On the solution of 3-D frequency dependent Crank–Nicolson FDTD scheme

Hasan Khaled Rouf † ‡, Fumie Costen†, Salvador G. Garcia †‡
†School of Electrical and Electronic Engineering, The University of Manchester, M60 1QD, U.K., hasan.rouf@postgrad.manchester.ac.uk, f.costen@cs.man.ac.uk
‡Faculty of Sciences, University of Granada, Granada, Spain, salva@ugr.es

Abstract—Sparse matrix solvers are an efficient alternative to direct solvers for the Crank–Nicolson Finite Difference Time Domain (CN–FDTD) method. The unconditional stability of CN–FDTD permits us to use time steps over the Courant–Friedrich–Lewy (CFL) limit of the classical FDTD method. However, when the time step is far above the CFL limit, the coefficient matrix is no longer diagonally dominant, and both direct and iterative solvers increase their solution time. In this paper, we compare two widely employed iterative solvers for sparse systems: Bi-Conjugate Gradient Stabilised (BiCGStab) and Generalised Minimum Residual (GMRES). A frequency dependent CN–FDTD (FD–CN–FDTD) for Debye media is formulated and employed for numerical testing. Results from our simulations manifest that BiCGStab outperforms GMRES in terms of computational efficiency.

Index Terms—Crank–Nicolson method, FDTD, Debye media, sparse matrix solver, iterative methods

I. INTRODUCTION

The Finite Difference Time Domain (FDTD) [1] method is one of the most widely used numerical techniques used in computational electrodynamics. FDTD is straightforward, robust, versatile and widely applicable for wideband systems. However, for many problems of interest it may become computationally inefficient, due to the upper limit for the time step imposed by the Courant–Friedrich–Lewy (CFL) stability condition [2]. This typically happens when a high spatial resolution is required to model fine geometries, leading to spatial sampling much shorter than the minimum wavelength of interest, enforcing an unnecessarily large FDTD time step for stability. There is a growing interest in overcoming this limitation by employing unconditionally stable implicit FDTD methods, for which time and space steps can be independently chosen. This trend will continue, because high accuracy in modeling is increasingly demanded with the massive improvement of computational resources. An alternative to the explicit FDTD is the Crank–Nicolson [3] FDTD (CN–FDTD) method, for being unconditionally stable beyond the CFL limit. As in the classical FDTD, CN–FDTD replaces the time and space derivatives by second order centered differences, but unlike FDTD, the fields affected by the curl operators are also averaged in time. The resulting scheme is an unconditionally stable fully implicit marching–on–in–time algorithm with high accuracy and low anisotropy [4].

There have been works attempting to simplify or approximate CN–FDTD such as, the alternating direction implicit (ADI–FDTD) method [5], the CN Douglas–Gunn method [6], CN cycle–sweep method [7], or CN approximate factorization splitting method [8]. These approximations suffer up to some extent of numerical errors, which may become severe for some practical applications [9], [10], [11]. Recently, both iterative preconditioned/non–preconditioned [12], [13] and direct solving of CN–FDTD [14] have been developed.

The selection of the matrix solver to handle the implicit system of equations arising in CN–FDTD, is the critical part to fully harness the advantages of using larger time–step, since the computational costs associated with the solver must be kept as low as possible. Either direct or iterative solvers can be employed. Direct solvers are not the preferred choice for large problems of practical importance because of their enormous memory requirements [14]. Iterative solvers based on Krylov subspace theory have proven to be suitable for practical problems [12], [13]. However computational efficiency of an iterative solver is application–dependent and therefore to determine the effectiveness of each method is essential.

In this paper, we present a new method to solve the 3D CN–FDTD method for frequency–dependent media (FD–CN–FDTD), based on Krylov–subspace solvers for sparse systems. Single–pole Debye media have been introduced into CN–FDTD by means of an auxiliary differential formulation [15]. The scheme results in a sparse system of linear equations involving the three components of the electric field, from which all the remaining field quantities are explicitly found. The remaining of this paper is organized as follows: Section (ii) introduces FD–CN–FDTD mathematically. The techniques to solve the system and numerical results are presented in section (iii). Section (iv) presents the conclusion and discuss on open issues for future work.

II. THEORY OF FD–CN–FDTD

Maxwell’s curl equations in material independent form are

\[
\nabla \times E = -\frac{\partial B}{\partial t}
\]

\[
\nabla \times H = \frac{\partial D}{\partial t}
\]
where $E$, $H$, $D$ and $B$ are electric field, magnetic field, electric flux density and magnetic flux density, respectively. The constitutive relationships for isotropic, linear, non–magnetic, single–pole Debye electrically–dispersive media are in frequency domain
\[ B = \mu_0 H \]  
\[ D = \varepsilon_0 (\varepsilon_\infty + \frac{\varepsilon_S - \varepsilon_\infty}{1 + j\omega\tau_D} - j\frac{\sigma}{\omega\varepsilon_0}) E \]

where $\varepsilon_0$ and $\mu_0$ are the free–space permittivity and permeability, $\varepsilon_\infty$ is the static permittivity, $\varepsilon_S$ is the optical permittivity, $\tau_D$ is the relaxation time and $\sigma$ is the static conductivity. Eq. (4) can be re-written as
\[(j\omega)^2 \tau_D D + j\omega D = (j\omega)^2 \varepsilon_0 \varepsilon_\infty \tau_D E + j\omega (\varepsilon_0 \varepsilon_S + \sigma \tau_D) E + \sigma E\]

By mapping $(j\omega)^m$ in frequency domain, into $\frac{\partial^n}{\partial t^n}$, in time domain, Eq. (5) can be written as
\[ \tau_D \frac{\partial^2 D}{\partial t^2} + \frac{\partial D}{\partial t} = \varepsilon_0 \varepsilon_\infty \tau_D \frac{\partial^2 E}{\partial t^2} + (\varepsilon_0 \varepsilon_S + \sigma \tau_D) \frac{\partial E}{\partial t} + \sigma E \]

The application of Crank–Nicolson method to Eqs. (1)–(5), after algebraic manipulation, yields an equation only with electric field $E^{n+1}_{i,j,k}$ terms:
\[ E^{n+1}_{x} = \frac{\xi_1}{\xi_4} \left( \frac{\Delta t}{2} \right)^2 \frac{1}{\mu} \frac{\partial^2 E^{n+1}_{y}}{\partial y^2} + \frac{\xi_1}{\xi_4} \left( \frac{\Delta t}{2} \right)^2 \frac{1}{\mu} \frac{\partial^2 E^{n+1}_{z}}{\partial z^2} \]
\[ + \frac{\xi_1}{\xi_4} \left( \frac{\Delta t}{2} \right)^2 \frac{1}{\mu} \frac{\partial^2 E^{n+1}_{x}}{\partial x^2} - \frac{\xi_1}{\xi_4} \left( \frac{\Delta t}{2} \right)^2 \frac{1}{\mu} \frac{\partial^2 E^{n+1}_{y}}{\partial x \partial y} \]
\[ - \frac{\xi_1}{\xi_4} \left( \frac{\Delta t}{2} \right)^2 \frac{1}{\mu} \frac{\partial^2 E^{n+1}_{y}}{\partial x \partial z} + \frac{\xi_1}{\xi_4} \left( \frac{\Delta t}{2} \right)^2 \frac{1}{\mu} \frac{\partial^2 E^{n+1}_{z}}{\partial x \partial z} \]
\[ + \frac{\xi_1}{\xi_4} \left( \frac{\Delta t}{2} \right)^2 \frac{1}{\mu} \frac{\partial^2 E^{n+1}_{x}}{\partial z^2} \]
where $\xi_1$, $\xi_2$, $\xi_3$, $\xi_4$, $\xi_5$, $\xi_6$ are space dependent constants function of the Debye parameters and of $\Delta t$.

Permutation of $x$, $y$ and $z$ in Eq. (7) yield the remaining two E–field equations. By applying them to each Yee–grid position, a system of linear equations of $AN = C$ is set up. $A$ is the coefficient matrix, $N$ represents a vector with the electric field components to be solved, and $C$ is the excitation vector. The system of equations $AN = C$ is solved first to find the $E$ field. $D$ and $H$ fields are calculated in an explicit manner from the $E$ field. In order to terminate the computational space simple Mur first order boundary condition are employed.

III. SOLUTION OF FD–CN–FDTD

A. Sparse Matrix

The coefficient matrix $A$ generated by FD–CN–FDTD is extremely large and highly sparse. The size of $A$ is $9(N_x)^3(N_y)^3(N_z)^3$, where $N_x$, $N_y$, $N_z$ are the size of the FDTD space in $x$, $y$ and $z$ directions. When the problem space is homogeneous, the coefficient matrix is symmetric and otherwise asymmetric. Fig. 1 shows the sparsity pattern of $A$ in the case where the entire computational space is filled with Debye parameters $\varepsilon_S = 6.2$, $\varepsilon_\infty = 3.5$, $\sigma = 0.029$ S/m and $\tau_D = 39.0$ ps. It bears noting that the sparsity pattern of Fig. 1 is similar to that of three dimensional Finite Difference Frequency Domain (FDFD) method [16], and therefore the discussions in this paper could also be useful to the FDFD researchers.

In this paper, two cases are considered for study. The first one consists of an inhomogeneous medium, in a cubic space of size $80^3$ cells, with 5 different media as shown in Fig. 2. The second one involves the same cubic space of the previous case, now filled with a homogeneous medium with Debye parameters $\varepsilon_S = 6.2$, $\varepsilon_\infty = 3.5$, $\sigma = 0.029$ S/m and $\tau_D = 39.0$ ps. In both cases a $z$–directed dipole hard source of modulated Gaussian pulse centered at 3 GHz was placed at the center of the computational space. Spatial sampling was uniform: $\Delta x = \Delta y = \Delta z = \Delta = 10^{-3}$m. The time step is taken equal or above the CFL stability condition of the explicit FDTD: $\Delta t = CFLN(\Delta/(c\sqrt{3})$ with CFLN henceforth referred to as CFL number, and $c$ the free–space light–speed. For the first case observed signals at (65, 40, 40) are shown in FIGURE when $CFLN = 1, 3, 5$ for FD–CN–FDTD alongwith the observation from explicit frequency-dependent FDTD.

B. Condition number and diagonal dominance

The ease of solution on a linear system of equations can be measured by the condition number of $A$. [17] reports that the condition number of the coefficient matrix $A$ for frequency–independent Crank–Nicolson scheme increases with the CFL.
C. Iterative Solvers

To solve FD–CN–FDTD either direct or iterative solvers can be used. Although direct solvers are robust and reliable [14], they are computationally more expensive than iterative solvers, unless parallelized, and require excessively large memory. For example, a 30$^3$ cells computational space solved by the direct solver using sparse Gaussian elimination requires 2.4 GB of memory whereas iterative solvers like BiCGStab (Bi-Conjugate Gradient Stabilised) and GMRES (Generalised Minimum Residual) require only 62 MB and 65 MB, respectively. Thus for practical problems iterative solvers should be used [18].

Depending on the homogeneity of the computational space FD–CN–FDTD yields symmetric and asymmetric matrices. Since for homogeneous problems the coefficient matrix is symmetric, iterative solvers like Conjugate Gradient (CG) method can be used. However practical problems are inhomogeneous, and we have to resort to iterative solvers for asymmetric matrices. Usual ones are GMRES, BiCGStab, or Conjugate Gradient Squared (CGS) (Appendix D of [19]). On the other hand CGS suffers from irregular convergence, which may lead to substantial build-up of rounding errors because this algorithm is based on squaring the residual polynomial [18] [19]. Therefore in the present study, we will focus only in GMRES and BiCGStab. An advantage of BiCGStab method is that its computational cost per iteration is similar to that of CGS, but it avoids the irregular convergence patterns of CGS while maintaining about the same rate of convergence. GMRES is said to be a very robust solver for nonsymmetric matrices. It leads to the smallest residual for a fixed number of iteration steps. But these steps become increasingly expensive and in order to limit the increasing storage requirements and work per iteration step, restarting is necessary. In this work GMRES(m) that restarts every m iterations [20] and BiCGStab of [21] were used.

D. Performance study of BiCGStab and GMRES

In this section we compare the performance of BiCGStab and GMRES both for the homogeneous and inhomogeneous media cases described earlier. Fig. [4] shows the convergence pattern for CFLN = 20, plotting the residual error as a function of the average number of iterations required to achieve a specified accuracy. For example, to make the residual error lower than $10^{-8}$ BiCGStab requires about 45 iterations whereas GMRES requires about 97 iterations in both cases. The convergence rate of the solvers is weakly affected by homogeneity. For a modest value of CFLN number of iterations would certainly be lowered than those showed in Figure[4]. Our finding of BiCGStab outperforming GMRES in computational efficiency is contrary to the findings of [17] which reports...
GMRES is the fastest for their frequency-independent CN–FDTD scheme.

Figure 5 shows how the average number of iterations, required by BiCGStab and GMRES to converge, increases with CFL numbers. Stopping criteria in this case was $10^{-13}$ and the reason for selecting this small value of convergence tolerance is, in FD–CN–FDTD $D = \epsilon E$ is used and $D$ can have a value of such small order because of $\epsilon$ (permittivity). GMRES stagnates when convergence tolerance is below $10^{-13}$ while BiCGStab can work even at a lower convergence tolerance. Both solvers require more iterations to converge as CFLN go up, but the rate of increase of iteration numbers with CFLN is higher for GMRES than for BiCGStab. Homogeneity does not affect significantly this rate, particularly, for BiCGStab. The change of iteration number with CFLN for convergence tolerance values from $10^{-12}$ to $10^{-3}$ can be assumed from Figure 4. In FD–CN–FDTD total number of iterations required to complete the simulation is decreased with CFLN but the increase of computational costs per iteration with CFLN as shown in Fig. 5 can undermine this positive effect unless the solution is very efficient.

Figure 6 plots the CPU time required by BiCGStab and GMRES as a function of CFLN. The case of stopping criterion of $10^{-13}$ was employed and simulated to reach a fixed time instant by letting the code run for $1200/\text{CFLN}$ time steps on a dual AMD Opteron 280 with 8GB of memory. Observe that the CPU–time decreases with CFLN, for both solvers, although GMRES requires more CPU time than BiCGStab.

Table I presents the memory required by the two solvers for three different computational space sizes. GMRES always requires more memory than BiCGStab.

From all the above, we can conclude that BiCGStab outperforms GMRES in computational efficiency. This finding is in contrary / opposite to that of the frequency-independent CN–FDTD scheme of [17]. The work of [17] is based on [1] and [2] while FD–CN–FDTD additionally involves [6] which has 2nd order time derivative terms. The FD–CN–FDTD involves nine field components in place of six for CN–FDTD and the sparsity pattern of the former has more bands / stripes than the latter [13]. Apart from this the concerned problem of simulation, implementation optimization, or guru–parameters (COMMENT: DIDNOT UNDERSTAND THIS PREFIX 'GURU-') tuning may be attributed to conclude which solver is the most efficient. (COMMENT: OUR COLLABORATOR IN JAPAN ALSO FOUND BICGSTAB BETTER THAN GMRES FOR FD-CN-FDTD, IN FACT BEST)

### IV. Conclusion

In this paper, we have presented a new Krylov–based approach to solve the three dimensional CN-FDTD method for frequency–dependent Debye dispersive media. Two best–known iterative methods, GMRES and BiCGStab, were compared in terms of the number of iteration requirements for

<table>
<thead>
<tr>
<th>Computational Size (cells)</th>
<th>$40^3$</th>
<th>$60^3$</th>
<th>$80^3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>BiCGStab</td>
<td>145Mb</td>
<td>487Mb</td>
<td>1.1Gb</td>
</tr>
<tr>
<td>GMRES</td>
<td>151Mb</td>
<td>507Mb</td>
<td>1.2Gb</td>
</tr>
</tbody>
</table>

**TABLE I**

<table>
<thead>
<tr>
<th></th>
<th>$40^3$</th>
<th>$60^3$</th>
<th>$80^3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>BiCGStab</td>
<td>145Mb</td>
<td>487Mb</td>
<td>1.1Gb</td>
</tr>
<tr>
<td>GMRES</td>
<td>151Mb</td>
<td>507Mb</td>
<td>1.2Gb</td>
</tr>
</tbody>
</table>
convergence with different CFLN, and in terms of CPU–time and memory requirements. BiCGStab outperforms GMRES when used with FD–CN–FDTD in every aspect of the study. It has also been pointed out that the diagonal–dominance of the coefficient matrix which degrades when CFLN increases, is a main reason for the increase of the CPU time needed by the solvers. We have implemented two preconditioners: incomplete LU with no fill-in ILU(0) and Sparse Approximate Inverse (SAI). ILU(0) does not give any improvement in convergence. However SAI gave slight improvement in reducing the number of iterations to converge but there were two major setbacks making it unsuitable for use. The time to compute the approximate inverse preconditioner is too long making the total CPU time longer than that without any preconditioner. [13] showed SAI can reduce iteration but did not mention the total CPU time. Secondly, the memory requirements of SAI restricted the maximum computational space to only $30^3$ cells. SAI also showed lack of robustness in our case. Further work is needed to tailor suitable preconditioners to improve the iterative solver convergence.

REFERENCES


