



CST MICROWAVE STUDIO®

HF DESIGN AND ANALYSIS



CST MWS VERSION

5

ADVANCED TOPICS

Copyright

© 1998-2003

CST – Computer Simulation Technology

All rights reserved.

Information in this document is subject to change without notice. The software described in this document is furnished under a license agreement or non-disclosure agreement. The software may be used only in accordance with the terms of those agreements.



No part of this documentation may be reproduced, stored in a retrieval system, or transmitted in any form or any means electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of CST.

Trademarks

Microsoft, Windows, Visual Basic for Applications are trademarks or registered trademarks of Microsoft Corporation. Sax Basic is trademark of Sax Software Corporation.

Other brands and their products are trademarks or registered trademarks of their respective holders and should be noted as such.

CST – Computer Simulation Technology

www.cst.com

December 11, 2003

Contents

CHAPTER 1 - INTRODUCTION	5
CHAPTER 2 - THE INNOVATIONS IN CST MICROWAVE STUDIO® VERSION 5	6
2.1 Structure Modeling.....	6
2.2 Meshing.....	7
2.3 Ports.....	8
2.4 General Performance improvements.....	9
2.5 Transient Solver.....	10
2.6 Eigenmode Solver.....	10
2.7 Frequency Domain Solver.....	11
2.8 Post Processing.....	12
2.9 Farfields.....	13
CHAPTER 3 - CAD DATA IMPORT AND HEALING	14
3.1 Import 2D Files.....	15
3.2 Import 3D Files.....	16
3.3 Inspect and Repair the Imported Model.....	18
3.4 Parameterize the Imported Model.....	25
CHAPTER 4 – THE SIMULATION METHOD	28
4.1 Background of the Simulation Method.....	28
The Finite Integration Technique (FIT).....	28
The CST MICROWAVE STUDIO® Solvers.....	31
References.....	32
4.2 Error Sources / Sources of Inaccuracy.....	33
Simulation Model is not in Agreement with Reality.....	33
Inaccuracies due to the Simulation.....	33
4.3 Mesh Generation.....	35
Methods of Mesh Generation.....	35
Global Mesh Parameters.....	36
Mesh Adaption.....	41
4.4 Mesh Tuning.....	45
Ratio Limit, Number of Mesh Points and Simulation Time.....	45
Local Mesh Parameters.....	45
4.5 The Multilevel Subgridding Scheme (MSS™).....	50
4.6 Performance Improvements.....	58
General Hints.....	58
Special Time Domain Topics.....	58
4.7 Auto-Regressive Filtering (AR-Filter).....	63
CHAPTER 5 - THE VBA MACRO LANGUAGE	70
5.1 Introduction.....	70
5.2 VBA Development Environment.....	71
VBA Help System.....	71

VBA Editor Shortcuts	72
5.3 VBA Language Elements	73
Subroutines and Functions	73
Variables, Data Types and Type Conversions	74
Applications, Objects and their Methods	75
Flow Control	77
File Operations	77
Graphical User Interface Builder	78
Mathematical Functions, Operators and Constants	78
5.4 Concepts of Macro Programming in CST MWS	79
Local Project Macros	80
Global Macros and <i>Global Macro Path</i>	80
5.5 Control Macros	81
Result1D Object	82
Result3D Object	85
Adding Data Items to the Navigation Tree	85
Traverse Folders and Select Items From the Navigation Tree	86
Access the Currently Displayed Plot Data	87
Access Farfield Data From a VBA Script	87
5.6 Structure Macros	89
Creating a Structure Macro or a New Project Template	89
Difficulties in Creating Structure Macros	91
Using Dialog Boxes in Structure Macros	92
Common Pitfalls in the Usage of Structure Macros	93
CHAPTER 6 - OVERVIEW OF INSTALLED VBA MACROS	94
6.1 File and Result Handling	94
File / Make Archive – Reporting a problem to CST	94
Wizard / Delete old results / Search and Delete	96
6.2 Structure Macros	97
Link to Agilent's Advanced Design System (ADS)	97
Macros for Construction of Coils	98
Construct / Create 2D-Curve analytical (xy/uv)	99
Construct / Create 3D-Curve analytical (xyz/uvw)	99
Construct / Multiple Discrete Ports	100
6.3 Creating Pictures, Videos and Reports	101
Graphics / Save Image	101
Graphics / Save Video	102
Report / PowerPoint Presentation	104
Report / PowerPoint Slide	105
MS-Excel 2000 and MS-Word 2000	105
6.4 Post Processing Individual Simulations	106
Farfield / Broadband Farfield Monitors	106
EMC / Calculate Broadband EMC-Norm	107
EMC / Renorm Probes+Voltages to EMC-frequency source	107
Results / ASCII Export + Energy / 2D-3D Export + Energy_Loss Calculation	108
Time Signal Processing / TDR Computation	110
Time Signal Processing / Exchange Excitation	111
Time Signal Processing / Eye Diagram	112
Touchstone + SPICE / Calculate SPICE S-Parameters	113
Touchstone + SPICE / Touchstone Reduction	113
Further examples for 0D- and 1D-Results	114
6.5 Postprocessing multiple simulations and running Parameter Sweeps	115
Wizard / Compare multiple runs	115
Wizard / Parameter + Mesh Study	120

Parameter Study via Parameter Table	121
6.6 Batch Mode and Job Control Systems	122
Running CST MICROWAVE STUDIO® in Batch Mode	122
CST Job Control Center	122
Distributing simulation jobs across a network of PCs	123
CHAPTER 7 – TEMPLATE BASED POSTPROCESSING	124
7.1 Framework to control Result Templates	125
Simple example to get started with Postprocessing templates	126
7.2 Overview of installed Result Templates	131
7.3 Farfield Result Templates	132
0D / Farfield	132
1D / Farfield (single frq)	133
1D / Farfield (broadband)	133
7.4 Result Templates for 2D/3D Results	134
Evaluate all fields along all curves	134
Mix 3D Fields	134
1D / Evaluate Field on predefined Curve	135
0D / Evaluate Field on predefined Curve	135
0D / Evaluate Field on predefined Face	135
1D / Evaluate Field along arbitrary Coordinates (1D Plot)	136
0D / Evaluate Field in arbitrary Coordinates (Field Values, 1D, 2D, 3D)	136
7.5 Result Templates for arbitrary 1D Results	137
Calculate all Z and Y Matrices	137
1D / Load 1D Data File (project and external)	137
7.6 Calculation Tools for Result Templates	138
1D / 1D Result from 1D Result (Rescale xy, FFT, Derivation, ...)	138
0D / 0D Value from 1D Result	139
1D / Mix 1D Results	139
0D / Mix 0D Results	139
CHAPTER 8 - NETWORK INSTALLATION	140
8.1 License Server Installation	140
8.2 Software Installation	142
8.3 Installation by Using an Application Server	143
8.4 Network Distributed Computing Installation	144

Chapter 1 - Introduction

This manual is primarily designed for advanced users of CST MICROWAVE STUDIO® (CST MWS) who are already familiar with the basic concepts of using the software.

Before you actually start reading this book, we strongly recommend that you carefully work through the *Getting Started* manual and the *Tutorials*. Since a lot of changes have been introduced in version 5 of CST MWS, it may be helpful even for experienced users of earlier versions to have a look at these documents again.

This collection of *Advanced Topics* offers some additional information on subjects which are usually of a more involved nature. The following list gives a short summary of this manual's contents:

- Chapter 2 provides a brief overview of the most important newly introduced features of CST MWS 5.
- Chapter 3 focuses on the basic principles of importing CAD data in two and three dimensions. Some hints are given on how to improve the robustness and reliability of exchanging CAD data files.
- Chapter 4 is dedicated to the FI method which is the underlying mathematical simulation technique. After a general introduction into the method's fundamentals, the PBA® technique (Perfect Boundary Approximation) is also introduced. Further sections in this chapter then explain possible sources of errors or inaccuracies and discuss meshing strategies. Finally some hints on how to tune the performance of a simulation are given.
- Chapter 5 provides an introduction to the key concepts of the VBA-based macro programming language used in CST MWS. You will obtain some information on the most important VBA elements by means of a couple of examples. A basic understanding of these language elements will be very helpful for you in order to write your own macros and user defined functions.
- Chapter 6 gives an overview of the preinstalled macros, which contain many time saving solutions to automate common tasks. For each of these macros an introduction is provided which should enable you to make use of that particular macro. You may even get some ideas of how to use automation for your daily work.
- Chapter 7 explains the available result templates and how they can be utilized to automate the post processing.
- Chapter 8 describes several options to install CST MWS on a local area network and how to setup the distributed network computing feature.

Please note that this book is not intended to be a complete reference guide. For successfully mastering these more advanced topics we strongly recommend that you visit a specialized training class. Please contact your local support center for more information.

Chapter 2 - The Innovations in CST MICROWAVE STUDIO® Version 5

This chapter provides an overview of some of the most important innovations in the new version 5 of CST MICROWAVE STUDIO®.

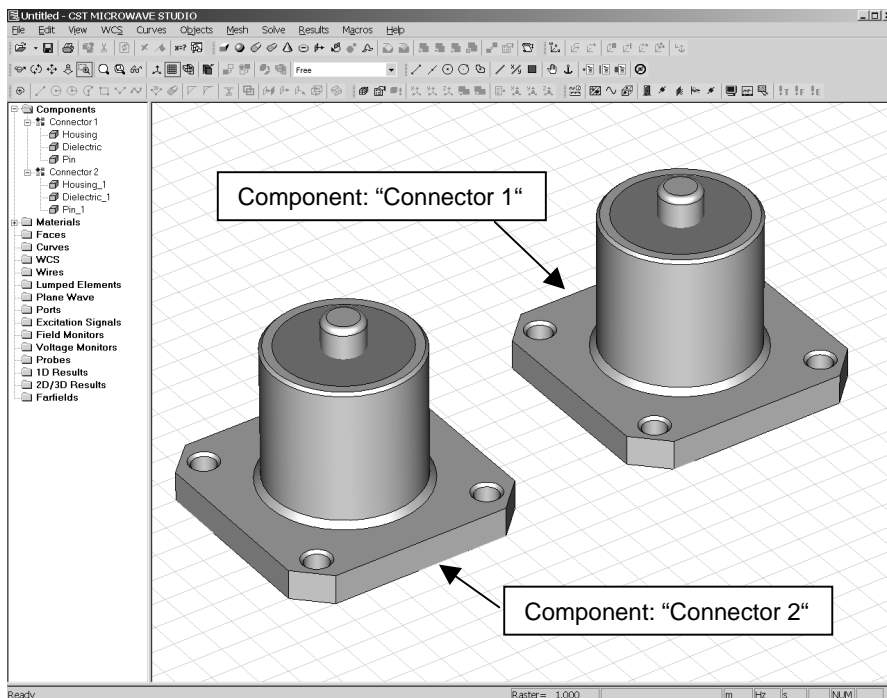
More than 1800 improvements and new features have been implemented since the release of version 4. Even though many of them are rather minor changes, they are nevertheless quite important since the improvements in usability make your daily work much easier. We are confident that you will notice a lot of changes just by opening CST MICROWAVE STUDIO® 5 and working with it.

A selection of new key features are described on the following pages in the order of a typical modeling process.

2.1 Structure Modeling

A variety of major changes have been introduced into the modeler since the organization of the structure has been completely revised. A so-called *component management* has been added to allow for hierarchical grouping of shapes. In addition to this, each shape belongs to a certain material.

The advantage of this approach is that parts of the structure (e.g. connectors, etc.) can now be stored as a single component, even if they consist of different materials.



The material properties themselves can now be optionally retrieved from a global data base which is also customizable according to your needs.

Furthermore, a multi-selection of shapes has been added. This feature allows for convenient operations (e.g. transformations, changes in the material and component associations, etc.) on more than one shape at a time.

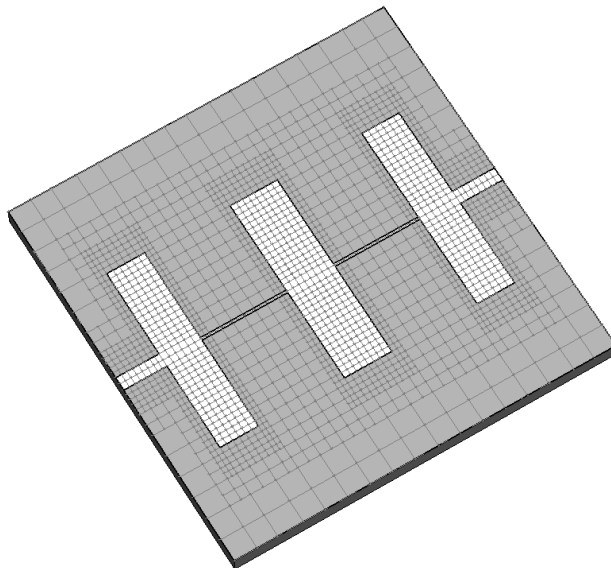
The ACIS¹ kernel has been upgraded to version 12 which contains significant improvements in the overall robustness and the performance of the modeling operations. An important enhancement of the Pro/E import capability is the import of encrypted files which is now supported, too.

The voxel data import tool has been enhanced in order to read generalized voxel data rather than the human model data set only.

In addition to the features mentioned here, a multitude of other changes and enhancements has been introduced into the modeling part of the software. Please refer to the online documentation for more details.

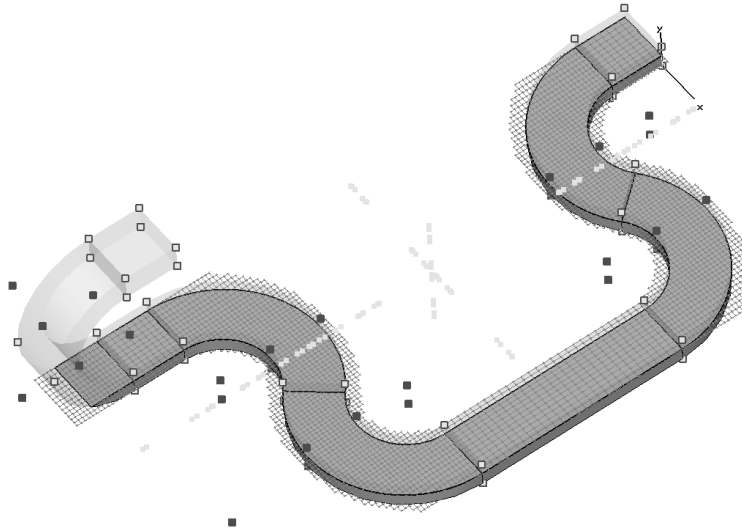
2.2 Meshing

The Multilevel Subgridding Scheme (MSS™) is a new extension to the PBA® and TST™ algorithms which has firstly been implemented in the transient solver. The MSS™ technique can yield a significant speed-up for the simulation of complex devices since mesh lines no longer need to extend through the entire computation domain. Therefore, a refined mesh can be concentrated more efficiently within the regions where it is actually needed.



¹ Parts of this software are owned by Spatial Corp.© 1986 – 2003. All Rights Reserved.

Another advantage of the MSS™ technique is the automatic exclusion of perfect conducting regions from the computation domain. This can additionally save time for structures which contain large perfect conducting volumes as shown in the following picture.



In addition to the introduction of the MSS™ technique, the TST™ method has also been enhanced in order to allow for thin sheets in waveguide ports and to handle lossy metal and dispersive material types.

In addition to these fundamental algorithmic enhancements, a variety of other changes has also been introduced. For example, the local mesh density control for individual shapes has been improved by adding local edge and volume refinement factors. Please refer to the online documentation for details.

2.3 Ports

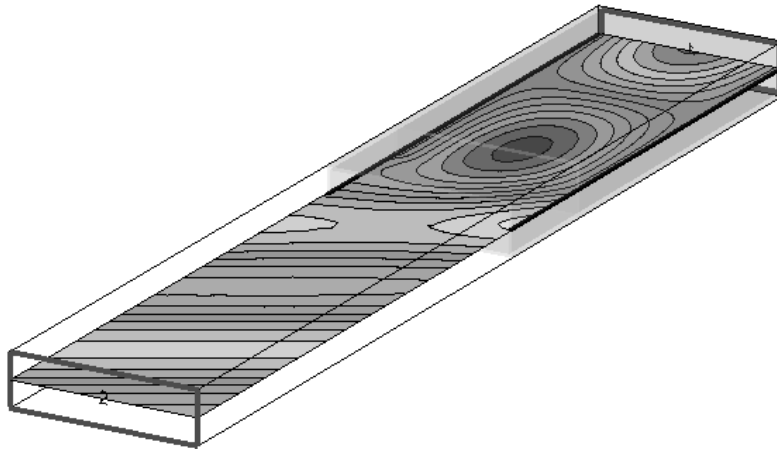
A major improvement has been implemented for the handling of ports. Waveguide ports can now be located inside the structure which is quite useful for e.g. antenna simulations. Furthermore, waveguide ports can now also be used together with plane wave excitations which is especially interesting for EMC problems.

Another enhancement has been made for the multipin waveguide ports where the same pins can now be re-used for the definition of the modes. This allows a very convenient handling of even and odd modes in the same simulation and therefore makes the calculation of the cross-talk very straightforward.

The frequency domain solver has been extended to handle lossy waveguide ports with simplification required. Consequently, the port mode solver has been improved to calculate modes in lossy structures as well.

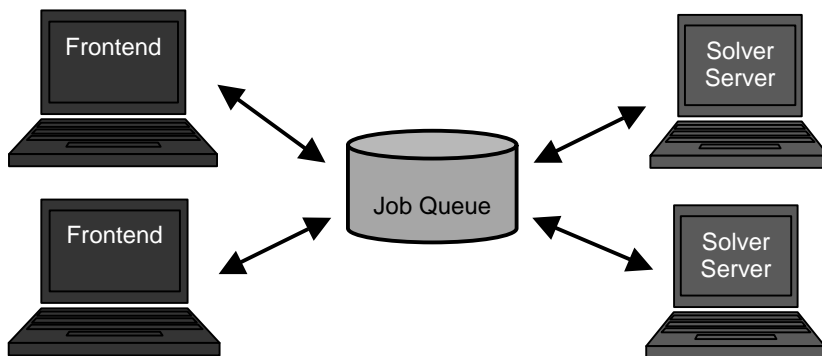
Another major achievement for the frequency domain solver is the handling of periodic ports including phase shifts. These so-called *Floquet* modes allow an efficient and highly

accurate simulation of radiation effects in periodic structures (e.g. PBG, FSS, periodic antenna arrays, etc.) without degradation of accuracy at slanted angles (as it usually occurs by using PML boundaries).



2.4 General Performance improvements

In order to further improve the performance of all solvers, a network distributed computing system has been developed.



This system is able to handle distributed port / mode excitations, yielding a tremendous speed-up especially for structures with many ports (e.g. packages), simulated with the transient solver. Furthermore, the system is also able to distribute parameter sweeps and even optimizations of all other solver types.

It is now possible to allocate up to 3 GByte of RAM to a single CST MWS process in Windows 2000 – Advanced Server or Windows XP Professional. During the boot process an option "/3GByte" can be set to activate this additional feature. Please contact your local support team to obtain further details.

2.5 Transient Solver

Besides this major enhancement concerning general performance, the transient solver can now handle simultaneous excitations at various ports with different excitation time signals (including an excitation signal library).

For efficient calculations of farfield spectra at isolated locations, time domain farfield probes have been added which handle this particular task much more efficiently than the frequency domain farfield monitors.

As another extension, the plane wave excitation can now also handle circularly or elliptically polarized plane waves.

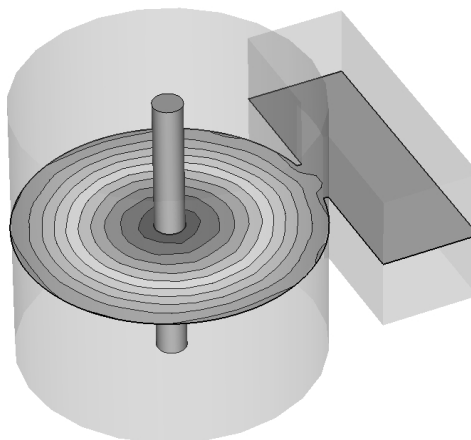
Last but not least, a new SPICE network extraction tool has been implemented based on a so-called *model order reduction* (MOR) technique. This new extraction model does no longer assume any fixed topology of the equivalent circuit and therefore has a much larger application range than the transmission line based network parameter extraction.

In addition to the new features mentioned above, a lot of other enhancements have been made to the transient solver as well. Please refer to the online documentation for further information.

2.6 Eigenmode Solver

In addition to the well established AKS solver, a new eigenmode solver has been added which no longer needs a guess for the highest eigenfrequency. This leads to a much more robust solution especially for very challenging eigenmode problems (e.g. large dielectric constants).

The new eigenmode solver is also able to directly calculate the modes in lossy structures. The following picture shows a mode calculated for a cavity with a lossy dielectric cylinder:



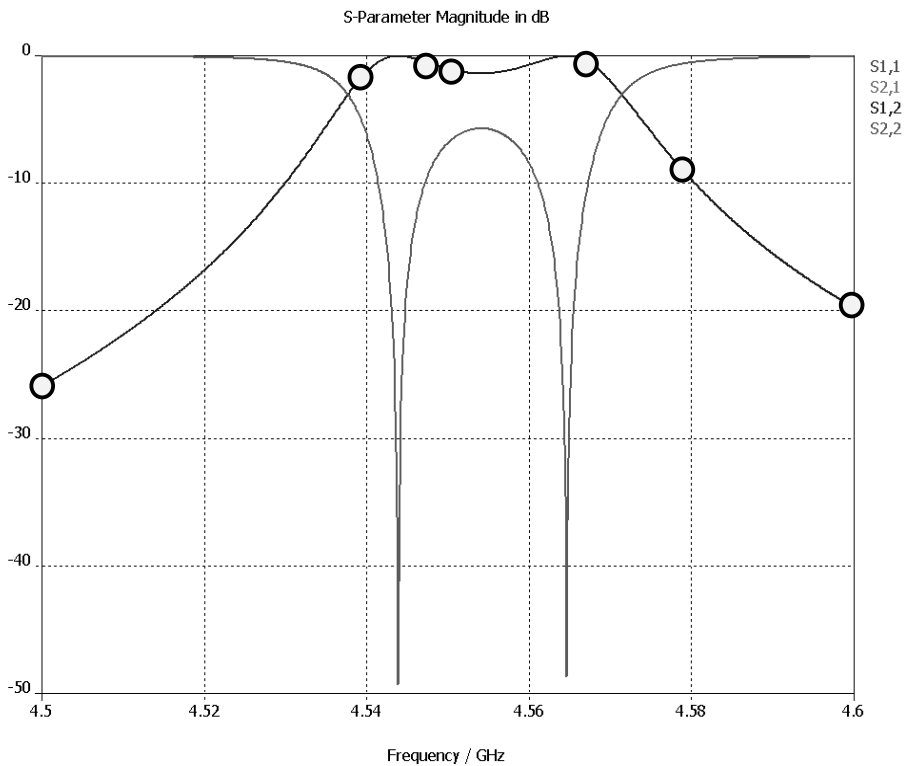
In the same way as described for the transient solver above, the eigenmode solver does also support network distributed parameter sweeps and optimizations.

2.7 Frequency Domain Solver

The frequency domain solver has been upgraded significantly compared to previous releases.

In particular, the solver speed has been improved by adding a so-called *direct matrix solver* (useful for relatively small structures or many ports) and by improving the convergence speed of the iterative solver.

An adaptive fast broadband frequency sweep has been added which calculates the S-parameters from a few calculated samples only. This technique can yield tremendous gains in performance compared to the traditional frequency stepping approach. The following picture shows an example where the broadband S-parameter curves are obtained by calculating only seven automatically chosen frequency points.



Many features of the time domain solver have now also been added to the frequency domain such as: frequency dependent material, extended field monitors, multipin port, internal waveguide port, parameter sweep, optimization and adaptive mesh refinement.

Besides these features which are now available in time and frequency domain, some special frequency domain solver features have also been implemented such as lossy waveguide ports and periodic boundaries with phase shift. As well as the specification of a phase shift between the boundary planes, a scan angle can also be optionally specified which allows for an easier handling of frequency sweeps for fixed angle of incidence.

The distributed network computing feature has also been added to the frequency domain solver for parameter sweeps and optimizations.

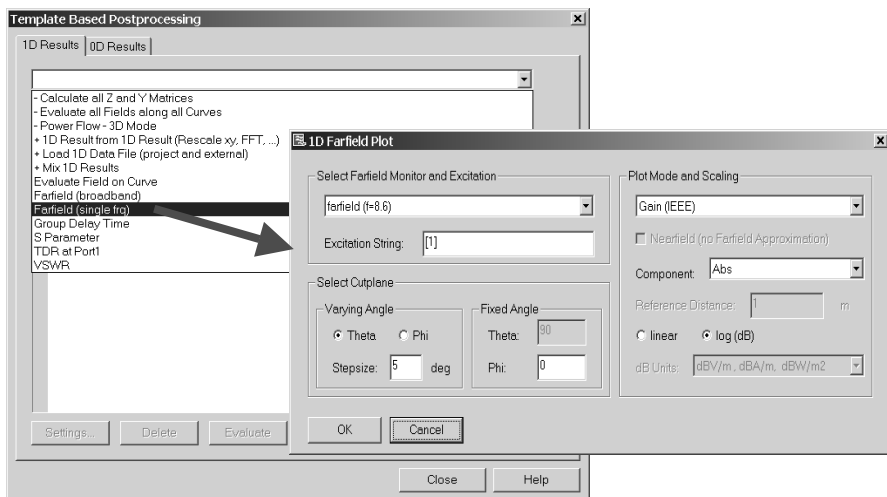
2.8 Post Processing

The xy-curve handling in the post processing has been enhanced by adding capabilities to scale curves, import external xy-data and copy / paste result curves in the navigation tree. Furthermore, the curves can also be pasted directly into an Excel sheet.

Besides these usability enhancements, the integration of fields across arbitrary faces has been added, e.g. to calculate power flows, shielding factors, etc.

The auto regressive (AR) filtering has been extended to deal with arbitrary probe signals and voltage monitors as well.

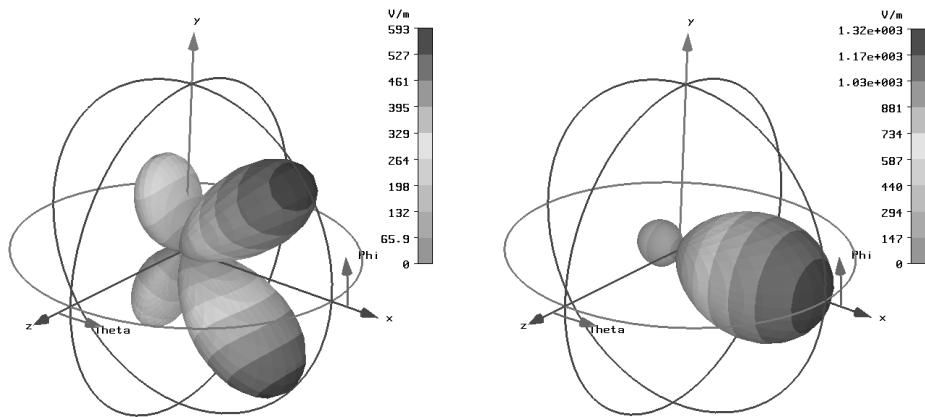
The so-called *Result Templates* now enable extremely flexible automated and customized post-processing. A chain of result processing steps can be defined, starting from the extraction of result data to the calculation with curve and point data up to the definition of extremely flexible optimization goals. The picture below shows the definition of farfield data extraction as an example:



2.9 Farfields

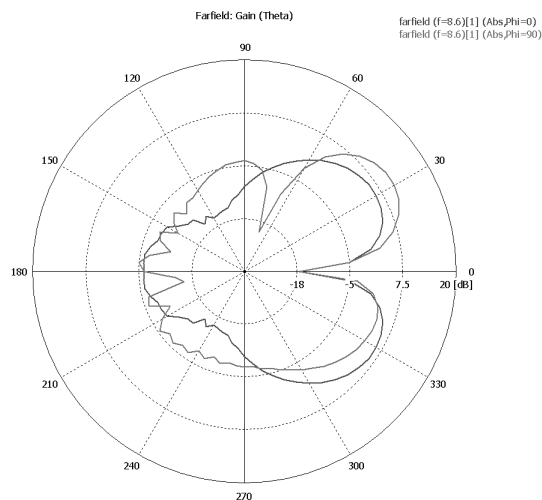
In addition to the general enhancements in the post processing, the farfield capabilities have been upgraded significantly.

Besides an improved visualization of the farfield's phase and automatic phase center calculations, the option to display farfields in the so-called *Ludwig* coordinate system has been added. The picture below shows the horizontal and vertical farfields in Ludwig3 coordinates, for instance:



Furthermore, a *nearfield* option has been added exploiting the full Green's functions to calculate the field components at a finite distance as well (including radial components).

Another useful feature is the Copy / Paste function for farfield plots which allows to quickly overlay plot data as shown in the picture below:



Chapter 3 - CAD Data Import and Healing

The ability to import geometrical CAD data from other systems is playing an increasingly important role in the operation of modern CAE tools. Reducing the need for re-modeling structures (which have already been drawn in a CAD system) for electromagnetic analysis can significantly reduce the cost for computer simulations.

CST MICROWAVE STUDIO® offers powerful import options for geometrical data in both two and three dimensions.

This chapter will provide an introduction on how the import options should be used without focusing on the actual dialog boxes. Please refer to the online documentation for more information.

The structural import process usually consists of two steps:

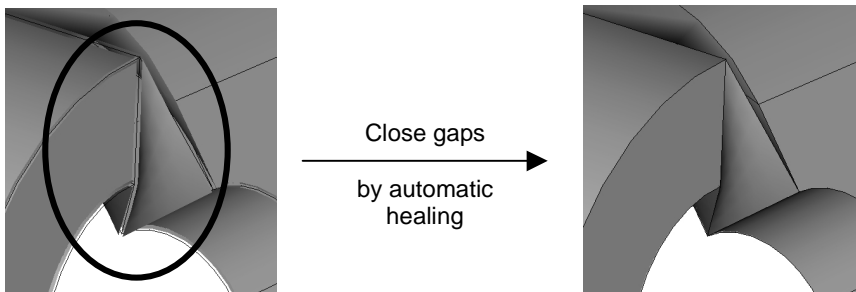
1. **Import** of the geometrical data and its conversion into ACIS models. Please note that the origin of the imported CAD model will automatically be adjusted to the origin of the currently active coordinate system.
2. **Healing** of the newly created ACIS models in order to obtain valid geometry.

The first step can be seen as a pure geometric language translation step. All inaccuracies in the imported models will remain during this stage. The second step then actually repairs the data such that valid ACIS models are finally obtained.

In general, problems during the import and healing process arise from different accuracy levels of the CAD system and the ACIS kernel used in CST MICROWAVE STUDIO®. The accuracy of a CAD system is usually the distance for which the system considers two points to be identical. The ACIS kernel uses an accuracy of $1e-6$ whereas some other CAD systems only operate at accuracies of $1e-3$.

Assuming that the CAD data has been created on a system with an accuracy level of $1e-3$, it may easily contain gaps of this size (which the CAD system would still consider to be closed). Importing such a model into the ACIS kernel afterwards would keep these gaps. However, in contrast to the less accurate CAD system, ACIS would not consider these gaps to be closed which may finally cause problems in creating solids. To circumvent these problems, sophisticated automatic healing technology can be used to close the gaps in the imported model.

The following screenshot illustrates this step in the case of a three dimensional part:



The healing modifies the imported model such that it finally forms a valid solid shape. The following sections will discuss the import process in 2D and 3D in more detail.

3.1 Import 2D Files

CST MICROWAVE STUDIO® currently supports the following 2D data file formats:

- DXF
- GDSII
- Gerber

We recommend using the DXF file format whenever possible because this is the most robust choice for 2D data transfer.

Two different levels of DXF import are available:

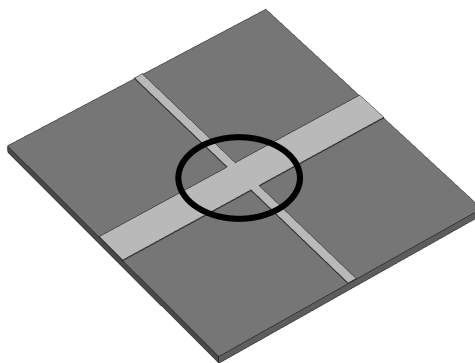
1. Standard DXF
2. Extended DXF

The main advantage of the optionally licensed *Extended DXF* import feature is the support of hierarchical data (so-called blocks) in the DXF file and the support of thick lines.

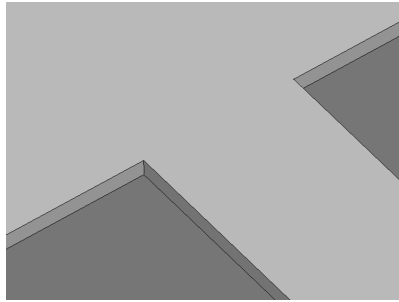
By default (if the option *Import as curves only* is NOT used), CST MICROWAVE STUDIO® tries to connect the imported 2D elements (lines, arcs, polygons, etc.) such that they form closed profiles. Additionally a healing option is available which automatically tries to close small gaps in the resulting profiles.

Once a closed profile is obtained, it will automatically be covered and extruded by the height specified for the corresponding DXF layer. If no height has been specified, CST MICROWAVE STUDIO® will import the DXF data as an infinitely thin double-sided face. You can use this face as a basis for further construction, but the PBA® method always requires a finite thickness for the simulation. You may, for instance, use the *Objects* ⇨ *Thicken Sheet* operation to create a solid from the face.

The following picture shows a typical (although very simple) model imported from a DXF file:



If you zoom the view into the structure you can check the finite thickness of the imported metallization layer:



Please note: Due to the PBA® technique these thin layers do not slow down the solution process.

Even though the import of 2D files is usually much less critical than the import of 3D models, problems may sometimes occur. The following hints should help to make the import as robust as possible:

1. **Organize your structure into layers.** Always put the structural drawing and additional data (such as hatchings, measure lines, text, etc.) on different layers and only import the structural layers.
2. **Only import those parts you are actually interested in.** Since CST MICROWAVE STUDIO® needs to build three dimensional objects from the two dimensional input data, the complexity of the model is heavily increased during the import process. The deletion of unwanted objects after importing the complete structure is unnecessarily inefficient.
3. **Try to use polygons in the drawing of the 2D files as opposed to a collection of independent lines.** Due to some accuracy constraints or a less accurate drawing, independent lines may have gaps which must be closed with a lot of effort during the import process.

Following these rules will usually yield successful import results. However, if you experience problems, you should read the *Inspect and Repair the Imported Model* section later in this chapter in order to get some information on how defects in the model can be cured.

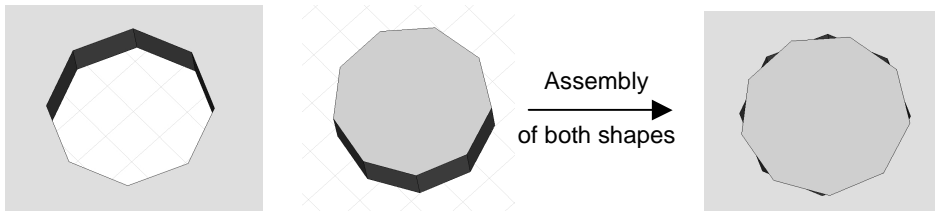
3.2 Import 3D Files

CST MICROWAVE STUDIO®'s software offers import filters for various 3D geometrical CAD formats:

- SAT (up to version 12)
- STL
- IGES (4.0, 5.2, 5.3)
- STEP (AP 203, AP 214)
- Pro/E (16 up to 2001)
- Catia 4

The SAT file format is the native data format of the ACIS kernel and should therefore be preferred whenever possible.

The STL data format only contains facet information. Importing this kind of data usually becomes problematic when more than one shape of this type is imported and the shapes touch each other. The following picture illustrates this problem:



Since the two faceted representations of the cylindrical hole and the cylinder are not identical, the assembly contains holes and overlappings rather than a cylinder perfectly fitting into the hole.

Please note: The STL import is often most useful in combination with the staircase meshing option. Please refer to the online documentation for more information.

The IGES and STEP data formats are based on standardized file format specifications, in contrast to proprietary file formats such as Pro/E and Catia 4.

The data translation process consists of two stages. First the CAD system converts and exports its data into a certain file format. Afterwards CST MICROWAVE STUDIO® reads the file and converts it into its own internal data structures. It is usually preferable to import the native data format of the CAD system in order to avoid the first data translation step. Since every translation step may entail a loss of accuracy, it is advantageous to avoid unnecessary translation steps.

With regard to standardized CAD data formats, the STEP format should be preferred to the IGES format because this also contains topology information about the relationship of neighbored faces. This important information must be recreated from pure geometry data if an IGES file is imported.

Therefore, the guidelines as to which data format should be used for which CAD system are as follows:

1. **CATIA 4:** Use the Catia 4 file format.
2. **Pro/E:** Use the Pro/E file format.
3. **ACIS based systems:** Use the SAT file format.
4. **Other CAD systems:** Try to use the STEP (AP203, AP 214) format, if available. Otherwise use the IGES format. If this is not available either, you have to use the STL format (which will most likely require to use staircase meshing).

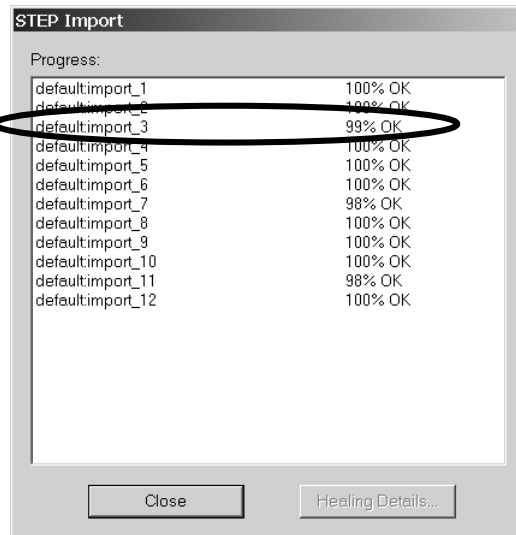
The following list of rules is applicable to all types of file formats and should help you to obtain best results for importing CAD data:

1. **Keep the complexity reasonably low:** Please bear in mind that you are going to run an electromagnetic analysis with your models. Importing an entire car including details, for instance, will exceed today's computational resources. Please import only the most relevant parts for highly complex models (e.g. the car's body).
2. **Avoid unnecessary blends in the CAD model:** Excessive application of blends will also increase the complexity in the data. You should always try to avoid unnecessary blends and other complex shape details (e.g. embossed text) in the exported files whenever possible.
3. **Avoid tiny model sizes:** Due to the limited numerical precision in the file formats you should use units in which the model has reasonably large absolute dimensions. If your model has a bounding box diagonal of 0.1 m, for instance, you should export it in mm in order to get a larger absolute diagonal of 100 (mm).
4. **Avoid Boolean operations before exporting the model:** Whenever possible, you should try to avoid Boolean operations between parts before exporting the model to a file. In the example of the STL import shown earlier, the problem is due to the insertion of a cylinder into a plate which then contains a hole. If the insertion had been omitted in this case, inserting the imported and healed shapes within CST MICROWAVE STUDIO® would not have caused any problems. This issue is most critical when faceted CAD data is used (e.g. STL), but still must be considered for the other more accurate data formats as well.
5. **Only export solid models:** CST MICROWAVE STUDIO® can only import solid models. Wireframe drawings are not supported.
6. **Try different export options:** Whenever you experience problems in any of the data translation steps, you should try to use different export options for your CAD system. You may need to experiment a bit in order to find the combination which works best for your system and your type of models.

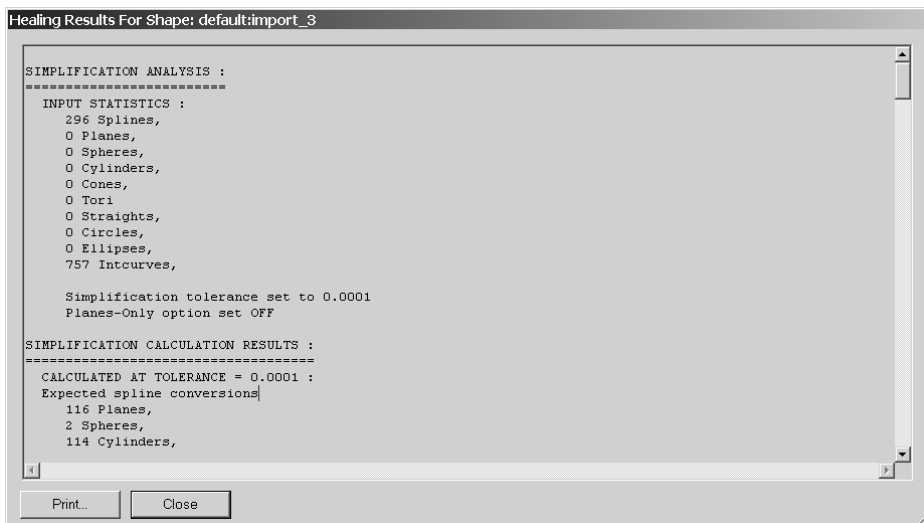
3.3 Inspect and Repair the Imported Model

As already outlined above, a healing step is necessary after importing a model from foreign CAD data. Therefore you should always use the option *Healing* in the CAD data import dialog boxes.

After importing and healing the data, a message box will be shown, informing you about the status of the model. The following picture shows such a message box as it will appear for a STEP file import:



Some shapes have been imported and healed without any problems (100% OK). However, some other shapes show minor defects indicated by a percentage of 98-99%. Clicking on the shape default:import_3 and pressing the *Healing Details* button would in this example provide a text output window as shown in the picture below:



In order to understand the healing process better, it is advantageous to know the meaning of this output. Thus we will in the following provide some explanations to the most important information in this text step by step.

The first section gives some statistics about the solid right after the import and before the actual healing process is started:

```

SIMPLIFICATION ANALYSIS :
=====
INPUT STATISTICS :
  296 Splines,
   0 Planes,
   0 Spheres,
   0 Cylinders,
   0 Cones,
   0 Tori
   0 Straights,
   0 Circles,
   0 Ellipses,
  757 Intcurves,

Simplification tolerance set to 0.0001
Planes-Only option set OFF

```

In the case investigated here, the imported shape consists of spline faces only. However, the healing technology tries to reduce the complexity of the imported data by converting as much of the imported spline faces to analytical trimmed surfaces as possible (simplification). The result of this stage is shown in the next section:

```

SIMPLIFICATION CALCULATION RESULTS :
=====
CALCULATED AT TOLERANCE = 0.0001 :
Expected spline conversions
  116 Planes,
   2 Spheres,
  114 Cylinders,
   8 Cones,
  48 Tori

SIMPLIFICATION FIX RESULTS :
=====
Simplification tolerance = 0.0001
no. of initial splines = 296
no. of final splines = 8

Number of analytics made from splines:
  116 Planes,
   2 Spheres,
  114 Cylinders,
   8 Cones,
  48 Tori
  757 Intcurves,
  115 Intcurves,
  408 Straights,
  234 Circles,
   0 Ellipses,

```

The simplification step was able to simplify 288 of the originally 296 spline faces to analytic geometry (e.g. planes, spheres, cylinders, etc.) here. The next section in the text files shows some information about the original status of the model's geometry:

```

GEOMBUILD ANALYSIS :
=====
geom build tol = 10
analytic solver tol = 0.0101
no. of edges = 757
no. of bad edges = 177
no. of coedges = 1514
no. of bad coedges = 360
no. of vertices = 471
no. of bad vertices = 51
no. of bad tangent edges = 44
no. of bad tangent edges analytic = 37
no. of G1 bad tangent edges analytic = 39
no. of bad tangent edges uv_uv = 1
no. of bad tangent edges boundary uv_uv = 1

```

```

no. of bad tangent edges uv_nonuv = 6
no. of bad tangent edges nonuv_nonuv = 0
no. of bad tangent edges 3_4_sided = 7
no. of surfaces = 296
no. of discontinuous surfaces = 0
percentage of good geom = 92

```

A lot of elements (171 edges and 51 vertices) have somehow invalid geometry data which indicates that this model clearly needs healing. The next sections in the text file show some more involved information and thus will be skipped in this introduction. However, the final section in the text file shows the result of the healing process:

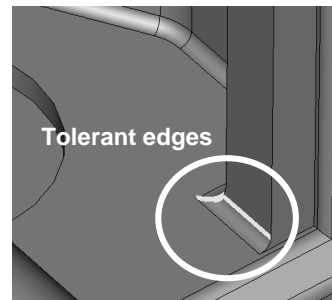
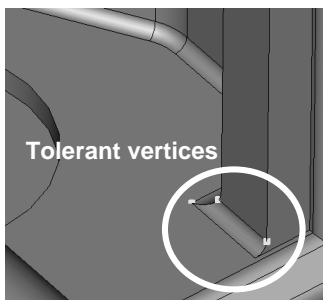
```

GEOMBUILD FIX RESULTS :
=====
Statistics of the healed body after geombuild fix :
no. of edges = 757
no. of bad edges = 2
no. of coedges = 1514
no. of bad coedges = 29
no. of vertices = 471
no. of bad vertices = 2
no. of bad tangent edges = 0
no. of bad tangent edges analytic = 0
no. of G1 bad tangent edges analytic = 7
no. of bad tangent edges uv_uv = 0
no. of bad tangent edges boundary uv_uv = 0
no. of bad tangent edges uv_nonuv = 0
no. of bad tangent edges nonuv_nonuv = 0
no. of bad tangent edges 3_4_sided = 0
no. of surfaces = 296
no. of discontinuous surfaces = 0
percentage of good geom = 99

```

The automatic healing was quite successful in this case. Only two edges and two vertices are still invalid.

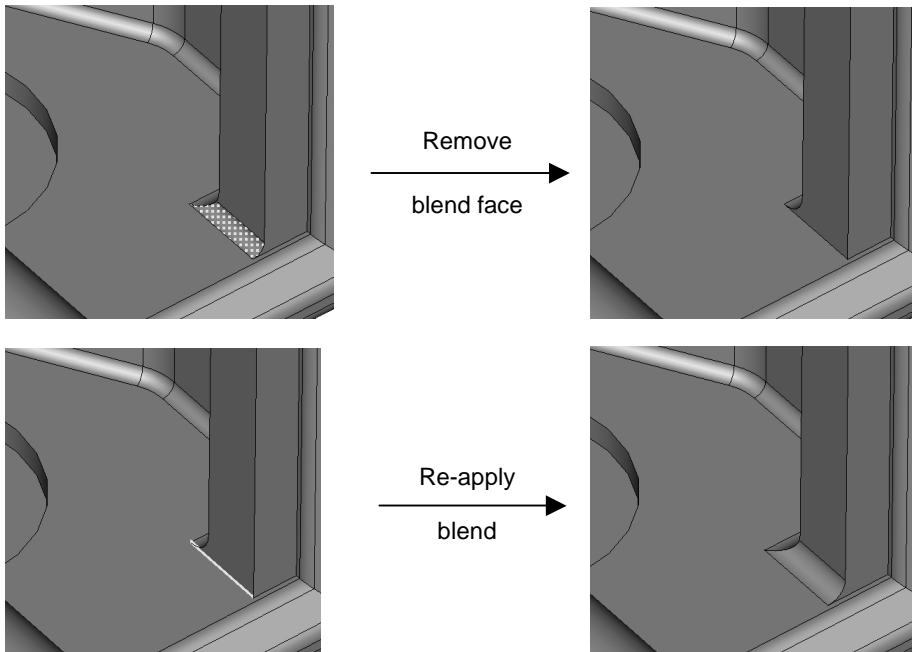
After getting the general information that there are some defects in the model, it is important to localize the problems. Usually the automatic healing will convert the incorrect elements to so-called *tolerant* objects. By applying tolerances, the modeler is able to perform a limited set of structural modifications on the model even if it is not 100% valid. Thus all erroneous elements become tolerant elements in the final stage of the automatic healing process. These tolerant elements can easily be visualized by selecting *Objects* ⇒ *Solid Healing Tools* ⇒ *Show Tolerant Vertices* and *Objects* ⇒ *Solid Healing Tools* ⇒ *Show Tolerant Edges*, respectively. Please note that a toolbar is available providing accelerated access to the healing tools.



In the example considered here, the problems arise mainly due to the imported blends' faces. Whenever you encounter such problems in the healing process we recommend contacting your CAD drawing department first. On many occasions a careful inspection of the CAD model at these locations may show some minor defects or inaccuracies in

the original model. After revising the drawing and re-importing the structure you should check whether the problem disappeared. If improving the quality of the CAD data is not possible or if the problem still persists, you may need to manually repair the model.

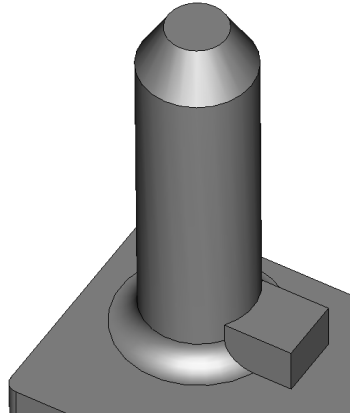
In the example shown here, it is sufficient to delete the inaccurate blend by using the *Local Modifications* tool. If you plan to re-apply the blend later and do not know its radius now, you should measure it before its removal. Afterwards you only need to select the blend's face by using the face pick tools and choose *Objects* ⇨ *Local Modifications* ⇨ *Remove Feature* from the main menu or the healing toolbar.



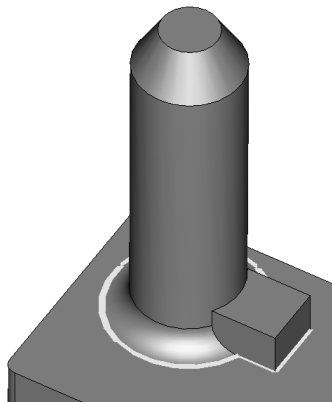
The procedure explained above is quite typical for the manual healing of invalid blends. However, in many cases it is advantageous to avoid these blends before exporting the model from the original CAD system. In many cases, the blends have little or no effect on the electromagnetic behavior of the structure, so there may be no need to re-apply them.

There are few areas in which the following sentence is more true than for CAD data exchange: **Keep it simple!**

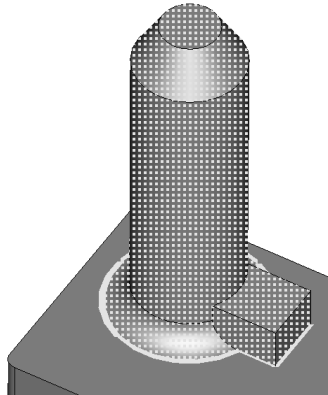
By investigating a second example the following steps will explain how unnecessary details can be simply removed from the model. Let us now assume that the pin shown in the picture below does not affect the electromagnetic properties and should thus be removed from the model:



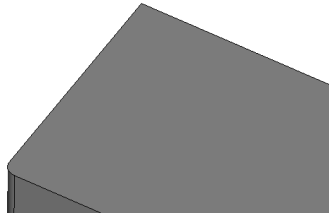
The easiest way to achieve this is to use the *Remove Feature* tool after selecting all faces of the unwanted structure detail. To easily obtain this face selection, you can activate the *Objects* ⇒ *Pick* ⇒ *Pick Edge Chain* tool (shortcut Shift+E) and double-click on the edge which connects the detail with the rest of the model:



After properly selecting this edge, you can activate the *Objects* ⇒ *Pick* ⇒ *Pick Face Chain* tool (shortcut Shift+F) and double-click on one of the detail's faces:



Now all faces which are connected to the currently picked one will be selected. However, the face selection will stop at previously picked edges. By proper selection of the picked edges you can thus advise the tool to select the faces of the unnecessary structure detail only. Finally the *Objects* ⇒ *Local Modifications* ⇒ *Remove Feature* tool can be used to remove the shape and to close the gaps:



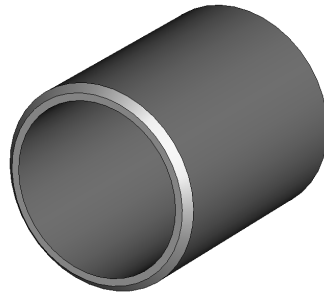
This section could only provide you with some ideas on how imported models can be healed manually if the automatic healing is unable to fix all problems. However, there are many more options available. The optimal strategies may also differ according to your structure and your originating CAD system. We strongly recommend that you attend a specialized training class on this topic in order to obtain best results. Please contact your support center for more details.

3.4 Parameterize the Imported Model

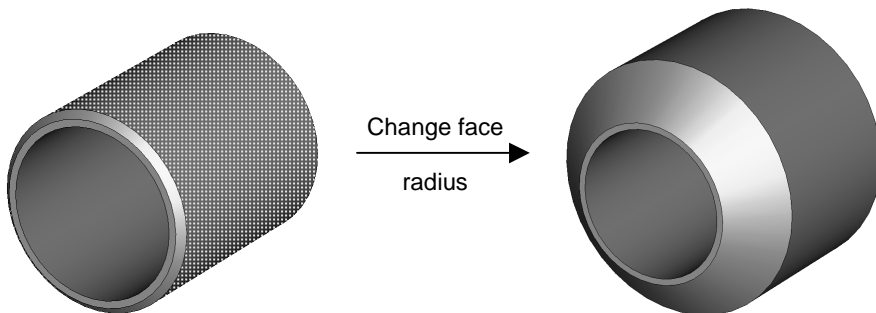
Quite often, the aim of the electromagnetic analysis is not just to know the given structure's behavior, but also to optimize it according to various goals. Such optimizations require the ability to modify the model even if it was obtained by importing a CAD data file.

CST MICROWAVE STUDIO® offers the outstanding possibility of parameterizing shapes for which no construction history exists by using so-called *Local Modifications*. The *Getting Started* manual contains a section which explains how these tools are generally applied. You should familiarize yourself with this functionality before you proceed to the next sections since they focus on the application of these operations to imported solids.

For the following explanations, let us assume that you wish to change the radius of the cylinder's outer face shown in the picture below:

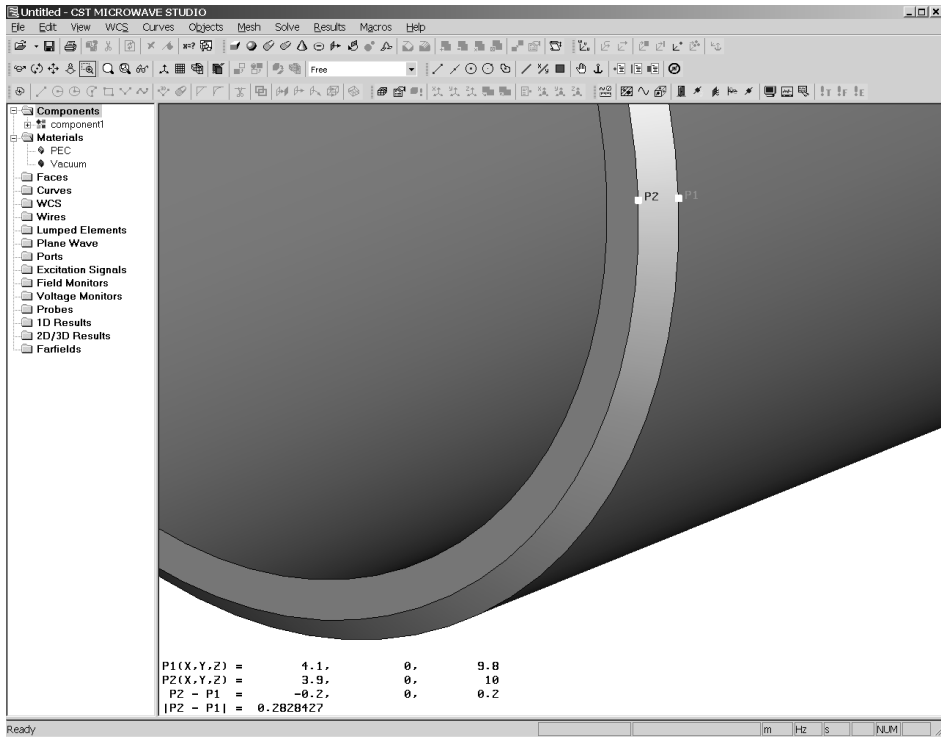


You could simply pick the corresponding face and apply the *Objects* ⇒ *Local Modifications* ⇒ *Change Face Radius* tool which would yield the following result:

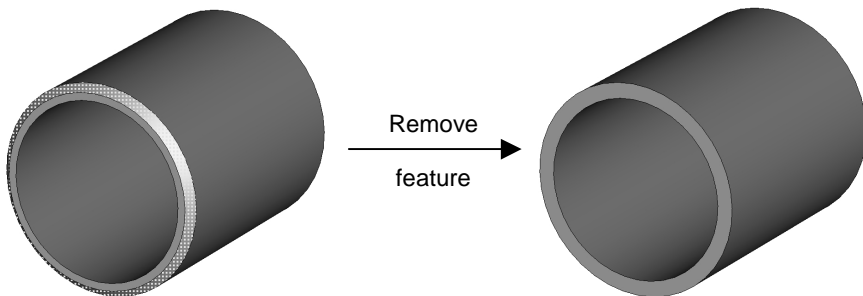


In order to close the resulting gaps, the chamfer's face is automatically extended up to its intersection with the modified cylindrical face. This results in an enlargement of the chamfer which is probably not what is required.

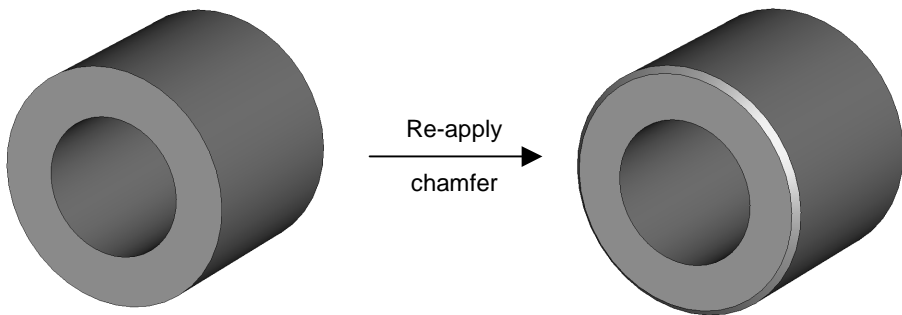
This problem can be circumvented by removing the chamfer first, then changing the face's radius and finally re-applying the chamfer. If you don't know the size of the chamfer you will need to measure it first by picking two points and checking the distance:



In this example, you would find the chamfer has a width size of 0.2. Now you can pick the chamfer's face and select the *Objects* ⇒ *Local Modifications* ⇒ *Remove Feature* tool:



The next step is to change the radius of the face and to re-apply the chamfer as shown in the picture below:



Please note that you can use a variable to specify the new radius of the face. This will allow you to fully parameterize (and thus later optimize) the model even though it has been imported from CAD data.

The example outlined above should provide you with some idea as to how imported models can be parameterized by using local modifications. Since many different (more or less efficient) solutions exist to achieve such a parameterization, we recommend that you visit a specialized training class on this topic. Please contact your support center for more information.

Chapter 4 – The Simulation Method

4.1 Background of the Simulation Method

The Finite Integration Technique (FIT)

CST MICROWAVE STUDIO® is a general-purpose electromagnetic simulator based on the Finite Integration Technique (FIT), first proposed by Weiland in 1976/1977 [1]. This numerical method provides a universal spatial discretization scheme, applicable to various electromagnetic problems, ranging from static field calculations to high frequency applications in time or frequency domain. In the following section the main aspects of this procedure will be explained and afterwards extended to specialized forms concerning the different solver types.

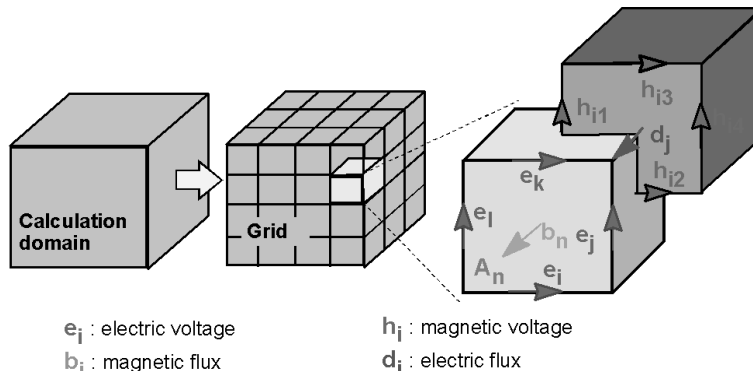
Unlike most numerical methods, FIT discretizes the following *integral* form of Maxwell's equations, rather than the differential one:

$$\oint_{\partial A} \vec{E} \cdot d\vec{s} = - \int_A \frac{\partial \vec{B}}{\partial t} \cdot d\vec{A} \quad , \quad \oint_{\partial A} \vec{H} \cdot d\vec{s} = \int_A \left(\frac{\partial \vec{D}}{\partial t} + \vec{J} \right) \cdot d\vec{A} \quad , \quad (1a)$$

$$\oint_{\partial V} \vec{D} \cdot d\vec{A} = \int_V \rho \, dV \quad , \quad \oint_{\partial V} \vec{B} \cdot d\vec{A} = 0 \quad . \quad (1b)$$

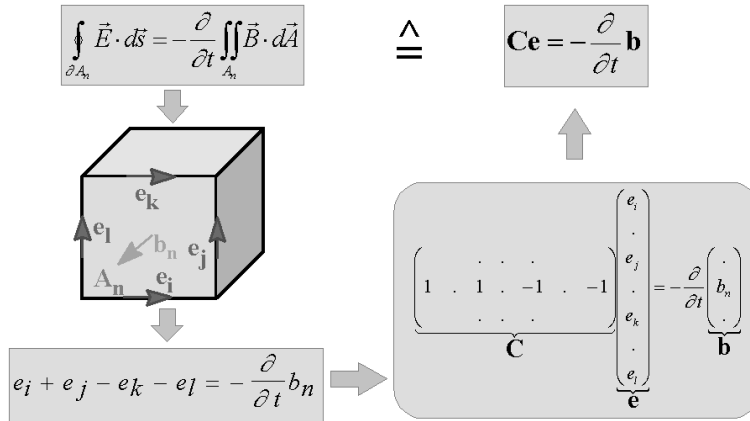
In order to solve these equations numerically a finite calculation domain is defined, enclosing the considered application problem. By creating a suitable mesh system, this domain is split up into several small cuboids, so-called grid cells. This *first* or *primary mesh* can be visualized in CST MICROWAVE STUDIO® in the *Mesh View*, however, internally a *second* or *dual mesh* is set up orthogonally to the first one.

The spatial discretization of Maxwell's equations is finally performed on this two orthogonal grid systems, where the new degrees of freedom are introduced as integral values as well. Referring to the following picture, the electric grid voltages \mathbf{e} and magnetic facet fluxes \mathbf{b} are allocated on the primary grid \mathbf{G} and the dielectric facet fluxes \mathbf{d} as well as the magnetic grid voltages \mathbf{h} on the dual grid $\tilde{\mathbf{G}}$ (indicated by the tilde):



Now Maxwell's equations are formulated for each of the cell facets separately as will be demonstrated in the following. Considering Faraday's law, the closed integral on the

equation's left side can be rewritten as a sum of four grid voltages without introducing any supplementary errors. Consequently, the time derivative of the magnetic flux defined on the enclosed primary cell facet represents the right-hand side of the equation, as illustrated in the figure below. By repeating this procedure for all available cell facets, the calculation rule can be summarized in an elegant matrix formulation, introducing the topological matrix **C** as the discrete equivalent of the analytical curl operator:



Applying this scheme to Ampère's law on the dual grid involves the definition of a corresponding discrete curl operator $\tilde{\mathbf{C}}$. Similarly the discretization of the remaining divergence equations (1b) introduces discrete divergence operators **S** and $\tilde{\mathbf{S}}$, belonging to the primary and dual grid, respectively. As previously indicated, these discrete matrix operators just consist of elements '0', '1' and '-1', representing merely topological information. Finally we obtain the complete discretized set of the so-called *Maxwell's Grid Equations* (MGE's):

$$\mathbf{C}\mathbf{e} = -\frac{d}{dt} \mathbf{b} \quad , \quad \tilde{\mathbf{C}}\mathbf{h} = \frac{d}{dt} \mathbf{d} + \mathbf{j} \quad , \quad (2a)$$

$$\tilde{\mathbf{S}}\mathbf{d} = \mathbf{q} \quad , \quad \mathbf{S}\mathbf{b} = \mathbf{0} \quad . \quad (2b)$$

Compared to the continuous form of Maxwell's equations, the similarity between both descriptions is obvious. Once again it should be mentioned that no additional error has yet been introduced. This essential point of FIT discretization is reflected in the fact that important properties of the continuous gradient, curl and divergence operators are still maintained in grid space:

$$\mathbf{S}\mathbf{C} = \tilde{\mathbf{S}}\tilde{\mathbf{C}} = \mathbf{0} \quad \Leftrightarrow \quad \text{div rot} \equiv 0 \quad (3a)$$

$$\mathbf{C}\tilde{\mathbf{S}}^T = \tilde{\mathbf{C}}\mathbf{S}^T = \mathbf{0} \quad \Leftrightarrow \quad \text{rot grad} \equiv 0 \quad (3b)$$

At this point it should be mentioned that even the spatial discretization of a numerical algorithm could cause long term instability. However, based on the presented fundamental relations (3), it can be shown that the FIT formulation is not affected by such problems, since the set of MGE's (2) maintain energy and charge conservation [2].

Finally, the missing material equations introduce the inevitable numerical inaccuracy due to the spatial discretization. By defining the necessary relations between voltages and fluxes their integral values have to be approximated over the grid edges and cell areas, respectively. Consequently, the resulting coefficients depend on the averaged material parameters as well as on the spatial resolution of the grid and are summarized again in correspondent matrices:

$$\vec{D} = \epsilon \vec{E} \quad \mathbf{d} = \mathbf{M}_\epsilon \mathbf{e} \quad (4a)$$

$$\vec{B} = \mu \vec{H} \quad \Rightarrow \quad \mathbf{b} = \mathbf{M}_\mu \mathbf{h} \quad (4b)$$

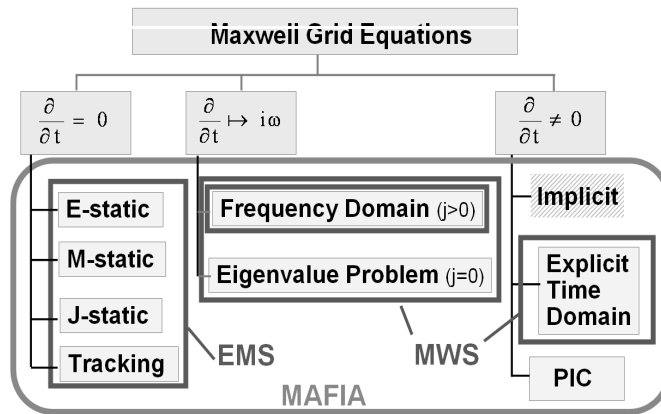
$$\vec{J} = \sigma \vec{E} + \vec{J}_s \quad \mathbf{j} = \mathbf{M}_\sigma \mathbf{e} + \mathbf{j}_s \quad (4c)$$

Now all matrix equations are available to solve electromagnetic field problems on the discrete grid space. The fact that the topological and metric information is divided into different equations has important theoretical, numerical and algorithmic consequences [2].

As demonstrated, the FIT formulation is a very general method and therefore can be applied to all frequency ranges, from DC to high frequencies (see figure below). All electromagnetic field regimes are already covered by CST's software package MAFIA®, whose development started more than 20 years ago.

Based on this long experience, the "STUDIO"-family development started in 1997. Here, several improvements concerning user interface, visualization and solver performance were integrated. However, the most fundamental change was the meshing strategy, the Perfect Boundary Approximation® (PBA) technique [3], particularly extended by the Thin Sheet Technique™ (TST) and the Multilevel Subgridding Scheme™ (MSS).

Finite Integration + PBA Statics to THz

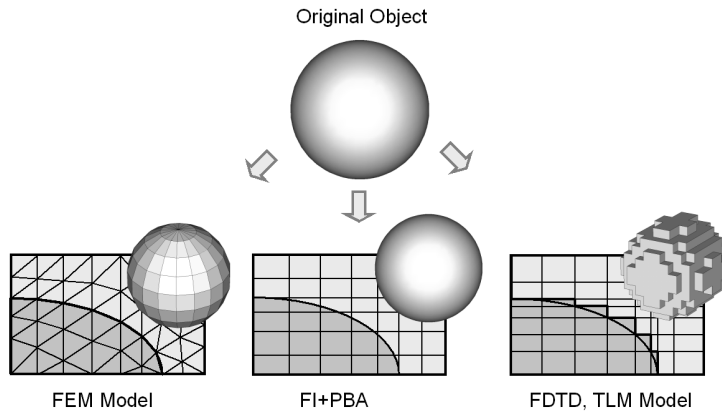


Currently, there are two packages available: CST EM STUDIO™ (EMS), the low-frequency package, which includes a variety of static and low frequency solvers and CST MICROWAVE STUDIO® (MWS), covering the high frequency range, both in transient and in harmonic state.

In the case of Cartesian grids, the FIT formulation can be rewritten in time domain to yield standard Finite Difference Time Domain methods (FDTD). However, whereas classical FDTD methods are limited to staircase approximations of complex boundaries, the PBA® technique applied to the FIT algorithm maintains all the advantages of structured Cartesian grids, while allowing an accurate modeling of curved structures.

In the figure below, the two "classical" geometry discretizations are shown: the Finite Element Method model on the left and the FDTD/TLM model on the right. The FIT model together with the PBA® theory in the middle combines the advantages of the other two

models. It offers both an excellent geometry approximation, without the segmentation of FE models or staircase approximation of FDTD codes, and high simulation speed, as in the FDTD and TLM methods.



The CST MICROWAVE STUDIO® Solvers

Three different solver types are available concerning high frequency electromagnetic field problems: transient, frequency domain and eigenmode solver. More details about these solvers can be found in the “Getting Started” manual. The purpose of the present section is to provide basic information about the solvers’ usefulness in simulation as well as to present the relevant equations on which the solvers are based.

□ Transient Solver

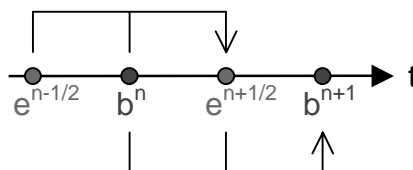
The CST MICROWAVE STUDIO® transient solver allows the simulation of a structure’s behavior in a wide frequency range in just a single computation run. Consequently this is an efficient solver for most driven problems (i.e. with nonzero sources), especially for devices with open boundaries or large dimensions. Please refer to section 4.5 to get some detailed information about the solver’s performance improvements.

The transient solver is based on the solution of the space discretized set of Maxwell’s Grid Equations (2), in which the time derivatives are substituted by central differences yielding the explicit update formulation for the loss-free case:

$$\mathbf{e}^{n+1/2} = \mathbf{e}^{n-1/2} + \Delta t \mathbf{M}_e^{-1} \left[\tilde{\mathbf{C}} \mathbf{M}_\mu^{-1} \mathbf{b}^n + \mathbf{j}_S^n \right], \quad (5a)$$

$$\mathbf{b}^{n+1} = \mathbf{b}^n - \Delta t \mathbf{C} \mathbf{e}^{n+1/2}. \quad (5b)$$

Regarding the relations above, the calculation variables are given by the electric voltages and the magnetic fluxes. Both unknowns are located alternately in time, as in the well-known leap-frog scheme, as demonstrated below:



For example, the magnetic flux at $t = (n + 1)\Delta t$ is computed from the magnetic flux at the previous time step $t = n\Delta t$ and from the electric voltage at half time step before, at $t = (n + 1/2)\Delta t$.

□ Frequency Domain Solver

The CST MICROWAVE STUDIO® frequency domain solver is useful for simulating relatively small-size problems, when the structure's behavior is required in just one or a few frequency points. Another application is given by cases where arbitrary periodic boundaries are needed.

The frequency domain solver is based on Maxwell's equations in the harmonic case ($\partial/\partial t \rightarrow i\omega$), which, in the loss-free case, lead to the following second-order relation:

$$\left(\text{curl}\mu^{-1}\text{curl}-\omega^2\varepsilon\right)\vec{E}=-i\omega\vec{J} \Rightarrow \left(\tilde{\mathbf{C}}\mathbf{M}_{\mu^{-1}}\mathbf{C}-\omega^2\mathbf{M}_{\varepsilon}\right)\mathbf{e}=-i\omega\mathbf{j} . \quad (6)$$

□ Eigenmode Solver

The CST MICROWAVE STUDIO® eigenmode solver allows the computation of the structure's eigenmodes and the corresponding eigenvalues. For closed structures (filters, resonating cavities, etc.) this is probably the solver of choice.

The eigenmode solver is based on the eigenvalue equation for non-driven and loss-free harmonic problems. The solution is obtained in the loss-free case via a Krylov-Subspace or Jacobi-Davidson method:

$$\text{curl}\mu^{-1}\text{curl}\vec{E}=\omega^2\varepsilon\vec{E} \Rightarrow \tilde{\mathbf{C}}\mathbf{M}_{\mu^{-1}}\mathbf{C}\mathbf{e}=\omega^2\mathbf{M}_{\varepsilon}\mathbf{e} . \quad (7)$$

For lossy problems, a Jacobi-Davidson solver is used.

For the calculation of highly resonant structures, the eigenmode solver offers an efficient way to derive the correspondent S-parameters by use of the modal analysis together with a dense spectrum calculation.

References

- [1] Weiland, T: A discretization method for the solution of Maxwell's equations for six-component fields: Electronics and Communication, (AEÜ), Vol. 31, pp. 116-120, 1977.
- [2] Weiland, T.: Time Domain Electromagnetic Field Computation with Finite Difference Methods. International Journal of Numerical Modelling, Vol. 9, pp. 295-319, 1996.
- [3] Krietenstein, B.; Schuhmann, R.; Thoma, P.; Weiland, T.: The Perfect Boundary Approximation technique facing the challenge of high precision field computation: Proc. of the XIX International Linear Accelerator Conference (LINAC'98), Chicago, USA, pp. 860-862, 1998.

4.2 Error Sources / Sources of Inaccuracy

Every method which tries to describe a real world problem using a numerical model conceals the danger of introducing errors, either because the model itself is not identical to the actual device, or because the numerical simulation is subject to errors. This section gives you an overview about the most important error sources. Many of them may be negligible in practice, but you always should be aware of them in order to make sure that you can rely on your results.

Simulation Model is not in Agreement with Reality

- **Geometric dimensions** are wrong or geometric details have been neglected. This happens quite often and can only be avoided by comparing the model carefully with the drawing and the real structure. The influence of small details can be studied by carrying out the same simulation with and without details.
- **Material parameters** are wrong, either because they are unknown and had to be chosen somehow, or because a value was taken which is valid for another frequency band but not for the band of interest. It might be necessary, especially when investigating a broader band, to enter the correct frequency dependency of the material values. For example, for some dielectrics a constant $\tan \delta$ is an appropriate choice rather than a constant conductivity.
- The **source of excitation** was not chosen correctly, e.g. a perfectly matched microstrip port instead of a coax fed microstrip with normalized impedance. Furthermore, for open ports such as microstrips or coplanar lines, the port size should be chosen to be sufficiently large in order to correctly capture the field pattern of the propagating modes. You should also bear in mind that discrete ports are typically not perfectly matched, even when they have the same impedance as the waveguide.
- The same considerations are valid for **output ports**, for which you should try to find a simulation model that best fits the measurement structure.
- Sometimes the **environment** is not considered correctly. Objects that are placed near the measured structure may influence the result and in this case should be included in the simulation model.

Inaccuracies due to the Simulation

- **Discretization error**: To ensure accurate results the electromagnetic fields have to be sampled sufficiently in space. Generally, the accuracy of the field solution increases with a finer mesh and it can be proven that convergence is ensured by the FI method. CST MICROWAVE STUDIO® supports convergence studies by automatic adaption of the mesh density parameters and by visualization of the convergence progress (see section 4.3).
- **Truncation error**: If the calculation is stopped before the time-domain signals have decayed down to zero, a so-called “truncation error” is introduced. A more detailed discussion of this topic can be found in section 4.4.
- The **geometry error** is the deviation between the entered CAD structure and the simulation model. Some methods that are based on orthogonal grids convert rounded structures into so-called staircase models before starting the simulation.

In Finite Element meshes, rounded structures are replaced by piecewise straight objects (Triangularization, see section 4.1). With the employed PBA technique (Perfect Boundary Approximation™), the geometrical error becomes very small or even negligible, especially for rounded conductors. However, if an object becomes small compared to the size of the mesh cells, the geometry error may not always be neglected.

- Some inaccuracy may be introduced through **boundary conditions**. In the case of radiating structures, the open space is simulated by so-called open boundaries. The employed PML technique (Perfectly Matched Layer) is well tested and very accurate even when the structure is near to the object. CST MICROWAVE STUDIO® chooses automatically the optimum size of the surrounding simulation box but in some rare cases it may be worthwhile checking the sensitivity of the result to changes of this box.
- **Waveguide ports** have a specific open boundary technique in which the waves are separated into their mode patterns. This technique allows even higher accuracies than the PML boundary condition for the surroundings. In cases of inhomogeneous ports such as microstrip lines, combined with very broadband excitations, it might be necessary to adjust special parameters in order to keep the error below -50 or -60 dB. As mentioned before, the size of the waveguide port also has to be defined large enough.
- Even if the field values are calculated correctly, **interpolation errors** may still occur, e.g. when deriving secondary field quantities or when calculating the field values at locations other than the grid edges.
- **Numerical errors** may be caused by the finite representation of numbers, but can typically be neglected for the explicit algorithm in time-domain. However, in case of the eigenmode or frequency domain solver a numerical error may arise when the correspondent matrix system is solved.

If you ensure that these requirements are fulfilled, you can be confident in your simulation results.

In the following two sections you will learn more about setting up the mesh parameters and performance improvements of the simulation.

4.3 Mesh Generation

After you have set up your model geometrically and assigned the appropriate power sources and boundary conditions, your model has to be translated into a computer accessible format. As you have already learned, the FI method, which is the foundation of CST MICROWAVE STUDIO®, is a volume discretization method. Your calculation domain has to be subdivided into small cells on which Maxwell's Equations have to be solved. CST MICROWAVE STUDIO® uses the FI formulation on structured orthogonal Cartesian grids. The mesh influences the accuracy as well as the speed of your simulation, so it is worthwhile spending a little bit of time on understanding this process.

Methods of Mesh Generation

In general there are three different ways to define a mesh: manually, automatically and adaptively. Their principles will be introduced briefly to you, before we discuss details of the implementation in CST MICROWAVE STUDIO®.

Manual Meshing

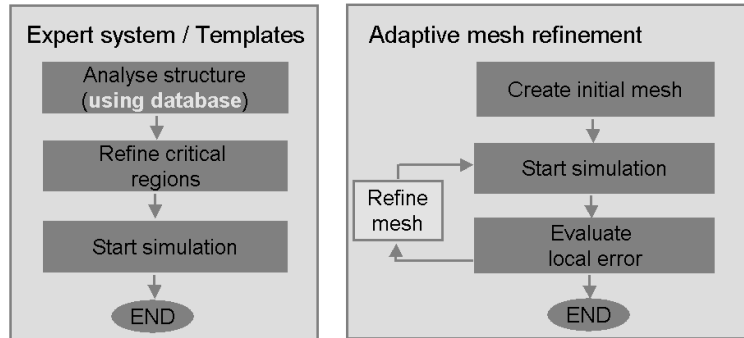
A manual mesh can be defined at any time, even before the geometrical model is generated. Consequently, you have to take care of every single mesh line, whether it fits the geometrical model or not and if it supports the electromagnetic field solution in an adequate way. This is an old fashioned way of working which we do not encourage you to follow.

Automatic Mesh Generation – Expert System

This is probably the most effective way of working with CST MICROWAVE STUDIO®. The mesh generator determines the important features of your structure and automatically creates a mesh, which represents your structure and the fields equally well. This means that the frequency range and dielectrics, metallic edges etc. are considered, as well as the facility to link certain mesh properties to single structure elements. At the first level, this mesh generator is governed by only a few settings, but there is a multitude of additional possibilities to influence the meshing. Furthermore, Templates facilitate the mesh creation and with a little bit of experience you will be able to obtain reliable solutions in a minimum of time.

Adaptive Meshing

Adaptive meshing replaces your expertise by repeatedly running the simulation and evaluating the solutions. Usually regions with high field concentration or field gradients are recognized, where the mesh is to be locally refined. If the deviation in the results falls below a given accuracy level, the adaption is terminated. This approach always improves the start solution at the expense of the simulation time. Since the CST MICROWAVE STUDIO® expert mesher always guarantees a reasonable initial mesh and thus a good starting solution for the mesh adaption process, the number of passes will be small.



This diagram shows the comparison of expert system meshing and adaptive mesh refinement.

However, modifications to the structure force conventional mesh adaption algorithms to start again from the beginning with each and every change of parameter. Therefore we have developed a first stage of the *adaptive expert system mesh refinement*, that trains our expert system for a given structure. It can then maintain the mesh properties while the associated structure parts are slightly modified.

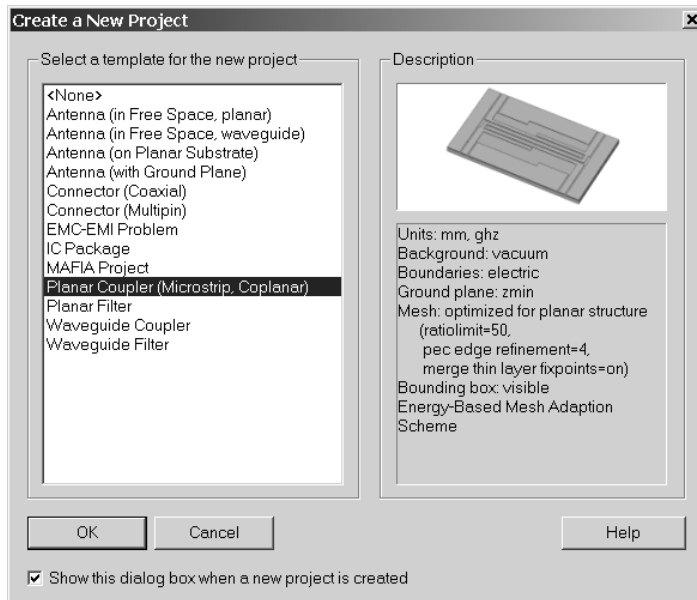
Naturally, you can train your own expertise by analyzing the steps the adaptive mesh refinement takes. Your knowledge can then increase the simulation speed.

Global Mesh Parameters

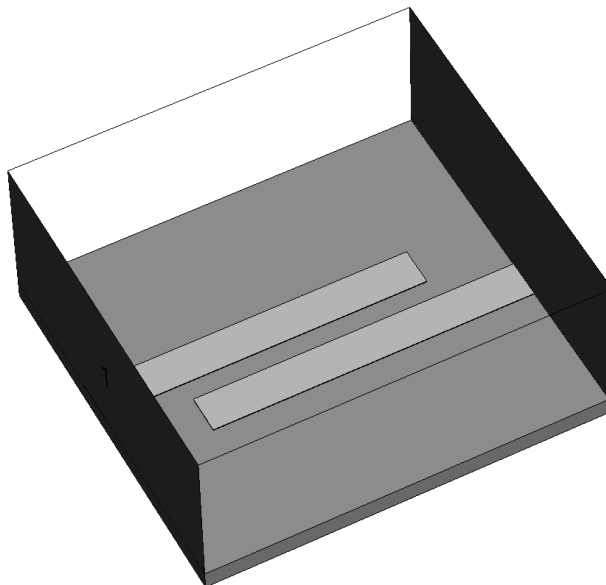
After this overview of meshing techniques, we will step into the details of the CST MICROWAVE STUDIO® mesh generation. CST MICROWAVE STUDIO® offers you a lot of adjustable parameters, distinguishing local from global ones. So, let us start with the most important global ones:


- Lines per wavelength
- Lower mesh limit
- Ratio limit (or smallest mesh cell)
- Refinement at PEC-edges

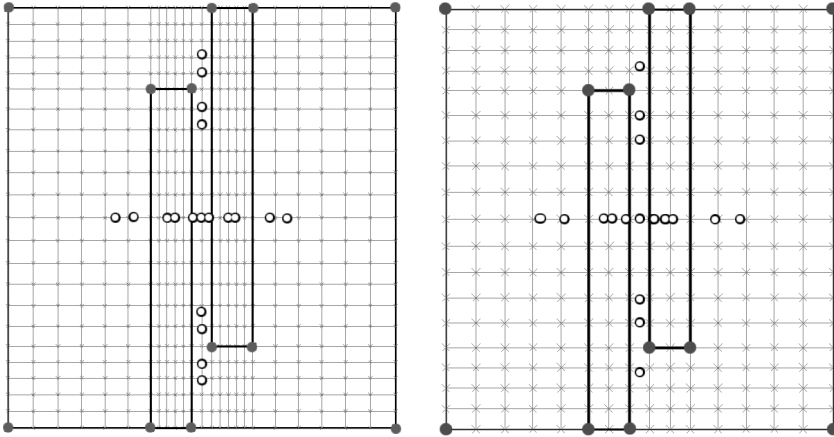
These parameters are also set by the templates and the adaptive expert system mesh refinement. Let us start a new project, e.g. a simple coupled microstrip line. In this case the following dialog appears:



We select the planar coupler template. The info box tells us which parameters are set by this template. It informs you that the mesh is optimized for planar structures which will now be explained. After creating our structure, adding some additional air above the microstrips and defining the ports, it will eventually look like this:




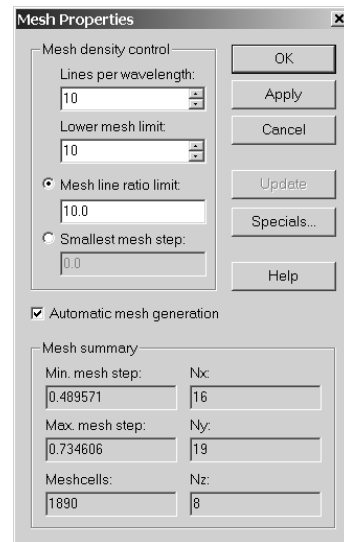
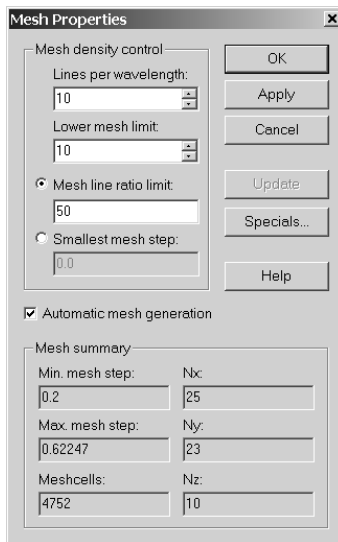
Before we can look at the mesh (*Mesh* ⇌ *Mesh View* or the corresponding icon ) , we have to define the frequency range. This is essential in order to set up the grid.



The picture on the left side shows the grid created using the template and on the right, the grid using the default settings. In order to discuss the differences, we have to summarize the features highlighted in this plot.

The red dots (grey in the picture) in the model are critical points (so-called *fixpoints*), at which the expert system finds it necessary to set mesh lines. They can be found on the bounding box, at the end of straight lines, at circle centres and radii. In addition to these fixpoints, the yellow dots (white in the picture) show points where the automatic mesh generation finds the need for improving the mesh density. Eventually, the metallic edges of the microstrips are marked by light green lines (dashed black lines in the picture).

Different settings of the main meshing parameters explain the different grids seen above. To have a closer look, just open the special settings dialog (*Mesh* ⇌ *Mesh Properties* or the corresponding icon ). Again, the template based settings are shown on the left, the default settings on the right.



□ Lines per wavelength

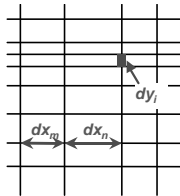
The lines per wavelength parameter describes the spatial sampling rate of the field. A value of 10 means that a plane wave propagating along one of the coordinate axes through our structure at the upper frequency of our defined frequency range is sampled at least 10 times (this is also the reason why the frequency range has to be defined before you can have a look at the mesh). Propagation in dielectric materials is taken into account here. This setting affects the mesh if the wavelength of the highest frequency is smaller than the structure size, which is the case in our example. Increasing this factor results in a better overall structure resolution.

□ Lower mesh limit

If, in contrast, the structure is smaller than the shortest wavelength, the lower mesh limit parameter is most likely to take effect. The diagonal of your calculation domain is taken and simply divided by the value of the parameter, which guarantees a minimum structure resolution.

□ Mesh line ratio limit / smallest mesh step

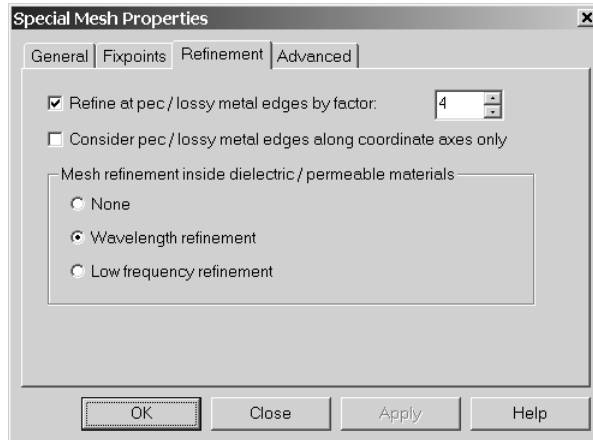
The mesh line ratio limit is probably the most important parameter, it finally governs the achievable structure resolution and might have a perceptible influence on the simulation time (see section 4.4). Generally it tells you how much smaller the smallest distance between two mesh points is, compared to the largest distance.



In our figure the ratio limit is determined by dx_n/dy_i . Practically it means that, if lines per wavelength and the ratio limit are set to 10, the smallest achievable spatial resolution is one hundredth of the wave length. Alternatively you could specify the absolute value of the smallest mesh step.

The mesh properties dialog provides you with some statistics about the total number of mesh cells, the smallest and the largest mesh step size and the number of mesh lines in each direction of the grid.

However, there are even more screws to tune your mesh! You can access them via the *Specials* button in the *Mesh Properties* dialog box. We will focus on the fourth important parameter that can be found in the sheet *Refinement*.

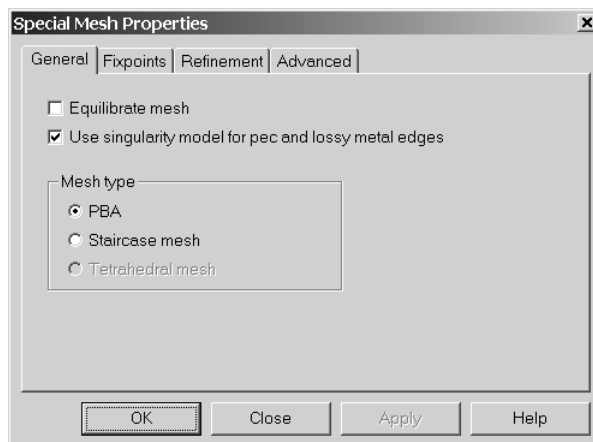


Refine at PEC edges

You have already learned that CST MICROWAVE STUDIO® recognizes metallic edges. As you know, the field singularities at these edges require a fine grid resolution in order to resolve them adequately. This factor allows you to define a mesh refinement particularly at PEC edges. In the planar coupler template it is set to 4, while the default value is 2. Please look again at the comparison of the two meshes and you will notice the local difference in grid resolution.

The option *Wavelength refinement* ensures that the lines per wavelength setting will also be fulfilled for the shortened wavelength inside the substrate.

Only PEC edges running along gridlines are marked with a green line indicating that a singularity model is applied here. You can also confine the refinement to those edges by applying the “*Consider pec / lossy metal edges along coordinate axes only*” check button. The singularity model itself can be switched on or off in the sheet *General*:

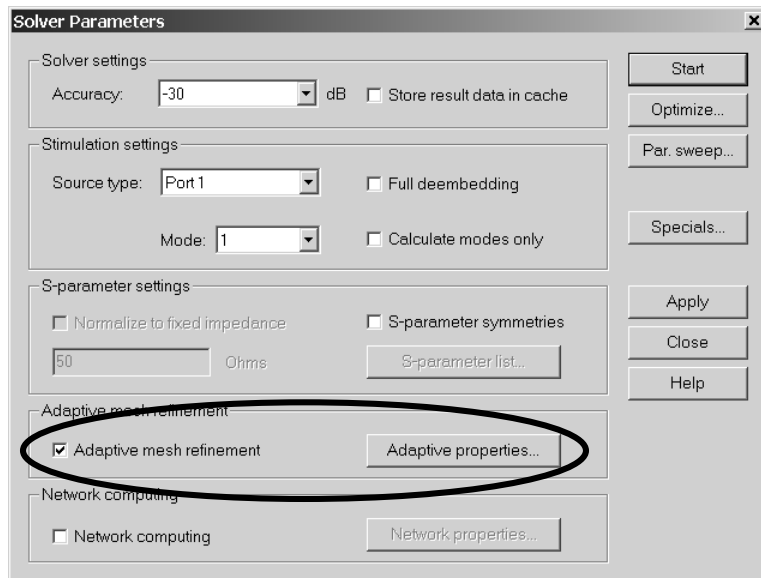


The *Equilibrate mesh* option allows you to reduce the local ratio between two adjacent mesh steps, in our diagram defined by dx_n/dx_m . This will result in a graded mesh.

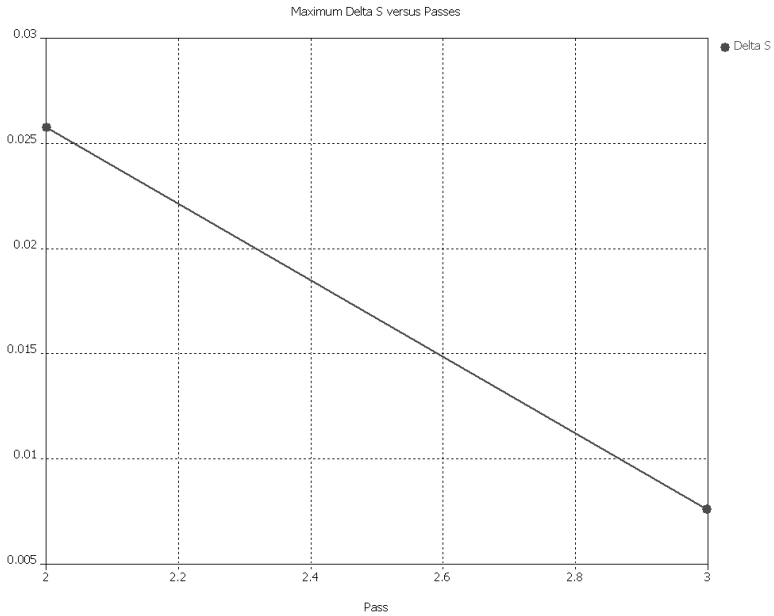
Switching off the PBA mesh for the whole structure is not recommended but might be useful for structure parts (will be discussed later).

Mesh Adaption

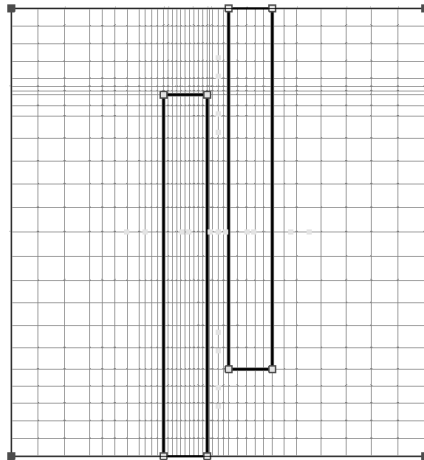
Simply starting the simulation will provide you with some numerical results. The question of how trustworthy they are remains open. The FI-method only guarantees that the more mesh cells you apply, the more accurate the solution will be. On the other hand, this could imply a huge resources overhead concerning the calculation time. You, therefore, have to find a trade-off between accuracy and time. The templates are set up in a way that this trade-off points more in the direction of saving resources. The adaptive meshing process helps you to gain confidence in the results. It is simply switched on by checking *Adaptive mesh refinement* button in the *Solver parameters* menu and then started by pressing *Start solver* (shown here for the transient solver):



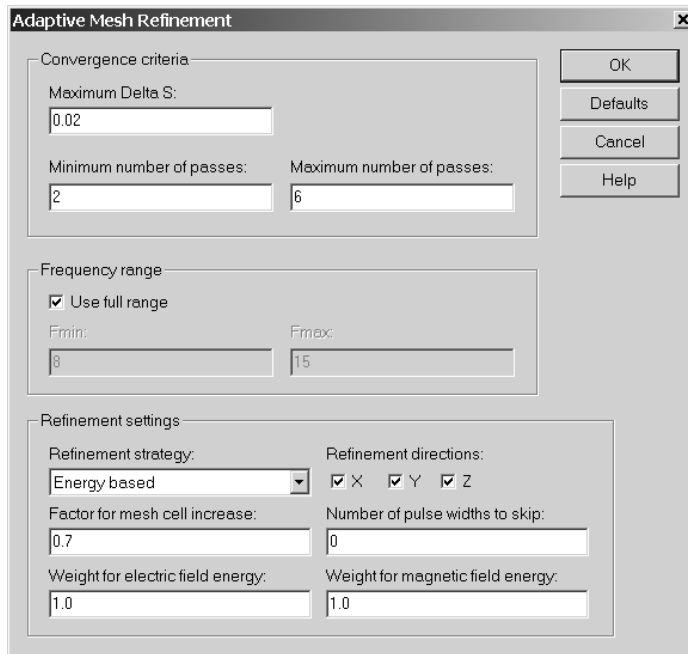
In our example the adaption stops after 3 passes, if the initial mesh was created using the template (after 6 passes using the default mesh). The S-parameter error (*Delta S*), which is defined as the maximum deviation of the absolute value of the complex difference of the S-parameters between two subsequent passes has fallen below 1%.



Looking at the mesh again we notice that a refinement along the PEC edges has been applied. In particular, additional mesh lines have been inserted between the strips and at the edges of the stripline belonging to the excitation (due to the high currents):



In the *Solver parameters* dialog you may have noticed the *Properties* button in the *Adaptive mesh refinement* frame already. Of course, the mesh adaption process is customizable, too:



Firstly, you can set the termination limit concerning the S-parameter error (*Delta S*). However, in order to determine any error at all, you need at least 2 runs. Naturally you can increase this number to judge the accuracy of the solution. The total simulation time for the adaption process is limited by the maximum number of passes. Finally, you can choose between two refinement strategies, the energy based (*adaptive energy based mesh refinement*), which is the default for planar structures, and the expert system based (which may be more useful for other structures and therefore is set by other project templates as default).

Energy based mesh refinement

This is the conventional approach to mesh adaption. While performing the electromagnetic field simulation, the energy density in the complete volume is calculated. Regions with high energy density and high field gradients are identified and the mesh is locally refined there, considering the settings shown in the dialog box above.

The refinement per pass is steered by the *Factor for mesh cell increase*. A setting of 0.7 means that 0.7 times more mesh cells are used for the next calculation than have been used for the previous one. In other words: the number of mesh cells increases by about 70 percent from pass to pass. Moreover, the refinement can be restricted to the X,Y,Z direction only.

This is a field based approach, which delivers mesh refinement in strategically important regions. The disadvantage of this method is that, in general, the refinement regions are not coupled to the structure parts nor to the global meshing parameters. Parameter studies and optimizations will lead to repeated adaption runs, which will diminish the throughput.

□ Adaptive expert system mesh refinement

The mesh parameters are directly influenced by this algorithm, i.e. the mesh is trained, and the obtained knowledge is kept for further calculations. The contents of the dialog box change if the *Refinement strategy* is set to *Expert system based*:

The screenshot shows the 'Adaptive Mesh Refinement' dialog box with the following settings:

- Convergence criteria:**
 - Maximum Delta S: 0.02
 - Minimum number of passes: 2
 - Maximum number of passes: 6
- Frequency range:**
 - Use full range
 - Fmin: 8
 - Fmax: 15
- Refinement settings:**
 - Refinement strategy: Expert system based
 - Refinement directions: X, Y, Z
 - Mesh increment: 5
 - Number of pulse widths to skip: 0
 - Weight for electric field energy: 1.0
 - Weight for magnetic field energy: 1.0

The *mesh increment* parameter determines the absolute step for *Lower mesh limit* and *Lines per wavelength* per adaption pass. The advantage of this method is that it is tolerant to structure changes and therefore will speed up highly accurate parameter studies and optimizations.

At the end of an successful expert system based mesh adaption, CST MICROWAVE STUDIO® informs you that the expert system has now been trained to yield accurate results for this structure. This means that it will not be necessary to start further adaption runs with this structure, since the quality of the mesh will be maintained even if you change the coupling length or the distance between the strips in order to perform parameter studies or run an automatic optimization. This is done by the *adaptive expert system mesh refinement*, which currently only adapts the so far discussed global meshing parameters.

The idea behind the *adaptive expert system mesh refinement* will be further developed to influence the local mesh settings as well. We see this as a very powerful method that works in synergy with our expert system meshing and will reveal its full strength together with our future developments.

4.4 Mesh Tuning

So far we have discussed how to obtain simulation results and how to check their reliability with a minimum effort spent on the meshing process. If you now feel confident with the global parameters and switches, we can go on to improve the mesh with respect to simulation time and memory requirements (note that there is an additional section 4.4 on simulation tuning). However, some points concerning the mesh need to be clarified first.

Ratio Limit, Number of Mesh Points and Simulation Time

The relation between the number of mesh points and the simulation time is quite simple. In the transient module an explicit time integration scheme is used (i.e. the solution is derived by simple (sparse) matrix vector multiplications), which means that the numerical effort scales linearly with the number of mesh points. Consequently the easiest way of reducing the calculation effort is to use symmetry conditions, which reduce the number of mesh points by up to a factor of 8.

The influence of the ratio limit is more complex, since explicit time integration schemes are conditionally stable. The stability limit for the time step Δt is given by the so-called Courant-Friedrichs-Levy (CFL)-criterion:

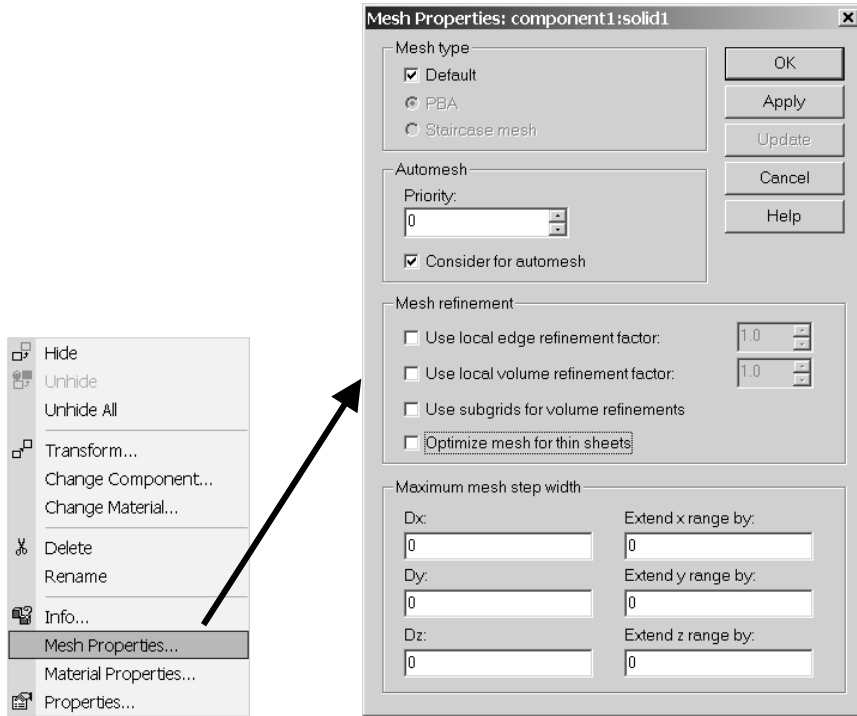
$$\Delta t \leq \frac{\sqrt{\varepsilon\mu}}{\sqrt{\left(\frac{1}{\Delta x}\right)^2 + \left(\frac{1}{\Delta y}\right)^2 + \left(\frac{1}{\Delta z}\right)^2}}$$

An illustrative explanation of its meaning is that it avoids energy jumping over one mesh cell in one time step. Thus the CFL-criterion has to be obeyed in every single mesh cell. This means that if there is just one single small mesh step in the grid, the global time step will also be small. Consequently, if possible, you should avoid small mesh cells, what is supported by PBA®, TST™ and MSS™ technology.

The trade-off between accuracy and resource consumption has to be discussed in a more detailed way if the use of the global parameters leads to an unreasonably large number of mesh cells. CST MICROWAVE STUDIO® also allows you to modify mesh parameters locally. These modifications are a significant part of our expert system meshing, and are linked to the geometric objects and are maintained during parameter studies and optimizations. Furthermore, these mesh parameters are helpful when handling flawed and faceted geometries or for the definition of meshing hierarchies.

Local Mesh Parameters

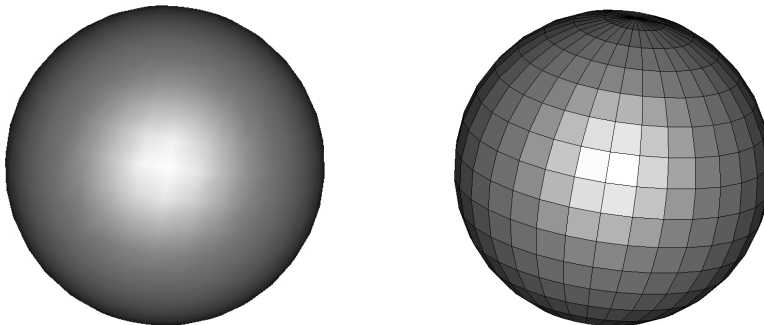
Until now we have discussed the straightforward approach to create a mesh and study the convergence by setting only a few parameters. CST MICROWAVE STUDIO® offers further useful ways of influencing the mesh, that can be applied for different reasons, but mostly to save mesh points and thus decrease the simulation time. After selecting one shape and pressing the right mouse button, you can open the *Mesh properties* dialog box from the context menu:



It allows you to change the influence that a single object has on the grid. Probably most important among the options is the decision whether the object shall be taken into account for the grid or not.

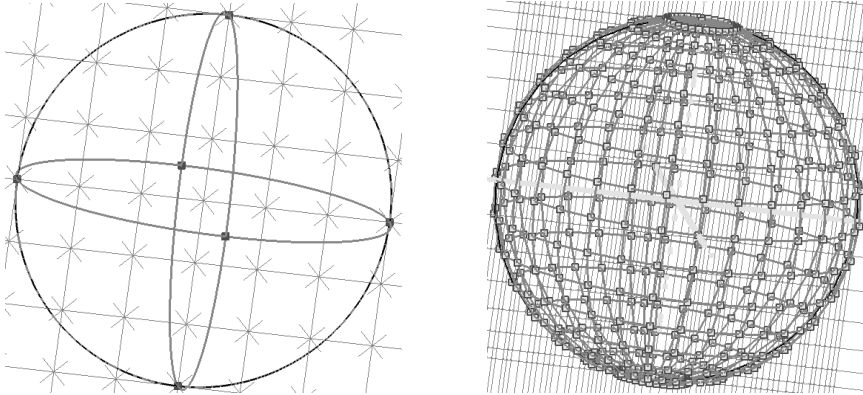
- Consider for automesh

If the *Consider for automesh* option is switched off, CST MICROWAVE STUDIO® will neither consider the dielectric properties, nor any fixpoint of the structure, not even the bounding box. This option is particularly useful when dealing with imported data. Consider that you import a sphere, but instead of an analytic description (left), you get a faceted representation of it (right):

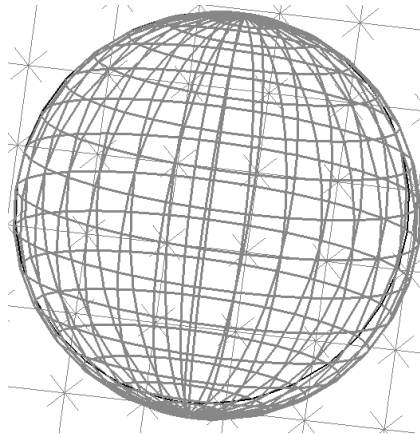


Although this is not a bad approximation of the sphere geometry, it is obviously not perfect. The CST MICROWAVE STUDIO® mesher always tries to model the shapes exactly as they have been defined. As a consequence, it will mesh this faceted approximation with fixpoints at the end of every straight line. And as you see below,

instead of a very efficient grid representation (left), you end up with an unnecessarily high number of mesh points (right):



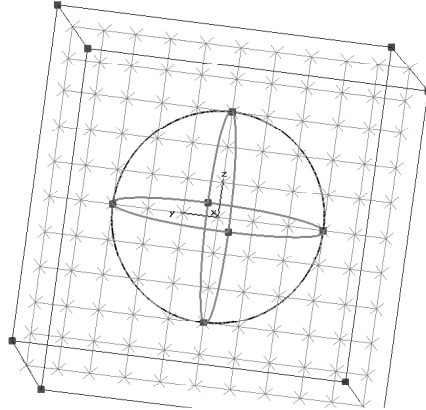
In case of a simple sphere a reconstruction can be easily carried out, but in general you have to accept your imported geometry. Switching off the *Consider for automesh* option will result in a representation like this.



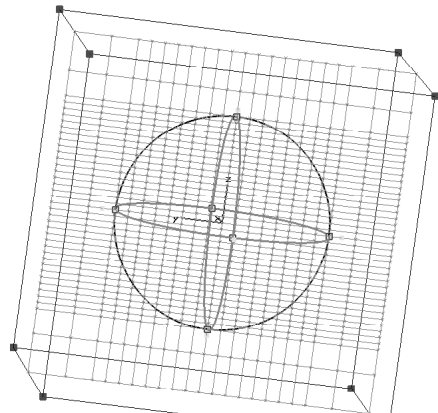
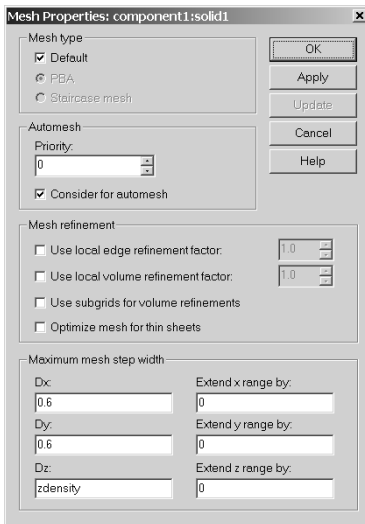
Now the fixpoints at the circumference are missing, which leads to a more efficient mesh representation of the object. This option is also useful if your imported geometry is flawed, because the fixpoints are usually located at the most critical points of a geometric object, or if your structure contains an object that forces a lot of additional meshlines, but does not contribute much to the field solution to your problem.

□ Maximum mesh step width

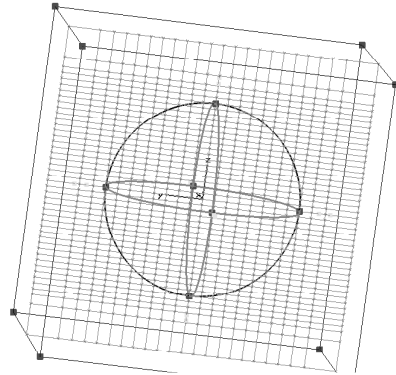
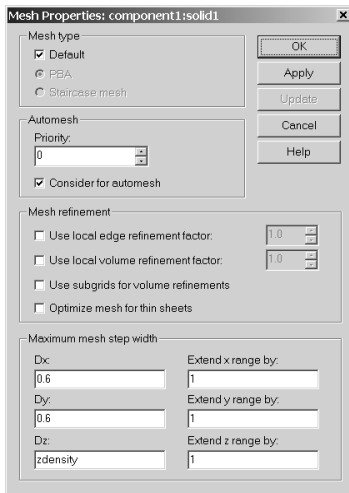
This option allows you to locally increase the mesh density. It is important if you want to maintain a coarser grid in the peripheral regions but accurately sample the fields in the area of interest. For simplicity, let us again consider the sphere with its default mesh:



If we want to refine the mesh in the sphere's volume, different values can be assigned to the three spatial directions in the *Maximum mesh step width* frame. You may also make use of parameters (e.g. $zdensity = 0.3$ here) which can be comfortably adjusted in the *Parameter List* (*Edit* ⇒ *Parameters...*).



This setting only affects the volume occupied by the object. Especially in the case of metallic objects, the adjacent surrounding space is even more important. CST MICROWAVE STUDIO® offers the possibility to simply increase the bounding box of the object since the step width settings are applied to the extended range as well:



□ Priority

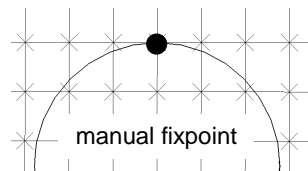
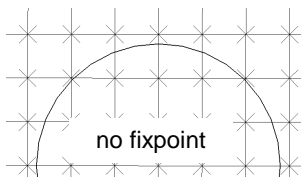
The *Priority* setting determines a weight for the fixpoints of one geometrical object compared to another one. If two fixpoints of different objects are so close together that the ratio limit does not allow the resolution of both of them, then a mesh line will be fixed to the point with the higher priority.

□ Mesh Type

The meshing algorithm can be individually determined for every object. The *Default* setting means that the global algorithm is taken, chosen in the *Specials* menu of the global *Mesh properties* dialog (usually PBA®). For corrupted import data, where the healing algorithm failed, or for very large and complex CAD imports, switching to the standard staircase mesh might be helpful.

□ Manually set Fixpoints

Above we have seen how too many fixpoints can be avoided by switching off the *Consider for automesh* option. As a result, the fixpoints at the circumference of the imported sphere are also missing. If you want to make sure that fixpoints are allocated to the circumference of the shape, it is possible to set them manually. The picking mechanism that you are already using for the structure generation creates fixpoints when applied in the *Mesh view*:

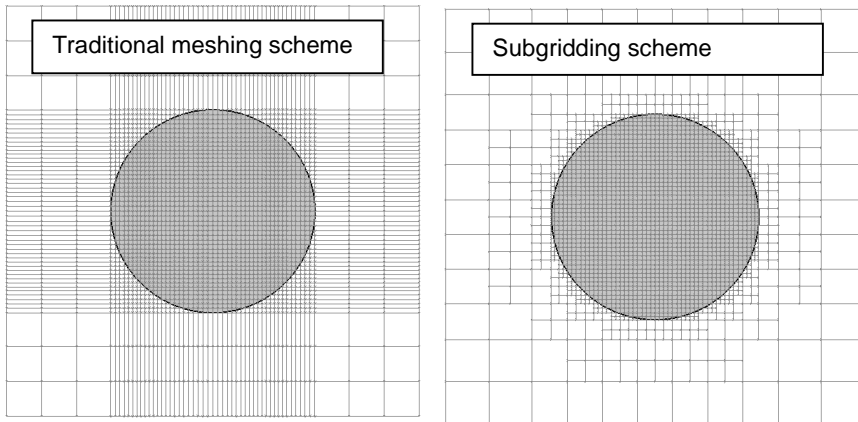


The manually defined fixpoints are marked by blue dots and have the highest priority in the mesh priority scheme. A case where fixpoints are helpful is at corner points where a strong field gradient appears.

This section has introduced you to the CST MICROWAVE STUDIO® meshing schemes and settings. For additional information please consult the online help!

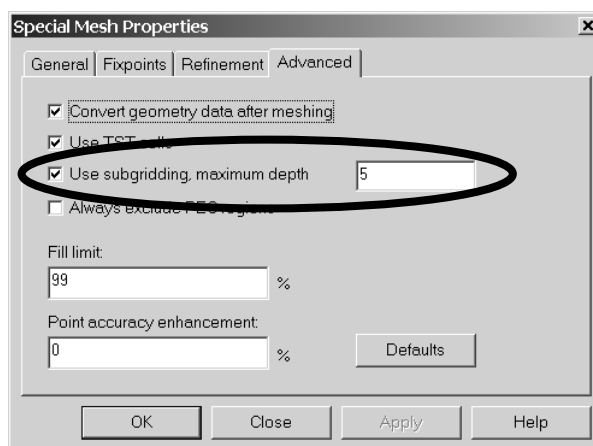
4.5 The Multilevel Subgridding Scheme (MSS™)

The Multilevel Subgridding Scheme (MSS™) marks a major improvement in the meshing flexibility and efficiency. In contrast to the traditionally used mesh where the mesh lines have to be extended through the entire computation domain, the MSS™ allows to locally concentrate the mesh. The following picture shows a mesh refinement inside a sphere by using a traditionally refined mesh as compared to the new subgridding technique:



This extension has been implemented in version 5.0 for the transient solver only, but will be added to the other solvers in future releases as well. Furthermore, the current implementation in version 5.0 does not reduce the amount of memory space required, but will improve the simulation speed. The next service packs will therefore be aimed towards improving the memory efficiency of the scheme.

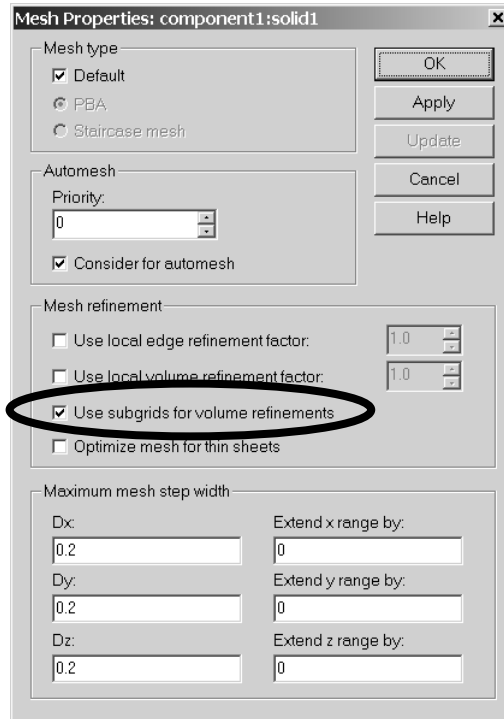
The MSS™ is activated by default and can be switched off in the *Mesh* → *Global Mesh Properties*, *Specials* dialog box in the *Advanced* tab:



The subgridding scheme can be de-activated by un-checking the *Use subgridding* box. The MSS™ can recursively subdivide any mesh cell into eight sub-cells where each

sub-cell can then be split further. The *maximum depth* of this recursive scheme can be specified in the dialog box in order to avoid excessively refined meshes.

Once the MSS™ is globally activated, it needs to be applied on a shape by shape basis by choosing Edit ⇒ *Mesh Properties* once a shape is selected:

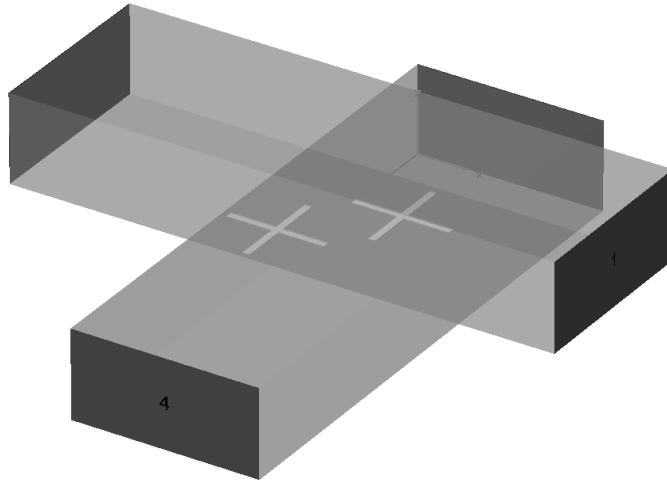


In this dialog box, the subgridding can be activated for a particular shape by checking the *Use subgrids for volume refinements* box. In particular, any volumetric refinement for the shape (e.g. dielectric refinements, local volume refinement factors or maximum step widths) are then realized by using subgrids as far as possible. Please note that only refinements based on a factor 2^n can be achieved by MSS™.

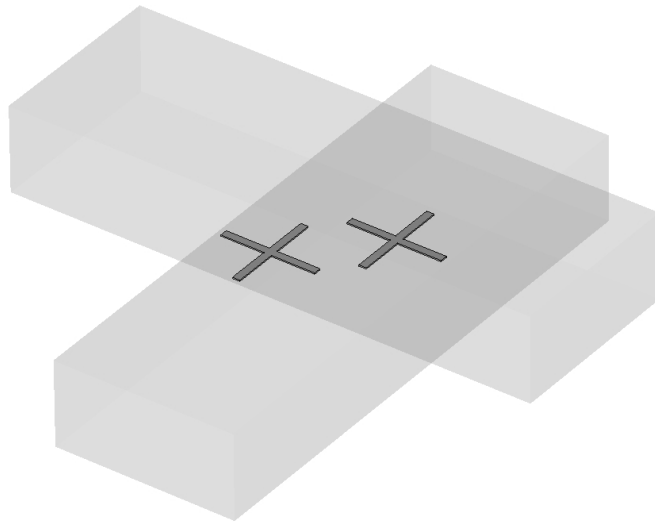
The following example should clarify the refinement strategy: Assuming that a shape has a dielectric constant of 4 and a local volume refinement factor of 3, the total volumetric refinement factor is $\sqrt{4} * 3 = 6$. The subgridding can only realize refinements of 4 or 8, so a refinement of 4 (two levels) will be achieved by using subgrids and the remaining factor $6 / 4 = 1.5$ will be additionally applied by using conventional mesh refinement.

Please note that changing the mesh properties can also be applied to a multiply selected group of shapes.

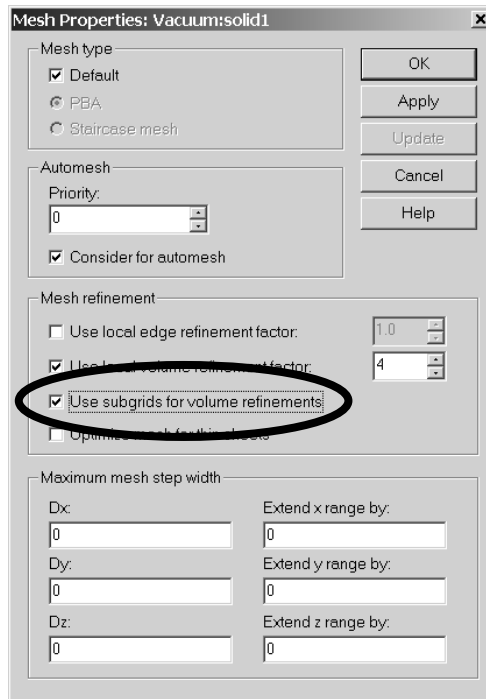
The following two examples (a waveguide and a stripline) should give you some basic idea about how the subgridding scheme can be efficiently used.

Example 1: Two cross slot coupled wave guides

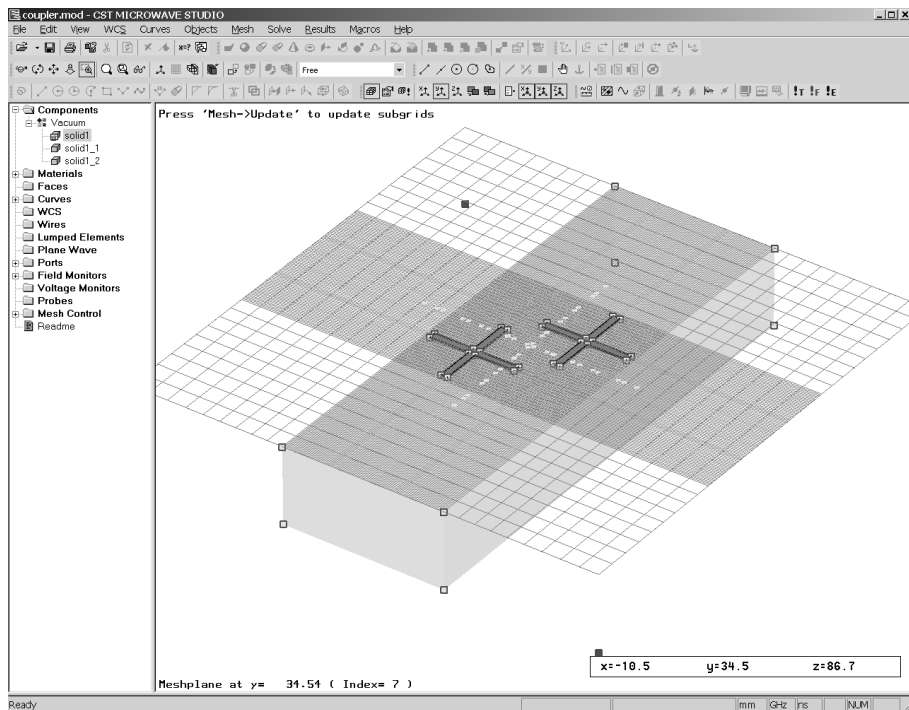
In order to efficiently apply the subgridding scheme, the shapes which need to be refined should be isolated such that their mesh properties can be set individually. In this particular case, the iris blends need to be refined:



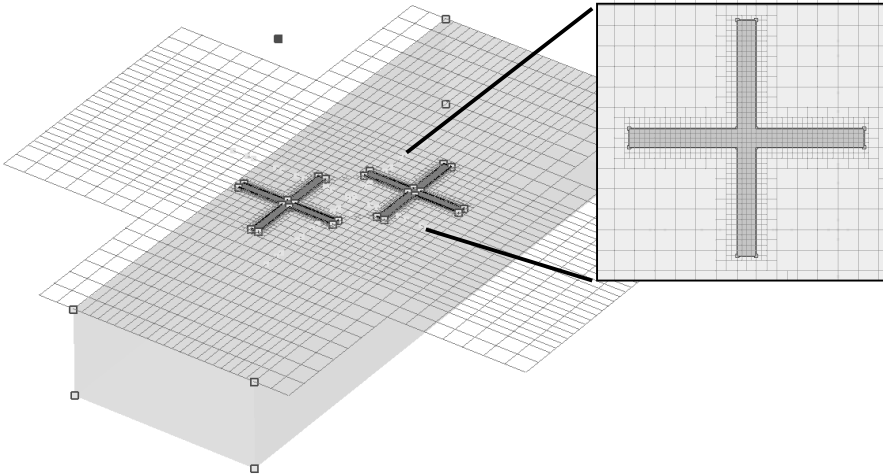
In a next step, a volumetric refinement by a factor of 4 is applied to this shape by using a local volume refinement factor and activating the subgridding in the *Shape Mesh Properties* dialog box:



In the mesh mode, you will first see a conventional mesh and a message that the subgrids need to be updated. The subgrids are not automatically calculated after every mesh change due to performance reasons.

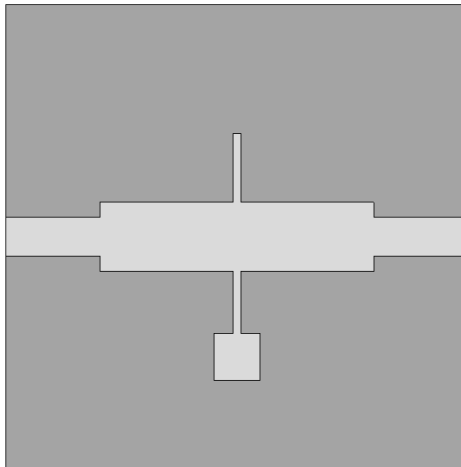


Pressing Mesh⇒*Update* will then calculate the subgrids and also automatically exclude perfectly conducting regions from the calculation domain.

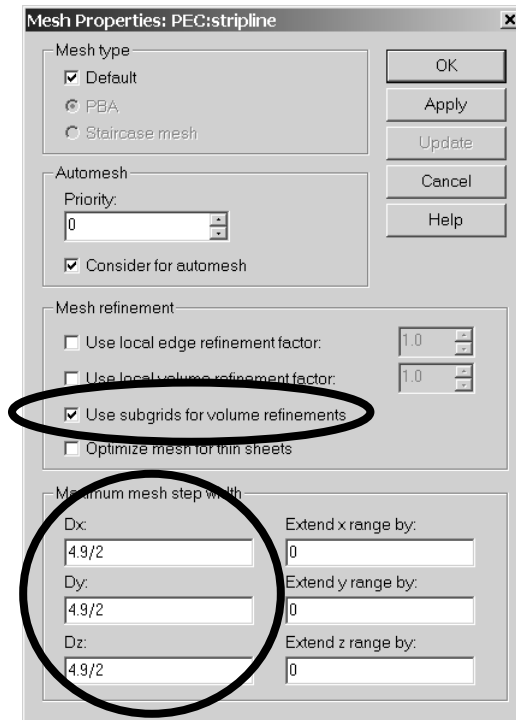


As it can be clearly seen, the MSS™ results in a highly efficient mesh here.

Example 2: A microstrip phase bridge

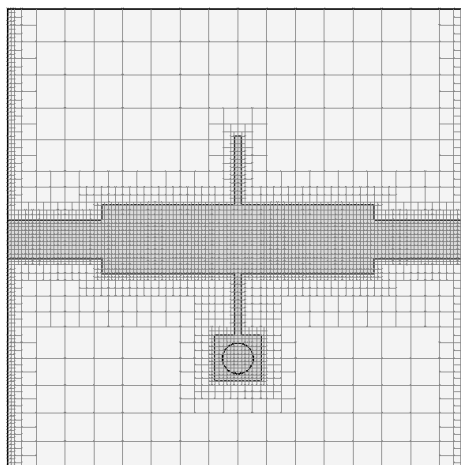


This kind of microstrip device usually requires a fine mesh around the metallisation strips. Therefore, the refinement settings are made again in the *Shape Mesh Properties* dialog box for the metallisation shape:



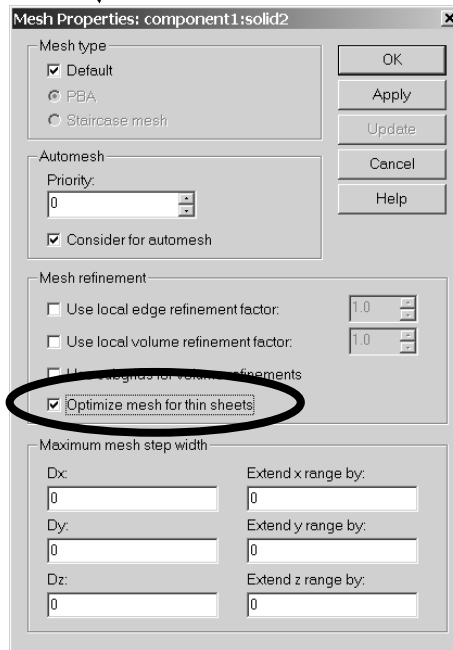
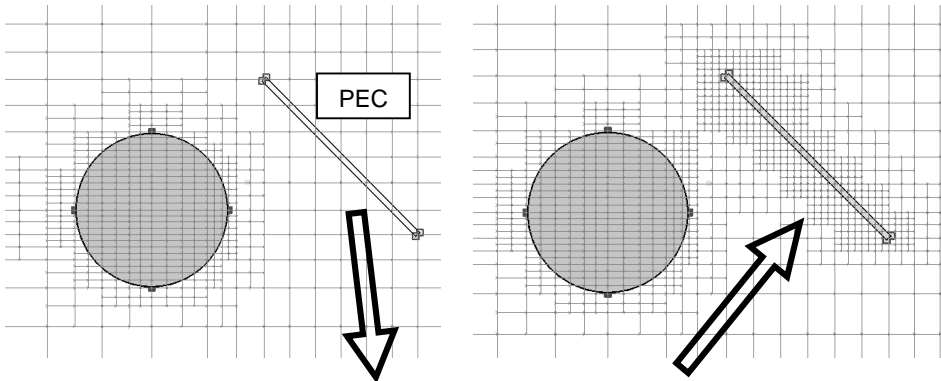
In this example, the smallest stub width is 4.9 mm. Therefore, a good choice for the discretization step width of the metallisation shape is $4.9 \text{ mm} / 2$ to obtain at least two mesh cells across the stub's width.

In order to specify a volumetric refinement (remember that the MSS™ only considers this type of refinement), you need to enter the maximum mesh step width in every spatial direction. This holds even if the refinement is mainly required in the conductor's plane. The refinement in the perpendicular direction is quite useful anyway to obtain accurate impedance calculations for the microstrips. The following picture shows the mesh in the plane of the metallisation after the Mesh \Rightarrow Update was performed:

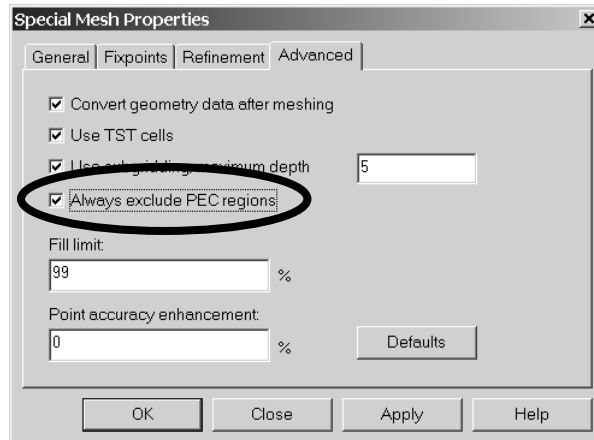


Please note that ports and objects made of lossy metal materials (as well as lossy metal boundaries) do not currently allow transitions between several levels of subgrids and are therefore refined as well. This limitation can reduce the efficiency of the scheme and will be addressed in future releases.

The TST™ model generally can be applied in combination with the subgridding scheme. The thin sheets, however, should be located on the finest grid level in order to achieve optimal accuracy. The *Optimize mesh for thin sheets* option in the *Shape Mesh Properties* Dialog box should therefore be activated for thin perfectly conducting sheets which are not aligned with the mesh planes:



The automatic exclusion of perfectly conducting materials can save a lot of computer resources. Please note that the detection of these domains is only active when the subgridding has been activated for at least one shape. Some structures, however, do not need subgrids at all, but the exclusion of perfect conducting volumes may still be desirable. In such cases, the domain exclusion feature can still be used by checking the *Always exclude PEC regions* box in the *Mesh* ⇒ *Global Mesh Properties, Specials* dialog box in the *Advanced* tab:



Please note: The parameter *Ratiolimit* only applies to the top-level mesh. It is recommended to use a rather low *ratiolimit* (e.g. 3-5) since submeshing can reach all required resolutions on one of the lower mesh-levels. A *Ratiolimit* of 3, combined with 5 submesh levels can resolve a factor $3 * 2^5 = 96$ between biggest and smallest meshstep.

Please refer to the online documentation for more information about the MSS™. We also strongly recommend that you participate in one of our special training classes held regularly at a location near you. Please ask your support center for details.

4.6 Performance Improvements

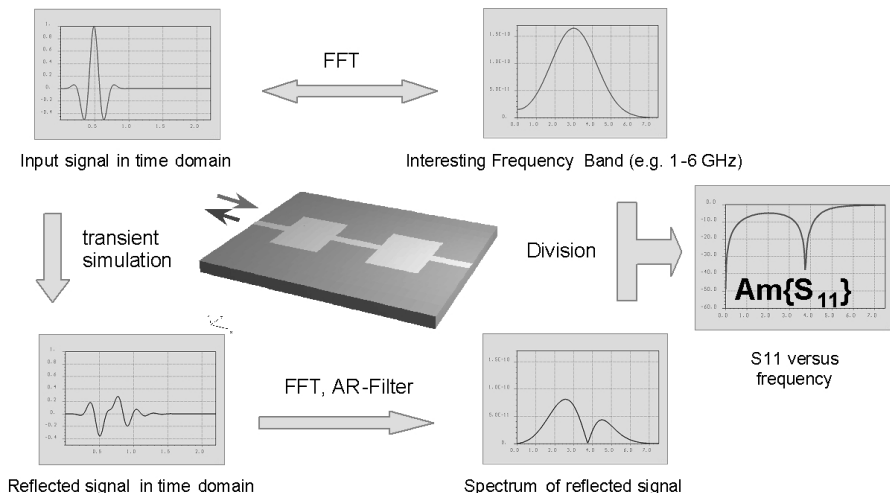
The following section shows how to increase the performance of your simulation runs and discusses some specific features of the time domain approach.

General Hints

- Always make use of geometric symmetry planes and of all known S-parameter symmetries.
- Avoid an unnecessarily large calculation domain size.
- Consider using the multi-processor support option.
- Use a CPU with an adequate clock rate.
- Choose the right solver module (e.g. eigenmode solver for resonant closed structures, please see also “Which Solver to use” in the online help).
- Consider splitting the structure into smaller parts and combining them in CST DESIGN STUDIO™.

Special Time Domain Topics

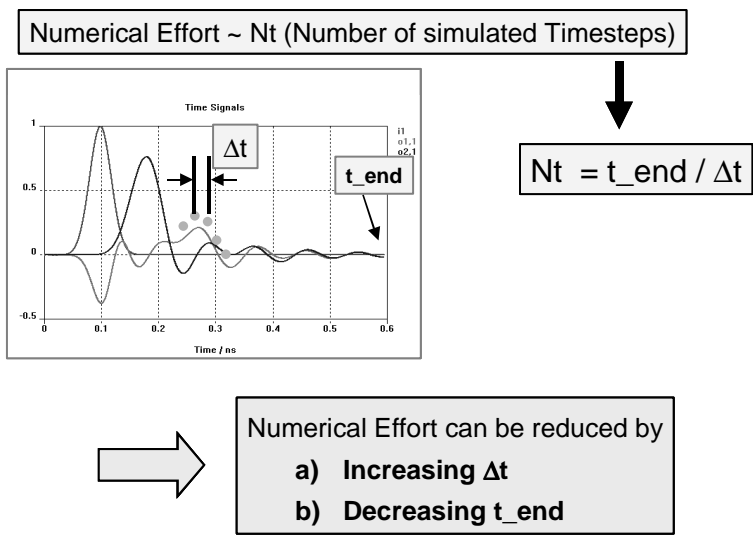
Time domain solvers naturally operate with time pulses, which can be easily transformed into the frequency domain by making use of a Fast Fourier Transformation (FFT). The S-parameters can then be derived from the resulting frequency domain spectra:



For instance, a division of the reflected signal by the input signal in the frequency domain yields the reflection factor S11. Within just one run in the time domain the full broadband information for the frequency band of interest can be extracted, without the risk of missing any sharp resonance peaks.

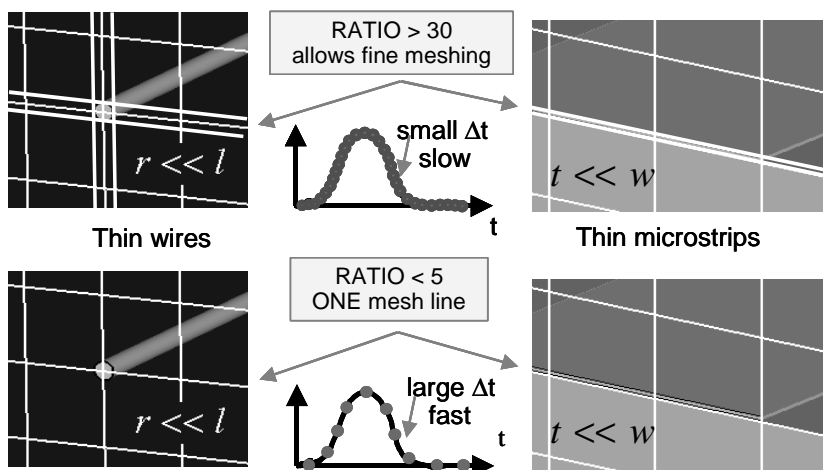
CST MICROWAVE STUDIO® automatically calculates the appropriate excitation time pulse from the frequency range setting. The default Gaussian shaped pulse guarantees a smooth spectrum in the frequency domain, which allows a reliable calculation of the S-parameters. The number of frequency samples in *Solver Parameters* ⇒ *Specials* ... ⇒

Solver \Rightarrow Number of F-Steps determines how many frequency points are calculated in the FFT. This does not require much calculation effort. Therefore, there is no need to reduce this number (default is 1000), which in fact would decrease the quality of the S-parameter plot.



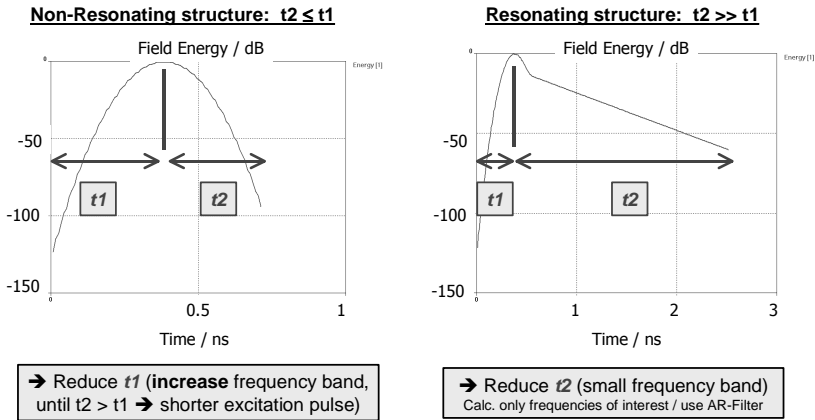
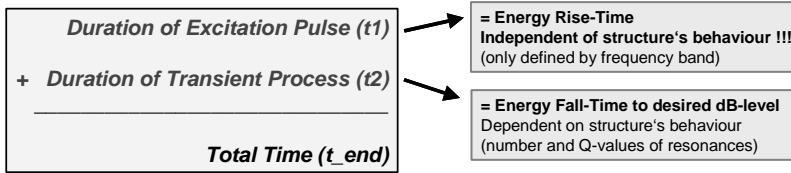
Let us have a closer look at the time pulses. The numerical effort and hence the total calculation time, is determined by the total number of timesteps to be calculated. It can be reduced in two ways: a) Increasing the timestep width Δt or b) reducing the time t_end .

a) Increasing Δt



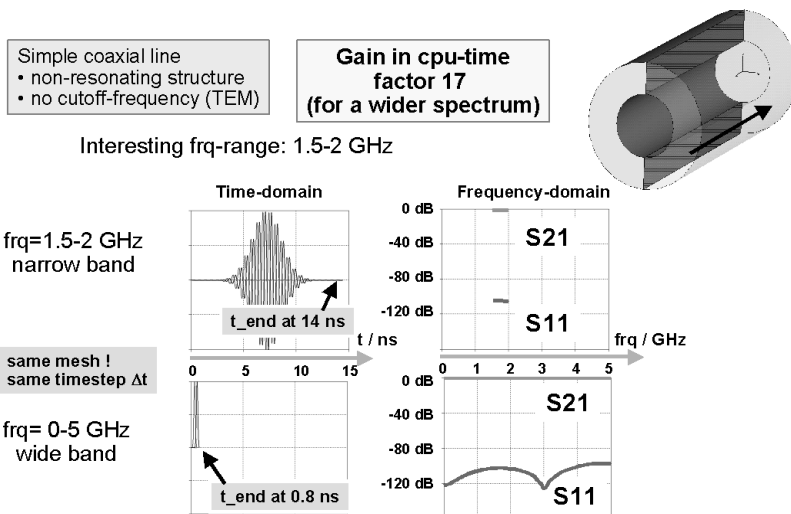
Since the time step Δt is proportional to the smallest mesh step, it can be increased by avoiding unnecessarily small grid steps (see also section 4.3). Here PBA® allows the discretization of thin wires or thin microstrip lines without using a mesh step for the wire radius or the conductor thickness. As can be seen in the picture above, a small ratio limit ensures that small mesh steps are forbidden.

b) Decreasing t_end



The total calculation time t_end is the sum of the duration of the excitation pulse t1 and the duration of the transient process t2. Depending on the structure's behavior it is useful to reduce either t1 or t2.

For *non-resonating structures* t_end can be shortened by reducing t1. This value is determined by the excitation pulse length and therefore directly influenced by the frequency range setting. The larger the bandwidth, the shorter the excitation time. In other words: for non-resonating structures you should always avoid a narrow frequency band. This is demonstrated in the next picture for a coaxial line. Just by defining a larger frequency band the calculation time can be significantly reduced:



Please note that for models without a cut-off frequency (e.g. TEM and microstrip structures), DC should always be included in the frequency range, since it shortens the excitation pulse by a factor of 2. For instance, the excitation time for a frequency range 0.01 ... 10 GHz is twice as long as for a range 0 ... 10 GHz.

Regarding the choice of an appropriate upper frequency limit, two points have to be considered: you should avoid a) the excitation of higher resonances and b) a large increase in the number of mesh points.

However, if t_{end} is determined by run-time effects, reducing t_1 is not very efficient. That means that if the propagating pulse arrives at the output long after the excitation signal has vanished, the reduction of t_1 does not influence the total time t_{end} significantly.

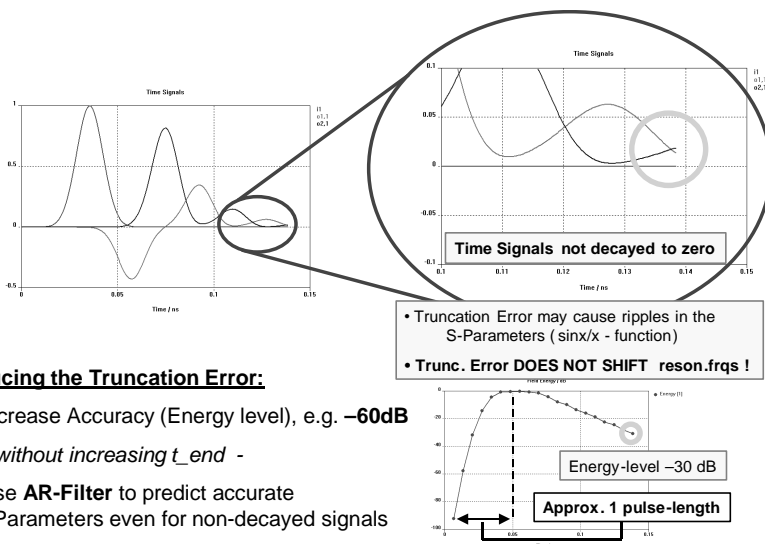
For *resonating structures* the reduction of t_1 has no remarkable effect either, since the total calculation time is dominated by the transient process.

However, if you are just interested in the resonance frequency of a system, one possibility to reduce the calculation time t_2 is to stop the simulation before the time signals have decayed to zero.

This can be achieved by reducing:

- 1) the accuracy in the *Solver Parameters* menu (Default setting: -30 dB. In this case the solver stops when the remaining energy in the calculation domain has decreased down to -30 dB compared to the maximum energy.)
- 2) the maximum number of pulses in *Solver Parameters* \Rightarrow *Specials ...* \Rightarrow *Steady State*. (Default setting: 20 pulse lengths. In this case the solver stops when the simulation time equals 20 lengths of the excitation pulse.)

If there is still a certain amount of energy left in the structure when stopping the solver, a so-called “truncation error” is made. It causes ripples in the S-Parameter curves, but does not shift the resonant frequency.



Reducing the Truncation Error:

- a) Increase Accuracy (Energy level), e.g. **-60dB**
or - without increasing t_{end} -
- b) Use **AR-Filter** to predict accurate S-Parameters even for non-decayed signals

In order to shorten the calculation time t_2 for resonant structures and in addition avoid the truncation error you can make use of the Autoregressive Filter (*Results* \Rightarrow *S-Parameter Calculations* \Rightarrow *AR Filter*). The necessary order of the filter increases with the number of resonances in the spectrum, therefore it may be helpful in some cases to reduce the bandwidth of the excitation pulse in order to have less resonance peaks. If

you have found appropriate settings for the AR Filter, it can be used online during the time domain calculation to stop the solver when it has found an answer with a sufficient accuracy (*Solver Parameters* ⇒ *Specials ...* ⇒ *AR Filter* ⇒ *Online AR Filter Analysis*). Please refer to the next section of this book to get a more detailed description of the AR Filter. Further information can be found in the online help

Some minor hints for increasing the simulation performance in the time domain are listed in the following:

- Avoid unnecessary field monitors (farfield monitors are less critical).
- Use open (PML) boundaries only for radiation problems.
- Do not include cut-off in your frequency range, if you are not interested in the results below cut-off. Choose the lower frequency limit 10% higher than the cut-off frequency of the waveguide.

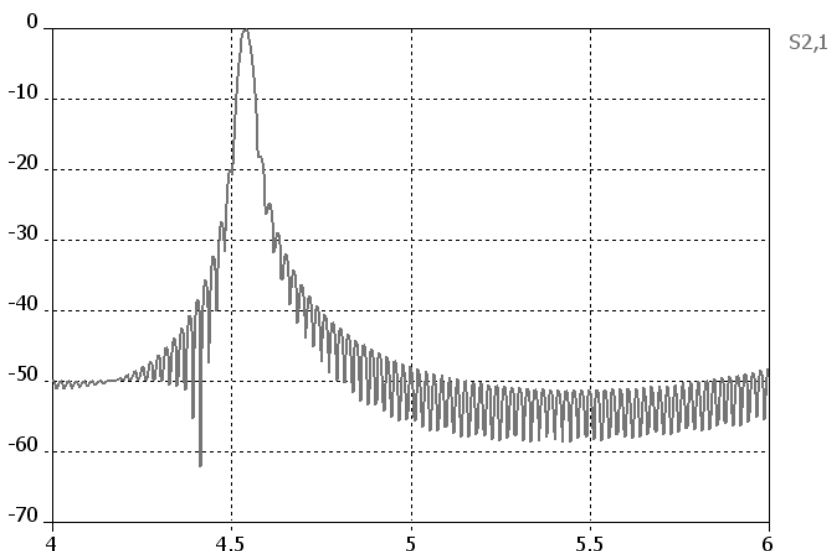
If you follow the hints and the recommendations given in the preceding sections, you will ensure that you achieve reliable and efficient simulation results. The efficiency can be further improved by making use of the extensive possibilities the integrated Visual Basic macro language offers. It is described in detail in the following two chapters.

4.7 Auto-Regressive Filtering (AR-Filter)

The performance of the transient solver can be significantly reduced when strongly resonating devices are simulated. The main reason for this is that the calculation of the S-parameters by using a Fourier Transform requires the time signals to have sufficiently decayed to zero, otherwise a so-called truncation error will occur.

Especially for highly resonating devices, a ringing in the time signals may occur such that the signals decay very slowly and therefore require long simulation times for accurate Fourier Transforms.

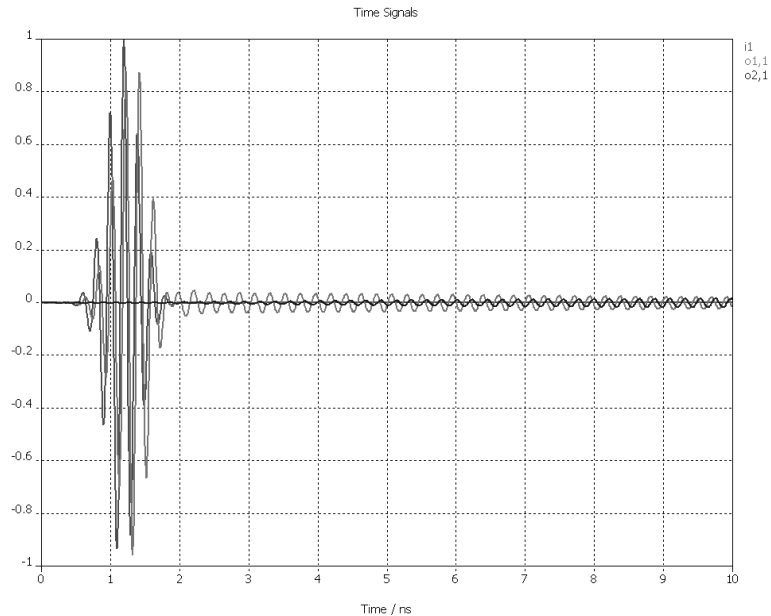
If a S-parameter calculation is performed by using time signals which still exhibit some oscillation, a ripple will appear on the S-parameters as shown in the picture below:



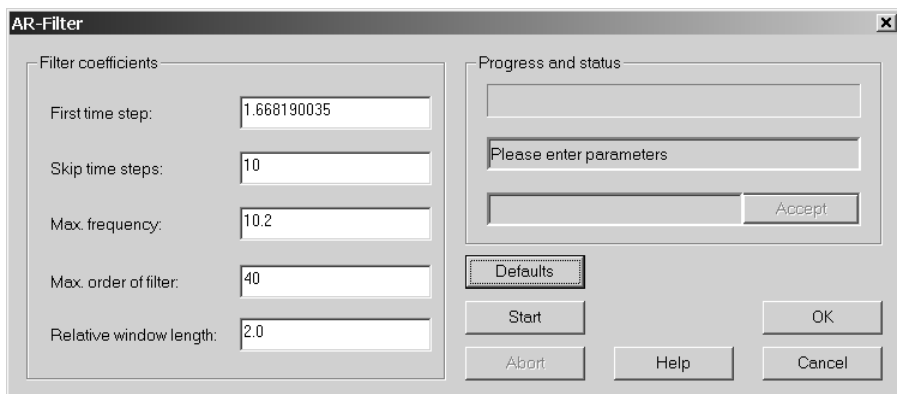
Please note that the curve's minimum or maximum locations (i.e. the position of the resonance frequencies) are not affected by the ripple, so if this is the only interesting information to be extracted from the simulation, the truncation error may be tolerated.

Otherwise, the usage of the AR-filter feature may be useful. The idea of this technique is to train an auto-regressive (AR) filter by using a short interval of the time signals. Afterwards, the AR-filter is used to predict the signal for the next time steps. Once the prediction and the actual simulation agree sufficiently well, the AR-filter contains all relevant information about the device. Therefore the simulation can be stopped and the S-parameters can be derived mathematically from the AR-filter representation.

The following picture shows a typical time signal for a strongly resonating structure:



It can be clearly seen that the time signals decay very slowly. Once such a simulation is completed (or aborted manually), you can enter the AR-filter dialog box by choosing *Results* ⇒ *Time Signal Calculations* ⇒ *AR-Filter for Port Signals*:

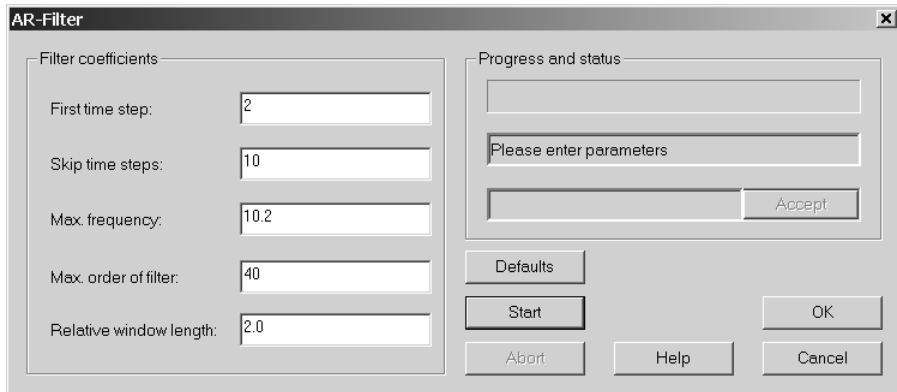


As a very first step, you should always press the *Defaults* button in order to let the dialog optimize its settings for the currently available time domain data.

As a next step, you should carefully inspect the signals and determine the time when the excitation pulse has vanished. In the example above, this may be at around 2 ns. Afterwards, you should enter this time in the *First time step* field in order to start the AR-filter analysis when the device's response is dominated by the resonant parts only.

Now you should check the order of the AR-filter. If the signal contains a few resonances only (e.g. you are expecting only a few poles inside the frequency band of interest), you

should start with an *Max. order of filter* of 40. After making these settings, the dialog box should look as follows:



Now you should press the *Start* button to perform the AR-filter analysis. Once the computation is finished, a dialog box will be shown containing some interesting information.

A header will provide some information about the AR-filter's settings and the currently investigated time signal:

```

=====
Analysis 1 of 2 for stimulation port 1, mode 1
=====

First time step      : 2.001828e+000 ns
Skip time steps     : 10
Max. frequency      : 1.020000e+001 GHz
Max. order of filter : 40
Relative Window length: 2.000000e+000

Input signal samples : 7913
Input signal length  : 5.617186e+001 ns

```

The AR-filter information is extracted from a window which is moved over the time signal. The following information displays the bounds of this window for each evaluation step and the energy error of the resulting filter approximation. This error should ideally be below 1e-3 in order to obtain a good accuracy.

step	window range ns	rel. wnd. length	filter order	energy error	pulses to calculate
1	1.7747e+000 - 4.6141e+000	2.50	32	4.83e-007	1.95e+000
2	1.8457e+000 - 4.6851e+000	2.50	33	4.80e-007	1.98e+000
3	1.9166e+000 - 4.7561e+000	2.50	30	1.00e-006	2.01e+000
4	1.9876e+000 - 4.8271e+000	2.50	30	1.14e-006	2.04e+000
5	1.4907e+000 - 4.8981e+000	3.00	36	8.60e-007	2.07e+000
6	1.5617e+000 - 4.9691e+000	3.00	37	1.38e-006	2.10e+000
7	1.6327e+000 - 5.0401e+000	3.00	35	2.68e-006	2.13e+000

```

-----
filter step 2 used for s-parameter calculation
-----

```

```

=====
Analysis 2 of 2 for stimulation port 1, mode 1
=====

```

```

First time step      : 2.001828e+000 ns
Skip time steps     : 10
Max. frequency      : 1.020000e+001 GHz
Max. order of filter: 40
Relative Window length: 2.000000e+000

```

```

Input signal samples : 7913
Input signal length  : 5.617186e+001 ns

```

step	window range ns	rel. wnd. length	filter order	energy error	pulses to calculate
1	6.3888e-001 - 4.6141e+000	3.50	1	1.00e+000	1.95e+000
2	2.4136e+000 - 4.6851e+000	2.00	31	6.54e-009	1.98e+000
3	2.4845e+000 - 4.7561e+000	2.00	33	1.49e-009	2.01e+000
4	2.5555e+000 - 4.8271e+000	2.00	32	7.95e-010	2.04e+000
5	2.6265e+000 - 4.8981e+000	2.00	32	1.91e-010	2.07e+000
6	2.6975e+000 - 4.9691e+000	2.00	31	1.23e-009	2.10e+000
7	2.7685e+000 - 5.0401e+000	2.00	31	4.99e-010	2.13e+000
8	2.8395e+000 - 5.1111e+000	2.00	31	1.53e-010	2.16e+000
9	2.9105e+000 - 5.1820e+000	2.00	30	4.50e-010	2.19e+000
10	2.4136e+000 - 5.2530e+000	2.50	37	3.76e-010	2.22e+000
11	2.4845e+000 - 5.3240e+000	2.50	37	4.13e-010	2.25e+000
12	2.5555e+000 - 5.3950e+000	2.50	35	1.61e-009	2.28e+000
13	2.6265e+000 - 5.4660e+000	2.50	37	4.10e-010	2.31e+000

```

filter step 8 used for s-parameter calculation

```

In this example, an excellent filter accuracy can be obtained for both time signals. As a next step, the S-parameters are calculated from the AR-filter's representation and the energy balance of the S-matrix is evaluated. The following lines show information about how well the energy conservation is maintained.

```

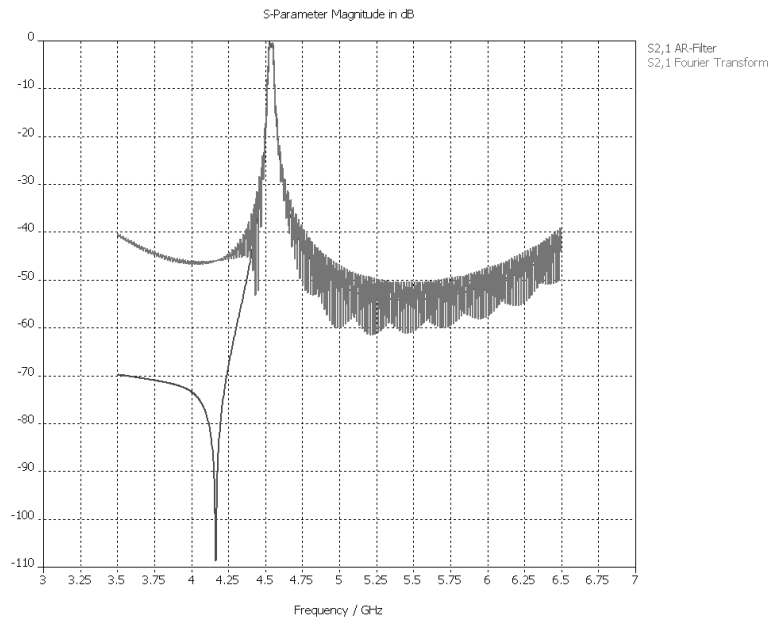
=====
Stimulation at port 1 (mode 1)
Average deviation in s-parameter energy balance: 6.094664e-004
Maximum deviation in s-parameter energy balance: 2.756003e-001
=====

=====
Maximum deviation in s-parameter energy balance = 0.2756
(regarding all successfull filter runs)
=====

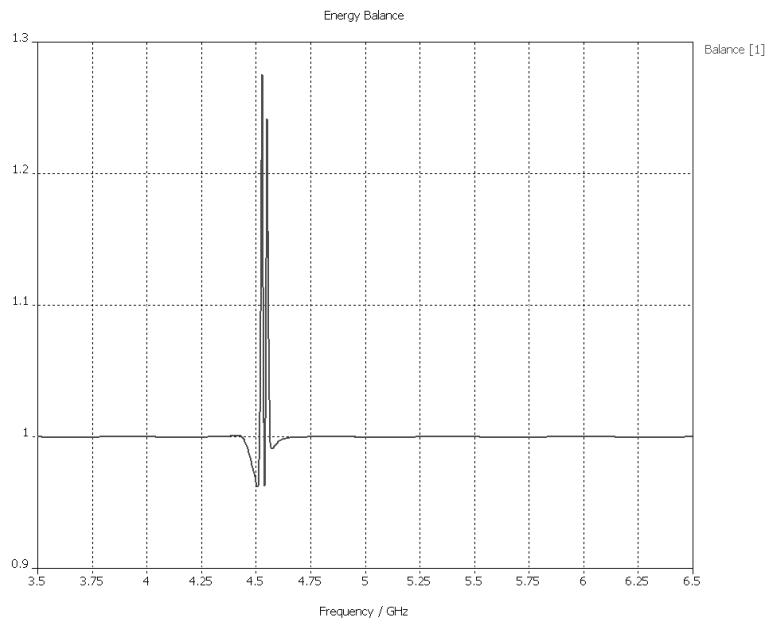
```

It can be seen from this data that the average energy error is sufficiently small, but the maximum error is much larger. This happens quite often if the S-parameters show some very sharp resonances and the locations of the poles and zeros in the various S-parameter curves do not match exactly. In most cases, this inaccuracy can simply be ignored.

The following picture shows an original S-parameter together with its AR-filter approximation (the curves may easily be overlaid by using the Copy / Paste feature of curves in the navigation tree):



Furthermore, the balance of the AR-filter S-parameters looks as follows:



The energy balance is quite close to one for the most part of the spectrum. Even if the balance is significantly disturbed close to the location of the peaks, this inaccuracy can hardly be seen in the S-parameter results.

Taking a closer look at the AR-filter output shown above, you can check that the AR-filter calculation uses time windows up to a maximum time of 5.466ns. This means that

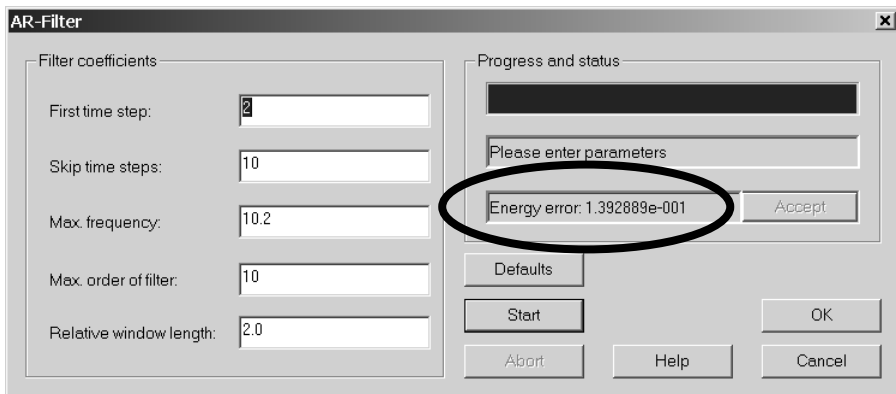
at least this maximum time needs to be simulated. In order to minimize the simulation time, you may consider decreasing the filter's order. This in turn would require shorter time windows. Every system, however, needs a certain minimum number of poles to be described properly. If the order is set lower than the actual order of the device's response, the accuracy will be degraded significantly.

Performing an AR-filter calculation by using a filter order of only 10 for this example would yield the following output:

```
=====
Stimulation at port 1 (mode 1)
Average deviation in s-parameter energy balance: 3.669134e-003
Maximum deviation in s-parameter energy balance: 1.463659e+000
=====

=====
Maximum deviation in s-parameter energy balance = 1.46366
(regarding all successful filter runs)
=====
```

Large deviations of the energy balance (e.g. significantly larger than 0.5) will usually indicate highly inaccurate results. Another criterion for the quality of the AR-filter approximation is the energy error shown in the dialog box after the AR-filter computation has finished:



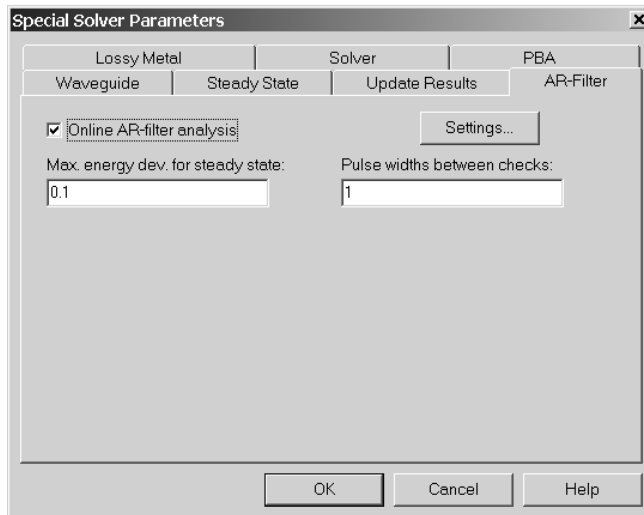
The energy error shown here should usually be lower than $1e-3$ for reliable results (it was $1.5e-10$ for a filter order of 40).

If the AR-filter yields inaccurate results and you can not improve its quality by increasing the *Max. order of filter* and the *First time step* settings, you should try to follow the some guidelines:

1. Avoid including DC (frequency = 0) in the simulation. The AR-filter approximation is most critical for the very low frequency range.
2. Use sufficiently small simulation bandwidths. Especially for narrow band filters, very small ranges of the spectrum will contain most of the significant device information.
3. It rarely makes sense to use filter orders of more than 100-150. It may be better to increase the *First time step* setting instead, or to reduce the simulation frequency range.

After the AR-filter has been successfully applied to a device in the post processing, it can be used as a so-called *Online AR-Filter* for subsequent calculations such as parameter sweeps or optimizations.

This feature can be activated from the *Specials* Dialog box (*AR-Filter* tab) from within the transient solver dialog box:



You can simply check the *Online AR-filter analysis* box and specify its parameters by pressing the *Settings* button. If an AR-filter calculation has been previously performed in the post processing before, its parameters will be retained here.

During the transient simulation, the AR-filter calculation will frequently be performed (*Pulse widths between checks*) and the balance of the resulting S-parameters will be checked. If the balance error turns out to be better than *Max. energy dev. for steady state*, the transient simulation will be automatically stopped.

We recommend that you refer to the online documentation for more information about this powerful feature or visit a specialized training class on this topic. Please contact your support center for more information.

Chapter 5 - The VBA Macro Language

5.1 Introduction

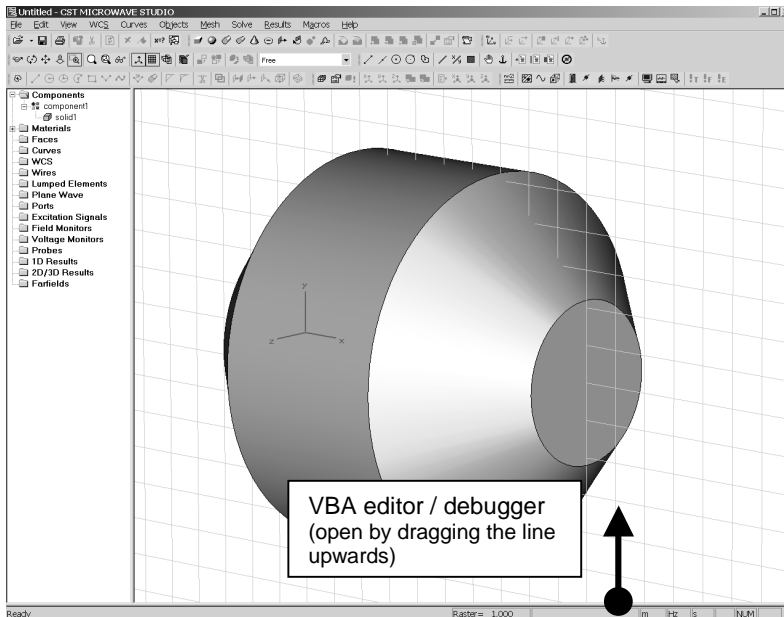
As an advanced user of CST MICROWAVE STUDIO® (CST MWS) you might from time to time come to a point where you identify a recurring task or just missing functionality.

CST MWS offers a powerful environment for automating any task within CST MWS or even in combination with external programs. The powerful VBA (Visual Basic for Applications) compatible macro language is equipped with a fully featured development environment including an editor and a debugger. An interface to OLE automation enables a seamless integration into the Windows environment (e.g. Microsoft Office®, MATLAB®, AutoCAD®, MathCAD®, Windows Scripting Host, etc.).

The following sections start with providing some general information on the VBA-based macro language before the actual integration into CST MWS is discussed. The explanations are supported by a variety of examples which should assist you in building your own macros. We strongly recommend that you work through this introduction which should only take a few hours to obtain a good working knowledge of macro programming in general.

This general introduction is followed by an overview of the result templates and macros which will automatically be pre-loaded during the software's installation procedure. The documentation for each of these scripts will contain some information about its functionality and how to use it efficiently. Many of these result templates or macros can be used immediately in order to assist you in your daily work.

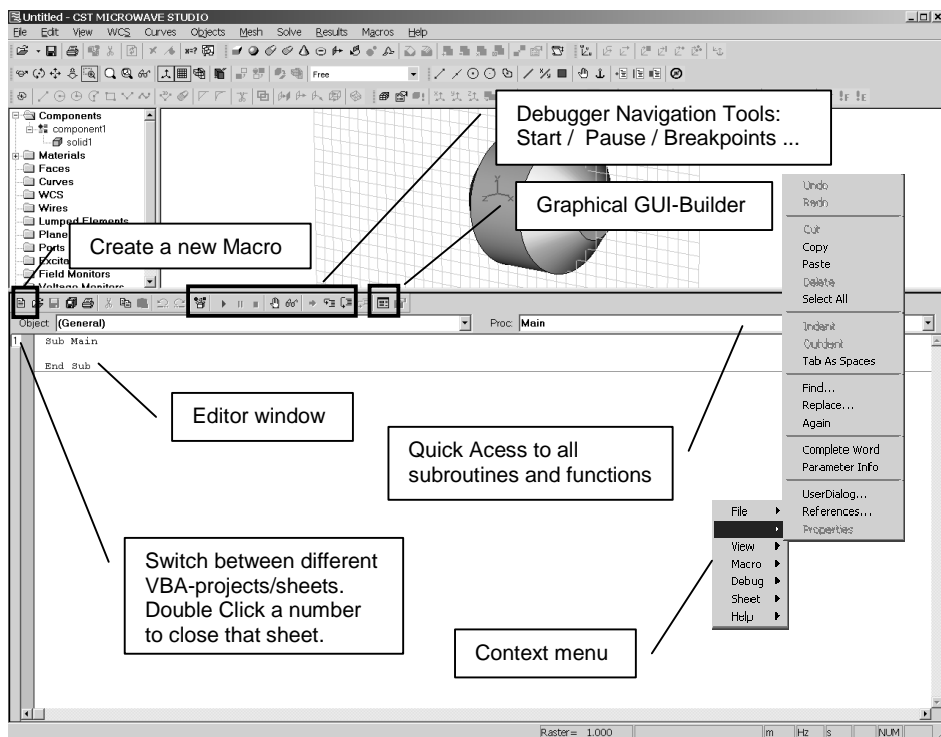
The VBA development environment can be accessed by dragging the window handle above the status bar upwards as shown in the picture below:



5.2 VBA Development Environment

The MWS user interface has a built-in VBA development environment consisting of an editor and a debugger. You should now have opened the editor window by dragging the corresponding handle as shown above.

The following picture shows some important controls which will be explained in the next sections:



In addition to its toolbar, the editor also features a context menu (opened by pressing the right mouse button) which contains more tools such as find, replace and many more.

VBA Help System

The VBA online help system can be easily accessed from within CST MWS by selecting *Help* ⇒ *VBA Macro Language* from the main menu.

The contents of this help system are divided into three different sections. The first section provides reference information on the basic VBA language elements. A second part contains specific information on the collection of CST MWS specific objects and a detailed explanation of their methods. Finally, the third section contains a selection of

examples which may either be used straight away or may give you some ideas for developing your own macros.

Pressing the F1-key within the editor window will provide some context specific help on the keyword which is positioned at the mouse cursor's location. If no keyword can be found, a general help page will be displayed allowing you to navigate through the help system.

VBA Editor Shortcuts

The following shortcuts are available within the VBA editor or debugger:

Ctrl+N	Open new file
Ctrl+O	Open existing file
Ctrl+S	Save current sheet
Ctrl+P	Print current sheet

Ctrl+Z	Undo
Ctrl+X	Cut
Ctrl+C	Copy
Ctrl+V	Paste
Del	Delete
Tab	Indent
Shift+Tab	Outdent
Ctrl+F	Find
Ctrl+R	Replace
F3	Find or replace again
Ctrl+Space	Complete word / command
Ctrl+I	Parameter info

F5	Run macro execution
Esc	Pause macro execution

F8	Debug step into
Shift+F8	Debug step over
Ctrl+F8	Debug step out
F7	Debug step to cursor
F9	Debug toggle breakpoint
Shift+Ctrl+F9	Clear all breakpoints
Shift+F9	Quick watch
Ctrl+F9	Add watch

5.3 VBA Language Elements

The following subsections will provide a rough overview of the typically used VBA language elements and thus should help you to get started with this programming language. The online documentation contains much more detailed information on these topics.


Subroutines and Functions

A VBA-script can simply be regarded as a collection of subroutines and functions. Each VBA program must have at least a subroutine called `Sub Main` which will automatically be called when the program is started. The following example illustrates how other functions can be called from within this main routine:


```
Option Explicit


Sub Main
    Dim s3 As String
    s3 = MultiStr ("Hello ",4)
    ' This is a comment (displayed in green colour)
    MsgBox s3
End Sub

Function MultiStr (s1 As String, n As Integer) As String
    Dim i As Integer
    Dim s2 As String
    s2 = ""
    For i = 1 To n
        s2 = s2 & s1 ← Breakpoint
    Next I
    MultiStr = s2
End Function
```

You should enter this simple example into the CST MWS VBA development environment while familiarizing yourself with the editor's functions. The example can then be started by pressing the corresponding toolbar button: . The program should then display a message box containing a single line of text: "Hello Hello Hello Hello".

You should now place the cursor over a keyword for which you want to see an explanation. Pressing the F1 key will then automatically invoke the online help system showing some detailed information about the particular keyword if available.

You can now set a breakpoint at the line "s2 = s2 & s1" by placing the cursor somewhere in this line and pressing the toolbar icon . The breakpoints are visualized by highlighting the corresponding line and showing a red dot at the beginning of the line.

Please start the macro execution again by pressing . The execution should then pause automatically at the breakpoint. Once the execution is paused, you can check the current values of a variable by just moving the mouse-pointer over the variable of interest. A tooltip will then appear informing you about the variable's content. You may also double-click on the variable (e.g. s2) and select the option *Debug* ⇒ *Add Watch* from the context menu (opened by pressing the right mouse button). The latter function

automatically places a so-called *variable watch* inside the “watch” frame on top of the editor window. This watch will then always display the current value of the variable making debugging very comfortable.

Even this small example demonstrates the ability of the interactive design environment for developing, testing and executing scripts. Before you are able to completely understand the script used in the example above, we need to introduce some other important language elements.

Variables, Data Types and Type Conversions

As in every programming language, VBA offers the ability to define variables as well. We strongly recommend declaring all variables before using them in order to prevent naming conflicts. The statement `Option Explicit` at the very beginning of the script (see the example above) forces the declaration of variables before they can be accessed by the program.

Each variable must belong to a certain data type with the most important ones being: Double, Integer, Long, String and Boolean.

Please note: VBA does also allow a very general data type called `Variant`. We do not usually recommend using this data type and will therefore skip it in this introduction.

Variables are declared using the `Dim` statement with the following syntax:

```
Dim s As String
Dim d As Double
Dim i As Integer
Dim j As Long
```

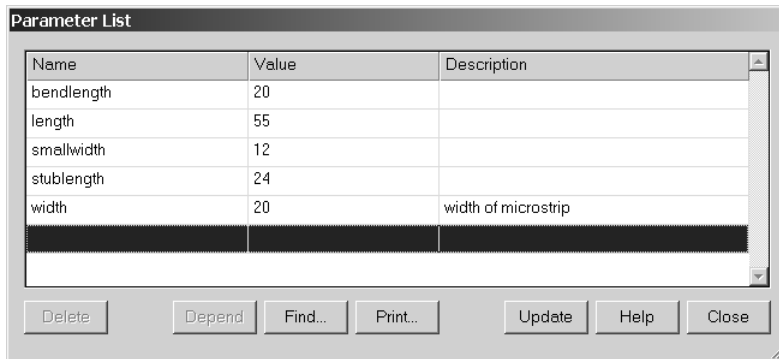
Instead of explicitly declaring the data type by using the `As` statement, you may also append a \$ sign to the variable name to indicate that it is of a string type. The same is valid for integer (%), long (&) and double (#) data types.

Thus the following declarations are equivalent, although we recommend using the first one:

```
Dim s As String           Dim s$
Dim d As Double          Dim d#
Dim i As Integer         Dim i%
Dim j As Long            Dim j&
```

Please note: The integer data type is only 16 bits wide and thus can only contain values within the range of $-32768, \dots, 32767$. Please use the long data type for storing 32 bit integer values.

All CST MWS parameters which are currently defined in the project via *Edit⇒Parameters* can automatically be accessed as `Double` variables from within the script. Problems may occur if the same names were accidentally used in a VBA script for local variables. To prevent these naming conflicts, we strongly recommend always using the statement `Option Explicit`. Following this rule, you would automatically get an error message in the event of multiply defined variables.



You can also change the CST MWS parameters from within the script by using the `StoreParameter` command as shown in the following code segment:

```
StoreParameter("width", 30)
```

VBA is quite sensitive to data types. Therefore type-conversions are sometimes necessary, e.g. when printing out a double value as a string by using the `MsgBox` function:

```
Dim d1 As Double
d1 = 1.23
MsgBox CStr(d1)
```

Here the function `CStr()` is used to convert the given argument into a string data type (Convert into String). Another common conversion function is `Cdbl()`, which can be used to convert an arbitrary data type into a double, if possible. If the conversion fails, an error will occur. A much more general way to evaluate an expression and to convert the result to a double data type is to use the `Evaluate()` function:

```
Dim d as Double
d = Cdbl("5")           OK
d = Cdbl("5+3")        ERROR
d = Evaluate("5+3")    OK
```

For further information, e.g. on how to declare array elements or to use constants, please refer to the Online Help.

Applications, Objects and their Methods

One of the most powerful parts of the VBA language is certainly the object-oriented approach, which allows the seamless combination and integration of component objects from various applications, e.g. MS Office® (Word®, PowerPoint®, Excel®, Outlook®), Matlab®, CST MWS. A single script may thus access functionality from several applications at the same time.

External objects need to be created by using the `CreateObject()` command before they can be accessed from within the script:

```
Dim word as object, ppt as object
Set word = CreateObject("Word.Application")
Set ppt = CreateObject("PowerPoint.Application")
```

After creating an external COM object, you are able to access its methods directly from within your VBA script. For more information on its VBA interface, please refer to the documentation for that particular application.

Since CST MWS also features a COM interface, the corresponding objects can also be accessed from third party applications as well. The following code segment shows how to start an control CST MWS from an external program by using VBA:

```
Sub Main
    Dim mws As Object
    Set mws = CreateObject("MWStudio.Application")

    mws.OpenFile "c:\temp\test.mod"

    Dim solver As Object
    Set solver = CreateObject("MWStudio.Solver")
    solver.Start

    mws.Save
    mws.Quit
    Set mws = Nothing
    Set solver = Nothing
End Sub
```

Please refer to the online documentation for more information on the particular methods available for the COM objects provided by CST MWS.

The access to the object's methods can be simplified by using a With – End With block as shown in the example below:

```
Sub Main
    With CreateObject("MWStudio.Application")

        .OpenFile "c:\temp\test.mod"

        With CreateObject("MWStudio.Solver")
            .Start
        End With

        .Save
        .Quit
    End With
End Sub
```

Please note that in the case of nested With – End With structures, the innermost block is always active.

The VBA interpreter of CST MWS allows access to its objects directly without using the CreateObject command. The methods of the application object are available as global functions and all other objects can be used as global objects. Therefore to use it within the CST MWS environment, the example shown above can be simplified to:

```
Sub Main
    OpenFile "c:\temp\test.mod"
```

```
Solver.Start  
Save  
End Sub
```

Flow Control

The VBA language offers a large variety of flow control elements. The most frequently used ones are the If – Then – Else – End If, While – Wend or For – Next statements shown in the examples below:

```
If Boolean_Expression Then  
    ' IF - STATEMENT  
Else  
    ' Else-STATEMENT  
End If  
  
While Boolean_Expression  
    STATEMENTS  
Wend  
  
For i = 1 To n  
    STATEMENTS  
Next i
```

Please find more information on these and some other flow control element in the online help which can be easily accessed by selecting *Help⇒VBA Macro Language* and choosing the topic *Language Reference Overview*.

File Operations

Accessing of files from a VBA script is rather straight forward. After opening a file for a particular purpose (reading or writing), a numbered stream can be assigned to it, which can be used to access the file. This is shown in the example below:


```
Dim sline as String  
  
Open "mydata.txt" For Input As #1  
    While Not EOF(1)  
        Line Input #1,sline  
        Debug.Print sline  
    Wend  
Close #1  
  
Open "mydata.txt" For Output As #2  
    Print #2, "Test Output"  
Close #2  
  
Open "mydata.txt" For Append As #3  
    Print #3, "Test Output 2"  
Close #3
```

Please note: The Debug.Print method is used to generate debug text output which is written to an output window in the development environment.

Any CST MWS result which is displayed in the navigation tree's 1D Results folder can be accessed more conveniently by using the `Result1D` object than by using direct file access. A similar `Result3D` object exists for the access of 3D result field data. The usage of these objects will be introduced later in this document.

Please find further information on file handling (e.g. checking directory contents with the `Dir$` command) in the `File Group` of the language reference part of the VBA online help system.

Graphical User Interface Builder

You can easily create and customize your own dialog boxes by using the graphical user interface builder which can be opened either by pressing the icon  or by selecting *Edit⇒UserDialog* from the context menu.

A customized dialog box can then be created by placing user interface components on the dialog sheet. However, describing all the details of creating user defined dialog boxes is beyond the scope of this manual. Please refer to the online help for more information.

Please note that most of the dialog's controls store their data as strings, so a conversion of strings to other data types may be necessary. As already shown earlier, this can be easily achieved in the common case of a string to double conversion by using the `CDbl()` or `Evaluate()` functions.

Another important source of information on this topic are the pre-loaded result templates or macros which will be explained in more detail later in this book. Most of these scripts contain user defined dialog boxes with a variety of controls.

Mathematical Functions, Operators and Constants

The VBA language offers various mathematical functions and operators. Once you are familiar with another programming language you may already know most of them.

However, there are a few functions which have a slightly different meaning when compared to other programming languages such as C or C++:

<code>Sqr(a)</code>	Square root of a
<code>a^2</code>	<code>a * a</code>
<code>Log(a)</code>	Natural logarithm of a
<code>Log(a) / Log(10)</code>	Logarithm of a to a base of 10

Please refer to the help on operators and general mathematics in the online help system.

In addition to the VBA standard language elements, CST MWS provides a collection of mathematical functions which are frequently required for scientific computing:

Pi	3.141592654....
<code>re(am,phD) / im(am,phD)</code>	Real or imaginary part of a complex number (am = amplitude, phD = phase in degrees)
<code>ASin(), ACos()</code>	Arc sin and Arc cos in radian

SinD(x), CosD(x), TanD(x)	Sin, cos, tan with x being the angle in degrees
ASinD(), ACosD(), ATnD()	Arc sin / Arc cos / Arc tan with the result being an angle in degrees
ATn2(y,x)	Polar angle of the complex number x + iy measured in radian
ATn2D(y,x)	Polar angle of the complex number x + iy measured in degrees
Evaluate(x)	Evaluates the mathematical expression x and returns its numerical value (as double)

Please note: All mathematical functions can also be used in dialog box entry fields whenever CST MWS allows you to enter a mathematical expression.

5.4 Concepts of Macro Programming in CST MWS

After introducing the fundamental concepts of VBA programming the next sections will focus on how the macro programming is integrated in CST MWS.

A couple of tools within CST MWS can be extended by providing so-called user defined functions:

- Project templates for the customization of the settings for particular structures
- Result templates for the automated extraction of simulation results
- User defined excitation functions for transient analysis (access from within the *Excitation signal* dialog box)
- User defined optimizer goal function (accessed from the *Optimization* dialog box)
- User defined parameter sweep watch function (access from within the *Parameter sweep* dialog box)

You can also create macros which can help to automate common tasks. Each of these macros can be assigned to either one of the following groups:

- Structure modeling macros: These macros need to be stored in the history list in order to ensure a proper structure update as a consequence of parametric changes.
- Control Macros: These macros do not need to be stored in the history list and are thus usually employed for post processing calculations.

The user defined functions are always stored as part of a certain project. The text files which contain the functions' scripts have the following file extensions:

Type of function	File name extension
User defined excitation function	usf
User defined goal function	gfc
User defined watch function	pfc

Therefore, if the project's name is "test.mod", the file name of a user defined goal function will be "test.gfc".

The storage of the structure or control macros can either be local (within the project) or global (available for all projects). The project templates and the result templates are always stored globally.

Local Project Macros

The local macros are stored as part of a particular project and obtain a numbered file extension, e.g. 000, 001, 002, etc.

Assuming the name of the project to be “test.mod”, the file names of the local macros could be any one of the following: “test.000”, “test.001”, etc.

Global Macros and *Global Macro Path*

All global macros are stored in a common folder to which we will refer as the *Global Macro Path* in the following. By default, its location is the “Macros” folder in the CST MWS installation directory.

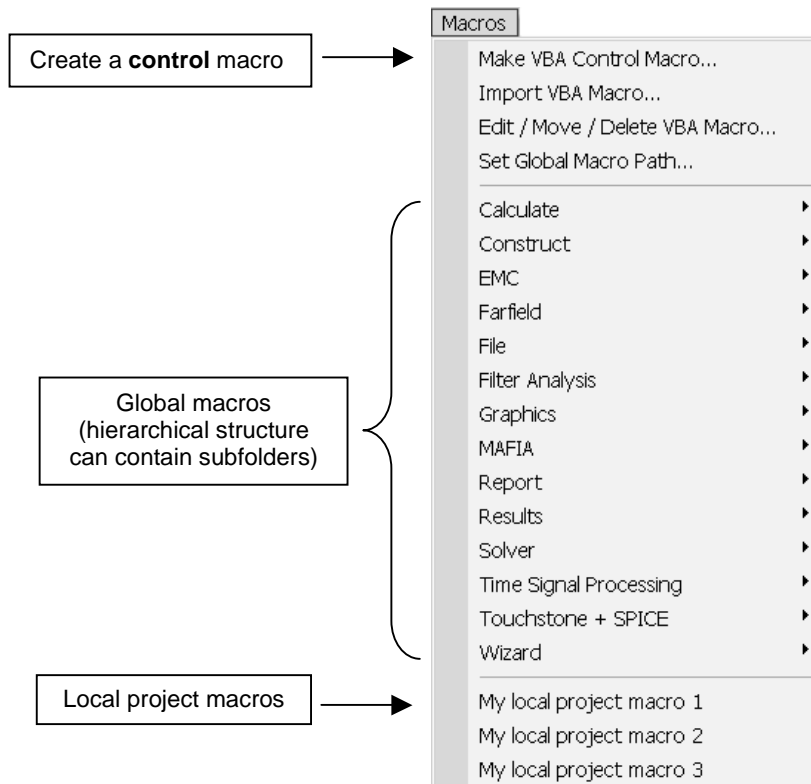
However, when working in a group of users you may change the location of this directory to a folder which is accessible to everyone. Each user should then specify its location from within the *Macros* ⇒ *Set Global Macro Path* dialog box.

The file names of global macros are always built by the keyword “macro” and a numbered extension, e.g. “macro.000”, “macro.001”, etc. It is not necessary to ensure a consecutive numbering of the macros.

By default, each installation of CST MWS preloads a collection of useful global macros. The file extension of these macros will start at 500 in order to avoid naming conflicts with your own global macros.

Furthermore, the global macro path contains two folders, the *New project templates* and the *Result templates*, where the corresponding scripts are stored.

The following picture shows the contents of the *Macros* submenu after a couple of macros has been created:



5.5 Control Macros

A certain macro belongs to the class of control macros if it does not need to be stored in the history list. The following list provides an overview of frequently used control macros:

- Any kind of numerical post processing of S-parameters, farfields, probes, voltages or electromagnetic fields.
- Export the screen's contents as pictures or videos in various formats.
- Project management, e.g. archiving models, cleaning up temporary working directories, etc.
- Control and execution of computational sequences.
- Comparison of result data from several projects or measurements.

The following sections will introduce the most important building blocks used by control macros. These code segments are frequently used in the rather complex pre-loaded result templates or macros which will be explained in more detail in the next chapters.

Please note: A control macro always has a "*" sign in front of its name, indicating that it is a control macro rather than a structure macro.

Result1D Object

The Result1D object offers convenient access to one dimensional result data. The following list gives an overview of which result types can be accessed by using this object (please refer to the online help for details of the file naming conventions):

1. Result data stored as part of the project with the file extension “sig”, e.g. time signals, S-parameters, etc.
2. Voltage monitor data
3. Probe results
4. VSWR results, Z+Y-matrices

The Result1D object allows the corresponding file to be opened and its contents to be accessed via methods in an object-oriented manner.

Many methods of the Result1D objects deliver the index of a particular element rather than its value directly (e.g. `.GetClosestIndexFromX`, `.GetGlobalMaximum`, `.GetFirstMinimum`). You can then access the numerical X/Y value of the indexed data point by using the object's functions `.GetX`, and `.GetY`.

The following examples demonstrate the usage of the Result1D object for various tasks. Please refer to the VBA online documentation for more information.

Example 1: Access S11 phase results which are stored in the file <project name>^p1(1)1(1).sig

```
Dim n As Long, i As Long
Dim vx As Double, vy As Double

With Result1D ("p1(1)1(1)") ' Open the phase of S11
    n = .GetN                ' Get number of frequency samples
    For i = 0 To n-1
        ' Read all points with zero based
        ' indices.
        vx = .GetX(i)        ' get frequency of data point
        vy = .GetY(i)        ' get phase of data point
    Next i
End With
```

In order to create a new data set containing the group delay time as a function of frequency (which can be calculated from the S-parameter's phase data), you would need to implement the following steps:

1. Create and open a new file
2. Loop over all frequency samples, calculate each frequency's delay time and write the result to file by using the `print` statement.
3. Close the result file

Example 2: Determine the maximum return loss and the magnitude of S11 at a frequency of 33 GHz (both in dB).

```
Dim maxr As Double, s11_33GHz As Double
Dim s11 As Object

Set s11 = Result1D("d1(1)1(1)") ' S11 in dB
maxr = s11.GetY(s11.GetGlobalMaximum())
s11_33GHz = s11.GetY(s11.GetClosestIndexFromX(33))
```

Example 3: Determine the frequency of the VSWR's global minimum

```
Solver.CalculateVSWR ' calculate the VSWR and
                    ' and store it in
                    ' <project name>^vswr1(1).sig

Dim fmin As Double
With Result1D("vswr1(1)")
    Fmin = .GetX(.GetGlobalMinimum())
        ' please note that this frequency is returned in
        ' the currently active frequency unit, e.g. GHz
        ' to obtain its value in Hz, you need to
        ' multiply it by Units.GetFrequencyUnitToSI
End With
```

Example 4: Calculate the real part of S11 at a certain frequency (here 0.65 GHz)

```
Dim a11 As Object
Dim p11 As Object
Set a11 = Result1D ("a1(1)1(1)")
Set p11 = Result1D ("p1(1)1(1)")

Dim n As Integer
Dim frq As Double, phase As Double, ampli As Double
Dim real As Double

frq=0.65
n=a11.GetClosestIndexFromX(frq)

phase = Pi/180.0 * p11.GetY(n)
ampli = a11.GetY(n)
real = ampli * cos(phase)
```

Example 5: Minimize the real part of S11 at a certain frequency by a user defined goal function. Please note that the function's body will automatically be provided by the optimizer once you choose to define a user defined goal function.

```
' userdefined goal function for optimizer
Option Explicit
Function GoalFunction() As Double
    ' code segment from example 4

    Dim a11 As Object
    Dim p11 As Object
    Set a11 = Result1D ("a1(1)1(1)")
    Set p11 = Result1D ("p1(1)1(1)")

    Dim n As Integer
    Dim frq As Double, phase As Double, ampli As Double
    Dim real As Double

    frq=0.65
    n=a11.GetClosestIndexFromX(frq)

    phase = Pi/180.0 * p11.GetY(n)
    ampli = a11.GetY(n)
    real = ampli * cos(phase)

    ' assign the result to the goal function

    GoalFunction = real

End Function
```

You may also access files which have been created by a VBA script using the Result1D object. However, these files must contain two columned X/Y data and the file name must have the following format: "<project name>^<result name>.sig" with <result name> being a freely chosen identifier for the result data.

In order to access external data, you may use a code segment similar to the following:

```
' just copying the file in order to access it by using the
' Result1D object

FileCopy "C:\TEMP\mydata.txt", GetProjectBaseName+"^mydata.sig"

With Result1D("mydata")
    .
    .
    .
End With
```

Result3D Object

The Result3D object offers a convenient access to 3D result file data, very similar to the previously discussed Result1D object. In particular, the Result3D object allows the loading and saving of binary 3D field data in addition to some basic mathematical operations such as add, subtract, multiply, etc.

The following example demonstrates how to load 3D result field data in order to calculate a superposition of fields which is then saved again as binary result field data.

Example: Calculate a superposition of 3D result fields

```
Dim field1 as Object, field2 as Object

' Load the 3D result data fields "e1" and "e2"
' (e.g. modes calculated by the eigenmode solver)

Set field1 = Result3D("^mode_e_1")
Set field2 = Result3D("^mode_e_2")

' First scale field1 and field2 by scalar factors

field1.ScalarMult 2
field2.ScalarMult 1.5

' Now add field2 to field1 (the latter will be overridden)

field1.Add field2

' Finally save field1 as a new result data field

field1.Save "^combination"
```

Please refer to the Online Help for more explanations about the methods available for this powerful object.

Adding Data Items to the Navigation Tree

CST MWS offers the possibility of adding two columned X/Y ASCII data files to the result tree which can then be displayed in the same way as other one dimensional data sets. The following exemplary code segment illustrates how this can be achieved for an external data file:

```
With ResultTree
    .Reset
    .Type "XYSignal"
    .Subtype "Linear"
    .Title "TDR: Time Dependent Impedance Z(t) in Ohms"
    .Xlabel "Time / " + Units.GetTimeUnit()
    .Ylabel "Z(t) "
```

```

        .Name "1D Results\TDR Results\Z(t) "
        .File "C:\Data\myfile.txt"
        .Add
    End With

```

In order to compare curves by plotting them together on the same graph, both curves must have identical titles, x-labels and y-labels. Otherwise CST MWS will display the message “Incompatible result data” when both curves are selected.

In the same way, 3D field result data can be added to the navigation tree:

```

With ResultTree
    .Reset
    .Type "Efield3D"
    .Name "3D Results\E-Field\My E-field"
    .File "C:\Data\myfile.m3d"
    .Add
End With

```

Traverse Folders and Select Items From the Navigation Tree

A particular item in the navigation tree can be easily selected by using the `SelectTreeItem` function. This function returns a Boolean value indicating whether the specified item existed and therefore could be selected.

The name of a tree item is specified by its sequence of folders starting at the tree's root with the folder names being separated by “\” characters, e.g.:

```
SelectTreeItem "1D Results\TDR Results\My Result"
```

You are able to traverse the tree starting from a particular item (e.g. the root item) by using a script similar to the following code segment:

```

Dim sItem As String
sItem = ResultTree.GetFirstChildName("Farfields")

While sItem <> ""
    SelectTreeItem sItem

    .
    .
    .

    sItem = ResultTree.GetNextItemName(sItem)
Wend

```

Please take care of the fact that the navigation tree's contents are mode dependent (e.g. results are not accessible in the mesh mode). In order to ensure that the program is currently not in mesh mode, you can use the following statement:

```

' ensure that the mesh mode has been switched off
Mesh.ViewMeshMode False

```


Access the Currently Displayed Plot Data

A couple of commands can be used to access the data of the currently active plot. Therefore, you should first select a particular plot by using the `SelectTreeItem` method before using one of the following commands:

- `GetFieldVector`
- `GetFieldFrequency`
- `GetFieldPlotMaximum`
- `GetFieldPlotMinimum`

Please refer to the online documentation for more information on the usage of these methods.

Example 1: Select the fundamental three dimensional eigenmode and determine its resonance frequency and the field vector at a particular position. Finally, display a message box informing about the maximum value of the field.

```
' select the corresponding navigation tree item to display
' the mode's electric field
SelectTreeItem "2D/3D Results\Modes\Mode 1\e"

' determine the mode's frequency
dFrequency = GetFieldFrequency()

Dim xp As Double, yp As Double, zp As Double
xp = 0.0
yp = 0.0
zp = 0.0

' get the field vector at a particular location (xp, yp, zp)
' the real and imaginary parts of the vector's components
' are stored in the variables: vxre, vyre, vzre (real parts)
' and vxim, vyim, vzim (imaginary parts)
Dim vxre As Double, vyre As Double, vzre As Double
Dim vxim As Double, vyim As Double, vzim As Double
GetFieldVector(xp, yp, zp, vxre, vyre, vzre, vxim, vyim, vzim )

' display the field's maximum strength
MsgBox "Max=" + CStr(GetFieldPlotMaximum)
```

Example 2: Plot a table's contents and export the data to an ASCII file

```
SelectTreeItem "Tables\s11"
ExportPlotData "test^mys11.sig"
```

Access Farfield Data From a VBA Script

Direct access to the farfield at a particular location can easily be obtained by using the `CalculatePoint` method of the `FarfieldPlot` object as shown in the example below:

```
FarfieldPlot.SetPlotMode "Directivity"
FarfieldPlot.CalculatePoint (dTheta, 0, "Abs", "farfield [1]")
```

The third argument of the .CalculatePoint method can be any member of the following list:

```
"Abs"/"Theta"/"Phi"/"Axial Ratio"/"Left Polarisation"/"Right
Polarisation"/"Th_Re"/"Th_Im"/"Ph_Re"/"Ph_Im"/"Th_Phase"/"Ph_P
hase"
```

Please note that the result obtained by the .CalculatePoint method depends on the currently selected plot mode. Please make sure that the plot mode is adjusted to the desired one (directivity / gain / efield / hfield / pfield / rcs / rcsw / scale2max) by using the .SetPlotMode method before accessing the farfield's data.

In addition to the direct access to individual farfield data, you can also create a one dimensional farfield plot and then export its data:

```
SelectTreeItem("Farfield\farfield [1]")

With FarfieldPlot
    .SetPlotMode "directivity"
    .Vary "theta"
    .ThetaStep 25
    .PlotType "3d"
    .Plot
End With

ExportPlotData "farfield.sig"
```

The following example demonstrates how to create a X/Y data file containing the vertical diagram by using the .CalculatePoint method rather than exporting plot data:

```
Dim dTheta As Double, dValue As Double

Open "C:\Data\mydata.txt" For Output As #1

For dTheta = 0 To 180 STEP 5

    dValue = FarfieldPlot.CalculatePoint( _
        dTheta, 0, "Abs", "farfield [1]")

    Print #1, CStr(dTheta) + " " + CStr(dValue)

Next dTheta

Close #1
```

5.6 Structure Macros

The main difference between structure macros and the previously introduced command macros is that structure macros are stored in the history list after their execution. Whenever a macro becomes part of the structure's construction, it is important that history updates also recall the macro execution in order to fully regenerate the structure.

Structure macros may be used for various purposes at different complexity levels. However, this document can only cover some fundamental concepts on structure macro development. If you frequently need to develop such macros, it is advisable to attend a training class specialized on this topic. Please contact your support center for more information.

Creating a Structure Macro or a New Project Template

You might identify that you frequently need to change some default settings for your particular kind of application. The following list gives a couple of examples for such default setting changes:

- Set the units to “mil” and “MHz” rather than use “mm” and “GHz”.
- Define a certain set of material properties
- Set the frequency range to a default of e.g. 0 - 3 GHz.
- Define farfield monitors for frequencies of 0.9 GHz and 1.8 GHz.


The most convenient way to achieve such changes to the default settings, dependent on the structure type, is to define project templates for these devices. Such a template is simply a particular kind of structure macro and can be created as such.

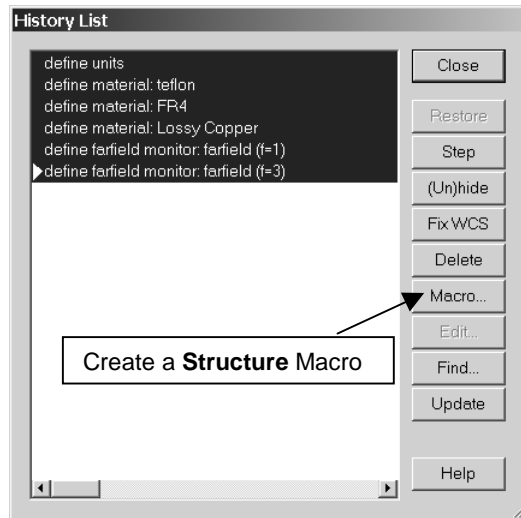
You should start creating your template with an empty history list. Please select *File* ⇒ *New* in order to start a new project.

Afterwards, you should perform all the frequently used steps which should later be automatically provided by the macro such as:

1. Define the units
2. Define default materials (e.g. a set of standard dielectric materials, etc.)
3. Define the background material
4. Define a default frequency range
5. Define a collection of frequently used monitors
6. Change some default settings for the mesh generation
7. Change some solver settings

This list can only provide some ideas on what can be customized by the template. You may also include other operations as well.

Once all necessary steps have been completed, you should open the history list by selecting *Edit* ⇒ *History List* or just pressing the corresponding toolbar icon . The history list will then appear as in the following picture:



You should then select all commands which should be included in the template and press the *Macro* button. Afterwards you will be prompted to assign a name to the macro (e.g. “My Defaults”) and press *Ok*.

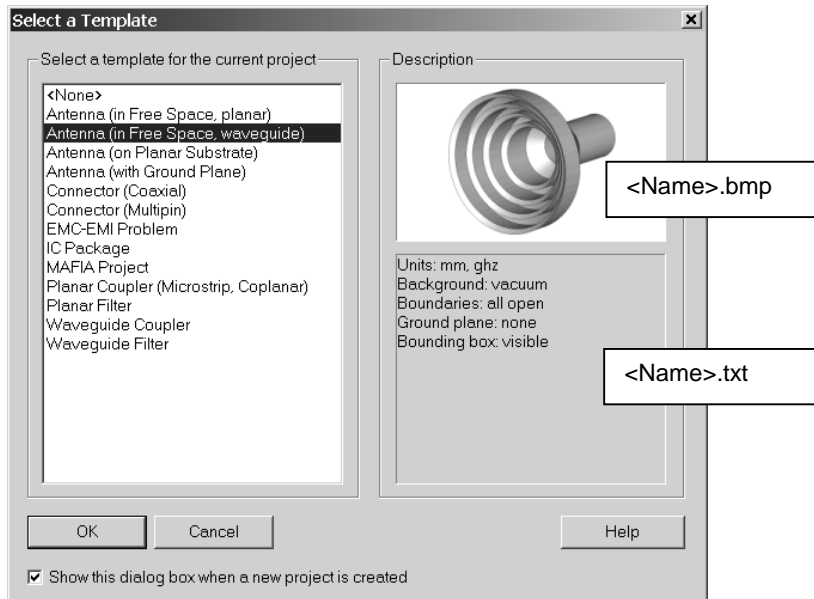
The VBA editor will then automatically be opened showing the contents of the new structure macro. The first line in the macro contains a comment line with the name of the macro.

You can now just call this project macro by selecting its name from the *Macros* menu. Once a structure macro is executed, it will be appended to the history list.

You can create global structure macros in the same way by just activating the *Make global macro* option in the dialog box where you also specify the name for the macro.

In order to create a project template, you need to select the contents of the macro between the *Sub main* and *End Sub* statements and copy / paste it into an ASCII text editor. You should then save this file in the *New Project Templates* folder in the *Global Macro Path*. Please ensure that the name of the file allows the proper identification of the macro’s purpose and that the file extension is set to “*tpl*”, e.g. “UMTS base station filters.*tpl*”.

In addition to the macro code you can also provide a text file with the same name but the extension “*txt*” rather than “*tpl*” describing the macros operation. This text file will automatically be shown when the template is selected in the *Create a New Project* dialog box. Similarly, a bitmap file (150 x 129 pixel, 256 colors) can also be specified.



Difficulties in Creating Structure Macros

The creation of structure macros by selecting a couple of lines in the history list and then automatically converting this into a VBA code sequence is very convenient.

However, it is important to note that just taking a few lines out of a construction history and building a macro from this will not usually work due to the following reasons:

1. The construction may depend on earlier operations in the history which are not part of the newly created macro. Therefore pick operations may fail or the structure elements for deletions or transformations simply do not exist. In order to avoid these problems we always recommend exporting a command sequence from the very beginning of the history list.
2. Naming conflicts may occur if the macro uses shape names which have already been defined elsewhere. This problem can only be solved by adding some additional coding in regard to the shape naming in the macro.

Because of these pitfalls, we do not recommend to use the macro functionality for simply re-using parts of a model's history to build a new structure.

We strongly suggest exporting the corresponding parts of the existing model as an SAT file first. In a second step you may import this file into the new project by using the working coordinate system for proper placement of the imported shapes. The import feature will automatically take care of possible naming conflicts and rename the imported shapes in order to avoid such problems.

The imported structure can then be modified to a certain extent by using the *Local Modification* commands. Please refer to the *Getting Started* manual for more information on this topic.

Using Dialog Boxes in Structure Macros

The creation of parametric structure macros usually need some user defined dialog boxes for defining a couple of structural parameters. However, these dialog boxes should only appear when the macro is called from the menu and should not be shown when the history list is processed.

This behavior can be achieved by using the VBA statements `BeginHide` and `EndHide` in the structure macro. All VBA code which is placed between these two lines will automatically be removed from the macro's code before it is written into the history list.

However, the dialog box code should usually provide some parameters which will later be used in the macro. Therefore, you need to declare these parameters before the `BeginHide` statement. The keyword `Assign` should then be placed right before the `EndHide` command for each parameter in order to define its current value which will then be stored as such in the history list. The following code segment shows the usage of these commands:

```
Dim dRadius As Double

BeginHide

    ' some code defining dRadius by using user dialog boxes
    .
    .
    .

    Assign("dRadius")
EndHide
```

After running this code segment, the history list item will contain the following VBA code only:


```
Dim dRadius As Double

dRadius = 0.23    ' or whatever has been assigned to it
                  ' by the user
```

Please note that using dialog boxes in structure macros is truly a highly advanced functionality.

Common Pitfalls in the Usage of Structure Macros

The following list summarizes a couple of possible pitfalls in the usage of structure macros:

1. The first comment line of a structure macro must not contain an asterisk * character preceding the macro's name (otherwise it would be considered as a command macro and would therefore not be stored in the history list).
2. A structure macro is only appended to the history list if it is called from the Macros menu. Running the macro in the VBA debugger by pressing the start button  will not store the macro's contents in the history list.
3. Naming conflicts with existing parts of the project may occur which need to be addressed carefully.

More information on VBA macro programming can be obtained from any one of the following sources:

1. The online help system which contains a book on macro programming (*Help⇒VBA Macro Language*). A few examples are provided there which contain code segments for various frequently used tasks.
2. The *Global Macro Path* contains a folder named "VBA_userdefined" which contains a collection of examples on accessing result data.
3. The examples folder in the CST MICROWAVE STUDIO® installation directory contains a "VBA" folder which also contains some examples on using VBA programming.

Chapter 6 - Overview of Installed VBA Macros

The installation of CST MICROWAVE STUDIO® (CST MWS) contains a selection of pre-loaded global macros which are useful to automate a variety of common tasks.

The following sections of this chapter will explain each of these macros which are collected in a couple of groups:

1. File and result handling
2. Structure macros pre and post processing
3. Creating pictures, videos and reports
4. Post processing Individual simulations
5. Post processing multiple simulations and running parameter sweeps
6. Batch mode and job control systems

The global macros are generally stored in the Global Macro Path which can be set by choosing *Macros* ⇒ *Set Global Macro Path*.

The macros can be started by selecting its name in the *Macros* menu. Please choose the menu item *Macros* ⇒ *More* in order to access all macros.

For more information on macro programming in general, please refer to the previous chapter which focuses on the VBA language and its application within CST MWS.

If you experience problems using the pre-loaded macros, please report them to support@cst.com.

6.1 File and Result Handling

This section deals with two macros for the file and result management:

1. File / Make Archive
2. Wizard / Delete Old Results

The *Make Archive* macro puts all CST MICROWAVE STUDIO® files into a single compressed ZIP archive. The *Delete Old Results* macro can be used to delete individual calculation results.

File / Make Archive – Reporting a problem to CST

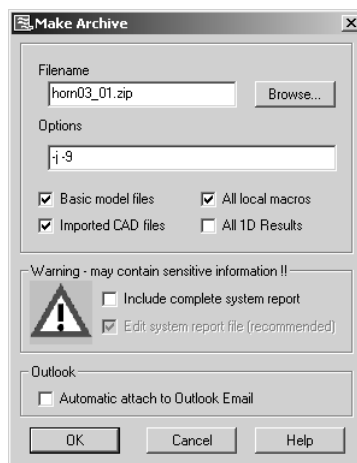
Use the *Make Archive* macro to collect all model data together into a single compressed ZIP file. This macro ensures that all necessary files required to reconstruct the simulation will be included, when reporting a problem to the CST support team.

It offers the choice to include 5 different sets of files:

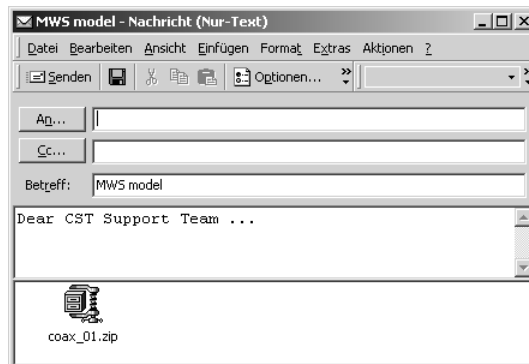
1. Basic model files (MOD, PAR, USF, GFC, PFC).
2. Imported CAD files (SAT, IGS, DXF, STEP, ...).
3. Local macros.
4. 1D Results.
5. Complete system report.

It is not recommended to include the *global macros* in the ZIP file, since they are normally not modified.

Please note: The complete system report may contain sensitive information. Please edit the system report file to make sure that only the relevant information will be added to the file.



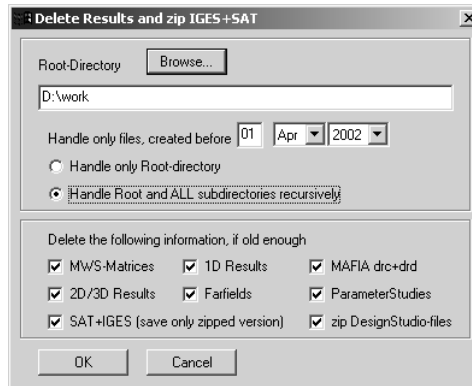
After pressing the *Ok* button, the ZIP archive will be created in the project folder. In addition, you can also choose to automatically open a new email (with Outlook 2000) and attach the ZIP file to it.



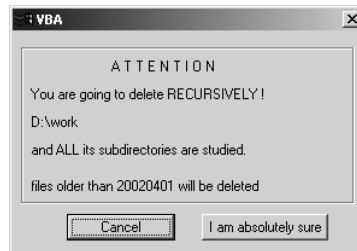
Wizard / Delete old results / Search and Delete

This macro helps to delete some of the calculation results and preserve only result data which is really needed. The erasing procedure can be controlled by information content and by creation date.

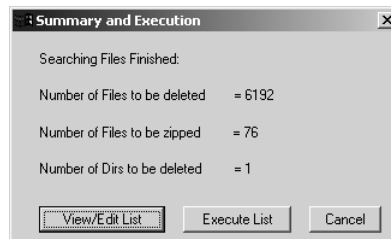
The biggest CST MWS files on the disc are typically the matrices and the monitor data (2D/3D results and farfields). This macro ensures that the files which waste the most disc space are deleted.



Since the deletion of files is an extremely sensitive task, you will have to answer some questions. Some more questions have to be answered when deleting newer, important calculation results to avoid that data will be deleted accidentally. In the last line of the following window, the date is displayed in a YYYYMMDD notation:



Firstly the macro collects all files to be deleted or zipped. You can view and edit the file list, before actually deleting the files.



By pressing the *Execute List* button the erasing procedure will be started.

6.2 Structure Macros

As explained in the previous chapter, structure macros are stored in the history list to make sure that the rebuild operation stays consistent.

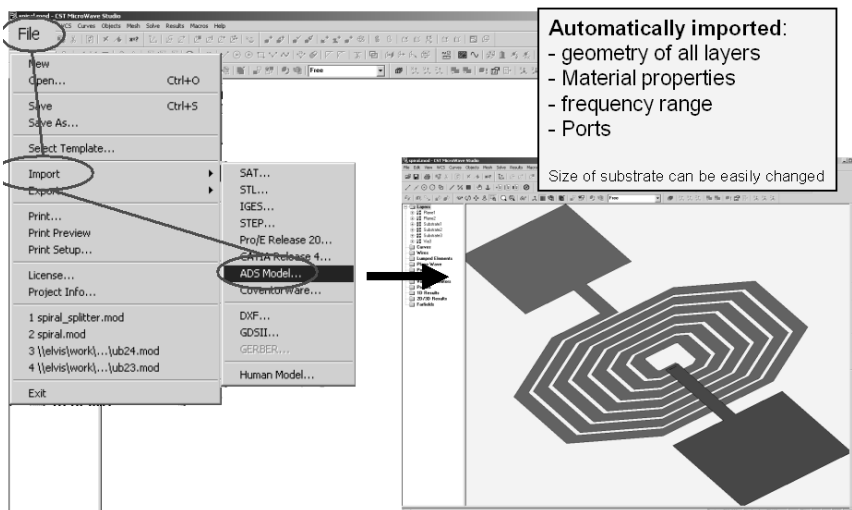
In this section the following macros are described:

- Link to Agilent's Advanced Design System (ADS)
- Macros for Construction of Coils
- Construct / Create 2D-Curve analytical (xy/uv)
- Construct / Create 3D-Curve analytical (xyz/uvw)
- Construct / Multiple Discrete Ports

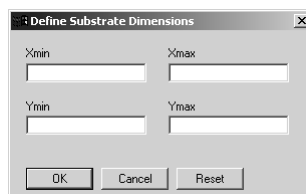
Please note: Structure macros are only added to the history list if they are executed from the pull down menu *Macros* ⇒ If executed from the VBA Debugger by pressing Start ▶, the contents are not written to the history list. This leads to rebuild problems during parameter sweeps or when reopening the model file.

Link to Agilent's Advanced Design System (ADS)

You can import project files from ADS via *File* ⇒ *Import* ⇒ *ADS Model*



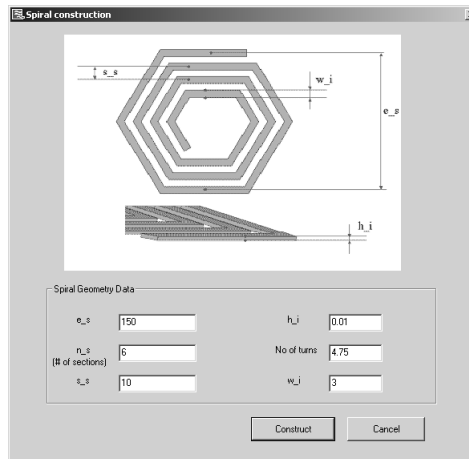
Rename the file “ADS_menu.cfg” to “menu.cfg” in the global macro folder and restart CST MWS. A new item to adjust the substrate dimensions is added to the menu, which can be accessed by *Solve* ⇒ *Define Substrate Dimensions*:



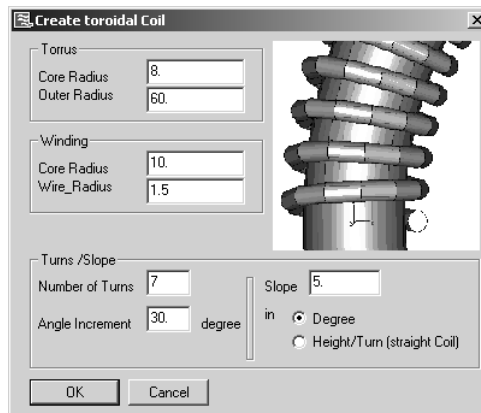
Macros for Construction of Coils

The installation of CST MWS automatically offers you numerous possibilities of coil modeling macros: Please take a look in the menu under *Macros* ⇒ *Construct* ⇒ *Coils* ⇒ ..., where you will find parametric solutions for rectangular, circular and even sectional coils.

Using those macros, you can easily draw complex spirals in CST MWS directly as e.g. the imported example from ADS shown on the previous page within a few seconds.



Also 3D Toroidal Coils can be drawn using one of these macros:



Construct / Create 2D-Curve analytical (xy/uv)

Construct / Create 3D-Curve analytical (xyz/uvw)

These macros enable the definition of curve elements via analytical functions in CST MWS.

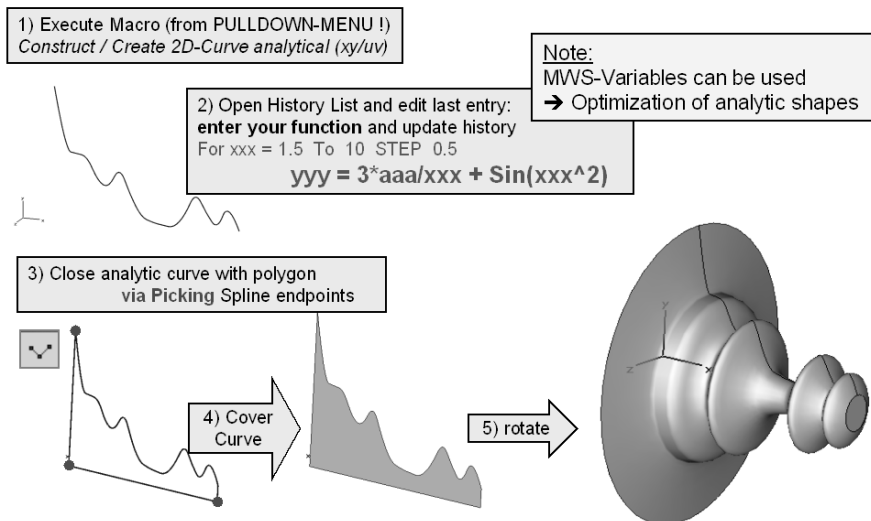
Rather than importing a list of points, you can comfortably define the analytical function itself.

This has the advantage that the curve can be still handled in a parametric way, which is very useful for optimization and parameter sweeps.

After the curve is created, all kinds of curve actions are possible:

- Creation of solids
- Creation of surfaces (e.g. to support the healing of CAD data)
- Generation of thin wires
- Definition of integration paths in the post processing

The following example shows how to generate a figure of rotation based on an analytical function:



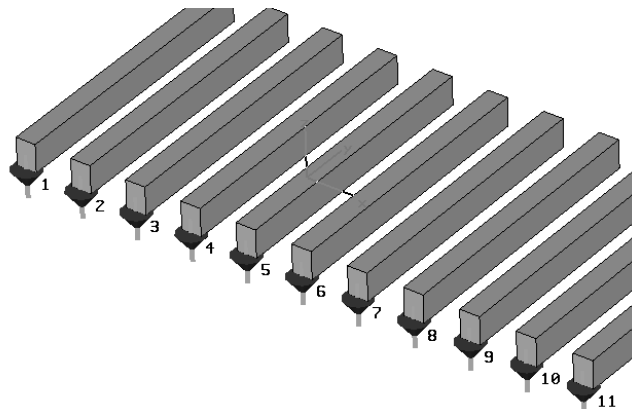
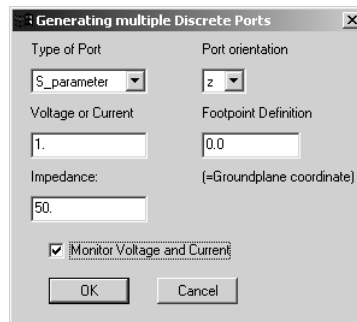
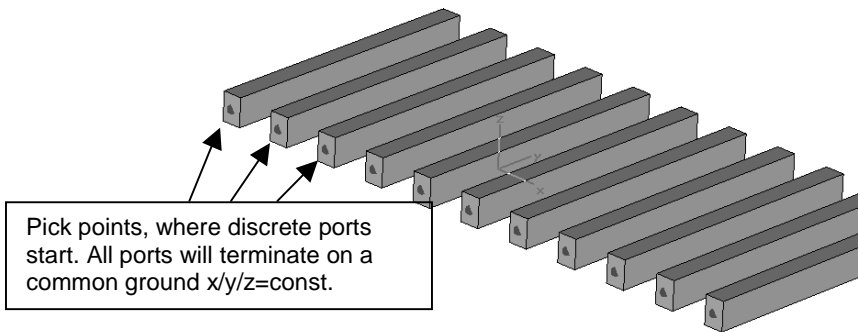
Please note: It is recommended to firstly execute the macro from the menu *Macros* ⇒ *Construct / Create 2D-Curve analytical (xy/uv)* containing a default curve and then open the history list and then enter your own analytical function by editing the last item as explained above.

Construct / Multiple Discrete Ports

This macro allows the simple definition of multiple discrete ports.

If all discrete ports share the same ground plane (e.g. $z=0$), a number of start points can be picked. This is quite similar to the multipin waveguide port. All the picked points are transformed into discrete ports by applying this macro. The ground plane location as well as port orientation have to be identical for all ports and are chosen in a separate dialog.

Please note: The macro is only designed for the global working coordinate system (xyz).



6.3 Creating Pictures, Videos and Reports

This paragraph deals with the following macros, which are used to create pictures, videos and reports for presentations:

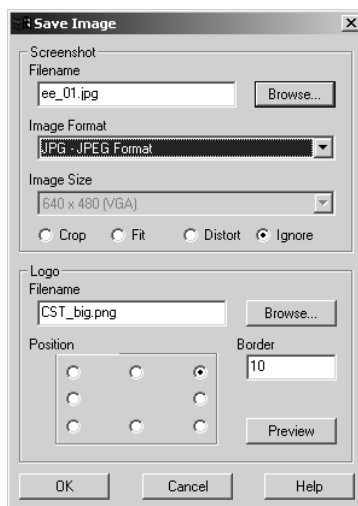
- Graphics / Save Image
- Graphics / Save Video
- Report / PowerPoint Presentation
- Report / PowerPoint Slide
- MS-Excel 2000 and MS-Word 2000

Graphics / Save Image

This macro can save screenshots in many different file formats:

BMP	Windows Bitmap
EPS	Encapsulated PostScript
GIF	CompuServe GIF
JPG	JPEG Format
PNG	Portable Network Graphic
PPM	Portable Pixmap Format
PSD	Adobe Photoshop Bitmap
PS	Adobe PostScript
TGA	Truevision Targa
TIF	Tagged Image File Format
XPM	X11 System Pixmap (color)
XWD	X11 Window Dump (color)

The macro enables the changing of the image size and the addition of a logo to the image:



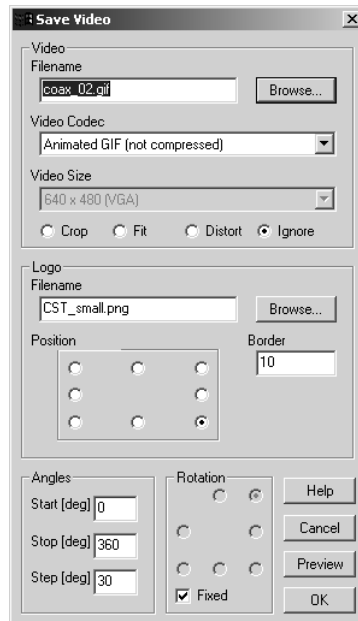
The image file will be stored in the project folder and automatically opened, if the file extension is linked to an existing image browser.

This macro requires the *ImageMagick* package installed in the global macro folder in the subfolder ImageMagick, which is created and installed automatically during the installation of CST MWS. For more information please push the *Help* button of the macro dialog.

Graphics / Save Video

This macro can store animations in several different file formats:

- MPEG II
- AVI using Microsoft Video 1 codec (requires bmp2avi.exe)
- AVI using the Radius Cinepak codec (requires bmp2avi.exe)
- AVI using the Ligos Indeo Video 5.11 codec
- Animated GIF (Not compressed)



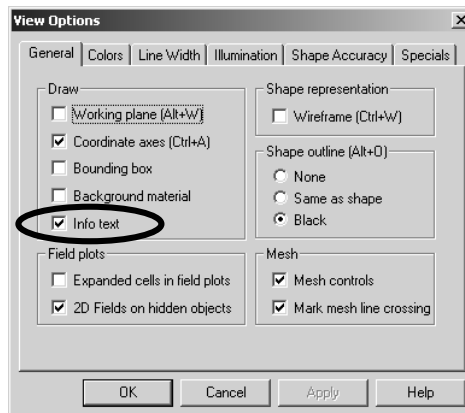
The video file will be stored in the project folder and automatically opened, if the file extension is linked to an existing video browser.

This macro requires the *ImageMagick* package installed in the global macro folder in the subfolder ImageMagick, which is created and installed automatically during the installation of CST MWS.

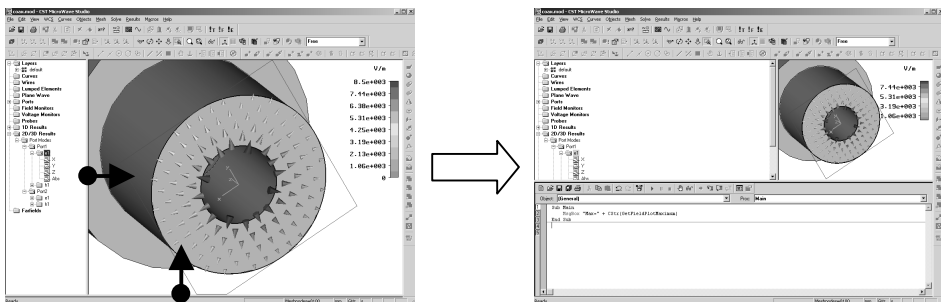
Some video formats require the bmp2avi utility, installed in the subfolder avi within the global macro folder. If you are interested in one of those formats, please download the utility from the support area of our web site (www.cst.com).

Hints:

- You can switch off the displayed info text in the View-Options (Alt-V or View⇒View Options...)



- To keep the file sizes of the pixel-oriented video files small, you can also consider decreasing the size of the drawing window, as shown in the following image. This procedure has the advantage of not rearranging the icon bars.



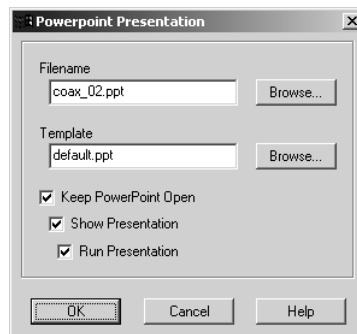
- During the video recording, the CST MWS window must be in the foreground without any popup dialogs in front of the main view.

For more details please push the *Help* button in the macro dialog.

Report / PowerPoint Presentation

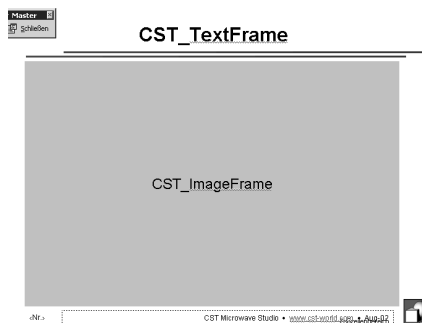
Once a transient simulation has finished, this macro automatically creates a PowerPoint presentation with some of the most important results of the simulation. This includes:

- 3D-view of the geometry
- Overview of the port modes
- Port signals In time domain
- S-parameters (linear, db and phase)
- Polar plot and smith chart
- Field energy in time domain
- Energy balance in frequency domain



The templates for the presentation are located within the global macro path in the subfolder templates. Templates are ordinary PowerPoint files that contain two special text frames in the slide master:

- Text frame with the caption "CST_TextFrame"
- Text frame with the caption "CST_ImageFrame"



These two special text frames denote the position and size of the headings and the images that will be inserted into the PowerPoint presentation. In the case of CST_TextFrame, it also controls the font, font attributes, font size, font color and the text alignment of the frame.

If you wish to create your own template, we recommend copying the existing file default.ppt and changing the style without deleting the two important CST text frames in the slide master.

If you want to change the default selection of slides, you can edit the macro and search for e.g. “Port Signals”, if you want to remove the port signals from the presentation. However, this procedure is only recommended if you constantly need a different, but always identical selection of results (e.g. if you always carry out eigenmode computations and never transient simulations).

The most flexible solution is to not edit the contents of this macro, but to use the related macro *Report / PowerPoint Slide*, which is explained in the following section.

Report / PowerPoint Slide

This macro adds the actual contents of the CST MWS drawing window to an open PowerPoint presentation. If no PowerPoint presentation is open, you will be asked for a template. The rules for templates are identical to the ones explained under Report / PowerPoint Presentation.

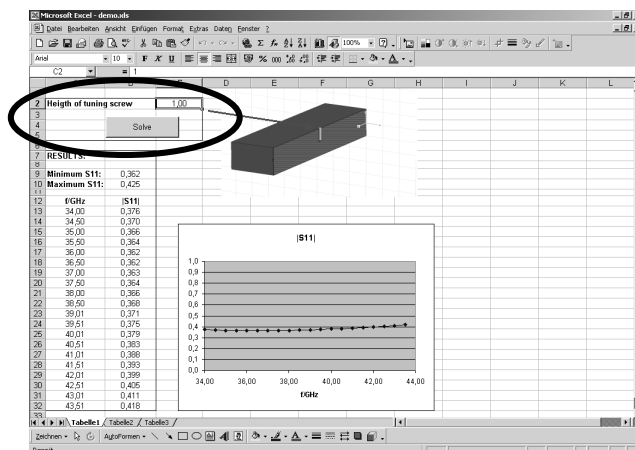
Please note: The active PowerPoint presentation has to fulfill the rules for templates, otherwise the addition of the slide will fail.

MS-Excel 2000 and MS-Word 2000

In the installation directory of CST MWS you will find under the folder Examples\VBA examples for connecting CST MWS to MS Word 2000 and MS Excel 2000.

□ Controlling CST MWS from MS Excel 2000

After opening the file “Examples\VBA\Excel2000\demo.xls” with MS Excel and activating its macros, you can change the height of the tuning screw and start the MWS simulation by pressing the button *Solve* within the excel sheet. CST MWS then automatically opens the model and performs the simulation using the new setting for the parameter height. After the simulation, all results within the *Excel* sheet are updated including the screenshot.



□ Creating a Word-Report from CST MWS:

Open the file “Examples\VBA\Word2000\wg.mod” within CST MWS and execute the local macro file “wg.000”, named “Word Report” (*Macros* ⇒ *Word Report*). You can then check the created word.doc in the same folder.

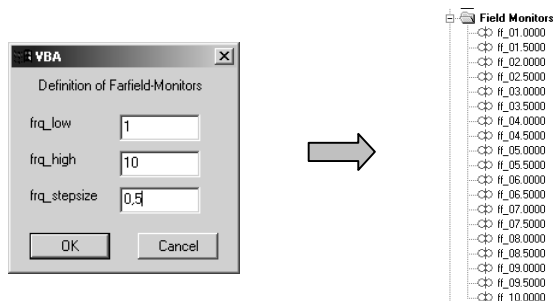
6.4 Post Processing Individual Simulations

In this section the following post processing macros for an individual simulation are presented:

- Farfield / Broadband Farfield Monitors
- EMC / Calculate Broadband EMC-Norm
- EMC / Renorm Probes+Voltages to EMC-Frequency Source
- Results / 2D-3D Export and Energy Calculation
- Time Signal Processing / TDR Computation
- Time Signal Processing / Exchange Excitation
- Time Signal Processing / Eye Diagram
- Touchstone + SPICE / Calculate SPICE S-Parameters
- Touchstone + SPICE / Touchstone Reduction
- Further Examples for 0D- and 1D-Results

Farfield / Broadband Farfield Monitors

This macro defines a set of monitors for broadband farfield analysis. Further down in this document (in Chapter 7.3) you will find corresponding post processing templates to automatically calculate broadband farfield properties such as gain, directivity, etc.

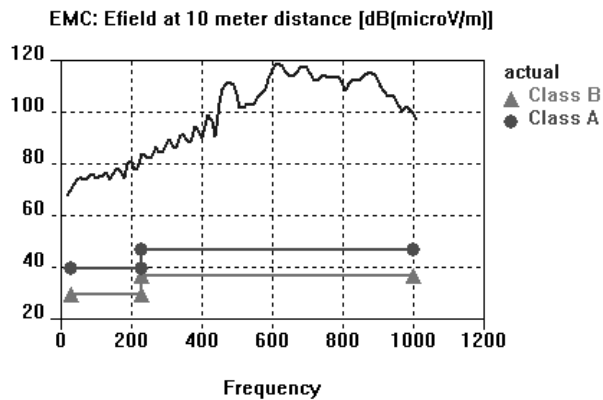


The next page shows you how to use these defined monitors to compare and to check the structure's radiation against EMC norms.

EMC / Calculate Broadband EMC-Norm

This macro evaluates the broadband behavior of maximum electric field strength ($\mu\text{V}/\text{m}$) in 3m and 10m distance of the structure. A comparison to the standard EMC norms is carried out automatically.

The macro needs the definition of broadband monitors before starting the transient simulation. Use the macro *Monitor / Broadband Farfield Monitors* to create the monitors.

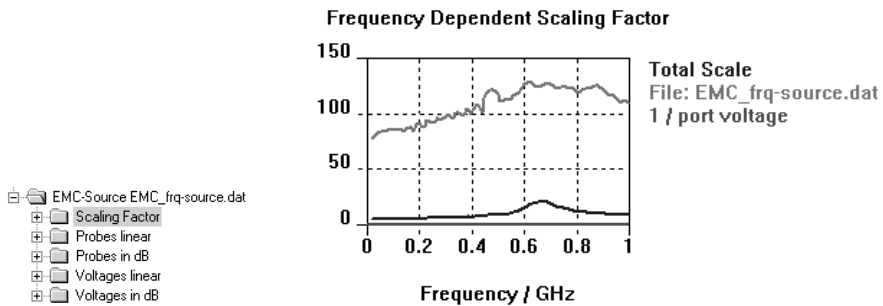
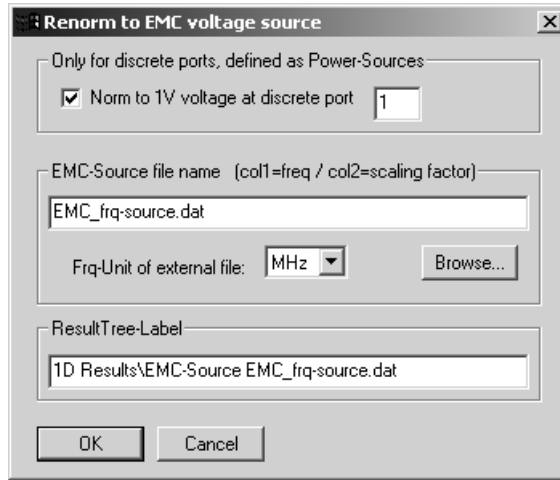


EMC / Renorm Probes+Voltages to EMC-frequency source

This macro performs a frequency scaling to all probe and voltage monitors. Two different ways of scaling can be applied:

1. for discrete ports, defined as power sources (normalized to 1 Watt peak power broadband), a voltage normalization to 1 Volt can be applied, if the port voltage has been monitored during the transient simulation.
2. Additionally or separately, an external data file can be used for scaling. The frequency unit of the external spectrum can be adjusted to be different from the project's unit.

All renormalized probes and voltages are displayed in a separate folder, which can be chosen by the user.

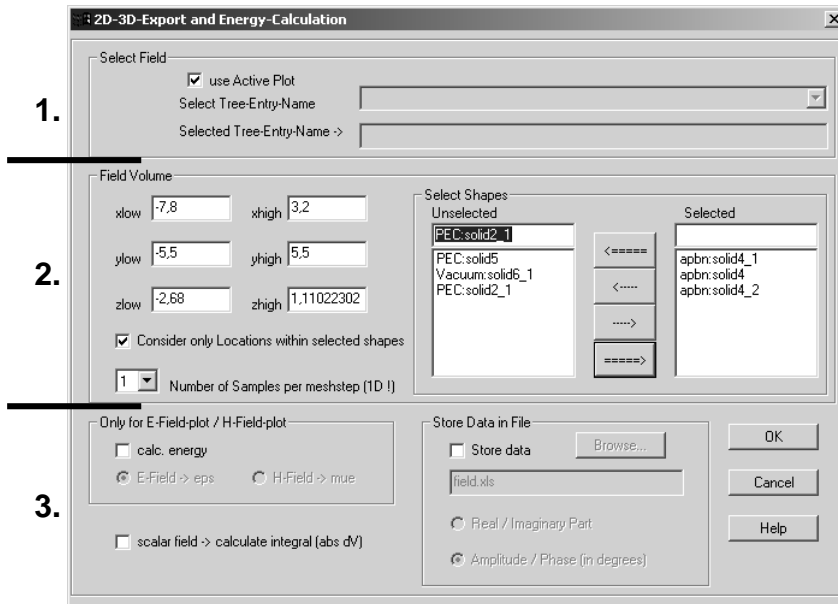


Results / ASCII Export + Energy / 2D-3D Export + Energy_Loss Calculation

This macro calculates the stored energy within a certain volume and, optionally, even within selected shapes only.

The usage of this macro can be split into 3 steps (also shown in the dialogue):

1. Select a field item (Which 2D/3D Result is the basis of the computation?)
2. Define the volume and sample points to be considered for integration and export.
3. Define the action from the following choice
 - Calculate stored energy (for selected E- or H-fields only)
 - Calculate volume integral of the absolute values
 - Export Sample points and corresponding field values into a ASCII file for further external processing.



Please note:

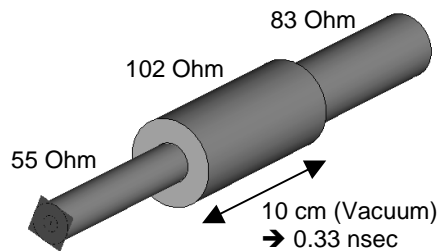
- The execution process might take several minutes, especially if the option “within Selected Shapes only” is chosen (the point test, whether a point is located within a selected shape, is extremely cpu-consuming).
- The field sampling per mesh cell can be adjusted, but it may make the calculation even slower (already the choice of 2 samples per 1D mesh step increases the number of 3d points by a factor of 8.)

Time Signal Processing / TDR Computation

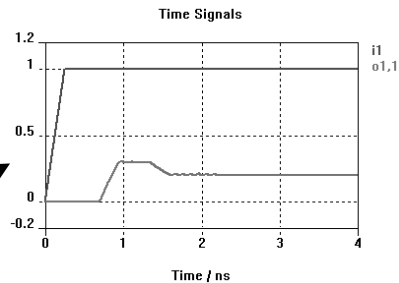
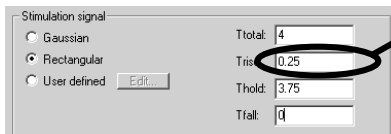
This macro evaluates the impedance profile along a TEM or similar structure based on the reflected time signals at a port.

TDR computation (Time Domain Reflection) is well known from measurement, where TDR is the only possibility to “look into a structure” and locate mismatches. Typically TDR is applied to TEM structures such as connectors or IC packages designed to have minimum return loss.

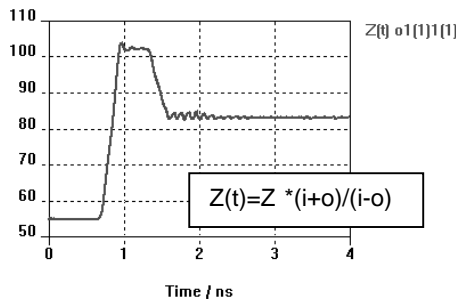
CST MWS with its transient solver is ideally suited for this task, since TDR pulses have to be short in time (to be able to resolve discontinuities in space) and thus extremely broadband.



Use rectangular pulse with rise time shorter than distance of discontinuities:



TDR Results: Time Dependent Impedance $Z(t)$ in Ohm



Please note:

- Choose the upper frequency limit (important for meshing) to at least $1/\text{Trise}$.
- The frequency spectrum of a rectangular pulse has zero points, where no energy is injected into the structure. At these frequencies S-Parameters may have some inaccuracies.

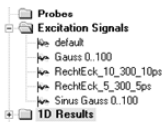
Time Signal Processing / Exchange Excitation

This macro can be used to exchange the excitation signal without rerunning the simulation. Based on the given S-Parameters (calculated over a sufficiently large frequency band), new excitation signals can be tested in the device. The macro is able to load and test arbitrary signals from Excitation Signals tree folder and the library.

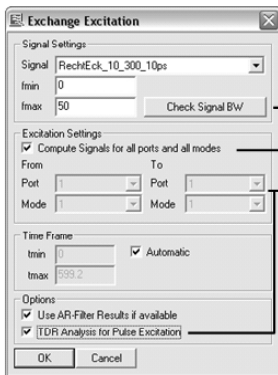
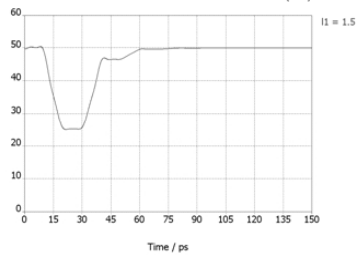
Please note that there is also TDR functionality for this macro.

This solution is also available as a Postprocessing template (Chapter 7) which is useful to exchange excitations and record the results for each parameter set within a parameter sweep.

Define/Load Excitation Signals from library to the Result Tree



Template Result for Parameter Sweeps here the TDR Result: Z1(1)

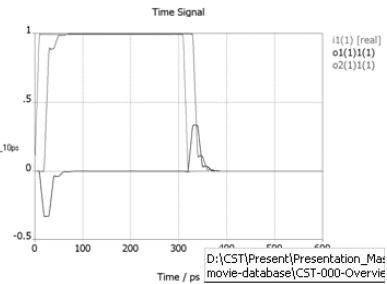


Compare Signal BW to S-Parameter BW

Select Ports/Modes for the Template Output



Additional Output

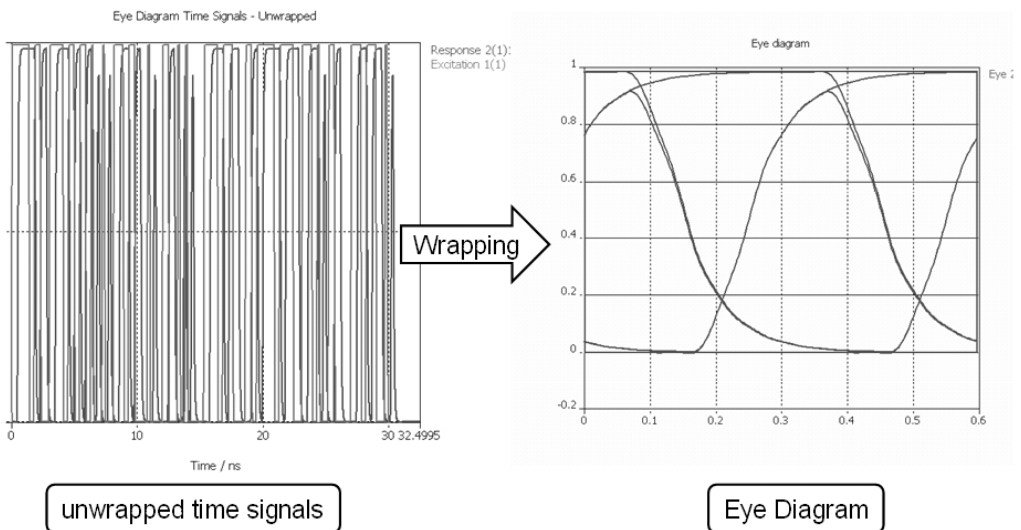
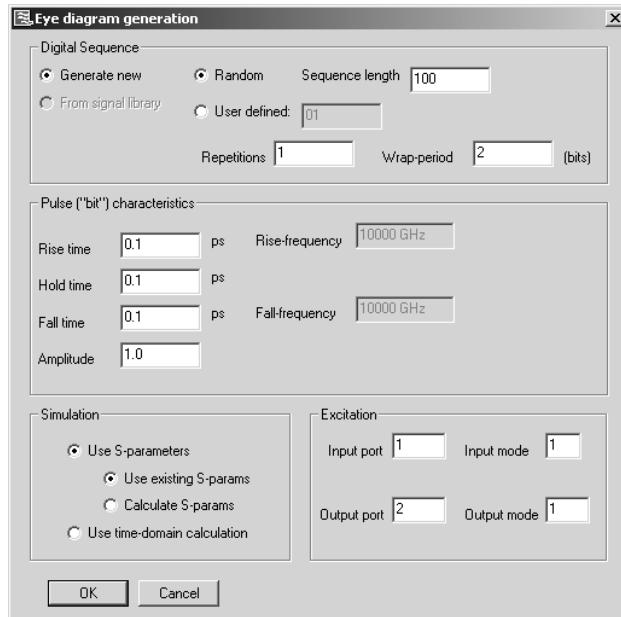


D:\CST\Present\Presentation_Mat
movie-database\CST-000-Overvie

Time Signal Processing / Eye Diagram

This macro is a special case of the macro Exchange Excitation just described. Here a bit sequence (randomly- or user-defined as 1,0 combination e.g. 10011011) is taken to drive the device. This sequence can be repeated several times. The bit characteristics can be defined separately.

At the end of the simulation, a wrapping of the transmitted signal is performed in order to see the opening of the eye.



Touchstone + SPICE / Calculate SPICE S-Parameters

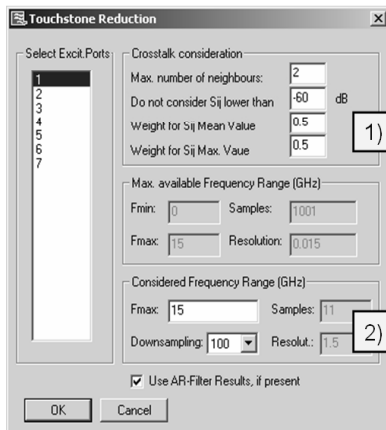
This macro is explained in the Getting Started document in the section “SPICE Network Model Extraction” close to the end of that book. There are 2 versions (TEM and MOR), depending on which extraction method has been used before.

Touchstone + SPICE / Touchstone Reduction

This macro allows you to reduce the number of ports for e.g. an IC-package simulation, where you are interested in having a small Touchstone file.

For specific excitation ports, the S-matrix will be reduced based on the electric coupling to other ports. Giving the maximum number of neighbours and a zero-threshold, the number of ports can be significantly reduced, especially for big packages with weak coupling to many ports.

The results (the complete new MWS models with the new number of ports as well as the corresponding Touchstone files) are automatically generated and located in a subfolder, named <project>.Touchstone_reduced. Furthermore, for each excitation port you get a reference table with the renumbered ports, as well the crosstalk level between all ports.



Touchstone Reduction by

- 1) Reducing number of ports (only important ports with high crosstalk)
- 2) Downsampling frequency range



New Ports are sorted by size of crosstalk Sij [dB]

Port-orig	Port-new	SdB_dc	SdB_mean wgt=0.5	SdB_max wgt=0.5	SdB_weighted
a) Excitation Port (Reflection)					
1	1	-9.55	-6.4	-4.95	-5.67
b) Galvan. Connected Port(s) (Transmission)					
4	2	-3.53	-5.02	-3.53	-4.27
7	3	-3.52	-4.85	-3.52	-4.18
c) Crosstalks, sorted by SdB_weighted-Value					
2	4	-72.3	-15.57	-13.33	-14.45
5	5	-70	-17.72	-13.56	-15.64
6	---	-71.03	-24.44	-18.25	-21.35
3	---	-73.92	-28.47	-19.18	-23.83
d) below Zero-Threshold (-60 dB)					

Further examples for 0D- and 1D-Results

In the the subfolder VBA_userdefined within the global macro folder, you will find some more example files for post processing:

- Multiplot of farfields at different frequencies
- Automatic extraction of losses and Q-values for 3D eigenmodes
- Calculation of RLC and Q from S11

You can manually access such a bas-file by loading it into the VBA-editor ( in IDE) and executing it ()

If you wish to automate the access to one of those 0D- or 1D-Results, you can use the following syntax:

```
'#include "mws_evaluate-results.lib"
Sub Main
  ` example 0D-Result
  Dim q1 As Double
  q1=Evaluate0D("0D-Result_3D-Eigenmode_Q_value1.BAS")
  MsgBox CStr(q1)

  ` example 1D-Result
  Evaluate1D("1D-Result_VSWR.BAS")
End Sub
```

6.5 Postprocessing multiple simulations and running Parameter Sweeps

This paragraph deals with the following macros, which are used to compare results of multiple projects, run parameter sweeps with advanced comparison possibilities or non-equidistantly sampled parameter ranges:

- Wizard / Compare multiple runs
- Wizard / Parameter + Mesh Study
- Wizard / Parameter Study (non-equidistant stepsize)

Before going into the details of the automatic comparison methods, it should be pointed out that there is a Copy-Paste functionality for arbitrary 1D-Result curves. Ctrl-C and Ctrl-V work within a project, across different models and also to bring displayed data into e.g. an Excel-Sheet or other tools.

Wizard / Compare multiple runs

This wizard enables you to compare all kinds of 1D Result data from different simulations with little user interaction.

The user is mainly involved in three tasks:

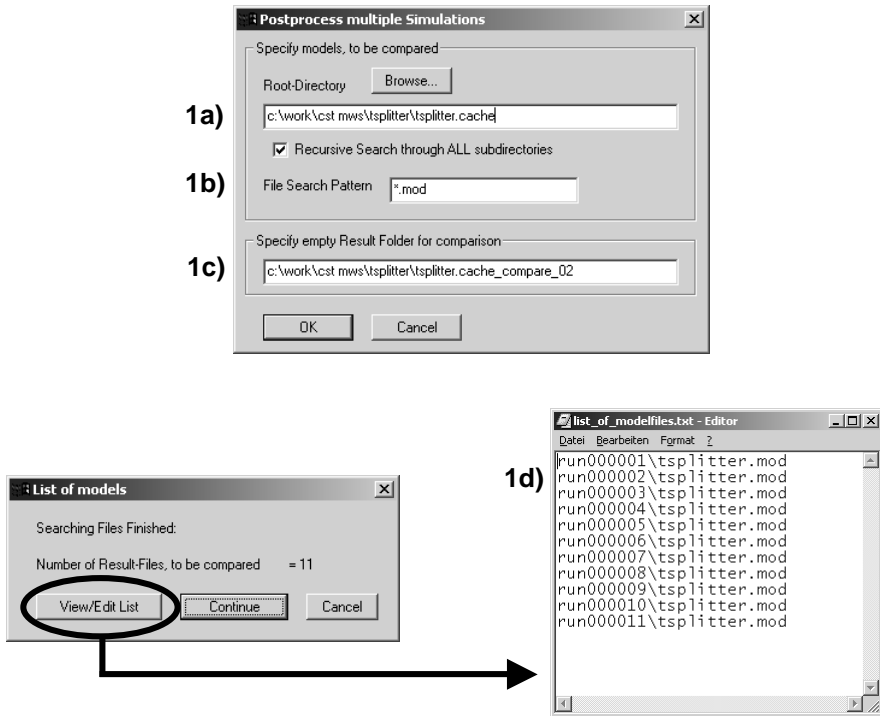
- 1) Selection of projects for comparison
- 2) Selection of result types for comparison
- 3) Comparison of the selected results

1) Selection of projects for comparison

To select the projects to be compared the user has to

- 1a) define a root directory in your windows system – if it exists, the result cache directory would be the default – and
- 1b) define an additional file template (e.g. coaxial*.mod) and
- 1c) specify an empty folder for the result documents of the comparison – the comparison folder – and
- 1d) in case not all or too many projects are selected for the comparison, edit the project list manually.

Please note: All project names are addressed relative to the root directory.



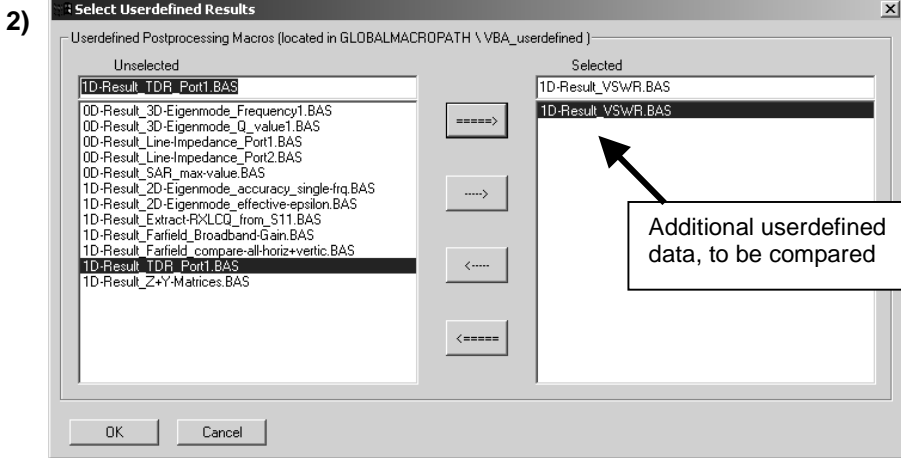
2) Selection of result types for comparison

A large part of the 1D result data is compared automatically. This includes

- all S-parameters (linear, dB, phase and polar),
- all existing AR-filter results,
- all time signals (both incoming and outgoing),
- all balance and energy curves,
- all probes and voltages (as time signals as well as frequency information) and
- all 1D- and 0D-Results, defined via the Postprocessing templates (for detailed information read chapter 7)
- all impedances, currents and voltages (for lumped elements and discrete ports).

In most cases, this default comparison is probably sufficient to extract the desired information.

However, within an extra dialog you may select additional data that is used for the comparison. This data can be either a single real scalar value per simulation or a sequence of scalar values (1D curve, e.g. VSWR) per simulation.

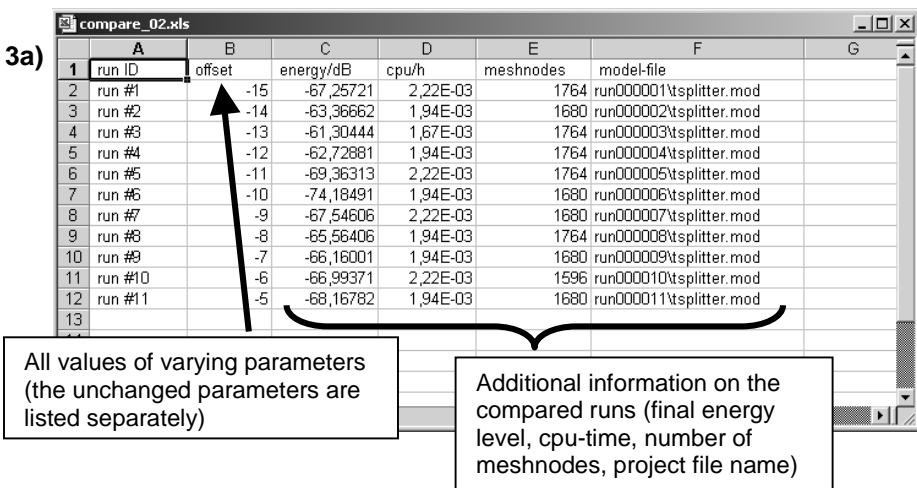


Once you have finished selecting the result types for the comparison by confirming with *Ok*, CST MWS will automatically collect all the results from the different projects and store them in a central document, located in the comparison folder. Depending on the number of projects and on the file sizes of the respective results, the collection can take a few minutes.

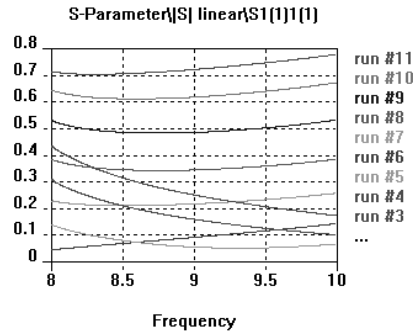
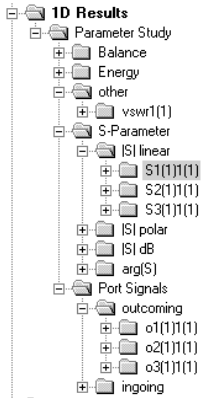
3) Comparison of the selected results

Finally, you obtain two different documents, containing the selected data:

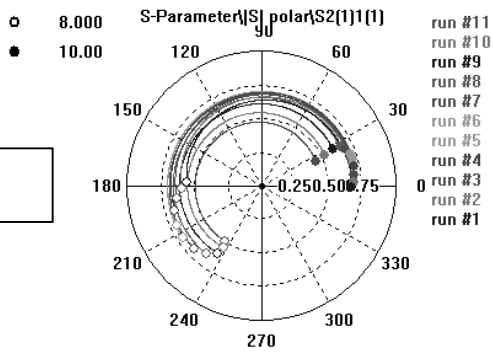
- 3a) An Excel sheet, showing the run IDs, the corresponding parameter values and original project file name.
- 3b) A new project file containing all the result data but not containing any geometry.



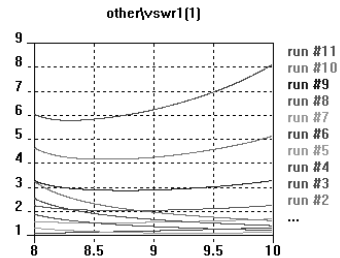
Whereas the Excel sheet serves as a reference to connect parameter values to run-ID numbers, all comparison plots are performed within the newly generated MWS project:



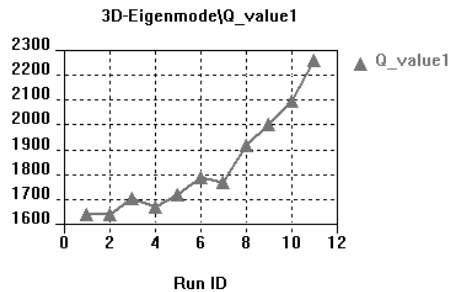
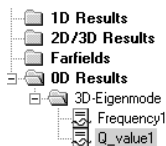
Example for comparing polar S-parameters



The userdefined selection (VSWR) is displayed in the tree under 1D Results \ other \ VSWR



Single real scalar values will be compared in a separate tree folder:



Implementing your own Result routines

As you have seen in the previous example, a number of examples for additional userdefined postprocessing already exists. All these files are located in a central directory within your installation (GLOBAL_MACRO_PATH \ VBA_userdefined), so that you can also access them manually or add your own examples.



In case you wish to add your own examples, please keep in mind the following guidelines:

1. Single real scalar values

- The filename has to have the pattern "0D-Result_*.BAS"
- The internal variable `sMacrobase` has to be set to that filename.
- The single scalar value should be assigned to the variable `cst_value`.

2. 1D-Results

- The filename has to have the pattern "1D-Result_*.BAS"
- The 1D result curve(s) should be stored in a two column ASCII file(s), named `GetProjectBaseName+ "^*.sig"`, so that they are automatically compared.
- Please do not change the header, nor the end of the macro which controls the error handling.

You can manually access such a bas-file by loading it into the VBA-editor ( in IDE) and executing it ()

If you wish to automate the access to one of those single scalar value or 1D curve results, you can use the following syntax:

```
'#include "mws_evaluate-results.lib"
Sub Main
  ` example 0D-Result
  Dim q1 As Double
  q1=Evaluate0D("0D-Result_3D-Eigenmode_Q_value1.BAS")
  MsgBox CStr(q1)

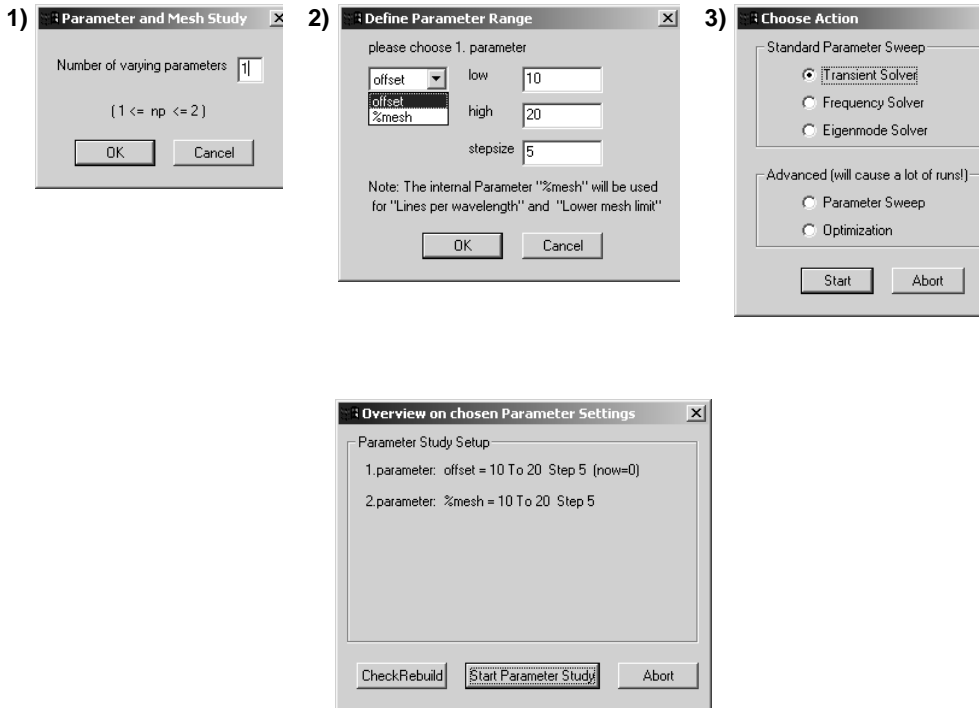
  ` example 1D-Result
  Evaluate1D("1D-Result_VSWR.BAS")
End Sub
```

Wizard / Parameter + Mesh Study

This wizard performs a one- or multi-dimensional parameter sweep. It is similar to the built-in solution that may be accessed from the solver dialog. However, there is a difference: All available 1D result curves are automatically compared at the end of the parameter sweep. Therefore, the same comparison strategy is used as introduced in the previous section.

By the use of different dialogs, the wizard requests the settings for

- 1) the number of varying parameters,
- 2) the parameter range and step width for each parameter and
- 3) the type of solver that should be started.



For each solver run, the wizard stores all data related to the current parameter setting in a subfolder, including the project file and the result data. Therefore an additional comparison can be performed using the "Wizard / Compare multiple runs" subsequently. This macro "Wizard / Parameter + Mesh Study" might be a useful addition to the built-in sweeping solution in the solver dialogs in the following cases:

- If you need to compare all the 1D results: the wizard automatically compares all available 1D Results, e.g. probe or voltage monitor data, without the need to define any extra watches or result templates.
- In the advanced case of performing two coupled parameter sweeps, e.g. to calculate the group velocity of a TWT structure as function of a geometrical parameter, this macro is of great use :

The group velocity calculation itself already requires a parameter sweep, varying the phase delay at the periodic boundary.

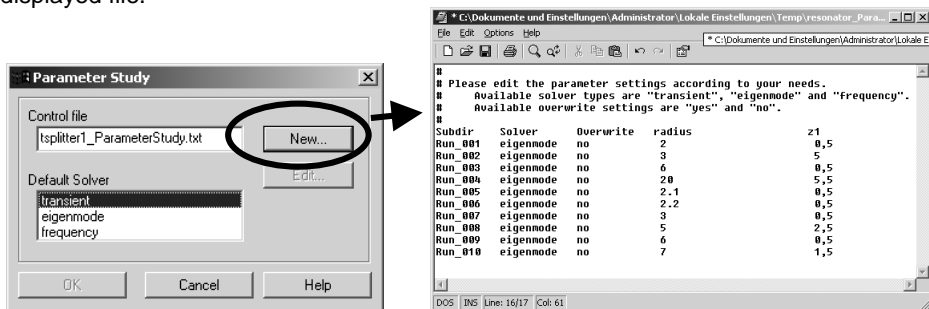
In a second outer loop a geometrical parameter could be varied, automatically leading to the desired multiple plot of group velocity curves.

- The second advanced case, starting an optimizer for any parameter setting, has maybe no practical background, however it is included for the sake of completeness.

Parameter Study via Parameter Table

The macro “Solver / Parameter Study (non-equidistant stepsize)” performs parameter sweeps according to given parameter values that are stored in a textfile.

If you wish to carry out parameter sweeps with arbitrarily changing parameter values, this solution is quite convenient. You just enter the parameter values of interest in the displayed file.



After all solver runs have been performed, the results can be compared using the “Wizard / Compare multiple runs”.

6.6 Batch Mode and Job Control Systems

This section mainly focuses on how running a sequence of calculations can be automated by using CST MICROWAVE STUDIO®.

The following topics are covered here:

1. Running CST MICROWAVE STUDIO® in batch mode
2. CST Job Control Center
3. Running simulation jobs across a network.

Running CST MICROWAVE STUDIO® in Batch Mode

CST MICROWAVE STUDIO® can be operated in a so-called *batch mode*. Whenever the program is called with a project file name as an argument, the specified file will be automatically loaded before the interactive mode is entered.

A collection of command line options is available to specify whether a simulation should automatically be started after the file is loaded rather than entering the interactive mode:

```
/t : starts the transient solver  
/e : starts the eigenmode solver  
/f : starts the frequency domain solver  
/o : starts the optimizer  
/p : starts the parameter sweep  
/n : starts the network parameter extraction
```

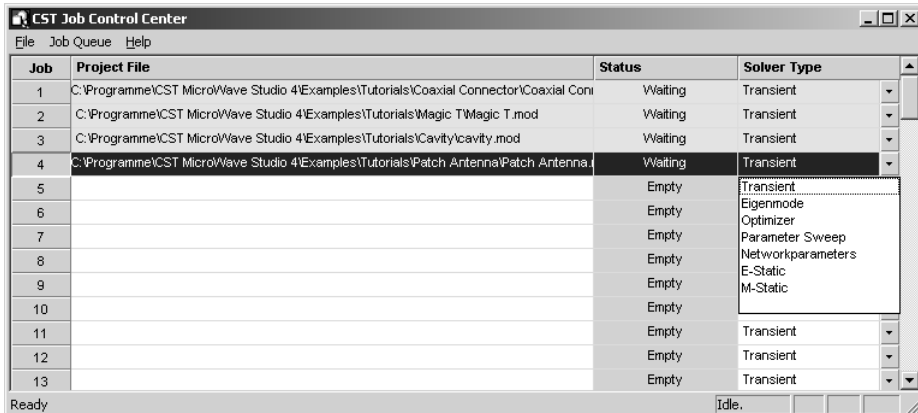
CST MICROWAVE STUDIO® will then automatically be terminated after the calculation has finished.

In addition to this batch mode, a VBA script file with the file extension “bas” can be optionally specified rather than a project file. This file will then be automatically executed after starting CST MICROWAVE STUDIO®.

CST Job Control Center

The CST Job Control Center offers a convenient way to setup and perform batch jobs without the need to call CST MICROWAVE STUDIO® with command line options.

This software can be started by selecting its item in the CST MICROWAVE STUDIO® folder of the Windows *Start* menu. The user interface is shown in the picture below:



You can simply add a job to the queue by selecting *Job Queue* ⇒ *Add Job*. Please specify the project name which should be queued for later execution. You should then select the corresponding *Solver Type* from the list in the last column.

The processing of the job queue can be started by selecting *Job Queue* ⇒ *Run* and stopped by choosing *Job Queue* ⇒ *Stop*. Once the queue has been processed, the status field will show the current status for each of the jobs.

Distributing simulation jobs across a network of PCs

CST MWS allows the distribution of multiple simulation tasks across a network of available PCs. Different types of this so-called *Distributed Computing* feature are available:

- Independent port-excitations in the transient solver can be solved simultaneously on different PCs (Note: The computation of the entire S-matrix requires the stimulation at each port).
- Parameter sweeps and optimization runs can be spread across the PC-network and thus perfectly parallelized in speed.

For more information on licensing and installation of this *Distributed Computing* Feature please read the final chapter of this book.

Chapter 7 – Template Based Postprocessing

This chapter introduces a new functionality of CST MWS version 5 which enables you to perform very powerful postprocessing tasks in a comfortable way.

Sometimes you might need a Postprocessing Calculator for 2D/3D Fields, 1D Signals or simply scalar values (0D Results); the Postprocessing Templates offer you solutions for all of these tasks.


However, the postprocessing does not just work for a single simulation. When running multiple simulations of the same project, such as Parameter Sweeping, Optimization Processes and Adaptive Mesh refinement passes, you might be interested to keep and compare not only the primary data such as S-Parameter information or eigenmode frequencies, but also other data, according to your definition such as

arbitrary 1D-Results

- Z, Y, VSWR versus frequency
- Farfield 1D plots at a single frequency
- Broadband farfield values
- Group Delay Time
- 1D Plots of 2D/3D Results along arbitrary curves
- FFT of existing time signal
- Exchange Excitation and TDR functionality
- Mixture of any of these 1D-Results using your own analytic formula
- ...

and arbitrary 0D-Results (single real scalar values)

- Min, max, mean, integral, ... value of any existing 1D-Result
- Q-value, Energy, Loss, Coupling Coefficient of Eigenmode
- Curve-, face- or volume integral of 2D/3D Result
- Mixture of any of these 0D-Results using your own analytic formula
- ...

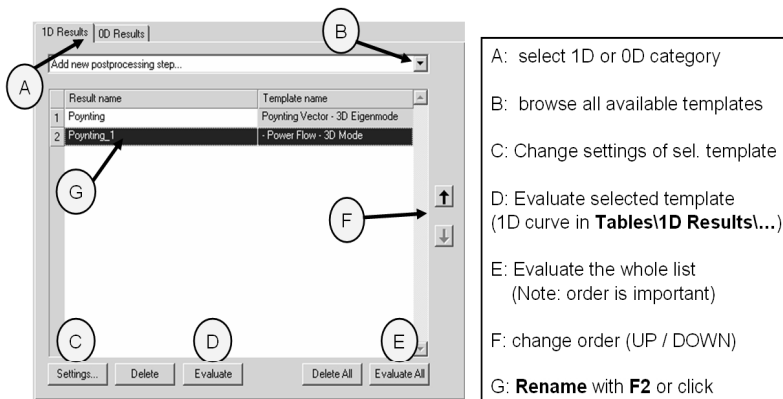
Results ⇨ *Template Based Postprocessing* (), Shortcut Ctrl-P) gives you this opportunity.

The following sections introduce the framework of this feature and afterwards present its application to different examples.

7.1 Framework to control Result Templates

The framework supports the definition of your tasks as well as their execution, storage and plotting.

The following picture shows you the various options in the definition dialogue:



The list of user defined tasks is automatically processed after each solver run. Therefore the tasks are also useful for rerunning a project with various simulation settings but with the same postprocessing. Please note that the execution takes place in the order shown in the definition table. It might be necessary to change the order (buttons F), especially if tasks refer to previously defined data.

The Framework manages data storage and visualization:

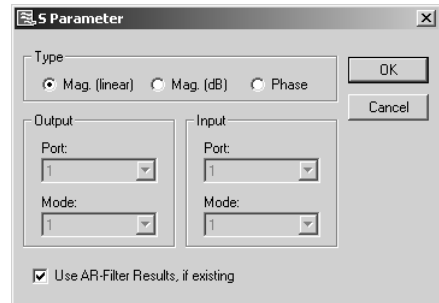
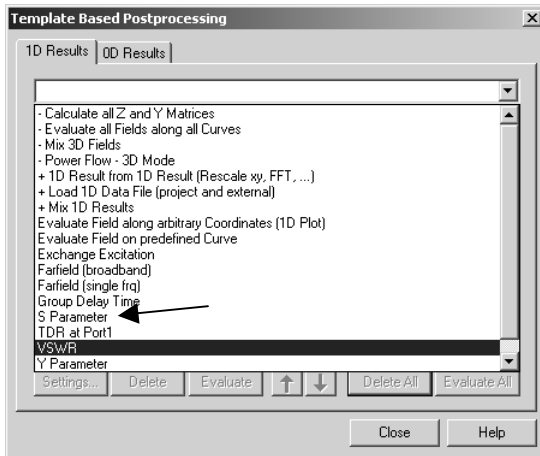
- 1D Results are shown in the tree under Tables\1D Results\Name
- 0D Results are stored in Tables\0D Results\Name . Additionally the current (last) result value is shown on the 0D-page of the definition window.

The following pages will introduce this framework step by step with a simple example, in order to get started and prepared for other tasks quickly.

Simple example to get started with Postprocessing templates

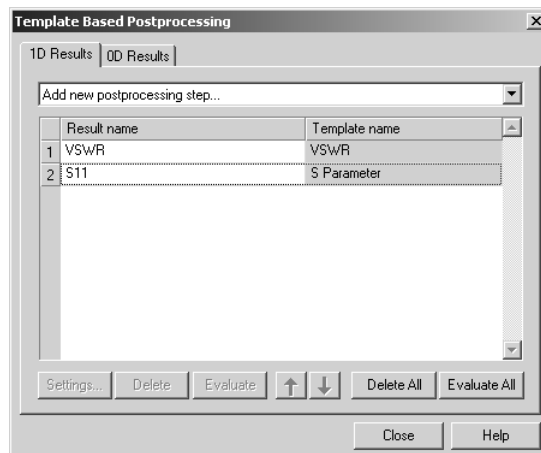
Let us assume you have simulated a device and you are interested in directly obtaining the *VSWR* and calculating the delivered Power ($1-S_{11}^2$). (You can take any example for this course, assuming it produces S-Parameters.)

Within the 1D Result page of the Template dialogue you should select the entries *VSWR* and in a second step *S Parameter*.

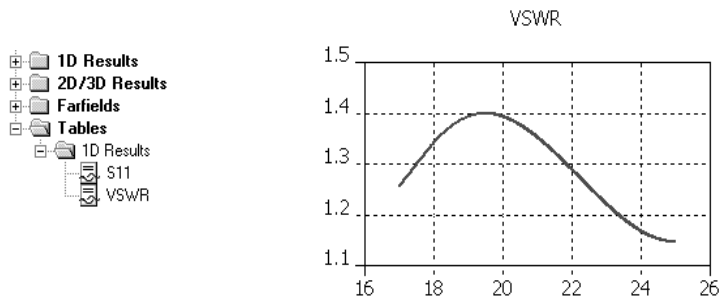


Please note: The S-parameter *S11* is also available and displayed directly in the 1D Results tree. However, since we would like to do further calculations with it, we have to explicitly load *S11* as template to be referred further down in the Mix1D Template.

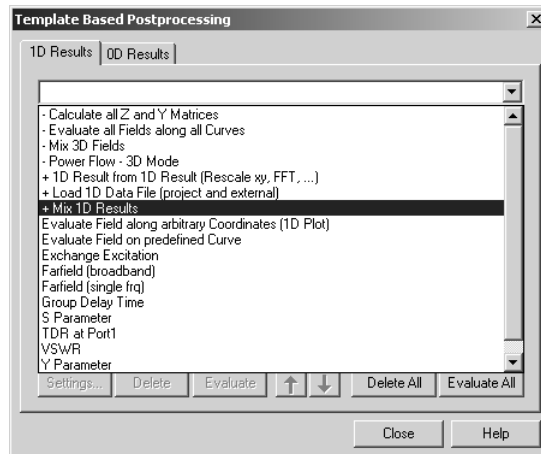
After confirming the defaults of the S Parameter window (we are interested in the linear amplitude of *S11*), you will see the following two 1D Results defined in your list:



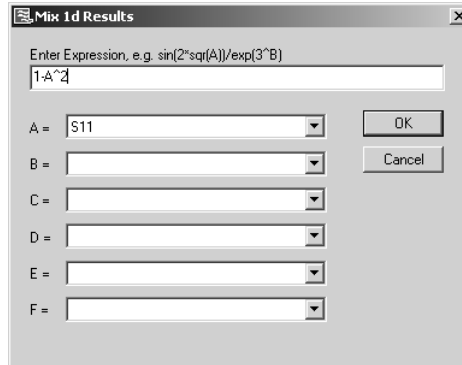
If the project has already been calculated and its results are present, you can now press *Evaluate All*, resulting in two tables in the tree, which you can select and plot. Otherwise please start the calculation with the transient solver, which at the end automatically evaluates all templates in the defined order.



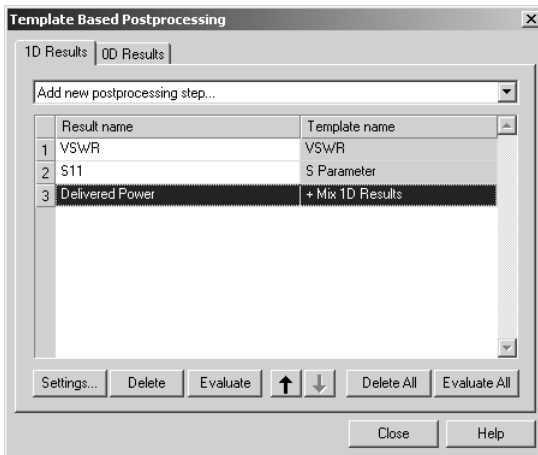
In a second step we can calculate the delivered power ($1-S_{11}^2$) using the 1D Template *Mix 1D Results*



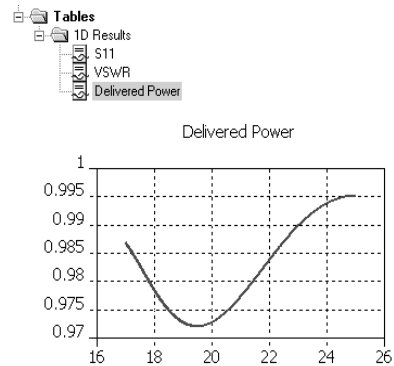
The following window opens, where we can mix arbitrary defined 1D Results using VBA expressions. If we select the synonym A for S11, our expression would be $1-A^2$.



In the framework we can finally change the name by clicking on the Result name field from Mix1D to e.g. Delivered Power. The evaluation (Button *Evaluate*) will immediately show the corresponding curve in the Table:

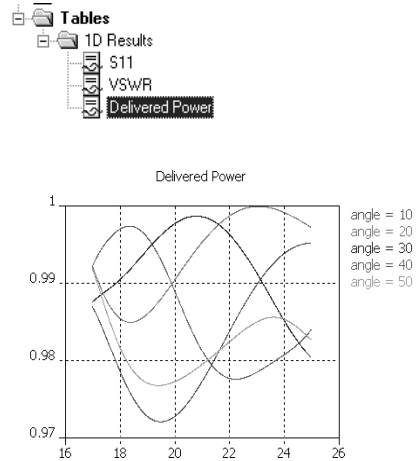


Evaluate:

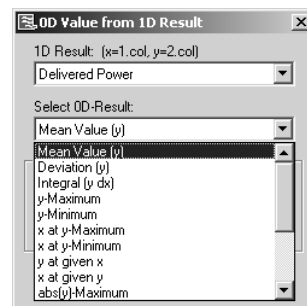
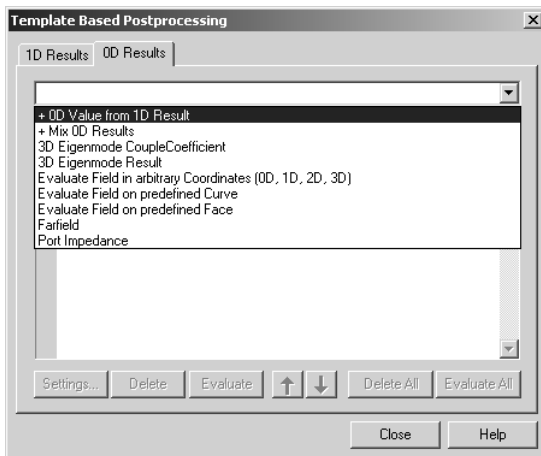


You also have the possibility to change the definition of any template. Once selected in the framework, the button *Settings* will be enabled and reopens the definition dialogue for checking purposes or even redefinition.

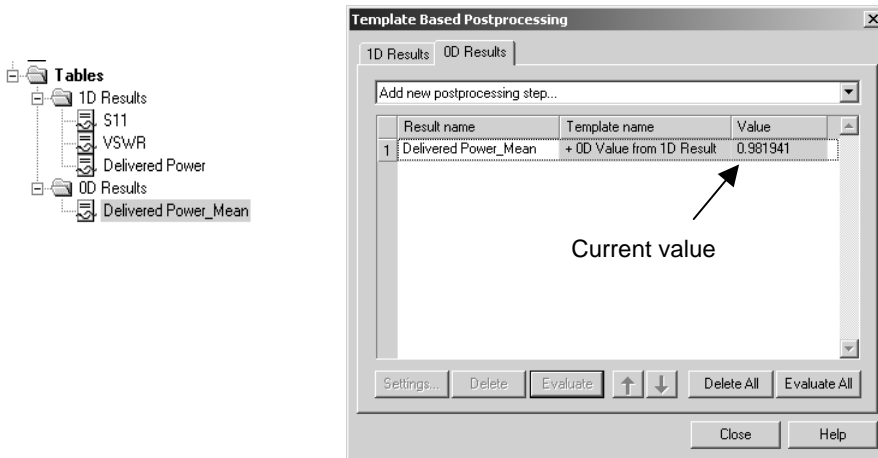
Once defined, these extra results such as VSWR and Delivered Power will be always recorded automatically. For example, if we run a Parameter Sweep, we see that all templates are automatically considered as watches. Thus after a parameter sweep we can directly compare VSWR or Delivered Power:



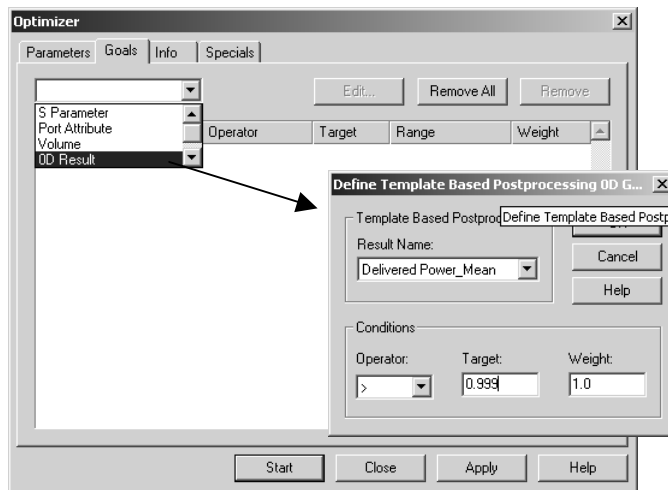
During an optimization process we might be interested in the best available power delivery broadband. Therefore we will define a 0D Result, calculating the mean value of the Delivered Power.



0D Results are also written and displayed in the Tables-folder of the navigation tree (necessary for multiple runs), however the current result values can also be quickly checked in the 0D Result page of the Template dialogue directly:



An optimization process of the structure can afterwards use any defined 0D Result for goal definition:



During the optimization we can observe the development of 1D and 0D Results in the tables.

From this example you should now be familiar with the principle concept of Postprocessing templates of CST MWS.

The template based Postprocessing system works in the same way for all 3 solvers (transient, frequency domain and eigenmode) and also for arbitrary result data, such as farfields, field plots, ... The following chapter will give an overview on the different options and their application to different data.

7.2 Overview of installed Result Templates

The following sections introduce the preinstalled templates and explain shortly their main function.

The first group of templates shows how to access farfield data through the templates:

1. 0D / Farfield
2. 1D / Farfield (single frq)
3. 1D / Farfield (broadband)

The next group introduces the access of 2D/3D field data data through the templates:

4. Evaluate all fields along all curves
5. Mix 3D Fields
6. 1D / Evaluate Field on predefined Curve
7. 1D / Evaluate Field along arbitrary Coordinates (1D Plot)
8. 0D / Evaluate Field on predefined Curve
9. 0D / Evaluate Field on predefined Face
10. 0D / Evaluate Field in arbitrary Coordinates (0D, 1D, 2D, 3D)

Afterwards, dealing with other 1D Result-files (Z, Y, Probes, Voltages, external data, ...) will be explained.

11. Calculate all Z and Y Matrices
12. 1D / Load 1D Data File (project and external)
13. 1D / Exchange Excitation
14. 1D / Group Delay Time

Finally, the powerful calculation tools are presented, giving you the opportunity to easily enter your analytic equation once and getting results after each run automatically, also defining optimization goals using any 0D Result. This group can be applied to all other templates, listed above.

15. 1D / 1D Result from 1D Result (Rescale xy, FFT, Derivation, ...)
16. 0D / 0D Value from 1D Result
17. 1D / Mix 1D Results
18. 0D / Mix 0D Results

7.3 Farfield Result Templates

There are 3 different possibilities to access farfield data: the quickest one (e.g. to define a simple farfield goal) is certainly the 0D / Farfield, where you can easily define one real number from a farfield result such as e.g. max gain 3d or 3dB-angle for a certain cut plane.

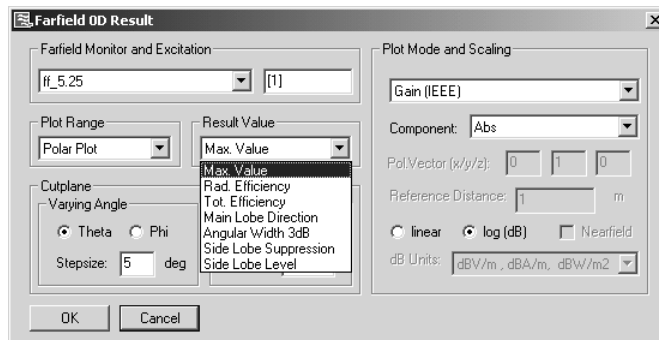
The 1D templates allow you to obtain either data versus angle (single frq) or data versus frequency (broadband), if a sufficient number of farfield frequencies has been monitored.

Starting with 1D templates, you can also switch at any time to 0D level using the general template *0D Value from 1D Result* and then be able to define a goal based on 1D farfields.

Please note: all farfield templates have a separate entry for *Excitation String* which, at this moment in time, has to be changed and entered manually, if different from port 1. For RCS simulations, the excitation string has to be changed to [pw], for excitation at port 4 please fill in [4].

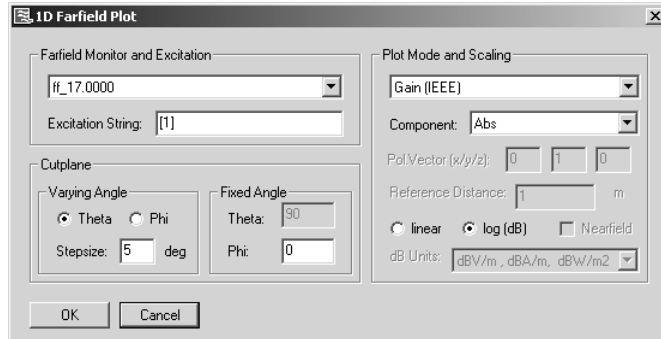
0D / Farfield

By using this template you can easily define a farfield watch or goal. It offers you a wide variety of options (3D plot, polar plot and single direction), of values, plot modes and components (including e.g. Ludwig3, which is required for Co- and Cross-Polarizaion calculation):



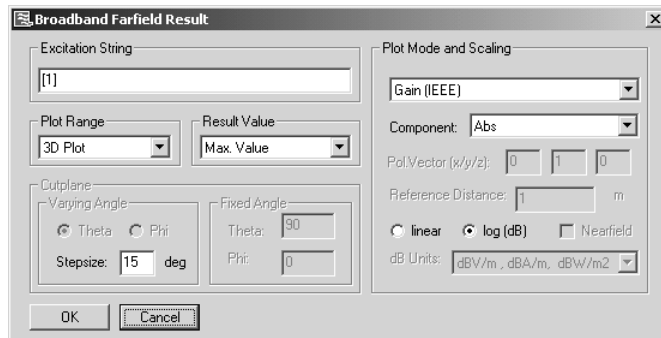
1D / Farfield (single frq)

This template allows the storage of a special cut plane (theta or phi is const.). The table will show this data as a Cartesian plot.



1D / Farfield (broadband)

This template allows the broadband storage of one special Result Value as function of frequency. All available farfield entries with the correct excitation string are considered.



The template requires the definition of multiple farfield monitors, which can be done in a comfortable way by using the macro *Farfield / Broadband Farfield Monitors* (see chapter 6.4). However, this template is not restricted to monitors defined by that macro. You can also add farfield monitors manually to have higher resolution locally.

7.4 Result Templates for 2D/3D Results

The following templates show you various options for extracting information out of 2D/3D Results: 1D plot or real value (field value at a fixed point, Curve, Face or Volume Integral).

Please note: Typically 2D/3D Results are stored as complex data for a fixed frequency. Therefore these results contain real part, imaginary part or amplitude (=magnitude) and phase respectively. Typically the real part is of highest interest (default setting). When switching to Magnitude, negative values will be considered as positive ones.

Evaluate all fields along all curves

This template performs the automatic evaluation of all fields along predefined curves.

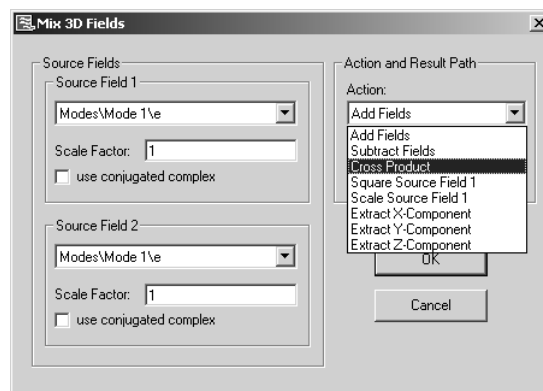
This action is typically combined with the template *Load 1D Data File* to specify the correct result from the many created curves for further operations within the template framework.

The template is also useful to obtain all possible result curves within just one mouse click.

Mix 3D Fields

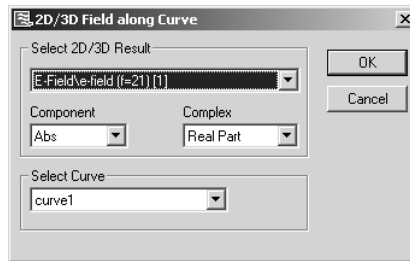
This template offers a calculator for 2D/3D Fields.

The result of this template is a new 3D Field, which will be written in 2D/3D Result Tree. The location in the tree is specified in the dialogue (default name is Combine). The newly created 3D result can be specified in other templates for 1D Plotting, curve-, face- or volume Integration.



1D / Evaluate Field on predefined Curve

This template extracts field data along predefined curves. One of the 3D Results can be selected as well as component and complex properties. The template is useful to create 1D plots of 3D fields.

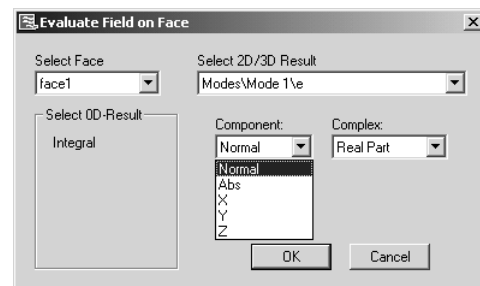
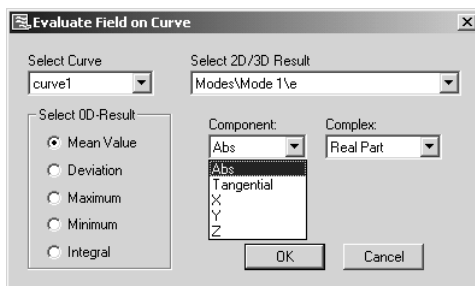


0D / Evaluate Field on predefined Curve

0D / Evaluate Field on predefined Face

These templates directly produce a 0D Result based on the curve or face definition.

Note: The curve-related value can also be obtained by plotting the 1D curve first and afterwards calculating the 0D Result by using the template *0D Value from 1D Result*. However, in that case, the length unit will not be automatically considered.

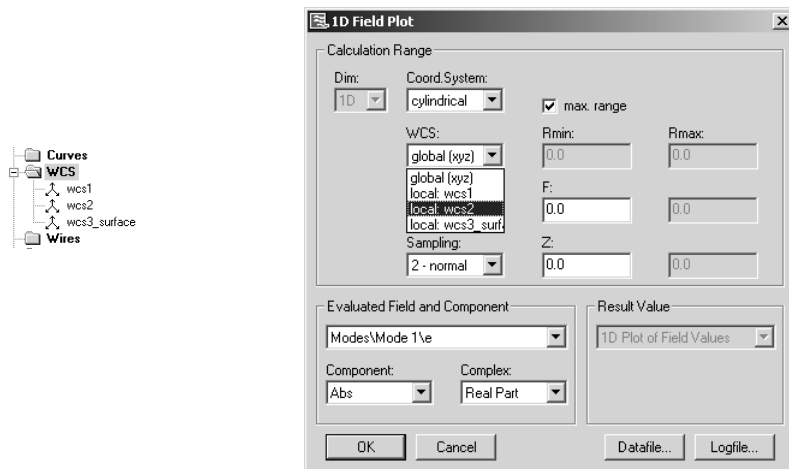


1D / Evaluate Field along arbitrary Coordinates (1D Plot)

This template extracts field data along arbitrary coordinates. It offers you the option of plotting the result value versus coordinate value (x,y,z,phi,...) in contrast to predefined curves, where the plot ordinate is curve length (always starting from zero).

It is necessary to define a curve first: assuming you wish to plot E-Field along the z-axis then you can use this template. The 1D Plot works in various coordinate systems (cartesian, cylindrical and spherical) in combination with different working coordinates (global, local1, local2, ..).

If you have defined and stored names for different local coordinate systems, you can do postprocessing along these local coordinates (e.g. plot along angle phi, defined in wcs1).



0D / Evaluate Field in arbitrary Coordinates (Field Values, 1D, 2D, 3D)

Similarly to the previously explained 1D Plot in arbitrary coordinates, this 0D template directly extracts the corresponding values.

The calculation range's dimension can be 0D (single point), 1D (line), 2D (face) or 3D (volume). Within each of this ranges (besides 0D) you can extract the following field values:

- Integral Value
- Mean Value
- Maximum Value
- Minimum Value
- Deviation (checking homogeneity)

7.5 Result Templates for arbitrary 1D Results

This section shows you how to load and connect different types of 1D data into the template framework in order to use them for further calculations, such as watches or goals.

Calculate all Z and Y Matrices

This template performs the automatic evaluation of all Z and Y curves in the tree.

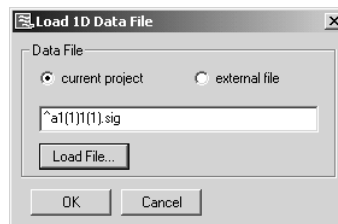
This action is typically combined with the template Load 1D Data File to specify the correct result out of the many created curves.

Here the table will be filled with $zam1(1)1(1)$ data.

1D / Load 1D Data File (project and external)

By using this template, you are able to load any data into the template system, even if the comfortable way with a separate dialogue is not available. You can browse any file (project or external), thus being able to define watches and goals based on them.

The following dialogue shows how you would load the amplitude of S11. Of course we have a more comfortable way for S11 with the 1D template *S Parameter*. However for probes, voltages, Y and Z results, this template can be used.



Example: Optimization to given external goal-curve.

Once you are interested in optimizing your device to match a given Goal-Curve (1D), which exists on disk in the form of an ASCII-table, you can link this curve as external file to the project and afterwards calculate the difference of both results (e.g. $\text{abs}(\text{current-goal})$). The Mix1D Template can be considered for this. An integration of this difference would then define the 0D Result which is used as to be minimized during the optimization process.

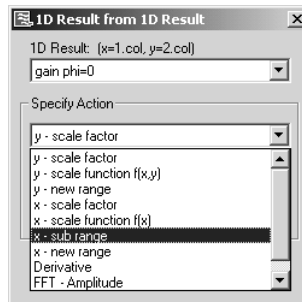
7.6 Calculation Tools for Result Templates

This section shows you the calculation possibilities which can be applied to any data in the Template framework. Using these templates you can split your equation into several smaller parts, thus being able to check and compare intermediate results as well. This gives you a powerful debugging possibility, even available during parameter sweeping and optimization runs.

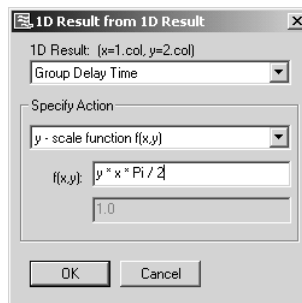
All these calculation tools have in common, that they consider any 1D Result data as a simple xy-curve, without any knowledge of units or physical meaning. Doing this, you have the advantage of having full control on all factors. However, sometimes a scaling unit needs to be added manually. (e.g. for line integration with unit mm, 1e-3 has to be multiplied to get back to the SI-system).

1D / 1D Result from 1D Result (Rescale xy, FFT, Derivation, ...)

This template is able to create a new 1D Result based on the original one, which might be useful to rescale x or y (also using analytical functions to e.g. go to logarithmic x-range: $x_{\text{new}}=\log(x)$) or to perform FFT or derivative on an existing curve.

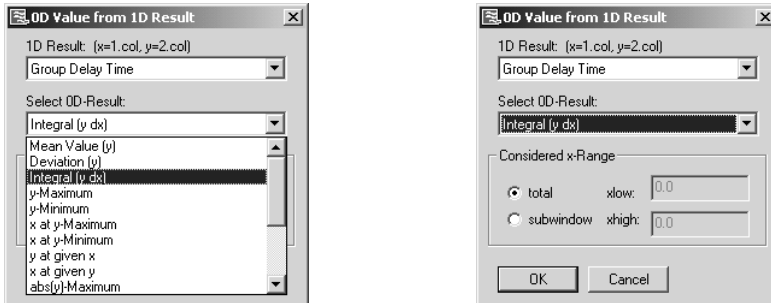


Example: For a lossfree 1-port resonator the Q-value should be calculated from the Group Delay Time, since the S11 amplitude does not show any resonance ($Q = t_{\text{group}} * 2\pi \text{ frq} / 4$). In a first step the Group Delay Time vers. Frq is calculated from S11-Phase as a 1D template, afterwards the following scale function will calculate Q out of it (here $x=\text{frq}$):



0D / 0D Value from 1D Result

Having a fixed 0D value to compare and to prove the device's quality is very often the most accurate criterion. This template gives you the opportunity to get the correct real value from your 1D Result in order to use it later as a goal or just as a watch.



Please note: The quality of an optimization process very much depends on the correct goal definition. Especially there are differences between the following similar looking goals and also between their optimization results !

- Maximize Mean Value (y)
- Maximize y-Maximum
- Maximize y-Minimum

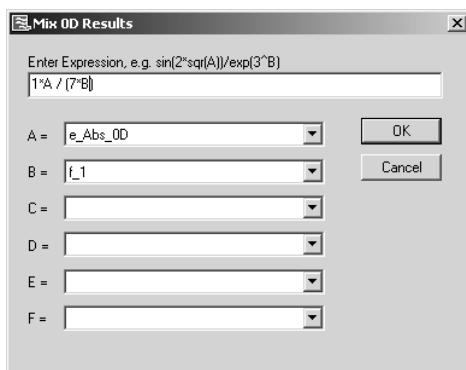
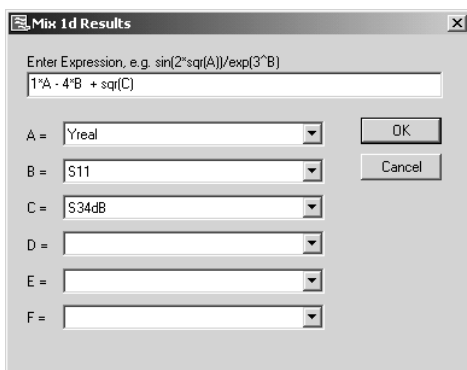
All three goals do maximization, however with the following difference:

- Maximize Mean Value (y) does not take special care on best or worst case
- Maximize y-Maximum is only looking on the best case
- Maximize y-Minimum is only looking on the worst case

1D / Mix 1D Results

0D / Mix 0D Results

Both for 1D as well as for 0D Results exists a calculator to type in analytic expressions in VBA-syntax. In section 7.1 this calculator is used to calculate $1-S_{11}^2$.



Chapter 8 - Network Installation

Network installations allow “floating” licenses to be shared between several computers connected in a local area network.

In order to make such license sharing possible, one computer must be configured to act as a so-called *license server*. The other computers (*clients*) must have access to this computer via a TCP/IP connection.

CST MICROWAVE STUDIO®'s software itself may then be installed locally on any client or installed globally on a particular computer in the network, the so-called *application server*.

The license server and the application server can be installed either on the same or on two different computers.

Please note that Windows NT 4, Windows 2000 or Windows XP require that you have administrator privileges during the entire installation procedure. Please ask your network administrator for assistance.

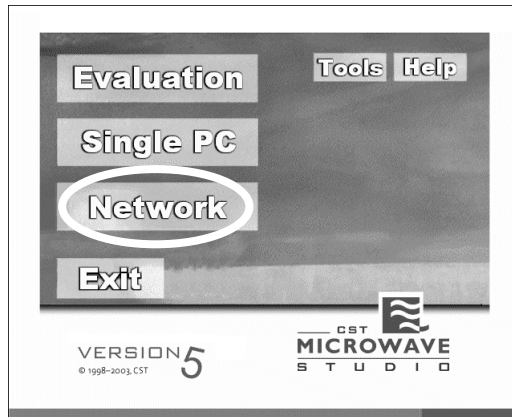
8.1 License Server Installation

The installation of network licenses is quite straightforward. As a first step, you need to install the license server on a computer in your network.

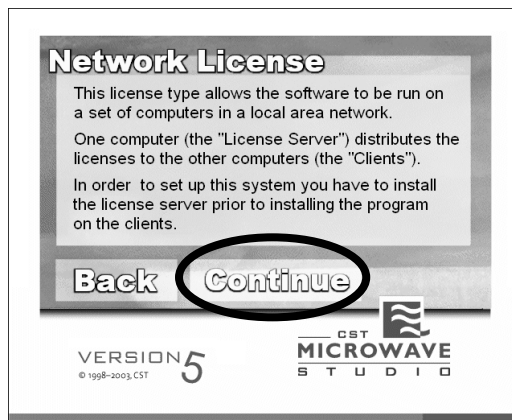
Before the software can be installed, the hardlock (dongle) has to be connected to the printer port of the license server. To do this, please switch off your computer, remove the printer cable (if any), connect the hardlock to the printer port and make sure that it fits properly. You may then re-attach the printer cable.

Note: If you have other hardlock-protected software products installed, some interference between the hardlocks is possible. In such cases, please remove all other hardlocks and re-try the installation of CST MICROWAVE STUDIO®. Please contact technical support for more information.

As a next step, please run the setup program from the CD by calling `setup.exe` (if the program does not start automatically after inserting the CD into the drive). Then the following menu will appear:



Please select the menu item *Network* by clicking on it with the mouse. Afterwards the menu will change as shown in the picture below:

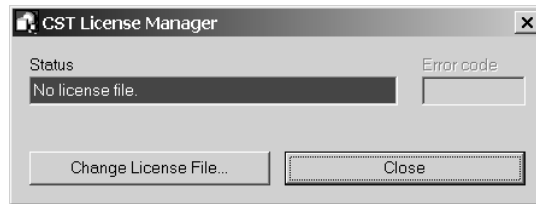


You can simply dismiss this dialog box by pressing the *Continue* button which will lead you to the actual network installation menu:



Here you should select the option *Install License Server*. This will start the license server's installation program. Please read the messages boxes carefully and follow the instructions on the screen.

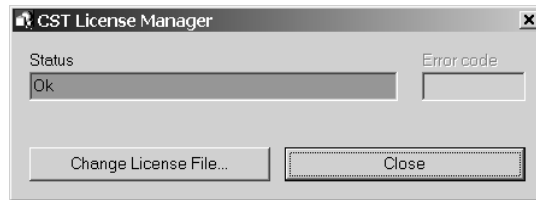
When the installation has finished you need to reboot your computer. Once this is complete, the license server configuration dialog box will open automatically:



Please note: This dialog box can also be opened at any time by selecting the *License Manager* entry in the *CST License Manager* section of the *Start* menu. Furthermore, it will open automatically whenever the license file expires.

All you need to do here is to press the *Change License File* button and specify the location of your license file. You should have received this file either by e-mail or on a license floppy disk.

Once this step is completed properly, the license server will automatically be started and the dialog box will look as follows:



Please contact technical support if you encounter problems during the license manager setup.

8.2 Software Installation

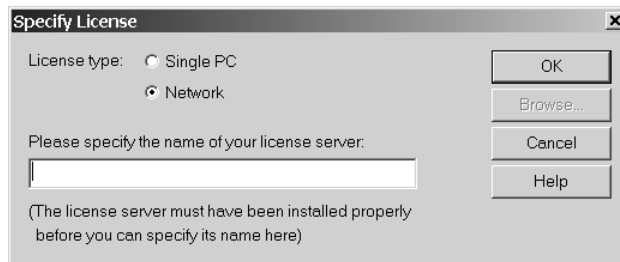
After the license server has been set up properly, you can install CST MICROWAVE STUDIO®'s software on the client PC's or on your application server.

The installation of the software on a particular PC is simple. Please insert the installation CD into its drive and select to install a network license. In the same way as described above you should now open the network installation menu:



Please select the option *Install CST MicroWave Studio* before you follow the instructions on the screen in order to install the software. Finally the computer needs to be rebooted.

When you call CST MICROWAVE STUDIO® for the first time (you still need administrator privileges), the following dialog box will appear:



In this dialog box, please select the *License type* “Network” before you enter your license server’s network computer name in the corresponding field. Afterwards press the *Ok* button to confirm these settings. If everything has been installed properly, CST MICROWAVE STUDIO® should start now. Please contact the technical support if you encounter any problems.

Please note: The installation of a **small network license** requires a *client hardlock* (dongle) to be installed on every computer which should be able to run CST MICROWAVE STUDIO®. The hardlock which is installed on the license server is referred to as the *server hardlock*. Please make sure that you do not mix up these hardlocks in order to avoid licensing problems.

8.3 Installation by Using an Application Server

By installing CST MICROWAVE STUDIO® on an application server you can avoid time consuming installations of the entire program on each client PC. Please make sure that every computer in the network which should be able to run CST MICROWAVE STUDIO® has file access to its installation directory on the application server.

The installation directory contains a folder named *Network Setup*. In order to allow a client PC to run CST MICROWAVE STUDIO® please open this directory in the Windows Explorer. Afterwards please run the setup.exe program and follow the instructions on the screen. This will install all necessary components on the client PC.

The *Start* menu will then contain the same entries as if the entire software had been installed on the client. Please note that whenever you start CST MICROWAVE STUDIO® now, you actually run the software from the application server's hard disk.

If your network slows down this procedure or is too unstable you should install the software on every client PC rather than use an application server. Application servers, on the other hand, have the advantage that software updates need only be installed once (ensuring that every computer in the network uses the same version of the software).

8.4 Network Distributed Computing Installation

Parameter sweeps and optimizations often require many individual simulations for different parameter configurations. Instead of running all these simulations step by step on a single computer, the runs may also be performed in parallel on several computers within the network.

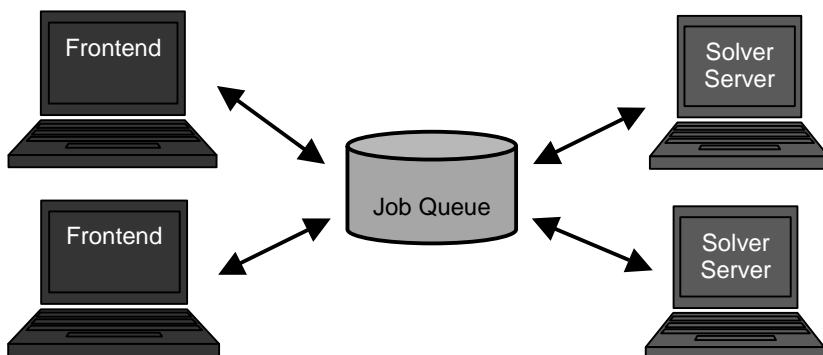
Furthermore, the transient solver requires a solver run for every port / mode excitation which can also be performed in parallel.

This kind of parallel processing results in a linear speedup with the number of available computers as compared to sequential computations.

CST MICROWAVE STUDIO® supports such distributed processing by introducing so-called *simulation frontends* and *solver servers*.

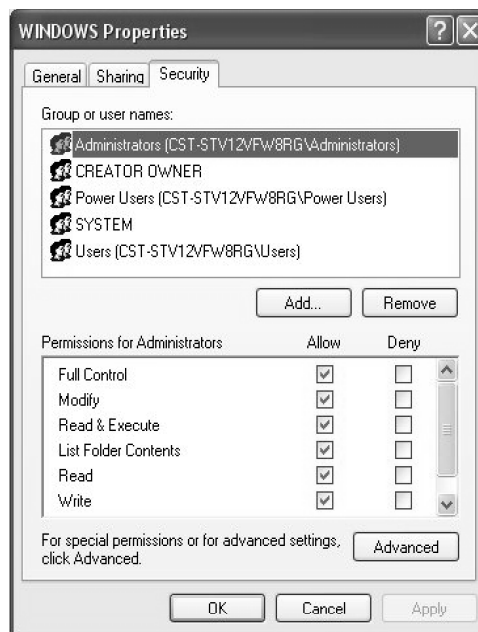
Each frontend can submit jobs to a common job queue directory within the network. Several computation servers will frequently check this queue. Once a new job has been submitted, a server will pick up this job, simulate it and allow the submitting frontend to access its results. Therefore, multiple jobs can be processed at the same time by different computation servers. The submitting frontend cares about the simulation control and collects the available results.

The following picture illustrates this system:



This section will cover the installation and the setup of the distributed computing system. Please make sure that you have administrator privileges before you continue with the following steps.

The first step is to create a folder on one of the computers in the network which will act as the job queue. Please ensure that everyone has full file access to this folder. Furthermore, the folder must be visible from every computer which should be used as either frontend or solver server. You can specify the access within the Security tab of the Folder Properties Dialog, as you can see in the following picture. On some Windows XP systems, you have to enter the “Folder Properties” of the Control Panel, in order to make the mentioned security tab visible.

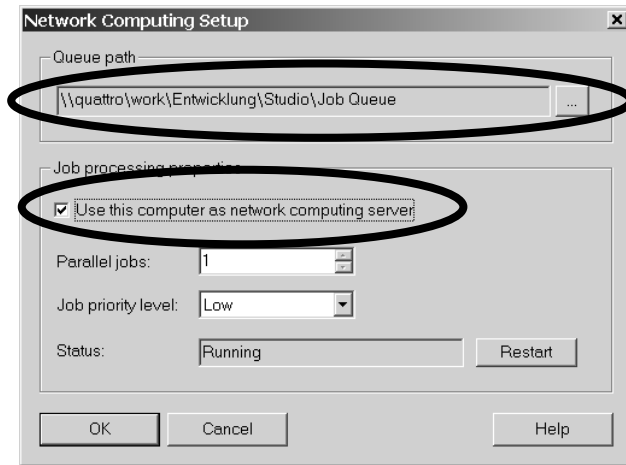


As a next step, you need to install CST MICROWAVE STUDIO® on every computer in your network which should act as a solver server.

In case that you have a node locked license, please refer to the *Getting Started* manual for more information about how to install the software. Please make sure to connect the hardlock to the corresponding computer before running the installation.

For floating licenses, please install the software as described in the previous section. Please note that in case of a small network license, you will also need to connect a hardlock to the particular computer during the installation.

Once the software is installed and is properly running, you have to configure the service which launches the CST MICROWAVE STUDIO® program on the computation server. Therefore start CST MICROWAVE STUDIO® and choose *File* ⇒ *Network Computing* from the main menu:



As a first step, you need to specify the queue path (the path to the network directory which has been chosen previously) by pressing the “...” button in the *Queue path* frame.

Afterwards, please check the option *Use this computer as network computing server* and specify the maximum number of parallel jobs on this system (usually 1). If the computer has more than one processor and sufficient memory, it can make sense to allow more than one simultaneous simulation run. As a next step, you should specify the Job priority level which should usually be set to *Low*.

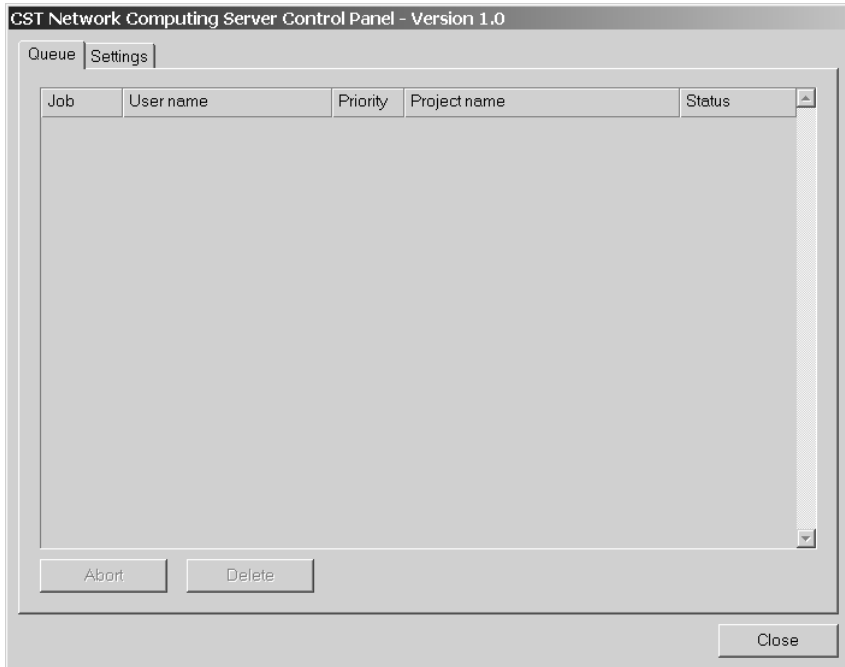
Finally, you can install and start the service on the computer by pressing the *Ok* button. After a couple of seconds, the following dialog box should appear



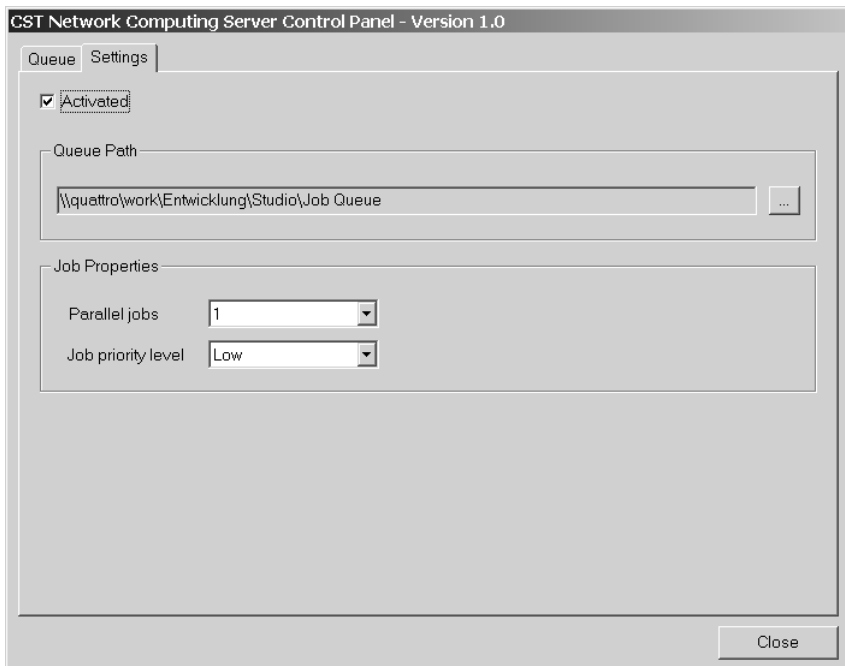
and the corresponding control panel icon should be shown in the task tray:



Once the service is set up properly, you can now close CST MICROWAVE STUDIO®. Please note that the hardlock is no longer required and can be disconnected. You can open the service control panel at any time by clicking on it's icon in the task tray:



The *Queue* page displays the current status of the queue (which is still empty). The *Settings* page allows you to modify the services' settings as shown below:

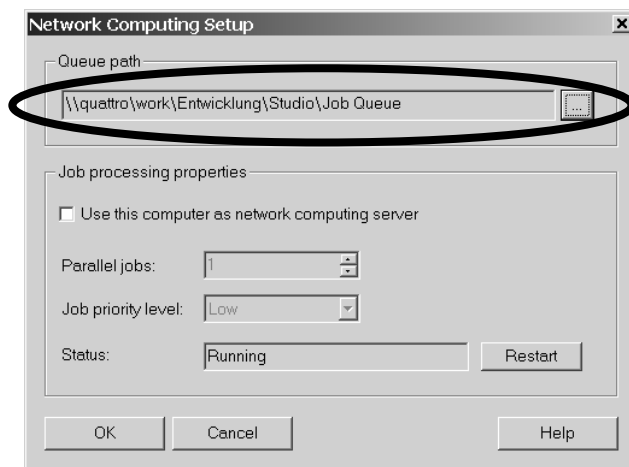


The only new option here is the *Activated* switch which allows to pause the job processing on the computer without de-installing the service. The activation status of the service will be reflected by the icon in the task tray.

You can now close the control panel. The service will be launched automatically whenever the computer is booted and will process jobs even if no user is logged onto the system.

Please repeat the steps above for every computer in your network which should act as a computation server. Assuming that all computation servers have been properly set up, we will now focus on how to use the distributed computing.

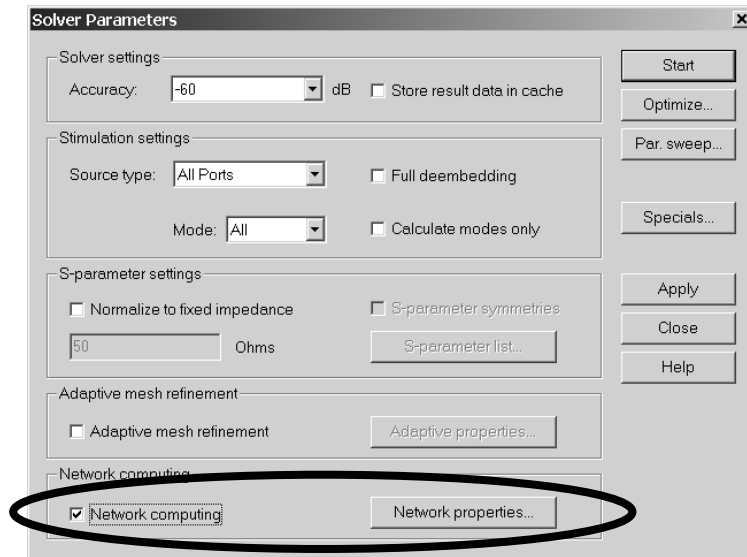
Before you can submit jobs into the queue, you have to specify the queue directory for the frontend, too (*File* ⇌ *Network Computing*):



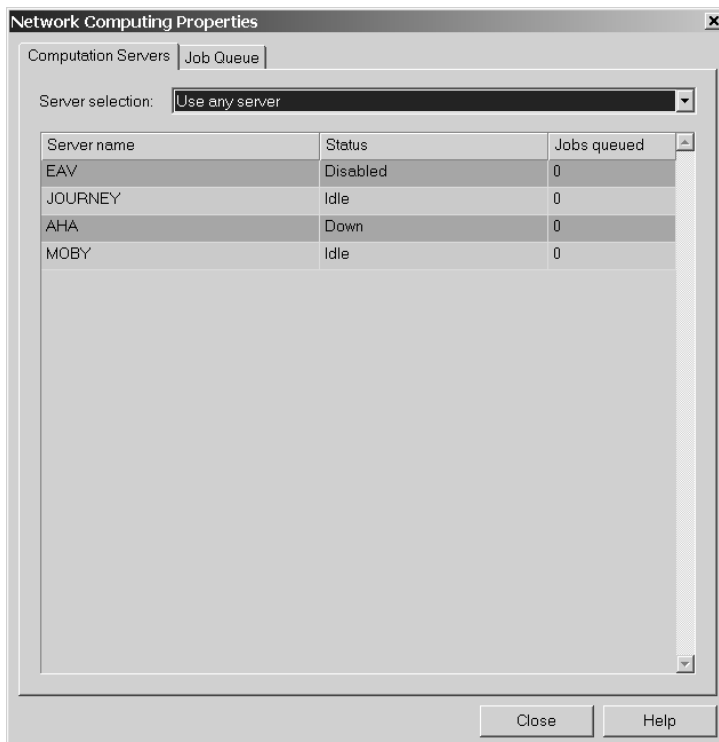
Please specify the name of the network queue here, but do not check the *Use this computer as network computing server* box unless you wish to install the service on this computer as well. Please note that you could use the same computer as a frontend and a solver server for the distributed computing feature.

The queue path will be stored in the Windows registry so you should only need to specify it once.

The transient solver control dialog box contains a frame called *Network computing*:



You can inspect the status of the currently available computation servers and the queue itself by pressing the *Network properties* button (which becomes available once the *Network computing* box is checked):

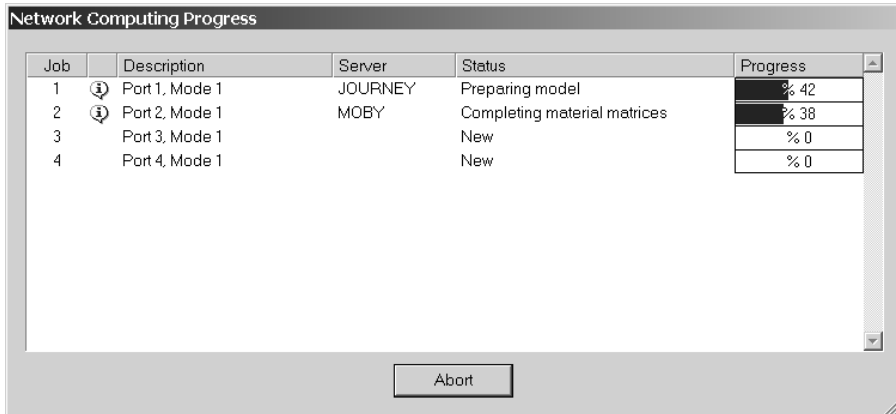


In this dialog box you can restrict the submitted jobs to run on a certain subset of computation servers only. The default setting *Use any server* allows the next available

solver server to automatically pick up the job. Please refer to the online documentation for more information about the available options.

Once the Network computing button is checked in the transient solver control dialog box, the next simulation will automatically be submitted to the network. This means that in case of a multiple port / mode excitation sequence, each excitation is submitted as an independent job to the network queue and therefore can be calculated in parallel.

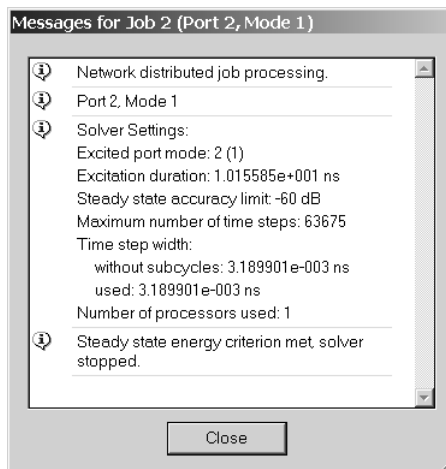
While the jobs are running or waiting for execution, a progress window will be shown informing you about the status of the individual jobs:



Job	Description	Server	Status	Progress
1	Port 1, Mode 1	JOURNEY	Preparing model	% 42
2	Port 2, Mode 1	MOBY	Completing material matrices	% 38
3	Port 3, Mode 1		New	% 0
4	Port 4, Mode 1		New	% 0

Abort

You can click on the icons displayed in the second column in order to open another dialog box containing information and warning or error messages of the particular job:



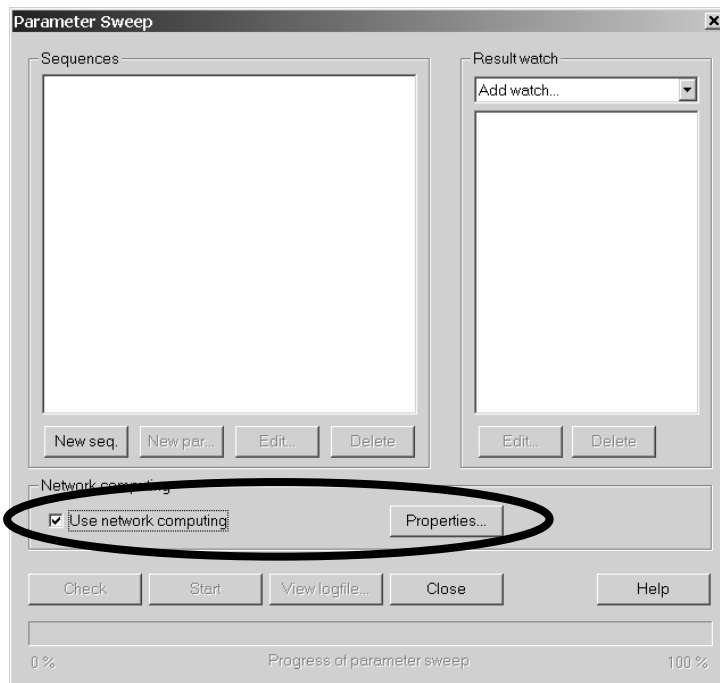
Messages for Job 2 (Port 2, Mode 1)

- Network distributed job processing.
- Port 2, Mode 1
- Solver Settings:
 - Excited port mode: 2 (1)
 - Excitation duration: 1.015585e+001 ns
 - Steady state accuracy limit: -60 dB
 - Maximum number of time steps: 63675
 - Time step width:
 - without subcycles: 3.189901e-003 ns
 - used: 3.189901e-003 ns
 - Number of processors used: 1
- Steady state energy criterion met, solver stopped.

Close

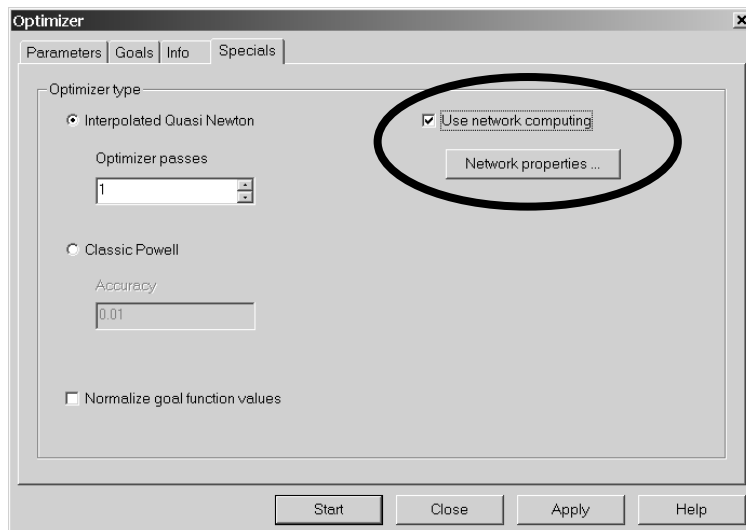
While this division of solver runs into individual port / mode excitation jobs is only available for the transient solver, the parallel job processing of parameter sweeps or optimizations is available for the eigenmode and frequency domain solvers as well.

For parameter sweeps, the distributed computing is activated by checking a *Use network computing* option in the corresponding dialog box:



Once the network computing is activated, the jobs for individual parameters are submitted to the queue. If the *Network computing* option is checked in the transient solver control dialog box, too, each of these network jobs may again submit secondary jobs to the queue.

In the same way, the network computing for optimizations can be activated from within the optimizer dialog boxes *Specials* tab:



We recommend that you refer to the online documentation for more information about this powerful feature or visit a specialized training class on this topic. Please contact your support center for more information.



© CST, 2003. ALL RIGHTS RESERVED.