



REOS99: A revised program for transition probability calculations including relativistic, correlation, and relaxation effects

S. Fritzsche^{a,1}, C. Froese Fischer^b, C.Z. Dong^a

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

^b *Computer Science Department, Vanderbilt University, Nashville, TN 37235, USA*

Received 20 July 1999; accepted 9 August 1999

Abstract

Recent years have seen an increasing demand on accurate transition probability calculations for open-shell atoms and ions. To facilitate such investigations, the program REOS [S. Fritzsche, C.F. Fischer, *Comput. Phys. Commun.* 99 (1997) 323] has been developed which includes the effects of relativity, correlation, and the rearrangement of the bound-state electron density within the framework of the multiconfiguration Dirac–Fock model. Often, however, very large wave function expansions are indeed needed to obtain sufficiently accurate results which cannot be handled by the original program. Shortcomes of this program concern in particular the demand on memory and CPU time as well as its tight binding to the IBM XL Fortran standard. To resolve these drawbacks we present here a revised version, REOS99, which has been developed in line with the ANSI standard Fortran 90/95. Care has been taken to exploit the advantages of this new standard and to improve the efficiency and portability of the program. © 2000 Elsevier Science B.V. All rights reserved.

NEW VERSION SUMMARY

Title of program: REOS99

Catalogue number: ADLF

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

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland. Users may also down-load a tar-file of the program `ratip99.tar` from our home page at the University of Kassel (<http://www.physik.uni-kassel.de/fritzsche/programs.html>)

Reference to original program: REOS [1]; catalogue number ADEJ

Authors of the original program: S. Fritzsche and C. Froese Fischer

Does the new version supersede the original program: Yes

Licensing provisions: None

Computer for which the new version has been tested: IBM RS 6000, PC Pentium II

Installations: University of Kassel (Germany)

Operating systems: IBM AIX 4.1.2+, Linux 6.1

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

¹ E-mail: s.fritzsche@physik.uni-kassel.de.

Memory required to execute with typical data: 5–20 MB

No. of bits in a word: All real variables are parameterized by a selected kind parameter and, thus, can easily be adapted to any required precision as supported by the compiler. Presently, the kind parameter is set to double precision (two 32-bit words) in the module `rabs_constant`

Peripheral used: Disk for input/output

CPU time required to execute test data: 78 minutes on a 266 MHz Pentium II processor

No. of bytes in distributed program, including test data, etc.: 4943 716 (please there is one file for the programs described in this paper and in Ref. [3]).

Distribution format: unencoded compressed tar file. This tar file includes the source code of about 12,000 lines, makefiles to generate the executables as well as test suites to check the installation

Keywords: Atomic, determinant basis, Einstein coefficients, electron rearrangement, lifetimes, multiconfiguration Dirac–Fock, relativistic, oscillator strength

Nature of the physical problem

Transition probability calculations for (open-shell) atoms have shown that large wave function expansions are often needed to obtain sufficient accurate results. For *ab initio* calculations, these wave functions must include the effects of relativity, electron correlations, and the rearrangement of the electron density within a common framework. The REOS99 program accounts for all of these effects for the computation of Einstein coefficients, oscillator strengths, and (radiative) lifetimes. It applies relativistic wave functions from the GRASP92 package [2] which need subsequently to be expanded into a determinant basis [3] in order to apply them to REOS99.

Reasons for the new version

Wave function expansions based on several ten thousand determinants and the computation of large transition arrays made a number of major modifications in the source code mandatory. Thereby, considerable improvements have been achieved by taking advantage of the new ANSI Fortran 90/95 standard. Many powerful features of this up-dated language like modules, derived data types, dynamic

allocation and deallocation of memory, and others have been exploited in order to develop a revised version which is forearmed for transition probability calculations into the next decade. Efficiency has been gained in particular due to a *bitwise* storage and treatment of the occupation numbers of the determinants and due to a parallel computation of all transitions and multipole components. The *bitwise* representation of occupation numbers is maintained by using the bit manipulation procedures of the Fortran 90/95 standard.

Restrictions onto the complexity of the problem

None. A few minor limitations arise for atoms with an open *f*-shell from the available list of the coefficients of fractional parentage (cfp) in GRASP92 which has fully been implemented only for subshells with $j \leq 7/2$.

Unusual features of the program

REOS99 is designed for interactive use. Following the interactive dialog at the beginning of the execution, a number of optional branches can be selected for the computation of transition probabilities. Two important branches are the *pre-calculation* of all radial integrals (as default) and an *assumed orthogonality* between the orbitals of the initial and final atomic states. Of course, the latter branch neglects a large part of the rearrangement effects of the bound-state density since all “overlap” contributions are omitted from the computations; however, this branch may result in a faster execution and may therefore be attractive for certain applications. A further acceleration has been achieved by introducing a cut-off parameter to neglect the contributions of the (many-electron) matrix elements for those pairs of determinants for which the product of the mixing coefficients is smaller than a given value; the default cut-off is set to 10^{-8} but can be overwritten interactively. In addition, individual transitions which need to be specified by the level numbers of the initial and final atomic states can be *selected*. The REOS99 program has entirely been rewritten in line with the ANSI standard Fortran 90/95 [4]; in particular all language features which are considered to become obsolete by the standard have been avoided.

References

- [1] S. Fritzsche, C.F. Fischer, *Comput. Phys. Commun.* 99 (1997) 323.
- [2] F.A. Parpia, C.F. Fischer, I.P. Grant, *Comput. Phys. Commun.* 94 (1996) 249.
- [3] S. Fritzsche, J. Anton, *Comput. Phys. Commun.* 124 (2000) 353.
- [4] M. Metcalf, J. Reid, *Fortran 90/95 Explained* (Oxford University Press, 1996).

LONG WRITE-UP

1. Experiences with accurate transition probability calculations

Ab initio studies of excitation and decay properties for open-shell atoms accentuate the endeavors which are often necessary for modeling *true* many-particle systems. While term and excitation energies can for many

atoms be predicted with admissible effort, most transition properties and cross sections, in contrast, appear to be rather sensitive to details of the theoretical model. As a well-known consequence, large-scale computations are then frequently required in order to obtain useful theoretical data. However, not many program environments are available so far which would enable this type of computation. To improve this situation, here we present REOS99, a revised program for the study of radiative transition probabilities. This program now supports much larger representations of the atomic wave functions than in a previous version [1] and, thus, might help to obtain a better understanding of the convergence of results as the size of the computation is increased. In addition, these developments may also lead to more reliable error estimates as needed for experimental analysis and interpretations.

To support the calculation of various (relativistic atomic) transition and ionization properties within a common framework, we developed the program RATIP [1,2] during the last decade. An increasing demand of transition and ionization data has recently been observed in astro- and plasma-physics for atoms and ions along different isoelectronic sequences. To generate useful data also for multiple and highly-charged ions, of course, the effects of relativity, correlation, and of the rearrangement of the electron density, which occurs when an atom undergoes a transition, must be treated consistently. For this reason, RATIP has been built on the wave functions of the atomic structure package GRASP92 [3], an implementation of the (relativistic) multiconfiguration Dirac–Fock (MCDF) model. Apart from the computation of transition probabilities and radiative lifetimes (using REOS99), RATIP, in addition, comprises the representation of symmetry-adapted functions from GRASP92 in a determinant basis, the generation of continuum orbitals as well as the computation of Auger rates and photoionization cross sections. Though the latter three components still need to be modified in order to bring them to the present standard, we intend to implement the “rearrangement effects” also for those processes which include (one) electron in the continuum.

The capabilities of RATIP in studying transition and autoionization properties have been demonstrated in a series of case studies (cf. [4,5] and references therein) during the last few years. But at the same time, our experiences in working with RATIP showed that a major revision of the package would be desirable for several reasons. There is first of all the obstacle that (i) most components of RATIP cannot be applied to wave function expansions of more than maximally a few thousand configuration state functions (CSF) which is not sufficient for many investigations. Also, there has been (ii) a tight binding of all components to the IBM XL Fortran standard, very similar to GRASP92, which, in the past, made it difficult to transport the code to other environments and, thus, (iii) to maintain the package over a long term. This binding complicated in particular the adaptation of further components to the present standard. Moreover, (iv) static allocation was used for representing several important arrays so that frequently an adaption of the source code to the respective problem needed to be performed.

To resolve these obvious drawbacks of the RATIP package, we started with REOS and re-wrote this component according to the ANSI Fortran 90/95 standard. This finally led to an entirely new implementation which makes use of modules, derived data structures, and many other features as suggested by this standard. Care has been taken to exploit the advantages of this standard as we will explain below; for the moment let us just stress that these developments clearly simplify the maintenance and distribution of REOS99 (and of the other components of RATIP) and, thus, can be considered as an important step to make RATIP ready for forthcoming years. Indeed, large-scale computations of a few ten thousand CSF are now easily feasible and this number will probably even be enhanced by more powerful computers in the near future. Computations of that size may not only enable the prediction of reliable probabilities for atoms and ions with open d - or even f -shells but will also allow a convergence analysis and the study of correlation effects for a variety of non-radiative processes.

Since the theoretical background of the REOS program has been explained before [1], we will not repeat many details again. In the next section instead, we shall give a short overview of the RATIP package and then describe the main modifications and the structure of the revised program. Here, we also explain how the program is to be distributed. To provide a test of the installation in Section 3, a few selected lines from the low-lying excitation spectrum of Fe X are calculated. Finally, a brief outlook is given in Section 4. We believe that with REOS99 (and its accompanying component CESD99) we worked out one of the first consistent implementations in atomic structure theory using the powerful features of Fortran 90/95. These revised components may therefore serve also as a starting point for further developments in this field.

2. Program structure and modifications

2.1. Overview to the RATIP package

If compared with previous standards, Fortran 90/95 can be considered as a rather new language that has benefited from Fortran's long tradition and from recent experience in designing modern languages [6]. Two important features of Fortran 90/95 (among others), in particular, has brought the language up-to-date with its alternatives: These features concern *modules* for generating protected and re-usable code as well as *derived types* which – along with pointer arrays in the current standard – enables the set up of very flexible data structures. In the new design of RATIP, we made elaborate use of these two concepts and of other features of Fortran 90/95 which enable a more object-oriented style for developing codes.

Though the RATIP package is currently built upon 15 modules not all of them need to be provided at the present level. REOS99 just applies 8 of these modules; of course, several of these modules will also be *used* by other components like, for instance, the module `rabs_constant` which contains the parameterization of all elementary data types as well as the definition of a few important and sufficiently general data types. This module is therefore *used* by all other modules.

Several derived types are defined in RATIP to build data types at different levels. A proper choice of such derived types in particular can facilitate and influence the further development of the code. We will focus on this point for a short moment in order to provide a better understanding of the present design for the reader. For example, two derived types at a rather low level concern a `determinant` and a `cs_function`, i.e. a symmetry-adapted CSF. While these data types may be declared like structured variables on their own, they may also form components of derived types at some higher level. Just in this sense, an allocatable array of `type(determinant)` forms for instance a component of a variable of `type(asf_det_basis)` which keeps all information about a whole determinant basis as well as about the internal representation of the atomic states in this basis. Another such example is the derived `type(reos_transition)` which, along with all its substructures, is specified by

```

type :: multipole_line
  character(len=2) :: multipole
  character(len=9) :: gauge
  real(kind=dp)    :: amplitude
  real(kind=dp), dimension(:, :, :, :), pointer :: radial_int
end type multipole_line
!
type :: reos_transition
  integer          :: asfi, asff, level_i, level_f, totalJ_i, totalJ_f
  integer          :: number_of_mlines
  character(len=1) :: parity_i, parity_f
  real(kind=dp)    :: energy
  real(kind=dp), dimension(:), pointer :: bessel0, bessel1, bessel2, &
                                         bessel3, bessel4, bessel5
  type(multipole_line), dimension(:), pointer :: mline
end type reos_transition

```

Obviously, the meaning of most *components* of these two derived types follows immediately from their names and needs not to be explained in great detail. The kind parameter `dp` for the parameterization of real variables in these data types is set to *double precision* on a standard 32-bit architecture in the module `rabs_constant`. In the declaration of `type(reos_transition)`, a pointer array of `type(multipole_line)` occurs as *one* of the components of the composite object; this intends to carry the information about the individual multipole contributions of the given transition. But while the structure of this component is completely specified, memory is

allocated to it only according to the (interactively) selected number of multipoles and gauge forms. In defining such derived types, some information like the parities and angular momenta of the atomic levels in the case above is redundant in the sense that the same data are stored also elsewhere in the program. However, by gathering and initializing related information in a separate type, one can conveniently access these data in any subprocedure which has an argument of the corresponding type. As seen from the declaration above, the data structure `type(reos_transition)` contains in addition several (allocatable) pointer arrays for storing Bessel functions; again, these arrays become only allocated and deallocated as they are actually needed for the computation of a multipole line. Thus, a whole transition array or any part of it can easily be handled within a program by specifying

```
integer :: number_of_transitions
type(reos_transition), dimension(:), allocatable :: transition
```

an array to which storage for the actual number of transitions is allocated only in course of the execution by

```
allocate(transition(1:number_of_transitions))
```

and can be deallocated later on.

The use of derived data types simplifies the development and maintenance of the code substantially. If, for example, *one* element of the array `transition`, say `transition(i)`, is an (incoming) argument to some procedure of the RATIP program, all information about this radiative line is easily accessible. A consistent application of this and similar derived types is one of the cornerstone which *prepares* RATIP for forthcoming years.

2.2. New capabilities and improvements

Several computational models for calculating transition probabilities and lifetimes are supported by REOS99. Apart from the *standard model* which includes the full relaxation of the bound-state electron density into the transition matrix, a second execution branch *assumes orthogonality* for the two sets of electron orbitals from the initial and final atomic states. This assumption is made independent of the fact whether all orbitals are mutually orthogonal to each other or not. In this approximation to the transition matrix, the “overlap” integrals are simply set to zero and, thus, the cofactors $D_{UV}(k|l)$ [cf. Ref. [1, Eq. (8)]] must either vanish identically or evaluate just to ± 1 . An *assumed orthogonality* simplifies the computation notably because all these cofactors can then be determined from the occupation numbers of the initial- and final-state determinants without calculating them explicitly. In particular for atoms and ions with more than, say, a few electrons this may accelerate the computation since not only the number of cofactors increases approximately as $(N - 1)^2$ with the number of electrons, N , but also the time to compute a single of these “overlap” determinants $D_{UV}(k|l)$ increases rapidly with its dimension. Note, however, that this branch neglects parts of the relaxation effects whose influence on the transition amplitudes may (strongly) vary from case to case.

Another new feature of REOS99 is the option to apply experimental energies for the calculation of transition probabilities. This option has been found very useful for studying transition probabilities of nearly neutral systems for which the fine structure of the (excited) atomic levels can often not be reproduced so accurately. By using the experimentally known energies for such systems, at least the energy dependence of the transition amplitudes can be taken into account properly for the different gauge forms.

Two further features concern the restart of a *previously aborted* calculation and the *pre-calculation* of all radial integrals. The restart of an aborted calculation is explained in the next subsection. Notably faster computations are achieved if the radial integrals for all selected transitions are calculated in advance just at the beginning of the execution, though this branch requires additional storage. Owing to the energy dependence of the radial integration in the relativistic theory, a set of independent integrals has to be kept for each selected transition. Therefore, the *pre-calculation* of the radial integrals is taken as default but can be overwritten interactively.

A further acceleration occurs for most computations owing to the parallel treatment of the selected transitions so that each cofactor $D_{UV}(k|l)$ only needs to be calculated once. Here, the explicit storage of the Bessel functions or radial integrals reduces the numerical effort remarkably. Particularly for large transition arrays, this speeds up the computation, a feature which we recently exploited for the Ni II spectra including a partially filled d -shell. Several tests on the number and symmetry of the one-particle functions have also been introduced to determine already in advance whether a given cofactor $D_{UV}(k|l)$ must vanish identically. For this purpose, a dynamically created array carries for all determinants the number of occupied orbitals for each symmetry block (κm) from which a *zero* value of the cofactors is easily derived.

Pointers and a dynamic storage management has been used already in the previous version. In Fortran 90/95, however, these concepts are supported much clearer and lead, together with derived data structures, to programs which are very flexible with respect to the size of the actual investigation. All arrays related to the representation of the atomic states, grid sizes, and the selected transitions are now allocated dynamically at running time. Therefore, there are virtually no other limitations than by the resources themselves. Dynamic allocation and the use of automatic data objects are used to solve all subtasks according to the Fortran 90/95 standard.

In REOS, the previous version, serious difficulties with the storage already occurred for wave function expansions of about a few thousand determinants. This made it necessary, in particular, to modify the internal representation of the occupation numbers of the determinants. These numbers are defined with respect to a reference list of one-electron orbital functions and are now stored *bitwise* within integers. The (intrinsic) bit manipulation procedures of Fortran 90/95 are then used to deal with the occupation in line with the data representation as provided by the CESD99 expansion file (see below).

2.3. Input and output files

REOS99 is controlled interactively at the beginning of execution. Besides the (needed) input data several options can be selected to modify the computational model or just to adapt the printout to the requirements of the user. In Section 3, we display a typical *input* dialog to test the installation of the program; it shows the standard dialog without displaying however the optional part of the input which is executed only if the question “*Modify the default set-up and printout of the program?*” is answered by *y(es)*.

For the execution of REOS99, a valid representation of the initial and final-state wave functions in a determinant basis is first of all required. Such a representation is obtained from the component CESD99 by using GRASP92 wave functions in a symmetry-adapted basis. Apart from information about the nuclear charge and model and the radial orbital functions, the CESD99 expansion (`.xpn`) file now contains the full representation of the atomic states. No further reference is made in REOS99 to the configuration symmetry list (`.csl`) files from GRASP92. The CESD99 expansion files include a description of the determinant basis as well as the expansion coefficients of the selected states in this basis. The occupation numbers of the determinants are stored within an appropriate number of “integers” to the CESD99 expansion file in line with the bitwise storage of the occupation numbers; this file has ASCII format to facilitate the file transfer from one system to another.

For the study of transition arrays, several initial and final atomic states may occur and be contained in the corresponding `.xpn` files. In this situation, it has been found useful to *group* the atomic levels with the same angular momentum and parity together and to carry out transition probability calculations just for these “groups” of levels. Even though such a computational scheme retains a large part of the rearrangement effects of the electron density, it still keeps the computation of large transition arrays feasible.

A summary of the transition probability calculations is finally written to the REOS99 `.sum` file. This file includes the date and time of the execution, information about the initial and final atomic levels as well as the *selected* transitions among them. It also contains two tables of the calculated transition probabilities and lifetimes. These tables list the transition energies, Einstein A and B coefficients, oscillator strengths as well as the decay widths in atomic units which are of particular interest for studying inner-shell processes.

2.4. Distribution and installation of the program

In Fortran 90/95, the *use* of modules allows a simple structure of the directories in which the code is distributed. This follows from the fact that modules provide a means of keeping the definition of (global) variables, data, derived types and their associated operations together. A proper design of the code should collect *everything* associated with some task within (one or several) modules which may contain a few thousand lines of code. This feature has been exploited for the RATIP package whose root directory `ratip` currently contains 8 modules and the two (main) programs `xcesd99.f` and `xreos99.f`. This root directory also includes several makefiles for generating the executables as well as test suites for the different components. For example, the command `make -f make-reos99` will generate the executable `xreos99` in the same directory. The name of the (Fortran 90/95) compiler and special compiler flags can easily be overwritten in the header of the makefile. Neither further libraries nor the adaptation of dimensions is required which facilitates the porting of the program to other environments. Till now, this revised implementation of the RATIP package has been tested extensively under the operating systems Linux and AIX. Owing to the compliance of the Fortran 90/95 standard, however, no serious difficulties should arise on other platforms.

For each component, we also provide a test suite in a subdirectory `test-component`; for example, `test-reos99` contains all necessary input and output files for the test cases as described below. This test ought to check the current installation of the program.

The simple structure of the `ratip` root facilitates the extension of the package by further components. In fact, this can be achieved by appending modules, (main) programs, and corresponding makefiles in order to generate the executables. Since several coworkers have contributed to different parts of the package, this structure ensures that further code developments can be made independently while the maintenance on both, users and developers side still remains straightforward. A few further instructions for the installation of the program are given in a `Read.me` file which is intended, in particular, for future up-dates where it will provide information about whether modules, makefiles and/or test suites have to be appended, replaced, or even removed from the `ratip` directory.

2.5. Subprograms

We do not intend to list and explain here all the module procedures of REOS99; those procedures which are specific to this component are contained in the module `rabs_reos`. Generally at the beginning of each module, all included module procedures are briefly described in alphabetic order. Further details can then be found in the header of each procedure. Apart from the modules `rabs_reos` and `rabs_cesd`, the other six modules provide procedures and functionality which are shared by different applications of RATIP, i.e. by different program components. Some of these modules will be (slightly) modified in the course of the adaptation of further components to the present standard but this does not change the overall structure.

3. Test data

The influence of relaxation effects on transition probabilities and lifetimes has been found important in a number of case studies [5,7]. To test REOS99 and to demonstrate its (new) capabilities, we discuss a computation of the electric-dipole allowed (E1) transitions among the levels of the $3s^23p^5$ and the $3s3p^6$, $3s^23p^43d$ configurations for Fe^{9+} ions. Lines from this spectrum are important in particular for astrophysical diagnostics since iron is one of the most abundant heavy elements in a large variety of astronomical objects. From this spectrum, we select the decay lines of the low-lying, even-parity levels with $J = 1/2$ and assume that the wave functions for the initial and final states have already been generated before by using GRASP92 [3]. These wave functions are provided in the subdirectory `test-reos99` which is distributed with the program; to generate them, we included up to quadruple excitations of the seven valence electrons outside the Ne-like core into the $3l$ subshells as well as single and double

```

REOS99: Calculation of relaxed-orbital transitions probabilities
and lifetimes within a determinant basis (Fortran 90 version)
(C) Copyright by S Fritzsche and others, Kassel (1999).

Generate debug printout ?
n
Enter a file name for the reos.sum file:
FeX-3sq4sd-reos.sum-cpc
Restart a previously aborted calculation ?
n
Enter the name of the isotope data file:
isodat26
loading isotope data file ...
... load complete;
Enter the transition multipoles, e.g. E1 M2 ... :
E1
Which units are to be used to express the transition energies ?
A      : Angstrom;
eV     : electron volts;
Hartree : Hartree atomic units;
Hz     : Hertz;
Kayser  : [cm**(-1)];
Kayser
Modify default set-up and printout of the program ?
n
Enter the name of the initial-state CESD .xpn expansion file:
This file must be in the new compact format.
FeX-initial.xpn
Enter the name of the final-state CESD .xpn expansion file:
This file must be in the new compact format.
FeX-final.xpn
Enter the name of the initial-state Radial WaveFunction File:
This file must be formatted.
FeX-initial.out-fmt
... load complete;
Enter the name of the final-state Radial WaveFunction File:
This file must be formatted.
FeX-final.out-fmt
... load complete;

```

Fig. 1. Interactive dialog for calculating transition probabilities with REOS99.

excitations into the $4l$ shells. By including these important classes of excitations we finally arrived (after applying CESD99) at wave function expansion of 5059 determinants for the two $3s^23p^5$ $J = 1/2$, and $J = 3/2$ ground-state levels and of 33994 determinants for the six $J = 1/2$ levels from the $3s3p^6$ and $3s^23p^43d$ configurations. A full account of this and of an even more enhanced computation has been presented elsewhere [8].

The wave functions of the initial and final states are stored in `test-reos99` in the files `FeX-initial.xpn` and `FeX-final.xpn`, respectively. We also provide in this directory a set of formatted and unformatted radial orbitals; these are given in `FeX-initial.rwf-fmt` and `FeX-initial.rwf` for the initial states and in corresponding files for the final states. The use of formatted radial orbitals is suggested for computers under Linux and supported if the logical flag `reos_use_formatted_rwf` in the header of the module `rabs_reos` is set to `.true`.

The computation of transition probabilities and (radiative) lifetimes is now carried out by following the interactive dialog from Fig. 1. The default input concerns the selection of the multipole components of the radiation field to be included, the wave function expansions (`.xpn` files), and a valid representation of the radial orbitals. There is also the possibility to *continue* a previously aborted computation if proper *restart files* were generated before. From `isodat26`, which is provided in the same directory, information about the nuclear parameters

are obtained. Although not displayed in Fig. 1, the optional part of the interactive dialog enables the user to overwrite default values. Several selections could be made concerning the computational model, the application of “experimental” energies, the radial grid, and others. To enable a later restart of a computation, the generation of two restart files would also have to be confirmed here; while the file `.res-trans` periodically overwrites the (yet incomplete) amplitudes of the selected transitions, the dump file `.res-dump`, in particular, may become very large since it contains (again) the full description of the initial and final states, information about the computational model, grid parameters, optional flags, etc. in an appropriate format. Most defaults of REOS99 are determined by logical flags which are defined in the header of the module `rabs_reos`; these flags need not to be explained here in detail since names were chosen to be self-explanatory.

The results of our test example above is provided in the file `FeX-3sq4sd-reos.sum-cpc`. The test run output displays the table of the transition probabilities and radiative lifetimes from this file along with further output of this test. To report the current status of the program, particularly in *large-scale* studies, a short printout is displayed on the standard stream if 10^8 matrix elements (as default) between initial and final-state determinants have been calculated. On present-day workstations, this results in approximately 10 lines per day to report how the calculation proceeds. This default value can also be overwritten by the interactive dialog.

As seen from the test output, a coherent treatment of relativistic, correlation, and relaxation effects results in a rather good agreement for the transition rates in Coulomb and Babushkin gauge (respective, velocity and length gauge in the nonrelativistic limit). The lifetime of the lowest even-parity $3s3p^6 J = 1/2$ level agrees pretty well with a beam-foil measurement of $2.70 \pm 0.20 \times 10^{-10}$ sec by Träbert [9]. Previous computations by Huang et al. [10], Fawcett [11], and Bhatia and Doschek [12] scattered among each other and with respect to the beam-foil value by about a factor of 2; by including correlation and relaxation effects to even a larger extent in our recent computations [8], we demonstrated that *ab initio* investigations can provide accurate and reliable data today.

4. Outlook

The program REOS for *relaxed-orbital oscillator strength* calculations was developed originally for wave functions from GRASP-2 [13] at the beginning of the 90ies. Since that time it has been maintained for almost one decade and has been exploited in a number of elaborate case studies. In this period, not only the capabilities but also the weaknesses of the program became obvious. The new version, REOS99, now resolves most of these defects. By using the concepts of Fortran 90/95 in a consistent manner, we present a revised version which is capable for large-scale *ab initio* computations for open-shell atoms. In a recent investigation, we studied the $3d^9-3d^84p$ and $3d^84s-3d^84p$ transitions in Ni II by including wave function expansions of up to 185,000 determinants for a few individual atomic levels. Obviously, this opens a doorway for calculating optical and ultraviolet spectra as needed in different fields in physics.

With the implementation of RATIP on the basis of Fortran 90/95, we start a new chapter in calculating atomic transition and ionization properties which combines the numerical strength of Fortran with an object-oriented approach. In addition, the new ANSI standard clearly supports parallel computations in the future for which we have recently started first investigations in the framework of MPI [14]. The new implementation of REOS (as part of the RATIP program) improves the application and the portability of the package on different platforms. Apart from transition probability calculations, further applications will be related to a user-friendly generation of continuum states which is currently under work. Such continuum spinors may be utilized for studying the emission of an electron, for instance in atomic photoionization or Auger processes, or for electron scattering calculations. The modular set-up of the RATIP package as described in Section 2 facilitates the extension of the package by further components which were developed by different collaborations in the past. Though the full adaptation of these components can not always be realized at once, the implementation and application of the two components CESD99 and REOS99 based on the ANSI standard Fortran 90/95 will certainly accelerate this process.

Acknowledgement

Even though this revised version REOS99 has no link anymore to the subprograms from GRASP92, we had to adapt a few procedures from this package in order to facilitate the file transfer. GRASP92 has mainly been written by Dr. Farid A. Parpia whom we would like to acknowledge gratefully. This work has been supported by the Deutsche Forschungsgemeinschaft (DFG) in the framework of the Schwerpunkt ‘Wechselwirkung intensiver Laserfelder mit Materie’. The research of one of the authors (CFF) has been supported by the Division of Chemical Sciences, Office of Basic Energy Sciences, Office of Energy Science, U.S. Department of Energy.

References

- [1] S. Fritzsche, C.F. Fischer, *Comp. Phys. Commun.* 99 (1997) 323.
- [2] S. Fritzsche, I.P. Grant, *Comp. Phys. Commun.* 103 (1997) 277.
- [3] F.A. Parpia, C.F. Fischer, I.P. Grant, *Comput. Phys. Commun.* 94 (1996) 249.
- [4] S. Fritzsche, B. Fricke, *Phys. Scr. T* 41 (1992) 45; *Phys. Rev. A* 45 (1991) 1465.
- [5] S. Fritzsche, I.P. Grant, *Phys. Lett. A* 186 (1994) 152;
C. Kohstall, S. Fritzsche, B. Fricke, W.-D. Sepp, *At. Data Nucl. Data Tables* 70 (63) (1998) 63;
S. Fritzsche, B. Fricke, D. Geschke, A. Heitmann, J.E. Sienkiewicz, *Astrophys. J.* 518 (1999) 994.
- [6] M. Metcalf, J. Reid, *Fortran 90/95 Explained* (Oxford University Press, 1996).
- [7] S. Fritzsche, C.F. Fischer, B. Fricke, *At. Data Nucl. Data Tables* 68 (1998) 149.
- [8] C.Z. Dong, S. Fritzsche, B. Fricke, W.-D. Sepp, *Mon. Notes R. Astr. Soc.* 307 (1999) 809.
- [9] E. Träbert, *J. Phys. B* 29 (1996) L217.
- [10] K.-N. Huang, Y.-K. Kim, K.T. Cheng, *At. Data Nucl. Data Tables* 28 (1983) 355.
- [11] B.C. Fawcett, *At. Data Nucl. Data Tables* 36 (1987) 151.
- [12] A.K. Bhatia, G.A. Doschek, *At. Data Nucl. Data Tables* 66 (1995) 97.
- [13] F.A. Parpia, I.P. Grant, GRASP-2 (unpublished, 1990).
- [14] M. Snir, S.W. Otto, S. Huss-Lederman, D.W. Walker, J. Dongarra, *MPI – The Complete Reference* (MIT Press, Cambridge, London, 1996).

TEST RUN OUTPUT

Initialize the set-up of the overlap integrals and transitions ...
 12 transitions have been initialized and will be calculated in this run of the program.
 set-up of 'symmetry blocks' for the initial-states complete;
 set-up of 'symmetry blocks' for the final-states complete;
 pre-calculation of 2304 radial integrals complete;
 ... initialization complete.

The following 12 transitions are selected:

I-level-F	I--J [^] P--F	Transition Energy (in Kays)	Multipoles
1 - 2	1/2+ 1/2-	2.7441791E+05	E1
1 - 1	1/2+ 3/2-	2.8978772E+05	E1
2 - 2	1/2+ 1/2-	3.7698499E+05	E1
2 - 1	1/2+ 3/2-	3.9235480E+05	E1
3 - 2	1/2+ 1/2-	4.0676550E+05	E1
3 - 1	1/2+ 3/2-	4.2213531E+05	E1
4 - 2	1/2+ 1/2-	4.2222749E+05	E1
4 - 1	1/2+ 3/2-	4.3759730E+05	E1
5 - 2	1/2+ 1/2-	5.3710929E+05	E1
5 - 1	1/2+ 3/2-	5.5247910E+05	E1
6 - 2	1/2+ 1/2-	5.6500232E+05	E1
6 - 1	1/2+ 3/2-	5.8037213E+05	E1

About 1.7197564600000000E+08 pairs of final- and initial-state matrix elements need to be calculated.
 0 billion 65316163 pairs calculated; 0 billion 34683836 neglected; 0 billion 10000000 in total;

=====
| Summary of all Transition Probabilities and Lifetimes |
=====

Levi-LevF	I - J	Parity -F	Energy (Kays)	Multipol	Gauge	Einstein coefficients			Oscillator strength GF	Decay width (eV)
						A (s ⁻¹)	gB (m s ⁻¹ J ⁻¹)			
1 - 2	1/2 +	1/2 -	2.74418D+05	E1	Babushkin	1.29100D+09	7.50300D+18	5.14045D-02	8.49752D-07	
1 - 2	1/2 +	1/2 -	2.74418D+05	E1	Coulomb	1.23931D+09	7.20257D+18	4.93462D-02	8.15727D-07	
1 - 1	1/2 +	3/2 -	2.89788D+05	E1	Babushkin	2.90213D+09	1.43226D+19	1.03623D-01	1.91021D-06	
1 - 1	1/2 +	3/2 -	2.89788D+05	E1	Coulomb	2.67843D+09	1.32186D+19	9.56358D-02	1.76298D-06	
2 - 2	1/2 +	1/2 -	3.76985D+05	E1	Babushkin	4.64005D+06	1.04015D+16	9.78982D-05	3.05414D-09	
2 - 2	1/2 +	1/2 -	3.76985D+05	E1	Coulomb	4.98551D+06	1.11759D+16	1.05187D-04	3.28152D-09	
2 - 1	1/2 +	3/2 -	3.92355D+05	E1	Babushkin	3.80779D+06	7.57150D+15	7.41677D-05	2.50633D-09	
2 - 1	1/2 +	3/2 -	3.92355D+05	E1	Coulomb	3.94359D+06	7.84153D+15	7.68128D-05	2.59572D-09	
3 - 2	1/2 +	1/2 -	4.06766D+05	E1	Babushkin	1.88332D+08	3.36076D+17	3.41299D-03	1.23962D-07	
3 - 2	1/2 +	1/2 -	4.06766D+05	E1	Coulomb	2.14818D+08	3.83340D+17	3.89298D-03	1.41396D-07	
3 - 1	1/2 +	3/2 -	4.22135D+05	E1	Babushkin	4.47353D+07	7.14235D+16	7.52743D-04	2.94453D-08	
3 - 1	1/2 +	3/2 -	4.22135D+05	E1	Coulomb	5.09119D+07	8.12851D+16	8.56676D-04	3.35109D-08	
4 - 2	1/2 +	1/2 -	4.22227D+05	E1	Babushkin	4.47347D+07	7.13758D+16	7.52404D-04	2.94449D-08	
4 - 2	1/2 +	1/2 -	4.22227D+05	E1	Coulomb	4.95971D+07	7.91339D+16	8.34187D-04	3.26454D-08	
4 - 1	1/2 +	3/2 -	4.37597D+05	E1	Babushkin	2.63811D+08	3.78107D+17	4.13089D-03	1.73644D-07	
4 - 1	1/2 +	3/2 -	4.37597D+05	E1	Coulomb	2.60978D+08	3.74048D+17	4.08654D-03	1.71779D-07	
5 - 2	1/2 +	1/2 -	5.37109D+05	E1	Babushkin	4.08685D+10	3.16772D+19	4.24779D-01	2.69001D-05	
5 - 2	1/2 +	1/2 -	5.37109D+05	E1	Coulomb	4.23560D+10	3.28302D+19	4.40240D-01	2.78792D-05	
5 - 1	1/2 +	3/2 -	5.52479D+05	E1	Babushkin	1.35644D+11	9.66050D+19	1.33251D+00	8.92826D-05	
5 - 1	1/2 +	3/2 -	5.52479D+05	E1	Coulomb	1.37474D+11	9.79080D+19	1.35048D+00	9.04869D-05	
6 - 2	1/2 +	1/2 -	5.65002D+05	E1	Babushkin	1.18695D+11	7.90362D+19	1.11489D+00	7.81262D-05	
6 - 2	1/2 +	1/2 -	5.65002D+05	E1	Coulomb	1.16991D+11	7.79022D+19	1.09889D+00	7.70052D-05	
6 - 1	1/2 +	3/2 -	5.80372D+05	E1	Babushkin	3.76222D+10	2.31138D+19	3.34912D-01	2.47634D-05	
6 - 1	1/2 +	3/2 -	5.80372D+05	E1	Coulomb	3.64282D+10	2.23802D+19	3.24283D-01	2.39775D-05	

Radiative lifetimes and widths

Level	Gauge	Lifetime	Width		
		Seconds	Hartrees	Kaysers	eV
1	Babushkin:	2.3848564D-10	6.3728360D-07	1.3986758D-01	1.7341377D-05
	Coulomb:	2.5524937D-10	5.9542942D-07	1.3068165D-01	1.6202466D-05
2	Babushkin:	1.1837349D-07	1.2839276D-09	2.8178953D-04	3.4937462D-08
	Coulomb:	1.1199344D-07	1.3570704D-09	2.9784252D-04	3.6927780D-08
3	Babushkin:	4.2906152D-09	3.5422189D-08	7.7742719D-03	9.6388723D-07
	Coulomb:	3.7632239D-09	4.0386379D-08	8.8637855D-03	1.0989698D-06
4	Babushkin:	3.2410130D-09	4.6893667D-08	1.0291970D-02	1.2760421D-06
	Coulomb:	3.2198294D-09	4.7202186D-08	1.0359682D-02	1.2844374D-06
5	Babushkin:	5.6653181D-12	2.6826911D-05	5.8878265D+00	7.2999771D-04
	Coulomb:	5.5608161D-12	2.7331058D-05	5.9984738D+00	7.4371624D-04
6	Babushkin:	6.3972674D-12	2.3757485D-05	5.2141653D+00	6.4647434D-04
	Coulomb:	6.5180691D-12	2.3317179D-05	5.1175293D+00	6.3449300D-04

REOS99 complete