## Contents

1 Introduction ............................................ 1-1
   1.1 Chapter Overview .................................. 1-2
   1.2 How to Read This Manual .......................... 1-3

2 Electromagnetic and Thermal Solvers ............... 2-1
   2.1 Finite-Difference Time-Domain Formulation .......... 2-1
   2.2 Conformal FDTD Formulation ........................ 2-3
   2.3 Alternating Direction Implicit Finite-Difference Time-Domain Formulation .... 2-4
   2.4 Dispersive Materials ................................... 2-4
   2.5 Metamaterials .......................................... 2-6
   2.6 Boundary Conditions .................................. 2-7
   2.7 Thin Resistive Sheets ................................. 2-9
   2.8 Metal Losses ......................................... 2-10
   2.9 Low Frequency Solver ................................ 2-10
   2.10 Thermal Solver ...................................... 2-12

3 Model Generation .................................. 3-1
   3.1 Parts of a Model .................................... 3-1
   3.2 Creating and Modifying Parts ....................... 3-1
   3.3 Entering Points into Modelling Dialogs .......... 3-9
   3.4 Grouping Parts ...................................... 3-10
   3.5 Interactive Modelling Tools ......................... 3-10
   3.6 Local Systems ....................................... 3-13
   3.7 Importing and Exporting ............................. 3-15
   3.8 Model History ...................................... 3-17
   3.9 Shortcut Keys ....................................... 3-18

4 Global Preferences .................................. 4-1
5 Electromagnetic Simulation

5.1 Simulation Settings ........................................... 5-1
5.2 Choosing the Appropriate EM Solver .......................... 5-3
5.3 Solid Region Settings .......................................... 5-6
5.4 Materials Database ........................................... 5-7
5.5 Source, Lumped Element and Sensor Settings ............... 5-8
5.6 Boundary Conditions .......................................... 5-11
5.7 Grid and Voxel Settings ....................................... 5-12
5.8 Running a Simulation ......................................... 5-12
5.9 Remote Simulation Tools ...................................... 5-13
5.10 Automatic Simulation Termination ............................ 5-17
5.11 Monitoring of Sensors and Manual Simulation Termination 5-17
5.12 Novice and Expert User Modes ............................... 5-18
5.13 Parallel iSolve Kernel ....................................... 5-18

6 Solids, Sources and Sensors

6.1 Solids .......................................................... 6-1
6.2 Sources ....................................................... 6-2
6.3 Sensors ........................................................ 6-7

7 Grid and Voxel Generation

7.1 Selecting a Grid Routine ...................................... 7-1
7.2 Interactive Grid Generator .................................... 7-2
7.3 Conventional Grid Generator .................................. 7-12
7.4 Voxelling Parameters ......................................... 7-14

8 Post Processor

8.1 Introduction .................................................... 8-1
8.2 Postprocessing Data Extraction ............................... 8-1
8.3 Simulation Results ............................................ 8-5
8.4 General Near-Field Extraction Options ....................... 8-13
8.5 SAR Extraction According to IEEE1529 ....................... 8-14
8.6 3-D Radiation Pattern Extraction ............................. 8-16
8.7 SEMCAD Postprocessing Viewer Types ....................... 8-19
8.8 Generating an Animated Image Series ......................... 8-31
8.9 Printout of Simulated Data and Results ..................... 8-32
8.10 File/Clipboard Export of Simulated Data and Results .... 8-37
COPYRIGHT

Copyright © 2002 - 2007 Schmid & Partner Engineering AG. All rights reserved.

This content of this document is provided by Schmid & Partner Engineering AG for informational purposes only, to licensed users of the SEMCAD-X software product and is furnished on an “AS IS” basis, that is, without any warranties, whatsoever, expressed or implied.

Information in this document is subject to change without notice. The software described in this document is furnished under a license agreement. The software may be used or copied only in accordance with the terms of this agreement. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, for any purpose other than the purchaser’s sole use, without the prior written permission of Schmid & Partner Engineering AG.

SEMCAD-X, iSAR, DASY and SPEAG are registered trademarks of Schmid & Partner Engineering AG.

Microsoft and Windows are registered trademarks of the Microsoft Corporation. Other brand and product names are trademarks or registered trademarks of their respective holders.

Schmid & Partner Engineering AG
Zeughausstrasse 43
8004 Zurich
Switzerland

Visit our Web site: http://www.semcad.com
Chapter 1

Introduction

SEMCAD-X is a 3-D full wave simulation environment based on the FDTD method, developed and provided by Schmid & Partner Engineering (SPEAG). The software is designed to address the electromagnetic TCAD needs of the wireless and medical sectors in terms of antenna design, EMC and dosimetry. SEMCAD-X provides standard RF solvers as well as Low Frequency solvers, the first commercial ADI-FDTD solver and solutions for coupled EM-thermal simulations. Furthermore, a range of specific method enhancements have been integrated. The 3-D solid modeler, being based on the ACIS modeling toolkit allows the rapid import and processing of various CAD formats and features a uniquely fast OGL based rendering engine. Moreover, a broad range of anatomical inhomogeneous high-resolution human and animal models are available involving grid-independent modeling. The postprocessor provides the extraction of any EM or SAR / temperature related result and its smooth and fast 3-D rendering within the GUI. Finally, the direct combination of postprocessors for SEMCAD-X and the DASY4 near-field scanners enable a direct comparison of simulated and measured data.

In addition, a multi-parameter multi-goal optimization engine based on Genetic Algorithms allows to optimize CAD imported and derived structures of any complexity. The newly introduced SPICE based Circuit Simulation Co-Engine performs full wave coupled co-simulation of EM and circuit analysis in time domain.

Since mid 2005, SEMCAD-X is interfaced with the newly introduced aXware Accelerator card and the aXware ClusterInABox, further reducing the runtime by at least 10 times. This dedicated hardware specifically optimized for FDTD and high bandwidth accelerator memory has proven to be more efficient and cost-effective than multi-processor workstations or multi-node clusters by decreasing initial and long-term maintenance costs. For more information see http://www.semcad.com.

In general, the FDTD method has successfully been applied to many different problems in electromagnetics such as scattering, radiation of antennas, optical applications, guided wave propagation and much more. Although this manual gives a brief overview of the FDTD method, it is not meant to replace a detailed introduction into its in-depth theory.
Note: As **SEMCAD-X light** is a lightweight version of **SEMCAD-X**, some of the features defined in this manual only apply to the full version. They have been left in this manual as a reference for the user, to see some of the capabilities that are available in the full version. This is especially the case for the Tools, which are available by default in the full version but can be fully programmed by the user by him/herself. Some of the major restrictions on the **light** versions include:

- Optimizer
- SPICE circuit co-simulator
- Thermal solver
- ELF solver
- aXware hardware acceleration
- reduced CAD capabilities, and
- reduced Tools

### 1.1 Chapter Overview

- Chapter 2 presents a short introduction into the theory of FDTD and ADI-FDTD methods. Their capabilities and restrictions are briefly discussed, and the features which are implemented into **SEMCAD-X** are introduced and explained in detail.

- Chapter 3 shows the usage of the modeling interface. The generation of geometrical solids is described, and the positioning of field-sensors and sources is explained.

- Chapter 4 outlines the range of preferences which the user can define for a number of settings applying to the different entities of **SEMCAD-X**, ranging from modeling view to the user access level.

- Chapter 5 gives a general overview of the simulation, how settings and parameters are defined and gives a general overview of the process flow ranging from model generation, through simulation, to the postprocessor.

- Chapter 6 gives a general overview of the principles of the geometry and material treatment, how to excite electromagnetic fields and how to extract parameters from the recorded data.

- Chapter 7 describes how to generate the mesh from the previously generated model. Considerations concerning the discretization of materials and sources are discussed, and the features of the mesh generator are explained.

- Chapter 8 describes the post processor capabilities of **SEMCAD-X**. The access to the simulation results is described, and methods for data visualization are outlined.

- Chapter 10 gives a detailed explanation of how to utilise the multiport S-parameter interface for the simulation of multiport filters, couplers, waveguides, etc.

- Chapter 11 describes the set of **Tools** which have been added as Python script extensions to the **SEMCAD-X** simulation platform, which gives the user further customization options and an environment where scripts with a user interface can be launched.

- Chapter 12 outlines the functionality of the **SEMCAD-X** Python based scripting environment.

The Appendix finally encloses additional information mainly related to Chapters 5 and 12.
1.2 How to Read This Manual

This manual is meant to give you a detailed overview of all aspects of SEMCAD-X. Chapter 2 gives you a brief introduction into the FDTD and ADI-FDTD methods. It does not replace a proper textbook on FDTD, so if you are completely new to the method you should refer to the literature.

In order to become familiar with the handling of SEMCAD-X you should read Chapter 5. It shows you how to set up a new project and describes the main elements of the user interface. Chapter 6 describes the way SEMCAD-X treats the materials of the geometrical solids in your model and how sources are placed to excite the fields, as well as which sensors to use to record which quantities.

Processing a project in SEMCAD-X consists of different steps: the modeling, the mesh generation and the actual FDTD simulation. Chapters 3, 7 and 5 describe how to go through these steps and settings concerning mesh generation and simulation; they are therefore strongly recommended reading.

Chapter 8 describes the features of the post processor. However, the handling of the post processor is very intuitive, so that you probably only have to refer to this chapter for detailed questions concerning the viewer types and additional statistical information provided by the post processor.

If you have questions beyond the manual, do not hesitate to send us an e-mail (semcad-support@speag.com). We are also glad to receive critiques or recommendations regarding the manual or the simulation environment which will help further improvements.

Schmid & Partner Engineering AG (SPEAG) is a spin-off company of the Swiss Federal Institute of Technology (ETH) in Zurich and continues to have many joint projects with the IT’IS Foundation. In the numerous publications of that group you can find a wealth of information related to the SEMCAD-X simulation environment. For more information see http://www.itis.ethz.ch.

Furthermore, an update list of publication related to SEMCAD-X can be found at http://www.semcad.com/simulation/publications/index_cover.php. The pages host abstracts and the papers can be downloaded.
1.2. How to Read This Manual
Chapter 2

Electromagnetic and Thermal Solvers

2.1 Finite-Difference Time-Domain Formulation

2.1.1 Discretization of Maxwell’s Equations

The Finite-Difference Time-Domain method (FDTD) proposed by Yee in 1966 is a direct solution of Maxwell’s curl equations in the time domain. The electric and magnetic field components are allocated in space on a staggered mesh of a Cartesian coordinate system (Figure 2.1). The E- and H-field components are updated in a leap-frog scheme using the finite-difference form of the curl which surrounds the component. The transient fields can be calculated when the initial field, boundary and source conditions are known.

Maxwell’s curl equations are discretized using a 2\textsuperscript{nd} order finite-difference approximation both in space and in time in an equidistantly spaced mesh. The first partial space and time derivatives lead to

\[
\frac{\partial F(i, j, k, n)}{\partial x} = \frac{F^n(i + 1/2, j, k) - F^n(i - 1/2, j, k)}{\Delta x} + O[(\Delta x)^2]
\]

(2.1)

\[
\frac{\partial F(i, j, k, n)}{\partial t} = \frac{F^{n+1/2}(i, j, k) - F^{n-1/2}(i, j, k)}{\Delta t} + O[(\Delta t)^2]
\]

(2.2)

with \(F^n\) as the electric (E) or magnetic (H) field at time \(n \cdot \Delta t\). \(i, j\) and \(k\) are the indices of the spatial lattice, and \(O[(\Delta x)^2]\) and \(O[(\Delta t)^2]\) are error terms.

Applying the central differences to Maxwell’s curl equations

\[
\nabla \times \mathbf{H} = \frac{\partial}{\partial t} \mathbf{e}_E + \sigma_E \mathbf{E} \tag{2.3}
\]

\[
\nabla \times \mathbf{E} = -\frac{\partial}{\partial t} \mu \mathbf{H} - \sigma_H \mathbf{H} \tag{2.4}
\]

with \(\sigma_E\) as the electric and \(\sigma_H\) as the magnetic losses for the proposed allocation of the fields in space and time leads, e.g., to Equation 2.3 for the \(E_x\) component.
2.1 Finite-Difference Time-Domain Formulation

Figure 2.1: 3D Yee cell showing the E- and H-field components in the staggered grid.

\[
\frac{E_x^{i,j,k}_{n+1} - E_x^{i,j,k}_{n}}{\Delta t} = \frac{1}{\epsilon_{i,j,k}} \left( \frac{H_z^{i,j+1/2}_{n+1} - H_z^{i,j-1/2}_{n}}{\Delta y} \right. \\
\left. - \frac{H_y^{i,j+1/2}_{n+1} - H_y^{i,j-1/2}_{n}}{\Delta z} - \sigma_{i,j,k} E_x^{i,j,k}_{n+1/2} \right)
\]  

(2.5)

Assuming the approximation

\[
E_x^{i,j,k}_{n+1/2} = \frac{E_x^{i,j,k}_{n+1} + E_x^{i,j,k}_{n}}{2},
\]  

(2.6)

Equation 2.5 can be reduced to the unknown \( E_x^{n+1} \) of the new time step, which yields

\[
E_x^{i,j,k}_{n+1} = \left( \frac{1 - \Delta t \sigma_{i,j,k}}{\Delta t \epsilon_{i,j,k}} \right) E_x^{i,j,k}_{n} + \\
\left( \frac{\Delta t}{\epsilon_{i,j,k}} \right) \left( \frac{H_z^{i,j+1/2}_{n+1} - H_z^{i,j-1/2}_{n}}{\Delta y} \right. \\
\left. - \frac{H_y^{i,j+1/2}_{n+1} - H_y^{i,j-1/2}_{n}}{\Delta z} \right).
\]  

(2.7)

Following this procedure, Maxwell’s curl equations can be derived and discretized to explicit expressions for all six field components.

2.1.2 Numerical Stability

For the explicit finite-difference scheme to yield a stable solution, the time step used for the updating must be limited according to the Courant–Friedrich–Levy (CFL) criterion. For the FDTD formulation...
of Maxwell’s equations on a staggered grid, this criterion reads

\[ \Delta t \leq \frac{1}{c \sqrt{ \frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2} + \frac{1}{(\Delta z)^2} } } \]

(2.8)

where \( \Delta x, \Delta y \) and \( \Delta z \) are the mesh steps of a Cartesian coordinate system and \( c \) the speed of light within the material of a cell.

From Equation 2.8 it is clear that the time step is directly related to the cell size. The cell size therefore has a significant impact on the computational requirements of a simulation. In an equidistantly spaced mesh, a reduction of the mesh step size by a factor of two will increase the necessary storage space by a factor of eight and the computation time by a factor of 16 (!). For non-uniform meshes, the impact of the smallest mesh cell on storage space is not as high. Nevertheless, the time step must be chosen for the smallest cell in the mesh and therefore affects the overall simulation time as well.

### 2.2 Conformal FDTD Formulation

As a powerful enhancement to reduce computational resources while maintaining its accuracy, SEMCAD-X features a Conformal FDTD Solver. The ability to use a conformal mesh with coarser spatial resolution than nonetheless produces the same accuracy as a fine staircasing mesh results in remarkable savings in memory requirements (fewer cells) and simulation time (larger time step).

Conformal FDTD is from its formulation similar to conventional FDTD but takes into account more geometrical details (see 2.2). Therefore, one cell can, e.g., represent more than one single dielectric material in contrast to the conventional FDTD algorithm. The error caused by the staircasing is subsequently largely reduced.

In SEMCAD-X, two algorithms of treating conformal FDTD have been integrated, depending on the involved material:

- **Dielectric model**: the area perpendicular to the edge is used to average the electromagnetic parameters. The method is always stable without any necessary reduction of the time step determined by the CFL stability criterion.

- **PEC (perfect electric conductor) model**: in this algorithm the effective PEC-free edge length and the PEC-free area are taken into account while time updating. In general, the improvement in accuracy is much more significant than for dielectric models. Furthermore, there is no need to reduce the conventional time step in order to guarantee numerical stability, i.e., the method is always stable. However the algorithm’s accuracy can be improved with a slightly reduced time step. Therefore the user can put a specific focus on either speed or accuracy. The time step reduction benefit can be dependent on the example.

In SEMCAD-X, both models are combined in the framework of the Conformal FDTD solver.

In case there is no PEC material involved, a CFL determined time step of 1 can be used, otherwise consider a slight reduction of the time step to 0.7 ... 0.8 its CFL value.

2.3 Alternating Direction Implicit Finite-Difference Time-Domain Formulation

The Alternating Direction Implicit Finite-Difference Time-Domain method for Maxwell’s equations (ADI-FDTD) differs only from FDTD by its time integration scheme. In the case of ADI-FDTD, the leap-frog time-integration scheme is replaced by an approximation of the Crank-Nicholson scheme applied to Yee discretization. A detailed description of the Alternating Direction Implicit Finite-Difference Time-Domain method used by SEMCAD-X is beyond the scope of this manual.

The ADI-FDTD method retains the unconditional stability and the global 2nd order accuracy of the Crank-Nicholson scheme while leading to lower computational costs, making it attractive for certain applications. This is the case for electrically over-discretized models, for instance due to the presence of very fine geometrical features or because of the need to operate at frequencies relatively low for the overall spatial discretization.

Please note that the error term grows with the time-step and at material interfaces.

One of the key issues when using ADI-FDTD is the choice of the time-step, which is no longer driven by the stability nor by the Nyquist limit but by the research of a compromise between an acceptable error term and a competitive calculation time compared to FDTD. Therefore the ADI-FDTD time-step in SEMCAD-X is specified as a factor Time Step factor of the CFL criteria for the FDTD simulation with the same spatial discretization.

The ADI-FDTD solver supports the same features as in the FDTD solver with the exception of the analytical absorbing boundary condition. That is due to the fact that the analytical absorbing boundary condition offers no significant speed-up compared to the UPML absorbing boundary condition for ADI-FDTD simulations.

Note: Conformal ADI-FDTD is now also supported, combining the advantages of C-FDTD and ADI-FDTD.

2.4 Dispersive Materials

Available dispersive material models are Debye, Lorentz, Drude, and the combination of Drude-Lorentz. These models can be assigned to any object in the project by using the Type drop down menu in material settings. The user can set $\varepsilon$, $\mu$ or both as dispersive through the What Is Dispersive drop down menu. The user can assign Dispersive material in the Solid Regions for the solids Type as shown in Figure 2.3.

The complex relative permittivities and permeabilities (replacing $\varepsilon$ by $\mu$) of the dispersive materials are defined for each method as follows:
Debye
\[ \tilde{\epsilon}(\omega) = \epsilon_\infty + \sum_{p=1}^{P} \frac{(\epsilon_0 - \epsilon_\infty) \cdot A_p}{1 + j \cdot \omega \cdot \tau_p} \]  \hspace{1cm} (2.9)

Lorentz
\[ \tilde{\epsilon}(\omega) = \epsilon_\infty + \sum_{p=1}^{P} \frac{(\epsilon_0 - \epsilon_\infty) \cdot A_p \cdot \omega_p^2}{\omega_p^2 + 2 \cdot j \cdot \omega \cdot \tau_p - \omega^2} \]  \hspace{1cm} (2.10)

Drude
\[ \tilde{\epsilon}(\omega) = \epsilon_\infty - \sum_{p=1}^{P} \frac{\omega_p^2}{\omega^2 - j \cdot \omega \cdot \tau_p} \]  \hspace{1cm} (2.11)

Drude-Lorentz
\[ \tilde{\epsilon}(\omega) = \epsilon_\infty + \sum_{p=1}^{P} \frac{(\epsilon_0 - \epsilon_\infty) \cdot A_p \cdot \omega_p^2}{\omega_p^2 + 2 \cdot j \cdot \omega \cdot \tau_p - \omega^2} - \sum_{p=1}^{P} \frac{\omega_p^2}{\omega^2 - j \cdot \omega \cdot \tau_p} \]  \hspace{1cm} (2.12)

where \( \tilde{\epsilon} \) is the complex permittivity, \( \omega = 2\pi f \) denotes the angular frequency, \( \epsilon_0 \) denotes the static permittivity, \( \epsilon_\infty \) is the permittivity at infinity, \( P \) is the number of poles, \( A_p \) is the amplitude of pole \( p \), \( \omega_p \) is the \( p \)th pole angular frequency, and \( \tau_p \) is a relaxation time of pole \( p \). The parameters Dispersive Electric Conductivity and Dispersive Magnetic Conductivity are used additively offset the conductivity curves.

**Note:** Dispersive materials are treated in the same way as Dielectric in terms of gridding and voxeling.

**Note:** Decreasing the Time Step Factor increases accuracy and stability but also simulation time: The Drude and Lorentz algorithms are accurate if \( \omega_p \Delta t \sim 0.1 \). The stability is ensured if:

Debye
\[ \text{the CFL-criterion (2.8) is satisfied,} \]

Lorentz
\[ \omega_p \Delta t \leq 2 \sqrt{\frac{\epsilon_\infty}{\epsilon_0}}, \]  \hspace{1cm} (2.13)

Drude
\[ \omega_p \Delta t \leq 2 \sqrt{\epsilon_\infty}, \]  \hspace{1cm} (2.14)

Drude-Lorentz
\[ \text{the more restrictive of the previous criteria is satisfied.} \]

**Tip:** To plot the \( \epsilon_r(f) \), electric \( \sigma(f) \), \( \mu_r(f) \) and magnetic \( \sigma(f) \) for Dispersive materials use the Dispersive Material Tool under Tools/Calculators (see Section 11.51).
2.5 Metamaterials

Metamaterials are supported in SEMCAD-X only using the software kernel. Currently, the permittivity and the permeability of metamaterials are described by a Drude-Lorentz model. The difference between metamaterials and dispersive materials is that it offers an option to activate nonlinear effects. This option enables the user to characterize third-order susceptibility $\chi^3$ effects such as the Kerr-effect and Raman-scattering. While the Kerr-effect adds a linear intensity dependence to the relative permittivity or permeability, Raman-scattering describes the scattering of light at phonons (lattice vibrations). The Kerr-effect is available for permeability and permittivity, however, Raman-scattering is effective only on the permittivity. The user can set the type of the desired solid object to Metamaterial using Type drop down menu under Solid Regions|Material settings.

The complex relative permittivities and permeabilities (replacing $\epsilon$ by $\mu$) of the metamaterials are defined as follows (without the Drude-Lorentz model):

**Kerr**

$$\tilde{\epsilon}(\omega) = \epsilon_\infty + \chi_3^p \cdot |E|^2$$  \hspace{1cm} (2.15)

**Raman**

$$\tilde{\epsilon}(\omega) = \epsilon_\infty + \sum_{r=1}^{R} \frac{\chi_3^r \omega_r^2}{\omega^2 + \frac{2 \omega \omega_r}{\tau_r} - \omega_r^2}$$  \hspace{1cm} (2.16)

where $\tilde{\epsilon}$ is the complex permittivity, $\omega = 2\pi f$ denotes the angular frequency, $\epsilon_\infty$ is the permittivity at infinity, $R$ is the number of phonons, $\chi_3^r$ is the amplitude phonon $r$, $\omega_r$ is the angular frequency of phonon $r$, and $\tau_r$ is the lifetime of phonon $r$. The time dependence of the polarization from Raman-scattering is then described by a convolution

$$P_{\text{Raman}}(t) = \epsilon_0 \cdot E(t) \cdot [\chi_3^{\text{Raman}}(t) * |E(t)|^2]$$  \hspace{1cm} (2.17)

**Note:** The nonlinear algorithm used for metamaterials is only stable if

$$|E|^2 < 0.9 \cdot \frac{\epsilon_\infty}{\chi^3}.  \hspace{1cm} (2.18)$$

Ensure that the field amplitudes near metamaterials are sufficiently below this value.
2.6 Boundary Conditions

**SEMCAD-X** provides three types of boundary conditions to truncate the computational domain: absorbing (ABC), perfectly conducting (PEC), perfectly magnetic (PMC) and Periodic. These types of boundary conditions can be arbitrarily combined for the six sides of the mesh. They are described in the following sections.

2.6.1 Absorbing Boundary Conditions

In the absence of absorbing or open boundary conditions (ABC), waves impinging on the outer boundary of the grid would be reflected back into the computational domain. Therefore, the mesh must be truncated with absorbing boundary conditions which either absorb incoming waves without reflection (UPML) or simulate a transparent boundary condition (analytical).

The ABC types **SEMCAD-X** offers are listed below:

1. Uniaxial Perfectly Matched Layers (UPML)

   UPML acts like an absorber in an anechoic chamber. It extends the computational domain with layers of perfectly matched absorbing material so that the incoming waves propagate without reflection while they are attenuated by successive lossy layers. When the waves reach the last layer, their amplitude is reduced by orders of magnitude and they are reflected back towards the computational domain.

   UPML was introduced by Stephen Gedney, and is closely related to the PML ABC proposed by Jean-Pierre Berenger. Both have the same absorbing efficiency; however, UPML can be implemented more efficiently than PML with respect to speed and memory. UPML ABC is recommended for all applications and has been highly optimized in **SEMCAD-X**. The truncation of inhomogeneous and lossy media is supported. This is very useful for numerical dosimetry since the non exposed side of the numerical phantoms can be truncated leading to smaller grids. Please note that the grid and the voxel/edge discretization are automatically extended normally to the boundary into the UPML.

   The efficiency of UPML can be controlled by its thickness and its absorbing profile. It provides absorption for all incidence angles and can be positioned very close to radiating structures (as long as it does not change the behavior of the radiating structures).

   The efficiency of UPML depends greatly on its thickness and its conductivity and power loss profile. These parameters are based on modified geometrical progressions and are calculated automatically by **SEMCAD-X** according to the level of strength (low, medium, high, very high) required by the user.

   When ADI-FDTD is used for frequencies relatively low for the overall spatial discretization, the UPML parameters are automatically adjusted too reflect the changing nature of the physical phenomena.
2.7. Thin Resistive Sheets

The UPML parameters can also be user-defined; in this case, the number of layers and the conductivity and power loss profiles can be controlled using a typical geometrical progression which has been proven to be preferable to the polynomial profile because it allows better control of the growth rates.

Please note that user-defined UPML profiles can adversely affect the stability of ADI-FDTD simulations.

2. Analytical Absorbing Boundary Condition

Theoretically, the transparent boundary condition is a non-local operator, since for each point on the boundary the values therein relate with the values at all points on the boundary. Therefore, substituting the transparent boundary condition by a local boundary condition, such as the analytical absorbing boundary condition (A-ABC), is actually making a high frequency approximation. In other words:

- the extent of the boundary must be large compared to the wavelength,
- the boundary must be put far away from large objects, and
- the efficiency is sensitive to the incident angle of the incoming wave.

The A-ABC implemented in SEMCAD-X is related to the one-way wave equation ABC. Therefore, it provides only total absorption for the normal incidence and is not recommended for the truncation of inhomogeneous and/or lossy media. The A-ABC is fast and does not require extension of the grid. However, it is not as efficient or flexible as the U-PML ABC described in the previous section. Its efficiency is roughly equivalent to the $2^n$th order Mur/Higdon implemented in previous versions, and has been proven to be satisfactory for many applications.

Please note that the analytical absorbing boundary condition cannot be selected for ADI-FDTD simulations.

2.6.2 Perfectly Conductive Boundary Condition

Perfectly conductive boundary conditions truncate the computational domain with perfectly conducting planes (PEC). The tangential components of the E-fields on the outer boundaries are set to zero.

2.6.3 Perfectly Magnetic Boundary Conditions

Perfectly magnetic boundary conditions truncate the computational domain with perfectly magnetic planes (PMC). The tangential components of the H-fields at half a cell from the outer boundaries are set to zero.

2.6.4 Periodic Boundary Conditions

Periodic boundary conditions can be used for structures with geometry exhibiting periodic characteristics. By using the periodic boundary conditions to terminate the computational domain it is only necessary to model and simulate a single period of the geometry. The geometry in the computational domain will be mirrored at the periodic boundary to represent the infinitely periodic structure.
2.7 Thin Resistive Sheets

2.7.1 Introduction

In standard FDTD, every solid region which is relevant for the simulation needs to be resolved in space (according to geometric dimensions and/or wavelength inside the material). Because a small spatial discretization results in a small time step, common FDTD simulations suffer from a limit imposed on the possible geometries and/or materials that can be simulated efficiently.

For the subgroup of thin resistive sheets (TRS) (i.e., electrically thin structures with significant conductivity) SEMCAD-X offers an algorithm that models their influence without the need to resolve the sheet thickness with the grid. The algorithm supports arbitrarily tilted and curved shapes.

2.7.2 Applicability

The TRS algorithm is very well suited for simulating structures that contain metallic sheets, which are either electrically thin or have finite conductivity, i.e., a percentage of an incident field will be transmitted through the sheet and the remainder will be reflected.

Tip: The Tools | Calculators | Thin Resistive Sheet Calculator can be used to assess if it is necessary to use TRS. Enter the relevant frequency, electric conductivity and sheet thickness to calculate the reflected and transmitted component (see also Section 11.53).

2.7.3 Usage

Set the type of the desired Surface or Solid object to PEC/Metal and check the Thin Resistive Sheet checkbox. Specify the material parameters. Note that for thin resistive sheets a new parameter called Sheet Thickness is available defining the thickness in micrometers. You can also transfer the parameters directly from the TRS calculation tool.

Note: The Region Characteristics property is hidden for TRS because they are always treated as Surface objects. Thus, for a volumes, only the surface is voxelled as a TRS with the specified thickness.
2.8 Metal Losses

The simulation may then be gridded, voxelled and run as described in the following sections. The simulation log will show the progress and output of the TRS algorithm. The computed reflection and transmission coefficients in the log show the behavior of each thin sheet if it was surrounded by free space in order to give an estimate of the TRS effect. For very high or very low reflection a sheet may behave like PEC or free space respectively.

Important points when using the TRS algorithm are:

- Because the TRS algorithm is assigning special update coefficients (and thus a new material) to almost every edge of the discretized sheet surface, it may be necessary to enable support for large material tables in File — Preferences — Solver Executables.

- When creating the grid, make sure that highly curved sheets are discretized with a sufficiently fine grid step of at least a few grid steps per curvature radius.

- Make sure that between two sheets or between the surfaces of a single folded sheet there is at least one grid line.

- The TRS algorithm is compatible to all solvers except the conformal variants (FIT/C-FDTD; FIT/C-ADI-FDTD).

2.8 Metal Losses

SEMCA-D-X incorporates the ability to add surface impedances to metallic solids. This is especially important in the modeling of nm scale problems.

An impedance can be added to solids in harmonic and broadband excitations and the user defines the surface impedance in terms of Electrical Conductivity, Magnetic Conductivity, Permittivity and Permeability. The surface impedance is defined by the equation:

\[ Z_s(\omega) = (1 + j)\sqrt{\frac{\omega\mu}{2\sigma}} \quad (2.19) \]

and,

\[ \sigma >> \omega\epsilon \]

2.9 Low Frequency Solver

2.9.1 Introduction

In section 2.1.1 Maxwell’s curl equations were introduced in time domain. They describe the physics of electromagnetic field propagation. The equations can be transformed into frequency domain by assuming a harmonic oscillation \( e^{j\omega t} \) where \( j \) is the complex number with \( j^2 = -1 \), \( \omega \) is the angular frequency \( \omega = 2\pi f \), and \( t \) denotes the time. In frequency domain Maxwell’s curl equations read

\[ \nabla \times \mathbf{E} = -j\omega \mathbf{B} = -j\omega\mu \mathbf{H} \quad (2.20) \]

\[ \nabla \times \mathbf{H} = j\omega \mathbf{D} + \mathbf{J} = j\omega\epsilon \mathbf{E} + \sigma \mathbf{E} + \mathbf{J}_{ext} \quad (2.21) \]

where the fields \( \mathbf{E} \), etc., are complex valued vectors (phasors).

As stated in the previous sections, the FDTD method efficiently solves problems of the size of several wavelengths and spatial resolutions down several ten thousandth part of a wavelength. However, while decreasing the frequency the wavelength is increasing and the FDTD method becomes more...
and more inefficient due to the explicit time integration. Fortunately, simultaneously some terms in equations (2.20) and (2.21) become negligible depending on the simulation settings and the frequency. Therefore, Maxwell’s curl equations can be simplified and, hence, are easier to solve.

All approximations share the characteristic, that the wavelength is much larger than the computational domain. Therefore, all fields are considered as ‘instantaneous’ and do not support any kind of propagation, i.e., have no wave characteristic. In formula, this first approximation reads

$$\lambda \gg \text{diag}(\Omega)$$  \hspace{1cm} (2.22)

where $\lambda$ is the wavelength and $\text{diag}(\Omega)$ denotes the diameter of the computational domain.

**2.9.2 Different Models**

In general, the models can be cataloged into two major categories: electro and magneto quasi-static approximation. The next subsections explain the details.

**Electro Quasi-Static**

The electro quasi-static approximation neglects the temporal change of the magnetic flux $\mathbf{B}$, i.e., equation (2.20) becomes

$$\nabla \times \mathbf{E} = 0.$$  \hspace{1cm} (2.23)

If the computational domain $\Omega$ is one-connected, then equation (2.23) implies that the electric field $\mathbf{E}$ is a potential field $\mathbf{E} = \nabla \phi$, where $\phi$ is a scalar potential function. Therefore, equation (2.21) can be rewritten as

$$\nabla \cdot \tilde{\epsilon} \nabla \phi = 0, \quad \tilde{\epsilon} := \epsilon_0 \epsilon_r - j\frac{\sigma}{\omega}$$  \hspace{1cm} (2.24)

where $\tilde{\epsilon}$ is the complex valued permittivity. Because the permittivity is complex valued, the potential $\phi$ is complex valued, as well.

The approximation is only valid for induced current free simulations (no Eddy currents). A rule of thumb is to compare the skin depth $\delta$ of a lossy dielectric with the size of the computational domain, i.e.,

$$\delta \gg \text{diag}(\Omega).$$  \hspace{1cm} (2.25)

A further simplification can be done, if one of the following conditions are satisfied for all dielectrics in the simulation settings:

$$\omega \epsilon \ll \sigma E : \quad \nabla \cdot \sigma E \nabla \phi = 0, \quad \text{Ohmic current dominated}$$ \hspace{1cm} (2.26)

$$\omega \epsilon \gg \sigma E : \quad \nabla \cdot \epsilon \nabla \phi = 0, \quad \text{electro static}$$ \hspace{1cm} (2.27)

With either approximation the potential $\phi$ became real valued.

All three described models are already in the **SEMCAD-X** V13.0 Bernina Release.

**Magneto Quasi-Static**

If the induced (Eddy) currents are not negligible, the magneto quasi-static approximation may provide a good model for further investigations. The temporal change of the displacement current $j \omega \mathbf{D}$ has to be neglected, i.e.,

$$\omega \epsilon \ll \sigma E$$  \hspace{1cm} (2.28)
for all dielectrics in the model. Then, Maxwell’s curl equations reduce to
\[
\nabla \times \mathbf{E} = -j\omega \mu \mathbf{H} \quad \text{(2.29)}
\]
\[
\nabla \times \mathbf{H} = \sigma E + \mathbf{J}_{\text{ext}}. \quad \text{(2.30)}
\]
The solution of the magneto quasi-static model are complex valued phasors \( \mathbf{E} \) and \( \mathbf{H} \).

This model is not yet available in the SEMCAD-X V13.0 Bernina Release.

### 2.10 Thermal Solver

#### 2.10.1 Introduction to the thermal solver

One of SEMCAD-X’s strengths is its suitability to determine fields induced in living tissue by EM sources. One of the many electromagnetic field induced effects is a tangible temperature elevation, so modeling the resulting thermal distribution in solids can be crucial. Because of this, a thermal solver has been integrated into the SEMCAD-X platform. This solver is tuned to perform optimally for BioEM problems, but attention has been paid during the design to keep it suitable for a broader range of applications.

Being able to simulate the temperature effects has various benefits:

- It is cheaper and easier to perform the experimental analysis
- It gives a more detailed image than measurements would
- The designer has more control over relevant parameters
- It is easier to decouple various factors (as in sensitivity studies, for example)
- It offers a good method to optimize antennas and other parameters

The goal of this solver is to get a high degree of reliability and a fine resolution, while executing the individual calculations quickly and to deliver the results within a reasonable timespan.

#### 2.10.2 SEMCAD-X and thermal simulations

From a mathematical point of view, the goal of the thermal solver is to solve a Poisson differential equation while considering a set of flexible boundary conditions. Additionally, a source term is present that describes the influence of the tissue perfusion and of metabolic processes in the body. These can contain non-linearities which makes the usage of ‘Fast Poisson Solvers’ (based on Fourier Transformation of the differential equation) and matrix-inversion\factorization methods unsuitable. In addition to this, the evolution of heating over time can be interesting in itself, which would suggest using a stepwise integration method for solving the equations. Therefore SEMCAD-X resorts to the finite differences time domain method (FDTD) of solving the problem by using a variant on a non-uniform grid.

This permits fast and locally highly resolved calculations (in critical and specially relevant regions) and the collection of information over the whole simulated time interval. Furthermore, non-uniform FDTD brings along the advantage of being able to use the EM simulation results on the same grids removing the need for interpolation and the associated error introduction. The same griddler and voxeler can be used for EM and thermal simulations.

The speed of FDTD permits the simulation of models with many millions of voxels. Another advantage of FDTD is that it allows every voxel to have its own set of material parameters. This permits highly
heterogeneous models to be simulated. For thermal simulations involving sophisticated bloodflow modelling this is of special relevance, as they ask for position dependent material inhomogeneities.

2.10.3 The Pennes bioheat equation

Developed in 1948 by Pennes, the ‘Bioheat Equation’ (PBE) is with certitude the most used model for thermal BioEM simulations. The formula is shown below:

$$\rho c \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) + \rho Q + \rho S - \rho_b c_b \rho \omega (T - T_b)$$  \hspace{1cm} (2.31)

where $k$ is the thermal conductivity, $S$ is the specific absorption rate, $\omega$ is the perfusion rate and $Q$ is the metabolic heat generation rate.

The heat is distributed through the simulation domain by diffusion. It is generated by metabolic body processes and by the deposited radiation energy. A homogeneous ‘heat-sink’ term has a cooling effect and models the heat removal due to blood circulation. Metabolic heat generation, heat capacity, heat conductivity, density and perfusion coefficients are all tissue specific. Equation 2.31 defines SAR distribution.

The PBE has been implemented in SEMCAD-X but a series of important extensions have been added to overcome existing short-comings. Still it was attempted to keep close to the Pennes model in order to retain the simplicity of his approach and to make it possible to reuse material parameters as they have been determined experimentally for the PBE.

2.10.4 Improvements to the Pennes bioheat equation

- An anisotropic conductivity: For every solid an anisotropic conductivity can be defined. Currently the main axes of the anisotropy have to be aligned to the grid axes, but it is intended that a full tensorial heat conductivity will be implemented in the next SEMCAD-X thermal solver release. The anisotropic heat conductivity can be used in various situations. An obvious one concerns the modeling of solids that, due to their makeup, actually have an anisotropic heat conductivity (e.g. striped muscle). Another, perhaps less obvious but more relevant example, would be the modeling of a more realistic perfusion. In the human body arteries and veins often run paired (counter current networks; e.g. in muscle). This leads to an effectively enhanced heat diffusion along the main blood-flow direction and can be modeled through an effective, anisotropic heat conductivity model.

- Temperature dependent parameters: The perfusion can be made linearly temperature dependent ($\omega = aT + b$) and the same is true for the electric conductivity ($\sigma = aT + b$) and the heat generation rate ($Q = aT + b$). This accounts for the fact that perfusion is strongly temperature dependant. Modeling this dependence can strongly increase the reliability of simulations, and having a temperature dependent $\sigma$ means that the resulting SAR that goes into the heating term becomes temperature dependent. This is a first approximation to the fact that the modified material parameters would actually change the results of the EM simulations as well - resulting in a modified heat source. Instead of repeating the EM simulation, we approximate the effect by using a temperature dependent $\sigma$. Instead of using a simple linear temperature dependence for the perfusion it is possible to specify a piecewise linear temperature dependence. This allows to approximate the thermoregulatory behavior more closely over a larger temperature range. In order to speed up the simulations only the voxels within a narrow band around transition temperatures are tested after each iteration and if necessary reassigned to other segments of the temperature dependence. The user can specify for each solid how often the list of voxels within the narrow-band is updated.
2.10. **Thermal Solver**

Temperature dependent heat transfer is a continuous function and the individual linear sections join seamlessly (no jumps at transitions temperatures, only change in slope).

- Discreet vasculature model / Thin wire model / Pseudo-1D boundary conditions (*DIVA* solver mode): *SEMCAD-X* permits to account for the influence of thin, nearly one dimension, thermally relevant structures. The temperature of voxels close to the surface of the thin object is used to estimate the flux through the object boundary using analytical models of the temperature behavior in that region. This can be used in the update equation for the thin structure. The thin structure is then allowed to interact with the bulk simulation after each time-step. One relevant example is *SEMCAD-X* ability to couple thermal simulations with a network of discreet vessels. The impact of large vessels on the surrounding can be highly relevant and cannot be modelled with the usual homogeneous models that cannot account for convective flow in vessels. An other example is provided by thin highly conductive structures (like wires) which result in extremely long simulations when modelled using a standard method, due to the large conductivity and the necessity for high resolution. The *DIVA* scheme allows to reduce the resolution, to increase the time step (by using an even larger cell size inside the 1D structures) and to still model the impact of the wire accurately. Furthermore the *DIVA* method can be used to model thin, nearly 1D boundaries (using Dirichlet, Neumann or Mixed boundary conditions, cp. 2.10.8). The grid resolution used to discretize the temperature distribution inside the thin object is can be set by specifying the bucket (equivalent of voxel in thin object) density of the number of buckets.

- Time dependent heat generation rate: It is possible to specify a time dependent heat generation rate. This permits the user to consider the changes of the body’s metabolic activity under thermal stress and the slow heating of exterior devices which are in contact with the body and heat up during usage. This can be relevant to mobile phones which include a battery that heats up during a conversation - an effect that can result in a temperature increase in the operator’s head comparable to the SAR induced heating.

### 2.10.5 Conformal mode and staircasing

A disadvantage of FDTD is that the voxeling of slanted surfaces results in a staircasing effect. This leads to an overestimation of the heat flux through surfaces. This error cannot be reduced by using a finer grid. Therefore it is possible to activate a conformal version of the simulation by checking the corresponding box in the simulation settings. The conformal solver extracts information about the local surface orientation and uses it to completely remove the staircasing errors at external interfaces, but at a cost of increased preprocessing. For internal interfaces (that constitute a smaller problem) an improved averaging method is used to reduce the staircasing error by modifying the local solid parameters. The staircasing corrections give excellent results, as soon as the grid step is small enough to resolve the local surface curvature.

### 2.10.6 Implementation

Various EM results can be used in parallel to simulate heating. This can be helpful when several antennas (or an antenna array) are operated independently. The fields can be added in a correlated way accounting for interference effects or in an uncorrelated way (only SAR added - e.g. for sources with uncorrelated phases, phase noise etc.). For coherent addition, phases and scaling factors can be specified for the various fields. The scaling factors scale the SAR (the E-fields get scaled by the square root). These heating sources can be switched on and off during the simulation time. This permits one to approximate time dependent EM sources in a staircased way. The next thermal solver version will include time-dependent, user-defined scaling functions permitting the simulation of pulsed heating.

It can be specified which solids are active during the simulation. The simulation will only be performed
on active solids, while the other solids will remain at the initial temperature. This has various advantages:

- The simulation time is reduced, as the calculations are only performed on the required parts of the model.
- Sections that reduce the maximal stable time-step do not need to be simulated, permitting the time-step to be increased, often resulting in considerably faster simulations. The time-step is most strongly reduced by active solids with low density or high thermal conductivity, therefore it is recommended to deactivate background solids (air) and metallic parts where not relevant for the thermal simulation.

2.10.7 Optimizing the EM settings for a thermal simulation

It is important to realize that metal parts have a completely different role in electromagnetic and thermal simulations. In EM simulations they dominate the field behaviour and have to be modelled separately to get accurate results. This means that they have to be voxeled in a different way. For thermal simulations, however, they have to be treated just like any other body (their extreme thermal conductivity has a very negative impact on the maximal stable time-step though).

The way in which metals are voxeled for EM simulations can result in voxels that seem to be unassigned and would be treated as background if the same grid was used for a thermal simulation. This does not affect the EM simulation results, but strongly disturbs the thermal simulations. Therefore it is recommended to decide before performing an EM simulation, whether the results will be used for a thermal simulation.

If the answer is yes, one should check the voxeling option Optimize for Thermal Simulation in the EM simulation before performing the simulation, in order to make sure that the calculations for the EM fields are performed on the same voxel model used for the thermal simulations. Furthermore, one should check the option Consider Thermo-Sensors in the griddler when generating a grid, which takes care of present thermal sensors in an optimal way.

2.10.8 Thermal Boundaries

Boundary conditions are needed at the boundaries of the active simulation domain. If no specific boundary has been specified a Dirichlet boundary condition is used and the temperature at the interface is fixed to 25°C.

The user can define boundaries in a very flexible way. For every solid A he can specify the boundary conditions acting on it at its boundary to any other solid B. Only the solid A is affected by this boundary condition and another one can be specified for the solid B at its interface to A. So it is possible, for example, to specify a fixed temperature of 37°C for muscle at its interface to a blood vessel, while the blood itself can receive a given flux from the muscle.

Various types of boundaries can be specified:

- Dirichlet: A fixed temperature is given for the interface. \( T = T_{\text{Boundary}} \)

- Neumann: A fixed heat flux is specified. The sign determines whether this flux flows in or out of the solid \( (kdT/dn = F_{\text{Boundary}}) \)

- Mixed: The flux depends on the local temperature and equilibrates it to the specified environment temperature based on a heat transfer coefficient \( (h) \). Additionally a given flux can be added:

\[
kdT/dn + h(T - T_{\text{Outside}}) = F_{\text{Boundary}}
\]
2.10. Thermal Solver

A radiative exchanged of the form:

\[ \sigma (T^4 - T_{\text{outside}}^4) \]

can be expressed as:

\[ \sigma (T - T_{\text{outside}})(T + T_{\text{outside}})(T^2 + T_{\text{outside}}^2) \]

which can, for small temperature differences, be approximated as:

\[ 4 \sigma (T - T_{\text{outside}})T_{\text{outside}}^3 \]

and described as a mixed boundary condition.

2.10.9 Thermal Dose

When assessing exposure to (EM induced) heat or for medical treatments involving heating up the patient, it is helpful to specify the amount of absorbed heat using a thermal dose. The most common dose concept is the CEM\(_{43}\) developed by Sapareto and Dewey. It is defined as

\[ CEM_{43} = \int_{t_{\text{final}}}^{0} R_{43}^{t_{\text{final}} - T(t)} dt \]

with \( R = 0.5 \) for \( T > 43 \), \( R = 0.25 \) for \( 39 > T \geq 39 \) and \( R = 0.5 \) for \( T \leq 39 \). If it has been specified in the simulation settings that the \( CEM_{43} \) should be determined, it becomes possible to activate recording of \( CEM_{43} \) for selected field sensors. One can specify for which solids the \( CEM_{43} \) is constantly calculated during the simulation (thereby taking into account the transient temperature behavior) and for which tissues the final temperature is used (\( CEM_{43} = t_{\text{final}} R_{43}^{t_{\text{final}} - T_{\text{final}}} \)).

Constantly updating the \( CEM_{43} \) slows down the simulation strongly but is a necessity if the transient behaviour affects the simulation.

2.10.10 Tissue Damage

When assessing exposure to (EM induced) heat or for medical treatments involving heating up the patient, it is common to calculate the resulting tissue damage. Often, an Arrhenius model is used to assess this:

\[ F = 1 - e^{-\Omega}, \Omega = \int_{t_{\text{final}}}^{0} Ae^{(E_A/R)T(t)} dt \]

Where \( F \) is the tissue damage, \( \Omega \) is the damage integral, \( E_A \) and \( A \) are the Arrhenius parameters and \( R \) is the universal gas constant. \( T \) has to be converted to \textit{Kelvin} for this evaluation.

If it has been specified in \textit{Simulation Settings} that the tissue damage should be recorded, the user gains access to activate this recording individually, for each field sensor in the simulation. One can specify for which solids the tissue damage is constantly calculated during the simulation (thereby taking into account the transient temperature behavior) and for which tissues the final temperature is used (\( \Omega = t_{\text{final}} A e^{(E_A/R)t_{\text{final}}} \)).

Constantly updating \( \Omega \) slows down the simulation strongly but is a necessity if the transient behaviour matters.

A future release of the thermal solver will allow the user to specify a dependence of tissue parameters on the tissue damage state (e.g., \( \omega = aF + b \)).
2.10.11 Phase Transitions

**SEMCAD-X** allows the user to specify (multiple) phase transitions for various tissues.

When tissue reaches specific temperatures, phase changes can occur. An example would be cell water evaporating at 100°C. Crossing these phase transition temperatures requires energy (specific enthalpy $h$). So, for a period, the temperature will stop rising (or falling) despite the energy increase (or decrease), resulting in a plateau of the temperature-time curve.

It is interesting to note that, for example, in the case of evaporating water, the water increases in volume by a factor of about 350, and most of the water leaves the tissue. Therefore, considerably less energy is released during the condensation. This can be considered by specifying a volume ratio for the solid, specified by $h_{down}$. If rehydration of the tissue can occur (water diffuses back into the cell, for instance), the user should specify this and fix the rate at which the water diffuses back (in **SEMCAD-X** the rehydration rate is an exponential rehydration).

2.10.12 Steady State Initialization

When the main interest of the user lies in obtaining the steady state thermal distribution, it can be helpful to use the **Steady State Init.** option in the **Simulation Settings**. The simulation then starts with a distribution that is obtained by setting the thermal conductivity to zero and solving the resulting equation exactly (there are special considerations used on the boundaries). This is a good first guess for the user when diffusion does not have a major influence on the steady state result. It means, however, that no information about the transient behavior will be available for analysis.
Chapter 3

Model Generation

3.1 Parts of a Model

In SEMCAD-X, a model is composed of the following parts:

- **Points** are used as construction aids and as position references.
- **Curves** are used as profiles or rails for constructing solid bodies.
- **Surfaces** can be used to represent 2D or quasi-2D structures, special treatment is applied when voxeling these parts.
- **Solids** represent physical bodies with a volume that make up the structure and represent different material types.
- **Groups** can be used to organize the different parts of the model and are especially useful for more complex structures which contain many parts.
- **Sources** define regions in the model where electromagnetic energy is injected into the model.
- **Lumped Elements** are inserted as edge elements to add passive resistive, capacitive or inductive functionality to the model.
- **Sensors** define regions in the model where field data is recorded during the simulation.

3.2 Creating and Modifying Parts

To assemble or modify a model, you should first create a new project or open an existing project. Make sure that the Model tab on top of the project pane is selected. You can now add different parts using all the tools which are shown in the modelling toolbar shown in Figure 3.1.

![Modelling toolbar](image)

Figure 3.1: Modelling toolbar

The modelling toolbar contains the tools which are used to create and modify solids, sources, lumped elements and sensors in the model. Some tools require that one or more parts be selected first. Observe the status line at the bottom of the main window. It gives you hints about what to do next.
3.2. Creating and Modifying Parts

In addition to the standard modelling toolbar, the additional toolbars (see Figure 3.2) offer additional functionality for further modeling functions found in the Tools menu.

Figure 3.2: Additional toolbars: modelling (top), parameterized modelling (center), local parameterized modeling (bottom left), simulation (bottom center) and transform and link (bottom right)

Some modeling tools ask you to specify points in the model. You can do this either by selecting Points \( \varepsilon \) that have already been added in the model or by entering values directly into the dialog (see Section 3.3). Also, many of the modeling dialogs can be Pinned \( \mathbb{R} \), which will keep the dialog open so that the operation can be repeated again.

All modeling operations can be undone. To undo a modeling step, select Edit | Undo from the menu or press the \( \Rightarrow \) button on the main toolbar. To undo an undo step, select Edit | Redo or press the \( \Rightarrow \) button. Also, Section 3.8 explains how to step backwards more than one modelling operation using the Model History.

Figure 3.3: Main toolbar

Figure 3.3 shows the main toolbar, which offers some general operations:

- For projects: New Project \( \mathbb{R} \), Open Project \( \mathbb{R} \) and Save \( \mathbb{R} \)
- For simulations: Run \( \mathbb{R} \), Run Batch \( \mathbb{R} \), Continue Simulation \( \mathbb{R} \), Stop and Save \( \mathbb{R} \), Abort \( \mathbb{R} \) and View Convergence \( \mathbb{R} \). For more information on these, see Section 5.11.
- For model parts: Cut \( \mathbb{R} \), Copy \( \mathbb{R} \), Paste \( \mathbb{R} \) and Delete \( \mathbb{R} \)
- For general views: Orbit \( \mathbb{R} \) and Pan \( \mathbb{R} \)
- For zooming views: Zoom \( \mathbb{R} \), Reset All \( \mathbb{R} \), Rubberband \( \mathbb{R} \) and Zoom To Selection \( \mathbb{R} \)
- For orthogonal views: Front View \( \mathbb{R} \), Back View \( \mathbb{R} \), Top View \( \mathbb{R} \), Bottom View \( \mathbb{R} \), Right View \( \mathbb{R} \) and Left View \( \mathbb{R} \)

Tip: Right clicking in the model view will also display a menu with viewing options.

Note: Different modes can be selected for adjusting how the model is rendered (e.g. Transparent, Edges, etc.) from the View menu.

It is possible to change the view while a modeling tool is active. For example, if you are in the middle of drawing a polyline, you can select the Pan \( \mathbb{R} \) icon from the main toolbar to move the model within the view. Press the button again to release it, and continue with the polyline.
3.2.1 Solid Properties, Streamline Operations and Model Unit

SEMCAD-X also offers streamline operations which can be performed on single objects by simply entering parameters directly in the Settings for the object. An example is shown in Figure 3.4 (left). The following streamline operations are offered:

- **Name**: change the name of the object.
- **Visible**: set or unset the visibility of the object in the model view.
- **Color**: change the color of the object.
- **Scale**: perform a uniform scaling of the object.
- **Rotation**: rotate the object around the X, Y, or Z axis.
- **Translation**: translate the object in the X, Y, or Z direction.

Bounding Box information containing the minimum and maximum coordinates of the bounding box for the object is also provided.

Figure 3.4: Streamline modelling operations for a solid can be entered directly into the Settings for the part (left), while direct modification of the dimensions is possible when using parameterized objects (right).

Finally, the Model Unit can also be set (also see Figure 3.4). By default the unit is set to \( \text{Model Unit} = 1 \text{ m} \). The user can change these settings for new projects.

**Note:** For large models (e.g. containing dimensions expressed in the meter range) it is recommended that the unit be set to \( \text{Model Unit} = 1 \text{ m} \). This will improve the visualization of the model and prevent possible clipping effects.

The remainder of this chapter defines the different modeling functions and operations that are available as they appear in the modeling toolbar, which is shown in Figure 3.1 and Figure 3.2.
### 3.2. Creating and Modifying Parts

#### 3.2.2 Static, Parameterized and Local Parameterized Modeling Operations

**SEMCAD-X** now supports both static and parameterized modeling objects. Static modeling objects require additional modeling operations to modify the dimensions since they are stored in terms of their absolute dimensions. However, for the parameterized modeling objects the dimensions are stored as variables. This allows the geometry of the object to be modified simply by modifying the parameters in the GUI (see Figure 3.4 (right)). Similarly, during an optimization of a structure, the parameterized objects can be modified by the optimizer by updating the variables.

In addition to parameterizing entire objects, **SEMCAD-X** has the ability to perform modeling operations on local and isolated sections of individual solids in the model. These actions can be accessed from the Local Modeling toolbar. These operations would typically modify the edges and faces of a selected solid.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selector</td>
<td>In selection mode, you can select parts by clicking on them in the model view. If the View mode is set as Transparent, the part will no longer be transparent in order to highlight it in the model.</td>
</tr>
<tr>
<td>Move</td>
<td>Translates the selected parts. Either enter the start and end points of the translation vector directly into the dialog, or select them in the model with the mouse.</td>
</tr>
<tr>
<td>Rotate</td>
<td>Rotates the selected parts around an axis. Either enter the start and end points of the rotation axis directly into the dialog, or select them in the model with the mouse. Enter the angle of rotation.</td>
</tr>
<tr>
<td>Scale</td>
<td>Scales the selected parts. Both uniform and non-uniform scales are offered.</td>
</tr>
<tr>
<td>Points</td>
<td>Creates a point in the model. Points are useful as references for later operations.</td>
</tr>
<tr>
<td>Polyline</td>
<td>Creates a polyline. A polyline consists of a series of connected straight lines. Enter each vertex of the polyline and then press Next. Use the Back button to erase the last vertex. Press the Done key to finish the polyline. Polylines can be elements for profiles used with extrusions and spins.</td>
</tr>
<tr>
<td>Parameterized Polyline</td>
<td>Converts a polyline to a parameterized object.</td>
</tr>
<tr>
<td>Spline</td>
<td>Creates a spline through a series of points. Enter each point on the spline, press Next to enter the next point and press Back to delete the last point on the spline. Press Done when all points have been entered.</td>
</tr>
<tr>
<td>Parameterized Rectangle</td>
<td>Creates a rectangle. Modify the parameters in the GUI to obtain the correct dimensions.</td>
</tr>
<tr>
<td>Arc</td>
<td>Creates a circular arc from a center point and two points on the edge. Enter the center point and then the two edge points. If the second point on the edge is the same as the first one on the edge, the arc is closed.</td>
</tr>
<tr>
<td>Parameterized Circle</td>
<td>Creates a circle. Modify the parameters in the GUI to obtain the correct dimensions.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Sphere</td>
<td>Creates a solid sphere. Enter the center of the sphere and the radius.</td>
</tr>
<tr>
<td>Parameterized Sphere</td>
<td>Generates a solid sphere. Modify the parameters in the GUI to obtain the correct dimensions.</td>
</tr>
<tr>
<td>Brick</td>
<td>Creates a solid brick. Enter two points on opposite corners of the brick.</td>
</tr>
<tr>
<td>Parameterized Brick</td>
<td>Generates a solid brick. Modify the parameters in the GUI to obtain the correct dimensions.</td>
</tr>
<tr>
<td>Cylinder</td>
<td>Creates a solid cylinder. Enter the radius and the top and bottom center points (see Section 11.11).</td>
</tr>
<tr>
<td>Parameterized Cylinder</td>
<td>Generates a solid cylinder. Modify the parameters in the GUI to obtain the correct dimensions.</td>
</tr>
<tr>
<td>Helix</td>
<td>Creates a solid helix. Enter the top and bottom center points, the radius, distance between turns and wire radius (see Section 11.15).</td>
</tr>
<tr>
<td>Parameterized Helix</td>
<td>Generates a solid cylinder. Modify the parameters in the GUI to obtain the correct dimensions.</td>
</tr>
<tr>
<td>Mirror</td>
<td>Mirrors selected solid. Enter a point in the mirror plane and a normal point such that the line between the 2 points is perpendicular to the mirror plane. Check the Make Copy box to copy and mirror the selected solid (see Section 11.18).</td>
</tr>
<tr>
<td>Array</td>
<td>Copies and translates geometry for fast modeling of arrays or repetitive structures (see Section 11.4).</td>
</tr>
<tr>
<td>Distance Tool</td>
<td>Calculates the distance between two points. A point can be created at the center of the line. (see Section 11.13).</td>
</tr>
<tr>
<td>Explode</td>
<td>Creates thin sheets of zero thickness from each face of the selected solid.</td>
</tr>
<tr>
<td>Create Shell from Body</td>
<td>Allows the user to create a hollow body from a solid. The user selects the solid which should form the outer shell, and the faces which should be excluded from the body, hence creating an opening into the shelled structure. The thickness of the shell walls must also be specified.</td>
</tr>
<tr>
<td>Imprint</td>
<td>When two solids intersect, this function creates an edge in both solids at the edges of the intersection.</td>
</tr>
<tr>
<td>Space Warp</td>
<td>Deforms the selected solid in the x, y and z axes as defined by the user-defined equations.</td>
</tr>
<tr>
<td>Get vertices</td>
<td>Creates points at all vertices of the selected parts. This is useful if you want to get reference points from existing solids or curves. Additionally, points at the centers of circular arcs and spherical faces are created by this function.</td>
</tr>
<tr>
<td>Extrude</td>
<td>Extrudes a profile into a solid. Select the profile to extrude and enter the two points for the extrusion vector. To create a solid part the profile should be closed. Otherwise a sheet will be created.</td>
</tr>
<tr>
<td>Parameterized Extrude</td>
<td>Extrudes the selected parameterized profile in the Z-direction. Once the extruded part has been created adjust the Amount, Translation and Rotation to achieve the required geometry.</td>
</tr>
</tbody>
</table>
# 3.2. Creating and Modifying Parts

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sweep</strong></td>
<td>Forms a solid body by sweeping a profile along a rail. Select the closed profile and the rail along which to sweep. There are restrictions about the relative positions of the profile and the rail. Sweeping works best if the rail starts perpendicular to the profile’s plane, inside the profile. You may want to experiment with different layouts and see what the modeling kernel can handle. Both the profile and the rail must be single, contiguous curves.</td>
</tr>
<tr>
<td><strong>Spin</strong></td>
<td>Forms a solid body by spinning a profile around an axis. Select the closed profile to spin, enter the coordinates of the rotation vector and the spin angle.</td>
</tr>
<tr>
<td><strong>Parameterized Spin</strong></td>
<td>Creates a parameterized object by rotating the selected parameterized object around the Y-Axis. Once the has been created modify the Angle as required.</td>
</tr>
<tr>
<td><strong>Skin</strong></td>
<td>Forms a body by pulling a skin over a series of profiles. If the profiles are closed curves, the resulting body will be a solid, capped with faces made from the first and the last profile. If the profiles are open curves, a sheet body will result. Select all the profiles and press Done to create the skin.</td>
</tr>
<tr>
<td><strong>Perpendicular Offset</strong></td>
<td>Enables the scaling of the selected solid perpendicular to the surface of the solid. Enter the distance of the offset from the surface in mm.</td>
</tr>
<tr>
<td><strong>Unite</strong></td>
<td>Unites two parts into a single part. Select all the parts from the model tree with the mouse while holding the Ctrl or Shift keys. Press Done to create the new part. Objects must touch or intersect to be united successfully.</td>
</tr>
<tr>
<td><strong>Subtract</strong></td>
<td>Subtracts one part from another. Select the two parts from the model tree with the mouse while holding the Ctrl or Shift key. Press Done to create the new part. The second part will be subtracted from the first one. It is possible to mix solids and curves in this operation. You may, for instance, cut out a piece of a curve by using a solid brick as a “knife”.</td>
</tr>
<tr>
<td><strong>Intersect</strong></td>
<td>Creates the intersection of two parts. This is the region common to both parts. Select the two parts (curves and/or solids) from the model tree with the mouse while holding the Ctrl or Shift key. Press Done to create the new part.</td>
</tr>
<tr>
<td><strong>Parameterized Combined</strong></td>
<td>Perform boolean operations on parameterized objects to create a combined parameterized object (see Section 11.62).</td>
</tr>
<tr>
<td><strong>Edit Parameterized</strong></td>
<td>Dependencies can be introduced to the selected parameterized object (see Section 11.59).</td>
</tr>
<tr>
<td><strong>Create Parameterized</strong></td>
<td>Adds parameterized control to the selected object.</td>
</tr>
<tr>
<td><strong>Remove Parameters</strong></td>
<td>Removes the parameterized control from the selected object.</td>
</tr>
</tbody>
</table>
New group

Creates a new group. The parts making up the model can be arranged in groups. Each group may hold points, curves, bodies, sources, sensors and other groups. The groups pane displays the hierarchy of groups as a tree; the parts pane shows the contents of the currently selected group, much like Windows Explorer displays folders and files. You may cut, copy and paste parts.

View Edges

Highlights the edges of the solids in the model.

View Hidden Edges

Displays the edges that are hidden from view.

Black Edges

Changes the colour of the solid’s edges to black.

Blend Edges

Transforms an edge between two faces of a solid into a smooth transitional face.

Chamfer Edges

Creates a beveled surface to eliminate an edge.

Extract Edges

Extracts the selected edges into Spline elements.

Move Faces

Moves the selected faces of a solid by a specified direction, in a defined distance. The solid remains as a solid structure, and is merely deformed (see Section 11.57).

Offset Faces

Offsets the selected faces in the direction of the normal to the selected faces. The solid remains as a solid structure.

Remove Faces

Removes the selected face and extrapolates the adjoining faces to intersect each other and create an edge. This could also be used to remove a hole from a solid.

Sweep Faces

Moves the selected faces along a path defined by the faces adjacent to those selected. This is especially useful for shortening a curved slot by moving the end face(s) along the curved path of the slot.

Heal Selected

Heals the selected solids, e.g. reverts normal vectors for imported CAD parts where the normal vectors may have been inverted during conversion.

3.2.3 Sources

Plane wave source

Creates a plane wave source. The plane wave source is represented as a wire-frame box in the model. Enter two points on diagonally opposite corners of the box. The edges of the box are always parallel to the grid axes. The electromagnetic properties of the source are set up in the simulation settings.

Edge source

Creates an edge source. The edge source is represented as a straight edge in the model, restricted to run parallel to one of the grid axes. Enter the two endpoints of the edge. The electromagnetic properties are set up later in the simulation settings.

Parameterized Edge Source (X)

Creates a parameterized edge source in the X-direction

Parameterized Edge Source (Y)

Creates a parameterized edge source in the Y-direction
3.2. Creating and Modifying Parts

<table>
<thead>
<tr>
<th>Parameterized Edge Source (Z)</th>
<th>Creates a parameterized edge source in the Z-direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waveguide source</td>
<td>Creates a waveguide source. The waveguide source is represented by a red 2D rectangular surface. This source can be used for exciting rectangular, parallel plate and circular waveguides or coaxial cables. For rectangular and parallel plate waveguides, place the source such that its size corresponds with the inner rectangle of the cross section of the guide (Figure 3.5 (a)). For cylindrical/coaxial feeds the rectangle should be tangential with the inner wall of the outer conductor and the surface of the rectangle should be totally covered by the geometry as shown in Figure 3.5 (b)-(c). The electromagnetic properties are set up later in the simulation settings. Note that, the user has to employ waveguide tools under Tools menu in order to build both the waveguide source and also the sensors which are necessary to extract S-parameters.</td>
</tr>
</tbody>
</table>

**Note:** Edge Sensors are automatically added at the terminals of all Edge Sources and Lumped Elements to record all significant time and frequency domain parameters. The names of the sensors are based on the names used for the Edge Source or Lumped Element.

Figure 3.5: Modeling of the Waveguide Source: (a) rectangular waveguide, (b) circular waveguide and (c) coaxial cable

3.2.4 Lumped Elements
Lumped Element

Creates a lumped element. The lumped element is represented as a straight edge in the model, restricted to run parallel to one of the grid axes. Enter the two endpoints of the edge. The lumped element can be used as a resistor, capacitor or inductor.

3.2.5 Sensors

Field Sensor

Creates a sensor to record electromagnetic fields. The sensor is represented as a 1D, 2D or 3D box in the model. Enter the two points on diagonally opposite corners. The edges of the box are always parallel to the grid axes.

Edge Sensor

Creates an edge sensor to record voltage, current, power, impedance, etc. The sensor is represented as a straight wire in the model. Enter the two endpoints to define the wire. The wire is always parallel to the grid axes.

Voltage sensor

 Creates a voltage sensor. The sensor is represented as a straight wire in the model. Enter the two endpoints to define the wire.

Current Sensor

Creates a current sensor. The sensor is represented as a polygon loop in the model. Enter the series of points to define the loop and press Done after the last corner. The loop will be closed automatically.

Port Sensor

The port sensor records the voltage along an integration path and the current which flows through a rectangular integration loop. First model the Voltage and Current Sensor separately and then use the Port Sensor to integrate the sensors into a single port. This sensor is very useful for microstrip-type applications.

Thermo Box Sensor

Creates a sensor to record thermal field distributions as a function of time. To define the box enter the two points on diagonally opposite corners. The edges of the box are always parallel to the grid axes.

Thermo Point Sensor

Records the temperature as a function of time at a specific point.

Note: The Far Field Sensor is now added automatically when the model is voxelized. If Automatic Padding is used the grid will be generated correctly with the Far Field Sensor inserted. See Section 7.2.4 for further information related to Padding.

3.3 Entering Points into Modelling Dialogs

Many of the modeling operations require that points are entered as input. Points can be entered in two ways: by directly typing the coordinates into the dialog or by selecting the position in the model view with the mouse. For the latter, the cursor can snap to discrete points in the model, e.g. to corner vertices of the parts in the model. Normally, the tracker moves in the XY plane, but can be moved in the in the positive or negative Z direction by holding the Ctrl key.
3.4 Grouping Parts

It is normally easier to use Points for modeling operations requiring many points to be entered. The dialog can be Pinned (see Figure 3.6) such that more than one point can be entered without closing it.

![Figure 3.6: Creating a new point](image)

3.4 Grouping Parts

Parts can be organized in Groups. Each group can hold simple parts and nested groups. You may copy and paste parts between groups. The nesting of groups is displayed in the groups pane at the left edge of the main window. One group is selected at a time. This is the current group. Its contents are listed in the parts pane to the right of the groups pane. Newly created parts are always added to the current group. To make a group the current group, select it in the groups pane or double-click it in the parts pane. The backspace key acts as a shortcut for selecting a group’s parent if the parts pane has the focus. Selecting a group in the parts pane causes all its member parts to be selected; and hiding a group causes all its members to be hidden.

3.5 Interactive Modelling Tools

Typically, when a model is generated, exact dimensions of the structures are known. However, this maybe not always be the case e.g., when positioning a mobile phone against a users head or adapting a hand grip to hold an object. In both of these cases the exact co-ordinates and angles for the required positioning might not be known. Special interactive modelling tools have been developed to treat these cases including:

- **Interactive Translation**: allows the selected parts to be interactively translated in either the X, Y or Z direction (Section 3.5.2).

- **Interactive Rotation**: allows the selected parts to be interactively rotated around the X-, Y- or Z-axis of the selected part (Section 3.5.3).

- **Interactive Linking**: allows multiple parts to be linked to a parent part, such that when interactive translation and/or rotation is performed, the linked children are moved with the parent (Section 3.5.5).
3.5.1 Pivots

Every part in a SEMCAD-X model has by definition a pivot or local co-ordinate system assigned to it (Figure 3.7). Interactive translation and rotation operations are performed with respect to the Pivot of the part rather than to the global origin. The pivot of a part can only be seen when one of the Link, Translate or Rotate tools are active and the part is selected.

![Pivot location and orientation at the center of the helix (left), interactive translation tool (center) and interactive rotation tool (right)](image)

**Figure 3.7:** Pivot location and orientation at the center of the helix (left), interactive translation tool (center) and interactive rotation tool (right)

**Note:** Currently, the use of interactive tools is not supported for parameterized objects.

**Note:** When a new part is created the default location of the pivot is typically at (0, 0, 0) and aligned with the global axes.

3.5.2 Interactive Translation

Select the Interactive Translation tool and select the part(s) to be translated:

1. Move the mouse over one of the three axes of the Pivot until it is highlighted in yellow (Figure 3.7).
2. Hold the left mouse button down and drag the object in either the positive or negative direction. The translation vector is printed to the screen.
3. Release the mouse button when the part is in the correct position.

**Note:** When the Interactive Translation tool is used the Translation Settings (Figure 3.4) are updated accordingly.

**Tip:** Translation to tracker point is also supported: choose the part to translate and hold the SHFT key down. The point tracker is now activated, move the mouse to the required location and left-click: the part will be moved such that its Pivot is aligned with the point.
3.5. Interactive Modelling Tools

3.5.3 Interactive Rotation

Select the Interactive Rotation tool and select the part(s) to be rotated:

1. Move the mouse over one of the four circles until it becomes highlighted in yellow (Figure 3.7): the red circle to rotate around the X-axis, the green circle to rotate around the Y-axis, the blue circle to rotate around the Z-axis and the black circle to rotate in the current view plane.

2. Hold the left mouse button down and rotate the object by moving in either the positive or negative angular direction. The rotation vector is printed to the screen.

3. Release the mouse button when the part is in the correct position.

Note: When using the Interactive Rotation tool the Rotation Settings (Figure 3.4) are updated accordingly.

3.5.4 Additional Functions for Interactive Rotation and Translation

When using the interactive tools the following additional function menu (Figure 3.8) can be accessed by right-clicking in the model window.

![Figure 3.8: Additional function menu for interactive tools](image)

- **Transform Pivot Only**: translation or rotation operations are only performed on the Pivot.
- **Reset Pivot To Center**: resets the location and orientation of the Pivot to the center of part.
- **Snap Transformation**:
- **Use Pivot System**: when activated, translations and rotations are performed in the direction of and about the Pivot axes instead of the global X-, Y- and Z-axis.
- **Show Links**: show link trees as defined in Section 3.5.5.

Tip: By default all translations and rotations are performed in the direction of and about the X-, Y-, Z-axis. By activating the Use Pivot System the transformations will be performed using the Pivot axes. It is then possible to e.g. rotate an object about its local Pivot Z-axis, which may not have the same orientation as the global Z-axis. Similarly an object can be translated along the local Pivot Y-axis.
### 3.5.5 Interactive Linking

A Link Tree can be created by linking multiple parts (children) to another part (parent) such that when interactive translation and rotation operations are performed on the parent the same operations will be carried out on the children.

1. Choose the Interactive Linking tool (the shortcut key is Ctrl+Shift+L).
2. Select the child(ren) object(s): this can be done either by selecting the objects directly in the model widow or in the parts list. To select multiple parts hold the Ctrl key down.
3. Link objects to the parent: once the last child has been selected hold the left mouse button down, move the mouse over the parent and release the mouse button. The children are now linked to the parent as shown in Figure 3.9.
4. Additional linking functions can be selected by right-clicking in the model window:
   - **Select Parent**: an alternative to Step 3, once the children have been selected this function allows the user to choose the parent from the parts list.
   - **Unlink Selection**: unlinks the selected objects.
   - **Unlink All Connected**: unlinks all connected parts in the current Link Tree.
   - **Selected All Connected**: selects all connected parts in the current Link Tree.

![Figure 3.9: Step 2: Multi-selection of children in GUI (left), Step 3: linking the children to the parent (center) and additional linking menu functions (right) ![Figure 3.9: Step 2: Multi-selection of children in GUI (left), Step 3: linking the children to the parent (center) and additional linking menu functions (right)](image)

### 3.6 Local Systems

Using Local Systems allows the user to create a local co-ordinate system and e.g. perform rotation and translations in this local co-ordinate system. This is especially useful, e.g. for positioning a mobile phone device against the head where a Local System can be defined at the ear to simplify the positioning. The process for creating a Local System can be defined in the following steps:

1. Create a Local System in the model using Tools | Head Positioning | Create Local System
2. Select or enter the correct Position and Translation for the Local System. Press Next.
3. Select the objects which will be Linked to the Local System.

**Tip:** Objects can also be linked to the Local System after it has been created. Instead of pressing Next press Done and then use the Interactive Linking tool to link the objects to the Local System.
3.6. Local Systems

Figure 3.10: Transformations can be either applied to the Local System or to the linked Entities only directly in the Setting.

Once the Local System has been created, transformations can be performed using \( \Delta^{\text{t}} \) and \( \Delta^{\text{rot}} \) tools or directly in the Settings of the Local System as shown in Figure 3.10. These transformations can either be applied to the Local System and the linked Entities or just to the linked Entities.

**Note:** Currently, the interactive rotate and translate tools can not be applied to a Local System.

### 3.6.1 Local Systems for Head Positioning

Local Systems offer an ideal platform to simplify head positioning against a mobile phone device: once a system has been defined for the head and for the device the Snap Local Systems tool can align the two systems and therefore position the head against the device. The steps can be described as follows:

1. Define an audio output point on the device where the ear point on the phantom will be positioned.
2. Define a Local System at the audio output point. It is not necessary to link the device to the Local System since the head will be moved to the device.

   **Tip:** The Local System should be defined with the same orientation as that of the head (see Figure 3.11).

3. Import either SAM\_Head\_LE.sat to position on the left ear or SAM\_Head\_RE.sat to position on the right ear. Both these .sat files have predefined Local System on the left or right ear. Alternatively, import the regular SAM\_Head.sat and define the Local System as described in the previous Section.

4. Choose Tools | Head Positioning | Snap Local Systems, select the Local System of the head, press Next, select the Local System of the device and press Done. The two Local Systems are now aligned.
5. The final movement for touch position involves moving the head until it touches the device: select the Local System of the head and change the Entity Rotation around the Y-Axis until the head is touching the device.

Figure 3.11: 1) Define the audio output point on the device, 2) create a Local System for the device 3) imported head model 4-5) align the Local Systems and rotate around the Y-Axis until the head touches the device.

3.7 Importing and Exporting

The following CAD formats can be imported into SEMCAD-X by selecting File | Import Model from the menu:

- ACIS® SAT and SAB
- IGES
- ProEngineer assemblies and parts
- CATIA
- STEP
3.7. Importing and Exporting

- 3DS
- STL
- I-DEAS
- Gerber
- DXF

**Tip:** It is possible to import multiple CAD files simultaneously by selecting the files using the `SHIFT` or `CTRL` keys while browsing for the files to be imported.

**Note:** For licensing reasons, not all of these formats are activated by default. Please contact the support team to enable importing of specific formats.

Furthermore, selecting File | Export Model model will enable SAT, IGES and STEP formats to be exported.

In addition to pure importing of CAD data, healing of parts can be applied. **SEMCAD-X** offers different options with respect to healing, for more detailed information see Chapter 4 or Section 3.2.2.

**Note:** The Group environment in **SEMCAD-X** may not be recognized by other CAD tools when exporting the model from **SEMCAD-X**.

### 3.7.1 Gerber Import

If using the Gerber or DXF importer, the user will need to give **SEMCAD-X** additional information after selecting the file to import. This information includes whether separate, but overlapping traces in the Gerber file should be combined into a single, more manageable entity. The required thickness of the layer that is imported should also be defined, as well as the **Arc deviation**. This value defines the maximum deviation that is allowed between the true arc in the Gerber or DXF file and the polygon approximation of it. It can be thought of as the maximum deviation allowed in approximating the arcs. This dialog is shown below in Figure 3.12

**Note:** The success of Gerber importation is highly dependant on the structure and quality of how the Gerber file was created.

### 3.7.2 Importing Voxel

It is also possible to import voxels into **SEMCAD-X** by using the Voxel Importer, which is accessible in the File menu.

The voxel importer takes one .raw file (volume), or several .bmp files (slices) and converts them into the ‘segmented slices’ format. The resulting file is subsequently loaded into the model.

The following is a description of the various fields in the voxel importer dialog (shown in Figure 3.13):

| File type | Select the desired input file type as .RAW or .BMP. |
Files
Use the Browse button to select the input file(s) (one .RAW or several .BMPs). If you specify several .BMP files, they are ordered based on their order in the file selection dialog under File name. Usually this corresponds to the order that the files are sorted. The only exception is the file holding the cursor focus (i.e. selected last), which will always be placed first in the list. Therefore, it is advisable to select the files in the list from the last file backwards.

Outline file
A file should be specified that will hold the segmented slices. This file will remain after the importing and can be used to quickly load the model again using File | Import Model | Segmented Slices File....

Voxel size
The dimensions of a single voxel.

Voxel count
The number of voxels in each direction (only necessary for .RAW files).

Bit depth
The bit depth (only necessary for .RAW files).

Use Selected Tissues
When checked, a file has to be specified for the Tissues file field, which holds the number of tissues to load (a simple .TXT file, one ASCII number per line).

Use Grouping
When checked, a file has to be specified under Grouping file. This file specifies if several voxel values should be grouped into a single tissue (e.g., If voxel values 1,2,3 and 4 should give a single tissue type of value 2 in the voxel file, enter the following pairs in the file : 1 2, 3 2, 4 2. This will map the values 1,3 and 4 onto 2 as well.) This can be used for grouping, or to convert voxel values.

Use Color Names
When checked, a file has to be specified under CN file that gives the names and the RGB colors for the various tissues.

3.8 Model History

The Model History window (Figure 3.14) can be opened from the Edit | Model History menu. The window shows a list of the modelling operations that have been performed. The green arrow points to the current state of the model in terms of the modelling operations.

- **Restore**
  Select the desired state in the history list. Push Restore and the model state will be restored to the selected state.

- **Undo**
  Sets the model state to the previous modelling operation.

- **Redo**
  Sets the model state to the next modelling operation. This operation is only meaningful if the model is in an early state than the last operation that was performed, i.e. if the green arrow is not pointing to the last item in the list.

- **Clear All**
3.9 Shortcut Keys

Saves the model in the current state and clears the modelling history.

3.9 Shortcut Keys

Certain shortcut keys are activated in SEMCAD-X to perform operations without using the mouse:

- Ctrl+C: Copy selection.
- Ctrl+X: Cut selection.
- Ctrl+V: Paste selection.
- Shift+F6: Toggle active focus window.
- Ctrl+H: Opens the model history window.
- F5: Run active Python script.
- Ctrl+A: Select all items in a list.
- Shift: Used in combination with the mouse to select a portions in a list.
- Ctrl: Used in combination with the mouse to select individual items in a list.
- Ctrl+Shift+M: Open interactive translation tool.
- Ctrl+Shift+R: Open interactive rotation tool.
- Ctrl+Shift+L: Open interactive linking tool.

Tip: Many of the shortcut keys can be applied at multiple locations within SEMCAD-X. For example, the copy and paste combinations can be used in the model environment to copy objects and groups or in the simulation environment for copying settings within a single simulation or between different simulations.
Figure 3.12: The Gerber/DXF file import settings dialog

Figure 3.13: The Import Voxels dialog
3.9. Shortcut Keys

Figure 3.14: The Model History window allows easy control, undoing and stepping backward through modelling operations that have been performed.
Chapter 4

Global Preferences

The Global Preferences are accessible via Preferences... in the File menu. In the Preferences the user can define a number of settings applying to the different entities of SEMCAD-X. Figure 4.1 shows an outline of the Preferences window.

![Figure 4.1: The Preferences window.](image)

The following parameters and settings can be specified within the Preferences.

**Note:** Depending on the status of the User Level, different settings are visible and accessible. Simply close and re-open the Preferences dialogue after changing the User Level.

<table>
<thead>
<tr>
<th>Colors</th>
<th>Background</th>
<th>Allows the color of the model background to be set.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modeling</td>
<td>Show Grid</td>
<td>Defines if the grid is shown in the model generation window or not.</td>
</tr>
<tr>
<td>Modeling</td>
<td>Show Coordinate System</td>
<td>Defines if the coordinate system (XYZ arrows) is shown in the model generation window or not.</td>
</tr>
<tr>
<td><strong>Cover Compound Slices</strong></td>
<td>This checkbox defines if the anatomical high-resolution models in SEMCAD-X are rendered with their cover shown or not. Showing the cover gives a nicer impression but generally slows the rendering process down. It is thus off by default.</td>
<td></td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td><strong>Auto-heal While Importing CAD Files</strong></td>
<td>When importing CAD data, the healing procedure is automatically activated. This might slow down the import process.</td>
<td></td>
</tr>
<tr>
<td><strong>Simplify Bodies While Healing</strong></td>
<td>Different levels of healing can be activated by SEMCAD-X. Setting this checkbox to on still performs healing but with reduced accuracy and number of triangles, however, in a faster way. It can also allow the healing of certain solids which might fail when healing with full accuracy.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Other</strong></th>
<th><strong>User Level</strong></th>
<th>Allows the mode for access and complexity for the user to be set. Detailed information is provided in Section 5.12.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Record PML Region of Overall Sensor</strong></td>
<td>Enables the user to record and visualize all fields within the PML ABC region in the simulation - in case of scientific interest regarding the PML performance. The feature is off by default.</td>
</tr>
<tr>
<td></td>
<td><strong>Do Compress SEMCAD X Documents</strong></td>
<td><em>Expert mode only.</em> Specify whether to compress SEMCAD-X Documents.</td>
</tr>
<tr>
<td></td>
<td><strong>Compression 1 (fast) ... 9 (best)</strong></td>
<td><em>Expert mode only.</em> Specify compression level.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>SEMCAD X Script</strong></th>
<th><strong>Script Folder</strong></th>
<th>Allows the user to specify a preferred location for SEMCAD-X scripts. This is the default folder location for the loading and saving of Python scripts from the GUI.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>Post Processor</strong></th>
<th><em>Expert mode only.</em></th>
<th>Allows the user to specify the number of steps (dynamics) regarding the color scale of the Contour Viewer in the Postprocessor.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Number of Contours 5x</strong></td>
<td><em>Expert mode only.</em></td>
<td>This field allows the user to specify the block size of the RAM cache allocated when performing the postprocessing. To enable and speed-up the extraction of large result data, the postprocessing can be performed in blocks. For larger cache sizes the extraction is faster, however, more memory is needed.</td>
</tr>
</tbody>
</table>

| **SEMCAD X Solver** | *Expert mode only.* | Enables the user to specify what kind of solver is to be used. |
**Custom Solver**

*Only of SEMCAD X Solver is set to custom.* Enables the user to specify an iSolve executable file different from the default one provided as installed by the SEMCAD-X installer.

**Custom Real Valued LF Solver**

*Only of SEMCAD X Solver is set to custom.* Enables the user to specify an LF-SolverReal executable file for real valued low frequency simulations.

**Custom Complex Valued LF Solver**

*Only of SEMCAD X Solver is set to custom.* Enables the user to specify an LF-SolverReal executable file for complex valued low frequency simulations.

### Rendering

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Transparency Amount</strong></td>
<td>Allows the user to specify the level of transparency applied to the solid models.</td>
</tr>
<tr>
<td><strong>Transparency Amount for Pre-selected Solids</strong></td>
<td>Specifies the transparency level of the user selected solids.</td>
</tr>
</tbody>
</table>

### Grid Engine

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Preferred Engine</strong></td>
<td>Enable the user to specify whether to show the interactive or the conventional grid engine by default.</td>
</tr>
<tr>
<td><strong>Show Info Dialog</strong></td>
<td>Enable the user to specify whether to show the info dialog for interactive gridding. This option can also be disabled by activating the check box <em>Do not show again</em> on the info dialog itself.</td>
</tr>
<tr>
<td><strong>Overriding of Existing Grids</strong></td>
<td>Allows the user to specify how to handle existing grids generated with the conventional grid engine when using the interactive grid engine.</td>
</tr>
<tr>
<td><strong>Remember Grid Override Choice</strong></td>
<td>Allows the user to specify whether SEMCAD-X shall remember the setting in Overriding of Existing Grids or show the conversion dialog.</td>
</tr>
</tbody>
</table>

### Advanced Gridding

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Enable Advanced Editor (Beta-Preview)</strong></td>
<td>Enable the user to specify to enable or disable the advanced grid mode.</td>
</tr>
</tbody>
</table>

### Power Tools

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Solver Process Priority</strong></td>
<td>Allows to specify the default process priority of the solver (within the OS, e.g. MS Windows).</td>
</tr>
<tr>
<td><strong>Allow All Sensors to be Disabled</strong></td>
<td>Allows the user to disable all the sensors within a simulation.</td>
</tr>
<tr>
<td><strong>Bounding Box Precision</strong></td>
<td>This field enables the user to specify the precision value of the bounding box for objects.</td>
</tr>
<tr>
<td><strong>Field Compression</strong></td>
<td>This value specifies the level of compression applied to the stored EM field data. A low compression results in faster loading of the data, however, the amount of storage space might increase. A high compression has the opposite effect.</td>
</tr>
<tr>
<td><strong>LF Gridding Preferences</strong></td>
<td></td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>-------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Max Step Multiplier</strong></td>
<td>Enable the user to specify how the interactive grid engine computes the max step for</td>
</tr>
<tr>
<td></td>
<td>low frequency simulations (the max step is computed by multiplying the maximal extent</td>
</tr>
<tr>
<td></td>
<td>of the model domain with the <strong>Max Step Multiplier</strong>.</td>
</tr>
<tr>
<td><strong>Base Res Multiplier</strong></td>
<td>Enable the user to specify how the interactive grid engine computes the baseline</td>
</tr>
<tr>
<td></td>
<td>resolution value for low frequency simulations (the baseline resolution is computed</td>
</tr>
<tr>
<td></td>
<td>by multiplying the maximal extent of the model domain with the <strong>Base Res Multiplier</strong>.</td>
</tr>
<tr>
<td><strong>Padding Multiplier</strong></td>
<td>Enable the user to specify how the interactive grid engine computes the padding for</td>
</tr>
<tr>
<td></td>
<td>low frequency simulations (the padding is computed by multiplying the maximal extent</td>
</tr>
<tr>
<td></td>
<td>of the model domain with the <strong>Padding Multiplier</strong>.</td>
</tr>
<tr>
<td><strong>Local Scale Multiplier</strong></td>
<td>Enable the user to specify the default local scale factor for solid regions for the</td>
</tr>
<tr>
<td></td>
<td>interactive grid engine for low frequency simulations.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Optimizer</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Show Non-Unicode Characters Info Dialog</strong></td>
<td>Enable the user to specify whether to show the info dialog regarding non-unicode support when using the optimizer. This option can also be disabled by activating the check box <strong>Do not show again</strong> on the info dialog itself.</td>
</tr>
<tr>
<td><strong>Using Non-Unicode Characters in File Paths</strong></td>
<td>Enable the user to specify whether to activate support for non-unicode characters when using the optimizer.</td>
</tr>
<tr>
<td><strong>Optimizer Default Folder</strong></td>
<td>Specifies the default folder used by the optimizer when support for non-unicode characters is enabled.</td>
</tr>
</tbody>
</table>
Chapter 5

Electromagnetic Simulation

Once the model, sources and sensors have been added to the project you are ready to set up and run the simulation. Click on the Simulations tab in the project pane. SEMCAD-X projects can have multiple runs for the same model, e.g. Harmonic and Broadband, to compare the effect of different Grid resolutions or to evaluate the effect of different Materials.

A new Simulation can be created by right clicking in the simulation tree and selecting New Simulation and then either Broadband or Harmonic. Alternatively, an existing simulation (including settings, grid, etc.) can be Copied and Pasted.

The remainder of this chapter describes in sequence the functionality and features for the different simulation settings that can be entered in SEMCAD-X before running a simulation or a batch of simulations.

Tip: All the settings described in this chapter are also summarized directly in the comment bar at the bottom left corner of the GUI when any of the setting dialogs is selected as shown in Figure 5.1.

5.1 Simulation Settings

Select Settings for the relevant simulation. The following parameters can be specified for the simulation:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Excitation</td>
<td>Name of the simulation.</td>
</tr>
<tr>
<td>Harmonic</td>
<td>A sinusoidal waveform is used as excitation of the simulation.</td>
</tr>
<tr>
<td>Broadband</td>
<td>A Gaussian sine waveform is used as excitation of the simulation.</td>
</tr>
</tbody>
</table>

Figure 5.1: Example of comment bar information for simulation settings
### 5.1. Simulation Settings

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Frequency</strong></td>
<td>This is the main frequency component of the sinusoidal waveform. This is the center frequency of operation.</td>
</tr>
<tr>
<td><strong>Bandwidth</strong></td>
<td>For <strong>Broadband</strong> simulations the bandwidth can also be specified for the excitation such that the excitation waveform will have a frequency spectrum from ((\text{Frequency} - \frac{\text{Bandwidth}}{2})) to ((\text{Frequency} + \frac{\text{Bandwidth}}{2})). In <strong>Broadband</strong> simulation mode it is possible to specify a customized signal profile by checking this checkbox.</td>
</tr>
<tr>
<td><strong>User Defined Signal</strong></td>
<td>User Defined Signal Input File An ASCI .txt file containing the data for the signal. The file should be formatted with signal time and the signal amplitude on each new line (see Figure 5.2). Any line not starting with a number is ignored.</td>
</tr>
<tr>
<td><strong>Simulation Time</strong></td>
<td>The duration of the simulation excitation waveform.</td>
</tr>
<tr>
<td><strong>Simulation Time Units</strong></td>
<td>The unit for the duration of the simulation. The unit can be chosen as <strong>Time Steps</strong>, <strong>Periods</strong> or <strong>Seconds</strong>. For <strong>Broadband</strong> simulations, the center Frequency is used for the Periods.</td>
</tr>
<tr>
<td><strong>Automatic Termination of</strong></td>
<td>Activates Autostop feature (see Section 5.10)</td>
</tr>
<tr>
<td><strong>Monitoring of Autostop</strong></td>
<td>Activates monitoring of all Autostop sensors during the simulation (see Section 5.11)</td>
</tr>
<tr>
<td><strong>Extract Frequencies</strong></td>
<td>Specify frequencies to record Harmonic data from <strong>Broadband</strong> simulations. These frequencies can be entered as separate entities, or as a range of frequencies.</td>
</tr>
<tr>
<td><strong>Solver</strong></td>
<td>Standard FDTD solver (see Section 2.1). ADI-FDTD solver (see Section 2.3). Conformal FDTD solver (see Section 2.2). Conformal ADI-FDTD solver (see Section 2.3). Low frequency simulations (see Section 2.9).</td>
</tr>
<tr>
<td><strong>aXware</strong></td>
<td>Run simulation using hardware accelerator card. Only supported for FDTD and C-FDTD solvers. Electro static simulation. Electro quasi static simulation where the ohmic current dominates the displacement current.</td>
</tr>
<tr>
<td><strong>Low Frequency Solver Type</strong></td>
<td>Electro Static Electro Quasi Static, Ohmic Current Dominated</td>
</tr>
<tr>
<td><strong>Optimization</strong></td>
<td>Choose between Speed or Memory for optimizing the performance the ADI solver. Choose between Speed or Accuracy for optimizing the performance the ADI solver. This setting determines the choice of the Time Step Factor, which can be seen when in the simulation output log when the simulation is started.</td>
</tr>
<tr>
<td><strong>Speed vs. Accuracy</strong></td>
<td>Conformal FDTD</td>
</tr>
<tr>
<td><strong>Time Step Factor</strong></td>
<td>Ratio of the time step of the simulation and the CFL criteria for the FDTD simulation. This ratio must be (\leq 1) to ensure stability.</td>
</tr>
</tbody>
</table>

---

A ratio of $\geq 14$ for electrically over-discretized models can lead to an acceptable error and competitive calculation times compared to the FDTD Solver.

When checked the user can specify custom Input and Output files for the simulations.

The user can keep notes explaining the details related to the specific model and simulation settings.

The simulation ID can be seen in this panel.

Figure 5.2: Example of a User Defined Signal Input File (left) for a modulated sinusoidal wave and the signal scope view of the signal

**Note:** When using a User Defined Signal the user should take care that the correct Frequency and Bandwidth are entered to ensure that an accurate grid is generated, i.e. if the signal contains frequency components much higher than those specified the grid will be too coarse which will result in inaccurate results and possible divergence.

### 5.2 Choosing the Appropriate EM Solver

**SEMCAD-X** offers a set of four different solver types, all of which have specific performance advantages adapted to a particular setup or simulation:

- FDTD
- FIT/C-FDTD
- ADI-FDTD
- FIT/C-ADI-FDTD
- Low Frequency Solver

**Tip:** aXware option: Both FDTD and FIT/C-FDTD can be run using the hardware acceleration option, but currently neither of the ADI schemes support it.

This section briefly outlines each solver’s capability and application range:
5.2. Choosing the Appropriate EM Solver

5.2.1  FDTD
- very robust general purpose solver
- well suited in the RF frequency range and above, or for grid resolutions of about $\lambda/50'000$
- suited for most applications involving largely inhomogeneous configurations
- features around 20 MCells/s on a P4 3.6 GHz (UPML ABC) computation speed
- not so well suited for very low frequencies

5.2.2  ADI-FDTD
- semi-implicit FDTD formulation allowing the time step determined by the CFL criterion to be overcome
- well suited for highly over discretized (very small spatial step compared to wavelength) simulation setups, as well as for simulations at very low frequencies
- typical time steps which can be chosen are in the range of 50 to 100 times the CFL time step or even more
- however, due to the semi-implicit scheme, the solver speed itself is about 6 times slower than the FDTD solver, thus an appropriate overtuning of the CFL time step should be chosen to achieve the desired effect
- the accuracy decreases the more the CFL time step is overtuned, however, this is highly case-dependent and may result in local effects only

5.2.3  FIT/C-FDTD
- the Conformal FDTD solver allows more geometrical details than the conventional FDTD solver to be taken into account
- the ability to use a conformal mesh with coarser spatial resolution that nonetheless produces the same accuracy as a fine staircasing mesh, results in remarkable savings in memory requirements (fewer cells) and simulation time (larger time step)
- different algorithms are applied for dielectric and perfect electric conducting (PEC) materials
- in particular, modeling of PEC materials which are curved or non conformally aligned to the FDTD grid is thus enhanced
- well suited for a broad variety of structures, mainly for PEC and dielectric objects which are oriented in the grid in a non conformal way, furthermore, certain structures can be modeled using Conformal FDTD which would be represented in conventional modeling with too coarse a resolution
- for PEC structures, the CFL time step might have to be slightly reduced (slightly longer computation time) for increased accuracy
- the algorithm is not yet suited for very thin PEC sheets, this will be added in a subsequent version
- source regions must be represented with two grid lines
5.2.4 FIT/C-ADI-FDTD

- conformal ADI-FDTD combines the benefits of staircased ADI with conformal FDTD
- application is best suited to highly over-discretized non-conformally aligned structures

5.2.5 aXware/ Cluster-In-A-Box Options

- high speed performance FDTD solver based on FPGA hardware acceleration
- up to 20 times (card V2.0) or more than 50 times (Cluster-In-A-Box) faster than the software solver
- well suited for a very broad range of applications due to the exceptional solver speed
- features around 400 MCells/s (card V2.0), around 800 MCells/s (Cluster-In-A-Box 800) and around 1500 MCells/s (Cluster-In-A-Box Devastator) computation speed
- computation speed is correlated with simulation size (number of cells) to be run on the card memory itself (1 GB for V2.0 card, i.e., about 28 MCells, or 128 MCells for Devastator), larger simulations will be shared between card memory and RAM on the motherboard
- Currently, most features are supported using aXware including FDTD, FIT/C-FDTD, Thin Resistive Sheet, Dispersive materials, etc.

5.2.6 Low Frequency Solver

In general, the low frequency solver is used for highly over-discretized simulations with a computational domain of a small fraction of the wavelength ($\lambda \gg diag(\Omega)$). Also consider theory in section 2.9. The simulation is performed in frequency domain.

Electro Quasi-Static Solver

- valid if the induced currents are negligible: skin depth $\delta$ of each dielectric is much larger than the computational domain ($\delta \gg diag(\Omega)$).
- in general, use model Electro Quasi Static.
- if $\omega \varepsilon \ll \sigma_E$ for each dielectric, use model Electro Quasi Static, Ohmic Current Dominated.
- if $\omega \varepsilon \gg \sigma_E$ for each dielectric, use model Electro Static.

Magneto Quasi-Static Solver

- valid if the displacement currents are negligible: $\omega \varepsilon \ll \sigma_E$ for each dielectric
- use model Magneto Quasi Static (not yet available in SEMCAD-X V13.0 Bernina).
5.3 Solid Region Settings

Once the relevant simulation Settings have been entered, select \textit{Solid Regions} and select the relevant solid in the table. The following parameters can be defined for the solids in the model:

<table>
<thead>
<tr>
<th>Region Mask</th>
<th>The solid is completely ignored for gridding, voxelbing and simulation. It can still be used as a target during optimization and region masks can be used as a target volume that various postprocessing methods can use for analysis.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the material or tissue.</td>
</tr>
<tr>
<td>Frequency</td>
<td>The frequency at which the specified parameters are valid.</td>
</tr>
<tr>
<td>Rel. Permittivity</td>
<td>The relative permittivity $\epsilon_r$ for the material.</td>
</tr>
<tr>
<td>Rel. Permeability</td>
<td>The relative permeability $\mu_r$ for the material.</td>
</tr>
<tr>
<td>Electric Conductivity</td>
<td>The electrical conductivity $\sigma_e$ for the material.</td>
</tr>
<tr>
<td>Magnetic Conductivity</td>
<td>The magnetic conductivity $\sigma_h$ for the material.</td>
</tr>
<tr>
<td>Density</td>
<td>The density of the material, which is used, e.g. in the calculation of SAR.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PEC/Metal Region Characteristics</th>
<th>Solids can be set as Volume, Surface, or Wire. See Section 7.4 for a detailed description.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thin Resistive Sheet</td>
<td>Enhanced treatment for thin metallic sheets (Section 2.7). The Rel. Permittivity $\epsilon_r$, electrical conductivity $\sigma_e$ and Sheet Thickness can be specified.</td>
</tr>
<tr>
<td>Lossy Metal</td>
<td>Enables a surface impedance to be set for the solid. This is specified in terms of Rel. Permittivity $\epsilon_r$, Rel. Permeability, Electrical Conductivity $\sigma_e$ and Magnetic Conductivity.</td>
</tr>
<tr>
<td>Potential Amplitude</td>
<td>Low frequency solver: specify the potential value of this metal object</td>
</tr>
<tr>
<td>Potential Phase</td>
<td>Low frequency solver: specify the phase of the potential, when using a complex valued low frequency solver (else it is ignored)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dispersive</th>
<th>Allows frequency dependant material parameters to be specified. See Section 2.4 for further details.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metamaterial</td>
<td>Allows frequency and intensity (third-order susceptibility) dependant material parameters to be specified. See Section 2.5 for further details.</td>
</tr>
</tbody>
</table>
Note: Lossy Metals can be used in harmonic or broadband (or transient) simulations and can use software or aXware accelerated solvers. The user need just specify the aXware switch in the Simulation Settings.

Local grid settings can also be specified for solid regions in the Grid Regions List, which is described in detail in Section 7.2.5.

### 5.4 Materials Database

**SEMCAD-X** also offers an integrated Material Database for importing, storing and assigning dielectric material parameters to single or multiple Solid Regions. A large number of frequently used materials including the commercially available ones are already stored in the database. Figure 5.3 shows the user interface which can be opened from the File | Materials menu. The database can be sorted according to material name, frequency or material type by clicking on table headings Material, Frequency or Notes, respectively. The interface allows the user to:

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New</td>
<td>Create a new material type to enter into the database.</td>
</tr>
<tr>
<td>Copy</td>
<td>Copy the selected material parameters from the database. Once the material has been Copied select the relevant Solid Region in the region list and Paste (CTRL+V) the parameters.</td>
</tr>
<tr>
<td>Paste</td>
<td>Paste the copied material with parameters to the database.</td>
</tr>
<tr>
<td>Save</td>
<td>Save the database to disk.</td>
</tr>
<tr>
<td>Import</td>
<td>Import a material database, e.g. the Gabriel parametric model database can be imported for anatomical material parameters.</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete the selected material from the database.</td>
</tr>
</tbody>
</table>

![Figure 5.3: Materials database user interface](image-url)
5.5. **Source, Lumped Element and Sensor Settings**

**Tip:** Common thermal tissue parameters are now also provided when importing the Gabriel parametric model database.

The user can append new materials to Material Database which then can then be reused in different simulations.

**Tip:** It is possible to assign material parameters to multiple Solid Regions. First Copy the material parameters either from the Materials Database or for a solid in the solid region list. Then multi-select (using SHIFT or CTRL keys) the solids to assign the material in the solid region list and Paste the material parameters.

**Note:** Dielectric material parameters can also be set for the Background material.

When simulating projects with many dielectric solids present (especially human tissue and human phantoms) and at many different frequencies, the assigning of material parameters can become tiresome. **SEMCAD-X** now includes a feature linked with the Materials Database that allows the easy allocation of these parameters. Material parameters can automatically be brought into the model from the Materials Database if the user names the solids in the model exactly as it appears in the Materials Database or puts that name into the Name field of the solid. Then the user can right-click on Solid Regions in the project tree and select Get Materials. This will copy all the properties from the corresponding entries in the Materials Database. If multiple entries are present in the Materials Database that have the same name, the one is selected that has the same frequency as the simulation. Alternatively Get Materials is available as well for an individual solid in the solids list. If activated from there the assignment is performed only for this solid and not for all the solids in the simulation. The matching is case sensitive.

Figure 5.4 shows a screenshot of this process.

### 5.5 Source, Lumped Element and Sensor Settings

**Note:** Every simulation must have at least one source and at least one sensor; otherwise, the simulation cannot be run.

Once the relevant material parameters have been entered, select Sources to enter the parameters for the excitation of the simulation:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>On</strong></td>
<td>Used to turn the selected Source on or off for the simulation when the model contains multiple sources.</td>
</tr>
<tr>
<td><strong>Edge Source</strong></td>
<td>Voltage, Current or Hard Source. For the Voltage Source an internal Resistance can be specified. See Section 6.2.1 for definitions.</td>
</tr>
<tr>
<td><strong>Plane Wave Source</strong></td>
<td>The incident angle of the plane wave is defined with the angles Theta, Phi and Psi. See Section 6.2.2 for further definitions.</td>
</tr>
</tbody>
</table>
Waveguide Source Mode Rectangular TE, Rectangular TM, Circular TE, Circular TM, Coaxial TEM, User Defined and Parallel Plate. The waveguide mode is defined with mode numbers m and n for rectangular and circular waveguide. The user must also specify Polarization Angle for circular waveguides. Similarly, the Polarization Direction must be specified in parallel plate waveguide source settings. For coaxial waveguide, Inner Radius must be entered. See Section 6.2.3 for further definitions.

Harmonic simulation Ramped Periods Number of periods used to ramp the envelope of the sinusoidal excitation waveform. Delayed Periods The number of periods to delay the excitation waveform.

Broadband simulation Amplitude Amplitude of the excitation wave form. Depending on the type of the source, the Amplitude will have different units. Time Shift The time delay for the excitation waveform. Phase Shift The phase delay for the excitation waveform.

When Sources is selected, the Scope window is opened showing the time profile for the excitation signal (see Figure 5.2).

Tip: The Scope is useful to help determine how long to run the simulation for, especially in Broadband mode, as the decay of the pulsed waveform can be displayed.

If the simulation has Lumped Elements in the model, select Lumped Elements to enter the resistive R, inductive L or capacitive C values:

Type
Resistor A resistive R circuit element.
Capacitor A capacitive C circuit element.
Inductor An inductive L element.
Resistor Parallel to Capacitor A parallel resistive R and capacitive C circuit network.

Now select the Sensors to specify what data will be recorded for the simulation:

Edge and Port Sensor Records time and frequency waveforms (voltage, current, impedance, power, etc.) along the edge of the cell in the grid.
Mode The Mode is synchronized to the simulation Mode; it is not necessary to modify it for this sensor.
Use for Autostop Consider this sensor when testing for simulation autostop (see Section 5.10)
Autostop Threshold Weak, Medium or Strict tolerance for autostop test condition.
Tolerance
### Field Sensor

Records electromagnetic (E, H, D, J, etc.) near-field data.

**On**

Turn the sensor on or off for the simulation.

**Mode**

- **Harmonic** mode, the field distribution is recorded and saved at the excitation Frequency of the simulation (e.g. $E(x,y,z,f_0)$).
- **Broadband** mode the field distribution is recorded for multiple frequencies specified in the Extract Frequency list.
- **Sampling** mode the field is recorded and saved as a function of time (e.g. $E(x,y,z,t)$). The starting and finishing points, as well as the number of samples can be specified by the user.

<table>
<thead>
<tr>
<th><strong>Record E Field</strong></th>
<th>When this box is checked, the E-field will be recorded.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Record H Field</strong></td>
<td>When this box is checked, the H-field will be recorded.</td>
</tr>
</tbody>
</table>

**On-The-Fly Fourier Transformation**

The Fourier transformation is performed on-the-fly for all frequencies specified in the simulation settings Extracted Frequencies without storing the time domain information. The simulation generally performs faster, but requires a lot more memory, depending on the sensor size.

**Overall Field**

The same as a Field Sensor but it records the fields throughout the whole grid.

### Far Field Sensor

Records near-field data and applies a near-to-far-field transformation to extract the far-field.

**On**

Turn the sensor on or off for the simulation.

**Mode**

- **Harmonic** mode the far-field is recorded at the excitation Frequency of the simulation.
- **Broadband** mode the far-field is recorded for multiple frequencies specified in the Extract Frequency list.

### Radar Cross-Section

Records E- and H-field, calculates the bistatic radar cross section.

**On**

Turn the sensor on or off for the simulation.

**Mode**

- **Harmonic** mode the radar cross section is recorded at the excitation Frequency of the simulation.
- **Broadband** mode the radar cross section is recorded for multiple frequencies specified in the Extract Frequency list.
The material dielectric properties can be copied through from the Materials Database.

**Note:** Using Field, Far Field or RCS sensors for Broadband simulations will result in larger result files saved on the disk. It is recommended that this functionality only be used with 2D or small-sized 3D Field Sensors.

**Tip:** Reducing the size of the result file to save disk space can be achieved by only selecting either the E- or H-field for Field Sensors, e.g. if a sensor is added to extract SAR related data, then it is not necessary to record the H-field.

Localized grid properties can also be specified for Sources, Lumped Elements and Sensors, as described in detail in Sections 7.2.5.

### 5.6 Boundary Conditions

Select **Boundaries** to select and modify the relevant boundaries to terminate the grid for the simulation (see Section 2.6 for detailed definitions):

| Mode  | Analytical | MUR absorbing boundary conditions ABCs are well suited to absorption of fairly uniform incident fields with lower computational requirements.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>UPML</strong></td>
<td>Uniaxial perfectly matched layer ABCs are well suited to absorption of highly nonuniform fields. Although the computational requirements are higher, the grid boundaries can be placed much closer to the radiating structure.</td>
</tr>
</tbody>
</table>
5.7 Grid and Voxel Settings

<table>
<thead>
<tr>
<th>Absorption level</th>
<th>Can be set to Low, Medium, High or Very High. The optimal number of absorbing UPML layers is only calculated during the simulation. Alternatively, selecting User-defined will allow the user to specify the exact number of layers and customize other options.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>ABC: Absorbing boundary condition, maps the FDTD computational domain to infinity. PEC: Electrical boundary condition. PMC: Magnetic boundary condition. Periodic: Periodic boundary condition.</td>
</tr>
</tbody>
</table>

Tip: When using UPML ABCs, the user can set different absorption levels based on the different boundaries, e.g. for a directional antenna use higher absorption on the boundary where the majority of the energy is directed.

5.7 Grid and Voxel Settings

Now select the Grid. A detailed description related to setting up the Grid and the different parameters can be found in Chapter 7. For most simple applications the default grid parameters are sufficient, especially if the Use Wavelength Units mode is used.

- Using the Interactive Grid Generator the grid will be automatically created when the Grid is selected.
- If the grid is not fine enough then modify the relevant regions using different Scale Factors, Modes, Refinement on boundaries/edges or Use local grid settings (see Section 7.2.2).
- Once a satisfactory grid has been created, then right click on and select Make Voxels.
- Right-click on the and select View Voxels to view the voxelled representation of the model.

5.8 Running a Simulation

Once all the simulation parameters have been entered and the grid and voxels have been made the simulation is ready to be run:

- Click on or right-click on the simulation and select Run.
- If there are multiple simulations to be run for the project click and select the simulations to run.
- Simulations from different projects can be run either using the Generate Simulation Batch File tool (Section 11.30), directly from the DOS command line or a batch file (Appendix B).

After the simulation is complete, the recorded results will be available for extraction and further post-processing (see Section 8).

5.8.1 SEMCAD X Solvers for 64 bit Systems

SEMCAD-X also provides native GUI and solvers for 64 bit systems. This allows the memory limit (and number of cells) given by 32 bit MS Windows or Linux systems to be overcome. Using these systems, huge simulation sizes can be addressed comprised of 1 billion mesh cells or more.
5.9 Remote Simulation Tools

The SEMCAD-X Windows Remote Solver allows users to execute simulations remotely. It offers two applications for managing remote simulation operations, allowing the interconnection among different computers with valid licenses. The tool consists of two applications: a SEMX Networking Server and SEMX Networking Client. The server application enables the use of a computer as a solver server, so that users can launch and queue simulations remotely. The client application allows connection to the available solver servers, and indicates the status of the sent simulation. Both applications are found in the WinRemoteSolver which is found in the folder where SEMCAD-X is installed. This chapter is a guide to configure both the server and the client programs in the Windows operating system.

5.9.1 Remote Solver Server Operation

The SEMX Networking Server application is the application that runs on the server computer, i.e., the computer where the simulations will be run. Once this application is started, the computer is turned into a simulation server accessible from any client computer within the same internal network (intranet of the company).

To set up the SEMCAD-X Remote Solver Server, follow these steps:

1. Start the SEMCAD-X Remote Solver Server by clicking on SEMCAD Networking Server (Figure 5.6).
2. Set the path to SEMCAD-X’s numerical solver isolve.exe, by pressing the button iSolve Path on SEMCAD Networking Server window (Figure 5.7). This action specifies which solver must be used for remote simulations.
3. Next, set the path where the simulation files will be saved by pressing the button Files Path.
4. Finally, enable the server by pressing the Open Networking button.

At this stage, the server starts broadcasting its availability in the LAN to run remote simulations. The SEMCAD Networking Server window shows the server queue. It indicates the name of the connecting...
5.9. Remote Simulation Tools

Figure 5.6: Starting the SEMCAD-X Remote Solver Server

Figure 5.7: SEMCAD-X Remote Solver Server window

machines, the IP address and the file name to be executed by SEMCAD-X’s numerical solver on the server side.

The button Kill Selected Job eliminates a simulation request from the server queue. It is worth noting that it does not interrupt execution of a simulation in the numerical solver. You can stop the networking service by clicking on Close Networking. The remote server networking can be restarted by clicking Open Networking. Finally, clicking on Quit will terminate the application.

**Note:** When simulations are run on the server a copy of both the Input and Output files are kept on the server in the folder specified by the user. These files can be deleted from time to time to free disk space on the server.

**Tip:** When pressing Open Networking, it is possible that a windows message appears asking you permission to open a specific socket. You must answer Unblock.

5.9.2 Remote Solver Client Operation

The client application can be launched from any Windows machine to connect to a previously set up server available in the local LAN. This application allows simulations to be remotely sent from
the client computer to one or more server computers for simulation. To connect to a Remote Solver Server, follow these steps:

1. Start the **SEMCAD-X** Remote Solver Client by clicking on **SEMCAD Networking Client** (Figure 5.8)

2. Click on **Open Networking** on **SEMCAD Networking Client** window (Figure 5.9). Automatically, a list of available Remote Solver Servers appears.

The upper part of the console displays the name of the available servers and their IP address. The **Queue** field indicates how many simulations are waiting in a particular server. Finally, the field **CPU usage** indicates how heavily the processor is being used. This information is only available for Windows machines. Linux machines simply display **Linux CIB**. Setting up a Linux Remote Solver Server is outside of the scope of this document.

3. Choose a server from the server list by clicking on the field. Then click on **Add Job** and pick the input file you want to simulate. Repeat this operation with each file to be sent to the server queue. While a remote simulation is running information is displayed in the **Jobs List** window.

The "Jobs List" window indicates the server name where the simulation files are being queued, the location of the file along with its name on the client’s side, along with each file’s status and progress (Figure 5.10).

The possible status for a job is:

- **Connecting**: The client is attempting to connect with the chosen server.
- **Accepting connection**: The server has accepted the request and queued the request.
5.9. Remote Simulation Tools

Figure 5.10: **SEMCAD-X** Remote Solver Client window with jobs queued

- **Sending input file**: The request has reached the top of the queue and the client is in the process of sending the input data file to the server.
- **Waiting in the server**: The input file has been successfully sent to the server, and is waiting to be executed by the numerical solver.
- **Waiting for results**: The simulation is running on the server.
- **Sending outfile**: The server has finished executing the simulation and is sending back the output file. This file is placed in the same folder as the input file in the client’s side.
- **ok, done**: The requested job has finished.
- **canceled**: The request has been cancelled by the client.

The **Progress** field indicates the state of the simulation that is being executed.

In the example shown in Figure 5.10, two simulations have been cancelled by the user, one is waiting in the queue, and one is being executed. The simulation being executed shows a progress of 14%.

If an error occurs during simulation on the server side, its content is displayed on the status window.

**Note**: Before sending the **Input** file to the server, the simulation must 1) first be gridded and voxelled 2) the **Input** file must be written by right clicking on the simulation and choosing **Write Solver Input File**.

5.9.3 Remarks

- The server stores all the sent input and generated output files in the directory selected in **File Path**. These files are not erased by the application, and it is left to the responsibility of the server administrator the task to maintain the directory.
- If two files sent have the same name, the server backups the earlier sent file by adding a suffix generated with the system’s date. In that way, the simulation files can be organized by filename to keep track of the order in which they were sent to the server.
- The firewall in the client’s side can block a file transfer to the solver server. If you can not send a file to an enabled Remote Solver Server, check that port 48993 is open on the client’s side.
- If in the client’s side you have both a wireless connection and a cable Ethernet one (or for that matter, more than one network interface enabled) this setup will interfere with the connection to the Remote Solver Server. This is because the server uses a secure connection to the client. Close or disable the extra available interfaces before trying to connect again.
5.10 Automatic Simulation Termination

SEMCA-D-X supports automatic simulation termination for all available solvers. During simulation, Edge and Port Sensors are tested for convergence, i.e. in Harmonic simulation mode steady-state condition is tested, and for Broadband simulation the signal is tested for complete decay. If the conditions are met then the simulation will automatically stop.

However, the Simulation Time is still used as an absolute termination point for the simulation even if steady-state or signal decay has not been reached.

Whereas the default Simulation Time defined by the system is a good average value to reach steady-state, the user can increase this time values as soon as automatic simulation termination is activated.

Three different levels for convergence testing are available: Weak, Medium or Strict. For most applications, Medium will ensure an accurate result. For structures with resonant behaviour the level can be set to Weak to reduce the simulation time.

Note: For large domains, if the desired steady-state should be reached within the entire computational domain then Edge and Port Sensors can be modeled and placed in user-defined extremities of the domain. The autostop feature can then be enabled for that sensor.

If required the autostop can be deactivated either in the global simulation Settings or for individual Edge or Port Sensors.

5.11 Monitoring of Sensors and Manual Simulation Termination

The voltage (U(t)) and current (I(t)) waveforms for all Edge and Port Sensors, with Use for Autostop activated, can be monitored during the simulation. By default the Monitoring of Autostop Sensors checkbox will be activated in the simulation Settings. Once the simulation has been started the waveforms can be monitored by clicking the icon, which will open the graph shown in Figure 5.11. Switch between U(t), I(t) and all available sensors using the Next and Back buttons.

Figure 5.11: Real-time monitoring of voltage and current waveforms during the simulation
5.12 Novice and Expert User Modes

The simulation can be terminated by the user in two different ways:

- **End and Save Simulation**: if the user sees that the simulation has reached steady-state from monitoring the results the simulation can be terminated such that the results are saved.
- **Abort**: the simulations terminates immediately and the results are not saved.

5.12 Novice and Expert User Modes

In the GUI there are certain parameters and attributes that may require more advanced understanding of either the FDTD method or knowledge specifically related to **SEMCAD-X**. For this reason, two different user modes are available: **Novice** and **Expert** mode (File | Preference to switch modes).

In **Novice** mode **SEMCAD-X** takes care of all of these parameters automatically. But in **Expert** mode the user is given full control to modify additional settings, while simultaneously having the responsibility of handling the synchronization of the settings.

The two modes mostly affect how the simulation settings are locked. These depend on the status of the grid, voxels and results of the simulation. For example, when the grid has been computed, the settings for the solids are locked (in **Novice** mode) since the grid is dependant on the solid parameters. In **Expert** mode, the user is given the freedom to modify the solid parameters without resetting the grid.

5.13 Parallel iSolve Kernel

In addition, **SEMCAD-X** features a parallel version of the iSolve kernel for multi-processor systems on a shared memory basis. This parallel kernel is available for both 32-Bit and 64-Bit platform architectures. In order to activate the parallel kernel, the user can proceed as follows:

1. Go to Preferences... via File in the **SEMCAD-X** main menu.

2. Set the **Solver Executable iSolve** under **SEMCAD X Solvers** by clicking in the empty edit field space (see Chapter 4 for more details).

   **Note**: In order to change the **Solver Executable iSolve** field, the **User Level** must be set to **Expert**. Please see Section 5.12 or Chapter 4 for more details.

3. Select the iSolve parallel executable file **iSolveP.exe** which is located within the **Solvers/iSolve** folder in the **SEMCAD-X** installation directory (e.g., in Program Files/).

The simulation will now be distributed equally on all available processors within the machine.

**Note**: In order to switch back to single processor usage, the **iSolve.exe** file can be chosen by application of the same procedure. Alternatively, the solver path text in the edit field can be deleted to make **SEMCAD-X** choose the default solver path.

**Note**: The OpenMP instruction-based parallel solver performance is highly case-dependent. It could increase performance by up to 35% for two processors, however, in certain cases no significant speedup will be observed.
To make use of the parallel solver capabilities, please request an appropriate license via semcad-support@speag.com.
5.13. Parallel iSolve Kernel


Chapter 6

Solids, Sources and Sensors

The generation of geometrical solids, field sources and sensors is described in Chapter 3. This chapter shows how SEMCAD-X treats the materials of the solids and explains the concepts of sources and sensors for the excitation and recording of the fields. The actual parameters and settings for the materials, sensors and sources are described in Chapter 5.

6.1 Solids

SEMCAD-X distinguishes dielectric solids, dispersive materials, thin resistive sheets, and perfect electric and magnetic conductors. The characteristics of the different material types are described in the following sections.

6.1.1 Dielectric Solids

The properties of a dielectric solid are permittivity, permeability, electric and magnetic losses and density\(^1\). They can be individually defined for each simulation run as described in Section 5.3 and Section 5.4.

![Figure 6.1: Interpolation of material properties for electric field components at material interfaces.](image)

The discretized numerical model of a dielectric solid consists of grid cells or voxels which are used to derive the update coefficients for the FDTD algorithm. Inside the solid, the update coefficients are directly calculated from the material parameters (Section 2.1.1). However, if a voxel touches a

\(^1\)The density is used in the thermal simulation and for the SAR calculation in the post processor and has no influence on the other simulation results.
material boundary, average permittivities and permeabilities are calculated by weighting the material parameters with the cross sections of the voxels surrounding the edge on the boundary (Figure 6.1).

6.1.2 PEC/Metal

FDTD treats perfect electric conductors (PEC) as boundary conditions for the electromagnetic field. This means that the field components a PEC-solid contains are set to zero. Furthermore, for the subgroup of thin resistive sheets (TRS) (i.e., electrically thin structures with significant conductivity) SEMCAD-X offers an algorithm that models their influence without the need to resolve the sheet thickness with the grid. The algorithm supports arbitrarily tilted and curved shapes. The details and applicability of the algorithm can be found in Chapter 2.

6.1.3 PMC

Similar to the PEC-solid, perfect magnetic conductors (PMC) are treated as boundary conditions for the electromagnetic field. In other words, the field components a PMC-solid contains are set to zero in FDTD simulations.

6.1.4 Dispersive Materials

Dispersive materials are currently supported in SEMCAD-X using the aXware hardware acceleration card. Debye, Lorentz, Drude and the combination of Drude-Lorentz models are offered. The complex relative permittivities of the dispersive materials are defined for each method in Chapter 2. The solids with the dispersive materials are treated in the same way as the dielectric objects in terms of gridding and voxelling.

6.2 Sources

For the excitation of electromagnetic fields, SEMCAD-X offers various source structures, which can be placed into the model as described in Section 3.2.3. Like the solids, the sources are modeled independently of the grid discretization. The grid adapts itself to the sources in the model. The different source types are described in the following sections.

6.2.1 Edge Sources

An edge source applies an electric field strength on one single edge on the primary Yee grid. This type of source can be used to excite structures like wire dipoles if no particularly designed antenna feedpoint or connector is needed. An edge source can be placed into the model using the button on the modeling toolbar.

The excitation of guided wave structures with an edge source is also possible. However, the fields have to travel along the guiding structure for a certain distance until the correct mode is established. This has to be considered when placing the sensors for the recording of the fields or signals.

In their CAD representation, edge sources have a finite length. If such a source is discretized with a mesh step size smaller than its actual dimensions, it will be extended to its original length using a perfectly conducting filament. This, however, can be avoided by choosing the discretization at the location of the source according to its length.
The FDTD kernel of **SEMCAD-X** yields three different ways of edge excitations which are described below.

### Hard Source

The hard source directly assigns a value defined by the excitation signal to the electrical field of the edge in the Yee grid. This means that the Yee algorithm does not affect grid edges which carry hard sources. As a consequence, all waves impinging on the source will be reflected. If you want to avoid these reflections, you should use a soft source.

### Voltage Source

A voltage source is an edge source with an internal resistance. This resistance takes energy out of the grid which would otherwise be reflected from a hard source. The advantage of the voltage source is that excitation pulses decay faster and steady state is reached after a shorter simulation time. The inner resistance of the voltage source should approximately match the input impedance of the source to make it absorb as much power as possible. However, it has no impact on the calculated feedpoint impedance.

### Current Source

If the expected line impedance is not resistive, the current source is an alternative to the voltage source. It is also an edge source but without any internal resistance.

#### 6.2.2 Plane Wave Excitation

In FDTD, the Total-Field Scattered-Field (TFSF) method is used for the excitation of a plane wave. Only a certain region of the calculation domain, the total-field region, propagates the plane wave. It must enclose all scatterers in the mesh. At the boundaries of the total-field region, the analytical solution of the incident plane wave is subtracted. The field caused by the scatterers inside the total-field region passes the boundaries and can propagate to the actual mesh truncation. A plane wave source can be placed into the model using the \( \text{button on the modeling toolbar.} \)

The plane wave source can be placed into lossy background.

Figure 6.2 show how the propagation angles \( \text{Theta, Phi and the polarization angle Psi can be chosen for incident E-field.} \)

#### 6.2.3 Waveguide Sources

**SEMCAD-X** offers a particular source for the excitation of different modes of different waveguide cross-sections. Rectangular, circular and coaxial waveguides can be excited by using the waveguide source. In addition, it is possible to implement a user-defined waveguide source. The source can be added to the model using the \( \text{button from the modeling toolbar.} \) Alternatively, the user has to employ Waveguides tools under Tools menu in order to automatically build both the waveguide source and also the sensors which are necessary to extract S-parameters. Note that the waveguide openings have to be terminated with perfectly matched absorbing layers to extract S-parameters (see Chapter 10.1). Thus, the waveguide source has to be placed at certain distance from the opening of the waveguide to avoid strong unphysical reflections due to the termination.

Implementation of the waveguide source for different waveguide cross-sections and with user-defined amplitudes is explained below.
Rectangular Waveguide Mode

The hollow rectangular waveguide has a width \(a\) and height \(b\) as shown in Figure 6.3 and can propagate the transverse electric (TE) and the transverse magnetic (TM) modes. The source has a rectangular outline normal to the propagation direction. The outline must exactly agree with the inner rectangle of the waveguide cross section (see Section 3.5). TE\(_{mn}\) and TM\(_{mn}\) modes can be excited using edge sources at each edges of the source plane. The transversal field component of the chosen mode is launched by each edge source having appropriate amplitude to reproduce the correct mode. For TE modes, the entire electric field is in the transverse plane, which is perpendicular to the length of the waveguide. For TM modes, the entire magnetic field is in the transverse plane. The convention for the mode numbers in the rectangular waveguides is as follows: the first subscript indicates the number of half-wave patterns in the \(a\) dimension, and the second subscript indicates the number of half-wave patterns in the \(b\) dimension.

Waveguide port (source) implementation requires three field sensors to be built in order to extract S-parameters from the field values recorded on these sensors. The Rectangular Waveguide (see Section refwaveguidetool) tool enables the user to automatically build the rectangular waveguide source and its associated sensors for given waveguide and source/sensor parameters. The user has to set the direction of the propagation (Direction), the distance between source and first sensor (Source-Sensor Distance), and the distance between the sensors (Sensor Spacing). The sensors has to be placed not very close to the source in order to be able to record correct mode distribution. In this way, the excited field produced by the edge sources will propagate along the homogeneous guiding section for a certain distance and the correct mode shape will be established. The extraction of the S-parameters depends on the distance between the sensors. The shorter distance results in more accurate results but smaller grid size, i.e., longer simulation.
Fig. 6.4 illustrates how to build the source and sensors for a waveguide. The size of the source should correspond with the inner rectangle of the cross section of the guide which is 7.112mm x 3.556mm in this example. The center of the waveguide cross section is given by the point (0,0,5) for this example and will be used as a reference for source/sensors placement (see Center Point in Rectangular Waveguide tool dialog window (Section 11.48)).

Figure 6.4: Illustration of rectangular waveguide source modeling

Circular Waveguide Mode

The hollow metal tube of circular cross section supports TE and TM modes. The waveguide source in this case is a square which must be tangential to the inner circle of the circular waveguide cross section (see Section 3.5). The procedure to excite the circular waveguide modes is similar to the rectangular waveguide case. The general classification of TE and TM is true for both circular and rectangular waveguides. However, the subscripts have a different meaning in the circular waveguides. The first subscript indicates the number of full-wave patterns around the circumference of the waveguide whereas the second subscript indicates the number of half-wave patterns across the diameter. In addition, the user should specify the polarization angle which determines the E-field direction of the mode defined by the mode numbers (see Fig. 6.5).

Coaxial Line Mode

The coaxial line supports TEM mode in addition to TE and TM waveguide modes. The waveguide source is implemented as a square which must be tangential to the inner wall of the outer conductor.
Sources

Figure 6.5: Illustration of circular waveguide source modeling

of the coaxial line (see Section 3.5). The excitation of TE and TM modes are not implemented yet. TEM mode of coaxial cable is excited using edge sources at each edges of the source plane. The user has to specify the radius of inner conductor of the coaxial structure.

Note: In order to excite the correct modes in circular waveguides and coaxial lines, the waveguide sources have to be placed a certain distance away from the first discontinuity of the structure. In this way, the excited field produced by the edge sources will propagate along the homogeneous guiding section for a certain distance and the correct mode shape will be established before reaching the discontinuity.

Parallel Plate Mode

The parallel plate waveguides support TEM mode in addition to TE and TM modes like coaxial cables. The waveguide source is implemented as a rectangle which must cover the inner rectangle of waveguide cross section. The excitation of TE and TM modes are not implemented yet. TEM mode is excited using edge sources at each edges of the source plane. The polarization direction for the parallel plate waveguide has to be specified by the user (see Fig. 6.6).

Figure 6.6: Illustration of parallel plate waveguide source modeling

User-defined Waveguide Source

The User Defined mode of the Waveguide Source enables the user to excite the field with a user-defined cross-sectional distribution. The amplitudes and time-shifts of E- and H-field on the cross-section of the waveguide have to be recorded in ASCII files and their names and locations have to be entered. The amplitude and time-shift data given in the ASCII files have to be on a uniform grid in the range
from \((x_0,y_0) = (0,0)\) to \((x_n,y_m) = (1,1)\) and in the following format:

\[
\begin{align*}
% \text{Comment lines starting with \% sign} \\
% \text{Ex, Hx values} \\
Ax(x_0,y_0), Ax(x_0,y_1), \ldots, Ax(x_0,y_m) \\
Ax(x_1,y_0), Ax(x_1,y_1), \ldots, Ax(x_1,y_m) \\
\ldots \ldots \ldots \\
Ax(x_n,y_0), Ax(x_n,y_1), \ldots, Ax(x_n,y_m) \\
% \text{Ey, Hy values} \\
Ay(x_0,y_0), Ay(x_0,y_1), \ldots, Ay(x_0,y_r) \\
Ay(x_1,y_0), Ay(x_1,y_1), \ldots, Ay(x_1,y_r) \\
\ldots \ldots \ldots \\
Ay(xs,y_0), Ay(xs,y_1), \ldots, Ay(xs,y_r) \\
% \text{End}
\end{align*}
\]

## 6.3 Sensors

For the recording of field components, parameter extraction and data visualization, SEMCAD-X offers a variety of sensor types designed for various purposes. These sensor objects are part of your CAD model, which means that they automatically adapt their geometrical resolution to the grid which you use for the simulation. How to generate and position these sensors is described in Section 3.2.5. Most of the sensors can record either time domain quantities or extract frequency domain results.

### 6.3.1 Field Sensor

The Field Sensor is used to record E- and/or H-fields. The post processor (Chapter 8) can directly display these quantities or calculate D- and B-fields, respectively. If the sensor is used in quasi-harmonic mode, the complex Poynting-vector and the SAR distribution can be calculated as well. As described in Section 3.2.5, the recording region can be defined as a box or a surface. Please note that the storage space requirements of the recorded fields can increase tremendously if a large field sensor records in the time domain mode.

An Overall Field sensor is automatically added for all harmonic simulations, but disabled for broadband simulations, by default. This is done in order to limit the storage space required. The Overall Field is basically a field sensor which encloses the entire computational domain and records the E- and/or H-fields.

### 6.3.2 Edge Sensor

The purpose of the Edge Sensor is to record and extract parameters from an Edge Source (Section 6.2.1). In connection with this source, it provides feedpoint voltage, current, impedance and power. In a time-domain simulation, the signals can be evaluated both in time and frequency domain.

The sensor records the E-field along the edge on which it is positioned and calculates the voltage according to

\[
V = E \cdot \Delta s
\]

\(^2\)SAR calculation is only possible if the sensor is a box, i.e., if it encompasses a volume of at least one cell.
6.3. Sensors

with $\Delta s$ as the length of the edge in the mesh. The current is calculated from the H-field loop immediately around this edge using

$$I = \sum_k H_k \cdot \Delta s_k.$$ 

The integration path for the voltage and the loop for the current calculation are oriented in mathematically positive sense.

Note: The Edge Sensor is automatically added for each Edge Source implemented in the simulation.

6.3.3 Voltage Sensor

The Voltage Sensor records a voltage in time domain or as a phasor quantity on a straight line in the grid which can be defined by two points at arbitrary positions (Section 3.2.5). SEMCAD-X will automatically choose the shortest integration path which connects these two points in the grid. The voltage is then calculated according to:

$$V = \sum_k E_k \cdot \Delta s_k$$

with $\Delta s_k$ as the lengths of the edges assigned to the integration path.

6.3.4 Current Sensor

The Current Sensor records a current in time domain or as phasor quantity on an arbitrarily shaped closed loop. The loop can be defined by a set of points as described in Section 3.2.5. SEMCAD-X will automatically select edges in the secondary grid following the shape of the loop. The current is then calculated according to

$$I = \sum_k H_k \cdot \Delta s_k$$

with $\Delta s_k$ as the lengths of the edges assigned to the integration loop.

6.3.5 Port Sensor

The Port Sensor can be used in a similar manner as the Edge Sensor (Section 6.3.2). The main difference between these sensors is that the Port Sensor records the voltage along a path of arbitrary length along one of the mesh axes and the current in a rectangular loop. The orientation of the path and the loop can be changed in the property window of this sensor. Section 3.2.5 shows how to place this sensor into the model. The quantities that can be calculated are the same as those of the Edge Sensor.

6.3.6 Far-Field Sensor

The far-field sensor enclosing the antenna should have a minimum distance of two or three mesh cells from the boundaries of the computational domain. The near-to-far-field transformation which is
applied for the computation of the radiation pattern enables the placement of the far-field sensor at a short distance from the radiating structure. In order to enable a reliable calculation of the far-field pattern, it is very important to enclose all PEC as well as lossy dielectric structures which participate in the radiation and the absorption mechanism.

**Note:** The Far-Field Sensor is automatically added for simulations with one or more Edge Source.

### 6.3.7 Radar Cross-Section

The radar cross section sensor is used to calculate and record bistatic RCS parameters. The RCS sensor should have a minimum distance of two or three mesh cells from the boundaries of the computational domain.

**Note:** The Radar Cross-Section Sensor is automatically added for simulations with a plane wave excitation.

### 6.3.8 Thermo Box Sensor

The Thermo Box Sensor records the thermal data for all voxels enclosed in the sensor volume. It allows the user to specify the range of simulation time for the sensor to record values. The number of samples in this specified simulation time can also be set by the user.

### 6.3.9 Thermo Point Sensor

The Thermo Point Sensor is used to record the thermal data at a single point in the model. Similar to the thermo box sensor, it is also possible to specify the simulation time range to record the data and the number of samples distributed evenly over the specified simulation time. In a future release of SEMCAD-X, an auto-stop function will be added which tests the thermo point sensor for convergence.
6.3. Sensors
Chapter 7

Grid and Voxel Generation

The FDTD method requires that the model and the surrounding computational domain be spatially divided into cells, where the FDTD update equations are applied. Since SEMCAD-X supports a full 3D ACIS® based modeling environment, the models that are generated or imported are not restricted to a predefined grid. In order to generate a geometrically accurate numerical representation of the model, the grid generator creates a nonuniform rectilinear grid, which automatically adapts to the details of the model.

This chapter describes the functionality of the grid generator, voxelizing routines and the different features that are provided. Since the accuracy of any FDTD simulation depends on the quality of the grid that is used, it is recommended that the user pay careful attention to the various concepts that are described in this chapter.

7.1 Selecting a Grid Routine

Currently, two grid generation routines are available in SEMCAD-X namely the Interactive and Conventional Grid Generator. The Interactive gridder has been designed to be more robust and offer extensive interactive functionality. In order to be more robust, tolerances are introduce to the grid parameters (e.g. to the grading ratio) giving the gridder more flexibility. In addition, there is no absolute Min Step for the Interactive gridder. While the Conventional Grid Generator meets the users parameters more strictly, under certain conditions it may be unable to meet all the specified conditions and the Min. Step or Grading Ratio is violated.

<table>
<thead>
<tr>
<th>Gridder</th>
<th>Advantages</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interactive</td>
<td>Robust, fast refinement, additional interactive functionality.</td>
<td>Limited control over exact minimum grid step.</td>
</tr>
<tr>
<td>Conventional</td>
<td>Maintenance of Min. Step, better control over simulation time.</td>
<td>Max. Grading Ratio infringed when grid parameters incompatible with the Baseline spacing.</td>
</tr>
</tbody>
</table>

By default the Interactive Grid Generator is applied when the grid is generated. The default grid generator can be changed in the File | Preferences menu under the Preferred Grid Engine.

Tip: The grid generator can be selected for a simulation by right clicking on the Grid and selecting e.g. Switch to Conventional Gridder (see Figure 7.1).
7.2 Interactive Grid Generator

As the name suggests, the Interactive Grid Generator offers the user a highly interactive platform to easily generate and customize the grid. This gridder has been designed to generate high quality grids that follow the user specifications in real-time. Thus, any change in the grid settings is immediately reflected in the grid which is automatically updated.

7.2.1 Overview

The new features available in the Interactive Grid Generator can be summarized as follows:

- **Immediate Grid Generation:** The grid is automatically generated upon clicking on the Grid for the first time. Whenever the user modifies any grid settings the grid is automatically recalculated.

- **Grid Regions List:** A list of all objects that support grid settings, including all solids, sources and sensors in the model, is available when the Grid is selected (Figure 7.3).

- **Picker Tool:** A Picker tool can be activated from the menu item Pick, which enables interactive selection of different solids directly in the 3D model window (Section 7.2.2).

- **Scaling Sliders:** All items in the Grid Regions list carry scaling factors which multiply the grid settings of the selected item. Clicking and holding the right mouse button will activate the sliders to dynamically change the scaling factor (Section 7.2.2).

- **Additional Gridding Modes:** a new Geometrical mode is available which performs geometrical analysis on the solid to estimate Curvature and Thickness Resolutions of the structure (Section 7.2.6).

**Tip:** If a grid was originally configured using the Conventional grider the settings can be easily imported to the Interactive grider using the Import Settings from Conventional Gridder shown in the menu in Figure 7.1.
7.2.2 Generating and Refining the Grid

This section describes processes and functionality for setting up and refining a grid using the Interactive Grid Generator. For more technical explanations regarding the grid settings please consult the remaining sections of this chapter.

Figure 7.2 shows the Interactive Grid Generator in action. The approach for generating and refining the grid is described as followed:

- Enter all relevant parameters: Settings, Solid Regions, Sources, Sensors, Lumped Elements, and Boundaries.
- Select the Grid in the simulation settings: the Interactive Grid Generator is initialized and the default grid is immediately created.
- The default Mode and Reference Baseline Weight are assigned to all grid objects according to Section 7.2.6 and Section 7.2.7 respectively.

If the initial grid requires further refinements the following procedure is suggested:

- Select the specific solid where the grid requires refinement. This can be done in two ways: by selecting the solid in the Grid Regions list or by selecting it directly in 3D modeling window using the Picker tool, which is activated when the grid is selected.
- When the solid is selected the grid is shown only in the region of the solid (see Figure 7.2). To view the overall grid select the Background.
7.2. Interactive Grid Generator

- Select the most suitable Mode (Section 7.2.6) for the solid, e.g. Geometrical will automatically resolve detailed structures, but may result in an overly refined grid.

- If the grid still requires refinement for the solid use the Scaling Factors, Local Grid Settings or Refinement on Boundaries or a combination to obtain the required resolution.

- Set the Mode for any solids which are not significant for gridding to Not relevant for grid or Regional only (Section 7.2.6). This will reduce the number of Baselines and give the grid more flexibility to capture the most significant detail. This is especially true for CAD based models.

- For overlapping solid regions containing significant detail increasing the Reference Baseline Weight enables the prioritizing of one solids baselines above or below another (Section 7.2.7).

- If necessary, decrease the Baseline Resolution for a solid to resolve significant baselines which are located close to one another.

- Change local grid step size for specific solid regions change the Local Scale Factor or the Local Scale Factor. This can be done directly in the Local Settings or it can be done interactively using the mouse and slider bars (activated by clicking and holding the right mouse button).

Once a satisfactory grid has been achieved right-click on the icon and choose Make Voxels. Once the voxels have been created, right-click on the icon and select View Voxels. Check that the model is discretized as required.

Picker Functionality

In order to simplify and speedup the customization of a grid before a simulation interactive functionality is supported. Choosing Pick from the context menu activates the interactive picker. Clicking in the model window will now select the object underneath the cursor and a small window showing all overlapping objects underneath the cursor appears in the lower right hand corner (see Figure 7.2). In this mode, the mouse wheel will scroll through all these objects. Moving the mouse over the model will pick continuously while holding the left mouse button.

Grid Regions List

<table>
<thead>
<tr>
<th>Name</th>
<th>Mode</th>
<th>Time Step</th>
<th>Scale</th>
<th>Wavelength</th>
<th>No of Baselines</th>
</tr>
</thead>
<tbody>
<tr>
<td>Background</td>
<td>Regional ...</td>
<td>2.52591e-011</td>
<td>1</td>
<td>0.18737</td>
<td>0</td>
</tr>
<tr>
<td>Edge Source 1</td>
<td>Sounding ...</td>
<td>3.31642e-012</td>
<td>1</td>
<td>0.18737</td>
<td>6</td>
</tr>
<tr>
<td>antenna+PCB/PCB_L1</td>
<td>Sounding ...</td>
<td>6.31477e-012</td>
<td>0.25</td>
<td>0.18737</td>
<td>6</td>
</tr>
<tr>
<td>antenna+PCB/PCB_L2</td>
<td>Sounding ...</td>
<td>6.31477e-012</td>
<td>0.25</td>
<td>0.18737</td>
<td>6</td>
</tr>
<tr>
<td>antenna+PCB/PCB_dielect</td>
<td>Regional ...</td>
<td>2.52591e-011</td>
<td>1</td>
<td>0.0936851</td>
<td>0</td>
</tr>
<tr>
<td>antenna+PCB/dielectric support</td>
<td>Not Relav...</td>
<td>N/A</td>
<td>1</td>
<td>0.0936851</td>
<td>0</td>
</tr>
<tr>
<td>antenna+PCB/feed arm</td>
<td>Sounding ...</td>
<td>4.29814e-012</td>
<td>1</td>
<td>0.18737</td>
<td>6</td>
</tr>
<tr>
<td>antenna+PCB/foot</td>
<td>Sounding ...</td>
<td>5.53889e-012</td>
<td>1</td>
<td>0.18737</td>
<td>6</td>
</tr>
<tr>
<td>antenna+PCB/ground</td>
<td>Sounding ...</td>
<td>1.00070e-011</td>
<td>1</td>
<td>0.18737</td>
<td>6</td>
</tr>
<tr>
<td>antenna+PCB/vis/vis</td>
<td>Regional ...</td>
<td>2.52591e-011</td>
<td>1</td>
<td>0.18737</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 7.3: Grid Regions List: all the grid items can be easily sorted by the different fields.

The Grid Regions List shown in Figure 7.3 is a list of all items (solids, sources, sensors) to which local grid parameters can be assigned. The list contains additional fields for each item showing information
related to the Mode, the estimated Time Step for the simulation, the Local Scale Factor, the local Wavelength in that region and the Number of Baselines fields.

In addition the Grid Regions List is used together with the Picker tool to highlight which object has been selected in order to assign local grid parameters. Multiple selection in the list is also possible and is used either when copy paste of local grid settings or to set the Mode with shortcut keys.

Tip: Local grid settings can be copied from a single item and pasted either to single or multiple items in the Grid Regions List.

Slider Functionality

Once an item has been selected in the Grid Region List (either by selecting the object directly in the 3D model window with the Picker tool or by selecting the object directly in the list) the grid resolution in the region of the selected item can be increased or decreased using sliders, which adjust the Local and Axial Scaling Factors. Open the slider either by clicking and holding the right mouse button in the 3D model window or by selecting Refine [object name] from the menu (Figure 7.1). Right clicking on the slider will switch between the four different sliders.

- adjusts the Local Scale Factor between 0.01 and 10.
- adjusts the Axial Scale Factor for the X-Axis between 0.1 and 10.
- adjusts the Axial Scale Factor for the Y-Axis between 0.1 and 10.
- adjusts the Axial Scale Factor for the Z-Axis between 0.1 and 10.

7.2.3 Global Grid Settings

Figure 7.4 shows an overview of the Global Settings:

<table>
<thead>
<tr>
<th>Overall Scale Factor</th>
<th>A constant between 0.01 and 10 which multiplies the Max. Step and the Baseline Resolution for all grid entities for all three axes.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Padding</td>
<td>The spacing added to the grid from the bounding box of the model to the grid boundary (Section 7.2.4). In this Padding mode the size of the Padding is chosen automatically.</td>
</tr>
<tr>
<td>Automatic</td>
<td>The Padding size can be manually set in this mode. Negative Padding is also supported, e.g. enabling truncation of solids.</td>
</tr>
<tr>
<td>Manual</td>
<td>No Padding is added. The grid will be terminated on the boundary of the model.</td>
</tr>
<tr>
<td>Off</td>
<td>The Padding that is added on the side of the grid in the direction of the negative axis.</td>
</tr>
<tr>
<td>Padding Low</td>
<td>The Padding that is added to the side of the grid in the direction of the positive axis.</td>
</tr>
<tr>
<td>Padding High</td>
<td></td>
</tr>
<tr>
<td>Unit Length Type</td>
<td>Off</td>
</tr>
<tr>
<td></td>
<td>Wavelength</td>
</tr>
<tr>
<td></td>
<td>Custom</td>
</tr>
</tbody>
</table>
7.2. Interactive Grid Generator

<table>
<thead>
<tr>
<th>Gang Axes</th>
<th>Scale Factor</th>
<th>Baseline Resolution</th>
<th>Max Step</th>
<th>Grading Ratio</th>
<th>Grading Ratio Relaxation</th>
</tr>
</thead>
<tbody>
<tr>
<td>X, Y, Z Axis</td>
<td>A constant between 0.01 and 10 which multiplies the Max Step and the Baseline Resolution for all grid entities in the selected axial direction.</td>
<td>The smallest distance at which two Baselines can be resolved. If any two baselines are closer than this distance one will be ignored according to the Reference Baseline Weighting. Baselines with maximum weight of 2000 are always included.</td>
<td>The maximum grid step for the solid region in the selected axial direction.</td>
<td>The maximum ratio of the length of two neighbouring cells along an axial direction in the grid.</td>
<td>Specify the percentage of local relaxation of the grading ratio to increase the dynamics of the gridder and to prevent over-refinement in areas with closely spaced Baselines.</td>
</tr>
</tbody>
</table>

**Tip:** It is recommended to work with the gridder with the Unit Length Type set to Wavelength. In this mode the grid uses the absolute wavelength, i.e. the grid will automatically be refined in regions containing high dielectric permittivity and permeability.

The default parameter for the global Max. Step is chosen as \( \frac{\lambda}{14} \), which is slightly more conservative than the rule-of-thumb of approximately \( \frac{\lambda}{10} \). However, a more conservative size is chosen to reduce dispersion errors which are automatically introduced from using a nonuniform grid. The default value for the Baseline Resolution is chosen as \( \frac{\lambda}{500} \). This parameter depends largely on the level of detail within the model that should be resolved by the gridder. If the value is not chosen small enough, then it is possible that certain Baselines could be ignored (see Section 7.2.6). If the value is chosen too small then the grid is given more freedom to resolve closely spaced Baselines which may result in a very small minimum grid step and in turn can increase the simulation time due to the small times step.

**Tip:** The estimated time step is shown for each grid region in the Grid Region List. Increasing the Baseline Resolution for the solid with the smallest time step could resulting in an increase in the overall time step, which will reduce the simulation time.

### 7.2.4 Padding

There are three possible Padding modes, namely Automatic, Manual and Off. Since the padding size is optimized based on the simulation configuration, it is recommend that the Automatic mode to be used in most cases. The size is adjusted based on the type of ABC that is chosen: smaller padding is applied for UPML ABCs, while larger padding is applied for Analytical ABCs. Furthermore, using the Automatic mode will guarantee that the correct number of cells are added for the use of the Far Field Sensor.

The Manual mode enables the user to customize the Padding. For example, when simulating part of a microstrip structure it might be required for the padding to be zero for two of the axes and non-zero for the third. Also, this mode supports negative padding, for example if a grid is desired that is smaller than the model, such that the boundaries truncate the solids.
Off will result in zero padding for all axes. This can be used, e.g., when PEC ABCs are selected to simulate the inside of a PEC enclosure.

### 7.2.5 Local Grid Settings

If the grid that is generated using the default settings is not fine enough, there are various local grid settings which can be applied to individual solids to add further refinement to the grid. Figure 7.4 shows an overview of the Local Settings:

<table>
<thead>
<tr>
<th>Mode</th>
<th>Geometrical and baseline modes for each grid region (Section 7.2.6).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not Relevant For Grid</td>
<td>The solid is completely transparent in terms of the grid generation. It will still be voxelled and present in the simulation.</td>
</tr>
<tr>
<td>Regional Only</td>
<td>No Baselines are considered during grid generation. However, the Max. Step is enforced in the entire area.</td>
</tr>
<tr>
<td>Bounding Box</td>
<td>Baselines are generated on the bounding box of the solid and the Max. Step is enforced in the entire area.</td>
</tr>
<tr>
<td>Geometrical</td>
<td>A geometric analysis is performed and the grid is adapted accordingly.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Local Scale Factor</th>
<th>A constant between 0.01 and 10 which scales the local Max Step for the solid region for all three axes.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Use Local Settings</th>
<th>Set local Baseline Resolution and Max Step for the region.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gang Axes</td>
<td>Synchronizes the grid settings for all grid axes when checked.</td>
</tr>
</tbody>
</table>

Figure 7.4: Global and local grid settings for the Interactive Grid Generator.
7.2. Interactive Grid Generator

<table>
<thead>
<tr>
<th>Reference Baseline Weight</th>
<th>An integer between 0 and 2000, which is used to prioritize treatment of Baselines for different solids, i.e. 2000 has the highest priority and 0 has the lowest (Section 7.2.7).</th>
</tr>
</thead>
<tbody>
<tr>
<td>X, Y, Z Axis Axis Scale Factor</td>
<td>A constant between 0.1 and 10 which scales the local Max. Step in the selected axial direction.</td>
</tr>
<tr>
<td>Baseline Resolution</td>
<td>The smallest distance that 2 baselines can have. If any two baselines are closer than this distance one will be ignored according to the Reference Baseline Weighting. Baselines with maximum weight are always included.</td>
</tr>
<tr>
<td>Max Step</td>
<td>The maximum grid step for the solid region in the selected axial direction.</td>
</tr>
<tr>
<td>Refine on Boundary</td>
<td>Scales the Max Step of the grid at the boundary of the object.</td>
</tr>
<tr>
<td>Curvature Resolution</td>
<td>Scales the grid resolution in subregions containing curvature.</td>
</tr>
<tr>
<td>Thickness Resolution</td>
<td>Scales the grid resolution in subregions where the solid is thin.</td>
</tr>
</tbody>
</table>

Tip: Local grid settings can be copied from a single item and pasted either to single or multiple items in the Grid Regions List.

7.2.6 Grid Region Modes

Several different gridding modes are available, influencing how Baselines and subregions are treated for each region in the Grid Region list.

**Note:** A Baseline is a line generated from the geometry used as a reference for placing gridlines when the grid is generated. Baselines are shown in the grid as green lines. If two Baselines are close together than the Baseline Resolution then one of the Baselines will be ignored and therefore shown as a blue line.

Choosing a the relevant mode for each region can have a significant influence on the size of the total grid.

Tip: To change the Mode of single or multiple selected items in the Grid Region list use the following shortcut keys:

- n to change the Mode to Not Relevant For Grid
- r to change the Mode to Region Only
- b to change the Mode to Bounding Box
- g to change the Mode to Geometrical

**Not Relevant For Grid**

The solid is completely transparent in terms of the grid generation, i.e. no local grid settings and no Baselines are generated for the grid in this mode. The solid is still voxelled and simulated.
Tip: To make a solid completely transparent in terms of the simulation activate the Region Mask checkbox in the Solid Regions. The solid is also transparent for grid generation.

Regional Only

No Baselines are generated for the solid. However, the Max Step is enforced, Local and Axial Scaling Factors and Local Grid Settings can be applied to the solid.

Note: By default the Mode for all Dielectric and Dispersive solids is set to Regional Only.

Tip: For highly complex models containing many parts setting the Mode of less significant parts to Region Only will reduce the number of Baselines in the grid, giving the gridder more freedom to resolve the most significant detail of the structure.

Bounding Box

Baselines are generated at the bounding box of the solid. The Max Step is enforced, Local Grid Settings, Scaling Factors and Refine on Boundary can be applied to the solid. Furthermore, the Use Inner Baselines checkbox can be activated for the X-, Y- and Z-Axis to generate Baselines on all edges for the solid.

Note: By default the Mode for all PEC and PMC solids is set to Bounding Box.

Tip: Applying Use Inner Baselines on highly detailed CAD data can result in a large number of Baselines being generated for the specific solid. This will decrease the flexibility of the gridder and make the grid more difficult to refine.

Geometrical

In this mode a geometric analysis is performed on the solid to assess the thickness and curvature of the structure and the grid is adapted accordingly. Tangential Baselines are generated for all surfaces that lie in any of the Cartesian planes and prioritized according to the area of the surface, i.e. Baselines generated from larger surfaces will receive a larger internal Baseline weight. In addition to Local Grid Settings and Scaling Factors, the Curvature and Thickness Resolution can be set. These two parameters either increase or decrease the cell size for regions of the solid containing curved or narrow geometrical detail.

Tip: The geometrical mode is very useful for automatically resolving the grid for structures like an helical antenna (Figure 7.5) or a microstrip line. Due to the robustness of this mode it can result in large grids when it is used on detailed CAD data since the grid will try to resolve all the detail of the part. An alternative approach for highly detailed parts might be to use Bounding Box mode and reduce the Max Step.

Note: Geometrical Mode is not supported by Sources, Sensors and Lumped Elements.
7.2. Interactive Grid Generator

Figure 7.5: Using Geometrical Mode: the helix (left) the grid automatically resolves the curvature and thickness of the structure. For the coupler (right) the grid is automatically refined in the regions of high curvature of the ring.

7.2.7 Reference Baseline Weight

Baselines from each item in the Grid Regions List can be prioritized by changing the Reference Baseline Weight between 0 (lowest priority) and 2000 (highest priority). The Reference Baseline Weight is used specifically when the distance between two Baselines is closer than the Baseline Resolution specified by the user. The griddler must then decide which of the two Baselines to ignore and this then done according to the Reference Baseline Weight.

Note: When a Baselines is ignored because it is closer than the Baseline Resolution it is shown as blue lines in the grid.

This is especially useful where multiple solids overlap: by increasing the priority of the most significant parts it will guarantee that the Baselines for that part are resolved by the grid.

The following default Mode and Reference Baseline Weight settings are applied:
Sources and Lumped Elements: Mode: Bounding Box, weight = 2000;
Sensors: Mode: Bounding Box, weight = 500;
PEC/Metal and PMC: Mode: Bounding Box, weight = 1500;
Dielectric and Dispersive: Mode: Regional only, weight = 1250.

7.2.8 Grid Generation For Low Frequency Solver

For application where the Low Frequency Solver is applied, grid generation is no longer dependant on the wavelength. For this reason, when the Low Frequency Solver is chosen the Unit Length Type is automatically set to to Meter. The two main criteria for grid generation are: 1) resolve the geometry of the structure, and 2) ensure that the Padding is large enough that the field has decade to zero, such that that no reflections introduced back into the computational domain. Based on these two criteria and information related to the size of the bounding box of the all the structures in the simulation, the default grid settings are chosen:

1. The longest of the three sides of the bounding box containing the whole model is used as a unit length $L$
2. The Padding is calculated as $2 \times L$
3. The Max Step is calculated as $0.2 \times L$.
4. The Baseline Resolution is calculated as $0.005 \times L$.
5. The Local Scale Factor is set as 0.2.

**Tip:** The above settings can all be modified in the LF Gridding Preferences menu as shown in Figure 7.6.

![Figure 7.6: LF Grid Preference menu.](image)

**Note:** The Local Scale Factor is set to 0.2 for all solids in the model resulting in a grid which is coarse in the Padding and refined in the area of the structure. It is still possible to further modify the grid by changing the Local Scale Factor, Mode, etc. to till is sufficiently resolves the structure.

### 7.2.9 Advanced Grid Editor (Preview Version)

If the Advanced Grid Editor has been enabled in the preferences (see section Advanced Gridding in the Preferences dialog) before loading the simulation, the subitem Advanced (Preview) in the simulation tree is visible (see figure 7.7). By clicking on it, the advanced gridding mode is enabled. This mode enables the user to set and delete custom baselines as well as inspect the grid lines in detail. The advanced gridding mode shows a list of the grid lines of the currently highlighted axis as well as a window with detailed properties if a customizeable grid line is highlighted.

**Note:** The advanced grid editor is available as a preview version only.

### Highlighting Grid Lines

By moving the mouse over a grid line and hovering for a small amount of time, the grid line gets highlighted (indicated by a yellow line). The grid line gets highlighted in the corresponding grid list and the property window is updated.

### Creating Custom Baselines

Custom baselines can be created by a left click in the grid. If a grid line or baseline was highlighted, a new custom baseline is created with properties based on the highlighted grid line. If no grid line is highlighted, a default custom baseline is created.
7.3 Conventional Grid Generator

Deleting Custom Baselines

By clicking and holding the right mouse button, the cursor changes into a red cross indicating that custom baselines can be deleted. By highlighting a custom baseline and clicking the left button (while still holding the right button) the custom baseline can be deleted.

Changing Custom Baselines

A highlighted custom baseline can be dragged by clicking and holding the left mouse button. Alternatively, after highlighting a custom baseline, all its properties can be altered interactively in the properties window.

Figure 7.7: Overview of the advanced gridder setup.

7.3 Conventional Grid Generator

7.3.1 Global and Local Grid Parameters

The definition of the global and local grid parameters for the Conventional gridder are the same as those defined in Section 7.2.3 and Section 7.2.5. The differences are that only a limited set of functionality is offered in the Conventional gridder and Local Grid Settings are applied in the Solid Regions list as shown in Figure 7.8. For a more detailed comparison of the two grid generators see Section 7.1.
7.3.2 Generating a New Grid and Mesh

Once the model has been completed, follow the steps below to generate a new grid and mesh using the Conventional Grid Generator:

1. Enter the Frequency for a harmonic simulation or Frequency and Bandwidth for a broadband simulation.
2. Set the required parts to PEC and enter the necessary parameters for the Dielectric parts.
3. Select or deselect the relevant sensors that are required for the simulation.
4. Choose the relevant ABC types.
5. Select the Conventional gridder (see Section 7.1).
6. Right click on and choose Compute Grid from Scratch.
7. If the grid is not fine enough, modify the relevant regions using Refinement on boundaries/edges or Use local grid settings. Alternatively, when using the Gridder: Tolerant / Customizable adjust the slider bar in the Gridder: Special Options to refine the grid.
8. When using Gridder: Strict / Min Step Guaranteed check if there are any Ignored Base lines or if the Max. Grading Ratio cannot be satisfied, then reduce the global Min. Step, use additional refinement, or modify the base line mode for the solids.
9. Once a satisfactory grid has been created, then right click on and select Make Voxels.
10. Right click on and select View Voxels to view the voxelled representation of the model.
11. If any thin PEC or dielectric solids are not voxelled as required, modify the Region Characteristics (see Section 7.4).

The remainder of this section defines the functionality of the different routines that can be used either to calculate and modify base lines or to compute the grid directly.
7.4. Voxelling Parameters

**Compute Grid from Scratch**

The grid is generated from scratch. Any old base lines (either from previously building the grid or that have been manually added) are ignored and overwritten. First, all base lines are recalculated based on the grid and solid settings; then all grid lines are calculated.

**Compute Base lines**

All base lines are recalculated based on the grid and solid settings. Any old base lines (either from previously generating the grid or that have been manually added) are ignored and overwritten.

**Edit Base lines**

Base lines can be added, deleted or edited. After editing the base lines, the grid must be generated using **Compute Grid from Base lines**; otherwise the modifications will not be taken into account.

**Expert Mode** only (see Section 5.12).

**Compute Grid from Base lines**

The grid lines are generated according to the set of base lines generated by **Compute Base lines** and possibly modified using **Edit Base lines**.

**Compute Grid and Make Voxels**

The grid is computed from scratch and then the voxels are calculated.

---

### 7.4 Voxelling Parameters

In **SEMCAD-X**, each solid in the model consists of a surface triangle mesh which is generated by ACIS® or from imported CAD data.

**Note:** The surface triangle mesh can be visualized by activating the View | Edges.

The surface triangle mesh is used together with the grid as input to generate the voxels for each of the solids. Some assumptions are made about the surface triangle mesh for certain cases which are listed below. For a solid, the following parameters are found in the Region Characteristics menu of the Material pane:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Volume</strong></td>
<td>The mesh is assumed to be closed. This is the default setting for solids and is generally true for all solids created within <strong>SEMCAD-X</strong>, with the exception of two cases: boxes that are created with zero thickness and bodies that were created from extruding a profile that was not closed.</td>
</tr>
<tr>
<td><strong>Surface</strong></td>
<td>The mesh is assumed to be open, e.g. shells or thin surfaces. A stair approximation will be generated by the voxeller for these bodies rather than a volume representation.</td>
</tr>
<tr>
<td><strong>Wire</strong></td>
<td>Out of the mesh a center line is estimated and voxelled for the body. The approximate diameter of the wire is needed to estimate the center line.</td>
</tr>
</tbody>
</table>
Approx. Wire Radius

The approximate wire radius.

Priority Group

Adjust the priority which solids will be voxelled in. See Section 7.4.1 for more details.

For thin PEC bodies edges are always assigned and the continuity is maintained. For dielectric objects this is not the case, since thin dielectric sheets can normally be considered electromagnetically insignificant. However, for certain special cases it may be desired that the continuity can be maintained for dielectric objects. This can be achieved using \textbf{Ensure dielectric continuity in THIN cases}.

\textbf{Note:} Ensure dielectric continuity in THIN cases is not supported by conformal voxel representation.

Additionally, in the \textbf{Voxeler Settings} pane (see Figure 7.9) the following parameters can be set:

\begin{figure}[h]
  \centering
  \includegraphics[width=0.5\textwidth]{Voxeler_Settings.png}
  \caption{Additional settings for the voxeler (left) and the list of all voxelled objects shown while viewing voxels (right).}
\end{figure}

- **Conformal Edge Resolution**: Adjusts the resolution for resolving intersections along a cell edge. This setting also influences the voxel mesh tolerance. It can only be edited in Expert mode (see Chapter 5.12).

- **Calculate Only Staircasing Voxels**: When using the standard FDTD solver, this checkbox is set by default and only staircase voxels will be calculated. When using the Conformal FDTD solver, this checkbox should be unchecked such that both staircase and conformal voxels will be calculated.

- **Optimize For Thermal Simulation**: Voxelling is optimized for thermal simulation, PEC/Metal objects are voxelled as volumes rather than just with edges.

- **Switch Staircasing/Conformal View**: This checkbox switches the voxel view between staircase and conformal view. It can only be checked/uncheck if both staircasing and conformal voxels have been calculated.
7.4. Voxelling Parameters

<table>
<thead>
<tr>
<th>Overlay Grid</th>
<th>When viewing the voxels it is possible to overlay the grid by checking this checkbox.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show Original Model</td>
<td>This checkbox enables/disables the displaying of the original model during voxel viewing.</td>
</tr>
<tr>
<td>Slicing</td>
<td>The following settings offer different possibilities for viewing voxels:</td>
</tr>
<tr>
<td>Not Sliced</td>
<td>The full voxelled model is viewed in three dimensions.</td>
</tr>
<tr>
<td>X, Y, Z Slice</td>
<td>a 2-D slice of the voxelled model is shown; the position of the slice can be adjusted.</td>
</tr>
<tr>
<td>Cubic Region</td>
<td>A limited cubic region of the voxelled model is shown in 3-D. The position and the size of the cube can be adjusted.</td>
</tr>
</tbody>
</table>

7.4.1 Voxelling Priority

In *SEMCAD-X* modeling objects are assigned a voxelling priority according to the following hierarchy:

- **Sources**: any Edge, Plane or Waveguide source has the highest voxelling priority
- **Sensors**: all Field, Edge, Far-Field and RCS sensors have the second highest priority
- **PEC/Metal, PMC**: all PEC and PMC objects follow sensors. Their voxelling priority is always higher than that of dielectric objects
- **Dielectric, Dispersive**: the lowest priority is assigned to all dielectric objects

The voxelling priority is then used to assign a Discretization Index (see Figure 7.9) which determines the order in which the objects are voxelled (the objects are voxelled from lowest to highest Discretization Index).

The Discretization Index is most meaningful for handling areas where two objects may overlap. In overlap areas, voxels will be assigned the material parameters from the object with the highest Discretization Index.

By default, the Discretization Index is calculated according to the order the parts have in the model tree, taking into account the voxelling priority. To change the Discretization Index the user can edit the Priority Group (see Figure 7.8), rather than changing the order of the object in the modeling tree.

The Priority Group has a range from 0 to 2000. By default each part is assigned a Priority Group of 1000, except the Background which is assigned 0.

Consider the following example as illustrated in Figure 7.10: the model contains three Dielectric parts Brick 1, Brick 2 and Sphere 3, which all overlap. By default the sphere’s material parameters will be assigned to the common overlapping area as it is the last part in the model tree list.

By increasing the Priority Group of the Brick 1 to say 1010 it will ensure that it has the highest Discretization Index of the three parts, followed by Sphere 3 and then Brick 2. This implies that the material parameters of Brick 1 will always be assigned to any overlapping regions with any of the other two parts. Similarly material parameters for Sphere 3 will be assigned to any overlapping regions shared with Brick 1.

**Note**: Changing the Priority Group does not affect the voxelling priority, i.e. a PEC/PMC object will always have a higher Discretization Index than a dielectric object. It only effects the order in which parts of the same Material Type (Dielectric, Dispersive, PEC/Metal, PMC) are voxelled.
Figure 7.10: Example of several parts overlapping in a model

**Note:** When working in Expert mode and additional parameter can be for solids. High Priority will set the priority of the selected object above all other solids, regardless of the Material Type.
Chapter 8

Post Processor

8.1 Introduction

The SEMCAD-X postprocessing engine provides enhanced data extraction and handling as well as visualization capabilities, enabling the same postprocessing environment within both the SEMCAD-X simulation platform and SPEAG’s DASY4 near-field scanners. Results files are platform independent, so a simulation that was run on a Linux or Windows (32 or 64 bit) machine may be viewed by using a 32 or 64 bit Windows computer with SEMCAD-X installed (memory space permitting).

In the near future SPEAG plans to entirely merge the DASY4 and the SEMCAD-X postprocessing engines, finally enabling direct comparison and cross-validation of measured and simulated data within the same environments, as well as, featuring the availability of the same standard phantoms and the correspondence of test positions, visualizations and data extraction/export within the same tool.

For further details please visit the DASY4 and SEMCAD-X WEB sites at www.dasy4.com and www.semcad.com, respectively.

8.2 Postprocessing Data Extraction

8.2.1 Access to the Simulation Results

All simulation results are accessible in the SEMCAD-X tree structure as depicted in Figure 8.1. For each completed simulation you will find the Results folder filled leading to a list of the sensors which your model contains. Selecting one of the sensors and then right clicking on one of the quantities gives a range of different extraction possibilities for the particular sensor type.

For some sensors, results are displayed directly within the list as is shown for the Sensor of the Edge Source, in Figure 8.2.
8.2. Postprocessing Data Extraction

Figure 8.1: Tree structure in **SEMCAD-X**.

![Tree structure in SEMCAD-X](image)

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Quantity</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Far Field</td>
<td>Avg P(0)</td>
<td></td>
</tr>
<tr>
<td>Field Sensor 1</td>
<td>P(x,y,z,f0)</td>
<td></td>
</tr>
<tr>
<td>Near Field</td>
<td>E(x,y,z,f0)</td>
<td></td>
</tr>
<tr>
<td>Sensor of source</td>
<td>Energy Density(x,y,z,f0)</td>
<td></td>
</tr>
<tr>
<td>Grid (259136 + 37 = 14,3032 Pixels)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voids</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Settings</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sources</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lumped elements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boundaries (Analytical)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 8.2: Direct display of results in the list.

![Direct display of results](image)

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Quantity</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall Field</td>
<td>Z(0)</td>
<td>8.05e-005 + j 0.01 A</td>
</tr>
<tr>
<td>Sensor of Edge Source 1</td>
<td>Z(f0)</td>
<td>0.8325 + j 0.35e-005 W</td>
</tr>
<tr>
<td></td>
<td>E(f0)</td>
<td>1.02</td>
</tr>
<tr>
<td></td>
<td>H(f0)</td>
<td>-0.00252 + j 0.498 V</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
8.2.2 The Viewers Tab

Overview

The Viewers Tab visually holds all results to be extracted in a tree and enables access to cached results and viewers via appropriate icons. Future versions of the SEMCAD-X postprocessor will enable a direct comparison of measured and/or simulated results via multiple selections within this area.

As soon as results are extracted or Send To Viewers, they will be listed in the Viewers Tab as shown in Figure 8.3.

![Figure 8.3: List of results and viewers in the Viewers Tab](image)

Selection of the desired viewer in the tree makes it become active. Right-click on either result or viewer in the Viewers Tab and further results can be directly extracted from there. Moreover, it allows one to show/hide the selected viewer or the entire range of active viewers. Figure 8.4 illustrates this procedure.

![Figure 8.4: Show/hide of single or multiple viewers.](image)

In addition, the Post Processor menu item allows one to perform a set of functions which apply to the viewers, such as closing all the viewers or removing all results. As shown in Figure 8.5, this option also enables one to make a printout of the active results as well as to define a reference impedance.

**Note:** The option Reference Impedance... which can be set in the Post Processor menu, allows one to define an impedance which will subsequently be used as reference for quantities like $|S_{11}|$, SWR, etc.

8.2.3 Saving and Loading Viewer State

It is possible to save the state of the Viewers tab and all the results that have already been extracted for later use. This is ideal for saving a lengthy data extraction or a set of extractions that would need to be reviewed by the user in the future.
Both of these actions can be accessed via the Post Processor menu, as shown in Figure 8.5. When saving the Viewers Tab state, a file named ‘CachedResults.h5’ is created in the project’s results directory. When loading the Viewers Tab state, this file is used to populate the tree structure containing the previously extracted results. Once the results are extracted, an appropriate viewer can be used to visualize the results as required.

8.2.4 Data Extraction and Visualization

The SEMCAD-X Postprocessor provides two basic ways for extraction and display of simulation results:

1) Data Extraction from the Project Tree and the Result List

A click with the right mouse button on one of the result icons opens a context menu of viewers which can display the corresponding data.

For each sensor or port which has been placed into the model during the model generation process, a corresponding dataset containing the results can be found using its given name in the Result List as described in the last section. Only those viewer types which are meaningful are enabled for a display of the chosen data; all other types are deactivated. For most of the datasets the viewer directly appears after the input via mouse, but some lead to an interface window where additional parameters can be specified.

Depending on the kind of result to be displayed, you have to decide on a type of viewer. Some viewers are able to display the same data but in a different manner. All viewers and their capabilities are described in Section 8.7.
2) Data Extraction from the Viewers Tab

In addition to directly selecting the result and its viewer from the Result List, the data extraction and display can be performed via the Viewers Tab. Therefore, results can be passed over via selection of Send To Viewers. In the Viewers Tab they can be further processed and result extraction can be performed.

Depending on the kind of result, the data extraction can take a short time as indicated by an appropriate progress bar. As soon as it is done, the post processed entries are added to the viewable list and displayed. Select one of the new tabs to activate and customize a specific measurement result, viewer or data extractor according to your needs, e.g., such as shown in Figure 8.6.

![Image](image_url)

Figure 8.6: The extracted results are finally displayed within the **SEMCAD-X** Postprocessing GUI and combined with one or several appropriate viewer dialogues. The entire visual representation is customizable via the user-friendly viewer dialogues.

8.2.5 Managing the Viewers

In order to customize and modify the parameters and graphics representation of a specific simulation result, select the viewer from the tree containing the viewer dialogue you are interested in. To delete a viewer, close the viewer from its dialogue by clicking on the appropriate Close button or select the close options from the tree.

8.3 Simulation Results

Depending on the sensor type and the applied simulation settings, different kinds of results with respect to spatial distribution as well as time and frequency domain can be extracted from the sensor. This type is indicated in brackets and attached to the result quantity.
8.3. Simulation Results

The specification \((x, y, z)\) represents 2-D or 3-D spatial distributions. Whereas \((t)\) indicates the result quantity can be extracted in the time domain, \((f)\) and \((f_0)\) specify its availability in frequency domain and at a specific harmonic excitation frequency, respectively.

8.3.1 Vectors and Phasors

Maxwell’s Equations are vector equations. Hence, the solutions are vector-valent time-dependent functions \(E(t)\) and \(H(t)\). As the following is equivalently true for both \(E(t)\) and \(H(t)\), we will only consider \(E(t)\) for the reason of brevity.

It can be shown that the solution in the most general case has the following form

\[
E(t) = E_0(t)e^{j\omega t} = E_0(t)(\cos(\omega t) + j \sin(\omega t))
\]

where we used Euler’s identity \(e^{j\omega t} = (\cos(\omega t) + j \sin(\omega t))\) where \(j^2 = -1\) and \(\omega\) is the spectral frequency of the electromagnetic wave. The course of the general solution with time in the complex plane is shown in figure 8.9. \(E_0(t) = (E_x(t), E_y(t), E_z(t))\) is the so-called amplitude describing the polarisation of the electromagnetic field (see figure 8.7) and \(e^{j\omega t}\) the phasor.

![Figure 8.7: A 3 dimensional vector and its decomposition into its components.](image)

If the amplitude is not time-dependent, then the electromagnetic wave is linearly polarised (see figure 8.8), otherwise, i.e., a time-dependent amplitude, the wave is circularly polarised (see figure 8.9).

This description is though only valid if the root mean square (RMS) (see below), which is the length of \(E(t)\), is constant in time.

Apparently, the solution is a complex, time-dependent vector. For such a function, several quantities can be adressed. An illustration of these is given in figures 8.7, 8.8 and 8.9.

- the entire vector \(E(t)\).
Figure 8.8: If the amplitude of an electromagnetic wave is constant, then it describes a circle in the complex plane when time passes by. The same is true for its vector components.

Figure 8.9: If the amplitude of an electromagnetic wave is periodic, then it describes an ellipse in the complex plane when time passes by. The same is true for its vector components.
8.3. Simulation Results

• the real part of \( E(t) \) is addressed as \( \Re(\mathbf{E}(t)) = E_0 \cos(\omega t) \) and contains the following vector components: \( E_x \cos(\omega t) \), \( E_y \cos(\omega t) \) and \( E_z \cos(\omega t) \).

• the imaginary part of \( E(t) \) is addressed as \( \Im(\mathbf{E}(t)) = E_0 \sin(\omega t) \) and contains the following vector components: \( E_x \sin(\omega t) \), \( E_y \sin(\omega t) \) and \( E_z \sin(\omega t) \).

• the modulus of \( E(t) = |\mathbf{E}(t)| \) is the length of the vector, i.e.
  \[
  |\mathbf{E}(t)| = \sqrt{\mathbf{E}(t)^2} = \sqrt{\Re(\mathbf{E}(t))^2 + \Im(\mathbf{E}(t))^2}.
  \]

• the phase is given by \( \phi = \omega t \) (see figure 8.8 and figure 8.9).

Having time-dependent, periodic functions \( f(t) \) it is possible define an average over time which is called the root mean square (RMS) and is defined as the absolute value of the temporal average:

\[
|M(f(t))| = \frac{1}{\tau} \int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} f(t) dt,
\]

where \( \tau = \frac{2\pi}{\omega} \) is the period of the applied function.

Hence, the real modulus of \( E(t) \) is given by \( \mathcal{M}(\Re(\mathbf{E}(t))) \) and the RMS modulus by \( \mathcal{M}(\sqrt{\mathbf{E}(t)^2}) \).

8.3.2 Definitions of Quantities

This section gives a summary of all available results and the related definitions, as functions of both electric and magnetic fields - the base entities within the Yee FDTD scheme.

Voltage (\( U \))

The voltage \( U(t) \) is defined as:

\[
U(t) = \int_{\Gamma} \mathbf{E}(\mathbf{r}, t) \cdot d\mathbf{l}
\]  \hspace{1cm} (8.1)

where \( \mathbf{E}(\mathbf{r}, t) \) is the electric field as a function of space and time integrated along a path \( \Gamma \).

Current (\( I \))

The current \( I(t) \) can be defined as an integration of the magnetic field \( \mathbf{H}(\mathbf{r}, t) \) along the boundary \( \partial F \) of area \( F \):

\[
I(t) = \oint_{\partial F} \mathbf{H}(\mathbf{r}, t) \cdot d\mathbf{l}
\]  \hspace{1cm} (8.2)

Impedance (\( Z \))

Using phasors, the two complex parameters voltage \( \mathbf{U} \) and current \( \mathbf{I} \) lead to the definition of the impedance:

\[
Z = \frac{\mathbf{U}}{\mathbf{I}}
\]  \hspace{1cm} (8.3)
Current Density ($J$)

For dielectric materials, the current density is defined as:

$$ J = \sigma E $$  \hspace{1cm} (8.4) 

where $\sigma$ is the electric conductivity and $E$ is the electric field.

For surface currents on metallic objects, the current density is calculated as:

$$ J = \nabla \times H - \frac{dD}{dt} $$  \hspace{1cm} (8.5) 

where $D$ is the dielectric displacement density (which in the case of perfect electric conductors is equal to zero) and $H$ is the magnetic field.

Standing Wave Ratio (SWR)

The SWR is defined as the ratio of the amplitude of a standing wave at an anti-node (minimum) to the amplitude at an adjacent node (maximum). It is given by:

$$ SWR = \frac{1 + \rho}{1 - \rho} $$  \hspace{1cm} (8.6) 

where $\rho$ is the reflection coefficient which is defined as:

$$ \rho = \left| \frac{Z_1 - Z_2}{Z_1 + Z_2} \right| = \frac{SWR - 1}{SWR + 1} $$  \hspace{1cm} (8.7) 

$Z_1$ is the impedance toward the source, $Z_2$ is the impedance toward the load.

Reflection Coefficient $S_{11}$

The reflection coefficient $|S_{11}|$ is defined similar to $\rho$ in the previous definition as:

$$ S_{11} = \left| \frac{Z_1 - Z_2}{Z_1 + Z_2} \right| $$  \hspace{1cm} (8.8) 

where $Z_1$ is the impedance towards the source and $Z_2$ represents the impedance towards the load.

Total Radiation Efficiency

The total radiation efficiency is described by the following parameters:

$$ \eta_{rad} = \frac{P_{rad}}{P_{src}} $$  \hspace{1cm} (8.9) 

$$ \eta_{mis} = (1 - |S_{11}|^2) $$  \hspace{1cm} (8.10) 

$$ \eta_{total} = \eta_{rad} \cdot \eta_{mis} $$  \hspace{1cm} (8.11)
8.3. Simulation Results

**Total Radiated Power (TRP)**

The Total Radiated Power (TRP) is a measure of the device’s transmitter performance. The TRP is the sum of all power radiated by the antenna, regardless of direction or polarization. TRP can be related to Conducted Power ($P_A$) by the following equation:

$$TRP = P_A \cdot \eta_{rad} \cdot \eta_{mis}$$  \hspace{1cm} (8.12)

**Note:** The conducted power is the power available to the load, i.e., TRP includes the effect of the antenna mismatch as defined by the standard.

**Total Isotropic Sensitivity (TIS)**

The Total Isotropic Sensitivity (TIS) is a measure of the device’s receiver performance. Assume a hypothetical scattered environment which provides equal mean incident powers in both of the orthogonal components $E_\theta$ and $E_\phi$, and a uniform distribution of angle of incidence (across all $\theta$, $\phi$) for both polarizations. Now assume the mean incident powers for both polarizations are simultaneously adjusted (while held equal to one another) so that the average power available to the EUTs receiver from the EUTs antenna when immersed in this environment is equal to the power required for the receiver to operate at its threshold of sensitivity (e.g., a specified bit error rate). If the EUT were now replaced with an ideal isotropic antenna which has equal gain in each linear polarization $E_\theta$ and $E_\phi$, in every direction, the mean power available from the ideal isotropic antenna immersed in this same scattered environment is defined as the Total Isotropic Sensitivity.

**Note:** It is assumed that the antenna is reciprocal.

**Power**

The complex power $P$ can be defined as:

$$P = \frac{1}{2} U \cdot I^*$$  \hspace{1cm} (8.13)

where $U$ is the complex voltage and $I^*$ is the conjugate of the complex current.

**D- and B-fields**

The dielectric displacement density is defined as:

$$D = \bar{\varepsilon} E, \quad \bar{\varepsilon} := \varepsilon_0 \varepsilon_r - j \frac{\sigma E}{\omega}$$  \hspace{1cm} (8.14)

where $\varepsilon_0$ is the dielectric permittivity of free space and $\varepsilon_r$ the isotropic relative permittivity, and $E$ is the electric field.

---

1. CTIA Certification, “Test Plan for Mobile Station Over the Air Performance: Method of Measurement for Radiated RF Power and Receiver Performance”, Revision 2.1, April 2005

The magnetic induction is defined as:

\[
\mathbf{B} = \mu_0 \mu_r \mathbf{H}
\]  

(8.15)

where \(\mu_0\) is the relative permeability of free space and \(\mu_r\) the isotropic relative permeability, and \(\mathbf{H}\) is the magnetic field.

**Poynting vector**

The complex Poynting vector is defined as:

\[
\mathbf{S}(\mathbf{r}) = \frac{1}{2} (\mathbf{E}(\mathbf{r}) \times \mathbf{H}^*(\mathbf{r}))
\]  

(8.16)

where \(\mathbf{E}\) is the complex electric field vector and \(\mathbf{H}^*\) is the conjugate of the complex magnetic field vector.

**Energy Density**

The field energy density for complex fields is defined as:

\[
d = \frac{1}{4}(\mathbf{E} \cdot \mathbf{D}^* + \mathbf{H} \cdot \mathbf{B}^*)
\]  

(8.17)

**Loss power density \(dP/dV\)**

The power loss density \(dP/dV\) in lossy materials is defined as:

\[
\frac{dP}{dV} = \sigma E|E|^2 = SAR \rho
\]  

(8.18)

**RMS \(|E|\)**

The root mean square value of the complex electric field \(\mathbf{E}\) is defined as:

\[
RMS|E| = \frac{1}{\sqrt{2}} |E|
\]  

(8.19)

**RMS \(|H|\)**

The root mean square value of the complex magnetic field \(\mathbf{H}\) is defined as:

\[
RMS|H| = \frac{1}{\sqrt{2}} |H|
\]  

(8.20)

**Potential \(\phi\)**

Electro quasi-static low frequency simulation calculate for one-connected computational domains the potential:

\[
\mathbf{E} = \nabla \phi
\]  

(8.21)
8.3. Simulation Results

Radar Cross-Section (RCS)

The radar cross section is a measure of a specific radar target’s ability to reflect radar signals in a given direction. It is a far-field parameter which is a function of transmitter position relative to target, target geometry and material composition, frequency and transmitter polarization. RCS is normalized to the power density of the incident wave at the target so that it does not depend on the distance of the target from the transmitter. RCS can be calculated using the formula:

\[ RCS = \lim_{r \to \infty} \left( \frac{4\pi r^2 |P_{\text{scat}}|}{|P_{\text{inc}}|} \right) \]  \hspace{1cm} (8.22)

where \( P_{\text{scat}} \) is the scattered power density and \( P_{\text{inc}} \) is the incident power density at the target.

Power flux through a surface

The mean power which flows through a surface \( A \) is defined as:

\[ \text{AvgP} = \int_A |\mathbf{S}| ds \]  \hspace{1cm} (8.23)

Absorption Related Quantities

1) SAR

The specific absorption rate (SAR) is defined as:

\[ \text{SAR} = \sigma_E |E|^2 = c \frac{dT}{dt} \]  \hspace{1cm} (8.24)

where \( c \) is the specific heat capacity, \( \sigma_E \) the electric conductivity, \( \rho \) the mass density of the tissue, \( E \) the induced electric field vector and \( dT/dt \) the temperature increase in the tissue.

The postprocessor provides numerous features and extraction capabilities with respect to SAR and absorption data. Therefore, each sensor which provides SAR extraction, features 3 different result instances related to SAR:

- \( \text{SAR}(x, y, z, f_0) \) allows the extraction of the SAR distribution such as defined above.
- \( \text{SpatialPeakSAR}[\text{IEEE } - 1529](x, y, z, f_0) \) allows the extraction of the Spatial-Peak SAR according to IEEE1529 such as described in section 8.5.
- \( \text{SARStatistics} \) provides the user with detailed information related to SAR and Power Loss, summarized in a table.

2) Power Loss

All quantities related to the power loss within lossy material are also easily accessible via the \( \text{SARStatistics} \) result instance. Quantities such as

- \( \text{TotalLoss} \)
- \( \text{TotalLossyMass} \)
- \( \text{TotalLossyVolume} \)
• MeanSAR
• etc.

are summarized and separated by material/solid type.

Grid / Voxels

Furthermore, the Postprocessor provides detailed simulation Grid and Voxel analysis capabilities which are available in the SEMCAD-X Project Tree. A number of appropriate extractors and viewers can consequently be selected either from the Project Tree or from the Viewers List.

8.4 General Near-Field Extraction Options

Once a Harmonic simulation has been run, near-field data is available at the excitation frequency \( f_0 \), e.g. \( H(x, y, z, f_0) \).

However, for a Broadband simulation, different extraction options are available, e.g.:

1. \( H(x, y, z, t) \): time domain near-field data can be extracted; no normalization is possible.
2. \( H(x, y, z, f_0) \): harmonic near-field data can be extracted and normalized at the frequencies specified in the Extracted Frequencies (the frequency is selected from the drop down list in the Normalization dialog shown below).
3. \( H(x, y, z, f) \): harmonic near-field data can be extracted at any frequency; no normalization is possible (the frequency is manually entered in the Normalization dialog shown below).

During the steps of simulation data extraction and display, the user is can perform a number of customizations such as normalization and material/solid specific processing.

Immediately following the selection of a result extraction, the Field Extraction Options dialog appears which allows the user to scale the simulation result from the Source Power to a desired Input Power level. For example, to compare simulated E-field to a measurement made with 2.5 W, enter 2.5 W for the Input Power. The Source Power is automatically recorded. In case multiple sources were present in the simulation, its sum is taken for the normalization.

Furthermore, for extracting fields as in item 2 and item 3, the desired frequency for extraction is either selected from the list or manually entered.

Activating the Include Mismatch checkbox will include the mismatch effects when extracting the field, i.e. if the source is not well matched the part of the Input Power will be reflected and therefore less power will reach the antenna resulting in lower near-field and far-field radiation.

Selection of extraction frequency, specification of the source power for normalization and antenna mismatch options.
8.5. SAR Extraction According to IEEE1529

Depending on the type of result and viewer selected, a subsequent dialog allows the user to include only specific solids in the extraction process.

<table>
<thead>
<tr>
<th>Solid</th>
<th>Material</th>
<th>Extremity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lens, LCD front</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>Lens, LCD back</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>Housing, lid</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>Housing, front</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>Housing, housing</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>Housing, back</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>Housing, front</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>Housing, back</td>
<td>false</td>
<td>false</td>
</tr>
</tbody>
</table>

Specification of solids which are used within the extraction process.

For certain viewers, a subsequent dialog enables to indicate which quantities should be included in the viewer or table representation.

<table>
<thead>
<tr>
<th>Table columns</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Solid</td>
<td>Material</td>
</tr>
<tr>
<td>Max. SAR at Mip</td>
<td></td>
</tr>
<tr>
<td>Max. SAR at Lcp</td>
<td></td>
</tr>
<tr>
<td>Max. SAR at Rs</td>
<td></td>
</tr>
<tr>
<td>Total SAR at Mip</td>
<td></td>
</tr>
<tr>
<td>Total SAR at Lcp</td>
<td></td>
</tr>
<tr>
<td>Total SAR at Rs</td>
<td></td>
</tr>
</tbody>
</table>

Selection of result quantities which are finally provided in the viewer and table representations.

**Note:** Multiple selection within the Extraction Options dialogs can be performed by pressing the Shift or Ctrl keys on the keyboard in addition to mouse clicking in order to choose an appropriate set of items. Furthermore, the lists can be sorted by clicking in the top row of the dialog.

8.5 SAR Extraction According to IEEE1529

Based on SCC-34, SC-2, WG-2 - Computational Dosimetry, IEEE P1529 (Recommended Practice for Determining the Spatial-Peak Specific Absorption Rate (SAR) Associated with the Use of Wireless Handsets - Computational Techniques) an algorithm has been implemented. The spatial-peak SAR can be computed over any required mass. All different tissue types, organs or parts of the model can be selected for the averaging process. In the SAR-dialog “body tissue” and “extremities tissue” can be distinguished. The distinction between these two tissue types is necessary when evaluating the compliance of wireless handsets with respect to the safety limits. Wrists, ankles, hands, feet and pinna are considered as extremities and have different safety limits than body tissue. When averaging
SAR in body tissue or extremity tissue over a required volume, only SAR values from the appropriate tissue type may be considered in the averaging. If the SAR is averaged over body tissue, the SAR values of extremity tissue voxels are set to zero and vice versa. The tissue volumes are defined as cubical volumes containing the appropriate tissue that are centered at the location. “Location” is defined as the center of the incremental volume (voxel).

8.5.1 The Algorithm

For body tissue, the spatial-peak SAR should be evaluated in cubical volumes that contain a mass that is within 5% of the required mass. The cubical volume centered at each location, as defined above, should be expanded in all directions until the desired value for the mass is reached, with no surface boundaries of the averaging volume extending beyond the outermost surface of the considered region of the model. In addition, the cubical volume should not consist of more than 10% air. If these conditions are not met, then the center of the averaging volume is moved to the next location. Otherwise, the exact size of the final sampling cube is found using an inverse polynomial approximation algorithm leading to very accurate results.

If one boundary of the averaging volume reaches the boundary of the sensor during the expansion of the averaging volume, it will not be evaluated at all. Reference is kept of all locations used and those not used for averaging the SAR. Average SAR values are assigned to the centered location in each valid averaging volume. All locations included in an averaging volume are marked to indicate that they have been used at least once. If a location has been marked as used but has never been in the center of a cube, the highest averaged SAR value of all other cubical volumes which have used this location for averaging is assigned to this location. Only those locations that are not part of any valid averaging volume should be marked as “unused”. For the case of an “unused” location a new averaging volume should be constructed with the unused location centered at one surface of the cube and expanding the other five surfaces of the cube evenly in all directions until the required mass is enclosed within this volume, regardless of the amount of included air. Of the six possible cubes with a surface centered on the unused location, the smallest cube which contains the required mass should be used.

For extremity tissue the evaluation process is the same as that used for body tissue described in the previous paragraph. As mentioned above during the evaluation of average SAR of regions assigned to “body tissue”, regions assigned to “extremity tissue” will be set to zero and vice versa.

8.5.2 SAR Calculation

In order to calculate the SAR in SEMCAD-X, you have to place a field sensor in your Model which completely encloses the area in which you want to calculate the SAR.

After the simulation, you can display the peak SAR distribution by selecting \texttt{SAR(x,y,z,f0)} in the simulation results in the project tree, or you can calculate and display the averaged SAR distribution by selecting \texttt{Spatial Peak SAR}[IEEE-1529](x,y,z,f0). For the averaged SAR you can specify an averaging mass, and you can select which solids must be included in the calculation (Figure 8.10). Furthermore, in between, the results can be normalized to 1 W by specification of the source power used in the simulation.

The result of the Spatial Peak SAR computation is finally accessible, e.g., via the View Details... dialogue (Figure 8.11) or via one of the appropriate viewers.
8.6 3-D Radiation Pattern Extraction

Via selection of the Far Field sensor in the Project Tree, a 2-D or 3-D radiation pattern can be extracted and represented within an appropriate viewer. The 3-D Far-Field Viewer displays the far-field and related chart types within the 3-D model. Each RMS far-field quantity is represented in a spherical coordinate system as a sphere whose local radius and color index are proportional to the value of the quantity in the direction defined by $(\theta)$ and $(\phi)$. Once the Spherical Field View has been selected, a dialog window pops up (as shown in Fig. 8.12). The user can specify the resolution of the sphere used to display the far-field diagrams and also its frequency in the case of a transient far-field sensor with multiple frequencies of extraction.

The sphere is represented by a quadrangular mesh whose resolution is parameterized by the number of longitudes along $(\phi)$ and the number of latitudes along $(\theta)$. As the number of longitudes or latitudes increases, the level of detail increases and so does the calculation time.

The user can also specify the spherical coordinate system used to represent the far-field quantities. By default, the spherical coordinate system has the same orientation as the Cartesian coordinate system used for the geometrical modeling and the simulation. The user can set a new origin for the $\phi$ angle, defined by the $X'$ vector, and a new origin for the $\theta$ angle, defined by the $Z'$ vector.

The Conducted Power and Receiver Sensitivity Power are used in measurements of the Total Radiated Power and Total Isotropic Sensitivity for CTIA Certification. In order to extract meaningful TRP and TIS values from the simulation, the Conducted Power and Receiver Sensitivity Power applied during...
the corresponding measurements must be entered into Farfield Options dialogue. This certification is discussed further in Section 8.6.1.

In the 3-D Far-Field panel (shown in Figure 8.13), far-field components are selected using the drop-down lists in the Data section. The following field components can be selected:

- Absolute value $|E|$ of $E$.
- Theta component $E_\theta$ of $E$.
- Phi component $E_\phi$ of $E$.
- Left polarization of $E$

$$E_L = \frac{1}{\sqrt{2}} \sqrt{(\text{Re}(E_\theta) + \text{Im}(E_\phi))^2 + (\text{Im}(E_\theta) - \text{Re}(E_\phi))^2}$$

- Right polarization

$$E_R = \frac{1}{\sqrt{2}} \sqrt{(\text{Im}(E_\theta) + \text{Re}(E_\phi))^2 + (\text{Re}(E_\theta) - \text{Im}(E_\phi))^2}$$

- The Directivity is defined as

$$D(\theta, \phi) = \frac{S(\theta, \phi)}{P_{\text{rad}}/(4\pi R^2)}$$

where $S(\theta, \phi)$ represents the far-field power density at $R = 1m$ and $P_{\text{rad}}$ the total radiated power.

- The power density $S$ of the selected far-field component $E_i$ can be expressed as

$$S = \frac{E_i^2}{Z_0}$$

where $Z_0$ is the characteristic wave impedance of free space.

- The Gain value $G(\theta, \phi)$ is defined as

$$G(\theta, \phi) = \frac{S(\theta, \phi)}{P_{\text{in}}/(4\pi R^2)}$$

with $P_{\text{in}}$ the input power of the emitter at $R = 1m$.

![Figure 8.12: Far-Field extraction property page.](image)
8.6. 3-D Radiation Pattern Extraction

For instance, to display the antenna gain, select Gain within the Data section drop-down list.

Once a far-field component and a chart type have been selected, the rendering is performed in the 3-D model view according to the settings applied in the viewer.

![Data export from 3-D far-field viewer.](image)

**Data Export** Once the 3-D far-field viewer has been extracted, various export routines can be called. The Export Data button enables export of 3-D far-field data to file. The Make X'Y', X'Z' and Y'Z' 2-D Chart buttons enable 2-D plotting of the selected plane. View Details... directly calls the HTML View Details chart for the far-field.
8.6.1 3-D Far-Field HTML viewer and CTIA Certification

General antenna and radiation values can be viewed in the HTML viewer associated with the 3-D Far-field viewer as seen in the example in Figure 8.14. Two new values have also been implemented into the SEMCAD-X Postprocessor; Total Radiated Power and Total Isotropic Sensitivity, based on standards set forth by CTIA\textsuperscript{2}.

![Figure 8.14: 3-D Far-Field HTML viewer](image)

Good radiation performance is critical to the effective operation of any mobile communication device and peak EIRP (Effective Isotropic Radiated Power) is not a good indication of this in the real world. In addition, the human head can alter the shape and peak value of the device’s radiation pattern.

Receiver performance is equally as important to the overall system performance. The downlink is integral to the quality of the devices operation and poor receiver radiation performance will cause the user, for example, to hear a low quality voice signal.

As described in the CTIA Certification, the radiated RF performance of the mobile communication device is measured by sampling the radiated transmit power of the mobile at various locations surrounding the device. A three-dimensional characterisation of the transmit performance of the device is pieced together by analysing the data from the spatially distributed measurements. All of these measured power values are then integrated to give a single value referred to as the Total Radiated Power.

The receiver performance of the mobile communication device is pieced together by analyzing the data from spatially distributed measurements of the unit’s sensitivity. All of the measured sensitivity values for the device are integrated to give a single figure value referred to as the Total Isotropic Sensitivity.

8.7 SEMCAD Postprocessing Viewer Types

The SEMCAD-X Postprocessor provides various viewer types in order to represent the results. Depending on the type of electromagnetic data and simulation mode which has been applied, the formatting can be performed using various methods. These are accessible in the Data section of the viewers.

\textsuperscript{2}CTIA Certification, “Test Plan for Mobile Station Over the Air Performance: Method of Measurement for Radiated RF Power and Receiver Performance”, Revision 2.1, April 2005
8.7. SEMCAD Postprocessing Viewer Types

Selection of result data formatting within the Data section of the viewer.

The corresponding available formatting options are summarized as follows:

<table>
<thead>
<tr>
<th>Formatting Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real Modulus</td>
<td>Displays the modulus (absolute value) of the real part of the field vector or selected component at the current phase.</td>
</tr>
<tr>
<td>RMS Modulus</td>
<td>Displays the root mean square of the field vector or the selected component. The phase selection is inoperative.</td>
</tr>
<tr>
<td>Real Part</td>
<td>Displays the real part of the field vector or selected component at the current phase. In contrary to Real Modulus, the display range encloses negative values.</td>
</tr>
<tr>
<td>Imaginary Part</td>
<td>Same as Real Part with a phase shift of 90°.</td>
</tr>
<tr>
<td>Phase</td>
<td>Displays the phase angle of the vector components. It is inoperative for vector view.</td>
</tr>
</tbody>
</table>

The following section describes the various viewer types used to interpret the SEMCAD-X simulation data, which the new SEMCAD-X Postprocessing engine now offers.

8.7.1 View Details

View Details creates an .html type report containing scalar field values (e.g., voltage, current and impedance) in a new field tab in the Chart Manager. Information including project, grid and run title are also printed on the report. To save the displayed text, highlight it, bring up the context menu with the right mouse button and then copy it to the clipboard.
8.7.2 Slice Field Viewer

The Sliced Field Viewer displays a slice of the field component in Cartesian coordinates, which is recorded by a two or three dimensional sensor. When the Sliced Field Viewer is selected, the Field Extraction Options dialog (shown below) appears, allowing the user to normalize the results and to select the relevant frequency. The slice is rendered in the simulation model. The viewer dialogue is named according to the field component, and also displays information concerning what frequency and what normalization was chosen in the Field Extraction option.

The new Postprocessor can now simultaneously render multiple field components from multiple sensors (illustrated in Figure 8.15). The display options (e.g., position, scaling, animation, transparency) for each component are then accessed in the Chart Manager, by selecting the tab of the field component. The features that the Sliced Field Viewer offers are presented in the tables below. Many of the features that are discussed in this section are common to the other SEMCAD Viewer types.
8.7. SEMCAD Postprocessing Viewer Types

**Data**  This list enables the user to select how the data is displayed. The user has a choice between Real, Imaginary and RMS Modulus, as well as being able to select a single vector component or the spatial absolute value of the electromagnetic field (see also section 8.7. The phase of the field can also be changed manually or animated by selecting the Animate button. The level of transparency can also be set by checking the Transparency box and adjusting the slider as desired.

| H(x,y,z)0 in dB |  
|----------------|---
| Data           | dB normalized to |
|                | 750e-555 V/m |

**Scaling**  The field distribution shown in the slice can either be displayed using a dB scale or linearly scaled. In both cases the minimum and maximum values can be adjusted manually. The color bar shows the color mapping that is used to display the field data. The Get Max button returns the maximum value of the whole sensor, which is used to normalize the field if a dB scale is used. The of Slice button performs the same but related to the current slice. The Autoscale button scales the field according to the value entered in this box. The Sync Scales checkbox can be used to synchronize the scaling when multiple fields are plotted simultaneously.

**Data section options in the Chart Manager.**

**Scaling options in the Chart Manager.**
Slice

Once the Cartesian plane which the slice lies in has been selected, the offset along this axis can be set. The Show Grid checkbox overlays the FDTD grid onto the slice. The Clip model checkbox clips the view of the model along the slice plane. The slice can also be moved automatically to the plane that contains the maximum field value in the sensor with the Go to Max button. The Animation... button is used to for the same purpose as the Animate button, but allows the user to save a sequence of screen shots showing the field variations. This feature can be used to make short movie clips for presentations (see also 8.8). The Export Slice button can be used to export the field distribution of the current slice to file.

Line Extraction

The Postprocessor also supports 2-D line charts. The Cartesian or time axis is first selected and then the offset of the other Cartesian coordinates in the sensor are chosen. The line chart is plotted using the Make 2-D Chart button. It is possible to plot more than one line on a single chart (see figure 8.16). Once the first chart has been plotted, the user will have the option to either add the next graphs to the original chart or to display it in a new chart.
8.7. SEMCAD Postprocessing Viewer Types

8.7.3 Surface Field Viewer

The Surface Field Viewer can be used to display fields on the surfaces of the SEMCAD-X model. Many of the features for this viewer are the same as those of the Sliced Field Viewer and have already been discussed in Section 8.7.2.
8.7.4 Vector Field Viewer

The Vector Field Viewer uses displays the field component using vectors. The size of the field component determines the size of the vector and the direction of the field component determines the direction that the vector is displayed.

Note: The density of the vectors is strongly dependent on the dB scale that is used, i.e. in order to increase the density of the vectors the minimum of the dB scale must be reduced.

8.7.5 Spherical Field Viewer

The Spherical Field Viewer is used for displaying a three dimensional Far-Field pattern recorded by the Far-Field Sensor. When it is selected, the 3-D Far-Field Options dialog is activated. Here the user can set the resolution for the number of points used to display the 3-D far-field. The coordinate system of the far-field can also be set, in order to align the $\theta$ and $\phi$ components with the coordinate system of the simulation. There is also an option to calculate the gain. The far-field is then calculated. The 3-D spherical view of the total far-field is then plotted. The users can then select plotting of other...
far-field polarizations, including $\phi$ and $\theta$ components, left and right components and the directivity.

Furthermore, a number of display options are available, which allow the user to format and position the 3-D representation as desired.

### 8.7.6 Iso Surface Field Viewer

The Iso Surface Field Viewer is used to plot three dimensional electromagnetic field components which have the same magnitude. In the tab in the Chart Manager, the value of the field component must be entered. The Add Iso Surface button is then used to create the iso surface. The 3-D iso surface is then displayed on the SEMCAD-X model (see Figure 8.20).
8.7.7 2-D Curve Plot View

The 2-D Curve Plot View can be used to display time or frequency domain data or components of the far field as function of the angle. You can activate the 2-D Curve Plot View by right clicking on a result in the Result List (Figure 8.21). You can also drag and drop the result icon into the chart manager as described in Section 8.2.

Figure 8.21: Selecting a 2-D Curve Plot View.

Depending on the selected quantity, a set of dialogs will appear which allow you to customize the results to be displayed. For the extraction of e.g. the feedpoint impedance, the frequency range is automatically selected according to the initial signal settings. Moreover, it can be selected if the real part, the imaginary part or the absolute value will be displayed (Figure 8.22). The impedance will then appear in a 2-D Curve Plot View Dialog (Figure 8.23). You can zoom into the plot range by selecting a rectangular area with the mouse. Right clicking into the data window will open a context menu with many options to customize the viewer style (Section 8.7.9).

Furthermore, it is also possible to plot the impedance \( Z(f) \) on a Smith Chart as shown in Figure 8.24.

If you want to have different simulation runs within a project and want to compare their results in the same 2-D Curve Plot View, just open the first viewer as described above. Then select the second result and choose the 2-D Curve Plot View, again. A dialog will appear asking you whether you want to display the new result in a separate dialog or whether you want to add them to one of the windows which already exist (Figure 8.25).
8.7. SEMCAD Postprocessing Viewer Types

Figure 8.22: Extraction of the feedpoint impedance.

Figure 8.23: Extracted feedpoint impedance in the 2-D Curve Plot View.

8.7.8 2-D Polar Plot View

Far-field radiation patterns can be visualized both in three dimensions (Section 8.6) or in a 2-D Polar Plot View. You can open a 2-D Polar Plot View by selecting the results of a far-field sensor in the project tree. In the Far-field options dialog (Figure 8.26), you can then specify the plane in which the far-field is extracted. In the following dialogs you will be able to select the components to be displayed and to add the new data to an already existing dialog as described in the previous section. Figure 8.27 shows the extracted 2-D far-field pattern in a 2-D Polar Plot View.

8.7.9 2-D Viewer Customization and Data Export

Within the 2-D Curve Plot View and the 2-D Polar Plot View, the user may change a number of individual settings in order to achieve maximum flexibility.
Zoom

To concentrate on a specific region within the chart, the Zoom function may be applied by clicking and dragging the mouse (forming a rectangle) in the region of interest. This region will be zoomed automatically and is displayed in the chart. To undo the zooming, choose the Undo Zoom command from the context menu by clicking on the chart using the right mouse button.

Maximize Window

It is possible to display a chart window in full screen mode via the Maximize ... command from the context menu by clicking on the chart using the right mouse button. In order to resize the window again to its initial size, either push ESC or click in the upper left corner of the window.

Indication of XY position

In the upper left corner of the chart window itself, the actual position of the mouse pointer is indicated by X and Y coordinates.

In addition, all viewer types may be customized individually with respect to scaling, labeling, font, grid lines, etc. within the customization dialogues. These context menus are easily reachable by clicking on the chart with the right mouse button.

Access to chart customization via the context menus.

The initial context menu enables the customization of basic features which are summarized and briefly explained as follows:

<table>
<thead>
<tr>
<th>Viewing Style</th>
<th>Enables switching between color and monochrome views.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Font Size</td>
<td>The size of the font may be changed between small, medium and large.</td>
</tr>
<tr>
<td>Numeric Precision</td>
<td>The numeric precision will be shown using one, two or three decimals.</td>
</tr>
<tr>
<td>Plotting Method</td>
<td>Various plotting methods are available, e.g., points, single lines, spline interpolated values and bars.</td>
</tr>
<tr>
<td>Data Shadows</td>
<td>Indicates whether the data is displayed with a shadow underlying or not.</td>
</tr>
</tbody>
</table>
8.7. SEMCAD Postprocessing Viewer Types

<table>
<thead>
<tr>
<th>Grid Lines</th>
<th>Via this topic a number of options for the display of grid and reference lines within the chart may be activated.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid In Front</td>
<td>Determines whether the grid lines are displayed in front of the actual result data or not.</td>
</tr>
<tr>
<td>Include Data Labels</td>
<td>By activation of this option, for every data point in the graph a small labeling using the corresponding data value is performed.</td>
</tr>
<tr>
<td>Mark Data Points</td>
<td>If this option is activated, every data point is marked with a little dot.</td>
</tr>
</tbody>
</table>

An even more advanced customization of a specific chart window is achievable by changing parameters within the Customization Dialog itself. By clicking the right mouse button inside the viewer window and selecting the Customization Dialog ... section, a dialog window will appear. This dialog allows modification of topics like axis extends, labeling, font type and size, type of plotting as well as coloring matters. Since the dialogue is self-explanatory, it will not be discussed further. For export purposes an Export Dialog is available by moving the mouse pointer in the chart window and clicking the right mouse button. After choosing the Export Dialog ... topic, an export dialog window will pop up. Using this dialog various kinds of data export such as printing or saving results to a file may be performed. For example, a printout is achieved by activating the MetaFile and Printer checkboxes and clicking on the Print ... button. Export of the data to the clipboard as well as saving data to files is handled in a similar manner.

![Advanced customization dialogue appearing by clicking within the viewer window (right button).](image1)

![Export dialogue enabling various export of simulation data.](image2)
8.8 Generating an Animated Image Series

The SEMCAD-X post processor allows the user to easily export data displayed in the main window into a sequence of images. Each of the images will contain a bitmap of the fields displayed in the main
8.9 Printout of Simulated Data and Results

The **SEMCAD-X** postprocessor enables to perform a printout (hardcopy or file) of relevant simulation parameters as well as the graphical representation of results and the setup. **SEMCAD-X** generates

![Figure 8.27: 2-D far-field radiation pattern in the 2-D Polar Plot View.](image)

Window with subsequent phase with a shift of 2°. This set of images can then be transformed into an animated sequence using an suitable software. In order to generate an image sequence, open the appropriate viewer, e.g. the **Slice Field View** and chose an appropriate perspective and scaling of the fields and the objects in the main window. All images will be exported with a white background. In order to generate the image sequence, click the **Animate** button in the viewer panel (Figure 8.28). A dialog will appear in which you can select a directory in which the exported screen shots will be saved. The images will be saved as bitmaps and be named `anim000000.bmp-anim000179.bmp`. Please note that the file format is not compressed and a lot of disk space may be required to store the data.

![Figure 8.28: Exporting an animated image sequence.](image)
this printout in HTML format in order to enable straight forward modifications and processing of the document in a user-friendly way. The printout is performed in a conventional way via the SEMCAD-X main menu bar in section Post Processor or via the toolbar icon.

8.9.1 Report Generator

The Generate Report... command or the Ctrl+R keyboard shortcut initiate the printout procedure as follows:

1. In a first step, the Report Generator Options dialogue appears which enables to specify a number of printing and formatting relevant topics via edit fields and buttons.

   Report Generator Options dialog permits the user to define the formatting of the printed document.

<table>
<thead>
<tr>
<th>Report directory</th>
<th>This path specifies the location on the harddisk that the printout will be stored as an HTML file.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Template</td>
<td>This path indicates which template is used to perform the formatting of the printout document. The SEMCAD Postprocessor allows the user to arrange the printout appearance according to a user defined formatting via templates. A predefined standard template is available on the SEMCAD-X installation CD, located within the Print_Templates folder. Please copy these templates on your local harddisk in order to process the printout.</td>
</tr>
<tr>
<td>Document title</td>
<td>In this field, the title of the document is specified. It will be passed to the final printout via the user defined template. The name of the project file is used as a default entry.</td>
</tr>
<tr>
<td>Document name</td>
<td>In this field, the name of the document is specified. It will be passed to the final printout via the user defined template. The name of the project file is used as a default entry.</td>
</tr>
<tr>
<td>Edit template...</td>
<td>This button starts the SEMCAD-X printout template editor which allows a user defined design of the printout document via template variables. This editor is explained in more detail within section 8.9.2.</td>
</tr>
<tr>
<td>New template...</td>
<td>This button allows the user to generate a new printout template and asks the user to specify a file name (ending in .htm).</td>
</tr>
</tbody>
</table>
8.9. Printout of Simulated Data and Results

Next

By pressing on this button, the printout is formatted according to the user’s template and input and displayed within a separate window in the Chart Manager area. All the chosen settings of the Report Generator Options are saved after every usage of the dialog. Therefore, in case all options have been previously defined, this button allows one to generate the final printout in one single step.

2. After having specified the necessary items within the Report Generator Options dialog, pressing on the Next button creates an appropriately arranged output according to the printout template, including the currently shown GUI setup itself, a color bar with scaling data, result data of all present viewers as well as data related to the simulation. This printout is shown within the SEMCAD-X HTML Report Viewer located in the Viewers area. Section 8.9.3 outlines its features in more detail.

8.9.2 Template Editor

The Template Editor allows the formatting of any SEMCAD-X printout according to a user’s definition. Furthermore, the generated template can be stored on the harddisk and be reloaded within the Report Generator Options dialog to define the formatting of the printed document.

Parameters are selectable via the drop-down list located in the upper left corner of the Template Editor. Thus, the user can add a simulation parameter just by selecting its specific entry of the form \( \{ X: Y \} \) from the drop-down list.

Like in a common text editor, the user can easily modify the location or the arrangement of these icons within the editor template. In addition, any user-defined text may be added to the template in a straightforward manner. The icons located in the top area of the Template Editor enable the application of general formatting features to the text, e.g., centering, numbering as well as bold or italics font styles.

Finally, the generated template is saved via the disk icon located in the Template Editor’s menu bar or by closing the dialog.

SEMCAD-X Printout Template Editor enabling a user-defined outlook of the printed document containing the simulation results.

**Note:** In order to perform a vertically non-spaced carriage return within the Template Editor, please use the Shift-Return key combination.
Via the drop-down list in the Template Editor, the user can add any simulation related parameters to the printout template.

8.9.3 SEMCAD HTML Report Viewer

The generated printout is finally added in the form of an additional viewer into the Viewers section. In case the template formatting has been performed appropriately, all parameters, simulation results and graphs are shown within the HTML Report Viewer.

Since the visualized content is formatted in pure HTML, the text as well as the pictures displayed in this viewer can be still edited according to the user’s specific needs and wishes. This can be performed in a similar manner as discussed in section 8.9.2, i.e., by editing like in a common text editor.

The Browser... button allows the display of the printout document within a browser window, e.g., to enable a user-friendly display and processing of its content.

The actual printout can be performed either directly by clicking on the printer icon located in the top area of the HTML Report Viewer or via the browser window.

**Note:** The printout feature assembles all data from the extracted viewers available in the Chart Manager. Please make sure that the desired results for the printout have been previously extracted. Moreover, don’t mix different electromagnetic or absorption result types within one printout, i.e., it does not make sense to display E-field and SAR distribution within the GUI using the same scale. In this case, please close the appropriate viewers before printing.
PDF printout of SEMCAD-X simulation parameters and results using the HTML Report Viewer and its printing facilities.

The SEMCAD-X HTML based printout feature allows, in case of large documents including multiple charts and viewers, a proper formatting and arrangement of the data on multiple pages.

---

The SEMCAD-X HTML print-out allows a clean formatting of large documents including measurement parameters, results and graphics, nicely arranged on multiple pages.

**Note:** Due to the standardized and simple nature of the HTML formatting, the printout document can be easily exported to and processed within other text editing environments, e.g., Microsoft Word, just by the Copy-Paste command applied to the HTML Report Viewer’s content. A printout to Acrobat PDF type documents can be achieved via selection of an appropriate Acrobat PDF related printer driver.

### 8.10 File/Clipboard Export of Simulated Data and Results

The SEMCAD-X postprocessor enables the export of all measured data to file or the clipboard, including all relevant coordinates and parameters. In order to perform the data export, use the SEMCAD-X Project Tree and select the result type to be exported from the list. A mouse right-click on one item displays the list of relevant viewers and extraction procedures. After choosing the Field Data Export... option, a normalization dialog will appear. After closing this dialog, the data will be extracted. Depending on the kind of result, the data extraction can take a few seconds.

Select one of the new tabs in order to customize and perform the actual export of measurement data, coordinates and parameters.
8.10. File/Clipboard Export of Simulated Data and Results

Figure 8.29: Export dialog within the SEMCAD-X postprocessor enabling various customizations as well as a preview feature with direct export capabilities.

The export dialog allows a customization of the data held within its instance, presented as subdivision into the following groups:

- The **Export Data** group enables the actual selection of the measurement result to be exported, e.g., the values for each X, Y and Z components or the Total field. Just activate the appropriate checkbox for an export of the specific data.

- The **Export Options** section of the dialog controls how the exported data is formatted. Furthermore, it allows to add the measurement coordinates to the text.
  - **Header Information** adds the type declaration of the data as well as its file source to the output field. **Coordinates** adds all the associated measurement coordinates. The last customization option lets the user specify for how many significant digits the numbers are formatted.

- The **Export Data...** button allows to export all the selected and formatted data to a ASCII text file. A dialog will pop up and let you specify the name and location of the text file.

- The **Show Data >>** button enables the user to observe the data to be exported in a preview form within the text field located just beside the two other sections. Using the mouse (left-click and drag), you can highlight a desired part of the displayed text and easily copy-paste it to an external program/editor via either the Ctrl-C and Ctrl-V keys or via mouse right-click within the selected text area.
Chapter 9

The Virtual Family

9.1 Anatomical CAD Models

The Virtual Family consists of four anatomical high-resolution models which are based on Magnetic Resonance Imaging data of two adults and two children. The images have been used to construct three-dimensional CAD objects of the tissues and organs. Figure 9.1 shows the four models, and Table 9.1 summarizes their physiological data. 84 different tissue and organ types are distinguished.
9.2 Virtual Family Tool

9.2.1 Getting Started

While the CAD files of the models can be loaded directly into SEMCAD-X, their use with different simulation software is possible by exporting them into a voxel based format. The Virtual Family Tool loads the CAD files of the models and allows their free positioning, scaling and rotating with respect to a reference coordinate system. A mesh can be generated in arbitrary resolution, and the models can be discretized (voxeled) and exported in a generic data format. The Virtual Family Tool can be launched from SEMCAD-X or SEMCAD-X light by clicking on File | Virtual Family Tool. Figure 9.2 shows the graphical user interface (GUI) of the Virtual Family Tool.
9.2.2 Handling the Anatomical Models

The CAD files of the Virtual Family can be loaded by selecting File | Open SAT File... in the main menu of the Virtual Family Tool, while the Model-Tab in the GUI is active. Once loaded, the model or parts of it can be moved, rotated or scaled with respect to the reference coordinate system. It is possible to delete tissues and organs of the model should they not be meshed and voxelized. The editing of the properties and operations which can be performed on the model or its parts correspond to those of SEMCAD-X and are explained in detail in Sections 3.2.1 and 3.2.2. Please note that it is not possible to save the changes that have been made to the model or its position.

9.2.3 Grid Generation

In order to mesh and export the model, switch to the Grid/Voxels-Tab in the main window of the Virtual Family Tool. There will be a default grid configuration which encloses the entire model. You can activate it by selecting the Grid icon in the Grid/Voxels-Pane (Figure 9.3). Please note that it is not possible to save the grid configuration.

For each axis of the reference coordinate system, the origin, the mesh step and the extent of the grid can be specified in mm. Additionally, scaling factors can be defined for each axis and the entire coordinate system (Global Scale). Changes of the grid settings can be viewed in the model window of the GUI. A slider to change the Global Scale can be activated by right-clicking into the model window and keeping the mouse button down (Section useinteractive).

---

1It is possible to load more than one model of the Virtual Family or to use the Virtual Family Tool to view other SAT-Files, but only the Virtual Family model which has been loaded first will be meshed and exported.
### 9.2.4 Creating and Exporting the Voxels

When the grid has been defined, the model can be voxeled and exported in a generic format (raw-file). In addition to the raw-file which contains the geometry information, a Material Table file will be written (plain text) which contains the indices of the tissues, their colors as RGB values and number and size of the mesh steps. When the voxel-icon in the Grid/Voxels-Pane is highlighted, you can specify names and paths for these two files (Figure 9.4). By right-clicking on the voxel-icon, a context menu will appear where you can select Make Voxels, which will initialize the discretization of the model and write the files to the locations you have specified.

![Figure 9.4: Export file settings of the Virtual Family Tool.](image)

The voxel data in the (raw-file) are in a plain binary format which contains one byte (8bit) for each voxel index. The tissue or organ for this index is given in the Material Table file. The content of the file is a three-dimensional array of the size $nx \times ny \times nz$ as given in the Material Table file. The position $p$ of the voxel with the three-dimensional index $i, j, k$ in the file can be calculated from

$$p = k \times nx \times ny + j \times ny + i$$

(9.1)

The when reading the file into a three-dimensional array of the size $nx \times ny \times nz$, the index $i, j, k$ can be calculated from the position $p$ using the following algorithm, where / represents an integer division and % represents the modulo operator:

$$k = p/(nx \times ny)$$

(9.2)

$$p = p%(nx \times ny)$$

(9.3)

$$j = p/nx$$

(9.4)

$$i = p%nx$$

(9.5)
Chapter 10

Extraction of Scattering Parameters

10.1 Introduction

The impedance/admittance description of microwave circuits becomes an abstraction in many respects since it is difficult to define and to measure the voltages, currents and impedances in a direct manner at microwave frequencies, especially for non-TEM lines. The directly measurable quantities are the magnitude and phase of a traveling wave in a given direction. In other words, the magnitudes and phases of incident, reflected and transmitted waves with respect to a junction can be measured by means of, for example, a small probe. The matrix describing the relationship between these directly measurable quantities is called the scattering matrix.

SEMCAD-X enables the user to extract S-parameters of the structure under consideration by employing S-Parameter Simulation Setup and S-Parameter Evaluation, or Stand Alone S-Parameter Evaluation or S-Parameter Evaluation from Z-Matrix tools. In this chapter, the procedure of S-parameter extraction in SEMCAD-X will be explained in detail. The first section of the chapter gives a short overview of the scattering matrix theory, and in the following sections, the S-Parameter tools are presented.

10.2 The Scattering Matrix

To illustrate the scattering matrix representation, consider the N-port network with equal characteristic impedance shown in Figure 10.1. $U_i^+$ is the amplitude of the voltage wave incident on port $i$ and $U_i^−$ is the amplitude of the voltage wave reflected from port $i$. The scattering matrix of this network is defined as follows:

\[
\begin{bmatrix}
U_1^- \\
U_2^- \\
\vdots \\
U_N^-
\end{bmatrix} =
\begin{bmatrix}
S_{11} & S_{12} & \cdots & S_{1N} \\
S_{21} & \vdots & & \vdots \\
\vdots & & S_{N1} & \cdots & S_{NN}
\end{bmatrix}
\begin{bmatrix}
U_1^+ \\
U_2^+ \\
\vdots \\
U_N^+
\end{bmatrix}
\]

(10.1)


---

1David M. Pozar, Microwave Engineering, Wiley, 2005.
10.2. The Scattering Matrix

Figure 10.1: An arbitrary $N$-port network.

The elements of the scattering matrix can be determined as:

$$S_{ij} = \left. \frac{U_i^-}{U_j^+} \right|_{U_k^+ = 0 \text{ for } k \neq j}$$

which means that $S_{ij}$ is found by driving (exciting) port $j$ with an incident wave of voltage $U_j^+$ and measuring (computing) the amplitude of the reflected wave $U_i^-$ coming out of port $i$. The incident waves on all ports except the $j$th are set to zero. In other words, all ports should be appropriately terminated to avoid reflections. Thus, $S_{ii}$ is the reflection coefficient obtained looking into port $i$ and $S_{ij}$ is the transmission coefficient from port $j$ to port $i$ when all ports are matched. The S-parameters of a network are properties only of the network itself and are defined under the condition that all ports are matched.

S-parameters of the network can also be given in terms of the reflected and incident wave amplitudes. For the ports with the characteristic impedance $Z_{0i}$, the wave amplitudes are defined as

$$a_i = U_i^+ / \sqrt{Z_{0i}},$$
$$b_i = U_i^- / \sqrt{Z_{0i}}$$

where $a_i$ and $b_i$ are the incident wave at the $i$th port and reflected wave from that port, respectively. The total voltages and currents at that port can be obtained as

$$U_i = U_i^+ + U_i^- = \sqrt{Z_{0i}}(a_i + b_i)$$
$$I_i = \frac{1}{Z_{0i}} (U_i^+ - U_i^-) = \frac{1}{\sqrt{Z_{0i}}}(a_i - b_i)$$

The scattering matrix can be written as

$$[b] = [S][a]$$
where

\[
S_{ij} = \left. \frac{b_i}{a_j} \right|_{a_k=0 \text{ for } k \neq j}
\]  

(10.8)

10.3 Extraction of the Scattering Parameters in SEMCAD-X

In order to extract the S-parameters, a number of simulations have to be build and run sequentially. The S-parameter tools in SEMCAD-X automatically build these necessary simulations and then evaluate and plot the resulting S-parameters in 2-D. SEMCAD-X provides three evaluation tools which are for waveguide sources and edge sources, and the user has to select the appropriate one according to source excitations and port configuration existing in the structure. In the following subsections, the working principles of these tools will be explained in detail. Certain restrictions and rules that have to be considered in order to apply them correctly and efficiently will also be presented.

10.3.1 S-parameter Simulation Setup Tool

Depending on the number of required S-parameters to be extracted, a certain number of simulations have to be built and run as a batch. The S-Parameter Simulation Setup tool shown in Figure 10.2 enables the user to automatically generate these simulations. Before employing this tool, a template simulation has to be prepared. The template simulation must have all necessary sources and port/field sensors placed according to required S-parameters. The field sensors instead of port sensors are necessary only for rectangular, circular, parallel plate waveguides and coaxial lines, i.e., when the waveguide source is used in the project. In order to match the ports which is required in order to use S-Parameter Evaluation tool (see Section 10.2), all ports in the project should be terminated with absorbing boundaries. In case of S-Parameter Evaluation from Z-Matrix tool, the ports should not be matched, i.e., not to be terminated with absorbing boundaries.

![Figure 10.2: The S-Parameter Simulation Setup tool.](image)

To illustrate how to prepare the template simulation with edge sources, consider again the N-port
10.3. Extraction of the Scattering Parameters in SEMCAD-X

structure shown in Figure 10.1. As explained in Section 10.2, to extract only \( S_{ij} \) value, only port \( j \) must be excited, i.e., all sources except the one at port \( j \) must be deactivated. The port sensors at port \( i \) and \( j \) must also be placed to record the voltage and current values to calculate the required S-parameter. If the user requires to obtain additional \( S_{kj} \) parameters where \( k = 1 \ldots N \) and \( k \neq i \), appropriate port sensors should be added to each port. If \( S_{kj} \) parameters are also required then an appropriate source should be placed at port \( i \). The names of the port sensors must be the same as their corresponding sources.

SEMCAD-X allows the user to extract the S-parameters of structures with rectangular/circular/parallel plate waveguide or coaxial line ports. In such cases, the user must use the waveguide sources and the field sensors instead of the edge sources and the port sensors. The placement of these sources and sensors necessary for the S-parameter extraction in waveguiding structures can be automatically done by using Rectangular Waveguide, Circular Waveguide, Parallel Plate Waveguide or Coaxial Line tools under Tools | Waveguides menu (see Figure 10.3). A waveguide source and three field sensors are placed automatically using the user defined settings (see Chapter 11). The names of the sources and the sensors must not be changed.

Figure 10.3: Rectangular Waveguide, Circular Waveguide, Parallel Plate Waveguide and Coaxial Line tools.
To sum up, depending on which S-parameters are to be extracted, the project must have the corresponding sources (edge or waveguide sources) and sensors (port or field sensors), as explained briefly above. Note that **SEMCAD-X** currently supports the computation of the S-matrix (not the generalized scattering matrix) of a given structure, i.e., it describes the reflection and transmission characteristics of the specified mode of the rectangular/circular waveguide and TEM mode of the coaxial line and parallel plate waveguide.

The model is now ready to be set up as a template simulation following the explanations given in Chapter 5. Once the grid has been made, the template simulation is selected and then the **S-Parameter Simulation Setup** tool is run from the **Tools | S-Parameter** menu. The user selects the ports to be excited and then the necessary simulations are generated automatically by copying the template project. Note that, all ports have to be selected if **S-parameter Evaluation from Z-Matrix** tool will be employed to extract the parameters. The number of simulations created are equal to the number of selected ports. Each simulation has only one active source to excite the corresponding port. These multiple simulations can be run by clicking the **Run Batch** icon and selecting all except the template simulation. After the simulations are complete, required S-parameters can be computed from the recorded results and then displayed as explained in the next section.

### 10.3.2 S-parameter Evaluation Tool

The **S-Parameter Evaluation** tool from the **Tools | S-Parameter** menu enables the user to extract and then display the S-parameters of the simulated structure (Figure 10.4). The project must have either edge sources and port sensors or waveguide sources and field sensors built as explained in the previous section. The user can select the evaluation type which can be the magnitude (in dB or linear) or the phase of S-parameters, input, transmitted and reflected power at ports. The tool automatically detects the simulations with results and the ports with corresponding sensors. If the model contains waveguide sources at the ports, a window appears in which the user must set the dielectric properties of waveguiding structures at ports by selecting from the list of existing solids in the model (Figure 10.5).

The main window (Figure 10.4) displays a possible set of S-parameters that can be extracted from the completed simulations present in the project. The user selects the set of S-parameters to be evaluated and the type of the evaluation, and then the evaluation process starts. Once the computation is completed another window appears showing the list of the extracted parameters. The user can select the parameters from the list which will be displayed as a 2-D plot. Note that, the tool also enables the user to export evaluation results in CITI and Touchstone formats, and also save them in text files (see Chapter 11).

Note that in order to be able to use this tool, there has to be at least one completed simulation which is appropriately generated by the **S-Parameter Simulation Setup** tool.

### 10.3.3 Stand Alone S-parameter Evaluation Tool

The **Stand Alone S-Parameter Evaluation** tool functions similar to the **S-Parameter Evaluation** tool except that it can only be used to extract S-parameters of a completed simulation having only one active edge source. The tool automatically detects all sensors existing and then displays a possible set of S-parameters that can be extracted from the simulation. The remaining process is the same as in the **S-parameter Evaluation** tool case.

### 10.3.4 S-parameter Evaluation from Z-matrix Tool

The **S-Parameter Evaluation from Z-Matrix** tool enables the user to extract the full S-parameter matrix of the simulated structure from the Z-matrix and then plot them in 2-D. It is designed for only edge source excitation and all the edge sources in the project have to be active in the template simulation.
10.3. Extraction of the Scattering Parameters in **SEMCAD-X**

![Figure 10.4: The S-Parameter Evaluation tool.](image1)

![Figure 10.5: The pop-up window to set the materials for the ports having the waveguide sources.](image2)

in order to be able to use this tool. The tool automatically detects the simulations which are produced using **S-Parameter Simulation Setup** tool and then uses the simulation results to extract full S-parameter matrix from Z-matrix using following equation

\[
[S] = ([Z] - [Z_0])([Z] + [Z_0])^{-1}[G_0]^{-1}
\]  

(10.9)

where

\[
[G_0] = \begin{bmatrix}
\sqrt{\frac{1}{z_{01}}} & 0 & \cdots & 0 \\
0 & \ddots & & \\
\vdots & & \ddots & \\
0 & \cdots & 0 & \sqrt{\frac{1}{z_{0N}}}
\end{bmatrix}
\]  \hspace{1cm} (10.10)

\[
[Z_0] = \begin{bmatrix}
Z_{01} & 0 & \cdots & 0 \\
0 & \ddots & & \\
\vdots & & \ddots & \\
0 & \cdots & 0 & Z_{0N}
\end{bmatrix}
\]  \hspace{1cm} (10.11)

The user can then select the S-parameters to be displayed as a 2-D plot once the evaluation is finished. In addition, full S-parameters and Z-matrix elements versus frequency are saved in separate text files under the same folder where the project is.
10.3. Extraction of the Scattering Parameters in **SEMCAD-X**
Chapter 11

Tool Menu Items

The Tools menu items have been added as Python script extensions to the SEMCAD-X simulation platform. These give the user further customization options and an environment in which scripts with a user interface can be launched.

11.1 Reload Tools

Each time an instance of SEMCAD-X is started, the Tools menu is automatically updated with all valid Python tool scripts stored in GUI/stdscripts/ located in the installation folder. The user can add scripts to this folder such that they will be added to the Tools menu. The Reload Tools refreshes the Tools menu if SEMCAD-X is already running when scripts are added.
11.2 Scripting | API Browser

The API Browser Tool can be used when writing a python script. It is a tree structure of all the classes and methods that are accessible in **SEMCAD-X**.

1. Open the API Browser Tool from the Tools | Scripting menu.

11.3 Scripting | Start Python Documentation Server

The Start Python Documentation Server Tool can be used to generate a webserver that displays the structure of all the classes and methods that are accessible in **SEMCAD-X**.
11.4 Modeling | Array Tool

The Array Tool can be used to build a three-dimensional array of any selected object in the model.

1. Select the object to set as unit element of the array.
2. Open the Array Tool from the Tools | Modeling menu.
3. Set the required number of elements in each dimension and their offsets in each direction. A preview of the array is shown in the modeling window (Figure 11.1 left).
4. Press Done to create the array (Figure 11.1 right).

![Array Tool](image)

Figure 11.1: Creating an array with the Array Tool

11.5 Modeling | Creating Bounding Box Points

The Creating Bounding Box Points tool creates points at the bounding box vertices of a selected solid.
The Bend tool enables the user to modify an entity or list of entities by bending it around an axis. Neutral Root, Bending Axis, and Bending Direction define a neutral plane for the bending operation. A neutral plane is the location where the material is not stretched or compressed during bending. The bending axis and bending direction should be perpendicular to each other. The cross product of these two vectors defines the positive and negative sides of the entity to be bent. The material above the neutral plane (along the bending direction) is compressed and the material below the neutral plane is stretched.

1. Select the entity to be modified.

2. Open the Bend tool from the Tools | Modeling menu.

3. Define the neutral plane by setting the following:
   - **Neutral Root**: position that defines the location of the neutral plane.
   - **Bending Axis**: vector that defines the rotational axis of the bending action.
   - **Bending Direction**: auxiliary vector that is used to define a bending plane.
   - **Bend Radius**: radius to the neutral plane.
   - **Bend Width**: width of the bend region.

4. Press Done to bend the selected object around the defined axis.

Figure 11.2: Bending a cube with the Bend tool
11.7 Modeling | Create Offset Sheet

The Create Offset Sheet tool enables the user to offset selected faces of a solid by an offset distance.

1. Open the Create Offset Sheet tool from the Tools | Modeling menu.
2. Select the faces to be modified.
3. Enter Offset distance.
4. Press Done to create the sheets.

11.8 Modeling | Create Shell from Body

The Create Shell from Body tool enables the user to create a hollow body from a solid. The user selects the solid which should form the outer shell, and the faces which should be excluded from the body, thus creating an opening into the shelled structure.

1. Select a solid.
2. Open the Create Shell from Body tool from the Tools | Modeling menu.
3. Select faces to be omitted.
4. Specify the Thickness of the shell.
5. Press Done to create the hollow body.
11.9 Modeling | Cover Edge Loop

The Cover Edge Loop tool enables the user to cover an open face of a selected object with a surface.

1. Open the Cover Edge Loop tool from the Tools | Modeling menu.

2. Select edges one by one to form a loop which is required to be covered by a face (Figure 11.3 left).

3. Press Done. The opening is covered with a surface (Figure 11.3 right).

Figure 11.3: Covering an opening of a selected object with a surface with the Cover Edge Loop tool
11.10 Modeling | Cover Wires with Sheets

The Cover Wires with Sheets tool enables the user to build surfaces which cover predefined polylines.

1. Open the Cover Wires with Sheets tool from the Tools | Modeling menu.

2. Select polylines to be covered (Figure 11.4 left). Check the Stitch resulting faces to stitch together the resulting faces that share common edges or a common face. Check the Use tolerant stitching to stitch the resulting faces if the common edges are inaccurate and not perfectly common. This will include a tolerance for defining the common edges.

3. Press Done to perform the operation (Figure 11.4 right).

Figure 11.4: Covering polylines with surfaces using the Cover Wires with Sheets tool
11.11 Modeling | Cylinder Tool

The Cylinder Tool enables the user to build a cylinder directly, instead of first modeling the circular cross-section and then extruding it.

1. Open the Cylinder Tool from the Tools | Modeling menu.

2. Set the Radius of the cylinder.

3. Define the Bottom and Top points of the cylinder, either by entering them manually or by selecting them in the model with the mouse.

4. A preview of the cylinder is shown in the modeler (Figure 11.5 left). Press Done to create the cylinder (Figure 11.5 right).

Figure 11.5: Creating a cylinder with the Cylinder Tool
11.12 Modeling | Display Bounding Boxes

The Display Bounding Boxes tool enables the user to display the bounding box of single or multiple parts in the model. This can be very useful, e.g., to access bounding box inconsistencies with CAD data, which normally can be repaired using Heal Selected:

1. Open Display Bounding Boxes from the Tools | Modeling menu.

2. The bounding boxes are shown as 3-D white boxes (see Figure 11.6). Check the Selected Objects Only to display bounding boxes only for the selected parts.

3. If either of the checkboxes in Create Bounding Box Vertices Of are set, then Data Points will be created on each of the bounding box corners when the Done button is pressed.

Tip: Change the background colour (File | Preferences) to an off-white or gray colour to ensure good visualization of the bounding boxes.

Figure 11.6: Display Bounding Boxes tool for a magic T waveguide
11.13 Modeling | Distance Tool

The Distance Tool enables the user to measure the distance between any two arbitrary points.

1. Open the Distance Tool from the Tools | Modeling menu.
2. Select the 2 points in the model between which you wish to calculate the distance. The distance is displayed on the top left corner of the dialogue and on the line between the 2 points.
3. Checking the Create point in the middle will create a point at the center of the line between the 2 points. Press Done to create the point and close the tool, press Next to create the point without closing the tool.

11.14 Modeling | Minimum Distance

The Minimum Distance Tool enables the user to measure the minimum distance between any two objects. It displays the minimum distance and can create two points at the endpoints of the minimum distance.

1. Open the Minimum Distance Tool from the Tools | Modeling menu.
2. Select the two objects that a minimum distance is required.
3. Indicate whether the points of minimum distance should be created in the model.
4. Click Done to create the points.
The Helix Tool can be used to build various complex helical configurations.

1. Open the Helix Tool from the Tools | Modeling menu.

2. Define the Bottom and Top points of the helix, either by entering them manually or by selecting them in the model with the mouse.

3. The following parameters can then be set for the helix: Radius: the radius of the helix; Distance between turns: the spacing between the turns of the helix; Wire radius: the radius of the wire used for the helix.

4. Other options include: Clockwise turned, Center top, Center bottom and Create splines only.

5. A preview of the helix is shown in the modeler (Figure 11.7 left). Press Done to create the helix (Figure 11.7 right).

Figure 11.7: Creating a helix with the Helix Tool
11.16 Modeling | Layerator

The Layerator tool enables the user to create a layered PCB in the Z-direction.

1. Select the solid that you want to be converted into a layered PCB.
2. Open the Layerator tool from the Tools | Modeling menu.
3. Enter the number of Layers that the PCB should have.
4. Enter the thickness of the PEC and dielectric layers (note that the number of layers multiplied by the sum of the PEC and dielectric layers MUST equal the height of the selected solid).
5. Indicate whether there should be a layer of PEC on the underside of the brick/PCB (note that this means that the PEC and dielectric layers MUST be thinner to fit the extra layer of PEC into the selected solid).
6. Indicate whether vias should be added to the PCB.
7. Indicate the number of vias in the X and Y directions.
8. Indicate the radius of the vias to be inserted (note that the diameter of the vias multiplied by the number of vias in a certain direction cannot be greater than the size of the selected solid in that direction).
9. Press 'Done' to create the desired object.
11.17 Modeling | Imprint

The **Imprint** tool enables the user to imprint curves of the intersection of two solids onto the faces of these solids. It calculates the curves of intersection of the two solids and imprints these curves onto the faces of the two solids. This results in the original faces being split either by breaking the face into several pieces or by adding slits to the face.

1. Select the solids.
2. Open the **Imprint** tool from the **Tools | Modeling** menu.
3. Press **Done** to make the imprints.

---

11.18 Modeling | Mirror Tool

The **Mirror Tool** enables the user to mirror the selected object with respect to the specified plane.

1. Select the object to be mirrored.
2. Open the **Mirror Tool** from the **Tools | Modeling** menu.
3. Specify the **Point on Plane** and **Normal Point** either by entering them manually or by selecting them in the model with the mouse. These two points will define the mirroring plane.
4. Enable **Make Copy** to mirror the copy of the selected object.
5. Press **Done** to create the mirrored object.
11.19 Modeling | Parts Library

The Parts Library tool shows a list of user defined .SAT files contained in a directory named Parts, which must be located in the InstallDirectory\GUI directory. This directory may contain sub-directories and the list of parts is categorized by the directory structure. To place a part into the Parts Library, export a complex or often used model as .SAT and copy it into the folder. To import a part into the current model:

1. Open the Parts Library tool from the Tools | Modeling menu.
2. Navigate to the required part to be inserted.
3. Select the required item’s checkbox.
4. Press Done to add the selected item to the model.

11.20 Modeling | Convert Polyhedron to Body

The Convert Polyhedron to Body tool enables the user to convert imported CAD faces into a solid ACIS body (e.g. for a .3ds import). This can take up a lot of memory as an ACIS face is generated for every triangle present in the imported file.

1. Open the Convert Polyhedron to Body tool from the Tools | Modeling menu.
2. Select the objects in the model that need to be converted into an ACIS body.
3. Press Done to perform the conversion.
The Space Warp tool enables the user to modify a body according to a given input law.

1. Select the body which will be modified.

2. Open the Space Warp tool from the Tools | Modeling menu.

3. Enter the Warp Law Function in the form of \( \text{vec}(f(x,y,z), g(x,y,z), h(x,y,z)) \)” and press Done.

Figure 11.8: Modifying a solid according to the warp law function \( \text{vec}(x, y + 3 \times (\sin(x \times 0.5)), z) \)
The Matching Circuit Tool tool enables the user to connect circuit model or load to a source.

1. Create two points that are the connection points for the source or generator. Name them "Gen_1" and "Gen_2".

2. Create two points that are the connection points of the antenna or load. Name them "Load_1" and "Load_2".

3. Open the Matching Circuit Tool from the Tools | Modeling menu.

4. Select the points that you have entered above, using the CTRL key for multiple selection.

5. Select the required matching circuit from the drop down list.

6. Press Done.
11.23 Modeling | Stitch

The Stitch tool enables the user to stitch together two ACIS bodies that share common edges or a common face. The tool then outputs one body, but maintains the common edge.

1. Open the Stitch tool from the Tools | Modeling menu.
2. Select the bodies in the model that share common edges and need to be stitched together.
3. If the common edges are inaccurate and not perfectly common, then check the Tolerant Stitching box. This will include a tolerance for defining the common edges.
4. Press Done to perform the Stitch operation.

11.24 Modeling | Create View Points

This tool creates two Points along the line of perspective in the modeling window.

11.25 Simulation | EM Multi-Simulation Setup

The EM Multi-Simulation Setup tool enables the user to automatically generate a number of simulations depending upon the number of sources available in the project. It uses a template project to build the other simulations and in each simulation only one of the sources is set active.

1. Select the template project.
2. Open the EM Multi-Simulation Setup tool from the Tools | Simulation menu.
3. Select the sources from the list to be excited and press Next to generate the number of simulations accordingly.
4. Press Done to close the window.
11.26 Simulation | Export Voxels

The Export Voxels tool enables the user to export voxel data from a selected simulation into specified files.

1. Select the simulation which has voxels to be exported.

2. Open the Export Voxels tool from the Tools | Simulation menu.

3. Specify Folder where the files are to be saved.

4. Specify File Names for the following:
   - X Axis, Y Axis and Z Axis: vectors of x-, y- and z-axis of the voxel data will be saved in these files.
   - Voxels: the index (i, j, k) and material properties of each voxel are saved in this file.
   - PEC edges: the index (i, j, k) and the direction of the PEC edges are saved in this file.
   - PMC edges: the index (i, j, k) and the direction of the PMC edges are saved in this file.

5. Check Add Comment Line box to add information lines at the beginning of each file to be saved, and specify the character in Begin Comment line With which will be added to the beginning of each comment line.

6. Select the Field Sensor that contains the voxels that need to be exported. If all the voxels in the simulation need to be exported, then select All Voxels.

7. Press Next to export the voxels.

8. To close the window, press Done.
The Remote Solver tool enables the user to run a simulation on a remote server, using an FTP connection.

1. Open the Remote Solver using FTP tool from the Tools | Simulation menu.
2. Enter the Remote host name of the server and the Remote login name.
3. Check Use selected simulation to launch the selected simulation on the remote solver. Alternatively, specify the Simulation Input file.
4. Check status only automatically streams the simulation log file to the console.
5. Press Done to connect to the server and to launch the simulation.
11.28 Simulation | Remote Solver Via SSH and SCP on Linux Machines

The Remote Solver Via SSH and SCP on Linux Machines tool enables the user to run a simulation on Linux machines.

1. Open the Remote Solver Via SSH and SCP on Linux Machines tool from the Tools | Simulation menu.

2. Enter the hostname in the Remote host name field. Specify the username in the Remote login name field. Specify the path on the remote machine where the input file should be copied and the simulation executed.

3. iSolve is the default solver name. Only change this field if the solver name has been changed on the remote machine.

4. For the simulation input file browse to the location of the input file on the local machine which will be copied and run.

5. Check Share folder when a shared folder is available for the remote machine and the input file is already located there i.e. the Simulation input file and Remote path are the same.

**Note:** Take care of text conventions for the remote linux host name and path: the host name should be written starting with \\.

**Tip:** Use a client like F-Secure to connect to the remote machine. In a terminal change directory (cd) to the remote folder, and type pwd to print the path. Copy the string and paste it into the remote path dialog in the tool and add / at the end.
11.29 Simulation | SAR Simulation Batch

The SAR Simulation Batch tool enables the user to automatically run a simulation, calculate the Spatial Peak Average SAR for various averaging masses, and write the values to a text file:

1. Open the SAR Simulation Batch tool from the Tools | Simulation menu.
2. Specify the name and location of the text file where the results should be stored.
3. Specify the name and location of the Simulation input file, the Sensor name which should be used for extracting the SAR and, if necessary, enter other averaging masses.
4. Press Done to start the simulation.

11.30 Simulation | Generate Simulation Batch File

Using this enables easy batching of simulations in the same .sem project file. To batch multiple simulations from different project files use the Generate Simulation Batch File tool:

1. Open the Generate Simulation Batch File tool from the Tools | Simulation menu.
2. Specify the name and location of the batch file for the simulations.
3. Select Run Batch File to run the simulations immediately. Alternatively, the batch file can be run later by double-clicking on it.
4. Specify the name and location of the Simulation Input Files.
5. Press Done to start the simulation.
The **Channel Setup** tool enables the user to set up various properties for mobile phone frequency band simulations.

1. Open the **Channel Setup** tool from the **Tools** | **Simulation** menu.
2. Specify the frequency band as either GSM or DCS.
3. Check **Separate Harmonic Simulations** if you want to run more than one harmonic simulation.
4. Check **Use aXware** if you want to use hardware acceleration to run your simulation.
5. In the **Model Solid Properties** box, specify in **Materials** the name of the model, in **Material Permittivity** its corresponding permittivity and in **Material Losses** its loss coefficient. You can specify several model properties by separating them by commas.
6. Press **Done**.
11.32 Simulation | T Multi-Simulation Setup

T Multi-Simulation Setup
The T Multi-Simulation Setup tool enables the user to automatically generate a number of thermal simulations depending upon the number of sources available in the template simulation. The resulting simulations can then be used in the Field Optimizer. The template will also then have a checkbox added to the Simulation Settings named MultiSim. This field indicates that the template simulation has child components linked to it.

1. Select the template thermal simulation.
2. Open the T Multi-Simulation Setup tool from the Tools | Simulation menu.
3. Select the sources from the list to be excited and press Next to generate the number of simulations accordingly.
4. Press Done to close the window.
The Brick Interpolation Tool enables the user to interpolate the result data from a selected sensor in a defined brick. The user can also specify the interpolation resolution.

1. Select the simulation in which the data from one of its sensor will be interpolated.

2. Open the Brick Interpolation Tool from the Tools | PostPro menu.

3. **Sensor Settings** are specified by setting following parameters:
   - **Sensor**: the sensor which has the data to be interpolated (all sensors available in the simulation will be displayed in a drop-down menu).
   - **Quantity**: the data to be interpolated (all available results on the specified sensor will be displayed in a drop-down menu).
   - **Open Slice Field Viewer**: the interpolated data will be displayed in a slice view if enabled.
   - **Auto-Adjust Preset**: if iSAR is selected, it ensures that the size of the brick and the interpolation resolution are appropriate to export the interpolated data into the iSAR environment.

4. Specify a corner and three edges of the interpolation brick by setting **Corner**, **Edge 1**, **Edge 2** and **Edge 3**. Note that Edges 1 and 2 are automatically adjusted to ensure the orthogonality.

5. Set the desired interpolation resolution on each edge using **Resolution along...**.

6. Press **Done** to perform the interpolation.
11.34 PostPro | Diffalyzer

The Diffalyzer tool enables the user to compare two sets of results on selected sensors in a defined brick. The user can also specify the resolution of the interpolated data.

1. Open the Diffalyzer tool from the Tools | PostPro menu.

2. Set Common Settings by specifying the following:
   - **Quantity**: the data to be compared (all available results on the specified sensor will be displayed in a drop-down menu).
   - **Absolute Difference**: the absolute value of the selected data will be calculated if checked.
   - **Open Slice Field Viewer**: the interpolated data will be displayed in a slice view if checked.

3. Specify two simulations and corresponding sensors which have the results to be compared. Set the following for Result A and Result B:
   - **Simulation**: the names of the simulations to be compared (all existing simulations with results will be displayed in a drop-down menu).
   - **Sensor**: the sensor which has the data to be compared (all sensors available in the selected simulation will be displayed in a drop-down menu).
   - **Weight**: the user can specify different weighting values for each result.

4. Specify a corner and three edges of the interpolation brick by setting Corner, Edge 1, Edge 2 and Edge 3. Note that Edges 1 and 2 are automatically adjusted to ensure the orthogonality.

5. Set the desired resolution on each edge using Resolution along....

6. Press Done to perform the required operation on the results.
The Field Combiner Evaluation tool can be used to combine the results of the same sensor but obtained in different simulations.

1. Open the Field Combiner Evaluation tool from the Tools | PostPro menu.
2. Choose the Field Sensor from the list of available sensors.
3. Choose the Quantity (the field to be combined) from the list.
4. Give a name to the output field using Combined Field Name box. Default is Combined Field.
5. Choose the Frequency from the list of extracted frequencies (if any).
6. Set Total Source Power which will be used in the normalization.
7. Enable Compute SAR(x,y,z) or Compute Avg SAR(x,y,z) to calculate the SAR or average SAR. In case of average SAR calculation, set Avg SAR Masses. Note that Compute Avg SAR(x,y,z) can be enabled when the Quantity is set to SAR(x,y,z,fo).
8. For each available simulation with results, set the Weighting Factor and Phase Shift Angle to perform the combine operation.
9. Press Done to perform the field combination.

11.36 PostPro | Field Combiner Simulation Setup

The Field Combiner Simulation Setup tool enables the user to automatically generate a certain number of simulations that will be used later by the Field Combiner Evaluation tool to combine available results. The template simulation with more than one source has to be prepared before using this tool. It copies the template simulation as many times as the number of sources available and in each copied simulation only one of the sources is set active. Open the tool from the Tools | PostPro menu and press Next to generate the simulations.
The purpose of the Hearing Aid Compatibility (HAC) extension is to enable measurements of the near electric and magnetic fields generated by wireless communication devices in the region controlled for use by a hearing aid in accordance with ANSI-C63.19-2001. As usually done in experimental assessment, electric and magnetic fields of a wireless device are scanned with free-space probes in a 5 by 5 cm area 10 mm above the acoustic point. The maximum field values in 9 sub-grids of the electrical and a magnetic field scan are evaluated automatically according to the rules defined in the standard and assigned a certain radiation class.

SEMCAD-X basically allows to perform the standard compliant postprocessing on the simulated data of a mobile device terminal.

How to use the HAC Tool:

1. Open an appropriate SEMCAD-X project including results of stored EM field data within a 3-D sensor (e.g., Overall Field). **Important:** The sensor must be large enough to include a volume as described above, located around the acoustic point of the device.

2. Open the HAC Tool from the Tools | PostPro menu.

3. Under HAC Settings: a) select the 3-D sensor which encloses the acoustic point; b) select RMS of either E- or H-field; c) specify both DUT and DUT Modulation Frequency.

4. Using one Corner and two Edge points, the appropriate 5 by 5 cm area 10 mm above the acoustic point can be defined by the user. The defined plane can be arbitrarily oriented in space and does not need to be aligned to the grid since SEMCAD-X uses spatial interpolation to extract the data onto the plane.

5. With the Resolution along... field, the interpolation resolution along the x and y axis can be defined.

6. Pressing Done and following the subsequent dialogues, the HAC extraction will be processed.

After the standard compliant extraction and processing of the simulated data, the results are shown. In addition to a visualization of the field, the exclusion of the three contiguous sub-grids on the perimeter, such as described in the ANSI-C63.19-2001 standard, is performed automatically. In order to allow a user defined selection of exclusion regions, SEMCAD-X also features a floating dialogue where the exclusion can be performed manually. The regions to be removed, are selected via checkbox (deactivate) and passed to the evaluation via the Update button. The automated exclusion can be re-activated by clicking on Auto. The outline of the actually shown field and exclusion will be automatically taken over to the printout.
The printout is subsequently performed using the **Generate Report...** item in the **Post Processor** menu or simply via the printer icon. Besides the common information, for Hearing Aid Compatibility Jobs automatic evaluation of **slot-averaged** results, exclusion of the three highest subgrids, application of the AWF factor (articulation weighting factor as defined in the ANSI-C63.19-2001 standard) as well as the assignment of the M category is automatically done by the SEMCAD environment.

**Note:** For printing of the HAC related report, please use an appropriate .htm print template from the Print_Templates folder on your CD.

![Figure 11.9: Hearing Aid Compatibility (HAC) postprocessing in SEMCAD-X.](image)
11.38 PostPro | Repack Result Files

The **Repack Result Files** tool allows the user to compress the result files in the current project. The level of compression can be set by the user. The minimum compression (0 - no compression) produces larger file sizes that load more quickly. The maximum compression (9) produces smaller file sizes that take longer to load.

1. Open the **Repack Result Files** tool from the **Tools | PostPro** menu.
2. Set the **Compression Level(0-9)**.
3. Check the **Eliminate Time-Domain Information** box to remove all time-domain data from the result file.
4. Press **Done** to perform the required action.

**Tip:** Running the Repack Result File tool with **Eliminate Time-Domain Information** can greatly reduce the size of result files for Broadband simulations with near-field or far-field sensors activated.
11.39 PostPro | Line Field Extractor Tool

The Line Field Extractor Tool is used to extract a 2-D plot from within a 3-D field sensor, along a Spline or Polyline that is fully enclosed by the field sensor. The Spline or Polyline must be defined in the model before the simulation is run.

1. After the simulation has run, right-click any sensor quantity, and select Send To Viewers. This initializes the simulation in the Postprocessor.

2. Open the Line Field Extractor Tool tool from the Tools | PostPro menu.

3. Select the Spline or Polyline which will be used for the line extraction and the Trajectory of Extraction.

4. For the Field Calculation value, select Absolute Value if the total field is required, Tangential to Line if the field component tangential to the trajectory at each sampling point is required, or Normal to Line if the field component normal to the trajectory at each sampling point is required.

5. Select the number of sampling points to take between each point on the Polyline, and enter it for Resolution per Segment. This value can have a maximum value of 1000.

6. If a Spline was selected, enter the number of segments that the Spline should be split into for Spline Sample Points. The Resolution per Segment would then be valid between these Spline Sample Points.

7. Select the checkbox for Extract EpsilonR and/or Extract SigmaE to extract the material information along the trajectory.

8. Select the checkbox of the field that needs to be extracted along the line.

9. Press Done to output the resulting 2-D plot.
11.40 PostPro | Load Result

The Load Result tool enables the user to load simulation results without having to open the .sem project file. This is very useful if the project contains a very detailed model and requires some time to open, or for comparing simulation results from different .sem project files:

1. Open the Load Result tool from the Tools | PostPro menu.

2. Browse to the output file of the simulation that you would like to evaluate. Press Next to get a list of all components that can be loaded.

3. Select the relevant check boxes for the components to be loaded.

4. Press Done to load the results into the postprocessor. The different viewers can then be called by right-clicking on the component in the list and selecting the desired Viewer.

**Note:** The default naming convention for the output file ends with _Output.h5.
11.41 PostPro | EMC/Antenna Efficiency/Gain(f)

The EMC/Antenna Efficiency/Gain(f) tool enables the user to plot the Total Efficiency, Radiation Efficiency and/or gain versus frequency of a broadband simulation. The E-Field at a specified distance from the source can also be calculated.

1. Select a simulation in the simulation tree with a broadband excitation, valid extraction frequencies and results.

2. Run the EMC/Antenna Efficiency/Gain(f) tool from the Tools | PostPro menu.

3. Select the properties that are required to be extracted and plotted.

4. Click Done.

**Note:** The extraction of the E-Field at 3m, for instance, is especially useful for EMC or EMI problems, and an example of this plot can be seen in the Figure beside.
11.42 PostPro | Thermo Field Combiner Evaluation

The Thermo Field Combiner Evaluation tool works in exactly the same way as the Field Combiner Evaluation tool, but works on thermal field results. You are also able to combine the field results from a single sensor at two different Time Index values. The number of Time Index values is defined by the number of snapshots that the user requested when setting up the Thermo-Sensor.

1. Open the Thermo Field Combiner Evaluation tool from the Tools | PostPro menu.

2. Choose the Field Sensor from the list of available thermo-sensors.

3. Choose the Quantity (the field to be combined) from the list.

4. Give a name to the output field using Combined Field Name box. Default is Combined Thermo Field.

5. Enter the Weighting Factor and required field snapshot in the Time Index. The same field can be added to itself at different time snapshots, if required.

6. Press Done to perform the field combination.
11.43 Device Modeling | Create Simple Flexible PCB

The Create Simple Flexible PCB enables the user to specify the dimensions and position of a simple flexible PCB object.

1. Open the Create Simple Flexible PCB tool from the Tools | Device Modeling menu.
2. In PCB Dimensions, specify the width, height and length of the object.
3. In PCB Lines, specify the number of lines and their size.
4. In PCB Bending, check Bend PCB if you want to specify a bending radius. The radius can be entered in the Bending Radius field.
5. In Model Options check Unite all PEC lines if you want to join all the PEC lines of the PCB.
6. Press Done to create the object.

11.44 PostPro | Import From TXT File

The Import From TXT File tool is used to import near-field data obtained from a measurement or a different simulation platform.

1. Open the Import From TXT File tool from the Tools | PostPro menu and browse to the text file to import.
2. If the file contains a header, check the Header Row checkbox.
3. Use the Scale Coordinates and/or Scale Values fields to scale the coordinate units or scale the field respectively.
4. Note: the first 3 columns in the file should be the x, y, z coordinates and the remaining columns contain the field data with columns separated by space or tab.
The Circular Waveguide tool allows user to automatically build a circular waveguide source and its corresponding sensors for given waveguide and source/sensor parameters.

1. Open the Circular Waveguide tool from the Tools | Waveguides menu.

2. Enter Radius of the circular waveguide.

3. Check Build Geometry, Source and Sensors

4. Set the following parameters: Direction: the direction of the port; Source-Sensor Distance: the distance between the source and the first sensor; Sensor Spacing: the distance between the sensors.

5. Set the position of the circular waveguide source by entering the coordinates of the Center Point manually or by selecting it in the model with the mouse.

6. Press Done to build the sources and the sensors in the model.
11.46 Waveguides | Coaxial Line

This tool enables the user to calculate either the impedance of a coaxial line, or the dimensions for a given impedance.

1. Open the Coaxial Line tool from the Tools | Waveguides menu.

2. To Calculate Impedance: enter Rel. Permittivity, Inner and Outer Radius; press Enter.

3. To Synthesize Impedance: enter Rel. Permittivity, Impedance and enter the Outer Radius (with Calculate Inner Radius checked) or the Inner Radius; press Enter.

4. Check Build Geometry, Source and Sensors, enter the Direction of the coaxial and enter the Center Point manually or by selecting it in the model with the mouse; press Done. A Waveguide Source is added and two profiles for the inner and outer conductors are added (Figure 11.10). Use the Extrude function to create the coaxial line from the two profiles.

Figure 11.10: Coaxial line representation (left) and modeled source, sensors and profiles (right)
11.47 Waveguides | Parallel Plate Waveguide

The Parallel Plate Waveguide tool enables the user to automatically build a rectangular parallel plate waveguide source and its corresponding sensors for given waveguide and source/sensor parameters.

1. Open the Parallel Plate Waveguide tool from the Tools | Waveguides menu.

2. Set the following parameters: Waveguide Width (a); Waveguide Height (b); Polarization Direction of the source.

3. To build the source and the sensors:
   - (a) Check Build Geometry, Source and Sensors box.
   - (b) Set the following parameters: Direction: the direction of the port; Source-Sensor Distance: the distance between the source and the first sensor; Sensor Spacing: the distance between the sensors.
   - (c) Set the position of the rectangular waveguide source by entering the coordinates of the Center Point manually or by selecting it in the model with the mouse.

4. Press Done to build the sources and the sensors in the model.
The Rectangular Waveguide tool enables the user to automatically build a rectangular waveguide source and its corresponding sensors for given waveguide and source/sensor parameters. It also calculates the propagation constants and cut-off frequencies of the first five modes of the rectangular waveguide for a given frequency, and displays whether the mode is propagating or evanescent.

1. Open the Rectangular Waveguide tool from the Tools | Waveguides menu.

2. Set the following parameters: Rel. Permittivity: the relative permittivity of the medium; Waveguide Width (a); Waveguide Height (b); Frequency.

3. Press Enter to make calculations for the propagation constants and the cut-off frequencies.

4. To build the source and the sensors:
   
   (a) Check Build Geometry, Source and Sensors box.

   (b) Set the following parameters: Direction: the direction of the port; Source-Sensor Distance: the distance between the source and the first sensor; Sensor Spacing: the distance between the sensors.

   (c) Set the position of the rectangular waveguide source by entering the coordinates of the Center Point manually or by selecting it in the model with the mouse.

5. Press Done to build the sources and the sensors in the model.
11.49 Calculators | Conductivity/Loss Tangent Calculator

The tool enables the user to calculate the conductivity from the loss tangent or vice versa.

1. Open the Conductivity/Loss Tangent Calculator tool from the Tools | Calculators menu.

2. Enter the Rel. Permittivity of the dielectric and the Frequency.

3. To calculate the Conductivity enter the Loss Tangent with the Calculate Conductivity checkbox checked; press Enter.

4. To calculate the Loss Tangent uncheck Calculate Conductivity and enter the Conductivity; press Enter.

5. To close the window, press Done.
11.50 Calculators | Linear Dispersive Constants Tool

The Linear Dispersive Constants Tool is used to calculate the constants needed to define a Debye dispersive material. It enables the user to define dispersive constants based on two $\epsilon_r$ and $\sigma$ values at two different frequencies. The tool provides an approximation to a linear dispersive material between the frequencies provided. It can also be used to copy these calculated values through to the model and to plot $\epsilon_r(f)$ and $\sigma(f)$ for a selected Solid Region.

1. Select the relevant simulation containing the Dispersive materials and open the Linear Dispersive Constants Tool from the Tools | Calculators menu. In order to use the plotting or copy-through functionality, at least one Solid Region in the model must be defined as a Debye dispersive material.

2. The Start Frequency and Stop Frequency are calculated by default, based on the Frequency and Bandwidth specified in the simulation. These fields can be edited, if required. The frequencies do not have to correspond to the minimum and maximum frequencies of the simulation.

3. The user must enter the Start Permittivity and Start Conductivity of the Solid Region for the first or lower frequency, and the Stop Permittivity and Stop Conductivity of the Solid Region for the second for higher frequency. These lower and higher frequencies must correspond to the Start Frequency and Stop Frequency defined above.

4. If the user merely wants to calculate the dispersive material constants, then press Next.

5. If the user wants to copy the calculated values through to a Solid Region in the model or plot the values of $\epsilon_r(f)$ and $\sigma(f)$, then Copy Values to Solid and Plot Dispersive Values must be selected.

6. The user must Select the Dispersive Material from the drop down list that need to have values copied through to, or plotted.

7. Press Next to perform all the selected options. This can be done recursively.

8. Press Done to close the Tool.
The Dispersive Material Tool is used to plot $\varepsilon_r(f)$, $\mu_r(f)$, electric $\sigma(f)$ and magnetic $\sigma(f)$ for Dispersive materials or Metamaterials.

1. Select the relevant simulation containing the Dispersive materials and open the Dispersive Material Tool from the Tools | Calculators menu.

2. The Frequency Min, Frequency Max and Frequency Step are calculated by default, based on the Frequency and Bandwidth specified in the simulation. These fields can be edited.

3. Select the material to evaluate from the Dispersive Materials drop down list.

4. Press Done to plot the data.
11.52 Calculators | Microstrip Line Calculator

The Microstrip Line Calculator tool enables the user to calculate the impedance of a lossless microstrip line for given dielectric properties, strip and substrate dimensions, or to calculate the width of a lossless microstrip for a given impedance. In either case, the guided wavelength is also computed and displayed.

1. Open the Microstrip Line Calculator tool from the Tools | Calculators menu.

2. To calculate the impedance:
   (a) Set the following parameters: Rel. Permittivity: the relative permittivity of the substrate \( \varepsilon_r \); Diel. Thickness: the thickness of the substrate \( d \); Strip Thickness: the thickness of the microstrip line \( t \); Frequency; Strip Width: the width of the microstrip line \( w \).
   (b) Press Enter to calculate the changes.

3. To calculate the width:
   (a) Uncheck Compute Impedance box.
   (b) Set the following parameters: Rel. Permittivity: the relative permittivity of the substrate; Diel. Thickness: the thickness of the substrate; Frequency; Impedance. Note that Strip Thickness is automatically set to zero to compute (synthesize) the width for the given impedance.
   (c) Press Enter to calculate the changes.

4. To close the window, press Done.
The Thin Resistive Sheet Calculation Tool calculates the transmission and reflection coefficients which will be used in the TRS algorithm (see Section 2.7). The tool can also be used to transfer material parameters to the simulation settings.

1. Select the relevant simulation containing the thin resistive sheets and open the Thin Resistive Sheet Calculation Tool from the Tools | Calculators.

2. A list of all TRS parts in the simulation are shown on the Solids list. The Frequency is extracted from the simulation settings.

3. Enter the sheet Thickness, Conductivity and Relative Permittivity: the Reflection, Transmission coefficients and Expected Error are calculated.

4. If the sheet has dielectric parts on either side, enter the parameters into the Upper and/or the Lower Region Coefficients to include the dielectrics effects on Reflection and Transmission Coefficients.

5. To transfer the relevant coefficients, select the TRS parts in the Solids list enter the sheet Thickness, Conductivity, Rel. Permittivity, Rel. Permeability and press Done.
The Wavelength Calculation Tool enables the user to calculate the wavelength and the skin depth for given medium properties and frequency.

1. Open the Wavelength Calculation Tool from the Tools | Calculators menu.

2. Set the following parameters: Rel. Permittivity: the relative permittivity of the medium; Rel. Permeability: the relative permeability of the medium; El. Conductivity: the electrical conductivity of the medium; Frequency.

3. Press Enter to calculate the wavelength and the skin depth.

4. To close the window, press Done.
11.55 Local Modeling | Blend Edges

The **Blend Edges** tool enables the user to blend any selected edge. The edges presented for blending should not form disjoint networks. Each of the edges has an associated blend radius.

1. Open the **Blend Edges** tool from the **Tools | Local Modeling** menu.

2. Select the edges to be blended. Use **Rubberband Selection** to select multiple edges at the same time.

3. Specify the **Blend Radius** and press **Done**.

11.56 Local Modeling | Chamfer Edges

The **Chamfer Edges** tool enables the user to generate chamfers between surfaces configured in such a way that the chamfer surface can be represented by a plane or a cone. The size and orientation of the chamfer are defined by specifying a range for each support. This range is defined to be the distance in the support between the edge being chamfered and the contact point on the support where the chamfer intersects the support.

1. Open the **Chamfer Edges** tool from the **Tools | Local Modeling** menu.

2. Select the edges to generate chamfer surfaces. Use **Rubberband Selection** to select multiple edges at the same time.

3. Specify the **Left Range** and **Right Range**, and then press **Done**.

![Figure 11.11: Blended (left) and chamfer blended (right) edges of a cube](image)
11.57 Local Modeling | Move Faces

The Move Faces tool enables the user to move selected faces through a transform which replaces the surfaces of the specified faces with the surfaces that are moved by the transform. The translation vector of the transform must be specified.

1. Open the Move Faces tool from the Tools | Local Modeling menu.

2. Define the Starting Point of Direction and Ending Point of Direction of the movement, either by entering them manually or by selecting them in the model with the mouse.

3. Select the faces to move. Use Rubberband Selection to select multiple faces at the same time.

4. Specify the length of the offset through Offset along Direction.

5. Press Done to move the selected faces along specified direction.

Figure 11.12: Moving selected faces of a blended cube
11.58 Local Modeling | Sweep Faces

The **Sweep Faces** tool enables the user to move the selected faces along a path defined by the faces adjacent to those selected. This is especially useful for shortening a curved slot by moving the end face(s) along the curved path of the slot.

1. Open the **Sweep Faces** tool from the Tools | Local Modeling menu.
2. Select the faces to be swept. Use **Rubberband Selection** to select multiple faces at the same time.
3. Specify the length of the sweep through **Sweep Distance**.
4. Press **Done** to sweep the selected faces.

11.59 Parametrized Modeling | Edit Parametrized

The **Edit Parametrized** tool enables the user to define the behavior and available options for a parametrized object in the model.

1. Open the **Edit Parametrized** tool from the Tools | Parametrized Modeling menu.
2. Select a parametrized solid from the Part list that is to be edited. The relevant parameters for the solid will be displayed. Check **Show Transformations** box to display the Translation and Rotation parameters.
3. Check the Visible boxes of the corresponding parameters that should be visible to the user.
4. For the Dependency field, enter a constant, an equation, another parameter’s value, or combination thereof, that the parameter is dependent on. If a value is entered into this field, **SEMCAD-X** will, by default, uncheck the Visible field.
5. Press **Done** to save the required changes.

The Dependency field will be calculated after any change in the parametrized solid.
The Cylinder Cone tool enables the user to build a parametrized cone.

1. Open the Cylinder Cone tool from the Tools | Parametrized Modeling menu. A parametrized cone with default values will be built automatically.

2. The following parameters can then be set for the cone: the MajorRadius and MinorRadius: the minor and major radius of the cone base; TopRadius: the radius of the cone top circle; Height: the height of the cone.

Figure 11.13: Parametrized cone with TopRadius=5 mm (left) and with TopRadius=0 mm (right)
The Turtle Polyline tool enables the user to build a parametrized polyline consisting of a number of segments which are defined by setting their lengths and their angle offsets.

1. Open the Turtle Polyline tool from the Tools | Parametrized Modeling menu.

2. Set the number of segments in Count data field. Note that, it is set to 5 as default.

3. For each segments, specify the length through WalkDistance and set the angle offset with respect to the previous segment in AngleOffset data field.

4. Enable Closed to obtain a closed polyline (see Figures below).

5. Enable Spline to obtain a spline instead of a polyline with straight lines.

![Figure 11.14: Modeling open and closed parametrized turtle polylines](image)
11.62 Parametrized Modeling | Create Combined

The Create Combined tool works in the same way as the regular Unite, Subtract and Intersect tools. A new solid is created based on the selected interaction between two or more selected parametrized objects.

1. Open the Create Combined tool from the Tools | Parametrized Modeling menu.
2. Select at least two parametrized bodies that must be combined.
3. Select the Operation that must be performed. This could be a Unite, Subtract or Intersect operation between the selected solids.
4. Check the Keep Operands box if the original parametrized parts are to be kept in the model after the combine function has been performed and the newly created solid added.
5. Press Done to perform the required action.

11.63 Parametrized Modeling | Create Extruded

The Create Extruded tool enables the user to create a parametrized solid by extruding a parametrized shape along the z-axis.

1. Select a parametrized shape.
2. Open the Create Extruded tool from the Tools | Parametrized Modeling menu.
3. Check Keep Shape box in order to keep the parametrized shape in the model after the parametrized object is created. Press Done.
11.64 Parametrized Modeling | Create Rotated

The Create Rotated tool enables the user to create a parametrized solid by rotating a parametrized shape along the \( y \)-axis.

1. Select a parametrized shape.

2. Open the Create Rotated tool from the Tools | Parametrized Modeling menu.

3. Check Keep Shape box in order to keep the parametrized shape in the model after the parametrized object is created. Press Done. By default, the parametrized shape is rotated 360°.
11.65 Parametrized Modeling | Create Sweep

The Create Sweep tool enables the user to form a parametrized object by sweeping a parametrized closed profile along a parametrized rail. There are restrictions about the relative positions of the profile and the rail. Sweeping works best if the rail starts perpendicular to the profile plane, inside the profile. Both the profile and the rail must be single, contiguous curves.

1. Select the parametrized profile and the parametrized rail along which to sweep.
2. Open the Create Sweep tool from the Tools | Parametrized Modeling menu.
3. Check Keep Shapes box in order to keep the profile and the rail in the model after the parametrized object is created.
4. Press Done to create the parametrized solid.

Figure 11.15: Modeling a parametrized object by sweeping a profile along a rail
11.66  Parametrized Modeling | Update Entity

The Update Entity tool enables the user to update any parametrized object geometry manually after any of its parameters have been changed.

11.67  Parametrized Modeling | Update All Parameterized Entities

The Update All Parameterized Entities tool enables the user to update all the parametrized objects’ geometry manually after parameters have been changed.

11.68  S-Parameter | S-Parameter Evaluation from Z-Matrix

The S-Parameter Evaluation from Z-Matrix tool enables the user to extract the full S-parameter matrix of the simulated structure from the Z-matrix and then plot them in 2-D. It is designed for only edge source excitation and all the edge sources in the project have to be active in the template simulation in order to be able to use this tool. The tool automatically detects the simulations which are produced using S-Parameter Simulation Setup tool and then uses the simulation results to extract full S-parameter matrix from Z-matrix. The user can then select the S-parameters to be displayed as a 2-D plot once the evaluation is finished. In addition, full S-parameters and Z-matrix elements versus frequency are saved in separate text files under the same folder where the project is.

11.69  S-Parameter | S-Parameter Evaluation

The S-Parameter Evaluation tool enables the user to extract and display the S-parameters of the simulated structure. The tool automatically detects the simulations with results and the ports with corresponding sensors. In order to be able to use this tool, there has to be at least one completed simulation which was appropriately generated by the S-Parameter Simulation Setup tool. More details of S-parameter evaluation are found in Chapter 10.
1. Open the S-Parameter Evaluation tool from the Tools | S-Parameter menu.

2. If the model contains the waveguide sources at ports, set the dielectric properties of waveguiding structures at ports by selecting the solids in the window, and then press Next, otherwise, go to the next step.

3. Select Evaluation Type from drop down menu. $|S|$ in dB: magnitude of S-parameters in dB; $|S|$ linear: magnitude of S-parameters; $P$ in: input power; $P$ reflected: reflected power; $P$ transmitted: transmitted power; Phase of $S$: phase of S-parameters. To display the modes excited at ports, select Show Modes.

4. Select the S-parameters to be evaluated from the list displayed.

5. If you want to store the extracted S-parameters,
   (a) Check Export to CITI to export the S-parameters in CITI format, and then specify the name and location of the file where the results are to be stored.
   (b) Check Export to Touchstone to export the S-parameters in Touchstone format, and then specify the name and location of the file where the results are to be stored.
   (c) Check use cache to save the S-parameters as text files in the same folder as the project.

Press Done.

6. Another window will appear showing the list of the extracted S-parameters once the computation is completed.

7. Select the S-parameters from the list for the 2-D plot. Multiple parameters can be selected by holding CTRL while selecting the item. Press Invert button to invert the selection. To select all parameters, press All, and press None button to reset the selection.

8. Press Next to see the plots of chosen S-parameters.
11.70 S-Parameter | S-Parameter Simulation Setup

The S-Parameter Simulation Setup tool enables the user to automatically generate a number of simulations necessary to extract the S-parameters of the structure. A template simulation with at least one port with appropriate sensors and one source must be prepared before using this tool. The details of S-parameter extraction procedure can be found in Chapter 10.

1. Select the template simulation.
2. Open the S-Parameter Simulation Setup tool from the Tools | S-Parameter menu.
3. Select the ports to be excited
4. Press Next to generate the necessary simulations
5. To close the window, press Done.
The Stand Alone S-Parameter Evaluation tool enables the user to extract and then display S-parameters of any completed simulation with only one active edge source. The tool automatically detects all sensors present and displays a possible set of S-parameters that can be extracted from the simulation. In order to be able to use this tool, there has to be at least one simulation with results and only one active source. The details of S-parameter extraction procedure can be found in Chapter 10.

1. Select the simulation with results and with only one active source.

2. Open the Stand Alone S-Parameter Evaluation tool from the Tools | S-Parameter menu.

3. Select the S-parameters to be evaluated from the list displayed.

4. If you want to store the extracted S-parameters,
   (a) Check Export to CITI to export the S-parameters in CITI format, and then specify the name and location of the file where the results are to be stored.
   (b) Check Export to Touchstone to export the S-parameters in Touchstone format, and then specify the name and location of the file where the results are to be stored.

5. Press Done and another window will appear showing the list of the extracted S-parameters.

6. Select the S-parameters for the 2-D plot. Multiple parameters can be selected by holding CTRL when the item is chosen. Press Invert button to invert the selection. To select all parameters, press All, and None button will reset the selection.

7. Press Next to see the plots of chosen S-parameters.
Chapter 12

Scripting Environment

12.1 Overview

SEMCAD-X features a scripting environment which allows the user to process the entire set of operations which are normally done via the GUI, performed via a text based script. This ranges from generating a model based on analytical formulas; generating the grid, assigning settings and running the solvers, up to the entire data extraction and postprocessing.

Due to its popularity, the Python scripting language has been chosen as the interface. Please see the following for further reference:

- [http://www.python.org/](http://www.python.org/)
- [http://docs.python.org/tut/tut.html](http://docs.python.org/tut/tut.html)

In addition, extensive printed literature is available regarding the Python scripting language.

12.2 Accessing the Scripting Environment

In a first step, the Scripting Bar can be activated via selection in the menu item View. Subsequently, either by clicking into the Scripting Bar and key commands Ctrl-O or via the menu item Script, an existing script can be opened. Additional commands such as for the generation of a new script are also accessible via this menu item.

As soon as a script has been loaded or generated, it can be started via keyboard F5 or via the menu. As shown in Figure 12.3 for a modeling example, the script is processed and the output of the scripting engine is directed to the Console Bar.

Note: A detailed reference on the provided SEMCAD-X scripter commands is available in Appendix C.

Note: Please consult the SEMCAD-X Tutorial for further information on the scripting engine. The Tutorial as well as the related Tutorial Examples were enriched by a number of scripting examples. In addition, a set of scripts is provided within the SEMCAD-X directory generated by the application installer.
12.2. Accessing the Scripting Environment

Figure 12.1: Activation of the Scripting Bar in SEMCAD-X

Figure 12.2: Scripting related actions accessible via the Script menu.

Figure 12.3: Generation of a analytical function based model using the scripter.
Chapter 13

References

13.1 References


### Appendix A

#### Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-D</td>
<td>One dimensional</td>
</tr>
<tr>
<td>2-D</td>
<td>Two dimensional</td>
</tr>
<tr>
<td>3-D</td>
<td>Three dimensional</td>
</tr>
<tr>
<td>ABC</td>
<td>Absorbing Boundary Condition</td>
</tr>
<tr>
<td>ADI</td>
<td>Alternating Direction Implicit</td>
</tr>
<tr>
<td>B</td>
<td>Magnetic induction</td>
</tr>
<tr>
<td>CAD</td>
<td>Computer Aided Design</td>
</tr>
<tr>
<td>CFL</td>
<td>Courant Friedrich Levy (stability number)</td>
</tr>
<tr>
<td>D</td>
<td>Dielectric displacement current</td>
</tr>
<tr>
<td>dP/dV</td>
<td>Loss power density</td>
</tr>
<tr>
<td>E</td>
<td>Electric field</td>
</tr>
<tr>
<td>EM</td>
<td>Electromagnetic</td>
</tr>
<tr>
<td>FDTD</td>
<td>Finite-Difference time-domain</td>
</tr>
<tr>
<td>H</td>
<td>Magnetic field</td>
</tr>
<tr>
<td>HDF</td>
<td>Hierarchic Data Format</td>
</tr>
<tr>
<td>HTML</td>
<td>Hyper Text Markup Language</td>
</tr>
<tr>
<td>I</td>
<td>Current</td>
</tr>
<tr>
<td>J</td>
<td>Current density</td>
</tr>
<tr>
<td>P</td>
<td>Power</td>
</tr>
<tr>
<td>PEC</td>
<td>Perfect Electric Conductor</td>
</tr>
<tr>
<td>Phasor</td>
<td>Complex representation of a harmonic amplitude and its phase</td>
</tr>
<tr>
<td>PMC</td>
<td>Perfect Magnetic Conductor</td>
</tr>
<tr>
<td>PML</td>
<td>Perfectly Matched Layer</td>
</tr>
<tr>
<td>PP</td>
<td>Post Processing</td>
</tr>
<tr>
<td>RMS</td>
<td>Root Mean Square</td>
</tr>
<tr>
<td>S</td>
<td>Pointing vector</td>
</tr>
<tr>
<td>S11</td>
<td>Reflection coefficient S11</td>
</tr>
<tr>
<td>SAR</td>
<td>Specific Absorption Rate</td>
</tr>
<tr>
<td>SEMCAD</td>
<td>Simulation Platform for Electromagnetic Compatibility, Antenna Design</td>
</tr>
<tr>
<td>SPEAG</td>
<td>Schmid &amp; Partner Engineering AG</td>
</tr>
<tr>
<td>STL</td>
<td>Stereolithography</td>
</tr>
<tr>
<td>SWR</td>
<td>Standing Wave Ratio</td>
</tr>
<tr>
<td>TE</td>
<td>Transverse Electric propagation mode</td>
</tr>
<tr>
<td>TEM</td>
<td>Transverse Electromagnetic propagation mode</td>
</tr>
<tr>
<td>TFSF</td>
<td>Total Field - Scattered Field</td>
</tr>
<tr>
<td>TM</td>
<td>Transverse Magnetic propagation mode</td>
</tr>
<tr>
<td>----</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>U</td>
<td>Voltage</td>
</tr>
<tr>
<td>V</td>
<td>Volume</td>
</tr>
<tr>
<td>Z</td>
<td>Impedance</td>
</tr>
</tbody>
</table>
Appendix B

Command Line Kernel 

The SEMCAD-X command line kernel iSolve allows an FDTD simulation to be run independently of the graphical user interface. This is very suitable for running simulations where the model occupies a large amount of memory: by running it through command line the memory that the model would occupy is freed and can be used by the solver.

B.1 Running SEMCAD-X projects with iSolve

Before running the simulation the user must first create an input file which is used by the solver. Once all the relevant simulation settings have been entered and the grid and voxels are made, right click on the simulation name and select Write Solver Input File. Now the simulation can be started either from a DOS command window or using a batch file.

To start the simulation in DOS iSolve must be called from the directory where it is installed (or the path must be specified).

It is also possible to start single or multiple simulations using a batch file. There is an example of such a batch file in the directory where iSolve is installed (normally C:\Program Files\SEMCAD X\Solvers\iSolve\iSolve\batch.cmd). Right click on the file and select edit to change the runpaths that are specified in the batch file to point to the simulations that should be run. Save and close the file and double click on it to run the batch.

B.2 iSolve Options

iSolve can be started with the following command line options:

- \( \texttt{-t} \): Test if license server is running and a valid license is available. This option can be used to check the license without having to start a simulation.
- \( \texttt{-c file} \): Use \texttt{file} as license file.
B.2. iSolve Options
Appendix C

SEMCAD X Python Reference

C.1 Global functions

`str browse_file()`

Opens a file browse dialog and lets the user choose a file.
_Return value_: returns full path to the chosen file.

`clear_console()`

Clears the output console.

`str input(str prompt)`

Opens a dialog and asks user for a string which is evaluated as Python code.
_Return value_: returns result of evaluated input string.

`str raw_input(str prompt)`

Opens a dialog and asks user for a string.
_Return value_: returns input string.

_Model model_

Global variable to access the current `Model` object in SEMCAD.

_Simulations simulations_

Global variable to access the current `Simulations` object in SEMCAD.

Example:

```
# Iterate through all simulations and print their names
# and check if they have results:
```
for S in simulations:
    if S.HasResults():
        print S.Name, ' has results'

C.2 Module SEMCAD

Data GetData() 

*Return value:* returns the data root of the project currently open in SEMCAD.

SetModelingMode() 

Sets modeling mode in SEMCAD. Make sure to call this function before making any changes to the CAD model in a SEMCAD project.

SetSimulationMode() 

Sets simulation mode in SEMCAD. Make sure to call this function before making any changes to the simulation settings.

SynSimsWithModel() 

Makes sure that all simulations in the project are synchronized to the current model. It is called automatically by SetSimulationMode().

SetViewerMode() 

Sets viewer mode in SEMCAD by switching to the Viewer panel.

OpenProject(str file_name) 

Opens a SEMCAD X project file.

NewProject() 

A new unnamed project will be created.

SaveProjectAs(str file_name) 

Saves the current state of the project to the specified filename.

SaveProject() 

Saves the current state of the project.
str GetProjectPath()

Return value: returns the full path of the project currently open in SEMCAD X.

Quit()

Closes the program.

UpdateView()

Updates the 3-D view.

SaveScreenShotAs(str file name)

Makes a screen shot of the current 3-D view and saves it to a TIFF file.

Return value: returns True in case of success.

bool IsRunning()

Return value: returns True if a simulation is running inside SEMCAD X.

C.2.1 class Data

Simulations Data.GetSimulations()

Return value: returns container of all simulations of the current SEMCAD project.

Model Data.GetModel()

Return value: returns interface to the current SEMCAD modeler.

C.2.2 class Model

Body Model.CreateBrick(Position P0, Position P1)

Creates a brick given by its 2 corner positions P0 and P1.

Return value: returns a reference to the newly created brick body entity.

Body Model.CreatePolyline(list positions)

Creates a poly-line body given by a Python list of Positions \([P_1, P_2, \ldots, P_N]\).

Return value: returns a reference to the newly created polyline body entity.
Body Model.CreateSweptBody(Body profile, Body rail)

Creates a swept body by sweeping the profile body along the rail body. profile and rail have to be poly-line entities.

Return value: returns a reference to the newly created swept body entity.

Point Model.CreatePoint(Position P)

Return value: returns a reference to the newly created point entity.

Body Model.CreateSphere(Position center, float radius)

Creates a sphere body centered around the position center with radius radius.

Return value: returns a reference to the newly created sphere body entity.

Body Model.CreateRotationBody(Body profile, Position P0, Position P1, float angle)

Creates a new body by rotating the profile around the axis given by the two positions P0 and P1 by the given angle.

Return value: returns a reference to the newly created rotation body entity.

Body Model.CreateExtrudedBody(Body profile, Position P0, Position P1)

Creates a new body by extruding the profile along the axis given by the two positions P0 and P1.

Return value: returns a reference to the newly created extruded body entity.

Body Model.CreateSkinBody(list list_of_profile_bodies)

Creates a new body by putting a skin over a Python list of profiles.

Return value: returns a reference to the newly created skin body entity.

Body Model.CreateArc(Position center, Position P0, Position P1)

Creates an arc body centered around the position center going from Position P0 to Position P1.

Return value: returns a reference to the newly created arc body entity.

Group Model.CreateGroup()

Creates a group entity.

Return value: returns a reference to the newly created group entity.

Body Model.CreateSpline(list positions)

Creates a spline-curve body given by a Python list of Positions \([P_1, P_2, \ldots, P_N]\).

Return value: returns a reference to the newly created spline body entity.
**Entity** *Model.CreateFieldSensor*(**Position** *P0*, **Position** *P1*)

Creates a field sensor given by its 2 corner positions *P0* and *P1*.

*Return value*: returns a reference to the newly created sensor entity.

**Entity** *Model.CreateEdgeSensor*(**Position** *P0*, **Position** *P1*)

Creates an edge sensor given by its 2 extremities *P0* and *P1*.

*Return value*: returns a reference to the newly created sensor entity.

**Entity** *Model.CreateVoltageSensor*(**Position** *P0*, **Position** *P1*)

Creates a voltage sensor given by its 2 extremities *P0* and *P1*.

*Return value*: returns a reference to the newly created sensor entity.

**Entity** *Model.CreateCurrentSensor*(list positions)

Creates a current sensor given by a Python list of Positions \([P_1, P_2, \ldots, P_N]\) defining a loop.

*Return value*: returns a reference to the newly created sensor entity.

**Entity** *Model.CreatePortSensor*(**Position** *P0*, **Position** *P1*, list loop positions)

Creates a port sensor given by a voltage sensor between *P0* and *P1* and a current loop given by a list of Positions \([P_1, P_2, \ldots, P_N] \).  

*Return value*: returns a reference to the newly created sensor entity.

**Entity** *Model.CreateEdgeSource*(**Position** *P0*, **Position** *P1*)

Creates an edge source given by its 2 extremities *P0* and *P1*.

*Return value*: returns a reference to the newly created source entity.

**Entity** *Model.CreateWaveguideSource*(**Position** *P0*, **Position** *P1*)

Creates a waveguide source given by its 2 corners *P0* and *P1*.

*Return value*: returns a reference to the newly created source entity.

list *Model.CreateVertices*(Entity *ent*)

Extracts all vertices from a given geometric entity as a list of points.

*Return value*: returns a list of created point entities.

list *Model.GetVertices*(Entity *ent*)

*Return value*: returns list of positions of crucial vertices of an entity.
list Model.GetPoints(Entity ent, float step)

Return value: returns list of positions of a sample of points from an entity (like a spline), spaced by step intervals.

Point Model.AsPoint(Entity ent)

Tests if the given entity ent is a point and returns a reference to it as a point.

Return value: returns the reference to a point in case the entity ent is a point and None in case ent is not a point.

Body Model.AsBody(Entity ent)

Tests if the given entity ent is a body and returns a reference to it as a body.

Return value: returns the reference to a body in case the entity ent is a body and None in case ent is not a body.

Group Model.AsGroup(Entity ent)

Tests if the given entity ent is a group and returns a reference to it as a group.

Return value: returns the reference to a group in case the entity ent is a group and None in case ent is not a group.

Model.Scale(Entity ent, float sx, float sy, float sz, Position origin)

Return value: no return.

Model.Move(Entity ent, Vec3 translation_vector)

Return value: no return.

Model.Rotate(Entity ent, Position origin, Vec3 axis, float angle)

Return value: no return.

Model.Reflect(Entity ent, Position origin, Vec3 plane_normal)

Mirrors specified entity with respect to the plane given by its origin and its normal vector.

Return value: no return.

Model.Offset(Body ent, float distance)

Offsets the surface of a body by distance.

Return value: no return.
**Transformation** `Model.GetTransformation(Entity ent)`

*Return value*: returns the transformation of the entity decomposed into a **Translate**, a **Rotate** and a **Scale** part.

**Model.SetTransformation(Entity ent, Transformation transf)**

Sets the transformation of an entity decomposed into a **Translate**, a **Rotate** and a **Scale** part.

*Return value*: no return.

**Model.Transform(Entity ent, Transformation transf)**

Applies a transformation to an entity.

*Return value*: no return.

**Body Model.Unite(Body tool, Body blank)**

Unites two bodies. the **blank** is modified and the **tool** is deleted.

*Return value*: returns result of union.

**Body Model.Subtract(Body tool, Body blank)**

Subtracts two bodies. the **blank** is modified and the **tool** is deleted.

*Return value*: returns result of subtraction.

**Body Model.Intersect(Body tool, Body blank)**

Intersects two bodies. the **blank** is modified and the **tool** is deleted.

*Return value*: returns result of intersection.

**Model.Copy()**

Copies current selection in the modeler to the clipboard.

*Return value*: no return.

**Model.Paste()**

Pastes current clipboard contents into the modeler.

*Return value*: no return.

**Model.Delete(Entity ent)**

Deletes an entity from the modeler.

*Return value*: no return.
### C.2. Module SEMCAD

**Entity Model.Clone(Entity ent)**

Makes a clone of an entity.

*Return value:* new cloned entity.

**Model.Delete()**

Deletes current selection in the modeler.

*Return value:* no return.

**Model.Undo()**

Undoes the last modification to the model.

*Return value:* no return.

**Model.Redo()**

Redoes the last modification to the model.

*Return value:* no return.

**Model.NoteState()**

Notes the current model state as an undo state.

*Return value:* no return.

**Model.Select(Entity ent, bool on)**

Marks an entity as selected if on is True. Unselects an entity if on is False.

*Return value:* no return.

**Model.SelectAll(bool on)**

Marks all entities as selected if on is True. Unselects all entities if on is False.

*Return value:* no return.

**Model.SelectRec(Entity entity, bool on)**

Marks an entity and all its children as selected if on is True.

*Return value:* no return.

**Model.Show(Entity ent, bool on)**

Marks an entity as visible if on is True. Hides an entity if on is False.

*Return value:* no return.
\begin{verbatim}
bool Model.IsSelected(Entity ent)

Return value: returns True if entity is selected.

bool Model.IsVisible(Entity ent)

Return value: returns True if entity is visible.

list Model.GetSelection()

Retrieves a list of selected entities in the modeler.
Return value: returns Python list of selected entities.

Group Model.GetRoot()

Return value: returns the root group entity of the model tree.

list Model.GetChildren(Entity ent)

Retrieves a list of child entities.
Return value: returns Python list of child entities.

list Model.CollectAll()

Retrieves a list of all entities.
Return value: returns Python list of all entities.

Model.Heal()

Heals the model.
Return value: no return.

Model.Heal(Entity ent)

Heals a specific entity in the model.
Return value: no return.

Box Model.GetBoundingBox(Entity ent)

Return value: Returns bounding box of Entity ent. The corners of the bounding can be accessed through the members Box.p0 and Box.p1.

Model.SetName(Entity ent, str name)

Sets the name of a model entity.
\end{verbatim}
Return value: no return.

\texttt{str Model.GetName(Entity ent)}

Return value: returns the name of a model entity.

\texttt{str Model.GetFullName(Entity ent)}

Return value: returns the name of a model entity prepended by the names of all ancestor groups.

\texttt{str Model.GetId(Entity ent)}

Return value: returns the unique identifier of a model entity as a string.

\texttt{Entity Model.Find(str Id)}

Return value: returns reference to model entity with a specified unique identifier.

\texttt{Model.ImportModel(str path)}

Imports a SAT model from a file.

\texttt{Group Model.GetCurrentGroup()}  
Retrieves the current group open in the modeler.  
Return value: returns current open group in the modeler.

\texttt{Model.SetCurrentGroup(Group grp)}

Sets the current open group in the modeler.

\texttt{Model.SetColor(Entity ent, float r, float g, float b)}

Sets the color of a model entity. The RGB color components \(r, g, b\) are defined in the range between 0 and 1.  
Return value: no return.

\texttt{Vec3 Model.LightPosition}

Defines the position of the light in the modeler as a 3-D vector.

\texttt{Vec3 Model.CameraEye}

Defines the eye position of the camera in the modeler as a 3-D vector.
*Vec3* **Model.CameraTarget**

Defines the target position of the camera in the modeler as a 3-D vector.

*Vec3* **Model.CameraUp**

Defines the up-direction of the camera in the modeler as a 3-D vector.

*float* **Model.ScalingFactor**

Defines the length of one unit in the model, measured in meters. For example, a value of 0.001 indicates that coordinates in the model are interpreted as millimeters.

*float* **Model.ToMeter(float model_units)**

*Return value:* returns `model_units` converted to meters.

*float* **Model.ToModelUnits(float meter_units)**

*Return value:* returns `meter_units` converted to model units.

*str* **Model.UnitStr**

Defines the units used in the modeler as a string. Read-only attribute.

### C.2.3 class Transformation

Is a representation of a geometrical transformation decomposed into a scaling part, a rotational part and a translational part.

*Vec3* **Transformation.Scale**

*Vec3* **Transformation.Rotate**

*Vec3* **Transformation.Translate**

### C.2.4 class Simulations

Represents the collection of simulations in a SEMCAD document.

*list* **Simulations.List**

Defines a Python list containing references to all simulations in the current SEMCAD project. Example:

```python
# Iterate through all simulations and print their names
for S in simulations.List:
    print S.Name
```
C.2. Module SEMCAD

```c
int Simulations.GetCount()
```

*Return value:* returns the number of simulations in the current SEMCAD project.

```c
Simulation Simulations.GetSimulation(int i)
```

*Return value:* returns a reference object to the i-th simulation in the current SEMCAD project.

```c
list Simulations.GetSelectedSimulations()
```

*Return value:* returns a Python list of references to selected simulations.

```c
Simulation Simulations.GetActiveSimulation()
```

*Return value:* Returns the active simulation.

C.2.5 class Simulation

This class contains all the operations and settings within one simulation in the current SEMCAD project.

```c
Simulation.ComputeBaseLines()
```

Computes only base lines which are used for the grid line generation.

```c
Simulation.ComputeGridFromBaseLines()
```

Computes grid lines based on given base lines.

```c
Simulation.ComputeGridFromScratch()
```

Computes base lines and generates final grid lines which are used for discretization in one step.

```c
Simulation.ComputeVoxels()
```

Computes voxels based on given grid lines.

```c
Simulation.Run()
```

Runs simulation.

```c
Simulation.ResetResults()
```

Deletes results of a simulation.
bool Simulation.HasResults()

_Return value:_ returns True if the simulation has results.

bool Simulation.HasVoxels()

_Return value:_ returns True if the simulation has voxels.

bool Simulation.HasGridLines()

_Return value:_ returns True if the simulation has grid lines.

list Simulation.GetSolidRegions()

Retrieves a Python list of all solid regions objects in a simulation. These solid region objects can be used to set material parameters and other parameters. See `SolidRegion`.

_Return value:_ returns a list of solid region objects.

SolidRegion Simulation.GetSolidRegion(Entity ent)

Retrieves a solid region object in a simulation corresponding to a model entity. The solid region object can be used to set material parameters and other parameters. See `SolidRegion`.

_Return value:_ returns a solid region object.

list Simulation.GetSources()

Retrieves a Python list of all source objects in a simulation. These source objects can be used to set source specific parameters.

_Return value:_ returns a list of source objects.

Source Simulation.GetSource(Entity ent)

Retrieves a source object in a simulation corresponding to a model entity. The source object can be used to set source specific parameters.

_Return value:_ returns a source object.

Source Simulation.GetSource(str id)

Retrieves a source object in a simulation with a given unique entity identifier. The source object can be used to set source specific parameters.

_Return value:_ returns a source object.

list Simulation.GetSensors()

Retrieves a Python list of all sensor objects in a simulation.

_Return value:_ returns a list of sensor objects.
**C.2. Module SEMCAD**

`Grid Simulation.GetGrid()`

*Return value:* returns the Grid object of the simulation. It allows to modify global settings for the grid generation, base lines and grid lines. See Grid class for details.

`str Simulation.Name`

Name of the simulation.

`float Simulation.SolverParameter`

Internal solver parameter.

`float Simulation.Frequency`

Center frequency used for the excitations in the simulation.

`list Simulation.ExtractedFrequencies`

List of extracted frequencies for broadband simulations.

`float Simulation.Bandwidth`

Frequency bandwidth used in the simulation.

`float Simulation.Time`

Total simulation time.

`SimulationTimeUnits Simulation.TimeUnits`

Units used for `Simulation.Time`. Can be *Seconds*, *TimeSteps*, *Periods*.

`Excitation Simulation.ExcitationType`

Specifies the excitation type of all sources in the simulation. Can be *Harmonic*, *Broadband*.

`bool Simulation.UseUserDefinedSignal`

Specifies that a user defined signal should be used as excitation in the broadband mode.

`str Simulation.UserDefinedSignalInputFile`

Specifies the file path containing the user defined excitation signal.
Grid Simulation.Grid

Defines the Grid object of the simulation. It allows to modify global settings for the grid generation, base lines and grid lines. See Grid class for details.

Boundaries Simulation.Boundaries

Defines the boundaries settings for a simulation.

Boundaries.ModeType Simulation.Boundaries.Mode

Specifies the type of absorbing boundaries. Can be Boundaries.UPML or Boundaries.Analytical.

Boundary Simulation.Boundaries.LowX
Boundary Simulation.Boundaries.HighX
Boundary Simulation.Boundaries.LowY
Boundary Simulation.Boundaries.HighX
Boundary Simulation.Boundaries.LowZ
Boundary Simulation.Boundaries.HighZ

These are the boundary settings for every direction. The Boundary settings contain following attributes:

Boundary.BoundaryType Boundary.Type

Defines the type of boundary. Can be Boundary.PEC, Boundary.PMC, Boundary.ABC.

Example:
# Get the first simulation
simulation = simulations.GetSimulation(0)
# Set the lower X boundary to PEC
simulation.Boundaries.LowX.Type = Boundary.PEC

int Boundary.Layers

Number of U-PML layers.

bool Boundary.UseMaxConductivity

Allows user-defined specifications of the conductivity and power loss profiles.

float Boundary.MaxConductivity

Conductivity of the first layer.
C.2. Module SEMCAD

\textit{float Boundary.kMaxOrder}

Growth factor of the power loss.

\textit{float Boundary.MaxConductivityOrder}

Growth factor of the conductivity.

\textit{Boundary.StrengthType Boundary.Strength}

Specifies the minimum level of absorption at the outer boundary:
- \textit{Boundary.LOW} \(\approx 90\%\)
- \textit{Boundary.MEDIUM} \(\approx 95\%\)
- \textit{Boundary.HIGH} \(\approx 99\%\)
- \textit{Boundary.VERYHIGH} \(\approx 99.9\%\)

C.2.6 Constants in SimulationTimeUnits

\textit{SimulationTimeUnits.Seconds}

\textit{SimulationTimeUnits.TimeSteps}

\textit{SimulationTimeUnits.Periods}

C.2.7 Constants in Excitation

\textit{Excitation.Harmonic}

\textit{Excitation.Broadband}

C.2.8 class Proxy

Contains all base settings for an entity within a simulation. An entity can be a solid, a source or a sensor.

\textit{str SolidRegion.Name}

The name of the entity. Is equivalent to the name of the corresponding entity in the modeler.

\textit{str SolidRegion.Id}

The unique identifier of the entity. Is equivalent to the identifier of the corresponding entity in the modeler.

\textit{float SolidRegion.BoundaryRefinementFactor}

Defines the boundary refinement factor for this entity used for the grid line generation.
**bool SolidRegion.RelevantForGrid**

Defines if the entity affects the grid line generation.

**bool SolidRegion.UseLocalGridSettings**

Defines if local grid settings should be applied for the grid line generation.

**float Grid.MaxStepX**

Defines the local maximum grid step constraint on the X-axis used for the grid line generation.

**float Grid.MaxStepY**

Defines the local maximum grid step constraint on the Y-axis used for the grid line generation.

**float Grid.MaxStepZ**

Defines the local maximum grid step constraint on the Z-axis used for the grid line generation.

**float Grid.MinStepX**

Defines the local minimum grid step constraint on the X-axis used for the grid line generation.

**float Grid.MinStepY**

Defines the local maximum grid step constraint on the Y-axis used for the grid line generation.

**float Grid.MinStepZ**

Defines the local minimum grid step constraint on the Z-axis used for the grid line generation.

**float Grid.MaxGradingRatioX**

Defines the local maximum grading ratio constraint on the X-axis used for the grid line generation.

**float Grid.MaxGradingRatioY**

Defines the local maximum grading ratio constraint on the Y-axis used for the grid line generation.

**float Grid.MaxGradingRatioZ**

Defines the local maximum grading ratio constraint on the Z-axis used for the grid line generation.
C.2.9 class SolidRegion

Contains all settings within a physical region defined by a solid body. These settings include material parameters and other settings defined within a solid region. These settings are valid for a specific simulation. It is derived from the Proxy class and therefore inherits all its attributes.

float SolidRegion.Epsilon

Defines the electric permittivity of the solid region.

float SolidRegion.Mue

Defines the permeability of the solid region.

float SolidRegion.SigmaE

Defines the electric conductivity of the solid region.

float SolidRegion.SigmaH

Defines the magnetic conductivity of the solid region.

float SolidRegion.Rho

Defines the density of the solid region.

RegionType SolidRegion.Type

Defines the type of region. can be Dielectric, PEC or PMC.

C.2.10 class Source

Contains all simulation settings of a source object in a specific simulation. It is derived from the Proxy class and therefore inherits all its attributes.
float Source.Theta
float Source.Phi
float Source.Psi
float Source.Amplitude
float Source.RampedPeriods
float Source.DelayedPeriods
float Source.TimeShift
float Source.PhaseShift

Defines if the solid region affects the grid line generation.

C.2.11 class SourceWaveguide

Contains all simulation settings of a waveguide source object in a specific simulation. It is derived from the Source class and therefore inherits all its attributes.

WaveguideType SourceWaveguide.Type

Defines the type of waveguide excitation. Can be SourceWaveguide.Hard or SourceWaveguide.Added.

WaveguideMode SourceWaveguide.Mode

Defines the mode of the waveguide.

int SourceWaveguide.m
int SourceWaveguide.n

float SourceWaveguide.InnerRadius

Defines the inner radius of a coaxial waveguide in meters.

float SourceWaveguide.PolarizationAngle

Defines the polarization angle in degrees for circular waveguides.
float SourceWaveguide.PolarizationDir
Defines the polarization direction for parallel plate waveguides.

C.2.12 class Grid
tuple Grid.GetGridLines(int axis)
Return value: returns a Python tuple with the generated grid lines in the given direction.

Grid.SetGridLines(int axis, tuple coordinates)
Sets the grid lines for a given direction as a Python tuple.

tuple Grid.GetBaseLines(int axis)
Return value: returns a Python tuple with the generated base lines in the given direction.

Grid.SetBaseLines(int axis, tuple coordinates)
Sets the base lines for a given direction as a Python tuple.

bool Grid.UseWaveLengthUnits
Defines if wavelength units should be used the grid settings.

float Grid.MaxStepX
Defines the maximum grid step constraint on the X-axis used for the grid line generation.

float Grid.MaxStepY
Defines the maximum grid step constraint on the Y-axis used for the grid line generation.

float Grid.MaxStepZ
Defines the maximum grid step constraint on the Z-axis used for the grid line generation.

float Grid.MinStepX
Defines the minimum grid step constraint on the X-axis used for the grid line generation.

float Grid.MinStepY
Defines the maximum grid step constraint on the Y-axis used for the grid line generation.
**float** Grid.MinStepZ
Defines the minimum grid step constraint on the Z-axis used for the grid line generation.

**float** Grid.MaxGradingRatioX
Defines the maximum grading ratio constraint on the X-axis used for the grid line generation.

**float** Grid.MaxGradingRatioY
Defines the maximum grading ratio constraint on the Y-axis used for the grid line generation.

**float** Grid.MaxGradingRatioZ
Defines the maximum grading ratio constraint on the Z-axis used for the grid line generation.

**PaddingModeType** Grid.PaddingMode
Defines the type of spacing added around the model. Can be **Off**, **Manual** or **Automatic**.
The padding values are defined through following properties:

**float** Grid.PaddingLowX
**float** Grid.PaddingHighX
**float** Grid.PaddingLowY
**float** Grid.PaddingHighY
**float** Grid.PaddingLowZ
**float** Grid.PaddingHighZ

### C.2.13 class Preferences

**float** Preferences.Transparency
Defines the transparency amount (0-1) used to render un-selected solids when transparency is enabled.

**float** Preferences.SelectedTransparency
Defines the transparency amount (0-1) used to render pre-selected solids when transparency is enabled.

### C.3 Module PostPro
Contains the interface for the access of the Post Processor functionality like the extraction of results.
bool InitSimulation(int i)

Initializes the postprocessing system using the i-th simulation in the project.

Return value: returns True on success. This function has to be called before performing any result extraction.

bool InitSelectedSimulation()

Initializes the postprocessing system using the currently selected simulation in the project.

Return value: returns True on success. This function has to be called before performing any result extraction.

DataField GetResult(str sensor_name, str quantity, bool silently)

Extracts a result DataField with quantity quantity from the sensor sensor_name using silent mode if silently is True. The Post Processor has to be initialized using InitSimulation(...) before. The result Data-Field is stored and managed inside the Post Processor cache after extraction. Don’t call del on the returned DataField object. Use DestroyDataField(...) to delete the DataField object from the cache.

Return value: returns the a reference to the result DataField in case of success. Returns object of type NoneType if extraction is not possible.

bool DestroyDataField(DataField data_field)

Removes and deletes the DataField object data_field from the Post Processor cache.

Return value: returns True in case of success.

DestroyAllDataFields()

Removes and deletes all DataField objects which are not referenced from the Post Processor cache.

Return value: no return.

Reset()

Removes and deletes all Data-Field objects which are not referenced from the Post Processor cache and closes all viewers.

Return value: no return.

Viewer CreateViewer(str viewer_type, DataField data_field)

Opens a viewer identified by the string viewer_type (see context menu in SEMCAD result tree) with the DataField object data_field as data source.

Possible default viewer types are:


Return value: Viewer object in case of success, NoneType in case of failure.
DataField CreateDataField(str name='''')

Creates a new DataField object in the cache.
Return value: returns the newly created DataField object.

DataField CloneDataField(DataField data_field, str name='''')

Creates an exact clone of the Data-Field object data_field in the cache with a new name.
Return value: returns the newly created Data-Field object.

list GetSensors()

Retrieves the list of sensor names from an initialized simulation. The Post Processor has to be initialized using InitSimulation(...) before.
Return value: returns list of strings with the names of all sensors in the simulation.

list GetQuantities(string sensor_name)

Retrieves the list of quantities from a given sensor with the name sensor_name. The Post Processor has to be initialized using InitSimulation(...) before.
Return value: returns list of strings with the names of all quantities in a given sensor.

string GetSimulationName()

Return value: returns the name of the simulation currently initialized in the Post Processor.

C.3.1 class DataField

Interface to a DataField object which represents a generic multi-dimensional array. It’s an array container for evaluated results.

str DataField.GetAttribute(str attribute_name)

Return value: returns an attribute named by attribute_name as a string.

dict DataField.GetAllAttributes()

Return value: returns a Python dictionary containing all attributes of the DataField.

DataField.SetAttribute(str attribute_name, str value)

Writes an attribute string value named by attribute_name into the attribute table of the DataField object.
Return value: no return.
list DataField.GetTable(str table_name='')

Extracts a table stored in the DataField.

*Return value:* Returns the table `table_name` stored in the DataField object as a list of lists with the following structure:

```
[[ "Label of column 1", "row 1", "row 2", ...], [ "Label of column 2", "row 1", "row 2", ...], ...]
```

DataField.InitReal(int n_components=1, int n_x=1, int n_y=1, int n_z=1, int n_time=1)

Initializes a real DataField object with given dimensions and allocates the necessary memory for it. `n_components` defines the number of components. `(n_x, n_y, n_z, n_time)` defines the number of array elements in every direction.

*Return value:* returns `True` in case of success and `False` in case of failure.

DataField.InitComplex(int n_components=1, int n_x=1, int n_y=1, int n_z=1, int n_time=1)

Initializes complex DataField object with given dimensions and allocates the necessary memory for it. `n_components` defines the number of components. `(n_x, n_y, n_z, n_time)` defines the number of array elements in every direction.

*Return value:* returns `True` in case of success and `False` in case of failure.

int DataField.GetSize(Axis axis)


int DataField.GetComponents()

*Return value:* returns the number of components.

DataField.Set(float value, int comp, int x_index=0, int y_index=0, int z_index=0, int time_index=0)

Sets value `value` in the DataField array for field component `comp` at the position given by the indices `(x_index, y_index, z_index, time_index).

*Return value:* no return.

DataField.SetComplex(complex value, int comp, int x_index=0, int y_index=0, int z_index=0, int time_index=0)

Sets value `value` in the DataField array of the complex field component `comp` at the position given by the indices `(x_index, y_index, z_index, time_index).

*Return value:* no return.

float DataField.Get(int comp, int x_index=0, int y_index=0, int z_index=0, int time_index=0)

*Return value:* Returns value in the DataField array for field component `comp` at position given by the indices `(x_index, y_index, z_index, time_index).
complex DataField.GetComplex(int comp, int x_index=0, int y_index=0, int z_index=0, int time_index=0)

*Return value:* Returns the complex field component `comp` at position given by the indices `(x_index, y_index, z_index, time_index)`.

DataField.SetCoor(float value, Axis axis=Axis.X, int x_index=0, int y_index=0, int z_index=0, int time_index=0)

Sets coordinate `value` of field on axis `axis` (can be `Axis.X`, `Axis.Y`, `Axis.Z`, `Axis.Time`) at position given by the indices `(x_index, y_index, z_index, time_index)`.

*Return value:* no return.

float DataField.GetCoor(Axis axis=Axis.X, int x_index=0, int y_index=0, int z_index=0, int time_index=0)

*Return value:* returns coordinate value of field on axis `axis` (can be `Axis.X`, `Axis.Y`, `Axis.Z`, `Axis.Time`) at position `(x_index, y_index, z_index, time_index)`.

DataField.ExtractAxis(Axis axis=Axis.X, int u=0, int v=0, int w=0)

*Return value:* Returns axis coordinates of axis `axis` (can be `Axis.X`, `Axis.Y`, `Axis.Z`, `Axis.Time`) as a Python tuple at the complementary location indices `(u, v, w)`. For `axis=0` the index triple `(u, v, w)` represents the axis at index `u` on the y-axis, at index `v` on the z-axis and at index `w` on the time axis. For axis aligned grids it is not necessary to use arguments `(u, v, w)`.

DataField.SetAxis(tuple values, Axis axis=Axis.X, int u=0, int v=0, int w=0)

*Return value:* Sets axis coordinates of axis `axis` (can be `Axis.X`, `Axis.Y`, `Axis.Z`, `Axis.Time`) as a Python tuple at the complementary location indices `(u, v, w)`. For `axis=Axis.X` the index triple `(u, v, w)` represents the axis at index `u` on the y-axis, at index `v` on the z-axis and at index `w` on the time axis. For axis aligned grids it is not necessary to use arguments `(u, v, w)`.

float DataField.ComputeMax()

*Return value:* Returns the maximum real value of all array elements.

float DataField.ComputeMin()

*Return value:* Returns the minimum real value of all array elements.

DataField.SetAxisUnits(Axis axis, str units)

Sets units `units` on the array axis `(0,1,2,3)`.

*Return value:* no return.
C.3. Module PostPro

`DataField.SetAxisLabel(Axis axis, str label)`

Sets label `label` for the array axis `axis` (0,1,2,3).

*Return value:* no return.

`DataField.SetCompUnits(int comp, str units)`

Sets units string `units` for the field component `comp`.

*Return value:* no return.

`str DataField.GetCompUnits(int comp)`

*Return value:* returns units for the field component `comp` as a string.

`DataField.SetCompLabel(int comp, str label)`

Sets label string `label` for the field component `comp` as a string.

*Return value:* no return.

`DataField.SetCompLabelRe(int comp, str label)`

Sets label string `label` for the real part of the field component `comp` as a string.

*Return value:* no return.

`DataField.SetCompLabelIm(int comp, str label)`

Sets label string `label` for the imaginary part of field component `comp` as a string.

*Return value:* no return.

`str DataField.GetCompLabel(int comp)`

*Return value:* returns label string for the field component `comp` as a string.

`str DataField.GetCompLabelRe(int comp)`

*Return value:* returns label string for the real part of the field component `comp` as a string.

`str DataField.GetCompLabelIm(int comp)`

*Return value:* returns label string for the field component `comp` as a string.

`DataField.+(DataField data_field)`

Operator which adds the `DataField` object on the right to the `DataField` object on the left element by element. The operator requires two `DataField` objects of the same type.
DataField\(\text{--}(\text{DataField}\ \text{data}\_\text{field})\)

Operator which subtracts the DataField object on the right from the DataField object on the left element by element. The operator requires two DataField objects of the same type.

Return value: no return.

DataField\(\text{\ast}(\text{float}\ \text{scalar})\)

Operator which scales the DataField object on the left by the float value scalar element by element.

Return value: no return.

DataField\(\text{\ast}(\text{complex}\ \text{scalar})\)

Operator which scales the DataField object on the left by the complex value scalar element by element. The DataField object itself has to be complex.

Return value: no return.

\begin{verbatim}
int DataField.GetByteSize()

Return value: no returns the total size of the DataField as bytes.
\end{verbatim}

\section*{C.3.2 class EvalOptions}

All the attributes in EvalOptions are accessible globally.

\begin{verbatim}
bool EvalOptions.NormalizeField

If True fields are normalized to a given power.
\end{verbatim}

\begin{verbatim}
float EvalOptions.NormalizeTargetPower

Normalizes fields to the given target input power if NormalizeField is True.
\end{verbatim}

\begin{verbatim}
complex EvalOptions.RefImpedance

The reference impedance used for impedance related evaluations.
\end{verbatim}

\begin{verbatim}
float EvalOptions.AveragedSARMass

The mass used to compute the averaged SAR distribution.
\end{verbatim}
EvalOptions.Set(str option, value)

More general evaluation options are accessible through Set and Get functions. Every option is identified by a string option.
Return value: no return.

str EvalOptions.Get(str option)

More general evaluation options are accessible through Set and Get functions.
Return value: returns the value of the specified option as a string.

Standard Evaluation Options

list EvalOptions.SolidMask

List of strings with names of solids used to extract certain results based on field distribution. Important: Set this option only after simulation has been initialized.

C.4 Module Math

C.4.1 class Vec3 and class Position

Vec3 and Position synonyms representing the same type.
It’s a simple 3-D vector class.

Vec3(float x, float y, float z)

Constructs a 3-D vector with components x,y,z.

Vec3(float x)

Constructs 3-D vector with components x,x,x.

float Vec3.Dot(Vec3 v)

Return value: computes dot-product with another vector v.

float Vec3.Length()

Return value: returns the length of the vector.

Vec3.GramSchmidt(Vec3 a, Vec3 b)

Constructs two vectors a and b which are orthogonal to this vector.
**Vec3.Normalize()**

Normalizes vector to a length of 1.

**[int i]**

The [ ]-operator can be used to get and to set the i-th component of the 3-D vector.

*Return value: i-th component of the 3-D vector as a float.*

**Vec3 + Vec3**

Two 3-D vectors can be added using the operator +.

*Return value: sum of the two vectors.*

**Vec3 - Vec3**

Two 3-D vectors can be subtracted using the operator -.

*Return value: difference of the two vectors.*

**Vec3 ^ Vec3**

The cross product of two 3-D vectors can be computed using the operator ^.

*Return value: cross product of the two vectors.*

**float * Vec3**

A 3-D vector can be scaled by a scalar using the operator *.

*Return value: the scaled vector.*

**Vec3.+= Vec3**

The 3-D vector on the right side is added to the vector on the left side.

*Return value: no return.*

**Vec3.-= Vec3**

The 3-D vector on the right side is subtracted from the vector on the left side.

*Return value: no return.*

**Vec3.*= float**

The 3-D vector on the left side is scaled by the scalar on the right side.

*Return value: no return.*