Table of Contents

Boundary conditions

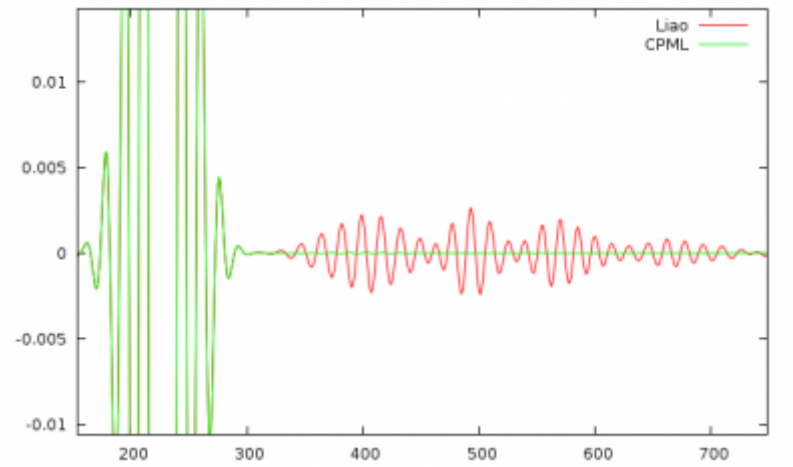
If we leave the computational volume as it is, the wave will be reflected from the boundary. In fact, this is similar to a reflection from perfect electric/magnetic conductor that would be located at computational volume faces as the tangential fields are nulled here (also normal components are nulled in this case). In order to simulate free space outside of a computational volume we need to use some special algorithm that will damp the wave leaving computational volume.

There are two absorbing boundary conditions implemented in GSvit. Liao boundary conditions is highly effective second order boundary condition, using space and time derivatives of the values at the computational volume boundary. Convolutional perfectly matched layer (CPML) is one of the best available boundary conditions for FDTD at present, providing a layer of graded optical properties and stretched coordinates that lead to absorption of an electromagnetic wave with almost no backreflection.

Sample parameter file: 2nd order boundary condition.
A 200x200x200 computational domain with single point source

Sample parameter file: CPML boundary condition.
A 200x200x200 computational domain with single point source.

To compare the efficiency of Liao boundary condition and CPML we performed a simulation of a short pulse traveling in the computational volume, showing the backreflection from one side. As seen below, CPML provides much smaller reflection. The difference between both approaches is even more pronounced if we consider waves that are incident at high angles with respect to computational volume boundary normal.



A special case is periodic boundary condition. Here the fields from one side of computational domain are copied to the other side so the space becomes periodic. Note that this approach is not valid for some cases, e.g. for planar wave illuminating the computational volume from an angle in TSF source formulation. To overcome this a special algorithm is necessary, not yet implemented in GSvit.

The use of boundary conditions in GSvit can be independent for every face - so it is possible to use nonabsorbing boundary condition for some faces of the computational domain only or to use different boundary conditions for different faces.