation is the need to evaluate integrals of the Hamiltonian in the basis set, i.e.

$$H_{IJ} = \langle \Phi_I | H | \Phi_J \rangle \quad , \tag{9}$$

where $\Phi_J = \chi_{j_1}^{(1)} \dots \chi_{j_f}^{(f)}$ is a *configuration* with the multi-index J. The combination of localised functions with easy kinetic energy evaluation makes DVRs and collocation methods efficient basis sets.

Both DVR and FFT bases effectively discretise the problem and methods using these are referred to as *grid-based*. The wavefunction expansion coefficients in the standard WP method are therefore amplitudes at the grid points. The standard WP method is simple and, as long as there are enough grid points, a numerically exact solution of the TDSE. It can also be used together with powerful integration schemes, such as split-operator or Chebyshev, to allow efficient propagation. It does, however, suffer from the exponential increase in the number of expansion coefficients with system size, i.e.

$$\text{computational effort} \sim N^f \quad , \tag{10}$$

where N is the basis set size for each degree of freedom. This limits the standard WP method to 3 or 4 degrees of freedom without the use of advanced DVR basis sets or grid pruning techniques [39].

MCTDH. In contrast to the standard WP method, the MCTDH wavefunction is expanded in a time-dependent basis, referred to as *single-particle functions* (SPFs) [2,34]

$$\Psi(\mathbf{q}, t) = \sum_{j_1 \dots j_p} A_{j_1 \dots j_p}(t) \varphi_{j_1}^{(1)}(Q_1, t) \dots \varphi_{j_p}^{(p)}(Q_p, t) \quad , \tag{11}$$

which are in turn expanded in a time-independent DVR or FFT basis which is referred to as the *primitive basis*

$$\varphi_j^{(\kappa)}(Q_\kappa, t) = \sum_i a_{ij}(t)\chi_i^{(\kappa)}(Q_\kappa) \quad . \tag{12}$$

The variational nature of the SPFs allows a more compact wavefunction expansion as fewer basis functions are needed than grid-points to follow the evolving wavepacket. Convergence on the numerically exact result is also guaranteed.

A key feature is that the SPFs can be multi-dimensional, i.e. rather than using f sets of one-dimensional functions, p sets of d-dimensional SPFs with coordinates $Q_1 = (q_1, q_2, \ldots, q_d)$ are used. This reduces the exponent for the scaling of the number of coefficients. The scaling is

$$\text{effort} \sim n^p + pnN^d \quad , \tag{13}$$

where n is the SPF basis set size and N the underlying primitive basis set size. The first term on the right hand side of Eq. (13) is the number of expansion coefficients, while the second term is the size of the d-dimensional grids describing the SPFs. There is thus an optimum balance between combining degrees of freedom together in a particle to reduce the first term, while not combining too many together to prevent the second term becoming too large. If all degrees of freedom are combined together the multi-dimensional SPF becomes the wavefunction of the standard WP method.

MCTDH can be used for systems with up to 20 degrees of freedom. The main bottleneck is the fact that, unlike in the standard WP method, the Hamiltonian matrix elements are time-dependent and need to be re-evaluated at each time step. This is prohibitively expensive unless the potential is a sum of products with the same (or lower) dimensions as the SPFs, i.e.

$$V(q_1, \ldots, q_f) = \sum_r c_r h_r^{(1)}(Q_1) \ldots h_r^{(p)}(Q_p) \quad , \tag{14}$$

which means that only one-dimensional integrals are required. A potential that is not a priori in this form can be efficiently fit using the POTFIT program that implements a procedure to optimally describe a potential in the desired sum of products form [10]. Alternatively, the VCHAM programs can be used to fit a vibronic coupling model potential suitable for the description of non-adiabatic photophysics [17].

For systems with more than one electronic state, two different approaches are possible. In the *multi-set* formalism, a different set of SPFs is used for each electronic state:

$$\Psi(\mathbf{q}, t) = \sum_s \sum_{j_1^s \ldots j_f^s} A_{j_1^s \ldots j_p^s}^{(s)}(t)\varphi_{j_1}^{(s,1)}(Q_1, t) \ldots \varphi_{j_p}^{(s,p)}(Q_p, t) \quad , \tag{15}$$

where s is the index over electronic states. In contrast, in the *single-set* formalism only one set of SPFs is used and the electronic states are explicitly included as a vector

$$\Psi(\mathbf{q}, t) = \sum_s \sum_{j_1 \ldots j_p} A_{j_1 \ldots j_p, s}(t)\varphi_{j_1}^{(1)}(Q_1, t) \ldots \varphi_{j_p}^{(p)}(Q_p, t)|s\rangle \quad . \tag{16}$$

The multi-set formalism requires more SPFs, but as they can adapt optimally to each state fewer configurations in total are required which can save effort.

ML-MCTDH. Larger systems can be treated by realising that multi-dimensional functions can be treated in an MCTDH form. This leads to a recursive layering structure, multi-layer MCTDH [23,40–42]

$$\Psi(\mathbf{q}, t) = \sum_{j_1 \ldots j_p} A_{j_1 \ldots j_p}^1(t)\varphi_{j_1}^{(1:1)}(Q_1^1, t) \ldots \varphi_{j_p}^{(1:p)}(Q_p^1, t)$$

$$\varphi_m^{1:\kappa}(\mathbf{Q}_\kappa^1, t) = \sum_{j_1 \ldots j_{d_\kappa}} A_{m:j_1 \ldots j_{d_\kappa}}^{2:\kappa}(t)\varphi_{j_1}^{(2:\kappa,1)}(Q_1^{2:\kappa}, t) \ldots \varphi_{j_{d_\kappa}}^{(2:\kappa, d_\kappa)}(Q_{d_\kappa}^{2:\kappa}, t)$$

$$\vdots \quad \vdots \tag{17}$$

$$\varphi_m^{l:\kappa}(\mathbf{Q}_\kappa^l, t) = \sum_{j_1 \ldots j_{d_\kappa}} A_{m:j_1 \ldots j_{d_\kappa}}^{l+1:\kappa}(t)\varphi_{j_1}^{(l+1:\kappa, 1)}(Q_1^{l+1:\kappa}, t) \ldots$$
$$\varphi_{j_{d_\kappa}}^{(l+1:\kappa, d_\kappa)}(Q_{d_\kappa}^{l+1:\kappa}, t)$$

where there is new superscript, l, referring to the layer. The lowest layer of SPFs is described by a grid as in MCTDH. Only the single-set formalism is possible with ML-MCTDH.

ML-MCTDH provides a very powerful tensor contraction of a multi-dimensional wavefunction and it has been used for calculations with hundreds of degrees of freedom [43,44]. In the QUANTICS implementation, all primitive basis sets and Hamiltonians that can be used with MCTDH can also be used with ML-MCTDH. The main drawback of the method is that it can be difficult to converge calculations, as all sets of functions in the layers need to be balanced. An initial contraction scheme also needs to be decided on, known as a *tree structure*. Poor choices lead to calculations that are harder to converge. The variational nature of the method, however, ensures that convergence leads to the numerically exact result.

GMCTDH. Rather than using grid-based SPFs, it is possible to use parametrised basis functions for some of the particles and propagate the parameters instead [19]

$$\Psi(\mathbf{q}, t) = \sum_{j_1 \ldots j_p} A_{j_1 \ldots j_p}(t)\Psi(\mathbf{q}, t) = \varphi_{j_1}^{(1)}(Q_1, t) \ldots \varphi_{j_m}^{(m)}(Q_m, t)$$
$$\times g_{j_{m+1}}^{(m+1)}(Q_1, t) \ldots g_{j_p}^{(p)}(Q_p, t) \quad , \tag{18}$$

where $g^{(\kappa)}$ are parametrised functions. While the formalism is general, at present Gaussian basis functions (GBF) have been implemented with the form

$$g(\mathbf{Q}, t) = \exp(\mathbf{Q}^T \mathbf{A}\mathbf{Q} + \mathbf{B}^T \mathbf{Q} + \mathbf{C}) \tag{19}$$

with the complex variational parameters $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$. If the width matrix, $\mathbf{A}$, is kept fixed, and these *frozen* Gaussians remain separable, these basis functions can be related to one-dimensional *Gaussian wavepackets*

$$g(q, t) = \exp(-\sigma(q - q_0)^2 + ip_0(q - q_0) + i\gamma) \tag{20}$$

with width σ, centre q_0, momenta p_0 and phase γ. The use of GBFs has two advantages over flexible grid-based SPFs. The first is that, while more basis functions will be required for convergence, the number of parameters that must be propagated will be reduced. The second is that as the basis functions are localised, information on the potential is only needed around the centre of the GBFs. Thus, for example, the integrals of an operator in the Gaussian basis can be analytically obtained using the *local harmonic approximation* (LHA) in which the potential is expanded to second order around the GBF centres. This allows general potential functions to be used, i.e. those not in sum of products form Eq. (14), at the expense of no longer having exact integrals.

The GMCTDH equations of motion are numerically unstable due to the non-orthogonal nature of the basis, which can be overcomplete. This can lead to linear dependencies with corresponding integrator problems. The method, however, has been shown to be very efficient for problems in which a quantum system is connected to a bath of harmonic oscillators [45,46].

vMCG. In the limit of only multi-dimensional frozen GBFs, the wavefunction ansatz is a superposition of GWPs [47]

$$\Psi(\mathbf{q}, t) = \sum_j A_j g_j(\mathbf{q}, t) \tag{21}$$

This is known as the variational multiconfiguration Gaussian method [27,47]. It can be related to other GWP approaches such

as Multiple Spawning [48] or Coherent Coupled States [49]. It can be shown that the variational equations of motion for the GWP centres in vMCG are trajectories that can be written as a classical part and a coupled "quantum" part. This coupling leads to much faster convergence than other GWP methods in which the GWPs follow classical trajectories [27].

clMCG. If the coupling between GWPs in the vMCG method is ignored, then the basis functions follow classical trajectories. clMCG is computationally faster and numerically more stable than vMCG, but has much slower convergence on the full quantum mechanical result and is dependent on the initial conditions chosen for a calculation [50].

iMCG. In the simplest variant of the method, the evolving wavefunction is represented by a superposition of functions and the initial weights of the GWPs are kept fixed. The *independent multi-configurational Gaussian* method is thus equivalent to solving the TDSE with a swarm of classical trajectories [50].

All of the MCG methods can be used in a *direct dynamics* mode [20,51,52]. In this, the potential functions are evaluated on-the-fly only when required using quantum chemistry programs. The methods are then referred to as DD-vMCG, DD-clMCG and DD-iMCG, respectively. The Quantics program has interfaces to a number of quantum chemistry codes including Gaussian [53], Molpro [54], Qchem [55] and Molcas [56].

4.2. The Liouville von Neumann equation

The LvN is solved using the Hilbert–Schmidt scalar product in place of the usual Hilbert space scalar product

$$\langle\langle A|B\rangle\rangle = Tr A^\dagger B \qquad (22)$$

and the associated Dirac–Frankel variational principle that reads

$$\langle\langle \delta\rho|i\dot\rho - \mathcal{L}(\rho)\rangle\rangle = 0 \quad , \qquad (23)$$

where $\delta\rho$ are variations in a trial density matrix [8]. There are two basic different trial density operators.

ρ**MCTDH (I).** The type I density operators have the form

$$\rho(\mathbf{q}, \mathbf{q}', t) = \sum_{j_1\ldots j_p} A_{j_1\ldots j_p}(t)\sigma_{j_1}^{(1)}(Q_1, Q_1', t)\ldots\sigma_{j_p}^{(p)}(Q_p, Q_p', t) \quad , \quad (24)$$

where $\sigma^{(\kappa)}$ are *single-particle density operators* (SPDOs). The variational principle then provides equations of motion for the expansion coefficients and the SPDOs.

ρ**MCTDH (II).** In contrast, type II density operators have the form

$$\rho(\mathbf{q}, \mathbf{q}', t) = \sum_{j_1\ldots j_p, k_1\ldots k_p} A_{j_1\ldots j_p, k_1\ldots k_p}(t)|\varphi_{j_1}^{(1)}(Q_1, t)\rangle\langle\varphi_{k_1}^{(1)}(Q_1', t)|\ldots$$
$$\times |\varphi_{j_p}^{(p)}(Q_p, t)\rangle\langle\varphi_{k_p}^{(p)}(Q_p', t)| \qquad (25)$$

where $\varphi^{(\kappa)}$ are analogous to the usual SPFs of MCTDH.

The properties of ρMCTDH(I) and ρMCTDH(II) have been evaluated [9,57] and while type(I) has the advantage of a quicker convergence, particularly for thermalised systems, it does not retain the total energy or norm by construction. Density matrix propagation requires significantly more effort than wavepacket propagation, but in the Quantics code it may be possible to use the multi-layer formalism or GWP basis sets to alleviate this.

4.3. The time-independent Schrödinger equation

For small systems (2–3 degrees of freedom), it is possible to obtain the eigenvalues of a Hamiltonian by diagonalising its matrix representation in a DVR. For this the Lanczos algorithm is used. In addition, a number of ways to obtain eigenvalues are

Table 2
Main programs in the Quantics Package.

Program	Description
quantics	Main program solving the TDSE, LvN and TISE for a wide range of Hamiltonians and initial conditions.
potfit	Transform a potential function into a sum of product form using the *potfit* algorithm [10]
vcham	Sets up and reads output from quantum chemistry calculations to fit the parameters for a Vibronic Coupling Hamiltonian [17]
filter	Employs *filter diagonalisation* to obtain eigenvalues of a Hamiltonian using time-dependent propagation [6]
fdc	As filter, but for complex Hamiltonians
analyse	This is a set of stand-alone programs (presently 40 codes) that can analyse the output from quantics. Using the free GNUPLOT program to provide visualisation, these programs can perform operations such as checking convergence by analysing grid and basis function populations; plotting the potential surfaces and evolving wavepacket; calculating spectra; analysing the database from a DD calculation; Etc.

possible based on propagation methods, and thus able to use the power of the MCTDH wavefunction form.

Energy Relaxation. If a wavepacket is propagated in imaginary time, $it \to \tau$, and at time τ the wavepacket can be expanded in the eigenfunctions of the Hamiltonian as:

$$\Psi(\tau) = \sum_i a_i\psi_i \exp^{-E_i\tau} \qquad (26)$$

where E_i are the eigenvalues and a_1 the contribution of the eigenstate to the initial wavepacket. It can be seen that the contribution of an eigenfunction decays exponentially with (imaginary) time to leave the ground-state eigenfunction at long-times. The algorithm is written to retain normalisation [34].

Block relaxation. To obtain excited-states, i.e. eigenstates other than the ground-state, it is possible to use *improved relaxation* [58]. In this, a combination of imaginary time propagation and diagonalisation of the Hamiltonian in the time-dependent basis allows a set of states to be obtained. The Davidson algorithm is used for the diagonalisation of the low-dimensional Hamiltonian.

Filter diagonalisation. The final method that can be used to obtain eigenenergies uses short real-time propagations and diagonalisation to "filter" out the eigenenergies in a pre-defined energy window [59].

5. Code structure

The Quantics Package is a set of programs, the main ones of which are listed in Table 2. The structure of the central quantics code is shown in Fig. 1 and demonstrates the general philosophy of the programs. The code is broken up into a set of independent modules representing the different stages of a quantum dynamics simulation. Communication between the modules is via files, allowing calculations to be stopped and started at the different stages. The input is given in, usually two, ascii files: These are described below in Section 6. The basic parsing of the input is made in the eingabe module (eingabe is german for input). This defines the type of calculation to be made, where to find the different input required, and basic parameters such as the start and final times for a propagation, of number of iterations for a Hamiltonian matrix diagonalisation.

A quantum dynamics calculation then requires four stages, each of which is dealt with by a different module.

1. The module rundvr "generates the DVR". Here, the system coordinates are defined, setting the number of nuclear
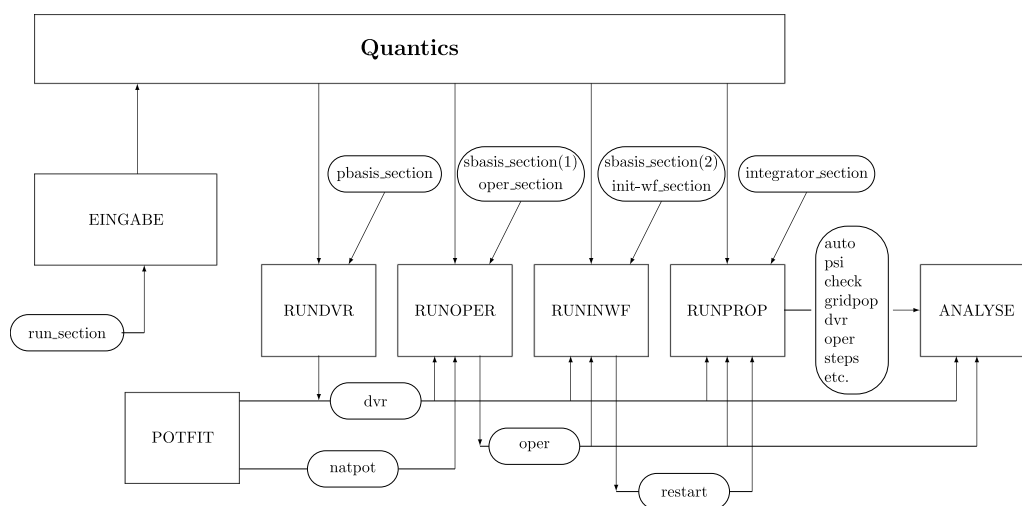
Fig. 1. Structure of the main quantics program. Square boxes represent the code modules. The ovals receiving output from the modules are files, while the ovals providing input to the modules are either files or sections in the input file.

degrees of freedom and number of electronic states. For each degree of freedom, a primitive basis set is defined and the integrals and grid points required to represent a Hamiltonian are calculated. Results are stored in the file dvr.

2. The module runoper "generates the operator". Here the operator information is parsed from the input, and the dvr file read to get the information on the system. An internal table is set up linking the operators in the terms defining the Hamiltonian. The operators are then given types. For example, a kinetic energy operator is usually a real matrix, and a term in the potential a real vector. Finally the operators are calculated. The information is then stored in the file oper. Any combinations of the coordinates into multi-dimensional "particles" in MCTDH calculations are automatically taken care of. The code also tries to optimally combine terms in the Hamiltonian. In addition to the Hamiltonian other operators are set up in an analogous way. These may be required to set up the initial wavefunction or for the analysis.

3. The module runinwf "generates the initial wavefunction". It can also generate density matrices. The module reads from the input information on how to build the initial wavefunction. The dvr and oper files are also read. The result from a previous calculation, e.g. a ground-state vibrational wavefunction from an energy relaxation calculation. If required an operator is applied to the initial guess, e.g. a dipole moment to provide initial excitation. The initial wavefunction is written to the restart file.

4. The module runprop reads the dvr, oper and restart files, and selects the appropriate routines for the asked for propagation. Various files are written with the asked for information. for example the wavefunction stored in the file psi at user specified intervals, the autocorrelation function is stored in auto, and various values such as the energy, norm, expectation value along each coordinate and diabatic state populations in check. These files are in general binary, and can be read by the analysis programs.

The package contains a large number of functions in a library to build up operators. These operators are all indexed and given a name so that they can be found after parsing the operators defined in the ascii input. Operators in the library include both simple operators such as $\frac{\partial^2}{\partial x^2}$, x^n, $\sin(x)$ as well as more complicated operators for angular momenta. Potential functions for particular molecular systems can be added via a interface. The interface is then available to all the programs in the package so that, for example a defined potential surface can be plotted, or a multi-dimensional potential function broken into a sum of products form using the potfit program.

Direct dynamics calculations, in which the potentials are calculated on the fly, store the information from the quantum chemistry calculations in a database (DB). There is a low-level interface to quantum chemistry codes that enables new codes to be used with minimal coding. During a propagation, quantics looks first in the database as to whether the potential surfaces have already been calculated at an appropriate point (i.e. nearby in space to the present point). Only if no suitable points are present will a quantum chemistry calculation be performed. The interface sets up the calculation in an ascii file from a template, runs the calculation and reads the output, storing the new point in the DB. The DB at the end of a calculation is then a representation of the potential surfaces and can be further analysed.

The program structure is designed to be able to provide the input to other programs. For example, the DVR information could be used by reading the dvr file. As a demonstration of the interaction between the package and an external program, an interface has been written to a surface hopping code from the group of Nadja Doslic in Zagreb. Using keywords, the quantics code generates an operator and then calls the Zagreb surface hopping code, which is entirely a stand-alone program as provided by the authors. When a potential energy and gradient is required to drive the trajectories in the swarm, the quantics potential routines are called. These can distinguish between analytic potentials and direct dynamics potentials, so the Zagreb surface hopping is able to use any operator on an even footing to the quantics own propagation methods. At the end of the calculation a short interface then read the surface hopping output and saves it in quantics format to be analysed by the analysis programs. The use of the same DB in either a vMCG or TSH direct dynamics calculations is shown in Fig. 2.

6. Code usage

The code is driven by an ascii input file. Illustrative examples are shown in Figs. 3, 4 and 7 to give a flavour of how a calculation is set up and controlled. For the most part, the text is free-format and case insensitive providing keywords and options. Comments can be added to the file, started with a # symbol. The file is divided into sections, which can go in any order. Each section
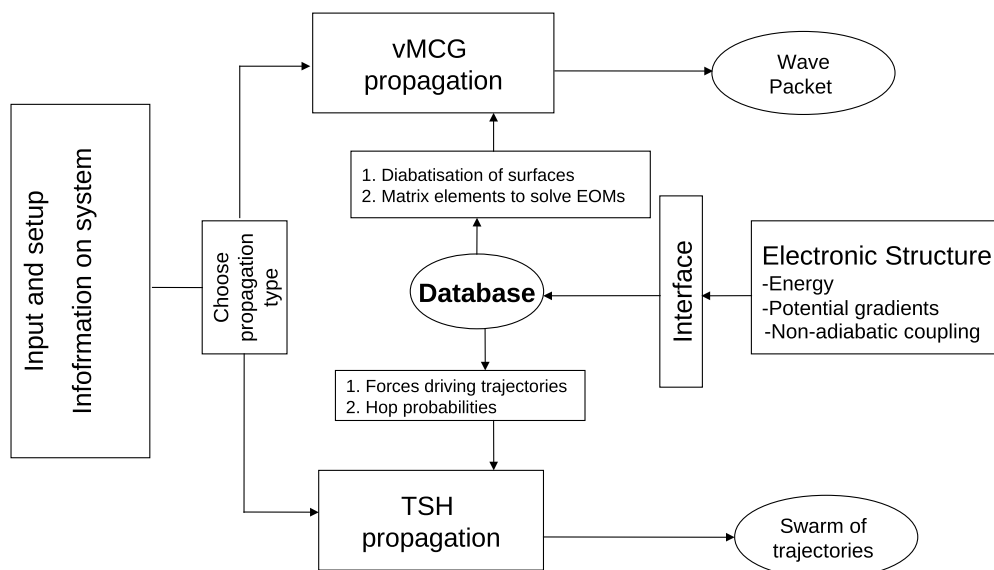
Fig. 2. Communication used by the direct dynamics part of the `quantics` program via a central database.

Table 3
Sections in a `quantics` input file.

RUN-SECTION:	Type of calculation to be made
PRIMITIVE-BASIS-SECTION:	The coordinates, no. grid points etc.
SPF-BASIS-SECTION:	The no. of MCTDH basis functions
OPERATOR-SECTION:	File containing the operator
INIT_WF-BASIS-SECTION:	The initial wavefunction
INTEGRATOR-SECTION:	Details of the integration scheme

reflects a part of a quantum dynamics calculation and are bound by the text XXX-section and end-XXX-section. The most usual sections are listed in Table 3. All the sections and keywords are listed in the package documentation.

The run-section defines the overall calculation, with information on the type of calculation, initial time, final time and the output files to be opened. An important keyword is name = string. This specifies a directory (the "name" directory) into which all the output files from a calculation are written. The output files have simple names such as psi or gridpop rather than the more common use of a filestem with filetype. The system is then defined in the primitive-basis-section. Here, labels are given to each degree of freedom in a list, one degree of freedom per line. The labels are a primary descriptor and any short ascii string can be used. These labels are used in subsequent sections to define the degrees of freedom, and are also used in the output.

In this section, a primitive basis is also defined for each degree of freedom. In Fig. 3 a three-dimensional calculation is specified with degrees of freedom labelled rd, rv and theta. For the rd coordinate an FFT grid is used with 68 equidistant points running from 1.0 au to 9.04 au. For the rv coordinate, a sine DVR is used with 48 points from 0.6 au to 6.24 au, while for the theta coordinate a Legendre DVR is used with 31 grid points based on the even Legendre functions. In Fig. 4, four nuclear degrees of freedom are specified, each using a harmonic oscillator DVR with different numbers of grid points. In addition in this calculation, an electronic "degree of freedom" is specified defining 2 electronic states.

In the spf-basis-section the number of single-particle functions (SPFs) are defined, as required for an MCTDH calculation. The degree of freedom labels are used to map the SPF basis onto the primitive basis. In Fig. 4 a "multi-set" basis is used,

with different functions defining each electronic state. Degrees of freedom can also be combined together in this section and in Fig. 4 the degrees of freedom v1 and v9a are treated together using two-dimensional SPFs. If degrees of freedom specified in the primitive-basis-section are not listed in this section, they are not included in the dynamical calculation, i.e. a cut through the potential surfaces defined by the full system is run. The coordinate for the missing degrees of freedom is set to zero unless otherwise specified using a "point" DVR, which is a specific single point.

The Hamiltonian (and other operators) are usually specified in a separate file. The file name is given in the operator-section. This file is described below, and examples shown in Figs. 5 and 6. Any changes to the parameters or operators in this file can be given here meaning that different calculations, for example changing the mass of an atom to change an isotope, can be run without editing the operator file. In Fig. 3 a complex absorbing potential (CAP) along the degree of freedom rd is added to remove the outgoing flux.

The initial SPFs are defined in the init_wf-section. Various ways of generating this are possible. In Fig. 4 the initial SPFs are harmonic oscillator eigenfunctions for each degree of freedom. In Fig. 3, the initial SPFs for rd are similar to harmonic oscillator functions (Hermite polynomials based on a Gaussian function), while for rv a set of eigenfunctions from a one-dimensional operator are used. This operator is specified in the operator file along with the Hamiltonian (see below). The final, theta, degree of freedom uses Legendre functions.

This information is used to generate the initial wavefunction. For an MCTDH calculation, a vector for the expansion coefficients is set up with the vector index being the multi-index for a configuration of SPFs. Unless otherwise specified, the first coefficient is given a value of 1.0 and all other coefficients are set to zero. In this way, the initial wavepacket is specified as a product of the first SPFs for each degree of freedom. Thus in Fig. 4 the initial wavepacket is a simple separable harmonic oscillator ground-state function (a Gaussian) along each degree of freedom. The initial electronic state is chosen to be the second. This thus simulates a vertical excitation of the ground-state wavefunction in the harmonic approximation. If a numerically exact calculation is performed, the initial wavepacket is set up by multiplying out the first SPFs for each degree of freedom on the full multidimensional grid. The coefficient vector is now a single number.

```
RUN-SECTION
name = hh2
propagation    tfinal = 120.d0    tout = 1.d0    tpsi = 1.d0
psi gridpop steps
title = H+H2 reactive scattering.
end-run-section

OPERATOR-SECTION
opname = h3j0
alter-labels
CAP_rd = CAP [ 6.04  0.002  3 ]   # starting point, strength, order
end-alter-labels
fast     # This speed up of the calculation works only for natpots and CMF.
end-operator-section

SPF-BASIS-SECTION
    rd     =   14
    rv     =   10
    theta  =   10
end-spf-basis-section

PRIMITIVE-BASIS-SECTION
#Label      DVR      N       Parameter
    rd      fft      68   1.00d0     9.040d0    #  xi, xf
    rv      sin      48   0.60d0     6.240d0    #  xi, xf
    theta   leg      31     0        even       # l_z, sym (all/even/odd)
end-primitive-basis-section

INTEGRATOR-SECTION
CMF/var =    0.1 , 1.d-5
BS/spf  =      7 , 1.d-6
SIL/A   =     30 , 1.d-5
end-integrator-section

INIT_WF-SECTION
build
    rd     gauss    4.50d0    -8.00d0      0.25d0    # r0,p0, sigma_r
    rv     eigenf   H2        pop=1                  # pop=1 -> ground state
    theta  leg      0         0            sym       # l_z, l, sym/no-sym
end-build
end-init_wf-section

end-input
```

Fig. 3. Example input file for a calculation of H + H$_2$ scattering in Jacobi coordinates.

In addition to building the initial wavefunction as described above, it is also possible to read the wavefunction from a previous calculation, such as the ground-state vibronic wavefunction from an energy relaxation calculation. Operators, such as a dipole operator, can also be defined and applied to an initial guess before the propagation begins.

The (optional) integrator-section defines the integrator to be used and the associated parameters. Defaults are given for any missing parameters. The input is finished with an end-input statement.

The files defining the operators for these calculations are shown in Figs. 5 and 6. The HAMILTONIAN-SECTION in both files

```
RUN-SECTION
name = pyr4    propagate tfinal = 120.0    tout = 0.5    tpsi= 1.0
psi  auto=twice  steps  gridpop
title =  pyrazine 4-mode multi-set, linear model, no combinations.
end-run-section

OPERATOR-SECTION
opname = pyrmod4
end-operator-section

SPF-BASIS-SECTION
multi-set
     v10a =     7, 7
     v6a  =    12, 11
     v1, v9a   =    6, 5
end-spf-basis-section

PRIMITIVE-BASIS-SECTION
     v10a    HO     22    0.0      1.0      1.0
     v6a     HO     32    0.0      1.0      1.0
     v1      HO     21    0.0      1.0      1.0
     v9a     HO     12    0.0      1.0      1.0
     el      el      2
end-primitive-basis-section

INIT_WF-SECTION
build
   init_state =  2
   v10a HO    0.0      0.0      1.00
   v6a  HO    0.0      0.0      1.00
   v1   HO    0.0      0.0      1.00
   v9a  HO    0.0      0.0      1.00
end-build
end-init_wf-section

end-input
```

Fig. 4. Example input file for a calculation of the absorption spectrum of pyrazine in normal mode coordinates.

defines the Hamiltonian. The operator is coded in a table where each row specifies the term in the operator expansion, and each column specifies the factor for a degree of freedom. This follows the idea that the operator is specified as a sum of products

$$H = \sum_r c_r h_r^{(1)}(q_1) h_r^{(2)}(q_2) h_r^{(3)}(q_3) \dots \qquad (27)$$

The line beginning modes defines the order of the degrees of freedom in the product. The first column is the expansion coefficients.

As described above in Section 5, quantics has a number of built in operators. Other parameters are defined in the PARAMETER-SECTION, where units such as H-mass, Angstrom or cm-1 can be used. In Fig. 5, the kinetic energy operator in

Jacobi coordinates is coded. This can be written:

$$T = \left(\frac{1}{2\mu_R R^2} + \frac{1}{2\mu_r r^2} \right) j^2 - \frac{1}{2\mu_R} \frac{\partial^2}{\partial R^2} - \frac{1}{2\mu_r} \frac{\partial^2}{\partial r^2} \qquad (28)$$

where R is the scattering coordinate and r the diatomic bond length, with reduced masses μ_R and μ_r, respectively. j is the angular momentum operator for the rotation of the diatomic relative to the scattering coordinate. The potential surface for H + H_2 system is defined by the label V. This relates to the operator defined in the LABELS-SECTION as the LSTH surface in Jacobi coordinates.

In Fig. 6, the Hamiltonian is a four dimensional linear vibronic coupling model in mass-frequency scaled coordinates. The table

```
OP_DEFINE-SECTION
title
H+H2 Reactive Scattering in Jacobian Coordinates. J=0.
Minimum of H_2 curve: 4.74746 eV. Zero point energy of vib.: 0.270 eV
end-title
end-op_define-section


PARAMETER-SECTION
mass_rd = 0.6666666666667, H-mass      # Reduced mass of H--H2 system
mass_rv = 0.50, H-mass                 # Reduced mass of H2 molecule
jtot    = 0                            # Total angular momentum
jbf     = 0                            # Projection on BF axis (K, or Omega).
end-parameter-section


HAMILTONIAN-SECTION
--------------------------------------------
  modes        | rd    | rv    | theta
--------------------------------------------
0.5/mass_rd    | q^-2  | 1     | j^2
0.5/mass_rv    | 1     | q^-2  | j^2
1.0            | KE    | 1     | 1
1.0            | 1     | KE    | 1
1.0            | V
--------------------------------------------
end-hamiltonian-section


LABELS-SECTION
V = lsth {jacobian}
end-labels-section


# The following one-dimensional hamiltonian is used to determine the
# eigenstates of H_2. These are then used as initial rv-spf's.

HAMILTONIAN-SECTION_H2
--------------------------------------------
  modes        | rd    | rv    | theta
--------------------------------------------
1.0            | 1     | KE    | 1
1.0            | 1     | v:H2  | 1
--------------------------------------------
end-hamiltonian-section


end-operator
```

Fig. 5. Input file defining the operators in a calculation for the scattering of H+H_2.

codes the operator

$$\mathbf{H} = \sum_i \frac{\omega_i}{2} \left(-\frac{\partial}{\partial q_i^2} + q_i^2 \right) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} -\Delta & 0 \\ 0 & \Delta \end{pmatrix} \quad (29)$$

$$+ \sum_{i=6a,1a,9a} \begin{pmatrix} \kappa_i^{(1)} q_i & 0 \\ 0 & \kappa_i^{(2)} q_i \end{pmatrix} + \begin{pmatrix} 0 & \lambda q_{10a} \\ \lambda q_{10a} & 0 \end{pmatrix} \quad (30)$$

where ω_i is the frequency of an oscillator, 2Δ the energy gap between the states at the Franck–Condon point, and $\kappa_i^{(\alpha)}$ and λ the coupling parameters.

In addition to the Hamiltonian, other operators can be defined. For example, in the HAMILTONIAN-SECTION_H2 section in Fig. 5, a one-dimensional H_2 operator is set up and given a label H2. This is used in the INIT_WF-SECTION of the input file in Fig. 3

```
OP_DEFINE-SECTION
title Pyrazine 4-mode model, linear coupling.  end-title
end-op_define-section

PARAMETER-SECTION
w10a   = 0.09357, ev
w6a    = 0.0740 , ev
w1     = 0.1273 , ev
w9a    = 0.1568 , ev
delta  = 0.46165, ev
lambda = 0.1825 , ev
k6a1   =-0.0964 , ev
k6a2   = 0.1194 , ev
k11    = 0.0470 , ev
k12    = 0.2012 , ev
k9a1   = 0.1594 , ev
k9a2   = 0.0484 , ev
end-parameter-section

HAMILTONIAN-SECTION
--------------------------------------------------
    modes       | el  | v10a | v6a | v1  | v9a
--------------------------------------------------
1.0*w10a        | 1   | KE   | 1   | 1   | 1
0.5*w10a        | 1   | q^2  | 1   | 1   | 1
1.0*w6a         | 1   | 1    | KE  | 1   | 1
0.5*w6a         | 1   | 1    | q^2 | 1   | 1
1.0*w1          | 1   | 1    | 1   | KE  | 1
0.5*w1          | 1   | 1    | 1   | q^2 | 1
1.0*w9a         | 1   | 1    | 1   | 1   | KE
0.5*w9a         | 1   | 1    | 1   | 1   | q^2
-delta          | S1&1 | 1   | 1   | 1   | 1
delta           | S2&2 | 1   | 1   | 1   | 1
lambda          | S1&2 | q   | 1   | 1   | 1
k6a1            | S1&1 | 1   | q   | 1   | 1
k6a2            | S2&2 | 1   | q   | 1   | 1
k11             | S1&1 | 1   | 1   | q   | 1
k12             | S2&2 | 1   | 1   | q   | 1
k9a1            | S1&1 | 1   | 1   | 1   | q
k9a2            | S2&2 | 1   | 1   | 1   | q
--------------------------------------------------
end-hamiltonian-section

end-operator
```

Fig. 6. Input file defining the vibronic coupling Hamiltonian for a calculation of the absorption spectrum of pyrazine.

to define the initial SPFs for the rv degree of freedom using the eigenfunctions of this operator.

In Fig. 7 an alternative format input for a direct dynamics calculation is shown. While it is possible to use the sections described so far, it is more intuitive to input the information on the system based on the atoms and Cartesian coordinates used in the associated quantum chemistry calculation. In place of the PRIMITIVE-BASIS-SECTION, SPF-BASIS-SECTION and INIT_WF-SECTION, an INITIAL-GEOMETRY-SECTION is used. This defines the atoms and coordinates of the geometry around which the initial wavepacket is based. This calculation runs in normal mode coordinates, specified by the direct = nmodes

```
RUN-SECTION
name = h2o-4s-dd
propagation      direct = nmodes       ngwp = 7
title =  water direct dynamics
tfinal = 100.0   tout = 0.5  tpsi= 0.5
psi  gridpop   update  steps  auto   normstop = 1.0d-6
end-run-section

INITIAL-GEOMETRY-SECTION
nstates = 4     init_state = 2
cartesian = angst
 O  0.0000000000  0.0000000000  0.1121548697
 H  0.0000000000  0.7611532929 -0.4805424348
 H  0.0000000000 -0.7611532929 -0.4805424348
end-cartesian
nmode
1A1    -0.0819 1705.12,   cm-1 width = 0.7076
2A1 0.1985 3835.80,   cm-1 width = 0.7326
3B2 0.0000 3964.83,   cm-1 width = 0.7810
end-nmode
end-initial-geometry-section

DIRDYN-SECTION
data = h2o-4s-dd_dddata
transfile = water-opt-nosym.out
qcprogram = molpro     method = cas
ener0 = -76.143191
dd_diab=global     dbsave     hess_upd
db = rdwr     dbmin = 0.25 dbave=0.0
nbasis = 24
subcmd = /home/software/bin/run_molpro09 , 2
end-dirdyn-section


end-input
```

Fig. 7. Example input file for a direct dynamics calculation of the photoexcitation of water using normal mode coordinates.

keyword in the RUN-SECTION. As a result, the labels for the normal modes along with the initial coordinates and frequencies are given in the nmode to end-nmode block.

The information required for the direct dynamics is given in the DIRDYN-SECTION. The quantum chemistry program to be used (MOLPRO) along with the method (CASSCF) are specified. Other parameters control how often new points are stored in the database (DB), which is stored in the directory specified by the data keyword. The options controlling the MOLPRO calculation are controlled using a template file, shown in Fig. 8. This is put in the DB directory and used to create the MOLPRO input file.

7. Installation and testing

The package is installed using a script, install_quantics, found in the install directory. On a standard linux installation (e.g. Opensuse, Ubuntu or Debian) with a gfortran or intel compiler the package should install automatically accepting the defaults of the script. The installation configuration can be changed by altering the configuration files, as described in the documentation. While the code is complete, it is possible to link to external BLAS and LAPACK for better performance. It is also possible to compile different versions of the code (e.g. serial, OpenMP, or different compilers), which then obtain a type, e.g. quantics.omp or quantics.intel.

To use the package, the environment variable $QUANTICS_DIR needs to be set, which points to the root directory of the installation. By default, the installation script writes this to the .bashrc file. Once installed, the test suite can be run by typing elk_test_gen in a new directory. If all the tests run without any error messages, the package is ready. The documentation can be found by pointing a web browser to $QUANTICS_DIR/doc/index.html. This gives access to a set of pages with descriptions of the programs and options. The installation procedure also compiles the user manual, which can also be accessed from this index page. This contains a tutorial, as well as detailed descriptions of how to use the code.

```
***water
nosym
geomtyp=xyz
geometry={
3
water geometry
<geometry>
}
basis=6-31G**

$Orbread:{matrop;read,oldorbitals,file=<orbfile>;save,oldorbitals,2140.2,orbitals}$

! State 1
{multi,gradient=1.0d-6,energy=1.0d-8;occ,9;
start,2140.2;
closed,1;
$Force:CPMCSCF,grad,1.1,accu=1.0d-8,record=5101.1;$
wf,10,1,0;
state,4;
weight,0.25,0.25,0.25,0.25;
orbital,2141.2;
orbprint,100;
}
$Force:{Force;SAMC,5101.1}$
$Freq:{Frequencies,analytical,noproject,new;print,hessian}$

! State 2
$Force:{multi,gradient=1.0d-6,energy=1.0d-8;occ,9;start,2141.2;closed,1;
CPMCSCF,grad,2.1,accu=1.0d-8,record=5102.1;wf,10,1,0;state,4;
weight,0.25,0.25,0.25,0.25;}$
$Force:{Force;SAMC,5102.1}$
$Freq:{Frequencies,analytical,noproject,new;print,hessian}$

! 2-1
$NACT:{multi,gradient=1.0d-6,energy=1.0d-8;occ,9;closed,1;start,2141.2;
cpmcscf,nacm,2.1,1.1,accu=1.0d-10,record=5105.1;wf,10,1,0;state,4;
weight,0.25,0.25,0.25,0.25;}$
$NACT:{Force;SAMC,5105.1}$
```

Fig. 8. Part of a direct dynamics template file for a 4-state calculation on water using MOLPRO to obtain the potential surfaces at the CASSCF level of theory. The directives for only 2 states are shown.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

References

[1] G.A. Worth, M.H. Beck, A. Jäckle, H.-D. Meyer, The heidelberg MCTDH package, Version 8.2 H.-D. Meyer, version 8.3 (2002), version 8.4 (2007) current version: 8.4.17, 2019, see http://mctdh.uni-hd.de.
[2] H.-D. Meyer, U. Manthe, L.S. Cederbaum, Chem. Phys. Lett. 165 (1990) 73–78.
[3] U. Manthe, H.-D. Meyer, L.S. Cederbaum, J. Chem. Phys 97 (1992) 3199–3213.
[4] A. Jäckle, H.-D. Meyer, J. Chem. Phys 109 (1998) 2614–2623.

[5] A. Jäckle, H.-D. Meyer, J. Chem. Phys. 105 (1996) 6778–6786.
[6] M.H. Beck, H.-D. Meyer, J. Chem. Phys. 114 (2001) 2036–2046.
[7] F. Gatti, M.H. Beck, G.A. Worth, H.-D. Meyer, PCCP 3 (2001) 1576–1582.
[8] A. Raab, I. Burghardt, H.-D. Meyer, J. Chem. Phys. 111 (1999) 8759–8772.
[9] A. Raab, H.-D. Meyer, Theor. Chem. Acc. 104 (2000) 358–369.
[10] A. Jäckle, H.-D. Meyer, J. Chem. Phys 104 (1996) 7974–7984.
[11] A. Jäckle, H.-D. Meyer, J. Chem. Phys 102 (1995) 5605–5615.
[12] M. Ehara, H.-D. Meyer, L.S. Cederbaum, J. Chem. Phys 105 (1996) 8865–8877.
[13] M.-C. Heitz, H.-D. Meyer, J. Chem. Phys 114 (2001) 1382–1392.
[14] S. Sukiasyan, H.-D. Meyer, J. Phys. Chem. A 105 (2001) 2604–2611.
[15] G.A. Worth, H.-D. Meyer, L.S. Cederbaum, J. Chem. Phys. 105 (1996) 4412–4426.
[16] A. Raab, G.A. Worth, H.-D. Meyer, L.S. Cederbaum, J. Chem. Phys. 110 (1999) 936–946.
[17] C. Cattarius, G.A. Worth, H.-D. Meyer, L.S. Cederbaum, J. Chem. Phys. 115 (2001) 2088–2100.
[18] C. Cattarius, A. Markmann, G.A. Worth, The VCHAM program, 2007, see http://www.pci.uni-heidelberg.de/tc/usr/mctdh/.
[19] I. Burghardt, H.-D. Meyer, L.S. Cederbaum, J. Chem. Phys. 111 (1999) 2927–2938.
[20] G.A. Worth, M.A. Robb, I. Burghardt, Faraday Discuss. 127 (2004) 307–323.
[21] B. Lasorne, M.J. Bearpark, M.A. Robb, G.A. Worth, Chem. Phys. Lett. 432 (2006) 604–609.
[22] B. Lasorne, M.A. Robb, G.A. Worth, PCCP 9 (2007) 3210–3227.
[23] O. Vendrell, H.-D. Meyer, J. Chem. Phys. 134 (2011) 44135.
[24] M. Brill, O. Vendrell, H.-D. Meyer, in: P. Piecuch, J. Maruani, G. Delgado-Barrio, S. Wilson (Eds.), Adv. Theo. At. Mol. Sys, in: Progress in Theoretical Chemistry and Physics, vol. 20, Springer, Netherlands, Dordrecht, 2009, pp. 67–80.
[25] O. Vendrell, F. Gatti, D. Lauvergnat, H.-D. Meyer, J. Chem. Phys. 127 (2007) 184302.
[26] G.W. Richings, G.A. Worth, J. Phys. Chem. A 119 (2015) 12457–12470.
[27] G.W. Richings, I. Polyak, K.E. Spinlove, G.A. Worth, I. Burghardt, B. Lasorne, Int. Rev. Phys. Chem. 34 (2015) 269–308.
[28] T.J. Frankcombe, M.A. Collins, G.A. and Worth, Chem. Phys. Lett. 489 (2010) 242–247.
[29] M. Mališ, J. Novak, G. Zgrablić, F. Parmigiani, N. Došlić, Phys. Chem. Chem. Phys. 19 (2017) 25970–25978.
[30] G.A. Worth, R.E. Carley, H.H. Fielding, Chem. Phys. 338 (2007) 220–227.
[31] M. Schröder, J.-L. Carreon-Macedo, A. Brown, PCCP 10 (2008) 850–856.
[32] T.J. Penfold, G.A. Worth, C. Meier, PCCP 12 (2010) 15616–15627.
[33] H.-D. Meyer, F. Gatti, G.A. Worth, Multidimensional Quantum Dynamics: MCTDH Theory and Applications, Wiley-VCH, Weinheim, Germany, 2009.
[34] M.H. Beck, A. Jäckle, G.A. Worth, H.-D. Meyer, Phys. Rep. 324 (2000) 1–105.
[35] H.-D. Meyer, G.A. Worth, Theo. Chem. Acc. 109 (2003) 251–267.
[36] R. Kosloff, J. Phys. Chem. 92 (1988) 2087–2100.
[37] C. Cerjan (Ed.), Numerical Grid Methods and their Application to Schrödinger's Equation, Kluwer Academic Publishers, Dordrecht, 1993.
[38] J.C. Light, I.P. Hamilton, J.V. Lill, J. Chem. Phys. 82 (1985) 1400–1409.
[39] T. Carrington, J. Chem. Phys. 146 (2017) 120902–120911.
[40] H. Wang, M. Thoss, J. Chem. Phys. 119 (2003) 1289–1299.
[41] U. Manthe, J. Chem. Phys. 128 (2008) 164116.
[42] H. Wang, J. Phys. Chem. A 119 (2015) 7951–7965.
[43] I.R. Craig, M. Thoss, H. Wang, J. Chem. Phys. 127 (2007) 144503.
[44] R. Borrelli, M. Thoss, H. Wang, W. Domcke, Mol. Phys. 110 (2012) 751–763.
[45] I. Burghardt, M. Nest, G.A. Worth, J. Chem. Phys. 119 (2003) 5364–5378.
[46] I. Burghardt, K. Giri, G.A. Worth, J. Chem. Phys. 129 (2008) 174104–174114.
[47] G.A. Worth, I. Burghardt, Chem. Phys. Lett. 368 (2003) 502–508.
[48] M. Ben-Nun, T.J. Martínez, Adv. Chem. Phys. 121 (2002) 439–512.
[49] D.V. Shalashilin, M.S. Child, J. Chem. Phys. 115 (2001) 5367–5375.
[50] K.E. Spinlove, M. Vacher, M. Bearpark, M.A. Robb, G.A. Worth, Chem. Phys. 482 (2017) 52–63.
[51] G.A. Worth, M.A. Robb, Adv. Chem. Phys. 124 (2002) 355–432.
[52] K.E. Spinlove, G.W. Richings, M.A. Robb, G.A. Worth, Farad. Discuss. 212 (2018) 191–215.
[53] M.J. Frisch, G.W. Trucks, H.B. Schlegel, G.E. Scuseria, M.A. Robb, J.R. Cheeseman, G. Scalmani, V. Barone, G.A. Petersson, H. Nakatsuji, X. Li, M. Caricato, A. Marenich, J. Bloino, B.G. Janesko, R. Gomperts, B. Mennucci, H.P. Hratchian, J.V. Ortiz, A.F. Izmaylov, J.L. Sonnenberg, D. Williams-Young, F. Ding, F. Lipparini, J. Goings, B. Peng, A. Petrone, T. Henderson, D. Ranasinghe, V.G. Zakrzewski, J. Gao, N. Rega, G. Zheng, W. Liang, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, K. Throssell, J.A. Montgomery Jr, J.E. Peralta, F. Ogliaro, M. Bearpark, J.J. Heyd, E. Brothers, K.N. Kudin, V.N. Staroverov, T. Keith, R. Kobayashi, J. Normand, K. Raghavachari, A. Rendell, J.C. Burant, S.S. Iyengar, J. Tomasi, M. Cossi, J.M. Millam, M. Klene, C. Adamo, R. Cammi, J.W. Ochterski, R.L. Martin, K. Morokuma, O. Farkas, J.B. Foresman, D.J. Fox, Gaussian 09, revision a.02, 2009, www.gaussian.com.
[54] H.-J. Werner, P.J. Knowles, G. Knizia, F.R. Manby, M. Schütz, P. Celani, W. Györffy, D. Kats, T. Korona, R. Lindh, A. Mitrushenkov, G. Rauhut, K.R. Shamasundar, T.B. Adler, R.D. Amos, A. Bernhardsson, A. Berning, D.L. Cooper, M.J.O. Deegan, A.J. Dobbyn, F. Eckert, E. Goll, C. Hampel, A. Hesselmann, G. Hetzer, T. Hrenar, G. Jansen, C. Köppl, Y. Liu, A.W. Lloyd, R.A. Mata, A.J. May, S.J. McNicholas, W. Meyer, M.E. Mura, A. Nicklass, D.P. O'Neill, P. Palmieri, D. Peng, K. Pflüger, R. Pitzer, M. Reiher, T. Shiozaki, H. Stoll, A.J. Stone, R. Tarroni, T. Thorsteinsson, M. Wang, Molpro, version 2015.1, a package of ab initio programs, 2015, http://www.molpro.net.
[55] Y. Shao, Z. Gan, E. Epifanovsky, A.T.B. Gilbert, M. Wormit, J. Kussmann, A.W. Lange, A. Behn, J. Deng, X. Feng, D. Ghosh, M. Goldey, P.R. Horn, L.D. Jacobson, I. Kaliman, R.Z. Khaliullin, T. Kus, A. Landau, J. Liu, E.I. Proynov, Y.M. Rhee, R.M. Richard, M.A. Rohrdanz, R.P. Steele, E.J. Sundstrom, H.L. Woodcock I.I.I., P.M. Zimmerman, D. Zuev, B. Albrecht, E. Alguire, B. Austin, G.J.O. Beran, Y.A. Bernard, E. Berquist, K. Brandhorst, K.B. Bravaya, S.T. Brown, D. Casanova, C.-M. Chang, Y. Chen, S. Chien, K.D. Closser, D.L. Crittenden, M. Diedenhofen, DiStasio Jr. R.A., H. Do, A.D. Dutoi, R.G. Edgar, S. Fatehi, L. Fusti-Molnar, A. Ghysels, A. Golubeva-Zadorozhnaya, J. Gomes, M.W.D. Hanson-Heine, P.H.P. Harbach, A.W. Hauser, E.G. Hohenstein, Z.C. Holden, T.-C. Jagau, H. Ji, B. Kaduk, K. Khistyaev, J. Kim, R.A. King, P. Klunzinger, D. Kosenkov, T. Kowalczyk, C.M. Krauter, K.U. Lao, A.D. Laurent, K.V. Lawler, S.V. Levchenko, C.Y. Lin, F. Liu, E. Livshits, R.C. Lochan, A. Luenser, P. Manohar, S.F. Manzer, S.-P. Mao, N. Mardirossian, A.V. Marenich, S.A. Maurer, N.J. Mayhall, E. Neuscammen, C.M. Oana, R. Olivares-Amaya, D.P. O'Neill, J.A. Parkhill, T.M. Perrine, R. Peverati, A. Prociuk, D.R. Rehn, E. Rosta, N.J. Russ, S.M. Sharada, S. Sharma, D.W. Small, A. Sodt, T. Stein, D. Stück, Y.-C. Su, A.J.W. Thom, T. Tsuchimochi, V. Vanovschi, L. Vogt, O. Vydrov, T. Wang, M.A. Watson, J. Wenzel, A. White, C.F. Williams, J. Yang, S. Yeganeh, S.R. Yost, Z.-Q. You, I.Y. Zhang, X. Zhang, Y. Zhao, B.R. Brooks, G.K.L. Chan, D.M. Chipman, C.J. Cramer, W.A. Goddard III, M.S. Gordon, W.J. Hehre, A. Klamt, H.F. Schaefer III, M.W. Schmidt, C.D. Sherrill, D.G. Truhlar, A. Warshel, X. Xu, A. Aspuru-Guzik, R. Baer, A.T. Bell, N.A. Besley, J.-D. Chai, A. Dreuw, B.D. Dunietz, T.R. Furlani, S.R. Gwaltney, C.-P. Hsu, Y. Jung, J. Kong, D.S. Lambrecht, W. Liang, C. Ochsenfeld, V.A. Rassolov, L.V. Slipchenko, J.E. Subotnik, T. van Voorhis, J.M. Herbert, A.I. Krylov, P.M.W. Gill, M. Head-Gordon, Mol. Phys. 113 (2015) 184–215.
[56] K. Andersson, M.R.A. Blomberg, M.P. Fülscher, G. Karlström, R. Lindh, P.-Å. Malmqvist, P. Neogrády, J. Olsen, B.O. Roos, A.J. Sadlej, M. Schütz, L. Seijo, L. Serrano-Andrés, P.E.M. Siegbahn, P.-O. Widmark, Molcas, version 4, 1997.
[57] A. Raab, H.-D. Meyer, J. Chem. Phys. 112 (2000) 10718–10729.
[58] H.-D. Meyer, F. Le Quéré, C. Léonard, F. Gatti, Chem. Phys. 329 (2006) 179–192.
[59] M.H. Beck, H.-D. Meyer, J. Chem. Phys. 109 (1998) 3730–3741.