

The RATIP program for relativistic calculations of atomic transition, ionization and recombination properties[☆]

S. Fritzsche^{a,b,c,*}

^a GSI Helmholtzzentrum für Schwerionenforschung, D-64291 Darmstadt, Germany

^b FIAS Frankfurt Institute for Advanced Studies, D-60438 Frankfurt am Main, Germany

^c Department of Physics, P.O. Box 3000, Fin-90014 University of Oulu, Finland

ARTICLE INFO

Article history:

Received 14 December 2011

Received in revised form 11 February 2012

Accepted 14 February 2012

Available online 21 February 2012

Keywords:

Atomic

Auger amplitudes and properties

Dirac–Coulomb–Breit Hamiltonian

Einstein coefficients

Multiconfiguration Dirac–Fock

Photoionization amplitudes and cross sections

Radiative and dielectronic recombination

Transition probability

ABSTRACT

During the past decade, the RATIP program has been developed to calculate the electronic structure and properties of atoms and ions. This code, which is now organized as a suite of programs, provides a powerful platform today to generate and evaluate atomic data for open-shell atoms, including level energies and energy shifts, transition probabilities, Auger parameters as well as a variety of excitation, ionization and recombination amplitudes and cross sections. Although the RATIP program focus on properties with just *one* electron within the continuum, recent emphasis was placed also on second-order processes as well as on the combination of different types of transition amplitudes in order to explore more complex spectra. Here, I present and discuss the (design of the) RATIP program and make available a major part of the code for public use. Selected examples show a few of its possible applications, while reference is made to a much wider range of computations as supported by the program. The RATIP program has been developed as a scalar FORTRAN 90/95 code and provides a simple *make* feature which help port the code to different platforms and architectures.

Program summary

Program title: RATIP

Catalogue identifier: AEMA_v1_0

Program summary URL: http://cpc.cs.qub.ac.uk/summaries/AEMA_v1_0.html

Program obtainable from: CPC Program Library, Queen's University, Belfast, N. Ireland

Licensing provisions: Standard CPC licence, <http://cpc.cs.qub.ac.uk/licence/licence.html>

No. of lines in distributed program, including test data, etc.: 256 562

No. of bytes in distributed program, including test data, etc.: 4975 979

Distribution format: tar.gz

Programming language: ANSI standard Fortran 90/95 and subsequent developments

Computer: PCs and workstations

Operating system: Suse, Debian and Ubuntu Linux

RAM: Memory requirements strongly depend on the size of the bound-state wave functions, the property considered as well as the special features selected during the computations.

Word size: All real variables are parametrized by a **selected kind parameter** and, thus, can easily be adapted to any required precision as supported by the compiler. Presently, the kind parameter is set to double precision (two 32-bit words) in the module `rabs_constant`.

Classification: 2.1, 2.9

Subprograms used:

Cat Id	Title	Reference
ADCU_v1_0	GRASP92	CPC 94 (1996) 249

Nature of problem: Ab-initio calculations of atomic properties and data are required in science and technology, not just within the traditional areas of astro and plasma physics but also in several recently emerging research fields. Hereby, often quite different demands arise with regard to the accuracy of the data, the elements of interest as well as their stage of ionization. Therefore, it is desirable to provide

[☆] This paper and its associated computer program are available via the Computer Physics Communications homepage on ScienceDirect (<http://www.sciencedirect.com/science/journal/00104655>).

* Correspondence to: GSI Helmholtzzentrum für Schwerionenforschung, D-64291 Darmstadt, Germany.

E-mail address: s.fritzsche@gsi.de.

a code which is applicable to all elements of the periodic table and which can help incorporate the dominant electron–electron correlation and relativistic effects on equal footings into the computations.

Solution method: Atomic bound-state wave functions from GRASP92 [1] for different levels and charge states are combined with continuum orbitals to calculate many-electron transition amplitudes and properties as derived from these amplitudes. Three major types of transition amplitudes refer to the electron–electron interaction, based on the Dirac–Coulomb–Breit Hamiltonian, the electron–photon interaction for the coupling of atoms to the radiation field as well as the electron–nucleus (hyperfine) interaction due to the electric and magnetic multipole fields of the nucleus. Apart from the electric-dipole approximation to the electron–photon interaction, this includes also other – electric and magnetic – multipole components of the radiation field. All computations are performed within the framework of the multiconfiguration Dirac–Fock (MCDF) method as implemented in GRASP92 [1] and its recent successors [2].

Restrictions: Relativistic calculations of atomic properties are restricted mainly by the size of the wave functions and the (virtual) excitations that can be taken into account with regard to a given set of reference configurations. Further restrictions of the present implementation concern:

- Despite the relativistic formulation of atomic properties based on Dirac's equation, all calculations are performed within the *no-pair* approximation; no attempt has been made to incorporate contributions from the negative continuum or radiative corrections beyond some simple estimate of the vacuum polarization and the electron self-energy to the level energies.
- Continuum orbitals are always generated within a static potential (of the corresponding ionic core) and are utilized to construct distorted waves with well-defined total angular momentum and parity. No continuum (interchannel) interactions are taken into account in the construction of scattering states if one (or more) electrons is in the continuum.
- As in GRASP92 [1], antisymmetric subshell states with more than two equivalent electrons are supported only for $j \leq 9/2$.
- If wave functions are defined with regard to different configuration lists to represent, for example, the initial and final state of a selected photo- or autoionizing transition, the same order of atomic orbitals (and usually also the same core) has to be used for generating the atomic bound states. The program terminates with an error message if this is not the case.
- The use of non-orthogonal orbital sets for the representation of initial, intermediate or final atomic states is supported only by a few selected programs, while “orthogonality” is assumed otherwise for the evaluation of the many-electron amplitudes apart from the active electrons.

Unusual features: The RATIP program is designed as a suite of *programs* where each of them help calculate one or a few closely related atomic properties, and for a given set of atomic levels. To make use of these programs, it is usually assumed that the wave functions for all bound states have been generated *before* by means of the GRASP92 [1] or some equivalent code. However, a clear and simple interface is made between the computation of the bound states and their use within the RATIP program [3] by applying only the (standard) input and output files from GRASP92, such as the definition of nuclear parameters (.iso), configuration lists (.csl), radial orbitals (.rwl) and mixing coefficient (.mix) files.

To specify the bound states of interest, most calculations within the RATIP program refer to the level numbers as they (do) occur in GRASP92 for a given configuration basis. Care has been taken that this selection and reference to the atomic levels can be handled flexibly but with some proper tests on the atomic property under consideration. Each program component of RATIP is controlled by an interactive dialog at the beginning of its execution and enables the user to select individual transitions as well as the particular mode of computation. All major results are usually compiled in tables and printed to some summary file, which is specific to each component. The units of energies, rates and cross sections in these tabulations can be specified during the input (from a number of possible choices) if the default is considered not to be appropriate.

Various (modern design) principles of Fortran 90/95 have been applied in developing the RATIP code [4], including the use of modules, the definition of derived data structures, the use of logical flags and the dynamic allocation of all important arrays. Therefore, there are no serious restrictions with regard to the number of open shells, nor to the grid size or the number of atomic transitions that can be calculated within a single run of some component. While some of RATIP's code is common to all programs and is provided by a number of core modules, each component usually refers also to some *own(ed)* data structures and procedures which are specific to its application.

Running time: 20 minutes on a standard laptop for all test cases.

References:

- [1] F.A. Parpia, C.F. Fischer, I.P. Grant, Comput. Phys. Commun. 94 (1996) 249.
- [2] P. Jönsson, X. He, C. Froese Fischer, I.P. Grant, Comput. Phys. Commun. 177 (2007) 597.
- [3] S. Fritzsche, J. Elec. Spec. Rel. Phen. 114–116 (2001) 1155.
- [4] M. Metcalf, J. Reid, Fortran 90/95 Explained, Oxford University Press, 1996.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

The recent years have seen a steadily increasing demand for accurate atomic data owing, for instance, to the development of new light sources and the remarkable progress in experimental and detector technology. Apart from the traditional research fields of astro and plasma physics (and their ongoing international research and infrastructure projects), accurate atomic data are required today also in more

recently emerging areas, such as atomic clocks, laser spectroscopy, quantum optics and metrology, free-electron laser research, VUV and X-ray lithography, or even material science, to name just a few [1]. In these research fields, however, quite different demands arise with regard to the accuracy of data, the elements of interest from the periodic table, or their stage of ionization and, more often than not, different theoretical data need to be combined with further information about the process of interest or due to effects of the environment.

To provide such atomic data for most (or all) elements from the periodic table, the multiconfiguration Dirac–Fock (MCDF) method has been found a very versatile tool for calculating a variety of properties [2]. When compared with other advanced many-body techniques, such as the random-phase approximation, many-body perturbation theory or the coupled-cluster approach, the MCDF method has the great advantage that it can be applied quite easily to excited and open-shell structures across the whole periodic table. Moreover, the MCDF method gives rise also to many-electron amplitudes which can be readily combined with each other for studying angle- and polarization-dependent properties and can be applied also to problems from the – time-independent or time-dependent – density matrix theory [3,4]. These advantages, together with the increasing quest for calculating ionization and capture processes with one (or more) electrons in the continuum, make it desirable to have a *computer code* available for relativistic calculations of atomic properties.

Here, I present (and provide) the RATIP program [5] for the calculation of various relativistic atomic transition, ionization and recombination properties within the MCDF model. This code has been designed as a suite of programs for open-shell atoms, including level energies, transition probabilities, Auger parameters, photoionization cross sections and angular parameters, radiative and dielectronic recombination rates, as well as various other atomic properties. The RATIP program is based on GRASP92 [6], the well-known and general-purpose atomic structure code, which can be used to generate the atomic (or ionic) bound states. These bound-state wave functions from GRASP92 are then combined, if appropriate, with properly constructed continuum orbitals in order to calculate the requested many-electron transition amplitudes and properties mentioned above. The RATIP program is presently designed as a scalar FORTRAN 90/95 code and utilizes a simple *make* feature to help port the code to different platforms and architectures.

In the next section, I shall briefly introduce the basic formulas and notations on which the RATIP program is built. Apart from a brief account of the relativistic Hamiltonian and the MCDF method itself, this includes the definition and use of the interaction and transition amplitudes that serve as the *building blocks* of the RATIP program. These amplitudes are central to most components and allow for simple extensions of the code. The general structure and organization of the program will be described in Section 3, beginning with a brief overview and the discussion of some guidelines which I followed in designing and implementing the code (Section 3.2). Apart from a few notes about Fortran 90/95 in implementing the RATIP program, this subsection also outlines the use of interface files as well as the interactive control of the program, while all *main* components of the code are described in Section 3.3 and the associated *toolbox* in Section 3.4. For each component, which is made available by the present distribution, I shall describe the implemented amplitudes or interactions, the interactive dialog at the beginning of the program's execution, the results and summary of the particular component, a few special features and reference to some recent applications. Because of the overall size of the program, however, no attempt has been made to describe neither the implementation of the code in detail nor to list all underlying (sub-)procedures. Many further features of the program can be understood and utilized by exploring the interactive dialogs of the various components. A few selected test examples are displayed in Section 4 to provide the user with a quick *test* of the installation, and a summary and outlook is finally given in Section 5.

2. Basic notations from atomic structure theory

The relativistic atomic structure theory has been developed during the past four decades and has been presented in detail in a number of recent text books [2,7]. Therefore, we shall restrict here ourselves to just the basic notations and expressions as they occur in the description of the code below, and shall refer for all further information to the literature. Details on the MCDF methods can be found in Ref. [8], and especially for its implementation within the GRASP(92) program in Refs. [6,9,10] and for the MCDFGME code, a second widely-used implementation of this method, in Ref. [11]. Atomic units are used throughout if not stated otherwise.

2.1. Relativistic Hamiltonians and interactions

As in nonrelativistic quantum theory, most relativistic structure calculations are based on a Hamiltonian which describes a fixed number of particles, N . By making use of Dirac's theory for the motion of each electron, one therefore arrives readily at the (so-called) *Dirac–Coulomb* Hamiltonian

$$H_{\text{DC}} = \sum_i h_{\text{D}}(\mathbf{r}_i) + \sum_{i < j} \frac{1}{r_{ij}} \quad (1)$$

where the one-electron Dirac operator

$$h_{\text{D}}(\mathbf{r}) = c\boldsymbol{\alpha} \cdot \mathbf{p} + \beta c^2 + V_{\text{nuc}}(r) \quad (2)$$

describes the kinetic energy of the electron and its interaction with the (external) nuclear potential $V_{\text{nuc}}(r)$, and where the second term in expression (1) just refers to the static Coulomb repulsion between each pair of electrons. Apart from (quite) heavy elements, indeed, the Dirac–Coulomb Hamiltonian has been found appropriate to describe the low-lying levels structure of most elements and is utilized within the GRASP code(s) in order to *define* the self-consistent field (SCF) of the N bound electrons [6,10]. Let us mention here, however, that the Hamiltonian (1) by itself is not bound from below and thus requires special care to prevent the electrons to fall into the “Dirac sea”. This so-called *Brown–Ravenhall disease* has been discussed in good detail in the literature [12,13]. For Dirac Hamiltonians, however, it has been shown in the literature that a rigorous lower bound to the upper part of the spectrum exists (and a corresponding upper bound to the lower part of the spectrum), provided that the trial functions are taken from a properly chosen functions space; see Ref. [2], § 5.6. In the MCDF method, a *variational collapse* can thus be avoided in the SCF procedure to the Hamiltonian (1) if proper boundary conditions are imposed upon the one-electron orbital functions.

In a more rigorous quantum-electrodynamical (QED) treatment of the many-electron system, it can be shown that the pairwise interaction between the electrons i and j is better described by the sum

$$\sum_{i<j} v_{ij} = \sum_{i<j} \left(\frac{1}{r_{ij}} + b_{ij} \right) \quad (3)$$

of the static Coulomb repulsion and the Breit interaction (b_{ij}) to correct for the magnetic and retardation contributions due to the relativistic motion of the electrons. An effective operator for the Breit interaction can be derived in perturbation theory to the electron–photon coupling (with respect to the number of virtually exchanged photons) and gives rise in Coulomb gauge to the so-called *transverse* Breit operator [13,14]

$$b_{12} = -(\boldsymbol{\alpha}_1 \cdot \boldsymbol{\alpha}_2) \frac{\cos(\omega r_{12})}{r_{12}} + (\boldsymbol{\alpha}_1 \cdot \nabla_1)(\boldsymbol{\alpha}_2 \cdot \nabla_2) \frac{\cos(\omega r_{12} - 1)}{\omega^2 r_{12}}, \quad (4)$$

and where $\omega = \frac{|\epsilon_1 - \epsilon_2|}{c}$ refers to the difference of the corresponding one-particle energies. From this expression, the *frequency-independent* (and original) Breit operator [15]

$$b_{12}^0 = -\frac{1}{2r_{12}} \left[\boldsymbol{\alpha}_1 \cdot \boldsymbol{\alpha}_2 + \frac{(\boldsymbol{\alpha}_1 \cdot \mathbf{r}_{12})(\boldsymbol{\alpha}_2 \cdot \mathbf{r}_{12})}{r_{12}^2} \right] \quad (5)$$

is obtained within the long-wavelength approximation, $\omega \rightarrow 0$. This zero-frequency approximation to the full transverse interaction neglects all contribution $\sim \alpha^4 Z^3$ (as well as in higher order in αZ) but is well suited for most computations of many-electron atoms and ions since the explicit frequency-dependence of expression (4) gives usually rise to (very) small corrections. It is therefore the frequency-independent form of the Breit interaction that is implemented in most components of the RATIP program.

If the Breit interaction is taken into account in expression (1), one obtains the *Dirac–Coulomb–Breit* Hamiltonian

$$H_{\text{DCB}} = \sum_i h_{\text{D}}(\mathbf{r}_i) + \sum_{i<j} \left(\frac{1}{r_{ij}} + b_{ij} \right) \quad (6)$$

which is often used in first-order perturbation theory to incorporate the relativistic contributions from the electron–electron interaction. The same form of the electron–electron interaction operator occurs in the computation of all Auger processes, the electron-impact excitation (or ionization) of atoms and ions as well as at several places elsewhere. Therefore, a simple notation and implementation of the corresponding interaction amplitudes will help to deal with further excitation and decay properties within the framework of the RATIP program. Further details about the electron–electron interaction amplitudes will be describe in Section 2.3 below.

In practice, the decision about the particular form of the Hamiltonian operator, i.e. of applying either the operator (1) or (6) or even some other form of the electron–electron interaction, is usually based upon physical arguments, such as the nuclear charge, the charge state of the ion, the shell structure of the atomic states of interest as well as the property which need to be considered. Apart from the proper choice of the Hamiltonian operator, the accuracy of most atomic calculation also depends very crucially on the many-electron basis (set) that is included into the computations. For a given wave function expansion, however, these two influences are difficult to discuss separately and often require a rather detailed analysis in order to monitor the convergence of some given property, if such a “monitoring” is possible at all. The specification of a suitable many-electron basis is indeed a *key* issue in generating (approximate) atomic bound and scattering states within the MCDF method and will now be described in some further detail.

2.2. The multiconfiguration Dirac–Fock method

Like in the symmetry-adapted nonrelativistic Hartree–Fock (HF) approximation, an atomic state function is written within the MCDF model as a linear combination of configuration state functions (CSF) with well-defined parity P , total angular momentum J , and its projection J_z

$$\psi_{\alpha}(PJM) = \sum_{r=1}^{n_c} c_r(\alpha) |\gamma_r P J M\rangle, \quad (7)$$

and where γ_r refers to all additional quantum numbers that are needed in order to specify the (N -electron) CSF uniquely. In ansatz (7), moreover, n_c denotes the number of CSF and $\{c_r(\alpha)\}$ the representation of the atomic state within the (given) CSF basis. In most standard computations, the set $\{|\gamma_r P J M\rangle\}$ of CSF are constructed as antisymmetrized products of a common set of *orthonormal* orbitals, making use of *jj*-coupling and a proper seniority scheme for the classification of the (antisymmetrized) subshell states [2]. In contrast to the HF (or Dirac–Fock approach for relativistic single-configuration calculations), however, both the radial (one-electron) functions as well as the expansion coefficients $\{c_r(\alpha), r = 1, \dots, n_c\}$ in the representation of the atomic state(s) are optimized simultaneously on the basis of the Dirac–Coulomb Hamiltonian. Further relativistic contributions due to the Breit interaction among the electrons or the polarization of the QED vacuum are then added perturbatively by diagonalizing the Dirac–Coulomb–Breit Hamiltonian (and vacuum polarization) matrix. Furthermore, estimates on the self-energy of the electrons involved in the representation of some state can be added to both, the level as well as transition energies.

While, at the first glance, ansatz (7) looks straightforward for generating the bound-state wave functions of some atom or ion, the proper choice of a physically *appropriate* basis $\{|\gamma_r P J M\rangle\}$ turns out to be less simple in practice and is often affected also by technical limitations with regard to the size and convergence of the wave functions. Especially shell structures with (several) open shells impose serious implications on the quality of the generated states. While the first successful structure calculations of simple atoms and ions in the 1960s and 1970s already led to surprisingly (accurate) results, in particular in view of the limited computational resources at that time, it soon became clear how sensitive many properties behave with regard to correlation effects, i.e. the contributions *beyond* the (single-configuration) Dirac–Fock approximation [16,17]. In bound-state calculations, therefore, the concept of either *complete* or *restricted* active spaces has been employed and found useful [18,19] but difficulties arise with open-shell structures or if the *electron continuum* need to be

included into the construction of atomic states. Here, I shall not discuss in detail how one can specify a proper CSF basis but *assume* that a reasonable representation of the bound-state functions is obtained by using the GRASP92 program [6] or some of its successors, and that this was done *before* the wave functions are to be applied within the RATIP program.

Standard MCDF calculations give rise to the energies and wave functions (E_α, ψ_α) for one or several selected bound states. From the symmetry of atoms and ions with regard to a rotation or inversion of the coordinates it is clear, moreover, that the Hamiltonian matrix is always block-diagonal in the total parity and angular momentum quantum numbers (PJM). Within the GRASP program(s), the generation of atomic bound states is therefore carried out in a series of steps which can be summarized as follows:

- Definition of nuclear parameters and the angular structure (and extent) of the CSF basis $\{\psi_\alpha(PJM)\}$ that is to be used in ansatz (7);
- Algebraic evaluation and computations of the spin-angular integrals (so-called angular coefficients) based on standard techniques from Racah's algebra;
- Initial estimates of all radial orbital functions and self-consistent field (SCF) calculations, based on the Dirac–Coulomb Hamiltonian;
- Configuration interaction (CI) calculations in order to incorporate further relativistic contributions into the Hamiltonian matrix and/or to enlarge the CSF basis beyond the given SCF model.

Often, these four steps are repeated at several stages by systematically enlarging the one-electron (active) orbital space.

While not much need to be said here about the explicit representation of the atomic and configuration states within the GRASP environment (see Ref. [2] for details), it is worth to consider in more detail the Hamiltonian matrix (elements) as perhaps the most frequently occurring interaction “amplitudes” within the RATIP program. For a given CSF basis, the representation of an atomic state in ansatz (7), i.e. the mixing coefficients $\mathbf{c}(\alpha) \equiv (c_1(\alpha), c_2(\alpha), \dots, c_{n_c}(\alpha))$, is obtained by solving the secular equation

$$\det(\mathbf{H} - E_\alpha^{(n_c)} \mathbf{I}) = 0, \quad (8)$$

where $E_\alpha^{(n_c)}(PJ)$ denotes the eigenvalue and

$$\mathbf{H} = (H_{rs}) = (\langle \gamma_r P J M | H | \gamma_s \bar{P} \bar{J} \bar{M} \rangle \delta_{P\bar{P}} \delta_{J\bar{J}} \delta_{M\bar{M}}) \quad (9)$$

the Hamiltonian matrix which is blockdiagonal in the total parity and angular momentum of the atom, independent of the particular choice of the electron–electron interaction in the Hamiltonian. Moreover, since the Hamiltonian matrix is real and symmetric, all atomic states are orthogonal for $E_\alpha \neq E_\beta$ or can be chosen in this way for $E_\alpha = E_\beta$. In practice, the decomposition of the many-electron matrix elements H_{rs} in expression (9) in central to every implementation of the MCDF method. By performing the integration over the spin-angular variables of all N electrons algebraically, it was shown that the matrix elements can always be written in the form [8]

$$H_{rs} = \sum_{ab} u_{rs}(ab) \langle a || h_D || b \rangle + \sum_L \sum_{abcd} v_{L;rs}(abcd) X_L(abcd). \quad (10)$$

In this expansion, $u_{rs}(ab)$ and $v_{L;rs}(abcd)$ are one- and two-particle (scalar) angular coefficients, while the $X_L(abcd)$ describe the effective interaction strengths of (formally) the four electrons a, b, c, d involved in the interaction. As usual, we here abbreviate the one-particle quantum numbers $a = (n_a, \kappa_a)$, $b = (n_b, \kappa_b)$, ... to have a compact notation for the subshells of equivalent electrons. The reduced matrix element in the first term of Eq. (10),

$$\langle a || h_D || b \rangle = \delta_{\kappa_a \kappa_b} \int_0^\infty dr \left[c Q_a \left(\frac{d}{dr} + \frac{\kappa_a}{r} \right) P_b + c P_a \left(-\frac{d}{dr} + \frac{\kappa_a}{r} \right) Q_b - 2c^2 Q_a Q_b + V_{\text{nuc}}(r) (P_a P_b + Q_a Q_b) \right], \quad (11)$$

contains the kinetic and potential energy of an electron in subshell a , and can be expressed in terms of the *large* and *small* (radial) components, $P_a(r)$ and $Q_a(r)$, of the corresponding one-electron orbital functions. These reduced matrix elements can be considered also as the one-particle *analogue* to the effective interaction strength $X_L(abcd)$ which will be further discussed in the following subsection. A similar decomposition of the many-electron matrix elements (amplitudes) as in Eq. (10) into a sum of *angular coefficient* \times *interaction strength* can be made for every physically relevant (interaction) amplitude, independent of the particular rank of the corresponding interaction operator. Since most atomic processes can be traced back to just a (very) few of such interaction or transition amplitudes, we made use of them as the *building blocks* in the design and implementation of the RATIP program.

Of course, the generalized angular coefficients in the decomposition of the many-electron amplitudes depend on the rank and parity of the requested interaction operator, and special programs exist within the GRASP environment to calculate these coefficients [20,21]. Typically, moreover, the (multiple) summation over all possible subshells a, b, \dots and (intermediate) ranks L in Eq. (10) is replaced by a single index [cf. Eqs. (15) and (16)] which runs through all non-vanishing coefficients of the corresponding type. We also note that the ranks L of the interaction strengths $X_L(abcd)$ that appears in Eq. (10) depend on the particular choice of the electron–electron interaction.

2.3. The concept of interaction and transition amplitudes

The (scalar) Hamiltonian matrix element (10) describes the interaction energy between the two CSF $|\gamma_r P J M\rangle$ and $|\gamma_s \bar{P} \bar{J} \bar{M}\rangle$, and these matrix elements can be readily utilized to describe the total (configuration) interaction of two given atomic states

$$\langle \psi_\alpha | H | \psi_\beta \rangle = \sum_{rs} c_r^*(\alpha) H_{rs} c_s(\beta). \quad (12)$$

In particular, $\langle \psi_\alpha | H | \psi_\alpha \rangle = E_\alpha$ is the total energy of level α . Using time-dependent perturbation theory and Fermi's Golden rule [2], moreover, most atomic properties can be written rather similarly in terms of (so-called) many-electron interaction amplitudes

$$\langle \psi_\alpha | T | \psi_\beta \rangle = \sum_{rs} c_r^*(\alpha) T_{rs} c_s(\beta), \quad (13)$$

and where T is then called the transition (or interaction) operator. Furthermore, most of these transition operators can be expressed in terms of spherical tensor operators of rank K (and magnetic projection Q) which transform under rotations like the spherical harmonics $Y_{KQ}(\theta, \phi)$: $T = \sum_{KQ} a(K, Q) T_{KQ}$. Instead of the complete matrix elements, that include the magnetic projection M of the given atomic or configuration states and tensor operators, one makes often use of the *reduced* matrix elements

$$T_{KQ;rs} \equiv \langle \gamma_r P_r J_r M_r | T_{KQ} | \gamma_s P_s J_s M_s \rangle = \langle J_r M_r, K Q | J_s M_s \rangle \langle \gamma_r P_r J_r || T_K || \gamma_s P_s J_s \rangle \quad (14)$$

by applying the Wigner–Eckart theorem [22]. Although these reduced transition amplitudes occur quite frequently in the literature in order to discuss and analyze atomic processes, not many codes exploit these amplitudes as the *natural* building blocks for describing the level structure and properties of (open-shell) atoms and ions. More often than not, instead, a decomposition of these transition amplitudes into various types of one- and two-particle (reduced) matrix elements or even radial integrals is made *well before* any implementation or coding is considered. Indeed, the different techniques that are used in the decomposition of the many-electron amplitudes and the large number of definitions (and notations) of angular- or radial-type integrals in the literature has hampered not only the comparison of different codes during the last decades but made it difficult also to (re-)use these entities for calculating other properties than of what the program developers considered originally. In the RATIP program, in contrast, we make (quite) consequent use of the many-electron amplitudes $\langle \psi_\alpha | T_{KQ} | \psi_\beta \rangle$ and $\langle \psi_\alpha || T_K || \psi_\beta \rangle$ as the central building blocks in describing the interaction among the electrons as well as with external particles and fields. This concept has helped simplify and maintain the code. Moreover, it (will) enables us to exploit the many-electron interaction and transition amplitudes in second- and higher-order processes, once an appropriate (intermediate) basis $\{\psi_\nu(PJM), \nu = 1, \dots, n_\nu\}$ has been constructed for some given process.

The reduced matrix elements of spherical tensor operators take the form

$$\langle \gamma_r P_r J_r || T_K || \gamma_s P_s J_s \rangle = \sum_t u(a_t b_t; K) X(a_t b_t; K) \quad (15)$$

for one-particle operators $T_K = \sum_i u_K(i)$ and

$$\langle \gamma_r P_r J_r || T_K || \gamma_s P_s J_s \rangle = \sum_t v_{L_t}(a_t b_t c_t d_t; K) X_{L_t}(a_t b_t c_t d_t; K) \quad (16)$$

for (symmetric) two-particle operators $T_K = \sum_{i < j} v_K(i, j)$, and where $X(a_t b_t; K)$ and $X_{L_t}(a_t b_t c_t d_t; K)$ refer again to the corresponding one- and two-particle (effective) interaction strengths, respectively. Of course, these interaction strengths are specific to the particular transition operator under consideration, and care has to be taken that the same phase convention applies if different interaction amplitudes are to be combined with each other.

In practice, there are two steps in the computation of the transition amplitudes (15) and (16) that are performed separately: The (pure) angular coefficients are usually calculated by means of the ANCO component for either a pair or a whole set of CSF [21], and all the non-vanishing coefficients are then returned in a *derived* list structure to the calling procedure. For these coefficients, the associated one- or two-particle interaction strengths are evaluated and summed up to form the requested matrix element T_{rs} of the transition matrix. To decrease the computational effort, these (one- and two-particle) interaction strengths are also *stored* and *re-utilized* in a few components; cf. Section 3.3 below.

To make further use of the many-electron (transition) amplitudes (15) and (16), most components of the RATIP program now provide a so-called transition amplitude (.trn) file. Apart from the amplitudes of interest, these files contain all quantum numbers that are required for their unique specification and can thus be utilized to exchange the physically relevant data between different program components. In the future, moreover, we expect to see applications of the RATIP program where one starts directly from the transition amplitudes files (calculated for either one or several elementary processes and interaction operators) and combines these amplitudes in order to generate data for more elaborate processes. A few first steps into this direction have been performed during the last years by studying the coherence transfer through the two-step Auger cascades of the noble gases after inner-shell excitation [23] or the dielectronic recombination of multiple and highly charged ions [24,25]. Moreover, these many-electron amplitudes are often also the *natural* starting point if atomic structure calculations are to be combined with either the – time-independent or time-dependent – density matrix theory [3,4].

In the RATIP program, there are three (elementary) types of interaction amplitudes that are central to the code:

2.3.1. Electron–electron interaction

The interaction among the electrons is described by the scalar operator [Eq. (3)]

$$V = V^{\text{Coulomb}} + V^{\text{Breit}} = \sum_{i < j} \left(\frac{1}{r_{ij}} + b_{ij} \right) \quad (17)$$

as it occurs in the Dirac–Coulomb–Breit Hamiltonian (6). This operator gives rise to the (reduced) interaction amplitudes

$$\langle \gamma_r P_r J_r || V || \gamma_s P_s J_s \rangle = \langle \gamma_r P_r J_r || V^{\text{Coulomb}} || \gamma_s P_s J_s \rangle + \langle \gamma_r P_r J_r || V^{\text{Breit}} || \gamma_s P_s J_s \rangle, \quad (18)$$

and, since for scalar operators ($K = Q = 0$), the Clebsch–Gordan coefficient in Eq. (14) evaluates to $\langle J_r M_r, 00 | J_s M_s \rangle = \delta_{J_r J_s} \delta_{M_r M_s}$, the full and reduced matrix elements coincide in this case with each other and need(ed) not to be distinguished in Section 2.2. The electron–electron interaction amplitudes (18) are utilized (and calculated) in various components of the RATIP program, including the AUGER, DIEREC, and RELCI components, and at several places elsewhere. For a given set of initial and final (hole) states, such nondiagonal amplitudes are written for instance to the `auger.trn` transition amplitude file as we shall discuss in Section 3.3.1 below.

Since the Coulomb and Breit interaction operators are both scalar operators and contribute *additively* to the total electron–electron interaction, the same (pure) angular coefficients $v_{L_i}(a_i b_i c_i d_i; K=0)$ occur in Eq. (16) in the evaluation of the amplitudes (18), although different angular and parity selection rules apply for the various interaction strengths. In the computation of these amplitudes, therefore, the Breit interaction can be added rather easily to the instantaneous Coulomb repulsion [cf. Eq. (10)]

$$X_L(abcd) = X_L^{\text{Coulomb}}(abcd) + X_L^{\text{Breit}}(abcd) \quad (19)$$

if this is appropriate for a given process. Instead of the Breit interaction, sometimes only the first terms from the low-frequency Breit operator is taken into account

$$b_{12}^G = -\frac{\alpha_1 \cdot \alpha_2}{2r_{12}}, \quad (20)$$

that describes the current–current interaction of the moving electrons and which is known as *Gaunt interaction* in the literature. For closed-shell atoms, the Gaunt interaction yields about 90% of the total (relativistic) energy shifts to the atomic levels [26], and gives rise to the corresponding effective interaction strength $X_L^{\text{Gaunt}}(abcd)$.

Here, we shall not discuss how these effective interaction strength can be calculated in detail for the various interaction operators, and by making use of the one-electron orbital functions a, b, c and d . For the Coulomb interaction, for example, the effective strength is given by [2]

$$X_L^{\text{Coulomb}}(abcd) = \delta(j_a, j_c, L) \delta(j_b, j_d, L) \Pi^e(\kappa_a, \kappa_c, L) \Pi^e(\kappa_b, \kappa_d, L) (-1)^L \langle \kappa_a \| \mathbf{C}_L \| \kappa_c \rangle \langle \kappa_b \| \mathbf{C}_L \| \kappa_d \rangle R_L(abcd) \quad (21)$$

where $R_L(abcd)$ denotes the relativistic Slater integral,

$$\Pi^e(\kappa_a, \kappa_b, L) = \begin{cases} 1 & \text{if } l_a + l_b + L \text{ even,} \\ 0 & \text{otherwise} \end{cases} \quad (22)$$

reflects the (even-parity) selection rule for the Coulomb interaction, and where the “triangular delta” $\delta(j_a, j_b, j_c)$ ensures the proper coupling of the one-electron angular momenta; this delta symbol is $\delta(j_a, j_b, j_c) = 1$ if j_a, j_b and j_c satisfy the triangular condition, and is zero otherwise. Similar, although more elaborate, expressions also arise for the Breit and Gaunt interactions (4) and (20) in the decomposition of the total electron–electron interaction [2].

2.3.2. Electron–photon interaction

The (relativistic) interaction of an electron with the radiation field has been frequently discussed in the literature and is described by the one-particle operator

$$\mathcal{R}_\lambda(\mathbf{k}) = \sum_i \alpha_i \cdot \mathcal{A}_{\lambda,i}(\mathbf{k}) = \sum_i \alpha_i \cdot \mathbf{u}_{\lambda,i} e^{i\mathbf{k} \cdot \mathbf{r}_i}. \quad (23)$$

where $\alpha_i = (\alpha_{i,x}, \alpha_{i,y}, \alpha_{i,z})$ denotes the vector of the Dirac matrices and $\mathcal{A}_{\lambda,i}(\mathbf{k})$ the vector potential of the radiation field (with polarization λ) as seen by the i -th electron. In this notation, λ denotes the helicity of the photons which are said to be right-circularly polarized for $\lambda = +1$ and left-circularly polarized for $\lambda = -1$, respectively. The further decomposition of the operator $\mathcal{R}_\lambda(\mathbf{k})$ into spherical tensor operators, i.e. the well-known electric and magnetic multipole fields, depend however on the particular choice of the quantization axis. In the helicity representation of the photon ($\mathbf{e}_z \| \mathbf{k}$), the vector potential is given by [22]

$$\mathcal{A}_\lambda(\mathbf{e}_z \| \mathbf{k}) = \sqrt{2\pi} \sum_L i^L [L]^{1/2} (\mathbf{A}_{L\lambda}^{\text{magnetic}} + i\lambda \mathbf{A}_{L\lambda}^{\text{electric}}) = \sqrt{2\pi} \sum_L \sum_{\pi=0,1} i^L [L]^{1/2} (i\lambda)^\pi \mathbf{A}_{L\lambda}^\pi \quad (24)$$

where, in the second line, $\pi = 0$ refers to the magnetic and $\pi = 1$ to the electric multipoles.

For photoionization or radiative capture processes of electrons, in contrast, it is often more convenient to choose the electron momentum as *quantization* axis and, thus, to rotate the vector potential (24) in space so that its z -axis coincides with the electron momentum. This requires then to include the Wigner rotation matrices of rank L , $D_{M\lambda}^L(\mathbf{k} \rightarrow \mathbf{e}_z)$, which has to be applied independently to each irreducible tensor component of the field, but has the advantage that the spin-angular integration can be carry out analytically by using the techniques from Racah's algebra [27].

In the RATIP program, we evaluate the radiative *bound–bound* transition amplitudes

$$\langle \psi_\alpha(PJ) \| H_\gamma(\pi L) \| \psi_\beta(\bar{P}\bar{J}) \rangle = \left\langle \psi_\alpha(PJ) \left\| \sum_i \alpha_i \cdot \mathbf{A}_{L,i}^\pi(\mathbf{k}) \right\| \psi_\beta(\bar{P}\bar{J}) \right\rangle \quad (25)$$

and the *bound–free* amplitudes

$$\langle \psi_\alpha(\tilde{P}\tilde{J}, \epsilon l j; PJ) \| H_\gamma(\pi L) \| \psi_\beta(\bar{P}\bar{J}) \rangle = i^{-l} e^{-i\Delta_\kappa} \left\langle \psi_\alpha(\tilde{P}\tilde{J}, \epsilon l j; PJ) \left\| \sum_i \alpha_i \cdot \mathbf{A}_{L,i}^\pi(\mathbf{k}) \right\| \psi_\beta(\bar{P}\bar{J}) \right\rangle \quad (26)$$

for the multipole fields (πL), and together with the proper phase for the *free* (outgoing or incoming) electron if we shall consider the ionization or capture of an electron ($\epsilon l j$) due to its interaction with the radiation field. The phase Δ_κ hereby includes both, the Coulomb phase as well as the phase shift due to the *non-Coulombic* part of the corresponding bound-state potential in which the continuum orbitals are generated. In practice, however, only phase differences are important in all computations. The reduced matrix elements (25) and (26) are taken again as building blocks in order to represent and to discuss a large number of atomic properties. An efficient evaluation and access to these matrix elements has been found crucial for studying different properties of few- and many-electron ions. Note that by

using the reduced form of the transition matrix, the building blocks (26) are independent of the particular choice of the quantization axis, i.e. the rotation matrices $D_{M\lambda}^L$ reside within the expressions of the transition matrix or statistical tensors and can be computed independently [28,29].

Apart from the boundary conditions, which are specific to the photoionization or radiative recombination of electrons, not much need to be said here about the further evaluation of these reduced matrix elements. They can be decomposed like the matrix elements (15) into (a sum of) products of *angular coefficient* \times *interaction strength*, where the (pure) angular coefficients $u(a_t b_t; L)$ are obtained internally from some adapted version of the MCT component of GRASP92 [6] and the reduced one-particle interaction strength $X(a_t b_t; L)$ were described in Ref. [8].

2.3.3. Electron–nucleus (hyperfine) interaction

Beside of the attractive (electric) field of the nucleus with charge Ze , each atomic electron also interacts with the electric and magnetic (multipole) fields of the nucleus if it has spin $I > 0$. This interaction is better known as “hyperfine interaction” in atomic physics whose two dominant contributions arise from the nuclear magnetic-dipole field $\mathbf{A} = \frac{\boldsymbol{\mu} \times \mathbf{r}}{r^3}$ and the electric-quadrupole field $\Phi(r) = \sum_{ij} \frac{r_i r_j}{2r^5} Q_{ij}$, respectively. In these expressions, $\boldsymbol{\mu}$ is the nuclear magnetic moment operator and Q_{ij} are the Cartesian components of the nuclear quadrupole operator. Higher magnetic and electric multipole fields may generally occur for nuclei with spin $I > 1$ but are typically negligible in their contribution to the hyperfine splitting of atomic levels [30].

For many-electron systems, the relativistic hyperfine interaction Hamiltonian can be written as

$$H_{\text{hfs}} = \sum_K M_K^{(n)} \cdot T_K^{(e)} \quad (27)$$

where $M_K^{(n)}$ and $T_K^{(e)}$ represent the spherical tensor operators of rank K from the nucleonic and electronic sector, respectively. For the combined system “electron + nucleus”, then, the hyperfine states can be formed as linear combination

$$\tilde{\psi}_\alpha(I, P; FM_F) = \sum_{r=1} \tilde{c}_r(\alpha) |I, \beta_r P J_r; FM_F\rangle \quad (28)$$

of hyperfine (basis) states of the same total angular momentum F (as well as the same electronic parity P) and where $\{\tilde{c}_r(\alpha)\}$ denotes the representation of these states in the hyperfine-coupled basis [31]

$$|I, \beta P J; FM_F\rangle = \sum_{M_I, M_J} |IM_I\rangle |\psi_\alpha(PJM_J)\rangle \langle IM_I JM_J | FM_F\rangle. \quad (29)$$

As for the atomic state functions (7), the representation $\{\tilde{c}_r(\alpha)\}$ of an atomic hyperfine state (28) is obtained by diagonalizing the Hamiltonian $H = H_{\text{DF(B)}} + H_{\text{hfs}}$ of the combined system “electrons + nucleus”. For this hyperfine part of this Hamiltonian, the matrix elements can be written after some standard angular momentum algebra as

$$\langle I, \beta P J; F | H_{\text{hfs}} | I, \tilde{\beta} \tilde{P} \tilde{J}; \tilde{F} \rangle = \delta_{P P'} \delta_{F F'} (-1)^{I+J+F} \sum_K \left\{ \begin{matrix} I & J & F \\ J & I & K \end{matrix} \right\} \langle \psi_\beta(PJ) \| T_K^{(e)} \| \psi_{\tilde{\beta}}(\tilde{P}\tilde{J}) \rangle \langle I \| M_K^{(n)} \| I \rangle, \quad (30)$$

and if nuclear excitations are ignored right from the beginning.

While the reduced nuclear matrix elements are determined geometrically by

$$\langle I \| M_1^{(n)} \| I \rangle = \mu \sqrt{\frac{(I+1)(2I+1)}{I}}, \quad (31)$$

$$\langle I \| M_2^{(n)} \| I \rangle = \frac{Q}{2} \sqrt{\frac{(I+1)(2I+1)(2I+3)}{I(2I-1)}}, \quad (32)$$

the corresponding electronic amplitudes require detailed atomic structure calculation. For an N -electron atom or ion, the electronic tensor operators are given by [32]

$$T_1^{(e)} = \sum_j^N -i\alpha \frac{(\boldsymbol{\alpha}_j \cdot \mathbf{l}_j) \mathbf{C}_1(j)}{r_j^2} \quad (33)$$

for the interaction with the magnetic-dipole field of the nucleus and

$$T_2^{(e)} = \sum_j^N -\frac{\mathbf{C}_2(j)}{r_j^3}, \quad (34)$$

for the electric-quadrupole field, and where α is the fine-structure constant, $\boldsymbol{\alpha}_i$ the vector of Dirac matrices as well as the C_K are (normalized) spherical tensors as before.

In the RATIP program, we compute the matrix elements (30) for both the magnetic-dipole and the electric-quadrupole operators in the electron–nucleus interaction. These matrix elements are often referred to as *hyperfine amplitudes* and are provided in the `.trn` transition amplitude file of the Hfs component. Although this component is not (yet) distributed with the present version of the RATIP program, these amplitudes are utilized, for example, to set-up and diagonalize the Hamiltonian matrix for $H = H_{\text{DC(B)}} + H_{\text{hfs}}$ in order to determine

the representation $\{\tilde{c}_r(\alpha)\}$ of the hyperfine states (28) in the hyperfine-coupled basis (29). They are also applied to compute hyperfine-quenched transition amplitudes and lifetimes [31], and are presently employed to better understand hyperfine-induced processes in highly-charged ions. For the computation of A and B hyperfine parameters for individual levels or radiative transitions, in contrast, the Hfs92 component [32] of GRASP92 is more suitable for systematically enlarged wave functions and became now part also of the GRASP2K code [10]. Moreover, a hyperfine-coupled basis is applied also in the HfsZEEMAN program [33] and can there be utilized for studying atomic transitions in an external magnetic field [34].

3. Structure of the program

3.1. Overview to the RATIP program

The RATIP program has been developed during the past 15 years in order to support the computation of different atomic properties for atoms and ions, and (almost) independent of their particular shell structure. Based on the Dirac–Coulomb(–Breit) Hamiltonian and the MCDF method [8], as briefly outlined above, the program help incorporate into the computations the dominant electron–electron correlation and relativistic effects on quite equal footings. Therefore, the RATIP program has been found applicable to almost all elements from the periodic table as well as to multiply and highly charged ions. Fig. 1 displays the overall structure of the code and those atomic processes (properties) which are presently supported by the program. Although RATIP is tightly bound to the GRASP92 program [6], owing to its use of the bound orbitals and the representation of atomic bound states from this code, it has been developed and maintained independently by making use of a clear *interface* between these two codes as I shall explain below. This interface makes it possible also to apply rather easily (if necessary) the bound-state wave functions from other versions of the GRASP code, although this feature is presently not supported.

Like the GRASP92 program and as seen from Fig. 1, RATIP consists of a suite of individual programs which need to be compiled and called separately. Each of these components serve for one or a few well-defined tasks in the calculation of atomic amplitudes and properties. Among the main components, for example, RELCI [35] supports the set-up and diagonalization of large Hamiltonian matrices, REOS [36] the calculation of transition probabilities, Einstein coefficients and (radiative) lifetimes, PHOTO the evaluation of photoionization cross sections and amplitudes, or AUGER the computation of various autoionization properties, just to mention a few. Apart from these main components (with a solid frame in Fig. 1), there are several *auxiliary* components (with dashed frames), such as ANCO [21], COWF or RCFP [37] which can be invoked separately but mainly act as *backbone* in order to facilitate more complex computations. Moreover, a TOOLBOX to the RATIP program now supports a number of (more or less) small but frequently occurring tasks in manipulating or printing out some intermediate data in course of the computations. All main components, that are distributed by the present version of the program, will be explained and summarized below in Section 3.3. Other components, such as COULEX for the calculation of Coulomb excitation rates and the alignment of ions, DIEREC for dielectronic recombination rates and resonance strengths, or SCHIFF for the evaluation of parity- and time-reversal violating interaction amplitudes and parameters, have been developed only recently and are not (yet) well enough tested for distribution. These components are shown with a dashed-dotted frame in Fig. 1 and will be made available only in some later version of the RATIP program.

While there exist already a long write-up for a few components within the *Computer Physics Communication* library [35–37], various changes have been made to this code over the years to correct for bugs and to keep it overall consistent. Similar as in the GRASP environment, the data exchange between the components is realized by a number of – input and output – files for which a short description of their the purpose and name convention is given below in Section 3.2. For a program of RATIP's size, however, it is obvious that not all (technical) details can be discussed in this write-up and that not always the same standard in the design of the program can be ensured for all components simultaneously. At present, the whole RATIP program is based on about 60 modules which contain the source code and which comprise in total about 800 (sub-)procedures and 110,000 lines of code from which approximately 60% is made public here to the user. In the next section, we shall first outline some of the general guidelines in designing and implementing the code which I have tried to follow over the years.

3.2. Guidelines for the design and implementation of the code

Since the mid 1990s, the RATIP program [5,38] has been designed as an *open* environment for analyzing atomic structures and properties. From the early days on, our intention was that new components and features can be added to the program rather easily and without the need for large changes to the existing code. Moreover, in order to make use of modern design principles, RATIP has been implemented within the framework of Fortran 90/95 and its descendants. Indeed, the use of Fortran 90/95 helped adapt the code to different compilers and platforms, although not much effort was spent in the past to port RATIP to other environments than those which became locally available (and necessary) over the years. By taking advantage of the Fortran 90/95 language standard, modern concepts were applied, such as modules, derived data types, data hiding, dynamic allocation and deallocation of memory, the specification of the numerical precision by proper `KIND` parameters as well as several others. Below, I shall explain several of these design principles in more detail and especially in view of the RATIP program. Today, one of our major concerns is the life-cycle of the code as well as improvements on the *object-oriented* design in order to make RATIP ready also for future applications.

A key principle for the implementation of the RATIP program has been the explicit use of *many-electron transition amplitudes*, wherever this appears reasonable. In several applications, the consequent use of these amplitudes helped implement new and efficient concepts into atomic structure theory. For example, these amplitudes are now exploited for studying the angular and polarization dependence of photon and/or electron emission processes, if combined with atomic density matrix theory (see [29] for references). A simple access to the many-electron amplitudes within either the time-independent or time-dependent density matrix theory might bring RATIP also to the frontier with regard to (photon-induced) short-pulse excitation, ionization and decay processes in many-electron atoms and ions.

3.2.1. Making use of the Fortran 90/95 standard

If compared with previous standards, Fortran 90/95 can be considered as rather a new language that has benefited from Fortran's long tradition and from the recent experience in designing programming languages [39]. Two important features of Fortran 90/95 (among

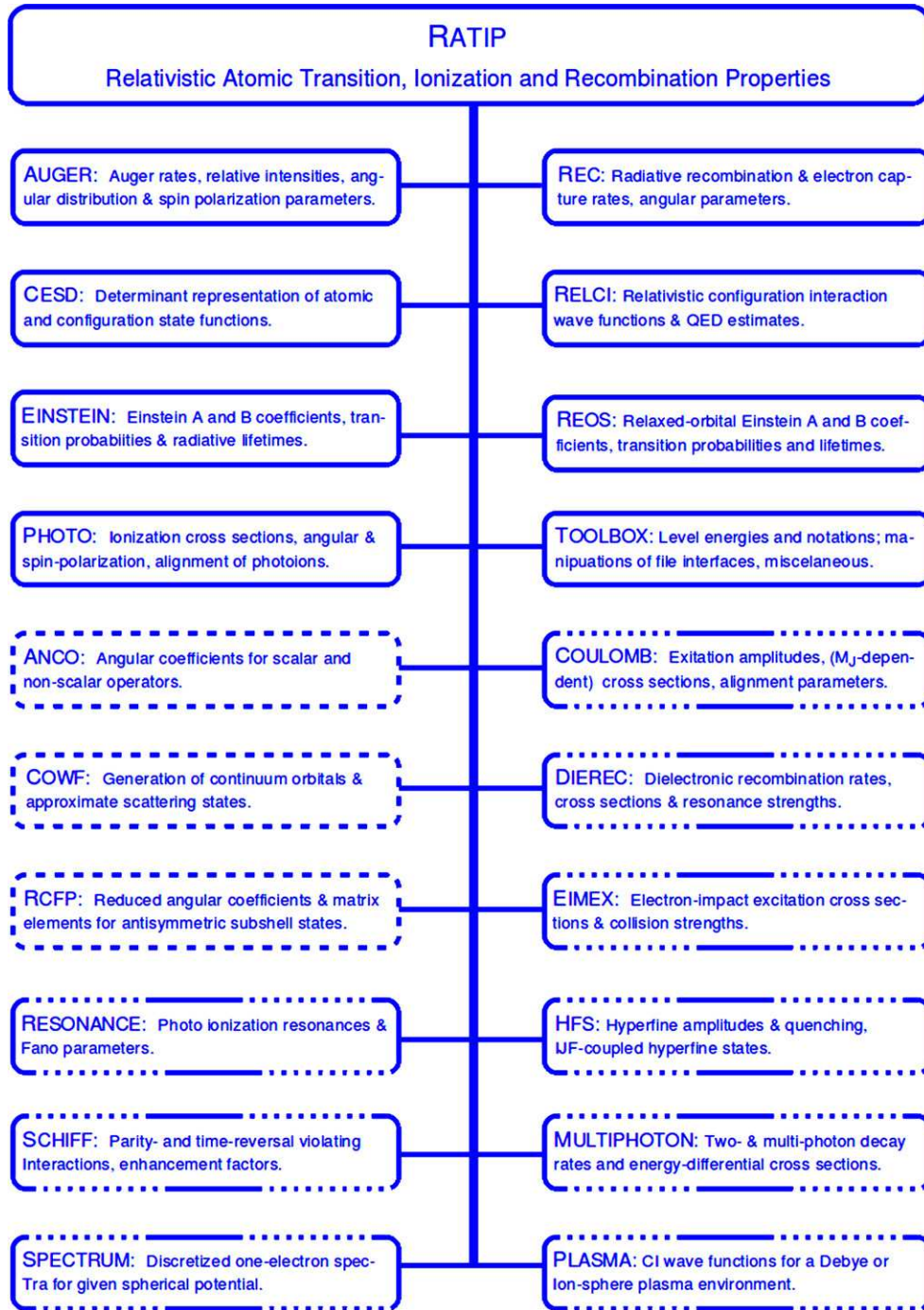


Fig. 1. Main structure of the RATIP program which is divided into individual program components. Apart from those components that are described below (solid frame), we also distribute some *auxiliary* components (dashed frame) in the present version that facilitate the evaluation of transition and interaction amplitudes. Several other components (dashed-dotted frame) are still under development or require further tests before they are considered for publication.

many others) have made this persistent language up-to-date with more recent language alternatives and concepts. They refer to the use of *modules* for generating protected and re-usable code and to *derived data types* that enables one to define flexible and well-adapted data structures for the research topic of interest. In the design of RATIP, we made elaborate use of these two and other features of Fortran 90/95 to move towards a more object-oriented access of the central entities in atomic structure theory.

Use of modules: This enables the user to include functionality from previous developments into new parts of the code without that the name space and the argument lists (of all subsequent calls) need to be understood in full detail. Modules also provide a simple structure for comprising and distributing code since they empower one to keep together the definition of closely related (global) variables and data, derived types as well as operations associated to these data structures. Within the RATIP program, modules are used in particular in two ways: (i) in order to establish and maintain a pool of subprocedures which deal with a number of frequently occurring tasks (and which form the *backbone* for all or, at least, many components), and (ii) to comprise all functionality that is specific for some given component. The use of a common backbone of code ensures a consistent treatment of data structures and elementary computational tasks within the

different program components; it also facilitates the combination of data in order to *derive* new data types. Here, we do not intent to list and explain all modules of the RATIP code separately but restrict ourselves to a few important ones. The following seven modules are central to almost all program components.

- rabs_constant*: contains the parametrization of all elementary data types by choosing proper *kind* parameters as well as the definition of a few important and sufficiently general data types. This module is therefore *used* by all other modules. The *kind* parameter *dp* for the parametrization of real variables in these data types is typically set to *double precision* on a standard 32-bit architecture in the module *rabs_constant* but can easily be re-defined to single precision for 64-bit processors.
- rabs_csl*: defines all major data types to deal with single CSF, sets of CSF and their use for representing atomic state functions (ASF). It contains also procedures to read in, copy or print out a configuration basis as a whole and to construct new basis sets.
- rabs_dirac_orbital*: contains variables and procedures that are related to the relativistic Dirac orbitals and their use in dealing with *one-particle* symmetry blocks.
- rabs_function_math*: comprises all procedures for the evaluation of elementary *math* functions and polynomials, such as the Bessel, (incomplete) Beta and Gamma functions or the Legendre polynomials as well as many others. It also contains functions for the transformation of coordinates, etc.
- rabs_function_string*: contains procedures for *string* manipulations together with the necessary data; string manipulation is utilized rather extensively in the code to support, for example, a simple input and output of angular momenta, multipoles, or sequences of level numbers during the execution of the program.
- rabs_grasp92*: contains several procedures from GRASP92 [6] that have been adapted or re-written to the internal use of the data structures and definitions. In GRASP92, these procedures were originally often part of the *lib92* library.
- rabs_nucleus*: stores all information about the nucleus and nuclear potential, and how this can be accesses during the computations.

Inside of each module, we defined a *common* style for the layout of the code: Following a short summary of the module and the list of use statements at the very beginning, the *specification part* of the module briefly describes all (module) procedures in alphabetic order. It also defines the data structure and logical flags that are specific to the module. Further details about the individual procedures can be found in the *contains part* in the header of each subprocedure; in this part, again, all functions and subroutines always occur in alphabetic order. In order to easily recognize the module in which a particular procedure is defined, the names of many module procedures begin with a string that just refers to the particular module; for example, the name of (almost) all procedures in *rabs_auger* also start with the prefix *auger_* . . . , and a similar convention is followed in many of the other components.

Dynamic storage allocation & use of pointers: While pointers and some dynamical allocation are used already in GRASP92 (as well as a number of previous Fortran dialects), the dynamic allocation of storage and arrays is now supported much more consistently in Fortran 90/95. This feature, combined with a careful use of derived data types, helps us to write code that is independent of the shell structure and size of the problem, i.e. independent of the number of atomic or configuration state functions, the parameters and size of the radial grid, or the number of transitions for some process under consideration. Almost all arrays are now allocated dynamically during run time and, hence, there are virtually no other limitations on the size or dimensions of the problem than given by the resources themselves.

Use of derived data types: This feature is essential for any *object-oriented* approach in the design and implementation of computer codes, and its importance cannot be overrated. A careful definition of such derived data types often decides how easily one can read and modify the code and, therefore, how modestly it can be adopted to some different task. While derived data types can be declared like structured variables on their own, they can be combined also to define derived types at some higher level of complexity. The definition and specification of properly derived types therefore affects the life-cycle of the program as a whole.

In RATIP, derived data types are utilized at three different levels (though there is some overlap between these levels):

- (a) Basic data structures: are utilized to refer to basic notations from (relativistic) atomic structure theory. Examples are `type(nkappa)` and `type(nkappam)` to simply define the “tupels” of quantum numbers (n, κ) and (n, κ, m_j) for atomic orbitals, or `type(orbital_function)` to comprise all information about the quantum numbers and the *large* and *small* (radial) components of an one-electron orbital function.
- (b) Basis functions and basis sets: Derived data types are employed here to keep all information together about the (symmetry-adapted) many-electron basis functions as well as about some *finite* set of such basis functions, either in symmetry-adapted form (CSF basis) or as set of determinants (determinant basis). Any of these basis sets can be combined also with the representation of one or several atomic states to form a (so-called) ASF basis. For example, a single CSF is internally defined as:

```
type, public :: cs_function
  integer(kind=ilb) :: totalJ
  character(len=1) :: parity
  integer(kind=ilb), dimension(:), pointer :: occupation
  integer(kind=ilb), dimension(:), pointer :: seniority
  integer(kind=ilb), dimension(:), pointer :: subshellJ
  integer(kind=ilb), dimension(:), pointer :: subshellX
end type cs_function
```

where the total angular momentum and parity is given by `totalJ` and `parity`, respectively, but where all further quantum numbers with regard to the occupation and coupling of the subshells are described by pointer arrays. The dimension of these arrays is specified only during execution due to the given shell structure of the atom or ion. A finite number (`nocsf`) of CSF are then combined into a CSF basis by:

```

type, public :: csf_basis
  integer :: nocsf          ! Number of CSF in the basis.
  integer :: nwshells       ! Number of (relativistic) subshells.
  integer :: nwcore         ! Number of (closed) core subshells.
  integer :: number_of_electrons
  type(nkappa), dimension(:), pointer      :: subshell
  type(csf_function), dimension(:), pointer :: csf
end type csf_basis

```

which includes the number and specification (as well as the order) of the subshells and the number of electrons to which this basis applies. This derived type structure together with the definition of a single ASF in the structure type(*as_function*), in turn, is utilized to define an ASF basis by:

```

type, public :: asf_basis
  integer      :: noasf          ! Number of considered ASF.
  real(kind=dp) :: average_energy ! Averaged energy of this set of ASF.
  type(as_function), dimension(:), pointer :: asf
  type(csf_basis) :: csf_set
end type asf_basis

```

Apart from the radial orbital functions, such an ASF basis contains all information as obtained from a standard GRASP92 run. Moreover, by using the specification

```
type(asf_basis) :: bound
```

and a proper (prior) allocation of storage for all the pointer arrays in these structures, the number of CSF or the orbital quantum numbers (n_1, κ_1) of the first subshell is accessed simply by typing `bound%csf_set%nocsf`, `bound%csf_set%subshell(1)%n` and `bound%csf_set%subshell(1)%kappa`, respectively. Similar type definitions are made to define a determinant basis or the representation of a (hyperfine structure) $|JF\rangle$ -coupled atomic basis (29).

- (c) Derived types specific to some application: are defined, in addition, in most program components. These type definitions help to deal more efficiently with the data that are given to or generated by the module procedures of the component. In the module *rabs_photo*, for example, the following derived type definitions are made:

```

type :: photo_channel
  integer      :: kappa, totalJ
  character(len=1) :: parity
  character(len=2) :: multipole
  character(len=9) :: gauge
  real(kind=dp)  :: phase, amplitude_re
  complex(kind=dp) :: amplitude
end type photo_channel
!
type :: photo_line
  integer      :: asfi, asff, level_i, level_f, totalJ_i, totalJ_f
  integer      :: No_channels
  character(len=1) :: parity_i, parity_f
  real(kind=dp)  :: p_energy, e_energy, cs_coulomb, cs_babushkin
  complex(kind=dp) :: beta_b, beta_c, xi_b, xi_c, eta_b, eta_c, &
                    zeta_b, zeta_c, alignment_b, alignment_c
  real(kind=dp), dimension(:), pointer :: bessell0, bessell1, bessell2, &
                    bessell3, bessell4, bessell5
  type(photo_channel), dimension(:), pointer :: channel
end type photo_line

```

to comprise all information about a particular line. Apart from the level numbers and the total angular momenta and parities of the initial and final states, this derived data structure enables one to specify a number of photoionization channels during the execution of the program, and which are characterized by the overall symmetry of the scattering state “photoion + electron” as well as the gauge and multipolarity due to the coupling of the radiation field. We will not explain here all the details for this derived data structure above but display these examples in order to provide the user with some basic *flavor* of how the RATIP code is organized internally. While the data structure of these derived types is completely specified, storage is allocated to it only according to the (interactively) selected numbers of photo lines, multipoles and gauge forms. Some of the information in these derived type definitions, such as the total angular momenta and parities of the atomic levels, is *redundant* in some cases (in the sense that the same data are usually stored also elsewhere in the program) but it simplifies the internal use of the underlying data if the relevant data are kept within the same derived type. Together with the specification

```
type(photo_line), dimension(:), pointer :: line
```


Table 1

Common file extensions used by the RATIP program. Apart from a short description of the file content, we here list the program components of the distributed version in which they are generated or utilized.

Extension	Content of file	Components
.csl, .inp	Configuration symmetry list file as usually generated by GRASP92; however, the TOOLBOX of the RATIP program also provides various features for merging and manipulating such CSF lists.	All main components & TOOLBOX.
.iso	Isodata file for information about the nuclear parameters and potential.	Almost all main components & TOOLBOX.
.mix	Mixing coefficient file that provides the representation of one or several atomic states; it has the same structure as in GRASP92 but is usually handled within RATIP in a <i>formatted</i> (ASCII) file format.	AUGER, CESD, EINSTEIN, PHOTO, REC, RELCI, REOS, TOOLBOX.
.rwf	Radial wave function file as usually generated by GRASP92; both, a unformatted and formatted version of this file are supported (automatically) by RATIP.	All main components & TOOLBOX.
.sum	Summary file that comprises the main results of the computations; it usually contains a copy of the interactive input, a short summary about the atomic levels involved as well as all (main) results and tabulations of the corresponding component.	Almost all components apart from TOOLBOX.
.trn	Transition amplitude file that provides all transition amplitudes for later use.	AUGER, EINSTEIN, PHOTO, REC, REOS.
.xpn	The Cesd expansion file provides a complete expansion of one or several selected ASF in a determinant basis.	CESD, REOS.

an argument `line(k)` to some module procedure of `rabs_photo` then enables one to access (and to keep) all necessary information about the k -th photo transition that is considered within the current run of the PHOTO component. Similarly complex type definitions occur also in other components of the RATIP program.

It is obvious from this brief discussion that a proper “set” of derived data types facilitates the (data) exchange between the different program components of RATIP. A consistent application of this and similar derived types has been found one of the cornerstone that helped *prepare* RATIP for forthcoming years and applications.

Logical flags: have been found useful to increase the readability and maintenance of the program. Such flags are defined in many modules of the RATIP code to make default or user-defined (boolean) variables available to the procedures of a given module or to all other modules which use it; these flags can often be modified interactively during input. We make usually use of rather long and self-explaining names to increase the readability of the code so that further explanations are not needed. For example, a branch of the code

```
if (logical_flag) then
  ...
end if
```

is executed only if the current value of this flag is `.true.` and is not entered otherwise (i.e. for `logical_flag = .false.`). Examples for such logical flags are `hamiltonian_XL_coulomb`, `hamiltonian_XL_gaunt`, `hamiltonian_XL_breit0`, ... to specify the interactions that are to be included into the computation of the Hamiltonian matrix (in the RELCI component). A careful use of these and further logical flags makes the source code more easily understandable and also helps in testing the program.

Internal termination points within the code: Despite of a careful handling of the input data, there (may) arise situations in which the code is executed with data for which it was not developed originally. This problem occurs indeed rather frequently if a procedure was written for a certain range of parameters but is later utilized by other components for which similar restrictions on the parameters do not apply. Since an detailed analysis of all input data in each individual procedure is quite cumbersome to code and maintain in the long term, we perform internally instead in the RATIP program only some “consistency checks” during the execution of a subprocedure in order to test for the (received) *value* of some input parameter or variable. These tests are usually made by means of an *if-else-end if* structures and are combined with a

```
stop "procedure_name(): program stop A."
```

statement to terminate the program, if a “mismatch” between the expected and obtained (range of) values is found. Such a simple `stop` statement provides the user with the name of the procedure and the exact location of this `stop` (numbered by A, B, ...) but leaves it otherwise to the developer (user) to handle this situation properly. We have found these internal termination points very helpful to recognize possible bugs already quite early during execution, often long before they cause some futile results or subsequent “hard failures” of the code.

3.2.2. Interface and output files

Each component of the RATIP program has to be *invoked* separately to perform one or several selected computations; it requires some specific input and usually provides output to the default list (usually the screen), a summary file (`.sum`) as well as, if appropriate, to further output files of the program component.

Before calling the main components of the RATIP program, it is supposed usually that all required bound-state wave functions were generated by means of GRASP92 [6]. For a given set of atomic levels (ASF), whose representation was generated in a single run of GRASP92, these bound-state wave functions are provided in terms of a configuration symmetry list file (`.csl`), a radial wave function file (`.rwf`) as well as a mixing coefficient file (`.mix`); cf. Table 1. Further information about the nuclear potential is taken from the isotope data file (`.iso`). These four files from GRASP92 contain all information about a desired set of (commonly optimized) atomic states. In many

components of RATIP, however, several sets of such wave function files (but typically with just one `.iso` file) are required to define the initial and final bound states of some particular process and, in a few but not yet distributed components, even some set(s) of intermediate states become necessary. These (files of) bound-state wave functions may refer also to a different number of electrons in the initial and final states as it is characteristic for the considered process.

Since RATIP was developed to support quite different applications, it is not surprising perhaps that different file formats are needed to communicate and handle the data. Here, we shall not explain these file formats in all detail; some further information about these files will be provided in Section 3.4 together with the short description of the individual program components. For a short overview about the *interface* files of the RATIP program, however, Table 1 lists the file (name) extensions, although a common extension does not imply automatically that all the corresponding files will have *exactly* the same format. For example, the transition amplitude (`.trn`) and summary files (`.sum`) usually depend on the atomic process and properties to be considered. Here, the same file extension only shows that they have a similar structure, and most of them are now handled in ASCII format to facilitate the use, file transfer and further developments on the code.

The results and some additional information about the current computations are written to the standard output (screen) but all major results are printed also to a (`.sum`) summary file of the corresponding component. These summary files also contain the date and time of execution (provided that the corresponding calls to the operating system are executed properly), a *copy* of the – interactively given – input as well as a short summary about the atomic levels and orbital functions from the prior GRASP92 calculations that are utilized by the present run of the program. A few typical input and summary files are distributed together with the program in the corresponding `test-component` directory. For a few components, such as PHOTO or RELCI, the summary file also displays a short statistics about the computation concerning, for example, the number and specification of ionizations channels, the number of calculated and/or *re-used* integrals, or about the internal storage management.

It has been appreciated by the users (and collaborators) in many applications of the RATIP code, that the main results of each component are provided in rather elaborate and carefully arranged tabulations. In these tables, the atomic levels and transitions are usually specified in terms of their level numbers as they occurred and were generated before within the GRASP92 program. Although these level numbers are generally not unique from a physics viewpoint, especially if highly-excited or inner-shell hole states are considered, they are very helpful to select individual levels and transitions and to simplify the data communication between different components of the RATIP program. Apart from the level number, however, these tabulations also provide the total angular momenta and parities of the atomic levels under consideration in a well readable form. For the printout of the tables, moreover, the user can often select between various orders and units in which the transition data are displayed by the program.

3.2.3. Interactive control of execution

The execution of all components of the RATIP program is *controlled* by some interactive dialog just at the beginning of the run; this “dialog” requests and collects all information which are necessary to specify the computation uniquely. Of course, the *answers* to this dialog can be compiled also within an ASCII file and given to the program as “standard” input when it is executed. Apart from the choice of units for the energies, cross sections or rates as well as the file names of all additional input and output files, this interactive dialog defines (or overwrites) all relevant logical flags, level number of the atomic states to be involved in the computations, or further restrictions upon the given problem or execution of the program. For the main (and distributed) components of the RATIP program, these “dialogs” will be displayed and explained below in Section 3.3, together with a short summary about their purpose, the implemented amplitudes and properties, special features and a few references to their previous application.

Apart from certain mandatory input data, many components also provide some optional control of the execution by which individual transitions from a larger transition array or specific interactions can be selected, or which help overwrite the standard computational mode and/or logical flags of the program. All levels and transitions between atomic bound states are handled during the input as well as internally by means of their level numbers as obtained before by the GRASP92 code. Again, this optional input control will be displayed in Section 3.3 for each program component to be distributed; this optional part of the input dialog is executed only if a question like ‘Modify the default set-up and printout of the program?’ occurs and is answered with `y(es)`. Alternatively, the default execution of the components can be changed by modifying the boolean initialization of the logical flags in the corresponding modules before they are compiled; these variables are then re-initialized always at the beginning of the execution.

3.2.4. Test suites

For several components of the RATIP program, and especially for those discussed in Section 4, we also provide some test suites in the corresponding sub-directories `test-component`. For example, the directory `test-photo` in the `ratip` root directory comprises all input and output files that were generated prior to the computations by means of the GRASP92 program as well as the PHOTO component for the example discussed in Section 4.1. These test directories are provided for some first “getting started” with the program and for some quick check of the current installation of the program.

3.3. Main components of the code

In this section, I shall briefly explain and summarize all those (main) components that are distributed in the present version of the RATIP program. These components enable the user to study photo excitation and decay properties as well as various photoionization, Auger and recombination processes. In addition, they support an efficient set-up and diagonalization of the Dirac–Coulomb–Breit Hamiltonian matrix (similar to the RC192 component from GRASP92) as well as the (complete) expansion of atomic or configurations states functions into a determinant basis. However, no attempt will be made here to compile or describe further theoretical and computational details behind the implementation other than provided already in Sections 2 and 3.

3.3.1. AUGER

The AUGER component supports the computation of nonradiative (electron emission) rates, relative Auger intensities and angular distribution parameters. It requires a representation of the atomic bound-state functions from GRASP92 for the initial and final-ionic states

(levels) of the autoionization process, while the continuum orbitals of the outgoing electron(s) and the final scattering states for the “photon + electron” are constructed internally by this component. The continuum orbitals are usually generated within an *optimized level* potential of the final state and are kept orthogonal to the bound-state orbitals of the final ion [40]. Standard Racah algebra techniques [8] are applied for the evaluation of the many-electron amplitudes [36], independent of how the radial orbitals of the initial and final bound states were generated before by means of the GRASP92 program. Apart from the Coulomb repulsion among the electrons, the AUGER component allows to incorporate also the (frequency-independent) Breit interaction into the Auger amplitudes.

- (i) Input files: `element.iso`, `initial-csl.inp`, `initial-scf.rwf`, `initial-relci.mix`, `final-csl.inp`, `final-scf.rwf`, `final-relci.mix`.
- (ii) Output files: `auger.sum`, `auger.trn`.

Implemented transition amplitudes: An autoionization of an (inner-shell) excited bound state occurs only if this initial level is energetically embedded into the continuum of the next higher charge state. The electron emission from such an embedded resonance then arise due to its coupling to one or several scattering states that are degenerate with the initial state and that involve a *free* electron. The theoretical description of this autoionization process requires in addition to impose proper boundary conditions upon the (one-electron continuum orbitals of the) outgoing electron and, in a more exact treatment, upon the scattering states themselves. Following Åberg and Howat [41], the Auger transition amplitude for the autoionization of the excited bound state $|\psi_i(P_i J_i M_i)\rangle$ with total angular momentum J_i , M_i and parity P_i into the final scattering state $|\psi_t\rangle = |\psi(P_t J_t M_t)\rangle$ is obtained as

$$V_{J_f J_i J_t} \equiv \langle (\psi_f, \epsilon \kappa) P_t J_t \| H - E \| \psi_i(P_i J_i) \rangle \delta_{J_i, J_t} \delta_{M_i, M_t} \delta_{P_i, P_t}, \quad (35)$$

and this amplitude is independent of the projection M_i of the initial state due to the rotational invariance of free atoms and the scalar character of the Hamiltonian. The possible final scattering states $|\psi_t\rangle$ arise from the coupling of the state $|\psi_f\rangle = |\psi(P_f J_f M_f)\rangle$ of the final ion with well-defined angular momentum J_f , M_f and parity P_f with the partial wave of the outgoing electron with energy ϵ and (one-electron) angular momentum

$$\kappa = \pm(j + 1/2) \quad \text{for } l = j \pm 1/2.$$

In AUGER, as in most prior computations of autoionization processes in the literature, a common set of *orthonormal* orbitals is supposed for the representation of the initial $|\psi_i\rangle$ and final (ionic) bound states $|\psi_f\rangle$ in the evaluation of the matrix elements (35). In this case, the transition operator $(H - E) \approx V$ simplifies to the electron–electron interaction operator and the Auger amplitude becomes

$$V_{J_f J_i J_t} \approx \langle (\psi_f, \epsilon \kappa) P_t J_t \| V \| \psi_i(P_i J_i) \rangle \delta_{J_i, J_t} \delta_{M_i, M_t} \delta_{P_i, P_t} \quad (36)$$

where, within the framework of the Dirac–Coulomb–Breit Hamiltonian, the interelectronic interaction is the sum of the Coulomb repulsion and Breit interaction, $V = V^{\text{Coulomb}} + V^{\text{Breit}}$; cf. Section 2.1 above. While the autoionization of light and medium elements are typically well described by including the Coulomb repulsion into the Auger amplitude (36), the Breit interaction has been found important for electron emission and capture processes of highly-charged ions [25,42].

The AUGER component calculates the (reduced) amplitudes in expression (36) for selected pairs of initial and final states and combines them into Auger rates, angular distribution and spin-polarization parameters. Apart from – the additional incorporation of – the Breit interaction, the user can include (or not) the exchange interaction between the free-electron and the bound-state density and has various possibilities to affect the transition energies as well as the printout of the program.

Dialog for controlling the execution: To control the AUGER component, the following questions are raised and the corresponding input is requested from the user; these requests usually appear together with some comments of how the data need to be entered. Some of the questions lead to subsequent requests, if answered with `y(es)`; this is indicated by `[y]` below, while `[y/n]` means that the corresponding question enables the user to overwrite internally some (predefined) boolean variable. For the sake of brevity, I shall not display here the “answers” to all the input requests, although they need to be given in order to proceed with the dialog. While some of the input data can be checked (and, if not appropriate, leads to a *re-do* request by the program), input data errors can cause the program to abort.

```
Enter a file name for the auger.sum file:
Enter the name of the isotope data file:
Which units are to be used to enter and to print the energies of the continuum orbitals ?
  A      : Angstrom;
  eV     : electron volts;
  Hartree : Hartree atomic units;
  Hz     : Hertz;
  Kayser  : [cm**(-1)];
Enter the maximal energy of the Auger transitions (in eV) to built-up the radial grid:
Modify default set-up and printout of the program ? [y]
Select individual transitions ? [y]
Enter one pair |i> - |f> of level numbers, e.g. 2 - 3; 2 - 0; 0 - 17 ...
(0 is here equivalent to all); <cr> if done.
Include exchange interactions into the generation of the continuum waves ? [y/n]
Include Breit interactions into the Auger matrix ? [y/n]
Calculate angular distribution parameters ? [y/n]
Calculate spin polarization parameters ? [y/n]
Sort transitions in ascending order of energy ? [y/n]
```

```

Read in and apply experimental energies for the calculation of Auger rates ? [y/n]
Auger rates are printed in SI units; use Hartree atomic units instead ? [y/n]
Print all selected transitions and their energies before the computation starts ? [y/n]
Print the CSF scheme each time a new one has been built ? [y/n]
Print the results for each individual transition immediately after its computation ? [y/n]
Print the final results to a (.trn) transition amplitude file ? [y]
Enter a file name for the auger.trn file:
Enter an (overall) shift of the Auger energies for all transitions (in eV):
    Use 0. or <cr> if no shift need to be applied.
Enter a minimal energy (>=0.) of the free electron (in eV):
    All other transitions are neglected from the computations;
    use 0. or <cr> if all possible transitions are to be taken into account.
Enter a maximal (-)kappa symmetry up to which continuum spinors are taken into account ?
    2 (up to p-waves), 4(f), 6(i), 8(k), ...; 0 or <cr> to include all possible waves.
The physical speed of light in atomic units is 137.035989500000;
    revise this value ? [y/n]
The default radial grid parameters for this case are:
:
    revise these values ? [y/n]
    ..... here now follows mandatory input again
Enter the name of the initial-state GRASP92 configuration symmetry list file:
Enter the name of the final-state GRASP92 configuration symmetry list file:
Enter the name of the initial-state GRASP92 mixing coefficient file:
Enter the name of the final-state GRASP92 mixing coefficient file:
Enter the name of the initial-state Radial WaveFunction File:
Enter the name of the final-state Radial WaveFunction File:

```

Results and summary: Apart from a short compilation of the atomic levels involved in the computations (cf. Section 3.2.2), the `auger.sum` summary file also provides a table of all selected transitions and transition amplitudes as derived from the total energies of the initial and final states as well the allowed orbital symmetries of the outgoing electron. This tabulation is made before the computation of the many-electron Auger amplitudes actually starts. Moreover, the summary file usually lists all amplitudes (36) and rates of the allowed partial waves, together with the total rates and angular parameters (if they were requested to be calculated in the present run). Finally, all transitions rates are tabulated in a neat table together with a total (sum) rate over all transitions which were calculated so far. Similar tabulations are produced also for the angular and spin-polarization parameters if they were calculated in the present run of the program. In the `auger.trn` transition amplitude file, the Auger amplitudes are listed (in a separate line) for each autoionization channel as defined by the final ionic level and the partial-wave symmetry κ of the emitted electron.

Special features: Since the number of possible Auger transitions increase rapidly if several inner-shell holes and/or valence electrons occur, the AUGER component provides additional features for reducing the complexity, apart from the selection of just some particular transitions. Beside of an energy threshold (i.e. a minimal energy) for the free electron, the user can specify a maximal angular momentum (negative κ number) for the partial waves to be considered in the construction of the scattering states, independent of the possible coupling of the free-electron orbitals to the ionic bound states. This selection is in line with the common experience that partial waves with high angular momenta does often not contribute at all (i.e. have zero amplitudes in most computational models) or are negligible. In addition, the AUGER components supports the use of an overall energy shift of all Auger transition energies or the use of *experimental* transition energies which are read in by the dialog above, transition by transition, *after* all the other input has been processed successfully.

Selected previous applications: The AUGER component has been widely used for studying electron emission processes after inner-shell excitation [43] and ionization of atoms [44] as well as for analyzing the coherence transfer through Auger cascades [23]. If the fine-structure splitting is small or comparable to the natural line widths of the levels, the angular emission of the (second-step) Auger electrons can be described no longer by some incoherent summation over the individual decay paths from the first and second step of the autoionization [29]. Then, the coherence transfer through all major and nearly degenerate decay channels need to be taken into account by combining, for instance, the many-electron Auger amplitudes from the AUGER component with the density matrix formalism. However, automatic computation of Auger cascades and the associated coherence transfer is not at present supported; such calculations have previously been carried out by means of some separately developed computer-algebra tools [45–47].

3.3.2. CESD

The CESD component performs a *complete* expansion of either the atomic or configuration state functions from GRASP92 into a determinant basis. Such expansions are useful, for example, for *relaxed-orbital* calculations, i.e. if the rearrangement of the electron density is to be included into the evaluation of (general) transition amplitudes by applying Löwdin's method [48]. In many-electron atoms and ions, the relaxation effects to the electron density are naturally incorporated into the wave functions if the initial and final states are optimized separately. However, this independent optimization leads to different sets of one-electron orbital function that are not quite orthogonal to each other. Within RATIP, the CESD component is required by REOS (cf. Section 3.3.7) for the computation of relaxed-orbital oscillator strength and transition probabilities as well as at several places elsewhere. An alternative and more elegant approach (but also elaborate implementation) makes use of a *biorthogonal* transformation of the radial orbitals [49] and has been realized in GRASP2K [10].

- (i) Input files: `cs1.inp`, `rci.mix` or `relci.mix`.
- (ii) Output files: `cesd.sum`, `cesd.xpn`.

Implemented expansion: While Slater determinants are perhaps the most simplest anti-symmetric N -electron functions, they do generally *not* reflect the rotational symmetry of free atoms or ions. They are eigenfunctions to the parity operator P and the projection of the total angular momentum, J_z , but not to the operator J^2 . The Slater determinants therefore contribute only to *reducible* representations with regard to the (SO_3) rotation group. For a given electron configuration, all the associated determinants hereby just differ by the magnetic quantum numbers $\{m\}$ of the occupied orbitals.

Mathematically, the CESD component performs a rotation of the symmetry-adapted (N -electron) basis $\{|\gamma P J M\rangle\}$ in ansatz (7) into a determinant basis

$$|\gamma P J M\rangle = \sum_{\{m\}} |U_{\{m\}}\rangle \langle \{m\} | \gamma P J M\rangle, \quad (37)$$

where $|U_{\{m\}}\rangle$ denotes a set of determinants from the *same* (electron) configuration as defined by the occupation numbers of $|\gamma P J M\rangle$. In this notation, $\{m\}$ abbreviates a set of magnetic quantum numbers of the corresponding Dirac orbitals with $\sum_i m_i = M$, and which here replaces the coupling information about the subshell states in the symmetry-adapted functions. The matrix elements $\langle \{m\} | \gamma P J M\rangle$ are *generalized* Clebsch–Gordon coefficients which include the coefficients of fractional parentage and which are known to form an orthogonal transformation matrix [8]. As seen from Eq. (37), the CESD component only concerns the representation of the atomic (or configuration) states but does not change these functions at all.

Dialog for controlling the execution:

```
Enter a file name for the cesd.sum file:
Enter the name of the configuration symmetry list file:
Expand the atomic state functions (ASF) ? [y/n]
Print non-standard output on the cesd expansion ? [y/n]
Select non-standard total M values for the expansion into Slater determinants ? [y/n]
    The default is to use M = +J for each individual determinant.
Enter the name of the GRASP92 mixing file:
Enter a name for the CESD eXPaNsion .xpn output File that is to be created:
```

Results and summary: The expansion of either some selected ASF or the given set of CSF as a whole, including the full specification of all determinants as well as the representation of the states in this basis, is written to the `cesd.xpn` expansion file. This file comprises the total numbers of determinants and orbitals, a (ordered) list of Dirac orbitals in terms of their quantum numbers ($n\kappa m$) as well as the occupation of these orbitals for each determinant of the generated basis. For each symmetry function, moreover, the total angular momentum and parity is printed out together with its representation (eigenvectors) in the pre-defined determinant basis. The `cesd.sum` summary file is short and only displays the list of orbitals and some statistics upon the expansion. The `cesd.xpn` expansion file may be needed as input to other RATIP programs.

Special features: CESD is typically applied to expand one or several atomic levels into a determinant basis but can be utilized also for the expansion of a (whole) CSF basis. This latter option, however, may lead to sizeable output files. In the expansion of atomic levels, the default is to set the total magnetic quantum number $M = +J$ in order to keep the number of determinants small. This default can be overwritten at input time. Moreover, in order to increase the efficiency of the program for large wave function expansions, the occupation numbers (0 or 1) of the Dirac orbitals in the specification of the determinants is internally treated *bitwise* by using the (intrinsic) bit manipulation procedures of Fortran 90/95. This reduces the storage requirements, both internally as well as in the `cesd.xpn` expansion file, and also accelerates the expansion process itself [38]. The CESD component supports the expansion of CSF with up to 5 open subshells only, although this limitation could be removed with moderate effort if needed by the user.

Selected previous applications: The CESD component has been mainly utilized together with REOS in calculations of (relaxed-orbital) transition probabilities and lifetimes; see Section 3.3.7. It has been recently applied, moreover, in the population analysis of excited three-electron states in carbon ions after single-electron capture in 0.5–1.1 MeV/amu $C^{4+}(1s2s^3S)$ –He collisions [50]. In these case studies, single-electron capture amplitudes from the two-center basis generator method were combined with MCDW wave functions in a determinant representation in order to explore different decay mechanisms for the metastable $C^{3+}(1s2s2p^4P)$ level. Further applications of the CESD component refer to two-electron-one-photon transitions, a second-order photon emission process that becomes possible due to the rearrangement of the electron density [51].

3.3.3. EINSTEIN

The EINSTEIN component helps calculate transition probabilities, Einstein A and B coefficients, oscillator strength as well as radiative lifetimes. Like for the OSCL component of GRASP92, the (standard) Racah algebra techniques are employed to evaluate the transition amplitudes, while the radial orbitals are handled independent for the initial and final states. Though the use of two orbital sets, which are not quite orthogonal to each other, causes some *inconsistency* in dealing with Fermi's Golden rule and the use of the Racah techniques, this scheme allows a quick and simple evaluation of the transition amplitudes and to incorporate at least parts of the rearrangement effects into the transition amplitudes and probabilities, if the initial and final states of the computations were generated separately. The EINSTEIN component also provides a transition amplitude file (`einstein.trn`) and a neat tabulation of the Einstein coefficients and oscillator strengths analogue to the REOS component; cf. Section 3.3.7. The EINSTEIN component can be applied both, for the absorption and emission of photons, by just choosing the initial and final stomic bound states properly.

- (i) Input files: `element.iso`, `initial-csl.inp`, `initial-scf.rwf`, `initial-relci.mix`, `final-csl.inp`, `final-scf.rwf`, `final-relci.mix`.
- (ii) Output files: `einstein.sum`, `einstein.trn`.

Implemented transition amplitudes: All properties concerning the absorption and emission of photons in transitions between atomic bound states can be traced back to the transition amplitudes (25), if calculated for selected pairs of initial and final states of an atom or ion. While most observations are related to allowed electric-dipole (E1) transitions between atomic states with total angular momenta $\Delta J \equiv |J_i - J_f| \leq 1$, higher multipoles (M1, E2, ...) in the electron-photon interaction can occur and affect the level population of atoms and ions. In EINSTEIN like in other components of the RATIP and GRASP92 programs, the transition amplitudes (25) of all selected (or possible) multipole transitions are calculated as linear combination

$$\langle \alpha_f J_f \| H_\gamma(\pi L) \| \alpha_i J_i \rangle = \sum_{r,s} c_{fr} c_{is} \left\langle \gamma_f P_f J_f \left\| \sum_p \alpha_p \cdot \mathbf{A}_{L,p}^\pi(\mathbf{k}) \right\| \gamma_i P_i J_i \right\rangle \quad (38)$$

of the reduced matrix elements between the CSF as they appear in the expansion of the atomic states, and where the notation of the multipoles were described in Section 2.3. The transition amplitudes (38) are then properly combined to obtain line strength, Einstein coefficients or lifetimes. These amplitudes give rise also to the alignment of atomic states or the angular distribution and polarization of emitted radiation, if properly combined with the amplitudes from prior (de-)excitation mechanisms of the atoms under investigation. Further details about the photo-induced excitation and ionization of atoms and ions can be found, for example, in Ref. [29].

Dialog for controlling the execution:

```
Enter a file name for the einstein.sum file:
Enter the name of the isotope data file:
Enter the transition multipoles, e.g. E1 M2 ... :
Which units are to be used to enter and to print the energies of the continuum orbitals ?
    A      : Angstrom;
    eV     : electron volts;
    Hartree : Hartree atomic units;
    Hz     : Hertz;
    Kayser  : [cm**(-1)];
Modify default set-up and printout of the program ? [y]
Select individual transitions ? [y]
Enter one pair |i> - |f> of level numbers, e.g. 2 - 3; 2 - 0; 0 - 17 ...
(0 is here equivalent to all); <cr> if done.
Sort transitions in ascending order of energy ? [y/n]
Read in and apply experimental energies for the calculation of transition probabilities ?
Einstein A and B coefficients are printed in SI units; use atomic units instead ? [y/n]
Print all selected transitions and their energies before the computation starts ? [y/n]
Print the results for each individual line immediately after its computation ? [y/n]
Write out the transition energies and amplitudes to an .trn file ? [y]
Enter a file name for the einstein.trn file:
The physical speed of light in atomic units is 137.035989500; revise this value ? [y/n]
The default radial grid parameters for this case are:
:
    revise these values ? [y/n]
    ..... here now follows mandatory input again
Enter the name of the initial-state GRASP92 configuration symmetry list file:
Enter the name of the final-state GRASP92 configuration symmetry list file:
Enter the name of the initial-state GRASP92 mixing coefficient file:
Enter the name of the final-state GRASP92 mixing coefficient file:
Enter the name of the initial-state Radial WaveFunction File:
Enter the name of the final-state Radial WaveFunction File:
```

Results and summary: Following the input data and the short compilation of atomic levels involved in the computations, the einstein.sum summary file provides a table of the selected transitions and multipoles (printed before the actual computations start) as well as a summary of all transition probabilities and lifetimes. This summary lists the transition data in terms of the symmetry of the initial and final level, the transition energy, Einstein coefficients, oscillator strength and the associated decay widths. In general, the atomic levels (and corresponding ASF) refer to different GRASP92 calculations for the initial and final states, although the *same* set of wave functions (.csl, .rwf and .mix files) can be used in both cases. From all calculated multipole transitions with positive (transition) energy, moreover, the lifetimes and width of the *upper* levels are compiled in the einstein.sum summary file. In the einstein.trn transition amplitude file, the transition amplitudes are listed, again in a separate line for each transition between two fine-structure levels, for further applications by other RATIP components, its toolbox or elsewhere.

Special features: All electric multipole transitions are always calculated in the (so-called) Babushkin *and* Coulomb gauges which, within the nonrelativistic limit, refer to the well-known length and velocity gauge form of the transition amplitudes, respectively [52]. Although an agreement of the two gauges only provides a *necessary* criterion for obtaining accurate transition data, possible deviations between the two gauge are often taken as “indicator” for the accuracy of the computations, especially if a large number of transitions were calculated within the same computational model. As default, possible multipole transitions between initial and final atomic levels are selected “in emission”, i.e. for photon energies $\hbar\omega = E_i - E_f > 0$, although absorption processes between atomic states with $E_i - E_f < 0$ can also be investigated if individual transitions are selected explicitly by the input data.

When compared to the REOS component (Section 3.3.7), EINSTEIN is typically faster in evaluating the transition amplitudes and, thus, is suitable for mass production of transition probability data or if a large number of radiative transitions are involved by some property.

Within the RATIP program, for example, the – central routine of the – EINSTEIN module is invoked within the DIEREC component for the computation of dielectronic recombination strengths or for studying the properties of hyperfine-quenched transitions.

Selected previous applications: The EINSTEIN component is frequently applied if a fast overview over some (large) transition array or the lifetime of rather highly excited states is needed [53]. In addition, this component has been utilized for studying the low-lying resonances in the dielectronic recombination of Si IV forming Si III ions [24] as well as the two-photon decay of highly-charged ions [54].

3.3.4. PHOTO

The PHOTO component computes and combines atomic photoionization amplitudes for generating level-dependent ionization cross sections, alignment parameters of the photoions and angular distribution parameters for the emitted electrons. Like for most other components, it requires a representation of the atomic bound-state functions (for example, from GRASP92) for the initial and final-ionic states (levels) of the photoionization process, while the continuum orbitals of the emitted electron(s) and the final scattering states for the “photoion + electron” are constructed internally by the PHOTO component. However, standard Racah algebra techniques [8] are applied for the evaluation of the many-electron amplitudes (39), independent of how the radial orbitals of the initial and final bound states were generated before by means of the GRASP92 program. In practice, PHOTO provides a quite simple access to the (direct) photoionization cross sections of atoms and ions, independent of their shell structure, but does not include the *resonances* that (may) occur in many photoionization properties.

- (i) Input files: element.iso, initial-csl.inp, initial-scf.rwf, initial-relci.mix, final-csl.inp, final-scf.rwf, final-relci.mix.
- (ii) Output files: photo.sum, photo.trn.

Implemented transition amplitudes: The photo-induced ionization of an initial bound state $|\psi_i(P_i J_i)\rangle$ with N electrons into some final state $|\psi_f(P_f J_f)\rangle$ of the $(N - 1)$ electron photo ion requires of course photon energies $\hbar\omega > E_i(N) - E_f(N - 1)$. The photoionization amplitudes for such an ionization process into some final scattering state $|\psi_t\rangle = |\psi(P_t J_t M_t)\rangle$ are described by

$$M_{J_i \rightarrow J_f J_t J_e}^{(\pi L)} \equiv \langle \psi_f, \epsilon \kappa | P_t J_t \| H_\gamma(\pi L) \| \psi_i(P_i J_i) \rangle, \quad (39)$$

where $H_\gamma(\pi L)$ refer to the electron–photon coupling operator in Eqs. (25) and (26). Again, boundary conditions need to be imposed upon the partial waves of the outgoing electron with energy ϵ and angular momentum κ , like for autoionization processes, but by including the proper phase (see, e.g. Ref. [41] for the discussion of these boundary conditions and possible improvements of the theory). In the PHOTO component, a common set of *orthonormal* orbitals is supposed for the evaluation of the (spin-angular part of the) many-electron amplitudes (39), independent of how the radial orbitals were generated. While the photoionization of atoms is clearly dominated by electric-dipole (E1) coupling of the electrons to the radiation field, higher multipole components of the electron–photon interaction become important at (very) high photon or in various angle-differential properties, if the E1 contribution is *geometrically* suppressed at certain angles.

The PHOTO component calculates the (reduced) amplitudes (39) for selected multipoles (E1, M1, ...), photon energies and pairs of initial and final states. It combines these amplitudes to partial and total photoionization cross sections, angular distribution and spin-polarization parameters. Like for all components that include a *free* electron in the transition amplitudes, the user can select (or not) to include the exchange interaction between the free-electron and the bound-state density as well as various restrictions upon the symmetry and energy of the outgoing electrons.

Dialog for controlling the execution:

```
Enter a file name for the photo.sum file:
Enter the name of the isotope data file:
Enter the transition multipoles, e.g. E1 M2 ... :
Which units are to be used to enter and to print the energies of the continuum orbitals ?
    A      : Angstrom;
    eV     : electron volts;
    Hartree : Hartree atomic units;
    Hz     : Hertz;
    Kayser  : [cm**(-1)];
Enter the maximal energy of the photo lines (in eV) to built-up the radial grid:
The photon energies can be entered either individually or by an appropriate interval and
step size. The default is a list of individual energies; revise this ? [n]
Enter an interval of photon energies and a corresponding stepsize:
    E_lower E_upper delta-E:
Modify default set-up and printout of the program ? [y]
Select individual transitions ? [y]
Enter one pair |i> - |f> of level numbers, e.g. 2 - 3; 2 - 0; 0 - 17 ...
(0 is here equivalent to all); <cr> if done.
Include exchange interactions into the generation of the continuum waves ? [y/n]
Calculate angular distribution parameters ? [y/n]
Calculate spin polarization parameters ? [y/n]
Sort all photoionization lines in ascending order of energy ? [y/n]
PI cross sections are printed in SI units; use Hartree atomic units instead ? [y/n]
Print all selected photoionization lines before the computation starts ? [y/n]
Print the CSF scheme each time a new one has been built ? [y/n]
```



```

Print the results for each individual line immediatly after its computation ? [y/n]
Print the final results to a (.trn) transition amplitude file ? [y]
Enter a file name for the photo.trn file:
Enter a shift for the transition energies which applies to all transitions (in eV):
Use 0. or <cr> if no shift need to be applied.
Enter a maximal (-)kappa symmetry up to which continuum spinors are taken into account ?
  2 (up to p-waves), 4(f), 6(i), 8(k), ...; 0 or <cr> to include all possible waves.
The physical speed of light in atomic units is 137.035989500; revise this value ? [y/n]
The default radial grid parameters for this case are:
:
  revise these values ? [y/n]
  ..... here now follows mandatory input again
Enter the name of the initial-state GRASP92 configuration symmetry list file:
Enter the name of the final-state GRASP92 configuration symmetry list file:
Enter the name of the initial-state GRASP92 mixing coefficient file:
Enter the name of the final-state GRASP92 mixing coefficient file:
Enter the name of the initial-state Radial WaveFunction File:
Enter the name of the final-state Radial WaveFunction File:

```

Results and summary: Following a short compilation of all bound levels involved in the computations, the `photo.sum` summary file tabulates the selected photolines as given by the (pairs of) initial and final atomic levels as well as the photon energy. The calculation of photoionization properties for the same initial and final states but for different photon energies has to be performed separately due to the different energies of the outgoing electrons, although the close relation of the corresponding amplitudes is taken into account in the implementation to accelerate the computations. The tabulation of the selected photolines also specifies (automatically) all possible *photoionization channels* as specified by the energy and symmetry of the partial wave, the gauge and multipole operators in the electron–photon coupling as well as the angular momentum and parity of the overall scattering states; this table is printed before the computation of the many-electron photoionization amplitudes actually start. Moreover, the `photo.sum` summary file also lists (if specified during input) all the selected photoionization amplitudes, phases and properties for each photoline immediately after its computation. Finally, all photoionization cross sections and other selected properties (angular distribution, alignment, ...) are printed in various tables for either a quick overview or to follow the energy dependence of the individual properties. In the `photo.trn` transition amplitude file, the photoionization amplitudes and phases are listed (in a separate line) for each photoionization channel, including all information about the partial waves, gauge and multipoles as well as the symmetry of the final scattering state to uniquely specify the amplitudes for further processing.

Special features: Beside the selection of the transition multipoles and individual transitions, the user can specify the photon energies either separately, energy by energy, or in terms of some energy interval and a given step size. This allows a quite flexible access to the photon energies and energy dependence of the photoionization cross sections and other properties. Furthermore, the maximal angular momentum (negative κ number) for the partial waves of the outgoing photo electron (photoionization channel) and an overall shift of all transition energies can be specified during the input time in order to keep large computations feasible.

Selected previous applications: Apart from cross section and angular distribution calculations, the PHOTO component has been applied for studying the interference of direct and resonant photoionization channels in atomic argon [55] as well as the linear magnetic and alignment dichroism in Auger-photoelectron coincidence spectroscopy of atomic tin [56,57]. Moreover, the PHOTO component helped analyze the sequential two-photon double ionization of various noble gases in intense free-electron laser (FEL) radiation [58,59]. In the RESONANCE component (not published in the present version), in addition, the many-electron amplitudes from the AUGER and PHOTO are currently combined in order to describe the photoionization of atoms and ions across some resonant region.

3.3.5. REC

The radiative electron capture (REC) is the *time-reversed* process to atomic photoionization and has been explored in good detail for multiply and highly-charged ions [60]. In this process, a free or quasi-free electron is captured by the ion under the emission of a photon that carries away the exceeding energy. The REC component supports the computation of total REC cross sections and angular distribution coefficients of singly and multiply charged ions. It requires again a representation of the atomic bound-state functions for the initial-ionic and final states (levels) of the electron capture process, while the continuum orbitals of the initially free electron(s) are constructed internally by the REC component. Standard Racah algebra techniques are applied for the evaluation of the many-electron amplitudes, independent of how the radial orbitals of the initial and final bound states were generated before. Although the REC process is closely related to the photoionization of atoms and ions (with basically the same many-electron amplitudes, cf. Eqs. (39) and (40)), the REC component has been developed as a separate program to compile and access the relevant properties of this capture process.

- (i) Input files: `element.iso`, `initial-csl.inp`, `initial-scf.rwf`, `initial-relci.mix`, `final-csl.inp`, `final-scf.rwf`, `final-relci.mix`.
- (ii) Output files: `rec.sum`, `rec.trn`.

Implemented transition amplitudes: The capture of a (quasi-)free electron by an initial ion in the state $|\psi_i(P_i J_i)\rangle$ (with $N - 1$ electrons) gives rise to the N -electron final bound state $|\psi_f(P_f J_f)\rangle$ and can be described in terms of the REC amplitudes

$$M_{J_i J_f \rightarrow J_f}^{(\pi L)} \equiv \langle \psi_f(P_f J_f) \| H_\gamma(\pi L) \| \psi_i(P_i J_i) \rangle, \quad (40)$$

where $|\psi_i(P_i J_i)\rangle = |(\psi_i, \epsilon \kappa) P_i J_i\rangle$ describes an initial scattering state with well-defined angular momentum and parity ($P_i J_i$) and where $H_\gamma(\pi L)$ refers again to the electron–photon coupling operator in Eq. (26). Here, the phase (boundary condition) must describe the capture

of an electron in line with the interchange of the initial and final states with regard to the photoionization amplitudes (39). In the REC component, again, a common set of *orthonormal* orbitals is supposed for the evaluation of the (spin-angular part of the) many-electron amplitudes (39), independent of how the bound radial orbitals were generated by means of the GRASP92 program. For multiply and highly-charged ions, moreover, the angular and polarization properties of the emitted photons are often affected by rather high multipoles up to, say, $L \approx 10$ [28].

The REC component calculates the (reduced) amplitudes (40) up to some given maximum multipole field (L_{\max}), for given kinetic energies of the incident electrons as well as pairs of initial and final states. It combines these amplitudes to partial and total capture cross sections and angular distribution. Moreover, this component allows to calculate the alignment of the final state of the ion *after* the capture of the electron has taken place.

Dialog for controlling the execution:

```
Enter a file name for the rec.sum file:
Enter the name of the isotope data file:
Which units are to be used to enter and to print the energies of the continuum orbitals ?
  A      : Angstrom;
  eV     : electron volts;
  Hartree : Hartree atomic units;
  Hz     : Hertz;
  Kayser  : [cm**(-1)];
Enter the maximal energy of the free electrons (in eV) to built-up the radial grid:
The electron energies can be entered either individually or by an appropriate interval
of kinetic energies and step size. At storage rings, the electron energies are
obtained from the projectile energies by:
  E^(electron) [keV] = E^(projectile) [MeV/u] / 1.8228885.
The default is a list of individual electron energies; revise this ? [n]
Enter another (positive) electron energy in eV or a negative number if done.
Enter the maximal (positive) kappa value to determine the partial waves
in the expansion of the electron wave, i.e. kappa = -1, 1, -2, ..., |max_kappa|:
Modify default set-up and printout of the program ? [y]
Select individual transitions ? [y]
Enter one pair |i> - |f> of level numbers, e.g. 2 - 3; 2 - 0; 0 - 17 ...
(0 is here equivalent to all); <cr> if done.
Include exchange interactions into the generation of the continuum waves ? [y/n]
Calculate angular distribution coefficients ? [y/n]
Calculate statistical tensors for the magnetic population of the final states ? [y/n]
Sort all REC lines in ascending order of energy ? [y/n]
REC capture cross sections are printed in SI units; use Hartree atomic units instead ? [y/n]
Print all selected REC lines before the computation starts ? [y/n]
Print the CSF scheme each time a new one has been built ? [y/n]
Print the results for each individual line immediately after its computation ? [y/n]
Print the final results to a (.trn) transition amplitude file ? [y]
Enter a file name for the rec.trn file:
Enter a shift for the transition energies which applies to all transitions (\xch{in eV}{ineV}):
Use 0. or <cr> if no shift need to be applied.
The physical speed of light in atomic units is 137.035989500; revise this value ? [y/n]
The default radial grid parameters for this case are:
:
  revise these values ? [y/n]
..... here now follows mandatory input again
Enter the name of the initial-state GRASP92 configuration symmetry list file:
Enter the name of the final-state GRASP92 configuration symmetry list file:
Enter the name of the initial-state GRASP92 mixing coefficient file:
Enter the name of the final-state GRASP92 mixing coefficient file:
Enter the name of the initial-state Radial WaveFunction File:
Enter the name of the final-state Radial WaveFunction File:
```

Results and summary: Following the input data and the compilation of the atomic bound states from the GRASP92 calculations, the `rec.sum` summary file first displays (if requested) an overview about all selected “capture lines” as determined by the (pairs of) initial and final atomic levels as well as the kinetic energy of the free electron or the photon energy of the emitted radiation, respectively. Each of these lines comprises usually a rather large number of *capture* amplitudes (40) which are specified in addition by the energy and symmetry of the partial wave, the gauge and multipolarity of the electron-photon interaction operator as well as the parity and angular momentum of the initial scattering states for the “ion + free electron”. This list of capture lines and amplitudes is selected and printed out before the actual calculations start. During the computations, then, the amplitudes, phases and all other selected properties are printed – if requested during the input – for each capture line immediately after these data have been calculated by the program. The capture cross sections and angular parameters (up to β_4 in the expansion of the angular distribution; cf. formulas (32)–(33) in Ref. [28]) are printed again at the end of the calculation in order to facilitate the discussion and exchange of these data. In the `rec.trn` transition amplitude file, moreover, the capture amplitudes and phases are listed for each capture channel in a separate line, including all information about the partial waves, gauge and multipoles as well as the symmetry of the final scattering state to uniquely specify the amplitudes for further data processing.

Special features: As usual for the RATIP code, individual capture lines can be selected by pairs of levels which refer to the initial and final bound-state computations; otherwise, all lines are taken which give formally rise to a positive energy of the emitted photons. In contrast to the EINSTEIN and PHOTO components, however, the multipole contributions in the electron–photon interaction cannot be specified separately but are determined due to the maximum partial wave (κ_{\max}) that is to be included in the expansion of the free-electron wave. All multipoles L and total angular momenta J_t are then taken into account which fulfill the two triangular conditions $\delta(J_f, j_{\max}, J_t)$ and $\delta(J_i, L, J_t)$ in the amplitude (40), where the symbol $\delta(a, b, c) \equiv 1$, if $a + b + c$ is integer and a, b, c may form a triangle, and is zero otherwise. Similar as in the PHOTO component, the kinetic energies of the incident electrons can be specified by the user either separately, energy by energy, or in terms of some energy interval and step size. Apart from the angular distribution coefficients β_ν , moreover, the statistical tensors of the final states can be calculated from the capture amplitudes which characterize the alignment and, hence, the magnetic sublevel population of the recombined ion after the electron has been captured.

Selected previous applications: The REC component has been utilized to investigate the radiative electron capture into high-Z, few-electron ions within the framework of the density matrix [28] as well as the alignment [61] and polarization transfer following the inner-shell photoionization of sodium-like ions [62].

3.3.6. RELCI

Like RCi92 in GRASP92 [6], the RELCI component generates the (level) energies and atomic state functions (7) due to the set-up and diagonalization of some proper Dirac–Coulomb(–Breit) Hamiltonian matrix; it exploits the block-diagonal structure of the Hamiltonian without that the underlying .csl list need to be divided into blocks of the same total angular momentum and parity. A long write-up for this component has been published previously, and only some adaptations of the code have been made since then [35]. When compared with the RCi92 program, RELCI is faster by a factor of about 3...8 and also requires less memory. In addition, RELCI enables one to incorporate the dominant QED contributions into the level energies. In the future, however, this component may become obsolete in the long term when an improved RCi becomes available within the GRASP environment, based on Fortran 90/95, and if the RATIP program is to be adapted to such a new version.

(i) Input files: `element.iso`, `csl.inp`, `scf.rwf`.

(ii) Output files: `relci.sum`, `reclci.mix`.

Implemented interactions: In dealing with large wave function expansions (7), the decomposition and computation of the Hamiltonian matrix (9) is often a very time-consuming task due to the (angular) integration over the $3N$ spin-angular coordinates for each matrix element (10). In RELCI, all angular coefficients are now obtained from the ANCO component [21] to determine and compute the necessary one- and two-particle effective interaction strengths. Since ANCO provides *pure angular* coefficients, the same set of these coefficients can be used to generate the Dirac–Coulomb, Dirac–Gaunt, or Dirac–Coulomb–Breit matrices.

The two dominant QED contributions to the level structure of free atoms and ions are known to be the vacuum polarization and self-energy of the electrons. These two contributions roughly scale with the fourth power of the nuclear charge, Z^4 , and may become comparable in size with the Breit interaction, if inner-shell electrons are involved in some atomic process. While the vacuum polarization gives rise to an effective one-particle interaction strength, and can thus be incorporated in the Hamiltonian matrix, much larger effort is required in order to incorporate the self-energy (SE) consistently into the computations. In the RELCI component, we follow Kim's suggestion [63] and estimate the SE of the electrons from the electron density within a small volume of, say 40–60 fm, around the nucleus. Within this volume, the (effective) charge ratio between the self-consistent orbitals from the GRASP92 calculations and the hydrogenic solution for a point-like nucleus is determined, and is multiplied with the (known and tabulated) SE of the hydrogen-like ions. For ions with a point-like nucleus, such tabulations are given by Mohr [64] for 1s, 2s, and 2p orbitals and by Mohr and Kim [65] for ns, np, and nd ($n = 3, 4, 5$) orbitals. For large wave function expansions, this method has been found quite stable and fast [66].

Dialog for controlling the execution:

```
Enter a file name for the relci.sum file:
Enter the name of the configuration symmetry list file:
Enter the name of the isotope data file:
Modify default set-up and printout of the program ? [y]
Include contributions of the frequency-independent Breit interaction ? [y/n]
Include vacuum polarization contributions to H ? [y/n]
Include specific mass shift contributions to H ? [y/n]
Estimate contributions from self-energy ? [y]
Add self-energy estimates to the total energies in the .mix output file ? [y/n]
  This can be useful in the computation of transition energies.
Store the effective interaction strengths in memory ? [y/n]
  This should always be true for a on-fly calculation of the Hamiltonian matrix.
Precalculate and keep Hamiltonian matrix in memory ? [y/n]
Diagonalize the full Hamiltonian matrix ? [y/n]
  (I.e. independent of the parameter hamiltonian_fullmatrix.)
Which units are to be used to enter and to print the energies ?
  A      : Angstrom;
  eV     : electron volts;
  Hartree : Hartree atomic units;
  Hz     : Hertz;
  Kayser  : [cm*(-1)];
The physical speed of light in atomic units is 137.035989500; revise this value ? [y/n]
```

```

The default radial grid parameters for this case are:
:
  revise these values ? [y/n]
  ..... here now follows mandatory input again
Enter the name of the GRASP92 Radial WaveFunction File:
Enter a file name for the relci.mix file:
Enter the serial number(s) of the level(s) to be calculated;
e.g. 1 3 4 7 - 20 48 69 - 85;

```

Results and summary: Basic information about the Hamiltonian matrix and its set up as well as about the eigenenergies and weights of the leading CSF in the expansion of the atomic states are printed to the `relci.sum` summary file, analogue and in a rather similar format as in the `Rci92` component of GRASP92. In particular, three tables display the total level energies and the level splittings with regard to the lowest and next-lowest level in the computations and, hence, provide the user with a quick overview about the calculated level structure. If the SE estimates were requested to be calculated, they are displayed for each subshell, in addition, and the (three) tables from above are repeated once more with the SE estimates included into the level energies. Moreover, the energies, overall symmetries and eigenvector decomposition of the atomic states, chosen by the user, are listed in the `relci.mix` mixing coefficient file for further application by other components.

Special features: While the supported *interactions* to the Hamiltonian are the same as in the `Rci92` component [6], RELCI supports in addition a number of computational models with regard to the storage and time management. As default, a full diagonalization of the Hamiltonian matrix by means of some Lapack library routine is made if the block-size of the Hamiltonian matrix is not larger than the (global integer) parameter `hamiltonian_fullmatrix` (which is set to 500 in the present distribution). In all other cases, the Davidson algorithm is utilized for the levels of interest, using the Fortran 77 implementation by Stathopoulos and Froese Fischer [67]. Another default is to use only RAM memory, and this is well suitable for all matrices with dimension $n \lesssim 100,000$ at most present-day computers. Apart from this default setting, however, a *disc mode* can be selected in which the Hamiltonian matrix elements are written sequentially on disc during the generation and are later retained from there for diagonalization without keeping the full matrix at any time in the internal storage. In practice, this *disc mode* has rarely been utilized by us and might be removed in some forthcoming version of the RELCI component.

The self-energy estimates are not automatically incorporated into the mixing coefficient file, although this can be specified during input. Moreover, the `relci.mix` mixing coefficient file is now printed as default in a *formatted* file format; these and some other defaults can be easily changed by setting some logical flags within the RELCI module.

Selected previous applications: RELCI has been applied indeed in a wide range of applications as it provides the necessary mixing coefficient files for most other components of the RATIP program [68]. In addition, it has been utilized also (in a slightly modified form) to study hyperfine-induced electric and magnetic multipole transitions or the isotope shift parameters of a few medium and heavy elements [69,70]

3.3.7. REOS

The REOS program was implemented as (one of) the first components of RATIP to facilitate the computation of *relaxed-orbital* oscillator strength, transition amplitudes and lifetimes. This component exploits Löwdin's expression [48] for the evaluation of the transition amplitudes and is based on Slater determinants that are constructed from two sets of not quite orthogonal orbitals. Within the framework of RATIP, therefore, a prior expansion of all atomic states of interest into a determinant basis is required and can be performed by means of the CESD component, before the REOS program can be applied. A long write-up of REOS was given in Refs. [36] and not much has been changed in its implementation since then. Like in the EINSTEIN program, REOS can be applied both, to the absorption or emission of photons in dependence of how the initial and final atomic states are chosen, but by including all contributions to the transition amplitudes that arise from the *non-orthogonality* of the radial orbital functions in the representation of these states. In addition, the REOS component also provides a transition amplitude file (`reos.trn`) for further applications of these (reduced) matrix elements.

(i) Input files: `element.iso`, `initial-cesd.xpn`, `initial-scf.rwf`, `final-cesd.xpn`, `final-scf.rwf`.

(ii) Output files: `reos.sum`, `reos.trn`.

Implemented transition amplitudes: Like the EINSTEIN component in Section 3.3.2, REOS evaluates the same (reduced multipole) transition amplitudes

$$\langle \alpha_f J_f \| H_\gamma(\pi L) \| \alpha_i J_i \rangle = \sum_{r,s} c_{fr} c_{is} \left\langle \gamma_f P_f J_f \left\| \sum_p \alpha_p \cdot \mathbf{A}_{L,p}^\pi(\mathbf{k}) \right\| \gamma_i P_i J_i \right\rangle, \quad (41)$$

for bound-state transitions $|\psi_i\rangle \rightarrow |\psi_f\rangle$ but for the case that the initial and final ASF are expressed as linear combination of Slater determinants. These *relaxed-orbital* amplitudes (41) are combined then in order to obtain line strength, Einstein A and B coefficients or radiative or lifetimes. See Ref. [36] for all further details about the implementation of the REOS component.

Dialog for controlling the execution:

```

Enter a file name for the reos.sum file:
Restart a previously aborted calculation ? [n]
Enter the name of the isotope data file:
Enter the transition multipoles, e.g. E1 M2 ... :
Which units are to be used to enter and to print the energies ?

```

```

A      : Angstrom;
eV     : electron volts;
Hartree : Hartree atomic units;
Hz     : Hertz;
Kayser  : [cm**(-1)];
Modify default set-up and printout of the program ? [y/n]
Select individual transitions ? [y]
Enter one pair |i> - |f> of level numbers, e.g. 2 - 3; 2 - 0; 0 - 17 ...
(0 is here equivalent to all); <cr> if done.
Assume orthogonality between the orbital sets of the initial and final atomic states ? [y/n]
This feature accelerates the computations but also neglects all relaxation effects.
Calculate all radial integrals only once at the beginning of the computation ? [y/n]
This feature accelerates the computations but may require considerable more memory.
Enable the restart of the calculation if the run cannot be completed successfully ? [y/n]
This feature dumps all necessary information to a restart file.
Sort transitions in ascending order of energy ? [y/n]
Read in and apply experimental energies in the calculation ? [y/n]
Einstein A and B coefficients are printed in SI units; use atomic units instead ? [y/n]
Print all selected transitions and their energies before the computation starts ? [y/n]
Write out the transition energies and amplitudes to an .trn file ? [y/n]
Enter a file name for the reos.trn file:
Fix a cutoff criterium other than '10e-8' to neglect small admixtures to the
computation of transition probabilities ? [n]
The physical speed of light in atomic units is 137.035989500; revise this value ? [y/n]
The default radial grid parameters for this case are:
:
revise these values ? [y/n]
..... here now follows mandatory input again
Enter the name of the initial-state CESD .xpn expansion file
Enter the name of the final-state CESD .xpn expansion file
Enter the name of the initial-state Radial WaveFunction File:
Enter the name of the final-state Radial WaveFunction File:

```

Results and summary: The .sum summary files of the REOS and EINSTEIN components have a very similar format and provide especially a neat table of all transition energies, Einstein coefficients, oscillator strength and the associated decay widths at the end of the computations. As before, the atomic levels (and corresponding ASF) refer typically to two different GRASP92 calculations and (subsequent) CESD expansions of the initial and final states into a determinant basis, although the *same* set of wave functions (here .xpn and .rwf files) can be utilized if both, the initial and final states were described before by the same run of GRASP92. From all the calculated multipole transitions with positive (transition) energy, moreover, the lifetimes and width of the *upper* levels are compiled in the reos.sum summary file, while the amplitudes are listed in the reos.trn file for further use in other components or within the toolbox of RATIP.

Special features: As in the EINSTEIN component, all electric-multipole transitions (E1, E2, ...) are always calculated within the Babushkin and Coulomb gauge, and which are better known as length and velocity gauge from the nonrelativistic theory. If no individual transitions are selected, i.e. special pairs of initial and final level numbers from prior GRASP92 calculations, all multipole transitions with positive (transition) energy are chosen automatically. Moreover, “experimental” transition energies can be specified explicitly by the user, an option that has been found helpful if the calculated fine-structure of the atomic levels is not accurate. A further “restart” option became quite obsolete during recent years and might be removed from the code in the future.

Selected previous applications: The REOS component has been applied earlier in a number of medium-to-large scale calculations of transition probabilities and lifetimes [71,72], and especially for multiply charged ions from the iron group [73–75]. More recently, REOS was utilized for studying two-electron-one-photon transitions [76], two-photon excitation [77] and decay cross section [78] as well as in dielectronic recombination studies.

3.4. The TOOLBOX of RATIP

With the increasing number and variety of applications of the RATIP program, several small tasks need to be performed more or less frequently with regard to the (de-)composition and manipulation of data. Most of these tasks do not justify an independent component (subprogram) but are included more conveniently within the (so-called) TOOLBOX of the RATIP program; these *tools* extend some development to which we referred as “utilities” before [79]. In most of these tasks, files from GRASP92 and/or RATIP components are handled, and all manipulations are controlled interactively. Since further tasks are added to the TOOLBOX from time to time, here I shall explain only the basic structure of these *tools* and leave it otherwise to the user to explore further features. Although various “new” files can be created by means of the TOOLBOX, most tasks print their results directly to screen.

The various tasks, that are supported by the TOOLBOX, can be selected from a (two-level) menu and are further controlled by some (interactive) “dialog”.

3.4.1. Main menu

The main menu of the TOOLBOX distinguishes between certain groups of (more or less) related tasks and is displayed immediately after this component has been invoked:

Select from the menu:

- A - Level energies, notations and weights.
- C - Manipulation of GRASP92 .csl lists and .mix files.
- D - Format or unformat a GRASP92 file.
- E - Properties of atomic orbitals.

- M - Miscellaneous.
- N - Nuclear distributions and parameters.

- Q - Quit the program.

For each of these groups, then, a submenu appears if the corresponding letter (not case sensitive) is selected. This two-level structures makes it easy to add other features or even to rearrange the sequence of tasks as this (may) become necessary or appropriate in the future.

3.4.2. Level energies and configuration states

For example, if we choose A (-- Level energies, notations and weights), the submenu

A

Select from the menu:

- 1 - Energy levels and level splittings from one or several .mix files.
- 2 - Energy levels and level splittings, extended form.
- 3 - Energy levels and level splittings from one .mix file with LSJ notations.
- 4 - Display the major jj-coupled CSF and their weights to atomic levels.
- 5 - Display the major LS-coupled CSF and their weights to atomic levels.

- r - Return to the main menu.

appears and enables one to tabulate the level energies from one or several GRASP92 calculations. This can be done even if these bound-state calculations refer to a different number of electrons, a feature that has been found very useful if photo- or autoionization processes are analyzed, or if one wishes to identify energetically allowed transitions from some large transition array. Task 2, for instance, enables one to specify the energy units in the printout of the data as well as the “reference” level with regard to which the (excitation) energies are taken. Task 3, moreover, also performs a transformation of the levels into a *LSJ*-coupled basis [80] and lists the corresponding leading contributions.

3.4.3. Manipulation of configuration lists and mixing coefficient files

If C (-- Manipulation of GRASP92 .csl lists and .mix files) is chosen, we will see the submenu:

C

Select from the menu:

- 1 - Exclude a number of CSF from a GRASP92 .csl list.
- 2 - Split a GRASP92 .csl list into J^P level groups.
- 3 - Merge two .csl lists from GRASP92 with the same core.
- 4 - Merge two .csl lists from GRASP92 with different cores.
- 5 - Condense a .csl list from GRASP92 on a single weight criterium.
- 6 - Condense a .csl list from GRASP92 as above but for some given levels only.
- 7 - Reduce a .mix file from GRASP92 using level numbers.
- 8 - Reduce a .mix file from GRASP92 using symmetry properties (*J* and *P*).
- 9 - Generate a pair-list from a given .csl list with regard to a reference list.

- r - Return to the main menu.

These features enables one to split, merge or condense some given .csl lists from GRASP92 due to further criterions which can be specified interactively. They also enable one to “remove” certain eigenvector (contributions) from a mixing coefficient file or configuration state list if these CSF have zero Hamiltonian matrix elements with regard to a given *reference* list of CSF.

3.4.4. Some miscellaneous features

If we choose M (-- Miscellaneous) from the main menu of the TOOLBOX, the submenu:

M

Select from the menu:

- 1 - Effective charge of a given orbital.

2 - Effective radial charge or charge density of a selected ASF.
 4 - GRASP92 grid-parameter calculator.
 r - Return to the main menu.

provides us with features to calculate the effective (radial) charge of either a given orbital $a = (n_a, \kappa_a)$,

$$\rho_a(r) = \int_0^r ds (P_a^2(s) + Q_a^2(s)) \quad (42)$$

or atomic state function $\psi_\alpha(PJM)$ from ansatz (7)

$$\rho_\alpha(r) = \sum_a \bar{q}_{\alpha,a} \rho_a(r), \quad (43)$$

and where $\bar{q}_{\alpha,a}$ refers to the mean occupation number of orbital a for the state ψ_α . These effective radial densities appear frequently in the semi-classical collision theory in order to account for a given impact parameter b the screening of the nuclear charge by the (surrounding) electrons [81]. In this submenu, we also provide a *grid-parameter calculator* that help determine the size of the grid or number of grid points, provided that the other grid parameters are given by the user.

These few (sub-)menus give a first impression about the TOOLBOX of the RATIP program; several other features already exists or might be added in the future and can be easily explored interactively.

3.5. Distribution and installation of the code

At present, as explained above, only those components of the RATIP program are distributed which have been tested and applied already in a good number of case studies, and which were summarized in Sections 3.3 and 3.4. For these components, the source code is comprised within 38 modules and four libraries which need to be compiled and linked to the code separately. The libraries `libblas.a`, `libdvdson.a`, `liblapackd.a` and `liblapacku.a` are related to some subroutines from GRASP92 [6] and can be *linked* directly from this code, if GRASP92 has been properly installed before. However, these libraries are appended here again for the sake of completeness.

All source code is provided by the `ratip-2012` root directory which, in addition, includes several make files for generating the executable as well as test suites for various components. As for the RELCI component before [35], the command `make -f make-component` generates the executable of the corresponding component (i.e. `xauger`, `xcesd`, `xeinstein`, `xphoto`, `xrelci`, ...). However, for using these makefiles, the script `make-environment` need to be adapted to the local environment and sourced in order to define a number of global variables for the compilation and linkage of the program. This script facilitates the use and transfer of the makefiles if the local architecture is to be changed, or if another compiler (or flags) are to be used, while the structure of the individual makefiles remains quite simple. Both, previous as well as the present version of RATIP has been found portable (more or less easily) to different compilers, including `gfortran` (up to version 4.6, recently), and can thus be used on different platforms.

The `ratip-2012` root directory is distributed as a compressed tar file `ratip-2012.tar.gz`, including all the makefiles, subdirectories and a short `Read.me`. From this file, the root is restored by the command `tar -xvf ratip-2012.tar`. The simple structure of the `ratip` root will facilitate the extension of the RATIP program by further components in the future; in fact, this is readily achieved by appending the additional modules, (main) programs, and corresponding makefiles. Since several coworkers helped in testing the code and analyzing results from the various components, this *modular* structure ensures that additional code developments can be performed and that the maintenance on both, the user and developer side, still remains straightforward.

4. Test examples

As obvious already from Fig. 1, example calculations can be displayed and discussed here only for a (very) few components of the RATIP program. Especially, we wish to show how the various program components *do* work together and how some (small-scale) examples may serve also as test of the local installation. For this reason, we shall consider the (sequential) *double* ionization of atomic cadmium which has been observed with synchrotron light between about 40–200 eV, and which is known to proceed predominantly via the single ionization of a 4d or 4p electron, and followed by a subsequent Auger decay of the inner-shell excited Cd^+ hole state [82]. For these test computations, we here *assume* that all bound-state wave functions were generated (before) by means of the GRASP92 program [6]; these wave functions are provided in the subdirectories `test-photo` and `test-auger` within the `ratip-2012` root directory. In Section 4.3, moreover, we shall demonstrate a few quick “calls” to the TOOLBOX component, and by making use of the same wave functions.

Of course, more advanced case studies (and tests) can be made by the user by running these or other components of the RATIP program, and by following the corresponding interactive dialogs. Usually, a basic understanding of the behavior of the *active* electron(s) and the atomic levels, which are involved in the underlying (atomic) process, should be sufficient in order to generate meaningful results. A few further tests of the RATIP code are provided also in other `test-component` subdirectories within the `ratip-2012` root and are briefly described in the corresponding `Read.me-component` files.

4.1. 4p and 4d photoionization of atomic cadmium

In Ref. [82], the double ionization of atomic cadmium by extreme ultra-violet (EUV) synchrotron radiation was investigated in detail. For such a low-intensity light source, the double ionization to the $\text{Cd}^{2+}4p^64d^{10}1S_0$ ground state mainly proceeds through the (photo-)

ionization and subsequent Auger decay of a $4d$ hole state and the formation of corresponding correlation satellites, whereas the *direct* double ionization were observed to be weak. In this work, angle-integrated single-photon double ionization electron spectra of Cd were measured at photon energies between 40–200 eV, both below and above the $4p$ threshold. Moreover, a multielectron coincidence technique was employed to reveal the complete electron pair energy distributions, and this gave rise to important information about the double ionization processes involved.

In the analysis of the observed electron spectra, MCDF calculations helped to clarify the main mechanisms for the formation of the Cd^{2+} ground state. Starting from the $4d^{10}5s^2\ ^1S_0$ ground state of neutral Cd ($Z = 48$), photoionization and Auger computations have been performed in order to better understand the photo- versus autoionization as well as the associated electron spectra. Typically, the successive ionization first leads to either a $4p$ or $4d$ hole state, and subsequently to a Cd^{2+} ion. Apart from the (nominal) $4d^{10}5s^2$, $4d^{10}5s$ and $4d^{10}$ reference configurations of the neutral atom as well as the singly and doubly-charged ions, all single and double excitations into the $5p$, $5d$, $6s$, $6p$, and $7s$ shells were taken into account in a series of calculations with systematically enlarged wave functions in order to provide a proper flexibility for the description of the (photo-)ionization and subsequent decay processes. For open d -shell structures, however, the size of the wave function expansions increases so rapidly that no attempt was made to include further virtual excitations in the construction of the wave functions. Moreover, the orbitals were optimized on the basis of the Dirac–Coulomb Hamiltonian (cf. Section 2.2) and the Breit interaction was included into the low-lying level structure of cadmium, while further QED corrections remain negligible typically as long as no deep core shells are involved into the ionization process.

Here, we shall not recall all details from the computations in Ref. [82] but restrict ourselves to single-configuration calculations in order to demonstrate the use of the PHOTO and AUGER components. For the $4d$ photoionization of the $4d^{10}5s^2\ ^1S_0$ ground level, the cross section for producing some final-ionic level $|\psi_f\rangle = |\psi(P_f J_f)\rangle$ is obtained within the electric-dipole approximation by

$$\sigma(P_f J_f) = \frac{4\pi^2 \alpha \omega}{3(2J_i + 1)} \sum_{\kappa_c, J_t} |D(\omega; J_f P_f, \epsilon \kappa_c : J_t P_t)|^2 \quad (44)$$

and can readily be calculated by means of the PHOTO component. In formula (44), the summation over the scattering states runs over all possible partial waves of the photoelectron with kinetic energy $\epsilon = E_i + \omega - E_f$ as well as over the allowed continuum states $|\psi_t\rangle \equiv |J_f P_f, \epsilon \kappa_c : J_t P_t\rangle$ with total angular momentum $J_t = 1$ and odd parity.

In the subdirectory test-photo, we have generated (single-configuration) wave functions for the initial $4d^{10}5s^2\ ^1S_0$ ground level of neutral cadmium and the four $(4p^5 4d^{10} + 4p^6 4d^9)5s^2\ ^2P_{1/2,3/2}$ and $^2D_{3/2,5/2}$ levels of the photoion by means of GRASP92 and the RELCI component. These wave functions are provided by the files cd-neutral-sc-csl.inp, cd-neutral-sc-scf.out and cd-neutral-sc-relci.mix for the neutral atom and by cd-singly-sc-csl.inp, ... for the photoion. The calculation of the photoionization cross sections (44) is then achieved with the help of the PHOTO components and by following the interactive dialog (cf. Section 3.3.4)

```
> ../xphoto
  PHOTO: Calculation of photoionization cross sections, angular distribution parameters,
        and others (Fortran 95 version).
  Enter a file name for the photo.sum file:
photo-neutral-singly-sc.sum
  Enter the name of the isotope data file:
isodat48
  loading isotope data file ...
  ... load complete;
  Enter the transition multipoles, e.g. E1 M2 ... :
E1
  Which units are to be used to enter and to print the energies of the continuum orbitals ?
    A      : Angstrom;
    eV     : electron volts;
    Hartree : Hartree atomic units;
    Hz     : Hertz;
    Kayser  : [cm**(-1)];
eV
  Enter the maximal energy of the photo lines (in eV) to built-up the radial grid;
300.
  The photon energies can be entered either individually or by
  an appropriate interval and step size.
  The default is a list of individual energies; revise this ?
Y
  Enter an interval of photon energies and a corresponding stepsize;
E_lower E_upper delta-E:
50. 151. 50.
  Modify default set-up and printout of the program ?
Y
  Select individual transitions ?
n
  Include exchange interactions into the generation of the continuum waves ?
Y
  Calculate angular distribution parameters ?
Y
  Calculate spin polarization parameters ?
```

```
n
:
:
```

from which we display only the first few lines here; all the answers to this “dialog” are given also within the file `job-photo-cd` in the subdirectory `test-photo` and can be executed more conveniently by the command `xphoto <job-photo-cd`. As seen from this dialog, we restrict the computations to the electric-dipole amplitudes and select photon energies between 50 and 151 eV, and with a given stepsize of 50 eV. Moreover, all (photo-)transitions between the initial $4d^{10}5s^2\ ^1S_0$ ground level and the four final levels of the $(4p^54d^{10} + 4p^64d^9)5s^2$ configurations are taken into account, including the *exchange* interaction between the photo electron and the bound-state electron density. Apart from the photoionization cross sections (as default), the angular distribution parameters β_2 are calculated in Babushkin (length) and Coulomb (velocity) gauge but no spin-polarization parameters are selected for the present run.

While the standard output of the `PHOTO` component is printed to screen (or some *log*-file if specified at the beginning), all important results are compiled also in the `photo-neutral-singly-sc.sum` summary file. This file contains a short tabulation for each photo line

Results for PI line from the transition 1 - 1: 0 + ----> 5/2 +

```
-----
Photon energy      = 5.00000000E+01 eV
Electron energy    = 3.38292891E+01 eV
Total cross section = 1.89559595E+01 Mb (Babushkin gauge)
                  = 1.25584454E+01 Mb (Coulomb gauge)
Beta              = 3.14584243E-01 ; 3.06212327E-01 (Babushkin; Coulomb)
Alignment (final) = -2.30680218E-01 ; -2.09351082E-01 (Babushkin; Coulomb)
```

Kappa	Total J ^P	Mp	Gauge	Amplitude		Real-Amplitude	Rate (Mb)	Phase
p	1-	E1	Babushkin	1.832E-03	-4.027E-03	-4.424E-03	5.48137E-04	3.569E+00
p	1-	E1	Coulomb	1.499E-03	-3.294E-03	-3.619E-03	3.66793E-04	3.569E+00
f-	1-	E1	Babushkin	4.529E-03	9.085E-04	-4.620E-03	5.97591E-04	1.769E+00
f-	1-	E1	Coulomb	4.296E-03	8.616E-04	-4.381E-03	5.37567E-04	1.769E+00
f	1-	E1	Babushkin	1.773E-02	-6.484E-03	-1.888E-02	9.98304E-03	1.220E+00
f	1-	E1	Coulomb	1.427E-02	-5.220E-03	-1.520E-02	6.46852E-03	1.220E+00
:								

as well as tables for the cross sections, angular distribution parameters and the alignment of the photo ions.

Individual and total photoionization cross sections :

```
-----
```

LevI-LevF	I- J / Parity -F	omega (eV)	e-Energy (eV)	CS-Babushkin (Mb)	CS-Coulomb (Mb)

1 - 1	0 + 5/2 +	5.00000E+01	3.38293E+01	1.896E+01	1.256E+01
1 - 1	0 + 5/2 +	1.00000E+02	8.38293E+01	5.206E+00	2.543E+00
1 - 1	0 + 5/2 +	1.50000E+02	1.33829E+02	1.656E-01	1.229E-01
1 - 2	0 + 3/2 +	5.00000E+01	3.31614E+01	1.337E+01	9.612E+00
:					

```
-----
```

Photoionization angular parameters:

```
-----
```

LevI-LevF	I- J / Parity -F	omega (eV)	e-Energy (eV)	Beta (Babushkin)	Beta (Coulomb)	Beta (Average)

1 - 1	0 + 5/2 +	5.00000E+01	3.38293E+01	3.146E-01	3.062E-01	3.104E-01
1 - 1	0 + 5/2 +	1.00000E+02	8.38293E+01	1.369E+00	1.466E+00	1.418E+00
1 - 1	0 + 5/2 +	1.50000E+02	1.33829E+02	5.131E-01	-4.277E-01	4.270E-02
1 - 2	0 + 3/2 +	5.00000E+01	3.31614E+01	3.503E-01	3.434E-01	3.469E-01
:						

```
-----
```

```
-----
```

LevI-LevF	I- J / Parity -F	omega (eV)	e-Energy (eV)	Alignment (Babushkin)	Alignment (Coulomb)	Alignment (Average)

1 - 1	0 + 5/2 +	5.00000E+01	3.38293E+01	-2.307E-01	-2.094E-01	-2.200E-01

```
-----
```

```

1 - 1      0 +      5/2 +      1.00000E+02      8.38293E+01      -2.889E-01      -2.946E-01      -2.918E-01
1 - 1      0 +      5/2 +      1.50000E+02      1.33829E+02      -7.145E-01      -6.853E-01      -6.999E-01
1 - 2      0 +      3/2 +      5.00000E+01      3.31614E+01      -2.227E-01      -2.227E-01      -2.227E-01
:

```

For a further analysis of the $4d$ and $4p$ photoelectron spectra, of course, more involved calculations need to be carried out since, for cadmium especially, the ionization of the subvalence shells is often accompanied also by some $5s \rightarrow (6s + 7s + 5p + 5d)$ shake-up transitions. This, in turn, also affects the subsequent autoionization to the $4d^{10}1S_0$ ground level and to a rather large number of $4d^9nlj^{2S+1}L_J$ final levels of Cd^{2+} , and this leads overall to quite complex electron spectra from which we considered here only some of the main features.

4.2. Subsequent Auger emission from the $4p$ and $4d$ hole states

To explore the subsequent autoionization of the $4p^{-1}$ and $4d^{-1}$ hole states of Cd^+ , we can make use of the AUGER component and the wave functions in `test-auger`. The final states of the prior photoionization process, that arise from the $(4p^54d^{10} + 4p^64d^9)5s^2$ configurations, now become the *initial* states of the autoionization, while further wave functions need(ed) to be generated for the $4p^6(4d^{10} + 4d^95s)$ levels of Cd^{2+} . As before, we make use here of pre-generated wave functions `cd-singly-sc-csl.inp`, ... for the Cd^+ ions and `cd-doubly-sc-csl.inp`, ... for the Cd^{2+} ions, respectively. Moreover, the input data for the AUGER component are compiled in the file `job-auger-cd` and can be executed by invoking `../xauger < job-auger-cd` in the `test-auger` subdirectory. Not much need to be said about this input in which we include again the exchange interaction for the outgoing electron but no Breit interaction into the (Auger) transition amplitudes and rates. In these single-configuration calculations, only the two $4p^{-1}$ holes states (levels 3 and 4 in `cd-singly-sc-csl.inp`, ...) contribute to the subsequent Auger emission as we will see again in the next subsection.

Apart from the standard output, which is printed to screen, all major results are listed again in the file `auger-singly-doubly-sc.sum`. This summary file contains a short table about each Auger line as well as a tabulation about all rates and lifetimes

```
Results for Auger transition      3 -      1 :      3/2 -      ---->      0 +
-----
```

```

Energy      =      54.1827348516964      eV
Total rate  =      278329046342.734      1/s

```

Kappa	Amplitude	Real-Amplitude	Rate (1/s)	Phase
p	-9.744E-04	-3.494E-04	-1.035E-03	2.783E+11
:				1.227E+00
:				
:				

Individual Auger rates :

LevI-LevF	I- J / Parity -F	Energy (eV)	Rate (1/s)
3 - 1	3/2 -	0 + 5.41827E+01	2.783E+11
3 - 2	3/2 -	3 + 4.57040E+01	1.476E+14
3 - 3	3/2 -	2 + 4.54448E+01	1.319E+14
3 - 4	3/2 -	1 + 4.50110E+01	8.989E+12
3 - 5	3/2 -	2 + 4.45783E+01	1.562E+14
4 - 1	1/2 -	0 + 6.00086E+01	4.084E+06
:			

Auger lifetimes, total rates and widths :

Level	Lifetime	Total rate	Width		
	Seconds	1/s	Hartrees	Kaysers	eV
3	2.2474550D-15	4.4494773D+14	1.0762769D-02	2.3621548D+03	2.9286998D-01
4	2.6045523D-15	3.8394314D+14	9.2871390D-03	2.0382914D+03	2.5271602D-01

In all these tables, the level numbers refer to the GRASP92 calculations for the corresponding initial and final (bound) states of the considered process. From the tabulation of the (individual) Auger rates, we can easily identify the $4d^{10}1S_0$ ground level (level 1 of the final states) as well as the 3D_3 , 3D_2 , 3D_1 , and 1D_2 levels of the $4d^95s$ configuration (levels 2–5). In the measurements [82], the latter four levels of the $4d^95p$ configuration were partially resolved and identified by means of (level) data known from optical spectroscopy.

For energies above of the 4p and 4s ionization threshold (not seen in this single-configuration calculations), the observed electron spectra for the double ionization of atomic cadmium are strongly influenced by Coster-Kronig (CK) transitions. This was seen in Ref. [82] in the intensity of the $4d^8 5s^2$ states which are mainly populated by super-CK decay of a 4p hole. However, in order to resolve the 4p photoelectron from the CK electron, photon energies above 100 eV are required due to large lifetime width. Rather independent of the initial photon energy, the Cd^{2+} ground state appears to always be reached through the same excited Cd^+ states in this energy range, despite the opening of various inner shells at the higher photon energies. In practice, this CK decay can also be analyzed by means of the AUGER component but has not been included in the present example in order to keep the (amount of) input and output data feasible.

4.3. Making use of the TOOLBOX

To understand the possible excitation and decay paths in the ionization of atoms and ions, one often wishes to first analyze the level structure of the different charge states that are involved into some process. For example, an ion can autoionize only if its (initial) state is *embedded* into the continuum of the next higher charge state, i.e. if $E_{\text{total}}(N) - E_{\text{total}}(N-1) > 0$. By using the TOOLBOX of the RATIP program and the wave functions prepared in Sections 4.1 and 4.2, we can easily identify the allowed decay channels by displaying the excitation energies of all levels from the $4p^6 4d^{10} + (4p^5 4d^{10} + 4p^6 4d^9) 5s^2 + 4p^6 (4d^{10} + 4d^9 5s)$ configurations together for the lowest three charge states of Cd as obtained in the GRASP92 computations above. This is achieved by (invoking the TOOLBOX component within the test-auger directory)

```
> ../xtoolbox

Select from the menu:

A - Level energies, notations and weights.
C - Manipulation of GRASP92 .csl lists, .mix and .out files.
D - Format or unformat a GRASP92 file.
E - Properties of atomic orbitals.

M - Miscellaneous.
N - Nuclear distributions and parameters.

Q - Quit the program.
A
Select from the menu:

1 - Energy levels and level splittings from one or several .mix files.
2 - Energy levels and level splittings, extended form.
3 - Energy levels and level splittings from one .mix file with LSJ notations.
4 - Display the major jj-coupled CSF and their weights to atomic levels.
:
2
Returns the energy levels and excitation energies of some ASF from one or several .mix files.
For the printout of the excitation energies, the energy units, an ascending/\xch{descending}{decending} order,
and the level, relative to which the excitation energies are taken, can be selected below.

Continue (y/n) ?
y
Which units are to be used to enter and to print the energies of the continuum orbitals ?
A      : Angstrom;
eV      : electron volts;
Hartree : Hartree atomic units;
Hz      : Hertz;
Kayser  : [cm**(-1)];
eV
Energies are printed in ascending order; use a \xch{descending}{decending} order instead ?
n
Enter the level number of the reference (zero-) level:
1
Enter one (or several) .mix mixing coefficient file(s):
../test-photo/cd-neutral-sc-relci.mix cd-singly-sc-relci.mix cd-doubly-sc-relci.mix
```

Level	J	Parity	[File / [Level]]	Total energy (Hartrees)	Level splitting (eV)	Excitation energy from the reference (eV)
1	0	+	[1, 1]	-5.58947727E+03	0.00000000E+00	0.00000000E+00
2	5/2	+	[2, 1]	-5.58888301E+03	1.61707109E+01	1.61707109E+01
3	3/2	+	[2, 2]	-5.58885846E+03	6.67845653E-01	1.68385566E+01

4	0	+	[3, 1]	-5.58859175E+03	7.25768728E+00	2.40962438E+01
5	3	+	[3, 2]	-5.58828016E+03	8.47873496E+00	3.25749788E+01
6	2	+	[3, 3]	-5.58827064E+03	2.59170701E-01	3.28341495E+01
7	1	+	[3, 4]	-5.58825469E+03	4.33828684E-01	3.32679782E+01
8	2	+	[3, 5]	-5.58823879E+03	4.32712839E-01	3.37006910E+01
9	3/2	-	[2, 3]	-5.58660057E+03	4.45782877E+01	7.82789787E+01
10	1/2	-	[2, 4]	-5.58638647E+03	5.82584204E+00	8.41048207E+01

Instead of the (energetically) lowest level of all three .mix files, we could have selected another energy unit as well as another reference level with regard to which the excitation energies are calculated and displayed. As easily seen from this tabulation, only levels 3 and 4 (from the singly-charged ion in cd-singly-sc-relci.mix-dcb) can contribute to the subsequent Auger emission as levels 1 and 2 are energetically too low.

As described in Section 3.4, the TOOLBOX helps in solving a number of small but (more or less) frequently occurring tasks. Further tasks are likely to be added as new requests and applications of the RATIP program will occur. The user is encouraged and may benefit from knowing and exploring the variety of tasks that can be handled by the TOOLBOX component. Here, we can only display a few selected tasks; as a final example, let us consider the overlap of the one-electron orbitals between the $4p^{-1}$ and $4d^{-1}$ hole states of Cd^+ and the $4p^6(4d^{10} + 4d^95s)$ levels of Cd^{2+} . This is obtained by calling again the xtoolbox (in the test-auger subdirectory) and gives rise to

```
> ../xtoolbox

Select from the menu:

:
E
Select from the menu:

1 - Radial properties of atomic orbitals.
2 - Overlaps between two not quite orthogonal orbital sets.

r - Return to the main menu.
2
Calculates the 'overlaps' of orbitals with the same symmetry for two not quite orthogonal
sets of orbitals. However, the orbitals within each set are supposed to be orthogonal.

Continue (y/n) ?
Y
Enter the name of the initial-state GRASP92 configuration symmetry list file:
cd-singly-sc-csl.inp
Loading configuration symmetry list file ...
There are      15 relativistic subshells;
there are      4 relativistic CSFs;
... load complete.
Enter the name of the final-state GRASP92 configuration symmetry list file:
cd-doubly-sc-csl.inp
Loading configuration symmetry list file ...
There are      15 relativistic subshells;
there are      5 relativistic CSFs;
... load complete.
Enter the name of the initial-state Radial WaveFunction File:
cd-singly-sc-scf.out
... load complete;
Enter the name of the final-state Radial WaveFunction File:
cd-doubly-sc-scf.out
... load complete;

Overlap integrals <n_f kappa | n_i kappa> :
-----

+ s symmetry: < 1s | 1s > = 1.0000E+00 < 1s | 2s > = -3.8034E-06 < 1s | 3s > = -1.9139E-06
----- < 1s | 4s > = -1.0598E-06 < 1s | 5s > = -3.0317E-07 < 2s | 1s > = 3.8033E-06
< 2s | 2s > = 1.0000E+00 < 2s | 3s > = -2.2078E-05 < 2s | 4s > = -1.5510E-05
< 2s | 5s > = -4.4588E-06 < 3s | 1s > = 1.9137E-06 < 3s | 2s > = 2.2075E-05
< 3s | 3s > = 1.0000E+00 < 3s | 4s > = -1.8095E-04 < 3s | 5s > = -4.7061E-05
< 4s | 1s > = 1.0571E-06 < 4s | 2s > = 1.5478E-05 < 4s | 3s > = 1.8073E-04
< 4s | 4s > = 9.9999E-01 < 4s | 5s > = -2.6449E-03 < 5s | 1s > = 3.2821E-07
< 5s | 2s > = 4.7586E-06 < 5s | 3s > = 4.9098E-05 < 5s | 4s > = 2.7600E-03
< 5s | 5s > = 9.9553E-01

+ p- symmetry: < 2p- | 2p-> = 1.0000E+00 < 2p- | 3p-> = -3.6302E-06 < 2p- | 4p-> = 5.1416E-05
----- < 3p- | 2p-> = 3.5536E-06 < 3p- | 3p-> = 1.0000E+00 < 3p- | 4p-> = 1.4899E-03
```

$$\langle 4p- | 2p- \rangle = -5.1420\text{E-}05 \quad \langle 4p- | 3p- \rangle = -1.4898\text{E-}03 \quad \langle 4p- | 4p- \rangle = 1.0000\text{E+}00$$

:

While the inner-shell electrons are not much affected by the (auto-)ionization of the system, a noticeable overlap occurs for the (sub-)valence shells and provides some *qualitative* insight why shake-up and shake-off processes play such an important role in the double ionization of cadmium.

5. Summary and outlook

The RATIP program for relativistic calculations of atomic transition, ionization and recombination properties has been presented and some of its recent applications briefly reviewed. Over the past decade, the number of properties that can be studied by this program has been increased continuously and has made RATIP a powerful tool for rather a broad community in physics, from atomic photoionization and electron spectroscopy to the study of highly-charged ions, the spectroscopy of heavy and superheavy elements, the generation of atomic data for astro and plasma physics, and up to the search for time-reversal violating interactions in atomic systems. Based on Fortran 90 and its subsequent standards, RATIP combines the numerical strength of Fortran with an object-oriented approach in dealing, for example, with quantum numbers, atomic and configuration states, or many-electron transition amplitudes.

From the various properties (and the code that is necessary to deal with them), those components of the RATIP program are explained in more detail and distributed in the present distribution, which have been tested already in a number of case studies. In particular, all components from Sections 3.3 and 3.4 are provided in the present distribution of the program. Other components still need either further development or more tests, and these components will be made available only later by some up-date of the program.

As already indicated in Fig. 1, there are several further program components which are currently under work or test, and which will make RATIP useful for future investigations. Apart from the incorporation of plasma effects within some *mean-field* approach into the calculation of transition amplitudes and probabilities of multiply charged ions [72], recent developments concern the following – given in alphabetic order, and more or less completed – components:

- CONCI generates approximate many-electron continuum states as they occur in the summation over intermediate states in all second- and higher-order transition processes. This component makes use of the one-electron orbital functions from the SPECTRUM component to construct ASF with one or more electrons in the continuum. An approximate many-electron spectrum is generated by diagonalizing the Hamiltonian matrix similar as in RELCI but with additional restrictions upon the “free-free” matrix elements of the Hamiltonian. Different approximations are currently explored but not much is known so far how well such a computational scheme can be employed for describing second- or higher-order processes.
- COULEX calculates Coulomb excitation amplitudes, cross sections and alignment parameters in ion–atom collisions. This component is based on a semi-classical picture in which the projectile passes the target at some distance b , the so-called impact parameter, with a fixed velocity $\beta_p = v_p/c$ along a straight-line trajectory, and where c is the speed of light here. A Coulomb excitation (or ionization) of the projectile may then occur in this picture due to the Lienard–Wiechert potential of the target atoms, as seen by the projectile electrons. In the COULEX component, the integration over the time and impact parameter is carried out analytically by means of a Fourier transformation, and the so obtained excitation amplitude is decomposed into a series of multipole components [83]. This approach has been found useful especially for multiple and highly-charged ions; cf. Ref. [84].
- DIEREC helps calculate energies, rates, capture cross sections and resonance strengths for the dielectronic recombination of atoms and ions. It makes use of three sets of wave functions to represent the *initial* ($N - 1$ electron) states before the capture, the *intermediate* resonant states as well as the *final* states after the radiative stabilization has taken place. DIEREC calls internally the AUGER and EINSTEIN components to automatically generate and combine all required Auger and radiative amplitudes for the property of interest in the given recombination system. All radiative amplitudes are calculated simultaneously in both, Babushkin (length) and Coulomb (velocity) gauge, in order to recognize possible limitations in the computational model.
- EIMEX supports the calculation of electron-impact excitation cross sections and collision strengths following the work of Hagelstein and Jung [85]. The collision strengths are closely related to the electron–electron interaction amplitudes in Section 2.3 but here includes a *free* electron on both sides, in the initial and final states of the transition matrix. Although the standard decomposition into angular coefficients and radial integrals can be applied, and thus allows to include both the Coulomb and Breit interaction into the collision strength, special care has to be taken in evaluating the radial parts of these amplitudes. In the EIMEX component, all required continuum orbitals are generated “on fly” in course of the computations.
- HFS evaluates the diagonal and nondiagonal hyperfine amplitudes (30) from Section 2.3 between atomic levels of the same parity and for some given total spin and moments of the nucleus. These amplitudes can further be utilized to calculate the hyperfine levels (28) by diagonalizing the Hamiltonian $H = H_{\text{DFB}} + H_{\text{hfs}}$ by means of the TOOLBOX component or to calculate hyperfine-quenched transition rates and lifetimes of isotopes with non-zero nuclear spin. In the latter case, the transition amplitudes need to be calculated *before* with the EINSTEIN or REOS programs. Moreover, the hyperfine amplitudes from HFS are currently applied also to explore the hyperfine splitting of the (low-lying) dielectronic resonances in the nonradiative electron capture of highly-charged ions.
- ISOTOPE SHIFT parameters can be calculated by some (not yet distributed) task of the TOOLBOX. For two given isotopes with mass numbers $A > A'$, the isotope shift can be parametrized as

$$\delta\nu^{A,A'} = (M^{\text{nms}} + M^{\text{sms}}) \frac{A - A'}{AA'} + F\delta(r^2)^{A,A'} \quad (45)$$

where M^{nms} and M^{sms} refer to the normal and specific mass shifts, respectively, and where F is known as the field-shift coefficient. From the individual transition frequencies, which need to be calculated separately for three or more isotopes, the mass-shift M and field-shift parameters F are obtained by solving Eq. (45) for different triples of isotopes. Here, all level or transition amplitude calculations are supposed to have been performed before by either the RELCI component or Rci92 from GRASP92; however, no attempt

has been made yet to incorporate the relativistic recoil corrections to the specific mass shift operator [86]. While formula (45) assumes that all higher powers in the charge distribution, $\delta\langle r^4 \rangle^{A,A'}, \delta\langle r^6 \rangle^{A,A'}, \dots$ can be captured simply by means of a single (field-shift) parameter F [87], this procedure enables one to deal with all isotope effects within the same computational framework. For several medium and heavy elements, in fact, laser-spectroscopic measurements of the isotope shifts have been carried out for different chains (of isotopes) in order to derive useful information about the nuclear spins and charge radii of different (radioactive) isotopes.

- **MULTIPHOTON** has been designed as new component for estimating second-order two-photon transition amplitudes. Again, such calculations include the summation (integration) over a – more or less complete – set of intermediate states as generated, for example, by means of the **CONCI** component. In the **MULTIPHOTON** component, the multipole decomposition of the photon field is utilized in order to explore the effects of the non-electric dipole contributions (other than $E1$) to the two-photon excitation and decay rates [77]. If appropriate, this approach is planned to be extended to various three- and multi-photon transition amplitudes in order to investigate various direct multi-photon ionization processes.
- **RESONANCE** is developed to evaluate the (resonant) photoionization cross sections and angular parameters across one or a few autoionizing resonances. This component utilizes the transition amplitudes files from the **PHOTO** and **AUGER** components and combines these amplitudes to compute the shape of individual resonances and to determine their Fano parameters [88].
- **SCHIFF** helps calculate various parity- and time-reversal violating amplitudes and enhancement factors that gives rise to atomic electric-dipole moments in second-order perturbation theory. These amplitudes include the (P, T) -odd interactions due to nuclear Schiff moment as well as various (P, T) -odd electron–nucleon interactions. These interactions usually depend on the nuclear spin and display strong singularities near to the nucleus [89]. The electronic enhancement factors due to these interactions are important for several on-going experiments on diamagnetic atoms, such as Xe, Hg and Ra [90].
- **SPECTRUM** computes a complete but discretized one-particle spectrum for either a bare nucleus or predefined (spherical) mean-field potential. This mean-field potential is derived from the occupation numbers of an atomic states, as either the direct part of the Dirac–Fock or some x -alpha potential [91]. The Dirac equation is solved for this potential and for each symmetry κ (the one-electron total angular momentum and parity) in a spherical box by using B-splines; these solutions are afterwards interpolated into the standard grids of **GRASP92** and **RATIP**. However, since all orbitals are normalized to *one*, like the bound-state orbitals, these orbital functions represent (spherical) wave packets with a certain energy distribution rather than electrons with well-defined energy. The one-electron orbitals of the **SPECTRUM** component are generated mainly for use by the **CONCI** program in order to construct approximate continuum states and for an (approximate) integration over the continuum.

Many of these recent developments mentioned here have been made in response to some given experiment or the requirement of users, and hence have been developed first of all with a particular atom or ion in mind. Therefore, these components need to be further generalized in order to make them applicable to a larger range of atoms and shell structures as it is typical for most components of the **RATIP** program.

As seen from the list of “new” components above, much of the present interest in studying the excitation and ionization dynamics of atoms focus on second- and higher-order processes that formally include a summation of transition amplitudes over the (complete) spectrum of the system. Apart from the discrete bound states, this generally requires an integration over the continuum part of the many-electron spectrum. Obviously, this is a highly non-trivial task, especially for multi-electron systems, since it requires to generate the continuum and to evaluate the *free–free* transition amplitudes. Examples of such second-order processes are the (single-photon) double ionization of atoms, the two-photon excitation and decay, sequential two-photon processes, double Auger processes, and several others. Any implementation of these physical properties requires first of all a good “physical insight” into the relevant part of the many-electron spectrum in order to reduce it to a finite and well feasible set of intermediate states. In **RATIP**, we will continue to developing the **SPECTRUM** and **CONCI** components in order to generate an intermediate set of atomic states *independently and before* these states are utilized to compute transition amplitudes. I hope this will make **RATIP** ready for a new generation of experiments that are performed at present or in the near future at free-electron lasers or some other high-intense light sources.

Acknowledgements

In developing the **RATIP** program, I have benefited a lot over the years from the discussion, comments and contributions of many collaborators and friends. In particular, I wish to acknowledge the helpful advices and contributions of Josef Anton, Jacek Bieron, Chenzhong Dong, Burkhard Fricke, Charlotte Froese Fischer, Gediminas Gaigalas, Ian P. Grant, Alexei Grum-Grzhimailo, Sami Heinäsmäki, Paul Indelicato, Per Jönsson, Nikolay M. Kabachnik, Yong-ki Kim, Fumihiko Koike, Peter Koval, Biswajit Saha, Wolf-Dieter Sepp, Joseph E. Sienkiewicz and Andrey Surzhykov.

References

- [1] T.R. Kallman, P. Palmeri, *Rev. Mod. Phys.* 79 (2007) 79;
H.P. Summers, et al., *Plasma Phys. Control. Fusion* 44 (2002) B323;
J.S. Sims, et al., *NIST Journal of Research* 1130 (2008) 131.
- [2] I.P. Grant, *Relativistic Quantum Theory of Atoms and Molecules: Theory and Computation*, Springer, 2007.
- [3] K. Blum, *Density Matrix Theory and Applications*, Plenum Press, New York and London, 1996.
- [4] V.V. Balashov, A.N. Grum-Grzhimailo, N.M. Kabachnik, *Polarization and Correlation Phenomena in Atomic Collisions*, Kluwer Academic Plenum Publishers, New York, 2000.
- [5] S. Fritzsche, *J. Electron Spectrosc. Relat. Phenom.* 114–116 (2001) 1155.
- [6] F.A. Parpia, C.F. Fischer, I.P. Grant, *Comput. Phys. Comm.* 94 (1996) 249.
- [7] W.R. Johnson, *Atomic Structure Theory: Lectures on Atomic Physics*, Springer, 2007.
- [8] I.P. Grant, *Relativistic effects in atoms and molecules*, in: S. Wilson (Ed.), *Methods in Computational Chemistry*, vol. 2, Plenum, New York, 1988, p. 1.
- [9] K.G. Dyall, et al., *Comput. Phys. Comm.* 55 (1989) 425.
- [10] P. Jönsson, et al., *Comput. Phys. Comm.* 177 (2007) 597.
- [11] https://dirac.spectro.jussieu.fr/mcdf/mcdf_code/mcdfgme_status.html.
- [12] J. Sucher, *Rep. Progr. Phys.* 41 (1978) 1781.

- [13] I. Lindgren, *Relativistic Many-Body Theory*, Springer Series on Atomic, Optical, and Plasma Physics, vol. 63, Springer, New York, 2011.
- [14] J. Sapirstein, *Phys. Scripta* 36 (1987) 801.
- [15] G. Breit, *Phys. Rev.* 39 (1932) 616.
- [16] Z. Cai, V. Meiser Umar, C. Froese Fischer, *Phys. Rev. Lett.* 68 (1992) 297.
- [17] S. Fritzsche, *Phys. Scripta T* 100 (2002) 37;
C.Z. Dong, S. Fritzsche, *Phys. Rev. A* 72 (2005) 012507.
- [18] B. Engels, *Theor. Chim. Acta* 86 (1993) 429.
- [19] J. Bieron, et al., *Phys. Rev. A* 60 (1999) 3547;
J. Bieron, et al., *Phys. Rev. A* 79 (2009) 052502;
J. Bieron, et al., *J. Phys. B* 37 (2004) L305.
- [20] I.P. Grant, *Comput. Phys. Comm.* 5 (1973) 263.
- [21] G. Gaigalas, S. Fritzsche, I.P. Grant, *Comput. Phys. Comm.* 139 (2001) 263.
- [22] M. Rose, *Elementary Theory of Angular Momentum*, Wiley, New York, 1957.
- [23] K. Ueda, et al., *J. Phys. B* 34 (2001) 107;
M. Kitajima, et al., *J. Phys. B* 34 (2001) 3829;
M. Kitajima, et al., *J. Phys. B* 35 (2002) 3327.
- [24] E.W. Schmidt, et al., *Phys. Rev. A* 76 (2007) 032717.
- [25] S. Fritzsche, A. Surzhykov, T. Stöhlker, *Phys. Rev. Lett.* 103 (2009) 113001.
- [26] W.R. Johnson, in: D. Liesen (Ed.), *Physics with Multiply Charged Ions*, in: NATO ASI Series, vol. 348, Plenum Press, New York, London, 1995, p. 1.
- [27] U. Fano, G. Racah, *Irreducible Tensorial Sets*, Academic Press, New York, 1959.
- [28] S. Fritzsche, A. Surzhykov, T. Stöhlker, *Phys. Rev. A* 72 (2005) 012704.
- [29] N.M. Kabachnik, et al., *Phys. Rep.* 451 (2007) 155.
- [30] I. Lindgren, A. Rosen, *Case Stud. At. Phys.* 4 (1974) 97.
- [31] W.R. Johnson, *Can. J. Phys.* 99 (2010) 1.
- [32] P. Jönsson, F.A. Parpia, C. Froese Fischer, *Comput. Phys. Comm.* 96 (1996) 239.
- [33] M. Andersson, P. Jönsson, *Comput. Phys. Comm.* 178 (2008) 156.
- [34] J. Grumer, M. Andersson, T. Brage, *J. Phys. B* 43 (2010) 074012.
- [35] S. Fritzsche, C. Froese Fischer, G. Gaigalas, *Comput. Phys. Comm.* 148 (2002) 103.
- [36] S. Fritzsche, C. Froese Fischer, *Comput. Phys. Comm.* 99 (1997) 323;
S. Fritzsche, C. Froese Fischer, *Comput. Phys. Comm.* 124 (2000) 340.
- [37] G. Gaigalas, S. Fritzsche, *Comput. Phys. Comm.* 134 (2001) 86.
- [38] S. Fritzsche, I.P. Grant, *Comput. Phys. Comm.* 92 (1995) 111;
S. Fritzsche, I.P. Grant, *Comput. Phys. Comm.* 103 (1997) 277.
- [39] M. Metcalf, J. Reid, *Fortran 90/95 Explained*, Oxford University Press, 1996.
- [40] S. Fritzsche, B. Fricke, W.-D. Sepp, *Phys. Rev. A* 45 (1992) 1465.
- [41] T. Åberg, G. Howat, in: W. Mehlhorn (Ed.), *Corpuscles and Radiation in Matter I*, in: *Encyclopedia of Physics*, vol. XXXI, Springer, Berlin, 1982, p. 469.
- [42] S. Fritzsche, et al., *Phys. Rev. A* 44 (1991) 388.
- [43] Y. Shimizu, et al., *J. Phys. B* 33 (2000) L685;
K. Ueda, et al., *Phys. Rev. Lett.* 90 (2003) 153005.
- [44] S. Fritzsche, *Phys. Lett. A* 180 (1993) 262;
A. De Fanis, et al., *Phys. Rev. A* 70 (2004) 040702.
- [45] S. Fritzsche, *Comput. Phys. Comm.* 103 (1997) 51;
S. Fritzsche, *Comput. Phys. Comm.* 111 (1998) 167;
S. Fritzsche, *Comput. Phys. Comm.* 139 (2001) 314.
- [46] A. Surzhykov, P. Koval, S. Fritzsche, *Comput. Phys. Comm.* 165 (2005) 139.
- [47] S. Fritzsche, *Int. J. Quant. Chem.* 106 (2006) 98.
- [48] P.O. Löwdin, *Phys. Rev.* 97 (1955) 1474.
- [49] J. Olsen, et al., *Phys. Rev. E* 52 (1995) 4499.
- [50] D. Strohschein, et al., *Phys. Rev. A* 77 (2008) 022706;
D. Röhrbein, T. Kirchner, S. Fritzsche, *Phys. Rev. A* 81 (2010) 042701.
- [51] C.Z. Dong, et al., *J. Phys. B* 39 (2006) 3121.
- [52] I.P. Grant, *J. Phys. B* 7 (1974) 1458.
- [53] C. Eichhorn, et al., *Phys. Rev. E* 80 (2009) 026401.
- [54] T. Radtke, A. Surzhykov, S. Fritzsche, *Phys. Rev. A* 77 (2008) 022507.
- [55] S. Fritzsche, et al., *Phys. Rev. A* 75 (2007) 012501;
S. Fritzsche, et al., *Phys. Rev. A* 78 (2008) 032514.
- [56] F. Da Pieve, *J. Phys. B* 38 (2005) 3619;
F. Da Pieve, *J. Phys. B* 40 (2007) 329.
- [57] J. Niskanen, et al., *Phys. Rev. A* 81 (2010) 013406.
- [58] S. Fritzsche, et al., *J. Phys. B* 41 (2008) 165601;
S. Fritzsche, et al., *J. Phys. B* 42 (2009) 145602;
S. Fritzsche, et al., *J. Phys. B* 44 (2011) 175602.
- [59] M. Kurka, et al., *J. Phys. B* 42 (2009) 141002.
- [60] T. Stöhlker, et al., *Phys. Scripta T* 110 (2004) 384.
- [61] A. Surzhykov, et al., *Phys. Rev. Lett.* 94 (2005) 203202;
A. Surzhykov, et al., *Phys. Rev. A* 73 (2006) 032716.
- [62] M.K. Inal, A. Surzhykov, S. Fritzsche, *Phys. Rev. A* 72 (2005) 042720;
L. Sharma, et al., *Phys. Rev. A* 81 (2010) 023419.
- [63] Y.-K. Kim, private communication, 1999.
- [64] P.J. Mohr, *Ann. Phys. (N.Y.)* 88 (1974) 52;
P.J. Mohr, *At. Data Nucl. Data Tables* 29 (1983) 453.
- [65] P.J. Mohr, Y.-K. Kim, *Phys. Rev. A* 45 (1992) 2723.
- [66] S. Fritzsche, et al., *Eur. Phys. J. D* 45 (2007) 107;
S. Fritzsche, et al., *Eur. Phys. J. D* 33 (2005) 15.
- [67] A. Stathopoulos, C.F. Fischer, *Comput. Phys. Comm.* 79 (1994) 268.
- [68] M. Sewtz, et al., *Phys. Rev. Lett.* 90 (2003) 163002.
- [69] F.C. Charlwood, et al., *Phys. Lett. B* 690 (2010) 346;
T.E. Cocolios, et al., *Phys. Rev. Lett.* 106 (2011) 052503.
- [70] M. Avgoulea, et al., *J. Phys. G* 38 (2011) 025104.
- [71] S. Fritzsche, I.P. Grant, *Phys. Lett. A* 186 (1994) 152;
S. Fritzsche, I.P. Grant, *Phys. Scripta* 50 (1994) 473.

- [72] B. Saha, S. Fritzsche, J. Phys. B 38 (2005) 1161;
B. Saha, S. Fritzsche, Phys. Rev. E 73 (2006) 036405.
- [73] C. Kohstall, et al., At. Data Nucl. Data Tables 70 (1998) 63;
S. Fritzsche, C. Froese Fischer, B. Fricke, At. Data Nucl. Data Tables 68 (1998) 149.
- [74] C.Z. Dong, et al., Mon. Not. R. Astron. Soc. 307 (1999) 809;
S. Fritzsche, et al., Mon. Not. R. Astron. Soc. 318 (2000) 263.
- [75] S. Fritzsche, C.Z. Dong, G. Gaigalas, At. Data Nucl. Data Tables 76 (2000) 155.
- [76] J.K. Saha, et al., Phys. Lett. A 373 (2009) 252.
- [77] A. Surzhykov, et al., Phys. Rev. A 71 (2005) 022509;
A. Surzhykov, et al., Phys. Rev. A 84 (2011) 022511.
- [78] S. Trotsenko, et al., Phys. Rev. Lett. 104 (2010) 033001.
- [79] S. Fritzsche, Comput. Phys. Comm. 141 (2001) 163;
J. Nikkinen, S. Fritzsche, S. Heinäsmäki, Comput. Phys. Comm. 175 (2006) 348.
- [80] G. Gaigalas, T. Zalandauskas, S. Fritzsche, Comput. Phys. Comm. 157 (2004) 239.
- [81] X. Ma, et al., Phys. Rev. A 64 (2001) 012704;
S. Fritzsche, T. Stöhlker, Phys. Scripta T 92 (2001) 311.
- [82] P. Linusson, et al., Phys. Rev. A 83 (2011) 023424.
- [83] A. Surzhykov, et al., Phys. Rev. A 77 (2008) 042722.
- [84] S. Fritzsche, P. Indelicato, T. Stöhlker, J. Phys. B 38 (2005) S707.
- [85] P.L. Hagelstein, K.R. Jung, At. Data Nucl. Data Tables 37 (1987) 121.
- [86] E. Gaidamauskas, et al., J. Phys. B 44 (2011) 175003.
- [87] G. Fricke, et al., At. Data Nucl. Data Tables 60 (1995) 177.
- [88] A.N. Grum-Grzhimailo, et al., J. Phys. B 38 (2005) 2545.
- [89] V.A. Dzuba, V.V. Flambaum, S.G. Porsev, Phys. Rev. A 80 (2009) 032120.
- [90] W.C. Griffith, et al., Phys. Rev. Lett. 102 (2009) 101601.
- [91] R. Gaspar, A. Nagy, J. Phys. B 19 (1986) 2793.