Genetic programming approach for electron- alkali-metal atom collisions

Salah Yaseen El-Bakry[1] and Amr Radi[2]

[1]Physics Department, Faculty of Science, Taibah University, Madinah Munawwarah, P.O.Box 344, Kingdom of Saudi Arabia.

[2]Physics Department, Faculty of Science, Ain Shams University, Abbassia, Cairo, Egypt

Email: elbakry_salah@Yahoo.com & Email: Amr_Radi@hotmail.com

Abstract: **New technique is presented for modeling the total cross sections of electron scattering by Na, K, Rb and Cs atoms in the low and intermediate energy regions. The calculations have been performed in the framework of genetic programming (GP) technique. The GP has been running based on the experimental data of the total collisional cross sections to produce the total cross sections for each target atom. The incident energy and atomic number as well as the static dipole polarizability have been used as input variables to find the functions that describe the total collisional cross sections of the scattering of electrons by alkali atoms. The experimental, calculated and predicted total collisional cross sections are compared. The discovered functions show a good match to the experimental data.**

Keywords: Genetic Programming, electron scattering, alkali atoms, total cross sections, Dipole Polarizability.

PACS number(s): 84.35.ti, 34.80.Dp, 34.85.tx

1. Introduction

Over the last three decades, there has been considerable interest in the electron-atom collision problem. Most of this work has concentrated on elastic scattering, excitation of the lowest energy states, excitation of higher lying states and ionization (see e.g. Refs.[1-5]). The investigations of the previous scattering problems have been carried out by employing numerous traditional approximate methods (see e.g. Refs.[6-10]). Alkali-metal atoms have been a subject of interest in several theoretical and experimental investigations in electron-atom collisions because of their various interesting properties: relatively simple structure, low ionization potentials (3.9-5.4eV), large polarizabilities [11,12] and existence of resonance lines in the visible or quartz ultraviolet part of the electromagnetic spectrum (which make the alkali metals interesting as components of stellar atmosphere and other plasmas). The total collisional cross sections have been measured or estimated by different experimental research groups [13-18] for electron impact on these atoms. Neural networks have become famous in the field of electron collisions with atoms [19]. Recently, Genetic programming (GP) has been one of researchers' interests in modeling of high energy physics as well as in different fields (see for example Refs. [20-25]) . Genetic programming is one of a number of machine learning techniques in which a computer program is given the elements of possible solutions to the problem (in our case energy, atomic number and polarizability). This technique, through a feedback mechanism, attempts to discover the best solution (in our case it will be a function) to the problem at hand, based on the programmers definition of success. The Genetic programming framework creates a program which consists of a series of linked nodes. Each node takes a number of arguments and supplies a single return value. There are two general types of nodes: functions (or operators) and terminals (variables and constants) [23]. The series of linked nodes can be represented as a tree where the leaves of the tree represent terminals and operators reside at the forks of the tree. In another words, in GP the programs are written as function which represented in expression trees. The tree elements are called nodes. The functions (F) have one or more inputs and produce a single output value. These provide the internal nodes in expression trees. The terminals (T) represent external inputs, constants and zero argument functions

In this paper, we have discovered the functions that describe the scattering of electrons by sodium, potassium, rubidium and cesium atoms using genetic programming technique. GP is fed once with the electron incident energy and the target atomic number (this case is referred to as "model 1") and once with the incident energy and the static dipole polarizability [12] of the alkali target atom (this case is referred to as "model 2") so that the outputs imitate the experimental data of the total collisional cross sections. To our knowledge, this is the first application of the genetic programming technique to the low and intermediate energy data of electron scattering by alkali atoms. Sections 2 and 3 of this paper deals with the introduction and the inputs of GP as well as the proposed

GP. Finally, in Sections 4 and 5, we present our results and conclusion.

2. Electron - alkali atom collision modeling using genetic programming

GP, evolves a population of computer programs, which are possible solution to a given optimization problem , using the Darwinian principle of survival of the fittest. It uses biologically inspired operations like reproduction, crossover and mutation. Each program or individual on the population is generally represented as a tree composed of functions (*,+) and data / terminals (X,Y) appropriate to the problem domain. For example, Fig. 1 shows the representation of the function +(*(x,y),*(x,*(x,y)))i.e.((x*(x*y))+(x*y)). To read trees in this fashion, one resolves the sub-trees in a bottom-up fashion, where F={*,+} and T={x, y}. The set of functions and set of terminals/inputs must satisfy the closure and sufficiency properties. The closure property demands that the function set is well defined and closed for any combination of arguments that it may encounter. On the other hand, the sufficiency property requires that the set of functions and the set of terminals be able to express a solution of problem. The function set may contain standard arithmetic operators, mathematical functions, logical operators, and domain-specific functions. The terminal set usually consists of feature variables and constants. Each individual in the population is assigned a fitness value, which quantifies how well it performs in the problem environment. The fitness value is computed by a problem dependent fitness function.

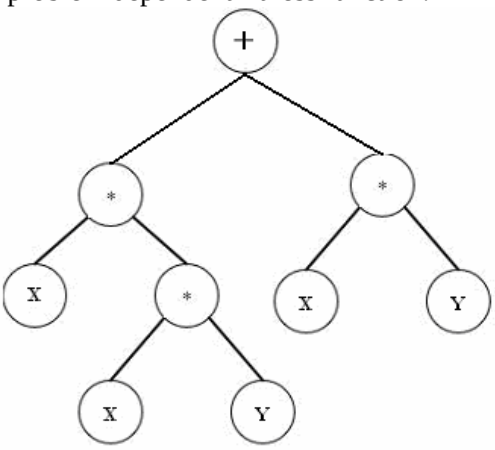

Fig.1. Tree representation of the equation +(*(x,y),*(x,*(x,y))) i.e. ((x*(x*y))+(x*y)).

A typical implementation of GP (i.e. the process of determining the best (or nearly best) solution to a problem in GP) involves the following steps:

1) GP begins with a randomly generated initial population of solutions.

2) A fitness value is assigned to each solution of the populations

3) A genetic operator is selected probabilistically.

Case i) If it is the reproduction operator, then an individual is selected (we use fitness proportion-based selection) from the current population and it is copied into the new population. Reproduction replicates the principle of natural selection and survival of the fittest.

Case ii) If it is the crossover operator, then two individuals are selected. We use tournament selection where number of individuals is taken randomly from the current population, and out of these, the best two individuals (in terms of fitness value) are chosen for the crossover operation. Then, we randomly select a sub tree from each of the selected individuals and interchange these two sub-trees. These two offspring are included in the new population. Crossover plays a vital role in the evolutionary process.

Case iii) if the selected operator is mutation, then a solution is (randomly) selected. Now; a sub-tree of the selected individual is randomly selected and replaced by a new randomly generated sub-tree. This mutated solution is allowed to survive in the new population. Mutation maintains diversity.

4) Continue step 3), until the new population gets solutions. This completes one generation.

5)Unlike genetic algorithm [26], GP will not converge. So, steps 2)-4) are repeated till a desired solution (may be 100% correct solution) is achieved. Other–wise, terminate the GP operation after a predefined number of generations.

3.The proposed genetic programming

Our approach is to use the experimental data of the total collisional cross sections at certain values of the incident energy of the electron, atomic number and the static dipole polarizability of the alkali target atom to produce the total cross sections (calculated) for each case (target atom). The alkali-metal atoms are highly polarizable targets. Therefore, reliable estimates of the effect of distortion of the alkali-metal target is rather essential to predict scattering parameters. Modeling of the experimental data has two fold way. The incident energy (ε) and atomic number (z) are used as input variables to find the suitable function , $\sigma_{t1}(\varepsilon, z)$, that describes the experimental data (this case is referred to as " model 1"). Also, the incident energy (ε) and static dipole polarizability (α) are used as input variables to find another suitable function, $\sigma_{t2}(\varepsilon, \alpha)$, that describes the available experimental data (this case is referred to as "model 2").

Our representation, the fitness function is calculated as a negative value of the total absolute performance error of the discovered function on the experimental data set, i.e. a lower error must correspond to a higher fitness. The total performance error can be defined for all the experimental data (i = 1 …, n) set as:

$$E = \sum_{j=1}^{n} \left| X_j - Y_j \right|^2 \qquad (1)$$

where X_j represent the experimental data for element j and Y_j represent the calculated data for element j. The running process stops when the error E is reduced to an acceptable level (0.00001).

The training data set which is used based on experimental data for the total scattering cross sections of the collisions of electrons with Na, K, Rb and Cs atoms [16-18].

To find $\sigma_{t1}(\varepsilon, z)$, according to "model 1", GP was run for 800 generations with a maximum population size of 1000. The operators (and selection probability) were: crossover with probability 0.9 and mutation with probability 0.01. The function set is {+,-, *, \, log, Log_2, sin, cos, sqr}, and the terminal set is {random constant from 0 to 10, the incident energy, the atomic number}. the``full'' initialisation method was used with an initial maximum depth of 27, and tournament selection with a tournament size of 8. The GP was run until the fitness function is reduced to an acceptable level (0.00001); once for each alkali-metal atom. The discovered function has been tested to associate the input patterns to the target output patterns using the error function.

To find $\sigma_{t2}(\varepsilon, \alpha)$, according to "model 2", GP has been run with the same previous conditions except the function set becomes {+,-, *, \, log, sqr, exp}, and the terminal set is {random constant from 0 to 10, the incident energy, the static dipole polarizability}.

The final discovered function $\sigma_{t1}(\varepsilon, z)$ for describing the electron collisions with alkali atoms at low and intermediate energies is (see also the appendix) given by

$\sigma_{t1}(\varepsilon, z) =$

$Log_2(z*(10-\log(\sin(z)/Log_2(z)))+(\varepsilon^2)*5.1343/(\sin(z)*Log_2(\varepsilon)))+(18.3816+z*1.1727/\varepsilon-(0.5147/(\cos(\sin(\sin(Log_2(z/0.93534)))/0.27459)))+Log_2(z)/0.0536)*\cos(0.3010*\log(\varepsilon)),$

(2)

where ε is the electron incident energy and z is the atomic number of the alkali target atom.

Also, the discovered function $\sigma_{t2}(\varepsilon, \alpha)$ is given by

$\sigma_{t2}(\varepsilon, \alpha) =$

$(7.2814-\log(0.1316*(\varepsilon)\text{^}1.5))*\sqrt{\log((\exp(\alpha)-2.0336)/(10+\sqrt{\alpha}+a))},$

(3)

such that

a=sqrt(0.9406+exp(ε*(7.2814-log(0.018401 *sqrt(ε)*exp(log(0.2231 *sqrt(ε)))*log(α *sqrt(exp(ε))-ε))))),

(4)

where ε is the incident energy and α is the static dipole polarizability of the alkali target atom.

4. Results

The discovered functions were tested using the experimental data of the total cross sections (measured in πa_0^2 units).The training data is based on experimental observations at incident energies ranging from 4.1 eV to 76.1 eV for e^--Na scattering [16], from 4.4 eV to 101.9 eV for e^--K scattering[16], from 2 eV to 77.5 eV for e^- - Rb scattering [17]. After convergence, the discovered functions $\sigma_{t1}(\varepsilon, z)$ and $\sigma_{t2}(\varepsilon, \alpha)$ have been used to predict the total collisional cross sections from 6.5 eV to 77.5 eV for e^- Cs scattering [18]. The values of the static dipole polarizabilities of Na, K, Rb and Cs are taken [12] as 163 a_0^3, 293 a_0^3, 319 a_0^3 and 358 a_0^3, respectively. Figure 2 displays a good match between the experimental data and the calculated (for Na, K and Rb) and predicted (for Cs only) total collisional cross sections of electrons with alkali atoms employing our discovered function $\sigma_{t1}(\varepsilon, z)$ (i.e. according to "model 1"). Figure 3 shows a best fitting between the experimental, calculated and predicted total collisional cross sections using our discovered function $\sigma_{t2}(\varepsilon, \alpha)$ (i.e. according to "model 2 ").

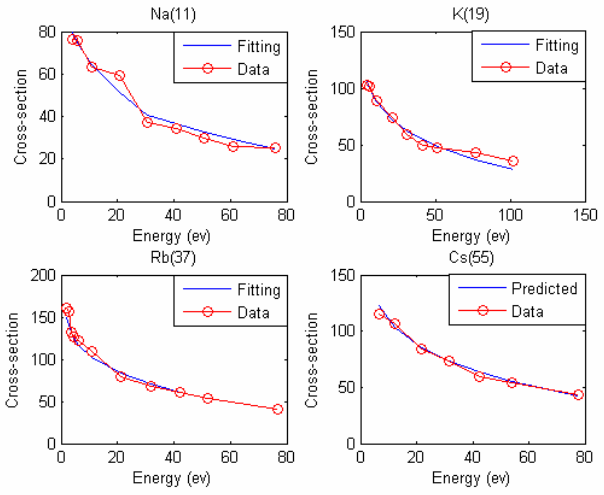

Fig.2. Comparison between the experimental, calculated and predicted total collisional cross sections (in πa_0^2) of e^- - Na, e^- - K, e^- - Rb and e^- - Cs scattering using our discovered function $\sigma_{t1}(\varepsilon, z)$.

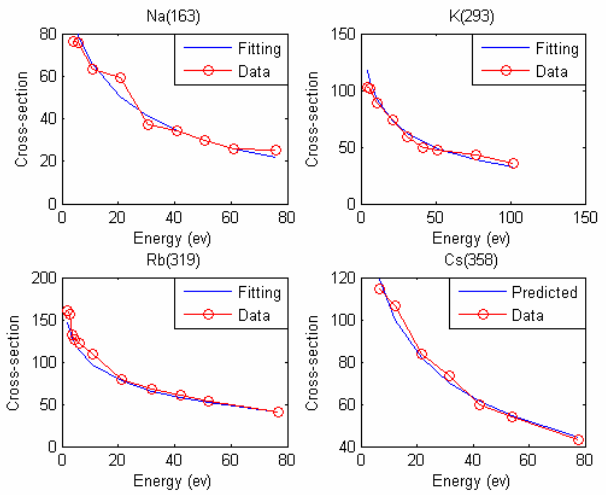

Fig.3. Comparison between the experimental, calculated and predicted total collisional cross sections (in πa_0^2) of e^- - Na, e^- - K, e^- - Rb and e^- - Cs scattering using our discovered function $\sigma_{t2}(\varepsilon,\alpha)$.

5. Conclusion

Genetic programming (GP) method is one of a number of machine learning techniques in which a computer program is given the elements of possible solution to the problem (in our case electron incident energy, atomic number, static dipole polarizability) and attempts, through a feedback mechanism, to discover the best function (in our case the total collisional cross section) to the problem at hand, based on the programmers definition of success. The program consists of a series of linked nodes which can be represented as a tree. Each node takes a number of arguments and supplies a single return value. There are two general types of nodes (or tree elements): functions(or operators) such as {*,+,-,\ , exp, log, Log_2, sin, cos, sqr}and terminals (constants and variables) such as { random constant from 0 to 10, the incident energy, the atomic number, polarizability}.The GP model seeks to imitate the biological processes of evolution, treating a tree or program as an "organism". Through natural selection and reproduction over a number of generations, the fitness of a population of organisms is improved. In our problem,GP method uses the experimental data of the total collisional cross sections at certain values of the electron incident energy, the atomic number and the static dipole polarizability of the alkali target atom to produce the total cross sections (calculated) for each alkali target atom.

Genetic programming has been run to model the electron-alkali-metal atom collisions at low and intermediate energies. Therefore, the present work presents a new technique for modeling the total collisional cross sections of electrons with sodium , potassium, rubidium and cesium alkali atoms based on GP technique. GP discovered the functions that describe the total collisional cross sections of electrons with alkali atoms in terms of the electron incident energy, atomic number and static dipole polarizability. The discovered functions show a good match to the experimental data. Moreover, the discovered functions are capable of predicting experimental data for the total collisional cross sections that are not used in the training session. Finally, we conclude that GP has become one of important research areas in the field of atomic collision physics.

Appendix

In this appendix we present the form of the discovered functions $\sigma_{t1}(\varepsilon,z)$ and $\sigma_{t2}(\varepsilon,\alpha)$ for describing electron collisions with alkali-metal atoms at low and intermediate energies in terms of the electron incident energy (ε) and atomic number (z) as well as in terms of the static dipole polarizability α. The discovered functions are given as:

$\sigma_{t1}(\varepsilon,z)=$

-(Log_2(+(*(/(*(Log_2(10),10),/(sin(z),/(*(z,z), Log_2(ε)))),sin(0.15518)),*(-(10,log(/(sin(z), Log_2(z)))),z))),*(/(+(+(/(0.63797,/(ε,z)),+(/(0.63797,/(cos(/(sin(sin(Log_2(/(z,0.93534)))),0.27459)),cos(+(0.35818,*(Log_2(10),0.64761))))),/(Log_2(z),/(0.98577,10)))),10),sin(10)),cos(/(log(ε), Log_2(10))))).

(A1)

Or in simpler form we can rewrite the previous form of $\sigma_{t1}(\varepsilon,z)$ as follows

$\sigma_{t1}(\varepsilon,z)=$

Log_2(z*(10-log(sin(z)/Log_2(z)))+(ε^2)*5.1343/(sin(z)*Log_2(ε)))+(18.3816+z*1.1727/ε-(0.5147/(cos(sin(sin(Log_2(z/0.93534)))/0.27459)))+Log_2(z)/0.0536)*cos(0.3010*log(ε).

(A2)

Also, the discovered function $\sigma_{t2}(\varepsilon,\alpha)$ is given by

$\sigma_{t2}(\varepsilon,\alpha)=$

(-(exp(log((log(10),sqrt(10)))),log(*(sqrt(*(ε,/(0.18401,10))),*(sqrt(sqrt(0.88473)), ε)))),sqrt(log(/(-(exp(α),exp(0.7098)),plus(plus(10,sqrt(plus(sqrt(0.88473),exp(*(-(exp(log(*(log(10),sqrt(10)))),log(*(sqrt(*(ε,/(0.18401,10))),*(exp(log(/(*(sqrt(ε),0.057973),0.25984))),log(-(*(sqrt(exp(ε)), α), ε))))))), ε))))),sqrt(α))))).

(A3)

Or in simpler form we can also rewrite the formula (A3) as follows

$\sigma_{t2}(\varepsilon,\alpha)$=
(7.2814-log(0.1316*(ε)^1.5))*sqrt(log((
exp(α)-2.0336)/(10+sqrt(α)+a))), (A4)

where
a=sqrt(0.9406+exp(ε *(7.2814-log(0.01
8401 *sqrt(ε)*exp(log(0.2231
*sqrt(ε)))*log(α *sqrt(exp(ε))- ε))))). (A5)

References

[1] T T Gien, *Chem.Phys.Lett.***139** 23 (1987); Phys.Rev.**A44** 5693 (1991); ibid.**9** 3203 (1976); Phys.Rev.**A16** 123 (1977); ibid.**16** 1736 (1977)

[2] K C Mathur and S P Purohit, Phys. Rev. **A42** 2696 (1990)

[3] V E Bubelev, D H Madison and M A Pinkerton, *J. Phys.* **B29** 1751 (1996)

[4] V Zeman, R P McEachran and A D Stauffer, Eur. Phys.J. **D1** 117 (1998)

[5] S P Khare and P Shobha ,J.Phys.**B4** 208 (1971)

[6] I Bray, D A Konovalov and I E McCarthy *Phys. Rev.* A44 5586 (1991); *Phys. Rev.* **A44** 7179 (1991)

[7] I Bray and IE McCarthy *Phys. Rev.* **A47** 317 (1993)

[8] IE McCarthy and A T Stelbovics, *Phys .Rev.* **A28** 2693 (1983)

[9] IE McCarthy, K Ratnavelu and Y Zhou *J.Phys.* **B26** 2733 (1993)

[10] A Z Msezane, Phys.Rev. **A37** 1787 (1988)

[11] T M Miller and B Bederson, Adv.At.Mol.Phys. **13** 1 (1977)

[12] E W McDaniel, Atomic collisions: Electron and photon projectiles (wiley, New York, 1989)

[13] L Kasden, T M Miller and B Bederson, *Phys .Rev.* **A8** 1562 (1973)

[14] P J Visconti, J A Slevin and K Rubin, *Phys .Rev.* **A3** 1310 (1971)

[15] S K Srivastava and L Vuskovic, J.Phys.**B13** 2633 (1980); L Vuskavic and S K Srivastava, ibid.**13** 4849 (1980)

[16] C K Kwan, W E Kauppila, R A Lukaszew, S P Parikh, T S Stein, Y J Wan and M S Dababneh, *Phys. Rev.* **A44** 1620 (1991)

[17] S P Parikh, W E Kauppila, C K Kwan, R A Lukaszew, D Przybyla,T S Stein, and S Zhou, *Phys. Rev.* **A47** 1535 (1993)

[18] E Surdutovich, W E Kauppila, C K Kwan, E G Miller, S P Parikh, K A Price and T S Stein, *Nucl. Inst. Meth.* **B221** *97* (2004)

[19] S Y El-Bakry and M Y El-Bakry, Indian J.Phys.**78** 1313 (2004)

[20] W. Banzhaf and P. Nordin and R. Keller and F. Francone, *Genetic Programming An Introduction: On the Automatic Evolution of Computer Programs and Its Applications*, Morgan Kaufmann Publishers, Inc. San Francisco, California,1998.

[21] R. Riolo, *Genetic Programming Theory and Practice*, Bill Worzel – Computers, 2003.

[22] J. Koza, M. Keane, M. Streeter, W. Mydlowec, J. Yu,, Genetic Programming IV: Routine Human-Competitive Machine Intelligence, Guido Lanza - Computers – 2005.

[23]R. Poli, ,*Genetic Programming for Feature Detection and Image Segmentation*, "Dec" School of Computer Science, University of Birmingham.1995

[24] K. Sharman and A. Alcazar and Y. Li,, *Evolving Signal Processing Algorithms by Genetic Programming*, First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA'95),473--480,1995.,

[25] M Y El-Bakry and A Radi, Int. J. Mod. Phys C, Vol. 17, issue 8 (2006).

[26] D.E. Goldberg, *Genetic Algorithm in Search, Optimization and Machine Learning. Addison-Wesley (1989).*