WIPL-D Pro: What is New in v20?

New features/improvements introduced in v20 are:

1. **WIPL-D scripting in Python:**
   a. **Open and run WIPL-D suites**
   b. **Access and modify symbols lists**
   c. **Obtain simulation results**
2. **Voxel Editor includes:**
   a. **Support for New voxel file formats**
   b. **Refined decimation for models of complex shapes**
3. **New object BoCS enables:**
   a. **Creation of objects by sweep of generatrix along composite 3D path**
4. **Medical Microwave Imaging (MMWI) Toolkit:**
   a. **Design of Helmet antenna system for tumor and stroke detection/monitoring**
5. **Current Generators grant:**
   a. **Rotations and translation of sources**
   b. **Usage of current sources in combination with symmetry planes (PEC, PMC, Half-Space)**
   c. **Max. number of current source elements exceeding 1 million**
   d. **Creating current generator files from IWP project**
6. **Domain Decomposition Solver:**
   a. **Quick evaluation of monostatic RCS (Example: For additional 360 directions CPU time equals that for bistatic RCS)**
7. **STL Editor provides:**
   a. **Refined meshing of extreme shapes (Example: Elongated brunches of treetop)**
   b. **10 times faster decimation and meshing**
8. **Improvements in WIPL-D Graph Viewer:**
   a. **Fast presentation of current distribution**
   b. **Extended limits for presentation of structures in conjunction with 3D graphs**
9. **Preview:**
   a. **Improvements in plate selection**
   b. **Improvements regarding Hide/Alt+L features**
   c. **Current Sources Threshold**
   d. **Improvements in displaying Waves**
10. **Improved multiprocessing for lower order basis functions**
11. **Other Features/Improvements:**
    a. **Improvements in IWP file import**
    b. **Improvements in handling projects with extremely large number of plates**
    c. **Working with Tables**
    d. **RAM Memory Limit**

1. WIPL-D scripting in Python

We are excited to announce the introduction of Python scripting in WIPL-D Pro. We prepared powerful, comprehensive, and easy-to-use scripting libraries that allow automation of various tasks. These libraries allow users to open and run WIPL-D suites, to access and modify symbols lists and to obtain simulation results.

The scripting language that has been used is Python 3.10. WIPL-D Python library is distributed as a wheel (.whl) file. It is a standard format installation of Python distributions and contains all the files and metadata required for the installation.

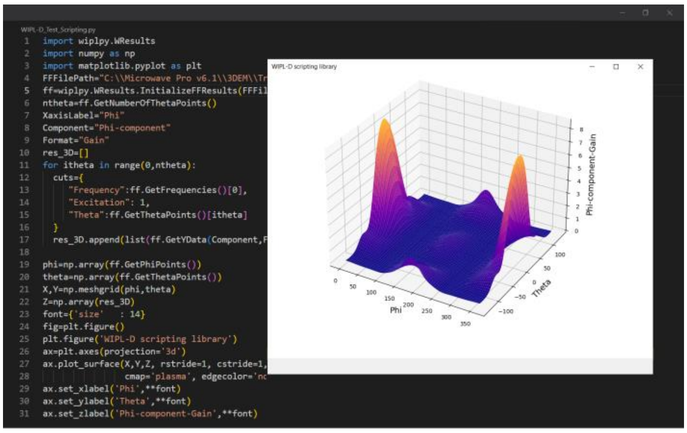


Fig. 1.1. WIPL-D Scripting in Python.

The WIPL-D scripting package provides three different modules: interface, symbols, and results.

a) Open and run WIPL-D suites

Simple operations such as opening and running the WIPL-D Pro (or alternatively WIPL-D Pro CAD) projects are accessible via *interface module*. In addition, this module provides access to project symbols.

The following script demonstrates how to open WIPL-D Pro suite and run training example “Intro_Task2” located in the installation directory.

```
import wiplpy.WiplInterface
WIPLDInstallDirectory="C:\Microwave Pro v6.2"
PathToProject="C:\\Microwave Pro
v6.2\\3DEM\\Training\\Models\\Intro
Task2\\Intro_Task2"
PRO=wiplpy.WiplInterface.InitializeWIPLDSuite(WIPLDIns
tallDirectory,"wipldpro")
PRO.Run(PathToProject)
```

b) Access and modify symbols lists

While the interface module itself has a built-in functionality of accessing and modifying the symbols list, the project symbols are also manageable via independent *symbols module*. Accessing the symbols via symbols module increases user's comfort since it does not require running instance of WIPL-D suite.

The following piece of code shows how to access the symbol named "A", print its current value, then change its value to 100, and print the new symbol value.

```
import wiplpy.WSymbols
PathToSMB="C:\\Microwave Pro
v6.2\\3DEM\\Training\\Models\\Intro
Task2\\Intro_Task2.smb"
SymbolsList=wiplpy.WSymbols.GetSymbols(PathToSMB)
Symb1=SymbolsList.GetSymbolByName("A")
print("The value of symbol A: ",Symb1)
SymbolsList.SetSymbolByName("A",100.0)
Symb1=SymbolsList.GetSymbolByName("A")
print("The new value of symbol A: ",Symb1)
```

c) Obtain simulation results

The *result module* enables access to WIPL-D output results. It allows one-dimensional access to results. Starting from one-dimensional results, obtaining two-dimensional results is just a step away, as it is supported by loops. No running instance of WIPL-D Suite is required for simulation results access.
The next script returns *s*11 parameters in dB at different frequency points, prints X-axis and Y-axis units and values.

```
import wiplpy.WResults
YZSFilePath="C:\\Microwave Pro
v6.2\\3DEM\\Training\\Models\\Eval Task6\\Slot-
Fed_Microstrip_Patch"
yzs=wiplpy.WResults.InitializeYZSResults(YZSFilePath)
cuts={"i":1,
      "j":1,
}
XaxisLabel="Frequency"
Component="Sparameter"
Format="MagdB"
resx=yzs.GetXData(XaxisLabel)
res=yzs.GetYData(Component,Format,XaxisLabel,cuts)
print("X-axis in: "+yzs.GetXUnit())
print(resx)
print("Y-axis in: "+yzs.GetYUnit())
print(res)
```

2. Voxel Editor

WIPL-D has upgraded this specialized program module intended to transform voxel data files into simulation ready electromagnetic models compatible with WIPL-D solvers.

The latest version of the program provides support for additional voxel file formats. At the same time, considerable effort has been put into refining the Decimation procedure, especially for models of complex shapes, to ensure better geometric representation and improved numerical efficiency.

a) Support for new voxel file formats

The functionality of the Decimation procedure has been expanded. From this software version, the mechanism also handles the voxel files of Mtype-Pvalue type. The results of the Decimation procedure, obtained for different decimation parameters applied to a single Mtype-Pvalue phantom, are demonstrated in Fig. 2.1.

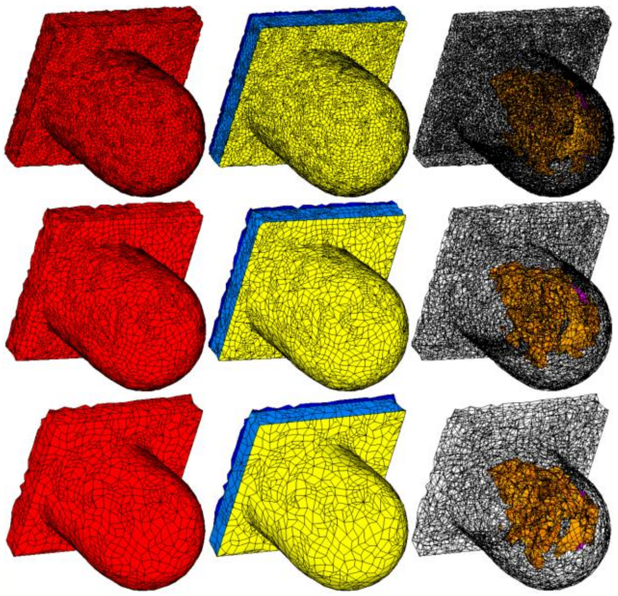


Fig. 2.1. Breast phantoms of various complexities produced for different decimation parameters applied to a single Mtype-Pvalue input file.

We have also adapted the Voxel Editor module so that it can process inputs in the form of NASA Snow Particle Phantoms. This phantom class is handling special type of files generated from NASA Goddard Space Flight Center (GSFC), i.e. NASA OpenSSP, representing complex-shaped snow particles. The appearance of the opening window for the NASA Snow Particle phantom input is displayed in Fig. 2.2.

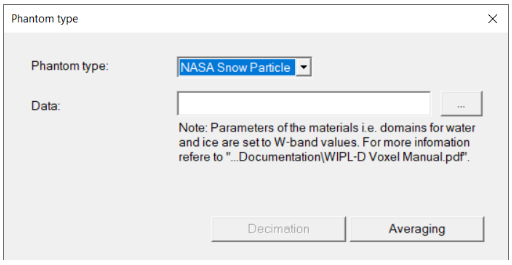


Fig. 2.2. NASA Snow Particle opening window.

b) Refined decimation for models of complex shapes

Several modifications have been made to the Decimation procedure algorithm. With these changes, for models of highly complex shapes, better quality factor outputs are obtained. It guarantees improved numerical efficiency during EM simulations.

Models of melting snowflakes of different decimation values (and thus the number of building elements) are presented in Fig. 2.3. It is visible that the model retains quality even with the drastic reduction in number of building elements (quadrilaterals).

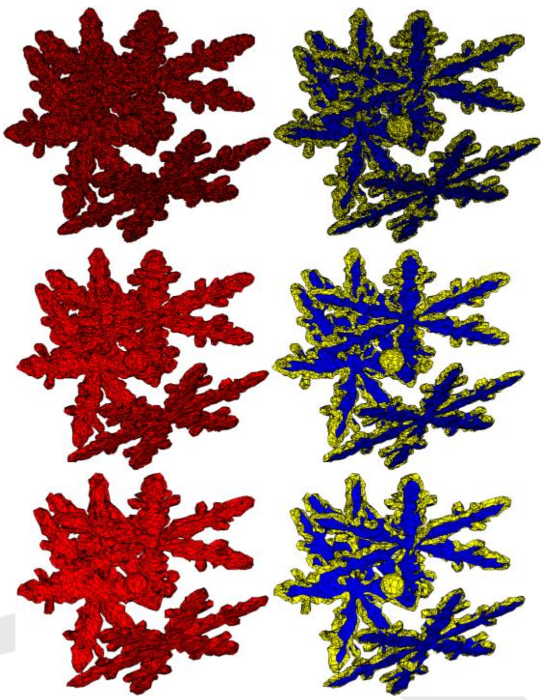


Fig. 2.3. Snowflake particles made from ice and melting water, as Decimation reduces the number of elements.

3. New object: BoCS

The new release brings improvement to the modelling process in WIPL-D Pro. Using the Object tool is the most important approach to create complex shapes. One of the widely used is body of constant cut (BoCC). It allows to sweep either a profile defined by arbitrary set of connected nodes or a generalized circle. The sweep path is a 2D profile. In v20, a new object is introduced to allow more complex sweeps and it’s called BoCS (BoCC in 3D space).

a) Creation of objects by sweep of generatrix along composite 3D path

BoCS object allows usage of arbitrary 3D paths. In its simplest form, the sweep path can be defined as a set of points, defined by x, y and z coordinate. This extends the abilities of BoCC to unlimited sweep profiles in 3D space. A simple example for a profile can be a square. Sweeping a square along a profile defined as set of 3D points is illustrated in Figure 3.1.

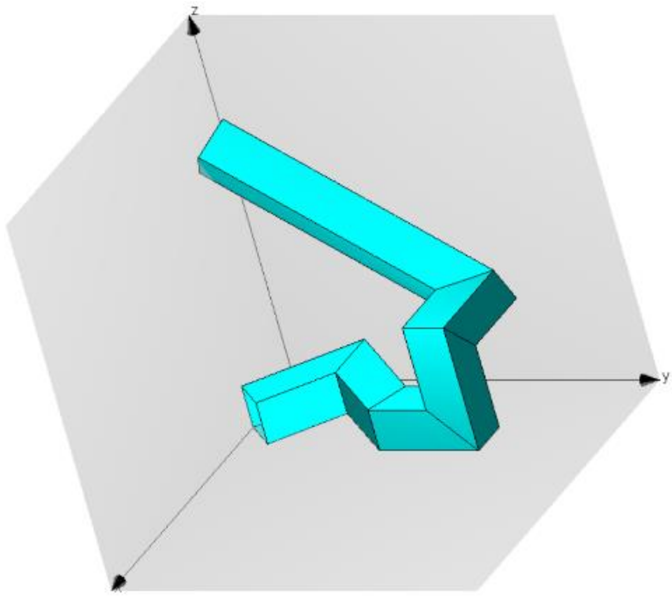
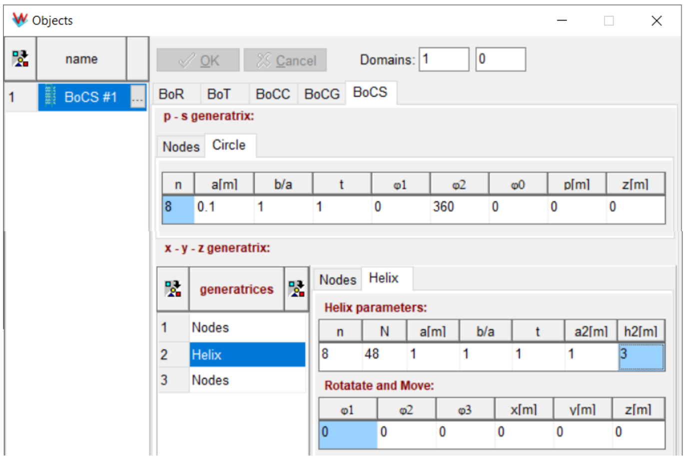


Fig. 3.1. BoCS object: Square swept along set of 3D points.

The new BoCS object offers additional possibilities regarding the sweep profile. It can be composite, defined by multiple sets of 3D points and multiple helix objects. The Nodes used as profiles are arranged sets of x, y, z coordinates, which are automatically connected to the last point of previous generatrix and to the first point of the following generatrix. Users can switch between defining generatrix as set of nodes, then a helix segment and then again nodes.

Each helix object can also be arbitrarily rotated and translated along x, y and z coordinates. This is particularly useful for spiral and helix objects. Typically, a wire coil is a helix object with the feeding lines at the start and end. Now BoCS allows to define the coil itself as helix, while the Nodes generatrix before and after is used to define the feeding line. The print-screen of GUI and the coil geometry is shown in Figure 3.2. The profile is a generalized circle with 8 segments. By changing this parameter only, users can obtain parametrized objects defined as cylinders with arbitrary number of segments (typically 4, 6, 8, 12…).



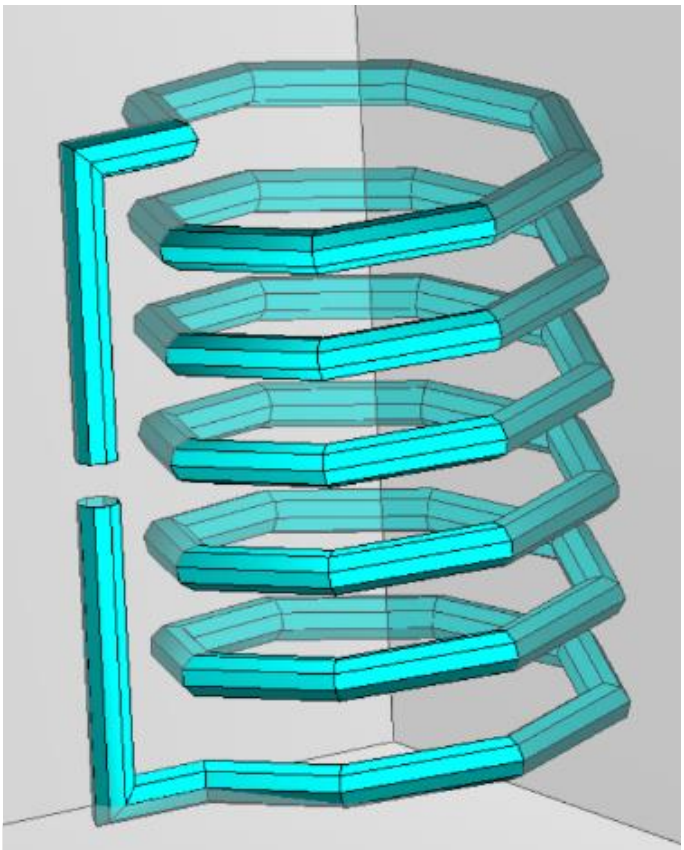


Fig. 3.2 BoCS object used to define coil with its feeding intro lines.

4. Medical Microwave Imaging (MMWI) Toolkit

In the new release of the software an advanced tool has been introduced – ***Helmet Designer*** to facilitate the design of an antenna array in the immediate vicinity of the human head phantom.

a) Design of Helmet antenna system for tumor and stroke detection/monitoring

In typical MMWI application the antenna array is positioned very closely to a human head, so for realistic modeling it is necessary to design the helmet to resemble the part of the head of interest as closely as possible. In our tool the helmet is built around the semi-ellipsoid geometry. Therefore, the best fit of the semi ellipsoid dimensions, i.e. main axes (Ahel, Bhel, Chel), must be found for accurate modeling. In addition, the definition of the translation vector (x0, y0, z0) to position the helmet on top of the head may be required.

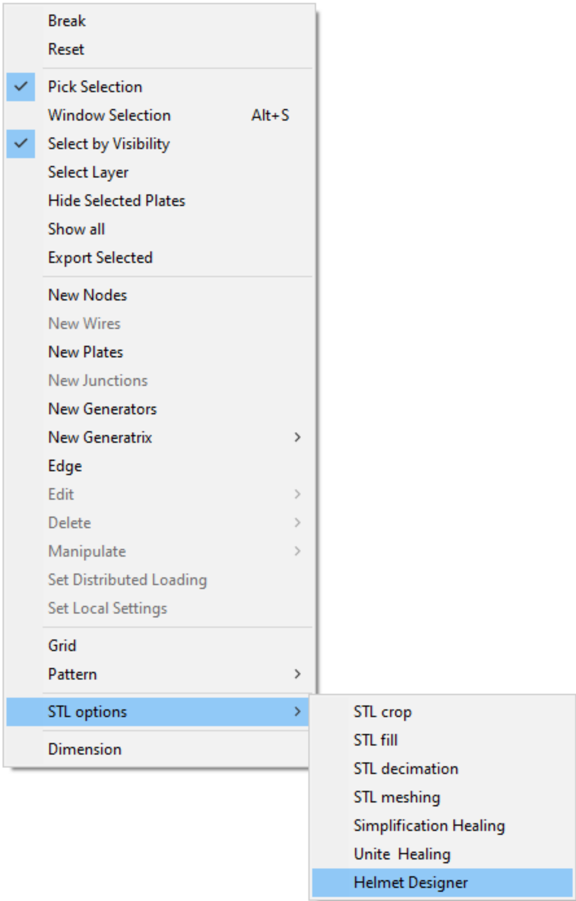


Fig. 4.1. How to start Helmet Designer.

The tool is activated by right click in the Preview Window and then choosing > STL options > Helmet Designer as presented Figure 4.1 This opens the Helmet Designer window presented in Figure 4.2 with predefined parameters. The parameters can be adjusted according to requirements of the particular helmet design such as semi-ellipsoid dimensions, number of antennas, description of antenna octagonal placeholder etc. as described in the corresponding manual.

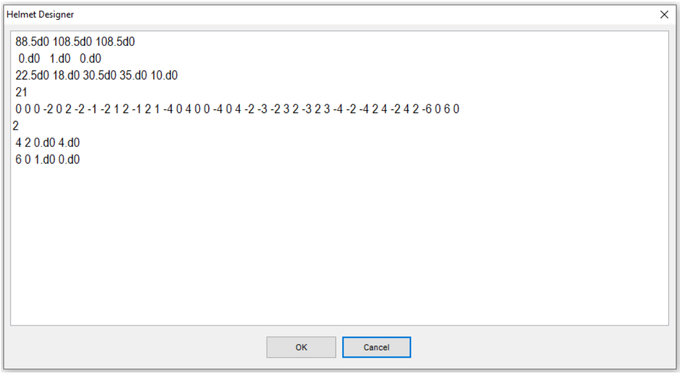


Fig. 4.2. Helmet Designer window with predefined parameters.

After the settings are confirmed by clicking OK, the Command Prompt window opens. As presented in Figure 4.3, the window lists the input parameters and offers 4 operation modes which can be selected by typing one of four numbers (0-3):

0 – basic positions (automatically determined by algorithm)

1 – corrected positions (manual adjustments)

2 – antennas pushed to the head (automatic antenna move towards the head, independent for each antenna, until the contact occurs either between an antenna and a head or between two antennas).

3 – A corresponding scenario project is created, and final rotation/translation parameters are available for inspection or editing.

Fig. 4.3. Command Prompt window where the operation mode is selected.

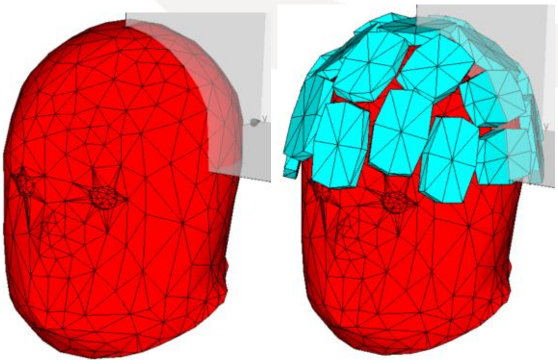


Fig. 4.4. The human head phantom and designed helmet with 21 antennas.

After the desired option is selected, the program executes the helmet design – an example of human head phantom and the helmet with 21 antenna array are presented in Figure 4.4.

If option 3 has been selected, the preview window shows the result (human head phantom surrounded by the antenna system) in the form of an STL project and offers the creation of a scenario. If the conversion is confirmed, a project scenario is created with the values for rotations and translations of the antennas written in the Symbols list. Being a scenario project, it allows the octagonal antenna placeholders created to be replaced by real antennas. The example of the antenna placeholders replaced with the realistic antenna is presented in Figure 4.5.

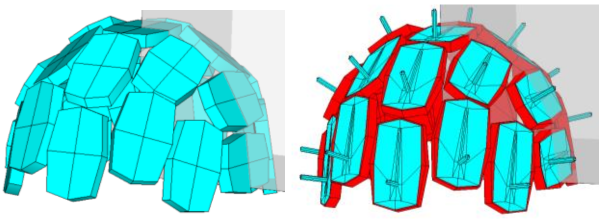


Fig. 4.5. The antenna placeholders replaced with realistic antennas – from Helmet designer to Scenario project.

5. Current Generators

Working with the current sources has been expanded to include several new features. The improvements will be demonstrated using a horn antenna. Currents induced by a horn antenna over an ellipsoid dielectric with electrical properties identical with vacuum are used as current generators. In Figure 5.1 the horn antenna surrounded by ellipsoid, corresponding radiation pattern diagram and current distribution over the ellipsoid are shown.

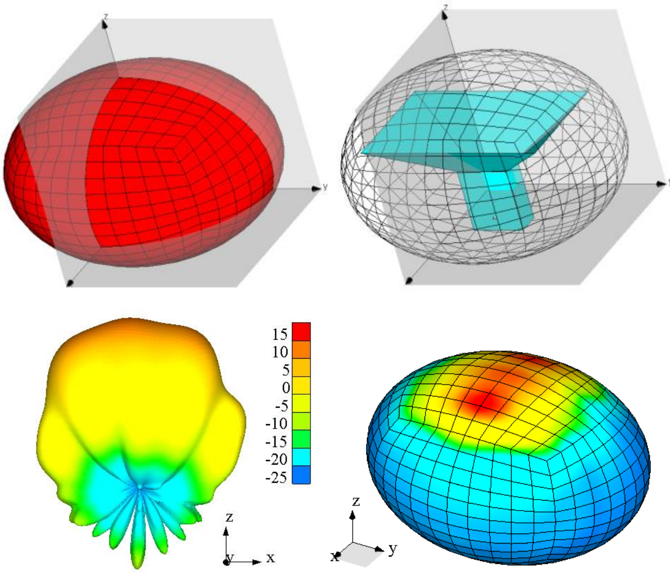


Fig. 5.1. The current sources derived from horn antenna surrounded by ellipsoid.

a) Rotations and translation of sources

At the import the current sources can be rotated and/or translated if required. The parameters specifying these operations can be entered at the corresponding fields, as presented in Figure 5.2.

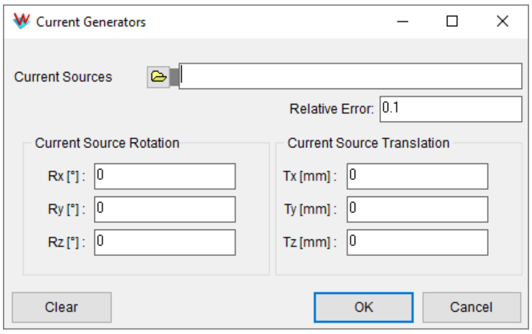


Fig. 5.2. Current generators import menu.

The effect of rotations and translations will be presented using a previously described horn antenna surrounded with the ellipsoid. The radiation pattern of the imported current sources without rotation/translation is displayed in Figure 5.3.

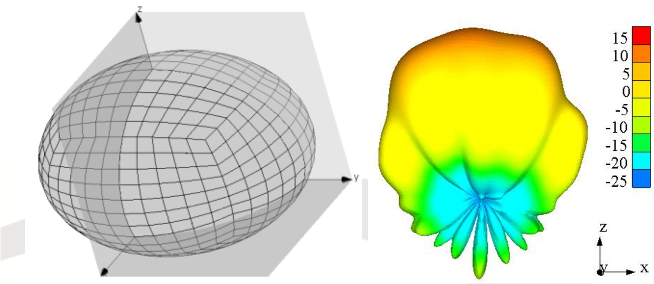


Fig. 5.3. Imported current sources without rotation/translation.

An example of how rotations/translations can be specified is presented in Figure 5.4 where rotations around x (-25°) and y (10°) axes are implemented. The sources are also translated along z axis (80 mm). The effect of preformed manipulations to radiation pattern is presented in Figure 5.5.

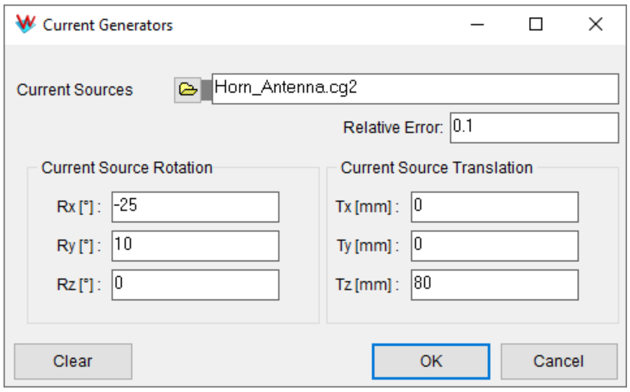


Fig. 5.4. Helmet Designer window with predefined parameters.

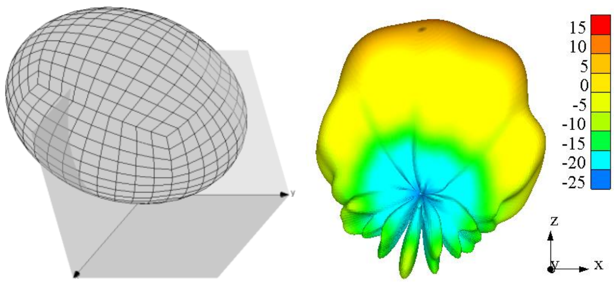


Fig. 5.5. The influence of rotation/translation of the sources to radiation pattern orientation.

b) Usage of current sources in combination with symmetry planes (PEC, PMC, Half-Space)

In the current software release current sources can be combined with a choice of ground planes (PEC, PMC, Half-Space). Using the same horn antenna as in the previous examples, the PEC ground plane is introduced to demonstrate the calculation of influence of the ground plane to radiation pattern. The PEC plane can be defined in the standard way, as illustrated in Figure 5.6. Alternatively, PEC can be defined using Green's function calculator. The same applies to PMC or Half-Space, as shown in Fig. 5.7.

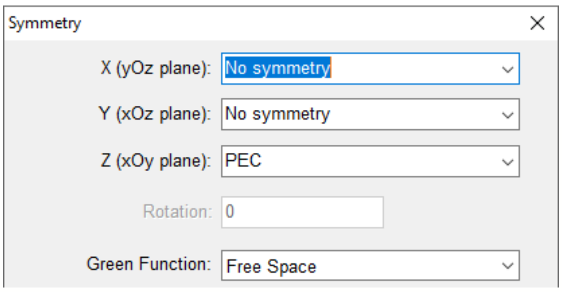


Fig. 5.6. Defining a PEC plane.

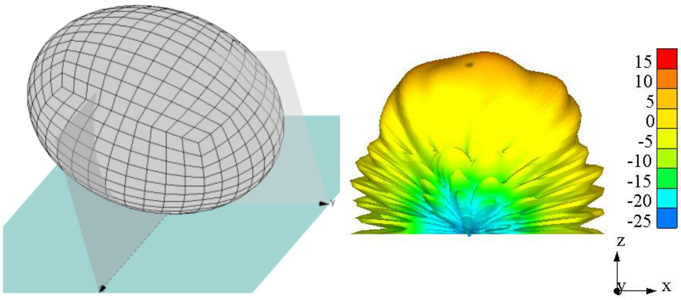


Fig. 5.7. The influence of the PEC plane on the horn antenna radiation pattern (rotation and translation as in the previous example).

c) Max. number of current source elements exceeding 1 million

The import and the algorithm for current source calculations have been improved to handle smoothly problems with the number of current source elements exceeding one million. Even twenty million elements have been processed successfully.

d) Creating current generator files from IWP project

Setting up the current sources has been made straightforward in the recent release as a single window can be used to set the parameters required. It is sufficient to select Density of points for current calculation and check the Create Current Sources field, as presented in Figure 5.8.

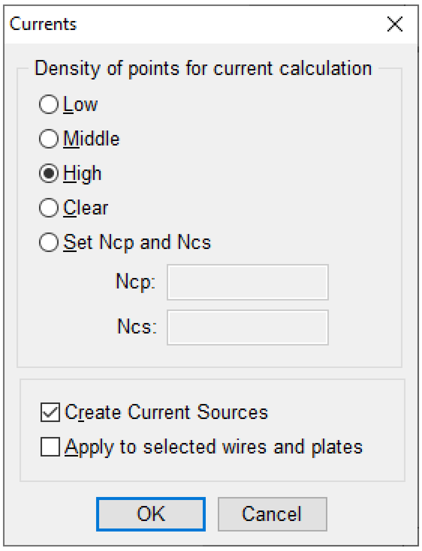


Fig. 5.8. Setting up the current source calculation.

6. Domain Decomposition Solver

Monostatic RCS simulations are accelerated. Example: Calculations of monostatic RCS for a single plane wave direction requires time T. Subsequent calculations for additional monostatic directions are considerably faster. The calculations of each another block comprising 360 new directions require the same time - T.

7. STL Editor

Algorithms for a) decimation of a triangular mesh, and b) their conversion into a quadrilateral mesh, are refined, thus enabling efficient handling of structures of extreme shapes (Example: Elongated brunches of treetop). Generally, these algorithms are accelerated around 10 times when compared with the previous version.

8. Improvements in WIPL-D Graph Viewer

A number of small improvements is done, among them the most important are a) accelerated presentation of current distribution, and b) extended limits for presentation of structures in conjunction with 3D graphs

9. Preview

Regarding the User Interface, several enhancements have been made to the Preview window.

a) Improvements in plate selection

Selection of multiple virtual plates (the plates created when predefined objects are used to create the model) is now running smoothly. Manipulations over such plate selections are also introduced.

b) Improvements regarding Hide/Alt+L features

Making a part of a structure invisible by using Alt+L feature now works properly even for models containing wires.

The hidden status of the plates is now canceled on every change in the number of plates.

c) Current Sources Threshold

For some applications the number of current sources can exceed millions of elements. In such cases the Preview image of the sources could become blurred or unclear and typically take a very long time to appear.

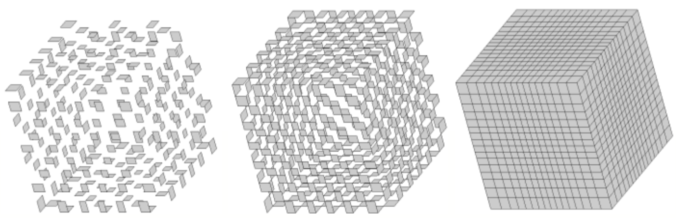


Fig. 9.1. Preview image of current sources for different current source threshold values.

For such cases the threshold limiting the number of sources displayed can be reduced by invoking the Current Sources Threshold option.

d) Improvements in displaying Waves

Improvements include displaying the waves in the Preview window with high fidelity – the waves now point towards coordinate system, are zoomed properly with the model and their graphical representation does not intersect model geometry.

10. Improved multiprocessing for lower order basis functions

Parallelization during Matrix fill-in phase is considerably improved. The improvements are especially noticeable for low basis function orders and large number of threads, as demonstrated for an average model in Figure 10.1.

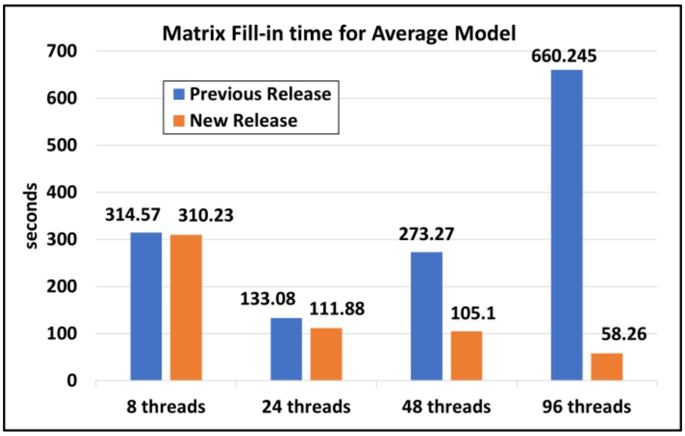


Fig. 10.1 Comparison of matrix fill-in times for an average model.

11.Other Features/Improvements

Several improvements have been included in the new version of the software to correct minor bugs and ensure smooth operation for better user experience.

a) Improvements in IWP file import

Importing native IWP format files is improved, especially for projects that use symbol definitions with frequency dependency.

b) Improvements in handling projects with extremely large number of plates

In the new release it is possible to handle projects with up to one million plates. However, to achieve this, the number of plates that are listed in the Plates table is limited to a value of 300.000. The limit is introduced to avoid memory issues. In case the project contains more plates, only the first 300.000 plates are listed in the table. At the same time, a total number of plates is stored inside the memory (including those not listed in the table), and all manipulations can be applied to the complete structure.

c) Working with Tables

Working with tables (plates, wires, domains, nodes etc.) like scrolling, multiple selection etc. has been significantly improved for user friendliness.

d) RAM Memory Limit

The previous numerical kernel limitation for the maximum RAM memory size of 2 TB has been removed.