

The convolution computation for Perfectly Matched Boundary Layer algorithm in finite differences

Herman Jaramillo

May 10, 2016

1 Introduction

This is an exercise to help on the understanding on some important issues on (CPML) Convolutional Perfect Matching Layer algorithms to attenuate edge effects in finite differences. From the many references, I cite Roden and Gedney's [4], and Pasalic and McGarry [2].

Also, some good notes ¹ by Steven Johnson.

2 The Stretching Factor

The idea well explained by Steven Johnson is as follows. There are two zones. The inside zone where we want to preserve the numerics and a buffer zone (the boundaries and its surroundings) where we want to attenuate reflections so that the numerical experiments looks like simulated in a free space medium with boundaries at infinity.

At the buffer zone (some distance away from the center), the wave equations are analytically continued into the upper (or lower, depending on the Fourier transform being studied. For causal functions the signs of the \mathbf{k} vector and ω are opposite. That is, a plane wave is write in its simplest form as $e^{\mathbf{k}\cdot\mathbf{x}-\omega t}$.) half complex plane so that the oscillation character changes to an

¹ math.mit.edu/~stevenj/18.369/pml.pdf

exponential damping character. The trick is done by the change of variable (assuming 1D for simplicity, but the idea is general)

$$\tilde{x} = x + i f(x) \tag{2.1}$$

where $i = \sqrt{-1}$. The function f acts as the damping factor. When $f = 0$ then the problem is not changed. This occurs in the zone of interest. Based on physical intuition f is chosen so that

$$\frac{df}{dx} = \frac{\sigma_x(x)}{\omega}, \tag{2.2}$$

and this is explained next. The reason for the ω in the denominator is that this election will make the problem independent of frequency for a given reflection angle. That is, it will avoid dispersive (frequency dependent) behavior. To see this, we observe from equations 2.1 and 2.2 that

$$\tilde{x}(x) = x + \frac{i}{\omega} \int_{x_0}^x \sigma_x(x') dx',$$

with x_0 the coordinate where the damping layer starts.

See for example that for $f \neq 0$ in the new coordinate system a plane wave along the x (with $k = k_x = k_1$) direction, in the new coordinate \tilde{x} , could be represented as

$$e^{k\tilde{x}} = e^{ikx} e^{-\frac{k}{\omega} \int_{x_0}^x \sigma_x(x') dx'}$$

and since $c_p = k/\omega$ is the phase velocity (which is constant in non-dispersive media) we see that the attenuation represented by the factor

$$e^{-\frac{k}{\omega} \int_{x_0}^x \sigma_x(x') dx'}$$

is independent of frequency (for a given propagation direction). To be sure that the attenuation factor works correctly, we need to guarantee that $\int_{x_0}^x \sigma_x(x') dx'$ is positive for positive k , (assuming only positive frequencies ω . Now, if the plane wave is defined as $e^{-\mathbf{k}\cdot\mathbf{x} + \omega t}$ the signs for σ_i and k_i should be opposite).

Note that if the wave is traveling along the negative direction $k < 0$ and the integral is negative (since the upper index is smaller than the lower index and then still the integral contribution is negative, so still the exponent is negative) then also here the factor is a damping factor.

The simplest election of σ_x is a constant in which case we would get the mapping

$$\tilde{x} \mapsto \left(1 + \frac{i\sigma}{\omega}\right)x = s_x x$$

where s_x is called a stretching factor.

Roden and Gedney's [4] use a more general stretching factor (still with σ_i constant),

$$s_i = \kappa_i + \frac{\sigma_i}{\alpha_i + i\omega\epsilon_0},$$

with $\alpha_i > 0$ and $\kappa_i \geq 1$. While the factor κ_i can be seen as an overall scaling, the factor α_i is justified as shift to the pole away from the real axis, which will improve accuracy for grazing incidence due to source proximity to the boundary or large offsets. Martin et. al., [3] reaffirm Roden and Gedney's [4] statements.

Here and in what follows the subindex i means x , if $i = 1$, y if $i = 2$ and z if $i = 3$; for the three-dimensional space.

Note that in their work the sign convention is opposite, since the imaginary unit i is in the denominator. They describe the plane wave with a minus “-” sign in front of the phase $\phi = \mathbf{k} \cdot \mathbf{x} - \omega t$.

Martin et. al., [3] use the stretching factor

$$s_x = 1 + \frac{\sigma_x}{i\omega}$$

where, as in Roden and Gedney, the sign convention is opposite to the one used here. Komatitsch and Martin [1] also claim that with the help of the new parameter α introduced by Roden and Gedney, better accuracy is obtained at grazing incidence angles.

Since a product in the frequency/wavenumber domain is convolution in time/space domain, and the FDTD (Finite Difference Time Domain) wave equation is implemented in time domain, we need to find the inverse Fourier (Laplace) transform of the stretching factor. After finding the stretching factor we should apply convolution. Next section deals with finding the inverse Laplace transform of the stretching factor, and the final section shows a recursion formula that speeds up the convolutional implementation.

3 Laplace Transform of the Stretching factor

The stretching parameter is given by

$$s_i = \kappa_i + \frac{\sigma_i}{\alpha_i + i\omega\epsilon_0} = \frac{\alpha_i \kappa_i + i\omega\epsilon_0\kappa_i + \sigma_i}{\alpha_i + i\omega\epsilon_0} \quad (3.3)$$

so

$$\begin{aligned} \frac{1}{s_i} &= \frac{\alpha_i + i\omega\epsilon_0}{\alpha_i \kappa_i + i\omega\epsilon_0\kappa_i + \sigma_i} \\ &= \frac{1}{\kappa_i} \frac{\alpha_i + s\epsilon_0}{\alpha_i + s\epsilon_0 + \frac{\sigma_i}{\kappa_i}} \\ &= \frac{1}{\kappa_i} \left(1 - \frac{\frac{\sigma_i}{\kappa_i}}{\alpha_i + s\epsilon_0 + \frac{\sigma_i}{\kappa_i}} \right) \end{aligned}$$

So the inverse Laplace transform of $1/s_i$ is given by

$$\begin{aligned} \mathcal{L}^{-1} \frac{1}{s_i} &= \frac{\delta(t)}{\kappa_i} - \frac{\sigma}{\kappa_i^2} \mathcal{L}^{-1} \left(\frac{1}{\alpha_i + s\epsilon_0 + \frac{\sigma_i}{\kappa_i}} \right) \\ &= \frac{\delta(t)}{\kappa_i} - \frac{\sigma}{\kappa_i^2 \epsilon_0} \mathcal{L}^{-1} \left(\frac{1}{s + \frac{\alpha_i}{\epsilon_0} + \frac{\sigma_i}{\kappa_i \epsilon_0}} \right) \\ &= \frac{\delta(t)}{\kappa_i} - \frac{\sigma_i}{\kappa_i^2 \epsilon_0} e^{-\left(\frac{\alpha_i}{\epsilon_0} + \frac{\sigma_i}{\kappa_i \epsilon_0}\right)t} u(t) \\ &= \frac{\delta(t)}{\kappa_i} + \zeta_i(t). \end{aligned}$$

with

$$\begin{aligned} \zeta_i(t) &= -\frac{\sigma}{\kappa_i^2 \epsilon_0} e^{-\gamma_i t} u(t) \\ \gamma_i &= \left(\frac{\alpha_i}{\epsilon_0} + \frac{\sigma_i}{\kappa_i \epsilon_0} \right) = \frac{1}{\epsilon_0} \left(\alpha_i + \frac{\sigma_i}{\kappa_i} \right) \end{aligned}$$

This agrees with Roden and Gedney's [4] result.

4 Implementation by Convolution

At the end, we should convolve the function $\zeta_i(t)$ with a differential operator ∂_i . Let us refer to the time variable t as n for the n -th grid coordinate and

use a super-index for it. Let us call the time-convolution at some given x , as ψ_x^n and at the grid point $(x, n\Delta t)$. Then by the definition of convolution

$$\begin{aligned}\psi_i^n &= (\zeta_x * \partial_i)^n \\ &= \int_0^{n\Delta t} (\partial_i)^{n\Delta t - \tau} \zeta_x(\tau) d\tau\end{aligned}$$

We now make use of the fact that we have the data in a grid, and if a staggered grid method is used, the partial derivative ∂_i , is defined half a time step between $m\Delta t$ and $(m+1)\Delta t$, so

$$\begin{aligned}\psi_i^n &= \sum_{m=0}^{n-1} \int_{m\Delta t}^{(m+1)\Delta t} (\partial_i)^{n\Delta t - \tau} \zeta_i(\tau) d\tau \\ &= \sum_{m=0}^{n-1} (\partial_i)^{n-(m+1/2)} \int_{m\Delta t}^{(m+1)\Delta t} \zeta_i(\tau) d\tau \\ &= \sum_{m=0}^{n-1} Z_i(m) (\partial_i)^{n-(m+1/2)},\end{aligned}$$

where

$$\begin{aligned}Z_i(m) &= \int_{m\Delta t}^{(m+1)\Delta t} \zeta_i(\tau) d\tau \\ &= -\frac{\sigma_i}{\epsilon_0 \kappa_i^2} \int_{m\Delta t}^{(m+1)\Delta t} e^{-\gamma t} u(t) \\ &= \frac{\sigma_i}{\epsilon_0 \kappa_i^2} \frac{e^{-\gamma t} |_{m\Delta t}^{(m+1)\Delta t}}{\gamma} \\ &= \frac{\sigma_i}{\epsilon_0 \kappa_i^2 \gamma} e^{-m\gamma\Delta t} (e^{-\gamma\Delta t} - 1) \\ &= \frac{\sigma_i}{\kappa_i^2 \alpha + \kappa_i \sigma_i} e^{-\left(\alpha_i + \frac{\sigma_i}{\kappa_i}\right) \frac{m\Delta t}{\epsilon_0}} \left[e^{-\left(\alpha_i + \frac{\sigma_i}{\kappa_i}\right) \frac{\Delta t}{\epsilon_0}} - 1 \right] \\ &= a_i e^{-\left(\alpha_i + \frac{\sigma_i}{\kappa_i}\right) \frac{m\Delta t}{\epsilon_0}}\end{aligned}$$

with

$$a_i = \frac{\sigma_i}{\kappa_i^2 \alpha + \kappa_i \sigma_i} \left[e^{-\left(\alpha_i + \frac{\sigma_i}{\kappa_i}\right) \frac{\Delta t}{\epsilon_0}} - 1 \right] = \frac{\sigma_i}{\kappa_i^2 \alpha + \kappa_i \sigma_i} (b_i - 1)$$

as in Roden and Gedney's [4]. Here

$$b_i = e^{-\left(\alpha_i + \frac{\sigma_i}{\kappa_i}\right) \frac{\Delta t}{\epsilon_0}}$$

as in Komatitsch and Martin [1].

4.1 The recursion formula

We found

$$\psi_i^n = \sum_{m=0}^{n-1} Z_i(m) (\partial_i)^{n-(m+1/2)} = Z_i(0) (\partial_i)^{n+1/2} + \sum_{m=0}^{n-2} Z_i(m+1) (\partial_i)^{n-(m+1+1/2)},$$

with

$$Z_i(m+1) = a_i e^{-\left(\alpha_i + \frac{\sigma_i}{\kappa_i}\right) \frac{(m+1)\Delta t}{\epsilon_0}} = Z_i(m) e^{-\left(\alpha_i + \frac{\sigma_i}{\kappa_i}\right) \frac{\Delta t}{\epsilon_0}} = Z_i(m) b_i.$$

so

$$\begin{aligned} \psi_i^n &= Z_i(0) (\partial_i)^{n+1/2} + \sum_{m=0}^{n-2} Z_i(m+1) (\partial_i)^{n-(m+1+1/2)} \\ &= Z_i(0) (\partial_i)^{n+1/2} + b_i \sum_{m=0}^{n-2} Z_i(m) (\partial_i)^{n-1-(m+1/2)} \\ &= Z_i(0) (\partial_i)^{n+1/2} + b_i \psi_i^{n-1} \\ &= a_i (\partial_i)^{n+1/2} + b_i \psi_i^{n-1} \end{aligned}$$

since $Z_i(0) = a_i$. This is equation (26) in Komatitsch and Martin [1].

This equation, thanks to the recursive properties of the exponential, provides an efficient implementation of the boundary layer method.

4.2 A particular implementation

Komatitsch and Martin [1] implemented the CPMBL, as follows:

For the function σ_i they used the symbol d_i and defined as

$$d_i(x_i) = d_0(x_i/L)^N$$

where:

$$\begin{aligned}
 x_1 &= x, \quad x_2 = y, \quad x_3 = z \\
 N &= 2 \\
 d_0 &= -(N + 1)v \log \frac{R_c}{2L}, \quad v = \text{velocity} \\
 R_c &= 0.001 \\
 L_i &= n_i d_i = 100 \quad (\text{thickness of absorbing layer})
 \end{aligned}$$

Also they chose

$$\kappa_i = 1.0$$

The value of α_i is picked linear from the maximum at the PML entrance with a value of $\alpha_{max} = \pi f_0$ (with f_0 the dominant frequency of the source wavelet) and 0 at the boundary. At the boundary they imposed Dirichlet boundary conditions of $v_x = v_z = 0$.

The variables: d_i , a_i , b_i , and α_i should be computed both at any integer and integer+half grid location according to the staggered grid recipe.

References

- [1] Komatitsch M and R. Martin. An unsplit convolutional perfectly matched layer improved at grazing incidence for seismic wave equation. *Geophysics*, 72(5):SM155–SM167, 2008.
- [2] D. Pasalic and McGarry R. Convolutional perfectly matched layer for isotropic and anisotropic acoustic wave equations. *SEG Technical Program Expanded Abstracts*, pages 2925–2929, 2010.
- [3] Martin R, D. Komatitsch, and A. Ezziani. An unsplit convolutional perfectly matched layer improved at grazing incidence for seismic wave propagation in poroelastic media. *Geophysics*, 73(4):T51–T61, 2008.
- [4] J. A. Roden and S. D. Gedney. Convolution pml (cpml): an efficient fdtd implementation of cfs-pml for arbitrary media. *Microwave and Optical Technology Letters*, 27(5):334–339, 2000.