Polyhedral approximation approach to molecular orbital graphics

Akio Koide, Akio Doi and Koichi Kajioka

IBM Japan Ltd., Science Institute, 5-19 Sanban-cho, Chiyoda-ku, Tokyo 102, Japan

Interactive molecular orbital display and manipulation system is developed under the concept of polyhedral approximation of equivalued surfaces of orbital and density functions. Two new methods for generating polyhedral data, 'cell-based' and 'propagation' methods, are presented. The use of stored polyhedral data largely simplifies the shaded-image synthesis, direct manipulation, and animation display of molecular orbitals. Manual overlay of frontier orbitals is useful to estimate the geometries of chemically reacting molecules.

Keywords: computer graphics, molecular orbital, polyhedral approximation, shaded images, animation display, frontier orbitals

received 10 December 1985, accepted 6 January 1986

Molecular graphics has been concerned mostly with the display of 3D configurations of atoms or functional groups[1]. However, displaying molecular orbital structures is much more helpful for chemists investigating the nature of chemical reactivity[2,3]. Molecular orbital calculations are becoming more popular due to the growth of computer power and lower costs. The purpose of this paper is to highlight desired functions and their implementations in molecular orbital graphics, by describing the system used by the authors. The paper will propose a polyhedral approximation approach for highly interactive molecular orbital display and manipulation. Two new methods, (i) 'cell-based' and (ii) 'propagation', will be presented in order to construct the polyhedral data approximating the equivalued surfaces of molecular orbital or electron density functions.

Molecular orbitals are functions of a single electron used to construct approximate solutions to the Schrödinger equation for electrons[4]. Their arguments are the coordinates of the electron and their values are generally complex numbers. By the appropriate choice of equivalent molecular orbitals, all values of functions can be made into real numbers. In this paper, the values are assumed to be real. The number of molecular orbitals in a single molecule is dependent on the required accuracy and the choice of solving methods such as SCF[4], MCSCF[5], CNDO, MINDO, and extended Hückel methods.

The equivalued surfaces are defined by

$$\varphi(x, y, z) = C \quad (1)$$

For molecular orbital and electron density functions given by $\varphi(x, y, z)$. In our system, the surface constant C in equation (1) is treated as a user-specified parameter. The use of multiple surface constants yields a 3D contour map. We simplified the graphic handling by the use of polyhedral data approximated to the equivalued surfaces. Shaded images can be generated rapidly from polyhedral data[6]. The rotation and overlay of molecular orbitals can be quickly performed on the polyhedral data. The playback display[8] of the created polyhedral data offers interactive animation capability. Due to the frontier orbital theory[2,7], the geometries of chemically reacting molecules can be estimated by overlaying the frontier orbitals, i.e. highest occupied molecular orbitals (HOMO) and lowest unoccupied molecular orbitals (LUMO).

System overview of molecular graphics system

Figures 1 and 2 show an overview of the system and the hardware configuration, respectively. The host computer is an IBM 3081 running VM/CMS; the graphic device is an IBM5080 graphics system, which has local 3D clipping, viewing transformation, area-filling facility and so on. The IBM5083 colour monitor with 1024 × 1024 pixels can display 256 colours simultaneously out of a possible 4096 colours from the pallet.

Molecular orbital data are generated outside the system. Their data files consist of:

- atomic orbital specification parameters and,
- expansion coefficients of molecular orbitals into atomic orbitals. Thus, as long as data files are previously prepared, the system can display nonstandard molecular orbitals, such as interaction frontier orbitals[7] and localized molecular orbitals[11], which are used in the analysis of reaction sites and stable molecular geometry, respectively. Electron density is calculated from molecular orbital data in the case of the SCF wavefunction[4].

The molecular orbital graphics system is composed of a polyhedral approximation subsystem, a display and manipulation subsystem, and an animation display subsystem. When surface constants are specified for the molecular orbitals, polyhedral data is generated and stored as graphic objects by the polyhedral approximation subsystem. The data is manipulated by using

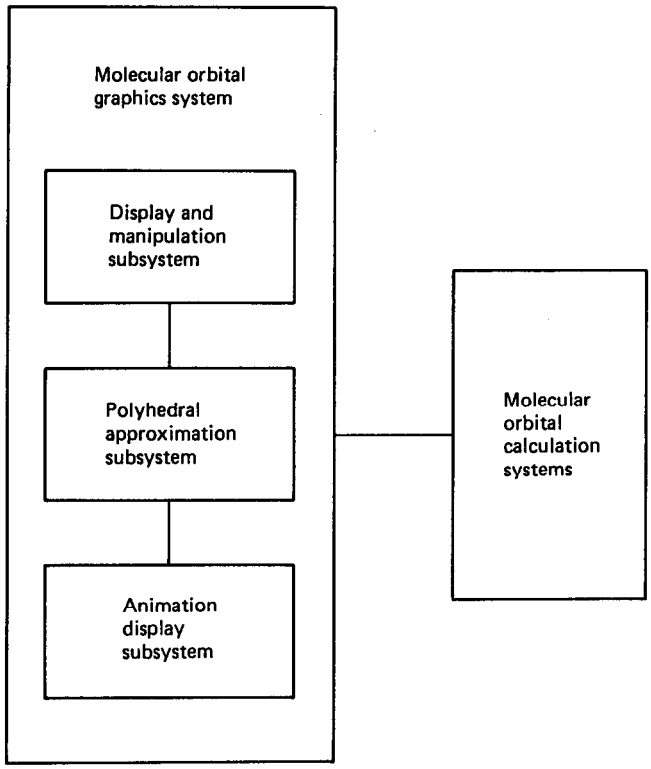

Figure 1. Overview of molecular orbital graphics system

the 'surface table', as shown in Figure 3; this will be explained in detail later in this paper. Once the polyhedral approximation is done, a user can display and

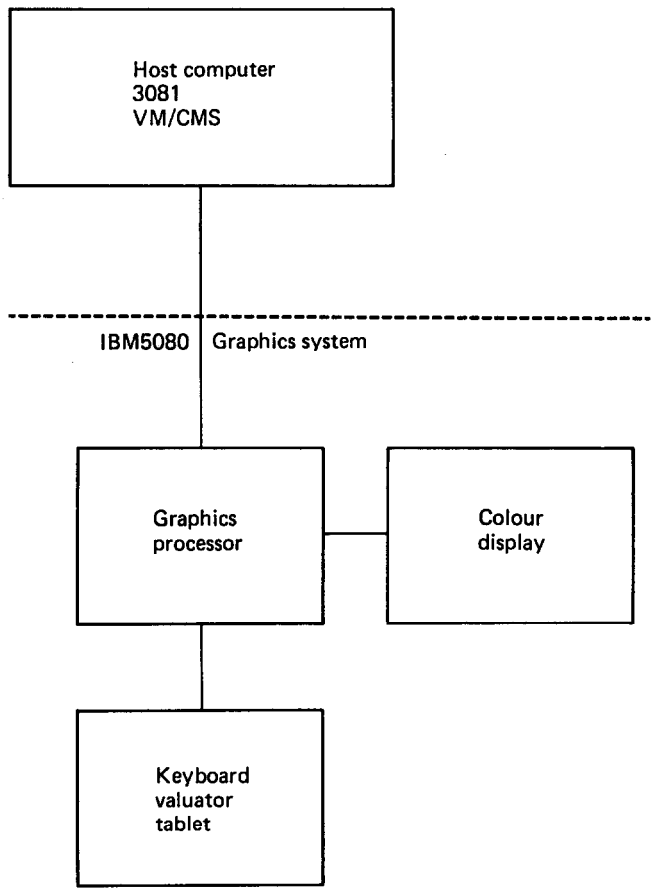

Figure 2. Hardware configuration

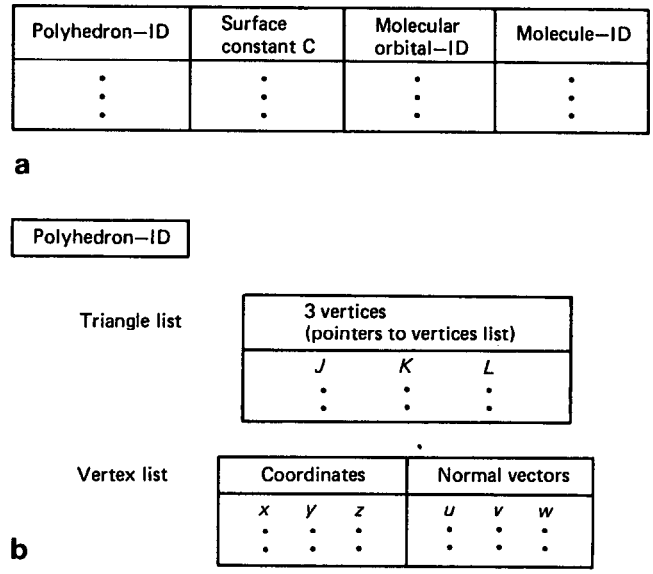

Figure 3. Data structure of polyhedral approximation; (a) surface table, (b) polyhedral data

overlay molecular orbitals in order to estimate the geometry of the two reacting molecules by using the display-and-manipulation subsystem. Alternatively he can investigate the deformation of molecular orbitals along the reaction path by using the animation subsystem.

Display and manipulation subsystem

The display and manipulation subsystem offers three types of display by using polyhedral data generated from molecular orbital and electron density functions. These displays are:

- polymarker display,
- vector display, and
- smoothly shaded image display.

The manipulation types are used:

- to pick molecular orbitals or molecules,
- to set the display types and attributes,
- to move molecules or the eye position, and
- to read atomic coordinates of displayed geometries.

Colour Plates 1–3 are examples of the different display types for the same molecular orbital, HOMO of SiH_2CH_2. The surface constants are ± 0.08. The colour difference indicates the sign of the constants. Colour Plate 4 is an example of a stereo contour map using the polymarker display; the orbital is LUMO of H_2O. The surface constants are ± 0.10, ± 0.08 and ± 0.06. The molecular orbitals in the examples were calculated using the SCF method[4] with the 6-31G contracted Gaussian basis set. Generally, the shape of LUMO is sensitive to the choice of the Gaussian basis set.

The purpose of polymarker and vector displays is to support the real-time smooth rotation and translation of molecular orbitals for a user's input through the valuators (dials) of the graphics workstation. These displays were implemented by using the local facility of the IBM5080 Graphics System. The hidden line treatment was not performed. Colour Plate 5 is an example of manual overlay of frontier orbitals in order to estimate

the geometry of the transition state of the reaction $C_2H_4O + F^- \rightarrow CFH_2CH_2O^-$. Here the transition state is a saddle point of the potential energy surface, which divides reactants from products. The displayed orbitals are LUMO of C_2H_4O and HOMO of F^-. Equivalued surfaces were generated with the constant ± 0.10. The intersection of the surfaces between different molecules indicates graphically the interaction of molecular orbitals. For two reacting molecules of the closed shell type, it is known[2,7] that the interaction among occupied molecular orbitals increases the energy, and that the interaction between HOMO and LUMO decreases the energy. Thus, the geometry of the transition state can be estimated as the geometry where HOMO and LUMO intersect each other. The obtained atomic coordinates of reacting molecules can be used as input data for other simulation programs.

The host computer generates the smoothly shaded images by the linear interpolation of normal vectors inside each triangle from polyhedral data[12]. The hidden surface was implemented by using the priority algorithm[6], which is essentially based on sorting triangles through the polyhedral data. The movement of graphic objects is specified through the keyboard by rotation angles around the x, y and z axes and by the shift width along x, y and z axes; that is to say, the movement in the shaded image display is discontinuous; while that in the polymarker and vector displays in continuous. Colour Plate 6 shows interaction frontier orbitals[7] in the transition state, the geometry of which was computed by relaxing forces from the initial geometry of Colour Plate 5. The comparison between Colour Plates 5 and 6 illustrates that the manual overlay of frontier orbitals is effective in estimating the approximate geometries of transition states.

Animation display subsystem

The animation subsystem offers a real-time playback display[8] of a sequence of previously generated polyhedral data. Polymarker, vector and area-filling displays are supported. Colour Plate 7 is an example of the area-filling display of the same orbital with the same surface as Colour Plates 1–3. Area-filling display is a simple version of the shaded image display which utilizes the local area-filling functions of the IBM5080 Graphics System. Animation display on graphic devices is advantageous because the user can interactively control the colour of the orbitals, the light and eye positions, and zoom; in addition to time control.

The subsystem consists of the graphic order generator, the display manager and the timer. Before the animation is displayed, graphic orders are generated from the polyhedral data in the host computer by the graphic order generator. The display manager monitors the animation display and consists of the input/output handler and the attention handler. The I/O handler processes input/output among external storage devices, the host computer, and the graphics workstation while pictures are synthesized inside the workstation. The attention handler offers the user interactive control even while the animation is being displayed. The user can initiate communication with the system by pressing a program function-key on the alphanumeric keyboard or the program function keyboard. The state messages such as 'displaying', 'wait for user's input', 'rotate', 'zoom up', etc. are shown on the indicator to the bottom. The timer is used for time control of animation.

Colour Plate 8 shows one of the scenes in the animation display of the reaction $CH_2SiH_2 + H_2O \rightarrow CH_3SiH_2OH$ along the Intrinsic Reaction Coordinate (IRC)[9,10]. The IRC is the path of steepest descent which starts at a saddle point. The IRC was computed from the first derivatives of SCF energy by solving an ordinary differential equation[9] with the fourth-order of the Runge–Kutta method[18]. The orange and brown portion (π-bonding) is the orbital losing charge. The blue and green portion is the orbital gaining charge. These are computed by the orbital transformation of the SCF wavefunction of the reacting total system[7]. From this animation, it was estimated that the charge transfer from the π-bonding orbital makes the eclipsed geometry unstable.

Definition and data structure of polyhedral approximation

We call a structured set of points which satisfies equation (1), the polyhedral approximation of the equivalued surfaces. The polyhedral approximation is described by the 'surface table' and 'polyhedral data' as shown in Figure 3. Each set of polyhedral data corresponds to one of the connected components of the equivalued surfaces. Polyhedral data is manipulated and displayed from the surface table by specifying molecule identifiers, molecular orbital identifiers, surface constants C in equation (1), and polyhedron identifiers or their combinations. Polyhedral data is generated for specified surface constants from molecular orbital data by the system. All faces of the polyhedral data are restricted to triangles to simplify data generation and graphic synthesis. The equivalued surfaces which are generated from the same molecular orbital or electron density function do not intersect.

The polyhedral data consists of the polyhedron identifier, the triangle list, and the vertex list. The vertex list is a collection of the coordinates and normal vectors of the vertices on the equivalued surface. The normal vectors $(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w})$ are defined by

$$\boldsymbol{u} = N\partial\varphi/\partial x,\ \boldsymbol{v} = N\partial\varphi/\partial y,\ \boldsymbol{w} = N\partial\varphi/\partial z$$

and

$$N = [(\partial\varphi/\partial x)^2 + (\partial\varphi/\partial y)^2 + (\partial\varphi/\partial z)^2]^{-1/2} \tag{2}$$

The normal vectors are required for the synthesis of shaded images[12,16]. No repetition of identical coordinates is allowed in the vertex list. The triangle list is a collection of the triangles expressed by the identifiers of their three vertices. The pointers to the vertex list are used as the vertex identifiers. If the edges are required for the display, graphic routines will generate them from the triangle and vertex lists. Any two vertices of the triangles form an edge, but the pair with the descendent order of pointers are ignored to avoid the double counting of the edges.

Cell-based method for constructing polyhedral approximation

The method proposed here consists of three steps:

- the evaluation of molecular orbital or electron density functions on the grid points,
- the linear interpolation of the functions inside of tetrahedrons spanned by the grids points, and
- the generation of vertices and triangles.

The normal vector on each vertex is approximated by the averages of normal vectors of the triangles adjacent to it. Instead of cubes, tetrahedrons were chosen as primitive cells for the interpolation because function values evaluated at four points are necessary and sufficient for a linear interpolation of any functions defined with respect to 3D coordinates. We call this method the cell-based method.

The molecular orbital or electron density function is evaluated on the grid points (i, j, k) whose coordinates are given by

$$(x_0 + i\,a, y_0 + j\,a, z_0 + k\,a) \tag{3}$$

Here the constant a is the grid interval, and i, j and k are integers. Only the grid interval is a parameter for controlling the quality of polyhedral approximation. In the implementation, the grid points for the function to be evaluated were appropriately restricted around the atoms. The following five types of tetrahedrons are formed from the grid points.

Tetrahedron 1 (i', j', k'), (i'', j', k'), (i', j'', k') and (i', j', k'')
Tetrahedron 2 (i'', j', k'), (i', j'', k'), (i'', j'', k') and (i'', j'', k'')
Tetrahedron 3 (i'', j', k'), (i'', j', k''), (i', j', k'') and (i'', j'', k'')
Tetrahedron 4 (i', j'', k'), (i', j'', k''), (i', j', k'') and (i'', j'', k'')
Tetrahedron 5 (i'', j', k'), (i', j'', k'), (i', j', k'') and (i'', j'', k'')
(4)

Here $i' = i$ for even values of i; otherwise $i' = i + 1$. $j' = j$ for even values of j; otherwise $j' = j + 1$. $k' = k$ for even values of k; otherwise $k' = k + 1$. i'', j'', and k'' denote $2i + 1 - i'$, $2j + 1 - j'$, and $2k + 1 - k'$, respectively.

When the linear interpolation is applied inside of each tetrahedron, the equivalued surface forms a polygon, if one exists. The vertices of the generated polygons are located on the edges of the tetrahedrons. Their positions can be determined only from the values of the function at both ends of the edges. Suppose that f_1 and f_2 are the values at the grid points connected by the edges of the tetrahedrons and that (x_1, y_1, z_1) and (x_2, y_2, z_2) are their coordinates. If $(f_1 - C)(f_2 - C) < 0$, then the coordinates of the vertex of the polygon are given by

$$(x, y\,z) = (1 - t)(x_1, y_1, z_1) + t(x_2, y_2, z_2)$$

where

$$t = (f_2 - f_1)/(C - f_1) \tag{5}$$

Therefore the polygons generated from different tetrahedrons are continuously connected to each other and form the polyhedral approximation. The generation of polygons by the linear interpolation can be classified into 15 cases by the signs of the differences, given by

$$\varphi(x, y, z) - C \tag{6}$$

at the four grid points of the tetrahedron as listed in Table 1. In case 9, the tetragon must be divided into two triangles. In cases 13, 14, and 15, the triangles are identical to the faces of tetrahedrons. In order to avoid duplication, the triangles for cases 14 and 15 are not registered.

A generated polyhedral data may be composed of several polyhedrons. The method cannot automatically separate the polyhedral data into connected components of polyhedrons. Post-processing, the regrouping of triangles and vertices, is required so that each of the groups should form connected polyhedrons. The regrouping can be carried out by applying a depth-first search algorithm[13] of the graph theory; the number of the required operations is proportional to the number of the edges and vertices.

Propagation method for constructing polyhedral approximations

The method proposed here consists of two steps: the search for the vertices which will be the seeds of new polyhedrons and the construction of a new polyhedron by the successive computation of vertices around the incomplete edges are defined as the edges which are adjacent to less than one triangle on the triangle list. We call this method the propagation method.

The advantages of the propagation method are:

- that it can generate high quality polyhedral data by using fewer vertices than the cell-based method,
- that it can directly yield the previously defined data structure of the polyhedral approximation, and
- that the vertices are composed of points accurately located on the equivalued surfaces.

Colour Plates 1–8 are constructed by the propagation method and Colour Plate 9 is constructed by the cell-based method. On the other hand, the implementation of the propagation method is somewhat more complicated than the cell-based method since poor control of successive generations of vertices and triangles will cause the overlapping of polyhedral surfaces as illustrated in Figure 4. Also the method requires the evaluation of partial derivatives of the functions in order to compute normal vectors on the vertices and to generate new vertices by a Newton–Raphson type of iteration procedure.

A brief outline of the propagation method will be given here. First, the molecular orbital or electron

Table 1. Classification of generated polygons inside of tetrahedron. Here N, Z and P are the numbers of grid points which have negative, zero, and positive values of $\varphi(x, y, z) - C$, respectively

Case	Number of grid points N	Z	P	Generated polygons
1	4	0	0	Nil
2	0	0	4	Nil
3	3	1	0	Nil
4	0	1	3	Nil
5	2	2	0	Nil
6	0	2	2	Nil
7	3	0	1	1 triangle
8	1	0	3	1 triangle
9	2	0	2	1 tetragon
10	2	1	1	1 triangle
11	1	1	2	1 triangle
12	1	2	1	1 triangle
13	1	3	0	1 triangle
14	0	3	1	1 triangle
15	0	4	0	4 triangles

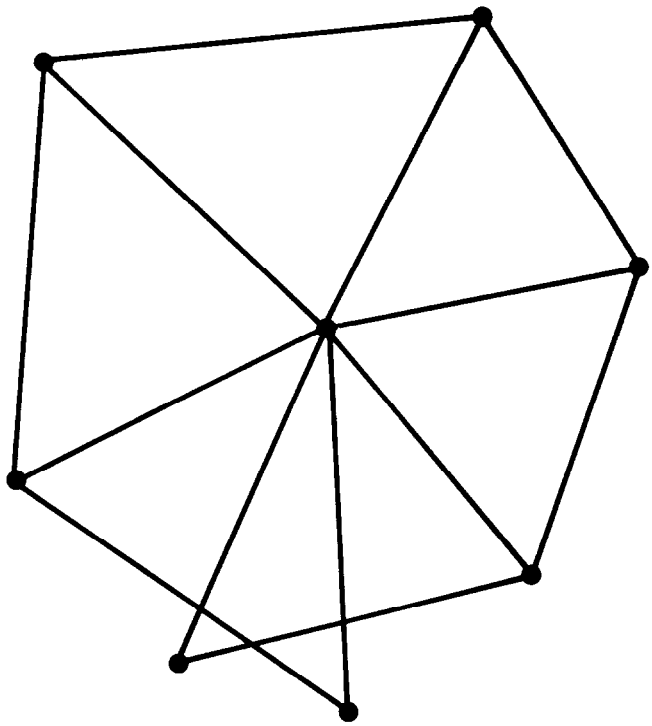

Figure 4. Surface overlapping caused by poor control of polyhedron generation

density functions on the grid points are evaluated. Also, for the performance, the grid points of the evaluation must be appropriately restricted around atoms. The adjacent grid points (i, j, k) and $(i, j, k + 1)$ are searched where the differences of equation (6) change the sign. If they are found, the line connecting them is checked to see if it intersects any previously generated polyhedrons or not. If the line intersects no polyhedrons, the generation of a new polyhedron is initiated.

The first vertex of the polyhedron is obtained by the Newton–Raphson type of iteration procedure

$$t_n = (C - \varphi(\boldsymbol{p}_n))/\partial\varphi/\partial\boldsymbol{p}_n \cdot \partial\varphi/\partial\boldsymbol{p}_n$$

and

$$\boldsymbol{p}_{n+1} = \boldsymbol{p}_n + t_n \partial\varphi/\partial\boldsymbol{p}_n \quad (7)$$

Here $\boldsymbol{p}_n$ are the position vectors (x_n, y_n, z_n). The first trial position vector $\boldsymbol{p}_0$ is taken to the grid point which gives a smaller value of the difference of equation (6). The convergency of $\boldsymbol{p}_n$ is judged by the simultaneous satisfaction of the conditions

$$|C - \varphi(\boldsymbol{p}_n)| < \Delta C$$

and

$$|\boldsymbol{p}_{n+1} - \boldsymbol{p}_n| < \Delta \boldsymbol{p} \quad (8)$$

The iteration procedure in equation (7) can be derived as follows: the continuously differentiable functions can be expanded with respect to a small displacement vector $\boldsymbol{d}$ into

$$\varphi(\boldsymbol{p} + \boldsymbol{d}) = \varphi(\boldsymbol{p}) + \partial\varphi/\partial\boldsymbol{p} \cdot \boldsymbol{d} + \ldots \quad (9)$$

By the substitution of $t\partial\varphi/\partial p$ for $\boldsymbol{d}$ in the expansion, one obtains the iteration procedure in equation (7).

The successive generation of the vertices and triangles around the incomplete edges consists of the following primitive operations.

- The generation of one vertex and one edge. (P1).
- The generation of one edge. (P2).
- The generation of one triangle. (P3).

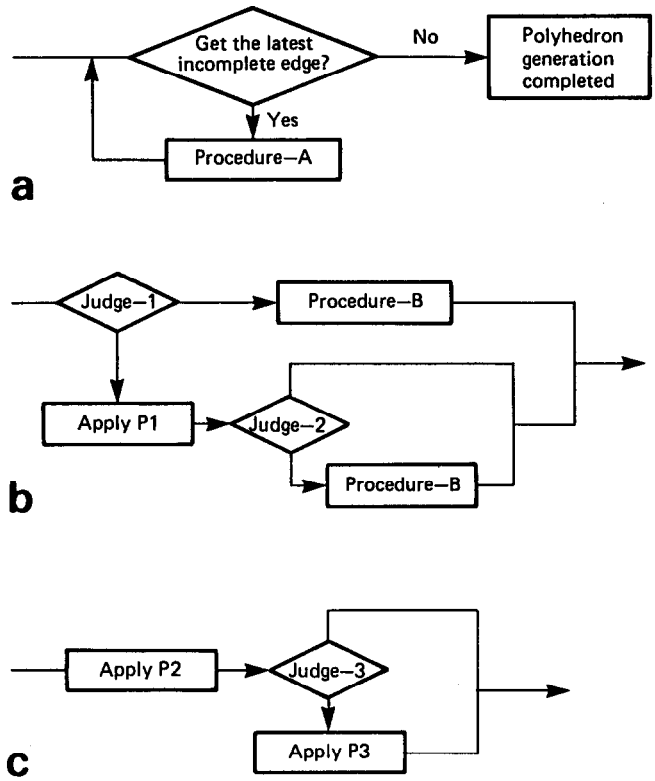

Figure 5. Control of polyhedron generation; (a) highest level, (b) procedure A (second level), (c) procedure B (third level)

In the first place, operation P_1 is applied to the first vertex. Then operations P1, P2 or P3 are applied successively to the generated incomplete edges using the flow chart in Figure 5. If all the incomplete edges are deleted, the generation of a new polyhedron will be completed. The incomplete edges are generated by the operations P1 and P2, and they are deleted by the operation P3. All the incomplete edges are supposed to be directed in order to distinguish between inner and outer sides of the incomplete edges. The direction of each edge is expressed as the pair of from-vertex and to-vertex.

Judge conditions in the flow chart

To explain the judge conditions in the flow chart of Figure 5, the edge condition is introduced. If the following conditions (E_1) or (E_2) are satisfied between two vertices, it is possible to state that the edge condition is satisfied.

- The distance between the two vertices is less than a given minimum distance. (E_1)
- The distance between the two vertices is less than a given maximum distance and the deflection angle between their normal vectors is smaller than a given threshold. (E_2)

Then Judge-1, Judge-2 and Judge-3 in the flow chart are stated as follows.

- Judge-1: take from-vertex of the latest incomplete edges. If there is a vertex on the incomplete edges such that it and the from-vertex satisfy the edge condi-

tion but they are not connected by an incomplete edge, then do the operation P2; otherwise do P1.
- Judge-2: take newly created vertex. If the edge condition is satisfied between it and the vertex which is connected to it through two incomplete edges, then do the operation P2. Otherwise do not do anything.
- Judge-3: if a newly created edge and the other two incomplete edges form a triangle and the edge condition are satisfied among its vertices, then do the operation P3.

In our experience, this control works well in excluding the overlapping of polyhedral surfaces although it cannot be proved mathematically.

Generation of one vertex and one edge

The length of edges and the deflection angles of normal vectors in polyhedral data are controlled by the above-mentioned edge condition and the operation P1. Operation P1 will be explained here.

A new vertex is obtained by the iteration procedure of equation (6) with an initial trial position which is appropriately displaced from the from-vertex of the latest incomplete edge. Let us denote the from-vertex by V. The displacement vector is chosen so that (i) it is perpendicular to the normal vector of the vertex V, (ii) it is deflected outside by 60° from the latest incomplete edge and (iii) the edge condition is satisfied. If the angle between the two incomplete edges at the vertex V is less than 120°, (ii) is replaced as follows. (ii') The direction of the displacement vector is taken to the bisector of the two edges.

CONCLUSION

The interactive manipulation aspect was emphasized for molecular orbital graphics in the present paper. This is important not only to understand the 3D structure of molecular orbitals, but also to estimate the geometry of reacting molecules, since the overlap of molecular orbitals gives valuable spatial information on the orbital interactions between them[7].

The polyhedral approximation approach was proposed to easily and economically implement flexible interactive manipulations and real time animation display. Many methods are available since polyhedral data is standard in computer graphics. For example, smoothly shaded images can be generated rapidly from the polyhedral data[6,12]. Shaded image display is more suitable for investigating spatial features of molecular orbitals than the conventional plannar contour (wire frame) representation[14]. Blinn's approach[17], i.e., the direct generation of shaded images without the use of polyhedral data, is not economical because each manipulation requires the time-consuming evaluation of molecular orbital or density functions.

Two new methods were proposed for constructing polyhedral data from the equivalued surfaces of arbitrary functions. In the conventional method[15], the contours are first computed on many parallel planes and then the points on the plannar contours are appropriately chosen and connected to each other. Our methods are more direct and simpler than the above described conventional method. In our system, the propagation method is utilized to generate polyhedral data. Since it produces a uniform distribution of vertices on the equivalued surfaces, the method is better suited to the polymarker display. On the other hand, the uniform distribution of vertices is not so important for creating shaded images. If a graphics workstation has a powerful shading engine and the polymarker display is not required, then the cell-based method is recommended because it is compact and simple in coding.

ACKNOWLEDGEMENT

The authors are indebted to Dr Kenichi Fukui (the President of Kyoto University of Industrial Arts and Textile Fibres) and Prof Hiroshi Fujimoto (University of Kyoto) for the discussion of interpreting displayed frontier orbitals in relation to chemical reaction.

REFERENCES

1 **Morffew, A J** 'Bibliography for molecular graphics' *J. Mol. Graph.* Vol 1 (1983) pp 17–23; **Richards, W G** 'Recent literature' *J. Mol. Graph.* Vol 2 (1984) pp 21–22; **Tucker, J B** 'Designing molecules by computer' *High Tech.* (January 1984) pp 52–59; **Dubis, J E et al.** 'Chemical ideograms and molecular computer graphics' *Visual Comput.* Vol 1 (1985) pp 49–64; **Vinter, J G** 'Molecular graphics for the medical chemist' *Chem. Br.* Vol 21 (1985) pp 32–36; **Hassall, C H** 'Computer graphics as an aid to drug design' *Chem. Br.* Vol 21 (1985) pp 39–46

2 **Fukui, K et al.** 'A molecular orbital theory of reactivity in aromatic hydrocarbons' *J. Chem. Phys.* Vol 20 (1952) pp 722–725; **Fukui, K** 'A simple quantum-theoretical interpretation of the chemical reactivity of organic compounds' in **Lowdin, P O and Pullman, B (eds)** *Molecular orbitals in chemistry physics, and biology* Academic Press (1964) pp 513–537; **Fukui, K** 'An MO-theoretical illumination for the principle of stereoselection' *Bull. Chem. Soc. Japan* Vol 39 (1966) pp 498–503; **Fukui, K** 'Recognition of stereochemical paths by orbital interaction' *Acc. Chem. Res.* Vol 4 (1971) pp 57–64

3 **Woodward, R B and Hoffman, R** 'Stereochemistry of electrocyclic reactions' *J. Am. Chem. Soc.* Vol 87 (1965) pp 395–397; **Woodward, R B and Hoffmann, R** 'Selection rules for sigmatropic reactions' *J. Am. Chem. Soc.* Vol 87 (1965) pp 2511–2513; **Woodward, R B and Hoffmann, R** *'The conservation of orbital symmetry'* Academic Press (1970)

4 **Roothaan, C C J** 'New developments in molecular orbital theory' *Rev. Modern Phys.* Vol 23 (1951) pp 69–89; **Roothaan, C C J** *Rev. Modern. Phys.* Vol 32 (1960) pp 179; **Roothaan, C C J and Bagus, P S** *Methods Computat. Phys.* Vol 2 (1963) p 47

5 **Yeager, D L and Jorgensen, P** 'Convergency studies of second and approximate second order multi-configurational Hatree–Fock procedures' *J. Chem. Phys.* Vol 71 (1979) pp 755–760

6 **Newman, W M and Sproull, R F** 'Principles of interactive computer graphics' McGraw-Hill (1981) pp 380–384

7 **Fujimoto, H and Fukui, K** 'Molecular interaction through orbitals' in **Ratajczak, H and Orville-**

Thomas, W J (eds) *Molecular interaction* John Wiley & Sons (1980) pp 89–116; **Fukui, K et al.** 'Interaction frontier orbitals' *J. Am. Chem. Soc.* Vol 103 (1981) pp 196–197; **Fujimoto, H et al.** 'Orbital transformation analysis. A simplification in the description of charge transfer' *J. Phys. Chem.* Vol 88 (1984) pp 3539–3544

8 **Magnenat-Thalmann, N and Thalmann, D** *Computer animation* Springer-Verlag (1985) p 16

9 **Fukui, K et al.** 'Constituent analysis of the potential gradient along a reaction coordinate, method and application to CH_4 + T reaction' *J. Am. Chem. Soc.* Vol 97 (1975) pp 1–7

10 **Miller, W H et al.** 'Reaction path Hamiltonian for polyatomic molecules' *J. Chem. Phys.* Vol 72 (1980) pp 99–112

11 **Edmiston, C and Ruedenberg, K** 'Localized atomic and molecular orbitals' *Rev. Modern Phys.* Vol 35 (1963) pp 457–465; **England, W E et al.** 'Localized molecular orbitals: a bridge between chemical intuition and molecular quantum mechanics; *Top. Curr. Chem.* Vol 23 (1971) pp 31–123

12 **Gouraud, H** 'Continuous shading of curved surface' *IEEE Trans. Comput.* Vol C-20 (1971) pp 623–629

13 **Sedgewick, R** *Algorithms* Addison-Welsley (1983) pp 381–383

14 **Jorgensen, W L** *PSI/77* Quantum Chemistry Exchange Program, available from Indiana University, USA

15 **Ganapathy, S and Dennely, T G** 'A new general triangulation method for plannar contour' *Comput. Graph.* Vol 16 (1982) pp 69–74; **Boissonat, J-H** 'Geometric structures for three-dimensional shape representation' *ACM Trans. Graph.* Vol 3 (1984) pp 266–286

16 **Foley, J D and Van Dam, A** 'Fundamentals of interactive computer graphics' Addison-Wiley (1984) p 575

17 **Blinn, J F** 'A generalization of algebraic surface drawing' ACM *Trans. Graph.* Vol 1 (1982) pp 235–256

18 **Lambert, J D** *Computational methods in ordinary differential equations* John Wiley & Sons (1973) p 120, pp 225–228