Table of Contents

Multithreading

Most of the processors have multiple cores, so using multiple threads can directly speedup the calculation, with almost linear scaling. For this, OpenMP library is used.

To use the mutithreading, set the number of threads in the Basic parameters menu (in XSvit graphical user interface) or use the THREADS directive if you write the parameter file directly. Note that on some systems the apparent number of cores available in the system is doubled (due to hyperthreading feature of the processors), but this not speedup anymore the calculation. It is always good to check up to which number of threads the calculation is speeding up. On large scale computational systems we have observed that the first significant drop from scaling linearity is when the memory needs to be accessed from different boards. As FDTD is quite simple, but memory transfers demanding, this performance drop is significant, so in practice we rarely use GSvit on more than 16 cores at the same time. This might differ if your HPC system is using a different architecture.

In the following example we have changed the number of threads from 1 to 4 on a workstation that we use for most of the calculations (2x Intel Xeon processor with 8 cores) and on a more regular computer (Lenovo Yoga notebook, Intel Core i7 processor). The computation time for different number of threads is shown below.

Sample parameter file: use of multithreading.
A 200x200x200 computational domain with a plane wave and dielectric sphere, run on 4 threads of the processor.