

13. Cubric 1992 J. Phys. B: At. Mol. Opt. Phys. **25**, 5069–83.
14. Eckhard M. and Schartner K.-H. 1983. Z. Phys. A – Atoms and Nuclei **312**, 321–328.
15. El Sherbini Th. M. and Van der Wiel M. J. 1972. Physica **62**, 119.
16. Hall R. J., McConkey A., Ellis K., Dawber G., McDonald M. A. and King G. C. 1992 J. Phys. B: At. Mol. Opt. Phys. **25**, 799–810.
17. Hayaishi et al. 1984. J. Phys. B: At. Mol. Opt. Phys. **17**, 3511–27.
18. Hayaishi et al. 1990. J. Phys. B: At. Mol. Opt. Phys. **23**, 1633–39.
19. Hayaishi et al. 1996. Phys. Rev. A **54**, 4064–8.
20. Holland D. M. P., Codling K., West J. B. and Marr G. V. 1979. J. Phys. B: Atom. Molec. Phys. **12**, 2465–84.
21. Huetz A., Salles P., Waymel D. and Mazeau J. 1991. J. Phys. B: At. Mol. Opt. Phys. **24**, 1917–33.
22. Kelly H. P. Adv. Theor. 1968. Phys. **2**, 75.
23. Kraessig B. 1996. J. Phys. B: At. Mol. Opt. Phys. **29**, 4255–65.
24. Kraessig B. and Schmidt V. 1992. J. Phys. B: At. Mol. Opt. Phys. **25**, L327–33.
25. Krause M. O., Carlson T. A. and Dismukes R. D. 1968. Phys. Rev. **170**, 37–47.
26. Lablanquie P., Eland J. H. D., Nenner I., Morin P., Delwiche J. and Hubin-Franskin M.-J. 1987. Phys. Rev. Lett. **58**, 992–995.
27. Lightner G. S., Van Brunt R. J. and Whitehead W. D. 1971. Phys. Rev. **A4**, 602–9.
28. Samson J. A. R. and Haddad G. N. 1974. Phys. Rev. Lett. **33**, 875.
29. Samson J. A. R., Stolte W. C., He Y.-X., Cutler J. N. and Lu. Y. 1998. Phys. Rev. A **57**, 1906–11.
30. Schartner K.-H., Lenz P., Moebus B. and Magel B. 1990. Physica Scripta, **41**, 853–856.
31. Schartner K.-H., Mentzel G., Magel B., Moebus B., Ehresmann A., Vollweiler F. and Schmoranzner H. 1993. J. Phys. B: At. Mol. Opt. Phys. **26**, L445–50.
32. Schmidt V. 1992. Rep. Prog. Phys. **55**, 1483–659.
33. Schmidt V., Sander N., Kuntzemueller H., Dhez P., Wuilleumier F. and Kaellne E. 1976 Phys. Rev. **A13**, 1748.
34. Wight G. R. and Van der Wiel M. J. 1976. J. Phys. B: Atom. Molec. Phys. **9**, 1319–27.
35. Wuilleumier F. and Krause M. O. 1974. Phys. Rev. **A10**, 242–58.
36. Yanghui Qiu, Jian-yhi Tang Joachim Burgdaefer and Jianyi Wang 1998. Phys. Rev. A **57**, R1489–92.

Д.А. Лазарев, Дм.А. Лазарев, В.А. Килин, В.М. Зеличенко

АВТОМАТИЗАЦИЯ РАБОТЫ С ОБЪЕКТАМИ КВАНТОВОЙ ТЕОРИИ УГЛОВОГО МОМЕНТА

Введение

В приложениях квантовой механики к исследованию атомов часто встречается задача вычисления многократных сумм произведений коэффициентов Клебша-Гордана (ККГ) и $3jm$ -символов Вигнера. В приближении сферической симметрии поля, создаваемого ядром и остальными электронами, квантовые числа орбитального момента и его проекции на некоторое направление являются точными. Дальнейший учет электростатического и спинорбитального взаимодействий оставляет точным лишь полный угловой момент атома. Собственные волновые функции оператора полного момента строятся как линейные комбинации произведений одноэлектронных волновых функций [1]. Коэффициенты этих линейных комбинаций, как известно, представляют произведения коэффициентов

Клебша-Гордана или связанных с ними $3jm$ -символов Вигнера. Поэтому выражения для матричных элементов различных операторов содержат многократные суммы ККГ и $3jm$ -символов. Такие суммы в большинстве случаев можно привести к менее громоздкому виду путем выделения из них инвариантных по отношению к вращениям величин типа $3nj$ -символов Вигнера. Для упрощения и наглядного представления подобных преобразований разработан графический метод [2, 3, 4]. Однако и в этом виде задача остается весьма трудоемкой. Поэтому желательно каким-либо образом автоматизировать процесс преобразования. В настоящей статье описывается программа QAMT, предназначенная для решения поставленной задачи.

Теоретические сведения

В целях единообразия все ККГ удобно преобразовать в $3jm$ -символы. В основу формального процесса

преобразования сумм произведений $3jm$ -символов могут быть положены три соотношения [3, гл. 12]:

$$\sum_a (-1)^{a-\alpha} \begin{pmatrix} a & a & c \\ \alpha & -\alpha & \gamma \end{pmatrix} = \Pi_a \delta_{c0} \delta_{\gamma 0}, \quad (1)$$

$$\sum_{\phi\chi} (-1)^{p-\phi+q-\chi} \begin{pmatrix} a & p & q \\ -\alpha & \phi & \chi \end{pmatrix} \begin{pmatrix} p & q & a' \\ -\phi & -\chi & \alpha' \end{pmatrix} = \frac{(-1)^{a-\alpha}}{\Pi_a^2} \delta_{a\alpha'} \delta_{a\alpha'}. \quad (2)$$

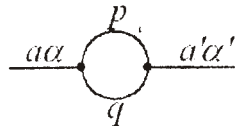
$$\sum_x (-1)^{q-x} \begin{pmatrix} a & b & q \\ \alpha & \beta & -\chi \end{pmatrix} \begin{pmatrix} q & d & c \\ \chi & \delta & \gamma \end{pmatrix} = (-1)^{2a} \sum_x (-1)^{x-\xi} \Pi_x^2 \begin{pmatrix} a & c & x \\ \alpha & \beta & -\xi \end{pmatrix} \begin{pmatrix} x & d & b \\ \xi & \delta & \beta \end{pmatrix} \begin{Bmatrix} b & d & x \\ c & a & q \end{Bmatrix}. \quad (3)$$

Обозначения, примененные здесь и далее, полностью соответствуют принятым в каноническом справочнике [3], поэтому мы не будем останавливаться на

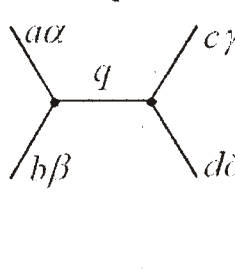
их описании. В упрощенном варианте графической техники квантовой теории углового момента выражения (1)-(3) имеют следующий вид:



$$= \Pi_a \delta_{c0} \delta_{\gamma 0}, \quad (1a)$$



$$= \frac{(-1)^{a-\alpha}}{\Pi_a^2} \delta_{aa'} \delta_{\alpha\alpha'}, \quad (2a)$$



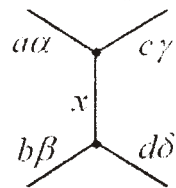
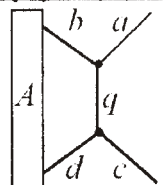
$$= (-1)^{2a} \sum_x \Pi_x^2 \begin{Bmatrix} b & d & x \\ c & a & q \end{Bmatrix} \times \quad (3a)$$


Рис. 1

Каждому 3jm-символу соответствует вершина диаграммы, каждому моменту – линия, по проекции момента, соединяющего две вершины, подразумевается суммирование.

цепи, например к моменту q , сокращает число звеньев этой цепи на единицу. Таким образом, последовательное применение (3a) в конце концов приводит цепь к соотношению ортогональности (2a), в результате использования которого число 3jm-символов в выражении уменьшается на два. Последовательная обработка всех имеющихся в диаграмме замкнутых цепей позволяет избавиться от большинства 3jm-символов.

Рассмотрим некоторую совокупность соединенных друг с другом вершин, образующих замкнутую цепь из некоторого числа звеньев. На рис. 2 показан фрагмент такой цепи $b-q-d$, замыкающийся где-то в блоке A . Применение формулы (3a) к любому из моментов



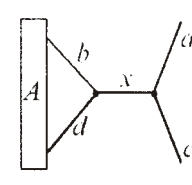
$$= (-1)^{2a} \sum_x \Pi_x^2 \begin{Bmatrix} b & d & x \\ c & a & q \end{Bmatrix} \times \quad (2)$$


Рис. 2

После удаления какой-либо цепи возможно возникновение суммирования по проекции момента, входящего только в один 3jm-символ. Такое суммирование может быть проведено при помощи (1a). Даль-

нейшее упрощение в ряде случаев может быть достигнуто преобразованием сумм произведений 6j-символов в 9j-символы и применением соотношения ортогональности 6j-символов [3, гл. 8].

Описание программы

Программа QAMT выполняет:

- упрощение сумм произведений ККГ и 3jm-символов в аналитическом виде;
- аналитический вывод выражений для угловых множителей диаграмм многочастичной теории возмущений;
- вычисление значений выражений при заданных угловых моментах;
- генерацию программного кода на языке Fortran для любых аналитических выражений.

Программа не имеет пользовательского интерфейса, работая с файлами специального формата. Имя входного файла данных передается программе в командной строке, а результат обработки записывается в файл с тем же именем и расширением «.txt».

Входной файл данных содержит некоторый набор команд, определяющих, какие данные и каким способом должны быть обработаны. Команда представляет собой зарезервированное слово, начинающее новую строку в файле, за которым может следовать перечень

аргументов. Команда и ее аргументы могут располагаться на нескольких строках. Строка (строки) команды должны оканчиваться точкой с запятой. Подстановка произвольного, кроме пробела и табуляции, символа перед командным словом исключает выполнение команды. Таким образом, любая строка, начинающаяся не с командного слова, считается комментарием. Команда и ее аргументы отделяются друг от друга одним или несколькими символами пробела.

Объектом, непосредственно содержащим данные, является выражение, или, на языке графической техники, «диаграмма». В любом расчете прежде всего должна быть создана диаграмма с некоторым именем *Name* при помощи команды `create_new_diagram [Name]`; При отсутствии аргумента *Name* диаграмма будет названа по умолчанию. Программа позволяет создать в одном файле до 20 диаграмм. Для загрузки данных в диаграмму предназначены команды, приведенные в табл. 1.

Таблица 1

Команда	Аналитическое выражение
<code>load_data;</code>	Инициализирует режим загрузки
<code>sums_over_momenta a b ... c;</code>	Сумма по моментам a, b, \dots, c . $\sum_{ab\dots c}$
<code>sums_over_projections a b ... c;</code>	Сумма по проекциям моментов a, b, \dots, c . $\sum_{m_a m_b \dots m_c}$
<code>momenta_phase (i) a (j) b ... (k) c;</code>	Фаза, содержащая моменты a, b, \dots, c . $(-1)^{i a + j b + \dots + k c}$. Здесь i, j, k – целые числа.
<code>projections_phase (i) a (j) b ... (k) c;</code>	Фаза, содержащая проекции моментов a, b, \dots, c . $(-1)^{i m_a + j m_b + \dots + k m_c}$. Здесь i, j, k – целые числа.
<code>[j]_factors (i) a (j) b ... (k) c;</code>	Множитель вида $(\sqrt{2a+1})^i (\sqrt{2b+1})^j \dots (\sqrt{2c+1})^k$ $\equiv \prod_a^i \prod_b^j \dots \prod_c^k$ Здесь i, j, k – целочисленные показатели, с которыми множители входят в выражение.
<code>clebsh_gordan_coeffs < a + b + Lab + ></code> <code>... < a - c + Lac + ></code>	Коэффициенты Клебша-Гордана $\langle a m_a b m_b L_{ab} M_{ab} \rangle \dots \langle a - m_a c m_c L_{ac} M_{ac} \rangle$ Знаки соответствуют знакам, с которыми входят соответствующие моменты проекции.
<code>3jm_symbols a b c ... a b d </code> <code> + + + - - + </code>	3jm-символы Вигнера $\begin{pmatrix} a & b & c \\ m_a & m_b & m_c \end{pmatrix} \dots \begin{pmatrix} a & b & d \\ -m_a & -m_b & m_d \end{pmatrix}$ Знаки соответствуют знакам, с которыми входят соответствующие моменты проекции.
<code>6j_symbols a b c ... a b d </code> <code> A B C E F G </code>	6j-символы Вигнера $\begin{Bmatrix} a & b & c \\ A & B & C \end{Bmatrix} \dots \begin{Bmatrix} a & b & d \\ E & F & G \end{Bmatrix}$
<code>9j_symbols a b c ... j11 j12 j13 </code> <code> A B C j21 j22 j23 </code> <code> E F G j31 j32 j33 </code>	9j-символы Вигнера $\begin{Bmatrix} a & b & c \\ A & B & C \\ E & F & G \end{Bmatrix} \dots \begin{Bmatrix} j_{11} & j_{12} & j_{13} \\ j_{21} & j_{22} & j_{23} \\ j_{31} & j_{32} & j_{33} \end{Bmatrix}$
<code>dipole_me < a s_a + D b s_b + ></code>	Матричный элемент оператора дипольного электрического перехода в форме длины. Знаки «+» соответствуют частичным, «-» – дырочным состояниям. $\langle a m_a s_a \mu_a D b m_b s_b \mu_b \rangle$.
<code>coulomb_me < a s_a + b s_b + V^k </code> <code> c s_c - d s_d - ></code>	Матричный элемент k -го члена разложения оператора кулоновского взаимодействия по полиномам Лежандра $\langle a m_a s_a \mu_a b m_b s_b \mu_b V^k c m_c s_c \mu_c d m_d s_d \mu_d \rangle$ Знаки «+» соответствуют частичным, «-» – дырочным состояниям.
<code>get_numerical_data (j, m) l;</code> <code>end_data;</code>	Устанавливает значения момента $l=j$ и его проекции $m=l$. Завершает ввод данных.

Имена угловых моментов могут быть произвольными словами длиной до восьми символов. Имена должны быть уникальными, т. е. в диаграмме не может быть два разных момента с одинаковыми именами. Даже если два момента заведомо равны, им необходимо присвоить разные имена. Программа различает верхний и нижний регистры в именах моментов, в то время как все команды набираются только в нижнем регистре.

После ввода данных диаграмма готова к преобразованиям, которые выполняются следующими операционными командами. Упрощение диаграммы вызывается командой `reduce`; , а вычисление значения – командой `evaluate`; . На любом этапе расчета диа-

грамма может быть выведена в выходной файл командой `output`; .

Операционные команды и команда `load_data`; действуют на так называемую активную диаграмму. После создания новой диаграммы она автоматически становится активной. Любую диаграмму *Name* из уже созданных можно сделать активной с помощью команды `set_active_daigram Name`; .

Результат работы программы записывается в выходной текстовый файл. Таб. 3 устанавливает соответствие между текстовыми данными выходного файла и соответствующими аналитическими выражениями.

Таблица 2

Команда	Назначение
<code>stop</code> ;	Останавливает обработку входного файла, полезна при отладке входных данных
<code>fortran</code> ;	Вызывает генерацию кода на языке Fortran 90 при каждом применении команды <code>output</code> ;
<code>noecho</code> ;	Отключает отображение команд в выходном файле
<code>delete_diagram [Name]</code> ;	Уничтожает диаграмму, освобождая оперативную память

Таблица 3

Текстовая строка выходного файла	Аналитическое выражение
<code>Sum(a b ... c)</code>	$\sum_{ab...c}$
<code>Phase(ia jb ... kc)</code>	$(-1)^{i+a+j+b+...+k+c}$
<code>[a]i [b]j ... [c]k</code>	$(\sqrt{2a+1})^i (\sqrt{2b+1})^j \dots (\sqrt{2c+1})^k$
<code>< a + b + Lab + > ... < a - c + Lac + ></code>	$\langle a m_a b m_b L_{ab} M_{ab} \rangle \dots \langle a - m_a c m_c L_{ac} M_{ac} \rangle$
<code>D(a,b) ... d(c,d)</code>	$\delta_{a,b} \delta_{c,d}$
<code>(a b c) ... (a b d)</code>	$\begin{pmatrix} a & b & c \\ 0 & 0 & 0 \end{pmatrix} \dots \begin{pmatrix} a & b & d \\ 0 & 0 & 0 \end{pmatrix}$
<code>{ a b c } ... { a b d }</code>	$\Delta(a,b,c) \dots \Delta(a,b,d)$
<code> a b c ... a b d </code> <code> + + + ... - - + </code>	$\begin{pmatrix} a & b & c \\ m_a & m_b & m_c \end{pmatrix} \dots \begin{pmatrix} a & b & d \\ -m_a & -m_b & m_d \end{pmatrix}$
<code> a b c ... a b d </code> <code> A B C ... E F G </code>	$\begin{Bmatrix} a & b & c \\ A & B & C \end{Bmatrix} \dots \begin{Bmatrix} a & b & d \\ E & F & G \end{Bmatrix}$
<code> a b c ... j11 j12 j13 </code> <code> A B C ... j21 j22 j23 </code> <code> E F G ... j31 j32 j33 </code>	$\begin{Bmatrix} a & b & c \\ A & B & C \\ E & F & G \end{Bmatrix} \dots \begin{Bmatrix} j_{11} & j_{12} & j_{13} \\ j_{21} & j_{22} & j_{23} \\ j_{31} & j_{32} & j_{33} \end{Bmatrix}$

Пример использования программы

В качестве примера использования программы QAMT рассмотрим вычисление углового множителя одной из диаграмм первого порядка многочастичной теории возмущений для процесса двойной фотоиони-

зации нейтрального атома с замкнутыми оболочками [5]. Фейнмановская диаграмма и соответствующее ей аналитическое выражение приведены на рис. 4.

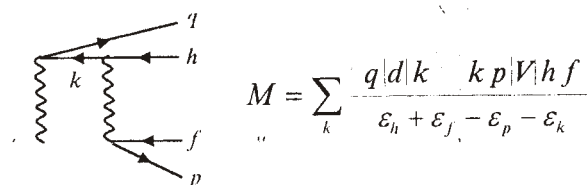


Рис. 4

$$M = \sum_k \frac{q|d|k \quad k p|V|h f}{\varepsilon_h + \varepsilon_f - \varepsilon_p - \varepsilon_k}$$

Здесь $\langle q|d|k \rangle$, $\langle k p|V|h f \rangle$ – соответственно дипольный и кулоновский матричные элементы. Одноэлектронные орбитали $|a \rangle$ в приближении центрального поля разделяются на функции $P_{\varepsilon_a}(r)$, $Y_{l m_a}(\theta, \varphi)$, $\chi_{\mu_a}(\sigma)$, зависящие только от радиальных, угловых и спиновых переменных соответственно, и характеризуются квантовыми числами энергии ε_a , орбитального момента l и его проекции m_a и проекции спина

$$\mu_a |a \rangle = |\varepsilon_a m_a \mu_a \rangle = \frac{1}{r} P_{\varepsilon_a}(r) Y_{l m_a}(\theta, \varphi) \chi_{\mu_a}(\sigma)$$

Примем следующий порядок сложения моментов в рамках приближения LS-связи. Пусть моменты дырок f и h образуют терм ионного остова $L_h S_h$, моменты электронов p и q связываются в терм электронной пары $L_p S_p$, затем термы иона и электронов связываются в полный терм LS системы "двукратный ион + два электрона". Обозначим через sa спин частицы a . Ниже приводится листинг входного файла и результат его обработки:

```
create_new_diagram DPI1;
load_data;
sums_over_projections      p  q  f  h  l  Lp  Lh;
clebsh_gordan_coefs      < p + q + | Lp + > < f + h + | Lh + > < Lp + Lh + | L + >;
sums_over_projections      cp cq cf ch sl  Sp  Sh;
clebsh_gordan_coefs      < cp + cq + | Sp + > < cf + ch + | Sh + > < Sp + Sh + | S + >;
dipole_me      < q cq + | D | l sl + >;
coulomb_me      < p cp + l sl + | V k | f cf - h ch - >;
get_numerical_data (3,0)p (1,0)q (0,0)f (1,0)h (2,0)l (2,0)Lp (1,0)Lh (1,1)L (3,0)k;
get_numerical_data (0.5,0)cp (0.5,0)cq (0.5,0)cf (0.5,0)ch (0.5,0)sl (0,0)Sp (0,0)Sh (0,0)S;
end_data;
evaluate;
reduce;
evaluate;
output;
```

Заметим, что значения проекций, по которым проводится суммирование, устанавливать не требуется. Таким образом, загруженные в программу числовые данные соответствуют двойной фотоионизации атома

A , в процессе которой из оболочек $n_1 s$ и $n_2 p$ в непрерывный спектр выбиваются два электрона $\varepsilon_1 f$ и $\varepsilon_2 p$: $A + \hbar\omega \rightarrow n_1 s n_2 p [^1 P] + (\varepsilon_1 f, \varepsilon_2 p) [^1 D]$.

Main output file of QAMT program (E:\Atom\Public\QAMT\DATA\DPI1.txt)

```
Input file: DPI1.amt
Control file: E:\Atom\Public\QAMT\DATA\DPI1.ctr
create_new_diagram DPI1; - diagram DPI1 is created.
load_data; - data loaded into diagram DPI1:

Sum( mp mq mf mh ml mlp mLh mcf mch mSp mSh mk)
Phase( +2p -mp +q -mq +3f -mf +h -mh +l -ml +Lp +mLp -Lh +mLh +mL +3cf -mcf
      -ch -mch +Sp +mSp -Sh +mSh +mS +2k -mk)
[p|l [q|] [f]l [h]l [l]2 [Lp]l [Lh]l [L]l [Sp]l [Sh]l [S]l
d(cq,ch) d(cp,cf) d(sl,ch) d(ch,cq) d(mch,-mcq)
d(cf,cp) d(mcf,-mcp) d(ch,sl) d(mch,-msl)
( q hv_1 l) ( p k f) ( l k h)
| p q Lp| | f h Lh| | Lp Lh L| | cf ch Sp| | cf ch Sh| | Sp Sh S|
| + + -| | + + -| | + + -| | - - -| | + + -| | + + -|
| q hv_1 l| | p k f| | l k h|
| - + +| | - + -| | - - -|

evaluate; - numerical value of DPI1 is -0.156492
reduce; - diagram DPI1 is reduced.
evaluate; - numerical value of DPI1 is -0.156492
output; - diagram DPI1:
```

Phase(+p +q +h -Lp -Lh +hv_1)


```
[p]1 [q]1 [f]1 [h]1 [l]2 [Lp]1 [Lh]1 [Sh]1 [hv_1]-1
d(cq,ch) d(cp,cf) d(sl,ch) d(ch,cq) d(mch,-mcq) d(cf,cp) d(mcf,-mcp) d(ch,sl)
d(mch,-msl) d(Sh,Sp) d(mSh,-mSp) d(S,0) d(mS,m0) d(hv_1,L) d(mhv_1,mL)
( q hv_1 l ) ( p k f ) ( l k h ) { p q Lp } { f h Lh } { Lp Lh hv_1 } { cf ch Sh }
{ Sh Sh 0 } { p Lh l }
| k h l | | l q hv_1 |
| Lh p f | | Lp Lh p |
```

ErrorCode = 0

Здесь $h\nu_1=1$ – угловой момент фотона. Вот аналитическая запись полученного выражения

$$(-1)^{p+q+h-L_p-L_h+1} \prod_{pqhL_pL_hS_h} / \prod_l \delta(S_h, S_p) \delta(S,0) \delta(M_S,0) \delta(l, L) \delta(\alpha, M_l) \\ \{p q L_p\} \{f h L_h\} \{L_p L_h L\} \{1/2 1/2 S_h\} \\ \begin{pmatrix} q & l & l \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} p & k & f \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} l & k & h \\ 0 & 0 & 0 \end{pmatrix} \begin{Bmatrix} k & h & l \\ L_h & p & f \end{Bmatrix} \begin{Bmatrix} l & q & 1 \\ L_p & L_h & p \end{Bmatrix}$$

Включение в рассмотренный пример команды `integer &` них, файл `amt_intr.inc`, объявляет общий блок `fortran`; вызывает появление в текущем каталоге переменных, которые требуются для вычисления. Вот двух файлов с кодом на языке Фортран 90. Первый из его листинг:

```
integer &
p, mp, q, mq, f, mf, h, mh, l, ml, Lp, mLp, Lh, mLh, L, mL, &
cp, mcp, cq, mcq, cf, mcf, ch, mch, sl, msl, Sp, mSp, Sh, mSh, S, mS, &
hv_1, mhv_1, k, mk, dummy
common /amt_interface/ &
p, mp, q, mq, f, mf, h, mh, l, ml, Lp, mLp, Lh, mLh, L, mL, &
cp, mcp, cq, mcq, cf, mcf, ch, mch, sl, msl, Sp, mSp, Sh, mSh, S, mS, &
hv_1, mhv_1, k, mk, dummy
```

Отметим, что все моменты объявлены целыми числами. Для передачи спинов удобно использовать удвоение всех моментов. В файле `fort_amt.f90` содержится код подпрограммы `DP11()` для вычисления углового множителя рассмотренной диаграммы:

```
! В переменной Value возвращается числовое значение диаграммы
subroutine DP11(Value)
implicit double precision (a-h,o-z)
include 'amt_intr.inc'
x(m) = dsqrt(m + 1.d0)
sum = 0.d0
Factor = 1.d0 &
* x(p) * x(q) * x(f) * x(h) * x(l)**2 * x(Lp) * x(Lh) * x(Sh) &
/ x(hv_1)
A0 = 1.d0 &
* w3(q,hv_1,l,0,0,0) &
* w3(p,k,f,0,0,0) &
* w3(l,k,h,0,0,0) &
* Triang(p,q,Lp) &
* Triang(f,h,Lh) &
* Triang(Lp,Lh,hv_1) &
* Triang(cf,ch,Sh) &
* Triang(Sh,Sh,0) &
* Triang(p,Lh,l)
Phase = (-1.d0)**(( &
+p +q +h -Lp -Lh +hv_1+0 &
)/2)
A3 = 1.d0
sum = sum + Phase * A3
A6 = 1.d0 &
* w6( k, h, l, Lh, p, f) &
* w6( l, q, hv_1, Lp, Lh, p)
A9 = 1.d0
Value = Factor * A0 * A6 * A9 * sum
return
end
```

Использованные здесь подпрограммы `w3()`, `w6()`, `Triang()` предназначены для вычисления значений $3j$ -, $6j$ -символов и проверки условия треугольника соответственно, они могут быть легко реализованы при помощи многочисленных формул, приведенных в [3]. Как видно из приведенного примера, в качестве

имен переменных программа QAMT берет имена моментов, поэтому для корректной работы сгенерированного кода имена моментов должны подчиняться общим правилам именования, принятым в языке Фортран. Вот пример кода, позволяющего использовать подпрограмму `DPI1()`:

```
double precision Value
include 'amt_intr.inc'
! Здесь должен располагаться блок определения числовых значений моментов.
call DPI1(Value)
! Теперь переменная Value содержит значение диаграммы.
end
```

Технические подробности и ограничения

Программа разработана на языке C++ с использованием компилятора Microsoft Visual C++ 5.0 и предназначена для работы в 32-разрядной среде операционных систем Windows 95/NT. Некоторая искусственность системы команд есть отражение внутренней структуры программы, первоначально разрабатывавшейся для интерфейса Windows.

Несмотря на то, что с точки зрения алгоритма программа не имеет ограничений по количеству сумм при вычислении числовых значений диаграмм, время расчета значительно увеличивается с ростом числа суммирований.

Заключение

С ростом мощности современных компьютеров не слишком большие суммы произведений ККГ и $3j$ -символов могут быть вычислены непосредственно (как это делает программа QAMT, если применить команду `evaluate;` до команды `reduce;`). Для более громоздких выражений необходимо их упрощение. QAMT позволяет свести выражения, содержащие многочисленные суммы по проекциям моментов, к более компактным формулам, состоящим из $6j$ - и $9j$ -символов и, возможно, некоторого количества сумм

по моментам. Последние выражения, как правило, вычисляются значительно быстрее. Автоматическое преобразование выражений, угловых множителей матричных элементов снимает с пользователя массу рутинной работы. Генерация кода также оказывается весьма полезной, так как позволяет избежать ошибок программирования.

Разработка данной программы поддержана грантом № 3355 фонда "Российские университеты".

Литература

1. Собельман И.И. Введение в теорию атомных спектров. – М.: Наука, 1977. – С.61–68.
2. Юцис А.П., Левинсон И.Б., Ванagas В.В. Математический аппарат теории момента количества движения. – Вильнюс: Минтис, 1965.
3. Варшалович Д.А., Москалев А.Н., Херсонский В.К. Квантовая теория углового момента. – Л.: Наука, 1975.
4. Эль-Баз Э., Кастель Б. Графические методы алгебры спинов. – М.: Мир, 1974.
5. Carter S. L., Kelly H. P. 1976. J. Phys. B 9, 1887.