Computer Physics Communications 182 (2011) 804–807

Contents lists available at ScienceDirect

Computer Physics Communications

www.elsevier.com/locate/cpc

Armchair or Zigzag? A tool for characterizing graphene edge ☆

Abhishek K. Singh [a,b], Evgeni S. Penev [a], Boris I. Yakobson [a,*]

[a] *Department of Mechanical Engineering & Materials Science and Department of Chemistry, Rice University, Houston, TX 77005, United States*
[b] *Materials Research Centre, Indian Institute of Science, Bangalore 560012, India*

ARTICLE INFO

Article history:
Received 10 November 2010
Accepted 19 November 2010
Available online 4 December 2010

Keywords:
Honeycomb lattice
Graphene
Edge atom types
Fortran 90/95

ABSTRACT

Electronic, magnetic, and structural properties of graphene flakes depend sensitively upon the type of edge atoms. We present a simple software tool for determining the type of edge atoms in a honeycomb lattice. The algorithm is based on nearest neighbor counting. Whether an edge atom is of armchair or zigzag type is decided by the unique pattern of its nearest neighbors. Particular attention is paid to the practical aspects of using the tool, as additional features such as extracting out the edges from the lattice could help in analyzing images from transmission microscopy or other experimental probes. Ultimately, the tool in combination with density-functional theory or tight-binding method can also be helpful in correlating the properties of graphene flakes with the different armchair-to-zigzag ratios.

Program summary

Program title: edgecount
Catalogue identifier: AEIA_v1_0
Program summary URL: http://cpc.cs.qub.ac.uk/summaries/AEIA_v1_0.html
Program obtainable from: CPC Program Library, Queen's University, Belfast, N. Ireland
Licensing provisions: Standard CPC licence, http://cpc.cs.qub.ac.uk/licence/licence.html
No. of lines in distributed program, including test data, etc.: 66 685
No. of bytes in distributed program, including test data, etc.: 485 381
Distribution format: tar.gz
Programming language: Fortran 90/95
Computer: Most UNIX-based platforms
Operating system: Linux, Mac OS
Classification: 16.1, 7.8
Nature of problem: Detection and classification of edge atoms in a finite patch of honeycomb lattice.
Solution method: Build nearest neighbor (NN) list; assign types to edge atoms on the basis of their NN pattern.
Running time: Typically ∼second(s) for all examples.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

The current research activity [1–6] on characterizing finite graphene structures, such as *nanoribbons* and *flakes*, and especially their edge morphology is so intense that perhaps even Pauli would have added “God made graphene; its edges were invented by the devil” [7]. The diabolic nature of edges in reduced dimensionality becomes more acute and for small enough systems is an ultimate factor determining the overall system properties. Structural, mechanical, electronic and magnetic properties of finite size graphene is solely determined by the edge atoms. Indeed, theoretically, the effect of edge morphology of graphene nanoribbons, nanoroads, quantum dots etc. on its properties has been well quantified [8–10], and recently experimentally verified [3]. Further experimental probes have revealed intriguing preference of a particular edge type (zigzag vs. armchair) [4,5]. Clearly, any discussion of graphene structure requires a reliable scheme for classification of edge atoms. The existence of pure zigzag and armchair directions belongs to the trivia about honeycomb lattice.

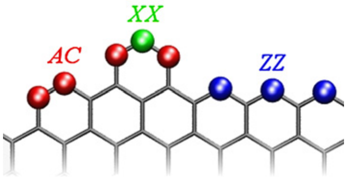
☆ This paper and its associated computer program are available via the Computer Physics Communications homepage on ScienceDirect (http://www.sciencedirect.com/science/journal/00104655).

* Corresponding author.
E-mail addresses: abhishek@mrc.iisc.ernet.in (A.K. Singh), epenev@rice.edu (E.S. Penev), biy@rice.edu (B.I. Yakobson).

0010-4655/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.cpc.2010.11.029

Algorithm 1 Edge detection and characterization.

```
initialize run                                          ▷ Parse command-line; allocate containers
build nearest neighbor list $nn$                        ▷ simple $O(N^2)$ counting
find edge lattice points
for $i \leftarrow 1, nedge$ do                          ▷ Assign edge types
    if one has 2 && other has 3 neighbors then
        $etyp_i \leftarrow AC$
    else if both neighbors have 3 neighbors each then
        $etyp_i \leftarrow ZZ$
    else if both neighbors have 2 neighbors each then   ▷ Graphical illustration of type assignment:
        cannot assign type
        $etyp_i \leftarrow XX$
    end if
    if requested then
        output $nn(i)$ nearest neighbors of $i$
    end if
end for
finalize simulation
```



Often, however, one needs to characterize dynamically evolving edge/interface morphology or irregularly shaped flakes which may not be a straightforward task. The type of graphene edge is of generic interest and relates to problems in lattice theory [11] and statistics of random walks on honeycomb lattice [12].

In Section 2 we describe a simple algorithm for assigning types to graphene edge atoms and then, in Section 3, introduce the EDGECOUNT software tool that implements it. Finally, in Section 4, some examples and practical issues related to the code usage are discussed.

2. Detection and classification of edge atoms in a truncated honeycomb lattice

The algorithm for assigning types to edge atoms is based on nearest-neighbor counting and is given in Algorithm 1. In brief, an edge atom with two nearest neighbor atoms, (i) which themselves have three nearest neighbors, is assigned a type *zigzag*, or (ii) one of them has two and another has three nearest atoms, is assigned a type *armchair*, or (iii) which themselves have two neighbors, are considered "unassigned" atoms. The computational burden of such an algorithm is almost exclusively localized in the nearest-neighbor list building step: counting the number of all atoms $nn(i)$ within some distance cutoff r_{cut} from atom i. Here this is implemented in a straightforward way as the expected application of the code is to analyze non-periodic systems, as is any finite patch of a hexagonal lattice. Therefore, the overall scaling of the algorithm is $O(N^2)$ for a system of N atoms. A minor optimization is attempted by introducing a cap nn_{max} on the nearest neighbor atom count in the inner loop over the atoms, thus skipping the rest of the loop for given atom i if the newly incremented $nn(i)$ leads to $nn(i) = nn_{\mathrm{max}}$.

3. Program description

EDGECOUNT is written in FORTRAN 90/95 and implements Algorithm 1. It operates as a typical command-line tool and its general synopsis for usage is

```
edgecount [OPTIONS] [FILE]
```

The present version of the tool assumes that the `FILE` contains the input geometry in the standard single-frame xyz file format:

input.xyz

```
<number_of_atoms>
comment_line
atom_symbol_1 x-coord_1 y-coord_1 z-coord_1
atom_symbol_2 x-coord_2 y-coord_2 z-coord_2
...
atom_symbol_N x-coord_N y-coord_N z-coord_N
```

Command-line options are provided for requesting more detailed help (`-h`), including in the output also nearest neighbors of the edge atoms (`-n`), and a verbose mode of operation (`-v`).

All output generated by EDGECOUNT is stored in three files in the current working directory: output.log, output.xyz, and edge.xyz.

3.1. *output.log*

This file contains indices of all edge atoms identified by the tool along with the assigned type. Basic statistics including the total number of armchair, zigzag, unassigned edge atoms, and the ratio of armchair to the total number of edge atoms with assigned type, is appended at the end of the file. In verbose mode this simple statistics is also duplicated to the standard output (display).

3.2. *output.xyz*

This output file is an extended version of the input .xyz geometry. Every coordinate record is extended to include the number of nearest neighbors of atom i, nn_i, followed by nn_i fields giving the atomic indices of the nearest neighbor atoms in the input geometry.

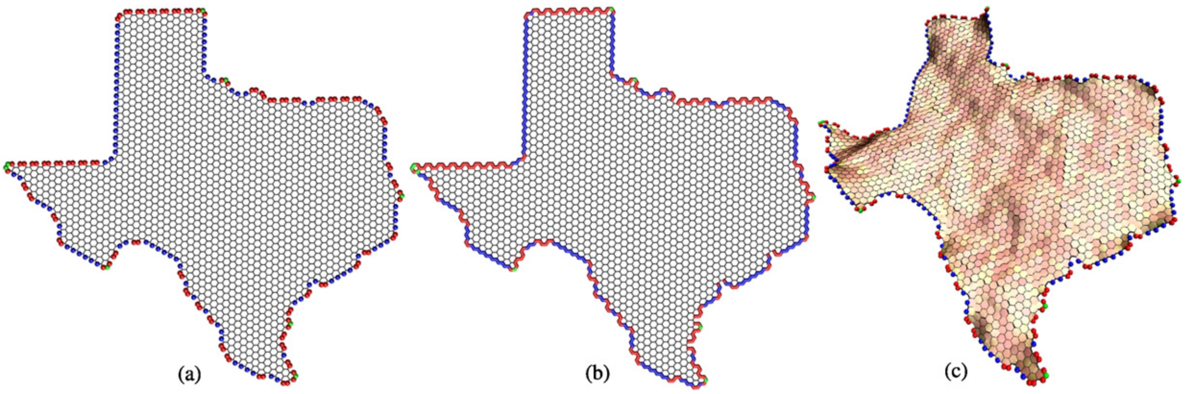


Fig. 1. Test system: a Texas-shaped graphene flake; input geometry texagon.xyz. (a) Edge atoms as identified by EDGECOUNT and output in edge.xyz; atomic types are color coded (cf. Algorithm 1) and image is over imposed on the initial geometry. (b) Edge "contour" consisting of edge atoms and their nearest neighbors. (c) Same as (a), but for a nonplanar geometry ("wrinkled texagon"). All images are generated with the VMD software [13]. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

output.xyz

```
<number_of_atoms>
comment_line
atom_symbol_1 x-coord_1 y-coord_1 z-coord_1 nn_1 index_1 ... index_nn_1
atom_symbol_2 x-coord_2 y-coord_2 z-coord_2 nn_2 index_1 ... index_nn_2
...
atom_symbol_N x-coord_N y-coord_N z-coord_N nn_N index_1 ... index_nn_N
```

3.3. *edge.xyz*

This output file contains all identified edge atoms in an extended .xyz format, similar to output.xyz. The input atomic symbol is replaced with a two-character generic identifier which is AC for an edge *armchair* atom, ZZ for an edge *zigzag* atom, and XX otherwise (unassigned atoms).

In the code distribution we have also provided a few simple utilities as *Mathematica* notebooks [14] which, given a source geometry, can be used to construct random flakes [15] as well as hexagonal patches with edges of a specified orientation, as those discussed in the examples below.

4. Examples

Consider the patch of a honeycomb lattice, shown in Fig. 1(a), consisting of $N = 3383$ atoms, and available as texagon.xyz. To analyze the edge structure of such a Texas-shaped graphene flake one can simply execute on the command line

```
edgecount texagon.xyz
```

The identified edge atoms with the corresponding edge types can be easily visualized along with the initial geometry as shown in Fig. 1(a). From the output protocol file output.log one can determine the assigned atomic types. It is straightforward to see, by inspecting the last record in output.log, that along the edge of such a flake the armchair atoms are slightly prevailing for the so constructed shape, with $n_{\mathrm{ac}} = N_{\mathrm{ac}}/(N_{\mathrm{ac}} + N_{\mathrm{zz}}) \simeq 0.545$. It should be noted, however, that this fraction depends sensitively on the orientation and size of the underlying hexagonal lattice. Note also that the leftmost ("El Paso") corner is not assigned a type as it is practically a single protruded hexagon. Furthermore, all "bays" along the "Gulf of Mexico" contain fully coordinated atoms which result in gaps along the edge contour, Fig. 1(b). The EDGECOUNT tool can in principle be applied also to the case of a nonplanar hexagonal lattice, as shown in Fig. 1(c).

As another example, we use the EDGECOUNT to characterize the edge of circular graphene flakes as a function of their size which is also suitable for testing the code robustness and benchmark its efficiency in terms of execution time. From simple geometry arguments one can easily evaluate the densities of AC, $n_{\mathrm{ac}}(\theta)$, and ZZ atoms, $n_{\mathrm{zz}}(\theta)$, along a given direction defined by the angle θ relative to the zigzag direction [16]:

$$n_{\mathrm{ac}}(\theta) = \frac{4}{\sqrt{3}} \sin\theta, \qquad n_{\mathrm{zz}}(\theta) = 2\sin(\pi/6 - \theta). \tag{1}$$

For a random path on a honeycomb lattice, averaging the above expressions over the physically essential interval $0 \leqslant \theta \leqslant \pi/6$, where $\sin\theta = \sin(\pi/6 - \theta)$, one finds

$$\overline{n_{\mathrm{ac}}}/\overline{n_{\mathrm{zz}}} = 2/\sqrt{3} \tag{2}$$

$$\frac{\overline{n_{\mathrm{ac}}}}{\overline{n_{\mathrm{ac}}} + \overline{n_{\mathrm{zz}}}} = \frac{2}{2+\sqrt{3}} \equiv 0.53589838\ldots \tag{3}$$

We have used EDGECOUNT to calculate the fraction of AC atoms for circular flakes of up to 3×10^5 atoms and result is displayed in Fig. 2. Indeed, as expected, for increasing flake radius the actual fraction tends to the limit value $2/(2+\sqrt{3})$, Eq. (3).

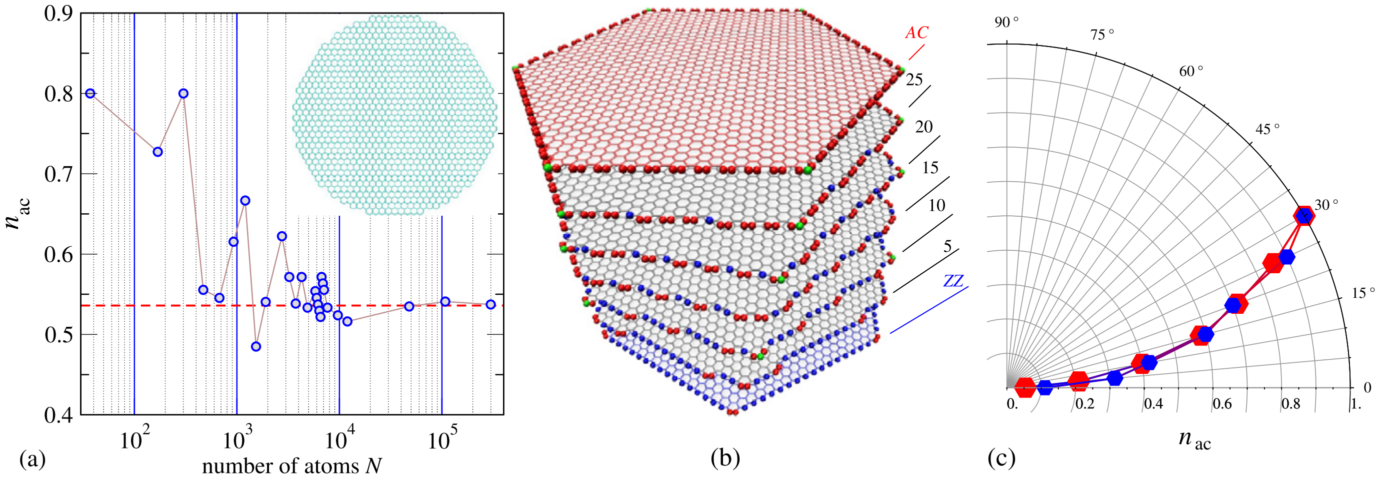


Fig. 2. (a) Fraction of armchair edge atoms n_{ac} for circular graphene flakes (inset) as a function of their size N. The horizontal dashed line shows the average fraction $2/(2+\sqrt{3})$, Eq. (3); (b) Geometries of the smaller-size hexagons ($N \simeq 2000$). (c) Polar plot of n_{ac} for two sets of different size hexagons with edges converted from ZZ to AC in step of 5°. Smaller symbols are for the smaller-size hexagons and the bigger symbols are for the larger-size set.

As a last example we consider the edge of a *hexagonal* flake which represents an extreme and peculiar case: Changing the orientation of the underlying hexagonal lattice relative to the edge lines from ZZ to AC direction the n_{ac} fraction can sweep almost the entire interval $[0, 1]$. We have constructed a set of seven hexagons cut out from a circular graphene flake of $N = 10^4$ atoms. The edge is transformed from ZZ to AC in steps of 5°, Fig. 2(b). Two sets of different sizes were used, large ($N \simeq 8200$) and small ($N \simeq 2000$) hexagons, and n_{ac} as obtained with EDGECOUNT is shown in the polar plot, Fig. 2(c). Since the corners of a ZZ hexagon are 12 AC atoms, therefore $n_{ac}(0)$ depends on the size of the hexagon. On the other hand, the corners of a purely AC hexagon are atoms of "unassigned" type and therefore $n_{ac}(30°) = 1$ for any size.

5. Summary

A tool to determine the edge type in a honeycomb lattice has been developed. An accurate determination of type of edge, in turn, enables the correct prediction of structural, electronic, and magnetic properties of such materials. The algorithm works on the pattern of the nearest neighbors, which determines the type of the edges. The software can be helpful [17] in analyzing the edges of any material which has a honeycomb lattice, such as graphene, h-BN, etc., with most straightforward usage for non-periodic simply-connected geometries. It is expected, however, to be also applicable with some provisions to multiply connected systems (e.g., graphene "anti-dots" or rings) as well as systems represented by supercells with applied periodic boundary conditions.

Acknowledgements

E.P. would like to acknowledge discussions with Thomas Oheix. This work was supported by the Office of Naval Research (MURI project) and by the National Science Foundation (NIRT, Award 0708096).

References

[1] K. Nakada, M. Fujita, G. Dresselhaus, M.S. Dresselhaus, Edge state in graphene ribbons: Nanometer size effect and edge shape dependence, Phys. Rev. B 54 (1996) 17954–17961.
[2] P. Koskinen, S. Malola, H. Häkkinen, Self-passivating edge reconstructions of graphene, Phys. Rev. Lett. 101 (2008) 115502.
[3] K.A. Ritter, J.W. Lyding, The influence of edge structure on the electronic properties of graphene quantum dots and nanoribbons, Nature Mater. 8 (2009) 235–242.
[4] X. Jia, M. Hofmann, V. Meunier, B.G. Sumpter, J. Campos-Delgado, J.M. Romo-Herrera, H. Son, Y.-P. Hsieh, A. Reina, J. Kong, M. Terrones, M.S. Dresselhaus, Controlled formation of sharp zigzag and armchair edges in graphitic nanoribbons, Science 323 (2009) 1701–1705.
[5] Ç.Ö. Girit, J.C. Meyer, R. Erni, M.D. Rossell, C. Kisielowski, L. Yang, C.-H. Park, M.F. Crommie, M.L. Cohen, S.G. Louie, A. Zettl, Graphene at the edge: Stability and dynamics, Science 323 (2009) 1705–1708.
[6] G.M. Rutter, N.P. Guisinger, J.N. Crain, P.N. First, J.A. Stroscio, Edge structure of epitaxial graphene islands, Phys. Rev. B 81 (2010) 245408.
[7] Paraphrasing the famous sentence "God made the bulk; surfaces were invented by the devil." credited to Wolfgang Pauli – see, e.g. A.D. Yofe, Contemp. Phys. 29 (1988) 411–414.
[8] A.K. Singh, B.I. Yakobson, Electronics and magnetism of patterned graphene nanoroads, Nano Lett. 9 (2009) 1540–1543.
[9] T. Wassmann, A.P. Seitsonen, A.M. Saitta, M. Lazzeri, F. Mauri, Clar's theory, π-electron distribution, and geometry of graphene nanoribbons, J. Am. Chem. Soc. 132 (2010) 3440–3451.
[10] A.K. Singh, E.S. Penev, B.I. Yakobson, Vacancy clusters in graphane as quantum dots, ACS Nano 4 (2010) 3510–3514.
[11] J.A. De Loera, R. Hemmecke, J. Tauzer, R. Yoshida, Effective lattice point counting in rational convex polytopes, J. Symb. Comput. 38 (2004) 1273–1302, http://www.math.ucdavis.edu/~latte/.
[12] B. Vidakovic, All roads lead to Rome-even in the honeycomb world, Am. Stat. 48 (1994) 234–236.
[13] W. Humphrey, A. Dalke, K. Schulten, VMD – Visual molecular dynamics, J. Mol. Graph. 14 (1996) 33–38, http://www.ks.uiuc.edu/Research/vmd/.
[14] Wolfram Research, Inc., Mathematica, version 7.0, Champaign, IL, 2008.
[15] T. Espinosa-Ortega, I.A. Luk'yanchuk, Y.G. Rubo, Density of states in randomly shaped graphene quantum dots, Superlattices and Microstructures (2010), doi:10.1016/j.spmi.2010.06.018.
[16] Y. Liu, A. Dobrinsky, B.I. Yakobson, Graphene edge from armchair to zigzag: the origins of nanotube chirality? Phys. Rev. Lett. 105 (2010) 235502.
[17] N. Barnes, Publish your computer code: It is good enough, Nature 467 (2010) 753.