Computer Physics Communications 303 (2024) 109279

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

Computer Programs in Physics

The hybrid anti-symmetrized coupled channels method (haCC) for the tRecX code ☆,☆☆

Hareesh Chundayil [a], Vinay P. Majety [a], Armin Scrinzi [b,*]

[a] Department of Physics and CAMOST, Indian Institute of Technology Tirupati, Yerpedu, India
[b] Ludwig Maximilian University, Theresienstrasse 37, 80333 Munich, Germany

ARTICLE INFO

Keywords:
Schrödinger solver
Strong field physics
Attosecond physics
Recursive structure

ABSTRACT

We present a new implementation of the hybrid antisymmetrized Coupled Channels (haCC) method in the framework of the tRecX (Scrinzi, 2022 [6]). The method represents atomic and molecular multi-electron functions by combining CI functions, Gaussian molecular orbitals, and a numerical single-electron basis. It is suitable for describing high harmonic generation and the strong-field dynamics of ionization. Fully differential photoemission spectra are computed by the tSurff method. The theoretical background of haCC is outlined and key improvements compared to its original formulation are highlighted. We discuss control of over-completeness resulting from the joint use of the numerical basis and Gaussian molecular orbitals by pseudo-inverses based on the Woodbury formula. Further new features of this tRecX release are the iSurff method, new input features, and the AMOS gateway interface. The mapping of haCC into the tRecX framework for solving the time-dependent Schrödinger equation is shown. Use, performance, and accuracy of haCC are discussed on the examples of high-harmonic generation and strong-field photo-emission by short laser pulses impinging on the Helium atom and on the linear molecules N_2 and CO.

Program summary

Program title: tRecX — time-dependent Recursive indeXing (tRecX = tSurff+irECS)
CPC Library link to program files: https://doi.org/10.17632/m9g2jc82sw.1
Developer's repository link: https://gitlab.physik.uni-muenchen.de/AG-Scrinzi/tRecX
Licensing provisions: GNU General Public License 2
Programming language: C++
External libraries: Eigen, arpack, lapack, blas, boost, FFTW (optional)
Journal Reference of previous version: A. Scrinzi, Comp. Phys. Comm., 270:108146, 2022.
Does the new version supersede the previous version: Yes
Reasons for the new version: Major new functionality: haCC — hybrid antisymmetrized coupled channels method
Summary of revisions: Main additions are haCC and iSurff. Code usage and compilation were improved.
Nature of problem: tRecX is a general solver for time-dependent Schrödinger-like problems, with applications mostly in strong field and attosecond physics. There are no technical restrictions on the spatial dimension of the problem with up to 6 spatial dimensions realized in the strong-field double ionization of Helium. Gaussian-based quantum chemical multi-electron atomic and molecular structure can be combined with the numerical basis. A selection of coordinate systems is available and any Hamiltonian involving up to second derivatives and arbitrary up to three dimensional potentials can be defined on input by simple scripts.
Solution method: The method of lines is used with spatial discretization by a flexible combination of one dimensional basis sets, DVR representations, discrete vectors, expansions into higher-dimensional eigenfunctions of user-defined operators, and Gaussian based molecular orbitals. Multi-electron Gaussian-based CI (configuration interaction) functions for neutrals and ions are combined with the numerical basis. Photo-emission spectra are calculated using the time-dependent surface flux method (tSurff) in combination with infinite range exterior

☆ The review of this paper was arranged by Prof. W. Jong.
☆☆ This paper and its associated computer program are available via the Computer Physics Communications homepage on ScienceDirect (http://www.sciencedirect.com/science/journal/00104655).

* Corresponding author.
E-mail addresses: vinay.majety@iittp.ac.in (V.P. Majety), Armin.Scrinzi@lmu.de (A. Scrinzi).

https://doi.org/10.1016/j.cpc.2024.109279
Received 19 March 2024; Received in revised form 1 May 2024; Accepted 6 June 2024
Available online 12 June 2024

complex scaling (irECS) for absorption. The code is object oriented and makes extensive use of tree-structures and recursive algorithms. Parallelization is by MPI. Code design and performance allow use in production as well as for graduate level training.

Contents

1. Introduction

The “hybrid anti-symmetrized coupled channels” (haCC) method unites quantum chemical multi-electron structure with a purely numerical description of non-perturbative strong field interactions. In its first formulation it was presented in [1] and it was used for applications that showed the essential role of anti-symmetrization and correlation for strong-field ionization rates [2], for fully differential photoemission from linear molecules [3], for benchmark ionization rates of atoms and small diatomics [4], and for demonstrating the absence of correlation effects in the attosecond delays in the photo-emission by elliptically polarized light [5]. The present major upgrade of public domain tRecX code [6] includes an improved version of haCC with all capabilities cited above.

The tRecX package is designed to be a high-performance, yet flexible and robust code with good maintainability and usability for Schrödinger-like time-dependent problems, where the original design was for applications in strong-field and attosecond physics. Its main use has been for computing the interaction of atomic and molecular systems in non-perturbatively strong laser fields. It implements a range of techniques such as irECS (infinite-range exterior complex scaling [7]), tSurff (the time-dependent surface flux method [8,9]), general and mixed gauges [10], and the FE-DVR method for complex scaling [11,12]. The most demanding applications of tRecX have been the computation of fully-differential double electron emission spectra of the Helium atom [13,14] at laser wave length from 10 to 800 nm. These capabilities were included with the first official release of tRecX. The applications for multi-electron systems [2–5] with arbitrary alignment between the direction of laser polarization and the molecular axis have not been part of the public code so far.

The new release keeps with the original tRecX design. A conscious effort of adhering to good programming practice is being made for ensuring re-usability and maintainability. The C++ code uses abstract and template classes for uniform and transparent code structure. The classes reflect concepts that are familiar in physics such as the linear and more specifically Hilbert space, operators that are usually but not necessarily linear maps, and wave functions. Discretization of the wave function is in terms of an abstract basis set class, whose specific implementation covers the whole range from discrete sets of vectors, over grids, finite-elements, standard basis sets such as spherical harmonics, all the way to expansions in terms of eigenfunctions of a user-defined operator. With the present release, Gaussian-based molecular orbitals and multi-electron CI functions were added to that list. These are combined in a tree-structured hierarchy that admits building correlated (non-product) bases from one- and multi-dimensional factors. For performance, numerical libraries such as Lapack [15], Eigen [16], and FFTW [17] are used on the low level. Parallelization is through MPI with automatic load-balancing according to self-measurement of the code. Also, non-trivial model Hamiltonians can be implemented quickly with little compromise in computational performance. Details of the tRecX structure and algorithms can be found in Ref. [6].

The present paper introduces the new formulation of haCC, explains key elements of its efficient implementation and how it fits into the wider tRecX framework. We show how haCC is invoked from tRecX, and how one can assess convergence and accuracy of a haCC calculation. Further, the new addition of the iSurff extension [18] to the tSurff

method is explained and demonstrated by examples. Various improvements in usability are highlighted as appropriate.

2. The haCC method

The purpose of haCC is to solve the Schrödinger equation for a multi-centered multi-electron system — a molecule — in presence of a dipole electric field with general polarization of the field vector $\vec{\mathcal{E}}(t)$. The Schrödinger equation for the problem has the form, in length gauge,

$$i\frac{d}{dt}\Psi(t) = \left[H_0 + \vec{\mathcal{E}}(t)\cdot\sum_{n=1}^{N}\vec{r}_n\right]\Psi(t), \tag{1}$$

where H_0 is the Hamiltonian of the field-free N-electron system. We use atomic units (au), where electron mass, Planck constant, and elementary charge are all $=1$: $m_e = \hbar = e^2 = 1$.

When $\vec{\mathcal{E}}(t)$ becomes large, at low laser frequencies ω, or on very short time scales there is typically a broad spectrum of continuum states involved in the dynamics that defies common expansion into standard basis functions. For systems with up to two electrons, expansion on grids or localized basis functions such as B-splines, finite elements (FE), or FE-DVR is feasible. For systems with more electrons and when the majority of electrons remains in near bound states, one needs to supplement the description by methods that are better suitable for near-bound electrons. Examples for this type of approach are the time-dependent R-matrix [19], B-spline R-matrix [20], and the XChem [21] codes.

2.1. The haCC expansion

The central assumption of haCC is that the dynamics is dominated by the motion of a single electron (called the "free electron" in the following) and that the correlation of this motion with the remaining electrons can be described by coupling between the first few excited ionic channels. In addition to that, the fully correlated field-free neutral ground state and possibly neutral excited states are included. This leads to an ansatz for the wave function in the form

$$|\Psi\rangle = \sum_{\mathcal{N}}|\mathcal{N}\rangle c_{\mathcal{N}} + \sum_{A}\left\{\sum_{i}a_i^\dagger|A\rangle c_i^A + \sum_{\alpha}a_\alpha^\dagger|A\rangle c_\alpha^A\right\}. \tag{2}$$

The field-free neutral $|\mathcal{N}\rangle$ states and the ionic $|A\rangle$ states are CI (configuration interaction) functions. Both sets of states are constructed from the same I Gaussian-based molecular orbitals $|i\rangle, i = 0,\dots I-1$. The free electron is described by an expansion into the molecular orbitals $|i\rangle$ and numerical basis functions $|\alpha_\perp\rangle$ with the creation operators $a_i^\dagger$ and $a_\alpha^\dagger$, respectively. The $|\alpha_\perp\rangle$ are explicitly orthogonalized to the $|i\rangle$ as

$$|\alpha_\perp\rangle = \left(1 - \sum_{i=0}^{I-1}|i\rangle\langle i|\right)|\alpha\rangle, \qquad |\alpha\rangle = |Y_{m_\alpha l_\alpha}\rangle|\xi_{n_\alpha}^{i_\alpha}\rangle. \tag{3}$$

The $\langle\phi,\theta|Y_{ml}\rangle = Y_{ml}(\phi,\theta)$ are standard spherical harmonics. For the radial discretization we use a FE-DVR method [11,12], where the $\langle r|\xi_n^i\rangle = \xi_n^i(r)$ are Lagrange polynomials at the Lobatto quadrature points r_i for the radial interval $r\in[r_n, r_{n+1}]$. On the last, semi-infinite interval $|r_N,\infty)$ we use Lagrange-polynomials times an exponential factor $\exp(-\kappa r)$ [7] and a Radau quadrature that include r_N as one of the quadrature points. The quadratures are used for the numerical calculation of integrals. By expressing multi-electron functions through the creation operators we ensure anti-symmetrization of the free-electron basis with the ionic states $|A\rangle$. Neutral and ionic functions can be, in principle, drawn from any Gaussian-based quantum chemistry code. As discussed below, the computation of matrix elements requires rather detailed access to reaction density matrices and generalized Dyson orbitals, which at present we obtain through a customized version of the COLUMBUS code [22].

An important feature of the α-basis is that the interval boundaries r_n can be freely chosen and the size of the angular expansion and degree of the Lagrange polynomials can be adjusted to the local properties of the wave function in a specific system. In particular, at the location of the atoms, both, bound and scattering parts of the solution will have cusp-like peaks. While the bound state part of the peaks is largely captured by the Gaussians of the molecular orbitals $|i\rangle$, peaks in the scattering part must be representable by the numerical functions $|\alpha\rangle$. By using small shells $[r_n, r_{n+1}]$ around the radial position of the atoms and a large angular expansion on those intervals, one can represent the peaks well without inflating the basis at larger distances from the atoms.

A very similar ansatz is used in Refs. [21,23]. The important distinction of the ansatz (2) from that expansion is our use of the dense numerical α-basis in the complete space including the domain of the molecular orbitals $|i\rangle$. This choice was made for being able to describe non-perturbative strong-field interactions where the so-called "recollision" processes play a key role. Recollsion determines high harmonic generation and photo-electron emission by long wavelength laser fields: the field drives electrons into large spatial `excursions of 10 au and more`of several 10 s of au, after which they recollide with their parent ion at a broad range of energies. The α-basis provides the necessary flexibility for describing this dynamics. It comes at the cost of computing two-electron integrals involving both, Gaussian-based molecular orbitals $|i\rangle$ and the numerical functions $|\alpha\rangle$, see Sec. 2.5.

2.2. Gauge

The expansion functions of haCC are chosen with the physical model assumption that ground state and first few excited states of neutral and ion play an essential role also when the external field $\vec{\mathcal{E}}(t)$ is present. Upon changing from length to velocity gauge all states, including the states considered essential, become time-dependent by the transformation factor $\exp[-i\vec{A}(t)\cdot\vec{r}]$ with the vector potential $\vec{A} = \int_{-\infty}^{t} d\tau\vec{\mathcal{E}}(\tau)$. Our assumption singles out length gauge as the gauge where the essential states are time-independent. This difference is particularly important in strong fields or at low laser frequencies ω, where the phases $\vec{A}\cdot\vec{r} \approx \vec{\mathcal{E}}\cdot\vec{r}/\omega$ vary significantly across the extension of the essential states.

For avoiding the costly re-computation of matrix elements at each time t, we use length gauge up to the radius where molecular orbitals become negligible. For reasons of numerical efficiency, we switch to a mixed gauge beyond that radius that asymptotically coincides with the velocity gauge. That gauge is constructed such that field-dependence of the matrix has the form of multiplications by $\mathcal{E}_i(t)$, $\vec{A}_i(t)$, and $A_i(t)A_j(t)$ with $i,j\in\{x,y,z\}$. This obviates the need for re-computing matrix elements at each step at the expense of additional quadrupole-like terms. With z-polarization, the interaction operator has three parts that are multiplied by $\mathcal{E}_z(t)$, $A_z(t)$, and $A_z^2(t)$ respectively, in general polarization it consists of 13 terms. Details of the approach and demonstration of its essential role for computing strong field effects can be found in [10].

2.3. Representation of operators

Our discretization space can be written as the direct sum $\mathcal{H} = \mathcal{N}\oplus\mathcal{I}\oplus\mathcal{A}$ of the neutral CI space $\mathcal{N}$ and the spaces where the ionic functions $|A\rangle$ are augmented by molecular orbitals $\mathcal{I} = \mathrm{span}(a_i^\dagger|A\rangle)$ and by the orthogonalized numerical α-functions $\mathcal{A} = \mathrm{span}(a_\alpha^\dagger|A\rangle)$. Any operator matrix $\widehat{M}$, including the overlap matrix, can be written in block-matrix form

$$\widehat{M} = \begin{pmatrix} \widehat{M}_{\mathcal{N}\mathcal{N}} & \widehat{M}_{\mathcal{N}\mathcal{I}} & \widehat{M}_{\mathcal{N}\mathcal{A}} \\ \widehat{M}_{\mathcal{I}\mathcal{N}} & \widehat{M}_{\mathcal{I}\mathcal{I}} & \widehat{M}_{\mathcal{I}\mathcal{A}} \\ \widehat{M}_{\mathcal{A}\mathcal{N}} & \widehat{M}_{\mathcal{A}\mathcal{I}} & \widehat{M}_{\mathcal{A}\mathcal{A}} \end{pmatrix}. \tag{4}$$

As written, each block is a full matrix except for some sparsity in the overlap matrix and possible zeros due to global symmetries like angular momentum. The exchange term of the Coulomb interaction will always create a full matrix. However, because of the orthogonalization of the

$|\alpha_\perp\rangle$ to the globally defined orbitals $|i\rangle$ also the blocks $\widehat{M}_{\mathcal{AA}}$ of any single particle operator and even the overlap block $\widehat{O}_{\mathcal{AA}}$ are full. Given the large dimension of, say, $\dim(\mathcal{A}) \sim 10^5$ and our need to separately apply multiple time-dependent operators of the mixed gauge dipole interaction this is a severe computational disadvantage. The fill in of the matrices originates from comparatively few molecular orbitals $\lesssim 50$, and it can therefore replaced by applying sparse matrices in combination with operators of rank $\lesssim 50$, which is computationally efficient.

One finds that matrix elements $M_{\alpha j}$ w.r.t. the orthonormal basis $|\alpha_\perp\rangle$ can be obtained from matrix elements $m^{(1)}_{\alpha l}$ for the local basis functions $|\alpha\rangle$ without the orthonormalization by

$$M_{\alpha j} = m^{(1)}_{\alpha j} + \sum_k U_{\alpha k} m^{(1)}_{kj} \qquad \text{with} \quad U_{\alpha k} := -\langle\alpha|k\rangle \tag{5}$$

and similarly

$$M_{\alpha\beta} = m^{(0)}_{\alpha\beta} + \sum_k U_{\alpha k} m^{(2)}_{k\beta} + \sum_k m^{(2)}_{\alpha k} U_{k\beta} + \sum_{kl} U_{\alpha k} m^{(2)}_{kl} U_{l\beta}. \tag{6}$$

The form of the $\widehat{m}^{(x)}$ will be discussed below.

In block-matrix form the part of the operator pertaining to the ionic channels $\mathcal{I} \oplus \mathcal{A}$ reads

$$\begin{pmatrix} \widehat{M}_{\mathcal{II}} & \widehat{M}_{\mathcal{IA}} \\ \widehat{M}_{\mathcal{AI}} & \widehat{M}_{\mathcal{AA}} \end{pmatrix} = \widehat{U}\widehat{m}\widehat{U}^\dagger \tag{7}$$

with

$$\widehat{U} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & \widehat{U}_{\mathcal{AI}} \end{pmatrix} \qquad \text{and} \qquad \widehat{m} = \begin{pmatrix} \widehat{m}^{(0)}_{\mathcal{II}} & \widehat{m}^{(1)}_{\mathcal{IA}} & \widehat{m}^{(1)}_{\mathcal{II}} \\ \widehat{m}^{(1)}_{\mathcal{AI}} & \widehat{m}^{(0)}_{\mathcal{AA}} & \widehat{m}^{(2)}_{\mathcal{IA}} \\ \widehat{m}^{(1)}_{\mathcal{II}} & \widehat{m}^{(2)}_{\mathcal{AI}} & \widehat{m}^{(2)}_{\mathcal{II}} \end{pmatrix} \tag{8}$$

The transformations $\widehat{U}$ involve only matrices where one dimension is equal to the number of orbitals $\dim(\mathcal{I}) = I \lesssim 50$ and $\widehat{m}^{(0)}_{\mathcal{AA}}$ is sparse except for the exchange term. The `class OperatorHaCC` that implements this type of operators is discussed below.

2.4. Inverse overlap and over-completeness

Denoting the expansion coefficients of Eq. (2) by the vector $\vec{c} = (c_{\mathcal{N}}, c^A_i, c^A_\alpha)$ and the time-dependent Hamiltonian matrix by $\widehat{H}(t)$, the time-evolution of the $\vec{c}$ is given by

$$\frac{d}{dt}\vec{c}(t) = -i\widehat{O}^{-1}\widehat{H}(t)\vec{c}(t). \tag{9}$$

In haCC, the overlap matrix $\widehat{O}$ has the structure

$$\widehat{O} = \begin{pmatrix} \mathbf{1}_{\mathcal{N}} & \widehat{O}_{\mathcal{NI}} & 0 \\ \widehat{O}^\dagger_{\mathcal{NI}} & \mathbf{1}_{\mathcal{I}} & 0 \\ 0 & 0 & \widehat{O}_{\mathcal{AA}} \end{pmatrix} \tag{10}$$

with the identity matrices $\mathbf{1}_{\mathcal{N}}$ and $\mathbf{1}_{\mathcal{I}}$ on the respective subspaces. It is easy to see that

$$\widehat{O}_{\alpha A,\beta B} = \langle A|a_\alpha a^\dagger_\beta|B\rangle = \delta_{AB}\langle\alpha_\perp|\beta_\perp\rangle = \delta_{AB}\langle\alpha|\beta\rangle - \sum_k U_{\alpha k}\overline{U}_{k\beta} \tag{11}$$

or in block matrix notation

$$\widehat{O}_{\mathcal{AA}} = \widehat{o}_{\mathcal{AA}} - \widehat{U}_{\mathcal{AI}}\widehat{U}^\dagger_{\mathcal{AI}}. \tag{12}$$

The two overlap blocks on $\mathcal{N} \oplus \mathcal{I}$ and $\mathcal{A}$ are independent. The structure of each block allows for the use of the Woodbury formula [24] by which one writes the inverse of a matrix of the form

$$\widehat{O} = \widehat{o} - \widehat{U}\widehat{U}^\dagger \tag{13}$$

as

$$\widehat{O}^{-1} = \widehat{o}^{-1} - \widehat{o}^{-1}\widehat{U}\widehat{C}^{-1}\widehat{U}^\dagger\widehat{o}^{-1} \tag{14}$$

with

$$\widehat{C} = \widehat{U}^\dagger\widehat{o}^{-1}\widehat{U} - \mathbf{1} \tag{15}$$

In our case, the $\widehat{o}$ for the $\mathcal{N} \oplus \mathcal{I}$ and $\mathcal{A}$ blocks are identity and diagonal, respectively, and the dimensions of $\widehat{C}$ are small. Application of the rhs. of Eq. (14) is thus computationally efficient.

The overlap blocks for both, $\mathcal{N} \oplus \mathcal{I}$ and $\mathcal{A}$, can become ill-conditioned, when the bases are sufficiently large. The ill-conditioning of $\widehat{O}$ re-appears as ill-conditioning of $\widehat{C}$. As $\widehat{C}$ is small, near-zero eigenvalues can be removed by diagonalizing and forming a pseudo-inverse

$$\widehat{C}^{-1}_\epsilon = \widehat{V}_\epsilon\Sigma^{-1}_\epsilon\widehat{V}^\dagger_\epsilon, \tag{16}$$

where eigenvalues $< \epsilon$ and the corresponding eigenvectors were removed from Σ_ϵ and $\widehat{V}_\epsilon$. The pseudo-inverse of the full matrix $\widehat{O}$

$$\widehat{O}^{-1}_\epsilon = \widehat{o}^{-1} - \widehat{o}^{-1}\widehat{U}\widehat{C}^{-1}_\epsilon\widehat{U}^\dagger\widehat{o}^{-1} \tag{17}$$

now acts only on the non-singular subspace and the operator

$$\widehat{P}_\epsilon := \widehat{O}\widehat{O}^{-1}_\epsilon \tag{18}$$

is a projector onto the subspace where $\widehat{O}$ is non-singular.

2.5. Matrix elements

Matrix elements between the various parts of the basis are best derived in second quantized notation writing single- and two-particle operators as, respectively,

$$S = S_{xy}a^\dagger_x a_y \qquad \text{and} \qquad T = T_{xr,ys}a^\dagger_x a^\dagger_r a_y a_s, \qquad x,y,r,s \in \{i,\alpha\}. \tag{19}$$

Here and in the following we imply summation over equal index pairs.

The matrix elements need to be derived separately for all combinations of our subspaces $\mathcal{N}, \mathcal{I}$ and $\mathcal{A}$. In the derivation one can take advantage of the orthonormalization $\langle j|\alpha_\perp\rangle = 0$. Only in the final step one expresses the matrix elements in terms of the $|\alpha\rangle$ and $|i\rangle$. For example, one obtains for the matrix elements of a single-particle operator $\widehat{S}_{\mathcal{AA}}$ on $\mathcal{A}$

$$\langle A|a_\alpha S a^\dagger_\beta|B\rangle = \delta_{AB}S_{\alpha\beta} + \delta_{\alpha\beta}\rho^{AB}_{ij}S_{ij}, \tag{20}$$

with the single particle reaction density matrices $\rho^{AB}_{ij} := \langle A|a_i a^\dagger_j|B\rangle$. One resolves the orthogonalization as

$$S_{\alpha\beta} = \langle\alpha_\perp|S|\beta_\perp\rangle = \langle\alpha|S|\beta\rangle + U_{\alpha k}\langle k|S|\beta\rangle + \langle\alpha|S|k\rangle U^\dagger_{\beta k} + U_{\alpha l}\langle l|S|k\rangle U^\dagger_{\beta k} \tag{21}$$

One can read off the matrix blocks $\hat{s}^{(0)}$ and $\hat{s}^{(2)}$ entering Eq. (8) in the case of single-particle operators $M = S$

$$\hat{s}^{(0)}_{\alpha A,\beta B} = \delta_{AB}\langle\alpha|S|\beta\rangle + \delta_{\alpha\beta}\rho^{AB}_{ij}\langle i|S|j\rangle, \tag{22}$$

$$\hat{s}^{(2)}_{uA,vB} = \delta_{AB}\langle u|S|v\rangle, \quad u = \alpha, i, \quad v = \beta, j. \tag{23}$$

An important practical complication for the haCC expansion arises from the combination of ionic CI states with single-electron orbitals: for two-particle operators $T = T_{ik,jn}a^\dagger_i a^\dagger_k a_j a_n$ one needs the three-particle reaction density matrices of the ionic states A, B:

$$\langle A|a_u a^\dagger_i a^\dagger_k a_j a_n a^\dagger_v|B\rangle = \langle A|a^\dagger_i a^\dagger_k a^\dagger_v a_u a_j a_n|B\rangle + \ldots \tag{24}$$

We are using a customized version of COLUMBUS [22], where we can directly access the full CI function to construct and save these matrices. The customized version of COLUMBUS is not publically available, but data for selected systems is supplied on a repository [25]. Data for other systems can be made available upon request.

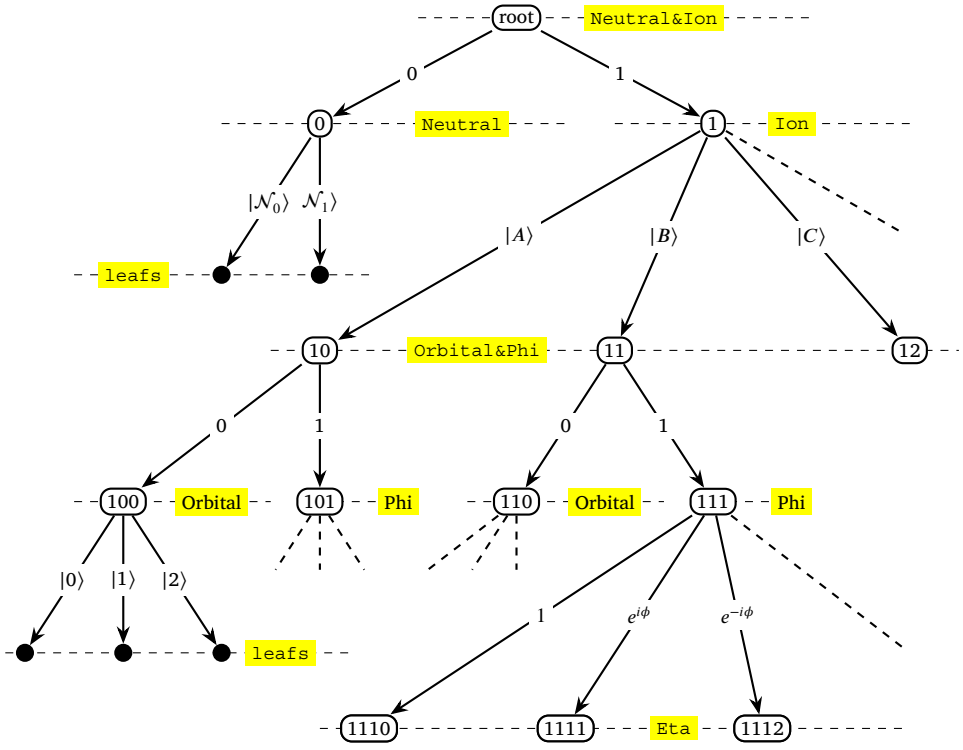

Fig. 1. Index hierarchy for haCC (abbreviated upper part). Axis names are indicated in yellow. Nodes on the respective levels are labeled by J_l, factor basis functions connect a node to the next-lower level. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

2.6. Representation in tRecX

The haCC expansion consists of a hierarchy of a "hybrid" discretizations within the recursive scheme of tRecX (see Ref. [6]). Hybrid here refers to the fact that the same domains in space are discretized with different types of expansion functions. The neutral CI states $|\mathcal{N}\rangle$ are defined on the same domain of multi-electron coordinates as the channel functions $a_i^\dagger|A\rangle$, the orbitals $|i\rangle$ and the numerical functions $|\alpha\rangle$ are both defined on $\mathbb{R}^3$.

In tRecX, any expansion is represented as a tree structure, starting from the total wave-function Ψ at its root and the branches attached to each node of the tree are labeled by all indices down to the node's position in the tree, see Fig. 1. The expansion coefficients $c_{\mathcal{N}}$, c_i^A, c_α^A are the leafs in a tree with separate branches for neutrals and channels and further in each channel for molecular orbitals i and the numerical functions α. Within each α, magnetic quantum numbers m, angular momentum l, the intervals $[r_n, r_{n+1}]$, and finally the individual Lagrange polynomials are themselves arranged in a tree. The tree-structure is recursive and induces, in a natural way, recursive algorithms for computing all pieces of the matrix. In the recursive algorithms one only needs the indices of the branches at given node, but one never deals with the multi-indices that would describe the exact position in the tree. The discretization tree of Fig. 1 is defined as `tRecX` input in the form

```
Axis: subset,    name,          functions,          order,  upperEnd,  nElem
      Neut,      Neutral,       CI[Neutral],        2
      Chan,      Ion,           CI[Ion],            3
      @Molecular, Orbital,      CHEMICAL
      @Perpendicular, Phi,      expIm,              3
                 Eta,           assocLegendre{Phi}, 5
                 Rn,            polynomial,         10,     20,        4
                 Rn,            polExp[0.5],        25
```

The first input line after `Axis` specifies that two neutral states should be used, the second line specifies three ionic channels. These are the states, $|\mathcal{N}_1\rangle$, $|\mathcal{N}_2\rangle$ and $|A\rangle, |B\rangle, |C\rangle$, respectively, in Fig. 1. The following lines define the expansion for the free electron at each ionic channel. It consists of an `Orbital` branch, which contains all I molecular orbitals and the $|\alpha\rangle$ basis with three functions $e^{im\phi}, m = 0, 1, -1$ on the `Phi` axis followed by `associatedLegendre` functions $P_l^{|m|}(\eta)$ with $l < 5$. The axis name `Eta` refers to $\eta = \cos\theta$. The two lines with axis name `Rn` define the radial finite element functions $\xi_n^i(r)$ (cf. Eq. (3)) with 4 equal size elements on $r \in [0, 20]$ and 10 Lagrange polynomials on each element and 25 polynomials damped by $\exp(-0.5r)$ on $[20, \infty)$. The `Rn` hierarchy level is not shown in Fig. 1.

For realistic systems the efficiency of calculations profits from a detailed adjustment of radial and angular expansion near the atoms and fine-tuning using input as above can become demanding. While this can still be done by the user, custom discretizations for atoms and molecules are provided together with the corresponding chemical data and only global parameters like polynomial degree and minimal angular momentum need to be specified in a summary command `AxisSystem`.

3. Classes specific for haCC

The most important classes in tRecX are discussed in [6]. The main operator class is `OperatorTree`, whose construction, data, and application is parallelized. At the leafs of an `OperatorTree` one finds `class OperatorFloor`, which usually are moderate size linear operators that implement structure like diagonality or tensor-product form. In haCC, because of the exchange terms, all floors of the field-free Hamiltonian $\hat{H}_0$ are full matrices.

The complete Hamiltonian matrix $\hat{H}$ appears as an `OperatorTree` at the root level that contains the $\hat{H}_0$ and all time-dependent terms of the mixed-gauge dipole interaction as its first level of branches. Each of these splits into the 2×2 blocks corresponding to the `Neutral` and `Ion` branches as in Fig. 1. The scheme recursively continues on each of these 4 blocks until the floor levels of left and right indices are reached.

3.1. `OperatorHaCC`

This class implements the construction and application of operators as in Eq. (7). It is derived from `OperatorTree`, but its `apply`-function is re-implemented as the application of, in sequence, $\hat{U}^\dagger$, $\hat{m}$, and $\hat{U}$. These three operators are each implemented as an ordinary `OperatorTree`. All terms of $\hat{m}$ operate on an extended index space, where an auxiliary orbital node is attached to each `Orbital&Phi` branch in addition to `Orbital` and `Phi`, see Fig. 1, and $\hat{U}$ maps from this extended space to the original. The cost for that transformation is low and it is done only once for all terms in the operator, as

$$\hat{H}(t) = \hat{U}\left[\hat{h}_0 + \sum_k f_k(t)\hat{h}_k\right]\hat{U}^\dagger, \tag{25}$$

where $f_k(t)$ denote the time-dependent factors $\mathcal{E}_z(t)$, $A_z(t)$, etc.

3.2. `BasisOrbitalNumericalGaussian`

Basis functions in tRecX are all derived from a base class `BasisAbstract` which requires the basis to have a `size()`. Derived classes are, e.g. `BasisGrid` for a set of discrete points on the given axis, or `BasisExpIm` for $e^{im\varphi}$. Multi-dimensional basis functions can be defined by a set of numerical functions, which then are expressed on a discretization. Ref. [6] contains a more comprehensive discussion.

The class `BasisOrbitalNumericalGaussian` represents the molecular orbitals by an expansion into spherical harmonics and also by the original expansion into Gaussians. The spherical expansion uses the same radial element boundaries r_n as the α-basis Eq. (3), but the expansion size on each element is automatically increased until one obtains a converged basis $|\gamma\rangle$, where overlap and kinetic energy agree with the exact Gaussian value to within a given tolerance. In that way, the tRecX standard scheme for operator definitions can be used for evaluating the matrix elements of any single-particle operators S as

$$\langle i|S|\alpha\rangle = \sum_{\gamma\gamma'}\langle i|\gamma\rangle(\hat{o}^{-1})_{\gamma\gamma'}\langle\gamma'|S|\alpha\rangle, \tag{26}$$

where $\hat{o}_{\gamma\gamma'} = \langle\gamma|\gamma'\rangle$ is the FE-DVR overlap for the γ-basis. For evaluating $\langle\gamma'|S|\alpha\rangle$ the standard tRecX scheme is used, where operators are defined in quasi-mathematical notation, see Ref. [6]. This enhances code flexibility and reduces programming errors.

3.3. `InverseHaCC` *and* `InversePerp`

The overlap consists of two independent blocks on $\mathcal{N}\oplus\mathcal{I}$ and $\mathcal{A}$, respectively. Pseudo-inverses for both blocks by the Woodbury formula are set up in `InverseHaCC`. On the top level, `InverseHybrid` implements Eq. (17) for the $\mathcal{N}\oplus\mathcal{I}$ block. The class `InversePerp` implements $\mathcal{A}\mathcal{A}$ pseudo-inverse block for the $|\alpha_\perp\rangle$.

4. The tSurff and iSurff methods

The method of time-dependent surface flux (tSurff) [8,9] allows the efficient computation of photoelectron spectra using simulation volumes that are much smaller than the size to which the wave function expands during the interaction with the laser pulse. In tSurff one integrates the flux that leaves the simulation volume with appropriate time-dependent phases that account for the fact that electron momenta get modified by the laser field long after electrons have left the simulation box. The spectral amplitude accumulated from time $t=0$ until $t=T$ has the general form

$$b(\vec{k},T) = \int_0^T \langle\chi_{\vec{k}}(t)|S(t)|\Psi(t)\rangle dt \tag{27}$$

where $\Psi(t)$ is the time-dependent wave function, $S(t)$ is a time-dependent flux operator at radius $r=R_c$ with a modification due to the external field, and $\chi_{\vec{k}}(t)$ are plane waves with time- and field-dependent phases. As $S(t)$ is non-zero only at $r=R_c$, all that is needed for computing spectra is the time-dependent flux at the boundary of the simulation volume R_c until all relevant amplitude has left the simulation volume. A comprehensive discussion of tSurff for single- and multi-electron emission can be found in Refs. [8,9].

Together tSurff and infinite range exterior scaling (irECS) [7], which is a mathematically rigorous and computationally efficient method for reflectionless absorption, form the name-giving constituent methods of the tRecX = tSurff+irECS code.

A method for computing differential photoelectron spectra from the wave function at the end of the laser pulse had been proposed in Ref. [26] and was used for the photo-emission spectrum from the H_2. That method relies on the solution of the resolvent equation for the full Hamiltonian. An analogous method can be derived from Eq. (27): one observes that, without the external field, $\chi_{\vec{k}}(t) = \exp[-i(t-T)|\vec{k}|^2/2]\chi_{\vec{k}}(T)$ corresponds to free motion, S becomes time-independent and coincides with the standard flux operator, and $\Psi(t) = \exp[-i(t-T)H_0]\Psi(T)$ can be obtained by exponentiation. With this, the integral Eq. (27) from time $t=T$ to $t=\infty$ reduces to

$$b(\vec{k}) = \lim_{\epsilon\downarrow 0}\langle\chi_{\vec{k}}(T)|S\left(H_0 - \frac{\vec{k}^2}{2} + i\epsilon\right)^{-1}|\Psi(T)\rangle. \tag{28}$$

Upon discretization $H_0 \to \hat{H}_0$ the limit may become ill-defined, which shows as numerical instability when $\vec{k}^2/2$ approaches any of the discrete eigenvalues E_i of $\hat{H}_0$. However, with complex scaling the continuous spectrum of H_0 rotates into the lower complex plane and so do the eigenvalues E_i of the complex scaled matrix $\hat{H}_0$ and the inverse can be directly evaluated by setting $\epsilon = 0$.

The computational challenge for using this formula is that, for each photoelectron energy $E = \vec{k}^2/2$ one needs to solve the linear system

$$(\hat{H}_0 - E\hat{O})|\varphi\rangle = \hat{O}|\Psi(T)\rangle \tag{29}$$

where $\hat{O}$ is the overlap matrix. For general $N\times N$ matrices $\hat{H}_0$, solving Eq. (29) scales as N^3. When $\hat{H}_0$ is block diagonal, as in the case of atoms, the effort reduces accordingly. In tRecX, rather than repeatedly solving the linear system, we use a spectral decomposition of the complex scaled $\hat{H}_0$, which is a one-time $\mathcal{O}(N^3)$ calculation. The decomposition has the form

$$\exp\left(-it\hat{O}_\epsilon^{-1}\hat{H}_0\right) = \hat{R}\exp\left(-it\hat{d}\right)\hat{L}^T\hat{O}, \tag{30}$$

where $\hat{R}$ and $\hat{L}$ are the matrices of right and left eigenvectors of generalized eigenvalue equations for the Hamiltonian matrix $\hat{H}_0$

$$\hat{H}_0\hat{R} = \hat{O}\hat{R}\hat{d}, \qquad \hat{L}^T\hat{H}_0 = \hat{d}\hat{L}^T\hat{O} \tag{31}$$

with the overlap matrix $\hat{O}$, its pseudo-inverse $\hat{O}_\epsilon^{-1}$, Eq. (17), and the diagonal matrix of eigenvalues $\hat{d}$. The distinction of right and left vectors is needed, as with absorption $\hat{H}_0$ is not hermitian $\hat{H}_0 \neq \hat{H}_0^\dagger$. For some bases, one may maintain complex symmetry $\hat{H}_0 = \hat{H}_0^T$, in which case $\hat{L} = \hat{R}$. Left and right eigenvectors are orthogonal w.r.t. $\hat{O}$ in the sense

$$\hat{L}^T\hat{O}\hat{R} = \hat{P}_\epsilon, \tag{32}$$

where $\hat{P}_\epsilon$ is the projector onto the non-singular subspace, Eq. (18).

The combination of tSurff with the time-independent integral Eq. (28) was first proposed in Ref. [18] under the name of iSurff. It is particularly useful for the low-energy part of the spectrum, for which wave-function density leaves the simulation box slowly and tSurff needs to be propagated for a long time after the end of the pulse. Another application is the presence of long-lived resonances, which may exhibit time-scales that far exceed the pulse duration. When comparing iSurff

to Ref. [26] one may use much smaller simulation boxes. This reduces both, the propagation time through the pulse and the time needed for solving the system spectral decomposition Eq. (30) or, alternatively, the linear system solving Eq. (29).

The present release of tRecX includes `iSurff` as a standard routine which is invoked by the input `Spectrum:iSurff=true`. The implementation uses classes from tSurff for computing surface values and the momentum-dependent phases, and maps to the momentum grid. For the spectral decomposition, the class `DiscretizationSpectral` is employed.

5. Parallelization and scaling

tRecX is parallelized using MPI but it will also compile without MPI, if no MPI implementation is detected on the system. Finest grains for parallelization are sections of the coefficient vectors of length $10 \sim 100$, the coefficient "floors", and the corresponding `OperatorFloor` blocks. Each coefficient floor is hosted on one process, and the `OperatorFloor` is "owned" by the process that hosts either its input or its output coefficients. The distribution of `OperatorFloor` is otherwise arbitrary and they are moved in order to achieve best load balancing. As the application cost of `OperatorFloor` varies largely due to varying sizes and partially sparse structure, load balancing is achieved by redistributing the `OperatorFloor`'s after setup, based on timing the application cost for each floor.

When using haCC, where $\hat{H}_0$ is full because of exchange and dominates application time, column-wise distribution gives good load balancing, if one adjusts column boundaries using the actual timings. The time-dependent parts have much lower application cost and use the same parallel layout as $\hat{H}_0$.

5.1. Class `FlattenedOperator`

This class controls distribution and parallel application of an `OperatorTree`. It "flattens" the original tree structure into a vector of `OperatorFloor`'s, which allows equal treatment of all `OperatorFloor`'s, independent of their position in the `OperatorTree`. The mapping is by pointers, which allows keeping the original tree structure for reference on all processes without creating copies of the `OperatorFloor`'s. For each `OperatorFloor`, only its owner process has actual data, while there are dummies with negligible memory on all others. The more complex classes `OperatorHaCC` or `InverseHybrid` are parallelized by flattening all their `OperatorTree` factors.

5.2. Operator setup

Considerable compute resources are consumed by the computation of matrix elements of two-particle operators on $\mathcal{A}$ and the mixed Gaussian-numerical matrix elements between $\mathcal{A}$ and $\mathcal{I}$, where up to three-particle reduced reaction density matrices enter (see Sec. 2.5). As to be expected, the most time-consuming part is the computation of exchange on the $\mathcal{A}$-subspace. The integrals are calculated by representing Coulomb repulsion of electrons $|\vec{r}-\vec{r}'|^{-1}$ in a multipole expansion for the angular degrees of freedom and a quadrature grid in the two radial coordinates (r, r') that is exact for the basis, see Ref. [13] for details. Parallelization is by distributing the pairs of radial quadrature points on a given FE-DVR mn-patch $[r_m, r_{m+1}] \times [r'_n, r'_{n+1}]$ across the parallel processes. This strategy was chosen over distributing whole patches, as the number of angular momenta and quadrature points can vary greatly between different mn-patches and load balancing is easier to achieve between the points within the same mn-patch. Variations of that strategy were used for parallelizing the evaluation of all other two-electron integrals. As setup still can take significant time, one can save the operators for a given expansion and re-use it for re-runs within the same expansion, where hashing ensures access to the correct matrices.

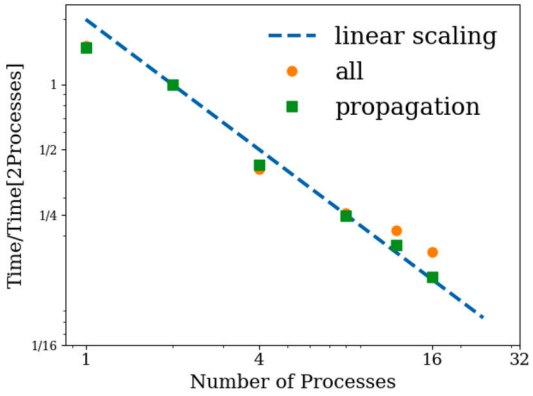

Fig. 2. Scaling of total computation and setup times for medium size haCC calculation. Results are comparable to the scaling reported in Ref. [6]. Times are normalized to the 2-process calculation, as some algorithms differ in the strictly scalar code with a single process.

5.3. Scaling

Floating operations and communication are dominated by $\hat{H}_0$, which is a full matrix because of exchange, whereas the time-dependent operators are implemented by sparse and partially local operations, as shown in Sec. 2.3. At any parallel run, tRecX will print measured load-balancing and the communication patterns of the major operators, which helps to detect load balancing problems. With the full matrix $\hat{H}_0$, load-balancing in haCC applications is typically within a few percent, with slightly poorer balancing for the remaining parts of the operator, which, however, consume only a fraction of the compute time.

Application of the full matrix and the limit to granularity in terms of the `OperatorFloor`'s allows favorable scaling only up to about 16 processes, depending on the problem size and hardware. Fig. 2 shows the scaling at fixed problem size ("strong scaling") for the example of photo-ionization of the CO molecule.

6. Input, documentation, output, and compilation

Input is defined through files and handled internally by `class ReadInput`, which also supports documentation of the code, as discussed in some detail in Ref. [6]. Otherwise it is conscious policy *not* to provide a separate document that describes the code, other than the present article and Ref. [6] that outline the capabilities of the code and its structure. Code use is systematically introduced by the tutorials that are provided with each release. The most important of these are discussed below and in Ref. [6]. Documentation of the input is written into the code using the class `ReadInput`. A short reference for the input options can be displayed by running tRecX without any arguments, more extended help for each input category is shown by the `-h` flag, e.g. `./tRecX -h=Axis` for a brief summary for every entry related to the input category `Axis`. Finally, for many inputs longer explanatory texts are written right with the respective `ReadInput:read` command. These are routinely processed during every code sanity check and automatically generate an up-to-date `UserManual.pdf` that is published with the code. For more technical purposes the `C++` is marked by Doxygen [27] documentation and further in-line comments.

Run directory Each time the code is started with a given input file name `SomeName.inp` it generates a "run directory" `SomeName/xxxx`, where the 4-digit code `xxxx` is the first available starting from `0000`. A copy of the original input and all outputs are placed into that directory.

A few python scripts for summary display of multiple runs, plotting and comparing spectra and submitting multiple runs to a queuing system are provided with the package. A brief description is given in `UserManual`, up-to-date info is displayed by running the scripts without arguments.

Compilation of the code is by `cmake/make`, which has been tested on both, Linux-based and Mac systems, but only the Linux version is actively maintained. The code makes use of a range of public domain libraries. In order to secure matching library versions the `Eigen` template library [16] and `ALGLIB` [28] are supplied with the distribution. Up-to-date build instructions and protocols of successful builds are included with the `README.md` file.

7. Usage examples

Up-to-date information about possible applications in the form of tutorials is shipped with the code [6] and is available also in the AMOS portal [29]. For the discussion of applications already available in the first release we refer the reader to [6]. The following applications all involve haCC.

Computation times strongly depend on the problem. In particular, long wave-lengths require higher angular momenta and longer propagation times. In general, because of $\hat{H}_0$ being a full matrix, times grow quadratically with discretization size. The examples below were chosen for easy reproducibility with compute times varying between ~ 5 minutes at wave-length of 20 nm on 4 cores of a laptop for the largest calculation shown below in Fig. 3 to three hours at 400 nm on 8 cores of a small cluster for the largest calculation of Fig. 5.

7.1. Photo-emission and high harmonic generation with Helium

While the Helium atom has significant correlation in its ground state, strong field processes are assumed to be effectively single-electron in many cases. We examine this assumption for three different situations: photoelectron spectra generated by a short pulse at the extreme ultraviolet (XUV) wave length, strong-field ionization at visible wavelength and high harmonic generation.

A minimal input for computing photo-emission is (see `tutorial/200HeHaCC`)

```
Chemical: data=../Chemical/He19_Standard
AxisSystem: lMax=4
Spectrum: radialPoints=200, plot=channels, iSurff=true
Laser: shape, I(W/cm2), FWHM,   lambda(nm)
       cos8,   1.e16, 3. OptCyc, 20
```

The input `AxisSystem` directs the code to use a default discretization that is specified together with the pre-processed quantum chemical data and that is safe for standard applications. As with any such calculation, convergence needs to be confirmed by varying the discretization. The maximal angular momentum `AxisSystem:lMax` must always be specified. For the given laser parameters, `lMax=4` is sufficient, as hardly more than 1 photon is absorbed. The spectrum will be computed on a momentum grid of 200 `radialPoints`, the plot will be for the channel-wise energy spectrum, and `iSurff` will be used. Laser pulse parameters are intensity 10^{16} W/cm^2, full width at half maximum (FWHM) of 3 optical cycles and central wave length of 20 nm. Inputs are understood to be in atomic units (au). An exception are names that explicitly indicate the input units, like `I(W/cm2)` and `lambda(nm)`. However, one can specify the input unit next to the input value. In this example, we have chosen to specify `FWHM` in optical cycles `OptCyc` rather than the default au. Conversely one could, e.g., give the intensity as `1 au`, which would correspond to $\approx 3.51\,10^{16}$ W/cm^2. Many other input units like seconds, meters, etc. can be given in a similar way, see Ref. [6] for a more comprehensive discussion.

The details of the discretization generated by the `AxisSystem` directive are written to a file `axisSystem` in the run directory. That file has the format of a standard tRecX input. Most importantly, it contains an `Axis` definition as discussed in Sec. 2.6 where the radial sections for the α-basis are chosen to match the given `Chemical:data`. Some of the parameters of that discretization can be changed by specifying, e.g. `AxisSystem:nChan` for the number of channels or `AxisSystem:rMax` for the radius of the simulation volume. More experienced users can include the file `axisSystem` into the input and change the given values, e.g., when performing a more detailed convergence study.

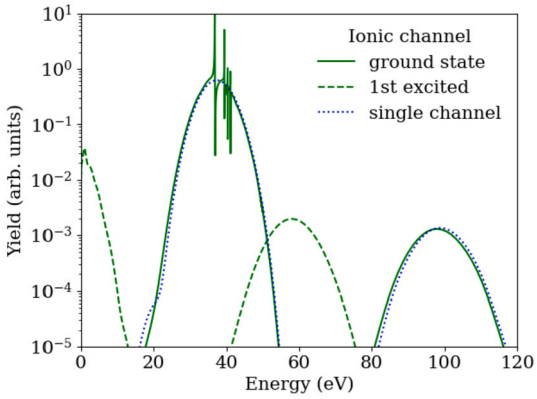

Fig. 3. Photo-emission from the He atom generated by a short intense pulse at a central wave length of 20 nm, intensity of 10^{16} W/cm^2, FWHM of 3 optical cycles and a $\cos^8$ pulse envelope.

By default, the calculation is with a single channel. The number of channels is set by

```
AxisSystem: nChan=2
```

for the ground and the first excited ionic states. Matrix sizes grow with the square of the number of channels, i.e. the 2-channel calculation will take about 4 times the resources of the single channel calculation. For `iSurff` the growth is with the 3rd power, as the eigenproblem for the full matrix needs to be solved.

Fig. 3 shows the photo-emission spectra for a single- and a two-channel calculation. The broad features are the single-photon peaks for the ground and first excited ionization channel. On top of the first peak of the ground state channel there is a group of 2s-np resonances due to the coupling to the excited channel, see Sec. 7.3. Except for that, at the given parameters, the only multi-channel effect in the spectrum is a small shift to lower energies in the two-channel calculation, due to a slightly lower ground state energy in the multi-channel calculation.

When doing several calculations with the same discretization one can add the lines

```
Cache:
hamiltonian[He19] interaction
```

which specifies that operators will be cached in a directory defaulting to `CACHE_TRECX` with filenames beginning with `He19` and unique hash indicating the actual discretization.

At longer wave length, one needs to increase the size of the angular momentum expansion. We show calculations at 400 nm wave length, for which we adjust the input lines to (see `tutorial/201He400`)

```
Dipole: velocity, length
AxisSystem: lMax=25
Laser: shape, I(W/cm2),   FWHM,   lambda(nm)
         cos2, 2.e14,  6. OptCyc, 400
```

The wavelength of 400 nm is in the visible, pulse duration now 6 optical cycles in order to get well-separated photo-emission peaks. Angular momenta have been increased to 25, as this is in the truly multi-photon regime. Also, we here use a $\cos^2$ pulse envelope. When used with very short pulses, the $\cos^2$ envelope can create artefacts, about which tRecX will issue a warning. However, the pulse shape is popular, and the artefacts are unimportant for the present demonstration purpose. The additional line `Dipole:velocity,length` will cause the dipole expectation values to be computed in, both, length- and velocity-form, from which harmonic responses are computed automatically. The acceleration form of the dipole cannot be used with haCC, because the code cannot form the derivative of the mean-field and exchange potentials.

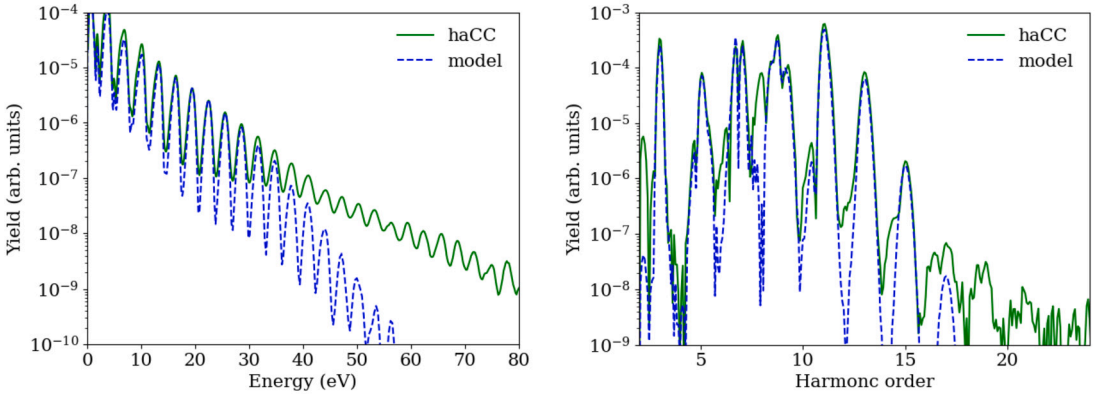

Fig. 4. Photoelectron (left panel) and high harmonic spectra (right panel) emitted from Helium in laser pulse with wave length 400 nm, peak intensity 2×10^{14} W/cm^2 and 6 optical cycles pulse duration. Single-electron model (blue, dashed) vs. haCC (green, solid).

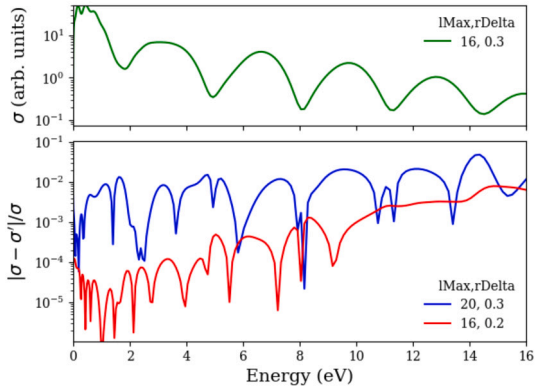

Fig. 5. Convergence of the photoelectron spectrum for CO in a 6-cycle laser pulse at wave length 400 nm and peak intensity 10^{14} W/cm^2. Spectrum σ (upper panel) and relative difference $|\sigma-\sigma'|/\sigma$ (lower panel) comparing σ with `lMax=20` angular momenta and the default radial spacing of `rDelta=0.3` with spectra σ' with 16 angular momenta (blue) and radial spacing 0.2 (red), respectively.

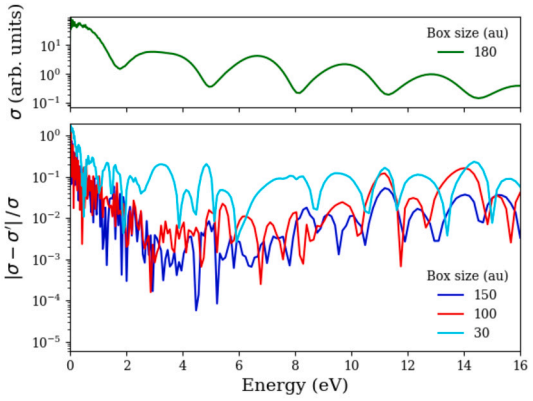

Fig. 6. Long-range Coulomb effects on photoelectrons. Comparing a calculation with box size 180 to calculations with sizes from 30 to 150 au. Up to box size of ~ 50 errors are on the scale of 10%. By the convergence from box sizes of 180 through 150 to 100, one estimates an accuracy at photoelectron peak heights at box size 180 of $\lesssim 1\%$, except near threshold.

Figs. 4(a) and (b) show photo-emission harmonic spectra for this case. The haCC calculations are numerically converged to within $\lesssim 10\%$ relative accuracy for the peak maxima in the range shown. For comparison, the figure includes a single-electron calculation with the model potential

$$V(r) = -\frac{1+\exp(-2.1405r)}{r}, \tag{33}$$

where the screening constant is adjusted for an ionization potential of 0.90186 a.u. which matches the ionization potential obtained out of the COLUMBUS calculations. Important differences appear in the high-energy part of photoelectron emission and in the noise level of the harmonic spectra. The haCC calculation shows significantly more yield at high electron energies. Without further interpreting the result at this point, we observe that the deviations become dramatic beyond photoelectron energies of 30 eV, which is the well-known 10 U_p cut-off for photo-emission by single recollision [30]. The haCC calculation includes correlation and anti-symmetrization, but the given difference may also reflect the different behavior of the electron wave functions near the nucleus in the two calculations.

7.2. Convergence and accuracies: photo-emission from CO

For the assessment of the accuracy of a tRecX calculation one needs to examine the numerical convergence of a given observable w.r.t. discretization parameters. We discuss this here on the example of the photoelectron spectrum of the diatomic CO in a laser pulse of 400 nm wavelength, 3 optical cycles FWHM pulse duration and intensity of 10^{14} W/cm^2. With the inputs

```
Chemical: data=../Chemical/COlarge5_Standard
AxisSystem: lMax=16
Spectrum: radialPoints=200, plot=channels
Laser: shape, I(W/cm2), FWHM,  lambda(nm)
        cos2,   1.e14, 3. OptCyc, 400
TimePropagation: end,     print
              4 OptCyc, 1/8 OptCyc
```

we obtain the photo-emission spectra from CO as shown in Fig. 5.

Convergence w.r.t. angular momenta and radial discretization Fig. 5 shows the relative differences of the spectrum compared to two calculations with more angular momenta `lMax=20` and higher density of DVR points on the radial axis, which is set by `AxisSystem:rDelta=0.2` compared to the default value of 0.3. Based on these relative differences we estimate that the calculation is numerically converged w.r.t. to these parameters to about 1% relative accuracy at the respective photoelectron peak maxima.

Long-range Coulomb effects The long-range Coulomb potential invariably affects photoelectron spectra. This can be exposed by increasing the radius of the simulation volume over the default value of 30 au (for CO)

```
AxisSystem: rMax=50
```

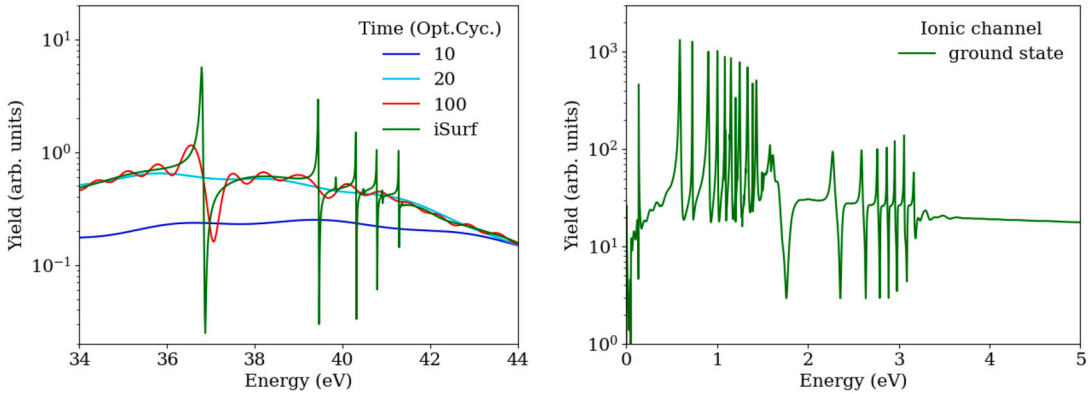

Fig. 7. Applications of iSurff: resonances due to doubly excited states. Left panel: detail of Fig. 3 obtained with iSurff (green), comparing to pure tSurff with finite propagation times. Right panel: resonances in the photo-ionization of N_2 into the ground state channel by a single-cycle pulse at 68 nm nominal wave length.

Note that there is a correlation between the radius and the number of required angular momenta: to some extent the growth of the volume is two-dimensional in case of cylindrical symmetry and one needs to re-establish angular convergence when using larger box size. Fig. 6 shows changes of about 10% throughout the main part of the spectrum as one decreases the simulation box size from a maximum of 180 au to 30. As to be expected, differences are largest at low energies that are most affected by the weak, but long-range Coulomb interaction. At energies $\gtrsim 2$ eV the calculation at box size 180 can be assumed to be converged to $\lesssim 1\%$ in relative accuracy. With the effect being due to the Coulomb potential, the box size needed for a given accuracy in a system with a given ionization potential can be estimated using a hydrogen-like model with the effective charge adjusted to match the ionization potential.

Clearly, no general accuracy statements can be made across all laser parameters and observables. As in any such calculation, careful investigation of convergence for a given purpose remains the user's responsibility.

The file `axisSystem` contains the definition of the discretization near the nuclei. The values provided have been tested to be safe for about 10% accuracies for photoelectron spectra in the range from XUV to 800 nm wavelength and intensities that lead to only partial ionization. In the present calculation convergence w.r.t. to these further parameters is well below 1%. When calculating at more extreme parameters, longer wave length or higher intensities, one also needs to test convergence w.r.t. the discretization near the nuclei. For that one would replace the input `AxisSystem` by the contents of the `axisSystem` file and examine sensitivity of the results to the various additional discretization parameters. A list of the relevant convergence parameters and explanations of their meaning can be found in the `UserManual`.

7.3. Application of iSurff: resonances in He and N_2

Fig. 7 shows resonances the ground state channel for the He atom and the N_2 molecule. The resonances arise from doubly excited states in the excited ionic channels. Because of the long lifetime of the resonances, one needs to use `iSurff=true` for resolving the peaks, which amounts to integration to infinite times.

In the He calculation (left panel) we include pure tSurff spectra, obtained with the inputs

```
Spectrum: iSurff=false
TimePropagation: end=10 OptCyc
```

with longer propagation times up to `end=100` optical cycles. At `end=10` optical cycles, significant part of the electron probability has not passed the surface, with 20 cycles the broad peak is essentially complete, but only at times ~ 100 optical cycles the peaks from the slowly decaying resonances start to appear. This illustrates how the long-lived resonances are typical applications for `iSurff`.

The right panel shows photo-emission from the N_2 molecule at wave-length of 68 nm and pulse duration of 1 optical cycles FWHM, with two groups of resonances which correspond to double excitations above the first and second ionization thresholds. The calculation is included here for showing the capabilities of tRecX and the further discussion of the resonances will be presented elsewhere.

7.4. The tRecX GUI on the AMOS gateway

An HTML-based GUI was designed for tRecX and implemented in the AMOS Gateway [29] that hosts several codes for atomic, molecular and optical sciences. The codes and a range of resources can be accessed through the gateway in a uniform way. The tRecX GUI is primarily for low-threshold access by first-time users. It provides all essential tutorials, that can be amended and changed all the way to completely new calculations. Results can be downloaded and graphics can be produced by interactive access to the `plot.py` script provided with the tRecX release. Fig. 8 shows a screenshot of the interface. Access to a free account to the AMOS Gateway, with limited resources, will be granted after review by one of administrators.

8. Conclusions

The haCC implementation in the present major update of tRecX enables a wider range of specialist users to perform correlated multi-electron calculations for atoms and small molecules. The tRecX framework ensures a uniform and error-safe handling of the calculations. By their nature, such calculations require, in general, significant compute times, except for the simplest atomic cases.

The code uses a new variant of haCC that makes explicit use of the molecular orbitals of the underlying CI calculation. This enhances the multi-center capability of haCC compared to its original formulation in Ref. [1]. An outline of that re-formulated method was presented, which includes the general approach to computing matrix elements, control of near linear dependencies by Woodbury-type decompositions, methods for handling complexity of haCC by recursive schemes, and efficient application of the operators.

We have demonstrated haCC on the example of laser-ionization of the Helium atom and diatomics. Extension to non-linear molecules is planed for a future upgrade. The important question of convergence has been discussed using the non-trivial example of the CO molecule by giving examples for the most important convergence checks, with a more complete list in tRecX's `UserManual.pdf`.

A further major new feature of this release is the iSurff [18] method. The method was re-derived within the existing tSurff formalism and its relation to a well-established earlier method [26] was pointed out. The resolution of long-lived resonances in broad-band XUV excitation of Helium was given as an example.

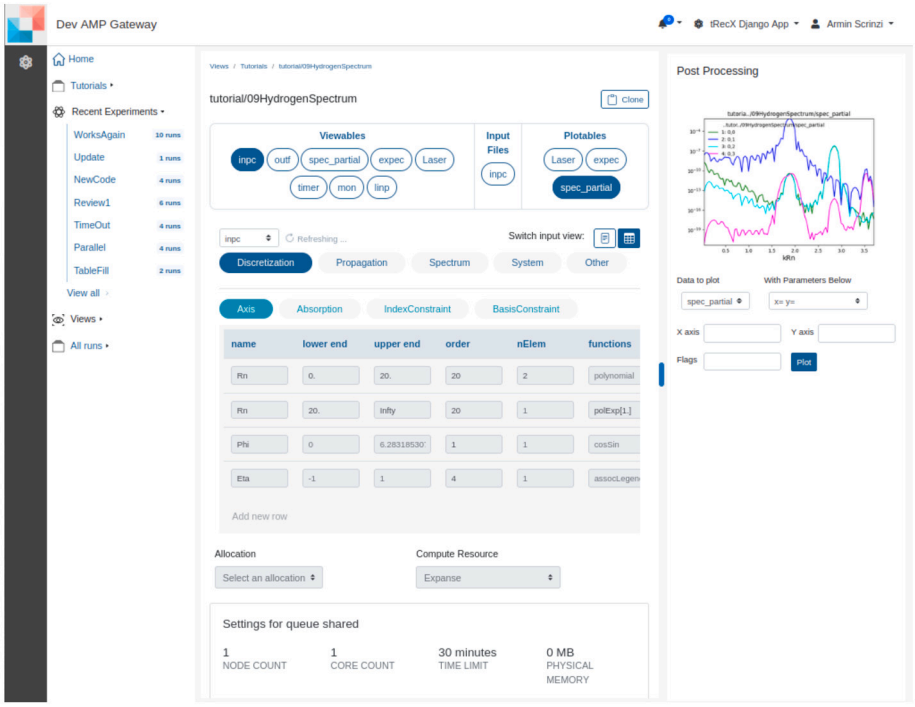

Fig. 8. The GUI for tRecX on the AMOS Gateway. Input items sorted by category, input and output files, and graphical display of the results can be accessed. Basic help on the items appears upon hovering over the buttons. Submission to a selection of resources is controlled in the lower part of the screen. Runs are grouped into "Experiments" shown in the left panel.

The new release contains a range of gradual improvements, like the possibility of saving operators for re-use in later calculations, simplifications for in- and output, extension of the automated documentation, a revision of the parallelization strategy, and improved scripts for data presentation.

The implementation philosophy of tRecX was maintained: it is designed for large scale applications with verifiable accuracies, as well as for training, education, and community development. A designated part of the tRecX development is to ensure user experience that is acceptable to a somewhat wider range of specialist users, including experimentalists who want to generate standard results or study simple models as well as theorist with more complex demands. We consider error safe and intuitive input, extensive consistency checks, and structurally enforced documentation as essential for achieving that goal.

CRediT authorship contribution statement

Hareesh Chundayil: Writing – review & editing, Validation, Software, Methodology. **Vinay P. Majety:** Writing – review & editing, Validation, Software, Methodology, Conceptualization. **Armin Scrinzi:** Writing – original draft, Visualization, Supervision, Software, Methodology, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Armin Scrinzi reports financial support was provided by LMU Munich. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgement

The code was originally developed by the authors and Alejandro Zielinski, with further contributions by, in alphabetic order, Christoph Berger, Jonas Bucher, Florian Egli, Jacob Liss, Mattia Lupetti, David Z. Manrique, Jonathan Rohland, Jörn Stöhler, Andreas Swoboda, Hakon Volkmann, Markus and Michael Weinmueller, and Jinzhen Zhu. V.P.M. acknowledges support from the Science and Engineering Research Board (SERB) India, SRG/2019/001169 startup research grant.

References

[1] Vinay Pramod Majety, Alejandro Zielinski, Armin Scrinzi, Photoionization of few electron systems: a hybrid coupled channels approach, New J. Phys. 17 (Jun 2015).
[2] Vinay Pramod Majety, Armin Scrinzi, Dynamic exchange in the strong field ionization of molecules, Phys. Rev. Lett. 115 (Sep 2015) 103002.
[3] Vinay Pramod Majety, Armin Scrinzi, Multielectron effects in strong-field ionization of CO_2: impact on differential photoelectron spectra, Phys. Rev. A 96 (Nov 2017) 053421.
[4] Vinay Pramod Majety, Armin Scrinzi, Static field ionization rates for multi-electron atoms and small molecules, J. Phys. B, At. Mol. Opt. Phys. 48 (24) (2015) 245603.
[5] Vinay Pramod Majety, Armin Scrinzi, Absence of electron correlation effects in the helium attoclock setting, J. Mod. Opt. 64 (10–11) (2017) 1026–1030.
[6] Armin Scrinzi, tRecX — an environment for solving time-dependent Schrödinger-like problems, Comput. Phys. Commun. 270 (2022) 108146.
[7] Armin Scrinzi, Infinite-range exterior complex scaling as a perfect absorber in time-dependent problems, Phys. Rev. A 81 (5) (May 2010) 053845.
[8] Liang Tao, Armin Scrinzi, Photo-electron momentum spectra from minimal volumes: the time-dependent surface flux method, New J. Phys. 14 (1) (Jan 2012) 013021.
[9] Armin Scrinzi, tSurff: fully differential two-electron photo-emission spectra, New J. Phys. 14 (8) (2012) 085008.
[10] Vinay Pramod Majety, Alejandro Zielinski, Armin Scrinzi, Mixed gauge in strong laser-matter interaction, J. Phys. B, At. Mol. Opt. Phys. 48 (2) (2015) 025601.
[11] T.N. Rescigno, C.W. McCurdy, Numerical grid methods for quantum-mechanical scattering problems, Phys. Rev. A 62 (3) (Aug 2000) 032706.
[12] Markus Weinmüller, Michael Weinmüller, Jonathan Rohland, Armin Scrinzi, Perfect absorption in Schrödinger-like problems using non-equidistant complex grids, J. Comput. Phys. 333 (2017) 199–211.
[13] Alejandro Zielinski, Vinay Pramod Majety, Armin Scrinzi, Double photoelectron momentum spectra of helium at infrared wavelength, Phys. Rev. A 93 (Feb 2016) 023406.

[14] Jinzhen Zhu, Armin Scrinzi, Electron double-emission spectra for helium atoms in intense 400-nm laser pulses, Phys. Rev. A 101 (Jun 2020) 063407.
[15] Lapack, http://www.netlib.org/lapack. (Accessed 12 January 2021).
[16] Eigen - a C++ template library for linear algebra, https://eigen.tuxfamily.org. (Accessed 30 April 2024).
[17] FFTW, http://www.fftw.org. (Accessed 12 January 2021).
[18] F. Morales, T. Bredtmann, S. Patchkovskii, Isurf: a family of infinite-time surface flux methods, J. Phys. B 49 (24) (Dec. 2016).
[19] Andrew C. Brown, Gregory S.J. Armstrong, Jakub Benda, Daniel D.A. Clarke, Jack Wragg, Kathryn R. Hamilton, Zdeněk Mašín, Jimena D. Gorfinkiel, Hugo W. van der Hart, Rmt: R-matrix with time-dependence. Solving the semi-relativistic, time-dependent Schrödinger equation for general, multielectron atoms and molecules in intense, ultrashort, arbitrarily polarized laser pulses, Comput. Phys. Commun. 250 (2020) 107062.
[20] Oleg Zatsarinny, Bsr: B-spline atomic R-matrix codes, Comput. Phys. Commun. 174 (4) (2006) 273–356.
[21] Jesús Gonzalez-Vázquez, Carlos Marante, Markus Klinker, Josep V. Borràs, Inés Corral, Luca Argenti, Fernando Martín, Xchem home page, https://doi.org/10.21950/GHWTML. (Accessed 16 June 2024), 2023.
[22] Hans Lischka, Thomas Mueller, Peter G. Szalay, Isaiah Shavitt, Russell M. Pitzer, Ron Shepard, COLUMBUS — a program system for advanced multireference theory calculations, Wiley Interdiscip. Rev. Comput. Mol. Sci. 1 (2) (2011) 191.
[23] Vicent J. Borràs, Pedro Fernández-Milán, Luca Argenti, Jesús Gonzalez-Vázquez, Fernando Martín, Photoionization cross sections and photoelectron angular distributions of molecules with xchem-2.0, Comput. Phys. Commun. 296 (2024) 109033.
[24] Max A. Woodbury, Inverting Modified Matrices, Statistical Research Group, Memo. Rep., vol. 42, Princeton University, Princeton, NJ, 1950, p. 42.
[25] Columbus-generated data (git repository), https://gitlab.physik.uni-muenchen.de/AG-Scrinzi/chemical. (Accessed 15 March 2024).
[26] A. Palacios, C.W. McCurdy, T.N. Rescigno, Extracting amplitudes for single and double ionization from a time-dependent wave packet, Phys. Rev. A 76 (4) (Oct. 2007).
[27] Doxygen, https://www.doxygen.nl. (Accessed 12 January 2021).
[28] Alglib, https://www.alglib.net/. (Accessed 30 April 2024).
[29] AMOS Gateway, https://amosgateway.org. (Accessed 15 March 2024).
[30] G.G. Paulus, W. Becker, W. Nicklich, H. Walther, Rescattering effects in above-threshold ionization - a classical-model, J. Phys. B 27 (21) (Nov. 1994) L703–L708.