Computer Generation of Spectra of Graphs: Applications to C_{60} Clusters and Other Systems

K. Balasubramanian* and Xiaoyu Liu

Department of Chemistry, Arizona State University, Tempe, Arizona, 85287-1604

Received 7 July 1987; accepted 5 October 1987

A computer code based on the Givens–Householder matrix diagonalization method is used to calculate the spectra of graphs containing a large number of vertices. The code is most general in that it can handle graphs containing 200 or more vertices. Further, the code can be used to generate the spectra of weighted graphs. The program requires as input only the neighborhood table of the graph. The spectra of many graphs are generated for the first time in less than a few minutes of computer time. Applications to a number of chemical systems including two forms (foot and hand) of the recently synthesized C_{60} cluster and the effect of bond alternation on these systems are discussed. In addition, the spectra of square and honeycomb lattices and the characteristic polynomials of the foot and hand forms of the C_{60} cluster are obtained.

INTRODUCTION

The construction of characteristic polynomials of graphs and their spectra[1–37] has been the subject of numerous investigations in recent years. Besides their importance to pure mathematics, there are many chemical applications of characteristic polynomials and spectra of graphs. They have applications in chemical kinetics,[38] quantum chemistry, dynamics of oscillatory reactions and in determining the stabilities of reaction networks,[36] lattice statistics,[39] estimating the stabilities of conjugated systems,[28] formulation of TEMO theorem,[39] enumeration of walks and self-returning walks,[3] and electronic structure of organic polymers and periodic structures.[5,12]

In an earlier investigation, Balasubramanian[1] developed a computer code based on a very elegant recursive matrix product method[3] to generate the characteristic polynomials for a variety of graphs containing a large number of vertices. This method was recently extended to edge-weighted graphs, vertex-weighted graphs, directed graphs, and signed graphs.[2] This method was also shown to be applicable to characteristic polynomials of organic polymers and periodic structures.[5] The characteristic polynomials of many graphs and lattices containing a large number of vertices could be obtained for the first time using this computer code.[1]

The present investigation is focused on applying a computer code to calculate the eigenvalues of graphs, also known as the spectra of graphs. The code is especially for graphs containing a large number of vertices. The computer code in Fortran 77 used here is based on the Givens–Householder method and requires only the neighborhood table of the graph. The code generates the spectra of many graphs in seconds and of larger graphs in a few minutes. The code can also handle edge-weighted graphs and vertex-weighted graphs. We apply this code to a number of graphs which include square lattices, honeycomb lattice graphs, pericondensed benzenoid hydrocarbons, and the two forms of the recently synthesized C_{60} (buckminsterfullerene) namely the football form and the hand form. The effect of bond alternation on the stabilities of these two forms is also discussed.

PRELIMINARIES AND ALGORITHMS

The adjacency matrix A of a graph is defined as

$$A_{ij} = \begin{cases} 1 & \text{if the vertices } i \text{ and } j \text{ are connected} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

*Alfred P. Sloan Fellow; Camille and Henry Dreyfus Teacher-Scholar.

Journal of Computational Chemistry, Vol. 9, No. 4, 406–415 (1988)

CCC 0192-8651/88/040406-10$04.00

The adjacency matrix of a graph is real and symmetric by definition. Thus this can be brought to a diagonal form by successive orthogonal transformations. The diagonal elements of the transformed diagonal matrix are the eigenvalues or commonly known as the spectrum of the graph.

There are two common procedures employed to find all the eigenvalues of a real-symmetric matrix. The first, known as the Jacobi method employs a sequence of orthogonal transformations shown below:

$$A_k = P_k^T A_{k-1} p_k, \qquad k = 1, 2, \ldots (A_0 = A) \tag{2}$$

The above transformation is repeated until A_k is brought to a diagonal form, (where P_k's are orthogonal matrices of the same order as A with all but two rows containing unit diagonal elements; the two rows with nonunit diagonal elements contain the orthogonal rotation elements). The major problem associated with the Jacobi's method is that some former nullified off-diagonal elements often become non-zero again during later transformation steps. This effect alone lengthens the total diagonalization process greatly, and the amount of calculation becomes extremely large for graphs containing a large number of vertices.

Using the second linear transformation, Householder[48–50] showed that by a finite sequence

$$A_k = R_k^T A_{k-1} R_k, \qquad k = 1, 2, \ldots, m \quad (A_0 = A) \tag{3}$$

of transformations, where R_k's are orthogonal rotation-reflection matrices of the form

$$R = I - 2\mathbf{U}\mathbf{U}^T, \tag{4}$$

any real-symmetric matrix A could be transformed to the tridiagonal form C,

$$C = A_m = \begin{vmatrix} c_1 & b_1 & & & & & \\ b_1 & \cdot & \cdot & & & & \\ & \cdot & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \cdot\ b_{n-1} & \\ & & & \cdot & \cdot & c_n & \\ & & & & b_{n-1} & & \end{vmatrix} \tag{5}$$

where $\mathbf{U}$ is a column vector such that $\|\mathbf{U}\|^2 = \mathbf{U}^T\mathbf{U} = 1$. A total of $n - 2$ transformations are needed to tridiagonalize a given matrix of order nxn. The eigenvalues of a tridiagonal matrix can be easily computed by many methods.

In the present investigation we adopt the Givens–Householder method[51] to calculate the spectra of graphs. The Givens–Householder method provides a very stable, efficient, and precise way to calculate the spectra of graphs compared with other numerical techniques since we need to compute all eigenvalues as opposed to selected eigenvalues. It should also be noted that the Givens–Householder procedure is commonly used in quantum chemical SCF-type calculations.[52] One could also obtain the eigenvectors using the Givens–Householder method. This method is based on the transformation described earlier to convert a real-symmetric matrix into a tridiagonal form. The method is described in ref. 50 in some detail. Thus we do not repeat this description.

BRIEF DESCRIPTION OF THE COMPUTER CODE

A Fortran 77 program for calculating the spectra of graphs containing large number of vertices has been developed. Since this program is intended for graphs containing up to 200 vertices, the adjacency matrix of the graph under consideration is generated by this program using the neighborhood table provided as input. The Givens–Householder subroutine to tridiagonalize a real-symmetric matrix written by Prosser[52] was interfaced with other routines developed here. This program needs only the set of vertices that are neighbors of a given vertex with labels less than the label of the given vertex. The program is designed efficiently so that only the lower triangle of the adjacency matrix need to be stored in a packed form.

The final program comprises the following parts. A main program called MAINSEPC reads all the input information, generates the adjacency matrix, allocates the exact amounts of memory spaces needed dynamically, and calls a subroutine named SUBSPEC which is the driver to calculate the spectrum of the input graph. The input for this code is similar to the input for the code developed earlier by Balasubramanian[1] for the characteristic polynomials of graphs. A listing of the code can be obtained from the

authors. The only difference between the input described in ref. 1 and the input for the present code is that the List-Directed Read statement is used in the program to accept input data; thus the field widths can be omitted. Therefore, if there are multiple data in an input line, only commas are needed to separate them. It really gives the user great convenience to input data.

The current version of the program can handle graphs containing up to 200 vertices. However, this is a minor restriction imposed by the current array dimensions which could be changed for graphs containing more vertices. We estimate that about 160K byte computer memory has to be allocated to store the packed adjacency matrix for graphs up to 200 vertices. The upper limit of 200 vertices is really large enough to handle many chemical graphs. Actually, there are quite a few computer systems, especially minisystems, and PCs where the operating systems allow only 32K or 64K byte dynamic memory space to each user. In this event, the sizes of the one-dimensional arrays in the main program should be changed to meet the memory requirements of these computer systems.

The program used here is very efficient in that it takes only fractions of a second for graphs containing 20 or fewer vertices and about 10 minutes for graphs containing 200 vertices on a VAX 11/730 system. In the next section, many applications, examples, and the CPU times are given to demonstrate this point. The spectra of all the graphs considered here were obtained using a VAX 11/730 system, which is about one third slower than the standard VAX 11/780 system or a microvax system.

RESULTS AND DISCUSSION

Two types of graphs of chemical interest are considered here, namely, nonweighted and weighted graphs. The adjacency matrix of a nonweighted or ordinary graph was defined earlier [expression (1)]. Clearly, the trace of the adjacency matrix of a nonweighted graph is zero:

$$\text{Trace}(A) = \sum_{i=1}^{n} a_{ii} = 0 \tag{6}$$

Since the trace is unchanged by an orthogonal linear transformation, the trace of the tri-diagonalized matrix for a nonweighted graph must also be zero, that is, the sum of the eigenvalues of a nonweighted graph must be zero:

$$\sum_{i=1}^{n} \lambda_i = 0. \tag{7}$$

According to Frobenius' theorem,[46,53] the limits of the graph spectrum are determined by the row module of the adjacency matrix:

$$\max_i |\lambda_i| \leq \|A\|_\infty = \max_i \sum_{j=1}^{n} |a_{ij}| \tag{8}$$

The direct application of this theorem to nonweighted graphs indicates that the limits of the spectrum of a nonweighted graph are determined by the maximal degree of a vertex, D_{max}, in the graph

$$-D_{max} \leq \lambda_i \leq D_{max} \tag{9}$$

Relations (7) and (9) can be used to check if the calculated eigenvalues obey these criteria.

The program to calculate the spectra was employed to find the spectra of many graphs for which closed analytical expressions exist, such as K_n (complete graphs which contain n vertices), L_n (a linear chain containing n vertices), C_n (cyclic graphs containing n vertices), and prism graphs as defined in ref. 22. The set of complete graphs, K_ns (n = 4 to 10) took 4.85 seconds of CPU time on a VAX 11/730 system. The C_n graphs (n = 3–10) took 6.22 s of CPU time on the same system. A prism graph (see ref. 22) which contains 50 vertices took 21.9 s.

The second type of graphs considered are weighted graphs. The adjacency matrix of an edge-weighted graph, A, is defined as

$$A = \{a_{ij}\} = \begin{cases} 0 & \text{if } i = j \\ w_{ij} & \text{if } i \neq j \text{ and } i \text{ and } j \text{ are neighbors} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

where w_{ij} is the weight of the edge connecting the vertices i and j. The normal adjacency matrix of a graph differs from the above matrix in that the weights of all the edges are the same and set to unity in the adjacency matrix. The adjacency matrix of a vertex-weighted graph is defined as

$$A = \{a_{ij}\} = \begin{cases} v_i & \text{if } i = j \\ 1 & \text{if } i \neq j \text{ and } i \text{ and } j \\ & \quad \text{are neighbors} \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

where v_i is the weight of the vertex i. Note that for some vertices, v_i could be zero. There is also the third possibility in which both the vertices and edges could be weighted.

$$A = \{a_{ij}\} = \begin{cases} v_i & \text{if } i = j \\ w_{ij} & \text{if } i \neq j \text{ and } i \text{ and } j \\ & \quad \text{are neighbors} \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

Based on the definitions of the adjacency matrices of the edge-weighted graphs, the vertex-weighted graphs, and edge- and vertex-weighted graphs, the following conclusions could be readily drawn from the Frobenius' Theorem discussed earlier:

$$\left.\begin{aligned} &\sum_{i=1}^{n} \lambda_i = 0 \\ &\lambda_{\max} \leq \max_i \left(\sum_{j=1}^{n} |w_{ij}| \right) \end{aligned}\right\} \text{for edge-only weighted graphs} \tag{13}$$

$$\left.\begin{aligned} &\sum_{i=1}^{n} \lambda_i = \sum_{i=1}^{n} v_i \\ &\lambda_{\max} \leq \max_i \left(|v_i| + \sum_{\substack{j=1 \\ j \neq i}}^{n} |w_{ij}| \right) \end{aligned}\right\} \text{for vertex and edge-weighted graphs} \tag{14}$$

One could use these results to check the spectra calculated by our code.

Table I shows the spectra of some linear weighted chain graphs. The weights are chosen on the assumption that these chains represent conjugated hydrocarbons and the ratio of the weights for single and double bonds (β_d/β_s) is given by the Ohno parameterization ($\beta_d/\beta_s = 1.159$) as described in Ohmine, Karplus, and Schulten[41] for polyenes. The CPU time taken for edge-weighted graphs (weights larger than unity) is generally greater than the time taken for the corresponding nonweighted graphs since it takes more iterations to tridiagonalize a weighted matrix. The CPU time taken for a weighted L_{20} is about 1.49 times that of the ordinary L_{20} graph (linear chain containing 20 vertices).

Table I. Spectra Of Some Edgeweighted-chain Graphs.

n	Eigenvalues[a]	Time(s)[b]
8	0.4787, 1.1134, 1.6663, 2.0317	2.50
10	0.4133, 0.9354, 1.4312, 1.8232, 2.0732	3.34
12	0.3680, 0.8078, 1.2475, 1.6265, 1.9161, 2.0973	4.33
14	0.3350, 0.7123, 1.1035, 1.4580, 1.7536, 1.9753, 2.1125	5.39
16	0.3099, 0.6386, 0.9889, 1.3169, 1.6048, 1.8405, 2.0153, 2.1228	6.58
18	0.2902, 0.5802, 0.8960, 1.1987, 1.4735, 1.7105, 1.9024, 2.0436, 2.1299	7.85
20	0.2744, 0.5328, 0.8195, 1.0992, 1.3591, 1.5911, 1.7890, 1.9480, 2.0643, 2.1352	9.19

[a]Because of the symmetry, only positive eigenvalues are listed in the table above.
[b]CPU time in seconds on a VAX 11/730 system.

Next, we consider a few vertex-weighted graphs. Figure 1 shows a linear chain with loops. The loops can be treated as additional weights to the diagonal elements of the adjacency matrix. Table II shows the calculated spectrum of this graph. One can easily check that the sum of the eigenvalues of this graph is the total number of loops for the graph in Figure 1. Figure 2 shows nitrogen-substituted coronene. This graph is an example of vertex-weighted graph. The weight of the nitrogen atom vertex was chosen as 0.5. With these weights, the calculated spectrum is shown in Table III.

Table II. The spectrum of a chain graph with loops (Fig. 1).

n	Eigenvalues	Time(s)[a]
6	−0.7084, 0.08674, 1.1657, 2.2196, 2.9030, 4.3334	2.03

[a]CPU time in seconds on a VAX 11/730 system.

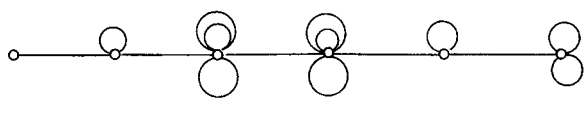

Figure 1. A linear chain which contains loops.

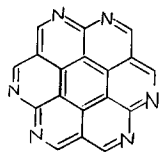

Figure 2. The graph of nitrogen-substituted coronene.

Table III. The spectrum of a nitrogen-substituted coronene graph (Fig. 2).

n	Eigenvalues					Time(s)[a]
24	−2.63807	−2.13691	−2.13691	−1.55640	−1.55640	
	−1.43790	−1.12216	−0.90498	−0.90498	−0.78078	
	−0.39129	−0.39129	0.71623	0.71623	1.11303	
	1.11303	1.26150	1.28078	1.70329	1.82214	
	1.82214	2.33819	2.33819	2.73335		11.73

[a]CPU time in seconds on a VAX 11/730 system.

We consider many interesting and nontrivial graphs, including the square lattice graphs in Figures 3 and 5 and the honeycomb lattice graph in Figure 4. The spectra of square lattices and hexagonal lattices are shown in Tables IV and V, respectively.

The coefficients in the characteristic polynomials of a graph can be generated from the following relations[53]

$$\text{if } f(\lambda) = |\lambda I - A| = \lambda^n + a_{n-1}\lambda^{n-1} + a_{n-2}\lambda^{n-2} + \cdots + a_1\lambda + a_o, \quad (15)$$

then

$$\begin{aligned} a_{n-1} &= (-1)^1(\lambda_1 + \lambda_2 + \cdots + \lambda_n) \\ a_{n-2} &= (-1)^2(\lambda_1\lambda_2 + \lambda_1\lambda_3 + \cdots + \lambda_{n-1}\lambda_n) \\ a_{n-3} &= (-1)^3(\lambda_1\lambda_2\lambda_3 + \lambda_1\lambda_2\lambda_4 + \cdots + \lambda_{n-2}\lambda_{n-1}\lambda_n) \\ &\cdots\cdots \\ a_1 &= (-1)^{n-1}\left(\frac{\lambda_1\lambda_2\cdots\lambda_n}{\lambda_1} + \frac{\lambda_1\lambda_2\cdots\lambda_n}{\lambda_2} + \cdots + \frac{\lambda_1\lambda_2\cdots\lambda_n}{\lambda_n}\right) \\ a_o &= (-1)^n(\lambda_1\lambda_2\cdots\lambda_n) \end{aligned} \quad (16)$$

It turns out, however, that it is an extremely inefficient way to generate the polynomial coefficients from the corresponding eigenvalues. Although it takes only fractions of a second to 20 minutes to generate the polynomial coefficients from the eigenvalues for graphs containing 20 or fewer vertices on a VAX 11/730 computer, it might take a large amount of CPU time for a graph containing 200 vertices. Actually for a graph containing 200 vertices, the total number of combinations of the 200 eigenvalues to generate the polynomial coefficients may be calculated as

$$N = \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{n} = 2^n - 1 = 2^{200} - 1 \sim 1.6 \times 10^{60} \quad (17)$$

It took about 60 h of CPU time on a VAX 11/730 to generate the characteristic polynomial of a graph containing only 24 vertices. Thus the code developed earlier by Balasubramanian[1] for just the characteristic polyno-

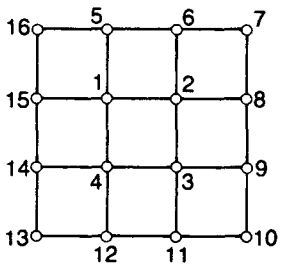

Figure 3. A square lattice graph containing 16 vertices.

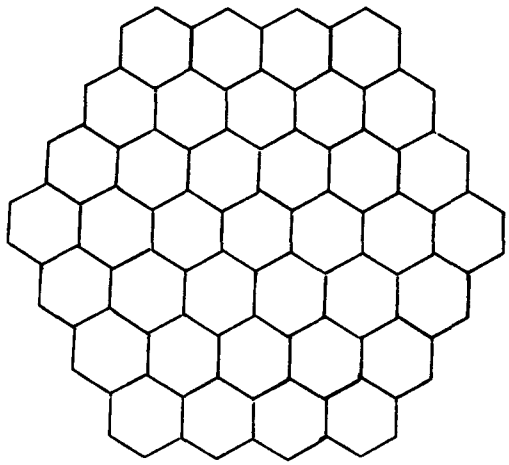

Figure 4. A honeycomb lattice graph containing 96 vertices.

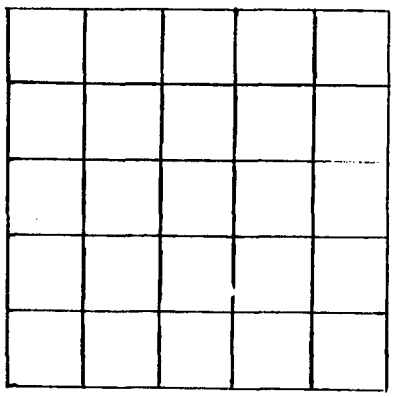

Figure 5. A square lattice graph containing 36 vertices.

Table IV. Spectra of some square lattice graphs.

n	Eigenvalues	Time(s)[b]
16 (Fig. 3)	0, 0, 0, 0, 1, 1, 1.2361, 2.2361, 2.2361, 3.2361	10.98
36 (Fig. 5)	0, 0, 0, 0, 0.2060, 0.4397, 0.6364, 0.7635, 0.8124, 0.9137, 1.3124, 1.3575, 1.6772, 1.7694, 2.2166, 2.2900, 2.4563, 2.9239, 3.0616, 3.5580	24.30
64	0, 0, 0, 0, 0, 0, 0, 0, 0.3473, 0.3473, 0.5321, 0.5321, 0.6527, 0.6527, 0.6946, 0.8794, 0.8794, 1.1848, 1.1848, 1.3473, 1.3473, 1.5321, 1.5321, 1.8794, 1.8794, 2, 2.2267, 2.2267, 2.5321, 2.5321, 2.8794, 2.8794, 3.0642, 3.4115, 3.4115, 3.7588	60.59

[a]Because of the symmetry, the negative eigenvalues are not listed here.
[b]CPU time in seconds on a VAX 11/730 system.

Table V. The spectra of some honeycomb lattice graphs.

n	Eigenvalues[a]	Time(s)[b]
54	0.3420, 0.3420, 0.5786, 0.6818, 0.6818, 1, 1, 1, 1, 1.0464, 1.2375, 1.2375, 1.3526, 1.3915, 1.3915, 1.6057, 1.6057, 1.8053, 1.8053, 1.9406, 2, 2.2155, 2.3095 2.3095, 2.6037, 2.6037, 2.8402	26.49
96 (Fig. 4)	0.2295, 0.2295, 0.3595, 0.4917, 0.4917, 0.6930, 0.7424, 0.7424, 0.8392, 0.8631, 1, 1, 1, 1.0451, 1.0451, 1.0673, 1.0673, 1.0760, 1.2520, 1.2799, 1.2799, 1.2989, 1.2989, 1.3633, 1.3633, 1.5774, 1.5869, 1.5895, 1.5895, 1.6175, 1.6978, 1.8638, 1.8638, 1.9252, 1.9544, 1.9544, 2.1310, 2.1310, 2.2536, 2.2536, 2.3414, 2.3928, 2.5213, 2.5820, 2.5820, 2.7628, 2.7628, 2.9052	101.84

[a]Because of the symmetry, only positive eigenvalues are listed here.
[b]CPU time in seconds on a VAX 11/730 system.

mials is much more efficient for that purpose since for these graphs it takes only a few seconds on the same computer to generate the characteristic polynomials.

APPLICATION TO THE C_{60} CLUSTER

Several studies[54–57] have recently produced an unusually stable cluster of carbon atoms, namely C_{60}. Laser vaporization of graphite in a supersonic nozzle has generated this highly stable cluster named "buckminsterfullerene." Ever since the appearance of this experimental work a number of theoretical investigations have been carried out by many investigators.[58–64] In summary, Klein and coworkers[59,60] and Schmalz *et al.*[61] have employed graph theoretical approaches which use valence bond techniques or conjugated circuit approach to explain the unusual stabilities of C_{60}.

Stone and Wales[58] and Haymet[62] have considered The Hückel approach while Haddon, Brus, and Raghavachari[63] have considered *ab initio* approach and more recently Shibuya and Yoshitani[64] have considered INDO calculations on two forms of the C_{60} cluster (foot and hand form).

The two structures (foot and hand) very recently considered by Shibuya and Yoshitani[64] are especially nice candidates for the computer code described here. Both the foot and

hand forms are shown in Figure 6. The foot form is the famous "soccerball" molecule considered before by others. Table VI shows the spectra of the (unweighted) soccerball and hand forms of the C_{60} cluster. Twice the sum of all positive eigenvalues in Table VI gives the total π-electronic energy of the two forms. This value is 93.1616 and 88.0574, respectively, in terms of the Hückel parameter β. The delocalization energy[26] per atom within the framework of Hückel theory of planar systems is defined by (18),

$$D = (E_\pi - 2n_b)/n\,, \qquad (18)$$

where n_b is the number of double bonds and n is the total number of atoms. Thus the delocalization energy for the foot and the hand forms are 0.5527 and 0.4676, respectively, implying that the foot form is more stable. Note that the delocalization energy of benzene is 0.3333 in the same scale. Thus, both the forms are more stable than benzene.

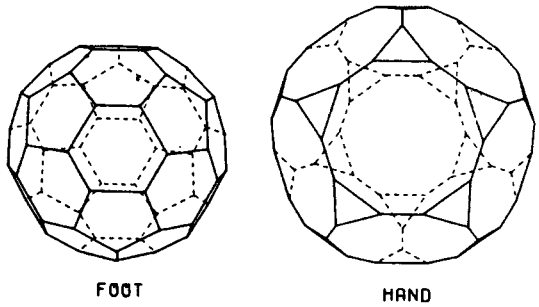

Figure 6. The soccerball (foot) form and the hand forms of the C_{60} cluster.

Another aspect of considerable interest, is the effect of bond alternation on these systems. The recent INDO calculations of Shibuya and Yoshitani[64] suggest bond alternation in these systems. Symmetrically, there are two types of bonds in the foot form, namely those shared by two hexagons and those shared by a hexagon and a pentagon. Elser and Haddon[65] as well as Mallion[66] have

Table VI. The spectra of two forms of the C_{60} cluster: soccerball and hand forms (Fig. 6).

	Eigenvalues	Time(s)[a]
Soccerball	−2.61803, −2.61803, −2.61803 −2.56155, −2.56155, −2.56155, −2.56155, −2.0, −2.0, −2.0, −2.0, −1.61803, −1.61803, −1.61803, −1.61803, −1.61803, −1.43828, −1.43828, −1.43828, −1.30278, −1.30278, −1.30278, −1.30278, −1.30278, −0.38197, −0.38197, −0.38197, −0.13856, −0.13856, −0.13856, 0.61803, 0.61803, 0.61803, 0.61803, 0.61803, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.56155, 1.56155, 1.56155, 1.56155, 1.82025, 1.82025, 1.82025, 2.30278, 2.30278, 2.30278, 2.30278, 2.30278, 2.75660, 2.75660, 2.75660, 3.0	42.65
Hand	−2.20270, −2.20270, −2.14723, −2.14723, −2.04561, −2.0, −2.0, −1.93316, −1.93316, −1.93120, −1.93120, −1.71973, −1.71973, −1.61803, −1.59070, −1.53211, −1.53211, −1.52794, −1.52794, −1.27738, −1.27738, −1.07404, −1.07404, −0.69920, −0.63903, −0.63903, −0.61803, −0.61803, −0.25864, −0.25864, −0.08163, −0.08163, −0.06971, −0.05992, −0.5992, 0.19359, 0.19359, 0.46077, 0.46077, 0.61803, 1.27924, 1.27924, 1.29092, 1.39405, 1.39405, 1.61803, 1.61803, 2.10369, 2.10369, 2.19683, 2.19683, 2.36362, 2.50088, 2.50088, 2.54358, 2.54358, 2.71217, 2.71217, 2.83586, 2.91481	51.78

[a]CPU time in seconds on a VAX 11/730 system.

Table VII. The spectra of the soccerball and hand forms of the C_{60} cluster with bond alternation. The ratios of the weights of alternate bonds are chosen in accordance to the ratio of their bond lengths.

	Eigenvalues	Time(s)[a]
Soccerball	−2.65523, −2.65523, −2.65523, −2.59766, −2.59766, −2.59766, −2.59766, −2.02505, −2.02505, −2.02505, −2.02505, −1.63516, −1.63516, −1.63516, −1.63516, −1.63516, −1.44306, −1.44306, −1.44306, −1.30466, −1.30466, −1.30466, −1.30466, −1.30466, −0.41917, −0.41917, −0.41917, −0.17229, −0.17229, −0.17229, 0.63516, 0.63516, 0.63516, 0.63516, 0.63516, 1.01261, 1.01261, 1.01261, 1.01261, 1.01261, 1.02505, 1.02505, 1.02505, 1.02505, 1.59766, 1.59766, 1.59766, 1.59766, 1.82503, 1.82503, 1.82503, 2.32925, 2.32925, 2.32925, 2.32925, 2.32925, 2.79032, 2.79032, 2.79032, 3.03720	49.05
Hand	−2.28119, −2.27954, −2.22422, −2.22280, −2.11490, −2.07881, −2.07881, −2.00974, −2.00971, −2.00704, −2.00685, −1.78505, −1.78489, −1.68905, −1.64783, −1.59332, −1.59329, −1.58656, −1.58622, −1.32177, −1.32096, −1.11582, −1.11576, −0.70722, −0.63411, −0.63398, −0.62678, −0.62676, −0.25240, −0.25230, −0.00517, −0.00440, −0.00760, −0.01619, −0.01681, 0.24517, 0.24521, 0.51997, 0.52010, 0.68905, 1.27848, 1.27930, 1.30388, 1.41853, 1.41884, 1.62486, 1.62487, 2.15239, 2.15454, 2.23929, 2.23941, 2.41658, 2.55965, 2.55972, 2.60238, 2.60330, 2.78062, 2.78097, 2.90801, 2.99150	66.53

[a]CPU time in seconds on a VAX 11/730 system.

considered bond alternation in the football form with the intent of calculating the magnetic susceptibilities using the London theory, although these authors have not reported the spectra of these systems with bond alternation. The INDO optimization[64] leads to an alternate bond length ratio of 1.0372 for the football form and 1.0788 for the hand form. This can be easily adapted into the present computer code for the spectra of graphs by modelling the structure as a weighted graph with the alternate weights being 1.0 and 1.0372 for the football and (1.0, 1.0788) for the hand form. The calculated spectra of these weighted graphs are shown in Table VII. The corresponding delocalization energies are 0.5753 and 0.5066 for the foot and hand forms, respectively. Thus, the structures with bond alternation show greater delocalization and stabilities. It is further noted that the difference in the delocalization energies of the foot and hand forms is reduced if bond alternation is included. Also, note that the degeneracies of the eigenvalues are partially removed by bond alternation. For example, the ninefold degeneracy of the 1.0 eigenvalue is changed to 5-fold and 4-fold degeneracies for the soccer form. A related quantity of interest is the characteristic polynomial of buckminsterfullerene which has not been obtained up to now. The value of such a polynomial is that it could generate the walk patterns on the graph, spectral moments, and many other quantities. The characteristic polynomials of the unweighted soccer molecule and the hand molecule are shown in Table VIII.

The above discussions lead to the observation that the computer code employed here can be applied to a number of chemically interesting graphs such as buckminsterfullerene, graphite, and other structures.

Table VIII. The characteristic polynomials of the two forms of the C_{60} cluster.

Soccerball	$\lambda^{60} - 90\lambda^{58} + 3825\lambda^{56} - 24\lambda^{55} - 102{,}160\lambda^{54} + 1{,}920\lambda^{53}$
	$+ 1{,}925{,}160\lambda^{52} - 72{,}240\lambda^{51} - 27{,}244{,}512\lambda^{50} + 1{,}700{,}640\lambda^{49}$
	$+ 300{,}906{,}380\lambda^{48} - 28{,}113{,}600\lambda^{47} - 2{,}661{,}033{,}600\lambda^{46}$
	$+ 347{,}208{,}896\lambda^{45} + 19{,}180{,}834{,}020\lambda^{44} - 3{,}327{,}625{,}680\lambda^{43}$
	$- 114{,}118{,}295{,}000\lambda^{42} + 25{,}376{,}437{,}920\lambda^{41} + 565{,}407{,}465{,}144\lambda^{40}$
	$- 156{,}652{,}575{,}440\lambda^{39} - 2{,}346{,}799{,}508{,}400\lambda^{38} + 792{,}175{,}427{,}520\lambda^{37}$
	$+ 8{,}189{,}116{,}955{,}350\lambda^{36} - 3{,}308{,}173{,}115{,}904\lambda^{35} - 24{,}056{,}403{,}184{,}260\lambda^{34}$
	$+ 11{,}466{,}942{,}645{,}600\lambda^{33} + 59{,}443{,}188{,}508{,}110\lambda^{32}$
	$- 33{,}076{,}275{,}953{,}760\lambda^{31} - 123{,}163{,}094{,}844{,}616\lambda^{30}$
	$+ 79{,}417{,}625{,}268{,}960\lambda^{29} + 212{,}712{,}221{,}820{,}840\lambda^{28}$
	$- 158{,}412{,}719{,}276{,}240\lambda^{27} - 303{,}315{,}997{,}028{,}160\lambda^{26}$
	$+ 261{,}359{,}090{,}670{,}624\lambda^{25} + 351{,}861{,}389{,}316{,}780\lambda^{24}$
	$- 354{,}145{,}195{,}147{,}200\lambda^{23} - 324{,}375{,}523{,}213{,}200\lambda^{22}$
	$+ 390{,}055{,}074{,}762{,}240\lambda^{21} + 228{,}227{,}031{,}040{,}884\lambda^{20}$
	$- 344{,}185{,}906{,}596{,}720\lambda^{19} - 112{,}654{,}402{,}736{,}360\lambda^{18}$
	$+ 238{,}553{,}091{,}055{,}200\lambda^{17} + 29{,}617{,}003{,}666{,}920\lambda^{16}$
	$- 126{,}428{,}882{,}536{,}240\lambda^{15} + 4{,}679{,}380{,}503{,}120\lambda^{14}$
	$+ 49{,}433{,}493{,}646{,}080\lambda^{13} - 8{,}131{,}429{,}397{,}135\lambda^{12}$
	$- 13{,}627{,}897{,}407{,}360\lambda^{11} + 3{,}576{,}552{,}321{,}006\lambda^{10}$
	$+ 2{,}527{,}365{,}617{,}120\lambda^{9} - 831{,}616{,}531{,}095\lambda^{8}$
	$- 310{,}065{,}067{,}080\lambda^{7} + 108{,}565{,}938{,}200\lambda^{6}$
	$+ 26{,}034{,}025{,}632\lambda^{5} - 7{,}440{,}712{,}560\lambda^{4}$
	$- 1{,}566{,}501{,}120\lambda^{3} + 186{,}416{,}640\lambda^{2}$
	$+ 54{,}743{,}040\lambda + 2{,}985{,}984$
Hand form	$\lambda^{60} - 85\lambda^{58} - 30\lambda^{57} + 3{,}410\lambda^{56} + 2{,}370\lambda^{55} - 85{,}455\lambda^{54}$
	$- 88{,}060\lambda^{53} + 1{,}492{,}435\lambda^{52} + 2{,}042{,}520\lambda^{51} - 19{,}184{,}671\lambda^{50}$
	$- 33{,}102{,}570\lambda^{49} + 186{,}707{,}565\lambda^{48} + 397{,}573{,}430\lambda^{47}$
	$- 1{,}390{,}121{,}230\lambda^{46} - 3{,}662{,}090{,}890\lambda^{45} + 7{,}855{,}517{,}090\lambda^{44}$
	$+ 26{,}406{,}924{,}150\lambda^{43} - 32{,}343{,}677{,}920\lambda^{42} - 150{,}781{,}297{,}030\lambda^{41}$
	$+ 83{,}531{,}452{,}127\lambda^{40} + 684{,}632{,}067{,}420\lambda^{39} - 20{,}391{,}548{,}450\lambda^{38}$
	$- 2{,}465{,}009{,}904{,}960\lambda^{37} - 1{,}024{,}593{,}907{,}925\lambda^{36}$
	$+ 6{,}954{,}651{,}514{,}960\lambda^{35} + 5{,}850{,}167{,}291{,}755\lambda^{34}$
	$- 14{,}958{,}627{,}794{,}920\lambda^{33} - 19{,}866{,}419{,}057{,}880\lambda^{32}$
	$+ 22{,}976{,}293{,}749{,}140\lambda^{31} + 47{,}750{,}190{,}575{,}552\lambda^{30}$
	$- 20{,}240{,}165{,}897{,}310\lambda^{29} - 84{,}239{,}394{,}381{,}800\lambda^{28}$
	$- 5{,}118{,}045{,}723{,}720\lambda^{27} + 108{,}121{,}246{,}812{,}700\lambda^{26}$
	$+ 49{,}035{,}160{,}019{,}020\lambda^{25} - 95{,}939{,}105{,}862{,}365\lambda^{24}$
	$- 83{,}964{,}643{,}588{,}990\lambda^{23} + 50{,}155{,}043{,}610{,}115\lambda^{22}$
	$+ 82{,}672{,}021{,}164{,}630\lambda^{21} - 3{,}571{,}608{,}573{,}487\lambda^{20}$
	$- 50{,}357{,}186{,}769{,}920\lambda^{19} - 15{,}703{,}185{,}449{,}940\lambda^{18}$
	$+ 17{,}344{,}159{,}957{,}290\lambda^{17} + 11{,}732{,}414{,}861{,}315\lambda^{16}$
	$- 2{,}027{,}516{,}492{,}030\lambda^{15} - 3{,}844{,}494{,}749{,}925\lambda^{14}$
	$- 613{,}473{,}115{,}120\lambda^{13} + 562{,}931{,}871{,}920\lambda^{12}$
	$+ 231{,}386{,}370{,}840\lambda^{11} - 15{,}530{,}650{,}781\lambda^{10}$
	$- 25{,}016{,}318{,}420\lambda^{9} - 3{,}716{,}176{,}115\lambda^{8}$
	$+ 643{,}576{,}390\lambda^{7} + 232{,}389{,}855\lambda^{6}$
	$+ 14{,}277{,}570\lambda^{5} - 2{,}566{,}275\lambda^{4}$
	$- 498{,}060\lambda^{3} - 35{,}520\lambda^{2}$
	$- 1{,}200\lambda - 16$

This research was supported in part by the National Science Foundation grant no. CHE 8520556. The authors would like to thank the two referees for their invaluable comments.

References

1. K. Balasubramanian, *J. Comput. Chem.*, **3,** 357 (1984).
2. K. Balasubramanian, *J. Comput. Chem.* (in press).
3. K. Balasubramanian, *Theor. Chim. Acta,* **65,** 49 (1984).
4. K. Balasubramanian, *Comput. Chem.*, **9,** 43 (1985).
5. K. Balasubramanian, *J. Comput. Chem.*, **6,** 656 (1985).
6. K. Balasubramanian, *Studies in Physical and Theoretical Chemistry,* **28,** 243–51 (1983).
7. M. Randić, *J. Comput. Chem.*, **1,** 386 (1980).
8. M. Randić, *SIAM J. Algebraic and Discrete Methods,* **6,** 145 (1985).
9. K. Balasubramanian, *Int. J. Quantum Chem.*, **21,** 581 (1982).
10. M. Randić, *Theor. Chim. Acta,* **62,** 485 (1983).
11. M, Randić, *J. Math. Chem.*, **1,** 145 (1987).
12. K. Balasubramanian, in *Chemical Applications of Graph Theory and Topology,* R. B. King and D. H. Rouvray Eds., Elsevier, New York, in press.
13. K. Balasubramanian, in *Mathematical and Computational Concepts in Chemistry,* N. Trinajstić Ed., Ellis Harwood, Chichester, 1986, pp. 22–33.
14. K. Balasubramanian, *Chem. Rev.*, **88,** 599 (1985).
15. H. Hosoya and N. Ohkami, *J. Comput. Chem.*, **4,** 585 (1983).
16. H. Hosoya and T. Yamaguchi, *Tetrahedron Lett.*, 4659 (1975).

17. K. Balasubramanian, in *Chemical Applications of Topology and Graph Theory,* R. B. King, Ed., Elsevier, New York, in press.
18. I. Gutman, *J. Math. Chem.*, **1,** 123 (1987).
19. H. Hosoya, *Theor. Chim. Acta,* **25,** 215 (1972).
20. H. Hosoya and M. Randić, *Theor. Chim. Acta,* **63,** 473 (1983).
21. K. Balasubramanian and M. Randić, *Int. J. Quantum Chem.*, **28,** 481 (1985).
22. D. M. Cvetković, M. Doob, and H. Sachs, *Spectra of Graphs*, Academic Press, New York, (1980).
23. H. Hosoya, *Bull. Chem. Soc. Jpn.*, **44,** 2332 (1971).
24. M. Randić, B. Baker, and A. F. Kleimer, *Int. J. Quantum Chem.* (in press).
25. M. Randić and A. F. Kleimer, *Ann. NY Acad. Sci.* (in press).
26. F. Pilar, *Elementary Quantum Chemistry*, McGraw-Hill, New York, 1968.
27. I. Gutman, *Theor. Chim. Acta,* **45,** 79 (1977).
28. I. Gutman and O. E. Polansky, *Theor. Chim. Acta,* **60,** 203 (1981).
29. P. Křivka, Z. Jericević, and N. Trinajstić, *Int. J. Quantum Chem.*, **S19,** 129 (1985).
30. E. Kirby, *J. Math. Chem.* (in press).
31. R. Ramaraj and K. Balasubramanian, *J. Comput. Chem.*, **6,** 122 (1985).
32. M. Randić, *J. Comput. Chem.*, **3,** 421 (1982).
33. R. Barakat, *Theor. Chim. Acta,* **69,** 35 (1985).
34. K. Balasubramanian and R. Ramaraj, *J. Comput. Chem.*, **5,** 441 (1985).
35. Tang Au-chin and K. Yuan-san, *Sci. Sinica,* **1,** 449 (1976); ibid. **3,** 218 (1977).
36. R. B. King, *Theor. Chim. Acta,* **56,** 269 (1980).
37. R. B. King, *Theor. Chim. Acta,* **44,** 223 (1977).
38. L. Glass, *J. Chem. Phys.*, **63,** 1325 (1975).
39. K. Balasubramanian, *J. Math. Chem.* (in press).
40. O. E. Polansky and M. Zandler, *J. Mol. Structure,* **84,** 361 (1982).
41. I. Ohmine, M. Karplus, and K. Schulten, *J. Chem. Phys.,* **68,** 2298 (1978).
42. E. Hückel, *Z. Physik*, **76,** 628 (1932).
43. C. A. Coulson and H. C. Longuet-Higgins, *Proc. Roy. Soc.*, **A192,** 16 (1947).
44. C. A. Coulson, *Proc. Roy. Soc.*, **A164,** 383 (1938).
45. E. C. Kirby, *Croat. Chem. Acta* (in press).
46. N. Trinajstić, *Chemical Graph Theory*, Volume I, CRC Press, Boca Raton, FL, 1983.
47. J. Greenstadt, *The Determination of the Characteristic Roots of a Matrix by the Jacobi Method,* Mathematical Methods for Digital Computers, A. Ralston and H. Wilf, Eds., Wiley, New York, 1960.
48. A. S. Householder, *The Theory of Matrices in Numerical Analysis,* Blaisdell, New York, 1964.
49. A. S. Householder and F. L. Bauer, *Numerical Math.*, **1,** 29 (1959).
50. J. H. Wilkinson, *Comput. J.*, **3,** 23 (1960).
51. J. Ortega, in *Mathematical Methods for Digital Computers*, Anthony Ralson and Herbert S. Wilf, Eds., Vol. II, Wiley, 1967, pp. 65–93.
52. F. Prosser is the author of the GIVENS subroutine originated from Indiana University.
53. Yang Huyin, *Applied Mathematics for Computer Science*, Vol. I, Metallurgical Industry Publishing Company, Beijing, 1979.
54. H. W. Kroto, J. R. Health, S. C. O'Brien, R. F. Curl, and R. E. Smalley, *Nature,* **318,** 162 (1985).
55. Y. Liu, S. C. O'Brien, Q. Zhang, J. R. Heath, F. K. Tittle, R. F. Curl, H. W. Kroton, and R. E. Smalley, *Chem. Phys. Lett.*, **126,** 215 (1986).
56. J. R. Heath, S. C. O'Brien, Q. Zhang, Y. Liu, R. F. Curl, H. W. Kroto, F. K. Tittle, and R. E. Smalley, *J. Am. Chem. Soc.*, **107,** 7779 (1985).
57. S. C. O'Brien, J. R. Heath, H. W. Kroto, R. F. Curl, and R. E. Smalley, *Chem. Phys. Lett.*, **132,** 99 (1986).
58. A. J. Stone and D. J. Wales, *Chem. Phys. Lett.*, **128,** 501 (1986).
59. D. J. Klein, T. G. Schmalz, G. E. Hite, and W. A. Seitz, *J. Am. Chem. Soc.*, **108,** 1301 (1986).
60. D. J. Klein, W. A. Seitz, and T. G. Schmaltz, *Nature,* **323,** 703 (1986).
61. T. G. Schmalz, W. A. Seitz, D. J. Klein, and G. E. Hite, *Chem. Phys. Lett.*, **130,** 203 (1986).
62. A. D. H. Haymet, *Chem. Phys. Lett.*, **122,** 421 (1985).
63. R. C. Haddon, L. E. Brus, and K. Raghavachari, *Chem. Phys. Lett.*, **125,** 459 (1986); **131,** 165 (1986).
64. Tai-ichi Shibuya and M. Yoshitani, *Chem. Phys. Lett.*, **137,** 13 (1987).
65. V. Elser and R. C. Haddon, *Nature*, **325,** 792 (1987).
66. R. B. Mallion, *Nature*, **325,** 780 (1987).