Computer Physics Communications 298 (2024) 109113

Contents lists available at ScienceDirect

Computer Physics Communications

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

Computer Programs in Physics

UKRmol-scripts: A Perl-based system for the automated operation of the photoionization and electron/positron scattering suite UKRmol+ [☆,☆☆]

Karel Houfek [a,*], Jakub Benda [a], Zdeněk Mašín [a], Alex Harvey [a,c], Thomas Meltzer [a],
Vincent Graves [b], Jimena D. Gorfinkiel [b]

[a] *Institute of Theoretical Physics, Faculty of Mathematics and Physics, Charles University, V Holešovičkách 2, 180 00 Praha 8, Czech Republic*
[b] *School of Physical Sciences, The Open University, Walton Hall, MK7 6AA, Milton Keynes, United Kingdom*
[c] *Max Born Institute, Max-Born-Straße 2A, 12489, Berlin*

ARTICLE INFO

Keywords:
Photoionization
Scattering
Transition moments
R-matrix

ABSTRACT

UKRmol-scripts is a set of Perl scripts to automatically run the UKRmol+ codes, a complex software suite based on the R-matrix method to model fixed-nuclei photoionization and electron- and positron-scattering for polyatomic molecules. Starting with several basic parameters, the scripts operatively produce all necessary input files and run all codes for electronic structure and scattering calculations as well as gather the more frequently required outputs. The scripts provide a simple way to run such calculations for many molecular geometries concurrently and collect the resulting data for easier post-processing and visualization. We describe the structure of the scripts and the input parameters as well as provide examples for photoionization and electron and positron collisions with molecules. The codes are freely available from Zenodo.

Program summary
Program Title: UKRmol-scripts
CPC Library link to program files: https://doi.org/10.17632/cpvc2473pt.1
Developer's repository link: https://gitlab.com/Uk-amor/UKRMol/UKRmol-scripts
Licensing provisions: GPLv3
Programming language: Perl
Nature of problem: Performing *ab initio* photoionization and low energy electron- and positron-scattering on polyatomic molecules requires selecting and setting a significant number of input parameters in order to model the physics in a numerically accurate way. These scripts streamline setting up and analyzing calculations based on the R-matrix method [1] using the UKRmol+ suite [2].
Solution method: The scripts provide automatic generation of input files and execution of programs from the UKRmol+ suite [2] using a number of parameters that describe both physical models and machine-dependent settings, and also outputs of the previous programs in the suite which are automatically read and analyzed. The resulting output files are then post-processed to collect target and scattering data for further analysis and simple plotting.
Additional comments including restrictions and unusual features: The scripts should be used with releases 3.2 of UKRmol-in and UKRmol-out [3] although they are compatible with earlier versions.

References

[1] P.G. Burke, R-Matrix Theory of Atomic Collisions: Application to Atomic, Molecular and Optical Processes, Springer, Berlin, 2011.
[2] Z. Mašín et al., UKRmol+: a suite for modelling electronic processes in molecules interacting with electrons, positrons and photons using the R-matrix method, Comput. Phys. Commum. 249 (2020) 107092.

[☆] The review of this paper was arranged by Prof. F. Martin.
[☆☆] 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 address: karel.houfek@mff.cuni.cz (K. Houfek).

https://doi.org/10.1016/j.cpc.2024.109113
Received 21 April 2023; Received in revised form 13 January 2024; Accepted 21 January 2024

[3] UKRMol+: UKRMol-in 3.2, https://doi.org/10.5281/zenodo.5799110, 2021; UKRMol+: UKRMol-out 3.2, https://doi.org/10.5281/zenodo.5799134, 2020.

---

1. Introduction

The increasing computational power available to Atomic, Molecular, and Optical (AMO) physicists has made it possible to run ever more accurate calculations to investigate photon and lepton-induced processes. This has led to more sophisticated, and therefore complex, software suites that require the user to provide input for several interrelated programs which need to be run, serially or in parallel, in a specific order. This is particularly true when the systems under study are molecules and computational chemistry approaches are used to generate the electronic wave functions.

One of the most established AMO codes for ab initio description of the electronic continuum of molecules is the UKRmol+ suite [1]. It is an implementation of the time-independent R-matrix method [2,3] to treat molecular photoionization, low-energy electron and positrons scattering for molecules and to provide input to the R-matrix with time-dependence codes RMT [4]. RMT models atoms and molecules in intense, ultrashort, arbitrarily-polarized laser pulses [5,6]. Among the processes that have been modeled for atoms using RMT are: attosecond transient absorption spectroscopy [7], reconstruction of attosecond beating by interference of two-photon transitions (RABBITT) with skewed laser polarisation [8], and XUV-initiated high harmonic generation (HHG) [9].

The UKRmol+ and RMT suites have so far been used to study one-photon and strong-field ionization and RABBIT time-delays for some diatomic and triatomic molecules [10,11]. In addition, UKRmol+ implements a recently developed time-independent approach to calculate transition moments for the multiphoton ionization, for any number of photons, of molecules in the perturbative regime [12]. The approach can be applied to the calculation of RABBIT time-delays [11], photoionization cross sections, and laboratory-frame photoelectron angular distributions.

Running the UKRmol+ suite for different types of physical processes requires, in principle, producing input files for between 5 and 10+ programs, defining both the models that will be used to describe the physics and the technical characteristics of the particular run, from executables to use, to the number of nodes, tasks, etc. The generation of input files for several consequently executed programs, in particular, is prone to human error: inconsistencies can lead to calculations that either fail or, worse still, produce physically meaningless results and in the process unnecessarily use up computing resources. Moreover, when calculations for many molecular geometries are needed, the manual generation of all inputs can also be tedious and time-consuming.

These problems were recognized and addressed for the UKRmol+ suite and its predecessor (the UKRmol suite [13]) around 10 years ago. A set of Perl scripts, named UKRmol-scripts, was developed to simplify the running of the suite and make it more accessible. The scripts produce the input for all the programs in the suite, run the codes in the appropriate order, and collect the data in a user-friendly format for analyzing, plotting, and comparing. They are significantly more straightforward to master by new users than the UKRmol+ suite itself but allow sufficient sophistication to enable cutting-edge calculations to be performed. The scripts are reasonably straightforward to modify, enabling users to include additional input if required and to update them as new functionality is introduced in the UKRmol+ suite. The scripts are available for download from Zenodo [14].

There have been other approaches to make UKRmol+, and other AMO codes, user-friendly. For example, a commercial package, QEC [15], is available to drive UKRmol+. A wider initiative, to make a range of AMO codes available to users, is the AMOSGateway [16,17]: this is a free, easy-to-use platform that gives access to a number of software suites that calculate cross sections for photoionization, electron and positron scattering, and solutions of the time-dependent Schrödinger equation in the presence of strong and ultrafast electromagnetic fields.

In this paper, we describe the UKRmol-scripts in detail. Our aim is to encourage more researchers to use them together with the UKRmol+ suite and exploit the software capabilities to treat photoionization processes, in particular those induced by ultrashort laser pulses. Section 2 briefly describes the UKRmol+ suite; Section 3 details the use of the UKRmol-scripts, Sections 4, 5 and 6 describe the specifics of running scattering, photoionization, and RMT input calculations respectively, including a description of the examples provided with the release. Section 7 describes some newer models to generate the target and all system configurations. Finally in Section 9, we present some brief conclusions.

2. The UKRmol+ suite

A detailed description of the UKRmol+ suite can be found in [1]. The codes are freely available for download from Zenodo [18,19]. Releases include a set of CMake scripts for the configuration, compilation, testing, and installation of the suite. The suite makes use of the GBTOlib library [20] for computing the required integrals.

The basis of the R-matrix approach is the separation of space into two regions. The boundary between them is a sphere of radius a. In the outer region, correlation as well as exchange between the unbound particle (when it is an electron) and the molecular electrons can be neglected. In the inner region, these effects need to be taken into account. Quantum chemistry approaches, including the use of Gaussian-type orbital (GTO) basis sets and configuration interaction, are employed to describe both the neutral or cationic bound molecular target states and those of the (anionic or neutral) target + unbound particle system. In the outer region, the interaction potential between the target and the unbound particle is modelled in terms of a multipole expansion.

The suite requires external quantum chemistry codes, such as Molpro [21], Molcas [22], or Psi4 [23], to generate bound molecular orbitals. The unbound particle is described in the inner region through the use of purpose-built Gaussian and/or B-spline basis sets, GTOs and BTOs respectively, to represent the continuum [1]. The configuration state functions used to describe both the target and the whole system need to be carefully selected to accurately describe the physics of the problem while ensuring the calculation remains tractable.

The suite allows the user to determine quantities associated with photoionization (cross sections, asymmetry parameters, Dyson orbitals, etc.) including the transition dipole moments for the R-matrix states. The latter, together with some additional information, can be packaged as input data to run RMT calculations. Alternatively, the user can compute electronically elastic and inelastic integral cross sections for electron and positron scattering, as well as resonance energies and lifetimes for the former collision. Output from the suite, in the form of scattering data (K, S, or T-matrices), can be used, for example, to determine angular differential cross sections.

We strongly recommend reading this paper in conjunction with the paper on the UKRmol+ suite [1] where all programs mentioned below are described in detail.

Table 1
Input in `dirs.pl` indicating the paths where executables, basis sets, templates, and output are located.

parameter	description	format
`bin_in`	directory for UKRmol-in executables	string
`bin_out`	directory for UKRmol-out executables	string
`molpro`	directory for Molpro executable	string
`psi4`	directory for Psi4 executable	string
`molcas`	directory for Molcas executable	string
`basis`	directory where basis sets are	string
`templates`	directory where input templates are	string
`libs`	directory where libraries are	string
`output`	main directory for output in the working directory	string

3. The UKRmol-scripts

The UKRmol+ scripts consist of three Perl libraries (modules) placed in the directory `lib` and several Perl script files in the directory `scripts`. The module `dirfile.pm` contains general routines for dealing with files and directories, `ukrmollib.pm` specific routines for working with the UKRmol+ suite, and `MultiSpace.pm` routines for multiple active space (MAS) support. The driving script `main.pl`, which operatively prepares inputs, runs programs, and reads and collects their outputs, is called from the command line as

```
perl main.pl dirs.pl config.pl geometry.pl model.pl
```

The auxiliary scripts `dirs.pl`, ..., `model.pl` contain all necessary settings, options, and parameters in the form of Perl hash arrays for a specific calculation and are described in detail below. Moreover, the scripts use the Perl library `ForkManager.pm` [24] for a simple parallelization (over target geometries and/or symmetries) if requested.

To run the UKRmol+ scripts successfully, it is first necessary to modify the settings in `dirs.pl`. Here, paths and directories are specified where the executable files of the UKRmol+ suite and Molpro, Molcas, or Psi4 are installed and where a copy of the directories `input.templates` and `basis.sets`, which are a part of the UKRmol-scripts package, are placed. The `input.templates` directory provides templates used by the scripts to prepare the inputs for UKRmol+ programs where all strings of the form `>>>...<<<` are replaced with appropriate values determined from the parameters and available outputs. The `basis.sets` directory contains standard GTO basis files for both bound molecular and continuum orbitals. The latter are generated separately using other programs from the UKRmol+ suite which are not driven by the scripts. The path to the directory `lib` can also be specified either by setting the environment variable `PERL5LIB` or using -I when running the Perl command.

The other three scripts, `config.pl`, `geometry.pl`, and `model.pl`, are used to set up a specific calculation. In `config.pl`, the user can specify options regarding the overall run of the scripts (whether to run a scattering or photoionization calculation, what quantities to save, etc.). In the script `model.pl`, all *physical* parameters characterizing the target and scattering models, except the geometries of the molecule, are specified. One or several geometries of the molecule can be given or generated in various ways in `geometry.pl`. We now describe the content of these files in detail. Note that the order of parameters given in the tables does not necessarily correspond to the order in the scripts.

3.0.1. Nomenclature for photoionization/scattering calculation

The UKRmol+ codes and the scripts run calculations for electron-molecule scattering or photoionization. In each case, a slightly different convention is used to define the total number of electrons in the system in the published R-matrix literature, cf. [25]. In the case of electron-molecule collisions, the number of electrons of the target molecule is N so the total number of electrons is $N+1$. In photoionization, the total number of electrons is denoted as N and is referred to as the initial (usually neutral) state of the molecule. The "target" states in the photoionization case are then the final (usually positive ion) $(N-1)$-electron states of the ionized molecule. In this paper, we follow the convention for electron scattering. Regardless of the type of calculation, the scripts specify in `nelectrons` the number of electrons of the "target" molecule (i.e. without accounting for the electron that is/will be unbound), see below.

3.1. Directories input

Paths to the UKRmol+, Molpro, Molcas, and Psi4 executables and to a copy of the directories `input.templates` and `basis.sets` are specified in `dirs.pl`. Molpro, Molcas, or Psi4 are used to generate bound molecular orbitals, but it is possible to run the UKRmol-scripts using an existing Molden file if appropriately placed in the directory structure. Table 1 lists all the parameters in `dirs.pl`.

3.2. Geometry input

The molecular geometry is provided in the `geometry.pl` file together with some additional parameters. It is possible to run the scripts for a single geometry or for a range of geometries that are explicitly specified or generated in the script. Table 2 lists first the general geometry-related parameters set in `geometry.pl` and then those specifying a particular geometry. The latter parameters form one hash array for each given geometry and can be copied manually or generated automatically in the script to get an array of all geometries. The scripts provide subroutines that help the user to generate geometries by stretching particular bonds or along one or more coupled normal vibrational modes and also a subroutine to read geometries from one or several files in Molden format [26]. Examples of how to generate a range of geometries are provided with the UKRmol-scripts package in the directory `examples`.

Parameters `start_at_geometry` and `stop_at_geometry` are only used when calculations are run for more than one geometry. It can be useful, for example, if a particular run fails for some geometries and one, therefore, needs to restart it from a certain geometry, or to manually parallelize the execution on a computer cluster. To run all generated geometries set `stop_at_geometry` to zero.

Table 2
Geometry-related input in `geometry.pl`. Here and below, a_0 denotes the Bohr radius, the atomic unit of length.

parameter	description	format
General parameters		
`suffix`	added to output directory name to distinguish runs	string
`geometry_labels`	labels used on the first line of output files	string
`correct_cm`	translate geometry so the center of mass and the origin of coordinates coincide	1: yes, 0: no
`length_unit`	units for the coordinates	0: a_0, 1: Å
`start_at_geometry`	only run for geometries number $\geq n$	integer n
`stop_at_geometry`	only run for geometries number $\leq n$	integer n
Specific geometry parameters		
`description`	identifies each geometry in output files	string
`gnuplot_desc`	identifies each geometry in plot files	string
`atoms`	each atom is specified by its symbol and its Cartesian coordinates in a single array, e.g. as in `["H", 0.0, 0.0, 0.0]`	array of arrays

Table 3
Basic configuration parameters in `config.pl`.

parameter	description	format
`suffix`	name to distinguish different runs	string
`print_info`	where runtime messages are printed, possible values are *file*, *screen*, or *none*	string
`add_files_to_backup`	additional files to back up	array of strings
`molpro`	whether Molpro will be used	0: no, 1: yes
`psi4`	whether Psi4 will be used	0: no, 1: yes
`scattering`	run the whole calculation or only the target part	0: target only, 1: all
`photoionization`	calculate photoionization cross sections	0: no, 1: yes
`rmt_interface`	generate RMT input	0: no, 1: yes
`skip_radden`	skip using RADDEN to calculate radial densities	0: no, 1: yes
`skip_time_delay`	skip calculating time-delays	0: no, 1: yes
`gather_data`	gather data in a single directory	0: no, 1: yes
`clean`	remove intermediate files after run except integrals file	0: no, 1: yes
`remove_moints`	remove integrals file after run	0: no, 1: yes
`save_eigenph`	save eigenphase sums in named files	0: no, 1: yes
`save_xsec`	save cross sections in named files	0: no, 1: yes
`save_channels`	save channel data in named files	0: no, 1: yes
`save_rmat_amp`	save raw boundary amplitudes in named files	0: no, 1: yes
`save_Kmatrix`	save K-matrices in named files	0: no, 1: yes
`save_Tmatrix`	save T-matrices in named files	0: no, 1: yes
`keep_inputs`	keep input files	0: no, 1: yes
`keep_outputs`	keep output files	0: no, 1: yes
`bound`	run bound state calculation	0: no, 1: yes

3.3. *Configuration input*

The file `config.pl` enables the user to specify runtime options of the calculation. Table 3 lists the basic parameters of the run. The user should modify them according to which quantum chemistry codes are available, what type of calculations to perform (photoionization, RMT input, electron/positron scattering, or target structure only), what output data to save in named files, etc.

By default, all files given on the command line after `perl main.pl` are saved into directory `scripts_backup`; `add_files_to_backup` enables the user to choose additional files to be saved.

The scripts can generate molecular orbitals either using the Molpro or Psi4 packages. The user must set either `molpro`, or `psi4` to 1 and the other to 0. Similarly, `photoionization` and `rmt_interface` are also mutually exclusive. However, `scattering` needs to be set to 1 even when photoionization or RMT input calculations are run.

Computation of radial densities using RADDEN and time delays is by default turned off, i.e. parameters `skip_radden` and `skip_time_delay` are set to 1. Instead of RADDEN it is much more efficient to compute the radial densities directly as part of the integral calculation done

Table 4
Expert configuration parameters for parallel execution and evaluation of integrals in `config.pl`.

parameter	description	format
Options for parallel execution		
`parallel_geom`	number of geometries to run in parallel	integer
`parallel_symm`	number of symmetries to run in parallel	integer
`mpi_integrals`	MPI launcher command for `SCATCI_INTEGRALS`	string
`mpi_scatci`	`MPI-SCATCI` launcher command	string
`mpi_rsolve`	`MPI_RSOLVE` launcher command	string
Runtime options for `SCATCI_INTEGRALS`		
`buffer_size`	size of temporary arrays for integral transformation	integer
`delta_r1`	length in a_0 of elementary radial quadrature needed for evaluation of mixed BTO/GTO integrals	real
`transform_alg`	integral transformation algorithm, 0: auto, 1: sparse, > 1: dense	integer

Table 5
Other expert configuration parameters in `config.pl`.

parameter	description	format
`use_templates`	generate input files from templates; if not, rerun calculations with user-modified input files	0: no, 1: yes
`use_cdenprop`	force use of `CDENPROP` for target calculation	0: no, 1: yes
`ight` and `ighs`	diagonalization method for the target and all-electron calculation, respectively; −1: Arpack, 0: Davidson, 1: Givens-Householder, 2: automatic	integer
`use_saved_ramps`	use amplitudes and channel data saved from a previous run	0: no, 1: yes
`run_{eigenp, tmatrx, ixsecs, reson, time_delay}`	selection of which (scattering) outer region programs to run; all can be typically switched off for photoionization runs and RMT data production	0: no, 1: yes
`dipelm_smooth`	whether to smooth dipole moments, 0: raw data, 1: smoothed data, 2: both	integer
`only`	which subset of programs to run, list of programs given in one string separated by a pipe	string

by `SCATCI_INTEGRALS`. This is readily achieved by setting the variable `calc_radial_densities = .true.` in the input template file `scatci_integrals.inp`. The radial density for each orbital is then output to a separate file in the root directory for the calculation. This information is helpful to determine the appropriate R-matrix radius. When `gather_data` is set to 1, data like cross sections, eigenphase sums, etc. for different geometries are collected in files that are mainly placed in the `collected_scattering_data` directory. The subsequent parameters in Table 3 can be used to control the deletion of temporary files created during the run and the saving of various output files.

The last parameter in Table 3, `bound`, can be used to set up calculations when the energies of possible bound states of the total system are needed, without performing scattering calculations. In this case, all integrals are evaluated over the whole radial range, that is, avoiding cutting off the 'tails' beyond the R-matrix radius, as done for inner region R-matrix calculations.

In addition to the basic runtime options explained above, there are expert options in `config.pl`. First, in Table 4, the options for parallel execution and evaluation of integrals by the code `SCATCI_INTEGRALS` are listed. Second, in Table 5, other expert options are summarized.

The user can decide whether to run two or more geometries and/or symmetries in parallel and whether to execute parallel versions of some programs. In the case of `mpi_integrals`, `mpi_scatci` and `mpi_rsolve`, the MPI launcher command should be given explicitly, for example: `mpirun -np 32`. When these are left empty, `SCATCI_INTEGRALS` is run serially and the older, serial versions of SCATCI and RSOLVE are used.

When BTO or mixed BTO/GTO continuum basis sets are used, the parameter `transform_alg` should be set to 1 as using these basis sets results in large sparse matrices.

Expert options in Table 5 are provided for users who need to run calculations in a special setting or to run particular codes only. These options are briefly described in the table and more details can be found in `config.pl`. They should be used with care.

Table 6
Available point groups and order of their irreducible representations.

Point Group	Order of irreducible representations
D_{2h}	A_g, B_{3u}, B_{2u}, B_{1g}, B_{1u}, B_{2g}, B_{3g}, A_u
C_{2v}	A_1, B_1, B_2, A_2
C_{2h}	A_g, A_u, B_u, B_g
D_2	A, B_3, B_2, B_1
C_2	A, B
C_s	A', A'',
C_i	A_g, A_u
C_1	A

Table 7
General information about the target molecule in `model.pl`.

parameter	description	format
`molecule`	name of the molecule, e.g. its chemical formula, used for directory names etc.	string
`atoms`	all the atoms in the molecule given by their standard symbols in one array	array of strings
`nelectrons`	number of target electrons	integer
`symmetry`	point group of the molecule	string

3.4. Model input

Settings in `model.pl` describe the physical model used for a particular calculation and are the main input required from the user. Since there are many parameters in `model.pl` it is convenient to group them as follows:

- general information about the target molecule,
- units to be used for outputs,
- details of molecular orbitals and their spaces, target states, scattering model, and total space-spin symmetries,
- settings for multiple active space approach,
- details of the pseudo-continuum orbitals,
- details of the continuum basis set,
- settings for the outer region calculation.

In addition to the groups above, some other parameters can be set. Each group of parameters is explained in detail in this section except the settings for the multiple active space approach which are described separately in Section 7.1. Most of the parameters are required for all types of calculations (photoionization, electron scattering, positron scattering, or the generation of input for RMT) but some are only needed for some of them. When this is the case, it is stated explicitly.

Several of the parameters in `model.pl`, such as the number of orbitals or target states used, are entered as arrays of integers, where each integer corresponds to an irreducible representation of the point group of the target molecule, chosen by the user. The position of each number in the array indicates the irreducible representation (or simply the symmetry) with the order as shown in Table 6. Since UKRmol+ works only with the basic Abelian groups listed in the table, the maximum number of symmetries is 8. When changing the point group to one with fewer symmetries (irreducible representations), it is not necessary to delete redundant elements as the scripts use only those elements up to the actual number of irreducible representations. It is recommended to set these redundant elements to zero anyway to avoid confusion.

3.4.1. General information about the target molecule

Table 7 summarizes the basic information regarding the molecule that should be given in `model.pl`. We note that `atoms` should include all the atoms in the molecule, so for example, for H_2O this will be:

```
'atoms',        ["O", "H", "H"],
```

The number of electrons of the target set in `nelectrons` can differ from the number for the neutral molecule. The charge of the target is then determined by the scripts from this parameter and `atoms`. The point groups that can be specified in `symmetry` are listed in Table 6. Note that the scripts do not check the consistency of this option with the geometry settings in `geometry.pl` so the user should ensure that they are consistent.

3.4.2. Units to be used for outputs

Units for the output produced in the calculations, that are set in `model.pl`, are summarized in Table 8.

3.4.3. Details of molecular orbitals and their spaces, target states, scattering model, and total space-spin symmetries

The parameters that define the type of molecular orbitals to be generated and how these are going to be split into the different frozen, active, and virtual spaces determine a large part of the physics that will be modelled in the calculation. Table 9 summarizes them. Note that the term *frozen* is used for both closed orbitals in SCF calculations and frozen orbitals in CI calculations in `model.pl` and thus the number of these orbitals is the same in basic models. More flexible models can be set up using the MAS approach, see Section 7.1.

Table 8

Units to be used for outputs in `model.pl`.

parameter	description	format
`r_unit`	units of length 0: a_0, 1: Å	integer
`e_unit`	energy units 0: E_h, 1: Ry, 2: eV	integer
`x_unit`	units for the cross sections 0: a_0^2, 1: Å^2	integer

Table 9

Details of the molecular orbital construction, orbital spaces, scattering model, and states in close-coupling expansion in `model.pl`.

parameter	description	format
`basis`	name of the basis set	string
`orbitals`	which type of orbitals are generated *HF*: Hartree-Fock, *natural*: CASSCF or MASSCF	string
`charge_of`	use orbitals of *target* or all-electron system (*scattering*), only used for photoionization or RMT	string
`select_orb_by`	which ordering of orbitals to use *molden* or *energy*	string
`ncasscf_states`	number of target states of each spin and space symmetry to be averaged if a SA-CASSCF calculation is run	hash array of integer arrays
`model`	scattering model *SE*, *SEP*, *CHF-A*, *CHF-B*, *CAS-A*, *CAS-B*, or *CAS-C*	string
`nfrozen`	number of frozen (closed) target orbitals	integer
`nactive`	number of active target orbitals	integer
`nvirtual`	number of virtual orbitals used to build L^2 functions	integer
`nreference`	number of reference orbitals	integer
`frozen_orbs`	frozen orbitals for each symmetry	integer array
`active_orbs`	active orbitals for each symmetry	integer array
`virtual_orbs`	virtual orbitals for each symmetry	integer array
`reference_orbs`	reference orbitals for each symmetry	integer array
`ntarget_states`	number of target states of each total spin and symmetry for which energy and permanent and transition moments are calculated	hash array of integer arrays
`ntarget_states_used`	number of target states included in the close-coupling expansion	integer
`scattering_states`	which total symmetries to run the scattering calculation for	hash array of integer arrays

The basis sets that can currently be indicated in `basis` and used are those provided with the package (in the `basis.sets` directory). The user can create the file for other standard (or tailor-made) basis sets.

The parameter `charge_of` is only used for photoionization or RMT input calculations and indicates whether orbitals for the target (usually a cation) or the total system are to be used.

As the scripts always use the first *n* orbitals (for all symmetries or each symmetry) requested by the user it is important to select an appropriate ordering using `select_orb_by`. When this is set to *molden*, the orbitals are used in the order in which they are saved in the Molden file. If set to *energy*, then the orbitals are used in energy order. This parameter is irrelevant if Hartree-Fock orbitals, which are always ordered by energy, are used.

The UKRmol-scripts will run either Molpro, Psi4, or Molcas to generate bound molecular orbitals, either using the Hartree-Fock (HF) method, the state-averaged complete active space self-consistent field (SA-CASSCF) approach, or the state-averaged multiple active space self-consistent field (SA-MASSCF) approach (which comes in two variants SA-ORMASSCF and SA-GASSCF). For details on how to set up the last approach, which is only available when using Molpro or Molcas, see Section 7.1. Table 10 summarises which quantum chemistry methods can be used with each quantum chemistry code.

When the SA-CASSCF approach is chosen, the frozen and active spaces as well as the states to be averaged, need to be selected. The latter is done using `ncasscf_states` to indicate how many states of each space-spin symmetry are averaged; note that currently the scripts only allow for state-averaging with equal weights for all the states. The frozen and active spaces will also be used to generate the target and total (all-electron) wave functions in the UKRmol+ suite, with the addition of some virtual orbitals.

The parameter `model` defines how the total wavefunctions are going to be built. The scattering models correspond to the standard approximation levels in low-energy electron and positron scattering [3,1]: static or static-exchange (SE), for positrons and electrons respectively, static plus polarization or static-exchange plus polarization (SEP), or close-coupling (CAS) where the choice of A, B or C leads to different types of L^2 functions. Additional models CHF-A and CHF-B are described in Sec. 7.

Table 10
Quantum chemistry methods available in the scripts; 'irrep' stands for irreducible representation.

Method	Molpro 2021.2	Psi4 1.7	Molcas 8.4
HF	✓	✓	✓
CASSCF	✓	✓	✓
ORMASSCF	✓		
GASSCF			✓
State Averaging: Single irrep	✓	✓	✓
State Averaging: Multiple irreps	✓		
State Averaging: Multiple spins	✓*		

*Not when using ORMASSCF.

Table 11
Details of the pseudo-continuum orbitals in `model.pl`.

parameter	description	format
`use_PCO`	switch to include PCO basis	0: no; 1: yes
`reduce_PCO_CAS`	selects which configurations to include	0: no; 1: yes
`maxl_PCO`	highest partial wave used in the PCO GTO basis	integer
`PCO_alpha0`	α_0 for PCO generation	real array
`PCO_beta`	β for PCO generation	real array
`num_PCOs`	number N_e of PCO generated	integer array
`PCO_gto_thrs`	threshold for allowed continuum GTO exponents	real array
`PCO_delthres`	deletion threshold for orthogonalization	real array

The number of frozen, active, and virtual orbitals to be used is set by `nfrozen`, `nactive`, and `nvirtual` parameters. By default, they will be selected in the order indicated by `select_orb_by`. However, it is possible to request a specific number of orbitals of each symmetry by using `frozen_orbs`, `active_orbs`, and `virtual_orbs`. The total number of chosen orbitals of each type must be equal to `nfrozen`, `nactive`, and `nvirtual`.

For SE calculations, no excitations of the target electron to virtual orbitals are included. For SEP calculations, the L^2 functions only involve single excitations from the active space into the virtual space. For close-coupling calculations, the active orbitals are used in the generation of the configurations for the target and the total wave functions. The virtual orbitals, in all cases, are only used to build the L^2 functions.

Parameters `nreference` and `reference_orbs` do not impact the physics described. They are required as input to program `CONGEN` that builds the configuration state functions for target and total wave functions relative to a reference determinant. Providing `nreference` and `reference_orbs` is particularly useful in the case of electron-rich targets and, when possible, it is suggested that all orbitals occupied in the ground state of the target plus a few additional orbitals of each symmetry are used.

For close-coupling calculations, one must set `ntarget_states_used` to the number of target states to be included in the close-coupling expansion. The program will select the `ntarget_states_used` lowest states out of the total number of states for which target properties are evaluated. Using `ntarget_states` the user can request that these properties are calculated for a specific number of states of each possible space-spin symmetry. The spin needs to be stated explicitly (so, for a target with an even number of electrons, 'singlet' and 'triplet' are usually specified) and then the number of states of each symmetry.

It is usually helpful to set `ntarget_states` such that the total number of states for which properties are calculated is larger than `ntarget_states_used`. This enables the user to check and ensure that no states are skipped in the close-coupling expansion as this could lead to non-physical features in the evaluated physical quantities [3].

In UKRmol+, once the integrals are evaluated, the inner region calculations are run separately for each space-spin symmetry of the total system. Sometimes (for example, if a specific resonance is to be investigated), it is desirable to run calculations for a reduced number of space-spin symmetries and `scattering_states` can be used for this purpose. As with `ntarget_states`, the spin needs to be stated explicitly (so, for a total system with an odd number of electrons, 'doublet' and 'quartet' are usually specified) and then a zero or one to indicate if that space-spin symmetry is required.

3.4.4. Details of the pseudo-continuum orbitals

The R-matrix with pseudostates method (RMPS) [27] is a generalization of the close-coupling method that uses an extra set of target orbitals, known as pseudo-continuum orbitals (PCO), to provide a representation of the discretized continuum in the inner region. Table 11 lists the parameters in `model.pl` associated with the use of PCOs and pseudostates.

The molecular implementation of this method uses even-tempered GTOs [28] where the exponents are determined by:

$$\alpha_i = \alpha_0 \times \beta^{i-1}, \quad i = 1, \ldots, N_e. \tag{1}$$

Values of α_0, β, and N_e can differ for each partial wave, thus parameters `PCO_alpha0`, `PCO_beta`, and `num_PCOs` are actually arrays of values. If these orbitals are not required, set `use_PCO` to zero and there is no need to modify any of the other parameters.

There is no restriction in principle on the maximum partial wave to be included. However, due to the computational cost of this additional basis set, this is normally kept as low as or lower than the maximum partial wave for the continuum orbitals. To avoid severe linear dependence,

Table 12
Parameters defining continuum orbitals in `model.pl`.

parameter	description	format
`use_GTO`	switch to use GTOs	0: no; 1: yes
`radius_GTO`	which GTO continuum basis to use	integer
`maxl_GTO`	highest partial wave for GTOs	integer
`use_BTO`	switch to use BTOs	0: no; 1: yes
`start_BTO`	radius where the BTOs start	integer
`rmatrix_radius`	R-matrix radius, where the BTOs end	real
`order_BTO`	order of the BTOs	integer
`no_of_BTO`	number of BTOs	integer
`maxl_BTO`	highest partial wave used for BTOs	integer
`delthres`	deletion thresholds for the continuum orthogonalization, one per symmetry	real array
`maxl_legendre_1el`	maximum L in Legendre expansion for nuclear attraction integrals	integer
`maxl_legendre_2el`	maximum L in Legendre expansion for 2-electron integrals	integer

`PCO_gto_thrs` ensures that the exponents of continuum and PCO GTOs are not too similar: continuum basis functions with exponents equal or bigger than `PCO_gto_thrs` are removed from the basis. If set to -1, then a default value of $\texttt{PCO\_alpha0} \times (\texttt{PCO\_beta} - 1)$ is used as threshold.

When PCOs are used, these are orthogonalized to the target bound orbitals first and then the continuum orbitals are orthogonalized to the set of target orbitals + PCOs. The parameter `PCO_delthres` contains deletion thresholds for each symmetry that fulfills the same role as `delthresh` for the continuum (see below and Table 12).

When PCOs are included in the calculation, target and total configurations involving single (and in the latter also double) occupancy of the PCOs are used. For models where the target wave functions are generated using an active space, the configurations that include PCOs can be built in two ways, controlled by the parameter `reduce_PCO_CAS`. When `reduce_PCO_CAS` equals 1, the configurations involving PCOs are generated by promoting an electron from an orbital occupied in the ground state (Hartree-Fock) configuration to a PCO. Alternatively, PCO configurations are generated by promoting an electron from the active space into a PCO. The latter often leads to a significantly larger number of configurations. For this reason, it is usual to set `reduce_PCO_CAS` to 1.

3.4.5. *Details of the continuum basis set*

The continuum orbitals can be expanded in a set of GTOs with predetermined exponents (these are read from the `basis.sets` directory), a set of BTOs that are defined via input parameters, or a mixed GTO-BTO basis set. All relevant parameters are listed in Table 12.

The parameter `radius_GTO` determines which continuum GTO basis will be used. Continuum GTO bases are provided in the `basis.sets` directory for the following radii: 4, 6, 10, 13, 15, and 18 a_0 for neutral targets and 10 and 13 a_0 for positively charged ones. Since the exponents of the continuum GTO basis must be optimized for a specific radius for each partial wave, the maximum angular momentum available for each combination of radius and charge state may differ. If a different value is set for `radius_GTO`, the user will need to generate the continuum GTO basis set using GTOBAS [29], a program from the UKRmol+ suite.

If no BTOs are included in the calculation, `rmatrix_radius` must be set equal to `radius_GTO`. If a mixed GTO-BTO basis is used, the parameter `start_BTO` must be chosen to be less than `radius_GTO` and a standard choice is $\texttt{start\_BTO} < \texttt{radius\_GTO} - 1\ a_0$.

Parameters `maxl_legendre_1el` and `maxl_legendre_2el` control the numerical evaluation of the nuclear one- and two-electron integrals involving BTOs in `SCATCI_INTEGRALS`; for details, see [1].

Since UKRmol+ works with orthogonal orbitals, the continuum basis sets must be orthogonalized both to the target orbitals that will be used and among themselves. The latter is done using symmetric orthogonalization and this requires setting a threshold [1]: continuum orbitals with eigenvalues of the overlap matrix smaller than this threshold, given by `delthres`, are deleted. Different thresholds can be given for different orbital symmetries, although usually the same value is used for all.

3.4.6. *Settings for the outer region calculation*

The outer region calculation will produce one or more photoionization or scattering quantities (cross sections, asymmetry parameters, K-matrices, etc.) for a grid of kinetic energies of the unbound particle. The grid is defined as one or more energy subranges j containing N_j energies given by:

$$E_i = E^j_{\text{inc}}(1) + E^j_{\text{inc}}(2) \times i \text{ for } i = 1, ..., N_j \tag{2}$$

where the starting energy of the subrange $E^j_{\text{inc}}(1)$, increment of energy $E^j_{\text{inc}}(2)$ and N_j are set in `model.pl` (see Table 13).

In the case of scattering calculations for neutral targets, the propagation radius `raf` is usually set to values larger than 50 a_0. The larger the value, the longer the time the outer region calculations take. However, `raf` of 100 a_0 can be necessary, for example, when channels are energetically close.

By default, the target calculation will produce dipole and quadrupole permanent and transition moments for all target states considered. `max_multipole` enables the user to decide whether any of these will be used to model the interaction of the target and the unbound particle in the outer region. In scattering calculations, it is customary to set it to 2. If set to zero, this switches off the interaction in the outer region. For photoionization calculations, the outer region calculation is set up differently, see Section 5.

Table 13
Parameters for the outer region calculation in `model.pl`.

parameter	description	format
`raf`	propagation radius	real
`max_multipole`	maximum multipole retained in expansion of long-range potential	integer ≤2
`nescat`	number of energies N_j in each subrange, string of integers separated by commas	integer array
`einc`	initial energy $E^j_{\text{inc}}(1)$ and energy increment $E^j_{\text{inc}}(2)$ for each subrange, pairs of real numbers separated by commas	real array
`maxi`	highest initial state for which cross sections, etc. are calculated	integer
`maxf`	highest final state for which cross sections, etc. are calculated	integer

3.5. General description of output

The output of one run of the scripts varies according to settings. In general, the scripts will create a directory for a specific run (either named automatically according to the model parameters and the `suffix` given in `geometry.pl` and `config.pl`, or specified explicitly by the user in `model.pl` using `directory`). The following subdirectories are created during the run:

`scripts_backup`: subdirectory where all the scripts (`.pl` files) used for the particular run are copied for later reference,
`logs`: subdirectory where all log files are stored; a `main.log` file and one `geom<n>.log` file for each geometry are created,
`geom<n>`: subdirectory where all input and output files of each program in the UKRmol+ suite for each geometry *n* are stored,
`collected_scattering_data`: subdirectory where the files with generated scattering data, such as eigenphase sums, T-matrices, and cross sections, are copied from subdirectories `geom<n>` for simpler handling and plotting.

In addition, the following files are created, usefully collecting data for several geometries:

`geometries` and `geometries.molden`: text files containing descriptions of all used molecular geometries, a simple sequence of numbers *n* is assigned to the geometries, which are used for naming of output files and subdirectories for each specific geometry,
`target.energies`: text file with energies of the target electronic states determined in the calculations for all geometries,
`target.dipole.moments`: text file with the permanent dipole moment of the ground target electronic state for all geometries,
`Rmatrix.energies`: text file with several of the lowest-energy R-matrix poles for all geometries,
`resonance.positions.and.widths`: text file with resonance positions and widths for all geometries for those resonances identified and characterised by the program `RESON` [30].

The scripts also create a few files (with the extension `.gp`) for easier plotting of target energies, eigenphase sums, and cross sections using the Gnuplot software. These files are useful to compare results for several geometries.

4. Running scattering calculations

The workflow for scattering calculations is shown in Figs. 4 and 5 of the UKRmol+ paper [1].

4.1. Input

For all types of calculations, the user should choose the geometry or geometries to be run. The ground state configuration of the target is also required. When the calculations include several target states, several tests to determine the most appropriate basis set, active space, and/or state averaging are usually needed. For this step, it is useful to set `scattering` = 0 to reduce the computational effort. Once the best model is chosen, setting `scattering` = 1 ensures that all requested programs are executed.

The lepton projectile is specified via the `positron_flag` in `model.pl`: it should be set to 0 for electron scattering and 1 for positron scattering. If not specified, the default value is 0 (electron scattering). In positron scattering, excitation of triplet target states is not allowed due to selection rules. Therefore, triplet `ncasscf_states` and `ntarget_states` should all be equal to 0 for positron scattering.

The default outer region programs run by the scripts are `SW_INTERF` and `RSOLVE` so that K-matrices are generated. The user should then select, in the `config.pl` file (see Table 5), which other programs to execute. This will determine what output is produced.

4.2. Outputs generated

The same output files are generated for electron and positron scattering, except `RESON` is not usually run for the latter so no resonance information is produced. In the case of positrons, the directory name will include “e+” (instead of “e-”).

In addition to the output file for each program in the UKRmol+ suite, a scattering calculation will produce several output files:

- In each `geom<n>` directory, files containing the contribution to the cross sections from each irreducible representation, as well as the eigenphase sum for each of these geometries.

- The same data will also be collected in the `collected_scattering_data` directory in an easy-to-plot format.
- If selected in the `config.pl` script, the K and/or T-matrices will be saved into `collected_scattering_data`.
- If resonances are identified and characterized using `RESON`, a file named `resonance.positions.and.widths` with the energy and width of the resonances detected for all geometries is generated.

4.3. Electron scattering example

We provide an example of a close-coupling calculation for H_2O. The calculation is run for four different geometries, generated automatically by stretching the OH bond lengths symmetrically. It uses the C_s point group.

The calculation employs the cc-pVTZ basis set and CASSCF orbitals generated by Molpro for the ground state. These orbitals have been saved in files named `h2o.molden`, one for each symmetry, to enable the user to run a UKRmol+ calculation without the need for Molpro to demonstrate the use of the `only` option of the scripts (see Table 5).

The $1a'$ orbital is kept frozen, the active space comprises five a' and one a'' orbitals. Five target states are included in the scattering calculation. The continuum is of GTO type only and includes partial waves up to $l = 4$. The R-matrix radius used is 10 a_0 and the propagation is carried out to 70 a_0.

The calculation is set up to run serially and to save cross sections and eigenphase sums only, collected in the `collected_scattering_data` directory. The following outer region programs are run: `EIGENP`, `TMATRX`, `IXSECS`, and `RESON`. Resonance information is collected in the usual file.

4.4. Positron scattering example

Two examples are provided for positron scattering calculations. The simpler one uses a similar model to that used for electron scattering from H_2O, but it is set to run for just one geometry. Again, the Molden file with the orbitals is provided.

The second example shows the use of pseudocontinuum orbitals in a positron calculation. In this case, the target is H_2 and the D_{2h} point group is used. No Molden file is provided for this example, so a quantum chemistry program is needed to run this test. The 6-311G** basis set is used to generate CASSCF orbitals for the ground state. The active space contains 9 orbitals of the following symmetry: three a_g, one b_{3u}, one b_{2u}, two b_{1u}, one b_{2g} and one b_{3g}. In addition, PCOs are included in the calculation: these are generated using $\alpha_0 = 0.17$ and $\beta = 1.4$ for $l = 0, 1, 2$.

15 singlet target states are included in the close-coupling scattering calculation, corresponding to the ground state and single excitations into the PCOs. The R-matrix radius is set to 13 a_0 and a continuum GTO basis with partial waves up to $l = 4$ is employed; again, the propagation radius is set to 70 a_0.

5. Running photoionization calculations

For calculations of (perturbative) single-photon ionization, the scripts implement the workflow depicted in Fig. 6 of the UKRmol+ paper [1]. The calculation formally proceeds as if for the case of electron-molecule scattering where the "target" molecule is represented by the final states of the photoionized system (typically a positive ion). In both cases, a continuum wavefunction is generated but for different asymptotic boundary conditions (outgoing wave in the case of scattering and incoming wave in the case of photoionization).

In addition to the calculation of the continuum wave functions, the photoionization run produces dipole matrix elements between the R-matrix basis states (the eigenfunctions of the Hamiltonian for the whole system in the inner region) which are used to assemble the physical photoionization dipole matrix elements. This is done by the `CDENPROP` program [25].

5.1. Input

The input parameters required to specify the molecular model are identical to the scattering calculation. In the usual case of photoionization of neutral molecules, the photoelectron moves, asymptotically, in the field of the charged molecule. As a result, some physical parameters of the calculation are chosen differently:

- `max_multipole` should be set to 1 since the dominant interaction in the outer region is Coulomb.
- For the same reason, the propagation radius `raf` should be set to the R-matrix radius (i.e. no propagation is performed). This choice has practical reasons too: in photoionization, backpropagation of the wavefunction from the asymptotic to the R-matrix radius would have to be performed which is subject to numerical instabilities in the case of closed channels. If long-range interactions in the outer region are important these should be effectively included by increasing the size of the R-matrix radius instead.
- In addition, the scripts automatically ensure that the nuclear contributions to the molecular (bound-bound) multipole transition moments are not included if `photoionization` is set to 1 (this sets the `isw` option of the `DENPROP` program). The nuclear contribution to the dipole matrix elements must be excluded since the bound-bound dipole transition moments are needed in the `CDENPROP` program to compute the light-driven dipole transitions between the R-matrix basis states. As a result, the multipole matrix elements generated by `DENPROP` from the photoionization run cannot be used in the outer region to represent the long-range electron-molecule interaction. For that reason, it is important to switch off the R-matrix propagation in the outer region.
- The symmetry of the initial molecular state is specified in `initialsym`. By default, the initial state in the photoionization dipole matrix element is taken as the lowest R-matrix basis state of that symmetry.
- `first_Ip`, in units of `e_unit`, can be used to specify the accurate first ionization potential (IP) of the molecule. This value is used only by the `DIPELM` program to shift the first IP to the accurate value when computing the photoionization observables. If this value is set to zero then the calculated IP is used.
- `dipelm_smooth` controls whether the smoothed photoionization observables should be produced by smoothing the partial wave dipole matrix elements. Typically, the photoionization of neutral molecules leads to a congested spectrum of sharp autoionization states. Consequently, smoothing of the observables may produce artifacts in the observables, particularly close to the threshold. For that reason, it is advised to always produce unsmoothed (unbiased) results too. Set `dipelm_smooth` to 0 for unsmoothed data, 1 for smoothed data, or 2 for both.

In photoionization, depending on the symmetry of the molecule, dipole transitions from the initial state, which has a specific symmetry, may only couple to a subset of the irreducible representations representing the continuum. In this case, only a subset of the `scattering_states` has to be generated, see Table 9.

5.2. Outputs generated

The output of the photoionization calculation consists of the standard orientation-averaged photoionization cross sections and angular distributions (β parameters) saved in the files with self-explanatory names starting with “`photo_`”. In addition, the partial-wave dipole matrix elements are saved on the files `pwdips-x`, `pwdips-y` and `pwdips-z`. These files are placed in the `geom<n>` directories.

5.3. Example

The example provided runs a close-coupling photoionization calculation for CH_4. A small active space of 8 electrons distributed in 6 orbitals is used to describe the ionic states of the molecule. The calculation is performed in the reduced D_2 point group. It includes four final states of the ion in the close-coupling expansion and a small Gaussian continuum with partial waves up to $l = 4$. The scattering model uses a small set of additional L^2 functions corresponding to single excitations out of the active space to a single virtual orbital. In addition, the calculation is performed using a set of CASSCF molecular orbitals pre-generated in Molpro (saved in the file `ch4+.molden`). The reference photoionization cross sections and angular distributions are provided too. Section 8 shows some of the output for this calculation.

6. Generating input for RMT

For simulations of laser-driven electronic processes in molecules, the RMT package [4] requires a single binary file `molecular_data` produced by the program `RMT_INTERFACE`. The associated workflow is shown in Fig. 7 in [1]. The most sizeable part of this file is the transition dipole moments between all pairs of R-matrix basis states. Further data, including a detailed description of outer-region channels, boundary amplitudes, and various useful coupling coefficients is also included.

Molecular RMT requires boundary amplitudes evaluated not only at the boundary between the inner and outer region but also at several equally spaced radii inside the R-matrix sphere. The number of these extra radii and their spacing are set to a reasonable default directly in the UKRmol-scripts input templates (18 and $0.08\,a_0$ respectively). The chosen values correspond to the default setup of RMT. The defined overlap region of $18 \times 0.08\,a_0 = 1.44\,a_0$ needs to be numerically free of the target orbitals, which may require the use of a larger R-matrix radius than the corresponding photoionization run. Note that there is no keyword in the scripts to change these values; if needed, they can be changed directly in the templates.

Note that the default workflow that uses the program `RMT_INTERFACE` has an internal limit on the molecular model size. If this is exceeded, the program will fail. When such a large model is needed, it is possible to use the `MPI-SCATCI`-based alternative UKRmol+ workflow (see Fig. 8 in [1]), which does not suffer from such limitations. This alternative mode is enabled by setting `parallel_symm` to “0” in `config.pl`.

6.1. Input

To generate the RMT file for a particular molecular model, one sets the options `scattering` and `rmt_interface` in `config.pl` to “1”; the option `photoionization` should be set to “0”. Otherwise, the remaining setup should correspond to that of a photoionization calculation. The only difference with respect to a typical photoionization calculation is the number of “scattering” symmetries: in one-photon ionization, sometimes only a subset of irreducible representations is required due to symmetry reasons. However, multi-photon or strong-field processes with arbitrary polarizations couple wave functions of all irreducible representations. As a consequence, the RMT interface mode requires setting all spin-accessible symmetries in `scattering_states` in `model.pl` to “1”.

6.2. Outputs generated

Even though the output of the scripts in this mode is similar to the photoionization run, the only relevant output is the file `molecular_data` present in the geometry folder.

6.3. Example

We provide a simple example to generate input data for RMT for a static-exchange calculation for H_2 molecule with a small Gaussian-only continuum basis. The calculation is performed in D_{2h} point group symmetry and uses a set of Hartree-Fock molecular orbitals pre-generated, this time, by Psi4. Dipole transition matrices between the singlet R-matrix states of all irreducible representations are generated to be used in the inner-region part of the RMT calculation and saved on the final output file `molecular_data` along with all other auxiliary quantities as described above.

7. Description of selected models

The standard SE, SEP, and close-coupling models, as well as the use of pseudocontinuum orbitals and pseudostates have been described in detail elsewhere, in particular in [1]. Here, we describe two newer approaches to modelling electron correlation and generating the configuration state functions for the target and all-electron systems.

Table 14
Using the MAS approach in `model.pl`.

parameter	description	format
`use_MASSCF`	use MASSCF for quantum chemistry 0: don't use MASSCF 1: ORMASSCF in Molpro 2: GASSCF in Molcas	integer
`use_MAS`	use MAS for target and scattering 0: don't use MAS 1: use ORMAS 2: use GAS	integer
`qchem_MAS`	allows for a different MAS for the quantum chemistry of the target and total calculations	array
`qchem_constraints`	additional constraints on allowed CSF	function
`MAS`	define the target active spaces	array
`constraints`	additional constraints on allowed CSF	function
`l2_MAS`	define the L^2 active spaces if different from an extra electron in `MAS`	array
`l2_constraints`	additional constraints on allowed CSF	function

7.1. Multiple active space (MAS) approach

The multiple active space approach (MAS) is a new method for defining the orbital active space for UKRmol+ calculations [31]. It is based on a generalisation of the occupation restricted multiple active space approach (ORMAS) [32] and the generalised active space approach (GAS) [33]. The input required to set this model is summarized in Table 14.

ORMAS and GAS were developed as techniques for excluding unnecessary CSF from the configuration space generated by the standard complete active space (CAS) approach. They work by dividing the active space into multiple subspaces and defining a maximum and minimum electron occupation for each subspace. ORMAS defines a local minimum and maximum, while GAS defines a cumulative minimum and maximum. It is often possible to reduce the number of CSF by an order of magnitude or more using the MAS approach while retaining chemical accuracy, both in the bound states [32,33] and in the scattering or photoionization observables [31].

A second benefit of the MAS approach is that it provides a simple yet flexible method of defining active spaces in general, allowing for sophisticated models beyond the predefined models available in the scripts.

MAS can be used both for generating the orbitals in the initial quantum chemistry calculation (ORMASSCF in Molpro and GASSCF in Molcas, we will use MASSCF to refer to both) and for specifying the active space in the following target and total wave function calculations.

A MAS subspace is defined by a triplet of elements: First is an array specifying the orbitals per symmetry in the subspace, or an integer specifying the total number of orbitals in the subspace. If the second method is used then the orbitals are picked in order, as defined by `select_orb_by`, and the orbitals per symmetry are populated automatically. Second is an array giving the minimum and maximum occupancy. Thirdly, a string describing the subspace, `"frozen"/"closed"/"active"`, this string is optional for target and total calculations but required when doing a MASSCF quantum chemistry calculation. For example, using the ORMAS version of the MAS approach, a 10-electron molecule with 2 electrons in a closed orbital, and single, double, and triple excitations out of a 6-orbital subspace into a 5-orbital subspace would be written as

```
$model{MAS} = [
    1,[2,2],"closed", 6,[5,8],"active", 5,[0,3],"active"
]
```

In the GAS version, this would be

```
$model{MAS} = [
    1,[2,2],"closed", 6,[7,10],"active", 5,[10,10],"active"
]
```

The parameter `constraints` allows for further constraints to be applied to the subspaces by passing an anonymous function that removes unwanted distributions of electrons across the subspaces. For example, using ORMAS, the following model,

```
$model{MAS} = [
    5,[6,10],"active", 2,[0,4],"active", 5,[0,1],"active"
]
$model{constraints} =
    sub {my $dist=shift; !($dist->[2] > 0 && $dist->[0] != 9);}
```

allows single excitations to the third subspace only when there are 9 electrons in the first subspace, if HF orbitals are used then this is equivalent to CAS(10,7) + single excitations from the HF configuration to 5 higher-lying orbitals. Note that when using constraints we need to set the L^2 space by hand, as the scripts are not clever enough to derive `l2_constraints` from `constraints` on their own.

If MAS is used for the target and total run but MASSCF is not used for the quantum chemistry calculation then all the variables that are usually set to define the active space for the quantum chemistry calculation still need to be set, i.e. `nfrozen`, `nactive`, `nvirtual`, and possibly `frozen_orbs`, `active_orbs` and `virtual_orbs`.

If MASSCF is used, then only `nvirtual` and possibly `virtual_orbs` need to be set as the rest are determined by the quantum chemistry MAS. In all cases `nreference`, and possibly `reference_orbs` still need to be set.

The MAS for the target and total calculations does not need to respect the type of active spaces defined in the quantum chemistry calculation. E.g. orbitals that are frozen or virtual in the quantum chemistry calculation can be specified as active in the target and total calculation. However, the total number of orbitals in the target calculation cannot exceed those defined in the quantum chemistry calculation. One thing to note is that any virtual orbitals retained from the quantum chemistry calculation that are not explicitly used in the MAS for the target and total calculation are automatically contracted with the continuum orbitals. This means that they are formally included as part of the 1-electron continuum orbital basis.

Several examples of `model.pl` files setting different MAS calculations are available in the directory `examples`.

7.2. Polarization-consistent coupled Hartree–Fock approach

The polarization-consistent coupled Hartree–Fock (PC-CHF) approach uses a simple Hartree–Fock-like description of the target states to model polarization and multi-channel effects in polyatomic molecular targets. The model is constructed in a self-consistent manner meaning that all of the target states implied by the polarization configurations are included in the total wavefunction. This model has been used recently [34] to model photoionization of three different molecules: H_2O, N_2O and formic acid.

PC-CHF is similar to the SEP model in that it retains a simpler HF-like description of the target states, but it couples polarization states to their implied ionic states in order to remove spurious resonances that are observed in SEP calculations. For more details on the model and how states are generated see Section 2.1 in [34].

The PC-CHF model is selected using the appropriate keyword in the `model` parameter of the `model.pl` file and by specifying HF orbitals in the parameter `orbitals`. Two variants of the PC-CHF model are available: *CHF-A* and *CHF-B*. Setting `model` to *CHF-A* is recommended for most use-cases. In this model, each HF-like target configuration corresponds to a single target state in the close-coupling expansion. Whereas, in *CHF-B* the target configurations of the same space-spin symmetry are effectively producing CI expansions of the resulting target states. The latter sub-model was designed to enable PC-CHF calculations for molecules with degenerate target molecular orbitals of different symmetries that lead to degenerate target HF-like states, as in the case of N_2O. In this situation, the degenerate HF-like CSFs from the same space-spin symmetry must be combined appropriately to produce a set of orthogonal wave functions. This is achieved most readily, without any changes to the UKRmol+ codes, by contracting all these configurations into a single CI expansion and proceeding as in the case of a CAS calculation.

8. A more detailed example output

To illustrate the output generated by the use of the UKRmol-scripts we show some of the results for the photoionization example described in Section 5. The cross section for one-photon ionization into the ground state of CH_4^+ obtained with the example input for CH_4 is plotted in Fig. 1. This cross section is saved in the file `photo_xsec` where each column, except for the first one that is the photon energy, corresponds to a specific final cation state. Note that in this case, since the ground state of the cation is triply degenerate, the cross section for photoionization into this state is obtained by adding the cross sections for the lowest 3 states in D_2 symmetry. The energies of all cation states included in the calculation are printed in file `target.energies`:

```
# Target energies / a.u.
#          doublet.A.1        doublet.B3.1        doublet.B2.1        doublet.B1.1
         -39.409615166       -39.770863568       -39.770863568       -39.770862392
```

One can see from the figure that this relatively simple model provides a reasonable description of the process. A more accurate model of one-photon ionization including its extension to two-photon above-threshold ionization and calculation of attosecond time-delays has been published recently [35].

A summarized example of the output in the `pwdips-z` file can be seen below. The vertical ellipsis dots indicate that some lines of the output are omitted. The first part of the file provides information on the cation and neutral state as well as on the channel properties; the second part lists the components of the partial wave dipole moments for each channel.

```
⋮

  Target GS:   -0.39770864D+02
  Bound state     Energy
        1      -0.40290469D+02

  Channel    Targ. 2S+1  I.Rep. g/u     l     m     Channel Thresholds
  --------------------------------------------------------------------
        1       1     2     1     0     1    -1      0.00000000D+00
        2       1     2     1     0     2     1      0.00000000D+00
        3       1     2     1     0     3    -3      0.00000000D+00
        4       1     2     1     0     3    -1      0.00000000D+00

⋮

       23       4     2     0     0     3     2      0.72249681D+00
       24       4     2     0     0     4    -4      0.72249681D+00
       25       4     2     0     0     4    -2      0.72249681D+00

  Channel     Electron Energy       real Y              imag Y              real Z               imag Z              real X              imag X
  ---------------------------------------------------------------------------------------------------------------------------------------------------
        1     0.18375000D-02      0.00000000D+00      0.00000000D+00      -0.20969161D+00     -0.47697360D-01      0.00000000D+00      0.00000000D+00
        1     0.36750000D-02      0.00000000D+00      0.00000000D+00      -0.20845583D+00     -0.48217373D-01      0.00000000D+00      0.00000000D+00
        1     0.55125000D-02      0.00000000D+00      0.00000000D+00      -0.20724157D+00     -0.48700959D-01      0.00000000D+00      0.00000000D+00
        1     0.73500000D-02      0.00000000D+00      0.00000000D+00      -0.20604717D+00     -0.49149313D-01      0.00000000D+00      0.00000000D+00
        1     0.91875000D-02      0.00000000D+00      0.00000000D+00      -0.20487106D+00     -0.49563661D-01      0.00000000D+00      0.00000000D+00
        1     0.11025000D-01      0.00000000D+00      0.00000000D+00      -0.20371175D+00     -0.49945258D-01      0.00000000D+00      0.00000000D+00
        1     0.12862500D-01      0.00000000D+00      0.00000000D+00      -0.20256787D+00     -0.50295378D-01      0.00000000D+00      0.00000000D+00

⋮
```

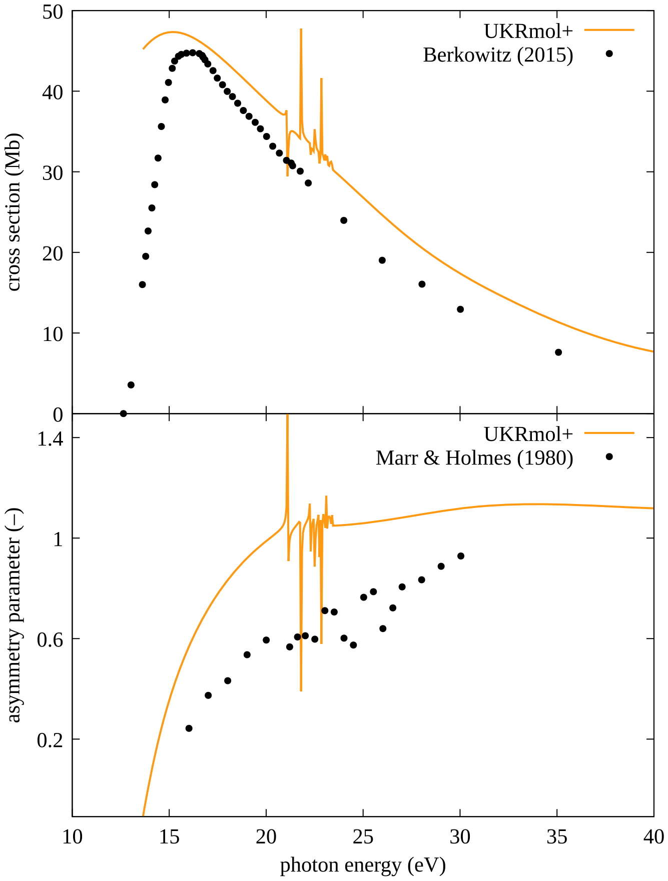

Fig. 1. Cross section (upper panel) and asymmetry parameter (bottom panel) for one-photon ionization into the ground state of CH_4^+ calculated with the UKRmol+ suite and the scripts and input described in this publication. Experimental data for the cross section is from J. Berkowitz [36] while the measurements of the asymmetry parameter come from Marr and Holmes [37].

9. Conclusions

We have described a set of Perl-based scripts that efficiently drive the complex UKRmol+ suite of codes for electron scattering and photoionization calculations in the fixed-nuclei approximation. The required user input has been condensed, as much as possible, to the physical parameters of the model to reduce the need to understand many technical switches and parameters of the calculation. The scripts provide the functionality to automatically generate at run-time the input files for the various programs of the suite and to extract and post-process the most relevant physical outputs of the calculation.

In addition, the scripts drive the prerequisite quantum chemistry calculations including the open-source codes Psi4 and Molcas. Therefore the full R-matrix calculations described above can be performed using only open-source software. We have provided several basic examples of scattering and photoionization calculations which the users can modify to generate their input for arbitrary molecules.

The scripts have been used for many years to generate the data required to model, for example, nuclear dynamics in resonant electron-molecule scattering [38–43], time-resolved photoelectron spectroscopy [44,45] and time-dependent ultrafast laser-molecule interactions [10,11], to name a few recent applications. The release of these scripts will enable many others in the AMO community to investigate photoionization and ultrafast processes either fully with the UKRmol+ suite or by generating the molecular data (e.g. transition dipoles) required by their codes and approaches.

CRediT authorship contribution statement

Karel Houfek: Conceptualization, Methodology, Software, Supervision, Writing – original draft, Writing – review & editing. **Jakub Benda:** Software, Writing – original draft, Writing – review & editing, Validation. **Zdeněk Mašín:** Conceptualization, Methodology, Software, Writing – original draft, Writing – review & editing. **Alex Harvey:** Software, Writing – original draft. **Thomas Meltzer:** Software, Validation, Writing – original draft. **Vincent Graves:** Software, Validation, Writing – original draft. **Jimena D. Gorfinkiel:** Conceptualization, Methodology, Software, Supervision, Validation, Writing – original draft, Writing – review & editing.

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.

Data availability

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

Acknowledgement

J. D. G. is grateful to Daniel Darby-Lewis who initiated the implementation of positron scattering calculations in the scripts supported by eCSE project eCSE13-14. The development of the UKRmol-scripts has been supported by EPSRC under grants EP/P022146/1 and EP/R029342/1. V. G. acknowledges support from the EPSRC Doctoral Training Partnership EP/T518165/1. Some of the developments were carried out as part of projects 20-15548Y of the Czech Science Foundation and the PRIMUS project (20/SCI/003) of Charles University.

Large-scale deployment of this software has been tested using computational resources supplied by the e-INFRA CZ project (ID:90254), supported by the Ministry of Education, Youth and Sports of the Czech Republic.

References

[1] Z. Mašín, J. Benda, J.D. Gorfinkiel, A.G. Harvey, J. Tennyson, UKRmol+: a suite for modelling electronic processes in molecules interacting with electrons, positrons and photons using the R-matrix method, Comput. Phys. Commun. 249 (2020) 107092, https://doi.org/10.1016/j.cpc.2019.107092.

[2] P.G. Burke, R-Matrix Theory of Atomic Collisions: Application to Atomic, Molecular and Optical Processes, Springer, Berlin, 2011.

[3] J. Tennyson, Electron-molecule collision calculations using the R-matrix method, Phys. Rep. 491 (2010) 29–76, https://doi.org/10.1016/j.physrep.2010.02.001.

[4] A.C. Brown, G.S. Armstrong, J. Benda, D.D. Clarke, J. Wragg, K.R. Hamilton, Z. Mašín, J.D. Gorfinkiel, H.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, https://doi.org/10.1016/j.cpc.2019.107062.

[5] L.R. Moore, M.A. Lysaght, L.A.A. Nikolopoulos, J.S. Parker, H.W. van der Hart, K.T. Taylor, The RMT method for many-electron atomic systems in intense short-pulse laser light, J. Mod. Opt. 58 (2011) 1132, https://doi.org/10.1080/09500340.2011.559315.

[6] D.D.A. Clarke, G.S.J. Armstrong, A.C. Brown, H.W. van der Hart, R-matrix-with-time-dependence theory for ultrafast atomic processes in arbitrary light fields, Phys. Rev. A 98 (2018) 053442, https://doi.org/10.1103/PhysRevA.98.053442.

[7] M. Hartmann, L. Hutcheson, G.D. Borisova, P. Birk, S. Hu, A.C. Brown, H.W. van der Hart, C. Ott, T. Pfeifer, Core-resonance line-shape analysis of atoms undergoing strong-field ionization, J. Phys. B: At. Mol. Opt. Phys. 55 (2022) 245601, https://doi.org/10.1088/1361-6455/ac9872.

[8] W. Jiang, G.S. Armstrong, J. Tong, Y. Xu, Z. Zuo, J. Qiang, P. Lu, D.D. Clarke, J. Benda, A. Fleischer, et al., Atomic partial wave meter by attosecond coincidence metrology, Nat. Commun. 13 (1) (2022) 1–9, https://doi.org/10.1038/s41467-022-32753-8.

[9] A.C. Brown, H.W. van der Hart, Extreme-ultraviolet-initated high-order harmonic generation: driving inner-valence electrons using below-threshold-energy extreme-ultraviolet light, Phys. Rev. Lett. 117 (2016) 093201, https://doi.org/10.1103/PhysRevLett.117.093201.

[10] J. Benda, J.D. Gorfinkiel, Z. Mašín, G.S.J. Armstrong, A.C. Brown, D.D.A. Clarke, H.W. van der Hart, J. Wragg, Perturbative and nonperturbative photoionization of H_2 and H_2O using the molecular R-matrix-with-time method, Phys. Rev. A 102 (2020) 052826, https://doi.org/10.1103/PhysRevA.102.052826.

[11] J. Benda, Z. Mašín, J.D. Gorfinkiel, Analysis of RABITT time delays using the stationary multiphoton molecular R-matrix approach, Phys. Rev. A 105 (2022) 053101, https://doi.org/10.1103/PhysRevA.105.053101.

[12] J. Benda, Z. Mašín, Multi-photon above threshold ionization of multi-electron atoms and molecules using the R-matrix approach, Sci. Rep. 11 (2021) 11686, https://doi.org/10.1038/s41598-021-89733-z.

[13] J.M. Carr, P.G. Galiatsatos, J.D. Gorfinkiel, A.G. Harvey, M.A. Lysaght, D. Madden, Z. Mašín, M. Plummer, J. Tennyson, H.N. Varambhia, UKRmol: a low-energy electron- and positron-molecule scattering suite, Eur. Phys. J. D 66 (3) (2012) 58, https://doi.org/10.1140/epjd/e2011-20653-6.

[14] K. Houfek, J. Benda, Z. Mašín, A. Harvey, T. Meltzer, V. Graves, J. Gorfinkiel, UKRmol-scripts, https://doi.org/10.5281/zenodo.7851856, Apr. 2023.

[15] B. Cooper, M. Tudorovskaya, S. Mohr, A. O'Hare, M. Hanicinec, A. Dzarasova, J.D. Gorfinkiel, J. Benda, Z. Mašín, A.F. Al-Refaie, P.J. Knowles, J. Tennyson, Quantemol electron collisions (QEC): an enhanced expert system for performing electron molecule collision calculations using the R-matrix method, Atoms 7 (4) (2019) 97, https://doi.org/10.3390/atoms7040097.

[16] AMOSgateway, https://amosgateway.org/, 2019.

[17] B.I. Schneider, K. Bartschat, O. Zatsarinny, K.R. Hamilton, I. Bray, A. Scrinzi, F. Martin, J. Gonzalez Vasquez, J. Tennyson, J.D. Gorfinkiel, R. Lucchese, S. Pamidighantam, Atomic and molecular scattering applications in an Apache Airavata Science Gateway, in: Practice and Experience in Advanced Research Computing, PEARC '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 270–277.

[18] Z. Mašín, J. Benda, A.G. Harvey, A. Al-Refaie, J.D. Gorfinkiel, J. Tennyson, UKRMol+: UKRMol-in, https://doi.org/10.5281/zenodo.5799110, Dec. 2021.

[19] UK R-matrix community, UKRMol+: UKRMol-out, https://doi.org/10.5281/zenodo.5799134, Dec. 2020.

[20] Z. Mašín, J. Benda, J.D. Gorfinkiel, GBTOlib: a high-performance library for evaluation of molecular integrals, https://doi.org/10.5281/zenodo.4118421, Oct. 2020.

[21] H.-J. Werner, P.J. Knowles, et al., MOLPRO, version, a package of ab initio programs, https://www.molpro.net.

[22] F. Aquilante, J. Autschbach, R.K. Carlson, L.F. Chibotaru, M.G. Delcey, L. De Vico, I.F. Galván, N. Ferré, L.M. Frutos, L. Gagliardi, M. Garavelli, A. Giussani, C.E. Hoyer, G. Li Manni, H. Lischka, D. Ma, P.r. Malmqvist, T. Mueller, A. Nenov, M. Olivucci, T.B. Pedersen, D. Peng, F. Plasser, B. Pritchard, M. Reiher, I. Rivalta, I. Schapiro, J. Segarra-Martí, M. Stenrup, D.G. Truhlar, L. Ungur, A. Valentini, S. Vancoillie, V. Veryazov, V.P. Vysotskiy, O. Weingart, F. Zapata, R. Lindh, Molcas 8: new capabilities for multiconfigurational quantum chemical calculations across the periodic table, J. Comput. Chem. 37 (5) (2016) 506–541, https://doi.org/10.1002/jcc.24221.

[23] D.G.A. Smith, L.A. Burns, A.C. Simmonett, R.M. Parrish, M.C. Schieber, R. Galvelis, P. Kraus, H. Kruse, R. Di Remigio, A. Alenaizan, A.M. James, S. Lehtola, J.P. Misiewicz, M. Scheurer, R.A. Shaw, J.B. Schriber, Y. Xie, Z.L. Glick, D.A. Sirianni, J.S. O'Brien, J.M. Waldrop, A. Kumar, E.G. Hohenstein, B.P. Pritchard, B.R. Brooks, H.F. Schaefer, A.Y. Sokolov, K. Patkowski, A.E. DePrince, U. Bozkaya, R.A. King, F.A. Evangelista, J.M. Turney, T.D. Crawford, C.D. Sherrill, PSI4 1.4: open-source software for high-throughput quantum chemistry, J. Chem. Phys. 152 (18) (2020) 184108, https://doi.org/10.1063/5.0006002.

[24] B. Szabó, Y. Champoux, G. Szabo, Parallel::ForkManager, https://metacpan.org/pod/Parallel::ForkManager, 2018.

[25] A.G. Harvey, D.S. Brambila, F. Morales, O. Smirnova, An R-matrix approach to electron–photon–molecule collisions: photoelectron angular distributions from aligned molecules, J. Phys. B: At. Mol. Opt. Phys. 47 (21) (2014) 215005, https://doi.org/10.1088/0953-4075/47/21/215005.

[26] G. Schaftenaar, J.H. Noordik, Molden: a pre- and post-processing program for molecular and electronic structures, J. Comput.-Aided Mol. Des. 14 (2) (2000) 123–134, https://doi.org/10.1023/A:1008193805436.

[27] K. Bartschat, E.T. Hudson, M.P. Scott, P.G. Burke, V.M. Burke, Electron-atom scattering at low and intermediate energies using a pseudo-state/R-matrix basis, J. Phys. B: At. Mol. Opt. Phys. 29 (1996) 115–123, https://doi.org/10.1088/0953-4075/29/1/015.

[28] J.D. Gorfinkiel, J. Tennyson, Electron-H_3^+ collisions at intermediate energies, J. Phys. B: At. Mol. Opt. Phys. 37 (20) (2004) L343–L350, https://doi.org/10.1088/0953-4075/37/20/l01.

[29] A. Faure, J.D. Gorfinkiel, L.A. Morgan, J. Tennyson, GTOBAS: fitting continuum functions with Gaussian-type orbitals, Comput. Phys. Commun. 144 (2) (2002) 224–241, https://doi.org/10.1016/S0010-4655(02)00141-8.

[30] J. Tennyson, C.J. Noble, RESON–A program for the detection and fitting of Breit-Wigner resonances, Comput. Phys. Commun. 33 (1984) 421–424, https://doi.org/10.1016/0010-4655(84)90147-4.

[31] A.G. Harvey, The multiple active space approach to electron-molecule collisions using the R-matrix method, in preparation.

[32] J. Ivanic, Direct configuration interaction and multiconfigurational self-consistent-field method for multiple active spaces with variable occupations. I. Method, J. Chem. Phys. 119 (18) (2003) 9364–9376, https://doi.org/10.1063/1.1615954.

[33] D. Ma, G. Li Manni, L. Gagliardi, The generalized active space concept in multiconfigurational self-consistent field methods, J. Chem. Phys. 135 (4) (2011) 044128, https://doi.org/10.1063/1.3611401.

[34] T. Meltzer, Z. Mašín, A polarization-consistent R-matrix model: application to the photoionization of molecules, J. Phys. B: At. Mol. Opt. Phys. 55 (3) (2022) 035201, https://doi.org/10.1088/1361-6455/ac4d74.

[35] D. Ertel, D. Busto, I. Makos, M. Schmoll, J. Benda, H. Ahmadi, M. Moioli, F. Frassetto, L. Poletto, C.D. Schröter, T. Pfeifer, R. Moshammer, Z. Mašín, S. Patchkovskii, G. Sansone, Influence of nuclear dynamics on molecular attosecond photoelectron interferometry, Sci. Adv. 9 (35) (2023) eadh7747, https://doi.org/10.1126/sciadv.adh7747.

[36] J. Berkowitz, Polyatomic molecules, in: J. Berkowitz (Ed.), Atomic and Molecular Photoabsorption, Academic Press, Amsterdam, 2015, pp. 391–509, Chapter 5, https://www.sciencedirect.com/science/article/pii/B9780128019436000058.

[37] G.V. Marr, R.M. Holmes, The angular distribution of photoelectrons from CH4 as a function of photon energy from near threshold to 30 eV, J. Phys. B: At. Mol. Opt. Phys. 13 (5) (1980) 939, https://doi.org/10.1088/0022-3700/13/5/018, https://dx.doi.org/10.1088/0022-3700/13/5/018.

[38] K. Houfek, M. Čížek, Lowest autodetachment state of the water anion, Eur. Phys. J. D 70 (2016) 107, https://doi.org/10.1140/epjd/e2016-70148-y.

[39] M. Zawadzki, M. Čížek, K. Houfek, R. Čurík, M. Ferus, S. Civiš, J. Kočišek, J. Fedor, Resonances and dissociative electron attachment in HNCO, Phys. Rev. Lett. 121 (14) (2018) 143402, https://doi.org/10.1103/PhysRevLett.121.143402.

[40] V. Alt, K. Houfek, Resonant collisions of electrons with O_2 via the lowest-lying $^2\Pi_g$ state of O_2^-, Phys. Rev. A 103 (2021) 032829, https://doi.org/10.1103/PhysRevA.103.032829.

[41] T.P. Ragesh Kumar, P. Nag, M. Ranković, T.F.M. Luxford, J. Kočišek, Z. Mašín, J. Fedor, Distant symmetry control in electron-induced bond cleavage, J. Phys. Chem. Lett. 13 (48) (2022) 11136–11142, https://doi.org/10.1021/acs.jpclett.2c03096.

[42] J. Dvořák, M. Ranković, K. Houfek, P. Nag, R. Čurík, J. Fedor, M. Čížek, Vibronic coupling through the continuum in the $e + CO_2$ system, Phys. Rev. Lett. 129 (1) (2022) 013401, https://doi.org/10.1103/PhysRevLett.129.013401.

[43] J. Dvořák, K. Houfek, M. Čížek, Vibrational excitation in the $e+CO_2$ system: nonlocal model of $\Sigma\Pi$ vibronic coupling through the continuum, Phys. Rev. A 105 (6) (2022) 062821, https://doi.org/10.1103/PhysRevA.105.062821.

[44] M. Richter, J. Gonzalez-Vazquez, Z. Mašín, D.S. Brambila, A. Harvey, F. Morales, F. Martín, Ultrafast imaging of laser-controlled non-adiabatic dynamics in NO_2 from time-resolved photoelectron emission, Phys. Chem. Chem. Phys. 21 (2019) 10038–10051, https://doi.org/10.1039/C9CP00649D.

[45] D.S. Brambila, A.G. Harvey, K. Houfek, Z. Mašín, O. Smirnova, Role of electronic correlations in photoionization of NO_2 in the vicinity of the $^2A_1/^2B_2$ conical intersection, Phys. Chem. Chem. Phys. 19 (30) (2017) 19673–19682, https://doi.org/10.1039/C7CP01643C.