APPLICATIONS OF THE MATRIX PACKAGE MATLAB IN COMPUTING THE
HOSEYA POLYNOMIAL OF Zig–Zag NANOTUBES

HOSSEIN SHABANI, ALI REZA ASHRAFI*

Institute of Nanoscience and Nanotechnology, University of Kashan, Kashan 87317-51167, Iran

The Hosoya polynomial of a molecular graph G is defined as

\[ W(G, x) = \sum_{\{u,v\} \subseteq V(G)} x^{d(u,v)} \]

where the sum is over all unordered pairs \{u,v\} of distinct vertices in G. In this paper an algorithm for computing the Hosoya polynomial of zig-zag nanotubes are given.

(Received October 15, 2009; accepted November 23, 2009)

Keywords: Zig-zag nanotube, Hosoya polynomial.

1. Introduction

Nanostructured materials have received a lot of attention because of their novel properties. Nanotubes are an important category of one-dimensional nanostructured materials can be prepared from carbon.1

A topological index is a real number that is derived from molecular graphs of chemical compounds. Such numbers based on the distances in a graph are widely used for establishing relationships between the structure of molecules and their physico-chemical properties. The distance between atoms of a molecular graph is the length of a minimal path connecting them. The Wiener index was the first distance based topological index introduced early by Harold Wiener.2 It is defined as the sum of distances between any two carbon atoms in the molecules, in terms of carbon-carbon bonds. We encourage the reader to consult papers3,4 and references therein, for further study on the topic.

We now recall some algebraic notations that will be used in the paper. Suppose G is a graph and e is an edge of G. If e connects the vertices u and v then we write e = uv. Let d(u,v) denote the distance between vertices u and v in G. The Hosoya polynomial of G is defined as

\[ W(G, x) = \sum_{\{u,v\} \subseteq V(G)} x^{d(u,v)} \]

where the sum is over all unordered pairs \{u,v\} of distinct vertices in G.5 Suppose V(G) = \{v_1, ..., v_n\} and M = [d_{ij}] denotes the distance matrix of G, where d_{ij} = d(v_i,v_j). Then one can see that W(G) = 1/2 \sum_i d_{ii} and W(G,x) = 1/2 \sum_i x^{d_{ii}}.

The problem of computing topological indices of nanostructures is introduced firstly by Diudea and his co-authors.6-11 He takes the armchair, zig-zag and TUC_4C_8(R/S) nanotubes into consideration and computed the Wiener index of these nanomaterials. The second author of this paper computed the Wiener index of a polyhex and TUC_4C_8(R/S) nanotubes and nanotori.12-18 In this paper we continue this program to compute the Hosoya polynomial of these nanomaterials. Our notation is standard and mainly taken from the book of Trinajstic19 and papers by Taeri and his co-authors [20-22]

* Corresponding author. (ashrafi@kashanu.ac.ir)
2. Main results

In this section we derive an exact formula for the Hosoya polynomial of zig–zag polyhex nanotubes, Figures 1–4. Since \( \frac{d}{dx}(W(G,x))|_{x=4} = W(G) \), the Wiener index of these nanomaterials are also computed.

Choose two base vertices \( v(1,1) \) and \( u(1,1) \) from the 2–dimensional lattice of \( T = TUV[m,n] \), Figure 2, where \( m \) is the number of rows and \( n \) is the number of zig-zags, Figure 3. Assume that \( D_u(1,1) \) is distance between vertex \( u(1,1) \) and all vertices of \( T \). This defines two matrix for the base vertices denoted by \( D_u(1,1) = [d_{ij}^{u(1,1)}] \) and \( D_v(1,1) = [d_{ij}^{v(1,1)}] \). Define three matrices \( A_{m\times(n/2)}^{u(1,1)} = [a_{ij}] \), \( A_{m\times(n/2+1)}^{v(1,1)} = [c_{ij}] \) and \( B_{(n/2+1)\times n} = [b_{ij}] \) as follows:

<table>
<thead>
<tr>
<th>( i = 1 )</th>
<th>( j = 1 )</th>
<th>( j = 2 )</th>
<th>( i &gt; 1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_{11} = 0 )</td>
<td>( a_{12} = 1 )</td>
<td>( )</td>
<td>( a_{ij} = a_{i1} ) when ( j ) is odd</td>
</tr>
<tr>
<td>( a_{i1} = a_{i1} + 1 )</td>
<td>( a_{i2} = a_{i1} + 1 )</td>
<td>( )</td>
<td>( a_{ij} = a_{i2} ) when ( j ) is even</td>
</tr>
<tr>
<td>( c_{11} = 0 )</td>
<td>( c_{12} = 1 )</td>
<td>( )</td>
<td>( c_{ij} = c_{i1} ) when ( j ) is odd</td>
</tr>
<tr>
<td>( c_{i1} = c_{i1} + 1 )</td>
<td>( c_{i2} = c_{i1} + 1 )</td>
<td>( )</td>
<td>( c_{ij} = c_{i2} ) when ( j ) is even</td>
</tr>
</tbody>
</table>

We now define a new matrix \( B_{m\times(n/2+1)} \), as \( b_{1j} = i-1, 1 \leq i \leq m \), and \( b_{ij} = b_{(i-1)j} + 1 \) for other entries. From these matrices, one can compute matrices \( D_{u(1,1)} \) and \( D_{v(1,1)} \) as follows:

\[
\begin{align*}
    d_{ij}^{u(1,1)} &= \max(a_{ij}, b_{ij}) & 1 \leq j \leq n/2 + 1 \\
    d_{ij}^{v(1,1)} &= \max(a_{ij}, c_{ij}) & 1 \leq j \leq n/2 + 1 \\
    d_{ij}^{u(1,1)} &= d_{(n-j+2)}^{u(1,1)} & j > n/2 + 1 \\
    d_{ij}^{v(1,1)} &= d_{(n-j+2)}^{v(1,1)} & j > n/2 + 1 
\end{align*}
\]

Set \( D_{u(1,1)} = [\Delta_i^{u(1,1)}]_{1 \leq i \leq m} \) and \( D_{v(1,1)} = [\Delta_i^{v(1,1)}]_{1 \leq i \leq m} \), such that \( \Delta_i \) denotes the \( i \)th row of the matrix. We also assume that the first row of \( D_{u(1,1)} \) and \( D_{v(1,1)} \) are as follows:

\[
\begin{align*}
    [d_{u(1,1)}^{u(1,1)}, d_{u(1,2)}^{u(1,1)}, d_{u(1,3)}^{u(1,1)}, \ldots, d_{u(n/2)}^{u(1,1)}, d_{u(n/2+1)}^{u(1,1)}], \\
    [d_{v(1,1)}^{v(1,1)}, d_{v(1,2)}^{v(1,1)}, d_{v(1,3)}^{v(1,1)}, \ldots, d_{v(n/2)}^{v(1,1)}, d_{v(n/2+1)}^{v(1,1)}].
\end{align*}
\]

Suppose \( D_{u(i,1)} \) and \( D_{v(i,1)} \) are distance matrices associated to the \( i \)th row of \( T \). Then,
Notice that matrices associated to other columns of $T$ are the same as the first one, Figure 3. Furthermore, two matrices $D_{u(i,j)}$ and $D_{v(i,j)}$ are obtained by replacement of the columns of $D_{u(i,1)}$ and $D_{v(i,1)}$, respectively. To compute the Hosoya polynomial of $T$, it is enough to count the equal entries of the distance matrix of $T$. But, the entries of the $i^{th}$ row of these matrices ($1 < i \leq m$) are appear $2n(m - i + 1)$ times, and the entries of the first row are appear $nm$ times. Suppose $D_{u(i,j)} = [d_{ij}^{u(1,1)}]$ and $D_{v(i,j)} = [d_{ij}^{v(1,1)}]$. So, we achieve two polynomials for vertices $u$ and $v$, as follows:

$$W_u(T, x) = \frac{1}{2} n \left[ \left( m \sum_{j=1}^{n} x^{d_{ij}^{u(1,1)}} \right) + \left( \sum_{i=2}^{m} 2(m - i + 1) \left( \sum_{j=1}^{n} x^{d_{ij}^{u(1,1)}} \right) \right) \right]$$

$$W_v(T, x) = \frac{1}{2} n \left[ \left( m \sum_{j=1}^{n} x^{d_{ij}^{v(1,1)}} \right) + \left( \sum_{i=2}^{m} 2(m - i + 1) \left( \sum_{j=1}^{n} x^{d_{ij}^{v(1,1)}} \right) \right) \right]$$

Therefore the Hosoya polynomial of $T$ is as follows:

$$W(T, x) = W_u(T, x) + W_v(T, x)$$
A MATLAB Program for Computing Hosoya Polynomial of Zig-Zag Nanotubes

reply = input('Enter number rows of graph zig-zag (m): ');
m=reply;
reply = input('Enter number columns of graph zig-zag (n): ');
n=reply;

%Au=[a_ij]
au=zeros(m,n/2+1);
au(1,1)=0;
au(1,2)=1;
for i=2:m
    if mod(i,2)==0
        au(i,1)=au(i-1,1)+1;
        au(i,2)=au(i,1)+1;
    else
        au(i,2)=au(i-1,2)+1;
        au(i,1)=au(i,2)+1;
    end
end
for j=3:n/2+1
    for i=1:m
        if mod(j,2)==0
            au(i,j)=au(i,1);
        else
            au(i,j)=au(i,2);
        end
    end
end

%Av=[c_ij]
av=zeros(m,n/2+1);
av(1,1)=0;
nav(1,2)=1;
for i=2:m
    if mod(i,2)==0
        av(i,2)=av(i-1,2)+1;
        av(i,1)=av(i,2)+1;
    else
        av(i,1)=av(i-1,1)+1;
        av(i,2)=av(i,1)+1;
    end
end
for j=3:n/2+1
    for i=1:m
        if mod(j,2)==0
            av(i,j)=av(i,1);
        else
            av(i,j)=av(i,2);
        end
    end
end
%B=[b_ij]
bz=zeros(m,n/2+1);
for i=1:m
    bz(i,1)=i-1;
end
for j=2:n/2+1
    for i=1:m
        bz(i,j)=bz(i,j-1)+1;
    end
end

%Du=[du_ij]
du=zeros(m,n);
for j=1:n/2+1
    for i=1:m
        du(i,j)=max{au(i,j),bz(i,j)};
    end
end
for j=n/2+2:n
    for i=1:m
        du(i,j)=du(i,n-j+2);
    end
end

%Dv=[dv_ij]
dv=zeros(m,n);
for j=1:n/2+1
    for i=1:m
        dv(i,j)=max{av(i,j),bz(i,j)};
    end
end
for j=n/2+2:n
    for i=1:m
        dv(i,j)=dv(i,n-j+2);
    end
end

%in matrix eu we have
eu=zeros(m,max(max(du))+1);
for i=1:m
    for j=1:n
        eu(i,du(i,j)+1)=eu(i,du(i,j)+1)+1;
    end
end

ku=[m,(2*m-2):-2:2];
kku=zeros(m,max(max(du)))+1;
for j=1:max(max(du))
    kku(:,j)=ku';
end;

hu=zeros(m,max(max(du)))+1;
lu=[m,(2*m-2):-2:2];
for i=1:m
    hu(i,:)=1/2*n*(lu(i)+hu(i,:));
end

%in matrix ev we have
ev=zeros(m,max(max(dv)))+1;
for i=1:m
    for j=1:n
        ev(i,dv(i,j)+1)=ev(i,dv(i,j)+1)+1;
    end
end

kv=[m, (2*m-2):-2:2];
kkv=zeros(m,max(max(dv)))+1);
for j=1:max(max(dv))
    kv(:,j)=kv';
end;
hv=zeros(m,max(max(dv)))+1,
lv=[m, (2*m-2):-2:2];
for i=1:m
    hv(i,:)=1/2*n*(lv(i)*ev(i,:));
end
    cv=hv(l,:);
for i=2:m
    cv=hv(i,:)+cv;
end
W(T)=zeros();
W(T)=cv+cv;

References