



UTILITIES for the RATIP package[☆]

S. Fritzsche

Fachbereich Physik, Universität Kassel, Heinrich-Plett-Str. 40, D-34132 Kassel, Germany

Accepted 18 June 2001

Abstract

During recent years, the RATIP package has been applied successfully for studying the behaviour and properties of open-shell atoms and ions within a relativistic framework. A series of case studies on the transition probabilities and lifetimes as well as for autoionization processes of different open-shell ions showed that it is possible today to make accurate predictions if all the dominant effects from relativity, correlation, and from the rearrangement of the electron density are treated on the same footings. Here, I present a *toolbox* for the RATIP package which supports the data handling and the interpretation of (complex) spectra. These UTILITIES have been found useful, in particular when large-scale applications ought to be carried out with RATIP. © 2001 Elsevier Science B.V. All rights reserved.

PROGRAM SUMMARY

Title of program: UTILITIES

Catalogue identifier: ADPD

Program Summary URL: <http://cpc.cs.qub.ac.uk/summaries/ADPD>

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland. Users may obtain the program also by down-loading a tar file `ratip-2001.tar` from our home page at the University of Kassel (<http://www.physik.uni-kassel.de/fritzsche/programs.html>)

Licensing provisions: None

Computer for which the program is designed and has been tested: IBM RS 6000, PC Pentium II

[☆] This program can be downloaded from the CPC Program Library under catalogue identifier: <http://cpc.cs.qub.ac.uk/summaries/ADPD>

E-mail address: s.fritzsche@physik.uni-kassel.de
(S. Fritzsche).

Installations: University of Kassel (Germany)

Operating systems: IBM AIX, Linux

Program language used in the new version: ANSI standard Fortran 90/95

Memory required to execute with typical data: 300 kB

No. of bits in a word: All real variables are parametrized by a selected `kind parameter` and, thus, can be adapted to any required precision if supported by the compiler. Currently, the `kind parameter` is set to double precision (two 32-bit words) as used also for other components of the RATIP package [1–3]

No. of bytes in distributed program, including test data, etc.: 3 509 413

Distribution format: tar gzip file

Number of lines in the program: about 35 000 altogether

Keywords: Atomic data and properties, Auger rate, branching ratio, configuration symmetry list, Einstein coefficient, file handling, lifetime, open-shell atom, RATIP utilities, relativistic

Nature of the physical problem

In the framework of the RATIP package, a toolbox supports a variety of (smaller) tasks. It facilitates, for example, the file and data handling in large-scale applications or the interpretation of complex spectra.

Method of solution

The main menu of the UTILITIES lists 16 tasks which can be selected interactively. Information, which is printed during the execution of the program, as well as an interactive dialog about the required input and (selected) output *guides* the user through the individual tasks.

Typical running time

The program responds promptly for most of the tasks below.

The responding time for other tasks, such as the generation of a relativistic *pair-correlation* basis, strongly depend on the size of the corresponding data files.

Unusual features of the program

A total of 16 different tasks are supported in the present version. Each of these tasks displays a short summary about its *function* and prompts for the required input and output data interactively.

References

- [1] S. Fritzsche, C.F. Fischer, C.Z. Dong, Comput. Phys. Commun. 124 (2000) 341.
- [2] G. Gaigalas, S. Fritzsche, Comput. Phys. Commun. 134 (2001) 86.
- [3] S. Fritzsche, J. Elec. Spec. Rel. Phen. 114–116 (2001) 1155.

LONG WRITE-UP

1. Introduction

Theoretical investigations on the level structure and the properties of atoms and ions have a long tradition. Over the years, this tradition — along with a steadily grown need of accurate data — has led to several powerful (program) tools using both, nonrelativistic [1,2] as well as relativistic approaches [3,4]. Today, reliable atomic data are required in a large number of research areas, including astro and plasma physics (the traditional ones) but also in many other, recently emerging fields like uv- and X-ray lithography, the generation of nanostructures, or the search for more efficient X-ray laser schemes.

In the relativistic domain of atomic structure theory, the RATIP package (for the calculation of relativistic atomic transition and ionization properties) supports case studies and *large-scale* applications for a variety of properties. A recently revised version of the package now applies the wave functions from the structure code GRASP92 [5] where an atomic state function (ASF) with given angular momentum and parity (J^P) is approximated by a linear combination of configuration state functions (CSF) of the same symmetry

$$|\psi_\alpha(PJM)\rangle = \sum_r^{n_c} c_r(\alpha) |\gamma_r P J M\rangle. \quad (1)$$

Often, large wave function expansions are required to obtain a sufficient accurate description of atomic

levels and properties. Expansions with a size, n_c , of several thousand and up to a hundred thousand CSF are nowadays applied in individual investigations. In its present form, RATIP first of all supports the computation of (relaxed) transition probabilities and lifetimes, the representation of symmetry-adapted functions within a determinant basis, the generation of continuum orbitals as well as calculations on Auger properties and photoionization cross sections. During the last two years, in particular, RATIP helped improve and enlarge the available data base on Einstein coefficients, oscillator strengths, and branching ratios for various multiple charged ions of the iron group. Fig. 1 displays the main structure of the RATIP package; a more detailed account of RATIP's recent (and potential future) applications has been given elsewhere [6].

The revised form of RATIP is divided into several groups for calculating level properties, bound-state transitions, (auto-) ionization parameters, and a few others tasks which are required when characterizing open-shell atoms. In this article, however, I will not give much information about these *main* components of the RATIP package [7–10] since a long write-up is usually available for most components (or will be prepared soon). These write-up's provide both, the theoretical background of the individual parts as well as reference to their recent applications. Here, instead, I present a set of UTILITIES for the RATIP program which facilitates its further application. Although these UTILITIES have mainly been developed for large computations (where often also a huge amount of data need to be processed), a number of similar tasks also

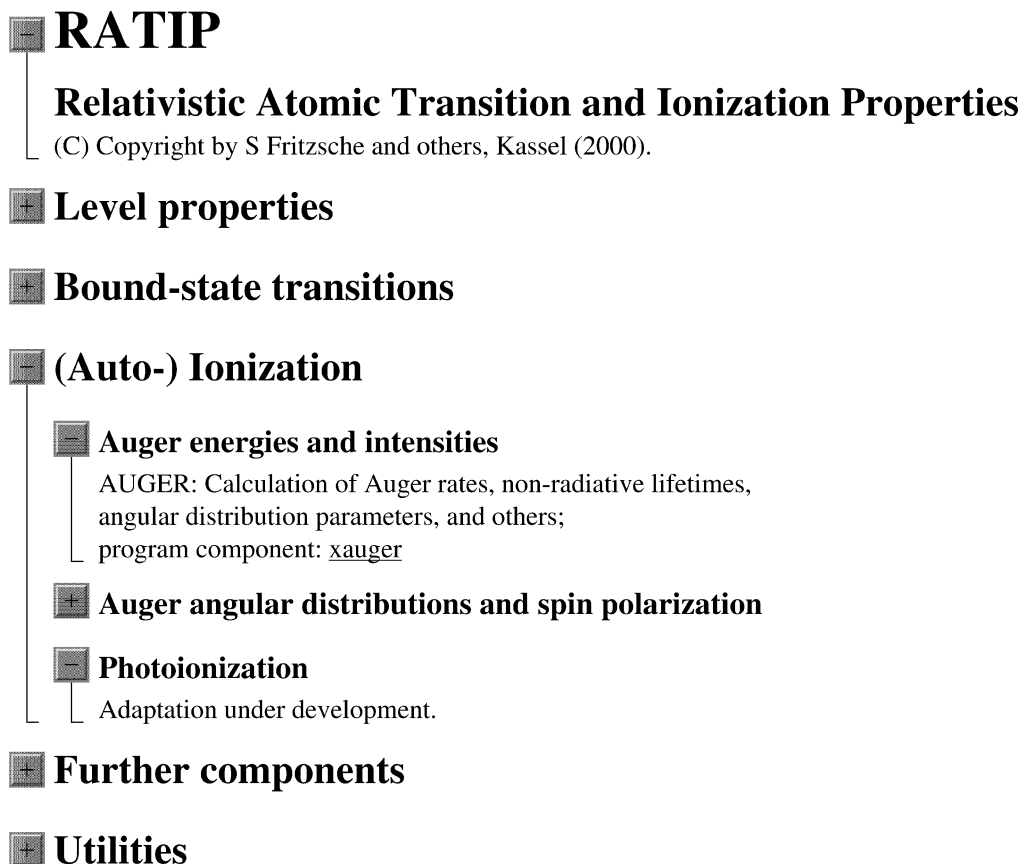


Fig. 1. Main structure of the RATIP package which is divided into several groups of programs. Details are displayed only for the calculation of Auger properties and photoionization cross sections.

occur in daily work when one looks, for example, for a fast estimate of some atomic decay rate or cross section. The UTILITIES support a variety of such small tasks including a convenient printout of data, the generation of suitable configuration bases, the rapid file transfer between different architectures — if not all of the computations can be run on the same platform — as well as several others.

In its latest development, the *long-term maintenance* of the RATIP program has been a major concern. Our experience showed that a rather large effort is typically needed to keep a *software package* of this size alive over a longer period. This is why we now make use of a more object-oriented design and, in particular, of the new concepts of Fortran 90/95. This *up-to-date* language standard enables one to design programs which are very flexible with respect to

the size of the actual investigation. In the new design of RATIP [7], all arrays which are related to the representation of atomic states, grid sizes, Hamiltonian and transition matrices, and many others are allocated dynamically at run time. Therefore, there are virtually no other limitations than those by the resources themselves. Most components now support wave function expansions of several ten (or even hundred) thousand configuration functions and, thus, are comparable in size to the best nonrelativistic computations. Owing to the use of spin- and symmetry-adapted CSF in the GRASP92 and RATIP environments, such expansions are also comparable with the best *molecular computation* at present which are often based on Slater determinants.

In the next section, I will briefly sketch the basic structure of these UTILITIES which are designed

as an independent component of RATIP. In several subsections, then the various groups of this program are explained in greater detail. But, although, I will outline the *concept* of each individual task, not all dialogs and branches of the program can be displayed and explained in detail. Starting from the main menu of the UTILITIES, the user is guided through the program with the help of interactive notes and dialogs. In Section 3, I recall how the RATIP program (as a whole) is distributed. Finally, a short summary about the program is meant to encourage all those users, who want to apply RATIP, that they may contact us and may express useful extensions to the program.

2. Program structure

2.1. A short overview

Lately, a number of case studies on the *ab-initio* calculation of transition probabilities have shown that large-scale expansions (1) are typically required in order to obtain accurate predictions. This was found true,

in particular, for most medium and weak transitions which often appear more sensitive to correlation contributions than the strong resonance lines. Detailed investigations have been carried out, for example, for several astro-physically relevant spectra of the iron group elements [11–13] as well as for the spectra of Ni II [14] and Cu II [15].

Of course, large configuration expansions (1) do not only increase the effort which is required to generate the wave functions and to calculate the transition amplitudes and cross sections. The size of these expansions also strongly influences the number of steps which have to be carried out during the computations, and how these wave functions need to be maintained. In practice, large expansions also result in a sizeable amount of intermediate and output data and, hence, an efficient computation of such expansions becomes almost impossible without the appropriate tools.

In order to *facilitate* large-scale applications, the UTILITIES of the RATIP package supports a number of small but frequently occurring tasks. Fig. 2 shows

Select from the menu:

- 1 - Energy levels and level splittings (simple).
- 2 - Energy levels and level splittings (extended).
- 3 - Display the major CSF and their weights to atomic levels.
- 4 - Display the 'names' of major LS terms (inactive).

- 5 - Exclude a number of CSF from a GRASP92 .csl list.
- 6 - Generate a CSF pair-list due to a given .csl reference list.
- 7 - Merge two .csl lists from GRASP92.
- 8 - Condense a .csl list from GRASP92 (standard).
- 9 - Condense a .csl list from GRASP92 (extended).

- 10 - Format .mix and .out files from GRASP92.
- 11 - Unformat .mix and .out files from GRASP92.

- 12 - Radial properties of atomic one-electron orbitals.
- 13 - Overlaps between two not quite orthogonal orbital sets.
- 14 - Effective radial charge or charge density of a selected ASF.

- 15 - Calculate branching ratios from REOS99 output.
- 16 - Generate transition probability tables from REOS99 output.

- q - Quit the program.

Fig. 2. Main menu of the xutility program.

the main menu of this program from which five groups of tasks can be distinguished. These *groups* will be explained in more detail below. A particular task is selected from this menu by entering the corresponding number at the prompt. Then, individually for each task, a short description appears on the screen summarizing the overall purpose and its basic requirements (concerning data and files). This is followed by an interactive dialog which leads the user through the selected task. For the sake of brevity, it is impossible to display all the different dialogs here; only a few examples can be shown below. However, the interactive set-up of the program should make it easy for the user to grasp the capabilities of the various subtasks.

Most tasks in Fig. 2 require one or several input files which must be provided by the user. From these files, all required information is obtained. Within the framework of GRASP92 and RATIP, various file formats need to be distinguished to set-up an useful interface between different components. These files describe either a CSF basis (*.csl* files) or they provide a representation of selected atomic states (within this basis) in terms of the configuration mixing coefficients (*.mix* files) and/or the radial orbital functions (*.rwf*). These three file formats were originally defined as *interfaces* between the program components of GRASP92 but they are utilized also for the RATIP package. In the description of the individual tasks below, I will often just refer to this short-hand notations (*.csl*, *.mix*, ...) while the relation among these files should be clear from the context.

2.2. Energy spectrum and main wave function components

When dividing large-scale computations into smaller, manageable parts, it is often useful to consider only a single *group* of levels with the same total angular momentum and parity J^P at a time, i.e. independently of other such groups. For these level groups, the Hamiltonian matrix is block-diagonal due to the symmetry of free atoms with respect to a rotation or inversion of the coordinates. Following such a procedure, however, the generated wave functions (from either the GRASP92 or RATIP output files) also refer only to the levels and wave functions of the corresponding group. In order to eventually combine the results from different computations again, the first two tasks [cf. Fig. 2] comprise

the results from one or several input files (back) into a single output. For this purpose, task 1 ('simple' version) displays the total energies of all levels (in a.u.) in ascending order. It also lists the excitation energies with respect to the lowest levels. The 'extended' form (task 2), in contrast, supports — if required — a reversed order of energies, different (energy) units to display the results as well as the selection of a particular *reference level* in order to list 'excitation' energies. Both tasks (1 and 2) prompt for the input of one or several *.mix* mixing coefficient files from which all the information is obtained; these files can be given either in a *formatted* or *unformatted* form — just as appropriate for the user. Fig. 3 shows the interactive dialog and the output from the extended form. In the printed output of these tasks, the *history* of the individual levels, if more than one *.mix* file was given, is kept by a pair of integers, for instance [3/2], which refers to the third ASF as described in the second *.mix* mixing coefficient file.

Two other tasks (3 and 4) of this group refer to the leading configurations and the (spectroscopic) *LS* notation of selected levels. In task 3, details about the leading terms are displayed by the weights and the particular form of the few major CSF. The (maximum) number of leading CSF (default 5) and the individual atomic levels (ASF; default all) can be selected interactively. Task 3 requires a *.csl* file which describes the CSF basis and a corresponding *.mix* file. The leading CSF in the expansion are displayed in exactly the same form as kept within the configuration symmetry list (*.csl*) file. As an example, Fig. 4 shows the output for the lowest two $^2P_{1/2,3/2}$ levels of the ground-state configuration of single-ionized Ne⁺ ions.

For the interpretation of many spectra and, in particular for inner-shell excitations, it is often inevitable to know the spectroscopic *LS*-notation of the leading terms. This not only supports the identification of lines in experimental spectra but also facilitates the communication with spectroscopists. For most cases, the *LS*-designation needs to be known for all the parent states (of the core shells) as well as the intermediate and the overall (nonrelativistic) *LS* symmetry. The derivation of such proper *LSJ names* from relativistic computations requires for a particular level the complete expansion of all leading CSF within a nonrelativistic basis; this feature (task 4) is still *inac-*

Returns the energy levels and excitation energies of some ASF as given by one or several mixing coefficient files. For printing excitation energies, the energy units, an ascending/decending order, and the level, relative to which the excitation energies are taken, can be selected.

Which units are to be used to print the results ?

A : Angstrom;
 eV : electron volts;
 Hartree : Hartree atomic units;
 Hz : Hertz;
 Kayser : [cm**(-1)];

eV

Energies are printed in ascending order; use a decending order instead ?

n

Enter the level number of the reference (zero-) level:

4

Enter one (or several) .mix mixing coefficient file(s):

file1.mix file2.mix

| Level | J | Parity | History [File / [Level]] | Total energy (Hartrees) | Level splitting (eV) | Excitation energy from the reference (eV) |
|-------|-----|--------|--------------------------------|-------------------------------|-----------------------------|--|
| 1 | 1/2 | + | [1, 1] | -1.27030939E+02 | 0.00000000E+00 | -6.41505944E+00 |
| 2 | 1/2 | + | [1, 2] | -1.27006045E+02 | 6.77400497E-01 | -5.73765895E+00 |
| 3 | 1/2 | + | [1, 3] | -1.26933438E+02 | 1.97573784E+00 | -3.76192110E+00 |
| 4 | 7/2 | + | [2, 1] | -1.26795190E+02 | 3.76192110E+00 | 0.00000000E+00 |
| 5 | 7/2 | + | [2, 2] | -1.26790137E+02 | 1.37499185E-01 | 1.37499185E-01 |
| 6 | 7/2 | + | [2, 3] | -1.26788161E+02 | 5.37697189E-02 | 1.91268904E-01 |
| 7 | 1/2 | + | [1, 4] | -1.26781710E+02 | 1.75540717E-01 | 3.66809621E-01 |

Fig. 3. Interactive dialog and typical output from task 2.

tive in the current version of the UTILITIES. However, ways and methods to display the full *LSJ* designation (of the leading terms) are presently being developed [16,17].

2.3. Generation of CSF lists

The choice of the many-electron CSF basis eventually determines the accuracy of *ab-initio* predictions which can be achieved for some property. A second group of tasks of the UTILITIES therefore supports the deliberate generation and manipulation of .cs1 lists as appropriate for a given problem. Of course, only little insight into the ‘physics’ of some particular shell structure would be needed, if the size of the basis could be increased arbitrarily. In practice, how-

ever, the size which is still manageable is limited by the available computer resources. Therefore, the aim must be to include the physically most important contributions in the configuration mixings while others are neglected. Till now, unfortunately, not much experience is available for most properties of free atoms and ions, that means *experience* at which level certain (virtual) excitations and mixing contributions become negligible. In the computation of transition probabilities, there have been cases where a mixing of less than 10^{-4} in the wave function expansion has modified the rates of weak transitions (and ‘shifted’ them towards the experimental data) by more than a factor of 2.

In standard computations, correlation contributions are often classified in terms of virtual excitations. Different *classes* of such excitations like singles (S), dou-

Weights of major contributors to ASF:

| Level | J | Parity | CSF contributions | | | | | | | | |
|-------|-----|--------|-------------------|----|-----|---------|----|-----|---------|----|-----|
| 1 | 1/2 | + | 0.41667 | of | 131 | 0.35282 | of | 27 | 0.22322 | of | 1 |
| 2 | 1/2 | + | 0.64037 | of | 27 | 0.22907 | of | 131 | 0.12396 | of | 1 |
| 3 | 1/2 | + | 0.87192 | of | 196 | 0.05030 | of | 1 | 0.02995 | of | 131 |

Definition of leading CSF:

| | | | | | | |
|------|---------|---------|---------|---------|---------|------|
| 1) | 1s (2) | 2s (2) | 2p-(2) | 2p (2) | 3s (1) | |
| | | | | 0 | 1/2 | |
| | | | | | | 1/2+ |
| 27) | 1s (2) | 2s (2) | 2p-(1) | 2p (3) | 3s (1) | |
| | | | 1/2 | 3/2 | 1/2 | |
| | | | | 1 | | 1/2+ |
| 131) | 1s (2) | 2s (2) | 2p (4) | 3s (1) | | |
| | | | | 1/2 | | |
| | | | | | | 1/2+ |
| 196) | 1s (2) | 2s (1) | 2p-(2) | 2p (4) | | |
| | | 1/2 | | | | |
| | | | | | | 1/2+ |

Fig. 4. Example output from task 3.

bles (D), and/or triples (T), ... can be distinguished by starting from a given set of *reference configurations*. Although the generation of such basis lists is supported by the GRASP92 component `gensl`, it typically leads to a dramatic increase of the number of CSF in the final wave functions expansions. Depending particularly on the shell structure of the atom or ion, CSF expansions of (hundreds of) thousands or even millions can easily be derived. By far not all of these CSF are equally important, though. In order to (de-)select certain subclasses of excitations, task 5 of the UTILITIES enables the user to *exclude* certain excitations from a given `.csl` list by defining a set of 'restrictions'. Fig. 5 shows the basic form of such restriction rules; they mainly refer to the (allowed) occupation in one or several of the (pre-selected) subshells. After one or several of these rules have been specified by the user, the program determines those CSF which will remain in the list and displays their number on screen. The user now has to confirm or to discard this selection before *additional* rules can be applied. This branch of the UTILITIES has been found helpful to quickly reduce the size of large CSF lists, starting from a well defined set of double or triple excitations. The interactively supported format of these

rules allows a fast and well-suited adjustment to the user's option.

Another (physical) argument for *reducing* the size of a configuration basis results from the Hamiltonian matrix as defined in this basis. Often only those CSF are found important in the wave function expansions of some atomic (or molecular) states, at least for the high-lying excitations, which are directly coupled to a set of (dominant) *reference* configurations, i.e. have nonvanishing matrix elements of the Hamiltonian. In the respective literature, this approximation is sometimes known as *pair-correlation* approach which, in relativistic theory, can be based on either the Dirac–Coulomb or Dirac–Coulomb–Breit Hamiltonian. Task 6 of the UTILITIES supports a reduction for either of these Hamiltonians by evaluating the angular coefficients of the corresponding matrix elements. It requires the input of two `.csl` lists, the CSF basis which is to be reduced and the *reference* basis. For evaluating the angular integrals, the new component ANCO [10] of the RATIP package is invoked and supports a fast evaluation. A new implementation of the angular coefficients became necessary to allow for large-scale wave function expansions and a more time-efficient set-up of the Hamiltonian and Auger transi-

Reduce a given .csl list from GRASP92 due to a set of 'restriction rules' which can be supplied interactively. These 'rules' can be defined in several ways; the general form of these rules is:

$$\text{subshell}_1 + \text{subshell}_2 + \dots + \text{subshell}_n \quad \text{rel_op} \quad N$$

where subshell_i is 1s, 2p, ..., rel_op denotes a relational operator, i.e. one of {<, <=, <\$, =, >, >=}, and $N (> 0)$ is the number of allowed particles in these subshells. A * denotes all subshells of a given layer; each restriction rule must be given on a separate line and may contain only peel shell orbitals. A <\$ means that all shells of this restriction must contribute to it with an occupation > 0 .

A few examples are:

$$4s + 4p^- + 4d \leq 3 \quad \text{or} \quad 5* < 2 \quad \text{or} \quad 3p + 3d > 12$$

$$4s + 4p < \$ 2 \quad \dots \text{ to omit } 4s(1) \ 4p(1) \text{ but not } 4s(2) \text{ or } 4p(2).$$

Use a nonrelativistic notation for the orbitals ?

Fig. 5. Concept and restriction rules of task 5.

tion matrices, respectively. Our new implementation is based on second quantization which is applied to both, the angular momentum and the quasi-spin space [18]. Unlike the former GRASP92 component MCP, ANCO now supports open subshells with $j = 9/2$ and has become faster by about a factor of 5. Since one usually considers only a rather small number of reference configurations, such pair-correlation lists can be generated very rapidly.

Task 7 of the toolbox can be invoked to *combine* two or more .csl lists into a single CSF basis. This task is of interest, for instance, if triple or quadruple excitations are to be taken into account within the valence shells while only singles and doubles are considered for all additional correlation layers. In this case, it is most convenient to generate two .csl lists independently and which are later combined into a single set. To keep the basis linear independent, of course, all doubly defined CSF need to be recognized (and removed) from the basis. A similar *file-merging* is provided by the GRASP92 component *mr_gcs1*; this program, however, requires the subshells to be defined in a particular order and to occur in both files in the same sequence. Moreover, *mr_gcs1* often requires a lot of time if large .csl lists need to be combined since each *additional* CSF, which is to be appended, has to be compared with all previously tested CSF. Both shortcomings of *mr_gcs1* have been

overcome in task 7; a much faster evaluation is now achieved by 'packing' the occupation and coupling information of all CSF into a particular array which is used for a first quick comparison, i.e. all the details about the particular coupling of the CSF are not longer required unless they lead to the same 'cross structure'. Concerning the set of active orbitals in the two .csl lists, task 7 only requests the same core and the same sequence for those subshells which occur in both lists. To avoid also these two *assumptions* would generally require a full re-coupling of the CSF and, thus, exceeds by far the scope of application of these UTILITIES.

Two further tasks (8 and 9) of this group enable the user to *condense* a .csl list in line with a corresponding .mix mixing coefficient file. Of course, this necessitates the previous set-up and diagonalization of the Hamiltonian matrix as obtained, for instance, from RELCI. The standard form (task 8) of such a *condensation* of the basis removes all CSF with a weight which is smaller than a given cutoff (input) parameter. In this standard form, the mixing contributions of all ASF which are defined by the .mix file are also taken into account and compared to the critical weight. Apart from the two file names (.csl and .mix) and the critical weight, no further input is needed for this particular task. An extended form (task 9), in contrast, allows to *select* one or more ASF for the condensation of the basis. Moreover, this task also supports the genera-

tion of a *condensed* .mix file, i.e. an approximate representation of the ASF in the condensed basis. In this new .mix file, the eigenvectors are ‘re-normalized’ but, without the recalculation of the Hamiltonian matrix in this basis, are of course not completely orthogonal to each other. For a small cutoff, however, this *condensed* representation of the ASF still provides a rather good approximation and *saves* the user to recalculate the Hamiltonian matrix.

2.4. File transfer

In the GRASP92 framework, the representation of the atomic levels often requires a rather large amount of disk storage (and also a frequent access during the SCF iteration). The representation of the ASF is therefore stored in *unformatted* form, concerning both the radial orbital functions as defined on the internal (logarithmic) grid as well as the mixing coefficients within the corresponding CSF basis. These unformatted files do not cause much problems as long as all computations are carried out on the same platform. (Note for large-scale computations, however, that the output to the .mix file leads to rather long records which may exceed the block lengths of some compilers and then may result in run-time errors.) But difficulties with the unformatted files mainly arise, if different parts of an application need to be carried out on different architectures. To facilitate the file transfer between the various architectures, tasks 10 and 11 support a *formatted* (ASCII) version of the .mix and .rwf files. While task 10 transfers *unformatted* files into *formatted* ones, task 11 does the opposite. Then, a file transfer is easily possible if the UTILITIES are installed and run on both platforms. In the RATIP package, moreover, the formatted form of the .mix and .rwf files is now also supported by all main components (CESD99, REOS99, RCFP, ANCO, ...) which have been adapted to the present standard. The particular form can be chosen in the source code of the individual components by a global flag. In `rabs_reos`, for instance, the file format of the .rwf files is determined simply by the boolean flag `reos_use_formatted_rwf_file` which is initialized to either `.true.` or `.false.` at the beginning of this file. To further simplify the use of the different formats, most tasks of the UTILITIES now recognize the format even automatically, a behaviour

which we will implement also for the other components in the future.

2.5. Charge distributions

To know the charge distribution of an atomic level not only provides insight into the (radial) extent of an atom or ion but also helps understand the excitation and decay characteristics of such systems. A well known example is the particular behaviour of atomic photoionization cross sections when the (radial) wave functions of the valence electrons ‘collapse’ towards the core *in course of* the rearrangement of the bound electron density. Other simple *physical pictures*, which are built on changes in the charge distribution of ions, are used in order to explain the *shake-up* or *shake-off* lines in Auger spectra, at least qualitatively. To gain such insight into the excitation and decay dynamics of atoms, three additional tasks (12–14) are designed to provide information about the properties of the radial (one-electron) orbitals and the spherical charge distribution of selected levels.

Task 12 displays the one-electron energies and the radial expectation values for a set of orbitals, obtained from a .mix files. Calling this task, the expectation values of the operators r^k are printed for the powers $k = -3, -1, 1, \text{ and } 2$, which are related to various physical operators in atomic structure theory. For two (different) sets of radial orbitals, moreover, task 13 evaluates and lists the nonvanishing *overlap* integrals as shown in Fig. 6. In the interpretation of Auger spectra, these overlap integrals are often taken to be proportional to the ‘shake’ probabilities, i.e. the probability for effective three- and many-electron processes which cannot simply be understood in an independent-particle model.

The effective radial charge or charge density of some selected levels, finally, is obtained from task 14. Here, the effective charge

$$Z_{\text{eff}}(R) = \int_0^R dr \rho(r) \quad (2)$$

is meant to be the integral over the electron charge density up to a given radius R . This effective charge has been applied to estimate the so-called ‘anti-screening’ in the projectile ionization of high- Z ions in collisions with light and medium target materials [19].

```

Overlap integrals <n_f kappa | n_i kappa> :
-----
+ s symmetry: < 1s | 1s > = 9.9998E-01    < 1s | 2s > = 5.9374E-03    < 1s | 3s > = 1.9881E-03
-----          < 1s | 4s > = -1.4363E-06    < 2s | 1s > = -5.9561E-03    < 2s | 2s > = 9.9984E-01
                  < 2s | 3s > = 7.3210E-03    < 2s | 4s > = -2.7667E-03    < 3s | 1s > = -1.9490E-03
                  < 3s | 2s > = -7.7342E-03    < 3s | 3s > = 9.9948E-01    < 3s | 4s > = 8.2346E-03
                  < 4s | 1s > = 6.7380E-06    < 4s | 2s > = 3.8549E-03    < 4s | 3s > = -7.1775E-03
                  < 4s | 4s > = 9.3590E-01

+ p- symmetry: < 2p- | 2p-> = 9.9969E-01    < 2p- | 3p-> = 9.2145E-03    < 3p- | 2p-> = -9.8028E-03
-----          < 3p- | 3p-> = 9.9953E-01

+ p symmetry: < 2p | 2p > = 9.9969E-01    < 2p | 3p > = 9.3748E-03    < 3p | 2p > = -1.0065E-02
-----          < 3p | 3p > = 9.9941E-01

```

Fig. 6. Output from task 13 which evaluates all nonvanishing overlap integrals of two sets of orbitals.

Task 14 requires the knowledge of the full wave function expansion and, thus, prompts the user for the configuration basis (.csl) and well as for the mixing coefficient (.mix) and radial orbital files (.rwf), respectively. Alternatively, this task can be utilized to obtain the charge density of a particular atomic level, given on the internally defined grid and written to some output file. Although these tasks may provide only *small* advantages for the user, they have proved very helpful in a number of case studies.

2.6. Data compilations

The main components of the RATIP package support the calculation of various ionization and decay properties including transition probability and Auger computations. By default, these programs provide a tabulation of all results of the respective run which has been found useful in many applications. Usually, however, these summaries *suppose* that all relevant information were obtained within a single run of the program, an assumption which often contravenes the requirements of large-scale computations and of mass productions of data. As mentioned before about such large computations, it may be better then to divide the atomic levels (and, thus, also the wave functions) into groups of the same J^P symmetry in which the Hamiltonian matrix is diagonal. To arrange later the results from independent calculations into suitable compilations, the output of different runs of the main components (for instance, REOS99) has to be reworked. In the present UTILITIES, there are two tasks which partially support an appropriate data compilation; other tasks are expected to be implemented as a respective

demand emerges. These tasks are not fully ‘developed’ yet in the sense that it will take more feedback and experience from the side of the users in order to provide a comfortable environment. Nevertheless, here I like to introduce them in order to encourage further stimulation from outside.

Although the *basic idea* of such a data compilation group is easy to express, its implementation is somewhat more cumbersome and often requires information which is not (yet) provided by the main program components. One example are the branching fractions in optical spectra. Apart from the wavelengths, the transition probabilities (or, equivalently, oscillator strengths) are of course the basic entities to describe such spectra. More frequently, however, intensity ratios are measured, for which the theoretical derivation typically requires information about *more* than two levels (not necessarily included in the same run). To facilitate the derivation of branching ratios for radiative decay spectra, task 15 provides a tabulation of all A values and branching fractions as obtained from the output of one or more .trn transition amplitude files from REOS99. These files contains all transition energies and amplitudes from a single run of the program.

Task 16, finally, serves a similar purpose. This task may provide a tabulation of transition probabilities and lifetimes from one or several REOS99 computations. Apart from the A values, as default, it enables one to select, additionally, one equivalent property like oscillator strengths, line strengths, line widths, etc. in order to keep the tabulation in a reasonable (and printable) size.

Other comparable tasks might get implemented as more computations in RATIP's framework are carried out in the future. Since demands similar to those for studying transition probabilities with REOS99 also arise in the computations of Auger spectra, further tasks will be appended to this section in the future. In Auger electron spectroscopy, for example, it has been common practise to provide entities (or branching ratios) relative to a particular *strong line*. For such tasks, additional information need to be read in by the UTILITIES.

3. Distribution of the program

The UTILITIES for the RATIP program have been developed in parallel with other components and, therefore, follows the same *design* principles. The present program is based on an enlarged interactive part in order to gather the information for the particular subtask which is invoked by the user. Although the UTILITIES are by themselves independent of other components, at source code level, they also utilize many of their (module) procedures which were part already of earlier releases. Overall, 17 modules are *used* by the present program, linking together more than 25,000 lines of code within about 500 procedures. Because of this tight binding of the UTILITIES to the main components, they will be distributed along with the RATIP package as a whole.

The full source, including the two libraries `dvdson` and `lapack`, is contained in the `ratip` root directory. This root also includes a number of makefiles for generating the executables as well as test suites for most components. To ensure a high portability of the whole package, the *script* file `make-environment` has lately been introduced to *declare* a number of global variables about the compiler and its (optional) compiler flags. By having defined these global variables, the user does not need to adapt the makefiles for each component of RATIP independently. The script `make-environment` only contains a very few lines which have to be adapted to the current environment; then, this script has to be *sourced* after any modification to it before the other makefiles can be invoked properly. An executable of each of the components is obtained by typing in the command `make -f make-component`;

here, component can be `anco`, `cesd99`, `rcfp`, `relci`, `reos99`, `utilities`, ... (and others in the future). The present version has been tested and found portable to different platforms including the IBM RS/6000 and the PC Linux world.

The *interactive* character of the UTILITIES makes it unnecessary to provide a test suite in addition to the few examples which were shown above. During the installation of the program, the proper file handling is the most error-prone part of the program, but this can easily be checked during the execution.

In conclusion, the UTILITIES of the RATIP program provides a useful toolbox which helps the user through a variety of frequently needed tasks. The majority of these tasks arose originally during several large-scale calculations which have been carried out with this package. They often support a flexible handling of data at some intermediate step of the computations. It is therefore likely that further, similar requirements will occur as the number of users (and research subjects) increase. One of the next step in developing RATIP will be the full adaptation of the program components which help calculate Auger rates and photoionization cross sections. These components have been tested in a number of computations and will be released soon.

The main menu of the UTILITIES can easily be extended to include additional tasks. At present, emphasis is paid to derive and provide the *LSJ* notation for some (given) atomic level. From several studies of resonantly excited Auger spectra it was found, that a proper spectroscopical notation helps analyze such spectra (and, sometimes, may even resolve misassignments) since the theoretical level structure by itself, i.e. the level ordering due to their total energies, is often not accurate enough if inner-shell excitations occur. Further developments are also expected for the data compilations as such requirements will arise by the users of RATIP.

Acknowledgement

I like to thank Drs. Wolfgang Wiese und Yong-Ki Kim for helpful discussions and also acknowledge the hospitality of the National Institute of Standards and Technology, Gaithersburg.

References

- [1] R.D. Cowan, *The Theory of Atomic Structure and Spectra*, University of California Press, Berkeley, 1981.
- [2] C. Froese Fischer, *Comput. Phys. Commun.* 128 (2000) 635.
- [3] J.P. Desclaux, *Comput. Phys. Commun.* 9 (1975) 31.
- [4] I.P. Grant, B.J. McKenzie, P.H. Norrington, D.F. Mayers, N.C. Pyper, *Comput. Phys. Commun.* 21 (1980) 207.
- [5] F.A. Parpia, C. Froese Fischer, I.P. Grant, *Comput. Phys. Commun.* 94 (1996) 249.
- [6] S. Fritzsche, *J. Elec. Spec. Rel. Phen.* 114–116 (2001) 1155.
- [7] S. Fritzsche, C. Froese Fischer, C.Z. Dong, *Comput. Phys. Commun.* 124 (2000) 343.
- [8] S. Fritzsche, J. Anton, *Comput. Phys. Commun.* 124 (2000) 354.
- [9] G. Gaigalas, S. Fritzsche, *Comput. Phys. Commun.* 134 (2001) 86.
- [10] G. Gaigalas, S. Fritzsche, I.P. Grant, *Comput. Phys. Commun.* 139 (2001) 263.
- [11] S. Fritzsche, C. Froese Fischer, B. Fricke, *At. Data Nucl. Data Tables* 68 (1998) 149.
- [12] C.Z. Dong, S. Fritzsche, B. Fricke, W.D. Sepp, *Monthly Notices Roy. Astron. Soc.* 307 (1999) 809.
- [13] S. Fritzsche, C.Z. Dong, E. Träbert, *Monthly Notices Roy. Astron. Soc.* 318 (2000) 263.
- [14] S. Fritzsche, C.Z. Dong, G. Gaigalas, *At. Data Nucl. Data Tables* 76 (2000) 155.
- [15] C.Z. Dong, S. Fritzsche, B. Fricke, *J. Elec. Spec. Rel. Phen.* 114–116 (2001) 157.
- [16] G. Gaigalas, S. Fritzsche, in preparation.
- [17] S. Fritzsche, *Comput. Phys. Commun.* 103 (1997) 51; S. Fritzsche, S. Varga, D. Geschke, B. Fricke, *Comput. Phys. Commun.* 111 (1998) 167; G. Gaigalas, S. Fritzsche, B. Fricke, *Comput. Phys. Commun.* 135 (2001) 219.
- [18] G. Gaigalas, Z. Rudzikas, C. Froese Fischer, *J. Phys. B* 30 (1997) 3747.
- [19] S. Fritzsche, T. Stöhlker, *Phys. Scr.* 92 (2001) 311.