Matrix diagonalization

"The Hückel method"

Brief description:

This work is based on the realization of a program used to perform Hückel calculations for conjugated hydrocarbons with an unknown number of atoms. The whole structure of the code in this work is based on subroutines to be able to make all the necessary calculations. The objective or main procedure is the diagonalization of symmetric matrices from the Jacobi method in order to diagonalize the Hamiltonian matrix and obtain the whole set of eigenvalues, which are the energies of the molecular orbitals and their corresponding eigenvectors (coefficients). From this point other calculations within the program are performed such as: the total energy of the π-system, the π-electron occupations and finally, the bond orders using a Mulliken analysis.

Algorithm:

The Hückel method is a very simple linear combination of atomic orbitals (LCAO) molecular orbitals method for the determination of energies of molecular orbitals of π- electrons in π-delocalized molecules.

In this method the calculation is based on a basis of atomic orbitals, which contain the 2pz orbitals of the carbon atoms. The procedure is based on trying to solve the set of equations determined from this equation (Fig. 1):

$$\sum_{j=1}^{n} c_j(H_{ij} - ES_{ij}) = 0 \quad (i = 1, \cdots, n)$$

- n → number of AOs in the basis.
- Hij → elements of the Hamiltonian matrix.
- Sij → elements of the overlap matrix.
- E → eigenvalues.
- cj → components of the eigenvectors.

Fig. 1: *Equation for Hückel method*

Finding nontrivial solutions to the simultaneous equations can be achieved by finding values of E such that:

$$\det([H_{ij} - ES_{ij}]) = 0$$

Fig. 2: *Equation to find E values*

Finally, to form the matrices corresponding to H and S, we assume that the atomic orbitals form an orthonormal set, that allows us to formulate that S is equivalent to the identity matrix (I) and the matrix corresponding to the Hamiltonian can be formulated as follows (Fig. 3):

$$H_{ij} = \begin{cases} \alpha, & i = j; \\ \beta, & i, j \text{ adjacent}; \\ 0, & \text{otherwise}. \end{cases}$$

Fig. 3: *Parametrization of the H matrix*

Implementation of the code:

The code created for this work has a well-defined structure. This structure is composed of nine distinct parts, which are divided into eight subroutines and the main program. It must be said that each of the subroutines has its function within the main program and in each of them specific calculations or procedures are performed.

The first subroutine corresponds to the creation of the topological matrix. This subroutine is divided into different steps until the target matrix is created. First, starting from a do loop and with an if loop inside it, only the number of carbon atoms in the structure is read and their positions are placed. Then the document used as input is closed and the distance matrix is created. This matrix is created from three do loops: the first two to go through the whole matrix from the initial atom and the last one is used to subtract the positions of consecutive atoms and enter the data in the equation to calculate these distances. The next part corresponds to the creation of the reduced matrix, done in a trivial way from the positions in the distance matrix. Finally, the topological matrix itself is created from the previously created reduced distance matrix. The main element of this process is the if that is used, since the distance limits are used to say if they are linked (1) or not (0).

Then the second subroutine is introduced, which is used to generate the Hamiltonian matrix (H). First the maximum dimensions of the matrix are defined and then with two do loops the values of each of the elements are defined according to their position in the matrix (ex: i=j hij=alpha).

Once the H matrix has been obtained, we proceed to its diagonalization using a new subroutine. First, the maximum dimensions of the matrices involved are defined, then the elements with the highest value within the matrix (non-diagonal) are found and then, theta is calculated to create the matrix. Then the identity matrix is created to be used later and certain values are substituted in this matrix (I) to end up having the transformation matrix created. Finally and under this order we proceed to perform the iterative computation of matrix containing eigenvectors in columns, the normalization of all eigenvectors and finally, the calculation of normalization constant.
The following matrix corresponds to the organization of the eigenvectors from the copy of the elements belonging to the diagonal of the matrix H in a vector. Then with the use of two do loops the obtained energies are organized in ascending order. The next two subroutines belong to the occupancy calculation and the Mulliken analysis. Finally, we proceed to the subroutine for the calculation of the link order and to the one pertaining to the calculation of the energy from the occupancies previously calculated.

Then, to finalize the code, the whole calculation is organized from the main program, which calls in an orderly way the subroutines previously commented to, finally, write the results belonging to the calculation.