

# On Comparisons of Time-Domain Scattering Schemes

Brian T. Nguyen  
Institute for Mathematics and its Applications  
University of Minnesota  
Minneapolis, MN 55455

January 29, 1998

## 1 Introduction

There are three ways to change the accuracy of time-domain electromagnetic and acoustic scattering simulations: change the grid resolution

$$N = \frac{\lambda}{h} \quad (1)$$

where  $\lambda$  is the wavelength and  $h$  is the grid spacing, change the numerical algorithm or both. Changing the algorithm can yield greater accuracy *and*, accompanying that, greater computational expense, for the same  $N$ . For the users of the algorithms, it would be helpful to know how these changes balance out, even if that is not the sole consideration. Unfortunately, comparisons of scattering schemes made in the literature usually lack a useful standard by which data from different publications can be directly compared. The present ways to present data do not do enough to facilitate direct comparisons between different works, especially those from authors with different backgrounds.

The purpose of this paper is to offer a way to present comparative error data so that the true cost of the schemes are apparent. First, we review some common ways to present data and illustrate why they are insufficient for cost comparison. Second, we offer a new way to present data which is easy to generate and reveal additional valuable information about the schemes being compared. As an example, we present a comparison of some new and some old schemes. By facilitating comparisons, we hope to generate more beneficial dialog between different disciplines such as electromagnetic scattering, acoustic scattering and numerical analysis.

## 2 Ways to Present Error Data

The “dispersion curve” of Figure 1 is often used to display the accuracy of a scheme. This is simply a

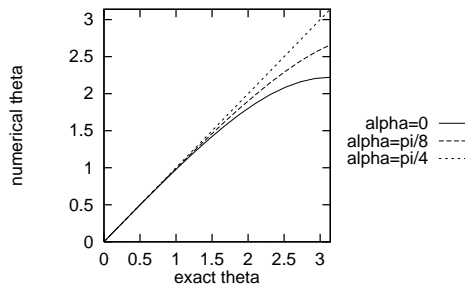


Figure 1: Dispersion curve.

plot of the numerical vs. exact values of  $\theta = \kappa h$ . This method is good for showing how the numerical value departs from the exact. Slow waves drop the numerical curve below the exact curve and fast waves raise the numerical curves above it. In multi-dimensions, several curves can be given, one for each chosen direction, but the plot is unable to truly accommodate variable propagation directions. This plot shows where the scheme is no longer accurate, but the region of good accuracy is hidden by the scale. The particular data shown in the figure is for the 2D Yee’s [6] scheme at the stability limit. It shows that this scheme is worst at  $\alpha = 0$  (along an axis) and exact at  $\alpha = \frac{\pi}{4}$  (along the quadrant bisector).

Figure 2 shows another popular plot, displaying the propagation speed as wave fronts in a 2D plane. The exact wave is represented by the unit circle, while slow waves form fronts inside and fast waves form fronts outside. Presumably, the optimal time step was used. In contrast to Figure 1, the error variation due to changing directions is brought out by this plot. The data shown is also for the 2D Yee’s scheme at the stability limit. The extremely poor accuracy of the coarsest resolution ( $N = 2$ ) is described well, but the region of useful accuracy is once again difficult to see due to the scale. The scale can be improved by recognizing that the errors are (and should be) symmetric

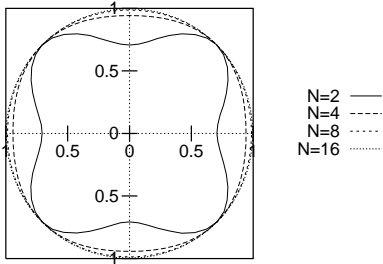


Figure 2: Wave fronts in a plane.

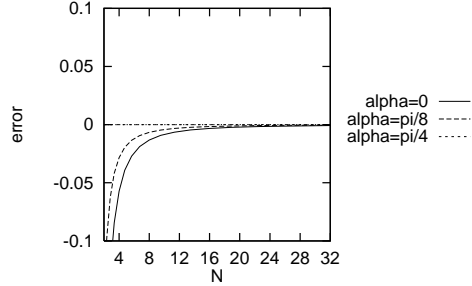


Figure 3:  $\epsilon_p$  vs.  $N$ .

with respect to a coordinate transformation, so that only the first half of the first quadrant ( $0 \leq \alpha \leq \frac{\pi}{4}$ ) needs to be displayed. Nevertheless, this plot still does not work well in 3D.

The dispersion curve and the wave front plots are useful in analyzing the numerical algorithm, because they visually display the numerical behavior of individual schemes. However they are less useful for users who are interested in the quantitative relationship between cost and a limited range of error values.

To concentrate on a more critical measure of accuracy, we turn to a plot that displays the error rather than the physical picture of the numerical behavior. Let us define the normalized phase speed error as

$$\epsilon_p = \frac{c_{\text{num}}}{c} - 1 \quad (2)$$

where  $c_{\text{num}}$  is the numerical propagation speed and  $c$  is the exact. The error depends on  $N$  and in addition, the time step size and the direction of wave propagation. Figure 3 shows some curves of the function  $\epsilon_p(N)$ . Presumably, the optimal time step was used. This relationship is directly helpful in deciding what resolution to use, but each curve represents only a single direction in space, so multi-dimensional variation is not visible. Also, the linear scale (chosen here) shows clearly only a limited range of error. In this plot, the log of  $\epsilon_p$  cannot be used because some schemes have curves that cross the  $\epsilon_p = 0$  axis.

One way to get a truly multi-dimensional measure of error is to parametrize it with a single representative value, such as the  $L_p$  norm of  $\epsilon_p$  over all possible propagation directions. Though any “p” may be chosen, we use the  $L_\infty$  norm, which picks out the worst error. This norm is easy to find, and little indication can be found (at the present time) that a different norm would be more appropriate. In addition to showing the resolution required for a certain  $\epsilon_p$ , cost can also be compared to  $\epsilon_p$ , which is difficult to do with the other ways of displaying accuracy. Below, we make all these comparisons for several numerical

schemes. These plots should be considered to be additional to the above plots but not to replace them. The information shown is valuable for assessing the cost-accuracy balances, but information on the specific behaviors of the schemes, such as the conditions for worst error, are not shown.

### 3 Comparison of Different Numerical Schemes

To demonstrate the information revealed by the new plots, we compare two types of schemes. The first type, upstream-biased leapfrog (UL) schemes, is described in reference [3, 4, 5]. It is unique in that upstream-differencing and neutral-stability was combined into one type of scheme. Let us refer to the second-order version by UL2 and the fourth-order version by UL4. The fourth-order version does not have a stable 3D counterpart in electromagnetics [3]. The standard scheme used for comparisons are the widely-used Yee’s scheme [6] and its fourth-order extensions as described by Deveze [1], Fang [2] and Nguyen [3, 4]. In reference [3], these schemes are shown to be the standard leapfrog (SL) schemes (except on a denser mesh), so we name the second-order scheme SL2 and the fourth-order scheme SL4. In addition, the sixth-order extension SL6, given in [3], is also considered.

Figures 4 and 5 plot various costs for 2D and 3D versions of the chosen schemes. The first of the four plots in each figure gives the resolution as a function of the required error limit on  $L_\infty \epsilon_p$ . This is not strictly a cost, but it is the easiest parameter to visualize. The second plot gives the actual number of cells required to discretize a unit wavelength domain ( $\lambda^2$  in 2D and  $\lambda^3$  in 3D). This parameter is not a cost either, but if other memory requirements are to be stored on the grid (material property, for example) or other computations must accompany the ud-

pate computations (extensions for dispersive media, for example), it is important to remember that these additional costs grow with the total number of cells required. (This may favor schemes that use a smaller resolution, even if the bare schemes are more expensive.) The third and fourth plots give the memory and floating-point operations (flops) required. These measures of computational expense consider only the bare scheme, not the additional memory or computational requirements just mentioned. The memory requirement is given in the number of floating-point numbers (FPN) required by the unit wavelength domain. It is computed by first determining that the SL schemes require 3/cell in 2D and 6/cell in 3D and that the UL schemes require 8/cell in 2D and 24/cell in 3D. The flop requirements are the number of floating-point operations needed to integrate one wave period in time. They are computed by first determining the number of flops to update one cell one time step and the number of time steps necessary to integrate one period ( $\frac{Nh}{c\Delta t}$ ). Floating-point operations are defined as any of the add, subtract, multiply and divide operators. None others, except assignments and flow controls were needed.

It is no surprise that the more complex schemes require smaller  $N$  for a given level of error and that they converge faster than the second-order schemes do. (The convergence rate is the inversely proportional to the slope of the log-log plots). Some information that can be quickly gained from these plots are (1) ranking by cost, (2) break-even points and (3) cost differences and ratios. The curve for the 2D UL4 scheme in levels out at the high-error end due to the restriction  $N \geq 2$  to avoid aliasing.

For the schemes considered in this article, the maximum stable time steps are used because they reduce the number of time steps required for a given computation, and they are also the values at which the lowest error is achieved. Schemes that achieve maximum accuracy at a time step lower than their maximum must be evaluated based on the cost of taking smaller time steps.

## 4 Conclusion

While it is important to consider cost vs. accuracy issues, other issues can be important (even decisive) in which scheme to use. A clear example is that a generalized scheme is inevitably more expensive than a specialized scheme, but some times, one must exclude the specialized scheme because it does not apply.

There are several other notes we would like to mention regarding the comparison of different schemes.

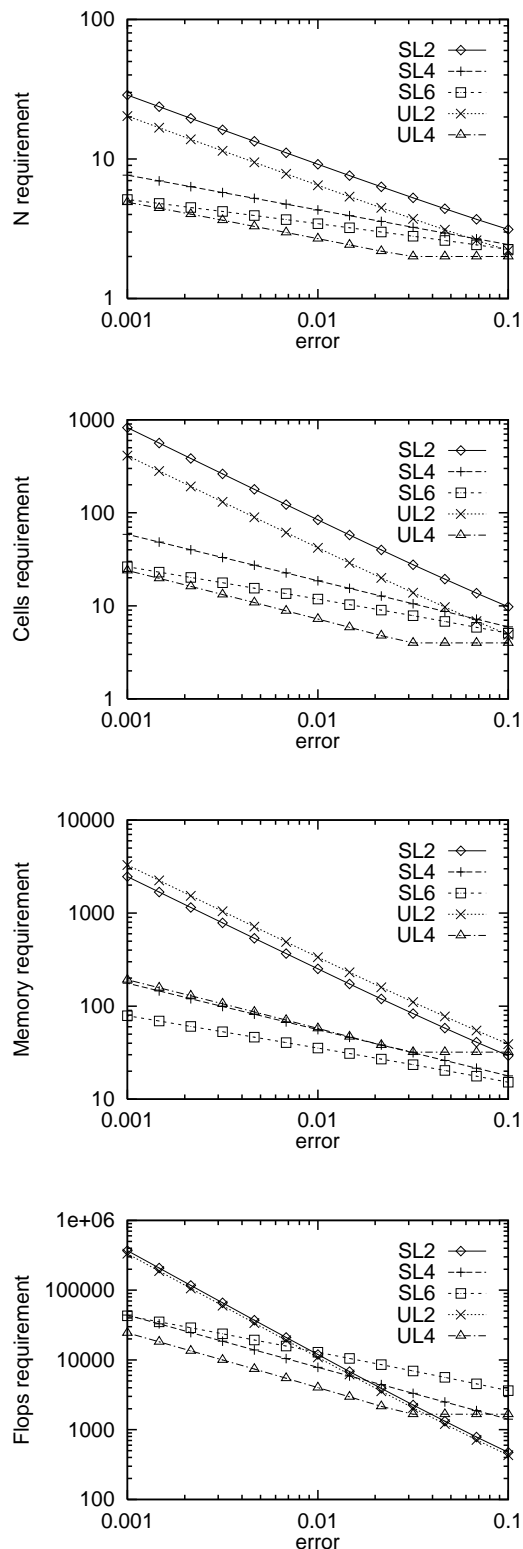


Figure 4: Computational resource requirements for 2D schemes.

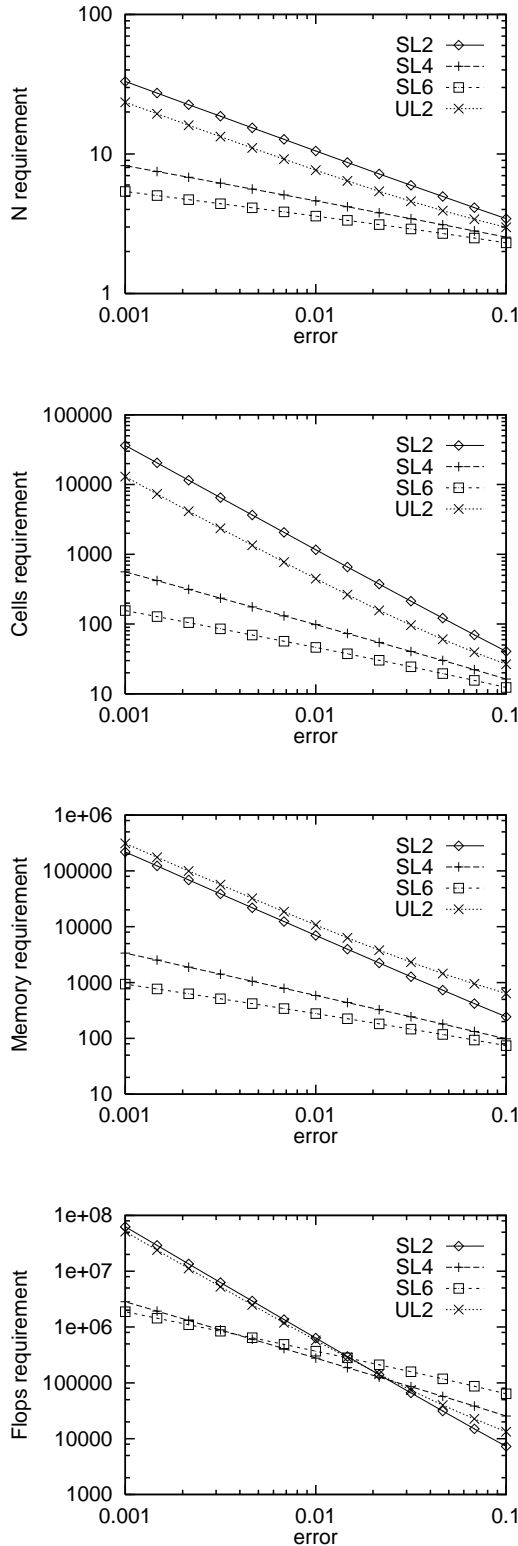


Figure 5: Computational resource requirements for 3D schemes.

Different schemes perform their best under different conditions. For example, Yee's scheme is best at the stability limit, but others may not be. Schemes should be examined for the conditions in which they would be used instead of conditions optimal for other schemes. Some schemes are developed using semi-discrete methods, leaving the time-stepping method flexible. For these schemes, it is important to note that the error of the semi-discrete methods may be improved or aggravated by the time-stepping scheme and that the total cost for these schemes cannot be determined until one chooses a time-stepping scheme for them. Without including the time-stepping scheme, semi-discrete schemes *cannot* be compared to schemes in which time marching is inseparable.

## References

- [1] Thierry Deveze. High order F.D.T.D. algorithm to reduce numerical dispersion and staircasing. In *10th Annual Review of Progress in Applied Computational Electromagnetics*, pages 61–68. Applied Computational Electromagnetics Society, March 1994.
- [2] J. Fang. *Time Domain Computation for Maxwell's Equations*. PhD thesis, University of California, Berkeley, California, November 1989.
- [3] Brian T. Nguyen. *Investigation of Three-Level Finite-Difference Time-Domain Methods for Multidimensional Acoustics and Electromagnetics*. PhD thesis, University of Michigan, Ann Arbor, MI, August 1996.
- [4] Brian T. Nguyen. High-order non-dissipative finite difference methods for time domain acoustics and electromagnetic equations. Technical Report IMA Preprint Series, IMA, 1997. In preparation.
- [5] Brian T. Nguyen and Philip L. Roe. Application of an upwind leapfrog method for electromagnetics. In *10th Annual Review of Progress in Applied Computational Electromagnetics*, pages 446–458. Applied Computational Electromagnetics Society, March 1994.
- [6] Kane S. Yee. Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. *IEEE Transactions on Antennas and Propagation*, AP-14(3):302–307, May 1966.