EXP-T

a relativistic multireference coupled cluster program

v 1.8.5

User manual

June 10, 2025

Contents

1 Introduction

The EXP-T program package is designed for high-precision modeling of electronic structure of atoms and molecules using the relativistic Fock space coupled cluster method (FS-RCC). Features and program components are listed in Sec. 1.1 and 1.2.

The authors will be grateful for any comments or suggestions:
exp-t-program@googlegroups.com
https://groups.google.com/d/forum/exp-t-program

1.1 Features of EXP-T

The EXP-T package does not include subroutines for solving the (Dirac-)Hartree-Fock equations and subsequent four-index transformation, so molecular integrals have to be imported from third party electronic structure packages. Currently EXP-T is interfaced to the DIRAC program package [1, 2], thus providing access to the wide variety of Hamiltonians and property operators implemented therein.

Electronic structure models available in EXP-T:

- single-point energy calculations with any point groups and (nearly) all Hamiltonians, implemented in DIRAC (4c-DC, X2Cmmf, two-component relativistic pseudopotential (RPP), scalar-relativistic or non-relativistic);
- ground state energy calculations (the $0h0p$ sector): CCSD, CCSD(T), CCSDT-n (n=1,2,3), CCSDT models;
- the FS-CCSD method for ionized and excited states is implemented for the $0h1p$, $1h0p$, $1h1p$, $0h2p$, $2h0p$, $0h3p$, $1h2p$, $2h1p$, $3h0p$ Fock space sectors;
- FS-CC models for excited states accounting for triples (CCSDT-n (n=1,2,3), CCSDT) are implemented for the $0h1p$, $1h0p$, $1h1p$, $0h2p$, $2h0p$, $0h3p$, $1h2p$, $3h0p$ Fock space sectors.

  The summary of all electronic structure models implemented in the EXP-T program system is given in the table:

FS sector	CCSD	CCSD+T(3)	CCSDT-1	CCSDT-2	CCSDT-3	CCSDT
$0h0p$	+	+	+	+	+	+
$1h0p$	+	t	t	t	t	t
$0h1p$	+	t	t	t	t	t
$2h0p$	+	–	t	t	t	t
$1h1p$	+	t	t	t	t	t
$0h2p$	+	t	t	t	t	t
$3h0p$	t	–	t	t	t	t
$2h1p$	t	–	–	–	–	–
$1h2p$	t	–	–	–	–	t
$0h3p$	+	t	t	t	t	t

"+" – implemented, "–" – not implemented, "t" – currently being tested

Features for property calculations and analysis of wavefunctions:

- analytic density matrices and property calculations (including $\mathcal{T}$-odd operators) for the ground-state CCSD and CCSD(T) models;

- finite-field [3] and finite-order [4] approaches to calculation of transition properties;
- (unrelaxed) analytic density matrix and properties for ground-state CCSD and CCSD(T);
- quasi-diabatization of spin-orbit (SO) coupled states and extraction of spin-orbit interaction matrix elements [5].
- construction of model natural spinors (NS) and natural transition spinors (NTS) [6, 7] for non-trivial Fock space sectors;
- one-electron properties *via* the interface to either DIRAC or the OneProp program by L. V. Skripnikov [8].
- finite-field calculations of diagonal two-electron properties (assuming that integrals are provided).

Features to deal with the intruder state problem:

- "dynamic" shifts of energy denominators [5];
- Padé extrapolation to the zero-shift limit [9];
- intermediate Hamiltonian technique for incomplete main model spaces (IH-IMMS) [10].

Features for high-performance calculations:

- OpenMP parallelization (for shared-memory systems);
- parallel calculations on NVIDIA GPUs using the CUDA platform (to be tested) [11].

1.2 Program components

`expt.x` Input processing + all CC calculations.

`heffman.x` Manipulations with effective Hamiltonian matrices, i. e. Padé extrapolations, finite-field transition property calculations, quasi-diabatization of spin-orbit-coupled states.

`expt_transform_natorb.x`

Transformation of model natural spinors to the basis of atomic orbitals.

`expt_diatomic.x`

Rovibrational levels of diatomic molecules and calculations of expectation values and transition properties for them.

`expt2pam.x` A tool for specifying molecular geometry, basis sets and pseudopotentials in a convenient human-readable format; the program automatically generates input files for other quantum chemistry packages.

`expt2spectrum.py`

A tool for convenient extraction of calculated information about electronic states and transitions between them from EXP-T output files.

1.3 Citation

We kindly ask you to acknowledge any use of the EXP-T program system that results in a published material using the following citation:

A. V. Oleynichenko, A. Zaitsevskii, E. Eliav. Towards high performance relativistic electronic structure modelling: the EXP-T program package. *Commun. Comput. Inf. Sci.* 1331, 375 (2020). doi: 10.1007/978-3-030-64616-5_33

In addition, credits must be given to the corresponding papers which describe the implementation and the underlying methodological developments:

- models including triple excitations in the low Fock space sectors: [12];
- Fock-space CC for the three-particle sector ($0h3p$): [13, 14];
- finite-field technique transition property calculations (general): [3];
- finite-field technique for transition matrix elements involving the $0h0p \to 1h1p$ excitations: [15];
- spin-orbit coupling calculations: [5];
- dynamic shifts of energy denominators (real): [5];
- dynamic shifts of energy denominators (imaginary, or their real simulation): [16];
- Padé extrapolation of a series of effective Hamiltonians to the zero-shift limit: [9];
- Intermediate Hamiltonian technique for incomplete main model spaces (IH-IMMS): [10];
- finite-order method of transition matrix elements calculations: [4, 17].

1.4 Credits

EXP-T is based on ideas and design solutions of the suite of FS-CC programs written by Ephraim Eliav and Uzi Kaldor [18, 19].

2 Installation from the source code

2.1 Compiling EXP-T

EXP-T is currently oriented at Unix-like operating systems.

Tools required for compiling the EXP-T package from the source code:

- C and Fortran compilers. Currently supported compiler systems are

`gcc/gfortran`	GNU compilers
`icc/ifort`	Intel compilers
`clang`	a C language family frontend for LLVM (for the C part)

- `cmake` (version 3.0.2 or higher) and `make` utilities:
  https://cmake.org/
  https://www.gnu.org/software/make/

Optional dependencies:

- Implementation of the BLAS/LAPACK linear algebra libraries (OpenBLAS [20] or Intel MKL [21] are recommended);
- NVIDIA drivers and CUDA Toolkit are required to perform parallel calculations on a GPU:
  https://www.nvidia.com/Download/index.aspx?lang=en-us
  https://developer.nvidia.com/cuda-toolkit

Download the EXP-T source code as a `*.tar.xz` file and unpack it:

```
tar xvf expt-X.Y.Z.tar.xz
```

Go to the EXP-T home directory, create the `build` directory and make it the current working dir:

```
mkdir build
cd build
```

Compilation *(Intel compilers)*:

```
CC=icc FC=ifort cmake ..
make
```

Compilation *(GNU compilers)*:

```
CC=gcc FC=gfortran cmake ..
make
```

We recommend specifying compilers explicitly (since the default C compiler may be other than `gcc`).

The `cmake` utility will try to locate BLAS/LAPACK libraries on your computer and link them to the project. If there are no pre-installed libraries on your machine, the internal OpenBLAS will be compiled and linked to EXP-T.

When using Intel MKL together with GNU compilers, you will probably need to explicitly specify the path to the MKL home directory by setting the environment variable `MKLROOT`. For example:

```
export MKLROOT=/opt/intel/mkl
```

If compilation was successful, the executable file `expt.x` will appear in the `build` directory.

We strongly recommend that compilation is followed by testing (see Sect 2.4).

Do not forget to add the directory containing binary files to the `PATH` environment variable!

2.1.1 Compiling on macOS

The default C compiler for macOS is clang (https://clang.llvm.org). Thus one can use it instead of gcc (however, the gcc/gfortran pair is still acceptable):

```
CC=clang FC=gfortran cmake ..
make
```

Note that by default gcc is just an alias for clang on macOS systems. We strongly recommend not to use GNU compilers installed with Homebrew. Instead, download the GNU compilers suitable for your version of macOS from:
http://hpc.sourceforge.net/

We recommend to use OpenBLAS attached to the EXP-T as a BLAS/LAPACK implementation.

2.2 Compiling with CUDA

Compilation of EXP-T does not require the user to have a CUDA-compatible GPU device. If the CUDA toolkit is missing, the CUDA code will simply be excluded from the compilation process.

EXP-T is adapted for parallel calculations on NVIDIA GPUs (the CUDA technology [11]).
Notes:

1. At the moment, the OpenMP+CUDA hybrid model as well as multi-GPU support is not implemented. Only one GPU can be used for calculations.
2. non-professional GPUs for gaming are not suitable for CUDA calculations (except for Kepler cards).
3. CUDA code in EXP-T was tested only for CUDA 9.1 and 10.0.
4. Only Intel and GNU compilers were tested for compatibility with CUDA.

To compile EXP-T with CUDA:

1. Make sure your graphics card supports CUDA. CUDA-compatible GPUs are listed here:
   https://developer.nvidia.com/cuda-gpus
2. Install proprietary NVIDIA drivers (not the nouveau driver!).
   https://www.nvidia.com/Download/index.aspx?lang=en-us.
   For Linux you can find drivers in your distribution's repository.
3. Install CUDA toolkit
   https://developer.nvidia.com/cuda-downloads
   or from distribution's repository.
4. Make sure that all drivers and CUDA toolkit are installed and work correctly. Compile the test suite in the samples directory of the CUDA home directory and run some of them. The most important are the tests employing the CUBLAS linear algebra library.
5. Compile EXP-T as described in Sec. 2.1. All CUDA tools will be located automatically by CMake.
6. Make sure that the results produced by EXP-T do not depend on the use of CUDA.

2.3 Troubleshooting

This section describes the most common problems which can occur during the compilation process. In case of any questions or problems, please don't hesitate to contact us. Don't forget to attach the output file (in case of problems with calculations) or the file with the output of the make VERBOSE=1 command (in case of problems with compilation).

Google Groups:
https://groups.google.com/d/forum/exp-t-program

E-mail:

exp-t-program@googlegroups.com

2.3.1 Outdated version of cmake

We highly recommend using cmake version 3.0.2 or higher, otherwise, problems may occur when building the internal OpenBLAS. The following command will show what version of cmake is installed:

```
cmake --version
```

If the cmake version is lower than 3.0.2, download the binary distribution from their official website https://cmake.org/download/. After installation do not forget to add the directory containing the cmake binary files to the PATH environment variable.

Installation of CMake binary files does not require root privileges.

2.3.2 Segmentation fault in heffman.x

The known problem is a possible segmentation fault inside the heffman.x program. It was discovered for the cases when EXP-T is compiled with the internal OpenBLAS and gfortran compilers (version 7.4 and 7.5). The segmentation fault occurs inside the zheev subroutine (diagonalization of a complex Hermitian matrix).

If this issue occurs, we recommend using a different version of the compiler or the BLAS/LAPACK implementation.

2.4 Testing

The test suite is located in the test directory. The testing system is written in the Python programming language (must be pre-installed on your machine). Either Python 2 or Python 3 can be used.

In order to run the test suite, run the following command:

```
make test
```

3 Running EXP-T

See Section 6 for a description of an additional utitily programs included in the EXP-T package.

A typical coupled cluster calculation consists of two sequential steps:

1. SCF and integral transformation within DIRAC (Section 3.1);
2. CC calculation within EXP-T (Section 3.2).

3.1 Precomputing spinors and molecular integrals: the DIRAC step

The interface to DIRAC was tested only for the DIRAC17, DIRAC18 and DIRAC19 releases.

DIRAC stores transformed molecular integrals in the following binary files:

`MRCONEE` information about SCF calculation, symmetry, occupation numbers, one-electron integrals;

`MDCINT` symmetry-unique nonzero two-electron integrals;

`MDPROP` integrals of one-electron property operators.

Run DIRAC and save the files containing the transformed integrals to the working directory using the `--get` option:

```
pam --noarch --mol=<mol-file> --inp=<inp-file> --get="MRCONEE MDCINT MDPROP"
```

If DIRAC starts in the parallel mode (MPI), it is necessary to choose the "scheme 4" transformation algorithm. By default, DIRAC uses the "scheme 6" algorithm which produces multiple `MDCINT` files [22] and currently EXP-T requires that all integrals are stored in a single file. In order to accomplish this, add the following lines to the DIRAC input file:

```
**MOLTRA
.SCHEME
4
```

To calculate properties, transition moments and intensities without resorting to the finite-field scheme property integrals transformed to the basis of molecular spinors are also required. We recommend explicitly asking DIRAC to calculate and transform these integrals by adding the `.PRPTRA` keyword together with the list of the required property operators to DIRAC input files:

```
**MOLTRA
.PRPTRA
 . . . . . . . .
**PROPERTIES
.DIPOLE
```

If the molecular-mean-field X2C Hamiltonian (X2Cmmf) was used at the DIRAC stage, one ***must*** *specify the* ***`x2cmmf`*** *flag in the EXP-T input file! (see also page 25).*

3.2 Coupled cluster calculation: the EXP-T step

Run an EXP-T job using the following command:

```
expt.x [options] <input-file>
```

The EXP-T output (including error messages) is flushed to the UNIX standard output (`stdout`). Use the output redirection operator ">" in case of time-consuming calculations:

```
nohup expt.x <input-file>  >  <output-file> &
```

Examples of DIRAC and EXP-T input files are given in Section 5.

3.3 Command-line arguments

Use the `expt.x --help` command to print a list of available command-line arguments:

```
Usage: expt.x [OPTION...] <input-file>
expt -- A Fock-Space Multireference Relativistic Coupled-Cluster Program

  -n, --no-clean             Do not clean scratch directory on exit
                             (use this option to keep cluster amplitudes etc)
  -s, --scratch=PATH         Path to scratch directory (default: ./scratch)
  -?, --help                 Print this help list and exit
      --usage                Print a short usage message and exit
  -V, --version              Print program version and exit

Mandatory or optional arguments to long options are also mandatory or optional
for any corresponding short options.

Please report bugs to <exp-t-program@googlegroups.com>.
```

4 Input files syntax and keywords

See Section 6 for the description of additional utitily programs included in the EXP-T package.

The input file format may change in future versions.

The EXP-T input file contains a list of instructions. Each instruction is placed on a separate line and consists of a keyword and a list of its arguments. The order of instructions in the input file can be arbitrary. Single-line comments start with `'#'` and continue until the end of the line. Input is case insensitive. EXP-T input files for typical calculations are given and discussed in details in Section 5. Molecular geometry, basis sets and pseudopotentials are specified only once in DIRAC input files at the molecular integrals calculation stage (not in EXP-T input files).

The detailed description of the keywords used in the EXP-T input file is provided below. For convenience, keywords are grouped into several sections.

General settings

`title` optional comment.

Syntax: `title <quoted-string>`

Default: no title

Example: `title "my first ccsd calculation"`

`arith` complex arithmetic will be used throughout the calculation.

Default: arithmetic depends on molecular symmetry (see [23] for details).

Syntax: `arith complex`

`print` print level.

Syntax: `print ( low || medium || high || debug || <quoted-string> )`

where `<quoted-string>` can be:

`"eff config"` analysis of electronic states in terms of effective configurations of active spinors;

`"model vectors"` expansions of model vectors in terms of model determinants;

`"model space"` detailed information about model space determinants.

One can also combine several `print` instructions.

Default: `print medium`

Example:

```
print low
print "eff config"
```

`degen_thresh` energy levels are considered as degenerate if the energy gap between them is lower than `degen_thresh` (in a. u.). It is recommended to decrease the threshold in case of finite-difference calculations of properties.

Syntax: `degen_thresh <real thresh> [au | ev | cm | mhz]`

where allowed units are:

`au`	atomic units (hartree; by default);
`ev`	electronvolts (eV);
`cm`	wavenumbers (cm^{-1});
`mhz`	megahertz (MHz).

Default: `degen_thresh 1e-8`
(approximately 0.002 cm^{-1})

`spinor_labels` manual assignment of symbolic names (labels) of specific atomic or molecular spinors (or spin-orbitals). Symbolic name is printed together with the symmetry of a spinor and its one-electron energy. This option can be useful to study composition of electronic states.

Syntax:

```
spinor_labels
  . . .
  <integer spinor-number> <string spinor-label>
  . . .
end
```

Default: disabled.

Example: labels of atomic spinors:

```
spinor_labels
  2  2s_1/2
  3  2p_1/2
  4  2p_3/2
  5  2p_3/2
end
```

Model

`sector` specifies the target Fock space sector. To the moment the $0h0p$, $0h1p$, $1h0p$, $1h1p$, $0h2p$, $2h0p$, $3h0p$, $0h3p$, $1h2p$ and $2h1p$ FS sectors are implemented (h – holes, p – particles).

Syntax: `sector <integer H>h<integer P>p`

Default: `sector 0h0p` (vacuum)

The code for the $1h2p$ and $2h1p$ sectors is experimental and is not well-tested!

`model` CC model (approximation to the cluster operator).

Syntax: `model ( ccs || ccd || ccsd || ccsd(t) || ccsd+t(3) || ccsdt-1 || ccsdt-2 || ccsdt-3 || ccsdt )`

Default: `model ccsd`

Notes:

- Keywords `ccsd(t)` and `ccsd+t(3)` are equivalent.
- The codes implementing the CCSD+T(3), CCSDT-n and CCSDT models in the non-trivial sectors and all CC models in the $0h3p$, $3h0p$, $1h2p$ and $2h1p$ sectors are experimental (currently being tested), to be used with care.

- The current implementation of the CCSD(T) model for the $0h0p$ sector is not very computationally efficient.
- The CCS and CCD models are implemented simply by setting all T_2 (CCS) or T_1 (CCD) amplitudes to zero, this does not make computation faster, and exists primarily for the purposes of testing and debugging.

`occ` occupation numbers of individual spinors in a vacuum state. This keyword can be used to setup calculations with a "high-spin" vacuum state [24]. This keyword is alternative to the `occ_irreps` keyword (see below).

Default: occupation numbers are read from files containing transformed molecular integrals

Syntax: `occ <list of the 0 and 1 digits separated by spaces>`

Example: 5 electrons and 10 spinors:

```
occ 1 1 0 0 0 1 1 1 0 0
```

CC calculations with high-spin vacuum states were tested for the 0h0p FS sector only.

It is rather convenient to use the `dirac_mointegral_export` routine [25] from the DIRAC package to generate string with occupation numbers.

`occ_irreps` specification of a vacuum state using total numbers of electrons in each irreducible representation. Irrep occupation numbers are ignored if occupation numbers of individual spinors are given (by the `occ` keyword, see above). Names of irreducible representations adopted in EXP-T for some abelian groups are listed in Appendix D.

Syntax: `occ_irreps <pairs irrep:occ>`

Example: occupation numbers for the ground triplet state of the O_2 molecule (the $\overline{D_{\infty h}}$ point group):

```
# for gerade irreps:
occ_irreps [1/2g+]:3 [1/2g-]:2 [3/2g+]:0 [3/2g-]:1
# for ungerade irreps:
occ_irreps [1/2u+]:2 [1/2u-]:2 [3/2u+]:1 [3/2u-]:1
```

Default: occupation numbers are read from MO integrals files.

Active space

`nacth/nactp` active (i. e. valence) space specification (by the overall number of active spinors, regardless of their symmetry). This method of specifying an active space is recommended for non-experts.

- `nacth` – number of highest occupied spinors considered as active *(active holes)*
- `nactp` – number of lowest virtual spinors considered as active *(active particles)*

It is convenient to specify an active space using the `nacth/nactp` keywords when exploring electronic states of molecules in a range of geometries (potential energy surfaces, transition moment functions etc). Using the `nacth/nactp` keywords one can specify either the overall number of active spinors (holes or particles) or the number of active spinors in each irreducible representation.

Syntax:

```
nacth <integer dim>
nactp <integer dim>
```

or

```
nacth <pairs irrep:number-of-spinors>
nactp <pairs irrep:number-of-spinors>
```

Default:

```
nacth 0
nactp 0
```

(no active spinors).

Example:

Let us consider low-lying electronic states of the La atom (see [14] for a discussion). Those are dominated by the $6s^n5d^m$ configurations, generating even electronic states. So it seems reasonable to include the $6s_{1/2}$, $5d_{3/2}$ and $5d_{5/2}$ atomic spinors of the La^{3+} cation into the active space and obtain target electronic states in the $0h3p$ Fock space sector. In order to make the input file easy to read, one can use two separate `nactp` directives, one for the “negative” projections of m_j, and one for the “positive” ones:

```
nactp [1/2g+]:3 [3/2g+]:2 [5/2g+]:1
nactp [1/2g-]:3 [3/2g-]:2 [5/2g-]:1
```

One can further include the $6p$ and $4f$ spinors of La^{3+} to the active space to extend the manifold of achievable electronic states:

```
nactp [1/2u+]:4 [3/2u+]:3 [5/2u+]:2 [7/2u+]:1
nactp [1/2u-]:4 [3/2u-]:3 [5/2u-]:2 [7/2u-]:1
```

`active` specifies the space of active (i. e. valence) spinors according to the energy criterion. Only spinors with one-electron energies lying in the range $\varepsilon_i \in [\varepsilon_{min}; \varepsilon_{max}]$ will be considered as the active ones (regardless of their symmetry). Specifying an active space using the `active energy` instruction is convenient in atomic calculations.

Syntax: `active energy <real eps_min> <real eps_max>`

Default: no active spinors

Example: `active energy -1.0 5.0`

Intermediate Hamiltonian for incomplete main model spaces (IH-IMMS)

Intermediate Hamiltonian technique formulated for incomplete main model spaces (IH-IMMS) can be used to achieve convergence of iterative solution of CC equations. The first paper [10] on the IH-IMMS method concerns only the special case of atomic systems. However, a structure of incomplete model spaces for molecules is typically simpler than for the special case of atoms; recent molecular applications with the detailed discussions on IH-IMMS for molecules can be found in [4] (the ThO molecule) and [26] (the AcF molecule).

The basic idea of the method is to split the whole model space into the “main” and ”intermediate” subspaces. Slater determinants $|\Phi_M\rangle$ which span the main model space must be chosen to describe the bulk of wavefunctions of target electronic states. FS-CC equations for amplitudes corresponding to excitations from main determinants remain unchanged. In contrast, equations for amplitudes associated with excitations from intermediate (“buffer”) determinants $|\Phi_I\rangle$ are solved with shifted energy denominators to suppress the intruder state problem. For each cluster amplitude corresponding to the excitation $I \to K$ from $|\Phi_I\rangle$ a shift parameter S_{IK} is calculated as:

$$S_{IK} = \overline{E}_0 - E_I,$$

where E_I stands for the (zero-order) energy of the $|\Phi_I\rangle$ model-space determinant and $\overline{E}_0$ stands for the "frontier" energy. S_{IK} is used to calculate a shift for energy denominator corresponding to a cluster amplitude using one of the formulas (6)–(9). All model space determinants $|\Phi_M\rangle$ with energies $E_M \leq E_0$ are considered as main ones. Determinants $|\Phi_I\rangle$ with energies $E_I > E_0$ are treated as intermediate ones.

To set up the IH-IMMS calculation:

1. define Fock space sectors for which the IH method will be enabled;
2. split the set of active spinors into several subspaces. In the present version of EXP-T only spinor energies can be used to specify these subspaces;
3. define model Slater determinants which will be considered as the "main" ones. Sets of "main" determinants are defined via the numbers of particles (holes) in each spinor subspace;
4. (optional) define the "frontier" energy $\overline{E}_0$. By default, the frontier energy is calculated automatically.

The use of IH-IMMS for the simple case of the carbon atom is considered in details in Section 5.4.

By default, IH-IMMS is disabled. IH settings are specified using the multi-line instruction `ih_imms`:

```
ih_imms
  <option-1>
  <option-2>
  <option-3>
  . . .
end
```

The following keywords can be used inside the `ih_imms` block:

`sectors` a list of Fock space sectors for which the IH-IMMS technique will be applied. This keyword is mandatory.

Syntax: `sectors <list-of-sector-labels>`

Default: no target FS sectors.

Example: `sectors 0h1p 0h2p`

`subspace` defines a subspace of active spinors with energies $\varepsilon \in [\varepsilon_{\text{min}}, \varepsilon_{\text{max}}]$. Subspaces are further used to define composition of main model space determinants via the `main_occ` keyword. One should use several `subspace` options in the input file to define more than one subspace of active spinors. This keyword is mandatory. At least one subspace must be specified.

Syntax: `subspace energy <real emin> <real emax>`

Default: no subspaces are defined.

Example: `subspace energy -1.0 -0.5`

`main_occ` defines a subspace of main model space determinants. The argument of this option contains the number of particles (holes) in the 1st, 2nd, ... subspaces of active spinors. One should use several `main_occ` options in the input file to define more than one subspace of main model space determinants. This keyword is mandatory, at least one subspace of determinants must be specified.

Syntax: `main_occ <list-of-integers>`

Default: no main model spaces are defined.

Example: imagine that we need to define model space determinants with 2 electrons on active spinors belonging to the 1st spinor subspace, 1 electron on spinors from the 2nd subspace and 0 electrons on the rest of active spinors (such a model determinant belongs to the $0h3p$ sector). The proper option will be as follows:

```
main_occ 2 1 0
```

`shift_type` specifies a formula for denominator shift to be used for cluster amplitudes associated with the "intermediate" model space determinants. See also the description for the `shift_type` keyword (Page 24).

Syntax: `shift_type ( none || real || realimag || imag || taylor )`

Default: `shift_type realimag`

`npower` attenuation parameter used in the shifting formula (see Eq. (6)–(9)).

Syntax: `npower <int n>`

Default: `npower 5`

`scale` a scaling factor for the shift parameter S_{IK}:

$$S_{IK} \quad\rightarrow\quad S'_{IK} = \alpha S_{IK}$$

Syntax: `scale <real factor>`

Default: `scale 1.0` (no scaling)

`frontier_energy` "frontier" energy $\overline{E}_0$ used to calculate shift for the given (intermediate) determinant.

Syntax: `frontier_energy <H>h<P>p [<real energy> | lower_bound | upper_bound ]`

Default: frontier energy is evaluated automatically as the maximum energy of main model space determinants (`frontier_energy <H>h<P>p upper_bound`):

$$\overline{E}_0 = \max_{\Phi_M \in \text{main}} E_M, \qquad E_M = \langle \Phi_M | H_0 | \Phi_M \rangle \,.$$

Example: shifting to the minimum energy of model space determinants:

```
frontier_energy 0h2p lower_bound
```

For some pathological cases the automatically determined frontier energy is too high resulting in a poor convergence of an IH-IMMS calculation. This problem can be solved by manually defining frontier energy equal to the minimal energy of model space determinants (`lower_bound`). In almost all practical cases studied to date answers (excitation energies and properties) were found to be pretty stable with respect to the particular value of the frontier energy.

Property calculations

`analyt_prop` analytic calculation of coupled cluster expectation values of one-electron property operators (including $\mathcal{T}$-odd ones). Matrix elements of property operators must be stored in a DIRAC's `MDPROP` unformatted file or a text file (see Appendix B). Λ-equations, (unrelaxed) analytic density matrix and properties are currently available only for the $0h0p$ sector (vacuum state) and the CCSD and CCSD(T) models (see refs. [27, 28] for the theoretical background).

Syntax: `analyt_prop <property-name>`

Default: disabled.

Example: calculation of x,y,z-components of the dipole moment.

```
analyt_prop XDIPLEN

analyt_prop YDIPLEN

analyt_prop ZDIPLEN
```

`density` analytic calculation of one-particle density matrix in the $0h0p$ sector (without contraction with a property operator matrix). Prints occupation numbers of CC natural orbitals. Analytic density matrices are currently implemented for the CCSD and CCSD(T) models only.

Syntax: `density 0h0p`

Default: disabled.

`hermit` perform symmetric orthogonalization of model vectors. Orthogonalized model vectors are used only to calculate properties and approximate density matrices. Matrices of effective Hamiltonians flushed to `HEFF` files (see Section 6.1) remain non-Hermitian.

Syntax: `hermit`

Default: "hermitization" is disabled.

`mdprop` enable calculations of expectation values of one-electron property operators and their transition (off-diagonal) matrix elements (this option can be used for arbitrary one-electron operators, not only dipole moments). Property operator matrix elements transformed to the basis of molecular spinors are to be loaded from the DIRAC's `MDPROP` unformatted file. To avoid huge outputs, use options `nroots` or `roots_cutoff` to restrict the set of target electronic states (or, alternatively, the `expt_spectrum.py` utility to conveniently extract data for the specific transitions, see Section 6.4). Note that the currently available maximum truncation level is quadratic in cluster amplitudes, which might be not enough for systems with strong dynamic correlations (and thus relatively large cluster amplitudes).

The details on the finite-order method of transition matrix element calculations are given in [4]. The implementation of its connected version together with the first benchmark for transition dipole matrix elements is reported in [17]. Within this approach a property matrix element O_{if} for a pair of electronic states ψ_i and ψ_f is calculated *via* the relation [29] (`scheme non-hermitian`):

$$O_{if} = \langle \tilde{\psi}_i^{\perp\perp} | (\Omega^\dagger \Omega)_{closed} \Omega^\dagger O \Omega | \tilde{\psi}_f \rangle \cdot N_i N_f^{-1}, \tag{1}$$

where $\tilde{\psi}_i^{\perp\perp}$ stands for the left model vector and the normalization factors are defined as

$$N_i = \langle \psi_i | \psi_i \rangle^{1/2} = \langle \tilde{\psi}_i | \Omega^\dagger \Omega | \tilde{\psi}_i \rangle^{1/2} \tag{2}$$

(the same for N_f). This definition of the effective property operator leads to the non-Hermitian property matrix, $O_{if} \neq O_{fi}^*$. Any hermitization procedure can be further applied to restore the hermiticity. The working equation contains disconnected diagrams, thus it can suffer from the lack of size-consistency.

There is also an alternative definition (`scheme hermitian`):

$$O_{if} = \langle \tilde{\psi}_i | \Omega^\dagger O \Omega | \tilde{\psi}_f \rangle \cdot N_i^{-1} N_f^{-1} \tag{3}$$

which is inherently Hermitian [29], but also contains disconnected diagrams.

Finally, the connected 2nd order expression for the non-hermitian effective property operator can be derived [17] (`scheme connected`):

$$O_{if} = \langle \tilde{\psi}_i^{\perp\perp} | \tilde{O} | \tilde{\psi}_f \rangle \cdot N_i N_f^{-1} \tag{4}$$

$$\tilde{O} \approx \left(O + T^\dagger O + OT + \frac{\{(T^\dagger)^2\}}{2} O + T^\dagger O T + O \frac{\{T^2\}}{2} - (T^\dagger T)_{cl} O \right)_{closed,conn} . \tag{5}$$

Expressions $\Omega^\dagger \Omega$ and $\Omega^\dagger O \Omega$ used in Eqs. (3) – (5) are evaluated by direct substitution of the FS-CC Ansatz $\Omega = \{e^T\}$ and have to be artificially truncated at some power of T (the truncation level is specified by the `approx` keyword). It is natural to use the same level of truncation for both these terms. Finally, the transformation properties with respect to the molecular point group operations of each specific property operator should be explicitly given (the `sym` keyword).

Syntax: `mdprop "<property-name>" [sym <irrep>] [approx <n> <m>] [scheme hermitian | non-hermitian | connected]`

`sym` symmetry of the property operator under consideration;

`approx` truncation level for the numerator $\Omega^\dagger O \Omega$ and the metric operator $\Omega^\dagger \Omega$:
`0` for the full neglect of terms containing cluster amplitudes,
`1` for the linear approximation,
`2` for the quadratic approximation (recommended option);

`scheme` particular form of the effective operator to be used:
`non-hermitian` – Eq. (1),
`hermitian` – Eq. (3),
`connected` – Eqs. (4) and (5) (recommended option).

Default: disabled.

Example: calculation of the whole set of possible dipole transition moments for a linear diatomic heteronuclear molecule (the $C_{\infty v}$ point group). Terms which are quadratic in cluster amplitudes are included (`approx 2 2`). The connected formulation is used. The list of instructions will be as follows:

```
mdprop "XDIPLEN" sym 1u+ approx 2 2 scheme connected
mdprop "XDIPLEN" sym 1u- approx 2 2 scheme connected
mdprop "YDIPLEN" sym 1u+ approx 2 2 scheme connected
mdprop "YDIPLEN" sym 1u- approx 2 2 scheme connected
mdprop "ZDIPLEN" sym 0u  approx 2 2 scheme connected
```

Matrix elements of some operators calculated by DIRAC can be purely imaginary even if the molecular point group allows one to use real arithmetics. If matrix elements of such an operator are needed, it is recommended to solve amplitude equations in real arithmetics, then switch *on the complex one using the* ***`arith complex`*** *instruction, restart the task from converged amplitudes and calculate property matrix elements of interest. For this reason we recommend to thoroughly check the diagnostics in the EXP-T output.*

The code for properties is currently experimental; we kindly ask you to report all failures and obviously incorrect results.

`mstdm` enable calculations of the model-space (MS) approximations to transition dipole moments (wavefunctions are represented by model vectors) [30, 31, 15]). To avoid huge outputs, use options `nroots` or `roots_cutoff` to restrict the set of target roots.

This option is equivalent to the `mdprop` instruction specified for all three directions of the dipole moment operator with the "`approx 0 0`" and "`scheme non-hermitian`" parameters.

Note that left model vectors are used in bra:

$$d_{if,\eta} = \langle \tilde{\psi}_i^{\perp\perp} | d_\eta | \tilde{\psi}_f \rangle \qquad \eta = x, y, z$$

and if the "hermitization" of the effective Hamiltonian matrix (specified by the `hermit` keyword) is disabled, the TDM matrix will be (normally slightly) non-Hermitian ($|d_{if}| \neq |d_{fi}|$).

This approach typically overestimates transition moments by 30-50% (see, for example, tests in [31, 32, 15]), hence we recommend using it only for semiquantitative estimates, e. g. the detection of the most intensive transitions and discerning between 0^+ and 0^- states in diatomic molecules (such $E1$-transitions are strictly forbidden).

Syntax: `mstdm`

Default: disabled.

`txtprop` enable calculations of expectation values and transition matrix elements of one-electron property operators whose matrix elements are specified in a text file. Property operator matrix elements are loaded from two separate formatted files. To avoid huge outputs, use options `nroots` and `roots_cutoff` to restrict the set of target roots. Other options are the same as for the `mdprop` keyword.

Syntax: `txtprop <string file_re> <string file_im> [sym <irrep>] [approx <n> <m>] [scheme hermitian | non-hermitian | connected]`

Default: disabled.

Example: atomic calculation of matrix elements of the electric field gradient (EFG):

```
txtprop efg_re_part.txt efg_im_part.txt sym 0g approx 2 2 scheme connected
```

(where the `efg_re_part.txt` and `efg_im_part.txt` files contain the real and imaginary parts of the EFG operator matrix, respectively).

The code for properties is currently experimental; we kindly ask you to report all failures and obviously incorrect results.

`natorb` calculation of model natural spinors (NS) and transition spinors (NTS) [6, 7].

Note: (transition) density matrices $d_{pq}^{IJ} = \langle \tilde{\psi}_I^{\perp\perp} | a_p^\dagger a_q | \tilde{\psi}_J \rangle$ are constructed with model vectors instead of true wavefunctions.

Syntax: `natorb [<rep>:<state> | <rep1:state1>-<rep2:state2>] ...`

Electronic states for which NS and NTS should be calculated are specified by the `<rep>:<state>` pair, where:

`<rep>` name of an irreducible representation (in square braces). See Appendix D for the list of irrep names used in EXP-T.

`<state>` sequential number of the state in this irreducible representation (1, 2, ...).

A pair of electronic states separated by a hyphen shows that NTS between these two states are to be calculated. You can ask EXP-T to calculate NS and NTS for arbitrary number of states (or pairs of states).

Default: calculation of NS and NTS is disabled.

Examples:

- NS for the ground ($^2\Sigma^+_{1/2}$) and the first excited ($^2\Sigma^+_{1/2}$) states of the RaCl molecule [33]:

  ```
  natorb [1/2+]:1 [1/2+]:2
  ```

- NTS for the $(1)0^+$ – $(2)0^+$ transition in the RbCs molecule [34]:

  ```
  natorb [0]:1-[0]:3
  ```

  (note that in the latter example the `[0]:2` state is excluded, since it is the $(1)0^-$ state which cannot be accessed by the direct transition from the ground state).

Iterative solution and convergence

`conv` specifies the requested convergence threshold (in amplitudes):

$$\left||t_K^{(n+1)}| - |t_K^{(n)}|\right| < \texttt{conv} \quad \forall K \quad \Rightarrow \text{converged}$$

The convergence threshold must be made more stringent for finite-difference calculations of expectation values of property operators.

Syntax: `conv <real thresh>`

Default: `1e-9`.

`crop` enables the CROP extrapolation technique to accelerate convergence [35, 36]. It is recommended to use the least possible number of last steps used for extrapolation: `crop 3`.

Syntax: `crop [ ( off || triples || <integer dim> ) ]`

Disable CROP: `crop off`

Enable CROP for triples: `crop triples`

Default: CROP technique is disabled.

It is highly recommended to enable CROP for triples if the CCSDT-n or CCSDT models are employed.

`damping` enable damping in the given FS sector. Amplitudes obtained at the n-th step will be mixed with amplitudes from the $(n-1)$-th step using the damping factor α:

$$t_K^{(n)'} = \alpha t_K^{(n-1)} + (1-\alpha) t_K^{(n)}$$

Damping can aid the convergence of equations in some non-pathological cases (when the lack of convergence is caused by amplitude oscillations rather than intruder states). Note that in the other cases damping can significantly slow down the convergence.

Syntax: `damping <H>h<P>p <integer last_step> <real factor>`

`<H>h<P>p`	FS sector in which these damping parameters will be applied;
`last_step`	at which iteration damping will be disabled;
`factor`	damping factor α (can be negative).

Default: disabled in all sectors.

Example: `damping 0h2p 100 0.5`

`diis` enables the DIIS extrapolation technique [37, 38].

Syntax: `diis [ ( off || triples || <integer dim> ) ]`

Disable DIIS: `diis off`

Enable DIIS for triples: `diis triples`

Default: `diis 10`

(DIIS is enabled for singles and doubles, disabled for triples).

It is highly recommended to enable DIIS for triples if the CCSDT-n or CCSDT models are employed.

`div_thresh` specifies the divergence threshold (in amplitudes). Solution of amplitude equation is diverged if for some t_K holds:

$$\left| |t_K^{(n+1)}| - |t_K^{(n)}| \right| > \texttt{div\_thresh} \quad \Rightarrow \text{diverged}$$

May be useful if few initial steps of the iterative solution are divergent. Note that this may also occur even in non-pathological cases.

Syntax: `div_thresh <real thresh>`

Default: `1e+3`.

`flush` indicates that cluster amplitudes must be flushed to disk every N iterations.

Normally cluster amplitudes are written to disk only if the convergence was reached. Use the `flush` instruction to simplify the restarting of your tasks. This option can be useful for very time-consuming calculations with poor convergence. It also allows one to change the convergence acceleration strategy if needed.

Syntax: `flush <integer N> iter`

Default: no flushing.

Example: `flush 5 iter`

`nroots` specifies the number of roots (electronic states) of interest in each irreducible representation (in the target Fock space sector). Model vectors analysis will be performed only for the `nroots` lowest states.

Syntax: `nroots <pairs irrep:number-of-roots>`

Default: all roots will be printed and analyzed.

Example: consider the electronic states of the RbCs molecule, corresponding to the first three dissociation limits (see, for example, [5, 34] and references therein). In the Hund's case *a* these electronic states are $X^1\Sigma^+$, $a^3\Sigma^+$, $b^3\Pi$, $A^1\Sigma^+$, $c^3\Sigma^+$, $B^1\Pi$. In the Hund's case *c* these states are classified according the projection of the total electronic angular momentum and can be rearranged as follows: (1-3)0^+, (1-3)0^-, (1-4)1 and (1)2 states. The names of irreps used in

EXP-T are given in Appendix D (the point group of RbCs is $\overline{C_{\infty v}}$). All output and analysis of model vectors will be restricted to these 11 states using the following instruction:

```
nroots [0]:6 [1+]:4 [1-]:4 [2+]:1 [2-]:1
```

(note that in DIRAC the 0^+ and 0^- states belong to the same irrep denoted as “0”).

`maxiter` maximum number of iterations allowed to solve the CC equations.

Syntax: `maxiter <integer max>`

Default: `maxiter 50`.

`reuse` indicates that sorted integrals and cluster amplitudes from the previous calculation should be reused; thus the calculation will be restarted. Cluster amplitudes read from disk will be used as an initial guess. If EXP-T fails to locate the files containing amplitudes, an initial guess will be obtained using MBPT(2).

Syntax: `reuse <list-of-arguments>`

Possible values of arguments:

`integrals` load one- and two-electron integrals (from a scratch directory);

`1-integrals` one-electron integrals only;

`2-integrals` two-electron integrals only;

`amplitudes` load amplitudes for all FS sectors occuring in the calculation;

`0h0p, 0h1p, 1h0p, 2h0p, 1h1p, 0h2p, 3h0p, 2h1p, 1h2p, 0h3p`
load amplitudes for the given Fock space sector only.

Default: no reuse

Note: files with converged amplitudes are flushed to the scratch directory after each calculation (see also Appendix C). To prevent the deletion of files with amplitudes or integrals it is necessary to use the `--no-clean` command-line option (see Section 3.3).

Example: `reuse integrals 0h0p 0h1p`

In the current version of EXP-T one cannot reuse integrals specific for the given sector. If molecular integrals were not previously sorted for some of the sectors, then all the integrals have to be re-sorted and the “`reuse integrals`” instruction cannot be used. This issue is planned to be fixed in future releases.

`roots_cutoff` specifies the upper energy bound for the electronic states to be analyzed and printed. Model vectors analysis and property calculations will be performed only for the states with energies lower than `roots_cutoff`. Allowed units of energy are eV, cm^{-1} or Hartrees.

Syntax: `roots_cutoff <real max_energy> ( ev || cm || au )`

Default: all roots will be printed and analyzed.

Example: `roots_cutoff 30000 cm`

`shift` sets the denominator shift parameters in the $n_h h\ n_p p$ Fock space sector.

Syntax:

```
shift <H>h<P>p <int n> <real shift_T1> <real shift_T2> [<real shift_T3>]
```

`n` attenuation parameter (Eqs. (6) – (9);

`shift_T1` shift value for the $T_1^{n_h h\ n_p p}$ cluster operator (singles). Must be omitted in the $2h0p$ and $0h2p$ FS sectors and in all sectors with three quasiparticles. It is recommended to set `shift_T1` = 0.0 in the $1h1p$ sector, unless you really understand the consequences of using a nonzero value. Note that the `shift_T1` value in this sector does not affect the resulting energies, it is needed only for property calculations;

`shift_T2` shift value for $T_2^{n_h h\ n_p p}$ (doubles). Must be omitted for the $3h0p$ and $0h3p$ sectors;

`shift_T3` shift value for $T_3^{n_h h\ n_p p}$ (triples). Can be used only for CC models accounting for triples in a non-perturbative way (CCSDT-n, CCSDT).

Default: $n = 3$, all shifts = 0.0.

Examples:

```
shift 0h1p 3 -0.3 -0.6 -0.9
shift 0h1p 3 -0.6
shift 1h1p 4 0.0 -0.5
```

`shift_type` indicates which formula for energy denominator shifts should be used [5, 9, 14].

`shift_type` is the same for all FS sectors. If `shift_type` is not equal to `none`, you must specify the shift parameters for at least one FS sector (see keyword `shift`; except for $0h0p$).

Syntax: `shift_type ( none || real || realimag || imag || taylor )`

Possible values:

`none` no shift

`real` real shift

$$D_K^{'} = D_K + S\left(\frac{S}{D_K + S}\right)^n \tag{6}$$

`realimag` real simulation of an imaginary shift [16]

$$D_K^{'} = D_K + \frac{S^2}{D_K}\left(\frac{S^2}{D_K^2 + S^2}\right)^n \tag{7}$$

`imag` imaginary shift [16]

$$D_K^{'} = D_K + iS\left(\frac{|S|}{|D_K + iS|}\right)^n \tag{8}$$

`taylor` extrapolated intermediate Hamiltonian-like shift [39]

$$D_K^{'} = (D_K + S)\frac{\left(1 - \frac{S}{D_K+S}\right)}{1 - \left(\frac{S}{D_K+S}\right)^{n+1}} \tag{9}$$

Default:

- if there are no shifts specified using the `shift` instruction:
  `shift_type none`
- otherwise (shifts are specified in some sectors):
  `shift_type realimag`

Shifts are never applied in the 0h0p sector!

`skip` indicates which sectors will be skipped in the FS-CC calculation. Can be useful in very costly calculations (like CCSDT) in order to avoid recalculation of the first steps in already converged sectors. To be used with the `reuse` option.

Syntax: `skip <list-of-sector-symbols>`

Default: disabled.

Example: `skip 0h0p 0h1p`

Interfaces

`integrals` sets paths to files containing transformed molecular integrals.

Syntax:
```
integrals <string "1-el int-s"> <string "2-el int-s">
[<string "properties int-s">]
```

Arguments:

- one-electron integrals file;
- two-electron integrals file;
- properties transformed integrals file (optional, is needed only for property calculations).

Default: `integrals MRCONEE MDCINT MDPROP`

Example: `integrals ../MRCONEE-Cinfv ../MDCINT-Cinfv /home/user/MDPROP`

`oneprop` imports one-electron property integrals from the `MDPROP` unformatted file. The one-electron operator $\hat{O}_1$ is added to the Fock operator at the integral sorting stage:

$$\hat{f}' = \hat{f} + \lambda \hat{O}_1$$

where λ is some perturbation parameter. EXP-T does not assume Kramers symmetry, so both $\mathcal{T}$-odd and $\mathcal{T}$-even operators are allowed.

Syntax: `oneprop <real L_re> <real L_im> <quoted-string property-name>`

`L_re, L_im` real and imaginary parts of the perturbation parameter λ;

`property-name` string identifier used to encode the property matrix in the `MDPROP` file (can be also found in the DIRAC output file).

Default: disabled.

Example:

```
oneprop 1e-5 0.0 "ZDIPLEN"
```

(an interaction of a system with an external electric field is added to the Fock operator).

`x2cmmf` since DIRAC's unformatted files do not contain any information about the relativistic Hamiltonian used, the `x2cmmf` flag **must** be used if the molecular-mean-field X2C Hamiltonian (X2Cmmf) was employed at the DIRAC stage.

Syntax: `x2cmmf`

Memory management

`compress` enables compression of all data written to disk. The LZ4 algorithm [40] is employed (it provides the fastest decompression to date). Compression can slow down calculations slightly (depends on the particular harddrive), but the disk space used can be reduced dramatically (up to 3 times).

Syntax: `compress`

Default: disabled (except for `disk_usage=4`)

`disk_usage` indicates which data should be stored on the disk. Note that very intensive disk usage will slow down calculations; however, disk usage is hardly avoidable in case of large tasks.

Syntax: `disk_usage <integer mode>`

Possible values of `<mode>`:

`0` all data is stored in RAM;

`1` tensors of rank ≥ 6 are stored on disk (T_3 amplitudes etc);

`2` tensors of rank ≥ 6 and $\langle pp||pp\rangle$ (4 virtual indices) two-electron integrals are stored on disk;

`3` tensors of rank ≥ 6, $\langle pp||pp\rangle$ and $\langle *p||pp\rangle$ (3 virtual indices) two-electron integrals are stored on disk;

`4` the same as `3`, + compression of all data. Is suitable for extremely large tasks only.

Default: `disk_usage 2`

($\langle pp||pp\rangle$ and tensors of rank ≥ 6 are stored on disk).

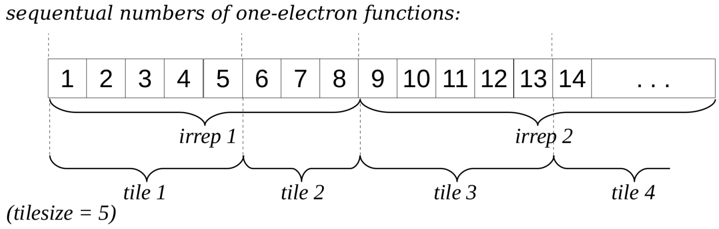
`memory` specifies the amount of memory that EXP-T can use for the job (just for the *dynamically* allocated space).

Syntax: `memory <real size> ( mb || gb )`

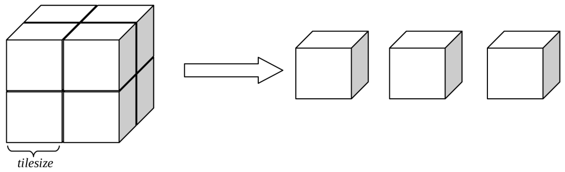
Min value: 10 Mb

Default: `memory 1 gb`

`tilesize` the whole set of one-electron functions (spinors or spin-orbitals) is divided into blocks: (1) by irrep; (2) then into sub-blocks with size not exceeding `tilesize` (actually the same as "tiles" in NWChem-TCE):



The N-dimensional arrays of integrals and amplitudes (tensors) also turn out to be divided into blocks containing $\sim$ (`tilesize`)^N elements:



`tilesize` must be large enough to place at least one N-dimensional array of size (`tilesize`)^N in RAM. We recommend using as large `tilesize` as possible on your machine, otherwise overheads for the memory management can become enormous.

Syntax: `tilesize <integer size>`

Default: `tilesize 100`

Parallel execution

`nthreads` number of OpenMP threads to be used for parallel calculations. Note that the scaling with respect to the number of threads is much better for large tasks (>400 spinors).

Syntax: `nthreads <integer n_omp_threads>`

Default: `nthreads 1` (sequential execution).

At the moment parallelization is not used at the integral sorting stage, hence OpenMP can accelerate only the amplitude equations solution stage.

It is recommended to always enable OpenMP parallelization in calculations accounting for triple excitations.

`openmp` the same as `nthreads`.

`openmp_algorithm` the approach to parallelization. Two parallelization algorithms are implemented in EXP-T:

`external` suits well for the problems with high granularity of tensors; it is recommended for all point groups except for C_1.

`internal` is recommended for the C_1 symmetry only.

Syntax: `openmp_algorithm ( external | internal )`

Default: `external`

`cuda` enables parallel calculations on an NVIDIA GPU (currently, only one GPU card is supported in calculations). Requires `nthreads 1` (the OpenMP+CUDA hybrid model is not implemented yet).

Syntax: `cuda`

Default: CUDA is disabled.

The CUDA code is outdated, to be used with care. This issue will be fixed in future releases.

5 Running typical calculations

The input and output files discussed in this section can be found in the **`examples`** *directory of the EXP-T release.*

5.1 Ground state energy with CCSD: the CO molecule

For input/output files see **`examples/CO_ccsd`**.

Consider the calculation of the CCSD correlation energy of the CO molecule in the cc-pVDZ [41] basis set.

DIRAC input files:

(`CO.mol`)

```
DIRAC
CO molecule, R = 2.132 bohrs
symmetry C1
C   2    0
        6.    1
C        0.0      0.0      0.0
LARGE BASIS cc-pVDZ
        8.    1
O        0.0      0.0      2.132
LARGE BASIS cc-pVDZ
FINISH
```

(`moltra.inp`)

```
**DIRAC
.TITLE
CO molecule
.WAVE FUNCTION
.4INDEX
**WAVE FUNCTION
.SCF
*SCF
.CLOSED
14
.ERGCNV
1E-12
**MOLTRA
.ACTIVE
all
*END OF INPUT
```

Use the following command to run DIRAC and obtain transformed molecular integrals:

```
pam --inp=moltra --mol=CO --get="MRCONEE MDCINT" --noarch
```

Note that the `--get` option is used so that `pam` exports the transformed molecular integrals (stored in the `MRCONEE` and `MDCINT` files) from DIRAC's scratch directory to the working directory. The `-noarch` option is used to avoid creation of a `.tgz` archive.

In some cases, it can be convenient to rename the files containing the integrals (this is not mandatory). For example,

```
mv MRCONEE MRCONEE-C1
mv MDCINT MDCINT-C1
```

Now the EXP-T input file `ccsd.inp` is set up:

```
# CO molecule, CCSD/cc-pVDZ
# non-relativistic, symmetry C1

# task title (of type "string")
title "CO/CCSD/cc-pVDZ"

# print level, default medium
print medium

# max number of iterations
maxiter 30

# convergence threshold (by cluster amplitudes)
conv 1e-9

# target Fock space sector
sector 0h0p

# CC model: ccsd, ccsdt-1, etc
model ccsd

# import integrals from
integrals MRCONEE-C1 MDCINT-C1
```

Running EXP-T:

```
expt.x ccsd.inp | tee ccsd.out
```

The string with the correlation energy can be found at the end of the output. Note that by default EXP-T flushes all output simply to `stdout`.

```
          SCF reference energy =    -112.820480227130
       CCSD correlation energy =      -0.298117912040
             Total CCSD energy =    -113.118598139170
```

5.2 Simple atomic FS-CCSD calculation: the Pb atom

For input/output files see `examples/Pb_fs_ccsd`.

We now turn to simple atomic relativistic FS-CCSD calculations. Since wave functions of most low-lying electronic states of one-electron atoms and atomic ions are effectively single-reference, it is not very interesting to apply the multireference methodology to such cases. In contrast, electronic states of the "two-electron" atoms can hardly be simulated within any other approaches except those based on the multireference wave functions. Consider for example low-lying states of the Pb atom corresponding to the $6s^2 6p^2$ configuration. The closed shell ground state $6s^2$ of Pb^{2+} can be considered as the Fermi vacuum, and one should select its $6p$-spinors as the active ones (overall 6 active spinors).

Pb atom is a purely relativistic system, and one should use either the four-component Hamiltonian or the relativistic pseudopotential (RPP) approach. The latter allows one to incorporate bulk of the Breit interaction [42] and (possibly) QED corrections [43, 44, 10]. Here the semilocal (valence) version of the N. S. Mosyagin's RPP for Pb is used (see [45] and references therein).

Input file for the FS-CCSD calculation will be as follows:

```
sector 0h2p
nactp 6
maxiter 500
memory 110 gb
conv 1e-7
disk_usage 0
nthreads 8
model ccsd
```

After the FS-CCSD calculation is finished, one can compare the calculated FS-CCSD excitation energies of Pb with their experimentally measured counterparts (7819, 10650, 21458 and 29467 cm^{-1}, respectively [46]):

```
 Level  Re(eigenvalue)    . . .    Rel eigv, eV  Rel eigv, cm-1  deg  symmetry
 ----------------------- . . . ----------------------------------------------
@    1   -0.8258538771    . . .    0.0000000000        0.000000   1   0g
@    2   -0.7916687782    . . .    0.9302239316     7502.761992   3   0g 1g+ 1g-
@    3   -0.7784232250    . . .    1.2906537953    10409.824894   5   0g 1g+ 1g- 2g+ 2g-
@    4   -0.7298818976    . . .    2.6115306047    21063.414836   5   0g 1g+ 1g- 2g+ 2g-
@    5   -0.6892150215    . . .    3.7181326781    29988.762480   1   0g
```

One can also find in the output file the calculated values of sequential ionization energies corresponding to the $Pb^+ \to Pb^{2+}$ and $Pb^0 \to Pb^{2+}$ processes, respectively:

```
 Ionization potential 0h1p -> 0h0p =  . . .  =  15.0036 eV =  121012.35 cm^-1
 Ionization potential 0h2p -> 0h1p =  . . .  =  22.4726 eV =  181253.98 cm^-1
```

The derived first ionization potential of the neutral lead atom would be:

$$\mathrm{IP}_1 = E(\mathrm{Pb}^+ \to \mathrm{Pb}^{2+}) - E(\mathrm{Pb}^0 \to \mathrm{Pb}^{2+}) = 181254 - 121012 = 60242 \text{ cm}^{-1}$$

(the experimental value is 59819 cm^{-1}).

Note that the obtained results are consistent with an observation that the typical error of the FS-CCSD model is of order 300–400 cm^{-1}. The remaining error can be significantly reduced by inclusion of triple excitations [12].

5.3 Electronic states and potential energy curves of a diatomic molecule: HgH

For input/output files see `examples/HgH_fs_ccsd`*.*

Let us consider a very simple (but typical) example of FS-CCSD calculation of a system of electronic states and corresponding potential energy curves of a heavy diatomic molecule of mercury hydride (HgH). Some time ago this molecule was considered as a suitable system for experimental studies of different parity non-conservation effects (see, for example, papers [47, 48, 49, 50, 51, 52] and references therein).

The exhaustive picture of the HgH electronic states was obtained using the MRCI method in [52]; potential energy curves calculated in this paper for 4 lowest-lying electronic states are shown in dashed

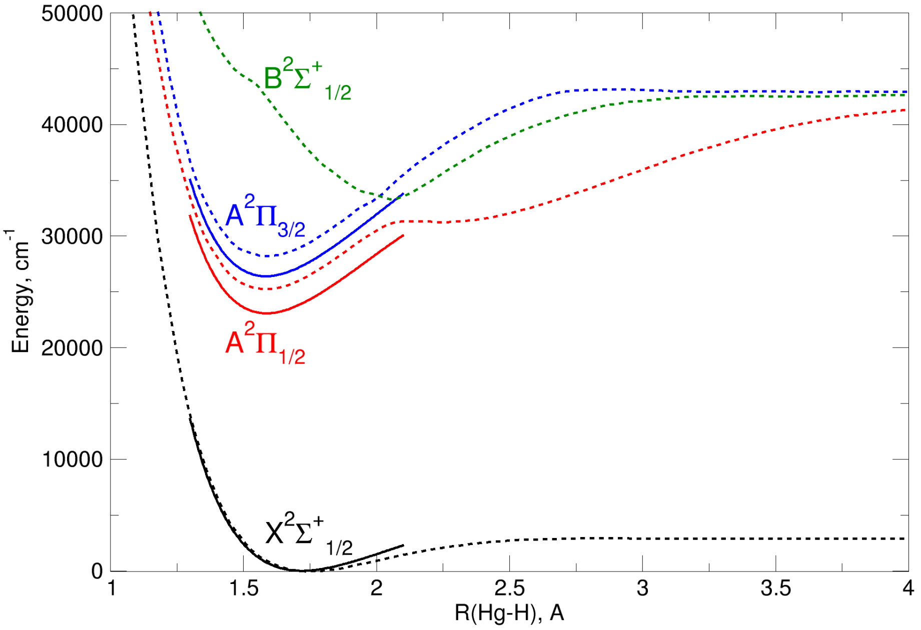


Figure 1: *Potential energy curves of low-lying electronic states of the HgH molecule calculated using the FS-CCSD method (solid lines; see text) and multireference configuration interaction [52] (dashed lines). Avoided crossing of the $A^2\Sigma^+_{1/2}$ and $B^2\Sigma^+_{1/2}$ states cannot be described within the model space comprising only 6 active spinors.*

on Fig. 1. We will try to reproduce three states ($X^2\Sigma^+_{1/2}$, $A^2\Pi_{1/2}$, $A^2\Pi_{3/2}$), thus the minimal active space describing these target states comprises 6 virtual spinors of HgH$^+$. The basis set employed here was borrowed from Ref. [47] and cropped to [$4s5d3d$] in order to drastically reduce the computational cost; the cc-pVDZ basis set [41] was placed on the hydrogen atom. Thus one shouldn't expect excellent agreement with either experimental data or high-quality calculations reported previously, but all the basic features of the considered problem are still preserved. 60 core electrons of Hg were substituted by the semi-local version of the small-core shape-consistent relativistic pseudopotential of N. S. Mosyagin et al [53].

We first calculate vertical excitation energies of HgH at the internuclear distance r(Hg-H) = 1.74 Å which is close to the experimental one. The first stage implying the relativistic Hartree-Fock calculation and transformation of molecular integrals, is performed using DIRAC:

```
pam --inp=moltra --mol=HgH --noarch --get="MRCONEE MDCINT"
```

Molecular geometry, basis set and RPP are specified in the `HgH.mol` file, while options for SCF and four-index transformation are given in `moltra.inp`.

It can be more convenient to use the special NWChem-like format to specify geometry, basis set and RPP data instead of using DIRAC `.mol` files (see Section 6.3 for details). In this case just convert the `HgH.exp` file attached to this example to the `.mol` format using the following commands:

```
expt2pam.x HgH.exp
mv dirac.mol HgH.mol
```

Input file for FS-CCSD calculation using EXP-T will be as follows (`ccsd.inp`):

```
# Test:
# (1) HgH molecule, FSCC scheme: HgH+ -> HgH
# (2) hamiltionian: 2-comp gatchina ECP
# (3) symmetry Cinfv
# (4) sector (0h,1p)
```

```
title "Test HgH+ -> HgH/gatchina ECP/Cinfv"
sector 0h1p
nactp 6
conv 1e-9
maxiter 200
diis 100
```

Running EXP-T:

```
expt.x ccsd.inp > ccsd.out
```

Information about electronic states is summarized in the end of an output file and is marked by the '@' symbol in the first position of a line:

```
 Level  Re(eigenvalue)  . .                Abs energy  . .  Rel eigv, cm-1  deg  symmetry
 ----------------------  . .  ------------------------  . .  ------------------------------
@    1   -0.2843267762  . .   -153.90380203063062936  . .        0.000000   2   1/2+ 1/2-
@    2   -0.1742398284  . .   -153.79371508284035031  . .    24161.292284   2   1/2+ 1/2-
@    3   -0.1588816394  . .   -153.77835689385861428  . .    27532.025149   2   3/2+ 3/2-
```

Vertical excitation energies are given in the column named "Rel eigv, cm-1". Projections of the total angular momentum Ω on the z-axis are given in the last column "symmetry". In this particular case there's not much point in comparing vertical excitation energies with their experimental counterparts since equilibrium distances in excited states are significantly displaced with respect to r_e in the ground state. However, by varying the Hg–H internuclear separation and collecting absolute energies given in a. u. (the "Abs energy" column) one can obtain potential energy curves in the range $r \in [1.3, 2.1]$ Å (see Fig. 1). Note that for larger r there is an avoided crossing of the $A^2\Sigma^+_{1/2}$ and $B^2\Sigma^+_{1/2}$ states. The latter state cannot be obtained within the model space comprising only 6 active spinors considered here.

To extract spectroscopic constants (r_e, T_e, ω_e, ...) one should solve the rotational-vibrational problem for each of the curves. It is convenient to use the `expt_diatomic.x` program (Section 6.5) or any other software (we also recommend the `VIBROT` code [54]) to do it. For example, the input file (`vib_rot_X2Sigma+.inp`) for `expt_diatomic.x` needed to calculate spectroscopic constants of the $X^2\Sigma^+_{1/2}$ state will be as follows (the format is self-explanatory):

```
#
# ground state of HgH
# X2Sigma+_1/2
#

# Hg-202 [a.m.u.]
masses 1.00782503223 201.97064340

rotlevels 0 0
viblevels 0 100

# numerov solver is used by default
grid_size 1000

potential angstrom atomic
1.30    -153.84194909
1.40    -153.87510640
```

```
1.50    -153.89291224
1.60    -153.90122623
1.70    -153.90380335
1.80    -153.90305815
1.90    -153.90052604
2.00    -153.89716570
2.10    -153.89357117
end
```

Running `expt_diatomic.x`:

```
expt_diatomic.x vib_rot_X2Sigma+.inp | tee vib_rot_X2Sigma+.out
```

Output file contains analysis of spectroscopic constant in three different approximations, (a) harmonic oscillator, (b) Morse oscillator, (c) exact numerical solution of the radial Schrödinger equation:

```
 > equilibrium parameters

   e(min) = -153.903855470153 a.u. = -33777991.944687 cm^-1
   re     = 3.2440 a.u. = 1.7167 angstrom

 > analysis in the harmonic oscillator + rigid rotor approximation

   we = 1446.5254 cm^-1
   Be = 5.704258 cm^-1
   De = 0.000355 cm^-1

 > analysis in the morse approximation

   (parameters are estimated at the minimum point only)
   alpha = 1.2462 bohr^-1
   De    = 0.025565 a.u. = 5611 cm^-1
   we    = 1446.53 cm^-1
   wexe  = 93.23 cm^-1
   vmax  = 7

 > rovibrational energy levels

                        energy           < r >         < B_v >         <prop>
     J   v             (cm^-1)           (Ang)         (cm^-1)

@    0   0            730.0496        1.746630        5.570576        0.000000
@    0   1           2249.5311        1.764043        5.560544        0.000000
```

Potential energy curves and spectroscopic constants can be further refined by:

- using of high-quality basis sets designed specifically for the problem under consideration (using the ANO approach [55], for example);
- using of full (non-local) versions of generalized relativistic pseudopotentials [56];
- extending the model space, possibly involving the intermediate Hamiltonian approach to bypass the intruder state problem. In principle, this can help to describe the avoided crossing $A^2\Sigma^+_{1/2} \sim B^2\Sigma^+_{1/2}$;
- accounting for triple excitations at some level of theory (even the FS-CCSDT-1 model in the $0h1p$ sector would be useful [12]).

5.4 Intermediate Hamiltonian for incomplete main model spaces: the C atom

For input/output files see `examples/C_atom_ih-imms`*.*

Intermediate Hamiltonian formulation of the FS-CC method allows one to mitigate intruder state problem. The recent formulation for incomplete main model spaces [10] works in a nearly black-box fashion.

Let us consider low-lying electronic states of the C atom. Using the NIST ASD database [57] we find that for the states below 80000 cm^{-1} the following configurations dominate in wave functions:

- $2s^2 2p^2$;
- $2s^1 2p^3$;
- $2s^2 2p^1 3s^1$;
- $2s^2 2p^1 3p^1$;
- $2s^2 2p^1 3d^1$;
- $2s^2 2p^1 4s^1$.

Within the Fock space coupled cluster formalism, electronic states dominated by the $2s^1 2p^3$ configuration (e. g. the $^5S_2^o$ state, 33735 cm^{-1}) can be accessed only in $1h3p$ or $0h4p$ sectors. Implementation of the corresponding FS-CC models is not available to date, thus we will not consider these states in the further discussion. In contrast, other states of the carbon atom with closed $2s^2$ subshell can be easily obtained in the $0h2p$ Fock space sector provided that the $2s^2$ state of the C^{2+} ion is considered as the Fermi vacuum. For the sake of simplicity we will not also consider electronic states involving shells with principal quantum number $n = 4$ (the generalization of the IH-IMMS procedure to such states is straightforward).

At this point we have several useful observations:

- active space should comprise at least $2p$, $3s$, $3p$, $3d$ spinors of the C^{2+} ion. These active spinors are needed to describe the manifold of electronic states below 80000 cm^{-1};
- only the $2p$ shell can be doubly occupied in some target electronic states ($2s^2 2p^2$);
- all target electronic states involve configurations with at least one $2p$-electron;
- we can expect that electronic states dominated by the $3s^2$, $3s^1 3p^1$, etc configurations will be very high in energy (and even lie well above the IP = 90820 cm^{-1}). These states arise inevitably in the $0h2p$ sector since the "classical" FS-CC is formulated for quasicomplete model spaces. Moreover, these *intruder states* will destroy convergence of amplitude equations;
- the linear span of these "malicious" model determinants can be considered as the *intermediate space*;
- the *main model space* should include only determinants belonging to the $2p^2$, $2p^1 3s^1$, $2p^1 3p^1$, $2p^1 3d^1$ configurations. These determinants will be dominating in the model vectors representing target electronic states of the carbon atom.

We start actual calculation with the molecular integral transformation in DIRAC:
(`C.mol`)

```
C atom
doubly augmented triple-zeta basis set (d-aug-cc-pVTZ)
symmetry Dinfh
C   1
```

```
        6.    1
C    0.0000000000    0.0000000000    0.0000000000
LARGE EXPLICIT    4    2    2    1    1
# s functions
F 10  2
       8236.00000000    0.00053100   -0.00011300
       1235.00000000    0.00410800   -0.00087800
        280.80000000    0.02108700   -0.00454000
         79.27000000    0.08185300   -0.01813300
         25.59000000    0.23481700   -0.05576000
          8.99700000    0.43440100   -0.12689500
          3.31900000    0.34612900   -0.17035200
          0.90590000    0.03937800    0.14038200
          0.36430000   -0.00898300    0.59868400
          0.12850000    0.00238500    0.39538900
F  4  0
          0.90590000
          0.12850000
          0.04402000
          0.01510000
# p functions
F  5  1
         18.71000000    0.01403100
          4.13300000    0.08686600
          1.20000000    0.29021600
          0.38270000    0.50100800
          0.12090000    0.34340600
F  4  0
          0.38270000
          0.12090000
          0.03569000
          0.01050000
# d functions
F  4  0
          1.09700000
          0.31800000
          0.10000000
          0.03140000
# f functions
F  3  0
          0.76100000
          0.26800000
          0.09440000
FINISH
```

(moltra.inp)

```
**DIRAC
.TITLE
Carbon atom
.4INDEX
```

```
.WAVE FUNCTION
**HAMILTONIAN
.X2Cmmf
.GAUNT
**WAVE FUNCTION
.SCF
*SCF
.CLOSED
4 0
.ERGCNV
1E-12
**MOLTRA
.SCHEME
 4
.ACTIVE
all
*END OF INPUT
```

Note that the `.X2Cmmf` keyword is used, thus it also must be used in the EXP-T input file.

Command line:

```
pam --inp=moltra --mol=C --noarch --get="MRCONEE MDCINT MDPROP"
```

Let us now look at the DIRAC output. We find that one-electron energies of our active spinors are (in a. u.):

$$\varepsilon_{2p} \sim -0.86$$
$$\varepsilon_{3s} \sim -0.36$$
$$\varepsilon_{3p} \sim -0.28$$
$$\varepsilon_{3d} \sim -0.23$$

We split these active spinors into two subsets. Spinors from the first subset can be doubly occupied in the main model space determinants, and spinors from the second subset can be only singly occupied. In some cases it can be useful to define the third subset of spinors which must be unoccupied in all main determinants. This procedure resembles that used to define an active space in the RASSCF method. For the C atom these subsets of active spinors are:

1. $2p$;

2. $3s$, $3p$, $3d$.

In terms of these two subspaces of active spinors the main model space determinants can be defined using the following directives inside the `ih_imms` block in the EXP-T input file:

- `main_occ 2 0`: determinants in which both electrons occupy spinors from the first subspace, the $2p^2$ configuration;

- `main_occ 1 1`: one electron occupy a spinor from the first subspace ($2p$), the other one occupy a spinor from the second subspace ($3s$, $3p$ or $3d$). These determinants correspond to the $2p^1 3s^1$, $2p^1 3p^1$ and $2p^1 3d^1$ configurations.

We will rely on the automatic determination of the `frontier_energy` parameter (by default, the uppermost energy of main model space determinants is used). Finally, the EXP-T input file will be as follows:

```
# relativistic CC model
sector 0h2p
model ccsd
nactp 24
x2cmmf

# intermediate Hamiltonian parameters.
# spinor subspaces are defined via energy ranges (in a. u.)
ih_imms
  sectors 0h2p
  subspace energy -1.0 -0.5  # 2p spinors
  subspace energy -0.5 -0.2  # 3s, 3p, 3d spinors
  main_occ 2 0               # 2p^2 determinants
  main_occ 1 1               # 2p3s, 2p3p, 2p3d determinants
end

# memory
memory 4 gb
disk_usage 0

# convergence
maxiter 300
diis 300
conv 1e-9

# print compositions of model vectors in terms of
# effective occupation numbers of active spinors
print "eff config"
```

Let us take a look at the table with calculated energy levels in the 0h2p sector in the EXP-T output file:

```
 Level  Re(eigenvalue) ...Rel eigv, cm-1  % main  deg  symmetry
 ----------------------...-------------------------------------
@    1   -1.2955562077 ...      0.000000   100.0   1   0g
@    2   -1.2954943219 ...     13.582372   100.0   3   0g 1g+ 1g-
@    3   -1.2953712703 ...     40.589069   100.0   5   0g 1g+ 1g- 2g+ 2g-
@    4   -1.2503086362 ...   9930.694080    99.9   5   0g 1g+ 1g- 2g+ 2g-
@    5   -1.1923569864 ...  22649.611052    99.8   1   0g
@    6   -1.0208724411 ...  60286.118415    99.9   1   0u
@    7   -1.0207910577 ...  60303.980008    99.9   3   0u 1u+ 1u-
@    8   -1.0206248742 ...  60340.453071    99.9   5   0u 1u+ 1u- 2u+ 2u-
@    9   -1.0089659280 ...  62899.295993    99.9   3   0u 1u+ 1u-
@   10   -0.9902128248 ...  67015.126416   100.0   3   0g 1g+ 1g-
@   11   -0.9824896577 ...  68710.165670   100.0   3   0g 1g+ 1g-
@   12   -0.9824017917 ...  68729.450018   100.0   5   0g 1g+ 1g- 2g+ 2g-
@   13   -0.9822669729 ...  68759.039338   100.0   7   0g 1g+ 1g- 2g+ 2g- 3g+ 3g-
@   14   -0.9751409822 ...  70323.013514   100.0   3   0g 1g+ 1g-
@   15   -0.9723223910 ...  70941.622775   100.0   1   0g
@   16   -0.9722734205 ...  70952.370551   100.0   3   0g 1g+ 1g-
@   17   -0.9721863073 ...  70971.489684   100.0   5   0g 1g+ 1g- 2g+ 2g-
@   18   -0.9616074654 ...  73293.277111    99.8   5   0g 1g+ 1g- 2g+ 2g-
@   19   -0.9497229655 ...  75901.623363    99.9   1   0g
@   20   -0.9322863070 ...  79728.527553   100.0   5   0u  1u+  1u-  2u+  2u-
@   21   -0.9307044048 ...  80075.714943   100.0   5   0u  1u+  1u-  2u+  2u-
@   22   -0.9306318565 ...  80091.637473   100.0   7   0u  1u+  1u-  2u+  2u-  3u+  3u-
@   23   -0.9305012392 ...  80120.304651   100.0   9   0u  1u+  1u-  2u+  2u-  3u+  3u-  ...
@   24   -0.9303011549 ...  80164.218067   100.0   3   0u  1u+  1u-
@   25   -0.9302612163 ...  80172.983578   100.0   5   0u  1u+  1u-  2u+  2u-
@   26   -0.9302131715 ...  80183.528205   100.0   7   0u  1u+  1u-  2u+  2u-  3u+  3u-
@   27   -0.9264080659 ...  81018.652356   100.0   7   0u  1u+  1u-  2u+  2u-  3u+  3u-
@   28   -0.9244722663 ...  81443.511260   100.0   5   0u  1u+  1u-  2u+  2u-
@   29   -0.9243950805 ...  81460.451580   100.0   3   0u  1u+  1u-
@   30   -0.9243547369 ...  81469.305985   100.0   1   0u
@   31   -0.9218925156 ...  82009.701090    99.9   3   0u  1u+  1u-
@   32   -0.5545617414 ... 162629.487352     9.6   1   0g
@   33   -0.5088229529 ... 172667.991098     0.0   1   0u
@   34   -0.5088105565 ... 172670.711776     0.0   3   0u  1u+  1u-
@   35   -0.5087857581 ... 172676.154402     0.0   5   0u  1u+  1u-  2u+  2u-
 ...
 Ionization potential 0h2p -> 0h1p = ... =  10.9819 eV =   88575.16 cm^-1
 Ionization potential 0h1p -> 0h0p = ... =  24.2720 eV =  195766.56 cm^-1
```

The column named "`% main`" contains total weights of main model space determinants in model vectors describing the given electronic states. It can be seen that these weights are very close to 100% for all target (or main) electronic states. Intermediate states (with weighs of main model space determinants not exceeding several percents) are also obtained as a by-product. However, they can hardly be considered as physical ones (compare their energies with the $\text{IP}_1 \sim 90000$ cm^{-1}). The greater the weight of main model space determinants, the less the dependence on the parameters of the applied denominator shifting scheme.

It is worth noting that in some cases convergence of amplitude equations can be achieved only by the introduction of additional active spinors, which are unoccupied in all main model space determinants. These spinors serves as buffer ones. In calculations accounting for triples such an extension of active space can result in a dramatic increase of computational cost. So it is recommended to thoroughly choose an upper bound of an active space to make it as small as possible. However, this is not a problem for calculations employing the CCSD model only. Introduction of additional buffer states also greatly improves stability of energy levels with respect to the shifting parameters (see Figures 3 and 4 in [10] for the case of electronic states of the Lu^+ ion).

5.5 Finite-field transition dipole moments calculations: the Rb atom

For input/output files see `examples/Rb_atom_TDMs_finite_field`.

See [31, 58, 3] for detailed discussions on the finite field (FF) technique for transition dipole moments. The algorithm of a typical FF calculation is as follows:

- solve the SCF problem for the non-perturbed Hamiltonian (no external fields) and obtain one-electron functions (spinors);

- transform SCF molecular integrals to the basis of "non-perturbed" one-electron functions;

- perform two FS-CC calculations with these integrals at external fields strengths $-F$ and $+F$;

- estimate TDMs or other off-diagonal matrix elements using the finite-difference-type formula:

$$\boldsymbol{d}^{FF}_{\eta,if} = (E_i - E_f) \lim_{\substack{F_\eta \to 0 \\ F_{\eta'}=0, \eta' \neq \eta}} \frac{\langle \tilde{\psi}_i^{\perp\perp}(-F_\eta) | \tilde{\psi}_f(+F_\eta) \rangle}{2F_\eta} \quad (\eta = x, y, z), \tag{10}$$

where $\tilde{\psi}_i^{\perp\perp}$ and $\tilde{\psi}_f$ are the left and right model vectors corresponding to the **i**nitial and **f**inal electronic states, respectively.

Consider the FF calculation of TDMs for the $^2S_{1/2} \leftrightarrow {}^2P^o_{1/2}$ and $^2S_{1/2} \leftrightarrow {}^2P^o_{3/2}$ transitions in the Rb atom. These states can be obtained in the $0h1p$ sector starting from the Rb$^+$ closed shell state:

$$\text{Rb}^+ \ (\text{0h0p}) \quad \rightarrow \quad \text{Rb}^0 \ (\text{0h1p})$$

The simplest active space required to describe the low-lying $^2S_{1/2}$, $^2P^o_{1/2}$ and $^2P^o_{3/2}$ states of Rb comprised $5s$- and $5p$-spinors of Rb$^+$ (overall 8 spinors, or 4 Kramers pairs). The basis set and relativistic pseudopotential used here were taken from [58] and reduced to $[7s7p5d3f2g]$.

Firstly, we perform the Hartree-Fock calculation of Rb$^+$ and subsequent transformation of molecular integrals. Note that transformed integrals over one-electron property operators (in this case, the dipole moment operator) are stored in the `MDPROP` unformatted file which should be copied from DIRAC scratch to the working directory:

```
pam --noarch --inp=moltra.inp --mol=Rb.mol --get="MRCONEE MDCINT MDPROP"
```

Other integrals are saved to the `MRCONEE` (one-electron Fock) and `MDCINT` (two-electron Coulomb) unformatted files. Note that the `.PRPTRA` option as well as the `**PROPERTIES/.DIPOLE` specification must be explicitly specified in the `moltra.inp` input file:

```
**MOLTRA
.PRPTRA
.SCHEME
4
.ACTIVE
energy -100.0 20.0 0.5
**PROPERTIES
.DIPOLE
```

These lines are used to ask DIRAC to calculate and transform dipole moment integrals (these integrals are used later to construct the perturbation contribution to the Fock matrix due to the external field). Also note that the external uniform electric field lowers spherical symmetry of the atom to $C_{\infty v}$.

Now we are ready to run an FS-CC calculation. At this step we use the `oneprop` directive to add the contribution of the an external field $\boldsymbol{F}$ contribution to the Fock operator:

$$f'_{pq} = f_{pq} + (d_x)_{pq}F_x + (d_y)_{pq}F_y + (d_z)_{pq}F_z$$

Here we use only the d_z component, stored in the `MDPROP` file in the `ZDIPLEN` record. The operator which is added to the Fock matrix must be fully symmetric. The first EXP-T input file (for the field strength $-F$):

```
# title for the task (of type "string")
title "Rb atom -- relativistic EA-CCSD calculation"

# max number of iterations
maxiter 50
```

```
# convergence threshold (by cluster amplitudes)
conv 1e-9

# target Fock space sector
sector 0h1p

# active space specification: 8 lowest virtual spinors (5s,5p1/2,5p3/2)
nactp 8

# add the external field contribution to the Fock matrix:
# Fock = Fock - d_z * F_z
# (F_z = - 1e-5)
oneprop -1e-5 0.0 "ZDIPLEN"
```

For the other field strength $+F$ it is convenient to reuse two-electron integrals, which have already been sorted (the `reuse` option). One-electron (Fock) integrals depend on the finite perturbation parameter and thus have to be recomputed. Cluster amplitudes converged at $-F$ can also serve as a good initial guess for the calculation at $+F$. This will lead to significant computational savings. Use the `--no-clean` option to prevent the deletion of files containing sorted two-electron integrals and converged amplitudes.

The second EXP-T input file (for the field strength $+F$):

```
title "Rb atom -- relativistic EA-CCSD calculation"
maxiter 50
conv 1e-9
sector 0h1p
nactp 8

reuse 2-integrals amplitudes
oneprop +1e-5 0.0 "ZDIPLEN"
```

Run EXP-T and save formatted files `HEFF` with effective Hamiltonian matrices to the working directory:

```
expt.x --no-clean ccsd_F-.inp | tee ccsd_F-.out
mv scratch/HEFF HEFF1
expt.x --no-clean ccsd_F+.inp | tee ccsd_F+.out
mv scratch/HEFF HEFF2
```

The electronic spectrum is given at the end of the EXP-T output file. Corresponding lines begin with the `@` symbol:

```
 Level  Re(eigenvalue)  . . .Rel eigv, cm-1  deg  symmetry
 ---------------------- . . .------------------------------
@    1   -0.1531234146  . . .      0.000000   2   1/2+ 1/2-
@    2   -0.0959179975  . . .  12555.137832   2   1/2+ 1/2-
@    3   -0.0948635671  . . .  12786.558570   2   1/2+ 1/2-
@    4   -0.0948635506  . . .  12786.562182   2   3/2+ 3/2-
```

(compare with experimental values 12578.950 cm^{-1} and 12816.545 cm^{-1} [46]; note that the $^2P^o_{3/2}$ state is split in the external electric field).

Finally, we turn to the calculation of transition moments. The input file for the `heffman.x` utility is as follows (see Section 6.1 for details and explanation):

```
file: HEFF1
file: HEFF2
sector: 0h1p
rep: 1
step: 0.00002
print: 1
```

Run `heffman.x`:

```
heffman.x < ff_tdm.inp | tee ff_tdm.out
```

Transition moments and Einstein coefficients can be found in the end of the output file `ff_tdm.out`:

```
 transition  energy, cm^-1  . . .    |d|^2       |d|        osc str   A,10^6 s-1
. . .
   1  ->  2     12555.138   . . .  2.627905  1.62108158    0.100220   10.537579
   1  ->  3     12786.559   . . .  5.173064  2.27443710    0.200921   21.911672
   2  ->  3       231.421   . . .  0.000000  0.00001167    0.000000    0.000000
```

Note: this is only the $|d_z|$ component of the full transition moment ***d****. It should be mentioned that the small, but non-zero value of $|d|$ in the last line is an artifact of the FF approach.*

Taking into account the degeneracy of the $^2P^o_{3/2}$ state and the spherical symmetry of the Rb atom (one can use the Wigner-Eckart theorem to restore the other components of $\boldsymbol{d}$), the full transition moments $|d|$ and corresponding Einstein coefficients can be calculated:

$$A(^2S_{1/2} \leftarrow {}^2P^o_{1/2}) = 31.6 \text{ s}^{-1}$$
$$A(^2S_{1/2} \leftarrow {}^2P^o_{3/2}) = 32.9 \text{ s}^{-1}$$

(compare with experimental values 34.0 s^{-1} and 37.0 s^{-1} [46]).

Results can be improved significantly by extending the basis set and active space [58].

5.6 Finite-order approach to transition dipole moment calculations: the Rb atom

For input/output files see `examples/Rb_atom_TDMs_finite_order`.

The finite-order approach to transition moment calculations consists in the substitution of the FS-CC wave functions $|\psi_i\rangle = \{e^T\}\,|\tilde{\psi}_i\rangle$ and $|\psi_f\rangle = \{e^T\}\,|\tilde{\psi}_f\rangle$ into the expression for the matrix element $O_{if} = \langle\psi_i|O|\psi_f\rangle$. Resulting infinite summations are truncated at some power of T. Detailed description of the theoretical background is given in [4, 17]. Actually the same approach was previously developed for atomic applications, see the papers [59, 60, 61] (and references therein).

Let us apply the finite-order techique to calculate transition dipole moments in Rb. Since property matrix elements are evaluated analytically, there is no need to reduce the symmetry from $D_{\infty h}$ to $C_{\infty v}$. We again begin with the DIRAC step to obtain unformatted files with transformed molecular integrals:

```
pam --inp=moltra --mol=Rb --noarch --get="MRCONEE MDCINT MDPROP"
```

The `MDPROP` file contains three entries representing matrices of the d_x, d_y and d_z components of the dipole moment operator (`XDIPLEN`, `YDIPLEN` and `ZDIPLEN`, respectively). For some reasons the `XDIPLEN` and `ZDIPLEN` matrices are purely imaginary (this was found in the series of preliminary numerical experiments), appealing to complex arithmetics in CC calculation. Such a situation is quite typical.

Thus we recommend to perform an FS-CC calculation of properties in two steps. At the first step we run the usual FS-CC calculation with the real arithmetics employed. The EXP-T input file (`ccsd.inp`) looks like:

```
title "Rb atom -- relativistic EA-CCSD calculation"
maxiter 50
conv 1e-9
sector 0h1p
nactp 8
disk_usage 0
```

At the second step we calculate transition moments (the input file `tdm.inp` is listed below). We should turn on complex arithmetics (`arith complex`) and restart the calculation from the pre-sorted integrals and converged amplitudes (they will be automatically expanded to the complex arithmetics):

```
title "Rb atom -- relativistic EA-CCSD calculation"
maxiter 50
conv 1e-9
sector 0h1p
nactp 8
disk_usage 0

reuse 2-integrals amplitudes
arith complex
print "model vectors"

mdprop "XDIPLEN" sym 1u+ approx 2 2 scheme connected
mdprop "XDIPLEN" sym 1u- approx 2 2 scheme connected
mdprop "YDIPLEN" sym 1u+ approx 2 2 scheme connected
mdprop "YDIPLEN" sym 1u- approx 2 2 scheme connected
mdprop "ZDIPLEN" sym 0u  approx 2 2 scheme connected
```

Note that the instruction `print "model vectors"` was added. It prints additional information about electronic energy levels and is mandatory for further successful processing of the output file with the `expt_spectrum.py` script (for more details on this auxiliary tool, see Section 6.4).

Running EXP-T:

```
expt.x --no-clean ccsd.inp > ccsd.out
expt.x --no-clean tdm.inp  > tdm.out
```

The second run of EXP-T typically produces huge output files since all possible pairs of transition matrix elements are calculated. It is recommended to restrict the manifold of target electronic states to be printed and analyzed using either the `nroots` or `roots_cutoff` keywords. The required TDM values can be found in the output file and then extracted manually or using the `expt_spectrum.py` script:

```
expt_spectrum.py tdm.out -irep 1/2g+ -H
```

where the `-irep` option defines the irrep to which the initial states belong, and `-H` enables hermitization of transition dipoles. The output will be as follows:

```
initial irreps:  ['1/2g+']
initial states:  all
final irreps  :  all
final states  :  all

transitions  1/2g+  ->  1/2u+
               e_i,cm^-1   e_f,cm^-1   . . .   |d|,a.u.    osc.str.
   1 ->    1      0.0000  12555.1431   . . .   1.756964    0.117726
   1 ->    1  12555.1431      0.0000   . . .   1.725401   -0.113534
                                       . . .   1.741111    0.115611
   1 ->    2      0.0000  12786.5663   . . .   2.480954    0.239065
   2 ->    1  12786.5663      0.0000   . . .   2.433780   -0.230060
                                       . . .   2.457254    0.234519
   . . .
transitions  1/2g+  ->  1/2u-          . . .
   . . .
               e_i,cm^-1   e_f,cm^-1   . . .   |d|,a.u.    osc.str.
   1 ->    1      0.0000  12555.1431   . . .   2.484722    0.235452
   1 ->    1  12555.1431      0.0000   . . .   2.440085   -0.227068
                                       . . .   2.462303    0.231222
   1 ->    2      0.0000  12786.5663   . . .   1.754299    0.119532
   2 ->    1  12786.5663      0.0000   . . .   1.720942   -0.115030
                                       . . .   1.737541    0.117260
   . . .
transitions  1/2g+  ->  3/2u+          . . .
   . . .
               e_i,cm^-1   e_f,cm^-1   . . .   |d|,a.u.    osc.str.
   1 ->    1      0.0000  12786.5663   . . .   3.038535    0.358597
   1 ->    1  12786.5663      0.0000   . . .   2.980759   -0.345090
                                       . . .   3.009509    0.351779
```

One can see that the finite-order technique in the quadratic approximation (`approx 2 2`) overestimates transition moments by $\sim 9\%$ (compared to the finite-field approach). It seems reasonable to recalculate transition moments in the linear approximation (`approx 1 0`); for the particular case of the Rb atom it gives nearly the same answers. This is not a surprising result since cluster amplutudes are relatively small (< 0.07). Thus even the linear approximation works correctly.

5.7 "High-spin" CCSD for open shell systems

For input/output files see `examples/O2_highspin`.

The example considered in this subsection is borrowed from the DIRAC program manual [24]. Let us calculate the energy of the ground (triplet) state of the O_2 molecule.

The DIRAC input file contains directives for the Hartree-Fock calculation and subsequent transformation of molecular integrals:

(`O2.mol`)

```
test of open-shell CCSD calculation
molecular oxygen at eq distance taken from NIST
automatic symmetry detection: DIRAC will identify the Dinfh group
C   1              A
       8.    2
O      0.0       0.0      0.60376
O      0.0       0.0     -0.60376
LARGE BASIS cc-pVDZ
FINISH
```

(moltra.inp)

```
!
! calculation of the triplet ground state of molecular oxygen
! (Dirac-Coulomb Hamiltonian)
**DIRAC
.TITLE
molecular oxygen -- ground state
.WAVE F
.4INDEX
**WAVE FUNCTIONS
.SCF
*SCF
.CLOSED SHELL
 6 8
.OPEN SHELL
1
2/4,0
*END OF
```

DIRAC writes occupation numbers of molecular spinors to the MRCONEE unformatted file, but only spinors belonging to doubly occupied shells (the .CLOSED_SHELL keyword) are labeled as occupied. This is why EXP-T will calculate the energy for the O_2^{2+} cation without manually specified occupation numbers.

EXP-T provides two ways to set the occupation numbers of molecular spinors:

- by setting the total number of occupied spinors in each irrep (the occ_irreps keyword). For example, occupation numbers for the open-shell triplet O_2 molecule can be specified using the following two instructions:

  ```
  occ_irreps [1/2g+]:3 [1/2g-]:2 [3/2g+]:0 [3/2g-]:1
  occ_irreps [1/2u+]:2 [1/2u-]:2 [3/2u+]:1 [3/2u-]:1
  ```

  Note that the 1*s*-orbitals of oxygen atoms are frozen here, and the corresponding electrons should not be accounted for in the occ_irreps instructions. For the example discussed:

  - 3 electrons occupy the lowest spinors belonging to the 1/2g+ irrep;
  - 1/2g-, 1/2u+, 1/2u-: 2 electrons in each irrep;
  - 3/2g-, 3/2u+, 3/2u-: 1 electron in each irrep.

- you can also specify the occupation number for *each* spinor explicitly:

  ```
  occ 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 ...
  ```

  To identify spinors which should be populated in the electronic state under consideration, we again recommend to start the calculation with the default occupation numbers, then to interrupt the task and find information about spinors and their symmetry in the output. Then the calculation is restarted with the correct occupation numbers.

  We recommend to use the dirac_mointegral_export program from the DIRAC package to process the MRCONEE unformatted file and get such a string of occupation numbers.

Due to the fact that non-canonical orbitals are used, the Fock matrix is not diagonal and the energy of the vacuum determinant $\langle \Phi_0 | H | \Phi_0 \rangle$ is not equal to the SCF energy obtained before (in the Hartree-Fock calculation with fractional occupation numbers for open-shells):

```
Total SCF energy =                              -149.686661451842
...
sorting one-electron integrals ...
  Fock matrix reconstruction ...
  SCF energy (energy of reference determinant) =    -149.718144633810 a.u.
  SCF energy (energy of reference determinant) was updated:
    old energy =    -149.686661451842 a.u.
    new energy =    -149.718144633810 a.u.
...
         SCF reference energy =    -149.718144633810
      CCSD correlation energy =      -0.366958682652
            Total CCSD energy =    -150.085103316462
```

6 Utility programs

6.1 heffman.x – manipulations with effective Hamiltonian matrices

The heffman.x program (written by A. Zaitsevskii) is used to perform manipulations with FS-CC effective Hamiltonian matrices. Implemented features are:

- the finite-field technique for transition moment calculations (e. g. TDMs) [31, 58, 3];
- extraction of spin-orbit interactions from full-relativistic models by projection [5];
- Pade extrapolation of series of effective Hamiltonians to the limit of zero energy denominator shifts [9].

heffman.x reads effective Hamiltonian matrices of effective Hamiltonians from formatted (text) files supported by both EXP-T and the latest versions of DIRAC.

- to compute finite-field transition moments two files with effective Hamiltonians must be provided, calculated at "field 1" and "field 2";
- to extract spin-orbit interaction matrix elements by the projection technique, two files with effective Hamiltonians must be provided, the first one for spin-orbit (nearly) switched off and the second one for spin-orbit switched on.

  In this case you cannot freeze spinors at the FS-CC step!
- to perform Pade extrapolation of the series of effective Hamiltonians, you should put a sequence of effective Hamiltonians to a single file using the cat command:

  ```
  cat heff_n_eq_1 heff_n_eq_2 heff_n_eq_3 > some_file_name
  ```

  Recall that each effective Hamiltonian matrix is calculated at its own attenuation parameter n. You can also perform extrapolation before transition moments or SO calculation, just use such "composite" files with effective Hamiltonians.

To run heffman.x, an input file containing a list of instructions must be redirected to stdin:

```
heffman.x < input_file
```

heffman.x input files are very similar to EXP-T input files, but keywords are separated from their arguments with the ':' symbol, and the symbol '#' at the beginning of a comment must be placed in the first position of the line.

List of instructions:

file: file with the first effective Hamiltonian or with a sequence of Hamiltonian matrices to be extrapolated, e. g.

- Hamiltonian for the first field value;
- Hamiltonian for the spin-orbit interaction (nearly) switched off;
- Hamiltonian for which you want to get the eigenstates;
- sequence of effective Hamiltonian matrices you want to Padé-extrapolate.

Syntax: `file: <string path-to-first-heff>`

If you are doing finite-field transition moment calculations or SO separation, you have to supply the second file (see the next item).

`file:` heff for another field value, or heff for the spin-orbit switched on, respectively. If there is no second file, the effective Hamiltonian eigenvectors and eigenvalues are written to the formatted file `HEFFEVF`.

Syntax: `file: <string path-to-second-heff>`

`sector:` the Fock space sector you are interested in (heff files for higher sectors contain several heff matrices).

Syntax: `sector: <H>h<P>p`

Example: `sector: 0h2p`

`rep:` irreducible representation you are interested in. As a rule, `rep: 1` stands for the fully symmetrical representation.

Syntax: `rep: <integer n>`

Numbering of representations starts from 1!

`main:`

- for the FF TDM calculations or the Padé extrapolation: the number of the lowest states or the range of states of the given sector and symmetry you want to see in the output file;
- for the SO extraction or the projected approximant construction [9] this keyword defines the subspace of states used for projection.

Syntax: `main: (<integer number> | <integer number1>-<integer number2>).`

Default: all states.

Example: `main: 10` (or, equivalently, `main: 1-10`)

`[L/M]` *(optional)* to perform Padé extrapolation, you have to indicate which matrix Padé approximant [L/M] should be constructed ([0/1], [1/1], [0/2] etc). You should have L+M+2 effective Hamiltonian matrices to build the [L/M] approximant. You can cut off all states above the subspace done by the `main` keyword (see above); to do this, add the `cut` keyword.

Syntax: `[L/M] [cut]`

Default: Padé extrapolation disabled

Example: `[0/1] cut`

`step:` *(only for FF calculations of transition moments.)* Difference of field strenghts (a. u.) between the two finite-field calculations. The presence of this line activates the FF TDM calculation mode (do not use this keyword in other types of calculations).

Syntax: `step: <real value>`

`scale:` *(only for SO calculations.)* Fraction of effective spin-orbit retained in “scalar” calculations to preserve good relativistic symmetries (typically of order 10^{-4}). Omit this line if you do not want to diabatize anything.

Syntax: `scale: <real value>`

`ground:` ground state energy (a. u.). This option is useful if one needs to compute energies with respect to some state (normally ground) which belongs to another symmetry or Fock space sector.

Syntax: `ground: <real energy>`

Default: `ground: 0.0`

`print:` print level. "`print: 0`" or "`print: 1`" are normally ok; enlarging this number, you can get large amount of superfluous information listed.

Syntax: `print: <integer level>`

Default: `print: 1`

`Bloch` activates the computation of finite-field transition moments with Bloch effective Hamiltonian eigenvectors (note: transition property matrices will be non-Hermitian, this has no physical meaning). Otherwise, FF computations are performed with des Cloizeaux effective Hamiltonian eigenvectors ("hermitization").

Syntax: `Bloch`

Default: "hermitization" is enabled in finite-field calculations

When using any of the techniques implemented in the `heffman.x` program, please, cite the corresponding paper.

6.2 `expt_transform_natorb.x` – transformation of model natural spinors to the AO basis

Having model vectors $|\tilde{\psi}_i\rangle$ one can construct approximate density $d_{pq} = \langle\tilde{\psi}_i|a_p^\dagger a_q|\tilde{\psi}_i\rangle$ and transition density matrices $d_{pq} = \langle\tilde{\psi}_i|a_p^\dagger a_q|\tilde{\psi}_j\rangle$ and then calculate model natural (transition) molecular spinors (NS and NTS, respectively) [6, 7]. These operations can be performed by EXP-T itself (see Section 4, keyword `natorb`). EXP-T stores NS and NTS expanded *in the basis of molecular spinors* in formatted files named `NATORB*.dat`. To use visualization tools one has to transform NS or NTS to the *AO basis*. This transformation can be performed by the `expt_transform_natorb.x` utility.

`expt_transform_natorb.x` requires two command-line arguments:

- path to the DIRAC output file containing molecular spinors expanded in the basis of atomic orbitals;
- path to the `NATORB*.dat` formatted file produced by EXP-T and stored in its scratch directory (contains natural spinors expanded in the basis of molecular spinors).

To ask DIRAC to print molecular spinors (vectors) to the output, enable the `.ANALYZE` module in the `**DIRAC` section and then add the following text to the `**ANALYZE` section of the DIRAC `.inp` input file [62]:

```
**ANALYZE
.PRIVEC
*PRIVEC
.AOLAB
.VECPRI
all
```

(`.AOLAB` = "print vectors in the AO basis"). For point groups with inversion one must add the second string specifying the range of molecular spinors:

```
.VECPRI
all
all
```

Running `expt_transform_natorb.x` (for example):

```
expt_transform_natorb.x dirac.out NATORB_0h1p_1:1.dat > nts.out
```

Transformed spinors are flushed to `stdout` again in the DIRAC four-column format and can be redirected to a file.

6.3 `expt2pam.x` – input file format converter

EXP-T does not provide subroutines for solution of Hartree-Fock equations and integral transformation, it completely relies on molecular integrals calculated with the help of any other quantum chemistry software. Thus EXP-T input files do not contain any information on molecular geometry, basis set and pseudopotentials. However, to simplify interaction with the third-party software and make the preparation of input files more convenient, the special file format was developed. It is designed to be quite flexible and free of any format restrictions typical for software written in Fortran.

The EXP-T-style input files containing information on molecular geometry, basis sets and pseudopotentials are recognized by the `expt2pam.x` program, which can produce input files for the following software:

- DIRAC [1, 2]: option `--dirac`;
- CFOUR [63]: option `--cfour`;
- OneProp [8]: option `--oneprop`.

The input file format closely resembles that of NWChem [64, 65, 66, 67], but is substantially simplified. The most important points on the format to be emphasized are:

- only Cartesian coordinates for molecular geometry definition are available to the moment;
- primitive Gaussian functions can be defined without the superfluous contraction coefficient `1.0`;
- primitive Gaussian functions representing the uncontracted part of the basis set with the given angular momentum L can be placed into the same block.

Below the example of such an input file demonstrating the main features of the `expt2pam` format is presented (the UO_2 molecule [56]):

```
#
# Uranium dioxide molecule UO2
#
geometry units angstrom
  U  0.0  0.0   0.000
  O  0.0  0.0   1.775
  O  0.0  0.0  -1.775
end

#
# Contracted basis set adapted for the 28-in-core GRPP by N.S. Mosyagin
#
basis
U S
 11931.71300000
  6539.04200000
  3678.23270000
  2139.54940000
  1288.45330000
   792.07718000
   484.69932000
   298.00127000
   186.73959000
   120.33238000
```

```
    75.85031400
    49.22312100
    30.97937000
    19.33898600
    11.90917900
     7.28108690
     4.52381560
     2.69147870
     1.62933420
     1.02810670
     0.60588439
     0.35564698
     0.20180019
     0.12481356
     0.06383616
U P
  8256.41640000
  4196.77840000
  2191.89420000
  1174.82830000
   644.61478000
   361.41388000
   206.27034000
   119.33670000
    70.14308600
    41.70407000
    24.70288000
    14.49639800
     8.57587000
     4.98086080
     2.84040830
     1.59903720
     0.86476597
     0.45724205
     0.23893352
     0.11879300
     0.05700662
U D
  6012.79360000
  2754.36670000
  1351.96790000
   699.79805000
   377.01141000
   209.73941000
   119.30191000
    69.11171400
    40.43241500
    23.65423900
    13.80639900
     8.06231670
     4.66375400
     2.63565040
     1.45796000
     0.77481216
     0.37248472
     0.17195534
     0.07613361
U F
  1163.36020000
   573.50499000
   285.71163000
   136.47329000
    69.43427500
    36.84465700
    19.84804900
    10.72168500
     5.70450120
     2.92433440
     1.43960400
     0.67385930
     0.29434970
     0.11323985
U G
    28.78919130    0.00068974   -0.00118120   -0.00737416    0.00494647   -0.10867784
    14.01320380   -0.00898795   -0.05421964    0.17284128   -0.36594951   -0.33163329
     6.82095857    0.07975524    0.18134441   -0.50920321    1.20716202    1.79274020
     3.32011698    0.29993353    0.65729862   -0.50617914   -0.67702022   -2.65141994
     1.61607443    0.39302861    0.04007231    1.01049051   -0.71780643    2.66362756
     0.78662787    0.34151054   -0.44799971    0.05077223    1.59628888   -2.01035588
     0.38289289    0.18576214   -0.38383279   -0.74551549   -1.12509753    0.90492997
U H
    14.01320380   -0.02449353   -0.07337288   -0.15906456   -0.23132505
     6.82095857    0.10594999    0.30442614    0.71623431    1.22980099
     3.32011698    0.35765256    0.65468802    0.09614021   -1.24473080
     1.61607443    0.43019757   -0.18764249   -0.95514471    0.32540556
     0.78662787    0.31957043   -0.48353526    0.41946211    0.73545263
     0.38289289    0.10197395   -0.22548628    0.48672026   -0.90679010
U I
     6.82095857    0.12524957   -0.37984959    0.81416792
     3.32011698    0.42961205   -0.61109425   -0.35518331
     1.61607443    0.42313291    0.38916709   -0.73501069
     0.78662787    0.31396902    0.50454838    0.88347742
end


#
# Relativistic pseudopotential (RPP) for Uranium by N.S. Mosyagin
# 28e in core, QED+Breit are included
#
ecp
```

```
U nelec 28
U UL
 1   573.3050904771354       -11.40084159918683
 1   192.8309334143903       -7.097553073552547
 1   59.69944003435170       -28.01857233638748
 1   41.19441327874497        42.95299466064716
 1   27.48466038227223       -39.29896548592660
 1   17.41575927233871        21.07523831226354
 1   10.74156012298913       -8.338244516015038
 1   6.516110723607048        3.069168416643151
 1   4.072460983953957       -1.269035430883378
 1   2.593752889495458       0.4038040550003643
 1   1.642108539873937      -0.7510043731679161E-001
 1  0.5908313234915240       0.3890633153407208E-002
U S
 0   197.0298350128211        6.000000000000000
 1   129.5661840079536       -355.1867566615911
 1   93.02818620674168        1191.419958549814
 1   66.13198912807648       -1657.083906766264
 1   45.36994786350133        1304.226236566319
 1   31.03973237770222       -448.9389445476668
 1   21.15323065428634        86.95891078559328
 1   8.842323178530615       -7.379538028367257
 1   2.503929576228269        3.473180786461057
 2   2.925942781684598       -5.455412335112918
U P
 0   8.311409755029183        5.000000000000000
 1   126.5274319867533       -10.06000092823524
 1   85.18677663478607        70.55297376279486
 1   51.21216731114048       -221.8073735731159
 1   29.49108937084788        312.1302683559188
 1   17.84600190857478       -299.1627976468299
 1   5.463491362893393        229.1321036359295
 2   8.266178701884957       -440.5446582608856
 2   5.564984596440512       -230.2801015020814
 2   3.621496981299310       0.6144684865466843
U D
 0   63.75445017844640        3.000000000000000
 1   35.83829800943973       -244.0643648227415
 1   25.77961295248244        369.6289272300128
 1   16.62718251626653       -66.05286509307552
 2   21.10083097720674       -511.0304327796694
 2   13.00689068383810        209.7715531529684
 2   7.652250767129301       -56.31696437943174
 2   4.429977591227489        19.86068552036704
 2   2.777850799855992       -8.087414881056247
 2   1.804331049757116        2.539779591468687
 2   1.174834421468355      -0.5000680379316691
 2  0.7571835654831253       0.4510869608679968E-001
end
```

```
#
# spin-orbit part of RPP for Uranium
#
so
U P
 1   126.5274319867533     -60.8272897577457000
 1   85.18677663478607      144.881442506022000
 1   51.21216731114048     -174.959781119600000
 1   29.49108937084788      113.392635098503000
 1   17.84600190857478     -60.8426035265438000
 1   5.463491362893393      45.8248263173737000
 2   8.266178701884957     -82.3448048559949000
 2   5.564984596440512     -47.9408659655601000
 2   3.621496981299310       0.4919304400803720
U D
 1   35.83829800943973      -0.4757330698863370
 1   25.77961295248244      -8.8286299074417900
 1   16.62718251626653       9.6090737588470800
 2   21.10083097720674       14.829424758465000
 2   13.00689068383810      -30.085848351789200
 2   7.652250767129301       10.782019233677400
 2   4.429977591227489      -3.9646231214668700
 2   2.777850799855992       1.4116480556389400
 2   1.804331049757116      -0.3371713502943750
 2   1.174834421468355       0.0452391741906467
 2  0.7571835654831253      -0.0020991859739297
U F
 1   573.3050904771354       1.9478068936173300
 1   192.8309334143903       0.3189229591513530
 1   59.69944003435170       1.1230953284521100
 1   41.19441327874497      -2.2059715944841200
 1   27.48466038227223       2.4430555431733500
 1   17.41575927233871      -1.7928190693741500
 1   10.74156012298913       1.0519241531739200
 1   6.516110723607048      -0.5026634876599670
 1   4.072460983953957       0.2165908956875290
 1   2.593752889495458      -0.0639500309090571
 1   1.642108539873937       0.0107075321136888
 1  0.5908313234915240       0.0000807064226547
end


#
# basis for oxygen: aug-cc-pVQZ-DK
#
basis
O S
      61420.00000000    0.00019890   -0.00004520
```

```
       9199.00000000    0.00088840   -0.00020260
       2091.00000000    0.00394250   -0.00089480
        590.90000000    0.01560720   -0.00360580
        192.30000000    0.05294610   -0.01230330
         69.32000000    0.14643820   -0.03646840
         26.97000000    0.30541480   -0.08316710
         11.10000000    0.39799790   -0.15217400
          4.68200000    0.21643540   -0.11462710
          1.42800000    0.01753570    0.29002590
          0.55470000   -0.00249230    0.58584370
          0.20670000    0.00095070    0.27673420
O S
          1.42800000
          0.55470000
          0.20670000
          0.06959000
O P
         63.42000000    0.00611070
         14.66000000    0.04189570
          4.45900000    0.16123290
          1.53100000    0.35667520
          0.53020000    0.44818370
          0.17500000    0.24505420
O P
          1.53100000
          0.53020000
          0.17500000
          0.05348000
O D
          3.77500000
          1.30000000
          0.44400000
          0.15400000
O F
          2.66600000
          0.85900000
          0.32400000
O G
          1.84600000
          0.71400000
end


#
# Empty-core semilocal RPP for oxygen by N.S. Mosyagin
#
ecp
O nelec 0
O UL
   1      2.443097724280e+02      1.369170316080e-01
   1      8.520362677383e+01     -2.852592197455e-01
   2      1.054731223008e+02      2.684494987739e+00
   2      1.825657247757e+01     -9.433884948610e-03
   2      4.806589859822e+00     -1.366233123930e-03
   2      1.753580551235e+00      4.456314085999e-04
   2      6.566864682586e-01     -2.781240176237e-04
   2      4.686645649504e-01      3.558106708362e-04
O S
   1      1.376748432210e+02      5.591554374814e-02
   1      1.378552088359e+01      3.868632083368e-03
   2      6.768869843542e-01      5.419543370664e-04
   2      4.618614187585e-01     -3.882304906071e-04
end

so
O P
   1      2.443097724280e+02     -9.027082481378e-02
   1      8.520362677383e+01      2.802454585393e-01
   2      1.054731223008e+02     -2.610156015857e+00
   2      1.825657247757e+01      1.266811741578e-02
   2      4.806589859822e+00      2.570019532778e-03
   2      1.753580551235e+00      2.973424726019e-04
   2      6.566864682586e-01      6.007087622745e-05
   2      4.686645649504e-01     -5.673502079255e-05
end
```

6.4 expt_spectrum.py – a program for extraction of transition moments from EXP-T output files

EXP-T calculations of transition matrix elements can produce huge output files. To simplify their analysis and extraction of subsets of matrix elements needed for a particular problem, the `expt_spectrum.py` script is provided. It is written in Python 3 and requires the pre-installed NumPy library.

Options of the `expt_spectrum.py` script are as follows:

```
$ expt_spectrum.py --help
usage: expt_spectrum.py [-h] [-u UPPER] [-irep INITIAL_IRREP] [-frep FINAL_IRREP]
       [-i INITIAL_STATE INITIAL_STATE] [-f FINAL_STATE FINAL_STATE]
       [-H] [-s] [-ev] filename

parses exp-t output files and collects information about electronic transitions
and their dipole moments

positional arguments:
  filename

optional arguments:
  -h, --help            show this help message and exit
  -u UPPER              upper energy limit for the electronic states considered
                        (cm^-1 or ev)
  -irep INITIAL_IRREP   irrep of initial electronic states
  -frep FINAL_IRREP     irrep of final electronic states
  -i INITIAL_STATE INITIAL_STATE
                        initial electronic state
  -f FINAL_STATE FINAL_STATE
                        final electronic state
  -H                    force hermiticity of transition moments
  -s                    print detailed information on electronic spectrum
  -ev                   use electron-volts units for energy
```

Names of irreducible representations used here are common for the EXP-T package and are listed in Appendix D.

For example, let us consider allowed $E1$-transitions in the AcF molecule. Its electronic states and $E1$-transitions between them were thoroughly studied in [26]. Suppose that calculated transition moments are stored in the EXP-T output file named `tdms.out`. The following command should be used to show *all* possible allowed transitions from the ground state ($\Omega = 0^+$):

```
expt_spectrum.py tdms.out -i 0 1
```

Note that DIRAC does not discern 0^+ and 0^- states, they both are denoted as `0`. The ground state is the first (`1`) state in the irrep with name `0`.

To calculate radiation lifetime of an excited state and estimate branching ratios during its decay, one should obtain information on transition dipole moments between this excited state and all states lying below it. For the particular case of the AcF molecule, the most strong transition from the $X0^+$ ground state is the transition to the (8)1 state. Thus this excited state seems to be the most easily accessed in experiment. To extract all information about transitions from (8)1 to the manifold of low-lying states, one can use the following command:

```
python expt_spectrum.py tdms.out -i 1+ 8 -u 26200
```

Here the `-u` option is used to restrict the range of final states to those lying below (8)1 (≈ 26170 cm^{-1}).

The `expt_spectrum.py` routine can be combined with Unix `grep` and `awk` utilities to easily extract transition dipole moment functions or even surfaces from a large set of EXP-T output files.

6.5 `expt_diatomic.x` – rotational-vibrational levels and properties of diatomic molecules

If one has a potential energy curve $U(R)$ for an electronic state of a diatomic molecule, one can solve the radial Schrödinger equation and obtain rovibrational levels and wavefunctions as well as any other properties (molecular constants, transition moments, Franck-Condon factors). The radial Schrödinger equation is written as:

$$\left[-\frac{\hbar^2}{2\mu}\frac{d^2}{dr^2} + \frac{J(J+1)}{2\mu r^2} + U(r) \right] \psi_{vJ}(r) = E_{vJ}\ \psi_{vJ}(r), \tag{11}$$

where

- $\mu = \frac{M_1 M_2}{M_1 + M_2}$ – reduced mass,
- v – vibrational quantum number,
- J – rotational quantum number.

Normally this second-order differential equation is solved numerically using some mesh of radial points (see, for example, [68, 69, 70]). Within the current implementation, the $U(R)$ function is interpolated using the cubic natural spline. The matrix version of either the Numerov method [71] or the 2nd order finite-difference method [68] is used to perform integration. Only bound rovibrational states can be treated to the moment.

6.5.1 Input file syntax

The syntax of input files for the solver is quite similar to that of the coupled cluster program. Single line comments start with the '`#`' symbol. The following directives are allowed:

`masses` masses of isotopes M_1 and M_2 (in amu).

Syntax: `masses <float M1> <float M2>`

`charge` net charge of the molecule q. For charged molecules the "charge-modified reduced mass" formula [72, 70] is used:

$$\mu = \frac{M_1 M_2}{M_1 + M_2 - m_e q}, \tag{12}$$

where m_e stands for the electron mass. Can be useful for light molecular ions.

Syntax: `charge <integer q>`

`viblevels` range of vibrational levels of interest [v_{min},v_{max}].

Syntax: `viblevels <integer vmin> <integer vmax>`

Default: `viblevels 0 0`

`rotlevels` range of rotational levels of interest [J_{min},J_{max}].

Syntax: `rotlevels <integer Jmin> <integer Jmax>`

Default: `rotlevels 0 0`

`grid_size` number of radial mesh points.

Syntax: `grid_size <integer npoints>`

Default: `grid_size 300`

`solver` integration algorithm:

`solver numerov`	Numerov method [71, 70] (default)
`solver fd2`	2nd order finite-difference method [68]

`mapping meshkov08 <float ra in Angstroms> <float beta>`

mapping formula for the radial variable (disabled by default). The reduced variable mapping was first proposed by V. V. Meshkov and co-authors [68]:

$$y(r) = \frac{r^\beta - r_a^\beta}{r^\beta + r_a^\beta}, \qquad y \in [-1, 1] \tag{13}$$

Mapping is available only for the finite-difference solver (`solver fd2`).

`write_psi` flush wavefunctions $\psi_{Jv}(R)$ to the formatted text files (disabled by default).

`potential` point-wise potential energy curves and/or property matrix element. The first (r) and the second ($U_1(r)$) columns are mandatory. The third (property) and the fourth ($U_2(r)$) columns are optional. If two potentials are specified, transition matrix elements will be calculated. To calculate Franck-Condon factors, use $M(r) = 1$ in the third column.

Syntax:

```
potential <units of distance> <units of energy>
 <float r> <float U1(r)> [<float M(r)> [<float U2(r)>] ]
 . . .
end
```

Allowed units of distance are: `angstrom`, `atomic`, `bohr`, `au`.

Allowed units of energy are: `cm-1`, `cm`, `atomic`, `hartree`.

6.5.2 Example: Franck-Condon factors for the AcOH$^+$ molecular ion

To estimate suitability of a molecule for direct laser cooling one should calculate vibrational branching ratios and Franck-Condon factors (FCFs) for the working electronic transition [73, 74]. Recently the AcOH$^+$ molecular ion was proposed as a candidate for direct laser cooling [75].

This molecule is linear in both the ground (1)1/2 and the first excited (1)3/2 electronic states. If we neglect the anharmonic couplings between stretching and bending modes, the total FCF between the (v_1', v_2', v_3') and the (v_1'', v_2'', v_3'') vibrational states can be factorized:

$$\mathrm{FCF} = |\, \langle v_1' v_3' | v_1'' v_3'' \rangle \,|^2 \times |\, \langle v_2' | v_2'' \rangle \,|^2 \tag{14}$$

The latter factor is the one-dimensional bending mode FCF and it is expected to be very close to unity since the corresponding sections of potential energy surfaces are perfectly parallel to each other. The overlap intergal over the v_3 stretch mode is expected to be nearly equal to unity and, which is more important, this factor will be the same for the $(v_1', v_2', 0)$ and $(v_1'', v_2'', 0)$ vibrational states analyzed here since this stretching mode is not excited in the states under consideration. Thus the total FCF can be approximated by the squared one-dimensional $\langle v_1' | v_1'' \rangle$ overlap integral. To calculate this integral one should solve the radial Schrödinger equation, considering the OH group as a pseudoatom.

The input file for the `expt_diatomic.x` program:

```
#
# calculation of Frank-Condon factors for the (1)1/2-(1)3/2 transition in AcOH+
# potential curves were calculated by the relativistic FS-CCSD(0h1p) method,
#
# for more details, see:
# A. Oleynichenko, L. V. Skripnikov, A. V. Zaitsevskii, V. V. Flambaum
# Laser-coolable AcOH+ ion for CP-violation searches (2022)
# arXiv:2112.02307 [physics.atom-ph]
#
#          Ac         OH
masses 227.028000 17.003000
rotlevels 0 0
viblevels 0 5

# r    potential1(r)   property(r)=1.0    potential2(r)
# units:  dist   energy
potential atomic atomic
0.340150702E+01  -0.451045595E+03   1.0  -0.451004746E+03
0.349599333E+01  -0.451064998E+03   1.0  -0.451025470E+03
0.359047964E+01  -0.451079018E+03   1.0  -0.451040834E+03
0.368496594E+01  -0.451088543E+03   1.0  -0.451051696E+03
0.377945225E+01  -0.451094322E+03   1.0  -0.451058791E+03
0.387393856E+01  -0.451096994E+03   1.0  -0.451062746E+03
```

```
0.396842486E+01  -0.451097103E+03   1.0  -0.451064095E+03
0.406291117E+01  -0.451095111E+03   1.0  -0.451063297E+03
0.415739747E+01  -0.451091413E+03   1.0  -0.451060740E+03
0.425188378E+01  -0.451086341E+03   1.0  -0.451056758E+03
0.434637009E+01  -0.451080177E+03   1.0  -0.451051631E+03
0.444085639E+01  -0.451073161E+03   1.0  -0.451045598E+03
0.453534270E+01  -0.451065491E+03   1.0  -0.451038858E+03
end
```

Run the job:

```
expt_diatomic.x AcOH+_FCF.inp | tee AcOH+_FCF.out
```

At the end of the output file one can find the table with transition moments (and their squares). We select only the FCFs with the wavenumber of the upper state $v'' = 0$:

```
    J' J''  v' v''   . . .  delta e,cm^-1            prop        |prop|^2
$    0   0   0   0   . . .      7279.8734       -0.942947        0.889149
$    0   0   1   0   . . .      6627.6841        0.325405        0.105888
$    0   0   2   0   . . .      5979.2731        0.069678        0.004855
$    0   0   3   0   . . .      5334.3122        0.010337        0.000107
$    0   0   4   0   . . .      4692.7001        0.001070        0.000001
$    0   0   5   0   . . .      4054.6587       -0.000079        0.000000
```

The sum of first three FCFs (see the last column) is equal to 0.9999, thus allowing one to conclude that the closed optical cycle can exist for the AcOH$^+$ molecular cation [75].

Appendix A Alphabetical index of keywords

Appendix B Formatted files for storing property operator matrices

EXP-T can be used to calculate matrix elements (expectation values and transition moments) of an arbitrary one-electron property operator not implemented in DIRAC. To carry out such a calculation EXP-T requires two text formatted files containing real and imaginary parts of a property operators matrix expressed in the basis of molecular spinors. Each of these two files consists of a list of lines storing indices (i, j) of a matrix element and an element A_{ij} itself (its real or imaginary part):

```
...
<integer i> <integer j> <real A_ij>
...
```

Numbering of indices starts from 1. Note that since one-electron property operators are Hermitian, the real part of a matrix must be symmetric while the imaginary part is always antisymmetric.

Appendix C Temporary files

During the calculation, EXP-T operates with temporary files stored in the *scratch* directory. We recommend to remove these temporary files after every calculation (except for those planned to be used in the future, like the `HEFF` files or files with cluster amplitudes). All types of temporary files generated by EXP-T are listed and described below.

`HINT` one-electron integrals: core Hamiltonian matrix + one-particle part of perturbation (if presented). Recommended for removal after calculation.

`VINT-*-*-*-*` two-electron integrals. These files are generated during the integral sorting stage. Recommended for removal after calculation.

Approximate number of files: $\sim N_b^4$ *(N_b – number of blocks into which the whole set of spinors is divided by symmetry and the* `tilesize` *parameter)*

`*.sb` a file containing single symmetry block of molecular integrals/cluster amplitudes. It is just a part of a diagram. During the calculation each file is mapped to some structure in RAM. Recommended for removal after calculation.

Approximate number of `.sb` *files:* $\sim 10 \times N_b^4$ *for CCSD,* $\sim N_b^6$ *for CCSDT-n and CCSDT models* $\Rightarrow$ ***can be very large!***

`*.dg` a file containing a diagram. Diagram files containing cluster amplitudes (`t1c.dg`, `t2c.dg` etc) can be used in subsequent calculations when constructing the initial approximation to amplitudes (see Section 4, keyword `reuse`).

`HEFF` a formatted (text) file containing effective Hamiltonian matrices. These files can be read by the `heffman.x` utility program used to perform Padé extrapolation of series of effective Hamiltonians [9], finite-field transition moment calculations [58, 3], spin-orbit coupling calculations [5].

`*DIPLEN` formatted files with dipole moment integrals (just its electronic part) in the basis of molecular spinors.

`MVCOEF**` binary files containing model vectors expanded in the basis of model determinants. These files are used to organize data flow inside EXP-T in the most convenient and logical manner; model vectors are used for calculation of model-space approximations of TDMs, model natural spinors, properties within the finite-order approach, etc.

`NATORB*.dat` formatted text files containing model natural spinors expanded in the basis of molecular spinors. These expansions can be transformed to the AO basis with the `expt_transform_natorb.x` utility program (see Section 6.2).

`density_0h0p.dat`

a formatted text file containing one-electron CC density matrix (in the basis of molecular spinors).

Appendix D Symmetry and irreducible representations

This section lists the names of irreducible representations (irreps) used in EXP-T.

- For non-relativistic groups, an irrep is determined by the spatial symmetry (Mulliken notation is used) and the M_s value (for $M_s >= 2$ only symbol 2 is used);
- For relativistic double groups we use the Mulliken-type notation (completely inherited from DIRAC);
- For $U(1) = \overline{C_{\infty v}}$ and $U(1) \times C_i = \overline{D_{\infty h}}$ irreps are determined by Ω values.

C_1

```
A_a
A_b
A_-3/2
A_+3/2
A_0
A_4
A_+1
A_-1
```

C_2

```
A_a    B_a
A_b    B_b
A_-3/2 B_-3/2
A_+3/2 B_+3/2
A_0    B_0
A_2    B_2
A_+1   B_+1
A_-1   B_-1
```

C_s

```
A’_a    A"_a
A’_b    A"_b
A’_-3/2 A"_-3/2
A’_+3/2 A"_+3/2
A’_0    A"_0"
A’_2    A"_2"
A’_+1   A"_+1
A’_-1   A"_-1
```

C_i

```
Ag_a    Au_a
Ag_b    Au_b
Ag_-3/2 Au_-3/2
Ag_+3/2 Au_+3/2
Ag_0    Au_0
Ag_2    Au_2
Ag_+1   Au_+1
Ag_-1   Au_-1
```

C_{2v}

```
A1_a    B2_a    B1_a    A2_a
A1_b    B2_b    B1_b    A2_b
A1_-3/2 B2_-3/2 B1_-3/2 A2_-3/2
A1_+3/2 B2_+3/2 B1_+3/2 A2_+3/2
A1_0    B2_0    B1_0    A2_0
A1_2    B2_2    B1_2    A2_2
A1_+1   B2_+1   B1_+1   A2_+1
A1_-1   B2_-1   B1_-1   A2_-1
```

C_{2h}

```
Ag_a    Bg_a    Bu_a    Au_a
Ag_b    Bg_b    Bu_b    Au_b
Ag_-3/2 Bg_-3/2 Bu_-3/2 Au_-3/2
Ag_+3/2 Bg_+3/2 Bu_+3/2 Au_+3/2
Ag_0    Bg_0    Bu_0    Au_0
Ag_2    Bg_2    Bu_2    Au_2
Ag_+1   Bg_+1   Bu_+1   Au_+1
Ag_-1   Bg_-1   Bu_-1   Au_-1
```

D_2

```
A_a    B3_a    B1_a    B2_a
A_b    B3_b    B1_b    B2_b
A_-3/2 B3_-3/2 B1_-3/2 B2_-3/2
A_+3/2 B3_+3/2 B1_+3/2 B2_+3/2
A_0    B3_0    B1_0    B2_0
A_2    B3_2    B1_2    B2_2
A_+1   B3_+1   B1_+1   B2_+1
A_-1   B3_-1   B1_-1   B2_-1
```

D_{2h}

```
Ag_a    B1u_a    B2u_a    B3g_a    B3u_a    B2g_a    B1g_a    Au_a
Ag_b    B1u_b    B2u_b    B3g_b    B3u_b    B2g_b    B1g_b    Au_b
Ag_-3/2 B1u_-3/2 B2u_-3/2 B3g_-3/2 B3u_-3/2 B2g_-3/2 B1g_-3/2 Au_-3/2
Ag_+3/2 B1u_+3/2 B2u_+3/2 B3g_+3/2 B3u_+3/2 B2g_+3/2 B1g_+3/2 Au_+3/2
Ag_0    B1u_0    B2u_0    B3g_0    B3u_0    B2g_0    B1g_0    Au_0
Ag_2    B1u_2    B2u_2    B3g_2    B3u_2    B2g_2    B1g_2    Au_2
Ag_+1   B1u_+1   B2u_+1   B3g_+1   B3u_+1   B2g_+1   B1g_+1   Au_+1
Ag_-1   B1u_-1   B2u_-1   B3g_-1   B3u_-1   B2g_-1   B1g_-1   Au_-1
```

$\overline{C_1}$

```
A
a
```

$\overline{C_2}$

```
1E 2E
a  b
```

$\overline{C_s}$

```
1E 2E
a  b
```

$\overline{C_i}$

```
AG AU
ag au
```

$\overline{C_{2v}}$

```
1E 2E
a  b
```

$\overline{D_2}$

```
1E 2E
a  b
```

$\overline{D_{2h}}$

```
1Eg 2Eg 1Eu 2Eu
ag  bg  au  bu
```

$\overline{C_{\infty v}}$

```
 1/2+  1/2-  3/2+  3/2-  5/2+  5/2-  7/2+  7/2-
 9/2+  9/2- 11/2+ 11/2- 13/2+ 13/2- 15/2+ 15/2-
17/2+ 17/2- 19/2+ 19/2- 21/2+ 21/2- 23/2+ 23/2-
25/2+ 25/2- 27/2+ 27/2- 29/2+ 29/2- 31/2+ 31/2-
   0     1+    1-    2+    2-    3+    3-    4+
   4-    5+    5-    6+    6-    7+    7-    8+
   8-    9+    9-   10+   10-   11+   11-   12+
  12-   13+   13-   14+   14-   15+   15-   16+
```

$\overline{D_{\infty h}}$

```
1/2g+  1/2g-  3/2g+  3/2g-  5/2g+  5/2g-  7/2g+  7/2g-
9/2g+  9/2g- 11/2g+ 11/2g- 13/2g+ 13/2g- 15/2g+ 15/2g-
1/2u+  1/2u-  3/2u+  3/2u-  5/2u+  5/2u-  7/2u+  7/2u-
9/2u+  9/2u- 11/2u+ 11/2u- 13/2u+ 13/2u- 15/2u+ 15/2u-
  0g     1g+    1g-    2g+    2g-    3g+    3g-    4g+
  4g-    5g+    5g-    6g+    6g-    7g+    7g-    8g+
  0u     1u+    1u-    2u+    2u-    3u+    3u-    4u+
  4u-    5u+    5u-    6u+    6u-    7u+    7u-    8u+
```

References

[1] DIRAC, a relativistic ab initio electronic structure program, Release DIRAC18 (2018), written by T. Saue, L. Visscher, H. J. Aa. Jensen, and R. Bast, with contributions from V. Bakken, K. G. Dyall, S. Dubillard, U. Ekstroem, E. Eliav, T. Enevoldsen, E. Fasshauer, T. Fleig, O. Fossgaard, A. S. P. Gomes, E. D. Hedegaard, T. Helgaker, J. Henriksson, M. Ilias, Ch. R. Jacob, S. Knecht, S. Komorovsky, O. Kullie, J. K. Laerdahl, C. V. Larsen, Y. S. Lee, H. S. Nataraj, M. K. Nayak, P. Norman, G. Olejniczak, J. Olsen, J. M. H. Olsen, Y. C. Park, J. K. Pedersen, M. Pernpointner, R. Di Remigio, K. Ruud, P. Salek, B. Schimmelpfennig, A. Shee, J. Sikkema, A. J. Thorvaldsen, J. Thyssen, J. van Stralen, S. Villaume, O. Visser, T. Winther, and S. Yamamoto (see http://diracprogram.org).

[2] T. Saue, R. Bast, A. S. P. Gomes, H. J. Aa. Jensen, L. Visscher, I. A. Aucar, R. Di Remigio, K. G. Dyall, E. Eliav, E. Fasshauer, T. Fleig, L. Halbert, E. D. Hedegård, B. Helmich-Paris, M. Iliaš, C. R. Jacob, S. Knecht, J. K. Laerdahl, M. L. Vidal, M. K. Nayak, M. Olejniczak, J. M. H. Olsen, M. Pernpointner, B. Senjean, A. Shee, A. Sunaga, and J. N. P. van Stralen. The DIRAC code for relativistic molecular calculations. *J. Chem. Phys.*, 152(20):204104, 2020.

[3] A. V. Zaitsevskii, L. V. Skripnikov, A. V. Kudrin, A. V. Oleinichenko, E. Eliav, and A. V. Stolyarov. Electronic transition dipole moments in relativistic coupled-cluster theory: the finite-field method. *Opt. Spectrosc. (English translation of Optika i Spektroskopiya)*, 124(4):451, 2018.

[4] A. Zaitsevskii, A. V. Oleynichenko, and E. Eliav. Theoretical molecular spectroscopy of actinide compounds: the ThO molecule. *Mol. Phys.*, REHE 2022 Special Issue:e2236246, 2023.

[5] A. Zaitsevskii, N. S. Mosyagin, A. V. Stolyarov, and E. Eliav. Approximate relativistic coupled-cluster calculations on heavy alkali-metal diatomics: Application to the spin-orbit-coupled $A^1\Sigma^+$ and $b^3\Pi$ states of RbCs and Cs_2. *Phys. Rev. A*, 96:022516, 2017.

[6] A. I. Krylov. From orbitals to observables and back. *J. Chem. Phys.*, 153(8):080901, 2020.

[7] T. Isaev, A. V. Oleynichenko, D. A. Makinskii, and A. Zaitsevskii. Optical cycling in charged complexes with Ra-N bonds. *Chem. Phys. Lett.*, 845:141301, 2024.

[8] http://qchem.pnpi.spb.ru/Leonid.

[9] A. Zaitsevskii and E. Eliav. Padé extrapolated effective Hamiltonians in the Fock space relativistic coupled cluster method. *Int. J. Quantum Chem.*, 118(23):e25772, 2018.

[10] A. Zaitsevskii, N. S. Mosyagin, A. V. Oleynichenko, and E. Eliav. Generalized relativistic small-core pseudopotentials accounting for quantum electrodynamic effects: Construction and pilot applications. *Int. J. Quantum Chem.*, 123(8):e27077, 2022.

[11] https://developer.nvidia.com/cuda-zone.

[12] A. V. Oleynichenko, A. Zaitsevskii, L. V. Skripnikov, and E. Eliav. Relativistic Fock space coupled cluster method for many-electron systems: non-perturbative account for connected triple excitations. *Symmetry*, 12(7):1101, 2020.

[13] L. V. Skripnikov, A. V. Oleynichenko, A. V. Zaitsevskii, D. E. Maison, and A. E. Barzakh. Relativistic fock space coupled-cluster study of bismuth electronic structure to extract the Bi nuclear quadrupole moment. *Phys. Rev. C*, 104:034316, 2021.

[14] E. Eliav, A. Borschevsky, A. Zaitsevskii, A. V. Oleynichenko, and U. Kaldor. Relativistic Fock-space coupled cluster method: theory and recent applications. In M. Yanez and R. J. Boyd, editors, *Comprehensive Computational Chemistry (First Edition)*, pages 79–93. Elsevier, Oxford, 1st edition, 2024.

[15] A. Zaitsevskii, A. V. Oleynichenko, and E. Eliav. Finite-field calculations of transition properties by the Fock space relativistic coupled cluster method: transitions between different Fock space sectors. *Symmetry*, 12(11):1845, 2020.

[16] A. V. Oleynichenko, L. V. Skripnikov, A. Zaitsevskii, E. Eliav, and V. M. Shabaev. Diagonal and off-diagonal hyperfine structure matrix elements in KCs within the relativistic Fock space coupled cluster theory. *Chem. Phys. Lett.*, 756:137825, 2020.

[17] A. V. Oleynichenko, A. Zaitsevskii, S. V. Kondratyev, and E. Eliav. Direct calculation of transition matrix elements in relativistic coupled cluster theory. *Opt. Spectrosc. (English translation of Optika i Spektroskopiya)*, 131:1471, 2023.

[18] U. Kaldor. The Fock Space Coupled Cluster Method: Theory and Application. *Theor. Chim. Acta*, 80:427 – 439, 1991.

[19] E. Eliav, U. Kaldor, and Y. Ishikawa. Open-shell relativistic coupled-cluster method with Dirac-Fock-Breit wave functions: energies of the gold atom and its cation. *Phys. Rev. A*, 49(3):1724–1729, 1994.

[20] OpenBLAS, An optimized BLAS library. https://www.openblas.net/.

[21] Intel oneAPI Math Kernel Library. https://www.intel.com/content/www/us/en/developer/tools/oneapi/onemkl.html.

[22] http://diracprogram.org/doc/master/manual/moltra.html#scheme.

[23] T. Saue and H. J. Aa Jensen. Quaternion symmetry in relativistic molecular calculations: The Dirac–Hartree–Fock method. *J. Chem. Phys.*, 111(14):6211–6222, 1999.

[24] http://www.diracprogram.org/doc/release-18/tutorials/highspin_cc/O2.html.

[25] https://www.mrcc.hu/index.php/documentation/interfaces.

[26] L. V. Skripnikov, A. V. Oleynichenko, A. Zaitsevskii, N. S. Mosyagin, M. Athanasakis-Kaklamanakis, M. Au, and G. Neyens. *Ab initio* study of electronic states and radiative properties of the AcF molecule. *J. Chem. Phys.*, 159(12):124301, 2023.

[27] J. D. Watts, J. Gauss, and R. J. Bartlett. Coupled-cluster methods with noniterative triple excitations for restricted open-shell Hartree–Fock and other general single determinant reference functions. Energies and analytical gradients. *J. Chem. Phys.*, 98(11):8718–8733, 1993.

[28] A. Shee, L. Visscher, and T. Saue. Analytic one-electron properties at the 4-component relativistic coupled cluster level with inclusion of spin-orbit coupling. *J. Chem. Phys.*, 145(18):184107, 2016.

[29] V. Hurtubise and K. F. Freed. The Algebra of Effective Hamiltonians and Operators: Exact Operators. In *Advances in Chemical Physics*, volume 83, pages 465–541. John Wiley & Sons, Inc., 1993.

[30] A. Hehn and L. Visscher. *(to be published)*.

[31] A. Zaitsevskii and A. P. Pychtchev. On the finite-field transition dipole moment calculations by effective Hamiltonian methods. *Eur. Phys. J. D*, 4(3):303, 1998.

[32] S. O. Adamson, A. Zaitsevskii, and N. F. Stepanov. *Ab initio* finite-field transition moment calculations. *J. Phys. B: At. Mol. Opt. Phys.*, 31(24):5275–5280, 1998.

[33] T. A. Isaev, A. V. Zaitsevskii, A. Oleynichenko, E. Eliav, A. A. Breier, T. F. Giesen, R. F. Garcia Ruiz, and R. Berger. Ab initio study and assignment of electronic states in molecular RaCl. *J. Quant. Spectrosc. Radiat. Transf.*, 269:107649, 2021.

[34] V. Krumins, A. Kruzins, M. Tamanis, R. Ferber, A. Pashov, A. V. Oleynichenko, A. Zaitsevskii, E. A. Pazyuk, and A. V. Stolyarov. The branching ratio of intercombination $A^1\Sigma^+ \sim b^3\Pi \to a^3\Sigma^+/X^1\Sigma^+$ transitions in the RbCs molecule: measurements and calculations. *J. Quant. Spectrosc. Radiat. Transf.*, 256:107291, 2020.

[35] M. Ziółkowski, V. Weijo, P. Jørgensen, and J. Olsen. An efficient algorithm for solving nonlinear equations with a minimal number of trial vectors: Applications to atomic-orbital based coupled-cluster theory. *J. Chem. Phys.*, 128(20):204105, 2008.

[36] P. Ettenhuber and P. Jørgensen. Discarding information from previous iterations in an optimal way to solve the coupled cluster amplitude equations. *J. Chem. Theory Comput.*, 11(4):1518–1524, 2015.

[37] P. Pulay. Convergence acceleration of iterative sequences. The case of SCF iteration. *Chem. Phys. Lett.*, 73(2):393 – 398, 1980.

[38] G. E. Scuseria, T. J. Lee, and H. F. Schaefer III. Accelerating the convergence of the coupled-cluster approach: the use of the DIIS method. *Chem. Phys. Lett.*, 130(3):236 – 239, 1986.

[39] E. Eliav, M. J. Vilkas, Y. Ishikawa, and U. Kaldor. Extrapolated intermediate Hamiltonian coupled-cluster approach: theory and pilot application to electron affinities of alkali atoms. *J. Chem. Phys.*, 122:224113, 2005.

[40] LZ4 – Fast LZ compression algorithm. http://www.lz4.org.

[41] T. H. Dunning. Gaussian basis sets for use in correlated molecular calculations. I. The atoms boron through neon and hydrogen. *J. Chem. Phys.*, 90(2):1007, 1989.

[42] A. N. Petrov, N. S. Mosyagin, A. V. Titov, and I. I. Tupitsyn. Accounting for the Breit interaction in relativistic effective core potential calculations of actinides. *J. Phys. B At. Mol. Opt. Phys.*, 37(23):4621–4637, 2004.

[43] T. Hangele, M. Dolg, M. Hanrath, X. Cao, and P. Schwerdtfeger. Accurate relativistic energy-consistent pseudopotentials for the superheavy elements 111 to 118 including quantum electrodynamic effects. *J. Chem. Phys.*, 136(21):214105, 2012.

[44] T. Hangele, M. Dolg, and P. Schwerdtfeger. Relativistic energy-consistent pseudopotentials for superheavy elements 119 and 120 including quantum electrodynamic effects. *J. Chem. Phys.*, 138(17):174113, 2013.

[45] N. S. Mosyagin, A. V. Zaitsevskii, and A. V. Titov. Generalized relativistic effective core potentials for superheavy elements. *Int. J. Quantum Chem.*, 120(2):e26076, 2020.

[46] J. E. Sansonetti and W. C. Martin. Handbook of Basic Atomic Spectroscopic Data. *J. Phys. Chem. Ref. Data*, 34(4):1559, 2005.

[47] N. S. Mosyagin, A. V. Titov, E. Eliav, and U. Kaldor. Generalized relativistic effective core potential and relativistic coupled cluster calculation of the spectroscopic constants for the HgH molecule and its cation. *J. Chem. Phys.*, 115(5):2007–2013, 2001.

[48] N. S. Mosyagin, A. V. Titov, R. J. Buenker, H.-P. Liebermann, and A. B. Alekseyev. GRECP/MRD-CI calculations on the Hg atom and HgH molecule. *Int. J. Quantum Chem.*, 88(5):681–686, 2002.

[49] M. G. Kozlov and A. Derevianko. Proposal for a sensitive search for the electric dipole moment of the electron with matrix-isolated radicals. *Phys. Rev. Lett.*, 97(6):063001, 2006.

[50] S. Sasmal, H. Pathak, M. K. Nayak, N. Vaval, and S. Pal. Search for parity and time reversal violating effects in HgH: Relativistic coupled-cluster study. *J. Chem. Phys.*, 144(12):124307, 2016.

[51] A. J. Geddes, L. V. Skripnikov, A. Borschevsky, J. C. Berengut, V. V. Flambaum, and T. P. Rakitzis. Enhanced nuclear-spin-dependent parity-violation effects using the ^{199}HgH molecule. *Phys. Rev. A*, 98:022508, 2018.

[52] S. Yin, X. Yuan, Y. Liu, H. Xu, and B. Yan. Spin-orbit coupling in low-lying electronic states of mercury hydride. *J. Quant. Spectrosc. Radiat. Transf.*, 229:120–129, 2019.

[53] N. S. Mosyagin, A. V. Titov, and Z. Latajka. Generalized relativistic effective core potential: Gaussian expansions of potentials and pseudospinors for atoms Hg through Rn. *Int. J. Quantum Chem.*, 63(6):1107–1122, 1997.

[54] D. Sundholm. VIBROT, http://www.chem.helsinki.fi/~sundholm/software/GPL/.

[55] J. Almlöf and Peter R. Taylor. *Atomic natural orbital (ANO) basis sets for quantum chemical calculations*, page 301–373. Elsevier, 1991.

[56] A. V. Oleynichenko, A. Zaitsevskii, N. S. Mosyagin, A. N. Petrov, E. Eliav, and A. V. Titov. LIBGRPP: A library for the evaluation of molecular integrals of the generalized relativistic pseudopotential operator over Gaussian functions. *Symmetry*, 15(1):197, 2023.

[57] https://www.nist.gov/pml/atomic-spectra-database.

[58] A. A. Medvedev, A. V. Stolyarov, A. Zaitsevskii, and A. Eliav. Relativistic calculations on the electric dipole transition probabilities of the RbAr exciplex. *Nonlinear Phenomena in Complex Systems*, 20(2):205, 2017.

[59] S. A. Blundell, W. R. Johnson, and J. Sapirstein. Relativistic all-order calculations of energies and matrix elements in cesium. *Phys. Rev. A*, 43(7):3407–3418, 1991.

[60] M. S. Safronova, W. R. Johnson, and A. Derevianko. Relativistic many-body calculations of energy levels, hyperfine constants, electric-dipole matrix elements, and static polarizabilities for alkali-metal atoms. *Phys. Rev. A*, 60:4476–4487, 1999.

[61] B. K. Sahoo, S. Majumder, H. Merlitz, R. Chaudhuri, B. P. Das, and D. Mukherjee. Electric dipole transition amplitudes for ^{207}Pb^{+}. *J. Phys. B: At. Mol. Opt. Phys.*, 39(2):355–363, 2005.

[62] https://www.diracprogram.org/doc/release-18/manual/analyze.html.

[63] D. A. Matthews, L. Cheng, M. E. Harding, F. Lipparini, S. Stopkowicz, T.-C. Jagau, P. G. Szalay, J. Gauss, and J. F. Stanton. Coupled-cluster techniques for computational chemistry: The CFOUR program package. *J. Chem. Phys.*, 152(21):214108, 2020.

[64] E. Apra, E. J. Bylaska, W. A. de Jong, N. Govind, K. Kowalski, T. P. Straatsma, M. Valiev, H. J. J. van Dam, Y. Alexeev, J. Anchell, V. Anisimov, F. W. Aquino, R. Atta-Fynn, J. Autschbach, N. P. Bauman, J. C. Becca, D. E. Bernholdt, K. Bhaskaran-Nair, S. Bogatko, P. Borowski, J. Boschen, J. Brabec, A. Bruner, E. Cauët, Y. Chen, G. N. Chuev, C. J. Cramer, J. Daily, M. J. O. Deegan, T. H. Dunning Jr., M. Dupuis, K. G. Dyall, G. I. Fann, S. A. Fischer, A. Fonari, H. Früchtl, L. Gagliardi, J. Garza, N. Gawande, S. Ghosh, K. Glaesemann, A. W. Götz, J. Hammond, V. Helms, E. D. Hermes, K. Hirao, S. Hirata, M. Jacquelin, L. Jensen, B. G. Johnson, H. Jónsson, R. A. Kendall,

M. Klemm, R. Kobayashi, V. Konkov, S. Krishnamoorthy, M. Krishnan, Z. Lin, R. D. Lins, R. J. Littlefield, A. J. Logsdail, K. Lopata, W. Ma, A. V. Marenich, J. Martin del Campo, D. Mejia-Rodriguez, J. E. Moore, J. M. Mullin, T. Nakajima, D. R. Nascimento, J. A. Nichols, P. J. Nichols, J. Nieplocha, A. Otero-de-la-Roza, B. Palmer, A. Panyala, T. Pirojsirikul, B. Peng, R. Peverati, J. Pittner, L. Pollack, R. M. Richard, P. Sadayappan, G. C. Schatz, W. A. Shelton, D. W. Silverstein, D. M. A. Smith, T. A. Soares, D. Song, M. Swart, H. L. Taylor, G. S. Thomas, V. Tipparaju, D. G. Truhlar, K. Tsemekhman, T. Van Voorhis, A. Vázquez-Mayagoitia, P. Verma, O. Villa, A. Vishnu, K. D. Vogiatzis, D. Wang, J. H. Weare, M. J. Williamson, T. L. Windus, K. Woliński, A. T. Wong, Q. Wu, C. Yang, Q. Yu, M. Zacharias, Z. Zhang, Y. Zhao, and R. J. Harrison. NWChem: Past, present, and future. *J. Chem. Phys.*, 152(18):184102, 2020.

[65] NWChem documentation. Basis sets. https://nwchemgit.github.io/Basis.html.

[66] NWChem documentation. Effective core potentials. https://nwchemgit.github.io/ECP.html.

[67] NWChem documentation. Geometries. https://nwchemgit.github.io/Geometry.html.

[68] V. V. Meshkov, A. V. Stolyarov, and R. J. Le Roy. Adaptive analytical mapping procedure for efficiently solving the radial Schrödinger equation. *Phys. Rev. A*, 78(5):052510, 2008.

[69] V. V. Meshkov, A. V. Stolyarov, and R. J. Le Roy. Rapid, accurate calculation of the s-wave scattering length. *J. Chem. Phys.*, 135(15):154108, 2011.

[70] R. J. Le Roy. LEVEL: A computer program for solving the radial Schrödinger equation for bound and quasibound levels. *J. Quant. Spectrosc. Radiat. Transf.*, 186:167–178, 2017.

[71] M. Pillai, J. Goglio, and T. G. Walker. Matrix Numerov method for solving Schrödinger's equation. *Am. J. Phys.*, 80(11):1017–1019, 2012.

[72] J. K. G. Watson. The isotope dependence of diatomic Dunham coefficients. *J. Mol. Spectrosc.*, 80(2):411–421, 1980.

[73] T. A. Isaev and R. Berger. Polyatomic candidates for cooling of molecules with lasers from simple theoretical concepts. *Phys. Rev. Lett.*, 116:063006, 2016.

[74] M. R. Tarbutt. Laser cooling of molecules. *Contemp. Phys.*, 59(4):356–376, 2018.

[75] A. V. Oleynichenko, L. V. Skripnikov, A. V. Zaitsevskii, and V. V. Flambaum. Laser-coolable AcOH$^+$ ion for $\mathcal{CP}$-violation searches. *Phys. Rev. A*, 105:022825, 2022.