Microstrip Patch Array with Feeding Network

Microstrip patch antennas are widely used in applications where small size, light weight, low cost and ease of installation are strongly required. These antennas represent low-profile devices conformable to both planar and non-planar surfaces. In addition, microstrip patch antenna arrays are used to achieve much larger antenna gain and\or beam steering. Generally, antenna arrays require more sophisticated numerical simulation analysis than standalone antenna element. Thus, a microstrip patch antenna array with feeding network will be simulated using WIPL-D Pro – a Method-of-Moments based full 3D EM solver.

WIPL-D Models

The WIPL-D Pro model of simulated microstrip patch antenna array is shown in Figure 1. In WIPL-D Pro, a microstrip patch array antenna can be modeled in several ways. One approach includes meshing of a single element and usage of WIPL-D features (*Anti-*)*Symmetry* and *Manipulations*/*Copy*. In this approach, only quarter of the given antenna is modeled. A single element mesh is multiplied to obtain the array model, along with the feeding network. Metal parts are considered to be perfectly conducting. The metallic losses can be added via the Distributed loading feature, without increasing the simulation requirements.

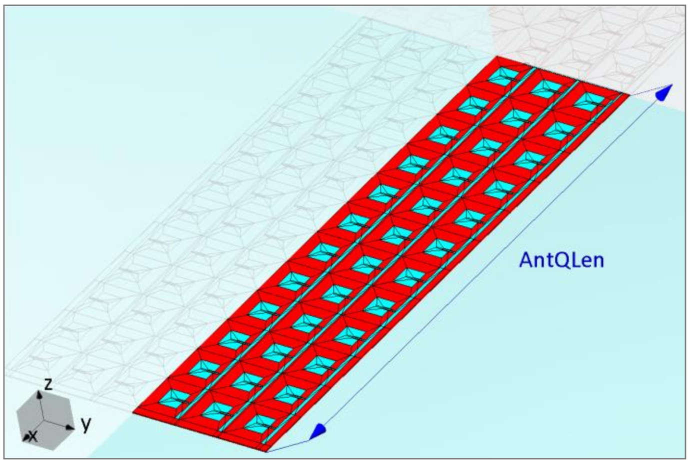

Figure 1. Quarter of microstrip array antenna

Simulated microstrip array antenna consists of 144 elements. Operating frequency is 24.2 GHz. This means that wavelength in the free space is about 12.4 mm. The half-length of the array (Figure 1) is 107.8 mm (about 9 wavelengths). The width of the array is approximately four times less than length. The dielectric substrate parameters are: $\varepsilon_r = 2.2 + j \cdot 0$ and $\mu_r = 1 + j \cdot 0$. Single antenna element is shown in Figure 2.

Simulations and Results

The model of microstrip patch array was simulated using standard desktop Intel® Core™ i7-7700 @ 3.60 GHz empowered with single low-end GPU card Nvidia GeForce GTX 1080.

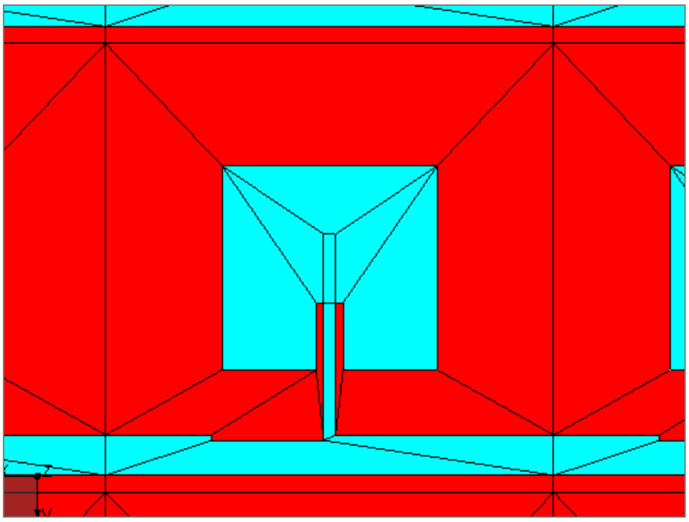

Figure 2. An element of microstrip array

3D radiation pattern is shown in Figure 3 while 2D radiation pattern (a phi-cut; Phi=0 Degrees) is shown in Figure 4.

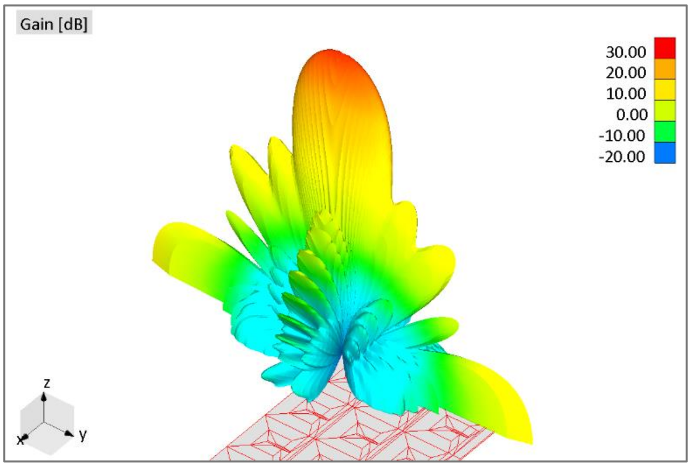

Figure 3. 3D radiation pattern with antenna array

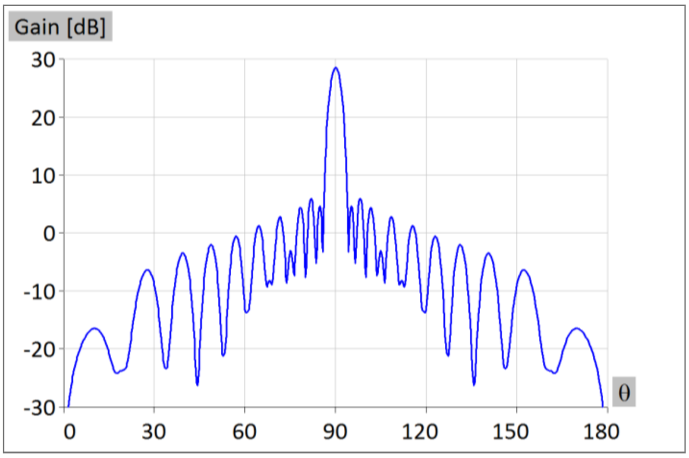

Figure 4. Radiation pattern, phi-cut (Phi=0)

Number of unknowns, computer memory required and CPU and GPU simulation times are presented in Table 1. CPU simulation time encompasses simulation time when both matrix fill-in and matrix inversion are performed on CPU. On the other hand, GPU simulation time encompasses simulation time when matrix fill-in is performed on CPU while matrix inversion is performed on GPU. No time used for post-processing is included in these simulation times, since it is identical in both cases and negligible for the total simulation time.

Table 1. Number of unknowns, computer memory required, CPU time and GPU time

Number of unknowns	Memory [GB]	CPU simulation time [sec]	GPU simulation time [sec]
23,705	4.2	341	256

Conclusion

We have demonstrated an approach with proper usage of WIPL-D features (*Symmetry* and *Manipulations*) enabling modelling the structures such is a microstrip patch antenna array. Furthermore, it is mentioned that only quarter of the structure was modeled.

Generally, if possible, only quarter of a simulated structure should be modeled (and simulated). Simulating quarter of the structure saves utilized computer resources and decreases simulation time. The simulation of quarter of the structure is particularly important for electrically large models.

WIPL-D Pro software enables successful simulations of this antenna by using Method-of-Moments (MoM). The WIPL-D MoM is particularly efficient for this type of simulation since it applies higher order basis functions (HOBFs) comparing to traditional low order rooftop functions (RWG). This yields usage of up to 2 wavelengths large mesh elements comparing to wavelength/10 mesh elements. This can be recognized in Figure 2 where the half of the wavelength patch is modeled by using couple of mesh elements without subdividing the mesh further. In addition, the WIPL-D kernel is based on quad meshed elements which inherently decreases number of unknowns.

The code execution is efficiently parallelized at multicore CPUs. This leads to low simulation times at inexpensive standard desktop PCs. The simulation can be even more efficient by using GPU cards (simply adding a GPU card to the existing hardware configuration). This is documented in Table 1 and it is particularly important for electrically large structures (e.g., arrays). In terms of WIPL-D simulation, the array simulated in this application note can be considered as small or moderate. Large arrays can include hundreds of array elements.