ON A NEW ALGORITHM FOR COMPUTING SYMMETRY OF BIG FULLERENES

Ali Reza ASHRAFI

Department of Mathematics, Faculty of Science, University of Kashan, Kashan 87317-51167, Iran; e-mail: ashrafi@kashanu.ac.ir

Received October 24, 2005
Accepted June 21, 2006

In this paper, a new algorithm is presented which is useful for computing the automorphism group of chemical graphs. We compare our algorithm with those of Druffel, Schmidt and Wang. It is proved that the running time of the present algorithm is better than the mentioned algorithms. Also, the automorphism group of Euclidean graph of isomers for the fullerenes C_{180}, C_{240}, C_{260}, C_{320}, C_{500} and C_{720} are computed.

Keywords: Fullerenes; Carbon clusters; Automorphism group of graph; Symmetry; Chemical graphs; Matrices; MATLAB.

An important area of research in nano science and nano technology is the carbon-based physics, more specifically the fullerene physics. The fullerene era was started in 1985 with the discovery of a stable C_{60} cluster and its interpretation as a cage structure with the familiar shape of a soccer ball, by Kroto et al.[1] The well-known fullerene, the C_{60} molecule, is a closed-cage carbon molecule with three-coordinate carbon atoms tiling the spherical or nearly spherical surface with a truncated icosahedral structure formed by 20 hexagonal and 12 pentagonal rings[2].

Fullerene chemistry is nowadays a well-established field of both theoretical and experimental investigations. The initial fascinating appeal, coming from their beautiful symmetry shifted later to real chemistry.

Combinatorial enumerations have found a wide-ranging application in chemistry, since chemical structural formulas can be regarded as graphs or three-dimensional objects. Fullerenes are molecules in the form of polyhedral closed cages made up entirely of n three coordinate carbon atoms and having 12 pentagonal and (n/2 – 10) hexagonal faces, where n is equal or greater than 20. To compute the number of isomers of fullerene, we must compute the symmetry of fullerenes under consideration. Fripertinger[3] computed the symmetry of some fullerenes to calculate the number of

doi:10.1135/cccc20061270

$C_{60}H_kCl_{60-k}$ molecules and Balasubramanian[4] applied the same approach to compute the number of $C_{60}H_{36}$ isomers.

Balasubramanian[5–13] in some leading papers considered the Euclidean matrix of a chemical graph to find its symmetry. He proved that for computing the symmetry of a molecule, it is sufficient to solve the matrix equation $P^t EP = E$, where E is the Euclidean matrix of the molecule under consideration and P varies on the set of all permutation matrices with the same dimension as E. He computed the Euclidean graphs and automorphism group for benzene, eclipsed and staggered forms of ethane and eclipsed and staggered forms of ferrocene[5].

Ashrafi[14] introduced an efficient algorithm for computing the symmetry of molecules. Using this algorithm in ref.[15], the authors computed the symmetry of fullerene C_{80} with the I_h point group symmetry. The goal of this paper is to improve this algorithm for computing the symmetry of molecules[16]. Using this algorithm, the symmetries of some big molecules are computed.

Throughout this paper, only complete weighted finite graphs are investigated. Our notation is standard and taken mainly from refs[17,18]. We encourage the reader to consult refs[1,2,19] for information on fullerenes.

THEORY

We first describe some definitions and notations which will be kept throughout.

A weighted graph $G = (V, E, w)$ is a combinatorial object consisting of an arbitrary set $V = V(G)$ of vertices, a set $E = E(G)$ of unordered pairs $\{x,y\} = xy$ of distinct vertices of G called edges, and a weighting function w, where w: $V(G) \to R$ assigns positive real numbers (weights) to edges. An automorphism of a weighted graph G is a permutation g of the vertex set of G with the property that: (i) for any vertices u and v, $g(u)$ and $g(v)$ are adjacent if and only if u is adjacent to v; (ii) for every edge e, $w(g(e)) = w(e)$. The set of all automorphisms of a graph G, with the operation of composition of permutations, is a permutation group on $V(G)$, denoted Aut(G).

An Euclidean graph[20] is a weighted graph related to a molecule with the adjacency matrix $D = [d_{ij}]$, where for $i \neq j$, d_{ij} is the Euclidean distance between the nuclei i and j. In this matrix d_{ii} can be taken as zero if all the nuclei are equivalent. Otherwise, one may introduce different weights for different nuclei.

By symmetry we mean the automorphism group symmetry of a graph. Randic[21,22], showed that a graph can be depicted in different ways such that its point group symmetry or three-dimensional (3D) perception may differ, but the underlying connectivity symmetry is still the same as characterized by the automorphism group of the graph.

For a permutation σ on n objects, the corresponding permutation matrix is an $n \times n$ matrix P_σ given by $P_\sigma = [x_{ij}]$, $x_{ij} = 1$ if $i = \sigma(j)$ and 0 otherwise. It is easy to see that $P_\sigma P_\tau = P_{\sigma\tau}$, for any two permutations σ and τ on n objects, and so the set of all $n \times n$ permutation matrices is a group isomorphic to the symmetric group S_n on n symbols. It is a well-known fact that a permutation σ of the vertices of a graph G belongs to its automorphism group if it satisfies $P_\sigma{}^{\mathrm{t}} A P_\sigma = A$, where A is the adjacency matrix of G. Suppose $\mathrm{Aut}(G) = \{\sigma_1, ..., \sigma_m\}$. The matrix $S_{\mathrm{G}} = [s_{ij}]$, where $s_{ij} = \sigma_i(j)$ is called a solution matrix for G. Clearly, for computing the automorphism group of G, it is enough to calculate a solution matrix for G.

In mathematics, groups are often used to describe symmetries of objects. This is formalized by the notion of a group action: every element of the group “acts” like a bijective map (or “symmetry”) on some set. To clarify this notion, we assume that G is a group and X is a set. G is said to act on X when there is a map $\phi : G \times X \to X$ such that all elements $x \in X$, (i) $\phi(e,x) = x$ where e is the identity element of G, and, (ii) $\phi(g, \phi(h,x)) = \phi(gh,x)$ for all $g, h \in G$. In this case, G is called a transformation group, X is called a G-set, and ϕ is called the group action. For simplicity we define $gx = \phi(g,x)$.

In a group action, a group permutes the elements of X. The identity does nothing, while a composition of actions corresponds to the action of the composition. For a given X, the set $\{gx \mid g \in G\}$, where the group action moves x, is called the group orbit of x. The subgroup which fixes is the isotropy group of x.

Following Druffel, Schmidt and Wang[23], the collection automorphisms of a given graph G with vertices $\{\nu_1, \nu_2, ..., \nu_n\}$ which leaves ν_1 through ν_{i-1} invariant and map ν_i onto ν_k, $i \le k \le n$, is denoted by $\Phi_{i,k}$. It is easy to see that $\Phi_{i,k}$, $i < k$ may be empty and it does not contain the identity automorphism. Then the set S containing one element from each set $\Phi_{i,k}$ is non-empty and $i < k$ forms a generating set for $\mathrm{Aut}(G)$. If S_k denote the set of all automorphisms in S which leave ν_1 through ν_{k-1} invariant and map ν_k onto ν_j for some $j \ne k$ then $\mathrm{Aut}(G)$ has exactly $(s_1 + 1)(s_2 + 1)...(s_n + 1)$ automorphisms, where $s_k = |S_k|$, $i \le k \le n$.

The symmetry of a graph does not need to be the same (i.e. isomorphic to) the molecular point group symmetry. However, it does represent the maximal symmetry which the geometrical realization of a given topological structure may possess. The automorphisms have other advantages such as in generating nuclear spin species, NMR spectra, nuclear spin statistics in molecular spectroscopy, chirality and chemical isomerism.

Now we discuss techniques that are useful in finding symmetry of molecules. We begin with some important results that, in certain situations, are very effective in establishing automorphisms of Euclidean graphs[14].

Lemma 1. Suppose $A = [a_{ij}]$ and $B = [b_{ij}]$ are two matrices and P_σ is a permutation matrix. If $B = P_\sigma A(P_\sigma)^\mathrm{t}$, $\sigma(i) = r$ and $\sigma(j) = s$, then $a_{rs} = b_{ij}$.

Lemma 2. Let $A = [a_{ij}]$ be the adjacency matrix of a weighted graph and σ be a permutation such that $A = P_\sigma A(P_\sigma)^\mathrm{t}$ and σ maps $i_1 \to j_1$, $i_2 \to j_2$, ..., $i_\mathrm{t} \to j_\mathrm{t}$. Then we have:

$$\begin{bmatrix} a_{i_1 i_1} & \cdot & \cdot & \cdot & a_{i_1 i_t} \\ \cdot & \cdot & & & \cdot \\ \cdot & & \cdot & & \cdot \\ \cdot & & & \cdot & \cdot \\ a_{i_t i_1} & \cdot & \cdot & \cdot & a_{i_t i_t} \end{bmatrix} = \begin{bmatrix} a_{j_1 j_1} & \cdot & \cdot & \cdot & a_{j_1 j_t} \\ \cdot & \cdot & & & \cdot \\ \cdot & & \cdot & & \cdot \\ \cdot & & & \cdot & \cdot \\ a_{j_t j_1} & \cdot & \cdot & \cdot & a_{j_t j_t} \end{bmatrix}$$

Lemma 3. Let G be an Euclidean graph, $A = \mathrm{Aut}(G)$ and $O_1, O_2, ..., O_t$ be orbits of the action of A on the vertices of G. Then for every $\alpha \in A$ and every positive integer i, $1 \le i \le t$, $\alpha(O_i) = O_i$.

RESULTS AND DISCUSSION

Our computations of symmetry properties of molecules were carried out with the use of GAP [24] (GAP stands for Groups, Algorithms and Programming). The name was chosen to reflect the aim of the system, which is a group theoretical software for solving computational problems in computational group theory. This software was constructed by the GAP team in Aachen. The motivation for this study is outlined in ref.[25] and the reader is encouraged to consult this paper for background material as well as basic computational techniques.

MATLAB [26] is an important software for working with matrices. In ref.[15], the authors wrote a MATLAB program for computing symmetry of fullerene molecule C_{80}. We apply Lemmas 1, 2 and 3, to improve this MATLAB program for computing a solution matrix for the automorphism group of Euclidean graphs. The main difference between our two MATLAB programs is that in the second program any vertex x of a graph can be mapped only on the vertices of the orbit $\{gx \mid g \in G\}$. This causes to find better running time in computing the symmetry group of molecules. We can also write another MATLAB Program based on Druffel, Schmidt and Wang's paper[23]. The

MATLAB programs used in this paper are available upon the request from the author.

To compare these programs, we compute the symmetry of seven big fullerenes C_{180}, C_{240}, C_{260}, C_{320}, C_{500}, C_{540} and C_{720}. In Table I, we determine the running time of three programs for computing the symmetry of these fullerenes. Also, we calculate the order of symmetry group. The Cartesian coordinates of these fullerenes are taken from the homepage of Dr S. Weber at http://jcrystal.com/steffenweber/.

In what follows, we explain only our calculations on symmetry of C_{540}. Suppose A is a solution matrix computed by our MATLAB program. To compute the automorphism group of Euclidean graph of this fullerene, we need a GAP program as follows:

```
B:=[];
N:=Size(A);
for i in [1,2..N] do
        d:=PermListList(A[1],A[i]);
        Add(B,d);
od;
G:=AsGroup(B);
GeneratorsOfGroup(G);
```

By our calculations the automorphism group of the Euclidean graph of C_{500} and C_{720} are trivial. We now apply our programs to compute a generating set for the automorphism group of C_{540}. Suppose $\{X, Y, Z\}$ is a generat-

TABLE I
Comparison of running times for three algorithms

Fullerene	Program 1	Program 2	Our last program	Group order
C_{180}	390.531	479.782	1070	120
C_{240}	2009	2783.1	4334.6	120
C_{260}	489.078	773.125	4760	60
C_{320}	3775	5087.1	13801	120
C_{500}	128.078	2785.2	16006	1
C_{540}	22841.61	31289.58	>100000	120
C_{720}	701.921	1321.43	2464.8	1

ing set for the automorphism group of this fullerene. Our calculations give the following permutations as a generating set for the symmetry group of this fullerene:

X:= (2,5)(3,4)(6,10)(7,9)(11,30)(12,29)(13,28)(14,27)(15,26)(16,25)(17,24)(18,23)(19,22)(20,1)
(32,35)(33,34)(36,40)(37,39)(41,60)(42,59)(43,58)(44,57)(45,56)(46,55)(47,54)(48,53)(49,52)
(50,51)(61,102)(62,101)(63,100)(64,99)(65,98)(66,97)(67,104)(68,103)(69,108)(70,107)
(71,106)(72,105)(73,114)(74,113)(75,112)(76,111)(77,110)(78,109)(79,84)(80,83)(81,82)
(85,86)(87,90)(88,89)(91,96)(92,95)(93,94)(115,138)(116,137)(117,136)(118,135)(119,134)
(120,133)(121,140)(122,139)(123,144)(124,143)(125,142)(126,141)(127,150)(128,149)
(129,148)(130,147)(131,146)(132,145)(151,174)(152,173)(153,172)(154,171)(155,170)
(156,169)(157,176)(158,175)(159,180)(160,179)(161,178)(162,177)(163,186)(164,185)
(165,184)(166,183)(167,182)(168,181)(187,228)(188,227)(189,226)(190,225)(191,224)
(192,223)(193,230)(194,229)(195,234)(196,233)(197,232)(198,231)(199,240)(200,239)
(201,238)(202,237)(203,236)(204,235)(205,210)(206,209)(207,208)(211,212)(213,216)
(214,215)(217,222)(218,221)(219,220)(241,314)(242,313)(243,318)(244,317)(245,316)
(246,315)(247,321)(248,322)(249,319)(250,320)(251,324)(252,323)(253,326)(254,325)
(255,330)(256,329)(257,328)(258,327)(259,300)(260,299)(261,298)(262,297)(263,296)
(264,295)(265,302)(266,301)(267,306)(268,305)(269,304)(270,303)(271,312)(272,311)
(273,310)(274,309)(275,308)(276,307)(277,278)(279,282)(280,281)(283,285)(284,286)
(287,288)(289,290)(291,294)(292,293)(331,404)(332,403)(333,408)(334,407)(335,406)
(336,405)(337,411)(338,412)(339,409)(340,410)(341,414)(342,413)(343,416)(344,415)
(345,420)(346,419)(347,418)(348,417)(349,386)(350,385)(351,390)(352,389)(353,388)
(354,387)(355,393)(356,394)(357,391)(358,392)(359,396)(360,395)(361,398)(362,397)
(363,402)(364,401)(365,400)(366,399)(367,372)(368,371)(369,370)(373,374)(375,378)
(376,377)(379,384)(380,383)(381,382)(421,433)(422,434)(423,435)(424,436)(425,446)
(426,445)(427,448)(428,447)(429,442)(430,441)(431,444)(432,443)(437,438)(439,440)
(453,458)(454,457)(455,460)(456,459)(465,478)(466,477)(467,480)(468,479)(469,474)
(470,473)(471,476)(472,475)(481,489)(482,490)(483,491)(484,492)(485,498)(486,497)
(487,500)(488,499)(493,494)(495,496)(501,517)(502,518)(503,519)(504,520)(505,522)
(506,521)(507,524)(508,523)(509,513)(510,514)(511,515)(512,516)(525,537)(526,538)
(527,539)(528,540)(529,533)(530,534)(531,535)(532,536),

Y:= (1,2)(3,5)(6,15)(7,14)(8,13)(9,12)(10,11)(16,30)(17,29)(18,28)(19,27)(20,26)(21,25)(22,24)
(31,35)(32,34)(36,60)(37,59)(38,58)(39,57)(40,56)(41,55)(42,54)(43,53)(44,52)(45,51)(46,50)
(47,49)(61,120)(62,119)(63,118)(64,117)(65,116)(66,115)(67,122)(68,121)(69,126)(70,125)
(71,124)(72,123)(73,132)(74,131)(75,130)(76,129)(77,128)(78,127)(79,102)(80,101)(81,100)
(82,99)(83,98)(84,97)(85,104)(86,103)(87,108)(88,107)(89,106)(90,105)(91,114)(92,113)
(93,112)(94,111)(95,110)(96,109)(133,138)(134,137)(135,136)(139,140)(141,144)(142,143)
(145,150)(146,149)(147,148)(151,156)(152,155)(153,154)(157,158)(159,162)(160,161)
(163,168)(164,167)(165,166)(169,228)(170,227)(171,226)(172,225)(173,224)(174,223)
(175,230)(176,229)(177,234)(178,233)(179,232)(180,231)(181,240)(182,239)(183,238)
(184,237)(185,236)(186,235)(187,210)(188,209)(189,208)(190,207)(191,206)(192,205)
(193,212)(194,211)(195,216)(196,215)(197,214)(198,213)(199,222)(200,221)(201,220)
(202,219)(203,218)(204,217)(241,278)(242,277)(243,282)(244,281)(245,280)(246,279)
(247,285)(248,286)(249,283)(250,284)(251,288)(252,287)(253,290)(254,289)(255,294)
(256,293)(257,292)(258,291)(259,264)(260,263)(261,262)(265,266)(267,270)(268,269)
(271,276)(272,275)(273,274)(295,404)(296,403)(297,408)(298,407)(299,406)(300,405)

(301,411)(302,412)(303,409)(304,410)(305,414)(306,413)(307,416)(308,415)(309,420)
(310,419)(311,418)(312,417)(313,386)(314,385)(315,390)(316,389)(317,388)(318,387)
(319,393)(320,394)(321,391)(322,392)(323,396)(324,395)(325,398)(326,397)(327,402)
(328,401)(329,400)(330,399)(331,372)(332,371)(333,370)(334,369)(335,368)(336,367)
(337,374)(338,373)(339,378)(340,377)(341,376)(342,375)(343,384)(344,383)(345,382)
(346,381)(347,380)(348,379)(349,350)(351,354)(352,353)(355,357)(356,358)(359,360)
(361,362)(363,366)(364,365)(421,441)(422,442)(423,443)(424,444)(425,454)(426,453)
(427,456)(428,455)(429,450)(430,449)(431,452)(432,451)(437,446)(438,445)(439,448)
(440,447)(457,458)(459,460)(461,470)(462,469)(463,472)(464,471)(465,466)(467,468)
(473,489)(474,490)(475,491)(476,492)(477,498)(478,497)(479,500)(480,499)(485,494)
(486,493)(487,496)(488,495)(501,509)(502,510)(503,511)(504,512)(505,514)(506,513)
(507,516)(508,515)(517,537)(518,538)(519,539)(520,540)(521,533)(522,534)(523,535)
(524,536)(525,529)(526,530)(527,531)(528,532),

Z:= (1,6,49,45,18)(2,7,50,41,19)(3,8,46,42,20)(4,9,47,43,16)(5,10,48,44,17)(11,12,13,14,15)
(21,30,52,35,39)(22,26,53,31,40)(23,27,54,32,36)(24,28,55,33,37)(25,29,51,34,38)
(56,60,59,58,57)(61,137,259,241,403)(62,138,260,242,404)(63,133,261,243,405)
(64,134,262,244,406)(65,135,263,245,407)(66,136,264,246,408)(67,144,265,247,409)
(68,143,266,248,410)(69,140,267,249,411)(70,139,268,250,412)(71,141,269,251,413)
(72,142,270,252,414)(73,149,271,253,415)(74,150,272,254,416)(75,145,273,255,417)
(76,146,274,256,418)(77,147,275,257,419)(78,148,276,258,420)(79,117,277,191,389)
(80,118,278,192,390)(81,119,279,187,385)(82,120,280,188,386)(83,115,281,189,387)
(84,116,282,190,388)(85,124,283,198,396)(86,123,284,197,395)(87,125,285,194,392)
(88,126,286,193,391)(89,122,287,195,393)(90,121,288,196,394)(91,129,289,203,401)
(92,130,290,204,402)(93,131,291,199,397)(94,132,292,200,398)(95,127,293,201,399)
(96,128,294,202,400)(97,299,209,171,369)(98,300,210,172,370)(99,295,205,173,371)
(100,296,206,174,372)(101,297,207,169,367)(102,298,208,170,368)(103,306,216,178,376)
(104,305,215,177,375)(105,302,212,179,377)(106,301,211,180,378)(107,303,213,176,374)
(108,304,214,175,373)(109,311,221,183,381)(110,312,222,184,382)(111,307,217,185,383)
(112,308,218,186,384)(113,309,219,181,379)(114,310,220,182,380)(151,351,333,317,227)
(152,352,334,318,228)(153,353,335,313,223)(154,354,336,314,224)(155,349,331,315,225)
(156,350,332,316,226)(157,358,340,324,234)(158,357,339,323,233)(159,359,341,320,230)
(160,360,342,319,229)(161,356,338,321,231)(162,355,337,322,232)(163,363,345,329,239)
(164,364,346,330,240)(165,365,347,325,235)(166,366,348,326,236)(167,361,343,327,237)
(168,362,344,328,238)(421,450,509,486,538)(422,449,510,485,537)(423,452,511,488,540)
(424,451,512,487,539)(425,430,458,501,506)(426,429,457,502,505)(427,432,460,503,508)
(428,431,459,504,507)(433,453,494,474,534)(434,454,493,473,533)(435,455,496,476,536)
(436,456,495,475,535)(437,442,513,481,478)(438,441,514,482,477)(439,444,515,483,480)
(440,443,516,484,479)(445,517,489,462,530)(446,518,490,461,529)(447,519,491,464,532)
(448,520,492,463,531)(465,526,522,498,470)(466,525,521,497,469)(467,528,524,500,472)
(468,527,523,499,471).

We record in the first and second columns of Table I the running time of our two programs. The third column of this table shows the running time of our last program, see ref.[15] From this table, we can see that our new algorithm and MATLAB Program has the best running time of the three algorithms.

This work was partially supported by IUT (CEAMA). I am greatly indebted to the referees, whose valuable criticisms and suggestions led me to rearrange the paper.

REFERENCES

1. Kroto H. W., Heath J. R., O'Brien S. C., Curl R. F., Smalley R. E.: *Nature* **1985**, *318*, 162.
2. Kroto H. W., Fichier J. E., Cox D. E.: *The Fullerene*. Pergamon Press, New York 1993.
3. Fripertinger H.: *MATCH Commun. Math. Comput. Chem.* **1996**, *33*, 121.
4. Balasubramanian K.: *Chem. Phys. Lett.* **2004**, *391*, 69.
5. Balasubramanian K.: *Chem. Phys. Lett.* **1995**, *232*, 415.
6. Balasubramanian K.: *J. Chem. Phys.* **1980**, *72*, 665.
7. Balasubramanian K.: *Int. J. Quantum Chem.* **1982**, *21*, 411.
8. Balasubramanian K.: *Chem. Rev.* **1985**, *85*, 599.
9. Balasubramanian K.: *Stud. Phys. Theor. Chem.* **1983**, *23*, 149.
10. Balasubramanian K.: *J. Chem. Phys.* **1981**, *75*, 4572.
11. Balasubramanian K.: *J. Phys. Chem. A* **2004**, *108*, 5527.
12. Balasubramanian K.: *Chem. Phys. Lett.* **2004**, *391*, 64.
13. Balasubramanian K.: *Chem. Phys. Lett.* **2004**, *391*, 69.
14. Ashrafi A. R.: *Chem. Phys. Lett.* **2005**, *406*, 75.
15. Ashrafi A. R., Ahmadi M. R.: *Cent. Eur. J. Chem.* **2005**, *3*, 647.
16. Ashrafi A. R., Hamadanian M.: *Croat. Chem. Acta* **2005**, *78*, 159.
17. Trinajstic N.: *Chemical Graph Theory*. CRC Press, Boca Raton, FL 1992.
18. Harary F.: *Graph Theory*. Addison–Wesley, Reading, MA 1969.
19. Vukičević D., Kroto H. W., Randić M.: *Croat. Chem. Acta* **2005**, *78*, 223.
20. Skiena S: *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematics*. Addison–Wesley, Reading, MA 1990.
21. Randić M.: *Chem. Phys. Lett.* **1976**, *42*, 283.
22. Randić M.: *J. Chem. Phys.* **1974**, *60*, 3920.
23. Druffel L. E., Schmidt D. C., Wang D.-L.: *SIAM J. Appl. Math.* **1978**, *34*, 593.
24. Schönert M., Besche H. U., Breuer Th., Celler F., Eick B., Felsch V., Hulpke A., Mnich J., Nickel W., Pfeiffer G., Polis U., Theißen H., Niemeyer A.: *GAP, Groups, Algorithms and Programming*. Department of Mathematics, RWTH Aachen 1995.
25. Ashrafi A. R.: *MATCH Commun. Math. Comput. Chem.* **2005**, *53*, 161.
26. Higham D. J., Higham N. J.: *MATLAB Guide*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA 2000.