Indian Journal of Chemistry
Vol. 41A, June 2003, pp. 1283-1294

Layer matrices and distance property descriptors

Mircea V Diudea*
Faculty of Chemistry and Chemical Engineering, Babes-Bolyai University, 3400 Cluj, Romania
and
Oleg Ursu
Department of Chemistry, Cleveland State University 2121 Euclid Avenue, Cleveland, Ohio 44115, USA

Received 19 November 2002

Several topological indices–numerical descriptors encoding topological attributes of a molecular graph have been used both in graph discriminating analysis and correlating studies for modeling a variety of physico-chemical properties and biological activities. However, only few software packages, viz., CODESSA, MOLCONN Z, DRAGON, TOSS MODE and POLLY, are available for calculating topological indices. These incorporate correlating analysis statistics, as well. The TOPOCLUJ software package is designed to calculate topological descriptors from topological matrices and/or polynomials. Several weighting schemes including group electronegativity, group mass and partial charges are proposed. Topological indices derived from the matrices like adjacency, connectivity, distance, detour, distance-path, detour-path, Cluj, their reciprocal matrices, walk-matrices, walk-operated matrices, layer- and shell-matrices have been successfully used in correlating studies and graph discriminating analysis during the last decade. Several novel topological matrices, like distance-path, Cluj (with its variants), layer-matrices, walk matrix, walk (triple matrix) operator, characteristic and "property" polynomials, and the corresponding topological descriptors may be calculated by the TOPOCLUJ software package.

Introduction

A sequence of numbers a_1, a_2,..., a_n, representing counting quantities of the vertices 1, 2,…,n in a graph G, can be presented as a vector. Further, if each entry in the above vector is just a vector (of dimension p), we have an array of dimensions $n \times p$, which is a non-square matrix for $p \neq n$. Such arrays are the layer matrices, introduced in the literature[1-5] in connection with the distance/path sequences in the graph[6-8] and have proved to be a strong ground for devising a variety of topological descriptors.

Another way to express a sequence of graph quantities is by a polynomial[9]; the most studied being the characteristic polynomial $Ch(G,x)$, calculated on the adjacency matrix of G (see below). Another polynomial, based on the distance sequence in G is the Hosoya polynomial[9].

In this paper, we present the layer matrices and their relationship with some well-known matrices, polynomials, and topological indices. The main aim of the paper is to illustrate the capabilities of the TOPOCLUJ software package[10] (written in DELPHI 4.0–available on request).

Definitions

Layer matrix LM

A layer of vertices located at distance k to the vertex i is defined as[9,11,12]:

$$G(i)_k = \{v |\ v \in V(G);\ d_{iv} = k\} \quad \ldots (1)$$

Define the partition of G with respect to i as:

$$G(i) = \{G(i)_k;\quad k \in [0,1,..,ecc_i]\} \quad \ldots (2)$$

with ecc_i being the eccentricity of i (*i.e.*, the largest distance from i to the other vertices of G). The entries in the layer matrix (of vertex property) LM, is defined as:

$$[\mathbf{LM}]_{i,k} = \underset{v|d_{i,v}=k}{\Omega} p_v \quad \ldots (3)$$

with the most used operation being the summation. The zero column is just the column of vertex properties $[\mathbf{LM}]_{i,0} = p_i$. Any atomic/vertex property can be considered as p_i. Moreover, any square matrix M can be taken as info matrix, i.e., the matrix supplying local/vertex properties as row sum (RS),

column sum (CS) or diagonal entries given by the Walk matrix[11,12] (see below).

Layer matrix, a collection of the above-defined entries, is defined as

$$\mathbf{LM}=\left\{[\mathbf{LM}]_{i,k};i\in V(G);k\in[0,1,..,d(G)]\right\} \quad \ldots (4)$$

with $d(G)$ being the diameter of the graph (i.e., the largest distance in G).

The layer matrix of vertex labeling **LLb** (see below) is the first layer matrix to be constructed. It represents just the graph partitions with respect to all its vertices.

Shell matrix (SM)

The entries in the shell/layer matrix (of pair vertex property) SM are defined as[13]:

$$[\mathbf{SM}]_{i,k}=\bigcup_{v|d_{i,v}=k}[\mathbf{M}]_{i,v} \quad \ldots (5)$$

with the most used operation being the summation.

Shell matrix is a collection of the above-defined entries and is defined as:

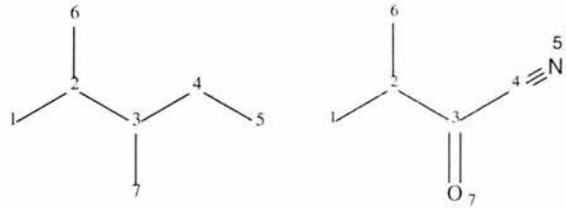
$$\mathbf{SM}=\left\{[\mathbf{SM}]_{i,k};i\in V(G);k\in[0,1,...,d(G)]\right\} \quad \ldots (6)$$

The zero column, $[\mathbf{SM}]_{i,0}=1$, in case of zero diagonal square info matrix but any other vertex property (written as diagonal entries) can be considered. The above definitions hold good in any graph and any square matrix.

Distance extended properties

Any property can be multiplied by distance (topological or genuine one) separating vertices in the graph. The method of achieving this "extension" is different. Examples are given in the next section.

The properties of the above-defined matrices are exemplified on the graphs G_1 and G_2.



G_1 G_2

The topological descriptors were calculated by the TOPOCLUJ software package[10] on the following basic matrices: adjacency, connectivity, distance, 3D-distance, detour, and four types of Cluj matrices.

Properties of layer matrices

Layer matrix LM building

Let us build first the **LLb** matrix (Table 1), by using an amended distance matrix, $\mathbf{DLb}=\mathbf{ILb}+\mathbf{D}_e$, where **I** is the diagonal unity matrix. The partitioning of G_1 with respect to its vertices is obtained by collecting the labeling of vertices located at distance k from the given vertex i (cf (1)). Recall that the non-diagonal entries in the distance matrix $\mathbf{D}_e$ count the number of edges on the shortest path joining two vertices i and j, while the diagonal entries are zero[9].

The most simple and essential is the counting property (i.e., the existence of a vertex in a given position is counted by 1 and absence by zero). The corresponding layer matrix **LC** is given in Table 2. Another example is the layer matrix of partial charges **LCH** (calculated on G_2).

Note that the rows in **LC** represent just the vertex distance degree sequences DDS_i (i.e., the number of vertices located at distance k from i)[9], while the column sums CS represent the graph DDS_k. **LC** is calculated when layer matrix is performed by **TOPOCLUJ** with no property is selected. **LCH** is calculated by clicking on the "atomic charge" in the "Properties" second window.

Table 1—Construction of **LLb** matrix

	DLb(G_1)								**LLb**(G_1)			
	1	2	3	4	5	6	7	k	0	1	2	3	4
1	*1*	1	2	3	4	2	3		1	{2}	{3, 6}	{4, 7}	{5}
2	1	*2*	1	2	3	1	2		2	{1, 3, 6}	{4, 7}	{5}	0
3	2	1	*3*	1	2	2	1		3	{2, 4, 7}	{1, 5, 6}	0	0
4	3	2	1	*4*	1	3	2		4	{3, 5}	{2, 7}	{1, 6}	0
5	4	3	2	1	*5*	4	3		5	{4}	{3}	{2, 7}	{1, 6}
6	2	1	2	3	4	*6*	3		6	{2}	{1, 3}	{4, 7}	{5}
7	3	2	1	2	3	3	*7*		7	{3}	{2, 4}	{1, 5, 6}	0

By multiplying *CS* by *k* (or 1/*k*) distance-extended properties are obtained. In the above example, Wiener[14-16] *W* and Harary[17-19] *H* indices are thus obtained (twice value, in Table 2). Such indices *I* are calculated as the half sum of the entries in a square matrix (i.e., the matrix supplying the property *p*):

$$I=(1/2)\sum_{i<j}[\mathbf{M}]_{i,j}=(1/2)\cdot\mathbf{u}\cdot\mathbf{M}\cdot\mathbf{u}^{\mathbf{T}} \quad \ldots (7)$$

with **u** and $\mathbf{u}^{\mathrm{T}}$ being the unity vector (of dimension *N*, *i.e.*, the number of vertices in *G*) and its transpose, respectively[20]. Reciprocal matrices are calculable by the window "Mathematical operations", and next "Reciprocal".

Within TOPOCLUJ program the partial charges are calculated as follows:

$$ch_{i,j}=\log\left[(S_j/S_i)^{1/(d_{i,j})^2}\right] \quad \ldots (8)$$

$$ch_i=\sum_j ch_{i,j} \quad \ldots (9)$$

In the above relationships, S_i, S_j represent the Sanderson group electronegativities SGEs, calculated for the hydride groups (i.e., the heavy atoms with their surrounding hydrogen atoms) in the molecule. The log function provides the sign for the partial charge ch_{ij}, viewed as a distance decreasing perturbation (see also ref. 21) of the i^{th} SGE produced by the atom j (see the exponent, where d_{ij} is the Euclidean distance separating atoms i and j).

The *N*×*N* array collecting the entries ch_{ij} is the charge matrix **CH**, whose row sums ch_i, represent the total partial charge on hydride group/atom *i* in the molecule (column *k* = 0 in **LCH**-Table 2).

Distance-extended property sum indicates, in this example, the location of the negative partial charge on the more eccentric atoms (*e.g.*, N and O).

LM of symmetric square matrices

The layer matrix of walk degrees $\mathbf{L}^{e}\mathbf{W}$ is illustrated next. The walk degree (of length *e*) equals the row sum in the adjacency matrix **A** (raised at power *e*). For *e*=1 one recovers the classical vertex degree (i.e., vertex valency)[11,12]. Recall that the entries in the adjacency matrix equal unity when two vertices *i* and *j* are connected by an edge and zero otherwise.

The distance extended property is now the valency[22] and the valency-distance index was patterned by several authors[20,23-26] (see footnote, Table 3). In matrix terms, the Cramer product $\mathbf{AD}_e$ is calculated.

The walk matrix **W**, running the $^{e}\mathbf{W}_{\mathrm{M}}$ *algorithm* (patterned by Diudea, Topan and Graovac[12]), allows to elude the raising at a power *e* of a square matrix. It evaluates a (topological) property of a vertex *i*, by iterative summation of the first neighbors contributions. The algorithm, called $^{e}\mathbf{W}_{M}$, is extended here to account for general graphs (with loops and multiple bonds):

$$\mathbf{M}+\mathbf{I}={}^{e}\mathbf{W}_{M};\ e=0 \quad \ldots (10)$$

$$[{}^{e+1}\mathbf{W}_{M}]_{ii}=2L_i[{}^{e}\mathbf{W}_{M}]_{ii}+\sum_{j\neq i}([\mathbf{M}]_{ij}[{}^{e}\mathbf{W}_{M}]_{jj}) \quad \ldots (11)$$

$$[{}^{e}\mathbf{W}_{M}]_{jj}=2L_j[{}^{e-1}\mathbf{W}_{M}]_{jj}+\sum_{k\neq j}([\mathbf{M}]_{jk}[{}^{e-1}\mathbf{W}_{M}]_{kk}) \quad \ldots (12)$$

$$[{}^{e+1}\mathbf{W}_{M}]_{ij}=[{}^{e}\mathbf{W}_{M}]_{ij}=[\mathbf{M}]_{ij} \quad \ldots (13)$$

Table 2—Layer matrix **LC** and layer matrix of partial charges **LCH**, (calculated on G_2)

	LC(G_1)						**LCH**(G_2)					
i \ *k*	0	1	2	3	4	RS	0	1	2	3	4	RS
1	1	1	2	2	1	7	0.087	0.186	0.282	-0.333	-0.223	0
2	1	3	2	1	0	7	0.186	0.370	-0.333	-0.223	0	0
3	1	3	3	0	0	7	0.195	-0.147	-0.048	0	0	0
4	1	2	2	2	0	7	0.015	-0.028	-0.162	0.175	0	0
5	1	1	1	2	2	7	-0.223	0.015	0.195	-0.162	0.175	0
6	1	1	2	2	1	7	0.087	0.186	0.282	-0.333	-0.223	0
7	1	1	2	3	0	7	-0.348	0.195	0.201	-0.048	0	0
CS	7	12	14	12	4		0	0.776	0.418	-0.924	-0.271	0
CS·*k*		12	28	36	16	92[a]		0.776	0.835	-2.771	-1.082	-2.241
CS·1/*k*		12	7	4	1	24[b]		0.776	0.209	-0.308	-0.068	0.610

(a) 2x Wiener index *W*; (b) 2x Harary index *H*

Table 3—Layer matrix of walk degrees $L^e W$

	$\mathbf{L}^1\mathbf{W}$ (G_1)						$\mathbf{L}^2\mathbf{W}$ (G_1)						$\mathbf{L}^3\mathbf{W}$ (G_1)					
$i \backslash k$	0	1	2	3	4	RS	0	1	2	3	4	RS	0	1	2	3	4	RS
1	1	3	4	3	1	12	3	5	9	7	2	26	5	12	17	14	4	52
2	3	5	3	1	0	12	5	12	7	2	0	26	12	22	14	4	0	52
3	3	6	3	0	0	12	6	12	8	0	0	26	12	26	14	0	0	52
4	2	4	4	2	0	12	4	8	8	6	0	26	8	16	18	10	0	52
5	1	2	3	4	2	12	2	4	6	8	6	26	4	8	12	18	10	52
6	1	3	4	3	1	12	3	5	9	7	2	26	5	12	17	14	4	52
7	1	3	5	3	0	12	3	6	9	8	0	26	6	12	20	14	0	52
CS	12	26	26	16	4		26	52	56	38	10		52	108	112	74	18	
CS·k		26	52	48	16	142[a]		52	112	114	40	318		108	224	222	72	626
CS·1/k		26	13	5.33	1	45.33		52	28	12.67	2.5	95.17		108	56	24.67	4.5	193.17

2x*I* (*I* = Degree-Distance Index; Ivanciuc, Dobrynin, Schultz, Estrada - see text)

Table 4—Walk count in the general graph G_2.

	$\mathbf{L}^1\mathbf{W}$ (**CON**(G_2))							$\mathbf{L}^1\mathbf{W}$ (**CON_loop**(G_2))						
1	1	1	0	0	0	0	0	1	1	0	0	0	0	0
2	1	3	1	0	0	1	0	1	3	1	0	0	1	0
3	0	1	4	1	0	0	2	0	1	4	1	0	0	2
4	0	0	1	4	3	0	0	0	0	1	4	3	0	0
5	0	0	0	3	3	0	0	0	0	0	3	5	0	0
6	0	1	0	0	0	1	0	0	1	0	0	0	1	0
7	0	0	2	0	0	0	2	0	0	2	0	0	0	4
	Sum of diagonal elements = 18							Sum of diagonal elements = 22						
	$\mathbf{L}^2\mathbf{W}$ (**CON**(G_2))							$\mathbf{L}^2\mathbf{W}$ (**CON_loop**(G_2))						
1	3	1	0	0	0	0	0	3	1	0	0	0	0	0
2	1	6	1	0	0	1	0	1	6	1	0	0	1	0
3	0	1	11	1	0	0	2	0	1	15	1	0	0	2
4	0	0	1	13	3	0	0	0	0	1	19	3	0	0
5	0	0	0	3	12	0	0	0	0	0	3	22	0	0
6	0	1	0	0	0	3	0	0	1	0	0	0	3	0
7	0	0	2	0	0	0	8	0	0	2	0	0	0	16
	Sum of diagonal elements = 56							Sum of diagonal elements = 84						

where **M** is any square matrix, **I** is the unity diagonal matrix and L_i is the number of loops attached to the atom i.

The algorithm starts with the diagonal entries $[{}^e\mathbf{W}_M]_{ii}$ equaling unity. In each of the following steps, $[{}^e\mathbf{W}_M]_{ii}$ become the row sums RS_i of the matrix **M** raised at a power e, $\mathbf{M}^e$:

$$[{}^e\mathbf{W}_M]_{ii} = \sum_j [\mathbf{M}^e]_{ij} = {}^e w_{M,i} \qquad \ldots (14)$$

These represent walk degrees, ${}^e w_{M,i}$, weighted by the property collected in M. The sum of all diagonal entries in ${}^e\mathbf{W}_M$ is twice the global graph invariant ${}^eW_M$, called molecular walk count:

$$\sum_i {}^e w_{M,i} = 2\,{}^eW_M = 2\,{}^eW_M(G) \qquad \ldots (15)$$

In the above, **M** = **A** or **CON**. When **M** = **D**, then ${}^eW_M$ defines the Wiener number of rank[27] e.

To account for the general graphs the algorithm needs the specification of the atoms bearing loops. It is input as a binary numbers column under the name "aproperty aloop" in the file of properties "name.prp" provided by the TOPOCLUJ program. In this case, the proper matrix is the connectivity matrix **CON**, whose non-diagonal entries are just the conventional bond orders: 0, 1, 2, 3 and 1.5, for non-bonding, single, double, triple and aromatic bonds. If the graph has no loops, the classical ${}^e\mathbf{W}_M$ algorithm is recovered.

Note that, the actual modification is equivalent to an extension of the **CON** matrix by additional rows with entries of value 2 for each loop attached to an atom and empty corresponding columns. The walks involved by the presence of loops are added to the walk counts supplied by the classical algorithm.

Examples are given in Table 4 for G_2, once for no loops and next for loops at atoms 5 and 7 (representing heteroatoms with unshared electrons in the loops). The parameter e is 1 and 2, respectively.

TOPOCLUJ calculates the W matrix on any square matrix.

The next example illustrates the behavior of the classical distance matrix $\mathbf{D}_e$; the property put in column $k = 0$ is the row sum in the distance matrix $RS(\mathbf{D}_e)$ and the corresponding **LM** is called layer of distance sum $\mathbf{LD}_e\mathbf{S}$. Any other square matrix can be manipulated in this manner.

The distance extended property is just the (topological) distance. In matrix terms, it means the Cramer product $\mathbf{D}_e\mathbf{D}_e$ (see footnote b, Table 5). Similar calculations can be performed on the detour matrix Δ_e. Recall that the non-diagonal entries in the detour matrix count the number of edges on the longest path joining two vertices i and j, while the diagonal entries are zero[9,28].

The path-defined distance and detour matrices may be calculated by the "Combinatorial matrix" command). They are defined as follows[9]:

$$[\mathbf{D}_p]_{ij} = \begin{cases} N_{p,(i,j)};(i,j)\in D(G),\ if\ i\neq j \\ 0\ if\ i=j \end{cases} \quad \ldots (16)$$

$$[\mathbf{\Delta}_p]_{ij} = \begin{cases} N_{p,(i,j)};(i,j)\in \Delta(G),\ if\ i\neq j \\ 0\ if\ i=j \end{cases} \quad \ldots (17)$$

$$N_{p,(i,j)} = \binom{[\mathbf{M}_e]_{ij}+1}{2} = (1/2)\left\{([\mathbf{M}_e]_{ij})^2 + [\mathbf{M}_e]_{ij}\right\} \quad \ldots (18)$$

$$\mathbf{M} = \mathbf{D};\mathbf{\Delta}$$

where $N_{p,(i,j)}$ represents the number of all internal paths[29] of length $1 \le |p| \le |(i,j)|$ included in the path (i,j). An example is given in Sect. 4.

LM of unsymmetric square matrices

An unsymmetric matrix, say the product matrix $\mathbf{AD}_e$, can participate either by its *RS* or *CS*, thus providing two different **LM**s (Table 6).

Extension by distance is equivalent to the Cramer multiplication *to the left* by $\mathbf{D}_e$ (of the matrix product $\mathbf{AD}_e$ - see footnote a, Table 5).

The Walk operator[22,27,30] $\mathbf{W}_{(M1,M2,M3)}$, provides matrices, usually unsymmetric, that can also be used in building **LM**. It is defined as[27]:

$$[\mathbf{W}_{(\mathbf{M}_1,\mathbf{M}_2,\mathbf{M}_3)}]_{ij} = {}^{[\mathbf{M}_2]_{ij}} W_{\mathbf{M}_1,i}[\mathbf{M}_3]_{ij} = [RS((\mathbf{M}_1)^{[\mathbf{M}_2]_{ij}})]_i[\mathbf{M}_3]_{ij} \quad \ldots (19)$$

where $W_{M1,i}$ is the walk degree, of elongation $[\mathbf{M}_2]_{ij}$, of the vertex i, weighted by the property collected in matrix $\mathbf{M}_1$ (*i.e.*, the i^{th} row sum of the matrix $\mathbf{M}_1$, raised to power $[\mathbf{M}_2]_{ij}$). The diagonal entries are zero. This matrix, that mixes three square matrices, is a true matrix operator (see below).

Within the $\mathbf{W}_{(M1,M2,M3)}$ matrix operator, the Hadamard algebra operates. In the case of matrix $\mathbf{W}_{(M1,1,M3)}$, (with $\mathbf{M}_2$=1, i.e., the matrix having all non-diagonal entries unity), the meaning of their vector sums is given in Eqs (20) to (22) (see ref. 22). The *CS* recovers the *RS*/*CS* of the corresponding Cramer

Table 5—Distance matrix $\mathbf{D}_e$ and layer of distance sum $\mathbf{LD}_e\mathbf{S}$

	$\mathbf{LD}_e\mathbf{S}(G_1)$							$\mathbf{D}_e(G_1)$							
i\k	0	1	2	3	4	*RS*		1	2	3	4	5	6	7	*RS*
1	15	10	24	26	17	92[a]		0	1	2	3	4	2	3	15
2	10	39	26	17	0	92		1	0	1	2	3	1	2	10
3	9	36	47	0	0	92		2	1	0	1	2	2	1	9
4	12	26	24	30	0	92		3	2	1	0	1	3	2	12
5	17	12	9	24	30	92		4	3	2	1	0	4	3	17
6	15	10	24	26	17	92		2	1	2	3	4	0	3	15
7	14	9	22	47	0	92		3	2	1	2	3	3	0	14
CS	92[a]	142	176	170	64	-	*CS*	15	10	9	12	17	15	14	92[a]
CS·*k*		142	352	510	256	1260[b]									
CS·1/*k*		142	88	56.67	16	302.67									

(a) 2x*W*; (b) $u(D_eD_e)u^T$

Table 6—Product matrix $\mathbf{AD}_e$ and corresponding $\mathbf{LAD}_e$_RS

	$\mathbf{LAD}$_$RS(G_1)$						$\mathbf{AD}(G_1)$							
$i \setminus k$	0	1	2	3	4	RS	1	2	3	4	5	6	7	RS
1	10	39	46	35	12	142	1	0	1	2	3	1	2	10
2	39	56	35	12	0	142	4	3	4	7	10	4	7	39
3	36	74	32	0	0	142	7	4	3	4	7	7	4	36
4	26	48	48	20	0	142	6	4	2	2	2	6	4	26
5	12	26	36	48	20	142	3	2	1	0	1	3	2	12
6	10	39	46	35	12	142	1	0	1	2	3	1	2	10
7	9	36	65	32	0	142	2	1	0	1	2	2	1	9
CS	142	318	308	182	44		*CS* 24	14	12	18	28	24	22	142
CS·*k*		318	616	546	176	1656[a]								
CS·1/*k*		318	154	60.67	11	543.67								

(a) $\mathbf{u}(\mathbf{D}_e(\mathbf{AD}_e)\mathbf{u}^T$

Table 7—$\boldsymbol{W}_{(A,I,De)}$ matrix operator and $\mathbf{LW}_{(A,I,De)}$_RS

	$\mathrm{LW}_{(A,I,De)}$_$RS(G_1)$						$W_{(A,I,De)}(G_1)$							
$i \setminus k$	0	1	2	3	4	RS	1	2	3	4	5	6	7	*RS*[c]
1	15	30	42	38	17	142	0	1	2	3	4	2	3	15
2	30	57	38	17	0	142	3	0	3	6	9	3	6	30
3	27	68	47	0	0	142	6	3	0	3	6	6	3	27
4	24	44	44	30	0	142	6	4	2	0	2	6	4	24
5	17	24	27	44	30	142	4	3	2	1	0	4	3	17
6	15	30	42	38	17	142	2	1	2	3	4	0	3	15
7	14	27	54	47	0	142	3	2	1	2	3	3	0	14
CS	142	280	294	214	64		*CS*[b] 24	14	12	18	28	24	22	142
CS·*k*		280	588	642	256	1766[a]								
CS·1/*k*		280	147	71.33	16	514.33								

(a) $\mathbf{u}(\mathbf{D}_e(\mathbf{W}_{(A,I,De)}))\mathbf{u}^T$; (b) $CS(\mathbf{W}_{(A,I,De)}) = CS(\mathbf{AD}_e)$; (c) $RS(\mathbf{W}_{(A,I,De)}) = RS(\mathbf{A}) \bullet RS(\mathbf{D}_e)$

Table 8—$\boldsymbol{W}_{(De,I,A)}$ matrix operator and $\mathbf{LW}_{(A,I,De)}$_CS

	$\mathbf{LW}_{(A,I,De)}$_$CS(G_1)$						$W_{(De,I,A)}(G_1)$							
$i \setminus k$	0	1	2	3	4	RS	1	2	3	4	5	6	7	RS
1	24	14	36	40	28	142	0	15	0	0	0	0	0	15
2	14	60	40	28	0	142	10	0	10	0	0	10	0	30
3	12	54	76	0	0	142	0	9	0	9	0	0	9	27
4	18	40	36	48	0	142	0	0	12	0	12	0	0	24
5	28	18	12	36	48	142	0	0	0	17	0	0	0	17
6	24	14	36	40	28	142	0	15	0	0	0	0	0	15
7	22	12	32	76	0	142	0	0	14	0	0	0	0	14
CS[c]	142	212	268	268	104		*CS*[b] 10	39	36	26	12	10	9	142
CS·*k*		212	536	804	416	1968[a]								
CS·1/*k*		212	134	89.33	26	461.33								

(a) $\mathbf{u}(\mathbf{D}_e(\mathbf{D}_e\mathbf{A})\mathbf{u}^T$; (b) $CS(\mathbf{W}_{(De,I,A)}) = RS(\mathbf{AD}_e)$; (c) $CS(\mathbf{LW}_{(A,I,De)}) = CS(\mathbf{L}(\mathbf{AD}_e))$

matrix product $\mathbf{M}_1\mathbf{M}_3$ while the *RS* equals the pairwise (Hadamard) product of the two properties collected in $\mathbf{M}_1$ and $\mathbf{M}_2$. The last result has no correspondence in Cramer algebra. Examples are given for $\mathbf{M}_1 = \mathbf{A}$ and $\mathbf{M}_3 = \mathbf{D}_e$ in Tables 7 and 8 (see also footnote a, Table 3).

$$CS(\mathbf{W}_{(M1,1,M3)}) = CS(\mathbf{M}_1\mathbf{M}_3) \quad \ldots (20)$$

$$CS(\mathbf{W}_{(M3,1,M1)}) = RS(\mathbf{M}_1\mathbf{M}_3) \quad \ldots (21)$$

$$RS(\mathbf{W}_{(M1,1,M3)}) = RS(\mathbf{W}_{(M3,1,M1)}) = RS(\mathbf{M}_1) \bullet RS(\mathbf{M}_3) \quad \ldots (22)$$

As a general property of **LM**s, *RS* in any row is constant and equal to *CS* ($k = 0$).

Shell matrices

The novel proposed shell matrix[13] provides a partitioning of the entries in a square matrix according to the vertex (distance) partitions in the graph. It means a true decomposition of the property collected in the square matrix in contributions brought by vertices pertaining to shells located at distance k around each vertex. But the property now depends on the vertex pair relationship.

SM of basic square matrices

Let us consider the behavior of the basic symmetric adjacency A, distance D_e and W_e matrices. Since the first and the last matrices are based on the connectivity relation, the corresponding SM will have only one column, at $k = 1$ (excepting the non-related $k = 0$ column) (Table 9).

In case of $\mathbf{SD}_e$, the distance extended property is again the distance but now the extension is equivalent to the Hadamard product $\mathbf{D}_e \bullet \mathbf{D}_e$. Recall that this kind of matrix product[31]: $[\mathbf{M}_a \bullet \mathbf{M}_b]_{ij} = [\mathbf{M}_a]_{ij}\,[\mathbf{M}_b]_{ij}$, operating within SM matrices, is basically different from the Cramer algebra working in case of **LM** matrices (see above).

The Wiener matrices W_e and W_p are defined only on trees.[32,33] They are calculable as the symmetric Cluj matrices (see the next section).

SM of Cluj matrices

Let now consider the unsymmetric Cluj matrix **UCJ**, defined on the distance concept[34,35] (for other extensions see refs. 36-39). In trees, its non-diagonal entries count the paths going to j through i, the number of which equals that of vertices located closer to i than to j. It is just the cardinality of the set of vertices/atoms obeying such condition; since in cycle-containing graphs more than one path could join i and j (and thus providing different sets $V_{i,p(i,j)_k}$ of vertices, referred to i, with respect to the path $p(i,j)_k$), the entries will be the maximal cardinality value, over all the paths $p(i,j)_k$:

$$V_{i,p(i,j)_k} = \{v \mid v \in V(G);\ d_{iv} < d_{jv};\ p(i,v)_h \cap p(i,j)_k = \{i\};\ \},\ h,k = 1,2,\ldots \quad \ldots (23)$$

$$[\mathbf{UCJ}]_{ij} = \max_{k=1,2,\ldots} \left| V_{i,p(i,j)_k} \right| \quad \ldots (24)$$

The diagonal entries are zero. The Cluj matrices are defined for any connected graph and it is worth noting

Table 9—SM of some basic square matrices

	$\mathbf{SA}(G_l)$						$\mathbf{SD}_e(G_l)$						$\mathbf{SW}_e(G_l)$					
i/k	0	1	2	3	4	$\sum_{k=1}^{d(G)}$	0	1	2	3	4	$\sum_{k=1}^{d(G)}$	0	1	2	3	4	$\sum_{k=1}^{d(G)}$
1	1	1	0	0	0	1	1	1	4	6	4	15	1	6	0	0	0	6
2	1	3	0	0	0	3	1	3	4	3	0	10	1	24	0	0	0	24
3	1	3	0	0	0	3	1	3	6	0	0	9	1	28	0	0	0	28
4	1	2	0	0	0	2	1	2	4	6	0	12	1	16	0	0	0	16
5	1	1	0	0	0	1	1	1	2	6	8	17	1	6	0	0	0	6
6	1	1	0	0	0	1	1	1	4	6	4	15	1	6	0	0	0	6
7	1	1	0	0	0	1	1	1	4	9	0	14	1	6	0	0	0	6
CS	7	12				12[a]	7	12	28	36	16	92[b]	7	92				92[b]
CS·k		12				12		12	56	108	64	240[c]		92				92
CS·1/k		12				12		12	14	12	4	42		92				92

(a) $2\times e$; (b) $2\times W$; (c) $\mathbf{u}(\mathbf{D}_e \bullet \mathbf{D}_e)\mathbf{u}^T$

that they were designed as an extension of the Wiener matrices (see below).

Any other property (*e.g.*, a property specifying the chemical nature of vertices) and any mathematical operation (other than summation) can be considered in view of defining the entries in the Cluj matrices of property[9,39]. We limit here to the cardinality, so that the entries will be integers (see below). The graph-theoretical Cluj matrices may be calculated as "basic matrices" by TOPOCLUJ (Table 10).

The column sums in the shell matrix **SUCJ** represent just the distance-based decomposition of the Cluj property.

The distance-extended property represents Cluj fragments and the involved equivalent matrix operation is **UCJ** • $\mathbf{D}_e$. Note that the Hadamard product is symmetric (*i.e.*, the same matrix product is obtained by operating both to the left and to right side). The extension by the reciprocal distance is important in modeling intra- or intermolecular phenomena, which show decreasing values as distance between the involving partners increases. In trees, the distance extended Cluj fragments are equal to the hyper-Wiener index contributions (see footnote b). Also note that the product **UCJ** • $3\mathbf{DD}_e$ leads to a property extended by the Euclidian distance.

The row sum *RS* and column sum *CS* in UCJ are related to those in the basic matrices as:

$$RS(\mathbf{UCJ}) = RS(\mathbf{W}_e) \quad \ldots (25)$$

$$CS(\mathbf{UCJ}) = CS(\mathbf{D}_e) \quad \ldots (26)$$

The *RS* and *CS* in distance extended matrix **D_UCJ** are related to the above basic matrices, defined for all vertex pairs:

$$RS(\mathbf{D\_UCJ}) = RS(\mathbf{W}_p) \quad \ldots (27)$$

$$CS(\mathbf{D\_UCJ}) = CS(\mathbf{D}_p) \quad \ldots (28)$$

Table 10—Unsymmetric Cluj matrix **UCJ** and corresponding shell matrix **SUCJ**

	SUCJ(G_1)						**UCJ**(G_1)							
i/*k*	0	1	2	3	4	$\sum_{k=1}^{d(G)}$	1	2	3	4	5	6	7	*RS*
1	1	1	2	2	1	6	0	1	1	1	1	1	1	6
2	1	15	6	3	0	24	6	0	3	3	3	6	3	24
3	1	15	13	0	0	28	4	4	0	5	5	4	6	28
4	1	8	4	4	0	16	2	2	2	0	6	2	2	16
5	1	1	1	2	2	6	1	1	1	1	0	1	1	6
6	1	1	2	2	1	6	1	1	1	1	1	0	1	6
7	1	1	2	3	0	6	1	1	1	1	1	1	0	6
CS	7	42	30	16	4	92[a]	*CS* 15	10	9	12	17	15	14	92
CS·*k*		42	60	48	16	166[b]								
CS·1/*k*		42	15	5.33	1	63.33								

(a) 2× Wiener index *W*; (b) 2× hyper-Wiener index *WW*

Table 11—Distance extended matrix **D_UCJ** and $\mathbf{D}_p$.

	D_UCJ (G_1)								$\mathbf{D}_p$ (G_1)							
i	1	2	3	4	5	6	7	*RS*	1	2	3	4	5	6	7	RS
1	0	1	2	3	4	2	3	15	0	1	3	6	10	3	6	29
2	6	0	3	6	9	6	6	36	1	0	1	3	6	1	3	15
3	8	4	0	5	10	8	6	41	3	1	0	1	3	3	1	12
4	6	4	2	0	6	6	4	28	6	3	1	0	1	6	3	20
5	4	3	2	1	0	4	3	17	10	6	3	1	0	10	6	36
6	2	1	2	3	4	0	3	15	3	1	3	6	10	0	6	29
7	3	2	1	2	3	3	0	14	6	3	1	3	6	6	0	25
CS	29	15	12	20	36	29	25	166[a]	*CS* 29	15	12	20	36	29	25	166[a]

(a) 2× hyper-Wiener index *WW*

In Eqs (25) and (27) $\mathbf{W}_e$ and $\mathbf{W}_p$ represent the Wiener matrices defined on edge and path, respectively.

Matrices **D_UCJ** and $\mathbf{D}_p$ are illustrated in Table 11.

Despite the different distribution of numbers (due to the reversed meaning: each entry counts the paths going to i through j), the *CS* in the shell matrix $\mathbf{SUCJ}^T$ of the transpose of the Cluj matrix (Table 12) are identical to those in **SUCJ** matrix (which is an expected result).

Let's now take a look on the symmetric Cluj matrix CJ. In trees, the following relationships hold good:

$$\mathbf{CJ} = \mathbf{CJ}_p = \mathbf{W}_p \quad \ldots (29)$$

$$\mathbf{CJ}_e = \mathbf{W}_e = \mathbf{CJ}_p \bullet \mathbf{A} \quad \ldots (30)$$

The symmetric matrix can be obtained from **UCJ** matrix by the Hadamard product with its transpose:

$$\mathbf{CJ} = \mathbf{UCJ} \bullet \mathbf{UCJ}^T \quad \ldots (31)$$

The CS in the shell matrix **SCJ** represent contributions to the hyper-Wiener global index (see footnote a -Table 13). The corresponding extension by distance is equivalent to $\mathbf{CJ}_p \bullet \mathbf{D}_e$ (and $\mathbf{W}_p \bullet \mathbf{D}_e$). In cycle-containing graphs, no relation exists between the Cluj matrices and Wiener matrices (the last ones being not defined in such graphs).

Note that, in trees, the sum of the distance extended contributions (summation going from $k = 1$ to $k = d(G)$) give a number that is equal to the Tratch number (*i.e.*, the number obtained from the authors' distance extended matrix **E** - see Table 13, footnote b).

The shell matrix of the walk operator $\mathbf{W}_{(A,1,De)}$ is illustrated in Table 14. Relations to the Cramer matrix operations are given as footnotes.

Table 12—Cluj matrix transpose $\mathbf{UCJ}^T$ and corresponding shell matrix $\mathbf{SUCJ}^T$

	$\mathbf{S(UCJ)}^T,(G_l)$						$(\mathbf{UCJ})^T\,(G_l)$							
i/k	0	1	2	3	4	$\sum_{k=1}^{d(G)}$	1	2	3	4	5	6	7	*RS*
1	1	6	5	3	1	15	0	6	4	2	1	1	1	15
2	1	6	3	1	0	10	1	0	4	2	1	1	1	10
3	1	6	3	0	0	9	1	3	0	2	1	1	1	9
4	1	6	4	2	0	12	1	3	5	0	1	1	1	12
5	1	6	5	4	2	17	1	3	5	6	0	1	1	17
6	1	6	5	3	1	15	1	6	4	2	1	0	1	15
7	1	6	5	3	0	14	1	3	6	2	1	1	0	14
CS	7	42	30	16	4	92[a]	*CS* 6	24	28	16	6	6	6	92[a]
CS·k		42	60	48	16	166[b]								
CS·1/k		42	15	5.33	1	63.33								

(a) 2×*W*; (b) 2×*WW*

Table 13—Cluj symmetric matrix **CJ** and corresponding shell matrix **SCJ**

	$\mathbf{SCJ}(G_l)$						$\mathbf{CJ}(G_l)$							
i/k	0	1	2	3	4	$\sum_{k=1}^{d(G)}$	1	2	3	4	5	6	7	*RS*
1	1	6	5	3	1	15	0	6	4	2	1	1	1	15
2	1	24	9	3	0	36	6	0	12	6	3	6	3	36
3	1	28	13	0	0	41	4	12	0	10	5	4	6	41
4	1	16	8	4	0	28	2	6	10	0	6	2	2	28
5	1	6	5	4	2	17	1	3	5	6	0	1	1	17
6	1	6	5	3	1	15	1	6	4	2	1	0	1	15
7	1	6	5	3	0	14	1	3	6	2	1	1	0	14
CS	7	92	50	20	4	166[a]	*CS* 15	36	41	28	17	15	14	166
CS·k		92	100	60	16	268[b]								
CS·1/k		92	25	6.67	1	124.67								

(a) 2× hyper Wiener index *WW*; (b) 2× Tratch index *Tr*

Table 14—Shell matrix of the walk operator $\mathbf{W}_{(A,I,De)}$

	$\mathbf{SW}_{(A,I,De)}(G_1)$							$\mathbf{W}_{(A,I,De)}(G_1)$							
i/k	0	1	2	3	4	$\sum_{k=1}^{d(G)}$		1	2	3	4	5	6	7	*RS*
1	1	1	4	6	4	15		0	1	2	3	4	2	3	15
2	1	9	12	9	0	30		3	0	3	6	9	3	6	30
3	1	9	18	0	0	27		6	3	0	3	6	6	3	27
4	1	4	8	12	0	24		6	4	2	0	2	6	4	24
5	1	1	2	6	8	17		4	3	2	1	0	4	3	17
6	1	1	4	6	4	15		2	1	2	3	4	0	3	15
7	1	1	4	9	0	14		3	2	1	2	3	3	0	14
CS	7	26	52	48	16	142[a]	*CS*	24	14	12	18	28	24	22	142
CS·k		26	104	144	64	338[b]									
CS·1/k		26	26	16	4	72[c]									

(a) $\mathbf{u}(\mathbf{AD}_e)\mathbf{u}^T$; (b) $\mathbf{u}(\mathbf{A}(\mathbf{D}_e \bullet \mathbf{D}_e))\mathbf{u}^T$; (c) $\mathbf{u}(\mathbf{A}(\mathbf{D}_e \bullet \mathbf{RD}_e))\mathbf{u}^T$

As a general property of **SM**s, *RS* vector equals *RS* in the info matrix **M**.

Indices calculated on the layer matrices

Layer matrices are used to derive two topological indices: (*i*) indices of *centrality* $C(LM)$ and (*ii*) indices of *centrocomplexity* $X(LM)$.

Indices of centrality

Indices of centrality $C(LM)$ look for the center of a graph and are defined as

$$C(LM)_i = \left[\sum_{k=1}^{ecc} \left([\mathbf{LM}]_{ik}^{2k} \right)^{1/(ecc)^2} \right]^{-1} \quad \ldots (32)$$

$$C(LM) = w \sum_i C(LM)_i \quad \ldots (33)$$

where *ecc* is the maximal distance in G (*i.e.*, max $d(i,k)$) and w is a weighting factor.

Indices of centrocomplexity

These indices express the location versus a vertex of high complexity (*e.g.*, a vertex of high degree or a heteroatom). They are defined as:

$$X(LM)_i = \sum_{k=0}^{ecc} [\mathbf{LM}]_{ik} \, 10^{-k} \quad \ldots (34)$$

$$X(LM) = w \sum_i X(LM)_i \quad \ldots (35)$$

An example of index calculation is given in Table 15, for the layer matrix LCON (with p_i being the row sum *RS* in the connectivity matrix CON and $w = 1$).

Table 15—Index calculation for the layer matrix **LCON**

	LCON_$RS(G_2)$; $w = 1$						
i/k	0	1	2	3	4	C_i	X_i
1	1	3	5	6	3	0.1579	1.3563
2	3	6	6	3	0	0.2312	3.6630
3	4	9	5	0	0	0.3557	4.9500
4	4	7	5	2	0	0.2458	4.7520
5	3	4	4	5	2	0.1711	3.4452
6	1	3	5	6	3	0.1579	1.3563
7	2	4	7	5	0	0.2153	2.4750
						1.5349	21.9978

Polynomials calculated on the layer matrices

The characteristic polynomial of a matrix is defined[9,40] by the relation:

$$Ch(G, M, x) = \det[x\mathbf{I} - \mathbf{M}(G)] \quad \ldots (36)$$

with I being the unit matrix of a pertinent order and **M** is a square matrix. The polynomial roots are just the eigenvalues of the matrix **M**. The string of the decreasing values of the eigenvalues is called the spectrum of the respective matrix.

The coefficients a_k of the characteristic polynomial (**M** = **A**, in which case the matrix symbol is omitted) of order N are calculable from the graph G on N vertices via:

$$Ch(G,x) = \sum_{k=0}^{N} a_k(G) \cdot x^{N-k} \quad \ldots (37)$$

More efficient are the numerical methods of linear algebra, such as the well-known recursive algorithms of Le Verier, Frame, or Fadeev[40-42].

The polynomial description of a molecular graph was used in the early Hűckel theory, the eigenvalues of the Hűckel matrix (a weighted, normalized adjacency matrix) evaluating the π-electron energy levels of the molecular orbitals in conjugated hydrocarbons. Other related topics using such description are Topological Resonance Energy TRE, Topological Effect on Molecular Orbitals, TEMO, the Aromatic Sextet Theory, AST, the Kekűle Structure Count, KSC, etc[9, 40].

A different polynomial description of a graph is based on finite sequences[9] of some graph invariants, such as the distance degree sequence or the sequence of the number of k-independent edge sets. The polynomial corresponding to the latter sequence was introduced by Hosoya and named the Z-counting polynomial[43].

A distance property polynomial was defined[13] as:

$$P(G,M,p,x)=\sum_{k=0}^{d(G)} p(G,M,k)\cdot x^k \qquad \ldots (38)$$

with $p(G,M,0)=P(G)=\sum_i p_i$ and p_i being a vertex property. In relation (3), $p(G,M,k)$ is twice the contribution to the global (molecular) property $P(G)$ of the vertex pairs located at distance k to each other, in the graph G. The summation runs from zero to $d(G)$, which is the *diameter* of G or the longest distance in G.

When the local property $p_i = 1$ (*i.e.*, the vertex count), $p(G, k)$ denotes the number of pair vertices separated by distance k in G, and the classical Hosoya polynomial[44] (more exactly twice this polynomial) is recovered. In this case, $p(G,0)=N$, where N stands for the number of vertices in the hydrogen depleted molecular graph.

The polynomial coefficients $p(G,M,k)$ may be calculated as the column sums in the layer matrices **LM** and **SM**. Thus, $P(G,M,p,x)$ is written as $P(G,LM,p,x)$ or $P(G,SM,p,x)$. For example, $P(G,SUCJ,ch,x)$ reads: the polynomial of the shell of unsymmetric Cluj matrix, calculated by partial charges. When p_i is 1, it can be omitted.

A distance-extended property can be calculated by evaluating the first derivative of the polynomial, for $x = 1$:

$$P'(G,M,p,1)=\sum_{k=1}^{d(G)} k\cdot p(G,M,k)=D\_P(G) \qquad \ldots (39)$$

In the case of $p_i = 1$, $P'(G,LC,1)$ is just the Wiener index. The property p_i can be taken either as a crude property (*i.e.*, the column zero in **LM**) or within some weighting scheme (*i.e.*, transformed by the sequence: **W**-operator $\mathbf{W}_{(M1,M2,M3)}$, **W(M)** matrix, **LM/SM**), as shown above. Any square matrix can be used as an info matrix for the layer matrices, thus resulting an unlimited number of property polynomials.

Several global descriptors can be derived from the above polynomials:

1. The sum of absolute values of the polynomial coefficients (see Hosoya's Z-counting polynomial[43] and also[45]):

$$SumP(G,x)=\sum_k |a_k(G,x)| \qquad \ldots (40)$$

Table 16—Distance Matrix D, Shell of Distance Matrix **SD**, Polynomials, roots, $MaxSpP(G,x)$ and $MinSpP(G,x)$ and $SumP(G,x)$ (in bold)

D_e										SD_e					
								x_i	k	0	1	2	3	4	Real x_i
1	0	1	2	3	4	2	3	**13.635**		1	1	4	6	4	**-0.044**
2	1	0	1	2	3	1	2	-0.432		1	3	4	3	0	-0.044
3	2	1	0	1	2	2	1	-0.665		1	3	6	0	0	-0.813
4	3	2	1	0	1	3	2	-1.309		1	2	4	6	0	**-0.813**
5	4	3	2	1	0	4	3	-2.000		1	1	2	6	8	
6	2	1	2	3	4	0	3	-3.006		1	1	4	6	4	
7	3	2	1	2	3	3	0	**-6.223**		1	1	4	9	0	
	$SumP(G,x)$							**6041**		7	12	28	36	16	**99**

$Ch(G,D,x)=x^7-120x^5-752x^4-1840x^3-2080x^2-1056x-192$

$P(G,SD,x)=16x^4+36x^3+28x^2+12x+7$

2. The maximal and minimal values of a spectrum (see the first eigenvalue of **A**, proposed by Cvetkovi□ & Gutman[46] for characterizing the branching of a graph, and also[47]):

$$MaxSpP(G,x)\,;\ MinSpP(G,x) \qquad \ldots(41)$$

Examples are given in Table 16.

Acknowledgement

This study was supported by the Romanian Grants 75/25 and 113/37, 2002.

References

1 Skorobogatov V A & Dobrynin A A, *MATCH - Commun math Comput Chem*, 23 (1988) 105.
2 Diudea M V, Minailiuc O M & Balaban A T, *J comput Chem*, 12 (1991) 527.
3 Dobrynin A A, *J math Chem*, 14 (1993) 175.
4 Yang Y, Lin J & Wang C, *J Graph Theory*, 39 (2002) 219.
5 Hu C Y & Xu L, *J chem Inf Comput Sci*, 36 (1996) 82.
6 Bonchev D, Mekanyan O & Balaban A T, *J chem Inf Comput Sci*, 29 (1989) 91.
7 Halberstam F Y & Quintas L V, *Distance and path degree sequences for cubic graphs* (Pace Univ. New York), 1982.
8 Randić M, Woodworth W L & Graovac A, *Int J quantum Chem*, 24 (1983) 435.
9 Diudea M V, Gutman I & Jäntschi L, *Molecular topology*, (NOVA, Huntington, New York), 2001.
10 Diudea M V & Ursu O, *TOPOCLUJ* (Copyright Babes-Bolyai Univ. Cluj), 2002.
11 Diudea M V, *J chem Inf Comput Sci*, 34 (1994) 1064.
12 Diudea M V, Topan M I & Graovac A, *J chem Inf Comput Sci*, 34 (1994) 1072.
13 Diudea M V, *MATCH - Commun math Comput Chem (MATCH)* (In press).
14 Wiener H, *J Am chem Soc*, 69 (1947) 17.
15 Diudea M V & Gutman I, *Croat chem Acta*, 71 (1998) 21.
16 Diudea M V, *J chem Inf Comput Sci*, 36 (1996) 833.
17 Ivanciuc O, Balaban T S & Balaban A T, *J math Chem*, 12 (1993) 309.
18 Plavšić D, Nikolić S, Trinajstić N & Mihalić Z, *J math Chem*, 12 (1993) 235.
19 Diudea M V, *J chem Inf Comput Sci*, 37 (1997) 292.
20 Estrada E, Rodriguez L & Gutierrez A, *MATCH - Commun math Comput Chem*, 35 (1997) 145.
21 Kier L B & Hall L H, *Molecular structure description: the electrotopological state*, (Academic Press, San Diego), 1999.
22 Diudea M V, *Croat chem Acta*, 72 (1999) 835.
23 Schultz H P, *J chem Inf Comput Sci*, 29 (1989) 227.
24 Ivanciuc O, *Rev Roum Chem*, 45 (2000) 587.
25 Dobrynin A & Kochetova A A, *J chem Inf Comput Sci*, 34 (1994) 1082.
26 Estrada E & Rodriguez L, *MATCH - Commun math Comput Chem*, 35 (1997) 157.
27 Diudea M V, *J chem Inf Comput Sci*, 36 (1996) 535.
28 Harary F, *Graph theory* (Addison-Wesley, Reading, M.A.), 1969.
29 Klein D J, Lukovits I & Gutman I, *J chem Inf Comput Sci*, 35 (1995) 50.
30 Diudea M V & Randić M, *J chem Inf Comput Sci*, 37 (1997) 1095.
31 Horn R A & Johnson C R, *Matrix analysis* (Cambridge Univ. Press, Cambridge), 1985.
32 Randić M, Guo X, Oxley T & Krishnapriyan H, *J chem Inf Comput Sci*, 33 (1993) 700.
33 Randić M, Guo X, Oxley T, Krishnapriyan H & L Naylor, *J chem Inf Comput Sci*, 34 (1994) 361.
34 Diudea M V, *J chem Inf Comput Sci*, 37 (1997) 300.
35 Diudea M V, *MATCH - Commun math Comput Chem*, 35 (1997) 169.
36 Diudea M V, Parv B & Gutman I, *J chem Inf Comput Sci*, 37 (1997) 1101.
37 Gutman I & Diudea M V, *J Serb chem Soc*, 63 (1998) 497.
38 Diudea M V, Katona G, Lukovits I & Trinajstić N, *Croat chem Acta*, 71 (1998) 459.
39 Jäntschi L, Katona G & Diudea M V, *MATCH - Commun math Comput Chem*, 41 (2000) 151.
40 Diudea M V & Ivanciuc O, *Molecular topology*, (COMPREX, Cluj), 1995.
41 Dwyes P S, *Linear computations*, (Wiley, New York) 1951.
42 Fadeev D K & Sominskii I S, *Problems in higher algebra*, (Freeman, San Francisco), 1965.
43 Hosoya H, *Bull chem Soc Japan*, 44 (1971) 2332.
44 Hosoya H, *Discrete appl Math*, 19 (1988) 239.
45 Ivanciuc O, *Rev Roum Chem*, 46 (2001) 129.
46 Cvetković D M & Gutman I, *Croat chem Acta, 49* (1977) *115.*
47 Diudea M V, Ivanciuć O, Nikolić S & Trinajstić N, *MATCH - Commun math Comput Chem*, 35 (1997) 41.
48 Randić M, *J chem Inf Comput Sci*, 41 (2001) 627.
49 Katritzky A R, Lobanov V S & Karelson M, *CODESSA reference manual*, (Ver. 2.0, Univ. Florida), 1996.
50 *MOLCONN-Z*, Hall associates consulting, (Quincy, MA, 02170, USA).
51 Todeschini R & Consonni V, *Handbook of molecular descriptors* (Wiley-VCH), 2000, Dragon 1.1 software package.
52 Gutierrez Y, Estrada E, Rodriguez L & Munoz I, *TOSS_MODE*, ver. 2.5 for Window, (Central Univ. Las Villas, Santa Clara, Cuba), 1995.
53 Basak S C, Harris D K & Magnuson V R, *POLLY* 2.3, (Copyright Univ. Minnesota), 1988.