

Application of Symmetry-Adapted Atomic Amplitudes

Stephan Fritzsche ^{1,2,3} ¹ Helmholtz-Institut Jena, Fröbelstieg 3, D-07743 Jena, Germany; s.fritzsche@gsi.de² GSI Helmholtzzentrum für Schwerionenforschung, D-64291 Darmstadt, Germany³ Theoretisch-Physikalisches Institut, Friedrich-Schiller-Universität Jena, D-07743 Jena, Germany

Abstract: Following the work of Giulio Racah and others from the 1940s onward, the rotational symmetry of atoms and ions, e.g., the conservation of angular momentum, has been utilized in order to efficiently predict atomic behavior, from their level structure to the interaction with external fields, and up to the angular distribution and polarization of either emitted or scattered photons and electrons, while this rotational symmetry becomes apparent first of all in the block-diagonal structure of the Hamiltonian matrix, it also suggests a straight and consequent use of symmetry-adapted interaction amplitudes in expressing the observables of most atomic properties and processes. We here emphasize and discuss how atomic structure theory benefits from exploiting this symmetry, especially if open-shell atoms and ions in different charge states need to be combined with electrons in the continuum. By making use of symmetry-adapted amplitudes, a large number of excitation, ionization, recombination or even cascade processes can be formulated rather independently of the atomic shell structure and in a language close to the formal theory. The consequent use of these amplitudes in existing codes such as GRASP will therefore qualify them to deal with the recently emerging demands for developing general-purpose tools for atomic computations.

Keywords: approximate atomic Green function; atom; atomic cascade; electron capture and emission; electron continuum; language for atomic computations; many-electron interaction amplitude; Racah's algebra; relativistic; rotational symmetry



Citation: Fritzsche, S. Application of Symmetry-Adapted Atomic Amplitudes. *Atoms* **2022**, *10*, 127.
<https://doi.org/10.3390/atoms10040127>

Academic Editors: Jacek Bieroń, Charlotte Froese Fischer and Per Jönsson

Received: 1 September 2022

Accepted: 23 October 2022

Published: 1 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Electronic structure calculations of atoms and ions have a long tradition in physics with many applications in basic research, spectroscopy and at various places elsewhere. Soon after the self-consistent Hartree–Fock (HF) theory was developed, Swirls [1] formulated a first set of *relativistic* self-consistent field (SCF) equations including exchange, i.e., the antisymmetry of the many-electron wave functions. These equations were built on the Dirac–Coulomb Hamiltonian and are currently referred to as Dirac–Hartree–Fock (DHF) equations. Using Racah's algebra, these coupled equations were re-examined and brought into a compact form by Grant in 1961 [2] and, since these days, form the foundation for almost all (relativistic) atomic structure codes which are based upon the variational principle.

Indeed, the role of Einstein's special relativity upon the electronic structure of atoms and ions became quickly evident when Grant explored and visualized the shape of hydrogenic wave functions for multiplied and highly charged ions [3]. Since then, various numerical solutions to the DHF equations have been developed and were utilized as starting point for both the multiconfiguration DHF (MCDHF) and the relativistic many-body perturbation theory (RMBPT). These developments also led to the first few relativistic general-purpose codes [4,5], from which the General-purpose Relativistic Atomic Structure Program (GRASP) is widely used today [6,7]. All these codes are explicitly based on Racah's algebra that combines the rotational symmetry of open-shell atoms with the re-coupling of angular momenta and the calculus of irreducible tensor operators [8–10], though perhaps with different goals and focus in their implementation [11–13].

Today, different applications of (relativistic) atomic structure theory come with quite distinct requirements with regard to the accuracy and complexity of the computations as well as to the shell structure of the atoms and ions of interest, and while the need for good atomic data can hardly be overestimated, the present-day demands call for a descriptive language of atomic computations and a re-organization of the codes in order to make optimal use of atomic theory. We shall explain and discuss below how the consequent use of symmetry-adapted amplitudes in general-purpose codes will help expand atomic computations towards second-order in perturbation theory and continuum processes, atomic cascades or even for alternative representations of the atomic states. In particular, we shall formulate a few guidelines how these amplitudes can simplify the communication with and within a program and help overcome present-day limitations. These guidelines (may) also facilitate to make GRASP and other structure codes ready for new applications of atomic theory.

This work is structured as follows. In Section 2, we first discuss the use and impact of symmetry-adapted basis states upon the implementation of general-purpose (relativistic) codes and especially upon GRASP. Apart from the indisputable successes and capabilities of this code, we shall explain several of its limitations that come along with present-day demands on atomic theory. In Section 3, we then argue how an explicit and consequent use of symmetry-adapted amplitudes paves the way towards recently emerging applications of atomic theory and may help establish a language for describing all necessary computations. Apart from the role of proper data types and various features from contemporary programming, we here briefly discuss JAC, the Jena Atomic Calculator [14], in which such a language has been partly realized, although with emphasis on ionization and recombination processes. Finally, a summary and a few conclusions are drawn in Section 4.

2. Impact of Symmetry-Adapted Basis States

2.1. First Steps towards a General-Purpose Code

Ian Grant's reformulation of the relativistic self-consistent theory in the 1960s, and especially his review from 1970 [15], set the scene for numerical solutions and, eventually, for a general-purpose MCDHF code, which became available from the Computer Physics Communications Program Library in 1980 [5,16]. This code made use of the calculus of irreducible tensor operators and Racah's algebra for (re-)coupling the angular momenta [8,17]. If combined with symmetry-adapted and antisymmetric (sub-)shell states, a concept originally advanced in nuclear theory [18,19], this approach led to a powerful machinery that helped predict the level energies and approximate atomic state functions (ASF) for atoms and ions with a complex shell structure [10,20]. These developments also stimulated massive relativistic calculations of transition probabilities [21,22], hyperfine structures [23,24] as well as (auto-)ionization [25,26] and scattering processes [27] for open-shell atoms and ions, because they all occur very frequently in nature.

While not much needs to be said about the history of relativistic structure theory, it is quite obvious today that the details and accuracy of any atomic computation critically depend on the representation of the ASF and the explicit definition of the many-electron basis. Like in the symmetry-adapted nonrelativistic HF approximation, an ASF is written within the MCDHF method as superposition of configuration state functions (CSF) with well-defined parity P , total angular momentum J and projection M [28],

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

and where γ_r refers to all additional quantum numbers that are needed in order to specify the (N -electron) CSF uniquely. In most standard computations the set $\{|\gamma_r PJM\rangle, r = 1, \dots, n_c\}$ of CSF are constructed as antisymmetrized products of a common set of *orthonormal* (one-electron) orbitals for a given atomic state or for a selected set of ASF. In the expansion (1), moreover, the notation $\mathbb{J} \equiv J^P$ has been introduced to just specify below

the overall symmetry of any level by its total angular momentum and parity. Together with a proper choice of the (relativistic) Hamiltonian, such a symmetry-adapted representation of the many-electron basis states also make atomic computations feasible well beyond the lowest excited levels of an atom and, hence, for a general use in atomic physics.

With the symmetry-adapted basis from above and a proper decomposition of all atomic interactions into spherical tensor operators, the (reduced) matrix elements of the Hamiltonian and almost all interaction operators can be written for symmetric one- and two-particle operators always as

$$T^{(K)} = \sum_j f^{(K)}(j; \dots) \quad \left\langle \gamma_r \mathbb{J}_r \parallel T^{(K)} \parallel \gamma_s \mathbb{J}_s \right\rangle = \sum_t u(a_t b_t; K) X(a_t b_t; K), \quad (2)$$

$$T^{(K)} = \sum_{i < j} g^{(K)}(i, j; \dots) \quad \left\langle \gamma_r \mathbb{J}_r \parallel T^{(K)} \parallel \gamma_s \mathbb{J}_s \right\rangle = \sum_t v(L_t; a_t b_t c_t d_t; K) X^{L_t}(a_t b_t c_t d_t; K), \quad (3)$$

and where the $X(a_t b_t; K)$ and $X^{L_t}(a_t b_t c_t d_t; K)$ denote the one- and two-particle interaction strengths, respectively. In this notation, the ... refer to the parametric (or model) dependence of the interaction operators and L_t to any internal rank in (re-)coupling the one-electron angular momenta, while the effective interaction strengths are specific to the particular interaction or transition operator of a given property or process, the (so-called) *angular coefficients* $u(a_t b_t; K)$ and $v(L_t; a_t b_t c_t d_t; K)$ only depend on the overall rank K of the operator and the shell structure of the two CSF involved, e.g., the shell occupation and the coupling of their associated subshell states. Various library functions have been developed and improved over the years [29–31] in order to calculate these angular integrals for any pair of CSF and for interaction operators of different rank. It is this—formally—rather simple decomposition (2) and (3) of all reduced matrix elements that makes the MCDHF or any of the associated configuration interaction codes with a symmetry-adapted basis often superior and much simpler to expand when compared to implementing most other methods. Of course, this advantage applies first of all also to GRASP and has given rise to a good number of benchmark results [32–34].

2.2. The GRASP Code: Successes, Capabilities and Limitations

The earlier developments of GRASP [5] have been pushed forward especially by Parpia and coworkers [6] by providing a modern implementation (at that time) of the SCF field and by supporting rather systematic enlargements of the CSF and orbital bases. Since these orbitals satisfy a coupled set of integro-differential equations, one for each participating subshell $n \kappa$, they are often solved iteratively and by keeping the other orbitals fixed. A systematic iteration scheme then varies the potentials and provides in turn updates of the orbital functions, until the successive changes are negligible for the given application. In such a finite-difference iteration scheme, the positron orbitals never occur explicitly in the implementation, although they are affected by the variational procedure [10].

The advancement and general-purpose design of GRASP later helped expand this code towards the generation of continuum orbitals [35], the electron-impact ionization with distorted waves [36] and the computation of various excitation and ionization properties [37], the incorporation of simple plasma models [38], and up to a relativistic version of the R-matrix method [39]. It also helped incorporate the rearrangement of the electron density in course of particle emission and absorption processes, and which typically results into sets of not quite orthogonal orbital functions for the representation of the initial and final states. In practice, this *non-orthogonality* can be accounted for in the evaluation of many-electron matrix elements (2)–(3) by either implementing a biorthonormal transformation of the two orbital sets [40] or by an expansion of the atomic states into Slater determinants [41].

Sure, the success of GRASP and its large number of applications are unquestionable. During the last two decades, much emphasis in developing these tools was placed especially upon the systematic improvement of the level energies and eigenvectors for quite general

open-shell atoms. In particular, the convergence of energies and transition rates have been improved and monitored in good detail [20]. By clearly distinguishing between the spectroscopically occupied and (so-called) correlation orbitals, the electron density near to the nucleus can often be refined considerably, which is in contrast to what is feasible with node counting and the orthogonality of the spectroscopic orbitals alone. Moreover, a parallelized computation of the angular coefficients and the Hamiltonian matrix now facilitates the use of much larger restricted-active spaces (RAS) in the ab initio representation of individual ASF. All these upgrades have sizeably enlarged the domain of the atomic computations and enables one today to provide, at least for selected properties, predictions with spectroscopic accuracy [42–44]. When compared with other advanced many-electron techniques [45], such as RMBPT or the coupled-cluster approach, the MCDHF method has the advantage that it can be quite readily applied also to (highly) excited and open-shell structures across the whole periodic table.

The present GRASP code [7] is an updated Fortran-95 version, based on blocks of CSF with well-defined symmetry $\mathbb{J} \equiv J^P$, and where the rearrangement of the electron density is partly kept for levels with different symmetry. Moreover, a number of message-passing-interface (MPI) programs were designed to execute all major parts in the calculation of SCF by using parallel computers. These developments have moved GRASP towards a more object-oriented design, while the data types and encapsulation as well as the hierarchy and dynamic use of code have not (yet) changed a lot. However, several tools have been added that now simplify the generation of CSF lists for using, for instance, multireference computational models with single and double excitations [46–48].

To reduce the computational effort, the decomposition of most (reduced) matrix elements is made, and often stored, in GRASP separately from their subsequent application. Indeed, this prior decomposition of the many-electron matrix elements requires to control and perform a good number of steps, usually by using or modifying available scripts. For the calculation of level energies, for example, one first needs to generate a CSF basis, a set of starting orbitals, the angular coefficients for the given basis and inter-electronic interaction as well as to run the SCF iteration, and all this before the energies and wave functions $\{E_\alpha, c_r(\alpha)\}$ can be finally obtained by setting-up and diagonalizing the Hamiltonian matrix. In practice, these steps often obscure the modeling of the properties and behavior that the user may have in mind, and this limitation applies quite regularly, if free electrons are to be involved into the computations. Since many processes are associated with two or more atomic bound states (levels), more often than not, different charge states of the atom need to be generated. Because of these individual steps also, which need to be carried out in rather fixed sequence, little effort has been spent so far to deal with the electron continuum in GRASP, not to mention the interplay or perhaps the modeling of a whole “chain” of elementary processes. In contrast to the bound orbitals, which are generated self-consistently, the continuum orbitals are typically solved within a static potential of the associated ionic core as well as for fixed energies, as obtained from energy-conservation arguments or the integration over the electron continuum.

The present version of GRASP has been found to be limited also by its *static* design and the quite technical language in its implementation, as perhaps typical for most Fortran and C codes. This basic design has hampered to extend the code towards new applications or the use of graphical interfaces as we shall described below. This design often hinders a transparent communication *with* and *within* the code that should be independent of the shell structure of the atoms or any particular application. From a physics viewpoint, furthermore, energies and wave functions usually just occur as an intermediate outcome in order to express cross sections, rates, rate coefficients or, more generally, to even synthesize spectra that are directly comparable with experiment. These practical demands also suggest that we deal with only those sub-spaces of the many-electron Hilbert space that need to be taken into account for a particular application, whereas the selection and construction of these sub-spaces requires physical understanding of the underlying atomic processes and much of this computational work should be automated and supported by a deliberate

language design in order to eventually apply the most specialized code for a given task. To make these limitations of GRASP perhaps even more explicit, let us next summarize a few recently emerging demands to atomic theory.

2.3. Recent Demands on Relativistic Atomic Theory

During the past decades, the demands to atomic structure and collision theory have greatly been enlarged from the accurate computation of a few low-lying level energies and properties towards quite massive applications in astro, plasma and technical physics, and at various places elsewhere. The recent advancements in developing new light sources, the research with free-electron lasers (FEL) at different wavelengths or the triumph with atomic clocks are just a few examples, where further progress will critically depend on atomic theory and our abilities to model atomic behavior. Obviously, however, these fields require atomic computations of quite different size and complexity, from simple estimates, to a fast modeling of atomic processes, to highly correlated computations of selected frequency shifts or cross sections, and sometimes even to generating rather large sets of atomic data. For one or a few inner-shell holes in any many-electron configuration, for example, the multiple release of Auger electrons quickly leads to very complex shell structures with tens or more fine-structure levels for each single configuration.

Clearly, this complexity of many atomic computations requires a concise language in order to readily distinguish and specify different models, to combine different steps in the computation and to eventually keep track and utilize all the generated data, while most atomic behavior can be naturally formulated by means of many-electron amplitudes (matrix elements), their explicit combination is often sophisticated and requires additional knowledge from collision and density matrix theory or about the electron dynamics in strong laser fields. In practice, therefore, these distinct requirements from different fields gave rise to developing specialized code, but which are often less available to the community. Or, if seen the other way around, these demands and their typical complexity may just illustrate that GRASP and all presently available codes still need further developments to make them ready for future use.

From the viewpoint of recent case studies, in particular the following demands to atomic theory have emerged and should be likely considered by general-purpose codes:

- Since most atomic behavior is qualitatively well understood (and is often readily distinguishable from each other) within the atomic shell model and its underlying electron configurations, this “shell-model view” need to be recognized and, possibly, re-adjusted by the user during various steps of (complex) computations. This need arises especially for processes with free electrons in either the initial and/or final (scattering) state of the atoms. Apart from the set of reference configurations and from specifying the virtual excitation in the construction of restricted active spaces (RAS), the set-up of atomic computations should therefore enable one to readily reconstruct the underlying shell occupation at quite different level of the computations. The reconstruction and analysis of the important shells during the modeling then help determine and adapt the subspaces that are (to be) taken into account into the quantum representation of the atomic states. In fact, the simple control of the computations by means of selecting and discarding (groups of) electron configurations can hardly be overrated for atoms with multiple inner-shell holes.
- A more flexible treatment of the electron continuum will enable one to model resonances embedded into the continuum, as well as the ionization, recombination and decay dynamics of free atoms and ions. It will help also to incorporate the continuum (interchannel) interactions in the construction of scattering states [49] or to construct approximate atomic Green functions [50], cf. Section 3.1.
- The increasing complexity of atomic computations suggests to perform many, if not most, of its steps automatically, based on a set of well-chosen *default* parameters. Apart from the incorporation of relativistic corrections, this refers not only to the generation of the (mean-field) SCF but also to calculating the symmetry-adapted

amplitudes (2)–(3) themselves, if the computational model and approximations can be expressed concisely enough. The simple use and overwriting of default values also determines the rate with which complex computations can be realized.

- Quite different observations are typically recorded by different spectroscopic communities owing to the—individual or coincident—measurement of photons, electrons and/or ions. The large number of experimental setups and scenarios then require a proper classification of atomic properties and processes as well as a language, close to the underlying physics, in order to avoid duplication and inhomogeneity of code, or the implementation of overly specific features.
- Recent advancements in short- and strong-field physics make it possible today to explore the electron dynamics of atoms under extreme conditions, and which clearly differ from standard (spectroscopic) measurements. Apart from a remarkable increase in the intensity of light pulses by six orders of magnitude or even more, the time structure of these pulses can be currently controlled down to a few cycles and help studying non-linear photoprocesses [51]. Moreover, the interaction of such pulses with atoms and ions often enforces (relativistic) theory to deal with time- and spatially-structured vector potentials $\mathbf{A}(\mathbf{r}, t)$, well beyond the typical electric-dipole approximation or the incorporation of additional multipoles. Still, these interactions can be expressed by symmetry-adapted many-electron amplitudes [52] and should therefore be handled by modern codes.

All these contemporary demands to atomic theory, and likely several more, underline the need for establishing a language that can be adapted to different communities and which can be shaped (and learned by the users) gradually. Moreover, since many researchers perform atomic computations just as part of their—theoretical or experimental—daily work, a simple communication and control of the program is mandatory for any general-purpose code.

3. Access to and Application of Symmetry-Adapted Amplitudes

3.1. Towards Current Fields of Research and Applications

The reduced many-electron amplitudes (2)–(3) are central to atomic theory, from the setup and diagonalization of the Hamiltonian matrix up to a large class of transition properties that just occur from first- or higher-order perturbation theory onwards. These amplitudes characterize individual interactions among the electrons or with external particles and fields and, altogether, determine the atomic behavior. These many-electron amplitudes generally combine two atomic (bound) states of either the same or two different charge states, and often also include free electrons in the continuum. The consequent use of these amplitudes in a given frame therefore enables us to formulate (and implement) all atomic computations in a language that remains close to the formal theory. Despite of this common knowledge, however, codes are still based on a quick and often ad hoc chosen decomposition of these many-electron amplitudes into one- and two-particle reduced matrix elements, or even into radial integrals, well before any implementation or coding is conducted. Such a decomposition then dismembers the quantum-mechanical expressions and make their use difficult in other, though perhaps closely related, tasks.

In practice, there are just six types of these many-electron amplitudes which frequently occur in computations and which refer to different (inner-)atomic interactions. Apart from the well-known electron–electron and electron–photon interaction amplitudes, the (hyperfine) interaction with the nuclear moments [43,53], the mass- and field-shift amplitudes [54,55], or the Coulomb excitation by charged particles [56] determine most spectroscopic observations. Indeed, all these reduced amplitudes can be written in the compact form (2)–(3).

Electron–electron interaction amplitude: The amplitudes $\langle \alpha \mathbb{J} \| V \| \beta \mathbb{J}' \rangle = \langle \alpha \mathbb{J} \| V \| \beta \mathbb{J} \rangle \delta_{\mathbb{J}\mathbb{J}'}$ often refer to the scalar interaction operator

$$V \equiv V^{(\text{Coulomb})} + V^{(\text{Breit})} = \sum_{i < j} \left(\frac{1}{r_{ij}} + b_{ij} \right), \tag{4}$$

as it occurs in the Dirac–Coulomb–Breit Hamiltonian $H_{DCB} = \sum_i h_D(\mathbf{r}_i) + V$. These amplitudes are of course frequently needed for the computation of the level structure itself, but also for (Auger) electron emission and capture processes, electron-impact excitation and scattering processes, and at many places elsewhere. Typically, the particular form of the electron–electron interaction, i.e., of either applying $V = V^{(Coulomb)}$ or $V = V^{(Coulomb)} + V^{(Breit)}$, or even some further approximation to 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, or any practical arguments.

Electron–photon amplitudes $\langle \alpha_f \mathbb{J}_f \parallel \mathbb{O}^{(M)} \parallel \alpha_i \mathbb{J}_i \rangle$: This interaction amplitude can be handled separately for the different multipole components $\mathbb{M} \in \{E1, M1, E2, \dots\}$ of the radiation field but requires to carefully distinguish between absorption and emission processes,

$$\langle \alpha_f \mathbb{J}_f \parallel \mathbb{O}^{(M, \text{absorption})} \parallel \alpha_i \mathbb{J}_i \rangle \equiv \langle \alpha_f \mathbb{J}_f \parallel \sum_{k=1}^N a_k^{(M)} \parallel \alpha_i \mathbb{J}_i \rangle = \langle \alpha_i \mathbb{J}_i \parallel \mathbb{O}^{(M, \text{emission})} \parallel \alpha_f \mathbb{J}_f \rangle^*$$

as well as between *bound–bound*, *bound–free* and *free–free* amplitudes, if some of the electron(s) appear in the continuum. This distinction is necessary also in order to ensure the correct phase relation between different multipole components, especially if different interaction amplitudes need to be combined. Here, $a_k^{(M)}$ denotes the electron–photon interaction operator for the annihilation of a photon with multipolarity $\mathbb{M} \equiv (L, p)$, and where $p = 0$ refers to the magnetic and $p = 1$ to the electric multipoles of rank L . Electron–photon interaction amplitudes frequently arise in all photoexcitation, ionization and capture processes as well as for the computation of transition probabilities, dielectronic recombination strength, multi-photon processes, atomic cascades, etc.

While we shall not discuss details of the underlying relativistic theory, let us outline the remarkable role of the many-electron amplitudes (2)–(3) and how they help to make atomic theory ready for contemporary appliance. We here assemble a few selected and desirable extensions which have been suggested by current experiments but were not considered (yet) by general-purpose atomic structure codes.

- (a) **Excitation schemes based on the atomic shell model:** Many atomic computations start from a given set of reference (electron) configurations in order to generate relevant subspaces of the many-electron Hilbert space. This is accomplished by specifying virtual excitations of (so-called) active electrons, while the selection of these subspaces certainly require a good physical understanding, there explicit setup is usually performed automatically. To support well-chosen subspaces for different applications, a proper notion need to be developed and implemented to readily manipulate ‘lists’ of electron configurations and configuration schemes. For the dielectronic recombination (DR) of a multiply-charged ion, for example, all configurations with a single (sub-) valence excitation as well as the capture of an additional electron into any valence shell, taken from a given list of such shells, contribute to some resonance strength and, hence, to the DR plasma rate coefficients, provided that the energy difference of the doubly-excited resonance and the initial level is positive, $E_d - E_i = \varepsilon > 0$. For rate coefficients, in particular, the setup of the computations can be simplified considerably if such an *excite-by-capture* scheme is established and readily applied to the given set of reference configurations. The associated list of configurations is then generated automatically. There are many other applications of atomic structure theory which can be made much simpler by establishing the notion of proper excitation and de-excitation schemes, and which are all based on the shell model.
- (b) **Coupling of atoms to the free-electron continuum:** If atoms or ions are resonantly excited into the continuum of the next higher charge state, they are inherently coupled to this “infinity of states” owing to the (non-local) electron–electron interaction. In atomic theory, this coupling is often expressed by means of photo- and autoionization amplitudes, and with the electron–photon and electron–electron interaction matrix

elements from above. Hereby, different ionization channels need to be distinguished due to the coupling of the —partial waves of—free electron with energy ε to the bound state of the remaining ion: $|\alpha_f \mathbb{J}_f, \varepsilon \kappa \mathbb{J}_t\rangle$ as well as due to the selection rules of the considered process. Any successful implementation of ionization (and recombination) processes therefore needs a straight and quick access to these many-electron amplitudes, including one or several free electron waves $|\varepsilon \kappa\rangle$.

- (c) **Modeling of atomic cascades:** Such cascades typically arise when inner-shell electrons are excited, or ionized, and subsequently lead to the emission of two or more electrons and/or photons. Indeed, atomic cascades are quite ubiquitous in nature and have therefore been explored for different scenarios, from precision measurements to the modeling of astrophysical spectra, and up to the search for exotic particles. Until the present, however, no quantitative analysis of such decay cascades appear often to be feasible owing to the large, or even huge, number of decay pathways which the atoms can undertake. Since most, if not all, of these cascades can normally be traced back to a proper combination of different symmetry-adapted amplitudes, they are suitable for general-purpose codes if the computation of these amplitudes is separated from the subsequent simulation of the associated ion, photon or electron distribution, or any other wanted information. In order to deal with different excitation and decay scenarios, Ref. [57] suggests a number of (cascade) *schemes* and *approaches* that are clearly discernible with regard to their complexity and computational costs. Typical schemes refer to the decay of inner-shell holes, including the prior excitation or ionization of electrons [58], the radiative or dielectronic capture of electrons [59], or the formation and photoemission from hollow ions. Many observations in astrophysics will be understood better, or still at all, if the interplay between the radiative and nonradiative decay pattern can be modeled with sufficient detail.
- (d) **Request of atomic Green functions:** In atomic and many-particle physics, Green functions often occur as propagators to formally represent the (integration over the) complete spectrum of the underlying Hamiltonian. Since such an integration over the complete spectrum is crucial also to many second- and higher-order perturbation processes, approximate atomic Green functions need to be developed and accessible rather similar to how the ASF can be applied.

A simple access to approximate Green functions requires however a decomposition into building blocks that are suitable for atomic structure theory [50]. By making use of the rotational symmetry and parity of the ASF $|\alpha_\nu \mathbb{J}_\nu M_\nu\rangle$, each Green function can be split into separate channels (continua) of well-defined symmetry $\mathbb{J} \equiv J^P$, quite similar to the one-electron Coulomb–Green function [60]. Within a finite basis, these channels then simply refer to a set of atomic levels $\{E_\nu^{(\mathbb{J})}, |\alpha_\nu^{(\mathbb{J})} M^{(\mathbb{J})}\rangle, \nu = 1, \dots, \nu^{(\mathbb{J})}\}$, all of the same total symmetry \mathbb{J} . Using this notation, an approximate atomic Green function is given by an array (list) of k such channels and is formally written as

$$\left[\left\{ E_\nu^{(\mathbb{J}_1)}, |\alpha_\nu^{(\mathbb{J}_1)} M^{(\mathbb{J}_1)}\rangle, \nu = 1, \dots, \nu^{(\mathbb{J}_1)} \right\}, \left\{ E_\nu^{(\mathbb{J}_2)}, |\alpha_\nu^{(\mathbb{J}_2)} M^{(\mathbb{J}_2)}\rangle, \nu = 1, \dots, \nu^{(\mathbb{J}_2)} \right\}, \dots \right]$$

$$\iff G_E = \sum_\nu \frac{|\alpha_\nu \mathbb{J}_\nu M_\nu\rangle \langle \alpha_\nu \mathbb{J}_\nu M_\nu|}{E_\nu - E} \equiv \sum_{\mathbb{J}} \sum_{M=-J}^J \sum_\nu^{\nu^{(\mathbb{J})}} \frac{|\alpha_\nu \mathbb{J} M\rangle \langle \alpha_\nu \mathbb{J} M|}{E_\nu^{(\mathbb{J})} - E} \tag{5}$$

or, in other words, the k -th channel of an approximate Green function is represented by a finite list of many-electron (pseudo-)levels $\{E_\nu^{(\mathbb{J}_k)}, |\alpha_\nu^{(\mathbb{J}_k)} M^{(\mathbb{J}_k)}\rangle, \nu = 1, \dots, \nu^{(\mathbb{J}_k)}\}$. As for the ASF (1), these pseudo-levels are obtained by diagonalizing the associated Hamiltonian matrix, though care has to be taken about the levels within the continuum. This is particularly true, if some atomic process leads to a multiple ionization (or capture) of electrons, such as the double Auger emission.

While, formally, each Green function is built from an infinite number of such channels, and including both parities $P = \pm 1$, only a few of these channels (continua) are in

practice relevant for any non-linear interaction process, either because of the symmetry of the underlying interaction operators or due to further insights into the physics.

- (e) **Nonlinear atomic processes:** The explicit use of the many-electron interaction amplitudes from above also facilitates the implementation of nonlinear, e.g., second- and higher-order processes, once an appropriate Green function has been constructed for some given property or process of interest. Well-known second-order processes are the two-photon absorption and emission, the resonant and two-photon ionization, the radiative and double Auger emission of atoms, or the (single-photon) double ionization, to just recall a few of them. For each of these processes, different selection rules apply and help to restrict the number of Green function channels with total symmetry $\mathbb{J}_1, \mathbb{J}_2, \dots$, that need to be generated. In these computations, the summation (integration) remains the most challenging part of all numerical computations owing to number of terms in the representation of the Green function (5). Further difficulties may arise from the free-free matrix elements that occur in all ionization and capture processes. However, the recent interest and observation of multi-photon or multi-electron processes make the calculation of such nonlinear processes by general-purpose codes such as GRASP for sure highly desirable.
- (f) **Different representation of atomic states:** In the MCDHF method, the representation $\{c_r(\alpha)\}$ of the ASF is fully determined by choosing the energy functional of the SCF and the CSF basis in terms of virtual excitations with regard to a set of (bound) reference configurations. These atomic states can then be directly applied also to evaluate the interaction amplitudes and, hence, the observables of interest. Other representations are obtained by adopting the functional of the underlying mean field or by treating certain classes of excitation perturbatively. Again, many of these possible extensions of atomic theory are formulated in terms of the electron–electron interaction amplitudes from above. Moreover, rather similar amplitudes arise if the dominant QED corrections are to be taken into account by an single-electron model Hamiltonian [61]

$$\mathbb{H}^{(\text{QED})} = \mathbb{H}^{(\text{SE})} + \mathbb{H}^{(\text{VP})} = \sum_j h_j^{(\text{QED})} = \sum_j \left(h_j^{(\text{SE})} + h_j^{(\text{VP})} \right),$$

that comprises the *effective* self-energy (SE) and vacuum-polarization (VP) terms. When compared to missing electronic correlations, these QED corrections are often less relevant as long as no inner-shell excitation are involved in any computed property or process [62].

Apart from such ‘traditional’ many-body techniques, the qubit representation of the atomic Hamiltonian and the use of quantum hardware has attracted much recent interest. Different transformations, such as the Jordan–Wigner or Bravyi–Kitaev transformation, aim to map the electronic structure of indistinguishable fermions upon distinguishable qubits and to use algorithms that are exponentially faster than the best-known classical algorithms. Up to the present, however, no efficient algorithm is known to solve electronic-structure problems in a fully general form [63]. It will therefore be beneficial to explore how the symmetry-adapted amplitudes from above can be transformed and evaluated by means of qubit Hamiltonians and quantum circuits.

These and several other applications may become indeed realistic within the near future if a proper set of symmetry-adapted atomic amplitudes are accessible and used in order to built up features of higher complexity. Apart from a descriptive language for specifying individual steps of the computations and a flexible user interface, this requires an efficient computational framework, such as GRASP, in order to support most atoms and ions across the periodic table.

3.2. A Descriptive Language for Atomic Computations

While the methods and numerical algorithms in atomic structure theory are usually explained by using a rather compact and well-established notation, suitable for the symmetry of free atoms and ions, their employment often suffers from a—more or less—sophisticated implementation and interface. The implemented jargon and data flow then hinder the extension of code towards emerging applications, which were not in mind right from the beginning of the code design.

These technical obstacles can be substantially lowered by using a descriptive atomic language that reflects the underlying formalism and that avoids most technical slang. Of course, such a language should be based on well-chosen data types which help express the common atomic structure and collision theory and which facilitate the communication with and within the program. Moreover, such a language should be suitable for interactive work and simple enough for both, seldom as well as more frequent use of the code. Figure 1 displays a few of the requirements for establishing a domain-specific and *descriptive* atomic language. Obviously, all these features are also relevant for a code such as GRASP.

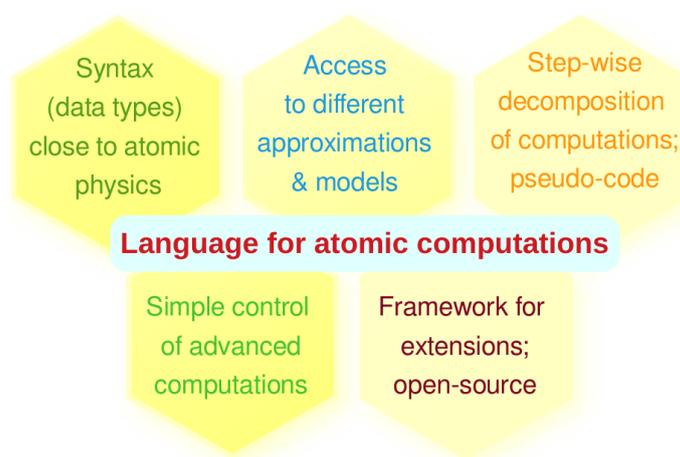


Figure 1. Requirements for establishing a domain-specific and *descriptive* atomic language.

Apart from a concise syntax, such a language and computational frame should:

- (1) Provide an intuitive user interface in which all computations can be expressed quite similar to the user's research work;
- (2) Supply features for dealing with general open-shell configurations and applications in atomic physics and elsewhere; such features refer for instance to a self-acting generation of (lists of) configurations or Green function channels by just selecting suitable classes of virtual excitations;
- (3) Help redefine the physical units to the needs of the user, both at input and output time;
- (4) Give access to different models and approximations as well as the decomposition of a given task into well-designed steps, rather similar to writing pseudo-code;
- (5) Resort to *default* values whenever feasible; indeed, all parameters, which can be derived from other input or which is irrelevant for a given computation, should be based on default values, though with keeping the user informed. On the other hand, it must be quite easy to overwrite these defaults for selected applications;
- (6) Enable the user to redefine physical constants or default settings;
- (7) Support a transparent data flow with and within the program, independent of the shell structure of the atoms or any particular application;
- (8) Help visualize large data sets as often required for the analysis of results and for debugging;
- (9) Support the interactive use (test) of individual data types, functions or single lines of the code.

These requirements are quite opposite to most previous—either FORTRAN or C—codes, for which simple extensions, a rapid prototyping or the use of graphical interfaces appear to be cumbersome. Moreover, these requirements on a high-level language cannot be realized, as outlined above, without a proper computational framework and programming language that is fast enough to allow for quantum (many-electron) computations, and which easily supports the implementation of all the necessary building blocks.

In contrast to earlier concepts, several programming languages are designed today with a number of powerful features, such as dynamic types, type-specializing, dynamic code loading or garbage collection, that are common to so-called productivity languages. Julia is one of these programming languages that enables one to build-up new syntax for domain-specific languages and that is fast enough for scientific computing [64,65]. This language also includes (i) an expressive *type system* that allows optional type annotations to support an offensive code specialization against run-time types; (ii) *multiple dispatch* to dynamically select the most suitable procedure for running the right code at the right time; (iii) *Just-In-Time* compilation; (iv) features for *parallelization*; as well as (v) rich metaprogramming facilities. It also provides “shared” arrays to distribute data to multiple parallel processes for high-performance computing. Together with its deliberate language design, all these technologies may help to realize a descriptive language that combines high productivity in developing code with good performance.

3.3. Impact of Proper Data Types

A proper choice and hierarchy of data types are indeed crucial for setting-up such a descriptive and domain-specific language which clearly reflects the formalism behind the program [66]. A proper type hierarchy hereby helps to express relationships between data types, while abstract data types are often utilized to model the behavior of the code and to select the code dynamically, the actual data is kept by and always flow through concrete data types. These concrete types usually exist either as primitive or composite types. In Julia, moreover, all types are said to be first-class and are utilized to select the code dynamically at run time. This is achieved by (so-called) multiple dispatch that refers to the dynamic selection and execution of code, i.e., the choice of the most specific method applicable to the types of its arguments. Multiple dispatch makes it easier to structure the programs close to the underlying physics. In addition, it obviously reduces the needs for argument checking at the begin of a function, sometimes also referred to as ad hoc polymorphism. In practice, Julia’s compiler automatically generates specialized functions according to the type of its arguments.

For an “atomic language”, of course, the (notion of all) data types should be readily understandable to any atomic physicist without much additional training. They need to be based on useful and frequently recurring objects in the computation of atomic energies, state vectors and processes, and they have to be independent from the particular shell structure of the atoms or ions of interest. Examples may refer to a `Shell`, `Configuration`, `Orbital`, `Level`, or even to a whole `Multiplet` of such levels. In practice, such a `Level` should provide access to all details of the associated ASF: $E, |\alpha JM\rangle$, including its radial wave functions, the values and coupling of the angular momenta as well as the (number of and) mixing of CSF within the given basis, while most data types remain hidden to the user, every new feature of the code typically requires a few data types that are particularly suitable. Table 1 lists a few selected data types from the JAC toolbox, a general-purpose code which will be briefly summarized in the next section.

General-purpose codes such as GRASP cannot be established without separating its functionality into independent and partly interchangeable modules. In each of these modules, an interface (should) express the elements that are provided and required by the module and that enables one to generate re-usable code without enforcing the user to take care about the name space and arguments of all subsequent calls. In atomic theory, an independent module can be readily assigned to individual steps in solving the MCDHF equations, such as the construction of the CSF basis or the SCF iteration, but also

for each atomic property and process of interest. In dealing with excitation, ionization and recombination processes, furthermore, the terms *line* and *channel* frequently occur in order to distinguish between the observed spectral features and the (symmetry-adapted) amplitudes that formally contribute to these features. For a photon emission process, for example, different channels must distinguish between different multipole components (and perhaps the gauge for the coupling of the radiation field), while a photoionization channel needs to also account for the partial waves of the outgoing electron. The associated data types `Channel` and `Line` will therefore occur in very different modules, though with slightly altered specification and constructors, while a very few derived types have meanwhile been introduced into GRASP, we expect that *this* and other general-purpose code will benefit from the specification and consequent use of a more advanced data type hierarchy.

Table 1. Selected data types (struct) from the JAC toolbox [14] for representing important building blocks from atomic structure theory. In total, there are at present about ~250 of these data structures in JAC, though most of them remain hidden to the user. A more detailed description of these data types can be obtained interactively, for instance by ? `Level`, to recall the purpose of this struct and the definition of all subfields.

Struct	Brief Explanation
<code>AbstractEeInteraction</code>	Abstract type to distinguish between different electron–electron interaction operators; it comprises the concrete (singleton) types <code>BreitInteraction</code> , <code>CoulombInteraction</code> , <code>CoulombBreit</code> .
<code>AbstractExcitationScheme</code>	Abstract type to support different excitation schemes, such as <code>DeExciteSingleElectron</code> , <code>DeExciteTwoElectrons</code> , <code>ExciteByCapture</code> , and several others.
<code>AbstractGreenApproach</code>	Defines an abstract type for approximating a many-electron Green function expansion, and which comprises the concrete (singleton) types <code>SingleCSFwithoutCI</code> and <code>CoreSpaceCI</code> .
<code>AbstractScField</code>	Abstract type for dealing with different self-consistent-field (SCF) potentials.
<code>AsfSettings</code>	Settings to control the SCF and CI calculations for a given multiplet of ASF.
<code>Atomic.Computation</code>	An atomic computation of one or several multiplets, including the SCF and CI calculations, as well as of selected properties or processes.
<code>Basis</code>	Relativistic atomic basis, including the full specification of the configuration space and radial orbitals.
<code>Configuration</code>	Non-relativistic electron configuration in terms of its shell occupation.
<code>ConfigurationR</code>	Relativistic electron configuration in terms of its subshell occupation.
<code>EmMultipole</code>	A multipole component of the electro-magnetic field, such as E1, M1, E2, ... and as specified by its parity and multipolarity.
<code>GreenChannel</code>	A single approximate Green function channel with well-defined symmetry \mathbb{J} .
<code>Level</code>	Atomic level in terms of its quantum numbers, symmetry, energy and its (possibly full) representation.
<code>LevelSelection</code>	List of levels that is specified by either the level numbers and/or level symmetries.
<code>LevelSymmetry</code>	$\mathbb{J} = J^P$ specifies the total angular momentum and parity of a particular level.

Table 1. *Cont.*

Struct	Brief Explanation
LSjjSettings	Settings to control the $jjj - LSJ$ transformation of the selected many-electron levels.
Multiplet	An ordered list of atomic levels, often associated with one or several configurations.
Nuclear.Model	A nuclear model of an atom to keep all nuclear parameters together.
Orbital	Relativistic radial orbital function that appears as ‘building block’ in order to define the many-electron CSF; its is typically given on a (radial) grid and comprises a large and small component.
Radial.Grid	Radial grid to represent the (radial) orbitals and to perform all radial integrations.
Radial.Potential	Radial potential function.
Shell	Non-relativistic shell, such as $1s, 2s, 2p, \dots$
Subshell	Relativistic subshell, such as $1s_{1/2}, 2s_{1/2}, 2p_{1/2}, 2p_{3/2}, \dots$

3.4. Jena Atomic Calculator

Several of the concepts, mentioned above, have been implemented and already utilized in JAC, the Jena Atomic Calculator [14]. This toolbox help calculate symmetry-adapted interaction amplitudes, properties as well as a large number of excitation and decay processes for atoms and ions with complex shell structure. It can be applied quite readily without much prior knowledge of the code. The (so-called) Atomic.Computation’s from this toolbox [cf. Table 1] are based on explicitly specified electron configurations and provide level energies, the representation of ASF or their classification within a LSJ -coupled basis. Since JAC’s very first design in 2017, the number of atomic properties and processes, that can be handled by this code, has steadily grown and now supports the generation of atomic data for astro and plasma physics [59]. Little needs to be said about JAC that has already been described elsewhere [14] and can readily be downloaded from the web [67]. However, even if JAC provides a convenient and very flexible environment for code development, verification and analysis, this toolbox still suffers from —more or less—severe efficiency and memory issues, or the simple re-use of angular coefficients, when compared to the GRASP program.

However, JAC already implements essential parts of a *descriptive* language by its careful design (and interplay) of data types and generic functions. For instance, Figure 2 displays the implementation of the electron–electron interaction amplitude $\langle (\alpha_f \mathbb{J}_f, \varepsilon \kappa) \mathbb{J}_i \| V \| \alpha_i \mathbb{J}_i \rangle$ within the JAC toolbox for the autoionization of an atom that is initially found in the level $\alpha_i \mathbb{J}_i$, into the final level $\alpha_f \mathbb{J}_f$ and with a free electron in the partial wave $|\varepsilon \kappa\rangle$. This symmetry-adapted amplitude refers to a single autoionization channel with the same total symmetry \mathbb{J}_i as given by the initial level. The two levels $\alpha_i \mathbb{J}_i$ and $\alpha_c \mathbb{J}_c = (\alpha_f \mathbb{J}_f, \varepsilon \kappa) \mathbb{J}_i$ enter the computations by the variables `iLevel` and `cLevel` and provide all information for the evaluation of the amplitude, including a simple access to the associated CSF bases, the electron orbitals, mixing coefficients and others. Moreover, while the spin-angular coefficients are calculated for a general symmetric (rank-0) two-particle operator, and separately for each pair of CSF from the initial and continuum level, the kind of electron–electron interaction is readily distinguished. The selection of the proper interaction operator makes use of the abstract data type `AbstractEeInteraction` (cf. Table 1) in order to distinguish between the $V^{(\text{Coulomb})}$, $V^{(\text{Breit})}$ or $V = V^{(\text{Coulomb})} + V^{(\text{Breit})}$, respectively. Finally, the matrix elements for a pair of CSF is weighted by the mixing coefficients of the initial and final levels: $\langle (\alpha_f \mathbb{J}_f, \varepsilon \kappa) \mathbb{J}_i \| V \| \alpha_i \mathbb{J}_i \rangle = \sum_{ci} c_c (\alpha_f \mathbb{J}_f)^* \cdot \langle \text{CSF}_c \| V \| \text{CSF}_i \rangle \cdot c_i (\alpha_i \mathbb{J}_i)$. A similar code applies for all other interaction amplitudes. Therefore, this example nicely shows how

the formal notion of the many-electron matrix elements finds its simple (and easily readable) implementation into the code. It also demonstrates that all data should be articulated in a form most natural for the programmer in order to enable future improvements. We therefore believe that all general-purpose codes should be developed along these or similar lines for using up the predictive power of relativistic atomic theory.

```

# Autoionization (Auger) interaction amplitudes between two ASF
amplitude = complexF64(0.)
#
for r = 1:length(cLevel.basis.csfs)
  for s = 1:length(iLevel.basis.csfs)
    if iLevel.basis.csfs[s].J != iLevel.J || iLevel.basis.csfs[s].parity != iLevel.parity
      continue # Proceed if selection rules are not fulfilled.
    end
    # Calculate the spin-angular coefficients
    subshells = cLevel.basis.subshells
    op = SpinAngular.TwoParticleOperator(0, plus, true)
    wa = SpinAngular.computeCoefficients(op, cLevel.basis.csfs[r], iLevel.basis.csfs[s], subshells)
    #
    me = 0.
    for co in wa # cycle over all spin-angular coefficients (co) of the given CSF pair
      if kind in [CoulombInteraction(), CoulombBreit()]
        me = me + co.V * InteractionStrength.XL_Coulomb(co.nu,
          cLevel.basis.orbitals[co.a], cLevel.basis.orbitals[co.b],
          iLevel.basis.orbitals[co.c], iLevel.basis.orbitals[co.d], grid)
      end
      if kind in [BreitInteraction(), CoulombBreit()]
        me = me + co.V * InteractionStrength.XL_Breit(co.nu,
          cLevel.basis.orbitals[co.a], cLevel.basis.orbitals[co.b],
          iLevel.basis.orbitals[co.c], iLevel.basis.orbitals[co.d], grid)
      end
    end
    amplitude = amplitude + cLevel.mc[r] * me * iLevel.mc[s]
  end
end
end

```

Figure 2. Implementation of the electron–electron interaction amplitudes Please cite the figure in the text and ensure the first citation of each figure appears in numerical order.

4. Summary and Conclusions

As argued and discussed above, the use of symmetry-adapted CSF with well-defined (total) symmetry $\mathbb{J} = J^P$ help more or less readily to decompose and compute (reduced) amplitudes such as $\langle \alpha \mathbb{J} M | \mathbb{T}^{(KQ)} | \beta \mathbb{J}' M' \rangle$ and $\langle \alpha \mathbb{J} | \mathbb{T}^{(K)} | \beta \mathbb{J}' \rangle$ for every operator \mathbb{T} of rank K . Indeed, these amplitudes are very central for describing the interaction among the electrons or with external particles and fields and, hence, for expressing atomic behavior in terms of (energy) shifts, rates, cross sections, angular distribution parameters and many other properties which are accessible to experimentation. These amplitudes should therefore always be smoothly accessible by any general-purpose code in order to apply (relativistic) atomic theory to emerging fields.

A detailed computational procedure with systematically improved atomic states still need to be worked out for many potential applications of (atomic) theory. Most of them are associated with free electrons in the continuum and will benefit especially from a consequent use of symmetry-adapted amplitudes. These applications also emphasize the need for a descriptive and concise language close to the underlying formalism. Further requests for developing general-purpose codes likely refer to a suitable categorization of physical scenarios, a detailed documentation of the code as well as to features for integrated testing. All these requirements are of present concern in developing GRASP and other codes and will make them ready for future challenges.

Funding: This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 861198-LISA-H2020-MSCA-ITN-2019.

Acknowledgments: This work is dedicated with genuine gratitude to Ian P. Grant for his long-term efforts in developing relativistic atomic structure theory.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Swirls, B. The relativistic self-consistent field. *Proc. Roy. Soc. Lond.* **1935**, *152*, 625.
2. Grant, I.P. Relativistic self-consistent fields. *Proc. Roy. Soc. A* **1961**, *262*, 555. [[CrossRef](#)]
3. Grant, I.P.; Burke, V.M. The effect of relativity on atomic wavefunctions. *Proc. Phys. Soc.* **1967**, *90*, 297.
4. Desclaux, J.P. A multiconfiguration relativistic Dirac-Fock program. *Comp. Phys. Commun.* **1975**, *9*, 31. [[CrossRef](#)]
5. Grant, I.P.; McKenzie, B.J.; Norrington, P.H.; Mayers, D.F.; Pyper, N.C. An atomic multiconfiguration Dirac-Fock package. *Comp. Phys. Commun.* **1980**, *21*, 207. [[CrossRef](#)]
6. Parpia, F.A.; Fischer, C.F.; Grant, I.P. GRASP92: A package for large-scale relativistic atomic structure calculations. *Comp. Phys. Commun.* **1996**, *94*, 249. [[CrossRef](#)]
7. Froese Fischer, C.; Gaigalas, G.; Jönsson, P.; Bieroń, J. GRASP2018—A Fortran 95 version of the General Relativistic Atomic Structure Package. *Comp. Phys. Commun.* **2019**, *13*, 520. [[CrossRef](#)]
8. Judd, R. *Operator Techniques in Atomic Spectroscopy*; McGraw-Hill: New York, NY, USA, 1963.
9. Johnson, W.R. *Atomic Structure Theory: Lectures on Atomic Physics*; Springer: Berlin/Heidelberg, Germany, 2007.
10. Grant, I.P. *Relativistic Quantum Theory of Atoms and Molecules: Theory and Computation*; Springer: Berlin/Heidelberg, Germany, 2007.
11. Indelicato, P. Available online: http://dirac.spectro.jussieu.fr/mcdf/mcdf_code/mcdfgme_status.html (accessed on 10 August 2022).
12. Fritzsche, S. A toolbox for studying the properties of open-shell atoms and ions. *J. Electron. Spectrosc. Relat. Phenom.* **2001**, *114–116*, 1155. [[CrossRef](#)]
13. Gaigalas, G.; Zalandauskas, T.; Fritzsche, S. Spectroscopic *LSJ* notation for atomic levels as obtained from relativistic calculations. *Comput. Phys. Commun.* **2004**, *157*, 239. [[CrossRef](#)]
14. Fritzsche, S. A fresh computational approach to atomic structures, processes and cascades. *Comp. Phys. Commun.* **2019**, *240*, 1. [[CrossRef](#)]
15. Grant, I.P. Relativistic calculation of atomic structures. *Adv. Phys.* **1970**, *19*, 747. [[CrossRef](#)]
16. McKenzie, B.J.; Grant, I.P.; Norrington, P.H. A program to calculate transverse Breit and QED corrections to energy levels in a MCDF environment. *Comp. Phys. Commun.* **1980**, *21*, 233. [[CrossRef](#)]
17. Brink, M.; Satchler, G.R. *Angular Momentum*, 2nd ed.; Clarendon Press: Oxford, UK, 1968.
18. de-Shalit, A.; Feshbach, H. *Theoretical Nuclear Physics: Nuclear Structure*; Wiley: Hoboken, NJ, USA, 1974.
19. Blatt, M.; Weisskopf, V.F. *Theoretical Nuclear Physics*; Springer: New York, NY, USA, 1979.
20. Froese Fischer, C.; Godefroid, M.; Brage, T.; Jönsson, P.; Gaigalas, G. Advanced multiconfiguration methods for complex atoms: I. Energies and wave functions. *J. Phys. B* **2016**, *49*, 182004. [[CrossRef](#)]
21. Froese Fischer, C.; Tachiev, G.; Irimia, A. Relativistic energy levels, lifetimes, and transition probabilities for the sodium-like to argon-like sequences. *Atom. Data Nucl. Data Tabl.* **2006**, *92*, 607. [[CrossRef](#)]
22. Dong, C.Z.; Fritzsche, S.; Fricke, B.; Sepp, W.-D. Branching ratios and lifetimes of the low-lying levels of Fe X. *Mon. Notes R. Astr. Soc.* **1999**, *307*, 809. [[CrossRef](#)]
23. Bieroń, J.; Froese Fischer, C.; Grant, I.P. Large-scale multiconfigurational Dirac-Fock calculations of the hyperfine-structure constants and determination of the nuclear quadrupole moment of ^{49}Ti . *Phys. Rev. A* **1999**, *59*, 4295. [[CrossRef](#)]
24. Bieroń, J.; Jönsson, P.; Froese Fischer, C. Effects of electron correlation, relativity, and nuclear structure on hyperfine constants of Be^+ and F^{6+} . *Phys. Rev. A* **1999**, *60*, 3547. [[CrossRef](#)]
25. Fritzsche, S.; Zschornack, G.; Musiol, G.; Soff, G. Interchannel interactions in highly-energetic radiationless transitions of neonlike ions. *Phys. Rev. A* **1991**, *44*, 388. [[CrossRef](#)]
26. Fritzsche, S.; Fricke, B.; Sepp, W.D. Reduced L_1 level-width and Coster-Kronig yields by relaxation and continuum interactions in atomic zinc. *Phys. Rev. A* **1992**, *45*, 1465. [[CrossRef](#)]
27. Sienkiewicz, J.E.; Fritzsche, S.; Grant, I.P. Relativistic configuration interaction approach to the elastic low-energy scattering of electrons from atoms. *J. Phys. B* **1995**, *28*, L633. [[CrossRef](#)]
28. Grant, I.P. Relativistic Effects in Atoms and Molecules. In *Methods in Computational Chemistry*; Wilson, S., Ed.; Plenum: New York, NY, USA, 1988; Volume 2, pp. 1–38.
29. Grant, I.P. A general program to calculate angular momentum coefficients in relativistic atomic structure. *Comp. Phys. Commun.* **1973**, *4*, 263. [[CrossRef](#)]
30. Gaigalas, G.; Fritzsche, S.; Grant, I.P. Calculation of pure angular coefficients in *jj*-coupling. *Comp. Phys. Commun.* **2001**, *139*, 263. [[CrossRef](#)]
31. Gaigalas, G.; Fritzsche, S. Angular coefficients for symmetry-adapted configuration states in *jj*-coupling. *Comput. Phys. Commun.* **2021**, *267*, 108086. [[CrossRef](#)]

32. Filippin, L.; Bieroń, J.; Gaigalas, G.; Godefroid, M.; Jönsson, P. Multiconfiguration calculations of electronic isotope-shift factors in Zn I. *Phys. Rev. A* **2017**, *96*, 042502. [[CrossRef](#)]
33. Eronen, T.; Fritzsche, S.; Geldhof, S.; Kelly, S.; Moore, I.D.; Pohjalainen, I.; Reponen, M.; Rinta-Antila, S.; Voss, A. Isotope shifts from collinear laser spectroscopy of doubly-charged yttrium isotopes. *Phys. Rev. A* **2018**, *97*, 042504.
34. Fischer, C.F.; Gaigalas, G. The Effect of correlation on spectra of the lanthanides: Pr³⁺. *Atoms* **2018**, *6*, 8. [[CrossRef](#)]
35. Perger, W.F.; Halabuka, Z.; Trautmann, D. Continuum wavefunction solver for GRASP. *Comp. Phys. Commun.* **1993**, *76*, 250. [[CrossRef](#)]
36. Zhang, H.L.; Sampson, D.H. Relativistic distorted-wave collision strengths and oscillator strengths for all possible $n = 2 - n = 3$ transitions in b-like ions. *Atom. Data Nucl. Data Tabl.* **1994**, *58*, 255. [[CrossRef](#)]
37. Fritzsche, S. The RATIP program for relativistic calculations of atomic transition, ionization and recombination properties. *Comp. Phys. Commun.* **2012**, *183*, 1525. [[CrossRef](#)]
38. Saha, B.; Fritzsche, S. Influence of dense plasma on the low-lying transitions in Be-like ions: Relativistic multiconfiguration Dirac-Fock calculation. *J. Phys. B* **2007**, *40*, 259. [[CrossRef](#)]
39. Norrington, P.H.; Grant, I.P. Low-energy electron scattering by Fe XXIII and Fe VII using the Dirac R-matrix method. *J. Phys. B* **1987**, *20*, 4869. [[CrossRef](#)]
40. Olsen, J.; Godefroid, M.; Jönsson, P.; Malmqvist, P.A.; Froese Fischer, C. Transition probability calculations for atoms using nonorthogonal orbitals. *Phys. Rev. E* **1995**, *52*, 4499. [[CrossRef](#)] [[PubMed](#)]
41. Fritzsche, S.; Grant, I.P. A program for the complete expansion of jj -coupled symmetry functions into Slater determinants. *Comp. Phys. Commun.* **1995**, *92*, 111. [[CrossRef](#)]
42. Jönsson, P.; Froese Fischer, C. Multiconfiguration Dirac-Fock calculations of the $2s^2\ ^1S_0 - 2s2p\ ^3P_1$ intercombination transition in C III. *Phys. Rev. A* **1998**, *57*, 4967. [[CrossRef](#)]
43. Yordanov, D.T.; Balabanski, D.L.; Bieroń, J.; Bissell, M.L.; Blaum, K.; Budinčević, I.; Fritzsche, S.; Frömmgen, N.; Georgiev, G.; Geppert, C.; et al. Spins, Electromagnetic moments and isomers of ^{107–129}Cd. *Phys. Rev. Lett.* **2013**, *110*, 192501. [[CrossRef](#)]
44. Raeder, S.; Ackermann, D.; Backe, H.; Block, M.; Cheal, B.; Chhetri, P.; Düllmann, C.E.; Van Duppen, P.; Even, J.; Ferrer, R.; et al. Nuclear properties of nobelium isotopes from laser spectroscopy. *Phys. Rev. Lett.* **2018**, *120*, 232503. [[CrossRef](#)] [[PubMed](#)]
45. Eliav, E.; Fritzsche, S.; Kaldor, U. Electronic structure theory of the superheavy elements. *Nucl. Phys.* **2015**, *A944*, 518. [[CrossRef](#)]
46. Bieroń, J.; Fischer, C.F.; Fritzsche, S.; Gaigalas, G.; Grant, I.P.; Indelicato, P.; Jönsson, P.; Pyykkö, P. Ab initio MCDHF calculations of electron-nucleus interactions. *Phys. Scr.* **2015**, *90*, 054011. [[CrossRef](#)]
47. Bieroń, J.; Filippin, L.; Gaigalas, G.; Godefroid, M.; Jönsson, P.; Pyykkö, P. Ab initio calculations of the hyperfine structure of zinc and evaluation of the nuclear quadrupole moment Q (Zn-67). *Phys. Rev. A* **2018**, *97*, 062505. [[CrossRef](#)]
48. Sato, T.K.; Asai, M.; Tsukada, K.; Kaneya, Y.; Toyoshima, A.; Makii, H.; Mitsukai, A.; Nagame, Y.; Osa, A.; Toyoshima, A.; et al. First ionization potentials of Fm, Md, No, and Lr: Verification of filling-up of $5f$ electrons and confirmation of the actinide series. *J. Amer. Chem. Soc.* **2018**, *140*, 14609. [[CrossRef](#)]
49. Åberg, T.; Howat, G. Theory of the Auger Effect. In *Corpuscles and Radiation in Matter I*; Encyclopedia of Physics Vol. XXXI; Mehlhorn, W., Ed.; Springer: Berlin/Heidelberg, Germany, 1982; p. 469.
50. Fritzsche, S.; Surzhykov, A. Approximate atomic Green functions. *Molecules* **2021**, *26*, 2660. [[CrossRef](#)] [[PubMed](#)]
51. Feldhaus, J. FLASH—The first soft X-ray free electron laser (FEL) user facility. *J. Phys. B* **2010**, *43*, 194002. [[CrossRef](#)]
52. Fritzsche, S.; Böning, B. Lorentz-force shifts in strong-field ionization with mid-infrared laser fields. *Phys. Rev. Res.* **2022**, *4*, 033031. [[CrossRef](#)]
53. Charlwood, F.C.; Billowes, J.; Campbell, P.; Cheal, B.; Eronen, T.; Forest, D.H.; Fritzsche, S.; Honma, M.; Jokinen, A.; Moore, I.D.; et al. Ground state properties of manganese isotopes across the $N = 28$ shell closure. *Phys. Lett. B* **2010**, *690*, 346. [[CrossRef](#)]
54. Cheal, B.; Cocolios, T.E.; Fritzsche, S. Laser spectroscopy of radioactive isotopes: Role and limitations of accurate isotope-shift calculations. *Phys. Rev. A* **2012**, *86*, 042501. [[CrossRef](#)]
55. Ferrer, R.; Barzakh, A.; Bastin, B.; Beerwerth, R.; Block, M.; Creemers, P.; Grawe, H.; de Groote, R.; Delahaye, P.; Fléchar, X.; et al. Towards high-resolution laser ionization spectroscopy of the heaviest elements in supersonic gas jet expansion. *Nat. Commun.* **2017**, *8*, 14520. [[CrossRef](#)]
56. Surzhykov, A.; Jentschura, U.D.; Stöhlker, T.; Gumberidze, A.; Fritzsche, S. Alignment of heavy few-electron ions following excitation by relativistic Coulomb collisions. *Phys. Rev. A* **2008**, *77*, 042722. [[CrossRef](#)]
57. Fritzsche, S.; Palmeri, P.; Schippers, S. Atomic cascade computations. *Symmetry* **2021**, *13*, 520. [[CrossRef](#)]
58. Schippers, S.; Martins, M.; Beerwerth, R.; Bari, S.; Holste, K.; Schubert, K.; Viefhaus, J.; Savin, D.W.; Fritzsche, S.; Müller, A.; et al. Near L-edge single and multiple photoionization of singly charged iron ions. *Astrophys. J.* **2017**, *849*, 5. [[CrossRef](#)]
59. Fritzsche, S. Dielectronic recombination strengths and plasma rate coefficients of multiply-charged ions. *Astron. Astrophys.* **2021**, *656*, A163. [[CrossRef](#)]
60. Wong, M.K.F.; Ye, E.H.Y. The Dirac Coulomb Green's function and its application to relativistic Rayleigh scattering. *J. Math. Phys.* **1985**, *26*, 1701. [[CrossRef](#)]
61. Shabaev, V.M.; Tupitsyn, I.I.; Yerokhin, V.A. Model operator approach to the Lamb shift calculations in relativistic many-electron atoms. *Phys. Rev. A* **2013**, *88*, 012513. [[CrossRef](#)]
62. Indelicato, P.; Santos, J.P.; Boucard, S.; Desclaux, J.-P. QED and relativistic corrections in superheavy elements. *Eur. Phys. J. D* **2007**, *45*, 155. [[CrossRef](#)]

63. Kandala, A.; Mezzacapo, A.; Temme, K.; Takita, M.; Brink, M.; Chow, J.M.; Gambetta, J.M. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature* **2017**, *549*, 242. [[CrossRef](#)] [[PubMed](#)]
64. Julia 1.8 Documentation. Available online: <https://docs.julialang.org/> (accessed on 10 August 2022).
65. Bezanson, J.; Chen, J.; Chung, B.; Karpinski, S.; Shah, V.B.; Vitek, J.; Zoubritzky, L. Julia: Dynamism and performance reconciled by design. *Proc. ACM Program. Lang.* **2018**, *2*, 120. [[CrossRef](#)]
66. Kwong, T. *Hands-On Design Patterns and Best Practices with Julia*; Packt Publishing: Birmingham, UK, 2020.
67. Fritzsche, S. JAC: User Guide, Compendium & Theoretical Background. Available online: <https://github.com/OpenJAC/JAC.jl> (accessed on 10 August 2022).