

Free and Open Source Software Codes for Antenna Design: Preliminary Numerical Experiments

Alessandro Fedeli* (*Assistant Professor, University of Genoa, Genoa, Italy*),
Claudio Montecucco (*Consultant, Atel Antennas s.r.l., Arquata Scrivia, Italy*),
Gian Luigi Gragnani (*Associate Professor, University of Genoa, Genoa, Italy*)

Abstract – In both industrial and scientific frameworks, free and open source software codes create novel and interesting opportunities in computational electromagnetics. One of the possible applications, which usually requires a large set of numerical tests, is related to antenna design. Despite the well-known advantages offered by open source software, there are several critical points that restrict its practical application. First, the knowledge of the open source programs is often limited. Second, by using open source packages it is sometimes not easy to obtain results with a high level of confidence, and to integrate open source modules in the production workflow. In the paper, a discussion about open source programs for antenna design is carried out. Furthermore, some preliminary numerical tests are presented and discussed, also in comparison with those obtained by means of commercial software. Results are related to the simulation of various typologies of antennas in order to assess the capabilities of open source software in different configurations. The presented comparisons show that, despite the above-mentioned limitations, the examined open source packages have similar performance with respect to their commercial counterparts.

Keywords – Antennas; Computational electromagnetics; Finite difference methods; Method of moments; Open source software; Time-domain analysis.

I. INTRODUCTION

Over the past years, simulation software for electromagnetic analysis and design has become more and more popular and today many commercial companies offer sophisticated packages targeting this topic. In general, these high-level commercial products contain all the tools necessary to manage the whole project flow. However, they present high costs, both at the time of purchase and for the maintenance of licenses and upgrades. Furthermore, being closed source, commercial code cannot be easily extended by the user or modified and studied for research purposes.

In the light of the last considerations, the availability of free/libre and open source software (FLOSS) [1] for electromagnetic design would be of great value not only for the scientific communities, but also for small companies, struggling to face the costs related to commercial products. Furthermore, open source codes could be proposed as an interesting alternative to commercial codes in courses about electromagnetic theory and design. Open source would also be

the natural choice when ethical concerns about science diffusion and reproducibility [2] are faced.

For these reasons, work has started, aiming to identify and evaluate possible open source programs that can be usefully employed for the design of antennas and possibly other passive electromagnetic devices such as, for example, filters, couplers, impedance adapters. The research also aims at finding software that allows for the pre- and post-processing of data. Furthermore, it has been decided to consider the possibility of easily integrating the design into the complete production and testing cycle.

First, let us note the differences between FLOSS and other programs that could be obtained free of charge. Free/libre and open source software is available in the form of source code and is freely modifiable and adaptable to the user needs (while some restrictions may exist for the redistribution of a modified version of a program), nor it is mandatory, although very common, that the program is distributed at no cost. Instead, other free programs are usually offered only in executable form and do not allow for any intervention, integration, or porting by the user. For these reasons, they are not considered in the paper, although sometimes they may be useful.

The study presents the preliminary phases of the project along with some numerical experiments with open source programs for antenna design. The present paper extends and discusses more deeply the research that has recently been presented at the MTTW'19 Workshop on "Microwave Theory and Techniques in Wireless Communications" [3].

In particular, a background discussion about the project flow and numerical techniques is carried out in Sections II and III, while in Sections IV and V two products are examined in more detail, namely OpenEMS [4] and xnc2c [5]. A set of selected open source tools for pre- and post-processing of data is briefly discussed in Section VI. Section VII is devoted to the presentation of some numerical examples, after which conclusions are drawn.

II. PROJECT FLOW

Without going into details, inside the general workflow of an antenna design – or, more generally, the design of any kind of electromagnetic device – a sequence of many actions can be recognised. According to the commonly used terminology, one must deal with:

* Corresponding author.
E-mail: alessandro.fedeli@unige.it

- 1) *pre-processing* – The geometric and electrical specifications are translated to an approximate numerical model, which is compatible with the calculation engine that will be used. Development may occur by means of a graphical interface, by writing command files in a special language, or mixing the two possibilities. During the pre-processing phase, the documentation (drawings, layout, list of components, etc.) needed for prototyping (and eventually for production) could also be generated;
- 2) *solution* – A proper solver is used for computing the solution of the electromagnetic problem. The solver stores results in a set of files, ready for being post-processed. Different typologies of solvers (which are common in the open source world) will be discussed in the next Sections;
- 3) *post-processing* – The obtained results are processed in order to extract the required data. Scripting languages or graphical user interfaces may be adopted to perform this operation. As a result, a set of output graphs or tables is usually produced, with the aim of verifying the design and the simulation, generating part of the documentation, organising comparisons. It is worth noting that the produced output may be used as input for other numerical simulations (e.g., antenna arrays).

III. NUMERICAL METHODS

Methods for electromagnetic simulation can be divided into various classes, depending on the numerical model used to approximate the Maxwell equations. Some software directly implements a numerical version of such equations, while other derives the fields by means of potentials. Furthermore, many codes are implemented in the frequency domain, while others work directly in the time domain. Not all possible numerical methods are implemented in an open-source version. As regards antenna design in particular, open source essentially focuses on the method of Finite Differences in the Time Domain (FDTD) and on the Method of Moments (MoM) [6].

FDTD is usually based on the leap-frog scheme proposed by Yee [6], possibly with (little) variants among various codes. The whole space that surrounds the antenna is modelled. Hence, for radiating problems, suitable absorbing conditions at the boundary of the simulation domain, simulating the Silver-Muller radiation conditions at infinity, must be included in the model.

The MoM has encountered large acceptance in the sector of electromagnetic modelling and computation. Programs that will be discussed are based on the so-called thin-kernel approximation for wires [6]. Within this approximation, the problem is basically modelled with a set of PEC wires, carrying current densities directed as the wire axis. Usually, with the MoM, one or more integral equations including the proper Green function are considered. In this case, boundary conditions are implicitly accounted for, and there is no need to model the space outside the antenna.

Conversely, the Finite Element Method (FEM), which is also widespread in the open source world for the solution of complicated engineering equations (e.g. [7]), currently does not seem to be ready for an open application in antenna design,

although a very promising framework, named ONELAB [8], is in rapid development.

A. Commercial Versus FLOSS Simulation Packages

There is a popular opinion that open source products are generally worse than their commercial analogues. In general, this is not necessarily true: there are many excellent quality open source products in the most varied fields and the scientific and academic worlds make more and more use of them. Unfortunately, as regards the electromagnetic simulation and antenna design in particular, it must be recognised that now commercial products can offer, in terms of usability and versatility, much more than the open source counterpart. In fact, computational electromagnetics at high frequencies represents a field in continuous evolution but still young, and this reflects on the open source community. For these reasons, the development of complete FLOSS packages for electromagnetic computation and design is going slow.

In general, high-level commercial products contain all the tools necessary to cover the entire flow of the project. However, they are typically expensive, both at the time of purchase and for the maintenance of licenses. Instead, the use of open source software implies the need to integrate non-homogeneous products, which are usually not characterised by comparable functionalities. To build an omni-comprehensive suite that goes from pre-processing to post-processing, it will certainly be necessary to integrate various software with other parts, which must be developed “in-house”. In spite of this fact, there are some packages that are already usable profitably and are constantly being developed. Furthermore, access to the sources and the existence of a community of developers that are usually very available, sometimes allow problems to be solved much more quickly than commercial products. A list, even though not exhaustive, of salient points for commercial packages as well for open source is reported in Table I.

IV. METHOD OF MOMENTS

The MoM is historically the one for which the first open source applications exist and the reference program for many open source numerical codes is definitely the NEC-2 [6], [9]. The NEC-2 is limited to the modelling of wire/pipe structures (possibly connected through networks of lumped components or transmission lines) operating in a vacuum. The possibility of modelling antennas and devices containing dielectric parts is therefore excluded, and when metal plates have to be modelled, they are approximated with a grid of wires. There are some limitations in the positioning of the intersections and the mutual positions of the wires, which have been highlighted over the years. Despite these known limitations, the NEC-2 has been and still is successfully used for the analysis and design of many types of antennas. The original Fortran source of the NEC-2 is still available, like the porting of the code in C [5] and C++ [10].

At present, there are no FLOSS graphic modelers nor other free programs available for generating input data. The possibility to process the original output data file through open source programs is also rather limited. Along the years, the NEC-2 has been used in many projects; a lot of research papers

about the program have been written, and many pre-packaged models of antennas of various kinds can be found on the Internet and can be used as a starting point for a new project. Furthermore, being a project originally developed by the U.S.A.

administration, the NEC-2 is accompanied by an extended documentation, containing the description of the theory as well as a user manual and many examples of use.

TABLE I
COMPARED FUNCTIONALITIES OF COMMERCIAL AND FLOSS SIMULATION CODES

	Commercial Codes	FLOSS codes
Graphical User Interface (GUI)	Yes	No
Multi-solver	Yes	No
Source code availability	No	Yes
Documentation	Yes	Not always
Scriptable with standard languages	Not always	Yes
Optimised code	Yes	Not always
Import in different format	Some open-source formats missing	Some proprietary formats missing
Export in different format	Some open-source formats missing	Some proprietary formats missing
License fee	Yes	No
Community of developers	No	Yes
Automatic generation of meshes	Yes	No
Open format for data	No	Yes
Easy to integrate with other software	Seldom	Yes
All-in-one package	Yes	No
Parallel	Yes (may require additional licenses)	Usually yes
Can use one or more GPUs	Not always (may require additional licenses)	Few codes

A. *nec2c* and *xnec2c*

Among the most interesting software using the NEC-2 engine, we focused on *nec2c* and on its companion *xnec2c*: *nec2c* [5] is a porting in C of the original code. It can easily replace the original, it allows for command line input and output and dynamically allocates memory resources, while the original Fortran code requires recompiling the program if the defined memory is insufficient. The companion program *xnec2c* [5] has been further extended to be multi-threaded and allows viewing and modifying the project in a more or less interactive way. It is only visual and lacks a complete textual output in the original NEC2 format. However, in recent versions, some data can be exported not only as images, but also as text files, in a format compatible with the graph program Gnuplot [11].

V. METHOD OF FINITE DIFFERENCES IN TIME DOMAIN

Most recently developed open source software is based on the FDTD method. The usual implementation of FDTD for electromagnetic simulations is always based on the scheme proposed by Yee, with only minor differences among various implementations. The method apparently offers two great advantages compared to the MoM: it does not require the inversion of large data matrices (and therefore has quite low memory needs) and, with a single simulation in the time domain, it is able to evaluate the antenna over a wide range of frequencies. The method also has disadvantages, including the need to approximate the model on regular grids. Nevertheless, it is widely used, even in commercial products. OpenEMS [4], *gprMax* [12] and *Meep* [13] are certainly among the most interesting open source implementations, while some other

software packages that are regularly maintained by the open source community are *Vulture* [14], *GSvit* [15], and *Angora* [16].

A. *OpenEMS*

The present paper focuses on *OpenEMS* [4], which is actually the solver of a well-integrated suite of programs for geometric modelling, resolution, output of data in various formats and integration with circuit simulation of electromagnetic devices and systems. *OpenEMS* is based on the FDTD implementation proposed by Rennings et al. [17]. The *OpenEMS* suite is scriptable by Matlab/Octave and partly by Python. Data can be exported in hdf5 file format, and some results, in particular the geometry of the problem, the radiation patterns and animations of fields can be dumped to *Paraview* [18]. Furthermore, there are some programs, notably *pcb-rnd* [19], that can export in a form suitable for *OpenEMS*.

VI. AUXILIARY TOOLS

Since one of the final goals of the project is that of creating a comprehensive chain based on FLOSS tools, including pre- and post-processing of data, in this section some of the most used, and useful, programs are briefly presented.

For CAD and meshing the usual tools used within FLOSS codes are perhaps *Gmsh*, which is also the software around which *ONELAB* [8] is developed, and *Salome* [20]. Other featured tools that can be profitably used are *FreeCAD* [21] and *SolveSpace* [22]. For visualisation of complex geometries, the ubiquitous tool *Paraview* [18] is mainly used. Note that *Paraview* could also generate its own mesh; however, this feature is usually considered when dealing with computer

graphics rendering and animation of complex scenarios but it is not common in numerical computation, where Paraview is mainly used in post-processing, for graphing three-dimensional radiation pattern, field distributions and model geometry.

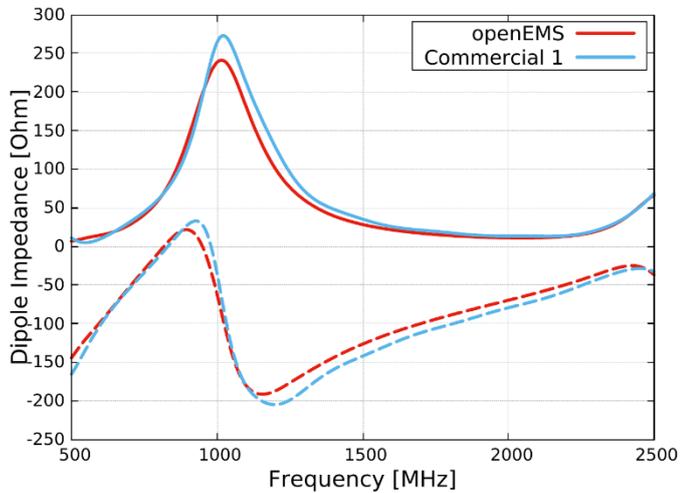


Fig. 1. Comparison of the impedances of a simple printed dipole. Red line: OpenEMS; Blue line: commercial code 1. Continuous lines: real part; Dashed lines: imaginary part.

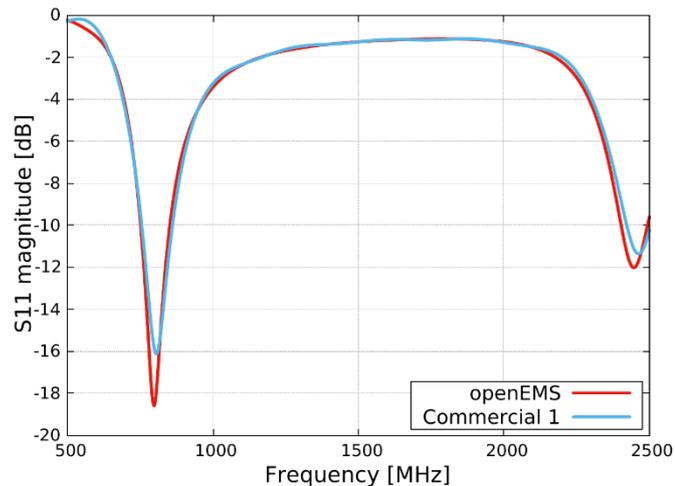


Fig. 2. Reflection coefficient of a simple printed dipole. Red line: openEMS; Blue line: commercial code 1.

Note that both Gmsh and Salome have also extended post-processing capabilities and could be integrated in the whole design chain. VisIt is another worth citing tool [23]. It is very complicated but can deal with two- and three-dimensional data, either scalar or vectorial, and can, by default, access more than 120 different data formats.

For graphing purposes, the program Gnuplot [11] is very widespread, while in the Python world the library Matplotlib [24] exists. Both tools have the capabilities of drawing the usual set of two-dimensional representations of data, for example, cartesian and polar plots, pie diagrams, scatter plots etc. Three-dimensional representations are also possible. Gnuplot is probably better for simple tasks, but it can also be used for sophisticated representations of data. Matplotlib is a little bit

more complex but can exploit the whole power of the Python programming language.

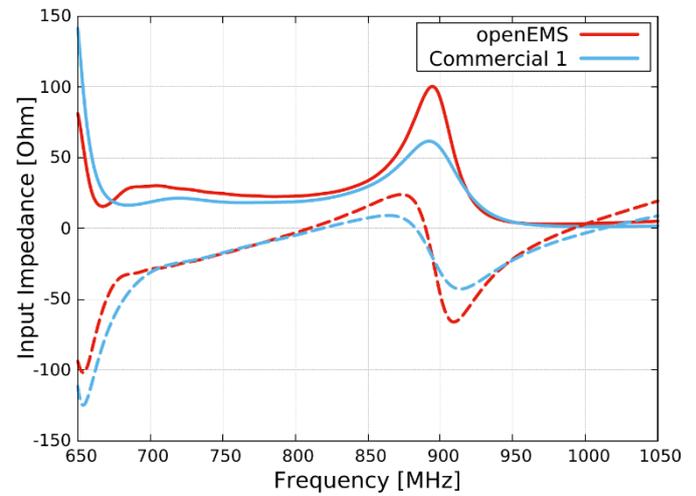


Fig. 3. Feeding point impedance of the three-element printed log-periodic antenna [33]. Red line: OpenEMS; Blue line: commercial code 1.

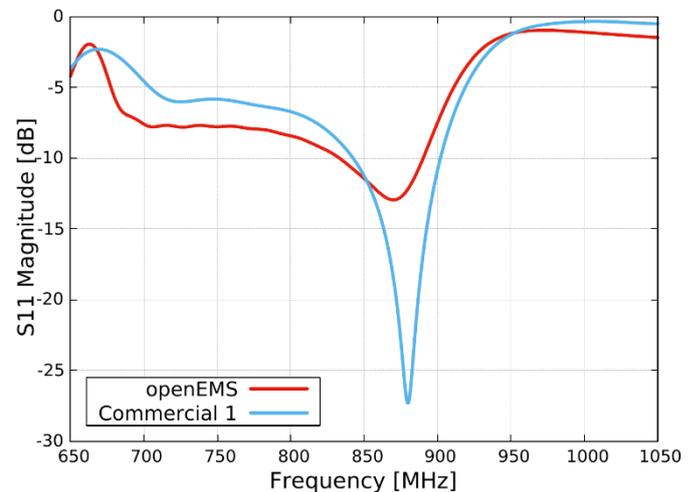


Fig. 4. Reflection coefficient of the three-element printed log-periodic antenna [33]. Red line: OpenEMS; Blue line: commercial code 1.

VII. SIMULATIONS AND COMPARISONS

In order to check the capabilities of open source, we designed some basic simulations; we compared the results among them and with those achieved by using two widespread commercial codes (identified with 1 and 2).

In this preliminary work, we focused our attention on two tools only, which, in our opinion, were worth using in an open source platform: OpenEMS and nec2c.

In the first example, a simple printed dipole was considered. This dipole has already been simulated by using a commercial tool, during a previous study about log-periodic antennas [25]. The dipole dimensions were 6 mm width and 144.2 mm total length.

In Fig. 1, the input impedances are shown, while in Fig. 2 the related magnitudes of the reflection coefficients are plotted. It can be seen that the match between the two results is good.

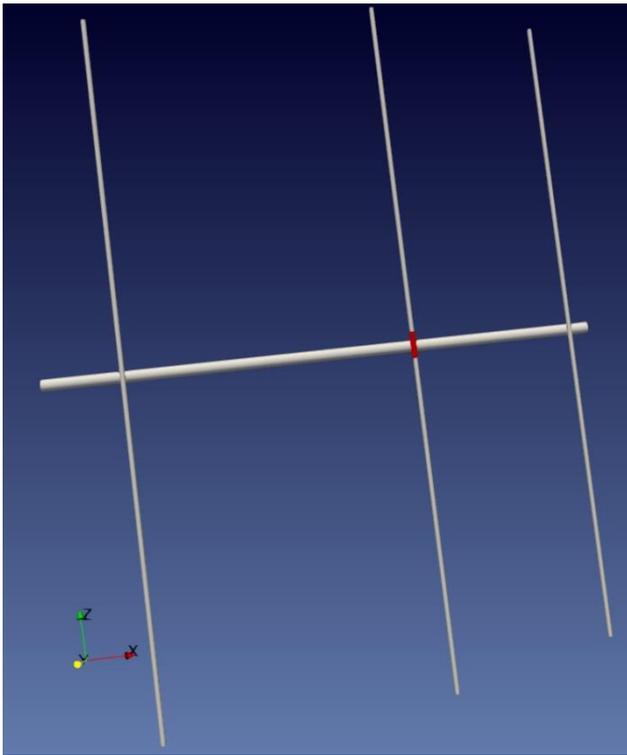


Fig. 5. Pictorial view of the three-element Yagi antenna.

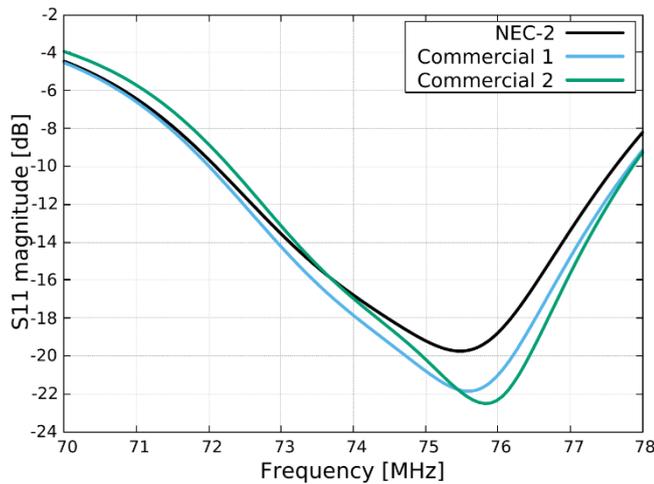


Fig. 6. Reflection coefficients of the analysed 3-element Yagi. Black line: nec2c; Blue line: commercial code 1; Green line: commercial code 2.

Starting from this and other results on single dipoles, a three-element printed log-periodic antenna was designed [25]. Even though the overall result is not as good as in the case of a single dipole, the agreement between commercial code 1 and OpenEMS is anyway acceptable, as it can be seen in Fig. 3, in which the input impedance is shown. The resulting reflection coefficient has also a fair agreement, although some differences can be noticed. These results are shown in Fig. 4.

The third test was about the design of a three-element Yagi antenna centred at 73.8 MHz. In Fig. 5, a pictorial view of the antenna is shown. For this case, we avoided to use OpenEMS since it would require a too fine cell discretization to provide reasonable results. The magnitude of the reflection coefficient

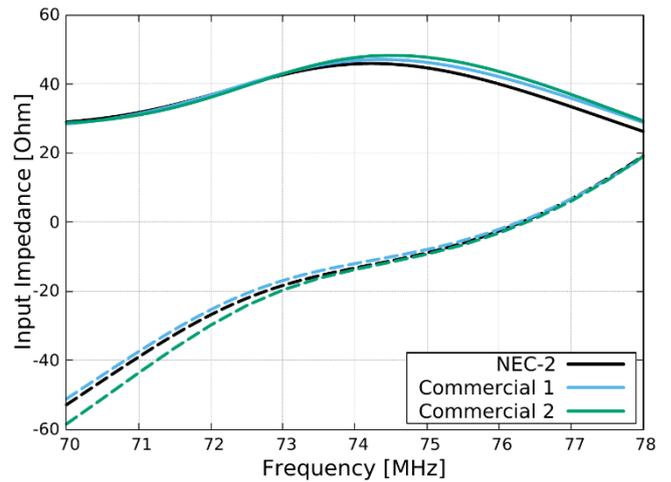


Fig. 7. Feeding point impedance of the analysed three-element Yagi. Black line: nec2c; Blue line: commercial code 1; Green line: commercial code 2. Continuous lines: real part; Dashed lines: Imaginary part.

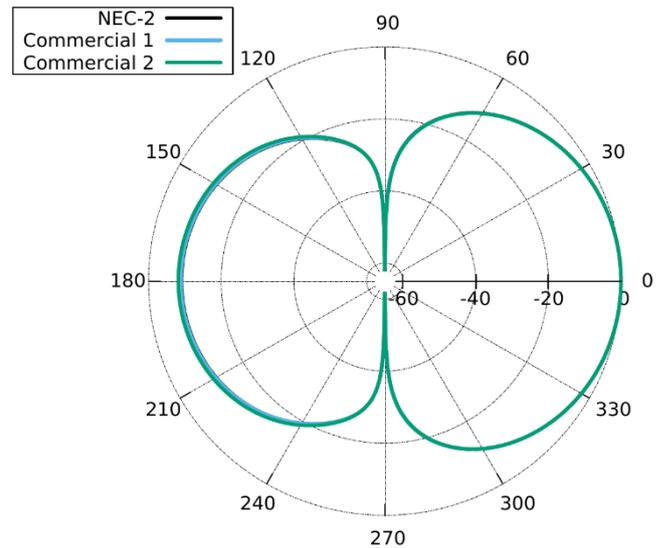


Fig. 8. Normalised radiation pattern of the analysed three-element Yagi in the E-plane. Black line: nec2c; Blue line: commercial code 1; Green line: commercial code 2.

at the feeding point of the antenna is shown in Fig. 6. In this case, results were compared also with commercial code 2. It can be noticed that the two commercial codes provide results that are in a very good agreement, while nec2c gives a curve that is a little bit more smoothed. Nevertheless, these small differences are more than acceptable.

This result is also confirmed by the values of the input impedance of the antenna, which are shown in Fig. 7. As for the radiation patterns, a comparison is shown in Fig. 8, in which the normalised pattern in the E-plane is illustrated, and Fig. 9 presents the normalised H-plane pattern. It can be seen that there are only very minor differences among the three results. The values of directivity are very close to each other, with differences that are less than 0.15 dB. Results about the main radiation parameters of the antenna are summarised in Table II.

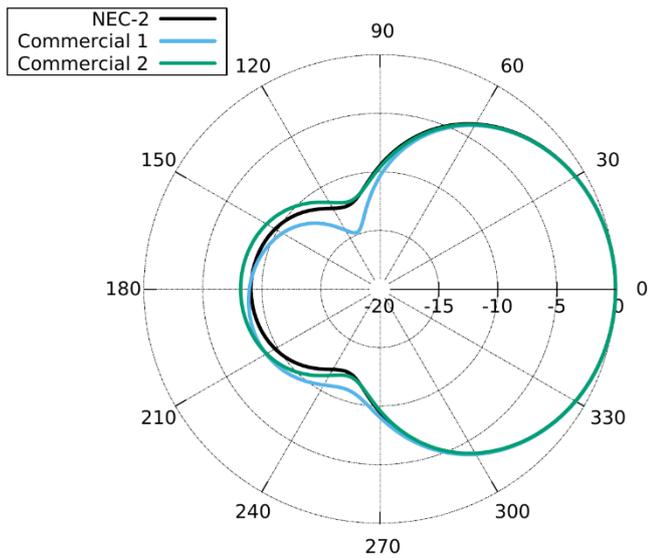


Fig. 9. Normalised radiation pattern of the analysed three-element Yagi in the H-plane. Black line: nec2c; Blue line: commercial code 1; Green line: commercial code 2.

TABLE II

MAIN RADIATION PARAMETERS OF THE YAGI ANTENNA FOR DIFFERENT SOLVERS

	NEC-2	Commercial 1	Commercial 2
Directivity, dBi	7.22	7.26	7.16
-3 dB aperture in the E-plane, Deg.	64.9	64.9	64.9
-3 dB aperture in the H-plane, Deg.	108.7	108.4	108.2
Front-to-Back ratio, dB	9.1	8.9	8.2

Another test aimed at comparing the behaviour of the two open source tools has been examined. To this end, the BiQuad antenna that can be found on the tutorial examples of OpenEMS was simulated also by using both nec2c and commercial code 1. In Fig. 10, a pictorial view of the antenna and its radiation pattern, produced by dumping to Paraview the OpenEMS results, is reported.

Figure 11 shows the comparison of the input impedances given by the three simulators. As can be seen, the real parts of the impedance are almost the same in the three cases, while some differences can be appreciated in the reactive parts. However, the behaviour is consistent, and the general agreement is good.

A comparison of the reflection coefficients, which are shown in Fig. 12, reveals that the shift in frequency of the resonance peak has a relative difference that is less than 4 % in any considered case.

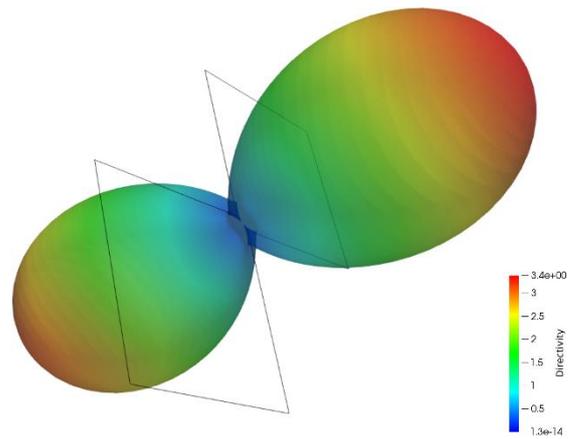


Fig. 10. A pictorial three-dimensional view of the radiation pattern of the BiQuad antenna. The picture was extracted from the OpenEMS simulation, by using Paraview. Directivity is in linear scale.

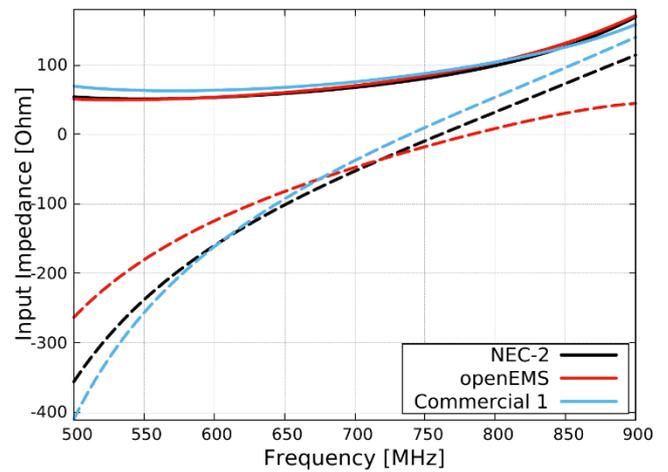


Fig. 11. Comparison of the results obtained by OpenEMS, nec2c, and a commercial code. Feeding point impedance of the BiQuad antenna. Red line: OpenEMS; Black line: nec2c. Blue line: commercial code 1. Continuous lines: real part; Dashed lines: imaginary part.

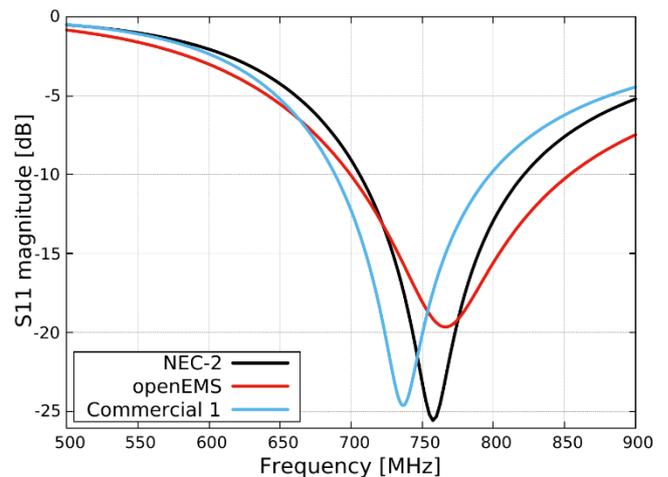


Fig. 12. Magnitudes of the reflection coefficients of the BiQuad antenna, as obtained by openEMS, nec2c, and a commercial code. Red line: OpenEMS; Black line: nec2c. Blue line: commercial code 1.

VIII. CONCLUSION

In this paper, a discussion about open software for electromagnetic design has been presented, together with a short list of the possible solutions that are available. Furthermore, some numerical experiments on different types of antennas have been reported and discussed. While this preliminary investigation has confirmed that commercial programs are still superior in managing the whole flow of the design, the research has also demonstrated the availability of open source packages that can provide good capabilities and similar quality of results compared to commercial codes. These open source packages could be satisfactorily used for the design of antennas or other electromagnetic devices. Furthermore, open source offers a wide variety of very high-quality tools for pre- and post-processing of data. Hence, while the integration among various open source packages is still at its early days, open source software can nowadays be considered as a promising alternative to commercial tools.

REFERENCES

- [1] R. Stallman, "FLOSS and FOSS." [Online]. Available: <https://www.gnu.org/philosophy/floss-and-foss.en.html>. [Accessed: Nov. 20, 2019].
- [2] V. Stodden et al., "Enhancing Reproducibility for Computational Methods," *Science*, vol. 354, no. 6317, pp. 1240–1241, Dec. 2016. <https://doi.org/10.1126/science.aah6168>
- [3] A. Fedeli, G. L. Gragnani, and C. Montecucco, "Numerical Experiments With Open Source Programs for Antenna Design," in *2019 IEEE Microwave Theory and Techniques in Wireless Communications (MTTW)*, Riga, Latvia, 2019, pp. 34–38. <https://doi.org/10.1109/mttw.2019.8897221>
- [4] T. Liebig, A. Rennings, S. Held, and D. Erni, "openEMS – a Free and Open Source Equivalent-Circuit (EC) FDTD Simulation Platform Supporting Cylindrical Coordinates Suitable for the Analysis of Traveling Wave MRI Applications," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 26, no. 6, pp. 680–696, Nov. 2013. <https://doi.org/10.1002/jnm.1875>
- [5] N. Kyriazis, "NEC2 in C & with GUI." [Online]. Available: <http://www.5b4az.org>. [Accessed: Nov. 20, 2019].
- [6] D. B. Davidson, *Computational Electromagnetics for RF and Microwave Engineering*. Cambridge University Press, 2010. <https://doi.org/10.1017/CBO9780511778117>
- [7] A. Bugajev, G. Jankevičiūtė, and N. Tumanova, "The Mathematical Modelling of Heat Transfer in Electrical Cables," *Electrical, Control and Communication Engineering*, vol. 5, no. 1, pp. 46–53, May 2014. <https://doi.org/10.2478/ecce-2014-0007>
- [8] C. Geuzaine et al., "ONELAB: Open Numerical Engineering Laboratory," in *11e Colloque National en Calcul des Structures*. Giens, France: CSMA, May 2013.
- [9] G. J. Burke and A. J. Poggio, "Numerical Electromagnetic Code (NEC) – Method of Moments," Lawrence Livermore National Laboratory, Tech. Rep., Jan. 1981.
- [10] T. Moltano, "necpp." [Online]. Available: <https://github.com/tmoltano/necpp>. [Accessed: Nov. 20, 2019].
- [11] P. K. Janert, *Gnuplot in Action, Understanding Data With Graphs*, 2nd Ed. Manning Publications Co., Shelter Island, NY, Mar. 2016.
- [12] C. Warren, A. Giannopoulos, and I. Giannakis, "gprmax: Open source Software to Simulate Electromagnetic Wave Propagation for Ground Penetrating Radar," *Computer Physics Communications*, vol. 209, pp. 163–170, Dec. 2016. <https://doi.org/10.1016/j.cpc.2016.08.020>
- [13] A. F. Oskooi et al., "Meep: A flexible Free-Software Package for Electromagnetic Simulations by the FDTD Method," *Computer Physics Communications*, vol. 181, no. 3, pp. 687–702, Mar. 2010. <https://doi.org/10.1016/j.cpc.2009.11.008>
- [14] I. D. Flintoft, "Vulture FDTD Code User Manual." [Online]. Available: <https://bitbucket.org/uoyae/vulture>. [Accessed: Nov. 20, 2019].
- [15] P. Klapetek. Gsvit. Department of nanometrology and technical length, Czech Metrology Institute. [Online]. Available: <http://gsvit.net/>. [Accessed: Nov. 20, 2019].
- [16] I. R. Capoglu and D. Zhang. (2012) Angora: A Free Software Package for Finite-Difference Time-Domain (FDTD) Electromagnetic Simulation. [Online]. Available: <http://www.angorafdd.org/>. [Accessed: Nov. 20, 2019].
- [17] A. Rennings, A. Lauer, C. Caloz, and I. Wolff, "Equivalent Circuit (EC) FDTD Method for Dispersive Materials: Derivation, Stability Criteria and Application Examples," in *Time Domain Methods in Electrodynamics*, P. Russer and U. Siart, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 211–238. https://doi.org/10.1007/978-3-540-68768-9_14
- [18] J. Ahrens, B. Geveci, and C. Law, "Paraview: An End-User Tool for Large-Data Visualization," in *Visualization Handbook*, C. D. Hansen and C. R. Johnson, Eds. Burlington: Butterworth-Heinemann, 2005, pp. 717–731. <https://doi.org/10.1016/b978-012387582-2/50038-1>
- [19] T. Palinkas, "Pcb-rnd." [Online]. Available: <http://repo.hu/projects/pcb-rnd/index.html>. [Accessed: Nov. 20, 2019].
- [20] A. Ribes and C. Caremoli, "Salome Platform Component Model for Numerical Simulation," in *Proceedings of the 31st Annual International Computer Software and Applications Conference – Vol. 02 – (COMPSAC 2007)*, Washington, DC, USA: IEEE Computer Society, 2007, pp. 553–564. <https://doi.org/10.1109/compsac.2007.185>
- [21] J. Riegel, W. Mayer, and Y. van Havre, "FreeCAD (Version 0.18.4)." [Online]. Available: <http://www.freecadweb.org>. [Accessed: Nov. 20, 2019].
- [22] J. Westhues, "SolveSpace." [Online]. Available: <https://github.com/solvespace/solvespace>. [Accessed: Nov. 20, 2019].
- [23] H. Childs et al., "VisIt: An End-User Tool For Visualizing and Analyzing Very Large Data," in *High Performance Visualization: Enabling Extreme-Scale Scientific Insight*, E. Wes Bethel, H. Childs, and C. Hansen, Eds. Chapman & Hall/CRC, Oct 2012, pp. 357–372.
- [24] B. W. Keller, *Mastering Matplotlib 2.X: Effective Data Visualization Techniques With Python*. Packt Publishing, 2018.
- [25] G. L. Gragnani, D. D. Caviglia, and C. Montecucco, "Design of Printed Log-Periodic Antennas for Long Range Communication Within a Wireless Sensor Network for Sea Water Quality Monitoring," *Automatic Control and Computer Sciences*, vol. 53, no. 4, pp. 364–375, Jul. 2019. <https://doi.org/10.3103/s0146411619040059>



Alessandro Fedeli received the B. sc. and M. sc. degrees in electronic engineering from the University of Genoa, Genoa, Italy, in 2011 and 2013, respectively, where he received the PhD degree in Science and Technology for Electronic and Telecommunications Engineering in 2017. He is currently a fixed-term Assistant Professor of electromagnetic fields at the Department of Electrical, Electronic, Telecommunications Engineering, and Naval Architecture (DITEN), University of Genoa, Genoa, Italy. His research activities, carried out at the Applied Electromagnetics Laboratory, are mainly focused

on the development and the application of computational electromagnetics techniques for the solution of forward and inverse scattering problems, and microwave imaging.

Dr. Fedeli is a member of the IEEE, the IEEE Antennas and Propagation Society and of the Italian Society of Electromagnetism.

Address: University of Genoa – DITEN, Via all'Opera Pia 11A, 16145 Genoa, Italy.

E-mail: alessandro.fedeli@unige.it

ORCID iD: <https://orcid.org/0000-0003-1638-4240>



Claudio Montecucco graduated in Electronic Engineering from the University of Genoa in 1995. From 1995 to 2009, he worked for various companies as an Antenna and RF Designer. Since 2009, he has been working as a Freelance Consultant and has been a Lead Engineer at Atel Antennas s.r.l., where he follows the development of new products. Furthermore, he continues to be a Fellow Researcher at the Department of Electric, Electronic, Telecommunication Engineering and Naval Architecture (DITEN) of the University of Genoa, where he carries out research activities.

Professional skills and working experience include electronic product design and development, as well as project management. His main research interests are in the fields of antennas, sensors, and RF devices.

Address: Atel Antennas s.r.l., Viale Caduti sul Lavoro 35, 15061 Arquata Scrivia, Italy.

E-mail: claudio.montecucco@tin.it

ORCID iD: <https://orcid.org/0000-0001-6440-6841>



Gian Luigi Gragnani received the Master's degree in Electronic Engineering from the University of Genoa, Genoa, Italy, in 1985. In the same year, he joined the Applied Electromagnetics Group with the Department of Biophysical and Electronic Engineering (DIBE), University of Genoa.

Subsequently, he has also cooperated with the Inter-University Research Centre for Interactions Between Electromagnetic Fields and Biological Systems (ICEMB) of which he is a former deputy director. Furthermore, since 1989, he has been responsible for the Applied Electromagnetics

Laboratory.

He is currently a Professor of electromagnetic fields at the Department of Electrical, Electronics, Telecommunication Engineering and Naval Architecture (DITEN) of the University of Genoa, where he delivers courses on antennas and on electromagnetic propagation. His current research interests are in the fields of electromagnetic scattering (both direct and inverse), microwave imaging, wideband antennas, computational electromagnetics and open source software, as well as electromagnetic compatibility.

Address: University of Genoa – DITEN, Via all'Opera Pia 11A, 16145 Genova, Italy.

E-mail: gianluigi.gragnani@unige.it

ORCID iD: <https://orcid.org/0000-0002-0002-979X>