Asian Journal of Chemistry Vol. 20, No. 2 (2008), 838-844

Calculation of the Symmetry of C_{24} Fullerene

AHMAD GHOLAMI* and ALI REZA ASHRAFI
Department of Mathematics, Faculty of Science, University of Kashan, Kashan, Iran
E-mail: gholami@kashanu.ac.ir

In this paper we present some MATLAB and GAP programs and use them to find the automorphism group of the Euclidean graph of the C_{24} fullerene. It is well known to associate an Euclidean graph to a molecule. Balasubramanian computed the Euclidean graphs and their automorphism groups for benzene, eclipsed and staggered forms of ethane and eclipsed and staggered forms of ferrocene. It is also proved that algorithm, which is useful for finding symmetry of molecules. Using this algorithm, a new simple method is described, by means of which it is possible to calculate the automorphism group of weighted graphs. We apply this method to compute the symmetry of fullerene C_{24}.

Key Words: Fullerene, Euclidean graph, Symmetry.

INTRODUCTION

Let G = (V,E) be a simple graph. G is called a weighted graph if each edge e is assigned a non-negative number w(e), called the weight of e. An unweighted graph G can be regarded as a weighted graph in which for all edges $e \in E(G)$, w(e) = 1. Euclidean graph of a molecule is a complete weighted graph in which the edges are weighted by the Euclidean distances of vertices.

An automorphism of a weighted graph G is a permutation g of the vertex set of G with the property that, (i) for any vertices u and v, g(u) and g(v) are adjacent if u is adjacent to v (ii) for every edge e, w(g(e)) = w(e). The set of all automorphisms of a weighted graph G, with the operation of composition of permutations, is a permutation group on V(G), denoted Aut(G).

By symmetry we mean the automorphism group symmetry of a graph. The symmetry of a graph, also called a topological symmetry, accounts only for the bond relations between atoms and does not fully determine molecular geometry. The symmetry of a graph does not need to be the same as (*i.e.* isomorphic to) the molecular point group symmetry. However, it does represent the maximal symmetry which the geometrical realization of a given topological structure may posses.

It was shown by Randic[1,2] that a graph can be depicted in different ways such that its point group symmetry or three dimensional perception may differ, but the underlying connectivity symmetry is still the same as characterized by the automorphism group of the graph. However, the molecular symmetry depends on the coordinates of the various nuclei which relate directly to their three dimensional geometry. Although the symmetry as perceived in graph theory by the automorphism group of the graph and the molecular group are quite different, it was shown by Subramanian[3] that the two symmetries are connected.

In this paper, we consider only weighted graphs. The motivation for this study is already outlined by various workers[3-16]. Our notation is standard and taken mainly from the previous reports[17-19].

COMPUTATIONAL METHOD

In this section we first describe some notation, which will be kept throughout. Let G be a group and N be a subgroup of G. N is called a normal subgroup of G, if for any $g \in G$ and $x \in N$, $g^{-1}xg \in N$. If H is another normal subgroup of G such that $H \cap N = \{e\}$ and $G = HN = \{xy \mid x \in H, y \in N\}$, then we say that G is a direct product of H and N denoted by $H \times N$. A group with no proper non-trivial normal subgroup is called simple group. Suppose X is a set. The set of all permutations on X, denoted by S_X, is a group which is called the symmetric group on X. In the case that, $X = \{1, 2, \ldots, n\}$, we denote S_X by S_n or Sym(n).

In recent years, a rapid spread of interest in the understanding, design and even implementation of group theoretical algorithms. These are gradually becoming accepted both as standard tools for a working group theoretician, like certain methods of proof and as worthwhile objects of study, like connections between notions expressed in theorems.

Our computations of the symmetry properties of molecules were carried out with the use of GAP[20]. GAP stands for Groups, Algorithms and Programming. The name was chosen to reflect the aim of the system, which is a group theoretical software for solving computational problems in computational group theory. This software was constructed by GAP's team in Aachen. GAP is a free and extendable software package. The term extendable means that you can write your own programs in the GAP language and use them in just the same way as the programs which form part of the system (the "library"). More information on the motivation and development of GAP to date can be found on GAP web page on http://www.gap-system.org. GAP contains a large library of functions, which are important for the calculations in this paper.

GAP contains several functions for working with finite groups. For the sake of completeness, we describe some of these functions which are useful throughout. Let a_1, a_2, …, a_r are permutations of {1,2,…,r}. The command "Group(a_1,a_2,…,a_r)" computes the group generated by permutations a_1, a_2, …, a_r. For two groups A and B, the commands "Size(A)", "Generators of Group(A)" and "Intersection(A,B)" compute the cardinality of the set A, a generator set for A and intersection of A and B, respectively. Finally the command "IsSimple(A)" determines whether or not A has a non-trivial proper normal subgroup. In this paper, we use freely these functions and the reader is encouraged to consult the manual of GAP[20] and work of Ashrafi *et al.*[14-16,21].

Consider the equation $(P_\sigma)^t AP_\sigma = A$, where A is the adjacency matrix of the weighted graph G. Suppose Aut(G) = {σ_1, σ_2,…, σ_m}. The matrix S_G = [s_{ij}], where $s_{ij} = \sigma_i(j)$ is called a solution matrix for G. Clearly, for computing the automorphism group of G, it is enough to calculate a solution matrix for G. The second author[16] proved a result that is useful for computing symmetry of molecules. Using this result, Lemma 1 and its Corollary, we present a MATLAB program[22] for computing a solution matrix for the automorphism group of Euclidean graphs.

A MATLAB Program for computing the symmetries of molecules

```
n=length(a);
    for i=1:n-1
        for j=i+1:n
            b(i,j)=norm(a(i,:)-a(j,:));
        end
    end
b(n,n)=0;
b=b+b';

function y=halat(s,a)
    t=1:length(a);
    m=length(s);
    t(s)=[];
    j=0;
        for i=t
            if min(min(a(1:m+1,1:m+1)==a([s,i],[s,i])))==1
                j=j+1;
                y(j)=i;
            end
        end

function s=hazf(s)
    m=size(s);
```

```
        for i=m(1):-1:1
            if min(s(i,:))==0
                s(i,:)=[];
            end
        end

function s=jaigasht(a)
    m=length(a);
        for i=1:m
            s(i,1)=i;
        end
        for j=2:m
            n=size(s);
            k=0;
                for i=1:n(1)
                    y=[halat(s(i,:),a)];
                        for r=1:length(y)
                            b(r+k,1:n(2)+1)=[s(i,:),y(r)];
                        end
                    k=k+length(y);
                end
            s=b;
            s=hazf(s);
        end
    b=0;
    n=size(s);
        for i=1:n(1)
            for j=1:n(2)
                b(i,s(i,j))=j;
            end
        end
    s=b;
```

Our program needs the Cartesian coordinates of the atoms to determine the Euclidean distances in the molecule under consideration. If we calculate these distances by HyperChem, Gaussian 98 or another software, then for computing the symmetry of molecule under consideration, it is enough to delete the first eight lines of the program and load the distance matrix of the molecule.

RESULTS AND DISCUSSION

In this section, we apply our program to compute the automorphism group of Euclidean graph of C_{24} molecule. The Cartesian coordinates of

C_{24} were computed by HyperChem and Gaussian 98. It is useful to mention that in present program the accuracy is very important. Our calculations on the symmetry of some fullerenes show that, if we change the accuracy then the automorphism group will be changed.

We now calculate the symmetry of fullerene C_{24}. Fullerenes are molecules in the form of polyhedral closed cages made up entirely of n three-coordinate carbon atoms and having 12 pentagonal and (n/2-10) hexagonal faces, where n is even and greater or equal 20, with the exception of n=22. Hence, the fullerene, C_{24}, (n = 24) has but 12 pentagons and 2 hexagons. Let G be the automorphism group of Euclidean graph of C_{24} fullerene with I_h symmetry point group. Consider this molecule (Fig. 1), to illustrate the Euclidean graphs and their automorphism group. As we mentioned before, we don't have to work with exact Euclidean distances since a mapping of weights into a set of integers suffices as long as different weights are identified with different integers. To illustrate let us use a Euclidean edge weighting for fullerenes C_{24} obtained from Table-1 and our program. Suppose A is the 24×24 matrix defined by Euclidean distances.

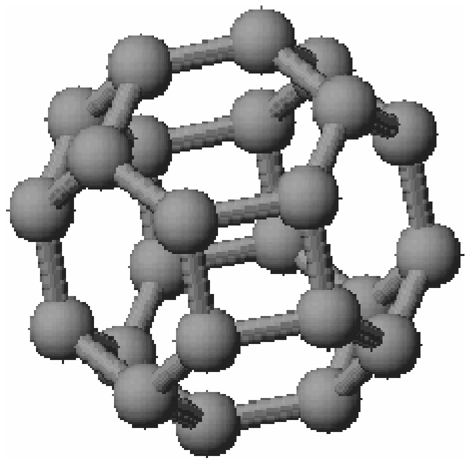

Fig. 1. The fulerene C_{24}

Not all 24! permutations of the vertices C_{24} belong to the automorphism group of its weighted graph since the weights of all the edges are not the same. For example, the permutation (1,2,3,4,5,6,7) does not belong to the automorphism group since the resulting graph does not preserve connectivity. Let X denote the set of all solutions of matrix equation $P^tAP = A$. Set $Y = \{\propto \in S_{24} \mid P_{\propto} \in X\}$. Then Y is the automorphism group of Euclidean graph of C_{24}. We now apply our MATLAB program to find a solution matrix for this group. Using the solution matrix of C_{24} and a simple

GAP program, we can find the structure of the automorphism group G of Euclidean graph of C_{24}. We mention that this program is very fast and its running time is less than 0.01 s. Our GAP program is as follows:

We computed below the distance matrix D for C_{24}, as follows:

0	2	4	4	3	4	4	5	4	6	5	6	7	4	3	4	7	6	10	6	10	9	10	11
2	0	4	4	5	6	7	3	4	4	3	4	4	6	5	4	10	6	7	6	10	11	10	9
4	4	0	8	4	3	4	4	4	3	6	5	4	5	6	7	6	7	6	10	11	10	9	10
4	4	8	0	6	5	4	6	7	5	4	3	4	3	4	4	6	10	6	7	9	10	11	10
3	5	4	6	0	4	4	8	4	7	11	10	6	7	9	10	4	4	6	10	6	3	4	5
4	6	3	5	4	0	4	7	4	9	10	11	10	8	7	6	4	4	10	6	5	4	3	6
4	7	4	4	4	4	0	6	3	10	6	10	9	4	4	3	8	5	11	5	6	7	6	10
5	3	4	6	8	7	6	0	4	4	9	7	4	10	11	10	6	4	4	10	6	5	4	3
4	4	4	7	4	4	3	4	0	4	10	6	3	6	10	9	5	8	5	11	10	6	7	6
6	4	3	5	7	9	10	4	4	0	7	8	4	11	10	6	10	4	4	6	5	6	3	4
5	3	6	4	11	10	6	9	10	7	0	4	4	7	8	4	6	10	4	4	4	5	56	3
6	4	5	3	10	11	10	7	6	8	4	0	4	9	7	4	10	6	4	4	3	6	5	4
7	4	4	4	6	10	9	4	3	4	4	4	0	10	6	3	11	5	8	5	6	10	6	7
4	6	5	3	7	8	4	10	6	11	7	9	10	0	4	4	4	6	10	4	3	4	5	6
3	5	6	4	9	7	4	11	10	10	8	7	6	4	0	4	4	10	6	4	4	3	6	5
4	4	7	4	10	6	3	10	9	6	4	4	3	4	4	0	5	11	5	8	7	6	10	6
7	10	6	6	4	4	8	6	5	10	6	10	11	4	4	5	0	3	9	3	4	4	4	7
6	6	7	10	4	4	5	4	8	4	10	6	5	6	10	11	3	0	3	9	7	4	4	4
10	7	6	6	6	10	11	4	5	4	4	4	8	10	6	5	9	3	0	3	4	7	4	4
6	6	10	7	10	6	5	10	11	6	4	4	5	4	4	8	3	9	3	0	4	4	7	4
10	10	11	9	6	5	6	6	10	5	4	3	6	3	4	7	4	7	4	4	0	4	8	4
9	11	10	10	3	4	7	5	6	6	5	6	10	4	3	6	4	4	7	4	4	0	4	8
10	10	9	11	4	3	6	4	7	3	6	5	6	5	6	10	4	4	4	7	8	4	0	4
11	9	10	10	5	6	10	3	6	4	3	4	7	6	5	6	7	4	4	4	4	8	4	0

G={(1),(3 ,4)(5,15)(6,14)(8,11)(9,16)(10,12)(18,20)(21,23)
(1,2) (5,8)(6.10)(7,13)(8,5)(11,15)(12,14)(17,19)(22,24),
(1,2)(3,4)(5,11)(6,12)(7,13)(8,15)(9,16)(10,14)(17,19)(18,20)(21,23)
(22,24)}

Conclusion

Suppose T is a complete weighted graph and Supp(T) = |{w(e)|e is an edge of T}|. If Supp(T) is large enough, for example greater than |V(T)|, then our algorithm and also our MATLAB program is very fast for computing the symmetry of the graph T. In particular, our program is suitable for computing symmetry of fullerenes. We applied our programs for computing symmetries of all molecules in fullerene gallery presented by Mitsuho Yoshida (for details http://www.cochem2.tutkie.tut.ac.jp/Fuller/higher/higherE.html) with running time less than 0.01 s for GAP program and less than 1 s for MATLAB program with some parallel Pentium IV computers. The maximum running time is obtained in the case of fullerene with I_h symmetry group.

We also mentioned that, our calculations with GAP and calculations done by Balasubramanian[3-9], Hao-Xu[10], Ivanov[11] and Ivanov-Schüürmann[12], show that the automorphism group of the Euclidean graph of every molecule is trivial or has an even number of elements.

REFERENCES

1. M. Randic, *Chem. Phys. Lett.*, **42**, 283 (1976).
2. M. Randic, *J. Chem. Phys.*, **60**, 3920 (1974).
3. K. Balasubramanian, *Chem. Phys. Lett.*, **232**, 415 (1995).
4. K. Balasubramanian, *J. Chem. Phys.*, **72**, 665 (1980).
5. K. Balasubramanian, *Intern. J. Quantum Chem.*, **21**, 411 (1982).
6. K. Balasubramanian, *Chem. Rev.*, **85**, 599 (1985).
7. K. Balasubramanian, *J. Phys. Chem.*, **108**, 5527 (2004).
8. K. Balasubramanian, *Chem. Phys. Lett.*, **391**, 64 (2004).
9. K. Balasubramanian, *Chem. Phys. Lett.*, **391**, 69 (2004).
10. J.-F. Hao and L. Xu, *Computers and Chemistry*, **26**, 119 (2002).
11. J. Ivanov, *J. Chem. Inf. Comput. Sci.*, **44**, 596 (2004).
12. J. Ivanov and G. Schüürmann, *J. Chem. Inf. Comput. Sci.*, **39**, 728 (1999).
13. H.C. Longuet-Higgins, *Mol. Phys.*, **6**, 445 (1963).
14. A.R. Ashrafi, *MATCH Commun. Math. Comput. Chem.*, **53**, 161 (2005).
15. G.A. Moghani, A.R. Ashrafi and M. Hamadanian, *J. Zhejiang Univ. Sci.*, **6B**, 222 (2005).
16. A.R. Ashrafi, *Chem. Phys. Lett.*, **406**, 75 (2005).
17. G.S. Ezra, Symmetry Properties of Molecules, Lecture Notes in Chemistry 28, Springer, Berlin-Hidelberg (1982).
18. W.C. Herndon, in ed.: R.B. King, Chemical Applications of Graph Theory and Topology, Physical and Theoretical Chemistry, Elsevier, Amsterdam, Vol. 28, pp. 231-242 (1983).
19. N. Trinajstic, Chemical Graph Theory, CRC Press, Boca Raton, FL (1992).
20. M. Schönert, H.U. Besche, Th. Breuer, F. Celler, B. Eick, V. Felsch, A. Hulpke, J. Mnich, W. Nickel, G. Pfeiffer, U. Polis, H. Theißen and A. Niemeyer, GAP, Groups, Algorithms and Programming, Lehrstuhl De für Mathematik, RWTH, Aachen (1995).
21. A.R. Ashrafi and M. Hamadanian, *Croat. Chim. Acta*, **76**, 299 (2003).
22. D.J. Higham and N.J. Higham, MATLAB Guide, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (2000).

(*Received*: 21 September 2006; *Accepted*: 24 September 2007) AJC-5904