Exploiting Multithreading of Modern CPU

Introduction

WIPL-D Pro is a well-established full wave 3D EM solver based on Method of Moments (MoM). Among the most recognized advantages of the code over the competitors, it is important to mention speed and accuracy of results.

Inherently, MoM is superior for simulation of electrically moderate and large problems, as well as radiating (open region) devices. In that sense, it is the default choice for all large-scale simulations, while the other techniques such as FEM, FDTD are mostly used for electrically small or geometrically complex problems (with numerous features and details).

However, WIPL-D MoM implementation is unique in so many ways. First of all, the mesh used is quadrilateral. Most available mesh algorithms are available for the triangular based mesh, which is used for almost every available full-wave EM tool. The quad mesh reduces number of required unknowns and mesh elements twice. The drawback is the complexity of meshing an arbitrary structure into quads. WIPL-D has developed in-house algorithms which yield optimum quad meshes.

Another advantage is the use of higher order basis functions (HOBF). Traditionally, rooftop or the first order elements are used in most EM codes, resulting in lambda/10 to lambda/6 mesh triangular elements. Even electrically small structure meshed that way comprises of rather large number of mesh elements. WIPL-D implementation uses adaptive order of polynomial representing current on the mesh element. Smallest details have the traditional first order, but 2 lambda mesh elements use the 8th order. That way, single model consists of mesh elements of dramatically variable size. Large metallic scaterreres can have numerous 2 lambda by 2 lambda quads on flat surfaces with even 1/100,000 lambda small details at the highly curved surfaces, or at complex details.

Such unusual approach has challenged WIPL-D developers to improve the kernel over the years. The MoM simulation consists of two main steps, fill-in of the MoM matrix and the matrix inversion. The matrix inversion is efficiently solved through well known LU decomposition. There are proven technologies such as Intel MKL library for the CPU platforms. The revolution of the field came with usage of the inexpensive CUDA enabled GPU cards. All this led to WIPL-D having fast MoM matrix solution over the years. But the process of matrix fill-in is still done through in-house routines, which are constantly improved.

The most important aspect of the matrix fill-in is to exploit the advantage of modern CPUs, which all have multi-core capabilities. Having a quad core CPU has become a standard for the desktop and laptop PCs, and is emerging as the standard for various mobile devices. WIPL-D solution has supported this fact, but has also supported the fact that modern desktop CPUs are available with 6, 8, 10, 12, 16 or more cores. The modern mother boards support having more than one such CPUs inside of affordable desktop PC or a small workstation (mostly 2 CPUs but also 3, 4…). In that sense, its not unusual to use desktop PCs with 20, 24 or 32 cores. If the multi-threading is active, number of threads is double of the number of cores: 40, 48 or 64.

WIPL-D kernel fully exploits such a huge advantage of modern PCs with highly efficient matrix fill-in where the efficiency is traditionally in the range between 95 and 99%. Electrically small structures, or structures with numerous details are the typical application where the modern WIPL-D MoM is used equally with the methods such as FEM or FDTD.

A specific attention should be devoted to ultra-wide band devices. This is a common request from modern communication applications. MoM requires that each frequency point is solved separately. The frequency band is thus subdivided into finite number of frequency points and then they are solved as individual EM problems. WIPL-D uses powerful in-house algorithm for interpolation of YZS or far-field results. This leads to minimum number of frequency points. However, regular UWB band can be as spread as 2-18 GHz and may require 10,20 or 50 frequency points with numerous resonances.

Another important aspect are complex CAD geometries. Quite often the realistic devices have tremendous number of details. A few examples can be: mobile phone, laptop, multilayer PCB boards with dozens of lines and pads. Although electrically small or moderate, such structures can have thousands or tens of thousands of mesh elements. Such small elements apply the first order of current polynomial, but number of unknowns in such EM problem can be quite large because of high number of mesh elements. With the assistance of inexpensive GPU cards (and the WIPL-D GPU solver), the matrix inversion is done in minutes on inexpensive hardware platforms. The solution which allows that MoM is used for such problems as easily as FEM or the FDTD are the multicore inexpensive modern CPUs. This application note will demonstrate how using such hardware can tremendously improve the simulation times and usability of the code for electrically small problems with numerous details or the UWB applications.

Benchmark models

The EM models used here are chosen either because of their complexity or the wide simulation band.

The first example is ridged horn operating in a very wide frequency band, showing multiple resonances. Another important aspect that the feeding of the structure is realistic, showing coaxial cable penetrating through the bottom wall into the designed back cavity. The coaxial cable is loaded with the dielectric. This results with over 250 quads (called plates) with only 3,300 unknowns needed for the simulation.

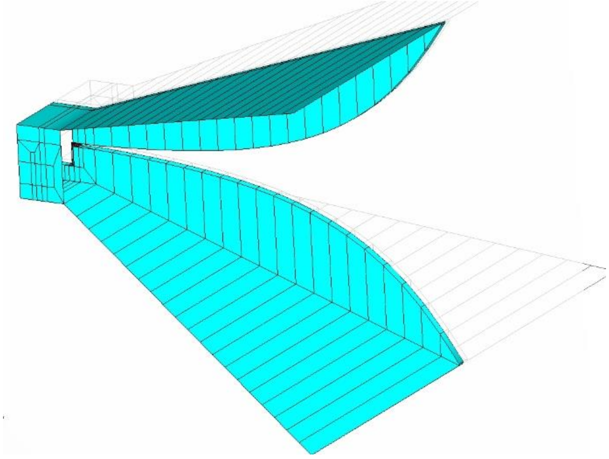

Figure 1, Ridged waveguide horn

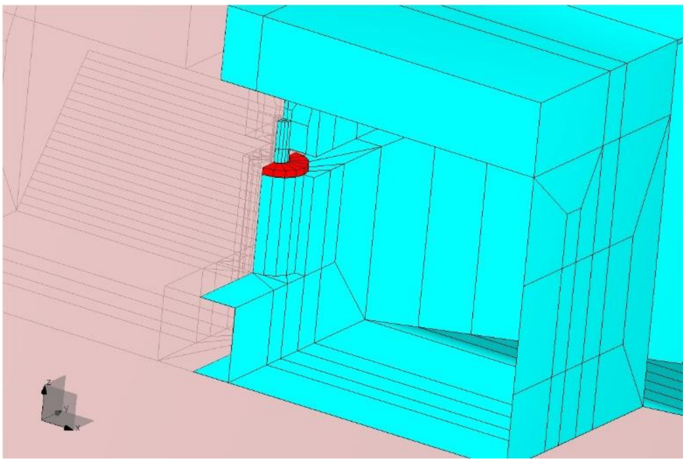

Figure 2, The details of the feeding

The return loss indicates numerous resonances. Owing to powerful interpolation, it is more than sufficient to simulate the model in 41 frequency point in a quite wide frequency band.

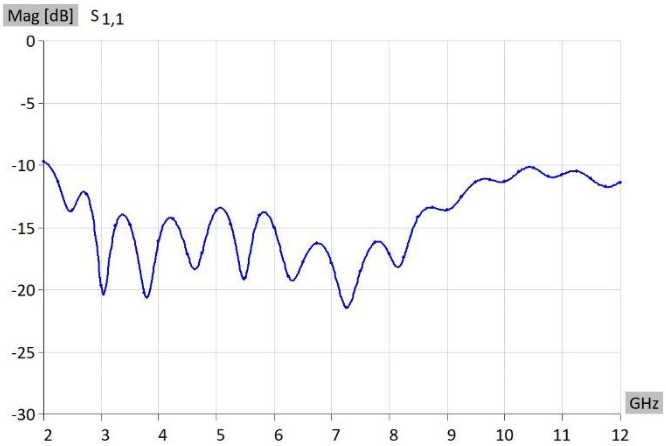

Figure 3, Return loss in the wide band

Another example is sinuous antenna printed on the dielectric substrate, highly complex with 10 turns and 4 arms. The final model has over 4,000 plates, with only 22,000 unknowns used. Practically the first order is used mostly throughout the model, without the use of HOBFs.

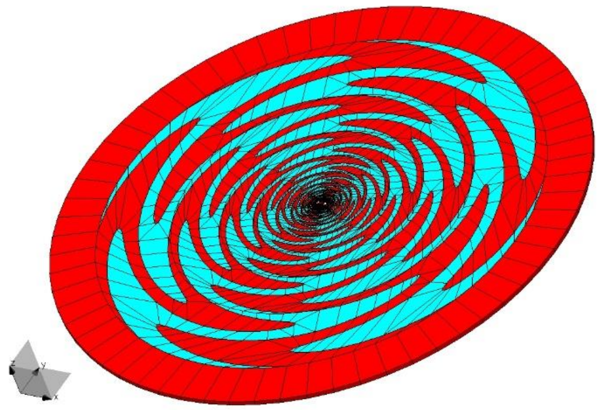

Figure 4, Sinuous antenna printed on the dielectric substrate

As a self-complementary structure, the real part of the antenna impedance should be 189 Ohms in a very wide frequency band, depending only at the size of the smallest and largest turns. In this case, the frequency band is practically 2-18 GHz.

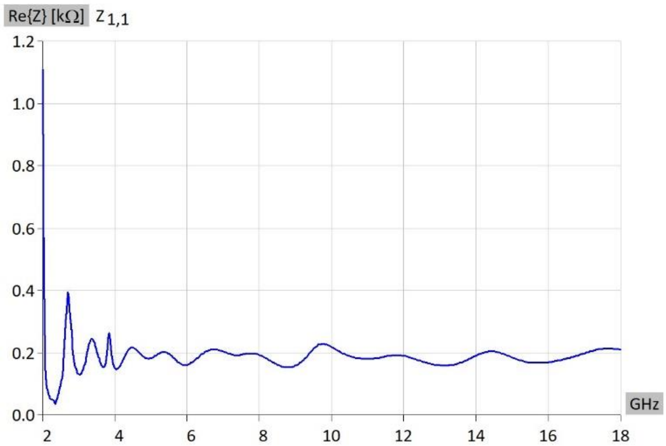

Figure 5, Antenna impedance in wide frequency spectrum for the sinuous antenna

In order to carefully determine the characteristic, the model was simulated in logarithmic scale. The majority of frequency points is required in the lower part of the band. Owing to large number of resonances, consequence of 10 turns, the number of frequency points was 49.

Table 1 illustrates the simulation times at regular desktop PC with 4 cores and a small workstation (2 CPUs, each with 12 cores).

Table 1 Summarized results.

Model	Number of unknowns	Number of frequency points	Matrix-fill in 4 cores [sec]	Matrix-fill in 24 cores [sec]
Ridged horn	3,395	41	85	37
Sinuous	21,710	49	3,283	882

The 4 cores PC has the following CPU:

Intel Core i7 7700 3.60 GHz

The 24 cores PC has the following CPUs (2 CPUs in total):

Intel Xeon Gold 5118 2.30 GHz.

Table 2 illustrates the proportion between the matrix fill-in and the matrix inversion if the entire simulation was done by using CPU. It can be seen that even the EM problem with the significant number of mesh elements, significant number of unknowns and high number of frequencies can be solved by using multithread CPU configuration in reasonable time.

Table 2 Matrix fill in and matrix inversion with 24 cores.

Model	Number of unknowns	Number of frequency points	Total matrix-fill in [sec]	Total matrix inversion [sec]
Ridged horn	3,395	41	37	16
Sinuous	21,710	49	882	1,244

As a last comparison, for the more demanding example of sinuous antenna, we compare the quad core desktop who is empowered with the inexpensive GPU card, Nvidia GTX 1080. The simulation times are this time given per frequency point for easier comparison, while the previous tables indicated total simulation times for the entire frequency band.

Table 3 Four cores desktop with single GPU compared to 24 cores workstation.

Configuration	Number of frequency points	Matrix-fill in [sec]	Matrix inversion [sec]
4 cores with Nvidia GTX 1080	1	67	18
24 cores	1	18	25

It can be observed that adding even a single GPU card to a standard desktop PC allows simulation of rather large models (~22,000 unknowns) faster than a workstation with 24 cores. Adding the inexpensive GPU card is a preferred solution in terms of matrix inversion and the configuration price. However, the 24 cores configuration speeds up both the matrix fill-in and the inversion. If possible, the ideal configuration is multi-thread workstation with even a single inexpensive GPU card.

Conclusion

The focus of the application note is set of examples where the advantages of MoM are theoretically minimal. The wide band and number of resonances requires large number of frequency points, which are simulated independently.

In addition, the examples are electrically small or moderate size structures where the number of unknowns is low, while size of elements prevents usage of HOBFs and the number of mesh elements is relatively high. The examples also include usage of dielectrics, in which case the matrix fill-in phase is even longer.

The simulation times are compared for a modern quad core CPU in a regular desktop PC (inexpensive configuration). The other platform is again a modest multi-core configuration with two affordable CPUs, each with 12 cores. The speed and performance of cores are significantly lower than at the quad core desktop (a consequence of difference in CPU speed 3.60 vs 2.30 GHz, cooling of multicores, required reliability of server CPUs etc).

However, highly efficient parallelization of the matrix fill-in phase is done at both platforms. Using the available multi-thread modern configurations with 20+ cores improves the simulation times for UWB applications or electrically small antennas with many details dramatically.

In the case of electrically large problems, the matrix inversion is usually the dominant phase in total simulation time. Parallelization efficiency of matrix fill-in increases with increasing number of unknown coefficients, i.e. electrical size of the problem. On the other hand, matrix inversion is done by using Intel MKL library, which also shows excellent efficiency of parallelization. All this extends the range of EM problems that can be solved by only using CPU with acceptable simulation times for EM and RF engineers.

At the end, the simulation times indicate that adding even a single inexpensive GPU card to standard desktop is the optimum solution for speeding up the matrix inversion. Using multiple cores is the solution for speeding-up the matrix fill-in. Combined, an ideal workstation for WIPL-D simulations is a multi-thread CPU with at least one inexpensive GPU card added for the matrix inversion phase.