

PAPER • OPEN ACCESS

Python script used as a simulator for the teaching of the electric field in electromagnetism course

To cite this article: K. L. Cristiano *et al* 2019 *J. Phys.: Conf. Ser.* **1247** 012044

View the [article online](#) for updates and enhancements.

You may also like

- [New results in high-resolution X-ray fluorescence spectroscopy](#)
Matjaž Žitnik, Matjaž Kavi, Klemen Buar *et al.*
- [Impact of cultural transformation campaigns around energy savings in hotels. Study in Bucaramanga and Valledupar, Colombia](#)
S Serrano, Y Muñoz and J Quiroga
- [QuantumATK: an integrated platform of electronic and atomic-scale modelling tools](#)
Søren Smidstrup, Troels Markussen, Pieter Vancraeyveld *et al.*



The Electrochemical Society
Advancing solid state & electrochemical science & technology

241st ECS Meeting

Vancouver, BC, Canada. May 29 – June 2, 2022



ECS Plenary Lecture featuring
Prof. Jeff Dahn,
Dalhousie University



Register now!



Python script used as a simulator for the teaching of the electric field in electromagnetism course

K. L. Cristiano¹, A. Estupiñán^{2,3} and D. A. Triana^{1,3}

¹ CIMBIOS, Escuela de Física, Universidad Industrial de Santander (UIS), Bucaramanga, Colombia

² Departamento de Matemáticas-Física, Facultad de Ciencias Exactas, Naturales y Agropecuarias. Universidad de Santander (UDES), Bucaramanga, Colombia

³ Departamento de Matemáticas, Universidad Autónoma de Bucaramanga (UNAB), Bucaramanga, Colombia

E-mail: dantrica@saber.uis.edu.co (D. A. Triana)

Abstract. We present in this article a Python script, based on a methodology to obtain the electric field produced by n electric charges. This tool was implemented in courses of electromagnetism and its laboratory in three institutions of higher education. The aim of this work was to incorporate information and communication technologies (ICTs) at the physics subjects, in accordance with the programs promoted by the Colombian Ministry of Education. We wanted to connect the students with sensory experiences of the physical phenomena that allow them to improve their experience of learning of subjects traditionally studied through the board. Finally, in this work, an interactive computational code was obtained, in which the electric field of the discrete and continuous charge distributions can be calculated, for the classical problems that are shown in an electrical physics course.

1. Introduction

When the topic of the electric field is taught, in the course of electromagnetism to level university level for engineers and physicists, it is usually approached from the point of view of Faraday and Gauss's law, this can be seen in the authors of the more popular books of electromagnetism like Serway [1], Sears [2], Giancoli [3], and Burbano [4], which study the electric field as a disturbance represented by lines that propagate through space, interacting with the electric charges and finally at different points in the space, they are pretty well described with a vector [5]. It's these point where headache of the students begins since they have to combine the vector analysis, with the electric field physics description. Usually, both abstract and imaginary concepts can't be bound, with one previous sense experience which the students have had.

The other hand, the Colombian Ministry of Education sponsors the use of information and communication technologies (ICTs) as a didactic tool for teaching and learning [6]. Some pedagogical strategies like Just in Time Teaching (JiTT) take into consideration the use of (ICTs) such as Google classroom, blackboard, and Moodle [7]. Another initiative from the University of Colorado founded by Nobel Laureate Carl Wieman [8]. They develop interactive simulations such a way that the students learn by exploring. So the aim to connect the students with sense experiences to improve him learn [9]. So, arose the idea to develop a methodology, that allowed made follow up on the electric field problems, at the same time it's viewed from software tool.



In this article, we decided to choose the programming language Python [10], one of the reasons is the great diversity of graphics libraries that this language uses, besides that it allows us to use a vector environment to perform the operations that were necessary to implement, for the calculation of the electric field generated by discrete and continuous charge distributions.

This paper is organized as follows. In section 2, we described the steps implemented in the code of Python, which we used to evaluate the electric field in any one point of the space. In the section 3 we present the results of this work, mainly, the application of this script in some typical cases, to solve problems of calculation of the electric field produced for discrete charge distributions and punctual charges, in addition to continuous charge distributions (line, ring and cylinder charged). Finally, section 4 presents our conclusions.

2. Methods

The board class allowed generating discussion with the students, about what should be the appropriate way to solve the problems of the electric field. Then, an algorithm or strategy to provide the solution on seven (vii) steps was defined, independently from electric charges position on the space [11]:

- (i) Define the system of reference. It's more intuitive for students to catch on the rectangular coordinates.
- (ii) Draw the position vectors pointed both the electric charges and the position vector to the point where the electric field will be calculated. These vectors come from the origin of the coordinate system.
- (iii) Draw the relative position vectors $\vec{r}_i$, that correspond to n differences between the vector at the point and vector at the electric charge.

$$\vec{r}_i = \vec{r}_p - \vec{r}_{qi} \quad (1)$$

- (iv) Find n magnitudes of the n relative position vectors.

$$r_i = \sqrt{\vec{r}_i \cdot \vec{r}_i} \quad (2)$$

- (v) Find the unit vectors of the n relative position vectors.

$$\hat{r}_i = \frac{\vec{r}_i}{r_i} \quad (3)$$

- (vi) From Coulomb law calculate n electric fields. At the practice, it will never distinguish each one, we always will see the total electric field at the point (x_p, y_p, z_p) .

$$\vec{E}_i = \frac{q_i}{4\pi\epsilon_0 r_i^2} \hat{r}_i \quad (4)$$

- (vii) From superposition principle, find it the one electric field vector at the point and its magnitude.

$$\vec{E}_p = \sum_i \vec{E}_i \quad (5)$$

Python is a free language of high level, oriented to objects, focused on the code readability of easy learning, was the ideal language for the construction of the interactive tool. Then, we choose Python programming language to write a script *charges.py* mean a class named *charges()*. The class contains a method *e_field()* that included the previous steps (Coulomb's Law) in order for the students understanding step by step the physical phenom. By separately methods is possible to draw the charges, charge position vectors, and relative position vectors. The electric charge distributions were building in other methods as *punctual_charges()*, *line()*, *ring()*, and *cylinder()*; applying operations of symmetric over the punctual charges.

3. Results

The main method follows the algorithm structure implemented with a while loop that responds to the number of electric charges. On the other hand, the Python module was development like a class named *charges()*. The module follows the algorithm structure implemented with several methods that respond to the number of electric charges. On the other hand, the charges, the position vectors with respect to the origin, and the relative position vectors were drawn in a graph based on the module *matplotlib.pyplot* [12]. The position vectors with respect to the origin and the relative position vectors are represented in a graph constructed from the module.

The Python module allows solving problems of the more several charges such as a cubic crystalline structure formed by positive ions (See Figure 1(a)), and electric charge distributions too. An advantage over the Phet simulator is the ability to represent and solve three-dimensional punctual charges configurations (See figures 1(b) and 1(c)).

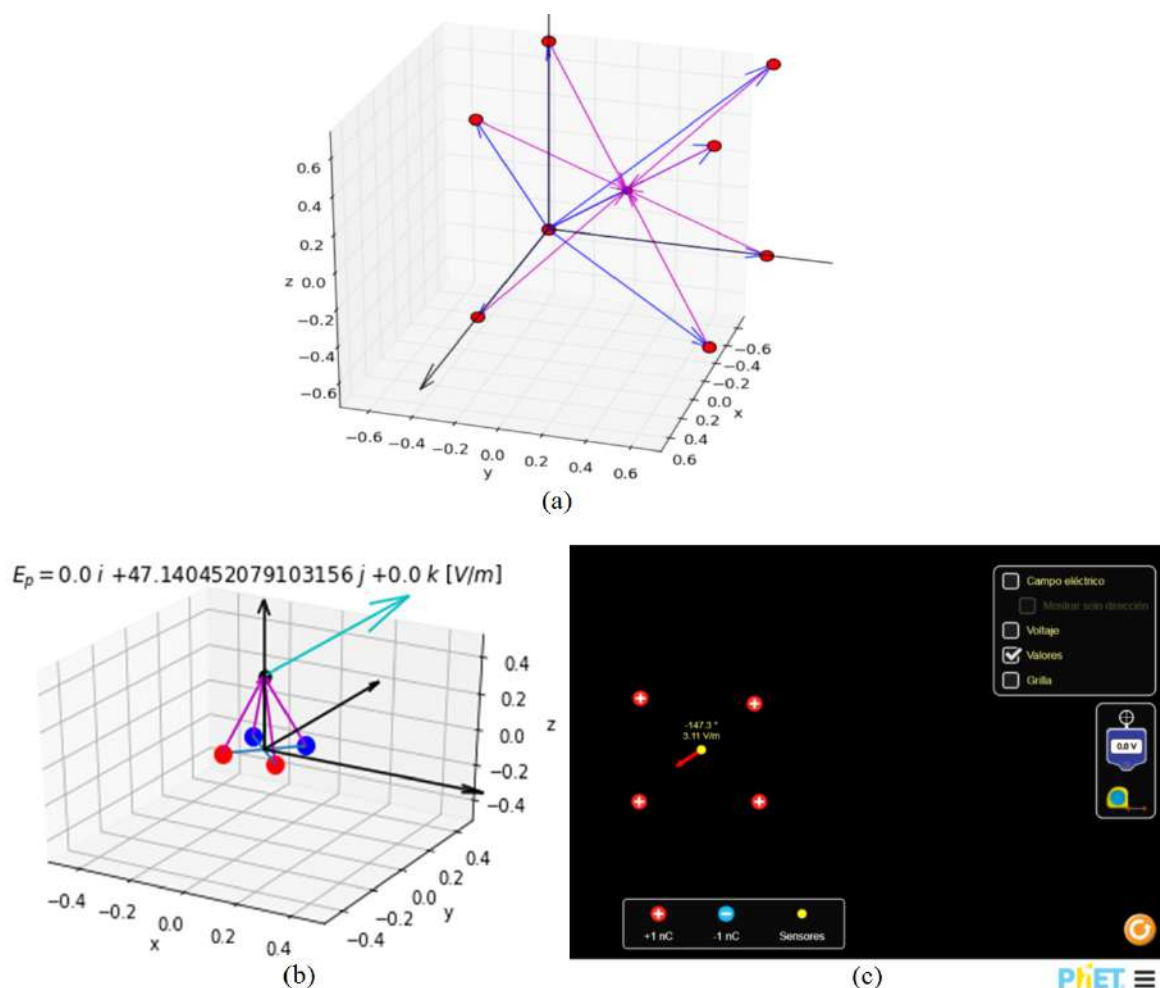


Figure 1: Cubic (a) and square (b) configurations of electric charge, both carried out in its own simulator. Square configuration of electric charges made in Phet simulator (c).

Positive electric charges were drawn in red while negative ones in blue, The point p where the electric field was calculated it has been drawn in black, the position vectors in blue and relative position vectors in magenta.

At the same time what the plot is obtained, it's possible to review in the terminal the magnitudes of the electric charges, the position vectors of the electric charges, the relative

position vectors and their magnitude and unit vector, to finish with the electric field vectors produced for each electric charge and the electric field vector from the superposition.

Once the code was written, we have implemented it in the subject of electrostatic, using the placement of electric charges for certain charge distributions, where each method has a continuous charge distribution specific, for example for the case of line and ring charged, as can be seen in the figures 2 and 3, respectively. The students have been able to recognize the punctual electric charges on the distributions and understand easily the subject.

In addition to the implementation in the code for linear charge distributions, in this computational tool a function was performed that allows the student to calculate the electric field generated by a surface charge distribution, for example a hollow cylinder, as can be seen in the Figure 4.

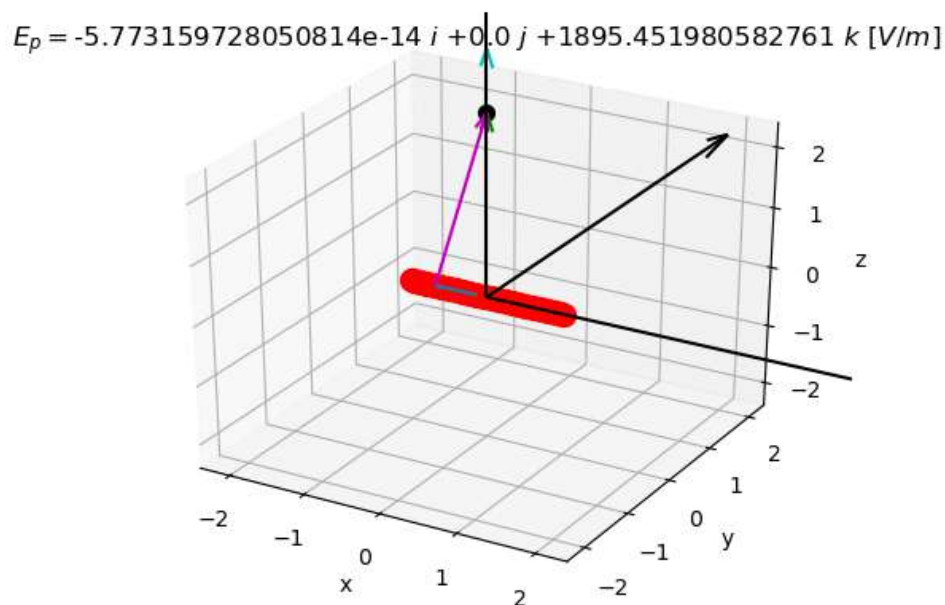


Figure 2: Calculation of the electric field along the z-axis, produced by a linear charge distribution.

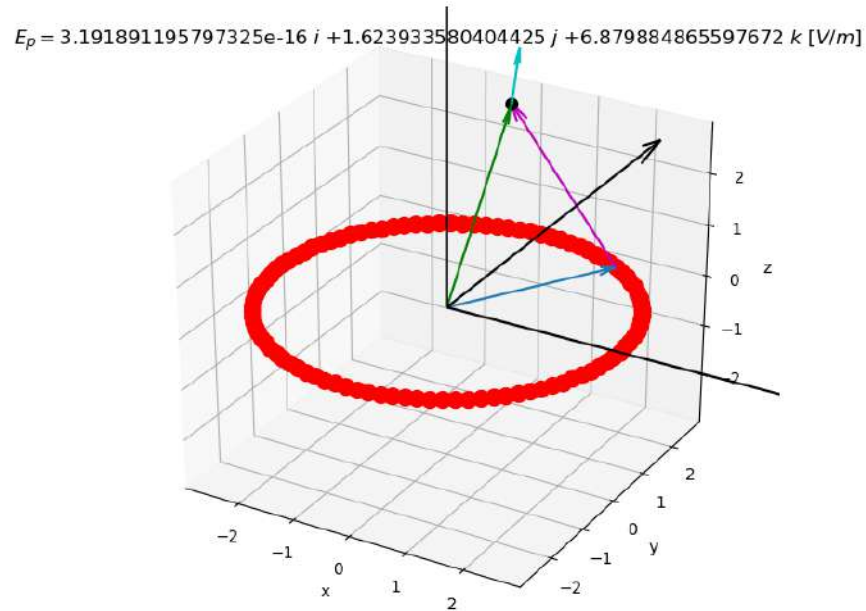


Figure 3: Calculation of the electric field at an arbitrary point, produced by a linear charge distribution in the form of a ring.

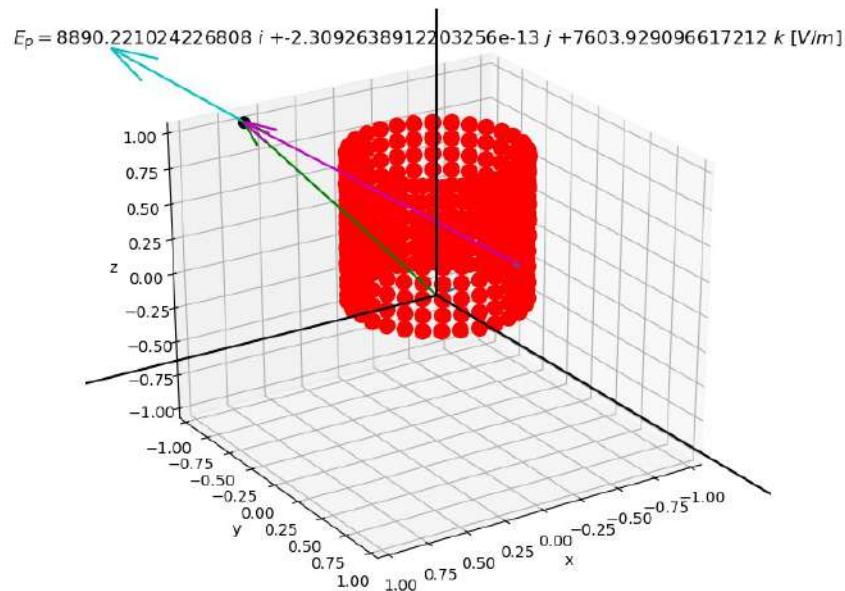


Figure 4: Calculation of the electric field at an arbitrary point, produced by a cylindrical charge distribution.

4. Conclusions

We wrote and implemented a methodology and an algorithm developed in Python to calculate the electric field produced by n electric charges, with which it's also possible to calculate electric fields produced by different continuous charge distributions modeled, using several punctual charges, distributed specifically for the desired case to study (line, ring, and cylinder charged).

In this document, we were able to show a very useful and novel computational tool, so that students can calculate the electric field of some types of electric charge distribution, where we were implemented this script of Python into electromagnetism class (theoretical and laboratory) in three Universities, which showed a greatly improved the experience of students accustomed to traditional teaching methods in their previous courses of calculus and mechanics.

Finally, we developed a computer code of free access for being used on Python 2.7, with the purpose of allowing students to interact easily with it, in addition to this digital tool being very useful to improve the understanding of the definition of the electric field. The code link in GitHub is <https://github.com/dantrica/Electrostatics>.

Acknowledgements

We offer special thanks to our research group (CIMBIOS), to the School of Physics from UIS for motivating the use of (ICTs) in the courses, thanks too UNAB and UTS, for being laboratories of creativity.

References

- [1] Serway R A, Jewett J W, Hernández A E G and López E F 2005 *Física para ciencias e ingeniería* vol 6 (Thomson)
- [2] Young H D, Freedman R A and Ford A L 2006 *Sears and Zemansky's university physics* vol 1 (Pearson education)
- [3] Giancoli D C and Garrido M V 2002 *Física para universitarios* 530 G5F5 2002 (Pearson Educación)
- [4] De Ercilla S B and Muñoz C G 2003 *Física general* (Editorial Tébar)
- [5] Burbano de Ercilla S, BURBANO GARCÍA E and GRACIA MUÑOZ C 1993 *Física general*
- [6] Caraballo Clavijo S V, Orozco Duarte L I *et al.* 2012 *Las TIC una herramienta efectiva en la gestión académica* B.S. thesis Universidad de La Sabana
- [7] Miranda D A, Sanchez M J and Forero O M 2017 *Journal of Physics: Conference Series* vol 850 p 012015
- [8] Wieman C E, Perkins K K and Adams W K 2008 Oersted medal lecture 2007: Interactive simulations for teaching physics: What works, what doesn't, and why
- [9] Furió-Mas C and Aranzabal J G 1998 *Enseñanza de las ciencias: revista de investigación y experiencias didácticas* **16** 131–146
- [10] Langtangen H P and Langtangen H P 2009 *A primer on scientific programming with Python* vol 2 (Springer)
- [11] Pollack G L and Stump D R 2005 *Electromagnetism*
- [12] Bassi S 2007 *PLoS computational biology* **3** e199