Multipole Perfectly Matched Layer for Finite-Difference Time-Domain electromagnetic modelling

Multipole Perfectly Matched Layer for Finite-Difference Time-Domain electromagnetic modelling

Antonios Giannopoulos

***Abstract*—A new multipole perfectly matched layer (PML) formulation is presented. Based on the stretched-coordinate approach the formulation, that utilises a recursive integration concept in its development, introduces a PML stretching function that is created as the sum of any given number of complex-frequency shifted (CFS) constituent poles. Complete formulae for up to a 3-pole formulation, to facilitate its implementation in finite-difference time-domain (FDTD) codes, are developed. The performance of this new multipole formulation compares favourably with existing higher order PMLs that instead utilise stretching functions that are developed as the product of elementary CFS constituent poles. It is argued that the optimisation of the new multipole PML could be more straightforward when compared to that of a higher order PML due to the absence of extra terms generated by the process of multiplication used in the development of the overall PML stretching function in higher order PMLs. The new multipole PML is found to perform very well when compared to standard CFS-PMLs requiring equivalent computational resources.**

***Index Terms*—Absorbing Boundary Conditions, Finite Difference Methods, Perfectly Matched Layer.**

I. Introduction

THIS paper presents a novel idea in creating more general perfectly matched layer (PML) stretching functions and a PML formulation to support their implementation in the finite-difference time-domain (FDTD) method which is widely used by the electromagnetic modelling community. The underlying design idea of this new general PML stretching function is based on a simple additive combination of different elementary complex frequency shifted (CFS) stretching functions. To use such a general stretching function a multipole PML (MPML) formulation is developed and tested.

From the inception of the PML, as a mechanism for outgoing wave absorption in terminating FDTD [1] [2] computational grids, the application of the PML resulted into a step-change in the use of FDTD for modelling complex electromagnetic problems. After the initial split-field formulation [3] and the independent introduction of the stretched-coordinate PML [4] [5] there have been a number of PML formulations for the FDTD method [6]. Soon it became apparent that although the PML was performing a lot better than local absorbing boundary conditions, based on approximations of one-way wave equations, it exhibited performance issues for a number of electromagnetic problems, especially ones involving evanescent waves [7] [8] and this was clearly demonstrated in wave-structure interaction problems [9]. The introduction of the complex frequency shifted PML (CFS-PML) [10] and its wider adoption [11] remedied somewhat the issues. A comprehensive review of the PML method in FDTD is given in [12].

The search however, for better performing PMLs continued and eventually led to the development of a second order PML formulation [13] in an effort to combine the benefits of the good absorption, offered primarily for body waves, by the original standard PML with the better performance of CFS-PML in cases where inhomogeneous waves were encountered. In [13] Correia and Jin introduced a split-field formulation of a 2nd order PML and since then unsplit FDTD formulations of 2nd order PMLs have been reported [14], [15] and a formulation for a general N-order PML was given in [16]. These have confirmed that at least a 2nd order PML can perform better than either a standard PML or a CFS-PML can do on their own, highlighting that increasing the order might benefit the PML absorption. Obviously, this increase in performance has to be weighted against the expected increase in the computational load. However, this was not taken into account when higher order PMLs where previously evaluated. One of the problems in developing a higher order PML is that the resulting stretching function is obtained as a product of individual CFS stretching functions. The underlying design idea for a higher order PML was that the combination will allow to reap the benefits of each term in the product. However, it is clear from a simple inspection that one will have to contemplate and deal with added terms resulting from multiplying together these stretching functions and as a result optimisation is not as straightforward [17]. In addition, careful consideration is required in examining new higher order stretching functions to ensure stability of the PML application [16].

The proposed formulation here is based on the simple combination of any number of required CFS stretching functions in a summation; hence calling it the multipole PML. The idea of using a summation draws parallels to the use of such multipole formulations in modelling the complex frequency dependent behaviour of materials (e.g. [18]). As no product of stretching functions is used in the development of the multipole PML no other extra terms as in the case of higher order PMLS are present. Hence, it is suggested that the optimisation of such a multipole PML should be in principle more straightforward when compared to the one of a higher order PML. Although,

A. Giannopoulos is with the School of Engineering, Institute for Infrastructure and Environment, The University of Edinburgh, Edinburgh, UK e-mail: A.Giannopoulos@ed.ac.uk

the derivation of the formulation might seem to be somewhat tedious the final result is simple to implement in existing FDTD codes [19] in a similar way as presented in [20] for the general CFS-PML and in [16] for higher order PMLs. The formulation presented here is given in a general form and for any required number of CFS poles and it does not add any more computational burden compared to existing higher order PML formulations. This paper focuses on the development of the MPML formulation leaving a detailed study of its optimisation for further research effort in future.

II. Theory

The fundamental concept of the PML is based on the use of a spatial complex stretching function that primarily through its imaginary part provides a mechanism for the effective attenuation of the electromagnetic waves inside the PML without significant reflection. In the continuous case a PML offers reflectionless transmission of the electromagnetic energy inside it in order to be effectively attenuated and eliminated. The numerical implementation is not reflectionless but it can be designed to offer great absorption performance. The original PML stretching function was of the form

$$s_{\text{std}} = \kappa + \frac{\sigma}{\jmath\omega\epsilon_0} \tag{1}$$

where often $\kappa = 1$ in most early uses. This can be shown to provide frequency independent attenuation for propagating waves where their wavenumber can be considered to be of the simple form ω/c. To mitigate problems with unsatisfactory PML performance involving inhomogeneous waves in a number of different settings [12] the complex frequency shifted (CFS) stretching function was introduced [10]. Defined as

$$s_{\text{cfs}} = \kappa + \frac{\sigma}{\alpha + \jmath\omega\epsilon_0} \tag{2}$$

it has been shown to improve PML performance in a number of cases. A further development was the introduction of higher order stretching functions in an attempt to enhance more the performance of PMLs. This was initially conceived to be best facilitated using a 2nd order stretching function of the form [13]

$$s_{\text{ho}} = s_{\text{std}} \times s_{\text{cfs}} = (\kappa_1 + \frac{\sigma_1}{\jmath\omega\epsilon_0})(\kappa_2 + \frac{\sigma_2}{\alpha_2 + \jmath\omega\epsilon_0}) \tag{3}$$

generalised to a Nth order PML in [16] as

$$s_{\text{ho}} = \prod_{i=1}^{N} \kappa_m + \frac{\sigma_m}{\alpha_m + \jmath\omega\epsilon_0} \tag{4}$$

In general, it is the imaginary part of the stretching function that is responsible for the attenuation offered by the PML [13]. Examining the imaginary parts of equations (1) , (2) and (3)

$$\Im(s_{\text{std}}) = \frac{\sigma}{\omega\epsilon_0} \tag{5}$$

$$\Im(s_{\text{cfs}}) = \frac{\sigma\omega\epsilon_0}{\alpha^2 + \omega^2\epsilon_0^2} \tag{6}$$

$$\Im(s_{\text{ho}}) = \frac{\kappa_2\sigma_1}{\omega\epsilon_0} + \frac{\kappa_1\epsilon_0\sigma_2\omega}{\alpha_2^2 + \epsilon_0^2\omega^2} + \frac{\alpha_2\sigma_1\sigma_2}{\epsilon_0\omega(\alpha_2^2 + \epsilon_0^2\omega^2)} \tag{7}$$

it is easily observed that the imaginary part of the higher order stretching function does not just provide the desired combination of that of a standard (1) and of a CFS (2) stretching function attenuation mechanisms, as given by (5) and (6), but it clearly introduces other terms that result from the process of multiplying the two stretching functions that provide, in this case, a 2nd order PML. In contrast, it is simple to show that a simple addition of the same constituent stretching functions given in (1) and (2) will have provided an imaginary part of the form

$$\Im(s) = \frac{\sigma_1}{\omega\epsilon_0} + \frac{\epsilon_0\sigma_2\omega}{\alpha_2^2 + \epsilon_0^2\omega^2} \tag{8}$$

which it seems that it is a better fit to the initial intention and design idea for combining different PML stretching functions. Although the higher order PML can be optimised to perform well [13] [16], and it has been demonstrated that can outperform same thickness simpler CFS-PMLs, the process of its optimisation is not as straightforward [17]. For example, an important constrain in selecting appropriate range of values for a 2nd order PML results from examining the real part of (3) that is easily obtained as

$$\Re(s_{\text{ho}}) = \kappa_2 + \frac{\sigma_2\alpha_2}{{\alpha_2}^2 + \omega^2{\epsilon_0}^2} - \frac{\sigma_1\sigma_2}{{\alpha_2}^2 + \omega^2{\epsilon_0}^2} \tag{9}$$

It is a requirement to ensure that $\Re(s_{\text{ho}}) \geq 1$ in order to guarantee stability as otherwise the PML introduces a real contraction of space [16]. This results in setting $\alpha_2 = \alpha_0 + \sigma_1$ [13] which is not really following the rationale of introducing the α parameter in CFS stretching functions. So, evidently the fact that extra terms, than the ones desired, are present in the combined 2nd order stretching function, that need to be taken into account, makes the 2nd order PML rather difficult to optimise in practice. This is a lot worst if higher order PMLs than the 2nd are to be considered.

In the following, a PML formulation based on a new multipole stretching function is proposed and its formulation is presented. Although, the concept is simple it appears that it has not been pursued before. This new PML termed the multipole PML (MPML) will have a stretching function of the form

$$s_{\text{mp}} = s_{\text{std}} + s_{\text{cfs}} = (\kappa_1 + \kappa_2) + \frac{\sigma_1}{\jmath\omega\epsilon_0} + \frac{\sigma_2}{\alpha_2 + \jmath\omega\epsilon_0} \tag{10}$$

or in the more general form

$$s_{\text{mp}} = \kappa + \sum_{m=1}^{N} \frac{\sigma_m}{\alpha_m + \jmath\omega\epsilon_0} \tag{11}$$

this will be shown that it compares favourably with higher order PMLs but it can be reasonably argued that it is easier to optimise and possibly to even extent to include a higher number of poles, if required, without having to consider as a cumbersome and restrictive optimisation as required by the higher order PML formulations. Further, selecting essential PML parameters (e.g.values for α_m) for the MPML follows the more familiar design idea and purpose according to their original development and introduction into PMLs (e.g. general inverse scaling of α values).

III. Formulation of the MPML

The development of the MPML follows a similar approach used in the development of the recursive integration PML (RIPML) as presented in [20] and [16] and uses similar underlying concepts.

Maxwell's equations in frequency domain and in stretched co-ordinates can be written compactly with the help of the cyclic notation $(i,j,k) \in (x,y,z),(y,z,x),(z,x,y)$ as

$$\jmath\omega\tilde{D}_i = \frac{1}{s_j}\frac{\partial \tilde{H}_k}{\partial j} - \frac{1}{s_k}\frac{\partial \tilde{H}_j}{\partial k} \tag{12}$$

$$\jmath\omega\tilde{B}_i = \frac{1}{s_k}\frac{\partial \tilde{E}_j}{\partial k} - \frac{1}{s_j}\frac{\partial \tilde{E}_k}{\partial j} \tag{13}$$

where s_u with $u \in (i,j,k)$ is a multipole CFS stretching function defined by

$$s_u = \kappa + \sum_{m=1}^{N} \frac{\sigma_{u_m}}{\alpha_{u_m} + \jmath\omega\epsilon_0} \tag{14}$$

where N is the number of general CFS poles that make up the multipole stretching function and its individual terms are of the general form presented originally in [10]. Introducing in (12) and (13) the simple variable transformation

$$\psi_u = \frac{1 - s_u}{s_u} \tag{15}$$

results in

$$\jmath\omega\tilde{D}_i = (1+\psi_j)\frac{\partial \tilde{H}_k}{\partial j} - (1+\psi_k)\frac{\partial \tilde{H}_j}{\partial k} \tag{16}$$

$$\jmath\omega\tilde{B}_i = (1+\psi_k)\frac{\partial \tilde{E}_j}{\partial k} - (1+\psi_j)\frac{\partial \tilde{E}_k}{\partial j} \tag{17}$$

Examining (16) and (17), in the PML region, they can be interpreted as the normal Maxwell's equations with added electric and magnetic field dependent currents as follows

$$\jmath\omega\tilde{D}_i = \frac{\partial \tilde{H}_k}{\partial j} - \frac{\partial \tilde{H}_j}{\partial k} + \tilde{J}_{ij} - \tilde{J}_{ik} \tag{18}$$

$$\jmath\omega\tilde{B}_i = \frac{\partial \tilde{H}_j}{\partial k} - \frac{\partial \tilde{H}_k}{\partial j} + \tilde{M}_{ik} - \tilde{M}_{ij} \tag{19}$$

where these currents are given by

$$\tilde{J}_{ij} = \psi_j \frac{\partial \tilde{H}_k}{\partial j}, \quad \tilde{J}_{ik} = \psi_k \frac{\partial \tilde{H}_j}{\partial k} \tag{20}$$

$$\tilde{M}_{ij} = \psi_j \frac{\partial \tilde{E}_k}{\partial j}, \quad \tilde{M}_{ik} = \psi_k \frac{\partial \tilde{E}_j}{\partial k} \tag{21}$$

The key concept in using the recursive integration idea is to efficiently calculate $\tilde{J}_{ij}$ and $\tilde{J}_{ik}$ in (20) and $\tilde{M}_{ij}$ and $\tilde{M}_{ik}$ in (21) leading to a simple implementation of the multipole PML as a correction that can be easily applied to the electromagnetic fields after updating the complete FDTD computational grid using standard FDTD equations. In this paper only the detailed procedure for obtaining $\tilde{J}_{ij}$ for a multipole PML is presented. The remaining field dependent electric and magnetic current terms found in (20) and (21) can be derived completely analogously.

Substituting in (20) $(1-s_j)/s_j$ for ψ_j results in

$$s_j\left(\tilde{J}_{ij} + \frac{\partial \tilde{H}_k}{\partial j}\right) = \frac{\partial \tilde{H}_k}{\partial j} \tag{22}$$

using ((14)) in ((20)) gives

$$\left(\kappa + \sum_{m=1}^{N} \frac{\sigma_{j_m}}{\alpha_{j_m} + \jmath\omega\epsilon_0}\right)\left(\tilde{J}_{ij} + \frac{\partial \tilde{H}_k}{\partial j}\right) = \frac{\partial \tilde{H}_k}{\partial j} \tag{23}$$

Defining a set of functions $\tilde{\Lambda}_{ij_m}$ for every $m \in [1,N]$ CFS pole as

$$\tilde{\Lambda}_{ij_m} = \left(\frac{\sigma_{j_m}}{\alpha_{j_m} + \jmath\omega\epsilon_0}\right)\left(\tilde{J}_{ij} + \frac{\partial \tilde{H}_k}{\partial j}\right) \tag{24}$$

allows $\tilde{J}_{ij}$ to be obtained by

$$\tilde{J}_{ij} = \frac{1-\kappa}{\kappa}\frac{\partial \tilde{H}_k}{\partial j} - \frac{1}{\kappa}\sum_{m=1}^{N} \tilde{\Lambda}_{ij_m} \tag{25}$$

It is obvious that because $\tilde{\Lambda}_{ij_m}$ contain itself terms involving $\tilde{J}_{ij}$ solving (25) is not simple. However, using the process of recursive integration for evaluating $\tilde{\Lambda}_{ij_m}$ allows for an efficient solution for $\tilde{J}_{ij}$, as it will be presented bellow. It is worth noting that balancing equation (25) in terms of the time instance that is evaluated (e.g. $n+\frac{1}{2}$) it is very important and an alternative formulation using an auxiliary differential equation approach will have resulted in difficulties to achieve such balance in a straightforward manner as $\tilde{\Lambda}_{ij_m}$ would have had to be evaluated half a time-step apart from $\tilde{J}_{ij}$.

Manipulating (24) algebraically results in

$$\alpha_{j_m}\tilde{\Lambda}_{ij_m} + \jmath\omega\epsilon_0\tilde{\Lambda}_{ij_m} = \sigma_{j_m}\tilde{J}_{ij} + \sigma_{j_m}\frac{\partial \tilde{H}_k}{\partial j} \tag{26}$$

following the main concept in developing the RIPML as discussed in [20] after dividing (26) by $\jmath\omega\epsilon_0$, rearranging and grouping similar terms gives

$$\tilde{\Lambda}_{ij_m} = \frac{1}{\jmath\omega}\Big[\frac{\sigma_{j_m}}{\epsilon_0}\tilde{J}_{ij} + \frac{\sigma_{j_m}}{\epsilon_0}\frac{\partial \tilde{H}_k}{\partial j} - \frac{\alpha_{j_m}}{\epsilon_0}\tilde{\Lambda}_{ij_m}\Big] \tag{27}$$

Transforming (27) into the time domain requires the evaluation of a time integral. However, there will be no need to calculate any time derivatives. So, transforming (27) into the time domain results in

$$\Lambda_{ij_m} = \int_0^{\tau} \frac{\sigma_{j_m}}{\epsilon_0}J_{ij} + \frac{\sigma_{j_m}}{\epsilon_0}\frac{\partial H_k}{\partial j} - \frac{\alpha_{j_m}}{\epsilon_0}\Lambda_{ij_m}\, d\tau \tag{28}$$

The magnetic field dependent currents $\tilde{J}_{ij}$ and $\tilde{J}_{ik}$ in (18) are evaluated at the same time instance as the magnetic field components $\tilde{H}_k$ and $\tilde{H}_j$ while the electric field dependent current $\tilde{M}_{ij}$ and $\tilde{M}_{ik}$ in (19) are evaluated at the same time instance as the electric field components $\tilde{E}_k$ and $\tilde{E}_j$. Therefore, after assuming that magnetic field components are evaluated at half steps (i.e. $n+1/2$) and that all field quantities

are zero for $t \leq 0$, applying the extended trapezoidal rule in (28) gives

$$\Lambda_{ij_m}^{n+1/2} = \sum_{p=0}^{n-1} \big[\frac{\sigma_{j_m}\Delta t}{\epsilon_0} J_{ij}^{p+1/2} + \frac{\sigma_{j_m}\Delta t}{\epsilon_0} \frac{\partial H_k^{p+1/2}}{\partial j} - \frac{\alpha_{j_m}\Delta t}{\epsilon_0} \Lambda_{ij_m}^{p+1/2} \big] + \frac{\sigma_{j_m}\Delta t}{2\epsilon_0} J_{ij}^{n+1/2} + \frac{\sigma_{j_m}\Delta t}{2\epsilon_0} \frac{\partial H_k^{n+1/2}}{\partial j} - \frac{\alpha_{j_m}\Delta t}{2\epsilon_0} \Lambda_{ij_m}^{n+1/2} \tag{29}$$

Rearranging ((29)) results in

$$\Big(1 + \frac{\alpha_{j_m}\Delta t}{2\epsilon_0}\Big) \Lambda_{ij_m}^{n+1/2} = \frac{\sigma_{j_m}\Delta t}{2\epsilon_0} J_{ij}^{n+1/2} + \frac{\sigma_{j_m}\Delta t}{2\epsilon_0} \frac{\partial H_k^{n+1/2}}{\partial j} + \Phi_{ij_m}^{n-1/2} \tag{30}$$

where the memory variable Φ_{ij_m} has been introduced to hold the summation

$$\Phi_{ij_m}^{n-1/2} = \sum_{p=0}^{n-1} \big[\frac{\sigma_{j_m}\Delta t}{\epsilon_0} J_{ij}^{p+1/2} + \frac{\sigma_{j_m}\Delta t}{\epsilon_0} \frac{\partial H_k^{p+1/2}}{\partial j} - \frac{\alpha_{j_m}\Delta t}{\epsilon_0} \Lambda_{ij_m}^{p+1/2} \big] \tag{31}$$

and therefore $\Lambda_{ij_m}^{n+1/2}$ can be obtained by

$$\Lambda_{ij_m}^{n+1/2} = \frac{\sigma_{j_m}\Delta t}{2\epsilon_0 + \alpha_{j_m}\Delta t} J_{ij}^{n+1/2} + \frac{\sigma_{j_m}\Delta t}{2\epsilon_0 + \alpha_{j_m}\Delta t} \frac{\partial H_k^{n+1/2}}{\partial j} + \frac{2\epsilon_0}{2\epsilon_0 + \alpha_{j_m}\Delta t} \Phi_{ij_m}^{n-1/2} \tag{32}$$

Considering the time domain version of (25) and the above result for Λ_{ij_m}, the required J_{ij} can be obtained by

$$J_{ij}^{n+1/2} = \frac{1-\kappa}{\kappa} \frac{\partial H_k^{n+1/2}}{\partial j} - \frac{1}{\kappa} \sum_{m=1}^{N} \frac{\sigma_{j_m}\Delta t}{2\epsilon_0 + \alpha_{j_m}\Delta t} J_{ij}^{n+1/2} - \frac{1}{\kappa} \sum_{m=1}^{N} \frac{\sigma_{j_m}\Delta t}{2\epsilon_0 + \alpha_{j_m}\Delta t} \frac{\partial H_k^{n+1/2}}{\partial j} - \frac{1}{\kappa} \sum_{m=1}^{N} \frac{2\epsilon_0 \Phi_{ij_m}^{n-1/2}}{2\epsilon_0 + \alpha_{j_m}\Delta t} \tag{33}$$

It is easily observed that $J_{ij}^{n+1/2}$ and the spatial derivative of the magnetic field $\partial H_k^{n+1/2}/\partial j$ are constant factors with respect to the summation index. Therefore, they can be brought outside the sum and hence $J_{ij}^{n+1/2}$ can be finally obtained, after some algebraic manipulations, in an easy to use form as

$$J_{ij}^{n+1/2} = \Big(\frac{1}{\kappa + \sum_{m=1}^{N} \frac{\sigma_{j_m}\Delta t}{2\epsilon_0 + \Delta t \alpha_{j_m}}} - 1 \Big) \frac{\partial H_k^{n+1/2}}{\partial j} - \Big(\frac{1}{\kappa + \sum_{m=1}^{N} \frac{\sigma_{j_m}\Delta t}{2\epsilon_0 + \alpha_{j_m}\Delta t}} \Big) \sum_{m=1}^{N} \frac{2\epsilon_0 \Phi_{ij_m}^{n-1/2}}{2\epsilon_0 + \alpha_{j_m}\Delta t} \tag{34}$$

The update of the $\Phi_{ij_m}^{n+1/2}$ memory variable follows by inspecting (31) and can be easily be written as

$$\Phi_{ij_m}^{n+1/2} = \Phi_{ij_m}^{n-1/2} + \frac{\sigma_{j_m}\Delta t}{\epsilon_0} J_{ij}^{n+1/2} + \frac{\sigma_{j_m}\Delta t}{\epsilon_0} \frac{\partial H_k^{n+1/2}}{\partial j} - \frac{\alpha_{j_m}\Delta t}{\epsilon_0} \Lambda_{ij_m}^{n+1/2} \tag{35}$$

It is obvious that getting a simple expression for $\Phi_{ij_m}^{n+1/2}$ requires substitutions first for $\Lambda_{ij_m}^{n+1/2}$ given by (32) and then for $J_{ij}^{n+1/2}$ as calculated by (34). After some tedious but simple algebraic manipulations $\Phi_{ij_m}^{n+1/2}$ can be obtained by

$$\Phi_{ij_m}^{n+1/2} = \frac{2\epsilon_0 - \alpha_{j_m}\Delta t}{2\epsilon_0 + \alpha_{j_m}\Delta t} \Phi_{ij_m}^{n-1/2} + \Big(\frac{2\Delta t \sigma_{j_m}}{2\epsilon_0 + \alpha_{j_m}\Delta t} \Big) \Big(\frac{1}{\kappa + \sum_{l=1}^{N} \frac{\sigma_{j_l}\Delta t}{2\epsilon_0 + \Delta t \alpha_{j_l}}} \Big) \frac{\partial H_k^{n+1/2}}{\partial j} - \Big(\frac{2\Delta t \sigma_{j_m}}{2\epsilon_0 + \alpha_{j_m}\Delta t} \Big) \Big(\frac{1}{\kappa + \sum_{l=1}^{N} \frac{\sigma_{j_l}\Delta t}{2\epsilon_0 + \Delta t \alpha_{j_l}}} \Big) \sum_{l=1}^{N} \frac{2\epsilon_0 \Phi_{ij_l}^{n-1/2}}{2\epsilon_0 + \alpha_{j_l}\Delta t} \tag{36}$$

In order to develop simple formulae for MPML implementations it is useful to define the following variables that are only computed once and depend only on the properties of s_u, the MPML stretching function

$$\begin{aligned} \mathrm{RA}_j &= \kappa + \sum_{m=1}^{N} \frac{\sigma_{j_m}\Delta t}{2\epsilon_0 + \alpha_{j_m}\Delta t} \\ \mathrm{RB}_{j_m} &= \frac{2\epsilon_0}{2\epsilon_0 + \alpha_{j_m}\Delta t} \\ \mathrm{RE}_{j_m} &= \frac{2\epsilon_0 - \alpha_{j_m}\Delta t}{2\epsilon_0 + \alpha_{j_m}\Delta t} \\ \mathrm{RF}_{j_m} &= \frac{2\Delta t \sigma_{j_m}}{2\epsilon_0 + \alpha_{j_m}\Delta t} \end{aligned}$$

with these (34) is presented compactly and simply as

$$J_{ij}^{n+1/2} = \Big\{ \frac{1}{\mathrm{RA}_j} - 1 \Big\} \frac{\partial H_k^{n+1/2}}{\partial j} - \frac{1}{\mathrm{RA}_j} \sum_{l=1}^{N} \mathrm{RB}_{j_l} \Phi_{ij_l}^{n-1/2} \tag{37}$$

and (36) can be similarly simplified as

$$\Phi_{ij_m}^{n+1/2} = \mathrm{RE}_{j_m} \Phi_{ij_m}^{n-1/2} + \frac{\mathrm{RF}_{j_m}}{\mathrm{RA}_j} \Big\{ \frac{\partial H_k^{n+1/2}}{\partial j} - \sum_{l=1}^{N} \mathrm{RB}_{j_l} \Phi_{ij_l}^{n-1/2} \Big\} \tag{38}$$

and then used to derive individual update equations for each memory variable Φ_{ij_m} as required.

One summation memory variable Φ_{ij_m} is required for every pole in the multipole PML presented here. Therefore, the memory footprint of this multipole PML formulation is the same as the one required by the higher order PML formulations presented in [16] . Φ_{ij_m} can simply be updated in the same computational loop after the application of the multipole PML as a "correction procedure" to the FDTD field components in the PML regions by using (37). Λ_{ij_m} is not required any more and there is no need to explicitly calculate

it or store it. Its use supported the development of the concept for the multipole PML but is not required by its numerical implementation in an FDTD algorithm. Similarly, J_{ij} does not need to be explicitly computed as the RHS of (37) can be used directly as needed. However, if desired it can be calculated "on the fly" and does not require additional computer memory.

A. One-pole MPML

Assuming that the PML stretching function s_u has only one pole, which is equivalent to the standard s_{cfs} ((2)), and is of the form

$$s_u = \kappa + \frac{\sigma_1}{\alpha_1 + \jmath\omega\epsilon_0} \tag{39}$$

the updates for $J_{ij}^{n+1/2}$ and $\Phi_{ij_1}^{n+1/2}$ are simply written as

$$J_{ij}^{n+1/2} = \left\{\frac{1}{\text{RA}_j} - 1\right\}\frac{\partial H_k^{n+1/2}}{\partial j} - \frac{1}{\text{RA}_j}\text{RB}_{j_1}\Phi_{ij_1}^{n-1/2} \tag{40}$$

followed by the update of $\Phi_{ij_1}^{n+1/2}$

$$\Phi_{ij_1}^{n+1/2} = \text{RE}_{j_1}\Phi_{ij_1}^{n-1/2} + \frac{\text{RF}_{j_1}}{\text{RA}_j}\left\{\frac{\partial H_k^{n+1/2}}{\partial j} - \text{RB}_{j_1}\Phi_{ij_1}^{n-1/2}\right\} \tag{41}$$

These formulae, as expected, result to a first order PML implementation that is exactly equivalent to the one reported in [20] and [16]. The Φ_{ij_1} used here is κ times the Φ_{ij_1} employed in the development of the higher order RIPML of [16]. By taking this into account the two formulations become identical. Numerical tests have verified this assertion.

B. Two-pole MPML

Assuming that the PML stretching function s_u has two poles, which is equivalent to having two s_{cfs}, or by setting $\alpha_1 = 0$ an additive combination of a standard streching function s_{std} with a s_{cfs}. The overall general form of s_u becomes

$$s_u = \kappa + \frac{\sigma_1}{\alpha_1 + \jmath\omega\epsilon_0} + \frac{\sigma_2}{\alpha_2 + \jmath\omega\epsilon_0} \tag{42}$$

and $J_{ij}^{n+1/2}$ is given by

$$J_{ij}^{n+1/2} = \left\{\frac{1}{\text{RA}_j} - 1\right\}\frac{\partial H_k^{n+1/2}}{\partial j} - \frac{1}{\text{RA}_j}\Big\{\text{RB}_{j_1}\Phi_{ij_1}^{n-1/2} + \text{RB}_{j_2}\Phi_{ij_2}^{n-1/2}\Big\} \tag{43}$$

and the updates for $\Phi_{ij_1}^{n+1/2}$ and $\Phi_{ij_2}^{n+1/2}$ can follow as

$$\Phi_{ij_1}^{n+1/2} = \text{RE}_{j_1}\Phi_{ij_1}^{n-1/2} + \frac{\text{RF}_{j_1}}{\text{RA}_j}\Big\{\frac{\partial H_k^{n+1/2}}{\partial j} - \text{RB}_{j_1}\Phi_{ij_1}^{n-1/2} - \text{RB}_{j_2}\Phi_{ij_2}^{n-1/2}\Big\} \tag{44}$$

$$\Phi_{ij_2}^{n+1/2} = \text{RE}_{j_2}\Phi_{ij_2}^{n-1/2} + \frac{\text{RF}_{j_2}}{\text{RA}_j}\Big\{\frac{\partial H_k^{n+1/2}}{\partial j} - \text{RB}_{j_1}\Phi_{ij_1}^{n-1/2} - \text{RB}_{j_2}\Phi_{ij_2}^{n-1/2}\Big\} \tag{45}$$

It should be noted that this 2-pole PML is not equivalent to the 2nd order PML presented in [16], [13] and [14]

C. Three-pole MPML

To develop a multipole PML with three poles the stretching function s_u takes the form

$$s_u = \kappa + \frac{\sigma_1}{\alpha_1 + \jmath\omega\epsilon_0} + \frac{\sigma_2}{\alpha_2 + \jmath\omega\epsilon_0} + \frac{\sigma_3}{\alpha_3 + \jmath\omega\epsilon_0} \tag{46}$$

and $J_{ij}^{n+1/2}$ can be easily obtained as

$$J_{ij}^{n+1/2} = \left\{\frac{1}{\text{RA}_j} - 1\right\}\frac{\partial H_k^{n+1/2}}{\partial j} - \frac{1}{\text{RA}_j}\Big\{\text{RB}_{j_1}\Phi_{ij_1}^{n-1/2} + \text{RB}_{j_2}\Phi_{ij_2}^{n-1/2} + \text{RB}_{j_3}\Phi_{ij_3}^{n-1/2}\Big\} \tag{47}$$

the updates for $\Phi_{ij_1}^{n+1/2}$, $\Phi_{ij_2}^{n+1/2}$ and $\Phi_{ij_3}^{n+1/2}$ can follow as

$$\Phi_{ij_1}^{n+1/2} = \text{RE}_{j_1}\Phi_{ij_1}^{n-1/2} + \frac{\text{RF}_{j_1}}{\text{RA}_j}\Big\{\frac{\partial H_k^{n+1/2}}{\partial j} - \text{RB}_{j_1}\Phi_{ij_1}^{n-1/2} - \text{RB}_{j_2}\Phi_{ij_2}^{n-1/2} - \text{RB}_{j_3}\Phi_{ij_3}^{n-1/2}\Big\} \tag{48}$$

$$\Phi_{ij_2}^{n+1/2} = \text{RE}_{j_2}\Phi_{ij_2}^{n-1/2} + \frac{\text{RF}_{j_2}}{\text{RA}_j}\Big\{\frac{\partial H_k^{n+1/2}}{\partial j} - \text{RB}_{j_1}\Phi_{ij_1}^{n-1/2} - \text{RB}_{j_2}\Phi_{ij_2}^{n-1/2} - \text{RB}_{j_3}\Phi_{ij_3}^{n-1/2}\Big\} \tag{49}$$

$$\Phi_{ij_3}^{n+1/2} = \text{RE}_{j_3}\Phi_{ij_3}^{n-1/2} + \frac{\text{RF}_{j_3}}{\text{RA}_j}\Big\{\frac{\partial H_k^{n+1/2}}{\partial j} - \text{RB}_{j_1}\Phi_{ij_1}^{n-1/2} - \text{RB}_{j_2}\Phi_{ij_2}^{n-1/2} - \text{RB}_{j_3}\Phi_{ij_3}^{n-1/2}\Big\} \tag{50}$$

Looking to increase computational efficiency, it is easy to note that the same terms in curly brackets { } are appearing in the calculation of both $J_{ij}^{n+1/2}$ and in the updates of all $\Phi_{ij_m}^{n+1/2}$ memory variables. So, these can be easily stored in a temporary variable in the beginning of the update process and reused efficiently simplifying greatly the computation. Further computational gains can be achieved by noting the common occurrence of $\text{RB}_{j_l}\Phi_{ij_l}^{n-1/2}$ and using the memory variable to hold that quantity instead of simply $\Phi_{ij_l}^{n-1/2}$.

It is important to note that contrary to the implementation of higher order PMLs [13] [16] after the correction of the corresponding field component by $J_{ij}^{n+1/2}$ the update of the memory variables $\Phi_{ij_m}^{n+1/2}$ does not have to proceed in any particular order. In terms of an FDTD implementation the PML dependent currents J_{ij}, J_{ik}, M_{ij} and M_{ik} need only to be applied in the PML regions after the normal update of all the fields in the FDTD grid. In terms of memory requirements N extra memory variables – one per pole – are required per stretched co-ordinate derivative. So, a two-pole MPML requires twice the memory of the standard CFS-PML formulation. The formulation is equally applicable with the standard PML stretching function by just letting $\alpha_m = 0$.

However, it is easily observable from the form of s_u that setting $\alpha_m = 0$ for more than one of the poles is wasting computer memory and resources.

IV. Numerical Results

To demonstrate the feasibility of a multipole PML as an alternative to a higher order PML formulation the numerical tests used in [16], as adopted from [13], are used here to test the proposed new MPML formulation. These tests are by no means exhaustive, they have been chosen for reasons of maintaining consistency in comparing with results obtained from previous similar works and because they have been used in the past for testing PML performance [2]. Parameters for CFS-PMLs and 2nd order PMLs have been obtained from [16] but optimum PML parameters for the multipole PMLs used here have been obtained using a trial and error approach.

Although, the computational requirements of a multipole PML are almost identical to the ones of an equivalent higher order PML, this clearly is not the case for comparisons with standard CFS-PMLs which clearly require at least less memory storage. In Table I a relative comparison of computational load in terms of computing time and memory storage, for a number of CFS-PMLs of various thickness is given. The basis for the comparison are the computational effort and memory storage required by an implementation of a 2-pole MPML. Negative numbers indicate a more efficient CFS-PML formulation for the two criteria. A 1-pole PML formulation has been used for the standard CFS-PML calculations. This is equivalent to the RIPML implementation of the CFS-PML [20]. The results of the comparison are restricted to the two examples presented here and used to inform the selection of test cases for evaluating relative PML performance. From analysing the results presented in Table I, it emerges that a 14-cell CFS-PML is more storage efficient from a 10-cell 2-pole PML whilst a 15-cell one is only slightly more demanding. However, for the 3D example only a 13-cell CFS-PML is more efficient, in terms of storage, from a 10-cell 2-pole MPML whilst a 14-cell one being marginally less storage efficient. However, interestingly when computational effort (i.e. run time) is considered it appears that only a 12-cell CFS-PML is more efficient from the 10-cell 2-pole MPML in 2D without any marginal differences and when considering the 3D case, only an 11-cell CFS-PML requires less computational time than the 10-cell 2-pole MPML.

When the size of the FDTD computational grid increases the PML becomes a smaller fraction of its dimensions which is not the case in the examples tested here. In such cases the thickness of CFS-PMLs that are more efficient than a 2-pole MPML, in terms of storage, could increase by 1 cell from the above given figures. However, the analogies relating to the computational effort do not change. The reason for this discrepancy between memory storage requirements and computational effort between 2-pole MPMLs and expanding standard CFS-PMLs, is easily understood considering that for every extra PML cell the computational grid should increase analogously and the computation of three, in the case of 2D domains, and of six, in the case of 3D ones, FDTD update equations are needed. Although, a 2-pole MPML doubles the memory storage footprint of the PML itself, it does not require the expansion of the computational domain, which is clearly needed for thicker CFS-PMLs. In addition, a 2-pole MPML requires modest additional computational effort, compared to a CFS-PML, than the effort required for updating of the FDTD field equations that are needed in addition to the extra PML cells when the size of the PML is increased. This is easy to establish comparing the one-pole MPML and the two-pole MPML equations presented here. Further, if more complex FDTD update equations need to be considered in such a comparison (e.g. when modelling frequency dispersive materials) the 2-pole MPML will have a further advantage as its computational load in terms of both storage and run time is independent of the nature of the underlying FDTD update equations.

TABLE I
Relative Computational Requirements of CFS-PMLs Compared to a 2-Pole 10-Cell MPML

PML Thickness	Time 2D	Storage 2D	Time 3D	Storage 3D
10 cells	-16.97%	-32.47%	-16.42%	-35.66%
11 cells	-9.11%	-26.22%	-3.96%	-27.50%
12 cells	-2.17%	-19.9%	+15.09%	-18.84%
13 cells	+5.01%	-13.3%	+32.26%	-9.67%
14 cells	+13.9%	-6.64%	+53.40%	+0.02%
15 cells	+33.56%	+0.15%	+76.79%	+10.24%

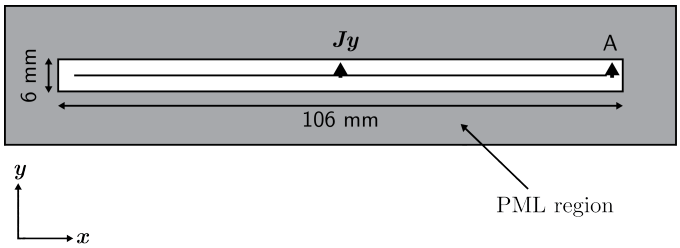

Fig. 1. TE_z FDTD model of a y-directed electric current source at the centre of a 126×26 1 mm cell TE_z. A 10-cell thick PML is included in the model. The sampling point A for the E_y field is located at the edge of the PEC sheet and is 3 cells away from the PML boundary.

In the following the more efficient CFS-PMLs in terms of memory storage and computational effort are compared to 2-pole MPMLs and 2nd order PMLs using the 2D and 3D modelling examples. In addition, comparisons with a standard CFS-PML of the same thickness as the 2-pole MPML are given as well as used in [16].

A. Line Source over finite 2D PEC Sheet

The 2D numerical test, as illustrated in Fig. (1), is based on a TE_z FDTD grid containing a y-directed electric current source J_y centred over a 100 cell wide PEC sheet. The time dependent current of the source is defined as

$$I(t) = -2\frac{(t-t_0)}{t_w}e^{-(\frac{t-t_0}{t_w})^2} \tag{51}$$

where $t_w = 26.53\,\text{ps}$ and $t_0 = 4t_w$.

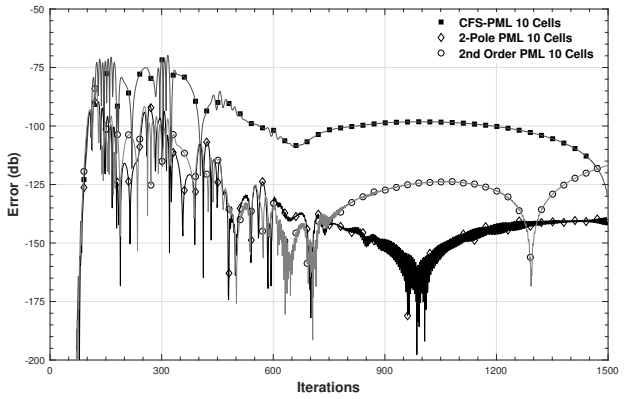

Fig. 2. Error in the E_y field component at point A for 2D TE$_z$ PEC sheet models terminated using a 10-cell CFS-PML, a 2nd Order RIPML as developed in [16] and a 2-pole MPML.

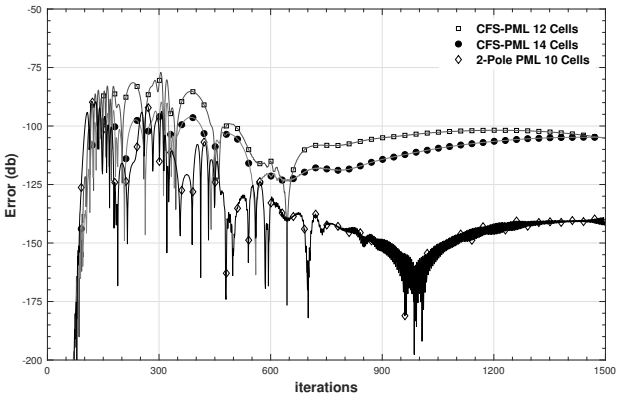

Fig. 3. Error in the E_y field component at point A for 2D TE$_z$ PEC sheet model using a 10-Cell 2-pole MPML and a 12-Cell and a 14-Cell CFS-PMLs.

To build the FDTD model a uniform 1mm square cell was used and the time step Δt was set to be $\Delta t = 0.99\Delta l/c\sqrt{(2)}$. The electric field E_y is sampled three cells away from the PML boundary. A reference solution was obtained using a large FDTD grid having truncation boundaries at a suitable distance that did not influence the computation of the electric field at location A. The metric used to get an estimate of the PML error at location A relative to the reference solution is given by the formula

$$\text{Error}_{\text{db}}|^n_{i,j} = 20\log_{10}\frac{\|E|^n_{i,j} - E_{\text{ref}}|^n_{i,j}\|}{\|E_{\text{ref}_{\text{max}}}|_{i,j}\|} \tag{52}$$

Errors from terminating the FDTD grid using a 10-cell CFS-PML, a 2nd order RIPML, and a 2-pole MPML, are illustrated in Fig. (2). In the following the value of σ_{opt} is given by [6]

$$\sigma_{\text{opt}} = \frac{m+1}{150\pi\Delta l} \tag{53}$$

where m is the order of the polynomial scaling used. In the case of the CFS-PML the parameters for s_{cfs} where [16]

$$\kappa = 1 + \kappa_{\text{max}}\Big(\frac{x}{d}\Big)^m$$
$$\sigma = 1.1\sigma_{\text{opt}}\Big(\frac{x}{d}\Big)^m, \;\; \alpha = 0.05$$

where $\kappa_{\text{max}} = 11$ and $m = 4$. d is the thickness of the PML and x is distance from the inner PML interface ($x \in [0, d]$) For the 2nd order RIPML the parameters were set to

$$\kappa_1 = 1$$
$$\sigma_1 = \sigma_{1_{\text{opt}}}\Big(\frac{x}{d}\Big)^6, \;\; \alpha_1 = 0$$
$$\kappa_2 = 1 + \kappa_{2_{\text{opt}}}\Big(\frac{x}{d}\Big)^3$$
$$\sigma_2 = \sigma_{2_{\text{opt}}}\Big(\frac{x}{d}\Big)^2, \;\; \alpha_2 = \alpha_0 + \sigma_1$$

where $\alpha_0 = 0.09$, $\kappa_{2_{\text{opt}}} = 7$, $\sigma_{1_{\text{opt}}} = 0.175/(150\pi\Delta l)$ adn $\sigma_{2_{\text{opt}}} = 2.5/(150\pi\Delta l)$ [16]

Optimum parameters of s_{mp} for the 2-pole MPML where found using trial and error to be

$$\kappa = 1 + \kappa_{\text{max}}\Big(\frac{x}{d}\Big)^{m_1}$$
$$\sigma_1 = 0.65\sigma_{\text{opt}}\Big(\frac{x}{d}\Big)^{m_2}, \;\; \alpha_1 = 0.11$$
$$\sigma_2 = 0.5\sigma_{\text{opt}}\Big(\frac{x}{d}\Big)^{m_3}, \;\; \alpha_2 = 0.05\Big(1 - \frac{x}{d}\Big)$$

where $\kappa_{max} = 7$, $m_1 = 4$ and for the first pole $m_2 = 2$ and α_1 is constant. For the second pole $m_3 = 8$ and α_2 is inversely linearly scaled from the maximum value at the inner PML interface to zero at the outer PML perfect electric conductor boundary. An interesting observation here is that using $m_3 = 8$, in this case, offers almost a negligible σ_2 in the first few PML layers as this high order scaling results in offering most of the added σ_2 at deeper PML layers. It is important to clarify that in a multipole PML the constituent CFS-PML poles do not act independently and it is the combined attenuation profile that they produce that is important. Having the extra degrees of freedom in defining the overall PML attenuation profile is what helps to improve the overall performance of the multipole PML. The performance of the 2-pole MPML is excellent when compared to the performance of the 2nd order PML requiring similar computational resources. There is a clear improvement in performance with regard to the 10-cell CFS-PML however, the computational effort is not comparable in this case.

In Fig. (3) the errors obtained from a 12-cell and a 14-cell CFS-PMLs and a 2-pole MPML are compared. These two CFS-PMLs have maximum PML sizes that result in being more efficient in terms of computational load and memory storage than the 2-pole MPML as obtained from Table I. It is clear that the 2-pole MPML is still outperforming the CFS-PMLs. In Table II the maximum errors as obtained by (52) are listed. Considering the given maximum errors and the late time reduction of error by the 2-pole MPML as well as the computational efficiency in terms of run time the suitability of a 2-pole MPML as a viable approach for terminating FDTD models is clearly evident.

TABLE II
MAXIMUM ERROR IN 2D CFS-PMLS AND A 2-POLE MPML

PML type	Maximum Error (db)
10-cell CFS-PML	-69.81
11-cell CFS-PML	-72.68
12-cell CFS-PML	-77.07
13-cell CFS-PML	-81.89
14-cell CFS-PML	-89.72
10-cell 2nd order RIPML	-83.30
10-cell 2-pole MPML	-89.42

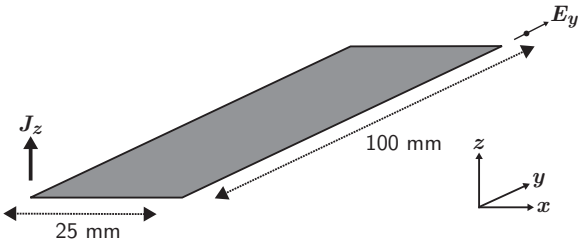

Fig. 4. A z-directed (J_z) electric current dipole source placed 1 mm above the corner of a 25×100 mm thin plate. The E_y field component is sampled 1 mm away from the plate's opposite corner [2].

B. Hertzian dipole response from a thin PEC plate

A model, presented in Fig. (4), of an elongated thin PEC plate (25×100 mm) is used to test the performance of the multipole PML formulation in a 3D FDTD code [13], [2]. The y-directed electric field response (E_y) one cell away from the thin PEC plate, due to a z-directed Hertzian dipole source placed diagonally opposite the field monitoring point and at 1mm above one of the PEC sheet corners, has been obtained. The FDTD grid used a uniform spatial-step $\Delta l = 1$ mm and a time-step $\Delta t = 1.906$ ps. The time variation of the source is the same as used in the previous 2D example and is given by (51). The FDTD grid was comprised of $51 \times 126 \times 26$ cells and the 2-pole MPML was set to have a 10-cell thickness. In the testing model only three cells of free space separated the PEC sheet target from the inner surface of the PMLs. A reference solution was obtained using a substantially larger FDTD model [19].

The time dependent error calculated using (52) is presented for a 10-cell CFS-PML, a 10-cell 2nd order RIPML and a 10-cell 2-pole MPML in Fig. (5). The parameters for the CFS-PML s_{cfs} where [16]

$$\kappa = 1 + \kappa_{\text{max}}\Big(\frac{x}{d}\Big)^{m}$$
$$\sigma = 1.1\sigma_{\text{opt}}\Big(\frac{x}{d}\Big)^{m}, \quad \alpha = 0.05$$

where $\kappa_{\text{max}} = 7$ and $m = 4$. For the 2-pole MPML optimum

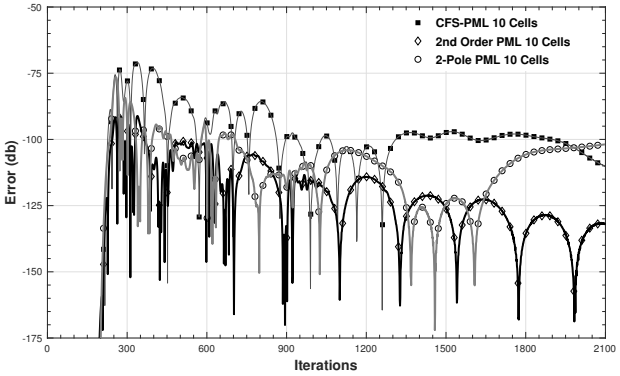

Fig. 5. Relative error in the response of the y-directed electric field component obtained one cell away from a PEC thin plate. A 10-cell thick PML has been used and was placed three cells away from the edge of the thin PEC sheet. Errors are presented for a CFS-PML, a 2nd order RIPML and a 2-pole MPML.

parameters for s_{mp} where found using a trial and error process

$$\kappa = 1 + \kappa_{\text{max}}\Big(\frac{x}{d}\Big)^{m_1}$$
$$\sigma_1 = 0.65\sigma_{\text{opt}}\Big(\frac{x}{d}\Big)^{m_2}, \quad \alpha_1 = 0.15\Big(1 - \frac{x}{d}\Big)^{m_3}$$
$$\sigma_2 = 0.65\sigma_{\text{opt}}\Big(\frac{x}{d}\Big)^{m_4}, \quad \alpha_2 = 0.8\Big(1 - \frac{x}{d}\Big)$$

where $\kappa_{\text{max}} = 11$, $m_1 = 4$ and for the first pole $m_2 = 4$ and $m_3 = 2$ as α_1 is inversely quadratically scaled from the maximum value at the inner PML interface to zero at the outer PML perfect electric conductor boundary. For the second pole $m_4 = 2$ and α_2 is inversely linearly scaled as in the 2D case. Finally, the parameters of the 2nd order RIPML stretching function s_{ho} were set as defined for the previous 2D example but with $\sigma_{1_{\text{opt}}} = 0.275/(150\pi\Delta l)$, $\sigma_{2_{\text{opt}}} = 2.75/(150\pi\Delta l)$ and $\alpha_0 = 0.07$ instead [16].

It is evident from Fig. (5) that the 2-pole MPML improves the overall performance of the boundary condition for the 3D case in a similar way as for the 2D case. An optimised 2-pole MPML having $\alpha_1 = 0$ had also been found to perform well in tests but it was clear that a 2-pole using two CFS poles MPML having different inverse polynomials for scaling of the α parameters performed better.

In Fig. (6) a comparison of the time dependent PML errors from an 11-cell and a 13-cell CFS-PMLs and the 10-cell 2-pole MPML is presented. The 11-cell CFS-PMLs corresponds to the maximum PML size that is more computational efficient in terms of run-time from a 10-cell 2-pole MPML. Equivalently, the 13-cell CFS-PML is the one that is more efficient in terms of memory storage than the 10-cell 2-pole MPML. The 10-cell 2-pole MPML is clearly better performing and as can be seen from Table III has a very small maximum error in addition to an improved late time performance.

V. CONCLUSION

A new PML formulation based on the stretched coordinate approach, that is agnostic of the underlying media properties, has been presented. The formulation, developed using recursive integration, utilises a new stretching function that

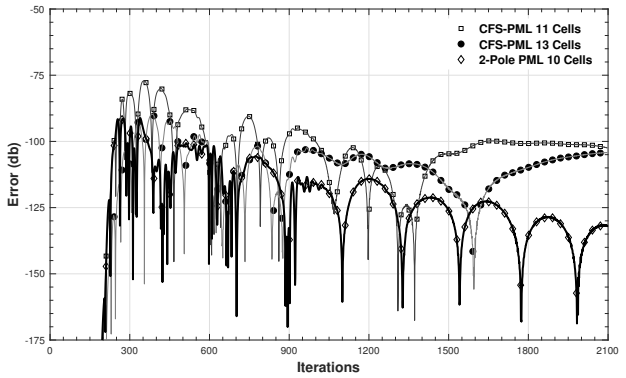

Fig. 6. Relative error in the response of the y-directed electric field component obtained one cell away from a PEC thin plate. Errors are presented for a 11-cell CFS-PML a thicker 13-cell CFS-PML and from a 10-cell 2-pole MPML.

TABLE III
MAXIMUM ERROR IN 3D CFS-PMLS AND A 2-POLE PML

PML type	Maximum Error (db)
10-cell CFS-PML	-71.00
11-cell CFS-PML	-77.14
12-cell CFS-PML	-83.65
13-cell CFS-PML	-89.75
14-cell CFS-PML	-95.50
10-cell 2nd order RIPML	-76.51
10-cell 2-pole MPML	-91.12

is build as a summation of elementary CFS-PML stretching functions thus introducing into the PML a generic multipole stretching. Application into an FDTD code is straightforward especially if the PML is considered as applying field dependent sources to relevant electric and magnetic fields. This approach allows the PML to be applied as a correction after normally updating the FDTD field components and simplifies coding greatly [19]. One important aspect is that the optimisation of the new multipole PML follows more naturally the design of the CFS-PML stretching function as the α parameters are found to require inverse scaling when compared to the PML conductivity terms. This is in agreement with the underlying design idea of the CFS-PML. Further, the fact that a multi-pole stretching function does not generate extra terms in its imaginary and real parts, contrary to the stretching function of higher order PMLs, is an advantage as it makes optimisation easier and does not require special arrangements of the PML parameters to guaranteed stability as is in the case for example of the 2nd order PML. In terms of stability very long runs, in excess of 10^6 iterations, performed during testing did not reveal any issues. The empirically optimised 2-pole MPML used in the examples presented here has been found to perform very well when compared to standard CFS-PMLs that in essence are equivalent to 1-pole MPMLs, requiring similar computational resources both in terms of execution time and of memory storage. As computer memory is becoming less of a limiting factor in FDTD modelling it is computational time that should be the main criterion in picking a suitable formulation for a PML. In this case the 2-pole MPML appears to have a clear and distinct performance advantage from an equivalently thick CFS-PML or a similar 2nd order PML. It is, however, important to stress as well the fact that significant further research effort is required in order to arrive at efficient optimisation guidelines in order for the MPML to be useful for practical everyday FDTD modelling problems without having to iteratively optimise its parameters which is time consuming and requires relevant PML expertise by the user. For example, an approach similar to [17], recently reported for optimising 2nd order PMLs, could be a potential starting point for such research effort.

REFERENCES

[1] K. Yee, "Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media," *Antennas and Propagation, IEEE Transactions on*, vol. 14, no. 3, pp. 302–307, May 1966.

[2] A. Taflove and S. C. Hagness, *Computational electrodynamics : the finite-difference time-domain method*, 3rd ed. Boston: Artech House, 2005.

[3] J.-P. Bérenger, "A perfectly matched layer for the absorption of electromagnetic-waves," *Journal of Computational Physics*, vol. 114, no. 2, pp. 185–200, 1994.

[4] W. C. Chew and W. H. Weedon, "A 3D perfectly matched medium from modified maxwell's equations with stretched coordinates," *IEEE Microwave and Guided Wave Letters*, vol. 7, no. 13, pp. 599–604, 1994.

[5] C. Rappaport, "Perfectly matched absorbing boundary conditions based on anisotropic lossy mapping of space," *Microwave and Guided Wave Letters, IEEE*, vol. 5, no. 3, pp. 90–92, 1995.

[6] S. Gedney, "An anisotropic perfectly matched layer-absorbing medium for the truncation of FDTD lattices," *IEEE Transactions On Antennas and Propagation*, vol. 44, no. 12, pp. 1630–1639, 1996.

[7] J. D. Moerloose and M. A. Stuchly, "Behavior of berenger's abc for evanescent waves," *IEEE Microwave and Guided Wave Letters*, vol. 5, no. 10, pp. 344–346, Oct 1995.

[8] J.-P. Bérenger, "Application of the CFS PML to the absorption of evanescent waves in waveguides," *Microwave and Wireless Components Letters, IEEE*, vol. 12, no. 6, pp. 218–220, Jun. 2002.

[9] ——, "Evanescent waves in PML's: Origin of the numerical reflection in wave-structure interaction problems," *IEEE Transactions On Antennas and Propagation*, vol. 47, no. 10, pp. 1497–1503, 1999.

[10] M. Kuzuoglu and R. Mittra, "Frequency dependence of the constitutive parameters of causal perfectly matched anisotropic absorbers," *Microwave and Guided Wave Letters, IEEE*, vol. 6, no. 12, pp. 447–449, Dec. 1996.

[11] J. A. Roden and S. Gedney, "An efficient FDTD implementation of the PML with CFS in general media," in *IEEE Antennas and Propagation Society International Symposium. Transmitting Waves of Progress to the Next Millennium*. IEEE, 2000, pp. 1362–1365.

[12] J.-P. Bérenger, *Perfectly Matched Layer (PML) for Computational Electromagnetics*, ser. Synthesis Lectures on Computational Electromagnetics, J.-P. Bérenger, Ed. Morgan & Claypool, 2007.

[13] D. Correia and J.-M. Jin, "On the development of a higher-order PML," *IEEE Transactions On Antennas and Propagation*, vol. 53, no. 12, pp. 4157–4163, 2005.

[14] S. Gedney and B. Zhao, "An Auxiliary Differential Equation Formulation for the Complex-Frequency Shifted PML," *Antennas and Propagation, IEEE Transactions on*, vol. 58, no. 3, pp. 838–847, 2010.

[15] N. Feng and J. Li, "Novel and efficient FDTD implementation of higher-order perfectly matched layer based on ADE method," *Journal of Computational Physics*, vol. 232, no. 1, pp. 318–326, Jan. 2013.

[16] A. Giannopoulos, "Unsplit Implementation of Higher Order PMLs," *Antennas and Propagation, IEEE Transactions on*, vol. 60, no. 3, pp. 1479–1485, 2012.

[17] H. Feng, W. Zhang, J. Zhang, and X. Chen, "Importance of double-pole CFS-PML for broad-band seismic wave simulation and optimal parameters selection," *Geophysical Journal International*, vol. 209, no. 2, pp. 1148–1167, May 2017.

[18] I. Giannakis and A. Giannopoulos, "A Novel Piecewise Linear Recursive Convolution Approach for Dispersive Media Using the Finite-Difference Time-Domain Method," *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 5, pp. 2669–2678, 2014.

[19] 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.
[20] A. Giannopoulos, "An Improved New Implementation of Complex Frequency Shifted PML for the FDTD Method," *Antennas and Propagation, IEEE Transactions on*, vol. 56, no. 9, pp. 2995–3000, Sep. 2008.

Antonios Giannopoulos received a B.Sc. (1991) in Geology from the Aristotle University of Thessaloniki, Greece and a D.Phil. (1997) in Electronics from The University of York, UK. His research interests include computational electromagnetics and in particular the application of the FDTD method on the numerical modelling of ground penetrating radar. In addition, he works on the development and application of geophysical techniques for infrastructure sensing applications. He created gprMax, a freely available FDTD simulator, and he is directing its continuous development and enhancement. He was the General Chair of the 9th International Workshop on Advanced Ground Penetrating Radar, Edinburgh, 2017. He is a member of SEG and EAGE.