Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb

CoSimPy: An open-source python library for MRI radiofrequency Coil EM/Circuit Cosimulation

Umberto Zanovello[a,*], Frank Seifert[b], Oriano Bottauscio[a], Lukas Winter[b], Luca Zilberti[a], Bernd Ittermann[b]

[a] Istituto Nazionale di Ricerca Metrologica (INRiM), Torino, Italy
[b] Physikalisch-Technische Bundesanstalt (PTB), Braunschweig and Berlin, Germany

ARTICLE INFO

Article history:
Received 26 October 2021
Revised 3 February 2022
Accepted 4 February 2022

Keywords:
Electromagnetic (EM) simulations
EM/Circuit cosimulation
Magnetic Resonance Imaging (MRI)
Radio-frequency (RF) coils
Open source Python library

ABSTRACT

Background and objectives: The Electromagnetic/Circuit cosimulation method represents a valuable and effective strategy to address those problems where a radiative structure has to interact with external supporting circuitries. This is of particular concern for Magnetic Resonance Imaging (MRI), radiofrequency (RF) coils design, where the supporting circuitry optimisation represents, generally, a crucial aspect. This article presents CoSimPy, an open-source Python circuit simulation library for Electromagnetic/Circuit cosimulations and specifically optimised for MRI, RF coils design.
Methods: CoSimPy is designed following an Object-orientated programming. In addition to the essential methods aimed to performed the Electromagnetic/Circuit cosimulations, many others are implemented both to simplify the standard workflow and to evaluate the RF coils performance. In this article, the theory which underlies the fundamental methods of CoSimPy is shown together with the basic framework of the library.
Results: In the paper, the reliability of CoSimPy is successfully tested against a full-wave electromagnetic simulations involving a reference setup. The library is made available under https://github.com/umbertozanovello/CoSimPy together with a detailed documentation providing guidelines and examples. CoSimPy is distributed under the Massachusetts Institute of Technology (MIT) license.
Conclusions: CoSimPy demonstrated to be an agile tool employable for Electromagnetic/Circuit cosimulations. Its distribution is meant to fulfil the needs of several researchers also avoiding duplication of effort in writing custom implementations. CoSimPy is under constant development and aims to represent a coworking environment where scientists can implement additional methods whose sharing can represent an advantage for the community. Finally, even if CoSimPy is designed with special focus on MRI, it represents an efficient and practical tool potentially employable wherever electronic devices made of radiative and circuitry components are involved.

1. Introduction

Electromagnetic (EM) simulation tools unquestionably represent a fundamental resource in antennas and electronic devices design [1–4]. Moreover, their role is becoming of great importance in solving electromagnetic compatibility issues for a generic device under test [5–7]. In many of these applications, radiative elements interact with supporting circuitries defining the final behaviour of the device. This is of specific concern in Magnetic Resonance Imaging (MRI) radiofrequency (RF) coil design. Except for particular cases [8,9], RF coils extensively deploy lumped circuitries for tuning, matching and, when needed, decoupling. Each of these elements has to be optimised to obtain the targeted performance of the RF coil. This means that, in principle, EM simulations have to be repeated for each tuning, matching and decoupling condition. Due to the strong interaction between the RF coil and the radiated subject, in most simulations a human body model, or at least a part of it, has to be included in the domain to obtain reliable results [10–12]. This obviously leads to a significant computational demand, limiting optimisations based on full EM simulations. EM/Circuit cosimulation represents an efficient and effective strategy to address the above limitations [13–16]. In this approach, a single multiport EM simulation is performed involving only the

[*] Corresponding author.
E-mail address: u.zanovello@inrim.it (U. Zanovello).

https://doi.org/10.1016/j.cmpb.2022.106684

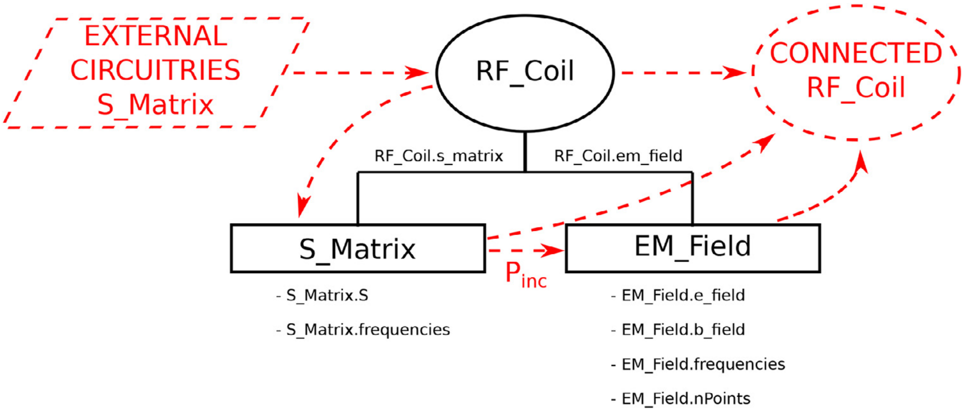


Fig. 1. Basic framework of the cosimulation library.

radiative parts of the RF coil. The S matrix and the associated EM field are extracted and used as input to the circuit simulation environment. Here, each port is replaced by the relevant circuitry and the S matrix of the connected RF coil, together with the generated EM field, is efficiently recomputed.

Whereas an EM/circuit cosimulation feature is provided by many commercial EM simulation bundles (*e.g.* CST, Ansoft, Agilent, etc.), to the best of the authors' knowledge, only one attempt has been made to provide the community with a freeware tool [17,18]. However, this tool has been implemented in Matlab which is not an open-source software. In this paper, CoSimPy, an open-source Python library developed for harmonic EM/circuit cosimulations, is presented.

After a multiport RF coil has been simulated and both the S matrix and generated EM field are available, its connection with another circuit can either concern each port separately (*single port connections*) or the RF coil as a whole (*full ports connection*). In *single port connections*, one or more RF coil ports are independently replaced by proper circuits described by their S matrices. Such circuits can either consist of single port lumped elements (*e.g.* tuning and decoupling capacitors) or multi-port circuits (*e.g.* T, PI matching networks). In *full ports connection*, all the ports can be involved simultaneously in the connection with a multiport external network [19,20]

CoSimPy can easily manage all these scenarios offering a handy tool for performing RF coil design and optimisation. In addition, a few limitations of the previous work [18] have been addressed (*e.g.* connections with multiport networks, resistors and inductors as external lumped elements). The library has been developed allowing easy scalability and implementation of custom methods.

In the following sections, the basic framework of the library is shown and the fundamental equations governing the main methods are reported. Finally, the reliability of the suggested implementation is tested against full-wave EM simulations and the results are reported.

2. Framework

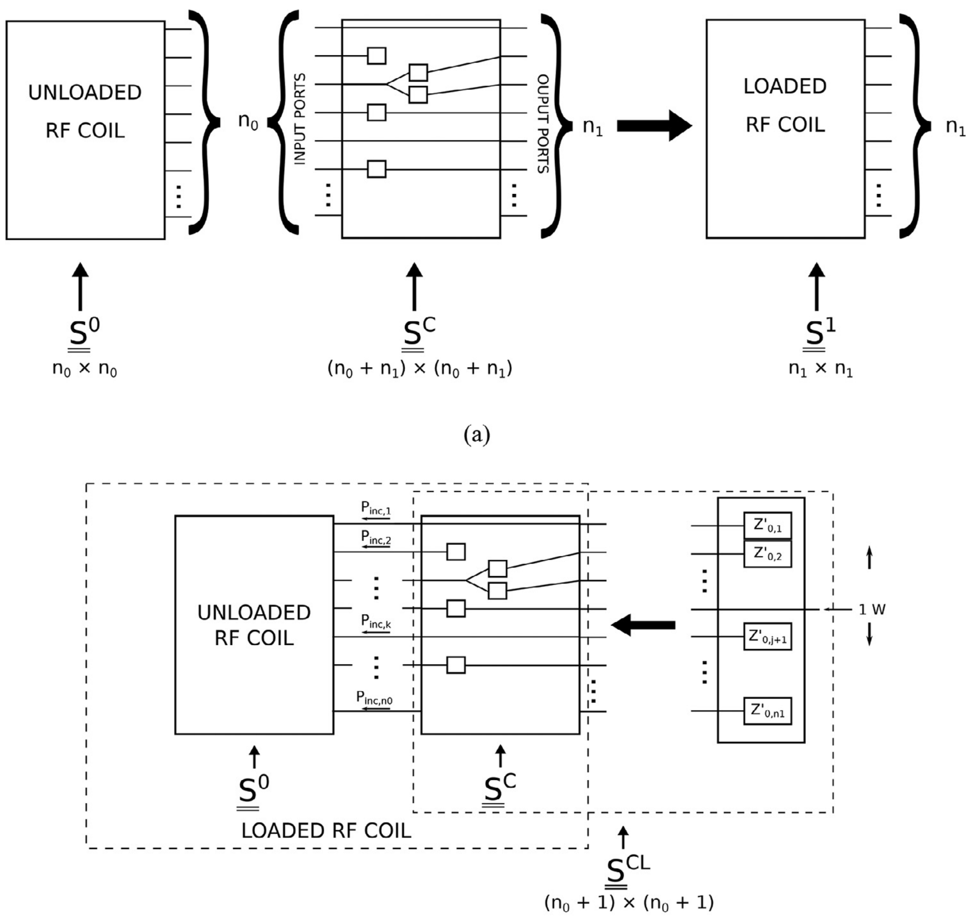
The basic framework of CoSimPy is sketched in Fig. 1 with solid black lines. *RF_Coil* is the main class of the library representing the simulated multiport device. An *RF_Coil* instance has two properties: *s_matrix* and *em_field*. The two properties are instances of the *S_Matrix* and *EM_Field* classes, respectively. The *S_Matrix* class is responsible for storing the S matrix of the simulated RF coil managing all the operations which can involve such a matrix (*e.g.* connection of the RF coil with other networks). This class has two basic properties, namely, *S* and *frequencies*. The former is an $n_{fs} \times n_p \times n_p$ array collecting all the S matrices associated with the n_p-port RF coil defined over n_{fs} frequency values. The latter is a one dimensional array listing all the n_{fs} frequency values over which the S matrices are defined. The *EM_Field* class is responsible for storing and managing the EM fields generated by the RF coil. This class has four main properties, namely, *e_field, b_field, frequencies* and *nPoints*. *e_field* and *b_field* are $n_{ff} \times n_p \times 3 \times n_{pt}$ arrays collecting the three Cartesian components of the electric field and magnetic flux density, respectively, defined over n_{ff} frequency values. A different EM field distribution is associated with each of the n_p ports of the RF coil corresponding to the EM field generated over n_{pt} Cartesian spatial points when the relevant port is supplied with 1 W incident power and all the other ports are closed on their characteristic impedance. The *nPoints* property is a 3-element list containing the number of spatial points along the three Cartesian dimensions of the simulated domain and allows to reconstruct the three-dimensional field distribution. The *frequencies* property is a one dimensional array listing the n_{ff} frequencies over which the EM field is defined. This array is a subset of the homonym property of the *S_Matrix* class.

The flow describing the connection between the RF coil and a generic external network, represented by an *RF_Coil* and *S_Matrix* instance respectively, is sketched in Fig. 1 by the red dashed lines. When a connection is performed, the *RF_Coil* instance invokes a dedicated method defined within the *S_Matrix* class. This method returns a new *S_Matrix* instance relevant to the connected RF coil and an array of complex coefficients, P_{inc}. Such coefficients represent the new powers incident to the n_p ports of the RF coil which are modified by the connection with the external circuitries. As such, they allow to obtain the EM field associated with the connected RF coil. Indeed, the coefficients P_{inc} serve as input for an *EM_Field* class method returning the appropriate *EM_Field* instance. Both the obtained *S_Matrix* and *EM_Field* instances become the properties of the new *RF_Coil* instance representing the connected RF coil.

3. Methods

The connection between the RF coil and a generic multiport network can be divided into two different stages. In the first stage the S matrix of the connected RF coil is computed. In the second stage, each port of the connected RF coil is supplied with 1 W incident power closing all the other ports on a characteristic impedance. The powers incident to all the ports of the unconnected RF coil are then computed. Such powers are then used to obtain the EM field generated by each port of the connected RF coil as shown below.

Even if, for the sake of practicality, two different methods are provided in CoSimPy for *single port connections* and *full ports con-*


UNLOADED
RF COIL
n_0
INPUT PORTS
OUPUT PORTS
n_1
LOADED
RF COIL
n_1
$\underline{\underline{S}}^0$
$n_0 \times n_0$
$\underline{\underline{S}}^C$
$(n_0 + n_1) \times (n_0 + n_1)$
$\underline{\underline{S}}^1$
$n_1 \times n_1$
UNLOADED
RF COIL
$P_{inc,1}$
$P_{inc,2}$
$P_{inc,k}$
$P_{inc,n0}$
$Z'_{0,1}$
$Z'_{0,2}$
$Z'_{0,j+1}$
$Z'_{0,n1}$
1 W
$\underline{\underline{S}}^0$
$\underline{\underline{S}}^C$
LOADED RF COIL
$\underline{\underline{S}}^{CL}$
$(n_0 + 1) \times (n_0 + 1)$

(a)

(b)

Fig. 2. S matrices connection.

nection, both the methods return the result by calling the same routine. This routine manages the connection as the cascade between the S matrix of the unconnected RF coil and another generic S matrix whose number of ports is higher than that of the unconnected RF coil. The procedure is sketched in Fig. 2a. $\underline{\underline{S}}^0$, $\underline{\underline{S}}^C$ and $\underline{\underline{S}}^1$ represent the S matrices associated with the unconnected RF coil, the supporting circuitries and the connected RF coil, respectively. n_0 and n_1 represent the number of ports of the unconnected and connected RF coil, and also correspond to the number of input and output ports of the supporting circuitries, respectively.

According to this procedure, if the ith port of the unconnected RF coil is not involved in the connection, it will be straight connected to the relevant oth output port. From the $\underline{\underline{S}}^C$ viewpoint, its ith row will be made of all zeros except 1 at the oth column and its oth row will be made of all zeros except 1 at the ith column.

In order to express $\underline{\underline{S}}^1$ as a function of $\underline{\underline{S}}^0$ and $\underline{\underline{S}}^C$, it is useful to split $\underline{\underline{S}}^C$ into four submatrices as it follows:

$$\underline{\underline{S}}^C = \begin{pmatrix} \underline{\underline{S}}^C_{11} & \underline{\underline{S}}^C_{12} \\ \underline{\underline{S}}^C_{21} & \underline{\underline{S}}^C_{22} \end{pmatrix} \tag{1}$$

where $\underline{\underline{S}}^C_{11}$ is a $n_0 \times n_0$ matrix, $\underline{\underline{S}}^C_{12}$ is a $n_0 \times n_1$ matrix, $\underline{\underline{S}}^C_{21}$ is a $n_1 \times n_0$ matrix and $\underline{\underline{S}}^C_{22}$ is a $n_1 \times n_1$ matrix. With reference to (1), $\underline{\underline{S}}^1$ is given by:

$$\underline{\underline{S}}^1 = \underline{\underline{S}}^C_{21}\left[\underline{\underline{S}}^{0^{-1}} - \underline{\underline{S}}^C_{11}\right]^{-1}\underline{\underline{S}}^C_{12} + \underline{\underline{S}}^C_{22} \tag{2}$$

There exist particular conditions where (2) cannot be used to compute $\underline{\underline{S}}^1$ since the inverse of the relevant matrices does not exist. One such example is $\underline{\underline{S}}^0$ representing a PI or T network fully composed of port characteristic impedances. In these cases, the algorithm tries to compute the impedance matrices $\underline{\underline{Z}}^0$ and $\underline{\underline{Z}}^C$ associated with $\underline{\underline{S}}^0$ and $\underline{\underline{S}}^C$. If $\underline{\underline{Z}}^C$ is divided in four submatrices following the same criteria used in (1), the impedance matrix of the connected RF coil can be obtained as:

$$\underline{\underline{Z}}^1 = \underline{\underline{Z}}^C_{22} - \underline{\underline{Z}}^C_{21}\underline{\underline{Z}}^{0^{-1}}\left[\underline{\underline{I}} - \underline{\underline{Z}}^C_{11}\underline{\underline{Z}}^{0^{-1}}\right]^{-1}\underline{\underline{Z}}^C_{12} \tag{3}$$

where $\underline{\underline{I}}$ is a $n_0 \times n_0$ identity matrix. $\underline{\underline{S}}^1$ is therefore obtained converting $\underline{\underline{Z}}^1$. If one or more impedance matrices that have to be inverted in (3) are singular, the algorithm raises a *Singular matrix* exception which can be handled externally to the library. In order to obtain the new EM fields generated by each port of the connected RF coil, it is necessary to compute the powers incident to all the n_0 ports of the unconnected RF coil when the n_1 ports of the connected RF coil are alternatively supplied with 1 W incident power and all the other n_1 - 1 ports are closed over a characteristic impedance. The new EM fields can be therefore obtained as the linear combination, weighted by the square root of the n_0 incident powers, of the EM fields generated by the n_0 ports of the unconnected RF coil [13].

Fig. 2b gives a graphical representation of the workflow when the EM field generated by the jth port of the connected RF coil is investigated. In Fig. 2b, the Z'_0 represent the characteristic impedances of the ports of the connected RF coil. Defining $\underline{\underline{S}}^{CL}$ as the S matrix resulting from the connection of $\underline{\underline{S}}^C$ and the n_1 - 1 characteristic impedances, it is possible to identify four submatrices:

$$\underline{\underline{S}}^{CL} = \begin{pmatrix} \underline{\underline{S}}^{CL}_{11} & \underline{\underline{S}}^{CL}_{12} \\ \underline{\underline{S}}^{CL}_{21} & \underline{\underline{S}}^{CL}_{22} \end{pmatrix} \tag{4}$$

where $\underline{\underline{S}}^{CL}_{11}$ is a $n_0 \times n_0$ matrix, $\underline{\underline{S}}^{CL}_{12}$ is a $n_0 \times 1$ matrix, $\underline{\underline{S}}^{CL}_{21}$ is a $1 \times n_0$ matrix and $\underline{\underline{S}}^{CL}_{22}$ is a 1×1 matrix. According to this definition, the voltage waves incident to the n_0 ports of the unconnected S matrix, when supplying the jth port of the connected RF coil with 1 W incident power and closing all the other ports on their characteristic impedances, can be obtained as:

$$\underline{V^{+}_{0,j}} = \left[\underline{\underline{I}} - \underline{\underline{S}}^{CL}_{11}\underline{\underline{S}}^{0}\right]^{-1}\underline{\underline{S}}^{CL}_{12}\sqrt{Z'_{0,j}P_{inc,j}} \tag{5}$$

where $V^{+}_{0,j}$ is a n_0 length vector, $Z'_{0,j}$ is a real value representing the characteristic impedance of the supplied port, $\underline{\underline{I}}$ is a $n_0 \times n_0$ identity matrix and $P_{inc,j}$ is the power incident to the jth port and it is equal to 1 W. The power incident to the kth port of the unconnected RF coil, can be obtained as:

$$P_{inc,j,k} = \frac{\left|V^{+}_{0,j,k}\right|^2}{Z_{0,k}} \tag{6}$$

where $V^{+}_{0,j,k}$ is the kth element of $\underline{V^{+}_{0,j}}$, $Z_{0,k}$ is a real value representing the characteristic impedance of the kth port of the unconnected RF coil.

The power in (6) can be "de-phased" with respect to the other powers, based on the phase of the corresponding incident voltage: $\angle V^{+}_{0,j,k}$. The complex factor that allows taking into account this de-phasing at the different ports is given by $e^{i\ \angle V^{+}_{0,j,k}}$ where i is the imaginary unit.

The powers incident to all the n_0 ports of the unconnected RF coil, when all the n_1 ports of the connected RF coil are alternatively supplied with 1 W incident power, can be collected, together with the relevant de-phasing factors, in a $n_0 \times n_1$ matrix $\underline{\underline{P}}_{inc}$. If the three Cartesian components of the magnetic flux density and electric field, generated in n_{pt} spatial points by the unconnected RF coil when the n_0 ports are alternatively supplied with 1 W incident power, are collected in a $3 \times n_{pt} \times n_0$ multidimensional matrix $\underline{\underline{\underline{F}}}^{0}$, the analogous matrix, $\underline{\underline{\underline{F}}}^{1}$, relevant to the connected RF coil, can be obtained as:

$$\underline{\underline{\underline{F}}}^{1} = \underline{\underline{\underline{F}}}^{0}\underline{\underline{P}}^{1/2}_{inc} \tag{7}$$

where, by definition, $\underline{\underline{P}}^{1/2}_{inc,j,k} = \sqrt{|P_{inc,j,k}|}e^{i\ \angle V^{+}_{0,j,k}}$ and the matrix multiplication is performed over the last two indices of $\underline{\underline{\underline{F}}}^{0}$ resulting in a $\underline{\underline{\underline{F}}}^{1}$ matrix whose dimension is $3 \times n_{pt} \times n_1$.

4. Results

To test the reliability of the cosimulation algorithm, the reference setup of Fig. 3 has been conceived. The setup consists of a 34 cm × 18 cm × 40 cm phantom (relative electric permittivity equal to 80 and electrical conductivity equal to 0.47 S/m) exposed to the RF field generated by two square loop RF coils (side length equal to 10 cm) sharing one conductor and placed 1 cm far from the phantom surface. In a first stage, a 7-port EM FDTD (finite-difference-time-domain) simulation has been performed with Sim4Life [21] over a frequency bandwidth equal to ±60 MHz with a central frequency equal to 128 MHz and 24 kHz spaced samples. The resulting 7 × 7 S matrix of the RF coil has been extracted together with the electric field and magnetic flux density inside the phantom. The EM field has been exported on an isotropic Cartesian grid with side equal to 4 mm and both the S parameters and EM field have been imported in the circuit simulation environment through specific CoSimPy routines.

In the following stage, each simulated port has been replaced, in the circuit simulation environment, by the lumped elements shown in Fig. 3. In particular, ports from 4 to 7 have been conceived for tuning the RF coils and have been replaced by the series of a tuning capacitor C_t and a resistor R. Port 3 has been conceived to decouple the two RF coils and has been replaced by a decoupling capacitor C_d. Finally, ports 1 and 2, which represent the actual supply ports of the two RF loop coils, have been connected to a 2-port matching network made of two capacitors $\mathrm{C_{m1}}$ and $\mathrm{C_{m2}}$ and a resistor. The output ports of the matching networks, $\mathrm{P_{N1}}$ and $\mathrm{P_{N2}}$ represent the supply ports of the final RF coils. The value for the resistances has been set to 1 Ω leading, approximately, to a ratio between the unloaded and loaded quality factors equal to 4 for the whole RF coil. A genetic optimisation algorithm [22] has been used, in combination with the cosimulation algorithm, to find the values of the four capacitors which maximise the total power accepted by the RF coils. The obtained capacitor values are C_t = 22 pF, C_d = 41 pF, $\mathrm{C_{m1}}$ = 240 pF, and $\mathrm{C_{m2}}$ = 38 pF.

The behaviour of the 2-port RF coil obtained in the circuit simulation environment has been compared with the outcomes obtained simulating the same 2-port device in Sim4Life.

Fig. 4 shows the comparison results. To provide an idea of how much the lumped elements, substituting the ports, affect the RF coil behaviour, the results related to the 7-port original simulation are also reported.

In Fig. 4a and b, both the absolute value and the phase of the $\mathrm{S_{11}}$ and $\mathrm{S_{12}}$ scattering parameters are compared. The results are shown only for $\mathrm{S_{11}}$ and $\mathrm{S_{12}}$ parameters since the problem is reciprocal and symmetric.

In Fig. 4c the power deposited on a phantom axial slice is reported as a function of its longitudinal index when port 1 is supplied with 1 W incident power and the remaining port is closed on its characteristic impedance.

Finally, in Fig. 4d the same comparison is performed for the value of the magnetic flux density along the longitudinal line crossing the centre of the phantom.

To give an example of a standard CoSimPy workflow, the code used to import the external data from Sim4Life, to generate the class instances, to connect the 7-port RF coil to the external circuitries and to extract the results in *numpy ndarrays*, is reported in the Appendix with reference to the problem analysed in this section.

5. Discussion

The matching between the results obtained from the circuit simulation environment and the 2-port simulation is satisfactory both in terms of S parameters ($\mathrm{S_{11}}$ average relative error equal to 0.7 % and 5% for the absolute values and phases respectively, $\mathrm{S_{12}}$ average relative error equal to 3 % and 0.8 % for the absolute values and phases respectively) and field quantities (average relative error equal to 0.4 % and 0.2 % for the electric field and magnetic flux density respectively).

The optimisation algorithm, involved in the estimation of the capacitor values for the 2-channel, 7-port reference problem, took about 5 min to run on an Intel® Core$^{\mathrm{TM}}$ i7–10,510 U CPU and about 191,500 iterations to meet the converge criteria (*i.e.* at least 150 iterations without any improvement). In each iteration, (2) was evaluated and the total accepted power was maximised by minimising the 1-norm of the vector composed by the real eigenvalues of the $\underline{\underline{S}}^{1^H}\underline{\underline{S}}^{1}$ matrix, where H means conjugate transpose [23]. In this case, the knowledge of the S matrix of the connected RF coil was sufficient to estimate the optimisation objective and the pow-

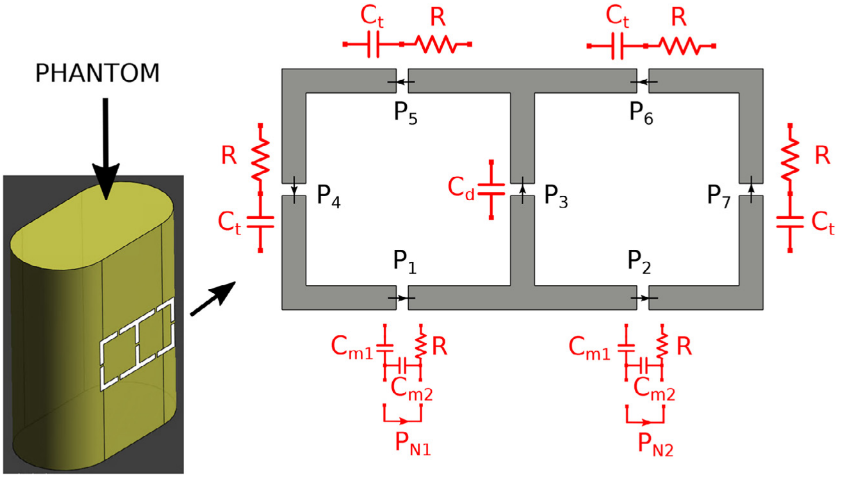
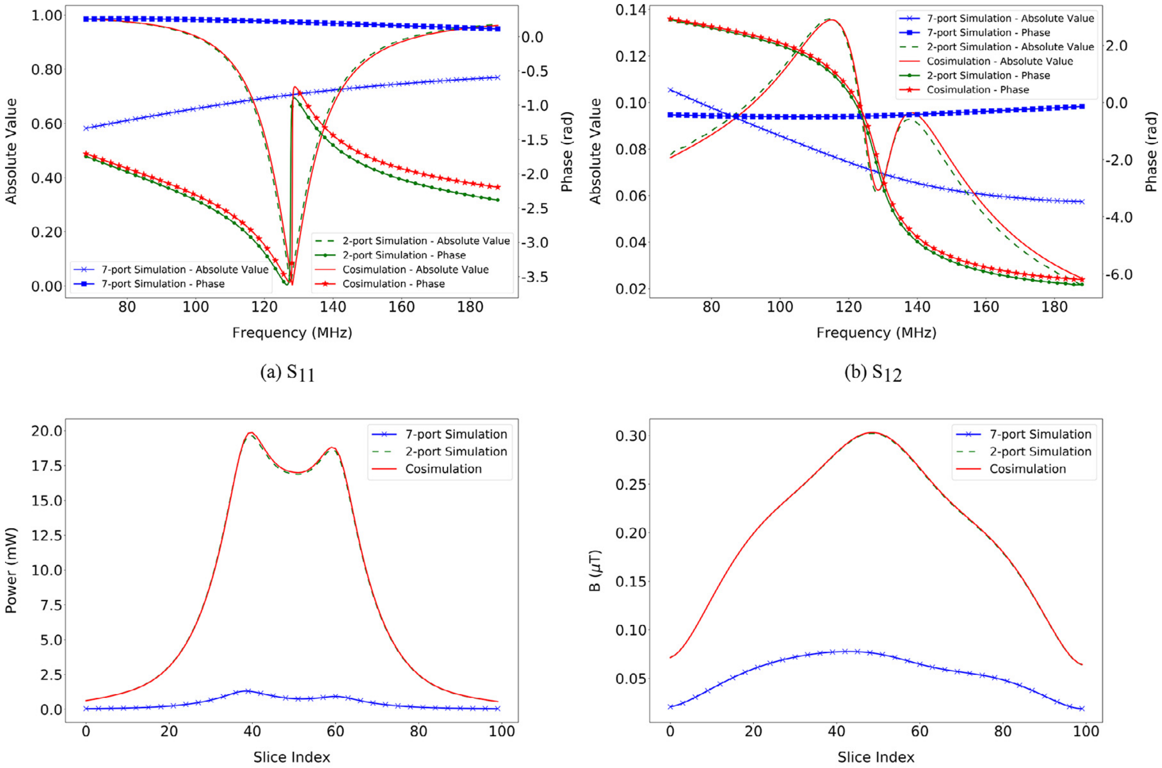


Fig. 3. Setup used for the validation of the cosimulation algorithm.



(a) S_{11}

(b) S_{12}

(c) Phantom slice power

(d) Magnetic flux density along the longitudinal line crossingthe centre of the phantom

Fig. 4. Comparison between cosimulation and simulation results.

ers incident to the unconnected RF coil ports were not calculated. In principle, there could exist circumstances where the EM field distribution is involved in the definition of the optimisation function. In such cases the knowledge of the S matrix of the connected RF coil is no more sufficient, the powers incident to the unconnected RF coil ports have to be computed with (5) and (6) and the new EM field distribution has to be obtained through (7). The increase of computation time depends on the number of ports of the unconnected/connected RF coil and the number of points in which the EM field is defined. With regard to the reference problem proposed in Section 4, the time needed by the optimisation algorithm for completing the same number of iterations, increased from 5 min to 2 h and 20 min. Such an increase is mainly due to (7) where the matrix multiplication involves a large $n_{pt} \times n_1$ matrix (382,500 × 7 in the reference problem) and has to be repeated for each component both of the electric field and magnetic flux density, for a total of six times. This however represents an "embarrassingly parallel" workflow and an improvement of the code involving a parallel strategy would be of reasonable effort and significant benefit. Furthermore, even if the reference RF coil has been optimised at a single frequency value, there could be cases where the RF coil is designed to work at multiple frequencies and its behaviour has to be consequently optimised [16,24]. Whereas CoSimPy is designed to deal with S matrices and EM fields defined at multiple frequencies, the code basically repeats all the operations for a number of times equal to the number of frequency val-

Table 1
Collection of CoSimPy input/output methods and relevant file formats.

	S_Matrix		EM_Field		RF_Coil	
	Method	Description	Method	Description	Method	Description
IMPORT	*importTouchstone*	The method generates an *S_Matrix* instance importing the S-, Z- or Y-parameters from a Touchstone file. Import file formats: • Rev. 1.1 IBIS Open Forum [25]; • Multi-column, multi-row ASCII file.	*importFields_s4l* *importFields_cst*	The method generates an *EM_Field* instance importing the EM field from ".mat" binary files. The files are generated by the Export Tool of Sim4Life, exporting the EM field on a regular grid. The method generates an *EM_Field* instance importing the EM field from: Multi-column, multi-row ASCII files; • ".h5" binary files. • The files are generated by the Export Tool of CST, exporting the EM field on a regular grid.	*saveRFCoil*	The method allows to save all the data relevant to the *RF_Coil* instance into a proprietary binary file with ".cspy" extension.
EXPORT	*exportTouchstone*	The method exports the S-, Z- or Y- parameters, defined within the *S_Matrix* instance, into a Touchstone file. Export file formats: • Rev. 1.1 IBIS Open Forum [25]; • Multi-column, multi-row ASCII file.	*exportXMF*	The method exports the EM field, defined within the *EM_Field* instance, in an ".h5" file. A ".xmf" file is also generated in combination with the previous ".h5" file.	*loadRFCoil*	The method allows to generate an *RF_Coil* instance loading all the data from a previously saved ".cspy" binary file.

ues. Even in this perspective, the code could be sensibly improved considering a parallelization strategy.

Along with the fundamental methods described in Section 3, many others have been implemented both to simplify and accelerate the workflow. Examples includes methods for importing and exporting the EM field distribution, importing and exporting the S parameters, saving and loading the *RF_Coil* instance according to a proprietary binary format, plotting the EM field distribution and the S parameters, computing voltages and currents at the RF coil ports, computing the RF coil TX/RX sensitivities, computing the electric power density and the total power deposited inside a non-zero conductivity load and computing the total power balance relevant to the RF coil (*i.e.* the relative amount of the power accepted, radiated, lost in supporting circuitries and deposited into a load). Table 1 collects all the input/output methods available along with the present version of CoSimPy. In the table, the generated/expected file formats are briefly described beside the relevant methods.

At the time of writing, other methods are being implemented and they will be available in the future versions of CoSimPy together with a parallelization of the code. These include methods for computing the Q matrix [26], for computing RF coil figures of merit (*e.g.* g-factor, filling factor [27,28]), for exporting the EM field distribution in different formats (at the time of writing, only a method has been implemented to export the EM field distribution in XMF format), for importing the EM field distribution according to the format adopted by different EM simulation software (only Sim4Life and CST are accounted at the time of writing) and for interacting with external EM simulation software through disposable APIs.

The main idea supporting the development of CoSimPy is to provide a free and handy tool to perform MRI RF coil EM/Circuit cosimulations also complementing existing EM simulation tools [29–32] already pursuing the "open-source imaging" philosophy [33] and other commercial tools where a circuit simulation interface is still missing [21]. Furthermore, CoSimPy is designed to be easily integrated in a homemade Python code (*e.g.* personalised optimisation algorithms, customised parametric analyses). In this perspective, CoSimPy could be useful even in those cases where the EM full-wave simulations are performed with a simulation bundle comprehensive of a circuit simulation interface [34,35].

To the best of the authors' knowledge, CoSimPy represents the only open-source alternative to an already existing EM/Circuit cosimulation Matlab-based tool [17,18]. In this regard, CoSimPy is completely Python-based, addresses some of the limitations of the previous tool and embeds a series of methods aimed to characterise an MRI RF coil behaviour and performance.

With respect to other open-source RF circuit simulation tools, not specifically intended for EM/Circuit cosimulations [36,37], CoSimPy differs not only for the implementation of the routines able to manage the interaction between supporting circuitries and generated EM field. Indeed, all the CoSimPy methods have been designed trying to make their use as straightforward as possible in the framework of MRI RF coil design and optimisation.

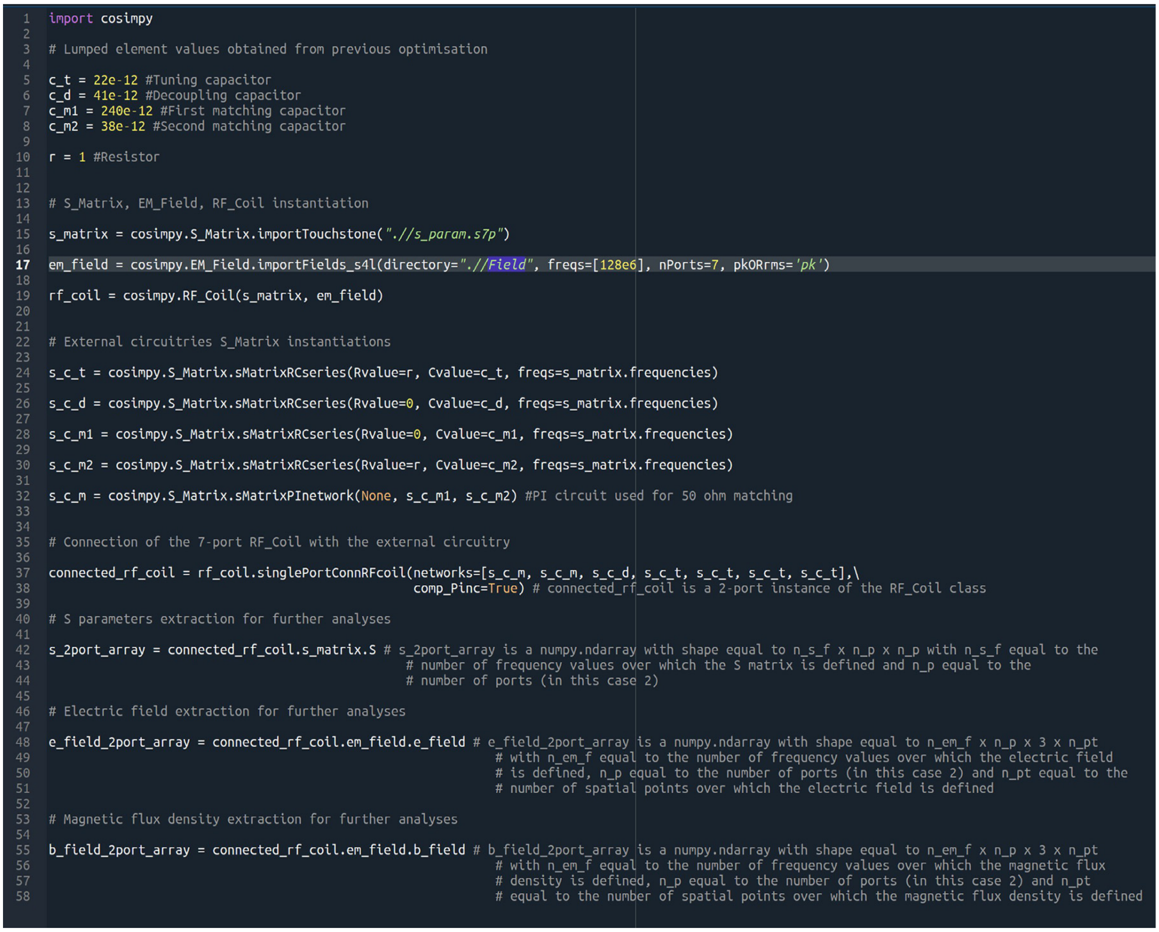
Finally, even if CoSimPy has been developed with the focus on MRI, it can be utilised, as it is, in many other contexts where a two-way link between EM and circuit simulations is needed (*e.g.* microwave engineering, hyperthermia, antenna design) [38–41].

6. Conclusion

In this paper, CoSimPy, an open-source Python library, developed for EM/circuit cosimulations, has been presented and successfully tested against full-wave simulations. The basic framework of the library has been shown together with the fundamental equations deployed by its main methods. A reference setup has been proposed to validate the library results comparing them with those obtained through an independent simulation.

CoSimPy has been designed with the aim of providing a free, practical and effective tool to perform MRI RF coil EM/Circuit cosimulations. Furthermore, to propose a coworking environment where scientists can implement additional methods whose sharing can represent an advantage for the community.

CoSimPy is available on the GitHub repository (https://github.com/umbertozanovello/CoSimPy). In addition to the source code, a



```
 1  import cosimpy
 2
 3  # Lumped element values obtained from previous optimisation
 4
 5  c_t = 22e-12 #Tuning capacitor
 6  c_d = 41e-12 #Decoupling capacitor
 7  c_m1 = 240e-12 #First matching capacitor
 8  c_m2 = 38e-12 #Second matching capacitor
 9
10  r = 1 #Resistor
11
12
13  # S_Matrix, EM_Field, RF_Coil instantiation
14
15  s_matrix = cosimpy.S_Matrix.importTouchstone(".//s_param.s7p")
16
17  em_field = cosimpy.EM_Field.importFields_s4l(directory=".//Field", freqs=[128e6], nPorts=7, pkORrms='pk')
18
19  rf_coil = cosimpy.RF_Coil(s_matrix, em_field)
20
21
22  # External circuitries S_Matrix instantiations
23
24  s_c_t = cosimpy.S_Matrix.sMatrixRCseries(Rvalue=r, Cvalue=c_t, freqs=s_matrix.frequencies)
25
26  s_c_d = cosimpy.S_Matrix.sMatrixRCseries(Rvalue=0, Cvalue=c_d, freqs=s_matrix.frequencies)
27
28  s_c_m1 = cosimpy.S_Matrix.sMatrixRCseries(Rvalue=0, Cvalue=c_m1, freqs=s_matrix.frequencies)
29
30  s_c_m2 = cosimpy.S_Matrix.sMatrixRCseries(Rvalue=r, Cvalue=c_m2, freqs=s_matrix.frequencies)
31
32  s_c_m = cosimpy.S_Matrix.sMatrixPInetwork(None, s_c_m1, s_c_m2) #PI circuit used for 50 ohm matching
33
34
35  # Connection of the 7-port RF_Coil with the external circuitry
36
37  connected_rf_coil = rf_coil.singlePortConnRFcoil(networks=[s_c_m, s_c_m, s_c_d, s_c_t, s_c_t, s_c_t, s_c_t],\
38                                                   comp_Pinc=True) # connected_rf_coil is a 2-port instance of the RF_Coil class
39
40  # S parameters extraction for further analyses
41
42  s_2port_array = connected_rf_coil.s_matrix.S # s_2port_array is a numpy.ndarray with shape equal to n_s_f x n_p x n_p with n_s_f equal to the
43                                                  # number of frequency values over which the S matrix is defined and n_p equal to the
44                                                  # number of ports (in this case 2)
45
46  # Electric field extraction for further analyses
47
48  e_field_2port_array = connected_rf_coil.em_field.e_field # e_field_2port_array is a numpy.ndarray with shape equal to n_em_f x n_p x 3 x n_pt
49                                                              # with n_em_f equal to the number of frequency values over which the electric field
50                                                              # is defined, n_p equal to the number of ports (in this case 2) and n_pt equal to the
51                                                              # number of spatial points over which the electric field is defined
52
53  # Magnetic flux density extraction for further analyses
54
55  b_field_2port_array = connected_rf_coil.em_field.b_field # b_field_2port_array is a numpy.ndarray with shape equal to n_em_f x n_p x 3 x n_pt
56                                                              # with n_em_f equal to the number of frequency values over which the magnetic flux
57                                                              # density is defined, n_p equal to the number of ports (in this case 2) and n_pt
58                                                              # equal to the number of spatial points over which the magnetic flux density is defined
```

Fig. A.1. Example of a typical workflow using CoSimPy. The code makes reference to the problem analysed in Section 4.

documentation is made available detailing the different methods, providing examples and guidelines for the installation.

Declaration of Competing Interest

All the authors disclose any financial and personal relationships with other people or organisations that could inappropriately influence the presented work

Acknowledgment

This project (RMG: 17IND01-RMG1 MIMAS) has received funding from the EMPIR programme co-financed by the Participating States and from the European Union's Horizon 2020 research and innovation programme.

Appendix

In this section, the essential part of the code used in the context of the reference problem proposed in Section 4, is reported. The two following preliminary arrangements have to be fulfilled:

1. The S parameters, obtained from the 7-port full-wave simulation, are saved according to the Touchstone file format specifications Rev. 1.1 provided by the IBIS Open Forum [25]. The Touchstone file, "s_param.s7p", is therefore saved in the same directory of the Python script;
2. A directory named "Field" is created in the same directory of the Python script. Inside the "Field" directory, the electromagnetic field distributions, generated by 1 W power incident to each of the seven ports of the simulated device, are stored. Each file is named "b(e)field_port<n>.mat", where "b" and "e" are used for magnetic and electric field distributions respectively and "<n>" denotes the port number (ranging, for the relevant case, from 1 to 7). These files are generated by Sim4Life on an isotropic Cartesian grid and represent the standard binary format used by Sim4Life for exporting the EM field distribution.

The code is shown in Fig. A.1. After importing the library, the capacitor and resistor values are declared (lines 5 – 10). Then, the S parameters and EM field distributions, previously computed through the full-wave simulation performed with Sim4Life, are imported (lines 15 and 17) and the *RF_Coil* instance of the 7-port device is generated (line 19). The *S_Matrix* instances relevant to the external circuitries are therefore defined (lines 24 – 32) and the 2-port *RF_Coil* instance is obtained connecting the original 7-port *RF_Coil* instance to the external circuitry (line 37). Finally, the S parameters and EM field distributions are extracted from the *S_Matrix* and *EM_Field* instances, respectively, into *numpy ndarray*, to allow for further manipulations external to CoSimPy (lines 37 – 55).

References

[1] S.D. Keller, W.D. Palmer, W.T. Joines, Electromagnetic modeling and simulation of a directly modulated patch antenna, IEEE Antennas Wirel. Propag. Lett. 9 (2010) 779–782.

[2] S. Koziel, A. Bekasiewicz, A structure and simulation-driven design of compact cpw-fed uwb antenna, IEEE Antennas Wirel. Propag. Lett. 15 (2016) 750–753.

[3] K. Syuhaimi, M. Fareq, Em/circuit cosimulation: a highly accurate method for microwave amplifier design, Univer. J. Comput. Sci. Eng. Technol. 1 (2) (2010) 127–132.

[4] M. Sarestoniemi, M. Hamalainen, J. Iinatti, An overview of the electromagnetic simulation-based channel modeling techniques for wireless body area network applications, IEEE Access 5 (2017) 10622–10632.
[5] O. Pacurar, A. Silaghi, A. De Sabata, Measurement and simulation of electrostatic discharge in the automotive industry, in: 2020 13th International Conference on Communications (COMM), 2020, pp. 443–446.
[6] X. Zhou, J. Li, H. Fan, A. Bhobe, P. Sochoux, J. Yu, High-frequency emc design verification through full-wave simulations and measurements in reverberation chamber, in: 2013 IEEE International Symposium on Electromagnetic Compatibility, 2013, pp. 299–305.
[7] L. Yang, C. Yang, Y. Tu, X. Wang, Q. Wang, Field-circuit co-simulation method for electrostatic discharge investigation in electronic products, IEEE Access 9 (2021) 33512–33521.
[8] A. Hurshkainen, M. Dubois, A. Nikulin, C. Vilmen, D. Bendahan, S. Enoch, S. Glybovski, R. Abdeddaim, Radio frequency coil for dual nuclei mr muscle energetics investigation based on two capacitively coupled periodic wire arrays, IEEE Antennas Wirel. Propag. Lett. 19 (5) (2020) 721–725.
[9] A.Priatna Basari, E.T. Rahardjo, F.Y. Zulkii, Numerical design of rf birdcage coil without lumped elements for mri 3t system, in: 2015 International Conference on Quality in Research (QiR), 2015, pp. 90–93.
[10] V. Kublanov, V. Borisov, M. Babich, Simulation the distribution of thermodynamic temperatures and microwave radiation of the human head, Comput. Methods Programs Biomed. 190 (2020) 105–377 01.
[11] J. Lotjonen, Construction of patient-specific surface models from mr images: application to bioelectromagnetism, Comput. Methods Programs Biomed. 72 (2) (2003) 167–178.
[12] F. Fanjul-Vélez, I. Salas-Garcia, N. Ortega-Quijano, J.L. Arce-Diego, Fdtd-based transcranial magnetic stimulation model applied to specific neurodegenerative disorders, Comput. Methods Programs Biomed. 118 (1) (2015) 34–43.
[13] M. Kozlov, R. Turner, Fast MRI coil analysis based on 3-D electromagnetic and RF circuit co-simulation, J. Magn. Reson. 200 (1) (2009) 147–152.
[14] F. Seifert, A. Cassara, G. Weidemann, B. Ittermann, Reliable and robust rf safety assessment of transmit array coils at ultrahigh fields, in: ISMRM Scientific Annual Meeting Proceedings (ISMRM, ed.), 22, 2014, p. 4891.
[15] A. Sadeghi-Tarakameh, E. Kazemivalipour, U. Gundogdu, S. Erdogan, E. Atalar, Accelerating the co-simulation method for the design of transmit array coils for mri, Magn. Reson. Mater. Phys., Biol. Med. 34 (2020) 165–178.
[16] N. Li, S. Liu, X. Hu, C. Luo, X. Zhang, Y. Li, Electromagnetic field and radio frequency circuit co-simulation for magnetic resonance imaging dual-tuned radio frequency coils, IEEE Trans. Magn. 54 (3) (2018) 1–4.
[17] A. Beqiri, J.W. Hand, J.V. Hajnal, S.J. Malik, Comparison between simulated decoupling regimes for specific absorption rate prediction in parallel transmit mri, Magn. Reson. Med. 74 (5) (2015) 1423–1434.
[18] A. Beqiri and S.J. Malik, “circuitcosimulation,” https://github.com/mriphysics/circuit_cosimulation/, 2015. [Online; accessed 14-May-2021].
[19] R.F. Lee, R.O. Giaquinto, C.J. Hardy, Coupling and decoupling theory and its application to the MRI phased array, Magn. Reson. Med. 48 (Jul 2002) 203–213.
[20] Z. Mahmood, P. McDaniel, B. Gu_erin, B. Keil, M. Vester, E. Adalsteinsson, L.L. Wald, L. Daniel, General design approach and practical realization of decoupling matrices for parallel transmission coils, Magn. Reson. Med. 76 (2016) 329–339.
[21] Z.M. Tech, “Sim4life,” https://zmt.swiss/, [Online; accessed 14-May-2021]. 2022
[22] R. Solgi, “geneticalgorithm,” https://github.com/rmsolgi/geneticalgorithm/, [Online; accessed 24-June-2021].
[23] E. Kazemivalipour, A. Sadeghi-Tarakameh, E. Atalar, Design of transmit array coils for mri by minimizing the modal refection coefficients, in: ISMRM Scientific Annual Meeting Proceedings ISMRM, 28, 2020, p. 762.
[24] A.W. Magill, R. Gruetter, Nested surface coils for multinuclear nmr, in: RF Coils for MRI, ch. 5, Wiley, 2012, pp. 39–50.
[25] IBIS Open forum, “Touchstone® file format specification Rev. 1.1.,” https://ibis.org/connector/touchstone_spec11.pdf, [Online; accessed 31-Jan-2022]. 2022
[26] I. Graesslin, H. Homann, S. Biederer, P. Bornert, K. Nehrke, P. Vernickel, G. Mens, P. Harvey, U. Katscher, A specific absorption rate prediction concept for parallel transmission mr, Magn. Reson. Med. 68 (5) (2012) 1664–1674.
[27] P.M. Robson, A.K. Grant, A.J. Madhuranthakam, R. Lattanzi, D.K. Sodickson, C.A. McKenzie, Comprehensive quantification of signal-to-noise ratio and g-factor for image-based and k-space-based parallel imaging reconstructions, Magn. Reson. Med. 60 (4) (2008) 895–907.
[28] R.A. Lemdiasov, A.A. Obi, R. Ludwig, A numerical postprocessing procedure for analysing radio frequency mri coils, Concept. Mag. Resonance Part A 38A (4) (2011) 133–147.
[29] J.F. Villena, A.G. Polimeridis, Y. Eryaman, E. Adalsteinsson, L.L. Wald, J.K. White, L. Daniel, Fast electromagnetic analysis of mri transmit rf coils based on accelerated integral equation methods, IEEE Trans. Biomed. Eng. 63 (11) (2016) 2250–2261.
[30] B. Guerin, J.F. Villena, A.G. Polimeridis, E. Adalsteinsson, L. Daniel, J.K. White, B.R. Rosen, L.L. Wald, Computation of ultimate sar amplification factors for radiofrequency hyperthermia in non-uniform body models: impact of frequency and tumour location, Int. J. Hyperthermia 34 (1) (2018) 87–100.
[31] A. Polimeridis, J. Villena, L. Daniel, J. White, Stable fft-jvie solvers for fast analysis of highly inhomogeneous dielectric objects, J. Comput. Phys. 269 (2014) 280–296.
[32] T. Liebig, A. Rennings, S. Held, 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, Int. J. Numer. Modell. Electron. Netw. Devices Fields 26 (6) (2013) 680–696.
[33] L. Winter, “opensourceimaging,” https://www.opensourceimaging.org/, [Online; accessed 25-June-2021].
[34] B.P. Tomas, H. Li, M.R. Anjum, Design and simulation of a birdcage coil using CST studio suite for application at 7t, IOP Conf. Ser.: Mater. Sci. Eng. 51 (2013) 012019 dec.
[35] R. Frass-Kriegl, S. Hosseinnezhadian, M. Poirier-Quinot, E. Laistler, J.-.C. Ginefri, Multi-loop radio frequency coil elements for magnetic resonance imaging: theory, simulation, and experimental investigation, Front. Phys. 7 (2020) 237.
[36] AD5GG, “Rf sim 99,” https://www.ad5gg.com/2017/04/06/free-rf-simulation-software/, [Online; accessed 2-September-2021].
[37] A. Arsenovic, L. Chen, M.F. Bauwens, H. Li, N.S. Barker, R.M. Weikle, An experimental technique for calibration uncertainty analysis, IEEE Trans. Microw. Theory Tech. 61 (1) (2013) 263–269.
[38] G. Zakka El Nashef, F. Torres, S. Mons, T. Reveyrand, T. Monediere, E. Ngoya, R. Quere, Em/circuit mixed simulation technique for an active antenna, IEEE Antennas Wirel. Propag. Lett. 10 (2011) 354–357.
[39] T. Wu, Z. Li, Y. Fan, Y. Wang, Y. Liu, T. Han, M. Tian, H. Zhou, H. Dong, L. Liu, A novel approach with field-circuit co-simulation and design of the receiver front- end, in: 2020 IEEE MTT-S International Wireless Symposium (IWS), 2020, pp. 1–3.
[40] D. Li, J. Zhu, S. Wu, X. Xiong, Y. Liu, Q.H. Liu, Field-circuit co-simulation for microwave metamaterials with nonlinear components, in: 2013 Proceedings of the International Symposium on Antennas Propagation, 02, 2013, pp. 1131–1133.
[41] L. Winter, C. Ozerdem, W. Homann, D. Santoro, A. Muller, H. Waiczies, R. Seemann, A. Graessl, P. Wust, T. Niendorf, Design and evaluation of a hybrid radiofrequency applicator for magnetic resonance imaging and rf induced hyperthermia: electromagnetic field simulations up to 14.0 tesla and proof-of-concept at 7.0 tesla, PLoS ONE 8 (2013) 1–12 04.