A UNIFIED METHOD FOR EIGENDECOMPOSITION OF
GRAPH PRODUCTS

A. Kaveh* and H. Rahami
Department of Civil Engineering, Iran University of Science and Technology,
Tehran-16, Iran

ABSTRACT
In this paper, a unified method is developed for calculating the eigenvalues of the weighted
adjacency and Laplacian matrices of three different graph products. These products have
many applications in computational mechanics, such as ordering, graph partitioning, and
subdomaining of finite element models.

Keywords: adjacency, Laplacian, Cartesian product, strong Cartesian product, direct
product, eigendecomposition, regular graph, graph product

1. INTRODUCTION
Graph theory has a long history, and its applications in structural mechanics and in particular
nodal ordering and graph partitioning are well documented in the literature, Kaveh [1-2].

Algebraic graph theory can be considered as a branch of graph theory, where eigenvalues
and eigenvectors of certain matrices are employed to deduce the principal properties of a
graph. In fact eigenvalues are closely related to most of the invariants of a graph, linking one
extremal property to another. These eigenvalues play a central role in our fundamental
understanding of graphs. There are interesting books on algebraic graph theory such as
Biggs [3], Cvetković et al. [4], and Godsil and Royle [5].

One of the major contributions in algebraic graph theory is due to Fiedler [6], where the
properties of the second eigenvalue and eigenvector of the Laplacian of a graph have been
introduced. This eigenvector, known as the Fiedler vector is used in graph nodal ordering
and bipartition, Refs. [7-9].

General methods are available in the literature for calculating the eigenvalues of matrices,
however, for matrices corresponding to special models, it is beneficial to make use of their
extra properties.

In this paper, a unified approach is developed for calculating the eigenvalues of the
adjacency and Laplacian matrices of three different graph products. These methods have

* Email-address of the corresponding author: alikaveh@iust.ac.ir
many applications in computational mechanics, such as ordering, graph partitioning, and subdomaining finite element models, Kaveh and Rahami [10,11].

2. DEFINITIONS

2.1 Definitions from Graph Theory
A graph $G(N,E)$ consists of a set of elements, $N(G)$, called nodes and a set of elements, $E(G)$, called edges, together with a relation of incidence which associates two distinct nodes with each edge, known as its ends. Two nodes of a graph are called adjacent if these nodes are the end nodes of an edge. An edge is called incident with a node if it is an end node of the edge. The degree of a node is the number of edges incident with the node. A subgraph $G_i$ of a graph $G$ is a graph for which $N(G_i) \subseteq N(G)$ and $E(G_i) \subseteq E(G)$, and each edge of $G_i$ has the same ends as in $G$. A path of $G$ is a finite sequence $P_i = \{n_0, m_1, n_1, ..., m_p, n_p\}$ whose terms are alternately distinct nodes $n_i$ and distinct members $m_i$ of $G$ for $1 \leq i \leq p$, and $n_{i-1}$ and $n_i$ are the two ends of $m_i$. A cycle is a path $(n_0, m_1, n_1, ..., m_p, n_p)$ for which $n_0 = n_p$ and $p \geq 3$; i.e. a cycle is a closed path. A cycle graph with $n$ nodes is denoted as $C_n$.

2.2 Eigenvalues and eigenvectors of matrix $A$
Consider a graph with weights assigned to its nodes and edges. The nodal weight vector is,

$$NW = [n_{wi}]; \ i = 1,2,...,n,$$

and edge weight vector is defined as:

$$EW = [ew_{ij}]; \ (i,j) = 1,..,n,$$

The adjacency matrix $A = [a_{ij}]_{n \times n}$ of a weighted graph $G$, containing $n$ nodes, is defined as:

$$a_{ij} = \begin{cases} ew_{ij} & \text{if } n_i \text{ is adjacent to } n_j \\ 0 & \text{otherwise} \end{cases}$$

For a non-weighted graph $ew_{ij}$ should be replaced by unity.
Consider the eigenproblem as

$$A \phi_i = \mu_i \phi_i$$

where $\mu_i$ is the eigenvalue and $\phi_i$ is the corresponding eigenvector. Since $A$ is a symmetric real matrix, all it’s eigenvalues are real and can be expressed as

$$\mu_1 \leq \mu_2 \leq \ldots \leq \mu_{n-1} \leq \mu_n$$
The largest eigenvalue \( \mu_n \) is the root of the characteristic equation of \( A \) with multiplicity 1. The corresponding eigenvector \( \phi_n \) is the only eigenvector with positive entries. This vector has attractive properties employed in geography and structural mechanics.

Gould [12] appears to have introduced the first important application on using the properties of \( \phi_n \) in calculating the accessibility indices of cities. The city with the highest accessibility corresponds to the largest entry of \( \phi_n \).

Grimes et al. [13] used the node with smallest accessibility as a pseudo-peripheral node corresponding to the node with least entry of \( \phi_n \). Kaveh [14] used the properties of \( \phi_n \) for complete nodal ordering.

2.3 Eigenvalues and eigenvectors of matrix \( L \)
The entries of the weighted Laplacian matrix \( L \) of a weighted graph is defined as:

\[
L = D - A, \tag{6}
\]

The entries of \( L \) are as follows:

\[
l_{ij} = \begin{cases} 
-\text{ew}_{ij} = -\text{ew}_{ji} & \text{if nodes } n_i \text{ and } n_j \text{ are adjacent} \\
\sum_{j=1}^{D_i} \text{ew}_{ij} & \text{for } i = j \\
0 & \text{otherwise}
\end{cases} \tag{7}
\]

In this relation \( \text{ew}_{ij} \) is the weight of the edge \( e_{ij} \), and \( D_i \) is the degree of the node \( n_i \). For a non-weighted graph, the degree matrix \( D = [d_{ij}]_{n \times n} \) is a diagonal matrix of node degrees. Here, the \( i \)th diagonal entry \( d_{ii} \) is equal to the degree of the node \( i \). Therefore, the entries of \( L \) are as:

\[
l_{ij} = \begin{cases} 
-1 & \text{if node } n_i \text{ is adjacent to } n_j \\
\text{deg}(n_i) & \text{if } i = j \\
0 & \text{otherwise}
\end{cases} \tag{8}
\]

Consider the following eigenproblem:

\[
Lv_i = \lambda_i v_i, \tag{9}
\]

where \( \lambda_i \) is the eigenvalue and \( v_i \) is the corresponding eigenvector. As for \( A \), all the eigenvalues of \( L \) are real. It can be shown that matrix \( L \) is a positive semi-definite matrix with

\[
0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n \tag{10}
\]
The second eigenvalue $\lambda_2$ and the corresponding eigenvector $v_2$ has attractive properties. Fiedler [6] has investigated various properties of $\lambda_2$. This eigenvalue is known as the \textit{algebraic connectivity} of a graph, and the corresponding eigenvector $v_2$ is known as the \textit{Fiedlers vector}.

Mohar [7] has applied $(\lambda_2,v_2)$ to different problems such as graph partitioning and ordering. Paulino et al. [15] used $v_2$ for element ordering and nodal numbering.

Pothen et al. [16], Simon [17], Seale and Topping [18], and Kaveh and Davaran [19] and Kaveh and Rahimi Bondarabady [20-21] have used the properties of $v_2$, for partitioning graphs. However, for calculating $\lambda_2$ when the entire model is considered, a fair amount of computational time and storage space is required. In this paper, for regular structural models, this goal is achieved by a far simple and more efficient analytical method.

\section{3. GRAPH PRODUCTS}

\subsection{3.1 Cartesian Product of Two Graphs}

Many structures have regular patterns and can be viewed as the Cartesian product of a number of simple graphs. These subgraphs, which are used in the formation of a model, are called the \textit{generators} of that model.

The simplest Boolean operation on a graph, is the Cartesian product $K \times H$ introduced by Sabidussi [22]. The Cartesian product is a Boolean operation $G = K \times H$ in which for any two nodes $u = (u_1,u_2)$ and $v = (v_1,v_2)$ in $N(K) \times N(H)$, the member $uv$ is in $E(G)$ whenever,

\begin{equation}
    u_1 = v_1 \text{ and } u_2v_2 \in E(H),
\end{equation}

or

\begin{equation}
    u_2 = v_2 \text{ and } u_1v_1 \in E(K).
\end{equation}

As an example, the Cartesian product of $K = P_2$ and $H = P_3$ is shown in Figure 1.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{cartesian_product.png}
\caption{The Cartesian product of two simple graphs.}
\end{figure}
Example: In this example, the Cartesian product $C_7 \times P_5$ of the path graph with 5 nodes denoted by $P_5$ and a cycle graph shown by $C_7$ is illustrated in Figure 2.

![Figure 2. Representations of $C_7 \times P_5$](image)

3.2 Strong Cartesian Product of Two Graphs
This is another Boolean operation, known as the strong Cartesian product. The strong Cartesian product is a Boolean operation $G = K \boxtimes H$ in which, for any two nodes $u = (u_1, u_2)$ and $v = (v_1, v_2)$ in $N(K) \times N(H)$, the member $uv$ is in $E(G)$ if:

1. $u_1 = u_2$ and $v_1v_2 \in M(K)$ is in $E(H)$,
2. $v_1 = v_2$ and $u_1u_2 \in M(K)$ is in $E(K)$,
3. $u_1u_2 \in E(K)$ and $v_1v_2 \in E(H)$

As an example, the strong Cartesian product of $K = P_2$ and $H = P_3$ is shown in Figure 3.

![Figure 3. The strong Cartesian product of two simple graphs](image)
Example: In this example, the strong Cartesian product $P_7 \boxtimes P_5$ of a path graph with 7 nodes, denoted by $P_7$, and the path graph $P_5$, is illustrated in Figure 4.

![Figure 4. Strong product representation of $P_7 \boxtimes P_5$](image)

3.3 Direct Product of Two Graphs
This is another Boolean operation known as the direct product introduced by Weichsel [23], who called it the Kronecker Product. The direct product is a Boolean operation $G = K \times H$ in which for any two nodes $u=(u_1,u_2)$ and $v=(v_1,v_2)$ in $N(K) \times N(H)$, the member $uv$ is in $E(G)$ if:

$$u_1v_1 \in M(K) \text{ and } u_2v_2 \in E(H). \quad (13)$$

As an example, the direct product of $K=P_2$ and $H=P_3$ is shown in Figure 5.

![Figure 5. The direct product of two simple graphs](image)

Example: The direct product $P_7 \times P_5$ of the path graph $P_7$ and path graph $P_5$ is illustrated in Figure 6.
3.4 Kronecker Product

The Kronecker product of two matrices $A$ and $B$, is the matrix we get by replacing the $ij$-th entry of $A$ by $a_{ij}B$, for all $i$ and $j$.

As an example,

$$
\begin{bmatrix}
1 & 1 \\
1 & 0
\end{bmatrix}
\otimes
\begin{bmatrix}
a & b \\
c & d
\end{bmatrix}
= 
\begin{bmatrix}
a & b & a & b \\
c & d & c & d \\
a & b & 0 & 0 \\
c & d & 0 & 0
\end{bmatrix}
$$

where entry 1 in the first matrix has been replaced by a complete copy of the second matrix.

The Kronecker product has the property that if $B$, $C$, $D$, and $E$ are four matrices, such that $BD$ and $CE$ exists, then:

$$(B \otimes C)(D \otimes E) = BD \otimes CE. \quad (15)$$

Thus, if $u$ and $v$ are vectors of the correct dimensions, then:

$$(B \otimes C)(u \otimes v) = Bu \otimes Cv. \quad (16)$$

If $u$ and $v$ are eigenvectors of $B$ and $C$, with eigenvalues $\lambda$ and $\mu$, respectively, then,

$$Bu \otimes Cv = \lambda \mu u \otimes v, \quad (17)$$

Whence $u \otimes v$ is an eigenvector of $B \otimes C$ with eigenvalue $\lambda \mu$. 

Figure 6. Direct product representation of $P_7 \times P_5$
4. A UNIFIED APPROACH FOR EIGENVALUES OF GRAPH PRODUCTS

Consider a block tri-diagonal matrix in the following form:

\[
M_{mn} = \begin{bmatrix}
A & B \\
B & C & B \\
& B & C & B \\
& & \ddots & \ddots & \ddots \\
& & & B & C & B \\
& & & & B & C & B \\
& & & & & B & A \\
\end{bmatrix},
\]

(18)

where \(A, B\) and \(C\) are \(m \times m\) matrix blocks. The matrix \(M_{mn}\) contains \(n\) blocks in each row and \(n\) blocks in each column. A matrix \(M_{mn}\) in the form of Eq. (18) will be denoted by \(M_{mn} = F(A_m, B_m, C_m)_{mn}\). Now we study various forms of \(M_{mn}\).

4.1 FORM 1 (for adjacency matrices)

This form corresponds to the adjacency matrices of three groups of graph products, namely Cartesian, Strong Cartesian and Direct products. Here, it is assumed that weights are associated with the nodes of the graph.

In this form,

\[
M_{mn} = F(A_m, B_m, A_m)_{mn},
\]

(19)

where

\[
A_m = F(a, b, a)_{m} \quad \text{and} \quad B_m = F(c, d, c)_{m}.
\]

(20)

The small characters are numbers, and the capital characters are matrices.

Consider \(T_k = F(0, 1, 0)_{k}\) with eigenvalues \(\lambda_k\), and denote the unit matrix by \(I_k\), where \(k\) is the dimension of the square matrices \(T_k\) and \(I_k\). Using the properties of the Kronecker products from linear algebra, the matrix \(M_{mn}\) can be decomposed as:

\[
M_{mn} = I_n \otimes A_m + T_n \otimes B_m.
\]

(21)

Substituting

\[
A_m = (aI_m + bT_m) \quad \text{and} \quad B_m = (cI_m + dT_m),
\]

(22)

leads to:
A UNIFIED METHOD FOR EIGENDECOMPOSITION OF GRAPH PRODUCTS

\[ M_{mn} = aI_n \otimes I_m + bI_n \otimes T_m + cT_n \otimes I_m + dT_n \otimes T_m. \]  \hspace{1cm} (23)

Therefore:

\[ \lambda = a + b\lambda_m + c\lambda_n + d\lambda_m\lambda_n. \]  \hspace{1cm} (24)

For graph with no weights, the Cartesian product, strong Cartesian product and direct product have coefficients \(a, b, c\) and \(\lambda\) as provided in Table 1.

<table>
<thead>
<tr>
<th>Type of Product</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>(\lambda)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cartesian</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>(\lambda = \lambda_m + \lambda_n)</td>
</tr>
<tr>
<td>Strong Cartesian</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>(\lambda = \lambda_m\lambda_n)</td>
</tr>
<tr>
<td>Direct</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>(\lambda = \lambda_m + \lambda_n - \lambda_m\lambda_n)</td>
</tr>
</tbody>
</table>

Path Graphs: For a path graph \(P_m\) with \(m\) nodes, we have the following special case,

\[ M_{mn} = F(A_m, B_m, A_m)_{mn}, \]  \hspace{1cm} (25)

where \(A_m = F(0, b, 0)_m\) and \(B_m = F(c, d, c)_m.\)  \hspace{1cm} (26)

The matrix \(M_{mn}\) can be decomposed as:

\[ M_{mn} = I_n \otimes A_m + T_n \otimes B_m \]
\[ = bI_n \otimes T_m + T_n \otimes (cI_m + dT_m) \]  \hspace{1cm} (27)
\[ = bI_n \otimes T_m + cT_n \otimes I_m + dT_n \otimes T_m. \]

Therefore:

\[ \lambda = b\lambda_m + c\lambda_n + d\lambda_m\lambda_n. \]  \hspace{1cm} (28)

Cycle Graphs: For a cycle graph \(C_m\), the matrix \(M_{mn}\) is a tri-diagonal matrix similar to that of the path graph, with the difference of \(A_m\) and \(B_m\) having an entry \(p\) in the two corners as:
A. Kaveh and H. Rahami

\[
F = \begin{bmatrix}
  * & * & p \\
  * & * & * \\
  * & * & * \\
  \vdots & \vdots & \vdots \\
  * & * & * \\
  p & * & *
\end{bmatrix}
\]

(29)

where \( p \) is a number.

4.2 FORM 2 (for Laplacian matrices)

This form appears in Laplacian matrices of weighted graphs for Cartesian products, and strong Cartesian and direct products after addition of boundary edges, Ref. [11].

Path Graphs: For a path graph we have:

\[
M_{mm} = F(A_m, B_m, C_m)_{mn},
\]

(30)

where

\[
A_m = F(a_1, b_1, c_1), \quad B_m = F(a_2, b_2, c_2), \quad C_m = F(a_3, b_3, c_3)
\]

(31)

For this form we have

\[
A = B + C \quad \text{and} \quad a_i = b_i + c_i \quad (i=1,2,3).\]

Assuming \( T_k = F(1,-1,2)_k \) with \( \lambda_k \) being the eigenvalues of \( T_k \), we have:

\[
M_{mm} = I_n \otimes (A + B)_m + T_n \otimes (-B)_m
\]

(32)

But

\[
(A+B)_m = (a_1+a_2+b_1+b_2)I_m - (b_1+b_2)T_m, \quad \text{and} \quad B_m = (a_2+b_2)I_m - b_2T_m.
\]

(33)

Therefore,

\[
M_{mm} = (a_1 + a_2 + b_1 + b_2)I_m - (b_1 + b_2)I_n \otimes T_m - (a_2 + b_2)T_n \otimes I_m + b_2T_n \otimes T_m
\]

(34)

and

\[
\lambda = (a_1 + a_2 + b_1 + b_2) - (b_1 + b_2)\lambda_n - (a_2 + b_2)\lambda_n + b_2\lambda_n \lambda_m
\]

(35)

For graph without weights, the Cartesian product, strong Cartesian product and direct product the coefficients \( A_m, B_m, C_m \) and \( \lambda \) are provided in Table 2.
Table 2. The coefficients of $\lambda$.

<table>
<thead>
<tr>
<th>Product</th>
<th>$A_m$</th>
<th>$B_m$</th>
<th>$C_m$</th>
<th>$\lambda$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cartesian</td>
<td>$F(2,-1,3)$</td>
<td>$F(-1,0,-1)$</td>
<td>$F(3,-1,4)$</td>
<td>$\lambda = \lambda_m + \lambda_n$</td>
</tr>
<tr>
<td>Strong Cartesian</td>
<td>$F(3,-1,4)$</td>
<td>$F(-1,-1,0)$</td>
<td>$F(4,0,4)$</td>
<td>$\lambda = 2\lambda_m + 2\lambda_n - \lambda_m \lambda_n$</td>
</tr>
<tr>
<td>Direct</td>
<td>$F(5,-2,2)$</td>
<td>$F(-2,-1,-1)$</td>
<td>$F(7,-1,8)$</td>
<td>$\lambda = 3\lambda_m + 3\lambda_n - \lambda_m \lambda_n$</td>
</tr>
</tbody>
</table>

Note: For weighted graphs, the weight of the added boundary edge should be considered as the weight of the diagonal (bracing) edges.

Cycle Graphs: For cycle graphs, $\lambda_m$ corresponding to a $T_m$ contains additional entries $-1$ in the two far corners.

Once the eigenvalues are found, the corresponding eigenvectors can be calculated. However, this can be done much simpler considering that the eigenvectors of $G$ are the Kronecker product of the eigenvectors of $K$ and $H$, i.e. $w_k = u_i \otimes v_j$, where $w_k$, $u_i$, and $v_j$ are the eigenvectors of $G$, $K$ and $H$, respectively.

**Example:** Consider the Cartesian product of $P_4$ and $P_5$. Let $G = P_4 \times P_5$ be a weighted graph with horizontal edges having weight 2 and the vertical ones with weight 4. In this case, the adjacency matrix $A$ will have the following form:

$$M_{mn} = F(A_5, B_5, C_5)_{54}, A_5 = F(0,2,0), B_5 = F(3,4,3)_5$$

which is the same as Form I, and with $\lambda_4$ and $\lambda_5$ being taken as the eigenvalues of $P_4$ and $P_5$, respectively, we have

$$\lambda = 2\lambda_5 + 3\lambda_4 + 4\lambda_5 \lambda_4; \quad \lambda_n = 2\cos \frac{k\pi}{n+1} \quad (k=1, \ldots, n),$$

and

$$\lambda_{\min} = -8.3182 \quad \text{and} \quad \lambda_{\max} = 8.3182.$$  

For the Laplacian matrix $L$, $\lambda_4$ and $\lambda_5$ correspond to the Laplacian of $P_4$ and $P_5$, and

$$\lambda_n = 2 - 2\cos \frac{k\pi}{n} \quad (k=0, \ldots, n-1)$$

with

$$M_{54} = F(A_5, B_5, C_5)_{54}, A_5 = F(5,-2,7)_5, B_5 = F(-3,0,-3)_5, C_5 = F(8,-2,10)_5$$

It can be observed that $A_5 = B_5 + C_5$ and $a_i = b_i + c_i$ (for i=1,2,3), which are the properties
corresponding to Form II. Therefore:

\[
\lambda = [(5) + (-3) + (-2) + (0)] - [(-2) + (0)] \lambda_3 - [(-3) + (0)] \lambda_4 + (0) \lambda_5 \lambda_4 = 2\lambda_5 + 3\lambda_4
\]

leading to

\[
\lambda_2 = 2(0.3820) + 3(0) = 0.7639
\]

If one considers \(C_5 \times P_4\), then \(\lambda_5\) corresponding to \(C_5\) should be employed. Then we will have numbers in the corner entries of the matrix and a similar method can be used.

For strong Cartesian product of \(P_4\) and \(P_5\), the adjacency matrix has Form I, however, for Laplacian matrix none of the forms discussed will be observed. For this case, edges are added to the boundary nodes (or the weights of the boundary edges are doubled) in order to construct regular graphs. After this operation, Form II is produced and the calculations are performed as before. For direct product, a similar operation can be employed for the Laplacian matrix \(L\).

In this paper, no comparison is made, since the present methods are applicable only to regular models. Due to the analytical nature of these approaches, the computational time is far less than the standard methods for calculating eigenvalues. For applications of the present methods, the reader may refer to References [11,24].

5. CONCLUDING REMARKS

The unified method presented for calculating the eigenvalues of the adjacency and Laplacian matrices of three different graph products, provides an efficient approach for calculating the eigenvalues of adjacency and Laplacian matrices of weighted and non-weighted graphs. The eigensolution of graphs has many applications in computational mechanics. Examples of such applications are nodal and element ordering for bandwidth, profile and frontwidth optimization, graph partitioning, and subdomaining of finite element models [13-21]. The present forms are also effective tools for calculating eigenvalues and eigenvectors of matrices arising from numerical methods for differential equations applied to structural mechanics problems. Such applications are presented in Refs. [11,24].

REFERENCES