How to Understand Time-Space Synchronized FDTD Algorithm

Part 4

David Ge (dge893@gmail.com)

In part 2 I explained one method of handling computing domain boundary by estimating a function derivative with nearest function values. In part 3 I explained how equations (52), (53) and (54) in TSS.PDF are used to express this method. In this part, I'll explain how the pseudo code on page 9 implement this method.

In part 1, I explained that for a second order derivative estimation, that is, $2M = 2, or\ M = 1$, the estimation is given by

$$\begin{bmatrix} \Delta s \dfrac{dv(w\Delta s)}{ds} \\ (\Delta s)^2 \dfrac{d^2 v(w\Delta s)}{ds^2} \end{bmatrix} \approx \begin{bmatrix} 1 & \dfrac{1}{2} \\ -1 & \dfrac{1}{2} \end{bmatrix}^{-1} \begin{bmatrix} v(w\Delta s + \Delta s) - v(w\Delta s) \\ v(w\Delta s - \Delta s) - v(w\Delta s) \end{bmatrix}$$

Because we only estimate $\Delta s \frac{dv(w\Delta s)}{ds}$, we only need the first row of the inverse matrix. Please remember this fact. Also remember that it is a constant matrix of *2M x 2M*.

For $M \geq 1$, the matrix is given by

$$\begin{bmatrix} 1 & 1/2! & \dots & 1/k! & \dots & 1/(2M-1)! & 1/(2M)! \\ 2 & 2^2/2! & \dots & 2^k/k! & \dots & 2^{2M-1}/(2M-1)! & 2^{2M}/(2M)! \\ \vdots & \vdots & \cdots & \eta^k/k! & \dots & \vdots & \vdots \\ M & M^2/2! & \dots & M^k/k! & \dots & M^{2M-1}/(2M-1)! & M^{2M}/(2M)! \\ -1 & 1/2! & \dots & (-1)^k/k! & \dots & (-1)^{2M-1}/(2M-1)! & (-1)^{2M}/(2M)! \\ -2 & 2^2/2! & \dots & (-2)^k/k! & \dots & (-2)^{2M-1}/(2M-1)! & (-2)^{2M}/(2M)! \\ \vdots & \vdots & \cdots & (-\eta)^k/k! & \dots & \vdots & \vdots \\ -M & M^2/2! & \dots & (-M)^k/k! & \dots & (-M)^{2M-1}/(2M-1)! & (-M)^{2M}/(2M)! \end{bmatrix}$$

Again, we only need to get the first row of the inverse matrix of the above matrix. Also, it is a constant matrix unrelated to field data and time.

But, the above estimation only works for regions away from the computing domain boundary. Let's call this "case 0" situation. What are other situations? Let's investigate this issue.

First, let's identify the condition for "case 0":

$$case\ 0: |w| \leq r_{max} - M$$

Next, let's consider the cases when $r_{max} \geq w > r_{max} - M$; there are *M* cases:

$$case\ 1: w = r_{max}$$

$$case\ 2: w = r_{max} - 1$$

$$\dots$$

$$case\ M: w = r_{max} - (M - 1)$$

Now let's consider the cases when $-r_{max} \leq w < M - r_{max}$, there are *M* cases:

$$case\ M + 1: w = -r_{max}$$

$$case\ M + 2: w = -r_{max} + 1$$

$$\dots$$

$$case\ 2M: w = -r_{max} + (M - 1)$$

Let's summarize all the *2M+1* cases determined by *w*:

case		*h*	*P*		*N*		*w*	
0	0	*0*	*M*		*-M*		$\lvert w \rvert < r_{max} - M$	
h	*1*	$r_{max} - w + 1$	*0*	*h-1*	-2M	*-(2M-h+1)*	$w = r_{max}$	$r_{max} \geq w > r_{max} - M$
	2		*1*		-2M+1		$w = r_{max} - 1$	
	...		...		...		...	
	M		*M-1*		-M-1		$w = r_{max} - M + 1$	
h + M	M+1	$r_{max} + w + 1$	*2M*	*2M-h+1*	*0*	*-(h-1)*	$w = -r_{max}$	$-r_{max} \leq w < M - r_{max}$
	M+2		*2M-1*		*-1*		$w = -r_{max} + 1$	
	...		...		...		...	
	2M		*M+1*		-M+1		$w = -r_{max} + M - 1$	

Our purpose is to estimate $v'(w\Delta s)$. Given a *w* value, we can get *P* and *N*, which determines which function values $v(w\Delta s + k\Delta s)$ to be used. The case number *h* (or *h+M*) determines which set of coefficients to be used.

For *h=0*, we already give the coefficients above.

For each $h \neq 0$ we also form a $2M \times 2M$ constant matrix, get its inverse matrix, taking the first row of the inverse matrix as the coefficients. See pseudo code on page 9, step 2 and step 3.