7

Transparent Boundary Conditions for Computational Domains - Part One

Boundary conditions in beam-propagation methods constitute an important topic that will be discussed at several points in this course. Here we address the two most simple, yet practically important approaches. The first one is a proper re-formulation of the boundary guard which we have already used on ad-hoc basis in several practical exercises.

7.1 Poor man‘s absorbing boundary improved

The boundary guard in the role of an absorbing boundary condition (ABC) has been characterized as a poor‘s man method for reasons that became apparent in several situations. In particular, we have seen that

- boundary reflectivity remains significant — there are ABC implementations that are more robust (e.g. they do not require parameter tuning), and also outperform the guard method in terms of achieving lower reflectivity
- performance depends on the chosen shape, or profile of the guard — a steep profile can produce sizable reflection
- performance also depends on how often the guard is applied — optimal parameters will therefore depend on the potentially variable integration step Δz (which may need adjustment e.g. when the evolution enters a region difficult to resolve)
- to avoid re-calculation of the guard windowing function, it can be applied periodically after fixed propagation distance intervals – this has the disadvantage that artificial modulations are introduced into the solution.

To formulate a better version of the absorbing layer playing the role of “transparent” boundary condition, one can apply methods described in the previous Section. In particular, in the propagation equation (6.3)

$$\frac{\partial \mathcal{E}}{\partial z} = \frac{i}{2k_0}\Delta_\perp \mathcal{E} + \frac{i\omega}{2cn(\omega)}\Delta\chi(x,y)\mathcal{E} \tag{7.1}$$

the medium susceptibility can be endowed with an artificial absorption which depends on the transverse location in the domain. For locations in the center of the computational “box” $Im\{\Delta\chi(x,y)\} = 0$, while the imaginary part increases as one approaches the domain edge. It is important to keep function $Im\{\Delta\chi(x,y)\}$ smooth, otherwise unwanted reflections can arise.

When working with this kind of absorption layer, it is convenient to parametrize the susceptibility in a way more directly tied to the effect it will cause. For example in the above equation the imaginary part of susceptibility can be

$$\frac{\partial \mathcal{E}}{\partial z} = \frac{i}{2k_0}\Delta_\perp \mathcal{E} - \frac{1}{\ell_{guard}} W(x,y)\mathcal{E} \quad , \quad 0 <= W(x,y) <= 1 \tag{7.2}$$

The properties of the boundary guard are thus separated into its strength, given by the characteristic absorption length ℓ_{guard}, and its shape given by the windowing function $W(x,y)$.

This formulation of course does not eliminate all drawbacks of the oversimplified implementations utilized in our initial exercises. However, it makes control of the numerical properties much easier. Most importantly, the absorptive effects and their spatial distribution do not depend (at least not too much) on the integration step. Secondly, by ensuring that W is bounded, one can avoid situations in which the length of the integration step Δz is actually bounded by the strength ob the artificial absorption.

The modified Crank-Nicolson method (Section 6.1) can be used to solve the system with absorptive layers attached to the sides of the domain.

Exercise: Calculate the dispersion relation for a homogeneous medium with a loss parametrized by ℓ_{guard} as in the above equation, and compare the resulting absorption rates for waves with zero and maximal transverse wavenumbers. The result should show that

- The resulting absorption length is different for ℓ_{guard}. For example, for zero transverse wavenumber it becomes
$$\frac{1}{\ell_{absorption}} = \frac{1}{\Delta z} \log\left[\frac{1+\frac{\Delta z}{2\ell_{guard}}}{1-\frac{\Delta z}{2\ell_{guard}}}\right]$$
- The effective absorption rate of waves with high transverse wavenumbers is less than that of waves with many grid points per transverse wavelength. This is yet another example showing that high spatial frequencies are in general more difficult to control.

7.2 Low-Cost Absorbing Boundary Conditions

The boundary treatment presented in this Section was first proposed by R. Hadley, and published in Opt. Lett. 16(1991)624 under the title *Transparent boundary condition for beam propagation.* The big advantage of the approach is that it requires negligible calculations, and that it is easy to implement. Moreover, it does not need additional absorbing layers to be added to the computational domain. This is why despite certain drawbacks, these boundaries are often used in practice.

The boundary problem in finite-difference beam propagation formulations originates in the need to evaluate discrete Laplacian operators for the solution at locations adjacent to the domain boundary. For example,

$$\partial_{xx} E \approx \frac{1}{\Delta x^2}\left[E(x-\Delta x) - 2E(x) + E(x+\Delta x)\right] \tag{7.3}$$

requires the value for $E(x+\Delta x)$ at $x+\Delta x$ which may correspond to a grid point "beyond" the edge of the computational box. In general, various methods of boundary condition implementation consist in constructing estimates for such "missing" field samples.

The idea proposed by Hadley is in spirit similar to what we encountered with the Mur's boundary treatment for a finite-difference Maxwell solver. There we assumed that in the immediate vicinity of the domain edge, the simulated solution only consists of waves that propagate

in the direction toward the boundary. Then it was justified to replace the wave equation by a one-way, first-order propagation equation.

Considering a computational domain that is restricted to $x <= 0$, we assume that at least for x not too distant from zero, the solution behaves as a plane wave, and is locally well approximated by

$$E(x) \approx A \exp[ikx] \ . \tag{7.4}$$

This represents a waveform that is about to propagate across the boundary and disappear. Here, the wavenumber k is not a priori known and has to be obtained from the available information about the simulated solution. Note that real values of k would represent a propagating solution that carries away portion of the beam's energy, while imaginary-valued k should occur for tails of evanescent waves. Energy should not be extracted through such an evanescent tail.

If (7.4) holds, the outside value of the field can be estimated through the ratio of the field intensity at neighboring sites like this:

$$E(x + \Delta x) \approx E(x) \frac{E(x)}{E(x - \Delta x)} \tag{7.5}$$

This can be often used to obtain reasonable values just outside of the computational domain. In particular, for a diffracting beam it works quite well.

Looking at the above expression, once can immediately see that there are potential issues:

- This approximation relies on a well-defined local wavenumber — which need not necessarily exist! Indeed, a simple superposition of two waves with different spatial frequencies can not be characterized by one.
- This boundary condition is not linear — one can rightly argue that mixing artificial nonlinearity into a purely linear problem is not healthy. Indeed, we will have examples when the boundary conditions based on the above nonlinear extrapolation performs rather poorly.
- This boundary condition may be problematic if the field present in the vicinity of the boundary consists of numerical noise. It works best if there is a smooth waveform that is easy to extrapolate.

The extrapolation ansatz (7.5) is not tied to a specific beam-propagation method. In principle, it can be applied with any finite-difference approach. Next, we look at the boundary condition implementation specifically for the Crank-Nicolson discretization.

7.2.1 ABC Modification of the C-N method

Recall that the Crank-Nicolson system of equations to solve at each step has the form

$$L^- E_{new} = L^+ E_{old}$$

where both sides contain a discrete Laplacian operator, an can not be applied to points directly at the boundary. The required modification is straightforward for the RHS, $L^+ E_{old}$, but nonlinearity gets in the way on the left-hand-side.

Modification in the right-hand-side of the C-N system:

An additional "ghost" grid point which will carry the amplitude value estimated as explained previously is added to the perimeter of the domain, in this case it is denoted as E_{N+1}

$$L^+ E_{old} = (1 + i\delta\Delta) E_{old} \tag{7.6}$$

$$(L^+E)_i = E_i + i\delta\Delta_{ij}E_j = E_i + i\delta(E_{i-1} - 2E_i + E_{i+1}) \tag{7.7}$$

At the boundary, $i = N$, E_{N+1} does not exist, and has to be "ghosted" by

$$E_{N+1} = e^{ik\Delta r} * E_N \equiv E_N/E_{N-1} * E_N \tag{7.8}$$

Thus, the last "row" of the RHS evaluation can be implemented through the same formula as for all the "inner" rows, but every time E_{N+1} is needed, it is first calculated as

$$E_{boundary} = E_N/E_{N-1} * E_N$$

and used subsequently in the finite-difference expression that applies to inner-domain grid points

$$(L^+E)_{N-1} = E_{N-1} + i\delta(E_{N-2} - 2E_N + E_{boundary}) \tag{7.9}$$

Recall that we have left "zero placeholders" in our implementation of the Crank-Nicolson discretization just for this purpose.

Modification of LHS:

The above can not be applied to

$$L^-E^{n+1} = (1 - i\delta\Delta)E^{n+1} \tag{7.10}$$

$$(L^-E)_i^{n+1} = E_i^{n+1} - i\delta\Delta_{ij}E_j^{n+1} = E_i^{n+1} - i\delta(E_{i-1}^{n+1} - 2E_i^{n+1} + E_{i+1}^{n+1}) \tag{7.11}$$

since the ratio E_N^{n+1}/E_{N-1}^{n+1} has not been evaluated yet. Instead, the same ratio from the previous step will be used,

$$E_{N+1}^{n+1} = e^{ik\Delta r} * E_N^{n+1} \approx E_N^n/E_{N-1}^n * E_N^{n+1} \ , \tag{7.12}$$

which is yet another example of *linearization* in the implicit discretization scheme. Having precalculated the ratio, the modification can be absorbed into the tri-diagonal matrix of the linear system. Consider the three diagonals of L^+, namely the arrays a, b, c, and the very last row of the system matrix L^+:

$$0 \ldots a(N), b(N)|c(N)$$

where $c(N)$ is actually unused because it is outside of the domain. Imagine it "meets" the ghost value given above; Because the latter contains the vector element that is normally multiplied by $b(N)$, the diagonal must be modified as

$$b(N) \to b(N) + c(N)E_N^n/E_{N-1}^n$$

In this formulation, $c(N)$ can stand for the upper-diagonal element of the linear-domain or radial-domain system. So this is directly applicable in both flavors of the C-N method we have discussed so far.