Computer Physics Communications 139 (2001) 263–278

Computer Physics Communications

www.elsevier.com/locate/cpc

Program to calculate pure angular momentum coefficients in jj-coupling ☆

Gediminas Gaigalas [a,b,*], Stephan Fritzsche [a], Ian P. Grant [c]

[a] *Fachbereich Physik, Universität Kassel, Heinrich-Plett-Str. 40, D-34132 Kassel, Germany*
[b] *Institute of Theoretical Physics and Astronomy, A. Goštauto 12, Vilnius 2600, Lithuania*
[c] *Mathematical Institute, University of Oxford, 24/29 St. Giles', Oxford OX1 3LB, UK*

Accepted 19 December 2000

Abstract

A program for computing pure angular momentum coefficients in relativistic atomic structure for any scalar one- and two-particle operator is presented. The program, written in Fortran 90/95 and based on techniques of second quantization, irreducible tensorial operators, quasispin and the theory of angular momentum, is intended to replace existing angular coefficient modules from GRASP92. The new module uses a different decomposition of the coefficients as sums of products of pure angular momentum coefficients, which depend only on the tensor rank of the interaction but not on its details, with effective interaction strengths of specific interactions. This saves memory and reduces the computational cost of big calculations significantly.

PROGRAM SUMMARY

Title of program: ANCO

Catalogue identifier: ADOO

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

Program Summary URL: http://cpc.cs.qub.ac.uk/summaries/ADOO

Licensing provisions: None

Computer for which the program is designed and 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

Memory required to execute with typical data: 100 kB

No. of bits in a word: All real variables are parametrized by a `selected kind parameter`. Currently this is set to double precision for consistency with other components of the RATIP package [1]

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

* Corresponding author.
E-mail address: gaigalas@itpa.lt (G. Gaigalas).

0010-4655/01/$ – see front matter
PII: S0010-4655(01)00213-2

No. of bytes in distributed program, including test data, etc.: 96 941

Distribution format: Compressed tar file. On a UNIX (or compatible) workstation, the command `uncompress` uncompress this file and the command `tar -xvf ratip_anco.tar` reconstructs the file structure

Keywords: Atomic many-body perturbation theory, complex atom, configuration interaction, effective Hamiltonian, energy level, Racah algebra, reduced coefficients of fractional parentage, reduced matrix element, relativistic, second quantization, standard unit tensors, tensor operators, 9/2-subshell

Nature of the physical problem
The matrix elements of a one-electron tensor operator $\widehat{A}^k$ of rank k with respect to a set of configuration state functions $|\gamma_i J_i\rangle$ can be written $\sum_{ab} t^k_{ij}(ab)\,(a|\widehat{A}^k|b)$ where the angular coefficients $t^k_{ij}(ab)$ are independent of the operator $\widehat{A}^k$, i, j are CSF labels and a, b run over the relevant interacting orbital labels. Similarly, the matrix elements of the Dirac–Coulomb Hamiltonian can be written in the form $\sum_{ab} t^0_{ij}(ab)\,(a|\widehat{H}_D|b) + \sum_k \sum_{abcd} v^k_{ij}(abcd)\,X^k(abcd)$, where $\widehat{H}_D$ is the one-electron Dirac Hamiltonian operator, with tensor rank zero, $v^k_{ij}(abcd)$ are pure angular momentum coefficients for two-electron interactions, and $X^k(abcd)$ denotes an effective interaction strength for the two-electron interaction. The effective interaction strengths for Coulomb and Breit interaction have different selection rules and make use of subsets of the full set of coefficients $v^k_{ij}(abcd)$.
Such matrix elements are required for the theoretical determination of atomic energy levels, orbitals and radiative transition data in relativistic atomic structure theory. The code is intended for use in configuration interaction or multiconfiguration Dirac–Fock calculations [2], or for calculation of matrix elements of the effective Hamiltonian in many-body perturbation theory [3].

Method of solution
A combination of second quantization and quasispin methods with the theory of angular momentum and irreducible tensor operators leads to a more efficient evaluation of (many-particle) matrix elements and to faster execution of the code [4].

Restrictions on the complexity of the problem
Tables of reduced matrix elements of the tensor operators $a^{(q\ j)}$ and $W^{(k_q k_j)}$ are provided for (nj) with $j = 1/2, 3/2, 5/2, 7/2$, and $9/2$. Users wishing to extend the tables must provide the necessary data.

Typical running time
3.5 s for both examples on a 450 MHz Pentium III processor.

Unusual features of the program
The program is designed for large-scale atomic structure calculations and its computational cost is less than that of the corresponding angular modules of GRASP92. The present version of the program generates pure angular momentum coefficients $t^0_{ij}(ab)$ and $v^k_{ij}(abcd)$, but coefficients $t^k_{ij}(ab)$ with $k > 0$ have not been enabled. An option is provided for generating coefficients compatible with existing GRASP92.
Configurational states with any distribution of electrons in shells with $j \leqslant 9/2$ are allowed. This permit user to take into account the single, double, triple excitations form open d- and f-shells for the systematic MCDF studies of heavy and superheavy elements (Z > 95).

References
[1] S. Fritzsche, C.F. Fischer, C.Z. Dong, Comput. Phys. Commun. 124 (1999) 240.
[2] I.P. Grant, in: S. Wilson (Ed.), Methods of Computational Chemistry, Vol. 2, Plenum Press, New York, 1988, pp. 1–71;
K.G. Dyall, I.P. Grant, C.T. Johnson, F.A. Parpia, E.P. Plummer, Comput. Phys. Commun. 55 (1989) 425;
F.A. Parpia, C.F. Fischer, I.P. Grant, Comput. Phys. Commun. 92 (1996) 249.
[3] G. Merkelis, G. Gaigalas, J. Kaniauskas, Z. Rudzikas, Izvest. Acad. Nauk SSSR, Phys. Ser. 50 (1986) 1403.
[4] G. Gaigalas, Lithuanian J. Phys. 39 (1999) 80.

LONG WRITE-UP

1. Introduction

The improved accuracy of modern experiments challenges theorists to match or exceed experimental precision. Models of many-electron atoms and ions require both relativistic and correlation effects to be taken into account; this can be done, for example, by using various versions of perturbation theory, the configuration interaction method, the multiconfiguration Hartree–Fock method [1] or the multiconfiguration Dirac–Fock method [2].

The evaluation of matrices of one- and two-electron operators for many-electron states in jj-coupling is customarily done by expressing each matrix element as a sum of products of angular coefficients and radial integrals. This strategy, based on earlier work by Fano, was adopted for the MCP program [3,4] for evaluating angular coefficients for the Coulomb interaction and the related MCBP program [5] for evaluating angular

coefficients for the Breit interaction. Whilst this strategy was adequate on the relatively low-powered computers of the 1970s, it is now possible and desirable to use very large configuration sets which require a more efficient strategy. The new program attains this objective by building up the angular coefficients for one- and two-electron interactions from a relatively small number of common spin-angular parts in the manner of [6–8].

The theoretical background is presented in Section 2, program organization in Section 3, testing and timing studies in Section 4 and simple test problems in Section 5.

2. Theoretical background

2.1. Dirac–Coulomb Hamiltonian

As usual, multiconfiguration self-consistent-field calculations in relativistic atomic theory are based on the Dirac–Coulomb Hamiltonian:

$$\widehat{H}_{DC} = \sum_{i=1}^{N} \widehat{H}_D(i) + \frac{1}{2} \sum_{i,j=1}^{N} \frac{1}{r_{ij}}. \tag{1}$$

$\widehat{H}_D(i)$ is the Dirac one-particle operator,

$$\widehat{H}_D = c\boldsymbol{\alpha}(i) \cdot \boldsymbol{p}_i + \big[\beta(i) - 1\big]c^2 - \frac{Z}{r_i}, \tag{2}$$

and the second term in (1) represents the Coulomb interaction of pairs of electrons. In Eq. (2) c denotes the speed of light; $\boldsymbol{\alpha}(i)$ and $\beta(i)$ are 4×4 Dirac matrices for the ith electron; r_i and $\boldsymbol{p}_i$ are the radial coordinate of the ith electron and its (3-) momentum, respectively. The first two terms of (2) comprise the Dirac kinetic energy operator.

Both configuration interaction and multiconfiguration Dirac–Hartree–Fock calculations require the matrix elements of $\widehat{H}_{DC}$ with respect to a basis of n-electron configurational states labelled $|\gamma_r J_r\rangle, r = 1, 2, 3, \ldots$. The present program computes the pure angular coefficients $t^k_{rs}(ab)$ (only for $k = 0$ in the present version) and $v^k_{rs}(abcd)$ in the expression

$$\langle \gamma_r J_r | \widehat{H}_{DC} | \gamma_s J_s \rangle = \sum_{ab} \left\{ t^0_{rs}(ab)(a|\widehat{H}_D|b) + \sum_k \sum_{cd} v^{(k)}_{rs}(abcd) X^{(k)}(abcd) \right\}, \tag{3}$$

where $X^{(k)}(abcd)$ is the effective interaction strength of the two-electron interaction with respect to the orbitals concerned. Here in (3), the interaction is the Coulomb potential only, where

$$\begin{aligned} X^{(k)}(abcd) \;=\; & (-1)^k \langle n_a l_a j_a \| C^{(k)} \| n_c l_c j_c \rangle \langle n_b l_b j_b \| C^{(k)} \| n_d l_d j_d \rangle \\ & \times R^k(n_a l_a j_a n_b l_b j_b n_c l_c j_c n_d l_d j_d), \end{aligned} \tag{4}$$

but the formalism can also be used for the Breit interaction, with a different definition of $X^{(k)}(abcd)$ and different selection rules. Eq. (3) is a rearrangement of the formula used in GRASP92 [9, Eq. (3.10)],

$$\langle \gamma_r J_r | \widehat{H}_{DC} | \gamma_s J_s \rangle = \sum_{ab} \left\{ T_{rs}(ab) I(ab) + \sum_k \sum_{cd} V^{(k)}_{rs}(abcd)\, R^{(k)}(abcd) \right\}, \tag{5}$$

where

$$I(ab) = \big(a|\widehat{H}_D|b\big)$$

and

$$R^{(k)}(abcd) = \big(ab\big|{r_<}^k / {r_>}^{k+1}\big|cd\big)$$

is a relativistic Slater integral over orbitals a, b, c, d (in the usual notation). Whilst the one-electron part is the same, Eq. (5) expands the effective interaction strength as a traditional sum of Slater integrals, and $V_{rs}^{(k)}(abcd)$ therefore differs from $v_{rs}^{(k)}(abcd)$. The latter can be used for *any* two-electron interaction, and it is no longer necessary to treat Coulomb and Breit interactions on a different footing in the manner of GRASP92. Other intractions such as the lowest-order normal mass shift and the specific mass shift [9, Eqs. (3.12–3.14)] can be handled within the same scheme. The way in which the coefficients are built up is described below.

2.2. One-particle operators

The matrix elements of a one-particle scalar operator $\widehat{F}^{(0)}$ between configuration state functions with u open shells can be expressed as a sum over one-electron contributions

$$\big(\psi_u^{\mathrm{bra}}(J)\big\|\widehat{F}^{(0)}\big\|\psi_u^{\mathrm{ket}}(J')\big)=\sum_{n_i\kappa_i,n_j\kappa_j}\big(\psi_u^{\mathrm{bra}}(J)\big\|\widehat{F}(n_i\kappa_i,n_j\kappa_j)\big\|\psi_u^{\mathrm{ket}}(J')\big), \tag{6}$$

where

$$\begin{aligned}&\big(\psi_u^{\mathrm{bra}}(J)\big\|\widehat{F}(n_i\kappa_i,n_j\kappa_j)\big\|\psi_u^{\mathrm{ket}}(J')\big)\\&\quad=(-1)^{\Delta+1}\sqrt{2j_i+1}\,R\big(j_i,j_j,\Lambda^{\mathrm{bra}},\Lambda^{\mathrm{ket}}\big)\delta(\kappa_i,\kappa_j)\big(n_i\kappa_i\big\|f^{(0)}\big\|n_j\kappa_j\big)\\&\qquad\times\Big\{\delta(n_i,n_j)\big(j_i^{N_i}\,\alpha_i Q_i J_i\big\|\big[a_{1/2}^{(q\,j_i)}\times a_{-1/2}^{(q\,j_i)}\big]^{(0)}\big\|j_i^{N_i}\,\alpha_i Q_i J_i\big)\\&\qquad\quad+\big(1-\delta(n_i,n_j)\big)\big(j_i^{N_i}\,\alpha_i Q_i J_i\big\|a_{1/2}^{(q\,j_i)}\big\|j_i^{N_i'}\alpha_i Q_i J_i\big)\big(j_j^{N_j}\alpha_j Q_j J_j\big\|a_{-1/2}^{(q\,j_j)}\big\|j_j^{N_j'}\,\alpha_j Q_j J_j\big)\Big\}.\end{aligned}$$

All states are defined in jj-coupling. $(\psi_u^{\mathrm{bra}}(J)\|$ and $\|\psi_u^{\mathrm{ket}}(J'))$ are respectively bra and ket functions with u open subshells, $\kappa\equiv(2j+1)(l-j)$, $(n_i\kappa_i\|f^{(0)}\|n_j\kappa_j)$ is the one-electron reduced matrix element of the operator $\widehat{F}^{(0)}$, $\Lambda^{\mathrm{bra}}\equiv(J_i,J_j,J_{i'},J_{j'})^{\mathrm{bra}}$ and $\Lambda^{\mathrm{ket}}\equiv(J_i,J_j,J_{i'},J_{j'})^{\mathrm{ket}}$ denote the respective sets of active subshell angular momenta. The operators $a_{m_q}^{(q\,j)}$ are second quantization operators in quasispin space of rank $q=1/2$. The operator $a_{1/2m_j}^{(q\,j)}=a_{m_j}^{(j)}$ creates electrons with angular momentum quantum numbers j,m_j and its conjugate $a_{-1/2\,m_j}^{(q\,j)}=\tilde{a}_{m_j}^{(j)}=(-1)^{j-m_j}a_{-m_j}^{(j)+}$ annihilates electrons with the same quantum numbers j,m_j in a given subshell.

2.2.1. Recoupling matrix

The recoupling matrix $R(j_i,j_j,\Lambda^{\mathrm{bra}},\Lambda^{\mathrm{ket}})$ in (6) is particularly simple. It is either a product of delta functions [6, Eq. (18)] when $n_i\kappa_i=n_j\kappa_j$ or a combination of delta functions and $6j$-coefficients [6, Eq. (22)] when $n_i\kappa_i\neq n_j\kappa_j$.

2.2.2. Matrix elements of irreducible tensor operators

By applying the Wigner–Eckart theorem in quasispin space we obtain the submatrix elements of operators of type $a_{m_q}^{(q\,j)}$ in the form [10]

$$\begin{aligned}&\big(j^N\alpha Q J\big\|a_{m_q}^{(q\,j)}\big\|j^{N'}\,\alpha' Q' J'\big)\\&\quad=-[Q]^{-1/2}\begin{bmatrix}Q' & 1/2 & Q\\ M_Q' & m_q & M_Q\end{bmatrix}\big(j^N\alpha Q J\big\|\big\|a^{(q\,j)}\big\|\big\|j\,\alpha' Q' J'\big),\end{aligned} \tag{7}$$

where we have used the conventional shorthand notation $(2k+1)\cdot\cdots\equiv[k,\ldots]$, and the last factor is a reduced coefficient of fractional parentage. The submatrix elements of the simplest compound tensor operator of type $[a_{m_{q2}}^{(q\,j)}\times a_{m_{q2}}^{(q\,j)}]^{(k_j)}$ uses

$$\begin{aligned}
&\left(nj^N\,\alpha QJ\left\|\left[a^{(q\,j)}_{m_{q1}}\times a^{(q\,j)}_{m_{q2}}\right]^{(k_j)}\right\|nj^{N'}\,\alpha' Q' J'\right)\\
&\quad=\sum_{k_q,m_q}[Q]^{-1/2}\begin{bmatrix} q & q & k_q\\ m_{q1} & m_{q2} & m_q\end{bmatrix}\begin{bmatrix} Q' & k_q & Q\\ M'_Q & m_q & M_Q\end{bmatrix}\\
&\qquad\times\left(nj\,\alpha QJ\left|\left|\left|W^{(k_q k_j)}\right|\right|\right|nj\,\alpha' Q' J'\right),
\end{aligned}\tag{8}$$

where $(nj\,\alpha QJ|||W^{(k_q k_j)}|||nj\,\alpha' Q' J')$ denotes the reduced matrix element of the tensor operator $W^{(k_q k_j)}(nj,nj)=[a^{(q\,j)}\times a^{(q\,j)}]^{(k_q k_j)}$ in quasi-spin space. In terms of the fully reduced coefficients of fractional parentage $(j\,\alpha QJ|||a^{(q\,j)}|||j\,\alpha' Q' J')$, we find

$$\begin{aligned}
&\left(nj\,\alpha QJ\left|\left|\left|W^{(k_q k_j)}\right|\right|\right|nj\,\alpha' Q' J'\right)\\
&\quad=(-1)^{Q+J+Q'+J'+k_q+k_j}[k_q,k_j]^{1/2}\sum_{\alpha'' Q'' J''}\begin{Bmatrix} q & q & k_q\\ Q' & Q & Q''\end{Bmatrix}\begin{Bmatrix} j & j & k_j\\ J' & J & J''\end{Bmatrix}\\
&\qquad\times\left(j\,\alpha QJ\left|\left|\left|a^{(q\,j)}\right|\right|\right|j\,\alpha'' Q'' J''\right)\left(j\,\alpha'' Q'' J''\left|\left|\left|a^{(q\,j)}\right|\right|\right|j\,\alpha' Q' J'\right).
\end{aligned}\tag{9}$$

This construction has the advantage that the completely reduced matrix elements on the right-hand side of (7) and (8) are independent of the occupation number of the shell, and so reduces requirements of storage in comparison with earlier work. These formulae are evaluated in the module `rabs_rcfp` [11].

The phase factor arises from the reordering needed to match the recoupled creation and annihilation operators in the bra and ket vectors. We have

$$\Delta=0,\tag{10}$$

when $n_i\kappa_i=n_j\kappa_j$; otherwise

$$\Delta=1+\sum_{r=a}^{b-1}N_r,\tag{11}$$

where N_r is the occupation number of subshell r, $a=\min\{i,j\}$, and $b=\max\{i,j\}$.

2.2.3. The one-electron submatrix element

It only remains to define the one-electron interaction matrix element

$$\left(n_i\kappa_i\left\|f^{(0)}\right\|n_j\kappa_j\right)$$

in (6). The only operator required in this implementation is the matrix element of the Dirac operator, a tensor operator of rank zero,

$$\left(n_i\kappa_i\left\|\widehat{H}_D\right\|n_j\kappa_j\right)=I(n_i l_i j_i,n_i l_i j_i)\delta(\kappa_i,\kappa_j),\tag{12}$$

where $I(n_i l_i j_i,n_i l_i j_i)$ is defined by [12, Eq. (22)]. The Dirac kinetic energy operator, denoted by T in [9, Eq. (3.13)], can be obtained from this by setting the nuclear charge $Z=0$. The coefficients $T_{rs}(ab)$ in (3) can now be identified by inserting (12) in (6). Meanwhile the pure angular coefficients $t^0_{rs}(ab)$ for one-electron operators can be identified by inserting

$$\left(n_i\kappa_i\left\|f^{(0)}\right\|n_j\kappa_j\right)=1$$

in (6).

2.3. Two-particle operators

According to [6], the matrix element of any two-particle scalar operator $\widehat{G}^{(kk0)}$ between configuration state functions with u open shells, can be written

$$\left(\psi_u^{\mathrm{bra}}(J)\left\|\widehat{G}^{(kk0)}\right\|\psi_u^{\mathrm{ket}}(J')\right)$$
$$= \sum_{n_i\kappa_i, n_j\kappa_j, n_{i'}\kappa_{i'}, n_{j'}\kappa_{j'}} \left(\psi_u^{\mathrm{bra}}(J)\left\|\widehat{G}(n_i\kappa_i, n_j\kappa_j, n_{i'}\kappa_{i'}, n_{j'}\kappa_{j'})\right\|\psi_u^{\mathrm{ket}}(J')\right), \tag{13}$$

where

$$\left(\psi_u^{\mathrm{bra}}(J)\left\|\widehat{G}(n_i\kappa_i, n_j\kappa_j, n_{i'}\kappa_{i'}, n_{j'}\kappa_{j'})\right\|\psi_u^{\mathrm{ket}}(J')\right),$$
$$= \sum_{k_{12}} (-1)^{\Delta} \Theta'(n_i l_i j_i, n_j l_j j_j, n_{i'} l_{i'} j_{i'}, n_{j'} l_{j'} j_{j'}, \Xi)$$
$$\times T\left(n_i j_i, n_j j_j, n_{i'} j_{i'}, n_{j'} j_{j'}, \Lambda^{\mathrm{bra}}, \Lambda^{\mathrm{ket}}, \Xi, \Gamma\right) R\left(j_i, j_j, j_{i'}, j_{j'}, \Lambda^{\mathrm{bra}}, \Lambda^{\mathrm{ket}}, \Gamma\right),$$

Γ specifies the recoupling scheme required for each matrix element and Ξ, when required, specifies the coupling scheme of the tensor operators defining each matrix element. The operator $\widehat{G}^{(kk0)}$ couples tensor operators of rank k for each electron to give an overall scalar operator.

From (13) we see that the matrix element of any two-particle operator can be written as a sum over all possible sets of active shell quantum numbers $n_i\kappa_i$, $n_j\kappa_j$, $n_{i'}\kappa_{i'}$, $n_{j'}\kappa_{j'}$. The systematic analysis of [6] aims to minimize the computation needed in this expansion. The parameter distributions are presented in Table 1. Note that for distributions 2–5 and 19–42 the subshell labels are ordered so that $\alpha < \beta < \gamma < \delta$, while for distributions 6–18 no conditions upon the ordering are imposed. We discuss these structures in more detail below.

2.3.1. Recoupling matrix

The recoupling coefficients defined in [3–5] did not reduce the recoupling coefficients to their simplest forms but relied on the analysis module of the NJSYM package (later NJGRAF) to perform the reduction mechanically. The analysis [6] leads to simpler forms denoted by $R(j_i, j_j, j_{i'}, j_{j'}, \Lambda^{\mathrm{bra}}, \Lambda^{\mathrm{ket}}, \Gamma)$. In the case of one interacting shell $R(j_i, j_j, j_{i'}, j_{j'}, \Lambda^{\mathrm{bra}}, \Lambda^{\mathrm{ket}}, \Gamma)$ reduces to delta functions [6, Eq. (18)]. For two, three and four interacting shells, the recoupling coefficients are given by [6, Eqs. (22), (23) and (24)], replacing l, L by j, J, respectively. The recoupling parameters Γ for each distribution can be found in Table 1.

2.3.2. Matrix elements of irreducible tensor operators

The expressions $T(n_i j_i, n_j j_j, n_{i'} j_{i'}, n_{j'} j_{j'}, \Lambda^{\mathrm{bra}}, \Lambda^{\mathrm{ket}}, \Xi, \Gamma)$ are matrix elements of standard subshell creation/annihilation operators

$$a = a_{m_q}^{(q\,j)}, \tag{14}$$
$$W = \left[a_{m_{q1}}^{(q\,j)} \times a_{m_{q2}}^{(q\,j)}\right]^{(k_{12})}, \tag{15}$$
$$aW = \left[a_{m_{q1}}^{(q\,j)} \times \left[a_{m_{q2}}^{(q\,j)} \times a_{m_{q3}}^{(q\,j)}\right]^{(k_{12})}\right]^{(k_2)}, \tag{16}$$
$$Wa = \left[\left[a_{m_{q1}}^{(q\,j)} \times a_{m_{q2}}^{(q\,j)}\right]^{(k_{12})} \times a_{m_{q3}}^{(q\,j)}\right]^{(k_2)}, \tag{17}$$
$$WW = \left[\left[a_{m_{q1}}^{(q\,j)} \times a_{m_{q2}}^{(q\,j)}\right]^{(k)} \times \left[a_{m_{q3}}^{(q\,j)} \times a_{m_{q4}}^{(q\,j)}\right]^{(k)}\right]^{(0)}. \tag{18}$$

The creation and annihilation operators in (14)–(18) refer to a single subshell. The evaluation of the submatrix elements of operators of type a (14) and the simplest compound tensor operator of type W (15) was explained in Section 2.1.2. For types (16)–(18), we use the formula

$$\left(nj^N\,\alpha Q J\left\|[U^{(k_1)}(nj) \times V^{(k_2)}(nj)]^{(k)}\right\|nj^{N'}\,\alpha' Q' J'\right)$$
$$= (-1)^{J+J'+k}[k]^{1/2} \sum_{\alpha'' Q'' J''} \begin{Bmatrix} k_1 & k_2 & k \\ J' & J & J'' \end{Bmatrix},$$
$$\times \left(nj^N\,\alpha Q J\left\|U^{(k_1)}(nj)\right\|nj^{N''}\,\alpha'' Q'' J''\right)\left(nj^{N''}\,\alpha'' Q'' J''\left\|V^{(k_2)}(nj)\right\|nj^{N'}\,\alpha' Q' J'\right), \tag{19}$$

Table 1
Scheme of the definitions for matrix elements of any two-particle operator. The operators a, W, aW and Wa defined in (14)–(18) act on the indicated subshells

No.	i	j	i'	j'	Γ	Ξ	α	β	γ	δ	φ
1.	α	α	α	α	–	j_α, k	WW	–	–	–	–
					–	j_α	W	–	–	–	–
2.	α	β	α	β	k	j_α, j_β	W	W	–	–	0
3.	β	α	β	α	k	j_α, j_β	W	W	–	–	0
4.	α	β	β	α	k_{12}	j_α, j_β	W	W	–	–	0
5.	β	α	α	β	k_{12}	j_α, j_β	W	W	–	–	0
6.	α	α	β	β	k_{12}	j_α, j_β	W	W	–	–	$j_\alpha + j_\beta + k_{12}$
7.	β	α	α	α	j_β	j_α, k_{12}	aW	a	–	–	$k + k_{12}$
8.	α	β	α	α	j_β	j_α, k_{12}	aW	a	–	–	k
9.	β	β	β	α	j_α	j_β, k_{12}	a	Wa	–	–	$k + k_{12}$
10.	β	β	α	β	j_α	j_β, k_{12}	a	Wa	–	–	k
11.	β	γ	α	γ	j_α, j_β, k	–	a	a	W	–	$1 + j_\alpha + j_\beta - k$
12.	γ	β	γ	α	j_α, j_β, k	–	a	a	W	–	$1 + j_\alpha + j_\beta - k$
13.	γ	β	α	γ	$j_\alpha, j_\beta, k_{12}$	–	a	a	W	–	$1 + j_\alpha + j_\beta - k_{12}$
14.	β	γ	γ	α	$j_\alpha, j_\beta, k_{12}$	–	a	a	W	–	$1 + j_\alpha + j_\beta - k_{12}$
15.	γ	γ	α	β	$j_\alpha, j_\beta, k_{12}$	–	a	a	W	–	$j_\alpha + j_\gamma + k_{12}$
16.	γ	γ	β	α	$j_\alpha, j_\beta, k_{12}$	–	a	a	W	–	$j_\alpha + j_\gamma$
17.	α	β	γ	γ	$j_\alpha, j_\beta, k_{12}$	–	a	a	W	–	$j_\alpha + j_\beta + k_{12}$
18.	β	α	γ	γ	$j_\alpha, j_\beta, k_{12}$	–	a	a	W	–	$j_\beta + j_\gamma$
19.	α	β	γ	δ	$j_\alpha, j_\beta, j_\gamma, j_\delta$	–	a	a	a	a	$j_\alpha + j_\beta + k_{12}$
20.	β	α	γ	δ	$j_\alpha, j_\beta, j_\gamma, j_\delta$	–	a	a	a	a	$j_\beta + j_\gamma + k_{12}$
21.	α	β	δ	γ	$j_\alpha, j_\beta, j_\gamma, j_\delta$	–	a	a	a	a	$j_\beta + j_\gamma$
22.	β	α	δ	γ	$j_\alpha, j_\beta, j_\gamma, j_\delta$	–	a	a	a	a	$j_\beta + j_\gamma$
23.	γ	δ	α	β	$j_\alpha, j_\beta, j_\gamma, j_\delta$	–	a	a	a	a	$j_\alpha + j_\delta + k_{12}$
24.	γ	δ	β	α	$j_\alpha, j_\beta, j_\gamma, j_\delta$	–	a	a	a	a	$j_\alpha + j_\delta + k_{12}$
25.	δ	γ	α	β	$j_\alpha, j_\beta, j_\gamma, j_\delta$	–	a	a	a	a	$j_\alpha + j_\delta$
26.	δ	γ	β	α	$j_\alpha, j_\beta, j_\gamma, j_\delta$	–	a	a	a	a	$j_\alpha + j_\delta$
27.	α	γ	β	δ	$j_\alpha, j_\beta, j_\gamma, j_\delta$	–	a	a	a	a	0
28.	α	γ	δ	β	$j_\alpha, j_\beta, j_\gamma, j_\delta$	–	a	a	a	a	0
29.	γ	α	δ	β	$j_\alpha, j_\beta, j_\gamma, j_\delta$	–	a	a	a	a	0
30.	γ	α	β	δ	$j_\alpha, j_\beta, j_\gamma, j_\delta$	–	a	a	a	a	0
31.	β	δ	α	γ	$j_\alpha, j_\beta, j_\gamma, j_\delta$	–	a	a	a	a	$j_\alpha + j_\beta + j_\gamma + j_\delta$
32.	δ	β	γ	α	$j_\alpha, j_\beta, j_\gamma, j_\delta$	–	a	a	a	a	$j_\alpha + j_\beta + j_\gamma + j_\delta$
33.	β	δ	γ	α	$j_\alpha, j_\beta, j_\gamma, j_\delta$	–	a	a	a	a	$j_\alpha + j_\beta + j_\gamma + j_\delta$
34.	δ	β	α	γ	$j_\alpha, j_\beta, j_\gamma, j_\delta$	–	a	a	a	a	$j_\alpha + j_\beta + j_\gamma + j_\delta$
35.	α	δ	β	γ	$j_\alpha, j_\beta, j_\gamma, j_\delta$	–	a	a	a	a	$1 + j_\gamma + j_\delta - k$
36.	δ	α	γ	β	$j_\alpha, j_\beta, j_\gamma, j_\delta$	–	a	a	a	a	$1 + j_\gamma + j_\delta - k$
37.	α	δ	γ	β	$j_\alpha, j_\beta, j_\gamma, j_\delta$	–	a	a	a	a	$1 + j_\gamma + j_\delta - k_{12}$
38.	δ	α	β	γ	$j_\alpha, j_\beta, j_\gamma, j_\delta$	–	a	a	a	a	$1 + j_\gamma + j_\delta - k_{12}$
39.	β	γ	α	δ	$j_\alpha, j_\beta, j_\gamma, j_\delta$	–	a	a	a	a	$1 + j_\alpha + j_\beta - k$
40.	γ	β	δ	α	$j_\alpha, j_\beta, j_\gamma, j_\delta$	–	a	a	a	a	$1 + j_\alpha + j_\beta - k$
41.	β	γ	δ	α	$j_\alpha, j_\beta, j_\gamma, j_\delta$	–	a	a	a	a	$1 + j_\alpha + j_\beta - k_{12}$
42.	γ	β	α	δ	$j_\alpha, j_\beta, j_\gamma, j_\delta$	–	a	a	a	a	$1 + j_\alpha + j_\beta - k_{12}$

where $U^{(k_1)}(nj)$, $V^{(k_2)}(nj)$ are either of type (14) or type (15). The occupation number N'' is defined by second quantization operators occurring in $U^{(k_1)}(nj)$ and $V^{(k_2)}(nj)$. The module `rabs_rcfp` [11] performs the evaluation of these formulae.

2.3.3. Phase factors Δ

These arise from the reordering necessary to match the recoupled creation and annihilation operators in bra and ket vectors contributing to the matrix element. For each of the cases considered in Table 1 we find

Cases 1–6:

$$\Delta = 0. \tag{20}$$

Cases 7–18:

$$\Delta = 1 + \sum_{r=i}^{j-1} N_r, \tag{21}$$

where N_r is the occupation number of subshell r. If $\alpha < \beta$, then $i = \alpha,\ j = \beta$, and if $\alpha > \beta$, then $i = \beta,\ j = \alpha$.

Cases 19–42:

$$\Delta = \sum_{k=\alpha}^{\beta-1} N_k + \sum_{k=\gamma}^{\delta-1} N_k. \tag{22}$$

2.3.4. The coefficients $\Theta'(n_i l_i j_i, n_j l_j j_j, n_{i'} l_{i'} j_{i'}, n_{j'} l_{j'} j_{j'}, \Xi)$

The effective interaction strength of order k of a two-electron operator, is denoted by

$$\left(n_i l_i j_i n_j l_j j_j \left\| g^{(kk)} \right\| n_{i'} l_{i'} j_{i'} n_{j'} l_{j'} j_{j'}\right)$$

in [7] and by

$$X^k(n_i l_i j_i, n_j l_j j_j, n_{i'} l_{i'} j_{i'}, n_{j'} l_{j'} j_{j'})$$

in [12]. The coefficients $\Theta'(n_i l_i j_i, n_j l_j j_j, n_{i'} l_{i'} j_{i'}, n_{j'} l_{j'} j_{j'}, \Xi)$ for the different cases tabulated in Table 1) have different multiplicative factors defined as follows:

Case 1: Single subshell ($\alpha\alpha\alpha\alpha$)

$$\begin{aligned}&\Theta'_{IIa}(n_\alpha l_\alpha j_\alpha, n_\alpha l_\alpha j_\alpha, n_\alpha l_\alpha j_\alpha, n_\alpha l_\alpha j_\alpha, \Xi)\\ &\quad = \frac{1}{2}[k]^{-1/2}\left(n_\alpha l_\alpha j_\alpha n_\alpha l_\alpha j_\alpha \left\| g^{(kk)} \right\| n_\alpha l_\alpha j_\alpha n_\alpha l_\alpha j_\alpha\right)\delta(k_{12}, k)\end{aligned} \tag{23}$$

and

$$\begin{aligned}&\Theta'_{IIb}(n_\alpha l_\alpha j_\alpha, n_\alpha l_\alpha j_\alpha, n_\alpha l_\alpha j_\alpha, n_\alpha l_\alpha j_\alpha, \Xi)\\ &\quad = (-1)^k[j_\alpha]^{-1/2}\left(n_\alpha l_\alpha j_\alpha n_\alpha l_\alpha j_\alpha \left\| g^{(kk)} \right\| n_\alpha l_\alpha j_\alpha n_\alpha l_\alpha j_\alpha\right)\delta(k_{12}, 0).\end{aligned} \tag{24}$$

Cases 2, 3, 11, 12, 27, 29, 31, 32, 35, 36, 39, 40: Subshell assignments $\alpha\beta\alpha\beta$, $\beta\alpha\beta\alpha$, $\beta\gamma\alpha\gamma$, $\gamma\beta\gamma\alpha$, $\alpha\gamma\beta\delta$, $\gamma\alpha\delta\beta$, $\beta\delta\alpha\gamma$, $\delta\beta\gamma\alpha$, $\alpha\delta\beta\gamma$, $\delta\alpha\gamma\beta$, $\beta\gamma\alpha\delta$, $\gamma\beta\delta\alpha$

$$\begin{aligned}&\Theta'(n_i l_i j_i, n_j l_j j_j, n_{i'} l_{i'} j_{i'}, n_{j'} l_{j'} j_{j'}, \Xi)\\ &\quad = (-1)^{\varphi}\frac{1}{2}[k]^{-1/2}\left(n_i l_i j_i n_j l_j j_j \left\| g^{(kk)} \right\| n_{i'} l_{i'} j_{i'} n_{j'} l_{j'} j_{j'}\right)\delta(k_{12}, k).\end{aligned} \tag{25}$$

Cases 6, 15–26: Subshell assignments $\alpha\alpha\beta\beta$, $\gamma\gamma\alpha\beta$, $\gamma\gamma\beta\alpha$, $\alpha\beta\gamma\gamma$, $\beta\alpha\gamma\gamma$, $\alpha\beta\gamma\delta$, $\beta\alpha\delta\gamma$, $\alpha\beta\delta\gamma$, $\beta\alpha\gamma\delta$, $\gamma\delta\alpha\beta$, $\delta\gamma\beta\alpha$, $\gamma\delta\beta\alpha$, $\delta\gamma\alpha\beta$

$$\Theta'(n_i l_i j_i, n_j l_j j_j, n_{i'} l_{i'} j_{i'}, n_{j'} l_{j'} j_{j'}, \Xi)$$
$$= (-1)^{1+k+\varphi} \frac{1}{2}[k_{12}]^{1/2} \begin{Bmatrix} j_i & j_{i'} & k \\ j_{j'} & j_j & k_{12} \end{Bmatrix} \left(n_i l_i j_i n_j l_j j_j \| g^{(kk)} \| n_{i'} l_{i'} j_{i'} n_{j'} l_{j'} j_{j'}\right). \quad (26)$$

Cases 4, 5, 7, 8, 9, 10, 13, 14, 28, 30, 33, 34, 37, 38, 41, 42: Subshell arrangements $\alpha\beta\beta\alpha$, $\beta\alpha\alpha\beta$, $\beta\alpha\alpha\alpha$, $\alpha\beta\alpha\alpha$, $\beta\beta\beta\alpha$, $\beta\beta\alpha\beta$, $\gamma\beta\alpha\gamma$, $\beta\gamma\gamma\alpha$, $\alpha\gamma\delta\beta$, $\gamma\alpha\beta\delta$, $\beta\delta\gamma\alpha$, $\delta\beta\alpha\gamma$, $\alpha\delta\gamma\beta$, $\delta\alpha\beta\gamma$, $\beta\gamma\delta\alpha$, $\gamma\beta\alpha\delta$

$$\Theta'(n_i l_i j_i, n_j l_j j_j, n_{i'} l_{i'} j_{i'}, n_{j'} l_{j'} j_{j'}, \Xi)$$
$$= (-1)^{\varphi} \frac{1}{2}[k_{12}]^{1/2} \begin{Bmatrix} j_i & j_{i'} & k \\ j_j & j_{j'} & k_{12} \end{Bmatrix} \left(n_i l_i j_i n_j l_j j_j \| g^{(kk)} \| n_{i'} l_{i'} j_{i'} n_{j'} l_{j'} j_{j'}\right). \quad (27)$$

The phase factors φ in expressions (25)–(27) are defined in column φ of Table 1. This construction exploits the common tensorial structure of any scalar two-electron operators as the Coulomb, Breit and Gaunt interactions [12] and exploits this similarity to simplify the calculation of spin-angular coefficients. The relativistic jj-coupling expressions for the effective interaction strength of the Coulomb interaction [12, Eq. (86)] is

$$\left(n_i l_i j_i n_j l_j j_j \| g^{(kk)} \| n_{i'} l_{i'} j_{i'} n_{j'} l_{j'} j_{j'}\right)$$
$$= (-1)^k \langle n_i l_i j_i \| C^{(k)} \| n_{i'} l_{i'} j_{i'} \rangle \langle n_j l_j j_j \| C^{(k)} \| n_{j'} l_{j'} j_{j'} \rangle R^k(n_i l_i j_i n_j l_j j_j n_{i'} l_{i'} j_{i'} n_{j'} l_{j'} j_{j'}). \quad (28)$$

We can now identify the coefficients $V^k_{rs}(abcd)$ of (3) by substituting the results above in (13). The same construction can be used for the Gaunt interaction (the leading part of the magnetic Breit interaction) [12, Eq. (91)] and for the full transverse Breit interaction [12, Eq. (101)], although the selection rules and the effective interaction strengths corresponding to (27) are, of course, different. The pure angular coefficients $v^k_{rs}(abcd)$ for two-electron operators are the same for all these operators since the $v^k_{rs}(abcd)$ can be identified by inserting

$$\left(n_i l_i j_i n_j l_j j_j \| g^{(kk)} \| n_{i'} l_{i'} j_{i'} n_{j'} l_{j'} j_{j'}\right) = 1$$

in (13).

3. Program organization

3.1. Overview of the program

The program ANCO constructs the pure angular coefficients $t^0_{rs}(ab)$ for one-electron operators and the $v^k_{rs}(abcd)$ coefficients contributing to matrix elements of the Dirac–Coulomb–Breit Hamiltonian. The coefficients $T_{rs}(ab)$ and $V^k_{rs}(abcd)$ used in GRASP92 and earlier version of the system are available as an option. The new format generates what we have called "pure" angular momentum coefficients which can be used unchanged with any one-particle tensor operator of rank 0, and any two-particle interaction. The Coulomb and Breit interactions use different subsets of the complete set of $v^k_{rs}(abcd)$ coefficients, which are selected automatically when multiplying by the relevant effective interaction strengths to complete the matrix element calculation. The MCP and MCBP modules of GRASP92 calculated the full matrix elements for each of these subsets, so that the new formulation reduces the computational overheads and the memory requirements, which renders ANCO more suitable for large scale problems.

There are two new modules `rabs_recoupling` and `rabs_anco` for extracting spin-angular coefficients relating to formula (13). The module `rabs_recoupling` evaluates recoupling coefficients $R(j_i, j_j, \Lambda^{\text{bra}}, \Lambda^{\text{ket}})$ and $R(j_i, j_j, j_{i'}, j_{j'}, \Lambda^{\text{bra}}, \Lambda^{\text{ket}}, \Gamma)$ as described in [6], module `rabs_rcfp` [11] evaluates the $T(n_i j_i, n_j j_j, n_{i'} j_{i'}, n_{j'} j_{j'}, \Lambda^{\text{bra}}, \Lambda^{\text{ket}}, \Xi, \Gamma)$ whilst `rabs_anco` evaluates all the contributions to (13) for both scalar one- and two-particle operators.

The program ANCO can be run in two modes. The interactive mode is intended for testing the program and for performing short calculations involving a small number of configurations. Normally the modules `rabs_anco`

and `rabs_recoupling` will be interfaced to GRASP92 or to some other program with compatible data structure to perform multiconfiguration or configuration–interaction relativistic calculations. The program can handle any combination of open subshells with $j \leqslant 9/2$, but subshells with $j > 9/2$ are only allowed if they contain not more than two electrons.

ANCO is written in Fortran 90/95 and is designed as an addition to the RATIP package [13]. The new Fortran 90/95 standard enables us to define new derived data types which will enable us to incorporate this module in our continuing development of large-scale computations for open-shell atoms and ions. The full definition of the various derived structures can be found in the module header of `rabs_anco`. Here we need only those types which concern the program output.

All information about pure spin-angular coefficients of scalar one- and two-particle operators ($t_{rs}(ab)$ and $v^k_{rs}(abcd)$ coefficients) of the Hamiltonian matrix or of some part of it is summarized in the derived data `anco_pair_list`.

```
type(anco_csf_pair), dimension(:), pointer :: anco_pair_list
```

which is defined by

```
type :: anco_csf_pair
   integer          :: r, s
   integer          :: no_t_coeff, no_v_coeff
   type(anco_t_coeff), dimension(:), pointer :: t_coeff
   type(anco_v_coeff), dimension(:), pointer :: v_coeff
end type anco_csf_pair
```

The integers `r` and `s` respectively index the bra- and the ket-configuration state functions (CSF) for the current matrix element. The variable `no_t_coeff` is the number of pure spin-angular coefficients of one-particle operators that can be constructed for the pair `r,s`, and the variable `no_v_coeff` gives corresponding data for two-particle matrix elements. The array `anco_v_coeff` contains pure spin-angular coefficients for two-particle scalar operators. It is defined by

```
type :: anco_v_coeff
   integer          :: k
   type(nkappa)     :: a, b, c, d
   real(kind=dp)    :: v
end type anco_v_coeff
```

where `k` is the rank k of the effective interaction strength, `a, b, c, d` point to the relevant subshells $n_i l_i j_i$, $n_j l_j j_j$, $n_{i'} l_{i'} j_{i'}$, $n_{j'} l_{j'} j_{j'}$, and the pure spin-angular coefficient itself is given in `v`. The array `anco_t_coeff` is defined in the same way.

Memory for the array `anco_pair_list` is allocated dynamically using the

```
allocate( anco_pair_list(1:number_of_pair_list_max))
```

instruction, and can be deallocated subsequently.

3.2. Interactive calculations

A typical interactive dialog for calculating spin-angular coefficients is shown in Fig. 1.

```
Enter a file name for the anco.sum file:
```

```
 ANCO: Calculation of angular coefficients for symmetry-adapted CSF functions
  from the GRASP92 structure program (Fortran 90 version)
  (C) Copyright by G Gediminas and others, Kassel (2000).

 Enter a file name for the  anco.sum  file:
test.sum
 Enter the name of the configuration symmetry list file:
argon-sd.inp
 Loading configuration symmetry list file ...
  There are  16  relativistic subshells;
  there are  761  relativistic CSFs;
  ... load complete.
 Generate only non-trivial angular coefficients which include
 (at least one) open shells ?
y
 Generate one-electron angular coefficients for scalar interactions ?
y
  Generate GRASP92-like T coefficients for scalar interactions ?
y
 Generate two-electron angular coefficients for scalar interactions ?
y
  Generate GRASP92-like V^k coefficients for scalar interactions ?
y
 Enter a file name for the  anco.vnu file:
test.vnu
```

Fig. 1. The typical interactive dialog for calculation spin-angular coefficients.

After this prompt, the user should insert the output file name to which the main output data will be written. This must be followed by the name of the input file listing CSF in the GRASP92 format. The next question

```
Generate only not trivial angular coefficients which include (at least one)
open shells ?
```

The response `y` or `Y` will cause the program to calculate coefficients for peel shells only; the reponse `n` or `N` the program calculate will yield data for all shells (open and closed).

The question

```
Generate one-electron angular coefficients for scalar interactions ?
```

This requires answer `n` or `N` if one-electron coefficients are not needed. If the user responds `y` or `Y`, then the prompt

```
Generate GRASP92-like T coefficients for scalar interactions ?
```

appears. The response `y` or `Y` causes GRASP92-like $T_{rs}(ab)$ coefficients to be generated, whereas the alternative `n` or `N` yields $t^0_{rs}(ab)$ coefficients.

A similar dialog follows for two-electron angular coefficients. A number of examples illustrate the usage of ANCO in this mode in Section 4 below.

The prompt

```
Enter a file name for the anco.vnu file:
```

permits the user to specify where the spin-angular coefficients should be stored.

The module ANCO needs one input file `.csl` containing the CSF list output by GRASP92 [9], which is generated by the program GENCSL.

The program creates two output files. The file `.sum` contains the summary of the problem. The other file `.vnu` contains the angular momentum coefficients and their characteristics. Its format for $T_{rs}(ab)$ and $V^k_{rs}(abcd)$ coefficients is the same as that of file `genmcp.dbg` generated by the module `genmcp` of GRASP92 [9]. The format for $t^0_{rs}(ab)$ and $v^k_{rs}(abcd)$ is very similar but without the sorting process which is used in GRASP92.

3.3. Distribution and installation of the program

As a new component of the RATIP package [13] similar to the module RCFP [11] ANCO will be distributed as a `tar` archive file of the directory `ratip_anco`. On a UNIX (or compatible) workstation, the command `tar -xvf ratip_anco.tar` reconstructs the file structure. The directory `ratip_anco` then contains the Fortran 90/95 modules `rabs_anco.f` and `rabs_recoupling.f`, the program `xanco.f` (the main program for interactive work) as well as the makefile `make-anco`. It also includes a number of examples in the subdirectory `test-anco` and a short `Read.me` which explains further details about the installation. Since the same file structure is preserved in both cases, the combination of ANCO with RATIP is simply achieved by running the command `cp -r ratip_anco/. ratip/.` inside the RATIP root directory; then `make -f make-anco` will generate the executable `xanco`, together with the other two components `xcesd99` [14] and `xreos99` [13] of the RATIP package. The name of the (Fortran 90/95) compiler and special compiler flags can be overwritten in the header of the makefile as necessary. Although ANCO uses six other modules which are part already of RATIP, no further adaptation of the program is needed. At present, the ANCO program has been installed and tested under the Linux and AIX operating systems but, owing to the compliance of the Fortran 90/95 standard, no difficulties should arise on any other platform.

The subdirectory `test-anco` lists a number of examples which demonstrate the usage of the program.

4. Timing and verification of ANCO

Tests and timing studies using the Dirac–Coulomb Hamiltonian only were performed for the $3s^2 3p^6\ {}^1S$ state of Ar I with the common closed shells $1s^2 2s^2 2p^6$ for different values of final orbital momentum J. The wave function expansions used were:

(1) 3SD: Single and double excitations from $3s^2 3p^6$ to the active set $\{3s, 3p, 3d\}$ contains 14 configuration state functions (CSF) for $J = 0$ and 34 CSF (the maximum) for $J = 2$.
(2) 3SDT: Single, double and triple excitations from $3s^2 3p^6$ to the active set $\{3s, 3p, 3d\}$. The maximum number of CSF is 145 for $J = 2$.
(3) 4SD: Single and double excitations from $3s^2 3p^6$ to the active set $\{3s, 3p, 3d, 4s, 4p, 4d, 4f\}$. The maximum number of CSF is 465 for $J = 2$.
(4) 4SDT: Single, double and triple excitations from $3s^2 3p^6$ to the active set $\{3s, 3p, 3d, 4s, 4p, 4d, 4f\}$.
(5) 5SD; Single and double excitations from $3s^2 3p^6$ to the active set $\{3s, 3p, 3d, 4s, 4p, 4d, 4f, 5s, 5p, 5d, 5f, 5g\}$.

We first considered simple cases with a small number of CSF (3SD, 3SDT, 4SD with J = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9). Although ANCO generates the full set of "pure" coefficients for both one- and two-particle operators, the calculation runs from 1.4–2.3 times faster than an equivalent calculation with GRASP92 because of the lower computational overheads. Table 2 demonstrates similar enhanced performance for the much larger 4SDT and 5SD eamples, showing the improvement expected for large-scale calculations.

From the results presented in the Table 2 we conclude that the new program is not much faster in simple cases, but does better in more complicated cases. The fact that ANCO calculates approximately twice the number of angular coefficients as GRASP92 reduces the effective cost per coefficient by a further factor of two.

Table 2
Timing Comparison for GRASP92 and ANCO codes. Times are given in hours, minutes, seconds

ASF expan.	Number of CSF	$T^k_{rs}(ab)$ or $t^k_{rs}(ab)$	$V^k_{rs}(abcd)$	$v^k_{rs}(abcd)$	Running time of GRASP92	ANCO	Speed -up
4SDT ($J=0$)	2149	3606	756 023	1 530 086	00:08:11	00:03:01	2.7
4SDT ($J=1$)	5786	14 017	4 070 156	8 188 130	00:59:01	00:15:19	3.9
4SDT ($J=2$)	8016	21 356	7 018 885	14 077 044	01:42:47	00:26:28	3.9
4SDT ($J=3$)	8378	21 342	7 634 136	15 290 955	01:53:55	00:30:37	3.7
4SDT ($J=4$)	7284	15 971	6 111 074	12 260 139	01:33:17	00:23:01	4.1
4SDT ($J=5$)	5349	9435	3 810 165	7 656 054	00:50:27	00:14:18	3.6
4SDT ($J=6$)	3370	4556	1 836 602	3 706 544	00:21:52	00:06:40	3.3
4SDT ($J=7$)	1788	1789	693 761	1 412 443	00:07:26	00:02:29	3.0
5SD ($J=0$)	468	621	75 192	150 455	00:00:32	00:00:17	1.9
5SD ($J=1$)	1134	2324	395 450	792 560	00:03:10	00:01:29	2.1
5SD ($J=2$)	1609	3704	697 651	1 395 839	00:06:27	00:02:44	2.4
5SD ($J=3$)	1584	3441	721 907	1 444 095	00:06:43	00:02:59	2.3
5SD ($J=4$)	1361	2500	558 223	1 117 681	00:05:15	00:02:17	2.3
5SD ($J=5$)	920	1361	314 909	632 306	00:02:30	00:01:22	1.8
5SD ($J=6$)	559	644	141 328	284 102	00:01:02	00:00:36	1.7
5SD ($J=7$)	259	226	44 137	89 398	00:00:15	00:00:12	1.3

Although the program is completely new, we have verified that the results presented agree completely with those obtained from GRASP92. We have also verified that the Breit interaction is treated correctly, although no data are presented here.

5. Examples

To illustrate the use of ANCO in its interactive mode, we studied Ar II. The program `genmcp` of GRASP92 [9] is used first to generate the CSF file `argon-sd.inp` before running the program `xanco`.

The first example does a GRASP92-style calculation. After checking all triangular conditions for $X^k_{\text{Coulomb}}(\alpha\beta\gamma\delta)$ (see expression (88) in [12]), it multiplies each pure-two particle coefficient by the factor $X^k_{\text{Coulomb}}(\alpha\beta\gamma\delta)$ and prints all non zero $V^k_{rs}(abcd)$ coefficients. With corresponding $T_{rs}(ab)$ coefficients, this generates a total of 11 279 coefficients. The second example, with the same input, calculates a total of 433 911 pure non-trivial spin-angular coefficients at one go, as is more convenient for large-scale calculations. However, only 11 279 coefficients from this set are required in the first example.

The Test Run Output displays the `.sum` files and the first 10 lines of `.vnu` files for both examples.

References

[1] C.F. Fischer, The Hartree–Fock Method for Atoms, Wiley, New York, 1977.
[2] I.P. Grant, M. Quiney, Adv. in Atomic Molecular Phys. 23 (1987) 37.
[3] I.P. Grant, Comput. Phys. Commun. 5 (1973) 264.

[4] I.P. Grant, Comput. Phys. Commun. 11 (1976) 397.
[5] N. Beatham, I.P. Grant, B.J. McKenzie, Comput. Phys. Commun. 18 (1979) 245.
[6] G. Gaigalas, Z. Rudzikas, C.F. Fischer, J. Phys. B: At. Mol. Opt. Phys. 30 (1997) 3747.
[7] G. Gaigalas, Lithuanian J. Phys. 39 (1999) 80.
[8] G. Gaigalas, S. Fritzsche, Z. Rudzikas, At. Data Nucl. Data Tables (accepted).
[9] F.A. Parpia, C.F. Fischer, I.P. Grant, Comput. Phys. Commun. 94 (1996) 249.
[10] G. Gaigalas, Z. Rudzikas, J. Phys. B: At. Mol. Phys. 29 (1996) 3303.
[11] G. Gaigalas, S. Fritzsche, Comput. Phys. Commun. 134 (1) (2001) 86.
[12] I.P. Grant, in: S. Wilson (Ed.), Methods in Computational Chemistry, Vol. 2, Plenum, New York, 1988, p. 1.
[13] S. Fritzsche, C.F. Fischer, C.Z. Dong, Comput. Phys. Commun. 124 (2000) 340.
[14] S. Fritzsche, J. Anton, Comput. Phys. Commun. 124 (1999) 353.

TEST RUN OUTPUT

```
.........................Example 1.........................

>>type example1.sum
 ANCO run at 15:50:39 on Feb 14 2000.

There are  17  electrons in the cloud
 in  761  relativistic CSFs
 based on  16  relativistic subshells.
 Total number of pair is: 289941

Generate only not trivial angular coefficients
 there are 716  Grasp92-like T coefficients
 there are 112078  Grasp92-like Vk coefficients
 the total number of coefficients is 112794

>> type example1.vnu            (first 10 lines of example1.vnu file)
ANCO
761 16 4
V^[( 2)]_[    1,    1] ( 3p , 3p ; 3p , 3p ) =  -1.200000000000E-01
V^[( 1)]_[    1,    1] ( 3p , 3d-; 3d-, 3p ) =   6.666666666667E-02
V^[( 3)]_[    1,    1] ( 3p , 3d-; 3d-, 3p ) =  -2.571428571429E-01
V^[( 1)]_[    2,    1] ( 3p , 3d ; 3d-, 3p ) =  -1.632993161855E-01
V^[( 2)]_[    3,    1] ( 3p , 4s ; 3p , 3d-) =   2.529822128135E-01
V^[( 1)]_[    3,    1] ( 3p , 4s ; 3d-, 3p ) =  -2.108185106779E-01
T^[    6,    1] ( 4d-, 3d-) =    1.000000000000E+00
V^[( 0)]_[    6,    1] ( 1s , 4d-; 1s , 3d-) =   2.000000000000E+00

.........................Example 2.........................

>>type example2.sum
ANCO run at 16:35:02 on Feb 14 2000.

There are  17  electrons in the cloud
 in  761  relativistic CSFs
 based on  16  relativistic subshells.
 Total number of pair is: 289941

Generate only not trivial angular coefficients
 there are 716  pure one-particle angular coefficients
 there are 433195  pure two-particle angular coefficients
 the total number of coefficients is 433911

>> type example2.vnu             (first 10 lines of example2.vnu file)

ANCO
761 16 4
 pure two-particle [( 1)]_[  1,  1] ( 3p , 3p ; 3p , 3p ) =  5.000000000000E-02
 pure two-particle [( 2)]_[  1,  1] ( 3p , 3p ; 3p , 3p ) = -1.500000000000E-01
 pure two-particle [( 3)]_[  1,  1] ( 3p , 3p ; 3p , 3p ) =  5.000000000000E-02
 pure two-particle [( 1)]_[  1,  1] ( 3p , 3d-; 3p , 3d-) =  3.000000000000E-01
```

```
pure two-particle [( 3)]_[  1,  1] ( 3p , 3d-; 3p , 3d-) = -2.000000000000E-01
pure two-particle [( 0)]_[  1,  1] ( 3p , 3d-; 3d-, 3p ) = -2.500000000000E-01
pure two-particle [( 1)]_[  1,  1] ( 3p , 3d-; 3d-, 3p ) = -2.500000000000E-01
pure two-particle [( 2)]_[  1,  1] ( 3p , 3d-; 3d-, 3p ) = -1.500000000000E-01
```