WIPL-D Pro v15: What is New in Numerical Engine?

New features/improvements introduced in v15 are:

1. **NEW generation of GPU Solver – 100,000 unknowns solved in less than 5 minutes!**
   a. **improved efficiency of a multiple GPU setup**
   b. **support for arbitrary number of GPUs**
   c. **ultra-fast in-core solution for multiple excitations**
2. **RCS calculation using real life antenna excitation - emulation of RCS measurements**
   a. **target illuminated by a real antenna radiation pattern (spherical wave)**
   b. **RCS at finite and infinite distance**
   c. **calculation of bistatic and monostatic RCS**
3. **Advanced GUI options**
   a. **exporting videos from the field and the currents animation**
   b. **graphic representation of currents for selected domains**
   c. **advanced phase center calculation**
   d. **real-time graphic representation of the cost function in Optimizer**
   e. **saving the specification of used hardware in the OWP file…**
4. **Enhancements in DDS (Domain Decomposition Solver)**
   a. **Memory requirements reduced by an order of magnitude**
   b. **efficient handling of multiple excitations**
   c. **fast convergence for structures with cavities**
   d. **GPU acceleration for sub-domain solutions**
   e. **comparison of results after each iteration**
5. **Fast in-core solution for multiple excitations where CPU is used for matrix inversion**
   a. **block forward/backward substitution**
   b. **applicable to monostatic RCS and antenna arrays**
6. **Memory/Solution management**
   The user can choose the classes of problems where the solution for current coefficients should be stored for future reuse. The data format can also be chosen.
7. **Improved CMA Solver**
   a. **increased accuracy for eigenvalues and eigenvector calculation**
   b. **improved mode tracking**
8. **GPU Cluster Solver**
   a. **advanced resource management**
   b. **more flexible licensing …**

1. NEW generation of GPU Solver

In the previous versions, the WIPL-D GPU solver was introduced to speed up the most time-consuming part of an EM simulation: MoM matrix inversion. Hardware configurations with up to 3 GPU cards were supported.

The efficiency of the method for multiple GPU platforms corresponded to the performance of relevant hardware platforms, mostly the available RAM, hard disks and CUDA enabled platforms. However, the performance and capabilities of hardware platforms is constantly growing.

Regarding the number of supported GPUs, the majority of contemporary motherboards enable connecting 3 to 4 GPUs. Additionally, a personal computer equipped with GPU Expander boxes could easily and cost-effectively utilize 8 or 16 additional GPUs. Motivated by the very successful utilization of GPUs for matrix inversion, and the parallelization algorithm utilizing up to 3 GPUs, we improved the algorithm to increase the efficiency for a greater number of GPUs. The maximum number of GPUs tested with the new algorithm is 8, which is in line with the most advanced hardware platforms available.

The improvements can be briefly summarized into 3 segments.

a) Improved efficiency of a multiple GPU setup

In the past few years GPU cards have become increasingly faster, cheaper and equipped with more RAM. In the same time, hardware platforms with up to 16 GPUs have emerged. As GPU VRAM is still rather limited in capacity, the matrix inversion is carried out by using block LU decomposition. The entire system matrix is stored in CPU RAM or on hard disks, in case of an out-of-core simulation, while only the processed blocks are stored in GPU VRAM. The time required for matrix multiplication of blocks is the most significant part of the overall time needed for solving large systems.

Parallelization of matrix multiplication in the new algorithm is performed on the level of a single block, making the load balance between several GPUs practically perfect, and independent of the number of GPUs and the number of matrix blocks.

b) Support for an arbitrary number of GPUs

Having in mind the new algorithm which has nearly perfect efficiency when several GPUs are used, the algorithm was successfully tested on platforms with more than 3 GPUs (up to 8). The efficiency of parallelization increases with the increase of the number of unknowns. Using more than 3 GPUs provides significant acceleration when the number of unknowns is higher than 20,000.

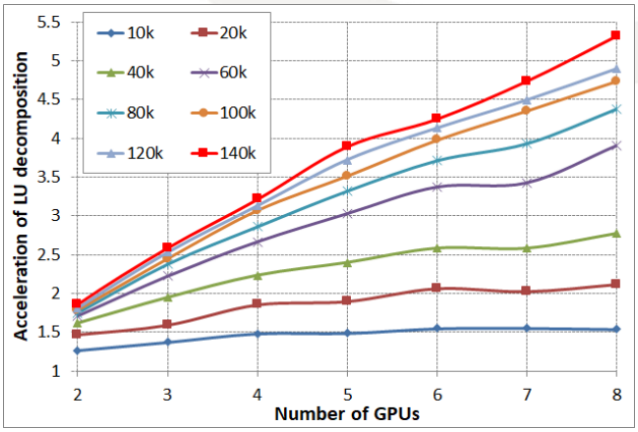
c) Ultra-fast in-core solution for multiple excitation

In the GPU solver from previous versions, the solution of monostatic RCS problems with a large number of radiation directions (e.g. 901, 1801, 3601 etc.) for electrically large EM problems was highly efficient only for out-of-core solutions. Forward and backward substitution are performed over blocks of excitations, rather then on a single excitation at a time. In v15, the same concept was applied to the in-core solution, resulting in an ultra-fast solution for multiple direction RCS problems (hundreds and thousands of directions).

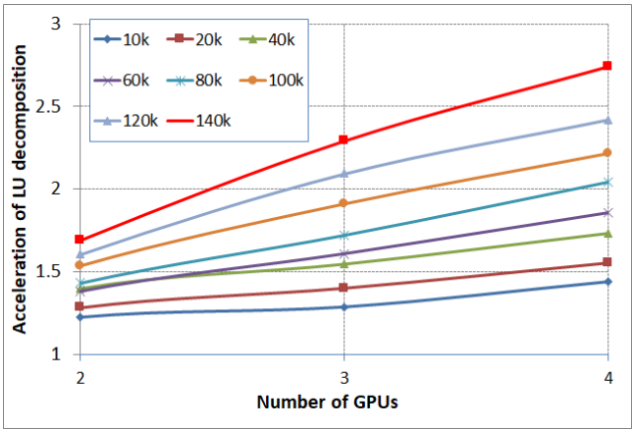
Examples:

In order to demonstrate the efficiency of the new algorithm, we present several studies. Single precision simulations are used in all of the studies. The same PC is used for all numerical experiments: Intel Xeon CPU E5-2660 v2 @2.2 GHz (2 processors), 256 GB of RAM, Win 10 Professional 64-bit. Two different GPU configurations are used: a) (up to) 8 identical GPUs GeForce GTX 680 (GPUs are placed in the Cubix XPRM-G3-82A GPU-Xpander); b) (up to) 4 identical GeForce GTX 1080 Ti GPUs. For GPU acceleration we used NVIDIA CUDA version 8.0.

First, the LU-decomposition for problems of different size, starting from 10,000 to 140,000 unknowns is presented. Acceleration of LU-decomposition versus the number of GPUs, for different number of unknown coefficients, is shown in diagrams in Fig. 1.



a) GPU type: GTX 680



b) GPU type: GTX 1080 Ti

Fig. 1. Acceleration of LU decomposition versus number of GPUs

LU-decomposition times for problems whose number of unknowns is between 100,000 and 140,000 are shown in Fig. 2. It can be seen that LU decomposition of 100,000 unknowns using 8 GTX 680 GPUs takes 7.7 minutes, while the same job is finished in 3.7 minutes with 4 modern GTX 1080 Ti GPUs.

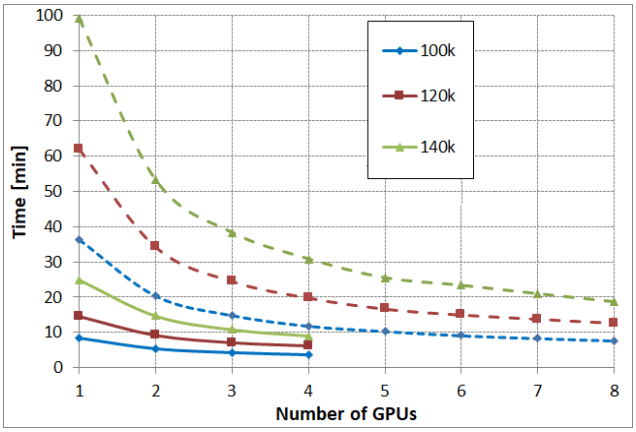


Fig. 2. LU decomposition time: 8 x GTX 680 (dashed line) and 4 x GTX 1080 Ti (solid line).

The new acceleration algorithm has been applied to a problem of antenna placement on an F-16 fighter. A half-wave dipole with a dielectric radome is placed on the fuselage of an F-16, as shown in Fig. 3. Frequency of interest is 2.5 GHz. Electrical length of the model is approximately 125 wavelengths.

Number of unknowns is equal to 138,193. Total simulation time on the aforementioned hardware platform (with 4 GTX 1080 Ti GPUs) is 19.5 min. Matrix fill-in requires 7.9 min, while additional 2.5 min is needed for forward and backward substitutions and post-processing. Finally, LU decomposition time is 9.1 min.

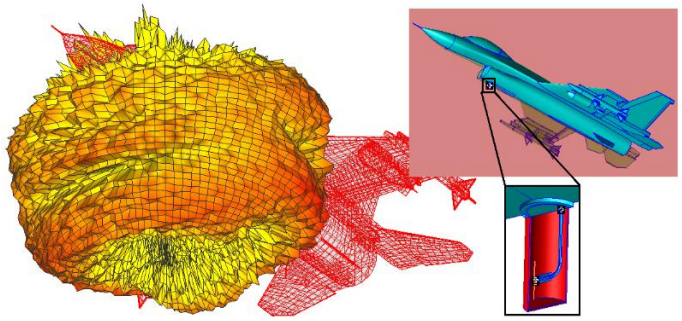


Fig. 3. Dipole antenna placed on F-16 fuselage, with the simulated radiation pattern.

The second example is significantly more demanding and requires a much higher number of unknowns: a T-72 tank at 3.6 GHz. The tank is 7.4 m long, which is 89 wavelengths. The problem is solved without applying symmetry as bistatic RCS in 1801 directions in the horizontal plane. The model originates from a CAD file and all details were kept during the mesh and simulation. The RCS wave arrives backside in the horizontal plane.

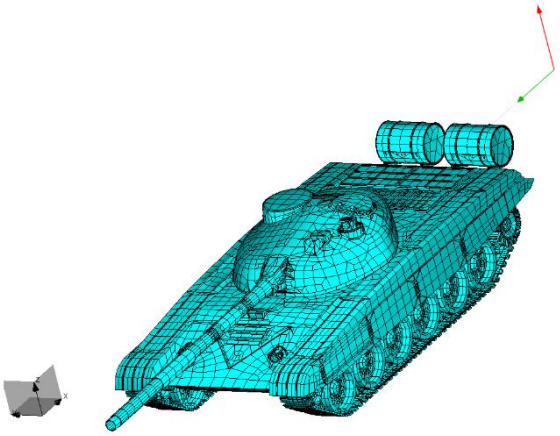


Fig. 4. T-72 tank meshed at 3.6 GHz

The hardware platform was on Amazon cloud solution supporting 8 Nvidia Tesla V100 GPU cards. The same simulation speed can be achieved by using significantly more affordable Nvidia GTX-1080 Ti. The problem requires 833,323 unknowns. The matrix inversion was run as "reduced" by using the out-of-core solver since the problem requires more than 5 TB of memory, which is larger than the available RAM size. The entire simulation lasted 20.8 h, where the matrix inversion lasted for approximately 16.4 h.

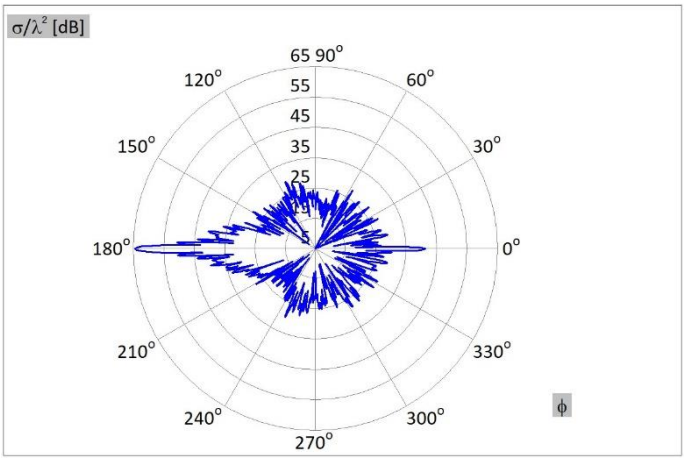


Fig. 5. Bistatic RCS of a T-72 tank at 3.6 GHz

2. RCS calculation using Field Generators Excitation

Traditionally, the RCS of an object is calculated by using plane wave excitation. However, the scattered field can also be calculated when the excitation is in the form of a real-life antenna pattern or an analytically defined radiation pattern. The incoming wave for those cases is not a plane wave, but a spherical wave. When the distance between the illuminator and the target is sufficiently large, the RCS result will coincide with the plane wave scatterer results. Depending on the complexity of the illuminating pattern, size of the target, the distance where the spherical wave turns into a plane wave can be 100, 1000 wavelengths or even more.

The improved numerical kernel in v15 simulates this type of problems in several steps. In the first step, the electric field of the field generator is calculated at the exact location of the target in free space. This field will be referred to as the incident field. It can originate from an imported radiation pattern of an antenna or from analytically defined field generators. In this step, the kernel also simulates the model consisting of the target object and the field generators, and then determines the current distribution over the target object. It is important to note that the coupling effects, in the sense of influence of the illuminated object on the radiation pattern of the excitation antenna, are not taken into account.

In the seconds step, the field generator is annulled. The scattered electric field is calculated as the field generated by the current distribution determined in the first simulation step. The RCS is then calculated from the scattered field and the incident field.

The simulation can be run as either monostatic or bistatic RCS. In the bistatic regime, the position of the field generator is fixed all the time (defined by the user).

In the monostatic regime, the field generator takes different positions (according to directions specified for far-field calculation). The orientation of the radiation pattern versus the illuminating object remains constant during simulations.

In CEM tools, the problems in simulation of an illuminator of a faraway target are: the electrical size of the target object, and the distance between the test antenna and the target object. The most efficient full wave solution of electrically large open area problems is based on the method of moments (MoM) applied to surface integral equations (SIEs). The simulations requirements are dependent only on the surface area in terms of lambda squared (there is no influence of the volume occupied by the target or the distance between the target and the antenna).

Examples:

The first example is the bistatic RCS of an aircraft scatterer illuminated by a field generator. The frequency of interest is 1 GHz, the length of the aircraft is around 10 m (almost 34 λ). The excitation is an analytically defined radiation pattern – a cosine shaped beam.

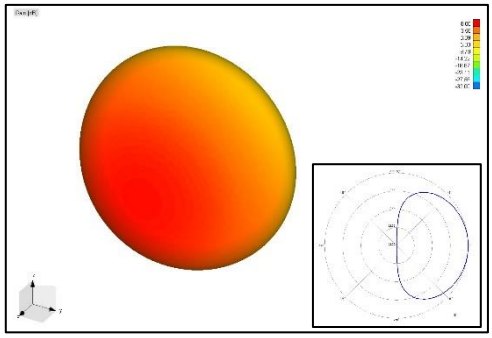


Fig. 6. Analytically defined field generator (cosine shape)

The incoming direction of the wave is to the nose of the aircraft, while the bistatic is calculated in the entire horizontal cut. The distance D is varied: 100 λ, 300 λ, 1000 λ and 3000 λ.

The RCS is calculated at a finite distance. For every simulated distance D, the radiation pattern is calculated at the same finite distance (and not in the infinity). This emulates the RCS measurements.

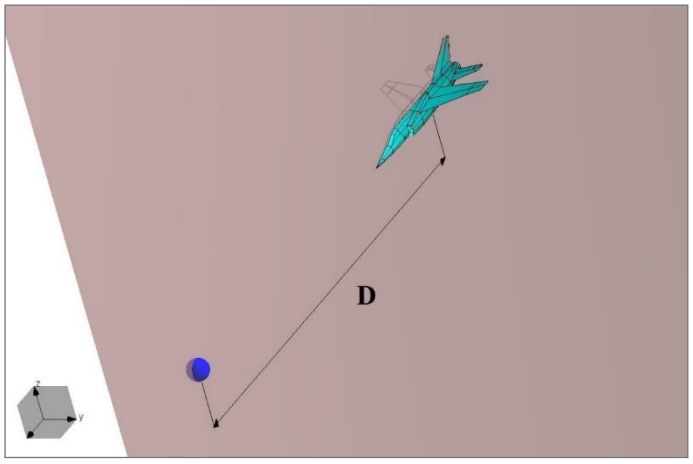


Fig. 7. Aircraft target illuminated by a field generator

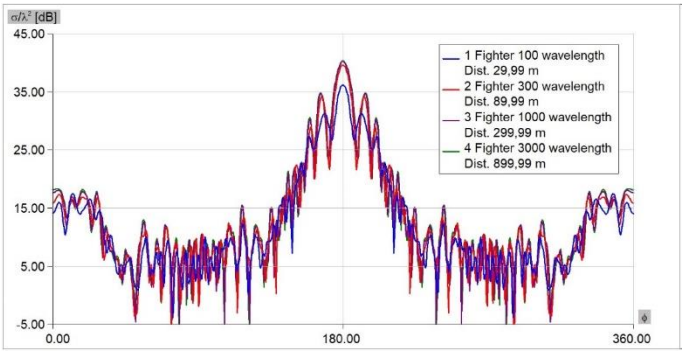


Fig. 8. Finite distance bistatic RCS for 4 different distances between field generator and target

If the distance is sufficiently large, the infinite distance RCS of the field generator excitation and the plane wave RCS must be identical. The results plotted in Fig. 9 confirm the accuracy of the calculations.

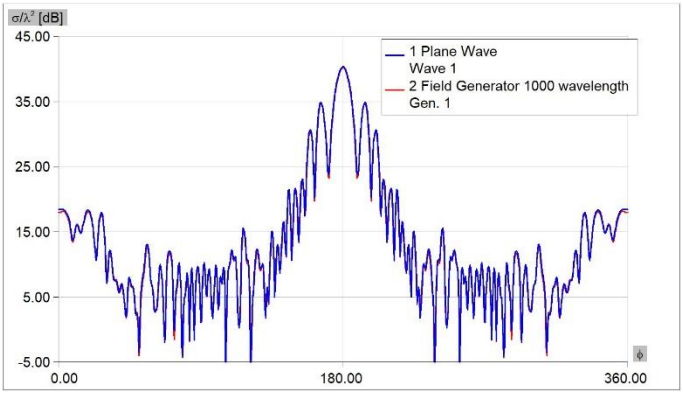


Fig. 9. Validating the accuracy of infinite bistatic RCS of a field generator excitation

The configuration used in the second example is identical to the first one, but the RCS regime is monostatic. The monostatic directions are in the vertical plane.

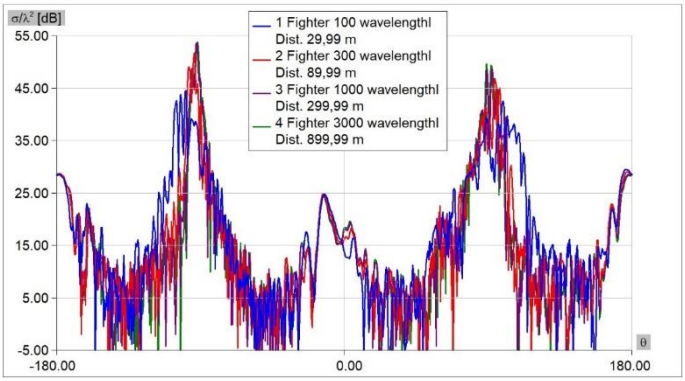


Fig. 10. Finite distance monostatic RCS for 4 different distances between field generator and target

Again, the accuracy is validated in Fig. 11 by comparing the infinite distance RCS with the plane wave RCS.

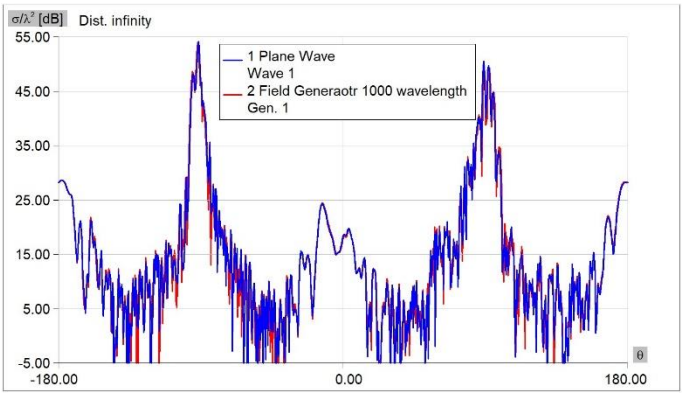


Fig. 11. Validating the accuracy of infinite monostatic RCS of a field generator excitation

3. Phase Center Calculation

WIPL-D Graph Viewer enables the calculation of the phase center for any given phase of the far field. The phase center can be calculated for a 3D pattern, part of a 3D pattern or a single radiation pattern cut (or merely of a segment of a single cut as defined by the user). Typically, the phase center is needed in the case where the antenna serves as an illuminator.

The feature calculates the phase center and recalculates the radiation pattern with respect to the phase center. The phase center is calculated for all three coordinates or only along one (x, y or z).

The use of the feature will be illustrated by using a rectangular horn antenna with a gain of approximately 10 dB. The phase of the radiation pattern can be displayed only for the Phi or the Theta component of gain (not for the total gain).

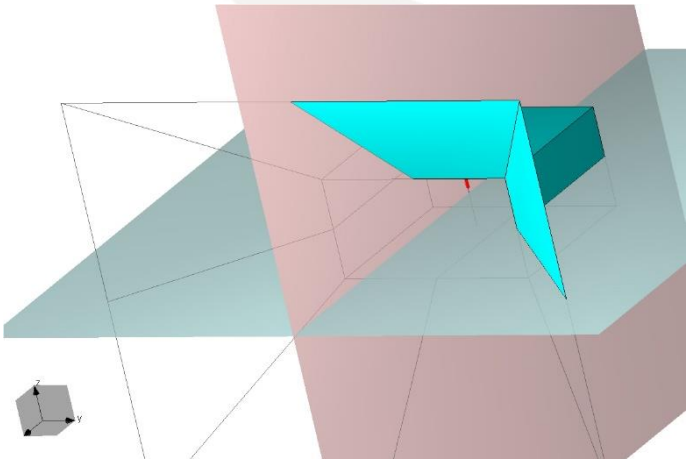


Fig. 12. Rectangular horn used for phase center calculations

The most important piece of the antenna radiation pattern is the part of space where the total gain is greater than -10 dB compared to the maximum gain. The radiation pattern of the horn antenna from Fig. 12 in that part of the space is presented in Fig. 13.

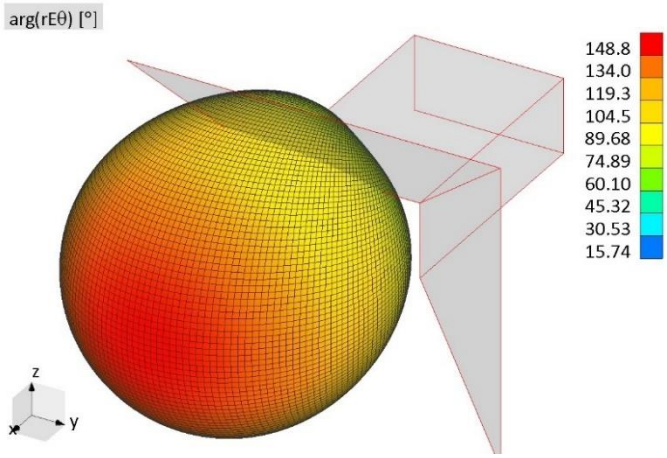


Fig. 13. Phase of theta component (with respect to the coordinate origin)

For the particular case of the horn antenna from Fig. 12, due to the symmetry, the phase center is located along the x-axis, and calculation is performed along the x-axis. The phase center is determined as x=64.2487 mm.

The phase of the theta component with respect to the calculated phase center is shown in Fig. 14.

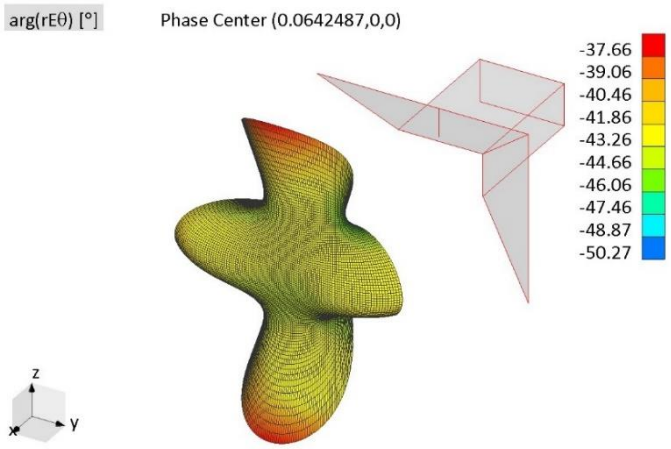
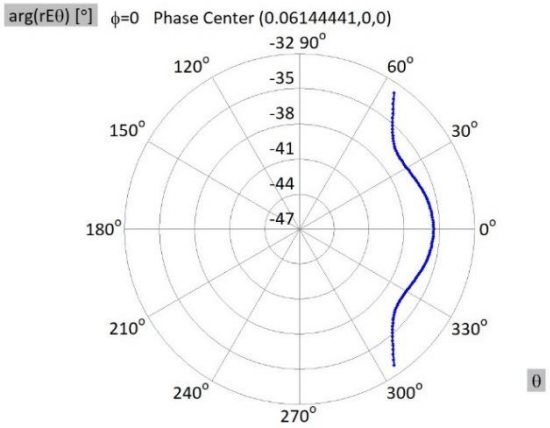
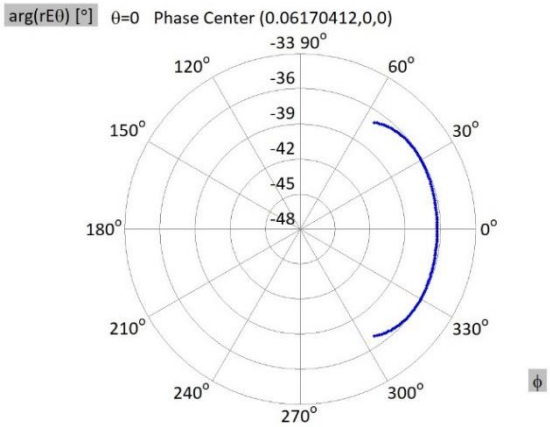


Fig. 14. Phase of theta component (with respect to the determined phase center)

The radiation pattern phase center can be calculated for any given cut. In this case, the two major cuts are Phi=0 degrees and Theta=0 degrees. Phase of theta field component for the two cuts, calculated for their specific phase centers (not identical for the 3D pattern and the two cuts) is shown in Fig. 15.



a) Phi=0 degrees cut



b) Theta = 0 degrees cut

Fig. 15. Re-calculated phase and the determined phase center for two major cuts

4. Enhancements in DDS (Domain Decomposition Solver)

Domain Decomposition Solver (DDS) is usually used for large problems (from approximately 0.5 million unknowns up to 5 million unknowns). The problems of such a large size are out of reach of direct GPU solution and are typically encountered when calculating scattering from electrically large objects or resolving antenna placement problems involving electrically large platforms.

After intensive testing of DDS it has been found that: a) hard drive disk space can be critical for large problems analyzed at multiple frequencies and with multiple excitations, b) CPU time needed for 20 excitations is doubled when compared to that needed for a single excitation, c) existence of resonant cavities (such as airplane inlets) can slow down the convergence a few times.

To maintain optimal performance even in the situations described above, the DDS algorithms are refined and new options are developed.

a) advanced memory management (resources reduced by an order of magnitude)

The memory required for the simulation is reduced by an order of magnitude. Before starting the simulation, the user can obtain the information whether there is enough free hard drive disk space or not for the intended number of frequencies, multiple excitations and maximum number of iterations. The user can also choose in the Configure menu to automatically delete all processing data after the simulation is finished with the given number of iterations, or to preserve them so that the accuracy of results can be improved by performing additional iterations. In the latter case, once the user decides not to perform additional iterations, all processing data can be removed using a specific command in the File menu.

b) efficient handling of multiple excitations

With the old algorithm, CPU time needed for 20 excitations was doubled when compared to that needed for a single excitation. Due to algorithm novelties, DDS can now handle 120 excitations requiring the same CPU time as previously required for 20 excitations when the old algorithm has been used. Usually, this number of excitations is more than enough to reconstruct monostatic RCS calculation in one plane using the option "Run as avoided with interpolation". Generally, the calculation of monostatic RCS is accelerated from 2-6 times.

c) user area groups (fast convergence for structures with cavities)

The user can manually specify a number of groups, in a user-friendly interface. Afterwards, the rest of the structure is grouped automatically. In this way, the user can encapsulate the cavities of the structure (e.g. inlets), or other parts of the structure of interest. Such encapsulation can reduce the number of iterations needed for good accuracy, especially if all encapsulated parts are strongly coupled, as in the case of resonant cavities.

As an example, let us consider a fighter 15.7 m long, with and without an inlet, as shown in Figs. 16 and 17. Total number of unknowns at 3 GHz is ~0.5 millions. Fig. 18 shows monostatic RCS results in the vertical plane. It is seen that the presence of an inlet dramatically changes RCS for directions between -60° to +60°.

In the case when a problem with an inlet is solved using regular grouping in DDS the convergence of the results is very slow, as shown in Fig. 19. It is seen that even after 7 iterations the results are still a few dBs below the accurate ones.

However, if the inlet is encapsulated by a single user defined group, the convergence is much faster, as can be seen in Fig. 20. Quite acceptable results are obtained even in the 0th iteration.

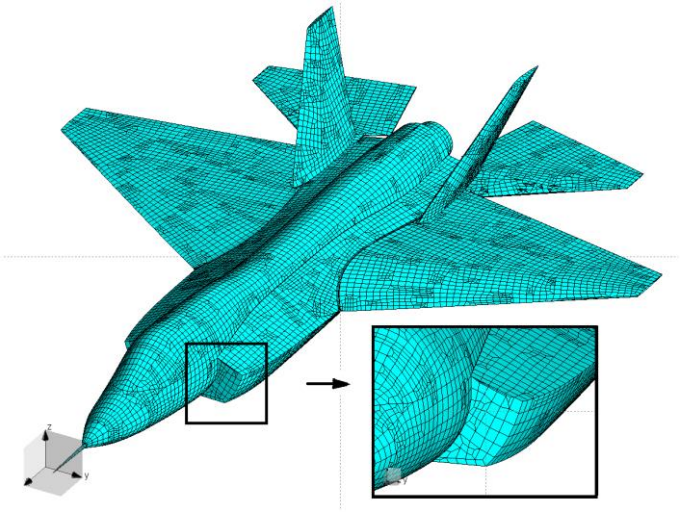


Fig. 16. Fighter modeled without an inlet

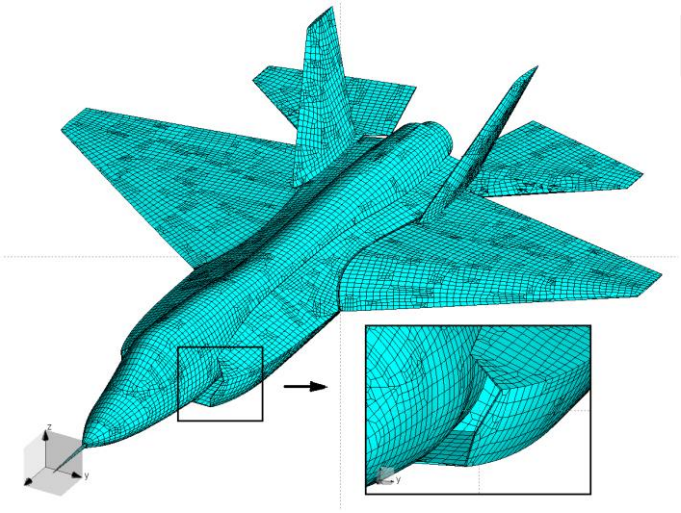


Fig. 17. Fighter modeled with an inlet

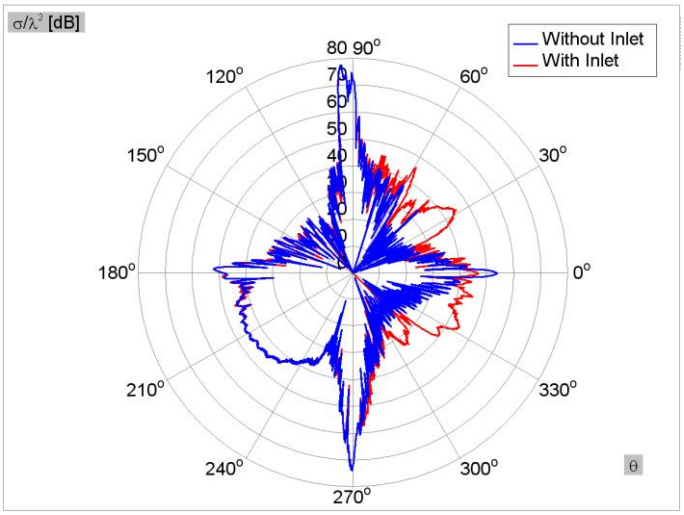


Fig. 18. Monostatic RCS at 3 GHz with and without an inlet

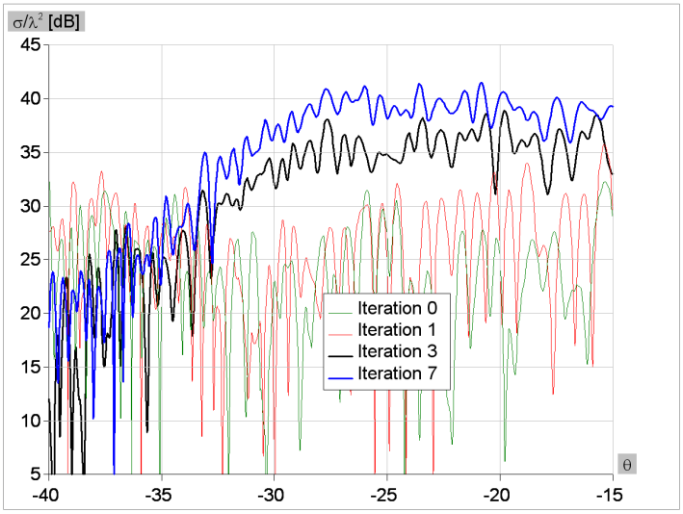


Fig. 19. Convergence of results using regular grouping

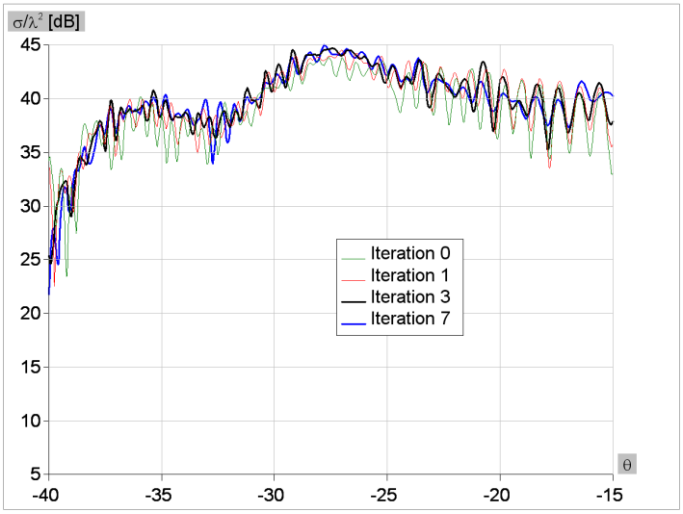


Fig. 20. Convergence of results using user defined groups for inlets

d) enabled usage of GPUs for sub-domain solutions

Originally, size of the groups and corresponding subdomain solutions was set by default to 3000 unknowns. However, in the case of "user area groups", this number can be much greater, up to 200 thousand unknowns as a part of a 5 million unknowns problem. In order to speed up the corresponding subdomain solution, the usage of GPUs is enabled. On the other hand, in the case of multi-core processors and a relatively small number of unknowns per group the GPU solution of a subdomain problem can be slower than the CPU solution. Having this in mind, it is up to the user to specify if GPUs should be used and to specify the limit for the number of unknowns. Depending on the speed of the CPU and the GPU solution the number of unknowns goes from 3000 to 15000.

e) comparison of results after each iteration

The results obtained by DDS after the 0th and the 1st iteration are very often acceptable from an engineering standpoint . Higher accuracy can be achieved by running a higher number of iterations. In particular, the user can choose in the Configure menu if the final results should be calculated after each iteration, which takes more time, or only after the final iteration. By inspecting results after each iteration, the user can obtain insight into the accuracy of results and can eventually decide whether to perform additional iterations to improve the accuracy.

5. Fast in-core solution for multiple excitation in a CPU solution

In a direct solution of multiple excitation problems (e.g., in calculation of monostatic RCS due to a number of incoming directions of a plane wave) the major part of simulation time is used for LU decomposition in the 1st step, while the unknown coefficients for each excitation are obtained by the forward and backward substitution in the 2nd step. In the previous version of WIPL-D Pro this 2nd step is performed sequentially (in series). However, parallelization of the 1st step performed in previous versions, and recent users' requests for an extremely high number of incoming directions, made the 2nd step become a bottleneck for RCS calculations. In this new version forward/backward substitution is performed in a block (in parallel), so that the 2nd step is accelerated by an order of magnitude.

6. Memory/Solution management

In the previous version of WIPL-D Pro, the unknown coefficients determined by simulation were stored in a DIS file in ASCII format in the case of a single frequency problem and could be reused for calculation of different sets of output results using the option Matrix inversion: Avoided. However, in the case of single frequency and many excitations (e.g. in monostatic RCS), such a file can occupy a huge amount of disk space. On the other hand, some users expressed interest to store the current coefficients in a range of frequencies.

Storing these coefficients as per users' preference and minimizing the storage requirement can be specified in the Configure menu. In particular, the user can choose to store the coefficients: a) always, b) for a single frequency, c) for a single excitation, d) for both, single frequency and single excitation, and e) never.

In addition, the user can choose the format: a) ASCII, b) binary (double), and c) binary (single). Binary (single) is the least space consuming.

7. Improved CMA Solver

Accuracy of eigenvalues and eigenvectors calculation is significantly increased in v15, resulting in correct characterization of a high number of characteristic modes, even those with very low modal significances. Beside this improvement, the mode tracking algorithm is additionally refined. This way, mode tracking is performed with high accuracy for structures with a large number of modes.

Folded dipole antenna, shown in Fig. 21, is analyzed in a wide frequency range, between 1 GHz and 5 GHz. Modal significance for the first 6 modes, in WIPL-D v14 and the new version (v15), are shown in Fig. 22.

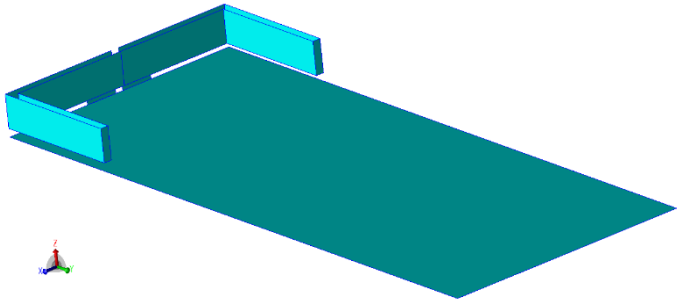
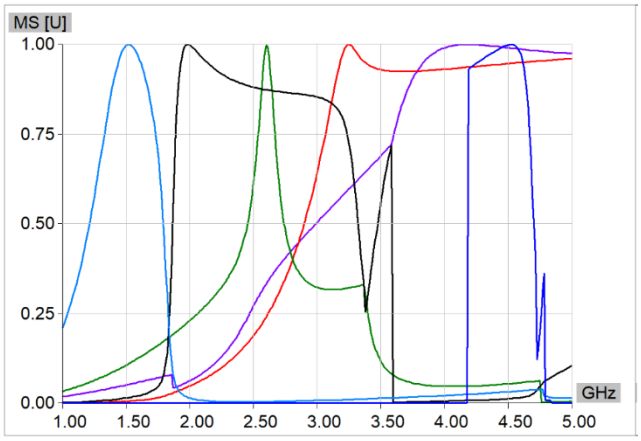
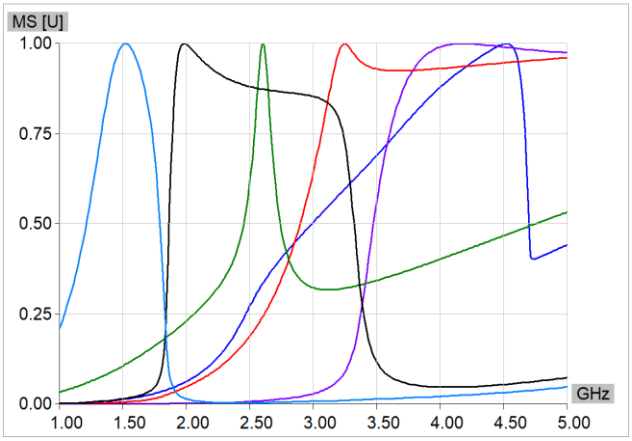


Fig. 21. Folded dipole.



a) WIPL-D Pro v14



b) WIPL-D Pro v15

Fig. 22. Modal significance for the first 6 modes.

8. GPU Cluster Solver

GPU Cluster Solver is significantly improved in v15. The most important new feature is a completely new **hardware resource management system**.

In the previous version, projects are simulated in the order which corresponds to the order of project submission. Only one project was simulated at a time, despite the number of used and the number of free cluster nodes being different. The new resource management system enables a parallel run of multiple projects, where each of them uses a number of nodes specified by a user. Projects are simulated in the order determined by specified priority. This way, a full utilization of available hardware resources is enabled, improving the efficiency of GPU Cluster simulations.

The licensing is also more flexible now. The cluster solution is limited by the total number of unknowns, which can be arbitrarily split into several simulations.

At any given moment, the number of unknowns available to run is determined as the difference between the total number of unknowns (of the cluster license) and the summed number of unknowns of all currently running simulations.