Mathematics Interdisciplinary Research **9** (2) (2024) 185 − 197

Original Scientific Paper

On Minimum Algebraic Connectivity of Tricyclic Graphs

Hassan Taheri and Gholam Hossein Fath-Tabar⋆

Abstract

Consider a simple, undirected graph $G = (V, E)$, where A represents the adjacency matrix and Q represents the Laplacian matrix of G. The second smallest eigenvalue of Laplacian matrix of G is called the algebraic connectivity of G. In this article, we present a Python program for studying the Laplacian eigenvalues of a graph. Then, we determine the unique graph of minimum algebraic connectivity in the set of all tricyclic graphs.

Keywords: Algebraic connectivity, Bicyclic graph, Tricyclic graph, Python programming language.

2020 Mathematics Subject Classification: 05C50, 90B10.

How to cite this article
H. Taheri and G. H. Fath-Tabar, On minimum algebraic connectivity of tricyclic graphs, *Math. Interdisc. Res.* **9** (2) (2024) 185-197.

1. Introduction

Let $G = (V, E)$ be a simple graph (that is undirected and finite graph without multiple or loops) on edge set $E = \{e_1, ..., e_m\}$ and vertex set $V = \{v_1, ..., v_n\}$. The size of a graph is its number of edges $m =| E(G) |$ and the order of a graph is its number of vertices $n =| V(G) |$. For each $v_i \in V(G)$, the degree of v_i is the number of edges incident with v_i, which is denoted by d_i or $d(v_i)$. Suppose that δ and Δ are minimum and maximum degree among all vertices, respectively. For any $e \in E(G)$, we use $G - e$ to denote the graph G which one edge is removed.

⋆Corresponding author (E-mail: fathtabar@kashanu.ac.ir)
Academic Editor: Saeid Alikhani
Received 25 September 2023, Accepted 23 March 2024
DOI: 10.22052/MIR.2024.253568.1437

Spectral graph theory [1–3] investigates properties of graphs using the spectrum of related matrices. Adjacency matrix and Laplacian matrix, widely studied, are frequently used in computer science. Adjacency matrix for a graph is denoted by $A = (a_{ij})$. The adjacency matrix is an $n \times n$ matrix related to the graph's vertices. The value of the a_{ij} is equal to the number of edges between vertex i and vertex j. Another much studied matrix is the Laplacian matrix, defined by $Q = D - A$, where D is the $n \times n$ diagonal matrix whose ith diagonal entry is $D_{i,i} = deg(v_i)$. The matrix D is called the degree matrix of G (see [4–6]).
Furthermore note that $Q = C^T C$, where C is the matrix whose rows are indexed by the edges of G and columns are indexed by its vertices, in which each row corresponding to the edge $e = \{u, v\}$, $(u < v)$, has a (1) in the column corresponding to u, a (-1) in that corresponding to v and 0 in every other place. Thus, Q is a symmetric and positive semi-definite matrix. The eigenvalues of a real symmetric matrix $M_{n\times n}$ are real numbers. The eigenvalues (or spectrum) of A and Q which are real eigenvalues, are called A-spectrum and Q-spectrum respectively. These eigenvalues will be shown by

$$\lambda_n(G) \leqslant ... \leqslant \lambda_2(G) \leqslant \lambda_1(G), \tag{1}$$

and

$$0 = \mu_n(G) \leqslant ... \leqslant \mu_2(G) \leqslant \mu_1(G) = \mu, \tag{2}$$

respectively.

Fiedler [7] showed that the second smallest Laplacian eigenvalue G disconnectes if and only if $\mu_{n-1}(G)$ equals 0. Thus, $\mu_{n-1}(G)$ is commonly referred to as the algebraic connectivity of G and is denoted by $\alpha(G)$. Recently, the algebraic connectivity has received significantly regard, see [8] for survey. It turned out many applications in computer science, theoretical chemistry, combinatorial optimization, etc (see [7–11]).

A k-cyclic connected graph is a graph with $K = m - n + 1$ as its cyclomatic number, where m and n, as previously mentioned, are the numbers of edges and vertices of G, respectively. If $K = 0$ then G is called a tree. In cases where K equals $1, 2$, or $3, G$ is called a unicyclic, bicyclic, or tricyclic graph, respectively. In this paper, the set of all bicyclic and tricyclic graphs are denoted by $\mathcal{B}_n$ and $\mathcal{T}_n$, respectively. Jianxi Li in [12] have sorted algebraic connectivity of bicyclic graphs. Here, firstly we write a Python program for studying Laplacian eigenvalues of a graph. Then we determine the unique graph in the set of all tricyclic graphs which has minimum algebraic connectivity. To see the standard notations, you can refer to [4, 13–18].

2. Results

For investigating the eigenvalues of Laplacian graphs, we consider the data set of graphs with 2 to 10 vertices. The number of connected and disconnected graphs

Table 1: The number of connected and disconnected graphs with 2 to 10 vertices.

order graph	number of connected graphs	number of disconnected graphs
2	2	1
3	4	2
4	11	6
5	34	21
6	156	112
7	1044	853
8	12346	11117
9	274668	261080
10	12005168	11716571

are shown in Table 1. We use these data set in Python programming language and are checked the eigenvalues of Laplacian graphs. Also, in the following, we show programs that we have implemented in the Python programming language for these data set of graphs.

Program 1.

```
import numpy as np
import networkx as nx

order = int(input("input the order from 2 to 10: "))

with open(f"./graphs/text/graph{order}c.txt", "r") as graph_file:
    lines = graph_file.readlines()
    for i in range(1, len(lines), order + 2):
        name = lines[i].strip("\n")
        matrix = [
            [int(f) for f in m.strip("\n").split()]
            for m in lines[i + 1 : i + order + 1]
        ]
        print(name)
        array = np.array(matrix)

        G = nx.convert_matrix.from_numpy_array(array)

        L = nx.linalg.laplacianmatrix.laplacian_matrix(G)

        print(array)
        print()
        print(L.A)
```

```
        print()
        print(np.linalg.eigh(L.A)[0])
        print("-" * 40)
```

The output of Program 1 is the Laplacian matrix of all graphs with the entered order along with their eigenvalues.

Program 2.

```
from collections import OrderedDict
import numpy as np
import networkx as nx

order = int(input("input the order from 2 to 10: "))

eigvals = dict()

with open(f"./graphs/text/graph{order}c.txt", "r") as graph_file:
    lines = graph_file.readlines()
    for i in range(1, len(lines), order + 2):
        name = lines[i].strip("\n")
        number = int(name.split(",")[0].strip("Graph "))
        matrix = [
            [int(f) for f in m.strip("\n").split()]
            for m in lines[i + 1 : i + order + 1]
        ]
        array = np.array(matrix)
        G = nx.convert_matrix.from_numpy_array(array)
        L = nx.linalg.laplacianmatrix.laplacian_matrix(G)
        evals = np.linalg.eigh(L.A)[0]
        for eval in evals:
            eval = float(f"{eval:.4f}")
            if eval not in eigvals:
                eigvals[eval] = [(number, order)]
            else:
                if (number, order) not in eigvals[eval]:
                    eigvals[eval].append((number, order))

print()
ordered_eigvals = OrderedDict(sorted(eigvals.items()))
for eigvalue, no_list in ordered_eigvals.items():
    if  7.5 < eigvalue < 8 :
        print(f"Eigenvalue: {eigvalue}:")
```

```
            for graph in no_list:
                print(f"Graph {graph[0]}, Order {graph[1]}")
        else:
            continue
        print("-" * 40)
```

The output of Program 2 is the number of all Laplacian graphs whose eigenvalues are in the specified range with the entered order. For example, in Tables 2 and 3, the results from Programs 1 and 2 demonstrate graphs of order 6. These graphs have Laplacian and sinless Laplacian eigenvalues ranging from 0 to 0.8.

Program 3.

```
import numpy as np
import networkx as nx

number, order = [int(n) for n in input("enter number and order for graph: ").split()]


with open(f"./graphs/text/graph{order}c.txt", "r") as graph_file:
        lines = graph_file.readlines()
        index = ((order + 2) * (number - 1)) + 1
        name = lines[index]
        print(name)
        matrix = [
            [int(f) for f in m.strip("\n").split()]
            for m in lines[index + 1 : index + order + 1]
        ]
        array = np.array(matrix)
        G = nx.convert_matrix.from_numpy_array(array)
        L = nx.linalg.laplacianmatrix.laplacian_matrix(G)
        print(array)
        print()
        print(L.A)
        print()
        evals = [float(f"{eval:.4f}") for eval in np.linalg.eigh(L.A)[0]]
        print(evals)
        print()
        print("-" * 40)
```

The output of Program 3 is the Laplacian graph matrix with the entered number

Table 2: The results of Program 1 display the graphs of order 6, with Laplacian eigenvalues ranging from 0 to 0.8.

input the order from 2 to 10: 6 (0< Laplacian eigenvalues<0.8)	
Eigenvalue: 0.2679:	Graph 19, Order 6
Eigenvalue: 0.3249:	Graph 5, Order 6
	Graph 21, Order 6
Eigenvalue: 0.382:	Graph 15, Order 6
Eigenvalue: 0.4131:	Graph 20, Order 6
Eigenvalue: 0.4384:	Graph 4, Order 6
	Graph 18, Order 6
	Graph 23, Order 6
	Graph 76, Order 6
	Graph 77, Order 6
Eigenvalue: 0.4859:	Graph 2, Order 6
	Graph 16, Order 6
	Graph 24, Order 6
	Graph 78, Order 6
Eigenvalue: 0.5858:	Graph 9, Order 6
	Graph 50, Order 6
Eigenvalue: 0.6314:	Graph 6, Order 6
	Graph 22, Order 6
Eigenvalue: 0.6571:	Graph 31, Order 6
Eigenvalue: 0.6972:	Graph 15, Order 6
	Graph 29, Order 6
	Graph 30, Order 6
	Graph 32, Order 6
	Graph 51, Order 6
Eigenvalue: 0.7216:	Graph 46, Order 6
Eigenvalue: 0.7312:	Graph 52, Order 6
Eigenvalue: 0.7639:	Graph 7, Order 6
	Graph 10, Order 6
	Graph 25, Order 6
	Graph 27, Order 6
	Graph 35, Order 6
	Graph 41, Order 6
	Graph 59, Order 6
	Graph 65, Order 6
	Graph 79, Order 6
	Graph 84, Order 6

input the order from 2 to 10: 6 (0<sinless Laplacian eigenvalues<0.8)	
Eigenvalue: 0.1338:	Graph 21, Order 6
Eigenvalue: 0.1578:	Graph 18, Order 6
Eigenvalue: 0.2015:	Graph 16, Order 6
Eigenvalue: 0.2204:	Graph 20, Order 6
Eigenvalue: 0.2215:	Graph 25, Order 6
	Graph 77, Order 6
Eigenvalue: 0.2434:	Graph 29, Order 6
Eigenvalue: 0.2497:	Graph 46, Order 6
Eigenvalue: 0.2534:	Graph 3, Order 6
Eigenvalue: 0.2602:	Graph 24, Order 6
Eigenvalue: 0.2679:	Graph 19, Order 6
Eigenvalue: 0.2823:	Graph 78, Order 6
Eigenvalue: 0.2907:	Graph 38, Order 6
Eigenvalue: 0.301:	Graph 6, Order 6
Eigenvalue: 0.3077:	Graph 33, Order 6
Eigenvalue: 0.3089:	Graph 27, Order 6
	Graph 42, Order 6
	Graph 65, Order 6
Eigenvalue: 0.3249:	Graph 5, Order 6
Eigenvalue: 0.3542:	Graph 50, Order 6
Eigenvalue: 0.3619:	Graph 36, Order 6
Eigenvalue: 0.3636:	Graph 48, Order 6
Eigenvalue: 0.382:	Graph 15, Order 6
	Graph 29, Order 6
	Graph 30, Order 6
	Graph 32, Order 6
	Graph 51, Order 6
Eigenvalue: 0.3918:	Graph 54, Order 6
Eigenvalue: 0.3961:	Graph 8, Order 6
	Graph 41, Order 6
	Graph 44, Order 6
Eigenvalue: 0.4113:	Graph 96, Order 6
Eigenvalue: 0.4284:	Graph 60, Order 6
Eigenvalue: 0.4298:	Graph 68, Order 6
Eigenvalue: 0.4384:	Graph 4, Order 6
	Graph 23, Order 6
Eigenvalue: 0.4558:	Graph 22, Order 6
Eigenvalue: 0.4628:	Graph 67, Order 6
Eigenvalue: 0.4629:	Graph 20, Order 6
Eigenvalue: 0.4703:	Graph 52, Order 6
Eigenvalue: 0.4711:	Graph 51, Order 6
Eigenvalue: 0.4746:	Graph 10, Order 6
Eigenvalue: 0.4812:	Graph 12, Order 6
	Graph 28, Order 6
	Graph 73, Order 6
	Graph 84, Order 6
Eigenvalue: 0.4859:	Graph 2, Order 6
Eigenvalue: 0.4889:	Graph 6, Order 6
Eigenvalue: 0.5359:	Graph 14, Order 6
Eigenvalue: 0.5443:	Graph 40, Order 6
Eigenvalue: 0.5463:	Graph 63, Order 6

Table 3: The results of Program 2 display the graphs of order 6, with sinless Laplacian eigenvalues ranging from 0 to 0.8.

Eigenvalue: 0.5858:	Graph 9, Order 6
	Graph 10, Order 6
	Graph 35, Order 6
	Graph 38, Order 6
	Graph 39, Order 6
	Graph 50, Order 6
	Graph 81, Order 6
	Graph 95, Order 6
Eigenvalue: 0.6208:	Graph 62, Order 6
Eigenvalue: 0.6224:	Graph 89, Order 6
Eigenvalue: 0.6277:	Graph 17, Order 6
Eigenvalue: 0.646:	Graph 34, Order 6
Eigenvalue: 0.6571:	Graph 31, Order 6
Eigenvalue: 0.6594:	Graph 37, Order 6
Eigenvalue: 0.6721:	Graph 16, Order 6
Eigenvalue: 0.6728:	Graph 32, Order 6
	Graph 49, Order 6
Eigenvalue: 0.6791:	Graph 35, Order 6
Eigenvalue: 0.6856:	Graph 57, Order 6
Eigenvalue: 0.6972:	Graph 15, Order 6
Eigenvalue: 0.7029:	Graph 61, Order 6
	Graph 71, Order 6
Eigenvalue: 0.7066:	Graph 64, Order 6
Eigenvalue: 0.7251:	Graph 90, Order 6
Eigenvalue: 0.7411:	Graph 97, Order 6
Eigenvalue: 0.7639:	Graph 7, Order 6
	Graph 30, Order 6
	Graph 59, Order 6
Eigenvalue: 0.7772:	Graph 33, Order 6
Eigenvalue: 0.7828:	Graph 60, Order 6
Eigenvalue: 0.7933:	Graph 92, Order 6

and order along with its eigenvalues. Firstly enter number and order for graph, it displays adjacency matrix, Laplacian matrix and eigenvalues of Laplacian matrix. For example, for Graph 265 and order 9, using Program 3, we obtain

$$\begin{pmatrix} 0&0&0&0&0&0&1&0&1\\ 0&0&0&0&0&0&1&0&1\\ 0&0&0&0&0&0&0&1&1\\ 0&0&0&0&0&0&0&1&1\\ 0&0&0&0&0&0&0&1&1\\ 0&0&0&0&0&0&0&0&1\\ 1&1&0&0&0&0&0&0&1\\ 0&0&1&1&1&0&0&0&1\\ 1&1&1&1&1&1&1&1&0 \end{pmatrix}$$

$$\begin{pmatrix} 2&0&0&0&0&0&-1&0&-1\\ 0&2&0&0&0&0&-1&0&-1\\ 0&0&2&0&0&0&0&-1&-1\\ 0&0&0&2&0&0&0&-1&-1\\ 0&0&0&0&2&0&0&-1&-1\\ 0&0&0&0&0&-1&0&0&-1\\ -1&-1&0&0&0&0&3&0&-1\\ 0&0&-1&-1&-1&0&0&4&-1\\ -1&-1&-1&-1&-1&-1&-1&-1&8 \end{pmatrix}$$

[0.0, 1.0, 1.0, 2.0, 2.0, 2.0, 4.0, 5.0, 9.0]

Lemma 2.1. *([2]). Suppose G is a graph, $\mid V(G) \mid = n$ and $G \ncong K_n$ (not-isomorphic to the complete graph) and suppose $G' = G + e$ is a new graph taken from G by adding a new edge e. After that the Laplacian eigenvalues of G and G' intermix, in other words*

$$\mu_i(G') \geqslant \mu_i(G) \geqslant \mu_{i+1}(G'), \tag{3}$$

for $1 \leqslant i \leqslant (n-1)$.

Lemma 2.2. *([19]). Let G be a connected graph of order n. Suppose $v_1, \dots, v_s (s > 2)$ are non-adjacent vertices of G and $N(v_1) = \cdots = N(v_s)$. Let G_t be a graph obtained from G by adding any t $(0 \leqslant t \leqslant \frac{s(s-1)}{2})$ edges among $v_1, \dots, v_s$. If $\mu(G) \neq d(v_1)$, then*

$$\mu(G) = \mu(G_t). \tag{4}$$

Theorem 2.3. *([12]). Suppose*

$$B \in \mathcal{B}_n - \{B_1, B_2, B_3, B_4\}, \tag{5}$$

where $13 \leqslant n$. Then $\alpha(B) > \alpha(B_4)$. Moreover,

$$\alpha(B_1) < \alpha(B_2) < \alpha(B_3) < \alpha(B_4), \tag{6}$$

where B_1, B_2, B_3, B_4 are shown in Figure 1.

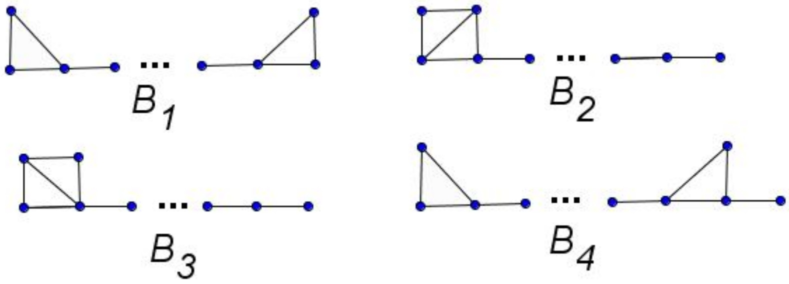

Figure 1: Bicyclic graph B_i (1$\leqslant$ i $\leqslant$ 4.)

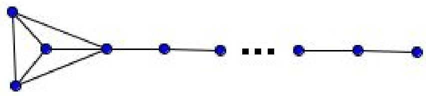

Figure 2: Tricyclic graph T_3.

Theorem 2.4. *Let $T \in \mathcal{T}_n$ with $n \geqslant 14$ and T_3 is shown in Figure 2. Then*

$$\alpha(T_3) \leqslant \alpha(T). \tag{7}$$

Proof. Let $T \in \mathcal{T}_n$ and C_k, C_l and C_w be three independent cycles in T, where $3 \leqslant k, l, w$. we can choose edge, say e, in C_i ($i = k, l$ and w) such that $T - e \in B_n - \{B_1, B_2\}$. So by Lemma 2.1, we have

$$\alpha(T) \geqslant \alpha(T - e). \tag{8}$$

In the following, we consider two cases.

Case 1: $T - e \ncong B_3$
By Theorem 2.3

$$\alpha(T - e) \geqslant \alpha(B_4) > \alpha(B_3). \tag{9}$$

Otherwise, by Lemma 2.2 we have

$$\alpha(B_3) = \alpha(T_3). \tag{10}$$

Thus

$$\alpha(T) > \alpha(T_3). \tag{11}$$

Case 2: $T - e \cong T_3$ By Lemma 2.2

$$\alpha(B_3) = \alpha(T_3). \tag{12}$$

As a result $\alpha(T-e)=\alpha(T_3)$.

Thus, for every $T\in\mathcal{T}_n$, $\alpha(T)\geqslant\alpha(T_3)$ and the equality holds if and only if $T\cong T_3$ $\square$

Conflicts of Interest. The authors declare that they have no conflicts of interest regarding the publication of this article.

Acknowledgments. The research of this paper is partially supported by the University of Kashan under grant no 159021.

References

[1] F. R. K. Chung, *Spectral Graph Theory*, CBMS Regional Conference Series in Mathematics, vol. 92, American Mathematical Soc., 1997

[2] D. M. Cvetković, M. Doob and H. Sachs, *Spectra of Graphs: Theory and Applications*, Johann Ambrosius Barth Verlag, Heidelberg, Leipzig, 1995.

[3] E. R. van Dam and W. H. Haemers, Which graphs are determined by their spectrum?, *Linear Algebra Appl.* **373** (2003) 241 – 272, https://doi.org/10.1016/S0024-3795(03)00483-X.

[4] W. N. Anderson Jr and T. D. Morley, Eigenvalues of the Laplacian of a graph, *Linear Multilinear Algebra* **18** (1985) 141 – 145, https://doi.org/10.1080/03081088508817681.

[5] M. Fiedler, A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory, *Czechoslovak Math. J.* **25** (1975) 619 – 633, https://doi.org/10.21136/CMJ.1975.101357.

[6] R. Merris, Laplacian matrices of graphs: a survey, *Linear Algebra Appl.* **197-198** (1994) 143 – 176, https://doi.org/10.1016/0024-3795(94)90486-3.

[7] M. Fiedler, Algebraic connectivity of graphs, *Czechoslov. Math. J.* **23** (1973) 298 – 305, https://doi.org/10.21136/CMJ.1973.101168.

[8] N. M. M. de Abreu, Old and new results on algebraic connectivity of graphs, *Linear Algebra Appl.* **423** (2007) 53 – 73, https://doi.org/10.1016/j.laa.2006.08.017.

[9] R. Nasiri, H. R. Ellahi, A. Gholami and G. H. Fath-Tabar, The irregularity and total irregularity of Eulerian graphs, *Iranian J. Math. Chem.* **9** (2018) 101 – 111, https://doi.org/10.22052/IJMC.2018.44232.1153.

[10] D. Vukičević and Z. Yarahmadi, One-alpha descriptor, *Iranian J. Math. Chem.* **9** (2018) 179 – 186, https://doi.org/ 10.22052/IJMC.2018.118091.1342.

[11] D. Cvetković, M. Doob, I. Gutman and A. Torgašev, Recent Results in the Theory of Graph Spectra, *Ann. Discrete Math.* **36**, North-Holland, Amsterdam, 1988.

[12] J. Li, J. M. Guo and W. C. Shiu, The orderings of bicyclic graphs and connected graphs by algebraic connectivity, *Electron. J. Combin.* **17** (2010) Research Paper 162, https://doi.org/10.37236/434.

[13] A. Z. Abdian, A. R. Ashrafi, L. W. Beineke, M. R. Oboudi and G. H. Fath-Tabar, Monster graphs are determined by their Laplacian spectra, *Rev. Un. Mat. Argentina* **63** (2022) 413 – 424, https://doi.org/10.33044/revuma.1769.

[14] M. Arabzadeh, G. H. Fath–Tabar, H. Rasoli and A. Tehranian, Estrada and L-estrada indices of a graph and their relationship with the number of spanning trees, *MATCH Commun. Math. Comput. Chem.* **90** (2023) 787 – 798, https://doi.org/10.46793/match.90-3.787A.

[15] G. K. Gök, Kirchhoff index and Kirchhoff energy, *Iranian J. Math. Chem.* **13** (2022) 175 – 185, https://doi.org/10.22052/IJMC.2022.246278.1619.

[16] T. Vetrik, Degree-based function index of graphs with given connectivity, *Iranian J. Math. Chem.* **14** (2023) 183 – 194, https://doi.org/ 10.22052/I-JMC.2023.252646.1699.

[17] M. Taheri-Dehkordi and G. H. Fath-Tabar, On the number of perfect star packing and perfect pseudo matching in some fullerene graphs, *Iranian J. Math. Chem.* **14** (2023) 7 – 18, https://doi.org/10.22052/IJMC.2022.248451.1669.

[18] M. Arabzadeh, G. H. Fath-Tabar, H. Rasouli and A. Tehranian, On the difference between Laplacian and signless Laplacian coefficients of a graph and its applications on the fullerene graphs, *Iranian J. Math. Chem.* **15** (2024) 39 – 50, https://doi.org/10.22052/IJMC.2024.254123.1808.

[19] J. -Y. Shao, J. -M. Guo and H. -Y. Shan, The ordering of trees and connected graphs by their algebraic connectivity, *Linear Algebra Appl.* **428** (2008) 1421–1438, https://doi.org/10.1016/j.laa.2007.08.031.

Hassan Taheri
Faculty of Mathematical Science,
Department of Pure Mathematics,
University of Kashan,
Kashan 87317-51167, I. R. Iran
e-mail: taherihassan65@gmail.com

Gholam Hossein Fath-Tabar
Faculty of Mathematical Science,
Department of Pure Mathematics,
University of Kashan,
Kashan 87317-51167, I. R. Iran
e-mail: fathtabar@kashanu.ac.ir