Kragujevac J. Sci. 36 (2014) 69-78. UDC 541.27:541.61

COMPUTING GRAPH ENERGY: AN ALTERNATIVE APPROACH

Hooman Fatoorehchi[1], Ivan Gutman[2] and Hossein Abolghasemi[1,3,*]

[1]*Center for Separation Processes Modeling and Nano-Computations,*
School of Chemical Engineering, College of Engineering, University of Tehran,
P. O. Box 11365-4563, Tehran, Iran
[*]Corresponding author
E-mails: hfatoorehchi@ut.ac.ir, abolghasemi.ha@gmail.com

[2]*Faculty of Science, University of Kragujevac,*
P. O. Box 60, 34000 Kragujevac, Republic of Serbia
E-mail: gutman@kg.ac.rs

[3]*Oil and Gas Center of Excellence, University of Tehran, Tehran, Iran*

(*Received January 11, 2014*)

ABSTRACT. The characteristic polynomial corresponding to the adjacency matrix of a graph is obtained by the Faddeev-Leverrier algorithm. Subsequently, an improved numerical scheme, based on the Newton-Raphson algorithm and the Adomian decomposition method, is applied to calculate the energy of the graph (the HMO total π-electron energy of a conjugated molecule). In addition, a nonlinear convergence accelerator, known as the Shanks transform, is employed to speed-up the calculation process. The proposed scheme is conceptually easy, straightforward, and shown to be superior to the classic Newton-Raphson algorithm in terms of computational volume.

Introduction

The energy of a graph is an eigenvalue-based graph invariant that attracted much attention in the contemporary mathematical and mathematico-chemical literature [1]. Its definition originates from an important quantity in the Hückel molecular orbital theory conjugated molecules, namely the total π-electron energy, E_π [2,3]. For a molecule with $n = 2k$ atoms, it can be shown that $E_\pi = 2\sum_{i=1}^{k} \lambda_i$ where $\lambda_i, i = 1, \ldots, k$, denote the top k eigenvalues of the adjacency matrix of the underlying molecular graph of the molecule [2,3]. For most (but not all) chemically relevant graphs, the equality $E_\pi = \sum_{i=1}^{n} |\lambda_i|$ holds. Motivated by this, one of the present authors conceived the *energy* of any graph *G*, defined as [4]

$$E(G) = \sum_{i=1}^{n} |\lambda_i| \,. \tag{1}$$

Details of the theory of graph energy, as well as an exhaustive list of references, can be found in the book [1].

As far as the numerical calculation of graph energy is concerned, the common practice is to use formula (1) and determine the graph eigenvalues by diagonalizing the adjacency matrix of the underlying (molecular) graph. In this paper, we describe an alternative approach for achieving this goal, that requires the calculation of the characteristic polynomial.

By $\phi(x)$ or $\phi(G, x)$ we denote the characteristic polynomial of the graph *G*. It is defined in the standard way as $\phi(x) = \det(xI - A)$, where *A* is the adjacency matrix of *G*, and *I* the unit matrix of appropriate order. Then the graph eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ are the solutions of the equation $\phi(x) = 0$.

Although the calculation of graph energy by calculating $\phi(x)$ and then solving $\phi(x) = 0$ appears to be unhandy from a practical point of view, it has a number of methodological advantages. Namely, as it is well understood [2,5-7], the characteristic polynomial (or more precisely: the coefficients of the characteristic polynomial) contains important and useful information on the structure of (molecular) graphs. Pursuing a matrix-diagonalization-based calculation of graph energy, this information is completely obscured.

The objective of this paper is to introduce a highly efficient combined method to evaluate all the eigenvalues of the adjacency matrix of a considered graph and to determine its energy through the definition given by Eq. (1). As it will be demonstrated, the proposed technique features simplicity, considerable degree of accuracy, and computational speed due to the synergistic effect of combining the classic Newton-Raphson algorithm with the Adomian decomposition method.

The Adomian decomposition method

Assuming the reader is familiar with the Newton-Raphson algorithm, we immediately jump to the basics of the Adomian decomposition method.

In order to illustrate the Adomian decomposition method (ADM), let us consider the following general functional equation:

$$u - N\left(u\right) = f \,, \tag{2}$$

where *N* is a nonlinear operator which maps a Banach space *E* into itself, *f* is a given function and *u* designates an unknown function. The ADM decomposes *u* as an infinite summation, $u = \sum_{i=0}^{\infty} u_i$ and *N* as $N\left(u\right) = \sum_{i=0}^{\infty} A_i$, where A_i are called the Adomian polynomials and can be obtained by the following formula [8]:

$$A_i = A_i\left(u_0, u_1, \ldots, u_i\right) = \frac{1}{i!}\frac{d^i}{d\lambda^i} N\left(\sum_{k=0}^{\infty} u_k \lambda^k\right)\Bigg|_{\lambda=0} . \tag{3}$$

By letting $u_0 = f$, the ADM generates other components of the solution via $u_{i+1} = A_i; \quad i \ge 0$, recursively. The convergence and reliability of this method have been approved by prior works (e.g. see [9,10]).

Elsewhere [11], Fatoorehchi and Abolghasemi have devised a completely different algorithm to generate the Adomian polynomials of any desired nonlinear operators. It is mainly based on string functions and symbolic programming. By setting the symbolic variable $NON = u_0 + u_1 + u_2 + \cdots + u_n$ when *n* is sufficiently large, the following function in MATLAB can return the Adomian polynomial components of a nonlinear operator acting upon *NON*.

Program AdomPoly: An alternative code for calculation of the Adomian polynomials

```
function sol=AdomPoly(expression,nth)
Ch=char(expand(expression));
s=strread(Ch, '%s', 'delimiter', '+');
for i=1:length(s)
t=strread(char(s(i)), '%s', 'delimiter', '*()expUlogsinh');
t=strrep(t,'^','*');
   if length(t)~=2
p=str2num(char(t));
sumindex=sum(p)-p(1);
   else
sumindex=str2num(char(t));
```

```
    end
list(i)=sumindex;
end
A='';
for j=1:length(list)
    if nth==list(j)
A=strcat(A,s(j),'+');
    end
end
N=length(char(A))-1;
F=strcat ('%',num2str(N),'c%n');
sol=sscanf(char(A),F);
```

For more background on the ADM, see the references [12-18].

The Shanks transform

The Shanks transform which is due to Daniel Shanks (1917-1996), is a nonlinear transform that effectively coverts a slowly converging sequence to a rapidly converging sequence [19]. The Shanks transformation $Sh\left(U_n\right)$ of the sequence U_n is defined as

$$Sh\left(U_n\right)=\frac{U_{n+1}U_{n-1}-U_n^2}{U_{n+1}-2U_n+U_{n-1}}. \tag{4}$$

Further speed-up may be achieved by successive implementation of the Shanks transformation, that is $Sh^2\left(U_n\right)=Sh\left(Sh\left(U_n\right)\right)$, $Sh^3\left(U_n\right)=Sh\left(Sh\left(Sh\left(U_n\right)\right)\right)$, etc. For more on application of the Shanks transform one is referred to [20].

The improved Newton–Raphson algorithm

In [21], Abbasbandy put forward a new family of improved numerical equation solvers by the help of the Adomian decomposition method. Because of the space limitation, we do not review his work here and suffice to build our own similar equation solver of higher order of accuracy.

Suppose that we are after the solution of a nonlinear equation,

$$f\left(x\right)=0, \tag{5}$$

with *r* being one of its roots. A fourth-order Taylor's expansion near *x* gives

$$f(x-h)=f\left(x\right)-hf'\left(x\right)+\frac{h^2}{2}f''\left(x\right)-\frac{h^3}{6}f'''\left(x\right)+\frac{h^4}{24}f^{iv}\left(x\right)+\mathrm{O}\left(h^5\right). \tag{6}$$

We are looking for an *h* such that $f(r) = f\left(x-h\right) = 0$. Therefore,

$$f(x-h) = 0 \approx f\left(x\right) - hf'\left(x\right) + \frac{h^2}{2} f''\left(x\right) - \frac{h^3}{6} f'''\left(x\right) + \frac{h^4}{24} f^{iv}\left(x\right). \quad (7)$$

Hence,

$$h = \frac{f\left(x\right)}{f'\left(x\right)} + \frac{h^2}{2}\frac{f''\left(x\right)}{f'\left(x\right)} - \frac{h^3}{6}\frac{f'''\left(x\right)}{f'\left(x\right)} + \frac{h^4}{24}\frac{f^{iv}\left(x\right)}{f'\left(x\right)}. \quad (8)$$

The foregoing equation is nonlinear with respect to *h*. In other words,

$$h = \frac{f\left(x\right)}{f'\left(x\right)} + N\left(h\right) = \frac{f\left(x\right)}{f'\left(x\right)} + \frac{f''\left(x\right)}{2f'\left(x\right)}h^2 - \frac{f'''\left(x\right)}{6f'\left(x\right)}h^3 + \frac{f^{iv}\left(x\right)}{24f'\left(x\right)}h^4. \quad (9)$$

By applying the ADM to Eq. (3), we find $h = \sum_{i=0}^{\infty} h_i$ where,

$$\begin{cases} h_0 = \dfrac{f\left(x\right)}{f'\left(x\right)}, \\ h_{i+1} = A_i, \quad i = 0, 1, 2, \ldots \end{cases} \quad (10)$$

and A_i are the Adomian polynomials decomposing the nonlinear part of Eq. (9).

In this way, a truncated solution for *h* can be obtained as $h \approx \sum_{i=0}^{m} h_i$. Therefore, an iterative relation for the a solution to Eq. (5) is found as,

$$x_{n+1} = x_n - \sum_{i=0}^{m} h_i \Bigg|_{x=x_n}.$$

Now, we let $m = 2$ to develop our relatively accurate solver as,

$$x_{n+1} = x_n - \frac{1}{144}\frac{f\left(x_n\right)}{f'^9\left(x_n\right)}\begin{bmatrix} 144f'^8\left(x_n\right) + 72f''\left(x_n\right)f'^6\left(x_n\right)f\left(x_n\right) - 24f'''\left(x_n\right)f'^5\left(x_n\right)f^2\left(x_n\right) \\ +6f^{iv}\left(x_n\right)f'^4\left(x_n\right)f^3\left(x_n\right) + 72f''^2\left(x_n\right)f'^4\left(x_n\right)f^2\left(x_n\right) \\ -60f'''\left(x_n\right)f''\left(x_n\right)f'^3\left(x_n\right)f^3\left(x_n\right) + 18f^{iv}\left(x_n\right)f''\left(x_n\right)f'^2\left(x_n\right)f^4\left(x_n\right) \\ +12f'''^2\left(x_n\right)f'^2\left(x_n\right)f^4\left(x_n\right) - 7f^{iv}\left(x_n\right)f'''\left(x_n\right)f'\left(x_n\right)f^5\left(x_n\right) \\ +f^{iv2}\left(x_n\right)f^6\left(x_n\right) \end{bmatrix}. \quad (11)$$

As noted above, the Shanks transform can optionally be applied in order to further improve the rate of convergence for the foregoing root:

$$Sh\left(x_n\right) = \frac{x_{n+1}x_{n-1} - x_n^2}{x_{n+1} - 2x_n + x_{n-1}}. \quad (12)$$

Computation of graph energy

Let *G* be a simple graph with *n* vertices and *m* edges. Additionally, suppose that *A* is the adjacency matrix of *G*. Consequently, there exists a characteristic polynomial $\phi\left(x\right)$ of the form

$$\phi\left(x\right)=a_0x^n+a_1x^{n-1}+a_2x^{n-2}+\cdots+a_{n-1}x+a_n\,. \tag{13}$$

The Faddeev–Leverrier scheme has maintained a dependable reputation in providing the characteristic polynomial of matrices among many rivals [22,23]. Quite some time ago, this was recognized in theoretical chemistry, where the method was presented and exemplified in due detail [24-28]. As a curiosity, we note that initially [24,25] the method was erroneously named by Frame, and only in the paper [25] its true mathematical origin was established.

Assuming that Λ is an *n*-by-*n* matrix, the Faddeev–Leverrier algorithm consists of the following steps:

$$\begin{cases}\Lambda_1=\Lambda\\ \Lambda_{i+1}=\Lambda\left(\Lambda_i+a_iI\right); \quad 1\le i\le n\end{cases} \tag{14}$$

$$\begin{cases}a_0=1\\ a_i=-\dfrac{trace\left(\Lambda_i\right)}{i}; \quad 1\le i\le n\end{cases} \tag{15}$$

Through recursion, Eqs. (14) and (15) provide all the $n+1$ coefficients of Eq. (13), or in other words the characteristic polynomial is found conveniently.

It is obvious that the roots of $\phi\left(x\right)=0$ give the *n* eigenvalues of the matrix *A*, i.e., $\lambda_1,\ldots,\lambda_n$. Since *A* is a symmetric matrix with zero trace, it easily follows that all the eigenvalues are real-valued and,

$$\lambda_1+\lambda_2+\cdots+\lambda_n=0 \tag{16}$$

Thus, to evaluate the energy of *G*, we only need to compute $n-1$ roots of the equation $\phi\left(x\right)=0$ through Eq. (11), optionally equipped with Eq. (12), and find the *n*-th root by virtue of Eq. (16). Afterwards, the only easy task is to calculate $E\left(G\right)$ via the definitional Eq. (1).

Numerical examples

For the sake of exemplification, the energies of five graphs with different structures are calculated through the proposed technique as well as by the classical Newton–Raphson algorithm. Details are given in Table 1. As the results in Table 1 indicate, the improved Newton–Raphson algorithm (I.N.R.) is much more computationally efficient than the classic Newton–Raphson algorithm (N.R.). Moreover, the use of the Shanks transform speeds up the convergence to the sought-after root for both the N.R. and the I.N.R. algorithms.

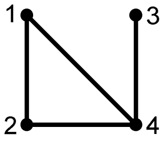
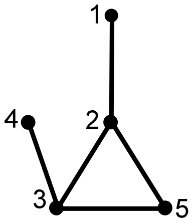
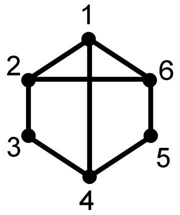
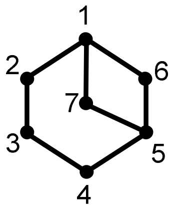
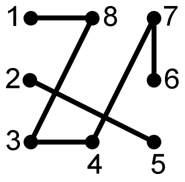
Example	Graph	$E(G)$	CPU time[1] [seconds]		Iteration Number[2]	
			(N.R.)	(I.N.R.)		
			N.R. + Shanks	I.N.R. + Shanks	(N.R.)	(I.N.R.)
1		4.9623886081	0.6050	$<10^{-4}$	60	9
			0.4011	$<10^{-4}$		
2		5.8416192529	28.3440	$<10^{-4}$	38729974	15
			18.2531	$<10^{-4}$		
3		8.1388052268	138.7340	$<10^{-4}$	136932483	16
			98.2487	$<10^{-4}$		
4		9.1152945826	846.9220	0.7500	985650275	21
			567.0015	0.20120		
5		8.9879184148	2857.2156	0.9705	1.1254×10^{9}	28
			2154.6985	0.4210		

Table 1. Results of five numerical examples obtained by the N.R. and the I.N.R. algorithms. [1]The simulations have been performed by a PC with a 2.66 GHz processor and 2.00 GB of RAM; [2]For the same value of the initial guess and a tolerance of 10^{-20}.

Conclusion

A new combined method for the calculation of the energy of a graph was presented. The method incorporates the Faddeev–Leverrier algorithm, the Newton–Raphson algorithm and the Adomian decomposition algorithm, forming a fast and accurate tool for evaluation of all the eigenvalues of the adjacency matrix of the target graph. Based on the comparisons, it was shown that the proposed scheme is superior to the classic Newton–Raphson algorithm, yielding the eigenvalues significantly faster. This advantage would be of interest from a computational point of view, especially for graphs with larger numbers of vertices and edges.

References:

[1] X. Li, Y. Shi, I. Gutman, *Graph Energy,* Springer, New York, 2012.

[2] A. Graovac, I. Gutman, N. Trinajstić, *Topological Approach to the Theory of Conjugated Molecules,* Springer, Berlin, 1977.

[3] I. Gutman, Topology and stability of conjugated hydrocarbons. The dependence of total π-electron energy on molecular topology, *J. Serb. Chem. Soc.* **70** (2005) 441–456.

[4] I. Gutman, The energy of a graph, *Ber. Math. Statist. Sekt. Forschungsz. Graz* **103** (1978) 1–22.

[5] J. R. Dias, *Molecular Orbital Calculations Using Chemical Graph Theory*, Springer, Berlin, 1993.

[6] N. Trinajstić, Computing the characteristic polynomial of a conjugated system using the Sachs theorem, *Croat. Chem. Acta* **49** (1977) 539–633.

[7] I. Gutman, Impact of the Sachs theorem on theoretical chemistry: A participant's testimony, *MATCH Commun. Math. Comput. Chem.* **48** (2003) 17–34.

[8] G. Adomian, *Solving Frontier Problems of Physics: The Decomposition Method,* Kluwer, Dordrecht, 1994.

[9] K. Abbaoui, Y. Cherruault, Convergence of Adomian's method applied to nonlinear equations, *Math. Comput. Model.* **9** (1994) 69–73.

[10] E. Babolian, J. Biazar, On the order of convergence of Adomian method, *Appl. Math. Comput.* **130** (2002) 383–387.

[11] H. Fatoorehchi, H. Abolghasemi, On calculation of Adomian polynomials by MATLAB, J. *Appl. Comput. Sci. Math*. **5** (2011) 85–88.

[12] G. Adomian, A review of the decomposition method and some recent results for nonlinear equations, *Math. Comput. Model.* **13** (1990) 17–43.

[13] H. Fatoorehchi, H. Abolghsemi, Adomian decomposition method to study mass transfer from a horizontal flat plate subject to laminar fluid flow, *Adv. Nat. Appl. Sci.* **5** (2011) 26–33.

[14] H. Fatoorehchi, H. Abolghasemi, A more realistic approach toward the differential equation governing the glass transition phenomenon, *Intermetallics* **32** (2012) 35–38

[15] H. Fatoorehchi, H. Abolghasemi, Improving the differential transform method: A novel technique to obtain the differential transforms of nonlinearities by the Adomian polynomials, *Appl. Math. Model.* **37** (2013) 6008–6017.

[16] H. Fatoorehchi, H. Abolghasemi, R. Rach, An accurate explicit form of the Hankinson–Thomas–Phillips correlation for prediction of the natural gas compressibility factor, *J. Petrol. Sci. Eng.* **117** (2014) 46-53.

[17] R. Rach, A bibliography of the theory and applications of the Adomian decomposition method, 1961-2011, *Kybernetes* **41** (2012) 1087–1148.

[18] B. Kundu, S. Wongwises, A decomposition analysis on convecting–radiating rectangular plate fins for variable thermal conductivity and heat transfer coefficient, *J. Franklin Inst.* **349** (2011) 966–984.

[19] D. Shanks, Nonlinear transformation of divergent and slowly convergent sequences, *J. Math. Phys. Sci.* **34** (1955) 1–42.

[20] A. R. Vahidi, B. Jalalvand, Improving the accuracy of the Adomian decomposition method for solving nonlinear equations, *Appl. Math. Sci.* **6** (2012) 487–497.

[21] S. Abbasbandy, Improving Newton–Raphson method for nonlinear equations by modified Adomian decomposition method, *Appl. Math. Comput.* **145** (2003) 887–893.

[22] A. S. Householder, *The Theory of Matrices in Numerical Analysis*, Blaisdell, New York, 1964.

[23] G. Helmberg, P. Wagner, On Faddeev–Leverrier's method for the computation of the characteristic polynomial of a matrix and of eigenvectors, *Lin. Algebra Appl.* **185** (1993) 219–233.

[24] K. Balasubramanian, The use of Frame's method for the characteristic polynomials of chemical graphs, *Theor. Chim. Acta* **65** (1984) 49–58.

[25] K. Balasubramanian, The use of Frame's method for characteristic polynomial of chemical graphs, *J. Comput. Chem.* **5** (1984) 387–394.

[26] P. Krivka, Ž. Jeričević, N. Trinajstić, On the computation of characteristic polynomial of a chemical graph, *Int. J. Quantum Chem. Quantum Chem. Symp.* **19** (1986) 129–147.

[27] M. Randić, On the evaluation of the characteristic polynomial via symmetric function theory, *J. Math. Chem.* **1** (1987) 145–152.

[28] K. Balasubramanian, Computer generation of characteristic polynomials of edge–weighted graphs, heterographs, and directed graphs, *J. Comput. Chem*. **9** (1988) 204–211.