A Look at the Boundary Conditions of the Forgy-Chew FDTD Algorithm and its Implications for use in PIC Codes

Ernest A. Baca
Andrew Greenwood
Keith Cartwright
Air Force Research Laboratory

In 1998 E. Forgy and W. C. Chew introduced a new "overlapped lattice" Finite-Difference Time-Domain (FDTD) method comprised of the superposition of the Yee FDTD lattice with the Discrete Space-Time (DST) lattice in a combined algorithm. While the combined algorithm involves increased computational overhead, it also provides increased stability allowing larger time increments and provides finer spatial sampling. But most importantly, the combined algorithm leads to numerical dispersion properties so low as to make a compelling case for implementation into the existing Particle-In-Cell (PIC) code called ICEPIC for use in electrically large PIC simulations or PIC simulations sensitive to phase errors. The low dispersion property originates from the complementary dispersion properties of both the staggered, non-collocated FDTD Yee algorithm and the staggered collocated Discrete Space-Time (DST) algorithm.

However, even before any advantages or disadvantages for PIC can be considered, a thorough examination of the EM behavior at material discontinuities must be undertaken to determine whether more is lost due to inaccurate reflection and transmission characteristics inherent in the method than is gained due to the very appealing numerical dispersion properties. While Forgy and Chew claim that the local stencil satisfies conditions on interfaces as in the Yee stencil, a question arises about how to deal with the charge distribution that would accompany a normal electric field component at the edge of a cell of perfectly conductive material. This presentation will look at the mathematical accuracy of the Forgy-Chew method at conductive and non-conductive boundaries (aligned and not aligned to the grid) and explore methods to overcome any deficiencies discovered. It will also discuss the advantages gained by the Forgy-Chew method within the PIC framework.